EE16B, Spring 2018
UC Berkeley EECS

Maharbiz and Roychowdhury

Lectures 6B & 7A: Overview Slides

Controller Canonical Form
Observability

Controller Canonical Form (CCF)

- Recall prior example: $\vec{x}[t+1] = \begin{bmatrix} 0 & 1 \\ a_1 & a_2 \end{bmatrix} \vec{x}[t] + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u[t]$

Controller Canonical Form (CCF)

- Recall prior example: $\vec{x}[t+1] = \begin{bmatrix} 0 & 1 \\ a_1 & a_2 \end{bmatrix} \vec{x}[t] + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u[t]$
  - char. poly.: $\lambda^2 - a_2\lambda - a_1$: nice simple formula

Controller Canonical Form (CCF)

- Recall prior example: $\vec{x}[t+1] = \begin{bmatrix} 0 & 1 \\ a_1 & a_2 \end{bmatrix} \vec{x}[t] + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u[t]$
  - char. poly.: $\lambda^2 - a_2\lambda - a_1$: nice simple formula
- Generalization: **Controller Canonical Form (CCF)**

- $A = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & & \ddots & \ddots & & \vdots \\ \vdots & & & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & \cdots & 0 & 1 \\ a_1 & a_2 & a_3 & \cdots & a_{n-1} & a_n \end{bmatrix}, \quad \vec{b} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ \vdots \\ 0 \\ 1 \end{bmatrix}$

Controller Canonical Form (CCF)

- Recall prior example: $\vec{x}[t+1] = \begin{bmatrix} 0 & 1 \\ a_1 & a_2 \end{bmatrix} \vec{x}[t] + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u[t]$
  - char. poly.: $\lambda^2 - a_2\lambda - a_1$: nice simple formula
- Generalization: **Controller Canonical Form (CCF)**

- $A = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & & \ddots & \ddots & & \vdots \\ \vdots & & & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & \cdots & 0 & 1 \\ a_1 & a_2 & a_3 & \cdots & a_{n-1} & a_n \end{bmatrix}, \quad \vec{b} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ \vdots \\ 0 \\ 1 \end{bmatrix}$

- char poly: $\lambda^n - a_n\lambda^{n-1} - a_{n-1}\lambda^{n-2} - \cdots - a_2\lambda - a_1$
  - ➔ not difficult to show this (though a bit tedious)
    - apply determinant formula using minors to the last row

Feedback on CCF

- System: $\frac{d}{dt}\vec{x}(t) = A\vec{x}(t) + \vec{b}u(t)$, with $(A, \vec{b})$ in CCF

Feedback on CCF

- System: $\frac{d}{dt}\vec{x}(t) = A\vec{x}(t) + \vec{b}u(t)$, with $(A, \vec{b})$ in CCF
  - apply feedback $\vec{k}$: $A \mapsto A - \vec{b}\vec{k}^T$

➔ $$A - \vec{b}\vec{k}^T = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & & \ddots & \ddots & & \vdots \\ \vdots & & & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & \cdots & 0 & 1 \\ a_1 - k_1 & a_2 - k_2 & a_3 - k_3 & \cdots & a_{n-1} - k_{n-1} & a_n - k_n \end{bmatrix}, \quad \vec{b} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ \vdots \\ 0 \\ 1 \end{bmatrix}$$

Feedback on CCF

- System: $\frac{d}{dt}\vec{x}(t) = A\vec{x}(t) + \vec{b}u(t)$, with $(A, \vec{b})$ in CCF
  - apply feedback $\vec{k}$: $A \mapsto A - \vec{b}\vec{k}^T$

$$\Rightarrow A - \vec{b}\vec{k}^T = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & & \ddots & \ddots & & \vdots \\ \vdots & & & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & \cdots & 0 & 1 \\ a_1 - k_1 & a_2 - k_2 & a_3 - k_3 & \cdots & a_{n-1} - k_{n-1} & a_n - k_n \end{bmatrix}, \quad \vec{b} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ \vdots \\ 0 \\ 1 \end{bmatrix}$$

- char poly: $\lambda^n - (a_n - k_n)\lambda^{n-1} - (a_{n-1} - k_{n-1})\lambda^{n-2} - \cdots - (a_2 - k_2)\lambda - (a_1 - k_1)$

Feedback on CCF

- System: $\frac{d}{dt}\vec{x}(t) = A\vec{x}(t) + \vec{b}u(t)$, with $(A, \vec{b})$ in CCF
  - apply feedback $\vec{k}$: $A \mapsto A - \vec{b}\vec{k}^T$

➔ $$A - \vec{b}\vec{k}^T = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & & \ddots & \ddots & & \vdots \\ \vdots & & & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & \cdots & 0 & 1 \\ a_1 - k_1 & a_2 - k_2 & a_3 - k_3 & \cdots & a_{n-1} - k_{n-1} & a_n - k_n \end{bmatrix}, \quad \vec{b} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ \vdots \\ 0 \\ 1 \end{bmatrix}$$

- char poly: $$\lambda^n - (a_n - k_n)\lambda^{n-1} - (a_{n-1} - k_{n-1})\lambda^{n-2} - \cdots - (a_2 - k_2)\lambda - (a_1 - k_1)$$

  ➔ its roots are the eigenvalues that determine stability

Assigning Desired Roots

- Suppose you want λ_1, λ_2, …, λ_n to be the roots

Assigning Desired Roots

- Suppose you want λ_1, λ_2, …, λ_n to be the roots
  - the char. poly. should equal: $(\lambda - \lambda_1)(\lambda - \lambda_2) \cdots (\lambda - \lambda_n)$
    - ➔ (why?)

Assigning Desired Roots

- Suppose you want $\lambda_1, \lambda_2, \ldots, \lambda_n$ to be the roots
  - the char. poly. should equal: $(\lambda - \lambda_1)(\lambda - \lambda_2)\cdots(\lambda - \lambda_n)$
    - (why?)
- Expand out $(\lambda - \lambda_1)(\lambda - \lambda_2)\cdots(\lambda - \lambda_n) \equiv \prod_{i=1}^{n}(\lambda - \lambda_i)$

Assigning Desired Roots

- Suppose you want $\lambda_1, \lambda_2, \ldots, \lambda_n$ to be the roots
  - the char. poly. should equal: $(\lambda - \lambda_1)(\lambda - \lambda_2)\cdots(\lambda - \lambda_n)$
    - ➔ (why?)
- Expand out $(\lambda - \lambda_1)(\lambda - \lambda_2)\cdots(\lambda - \lambda_n) \equiv \prod_{i=1}^{n}(\lambda - \lambda_i)$
  - $$\begin{aligned}\prod_{i=1}^{n}(\lambda - \lambda_i) = & \lambda^n - (\lambda_1 + \lambda_2 + \cdots + \lambda_n)\lambda^{n-1} \\ & + \big[\lambda_1(\lambda_2 + \lambda_3 + \cdots + \lambda_n) + \lambda_2(\lambda_3 + \lambda_4 + \cdots + \lambda_n) \\ & \quad + \cdots + \lambda_{n-1}\lambda_n\big]\lambda^{n-2} \\ & + \cdots + (-1)^n \lambda_1 \lambda_2 \cdots \lambda_n\end{aligned}$$

Assigning Desired Roots

- Suppose you want $\lambda_1, \lambda_2, \ldots, \lambda_n$ to be the roots
  - the char. poly. should equal: $(\lambda - \lambda_1)(\lambda - \lambda_2)\cdots(\lambda - \lambda_n)$
    - ➔ (why?)
- Expand out $(\lambda - \lambda_1)(\lambda - \lambda_2)\cdots(\lambda - \lambda_n) \equiv \prod_{i=1}^{n}(\lambda - \lambda_i)$
  - $$\prod_{i=1}^{n}(\lambda - \lambda_i) = \lambda^n \overbrace{- (\lambda_1 + \lambda_2 + \cdots + \lambda_n)}^{\gamma_n}\lambda^{n-1} + \big[\lambda_1(\lambda_2+\lambda_3+\cdots+\lambda_n) + \lambda_2(\lambda_3+\lambda_4+\cdots+\lambda_n) + \cdots + \lambda_{n-1}\lambda_n\big]\lambda^{n-2} + \cdots + (-1)^n\lambda_1\lambda_2\cdots\lambda_n$$

Assigning Desired Roots

- Suppose you want λ_1, λ_2, …, λ_n to be the roots
  - the char. poly. should equal: $(\lambda-\lambda_1)(\lambda-\lambda_2)\cdots(\lambda-\lambda_n)$
    - (why?)
- Expand out $(\lambda-\lambda_1)(\lambda-\lambda_2)\cdots(\lambda-\lambda_n) \equiv \prod_{i=1}^{n}(\lambda-\lambda_i)$
  - $$\prod_{i=1}^{n}(\lambda-\lambda_i) = \lambda^n \underbrace{-(\lambda_1+\lambda_2+\cdots+\lambda_n)}_{\gamma_n}\lambda^{n-1} + \underbrace{\left[\lambda_1(\lambda_2+\lambda_3+\cdots+\lambda_n)+\lambda_2(\lambda_3+\lambda_4+\cdots+\lambda_n) + \cdots + \lambda_{n-1}\lambda_n\right]}_{\gamma_{n-1}}\lambda^{n-2} + \cdots + (-1)^n\lambda_1\lambda_2\cdots\lambda_n$$

Assigning Desired Roots

- Suppose you want λ_1, λ_2, …, λ_n to be the roots
  - the char. poly. should equal: $(\lambda - \lambda_1)(\lambda - \lambda_2)\cdots(\lambda - \lambda_n)$
    - (why?)
- Expand out $(\lambda - \lambda_1)(\lambda - \lambda_2)\cdots(\lambda - \lambda_n) \equiv \prod_{i=1}^{n}(\lambda - \lambda_i)$
  - $$\prod_{i=1}^{n}(\lambda - \lambda_i) = \lambda^n \underbrace{- (\lambda_1 + \lambda_2 + \cdots + \lambda_n)}_{\gamma_n}\lambda^{n-1} + \underbrace{[\lambda_1(\lambda_2 + \lambda_3 + \cdots + \lambda_n) + \lambda_2(\lambda_3 + \lambda_4 + \cdots + \lambda_n) + \cdots + \lambda_{n-1}\lambda_n]}_{\gamma_{n-1}}\lambda^{n-2} + \cdots + \underbrace{(-1)^n \lambda_1 \lambda_2 \cdots \lambda_n}_{\gamma_1}$$

Assigning Desired Roots

- Suppose you want λ_1, λ_2, …, λ_n to be the roots
  - the char. poly. should equal: $(\lambda-\lambda_1)(\lambda-\lambda_2)\cdots(\lambda-\lambda_n)$
    - ➔ (why?)
- Expand out $(\lambda-\lambda_1)(\lambda-\lambda_2)\cdots(\lambda-\lambda_n) \equiv \prod_{i=1}^{n}(\lambda-\lambda_i)$
  - $$\prod_{i=1}^{n}(\lambda-\lambda_i) = \lambda^n \underbrace{-(\lambda_1+\lambda_2+\cdots+\lambda_n)}_{\gamma_n}\lambda^{n-1} + \underbrace{[\lambda_1(\lambda_2+\lambda_3+\cdots+\lambda_n)+\lambda_2(\lambda_3+\lambda_4+\cdots+\lambda_n) + \cdots + \lambda_{n-1}\lambda_n]}_{\gamma_{n-1}}\lambda^{n-2} + \cdots + \underbrace{(-1)^n\lambda_1\lambda_2\cdots\lambda_n}_{\gamma_1}$$
  - equate coefficients against $\lambda^n - (a_n-k_n)\lambda^{n-1} - (a_{n-1}-k_{n-1})\lambda^{n-2} - \cdots - (a_2-k_2)\lambda - (a_1-k_1)$

Assigning Desired Roots

- Suppose you want $\lambda_1, \lambda_2, \ldots, \lambda_n$ to be the roots
  - the char. poly. should equal: $(\lambda-\lambda_1)(\lambda-\lambda_2)\cdots(\lambda-\lambda_n)$
    - ➔ (why?)
- Expand out $(\lambda-\lambda_1)(\lambda-\lambda_2)\cdots(\lambda-\lambda_n) \equiv \prod_{i=1}^{n}(\lambda-\lambda_i)$
  - $$\prod_{i=1}^{n}(\lambda-\lambda_i) = \lambda^n \underbrace{-(\lambda_1+\lambda_2+\cdots+\lambda_n)}_{\gamma_n}\lambda^{n-1} + \underbrace{\left[\lambda_1(\lambda_2+\lambda_3+\cdots+\lambda_n)+\lambda_2(\lambda_3+\lambda_4+\cdots+\lambda_n) + \cdots + \lambda_{n-1}\lambda_n\right]}_{\gamma_{n-1}}\lambda^{n-2} + \cdots + \underbrace{(-1)^n\lambda_1\lambda_2\cdots\lambda_n}_{\gamma_1}$$
  - equate coefficients against $\lambda^n - (a_n - k_n)\lambda^{n-1} - (a_{n-1}-k_{n-1})\lambda^{n-2} - \cdots - (a_2-k_2)\lambda - (a_1-k_1)$
    - ➔ $$\begin{aligned} a_n - k_n &= -\gamma_n \\ a_{n-1} - k_{n-1} &= -\gamma_{n-1} \\ &\vdots \\ a_1 - k_1 &= -\gamma_1 \end{aligned}$$

Assigning Desired Roots

- Suppose you want $\lambda_1, \lambda_2, \ldots, \lambda_n$ to be the roots
  - the char. poly. should equal: $(\lambda-\lambda_1)(\lambda-\lambda_2)\cdots(\lambda-\lambda_n)$
    - ➔ (why?)
- Expand out $(\lambda-\lambda_1)(\lambda-\lambda_2)\cdots(\lambda-\lambda_n) \equiv \prod_{i=1}^{n}(\lambda-\lambda_i)$
  - $$\prod_{i=1}^{n}(\lambda-\lambda_i) = \lambda^n \underbrace{-(\lambda_1+\lambda_2+\cdots+\lambda_n)}_{\gamma_n}\lambda^{n-1} + \underbrace{\left[\lambda_1(\lambda_2+\lambda_3+\cdots+\lambda_n)+\lambda_2(\lambda_3+\lambda_4+\cdots+\lambda_n)+\cdots+\lambda_{n-1}\lambda_n\right]}_{\gamma_{n-1}}\lambda^{n-2} + \cdots + \underbrace{(-1)^n\lambda_1\lambda_2\cdots\lambda_n}_{\gamma_1}$$
  - <u>equate coefficients</u> against $\lambda^n-(a_n-k_n)\lambda^{n-1}-(a_{n-1}-k_{n-1})\lambda^{n-2}-\cdots-(a_2-k_2)\lambda-(a_1-k_1)$
    - ➔ $$\begin{aligned} a_n-k_n &= -\gamma_n \\ a_{n-1}-k_{n-1} &= -\gamma_{n-1} \\ &\vdots \\ a_1-k_1 &= -\gamma_1 \end{aligned} \Rightarrow \begin{cases} k_n = \gamma_n - a_n \\ k_{n-1} = \gamma_{n-1} - a_{n-1} \\ \vdots \\ k_1 = \gamma_1 - a_1 \end{cases}$$

Assigning Desired Roots

- Suppose you want $\lambda_1, \lambda_2, \ldots, \lambda_n$ to be the roots
  - the char. poly. should equal: $(\lambda - \lambda_1)(\lambda - \lambda_2)\cdots(\lambda - \lambda_n)$
    - ➔ (why?)
- Expand out $(\lambda - \lambda_1)(\lambda - \lambda_2)\cdots(\lambda - \lambda_n) \equiv \prod_{i=1}^{n}(\lambda - \lambda_i)$
  - $$\prod_{i=1}^{n}(\lambda - \lambda_i) = \lambda^n \underbrace{-(\lambda_1 + \lambda_2 + \cdots + \lambda_n)}_{\gamma_n}\lambda^{n-1} + \underbrace{\left[\lambda_1(\lambda_2+\lambda_3+\cdots+\lambda_n) + \lambda_2(\lambda_3+\lambda_4+\cdots+\lambda_n) + \cdots + \lambda_{n-1}\lambda_n\right]}_{\gamma_{n-1}}\lambda^{n-2} + \cdots + \underbrace{(-1)^n\lambda_1\lambda_2\cdots\lambda_n}_{\gamma_1}$$
  - equate coefficients against $\lambda^n - (a_n - k_n)\lambda^{n-1} - (a_{n-1} - k_{n-1})\lambda^{n-2} - \cdots - (a_2 - k_2)\lambda - (a_1 - k_1)$
    - ➔ $$\begin{aligned} a_n - k_n &= -\gamma_n \\ a_{n-1} - k_{n-1} &= -\gamma_{n-1} \\ &\vdots \\ a_1 - k_1 &= -\gamma_1 \end{aligned} \Rightarrow \begin{cases} k_n = \gamma_n - a_n \\ k_{n-1} = \gamma_{n-1} - a_{n-1} \\ \quad\vdots \\ k_1 = \gamma_1 - a_1 \end{cases}$$

these feedback coeffs will place the eigenvalues at the desired locations

Assigning Desired Roots

- Suppose you want $\lambda_1, \lambda_2, \ldots, \lambda_n$ to be the roots
  - the char. poly. should equal: $(\lambda-\lambda_1)(\lambda-\lambda_2)\cdots(\lambda-\lambda_n)$
    - ➔ (why?)
- Expand out $(\lambda-\lambda_1)(\lambda-\lambda_2)\cdots(\lambda-\lambda_n) \equiv \prod_{i=1}^{n}(\lambda-\lambda_i)$
  - $$\prod_{i=1}^{n}(\lambda-\lambda_i) = \lambda^n \underbrace{-(\lambda_1+\lambda_2+\cdots+\lambda_n)}_{\gamma_n}\lambda^{n-1} + \underbrace{\left[\lambda_1(\lambda_2+\lambda_3+\cdots+\lambda_n)+\lambda_2(\lambda_3+\lambda_4+\cdots+\lambda_n)+\cdots+\lambda_{n-1}\lambda_n\right]}_{\gamma_{n-1}}\lambda^{n-2} + \cdots + \underbrace{(-1)^n\lambda_1\lambda_2\cdots\lambda_n}_{\gamma_1}$$
  - equate coefficients against $\lambda^n-(a_n-k_n)\lambda^{n-1}-(a_{n-1}-k_{n-1})\lambda^{n-2}-\cdots-(a_2-k_2)\lambda-(a_1-k_1)$
    - ➔ $$\begin{aligned} a_n-k_n &= -\gamma_n \\ a_{n-1}-k_{n-1} &= -\gamma_{n-1} \\ &\vdots \\ a_1-k_1 &= -\gamma_1 \end{aligned} \Rightarrow \begin{cases} k_n = \gamma_n - a_n \\ k_{n-1} = \gamma_{n-1}-a_{n-1} \\ \vdots \\ k_1 = \gamma_1 - a_1 \end{cases}$$

these feedback coeffs will place the eigenvalues at the desired locations

We just showed: if a system is in CCF, feedback can move its eigenvalues to any desired locations

CCF and Eigenvalue Placement: Examples

- $A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 2 & 3 \end{bmatrix}, \vec{b} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$

CCF and Eigenvalue Placement: Examples

- $A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 2 & 3 \end{bmatrix}, \vec{b} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \Rightarrow A - \vec{b}\vec{k}^T = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 - k_1 & 2 - k_2 & 3 - k_3 \end{bmatrix}$

CCF and Eigenvalue Placement: Examples

- $A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 2 & 3 \end{bmatrix}, \vec{b} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \Rightarrow A - \vec{b}\vec{k}^T = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 - k_1 & 2 - k_2 & 3 - k_3 \end{bmatrix}$
  - char. poly.: $\lambda^3 - (3-k_3)\lambda^2 - (2-k_2)\lambda - (1 - k_1)$

CCF and Eigenvalue Placement: Examples

- $A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 2 & 3 \end{bmatrix}, \vec{b} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \Rightarrow A - \vec{b}\vec{k}^T = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 - k_1 & 2 - k_2 & 3 - k_3 \end{bmatrix}$
  - char. poly.: $\lambda^3 - (3-k_3)\lambda^2 - (2-k_2)\lambda - (1 - k_1)$
  - desired char. poly.: $(\lambda - \lambda_1)(\lambda - \lambda_2)(\lambda - \lambda_3)$

CCF and Eigenvalue Placement: Examples

- $A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 2 & 3 \end{bmatrix}, \vec{b} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \Rightarrow A - \vec{b}\vec{k}^T = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1-k_1 & 2-k_2 & 3-k_3 \end{bmatrix}$
  - char. poly.: $\lambda^3 - (3-k_3)\lambda^2 - (2-k_2)\lambda - (1-k_1)$
  - desired char. poly.: $(\lambda - \lambda_1)(\lambda - \lambda_2)(\lambda - \lambda_3)$
    - say we want: $\lambda_1 = \lambda_2 = \lambda_3 = 0$

CCF and Eigenvalue Placement: Examples

- $A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 2 & 3 \end{bmatrix}, \vec{b} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \Rightarrow A - \vec{b}\vec{k}^T = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 - k_1 & 2 - k_2 & 3 - k_3 \end{bmatrix}$
  - char. poly.: $\lambda^3 - (3-k_3)\lambda^2 - (2-k_2)\lambda - (1-k_1)$
  - desired char. poly.: $(\lambda - \lambda_1)(\lambda - \lambda_2)(\lambda - \lambda_3)$
    - say we want: $\lambda_1 = \lambda_2 = \lambda_3 = 0 \Rightarrow (\lambda - \lambda_1)(\lambda - \lambda_2)(\lambda - \lambda_3) \equiv \lambda^3$

CCF and Eigenvalue Placement: Examples

- $A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 2 & 3 \end{bmatrix}, \vec{b} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \Rightarrow A - \vec{b}\vec{k}^T = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 - k_1 & 2 - k_2 & 3 - k_3 \end{bmatrix}$
  - char. poly.: $\lambda^3 - (3-k_3)\lambda^2 - (2-k_2)\lambda - (1 - k_1)$
  - desired char. poly.: $(\lambda - \lambda_1)(\lambda - \lambda_2)(\lambda - \lambda_3)$
    - say we want: $\lambda_1 = \lambda_2 = \lambda_3 = 0 \Rightarrow (\lambda - \lambda_1)(\lambda - \lambda_2)(\lambda - \lambda_3) \equiv \lambda^3$
      - then $k_3 = 3, k_2 = 2, k_1 = 1$

CCF and Eigenvalue Placement: Examples

- $A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 2 & 3 \end{bmatrix}, \vec{b} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \Rightarrow A - \vec{b}\vec{k}^T = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1-k_1 & 2-k_2 & 3-k_3 \end{bmatrix}$
  - char. poly.: $\lambda^3 - (3-k_3)\lambda^2 - (2-k_2)\lambda - (1-k_1)$
  - desired char. poly.: $(\lambda-\lambda_1)(\lambda-\lambda_2)(\lambda-\lambda_3)$
    - say we want: $\lambda_1 = \lambda_2 = \lambda_3 = 0 \Rightarrow (\lambda-\lambda_1)(\lambda-\lambda_2)(\lambda-\lambda_3) \equiv \lambda^3$
      - then $k_3 = 3, k_2 = 2, k_1 = 1$
    - or, if we want: $\lambda_1 = -1, \lambda_2 = -2, \lambda_3 = -3$

CCF and Eigenvalue Placement: Examples

- $A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 2 & 3 \end{bmatrix}, \vec{b} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \Rightarrow A - \vec{b}\vec{k}^T = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1-k_1 & 2-k_2 & 3-k_3 \end{bmatrix}$
  - char. poly.: $\lambda^3 - (3-k_3)\lambda^2 - (2-k_2)\lambda - (1-k_1)$
  - desired char. poly.: $(\lambda-\lambda_1)(\lambda-\lambda_2)(\lambda-\lambda_3)$
    - say we want: $\lambda_1 = \lambda_2 = \lambda_3 = 0 \;\Rightarrow (\lambda-\lambda_1)(\lambda-\lambda_2)(\lambda-\lambda_3) \equiv \lambda^3$
      - then $k_3 = 3, k_2 = 2, k_1 = 1$
    - or, if we want: $\lambda_1 = -1, \lambda_2 = -2, \lambda_3 = -3$
      - $(\lambda-\lambda_1)(\lambda-\lambda_2)(\lambda-\lambda_3) \equiv \lambda^3 + 6\lambda^2 + 11\lambda + 6$

CCF and Eigenvalue Placement: Examples

- $A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 2 & 3 \end{bmatrix}, \vec{b} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \Rightarrow A - \vec{b}\vec{k}^T = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1-k_1 & 2-k_2 & 3-k_3 \end{bmatrix}$
  - char. poly.: $\lambda^3 - (3-k_3)\lambda^2 - (2-k_2)\lambda - (1-k_1)$
  - desired char. poly.: $(\lambda-\lambda_1)(\lambda-\lambda_2)(\lambda-\lambda_3)$
    - say we want: $\lambda_1 = \lambda_2 = \lambda_3 = 0 \;\Rightarrow (\lambda-\lambda_1)(\lambda-\lambda_2)(\lambda-\lambda_3) \equiv \lambda^3$
      - then $k_3 = 3, k_2 = 2, k_1 = 1$
    - or, if we want: $\lambda_1 = -1, \lambda_2 = -2, \lambda_3 = -3$
      - $(\lambda-\lambda_1)(\lambda-\lambda_2)(\lambda-\lambda_3) \equiv \lambda^3 + 6\lambda^2 + 11\lambda + 6$
      - $\begin{array}{l} -(3-k_3) = 6 \\ -(2-k_2) = 11 \\ -(1-k_1) = 6 \end{array}$

CCF and Eigenvalue Placement: Examples

- $A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 2 & 3 \end{bmatrix}, \vec{b} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \Rightarrow A - \vec{b}\vec{k}^T = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1-k_1 & 2-k_2 & 3-k_3 \end{bmatrix}$
  - char. poly.: $\lambda^3 - (3-k_3)\lambda^2 - (2-k_2)\lambda - (1-k_1)$
  - desired char. poly.: $(\lambda-\lambda_1)(\lambda-\lambda_2)(\lambda-\lambda_3)$
    - say we want: $\lambda_1 = \lambda_2 = \lambda_3 = 0 \Rightarrow (\lambda-\lambda_1)(\lambda-\lambda_2)(\lambda-\lambda_3) \equiv \lambda^3$
      - then $k_3 = 3, k_2 = 2, k_1 = 1$
    - or, if we want: $\lambda_1 = -1, \lambda_2 = -2, \lambda_3 = -3$
      - $(\lambda-\lambda_1)(\lambda-\lambda_2)(\lambda-\lambda_3) \equiv \lambda^3 + 6\lambda^2 + 11\lambda + 6$
      - $\begin{matrix} -(3-k_3) = 6 \\ -(2-k_2) = 11 \\ -(1-k_1) = 6 \end{matrix} \Rightarrow \begin{cases} k_3 = 9 \\ k_2 = 13 \\ k_1 = 7 \end{cases}$

Converting Systems to CCF

- But CCF seems a very special/restrictive form …
  - … key question: what systems are in CCF?

Converting Systems to CCF

- But CCF seems a very special/restrictive form …
  - … key question: what systems are in CCF?
- A: **any controllable system** can be converted to CCF!

Converting Systems to CCF

- But CCF seems a very special/restrictive form …
  - … key question: what systems are in CCF?
- A: **any controllable system** can be converted to CCF!
- Here's how you do it:

1. Given any state-space system: $\frac{d}{dt}\vec{x}(t) = A\vec{x}(t) + \vec{b}u(t)$

Converting Systems to CCF

- But CCF seems a very special/restrictive form …
  - … key question: what systems are in CCF?
- A: **any controllable system** can be converted to CCF!
- Here's how you do it:
  1. Given any state-space system: $\frac{d}{dt}\vec{x}(t) = A\vec{x}(t) + \vec{b}u(t)$
  2. Form its controllability matrix: $R_n \triangleq \left[\vec{b}, A\vec{b}, A^2\vec{b}, \cdots, A^{n-1}\vec{b}\right]$

Converting Systems to CCF

- But CCF seems a very special/restrictive form …
  - … key question: what systems are in CCF?
- A: **any controllable system** can be converted to CCF!
- Here's how you do it:
  1. Given any state-space system: $\frac{d}{dt}\vec{x}(t) = A\vec{x}(t) + \vec{b}u(t)$
  2. Form its controllability matrix: $R_n \triangleq \left[\vec{b}, A\vec{b}, A^2\vec{b}, \cdots, A^{n-1}\vec{b}\right]$

full rank if system controllable;
and square, hence invertible

Converting Systems to CCF

- But CCF seems a very special/restrictive form …
  - … key question: what systems are in CCF?
- A: **any controllable system** can be converted to CCF!
- Here's how you do it:
  1. Given any state-space system: $\frac{d}{dt}\vec{x}(t) = A\vec{x}(t) + \vec{b}u(t)$
  2. Form its controllability matrix: $R_n \triangleq \left[\vec{b}, A\vec{b}, A^2\vec{b}, \cdots, A^{n-1}\vec{b}\right]$
  3. Compute its inverse: R_n^{-1}

full rank if system controllable;
and square, hence invertible

Converting Systems to CCF

- But CCF seems a very special/restrictive form …
  - … key question: what systems are in CCF?
- A: **any controllable system** can be converted to CCF!
- Here's how you do it:
  1. Given any state-space system: $\frac{d}{dt}\vec{x}(t) = A\vec{x}(t) + \vec{b}u(t)$
  2. Form its controllability matrix: $R_n \triangleq \left[\vec{b}, A\vec{b}, A^2\vec{b}, \cdots, A^{n-1}\vec{b}\right]$
  3. Compute its inverse: R_n^{-1}
  4. Grab the last row of R_n^{-1}: call it $\vec{q}^T$

full rank if system controllable;
and square, hence invertible

Converting Systems to CCF

- But CCF seems a very special/restrictive form …
  - … key question: what systems are in CCF?
- A: **any controllable system** can be converted to CCF!
- Here's how you do it:
  1. Given any state-space system: $\frac{d}{dt}\vec{x}(t) = A\vec{x}(t) + \vec{b}u(t)$
  2. Form its controllability matrix: $R_n \triangleq \left[\vec{b}, A\vec{b}, A^2\vec{b}, \cdots, A^{n-1}\vec{b}\right]$
  3. Compute its inverse: R_n^{-1}
  4. Grab the <u>last row</u> of R_n^{-1}: call it $\vec{q}^T$

full rank if system controllable; and square, hence invertible

- $R_n^{-1} = \begin{bmatrix} \rule{4cm}{0.4pt} \\ \rule{4cm}{0.4pt} \\ \vdots \\ \rule{4cm}{0.4pt} \\ \longleftarrow \quad \vec{q}^T \quad \longrightarrow \end{bmatrix}$; ($\vec{q}$ is a col. vector; $\vec{q}^T$ is a row vector)

Converting Systems to CCF (contd.)

5. Form the basis transformation matrix $T \triangleq \begin{bmatrix} \longleftarrow & \vec{q}^T & \longrightarrow \\ \longleftarrow & \vec{q}^T A & \longrightarrow \\ \longleftarrow & \vec{q}^T A^2 & \longrightarrow \\ & \vdots & \\ \longleftarrow & \vec{q}^T A^{n-1} & \longrightarrow \end{bmatrix}$

Converting Systems to CCF (contd.)

T will be full rank, hence non-singular and invertible

5. Form the basis transformation matrix $T \triangleq \begin{bmatrix} \longleftarrow & \vec{q}^T & \longrightarrow \\ \longleftarrow & \vec{q}^T A & \longrightarrow \\ \longleftarrow & \vec{q}^T A^2 & \longrightarrow \\ & \vdots & \\ \longleftarrow & \vec{q}^T A^{n-1} & \longrightarrow \end{bmatrix}$

Converting Systems to CCF (contd.)

T will be full rank, hence non-singular and invertible

5. Form the basis transformation matrix $T \triangleq \begin{bmatrix} \longleftarrow & \vec{q}^T & \longrightarrow \\ \longleftarrow & \vec{q}^T A & \longrightarrow \\ \longleftarrow & \vec{q}^T A^2 & \longrightarrow \\ & \vdots & \\ \longleftarrow & \vec{q}^T A^{n-1} & \longrightarrow \end{bmatrix}$

6. Define $\vec{z}(t) = T\vec{x}(t) \Leftrightarrow \vec{x}(t) = T^{-1}\vec{z}(t)$

Converting Systems to CCF (contd.)

T will be full rank, hence non-singular and invertible

5. Form the basis transformation matrix $T \triangleq \begin{bmatrix} \longleftarrow & \vec{q}^T & \longrightarrow \\ \longleftarrow & \vec{q}^T A & \longrightarrow \\ \longleftarrow & \vec{q}^T A^2 & \longrightarrow \\ & \vdots & \\ \longleftarrow & \vec{q}^T A^{n-1} & \longrightarrow \end{bmatrix}$

6. Define $\vec{z}(t) = T\vec{x}(t) \Leftrightarrow \vec{x}(t) = T^{-1}\vec{z}(t)$

7. Write the system in terms of $\vec{z}(t)$:

- $$\frac{d}{dt}\vec{z}(t) = \underbrace{TAT^{-1}}_{\hat{A}}\vec{z}(t) + \underbrace{T\vec{b}}_{\hat{\vec{b}}}u(t)$$
  $$\vec{x}(t) = T^{-1}\vec{z}(t)$$

Converting Systems to CCF (contd.)

T will be full rank, hence non-singular and invertible

5. Form the basis transformation matrix $T \triangleq \begin{bmatrix} \longleftarrow & \vec{q}^T & \longrightarrow \\ \longleftarrow & \vec{q}^T A & \longrightarrow \\ \longleftarrow & \vec{q}^T A^2 & \longrightarrow \\ & \vdots & \\ \longleftarrow & \vec{q}^T A^{n-1} & \longrightarrow \end{bmatrix}$
6. Define $\vec{z}(t) = T\vec{x}(t) \Leftrightarrow \vec{x}(t) = T^{-1}\vec{z}(t)$
7. Write the system in terms of $\vec{z}(t)$:

- $$\frac{d}{dt}\vec{z}(t) = \underbrace{TAT^{-1}}_{\hat{A}}\vec{z}(t) + \underbrace{T\vec{b}}_{\hat{\vec{b}}}u(t)$$
$$\vec{x}(t) = T^{-1}\vec{z}(t)$$

similarity transformation

Converting Systems to CCF (contd.)

T will be full rank, hence non-singular and invertible

5. Form the basis transformation matrix $T \triangleq \begin{bmatrix} \longleftarrow & \vec{q}^T & \longrightarrow \\ \longleftarrow & \vec{q}^T A & \longrightarrow \\ \longleftarrow & \vec{q}^T A^2 & \longrightarrow \\ & \vdots & \\ \longleftarrow & \vec{q}^T A^{n-1} & \longrightarrow \end{bmatrix}$
6. Define $\vec{z}(t) = T\vec{x}(t) \Leftrightarrow \vec{x}(t) = T^{-1}\vec{z}(t)$
7. Write the system in terms of $\vec{z}(t)$:

- $$\frac{d}{dt}\vec{z}(t) = \underbrace{TAT^{-1}}_{\hat{A}}\vec{z}(t) + \underbrace{T\vec{b}}_{\hat{\vec{b}}}\,u(t)$$
$$\vec{x}(t) = T^{-1}\vec{z}(t)$$

equivalent to the original system: $u(t) \mapsto \vec{x}(t)$ is the same

similarity transformation

Converting Systems to CCF (contd.)

T will be full rank, hence non-singular and invertible

5. Form the basis transformation matrix $T \triangleq \begin{bmatrix} \longleftarrow & \vec{q}^T & \longrightarrow \\ \longleftarrow & \vec{q}^T A & \longrightarrow \\ \longleftarrow & \vec{q}^T A^2 & \longrightarrow \\ & \vdots & \\ \longleftarrow & \vec{q}^T A^{n-1} & \longrightarrow \end{bmatrix}$
6. Define $\vec{z}(t) = T\vec{x}(t) \Leftrightarrow \vec{x}(t) = T^{-1}\vec{z}(t)$
7. Write the system in terms of $\vec{z}(t)$:
   - $$\frac{d}{dt}\vec{z}(t) = \underbrace{TAT^{-1}}_{\hat{A}}\vec{z}(t) + \underbrace{T\vec{b}}_{\hat{\vec{b}}}\,u(t)$$
     $$\vec{x}(t) = T^{-1}\vec{z}(t)$$

equivalent to the original system: $u(t) \mapsto \vec{x}(t)$ is the same

similarity transformation

8. $(\hat{A}, \hat{\vec{b}})$ will be in CCF!

Converting Systems to CCF (contd.)

T will be full rank, hence non-singular and invertible

5. Form the basis transformation matrix $T \triangleq \begin{bmatrix} \longleftarrow & \vec{q}^T & \longrightarrow \\ \longleftarrow & \vec{q}^T A & \longrightarrow \\ \longleftarrow & \vec{q}^T A^2 & \longrightarrow \\ & \vdots & \\ \longleftarrow & \vec{q}^T A^{n-1} & \longrightarrow \end{bmatrix}$
6. Define $\vec{z}(t) = T\vec{x}(t) \Leftrightarrow \vec{x}(t) = T^{-1}\vec{z}(t)$
7. Write the system in terms of $\vec{z}(t)$:
   - $$\frac{d}{dt}\vec{z}(t) = \underbrace{TAT^{-1}}_{\hat{A}}\vec{z}(t) + \underbrace{T\vec{b}}_{\hat{\vec{b}}}u(t)$$
     $$\vec{x}(t) = T^{-1}\vec{z}(t)$$

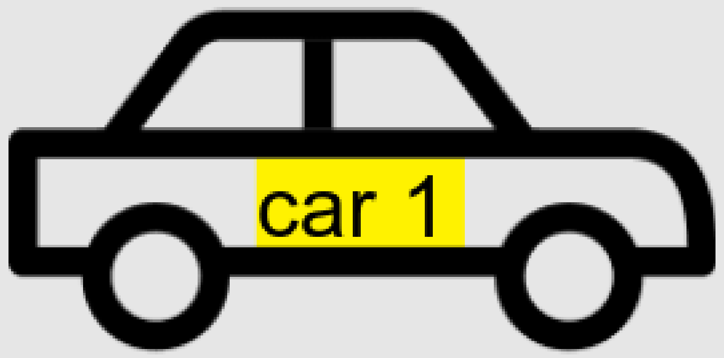
equivalent to the original system: $u(t) \mapsto \vec{x}(t)$ is the same

similarity transformation

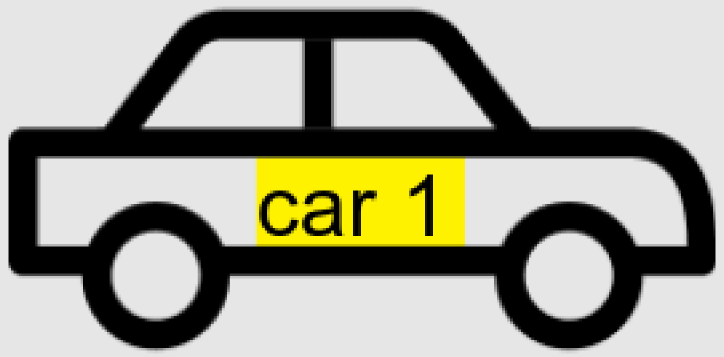
8. $(\hat{A}, \hat{\vec{b}})$ will be in CCF!

- Proof: see the handwritten notes

Example: co-operative car control



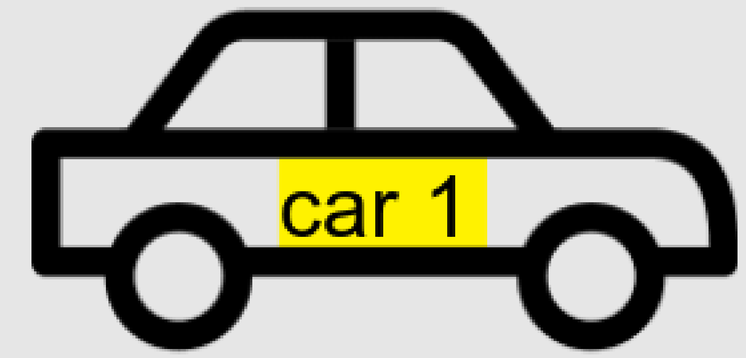
Example: co-operative car control



position: p_1 velocity: v_1 accel: a_1

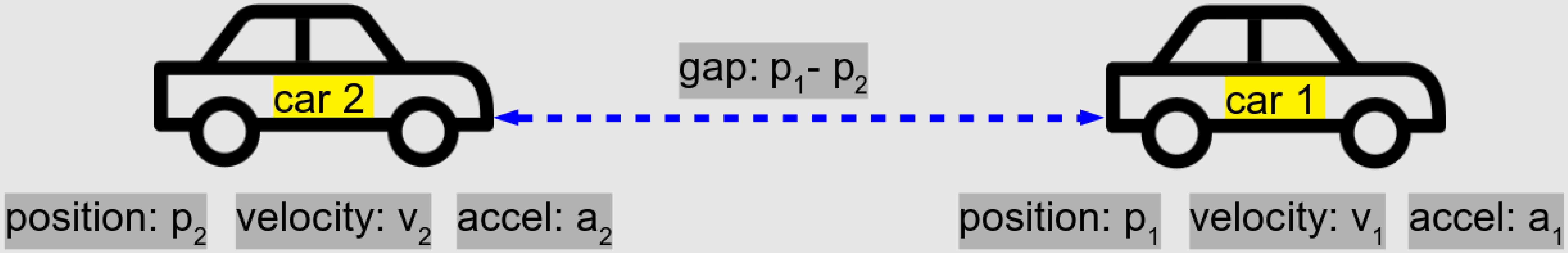
Example: co-operative car control

position: p_2 velocity: v_2 accel: a_2

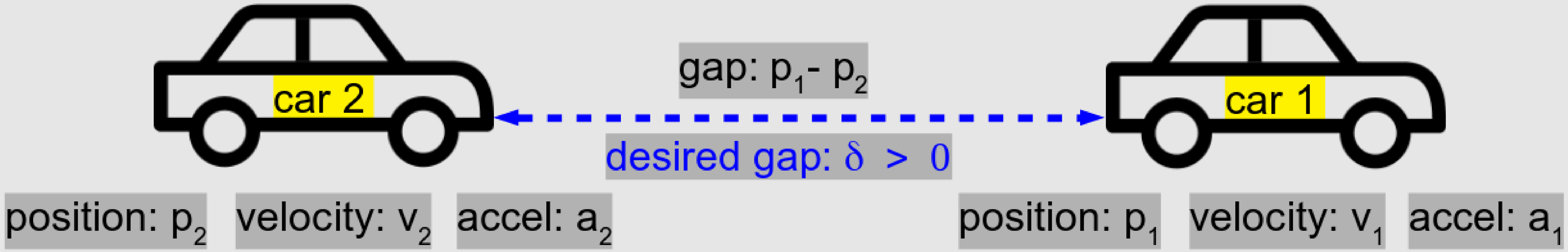


position: p_1 velocity: v_1 accel: a_1

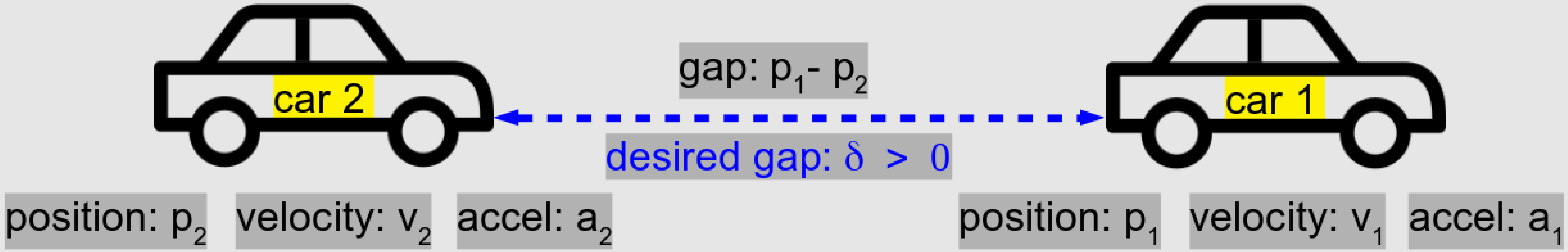
Example: co-operative car control



Example: co-operative car control



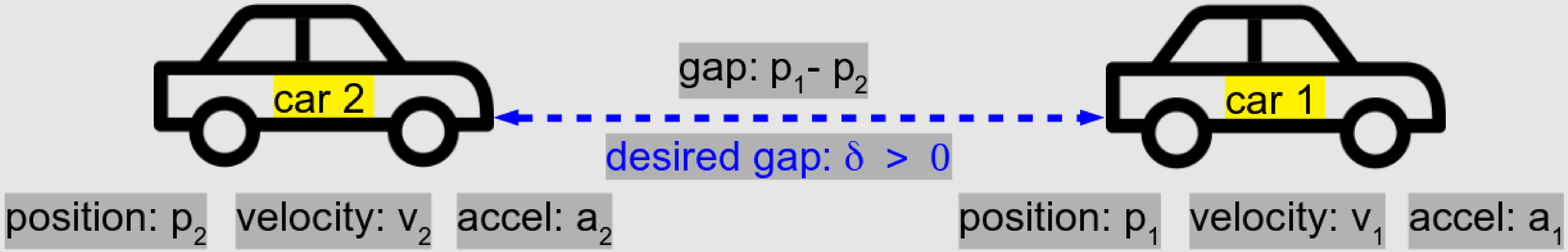
Example: co-operative car control



$$\frac{dp_1}{dt} = v_1(t)$$

$$\frac{dv_1}{dt} = a_1(t)$$

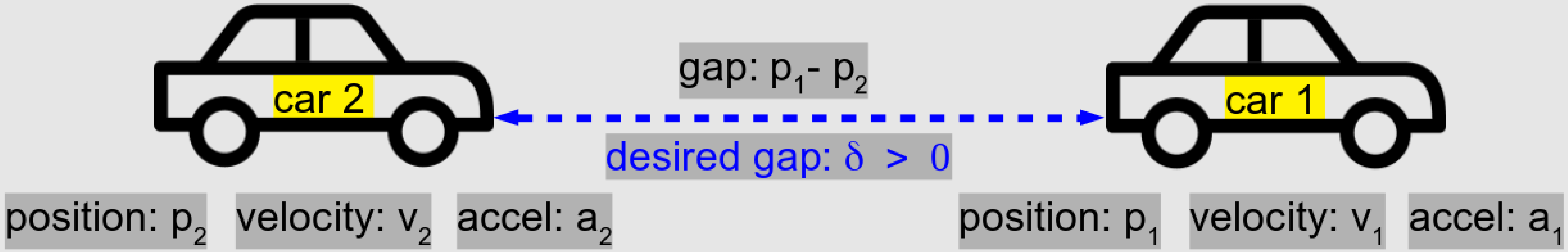
Example: co-operative car control



$$\frac{dp_2}{dt} = v_2(t) \qquad \frac{dp_1}{dt} = v_1(t)$$

$$\frac{dv_2}{dt} = a_2(t) \qquad \frac{dv_1}{dt} = a_1(t)$$

Example: co-operative car control



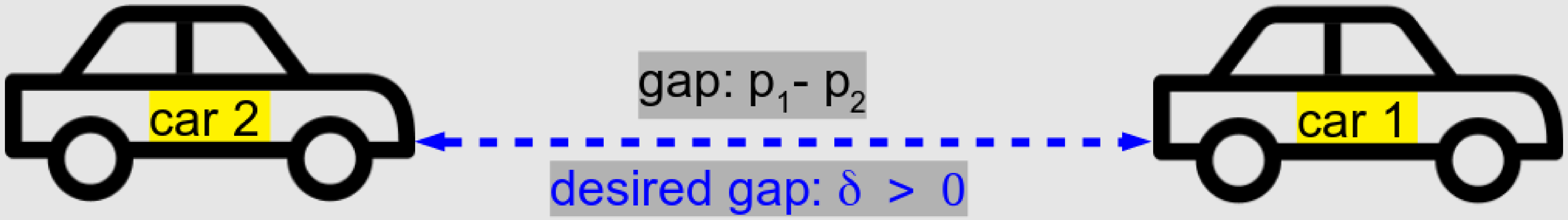
$$\frac{dp_2}{dt} = v_2(t) \qquad\qquad \frac{dp_1}{dt} = v_1(t)$$

$$\frac{dv_2}{dt} = a_2(t) \qquad\qquad \frac{dv_1}{dt} = a_1(t)$$

$$\frac{d(p_1 - p_2)}{dt} = v_1(t) - v_2(t)$$

$$\frac{d(v_1 - v_2)}{dt} = a_1(t) - a_2(t)$$

Example: co-operative car control



position: p_2 velocity: v_2 accel: a_2

position: p_1 velocity: v_1 accel: a_1

$$\frac{dp_2}{dt} = v_2(t) \qquad \frac{dp_1}{dt} = v_1(t)$$

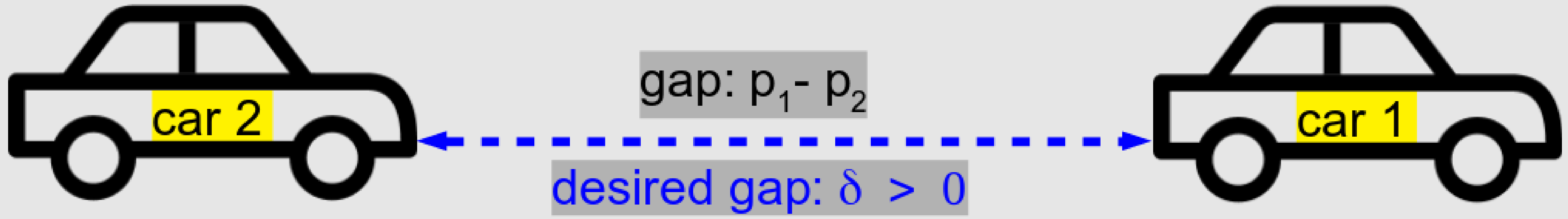
$$\frac{dv_2}{dt} = a_2(t) \qquad \frac{dv_1}{dt} = a_1(t)$$

$$\frac{d(p_1 - p_2)}{dt} = v_1(t) - v_2(t)$$

$$\frac{d(v_1 - v_2)}{dt} = a_1(t) - a_2(t)$$

call this u(t), the input

Example: co-operative car control



position: p_2 velocity: v_2 accel: a_2

position: p_1 velocity: v_1 accel: a_1

$$\frac{dp_2}{dt} = v_2(t) \qquad \frac{dp_1}{dt} = v_1(t)$$

$$\frac{dv_2}{dt} = a_2(t) \qquad \frac{dv_1}{dt} = a_1(t)$$

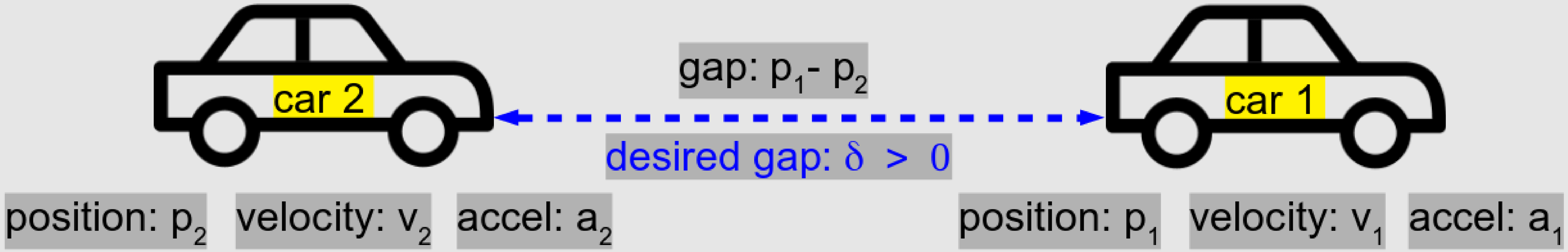
$$\frac{d(p_1 - p_2)}{dt} = v_1(t) - v_2(t)$$

call this $x_2(t)$

$$\frac{d(v_1 - v_2)}{dt} = a_1(t) - a_2(t)$$

call this $u(t)$, the input

Example: co-operative car control



$$\frac{dp_2}{dt} = v_2(t) \qquad \frac{dp_1}{dt} = v_1(t)$$

$$\frac{dv_2}{dt} = a_2(t) \qquad \frac{dv_1}{dt} = a_1(t)$$

define
$x_1(t) = p_1(t) - p_2(t) - \delta$

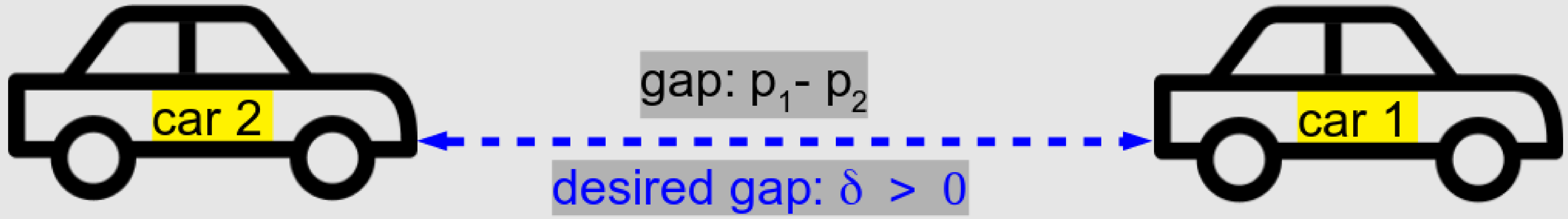
$$\frac{d(p_1 - p_2)}{dt} = v_1(t) - v_2(t)$$

call this $x_2(t)$

$$\frac{d(v_1 - v_2)}{dt} = a_1(t) - a_2(t)$$

call this u(t), the input

Example: co-operative car control



position: p_2 velocity: v_2 accel: a_2

position: p_1 velocity: v_1 accel: a_1

$$\frac{dp_2}{dt} = v_2(t) \qquad \frac{dp_1}{dt} = v_1(t)$$

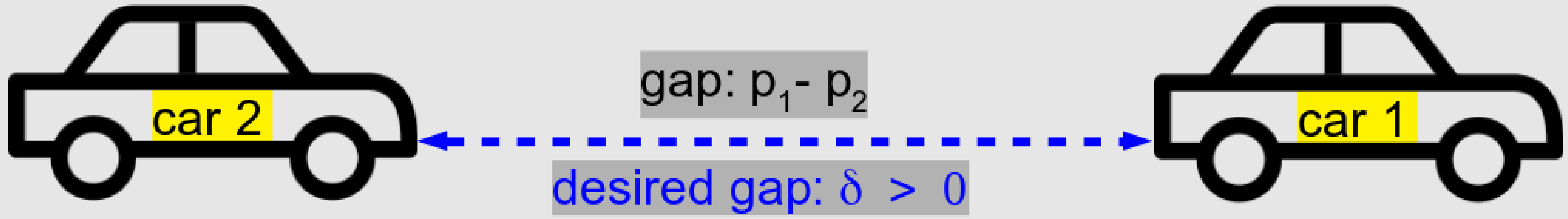
$$\frac{dv_2}{dt} = a_2(t) \qquad \frac{dv_1}{dt} = a_1(t)$$

define
$x_1(t) = p_1(t) - p_2(t) - \delta$

$$\frac{d(\underbrace{p_1 - p_2}_{x_1(t)+\delta})}{dt} = v_1(t) - v_2(t) \longleftarrow \text{call this } x_2(t)$$

$$\frac{d(v_1 - v_2)}{dt} = a_1(t) - a_2(t) \longleftarrow \text{call this } u(t)\text{, the input}$$

Example: co-operative car control



position: p_2 velocity: v_2 accel: a_2

position: p_1 velocity: v_1 accel: a_1

$$\frac{dp_2}{dt} = v_2(t) \qquad \frac{dp_1}{dt} = v_1(t)$$

$$\frac{dv_2}{dt} = a_2(t) \qquad \frac{dv_1}{dt} = a_1(t)$$

define
$x_1(t) = p_1(t) - p_2(t) - \delta$

$$\frac{d(\underbrace{p_1 - p_2}_{x_1(t)+\delta})}{dt} = v_1(t) - v_2(t)$$

call this $x_2(t)$

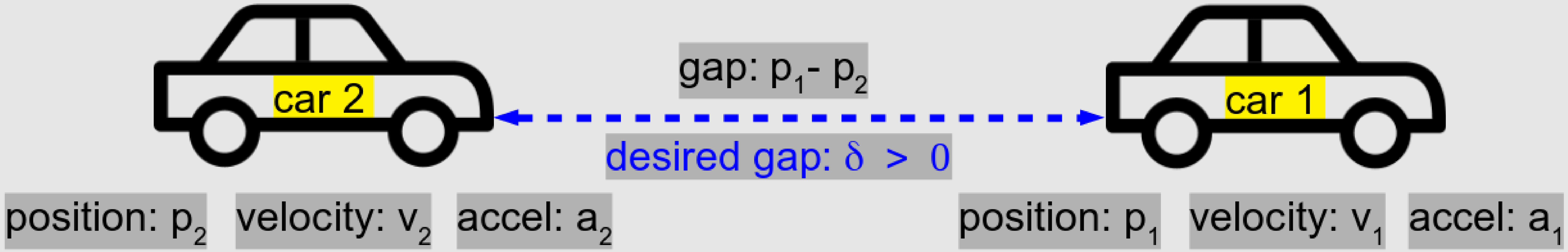
$$\frac{d(v_1 - v_2)}{dt} = a_1(t) - a_2(t)$$

call this u(t), the input

$$\frac{dx_1}{dt} = x_2(t)$$

$$\frac{dx_2}{dt} = u(t)$$

Example: co-operative car control



$$\frac{dp_2}{dt} = v_2(t) \qquad \frac{dp_1}{dt} = v_1(t)$$

$$\frac{dv_2}{dt} = a_2(t) \qquad \frac{dv_1}{dt} = a_1(t)$$

define
$x_1(t) = p_1(t) - p_2(t) - \delta$

$$\frac{d(\underbrace{p_1 - p_2}_{x_1(t)+\delta})}{dt} = v_1(t) - v_2(t)$$

call this $x_2(t)$

$$\frac{d(v_1 - v_2)}{dt} = a_1(t) - a_2(t)$$

call this $u(t)$, the input

$$\frac{dx_1}{dt} = x_2(t)$$

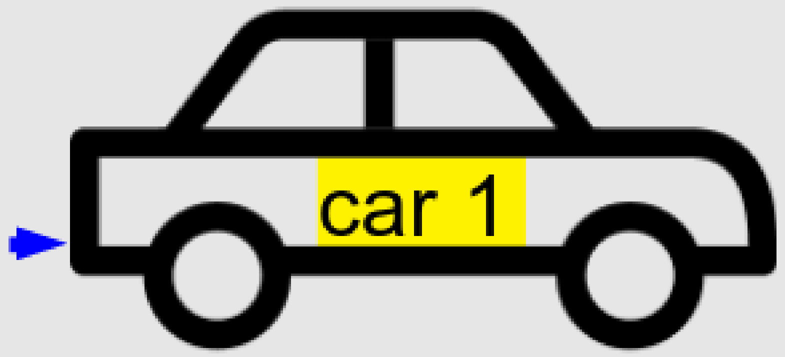
$$\frac{dx_2}{dt} = u(t)$$

$$\Rightarrow \frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \overbrace{\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}}^{A} \overbrace{\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}}^{\vec{x}(t)} + \overbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}^{\vec{b}(t)} u(t)$$

Example: co-operative car control

gap: $p_1 - p_2$

desired gap: $\delta > 0$



position: p_2 velocity: v_2 accel: a_2

position: p_1 velocity: v_1 accel: a_1

$$\frac{dp_2}{dt} = v_2(t) \qquad \frac{dv_2}{dt} = a_2(t)$$

define
$x_1(t) = p_1(t) - p_2(t) - \delta$

$$\frac{dp_1}{dt} = v_1(t) \qquad \frac{dv_1}{dt} = a_1(t)$$

$x_1(t) + \delta$

$$\frac{d(p_1 - p_2)}{dt} = v_1(t) - v_2(t)$$

call this $x_2(t)$

$$\frac{d(v_1 - v_2)}{dt} = a_1(t) - a_2(t)$$

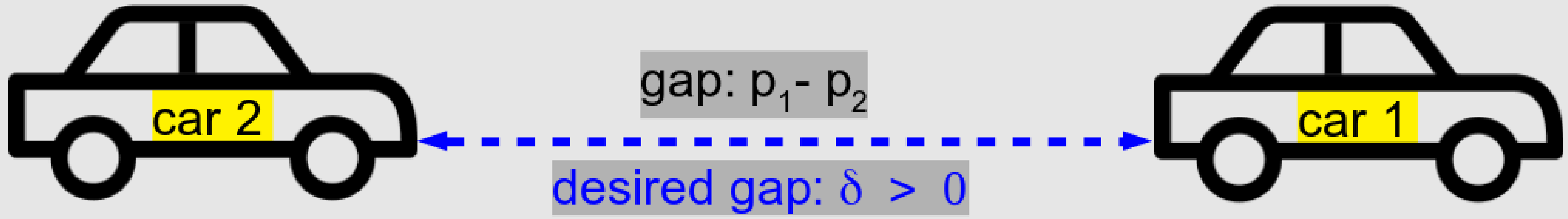
call this u(t), the input

$$\frac{dx_1}{dt} = x_2(t) \qquad \frac{dx_2}{dt} = u(t)$$

$$\Rightarrow \frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \overbrace{\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}}^{A} \overbrace{\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}}^{\vec{x}(t)} + \overbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}^{\vec{b}(t)} u(t)$$

BIBO UNSTABLE

Example: co-operative car control



position: p_2 velocity: v_2 accel: a_2

position: p_1 velocity: v_1 accel: a_1

$$\frac{dp_2}{dt} = v_2(t)$$
$$\frac{dv_2}{dt} = a_2(t)$$

define
$x_1(t) = p_1(t) - p_2(t) - \delta$

$$\frac{dp_1}{dt} = v_1(t)$$
$$\frac{dv_1}{dt} = a_1(t)$$

$x_1(t) + \delta$

$$\frac{d(p_1 - p_2)}{dt} = v_1(t) - v_2(t)$$

call this $x_2(t)$

$$\frac{d(v_1 - v_2)}{dt} = a_1(t) - a_2(t)$$

call this u(t), the input

with IC = $[0, 0]^T$ and u(t) = $-\varepsilon$, the cars will hit each other in

$$T = \sqrt{\frac{2\delta}{\epsilon}}$$

$$\frac{dx_1}{dt} = x_2(t)$$
$$\frac{dx_2}{dt} = u(t)$$

$$\Rightarrow \frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \overbrace{\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}}^{A} \overbrace{\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}}^{\vec{x}(t)} + \overbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}^{\vec{b}(t)} u(t)$$

BIBO UNSTABLE

Co-op. Car Control (contd.)

- introduce state feedback: $A \mapsto \begin{bmatrix} 0 & 1 \\ -k_1 & -k_2 \end{bmatrix}$

Co-op. Car Control (contd.)

- introduce state feedback: $A \mapsto \begin{bmatrix} 0 & 1 \\ \boxed{-k_1} & -k_2 \end{bmatrix}$

position feedback

Co-op. Car Control (contd.)

- introduce state feedback: $A \mapsto \begin{bmatrix} 0 & 1 \\ \boxed{-k_1} & \boxed{-k_2} \end{bmatrix}$

position feedback

velocity feedback

Co-op. Car Control (contd.)

- introduce state feedback: $A \mapsto \begin{bmatrix} 0 & 1 \\ -k_1 & -k_2 \end{bmatrix}$

position feedback

velocity feedback

both sensed by Doppler radar

Co-op. Car Control (contd.)

- introduce state feedback: $A \mapsto \begin{bmatrix} 0 & 1 \\ -k_1 & -k_2 \end{bmatrix}$

position feedback — velocity feedback

both sensed by Doppler radar

- eigenvalues:
  - $\lambda_{1,2} = -\frac{k_2}{2} \pm \frac{1}{2}\sqrt{k_2^2 - 4k_1}$

Co-op. Car Control (contd.)

- introduce state feedback: $A \mapsto \begin{bmatrix} 0 & 1 \\ -k_1 & -k_2 \end{bmatrix}$

position feedback

velocity feedback

both sensed by Doppler radar

- eigenvalues:
  - $\lambda_{1,2} = -\frac{k_2}{2} \pm \frac{1}{2}\sqrt{k_2^2 - 4k_1}$
- stabilization
  - $k_2 > 0, \quad k_1 > 0$ ensures eigenvalues have -ve real parts

Co-op. Car Control (contd.)

- introduce state feedback: $A \mapsto \begin{bmatrix} 0 & 1 \\ -k_1 & -k_2 \end{bmatrix}$

position feedback   velocity feedback

both sensed by Doppler radar

- eigenvalues:
  - $\lambda_{1,2} = -\frac{k_2}{2} \pm \frac{1}{2}\sqrt{k_2^2 - 4k_1}$
- stabilization
  - $k_2 > 0, \quad k_1 > 0$ ensures eigenvalues have -ve real parts
- small errors in the acceleration u(t) → only small changes to the desired distance δ
  - see handwritten notes for details

Controllable Systems can be Stabilized

- So far, we have shown that:
  - CCF systems can be stabilized by feedback
  - Controllable systems can be put in CCF

Controllable Systems can be Stabilized

- So far, we have shown that:
  - CCF systems can be stabilized by feedback
  - Controllable systems can be put in CCF
- → Controllable systs. can be stabilized by feedback

Controllable Systems can be Stabilized

- So far, we have shown that:
  - CCF systems can be stabilized by feedback
  - Controllable systems can be put in CCF
- → Controllable systs. can be stabilized by feedback
  - but <u>not necessary</u> to first convert to CCF to stabilize

Controllable Systems can be Stabilized

- So far, we have shown that:
  - CCF systems can be stabilized by feedback
  - Controllable systems can be put in CCF
- $\rightarrow$ Controllable systs. can be stabilized by feedback
  - but <u>not necessary</u> to first convert to CCF to stabilize
    - ➔ just write out the char. poly. of $A - \vec{b}\vec{k}^T$ directly
      - will be a linear expression in $k_1, k_2, \ldots, k_n$

Controllable Systems can be Stabilized

- So far, we have shown that:
  - CCF systems can be stabilized by feedback
  - Controllable systems can be put in CCF
- $\rightarrow$ Controllable systs. can be stabilized by feedback
  - but <u>not necessary</u> to first convert to CCF to stabilize
    - ➔ just write out the char. poly. of $A - \vec{b}\vec{k}^T$ directly
      - will be a linear expression in k_1, k_2, …, k_n
    - ➔ match coeffs. of λ^k against those of $(\lambda - \lambda_1)(\lambda - \lambda_2)\cdots(\lambda - \lambda_n)$
      - will obtain a linear system of equations in $\vec{k}$: $M\vec{k} = \vec{r}$

Controllable Systems can be Stabilized

- So far, we have shown that:
  - CCF systems can be stabilized by feedback
  - Controllable systems can be put in CCF
- → Controllable systs. can be stabilized by feedback
  - but <u>not necessary</u> to first convert to CCF to stabilize
    - ➔ just write out the char. poly. of $A - \vec{b}\vec{k}^T$ directly
      - will be a linear expression in $k_1, k_2, \ldots, k_n$
    - ➔ match coeffs. of λ^k against those of $(\lambda - \lambda_1)(\lambda - \lambda_2)\cdots(\lambda - \lambda_n)$
      - will obtain a linear system of equations in $\vec{k}$: $M\vec{k} = \vec{r}$

determined by the entries of A, b, and by $\lambda_1, \ldots, \lambda_n$

Controllable Systems can be Stabilized

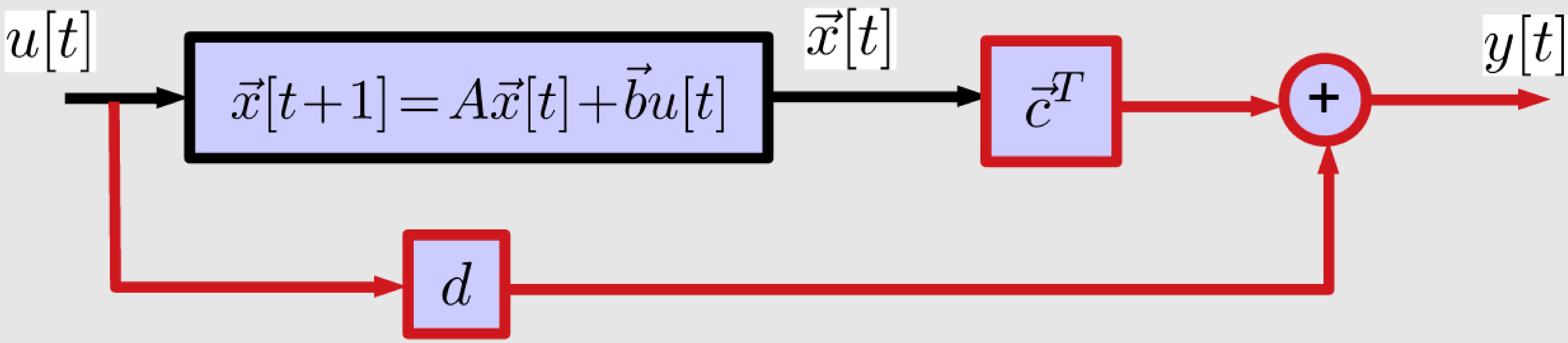
- So far, we have shown that:
  - CCF systems can be stabilized by feedback
  - Controllable systems can be put in CCF
- → Controllable systs. can be stabilized by feedback
  - but <u>not necessary</u> to first convert to CCF to stabilize
    - ➔ just write out the char. poly. of $A - \vec{b}\vec{k}^T$ directly
      - will be a linear expression in $k_1, k_2, \ldots, k_n$
    - ➔ match coeffs. of λ^k against those of $(\lambda - \lambda_1)(\lambda - \lambda_2)\cdots(\lambda - \lambda_n)$
      - will obtain a linear system of equations in $\vec{k}$: $M\vec{k} = \vec{r}$
    - ➔ solve $M\vec{k} = \vec{r}$ for $\vec{k}$ (usually numerically)

determined by the entries of A, b, and by $\lambda_1, \ldots, \lambda_n$

Observability [Back to Discrete]

- suppose we have just a **SCALAR output** $y[t]$
  - i.e., don't have access to all of $\vec{x}[t]$ for feedback

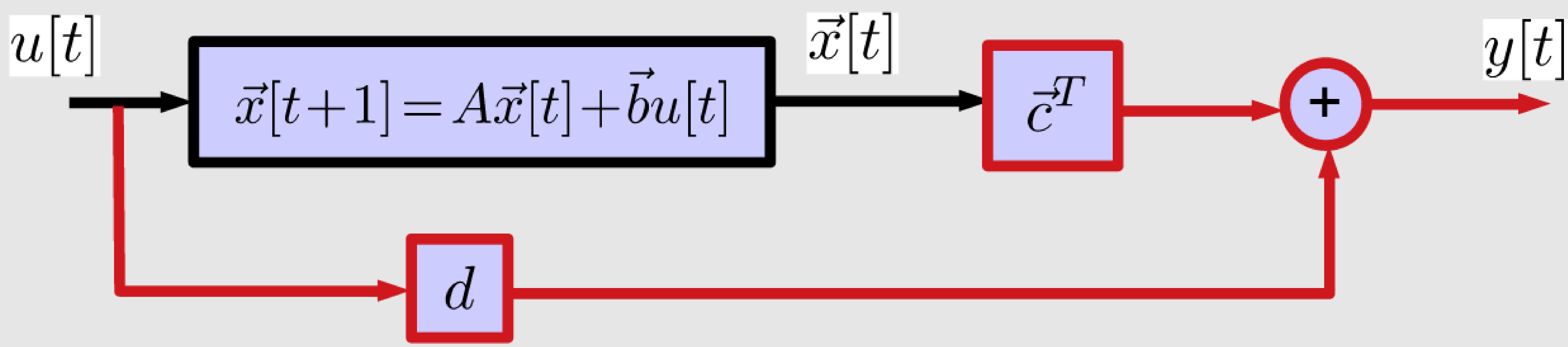
$$y[t] = \vec{c}^T \vec{x}[t] + du[t]$$



Observability [Back to Discrete]

- suppose we have just a **SCALAR output** $y[t]$
  - i.e., don't have access to all of $\vec{x}[t]$ for feedback
  - can we recover $\vec{x}[t]$ just from observations of $y[t]$?

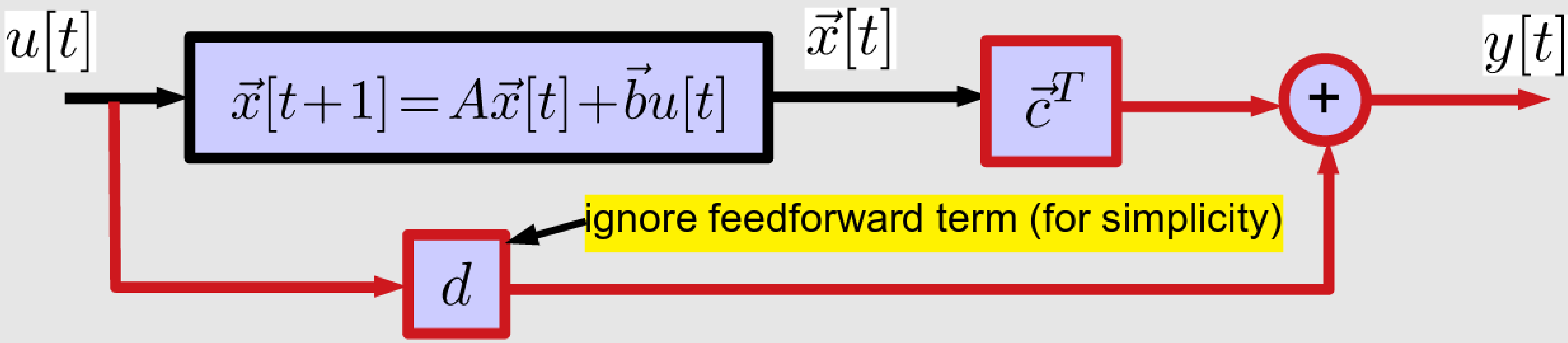
$$y[t] = \vec{c}^T \vec{x}[t] + du[t]$$



Observability [Back to Discrete]

- suppose we have just a **SCALAR output** $y[t]$
  - i.e., don't have access to all of $\vec{x}[t]$ for feedback
  - can we recover $\vec{x}[t]$ just from observations of $y[t]$?

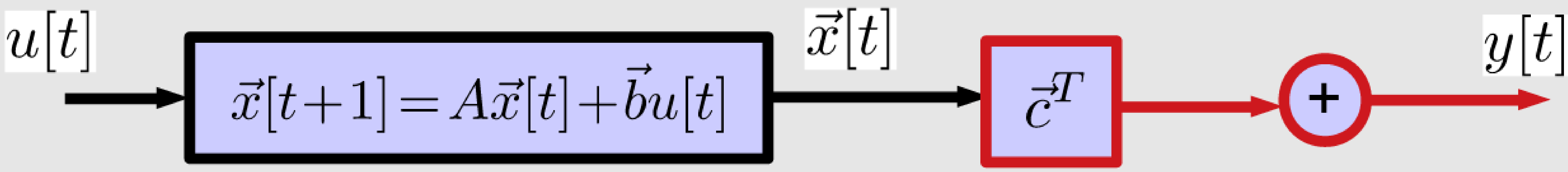
$$y[t] = \vec{c}^T \vec{x}[t] + du[t]$$



Observability [Back to Discrete]

- suppose we have just a **SCALAR output** $y[t]$
  - i.e., don't have access to all of $\vec{x}[t]$ for feedback
  - can we recover $\vec{x}[t]$ just from observations of $y[t]$?

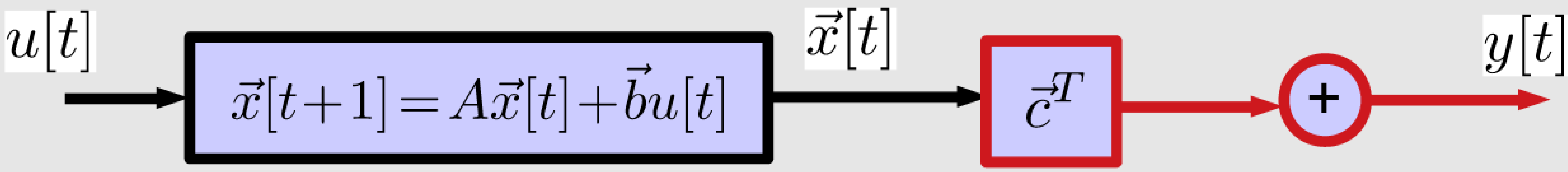
$$y[t] = \vec{c}^T \vec{x}[t] + du[t]$$



Observability [Back to Discrete]

- suppose we have just a **SCALAR output** $y[t]$
  - i.e., don't have access to all of $\vec{x}[t]$ for feedback
  - can we recover $\vec{x}[t]$ just from observations of $y[t]$?

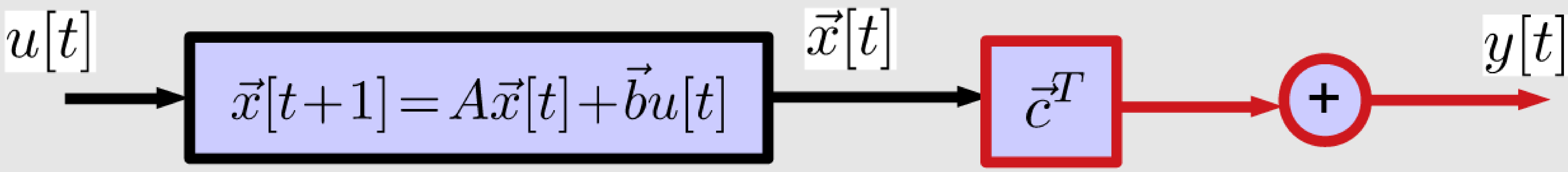
$$y[t] = \vec{c}^T \vec{x}[t]$$



Observability [Back to Discrete]

- suppose we have just a **SCALAR output** $y[t]$
  - i.e., don't have access to all of $\vec{x}[t]$ for feedback
  - can we recover $\vec{x}[t]$ just from observations of $y[t]$?

$$y[t] = \vec{c}^T \vec{x}[t]$$

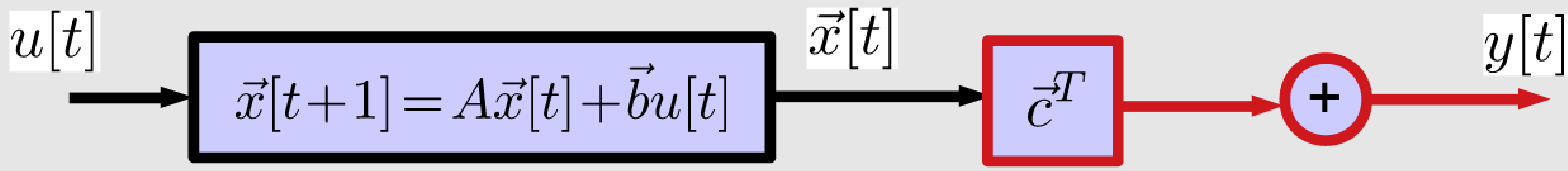


- More precisely:
  - suppose we know: A, $\vec{b}$, $\vec{c}^T$ and u[t]
    - and can measure y(t)
  - can we recover $\vec{x}[t]$?

Observability [Back to Discrete]

- suppose we have just a **SCALAR output** $y[t]$
  - i.e., don't have access to all of $\vec{x}[t]$ for feedback
  - can we recover $\vec{x}[t]$ just from observations of $y[t]$?

$$y[t] = \vec{c}^T \vec{x}[t]$$



- More precisely:
  - suppose we know: A, $\vec{b}$, $\vec{c}^T$ and u[t]
  - and can measure y(t)
  - can we recover $\vec{x}[t]$?

If yes: the system is called **OBSERVABLE**

The Observability Matrix

- We know that $\vec{x}[t] = A^{t-1}\,\vec{x}[0] + \sum_{i=1}^{t} A^{t-i}\vec{b}u[i-1]$

The Observability Matrix

we know (or can calculate) these

- We know that $\vec{x}[t] = \boxed{A^{t-1}}\,\vec{x}[0] + \boxed{\sum_{i=1}^{t} A^{t-i}\vec{b}u[i-1]}$

The Observability Matrix

we know (or can calculate) these

- We know that $\vec{x}[t] = \boxed{A^{t-1}}\,\boxed{\vec{x}[0]} + \boxed{\sum_{i=1}^{t} A^{t-i}\vec{b}u[i-1]}$

the only unknown

The Observability Matrix

we know (or can calculate) these

- We know that $\vec{x}[t] = \boxed{A^{t-1}}\,\boxed{\vec{x}[0]} + \boxed{\sum_{i=1}^{t} A^{t-i}\vec{b}u[i-1]}$

the only unknown

- Suppose u[t]=0

The Observability Matrix

we know (or can calculate) these

- We know that $\vec{x}[t] = \boxed{A^{t-1}}\,\boxed{\vec{x}[0]} + \boxed{\sum_{i=1}^{t} A^{t-i}\vec{b}u[i-1]}$

the only unknown

- Suppose u[t]=0
  - then $\vec{x}[t] = A^{t-1}\vec{x}[0]$. Write out $y[t] = \vec{c}^T\vec{x}[t]$:

$$
\begin{aligned}
y[0] &= \vec{c}^T\vec{x}[0] \\
y[1] &= \vec{c}^T\vec{x}[1] = \vec{c}^T A\vec{x}[0] \\
y[2] &= \vec{c}^T\vec{x}[2] = \vec{c}^T A^2\vec{x}[0] \\
&\vdots \\
y[n-1] &= \vec{c}^T\vec{x}[2] = \vec{c}^T A^{n-1}\vec{x}[0]
\end{aligned}
$$

The Observability Matrix

we know (or can calculate) these

- We know that $\vec{x}[t] = \boxed{A^{t-1}}\boxed{\vec{x}[0]} + \boxed{\sum_{i=1}^{t} A^{t-i}\vec{b}u[i-1]}$

the only unknown

- Suppose u[t]=0
  - then $\vec{x}[t] = A^{t-1}\vec{x}[0]$. Write out $y[t] = \vec{c}^T\vec{x}[t]$:

$$\begin{bmatrix} y[0] \\ y[1] \\ y[2] \\ \vdots \\ y[n-1] \end{bmatrix} \triangleq \begin{bmatrix} \longleftarrow & \vec{c}^T & \longrightarrow \\ \longleftarrow & \vec{c}^T A & \longrightarrow \\ \longleftarrow & \vec{c}^T A^2 & \longrightarrow \\ & \vdots & \\ \longleftarrow & \vec{c}^T A^{n-1} & \longrightarrow \end{bmatrix} \vec{x}[0]$$

$$\begin{aligned} y[0] &= \vec{c}^T\vec{x}[0] \\ y[1] &= \vec{c}^T\vec{x}[1] = \vec{c}^T A\vec{x}[0] \\ y[2] &= \vec{c}^T\vec{x}[2] = \vec{c}^T A^2\vec{x}[0] \\ &\vdots \\ y[n-1] &= \vec{c}^T\vec{x}[2] = \vec{c}^T A^{n-1}\vec{x}[0] \end{aligned}$$

The Observability Matrix

we know (or can calculate) these

- We know that $\vec{x}[t] = \boxed{A^{t-1}}\,\boxed{\vec{x}[0]} + \boxed{\sum_{i=1}^{t} A^{t-i}\vec{b}u[i-1]}$

the only unknown

- Suppose u[t]=0
  - then $\vec{x}[t] = A^{t-1}\vec{x}[0]$. Write out $y[t] = \vec{c}^T\vec{x}[t]$:

$$\begin{bmatrix} y[0] \\ y[1] \\ y[2] \\ \vdots \\ y[n-1] \end{bmatrix} \triangleq \boxed{\begin{bmatrix} \longleftarrow & \vec{c}^T & \longrightarrow \\ \longleftarrow & \vec{c}^T A & \longrightarrow \\ \longleftarrow & \vec{c}^T A^2 & \longrightarrow \\ & \vdots & \\ \longleftarrow & \vec{c}^T A^{n-1} & \longrightarrow \end{bmatrix}} \vec{x}[0]$$

$$\begin{aligned} y[0] &= \vec{c}^T\vec{x}[0] \\ y[1] &= \vec{c}^T\vec{x}[1] = \vec{c}^T A\vec{x}[0] \\ y[2] &= \vec{c}^T\vec{x}[2] = \vec{c}^T A^2\vec{x}[0] \\ &\vdots \\ y[n-1] &= \vec{c}^T\vec{x}[2] = \vec{c}^T A^{n-1}\vec{x}[0] \end{aligned}$$

observability matrix (nxn)

The Observability Matrix

we know (or can calculate) these

- We know that $\vec{x}[t] = \boxed{A^{t-1}}\,\boxed{\vec{x}[0]} + \boxed{\sum_{i=1}^{t} A^{t-i}\vec{b}u[i-1]}$

the only unknown

- Suppose u[t]=0
  - then $\vec{x}[t] = A^{t-1}\vec{x}[0]$. Write out $y[t] = \vec{c}^T\vec{x}[t]$:

$$\begin{bmatrix} y[0] \\ y[1] \\ y[2] \\ \vdots \\ y[n-1] \end{bmatrix} \triangleq \boxed{\begin{bmatrix} \longleftarrow & \vec{c}^T & \longrightarrow \\ \longleftarrow & \vec{c}^T A & \longrightarrow \\ \longleftarrow & \vec{c}^T A^2 & \longrightarrow \\ & \vdots & \\ \longleftarrow & \vec{c}^T A^{n-1} & \longrightarrow \end{bmatrix}} \vec{x}[0]$$

$$\begin{aligned} y[0] &= \vec{c}^T\vec{x}[0] \\ y[1] &= \vec{c}^T\vec{x}[1] = \vec{c}^T A\vec{x}[0] \\ y[2] &= \vec{c}^T\vec{x}[2] = \vec{c}^T A^2\vec{x}[0] \\ &\vdots \\ y[n-1] &= \vec{c}^T\vec{x}[2] = \vec{c}^T A^{n-1}\vec{x}[0] \end{aligned}$$

observability matrix (nxn)

must be full-rank/non-singular/invertible to recover $\vec{x}(t)$ uniquely from measurements of y(t)

Observability: An Example

- $$\begin{bmatrix} x_1[t+1] \\ x_2[t+1] \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x_1[t] \\ x_2[t] \end{bmatrix}$$

Observability: An Example

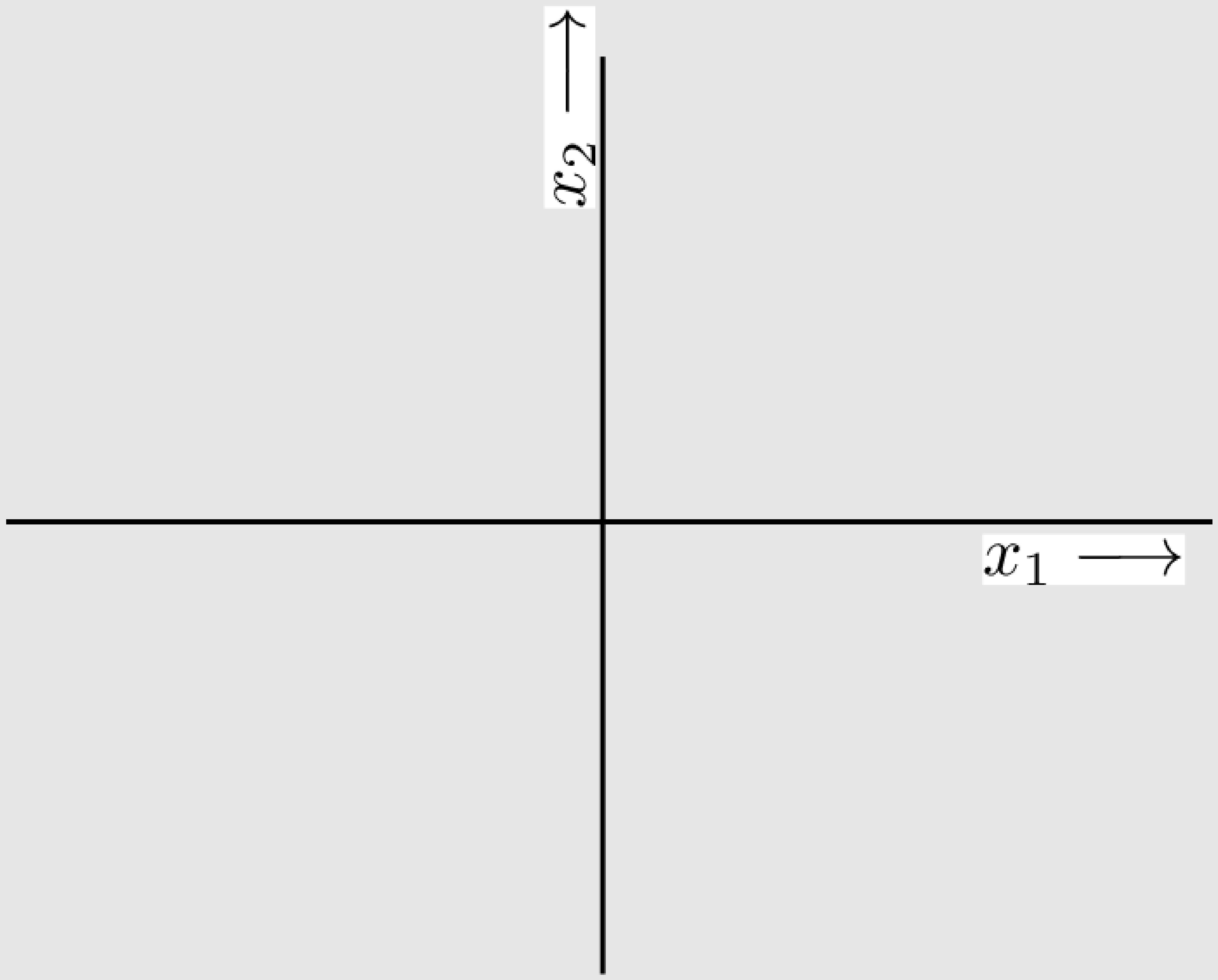
- $$\begin{bmatrix} x_1[t+1] \\ x_2[t+1] \end{bmatrix} = \boxed{\begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}} \begin{bmatrix} x_1[t] \\ x_2[t] \end{bmatrix}$$

this is a “rotation matrix” - call it A

Observability: An Example

- $$\begin{bmatrix} x_1[t+1] \\ x_2[t+1] \end{bmatrix} = \boxed{\begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}} \begin{bmatrix} x_1[t] \\ x_2[t] \end{bmatrix}$$

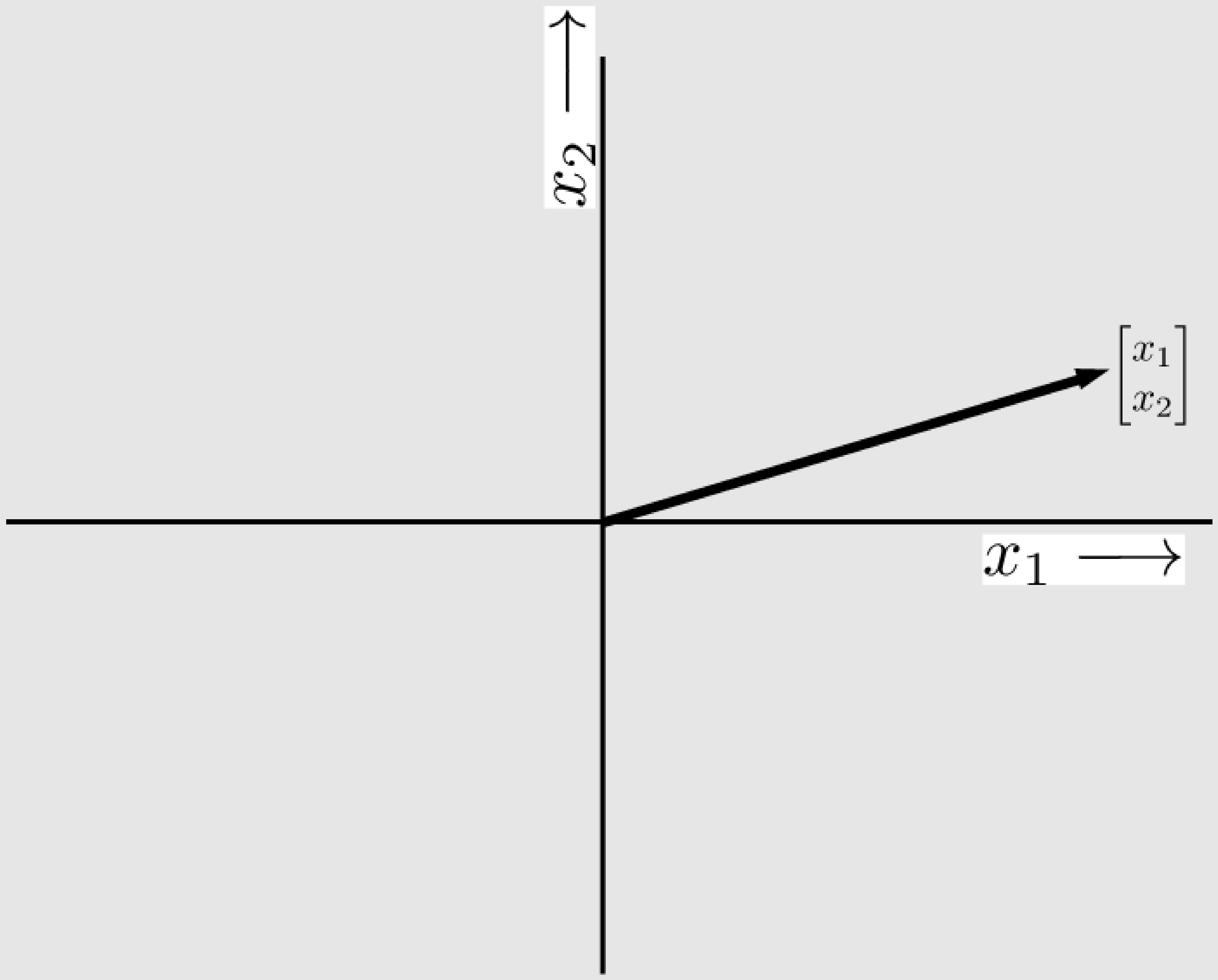
this is a "rotation matrix" - call it A



Observability: An Example

- $$\begin{bmatrix} x_1[t+1] \\ x_2[t+1] \end{bmatrix} = \boxed{\begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}} \begin{bmatrix} x_1[t] \\ x_2[t] \end{bmatrix}$$

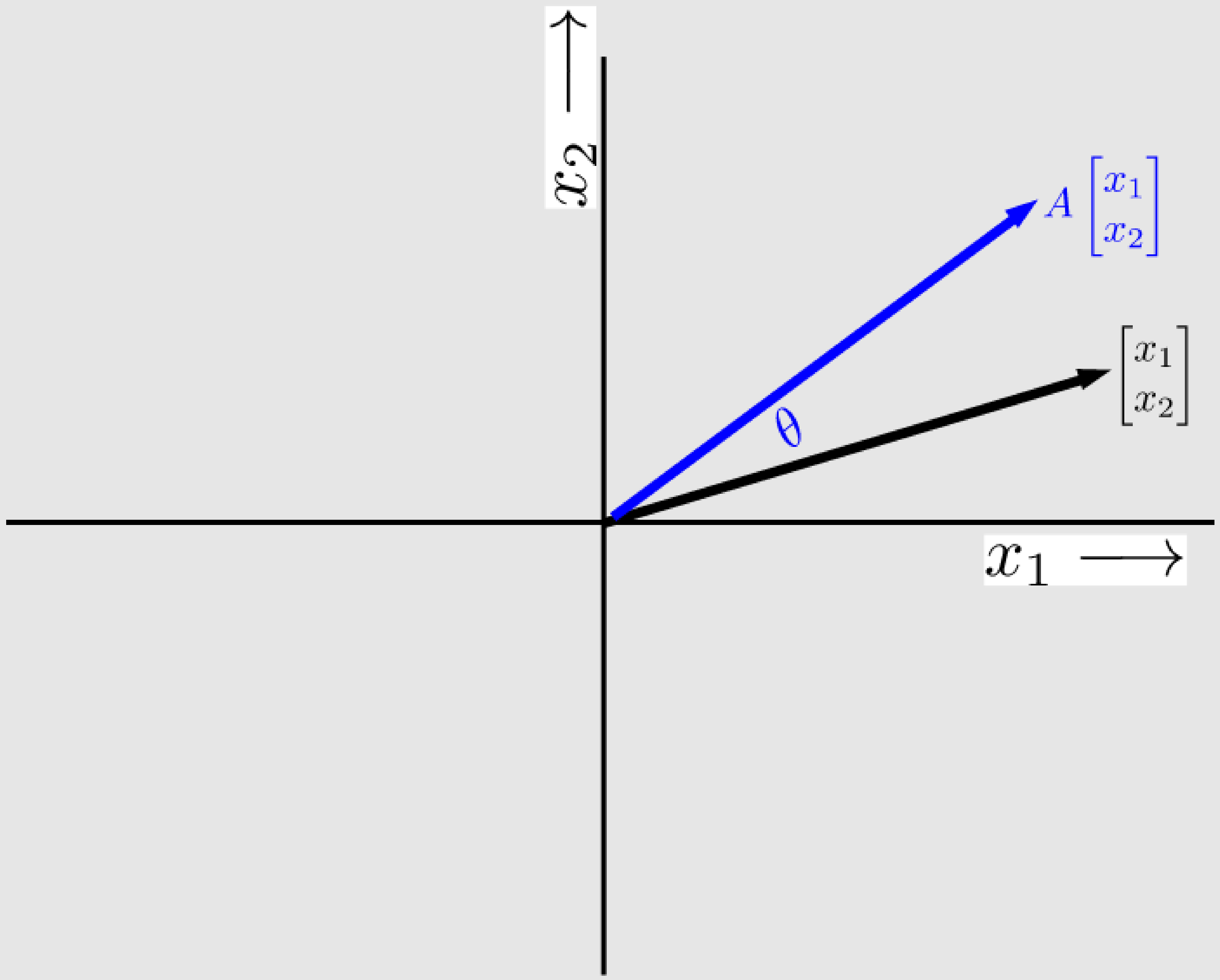
this is a "rotation matrix" - call it A



Observability: An Example

- $$\begin{bmatrix} x_1[t+1] \\ x_2[t+1] \end{bmatrix} = \boxed{\begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}} \begin{bmatrix} x_1[t] \\ x_2[t] \end{bmatrix}$$

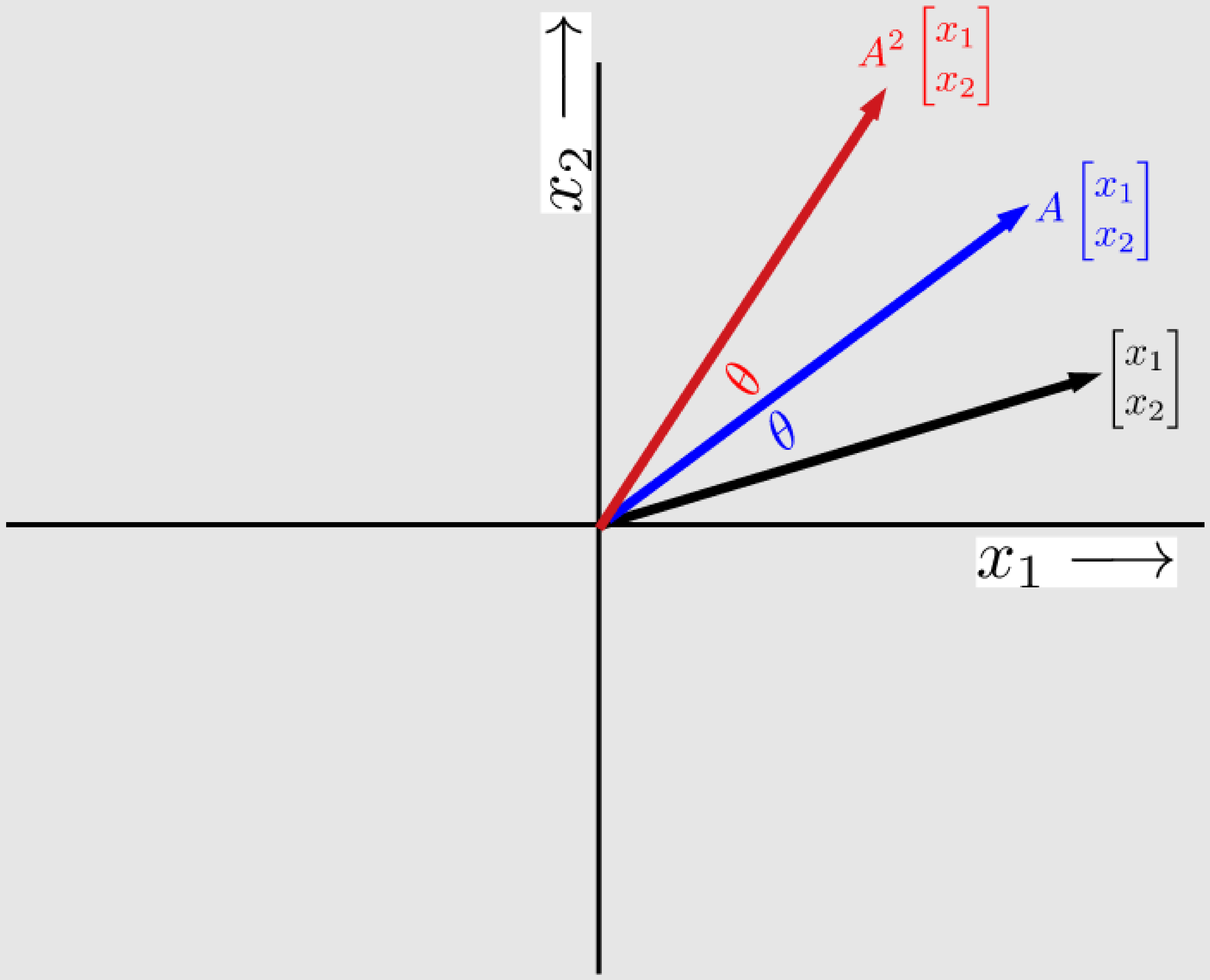
this is a "rotation matrix" - call it A



Observability: An Example

- $$\begin{bmatrix} x_1[t+1] \\ x_2[t+1] \end{bmatrix} = \boxed{\begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}} \begin{bmatrix} x_1[t] \\ x_2[t] \end{bmatrix}$$

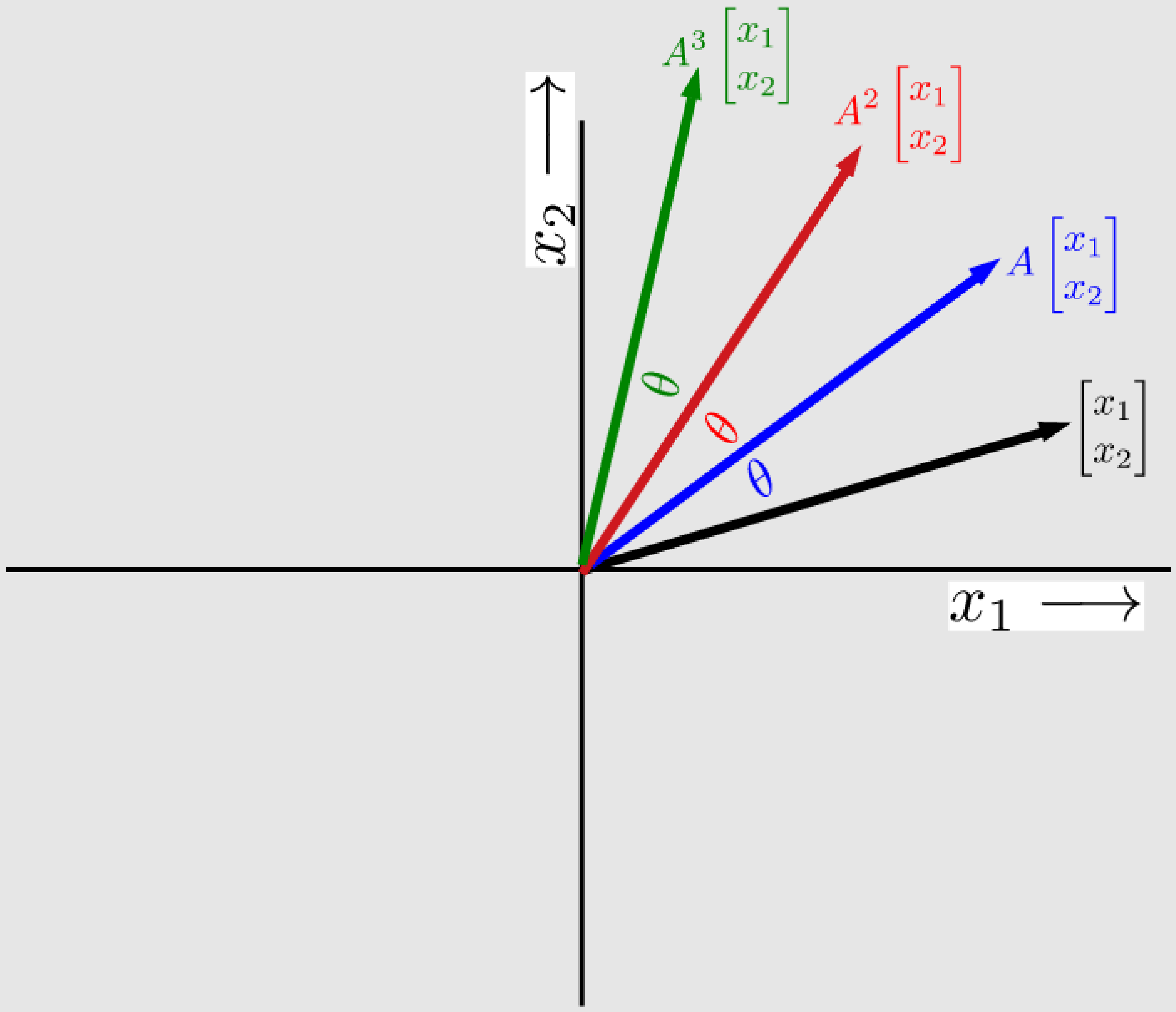
this is a "rotation matrix" - call it A



Observability: An Example

- $$\begin{bmatrix} x_1[t+1] \\ x_2[t+1] \end{bmatrix} = \boxed{\begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}} \begin{bmatrix} x_1[t] \\ x_2[t] \end{bmatrix}$$

this is a “rotation matrix” - call it A

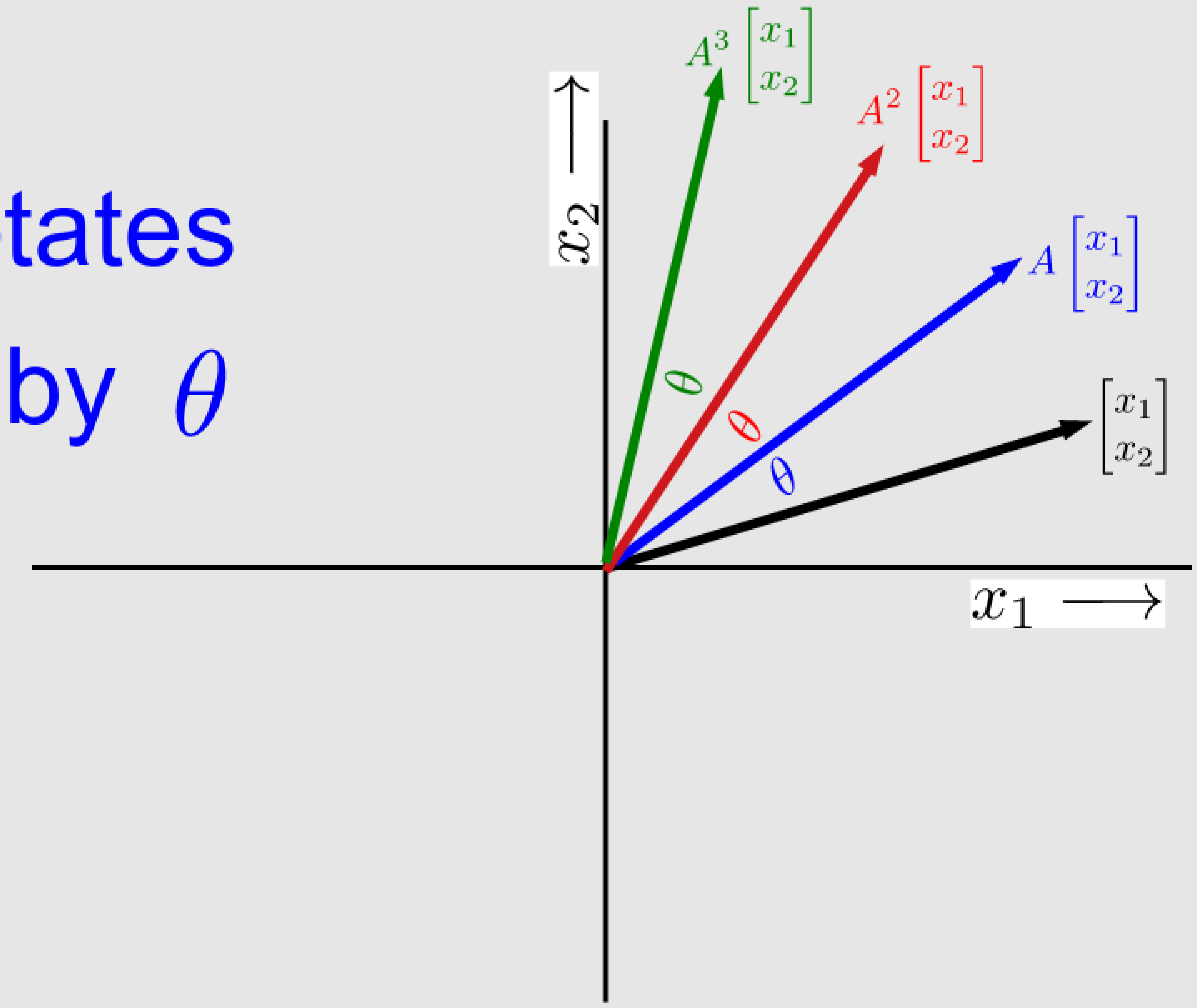


Observability: An Example

- $$\begin{bmatrix} x_1[t+1] \\ x_2[t+1] \end{bmatrix} = \boxed{\begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}} \begin{bmatrix} x_1[t] \\ x_2[t] \end{bmatrix}$$

this is a "rotation matrix" - call it A

- Each application of A rotates by θ



Observability: Example (contd.)

- $\begin{bmatrix} x_1[t+1] \\ x_2[t+1] \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x_1[t] \\ x_2[t] \end{bmatrix}$,

Observability: Example (contd.)

- $\begin{bmatrix} x_1[t+1] \\ x_2[t+1] \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x_1[t] \\ x_2[t] \end{bmatrix}$, $y[t] = x_1[t] = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{\vec{c}^T} \vec{x}[t]$

Observability: Example (contd.)

- $\begin{bmatrix} x_1[t+1] \\ x_2[t+1] \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x_1[t] \\ x_2[t] \end{bmatrix}$, $y[t] = x_1[t] = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{\vec{c}^T} \vec{x}[t]$

- Observability matrix: $O \triangleq \begin{bmatrix} \leftarrow \vec{c}^T \rightarrow \\ \leftarrow \vec{c}^T A \rightarrow \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \cos(\theta) & -\sin(\theta) \end{bmatrix}$

Observability: Example (contd.)

- $\begin{bmatrix} x_1[t+1] \\ x_2[t+1] \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x_1[t] \\ x_2[t] \end{bmatrix}$, $y[t] = x_1[t] = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{\vec{c}^T} \vec{x}[t]$
- Observability matrix: $O \triangleq \begin{bmatrix} \leftarrow \vec{c}^T \rightarrow \\ \leftarrow \vec{c}^T A \rightarrow \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \cos(\theta) & -\sin(\theta) \end{bmatrix}$
- Determinant of O: $det(O) = -\sin(\theta)$

Observability: Example (contd.)

- $\begin{bmatrix} x_1[t+1] \\ x_2[t+1] \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x_1[t] \\ x_2[t] \end{bmatrix}$, $y[t] = x_1[t] = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{\vec{c}^T} \vec{x}[t]$

- Observability matrix: $O \triangleq \begin{bmatrix} \leftarrow \vec{c}^T \rightarrow \\ \leftarrow \vec{c}^T A \rightarrow \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \cos(\theta) & -\sin(\theta) \end{bmatrix}$

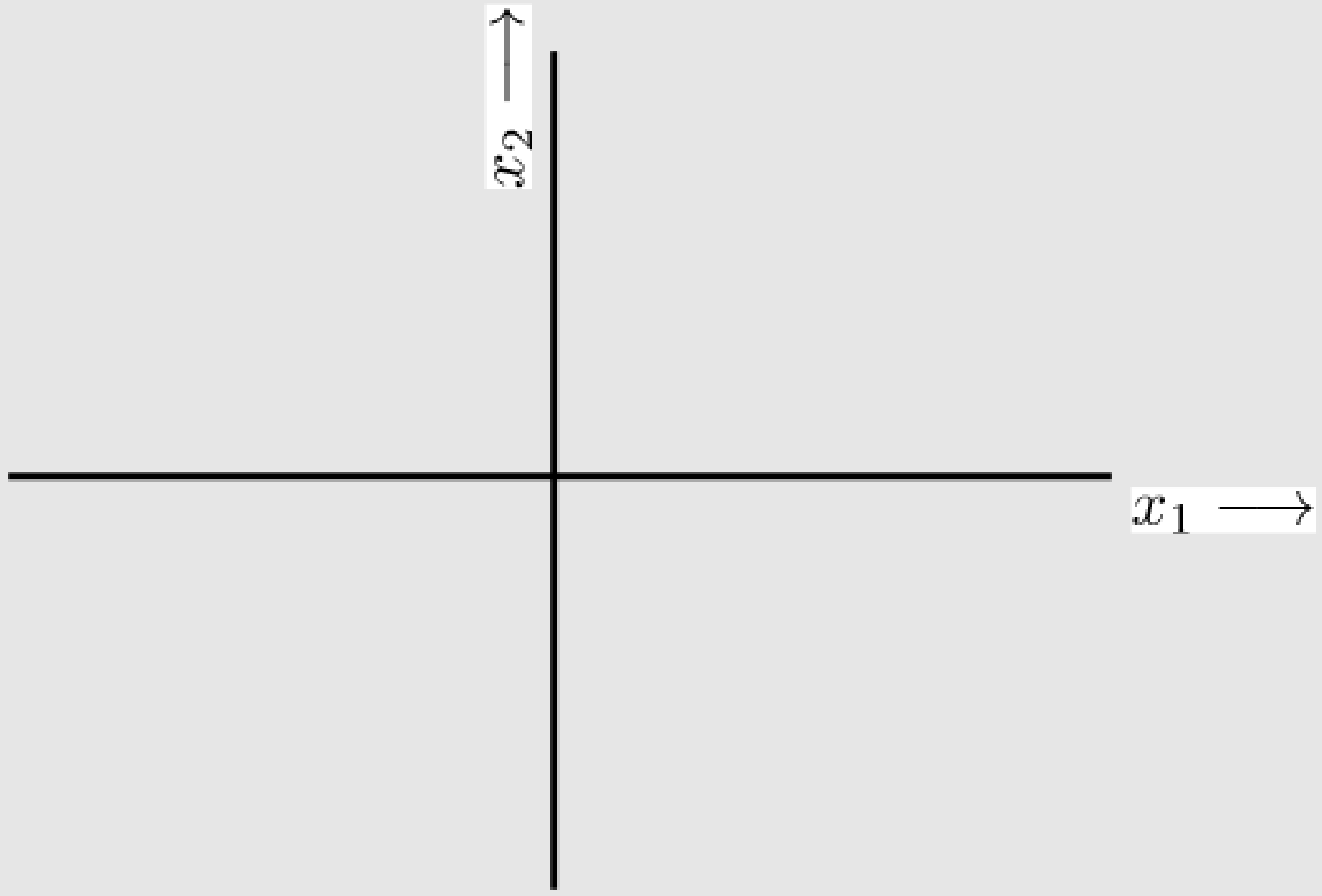
- Determinant of O: $det(O) = -\sin(\theta)$
  - non-zero if $\theta \neq 0, \pi, 2\pi, \cdots, i\pi$ $\rightarrow$ observable

Observability: Example (contd.)

- $\begin{bmatrix} x_1[t+1] \\ x_2[t+1] \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x_1[t] \\ x_2[t] \end{bmatrix}$, $y[t] = x_1[t] = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{\vec{c}^T} \vec{x}[t]$
- Observability matrix: $O \triangleq \begin{bmatrix} \leftarrow \vec{c}^T \rightarrow \\ \leftarrow \vec{c}^T A \rightarrow \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \cos(\theta) & -\sin(\theta) \end{bmatrix}$
- Determinant of O: $det(O) = -\sin(\theta)$
  - non-zero if $\theta \neq 0, \pi, 2\pi, \cdots, i\pi$ $\rightarrow$ observable
  - 0 if $\theta = i\pi$ $\rightarrow$ not observable

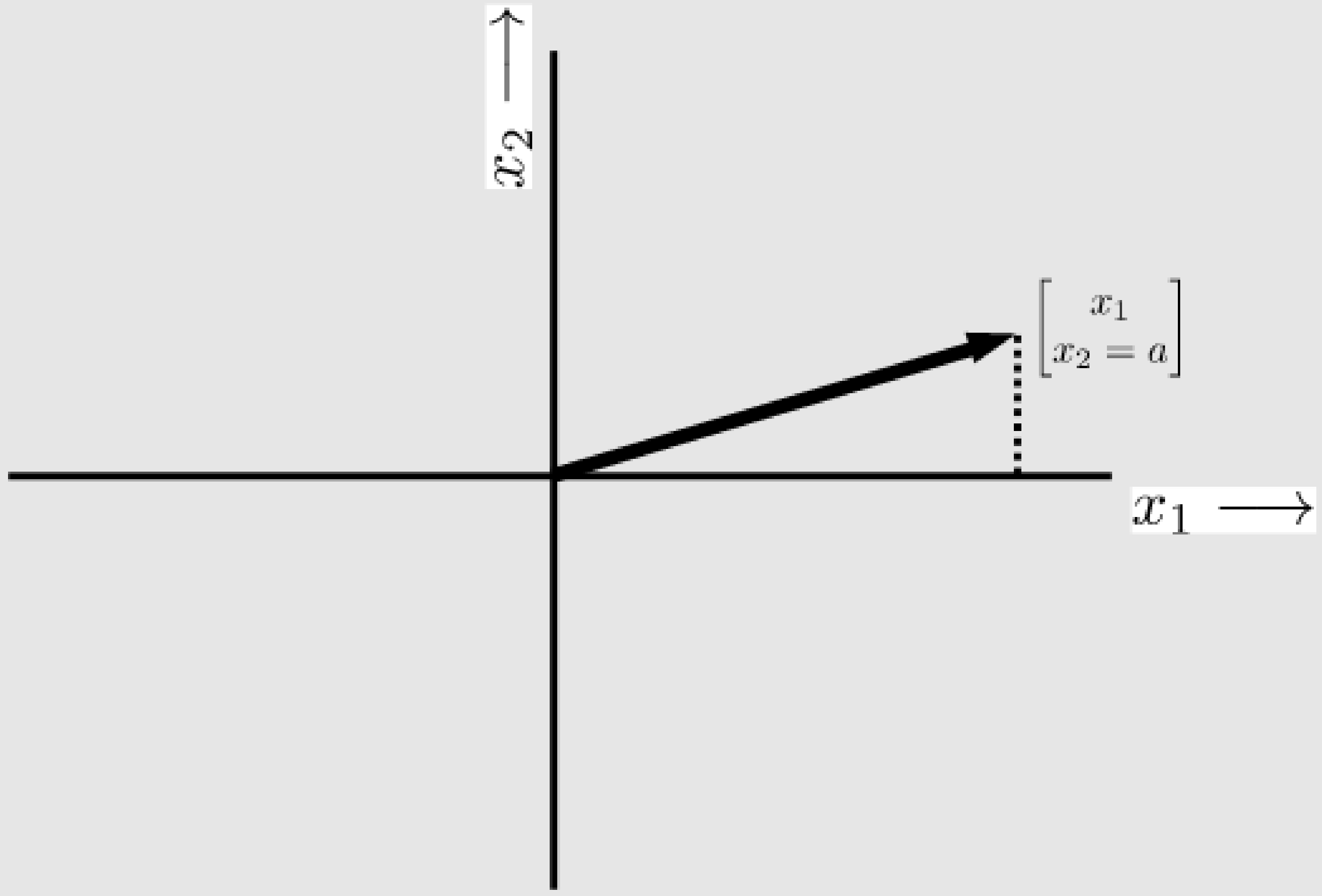
Observability: Example (contd.)

- $\begin{bmatrix} x_1[t+1] \\ x_2[t+1] \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x_1[t] \\ x_2[t] \end{bmatrix}$, $y[t] = x_1[t] = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{\vec{c}^T} \vec{x}[t]$
- Observability matrix: $O \triangleq \begin{bmatrix} \leftarrow \vec{c}^T \rightarrow \\ \leftarrow \vec{c}^T A \rightarrow \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \cos(\theta) & -\sin(\theta) \end{bmatrix}$
- Determinant of O: $det(O) = -\sin(\theta)$
  - non-zero if $\theta \neq 0, \pi, 2\pi, \cdots, i\pi$ → observable
  - 0 if $\theta = i\pi$ → not observable



Observability: Example (contd.)

- $\begin{bmatrix} x_1[t+1] \\ x_2[t+1] \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x_1[t] \\ x_2[t] \end{bmatrix}$, $y[t] = x_1[t] = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{\vec{c}^T} \vec{x}[t]$
- Observability matrix: $O \triangleq \begin{bmatrix} \leftarrow \vec{c}^T \rightarrow \\ \leftarrow \vec{c}^T A \rightarrow \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \cos(\theta) & -\sin(\theta) \end{bmatrix}$
- Determinant of O: $det(O) = -\sin(\theta)$
  - non-zero if $\theta \neq 0, \pi, 2\pi, \cdots, i\pi$ → observable
  - 0 if $\theta = i\pi$ → not observable

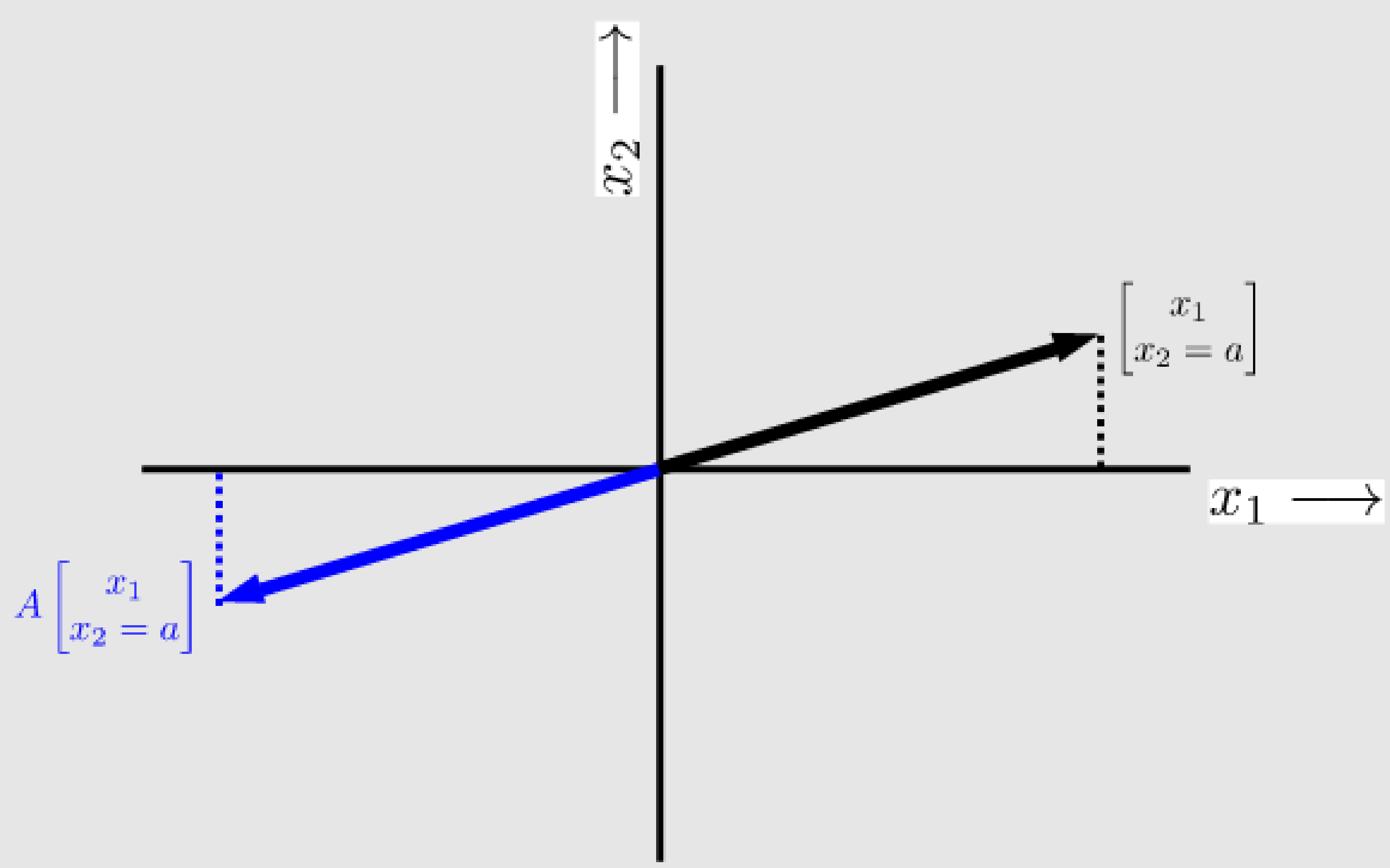


Observability: Example (contd.)

- $\begin{bmatrix} x_1[t+1] \\ x_2[t+1] \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x_1[t] \\ x_2[t] \end{bmatrix}$, $y[t] = x_1[t] = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{\vec{c}^T} \vec{x}[t]$

- Observability matrix: $O \triangleq \begin{bmatrix} \leftarrow \vec{c}^T \rightarrow \\ \leftarrow \vec{c}^T A \rightarrow \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \cos(\theta) & -\sin(\theta) \end{bmatrix}$

- Determinant of O: $det(O) = -\sin(\theta)$
  - non-zero if $\theta \neq 0, \pi, 2\pi, \cdots, i\pi$ → observable
  - 0 if $\theta = i\pi$ → not observable

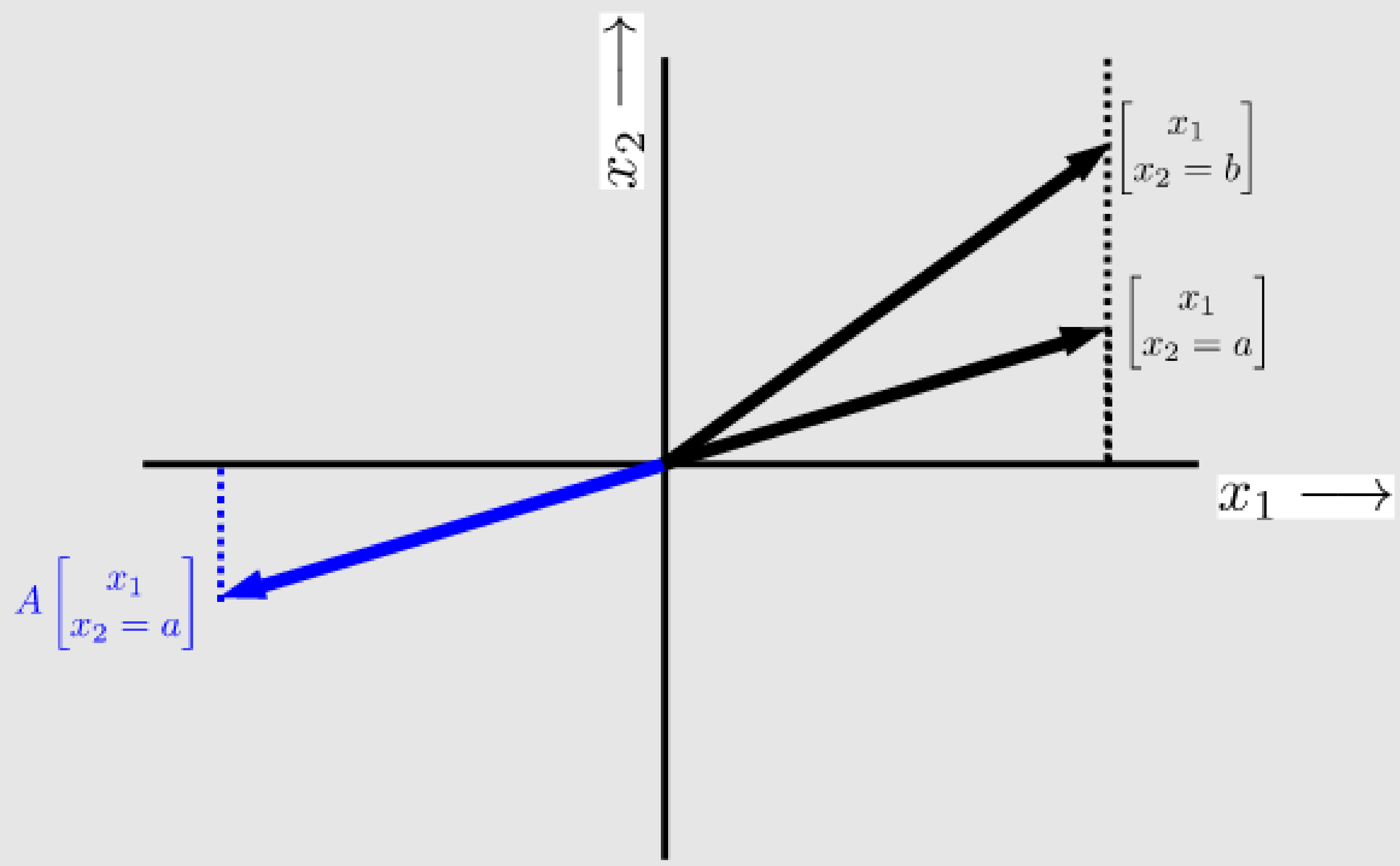


Observability: Example (contd.)

- $\begin{bmatrix} x_1[t+1] \\ x_2[t+1] \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x_1[t] \\ x_2[t] \end{bmatrix}$, $y[t] = x_1[t] = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{\vec{c}^T} \vec{x}[t]$

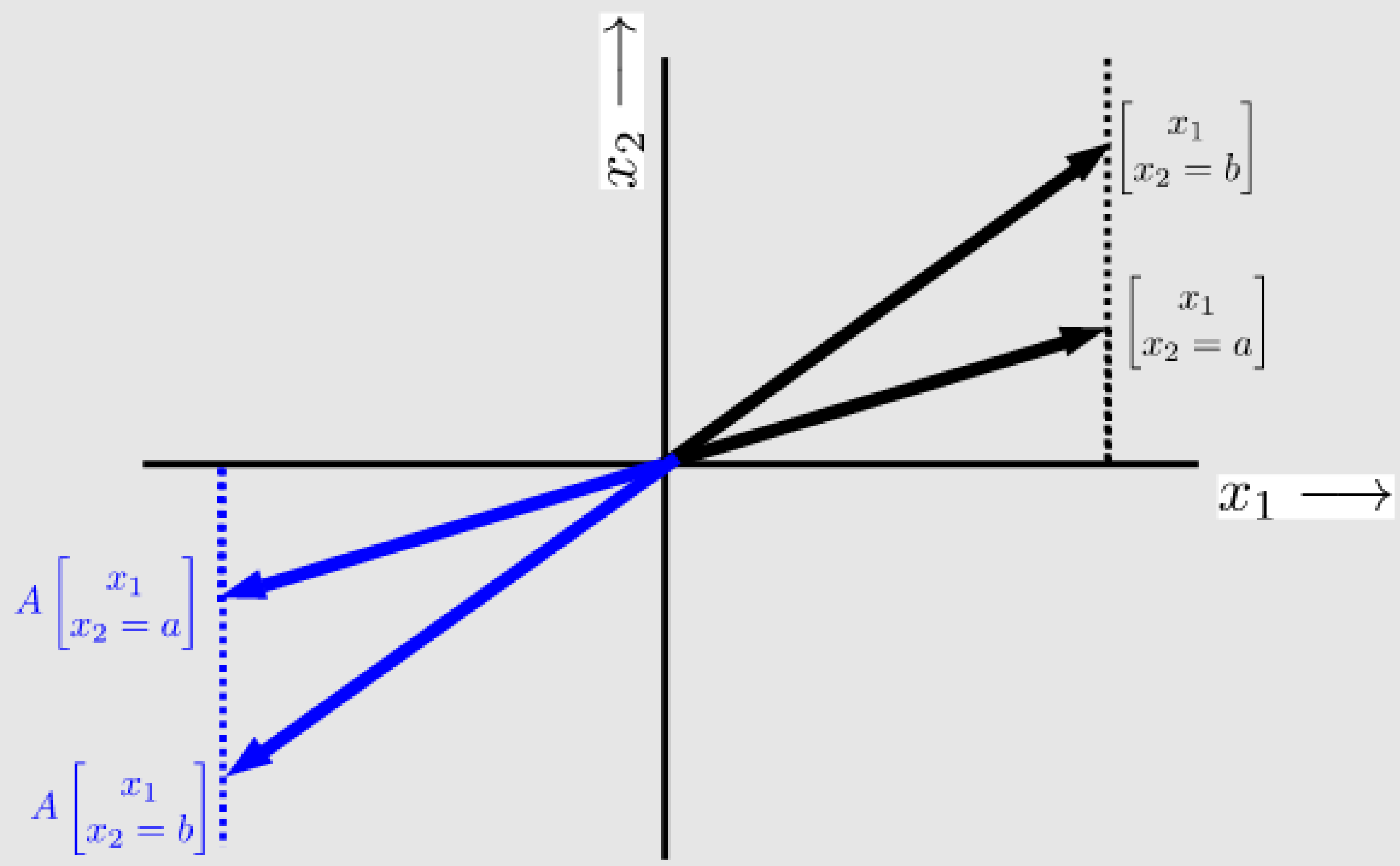
- Observability matrix: $O \triangleq \begin{bmatrix} \leftarrow \vec{c}^T \rightarrow \\ \leftarrow \vec{c}^T A \rightarrow \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \cos(\theta) & -\sin(\theta) \end{bmatrix}$

- Determinant of O: $det(O) = -\sin(\theta)$
  - non-zero if $\theta \neq 0, \pi, 2\pi, \cdots, i\pi$ → observable
  - 0 if $\theta = i\pi$ → not observable



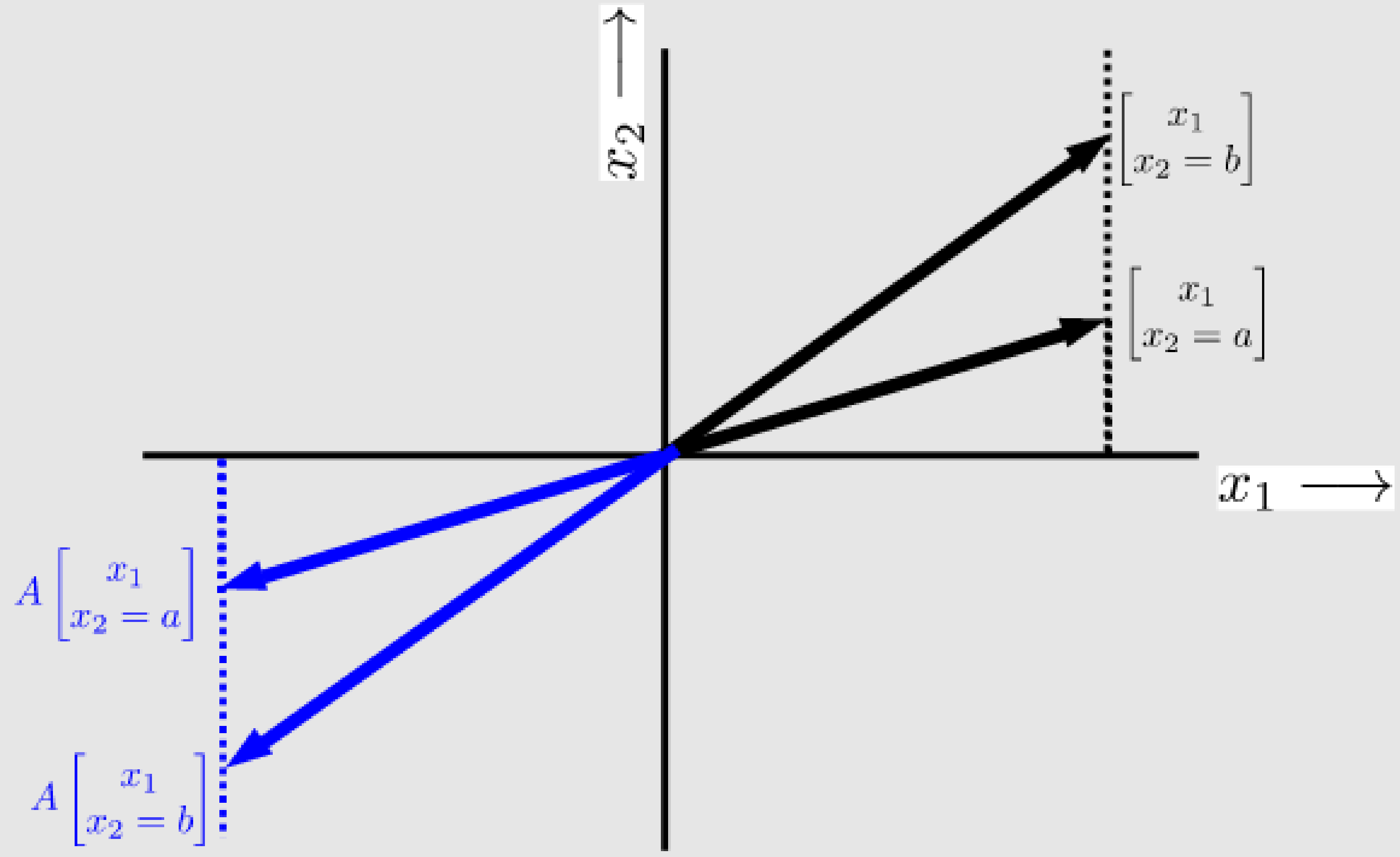
Observability: Example (contd.)

- $\begin{bmatrix} x_1[t+1] \\ x_2[t+1] \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x_1[t] \\ x_2[t] \end{bmatrix}$, $y[t] = x_1[t] = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{\vec{c}^T} \vec{x}[t]$
- Observability matrix: $O \triangleq \begin{bmatrix} \leftarrow \vec{c}^T \rightarrow \\ \leftarrow \vec{c}^T A \rightarrow \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \cos(\theta) & -\sin(\theta) \end{bmatrix}$
- Determinant of O: $det(O) = -\sin(\theta)$
  - non-zero if $\theta \neq 0, \pi, 2\pi, \cdots, i\pi$ $\rightarrow$ observable
  - 0 if $\theta = i\pi$ $\rightarrow$ not observable



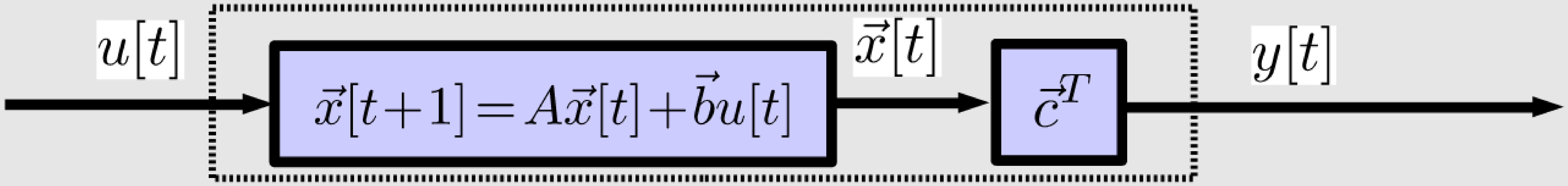
Observability: Example (contd.)

- $\begin{bmatrix} x_1[t+1] \\ x_2[t+1] \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x_1[t] \\ x_2[t] \end{bmatrix}$, $y[t] = x_1[t] = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{\vec{c}^T} \vec{x}[t]$
- Observability matrix: $O \triangleq \begin{bmatrix} \leftarrow \vec{c}^T \rightarrow \\ \leftarrow \vec{c}^T A \rightarrow \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \cos(\theta) & -\sin(\theta) \end{bmatrix}$
- Determinant of O: $det(O) = -\sin(\theta)$
  - non-zero if $\theta \neq 0, \pi, 2\pi, \cdots, i\pi$ → observable
  - 0 if $\theta = i\pi$ → not observable
    - ➔ **cannot recover x_2 uniquely**



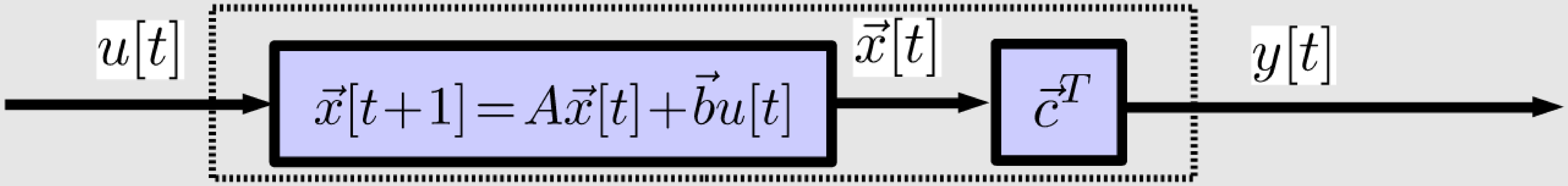
Observers

- Can we make a **system that recovers $\vec{x}[t]$ from y[t] in real time**?
  - (we can use our knowledge of A, $\vec{b}$, u[t] – and y[t])



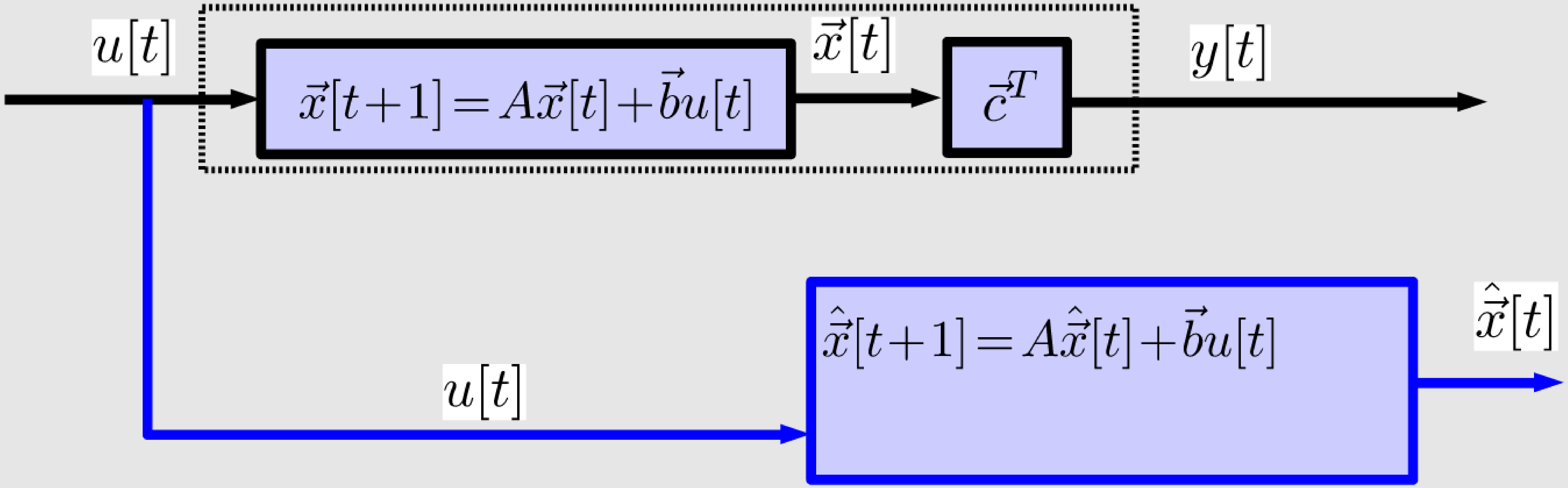
Observers

- Can we make a system that recovers $\vec{x}[t]$ from y[t] **in real time**?
  - (we can use our knowledge of A, $\vec{b}$, u[t] – and y[t])
- YES! (if the system is observable – as it will turn out)



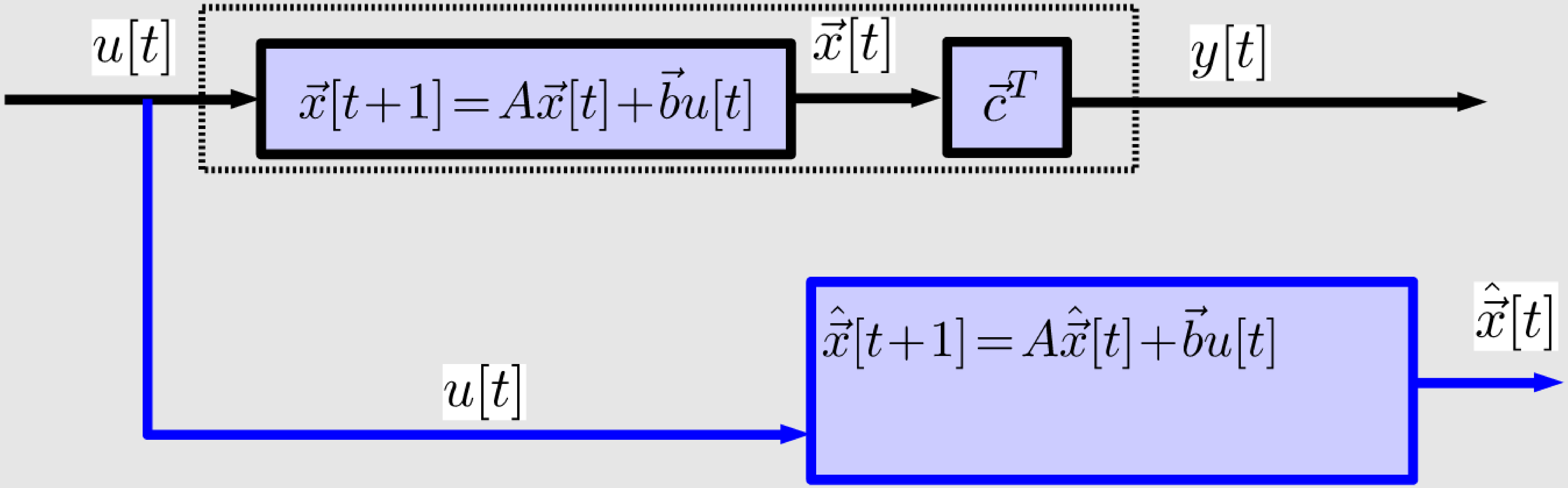
Observers

- Can we make a system that recovers $\vec{x}[t]$ from y[t] **in real time**?
  - (we can use our knowledge of A, $\vec{b}$, u[t] – and y[t])
- YES! (if the system is observable – as it will turn out)
  - first: make a clone of the system



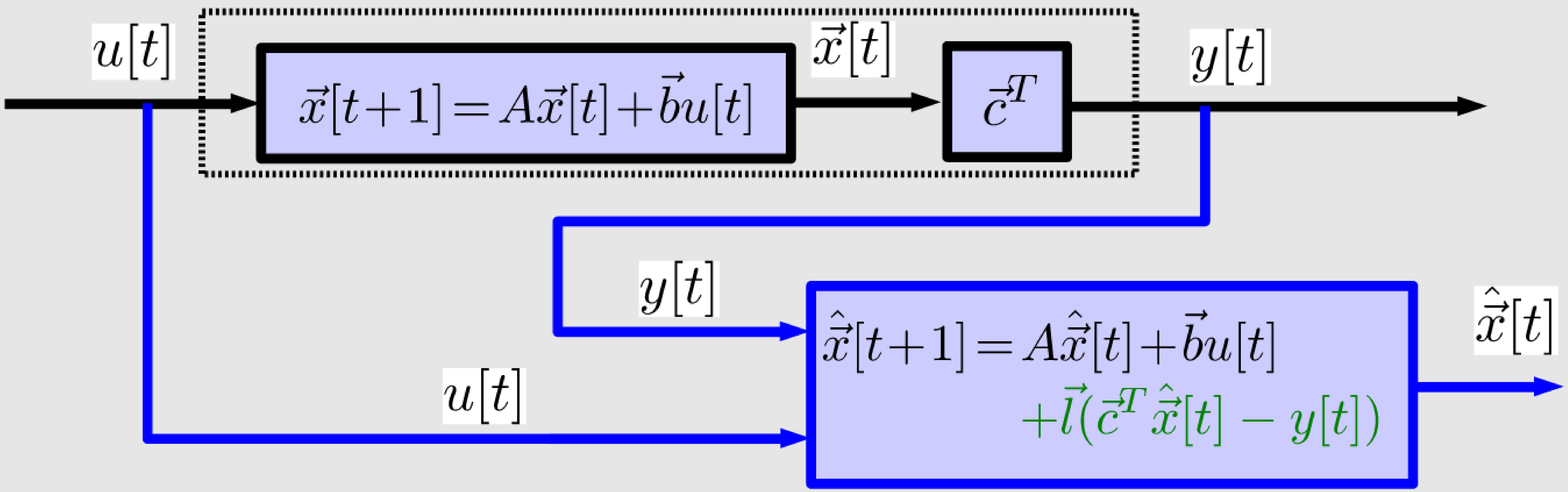
Observers

- Can we make a system that recovers $\vec{x}[t]$ from y[t] **in real time**?
  - (we can use our knowledge of A, $\vec{b}$, u[t] – and y[t])
- YES! (if the system is observable – as it will turn out)
  - first: make a clone of the system
  - next: incorporate the **difference between the outputs of the actual system and the clone**



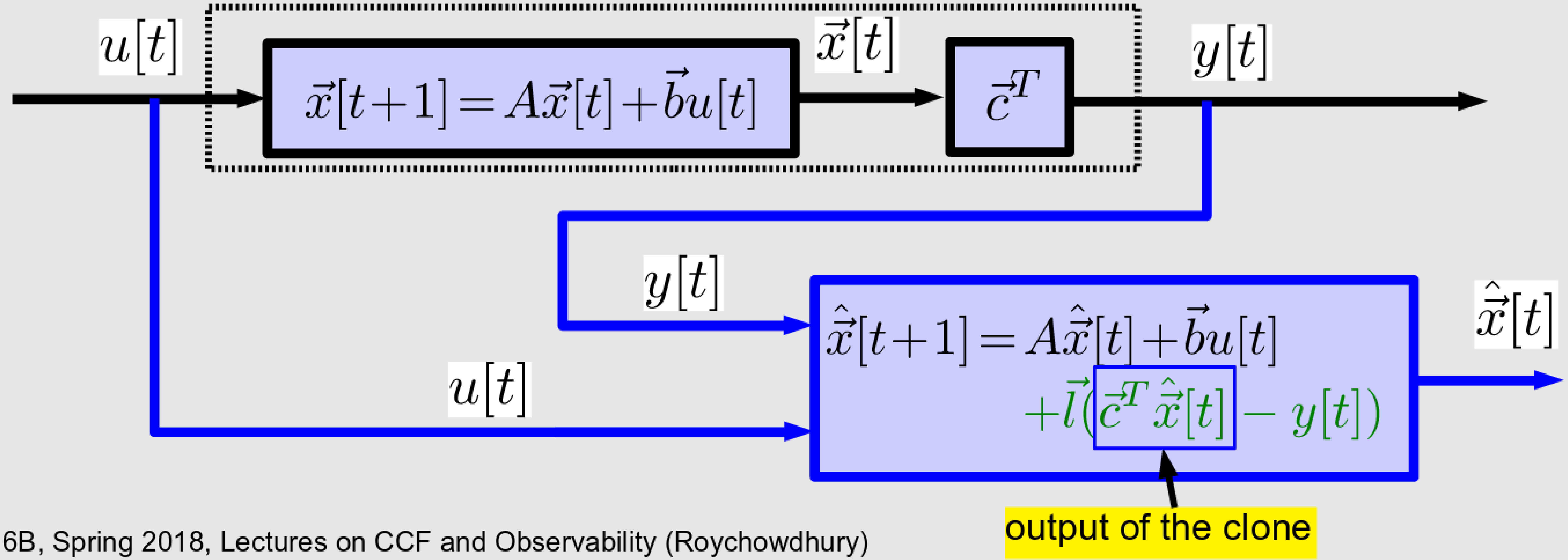
Observers

- Can we make a **system that recovers $\vec{x}[t]$ from y[t] in real time**?
  - (we can use our knowledge of A, $\vec{b}$, u[t] – and y[t])
- YES! (if the system is observable – as it will turn out)
  - first: make a clone of the system
  - next: incorporate the **difference between the outputs of the actual system and the clone**



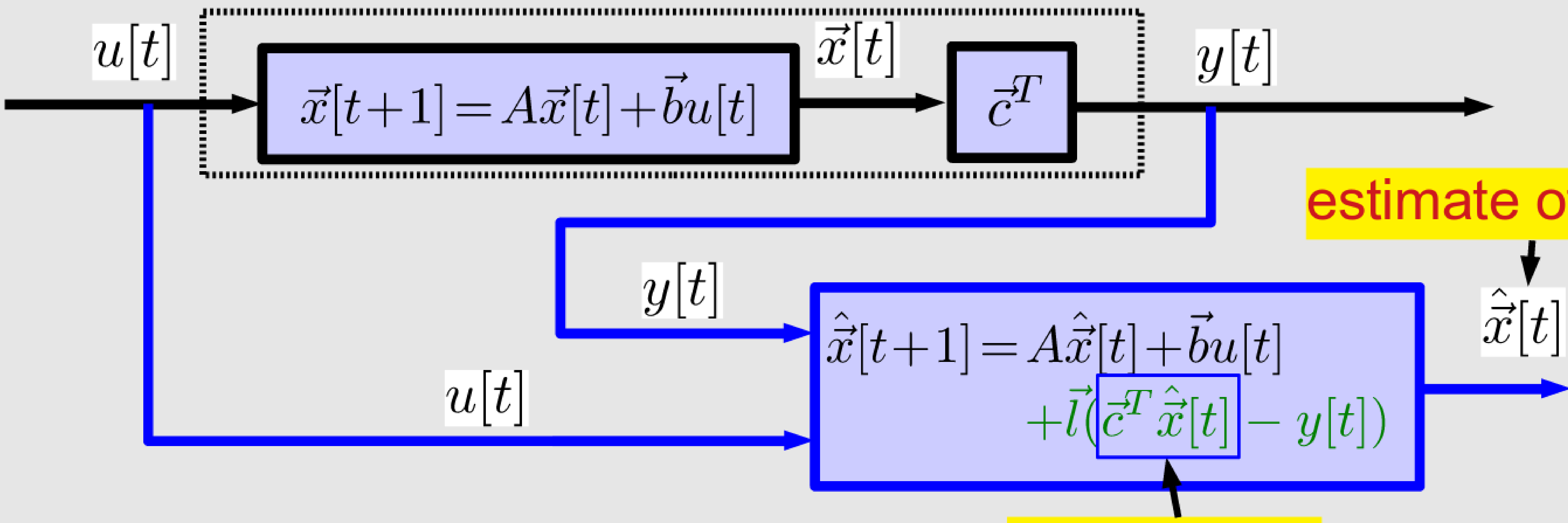
Observers

- Can we make a system that recovers $\vec{x}[t]$ from y[t] **in real time**?
  - (we can use our knowledge of A, $\vec{b}$, u[t] – and y[t])
- YES! (if the system is observable – as it will turn out)
  - first: make a clone of the system
  - next: incorporate the **difference between the outputs of the actual system and the clone**



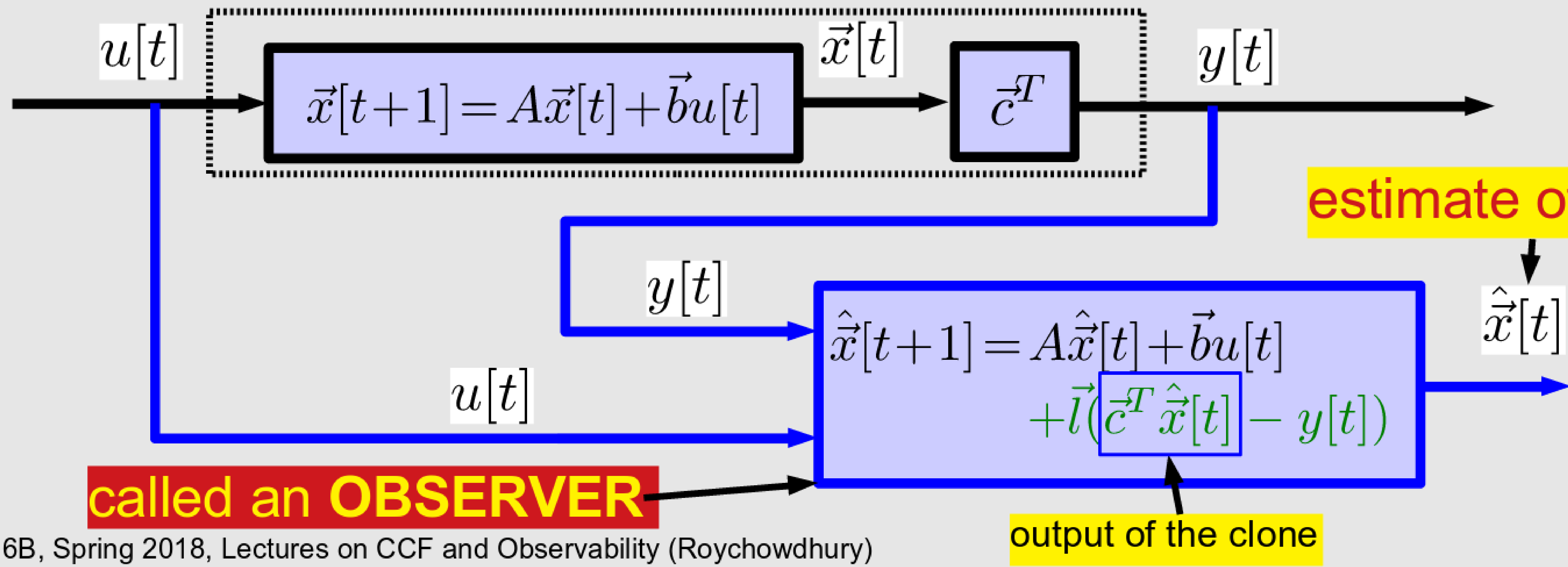
Observers

- Can we make a system that recovers $\vec{x}[t]$ from y[t] **in real time**?
  - (we can use our knowledge of A, $\vec{b}$, u[t] – and y[t])
- YES! (if the system is observable – as it will turn out)
  - first: make a clone of the system
  - next: incorporate the **difference between the outputs of the actual system and the clone**



Observers

- Can we make a **system that recovers $\vec{x}[t]$ from y[t] in real time**?
  - (we can use our knowledge of A, $\vec{b}$, u[t] – and y[t])
- YES! (if the system is observable – as it will turn out)
  - first: make a clone of the system
  - next: incorporate the **difference between the outputs of the actual system and the clone**



Observers – Why/How They Work

- Observer: $\hat{\vec{x}}[t+1] = A\hat{\vec{x}}[t] + \vec{b}u[t] + \vec{l}(\vec{c}^T\hat{\vec{x}}[t] - y[t])$

Observers – Why/How They Work

- Observer: $\hat{\vec{x}}[t+1] = A\hat{\vec{x}}[t] + \vec{b}u[t] + \vec{l}(\underbrace{\vec{c}^T\hat{\vec{x}}[t] - y[t]}_{\text{error in predicted output (\textbf{scalar})}})$

Observers – Why/How They Work

- Observer: $\hat{\vec{x}}[t+1] = A\hat{\vec{x}}[t] + \vec{b}u[t] + \underbrace{\vec{l}}_{\text{error feedback vector - TBD}}(\underbrace{\vec{c}^T \hat{\vec{x}}[t] - y[t]}_{\text{error in predicted output (scalar)}})$

Observers – Why/How They Work

- Observer: $\hat{\vec{x}}[t+1] = A\hat{\vec{x}}[t] + \vec{b}u[t] + \underbrace{\vec{l}}_{\text{error feedback vector - TBD}}(\underbrace{\vec{c}^T\hat{\vec{x}}[t] - y[t]}_{\text{error in predicted output (\textbf{scalar})}})$
- Define a state prediction error: $\vec{\epsilon}[t] \triangleq \hat{\vec{x}}[t] - \vec{x}[t]$

Observers – Why/How They Work

- Observer: $\hat{\vec{x}}[t+1] = A\hat{\vec{x}}[t] + \vec{b}u[t] + \vec{l}(\underbrace{\vec{c}^T\hat{\vec{x}}[t] - y[t]}_{\text{error in predicted output (scalar)}})$

error feedback vector - TBD

error in predicted output (**scalar**)

- Define a state prediction error: $\vec{\epsilon}[t] \triangleq \hat{\vec{x}}[t] - \vec{x}[t]$
  - then we can derive (**move to xournal**):
    - ➔ $\vec{\epsilon}[t+1] = (A + \vec{l}\,\vec{c}^T)\vec{\epsilon}[t]$

Observers – Why/How They Work

- Observer: $\hat{\vec{x}}[t+1] = A\hat{\vec{x}}[t] + \vec{b}u[t] + \underbrace{\vec{l}}_{\text{error feedback vector - TBD}}\Big(\underbrace{\vec{c}^T\,\hat{\vec{x}}[t] - y[t]}_{\text{error in predicted output (\textbf{scalar})}}\Big)$

- Define a state prediction error: $\vec{\epsilon}[t] \triangleq \hat{\vec{x}}[t] - \vec{x}[t]$
  - then we can derive (**move to xournal**):
    - ➔ $\vec{\epsilon}[t+1] = (A + \vec{l}\,\vec{c}^T)\vec{\epsilon}[t]$
  - would like $\vec{\epsilon}[t] \to 0$ as t increases (i.e., $\hat{\vec{x}}[t] \to \vec{x}[t]$)

Observers – Why/How They Work

- Observer: $\hat{\vec{x}}[t+1] = A\hat{\vec{x}}[t] + \vec{b}u[t] + \vec{l}(\underbrace{\vec{c}^T\hat{\vec{x}}[t] - y[t]}_{\text{error in predicted output (\textbf{scalar})}})$

  error feedback vector - TBD

- Define a state prediction error: $\vec{\epsilon}[t] \triangleq \hat{\vec{x}}[t] - \vec{x}[t]$
  - then we can derive (**move to xournal**):
    - $\vec{\epsilon}[t+1] = (A + \vec{l}\,\vec{c}^T)\vec{\epsilon}[t]$
  - would like $\vec{\epsilon}[t] \rightarrow 0$ as t increases (i.e., $\hat{\vec{x}}[t] \rightarrow \vec{x}[t]$)
    - **choose $\vec{l}$ to make the eigenvalues of $A + \vec{l}\,\vec{c}^T$ stable!**

Observers – Why/How They Work

- Observer: $\hat{\vec{x}}[t+1] = A\hat{\vec{x}}[t] + \vec{b}u[t] + \vec{l}\,(\underbrace{\vec{c}^T\,\hat{\vec{x}}[t] - y[t]}_{\text{error in predicted output (scalar)}})$

  error feedback vector - TBD

- Define a state prediction error: $\vec{\epsilon}[t] \triangleq \hat{\vec{x}}[t] - \vec{x}[t]$
  - then we can derive (**move to xournal**):
    - $\vec{\epsilon}[t+1] = (A + \vec{l}\,\vec{c}^T)\vec{\epsilon}[t]$
  - would like $\vec{\epsilon}[t] \to 0$ as t increases (i.e., $\hat{\vec{x}}[t] \to \vec{x}[t]$)
    - **choose $\vec{l}$ to make the eigenvalues of $A + \vec{l}\,\vec{c}^T$ stable!**
  - **strong analogy w controllability (recall $A - \vec{b}\,\vec{k}^T$)**

Observers – Why/How They Work

- Observer: $\hat{\vec{x}}[t+1] = A\hat{\vec{x}}[t] + \vec{b}u[t] + \vec{l}(\underbrace{\vec{c}^T\hat{\vec{x}}[t] - y[t]}_{\text{error in predicted output (scalar)}})$

  error feedback vector - TBD

- Define a state prediction error: $\vec{\epsilon}[t] \triangleq \hat{\vec{x}}[t] - \vec{x}[t]$
  - then we can derive (**move to xournal**):
    - $\vec{\epsilon}[t+1] = (A + \vec{l}\,\vec{c}^T)\vec{\epsilon}[t]$
  - would like $\vec{\epsilon}[t] \to 0$ as t increases (i.e., $\hat{\vec{x}}[t] \to \vec{x}[t]$)
    - **choose $\vec{l}$ to make the eigenvalues of $A + \vec{l}\,\vec{c}^T$ stable!**
  - **strong analogy w controllability (recall $A - \vec{b}\,\vec{k}^T$)**
    - evs of $A + \vec{l}\,\vec{c}^T$ = evs of $A^T + \vec{c}\,\vec{l}^T \rightarrow$ $-\vec{c} \mapsto \vec{b}, \quad \vec{l}^T \mapsto \vec{k}^T$

Observers – Why/How They Work

- Observer: $\hat{\vec{x}}[t+1] = A\hat{\vec{x}}[t] + \vec{b}u[t] + \vec{l}\underbrace{(\vec{c}^T\hat{\vec{x}}[t] - y[t])}_{\text{error in predicted output (scalar)}}$

  error feedback vector - TBD

  error in predicted output (**scalar**)

- Define a state prediction error: $\vec{\epsilon}[t] \triangleq \hat{\vec{x}}[t] - \vec{x}[t]$
  - then we can derive (**move to xournal**):
    - $\vec{\epsilon}[t+1] = (A + \vec{l}\,\vec{c}^T)\vec{\epsilon}[t]$
  - would like $\vec{\epsilon}[t] \to 0$ as t increases (i.e., $\hat{\vec{x}}[t] \to \vec{x}[t]$)
    - **choose $\vec{l}$ to make the eigenvalues of $A + \vec{l}\,\vec{c}^T$ stable!**
  - **strong analogy w controllability (recall $A - \vec{b}\,\vec{k}^T$)**
    - evs of $A + \vec{l}\,\vec{c}^T$ = evs of $A^T + \vec{c}\,\vec{l}^T \to$ $-\vec{c} \mapsto \vec{b}, \quad \vec{l}^T \mapsto \vec{k}^T$
- **i.e., can always make $A + \vec{l}\,\vec{c}^T$ stable if $(A^T, -\vec{c})$ is controllable** **(using previous controllability + feedback result)**

Observers – Why/How (contd.)

- $(A^T, -\vec{c})$ controllable $\rightarrow$ $-\left[\vec{c} \,|\, A^T\vec{c} \,|\, \cdots \,|\, (A^T)^{n-2}\vec{c} \,|\, (A^T)^{n-1}\vec{c}\right]$ must be full rank

Observers – Why/How (contd.)

- $(A^T, -\vec{c})$ controllable $\rightarrow -\left[\vec{c} \,|\, A^T\vec{c} \,|\, \cdots \,|\, (A^T)^{n-2}\vec{c} \,|\, (A^T)^{n-1}\vec{c}\right]$ must be full rank
  - $\rightarrow \left[\vec{c} \,|\, A^T\vec{c} \,|\, \cdots \,|\, (A^T)^{n-2}\vec{c} \,|\, (A^T)^{n-1}\vec{c}\right]^T$ must be full rank

Observers – Why/How (contd.)

- $(A^T, -\vec{c})$ controllable $\rightarrow -\left[\vec{c}\,|\, A^T\vec{c}\,|\, \cdots \,|\, (A^T)^{n-2}\vec{c}\,|\, (A^T)^{n-1}\vec{c}\right]$ must be full rank
  - $\rightarrow \left[\vec{c}\,|\, A^T\vec{c}\,|\, \cdots \,|\, (A^T)^{n-2}\vec{c}\,|\, (A^T)^{n-1}\vec{c}\right]^T$ must be full rank
  - $\rightarrow \begin{bmatrix} \longleftarrow & \vec{c}^T & \longrightarrow \\ \longleftarrow & \vec{c}^T A & \longrightarrow \\ \longleftarrow & \vec{c}^T A^2 & \longrightarrow \\ & \vdots & \\ \longleftarrow & \vec{c}^T A^{n-1} & \longrightarrow \end{bmatrix}$ must be full rank

Observers – Why/How (contd.)

- $(A^T, -\vec{c})$ controllable $\rightarrow -\left[\vec{c} \,|\, A^T\vec{c} \,|\, \cdots \,|\, (A^T)^{n-2}\vec{c} \,|\, (A^T)^{n-1}\vec{c}\right]$ must be full rank
  - $\rightarrow \left[\vec{c} \,|\, A^T\vec{c} \,|\, \cdots \,|\, (A^T)^{n-2}\vec{c} \,|\, (A^T)^{n-1}\vec{c}\right]^T$ must be full rank
  - $\rightarrow \begin{bmatrix} \longleftarrow & \vec{c}^T & \longrightarrow \\ \longleftarrow & \vec{c}^T A & \longrightarrow \\ \longleftarrow & \vec{c}^T A^2 & \longrightarrow \\ & \vdots & \\ \longleftarrow & \vec{c}^T A^{n-1} & \longrightarrow \end{bmatrix}$ must be full rank

just the OBSERVABILITY matrix

Observers – Why/How (contd.)

- $(A^T, -\vec{c})$ controllable $\rightarrow$ $-\left[\vec{c}\,|\,A^T\vec{c}\,|\,\cdots\,|\,(A^T)^{n-2}\vec{c}\,|\,(A^T)^{n-1}\vec{c}\right]$ must be full rank
  - $\rightarrow \left[\vec{c}\,|\,A^T\vec{c}\,|\,\cdots\,|\,(A^T)^{n-2}\vec{c}\,|\,(A^T)^{n-1}\vec{c}\right]^T$ must be full rank
  - $\rightarrow \begin{bmatrix} \longleftarrow & \vec{c}^T & \longrightarrow \\ \longleftarrow & \vec{c}^T A & \longrightarrow \\ \longleftarrow & \vec{c}^T A^2 & \longrightarrow \\ & \vdots & \\ \longleftarrow & \vec{c}^T A^{n-1} & \longrightarrow \end{bmatrix}$ must be full rank

just the **OBSERVABILITY matrix**

- **Conclusion: if a system is observable, we can build an observer for it whose estimate $\hat{\vec{x}}[t]$ will approximate $\vec{x}[t]$ more and more closely with t**

Observer: Rotation Matrix Example

- $\begin{bmatrix} x_1[t+1] \\ x_2[t+1] \end{bmatrix} = \underbrace{\begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}}_{A} \begin{bmatrix} x_1[t] \\ x_2[t] \end{bmatrix}$, $y[t] = x_1[t] = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{\vec{c}^T} \vec{x}[t]$

Observer: Rotation Matrix Example

- $\begin{bmatrix} x_1[t+1] \\ x_2[t+1] \end{bmatrix} = \underbrace{\begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}}_{A} \begin{bmatrix} x_1[t] \\ x_2[t] \end{bmatrix}$, $y[t] = x_1[t] = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{\vec{c}^T} \vec{x}[t]$
- example: $\theta = \dfrac{\pi}{2} \longrightarrow A = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \longrightarrow O = \begin{bmatrix} \leftarrow \vec{c}^T \rightarrow \\ \leftarrow \vec{c}^T A \rightarrow \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$

Observer: Rotation Matrix Example

- $$\begin{bmatrix} x_1[t+1] \\ x_2[t+1] \end{bmatrix} = \underbrace{\begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}}_{A} \begin{bmatrix} x_1[t] \\ x_2[t] \end{bmatrix}, \quad y[t] = x_1[t] = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{\vec{c}^T} \vec{x}[t]$$
- example: $\theta = \frac{\pi}{2} \longrightarrow A = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \longrightarrow O = \begin{bmatrix} \leftarrow \vec{c}^T \rightarrow \\ \leftarrow \vec{c}^T A \rightarrow \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$

full rank

Observer: Rotation Matrix Example

- $$\begin{bmatrix} x_1[t+1] \\ x_2[t+1] \end{bmatrix} = \underbrace{\begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}}_{A} \begin{bmatrix} x_1[t] \\ x_2[t] \end{bmatrix}, \quad y[t] = x_1[t] = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{\vec{c}^T} \vec{x}[t]$$
- example: $\theta = \frac{\pi}{2} \longrightarrow A = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \longrightarrow O = \begin{bmatrix} \leftarrow \vec{c}^T \rightarrow \\ \leftarrow \vec{c}^T A \rightarrow \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$
  - side note: eigenvalues of A: $\pm \jmath \longrightarrow$ BIBO unstable

full rank

Observer: Rotation Matrix Example

- $\begin{bmatrix} x_1[t+1] \\ x_2[t+1] \end{bmatrix} = \underbrace{\begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}}_{A} \begin{bmatrix} x_1[t] \\ x_2[t] \end{bmatrix}$, $y[t] = x_1[t] = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{\vec{c}^T} \vec{x}[t]$
- example: $\theta = \frac{\pi}{2} \longrightarrow A = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \longrightarrow O = \begin{bmatrix} \leftarrow \vec{c}^T \rightarrow \\ \leftarrow \vec{c}^T A \rightarrow \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$
  - side note: eigenvalues of A: $\pm \jmath \longrightarrow$ BIBO unstable

    full rank
  - let $\vec{l} = \begin{bmatrix} l_1 \\ l_2 \end{bmatrix}$, then $A + \vec{l}\vec{c}^T = \begin{bmatrix} l_1 & -1 \\ 1 + l_2 & 0 \end{bmatrix}$

Observer: Rotation Matrix Example

- $\begin{bmatrix} x_1[t+1] \\ x_2[t+1] \end{bmatrix} = \underbrace{\begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}}_{A} \begin{bmatrix} x_1[t] \\ x_2[t] \end{bmatrix}$, $y[t] = x_1[t] = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{\vec{c}^T} \vec{x}[t]$
- example: $\theta = \frac{\pi}{2} \longrightarrow A = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \longrightarrow O = \begin{bmatrix} \leftarrow \vec{c}^T \rightarrow \\ \leftarrow \vec{c}^T A \rightarrow \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$
  - side note: eigenvalues of A: $\pm \jmath \longrightarrow$ BIBO unstable

    full rank
  - let $\vec{l} = \begin{bmatrix} l_1 \\ l_2 \end{bmatrix}$, then $A + \vec{l}\vec{c}^T = \begin{bmatrix} l_1 & -1 \\ 1 + l_2 & 0 \end{bmatrix}$
    - ➔ eigenvalues (see the notes): $\lambda_{1,2} = \frac{l_1}{2} \pm \frac{l_1^2 - 4(1+l_2)}{2}$

Observer: Rotation Matrix Example

- $\begin{bmatrix} x_1[t+1] \\ x_2[t+1] \end{bmatrix} = \underbrace{\begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}}_{A} \begin{bmatrix} x_1[t] \\ x_2[t] \end{bmatrix}$, $y[t] = x_1[t] = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{\vec{c}^T} \vec{x}[t]$
- example: $\theta = \frac{\pi}{2} \longrightarrow A = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \longrightarrow O = \begin{bmatrix} \leftarrow \vec{c}^T \rightarrow \\ \leftarrow \vec{c}^T A \rightarrow \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$
  - side note: eigenvalues of A: $\pm\jmath \longrightarrow$ BIBO unstable

    full rank
  - let $\vec{l} = \begin{bmatrix} l_1 \\ l_2 \end{bmatrix}$, then $A + \vec{l}\vec{c}^T = \begin{bmatrix} l_1 & -1 \\ 1 + l_2 & 0 \end{bmatrix}$
    - ➔ eigenvalues (see the notes): $\lambda_{1,2} = \frac{l_1}{2} \pm \frac{l_1^2 - 4(1 + l_2)}{2}$
      - and can easily show: $l_1 = \lambda_1 + \lambda_2, \quad l_2 = \lambda_1 \lambda_2 - 1$

Observer: Rotation Matrix Example

- $\begin{bmatrix} x_1[t+1] \\ x_2[t+1] \end{bmatrix} = \underbrace{\begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}}_{A} \begin{bmatrix} x_1[t] \\ x_2[t] \end{bmatrix}$, $y[t] = x_1[t] = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{\vec{c}^T} \vec{x}[t]$
- example: $\theta = \frac{\pi}{2} \longrightarrow A = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \longrightarrow O = \begin{bmatrix} \leftarrow \vec{c}^T \rightarrow \\ \leftarrow \vec{c}^T A \rightarrow \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$ full rank
  - side note: eigenvalues of A: $\pm \jmath \longrightarrow$ BIBO unstable
  - let $\vec{l} = \begin{bmatrix} l_1 \\ l_2 \end{bmatrix}$, then $A + \vec{l}\vec{c}^T = \begin{bmatrix} l_1 & -1 \\ 1 + l_2 & 0 \end{bmatrix}$
    - ➔ eigenvalues (see the notes): $\lambda_{1,2} = \frac{l_1}{2} \pm \frac{l_1^2 - 4(1 + l_2)}{2}$
      - and can easily show: $l_1 = \lambda_1 + \lambda_2, \quad l_2 = \lambda_1 \lambda_2 - 1$
- i.e., can set $\vec{l}$ to obtain any desired eigenvalues

Observer: Rotation Matrix Example

- $\begin{bmatrix} x_1[t+1] \\ x_2[t+1] \end{bmatrix} = \underbrace{\begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}}_{A} \begin{bmatrix} x_1[t] \\ x_2[t] \end{bmatrix}$, $y[t] = x_1[t] = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{\vec{c}^T} \vec{x}[t]$
- example: $\theta = \frac{\pi}{2} \rightarrow A = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \rightarrow O = \begin{bmatrix} \leftarrow \vec{c}^T \rightarrow \\ \leftarrow \vec{c}^T A \rightarrow \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$
  - side note: eigenvalues of A: $\pm \jmath \rightarrow$ BIBO unstable

    full rank
  - let $\vec{l} = \begin{bmatrix} l_1 \\ l_2 \end{bmatrix}$, then $A + \vec{l}\vec{c}^T = \begin{bmatrix} l_1 & -1 \\ 1 + l_2 & 0 \end{bmatrix}$
    - eigenvalues (see the notes): $\lambda_{1,2} = \frac{l_1}{2} \pm \frac{l_1^2 - 4(1+l_2)}{2}$
      - and can easily show: $l_1 = \lambda_1 + \lambda_2, \quad l_2 = \lambda_1 \lambda_2 - 1$
- i.e., can set $\vec{l}$ to obtain any desired eigenvalues
  - warning: if complex, **ensure evs are complex conjugates**
    - **what will happen if you don't?**

Observer: Rot. Matrix Example (contd.)

- $\begin{bmatrix} x_1[t+1] \\ x_2[t+1] \end{bmatrix} = \underbrace{\begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}}_{A} \begin{bmatrix} x_1[t] \\ x_2[t] \end{bmatrix}$, $y[t] = x_1[t] = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{\vec{c}^T} \vec{x}[t]$

Observer: Rot. Matrix Example (contd.)

- $$\begin{bmatrix} x_1[t+1] \\ x_2[t+1] \end{bmatrix} = \underbrace{\begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}}_{A} \begin{bmatrix} x_1[t] \\ x_2[t] \end{bmatrix}, \quad y[t] = x_1[t] = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{\vec{c}^T} \vec{x}[t]$$
- now try: $\theta = \pi \longrightarrow A = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \longrightarrow$ not observable (recall)

Observer: Rot. Matrix Example (contd.)

- $$\begin{bmatrix} x_1[t+1] \\ x_2[t+1] \end{bmatrix} = \underbrace{\begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}}_{A} \begin{bmatrix} x_1[t] \\ x_2[t] \end{bmatrix}, \quad y[t] = x_1[t] = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{\vec{c}^T} \vec{x}[t]$$
- now try: $\theta = \pi$ $\longrightarrow A = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \longrightarrow$ not observable (recall)
  - $$A + \vec{l}\vec{c}^T = \begin{bmatrix} -1 + l_1 & 0 \\ l_2 & -1 \end{bmatrix}$$
    - eigenvalues (see the notes): $\lambda_1 = -1, \quad \lambda_2 = l_1 - 1$

Observer: Rot. Matrix Example (contd.)

- $\begin{bmatrix} x_1[t+1] \\ x_2[t+1] \end{bmatrix} = \underbrace{\begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}}_{A} \begin{bmatrix} x_1[t] \\ x_2[t] \end{bmatrix}$, $y[t] = x_1[t] = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{\vec{c}^T} \vec{x}[t]$
- now try: $\theta = \pi \longrightarrow A = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \longrightarrow$ not observable (recall)
- $A + \vec{l}\vec{c}^T = \begin{bmatrix} -1 + l_1 & 0 \\ l_2 & -1 \end{bmatrix}$
  - eigenvalues (see the notes): $\boxed{\lambda_1 = -1,}$ $\lambda_2 = l_1 - 1$

cannot be changed/stabilized using $\vec{l}$

Observability: The Continuous Case

- Observability for C.T. state-space systems
  - and implications for placing observer eigenvalues
- **EXACTLY THE SAME CRITERIA**
  - $$\begin{bmatrix} \longleftarrow & \vec{c}^T & \longrightarrow \\ \longleftarrow & \vec{c}^T A & \longrightarrow \\ \longleftarrow & \vec{c}^T A^2 & \longrightarrow \\ & \vdots & \\ \longleftarrow & \vec{c}^T A^{n-1} & \longrightarrow \end{bmatrix}$$ **must be full rank**
- Stability for C.T. means Re(eigenvalues) < 0

Observers: Accurate Positioning

- Physical motion is inherently marginally stable
  - due to the relationship between position, velocity and acceleration
    - ➔ $\dot{x} = v, \quad \dot{v} = a$

Observers: Accurate Positioning

- Physical motion is inherently marginally stable
  - due to the relationship between position, velocity and acceleration
    - ➔ $\dot{x} = v, \quad \dot{v} = a$
      - small error in a $\rightarrow$ growing error in v
      - small error in v $\rightarrow$ growing error in x

Observers: Accurate Positioning

- Physical motion is inherently marginally stable
  - due to the relationship between position, velocity and acceleration
    - ➔ $\dot{x} = v, \quad \dot{v} = a$
      - small error in a $\rightarrow$ growing error in v
      - small error in v $\rightarrow$ growing error in x
- You are in a car in a featureless desert

Observers: Accurate Positioning

- Physical motion is inherently marginally stable
  - due to the relationship between position, velocity and acceleration
    - ➔ $\dot{x} = v, \quad \dot{v} = a$
      - small error in a $\rightarrow$ growing error in v
      - small error in v $\rightarrow$ growing error in x
- You are in a car in a featureless desert
  - you know the position where you started

Observers: Accurate Positioning

- Physical motion is inherently marginally stable
  - due to the relationship between position, velocity and acceleration
    - ➔ $\dot{x} = v, \quad \dot{v} = a$
      - small error in a $\rightarrow$ growing error in v
      - small error in v $\rightarrow$ growing error in x
- You are in a car in a featureless desert
  - you know the position where you started
  - you record your acceleration (along x and y directions)

Observers: Accurate Positioning

- Physical motion is inherently marginally stable
  - due to the relationship between position, velocity and acceleration
    - ➔ $\dot{x} = v, \quad \dot{v} = a$
      - small error in a $\rightarrow$ growing error in v
      - small error in v $\rightarrow$ growing error in x
- You are in a car in a featureless desert
  - you know the position where you started
  - you record your acceleration (along x and y directions)
  - to estimate your current position
    - ➔ you integrate accel./velocity to predict your current position

Observers: Accurate Positioning

- Physical motion is inherently marginally stable
  - due to the relationship between position, velocity and acceleration
    - ➔ $\dot{x} = v, \quad \dot{v} = a$
      - small error in a $\longrightarrow$ growing error in v
      - small error in v $\longrightarrow$ growing error in x
- You are in a car in a featureless desert
  - you know the position where you started
  - you record your acceleration (along x and y directions)
  - to estimate your current position
    - ➔ you integrate accel./velocity to predict your current position
    - ➔ but inevitable small errors (eg, play in accelerator) make your predicted position more and more inaccurate (m. stability)
      - soon, your prediction becomes completely useless – miles from where you really are

Observers: Accurate Positioning

- Physical motion is inherently marginally stable
  - due to the relationship between position, velocity and acceleration
    - ➔ $\dot{x} = v, \quad \dot{v} = a$
      - small error in a → growing error in v
      - small error in v → growing error in x
- You are in a car in a featureless desert
  - you know the position where you started
  - you record your acceleration (along x and y directions)
  - to estimate your current position
    - ➔ you integrate accel./velocity to predict your current position
    - ➔ but inevitable small errors (eg, play in accelerator) make your predicted position more and more inaccurate (m. stability)
      - soon, your prediction becomes completely useless – miles from where you really are
  - **NOT A VERY PRACTICALLY USEFUL WAY TO LOCATE YOURSELF**

Observers for Positioning (contd.)

- Enter GPS
  - you have a GPS receiver and position calculator

Observers for Positioning (contd.)

- Enter GPS
  - you have a GPS receiver and position calculator
    - ➔ but GPS isn't perfectly accurate either (though much better than our integration technique, aka "dead reckoning")
      - can easily be a few 10s of feet off

Observers for Positioning (contd.)

- Enter GPS
  - you have a GPS receiver and position calculator
    - ➔ but GPS isn't perfectly accurate either (though much better than our integration technique, aka "dead reckoning")
      - can easily be a few 10s of feet off
- Can we combine dead reckoning and GPS
  - for better accuracy than GPS alone?

Observers for Positioning (contd.)

- Enter GPS
  - you have a GPS receiver and position calculator
    - ➔ but GPS isn't perfectly accurate either (though much better than our integration technique, aka "dead reckoning")
      - can easily be a few 10s of feet off
- Can we combine dead reckoning and GPS
  - for better accuracy than GPS alone?
- **YES**: feed GPS position data into an **observer**!

Observers for Positioning (contd.)

- Enter GPS
  - you have a GPS receiver and position calculator
    - ➔ but GPS isn't perfectly accurate either (though much better than our integration technique, aka "dead reckoning")
      - can easily be a few 10s of feet off
- Can we combine dead reckoning and GPS
  - for better accuracy than GPS alone?
- **YES**: feed GPS position data into an **observer**!
  - stabilize the observer by choosing $\vec{l}$ wisely

Observers for Positioning (contd.)

- Enter GPS
  - you have a GPS receiver and position calculator
    - but GPS isn't perfectly accurate either (though much better than our integration technique, aka "dead reckoning")
      - can easily be a few 10s of feet off
- Can we combine dead reckoning and GPS
  - for better accuracy than GPS alone?
- **YES**: feed GPS position data into an **observer**!
  - stabilize the observer by choosing $\vec{l}$ wisely
  - even with perpetual small GPS and acceleration errors
    - **the observer's estimate is far better than just the GPS alone!***

Observers for Positioning (contd.)

* see the notes for the math

- Enter GPS
  - you have a GPS receiver and position calculator
    - ➔ but GPS isn't perfectly accurate either (though much better than our integration technique, aka "dead reckoning")
      - can easily be a few 10s of feet off
- Can we combine dead reckoning and GPS
  - for better accuracy than GPS alone?
- **YES**: feed GPS position data into an **observer**!
  - stabilize the observer by choosing $\vec{l}$ wisely
  - even with perpetual small GPS and acceleration errors
    - ➔ **the observer's estimate is far better than just the GPS alone!***

Observers for Positioning (contd.)

* see the notes for the math

- Enter GPS
  - you have a GPS receiver and position calculator
    - but GPS isn't perfectly accurate either (though much better than our integration technique, aka "dead reckoning")
      - can easily be a few 10s of feet off
- Can we combine dead reckoning and GPS
  - for better accuracy than GPS alone?
- **YES**: feed GPS position data into an **observer**!
  - stabilize the observer by choosing $\vec{l}$ wisely
  - even with perpetual small GPS and acceleration errors
    - **the observer's estimate is far better than just the GPS alone!***
- This is what all serious navigational systems use

Observers for Positioning (contd.)

* see the notes for the math

- Enter GPS
  - you have a GPS receiver and position calculator
    - → but GPS isn't perfectly accurate either (though much better than our integration technique, aka "dead reckoning")
      - can easily be a few 10s of feet off
- Can we combine dead reckoning and GPS
  - for better accuracy than GPS alone?
- **YES**: feed GPS position data into an **observer**!
  - stabilize the observer by choosing $\vec{l}$ wisely
  - even with perpetual small GPS and acceleration errors
    - → **the observer's estimate is far better than just the GPS alone!***
- This is what all serious navigational systems use
  - with an additional twist: $\vec{l}$ keeps updating, becomes $\vec{l}[t]$

Observers for Positioning (contd.)

* see the notes for the math

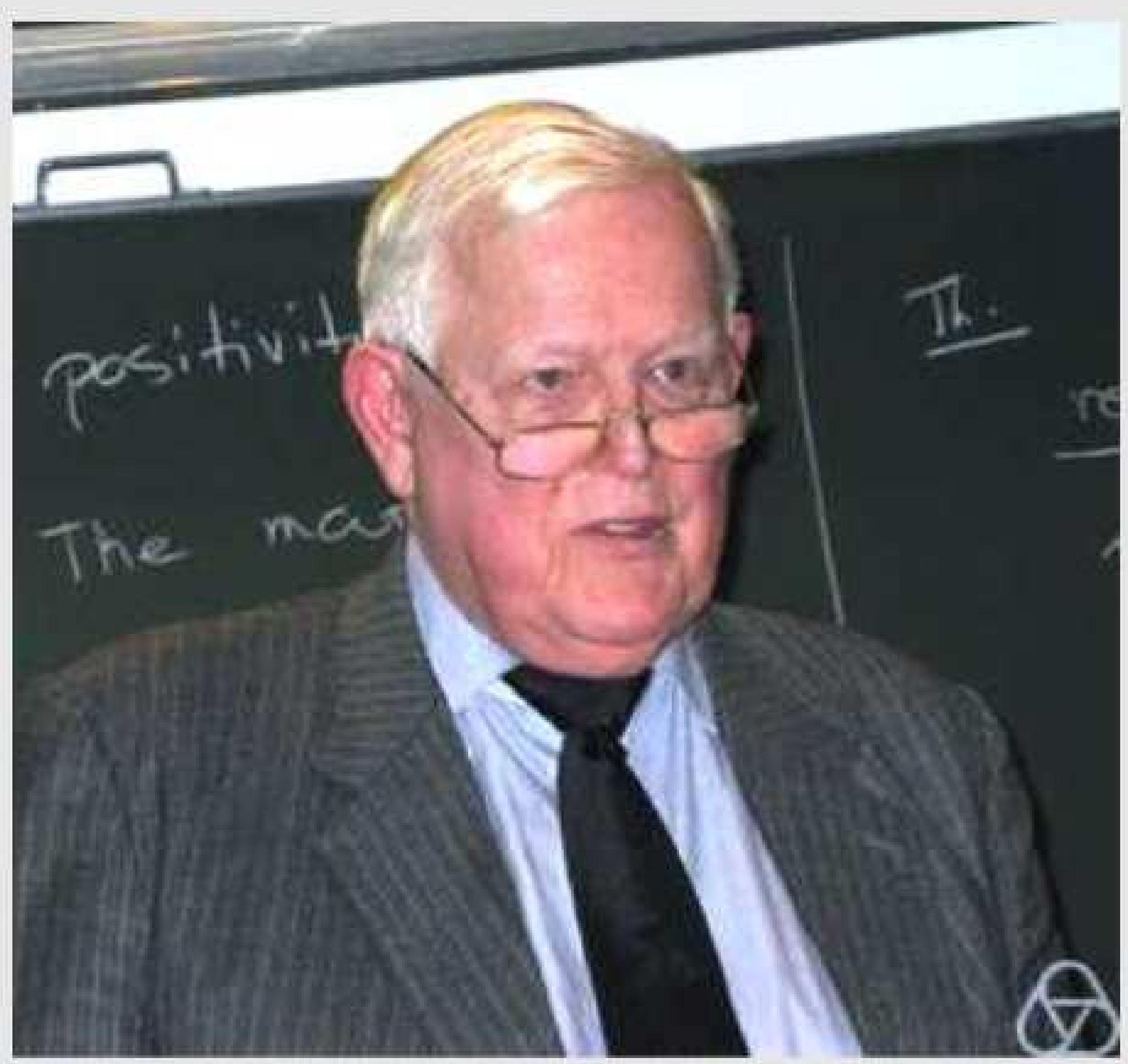
- Enter GPS
  - you have a GPS receiver and position calculator
    - but GPS isn't perfectly accurate either (though much better than our integration technique, aka "dead reckoning")
      - can easily be a few 10s of feet off
- Can we combine dead reckoning and GPS
  - for better accuracy than GPS alone?
- **YES**: feed GPS position data into an **observer**!
  - stabilize the observer by choosing $\vec{l}$ wisely
  - even with perpetual small GPS and acceleration errors
    - **the observer's estimate is far better than just the GPS alone!***
- This is what all serious navigational systems use
  - with an additional twist: $\vec{l}$ keeps updating, becomes $\vec{l}[t]$
  - **this is the famous KALMAN FILTER**
    - **used in all rockets, drones, autonomous cars, ships, ...**

Rudolf Kálmán
"inventor" of control theory: 1950s/60s

- state-space representations
- stability, controllability, observability and implications
- Kalman filter

Rudolf Kálmán
"inventor" of control theory: 1950s/60s

- state-space representations
- stability, controllability, observability and implications
- Kalman filter
  - initially received with "vast skepticism" - not accepted for publication!


Rudolf Kálmán
"inventor" of control theory: 1950s/60s

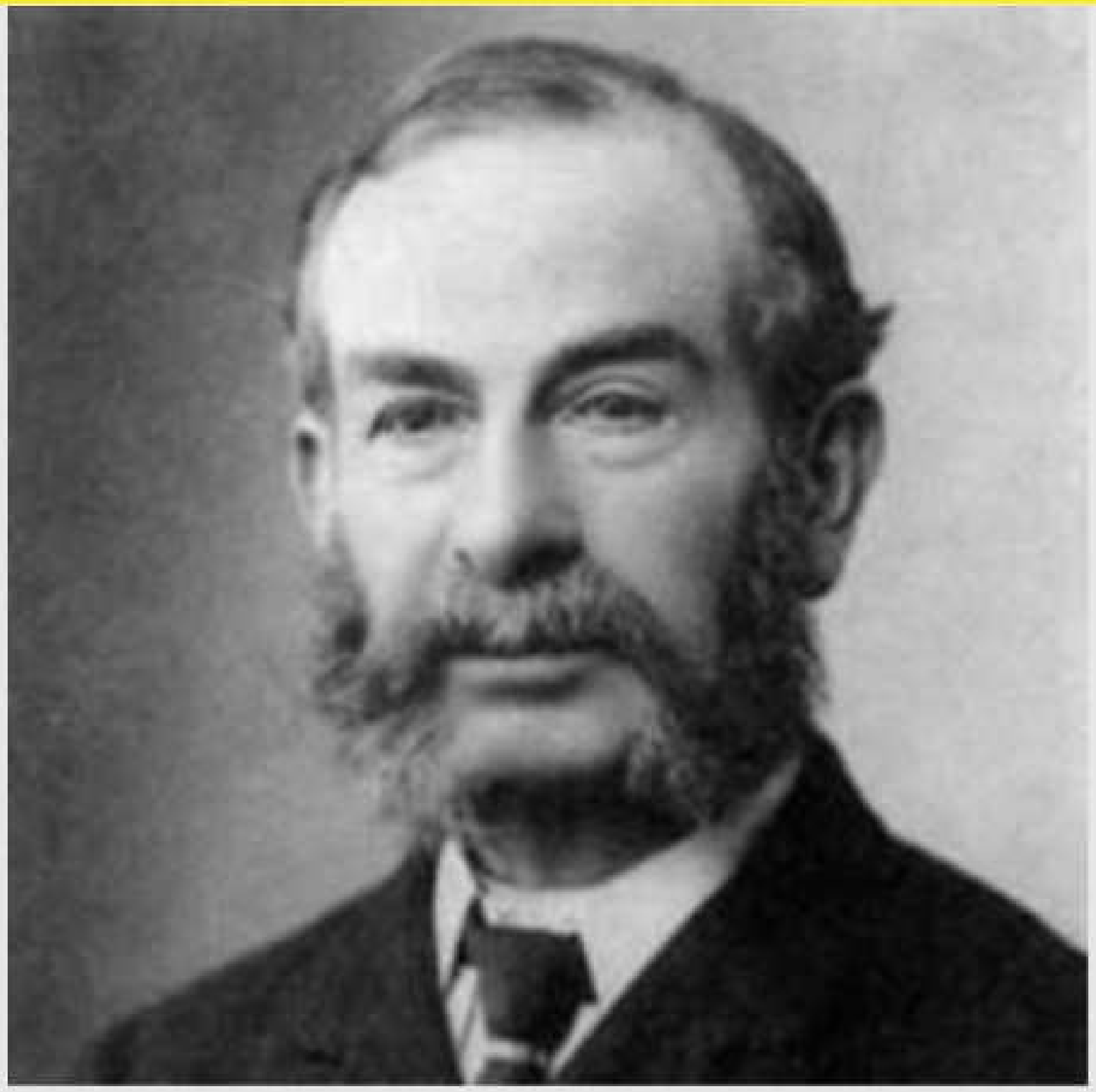
- state-space representations
- stability, controllability, observability and implications
- Kalman filter
  - initially received with "vast skepticism" - not accepted for publication!
  - later adopted by the Apollo rocket program, the Space Shuttle, submarines, cruise missiles, UAVs/drones, autonomous vehicles, ...



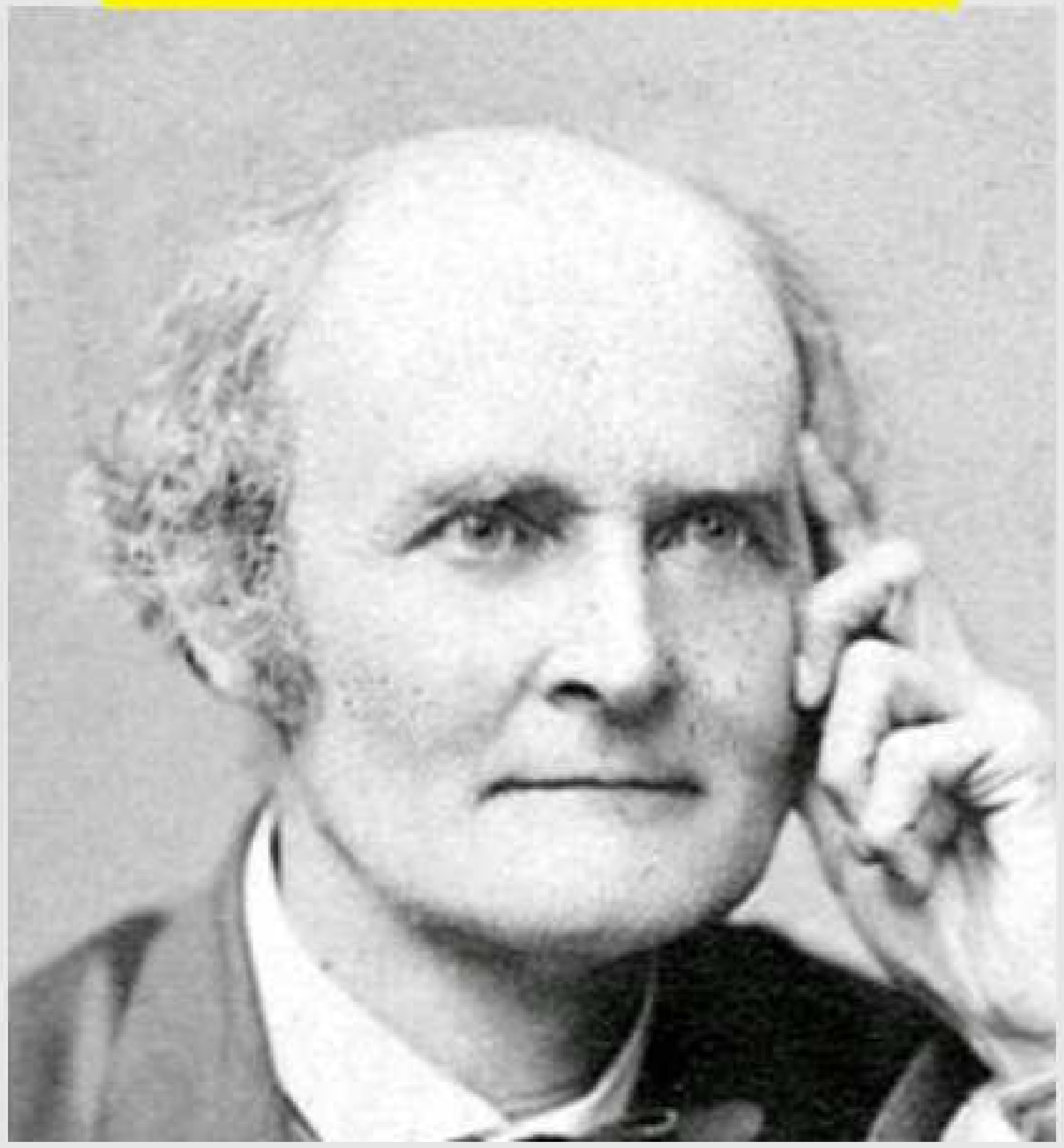
Who Invented Eigendecomposition?

1852 - 1858

James Joseph Sylvester (1814-97)



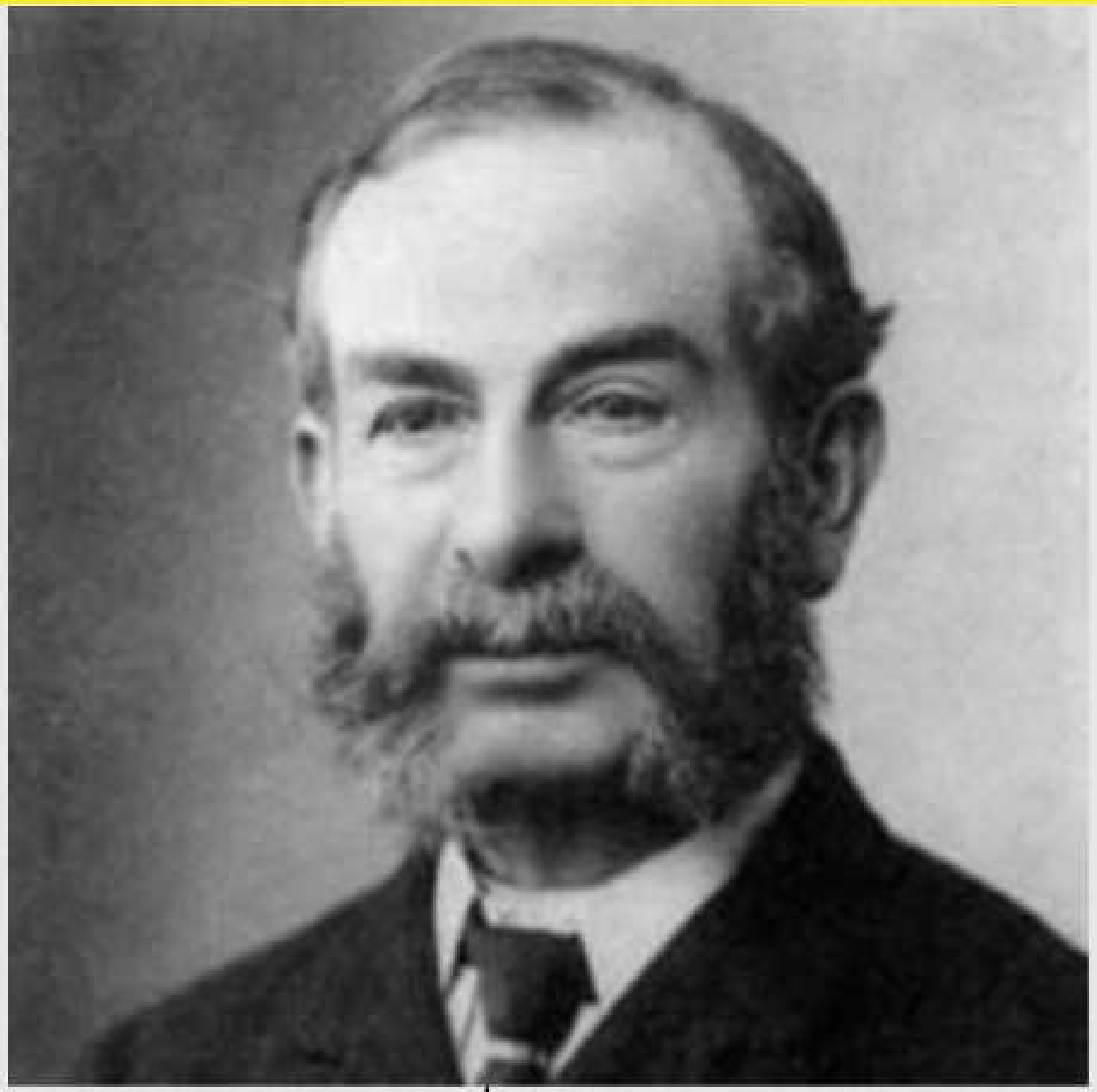
Arthur Cayley (1821-95)



Who Invented Eigendecomposition?

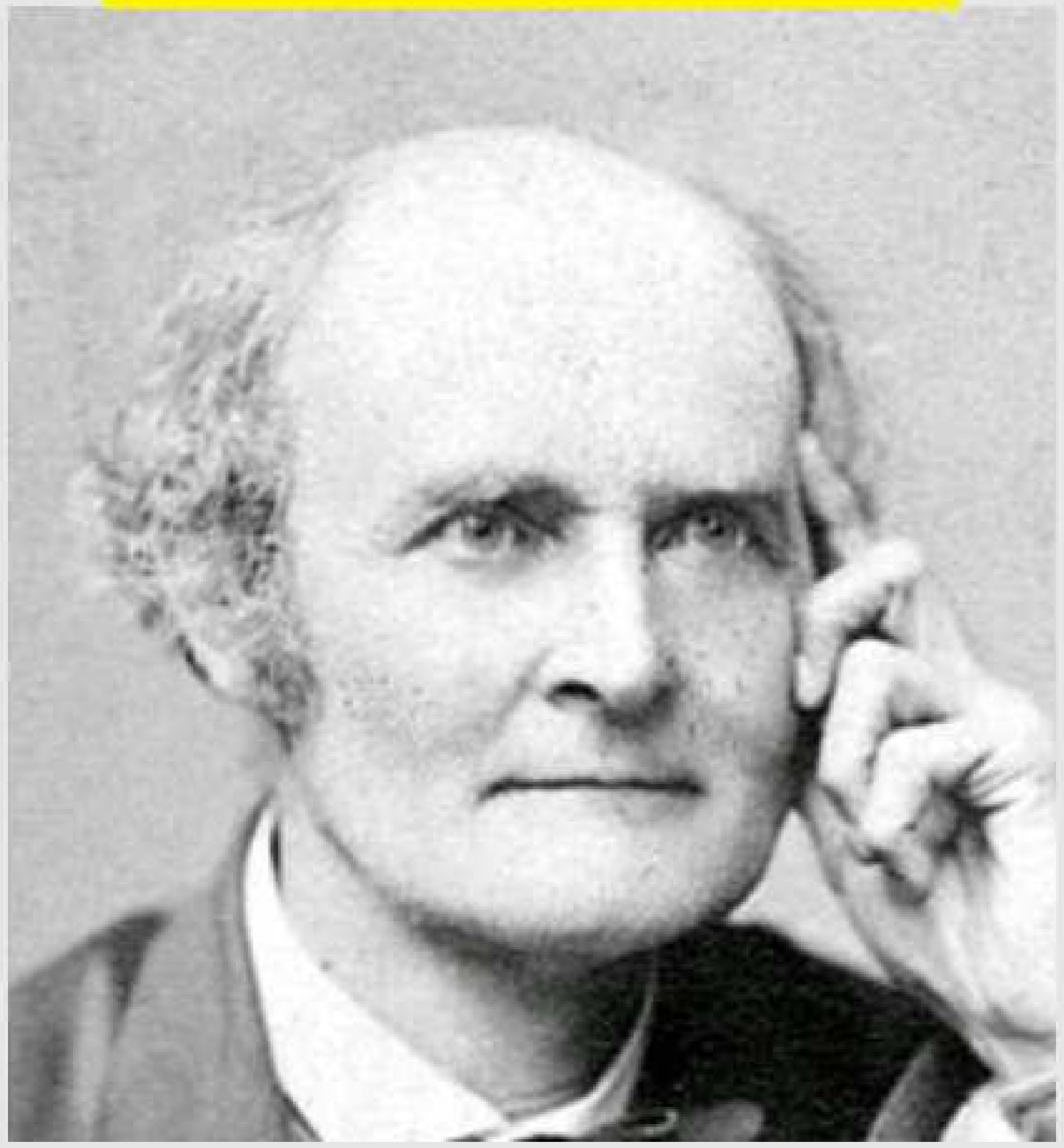
1852 - 1858

James Joseph Sylvester (1814-97)



also coined the term “matrix”

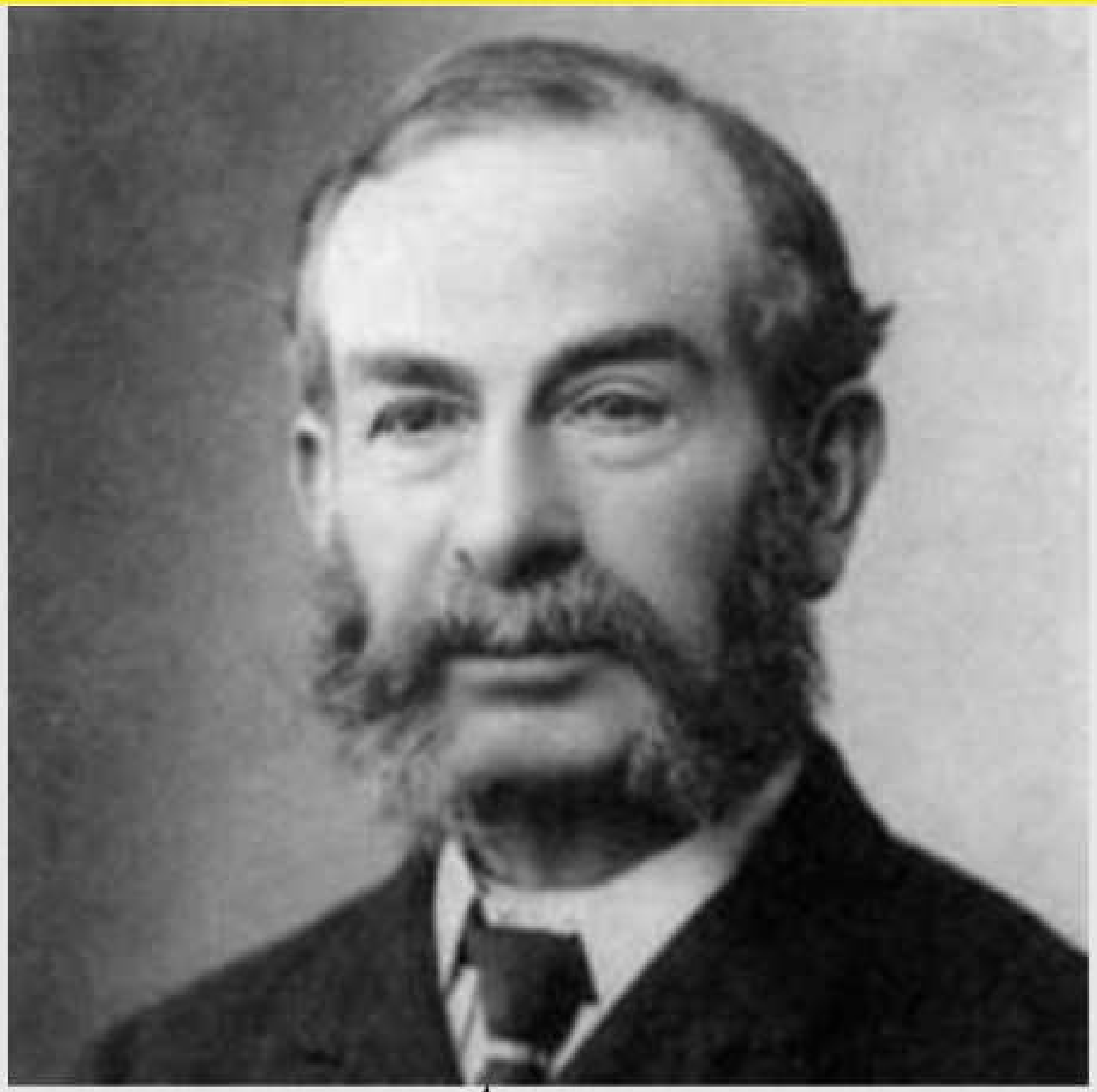
Arthur Cayley (1821-95)



Who Invented Eigendecomposition?

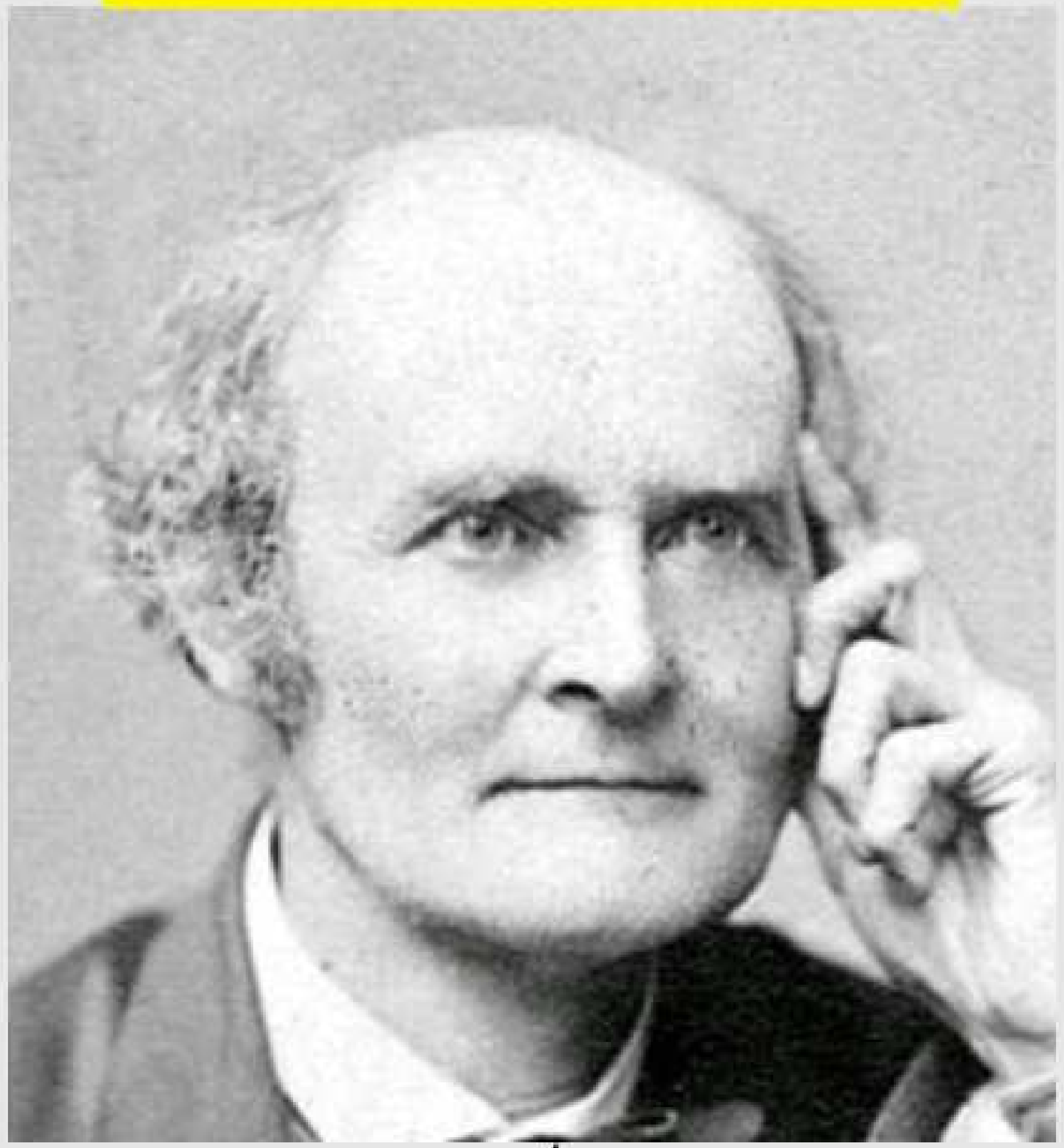
1852 - 1858

James Joseph Sylvester (1814-97)



also coined the term “matrix”

Arthur Cayley (1821-95)



Cayley-Hamilton Theorem

Who Invented Matrices?

- **known and used in China - before 100BC (!)**
  - explained in *Nine Chapters of the Mathematical Art* (1000-100 BC)
    - ➔ used to solve simultaneous eqns; they knew about determinants
  - 1545: brought from China to Italy (by **Cardano**)

Who Invented Matrices?

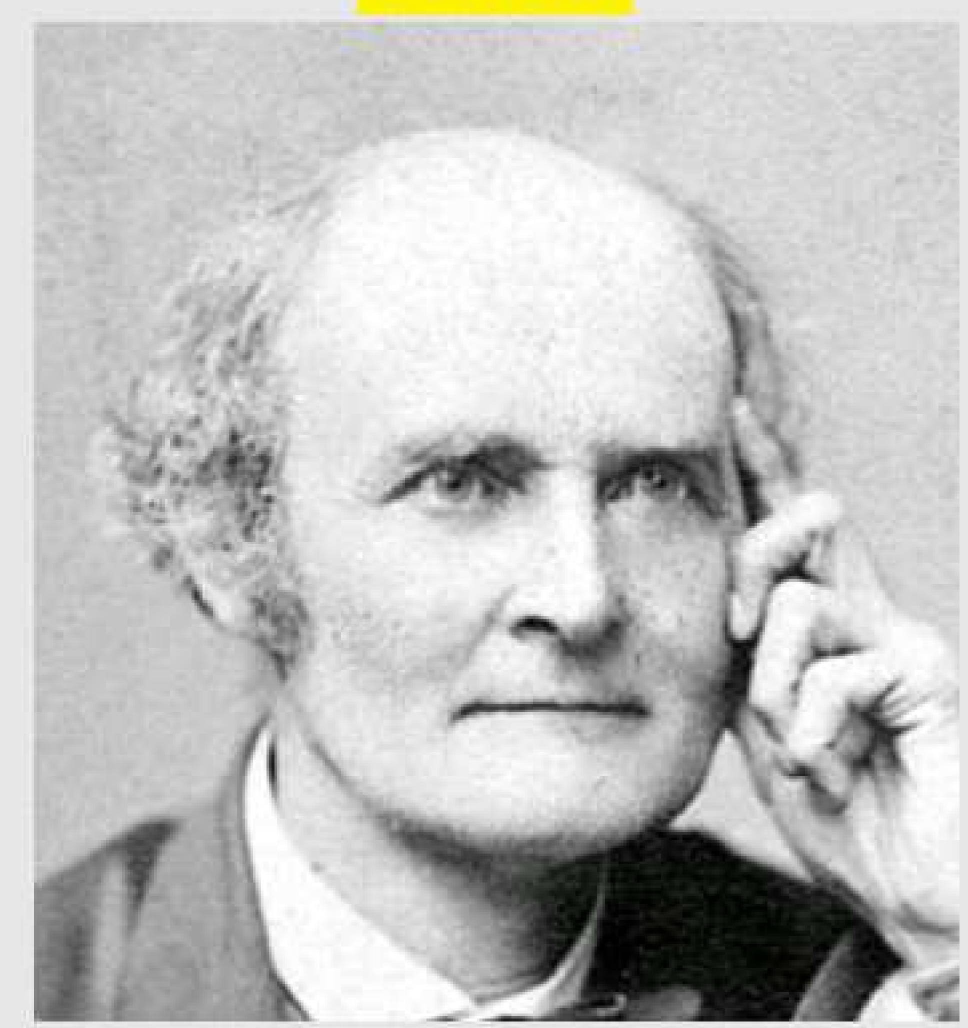
- **known and used in <u>China</u> - before 100BC (!)**
  - explained in *Nine Chapters of the Mathematical Art* (1000-100 BC)
    - used to solve simultaneous eqns; they knew about determinants
  - 1545: brought from China to Italy (by **Cardano**)
- 1683: **Seki** ("Japan's Newton") used matrices
- developed in Europe by **Gauss** and many others
  - finally, into its modern form by **Cayley** (mid 1800s)

Cardano

Gauss

Seki

Cayley



Charles Proteus Steinmetz
inventor of the phasor

- "Complex Quantities and their Use in Electrical Engineering", July 1893
  - revolutionized AC circuit/transmission calculations

Charles Proteus Steinmetz
inventor of the phasor

- "Complex Quantities and their Use in Electrical Engineering", July 1893
  - revolutionized AC circuit/transmission calculations

- suffered from hereditary dwarfism, hunchback, and hip dysplasia

Charles Proteus Steinmetz
inventor of the phasor

- "Complex Quantities and their Use in Electrical Engineering", July 1893
  - revolutionized AC circuit/transmission calculations

- suffered from hereditary dwarfism, hunchback, and hip dysplasia