DEPARTMENT OF ELECTROMECHANICAL, SYSTEMS AND METAL ENGINEERING

CONTROL THEORY

Kevin Dekemele, Jasper Juchem, Michel De Roeck

UNIVERSITEIT GENT

ArcelorMittal

CONTENT (9-12H, 13-14H)

- Control – an overview
- State space
  - Model & linearization
  - Stabilization
  - Feedback & Linear Quadratic Control
  - Observation & Kalman Filter

System dynamical properties:
Stabilize around equilibrium &
observe with disturbance,
noise and limited information

CONTENT (14-17H)

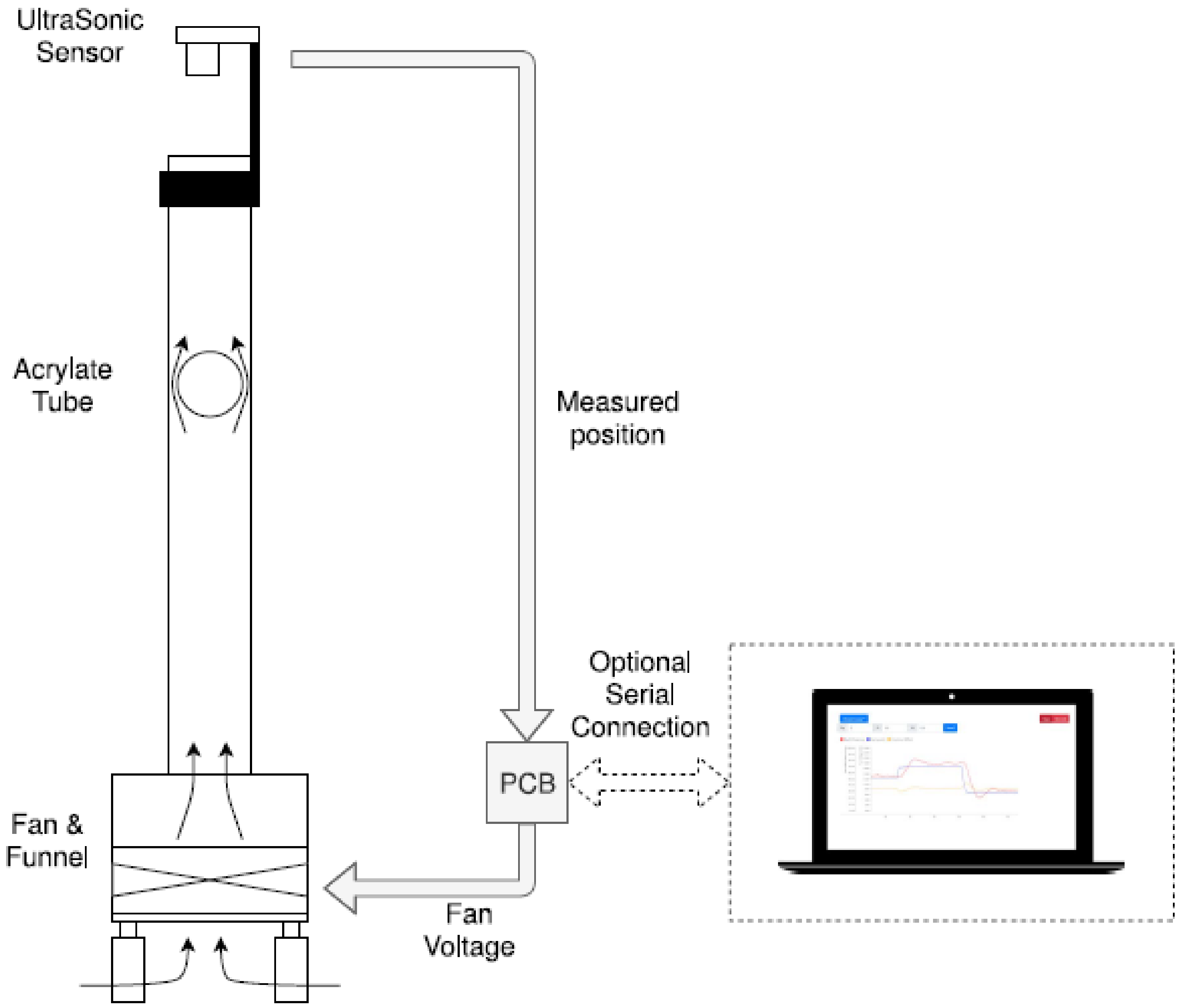
- Control Architectures
  - Input-output models
  - Control Algorithms
    - LQR
    - LQG
  - MIMO control: decoupling
  - Model Predictive control

Assuming stability & observation, how do we get where we want, given constraints, costs, …. ?

UNIVERSITEIT GENT

CONTROL: AN OVERVIEW

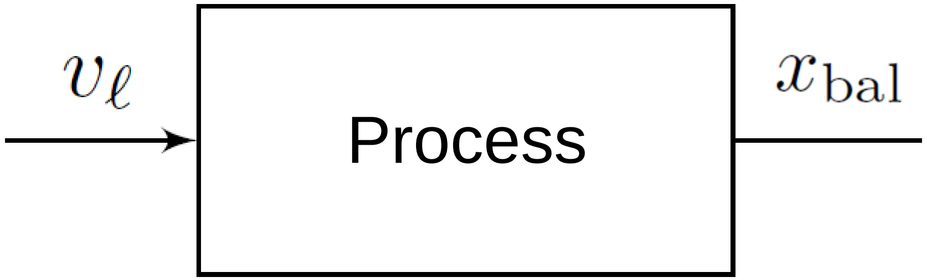
SYSTEM: LEVITATION OF A BALL



UNIVERSITEIT GENT

IDEAL PROCESS

= Input fully determines output



Model $$m\ddot{x} = -mg + c(v_\ell - \dot{x})^2$$

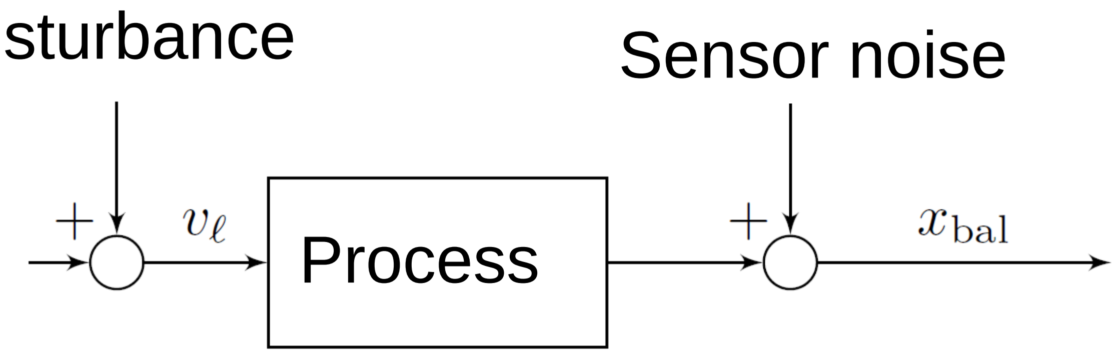
Equilibrium $$mg = cv_\ell^2$$

Hard to stabilize

Position difficult to choose

UNIVERSITEIT GENT

REAL PROCESS

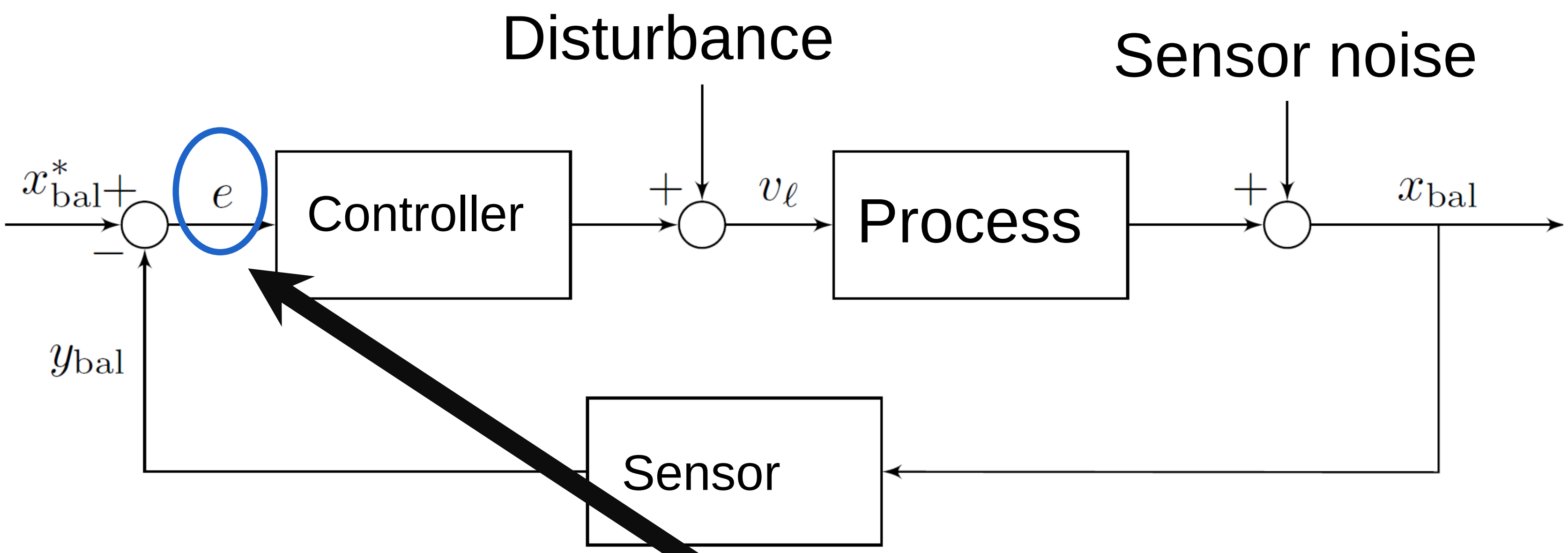


Disturbance: ‘Input’ we have no control over,

Changes in assumed model

Sensor noise: Sensor never exact

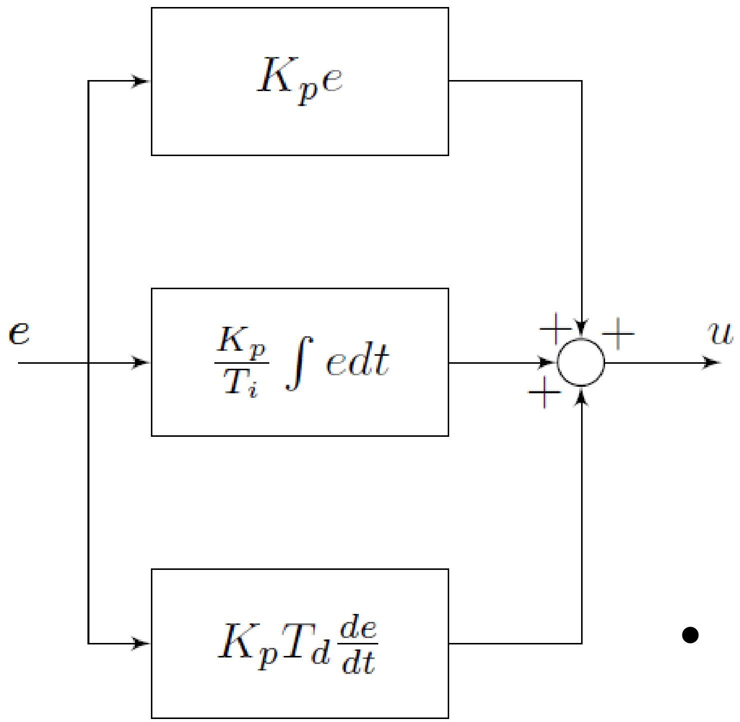
FEEDBACK AND CONTROL ALGORITMH



- Sensor dynamics: Usually much faster than process (e.g. thermocouple)
- Control algorithm: Algorithm to modify input based on measured value and option setpoint
- Optimize speed, energy cost, stabilize, tracking (error=0)

UNIVERSITEIT GENT

CONTROL ALGORITHM



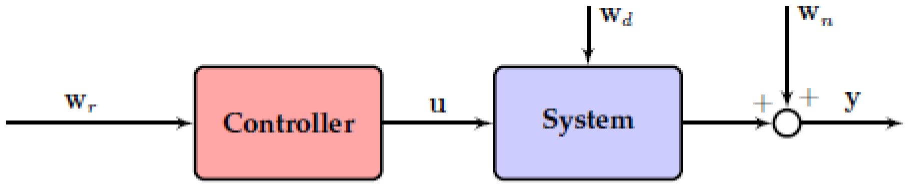
$$u(t) = K_p\left(e(t) + \frac{1}{T_i}\int_0^t e(\tau)d\tau +) + T_d\frac{de(t)}{dt}\right)$$

$$u(t) = K_p(e(t) + \frac{1}{T_i}\sum_{i=0}^{n} e(i)t_s + T_d\frac{e(t)-e(t-t_s)}{ts}$$

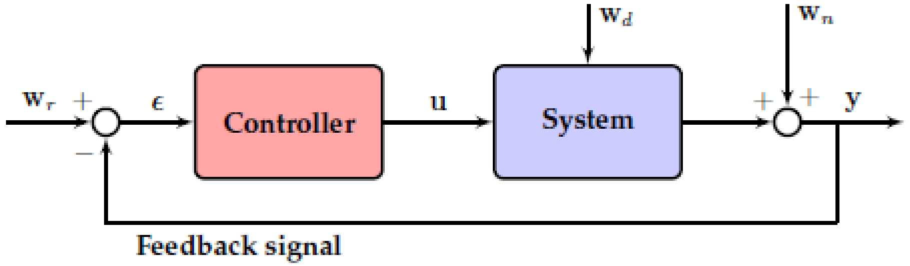
- P: $K_p e(t)$ control as long as there is an error
- I: $\frac{K_p}{T_i}\sum_{i=0}^{n} e(i)t_s$ control as long as a sum of error
- D: $K_p T_d \frac{e(t)-e(t-t_s)}{ts}$ control as long as there is a derivative of the error

OPEN VS CLOSD

- Based on history or existing model, provide input



- Based on measured output, provide input

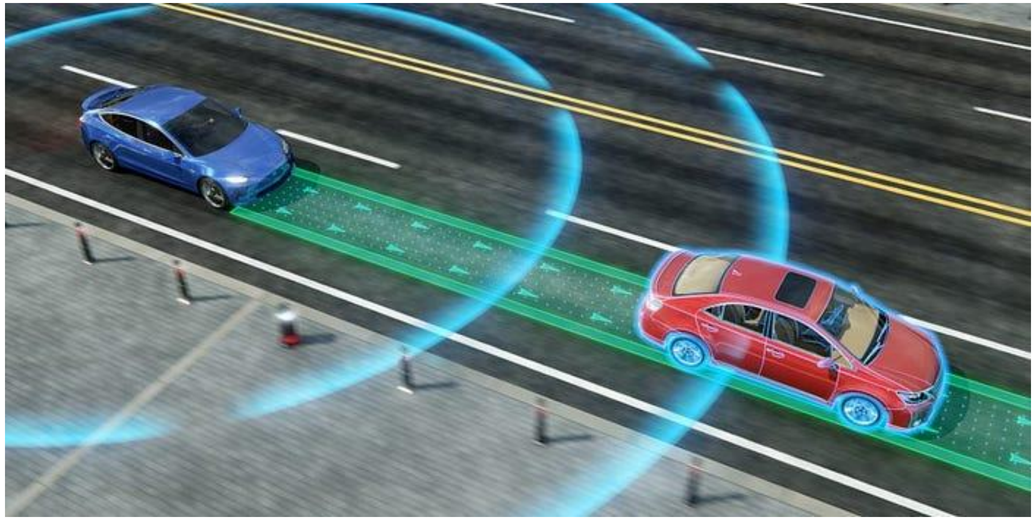


- Fuel rate model $y = u$ cruise control
- Crane positioning

UNIVERSITEIT GENT

SELF DRIVING

- Vehicle dynamics
- Tracking road
- Disturbance road



- Sensors & GPS
- Decide fuel and steer angels based on this information

AEROSPACE

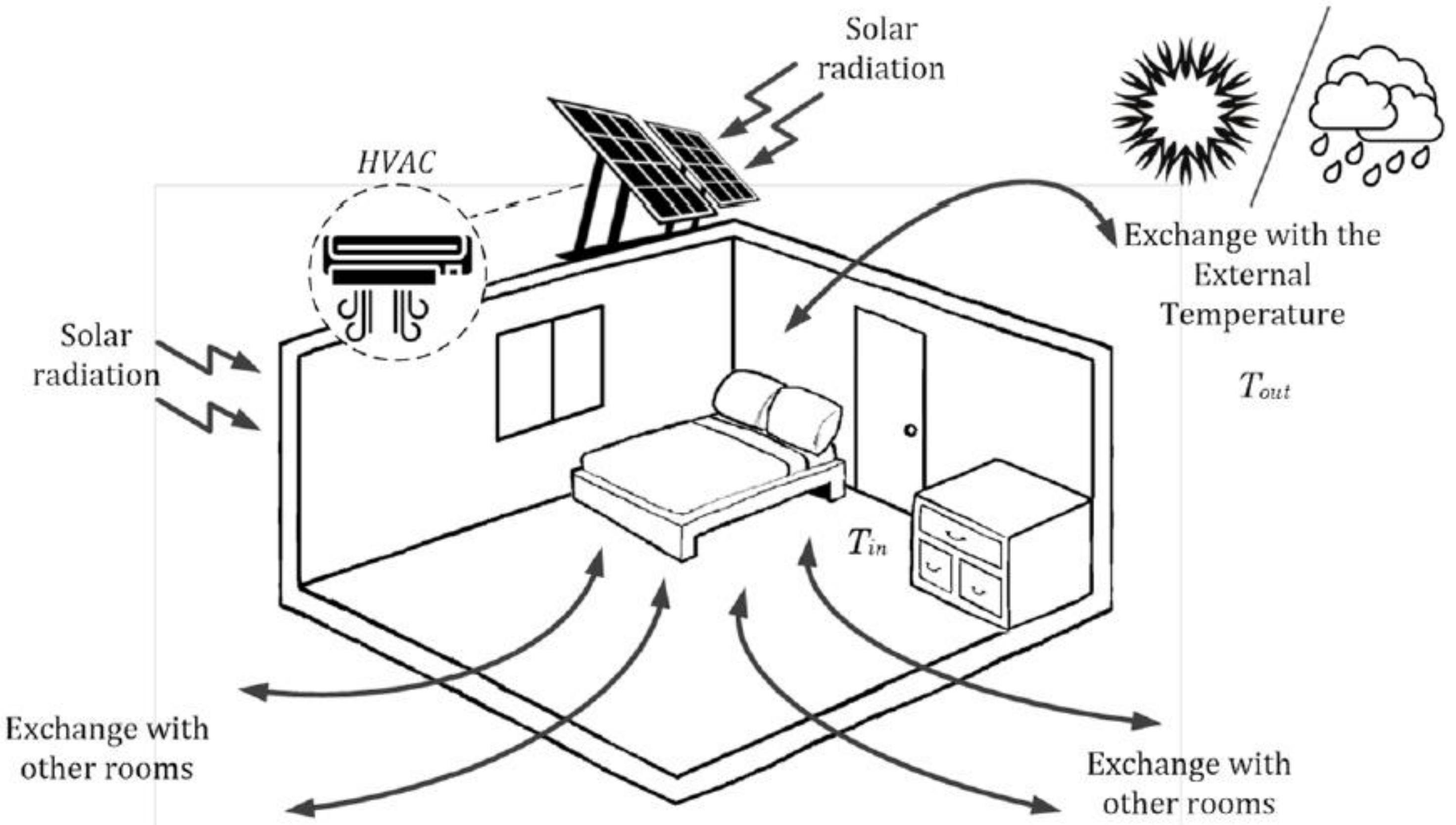
– Rocket dynamics
– Stabilize and desired location
– Side and main thrusters

https://users.ugent.be/~kdkemele/rocket2018/

UNIVERSITEIT GENT

HEATING

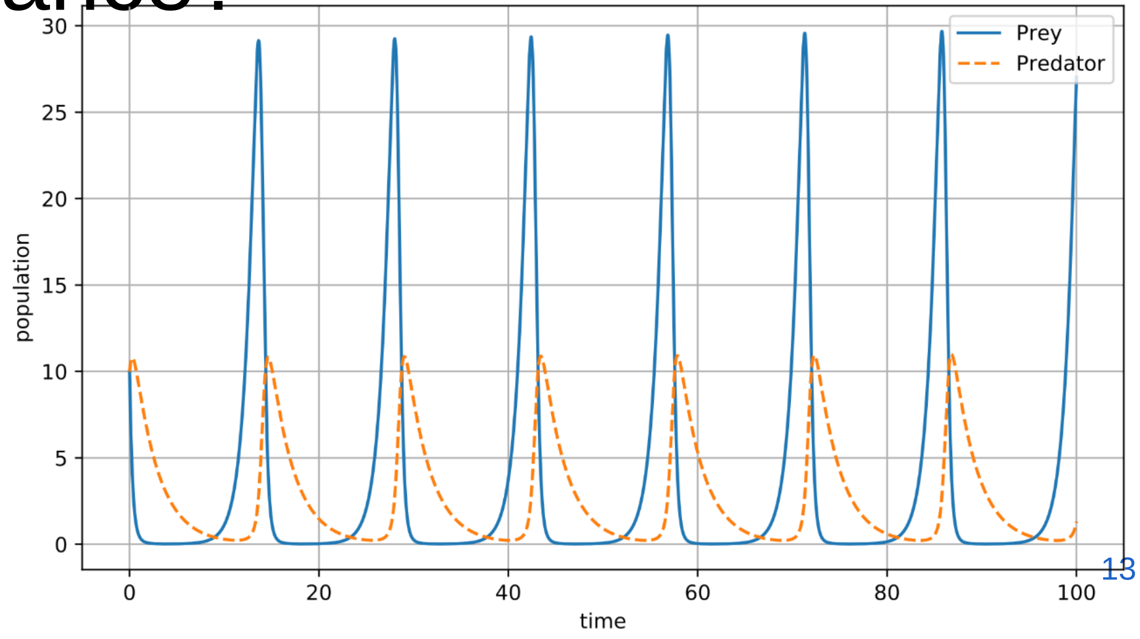
- Heating room
- Disturbance (window, sun)
- Lowering energy use
- Energy balance



UNIVERSITEIT
GENT

PREDATOR PREY/ INFECTION

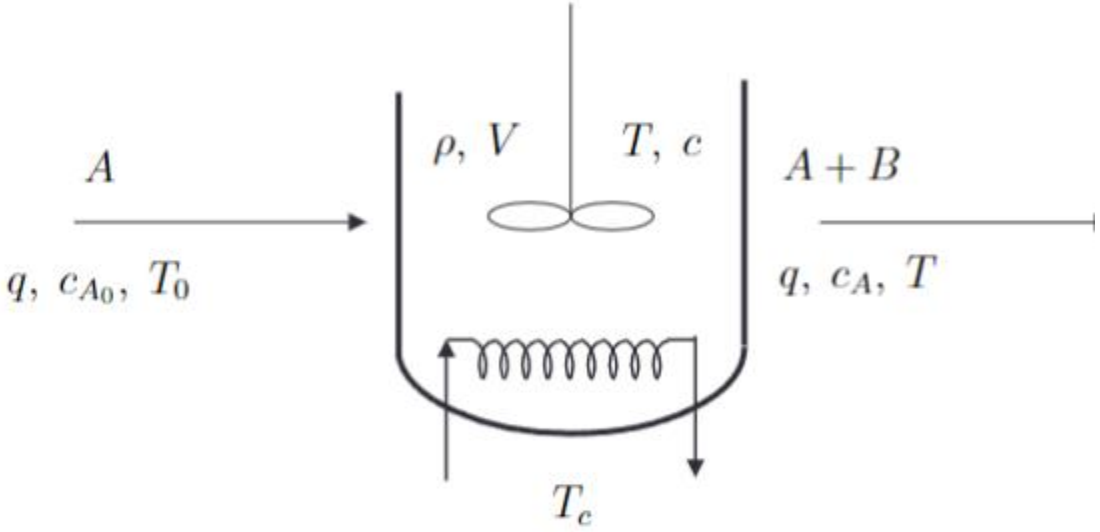
- Many prey -> many predators
- Too much compitino and less food -> less predators
- Less predators -> more prey
- Compition model
- What happens with disturbance?

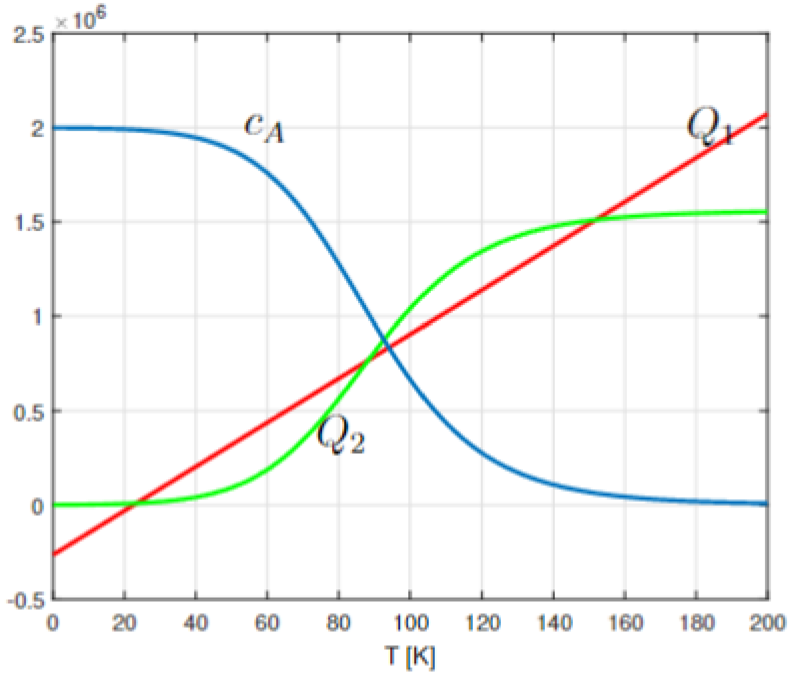


UNIVERSITEIT GENT

(BIO) CHEMICAL REACTOR

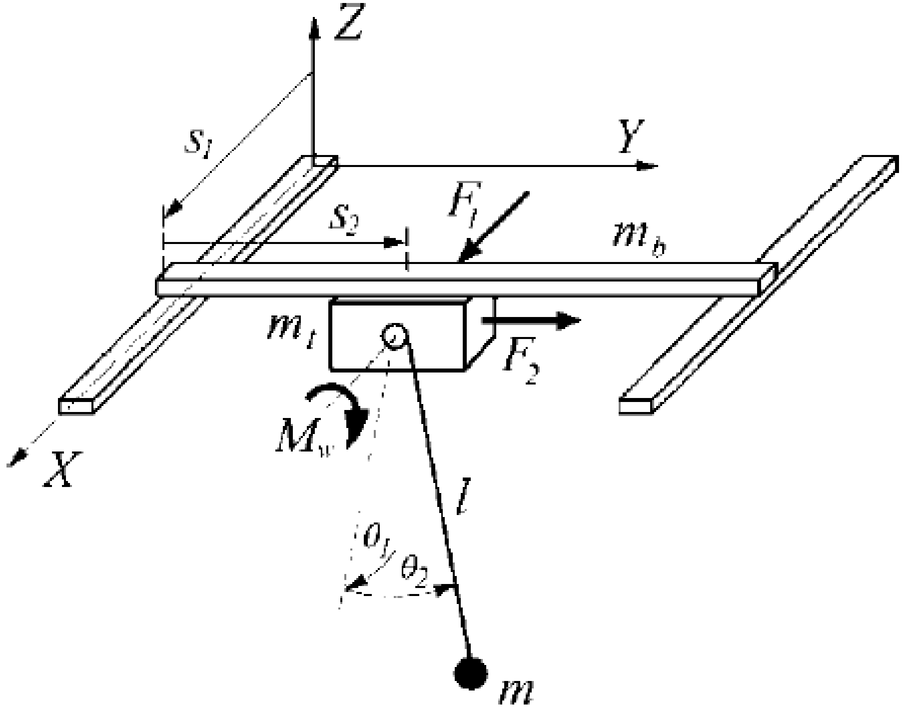
- Balance Removed heat = generated heat
- ‘Instable’ points might provide best generation of substance B
- Feedback

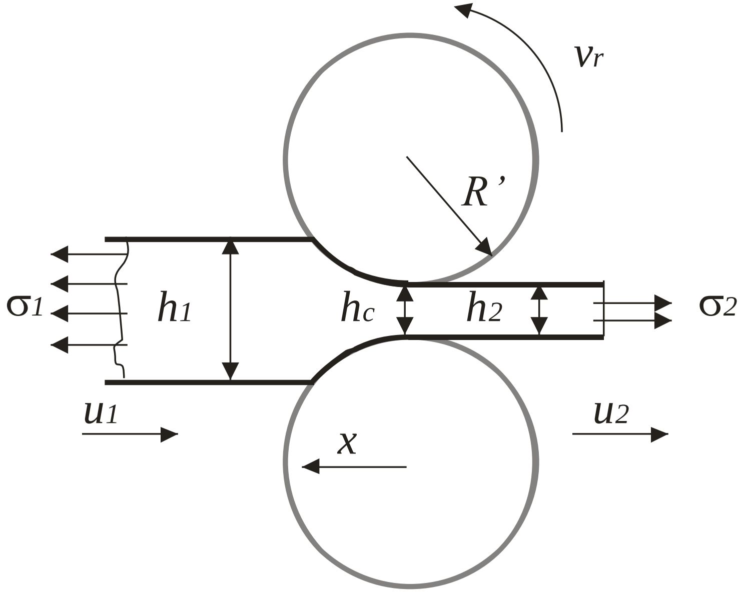




UNIVERSITEIT GENT

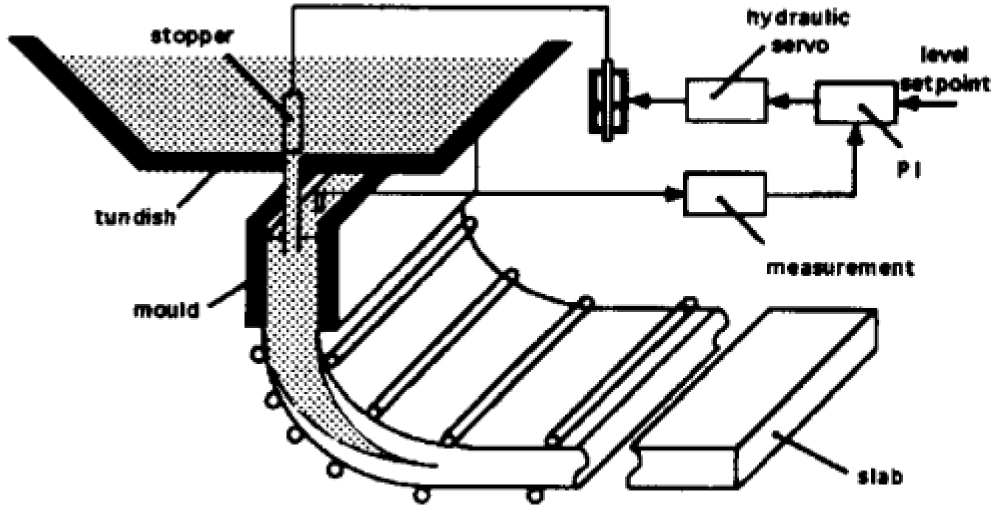
STEEL FACTORY

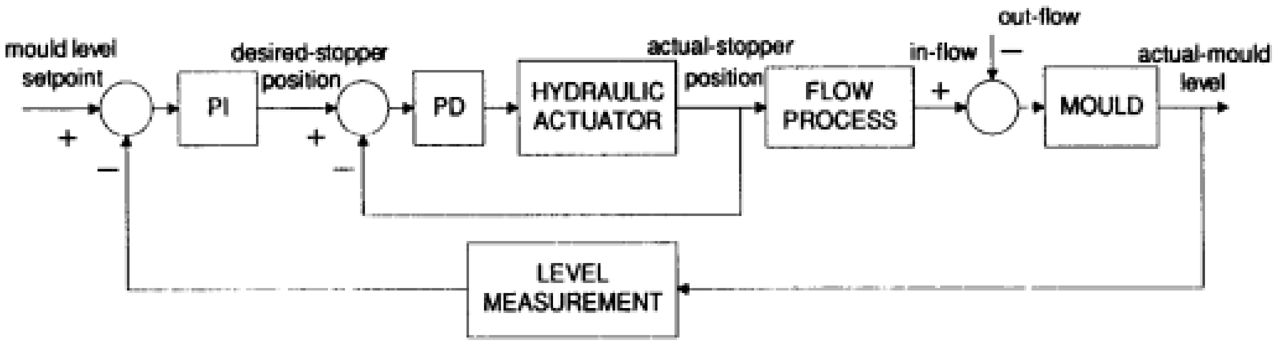




CONTINUOUS CASTER

- Level of steel control
- Determines quality
- Two cascades systems
  1. Stopper (fast)
  2. Level of caster (slow)





STATE SPACE

MATRICES AND VECTORS

Matrix

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1m} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nm} \end{bmatrix} \in \mathbb{R}^{n \times m}$$

Vectors

$$x = \begin{bmatrix} x_1 & \dots & x_m \end{bmatrix} \in \mathbb{R}^{1 \times m}$$

$$x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \in \mathbb{R}^{n \times 1}$$

Determinant

$$\det(A) = \begin{vmatrix} a_{11} & \cdots & a_{1m} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nm} \end{vmatrix} \in \mathbb{R}$$

$$\det(A) = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11} \cdot a_{22} - a_{12} \cdot a_{21}$$

UNIVERSITEIT GENT

EIGENVECTORS AND VALUES

– Matrix maps column vector into another one with transformation:

$$y = Ax,\ A \in \ \mathbb{R}^{n\times n}$$

Rotation matrix: $A = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$

– Eigenvector v maps to itself with a factor λ (= eigenvalue)

$$v\lambda = Av$$

EIGENVECTORS AND VALUES: STEPS

$$v\lambda = Av \Rightarrow 0 = (A - I\lambda)v$$

Determine n eigenvalues λ: $\det(A - I\lambda) = 0$

Then compute n eigenvectors

$$(A - I\lambda_i)v_i \quad = 0$$

EIGENVECTOR AND VALUES: DIAGONALIZATION

$V = [v_1\ v_2 \ldots v_n]$ and $\Lambda = \mathrm{d}iag(\lambda_1, \lambda_2, \ldots, \lambda_n)$

$$V\Lambda = AV$$

$$V^{-1}V\Lambda = V^{-1}AV$$

$$\Lambda = V^{-1}AV$$

BASIC MATHEMATICS: TAYLOR SERIES

- $f(x) \approx f(a) + f'(a)(x-a) + \frac{f''(a)(x-a)^2}{2} + \cdots$
- Change of coordinates $(x-a) = \Delta x$

- $f(x,y,z) \approx f(a,b,c) + \frac{\partial f(a,b,c)}{\partial x}\left(\underbrace{x-a}_{\Delta x}\right) + \frac{\partial f(a,b,c)}{\partial y}\left(\underbrace{y-b}_{\Delta y}\right) + \frac{\partial f(a,b,c)}{\partial z}\left(\underbrace{z-c}_{\Delta z}\right)$

UNIVERSITEIT GENT

MODELS

Let of Linear Time Invariant first order ODEs (LTI systems)

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

$$y = Cx + Du$$

- State vector $x \in \mathbb{R}^{n \times 1}$
- Input vector $u \in \mathbb{R}^{p \times 1}$ (How to intervene in dynamics)
  - Disturbances (inputs, but not from us)
- Output vector $y \in \mathbb{R}^{q \times 1}$ (What we can measure)
- $A \in \mathbb{R}^{n \times n}$: System Matrix
- $B \in \mathbb{R}^{n \times p}$: Input Matrix
- $C \in \mathbb{R}^{q \times n}$: Output Matrix

MODELS: DISCRETE TIME

Set of linear Difference equations:

$$x_{k+1} = A_d x_k + B_d u_k$$

$$y_k = C_d x_k + D_d\, u_k$$

- Used in Model Predictive control
- Conversion Continuous to discrete time:

$$\mathrm{A}_d = e^{\mathbf{A}\Delta t},$$

$$\mathrm{B}_d = \int_0^{\Delta t} e^{\mathbf{A}\tau}\mathrm{B}\,\mathrm{d}\tau,$$

$$\mathrm{C}_d = \mathrm{C},$$

$$\mathrm{D}_d = \mathrm{D}.$$

UNIVERSITEIT GENT

LINEARIZATION

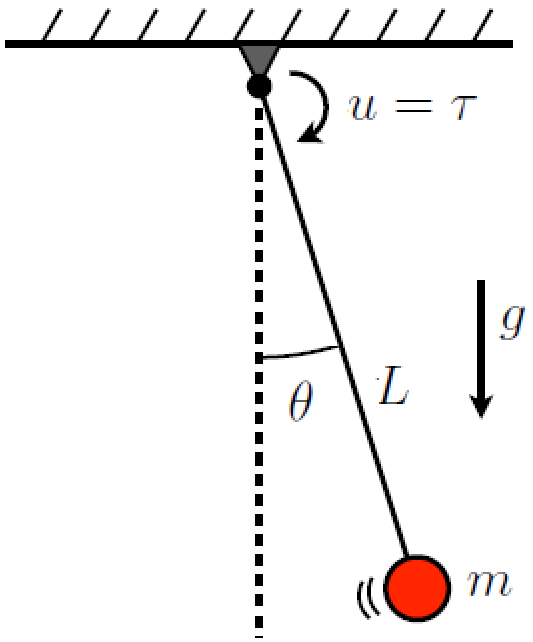
$$\dot{x} = f(x, u)$$
$$y = g(x, u)$$

Taylor series around operational or equilibrium point $(\dot{x} = 0)$, $(\bar{x}, \bar{u})$

$$f(\Delta x, \Delta u) \approx f(\bar{x}, \bar{u}) + \underbrace{\frac{\partial f(\bar{x}, \bar{u})}{\partial x}}_{A} \Delta x + \underbrace{\frac{\partial f(\bar{x}, \bar{u})}{\partial u}}_{B} \Delta u$$

UNIVERSITEIT GENT

PENDULUM



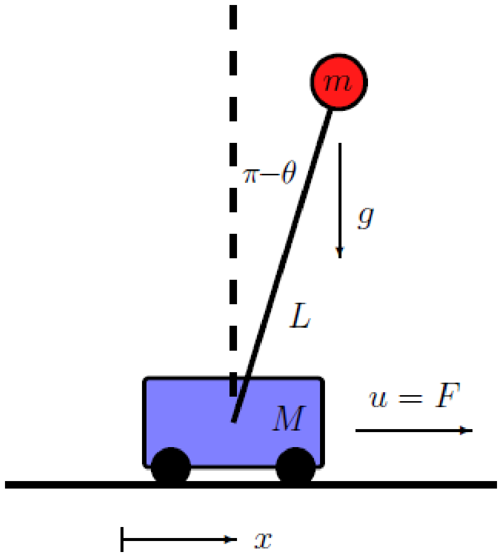
$$\ddot{\theta} = -\frac{g}{L}\sin\theta + u$$

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} \implies \frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ -(g/L)\sin(x_1) + u \end{bmatrix}.$$

$$\underbrace{\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ g/L & 0 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u}_{\text{pendulum up, } \lambda = \pm\sqrt{g/L}}$$

$$\underbrace{\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -g/L & 0 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u}_{\text{pendulum down, } \lambda = \pm i\sqrt{g/L}}$$

PENDULUM ON CART = CRANE



$$\dot{x} = v,$$

$$\dot{v} = \frac{-m^2L^2 g\cos(\theta)\sin(\theta) + mL^2(mL\omega^2\sin(\theta) - \delta v) + mL^2 u}{mL^2(M + m(1-\cos(\theta)^2))},$$

$$\dot{\theta} = \omega,$$

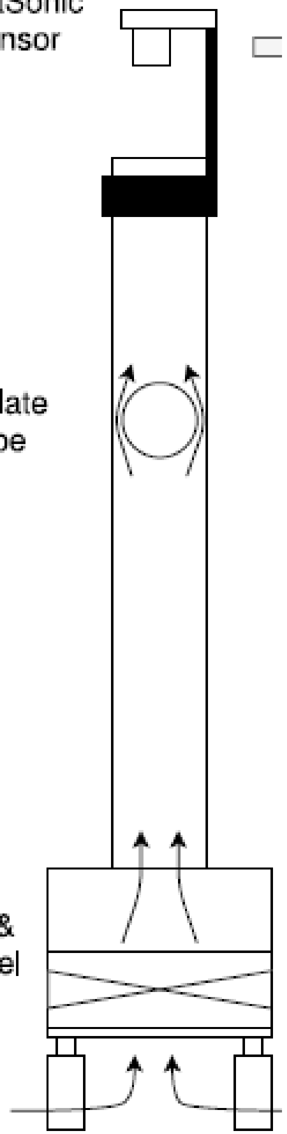
$$\dot{\omega} = \frac{(m+M)mgL\sin(\theta) - mL\cos(\theta)(mL\omega^2\sin(\theta) - \delta v) - mL\cos(\theta)u}{mL^2(M + m(1-\cos(\theta)^2))}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\delta/M & bmg/M & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -b\delta/ML & -b(m+M)g/ML & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 1/M \\ 0 \\ b/ML \end{bmatrix} u$$

b = 1 (up)

b = -1 (down)

LEVITATION OF BALL



$$m\ddot{x} = -mg + c(v_\ell - \dot{x})^2$$

$$mg = cv_\ell^2$$

$$\begin{bmatrix} \dot{x}_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -2\sqrt{\frac{mg}{c}} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 2\sqrt{\frac{mg}{c}} \end{bmatrix} u$$

$$y = [1\ 0] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + [0]u$$

We only measure position

COMPETITION MODEL

$x_1(t)$: Rabbit population
$x_2(t)$: Sheep population

$$\dot{x}_1 = x_1(3 - x_1 - 2x_2)$$
$$\dot{x}_2 = x_2(2 - x_1 - x_2)$$

$$A = \begin{bmatrix} 3 - 2\bar{x}_1 - 2\bar{x}_2 & -2\bar{x}_1 \\ -\bar{x}_2 & 2 - \bar{x}_1 - 2\bar{x}_2 \end{bmatrix}$$

$$\begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 2 \end{bmatrix}, \begin{bmatrix} 3 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

UNIVERSITEIT GENT

EIGENVALUES & EIGENVECTORS

- Autonomous system: $\dot{x} = Ax$
- Solution $x(t) = e^{At}x(0)$ $\qquad e^{At}$??? $\quad A \in \mathbb{R}^{n\times n}$
- Eigenvalues/vectots of A , Λ and V
- Eigenmodes $Vz(t) = x$
- $V\dot{z} = AVz \Rightarrow V^{-1}V\dot{z} = V^{-1}AVz$
- $\dot{z} = \Lambda\, z \Rightarrow z(t) = e^{\Lambda t}z(0)$

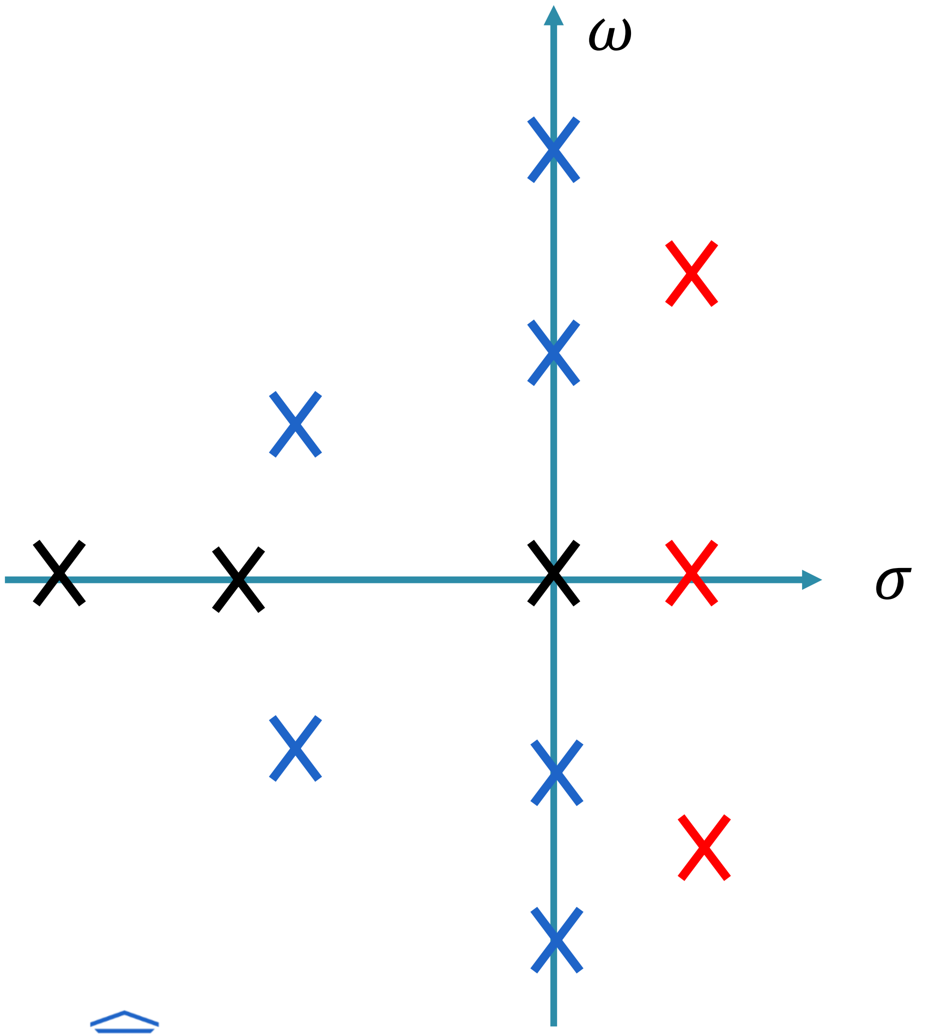
$$e^{\Lambda t} = \begin{bmatrix} e^{\lambda_1 t} & 0 & \cdots & 0 \\ 0 & e^{\lambda_2 t} & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & e^{\lambda_n t} \end{bmatrix} \in \mathbb{R}^{n\times n}$$

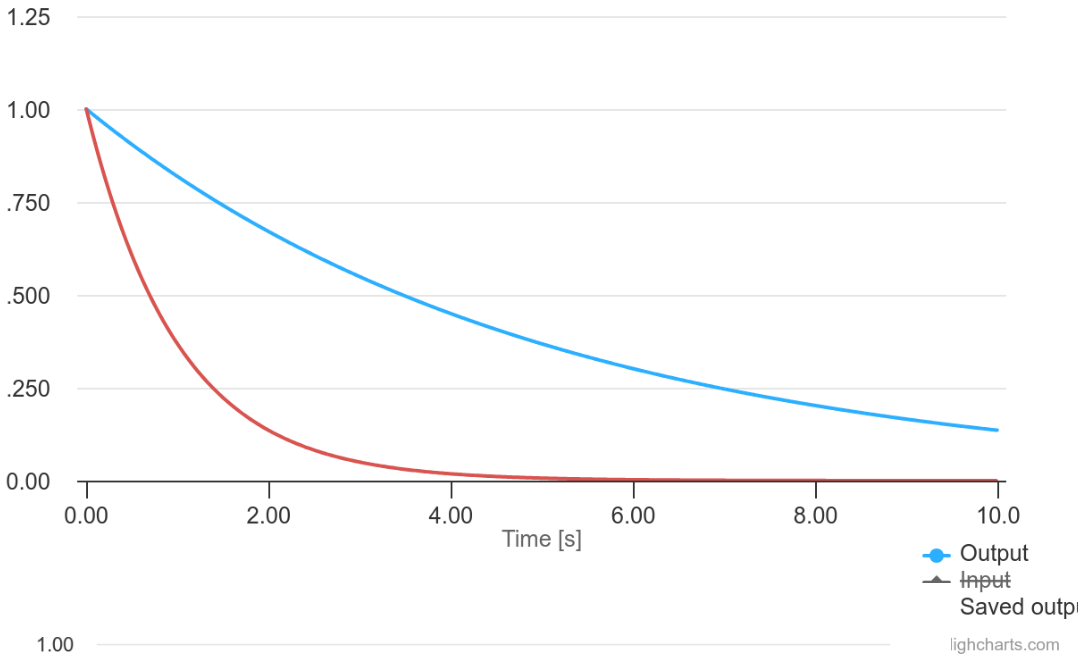
EIGENVALUES & EIGENVECTORS

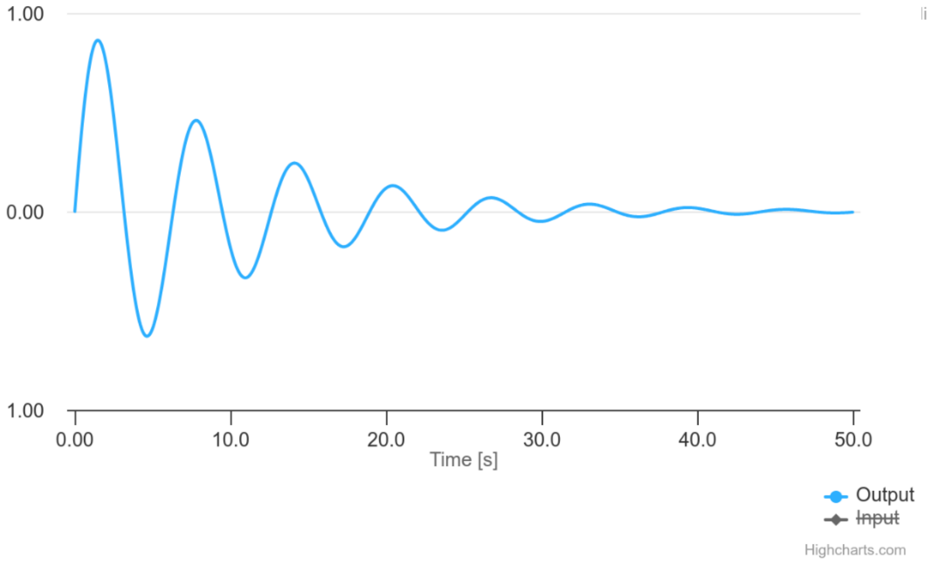
- $\dot{x} = Ax \Rightarrow \dot{x} = V\Lambda V^{-1}x$
- Solution $x(t) = e^{V\Lambda V^{-1}t}x(0) \Rightarrow Ve^{\Lambda t}\, V^{-1}x(0)$
- $x_k(t) = \sum_{i=1}^{n} v_i(k)\, e^{\lambda_i t} v_i^{-1}(k) x_i(0)$
- $\lambda_i = \sigma_i + j\omega_i$

What types of time responses are there??

EIGENVALUES & STABILITY





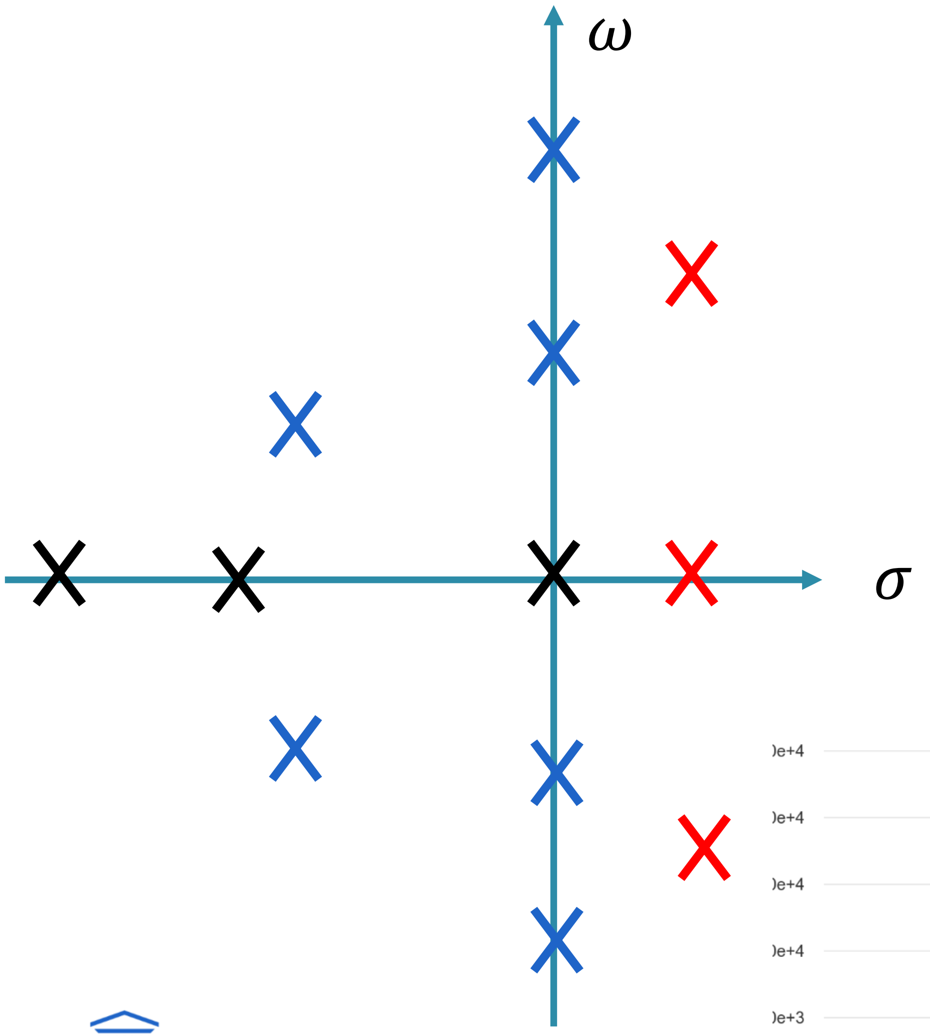


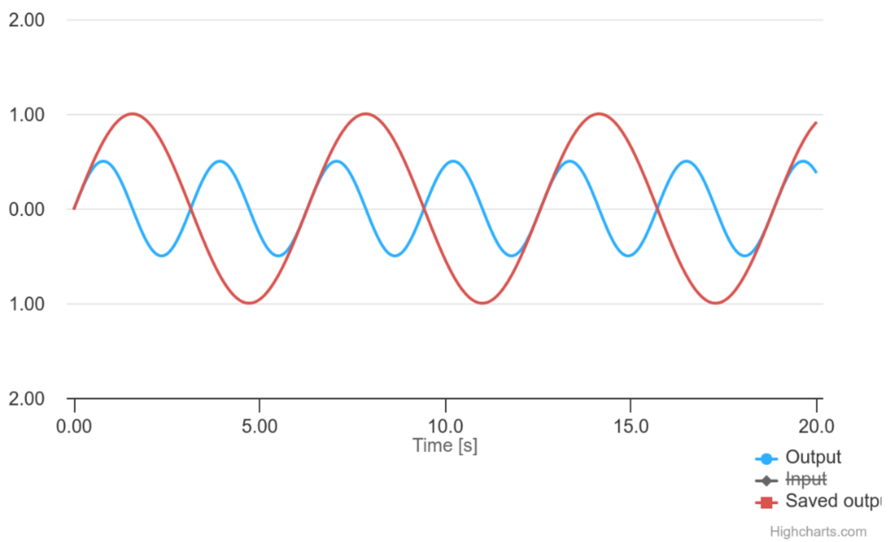
Oscillations
Process
Unstable

UNIVERSITEIT GENT

Figures:
https://odysc.ugent.be

EIGENVALUES & STABILITY

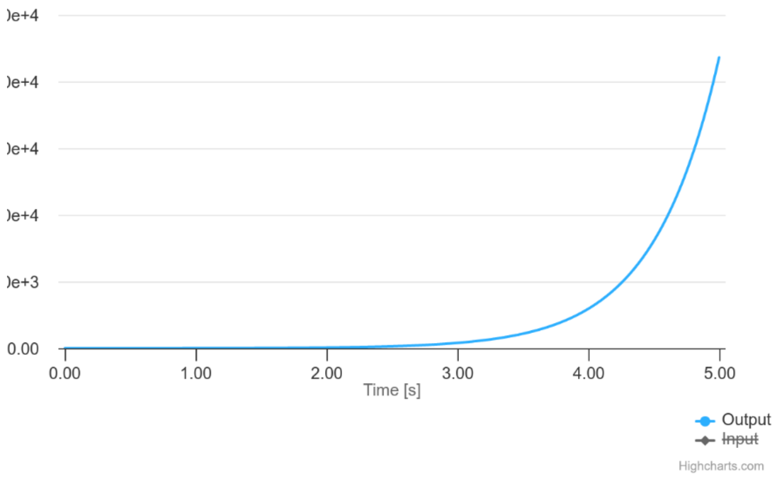


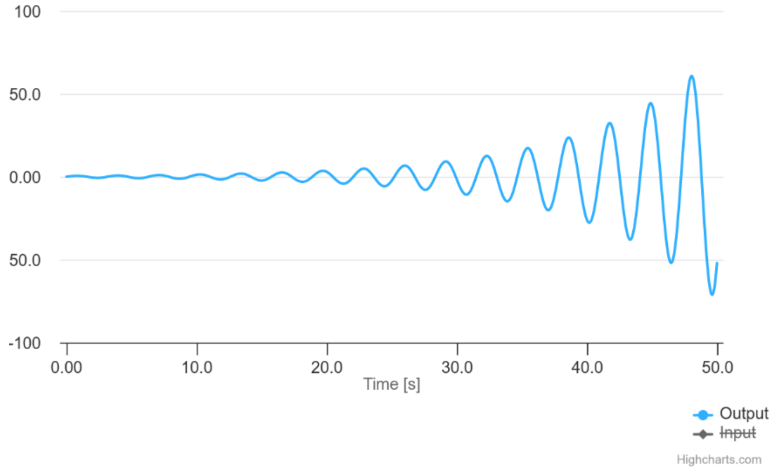


Oscillations

Process

Unstable



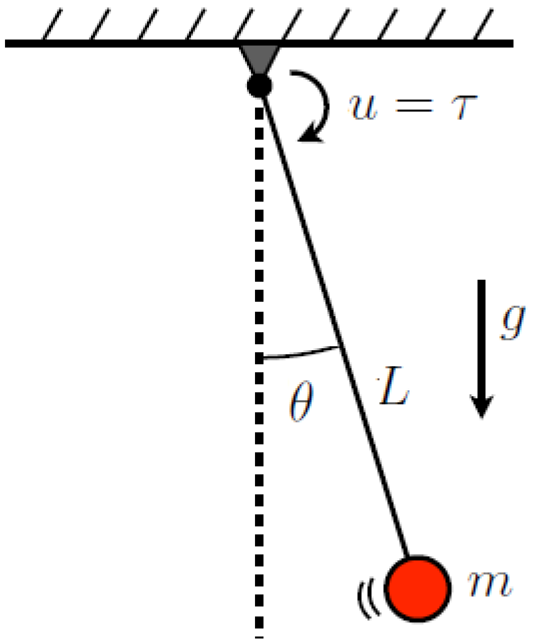


UNIVERSITEIT GENT

EIGENVALUES, EIGENVECTORS & STABILITY

- Eigenvalues in dynamical system meaning:
  - Stability & Speed
  - Response is exponential of eigenvalues, weighted by eigenvectors
  - (if x(0) is an eigenvector, you stay on this vector)

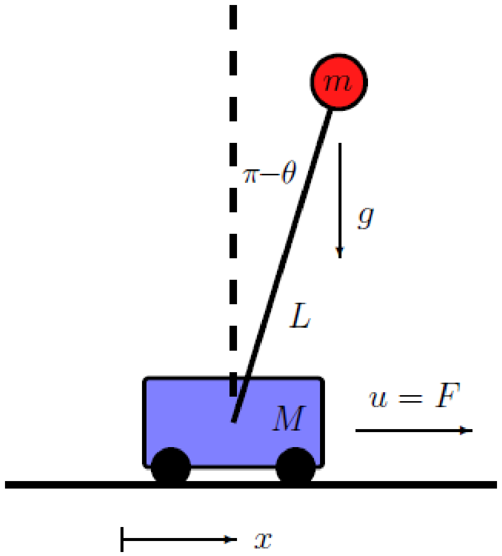
PENDULUM



$$\underbrace{\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ g/L & 0 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u}_{\text{pendulum up, } \lambda = \pm\sqrt{g/L}}$$

$$\underbrace{\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -g/L & 0 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u}_{\text{pendulum down, } \lambda = \pm i\sqrt{g/L}}$$

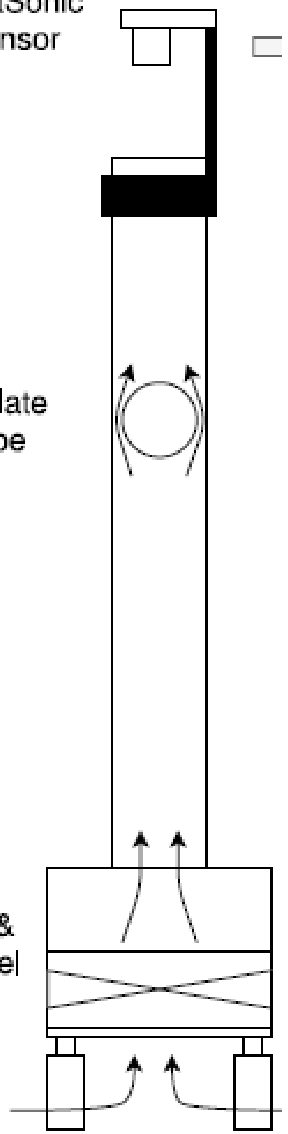
PENDULUM ON CART = CRANE



$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\delta/M & bmg/M & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -b\delta/ML & -b(m+M)g/ML & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 1/M \\ 0 \\ b/ML \end{bmatrix} u$$

SHOW IN PYTHON

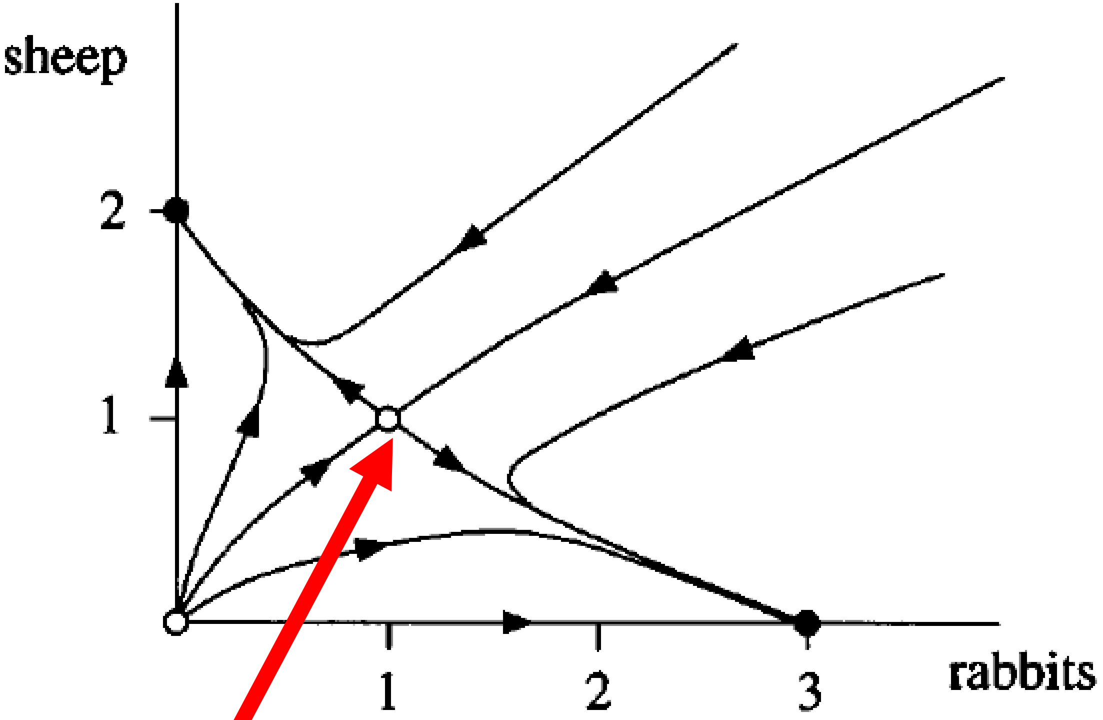
LEVITATION OF BALL



$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} = \begin{bmatrix}0 & 1\\ 0 & -2\sqrt{\frac{mg}{c}}\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} + \begin{bmatrix}0\\ 2\sqrt{\frac{mg}{c}}\end{bmatrix}u$$

$$\lambda_{1,2} = 0, -2\sqrt{\frac{mg}{c}}$$

COMPETITION MODEL



Intervene by adding/removing sheep or rabbits

=FEEDBACK

$$\begin{bmatrix}\bar{x}_1\\\bar{x}_2\end{bmatrix}=\begin{bmatrix}0\\0\end{bmatrix},\begin{bmatrix}0\\2\end{bmatrix},\begin{bmatrix}3\\0\end{bmatrix},\begin{bmatrix}1\\1\end{bmatrix}$$

UNIVERSITEIT GENT

‘FORCED’ RESPONSE

- $\dot{x} = Ax + Bu$
- $x(t) = \int_0^t e^{A(t-\tau)} Bu(\tau) d\tau$ (convolution integral)

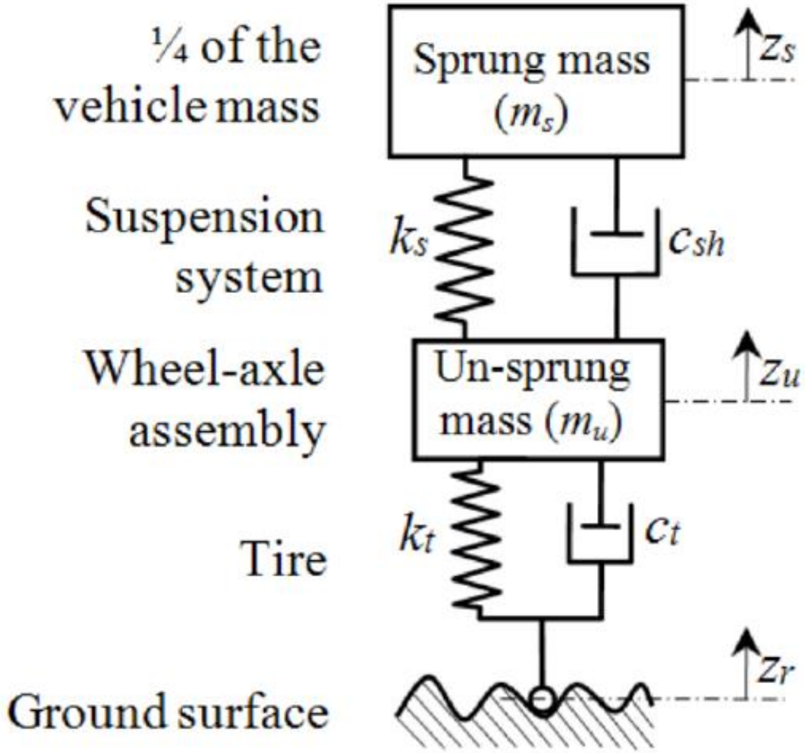
- Forced response
  - Transient determined by eigenvalues & vectors
  - Steady state part, weighted version of u

- Python example: Intro to control toolbox
- Response to frequency: gain and phase

- See later (Input/Output & Laplace)

UNIVERSITEIT GENT

EXAMPLE FORCED: SUSPENSION



Parameter	Value
Sprung mass m_s	332 kg
Unsprung mass m_u	100 kg
Suspension spring k_s	48000 N/m
Tire spring k_t	200 000 N/m
Suspension damper c_s	1000 Nm/s

Lajqi, S., & Pehan, S. (2012).
Designs and optimizations of active and semi-active non-linear suspension systems for a terrain vehicle.
Strojniški vestnik-Journal of Mechanical Engineering, *58*(12), 732-743.

UNIVERSITEIT GENT

STATE FEEDBACK

- Eigenvalues determine stability, speed & oscillations
- Can we choose u such that we can change the eigenvalues? STATE FEEDBACK = $u = -Kx$

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

$$\dot{x} = \underbrace{(A - BK)}_{\text{New system matrix}} x$$

- Requires **FULL** state knowledge

UNIVERSITEIT GENT

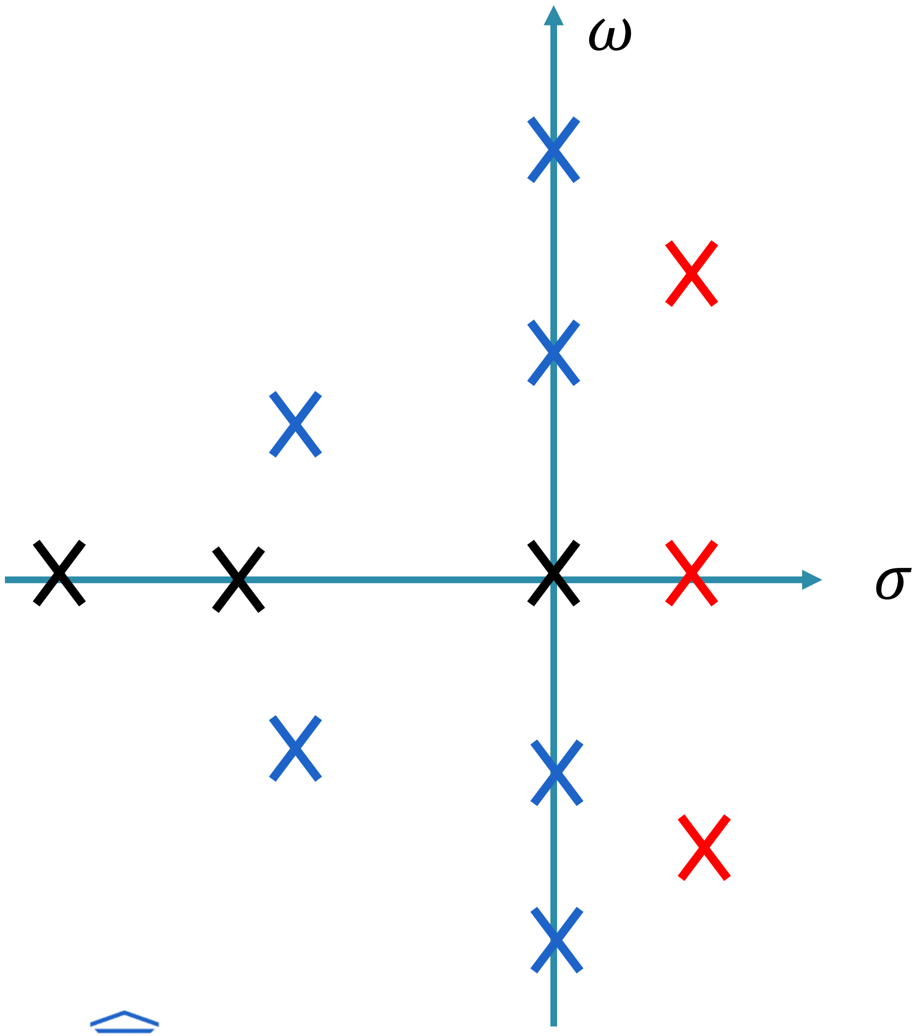
CONTROLLABILITY

- Controllability matrix

$$C = [B\ AB\ A^2B\ \cdots\ A^{n-1}B]$$

- If rank matrix is n, system is controllable
  - Arbitrary eigenvalue placement through state feedback $u = -Kx$
  - Reachability: any state x can be reached in finite time, for some u
- Python inverted pendulum

POLE PLACEMENT



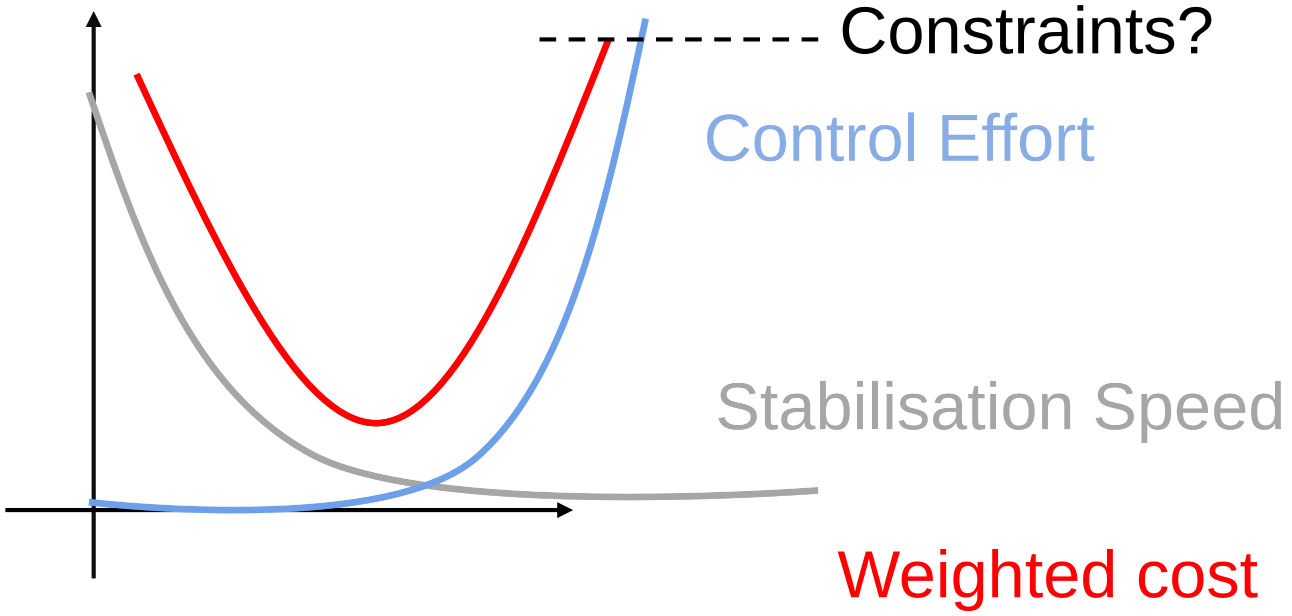
- Eigenvalue on the real axis
- As negative as possible?

$$\dot{x} = \underbrace{(A - BK)}_{\text{New system matrix}} x$$

- Python command ‘Place’
- Inverted pendulum example

POLE PLACEMENT

- Why is there a balance?
- Nonlinear might not be actually stable??



UNIVERSITEIT GENT

DESIGNING FEEDBACK: LINEAR QUADRATIC CONTROL

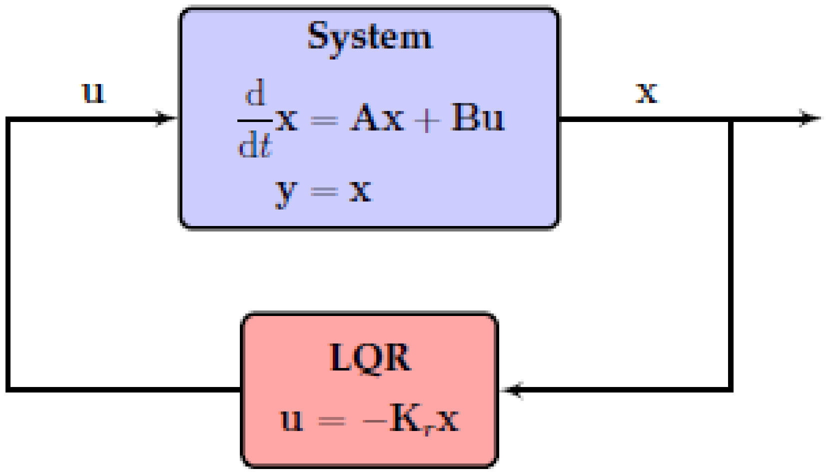
- Controllabilty means a $u = -Kx$ can stabilize and change eigenvalues. *How to choose K ?*
- *Weight optimization between fast stabilization and input cost*
- $J = \int_0^{\infty} (x^T Q x + u^T R u)\, dt \qquad J = \sum_{n=0}^{\infty} x^T Q x + u^T R u$

- Q: Weight matrix states
- R: Weight matrix input ("Energy")

UNIVERSITEIT GENT

DESIGNING FEEDBACK: LINEAR QUADRATIC CONTROL

- $J = \sum_{n=0}^{\infty} x^T Q x + u^T R u$
- Optimal $K_r = R^{-1} B^T S$
- Where S is the solution of a 'stationary Ricatti' equation:
- $A^T S + SA - SBR^{-1} B^T S + Q = 0$
- Inverted pendulum example



UNIVERSITEIT GENT

DESIGNING FEEDBACK: LQR REFERENCE

- We want $x = x_r$
- $\dot{x}^* = Ax^* + Bu^* \;\; where\; x^* = x - \; x_r$ and $u^* = u - \; u_r$
- Stable if $u^* = -\mathrm{K}x^*$
- $0 = Ax_r + Bu_r$
- So same optimization holds, different meaning input cost

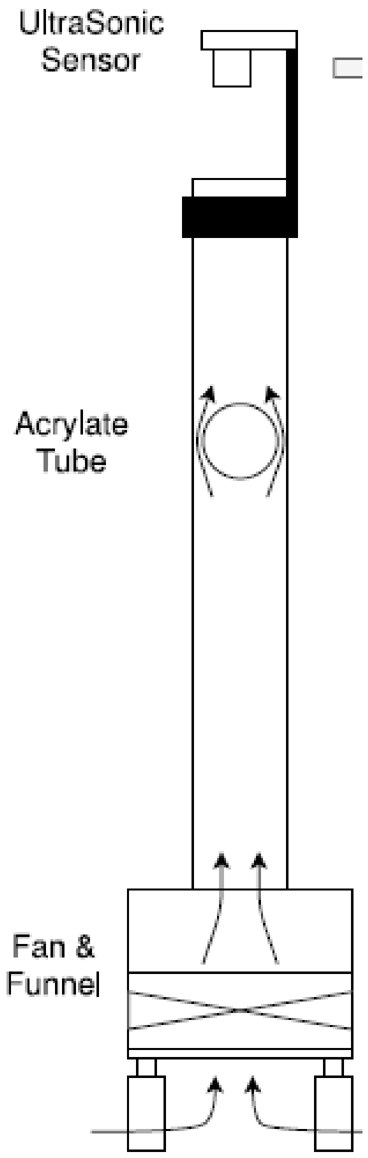
UNIVERSITEIT GENT

STATE ESTIMATION

- Until now: Knowledge full state x

- Output equation! $y = Cx + Du$

- Not all state can be measured because of limited sensors, measurement locations



UNIVERSITEIT GENT

OBSERVABILITY

– Observability matrix $O = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix}$

– If rank matrix is n, system is observable
  - any state x can be estimated from y in finite time, for some u
  - Similar like controllability second meaning!

UNIVERSITEIT GENT

OBSERVER

Estimated dynamics based on y

$$\dot{\hat{x}} = A\hat{x} + Bu + K_f(y - \hat{y})$$
$$\hat{y} = C\hat{x}$$
$$\dot{\hat{x}} = (A - K_f C)\hat{x} + [B\ K_f]\begin{bmatrix} u \\ y \end{bmatrix}$$

Estimation error dynamics: $\epsilon = x - \hat{x}$

$$\dot{\epsilon} = (A - K_f C)\epsilon$$

Pole placement/stabilisation! But how to choose K_f?

UNIVERSITEIT GENT

KALMAN FILTER

Observer gain optimization under system and measurement noise

$$\dot{x} = Ax + Bu + w_d, \qquad w_d \in \mathbb{R}^{n\times 1},\ in\ \mathcal{N}(0, \sigma_d^2)$$
$$y = Cx + Du + w_n, \qquad w_n \in \mathcal{N}(0, \sigma_n^2)$$

$\mathbb{E}(w_d)$=$\mathbb{E}(w_n)$=0 $\qquad \mathbb{E}(w_d w_d^T)$=$V_d$ $\qquad \mathbb{E}(w_n w_n^T)$=$V_n$

$$\dot{\epsilon} = (A - K_f C)\epsilon + w_d - K_f v_k \qquad J = \mathbb{E}(\epsilon\epsilon^T)$$

Optimal $K_f = SC^T V_n^{-1}$

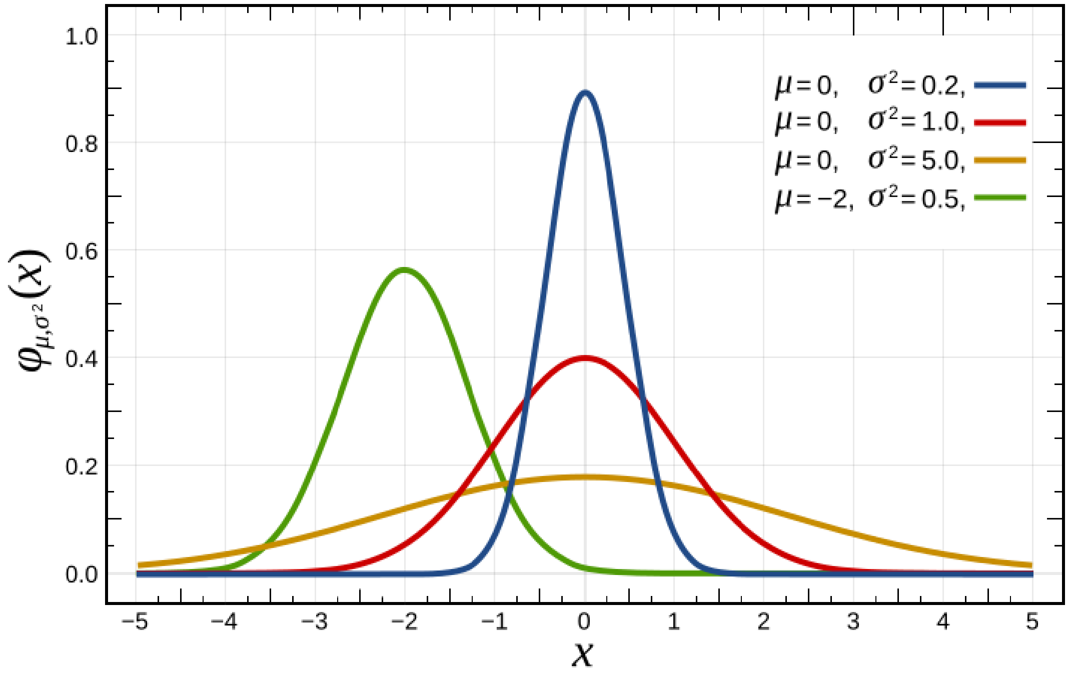
Where S is the solution of a 'stationary Ricatti' equation:

$$SA^T + AS - SC^T V_n^{-1} CS + V_d = 0$$

Large input noise: Trust measurement, more gain

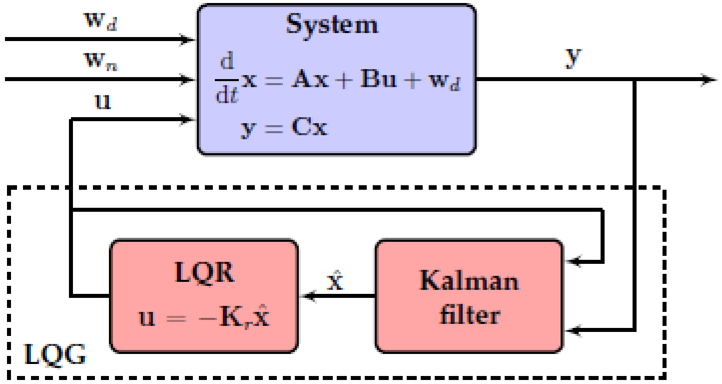
Large output noise: Trust estimation, less gain

Matlab example



UNIVERSITEIT GENT

ESTIMATION & CONTROL



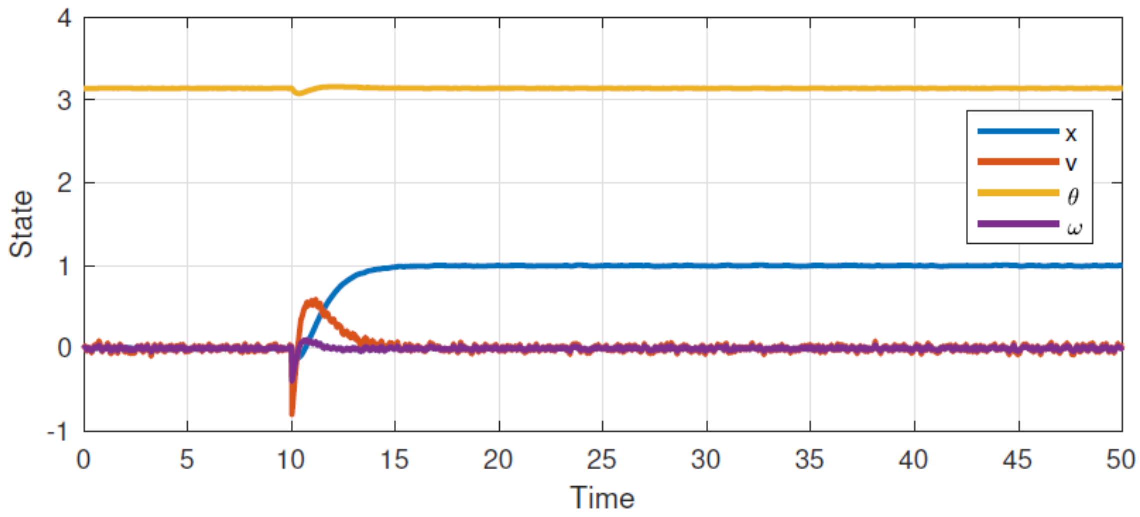
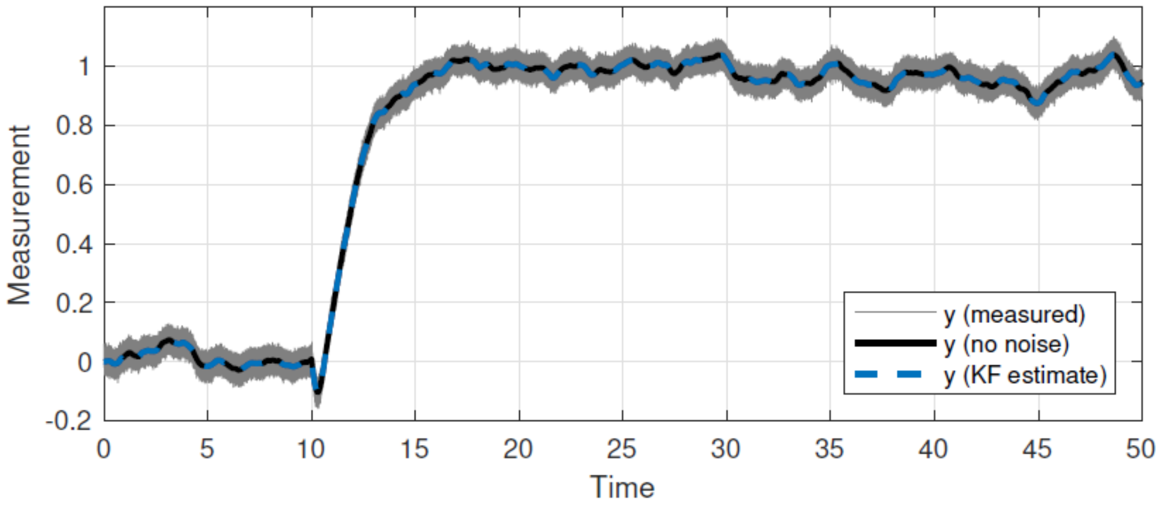


Figure 8.18: Output response using LQG feedback control.



EXAMPLE: INVERTED PENDULULM

- Robustness? Does control still work if model estimation wrong?
- https://www.youtube.com/watch?v=D3bblng-Kcc

UNIVERSITEIT GENT

ADVANCED TOPICS?

- Nonlinear Kalman
- Reduced observer (not all states)
- Stopgap: Linear Parameter varying:

$$\dot{x} = A(t)x + B(t)u$$

- Advanced libraries usually MATLAB & SIMULINK

REFERENCES

- Linear Systems (Course Ghent University, Gert De Cooman)
- Control System Design and Management (Course University of Newcastle, Julio Braslavsky)
- Brunton, S. L., & Kutz, J. N. (2022). *Data-driven science and engineering: Machine learning, dynamical systems, and control*. Cambridge University Press.
- Chevalier, A., Dekemele, K., Juchem, J., & Loccufier, M. (2021). Student feedback on educational innovation in control engineering: active learning in practice. *IEEE Transactions on Education*, *64*(4), 432-437.
- Strogatz, S. H. (2018). *Nonlinear dynamics and chaos: with applications to physics, biology, chemistry, and engineering*. CRC press.

UNIVERSITEIT GENT

INPUT-OUTPUT MODELS

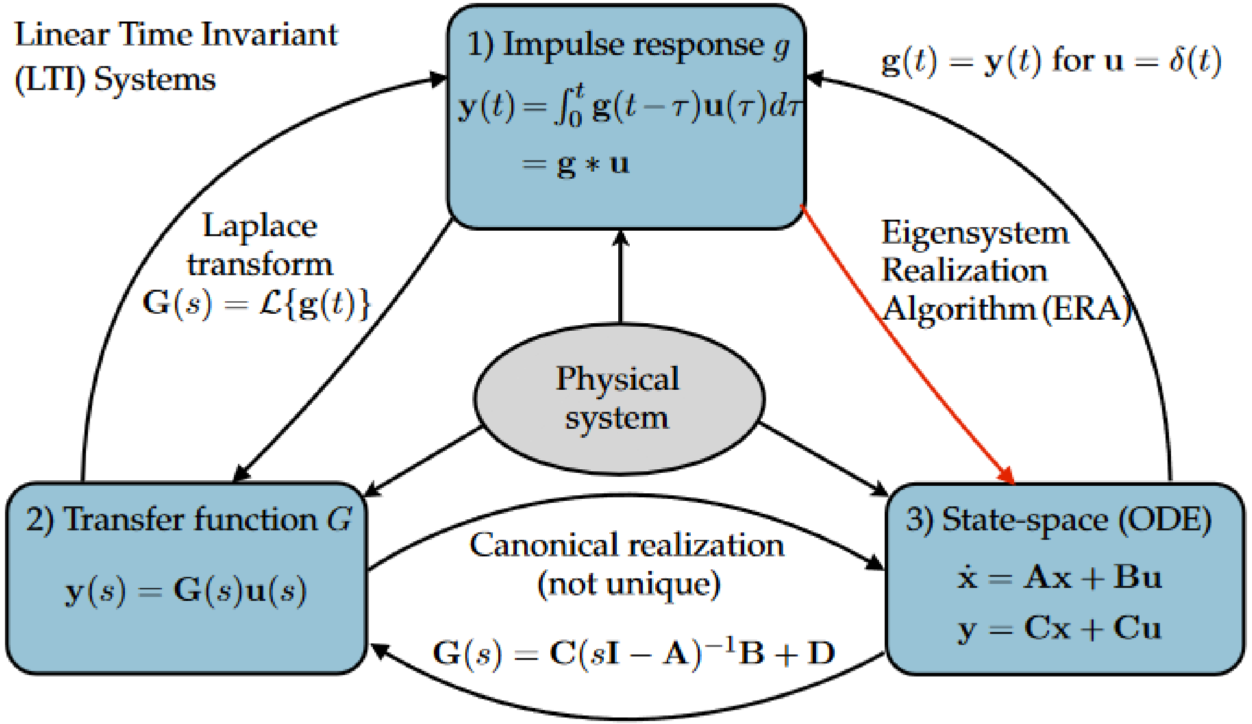
UNIVERSITEIT GENT

LAPLACE TRANSFORM

- The Laplace transform:

$$\mathcal{L}\left\{\frac{d}{dt}f(t)\right\} = \int_{0^-}^{\infty} \underbrace{\frac{d}{dt}f(t)}_{dv}\,\underbrace{e^{-st}}_{u}\,dt$$
$$= \left[f(t)e^{-st}\right]_{t=0^-}^{t=\infty} - \int_{0^-}^{\infty} f(t)(-se^{-st})dt$$
$$= f(0^-) + s\mathcal{L}\{f(t)\}.$$

- In short: to go from differential - to algebraic equations



UNIVERSITEIT GENT

FROM STATE-SPACE TO INPUT-OUTPUT MODELS

- We are interested in the relationship between the inputs and (measurable) outputs (i.e. not necessarily all the states)
- Consider a linear system

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx + Du \end{cases}$$

Often: $D = 0$

$G(s) = C(sI - A)^{-1}$

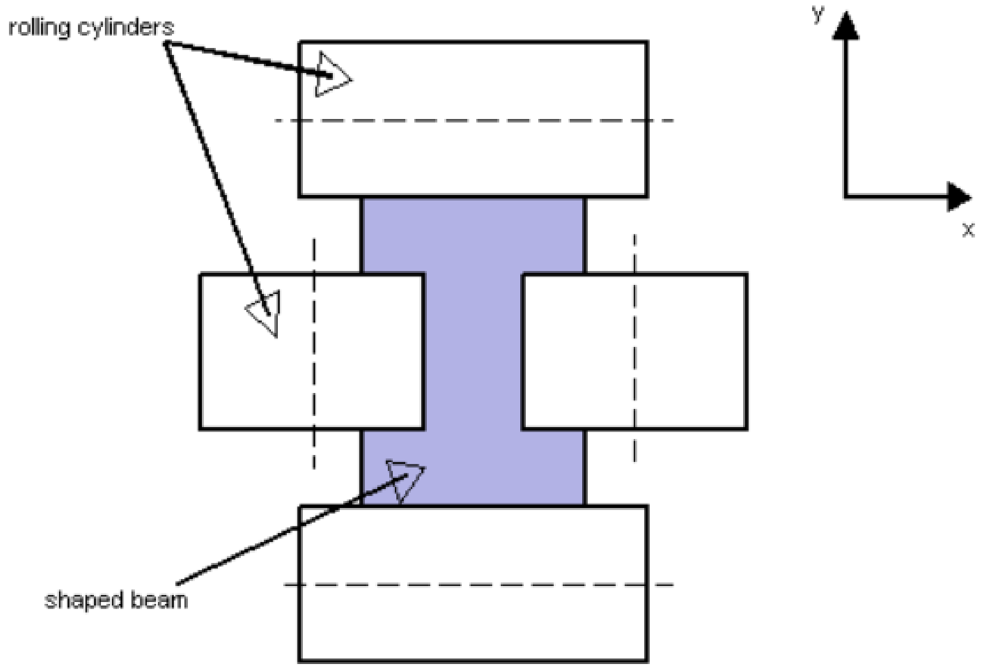
- Laplace transform:

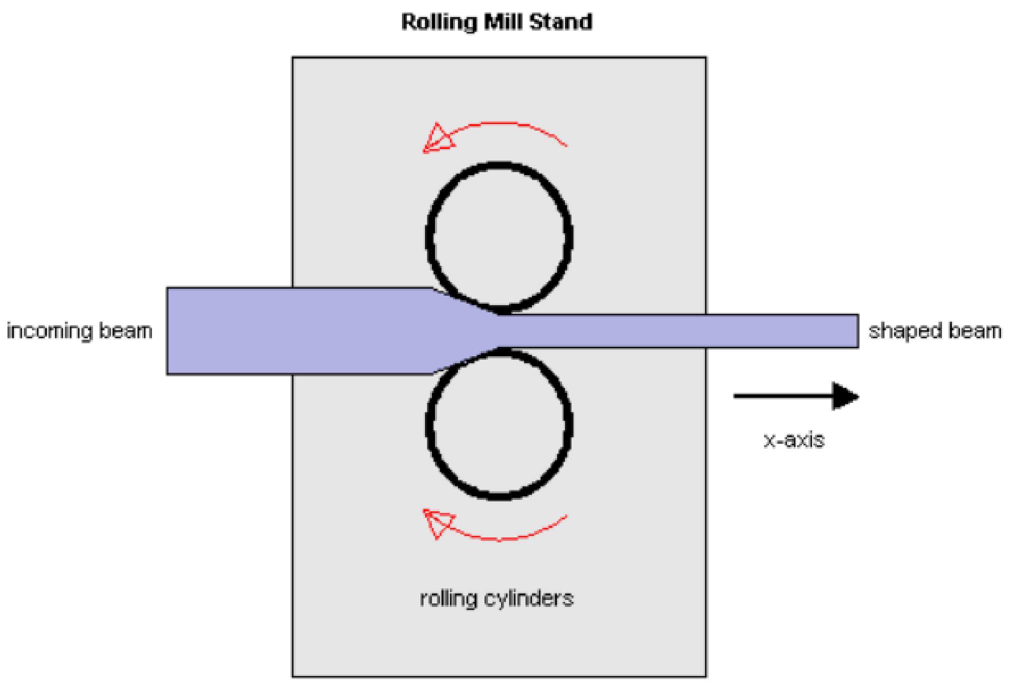
$$\begin{cases} sX = AX + BU \\ Y = CX + DU \end{cases} \implies Y = \underbrace{\left[C\left(sI - A\right)^{-1} + D\right]}_{G(s)} U$$

UNIVERSITEIT GENT

MIMO CONTROL

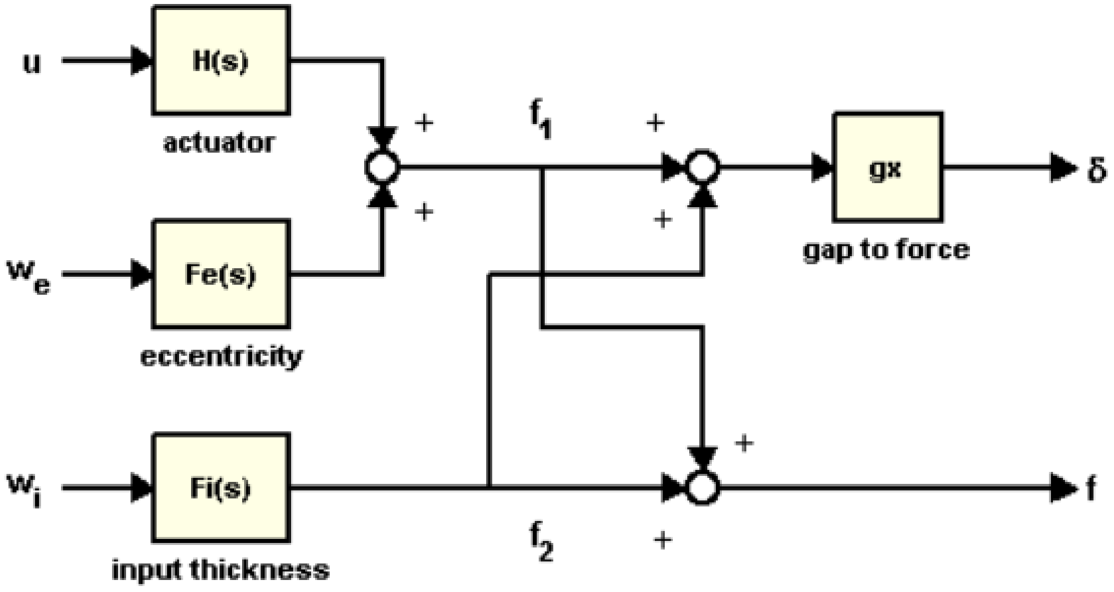
BEAM ROLLING: EXAMPLE





- Goal: track static reference thickness w_r

BEAM ROLLING: PROCESS (X-DIRECTION)



- u: controlled input
- w_e: eccentricity (disturbance)
- w_i: input thickness (disturbance)

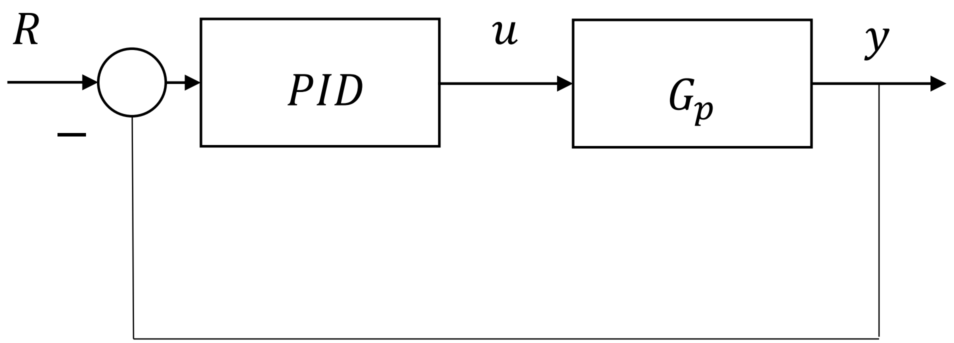
- δ: gap thickness (not measurable)
- f: rolling force (measurable)

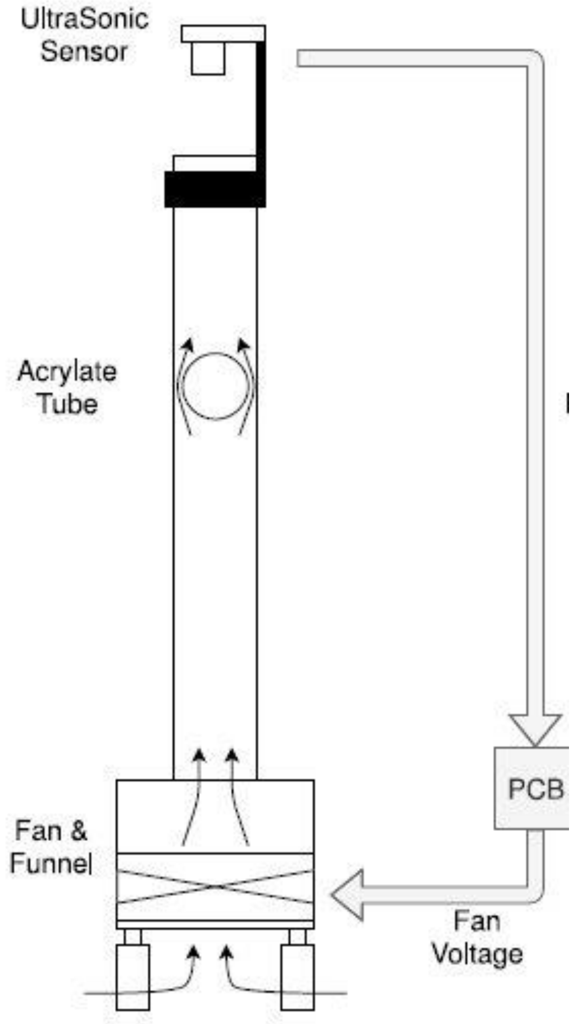
- Dynamics

$$H_x = \frac{2.4 \times 10^8}{s^2 + 72s + 90^2} \qquad F_{ex} = \frac{3 \times 10^4 s}{s^2 + 0.125s + 6^2} \qquad F_{ix} = \frac{10^4}{s + 0.05}$$

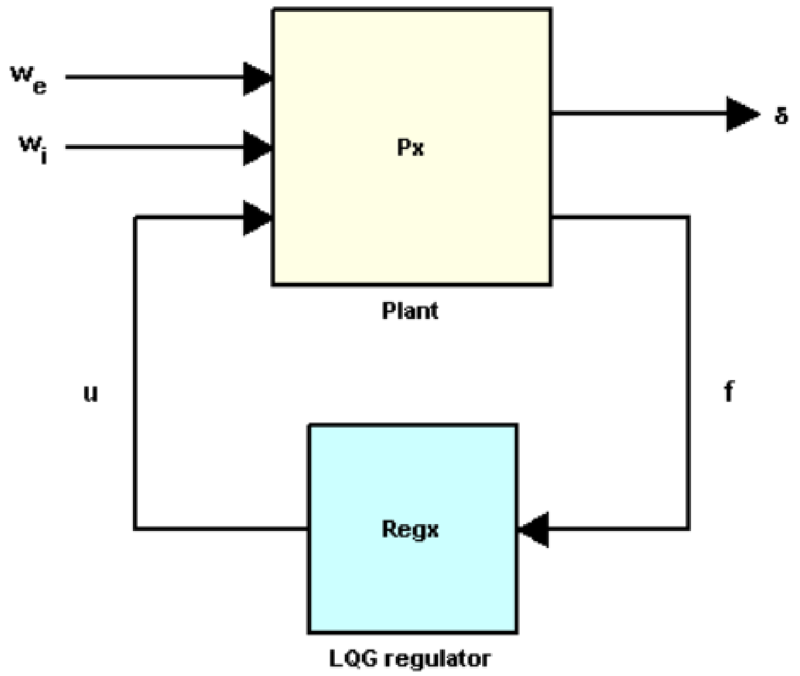
BASIC CONTROL: PID

- Easy tuning rules:
  - Oscillation test results in K_u
  - E.g. Ziegler-Nichols: K_P, K_I, K_D
- More involved tuning based on some quality labels:
  - Overshoot
  - Settling time





BEAM ROLLING: LQR CONTROL (X-DIRECTION)



- Define cost function:

$$C(u) = \int_0^\infty y^T Q y + u^T R u dt$$

- Wherein

$$Q \in \mathbb{R}^{p \times p}$$

$$R \in \mathbb{R}^{m \times m}$$

- Result: control law

$$u = -K(x - w_r), \quad K \in \mathbb{R}^{m \times n}$$

- But…

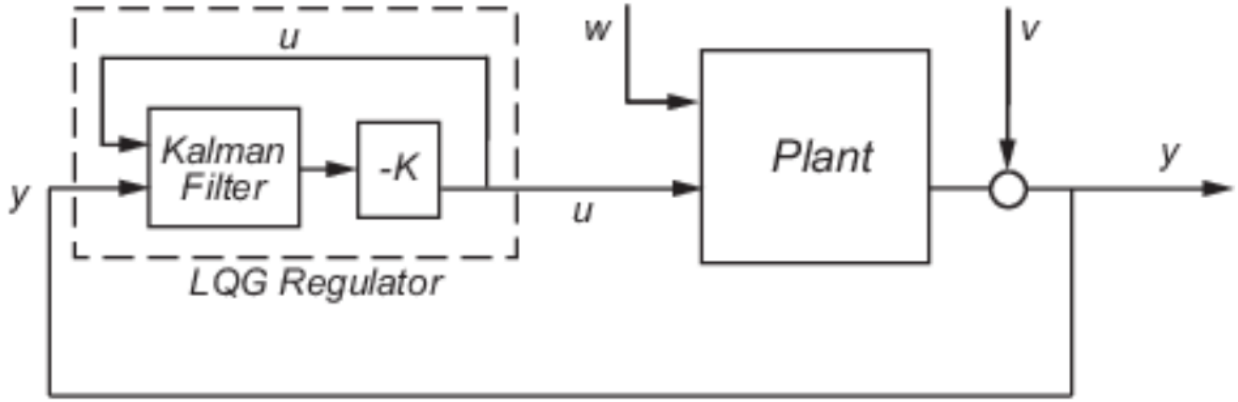
$$x???$$

Python exercise

UNIVERSITEIT GENT

LQG CONTROL

- Boils down to: Kalman + LQR



- Kalman to estimate the state

$$y \xrightarrow{F_{Kalman}} \hat{x}$$

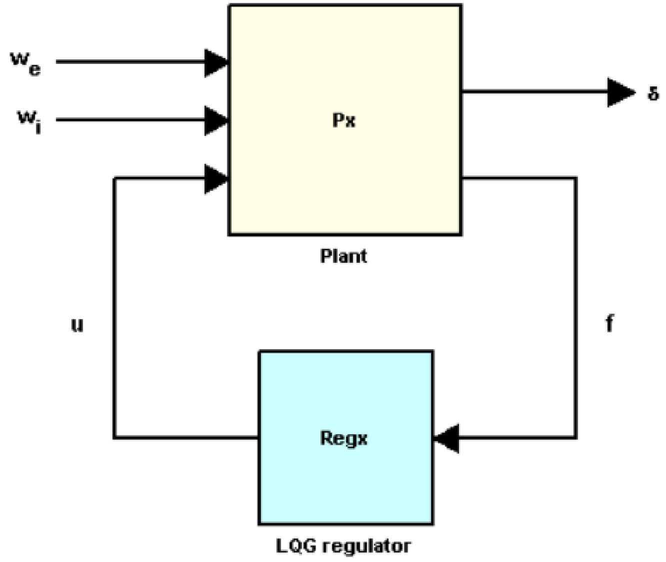
- u: plant input
- y: plant output
- w: input disturbance
- v: output disturbance

- LQR, using an estimate of the full state:

$$u = -K\hat{x}, \quad K \in \mathbb{R}^{m \times n}$$

UNIVERSITEIT GENT

BEAM ROLLING: LQG CONTROL (X-DIRECTION)



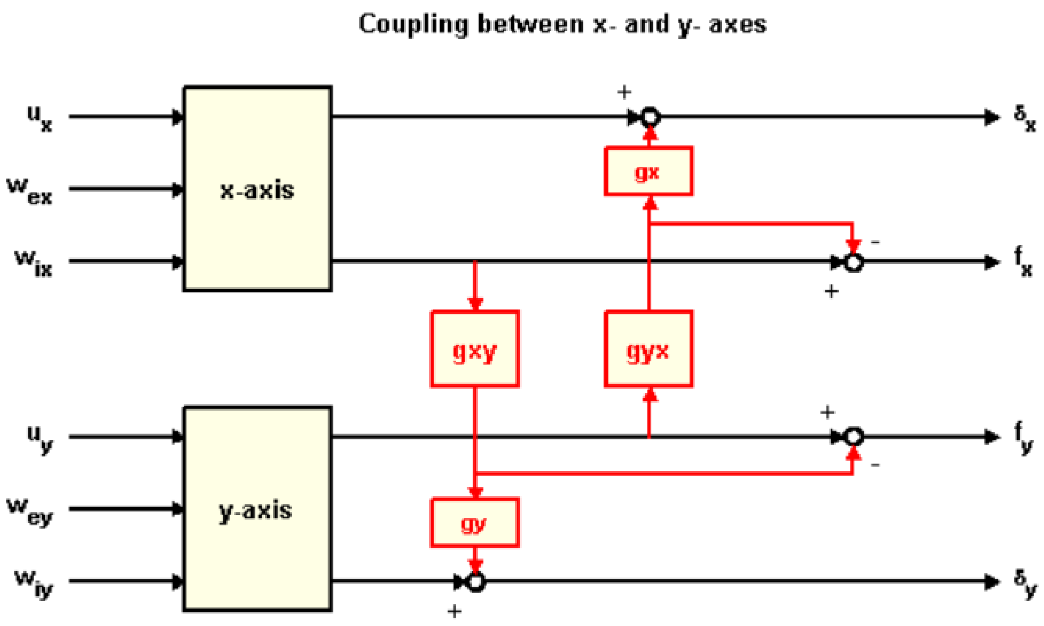
- u: controlled input
- w_e: disturbance (eccentricity)
- w_i: disturbance (input thickness)

- δ: gap thickness (not measurable)
- f: rolling force (measurable)

Python implementation

UNIVERSITEIT GENT

BEAM ROLLING: X & Y DIRECTION



- Interactions exist:
  - Example: Poisson coefficient
- Complicating control:
  - Input u_x influences f_y
  - Input u_y influences f_x
  - Cross coupling
- How to deal with this?

DECOUPLED CONTROL

- Coupling between u_i and y_j exist
- How to deal with this?
    1. Ignore coupling (Decentralised Control):
        - LQR or LQG for separated models → suboptimal solution
        - ‘What is good for one might be bad for the other’
    2. Take coupling into account (Centralised Control):
        - Difficult to comprehend
        - LQR or LQG: complicated model and often suboptimal
    3. Decouple in control scheme!
- To what degree do u_j and y_i interact?
    - Relative Gain Array (RGA)

DECOUPLED CONTROL: RELATIVE GAIN ARRAY (1)

- Consider the input-output model: $y(s) = G(s)u(s)$

- Gain from u_j to y_i (all other loops open):

$$\left.\frac{\partial y_i}{\partial u_j}\right|_{u_k=cst,k\neq j} = g_{ij}$$

- Gain from u_j to y_i (all other loops closed):

$$\left.\frac{\partial y_i}{\partial u_j}\right|_{y_k=cst,k\neq j} = \hat{g}_{ij}$$

- With:

$$\hat{g}_{ij} = \frac{1}{[G^{-1}]_{ij}}$$

- Relative gain:

$$\lambda_{ij} := \frac{g_{ij}}{\hat{g}_{ij}} = [G]_{ij}[G^{-1}]_{ji}$$

UNIVERSITEIT GENT

DECOUPLED CONTROL: RELATIVE GAIN ARRAY (2)

- Apply a step input Δu_j keeping u_k constant ($\mathrm{k} \neq j$) and measure the output change Δy_i

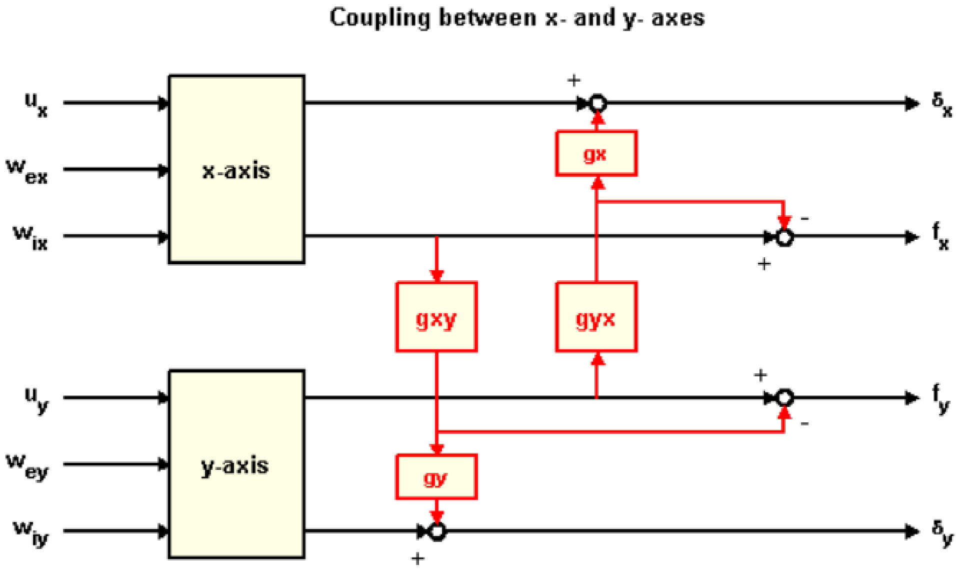
$$g_{ij} = \left.\frac{\Delta y_i}{\Delta u_j}\right|_{u_k=cst,k\neq j}$$

- Apply a step input Δu_j keeping y_k constant ($\mathrm{k} \neq j$) (i.e. keeping all other loops closed and measure the output change Δy_i

$$\hat{g}_{ij} = \left.\frac{\Delta y_i}{\Delta u_j}\right|_{y_k=cst,k\neq j}$$

UNIVERSITEIT GENT

BEAM ROLLING: DECOUPLING CONTROL (1)



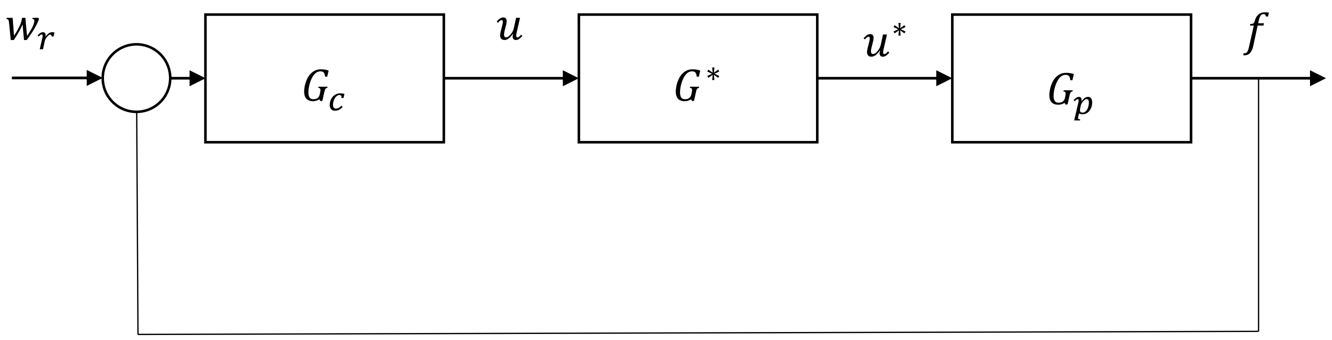
- Input-output model:

$$y(s) = G(s)u(s), \quad G(s) \in \mathbb{R}^{4\times 6}$$

- With:

$$y = \begin{bmatrix} \delta_x \\ f_x \\ \delta_y \\ f_y \end{bmatrix} \quad \& \quad u = \begin{bmatrix} u_x \\ w_{ex} \\ w_{ix} \\ u_y \\ w_{ey} \\ w_{iy} \end{bmatrix}$$

BEAM ROLLING: DECOUPLING CONTROL (2)



G^*? → Exercise

- G^* such that inputs and outputs are decoupled i.e. no interaction between u_x and f_y and u_y and f_x
- We aim at static decoupling: $\lim_{s\to 0} sG(s) = \tilde{G}_{stat}$

UNIVERSITEIT GENT

BEAM ROLLING: DECOUPLING CONTROL (3)

- We aim at static decoupling: $\lim_{s \to 0} sG(s) = \tilde{G}_{stat}$
- Consider the new input-output model:

$$\begin{bmatrix} f_x \\ f_y \end{bmatrix} = \tilde{G}_{stat} \begin{bmatrix} u_x \\ u_y \end{bmatrix}$$

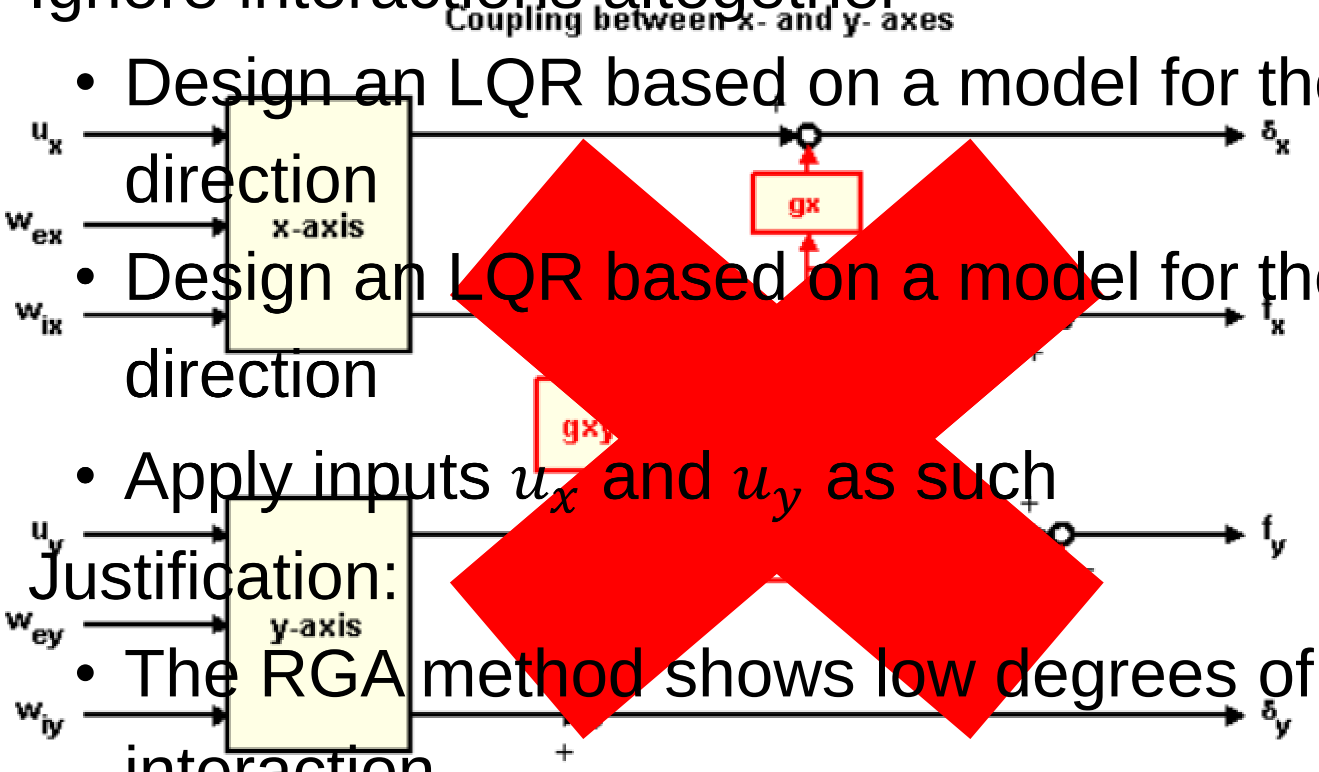
- Wherein:

$$\tilde{G}_{stat} = \begin{bmatrix} 29629.63 & -95686.98 \\ -28148.15 & 100723.14 \end{bmatrix}$$

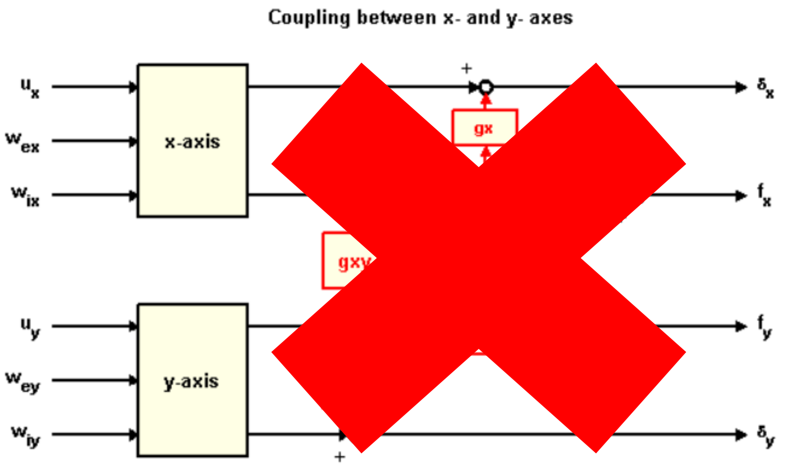
- Goal: find $\tilde{G}_{stat}^{\,*}$ for $y = \tilde{G}_{stat}\tilde{G}_{stat}^{\,*} u$ such that y and u are decoupled

UNIVERSITEIT GENT

BEAM ROLLING: DECENTRALISED CONTROL



- Ignore interactions altogether
  - Design an LQR based on a model for the x-direction
  - Design an LQR based on a model for the y-direction
  - Apply inputs u_x and u_y as such
- Justification:
  - The RGA method shows low degrees of interaction
  - Note: coupling w_e & w_i w.r.t. f are not considered



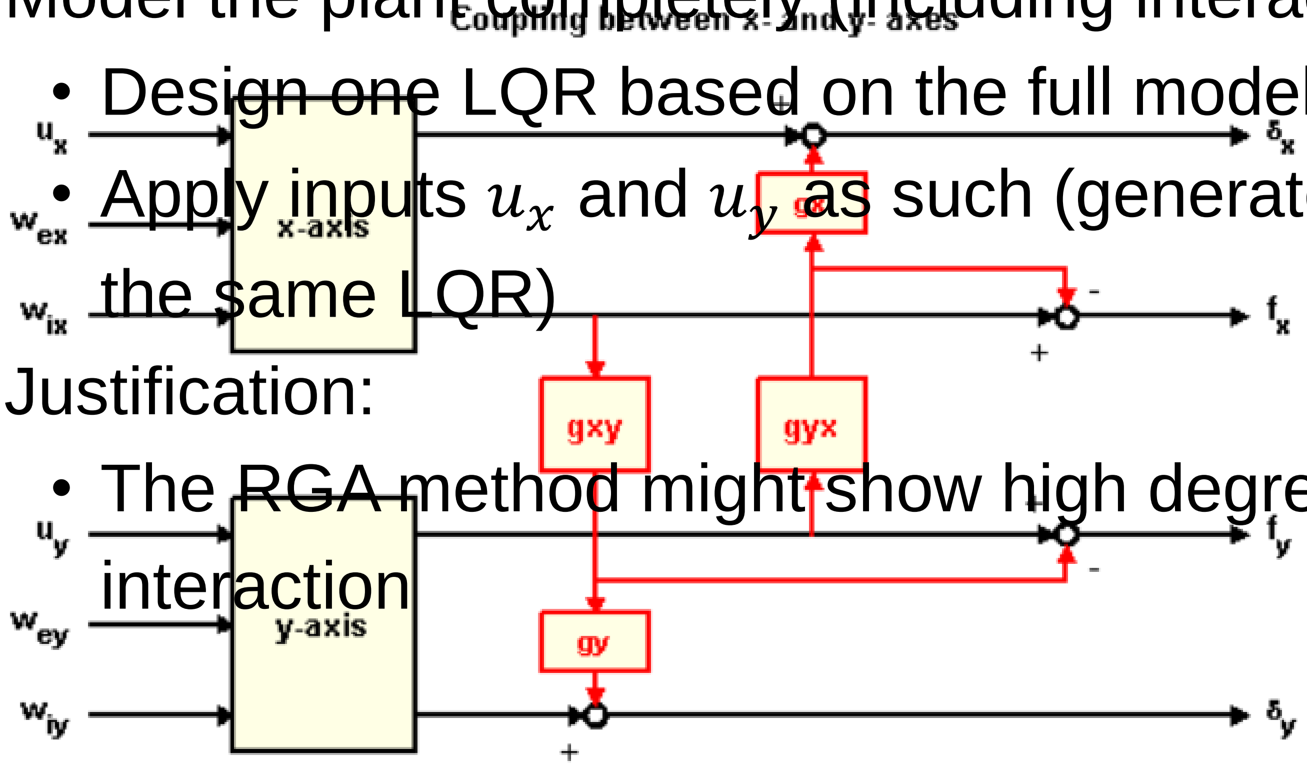
RGA-matrix

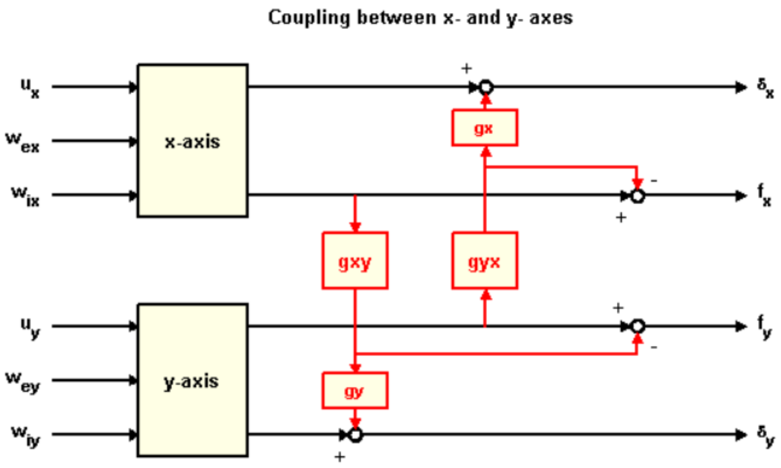
$$\Lambda = \begin{bmatrix} 1.014 & -0.014 \\ -0.014 & 1.014 \end{bmatrix}$$

UNIVERSITEIT GENT

BEAM ROLLING: CENTRALISED CONTROL

- Model the plant completely (including interactions)
  - Design one LQR based on the full model
  - Apply inputs u_x and u_y as such (generated by the same LQR)
- Justification:
  - The RGA method might show high degrees of interaction

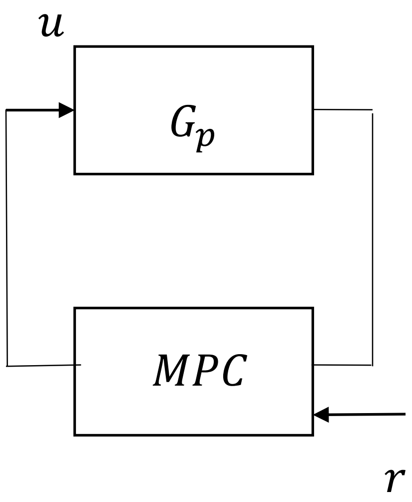




UNIVERSITEIT GENT

MODEL PREDICTIVE CONTROL (MPC):

- Unaddressed topics so far:
  - Constraint equations
  - Nonlinear models
  - Dead time
  - Dealing measured disturbances
- → MPC

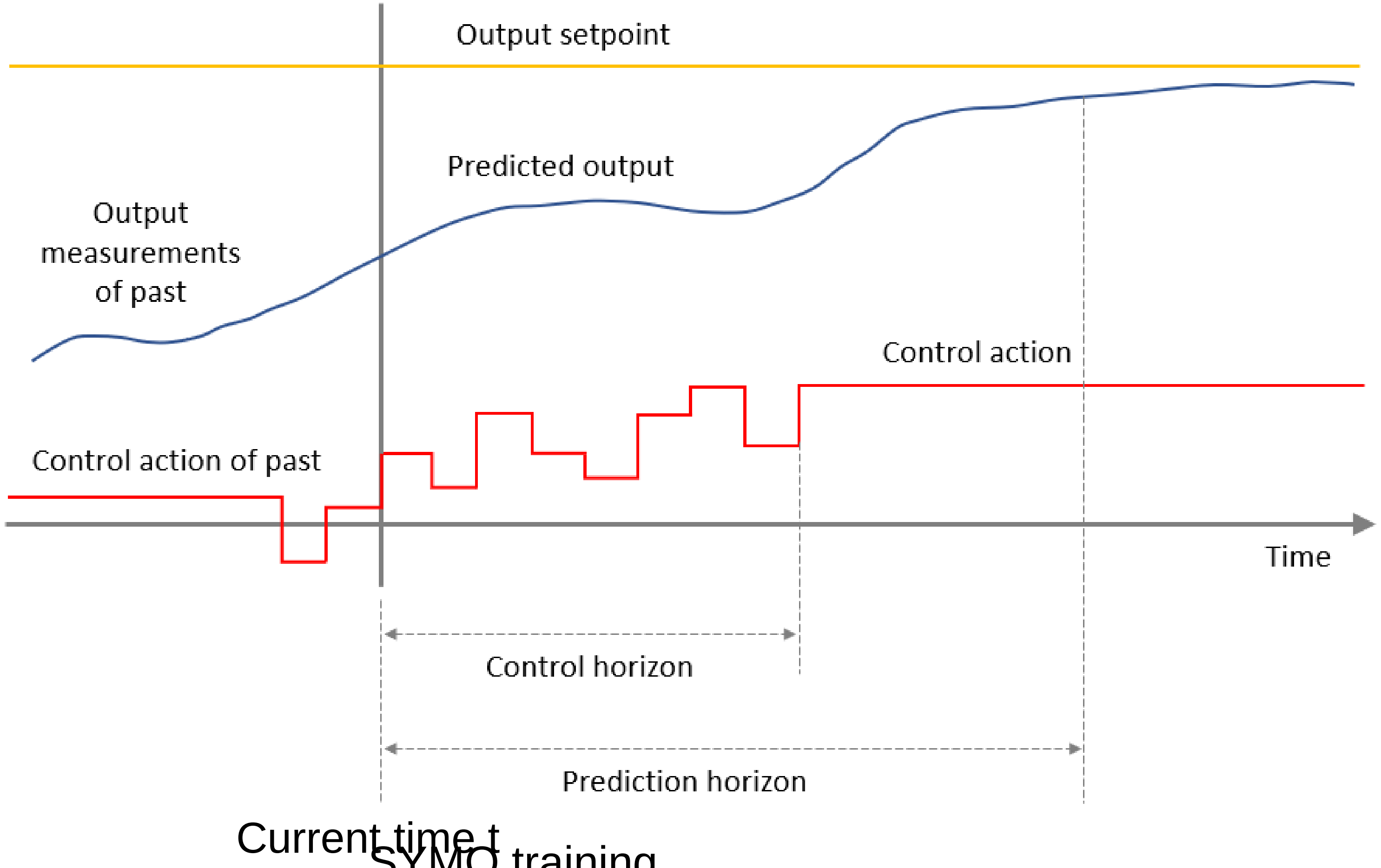


UNIVERSITEIT GENT

MPC:

MPC :

- Feedback control for MIMO systems
- Input based on real-time optimisation
- Optimisation based on current state and predicted future output



UNIVERSITEIT GENT

MPC

- Based on a model (state-space):

$$\begin{cases} x(k+1) = Ax(k) + Bu(k) \\ y(k) = Cx(k) + Du(k) \end{cases}$$

- And a well-defined cost function: $J(x, u, y)$

- Taking into account some constraints:

$$U_{i,min} \leq u_i(k) \leq U_{i,max}$$

$$X_{i,min} \leq x_i(t) \leq X_{i,max}$$

- Minimisation of the cost function, based on the predicted output $y(t+k|t), k = 1 \dots N$ (and therefore the predicted error) results in $u(t+k|t), k = 1 \dots N$ over the prediction horizon

- Only $u(t+1|t)$ applied

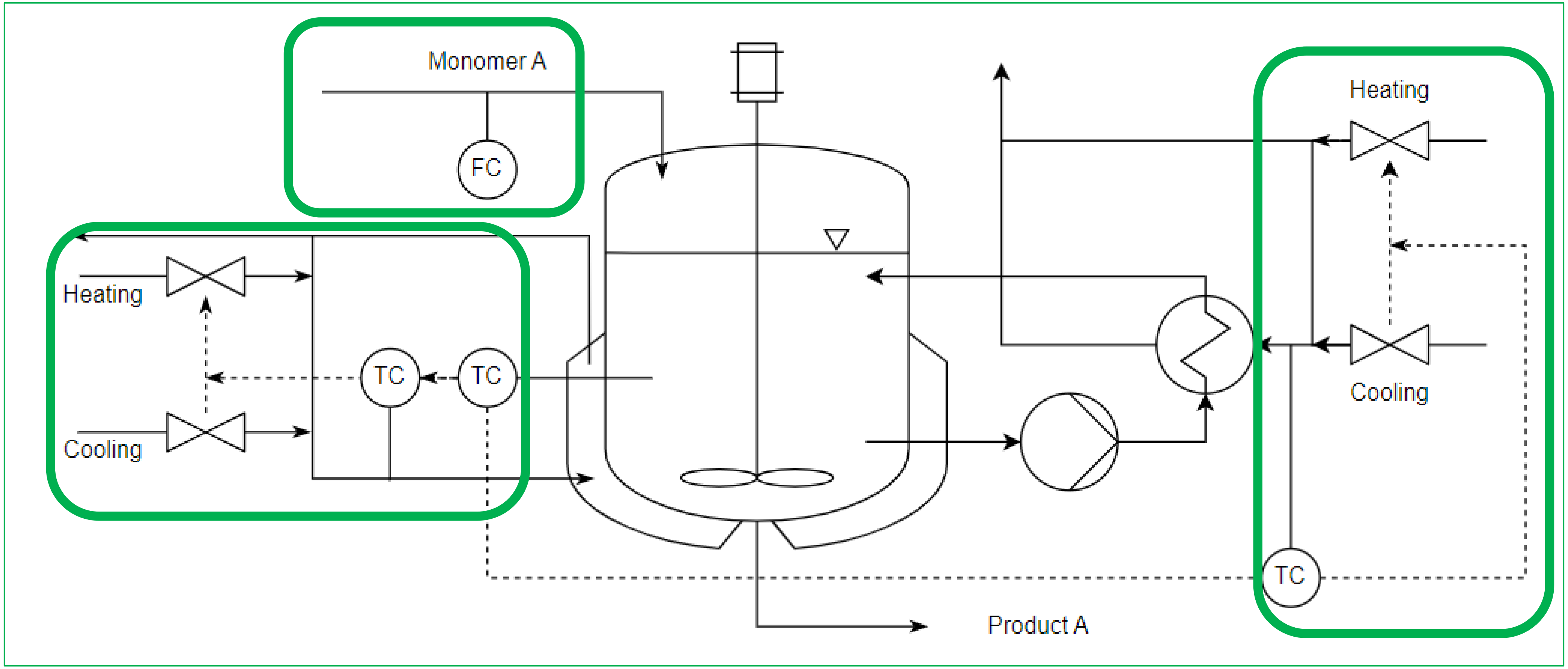
Possibly: $(N_u < N)$

$$u(t+k) = \begin{cases} u(t+k), & k < N_u \\ u(t+N_u), & k \geq N_u \end{cases}$$

UNIVERSITEIT GENT

PROCESS: INDUSTRIAL POLYMERISATION REACTOR

Input



UNIVERSITEIT GENT

Ref: https://www.do-mpc.com/en/latest/example_gallery/industrial_poly.html

MODEL THROUGH CONSERVATION

Parameter uncertainty

$$
\begin{aligned}
\dot m_{\mathrm{W}} &= \dot m_{\mathrm{F}}\,\omega_{\mathrm{W,F}}\\
\dot m_{\mathrm{A}} &= \dot m_{\mathrm{F}}\omega_{\mathrm{A,F}} - k_{\mathrm{R1}}\,m_{\mathrm{A,R}} - k_{\mathrm{R2}}\,m_{\mathrm{AWT}}\,m_{\mathrm{A}}/m_{\mathrm{ges}},\\
\dot m_{\mathrm{P}} &= k_{\mathrm{R1}}\,m_{\mathrm{A,R}} + p_1\,k_{\mathrm{R2}}\,m_{\mathrm{AWT}}\,m_{\mathrm{A}}/m_{\mathrm{ges}},\\
\dot T_{\mathrm{R}} &= 1/(c_{\mathrm{p,R}}m_{\mathrm{ges}})\,[\dot m_{\mathrm{F}}\,c_{\mathrm{p,F}}\,(T_{\mathrm{F}}-T_{\mathrm{R}}) + \Delta H_{\mathrm{R}} k_{\mathrm{R1}} m_{\mathrm{A,R}} - k_{\mathrm{K}}A\,(T_{\mathrm{R}}-T_{\mathrm{S}})\\
&\quad - \dot m_{\mathrm{AWT}}\,c_{\mathrm{p,R}}\,(T_{\mathrm{R}}-T_{\mathrm{EK}})],\\
\dot T_{S} &= 1/(c_{\mathrm{p,S}}m_{\mathrm{S}})\,[k_{\mathrm{K}}A\,(T_{\mathrm{R}}-T_{\mathrm{S}}) - k_{\mathrm{K}}A\,(T_{\mathrm{S}}-T_{\mathrm{M}})],\\
\dot T_{\mathrm{M}} &= 1/(c_{\mathrm{p,W}}m_{\mathrm{M,KW}})\,[\dot m_{\mathrm{M,KW}}\,c_{\mathrm{p,W}}\,(T_{\mathrm{M}}^{\mathrm{IN}}-T_{\mathrm{M}})\\
&\quad + k_{\mathrm{K}}A\,(T_{\mathrm{S}}-T_{\mathrm{M}})] + k_{\mathrm{K}}A\,(T_{\mathrm{S}}-T_{\mathrm{M}})],\\
\dot T_{\mathrm{EK}} &= 1/(c_{\mathrm{p,R}}m_{\mathrm{AWT}})\,[\dot m_{\mathrm{AWT}}c_{\mathrm{p,W}}\,(T_{\mathrm{R}}-T_{\mathrm{EK}}) - \alpha\,(T_{\mathrm{EK}}-T_{\mathrm{AWT}})\\
&\quad + k_{\mathrm{R2}}\,m_{\mathrm{A}}\,m_{\mathrm{AWT}}\Delta H_{\mathrm{R}}/m_{\mathrm{ges}}],\\
\dot T_{\mathrm{AWT}} &= [\dot m_{\mathrm{AWT,KW}}\,c_{\mathrm{p,W}}\,(T_{\mathrm{AWT}}^{\mathrm{IN}}-T_{\mathrm{AWT}}) - \alpha\,(T_{\mathrm{AWT}}-T_{\mathrm{EK}})]/(c_{\mathrm{p,W}}m_{\mathrm{AWT,KW}}),
\end{aligned}
$$

$$
m_F^{acc} = \dot m_F
$$

$$
T_{adiab} = \frac{\Delta H_R}{c_{p,R}}\frac{m_A}{m_A + m_W + m_P} + T_R
$$

$$
\begin{aligned}
U &= m_{\mathrm{P}}/(m_{\mathrm{A}}+m_{\mathrm{P}}),\\
m_{\mathrm{ges}} &= m_{\mathrm{W}} + m_{\mathrm{A}} + m_{\mathrm{P}},\\
k_{\mathrm{R1}} &= k_0\,e^{\frac{-E_a}{R(T_{\mathrm{R}}+273.15)}}\,(k_{\mathrm{U1}}\,(1-U) + k_{\mathrm{U2}}U),\\
k_{\mathrm{R2}} &= k_0\,e^{\frac{-E_a}{R(T_{\mathrm{EK}}+273.15)}}\,(k_{\mathrm{U1}}\,(1-U) + k_{\mathrm{U2}}U),\\
k_{\mathrm{K}} &= (m_{\mathrm{W}}k_{\mathrm{WS}} + m_{\mathrm{A}}k_{\mathrm{AS}} + m_{\mathrm{P}}k_{\mathrm{PS}})/m_{\mathrm{ges}},\\
m_{\mathrm{A,R}} &= m_{\mathrm{A}} - m_{\mathrm{A}}m_{\mathrm{AWT}}/m_{\mathrm{ges}}.
\end{aligned}
$$

UNIVERSITEIT GENT

CONSTRAINTS

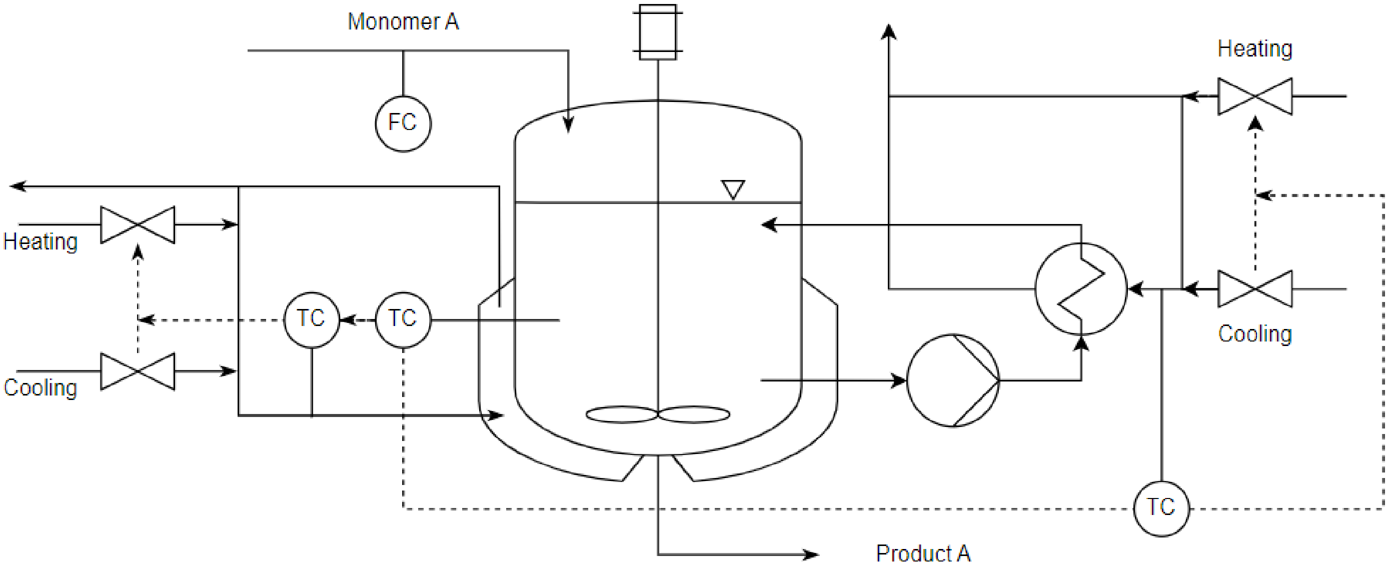
Control	Min.	Max.	Unit
$\dot{m}_F$	0	30,000	kg h$^{=-1}$
T_M^{IN}	60	100	°C
T_{AWT}^{IN}	60	100	°C

State	Init. cond.	Min.	Max.	Unit
m_W	10,000	0	inf.	kg
m_A	853	0	inf.	kg
m_P	26.5	0	inf.	kg
T_R	90.0	$T_{set} - 2.0$	$T_{set} + 2.0$	°C
T_S	90.0	0	100	°C
T_M	90.0	0	100	°C
T_{EK}	35.0	0	100	°C
T_{AWT}	35.0	0	100	°C
T_{adiab}	104.897	0	109	°C
m_F^{acc}	0	0	30,000	kg

OBJECTIVE

- Produce $m_P = 20680\ kg$ as fast as possible
- Smooth control performance: penalty on input changes

$$J(x,u,z) = \sum_{k=0}^{N} \left(\underbrace{l(x_k, z_k, u_k, p_k, p_{\text{tv},k})}_{\text{Lagrange term}} + \underbrace{\Delta u_k^T R \Delta u_k}_{\text{r-term}} \right) + \underbrace{m(x_{N+1})}_{\text{Mayer term}}$$



UNIVERSITEIT GENT