MIMO systems.
Full state linear feedback controller.
LQR.

Elena VANNEAUX

elena.vanneaux@ensta-paris.fr

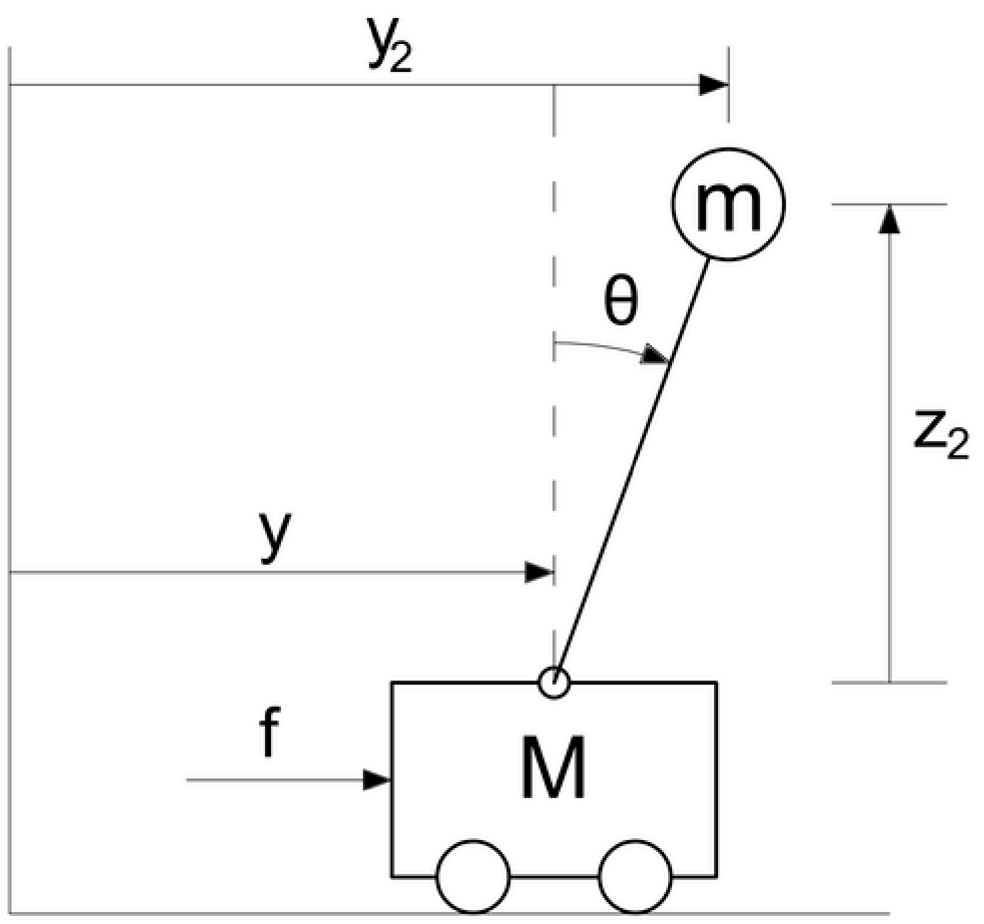
Course grade breakdowns

Labs - 40%

Final test - 30%

Final project - 30 %

Cart-pole control



Inverted pendulum on the cart can be modeled as follows

$$(M+m)\ddot{y} + b\dot{y} + ml\ddot{\theta}\cos\theta - ml\dot{\theta}^2\sin(\theta) = F$$

$$ml\cos(\theta)\ddot{y} + (I+ml^2)\ddot{\theta} - mgl\sin\theta = 0$$

where $F = u + w$, i.e. control + disturbance

Or in canonical state space ODE form

$$\begin{cases} \dot{y} = y_1 \\ \dot{y}_1 = \dfrac{-m^2l^2g\cos\theta\sin\theta + \left(I+ml^2\right)\left(ml\theta_1^2\sin\theta + F - by_1\right)}{(I+ml^2)(M+m) - m^2l^2\cos^2\theta} \\ \dot{\theta} = \theta_1 \\ \dot{\theta}_1 = \dfrac{(M+m)mgl\sin\theta + by_1 ml\cos\theta - m^2l^2\theta_1^2\cos\theta\sin\theta - mlF\cos\theta}{(M+m)(I+ml^2) - m^2l^2\cos^2\theta} \end{cases}$$

Cart-pole control

Lineralized model

$$\underbrace{\begin{bmatrix} \dot{y} \\ \dot{y}1 \\ \dot{\theta} \\ \dot{\theta}_1 \end{bmatrix}}_{x} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{-(I+ml^2)b}{I(M+m)+Mml^2} & \frac{-gm^2l^2}{I(M+m)+Mml^2} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{mlb}{I(M+m)+Mml^2} & \frac{mgl(M+m)}{I(M+m)+Mml^2} & 0 \end{bmatrix}}_{A} \begin{bmatrix} y \\ y_1 \\ \theta \\ \theta_1 \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ \frac{I+ml^2}{I(M+m)+Mml^2} \\ 0 \\ \frac{-ml}{I(M+m)+Mml^2} \end{bmatrix}}_{B=D} (u+w)$$

Design a PID controller such that

$$\theta(t) \to 0,$$

i.e. $C = [0\ 0\ 1\ 0]$, $y = Cx$, $x = \begin{bmatrix} y \\ y_1 \\ \theta \\ \theta_1 \end{bmatrix}$

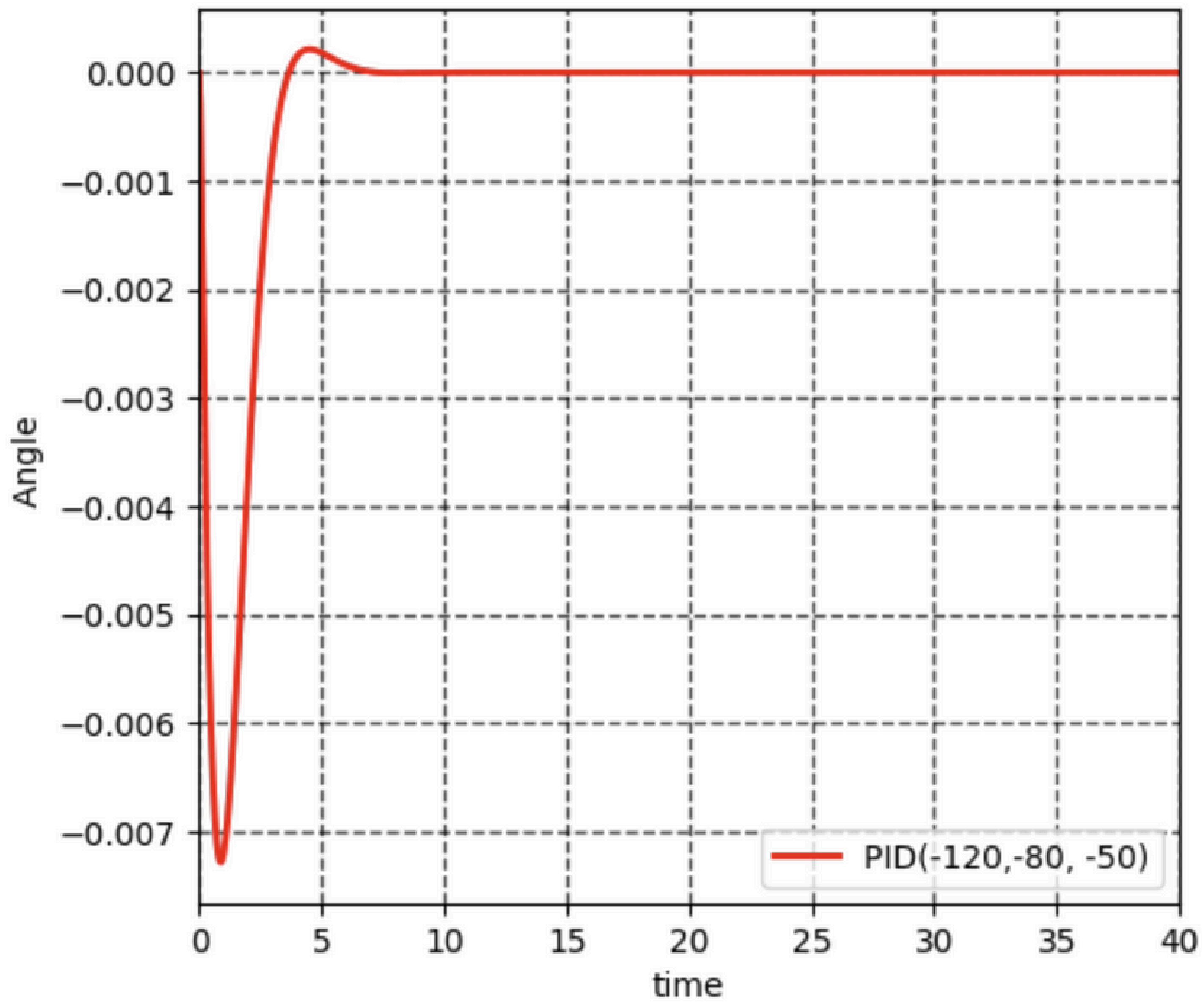
Cart-pole control. PID.

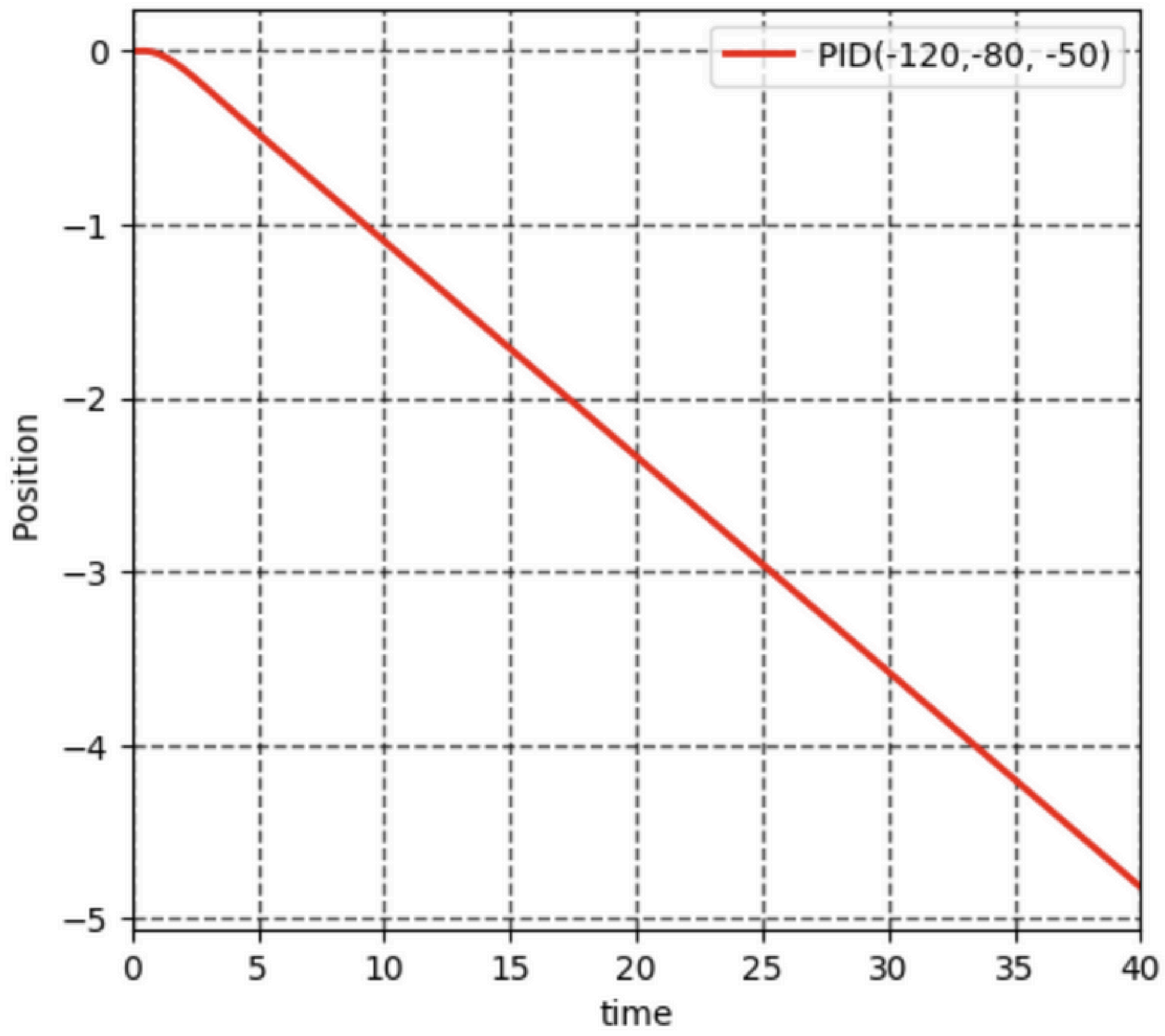
$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)\, d\tau + K_d \dot{e}(t)$$

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

$D = B$

$w = 0.1$





The controller keeps pendulum in up right position, but position of the cart goes to infinity....

Cart-pole control

Lineralized model

$$\underbrace{\begin{bmatrix} \dot{y} \\ \dot{y}1 \\ \dot{\theta} \\ \dot{\theta}_1 \end{bmatrix}}_{x} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{-(I+ml^2)b}{I(M+m)+Mml^2} & \frac{-gm^2 l^2}{I(M+m)+Mml^2} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{mlb}{I(M+m)+Mml^2} & \frac{mgl(M+m)}{I(M+m)+Mml^2} & 0 \end{bmatrix} \begin{bmatrix} y \\ y_1 \\ \theta \\ \theta_1 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{I+ml^2}{I(M+m)+Mml^2} \\ 0 \\ \frac{-ml}{I(M+m)+Mml^2} \end{bmatrix} (u+w)$$

Design a PID controller such that

$$\theta(t) \to 0, \quad \text{and } y(t) \to 0$$

$$\text{i.e. } C = [0\ 0\ 1\ 0],\ [1\ 0\ 0\ 0], \quad y = Cx, \quad x = \begin{bmatrix} y \\ y_1 \\ \theta \\ \theta_1 \end{bmatrix}$$

Cart-pole control

Lineralized model

$$\underbrace{\begin{bmatrix} \dot{y} \\ \dot{y}1 \\ \dot{\theta} \\ \dot{\theta}_1 \end{bmatrix}}_{x} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{-(I+ml^2)b}{I(M+m)+Mml^2} & \frac{-gm^2 l^2}{I(M+m)+Mml^2} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{mlb}{I(M+m)+Mml^2} & \frac{mgl(M+m)}{I(M+m)+Mml^2} & 0 \end{bmatrix}} \begin{bmatrix} y \\ y_1 \\ \theta \\ \theta_1 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{I+ml^2}{I(M+m)+Mml^2} \\ 0 \\ \frac{-ml}{I(M+m)+Mml^2} \end{bmatrix} (u + w)$$

Design a feedback controller u = g(x) such that

$$x(t) \to 0$$

robustly to any initial condition $x(0) = x_0$,

and any disturbance $w(t)$

Regulator for LTI systems

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

Design a feedback controller u = g(Cx) such that

$x(t) \to 0$

robustly to any initial condition $x(0) = x_0$,

and any disturbance $w(t)$

Regulator for LTI systems

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

$$y = Cx$$

Design a linear feedback controller u = -Ky such that

$x(t) \to 0$

robustly to any initial condition $x(0) = x_0$,

and any disturbance $w(t)$

Regulator for LTI systems

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

$$y = Cx$$, let $C = \text{eye}(n)$, i.e $y = x$

Design a linear full state feedback controller u = -Kx such that

$x(t) \to 0$

robustly to any initial condition $x(0) = x_0$,

and any disturbance $w(t)$

Regulator for LTI systems

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

$y = x$

Design a linear full state feedback controller u = -Kx such that

$x(t) \to 0$

robustly to any initial condition $x(0) = x_0$,

and any disturbance $w(t)$

Regulator for LTI systems

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

$$y = x$$

Closed-loop system

$$\begin{cases} \dot{x} = (A - BK)x + Dw \\ y = x \end{cases}$$

Design a linear full state feedback controller u = -Kx such that

$$x(t) \to 0$$

robustly to any initial condition $x(0) = x_0$,

and any disturbance $w(t)$

Let's first consider the case with no disturbance

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

$y = x$

Closed-loop system

$\dot{x} = (A - BK)x + \underbrace{Dw}_{=0}$

$y = x$

Design a linear full state feedback controller u = -Kx such that

$x(t) \to 0$

robustly to any initial condition $x(0) = x_0$,

and any disturbance $w(t)$

Let's first consider the case with no disturbance

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

what matrix K should be like, to ensure that

$$x(t) \to 0$$

robustly to any initial condition $x(0) = x_0$?

Let's first consider the case with no disturbance

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

what matrix K should be like, to ensure that

$x(t) \to 0$

robustly to any initial condition $x(0) = x_0$?

Asymptotic Stability. The system $\dot{\mathbf{x}}(\mathbf{t}) = \mathbf{A}\mathbf{x}(\mathbf{t})$ is asymptotically stable if *every* finite initial state $\mathbf{x}_0$ excites a bounded response $\mathbf{x}(\mathbf{t})$ that approaches 0 as $\mathbf{t} \to \infty$.

Let's first consider the case with no disturbance

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

what matrix K should be like, to ensure that

$$x(t) \to 0$$

robustly to any initial condition $x(0) = x_0$?

i.e. $\dot{x} = (A - BK)x$ should be asymptotically stable

Let's first consider the case with no disturbance

Theorem (Internal Stability). The equation $\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t)$ is Asymptotically stable if and only if all eigenvalues of $\mathbf{A}$ have negative real parts.

i.e. $\dot{x} = (A - BK)x$ should be **asymptotically stable**

Let's first consider the case with no disturbance

Theorem (Internal Stability). The equation $\dot{x}(t) = Ax(t)$ is Asymptotically stable if and only if all eigenvalues of A have negative real parts.

i.e. matrix K should be such that
all eigenvalues of matrix (A-BK) have negative real parts

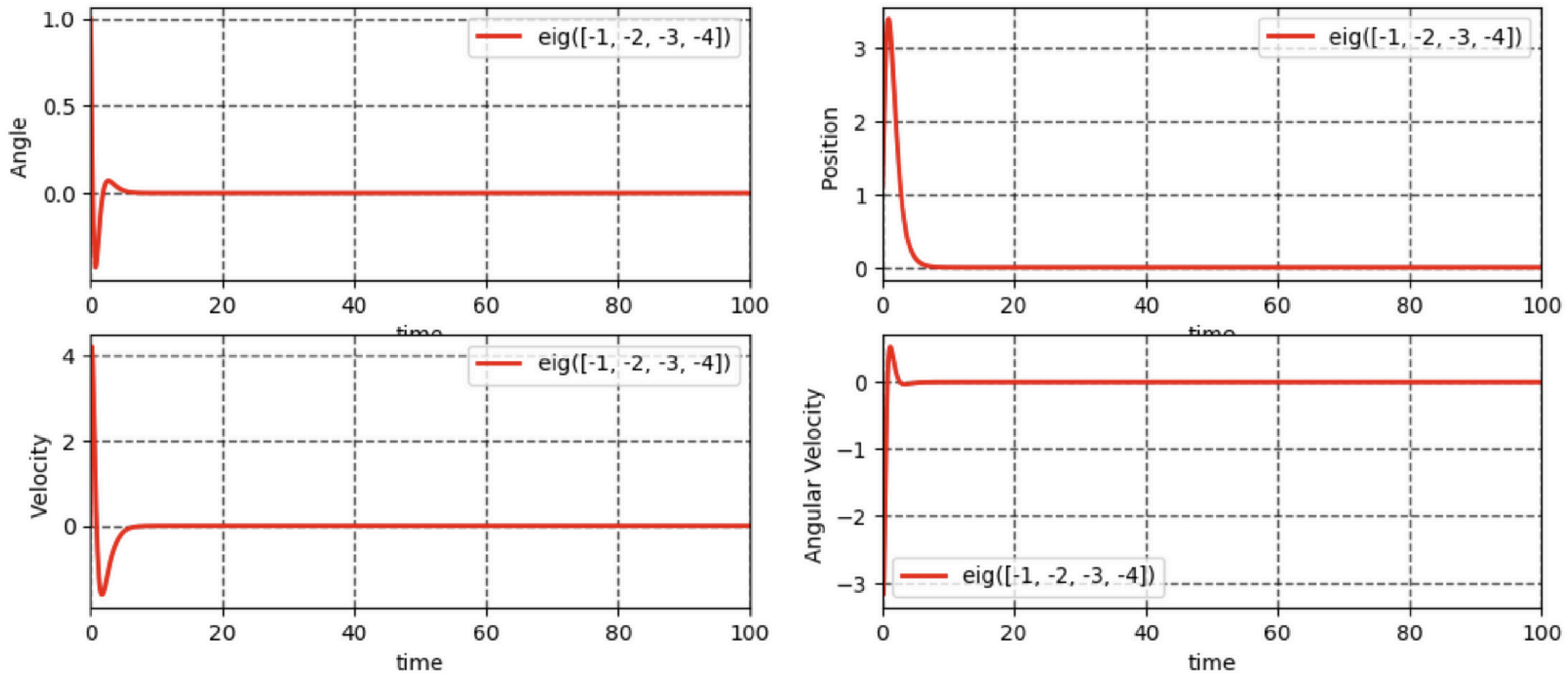
i.e. $\dot{x} = (A - BK)x$ **should be asymptotically stable**

Eigenvalues assigntement

Theorem (Controllability and Feedback — MIMO). The pair $(\mathbf{A} - \mathbf{B}\mathbf{K}, \mathbf{B})$, for any $\mathbf{p} \times \mathbf{n}$ real matrix $\mathbf{K}$ is controllable if and only if the pair $(\mathbf{A}, \mathbf{B})$ is controllable.

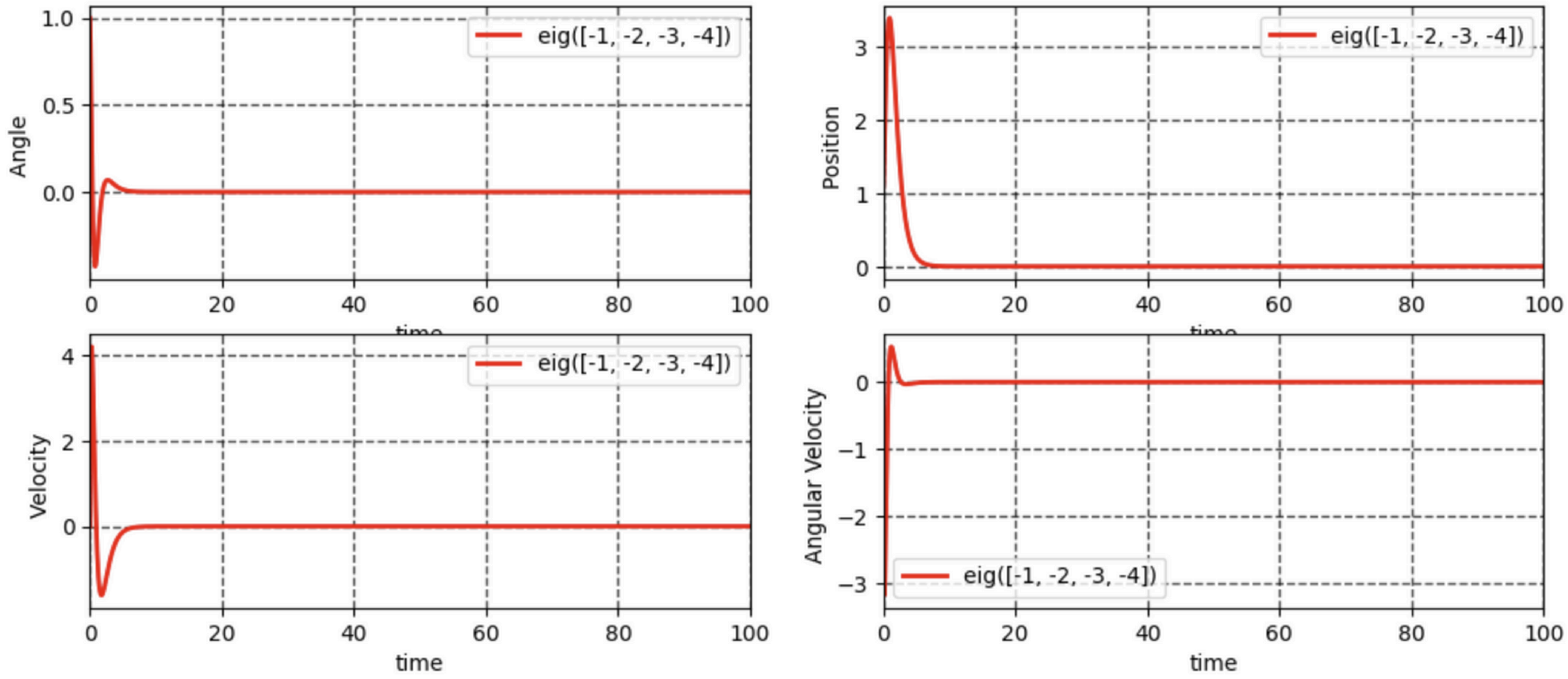
Theorem (Eigenvalue assignment — MIMO). All eigenvalues of $(\mathbf{A} - \mathbf{B}\mathbf{K})$ can be assigned arbitrarily (provided complex eigenvalues are assigned in conjugated pairs) by selecting a real constant $\mathbf{K}$ if and only if $(\mathbf{A}, \mathbf{B})$ is controllable.

Cart-pole control.
Linear full state feedback controller



$w(t) = 0$, $x_0 = (0, 0, 1, 0)$

Cart-pole control.
Linear full state feedback controller



$w(t) = 0$, $x_0 = (1, 0, 1, 0)$

Eigenvalues assigntement

control.place

`control.place(A, B, p)` [source]

Place closed loop eigenvalues.

K = place(A, B, p)

Parameters
- **A** (*2D array_like*) – Dynamics matrix
- **B** (*2D array_like*) – Input matrix
- **p** (*1D array_like*) – Desired eigenvalue locations

Returns **K** – Gain such that A - B K has eigenvalues given in p

Return type 2D array (or matrix)

Notes

Algorithm

This is a wrapper function for `scipy.signal.place_poles()`, which implements the Tits and Yang algorithm [1]. It will handle SISO, MISO, and MIMO systems. If you want more control over the algorithm, use `scipy.signal.place_poles()` directly.

Limitations

The algorithm will not place poles at the same location more than rank(B) times.

Python control system library

Eigenvalues assigntement

control.place

`control.place(A, B, p)` [source]

Place closed loop eigenvalues.

K = place(A, B, p)

Parameters
- **A** (*2D array_like*) – Dynamics matrix
- **B** (*2D array_like*) – Input matrix
- **p** (*1D array_like*) – Desired eigenvalue locations

Returns K – Gain such that A - B K has eigenvalues given in p

Return type 2D array (or matrix)

Notes

Algorithm

This is a wrapper function for `scipy.signal.place_poles()`, which implements the Tits and Yang algorithm [1]. It will handle SISO, MISO, and MIMO systems. If you want more control over the algorithm, use `scipy.signal.place_poles()` directly.

Limitations

The algorithm will not place poles at the same location more than rank(B) times.

control.place_varga

`control.place_varga(A, B, p, dtime=False, alpha=None)` [source]

Parameters
- **A** (*2D array_like*) – Dynamics matrix
- **B** (*2D array_like*) – Input matrix
- **p** (*1D array_like*) – Desired eigenvalue locations
- **dtime** (*bool, optional*) – False for continuous time pole placement or True for discrete time. The default is dtime=False.
- **alpha** (*float, optional*) –
  If *dtime* is false then place_varga will leave the eigenvalues with real part less than alpha untouched. If *dtime* is true then place_varga will leave eigenvalues with modulus less than alpha untouched.
  By default (alpha=None), place_varga computes alpha such that all poles will be placed.

Returns K – Gain such that A - B K has eigenvalues given in p.

Return type 2D array (or matrix)

See also

`place`, `acker`

Notes

This function is a wrapper for the slycot function sb01bd, which implements the pole placement algorithm of Varga [1]. In contrast to the algorithm used by place(), the Varga algorithm can place multiple poles at the same location. The placement, however, may not be as robust.

Python control system library

Eigenvalues assigntement

Example (Nonuniqueness of K in MIMO state feedback). As a simple MIMO system consider the second order system with two inputs

$$\dot{x}(t) = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} u(t)$$

The system has two eigenvalues at $s = 0$, and it is controllable, since $B = I$, so $\mathcal{C} = [\, B \;\; AB \,]$ is full rank.

Let's consider the state feedback

$$u(t) = -Kx(t) = \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix} x(t)$$

Then the closed loop evolution matrix is

$$A - BK = \begin{bmatrix} -k_{11} & -k_{12} \\ 1-k_{21} & -k_{22} \end{bmatrix}$$

Eigenvalues assigntement

Example (Continuation). Suppose that we would like to place both closed-loop eigenvalues at $s = -1$, i.e., the roots of the characteristic polynomial $s^2 + 2s + 1$. Then, one possibility would be to select

$$\begin{cases} k_{11} = 2 \\ k_{12} = 1 \\ k_{21} = 0 \\ k_{22} = 0 \end{cases} \Rightarrow \mathbf{A} - \mathbf{B}\mathbf{K} = \begin{bmatrix} -2 & -1 \\ 1 & 0 \end{bmatrix} \Rightarrow \text{eigenvalues at } s = -1$$

But the alternative selection

$$\begin{cases} k_{11} = 1 \\ k_{12} = \text{free} \\ k_{21} = 1 \\ k_{22} = -1 \end{cases} \Rightarrow \mathbf{A} - \mathbf{B}\mathbf{K} = \begin{bmatrix} -1 & k_{12} \\ 0 & -1 \end{bmatrix} \Rightarrow \text{also eigenvalues at } s = -1$$

As we see, **there are infinitely many** possible selections of $\mathbf{K}$ that will give the same eigenvalues of $(\mathbf{A} - \mathbf{B}\mathbf{K})$! $\square$

Eigenvalues assigntement

The "excess of freedom" in MIMO state feedback design could be a problem if we don't know how to best use it. . .

There are several ways to tackle the problem of selecting $\mathbf{K}$ from an infinite number of possibilities, among them

- Optimal Design. Computes the **best** $\mathbf{K}$ by optimising a suitable cost function.

Linear Quadratic Regulator

Theorem (LQR). Consider the state space system

$$\begin{aligned} \dot{x} &= Ax + Bu, \quad && x \in \mathbb{R}^n, u \in \mathbb{R}^p \\ y &= Cx, && y \in \mathbb{R}^q \end{aligned}$$

and the performance criterion

$$J = \int_0^\infty \left[x^T(t) Q x(t) + u^T(t) R u(t) \right] dt, \tag{J}$$

where Q is non negative definite and R is positive definite. Then the optimal control minimising (J) is given by the **linear** state feedback law

$$\boxed{u(t) = -Kx(t)} \quad \text{with} \quad \boxed{K = R^{-1} B^T P}$$

and where P is the unique positive definite solution to the matrix **Algebraic Riccati Equation** (ARE)

$$\boxed{A^T P + PA - PBR^{-1} B^T P + Q = 0}$$

Linear Quadratic Regulator

`control.lqr(A, B, Q, R[, N])` [source]

Linear quadratic regulator design.

The lqr() function computes the optimal state feedback controller u = -K x that minimizes the quadratic cost

$$J = \int_0^\infty (x'Qx + u'Ru + 2x'Nu)dt$$

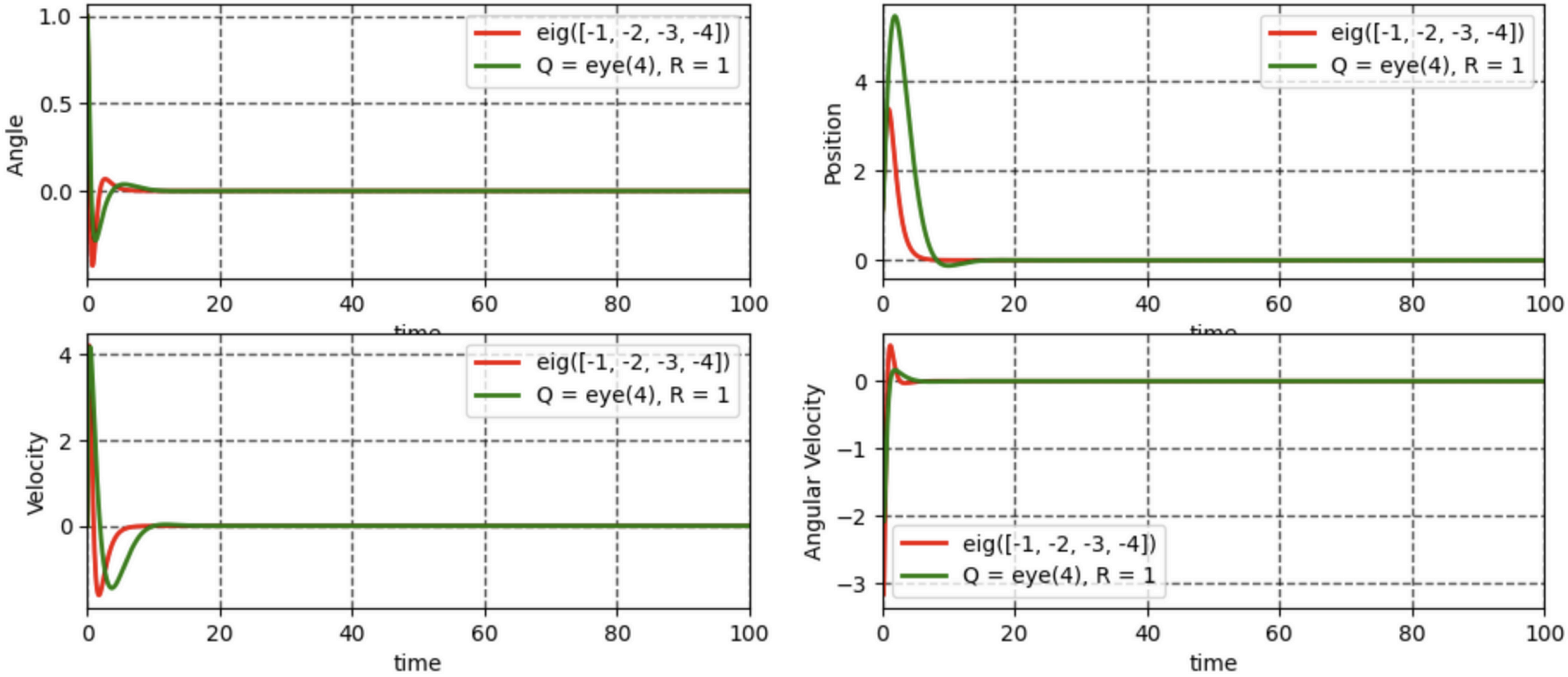
The function can be called with either 3, 4, or 5 arguments:

- `K, S, E = lqr(sys, Q, R)`
- `K, S, E = lqr(sys, Q, R, N)`
- `K, S, E = lqr(A, B, Q, R)`
- `K, S, E = lqr(A, B, Q, R, N)`

where *sys* is an *LTI* object, and *A*, *B*, *Q*, *R*, and *N* are 2D arrays or matrices of appropriate dimension.

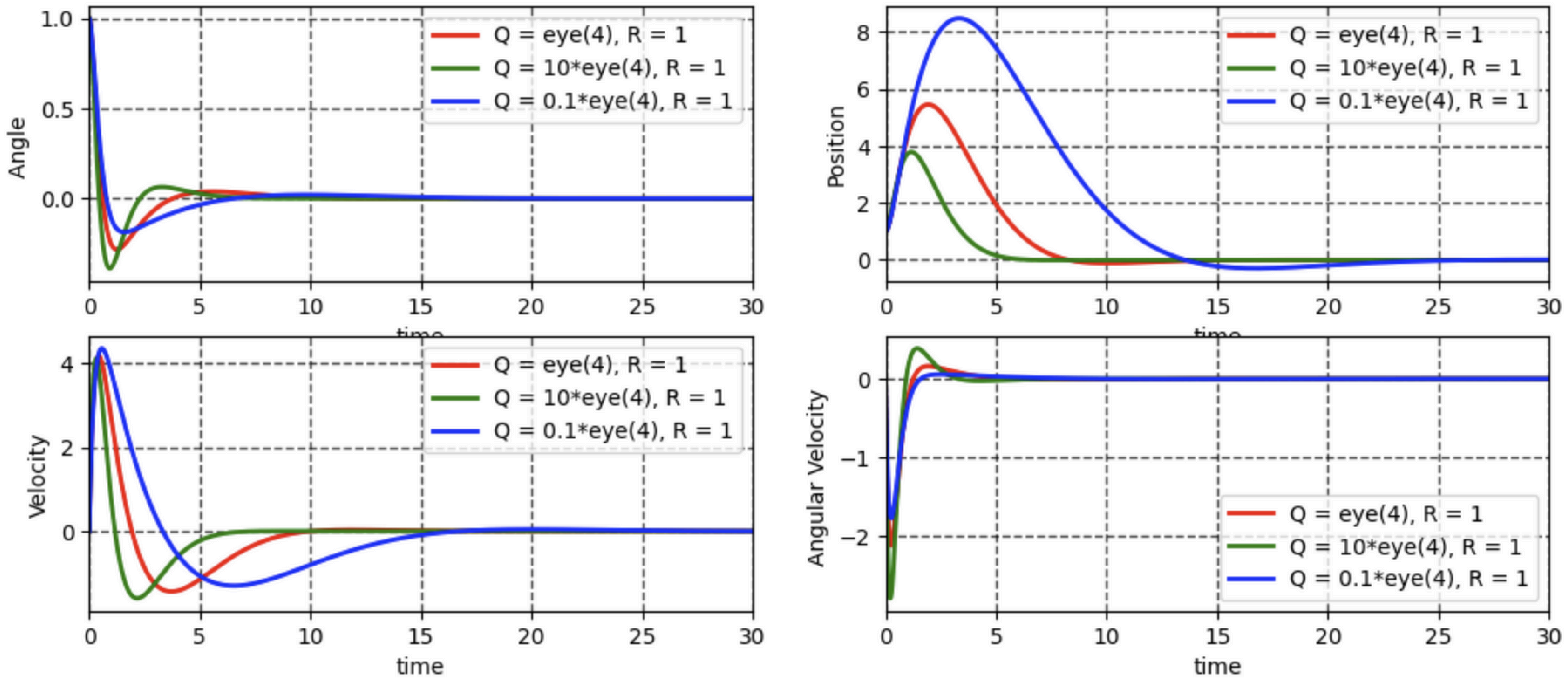
Python control system library

Cart-pole control. LQR.



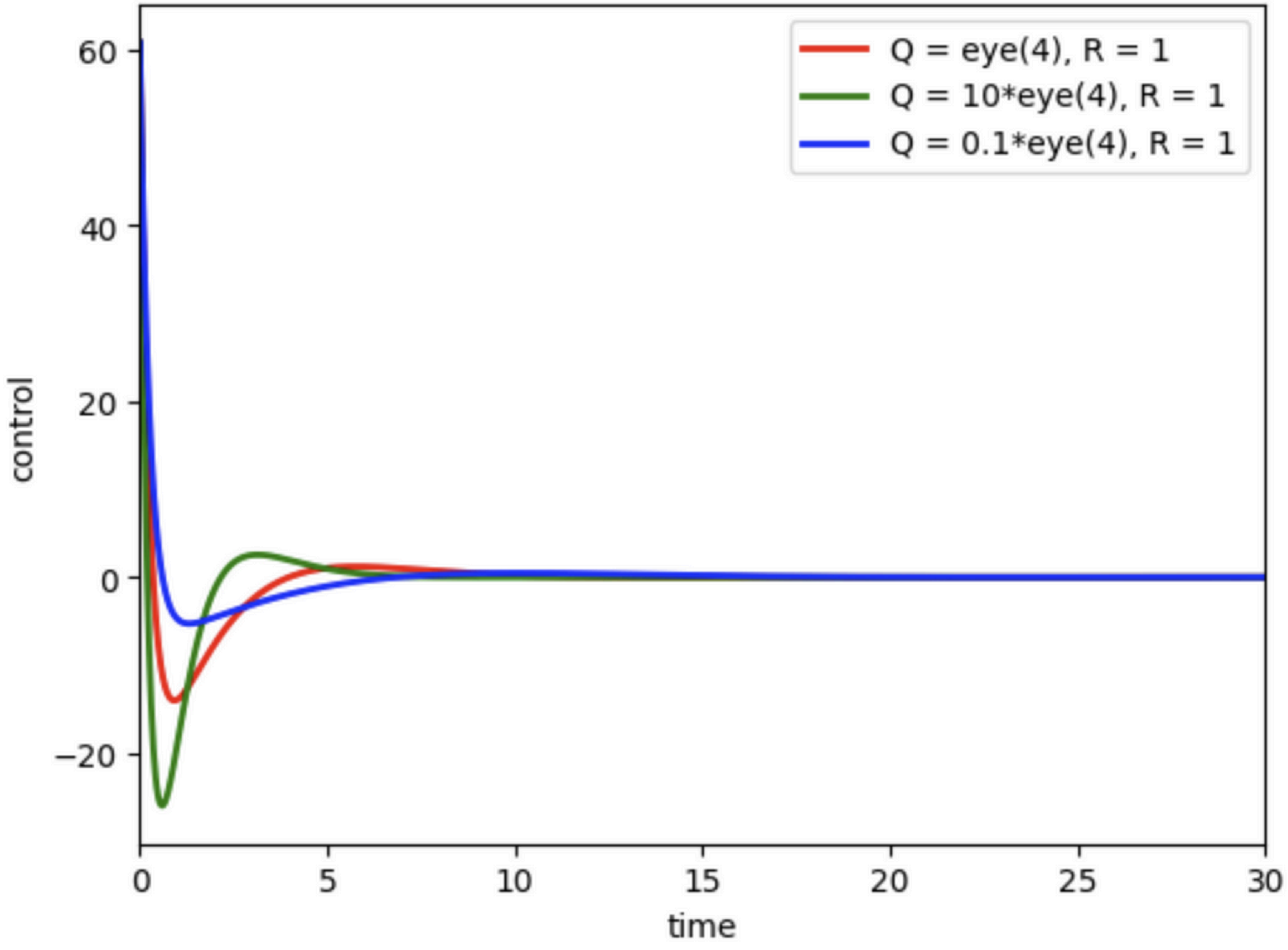
$w(t) = 0$, $x_0 = (1, 0, 1, 0)$

Cart-pole control. LQR.



When Q is “larger” the state converges “faster”

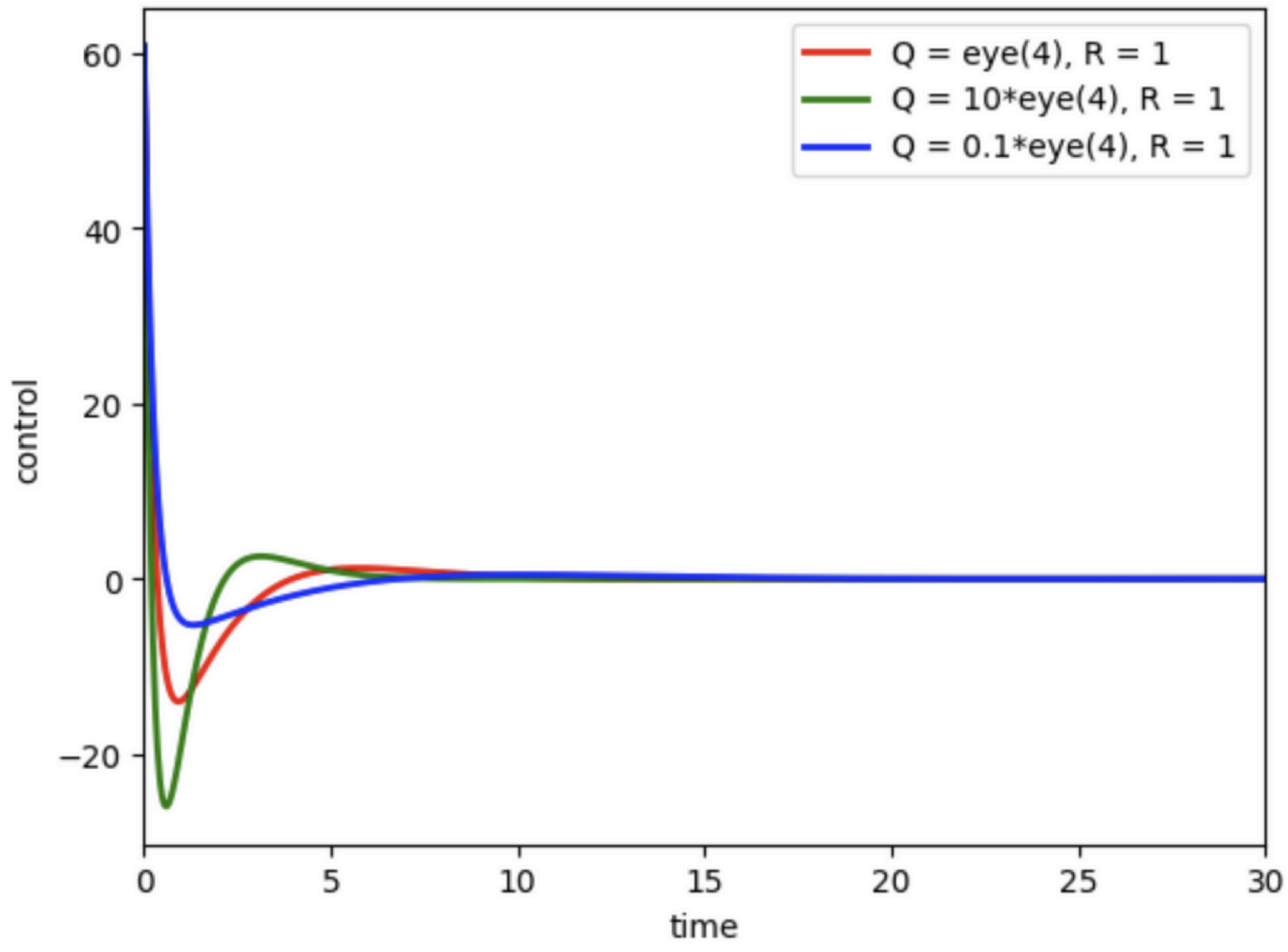
Cart-pole control. LQR.



but, to converge faster we need to use more aggressive control

$$u = -Kx$$

Cart-pole control. LQR.



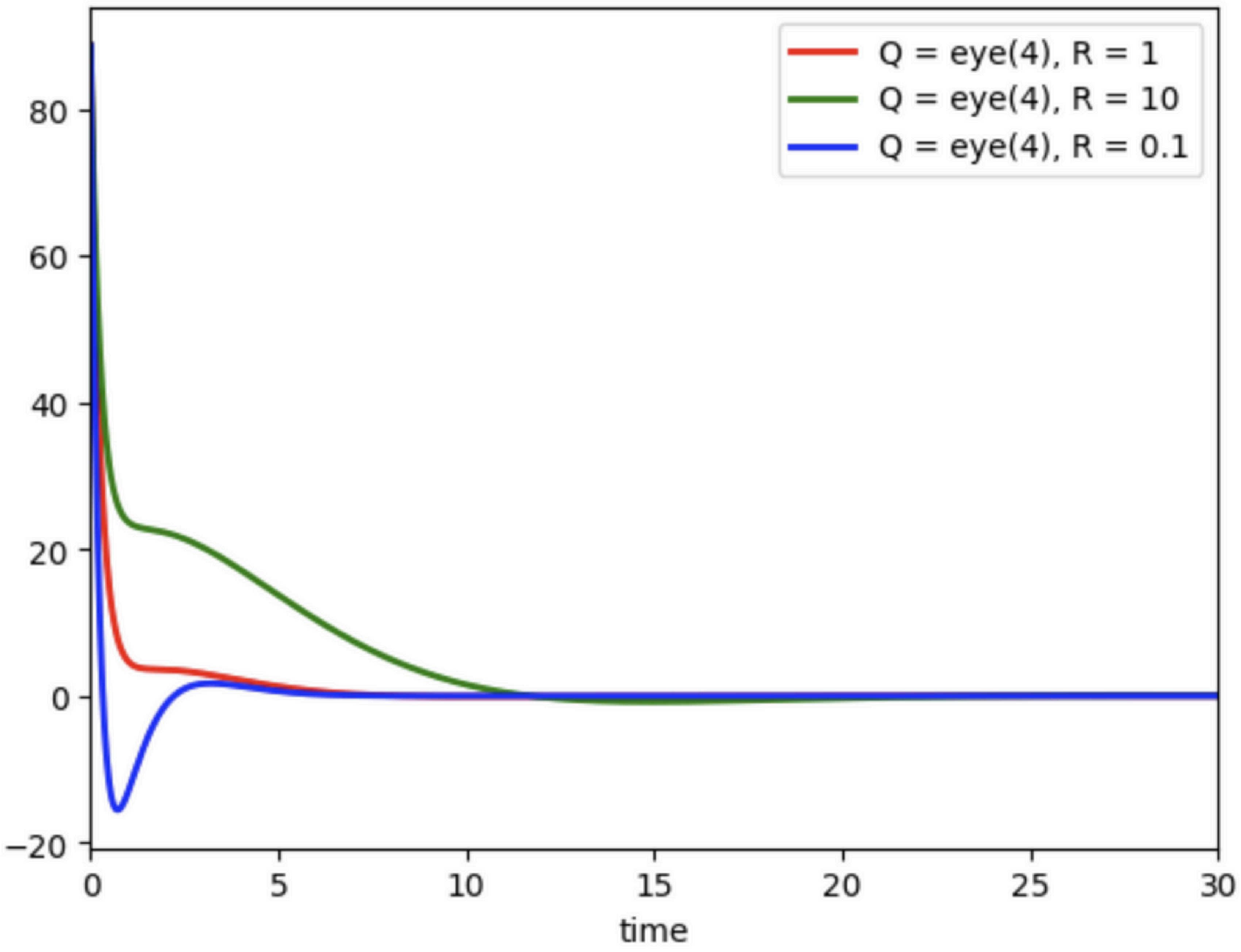
but, to converge faster we need to use more aggressive control

which is not always feasible due the actuator constraints

$||u|| \leq const$

$u = -Kx$

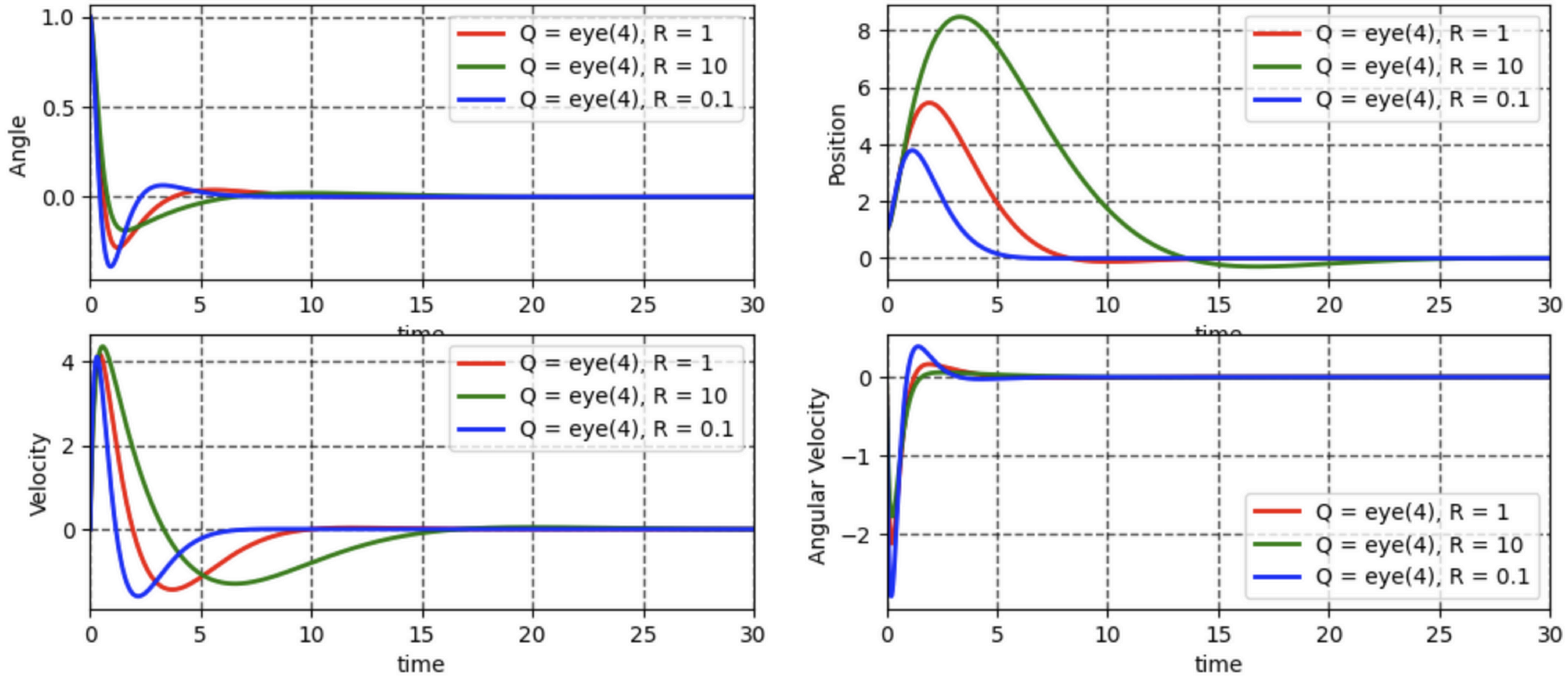
Cart-pole control. LQR.



by increasing R we ask the controller to less aggressive

$$u = -Kx$$

Cart-pole control. LQR.



but the state converges slower

Linear Quadratic Regulator

In hindsight, LQR can be interpreted as *optimal pole placement* for

$$\dot{x}(t) = (A - BK^\star)\, x(t)$$

trading off minimal state deviation and minimal control energy:

$$K^\star = \text{argmin}_K \int_0^\infty \underbrace{x(t)^\top Q\, x(t)}_{\text{state deviation}} + \underbrace{x(t)^\top K^\top R K\, x(t)}_{\text{control energy}}\, dt$$

We have designed a regulator for non disturbed case

i.e. if (A,B) is controllable then
we always can chose matrix K such that
all eigenvalues of matrix (A-BK)
have negative real parts

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

i.e. design a linear full state feedback controller u = -Kx such that

$x(t) \to 0$ robustly to any initial condition $x(0) = x_0$

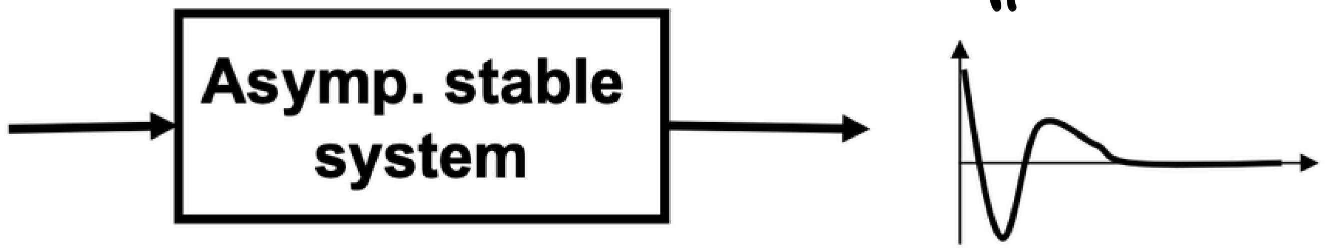
What if we do have disturbances?

Stability of LTI systems

Asymptotic Stability. The system $\dot{x}(t) = Ax(t)$ is asymptotically stable if *every* finite initial state x_0 excites a bounded response $x(t)$ that approaches 0 as $t \to \infty$.

$\dot{x} = (A - BK)x$ $\quad \|x(t)\| \to 0$

$w(t) = 0$

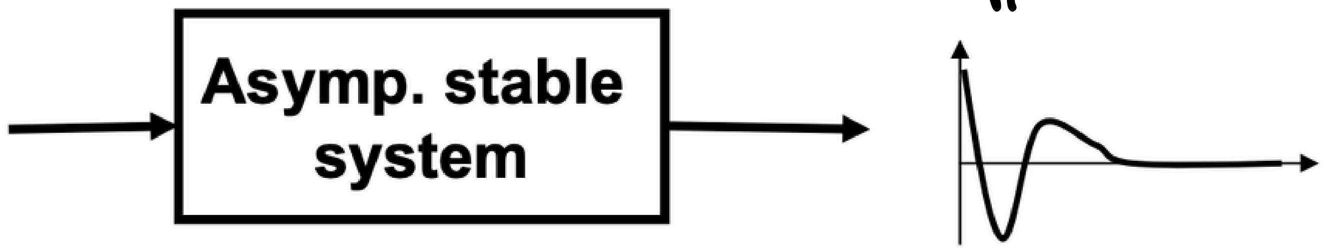


Stability of LTI systems

Asymptotic Stability. The system $\dot{x}(t) = Ax(t)$ is asymptotically stable if *every* finite initial state x_0 excites a bounded response $x(t)$ that approaches 0 as $t \to \infty$.

$\dot{x} = (A - BK)x$ $\quad \|x(t)\| \to 0$

$w(t) = 0$



BIBO Stability. A system is BIBO (**bounded-input bounded-output**) stable if *every* bounded input produces a bounded output.

ICs=0

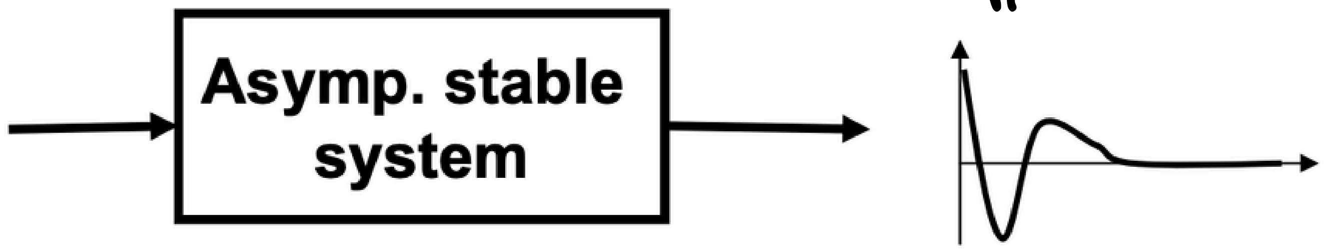
$\|w(t)\|$ $\quad \|x(t)\|$

It is know that...

Asymptotic Stability. The system $\dot{\mathbf{x}}(t) = A\mathbf{x}(t)$ is asymptotically stable if *every* finite initial state $\mathbf{x}_0$ excites a bounded response $\mathbf{x}(t)$ that approaches 0 as $t \to \infty$.

$\dot{x} = (A - BK)x$ $\quad \|x(t)\| \to 0$

$w(t) = 0$



BIBO Stability. A system is BIBO (**bounded-input bounded-output**) stable if *every* bounded input produces a bounded output.

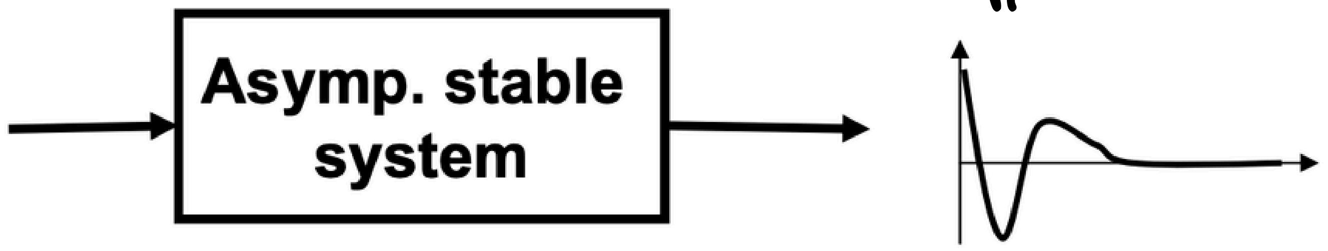
$\|w(t)\|$ ICs=0 $\|x(t)\|$

It is know that...

Asymptotic Stability. The system $\dot{\mathbf{x}}(t) = A\mathbf{x}(t)$ is asymptotically stable if *every* finite initial state $\mathbf{x}_0$ excites a bounded response $\mathbf{x}(t)$ that approaches 0 as $t \to \infty$.

$\dot{x} = (A - BK)x$ $\quad \|x(t)\| \to 0$

$w(t) = 0$



BIBO Stability. A system is BIBO (**bounded-input bounded-output**) stable if *every* bounded input produces a bounded output.

ICs=0

$\|w(t)\|$ $\quad \|x(t)\|$

Let’s disturbance is bounded...

i.e. if (A,B) is controllable then
we always can chose matrix K such that
all eigenvalues of matrix (A-BK)
have negative real parts

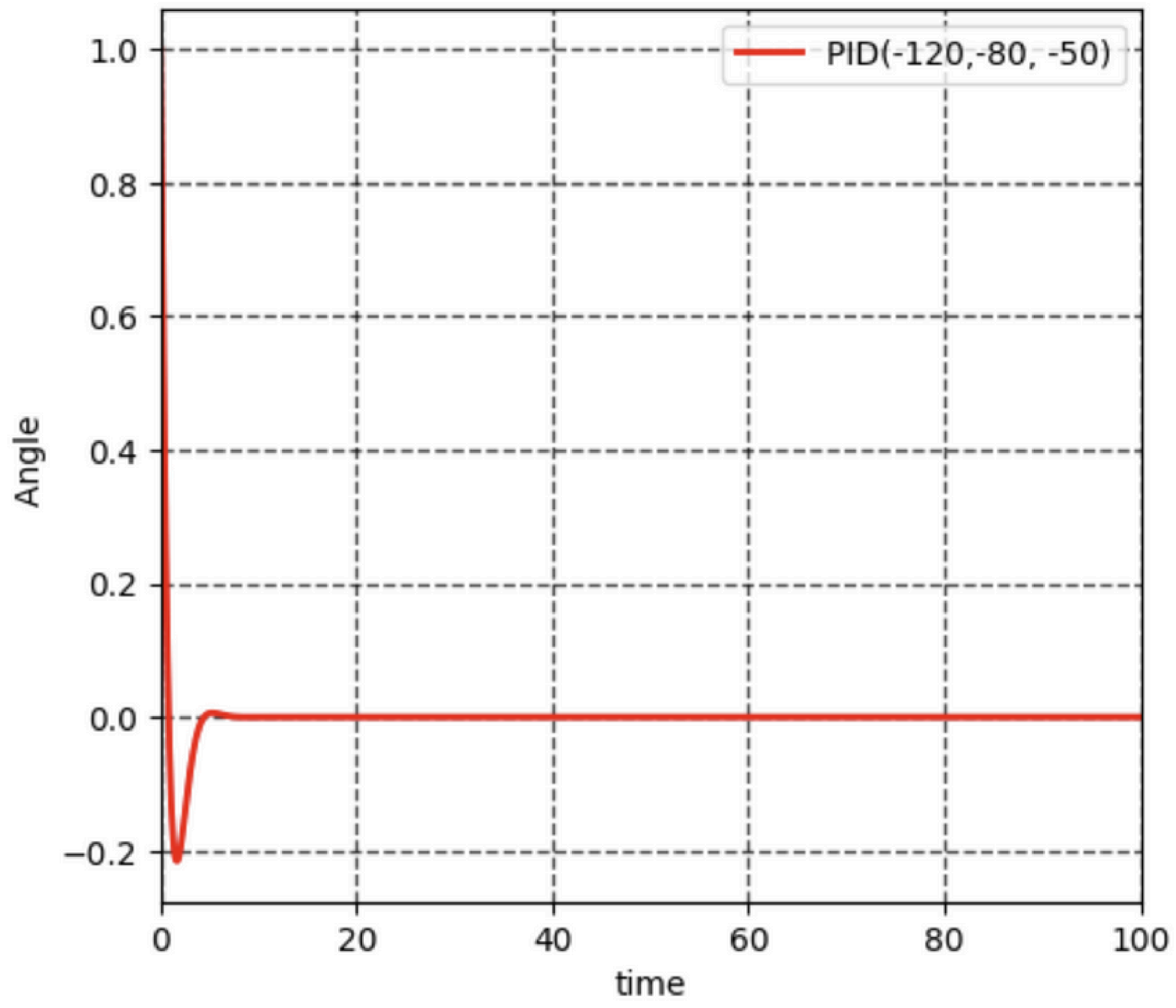
Consequently $\dot{x} = (A - BK)x$ asymptotically stable

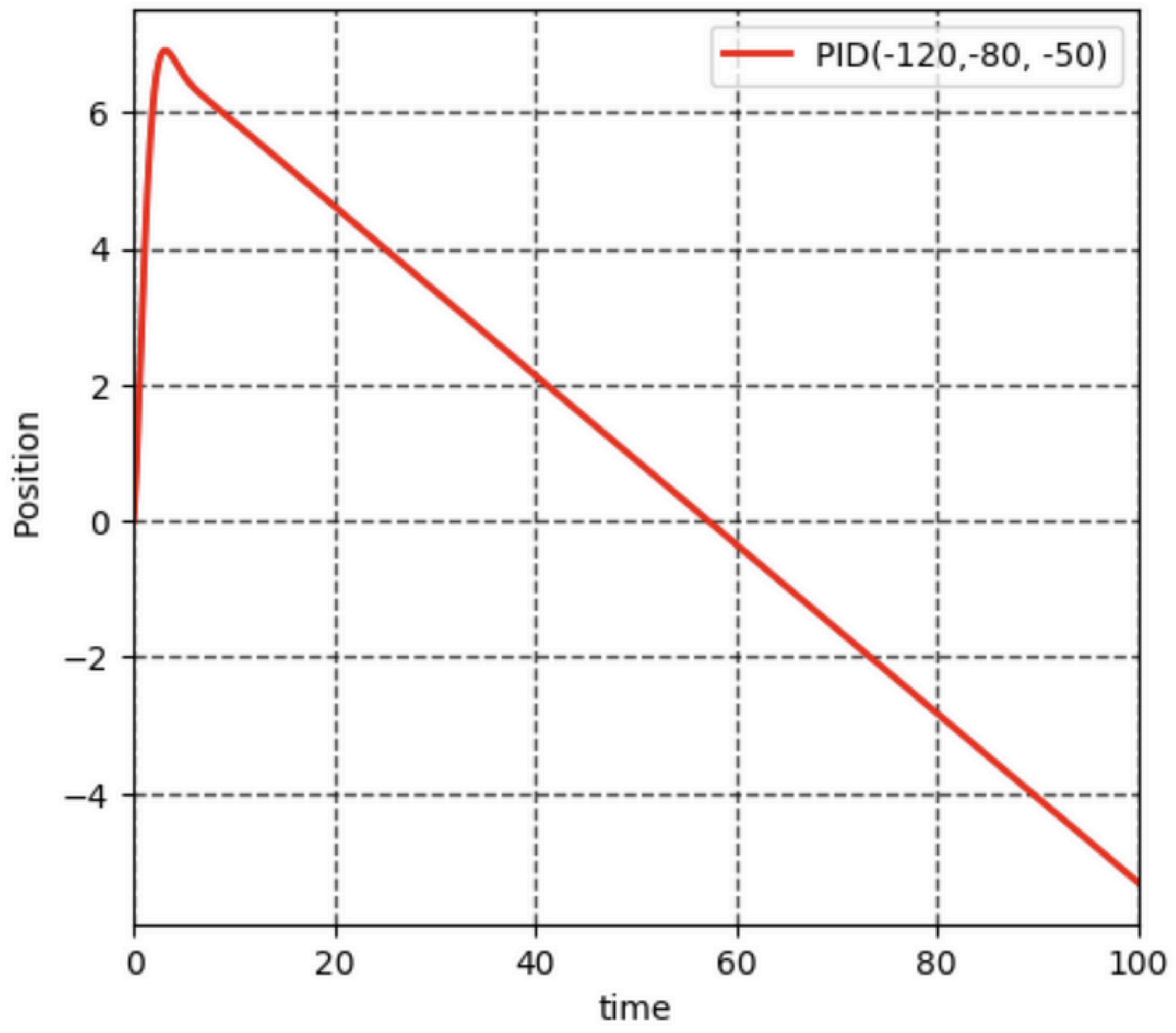
i.e. design a linear full state feedback controller u = -Kx such that

$\|x(t)\| \leq const$ robustly to any initial condition $x(0) = x_0$

and any bounded disturbance $w(t)$, $\|w(t)\| \leq const$

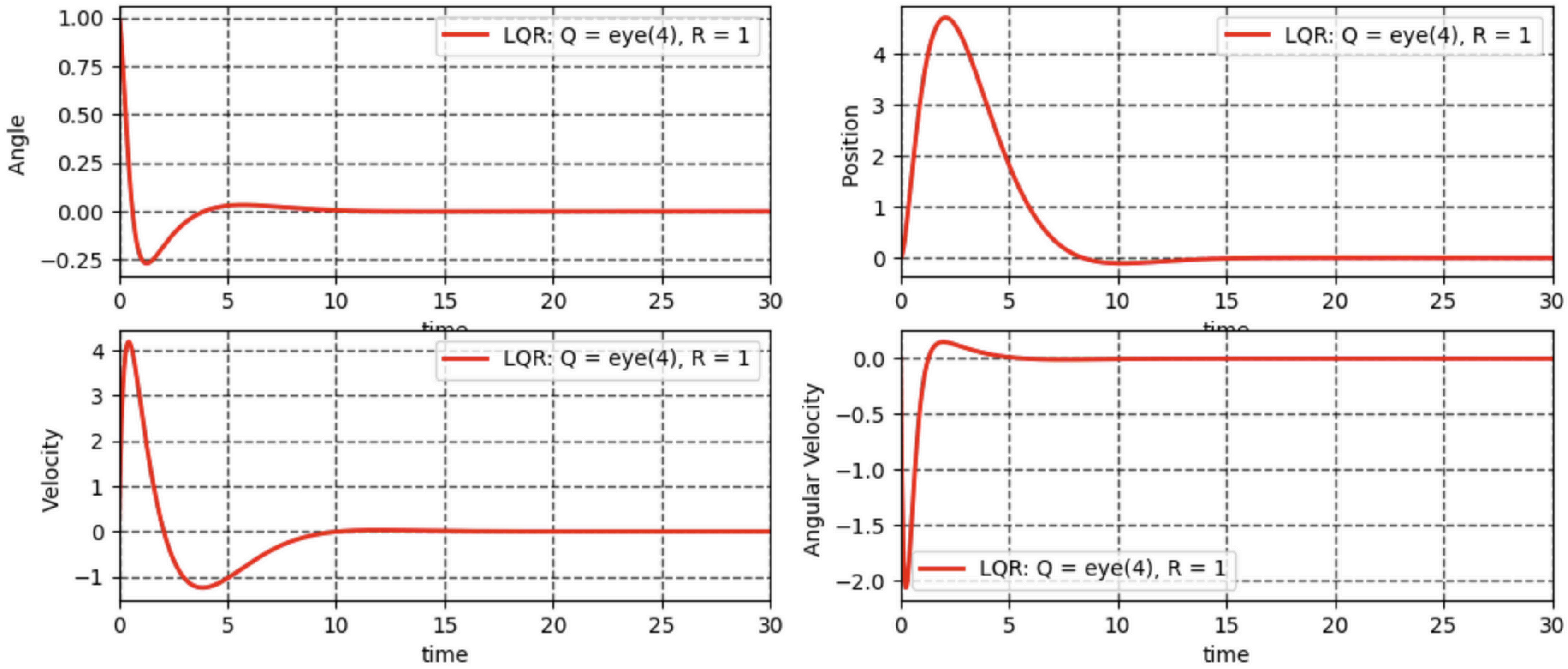
Cart-pole control. PID.





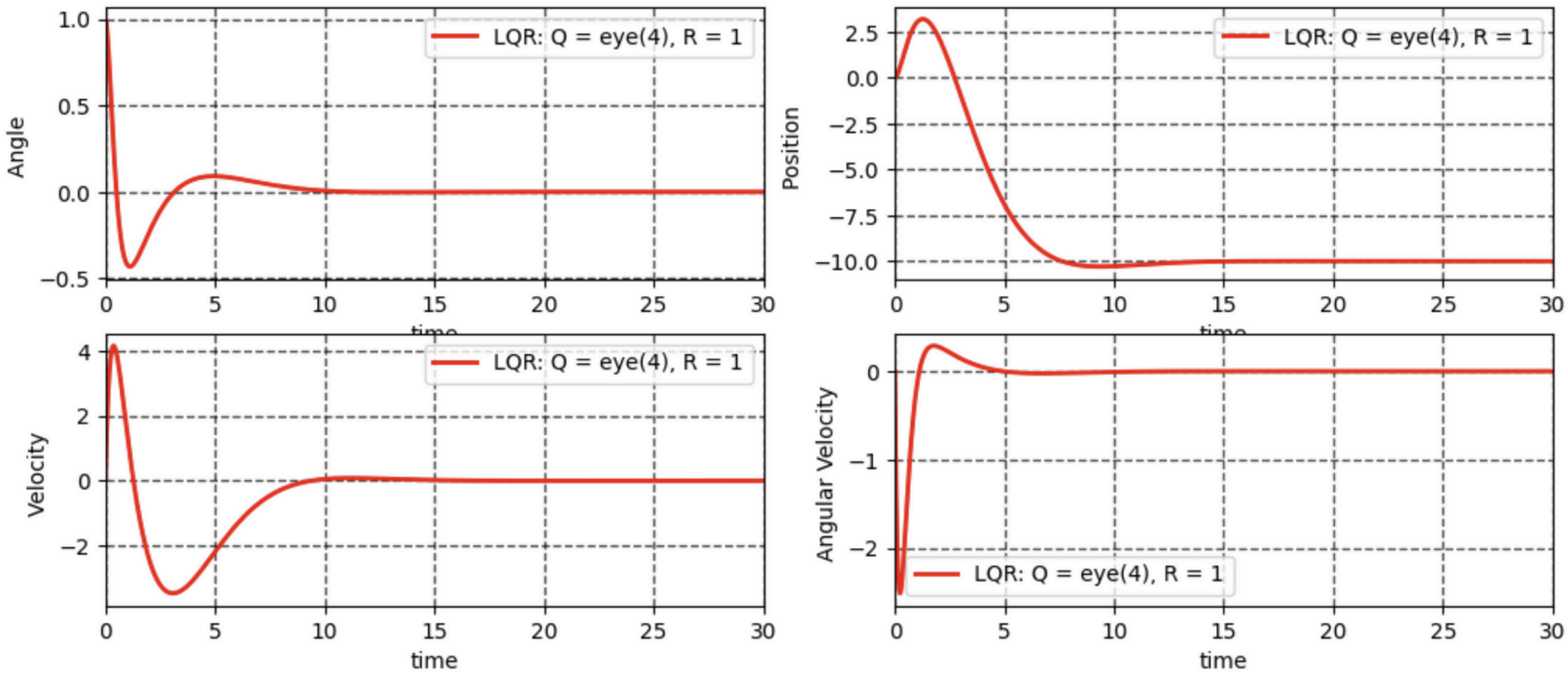
$w(t) = 0.1, \quad x_0 = (0, 0, 1, 0)$

Cart-pole control. LQR



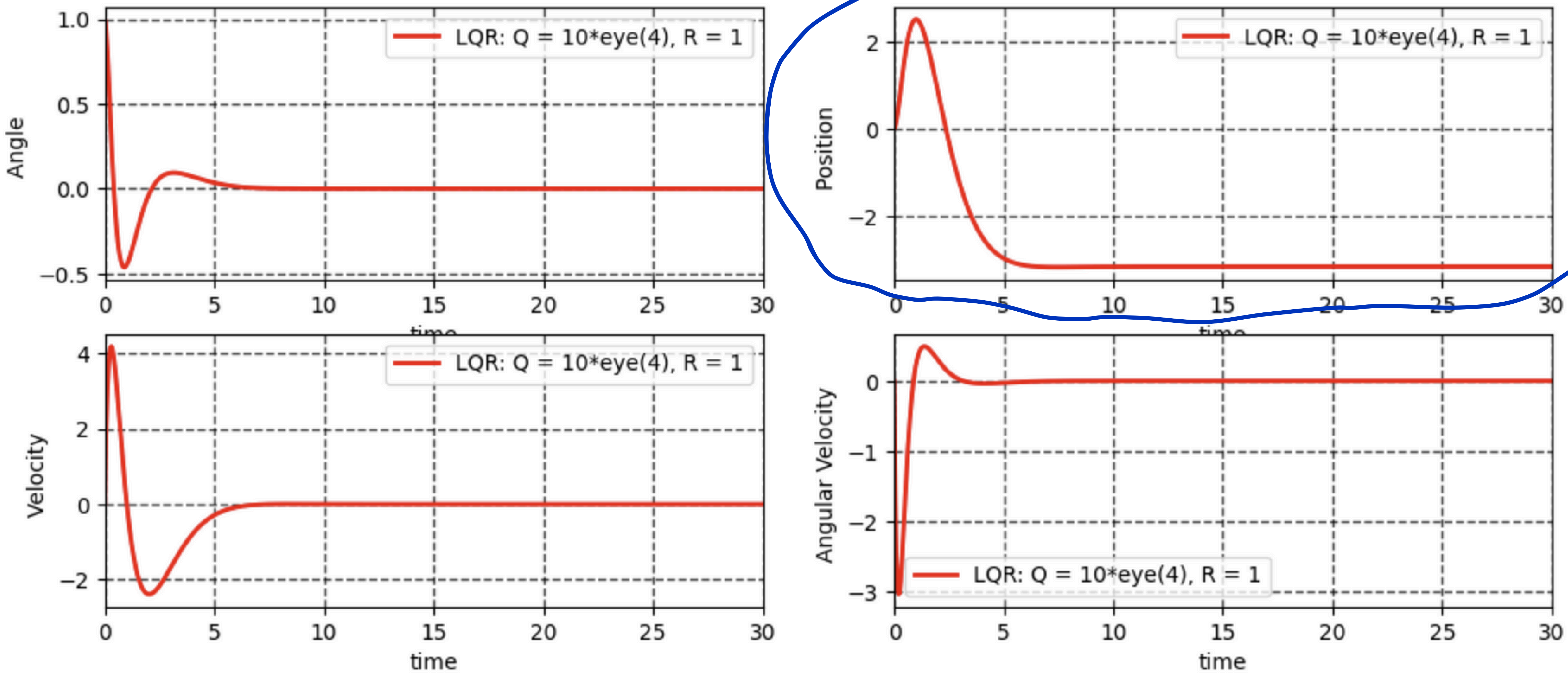
$w(t) = 0.1, \quad x_0 = (0, 0, 1, 0)$

Cart-pole control. LQR



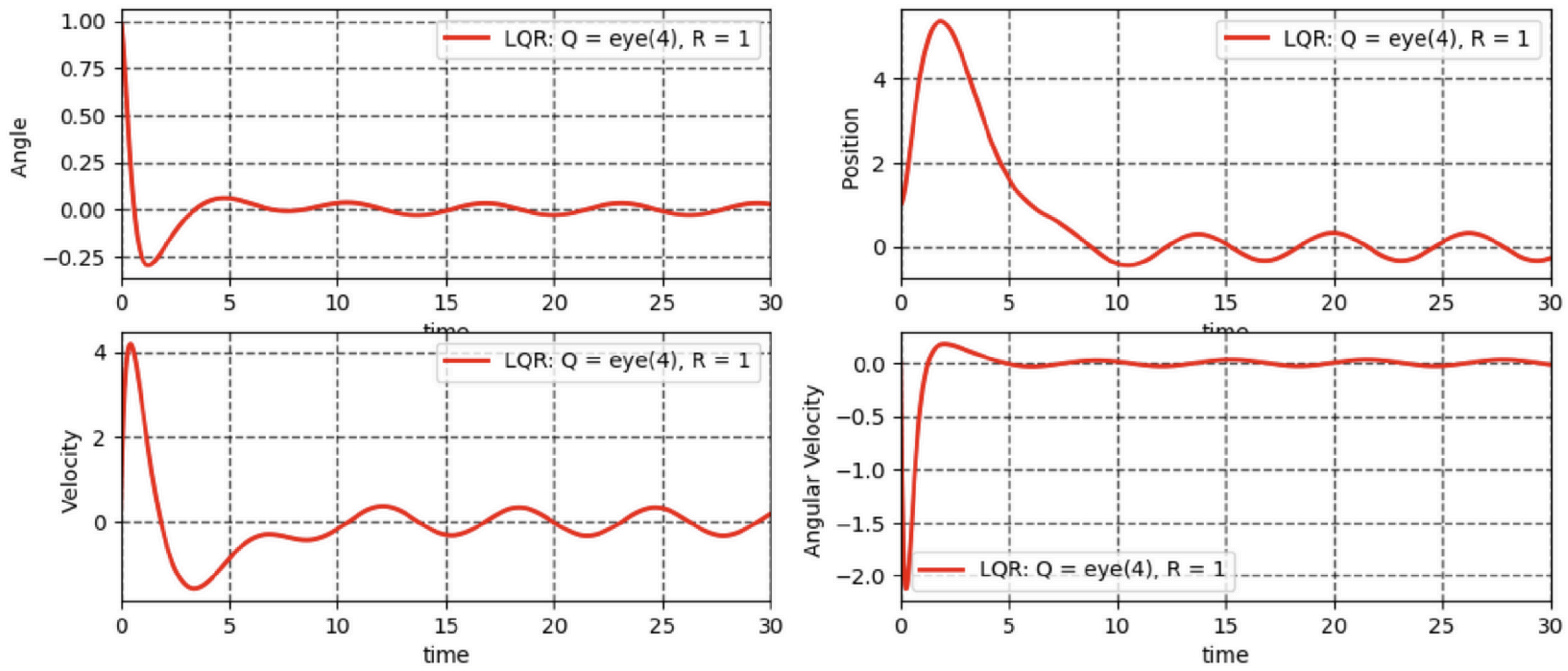
$w(t) = 10, \quad x_0 = (0, 0, 1, 0)$

Cart-pole control. LQR



$w(t) = 10$, $x_0 = (0, 0, 1, 0)$

Cart-pole control.
Linear full state feedback controller.



$w(t) = \sin(t)$, $x(0) = (1, 0, 1, 0)$

Stabilisation

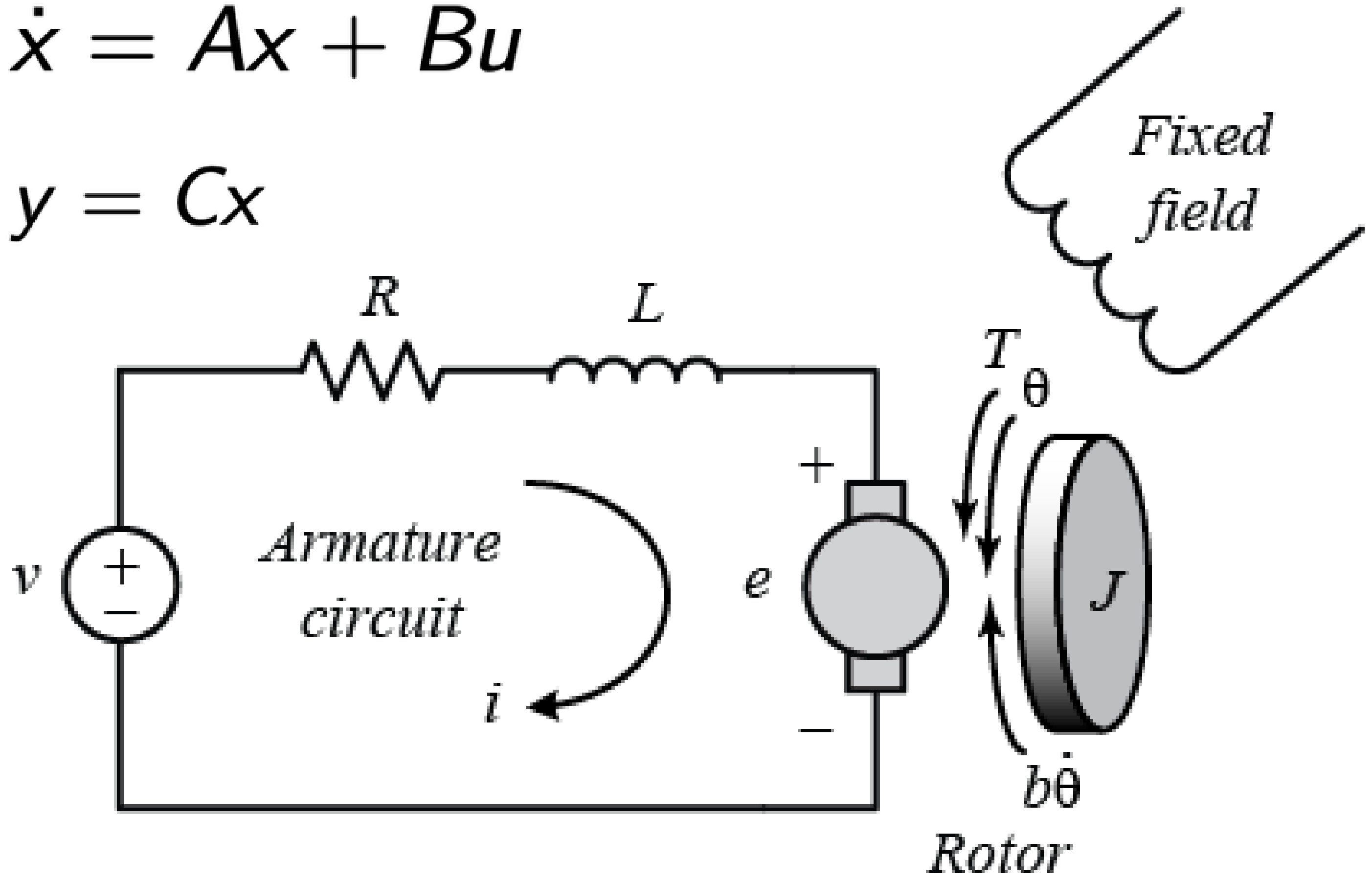
vs

Reference tracking

DC motor control design

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

$$y = Cx$$



A common actuator in control systems is the DC motor. It directly provides rotary motion and, coupled with wheels or drums and cables, can provide translational motion.

$$A = \begin{pmatrix} -\frac{b}{J} & \frac{K}{J} \\ -\frac{K}{L} & -\frac{R}{L} \end{pmatrix}, \quad B = \begin{pmatrix} 0 \\ \frac{1}{L} \end{pmatrix}$$

$$C = (1, 0), \, x = \begin{pmatrix} \dot{\theta} \\ i \end{pmatrix}, \, r(t) = 1 \; rad/sec$$

Reference tracking

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

$y = x$

Design a linear full state feedback controller u = -Kx such that

$x(t) \to x_{ref}(t)$

robustly to any initial condition $x(0) = x_0$,

and any disturbance $w(t)$

Reference tracking

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

$y = x$

add another term

Design a linear full state feedback controller u = -K(x - x_ref) s.t.

$x(t) \to x_{ref}(t)$

robustly to any initial condition $x(0) = x_0$,

and any disturbance $w(t)$

Reference tracking

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

substitute

$y = x$

Design a linear full state feedback controller u = -K(x - x_ref) s.t.

$x(t) \to x_{ref}(t)$

robustly to any initial condition $x(0) = x_0$,

and any disturbance $w(t)$

Reference tracking

$$\dot{x} = (A - BK)x + \underbrace{BKx_{ref}}_{\text{acts as a disturbance}} + Dw$$

The faster non disturbed system converges to zero, the better it tracks a reference trajectory.

Unfortunately, the faster it converges, the more energy is required.

The precise tracking of x_ref is not guaranteed

Reference tracking

$$\dot{x} = Ax + Bu + Dw$$, n - states

$y = x$ q - controls

Let us assume that

1) $x\_ref = (x^0_{ref}, \ldots, x^{n-1}_{ref})$ is constant

2) number of non-zero elements in x_ref is less (or equal) than number of control inputs

Robust tracking: integral action

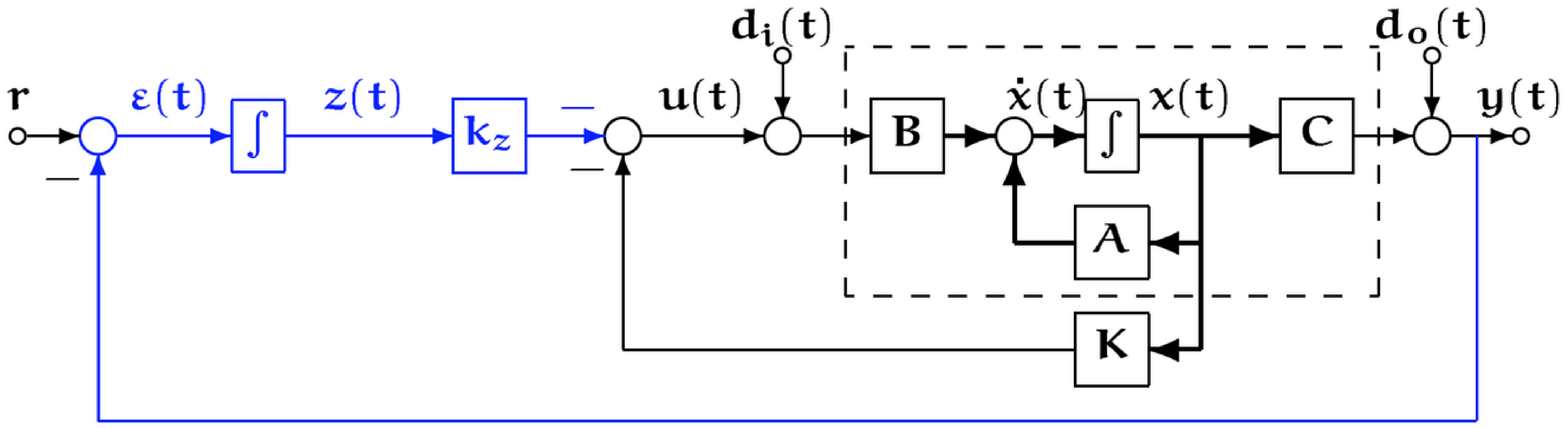
We now introduce a **robust** approach to achieve constant reference tracking by state feedback. This approach consists in the **addition of integral action** to the state feedback, so that

- the error $\varepsilon(t) = r - y(t)$ will approach 0 as $t \to \infty$, and this property will be preserved
  - under moderate uncertainties in the plant model
  - under constant input or output disturbance signals.

Let’s start with single input, single non - zero constant reference to track

Robust tracking: integral action

The State Feedback with Integral Action scheme:



The main idea in the addition of integral action is to **augment the plant** with an extra state: the integral of the tracking error $\varepsilon(t)$,

— corresponding state

$$\dot{z}(t) = r - y(t) = r - Cx(t) \tag{IA1}$$

non zero element of x-ref

The control law for the **augmented plant** is then

$$u(t) = -\begin{bmatrix} K & k_z \end{bmatrix} \begin{bmatrix} x(t) \\ z(t) \end{bmatrix} \tag{IA2}$$

Robust tracking: integral action

$$\begin{bmatrix} \dot{x}(t) \\ \dot{z}(t) \end{bmatrix} = \underbrace{\begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix}}_{A_a} \begin{bmatrix} x(t) \\ z(t) \end{bmatrix} - \underbrace{\begin{bmatrix} B \\ 0 \end{bmatrix}}_{B_a} \underbrace{\begin{bmatrix} K & k_z \end{bmatrix}}_{K_a} \begin{bmatrix} x(t) \\ z(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} r$$

$$= (A_a - B_a K_a) \begin{bmatrix} x(t) \\ z(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} r$$

The state feedback design with integral action can be done as a normal state feedback design for the **augmented plant**

If $\mathbf{K}_a$ is designed such that the closed-loop augmented matrix $(\mathbf{A}_a - \mathbf{B}_a \mathbf{K}_a)$ is rendered Hurwitz, then necessarily in steady-state

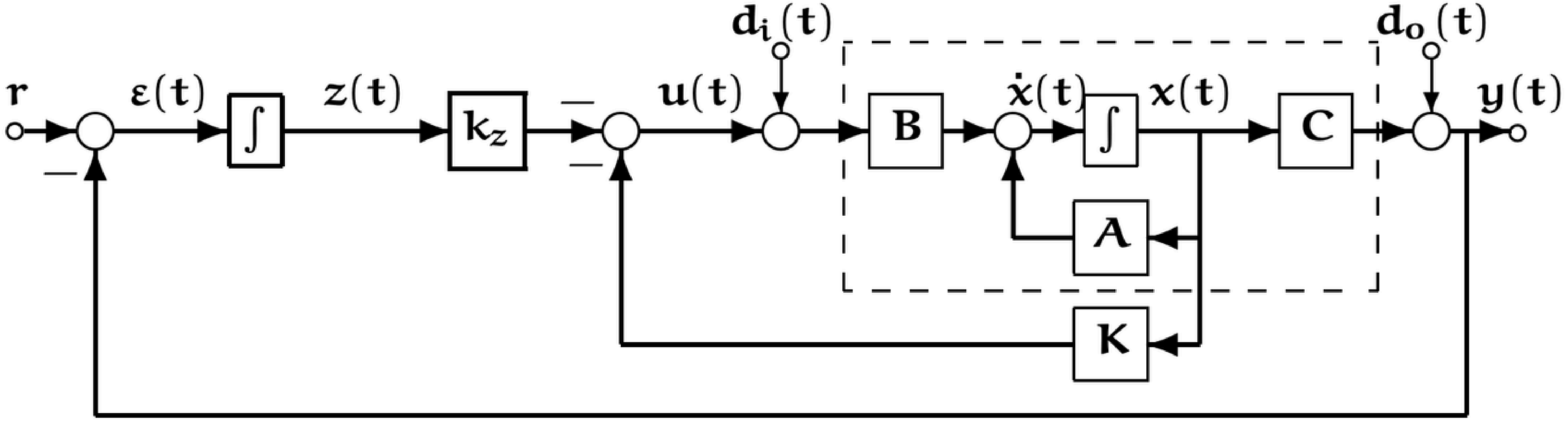
$$\lim_{t \to \infty} \dot{z}(t) = 0 \quad \Rightarrow \quad \lim_{t \to \infty} y(t) = r, \quad \text{achieving tracking.}$$

Robust tracking: integral action for MIMO system

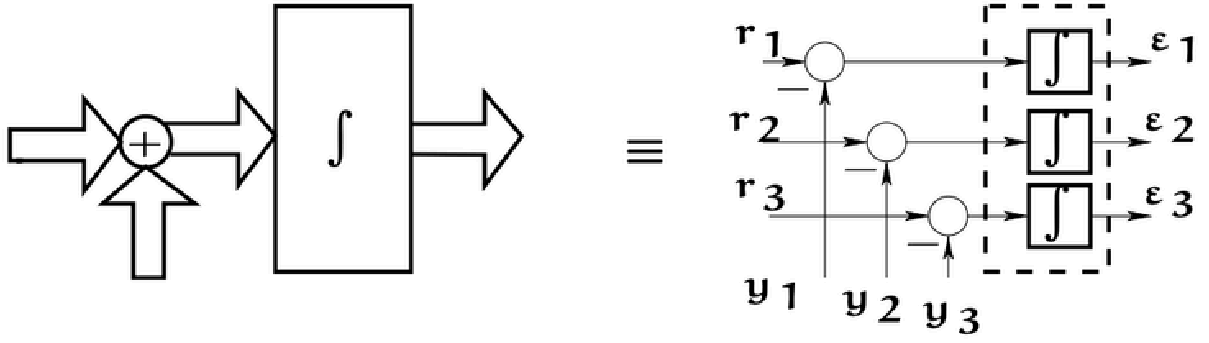
Robust tracking for MIMO system

Tracking with Integral Action is subject to the same restrictions: **we can only achieve asymptotic tracking of a maximum of as many outputs as control inputs are available.**

The scheme and computation procedure is the same as in SISO



Note that now the **integral action** is applied to each of the q reference input channels.



Robust tracking for MIMO system

The procedure to compute $\mathbf{K}$ and $\mathbf{k_z}$ for the state feedback control with integral action is exactly as in the SISO case,

corresponding state vector

$\dim(z) \leq q \rightarrow$

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

non zero elements of x-vec

$$\mathbf{u}(t) = \begin{bmatrix} \mathbf{K} & \mathbf{k_z} \end{bmatrix} \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{z}(t) \end{bmatrix}$$

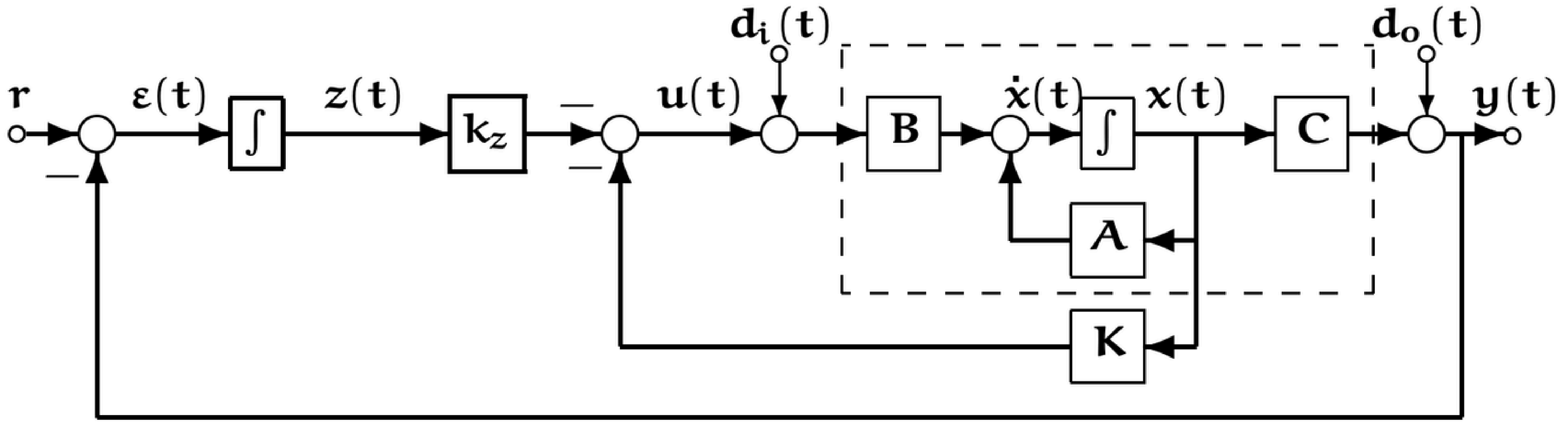
where $\mathbf{K_a} = [\,\mathbf{K}\ \mathbf{k_z}\,]$ is computed to place the eigenvalues of the **augmented plant** $(\mathbf{A_a}, \mathbf{B_a})$ at desired locations, where

$$\mathbf{A_a} = \begin{bmatrix} \mathbf{A} & 0_{n\times q} \\ -\mathbf{C} & 0_{q\times q} \end{bmatrix}, \quad \mathbf{B_a} = \begin{bmatrix} \mathbf{B} \\ 0_{q\times p} \end{bmatrix}$$

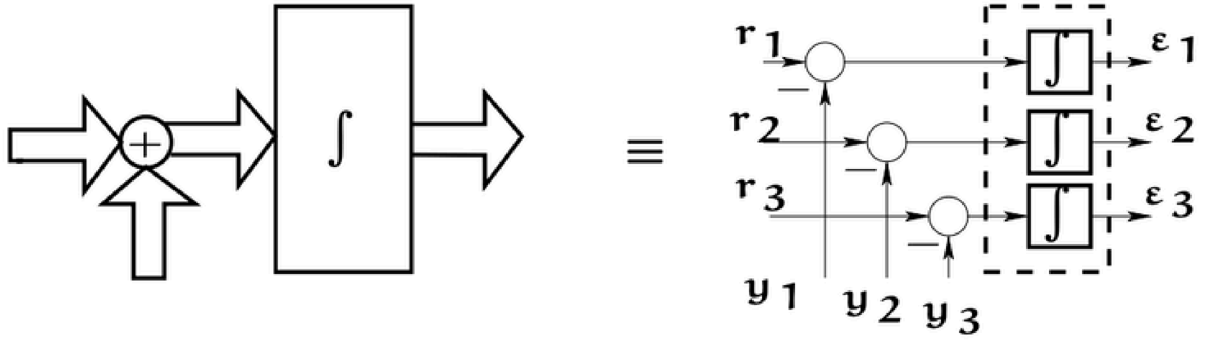
Robust tracking for MIMO system

Tracking with Integral Action is subject to the same restrictions: **we can only achieve asymptotic tracking of a maximum of as many outputs as control inputs are available.**

The scheme and computation procedure is the same as in SISO



Note that now the **integral action** is applied to each of the q reference input channels.



What happens if system is not controllable?

What happens if system is not controllable?

We have seen that if a state equation is controllable, then we can assign its eigenvalues arbitrarily by state feedback. But, what happens when the state equation is **not** controllable?

We know that we can take any state equation to the Controllable/Uncontrollable Canonical Form

$$\begin{bmatrix} \dot{\bar{\mathbf{x}}}_{\mathcal{C}} \\ \dot{\bar{\mathbf{x}}}_{\tilde{\mathcal{C}}} \end{bmatrix} = \underbrace{\begin{bmatrix} \bar{\mathbf{A}}_{\mathcal{C}} & \bar{\mathbf{A}}_{12} \\ 0 & \bar{\mathbf{A}}_{\tilde{\mathcal{C}}} \end{bmatrix}}_{\bar{\mathbf{A}}} \begin{bmatrix} \bar{\mathbf{x}}_{\mathcal{C}} \\ \bar{\mathbf{x}}_{\tilde{\mathcal{C}}} \end{bmatrix} + \begin{bmatrix} \bar{\mathbf{B}}_{\mathcal{C}} \\ 0 \end{bmatrix} u$$

Because the evolution matrix $\bar{\mathbf{A}}$ is **block-triangular**, its eigenvalues are the union of the eigenvalues of the diagonal blocks: $\bar{\mathbf{A}}_{\mathcal{C}}$ and $\bar{\mathbf{A}}_{\tilde{\mathcal{C}}}$.

What happens if system is not controllable?

The state feedback law

$$\begin{aligned} \mathbf{u} &= \mathbf{r} - \mathbf{K}\mathbf{x} \\ &= \mathbf{r} - \bar{\mathbf{K}}\bar{\mathbf{x}} \\ &= \mathbf{r} - \begin{bmatrix} \bar{\mathbf{K}}_{c} & \bar{\mathbf{K}}_{\tilde{c}} \end{bmatrix} \begin{bmatrix} \dot{\bar{\mathbf{x}}}_{c} \\ \dot{\bar{\mathbf{x}}}_{\tilde{c}} \end{bmatrix} \end{aligned}$$

yields the closed-loop system

$$\begin{bmatrix} \dot{\bar{\mathbf{x}}}_{c} \\ \dot{\bar{\mathbf{x}}}_{\tilde{c}} \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{A}}_{c} - \bar{\mathbf{B}}_{c}\bar{\mathbf{K}}_{c} & \bar{\mathbf{A}}_{12} - \bar{\mathbf{B}}_{c}\bar{\mathbf{K}}_{\tilde{c}} \\ 0 & \bar{\mathbf{A}}_{\tilde{c}} \end{bmatrix} \begin{bmatrix} \bar{\mathbf{x}}_{c} \\ \bar{\mathbf{x}}_{\tilde{c}} \end{bmatrix} + \begin{bmatrix} \bar{\mathbf{B}}_{c} \\ 0 \end{bmatrix} \mathbf{r}.$$

We see that the eigenvalues of $\bar{\mathbf{A}}_{\tilde{c}}$ are **not** affected by the state feedback, so they remain **unchanged**.

The value of $\bar{\mathbf{K}}_{\tilde{c}}$ is **irrelevant** — the uncontrollable states cannot be affected.

What happens if system is not controllable?

We conclude that the condition of **Controllability** is not only sufficient, but also necessary to place **all** eigenvalues of $\mathbf{A} - \mathbf{BK}$ in desired locations.

A notion of interest in control that is weaker than that of **Controllability** is that of Stabilisability.

Stabilisability. The system

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t), \end{aligned}$$

is said to be **stabilisable** if all its **uncontrollable states are asymptotically stable**.

This condition is equivalent to asking that the matrix $\bar{\mathbf{A}}_{\tilde{c}}$ be **Hurwitz**.