A systems approach to fluid dynamics: input-output models

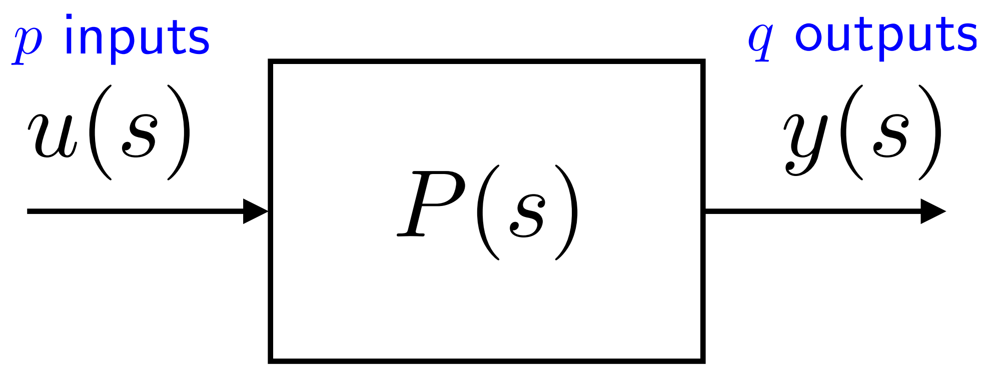
UNH/IAM Workshop
November 2015

Simon Illingworth
University of Melbourne

overview

terminology

- frequency domain

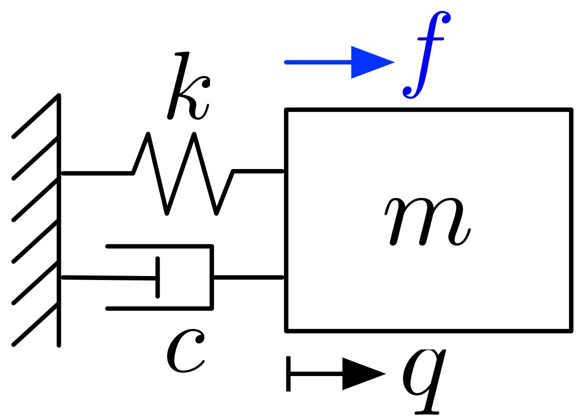


$$\underset{q\times 1}{y(s)} = \underset{q\times p}{P(s)}\,\underset{p\times 1}{u(s)}$$

- time domain: we will sometimes need the state-space description:

$$\dot{x}(t) = A\,x(t) + B\,u(t)$$
$$y(t) = C\,x(t) + D\,u(t)$$

frequency response: SISO case



$$m\ddot{q}(t) + c\dot{q}(t) + kq(t) = f(t)$$

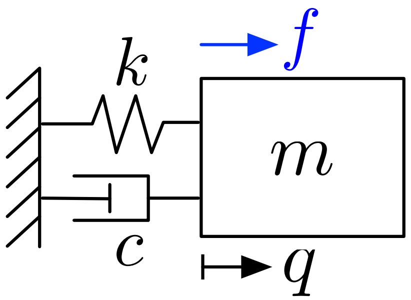
$\mathcal{L}$

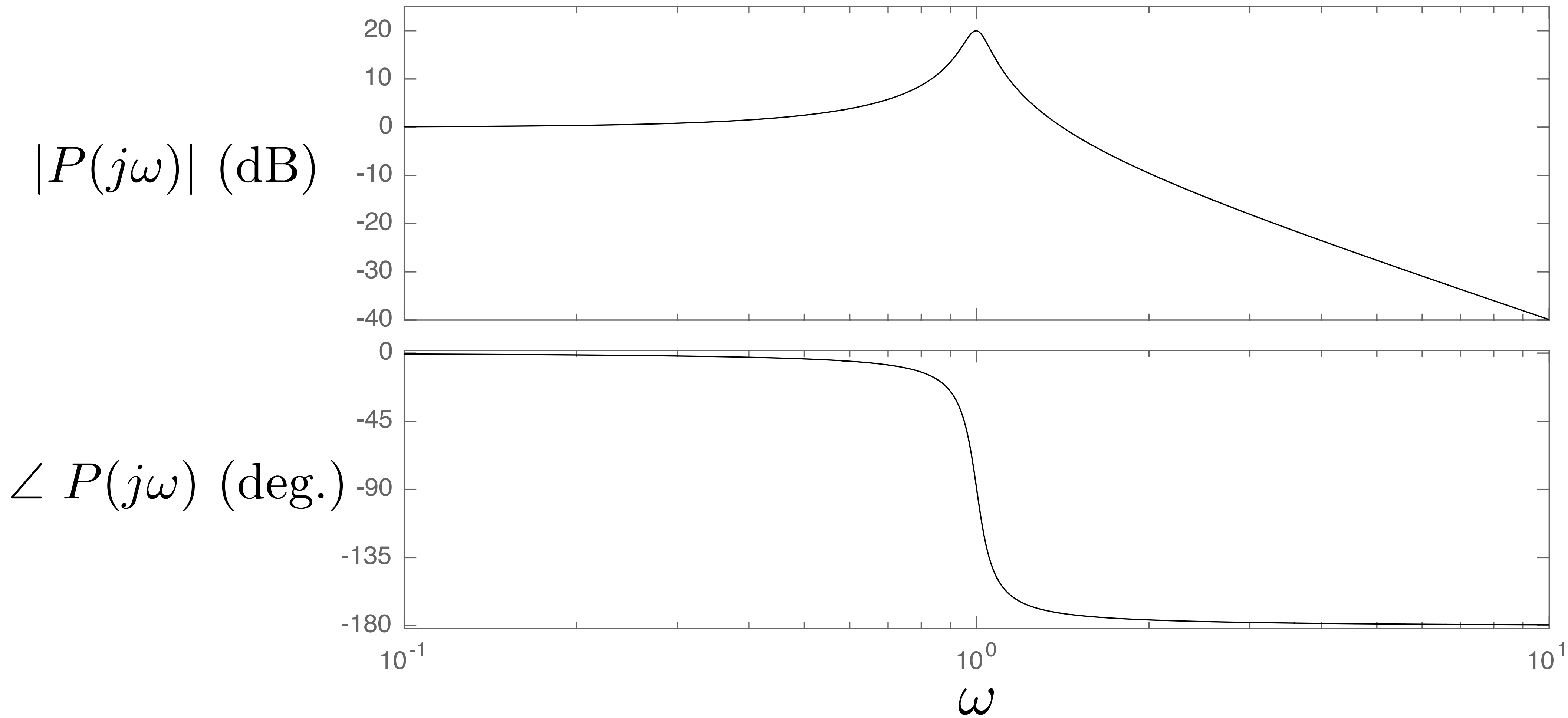
$$P(s) = \frac{Q(s)}{F(s)} = \frac{1}{ms^2 + cs + k}$$

$s = j\omega$

$$P(j\omega) = \frac{1}{(k - m\omega^2) + jc\,\omega}$$

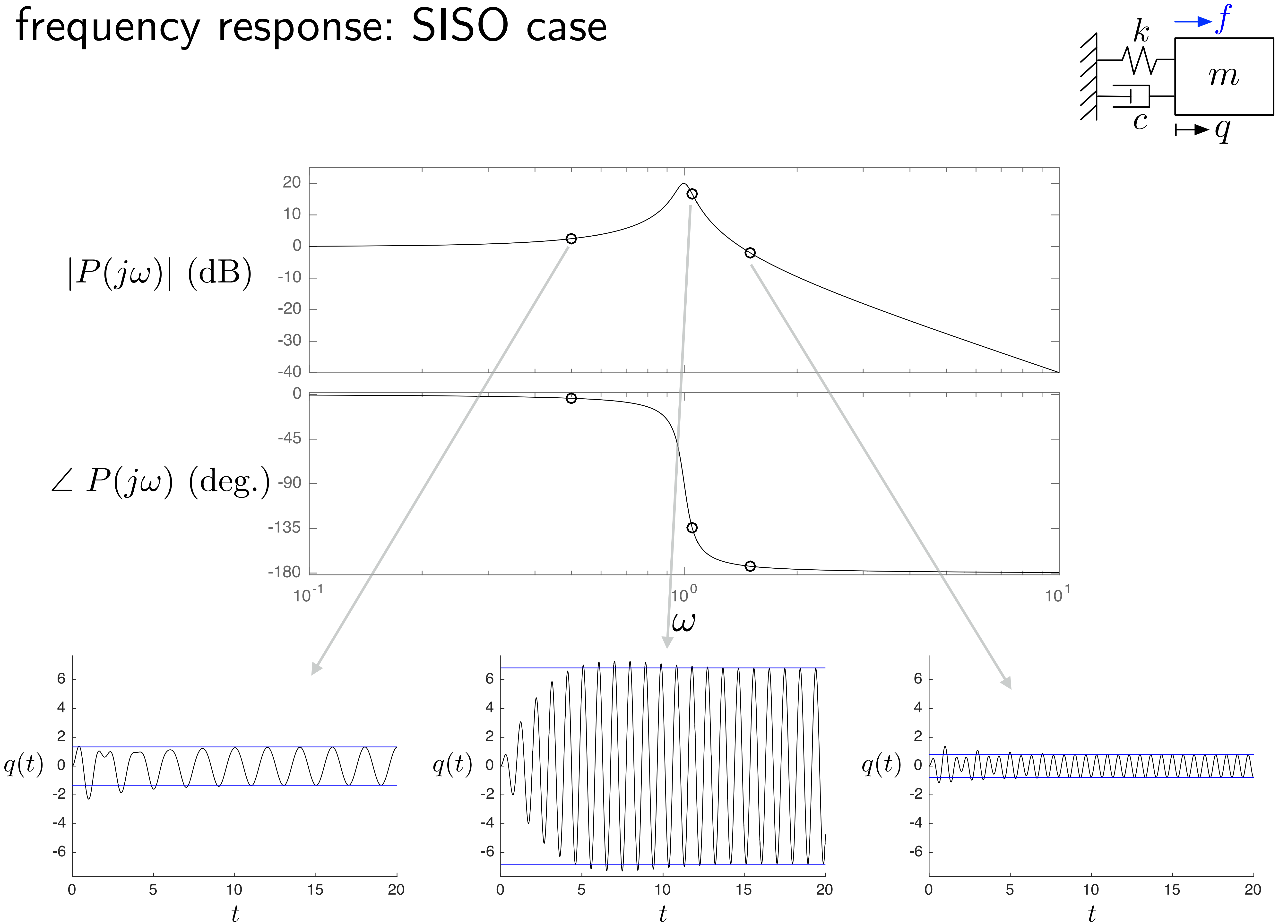
frequency response: SISO case



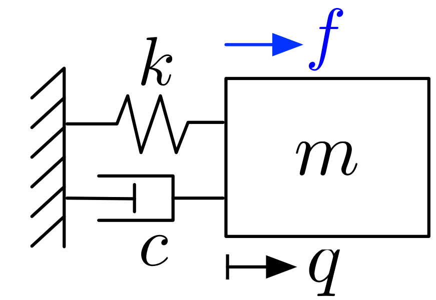


$$P(j\omega) = \frac{1}{(k - m\omega^2) + jc\omega}$$

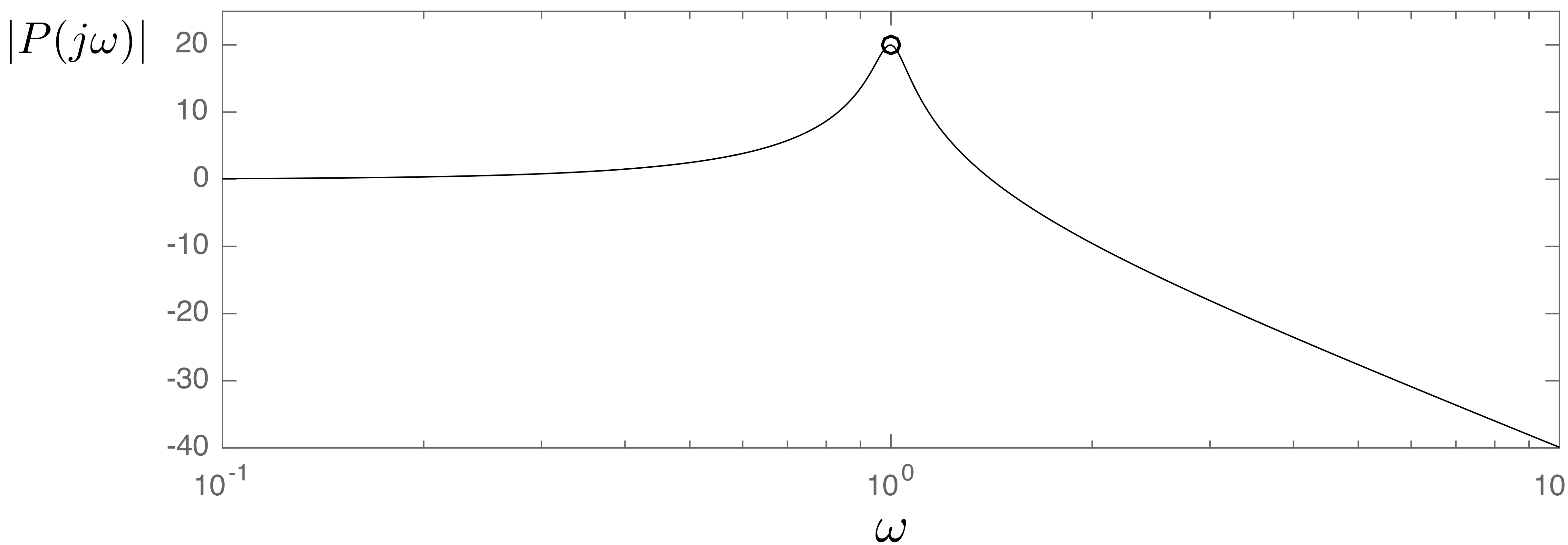
frequency response: SISO case



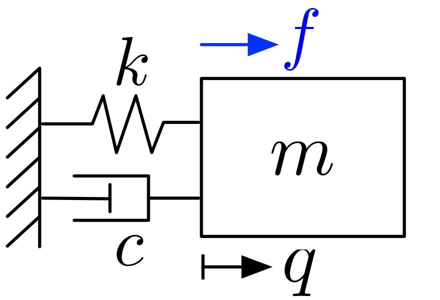
the ∞-norm: SISO case



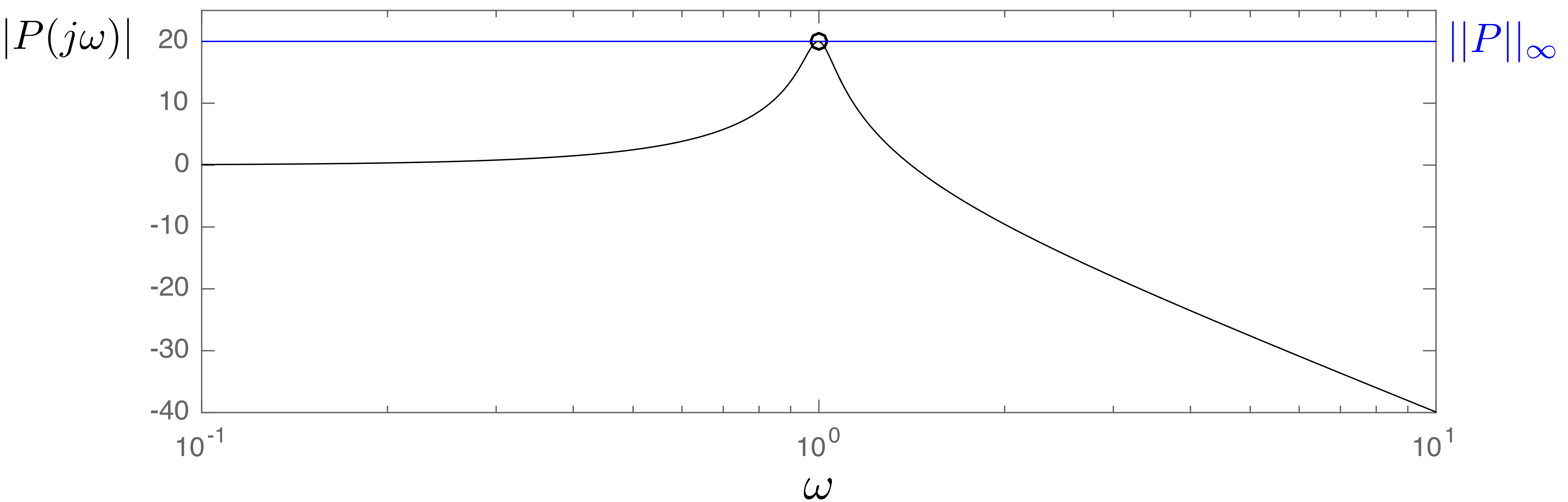
$$||P||_\infty = \max_\omega \ P(j\omega)$$



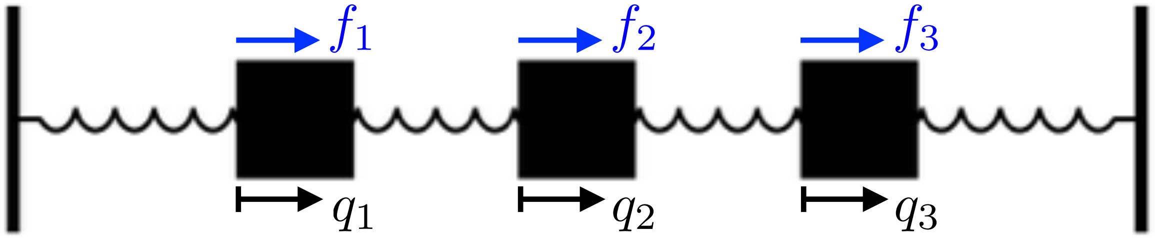
the ∞-norm: SISO case



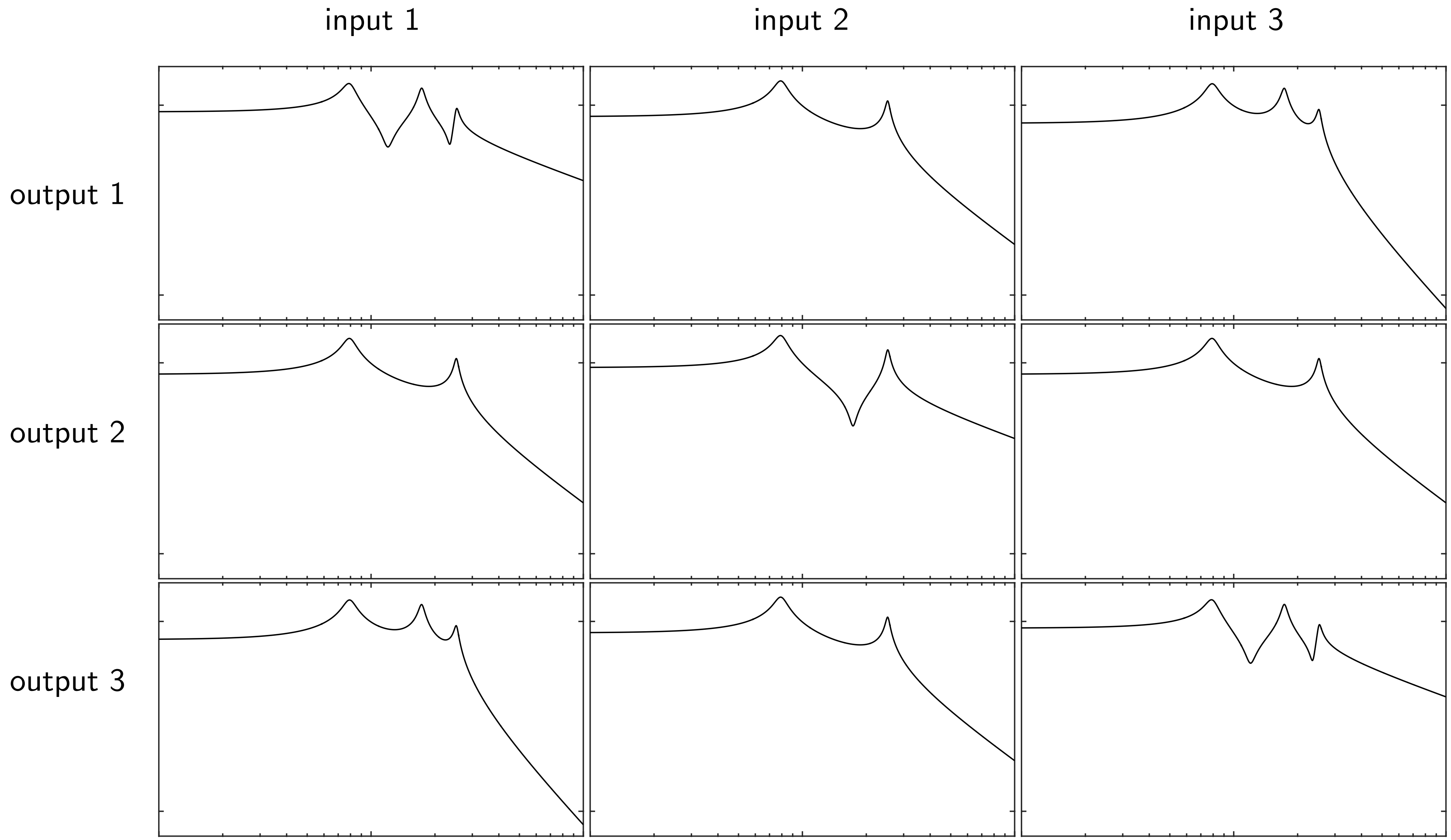
$$||P||_\infty = \max_\omega \; P(j\omega)$$

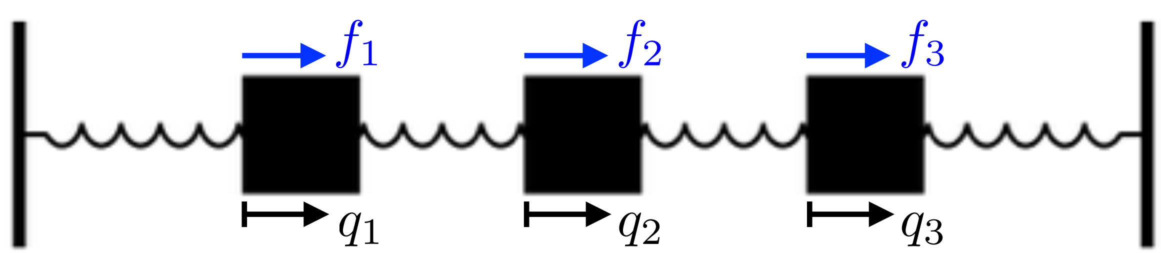


frequency response: MIMO case

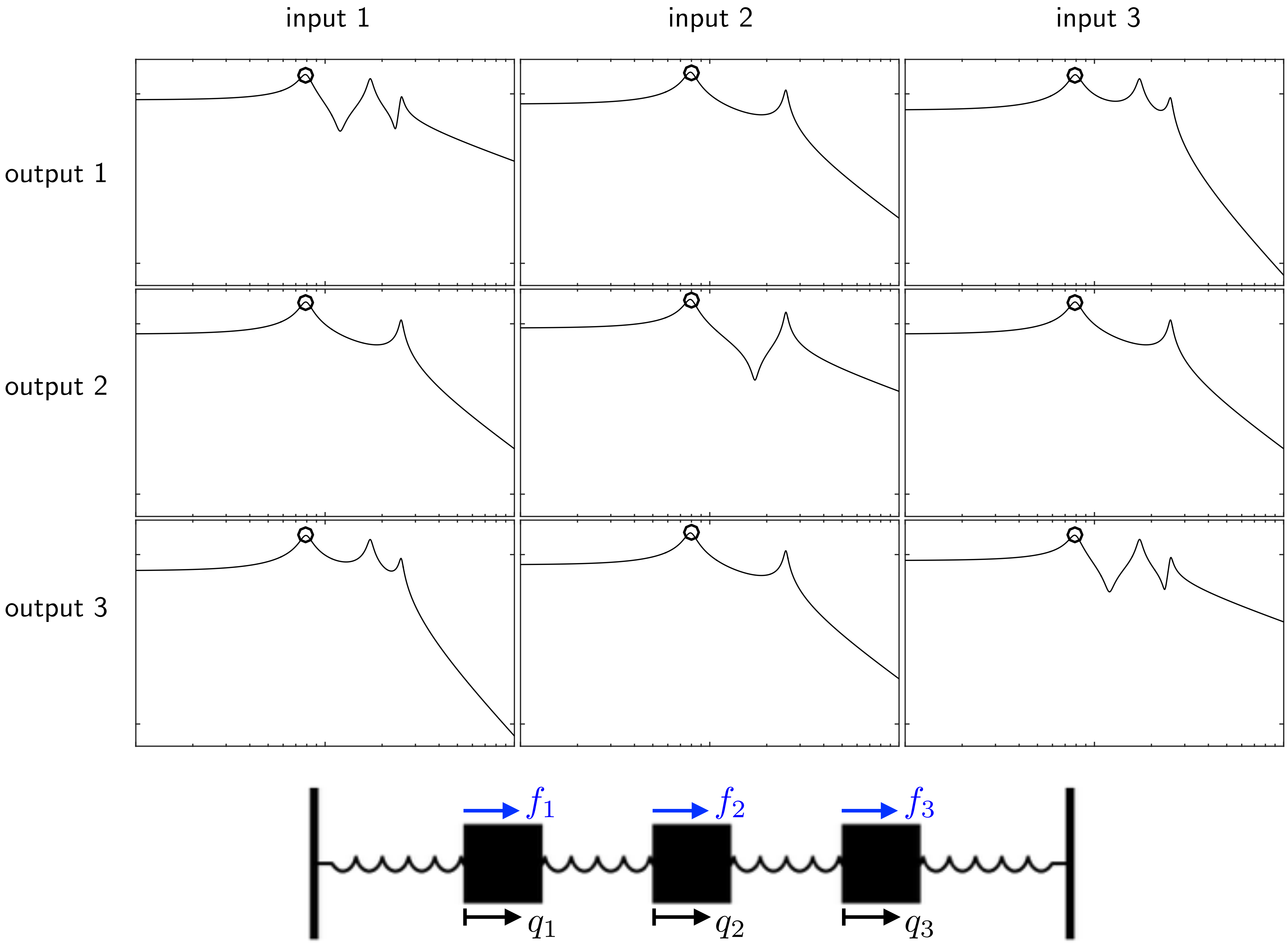


frequency response: MIMO case

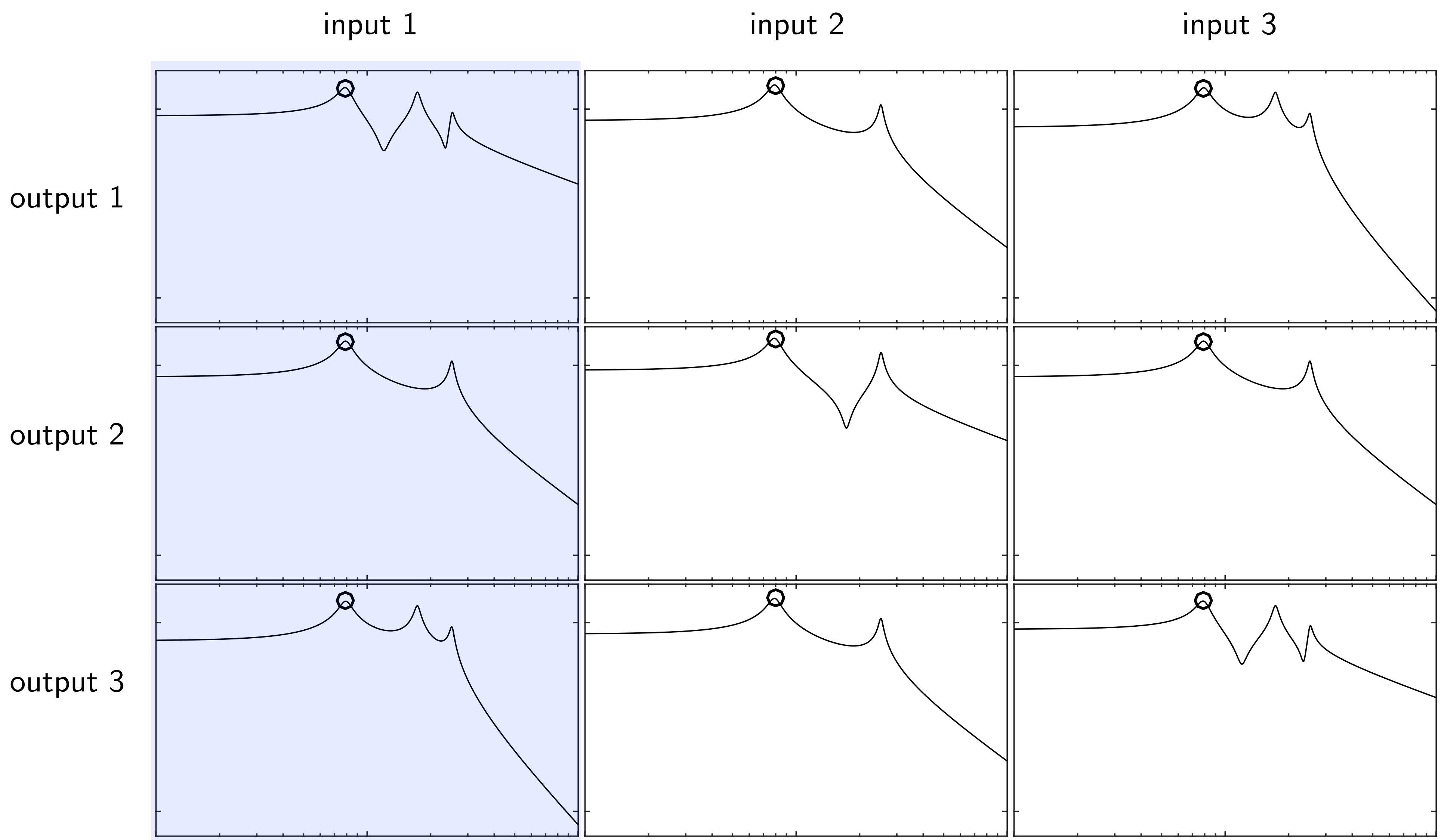


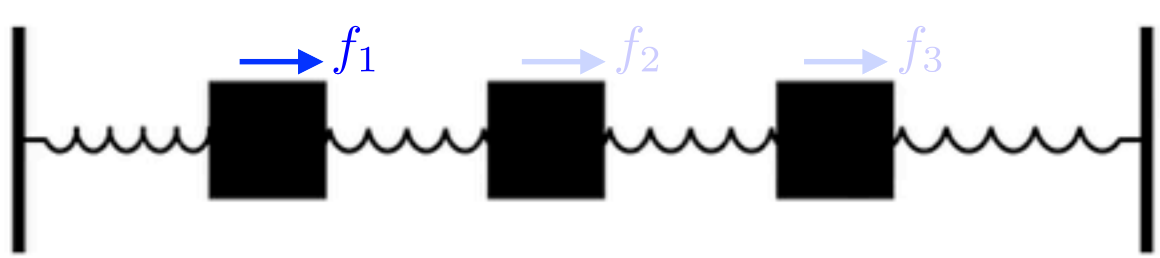


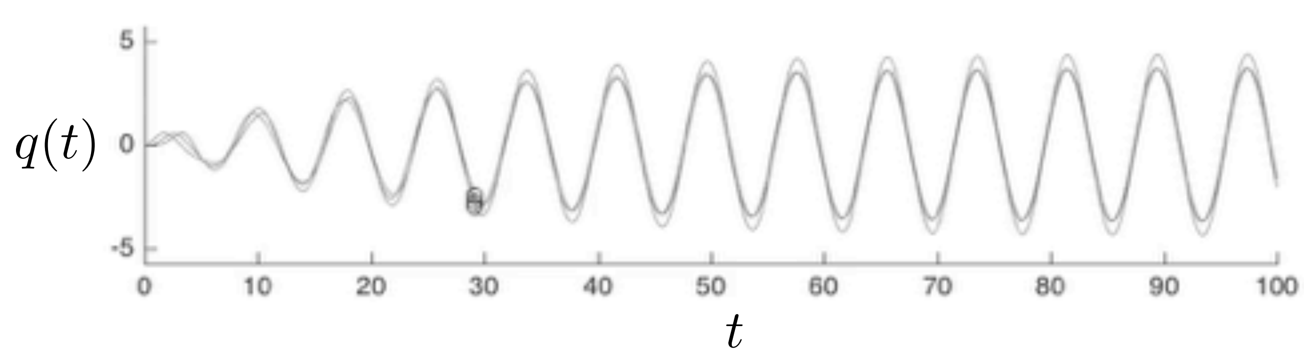
frequency response: MIMO case



frequency response: MIMO case

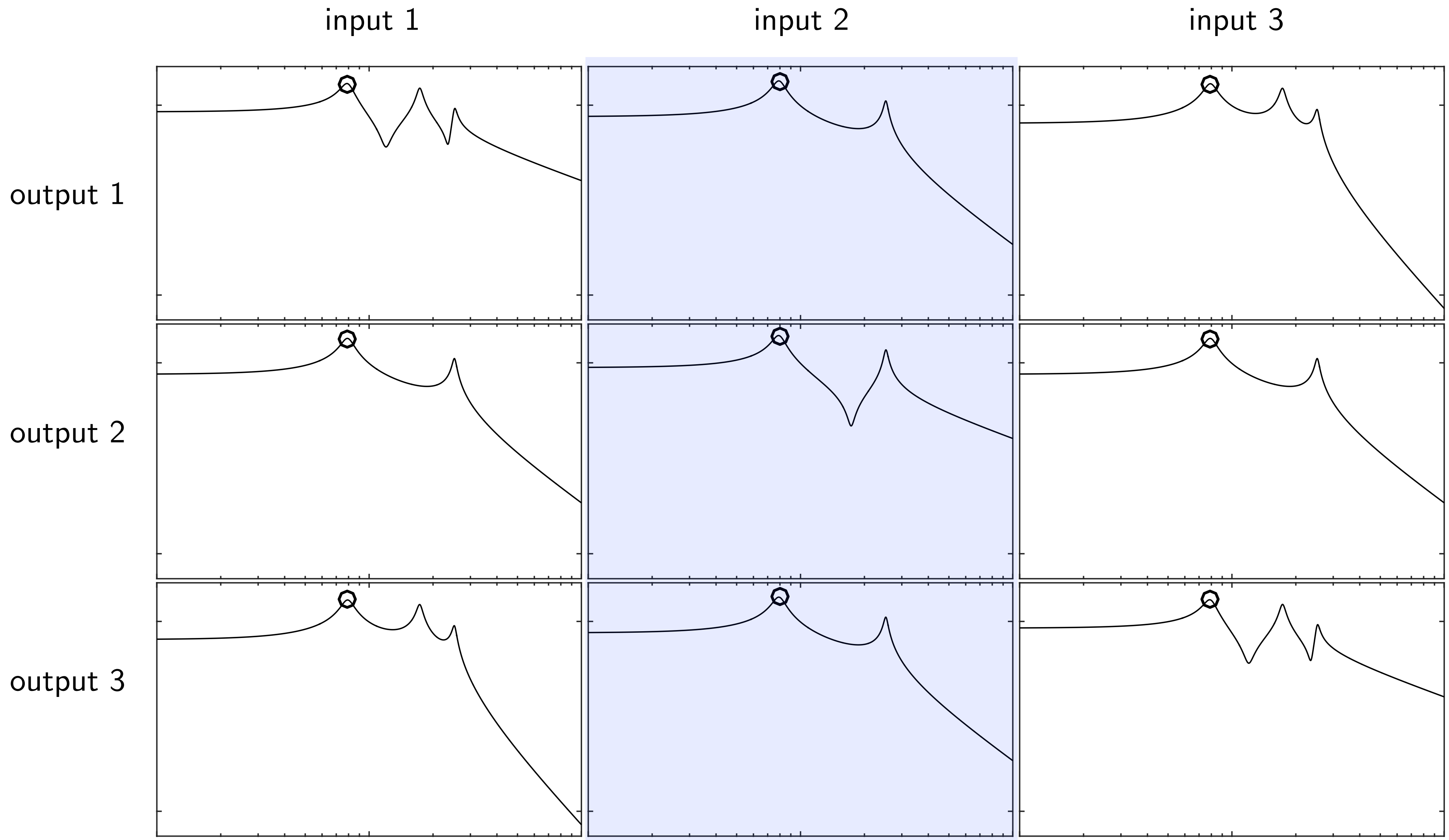


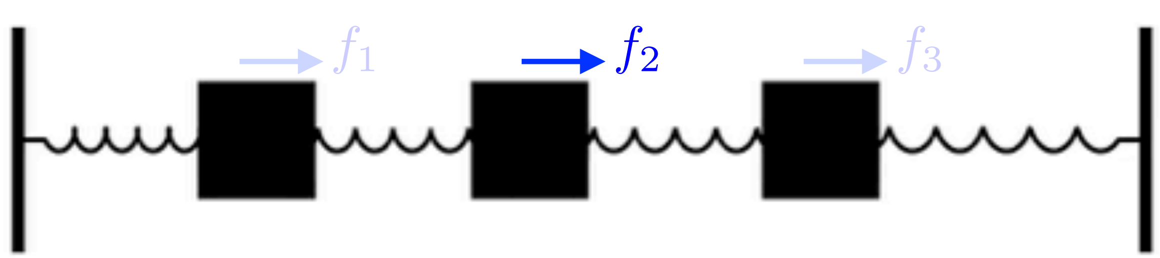


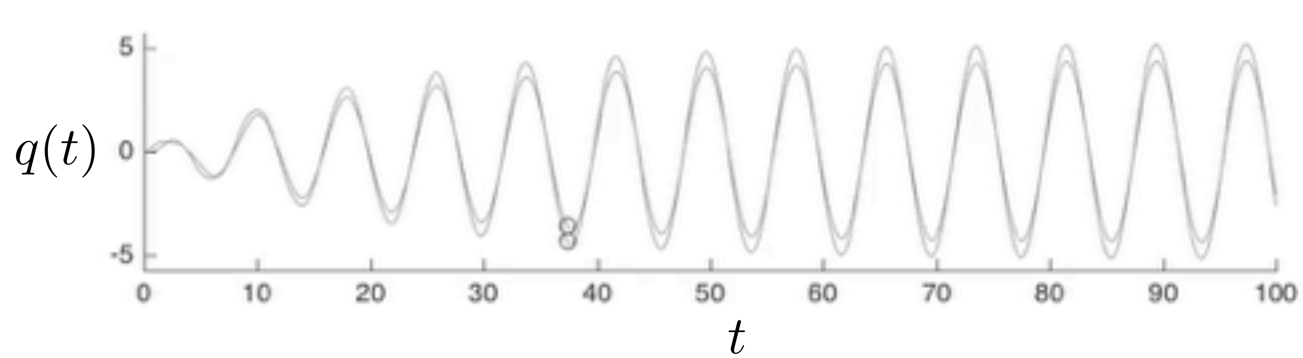


$$f(t) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \cos(\omega_f t)$$

frequency response: MIMO case

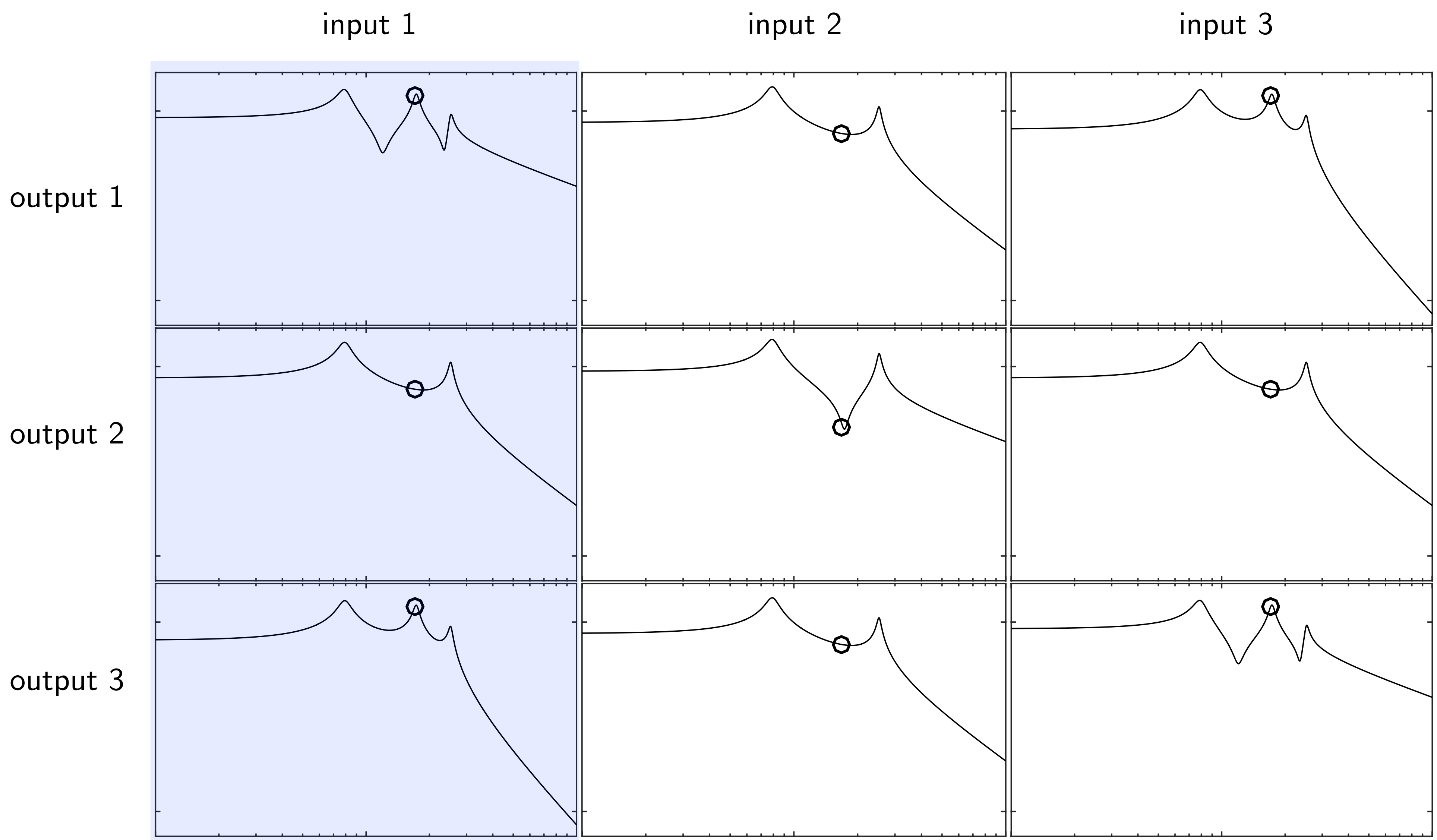


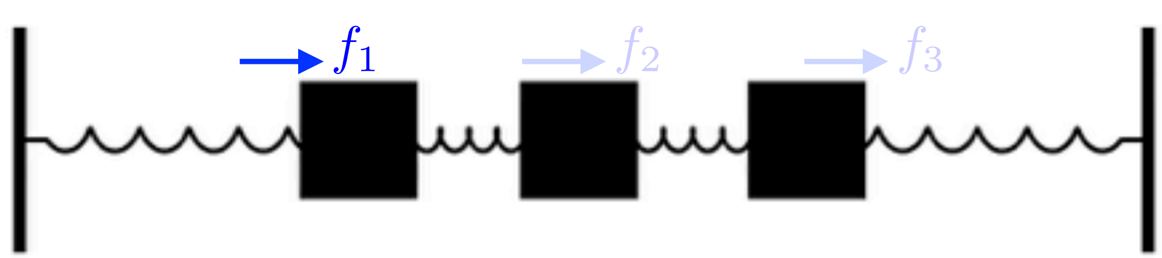


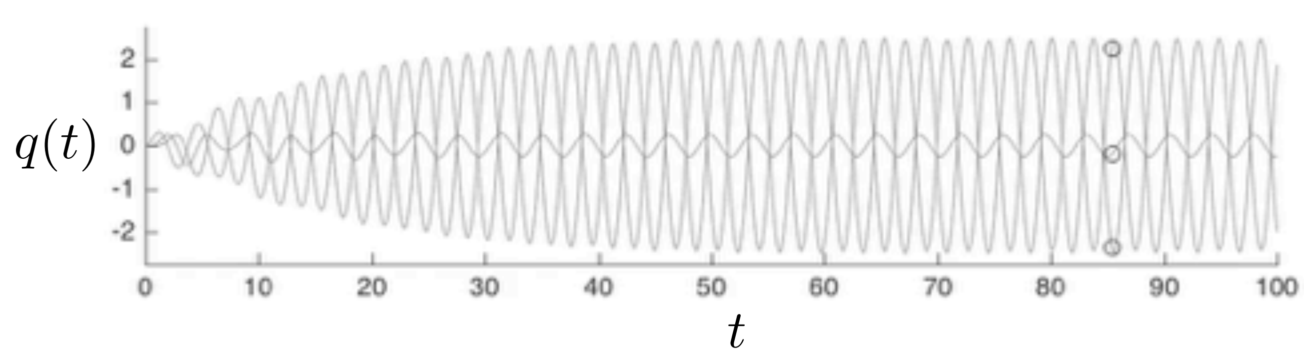


$$f(t) = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \cos(\omega_f t)$$

frequency response: MIMO case

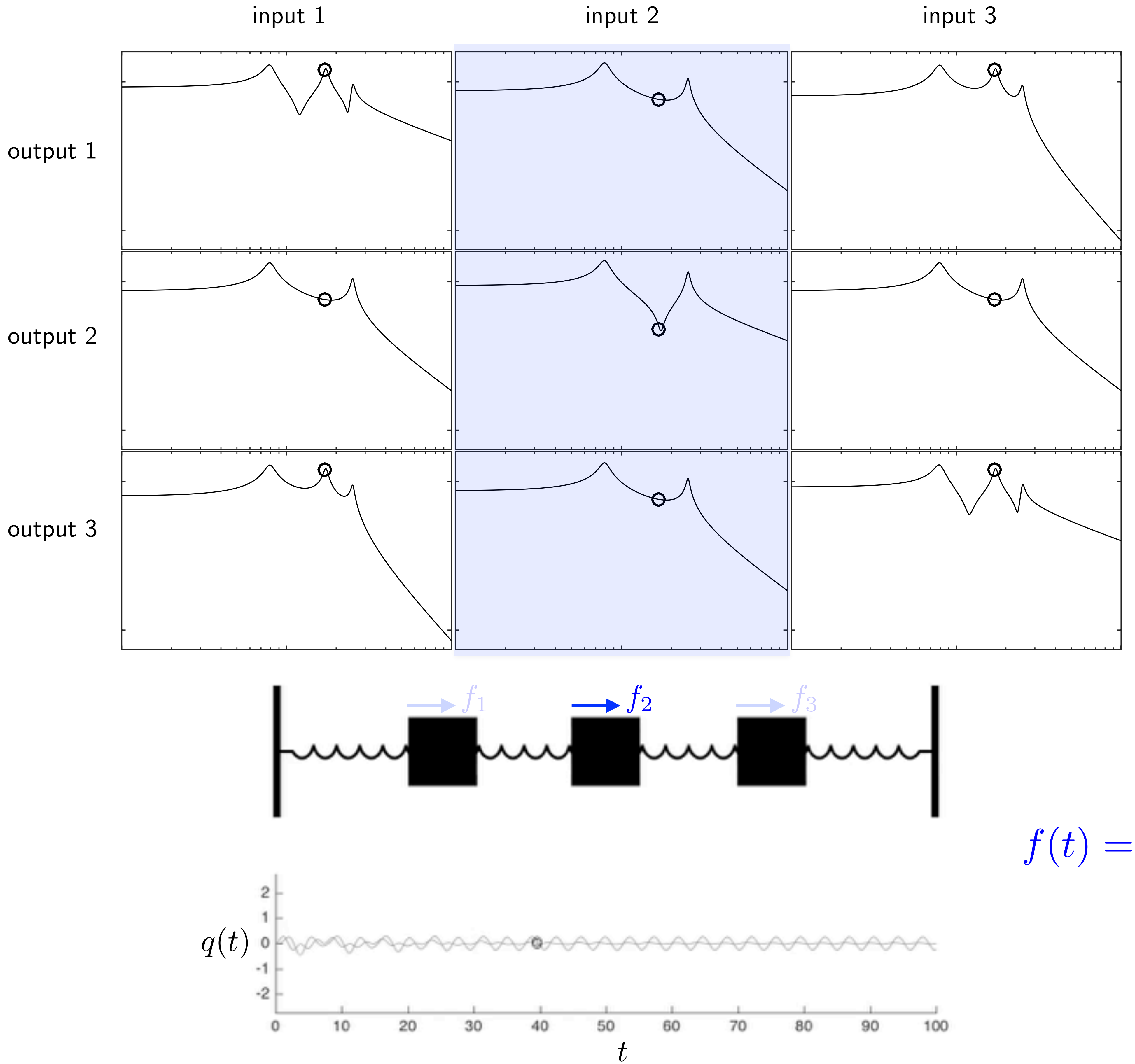






$$f(t) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \cos(\omega_f t)$$

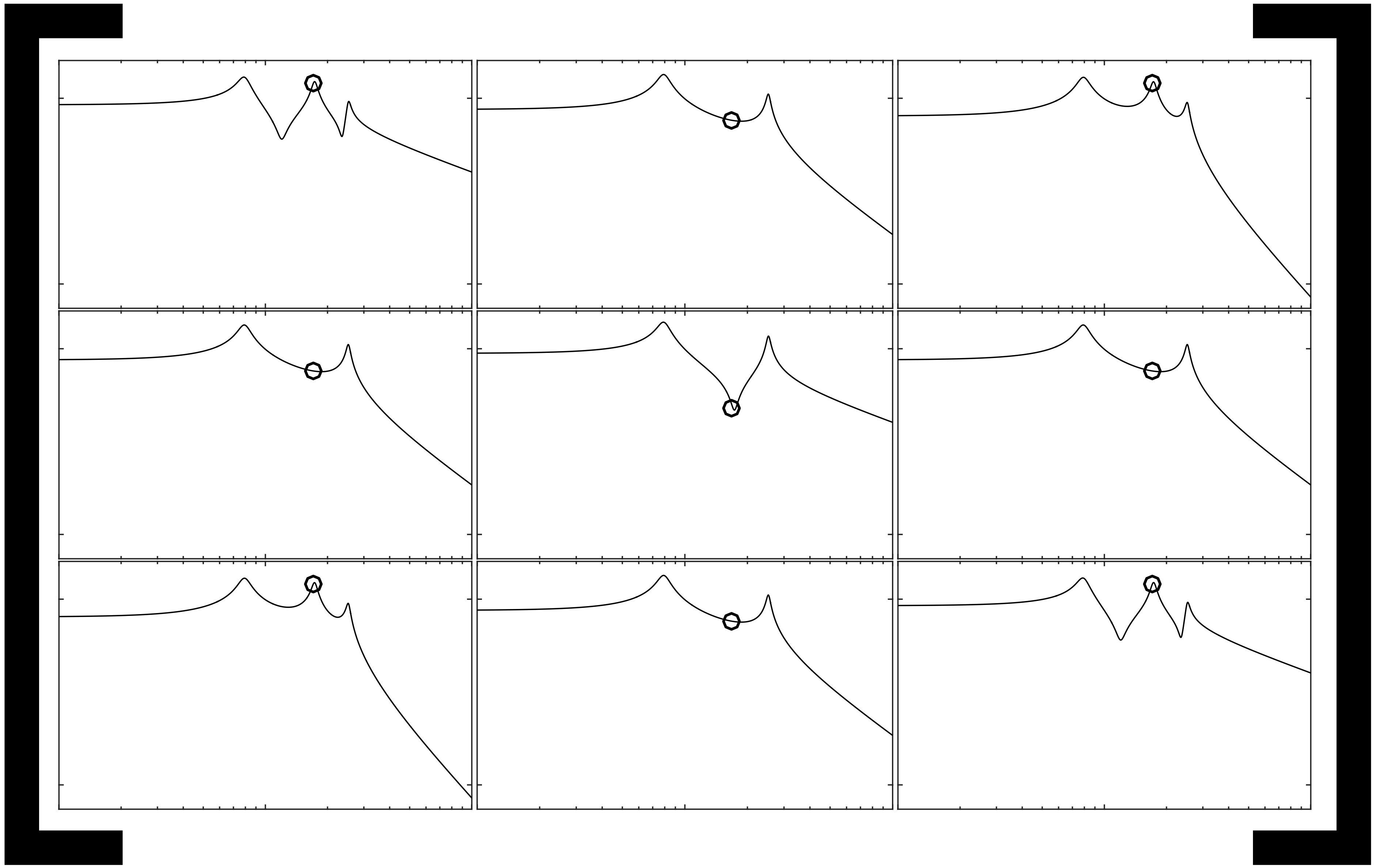
frequency response: MIMO case



$$f(t) = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \cos(\omega_f t)$$

frequency response: MIMO case

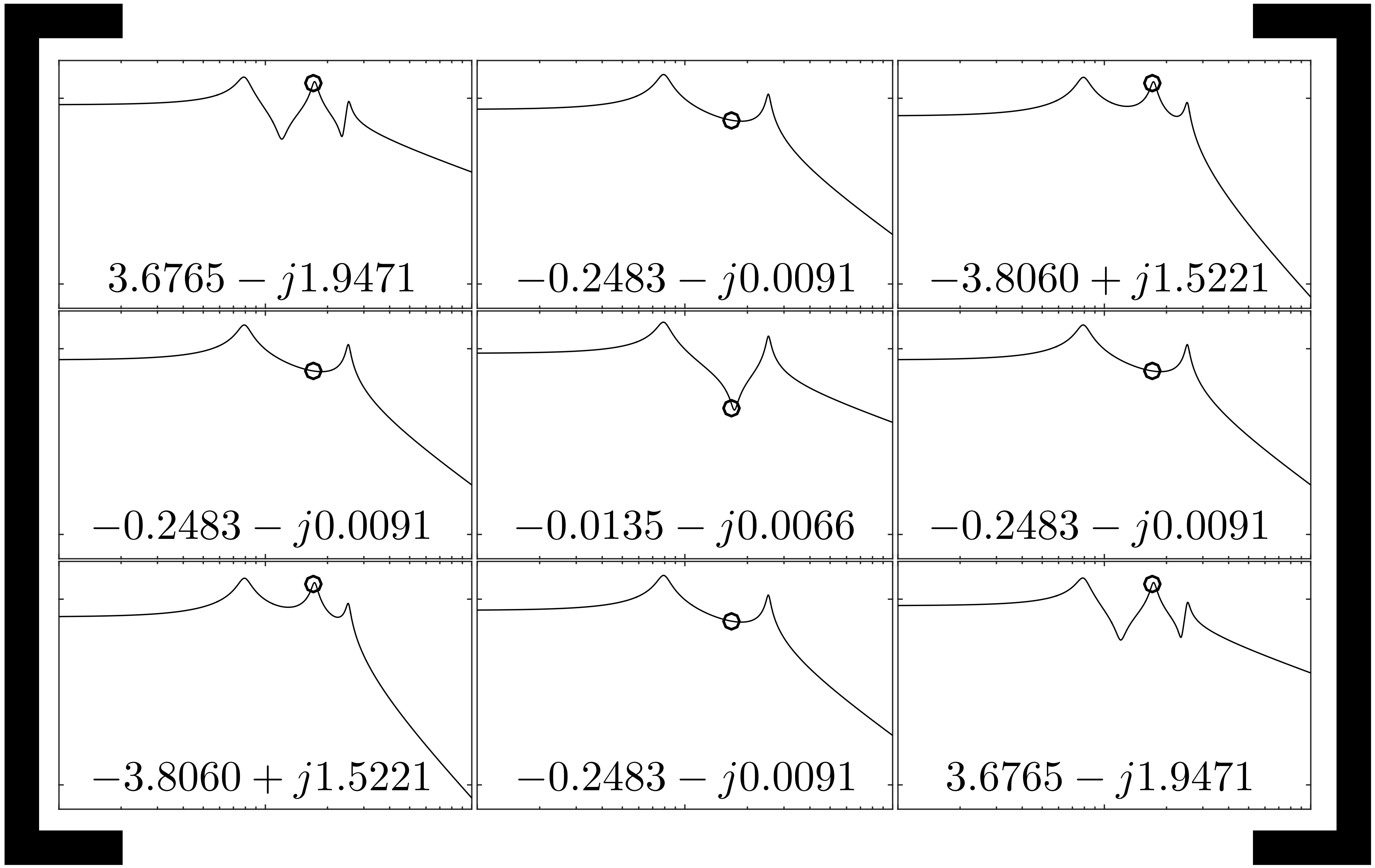
$$P(j\omega) =$$



$$Q(j\omega) = P(j\omega)F(j\omega)$$

frequency response: MIMO case

$P(j\omega) =$



$$Q(j\omega) = P(j\omega)F(j\omega)$$

frequency response: MIMO case

$$P(j\omega) = \begin{bmatrix} P_{11}(j\omega) & P_{12}(j\omega) & P_{13}(j\omega) \\ P_{21}(j\omega) & P_{22}(j\omega) & P_{23}(j\omega) \\ P_{31}(j\omega) & P_{32}(j\omega) & P_{33}(j\omega) \end{bmatrix}$$

$$P(j\omega) = U\Sigma V^*$$

$$P(j\omega) = \begin{bmatrix} | & | & | \\ u_1 & u_2 & u_3 \\ | & | & | \end{bmatrix} \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix} \begin{bmatrix} — & v_1 & — \\ — & v_2 & — \\ — & v_3 & — \end{bmatrix}$$

$$P(j\omega) = \sigma_1 u_1 v_1^* + \sigma_2 u_2 v_2^* + \sigma_3 u_3 v_3^*$$

- interpretation: σ_i: *gains* of the plant

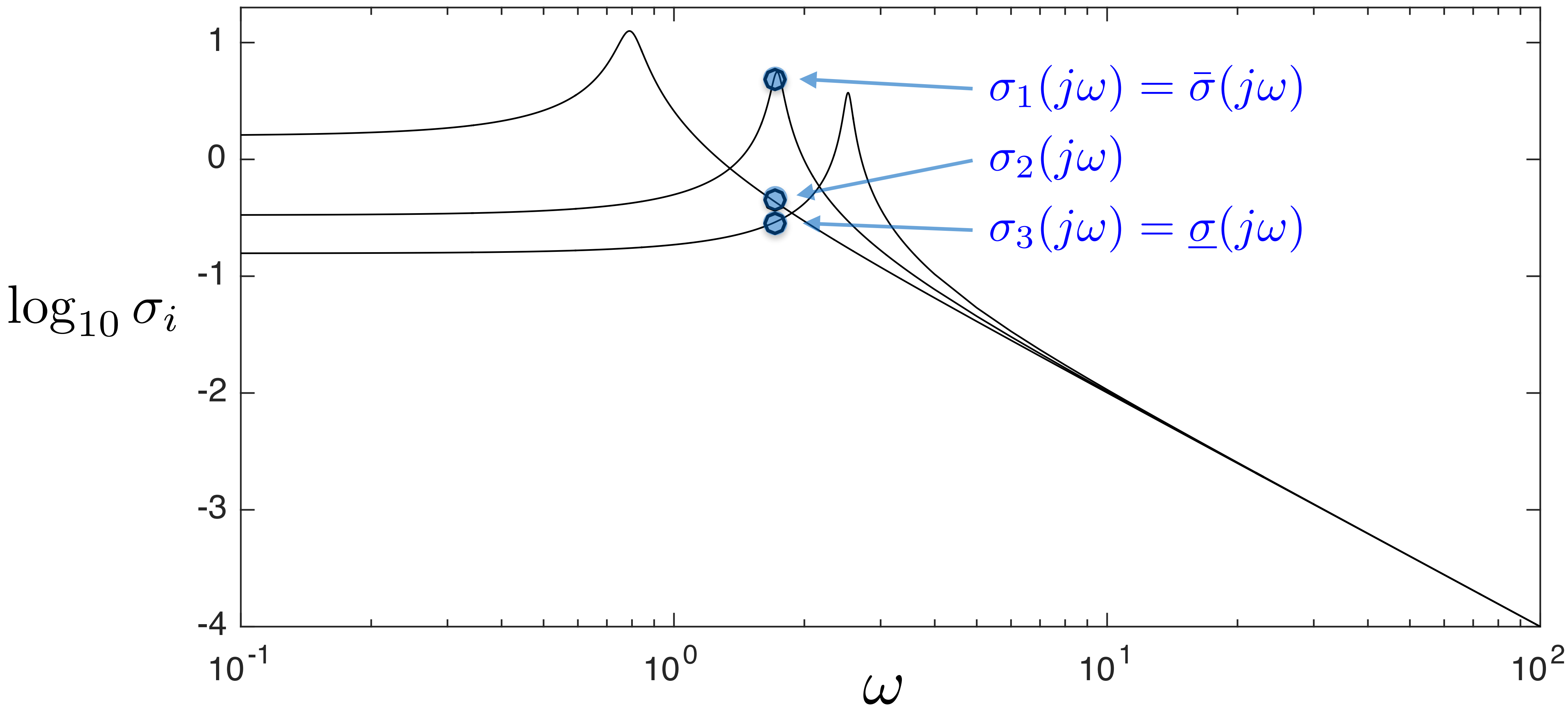
columns of V: *input directions* of the plant

columns of U: *output directions* of the plant

(columns of U and V are orthogonal and of unit length)

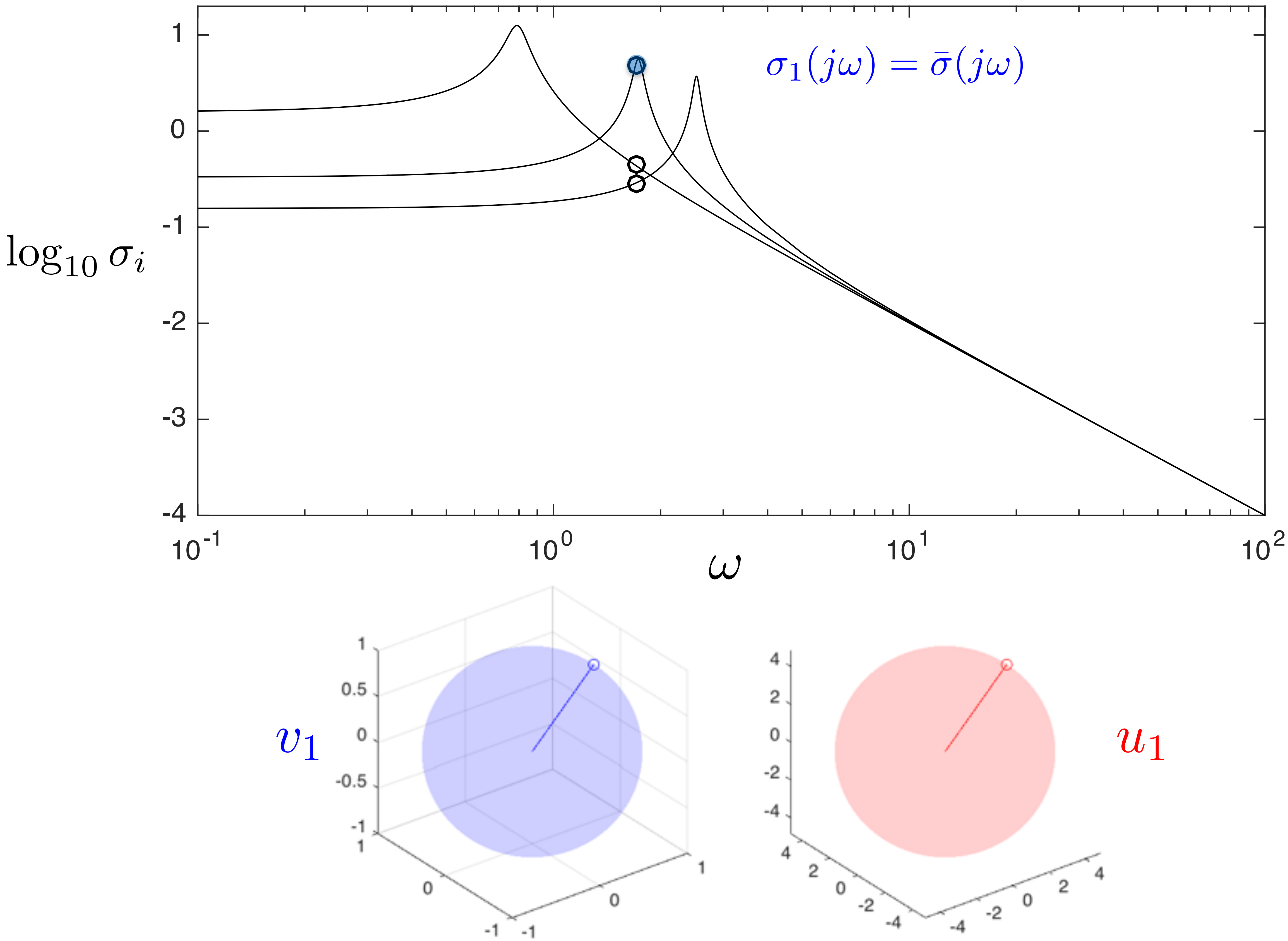
frequency response: MIMO case

$$P(j\omega) = \begin{bmatrix} | & | & | \\ u_1 & u_2 & u_3 \\ | & | & | \end{bmatrix} \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix} \begin{bmatrix} — & v_1 & — \\ — & v_2 & — \\ — & v_3 & — \end{bmatrix}$$

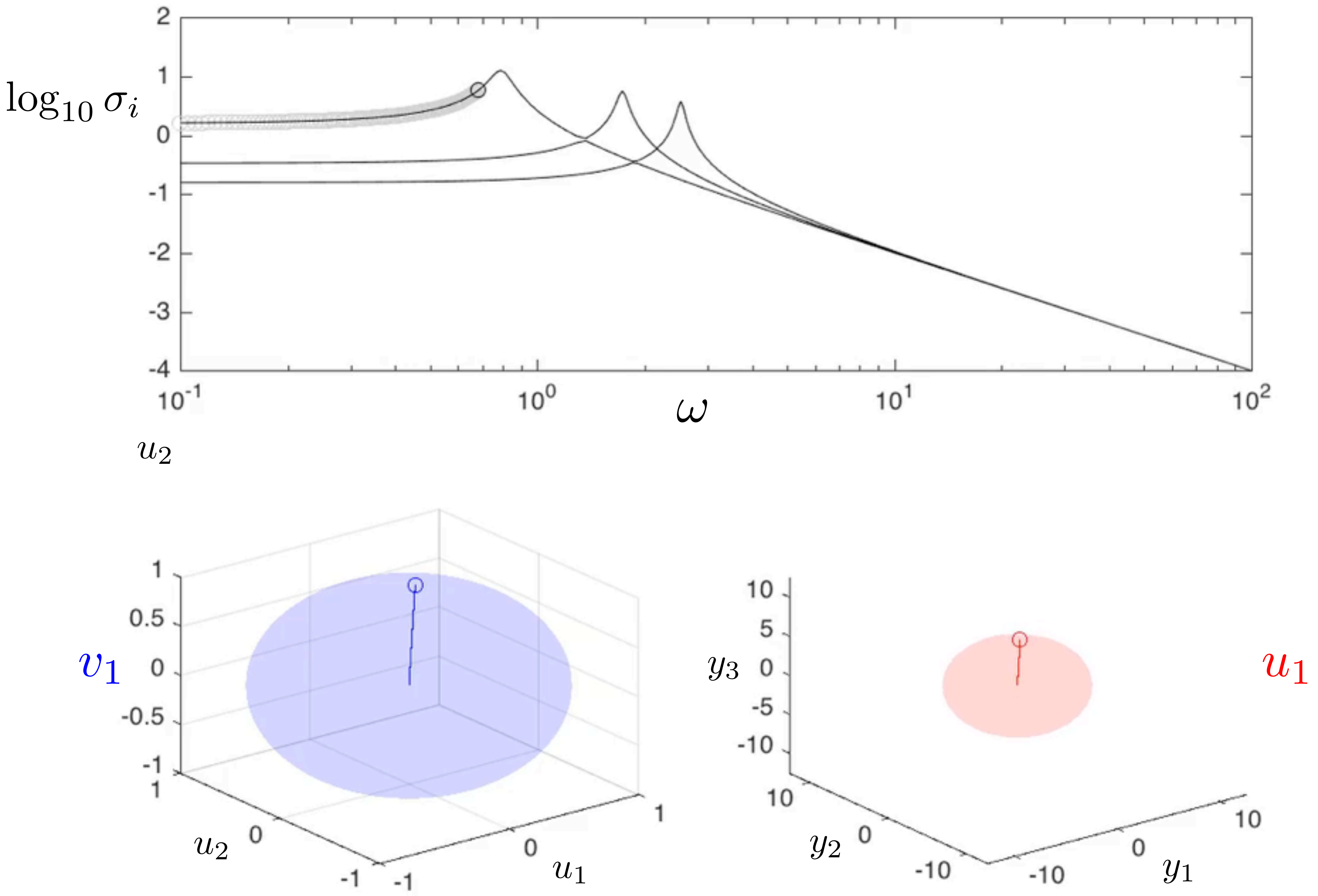


frequency response: MIMO case

$$P(j\omega) = \begin{bmatrix} | & | & | \\ u_1 & u_2 & u_3 \\ | & | & | \end{bmatrix} \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix} \begin{bmatrix} — & v_1 & — \\ — & v_2 & — \\ — & v_3 & — \end{bmatrix}$$



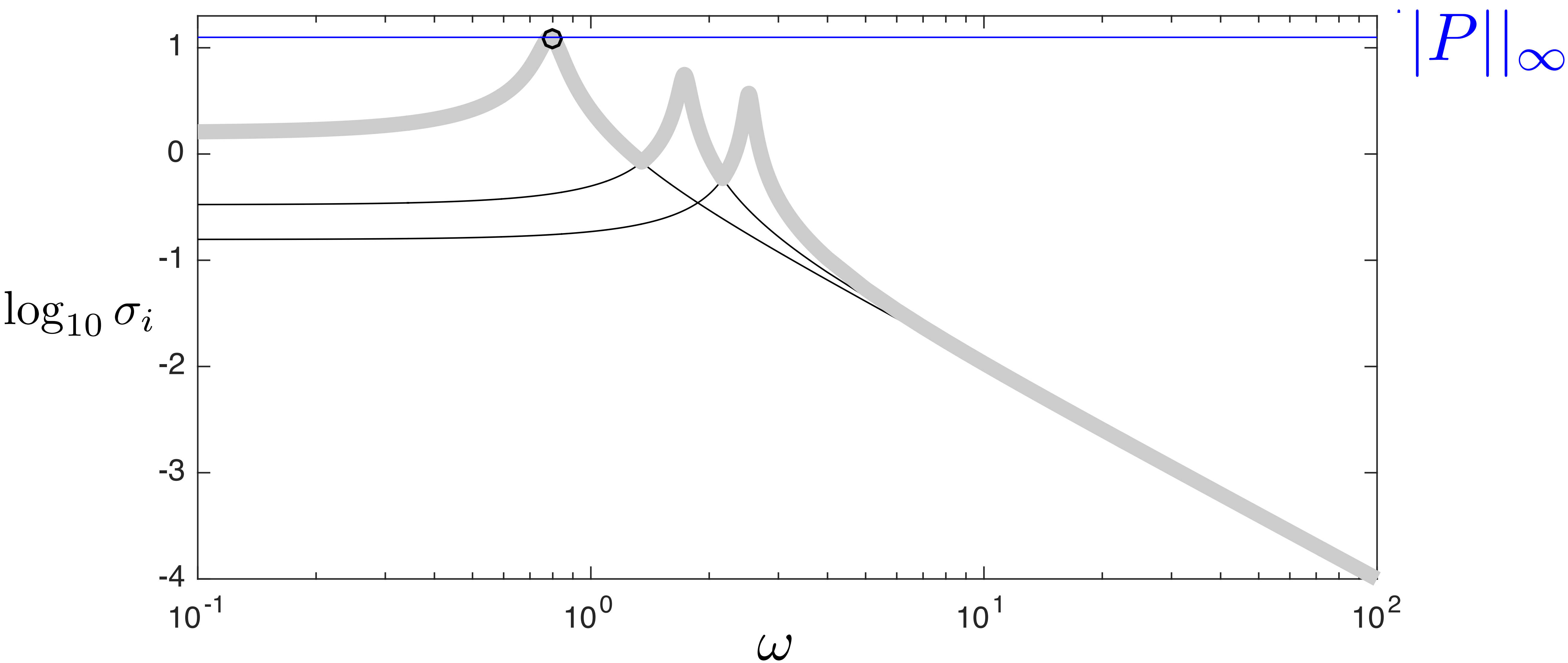
frequency response: MIMO case



the ∞-norm: MIMO case

SISO: $$||P||_\infty = \max_\omega \; P(j\omega)$$

MIMO: $$||P||_\infty = \max_\omega \; \sigma_1(P(j\omega))$$

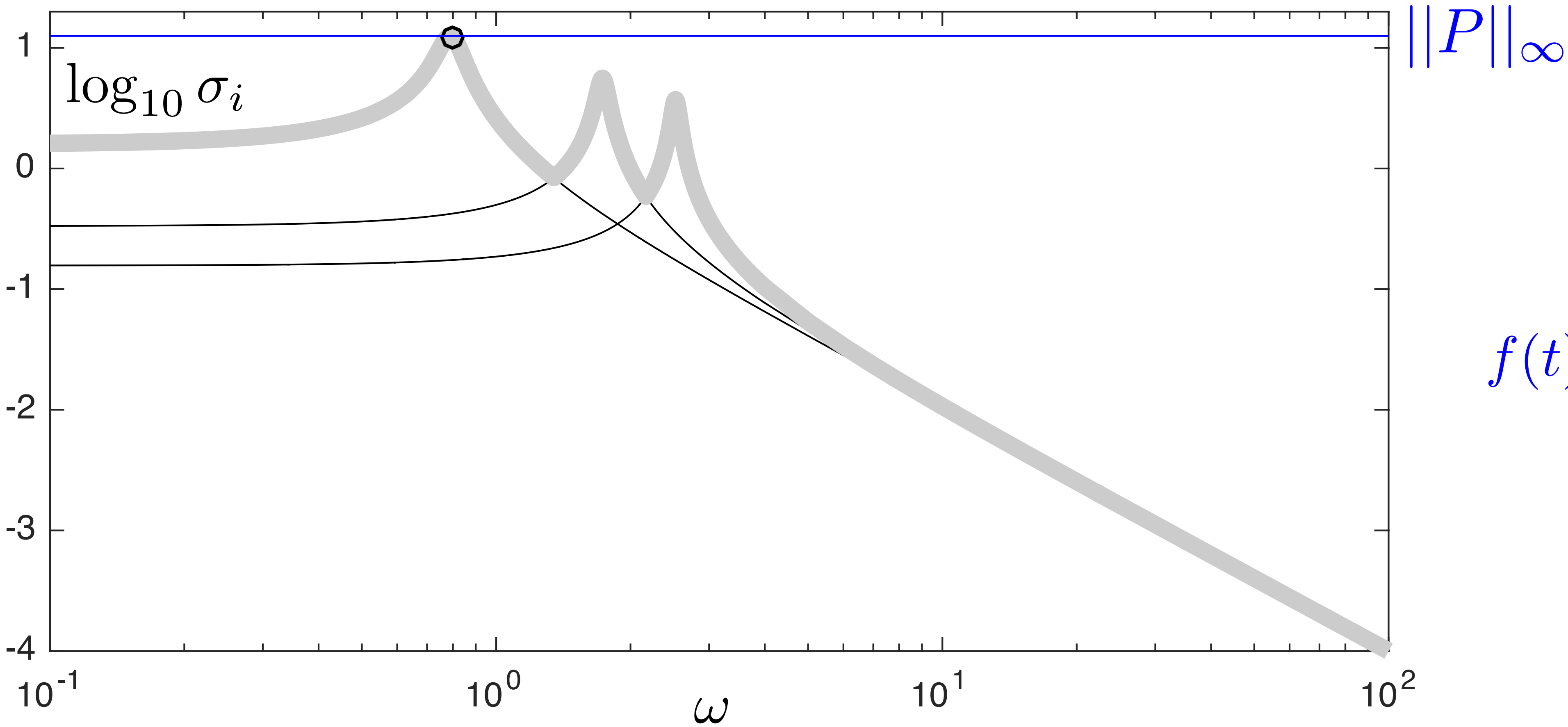


$\underline{\infty\text{-norm}}$: 'worst case' over all frequencies and all directions

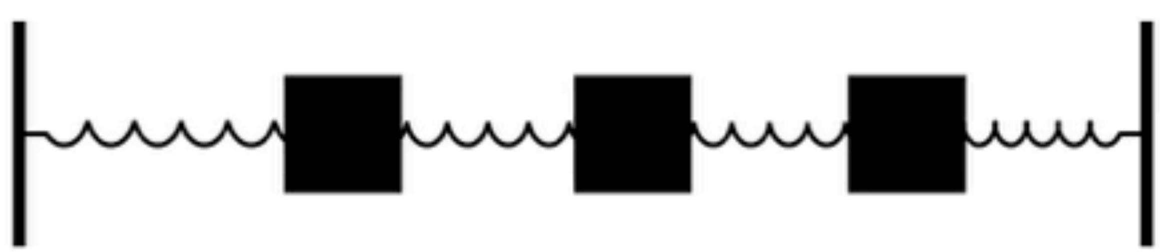
compare with

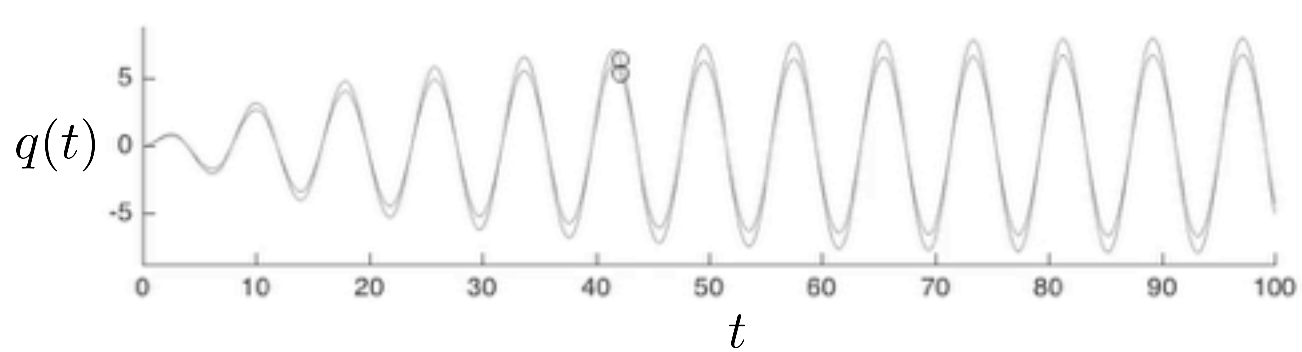
$\underline{2\text{-norm}}$: average over all frequencies and directions

∞-norm: 'worst case' over all frequencies and all directions



$$f(t) = \begin{bmatrix} 0.542 \\ 0.643 \\ 0.542 \end{bmatrix} \cos(0.792\ t)$$





the ∞-norm and model reduction

- suppose we have a plant of order n

$$P_n$$

- and we want to approximate it by a reduced-order plant of order r<n

$$P_r$$

- a good measure of the 'distance' between them is

$$||P_n - P_r||_\infty$$

balanced truncation

$$\dot{x}(t) = A\,x(t) + B\,u(t)$$
$$y(t) = C\,x(t) + D\,u(t)$$

- suppose we decompose the state into two parts, $x = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^T$

$$\begin{bmatrix} \dot{x_1}(t) \\ \dot{x_2}(t) \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u(t)$$
$$y(t) = \begin{bmatrix} C_1 & C_2 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + D\,u(t)$$

- we will keep x_1 and throw away x_2, leaving us with

$$\dot{x_1}(t) = A_1\,x_1(t) + B_1\,u(t)$$
$$y(t) = C_1\,x_1(t) + D\,u(t)$$

balanced truncation

how many states can we throw away, and how should we go about it?

balanced truncation

how many states can we throw away, and how should we go about it?

balanced truncation

how many states can we throw away, and how should we go about it?

enter the observability/controllability Gramians

- observability Gramian:

$$W_o = \Psi_o^* \Psi_o = \int_0^\infty e^{A^*\tau} C^* C e^{A\tau} \mathrm{d}\tau$$

- <u>interpretation</u>: for zero input, $u(t) = 0$ and initial state x_0, the resulting output has energy

$$||y||_2^2 = \int_0^\infty y^*(\tau) y(\tau) \mathrm{d}\tau = \int_0^\infty (C e^{A\tau} x_0)^* C e^{A\tau} x_0 \mathrm{d}\tau = x_0^* W_o x_0$$

- if we consider initial states with $|x_0| = 1$, some will give higher output norms than others
- states giving larger output norms are considered more observable

enter the observability/controllability Gramians

- controllability Gramian:

$$W_c = \Psi_c \Psi_c^* = \int_0^\infty e^{\tau A} B B^* e^{\tau A^*} \mathrm{d}\tau$$

- interpretation: if we want to reach a state x_0, then the minimum input energy required to get there is

$$||u||^2 = x_0^* W_c^{-1} x_0$$

- if we consider initial states with $|x_0| = 1$, some are easier to "drive to" than others
- states that are easier to "drive to" are considered more observable

balanced truncation: change to coordinates in which the observability and controllability Gramians are *equal and diagonal*

- guaranteed error bound:

$$\sigma_{r+1} \leq ||P - P_r||_\infty \leq 2(\sigma_{r+1} + \sigma_{r+2} + \cdots + \sigma_n)$$

- ("twice the sum of the tails")

- note: balanced truncation is not optimal, but is (provably) not far off

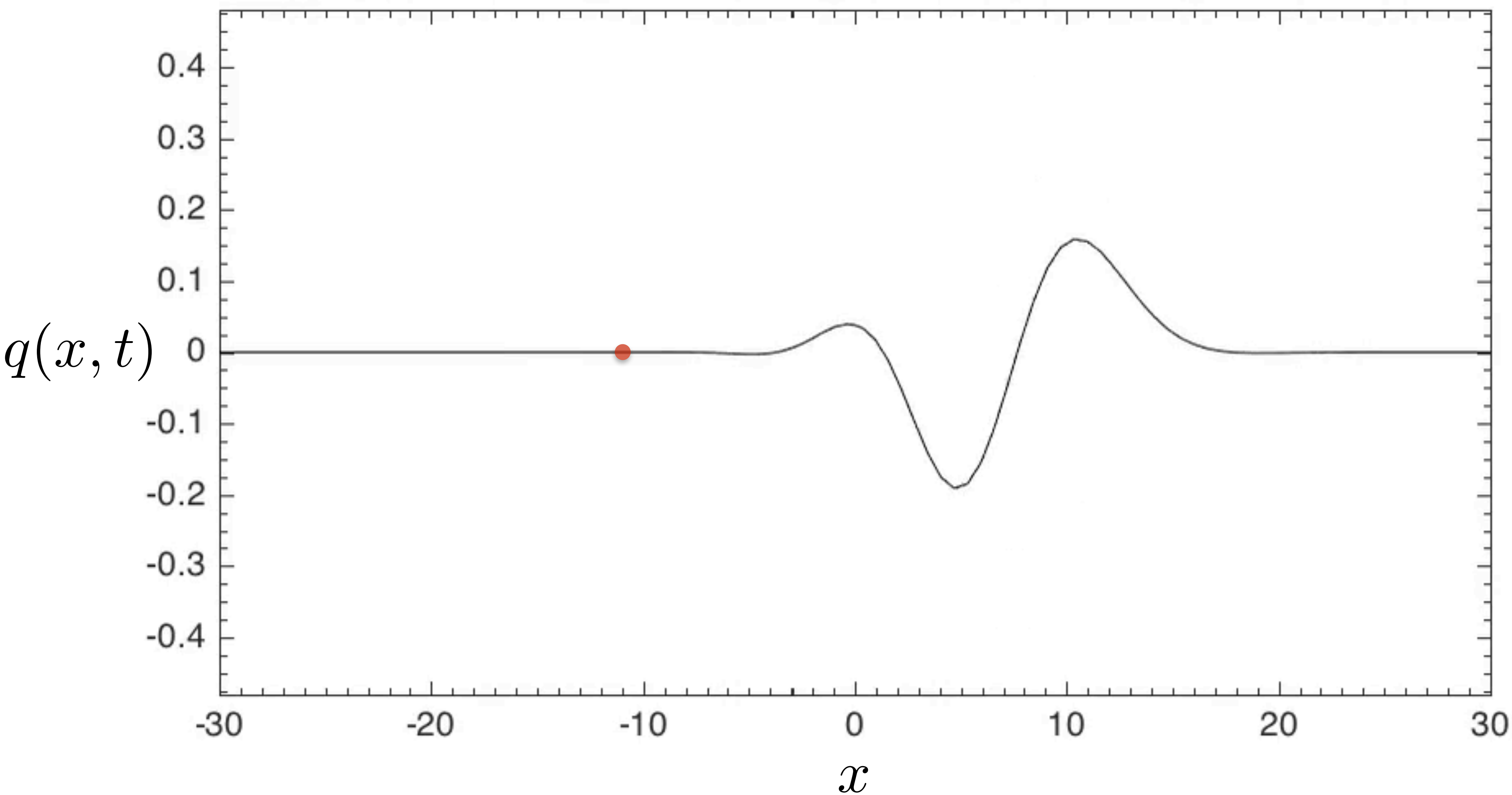
balanced truncation: applied to Ginzburg-Landau system

$$\boxed{\frac{\partial q}{\partial t}(x,t) = \left(-\nu\frac{\partial}{\partial x} + \gamma\frac{\partial^2}{\partial x^2} + \mu(x)\right) q(x,t)} \quad -\infty < x < \infty$$

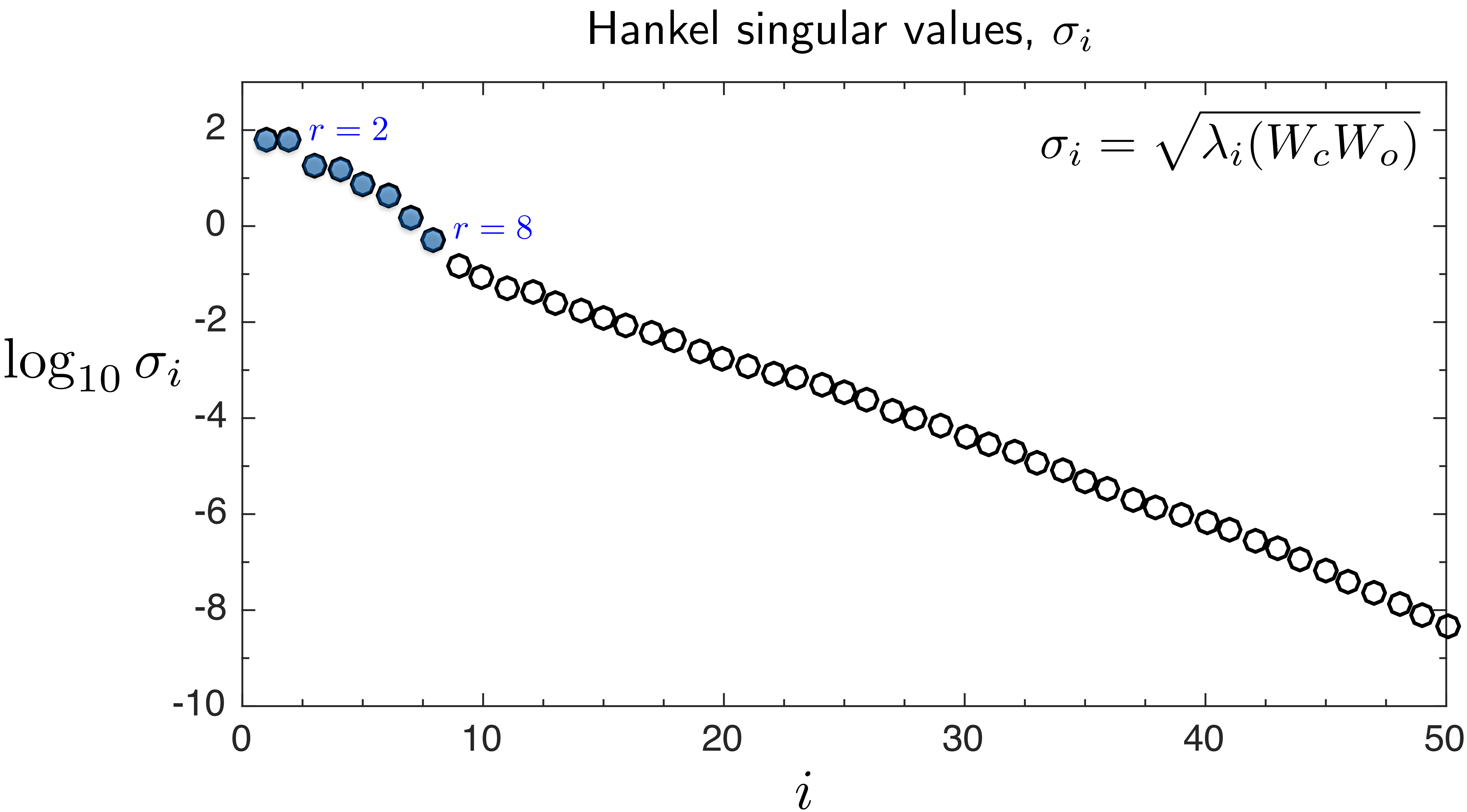
$$\begin{aligned} \text{convection:} \quad & \nu = U + i2c_u \\ \text{dissipation:} \quad & \gamma = 1 + ic_d \\ \text{growth/decay:} \quad & \mu(x) = \mu_0 - c_u^2 + \mu_2 x^2/2 \end{aligned}$$

discretized using Hermite collocation method

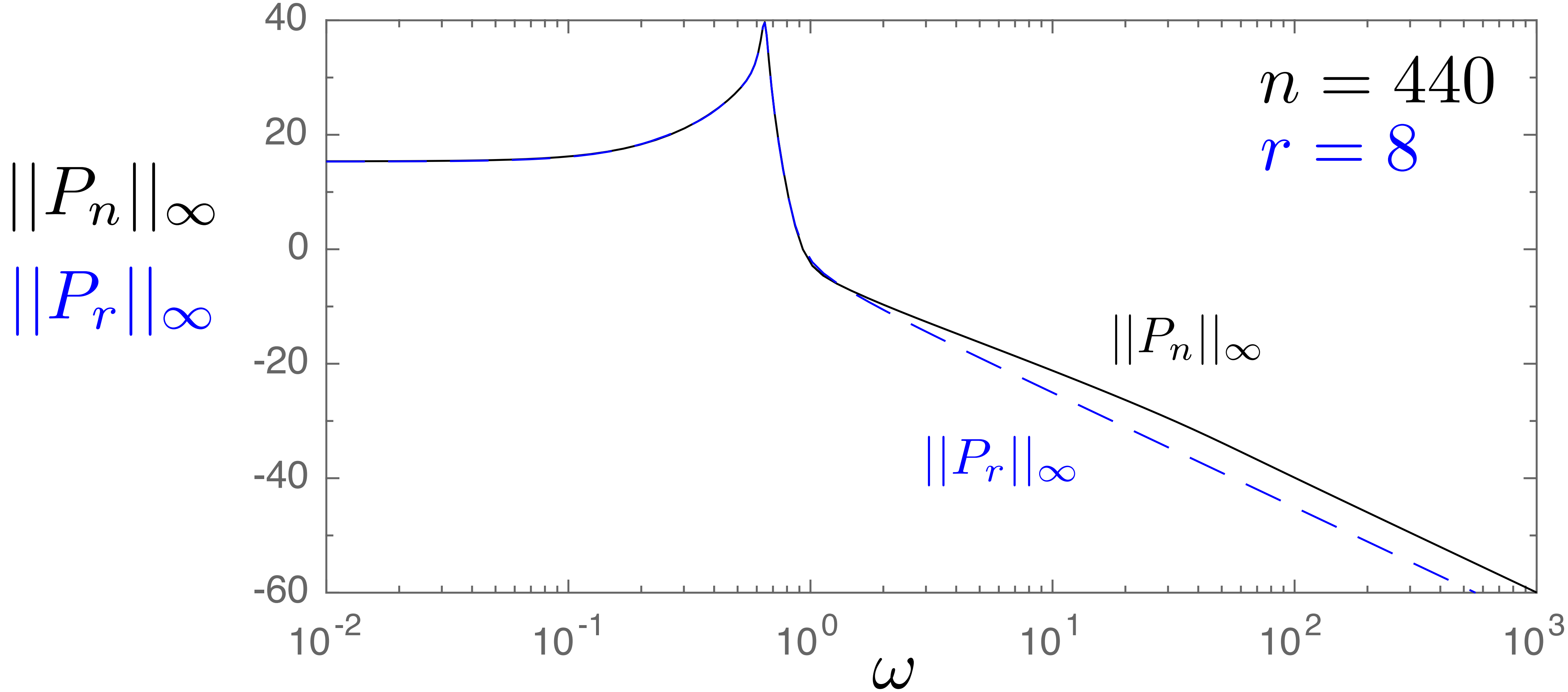
balanced truncation: applied to Ginzburg-Landau system



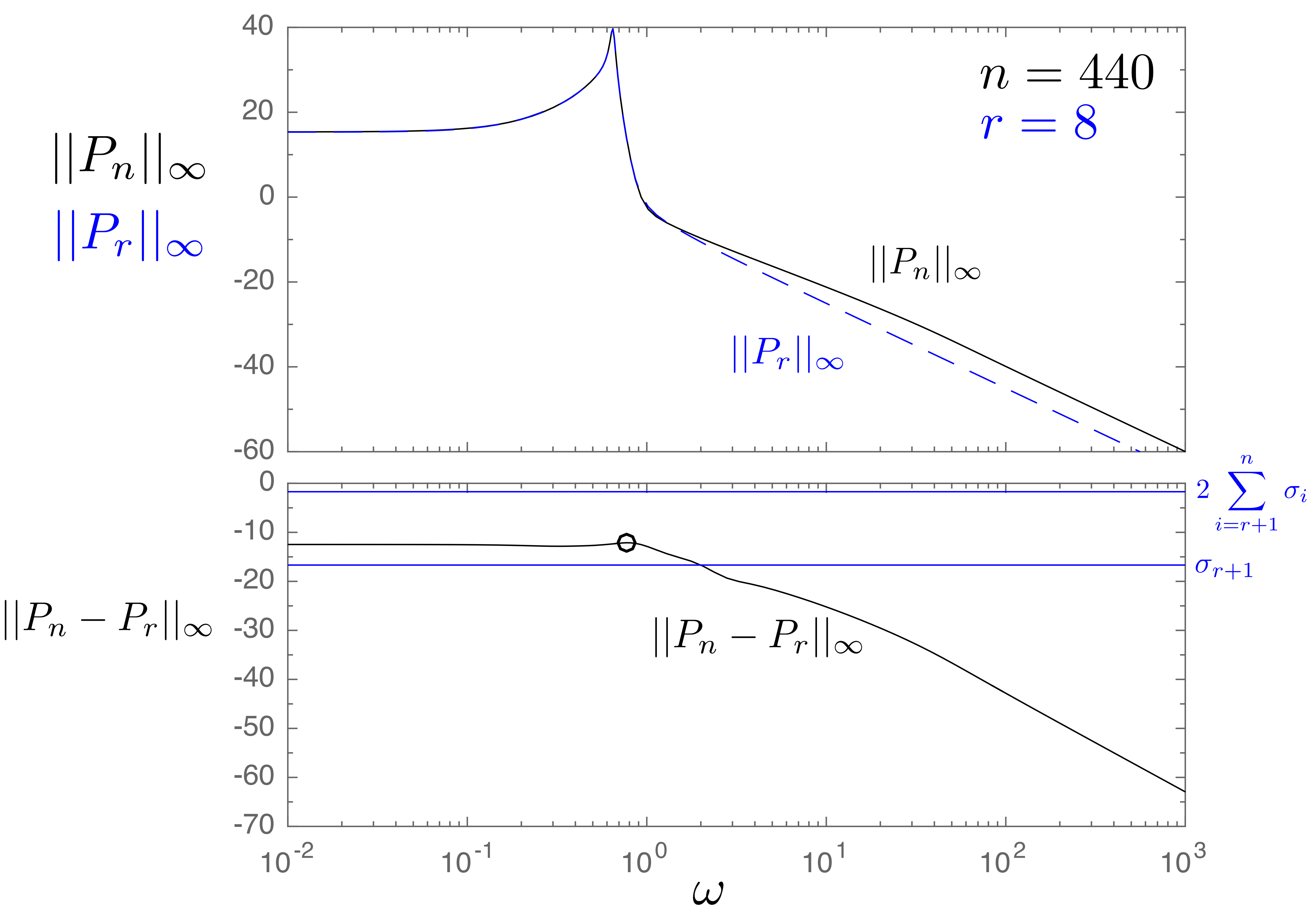
balanced truncation: applied to Ginzburg-Landau system



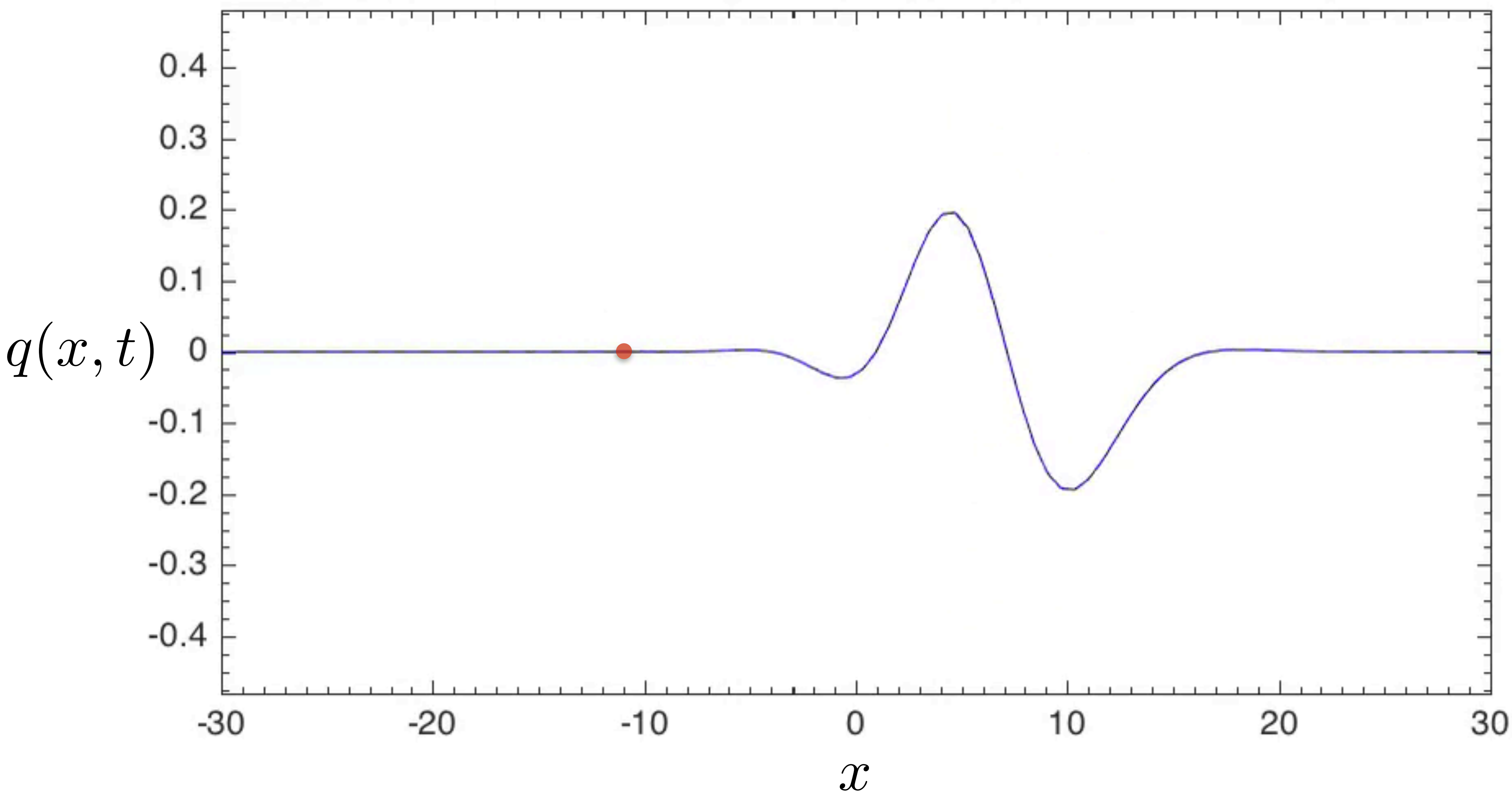
balanced truncation: applied to Ginzburg-Landau system



balanced truncation: applied to Ginzburg-Landau system



balanced truncation: applied to Ginzburg-Landau system



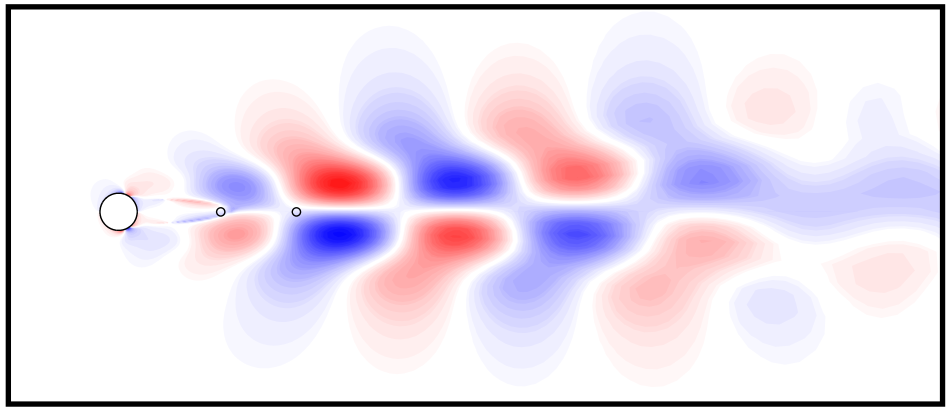
some applications

some applications

1. estimation
2. control

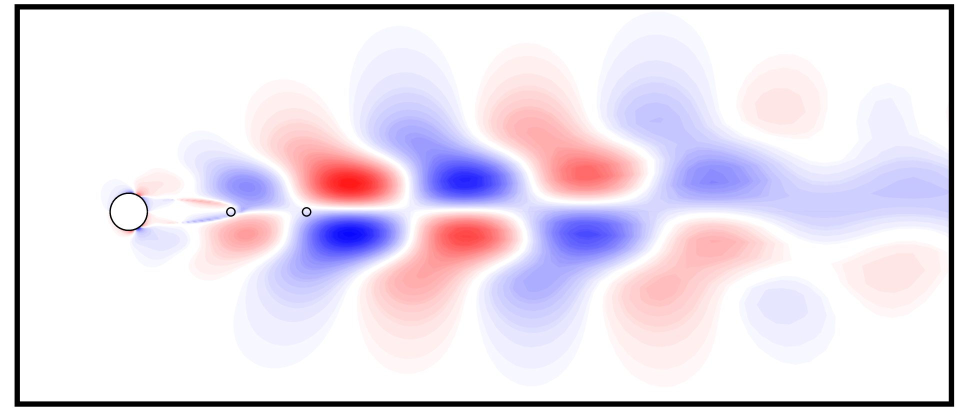
some applications

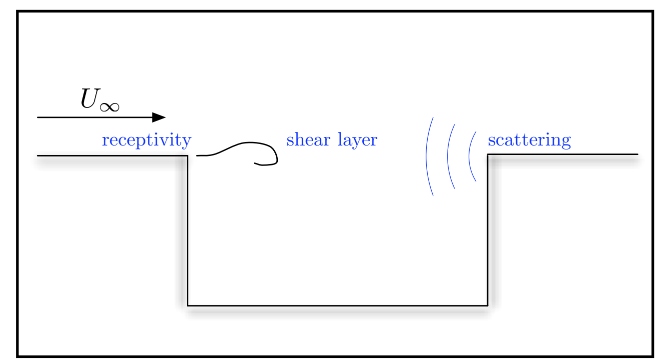
1. estimation

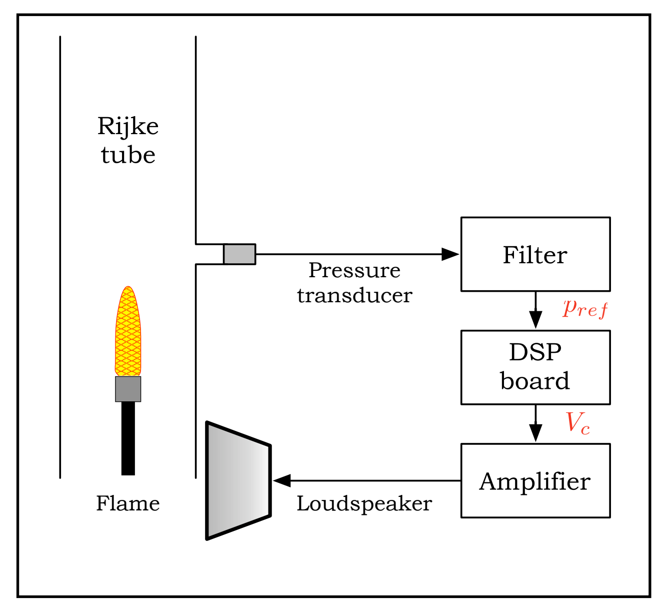


$$\frac{\partial q}{\partial t}(x,t) = \left(-\nu\frac{\partial}{\partial x} + \gamma\frac{\partial^2}{\partial x^2} + \mu(x)\right) q(x,t)$$

2. control







the take-away messages

1. estimation

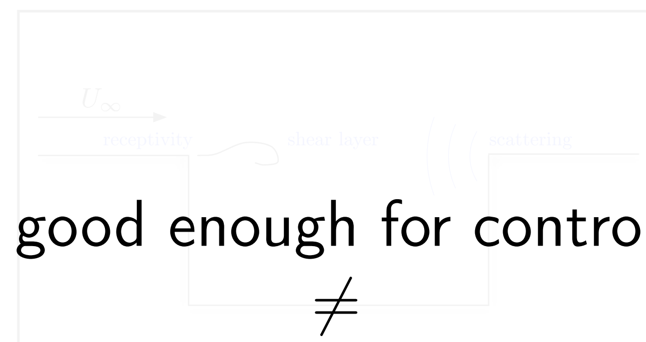
good results even for $n \gg q$

$$\frac{\partial q}{\partial t}(x,t) = \left(-\nu\frac{\partial}{\partial x} + \gamma\frac{\partial^2}{\partial x^2} + \mu(x)\right) q(x,t)$$

'local' or 'integral' sensing can work

2. control

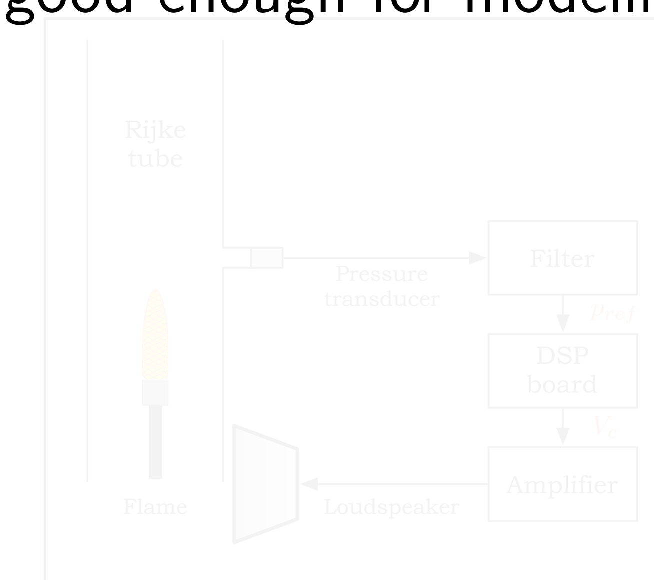
robust controllers are (very) forgiving



good enough for control

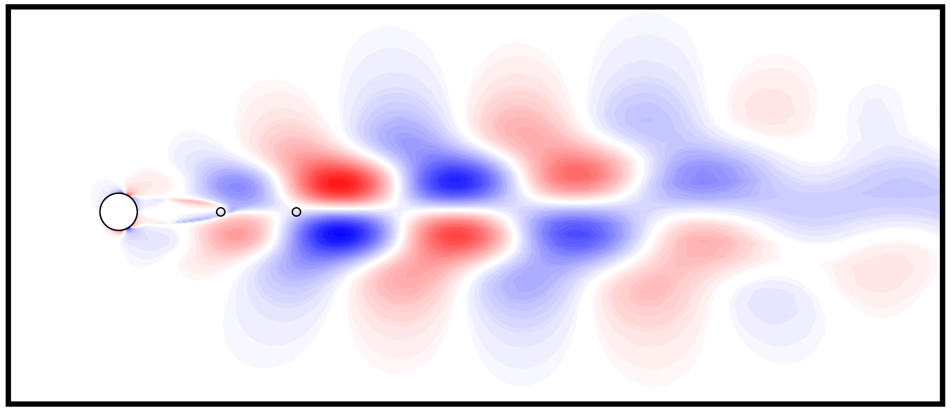
$\neq$

good enough for modelling



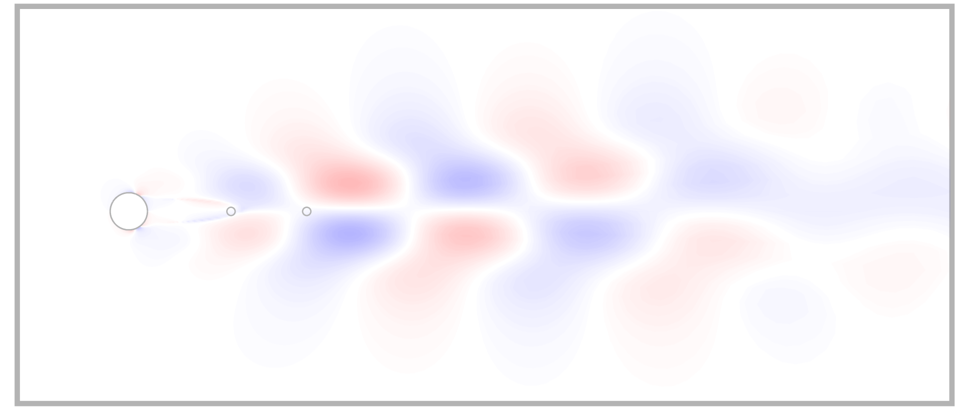
some applications

1. estimation

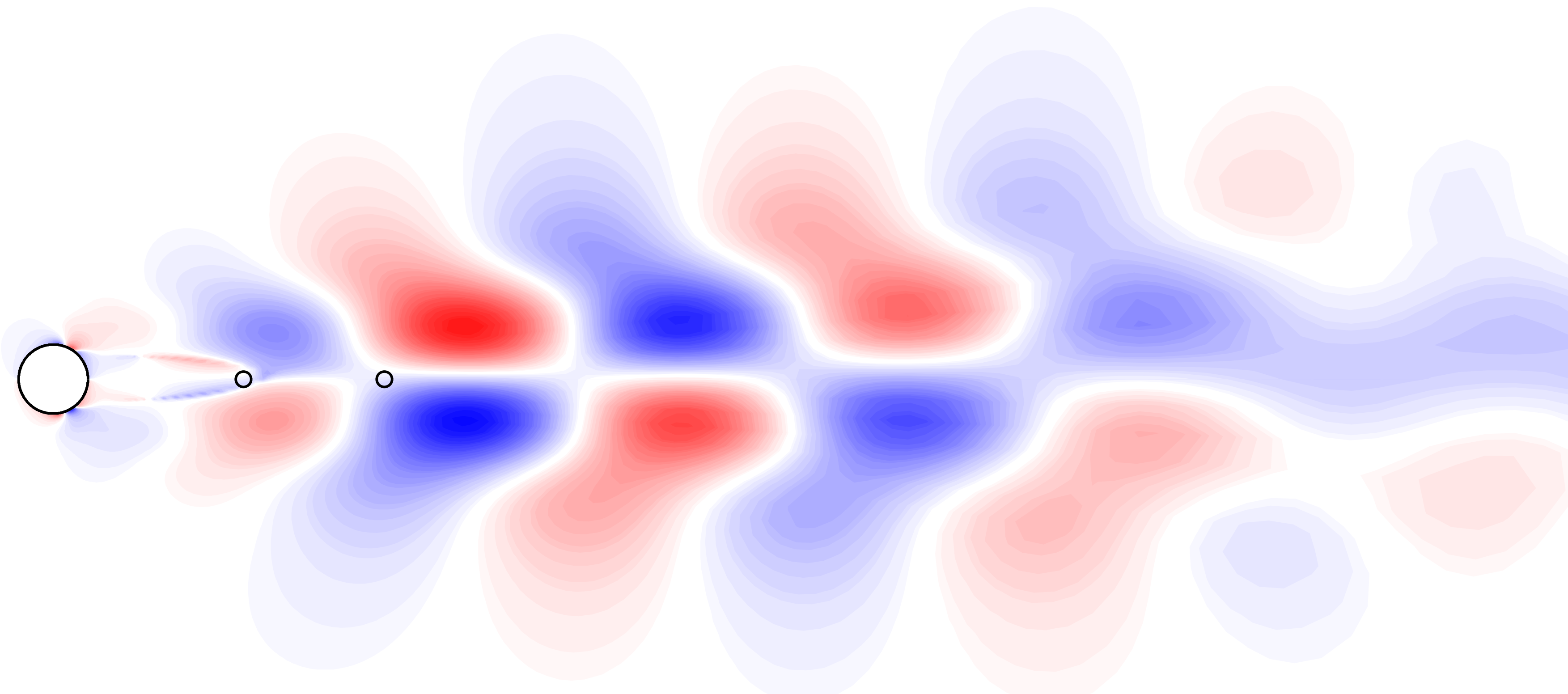


$$\frac{\partial q}{\partial t}(x,t) = \left(-\nu\frac{\partial}{\partial x} + \gamma\frac{\partial^2}{\partial x^2} + \mu(x)\right) q(x,t)$$

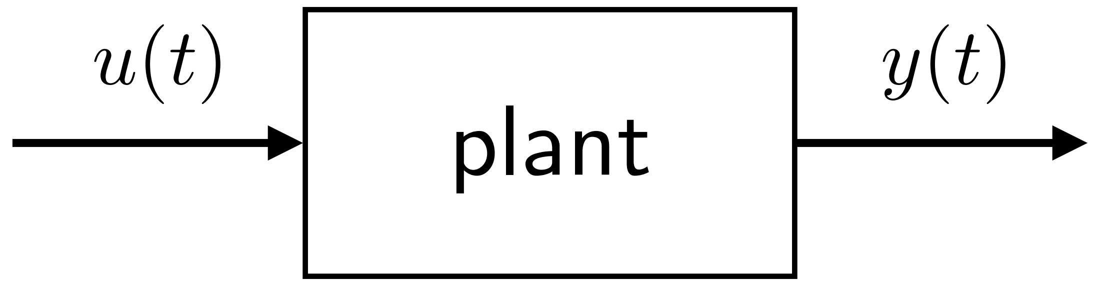
2. control



estimation



dynamic estimation

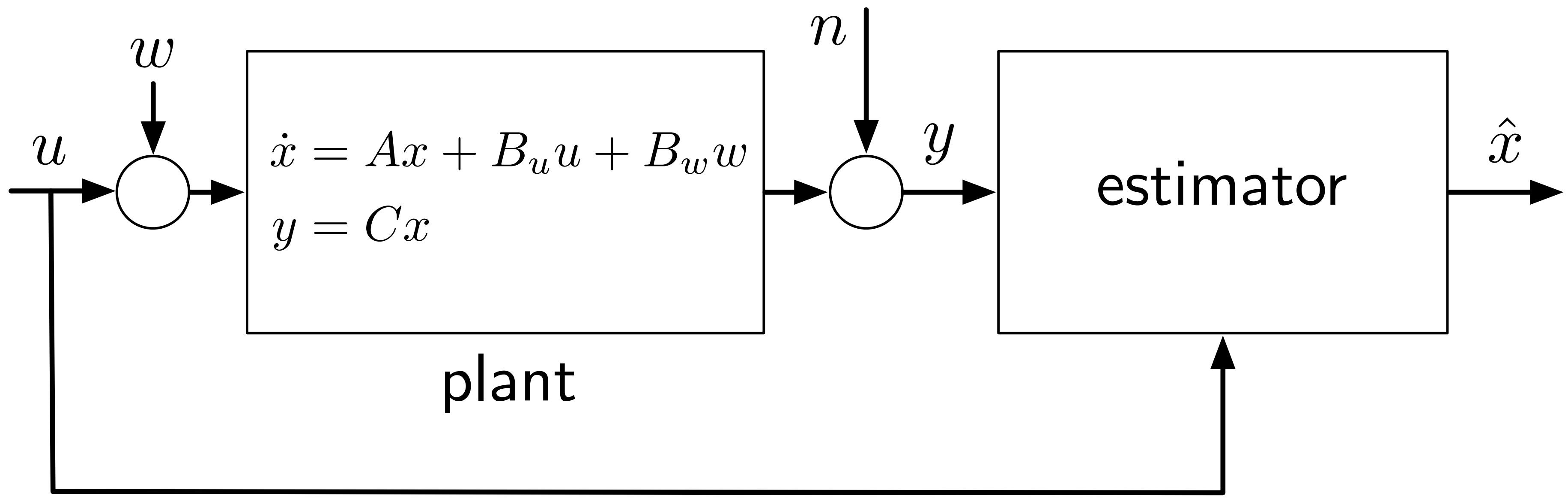


$$\dot{x}(t) = Ax(t) + Bu(t)$$
$$y(t) = Cx(t) + Du(t)$$

$u(t)$ is the *input*
$y(t)$ is the *output*
$x(t)$ is the *state*, $x(t) \in \mathbb{R}^n$

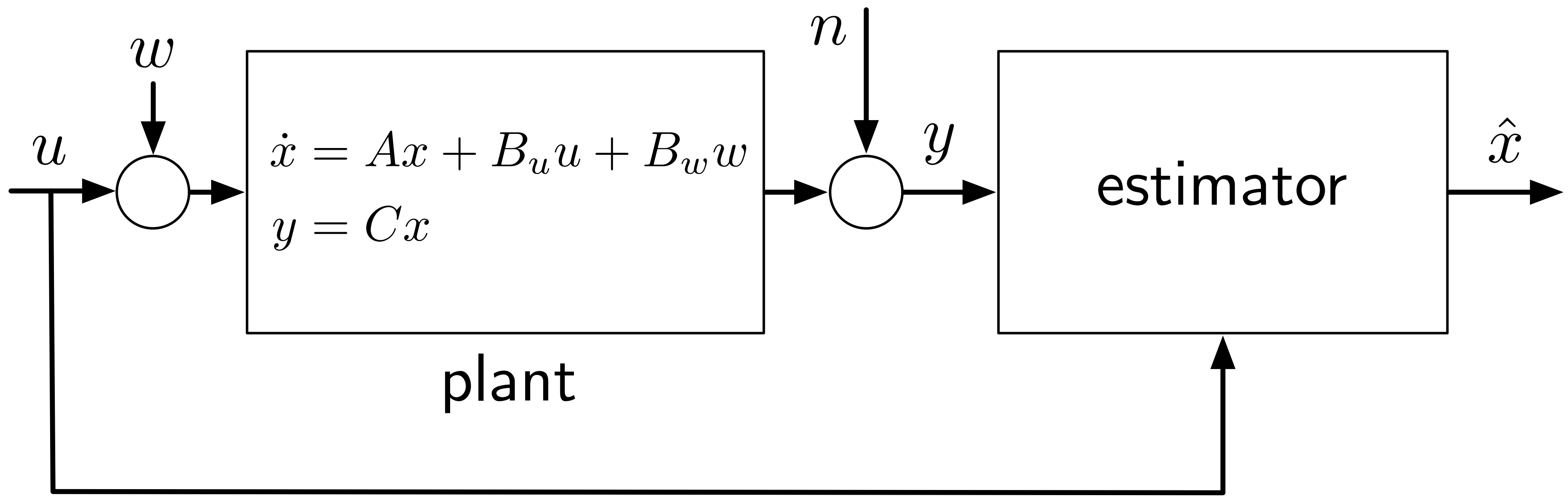
$$\dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t) + \boxed{L(y(t) - \hat{y}(t))}$$
$$\hat{y}(t) = C\hat{x}(t) + Du(t)$$

dynamic estimation



- define error: $$e(t) = \hat{x}(t) - x(t)$$
- then error satisfies: $$\dot{e}(t) = (A - LC)\, e(t)$$
- can specify error dynamics by suitable choice of the matrix L
- Kalman filter amounts to a specific choice of L

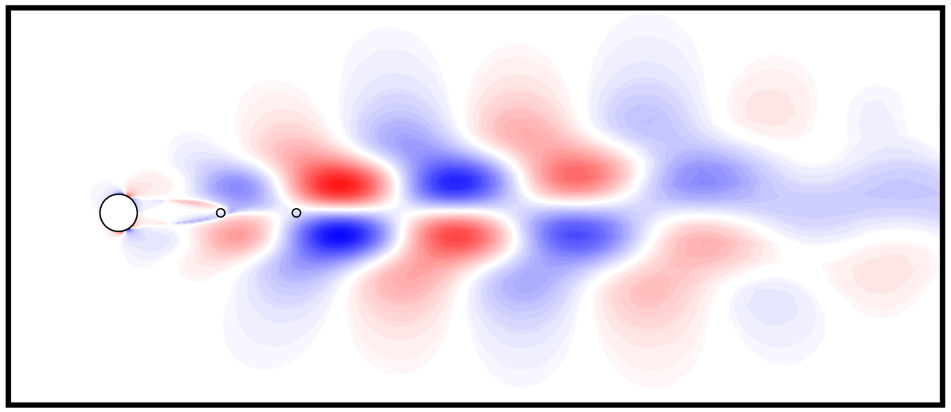
dynamic estimation



- the Kalman filter:
  - is dynamic: i.e. it uses time-resolved data to form an estimate
  - accommodates unknown disturbances w and sensor noise n in its framework

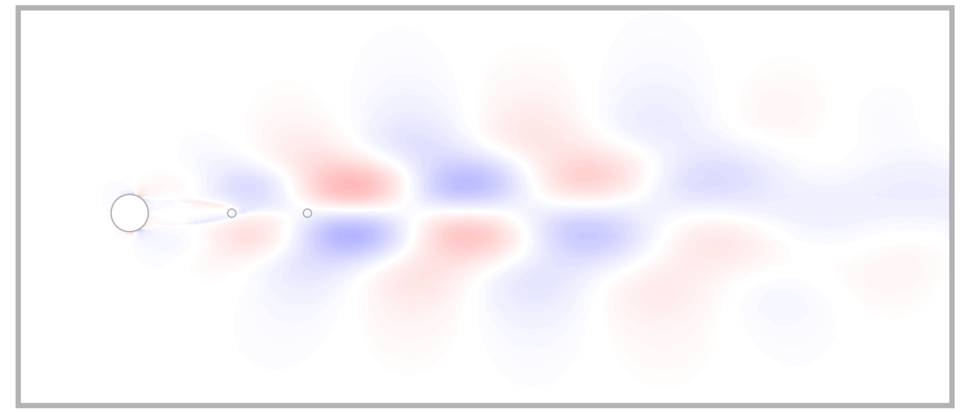
some applications

1. estimation

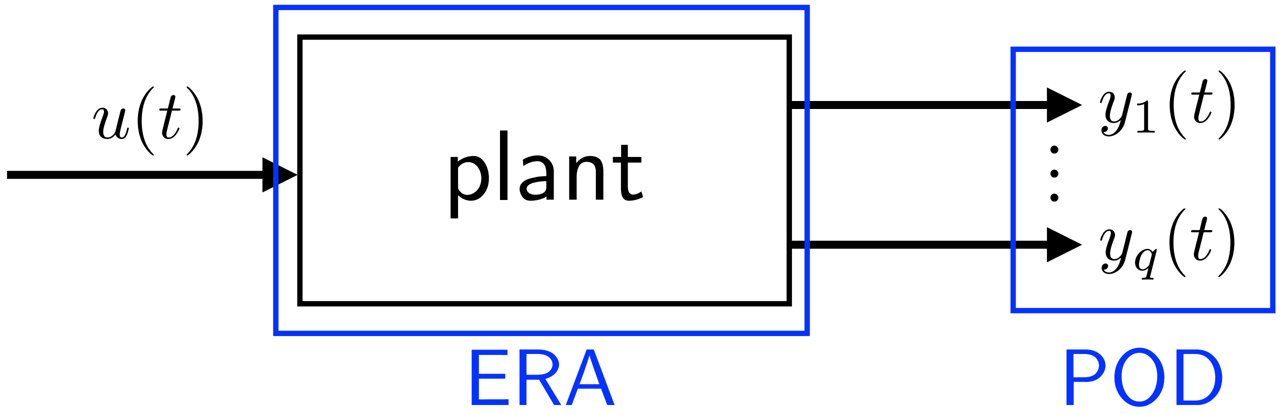


$$\frac{\partial q}{\partial t}(x,t) = \left(-\nu\frac{\partial}{\partial x} + \gamma\frac{\partial^2}{\partial x^2} + \mu(x)\right) q(x,t)$$

2. control



POD is used to reduce the number of outputs



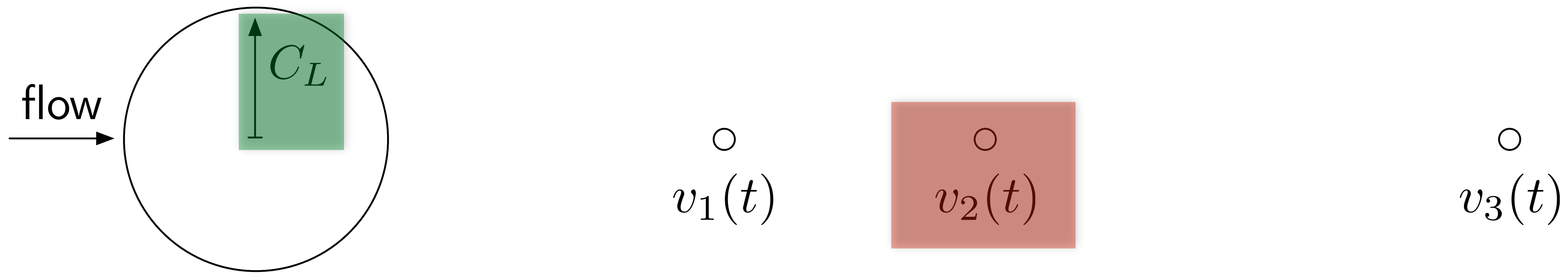
- DNS has $256 \times 220 = 56320$ outputs
- solution: decompose output into leading POD modes
- 31 POD modes are used
- the ERA model order is 29 (order 4 performs almost as well)

results

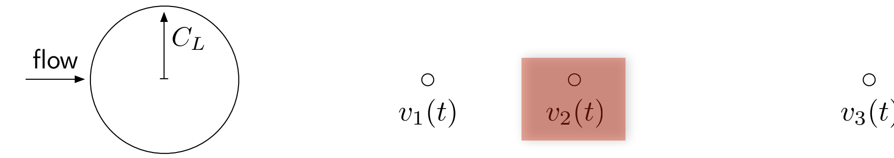
- we estimate the entire flow using

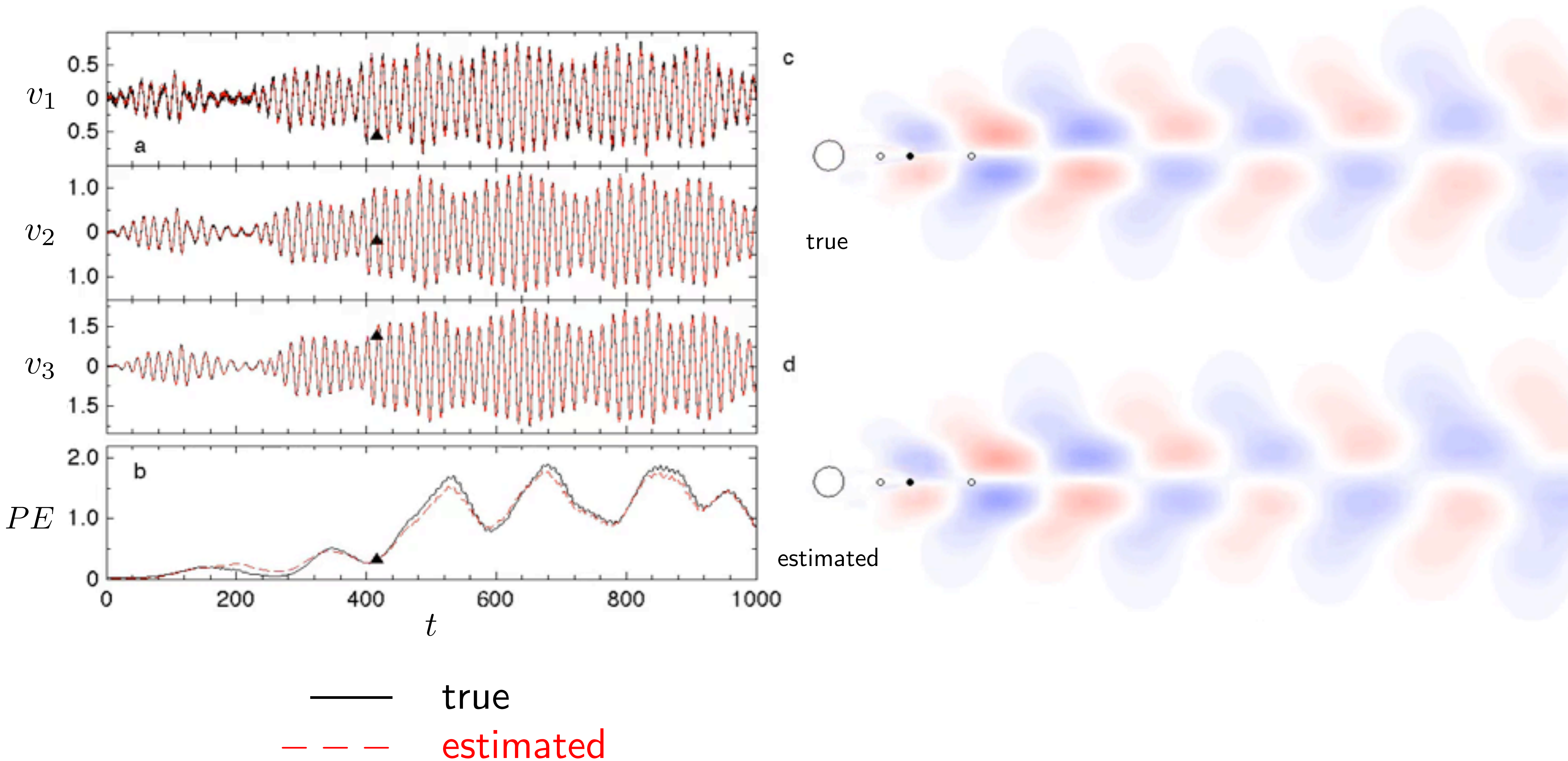
i. transverse velocity at sensor two only

ii. lift force only



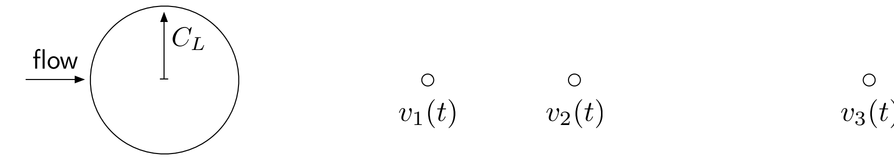
transverse velocity at sensor two only



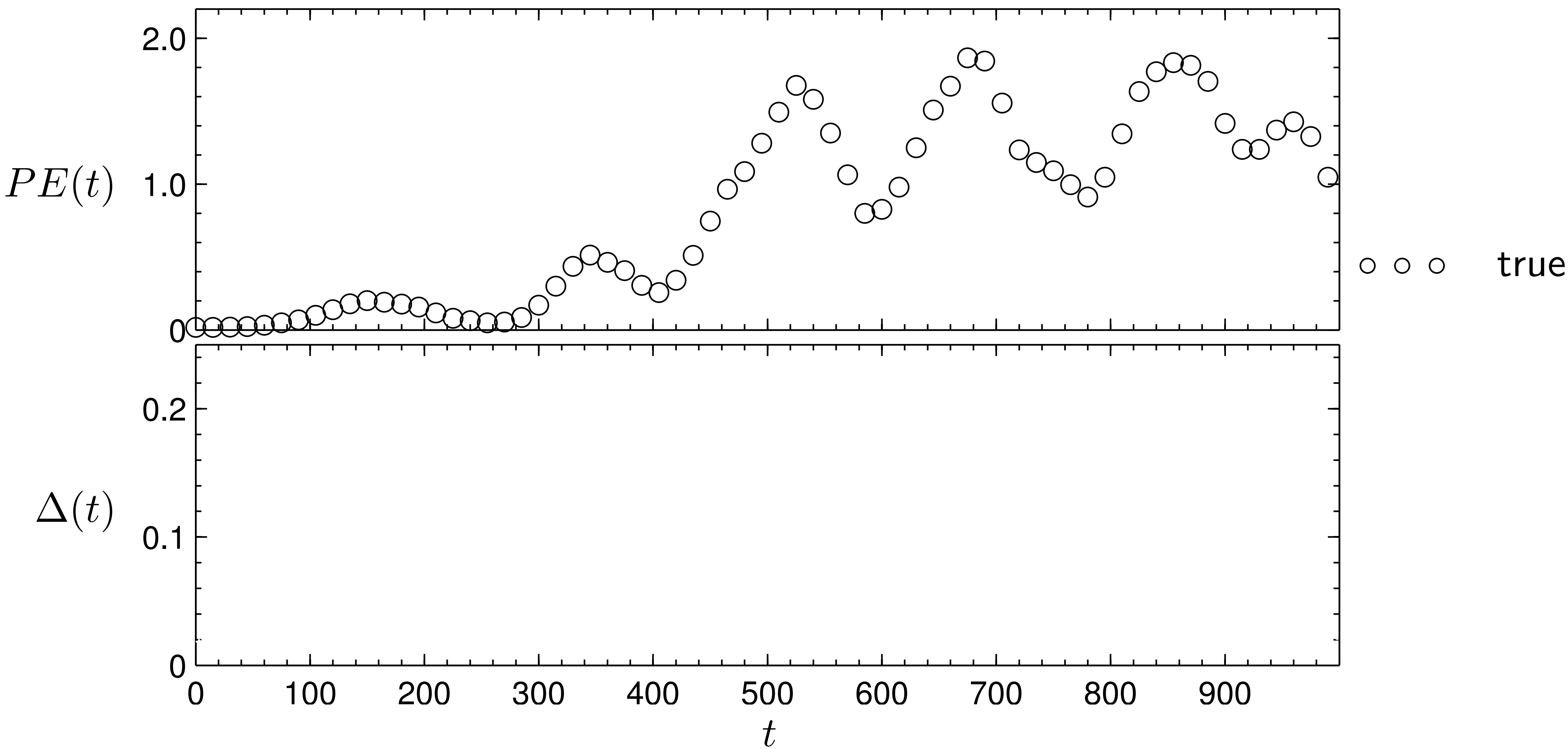


good results even for $n \gg q$

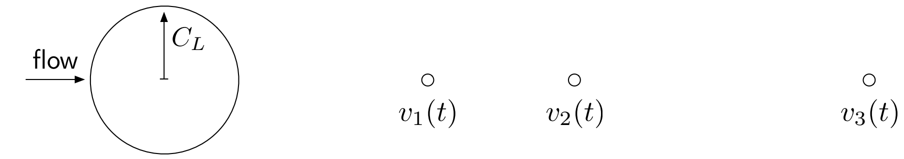
performance in time



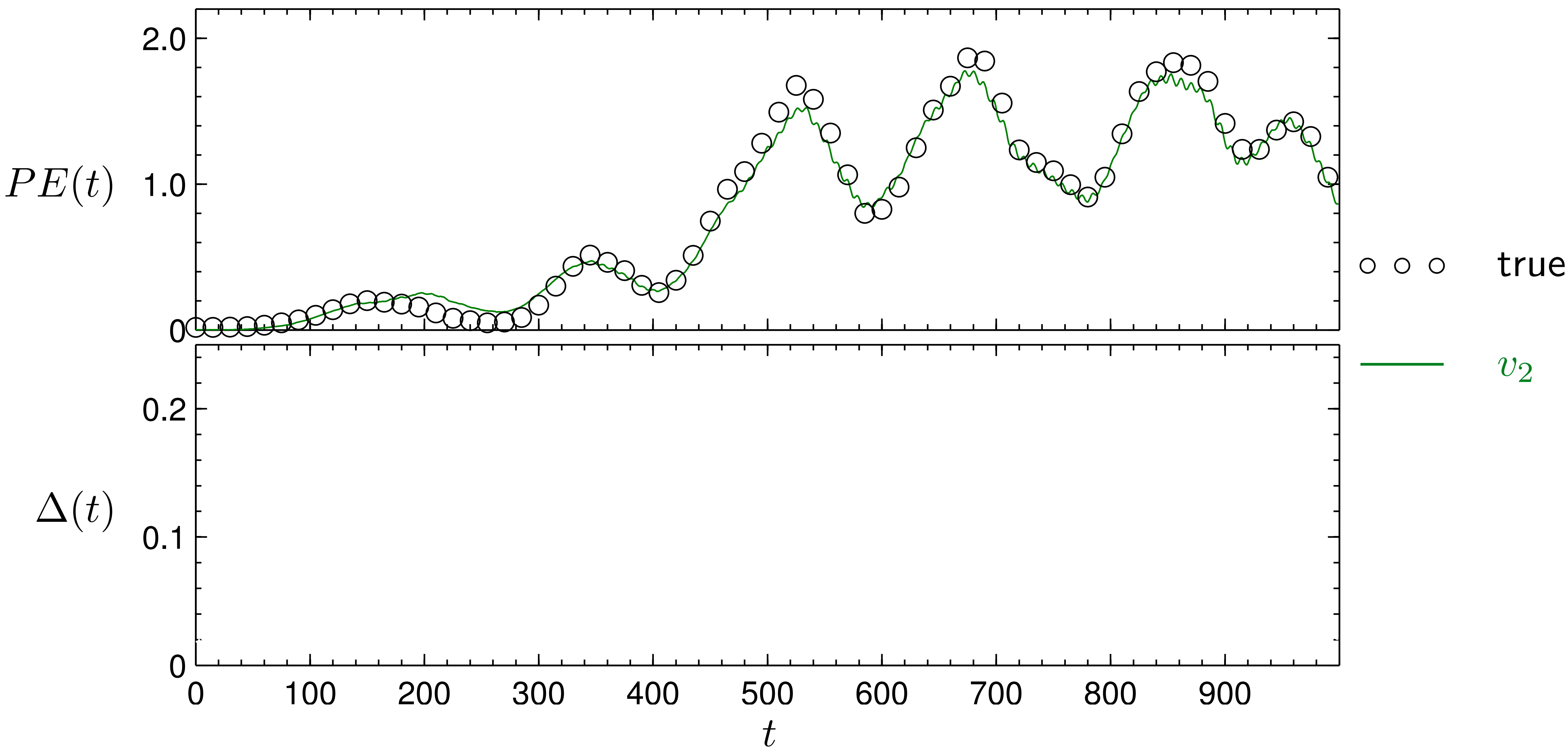
$$PE(t) = \frac{1}{2} \iint u_{_T}^2(x, y, t)\, \mathrm{d}x\, \mathrm{d}y \qquad u_{_T} = \sqrt{u^2 + v^2}$$



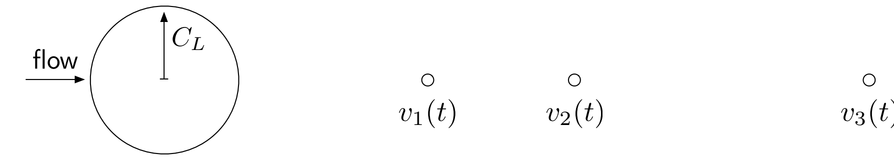
performance in time



$$PE(t) = \frac{1}{2} \iint u_T^2(x, y, t)\,\mathrm{d}x\,\mathrm{d}y \qquad u_T = \sqrt{u^2 + v^2}$$

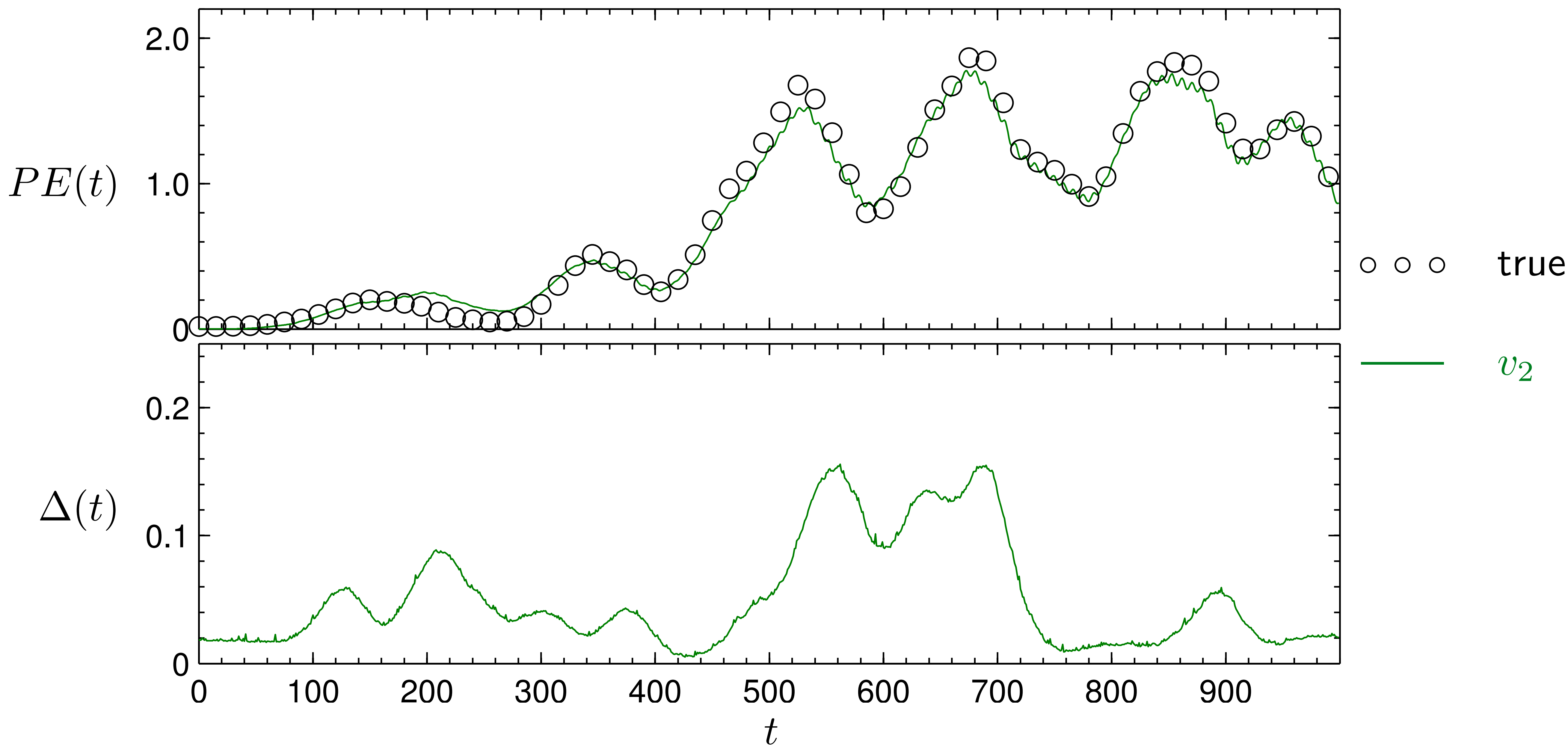


performance in time

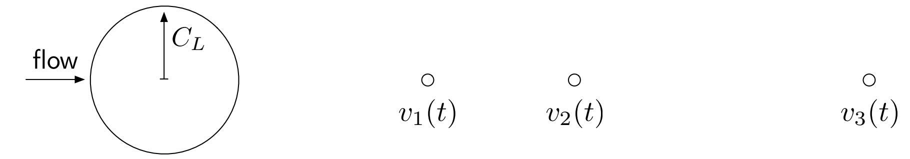


$$PE(t) = \frac{1}{2} \iint u_T^2(x, y, t)\,\mathrm{d}x\,\mathrm{d}y \qquad u_T = \sqrt{u^2 + v^2}$$

$$\Delta(t) = \frac{1}{2} \iint [\hat{u}_T(x, y, t) - u_T(x, y, t)]^2\,\mathrm{d}x\,\mathrm{d}y$$

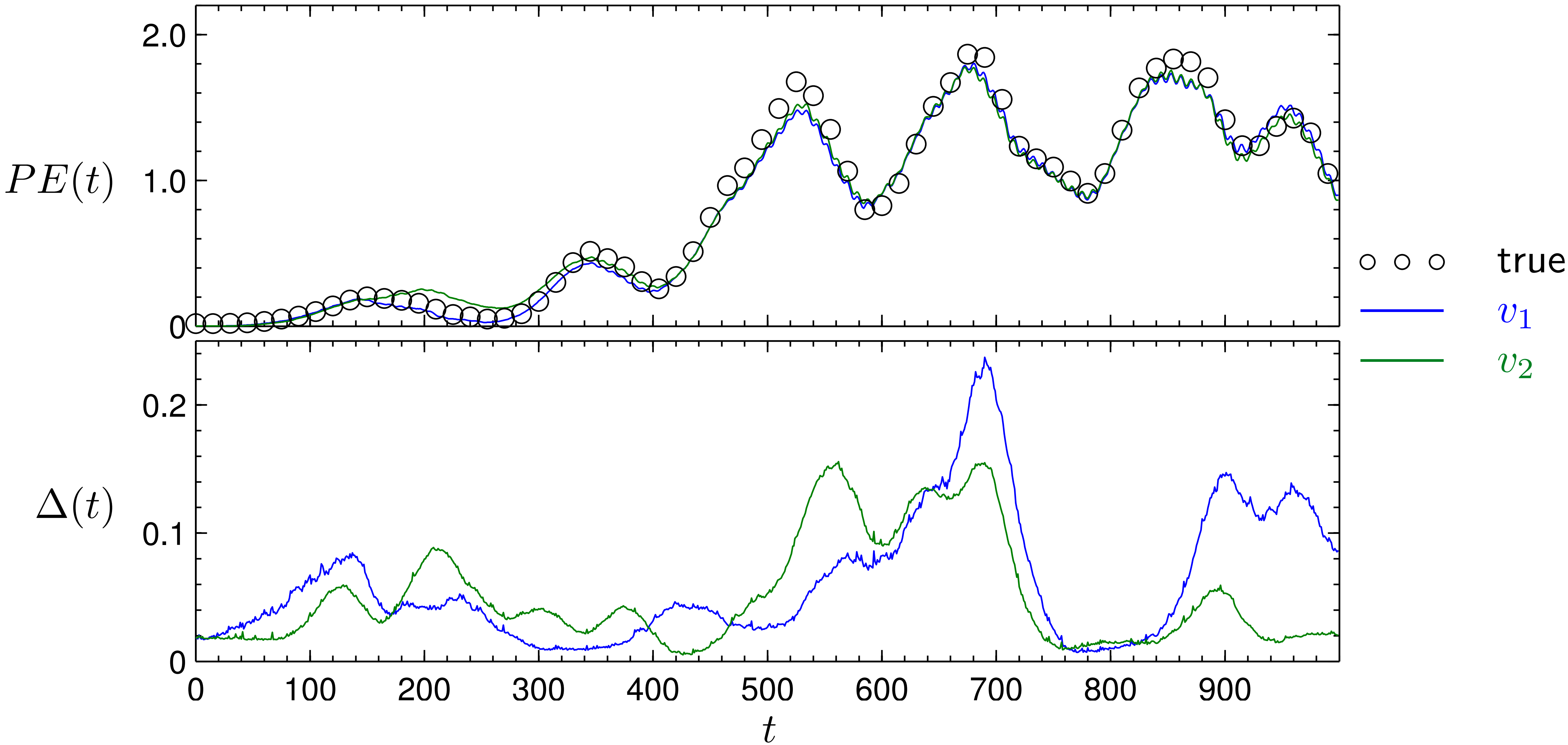


performance in time

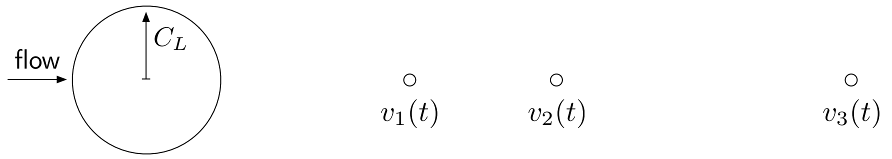


$$PE(t) = \frac{1}{2} \iint u_T^2(x,y,t)\,\mathrm{d}x\,\mathrm{d}y \qquad u_T = \sqrt{u^2 + v^2}$$

$$\Delta(t) = \frac{1}{2} \iint \left[\hat{u}_T(x,y,t) - u_T(x,y,t)\right]^2 \mathrm{d}x\,\mathrm{d}y$$

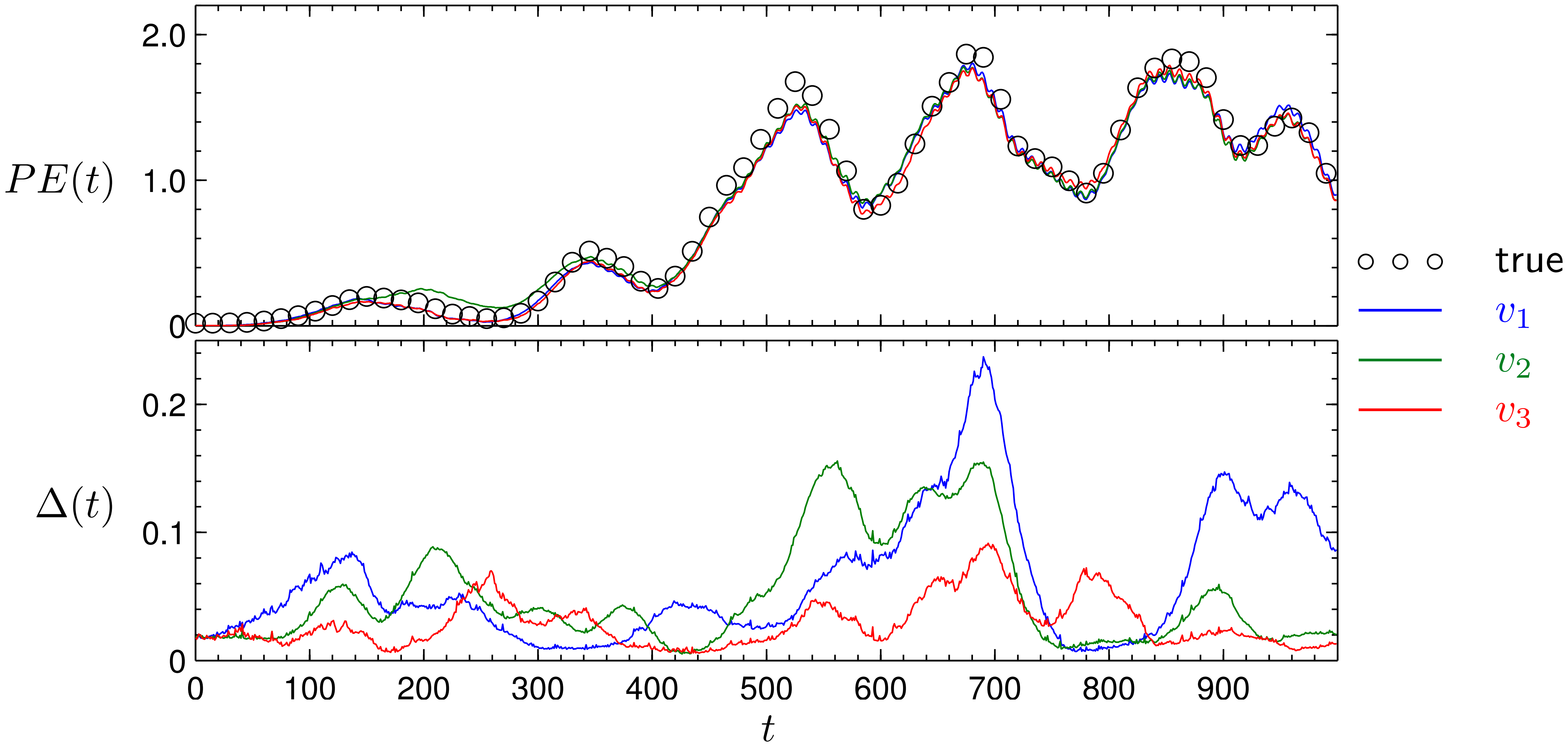


performance in time

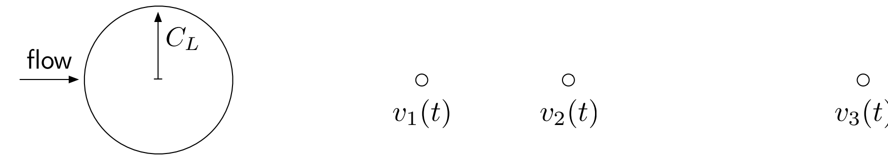


$$PE(t) = \frac{1}{2}\iint u_T^2(x,y,t)\,\mathrm{d}x\,\mathrm{d}y \qquad u_T = \sqrt{u^2+v^2}$$

$$\Delta(t) = \frac{1}{2}\iint \left[\hat{u}_T(x,y,t) - u_T(x,y,t)\right]^2 \mathrm{d}x\,\mathrm{d}y$$

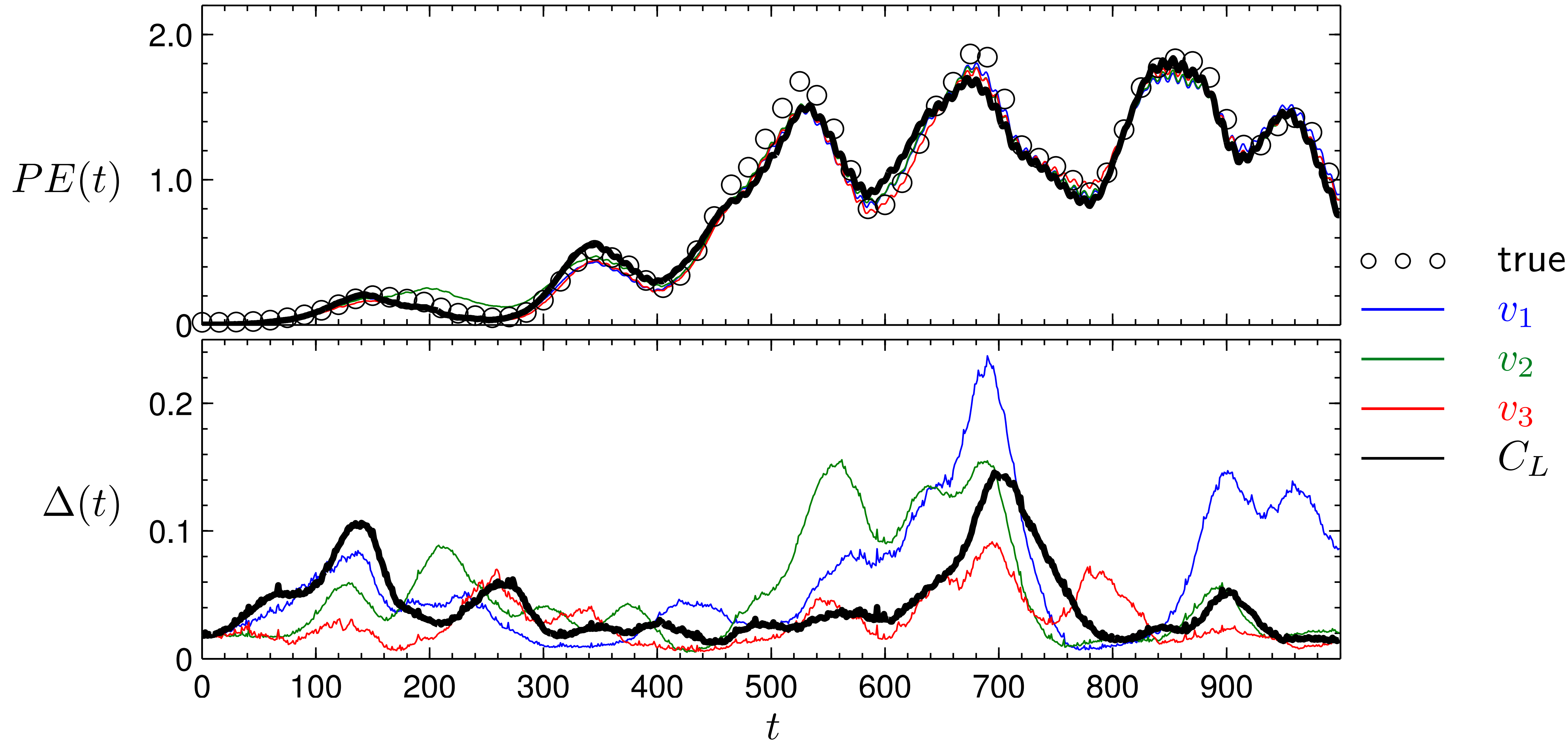


performance in time

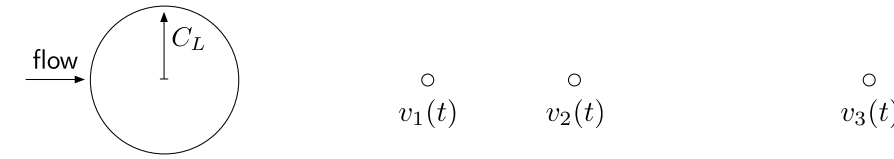


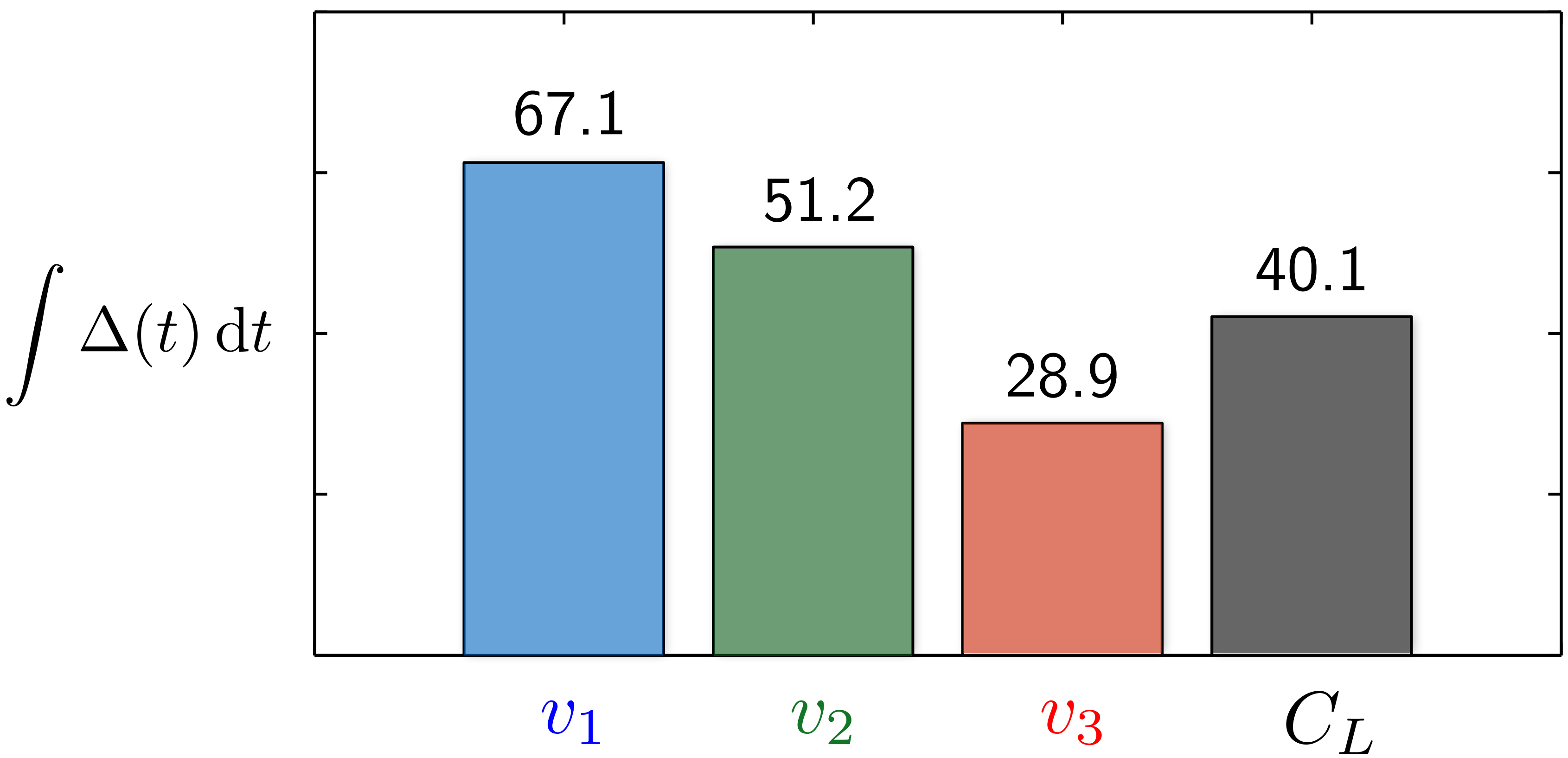
$$PE(t) = \frac{1}{2}\iint u_T^2(x,y,t)\,\mathrm{d}x\,\mathrm{d}y \qquad u_T = \sqrt{u^2+v^2}$$

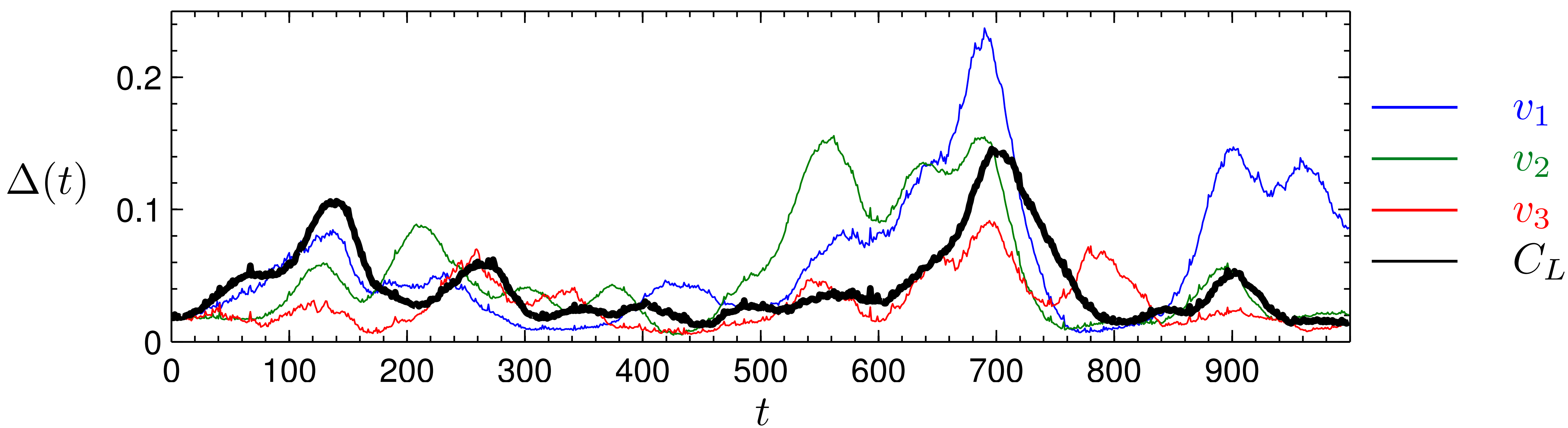
$$\Delta(t) = \frac{1}{2}\iint [\hat{u}_T(x,y,t) - u_T(x,y,t)]^2\,\mathrm{d}x\,\mathrm{d}y$$



performance in time





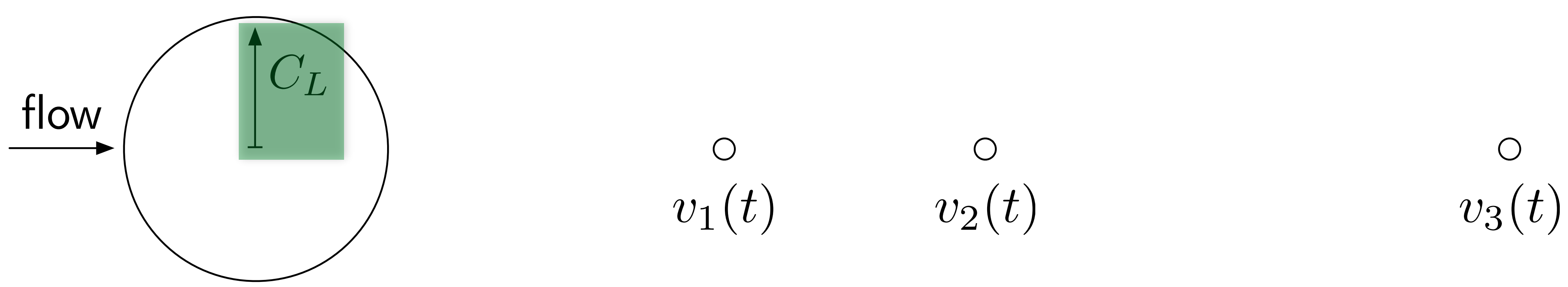


results

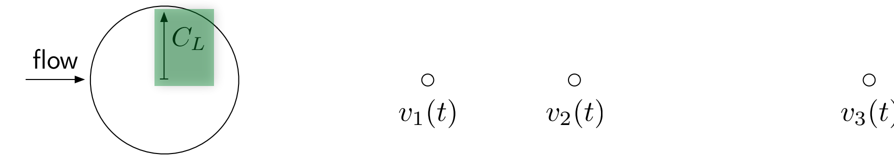
- we estimate the entire flow using

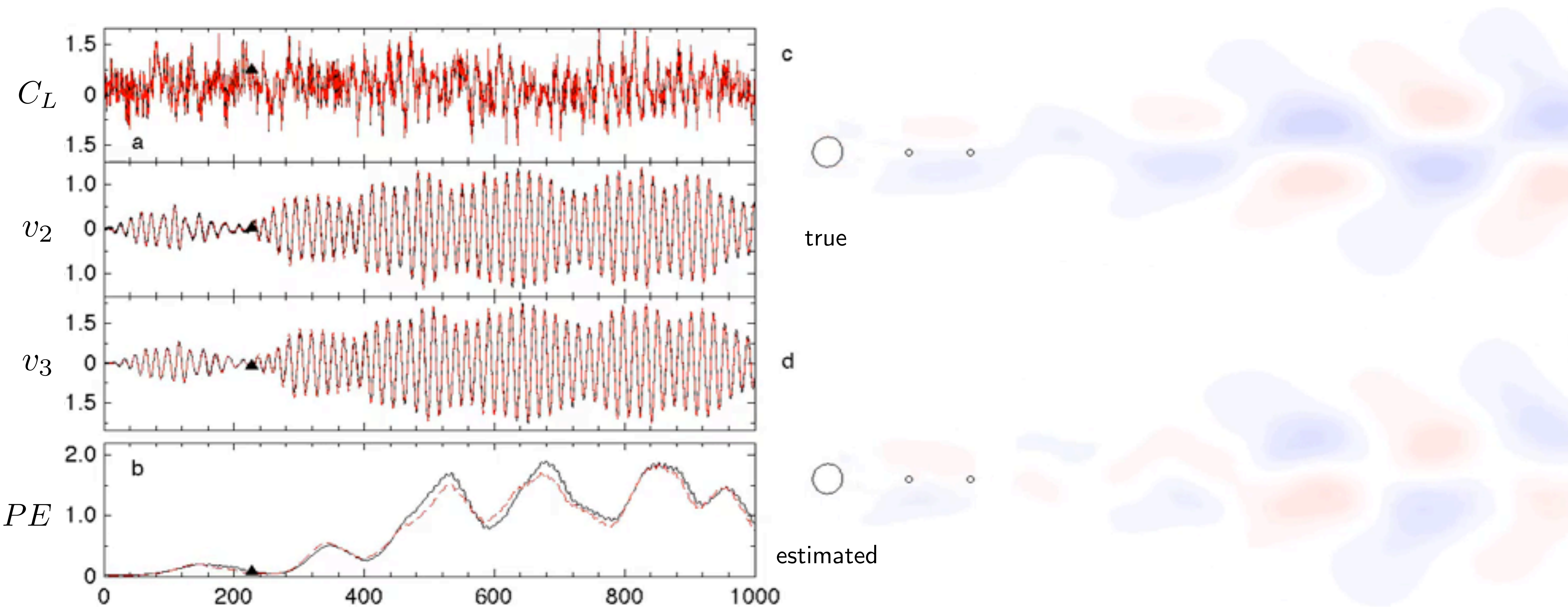
  i. transverse velocity at sensor two only

  ii. lift force only



lift force only





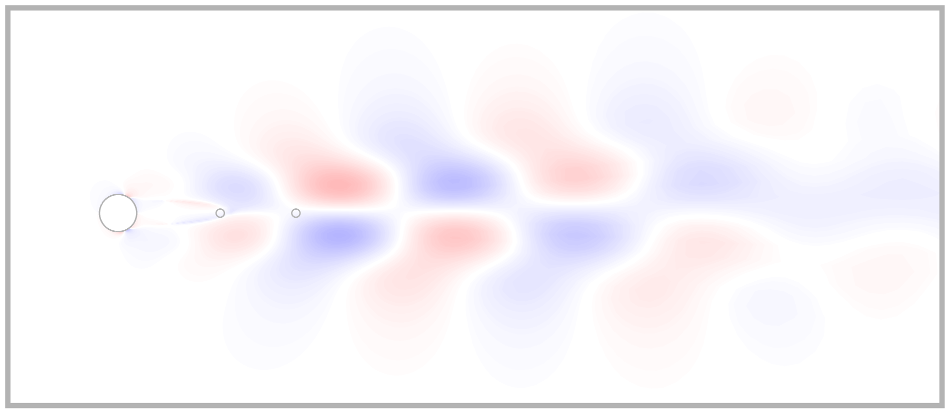
—— true

– – – estimated

'local' or 'integral' sensing can work

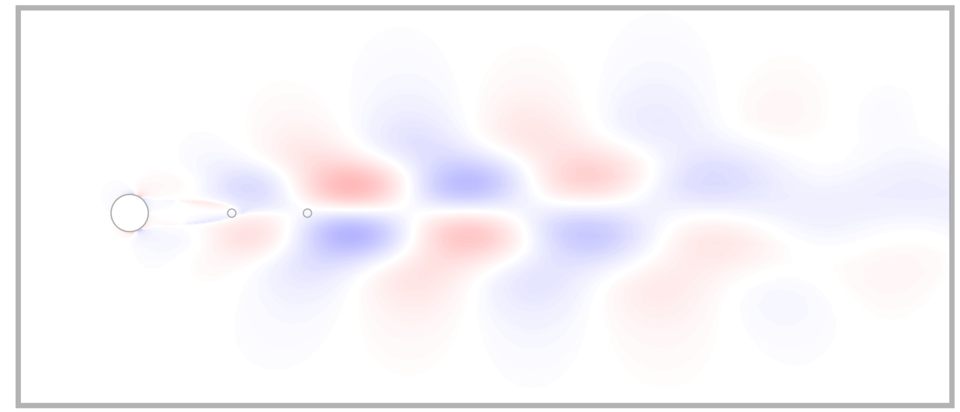
some applications

1. estimation

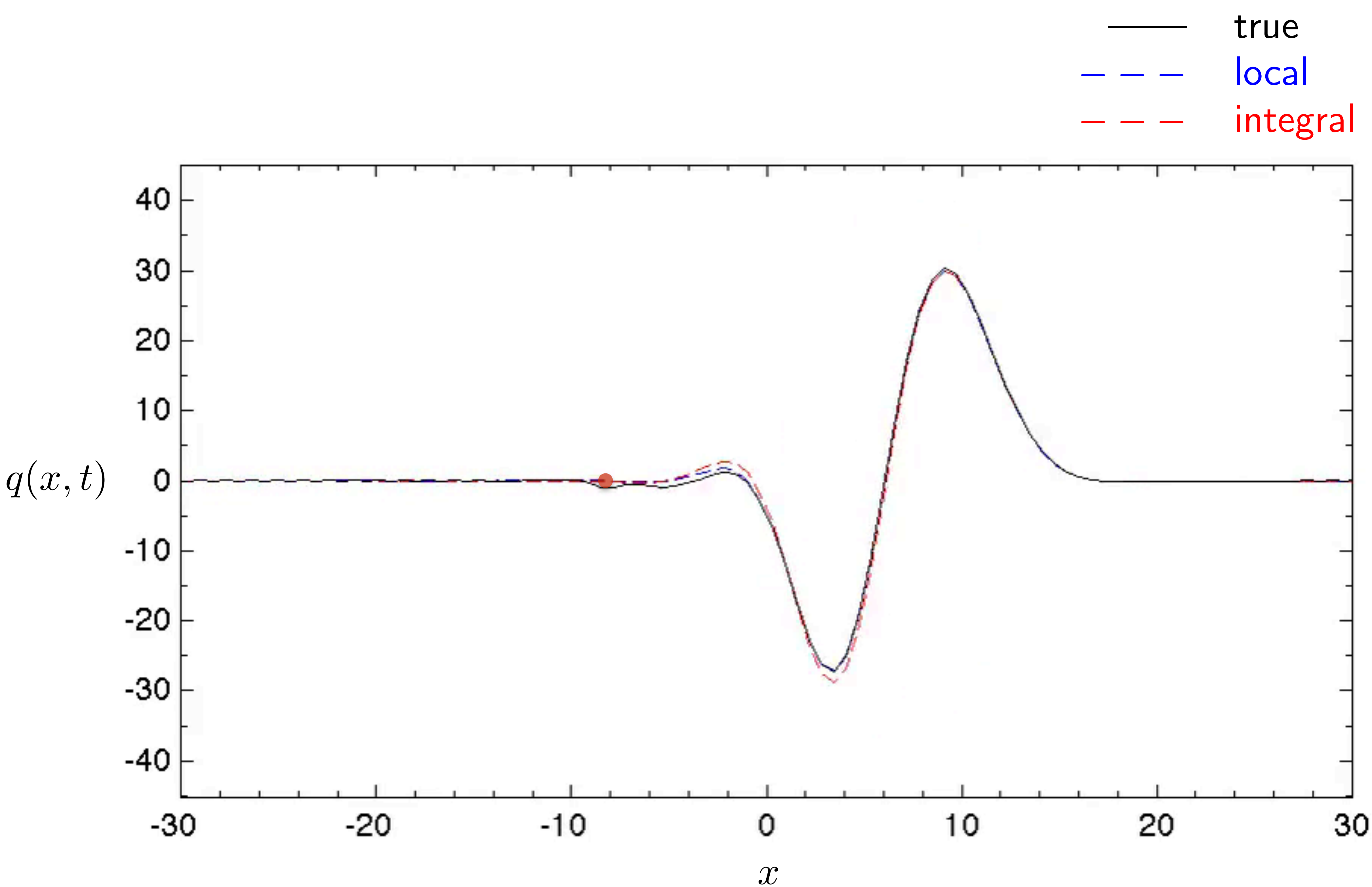


$$\frac{\partial q}{\partial t}(x,t) = \left(-\nu\frac{\partial}{\partial x} + \gamma\frac{\partial^2}{\partial x^2} + \mu(x)\right) q(x,t)$$

2. control

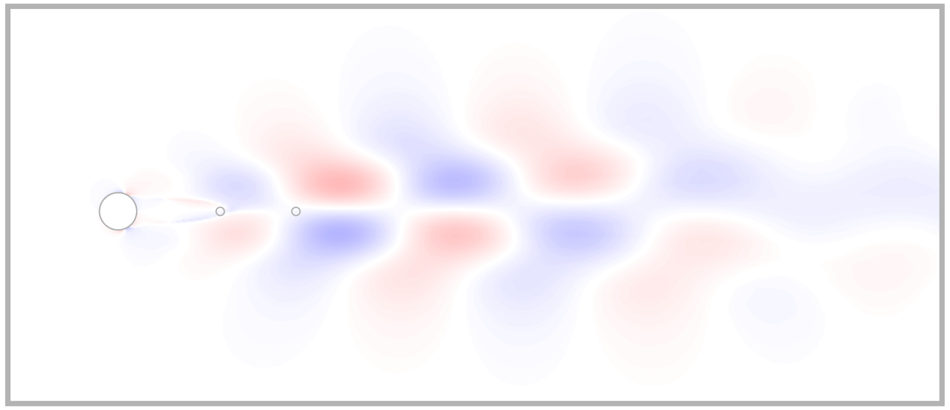


Ginzburg-Landau system



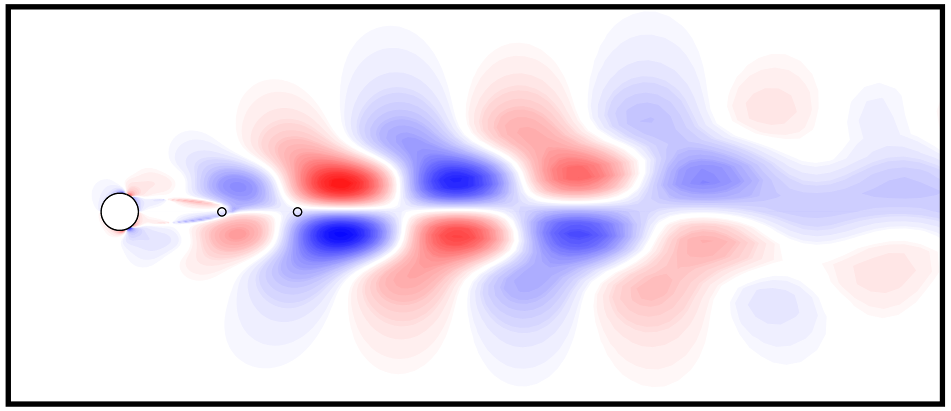
some applications

1. estimation

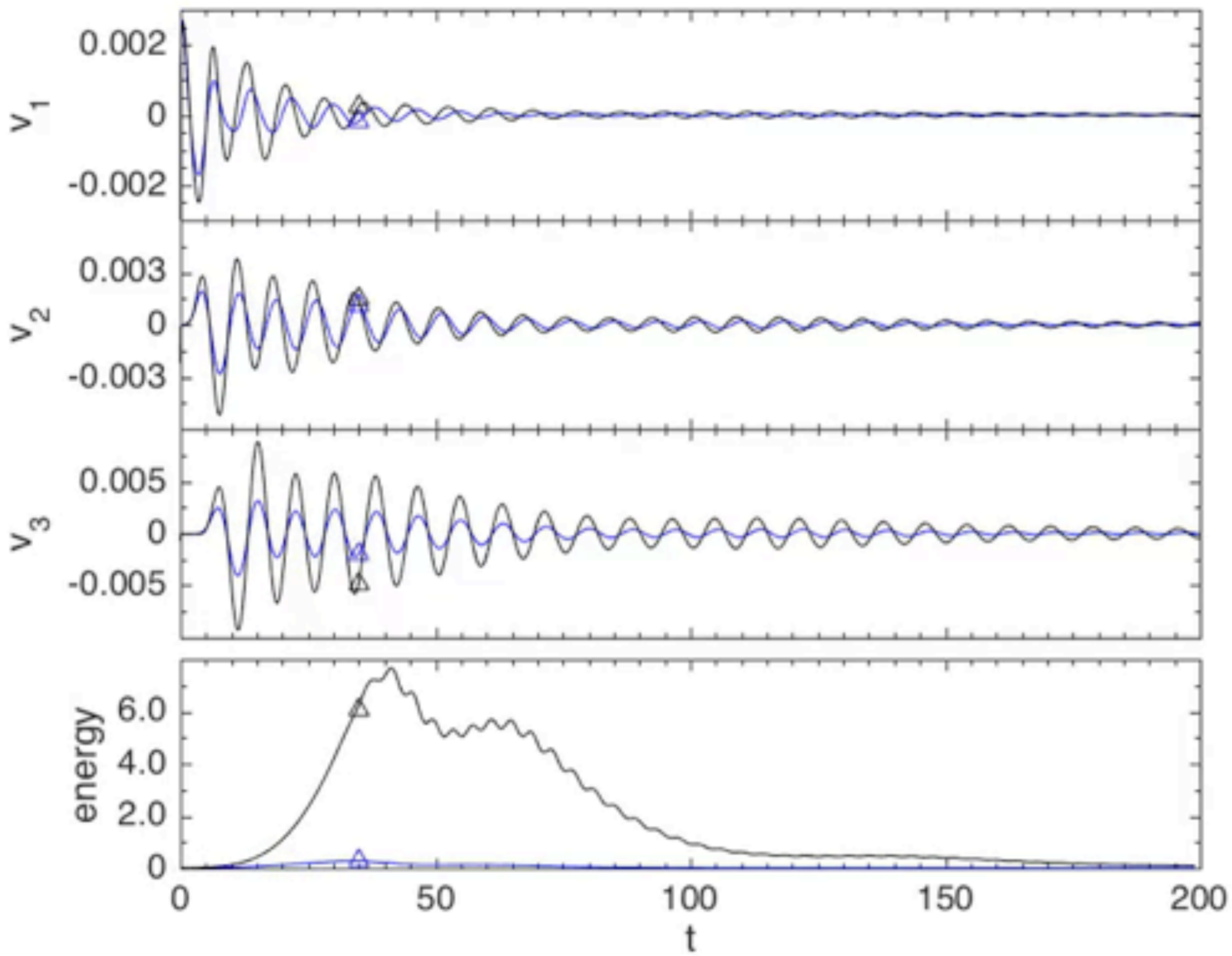


$$\frac{\partial q}{\partial t}(x,t) = \left(-\nu \frac{\partial}{\partial x} + \gamma \frac{\partial^2}{\partial x^2} + \mu(x)\right) q(x,t)$$

2. control



a single controller can be found that stabilizes for all $50 \leq \mathrm{Re} \leq 100$

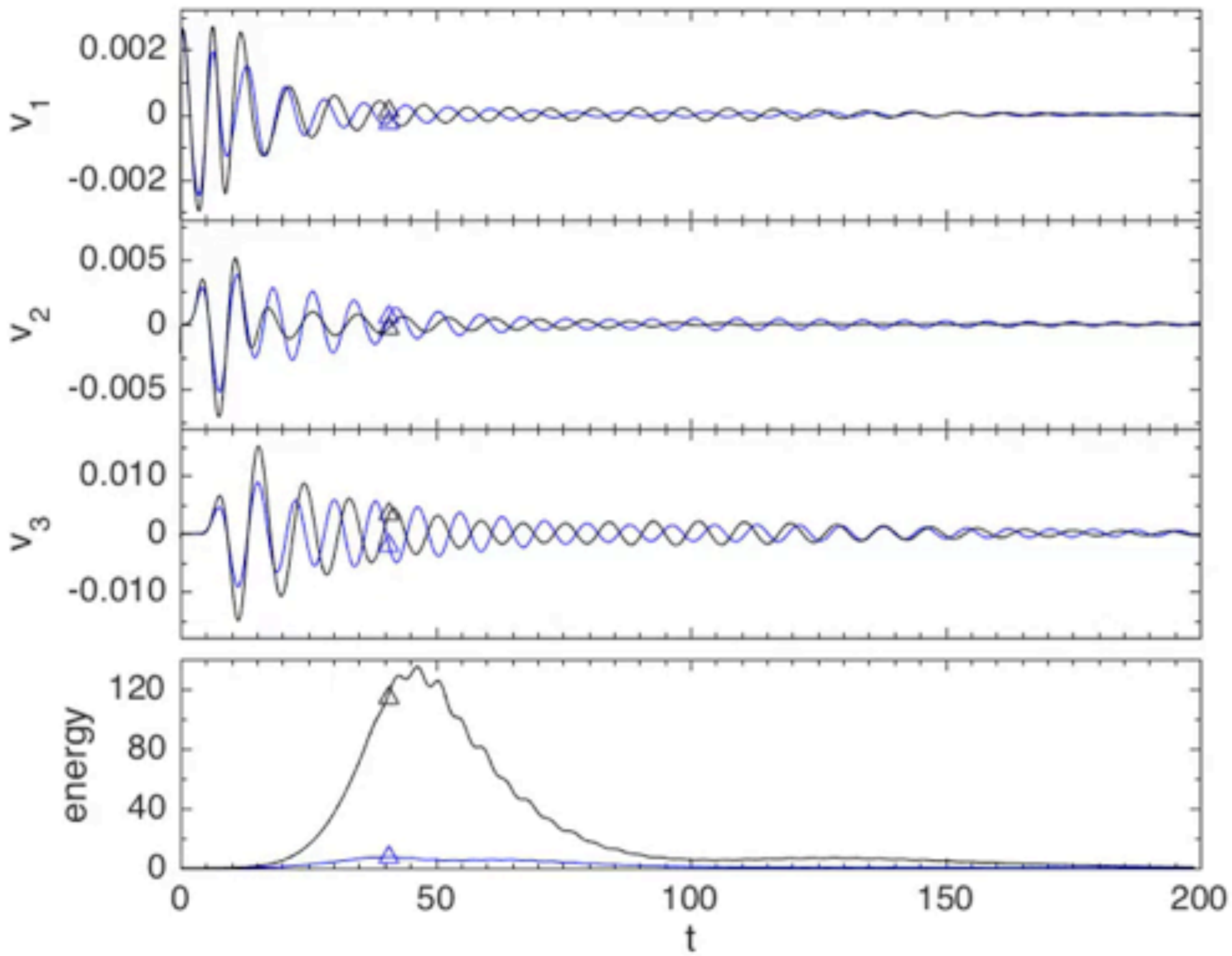


Re = 50

Re = 75

robust controllers are (very) forgiving

a single controller can be found that stabilizes for all $50 \leq \mathrm{Re} \leq 100$



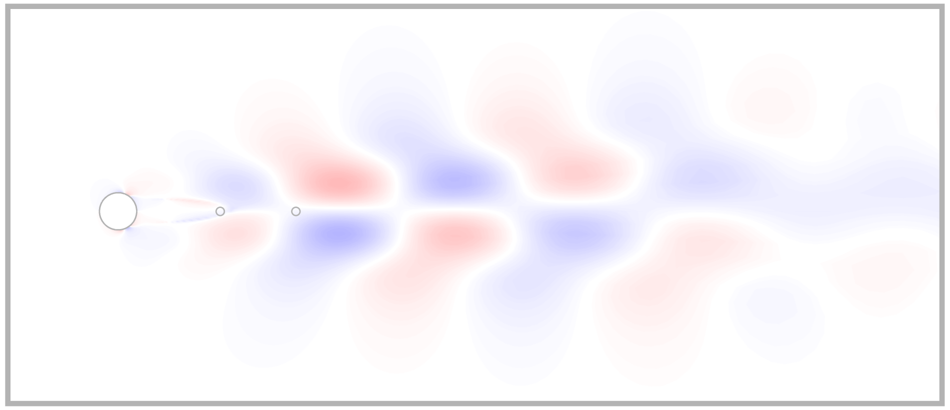
Re = 75

Re = 100

robust controllers are (very) forgiving

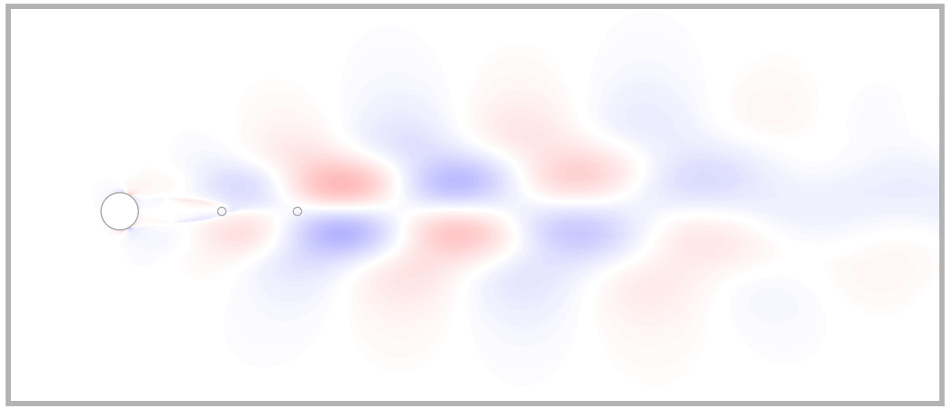
some applications

1. estimation

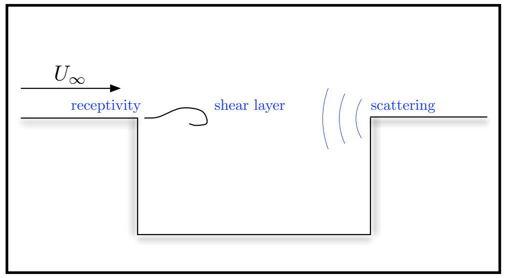


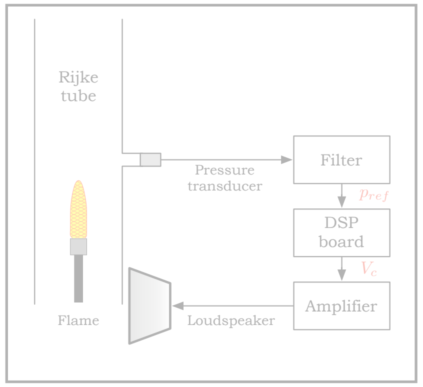
$$\frac{\partial q}{\partial t}(x,t) = \left(-\nu \frac{\partial}{\partial x} + \gamma \frac{\partial^2}{\partial x^2} + \mu(x)\right) q(x,t)$$

2. control



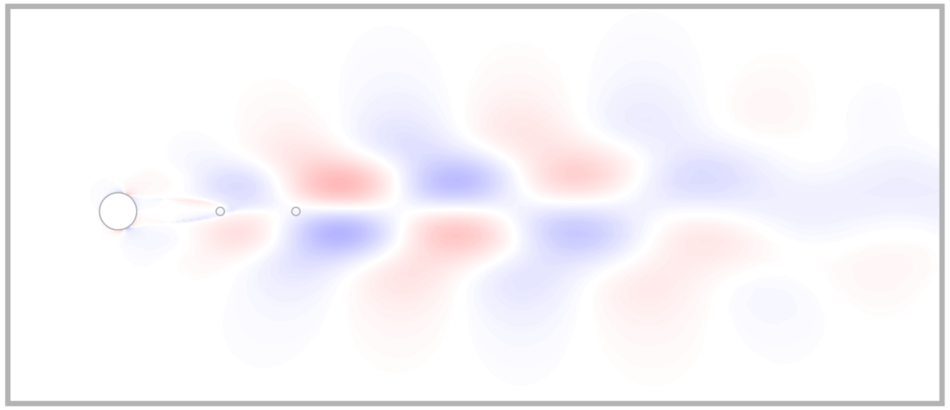
$$\frac{\partial q}{\partial t}(x,t) = \left(-\nu \frac{\partial}{\partial x} + \gamma \frac{\partial^2}{\partial x^2} + \mu(x)\right) q(x,t)$$





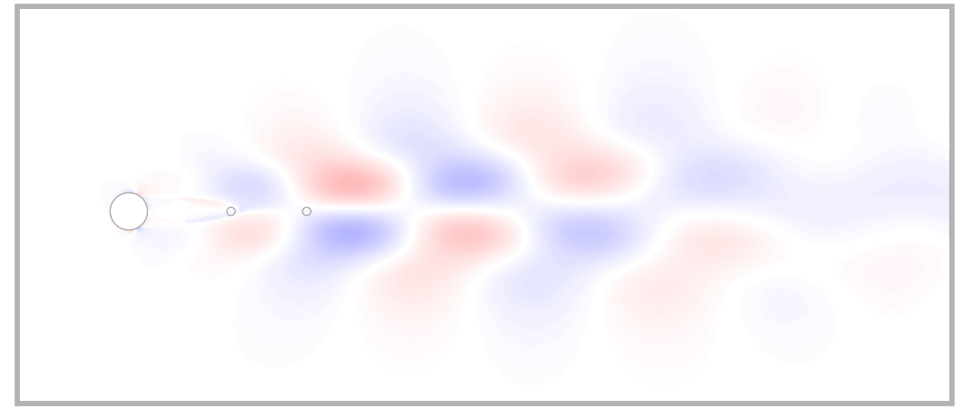
some applications

1. estimation

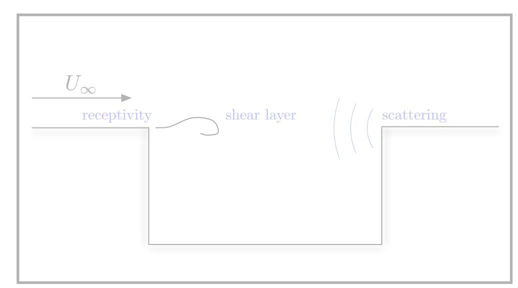


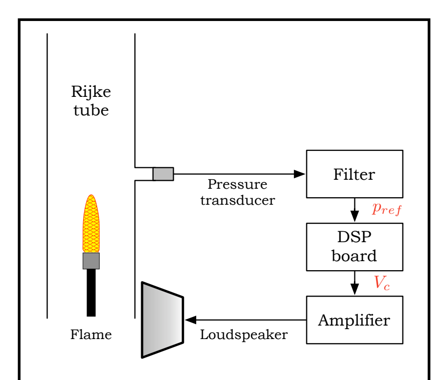
$$\frac{\partial q}{\partial t}(x,t) = \left(-\nu\frac{\partial}{\partial x} + \gamma\frac{\partial^2}{\partial x^2} + \mu(x)\right) q(x,t)$$

2. control

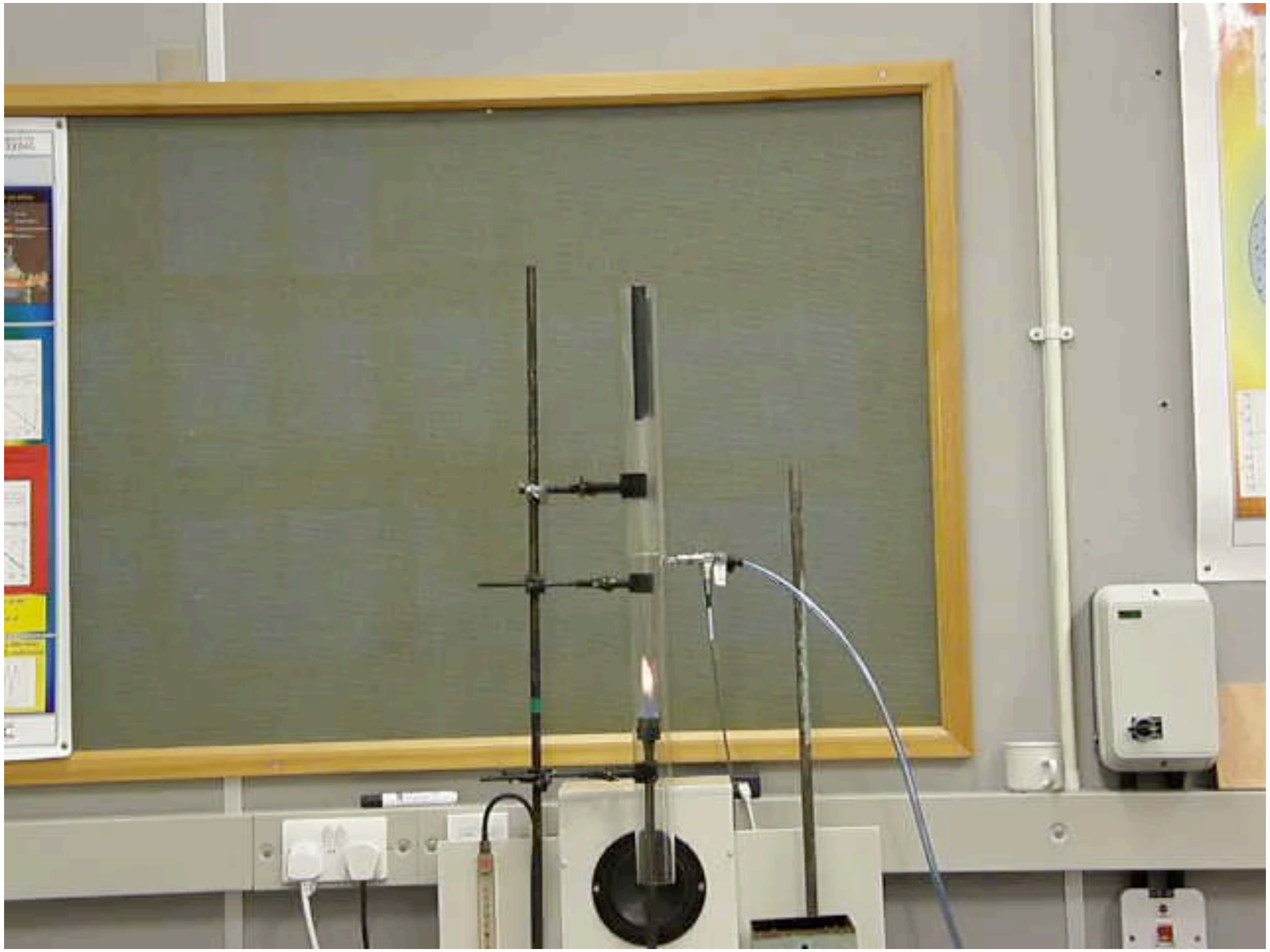


$$\frac{\partial q}{\partial t}(x,t) = \left(-\nu\frac{\partial}{\partial x} + \gamma\frac{\partial^2}{\partial x^2} + \mu(x)\right) q(x,t)$$



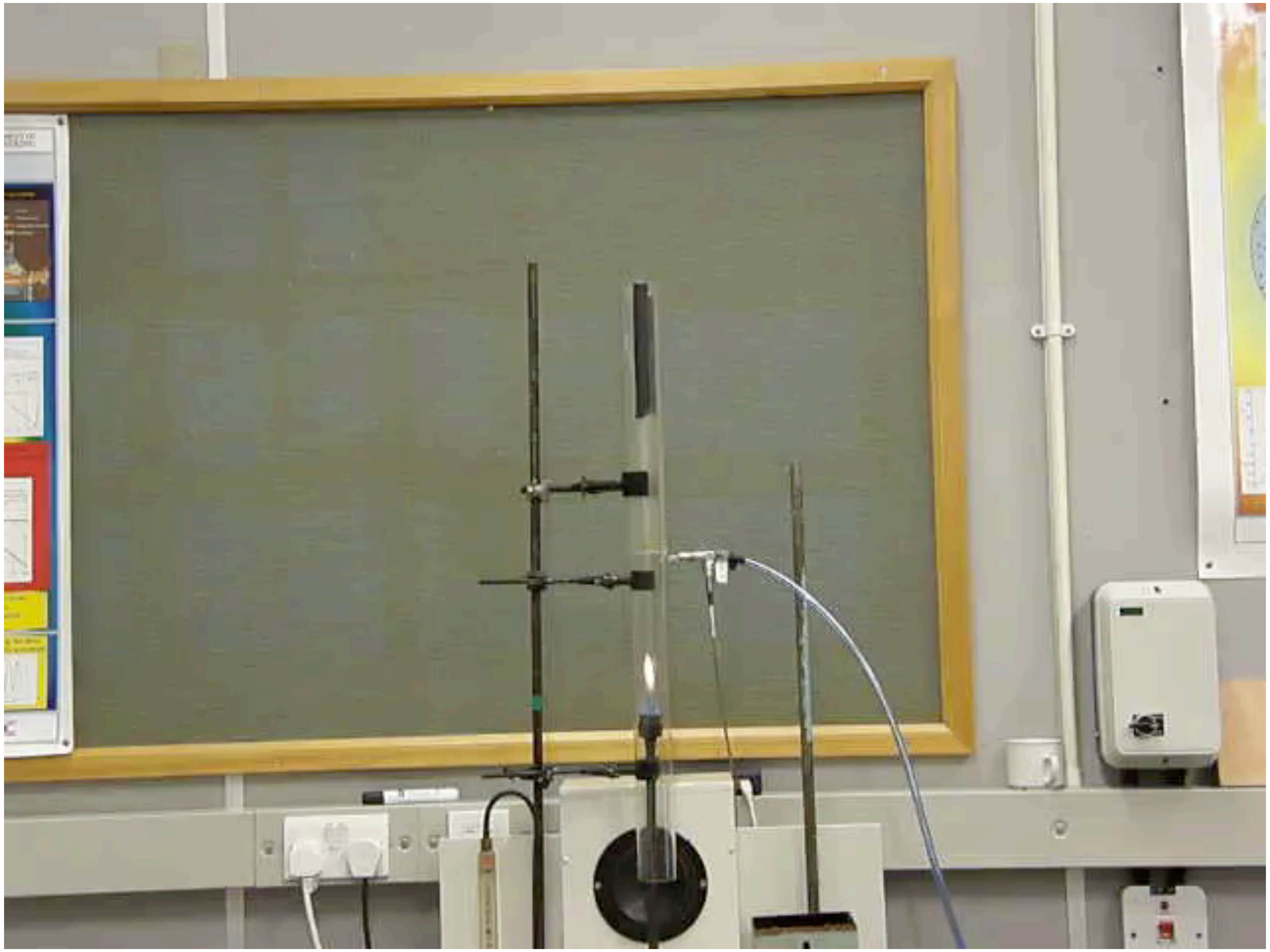


without control



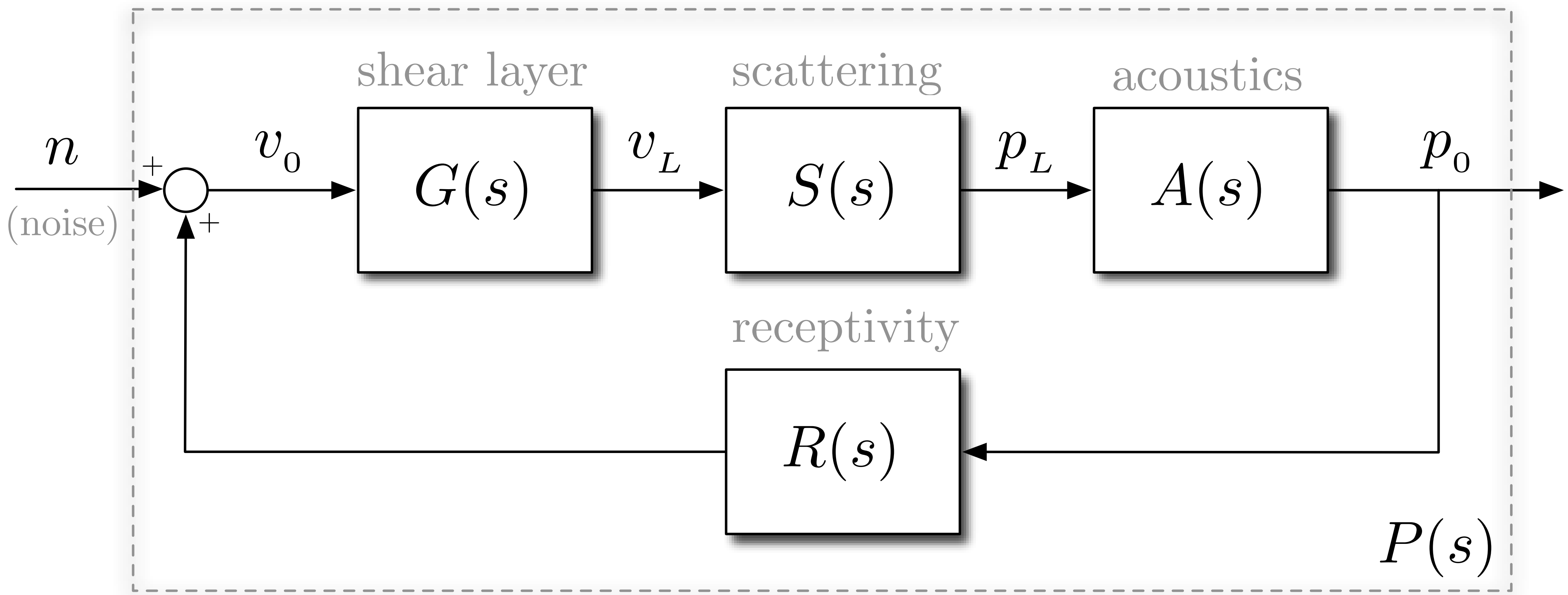
robust controllers are (very) forgiving

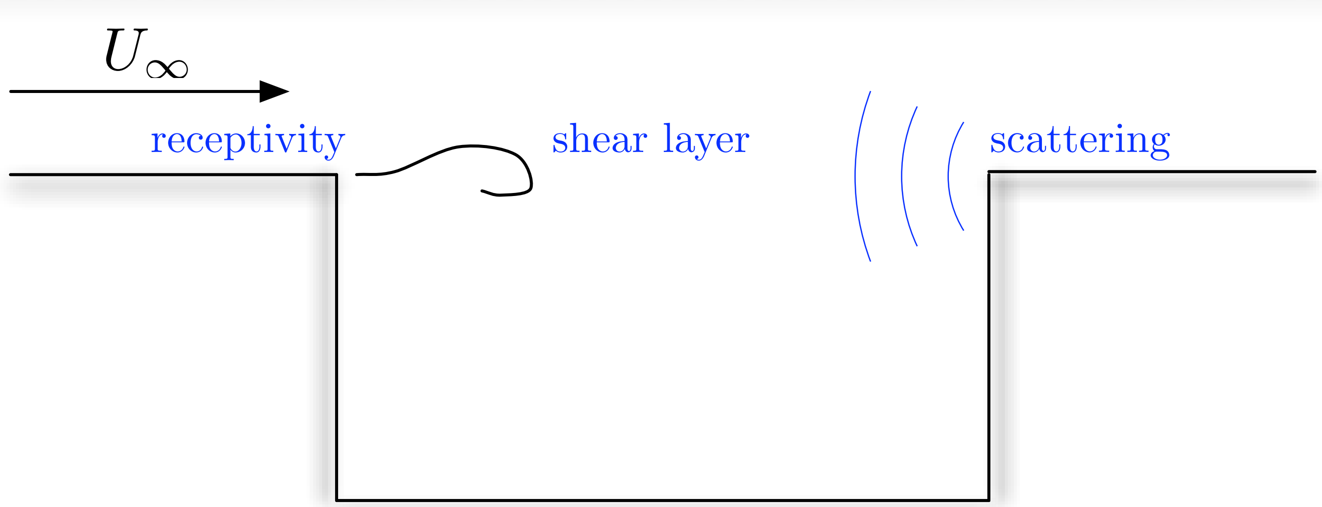
with control



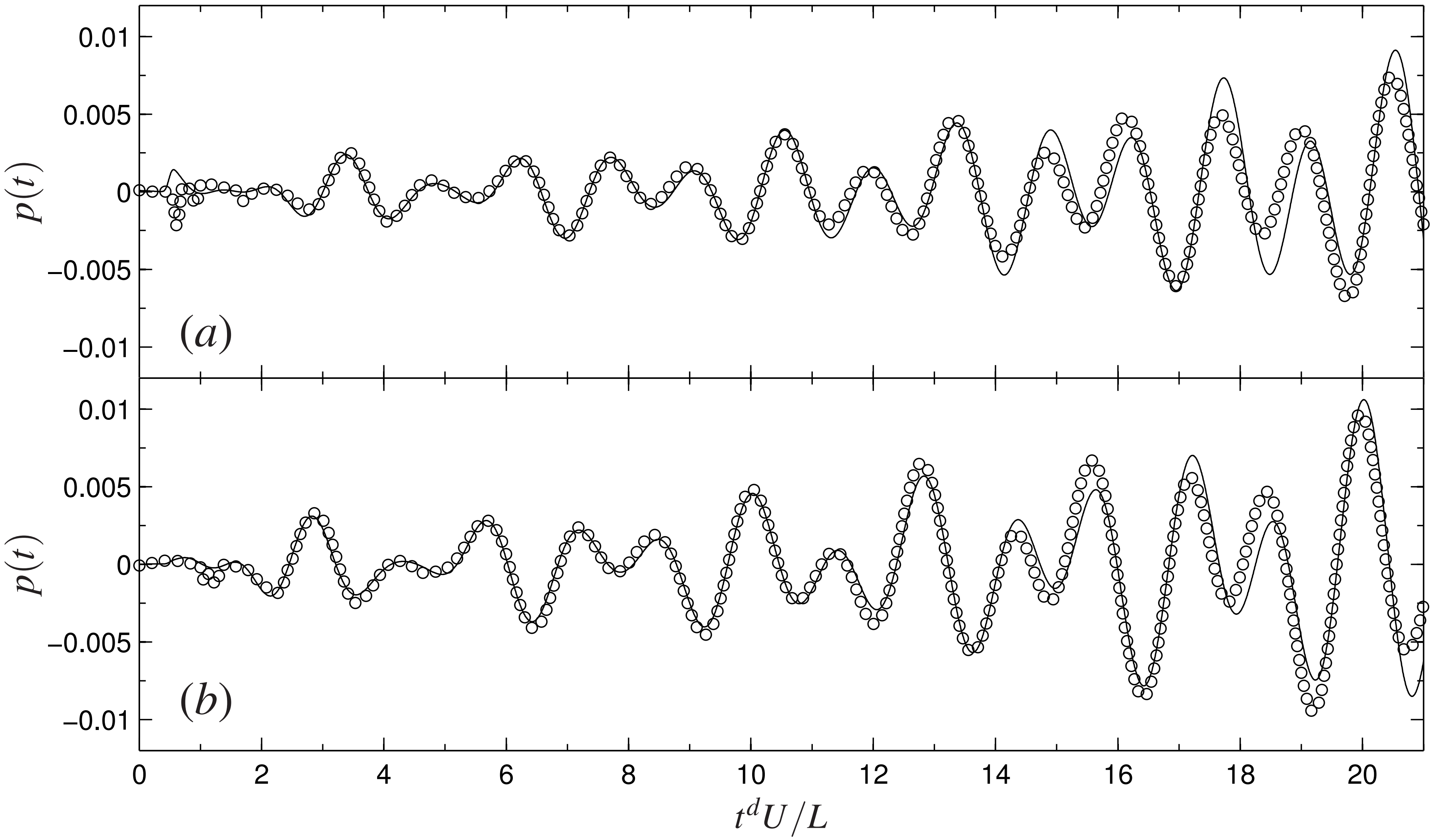
robust controllers are (very) forgiving

compressible cavity oscillations

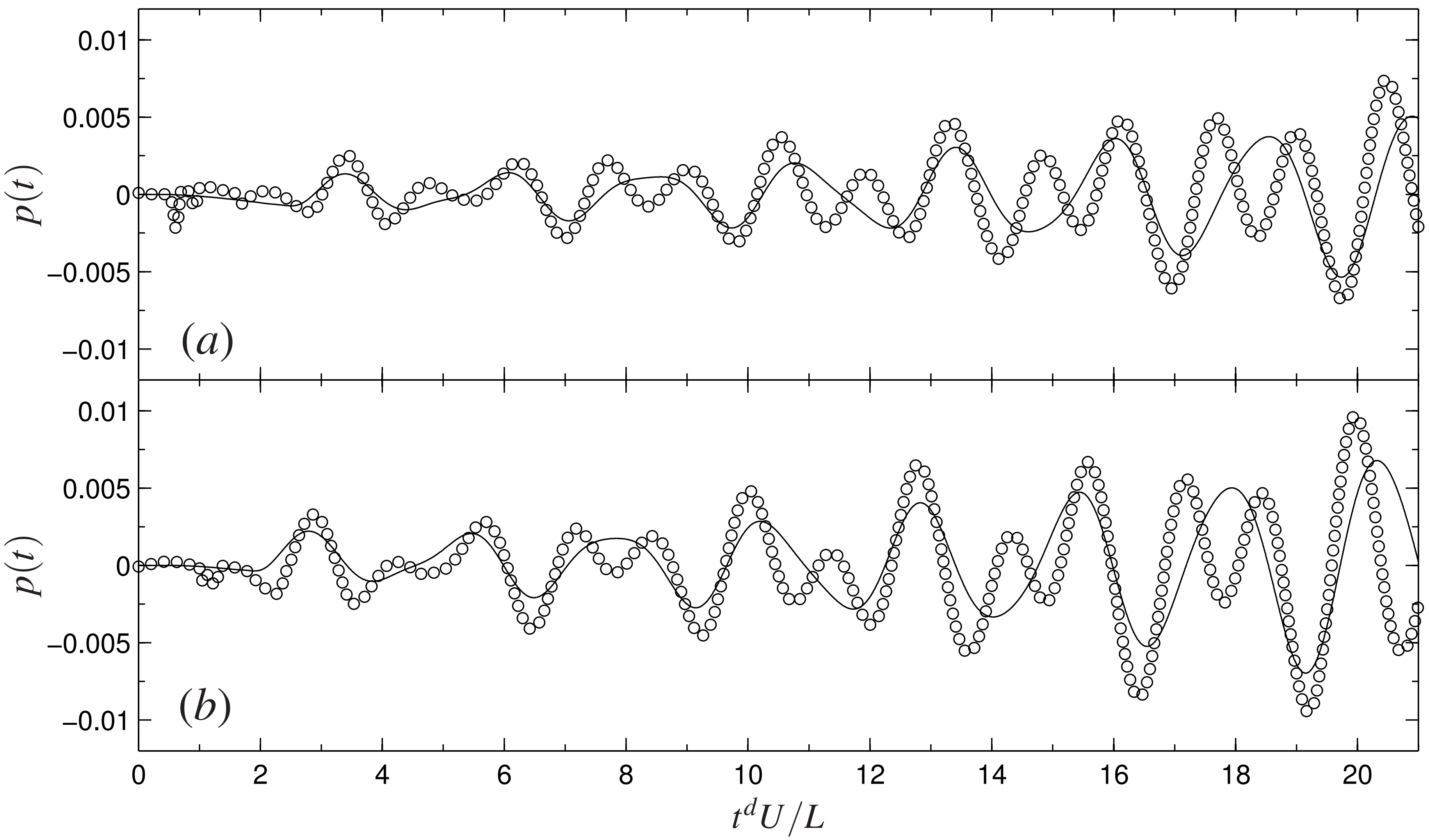




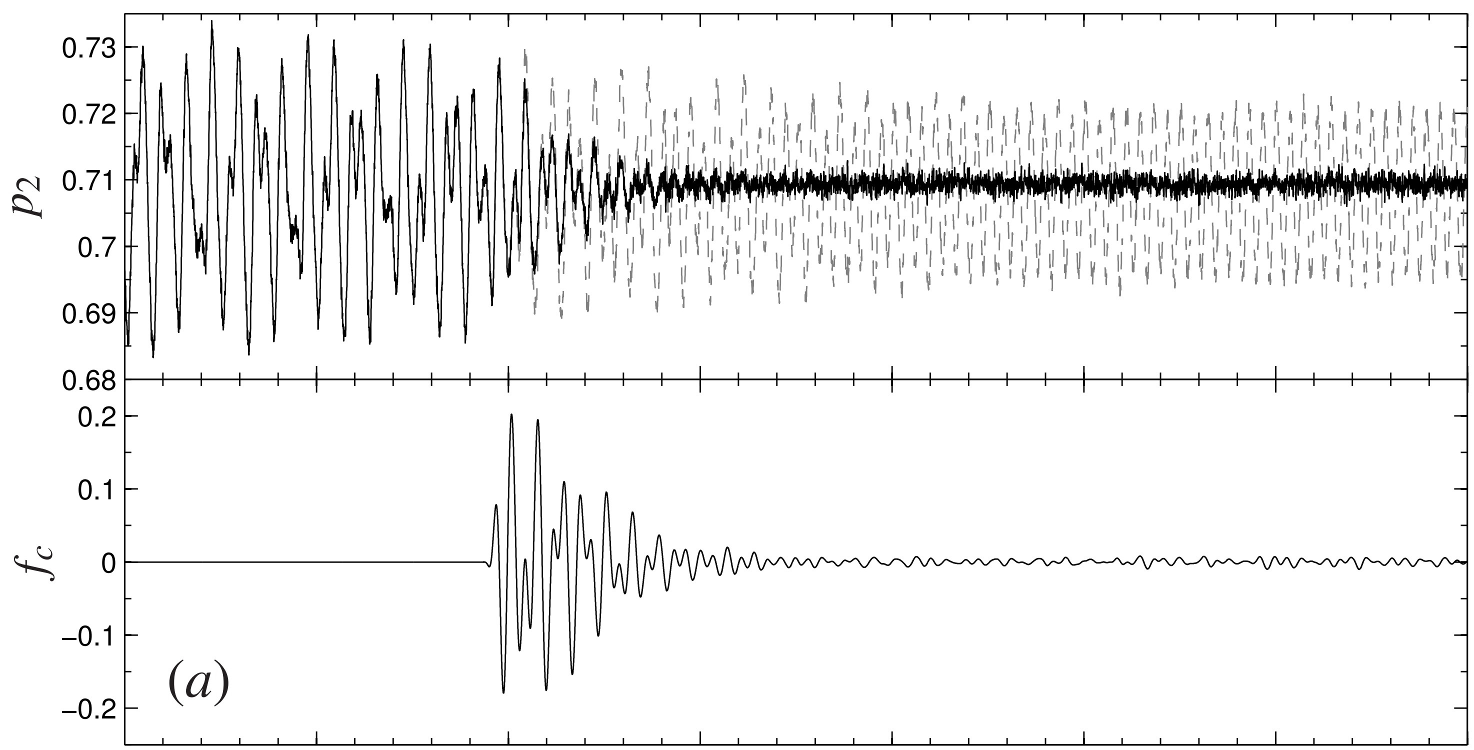
linear model 1: measured directly —impulse response



linear model 2: simple constituent models —impulse response

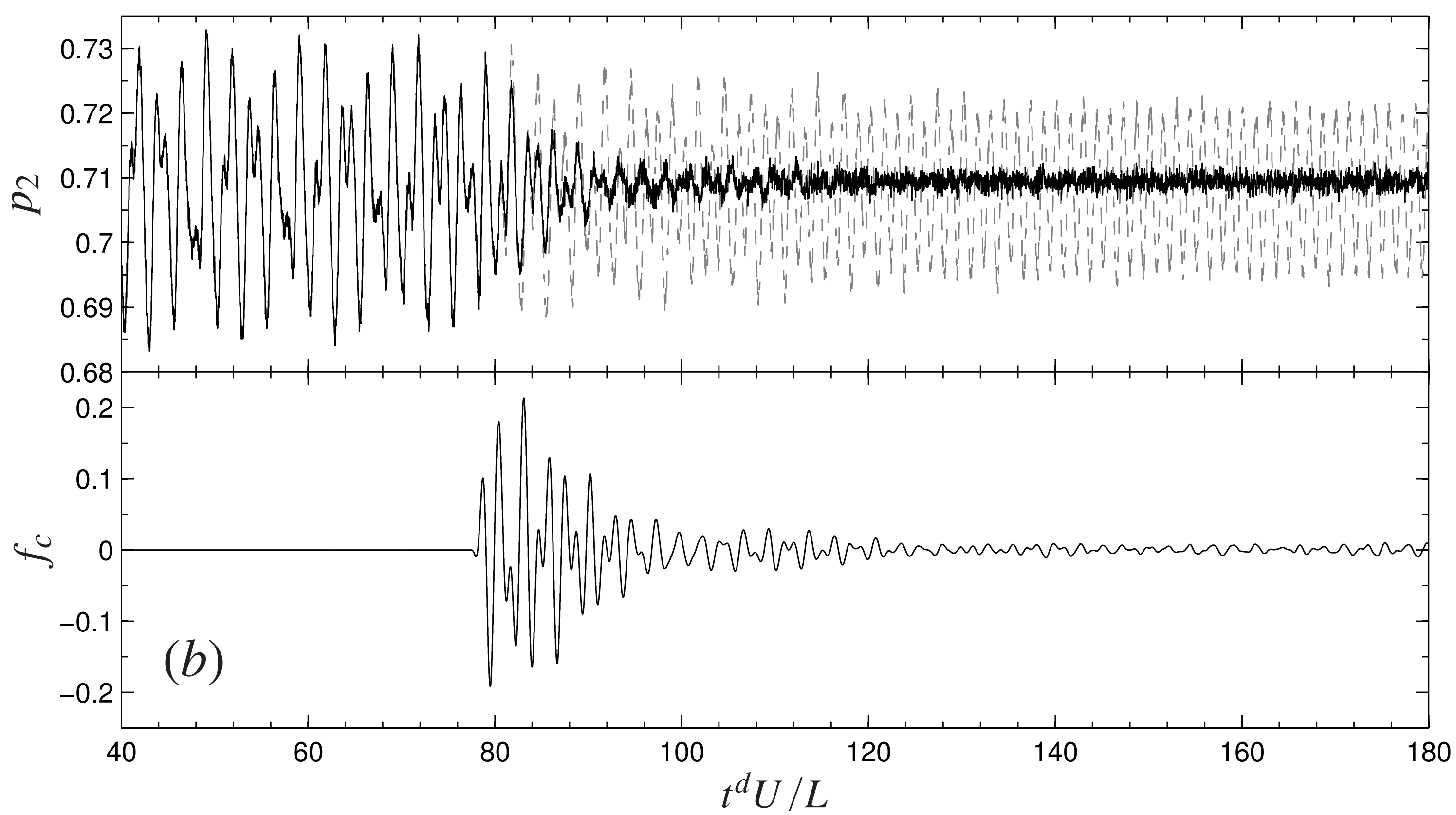


good enough for control $\neq$ good enough for modelling

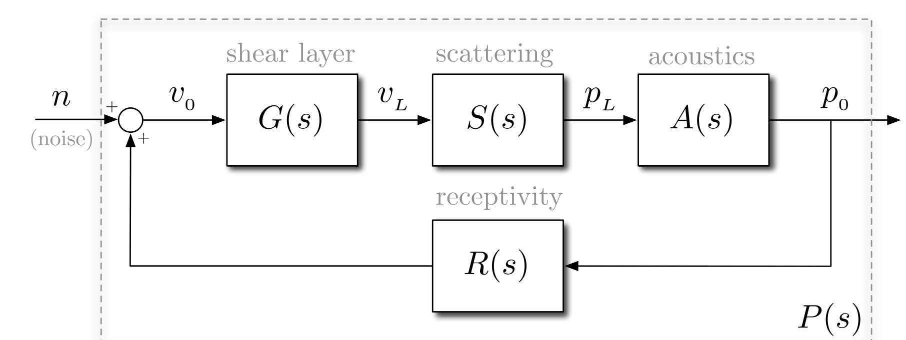


controller designed using
<u>linear model 1</u>

$$
u(t) \rightarrow \boxed{\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) \end{aligned}} \rightarrow y(t)
$$



controller designed using
<u>linear model 2</u>



good enough for control $\neq$ good enough for modelling