EE16B, Spring 2018
UC Berkeley EECS

Maharbiz and Roychowdhury

Lectures 4B & 5A: Overview Slides

Linearization and Stability

Linearization

- Approximate a nonlinear system by a linear one
  - (unless it's linear to start with)
- then apply powerful linear analysis tools
  - gain precise understanding → insight and intuition

Linearization

- Approximate a nonlinear system by a linear one
  - (unless it's linear to start with)
  - then apply powerful linear analysis tools
    - gain precise understanding → insight and intuition
- Consider a scalar system first
  - in state space form with additive input (for simplicity)

$$\frac{d}{dt}x(t) = f(x(t)) + u(t)$$

Linearization

- Approximate a nonlinear system by a linear one
  - ➔ (unless it's linear to start with)
  - then apply powerful linear analysis tools
    - ➔ gain precise understanding → insight and intuition
- Consider a scalar system first
  - in state space form with additive input (for simplicity)

$$\frac{d}{dt}x(t) = f(x(t)) + u(t)$$

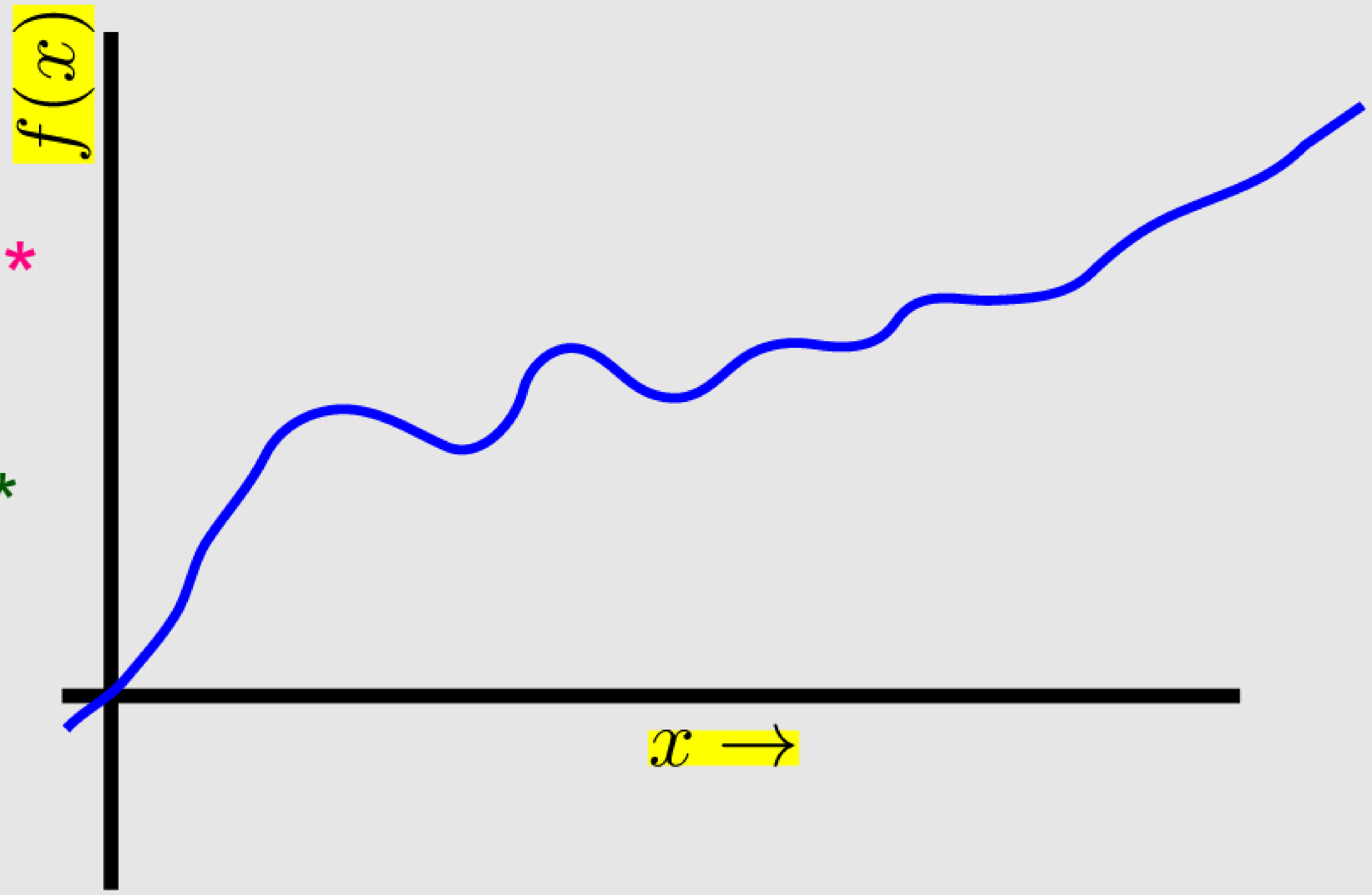
  - step 1: choose DC input u*
    - ➔ find DC soln. x*

$0 = f(x^*) + u^*$: solve for x*

Linearization

- Approximate a nonlinear system by a linear one
  - (unless it's linear to start with)
  - then apply powerful linear analysis tools
    - gain precise understanding → insight and intuition
- Consider a scalar system first
  - in state space form with additive input (for simplicity)

$$\frac{d}{dt}x(t) = f(x(t)) + u(t)$$



  - step 1: choose DC input u*
    - find DC soln. x*

$0 = f(x^*) + u^*$: solve for x*

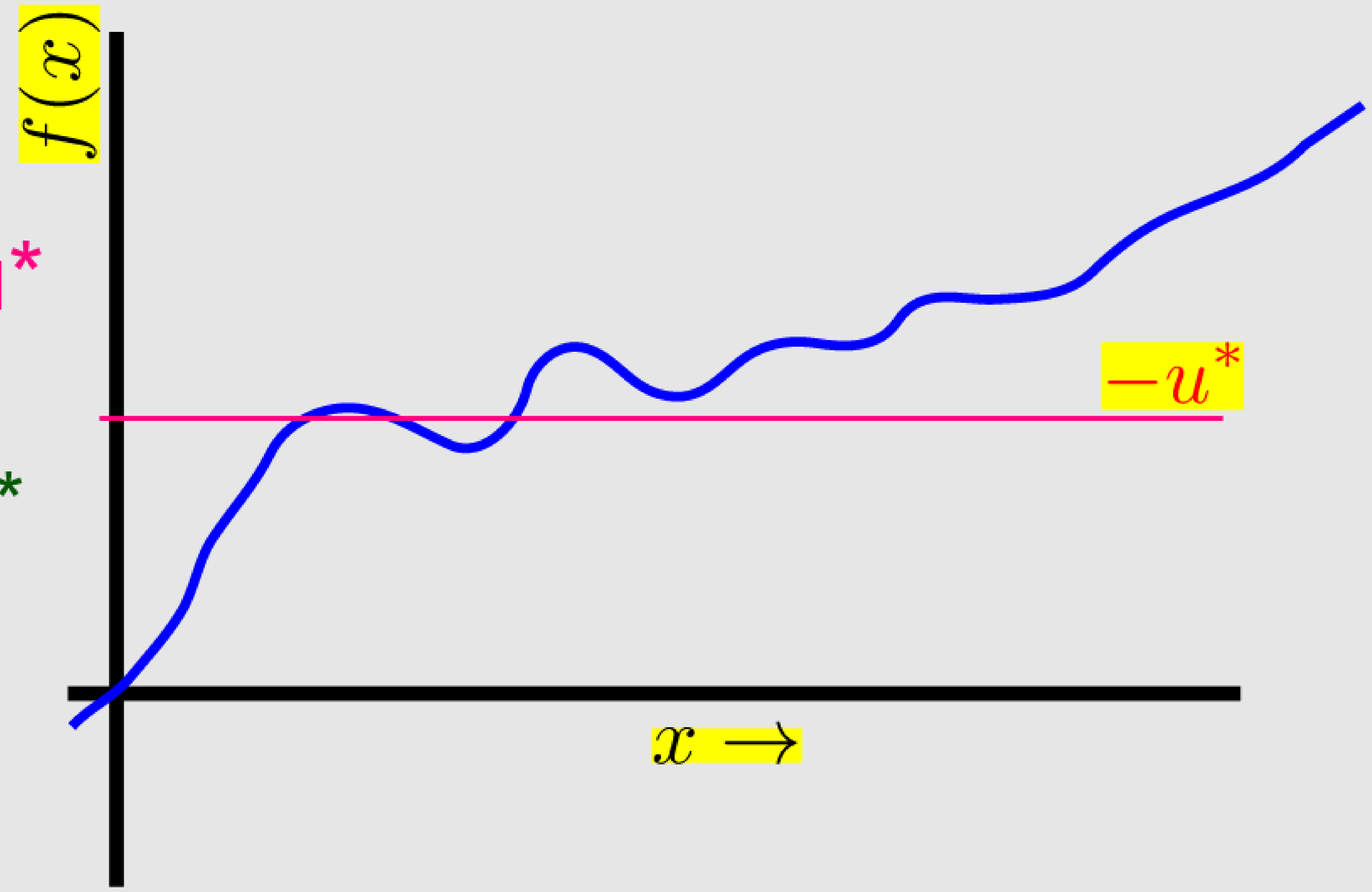
Linearization

- Approximate a nonlinear system by a linear one
  - ➔ (unless it's linear to start with)
  - then apply powerful linear analysis tools
    - ➔ gain precise understanding → insight and intuition
- Consider a scalar system first
  - in state space form with additive input (for simplicity)

$$\frac{d}{dt}x(t) = f(x(t)) + u(t)$$

  - step 1: choose DC input u*
    - ➔ find DC soln. x*

$0 = f(x^*) + u^*$: solve for x*



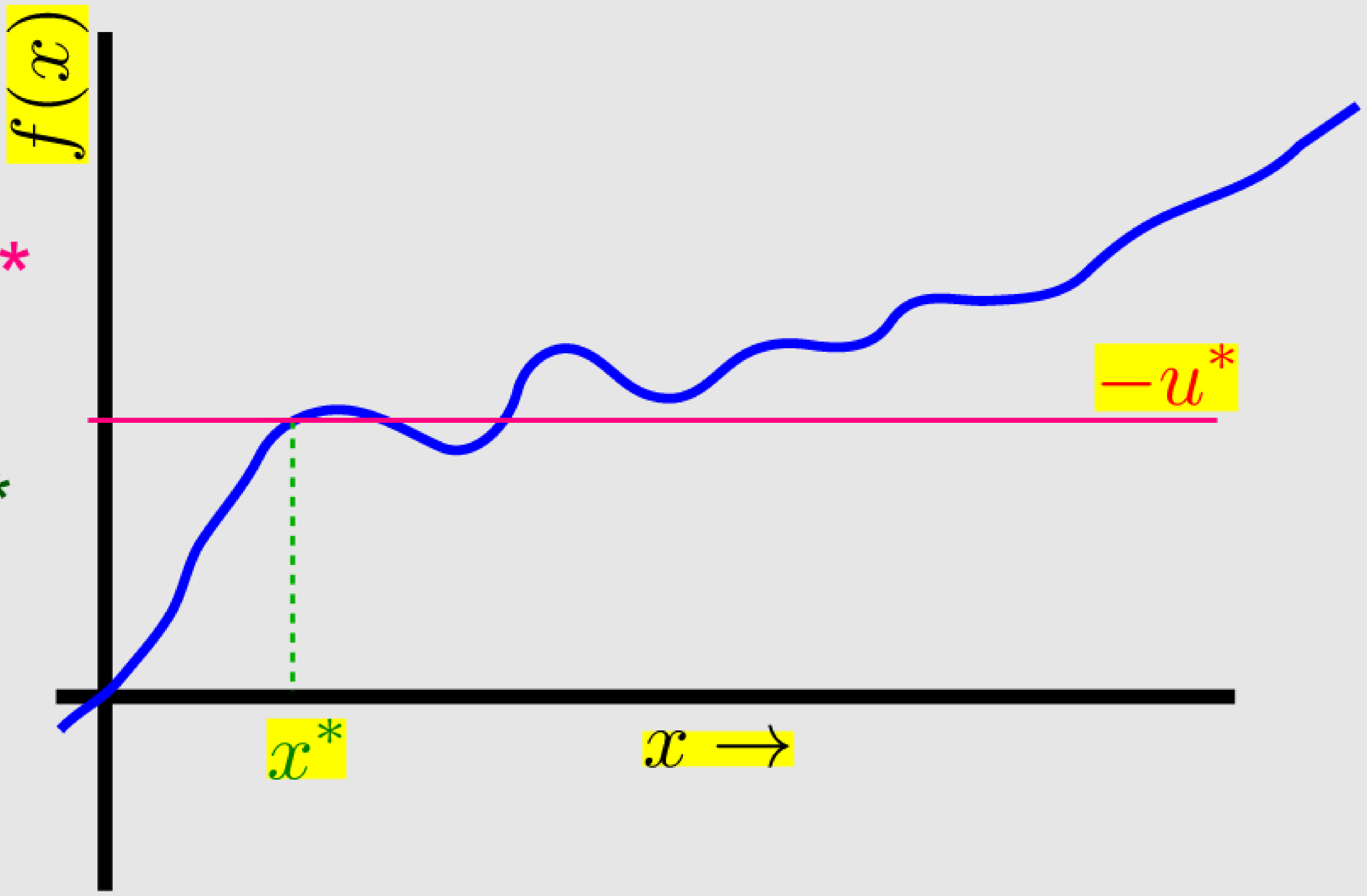
Linearization

- Approximate a nonlinear system by a linear one
  - ➔ (unless it's linear to start with)
  - then apply powerful linear analysis tools
    - ➔ gain precise understanding → insight and intuition
- Consider a scalar system first
  - in state space form with additive input (for simplicity)

$$\frac{d}{dt}x(t) = f(x(t)) + u(t)$$

  - step 1: choose DC input u*
    - ➔ find DC soln. x*

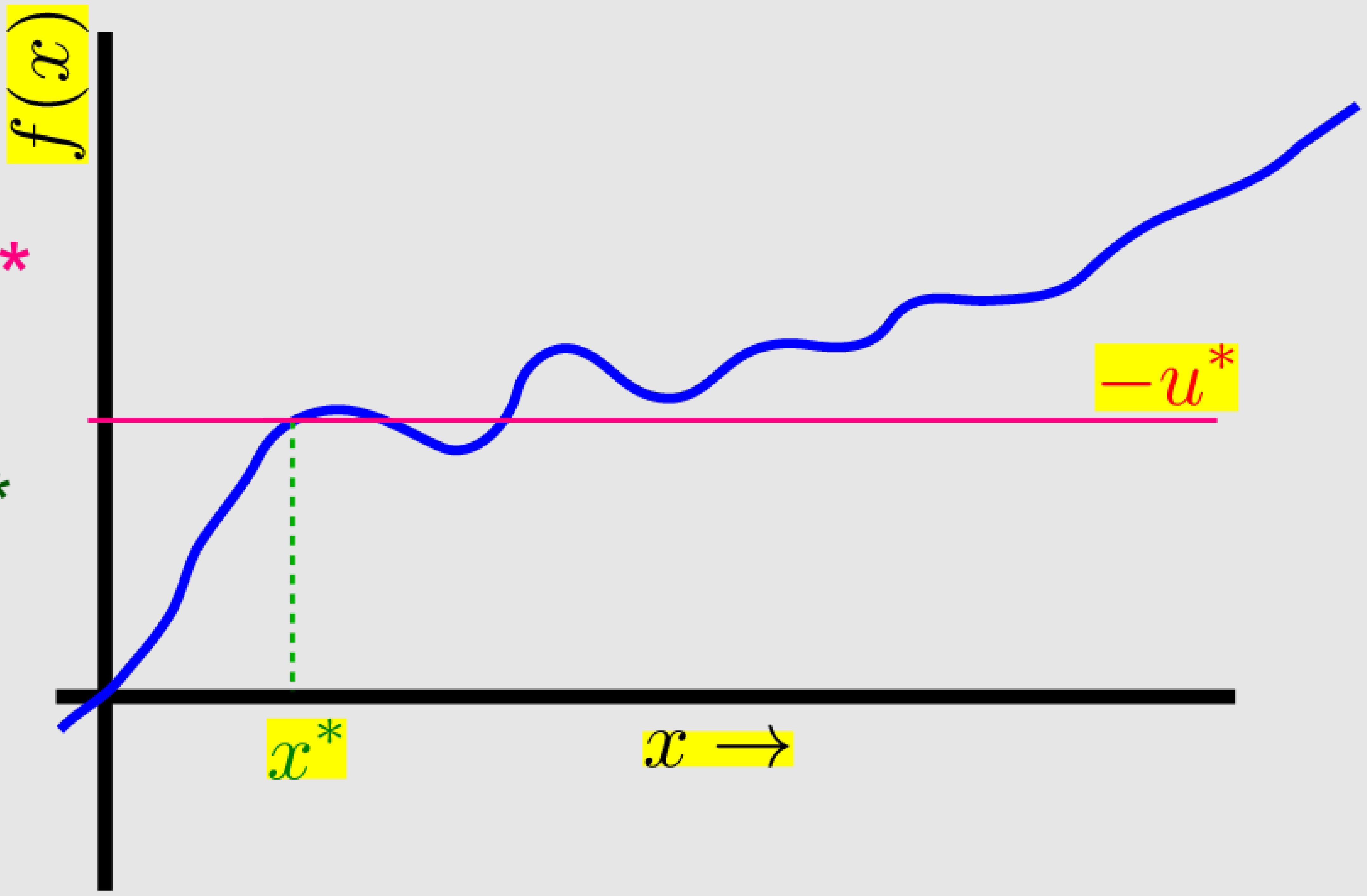
$0 = f(x^*) + u^*$: solve for x*



Linearization

- Approximate a nonlinear system by a linear one
  - → (unless it's linear to start with)
  - then apply powerful linear analysis tools
    - → gain precise understanding → insight and intuition
- Consider a scalar system first
  - in state space form with additive input (for simplicity)

$$\frac{d}{dt}x(t) = f(x(t)) + u(t)$$



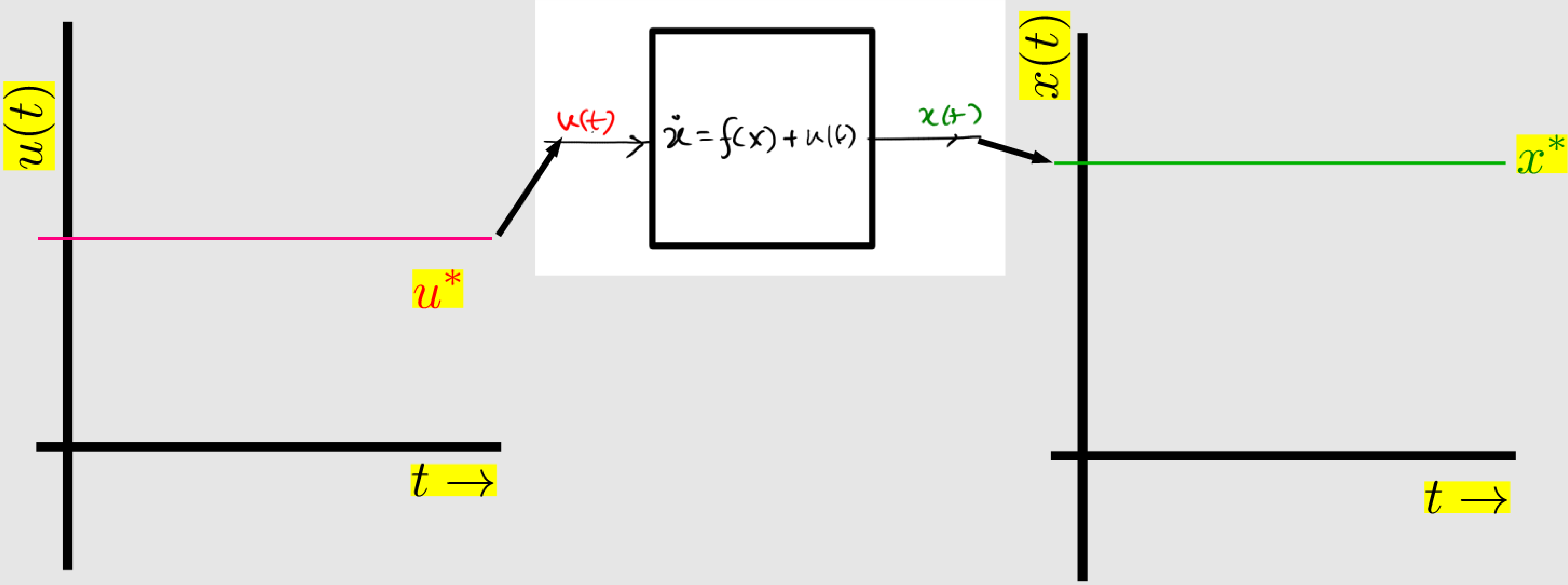
  - step 1: choose DC input u*
    - → find DC soln. x*

      $0 = f(x^*) + u^*$: solve for x*

    - → x* is an equilibrium point
      - aka DC operating point
    - for input u*

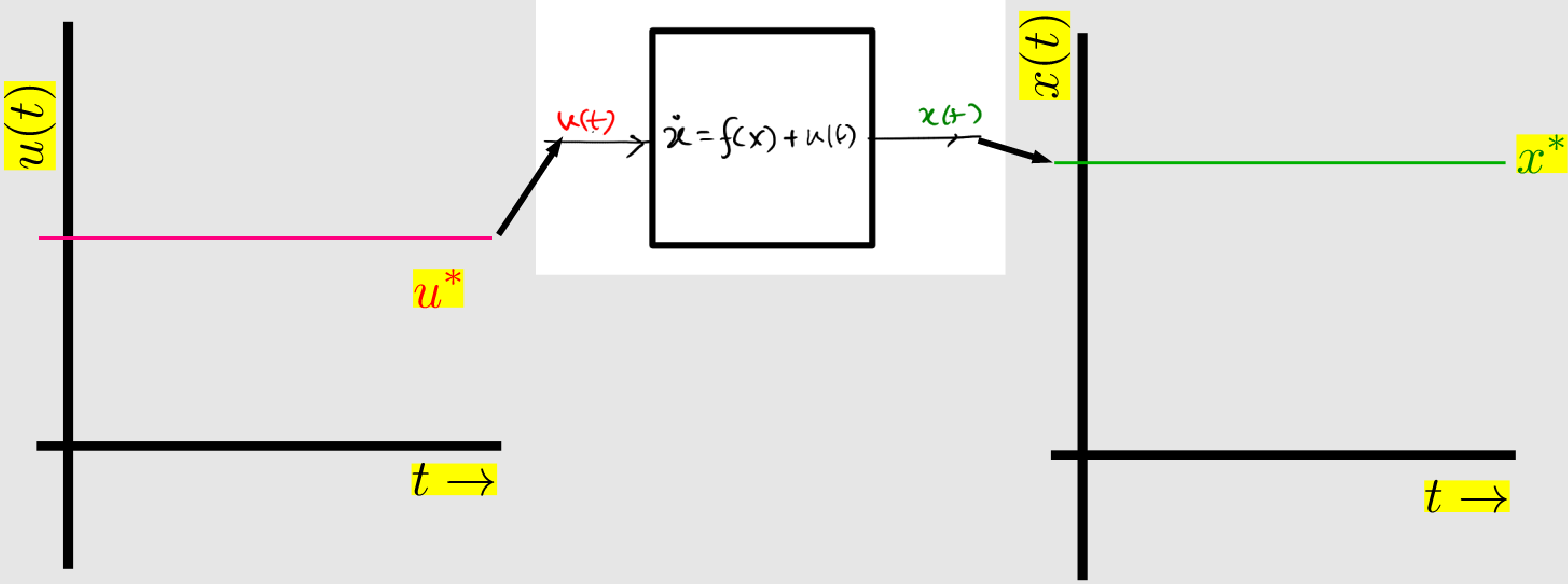
Linearization (contd. - 2)

- DC operation (equilibrium), viewed in time



Linearization (contd. - 2)

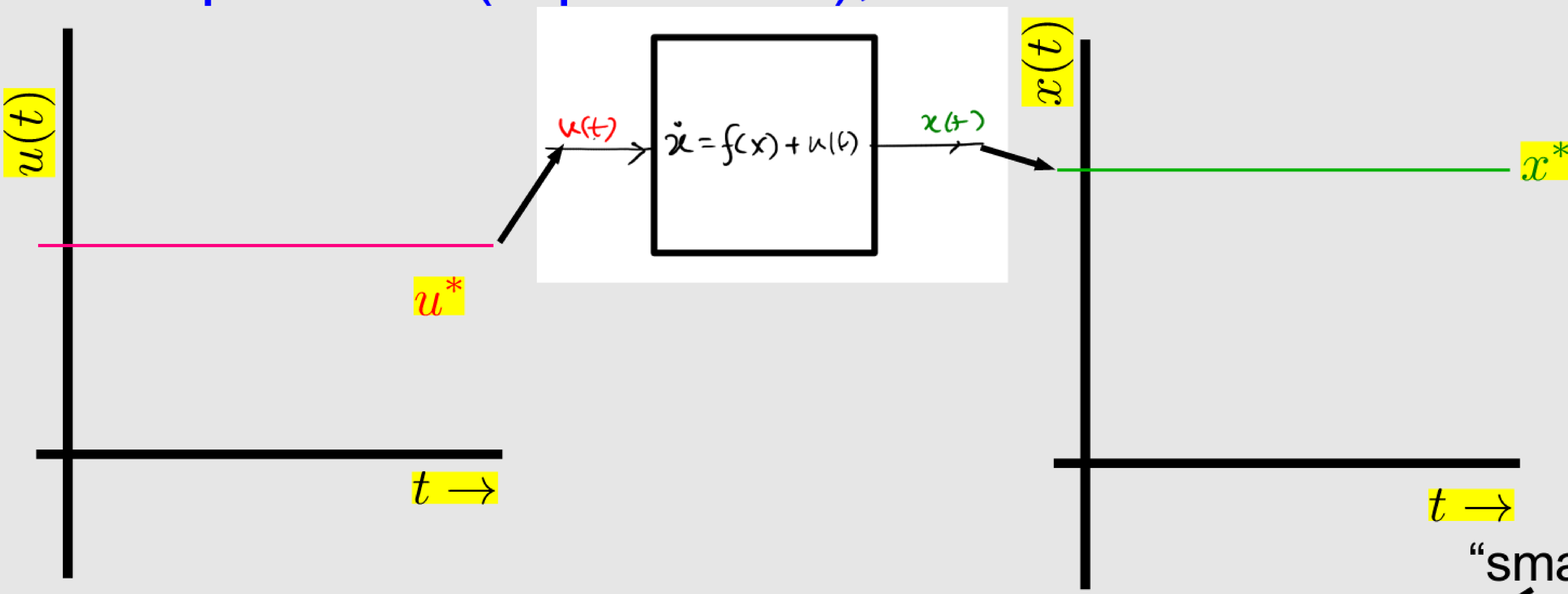
- DC operation (equilibrium), viewed in time



- Now, **perturb** the input a little: $u(t) = u^* + \Delta u(t)$

Linearization (contd. - 2)

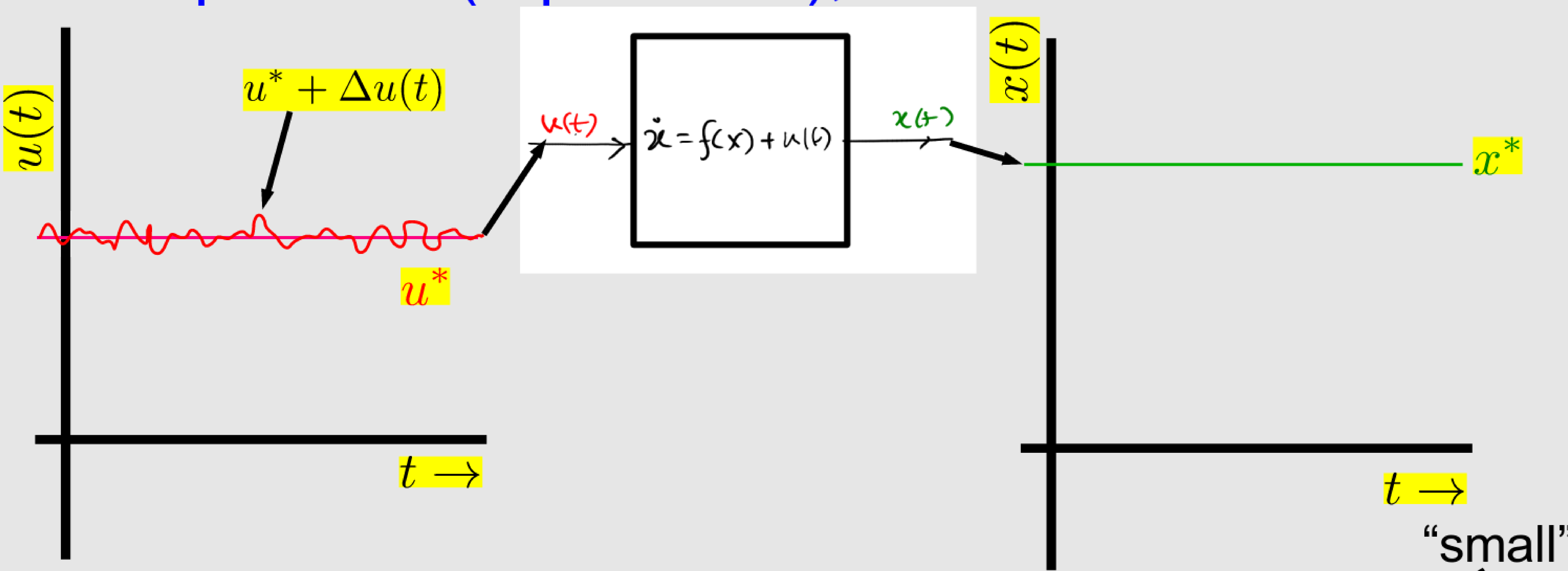
- DC operation (equilibrium), viewed in time



- Now, **perturb** the input a little: $u(t) = u^* + \Delta u(t)$ "small"

Linearization (contd. - 2)

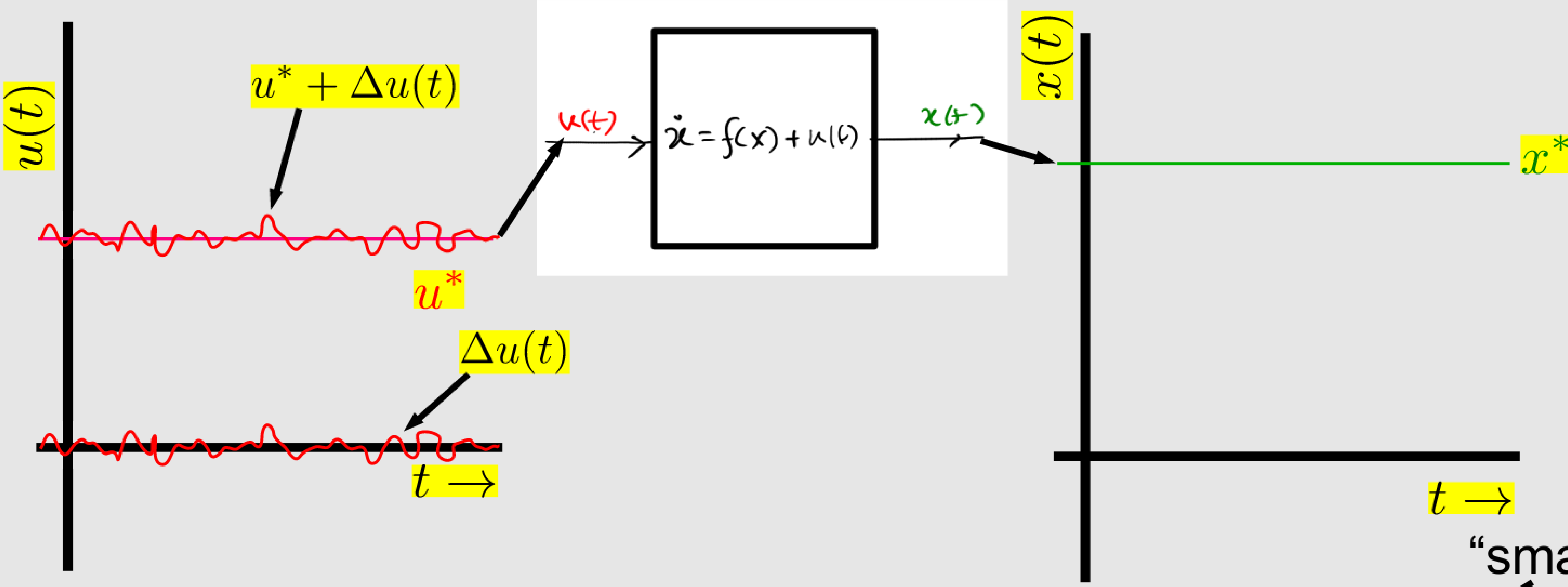
- DC operation (equilibrium), viewed in time



- Now, **perturb** the input <u>a little</u>: $u(t) = u^* + \Delta u(t)$

Linearization (contd. - 2)

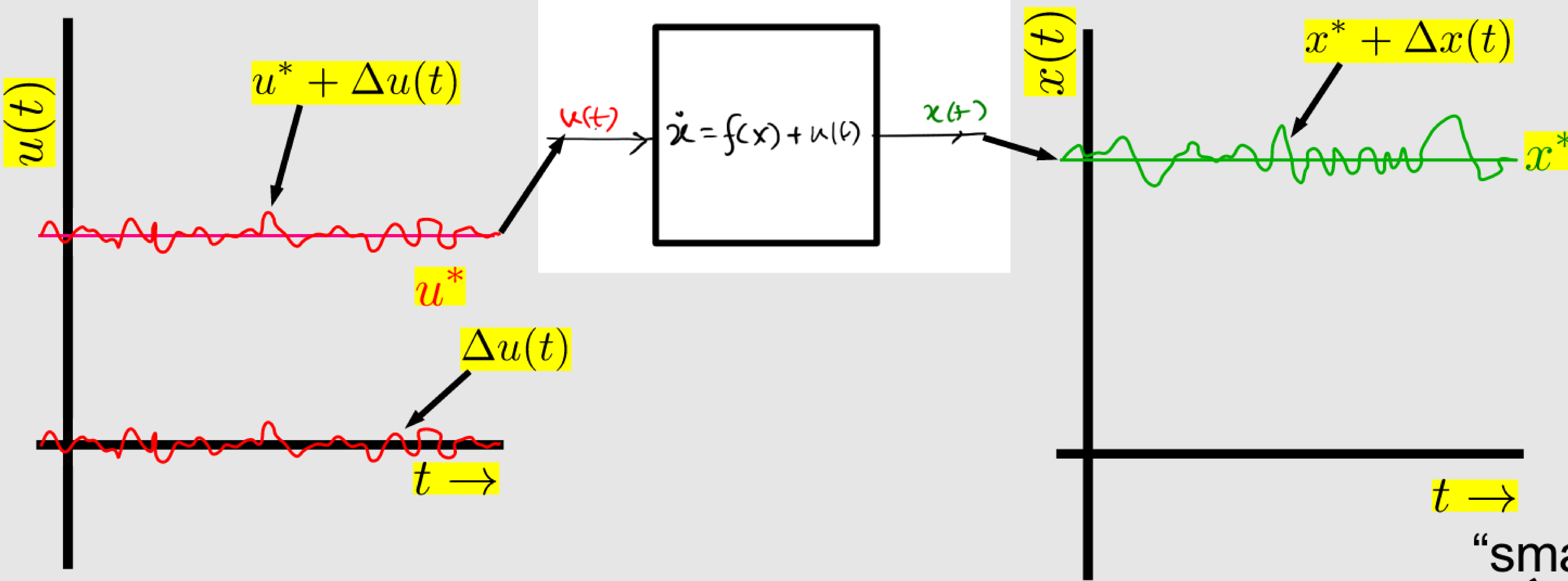
- DC operation (equilibrium), viewed in time



- Now, **perturb** the input a little: $u(t) = u^* + \Delta u(t)$ "small"

Linearization (contd. - 2)

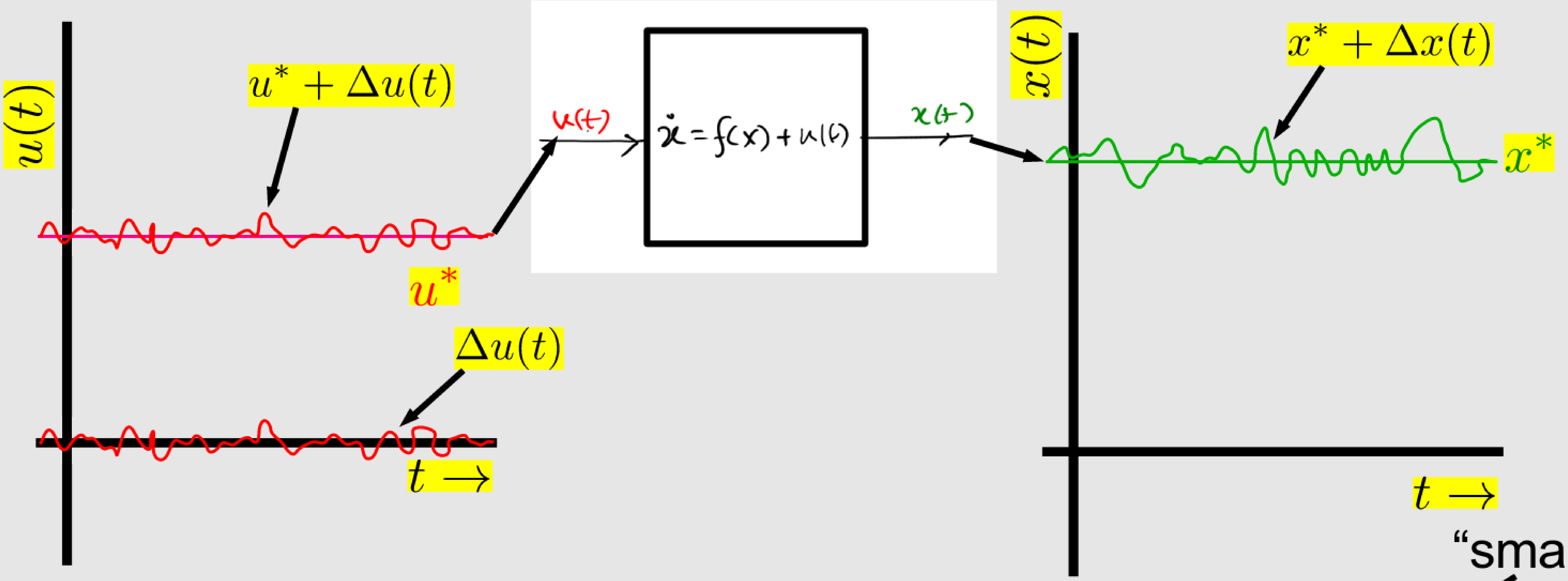
- DC operation (equilibrium), viewed in time



- Now, **perturb** the input <u>a little</u>: $u(t) = u^* + \Delta u(t)$ "small"

Linearization (contd. - 2)

- DC operation (equilibrium), viewed in time

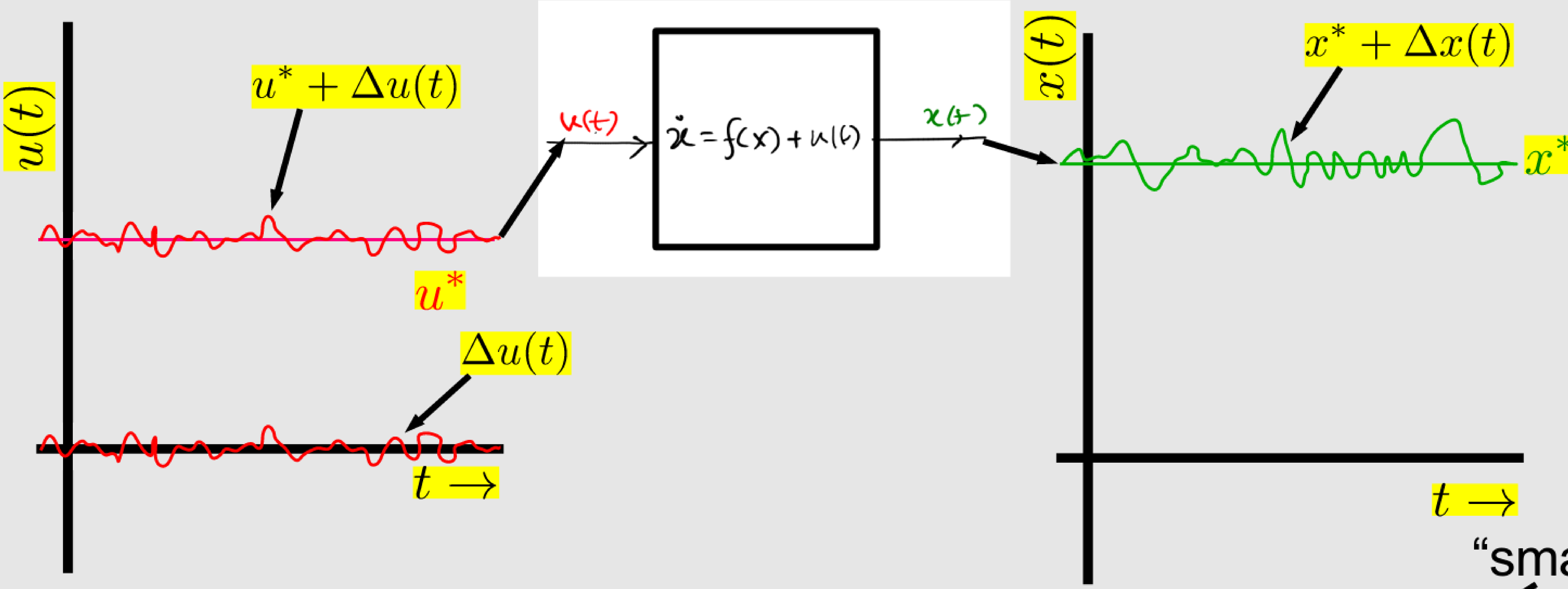


"small"

- Now, **perturb** the input <u>a little</u>: $u(t) = u^* + \Delta u(t)$
- <u>**Suppose**</u> x(t) responds by also changing <u>a little</u>
  - $x(t) = x^* + \Delta x(t)$

Linearization (contd. - 2)

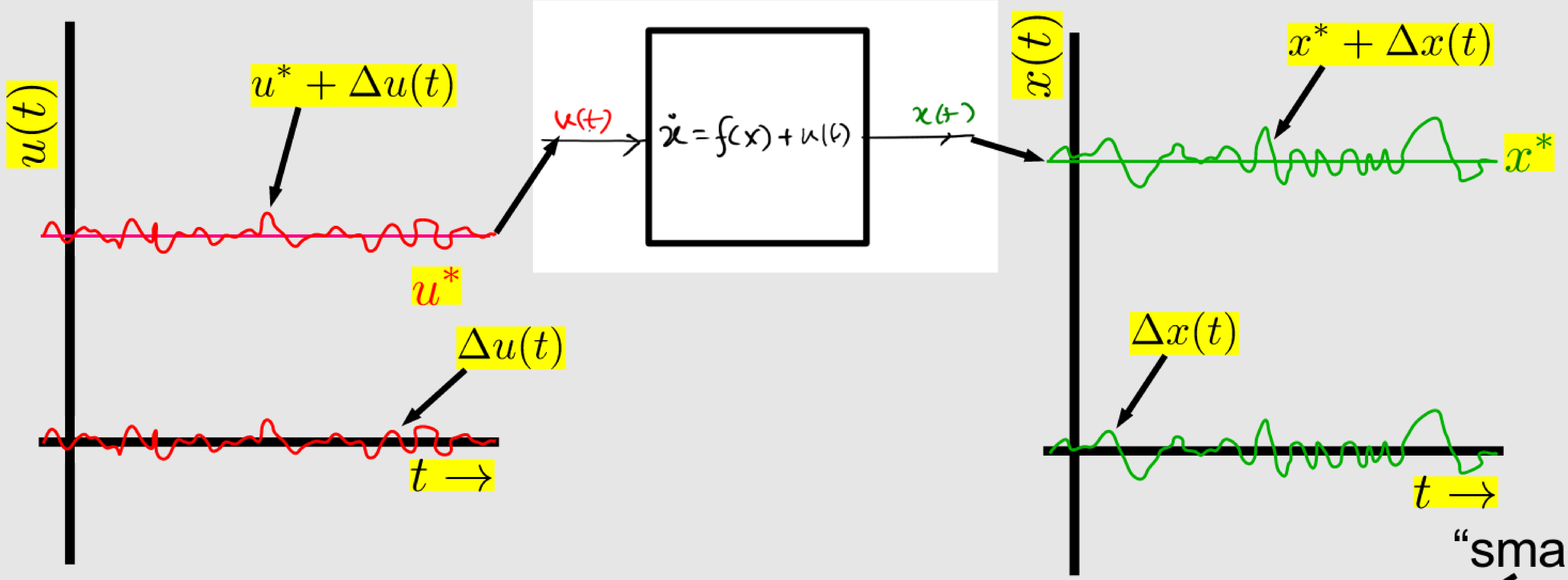
- DC operation (equilibrium), viewed in time



- Now, **perturb** the input a little: $u(t) = u^* + \Delta u(t)$ "small"
- Suppose x(t) responds by also changing a little
  - $x(t) = x^* + \Delta x(t)$ ← **ASSUMED** "small"

Linearization (contd. - 2)

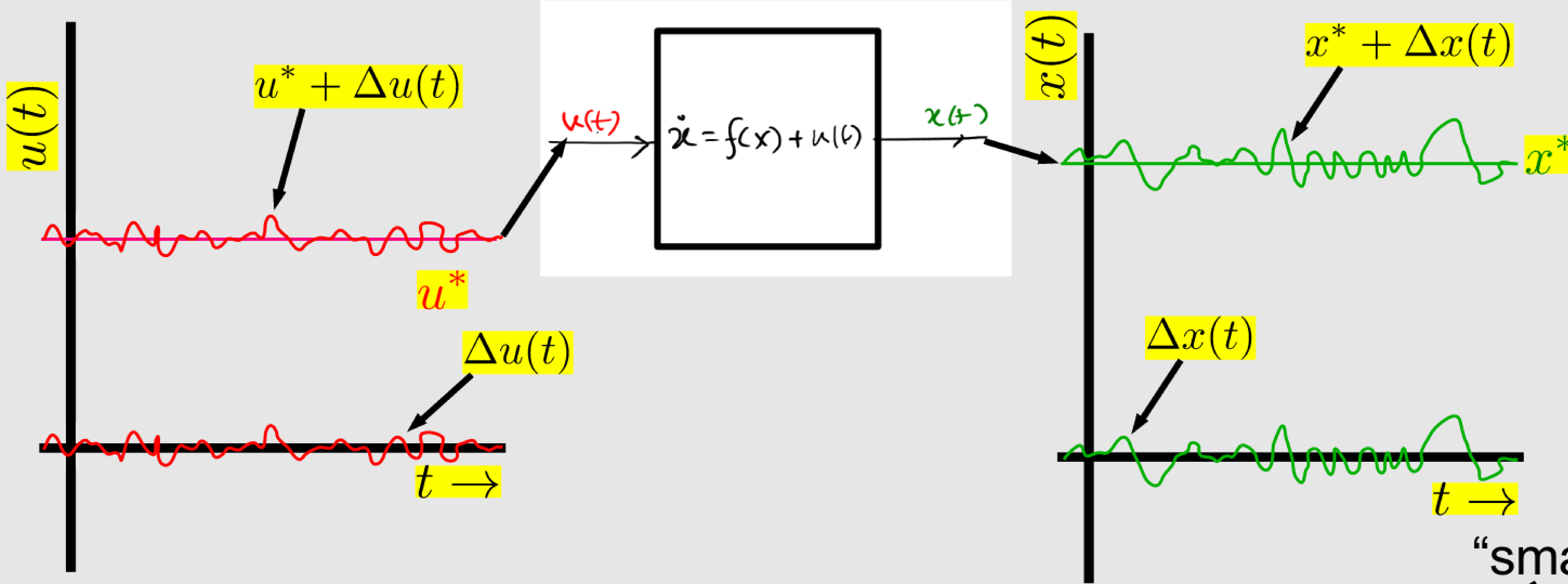
- DC operation (equilibrium), viewed in time



- Now, **perturb** the input <u>a little</u>: $u(t) = u^* + \Delta u(t)$ "small"
- <u>**Suppose**</u> x(t) responds by also changing <u>a little</u>
  - $x(t) = x^* + \Delta x(t)$ ← **ASSUMED** "small"

Linearization (contd. - 2)

- DC operation (equilibrium), viewed in time

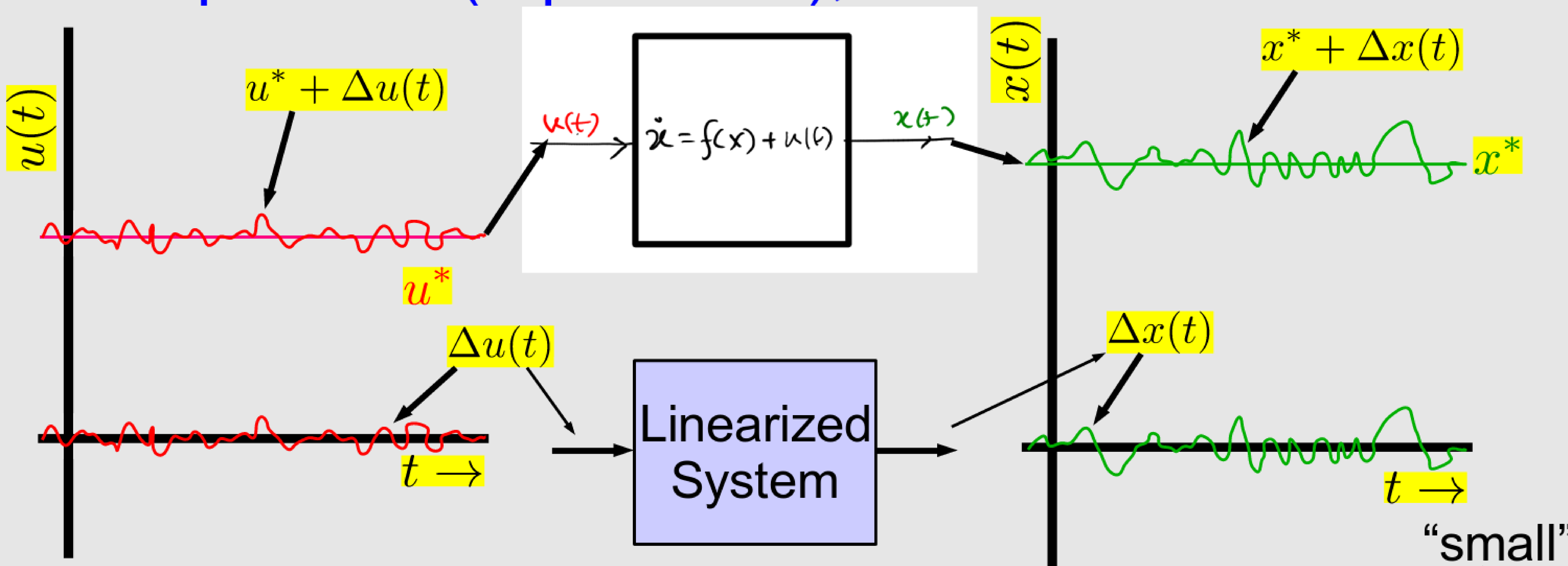


"small"

- Now, **perturb** the input a little: $u(t) = u^* + \Delta u(t)$
- **Suppose** x(t) responds by also changing a little
  - $x(t) = x^* + \Delta x(t)$ ← **ASSUMED** "small"
- Goal: find system relating Δu(t) and Δx(t) directly

Linearization (contd. - 2)

- DC operation (equilibrium), viewed in time



- Now, **perturb** the input a little: $u(t) = u^* + \Delta u(t)$
- **Suppose** x(t) responds by also changing a little
  - $x(t) = x^* + \Delta x(t)$ ← **ASSUMED** "small"
- Goal: find system relating Δu(t) and Δx(t) directly

Linearization (contd. - 3)

- Basic notion: replace f(x) by its tangent line at x*

Linearization (contd. - 3)

- Basic notion: replace f(x) by its tangent line at x*
- Mathematically: expand f(x*+Δx) in Taylor Series

$$f(x^* + \Delta x) \simeq f(x^*) + \left.\frac{df}{dx}\right|_{x^*} \Delta x + \frac{1}{2} \left.\frac{d^2 f}{dx^2}\right|_{x^*} \Delta x^2 + \cdots$$

Linearization (contd. - 3)

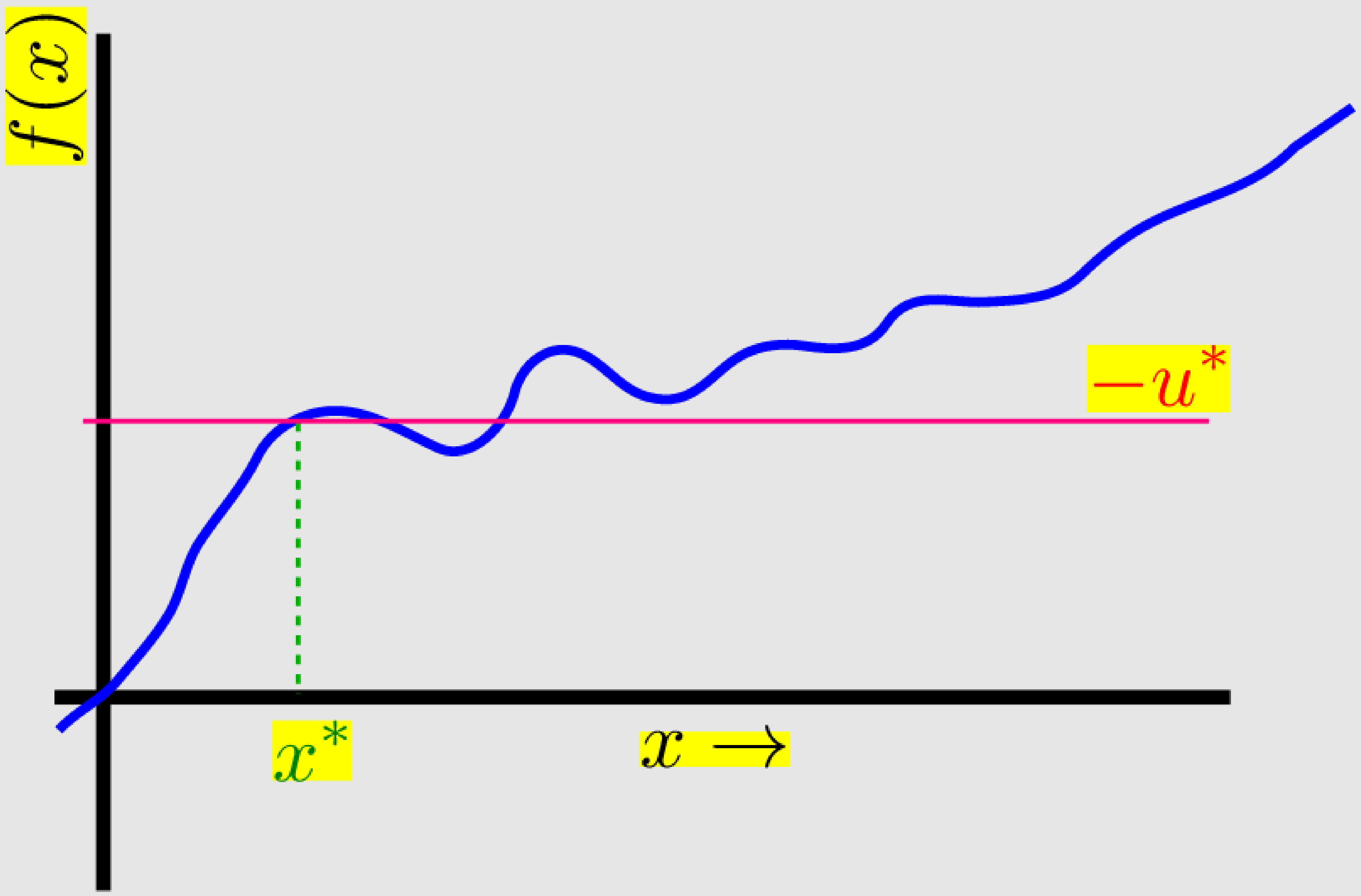
- Basic notion: replace f(x) by its tangent line at x*
- Mathematically: expand f(x*+Δx) in Taylor Series

$$\boxed{f(x^* + \Delta x) \simeq f(x^*) + \left.\frac{df}{dx}\right|_{x^*} \Delta x} + \frac{1}{2} \left.\frac{d^2 f}{dx^2}\right|_{x^*} \Delta x^2 + \cdots$$

Linearization (contd. - 3)

- Basic notion: replace f(x) by its tangent line at x*
- Mathematically: expand f(x*+Δx) in Taylor Series

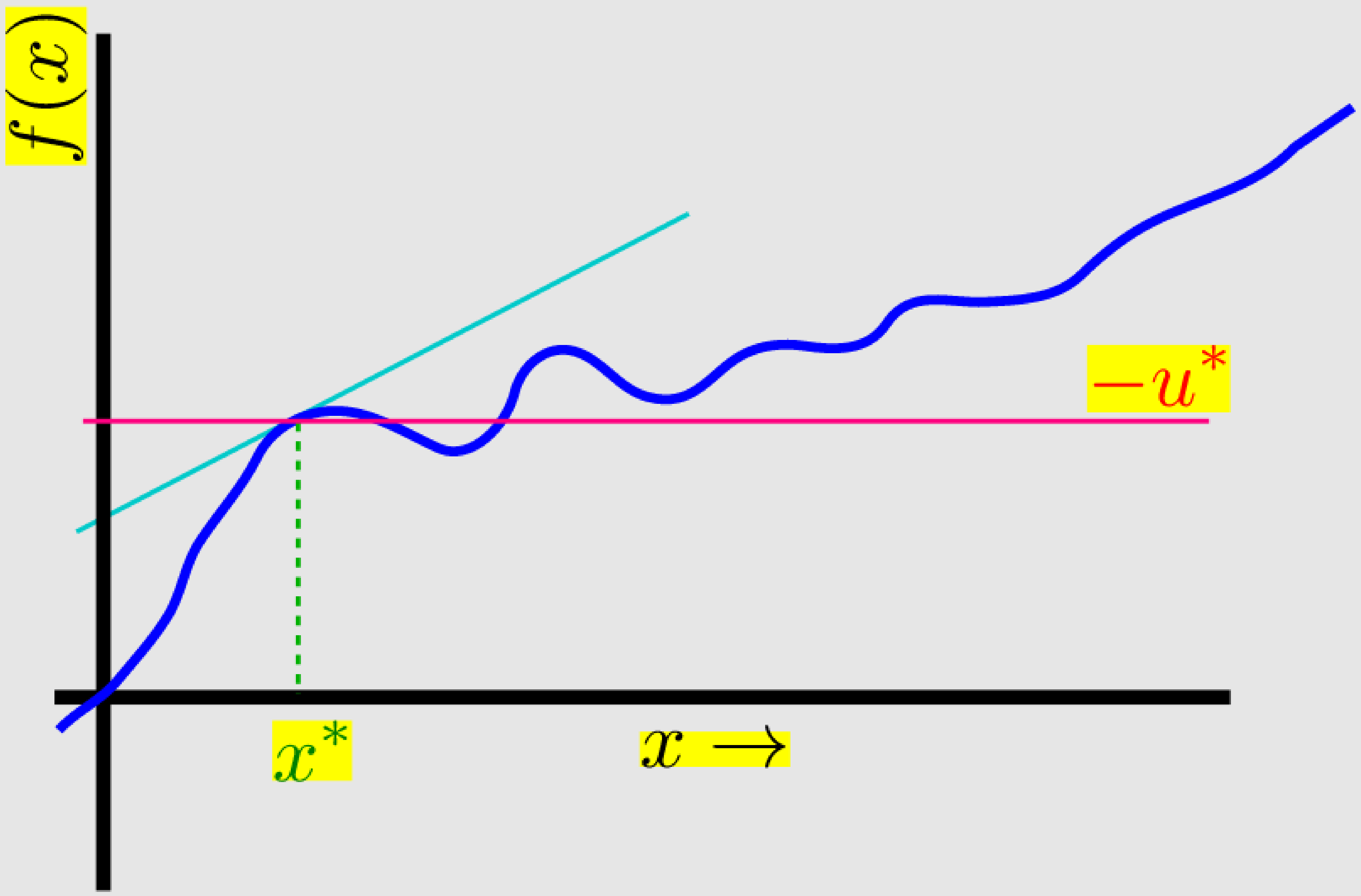
$$\boxed{f(x^* + \Delta x) \simeq f(x^*) + \left.\frac{df}{dx}\right|_{x^*} \Delta x} + \frac{1}{2}\left.\frac{d^2 f}{dx^2}\right|_{x^*} \Delta x^2 + \cdots$$



Linearization (contd. - 3)

- Basic notion: replace f(x) by its tangent line at x*
- Mathematically: expand f(x*+Δx) in Taylor Series

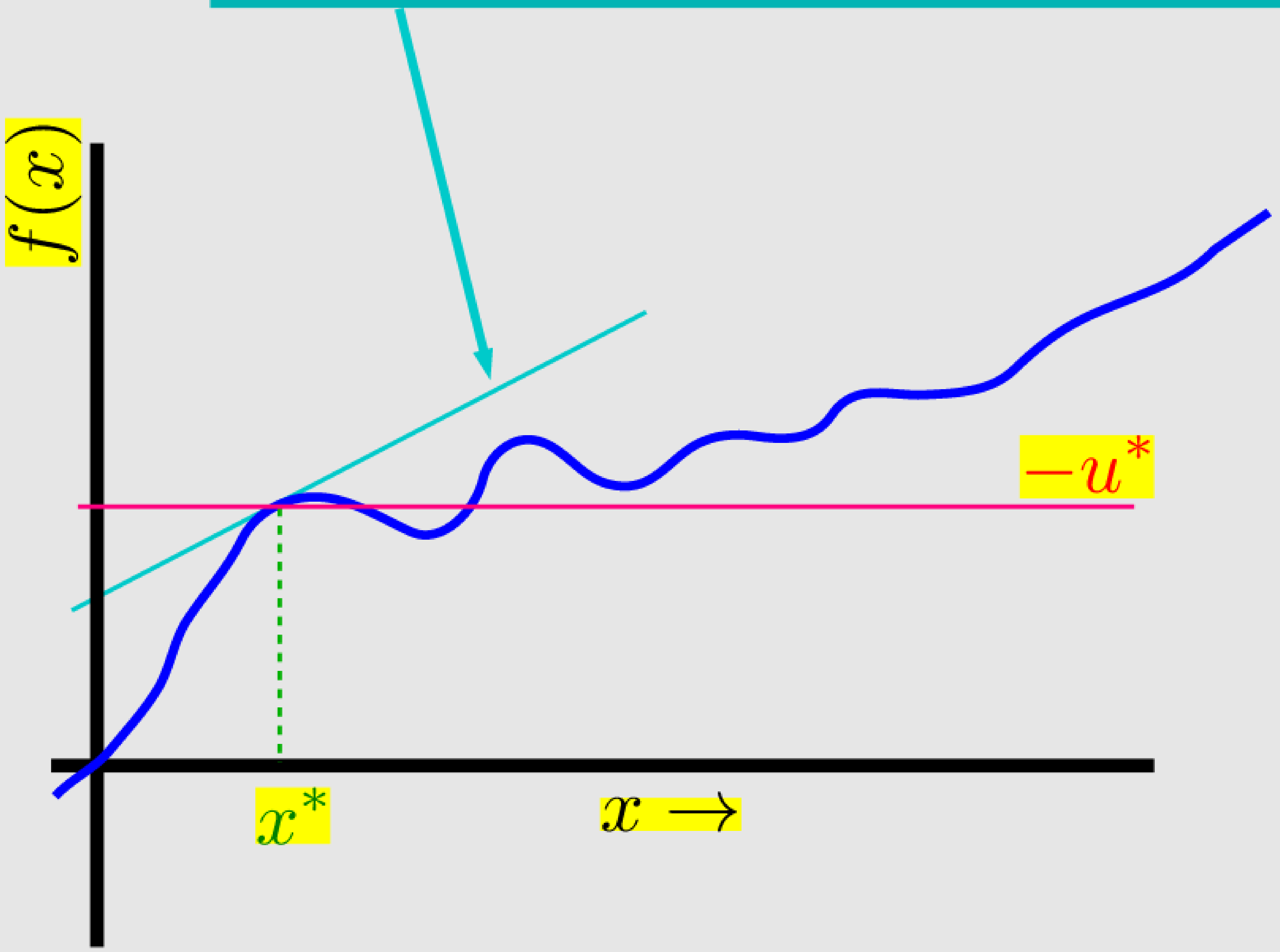
$$\boxed{f(x^* + \Delta x) \simeq f(x^*) + \left.\frac{df}{dx}\right|_{x^*} \Delta x} + \frac{1}{2} \left.\frac{d^2 f}{dx^2}\right|_{x^*} \Delta x^2 + \cdots$$



Linearization (contd. - 3)

- Basic notion: replace f(x) by its tangent line at x*
- Mathematically: expand f(x*+Δx) in Taylor Series

$$f(x^* + \Delta x) \simeq f(x^*) + \left.\frac{df}{dx}\right|_{x^*} \Delta x + \frac{1}{2} \left.\frac{d^2 f}{dx^2}\right|_{x^*} \Delta x^2 + \cdots$$

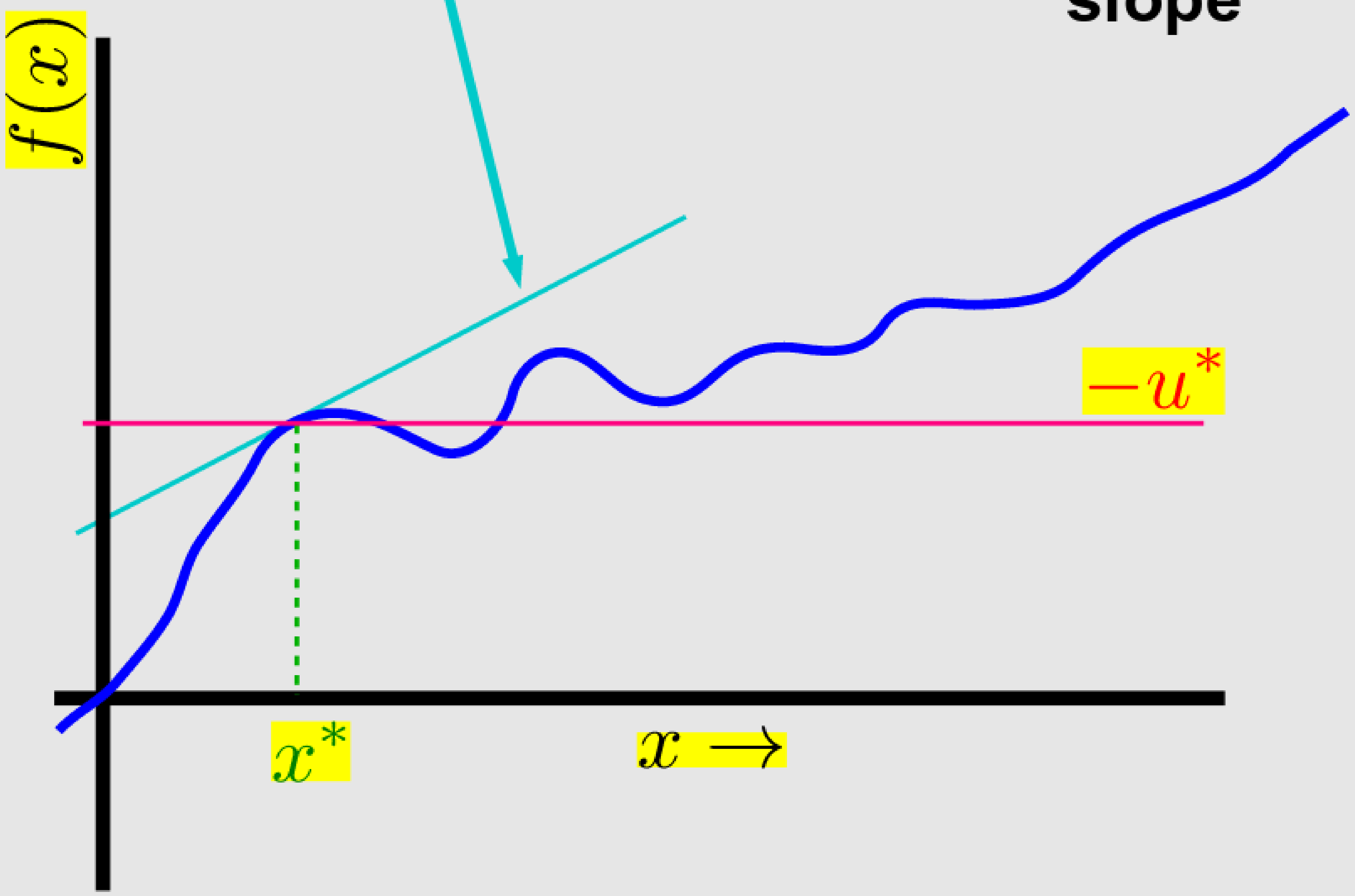


Linearization (contd. - 3)

- Basic notion: replace f(x) by its tangent line at x*
- Mathematically: expand f(x*+Δx) in Taylor Series

$$f(x^* + \Delta x) \simeq f(x^*) + \left.\frac{df}{dx}\right|_{x^*} \Delta x + \frac{1}{2}\left.\frac{d^2 f}{dx^2}\right|_{x^*} \Delta x^2 + \cdots$$

slope

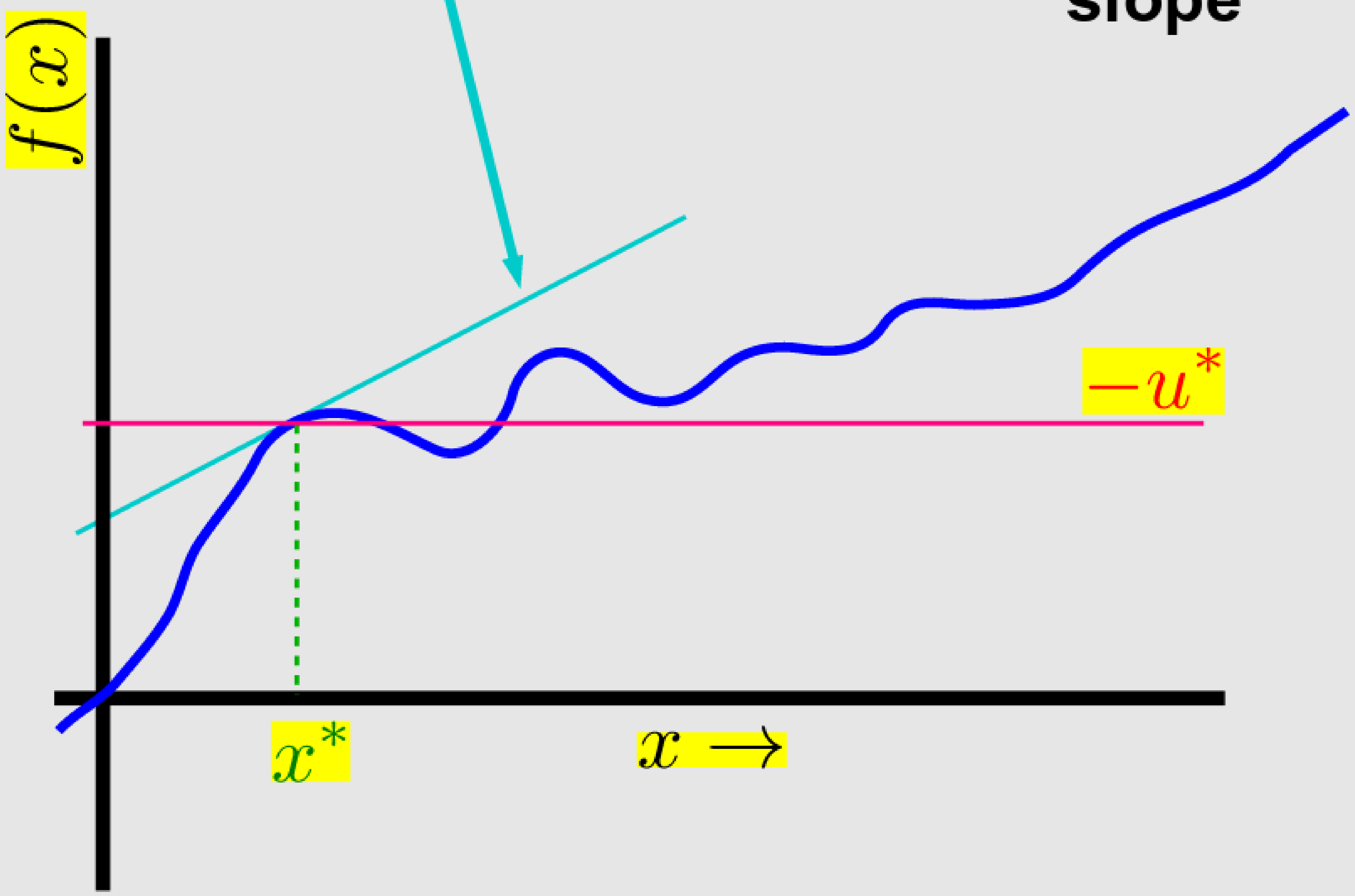


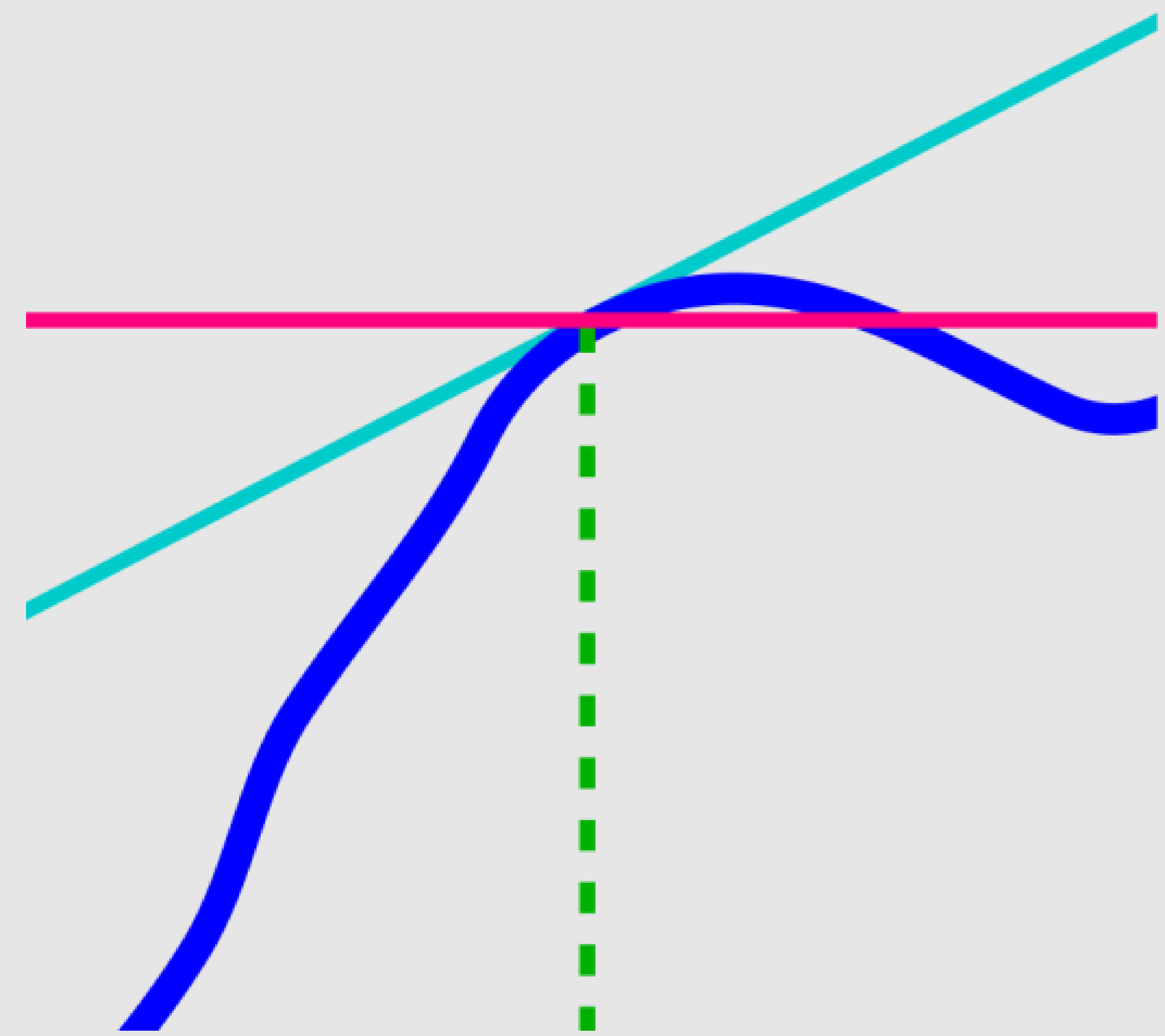
Linearization (contd. - 3)

- Basic notion: replace f(x) by its tangent line at x*
- Mathematically: expand f(x*+Δx) in Taylor Series

$$f(x^* + \Delta x) \simeq f(x^*) + \left.\frac{df}{dx}\right|_{x^*} \Delta x + \frac{1}{2}\, \left.\frac{d^2 f}{dx^2}\right|_{x^*} \Delta x^2 + \cdots$$

slope



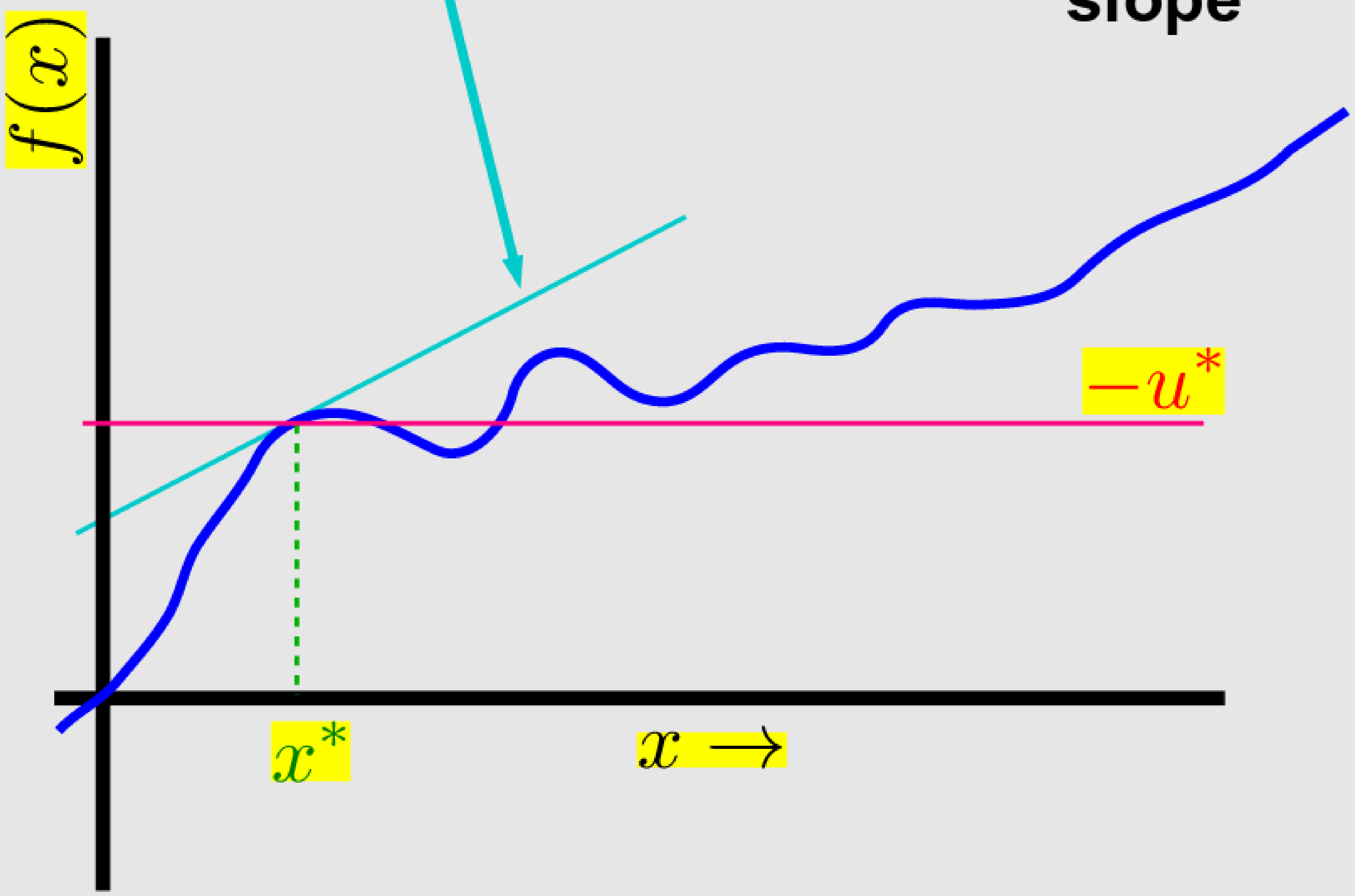


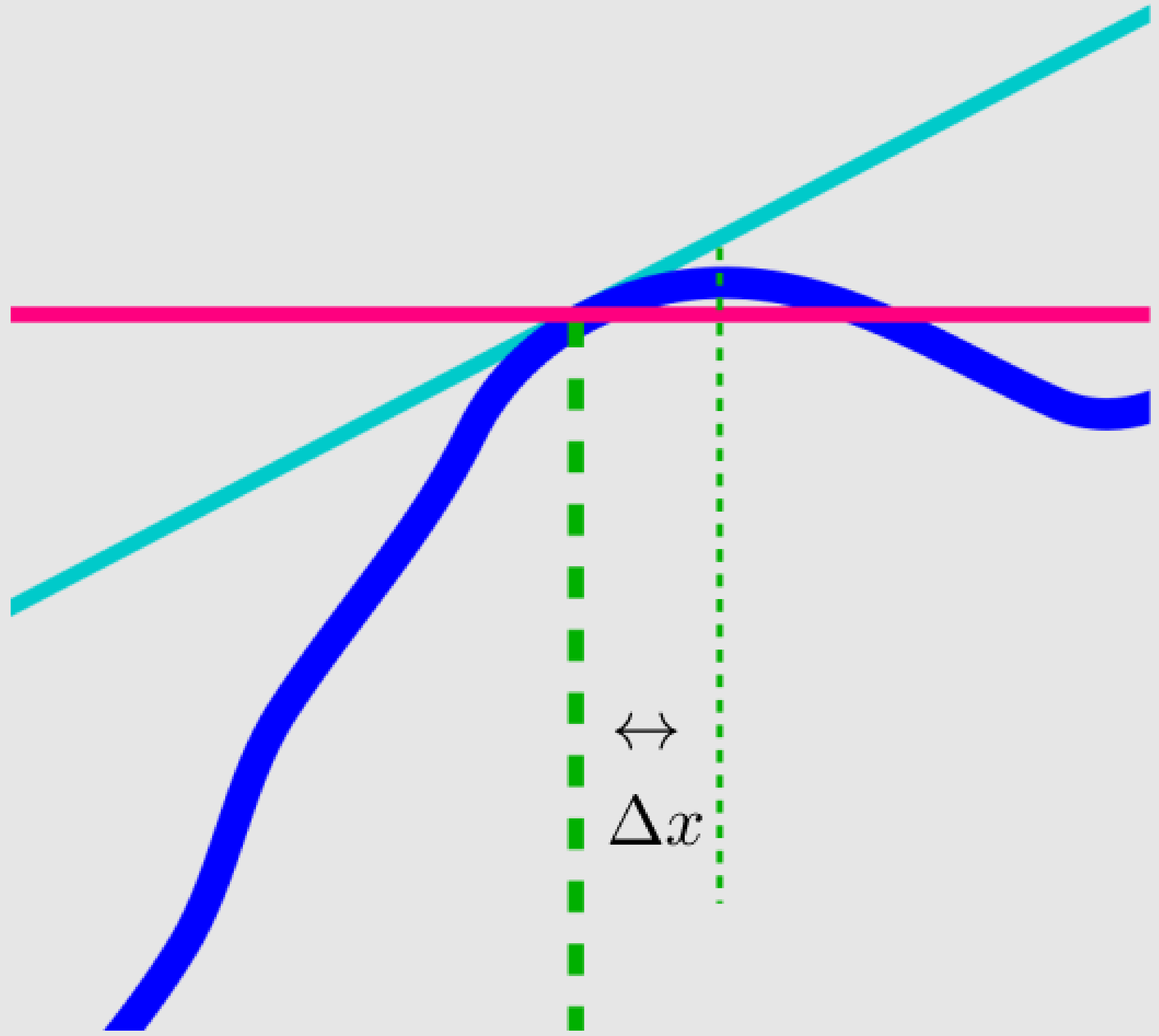
Linearization (contd. - 3)

- Basic notion: replace f(x) by its tangent line at x*
- Mathematically: expand f(x*+Δx) in Taylor Series

$$f(x^* + \Delta x) \simeq f(x^*) + \left.\frac{df}{dx}\right|_{x^*} \Delta x + \frac{1}{2} \left.\frac{d^2 f}{dx^2}\right|_{x^*} \Delta x^2 + \cdots$$

slope



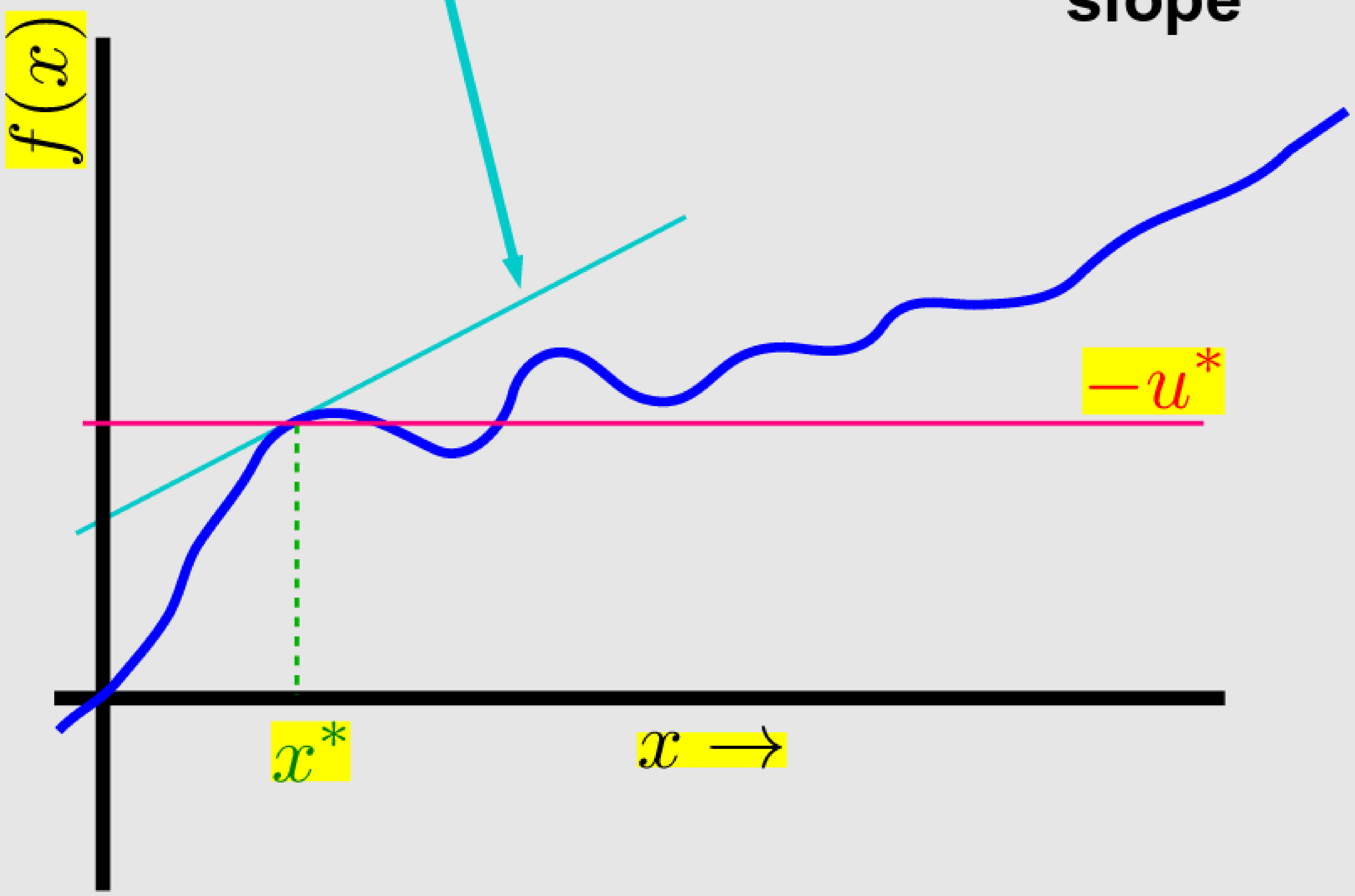


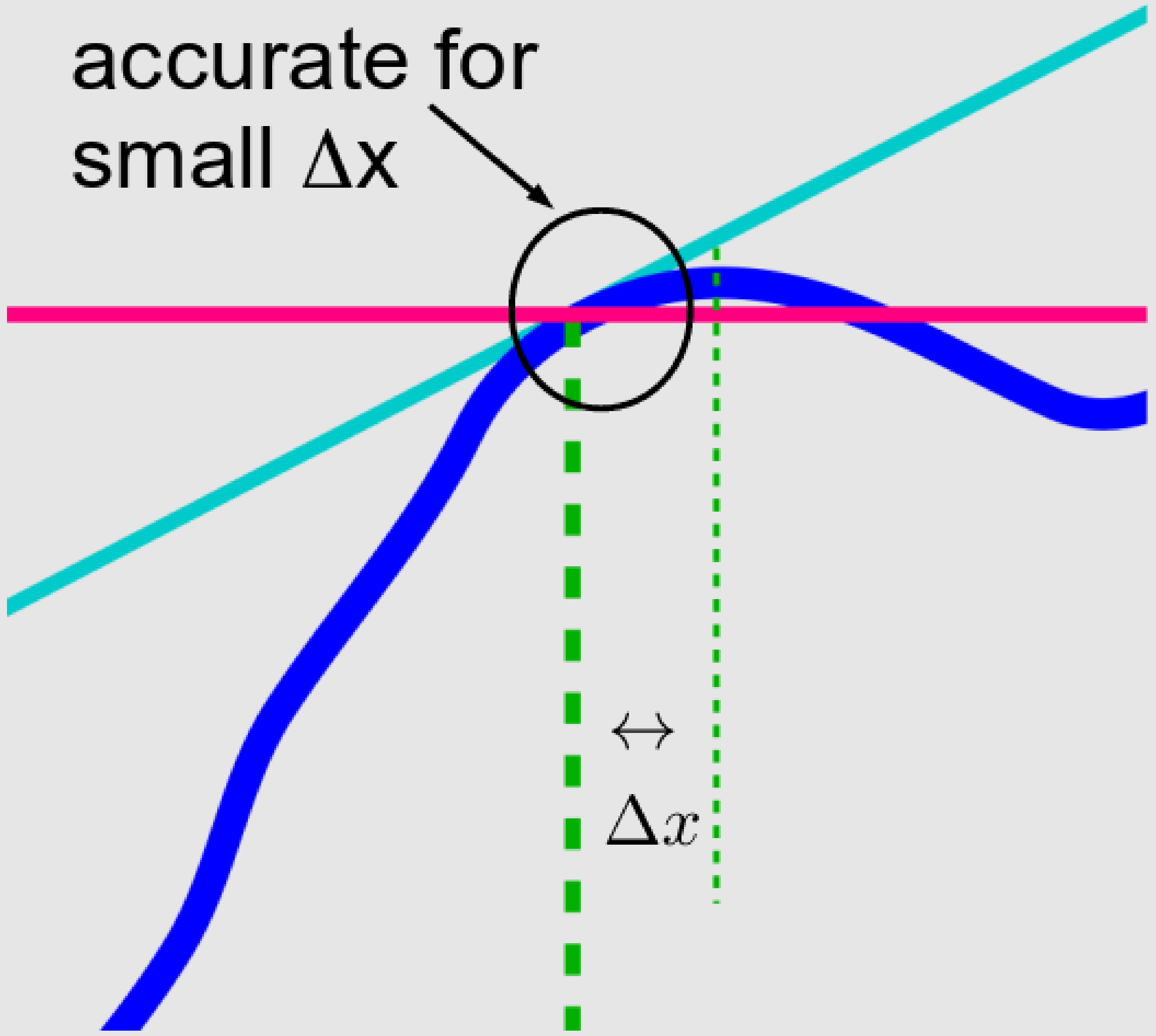
Linearization (contd. - 3)

- Basic notion: replace f(x) by its tangent line at x*
- Mathematically: expand f(x*+Δx) in Taylor Series

$$f(x^* + \Delta x) \simeq f(x^*) + \left.\frac{df}{dx}\right|_{x^*} \Delta x + \frac{1}{2} \left.\frac{d^2 f}{dx^2}\right|_{x^*} \Delta x^2 + \cdots$$

slope



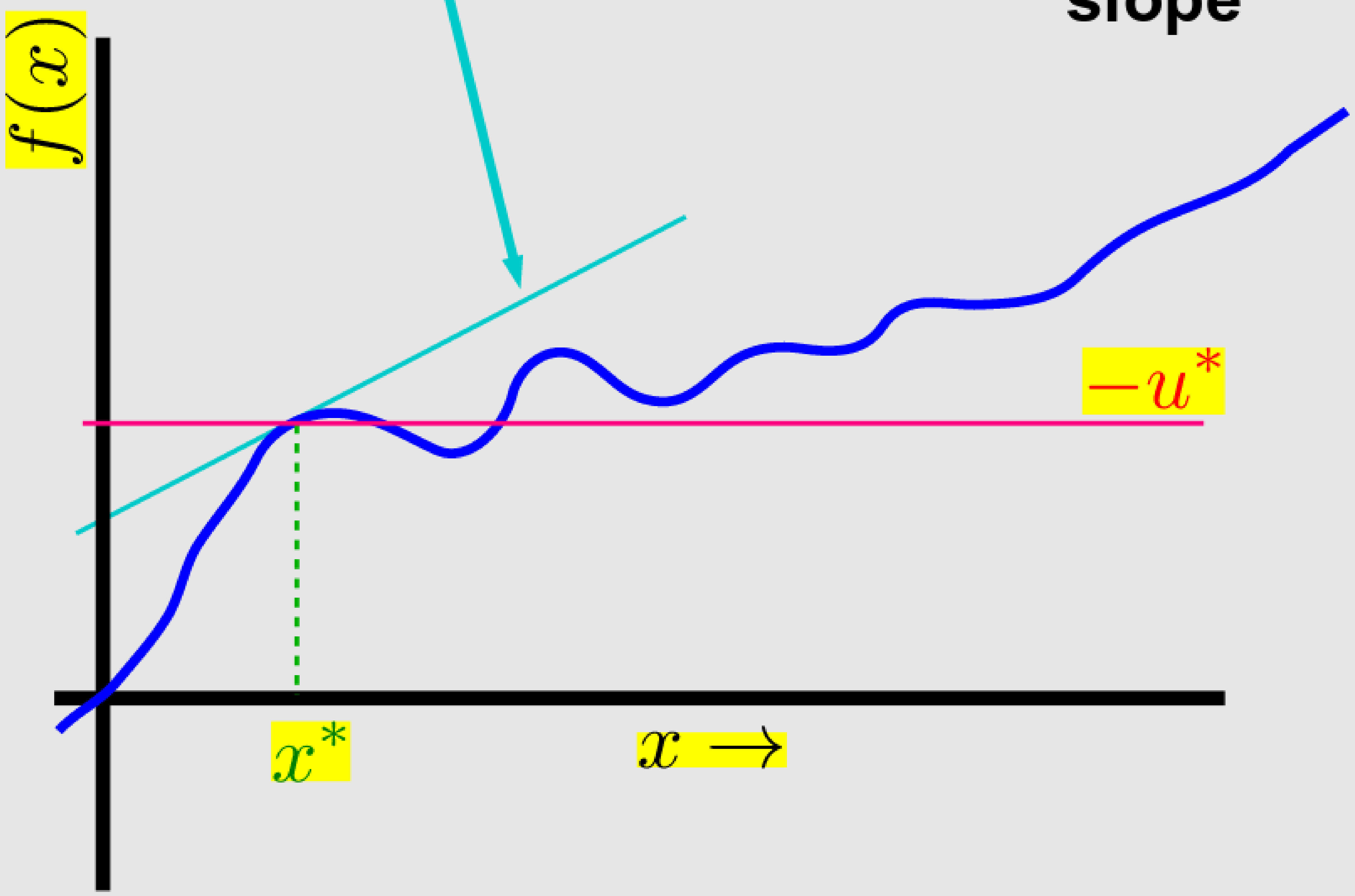


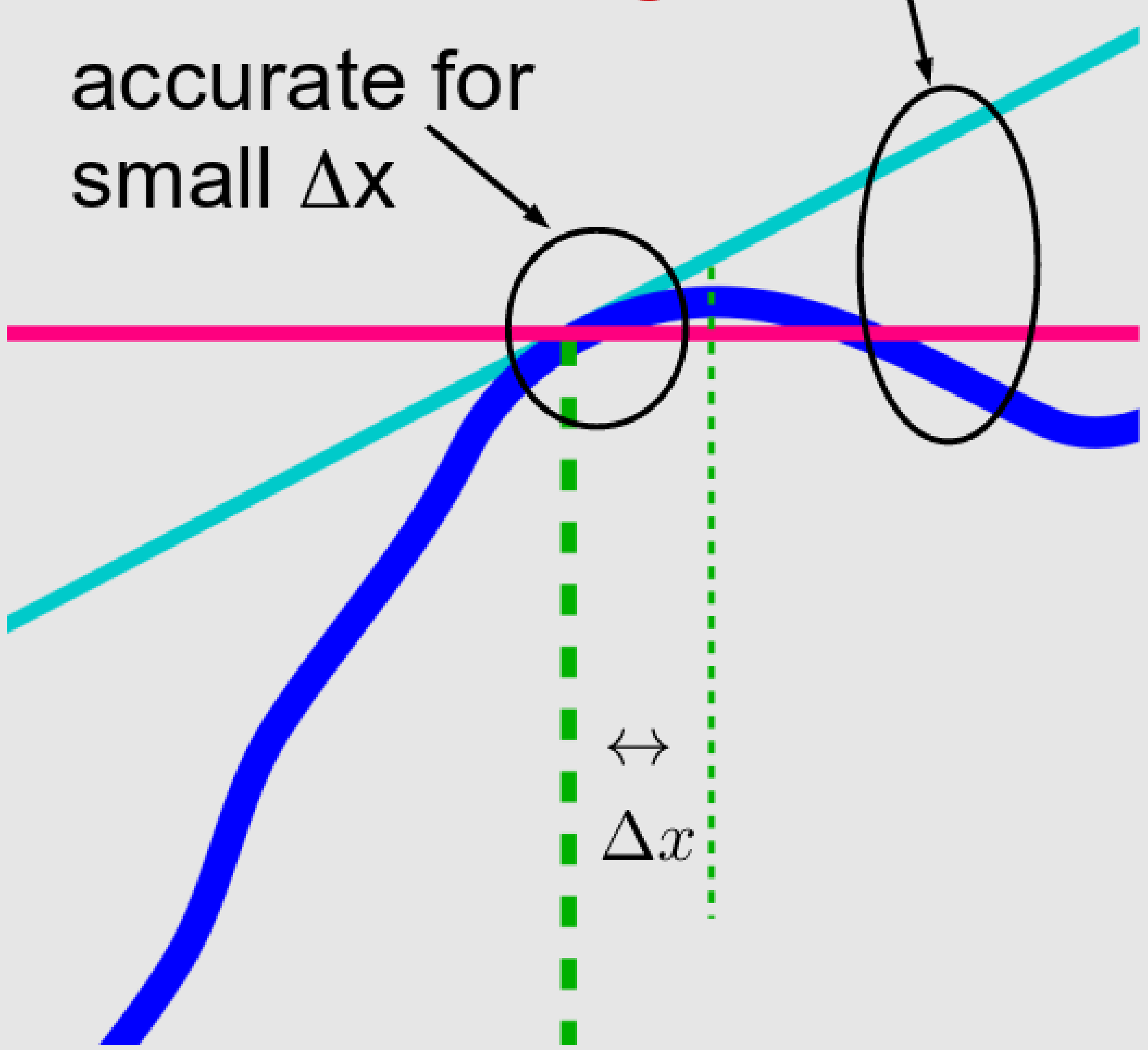
Linearization (contd. - 3)

- Basic notion: replace f(x) by its tangent line at x*
- Mathematically: expand f(x*+Δx) in Taylor Series

$$f(x^* + \Delta x) \simeq f(x^*) + \left.\frac{df}{dx}\right|_{x^*} \Delta x + \frac{1}{2} \left.\frac{d^2 f}{dx^2}\right|_{x^*} \Delta x^2 + \cdots$$

slope





Linearization (contd. - 4)

- applying the Taylor linearization
  - (move to xournal)

$f(x^*) + u^* = 0$

$\frac{dx}{dt} = f(x(t)) + u(t)$, with 1) $u = u^* + \Delta u(t)$ and 2) $x(t) = x^* + \Delta x(t)$

call this J

$\frac{d}{dt}(x^* + \Delta x(t)) \simeq \cancel{f(x^*)} + \left.\frac{df}{dx}\right|_{x^*} \Delta x(t) + \cancel{u^*} + \Delta u(t)$

Linearization (contd. - 4)

- applying the Taylor linearization

$\hookrightarrow f(x^*)+u^*=0$

$$\frac{dx}{dt} = f(x(t)) + u(t), \quad \text{with } 1)\ u = u^* + \Delta u(t) \text{ and } 2)\ x(t) = x^* + \Delta x(t)$$

call this J

$$\frac{d}{dt}(x^* + \Delta x(t)) \simeq \cancel{f(x^*)} + \left.\frac{df}{dx}\right|_{x^*} \Delta x(t) + \cancel{u^*} + \Delta u(t)$$

$$\boxed{\frac{d}{dt}\Delta x(t) \simeq J \Delta x(t) + \Delta u(t)} \longleftarrow \text{LINEARIZED SYSTEM}$$

Linearization (contd. - 4)

- applying the Taylor linearization

$$f(x^*) + u^* = 0$$

$$\frac{dx}{dt} = f(x(t)) + u(t), \quad \text{with 1) } u = u^* + \Delta u(t) \text{ and 2) } x(t) = x^* + \Delta x(t)$$

call this J

$$\frac{d}{dt}(x^* + \Delta x(t)) \simeq \cancel{f(x^*)} + \left.\frac{df}{dx}\right|_{x^*} \Delta x(t) + \cancel{u^*} + \Delta u(t)$$

$$J \triangleq \left.\frac{df}{dx}\right|_{x^*}$$

$$\boxed{\frac{d}{dt}\Delta x(t) \simeq J \Delta x(t) + \Delta u(t)}$$

LINEARIZED SYSTEM

Linearization (contd. - 4)

- applying the Taylor linearization

$\rightarrow f(x^*) + u^* = 0$

$\frac{dx}{dt} = f(x(t)) + u(t)$, with 1) $u = u^* + \Delta u(t)$ and 2) $x(t) = x^* + \Delta x(t)$

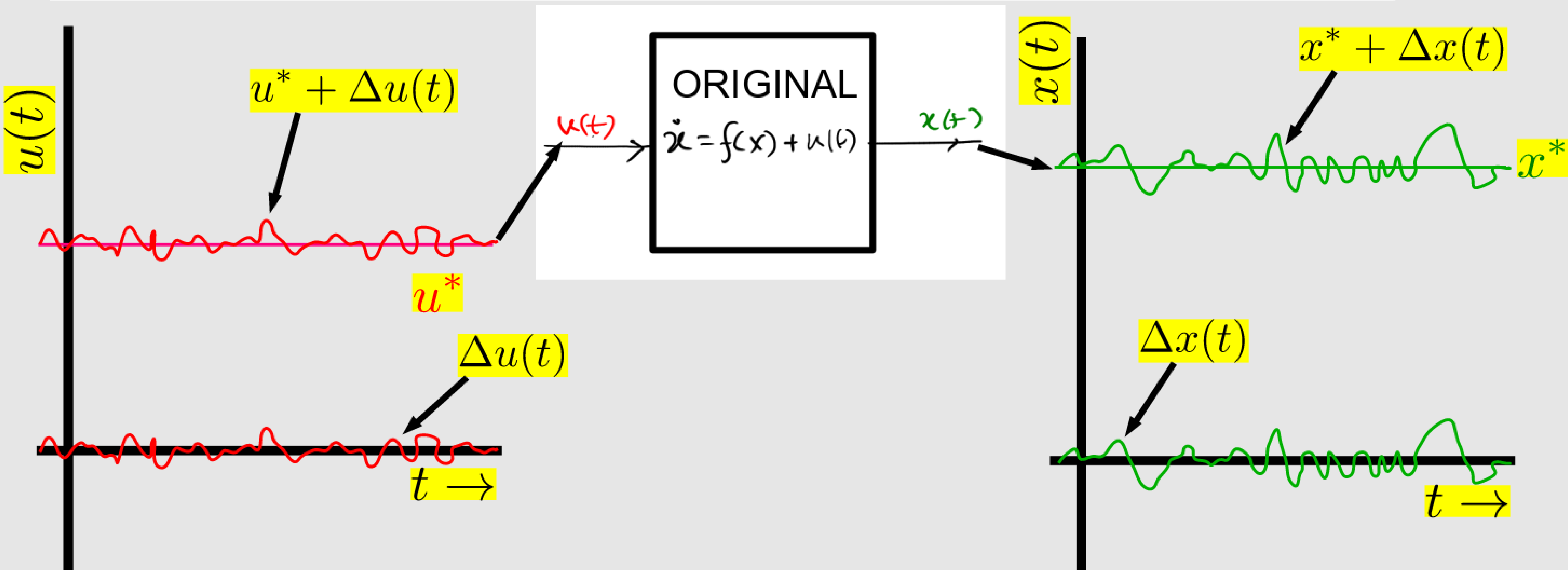
call this J

$\frac{d}{dt}(x^* + \Delta x(t)) \simeq \cancel{f(x^*)} + \frac{df}{dx}\Big|_{x^*} \Delta x(t) + \cancel{u^*} + \Delta u(t)$

$$J \triangleq \frac{df}{dx}\Big|_{x^*}$$

$$\boxed{\frac{d}{dt}\Delta x(t) \simeq J\Delta x(t) + \Delta u(t)}$$

LINEARIZED SYSTEM



Linearization (contd. - 4)

- applying the Taylor linearization

$$\frac{dx}{dt} = f(x(t)) + u(t), \quad \text{with 1) } u = u^* + \Delta u(t) \text{ and 2) } x(t) = x^* + \Delta x(t)$$

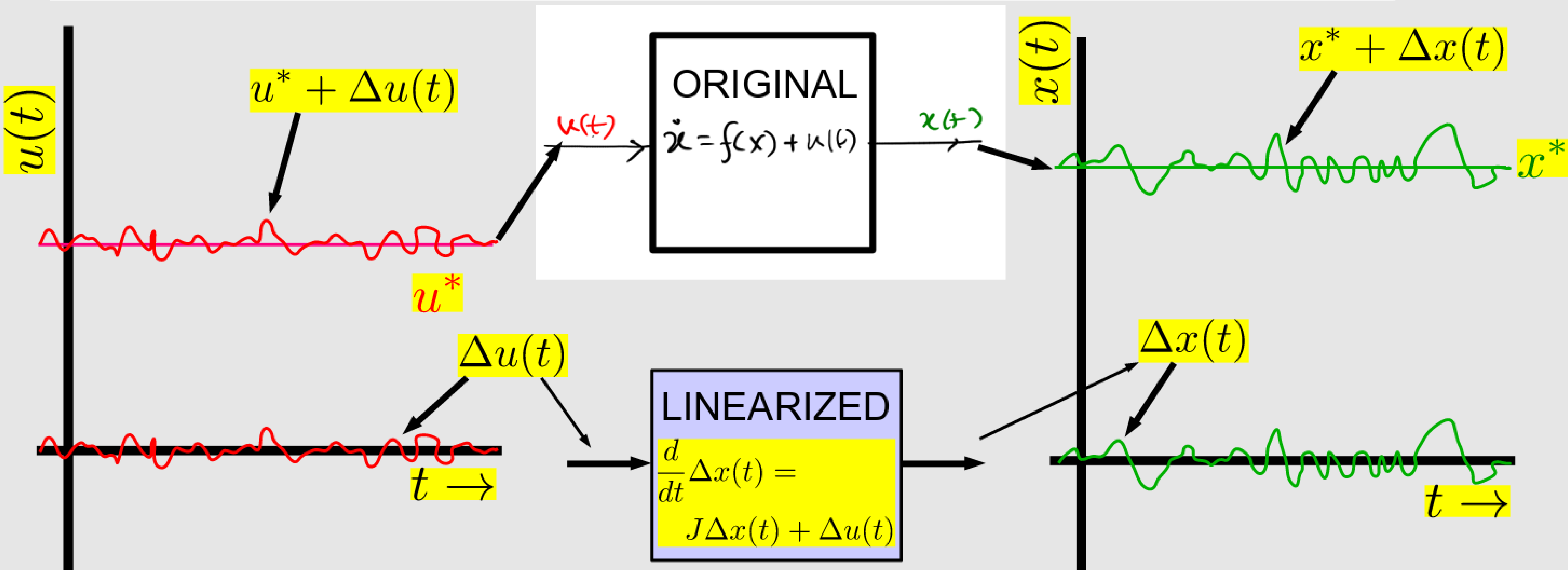
$\rightarrow f(x^*) + u^* = 0$

call this J

$$\frac{d}{dt}(x^* + \Delta x(t)) \simeq \cancel{f(x^*)} + \frac{df}{dx}\Big|_{x^*} \Delta x(t) + \cancel{u^*} + \Delta u(t)$$

$$J \triangleq \frac{df}{dx}\Big|_{x^*}$$

$$\boxed{\frac{d}{dt}\Delta x(t) \simeq J\Delta x(t) + \Delta u(t)}$$ ← LINEARIZED SYSTEM



Linearization of Vector S.S. Systems

- Now: the full S.S.R: $\frac{d}{dt}\vec{x}(t) = \vec{f}(\vec{x}(t), \vec{u}(t))$

Linearization of Vector S.S. Systems

- Now: the full S.S.R: $\frac{d}{dt}\vec{x}(t) = \vec{f}(\vec{x}(t), \vec{u}(t))$
- step 1: **find a DC. op. pt. (equilibrium pt.)**
  - $\vec{0} = \vec{f}(\vec{x}^*, \vec{u}^*)$

Linearization of Vector S.S. Systems

- Now: the full S.S.R: $\frac{d}{dt}\vec{x}(t) = \vec{f}(\vec{x}(t), \vec{u}(t))$
- step 1: **find a DC. op. pt. (equilibrium pt.)**
  - $\vec{0} = \vec{f}(\vec{x}^*, \vec{u}^*)$ ← DC input

Linearization of Vector S.S. Systems

- Now: the full S.S.R: $\frac{d}{dt}\vec{x}(t) = \vec{f}(\vec{x}(t), \vec{u}(t))$
- step 1: **find a DC. op. pt. (equilibrium pt.)**
  - $\vec{0} = \vec{f}(\vec{x}^*, \vec{u}^*)$ ← DC input

    ↑ DC solution

Linearization of Vector S.S. Systems

- Now: the full S.S.R: $\frac{d}{dt}\vec{x}(t) = \vec{f}(\vec{x}(t), \vec{u}(t))$
- step 1: **find a DC. op. pt. (equilibrium pt.)**
  - $\vec{0} = \vec{f}(\vec{x}^*, \vec{u}^*)$ ← DC input

DC solution

Solving for this is often difficult, even using computational methods

Linearization of Vector S.S. Systems

- Now: the full S.S.R: $\frac{d}{dt}\vec{x}(t) = \vec{f}(\vec{x}(t), \vec{u}(t))$
- step 1: **find a DC. op. pt. (equilibrium pt.)**
  - $\vec{0} = \vec{f}(\vec{x}^*, \vec{u}^*)$ ← DC input

    DC solution

    Solving for this is often difficult, even using computational methods
- The linearized system is (see handwritten notes for derivation)

$$\frac{d}{dt}\Delta\vec{x}(t) = \boldsymbol{J}_x(\vec{x}^*, \vec{u}^*)\Delta\vec{x}(t) + \boldsymbol{J}_u(\vec{x}^*, \vec{u}^*)\Delta\vec{u}(t))$$

Linearization of Vector S.S. Systems

- Now: the full S.S.R: $\frac{d}{dt}\vec{x}(t) = \vec{f}(\vec{x}(t), \vec{u}(t))$
- step 1: **find a DC. op. pt. (equilibrium pt.)**
  - $\vec{0} = \vec{f}(\vec{x}^*, \vec{u}^*)$ ← DC input

    DC solution

    Solving for this is often difficult, even using computational methods
- The linearized system is (see handwritten notes for derivation)

$$\frac{d}{dt}\Delta\vec{x}(t) = \boldsymbol{J}_x(\vec{x}^*, \vec{u}^*)\Delta\vec{x}(t) + \boldsymbol{J}_u(\vec{x}^*, \vec{u}^*)\Delta\vec{u}(t))$$

n-vector

Linearization of Vector S.S. Systems

- Now: the full S.S.R: $\frac{d}{dt}\vec{x}(t) = \vec{f}(\vec{x}(t), \vec{u}(t)))$
- step 1: **find a DC. op. pt. (equilibrium pt.)**
  - $\vec{0} = \vec{f}(\vec{x}^*, \vec{u}^*)$ DC input

    DC solution

    Solving for this is often difficult, even using computational methods
- The linearized system is (see handwritten notes for derivation)

$$\frac{d}{dt}\Delta\vec{x}(t) = \boldsymbol{J}_x(\vec{x}^*, \vec{u}^*)\Delta\vec{x}(t) + \boldsymbol{J}_u(\vec{x}^*, \vec{u}^*)\Delta\vec{u}(t))$$

n-vector **nxn matrix**

Linearization of Vector S.S. Systems

- Now: the full S.S.R: $\frac{d}{dt}\vec{x}(t) = \vec{f}(\vec{x}(t), \vec{u}(t))$
- step 1: **find a DC. op. pt. (equilibrium pt.)**
  - $\vec{0} = \vec{f}(\vec{x}^*, \vec{u}^*)$ DC input

    DC solution

    Solving for this is often difficult, even using computational methods
- The linearized system is (see handwritten notes for derivation)

$$\frac{d}{dt}\Delta\vec{x}(t) = \boldsymbol{J}_x(\vec{x}^*, \vec{u}^*)\Delta\vec{x}(t) + \boldsymbol{J}_u(\vec{x}^*, \vec{u}^*)\Delta\vec{u}(t))$$

n-vector **nxn matrix** m-vector

Linearization of Vector S.S. Systems

- Now: the full S.S.R: $\frac{d}{dt}\vec{x}(t) = \vec{f}(\vec{x}(t), \vec{u}(t))$
- step 1: **find a DC. op. pt. (equilibrium pt.)**
  - $\vec{0} = \vec{f}(\vec{x}^*, \vec{u}^*)$ ← DC input

    DC solution

    Solving for this is often difficult, even using computational methods
- The linearized system is (see handwritten notes for derivation)

$$\frac{d}{dt}\Delta\vec{x}(t) = \boldsymbol{J}_x(\vec{x}^*, \vec{u}^*)\Delta\vec{x}(t) + \boldsymbol{J}_u(\vec{x}^*, \vec{u}^*)\Delta\vec{u}(t))$$

n-vector **nxn matrix** **nxm matrix** m-vector

Linearization of Vector S.S. Systems

- Now: the full S.S.R: $\frac{d}{dt}\vec{x}(t) = \vec{f}(\vec{x}(t), \vec{u}(t))$
- step 1: **find a DC. op. pt. (equilibrium pt.)**
  - $\vec{0} = \vec{f}(\vec{x}^*, \vec{u}^*)$ ← DC input

    DC solution

    Solving for this is often difficult, even using computational methods
- The linearized system is (see handwritten notes for derivation)

$$\frac{d}{dt}\Delta\vec{x}(t) = \boldsymbol{J}_x(\vec{x}^*, \vec{u}^*)\Delta\vec{x}(t) + \boldsymbol{J}_u(\vec{x}^*, \vec{u}^*)\Delta\vec{u}(t))$$

n-vector **nxn matrix** **nxm matrix** m-vector

- What are $\boldsymbol{J}_x$ and $\boldsymbol{J}_u$?

Linearization of Vector S.S. Systems

- Now: the full S.S.R: $\frac{d}{dt}\vec{x}(t) = \vec{f}(\vec{x}(t), \vec{u}(t))$
- step 1: **find a DC. op. pt. (equilibrium pt.)**
  - $\vec{0} = \vec{f}(\vec{x}^*, \vec{u}^*)$ ← DC input

    DC solution

    Solving for this is often difficult, even using computational methods
- The linearized system is (see handwritten notes for derivation)

$$\frac{d}{dt}\Delta\vec{x}(t) = \boldsymbol{J}_x(\vec{x}^*, \vec{u}^*)\Delta\vec{x}(t) + \boldsymbol{J}_u(\vec{x}^*, \vec{u}^*)\Delta\vec{u}(t))$$

n-vector **nxn matrix** **nxm matrix** m-vector

- What are $\boldsymbol{J}_x$ and $\boldsymbol{J}_u$?
  - called **Jacobian or gradient matrices**

Jacobian (Gradient) Matrices

- If: $\vec{x}(t) = \begin{bmatrix} x_1 \\ \vdots \\ \vdots \\ x_n \end{bmatrix}, \quad \vec{u}(t) = \begin{bmatrix} u_1 \\ \vdots \\ u_m \end{bmatrix}, \quad \vec{f}(\vec{x},\vec{u}) = \begin{bmatrix} f_1(x_1,\cdots,x_n;u_1,\cdots,u_m) \\ \vdots \\ f_n(x_1,\cdots,x_n;u_1,\cdots,u_m) \end{bmatrix}$, then

Jacobian (Gradient) Matrices

- If: $\vec{x}(t) = \begin{bmatrix} x_1 \\ \vdots \\ \vdots \\ x_n \end{bmatrix}, \quad \vec{u}(t) = \begin{bmatrix} u_1 \\ \vdots \\ u_m \end{bmatrix}, \quad \vec{f}(\vec{x}, \vec{u}) = \begin{bmatrix} f_1(x_1, \cdots, x_n; u_1, \cdots, u_m) \\ \vdots \\ f_n(x_1, \cdots, x_n; u_1, \cdots, u_m) \end{bmatrix}$, then

$$\boldsymbol{J}_x(\vec{x}, \vec{u}) = \nabla_x \vec{f}(\vec{x}, \vec{u}) = \left.\frac{\partial \vec{f}}{\partial \vec{x}}\right|_{\vec{x},\vec{u}} \triangleq \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_{n-1}} & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_{n-1}} & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ \frac{\partial f_{n-1}}{\partial x_1} & \frac{\partial f_{n-1}}{\partial x_2} & \cdots & \frac{\partial f_{n-1}}{\partial x_{n-1}} & \frac{\partial f_{n-1}}{\partial x_n} \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_{n-1}} & \frac{\partial f_n}{\partial x_n} \end{bmatrix}$$

Jacobian (Gradient) Matrices

- If: $\vec{x}(t) = \begin{bmatrix} x_1 \\ \vdots \\ \vdots \\ x_n \end{bmatrix}, \quad \vec{u}(t) = \begin{bmatrix} u_1 \\ \vdots \\ u_m \end{bmatrix}, \quad \vec{f}(\vec{x},\vec{u}) = \begin{bmatrix} f_1(x_1,\cdots,x_n;u_1,\cdots,u_m) \\ \vdots \\ f_n(x_1,\cdots,x_n;u_1,\cdots,u_m) \end{bmatrix}$, then

nxn matrix

$$\boldsymbol{J}_x(\vec{x},\vec{u}) = \nabla_x \vec{f}(\vec{x},\vec{u}) = \left.\frac{\partial \vec{f}}{\partial \vec{x}}\right|_{\vec{x},\vec{u}} \triangleq \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_{n-1}} & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_{n-1}} & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ \frac{\partial f_{n-1}}{\partial x_1} & \frac{\partial f_{n-1}}{\partial x_2} & \cdots & \frac{\partial f_{n-1}}{\partial x_{n-1}} & \frac{\partial f_{n-1}}{\partial x_n} \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_{n-1}} & \frac{\partial f_n}{\partial x_n} \end{bmatrix}$$

Jacobian (Gradient) Matrices

- If: $\vec{x}(t) = \begin{bmatrix} x_1 \\ \vdots \\ \vdots \\ x_n \end{bmatrix}, \quad \vec{u}(t) = \begin{bmatrix} u_1 \\ \vdots \\ u_m \end{bmatrix}, \quad \vec{f}(\vec{x}, \vec{u}) = \begin{bmatrix} f_1(x_1, \cdots, x_n; u_1, \cdots, u_m) \\ \vdots \\ f_n(x_1, \cdots, x_n; u_1, \cdots, u_m) \end{bmatrix}$, then

nxn matrix

$$\boldsymbol{J}_x(\vec{x}, \vec{u}) = \nabla_x \vec{f}(\vec{x}, \vec{u}) = \left.\frac{\partial \vec{f}}{\partial \vec{x}}\right|_{\vec{x}, \vec{u}} \triangleq \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_{n-1}} & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_{n-1}} & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ \frac{\partial f_{n-1}}{\partial x_1} & \frac{\partial f_{n-1}}{\partial x_2} & \cdots & \frac{\partial f_{n-1}}{\partial x_{n-1}} & \frac{\partial f_{n-1}}{\partial x_n} \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_{n-1}} & \frac{\partial f_n}{\partial x_n} \end{bmatrix}$$

$$\boldsymbol{J}_u(\vec{x}, \vec{u}) = \nabla_u \vec{f}(\vec{x}, \vec{u}) = \left.\frac{\partial \vec{f}}{\partial \vec{u}}\right|_{\vec{x}, \vec{u}} \triangleq \begin{bmatrix} \frac{\partial f_1}{\partial u_1} & \frac{\partial f_1}{\partial u_2} & \cdots & \frac{\partial f_1}{\partial u_{m-1}} & \frac{\partial f_1}{\partial u_m} \\ \frac{\partial f_2}{\partial u_1} & \frac{\partial f_2}{\partial u_2} & \cdots & \frac{\partial f_2}{\partial u_{m-1}} & \frac{\partial f_2}{\partial u_m} \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ \frac{\partial f_{n-1}}{\partial u_1} & \frac{\partial f_{n-1}}{\partial u_2} & \cdots & \frac{\partial f_{n-1}}{\partial u_{m-1}} & \frac{\partial f_{n-1}}{\partial u_m} \\ \frac{\partial f_n}{\partial u_1} & \frac{\partial f_n}{\partial u_2} & \cdots & \frac{\partial f_n}{\partial u_{m-1}} & \frac{\partial f_n}{\partial u_m} \end{bmatrix}$$

Jacobian (Gradient) Matrices

- If: $\vec{x}(t) = \begin{bmatrix} x_1 \\ \vdots \\ \vdots \\ x_n \end{bmatrix}, \quad \vec{u}(t) = \begin{bmatrix} u_1 \\ \vdots \\ u_m \end{bmatrix}, \quad \vec{f}(\vec{x},\vec{u}) = \begin{bmatrix} f_1(x_1,\cdots,x_n;u_1,\cdots,u_m) \\ \vdots \\ f_n(x_1,\cdots,x_n;u_1,\cdots,u_m) \end{bmatrix}$, then

nxn matrix

$$J_x(\vec{x},\vec{u}) = \nabla_x \vec{f}(\vec{x},\vec{u}) = \left.\frac{\partial \vec{f}}{\partial \vec{x}}\right|_{\vec{x},\vec{u}} \triangleq \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_{n-1}} & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_{n-1}} & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ \frac{\partial f_{n-1}}{\partial x_1} & \frac{\partial f_{n-1}}{\partial x_2} & \cdots & \frac{\partial f_{n-1}}{\partial x_{n-1}} & \frac{\partial f_{n-1}}{\partial x_n} \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_{n-1}} & \frac{\partial f_n}{\partial x_n} \end{bmatrix}$$

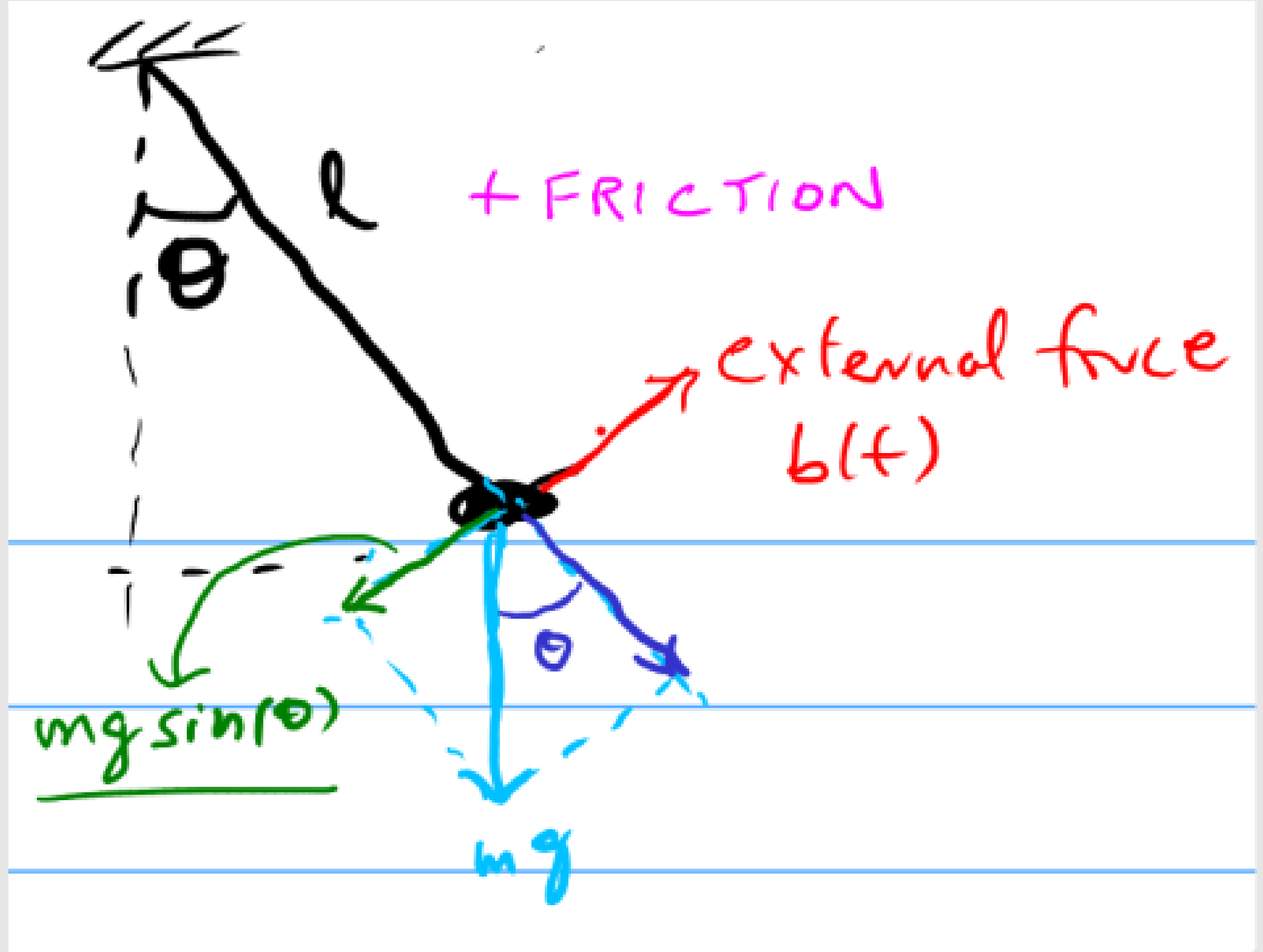
nxm matrix

$$J_u(\vec{x},\vec{u}) = \nabla_u \vec{f}(\vec{x},\vec{u}) = \left.\frac{\partial \vec{f}}{\partial \vec{u}}\right|_{\vec{x},\vec{u}} \triangleq \begin{bmatrix} \frac{\partial f_1}{\partial u_1} & \frac{\partial f_1}{\partial u_2} & \cdots & \frac{\partial f_1}{\partial u_{m-1}} & \frac{\partial f_1}{\partial u_m} \\ \frac{\partial f_2}{\partial u_1} & \frac{\partial f_2}{\partial u_2} & \cdots & \frac{\partial f_2}{\partial u_{m-1}} & \frac{\partial f_2}{\partial u_m} \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ \frac{\partial f_{n-1}}{\partial u_1} & \frac{\partial f_{n-1}}{\partial u_2} & \cdots & \frac{\partial f_{n-1}}{\partial u_{m-1}} & \frac{\partial f_{n-1}}{\partial u_m} \\ \frac{\partial f_n}{\partial u_1} & \frac{\partial f_n}{\partial u_2} & \cdots & \frac{\partial f_n}{\partial u_{m-1}} & \frac{\partial f_n}{\partial u_m} \end{bmatrix}$$

Example: Linearizing the Pendulum

- Pendulum:
  - (move to xournal)

$$\frac{d\vec{x}}{dt} = \begin{bmatrix} v_\theta \\ -\frac{g}{l}\sin(\theta) - \frac{k}{m} v_\theta + \frac{b(t)}{ml} \end{bmatrix}$$



$$\vec{x} = \begin{bmatrix} \theta \\ v_\theta \end{bmatrix}, \quad \vec{u} = \begin{bmatrix} b(t) \end{bmatrix}$$

— $n=2$, $m=1$.

— DC input: $u(t) \equiv 0 = u^*$ (no force)

— DC solution: $\vec{x}(t) = \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \vec{x}^*$ ($\theta^* = 0$, $v_\theta^* = 0$) : at rest

$\longrightarrow$ Therefore $\vec{\Delta x} \equiv \vec{x}$, $\Delta u \equiv u \implies$ $u(t)$ is small, assume $x(t)$ is small

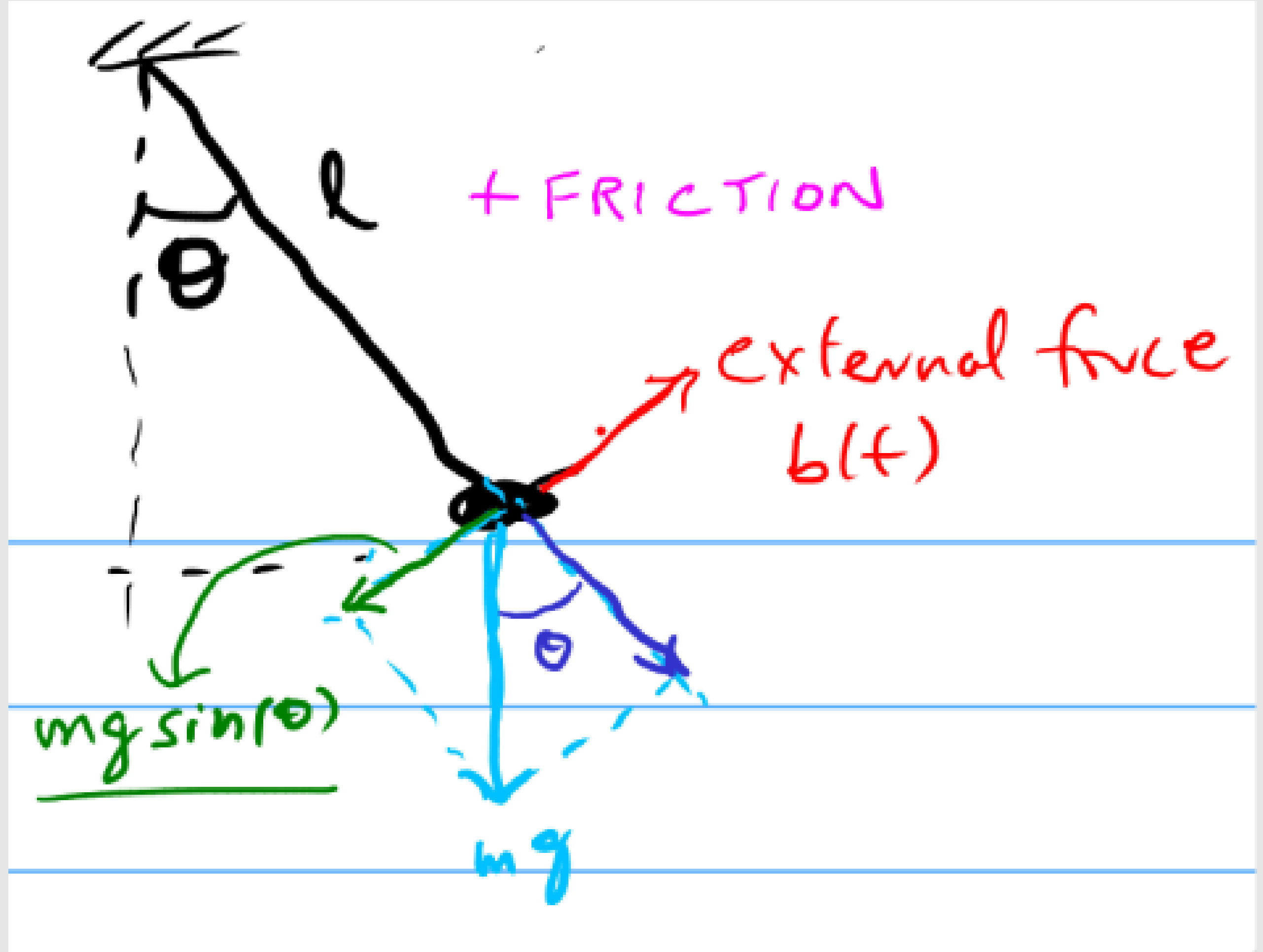
$$\longrightarrow \quad J_{\vec{x}} = \overbrace{\begin{bmatrix} 0 & 1 \\ -\frac{g}{l}\underbrace{\cos(\theta^*)}_{1} & -\frac{k}{m} \end{bmatrix}}^{2\times 2 \text{ matrix}} ; \quad J_{\vec{u}} = \overbrace{\begin{bmatrix} 0 \\ +\frac{1}{ml} \end{bmatrix}}^{2\times 1 \text{ "matrix"}}$$

Example: Linearizing the Pendulum

- Pendulum:

$$\frac{d\vec{x}}{dt} = \begin{bmatrix} v_\theta \\ -\frac{g}{l}\sin(\theta) - \frac{k}{m} v_\theta + \frac{b(t)}{ml} \end{bmatrix}$$

$$\vec{x} = \begin{bmatrix} \theta \\ v_\theta \end{bmatrix}, \quad \vec{u} = [b(t)]$$



— $n=2$, $m=1$.

— DC input: $u(t) \equiv 0 = u^*$ (no force)

— DC solution: $\vec{x}(t) = \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \vec{x}^*$ ($\theta^* = 0$, $v_\theta^* = 0$) : at rest

→ Therefore $\vec{\Delta x} \equiv \vec{x}$, $\Delta u \equiv u$ $\Longrightarrow$ $u(t)$ is small, assume $x(t)$ is small

2x2 matrix

2x1 "matrix"

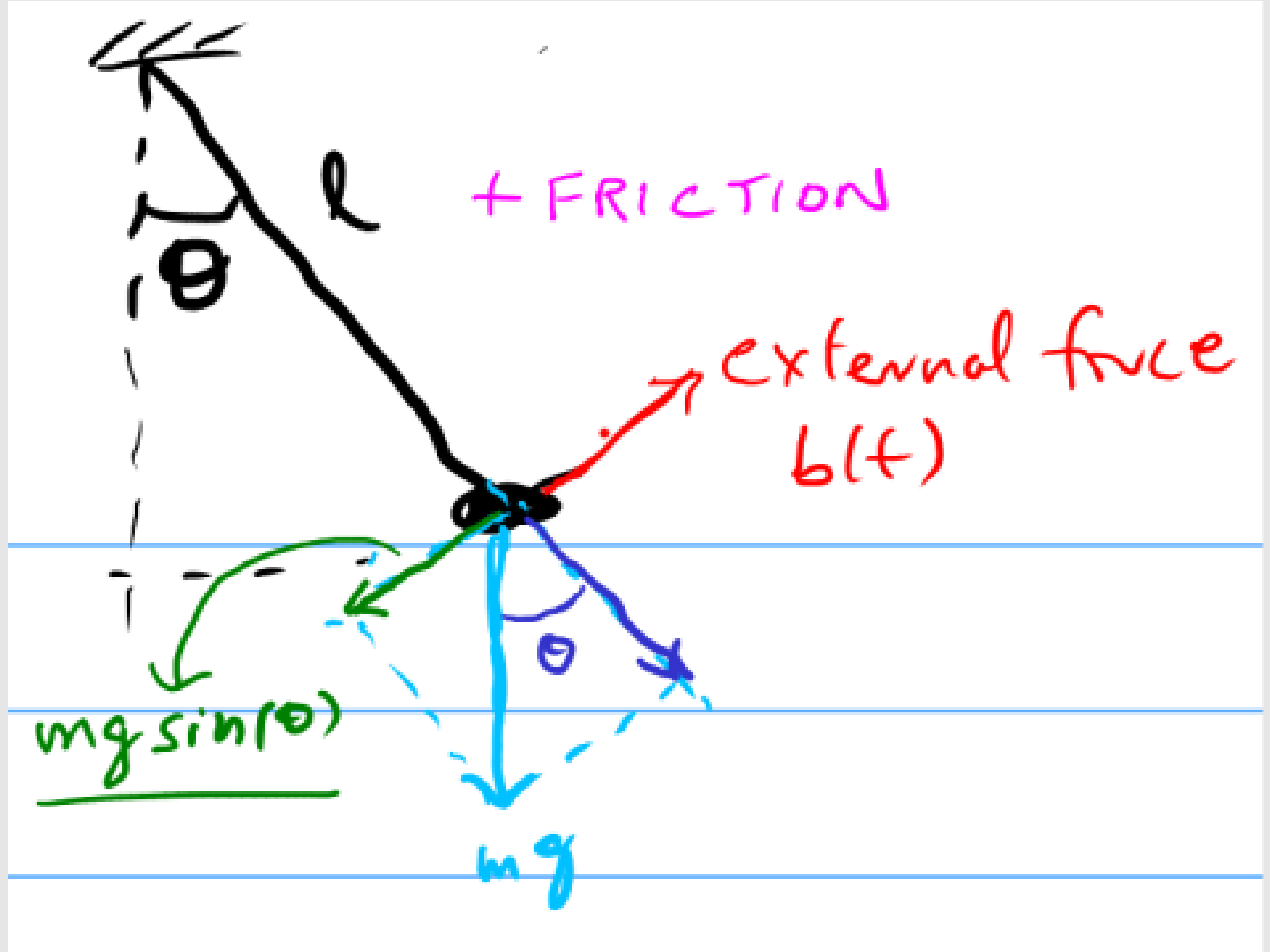
→ $J_{\vec{x}} = \begin{bmatrix} 0 & 1 \\ -\frac{g}{l}\cos(\theta^*) & -\frac{k}{m} \end{bmatrix}$; $J_{\vec{u}} = \begin{bmatrix} 0 \\ \frac{+1}{ml} \end{bmatrix}$

Example: Linearizing the Pendulum

- Pendulum:

$$\frac{d\vec{x}}{dt} = \begin{bmatrix} v_\theta \\ -g/l \sin(\theta) - k/m \, v_\theta + \frac{b(t)}{ml} \end{bmatrix}$$

$$\vec{x} = \begin{bmatrix} \theta \\ v_\theta \end{bmatrix}, \quad \vec{u} = [b(t)]$$



— $n=2$, $m=1$.

— DC input: $u(t) \equiv 0 = u^*$ (no force)

— DC solution: $\vec{x}(t) = \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \vec{x}^*$ ($\theta^* = 0$, $v_\theta^* = 0$) : at rest

→ Therefore $\vec{\Delta x} \equiv \vec{x}$, $\Delta u \equiv u$ ⟹ $u(t)$ is small, assume $x(t)$ is small

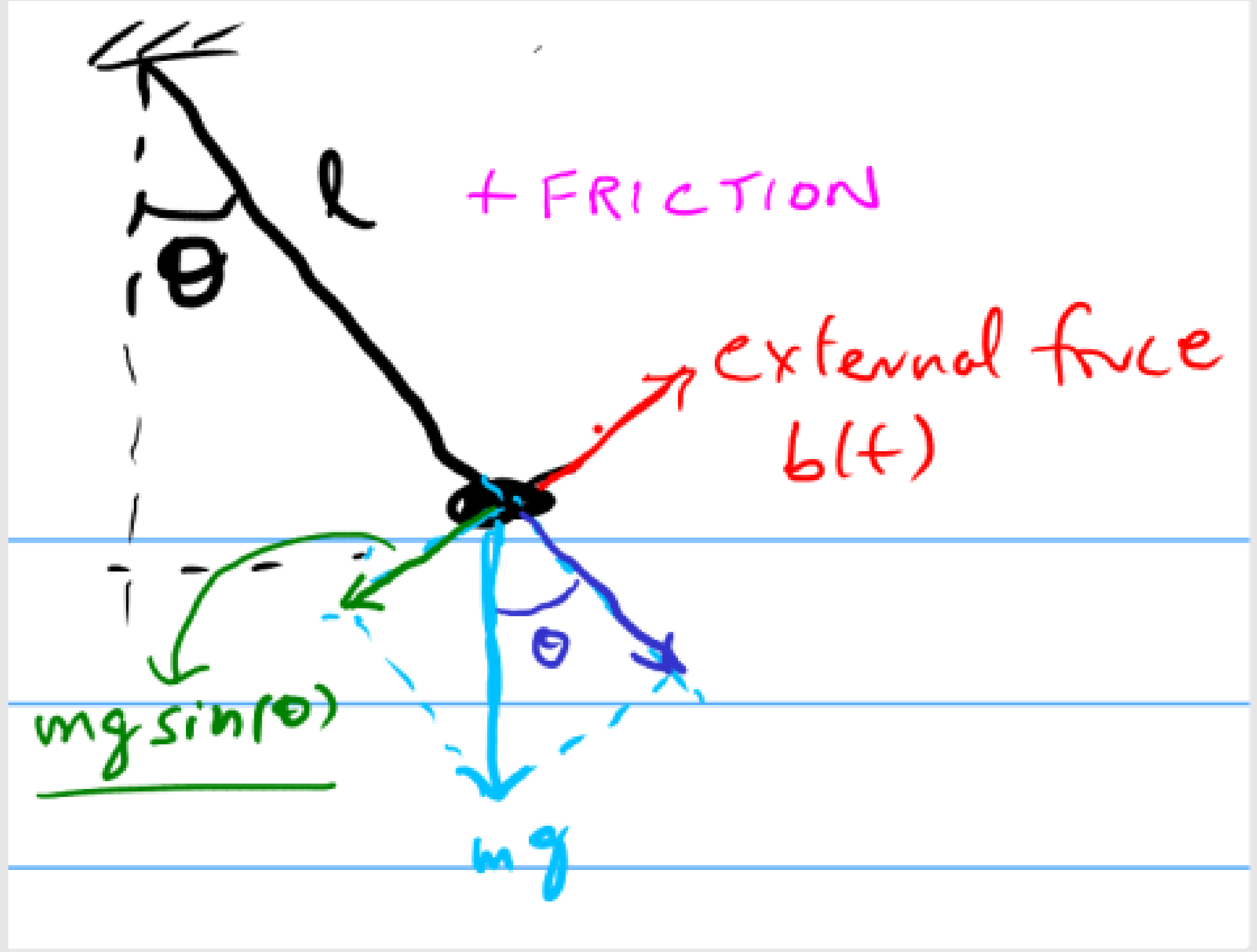
→ $J_{\vec{x}} = \overbrace{\begin{bmatrix} 0 & 1 \\ -g/l \underbrace{\cos(\theta^*)}_{1} & -k/m \end{bmatrix}}^{2\times 2 \text{ matrix}}$; $J_{\vec{u}} = \overbrace{\begin{bmatrix} 0 \\ +\frac{1}{ml} \end{bmatrix}}^{2\times 1 \text{ "matrix"}}$

$$\boxed{\frac{d\vec{x}}{dt} = \begin{bmatrix} 0 & 1 \\ -g/l & -k/m \end{bmatrix} \vec{x} + \begin{bmatrix} 0 \\ +\frac{u(t)}{ml} \end{bmatrix}}$$

Example: Linearizing the Pendulum

- Pendulum:

$$\frac{d\vec{x}}{dt} = \begin{bmatrix} v_\theta \\ -\frac{g}{l}\sin(\theta) - \frac{k}{m} v_\theta + \frac{b(t)}{ml} \end{bmatrix}$$



$$\vec{x} = \begin{bmatrix} \theta \\ v_\theta \end{bmatrix}, \quad \vec{u} = [b(t)]$$

— $n=2$, $m=1$.

— DC input: $u(t) \equiv 0 = u^*$ (no force)

— DC solution: $\vec{x}(t) = \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \vec{x}^*$ ($\theta^* = 0$, $v_\theta^* = 0$) : at rest

→ Therefore $\vec{\Delta x} \equiv \vec{x}$, $\Delta u \equiv u$ ⟹ $u(t)$ is small, assume $x(t)$ is small

2×2 matrix; 2×1 "matrix"

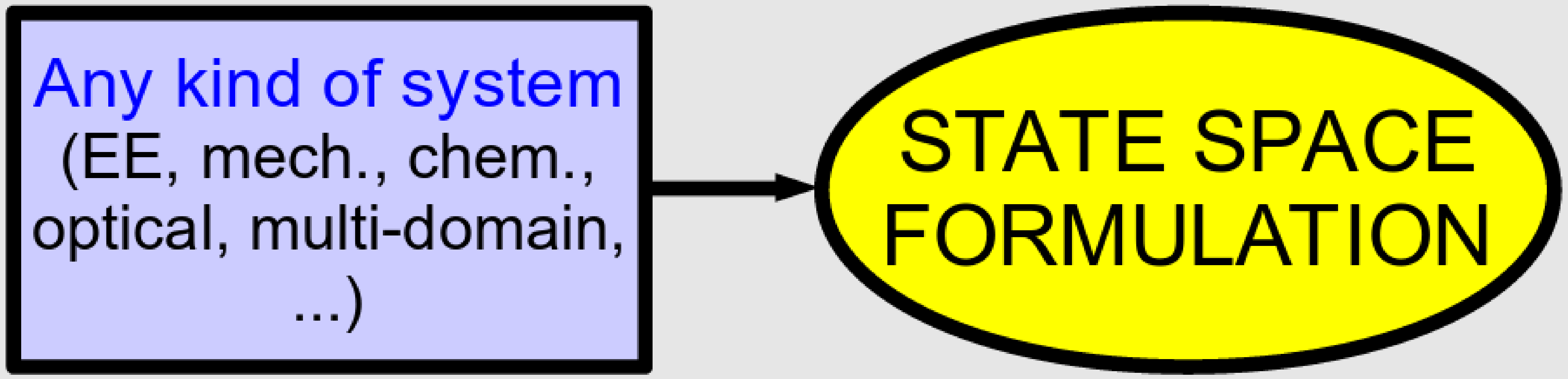
$$\rightarrow J_{\vec{x}} = \begin{bmatrix} 0 & 1 \\ -\frac{g}{l}\underbrace{\cos(\theta^*)}_{1} & -\frac{k}{m} \end{bmatrix}; \quad J_{\vec{u}} = \begin{bmatrix} 0 \\ +\frac{1}{ml} \end{bmatrix}$$

$$\boxed{\frac{d\vec{x}}{dt} = \begin{bmatrix} 0 & 1 \\ -\frac{g}{l} & -\frac{k}{m} \end{bmatrix} \vec{x} + \begin{bmatrix} 0 \\ +\frac{u(t)}{ml} \end{bmatrix}}$$

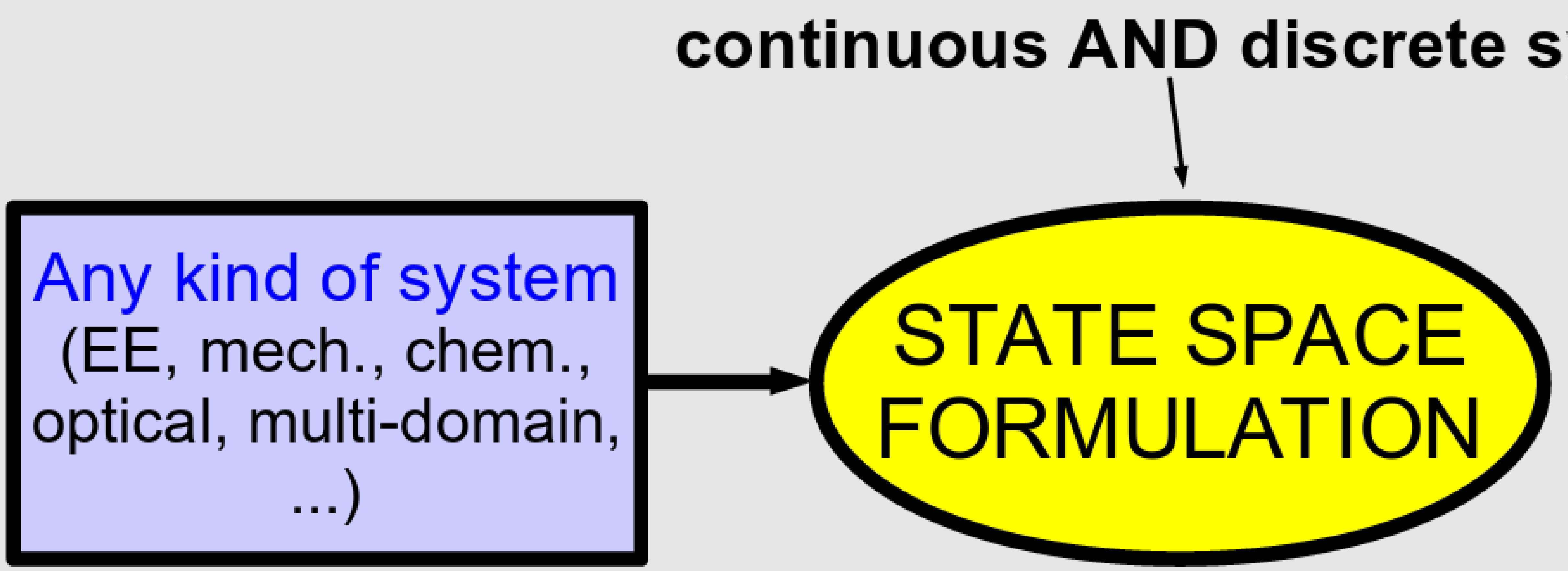
- Compare against $\sin(\theta) \approx \theta$ approximation (prev. class)

$$\frac{d\vec{x}}{dt} = \begin{bmatrix} 0 & 1 \\ -\frac{g}{l} & -\frac{d}{m} \end{bmatrix} \vec{x}(t) + \begin{bmatrix} 0 \\ \frac{1}{ml} \end{bmatrix} \vec{u}(t)$$

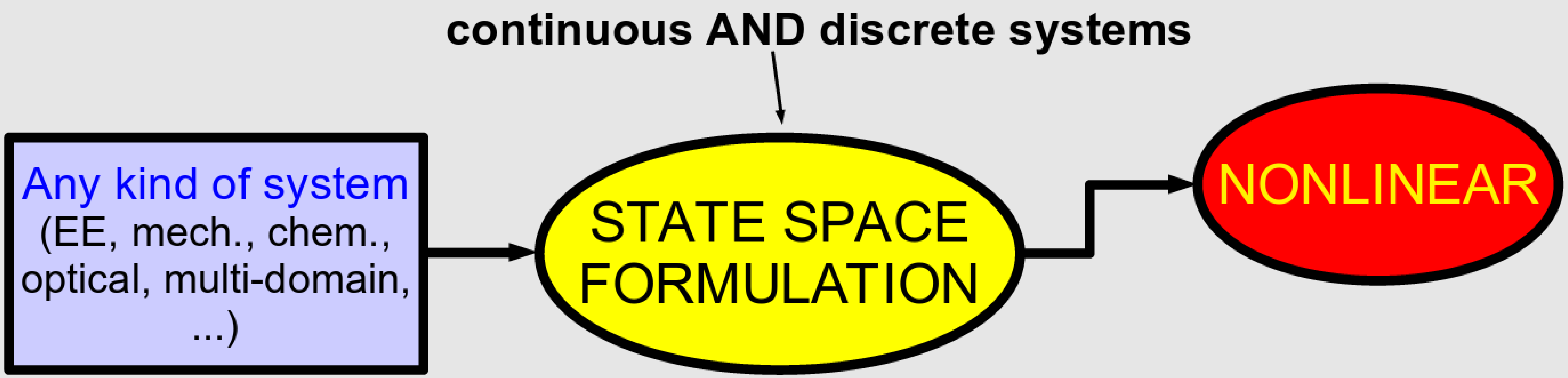
Where We Are Now



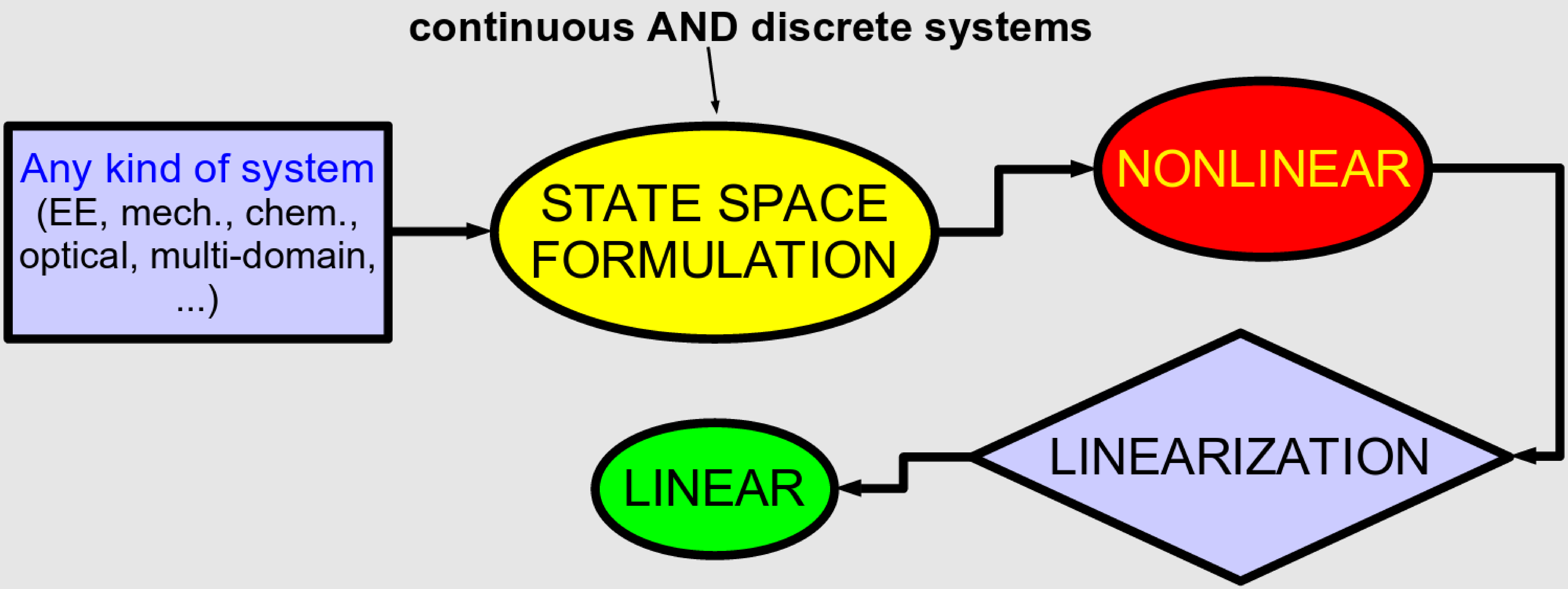
Where We Are Now



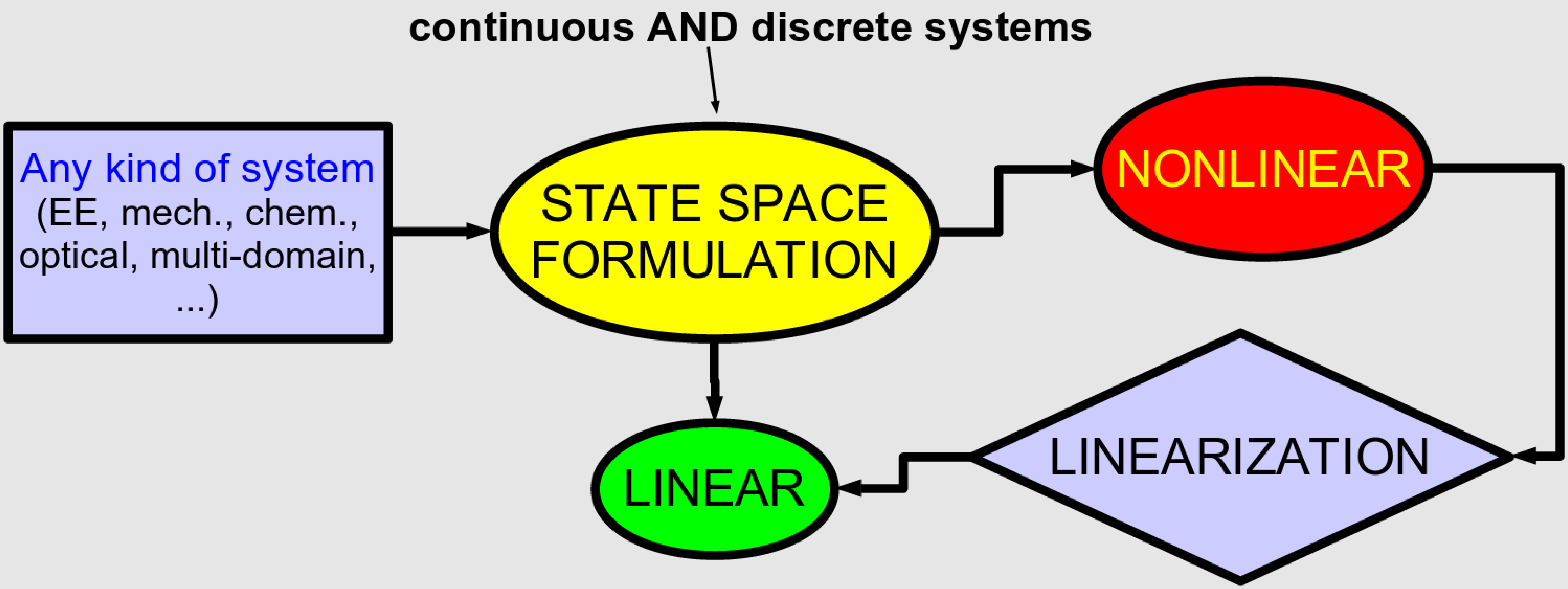
Where We Are Now



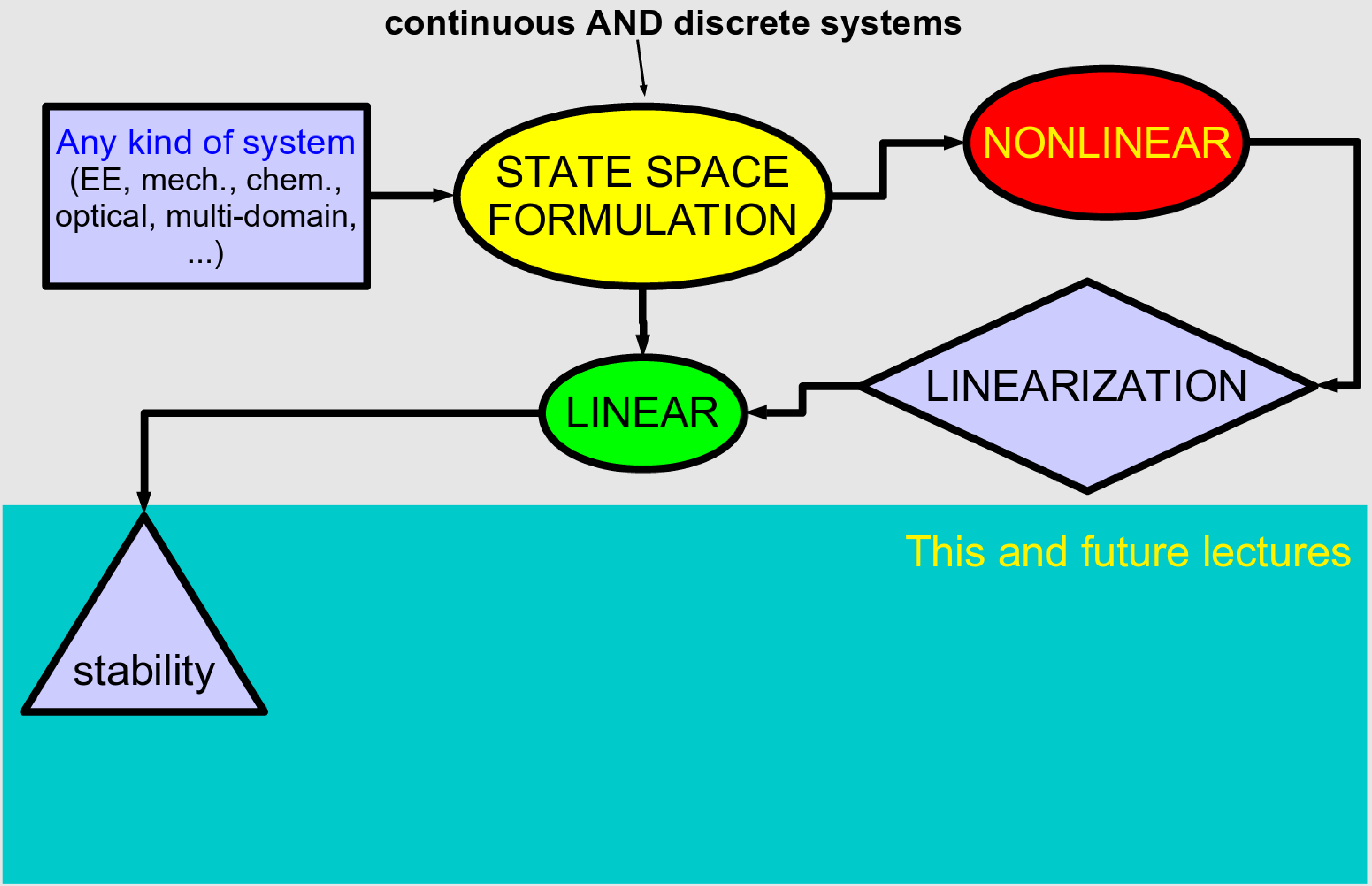
Where We Are Now



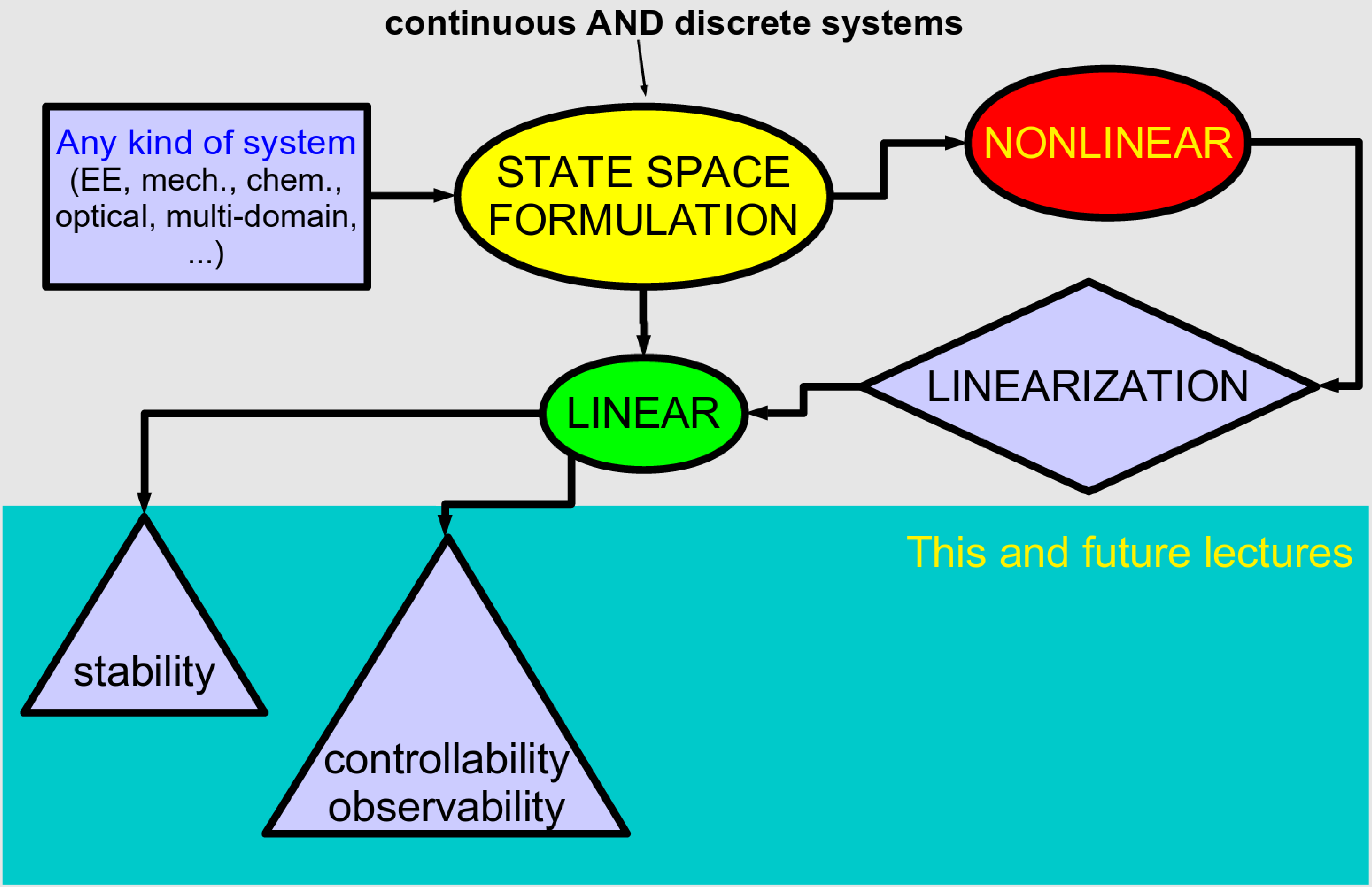
Where We Are Now



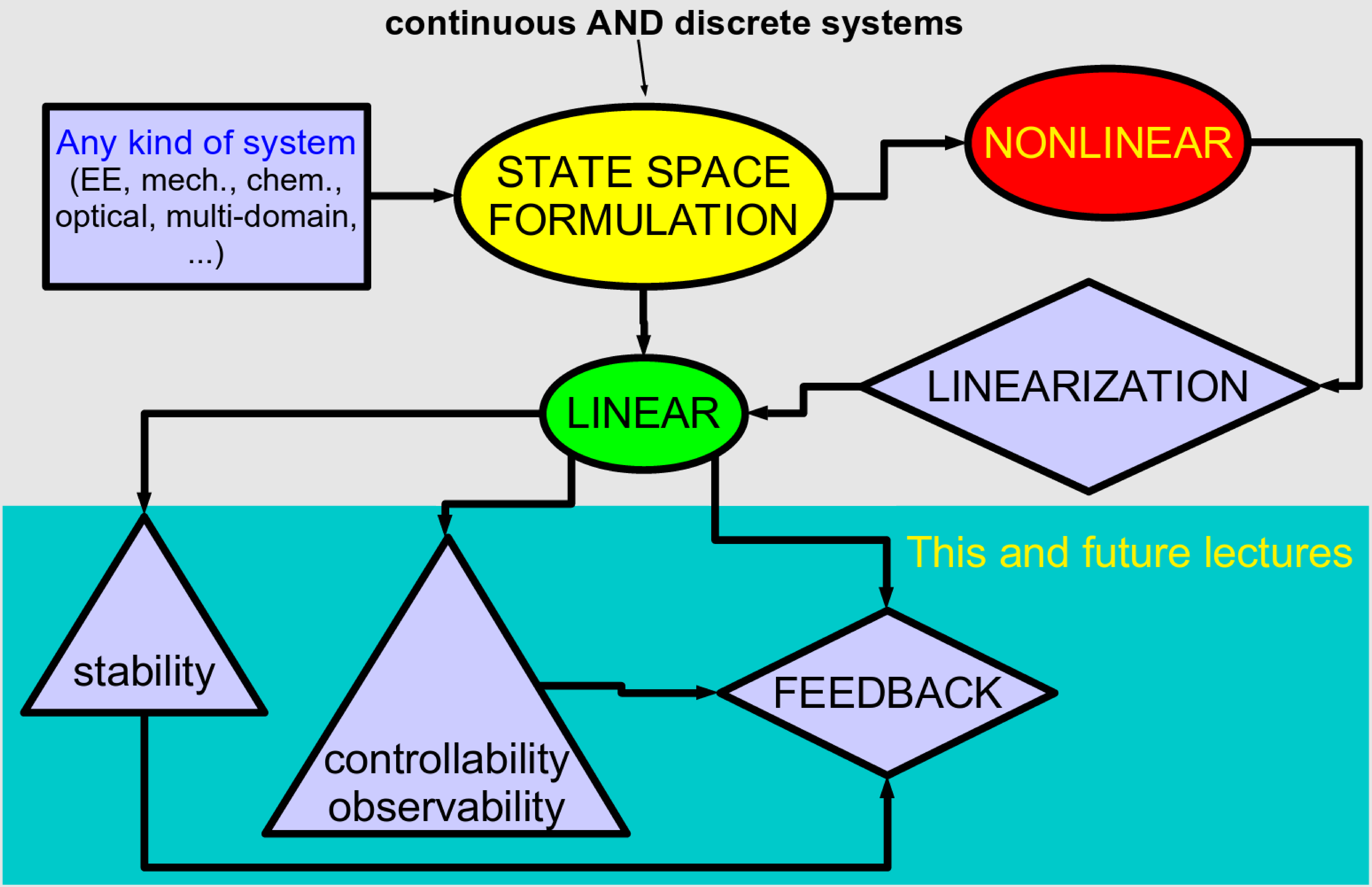
Where We Are Now



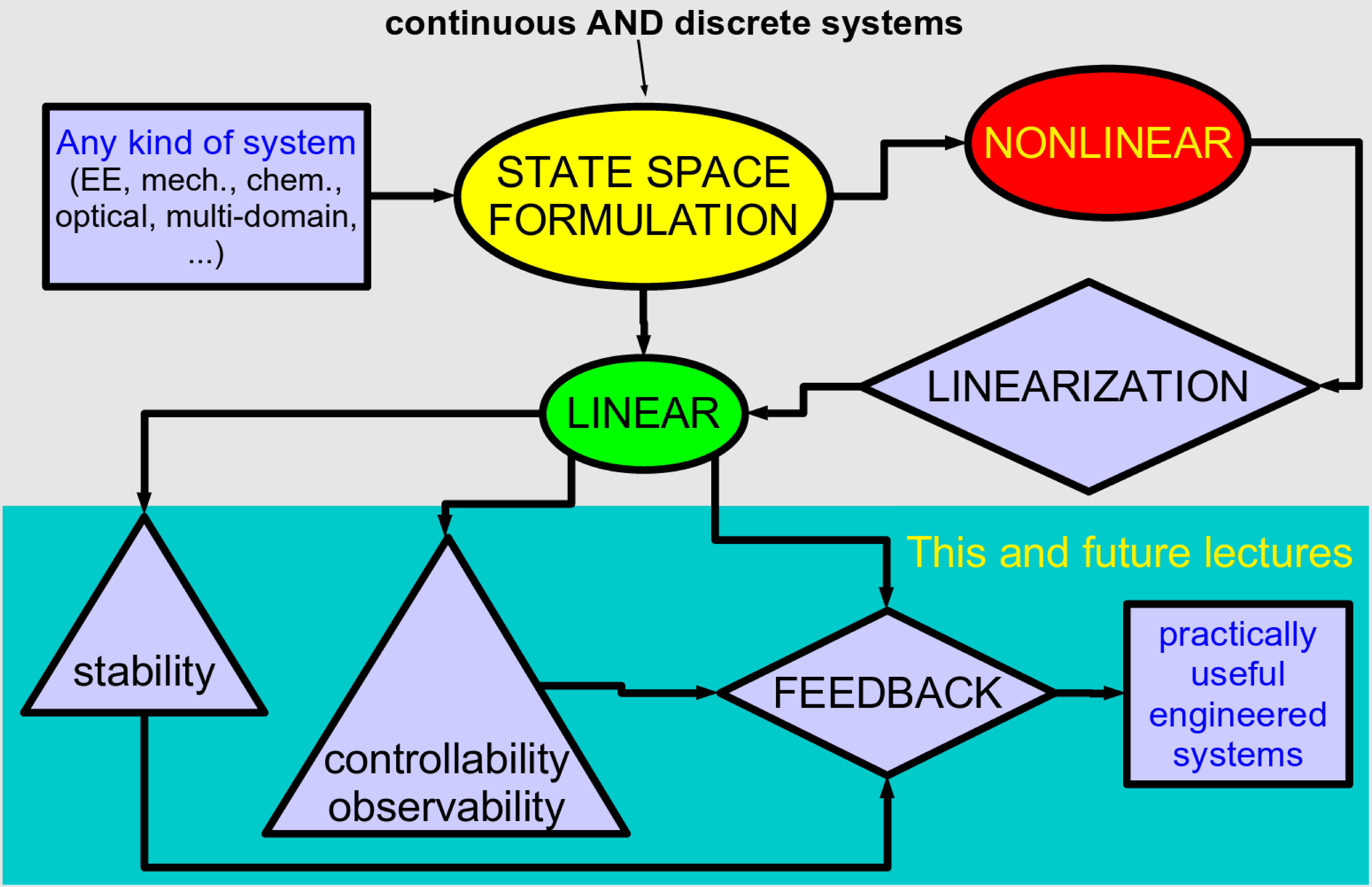
Where We Are Now



Where We Are Now

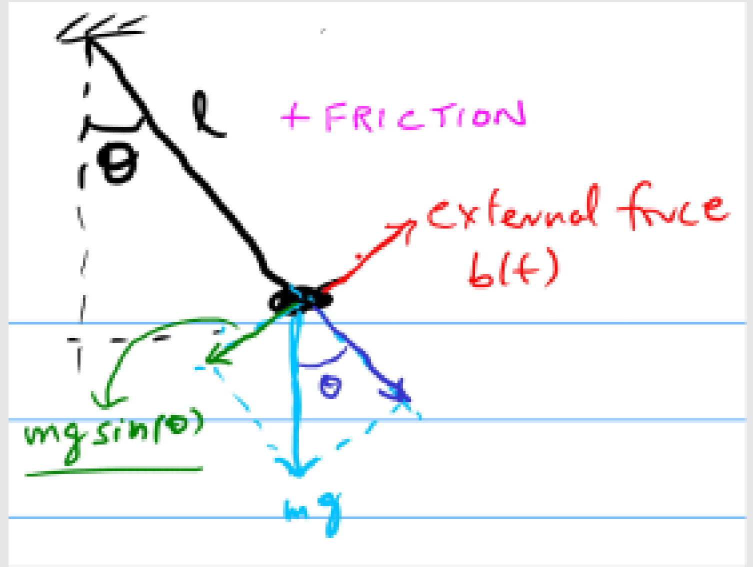


Where We Are Now



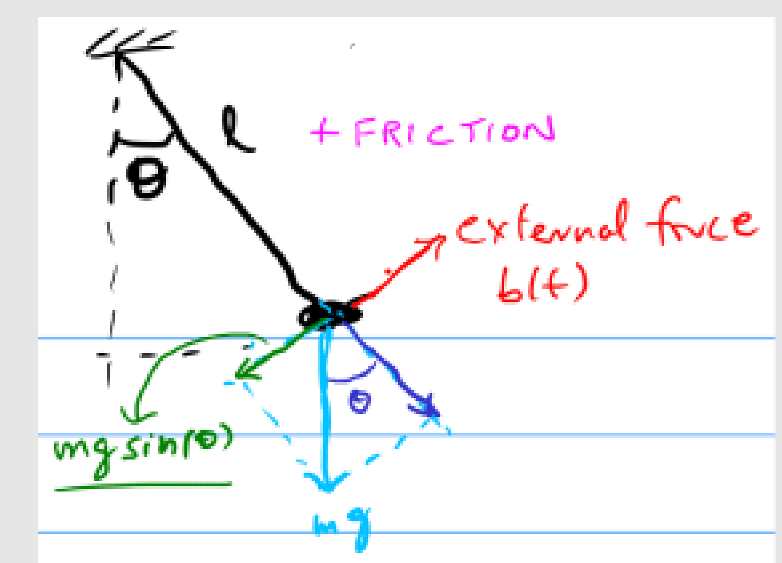
Pendulum: Inverted Solution

- Pendulum: $$\frac{d\vec{x}}{dt} = \begin{bmatrix} v_\theta \\ -\frac{g}{\ell}\sin(\theta) - \frac{k}{m} v_\theta + \frac{b(t)}{m\ell} \end{bmatrix}$$ $$\vec{x} = \begin{bmatrix} \theta \\ v_\theta \end{bmatrix}, \quad \vec{u} = [b(t)]$$



Pendulum: Inverted Solution

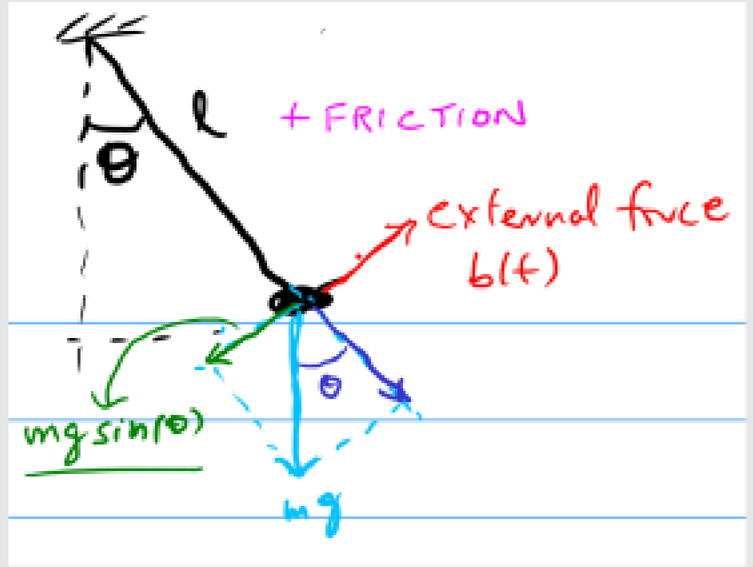
- Pendulum: $\frac{d\vec{x}}{dt} = \begin{bmatrix} v_\theta \\ -g/l \sin(\theta) - k/m\, v_\theta + \frac{b(t)}{ml} \end{bmatrix}$ $\quad \vec{x} = \begin{bmatrix} \theta \\ v_\theta \end{bmatrix}, \quad \vec{u} = [b(t)]$
- DC input: b(t) = b*



Pendulum: Inverted Solution

- Pendulum: $\frac{d\vec{x}}{dt} = \begin{bmatrix} v_\theta \\ -g/l \sin(\theta) - k/m\, v_\theta + \frac{b(t)}{ml} \end{bmatrix}$ $\vec{x} = \begin{bmatrix} \theta \\ v_\theta \end{bmatrix}$, $\vec{u} = [b(t)]$

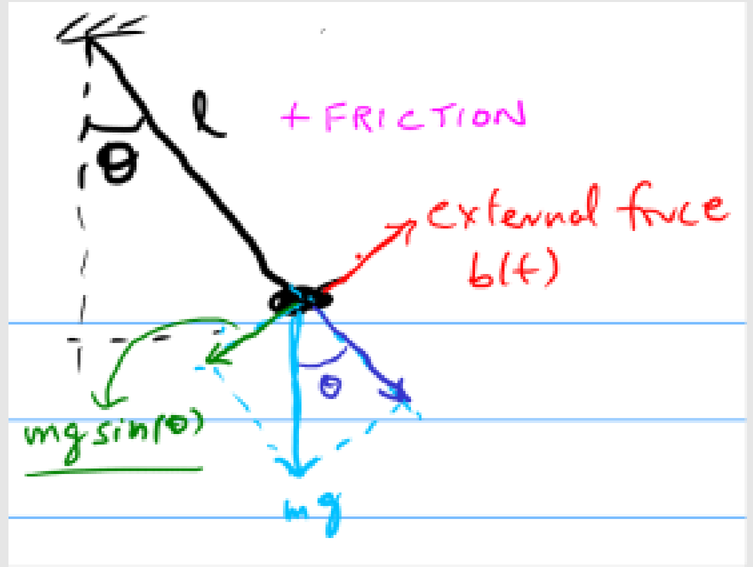
- DC input: b(t) = b*
- DC solution: dx/dt = 0 → $v_\theta = 0$, $\frac{g}{l}\sin(\theta) = \frac{b^*}{ml}$



Pendulum: Inverted Solution

- Pendulum: $\dfrac{d\vec{x}}{dt} = \begin{bmatrix} v_\theta \\ -g/l \sin(\theta) - k/m\, v_\theta + \dfrac{b(t)}{ml} \end{bmatrix}$ $\qquad \vec{x} = \begin{bmatrix} \theta \\ v_\theta \end{bmatrix}, \quad \vec{u} = \begin{bmatrix} b(t) \end{bmatrix}$

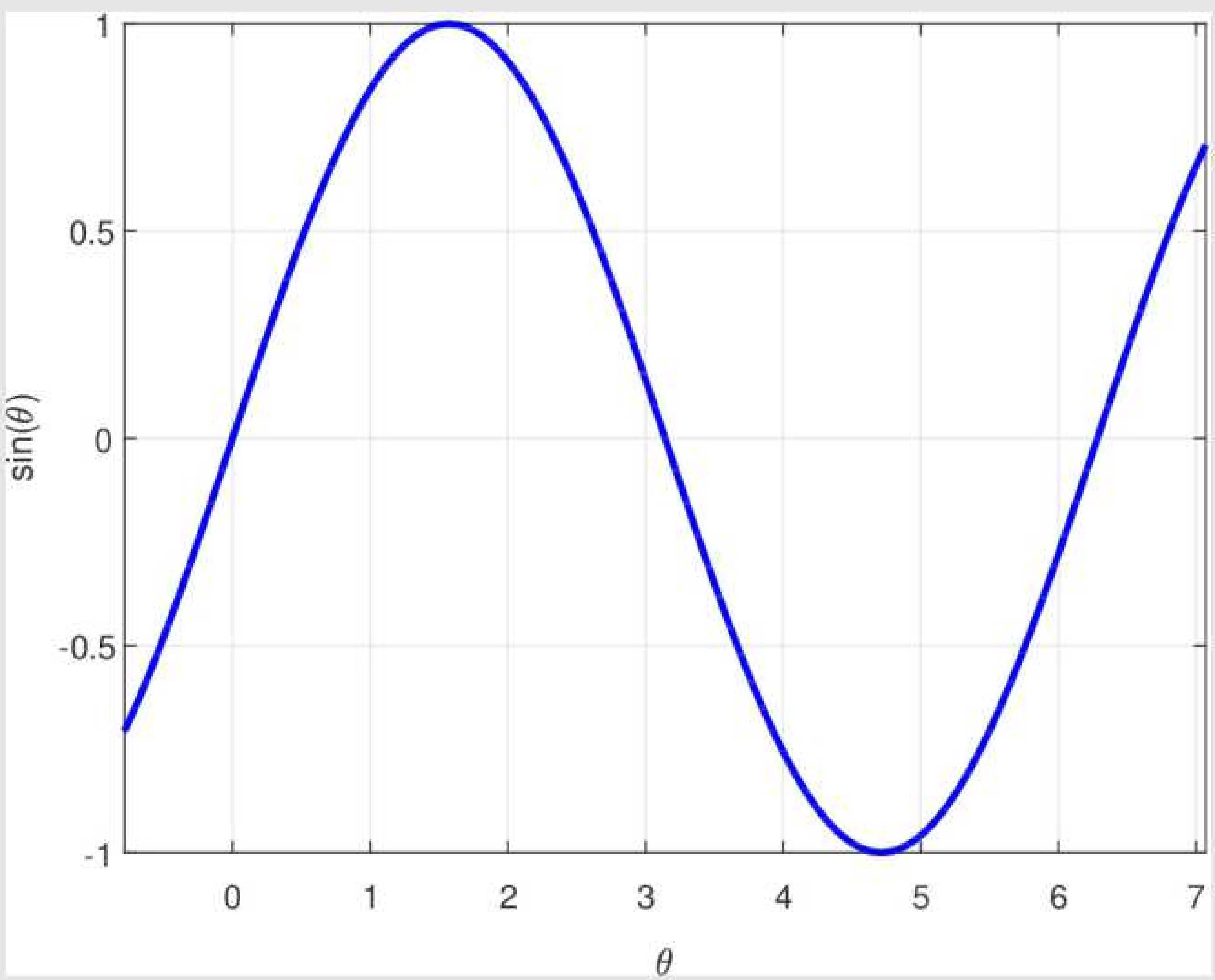
- DC input: b(t) = b*
- DC solution: dx/dt = 0 $\longrightarrow$ $v_\theta = 0$, $\dfrac{g}{l}\sin(\theta) = \dfrac{b^*}{ml}$ $\Rightarrow \sin(\theta) = \dfrac{b^*}{mg}$

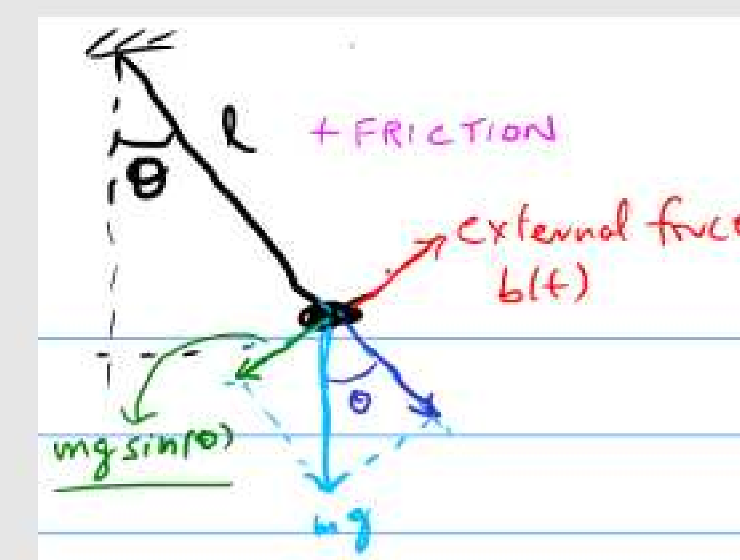


Pendulum: Inverted Solution

- Pendulum: $\frac{d\vec{x}}{dt} = \begin{bmatrix} v_\theta \\ -\frac{g}{l}\sin(\theta) - \frac{k}{m} v_\theta + \frac{b(t)}{ml} \end{bmatrix}$ $\quad \vec{x} = \begin{bmatrix} \theta \\ v_\theta \end{bmatrix}, \quad \vec{u} = [b(t)]$

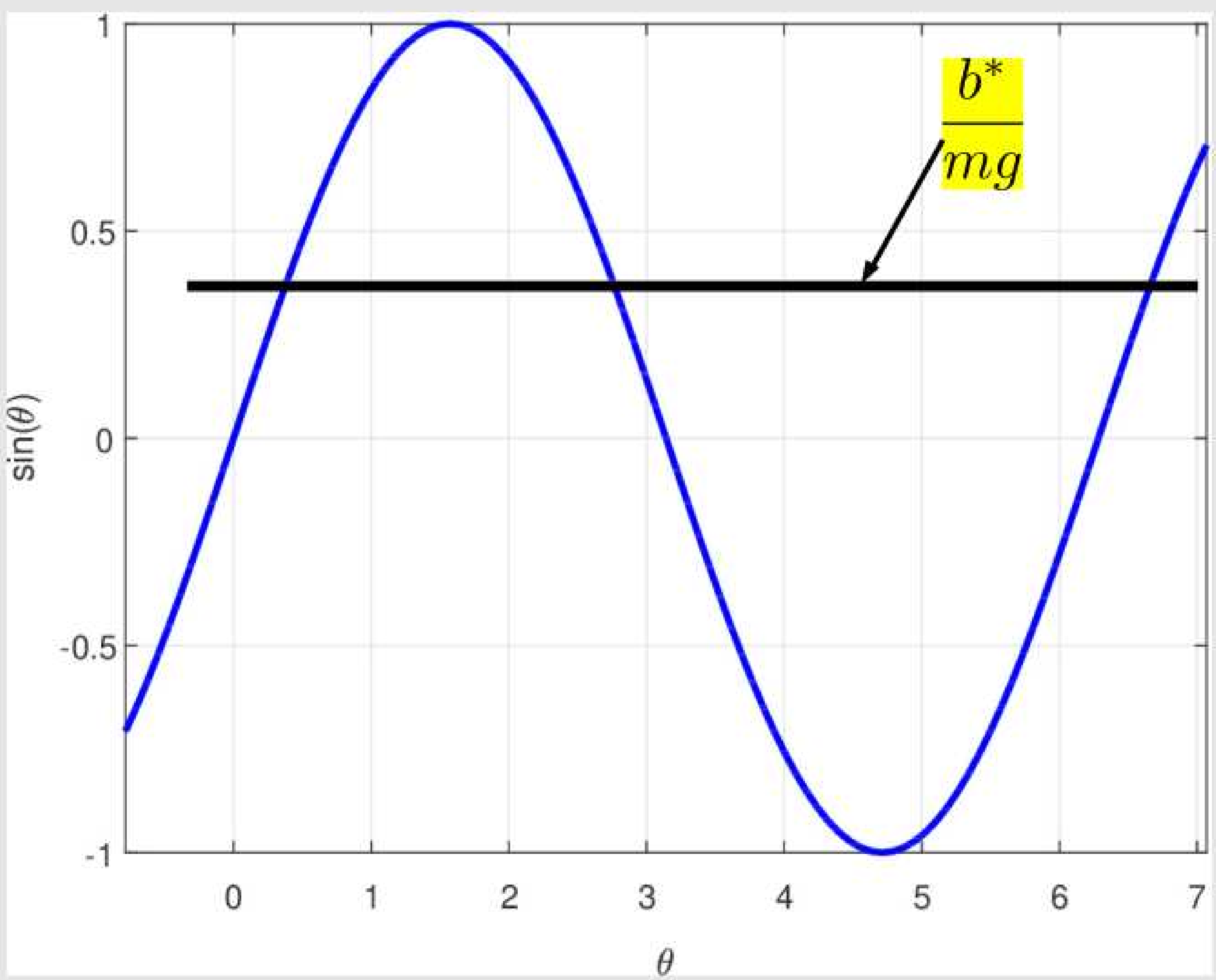
- DC input: b(t) = b*
- DC solution: dx/dt = 0 $\longrightarrow$ $v_\theta = 0$, $\frac{g}{l}\sin(\theta) = \frac{b^*}{ml}$ $\Rightarrow \sin(\theta) = \frac{b^*}{mg}$

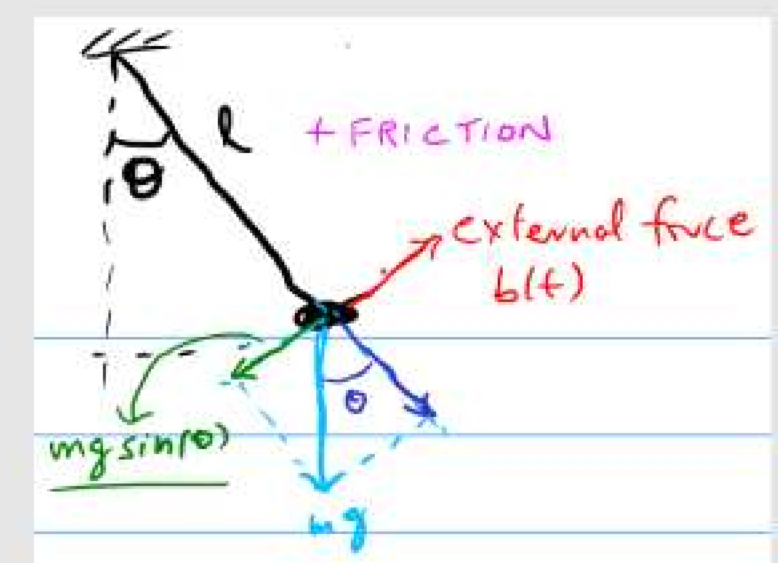




Pendulum: Inverted Solution

- Pendulum: $\frac{d\vec{x}}{dt} = \begin{bmatrix} v_\theta \\ -\frac{g}{l}\sin(\theta) - \frac{k}{m} v_\theta + \frac{b(t)}{ml} \end{bmatrix}$ $\vec{x} = \begin{bmatrix} \theta \\ v_\theta \end{bmatrix}$, $\vec{u} = [b(t)]$
- DC input: b(t) = b*
- DC solution: dx/dt = 0 $\longrightarrow$ $v_\theta = 0$, $\frac{g}{l}\sin(\theta) = \frac{b^*}{ml}$ $\Rightarrow \sin(\theta) = \frac{b^*}{mg}$

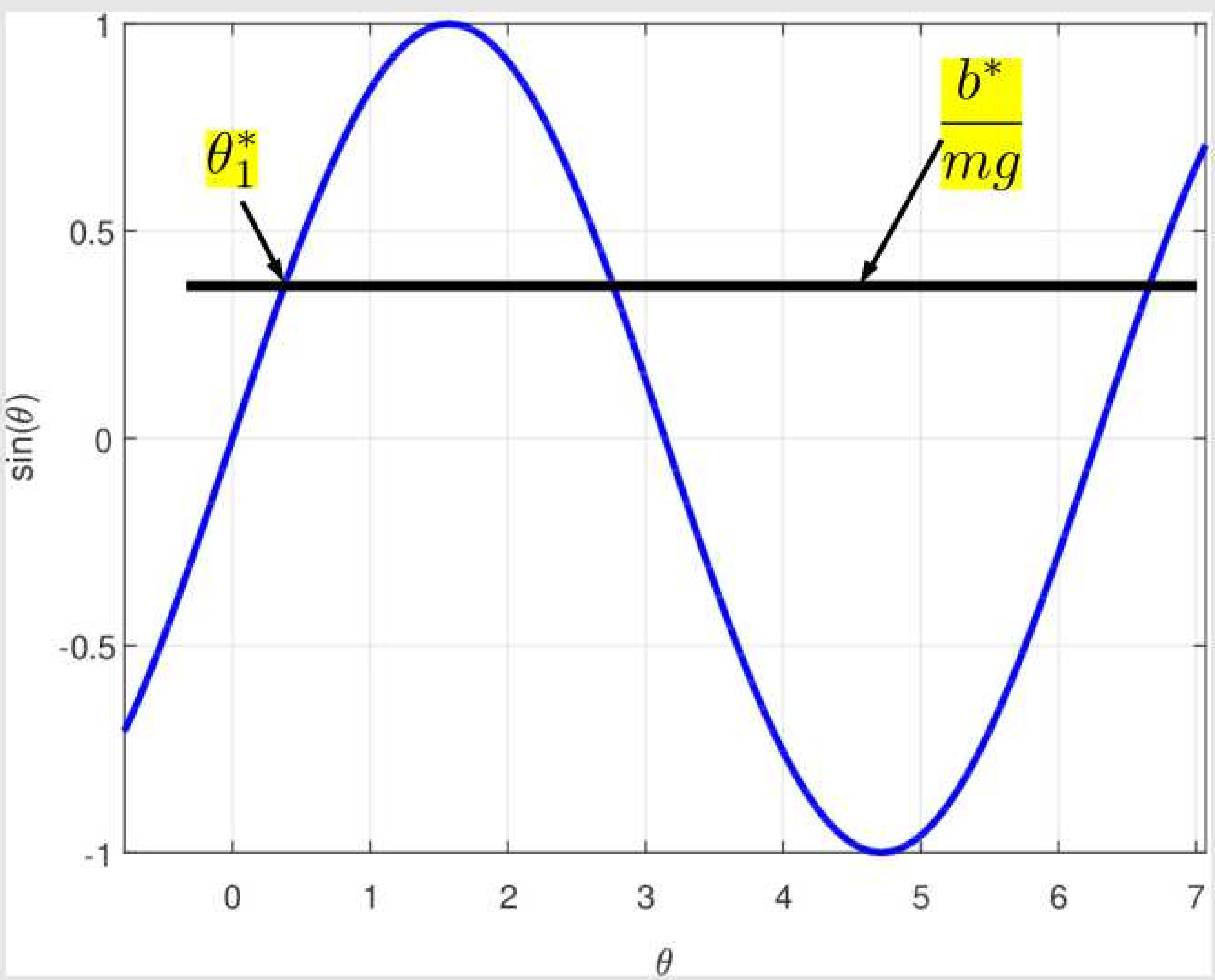


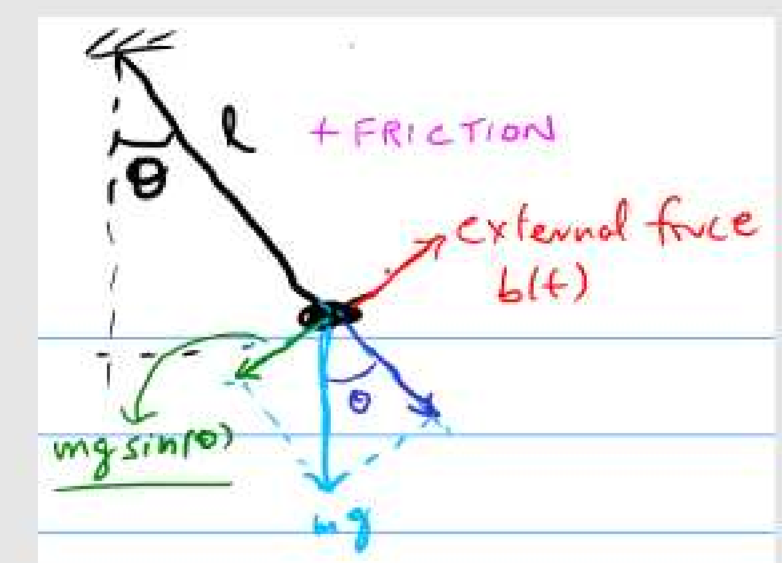


Pendulum: Inverted Solution

- Pendulum: $$\frac{d\vec{x}}{dt} = \begin{bmatrix} v_\theta \\ -\frac{g}{l}\sin(\theta) - \frac{k}{m} v_\theta + \frac{b(t)}{ml} \end{bmatrix}$$ $\vec{x} = \begin{bmatrix} \theta \\ v_\theta \end{bmatrix}, \quad \vec{u} = [b(t)]$

- DC input: b(t) = b*
- DC solution: dx/dt = 0 $\longrightarrow$ $v_\theta = 0$, $\frac{g}{l}\sin(\theta) = \frac{b^*}{ml}$ $\Rightarrow \sin(\theta) = \frac{b^*}{mg}$

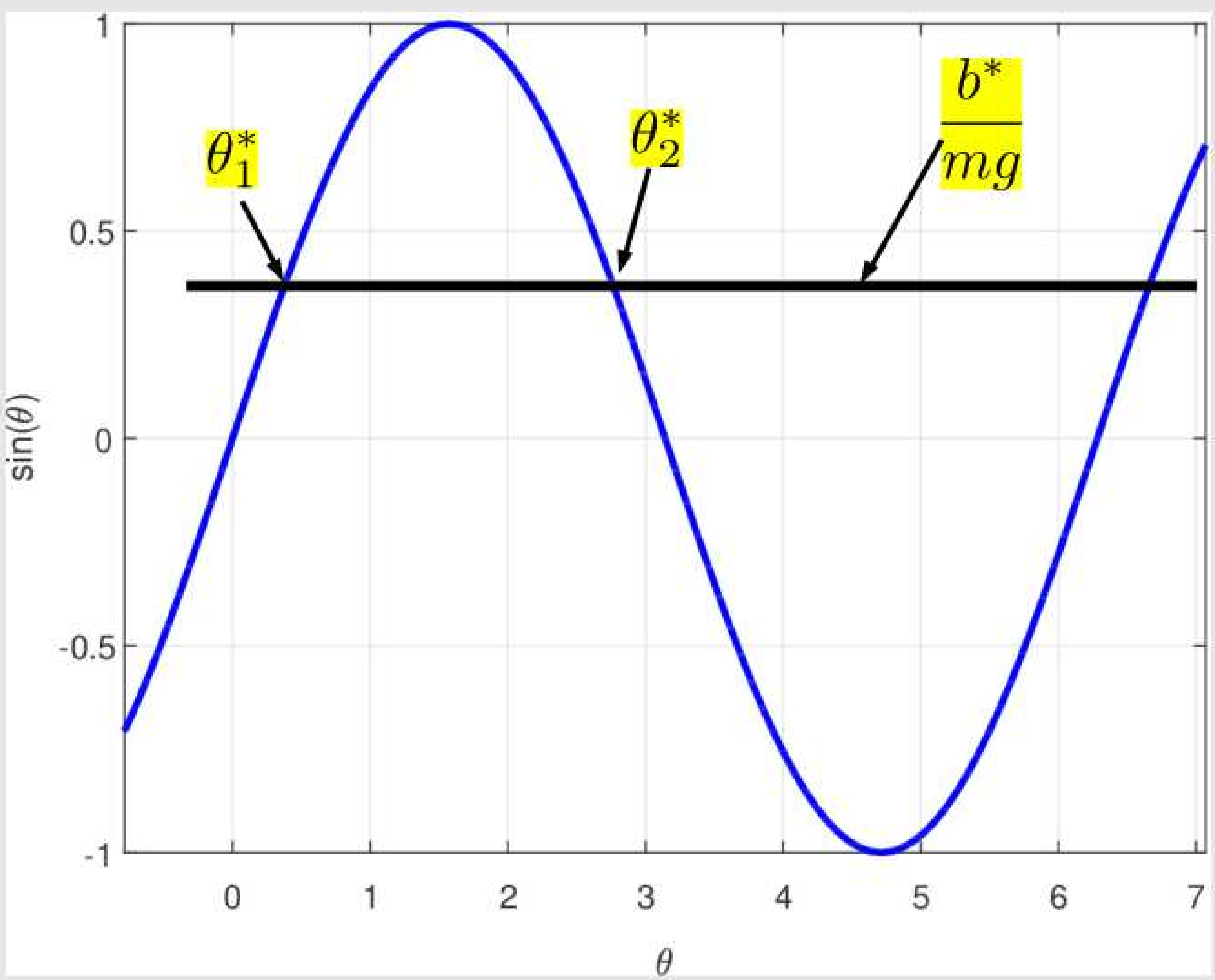


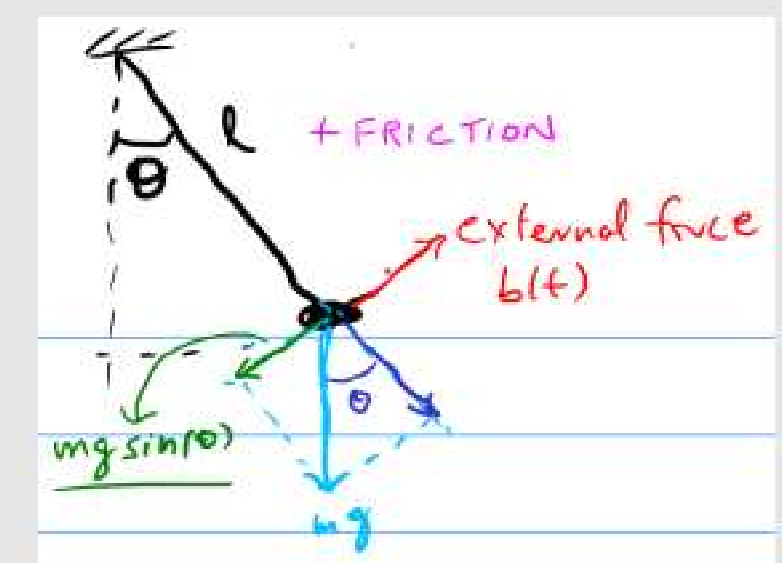


Pendulum: Inverted Solution

- Pendulum: $\frac{d\vec{x}}{dt} = \begin{bmatrix} v_\theta \\ -g/l \sin(\theta) - k/m\, v_\theta + \frac{b(t)}{ml} \end{bmatrix}$ $\quad \vec{x} = \begin{bmatrix} \theta \\ v_\theta \end{bmatrix}, \quad \vec{u} = [b(t)]$

- DC input: b(t) = b*
- DC solution: dx/dt = 0 $\longrightarrow$ $v_\theta = 0$, $\frac{g}{l}\sin(\theta) = \frac{b^*}{ml}$ $\Rightarrow \sin(\theta) = \frac{b^*}{mg}$

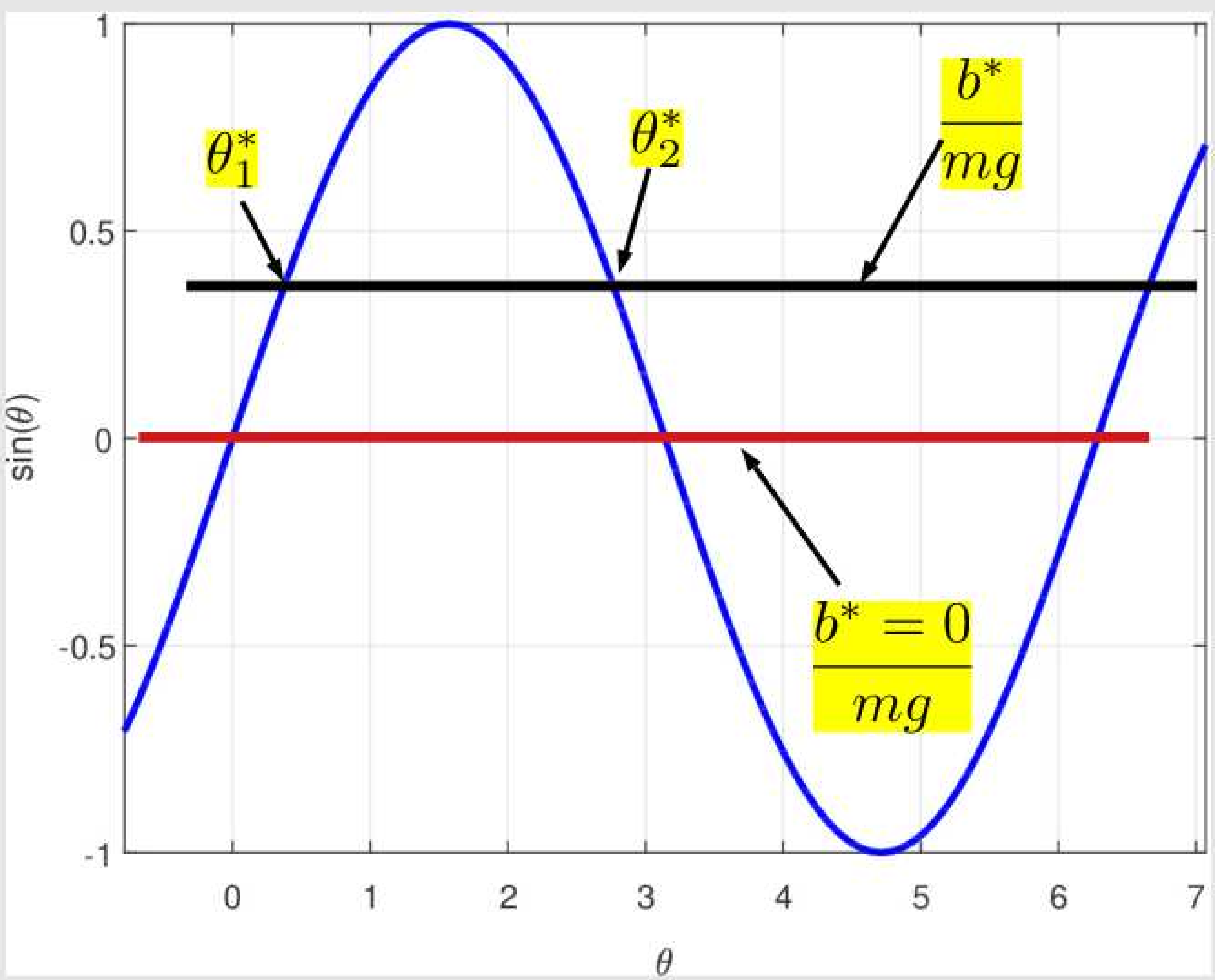


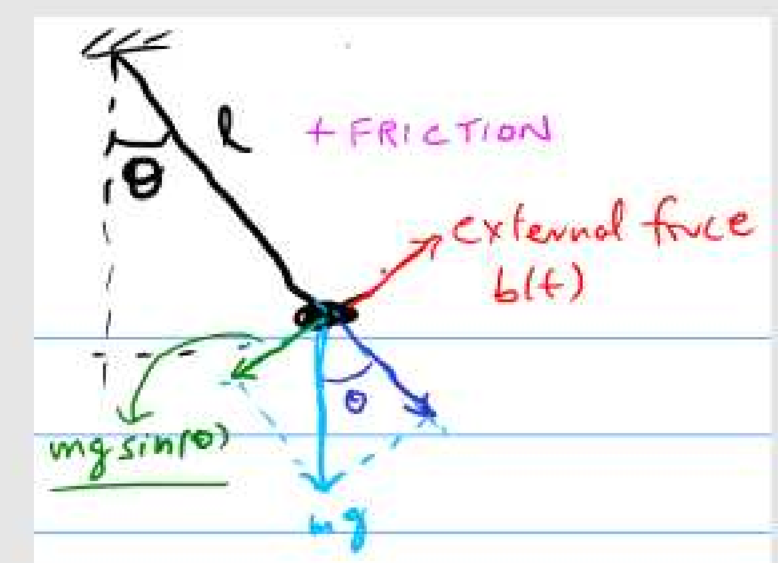


Pendulum: Inverted Solution

- Pendulum: $\frac{d\vec{x}}{dt} = \begin{bmatrix} v_\theta \\ -\frac{g}{l}\sin(\theta) - \frac{k}{m} v_\theta + \frac{b(t)}{ml} \end{bmatrix}$ $\quad \vec{x} = \begin{bmatrix} \theta \\ v_\theta \end{bmatrix}, \; \vec{u} = [b(t)]$

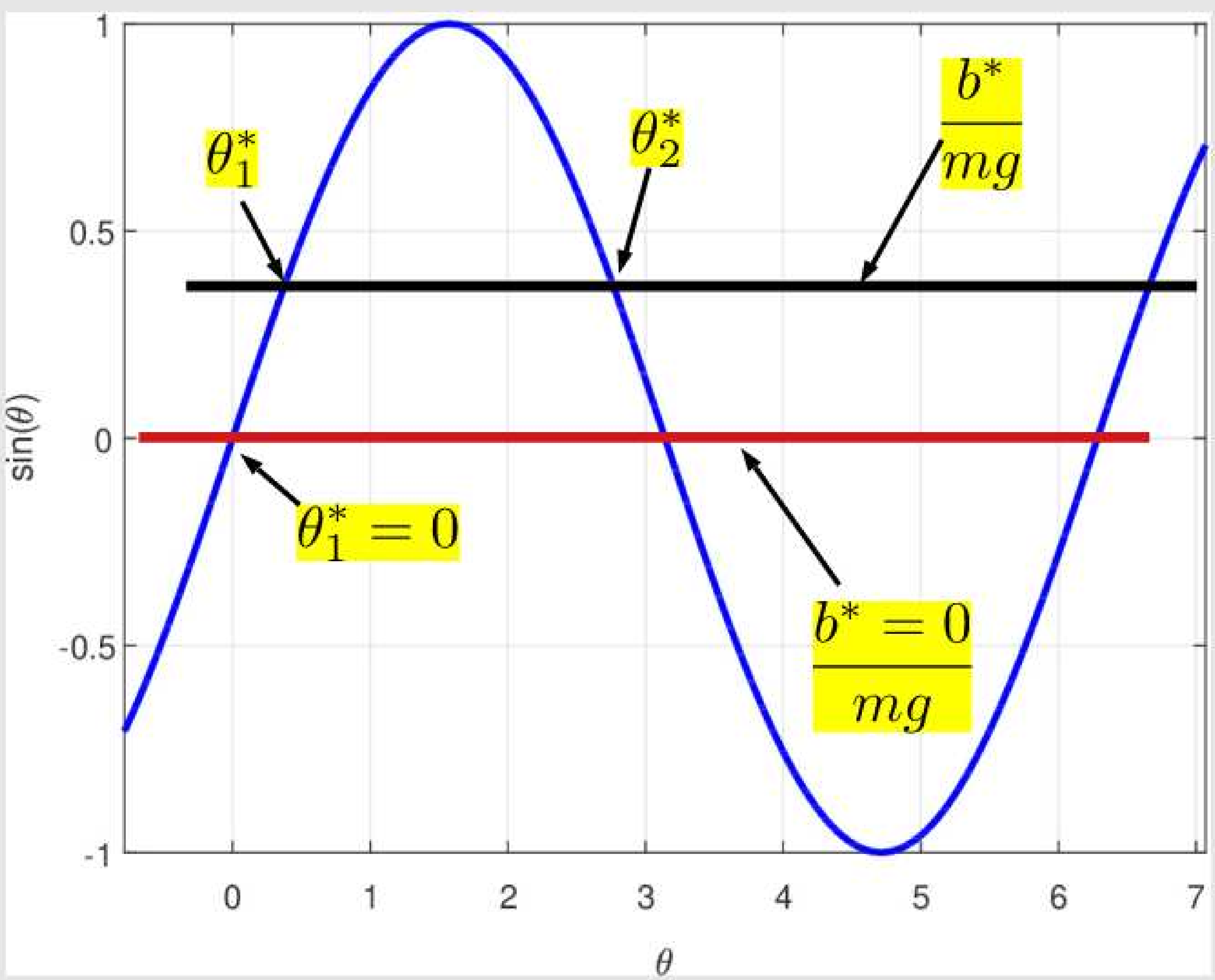
- DC input: $b(t) = b^*$
- DC solution: $dx/dt = 0 \longrightarrow v_\theta = 0$, $\frac{g}{l}\sin(\theta) = \frac{b^*}{ml} \Rightarrow \sin(\theta) = \frac{b^*}{mg}$

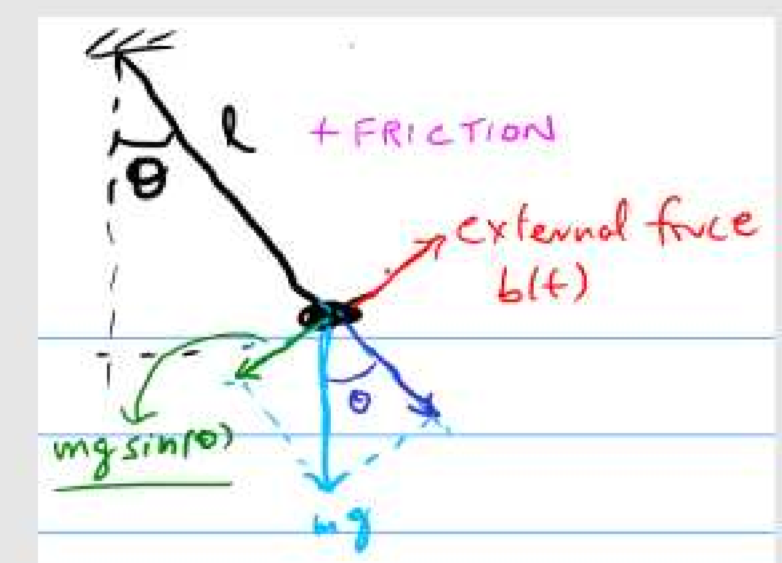




Pendulum: Inverted Solution

- **Pendulum:** $$\frac{d\vec{x}}{dt} = \begin{bmatrix} v_\theta \\ -g/l \sin(\theta) - k/m\, v_\theta + \frac{b(t)}{ml} \end{bmatrix}$$ $\vec{x} = \begin{bmatrix} \theta \\ v_\theta \end{bmatrix}$, $\vec{u} = [b(t)]$
- DC input: b(t) = b*
- DC solution: dx/dt = 0 $\longrightarrow$ $v_\theta = 0$, $\frac{g}{l}\sin(\theta) = \frac{b^*}{ml}$ $\Rightarrow \sin(\theta) = \frac{b^*}{mg}$

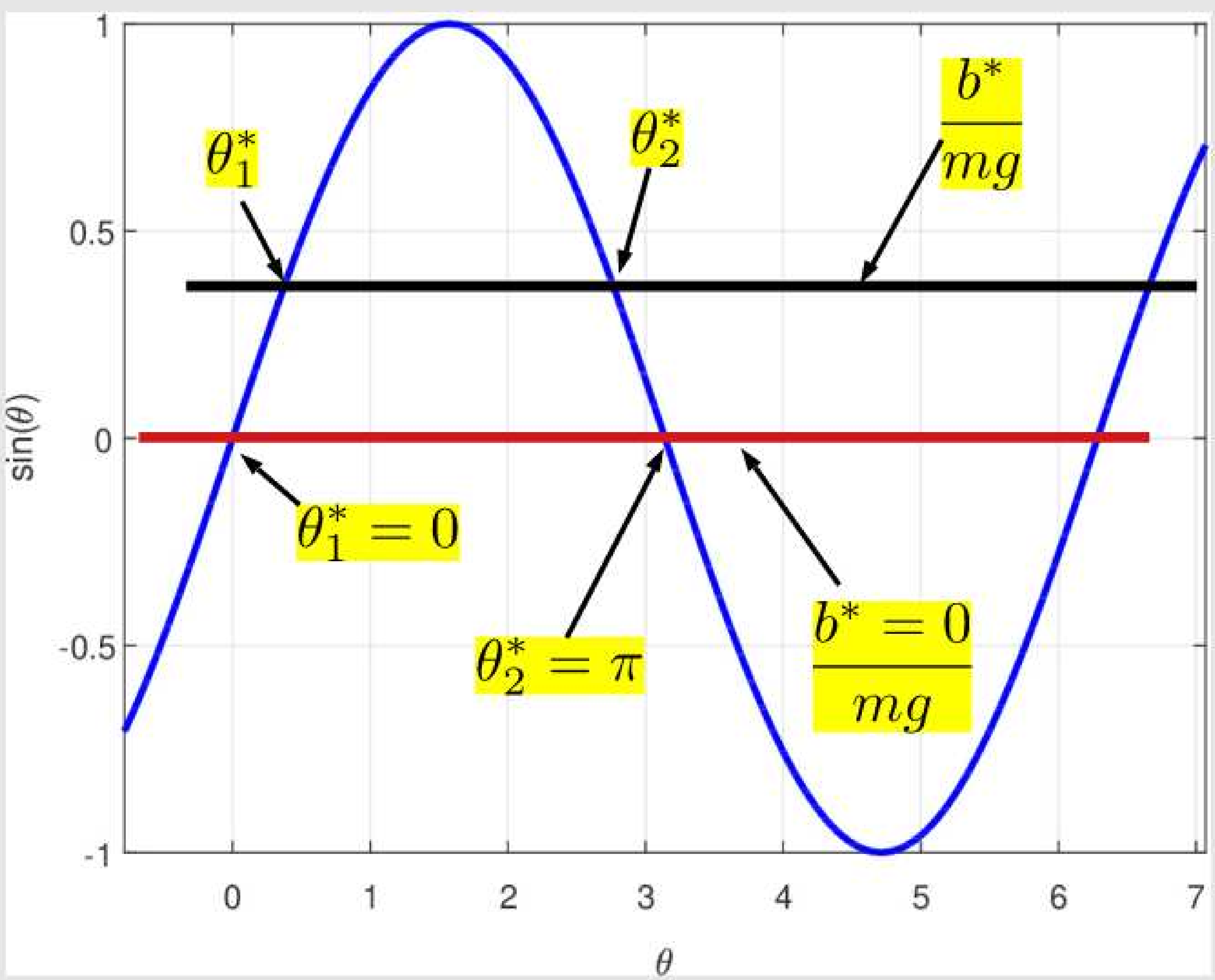


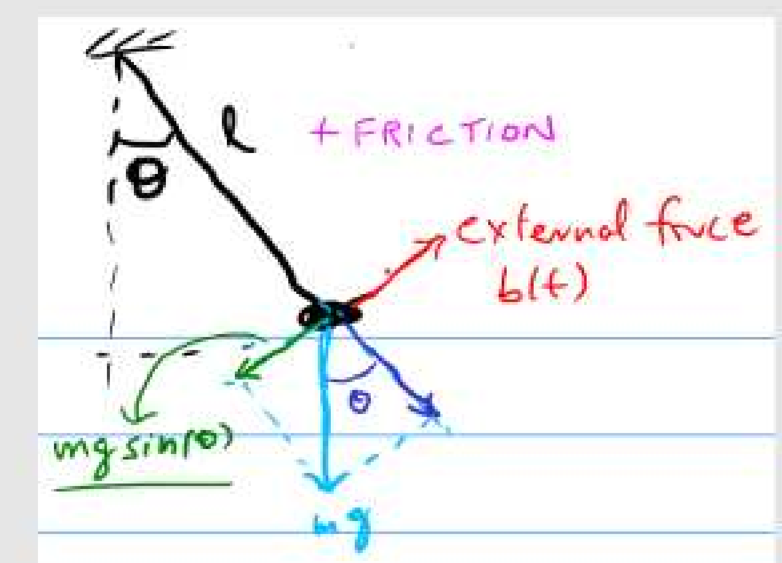


Pendulum: Inverted Solution

- Pendulum: $\frac{d\vec{x}}{dt} = \begin{bmatrix} v_\theta \\ -g/l \sin(\theta) - k/m\, v_\theta + \frac{b(t)}{ml} \end{bmatrix}$ $\qquad \vec{x} = \begin{bmatrix} \theta \\ v_\theta \end{bmatrix}, \quad \vec{u} = [b(t)]$

- DC input: b(t) = b*
- DC solution: dx/dt = 0 $\longrightarrow$ $v_\theta = 0$, $\frac{g}{l}\sin(\theta) = \frac{b^*}{ml}$ $\Rightarrow \sin(\theta) = \frac{b^*}{mg}$

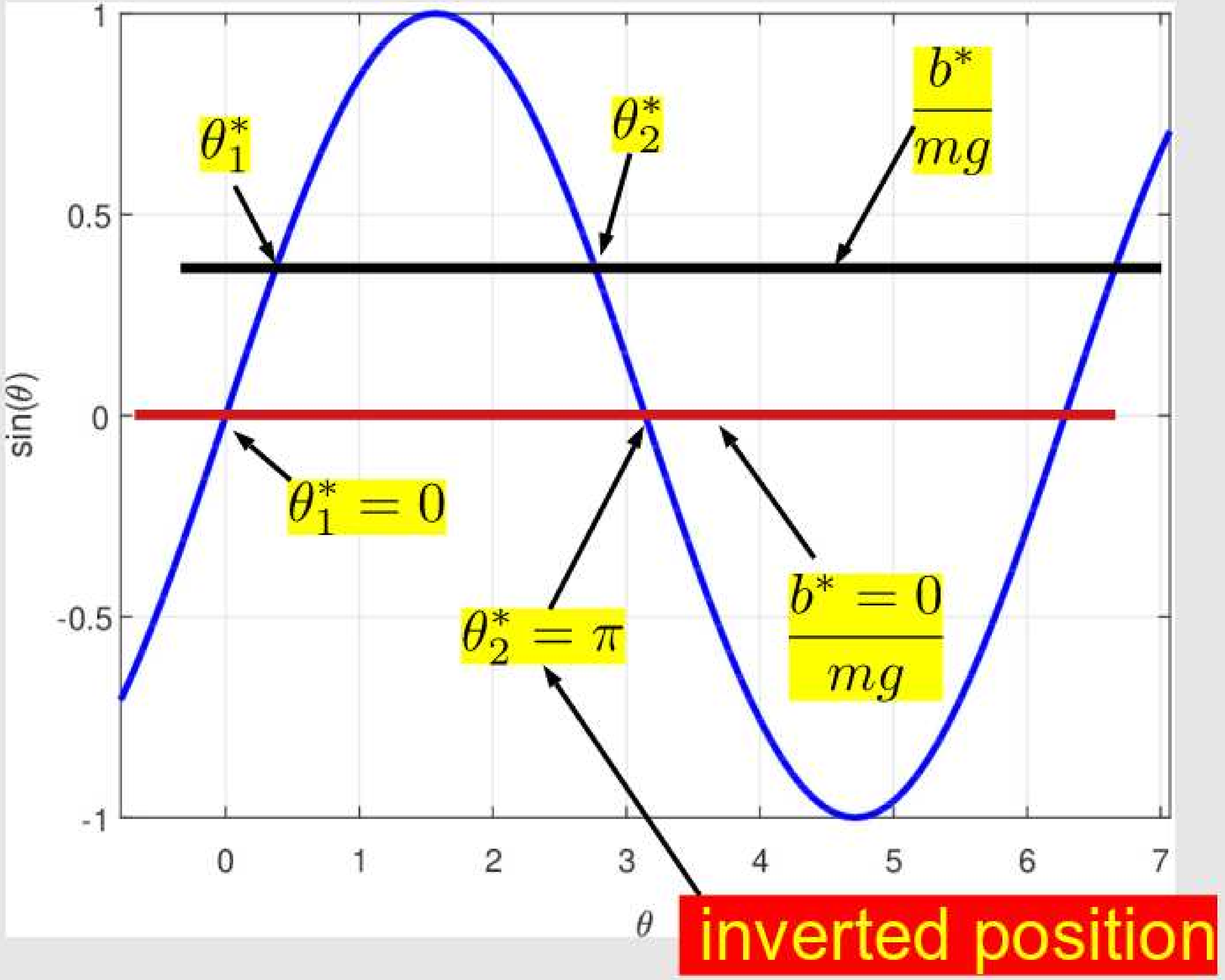


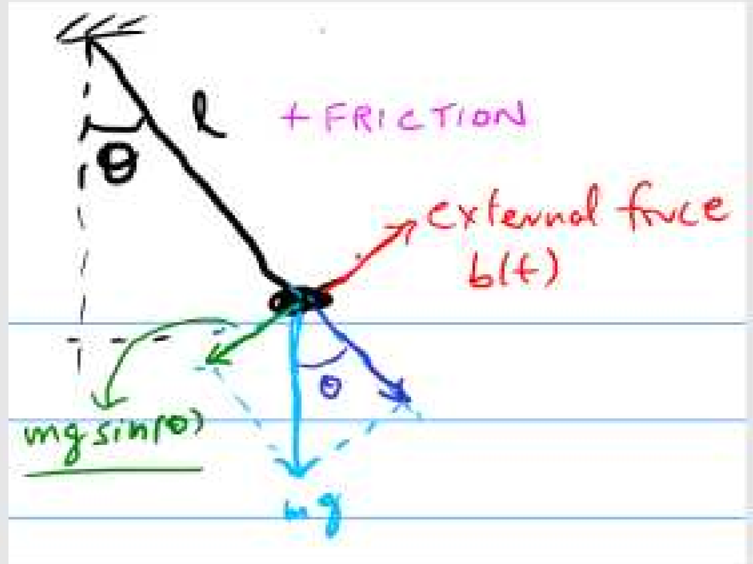


Pendulum: Inverted Solution

- Pendulum: $\frac{d\vec{x}}{dt} = \begin{bmatrix} v_\theta \\ -g/l \sin(\theta) - k/m\, v_\theta + \frac{b(t)}{ml} \end{bmatrix}$ $\vec{x} = \begin{bmatrix} \theta \\ v_\theta \end{bmatrix}$, $\vec{u} = [b(t)]$

- DC input: b(t) = b*
- DC solution: dx/dt = 0 $\longrightarrow$ $v_\theta = 0$, $\frac{g}{l}\sin(\theta) = \frac{b^*}{ml} \Rightarrow \sin(\theta) = \frac{b^*}{mg}$

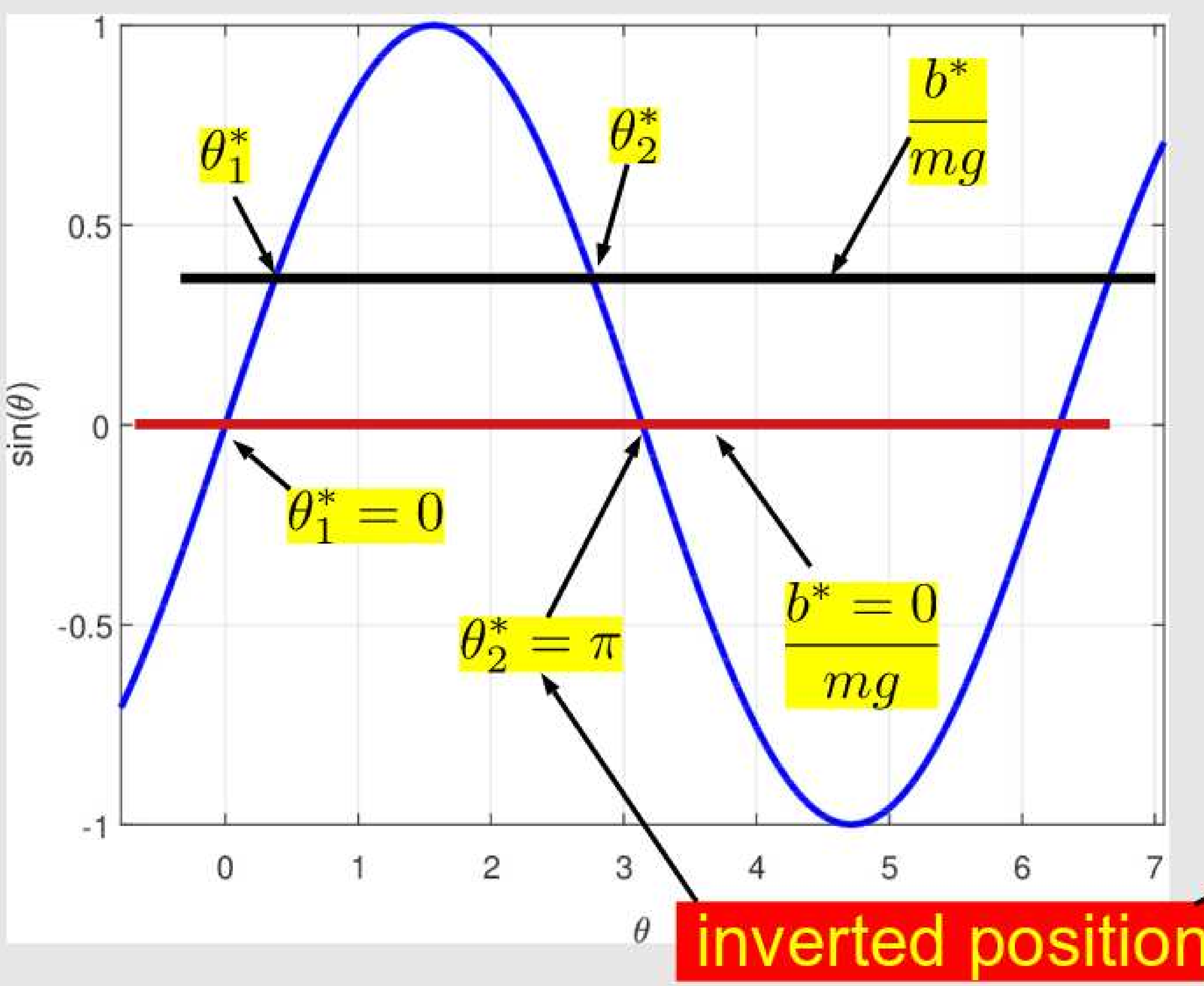


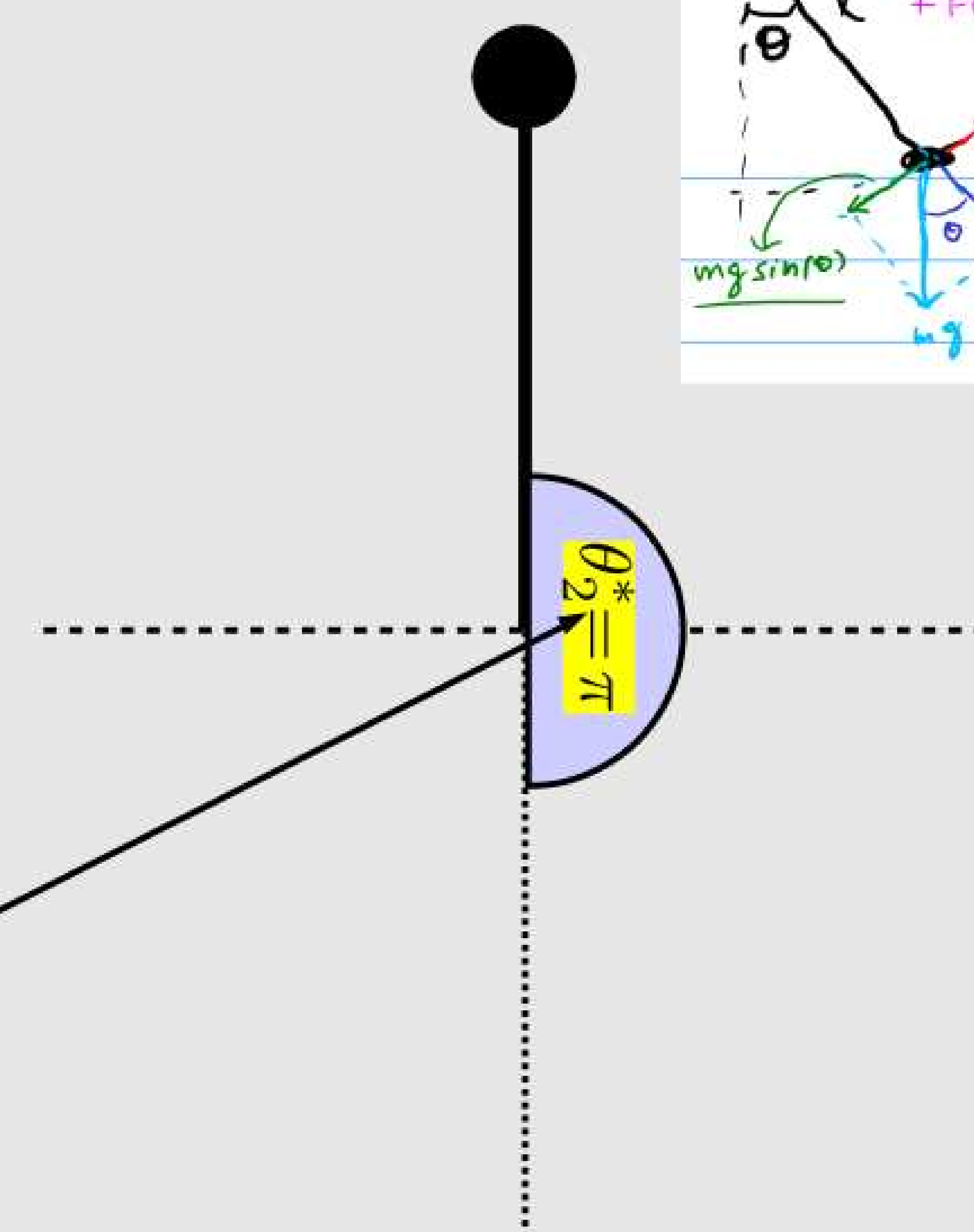


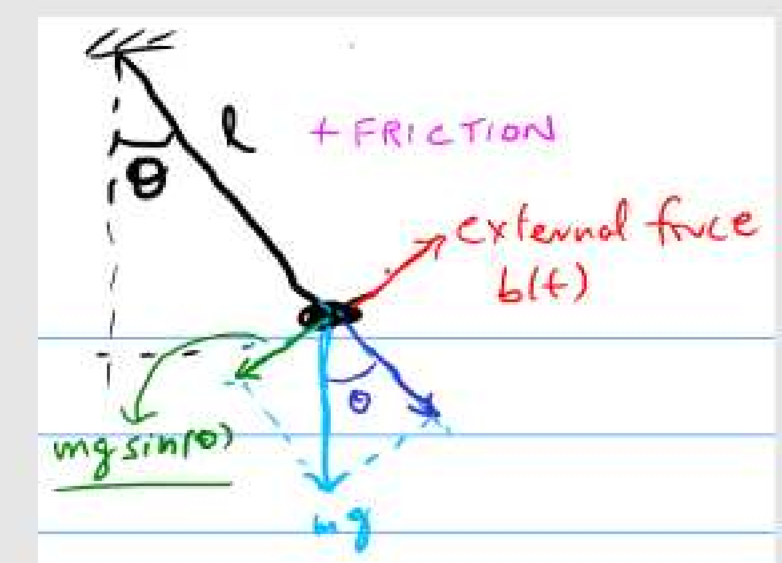
Pendulum: Inverted Solution

- Pendulum: $\dfrac{d\vec{x}}{dt} = \begin{bmatrix} v_\theta \\ -g/l \sin(\theta) - k/m\, v_\theta + \dfrac{b(t)}{ml} \end{bmatrix}$ $\vec{x} = \begin{bmatrix} \theta \\ v_\theta \end{bmatrix}$, $\vec{u} = [b(t)]$

- DC input: b(t) = b*
- DC solution: dx/dt = 0 $\longrightarrow$ $v_\theta = 0$, $\dfrac{g}{l}\sin(\theta) = \dfrac{b^*}{ml}$ $\Rightarrow \sin(\theta) = \dfrac{b^*}{mg}$







Inverted Pendulum: Linearization

- $$\frac{d\vec{x}}{dt} = \begin{bmatrix} v_\theta \\ -g/l \sin(\theta) - k/m\, v_\theta + \frac{b(t)}{ml} \end{bmatrix}$$ $$\vec{x} = \begin{bmatrix} \theta \\ v_\theta \end{bmatrix}, \quad \vec{u} = [b(t)]$$

Inverted Pendulum: Linearization

- $$\frac{d\vec{x}}{dt} = \begin{bmatrix} v_\theta \\ -g/l \sin(\theta) - k/m\, v_\theta + \frac{b(t)}{ml} \end{bmatrix}$$ $$\vec{x} = \begin{bmatrix} \theta \\ v_\theta \end{bmatrix}, \quad \vec{u} = [b(t)]$$

- $$J_x(\vec{x}_2^*, 0) = \begin{bmatrix} 0 & 1 \\ -\frac{g}{l}\cos(\theta_2^*) & -\frac{k}{m} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ +\frac{g}{l} & -\frac{k}{m} \end{bmatrix}$$

Inverted Pendulum: Linearization

- $$\frac{d\vec{x}}{dt} = \begin{bmatrix} v_\theta \\ -g/l \sin(\theta) - k/m\, v_\theta + \frac{b(t)}{ml} \end{bmatrix}$$ $\vec{x} = \begin{bmatrix} \theta \\ v_\theta \end{bmatrix}$, $\vec{u} = [b(t)]$

non-inverted

$$\boxed{\frac{d\vec{x}}{dt} = \begin{bmatrix} 0 & 1 \\ -g/l & -k/m \end{bmatrix} \vec{x} + \begin{bmatrix} 0 \\ +\frac{u(t)}{ml} \end{bmatrix}}$$

- $$J_x(\vec{x}_2^*, 0) = \begin{bmatrix} 0 & 1 \\ -\frac{g}{l}\cos(\theta_2^*) & -\frac{k}{m} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ +\frac{g}{l} & -\frac{k}{m} \end{bmatrix}$$

Inverted Pendulum: Linearization

- $$\frac{d\vec{x}}{dt} = \begin{bmatrix} v_\theta \\ -g/l \sin(\theta) - k/m\, v_\theta + \frac{b(t)}{ml} \end{bmatrix}$$ $\vec{x} = \begin{bmatrix} \theta \\ v_\theta \end{bmatrix}$, $\vec{u} = [b(t)]$

non-inverted

$$\boxed{\frac{d\vec{x}}{dt} = \begin{bmatrix} 0 & 1 \\ -g/l & -k/m \end{bmatrix} \vec{x} + \begin{bmatrix} 0 \\ +\frac{u(t)}{ml} \end{bmatrix}}$$

- $$J_x(\vec{x}_2^*, 0) = \begin{bmatrix} 0 & 1 \\ -\frac{g}{l}\cos(\theta_2^*) & -\frac{k}{m} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ +\frac{g}{l} & -\frac{k}{m} \end{bmatrix}$$
- Linearization: $$\frac{d}{dt}\begin{bmatrix} \Delta\theta(t) \\ \Delta v_\theta(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ +\frac{g}{l} & -\frac{k}{m} \end{bmatrix} \begin{bmatrix} \Delta\theta(t) \\ \Delta v_\theta(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{ml} \end{bmatrix} \Delta b(t)$$

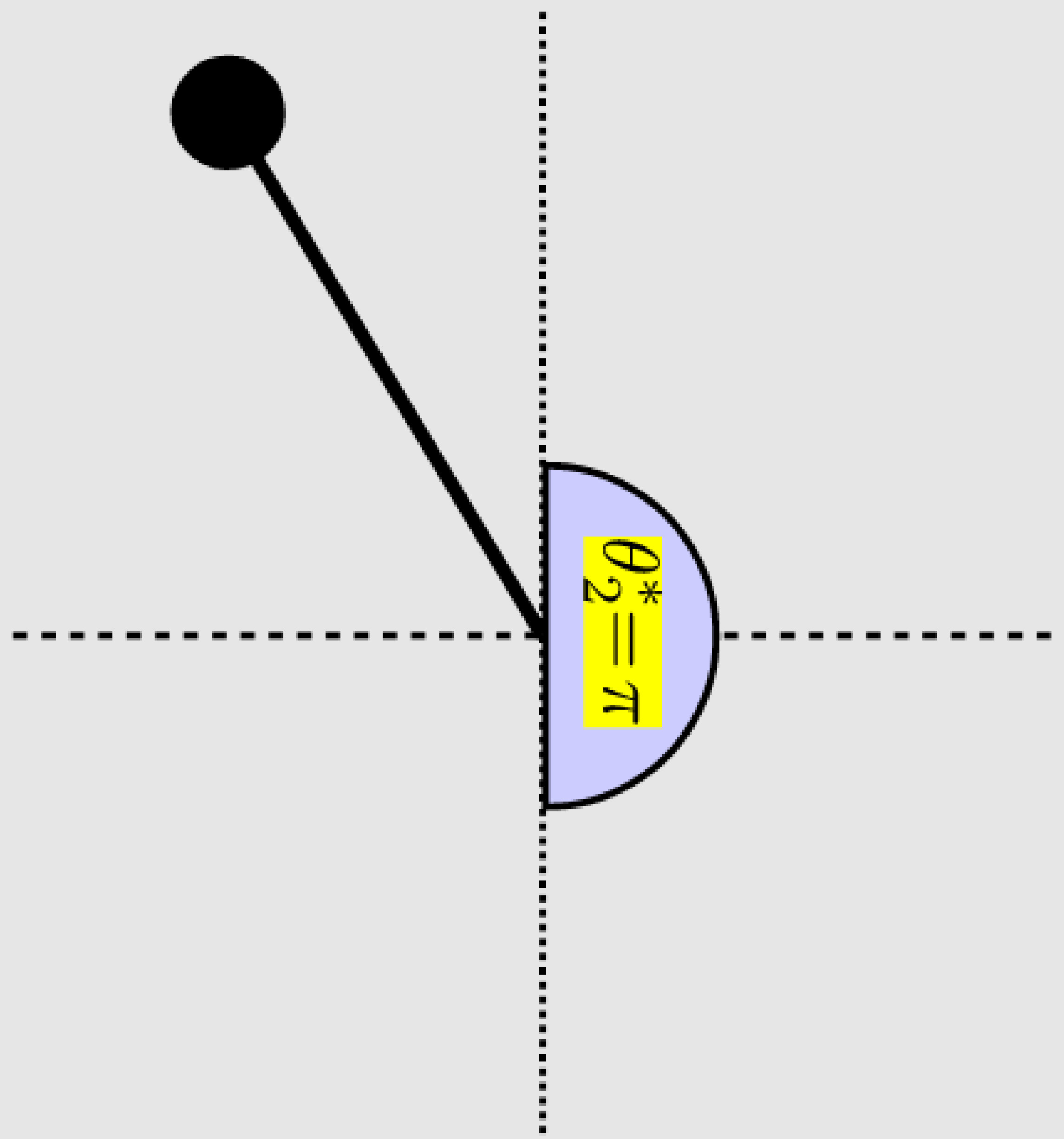
Inverted Pendulum: Linearization

- $$\frac{d\vec{x}}{dt} = \begin{bmatrix} v_\theta \\ -g/l \sin(\theta) - k/m\, v_\theta + \frac{b(t)}{ml} \end{bmatrix}$$ $\vec{x} = \begin{bmatrix} \theta \\ v_\theta \end{bmatrix}$, $\vec{u} = [b(t)]$

non-inverted

$$\boxed{\frac{d\vec{x}}{dt} = \begin{bmatrix} 0 & 1 \\ -g/l & -k/m \end{bmatrix} \vec{x} + \begin{bmatrix} 0 \\ +\frac{u(t)}{ml} \end{bmatrix}}$$

- $$J_x(\vec{x}_2^*, 0) = \begin{bmatrix} 0 & 1 \\ -\frac{g}{l}\cos(\theta_2^*) & -\frac{k}{m} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ +\frac{g}{l} & -\frac{k}{m} \end{bmatrix}$$
- Linearization: $$\frac{d}{dt}\begin{bmatrix} \Delta\theta(t) \\ \Delta v_\theta(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ +\frac{g}{l} & -\frac{k}{m} \end{bmatrix} \begin{bmatrix} \Delta\theta(t) \\ \Delta v_\theta(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{ml} \end{bmatrix} \Delta b(t)$$



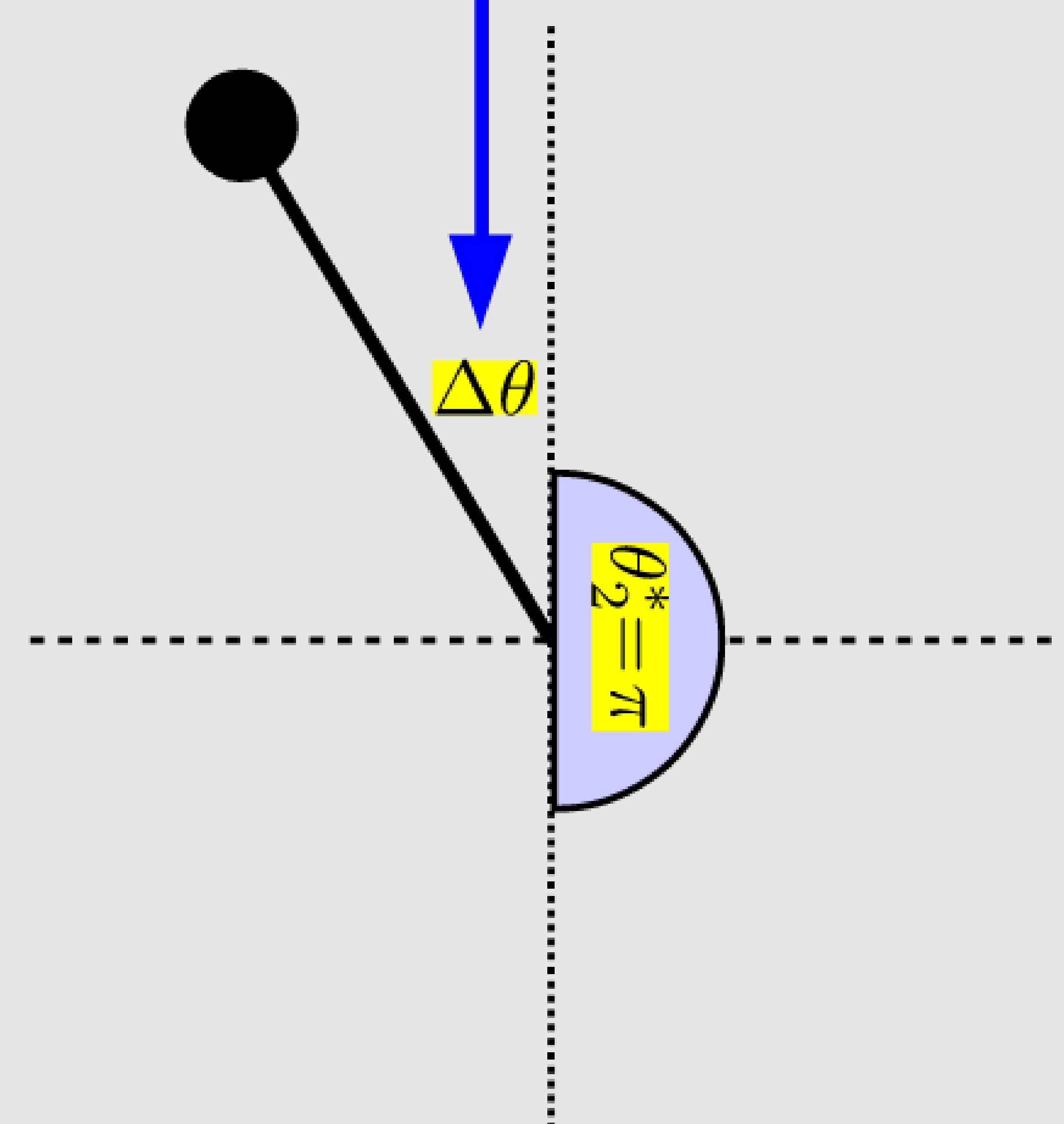
Inverted Pendulum: Linearization

- $$\frac{d\vec{x}}{dt} = \begin{bmatrix} v_\theta \\ -g/l \sin(\theta) - k/m\, v_\theta + \frac{b(t)}{ml} \end{bmatrix}$$ $$\vec{x} = \begin{bmatrix} \theta \\ v_\theta \end{bmatrix}, \quad \vec{u} = [b(t)]$$

non-inverted

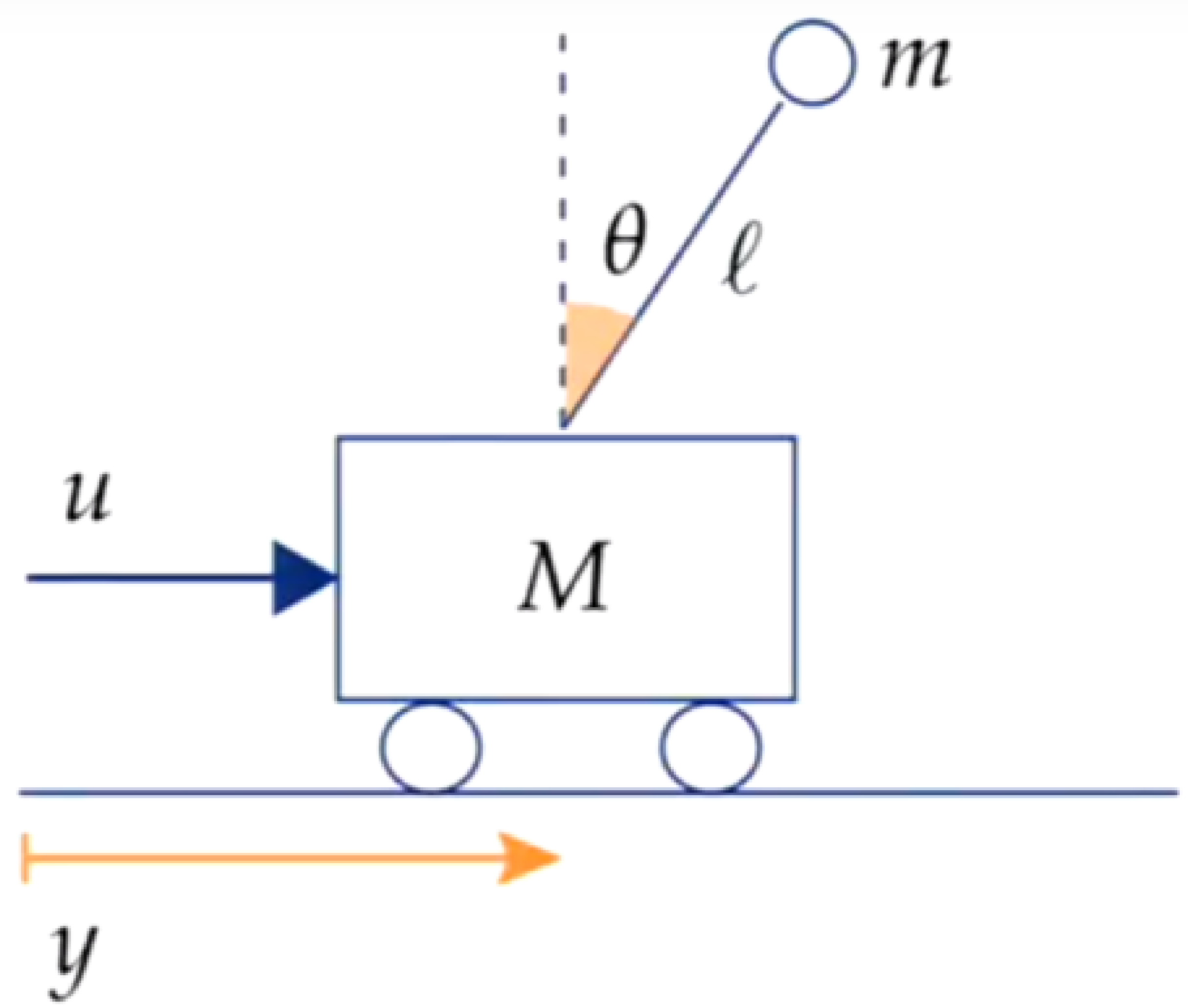
$$\frac{d\vec{x}}{dt} = \begin{bmatrix} 0 & 1 \\ -g/l & -k/m \end{bmatrix} \vec{x} + \begin{bmatrix} 0 \\ + \frac{u(t)}{ml} \end{bmatrix}$$

- $$J_x(\vec{x}_2^*, 0) = \begin{bmatrix} 0 & 1 \\ -\frac{g}{l}\cos(\theta_2^*) & -\frac{k}{m} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ +\frac{g}{l} & -\frac{k}{m} \end{bmatrix}$$
- Linearization: $$\frac{d}{dt}\begin{bmatrix} \Delta\theta(t) \\ \Delta v_\theta(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ +\frac{g}{l} & -\frac{k}{m} \end{bmatrix} \begin{bmatrix} \Delta\theta(t) \\ \Delta v_\theta(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{ml} \end{bmatrix} \Delta b(t)$$



Pole & Cart (Inverted Pendulum ++)

- Slightly more complicated example

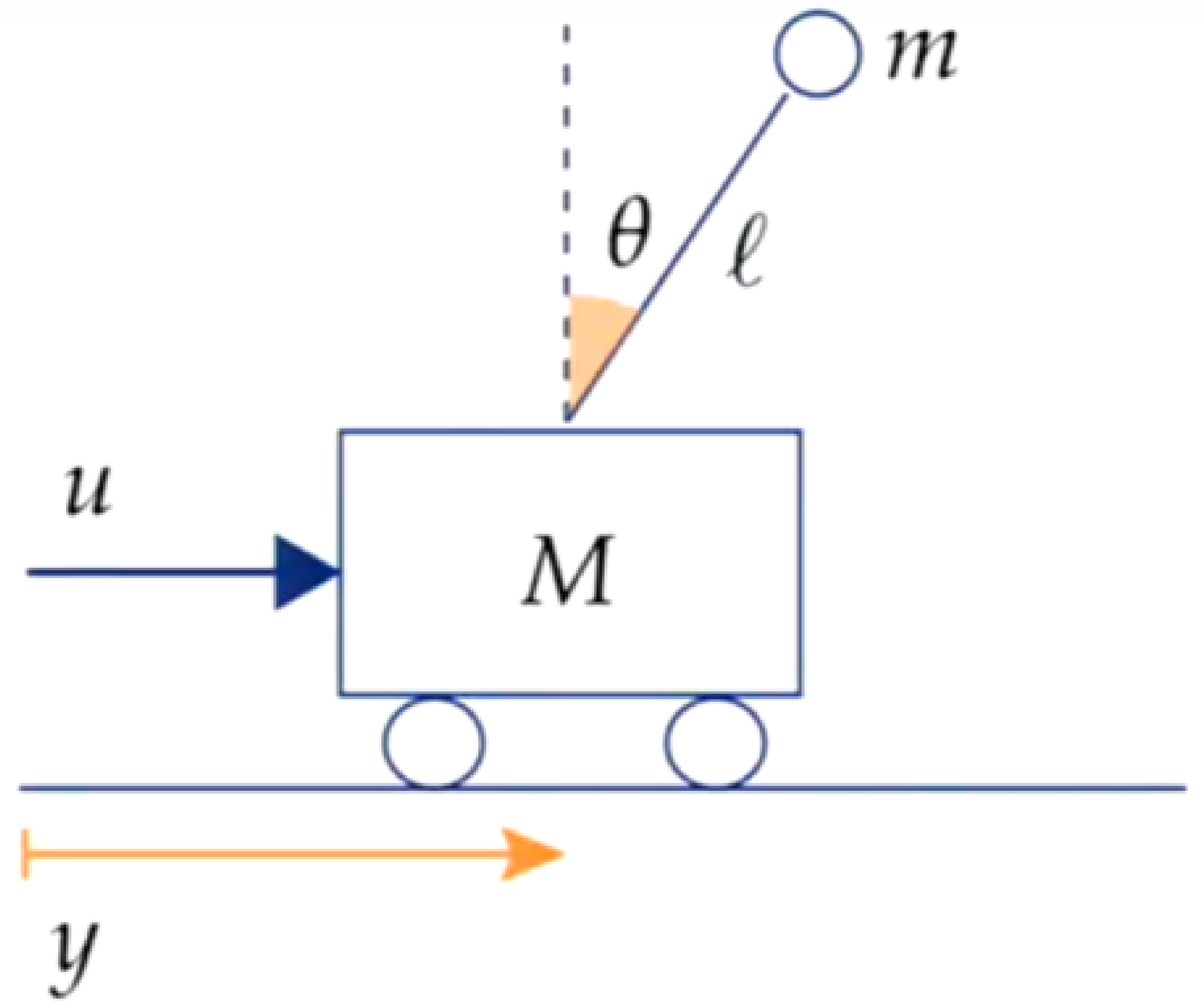


$$\ddot{y} = \frac{1}{\frac{M}{m} + \sin^2\theta}\left(\frac{u}{m} + \dot{\theta}^2 \ell \sin\theta - g\sin\theta\cos\theta\right)$$

$$\ddot{\theta} = \frac{1}{\ell(\frac{M}{m} + \sin^2\theta)}\left(-\frac{u}{m}\cos\theta - \dot{\theta}^2 \ell \cos\theta\sin\theta + \frac{M+m}{m} g\sin\theta\right)$$

Pole & Cart (Inverted Pendulum ++)

- Slightly more complicated example



$$\ddot{y} = \frac{1}{\frac{M}{m} + \sin^2\theta}\left(\frac{u}{m} + \dot{\theta}^2 \ell \sin\theta - g\sin\theta\cos\theta\right)$$

$$\ddot{\theta} = \frac{1}{\ell(\frac{M}{m} + \sin^2\theta)}\left(-\frac{u}{m}\cos\theta - \dot{\theta}^2\ell\cos\theta\sin\theta + \frac{M+m}{m} g\sin\theta\right)$$

- $\rightarrow$ discussion / HW

Stability

- Basic idea: **perturb** system **a little** from equilibrium

Stability

- Basic idea: **perturb** system **a little** from equilibrium
  - does it come back? yes → STABLE

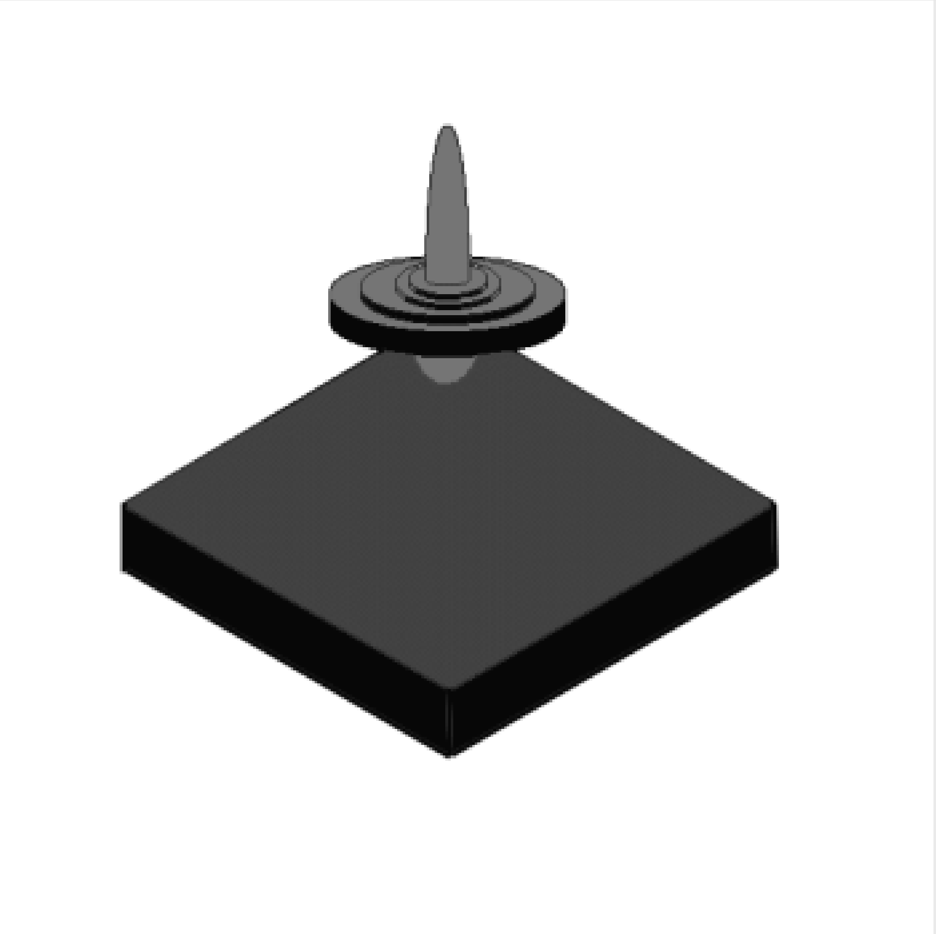
Stability

- Basic idea: **perturb** system **a little** from equilibrium
  - does it come back? yes → STABLE

- More precisely:
  - small perturbations → small responses

Stability

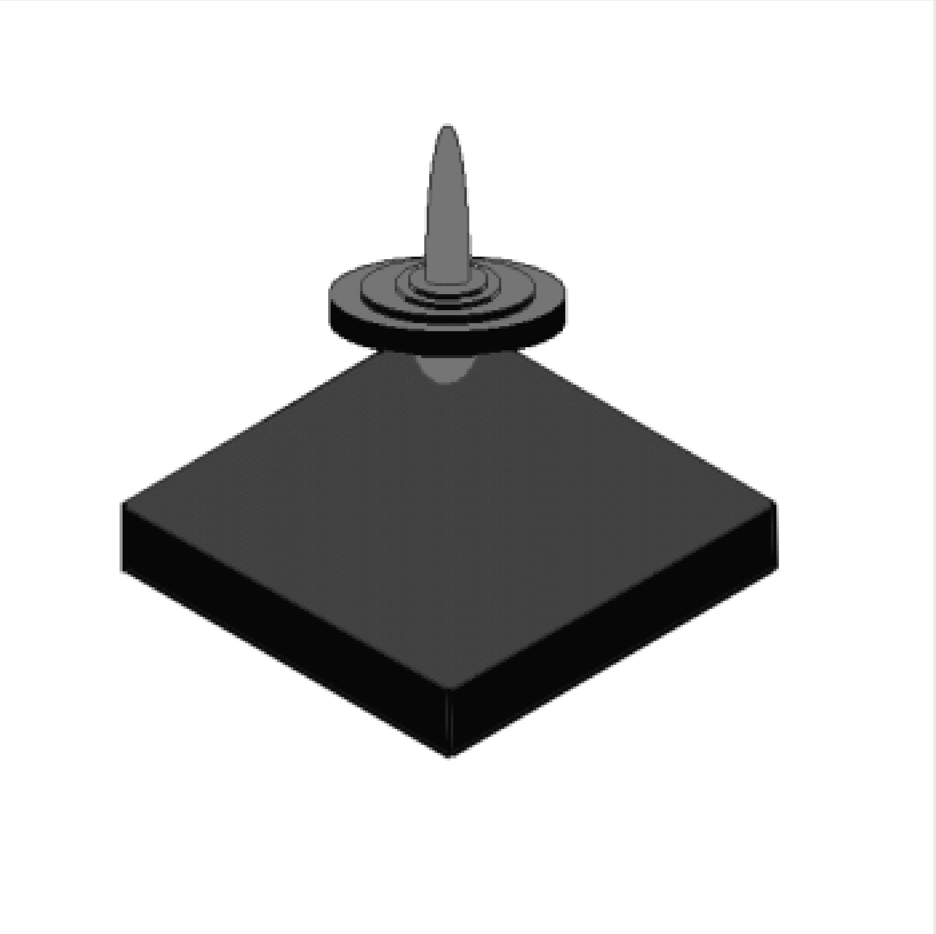
- Basic idea: **perturb** system **a little** from equilibrium
  - does it come back? yes → STABLE



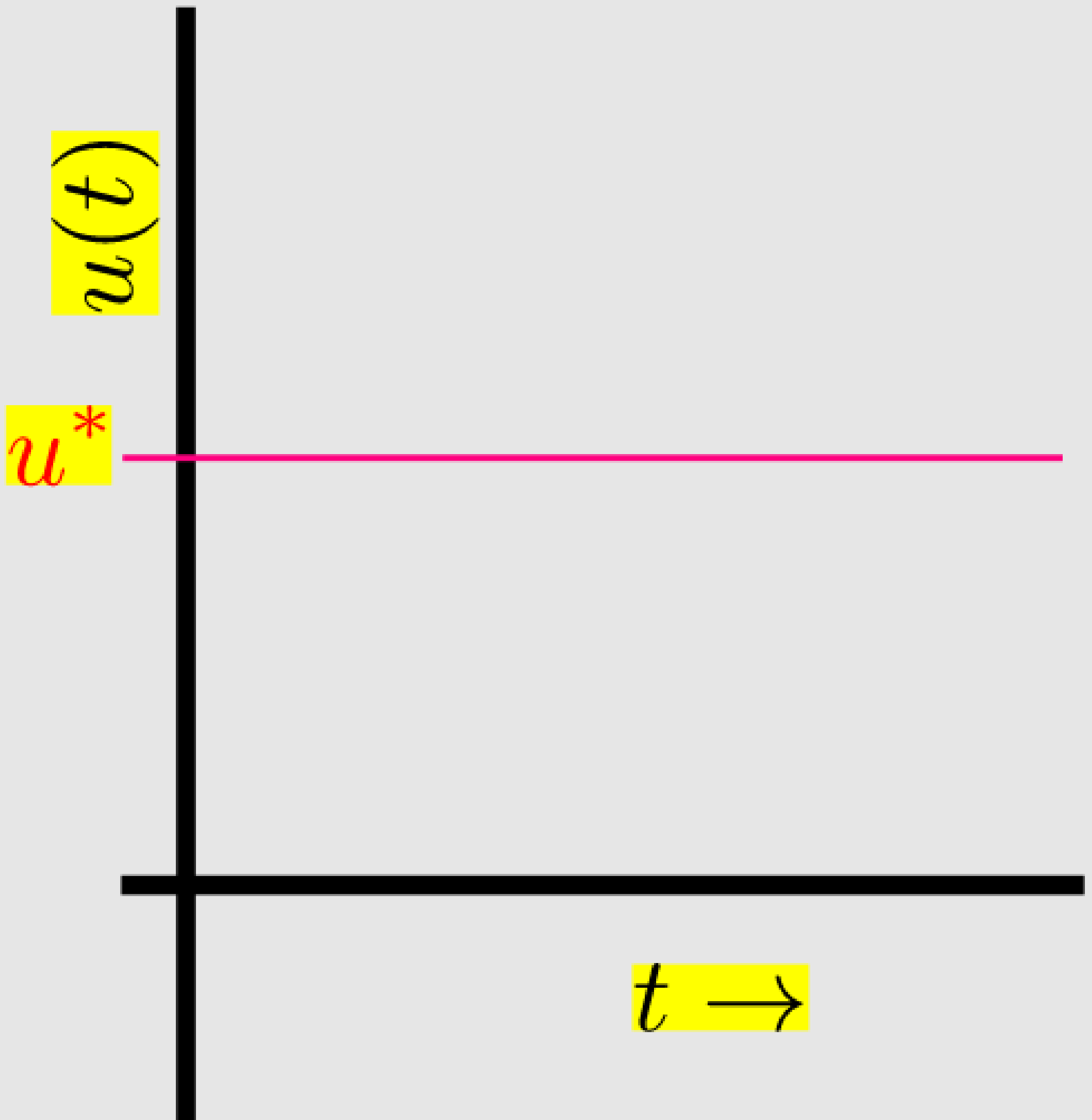
- More precisely:
  - small perturbations → small responses

Stability

- Basic idea: **perturb** system **a little** from equilibrium
  - does it come back? yes → STABLE

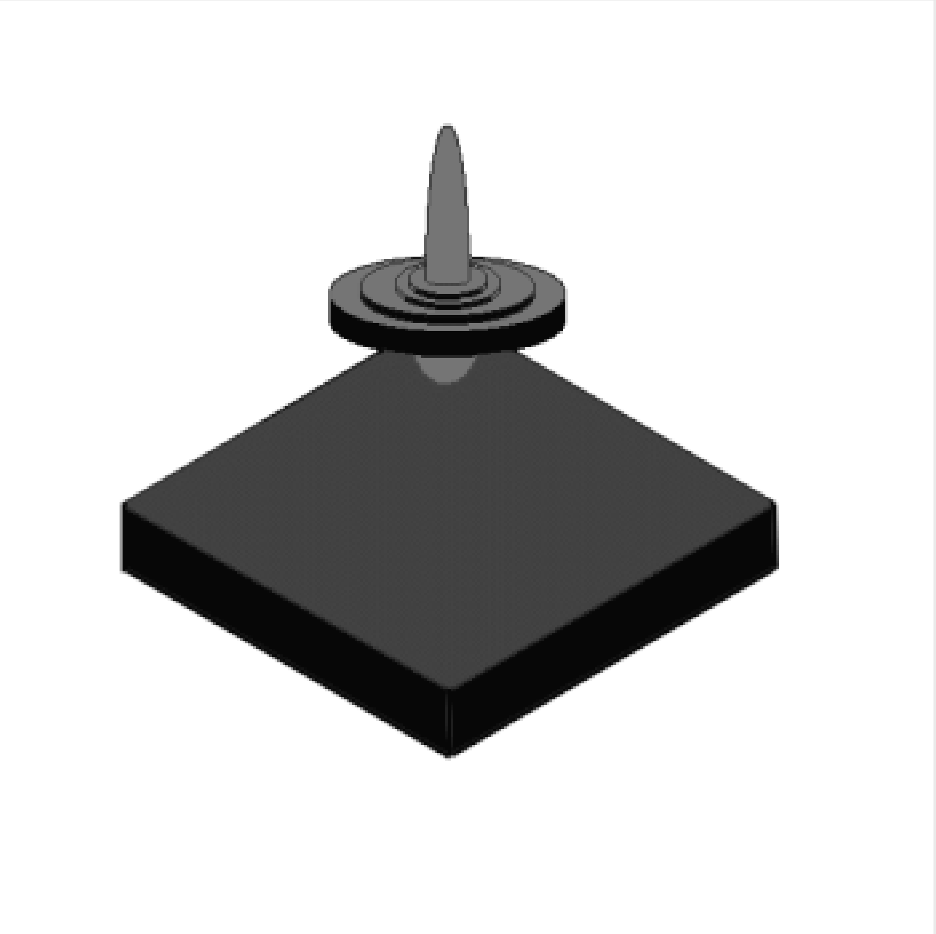


- More precisely:
  - small perturbations → small responses

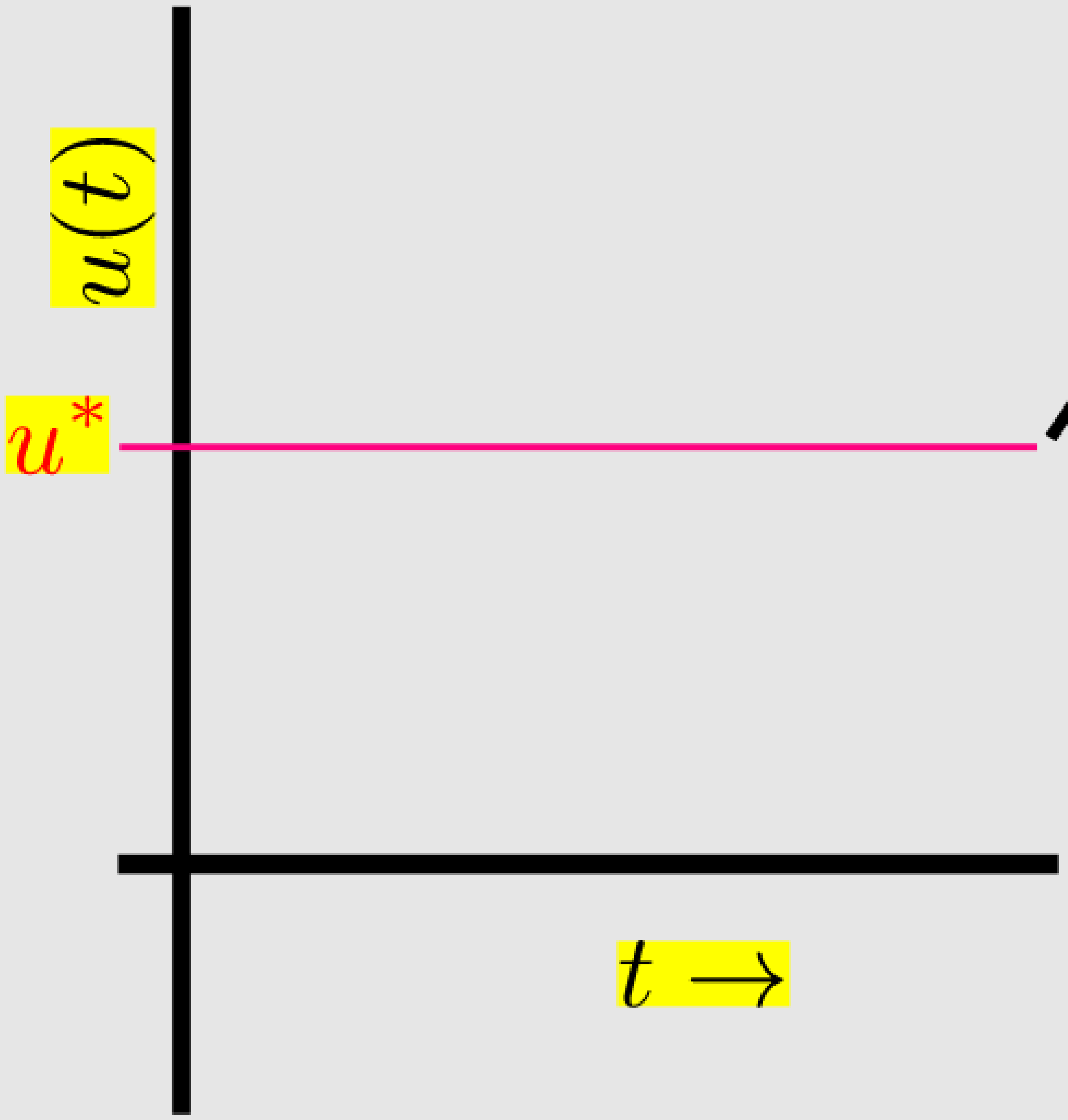


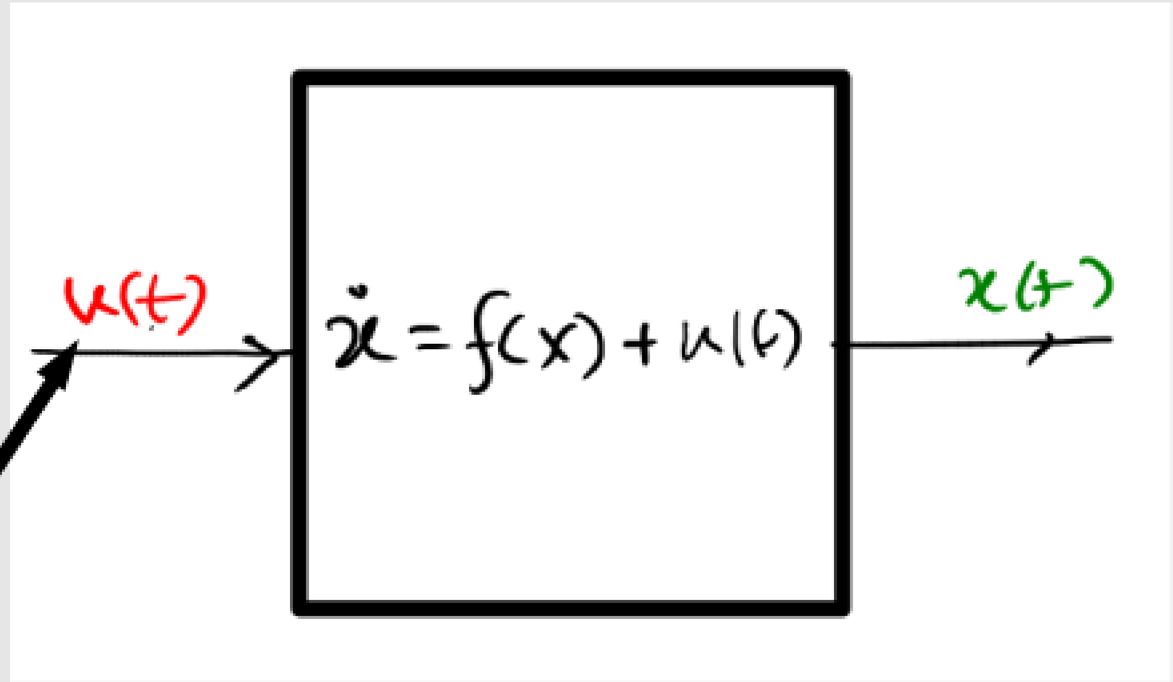
Stability

- Basic idea: **perturb** system **a little** from equilibrium
  - does it come back? yes → STABLE



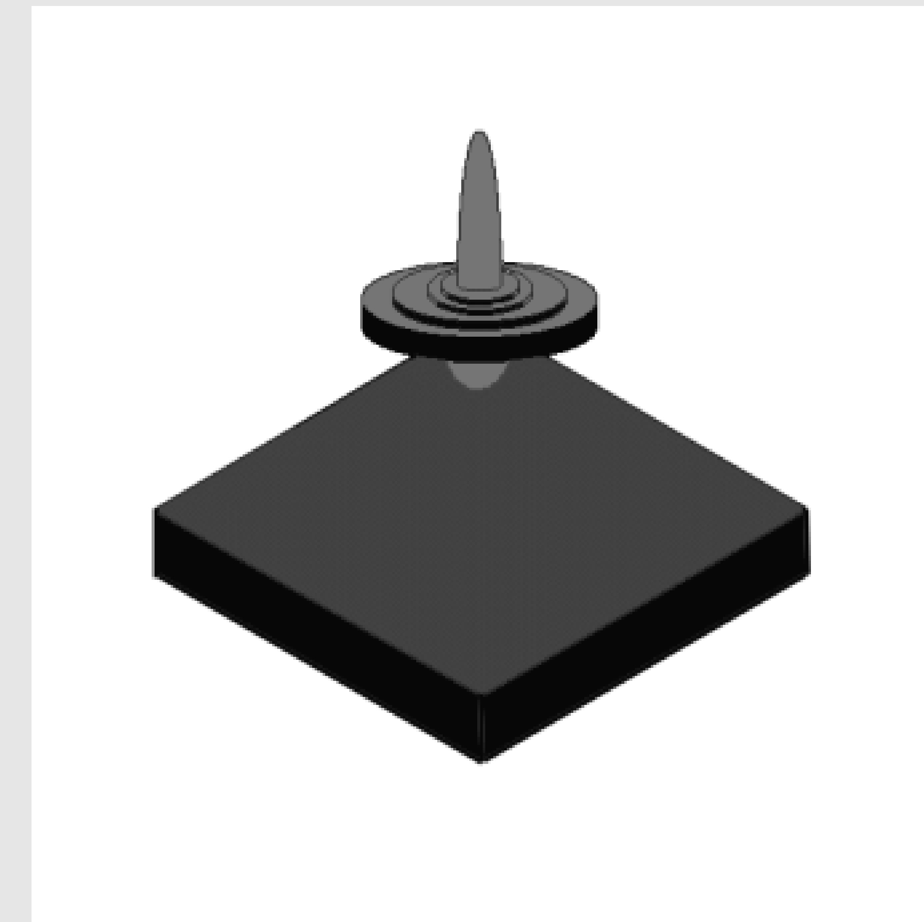
- More precisely:
  - small perturbations → small responses



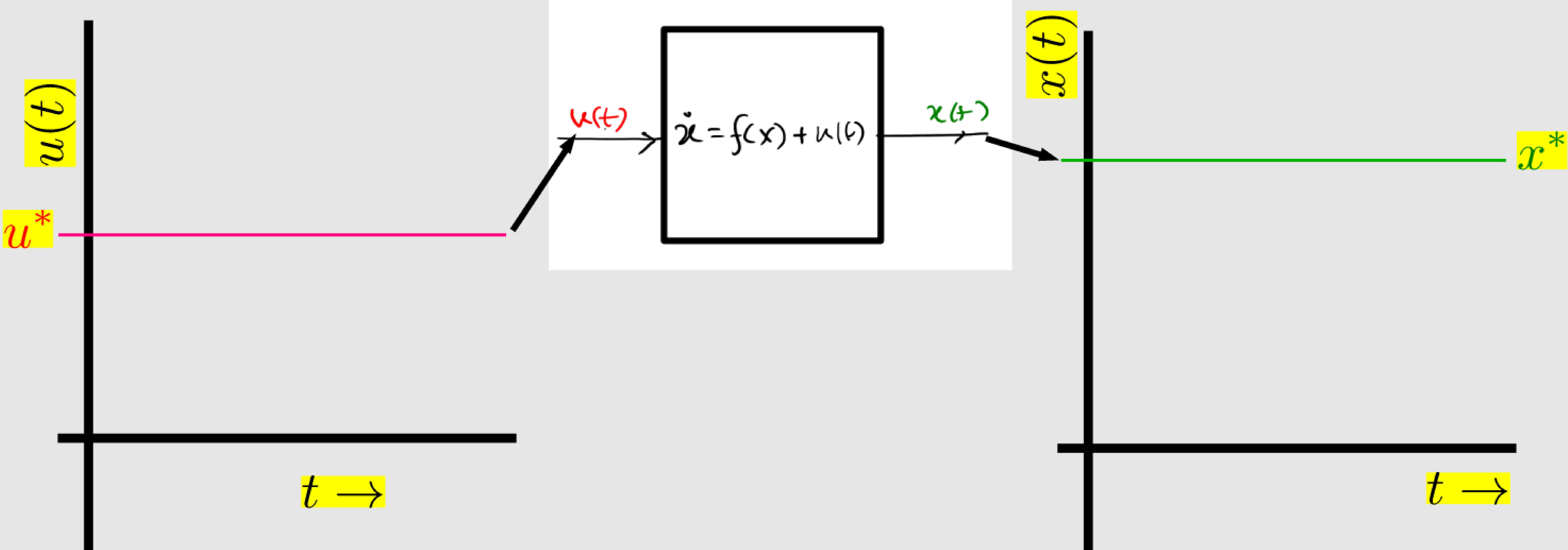


Stability

- Basic idea: **perturb** system **a little** from equilibrium
  - does it come back? yes → STABLE

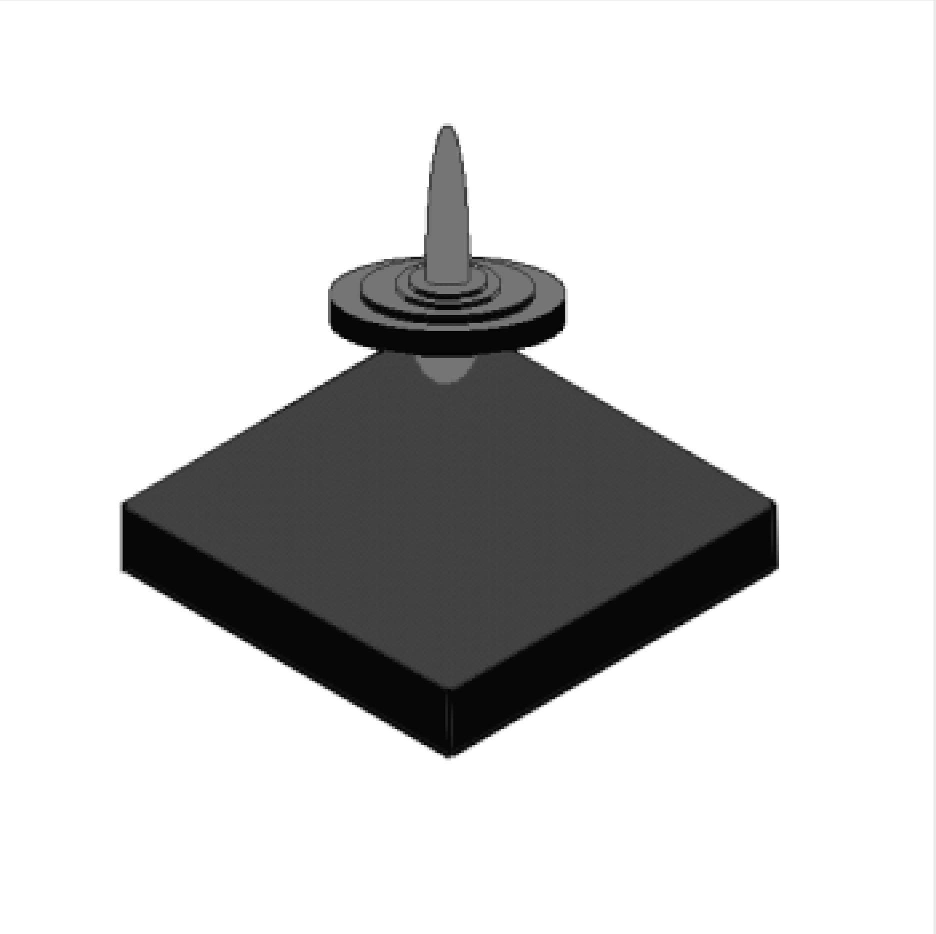


- More precisely:
  - small perturbations → small responses

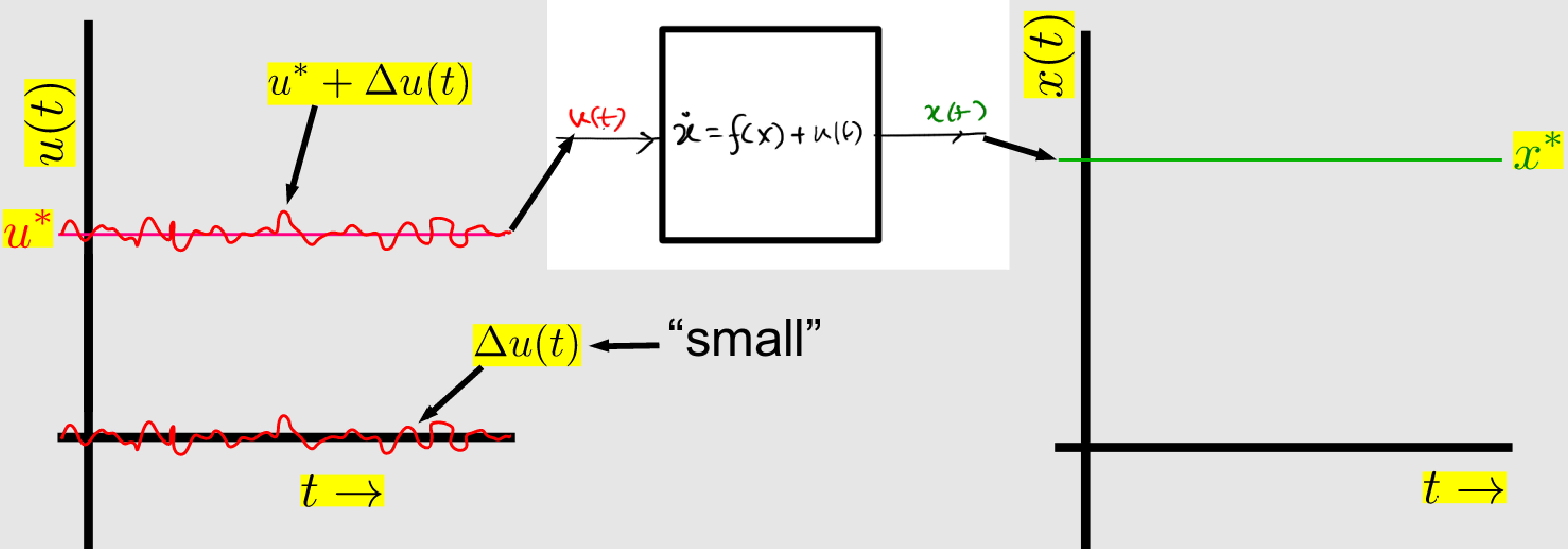


Stability

- Basic idea: **perturb** system **a little** from equilibrium
  - does it come back? yes → STABLE

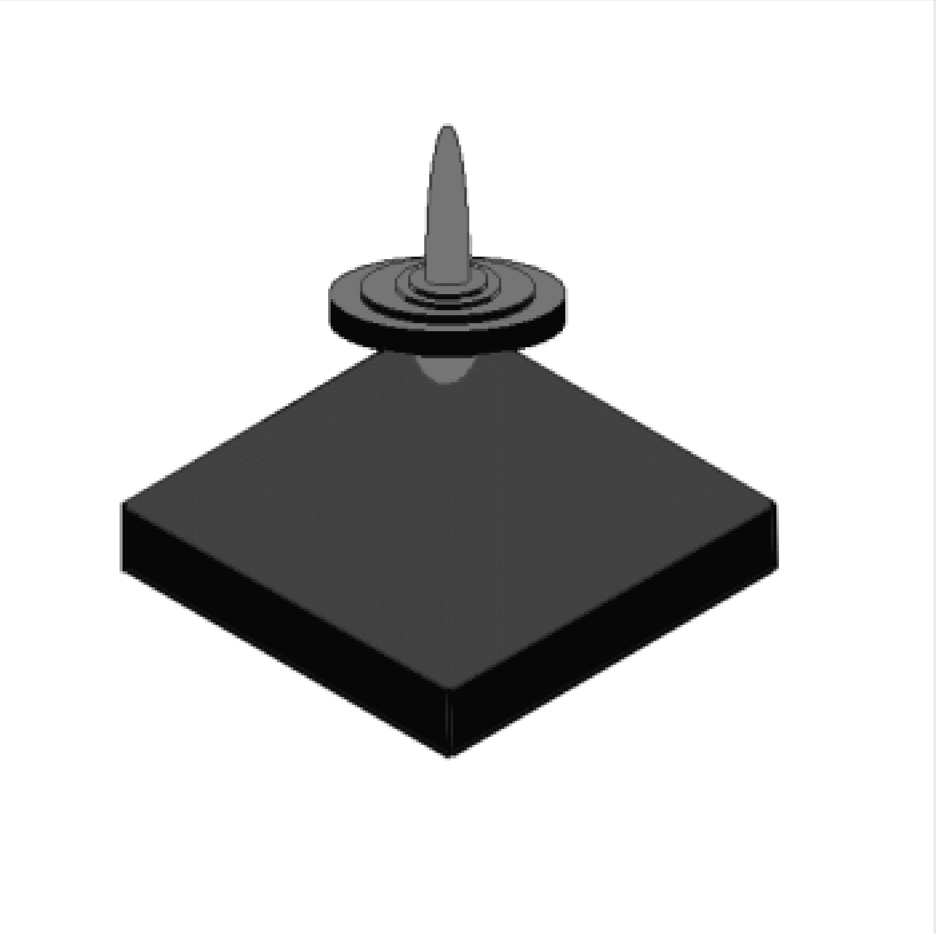


- More precisely:
  - small perturbations → small responses

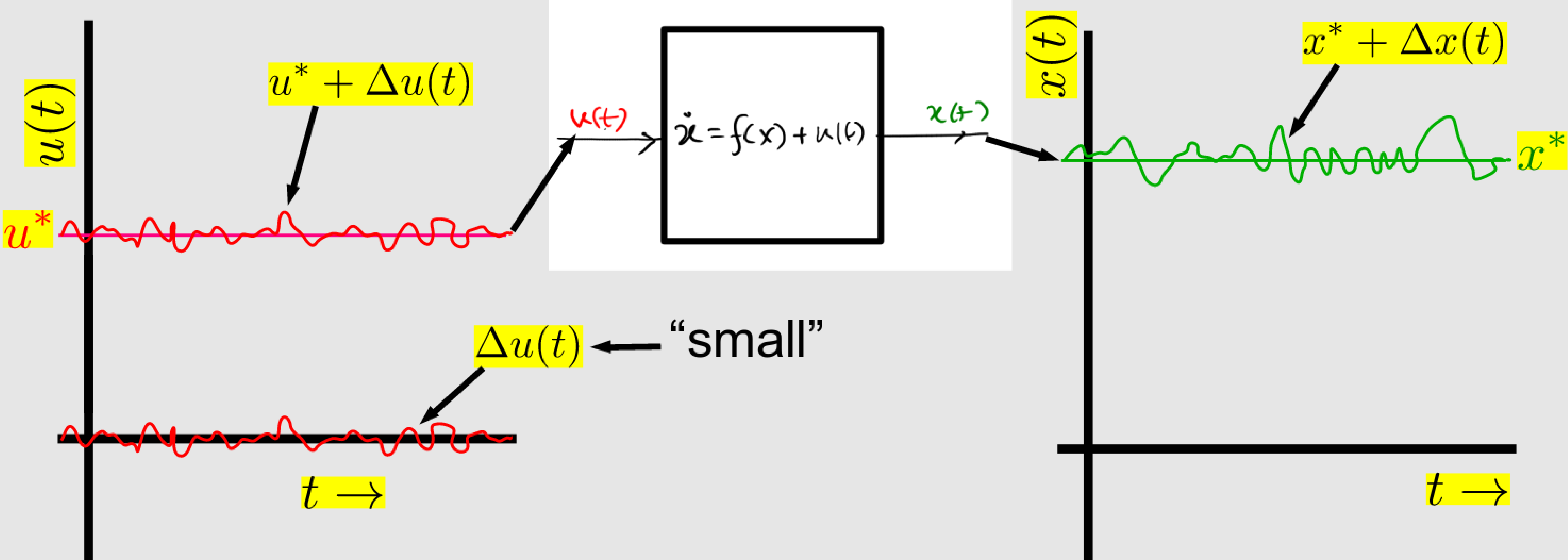


Stability

- Basic idea: **perturb** system **a little** from equilibrium
  - does it come back? yes → STABLE

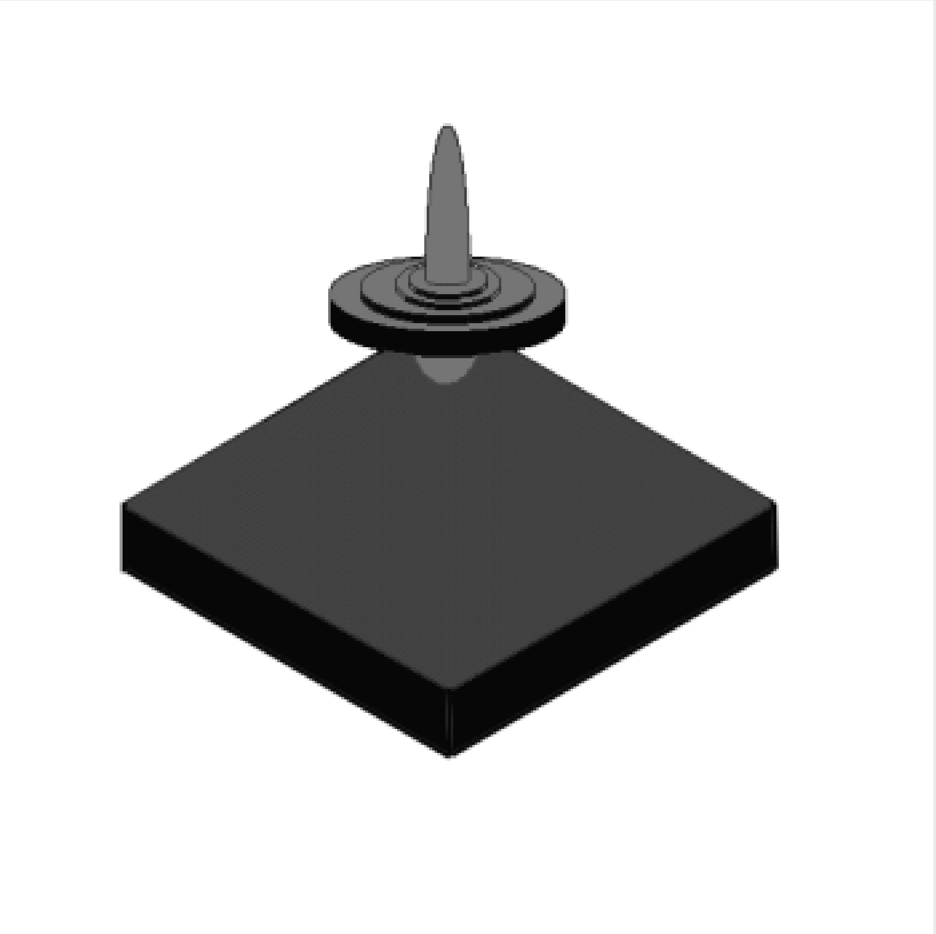


- More precisely:
  - small perturbations → small responses

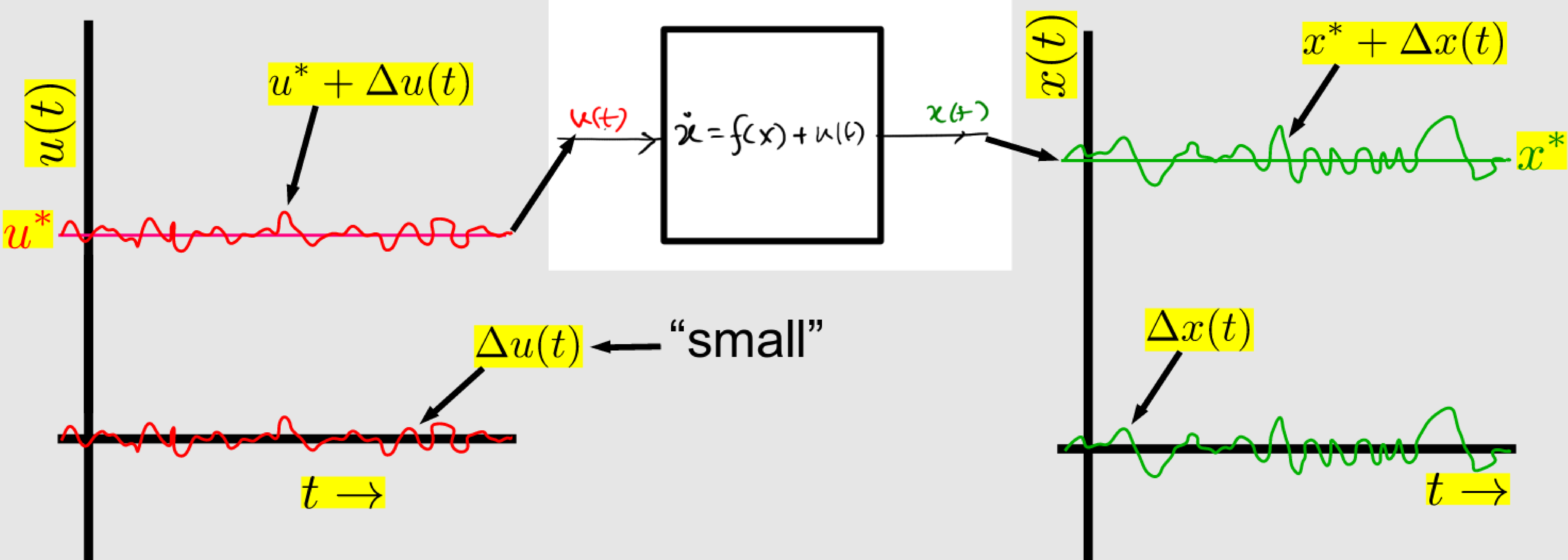


Stability

- Basic idea: **perturb** system **a little** from equilibrium
  - does it come back? yes → STABLE

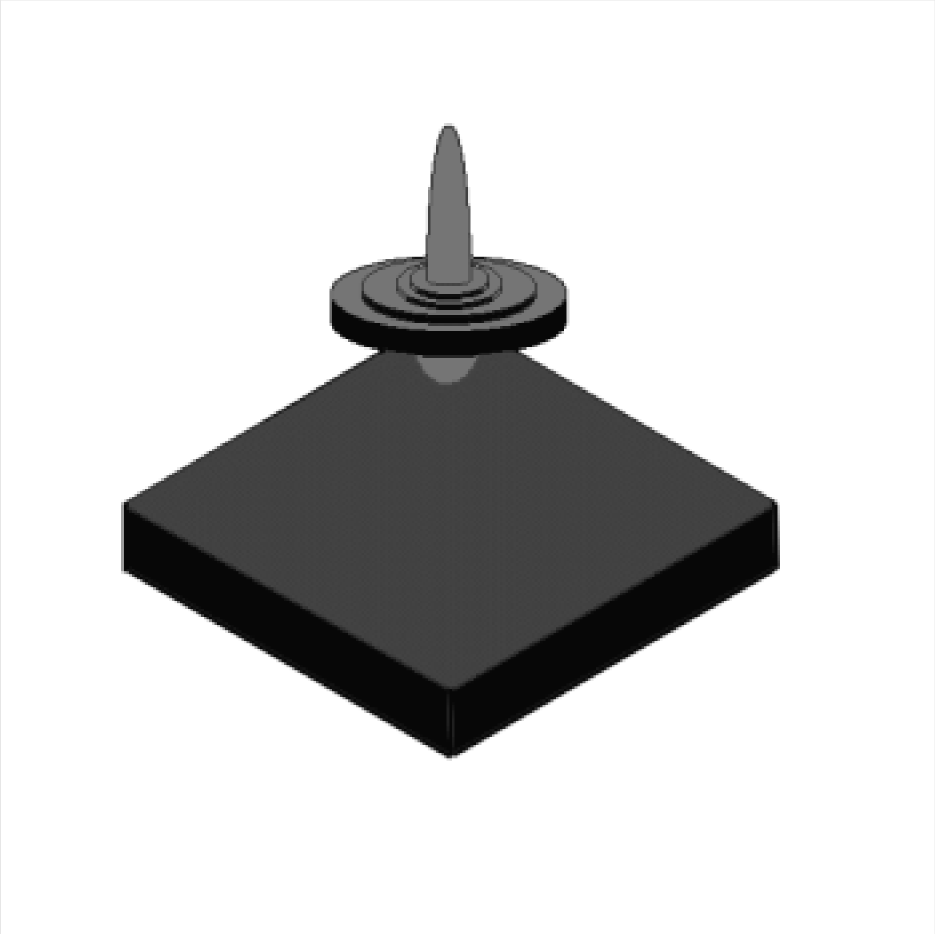


- More precisely:
  - small perturbations → small responses

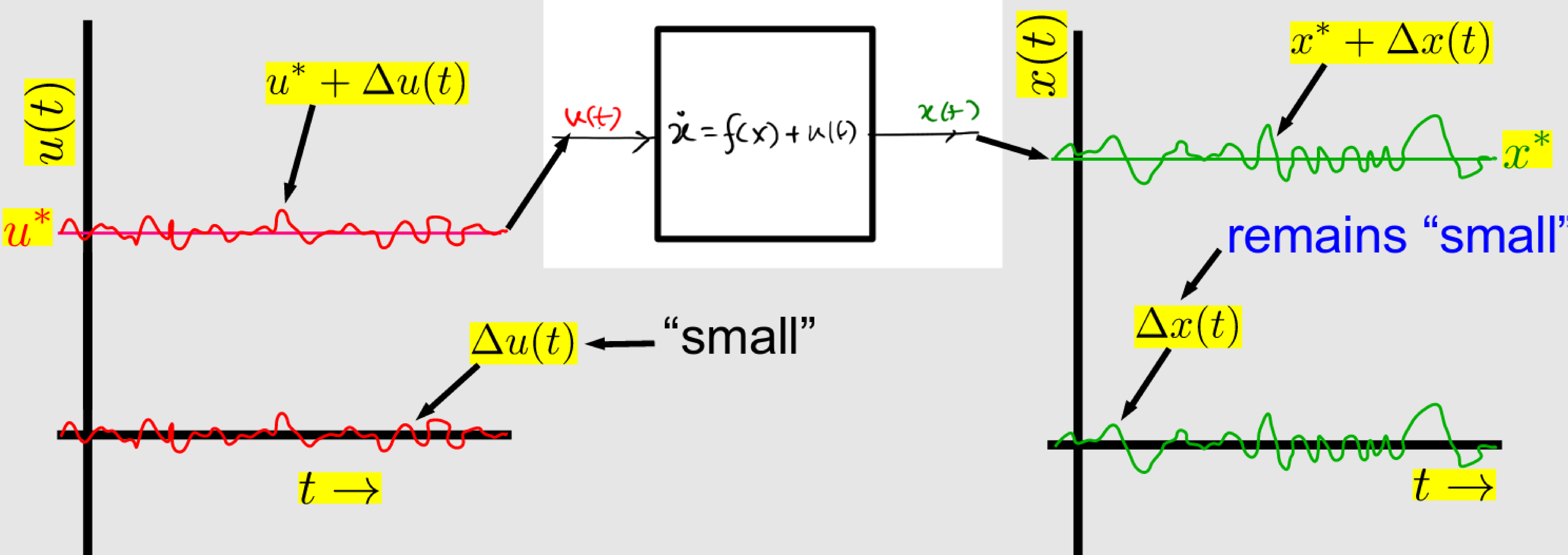


Stability

- Basic idea: **perturb** system **a little** from equilibrium
  - does it come back? yes → STABLE

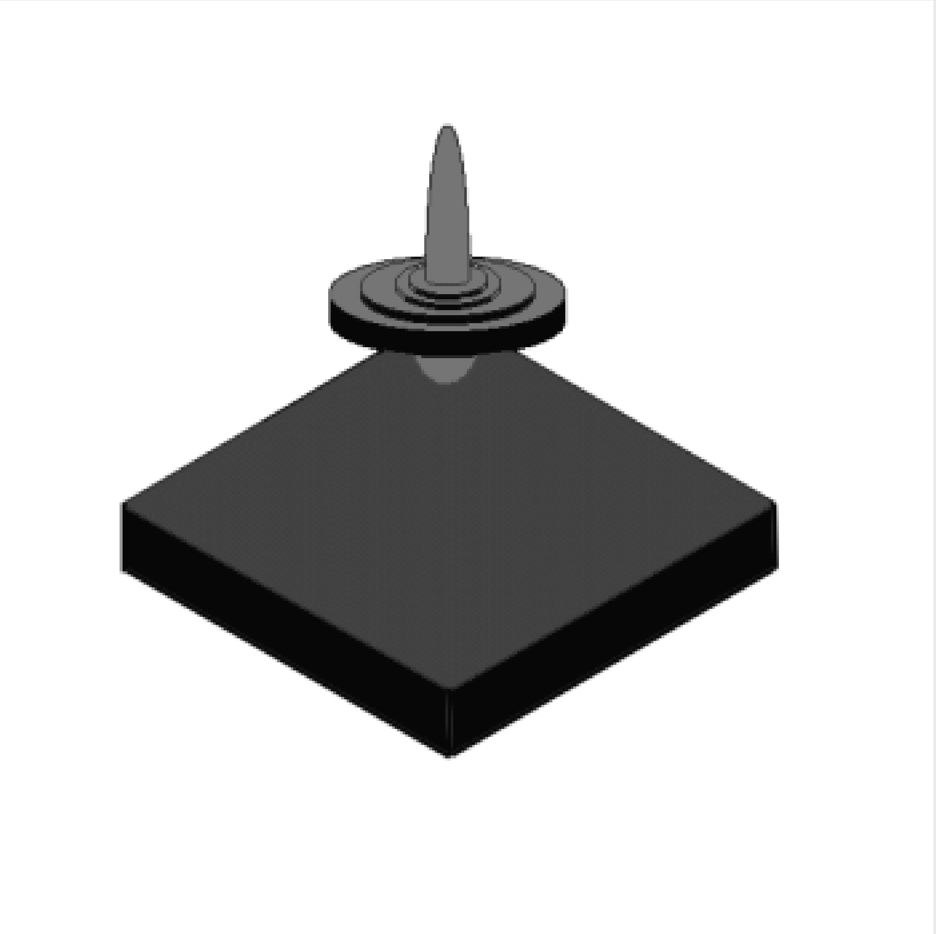


- More precisely:
  - small perturbations → small responses

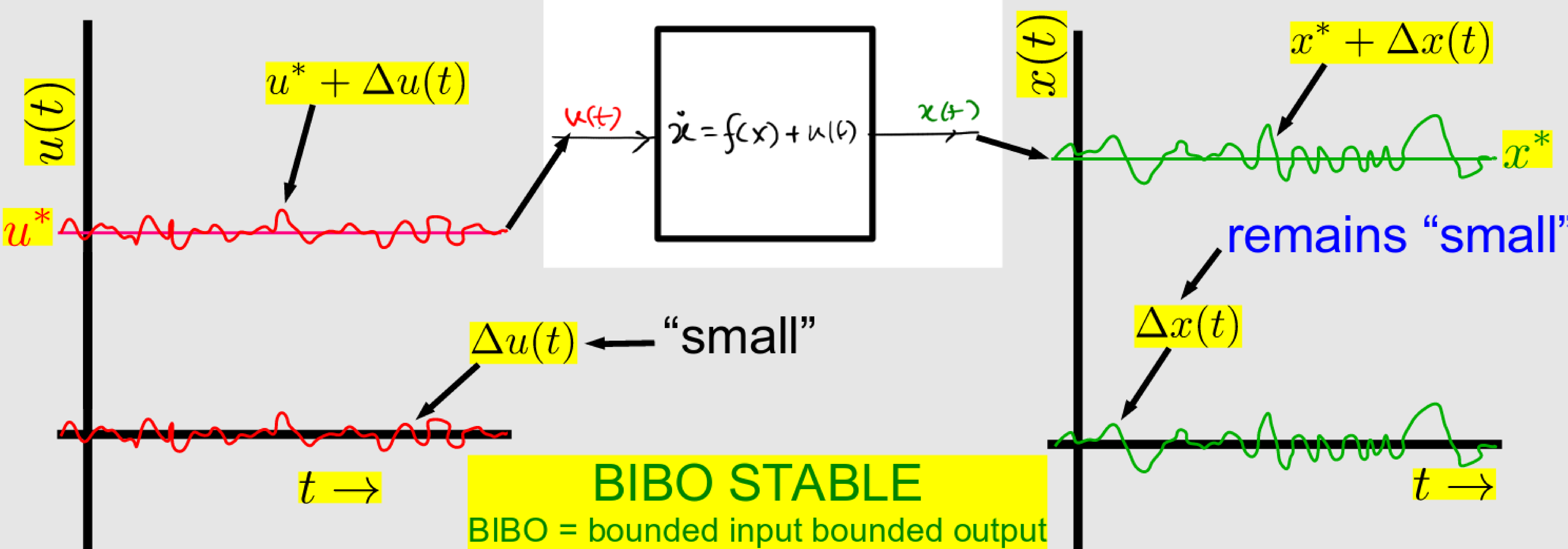


Stability

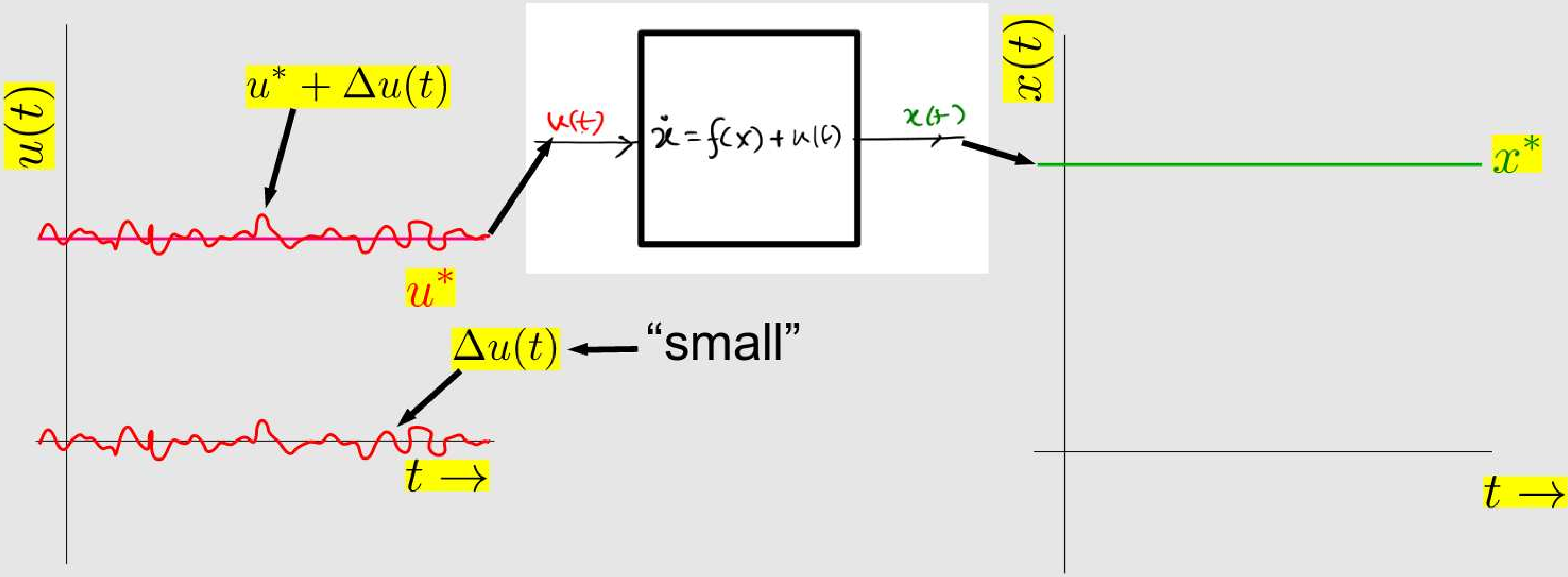
- Basic idea: **perturb** system **a little** from equilibrium
  - does it come back? yes → STABLE

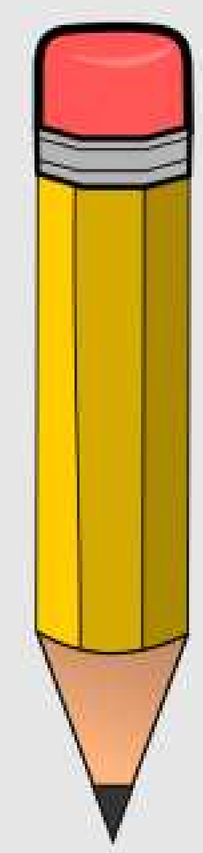


- More precisely:
  - small perturbations → small responses

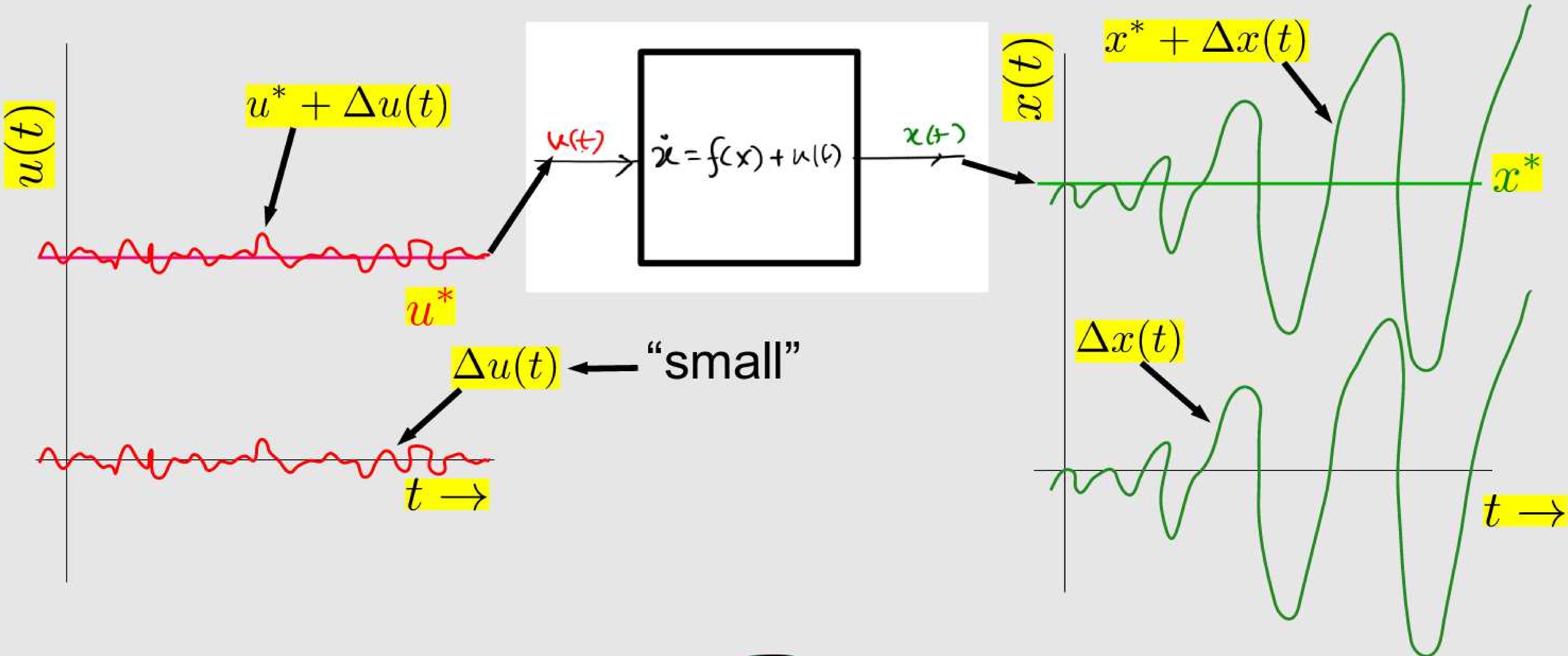


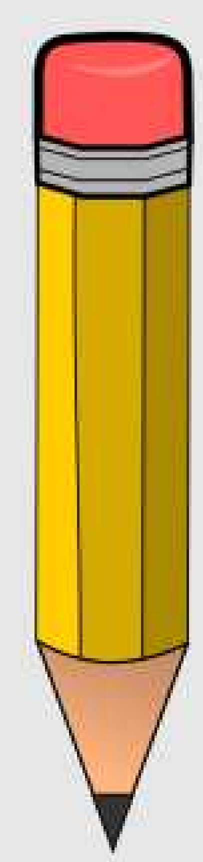
Stability (contd. - 2)



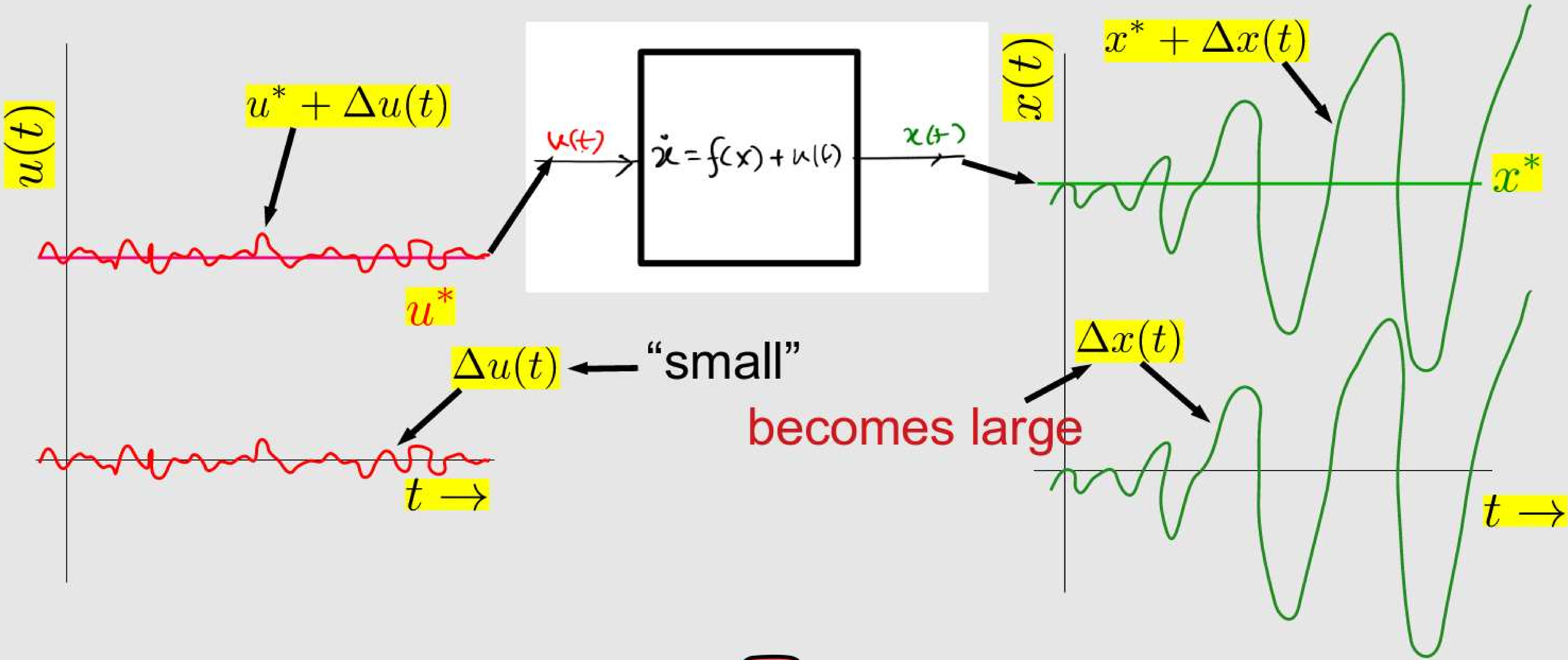


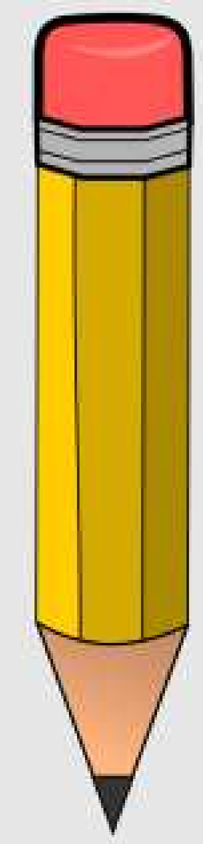
Stability (contd. - 2)



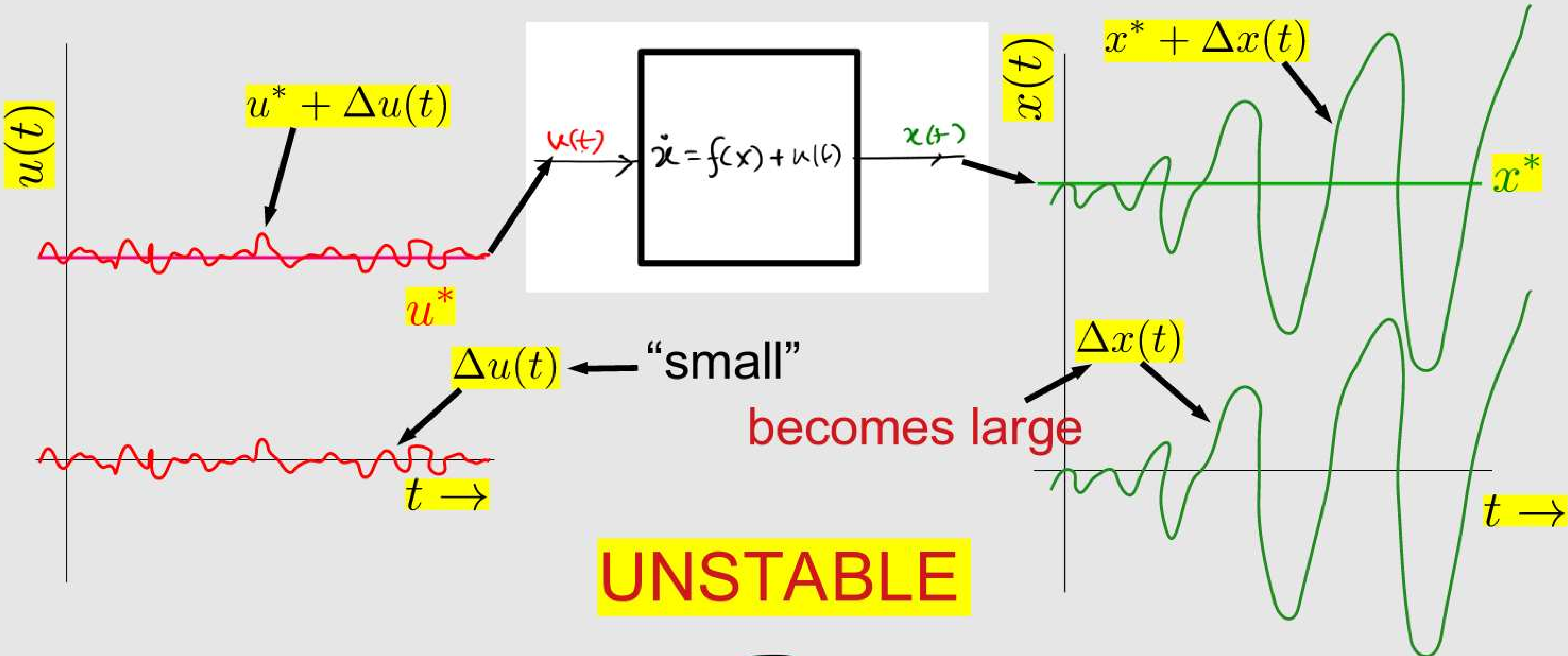


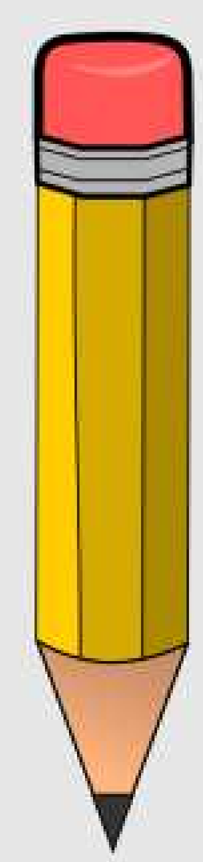
Stability (contd. - 2)





Stability (contd. - 2)





Stability: the Scalar Case

- Analysis: start w scalar case: $\frac{d}{dt}\Delta x(t) = a\Delta x(t) + b\Delta u(t)$
  - [already linear(ized); everything is real]

Stability: the Scalar Case

- Analysis: start w scalar case: $\frac{d}{dt}\Delta x(t) = a\Delta x(t) + b\Delta u(t)$ real
  - [already linear(ized); everything is real]

Stability: the Scalar Case

- Analysis: start w scalar case: $\frac{d}{dt}\Delta x(t) = \overset{\text{real}}{a}\Delta x(t) + \overset{\text{real}}{b}\Delta u(t)$
  - [already linear(ized); everything is real]

- Solution: $\Delta x(t) = \Delta x(0)e^{at} + \int_0^t e^{a(t-\tau)}\, b\Delta u(\tau)\, d\tau$
  - [obtained by, eg, the method of integrating factors (Piazza: @88)]

Stability: the Scalar Case

- Analysis: start w scalar case: $\frac{d}{dt}\Delta x(t) = a\Delta x(t) + b\Delta u(t)$ real
  - [already linear(ized); everything is real]

initial condition term

- Solution: $\Delta x(t) = \boxed{\Delta x(0)e^{at}} + \int_0^t e^{a(t-\tau)}\, b\Delta u(\tau)\, d\tau$
  - [obtained by, eg, the method of integrating factors (Piazza: @88)]

Stability: the Scalar Case

- Analysis: start w scalar case: $\frac{d}{dt}\Delta x(t) = a\Delta x(t) + b\Delta u(t)$ real
  - [already linear(ized); everything is real]

initial condition term

input term **(convolution)**

- Solution: $\Delta x(t) = \boxed{\Delta x(0) e^{at}} + \boxed{\int_0^t e^{a(t-\tau)}\, b\Delta u(\tau)\, d\tau}$
  - [obtained by, eg, the method of integrating factors (Piazza: @88)]

Stability: the Scalar Case

- Analysis: start w scalar case: $\frac{d}{dt}\Delta x(t) = a\Delta x(t) + b\Delta u(t)$
  - real
  - [already linear(ized); everything is real]

initial condition term

input term **(convolution)**

- Solution: $\Delta x(t) = \Delta x(0)e^{at} + \int_0^t e^{a(t-\tau)}\, b\Delta u(\tau)\, d\tau$ $\quad e^{at} * (b\Delta u(t))$
  - [obtained by, eg, the method of integrating factors (Piazza: @88)]

Stability: the Scalar Case

- Analysis: start w scalar case: $\frac{d}{dt}\Delta x(t) = a\Delta x(t) + b\Delta u(t)$ real
  - [already linear(ized); everything is real]

initial condition term

input term **(convolution)**

- Solution: $\Delta x(t) = \Delta x(0)e^{at} + \int_0^t e^{a(t-\tau)}\; b\Delta u(\tau)\,d\tau$ $e^{at} * (b\Delta u(t))$
  - [obtained by, eg, the method of integrating factors (Piazza: @88)]
- The initial condition term: $\Delta x(0)e^{at}$. Say $\Delta x(0) \neq 0.$

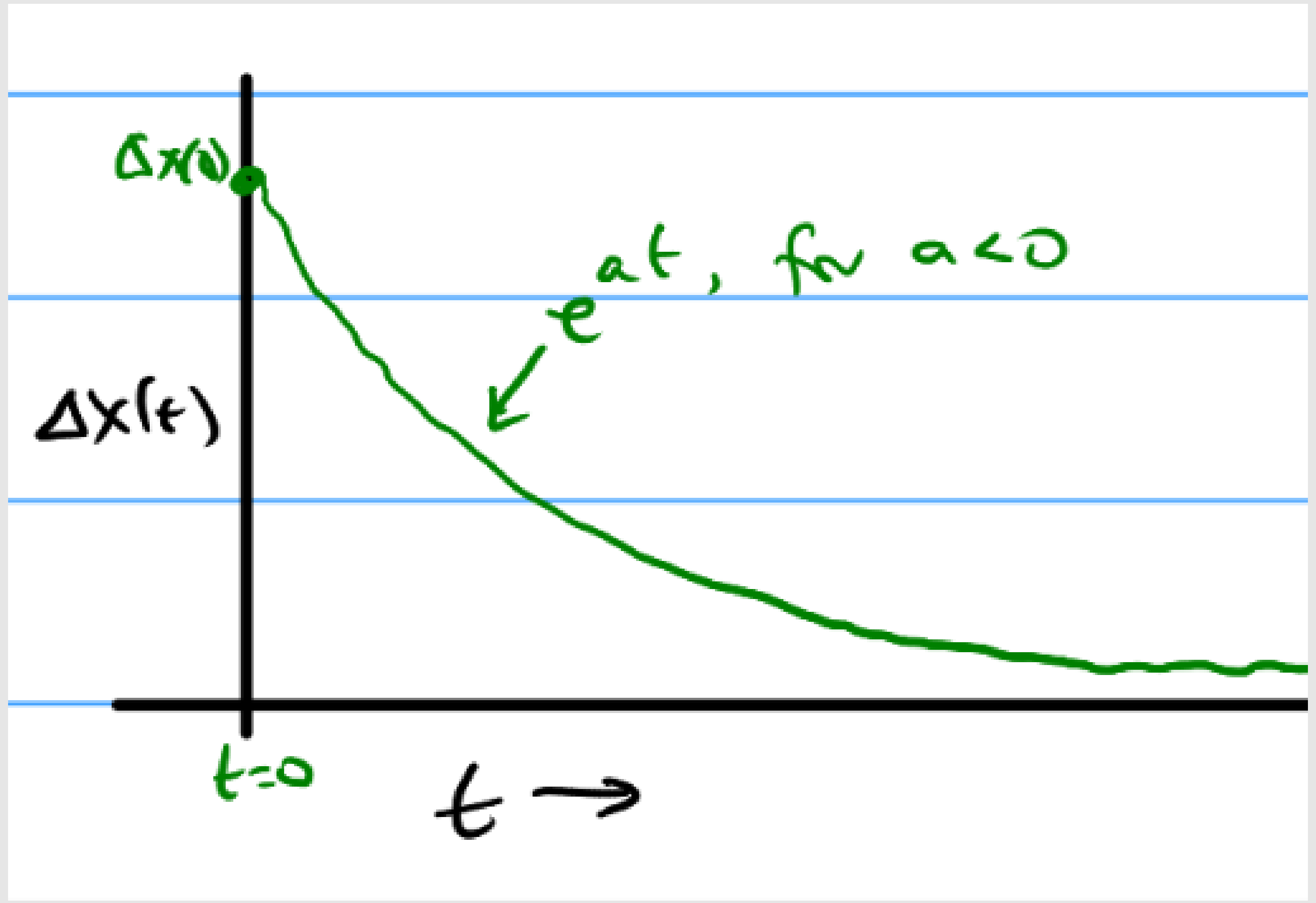
Stability: the Scalar Case

- Analysis: start w scalar case: $\frac{d}{dt}\Delta x(t) = a\Delta x(t) + b\Delta u(t)$ real
  - [already linear(ized); everything is real]

initial condition term

input term **(convolution)**

- Solution: $\Delta x(t) = \Delta x(0)e^{at} + \int_0^t e^{a(t-\tau)}\, b\Delta u(\tau)\, d\tau$ $e^{at} * (b\Delta u(t))$
  - [obtained by, eg, the method of integrating factors (Piazza: @88)]
- The initial condition term: $\Delta x(0)e^{at}$. Say $\Delta x(0) \neq 0$.



Stability: the Scalar Case

- Analysis: start w scalar case: $\frac{d}{dt}\Delta x(t) = a\Delta x(t) + b\Delta u(t)$ real
  - [already linear(ized); everything is real]

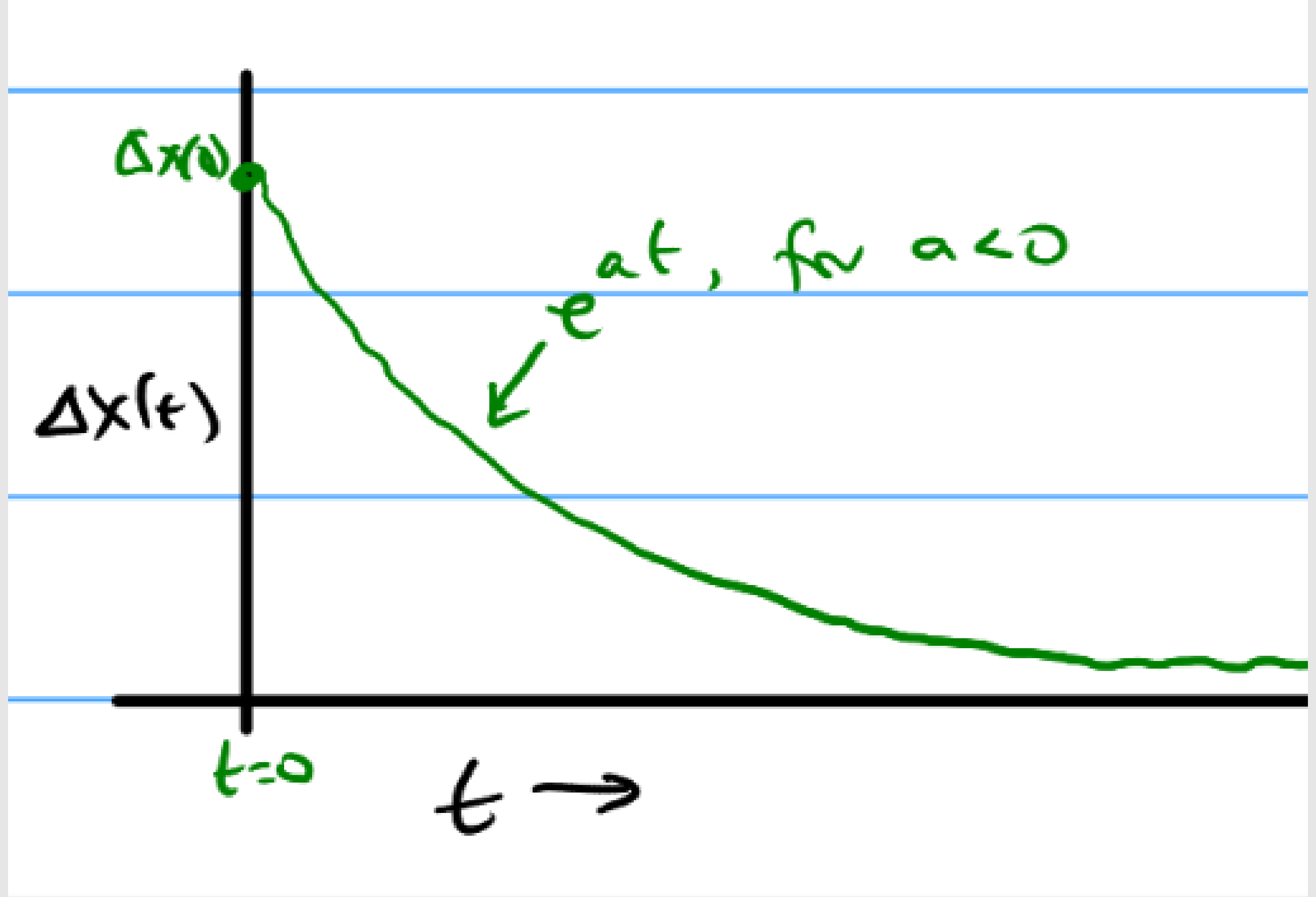
initial condition term

input term **(convolution)**

- Solution: $\Delta x(t) = \Delta x(0)e^{at} + \int_0^t e^{a(t-\tau)}\, b\Delta u(\tau)\, d\tau$ $e^{at} * (b\Delta u(t))$
  - [obtained by, eg, the method of integrating factors (Piazza: @88)]
- The initial condition term: $\Delta x(0)e^{at}$. Say $\Delta x(0) \neq 0$.

a<0: dies down

STABLE



Stability: the Scalar Case

- Analysis: start w scalar case: $\frac{d}{dt}\Delta x(t) = a\Delta x(t) + b\Delta u(t)$ (real: a, b)
  - [already linear(ized); everything is real]

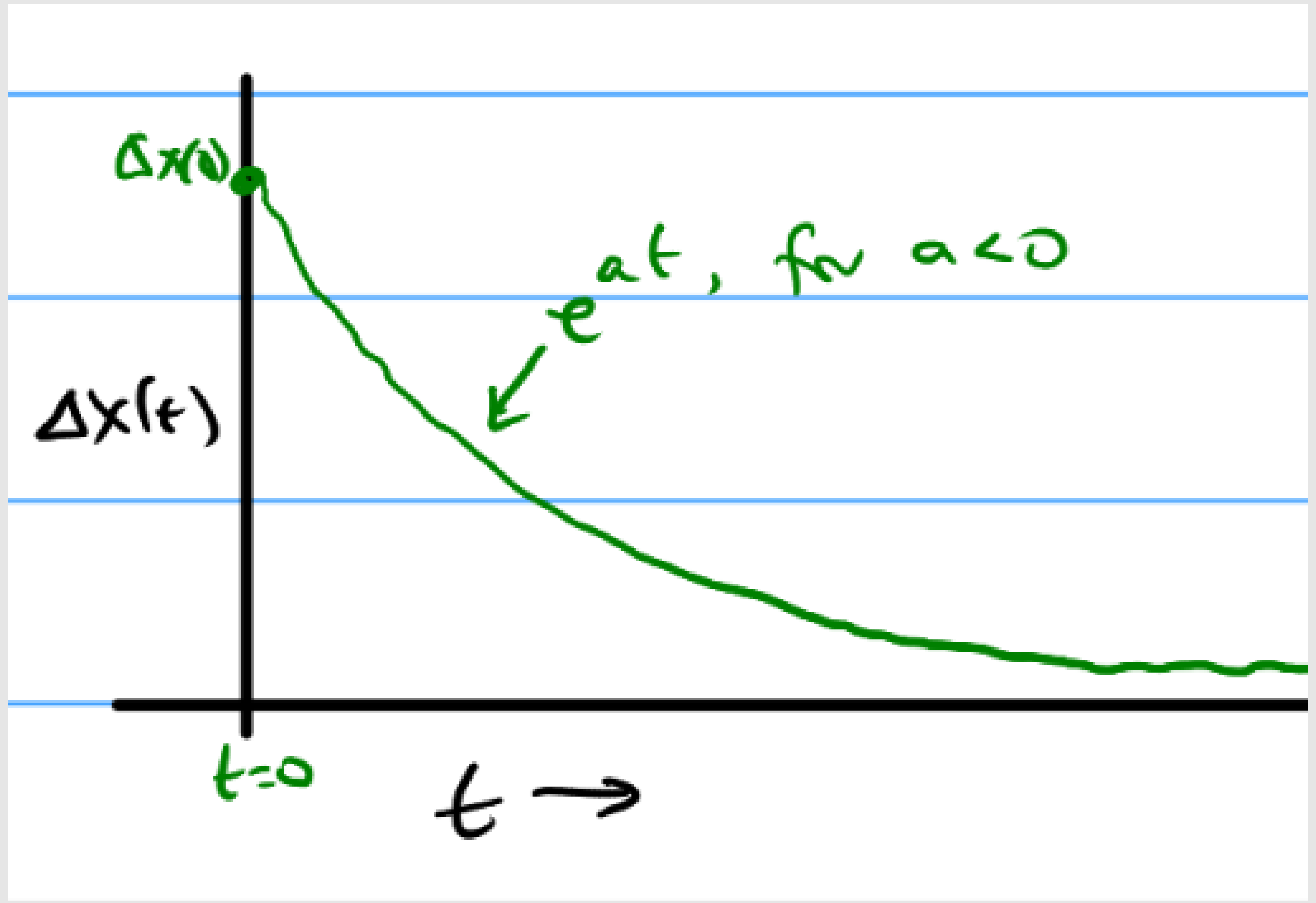
initial condition term

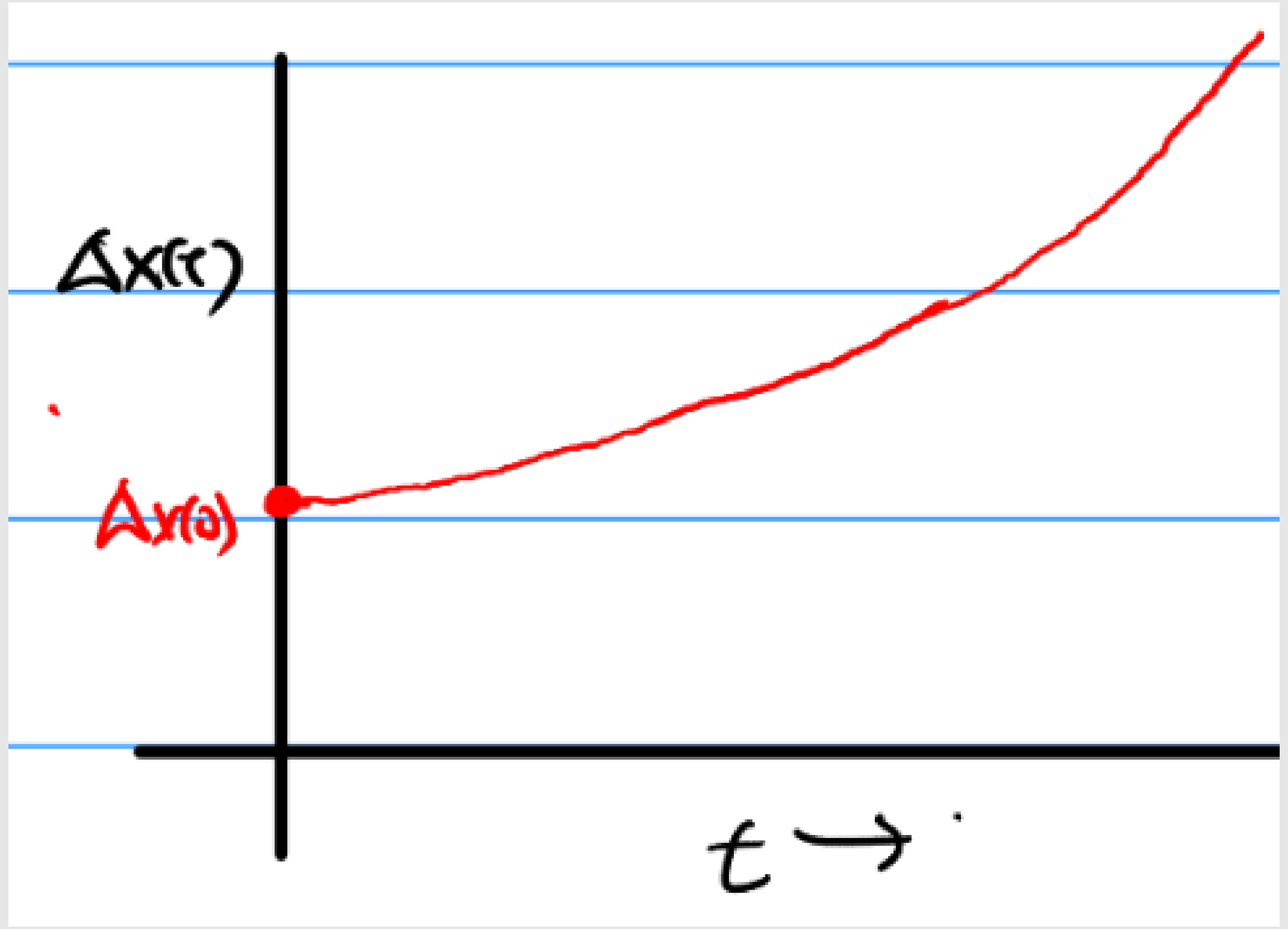
input term **(convolution)**

- Solution: $\Delta x(t) = \Delta x(0)e^{at} + \int_0^t e^{a(t-\tau)}\, b\Delta u(\tau)\, d\tau$ ← $e^{at} * (b\Delta u(t))$
  - [obtained by, eg, the method of integrating factors (Piazza: @88)]
- The initial condition term: $\Delta x(0)e^{at}$. Say $\Delta x(0) \neq 0$.

a<0: dies down

STABLE





Stability: the Scalar Case

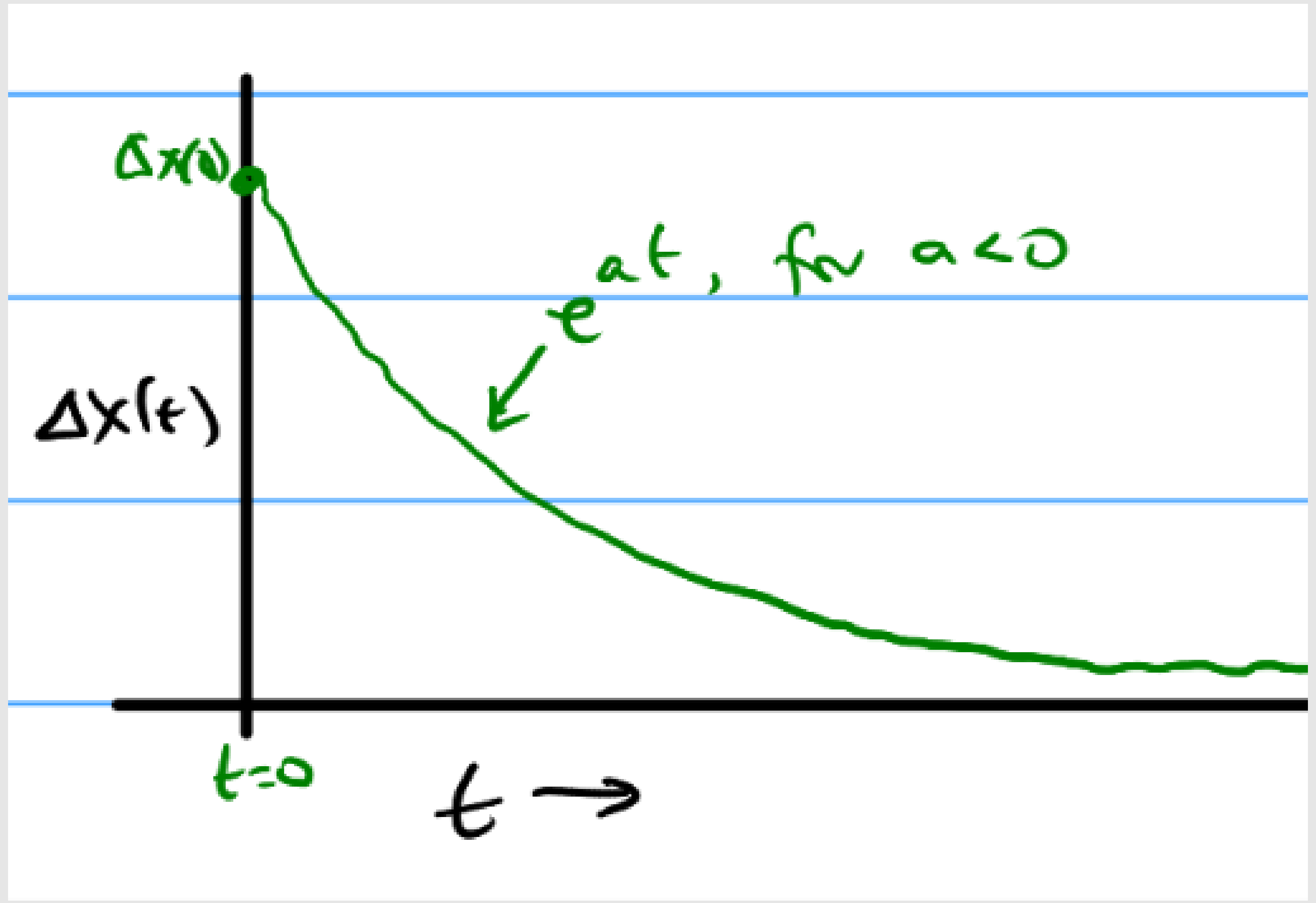
- Analysis: start w scalar case: $\frac{d}{dt}\Delta x(t) = a\Delta x(t) + b\Delta u(t)$ — real
  - [already linear(ized); everything is real]

initial condition term

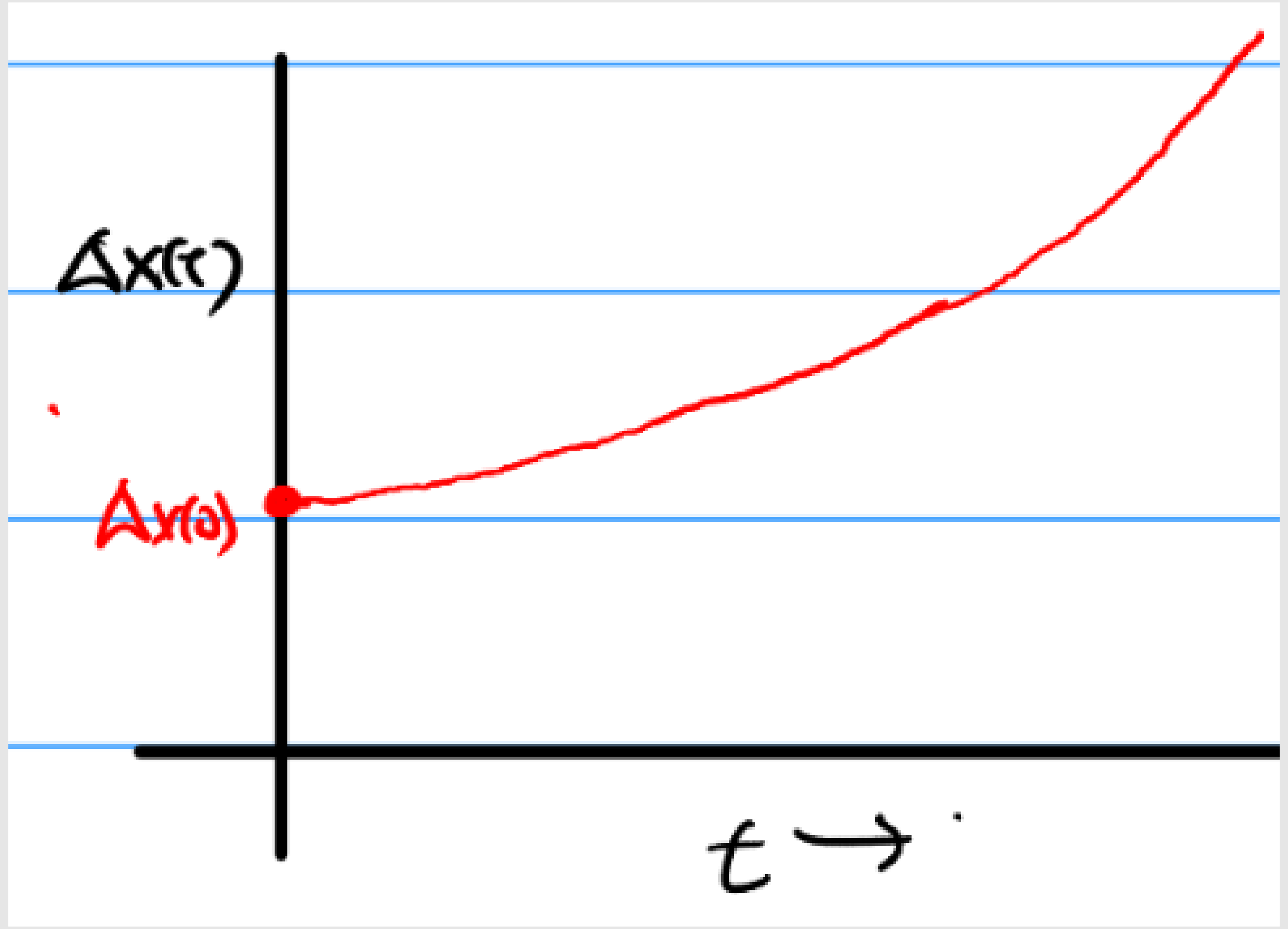
input term **(convolution)**

- Solution: $\Delta x(t) = \Delta x(0)e^{at} + \int_0^t e^{a(t-\tau)}\, b\Delta u(\tau)\, d\tau$ — $e^{at} * (b\Delta u(t))$
  - [obtained by, eg, the method of integrating factors (Piazza: @88)]
- The initial condition term: $\Delta x(0)e^{at}$. Say $\Delta x(0) \neq 0$.

a<0: dies down
STABLE



a>0: blows up
UNSTABLE



Stability: the Scalar Case

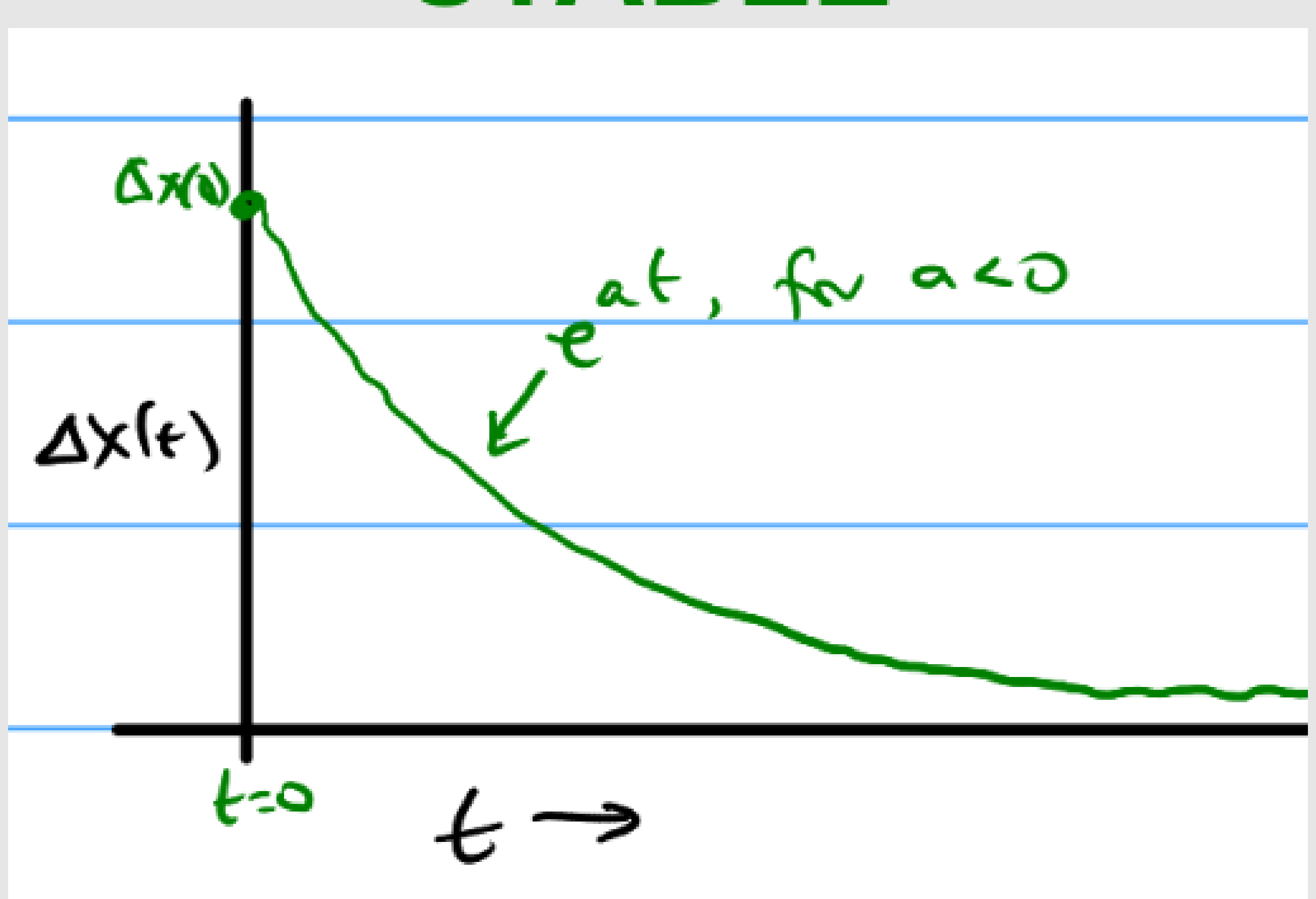
- Analysis: start w scalar case: $\frac{d}{dt}\Delta x(t) = a\Delta x(t) + b\Delta u(t)$
  - real
  - [already linear(ized); everything is real]

initial condition term

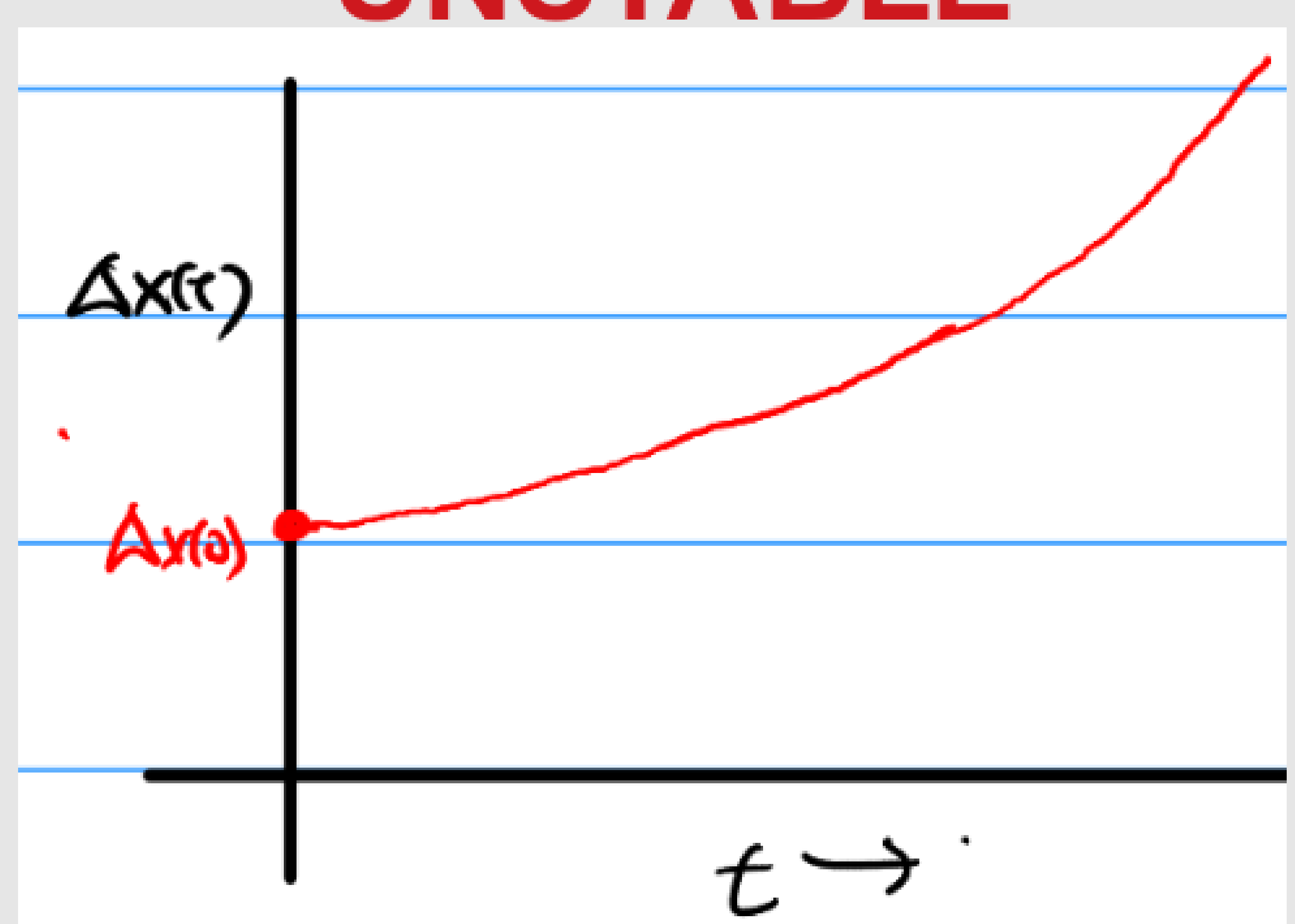
input term **(convolution)**

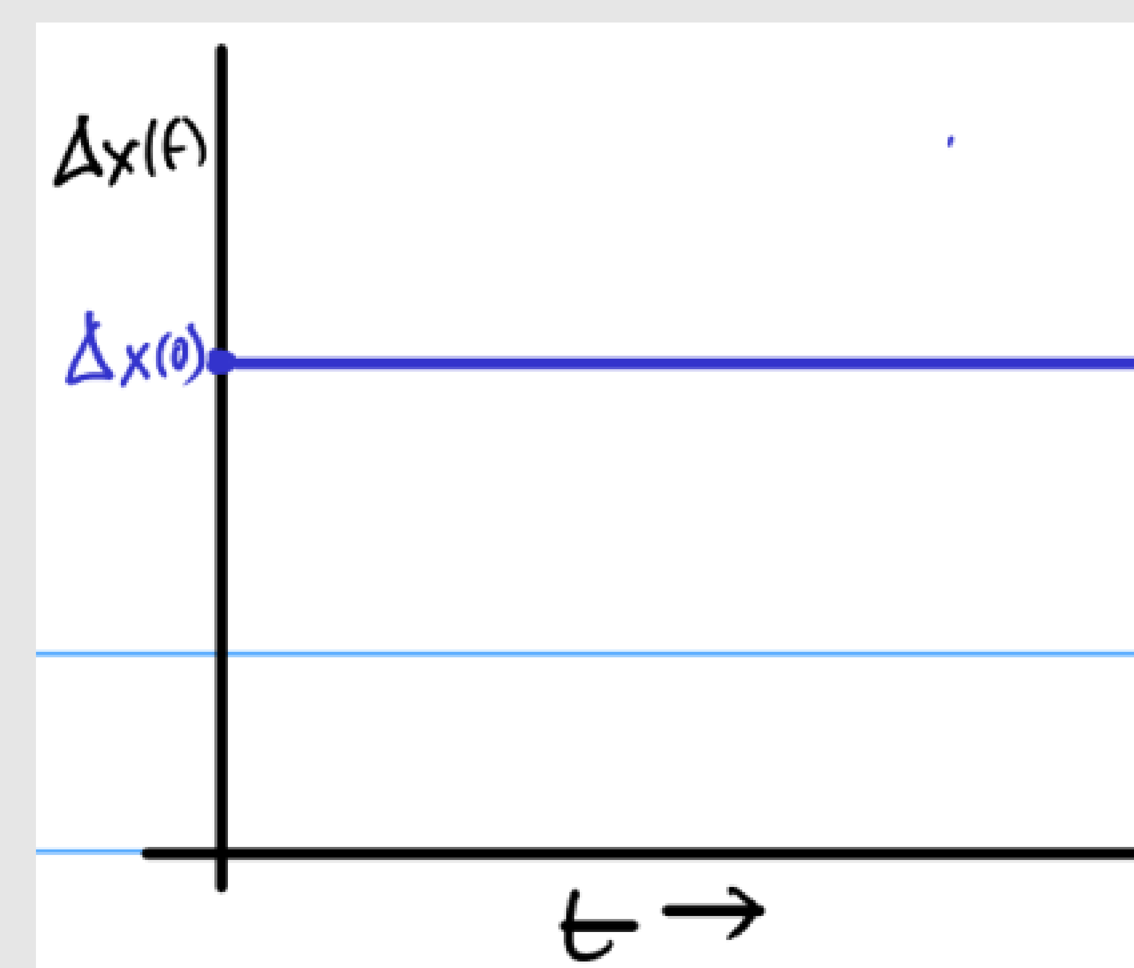
- Solution: $\Delta x(t) = \Delta x(0)e^{at} + \int_0^t e^{a(t-\tau)}\, b\Delta u(\tau)\, d\tau$ $\qquad e^{at} * (b\Delta u(t))$
  - [obtained by, eg, the method of integrating factors (Piazza: @88)]
- The initial condition term: $\Delta x(0)e^{at}$. Say $\Delta x(0) \neq 0$.

a<0: dies down
STABLE



a>0: blows up
UNSTABLE

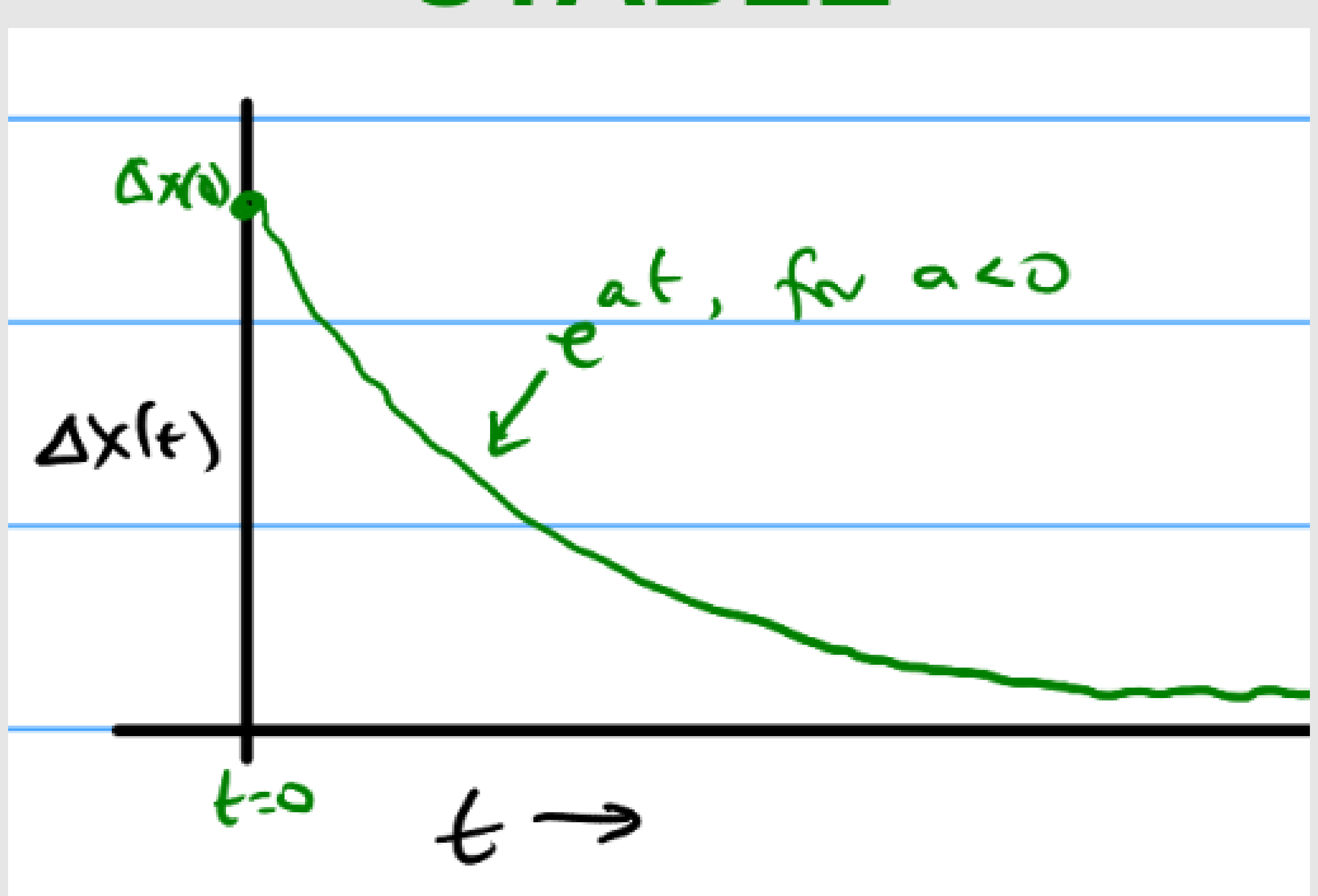




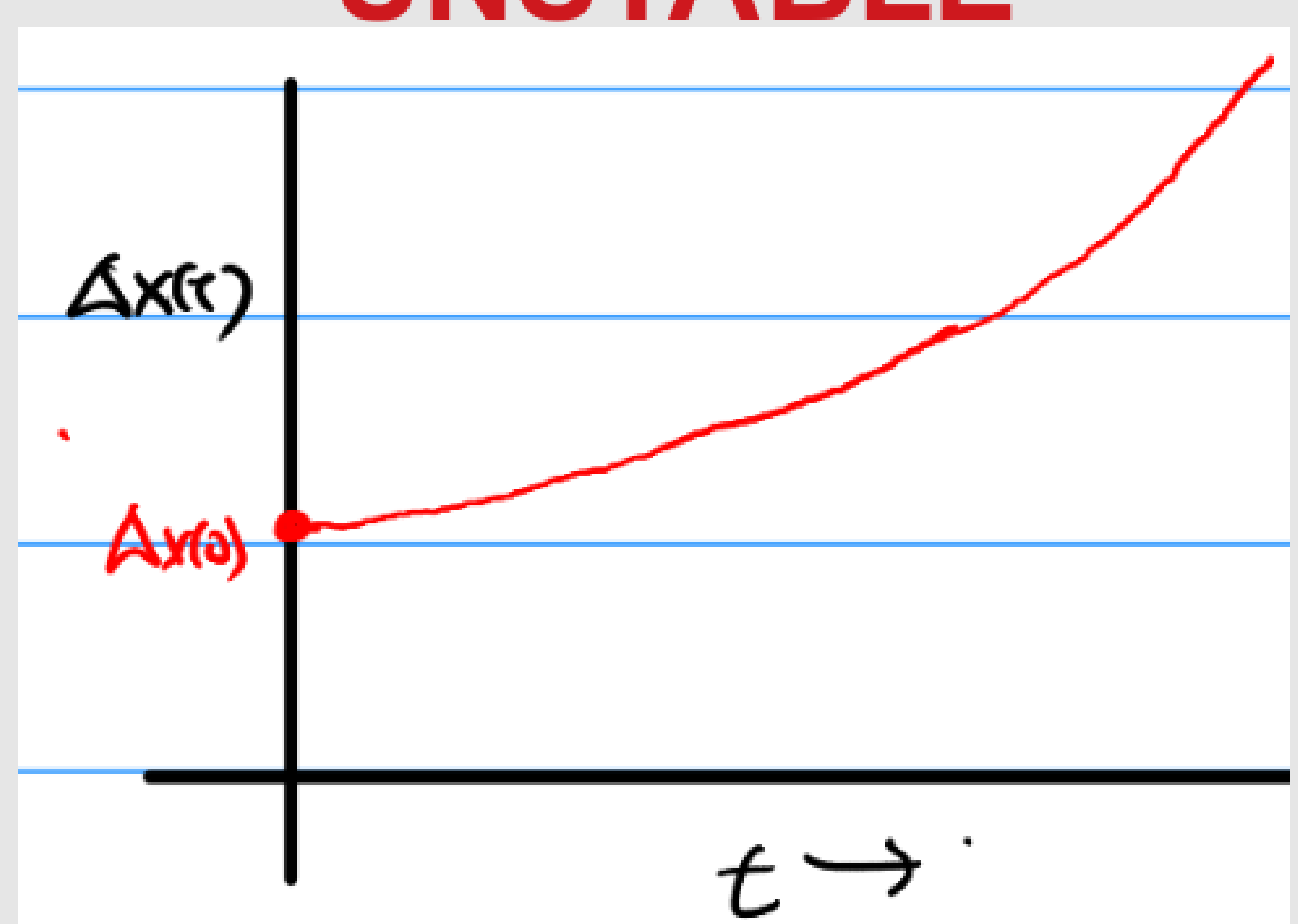
Stability: the Scalar Case

- Analysis: start w scalar case: $\frac{d}{dt}\Delta x(t) = a\Delta x(t) + b\Delta u(t)$ (real: a, b)
  - [already linear(ized); everything is real]
- Solution: $\Delta x(t) = \underbrace{\Delta x(0)e^{at}}_{\text{initial condition term}} + \underbrace{\int_0^t e^{a(t-\tau)}\, b\Delta u(\tau)\, d\tau}_{\text{input term (convolution)}}$ ← $e^{at} * (b\Delta u(t))$
  - [obtained by, eg, the method of integrating factors (Piazza: @88)]
- The initial condition term: $\Delta x(0)e^{at}$. Say $\Delta x(0) \neq 0$.

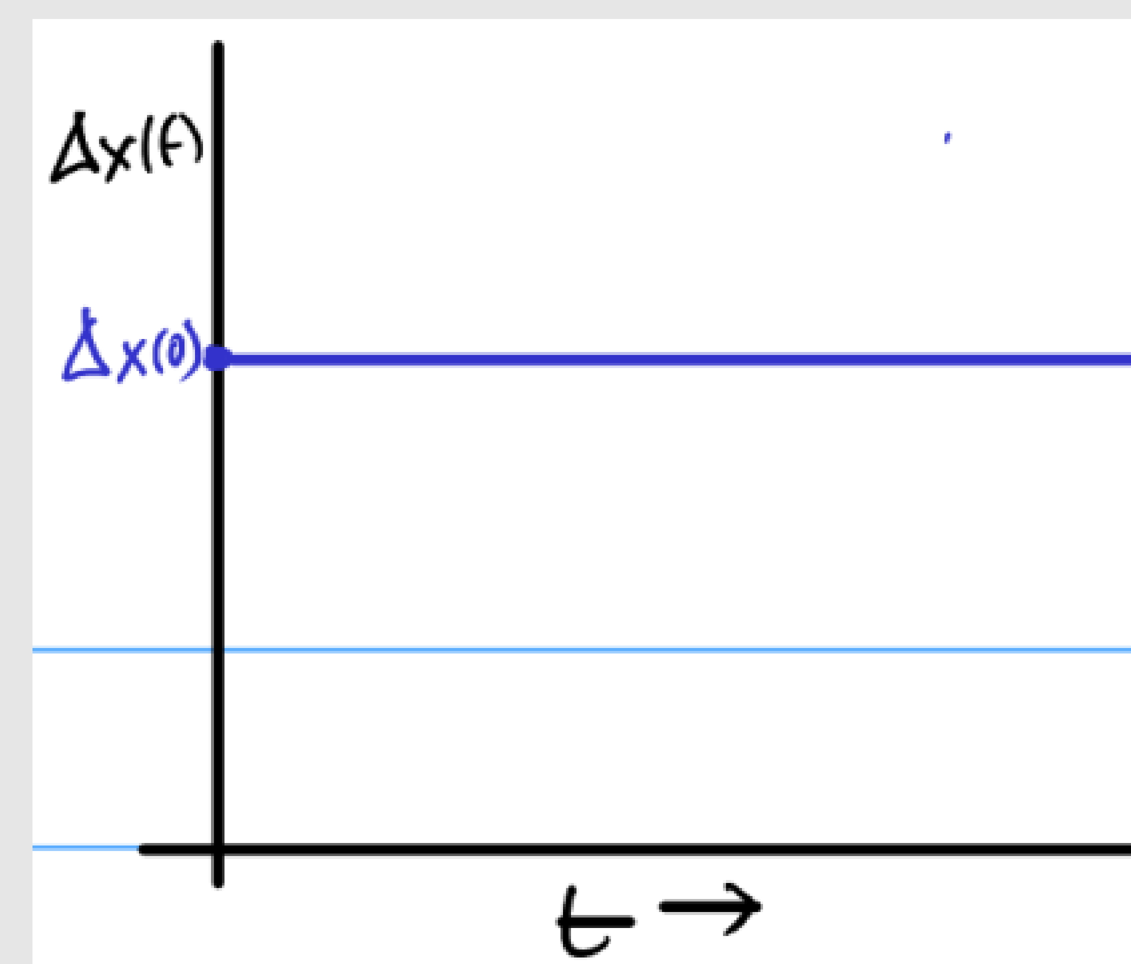
a<0: dies down
STABLE



a>0: blows up
UNSTABLE



a=0: stays the same
MARGINALLY STABLE



Stability: Scalar Case (contd.)

- Solution: $\Delta x(t) = \Delta x(0) e^{at} + \int_0^t e^{a(t-\tau)} \, b\Delta u(\tau) \, d\tau$ $e^{at} * (b\Delta u(t))$

input term **(convolution)**

Stability: Scalar Case (contd.)

- Solution: $\Delta x(t) = \Delta x(0) e^{at} + \int_0^t e^{a(t-\tau)}\, b\Delta u(\tau)\, d\tau$ ⟵ $e^{at} * (b\Delta u(t))$

  input term **(convolution)**

- Can show (see handwritten notes):
  - if a<0: $e^{at} * (b\Delta u(t))$ bounded if Δu(t) bounded: **BIBO stable**

Stability: Scalar Case (contd.)

- Solution: $\Delta x(t) = \Delta x(0)e^{at} + \int_0^t e^{a(t-\tau)}\, b\Delta u(\tau)\, d\tau$ ← $e^{at} * (b\Delta u(t))$

  input term **(convolution)**

- Can show (see handwritten notes):
  - if a<0: $e^{at} * (b\Delta u(t))$ bounded if Δu(t) bounded: **BIBO stable**
  - if a>0: $e^{at} * (b\Delta u(t))$ unbounded even if Δu(t) bounded: **UNSTABLE**

Stability: Scalar Case (contd.)

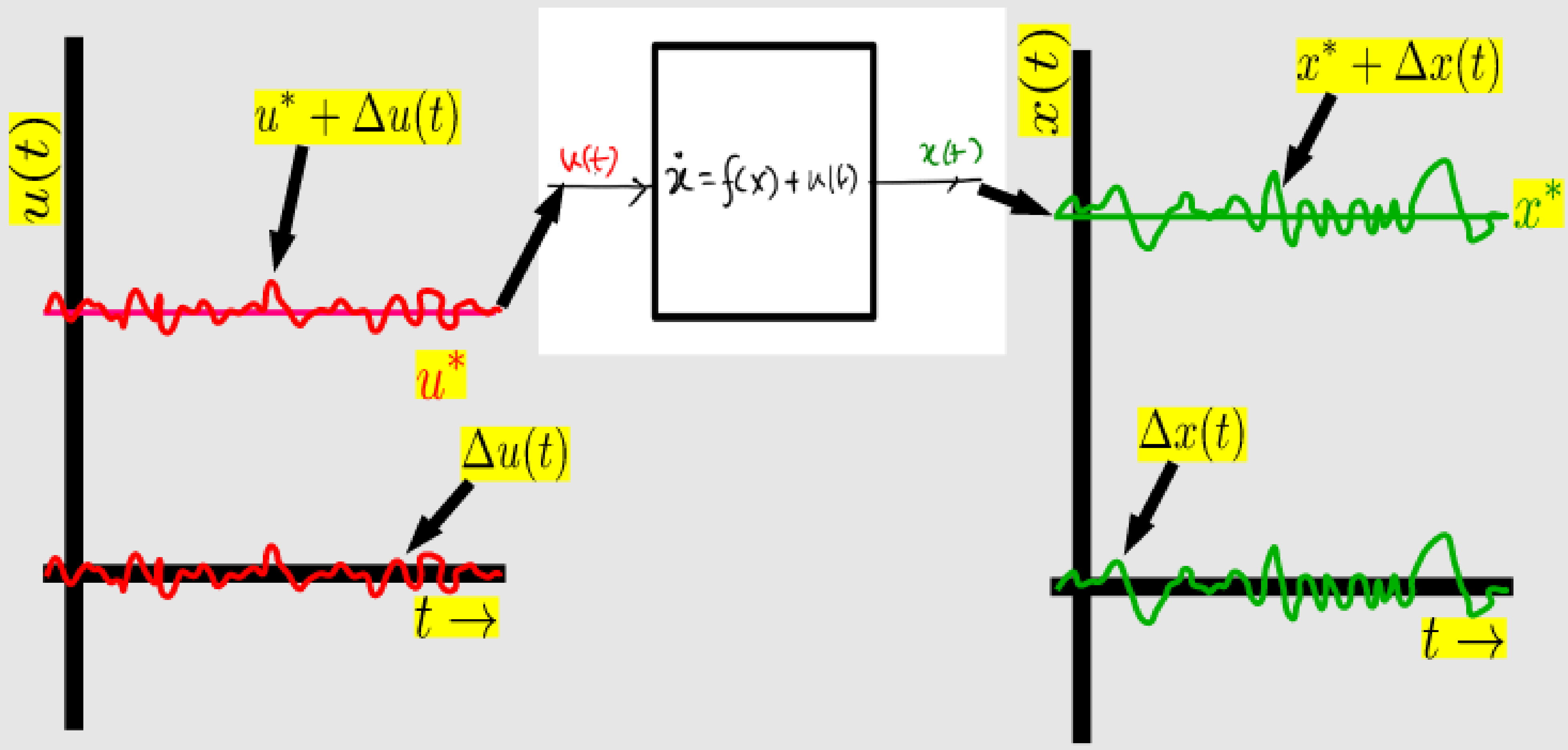
- Solution: $\Delta x(t) = \Delta x(0)e^{at} + \int_0^t e^{a(t-\tau)}\, b\Delta u(\tau)\, d\tau$ ← $e^{at} * (b\Delta u(t))$
  input term **(convolution)**
- Can show (see handwritten notes):
  - if a<0: $e^{at} * (b\Delta u(t))$ bounded if Δu(t) bounded: **BIBO stable**
  - if a>0: $e^{at} * (b\Delta u(t))$ unbounded even if Δu(t) bounded: **UNSTABLE**
  - if a=0: $e^{at} * (b\Delta u(t))$ unbounded even if Δu(t) bounded: **UNSTABLE**

Stability: Scalar Case (contd.)

- Solution: $\Delta x(t) = \Delta x(0) e^{at} + \int_0^t e^{a(t-\tau)}\, b\Delta u(\tau)\, d\tau$ ← $e^{at} * (b\Delta u(t))$

  input term **(convolution)**
- Can show (see handwritten notes):
  - if a<0: $e^{at} * (b\Delta u(t))$ bounded if Δu(t) bounded: **BIBO stable**
  - if a>0: $e^{at} * (b\Delta u(t))$ unbounded even if Δu(t) bounded: **UNSTABLE**
  - if a=0: $e^{at} * (b\Delta u(t))$ unbounded even if Δu(t) bounded: **UNSTABLE**

a<0: BIBO STABLE

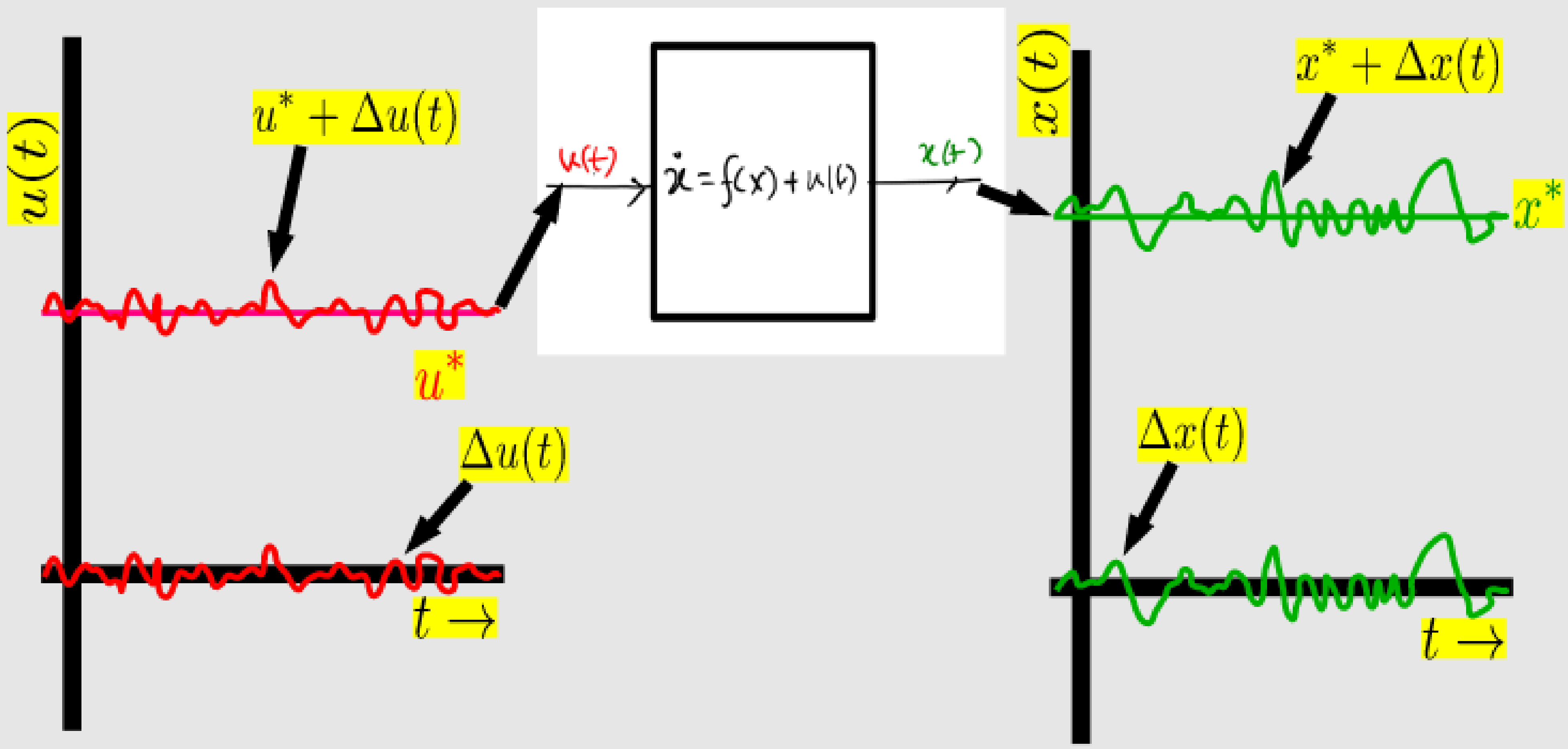


Stability: Scalar Case (contd.)

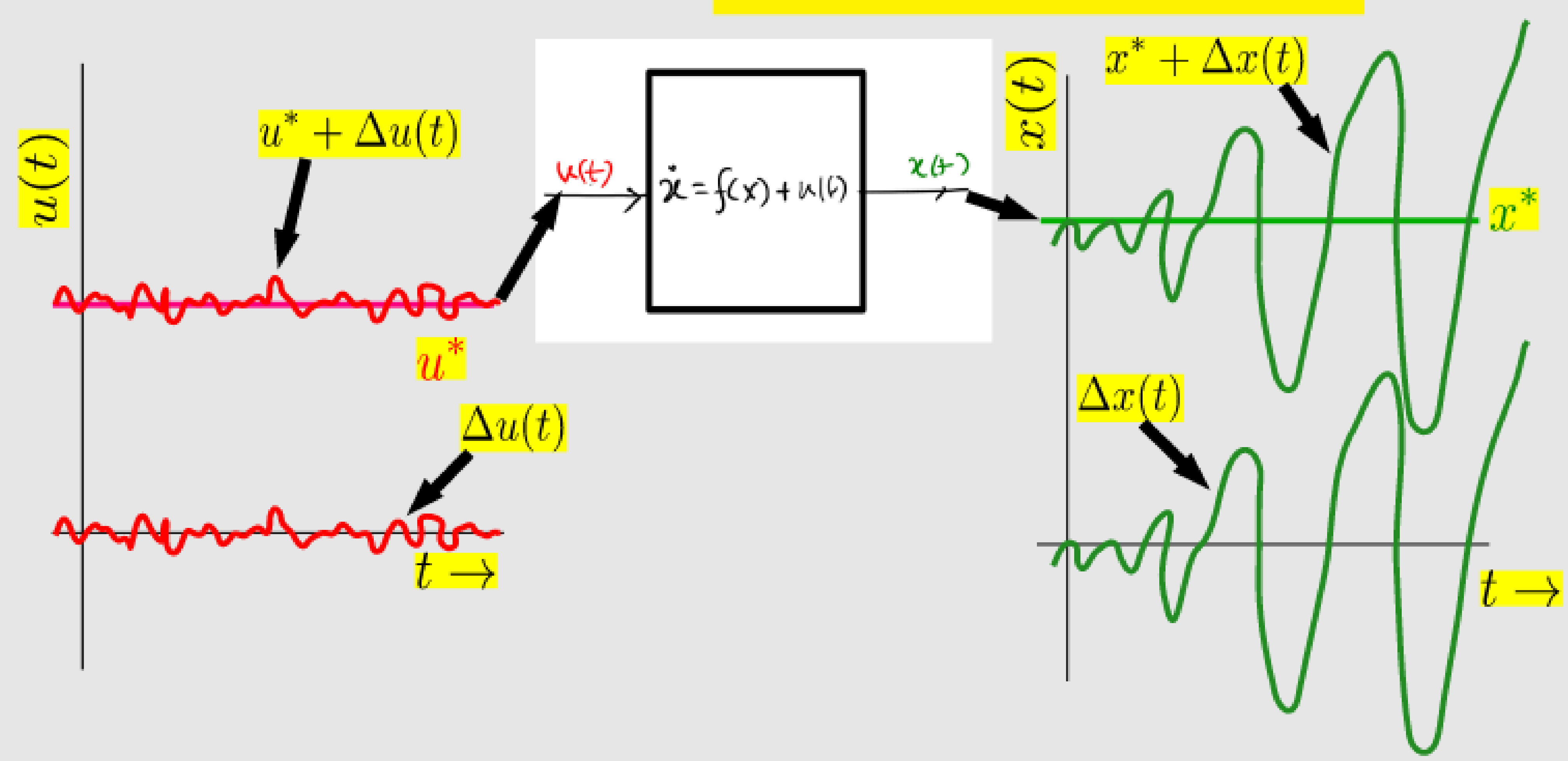
- Solution: $\Delta x(t) = \Delta x(0)e^{at} + \int_0^t e^{a(t-\tau)}\, b\Delta u(\tau)\, d\tau$ $e^{at} * (b\Delta u(t))$

  input term **(convolution)**
- Can show (see handwritten notes):
  - if a<0: $e^{at} * (b\Delta u(t))$ bounded if Δu(t) bounded: **BIBO stable**
  - if a>0: $e^{at} * (b\Delta u(t))$ unbounded even if Δu(t) bounded: **UNSTABLE**
  - if a=0: $e^{at} * (b\Delta u(t))$ unbounded even if Δu(t) bounded: **UNSTABLE**

a<0: BIBO STABLE



a≥0: UNSTABLE



The Vector Case: Eigendecomposition

- The vector case: $\frac{d}{dt}\Delta\vec{x}(t) = A\Delta\vec{x}(t) + B\Delta\vec{u}(t)$
  - [already linear(ized); everything is real]

The Vector Case: Eigendecomposition

- The vector case: $\frac{d}{dt}\Delta\vec{x}(t) = A\Delta\vec{x}(t) + B\Delta\vec{u}(t)$
  - [already linear(ized); everything is real] real matrices

The Vector Case: Eigendecomposition

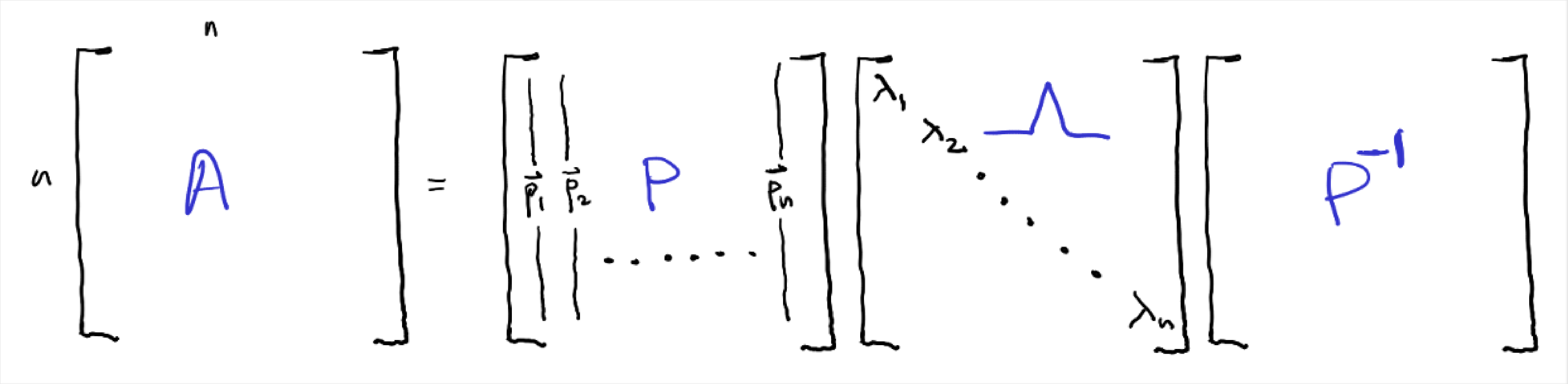
- The vector case: $\frac{d}{dt}\Delta\vec{x}(t) = A\Delta\vec{x}(t) + B\Delta\vec{u}(t)$
  - [already linear(ized); everything is real] real matrices
- Can be "decomposed" into n scalar systems
  - the key idea: to eigendecompose A

The Vector Case: Eigendecomposition

- The vector case: $\frac{d}{dt}\Delta\vec{x}(t) = A\Delta\vec{x}(t) + B\Delta\vec{u}(t)$
  - [already linear(ized); everything is real] real matrices
- Can be “decomposed” into n scalar systems
  - the key idea: to eigendecompose A

if time, move to xournal

- (recap) eigendecomposition: given an nxn matrix A:*

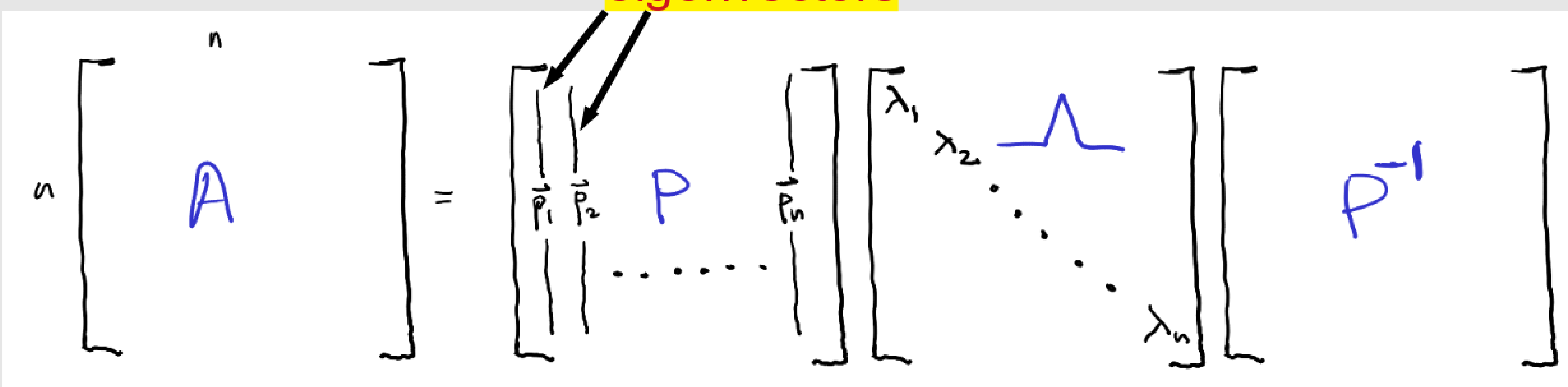


* diagonalization always possible if all eigenvalues distinct (assumed)

The Vector Case: Eigendecomposition

- The vector case: $\frac{d}{dt}\Delta\vec{x}(t) = A\Delta\vec{x}(t) + B\Delta\vec{u}(t)$
  - [already linear(ized); everything is real] real matrices
- Can be "decomposed" into n scalar systems
  - the key idea: to eigendecompose A
- (recap) eigendecomposition: given an nxn matrix A:*

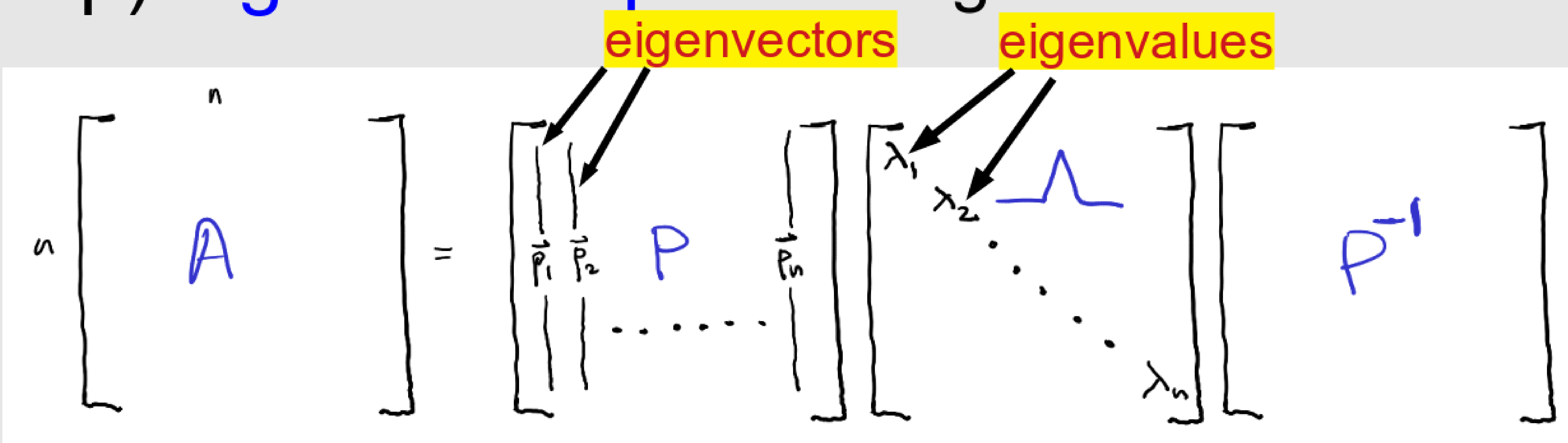
eigenvectors



* diagonalization always possible if all eigenvalues distinct (assumed)

The Vector Case: Eigendecomposition

- The vector case: $\frac{d}{dt}\Delta\vec{x}(t) = A\Delta\vec{x}(t) + B\Delta\vec{u}(t)$
  - [already linear(ized); everything is real] real matrices
- Can be “decomposed” into n scalar systems
  - the key idea: to eigendecompose A
- (recap) eigendecomposition: given an nxn matrix A:*



* diagonalization always possible if all eigenvalues distinct (assumed)

The Vector Case: Eigendecomposition

- The vector case: $\frac{d}{dt}\Delta\vec{x}(t) = A\Delta\vec{x}(t) + B\Delta\vec{u}(t)$
  - [already linear(ized); everything is real] real matrices
- Can be "decomposed" into n scalar systems
  - the key idea: to eigendecompose A
- (recap) eigendecomposition: given an nxn matrix A:*

eigenvectors eigenvalues

$$\begin{array}{cc} & n \\ n & \begin{bmatrix} & & \\ & A & \\ & & \end{bmatrix} \end{array} = \underbrace{\begin{bmatrix} \vec{p}_1 & \vec{p}_2 & \cdots & \vec{p}_n \end{bmatrix}}_{P} \underbrace{\begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{bmatrix}}_{\Lambda} \begin{bmatrix} & & \\ & P^{-1} & \\ & & \end{bmatrix}$$

$$\begin{bmatrix} & & \\ & A & \\ & & \end{bmatrix} \underbrace{\begin{bmatrix} \vec{p}_1 & \vec{p}_2 & \cdots & \vec{p}_n \end{bmatrix}}_{P} = \underbrace{\begin{bmatrix} \vec{p}_1 & \vec{p}_2 & \cdots & \vec{p}_n \end{bmatrix}}_{P} \underbrace{\begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{bmatrix}}_{\Lambda}$$

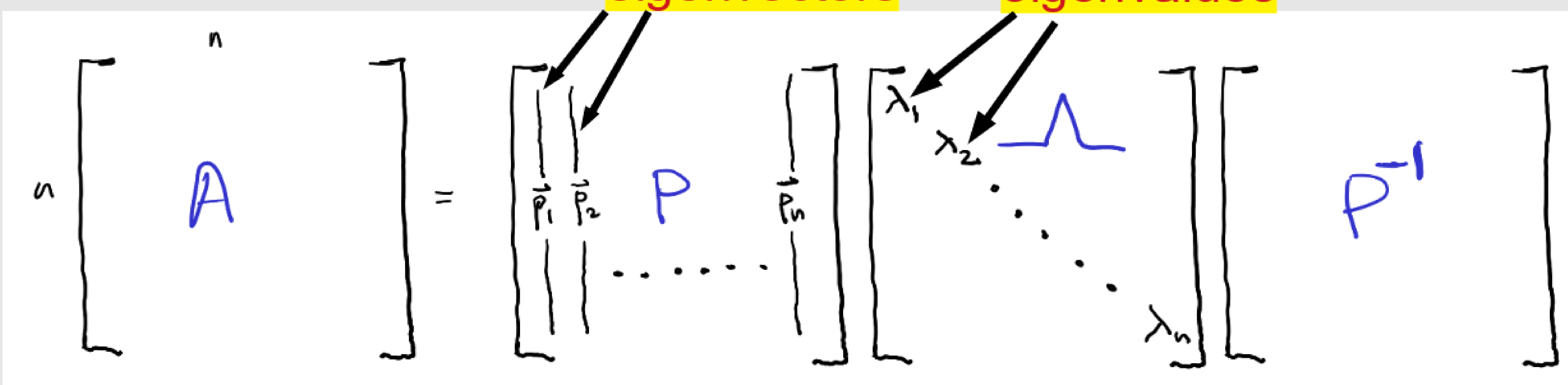
$$A\vec{p}_i = \lambda_i \vec{p}_i$$
$$i = 1, \cdots, n$$

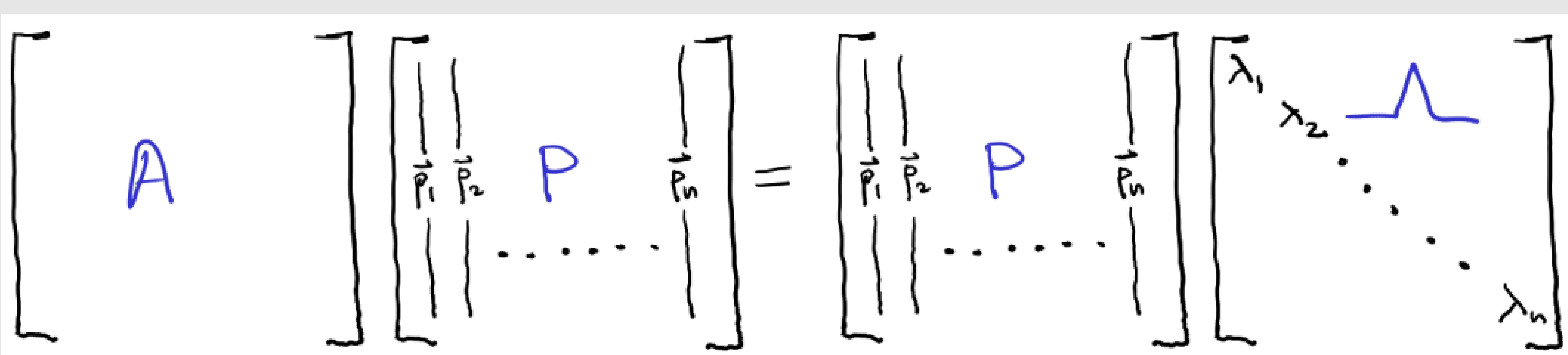
* diagonalization always possible if all eigenvalues distinct (assumed)

The Vector Case: Eigendecomposition

- The vector case: $\frac{d}{dt}\Delta\vec{x}(t) = A\Delta\vec{x}(t) + B\Delta\vec{u}(t)$
  - [already linear(ized); everything is real] real matrices
- Can be "decomposed" into n scalar systems
  - the key idea: to eigendecompose A
- (recap) eigendecomposition: given an nxn matrix A:*

eigenvectors eigenvalues





same thing

$$A\vec{p}_i = \lambda_i \vec{p}_i$$
$$i = 1, \cdots, n$$

* diagonalization always possible if all eigenvalues distinct (assumed)

Eigendecomposition (contd.)

- eigenvalues and determinants
  - $A\vec{p} = \lambda\vec{p}$

Eigendecomposition (contd.)

- eigenvalues and determinants
  - $A\vec{p} = \lambda\vec{p} \Leftrightarrow (A - \lambda I)\,\vec{p} = \vec{0}$

Eigendecomposition (contd.)

- eigenvalues and determinants
  - $A\vec{p} = \lambda\vec{p} \Leftrightarrow \boxed{(A - \lambda I)}\vec{p} = \vec{0}$

must be singular in order to support a non-zero solution for $\vec{p}$

Eigendecomposition (contd.)

- eigenvalues and determinants
  - $A\vec{p} = \lambda\vec{p} \Leftrightarrow \boxed{(A - \lambda I)}\vec{p} = \vec{0}$

    must be singular in order to support a non-zero solution for $\vec{p}$

  - i.e., $det(A - \lambda I) = 0$

Eigendecomposition (contd.)

- eigenvalues and determinants
  - $A\vec{p} = \lambda\vec{p} \Leftrightarrow \boxed{(A - \lambda I)}\vec{p} = \vec{0}$

    must be singular in order to support a non-zero solution for $\vec{p}$

  - i.e., $det(A - \lambda I) = 0$
- $p_A(\lambda) \triangleq det(A - \lambda I) = \lambda^n + c_n\lambda^{n-1} + \cdots + c_2\lambda + c_1$

Eigendecomposition (contd.)

- eigenvalues and determinants
  - $A\vec{p} = \lambda\vec{p} \Leftrightarrow \boxed{(A - \lambda I)}\vec{p} = \vec{0}$

    must be singular in order to support a non-zero solution for $\vec{p}$

  - i.e., $det(A - \lambda I) = 0$

- $p_A(\lambda) \triangleq \boxed{det(A - \lambda I) = \lambda^n + c_n\lambda^{n-1} + \cdots + c_2\lambda + c_1}$

  characteristic polynomial of A

Eigendecomposition (contd.)

- eigenvalues and determinants
  - $A\vec{p} = \lambda\vec{p} \Leftrightarrow \boxed{(A - \lambda I)}\vec{p} = \vec{0}$

    must be singular in order to support a non-zero solution for $\vec{p}$

  - i.e., $det(A - \lambda I) = 0$

- $p_A(\lambda) \triangleq \boxed{det(A - \lambda I) = \lambda^n + c_n\lambda^{n-1} + \cdots + c_2\lambda + c_1}$

  characteristic polynomial of A

- the roots of the char. poly. are the eigenvalues

Eigendecomposition (contd.)

- eigenvalues and determinants
  - $A\vec{p} = \lambda\vec{p} \Leftrightarrow \boxed{(A - \lambda I)}\vec{p} = \vec{0}$

    must be singular in order to support a non-zero solution for $\vec{p}$

  - i.e., $det(A - \lambda I) = 0$
- $p_A(\lambda) \triangleq \boxed{det(A - \lambda I) = \lambda^n + c_n\lambda^{n-1} + \cdots + c_2\lambda + c_1}$

  characteristic polynomial of A

- the roots of the char. poly. are the eigenvalues
  - factorized form: $p_A(\lambda) = (\lambda - \lambda_1)(\lambda - \lambda_2)\cdots(\lambda - \lambda_n)$

Eigendecomposition (contd.)

- eigenvalues and determinants
  - $A\vec{p} = \lambda\vec{p} \Leftrightarrow \boxed{(A - \lambda I)}\vec{p} = \vec{0}$

    must be singular in order to support a non-zero solution for $\vec{p}$
  - i.e., $det(A - \lambda I) = 0$
- $p_A(\lambda) \triangleq \boxed{det(A - \lambda I) = \lambda^n + c_n\lambda^{n-1} + \cdots + c_2\lambda + c_1}$

  characteristic polynomial of A
- the roots of the char. poly. are the eigenvalues
  - factorized form: $p_A(\lambda) = (\lambda - \lambda_1)(\lambda - \lambda_2)\cdots(\lambda - \lambda_n) = 0$

Eigendecomposition (contd.)

- eigenvalues and determinants
  - $A\vec{p} = \lambda\vec{p} \Leftrightarrow \boxed{(A - \lambda I)}\vec{p} = \vec{0}$

    must be singular in order to support a non-zero solution for $\vec{p}$

  - i.e., $det(A - \lambda I) = 0$
- $p_A(\lambda) \triangleq \boxed{det(A - \lambda I) = \lambda^n + c_n\lambda^{n-1} + \cdots + c_2\lambda + c_1}$

  characteristic polynomial of A

- the roots of the char. poly. are the eigenvalues
  - factorized form: $p_A(\lambda) = (\lambda - \lambda_1)(\lambda - \lambda_2)\cdots(\lambda - \lambda_n) = 0$
  - in general, n roots → n eigenvalues $\{\lambda_1, \lambda_2, \ldots, \lambda_n\}$

The Vector Case: Diagonalization

- Applying eigendecomposition: diagonalization
  - → (move to xournal)

— $\dfrac{d\vec{\Delta x}(t)}{dt} = A\,\vec{\Delta x}(t) + B\,\vec{\Delta u}(t)$

— eigendecompose A: $A = P \Lambda P^{-1}$

how? there are standard techniques — eg, in python & MATLAB
— if REALLY interested: take an advanced numerical analysis course.

— $\dfrac{d\vec{\Delta x}(t)}{dt} = P \Lambda P^{-1} \vec{\Delta x}(t) + B\,\vec{\Delta u}(t)$

— or (P is invertible): $P^{-1} \dfrac{d}{dt} \vec{\Delta x}(t) = \Lambda P^{-1} \vec{\Delta x}(t) + P^{-1} B\,\vec{\Delta u}(t)$

— or $\dfrac{d}{dt} \underbrace{\left(P^{-1}\vec{\Delta x}(t)\right)}_{\text{call this } \vec{\Delta y}(t)} = \Lambda \left(P^{-1}\vec{\Delta x}(t)\right) + \left(P^{-1}B\right)\vec{\Delta u}(t)$

call this $\vec{\Delta y}(t)$, i.e, $\vec{\Delta y}(t) \triangleq P^{-1}\vec{\Delta x}(t) \Leftrightarrow \vec{\Delta x}(t) = P\,\vec{\Delta y}(t)$

— $\dfrac{d}{dt}\vec{\Delta y}(t) = \Lambda \vec{\Delta y}(t) + \underbrace{\left(P^{-1}B\right)\vec{\Delta u}(t)}_{\text{call this } \vec{\Delta b}(t)}$

call this $\vec{\Delta b}(t)$, ie, $\vec{\Delta b}(t) = \left(P^{-1}B\right)\vec{\Delta u}(t)$

$$\frac{d}{dt}\begin{bmatrix} \Delta y_1(t) \\ \Delta y_2(t) \\ \vdots \\ \Delta y_n(t) \end{bmatrix} = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix} \begin{bmatrix} \Delta y_1(t) \\ \vdots \\ \Delta y_n(t) \end{bmatrix} + \begin{bmatrix} \Delta b_1(t) \\ \Delta b_2(t) \\ \vdots \\ \Delta b_n(t) \end{bmatrix}.$$

$$\frac{d}{dt}\Delta y_i(t) = \lambda_i \Delta y_i(t) + \Delta b_i(t)$$

The Vector Case: Diagonalization

- Applying eigendecomposition: diagonalization

$$\frac{d}{dt}\begin{bmatrix}\Delta y_1(t)\\ \Delta y_2(t)\\ \vdots\\ \Delta y_n(t)\end{bmatrix} = \begin{bmatrix}\lambda_1 & & \\ & \ddots & \\ & & \lambda_n\end{bmatrix}\begin{bmatrix}\Delta y_1(t)\\ \vdots\\ \Delta y_n(t)\end{bmatrix} + \begin{bmatrix}\Delta b_1(t)\\ \Delta b_2(t)\\ \vdots\\ \Delta b_n(t)\end{bmatrix}$$

$$\frac{d}{dt}\Delta y_i(t) = \lambda_i \Delta y_i(t) + \Delta b_i(t)$$

— $\frac{d\Delta\vec{x}(t)}{dt} = A\,\vec{\Delta x}(t) + B\,\vec{\Delta u}(t)$

— eigendecompose A: $A = P \Lambda P^{-1}$

how? there are standard techniques
— eg, in python & MATLAB
— if REALLY interested: take an advanced numerical analysis course.

— $\frac{d\vec{\Delta x}(t)}{dt} = P \Lambda P^{-1}\vec{\Delta x}(t) + B\,\vec{\Delta u}(t)$

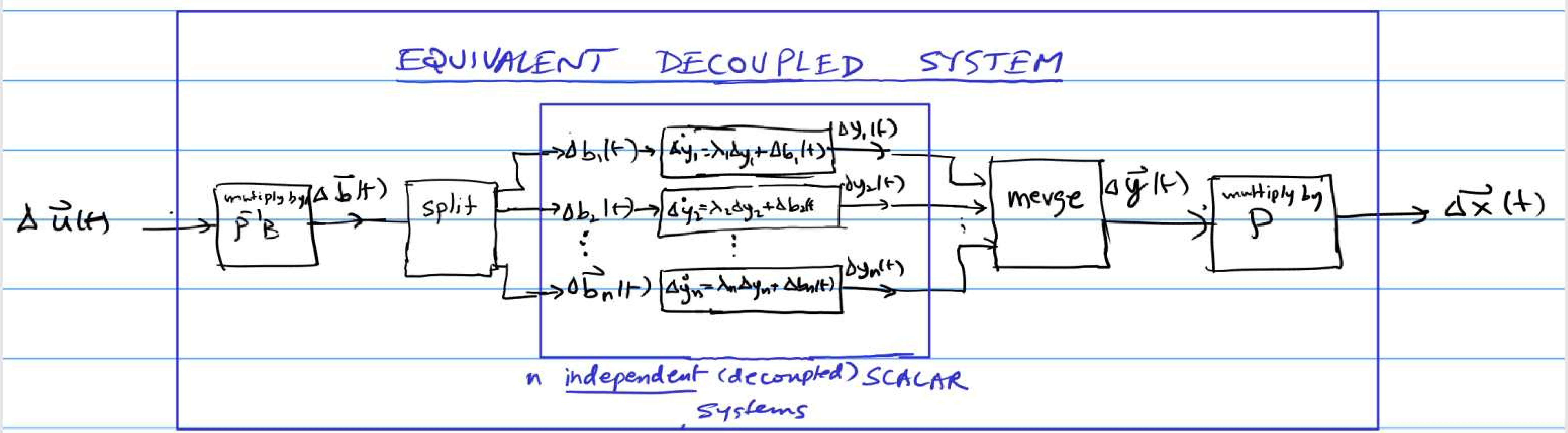
— or (P is invertible): $P^{-1}\frac{d}{dt}\vec{\Delta x}(t) = \Lambda P^{-1}\vec{\Delta x}(t) + P^{-1}B\,\vec{\Delta u}(t)$

— or $\frac{d}{dt}\underbrace{\left(P^{-1}\vec{\Delta x}(t)\right)}_{\text{call this } \vec{\Delta y}(t)} = \Lambda\left(P^{-1}\vec{\Delta x}(t)\right) + \left(P^{-1}B\right)\vec{\Delta u}(t)$

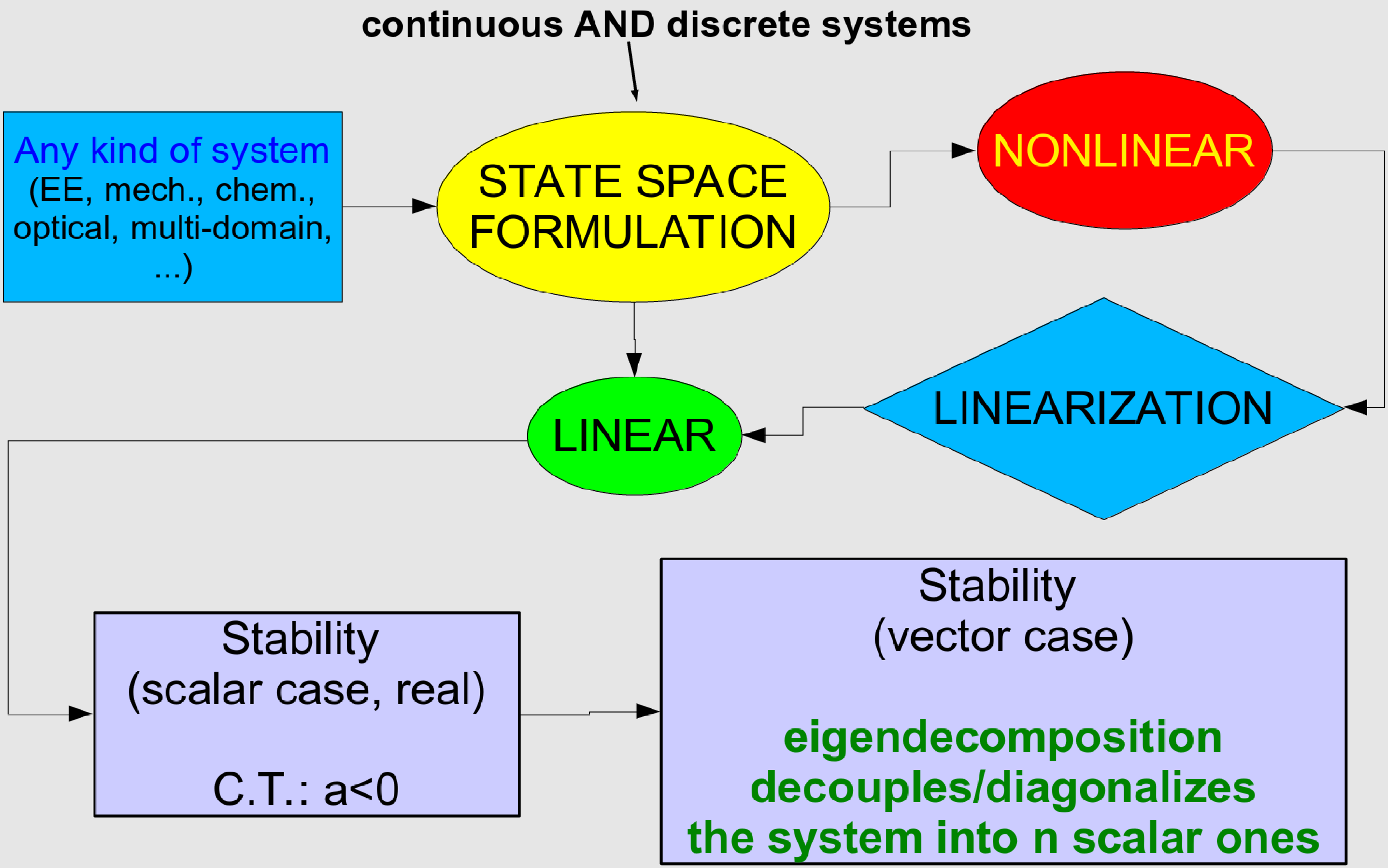
call this $\vec{\Delta y}(t)$, i.e, $\vec{\Delta y}(t) \triangleq P^{-1}\vec{\Delta x}(t) \Leftrightarrow \vec{\Delta x}(t) = P\,\vec{\Delta y}(t)$

— $\frac{d}{dt}\vec{\Delta y}(t) = \Lambda\,\vec{\Delta y}(t) + \underbrace{\left(P^{-1}B\right)\vec{\Delta u}(t)}$

call this $\vec{\Delta b}(t)$, ie, $\vec{\Delta b}(t) = \left(P^{-1}B\right)\vec{\Delta u}(t)$



Where We Were Before



Stability: the Vector Case

- $$\frac{d}{dt}\Delta y_i(t) = \lambda_i \Delta y_i(t) + \Delta b_i(t)$$
  $i = 1, \cdots, n$

Stability: the Vector Case

- $$\frac{d}{dt}\Delta y_i(t) = \lambda_i \Delta y_i(t) + \Delta b_i(t)$$
  $i = 1, \cdots, n$
- System stable if **each system is stable**

Stability: the Vector Case

- $\frac{d}{dt} \Delta y_i(t) = \lambda_i \Delta y_i(t) + \Delta b_i(t)$ $\quad i = 1, \cdots, n$
- System stable if **each system is stable** $\lambda_i < 0$

Stability: the Vector Case

- $\frac{d}{dt} \Delta y_i(t) = \lambda_i \Delta y_i(t) + \Delta b_i(t)$ $\quad i = 1, \cdots, n$ provided λ_i is REAL $\rightarrow$ $\lambda_i < 0$
- System stable if **each system is stable**

Stability: the Vector Case

- $\frac{d}{dt} \Delta y_i(t) = \lambda_i \Delta y_i(t) + \Delta b_i(t)$ provided λ_i is REAL $\rightarrow$ $\lambda_i < 0$

  $i = 1, \cdots, n$
- System stable if **each system is stable**
- **Complication: eigenvalues can be complex**
  - reason: real matrices A can have complex eigen{vals,vecs}

Stability: the Vector Case

- $\frac{d}{dt} \Delta y_i(t) = \lambda_i \Delta y_i(t) + \Delta b_i(t)$ $\quad i = 1, \cdots, n$ $\quad$ provided λ_i is REAL $\rightarrow$ $\lambda_i < 0$
- System stable if **each system is stable**
- **Complication: eigenvalues can be complex**
  - reason: real matrices A can have complex eigen{vals,vecs}
  - examples: (also demo in MATLAB)

→ $$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} = \frac{1}{\sqrt{2}} \overset{P}{\begin{bmatrix} 1 & 1 \\ -j & j \end{bmatrix}} \overset{\Lambda}{\begin{bmatrix} j & \\ & -j \end{bmatrix}} \overset{P^{-1}}{\begin{bmatrix} 1 & j \\ 1 & -j \end{bmatrix}} \frac{1}{\sqrt{2}}$$

→ $$\begin{bmatrix} 0 & 1 \\ -1 & -\sqrt{2} \end{bmatrix} = \overset{P}{\begin{bmatrix} \frac{1+j}{2} & \frac{1-j}{2} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}} \overset{\Lambda}{\begin{bmatrix} \frac{-1+j}{\sqrt{2}} & \\ & \frac{-1-j}{\sqrt{2}} \end{bmatrix}} \overset{P^{-1}}{\begin{bmatrix} -j & \frac{1+j}{\sqrt{2}} \\ j & \frac{1-j}{\sqrt{2}} \end{bmatrix}}$$

Stability: the Vector Case (contd.)

- If A real, eigen{v,v}s come in **complex conjugate pairs**
  - $A\vec{p}_i = \lambda_i \vec{p}_i \Rightarrow \overline{A}\,\overline{\vec{p}_i} = \overline{\lambda_i}\,\overline{\vec{p}_i} \Rightarrow A\,\overline{\vec{p}_i} = \overline{\lambda_i}\,\overline{\vec{p}_i}$

Stability: the Vector Case (contd.)

- If A real, eigen{v,v}s come in **complex conjugate pairs**
  - $A\vec{p}_i = \lambda_i \vec{p}_i \Rightarrow \overline{A}\,\overline{\vec{p}_i} = \overline{\lambda_i}\,\overline{\vec{p}_i} \Rightarrow A\,\overline{\vec{p}_i} = \overline{\lambda_i}\,\overline{\vec{p}_i}$
- Implications (details in handwritten notes)
  - **internal quantities** in the decomposition come in conjugate pairs

Stability: the Vector Case (contd.)

- If A real, eigen{v,v}s come in **complex conjugate pairs**
  - $A\vec{p}_i = \lambda_i \vec{p}_i \Rightarrow \overline{A}\,\overline{\vec{p}_i} = \overline{\lambda_i}\,\overline{\vec{p}_i} \Rightarrow A\,\overline{\vec{p}_i} = \overline{\lambda_i}\,\overline{\vec{p}_i}$
- Implications (details in handwritten notes)
  - **internal quantities** in the decomposition come in conjugate pairs
    - the cols of P $\vec{p}_i$, rows of P^{-1}, $\Delta b_i(t)$, the eigenvalues λ_i, $\Delta y_i(t)$

Stability: the Vector Case (contd.)

- If A real, eigen{v,v}s come in **complex conjugate pairs**
  - $A\vec{p}_i = \lambda_i \vec{p}_i \Rightarrow \overline{A}\,\overline{\vec{p}_i} = \overline{\lambda_i}\,\overline{\vec{p}_i} \Rightarrow A\,\overline{\vec{p}_i} = \overline{\lambda_i}\,\overline{\vec{p}_i}$
- Implications (details in handwritten notes)
  - **internal quantities** in the decomposition come in conjugate pairs
    - the cols of P $\vec{p}_i$, rows of P^{-1}, $\Delta b_i(t)$, the eigenvalues λ_i, $\Delta y_i(t)$

$$\Delta\vec{x}(t) = P\Delta\vec{y}(t) = \sum_{i=1}^{n} \vec{p}_i \Delta y_i(t)$$

$$= \vec{p}_1\Delta y_1(t) + \vec{p}_2\Delta y_2(t) + \vec{p}_3\Delta y_3(t) + \cdots + \vec{p}_n\Delta y_n(t)$$

Stability: the Vector Case (contd.)

- If A real, eigen{v,v}s come in **complex conjugate pairs**
  - $A\vec{p}_i = \lambda_i \vec{p}_i \Rightarrow \overline{A}\,\overline{\vec{p}_i} = \overline{\lambda_i}\,\overline{\vec{p}_i} \Rightarrow A\,\overline{\vec{p}_i} = \overline{\lambda_i}\,\overline{\vec{p}_i}$
- Implications (details in handwritten notes)
  - **internal quantities** in the decomposition come in conjugate pairs
    - the cols of P $\vec{p}_i$, rows of P^{-1}, $\Delta b_i(t)$, the eigenvalues λ_i, $\Delta y_i(t)$

$$\Delta\vec{x}(t) = P\Delta\vec{y}(t) = \sum_{i=1}^{n} \vec{p}_i \Delta y_i(t)$$

real (say)

real (say)

$$= \vec{p}_1 \Delta y_1(t) + \vec{p}_2 \Delta y_2(t) + \vec{p}_3 \Delta y_3(t) + \cdots + \vec{p}_n \Delta y_n(t)$$

Stability: the Vector Case (contd.)

- If A real, eigen{v,v}s come in **complex conjugate pairs**
  - $A\vec{p}_i = \lambda_i \vec{p}_i \Rightarrow \overline{A}\,\overline{\vec{p}_i} = \overline{\lambda_i}\,\overline{\vec{p}_i} \Rightarrow A\,\overline{\vec{p}_i} = \overline{\lambda_i}\,\overline{\vec{p}_i}$
- Implications (details in handwritten notes)
  - **internal quantities** in the decomposition come in conjugate pairs
    - the cols of P $\vec{p}_i$, rows of P^{-1}, $\Delta b_i(t)$, the eigenvalues λ_i, $\Delta y_i(t)$

$$\Delta\vec{x}(t) = P\Delta\vec{y}(t) = \sum_{i=1}^{n} \vec{p}_i \Delta y_i(t)$$

complex conjugate pair (say): sum is real

real (say)

real (say)

$$= \vec{p}_1 \Delta y_1(t) + \vec{p}_2 \Delta y_2(t) + \vec{p}_3 \Delta y_3(t) + \cdots + \vec{p}_n \Delta y_n(t)$$

Stability: the Vector Case (contd.)

- If A real, eigen{v,v}s come in **complex conjugate pairs**
  - $A\vec{p}_i = \lambda_i \vec{p}_i \Rightarrow \overline{A}\,\overline{\vec{p}_i} = \overline{\lambda_i}\,\overline{\vec{p}_i} \Rightarrow A\,\overline{\vec{p}_i} = \overline{\lambda_i}\,\overline{\vec{p}_i}$
- Implications (details in handwritten notes)
  - **internal quantities** in the decomposition come in conjugate pairs
    - the cols of P $\vec{p}_i$, rows of P^{-1}, $\Delta b_i(t)$, the eigenvalues λ_i, $\Delta y_i(t)$

$$\Delta\vec{x}(t) = P\Delta\vec{y}(t) = \sum_{i=1}^{n} \vec{p}_i \Delta y_i(t)$$

$$= \vec{p}_1 \Delta y_1(t) + \vec{p}_2 \Delta y_2(t) + \vec{p}_3 \Delta y_3(t) + \cdots + \vec{p}_n \Delta y_n(t)$$

always real

real (say)

complex conjugate pair (say): sum is real

real (say)

Stability: the Vector Case (contd.)

- If A real, eigen{v,v}s come in **complex conjugate pairs**
  - $A\vec{p}_i = \lambda_i \vec{p}_i \Rightarrow \overline{A}\,\overline{\vec{p}_i} = \overline{\lambda_i}\,\overline{\vec{p}_i} \Rightarrow A\,\overline{\vec{p}_i} = \overline{\lambda_i}\,\overline{\vec{p}_i}$
- Implications (details in handwritten notes)
  - **internal quantities** in the decomposition come in conjugate pairs
    - the cols of P $\vec{p}_i$, rows of P^{-1}, $\Delta b_i(t)$, the eigenvalues λ_i, $\Delta y_i(t)$

$$\Delta\vec{x}(t) = P\Delta\vec{y}(t) = \sum_{i=1}^{n} \vec{p}_i \Delta y_i(t)$$

always real

real (say)

complex conjugate pair (say): sum is real

real (say)

$$= \vec{p}_1\Delta y_1(t) + \vec{p}_2\Delta y_2(t) + \vec{p}_3\Delta y_3(t) + \cdots + \vec{p}_n\Delta y_n(t)$$

$$\boxed{\Delta y_1(0)e^{\lambda_1 t} + \int_0^t e^{\lambda_1(t-\tau)}\ \Delta b_1(\tau)\,d\tau}$$

just like the real scalar case

Stability: the Vector Case (contd.)

- If A real, eigen{v,v}s come in **complex conjugate pairs**
  - $A\vec{p}_i = \lambda_i \vec{p}_i \Rightarrow \overline{A}\,\overline{\vec{p}_i} = \overline{\lambda_i}\,\overline{\vec{p}_i} \Rightarrow A\,\overline{\vec{p}_i} = \overline{\lambda_i}\,\overline{\vec{p}_i}$
- Implications (details in handwritten notes)
  - **internal quantities** in the decomposition come in conjugate pairs
    - the cols of P $\vec{p}_i$, rows of P^{-1}, $\Delta b_i(t)$, the eigenvalues λ_i, $\Delta y_i(t)$

$$\Delta\vec{x}(t) = P\Delta\vec{y}(t) = \sum_{i=1}^{n} \vec{p}_i \Delta y_i(t)$$

$$= \vec{p}_1 \Delta y_1(t) + \underbrace{\vec{p}_2 \Delta y_2(t) + \vec{p}_3 \Delta y_3(t)} + \cdots + \vec{p}_n \Delta y_n(t)$$

always real

real (say)

complex conjugate pair (say): sum is real

real (say)

$$\boxed{\Delta y_1(0)e^{\lambda_1 t} + \int_0^t e^{\lambda_1(t-\tau)}\ \Delta b_1(\tau)\, d\tau}$$

just like the real scalar case

j^{th} component of vector

derived from $\Delta y_2(0)$ and $\vec{p}_2$

$\lambda_r + \jmath\lambda_i = \lambda_2 = \overline{\lambda_3}$

$$e^{\lambda_r t}\left[\cos(\lambda_i t)\Delta\tilde{y}_{ij}(0) + \sin(\lambda_i t)\Delta\hat{y}_{ij}(0) + \right.$$

$$\left\{e^{\lambda_r t}\cos(\lambda_i t)\right\} * \left\{\Delta\tilde{b}_{ij}(t)\right\} + \left\{e^{\lambda_r t}\sin(\lambda_i t)\right\} * \left\{\Delta\hat{b}_{ij}(t)\right\}$$

derived from $\Delta b_2(t)$ and $\vec{p}_2$

Stability: the Vector Case (contd.)

- If A real, eigen{v,v}s come in **complex conjugate pairs**
  - $A\vec{p}_i = \lambda_i \vec{p}_i \Rightarrow \overline{A}\,\overline{\vec{p}_i} = \overline{\lambda_i}\,\overline{\vec{p}_i} \Rightarrow A\,\overline{\vec{p}_i} = \overline{\lambda_i}\,\overline{\vec{p}_i}$
- Implications (details in handwritten notes)
  - **internal quantities** in the decomposition come in conjugate pairs
    - the cols of P $\vec{p}_i$, rows of P^{-1}, $\Delta b_i(t)$, the eigenvalues λ_i, $\Delta y_i(t)$

$$\Delta\vec{x}(t) = P\Delta\vec{y}(t) = \sum_{i=1}^{n} \vec{p}_i \Delta y_i(t)$$

always real

real (say)

complex conjugate pair (say): sum is real

real (say)

$$= \vec{p}_1 \Delta y_1(t) + \underbrace{\vec{p}_2 \Delta y_2(t) + \vec{p}_3 \Delta y_3(t)}_{} + \cdots + \vec{p}_n \Delta y_n(t)$$

$$\boxed{\Delta y_1(0) e^{\lambda_1 t} + \int_0^t e^{\lambda_1 (t-\tau)}\, \Delta b_1(\tau)\, d\tau}$$

just like the real scalar case

j^{th} component of vector

derived from $\Delta y_2(0)$ and $\vec{p}_2$

$\lambda_r + \jmath\lambda_i = \lambda_2 = \overline{\lambda_3}$

IC term

$$e^{\lambda_r t}\left[\cos(\lambda_i t)\Delta\tilde{y}_{ij}(0) + \sin(\lambda_i t)\Delta\hat{y}_{ij}(0)\right] +$$

$$\left\{e^{\lambda_r t}\cos(\lambda_i t)\right\} * \left\{\Delta\tilde{b}_{ij}(t)\right\} + \left\{e^{\lambda_r t}\sin(\lambda_i t)\right\} * \left\{\Delta\hat{b}_{ij}(t)\right\}$$

derived from $\Delta b_2(t)$ and $\vec{p}_2$

Stability: the Vector Case (contd.)

- If A real, eigen{v,v}s come in **complex conjugate pairs**
  - $A\vec{p}_i = \lambda_i \vec{p}_i \Rightarrow \overline{A}\,\overline{\vec{p}_i} = \overline{\lambda_i}\,\overline{\vec{p}_i} \Rightarrow A\,\overline{\vec{p}_i} = \overline{\lambda_i}\,\overline{\vec{p}_i}$
- Implications (details in handwritten notes)
  - **internal quantities** in the decomposition come in conjugate pairs
    - the cols of P $\vec{p}_i$, rows of P^{-1}, $\Delta b_i(t)$, the eigenvalues λ_i, $\Delta y_i(t)$

$$\Delta\vec{x}(t) = P\Delta\vec{y}(t) = \sum_{i=1}^{n} \vec{p}_i \Delta y_i(t)$$

always real

real (say)

complex conjugate pair (say): sum is real

real (say)

$$= \vec{p}_1\Delta y_1(t) + \underbrace{\vec{p}_2\Delta y_2(t) + \vec{p}_3\Delta y_3(t)}_{} + \cdots + \vec{p}_n\Delta y_n(t)$$

$$\boxed{\Delta y_1(0)e^{\lambda_1 t} + \int_0^t e^{\lambda_1(t-\tau)}\,\Delta b_1(\tau)\,d\tau}$$

just like the real scalar case

j^{th} component of vector

derived from $\Delta y_2(0)$ and $\vec{p}_2$

$\lambda_r + \jmath\lambda_i = \lambda_2 = \overline{\lambda_3}$

IC term

$$e^{\lambda_r t}\left[\cos(\lambda_i t)\Delta\tilde{y}_{ij}(0) + \sin(\lambda_i t)\Delta\hat{y}_{ij}(0)\right] +$$

$$\left\{e^{\lambda_r t}\cos(\lambda_i t)\right\} * \left\{\Delta\tilde{b}_{ij}(t)\right\} + \left\{e^{\lambda_r t}\sin(\lambda_i t)\right\} * \left\{\Delta\hat{b}_{ij}(t)\right\}$$

input term **(convolution)**

derived from $\Delta b_2(t)$ and $\vec{p}_2$

Stability: the Vector Case (contd. - 2)

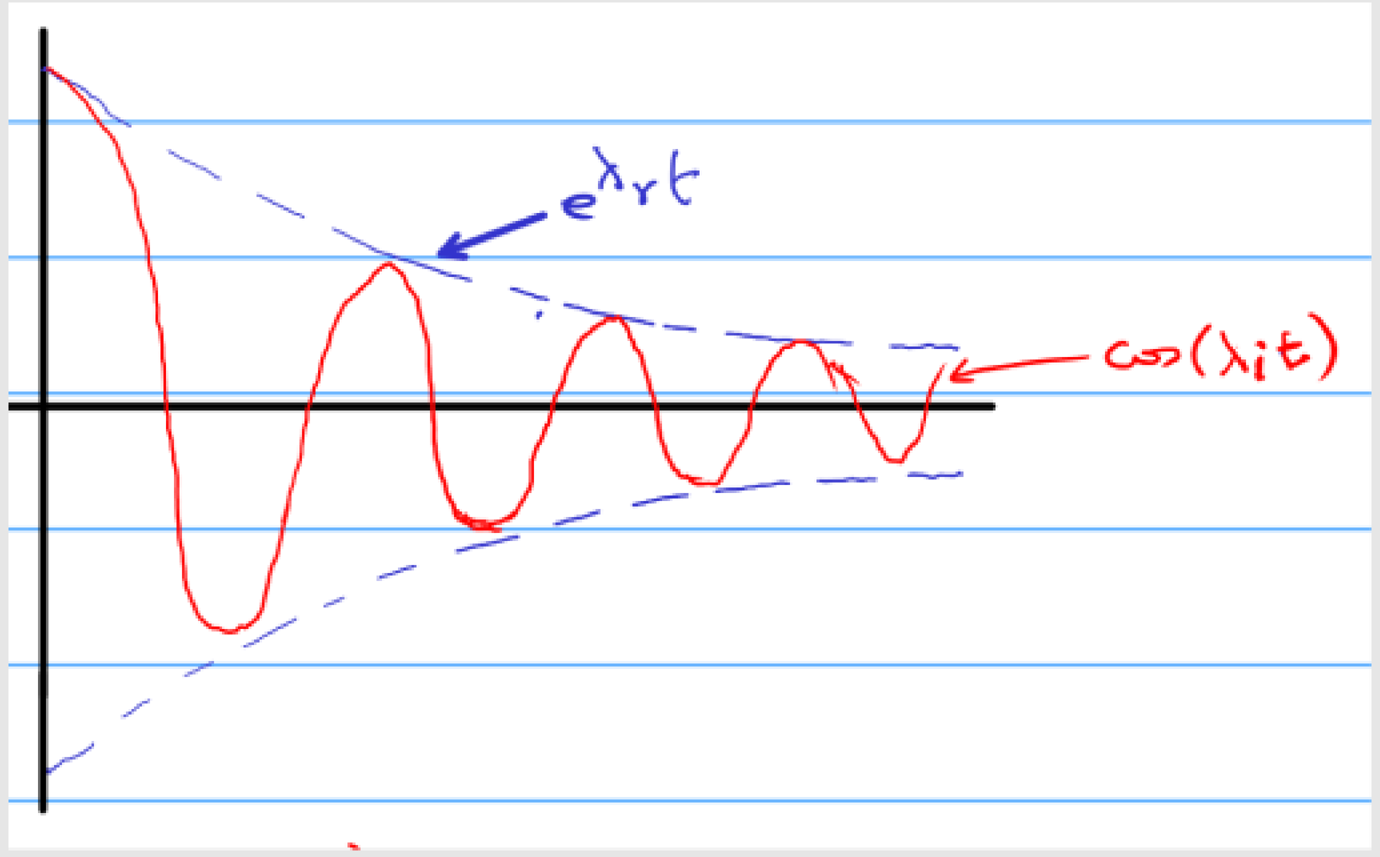
- Initial condition terms: $e^{\lambda_r t}\left[\cos(\lambda_i t)\Delta\tilde{y}_{ij}(0) + \sin(\lambda_i t)\Delta\hat{y}_{ij}(0)\right]$

Stability: the Vector Case (contd. - 2)

- Initial condition terms: $e^{\lambda_r t}\left[\cos(\lambda_i t)\Delta\tilde{y}_{ij}(0) + \sin(\lambda_i t)\Delta\hat{y}_{ij}(0)\right]$

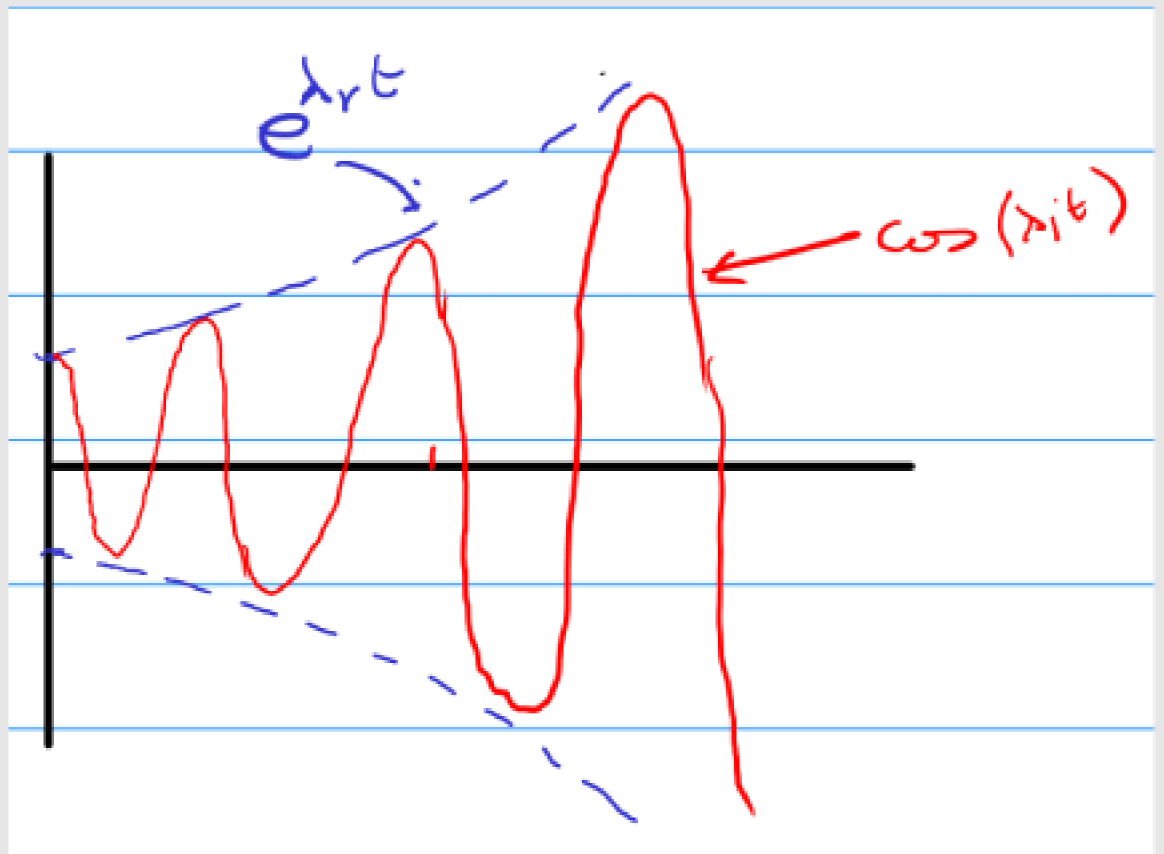
λ_r<0: envelope dies down

STABLE



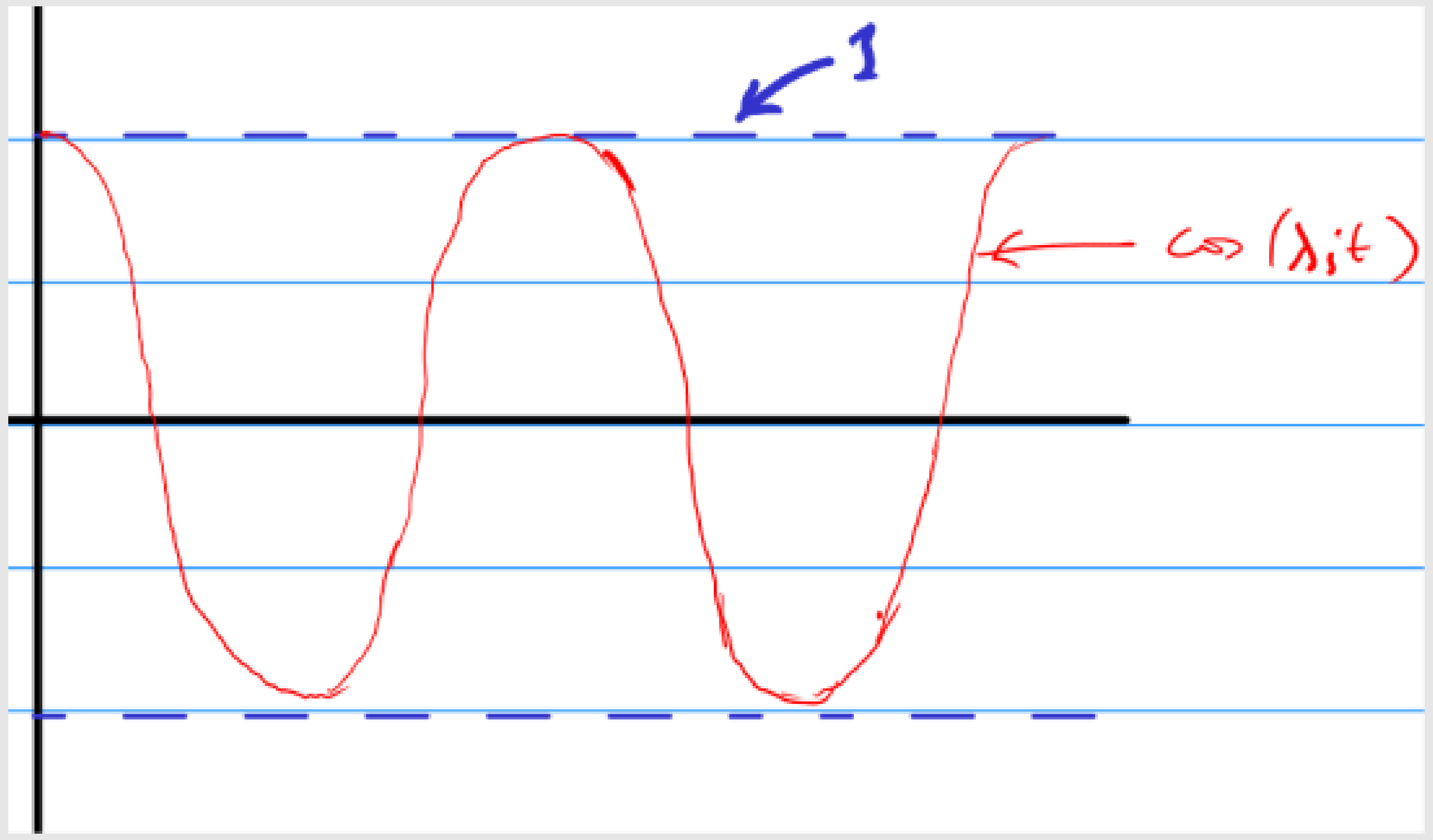
λ_r>0: envelope blows up

UNSTABLE



λ_r=0: const. envelope

MARGINALLY STABLE



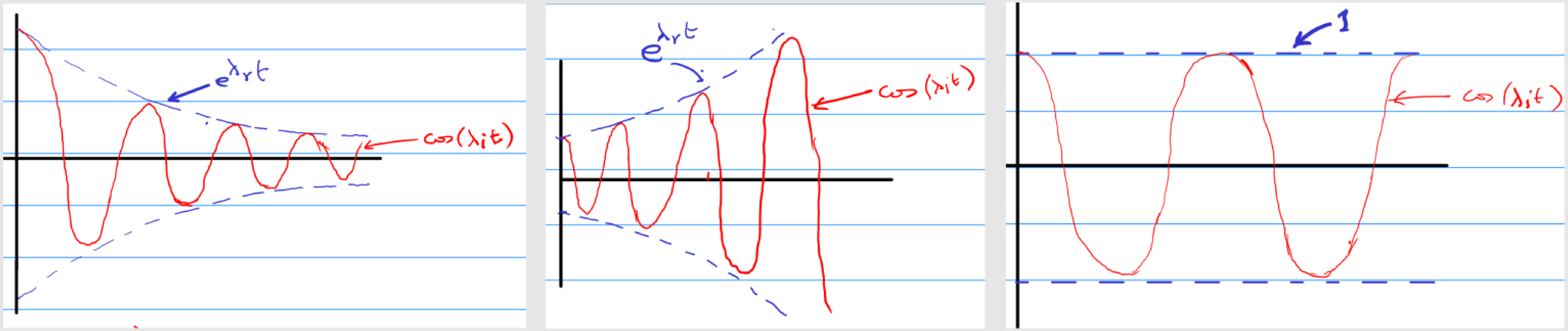
Stability: the Vector Case (contd. - 2)

- Initial condition terms: $e^{\lambda_r t}\left[\cos(\lambda_i t)\Delta\tilde{y}_{ij}(0) + \sin(\lambda_i t)\Delta\hat{y}_{ij}(0)\right]$

$\lambda_r<0$: envelope dies down — **STABLE**

$\lambda_r>0$: envelope blows up — **UNSTABLE**

$\lambda_r=0$: const. envelope — **MARGINALLY STABLE**

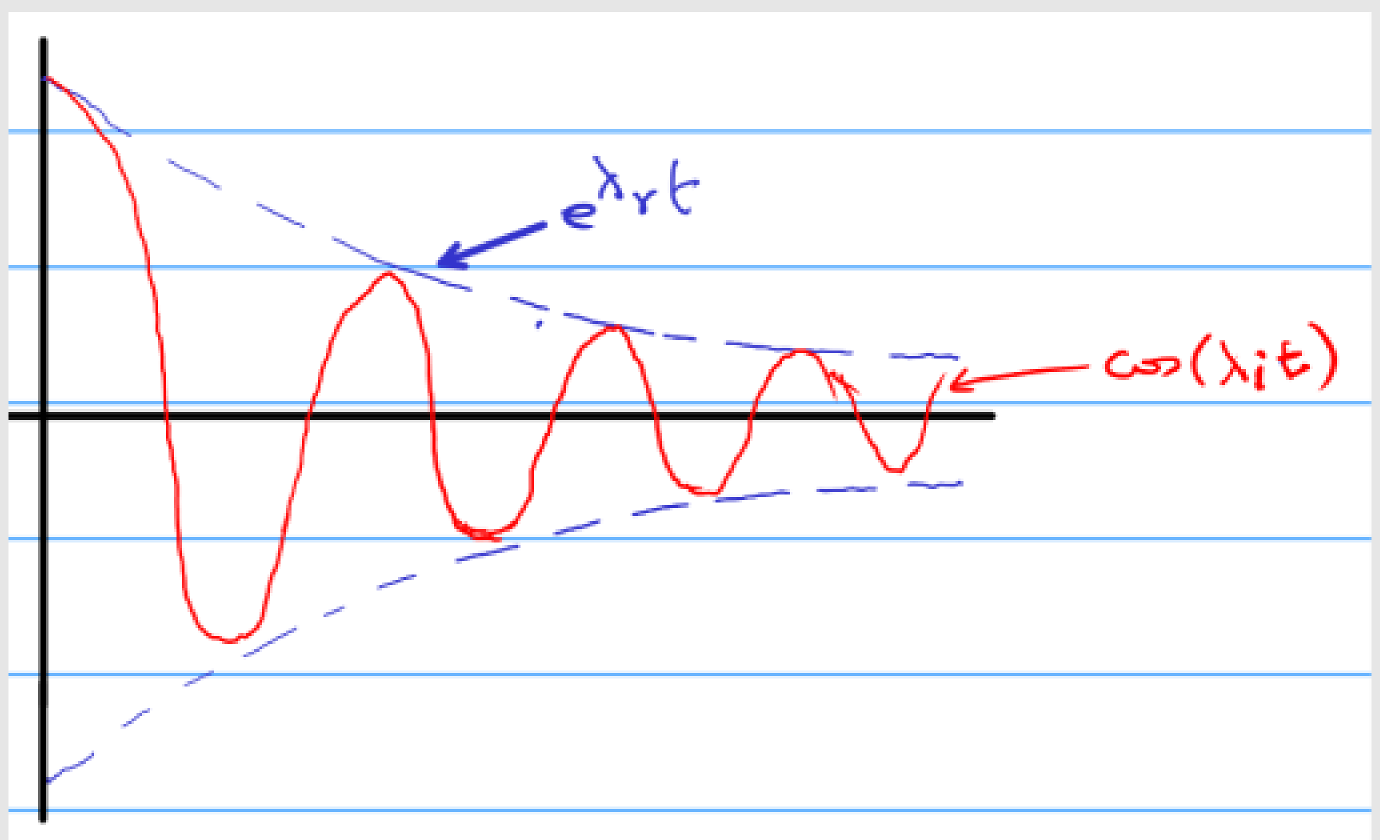


- Input conv. terms: $\left\{e^{\lambda_r t}\cos(\lambda_i t)\right\} * \left\{\Delta\tilde{b}_{ij}(t)\right\} + \left\{e^{\lambda_r t}\sin(\lambda_i t)\right\} * \left\{\Delta\hat{b}_{ij}(t)\right\}$

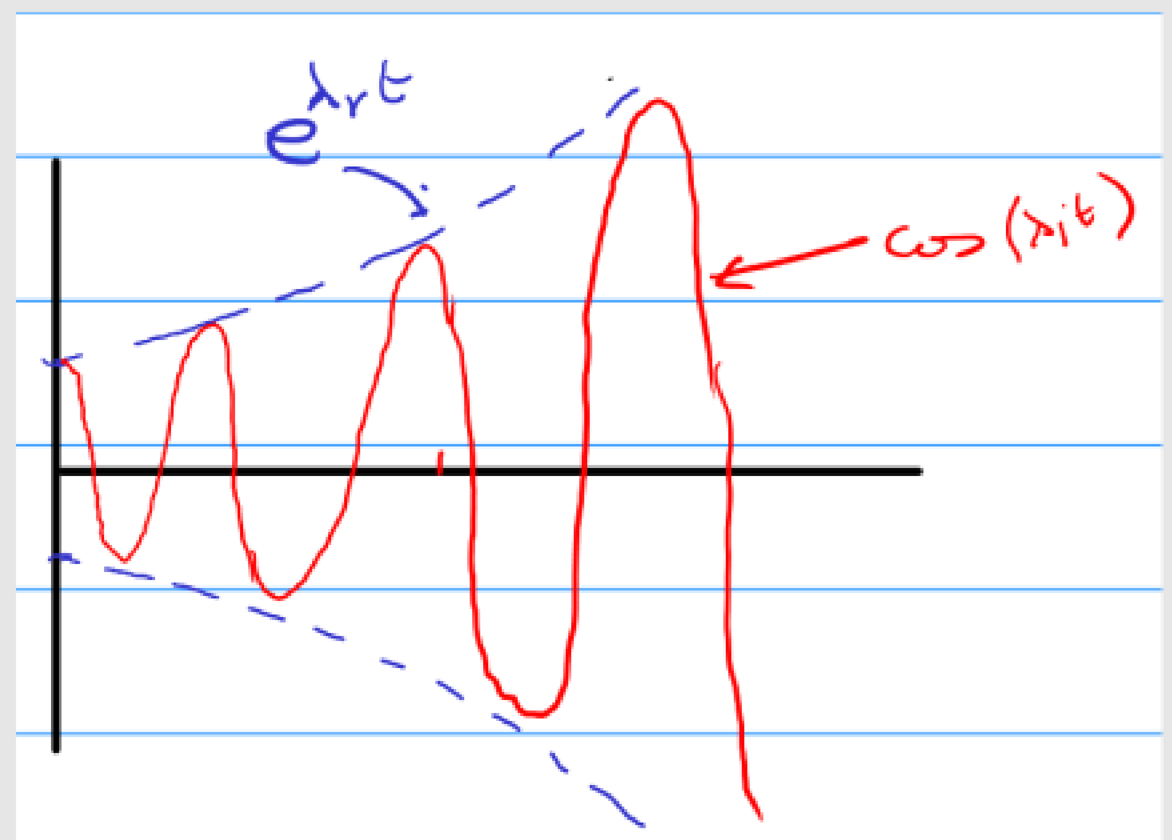
Stability: the Vector Case (contd. - 2)

- Initial condition terms: $e^{\lambda_r t}\left[\cos(\lambda_i t)\Delta\tilde{y}_{ij}(0) + \sin(\lambda_i t)\Delta\hat{y}_{ij}(0)\right]$

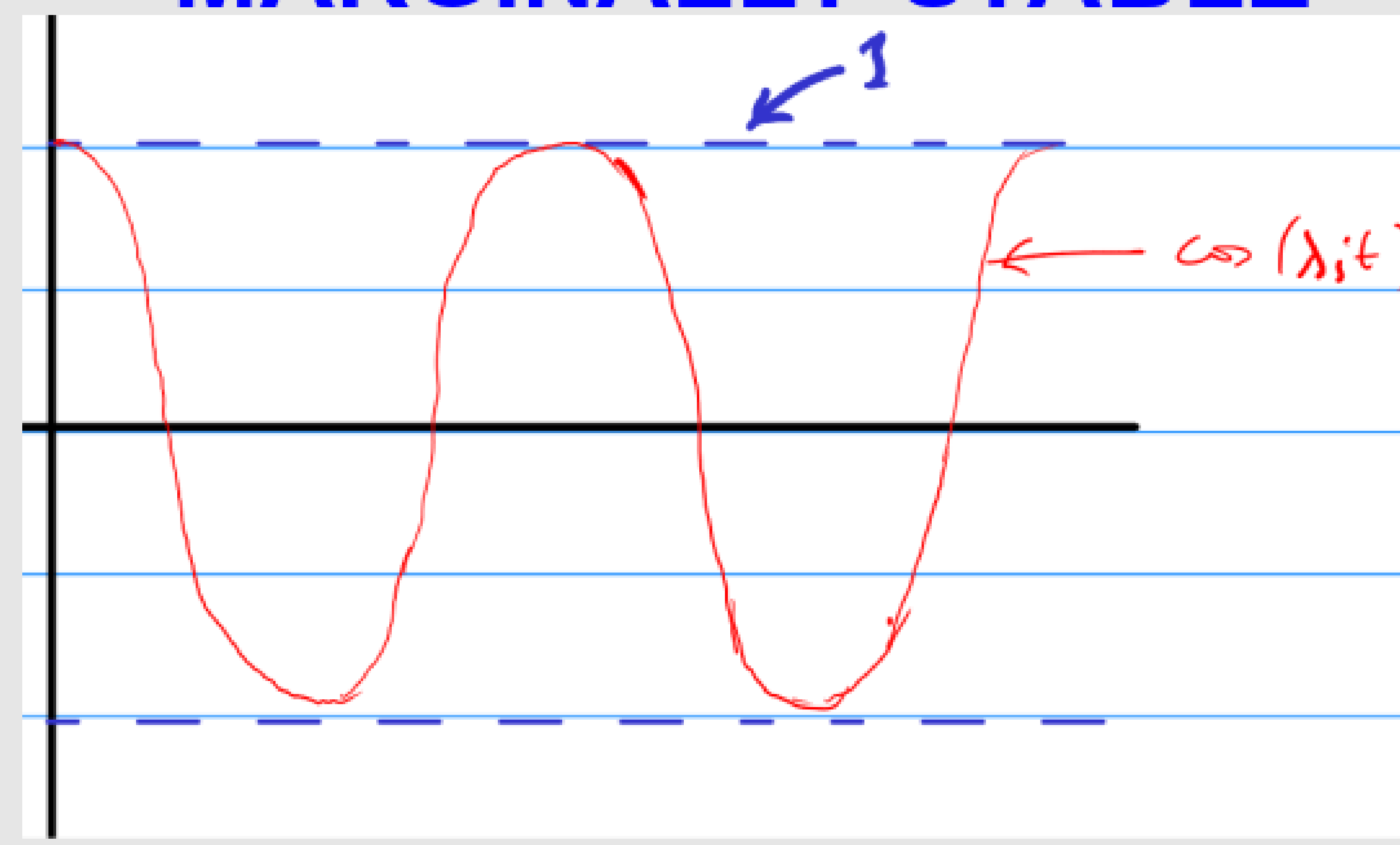
$\lambda_r<0$: envelope dies down **STABLE**



$\lambda_r>0$: envelope blows up **UNSTABLE**



$\lambda_r=0$: const. envelope **MARGINALLY STABLE**



- Input conv. terms: $\{e^{\lambda_r t}\cos(\lambda_i t)\} * \left\{\Delta\tilde{b}_{ij}(t)\right\} + \{e^{\lambda_r t}\sin(\lambda_i t)\} * \left\{\Delta\hat{b}_{ij}(t)\right\}$
  - can show (see notes) that:
    - if $\lambda_r<0$: $\Delta\vec{x}(t)$ bounded if $\Delta u(t)$ bounded: **BIBO stable**
    - if $\lambda_r>0$: $\Delta\vec{x}(t)$ unbounded even if $\Delta u(t)$ bounded: **UNSTABLE**
    - if $\lambda_r=0$: $\Delta\vec{x}(t)$ unbounded even if $\Delta u(t)$ bounded: **UNSTABLE**

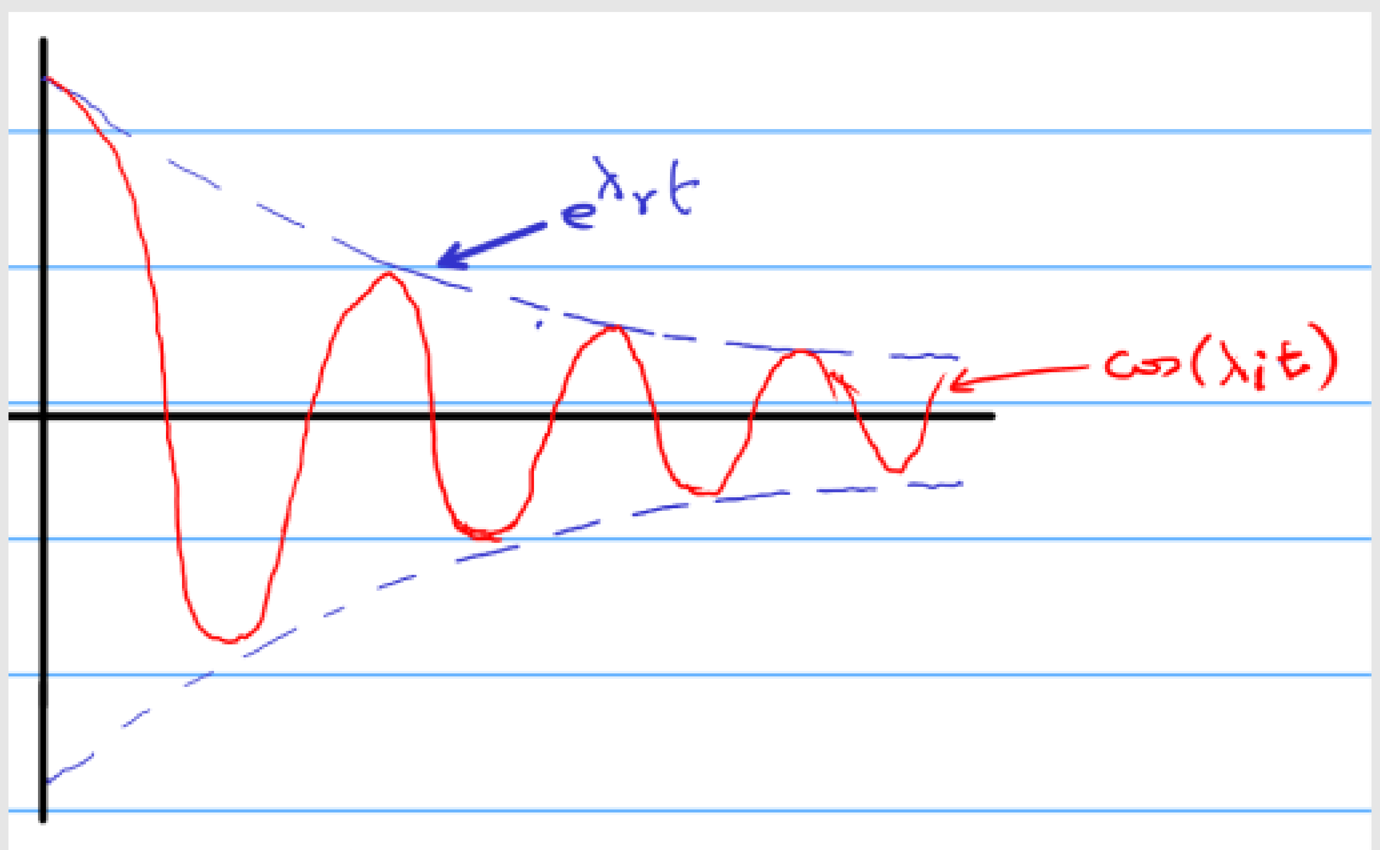
Stability: the Vector Case (contd. - 2)

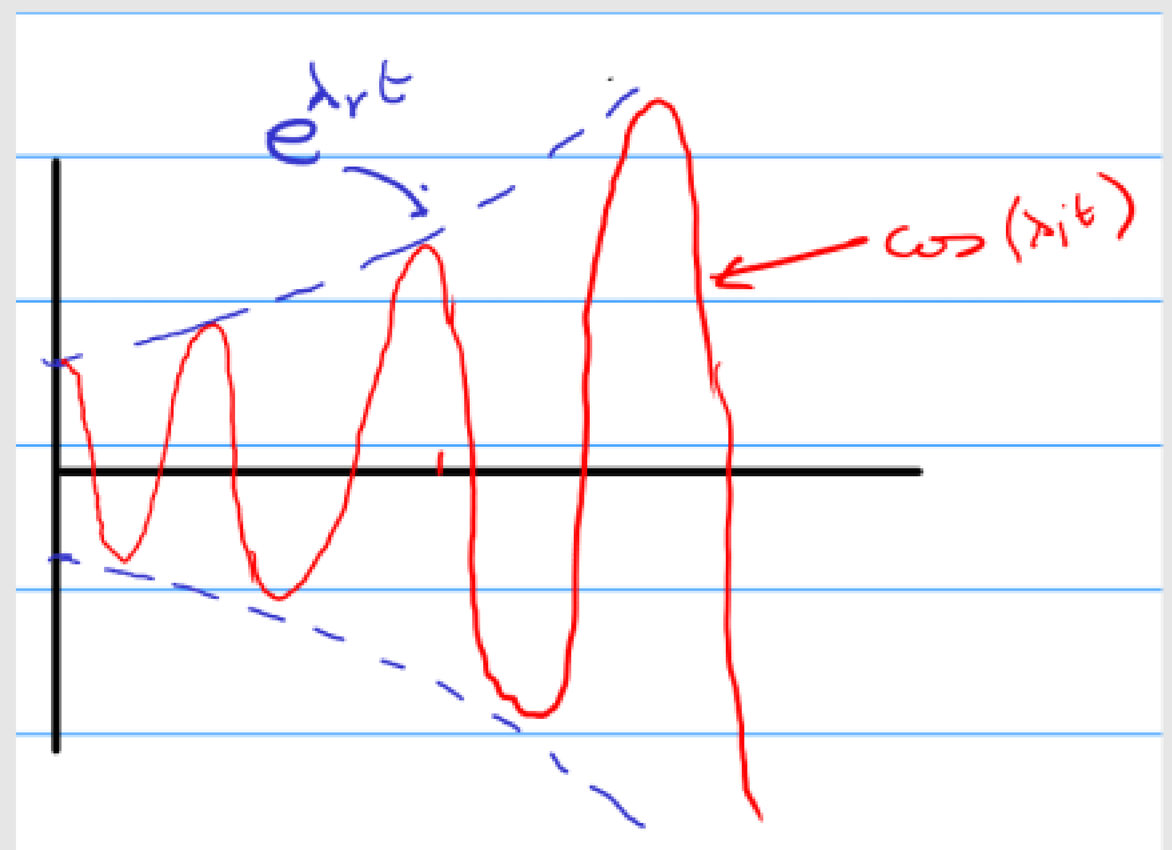
- Initial condition terms: $e^{\lambda_r t}\left[\cos(\lambda_i t)\Delta\tilde{y}_{ij}(0) + \sin(\lambda_i t)\Delta\hat{y}_{ij}(0)\right]$

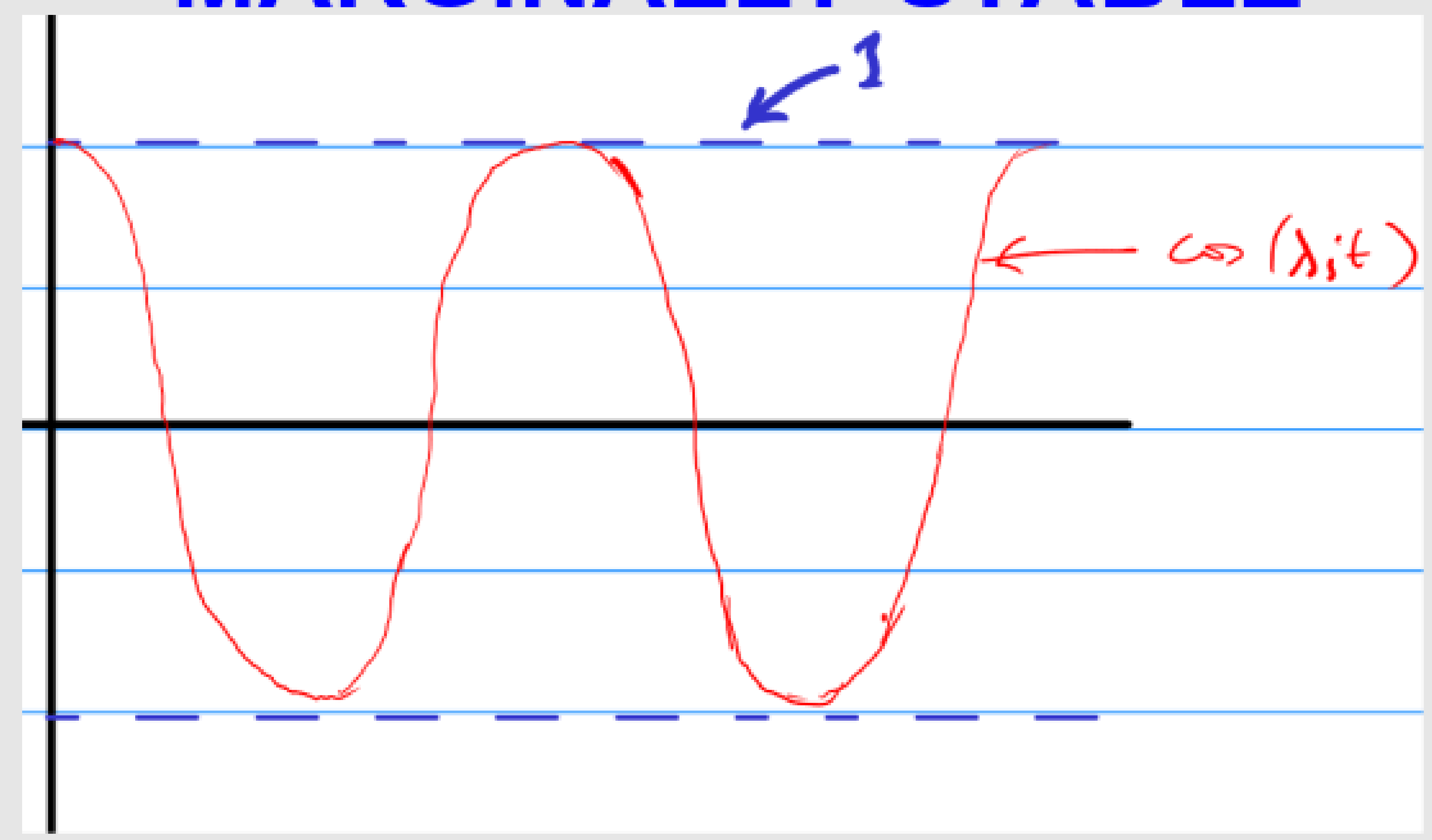
λ_r<0: envelope dies down — **STABLE**

λ_r>0: envelope blows up — **UNSTABLE**

λ_r=0: const. envelope — **MARGINALLY STABLE**



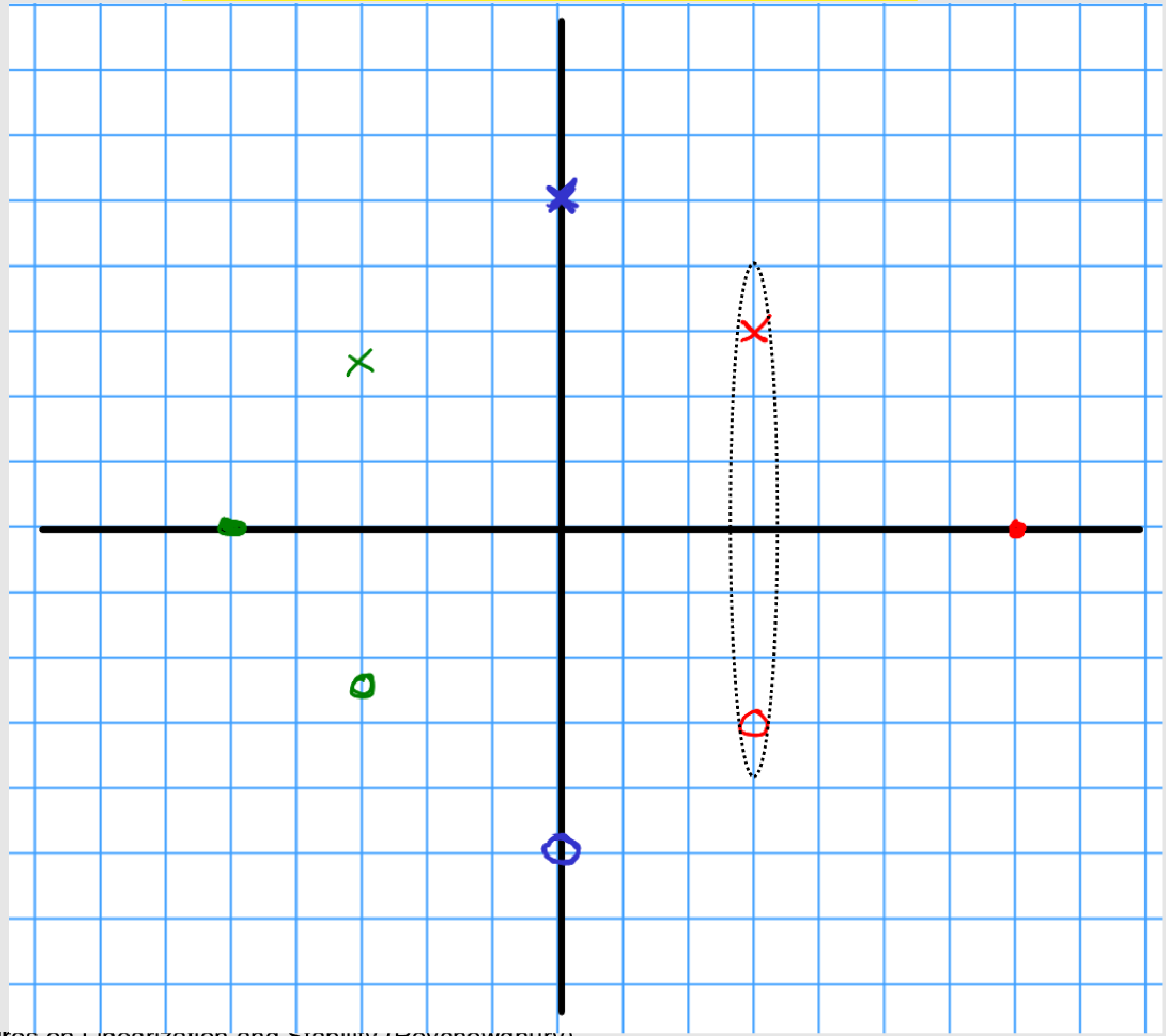




- Input conv. terms: $\left\{e^{\lambda_r t}\cos(\lambda_i t)\right\} * \left\{\Delta\tilde{b}_{ij}(t)\right\} + \left\{e^{\lambda_r t}\sin(\lambda_i t)\right\} * \left\{\Delta\hat{b}_{ij}(t)\right\}$
  - can show (see notes) that: **same as for real eigenvalues, but using the real parts of complex eigenvalues**
    - → if λ_r<0: $\Delta\vec{x}(t)$ bounded if Δu(t) bounded: **BIBO stable**
    - → if λ_r>0: $\Delta\vec{x}(t)$ unbounded even if Δu(t) bounded: **UNSTABLE**
    - → if λ_r=0: $\Delta\vec{x}(t)$ unbounded even if Δu(t) bounded: **UNSTABLE**

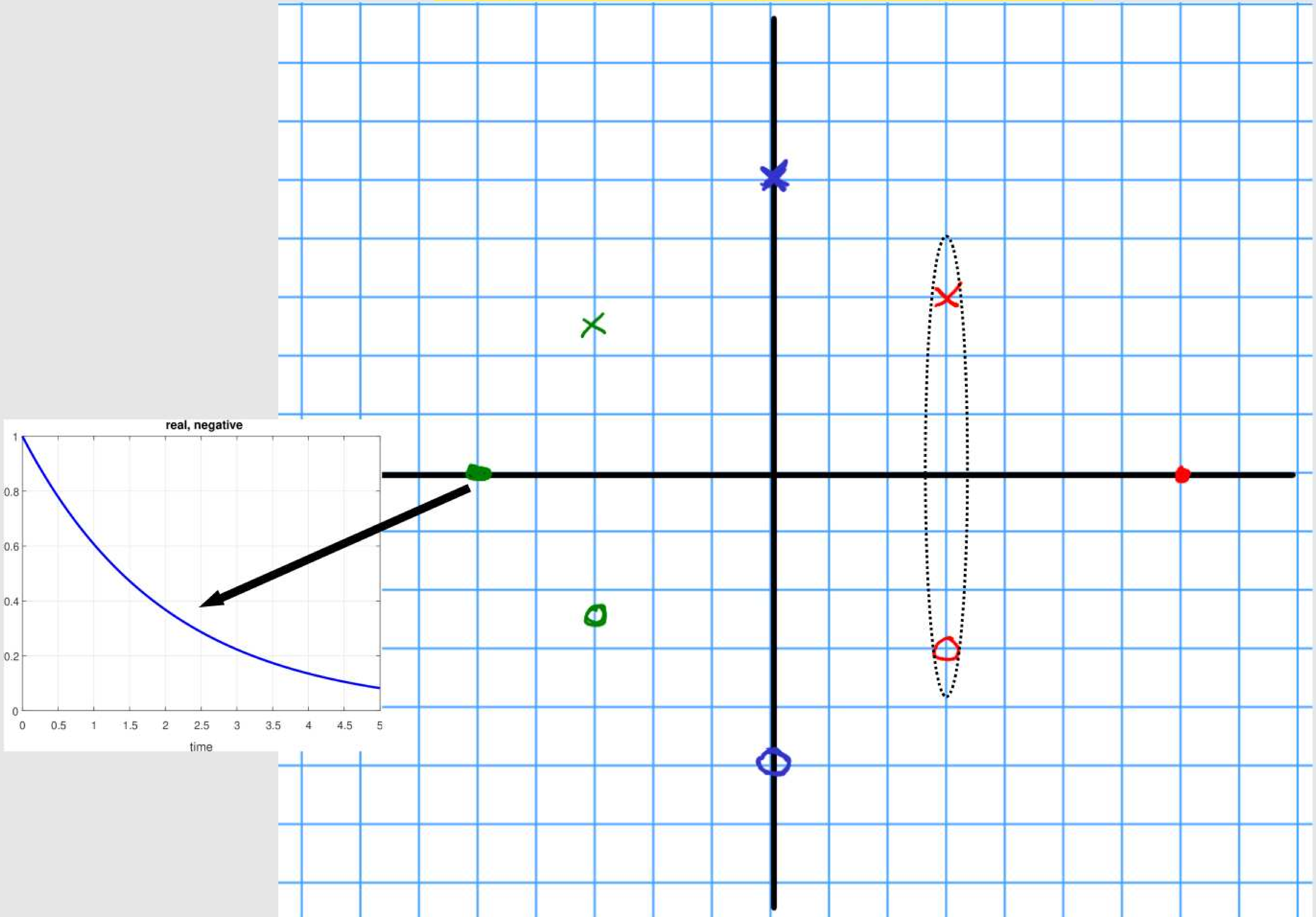
Eigenvalues and Responses (continuous)

complex plane for plotting eigenvalues



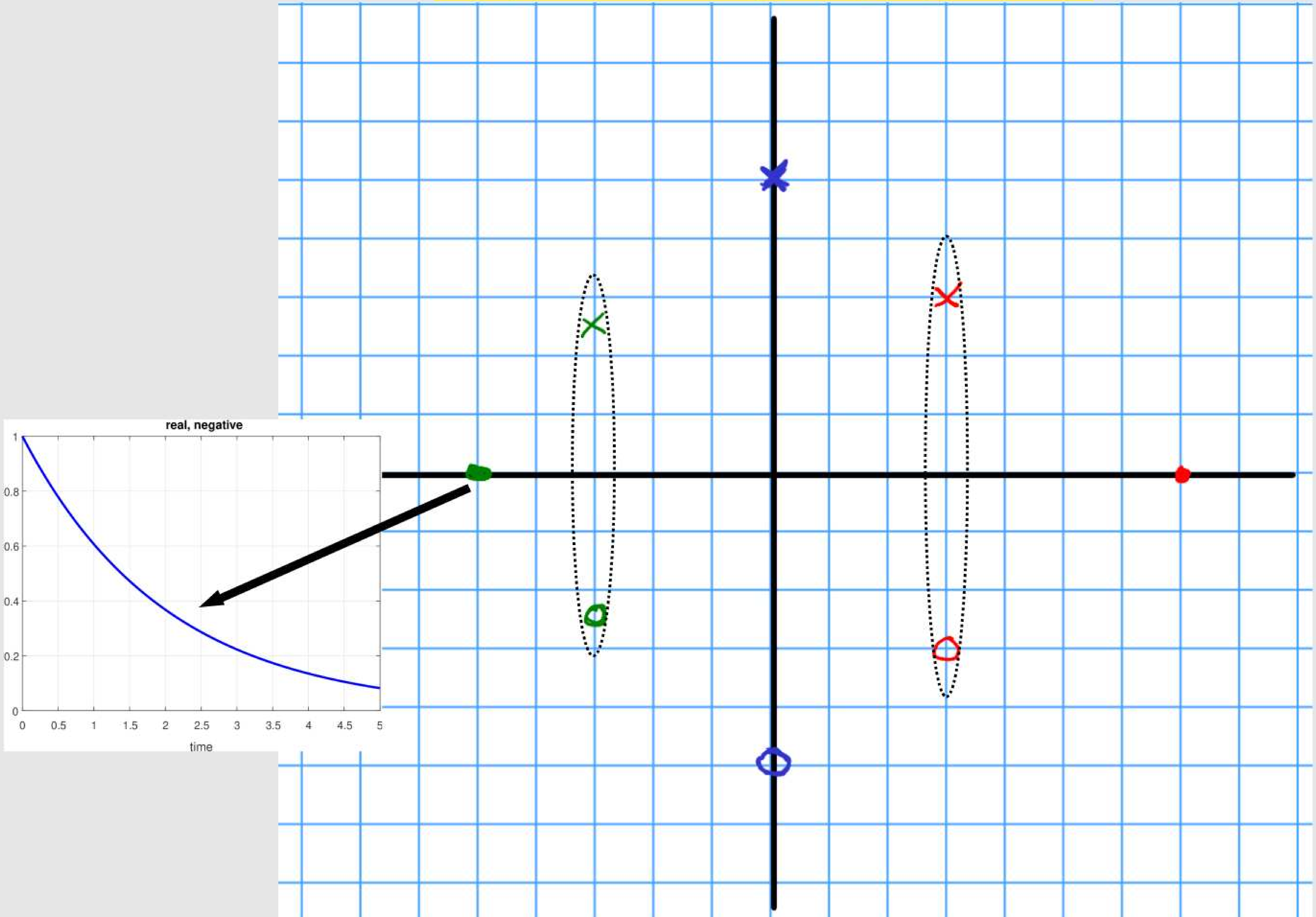
Eigenvalues and Responses (continuous)

complex plane for plotting eigenvalues

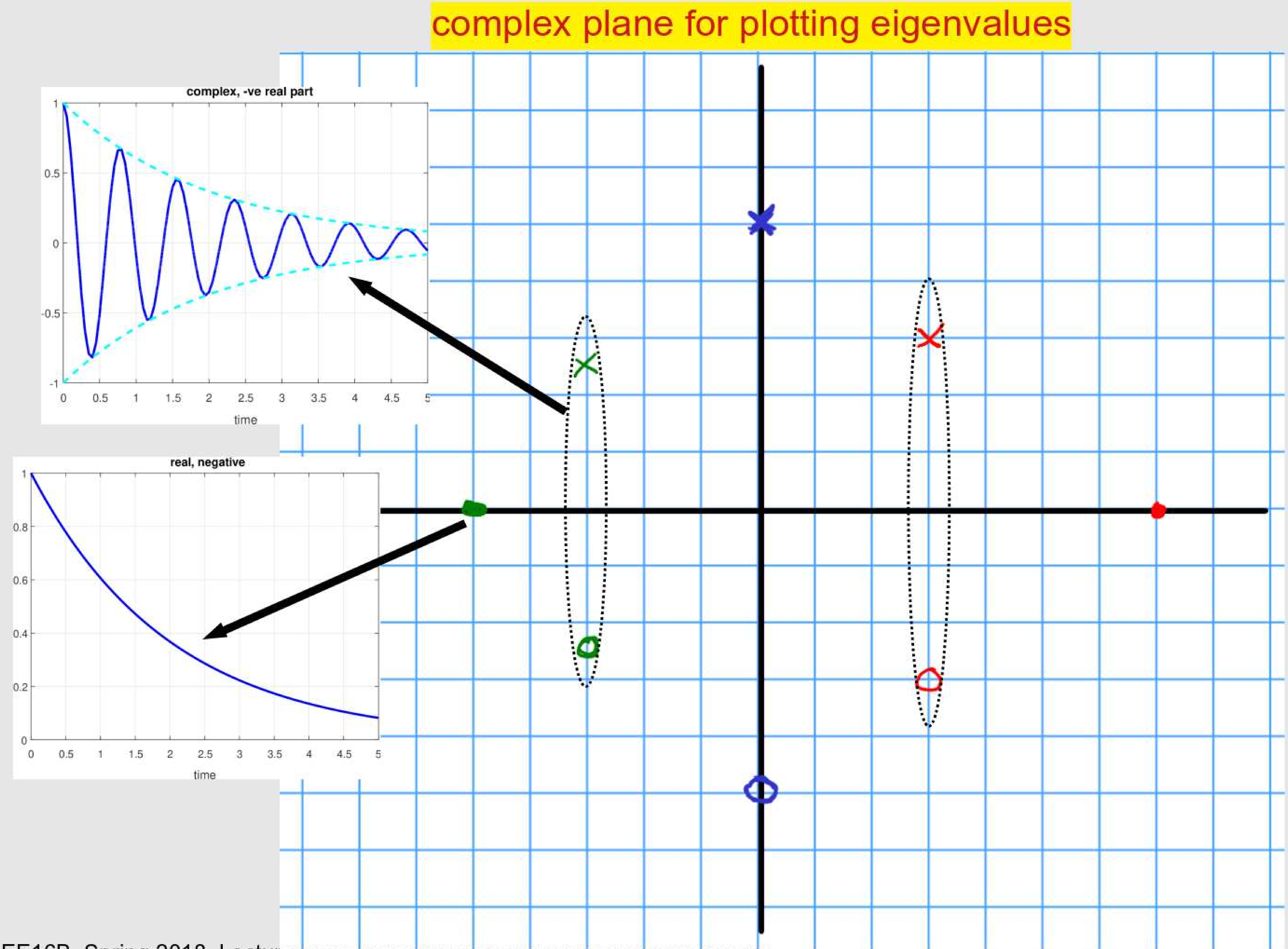


Eigenvalues and Responses (continuous)

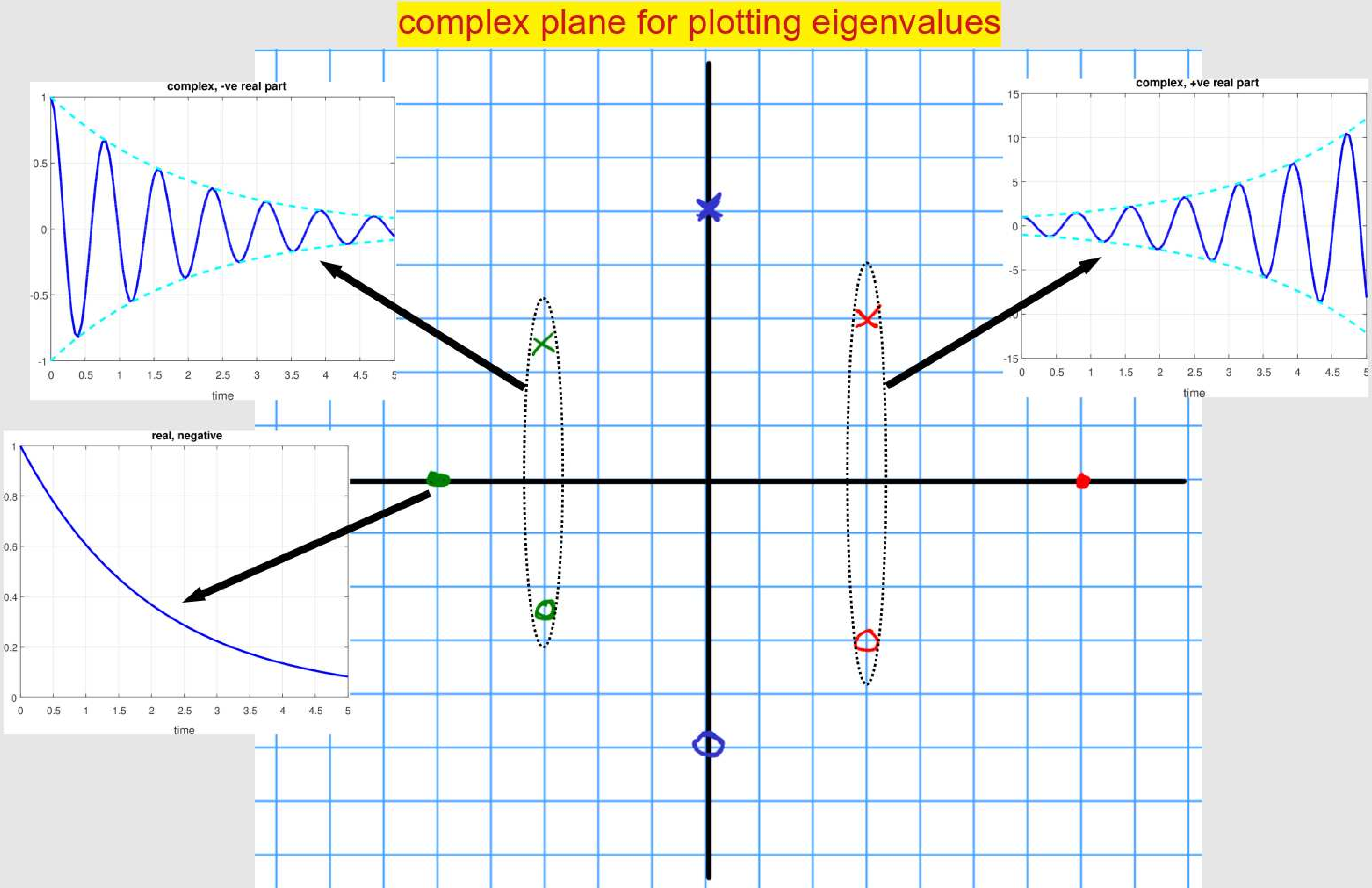
complex plane for plotting eigenvalues



Eigenvalues and Responses (continuous)

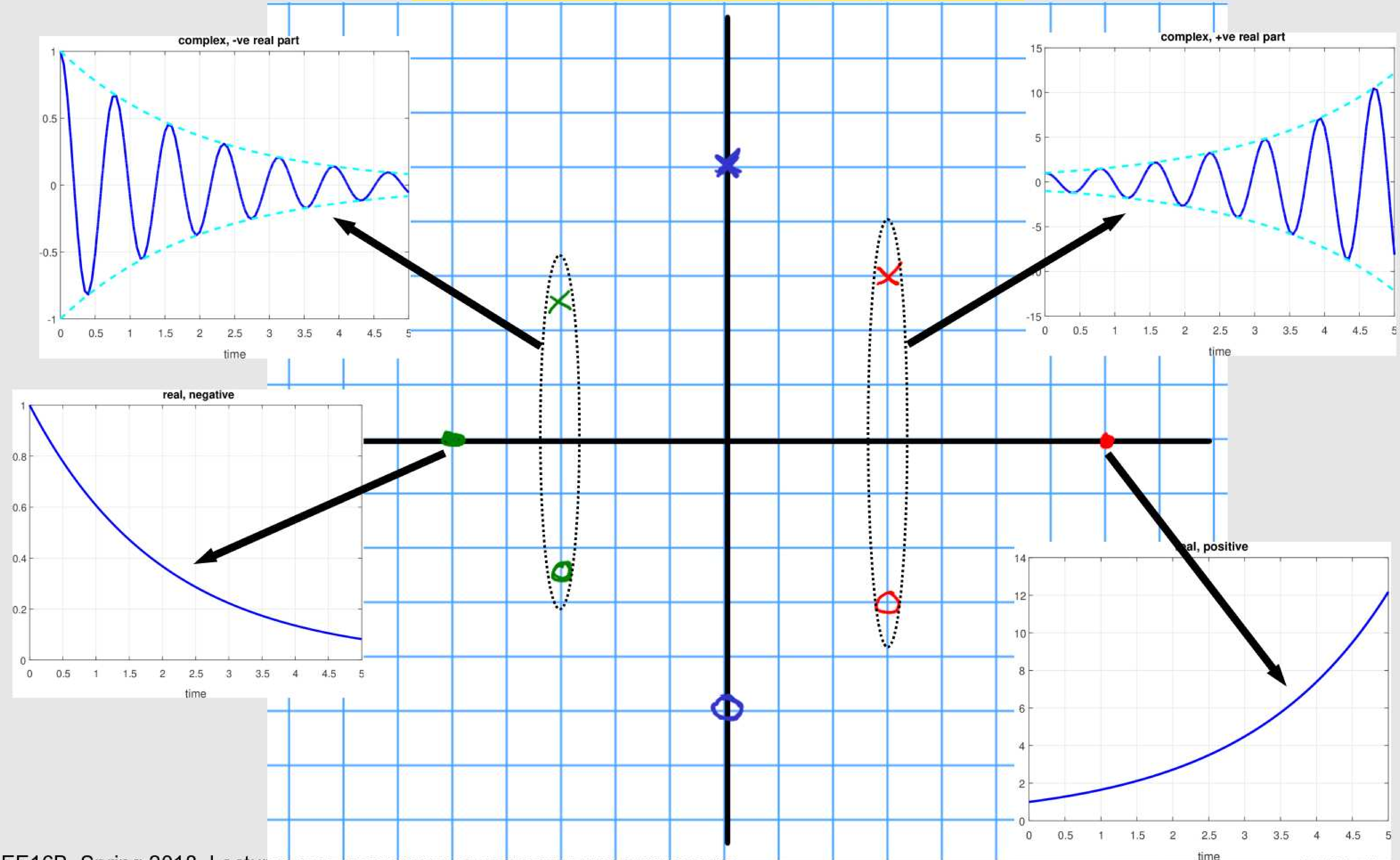


Eigenvalues and Responses (continuous)

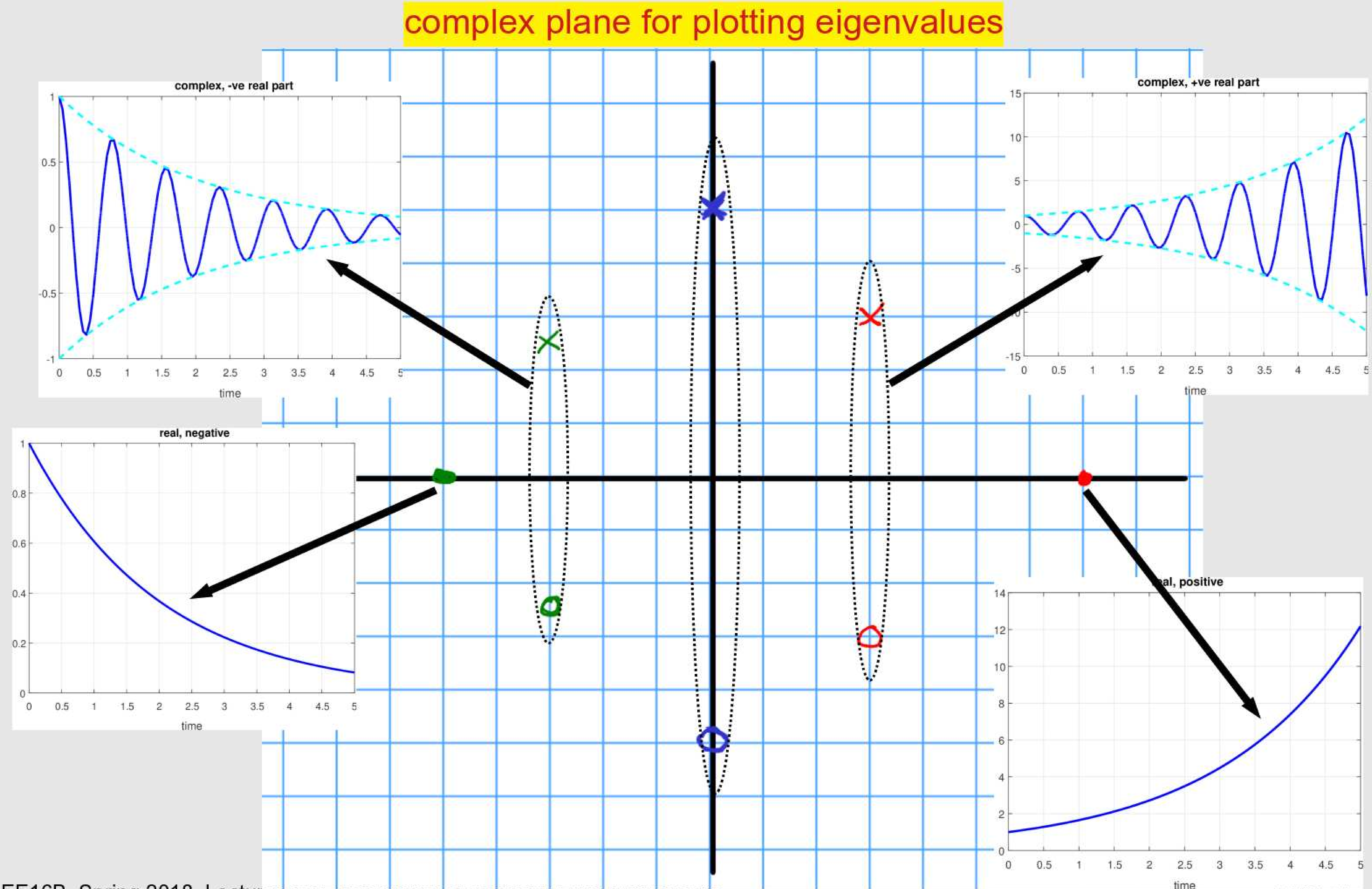


Eigenvalues and Responses (continuous)

complex plane for plotting eigenvalues

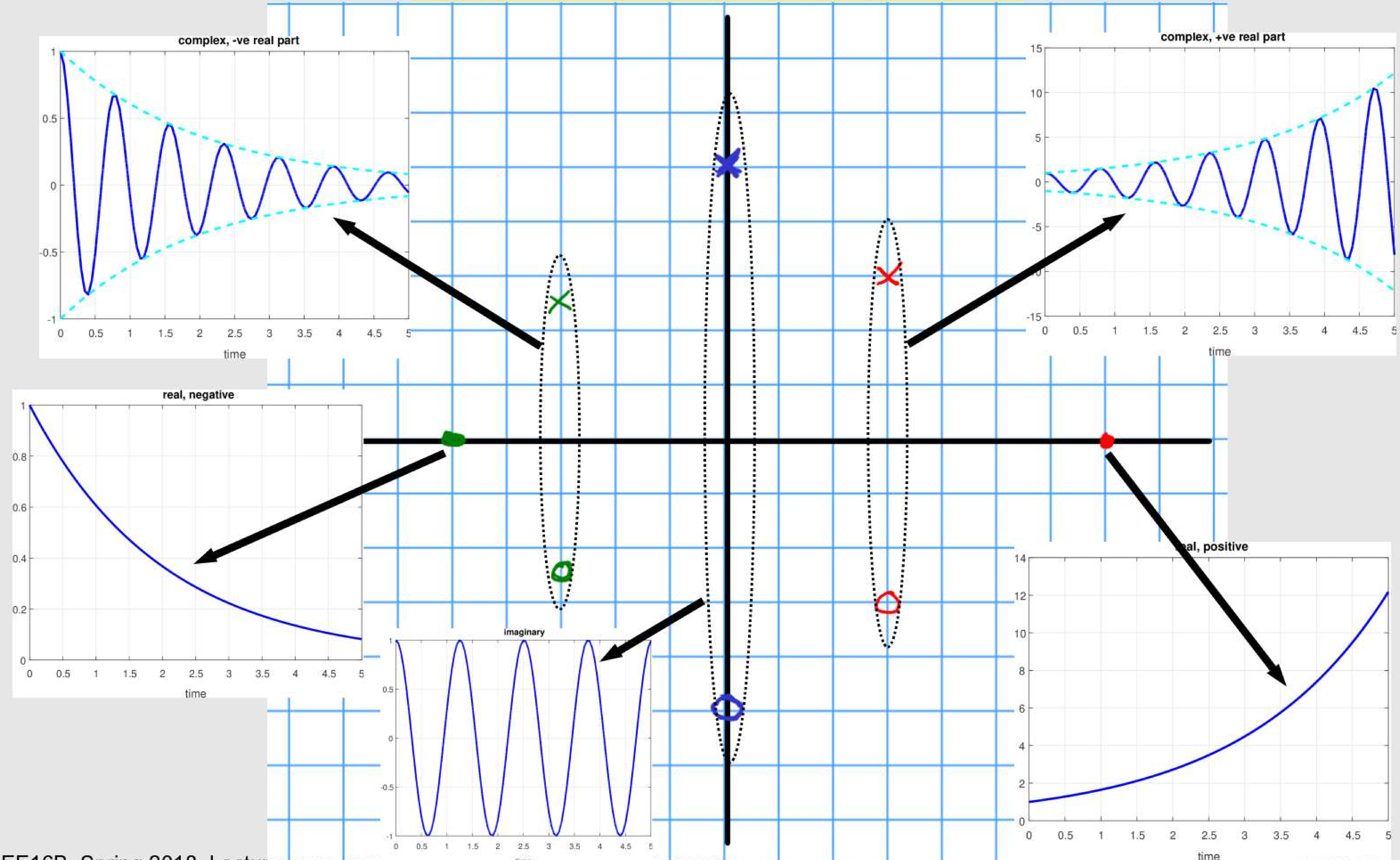


Eigenvalues and Responses (continuous)



Eigenvalues and Responses (continuous)

complex plane for plotting eigenvalues



Eigenvalues of Linearized Pendulum

- (move to xournal)

$$\boxed{\frac{d\vec{x}}{dt} = \begin{bmatrix} 0 & 1 \\ -g/l & -k/m \end{bmatrix} \vec{x} + \begin{bmatrix} 0 \\ +\frac{u(t)}{ml} \end{bmatrix}}$$

want non-zero solution

$$A\vec{p} = \lambda\vec{p} \implies (A - \lambda I)\vec{p} = 0 \implies \underbrace{\begin{bmatrix} -\lambda & 1 \\ -g/l & -\frac{k}{m} - \lambda \end{bmatrix}}_{\text{det should } = 0} \vec{p} = 0$$

$$\lambda(\lambda + k/m) + \frac{g}{l} = 0 \implies \lambda^2 + \frac{k}{m}\lambda + \frac{g}{l} = 0 \implies \boxed{\lambda_{1,2} = \frac{-k}{2m} \pm \frac{1}{2}\sqrt{\frac{k^2}{m^2} - \frac{4g}{l}}}$$

Eigenvalues of Linearized Pendulum

$$\frac{d\vec{x}}{dt} = \begin{bmatrix} 0 & 1 \\ -g/l & -k/m \end{bmatrix} \vec{x} + \begin{bmatrix} 0 \\ +\frac{u(t)}{ml} \end{bmatrix}$$

$$\boxed{\lambda_{1,2} = \frac{-k}{2m} \pm \frac{1}{2}\sqrt{\frac{k^2}{m^2} - \frac{4g}{l}}}$$

want non-zero solution

$$A\vec{p} = \lambda\vec{p} \implies (A - \lambda I)\vec{p} = 0 \implies \underbrace{\begin{bmatrix} -\lambda & 1 \\ -g/l & -\frac{k}{m} - \lambda \end{bmatrix}}_{\det \text{ should } = 0} \vec{p} = 0$$

$$\lambda(\lambda + k/m) + \frac{g}{l} = 0 \implies \lambda^2 + \frac{k}{m}\lambda + \frac{g}{l} = 0 \implies \boxed{\lambda_{1,2} = \frac{-k}{2m} \pm \frac{1}{2}\sqrt{\frac{k^2}{m^2} - \frac{4g}{l}}}$$

Eigenvalues of Linearized Pendulum

$$\frac{d\vec{x}}{dt} = \begin{bmatrix} 0 & 1 \\ -g/l & -k/m \end{bmatrix} \vec{x} + \begin{bmatrix} 0 \\ +\frac{u(t)}{ml} \end{bmatrix}$$

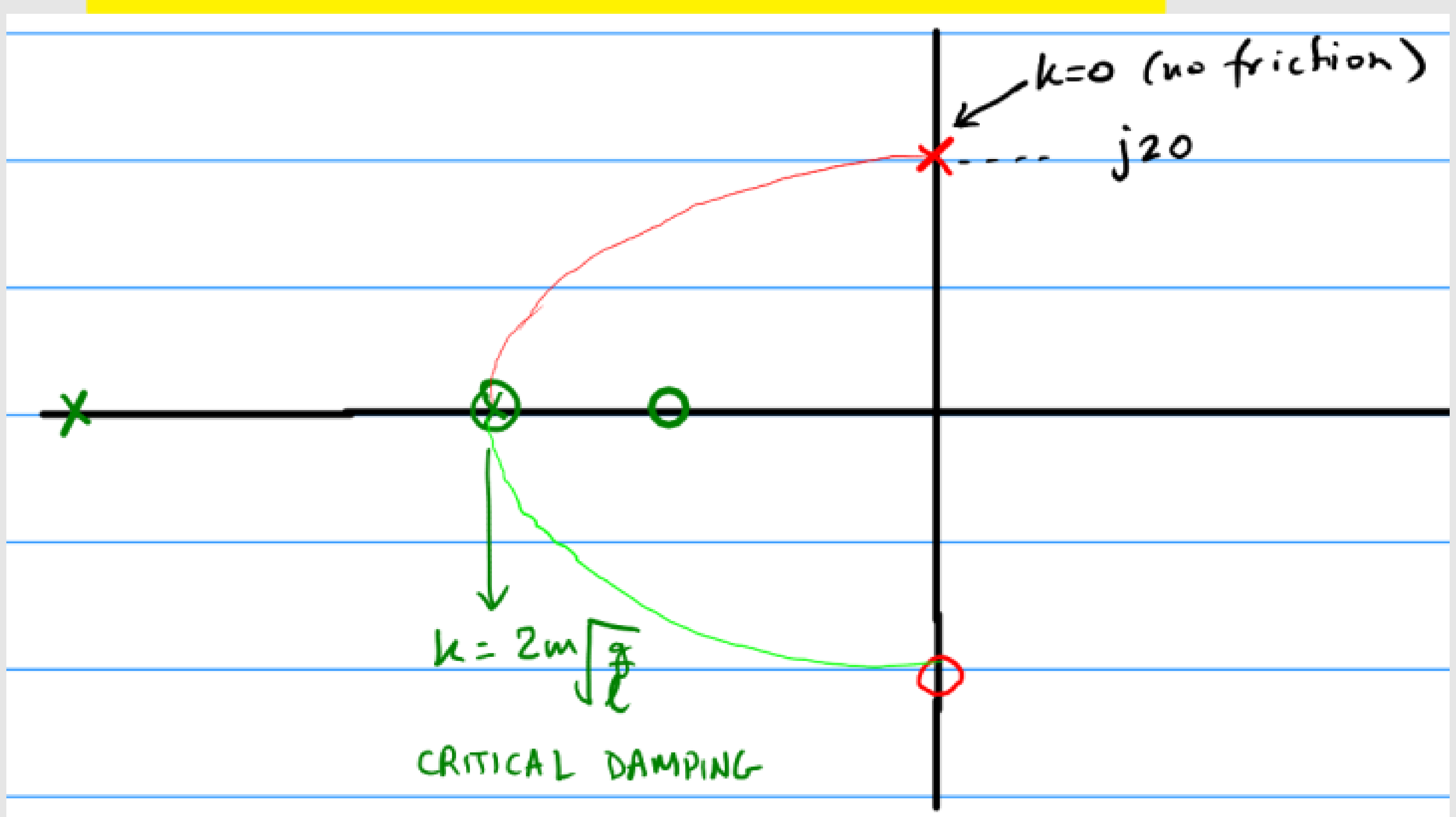
$$\lambda_{1,2} = \frac{-k}{2m} \pm \frac{1}{2}\sqrt{\frac{k^2}{m^2} - \frac{4g}{l}}$$

want non-zero solution

$$A\vec{p} = \lambda\vec{p} \implies (A - \lambda I)\vec{p} = 0 \implies \underbrace{\begin{bmatrix} -\lambda & 1 \\ -g/l & -\frac{k}{m} - \lambda \end{bmatrix}}_{\det \text{ should } = 0} \vec{p} = 0$$

$$\lambda(\lambda + k/m) + \frac{g}{l} = 0 \implies \lambda^2 + \frac{k}{m}\lambda + \frac{g}{l} = 0 \implies \boxed{\lambda_{1,2} = \frac{-k}{2m} \pm \frac{1}{2}\sqrt{\frac{k^2}{m^2} - \frac{4g}{l}}}$$

plot eigenvalues as k changes



Eigenvalues of Linearized Pendulum

$$\frac{d\vec{x}}{dt} = \begin{bmatrix} 0 & 1 \\ -g/\ell & -k/m \end{bmatrix} \vec{x} + \begin{bmatrix} 0 \\ +\frac{u(t)}{m\ell} \end{bmatrix}$$

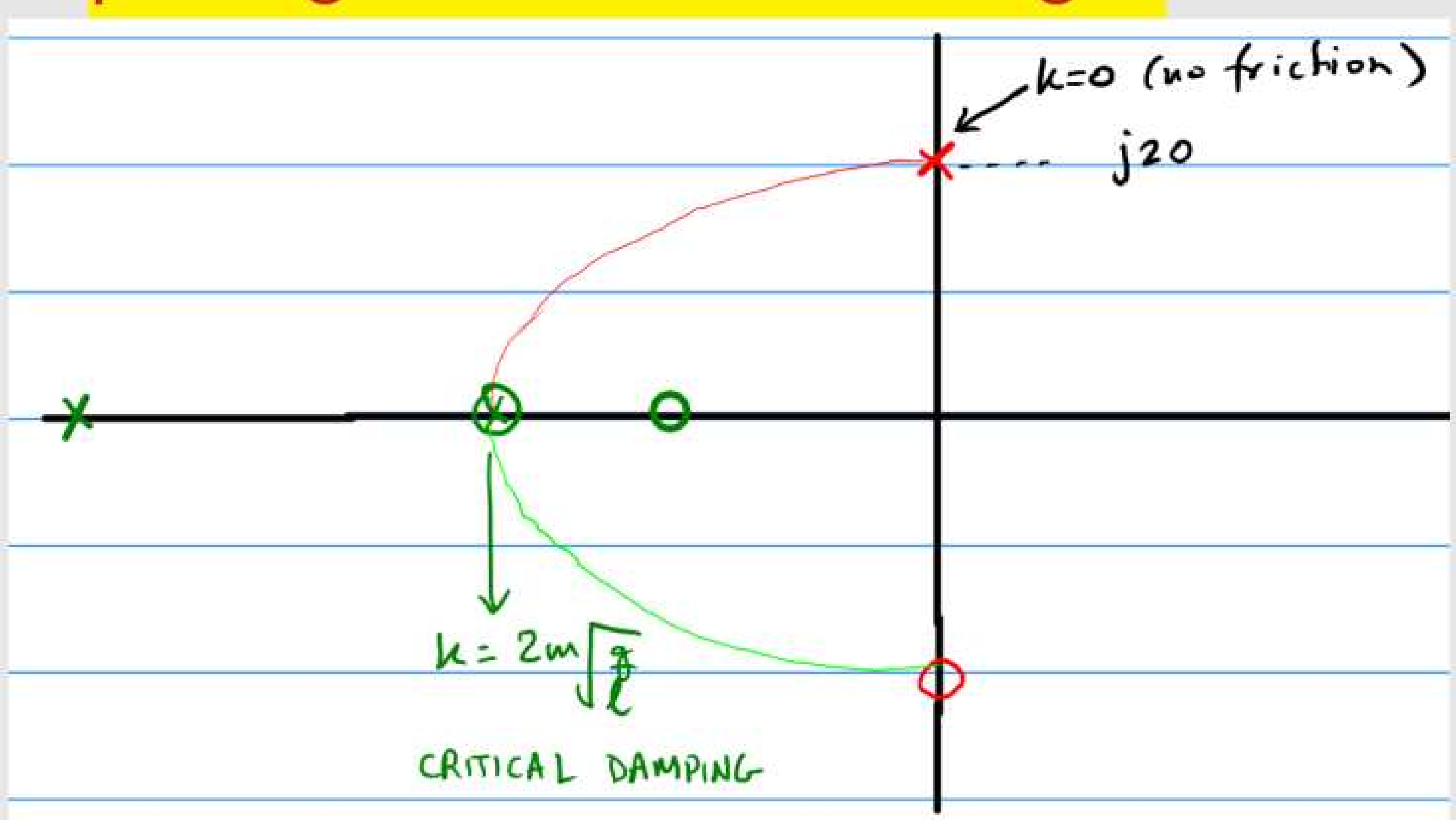
$$\boxed{\lambda_{1,2} = \frac{-k}{2m} \pm \frac{1}{2}\sqrt{\frac{k^2}{m^2} - \frac{4g}{\ell}}}$$

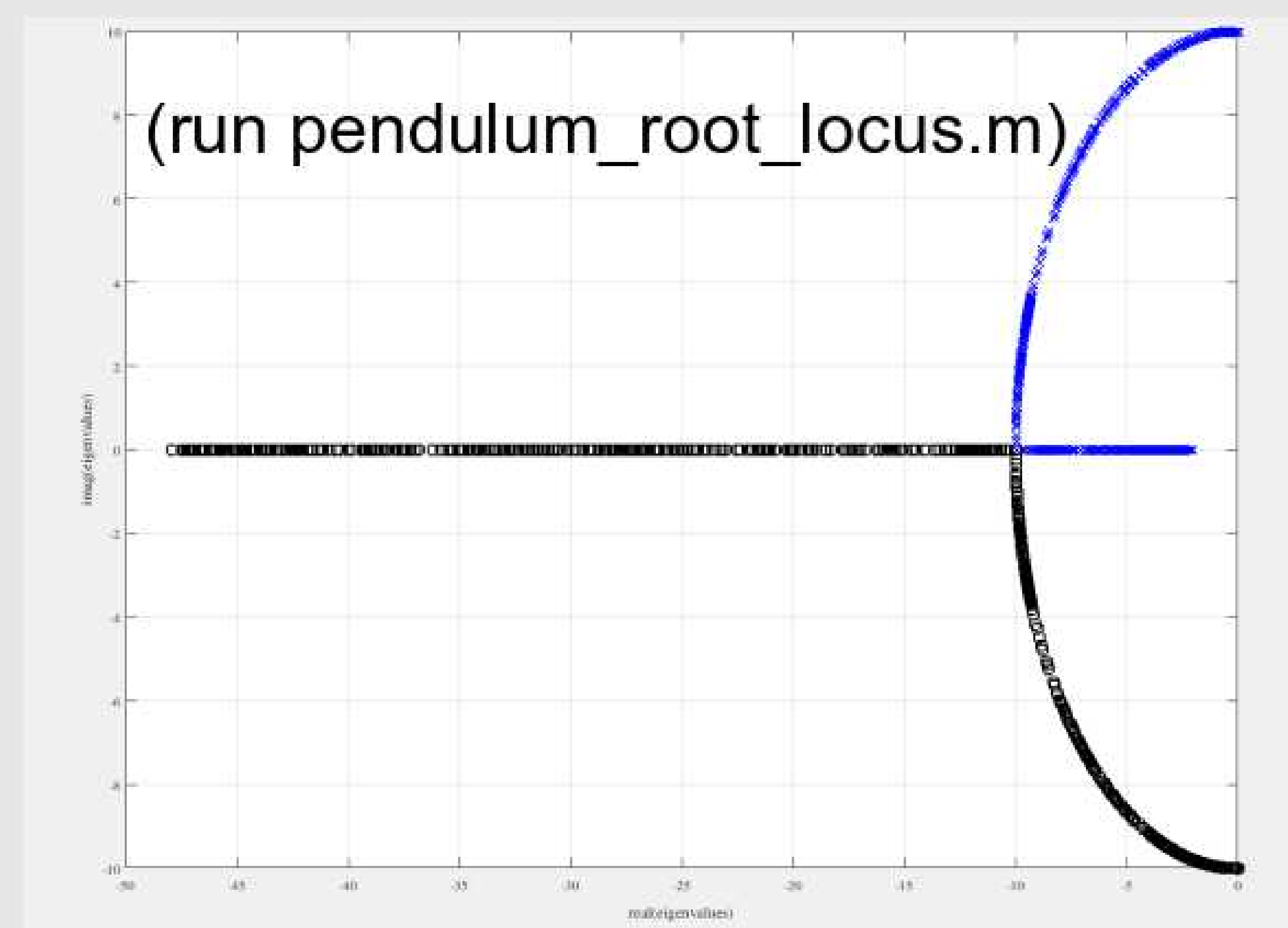
want non-zero solution

$$A\vec{p} = \lambda\vec{p} \implies (A - \lambda I)\vec{p} = 0 \implies \underbrace{\begin{bmatrix} -\lambda & 1 \\ -g/\ell & -\frac{k}{m} - \lambda \end{bmatrix}}_{\text{det should} = 0} \vec{p} = 0$$

$$\lambda(\lambda + k/m) + \frac{g}{\ell} = 0 \implies \lambda^2 + \frac{k}{m}\lambda + \frac{g}{\ell} = 0 \implies \boxed{\lambda_{1,2} = \frac{-k}{2m} \pm \frac{1}{2}\sqrt{\frac{k^2}{m^2} - \frac{4g}{\ell}}}$$

plot eigenvalues as k changes





Eigenvalues of Linearized Pendulum

$$\boxed{\frac{d\vec{x}}{dt} = \begin{bmatrix} 0 & 1 \\ -g/\ell & -k/m \end{bmatrix} \vec{x} + \begin{bmatrix} 0 \\ +\frac{u(t)}{m\ell} \end{bmatrix}}$$

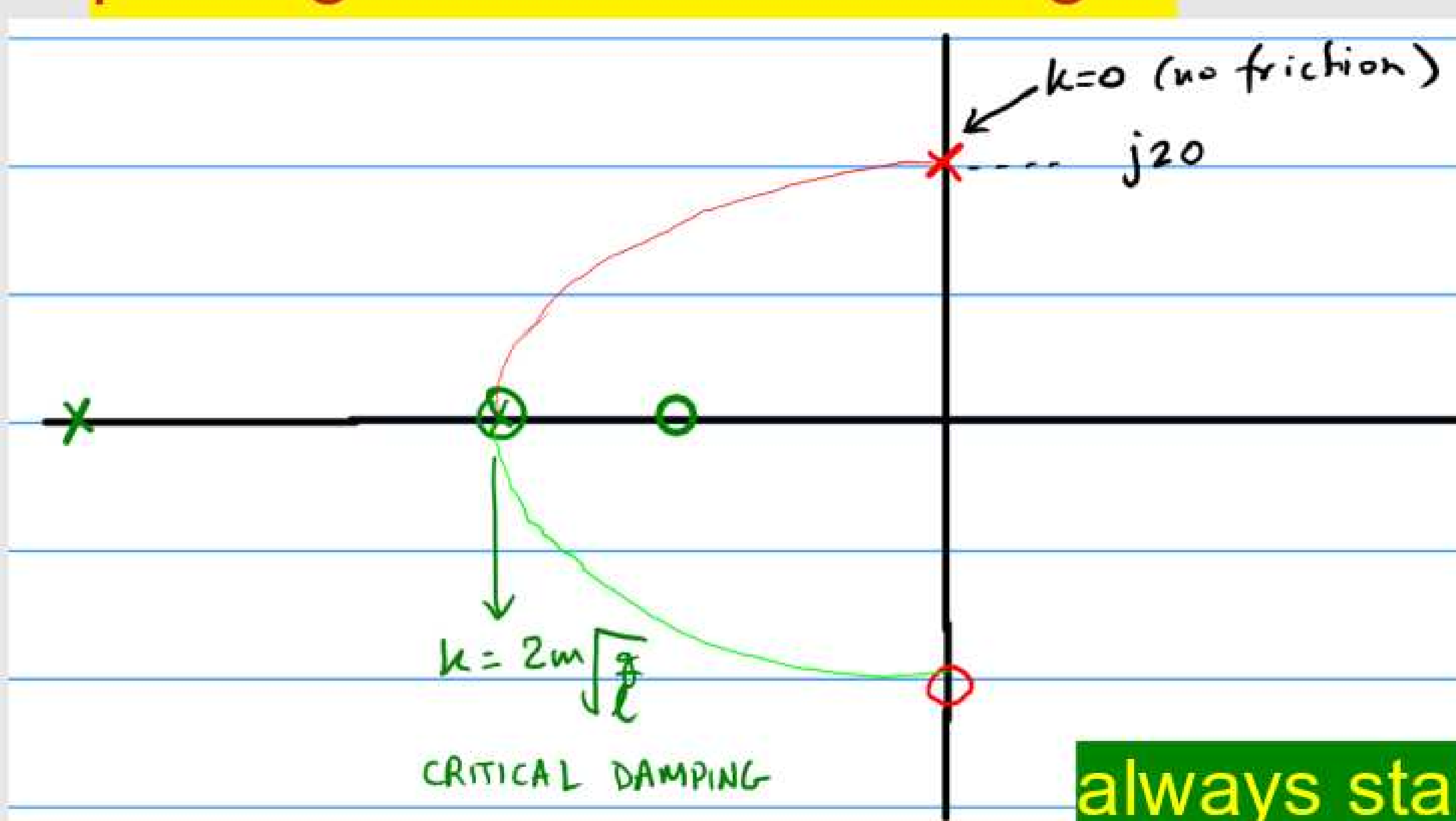
$$\boxed{\lambda_{1,2} = \frac{-k}{2m} \pm \frac{1}{2}\sqrt{\frac{k^2}{m^2} - \frac{4g}{\ell}}}$$

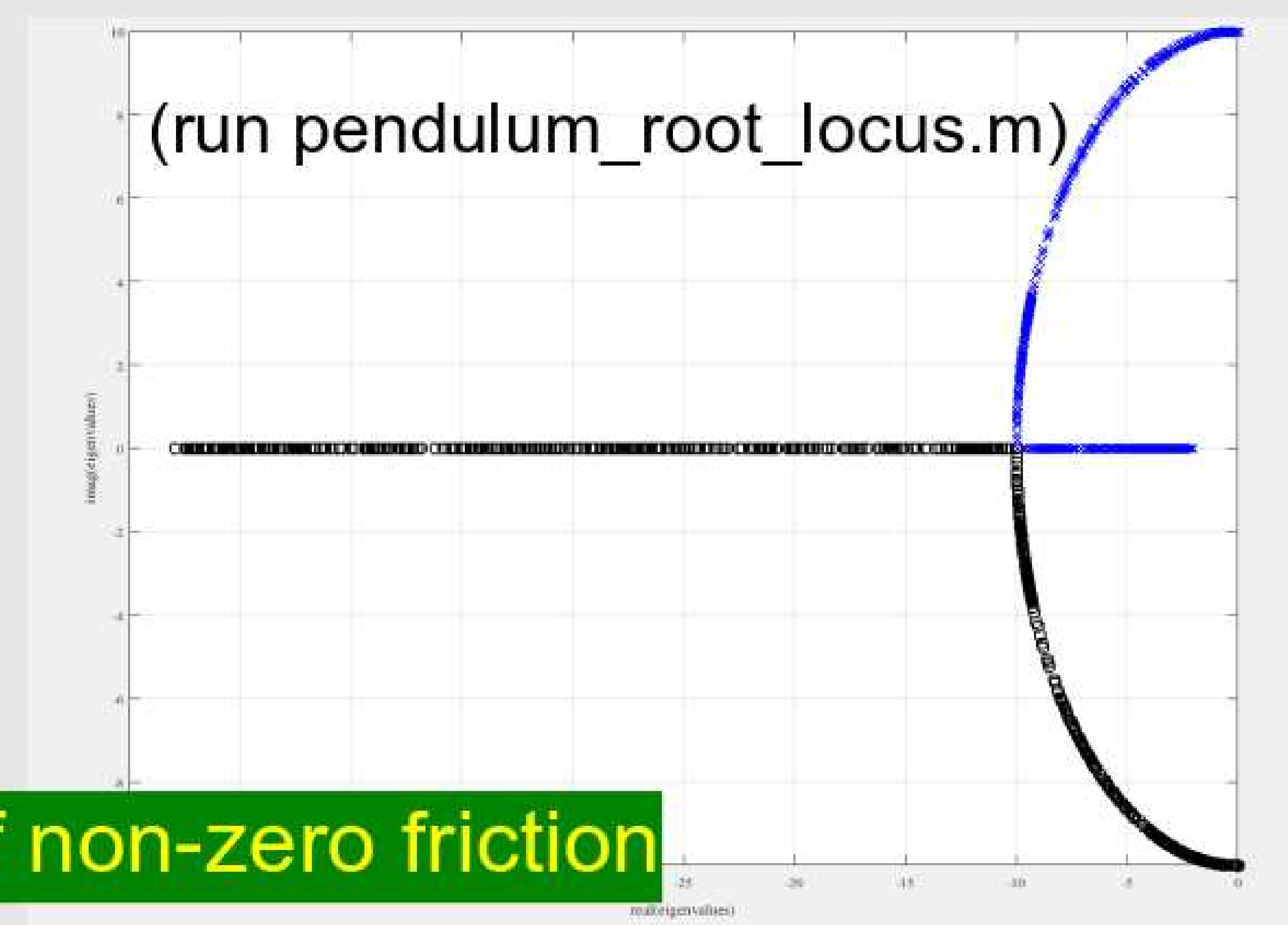
want non-zero solution

$$A\vec{p} = \lambda\vec{p} \implies (A - \lambda I)\vec{p} = 0 \implies \underbrace{\begin{bmatrix} -\lambda & 1 \\ -g/\ell & -\frac{k}{m} - \lambda \end{bmatrix}}_{\det \text{ should } = 0} \vec{p} = 0$$

$$\lambda(\lambda + k/m) + \frac{g}{\ell} = 0 \implies \lambda^2 + \frac{k}{m}\lambda + \frac{g}{\ell} = 0 \implies \boxed{\lambda_{1,2} = \frac{-k}{2m} \pm \frac{1}{2}\sqrt{\frac{k^2}{m^2} - \frac{4g}{\ell}}}$$

plot eigenvalues as k changes





always stable if non-zero friction

Eigenvalues of Inverted Pendulum

- $A = J_x = \begin{bmatrix} 0 & 1 \\ +\frac{g}{l} & -\frac{k}{m} \end{bmatrix}$

Eigenvalues of Inverted Pendulum

- $A = J_x = \begin{bmatrix} 0 & 1 \\ +\frac{g}{l} & -\frac{k}{m} \end{bmatrix}$ $\qquad$ $\lambda_{1,2} = -\frac{k}{2m} \pm \frac{1}{2}\sqrt{\frac{k^2}{m^2} + 4\frac{g}{l}}$

Eigenvalues of Inverted Pendulum

- $A = J_x = \begin{bmatrix} 0 & 1 \\ +\frac{g}{l} & -\frac{k}{m} \end{bmatrix}$ $\quad\quad$ $\lambda_{1,2} = -\frac{k}{2m} \pm \boxed{\frac{1}{2}\sqrt{\frac{k^2}{m^2} + 4\frac{g}{l}}}$

always real, always greater than $\frac{k}{2m}$

Eigenvalues of Inverted Pendulum

- $A = J_x = \begin{bmatrix} 0 & 1 \\ +\frac{g}{l} & -\frac{k}{m} \end{bmatrix}$

$$\lambda_{1,2} = -\frac{k}{2m} \pm \boxed{\frac{1}{2}\sqrt{\frac{k^2}{m^2} + 4\frac{g}{l}}}$$

always real, always greater than $\frac{k}{2m}$

one eigenvalue always positive!

Eigenvalues of Inverted Pendulum

- $A = J_x = \begin{bmatrix} 0 & 1 \\ +\frac{g}{l} & -\frac{k}{m} \end{bmatrix}$ $\qquad \lambda_{1,2} = -\frac{k}{2m} \pm \boxed{\frac{1}{2}\sqrt{\frac{k^2}{m^2} + 4\frac{g}{l}}}$

always real, always greater than $\frac{k}{2m}$

one eigenvalue always positive!

always unstable!

Eigenvalues of Inverted Pendulum

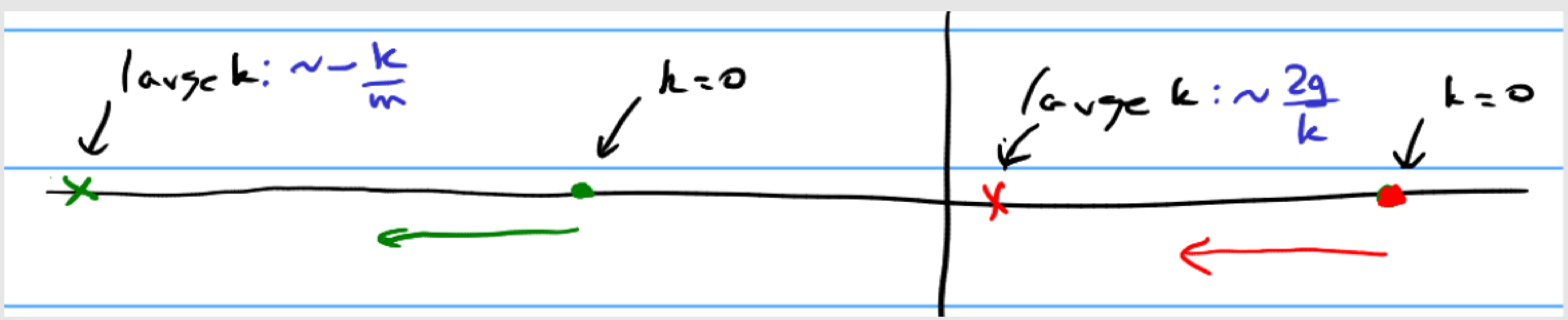
- $A = J_x = \begin{bmatrix} 0 & 1 \\ +\frac{g}{l} & -\frac{k}{m} \end{bmatrix}$

$$\lambda_{1,2} = -\frac{k}{2m} \pm \boxed{\frac{1}{2}\sqrt{\frac{k^2}{m^2} + 4\frac{g}{l}}}$$

always real, always greater than $\frac{k}{2m}$

one eigenvalue always positive!

always unstable!



Stability for Discrete Time Systems

- The scalar case: $\Delta x[t+1] = a\Delta x[t] + b\Delta u[t]$, IC Δx[0]

Stability for Discrete Time Systems

real

- The scalar case: $\Delta x[t+1] = a\Delta x[t] + b\Delta u[t]$, IC $\Delta x[0]$

Stability for Discrete Time Systems

real

- The scalar case: $\Delta x[t+1] = a\Delta x[t] + b\Delta u[t]$, IC Δx[0]
  - [already linear(ized); everything is real]

Stability for Discrete Time Systems

- The scalar case: $\Delta x[t+1] = \overset{\text{real}}{a}\Delta x[t] + \overset{\text{real}}{b}\Delta u[t]$, IC $\Delta x[0]$
  - [already linear(ized); everything is real]
  - (move to xournal?)

$t=0$: $\Delta x[1] = a\Delta x[0] + b\Delta u[0]$

$t=1$: $\Delta x[2] = a\Delta x[1] + b\Delta u[1] = a^2\Delta x[0] + ab\Delta u[0] + b\Delta u[1]$

$t=2$: $\Delta x[3] = a\Delta x[2] + b\Delta u[2] = a^3\Delta x[0] + a^2 b\Delta u[0] + ab\Delta u[1] + b\Delta u[2]$

$\vdots$

$\rightarrow \Delta x[t] = a^t\Delta x[0] + a^{t-1}b\Delta u[0] + a^{t-2}b\Delta u[1] + \cdots + ab\Delta u[t-2] + b\Delta u[t-1]$

$$= \underbrace{a^t\Delta x[0]}_{\text{I.C. term}} + \underbrace{\sum_{i=1}^{t} a^{t-i}b\Delta u[i-1]}_{\text{discrete-time convolution} = \text{input term}}$$

- $$\Delta x[t] = a^t\Delta x[0] + \sum_{i=1}^{t} a^{t-i}b\Delta u[i-1]$$

Stability for Discrete Time Systems

real

- The scalar case: $\Delta x[t+1] = a\Delta x[t] + b\Delta u[t]$, IC $\Delta x[0]$
  - [already linear(ized); everything is real]

$t=0$: $\Delta x[1] = a\Delta x[0] + b\Delta u[0]$

$t=1$: $\Delta x[2] = a\Delta x[1] + b\Delta u[0] = a^2\Delta x[0] + ab\Delta u[0] + b\Delta u[1]$

$t=2$: $\Delta x[3] = a\Delta x[2] + b\Delta u[2] = a^3\Delta x[0] + a^2 b\Delta u[0] + ab\Delta u[1] + b\Delta u[2]$

$\vdots$

$$\Delta x[t] = a^t\Delta x[0] + a^{t-1}b\Delta u[0] + a^{t-2}b\Delta u[1] + \cdots + ab\Delta u[t-2] + b\Delta u[t-1]$$

$$= \underbrace{a^t\Delta x[0]}_{\text{I.C. term}} + \underbrace{\sum_{i=1}^{t} a^{t-i} b\Delta u[i-1]}_{\text{discrete-time convolution} \,=\, \text{input term}}$$

- $$\Delta x[t] = a^t\Delta x[0] + \sum_{i=1}^{t} a^{t-i} b\Delta u[i-1]$$

Stability for Discrete Time Systems

- The scalar case: $\Delta x[t+1] = a\Delta x[t] + b\Delta u[t]$, IC $\Delta x[0]$

  real

  - [already linear(ized); everything is real]

$t=0$: $\Delta x[1] = a\Delta x[0] + b\Delta u[0]$

$t=1$: $\Delta x[2] = a\Delta x[1] + b\Delta u[1] = a^2\Delta x[0] + ab\Delta u[0] + b\Delta u[1]$

$t=2$: $\Delta x[3] = a\Delta x[2] + b\Delta u[2] = a^3\Delta x[0] + a^2 b\Delta u[0] + ab\Delta u[1] + b\Delta u[2]$

$\vdots$

$\longrightarrow$ $\Delta x[t] = a^t\Delta x[0] + a^{t-1}b\Delta u[0] + a^{t-2}b\Delta u[1] + \cdots + ab\Delta u[t-2] + b\Delta u[t-1]$

$$= \underbrace{a^t\Delta x[0]}_{\text{I.C. term}} + \underbrace{\sum_{i=1}^{t} a^{t-i} b\Delta u[i-1]}_{\text{discrete-time convolution} \,=\, \text{input term}}$$

- $\Delta x[t] = a^t\Delta x[0] + \sum_{i=1}^{t} a^{t-i} b\Delta u[i-1]$

  IC term

Stability for Discrete Time Systems

- The scalar case: $\Delta x[t+1] = \underbrace{a}_{\text{real}}\Delta x[t] + \underbrace{b}_{\text{real}}\Delta u[t]$, IC $\Delta x[0]$
  - [already linear(ized); everything is real]

- $\Delta x[t] = \underbrace{a^t \Delta x[0]}_{\text{IC term}} + \underbrace{\sum_{i=1}^{t} a^{t-i} b \Delta u[i-1]}_{\text{input term (discrete convolution)}}$

IC term

input term **(discrete convolution)**

$t=0$: $\Delta x[1] = a\Delta x[0] + b\Delta u[0]$

$t=1$: $\Delta x[2] = a\Delta x[1] + b\Delta u[1] = a^2\Delta x[0] + ab\Delta u[0] + b\Delta u[1]$

$t=2$: $\Delta x[3] = a\Delta x[2] + b\Delta u[2] = a^3\Delta x[0] + a^2 b\Delta u[0] + ab\Delta u[1] + b\Delta u[2]$

$\vdots$

$$\Delta x[t] = a^t\Delta x[0] + a^{t-1}b\Delta u[0] + a^{t-2}b\Delta u[1] + \cdots + ab\Delta u[t-2] + b\Delta u[t-1]$$

$$= \underbrace{a^t\Delta x[0]}_{\substack{\text{I.C.}\\ \text{term}}} + \underbrace{\sum_{i=1}^{t} a^{t-i} b\Delta u[i-1]}_{\substack{\text{discrete-time}\\ \text{convolution}}} = \text{input term}$$

Stability for Discrete Time Systems

- The scalar case: $\Delta x[t+1] = a\Delta x[t] + b\Delta u[t]$, IC $\Delta x[0]$
  - real
  - [already linear(ized); everything is real]

$t=0$: $\Delta x[1] = a\Delta x[0] + b\Delta u[0]$

$t=1$: $\Delta x[2] = a\Delta x[1] + b\Delta u[0] = a^2\Delta x[0] + ab\Delta u[0] + b\Delta u[1]$

$t=2$: $\Delta x[3] = a\Delta x[2] + b\Delta u[2] = a^3\Delta x[0] + a^2 b\Delta u[0] + ab\Delta u[1] + b\Delta u[2]$

$\vdots$

$$\Delta x[t] = a^t\Delta x[0] + a^{t-1}b\Delta u[0] + a^{t-2}b\Delta u[1] + \cdots + ab\Delta u[t-2] + b\Delta u[t-1]$$

$$= \underbrace{a^t\Delta x[0]}_{\text{I.C. term}} + \underbrace{\sum_{i=1}^{t} a^{t-i} b\Delta u[i-1]}_{\text{discrete-time convolution} \,=\, \text{input term}}$$

- $\Delta x[t] = a^t\Delta x[0] + \sum_{i=1}^{t} a^{t-i} b\Delta u[i-1]$

IC term

input term **(discrete convolution)**

- Initial Condition term: $a^t\Delta x[0]$

Stability for Discrete Time Systems

- The scalar case: $\Delta x[t+1] = a\Delta x[t] + b\Delta u[t]$, IC $\Delta x[0]$
  - real
  - ➔ [already linear(ized); everything is real]

$t=0$: $\Delta x[1] = a\Delta x[0] + b\Delta u[0]$

$t=1$: $\Delta x[2] = a\Delta x[1] + b\Delta u[0] = a^2\Delta x[0] + ab\Delta u[0] + b\Delta u[1]$

$t=2$: $\Delta x[3] = a\Delta x[2] + b\Delta u[2] = a^3\Delta x[0] + a^2 b\Delta u[0] + ab\Delta u[1] + b\Delta u[2]$

$\vdots$

$$\Delta x[t] = a^t\Delta x[0] + a^{t-1}b\Delta u[0] + a^{t-2}b\Delta u[1] + \cdots + ab\Delta u[t-2] + b\Delta u[t-1]$$

$$= \underbrace{a^t\Delta x[0]}_{\text{I.C. term}} + \underbrace{\sum_{i=1}^{t} a^{t-i} b\Delta u[i-1]}_{\text{discrete-time convolution}} = \text{input term}$$

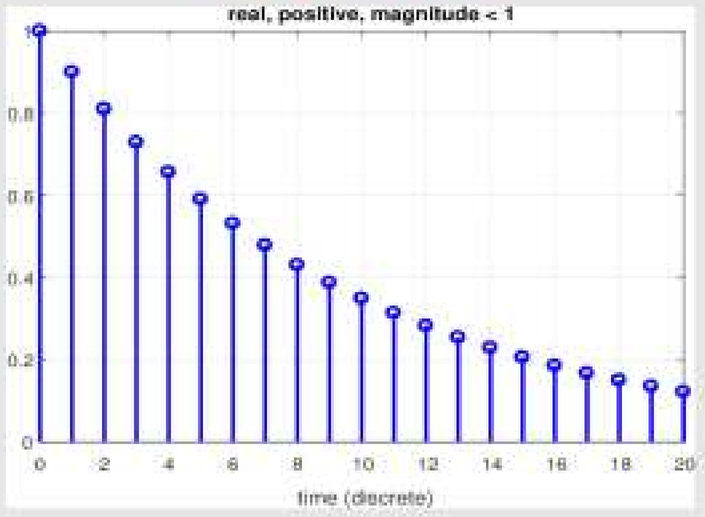
- $\Delta x[t] = \boxed{a^t\Delta x[0]} + \boxed{\sum_{i=1}^{t} a^{t-i}b\Delta u[i-1]}$

IC term

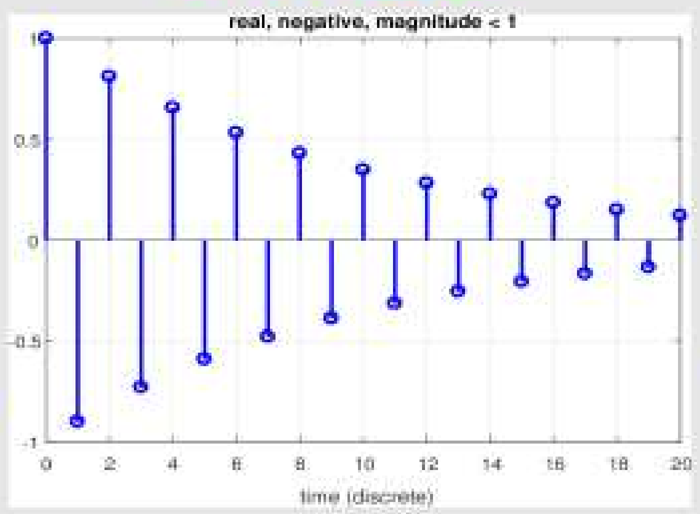
input term **(discrete convolution)**

- Initial Condition term: $a^t\Delta x[0]$

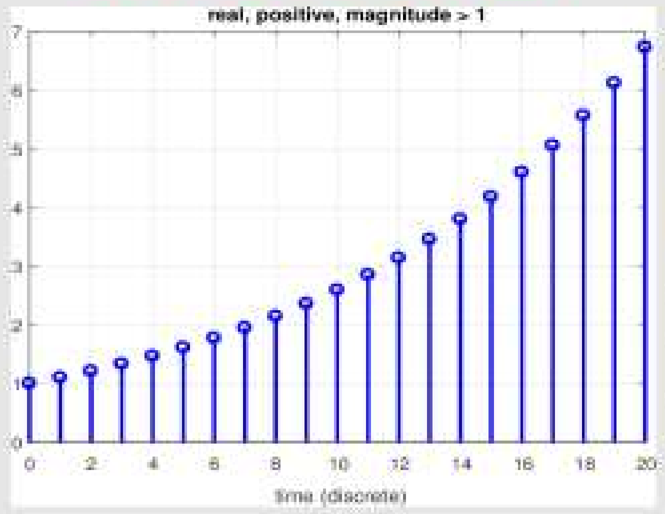
0<a<1: dies down
STABLE



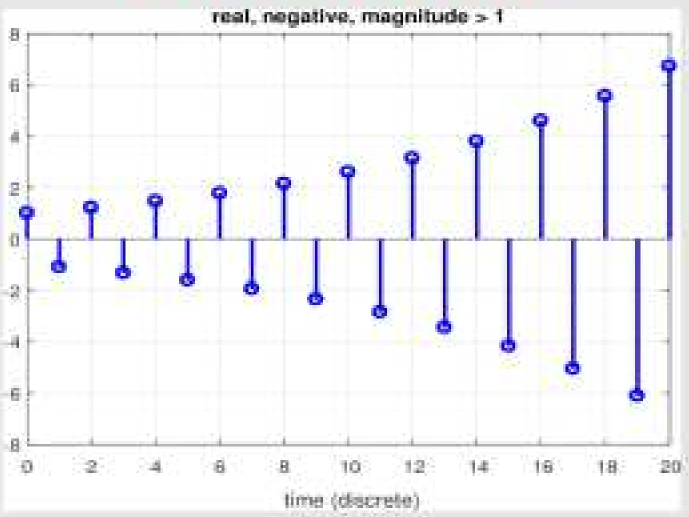
-1<a<0: dies down
STABLE



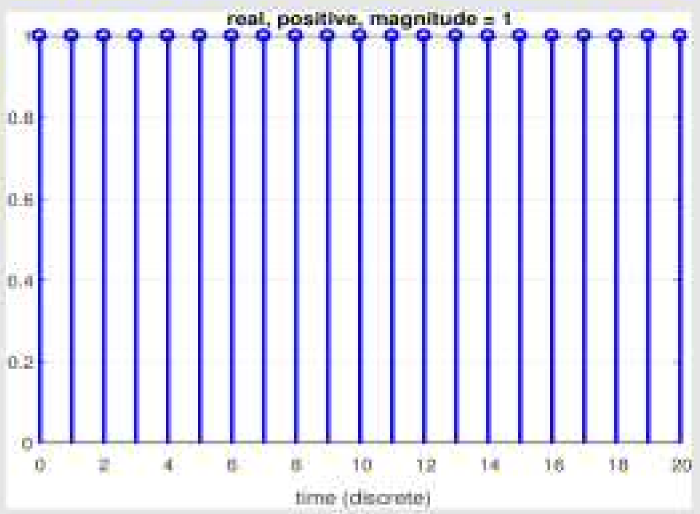
a>1: blows up
UNSTABLE



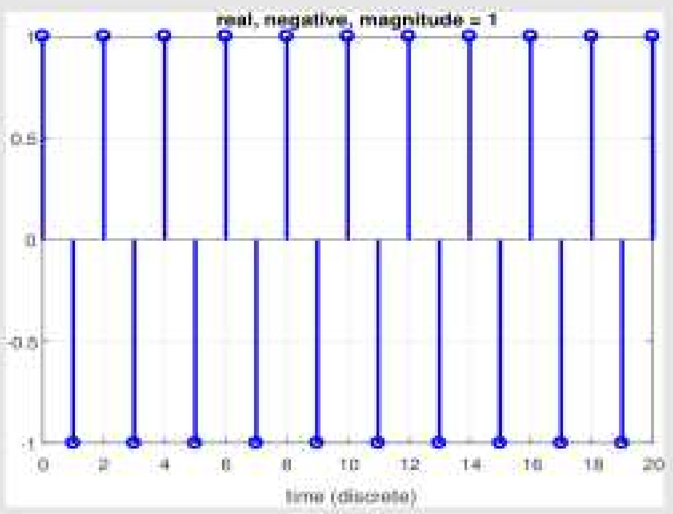
a<-1: blows up
UNSTABLE



a=1: constant
MARGINALLY STABLE



a=-1: constant
MARGINALLY STABLE



Stability for Discrete Time Systems

- The scalar case: $\Delta x[t+1] = a\Delta x[t] + b\Delta u[t]$, IC Δx[0]

  real

  - [already linear(ized); everything is real]

$t=0$: $\Delta x[1] = a\Delta x[0] + b\Delta u[0]$

$t=1$: $\Delta x[2] = a\Delta x[1] + b\Delta u[0] = a^2\Delta x[0] + ab\Delta u[0] + b\Delta u[1]$

$t=2$: $\Delta x[3] = a\Delta x[2] + b\Delta u[2] = a^3\Delta x[0] + a^2 b\Delta u[0] + ab\Delta u[1] + b\Delta u[2]$

$\vdots$

$\Delta x[t] = a^t\Delta x[0] + a^{t-1}b\Delta u[0] + a^{t-2}b\Delta u[1] + \cdots + ab\Delta u[t-2] + b\Delta u[t-1]$

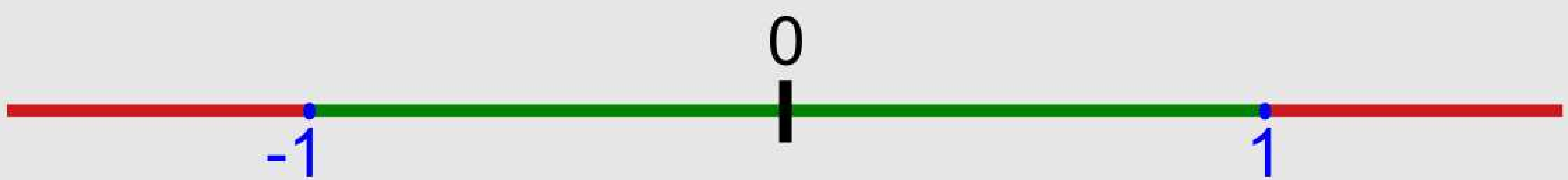
$= \underbrace{a^t\Delta x[0]}_{\text{I.C. term}} + \underbrace{\sum_{i=1}^{t} a^{t-i} b\Delta u[i-1]}_{\text{discrete-time convolution}} = \text{input term}$

- $\Delta x[t] = a^t\Delta x[0] + \sum_{i=1}^{t} a^{t-i} b\Delta u[i-1]$

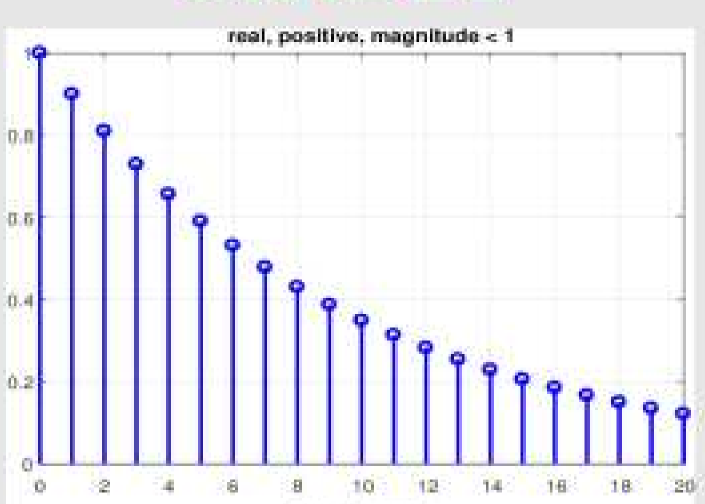
IC term

input term **(discrete convolution)**

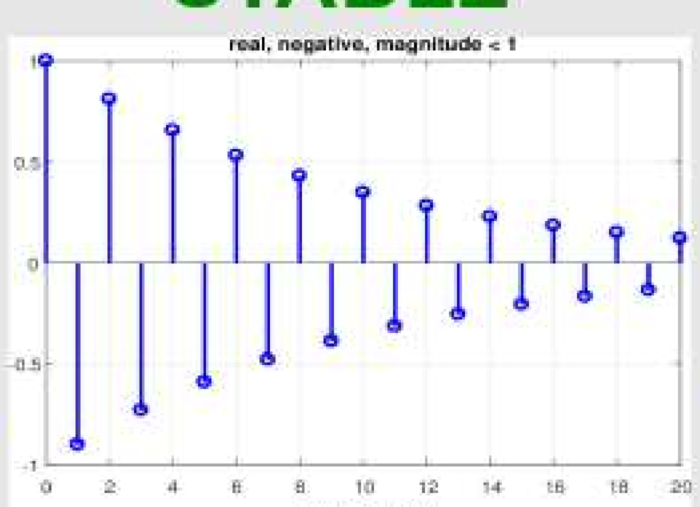
- Initial Condition term: $a^t\Delta x[0]$



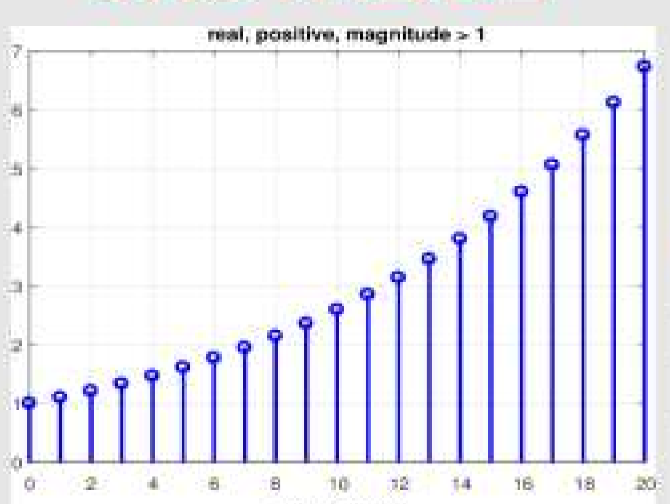
0<a<1: dies down
STABLE



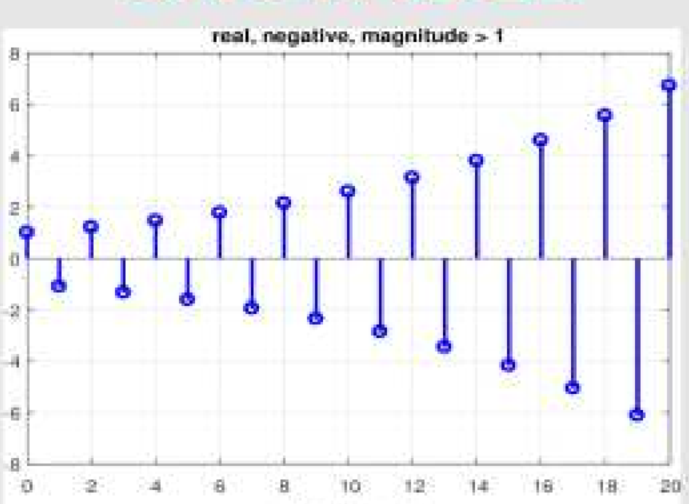
-1<a<0: dies down
STABLE



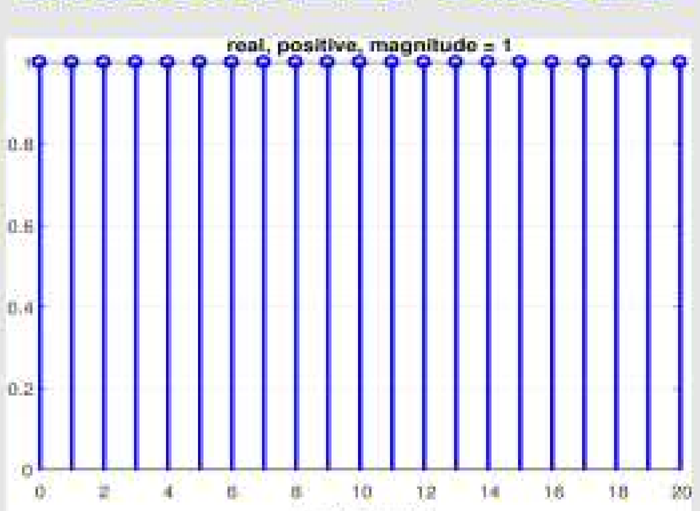
a>1: blows up
UNSTABLE



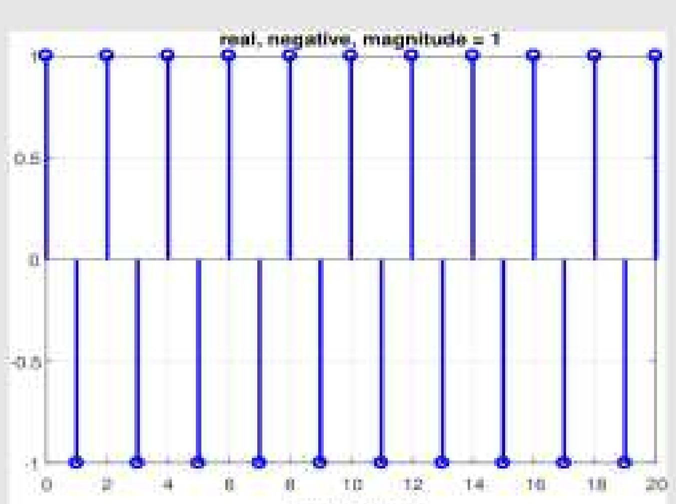
a<-1: blows up
UNSTABLE



a=1: constant
MARGINALLY STABLE



a=-1: constant
MARGINALLY STABLE



Stability for Discrete Time Systems

- The scalar case: $\Delta x[t+1] = a\Delta x[t] + b\Delta u[t]$, IC $\Delta x[0]$
  - real
  - ➔ [already linear(ized); everything is real]

$t=0$: $\Delta x[1] = a\Delta x[0] + b\Delta u[0]$

$t=1$: $\Delta x[2] = a\Delta x[1] + b\Delta u[0] = a^2\Delta x[0] + ab\Delta u[0] + b\Delta u[1]$

$t=2$: $\Delta x[3] = a\Delta x[2] + b\Delta u[2] = a^3\Delta x[0] + a^2 b\Delta u[0] + ab\Delta u[1] + b\Delta u[2]$

$\vdots$

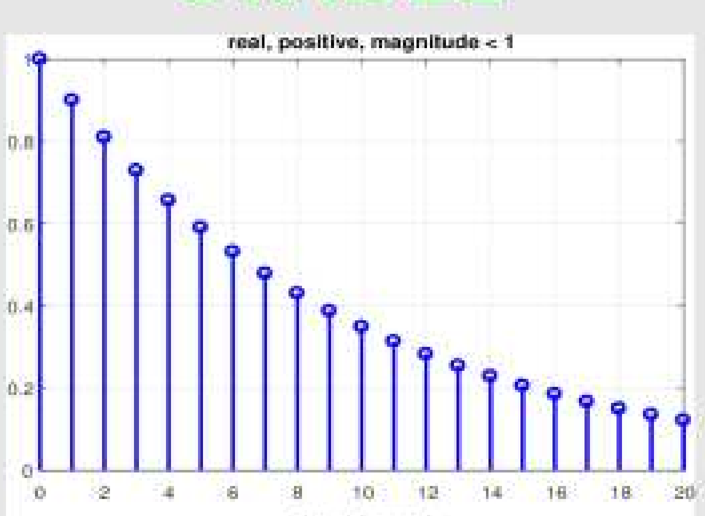
$\Delta x[t] = a^t\Delta x[0] + a^{t-1}b\Delta u[0] + a^{t-2}b\Delta u[1] + \cdots + ab\Delta u[t-2] + b\Delta u[t-1]$

$= \underbrace{a^t\Delta x[0]}_{\text{I.C. term}} + \underbrace{\sum_{i=1}^{t} a^{t-i} b\Delta u[i-1]}_{\text{discrete-time convolution}}$ = input term

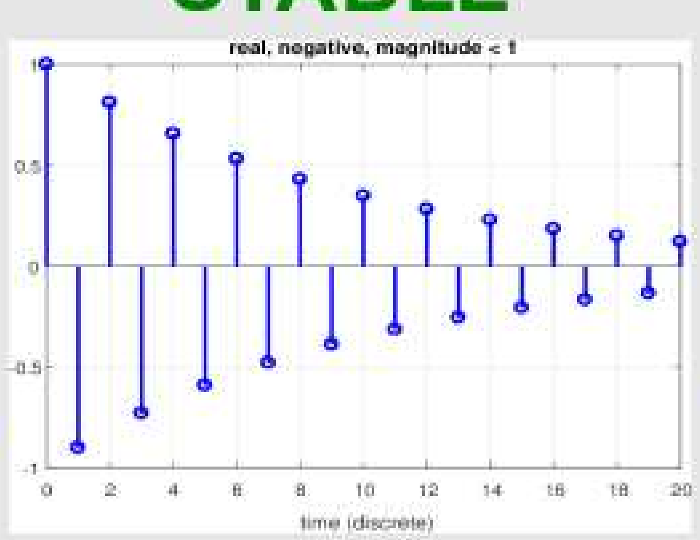
- $\Delta x[t] = \underbrace{a^t\Delta x[0]}_{\text{IC term}} + \underbrace{\sum_{i=1}^{t} a^{t-i} b\Delta u[i-1]}_{\text{input term (discrete convolution)}}$

- Initial Condition term: $a^t\Delta x[0]$

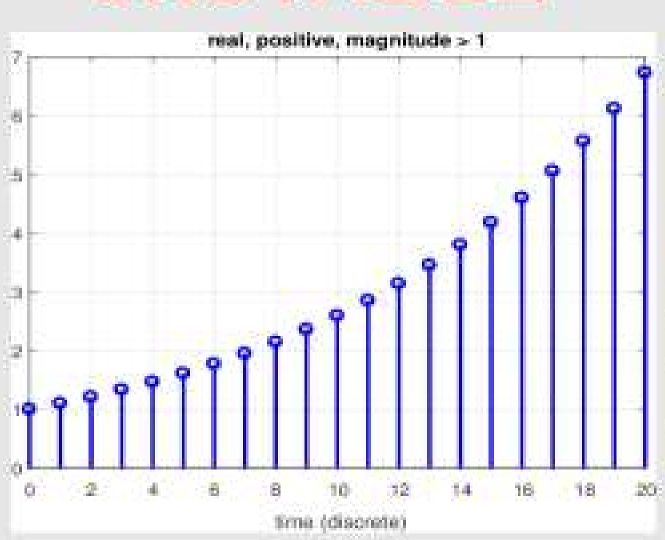
0<a<1: dies down
STABLE



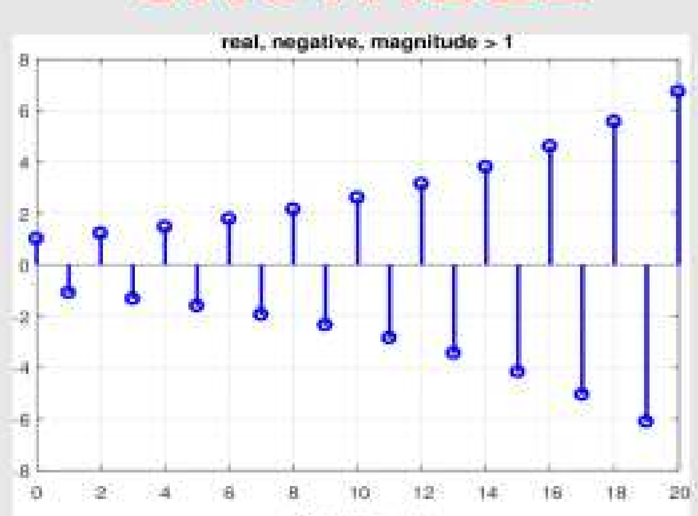
-1<a<0: dies down
STABLE



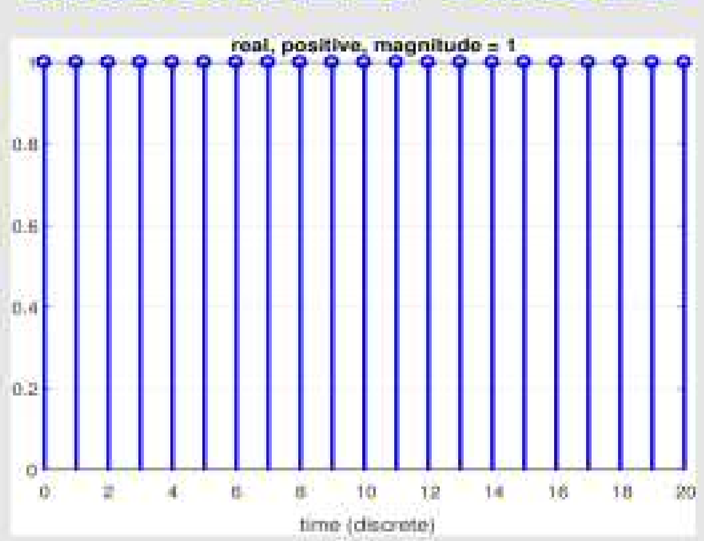
a>1: blows up
UNSTABLE



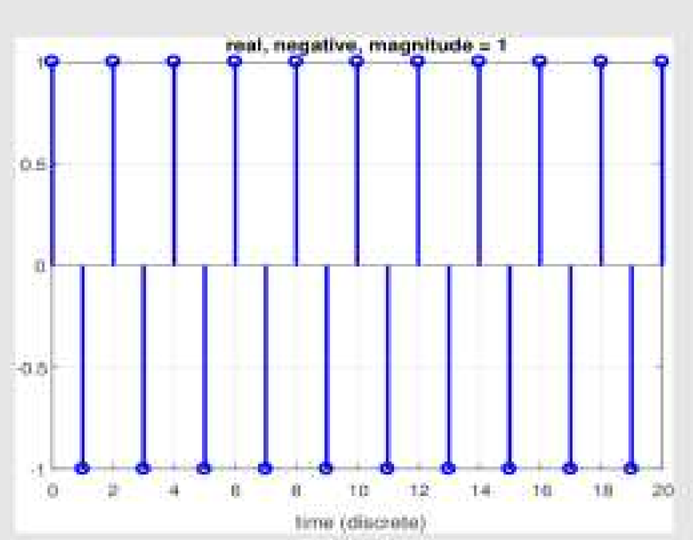
a<-1: blows up
UNSTABLE



a=1: constant
MARGINALLY STABLE



a=-1: constant
MARGINALLY STABLE



Stability for Discrete Time Systems

- The scalar case: $\Delta x[t+1] = a\Delta x[t] + b\Delta u[t]$, IC $\Delta x[0]$
  - real
  - ➔ [already linear(ized); everything is real]

$t=0$: $\Delta x[1] = a\Delta x[0] + b\Delta u[0]$

$t=1$: $\Delta x[2] = a\Delta x[1] + b\Delta u[0] = a^2\Delta x[0] + ab\Delta u[0] + b\Delta u[1]$

$t=2$: $\Delta x[3] = a\Delta x[2] + b\Delta u[2] = a^3\Delta x[0] + a^2 b\Delta u[0] + ab\Delta u[1] + b\Delta u[2]$

$\vdots$

$$\Delta x[t] = a^t\Delta x[0] + a^{t-1}b\Delta u[0] + a^{t-2}b\Delta u[1] + \cdots + ab\Delta u[t-2] + b\Delta u[t-1]$$

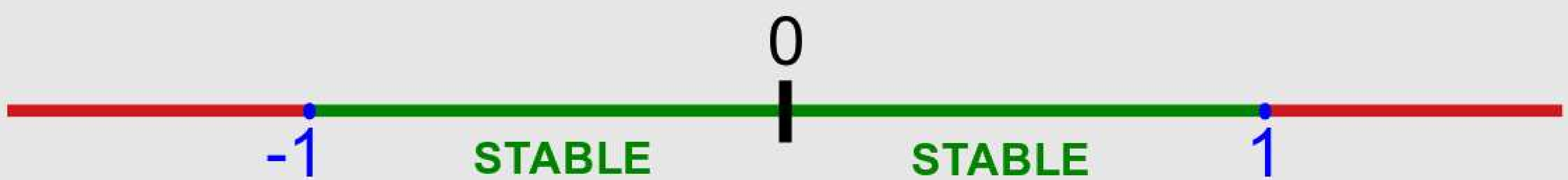
$$= \underbrace{a^t\Delta x[0]}_{\text{I.C. term}} + \underbrace{\sum_{i=1}^{t} a^{t-i}b\Delta u[i-1]}_{\text{discrete-time convolution} = \text{input term}}$$

- $\Delta x[t] = a^t\Delta x[0] + \sum_{i=1}^{t} a^{t-i}b\Delta u[i-1]$

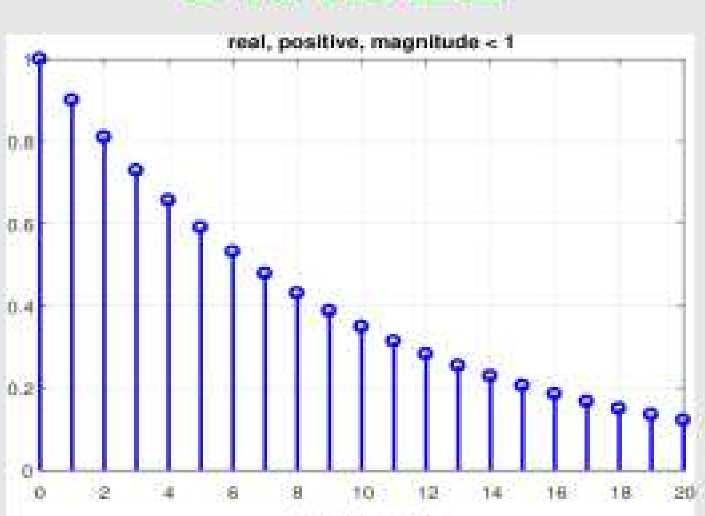
IC term

input term **(discrete convolution)**

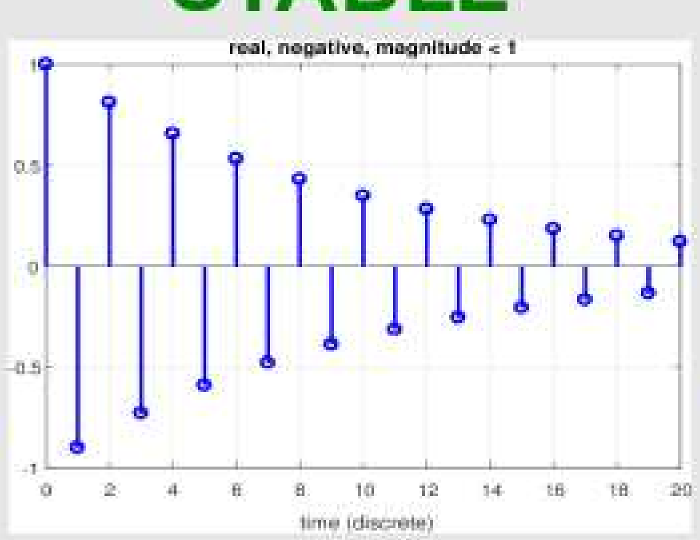
- Initial Condition term: $a^t\Delta x[0]$



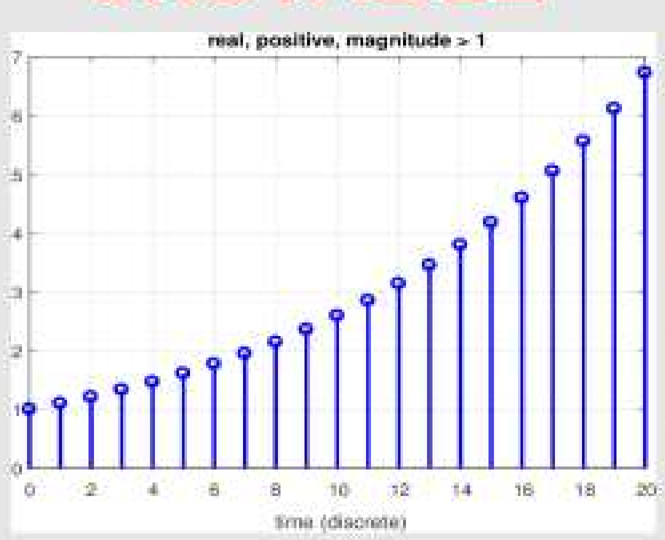
0<a<1: dies down
STABLE



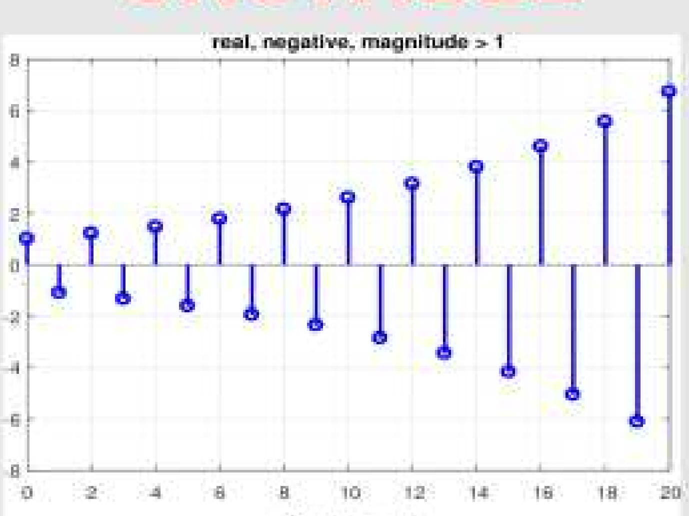
-1<a<0: dies down
STABLE



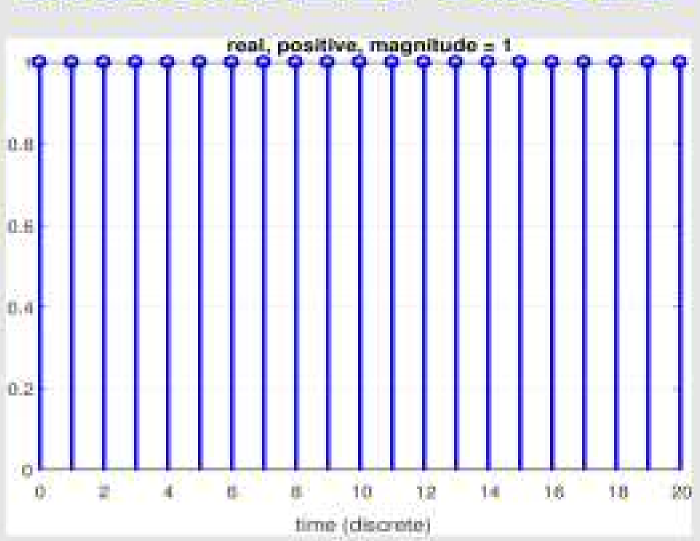
a>1: blows up
UNSTABLE



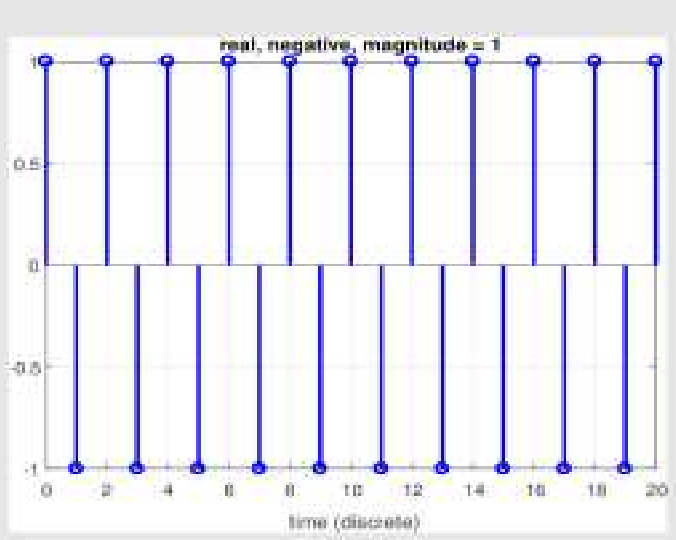
a<-1: blows up
UNSTABLE



a=1: constant
MARGINALLY STABLE



a=-1: constant
MARGINALLY STABLE



Stability for Discrete Time Systems

- The scalar case: $\Delta x[t+1] = a\Delta x[t] + b\Delta u[t]$, IC $\Delta x[0]$
  - real
  - → [already linear(ized); everything is real]

$t=0$: $\Delta x[1] = a\Delta x[0] + b\Delta u[0]$

$t=1$: $\Delta x[2] = a\Delta x[1] + b\Delta u[1] = a^2\Delta x[0] + ab\Delta u[0] + b\Delta u[1]$

$t=2$: $\Delta x[3] = a\Delta x[2] + b\Delta u[2] = a^3\Delta x[0] + a^2 b\Delta u[0] + ab\Delta u[1] + b\Delta u[2]$

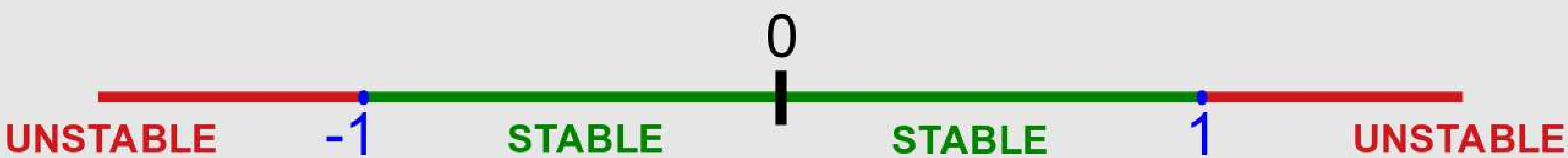
$\vdots$

$$\Delta x[t] = a^t\Delta x[0] + a^{t-1}b\Delta u[0] + a^{t-2}b\Delta u[1] + \cdots + ab\Delta u[t-2] + b\Delta u[t-1]$$

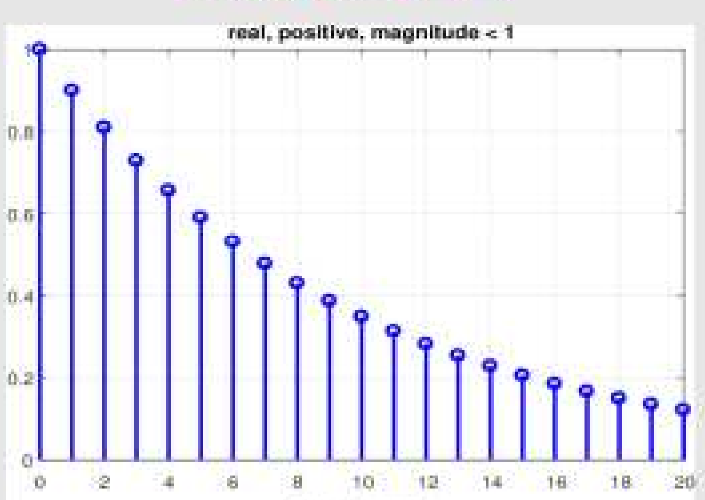
$$= \underbrace{a^t\Delta x[0]}_{\text{I.C. term}} + \underbrace{\sum_{i=1}^{t} a^{t-i}b\Delta u[i-1]}_{\text{discrete-time convolution}} = \text{input term}$$

- $$\Delta x[t] = \underbrace{a^t\Delta x[0]}_{\text{IC term}} + \underbrace{\sum_{i=1}^{t} a^{t-i}b\Delta u[i-1]}_{\text{input term (discrete convolution)}}$$

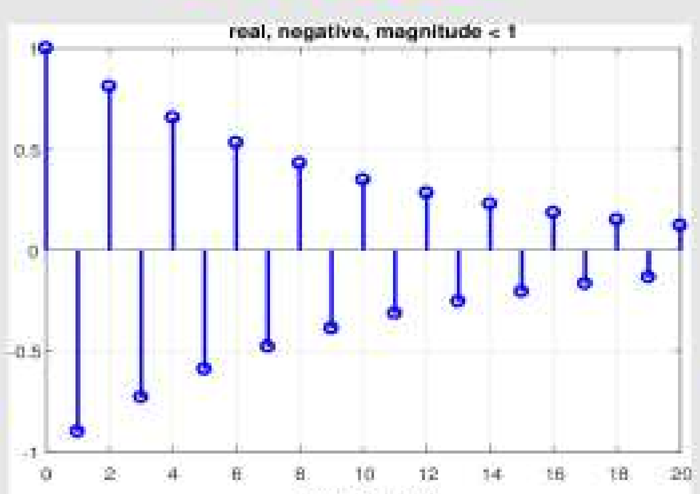
- Initial Condition term: $a^t\Delta x[0]$



0<a<1: dies down
STABLE

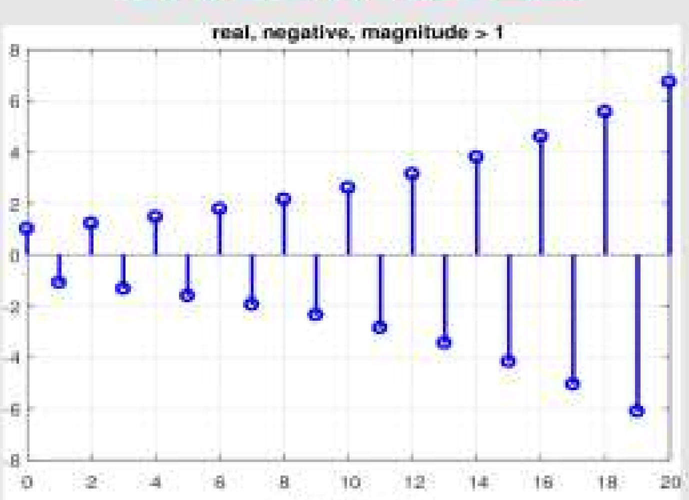


-1<a<0: dies down
STABLE

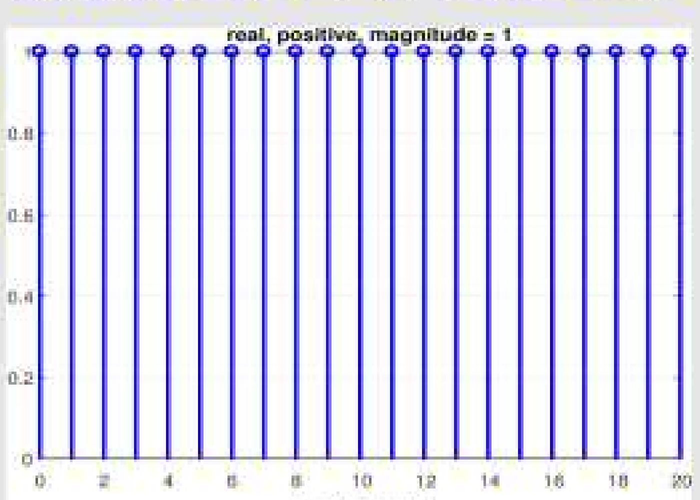


a>1: blows up
UNSTABLE

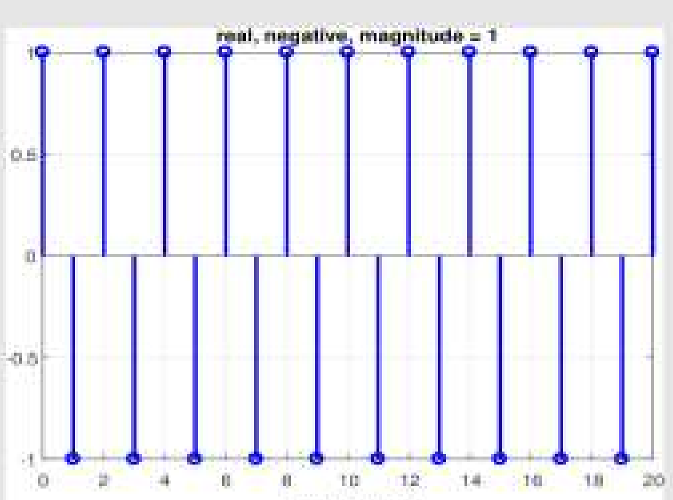
a<-1: blows up
UNSTABLE



a=1: constant
MARGINALLY STABLE



a=-1: constant
MARGINALLY STABLE



Stability for Discrete Time Systems

- The scalar case: $\Delta x[t+1] = a\Delta x[t] + b\Delta u[t]$, IC Δx[0]
  - real
  - ➔ [already linear(ized); everything is real]

$t=0$: $\Delta x[1] = a\Delta x[0] + b\Delta u[0]$

$t=1$: $\Delta x[2] = a\Delta x[1] + b\Delta u[0] = a^2\Delta x[0] + ab\Delta u[0] + b\Delta u[1]$

$t=2$: $\Delta x[3] = a\Delta x[2] + b\Delta u[2] = a^3\Delta x[0] + a^2 b\Delta u[0] + ab\Delta u[1] + b\Delta u[2]$

$\vdots$

$\Delta x[t] = a^t\Delta x[0] + a^{t-1} b\Delta u[0] + a^{t-2} b\Delta u[1] + \cdots + ab\Delta u[t-2] + b\Delta u[t-1]$

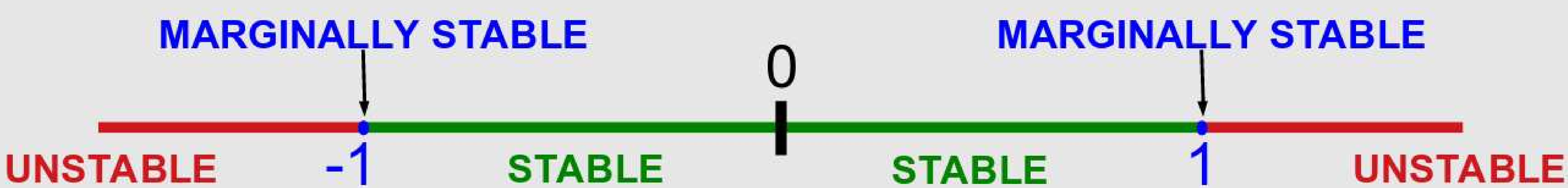
$= \underbrace{a^t\Delta x[0]}_{\text{I.C. term}} + \underbrace{\sum_{i=1}^{t} a^{t-i} b\Delta u[i-1]}_{\text{discrete-time convolution}} = \text{input term}$

- $\Delta x[t] = \underbrace{a^t\Delta x[0]}_{\text{IC term}} + \sum_{i=1}^{t} a^{t-i} b\Delta u[i-1]$

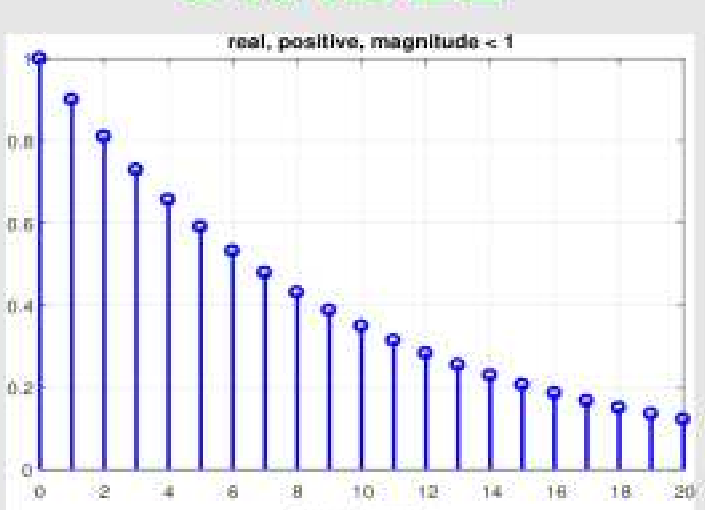
IC term

input term **(discrete convolution)**

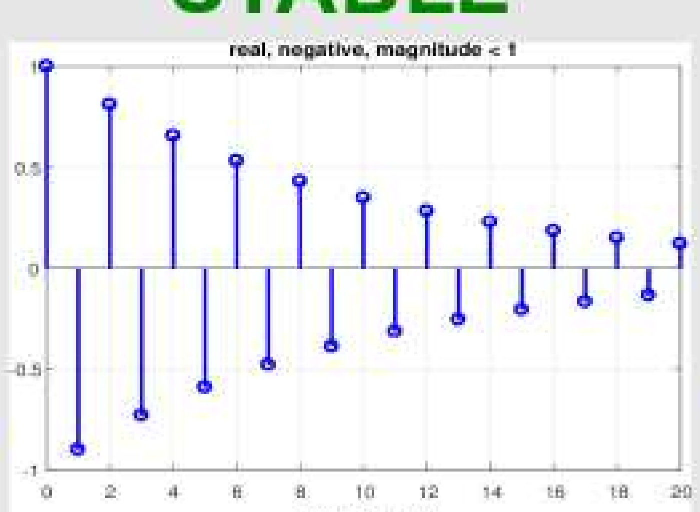
- Initial Condition term: $a^t\Delta x[0]$



0<a<1: dies down
STABLE

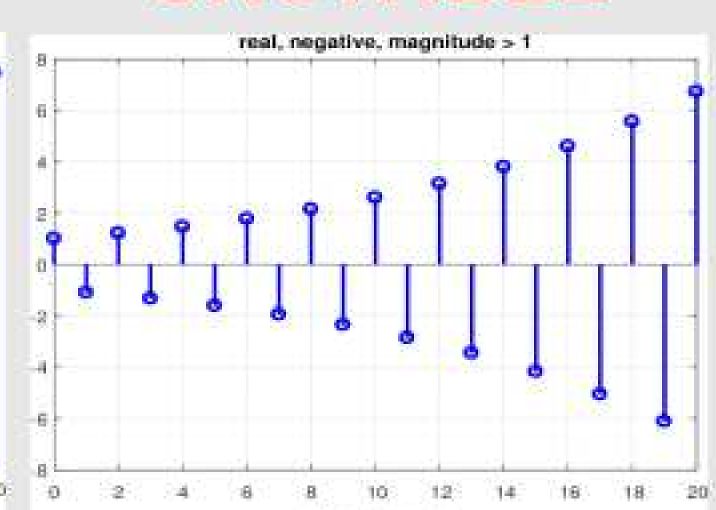


-1<a<0: dies down
STABLE

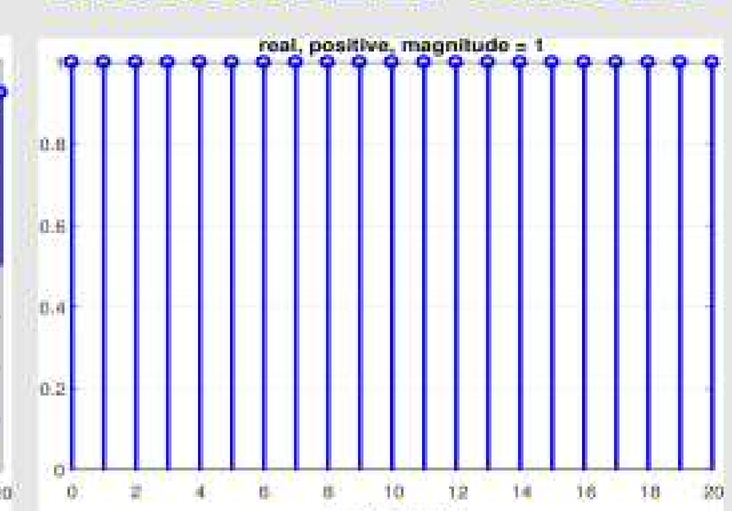


a>1: blows up
UNSTABLE

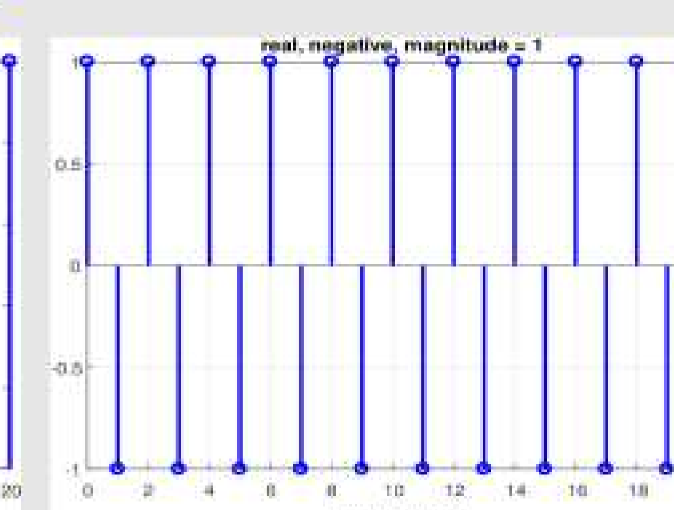
a<-1: blows up
UNSTABLE



a=1: constant
MARGINALLY STABLE



a=-1: constant
MARGINALLY STABLE



Scalar Discrete-Time Stability (contd.)

- Solution: $\Delta x[t] = a^t \Delta x[0] + \sum_{i=1}^{t} a^{t-i} b \Delta u[i-1]$

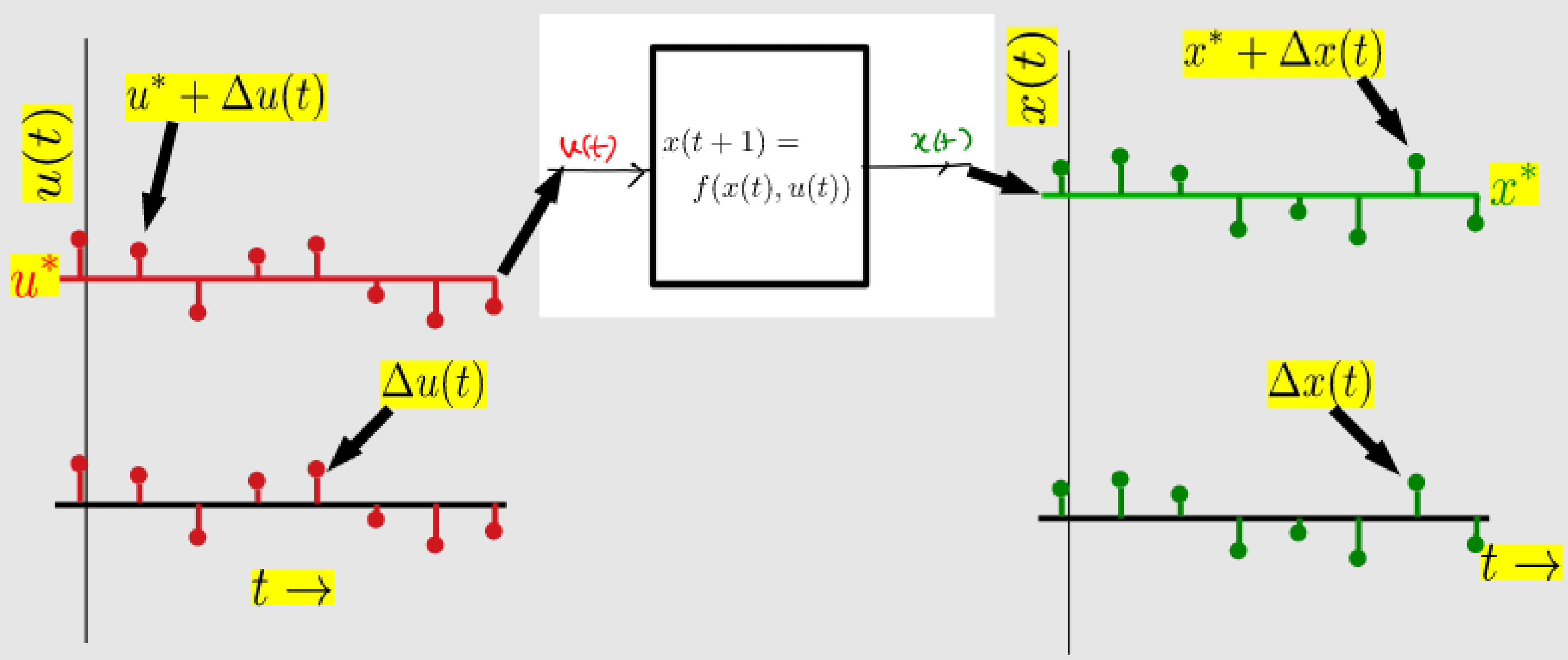
input term **(d. convolution)**

Scalar Discrete-Time Stability (contd.)

- Solution: $\Delta x[t] = a^t \Delta x[0] + \sum_{i=1}^{t} a^{t-i} b \Delta u[i-1]$

  input term **(d. convolution)**
- Can show (see handwritten notes):
  - if |a|<1: Δx(t) bounded if Δu(t) bounded: **BIBO stable**

|a|<1: BIBO STABLE



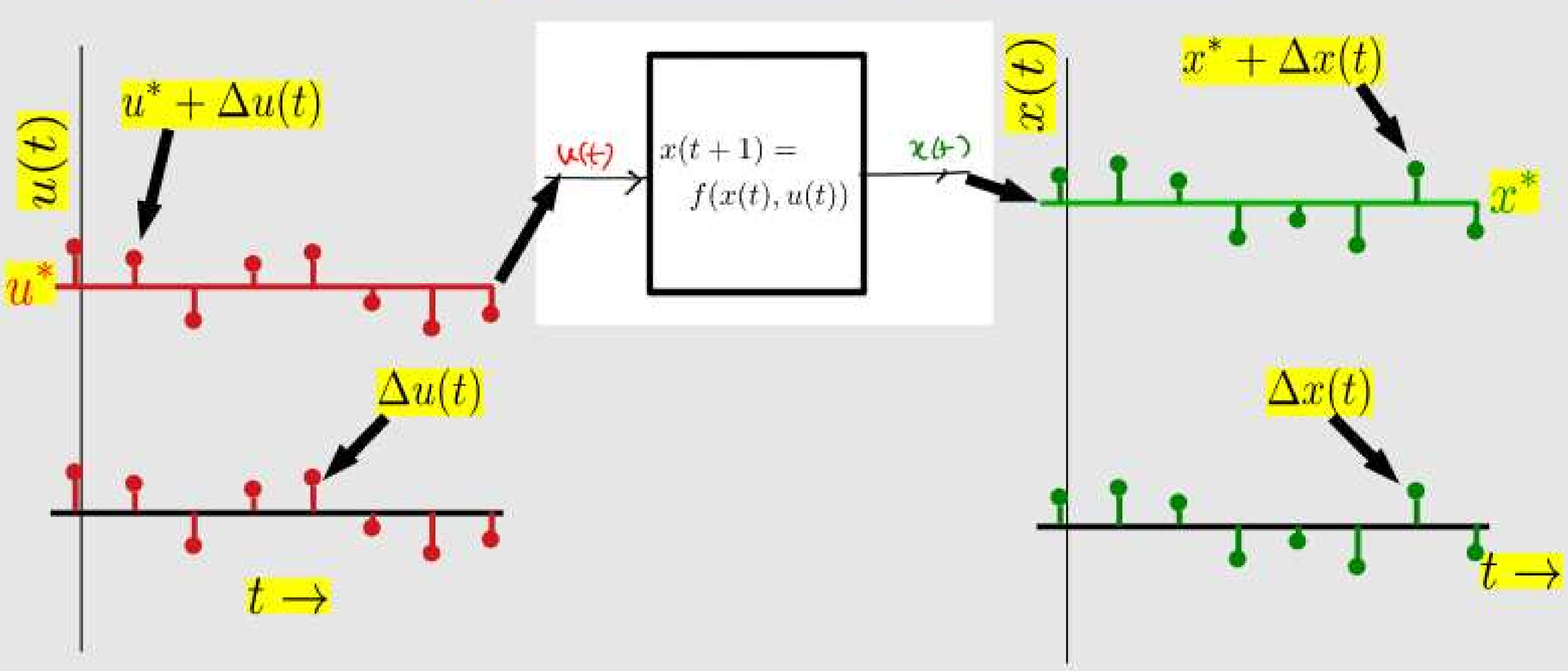
Scalar Discrete-Time Stability (contd.)

- Solution: $\Delta x[t] = a^t \Delta x[0] + \sum_{i=1}^{t} a^{t-i} b \Delta u[i-1]$

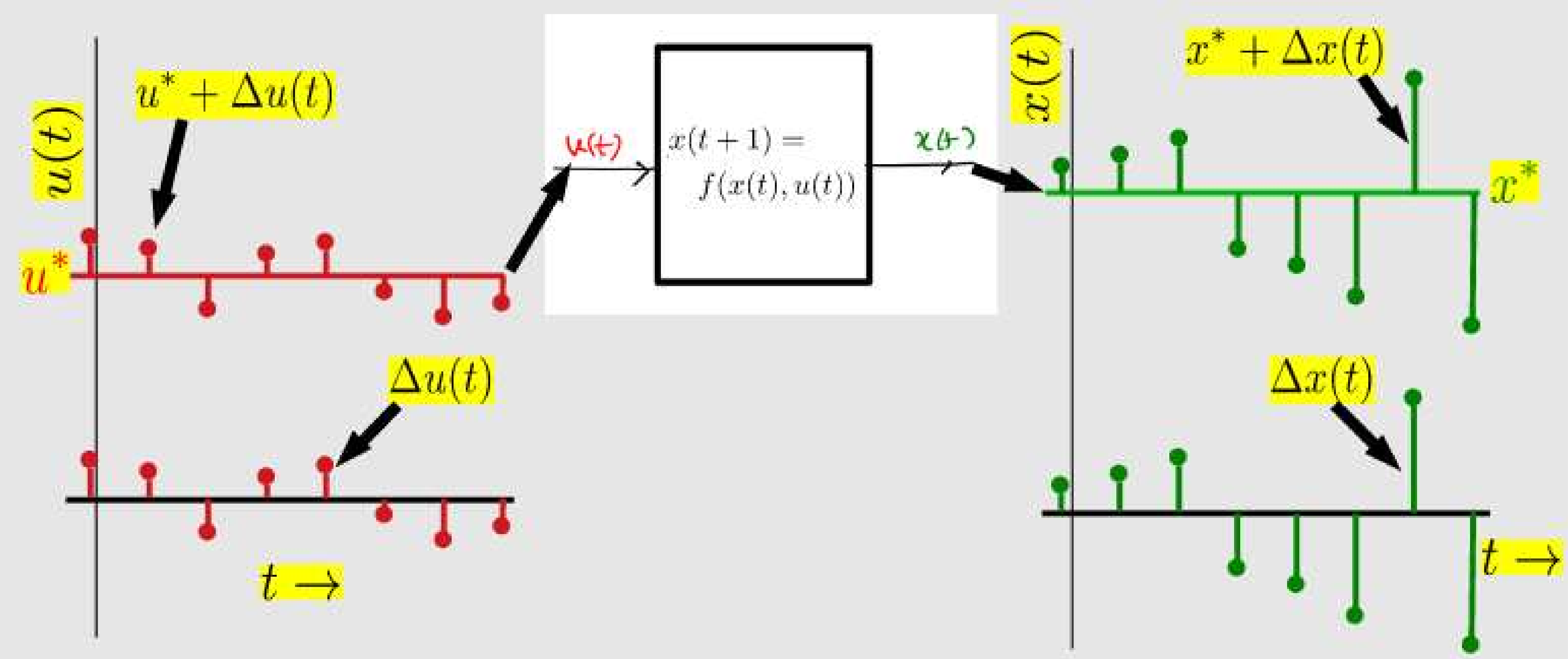
  input term **(d. convolution)**

- Can show (see handwritten notes):
  - if |a|<1: Δx(t) bounded if Δu(t) bounded: **BIBO stable**
  - if |a|>1: Δx(t) unbounded even if Δu(t) bounded: **UNSTABLE**
  - if |a|=1: Δx(t) unbounded even if Δu(t) bounded: **UNSTABLE**

|a|<1: BIBO STABLE



|a|≥1: UNSTABLE



Discrete Time Stability: the Vector Case

- The vector case: $\Delta\vec{x}[t+1] = A\Delta\vec{x}[t] + B\Delta\vec{u}[t]$

Discrete Time Stability: the Vector Case

- The vector case: $\Delta\vec{x}[t+1] = A\Delta\vec{x}[t] + B\Delta\vec{u}[t]$

real matrices

Discrete Time Stability: the Vector Case

- The vector case: $\Delta\vec{x}[t+1] = A\Delta\vec{x}[t] + B\Delta\vec{u}[t]$
  - [already linear(ized); everything is real] real matrices

Discrete Time Stability: the Vector Case

- The vector case: $\Delta\vec{x}[t+1] = A\Delta\vec{x}[t] + B\Delta\vec{u}[t]$
  - [already linear(ized); everything is real] real matrices
- Eigendecompose A: $A = P\Lambda P^{-1}$
- Define: $\Delta\vec{y}[t] \triangleq P^{-1}\Delta\vec{x}[t] \quad \Leftrightarrow \quad \Delta\vec{x}[t] \triangleq P\Delta\vec{y}[t]$
  - $\Delta\vec{b}[t] \triangleq P^{-1}\Delta\vec{u}[t]$

Discrete Time Stability: the Vector Case

- The vector case: $\Delta\vec{x}[t+1] = A\Delta\vec{x}[t] + B\Delta\vec{u}[t]$
  - [already linear(ized); everything is real] real matrices
- Eigendecompose A: $A = P\Lambda P^{-1}$
- Define: $\Delta\vec{y}[t] \triangleq P^{-1}\Delta\vec{x}[t] \quad \Leftrightarrow \quad \Delta\vec{x}[t] \triangleq P\Delta\vec{y}[t]$
  - $\Delta\vec{b}[t] \triangleq P^{-1}\Delta\vec{u}[t]$
- Decomposed system: $\Delta\vec{y}_i[t+1] = \lambda_i\Delta\vec{y}[t] + \Delta\vec{b}_i[t]$ $\quad i = 1, \cdots, n$

Discrete Time Stability: the Vector Case

- The vector case: $\Delta\vec{x}[t+1] = A\Delta\vec{x}[t] + B\Delta\vec{u}[t]$
  - [already linear(ized); everything is real] real matrices
- Eigendecompose A: $A = P\Lambda P^{-1}$
- Define: $\Delta\vec{y}[t] \triangleq P^{-1}\Delta\vec{x}[t] \quad \Leftrightarrow \quad \Delta\vec{x}[t] \triangleq P\Delta\vec{y}[t]$
  - $\Delta\vec{b}[t] \triangleq P^{-1}\Delta\vec{u}[t]$
- Decomposed system: $\Delta\vec{y}_i[t+1] = \underset{\text{scalar}}{\lambda_i}\Delta\vec{y}[t] + \Delta\vec{b}_i[t]$, $i = 1, \cdots, n$

Discrete Time Stability: the Vector Case

- The vector case: $\Delta\vec{x}[t+1] = A\Delta\vec{x}[t] + B\Delta\vec{u}[t]$
  - [already linear(ized); everything is real] real matrices
- Eigendecompose A: $A = P\Lambda P^{-1}$
- Define: $\Delta\vec{y}[t] \triangleq P^{-1}\Delta\vec{x}[t] \quad \Leftrightarrow \quad \Delta\vec{x}[t] \triangleq P\Delta\vec{y}[t]$
  - $\Delta\vec{b}[t] \triangleq P^{-1}\Delta\vec{u}[t]$
- Decomposed system: $\underbrace{\Delta\vec{y}_i[t+1]}_{\text{scalar}} = \lambda_i\Delta\vec{y}[t] + \Delta\vec{b}_i[t]$, $i = 1, \cdots, n$
  - same as scalar case, but λ_i now complex

Discrete Time Stability: the Vector Case

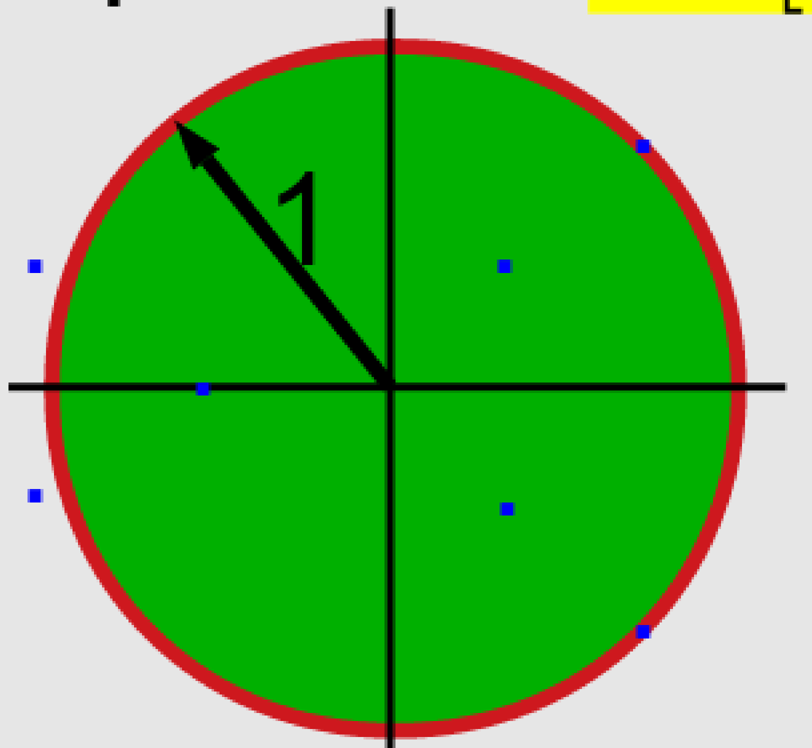
- The vector case: $\Delta\vec{x}[t+1] = A\Delta\vec{x}[t] + B\Delta\vec{u}[t]$
  - [already linear(ized); everything is real] real matrices
- Eigendecompose A: $A = P\Lambda P^{-1}$
- Define: $\Delta\vec{y}[t] \triangleq P^{-1}\Delta\vec{x}[t] \quad \Leftrightarrow \quad \Delta\vec{x}[t] \triangleq P\Delta\vec{y}[t]$
  - $\Delta\vec{b}[t] \triangleq P^{-1}\Delta\vec{u}[t]$
- Decomposed system: $\Delta\vec{y}_i[t+1] = \underbrace{\lambda_i}_{\text{scalar}}\Delta\vec{y}[t] + \Delta\vec{b}_i[t]$, $i = 1, \cdots, n$
  - same as scalar case, but λ_i now complex
  - same form for $\Delta\vec{x}[t]$ as for the continuous case
    - complex conjugate terms always present in pairs $\rightarrow$ $\Delta\vec{x}[t]$ real

Discrete Time Stability: the Vector Case

- The vector case: $\Delta\vec{x}[t+1] = A\Delta\vec{x}[t] + B\Delta\vec{u}[t]$
  - [already linear(ized); everything is real] real matrices
- Eigendecompose A: $A = P\Lambda P^{-1}$
- Define: $\Delta\vec{y}[t] \triangleq P^{-1}\Delta\vec{x}[t] \quad \Leftrightarrow \quad \Delta\vec{x}[t] \triangleq P\Delta\vec{y}[t]$
  - $\Delta\vec{b}[t] \triangleq P^{-1}\Delta\vec{u}[t]$

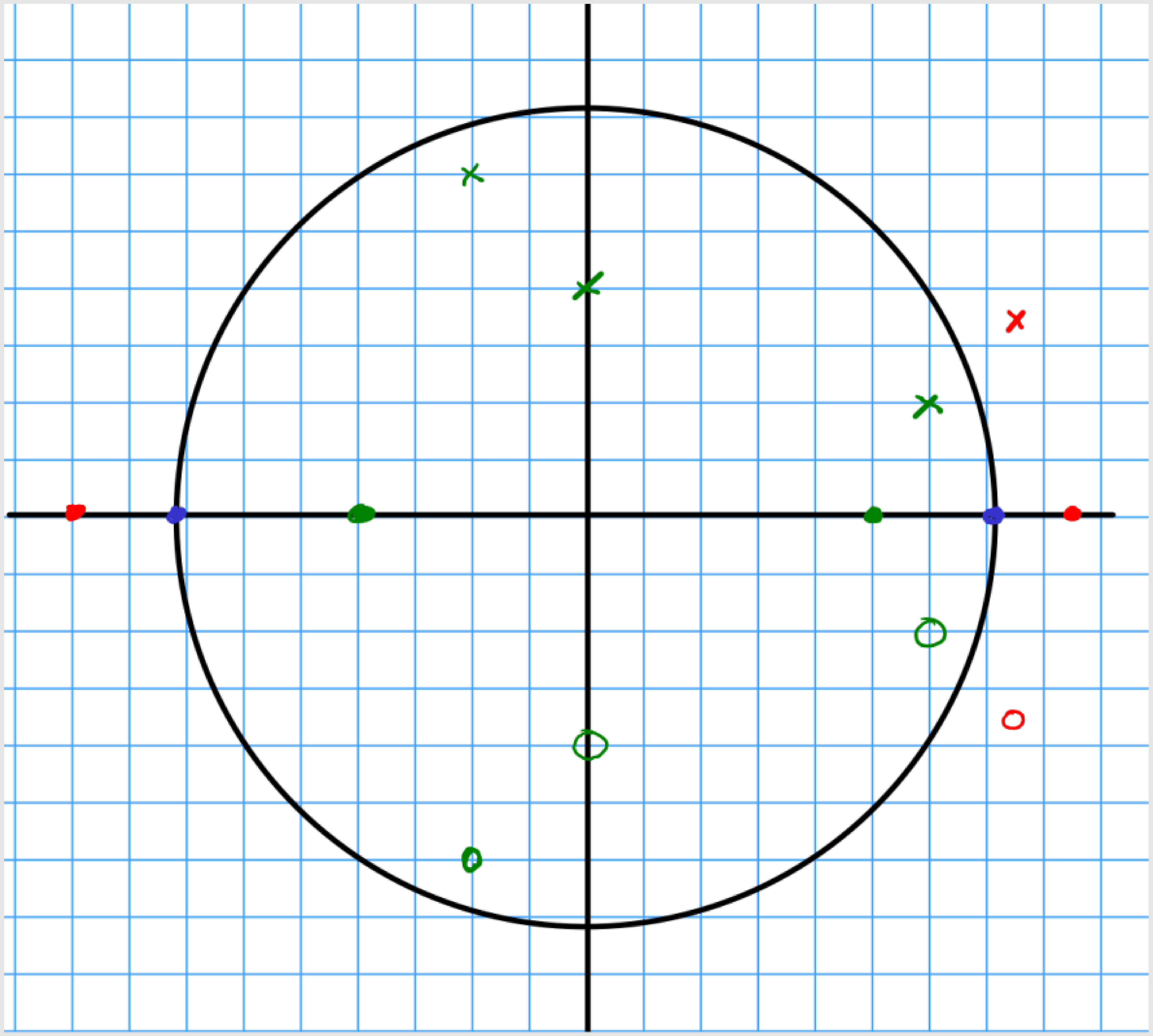
scalar $\qquad i = 1, \cdots, n$

- Decomposed system: $\Delta\vec{y}_i[t+1] = \lambda_i\Delta\vec{y}[t] + \Delta\vec{b}_i[t]$
  - same as scalar case, but λ_i now complex
  - same form for $\Delta\vec{x}[t]$ as for the continuous case
    - complex conjugate terms always present in pairs $\rightarrow$ $\Delta\vec{x}[t]$ real
- Stability:
  - BIBO stable iff $|\lambda_i| < 1$, $i = 1, \cdots, n$



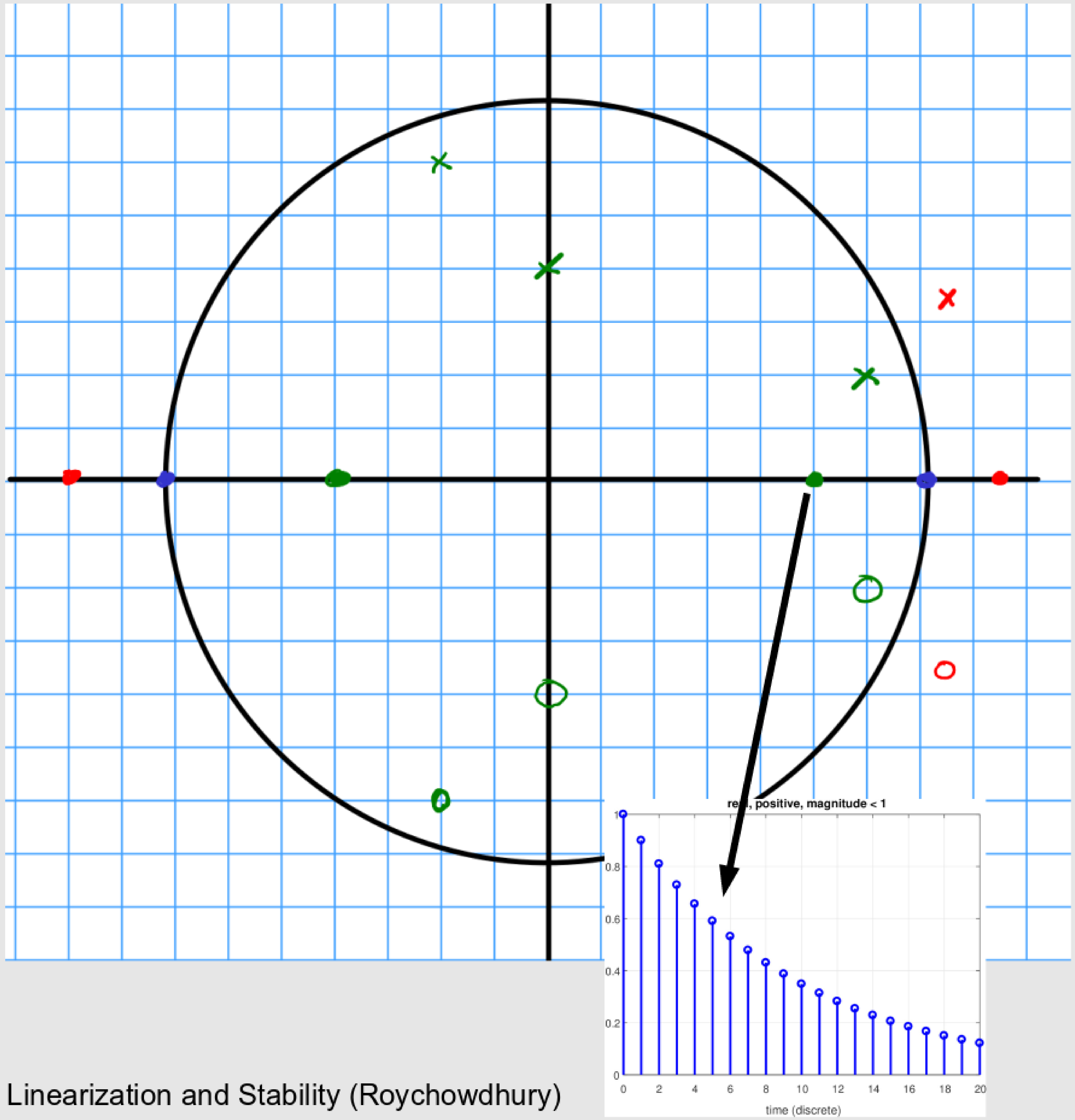
Eigenvalues and IC Responses (discrete)

complex plane for plotting eigenvalues



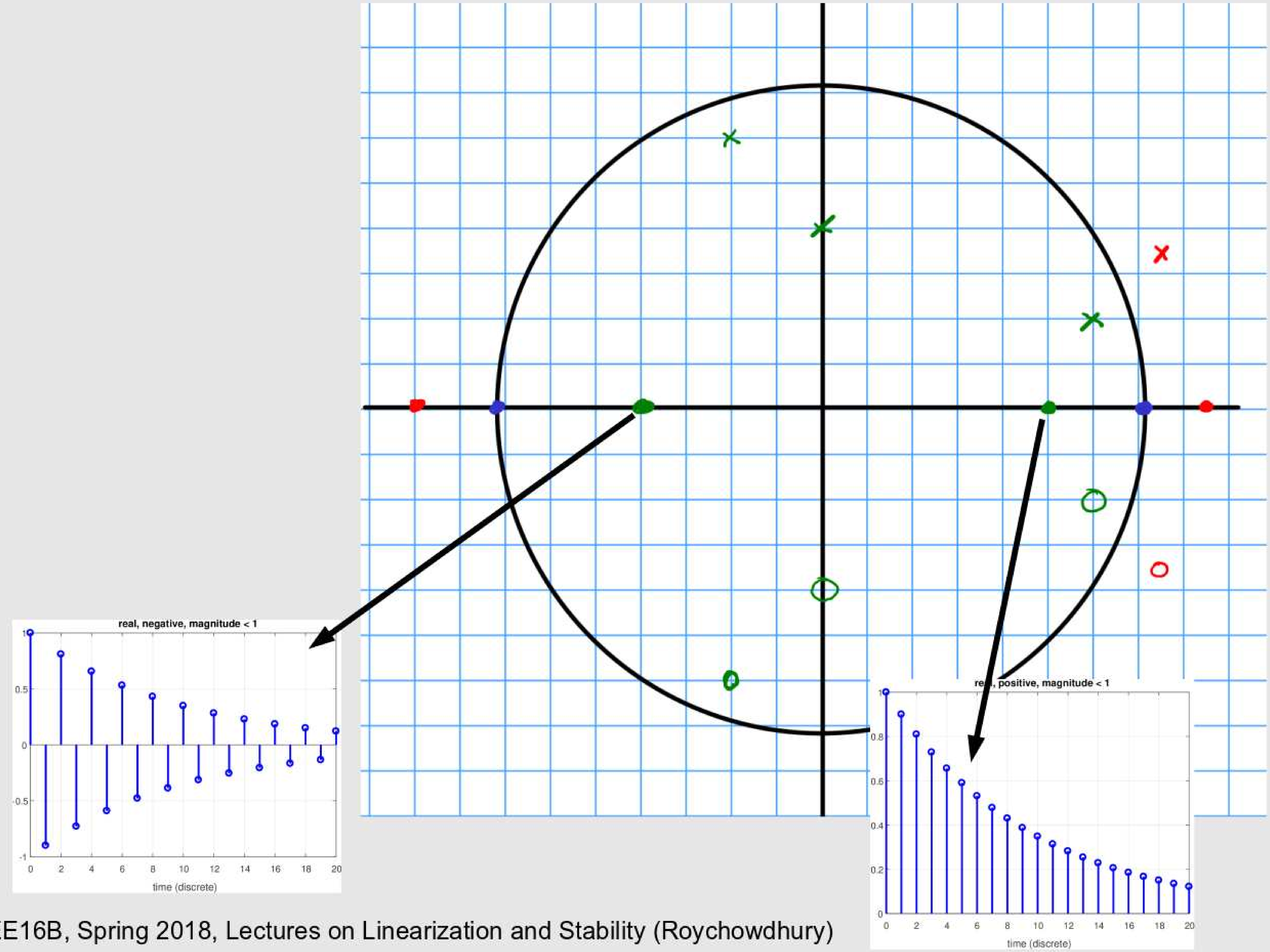
Eigenvalues and IC Responses (discrete)

complex plane for plotting eigenvalues



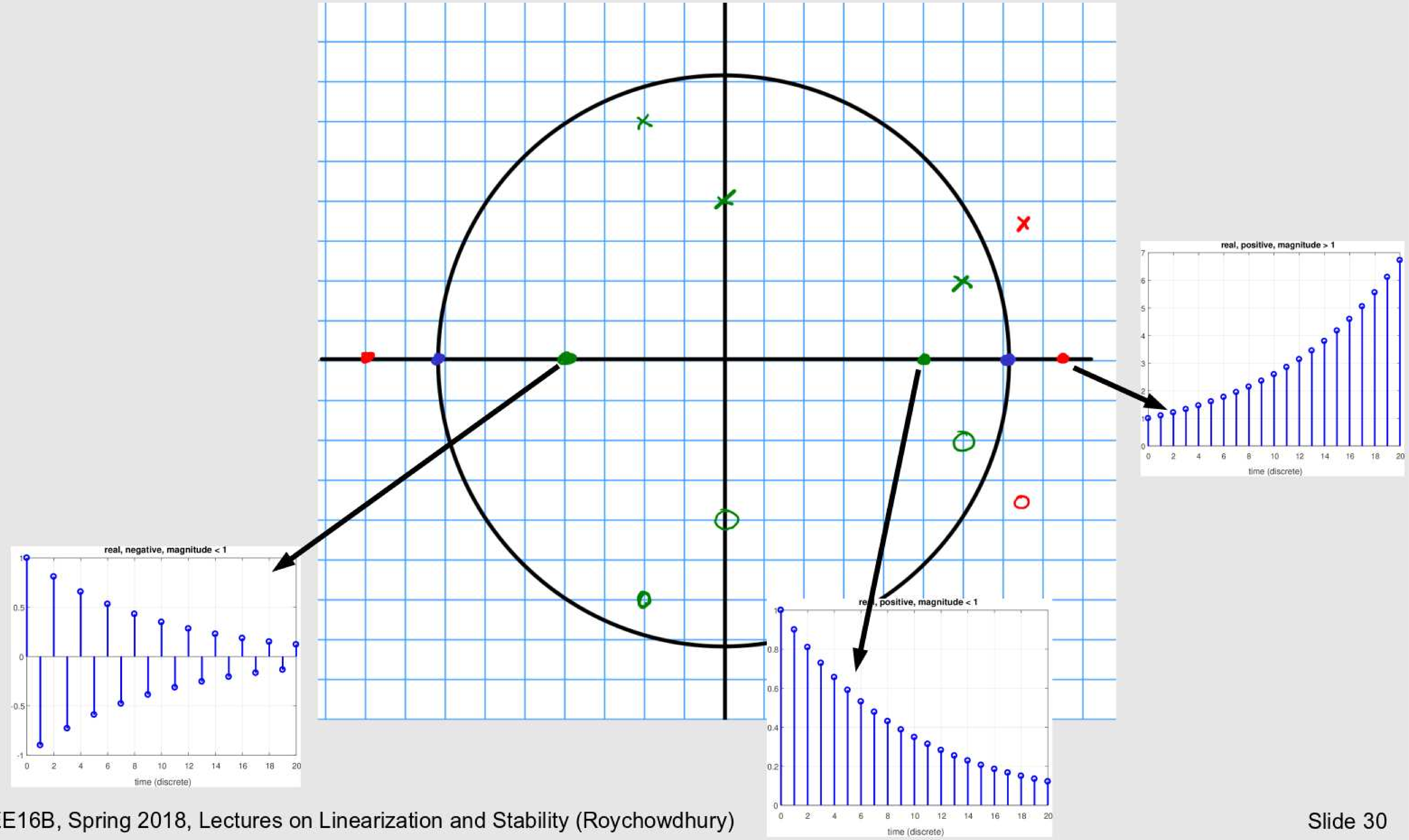
Eigenvalues and IC Responses (discrete)

complex plane for plotting eigenvalues



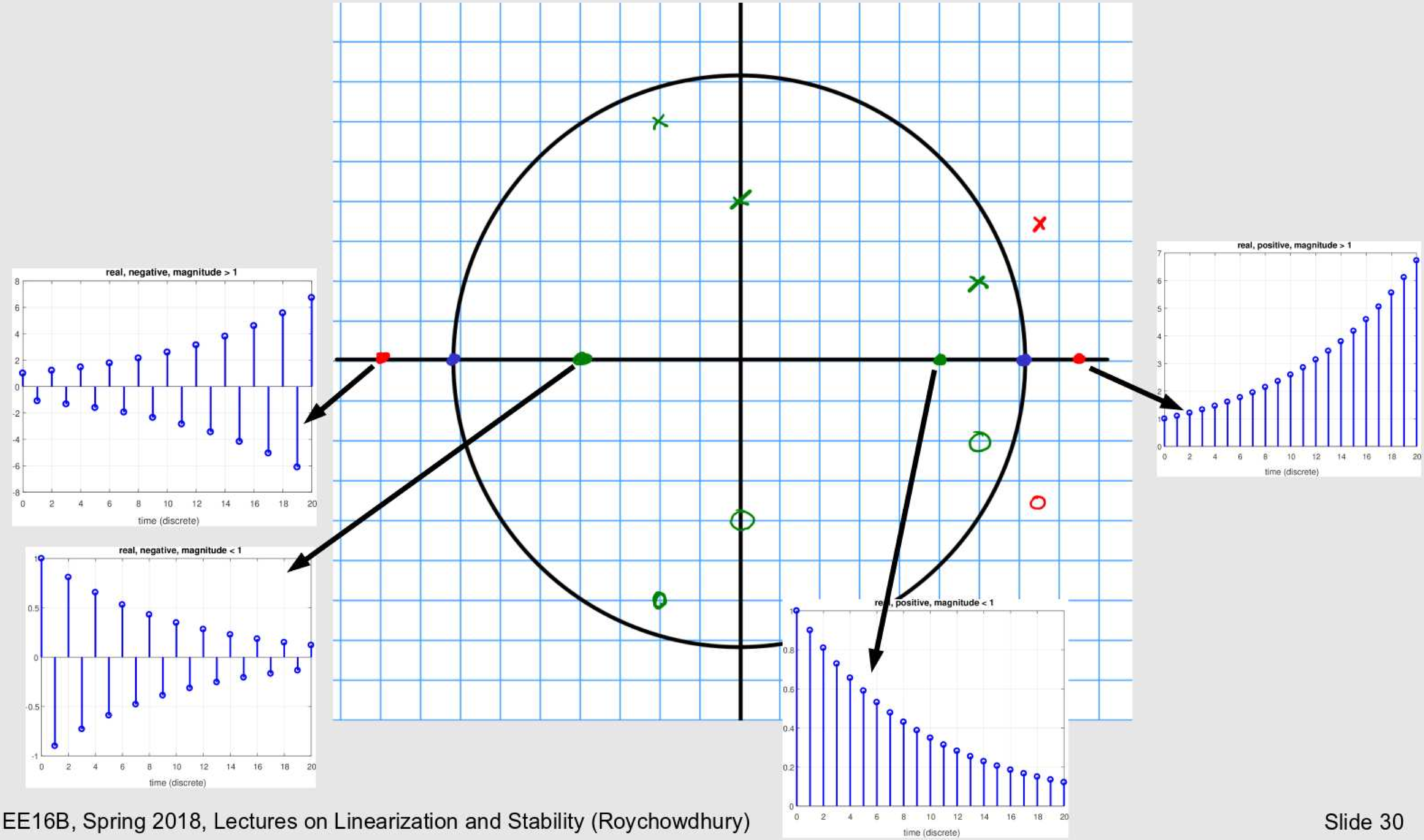
Eigenvalues and IC Responses (discrete)

complex plane for plotting eigenvalues



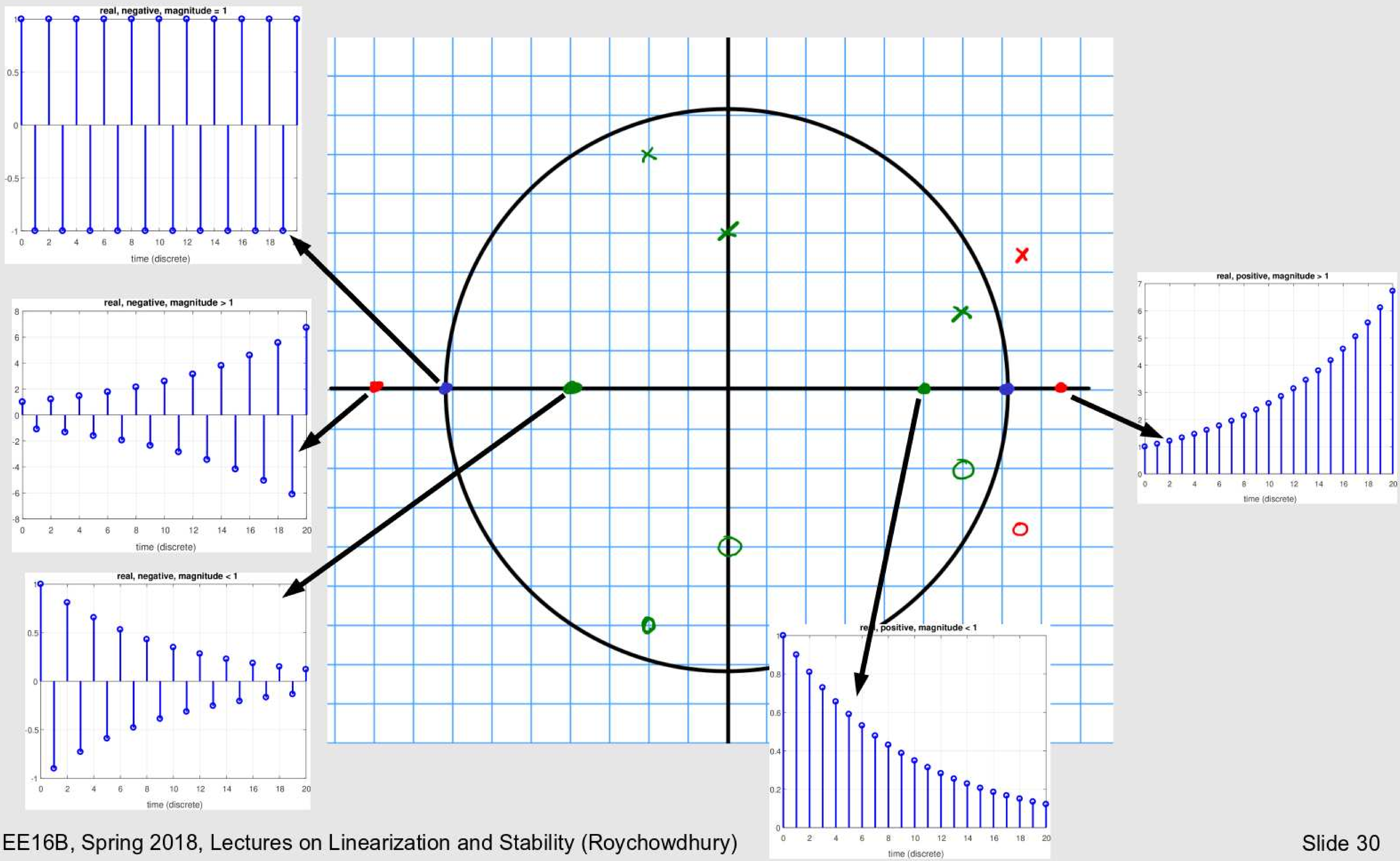
Eigenvalues and IC Responses (discrete)

complex plane for plotting eigenvalues



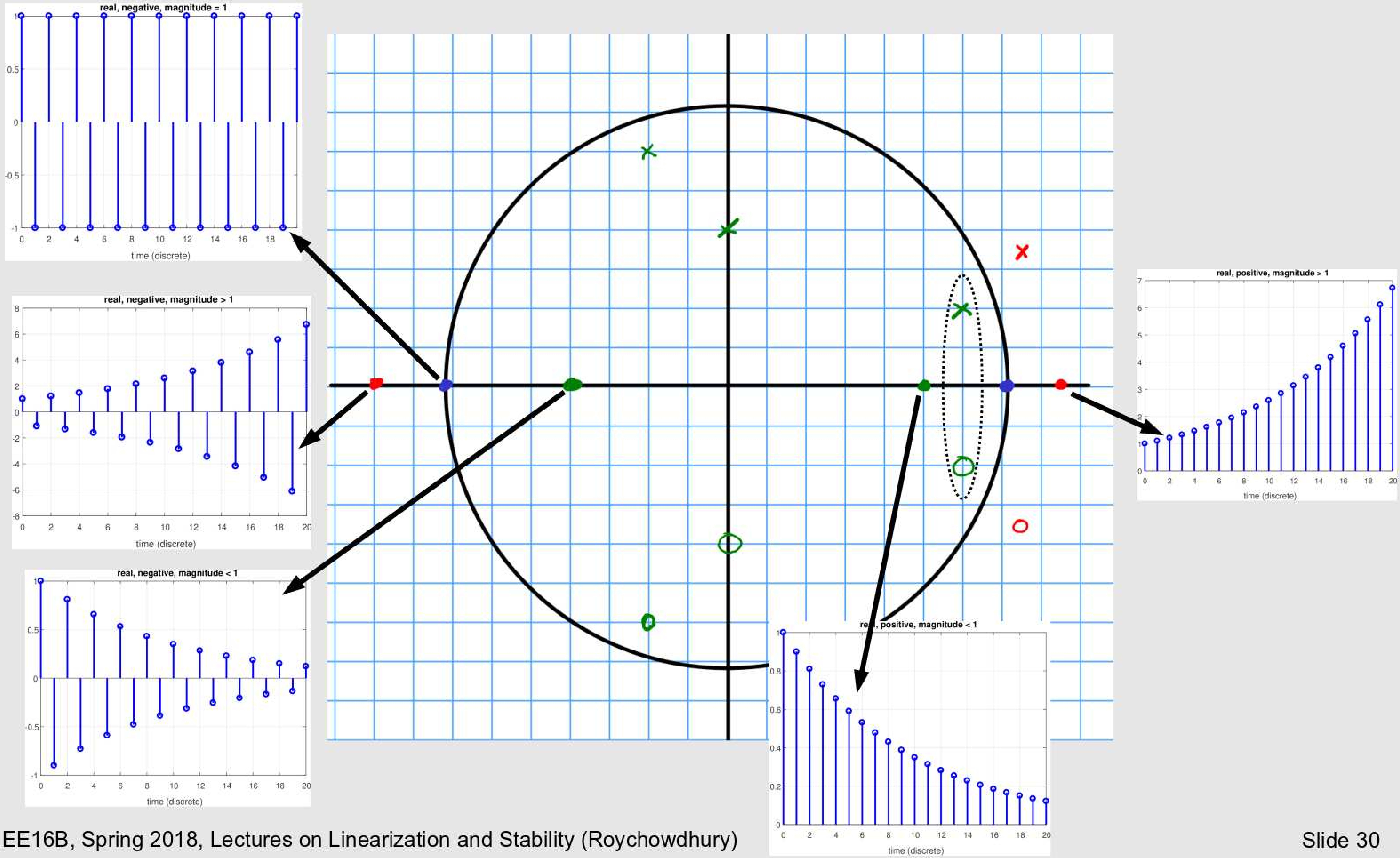
Eigenvalues and IC Responses (discrete)

complex plane for plotting eigenvalues



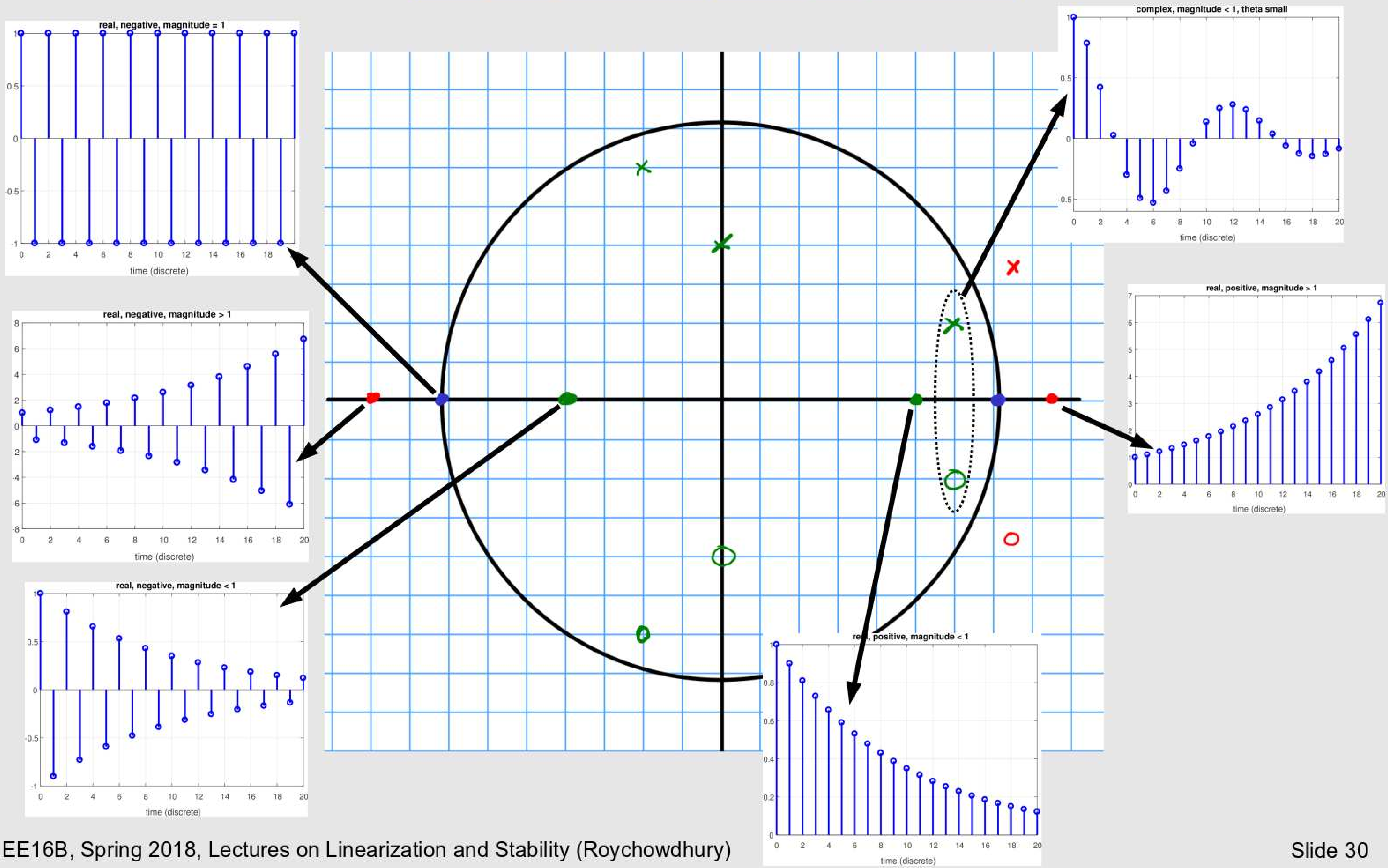
Eigenvalues and IC Responses (discrete)

complex plane for plotting eigenvalues



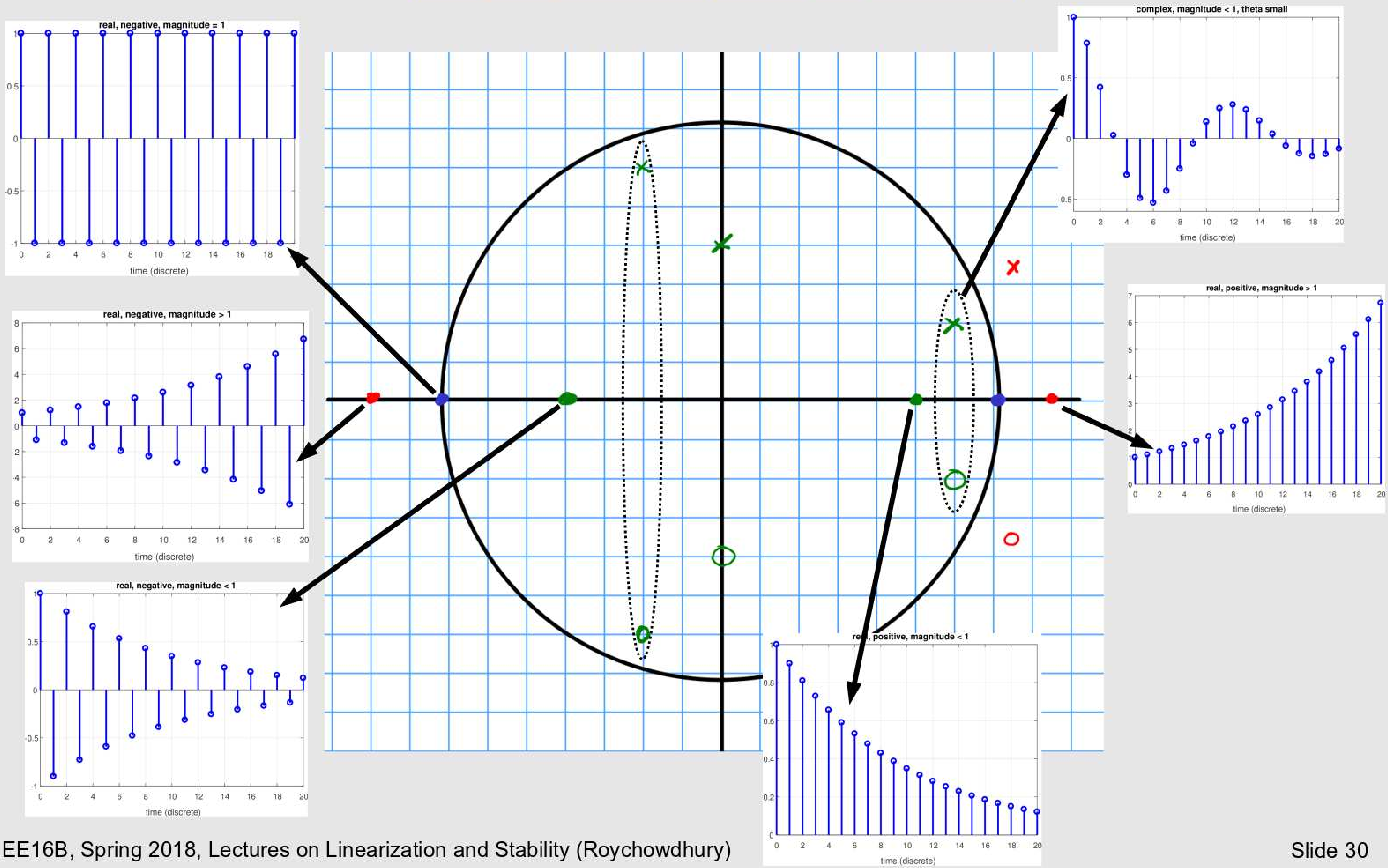
Eigenvalues and IC Responses (discrete)

complex plane for plotting eigenvalues



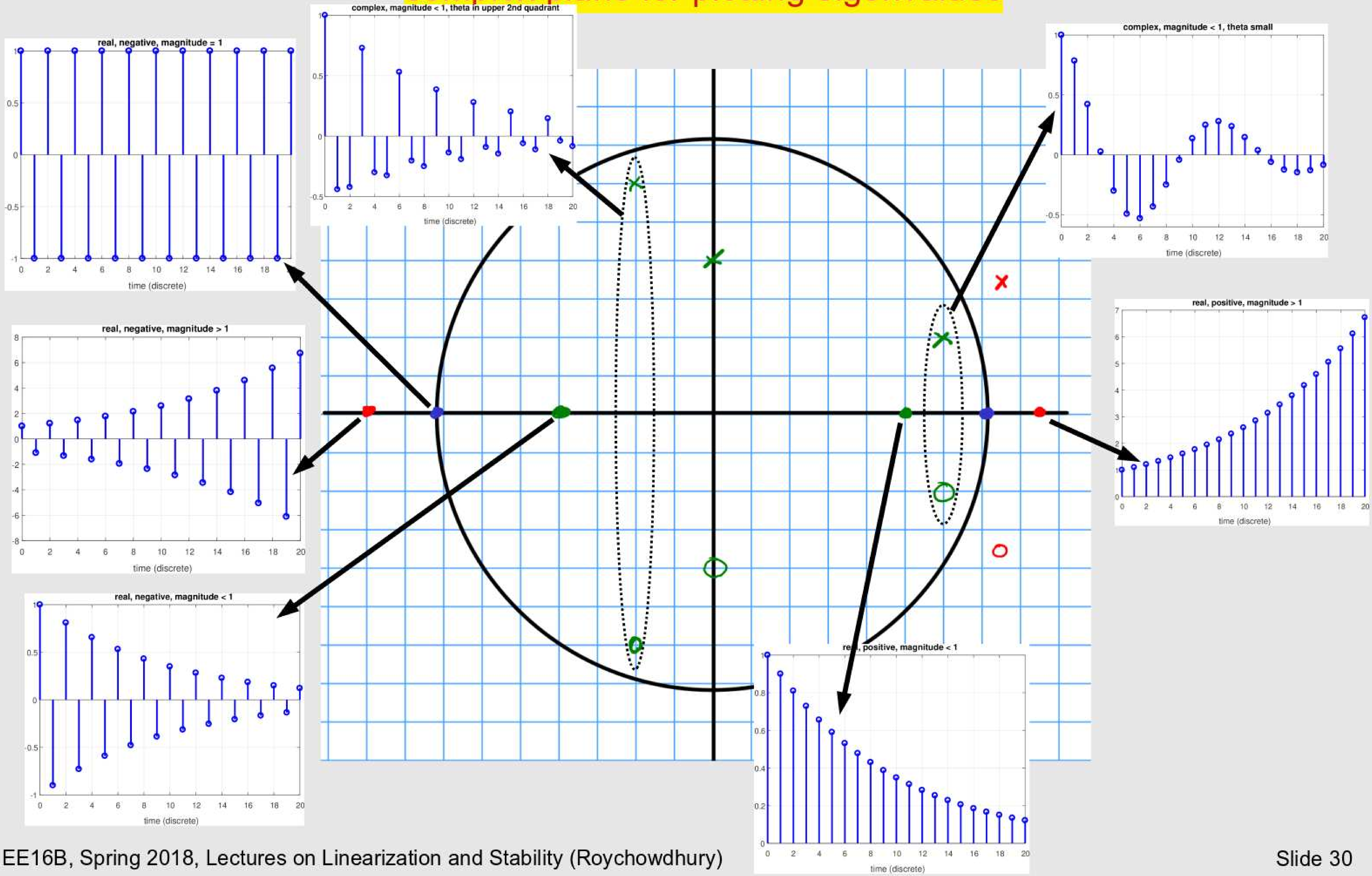
Eigenvalues and IC Responses (discrete)

complex plane for plotting eigenvalues

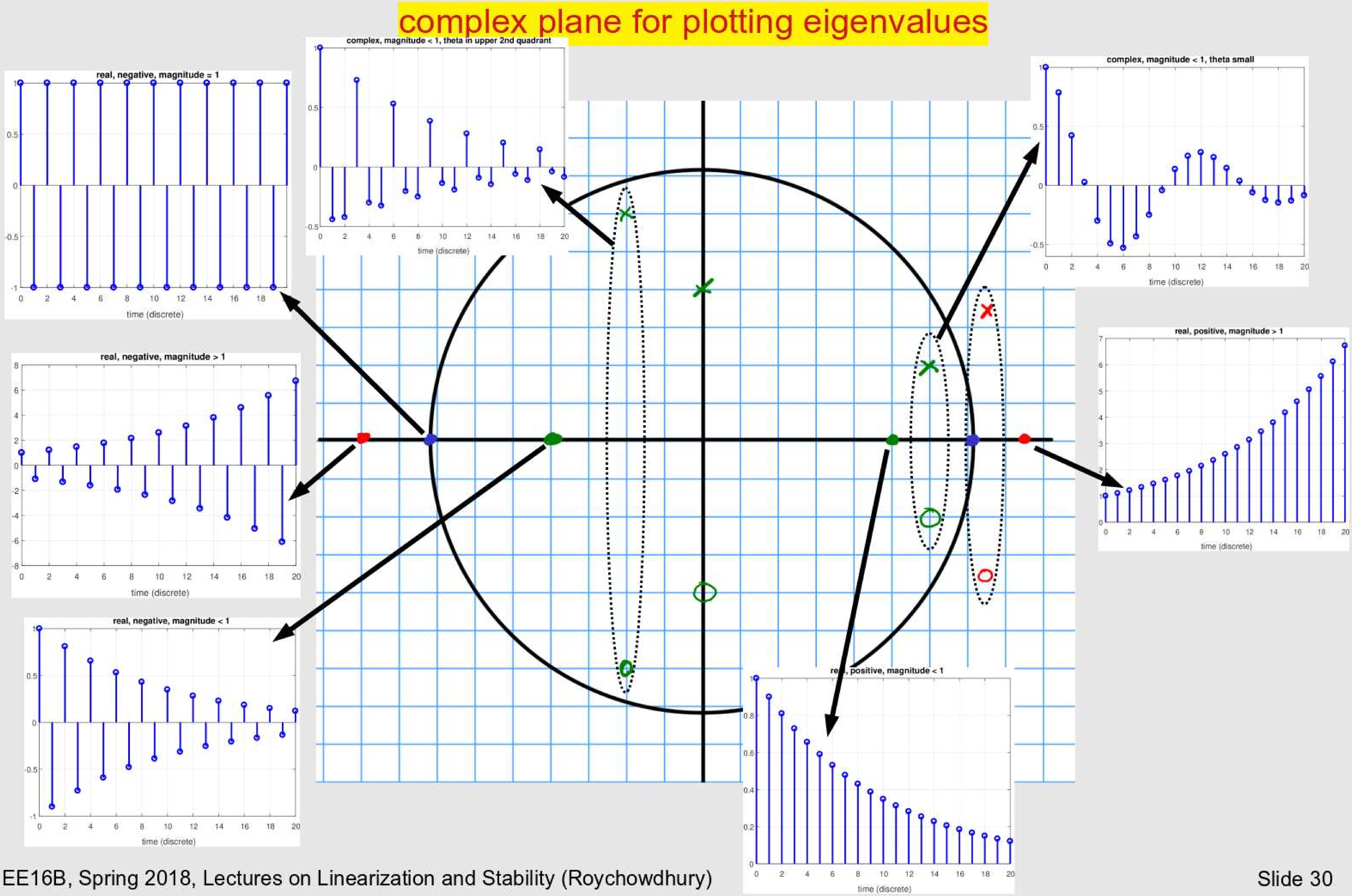


Eigenvalues and IC Responses (discrete)

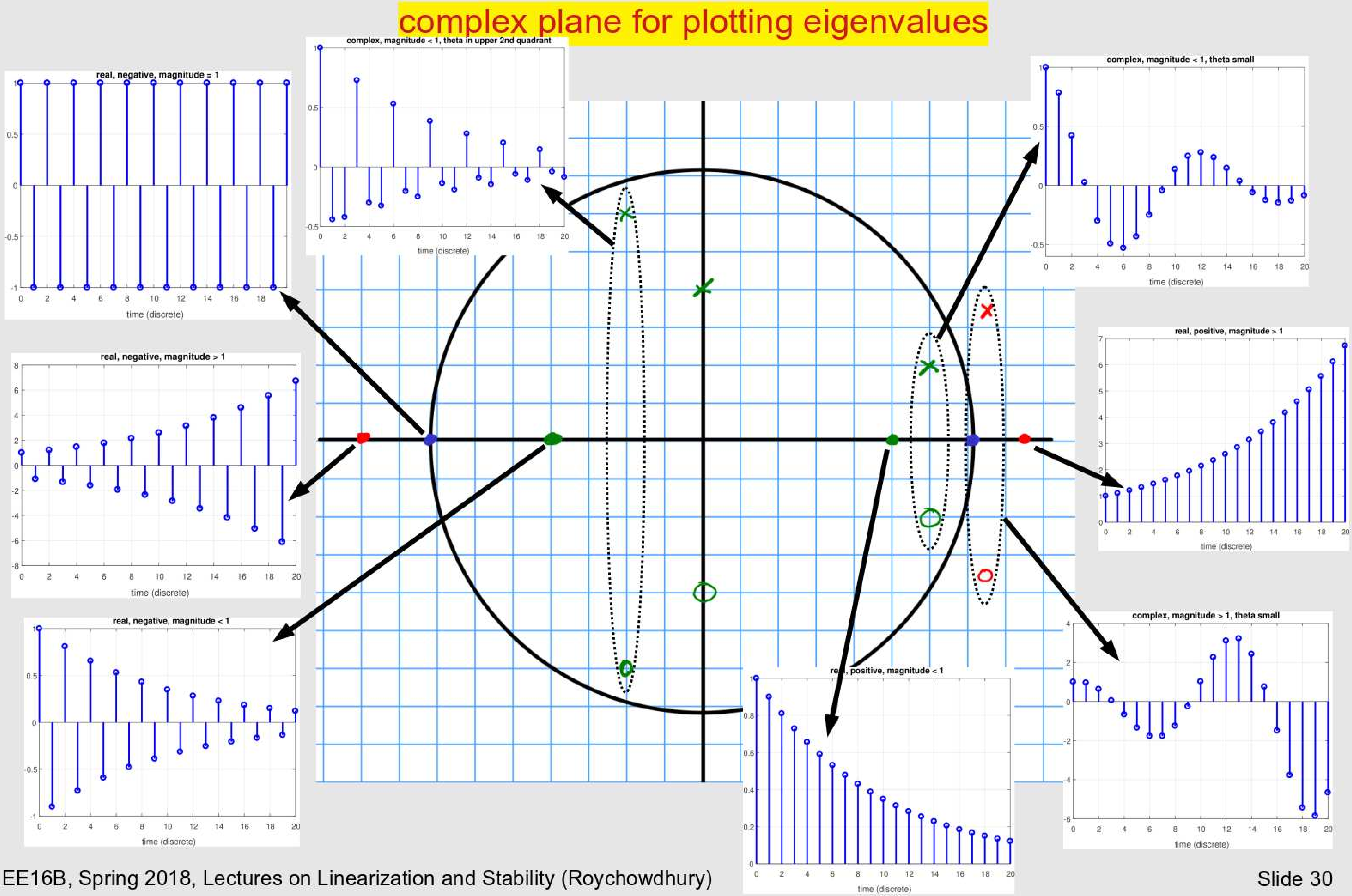
complex plane for plotting eigenvalues



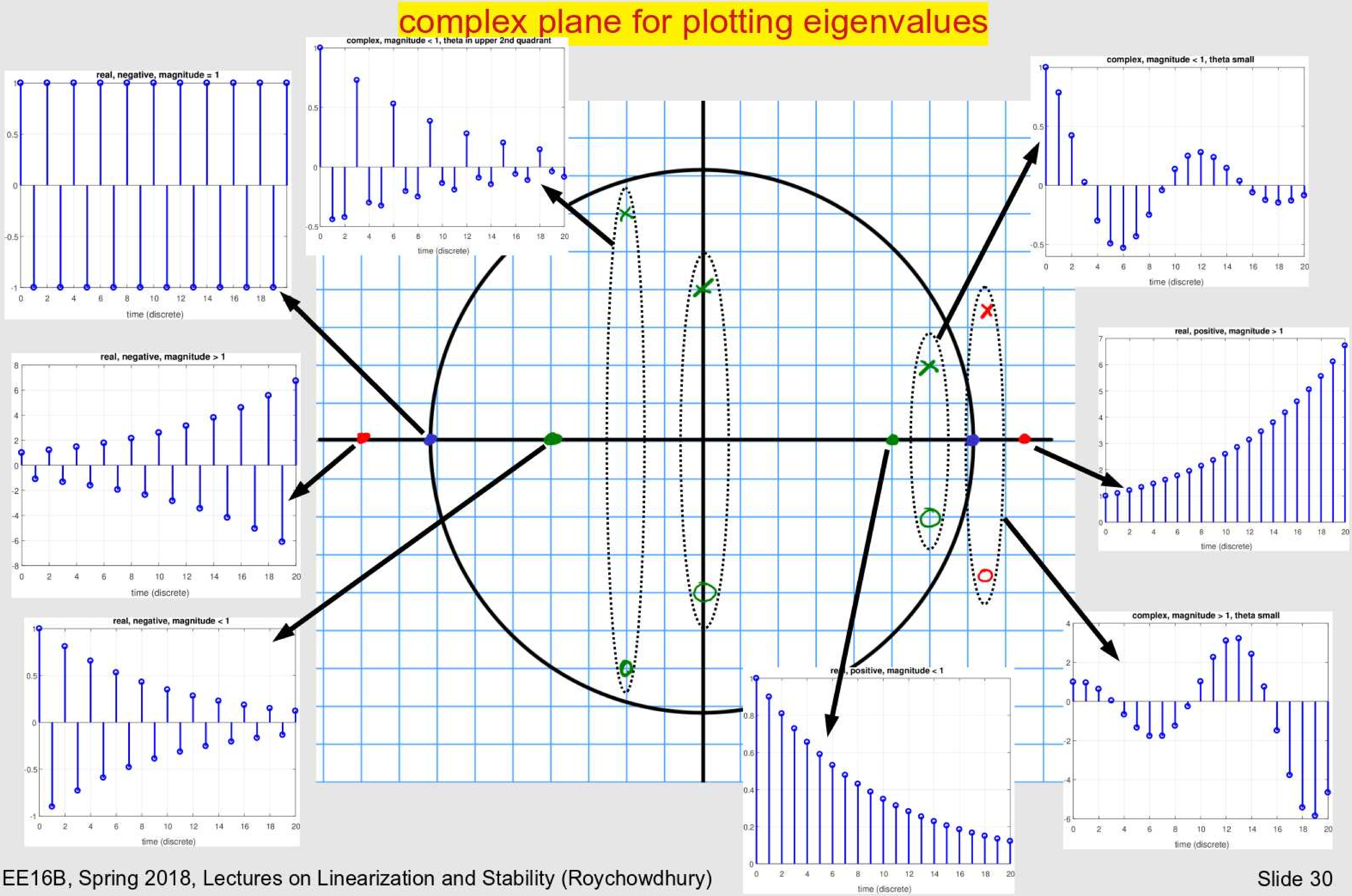
Eigenvalues and IC Responses (discrete)



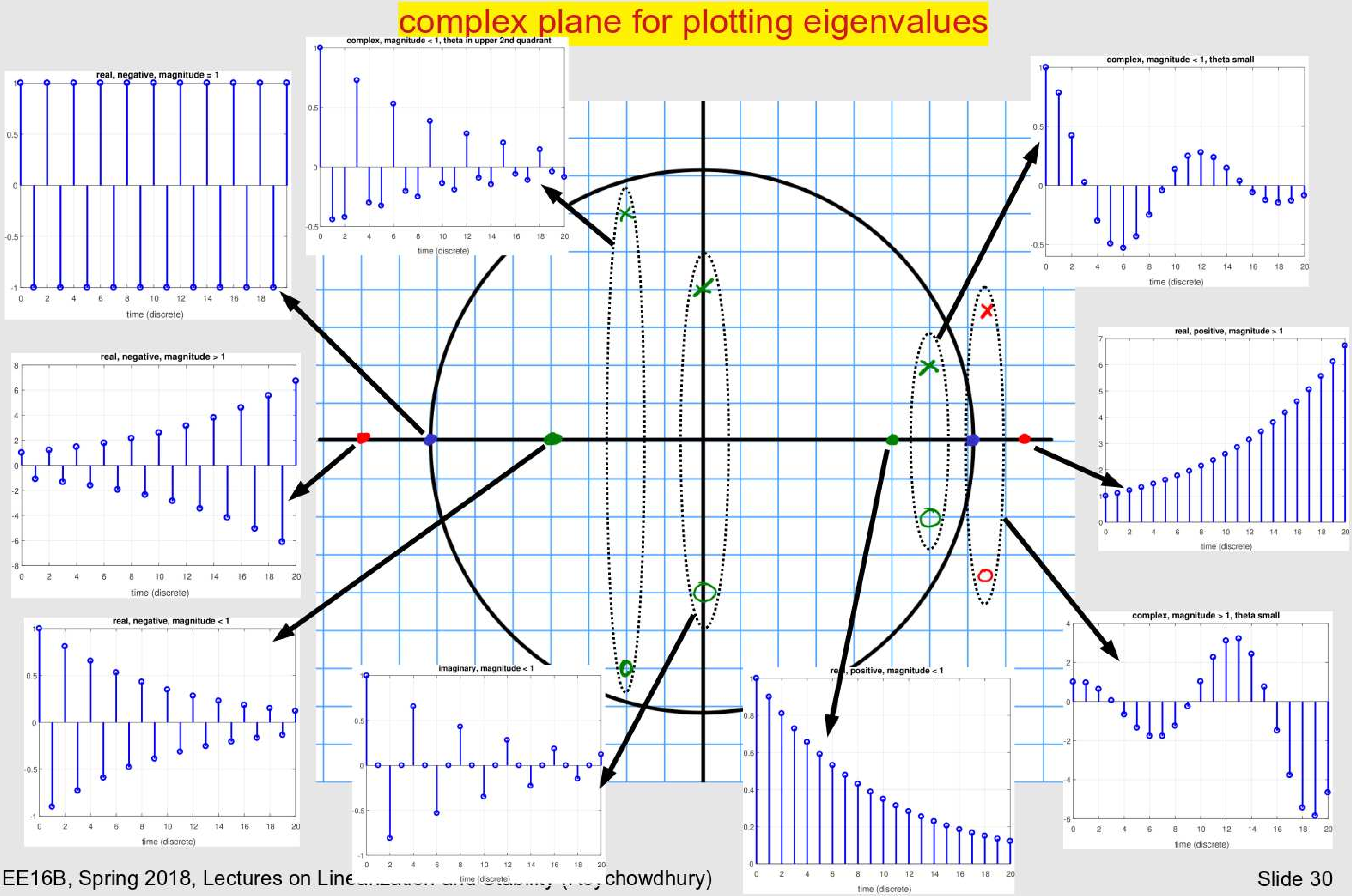
Eigenvalues and IC Responses (discrete)



Eigenvalues and IC Responses (discrete)



Eigenvalues and IC Responses (discrete)



Summary

- Linearization
  - scalar and vector cases
    - ➔ example: pendulum, (pole-cart)
- Stability
  - scalar and vector cases
    - ➔ continuous: real parts of eigenvalues determine stability
      - pendulum: stable and unstable equilibria
        - eigenvalue vs friction plots (root-locus plots)
    - ➔ discrete: magnitudes of eigenvalues determine stability