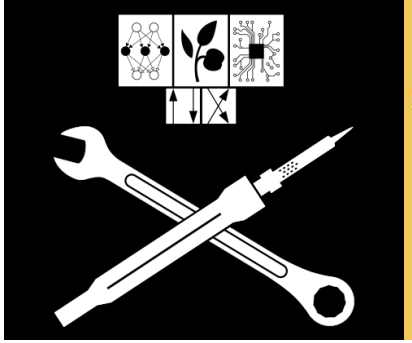
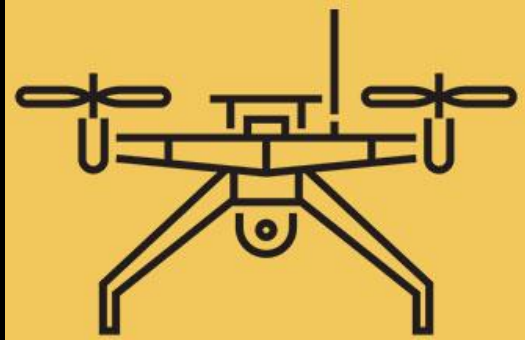
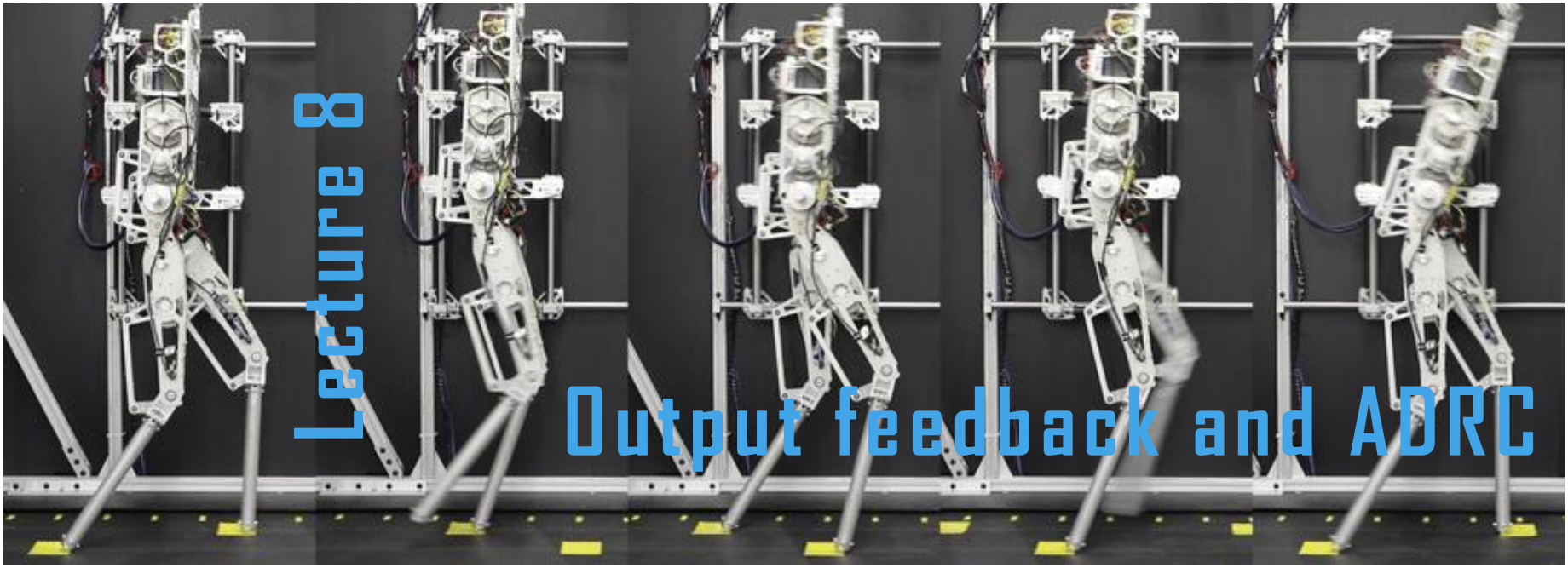






Jerome Jouffroy, Professor
jerome@sdu.dk

SDU

Today's lecture

- Output feedback using an observer
- The separation principle
- Tracking control and the general controller structure
- ADRC: estimating disturbances
- ADRC: Extended-State Observers (ESO)
- ADRC: closing the loop and implementation

Output feedback using an observer

When we do not measure the state, we cannot use

$$u(t) = -\mathbf{K}\mathbf{x}(t)$$

BUT we have an observer

$$\begin{cases} \dot{\hat{\mathbf{x}}}(t) = \mathbf{A}\hat{\mathbf{x}}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{L}(y(t) - \hat{y}(t)) \\ \hat{y}(t) = \mathbf{C}\hat{\mathbf{x}}(t) \end{cases}$$

→ we can then use feedback from the estimate

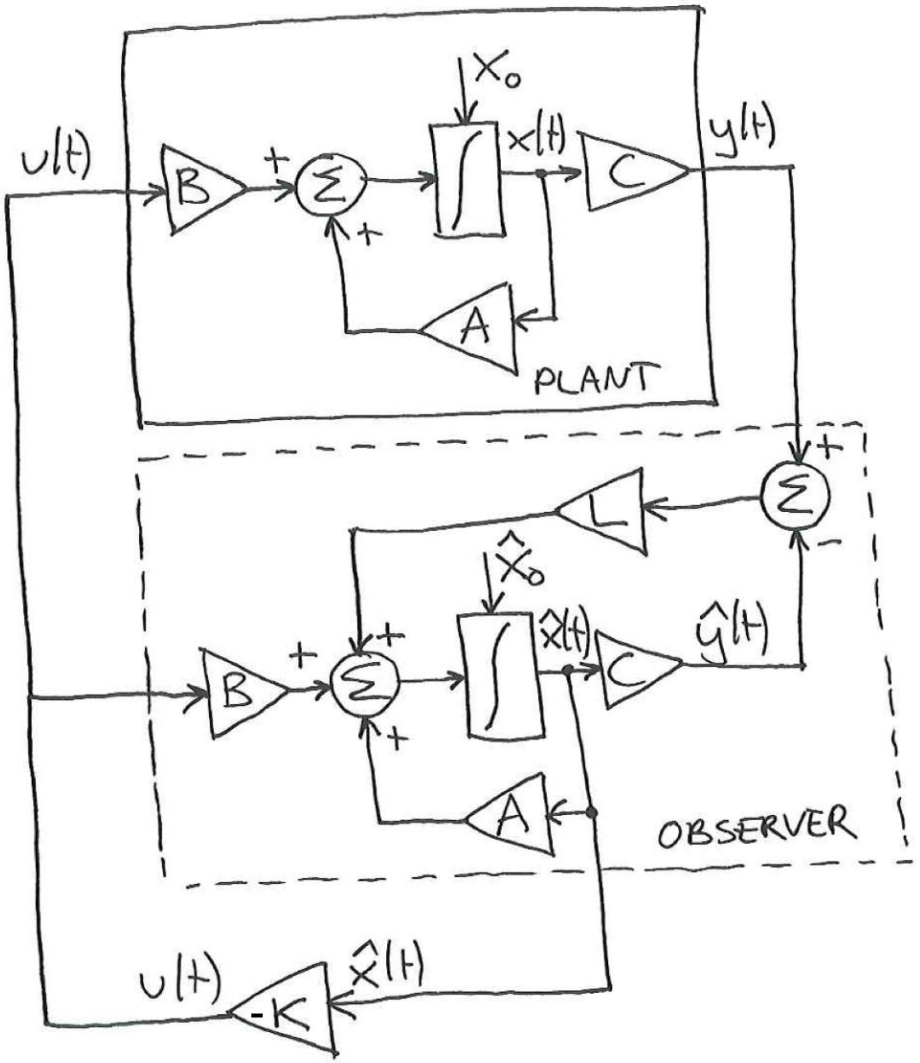
$$u(t) = -\mathbf{K}\hat{\mathbf{x}}(t)$$

Putting this control law and observer together gives

$$\begin{cases} \dot{\hat{\mathbf{x}}}(t) = (\mathbf{A} - \mathbf{LC} - \mathbf{BK})\,\hat{\mathbf{x}}(t) + \mathbf{L}y(t) \\ u(t) = -\mathbf{K}\hat{\mathbf{x}}(t) \end{cases}$$

→ Dynamic system! ($\neq$ static state-feedback)

Implementation



Remark: reference signal, disturbance rejection, etc. can be added

SDU

Application to tracking control

The previous observer-controller scheme can be completed with desired trajectories to do tracking:

$$\Delta u(t) = -\mathbf{K}\Delta\mathbf{x}(t)$$

with $\begin{cases} \Delta u(t) := u(t) - u_d(t) \\ \Delta\mathbf{x}(t) := \mathbf{x}(t) - \mathbf{x}_d(t) \end{cases}$

But instead of

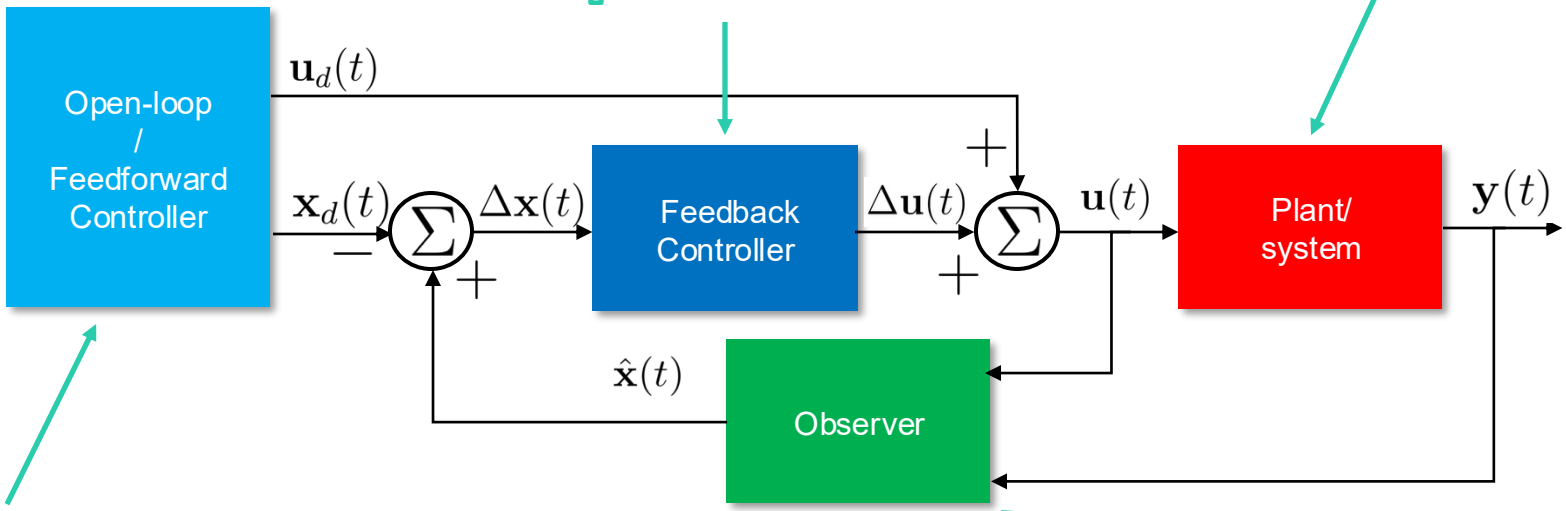
$$u(t) = -\mathbf{K}\left(\mathbf{x}(t) - \mathbf{x}_d(t)\right) + u_d(t)$$

we use

$$u(t) = -\mathbf{K}(\hat{\mathbf{x}}(t) - \mathbf{x}_d(t)) + u_d(t)$$

General controller structure

Putting controller/observer/desired trajectories gives the general controller structure...



eg. linear state-feedback, PID, MPC, feedback linearization, sliding mode, ADRC, etc.

eg. car, AUV, plane, robot arm, legged robot, mobile robot, drone, etc.

eg. polynomials, sinusoids, splines, Bezier curves, guidance controllers, etc.

eg. observer, Kalman Filter, EKF, UKF, Particle Filter, etc.

... which is present in many (if not all) advanced control systems

The separation principle (1/2)

Question: what is the influence of the tuning of the observer on the tuning of the controller?

Recall that for a tracking controller without observer, we have the tracking error dynamics $\frac{d}{dt}(\Delta\mathbf{x}) = \mathbf{A}\Delta x - \mathbf{B}\mathbf{K}\Delta\mathbf{x}$

because of the tracking controller $\Delta u = -\mathbf{K}\Delta\mathbf{x}$

"Plug" the estimate into the tracking controller

$$u(t) = -\mathbf{K}\left(\hat{\mathbf{x}}(t) - \mathbf{x}_d(t)\right) + u_d(t).$$

and use the estimation error

$$\tilde{\mathbf{x}}(t) := \hat{\mathbf{x}}(t) - \mathbf{x}(t)$$

to get

$$u(t) = -\mathbf{K}\left(\tilde{\mathbf{x}}(t) + \mathbf{x}(t) - \mathbf{x}_d(t)\right) + u_d(t)$$

The separation principle (2/2)

start again with $$\begin{aligned} u(t) &= -\mathbf{K}\left(\tilde{\mathbf{x}}(t) + \mathbf{x}(t) - \mathbf{x}_d(t)\right) + u_d(t) \\ &= -\mathbf{K}\left(\tilde{\mathbf{x}}(t) + \Delta\mathbf{x}(t)\right) + u_d(t) \end{aligned}$$

so that we have $$\Delta u(t) = -\mathbf{K}\Delta\mathbf{x}(t) - \mathbf{K}\tilde{\mathbf{x}}(t)$$

which gives the tracking error dynamics influenced by the estimation error

$$\frac{d}{dt}\left(\Delta\mathbf{x}(t)\right) = (\mathbf{A} - \mathbf{BK})\Delta\mathbf{x}(t) - \mathbf{BK}\tilde{\mathbf{x}}(t)$$

Combining the above with $\dot{\tilde{\mathbf{x}}}(t) = (\mathbf{A} - \mathbf{LC})\tilde{\mathbf{x}}$ gives

$$\frac{d}{dt}\begin{bmatrix}\tilde{\mathbf{x}} \\ \Delta\mathbf{x}\end{bmatrix} = \begin{bmatrix}\mathbf{A} - \mathbf{LC} & 0 \\ -\mathbf{BK} & \mathbf{A} - \mathbf{BK}\end{bmatrix}\begin{bmatrix}\tilde{\mathbf{x}} \\ \Delta\mathbf{x}\end{bmatrix}$$

cascade of 2 stable systems → As long as both estimation and tracking error dynamics are stable, "they can be tuned separately"

Active Disturbance Rejection Control

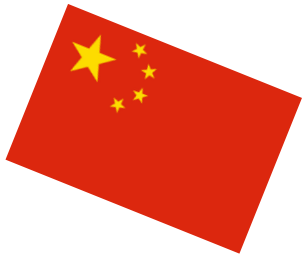
All the state-space techniques we have seen so far need a model...

→ model can be difficult to obtain, time-consuming

PID is a well-known "model-free" technique...

... is there a state-space counterpart?

→ Active Disturbance Rejection Control



- one of its embodiments originally emerged in China
- relies heavily on the observer concept
- mostly model-free, if a couple of things need to be known about the model
- outperforms the PID controller in quite a few applications

Estimating disturbances (1/3)

Consider the scalar system

$$\dot{x} = ax + bu + d \qquad (1)$$

with $y = x$ where everything is known except for the constant disturbance d

Question: can we estimate d?

rewrite (1) as $\begin{cases} \dot{x} = ax + bu + d \\ \dot{d} = 0 \end{cases}$

or, setting $x_d := d$, one gets $\begin{cases} \dot{x} = ax + x_d + bu \\ \dot{x}_d = 0 \end{cases}$

so that we have the state-space representation

$$(2) \quad \begin{cases} \dfrac{d}{dt}\begin{bmatrix} x \\ x_d \end{bmatrix} = \underbrace{\begin{bmatrix} a & 1 \\ 0 & 0 \end{bmatrix}}_{\mathbf{A}_E}\begin{bmatrix} x \\ x_d \end{bmatrix} + \underbrace{\begin{bmatrix} b \\ 0 \end{bmatrix}}_{\mathbf{B}_E} u \\ y(t) = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{\mathbf{C}_E}\begin{bmatrix} x \\ x_d \end{bmatrix} \end{cases} \quad \text{with } \mathbf{x}_E = [x, x_d]^T$$

Since system (2) is observable, it is easy to design an observer!

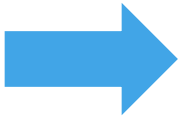
Estimating disturbances (2/3)

More generally:

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}(u + d), & \mathbf{x}(0) = \mathbf{x}_0 \\ y = \mathbf{C}\mathbf{x} \end{cases}$$

(with d again a constant disturbance)

Then, set $x_d := d$

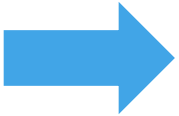


(1)

$$\begin{cases} \dot{\mathbf{x}}_E = \mathbf{A}_E\mathbf{x}_E + \mathbf{B}_E u \\ y = \mathbf{C}_E\mathbf{x}_E \end{cases}$$

with

$$\mathbf{A}_E = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ 0 & 0 \end{bmatrix} \quad \text{and} \quad \mathbf{B}_E = \begin{bmatrix} \mathbf{B} \\ 0 \end{bmatrix} \quad \text{and} \quad \mathbf{C}_E = \begin{bmatrix} \mathbf{C} & 0 \end{bmatrix}$$



again, easy to design an observer for (1)

Estimating disturbances (3/3)

What if the disturbance is a sine wave?

$$d(t) = A\sin(\omega t + \phi)$$

unknown: A, ϕ — known: ω

is a solution of ODE $\ddot{d} = -\omega^2 d$

set $\mathbf{x}_d = \begin{bmatrix} x_{d,1} \\ x_{d,2} \end{bmatrix} := \begin{bmatrix} d \\ \dot{d} \end{bmatrix}$

$$\begin{cases} \dot{\mathbf{x}}_d = \mathbf{A}_d \mathbf{x}_d \\ d = \mathbf{C}_d \mathbf{x}_d \end{cases}$$

with $\mathbf{A}_d = \begin{bmatrix} 0 & 1 \\ -\omega^2 & 0 \end{bmatrix}$ and $\mathbf{C}_d = \begin{bmatrix} 1 & 0 \end{bmatrix}$

This gives the extended state-representation

$$\begin{cases} \dot{\mathbf{x}}_E = \mathbf{A}_E \mathbf{x}_E + \mathbf{B}_E u \\ y = \mathbf{C}_E \mathbf{x}_E \end{cases}$$

with $\mathbf{A}_E = \begin{bmatrix} \mathbf{A} & \mathbf{B}\mathbf{C}_d \\ 0 & \mathbf{A}_d \end{bmatrix}$

and $\mathbf{B}_E = \begin{bmatrix} \mathbf{B} \\ 0 \end{bmatrix}$ and $\mathbf{C}_E = \begin{bmatrix} \mathbf{C} & 0 \end{bmatrix}$

Extended-State Observers (ESO) (1/2)

Central concept in ADRC: the use of ESO for nonlinear systems

First-order case: $\dot{y} = f(y, w, t) + bu$

unknown — known

with $y = x$

Main idea: On a small interval, function f can be seen as a constant!

Hence we also have $\dot{f} \approx 0$ (on a small time interval)

→ we can design the extended-state observer

$$\begin{cases} \dot{\hat{y}} = \hat{f} + bu + l_1(y - \hat{y}) \\ \dot{\hat{f}} = l_2(y - \hat{y}) \end{cases}$$

(assuming that l_1, l_2 are tuned so that the observer is fast enough so that it follows f)

vectorial form:

$$\begin{cases} \dot{\hat{\mathbf{x}}}_E = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \hat{\mathbf{x}}_E + \begin{bmatrix} b \\ 0 \end{bmatrix} u + \begin{bmatrix} l_1 \\ l_2 \end{bmatrix} (y - \hat{y}) \\ \hat{y} = \begin{bmatrix} 1 & 0 \end{bmatrix} \hat{\mathbf{x}}_E \end{cases} \quad \text{with} \quad \hat{\mathbf{x}}_E = [\hat{y}, \hat{f}]^T$$

Extended-State Observers (ESO) (2/2)

Second-order case:

$$\ddot{y} = f(y, \dot{y}, w, t) + bu$$

unknown — known

OR (state-space representation)

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = f(x_1, x_2, w, t) + bu \\ y = x_1 \end{cases}$$

→ ESO:

$$\begin{cases} \dot{\hat{\mathbf{x}}}_E = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \hat{\mathbf{x}}_E + \begin{bmatrix} 0 \\ b \\ 0 \end{bmatrix} u + \begin{bmatrix} l_1 \\ l_2 \\ l_3 \end{bmatrix} (y - \hat{y}) \\ \hat{y} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \hat{\mathbf{x}}_E \end{cases}$$

Closing the loop: ADRC (1/2)

First-order case:

$$\dot{y} = f(y, w, t) + bu$$

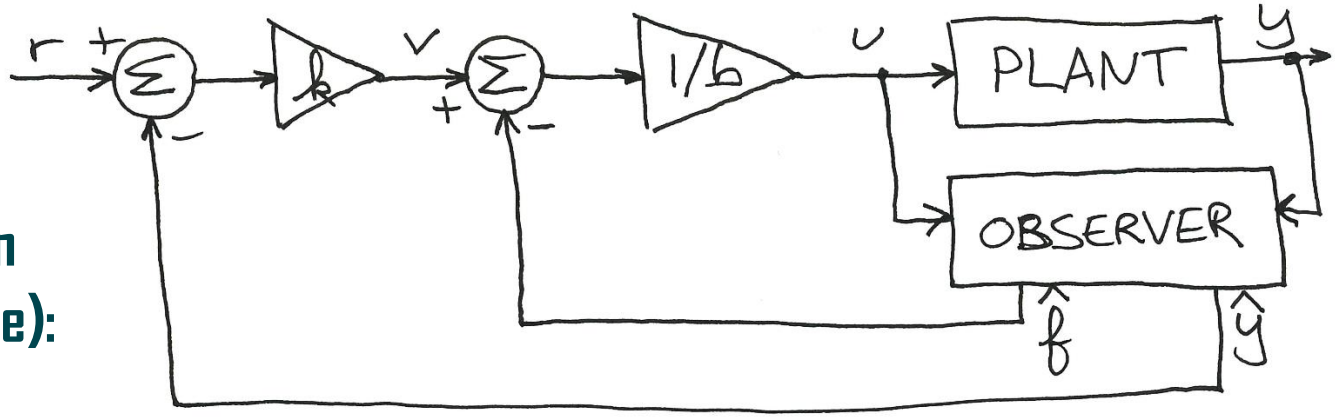
→ disturbance-rejection controller

$$u = \frac{-\hat{f} + v}{b}$$

ESO implies $\hat{f} \approx f$ → $\dot{y} = v$

→ stabilizing controller $v = -k\hat{y}$ stabilizes the integrator

implementation (with reference):



Closing the loop: ADRC (2/2)

Second-order case:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = f(x_1, x_2, w, t) + bu \\ y = x_1 \end{cases}$$

→ disturbance-rejection controller

$$u = \frac{-\hat{f} + v}{b}.$$

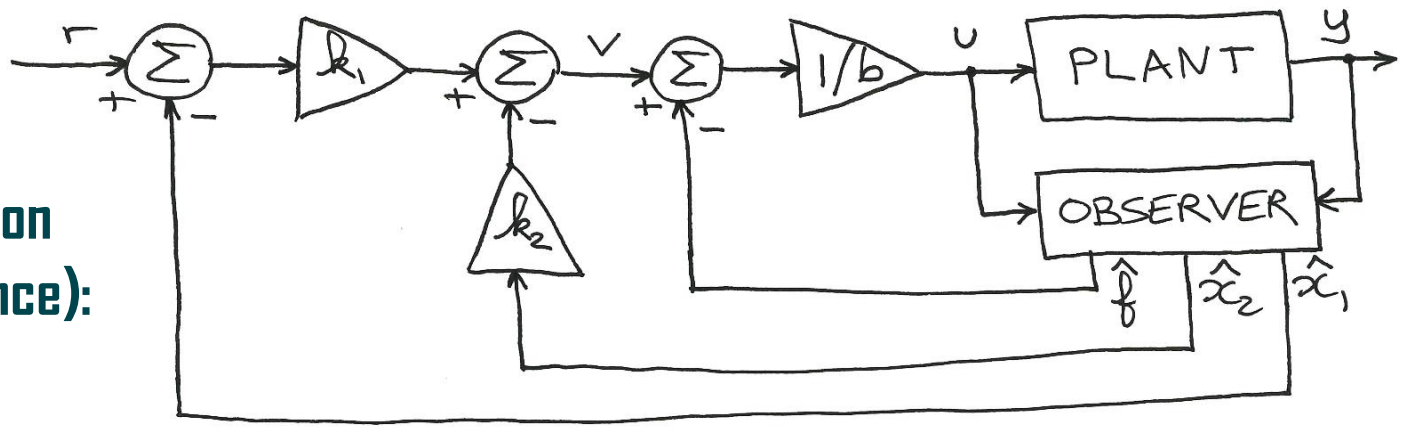
gives the double integrator

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = v \end{cases}$$

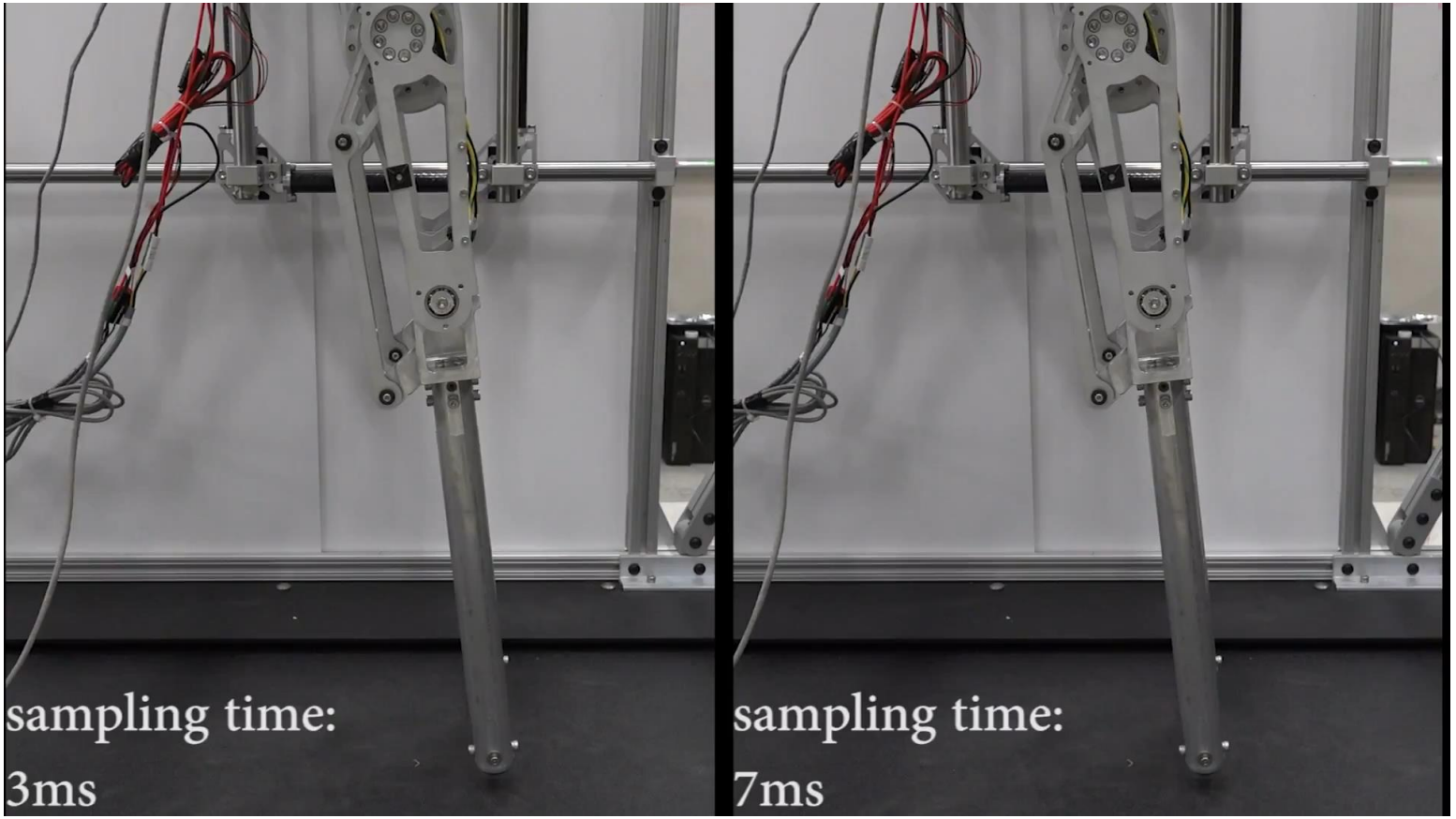
→ stabilized by

$$v = -k_1\hat{x}_1 - k_2\hat{x}_2$$

implementation (with reference):



Video: effect of discretization on robot leg



Video: ADRC vs PID controller