



Aalto-yliopisto  
Sähkötekniikan  
korkeakoulu

# ELEC-C1110

## Automaatio- ja systeemi- tekniikan perusteet

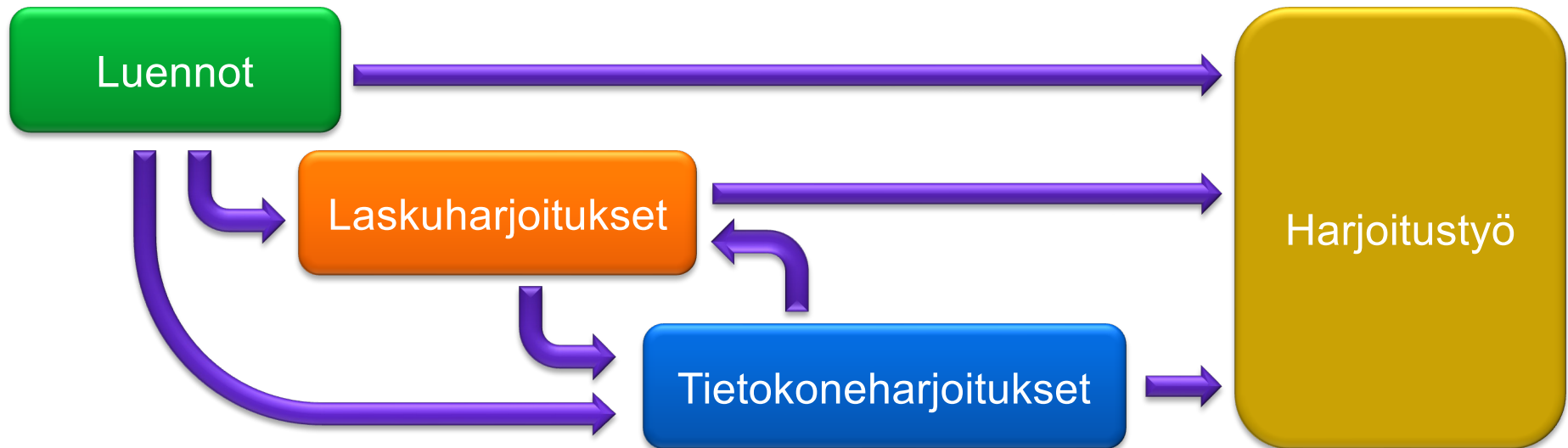
Luento 10

Kertaus ja yhteenveto kurssin asioista

25.3.2024, Joni Pajarinen

# Kurssin eteneminen

- Mallintaminen (luennot + laskari + tietokoneharj.)
- Dynamiikka (luennot + laskari + tietokoneharj.)
- Sääto (luennot + laskari + tietokoneharj.)
- Kinematiikka (luennot + laskari + tietokoneharj.)



# Lista kurssin aiheista

- Automaatio- ja systeemitekniikan määritelmät, taustaa, historiaa
- Takaisinkytkentä ja säätö, avoin/suljettu järjestelmä, peruseriaate
- SISO/MIMO-prosessit
- Dynaamisen systeemin mallintaminen, lohkokaavioesitys
- Numeerinen integrointi
- Mallin kalibrointi näytteistä, lineaarinen malli, epälineaarinen malli, pienimmän neliövirheen minimointi
- Gradienttimenetelmä, kantafunktioimalli, neuroverkko
- Valmiin ja tuntemattoman mallin yhdistäminen
- Virhelähteet mittauksessa
- Anturit, eri anturityyppejä
- Toimilaitteet, näytteistys
- Takaisinkytkentä ja säätö yleisesti
- P/PI/PD/PID –säätimet
- Dynaamiset ominaisuudet aikatasossa, stabiilius, värähtely
- Roboteista yleisesti
- Robotin suora kinematiikka
- Robotin käänteinen kinematiikka
- Robotin liikekinematiikka
- Robotin nivelkulmien säätö
- Robotin liiketrajektorin säätö
- Robotin säädön myötäkytkentä, dynamiikan huomiointi
- Uudelleen suunnittelu
- Kaskadisäätö

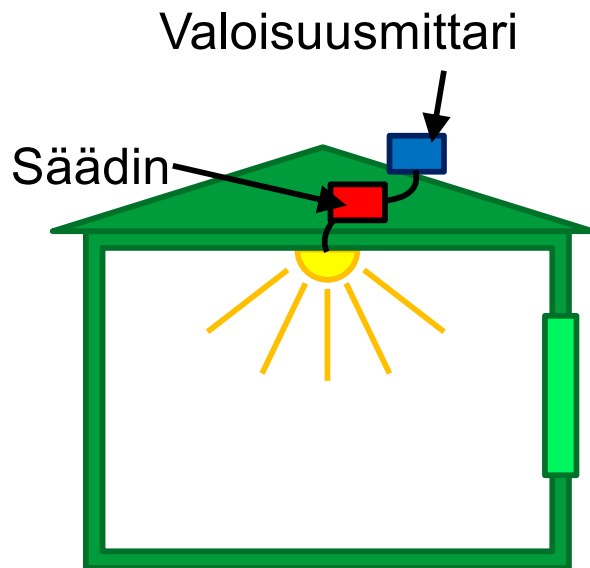
# Automaatio- ja systeemitekniikan määritelmät, taustaa, historiaa

- Automaatio: Toiminta ilman ihmisen ohjaavaa tai suorittavaa osuutta
- Systeemitekniikka: Oppi dynaamisista järjestelmistä ja niiden ohjaamisesta
- Sovelluksia
  - Prosessiteollisuus ja energiantuotanto
  - Kappaletavarateollisuus
  - Yksittäiset koneet ja laitteet

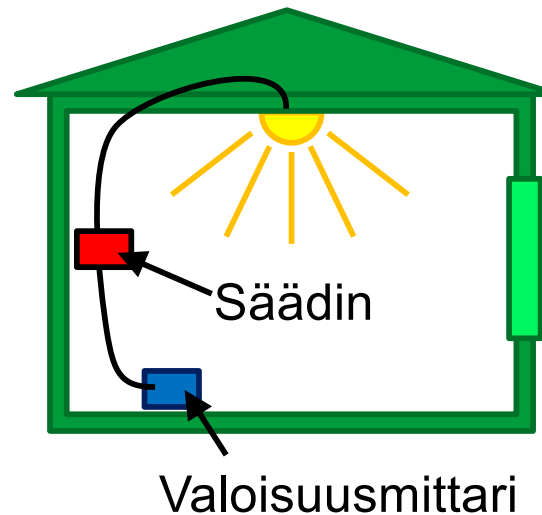
# Mallintaminen

# Takaisinkytkentä ja säätö, avoin/suljettu järjestelmä, perusperiaate

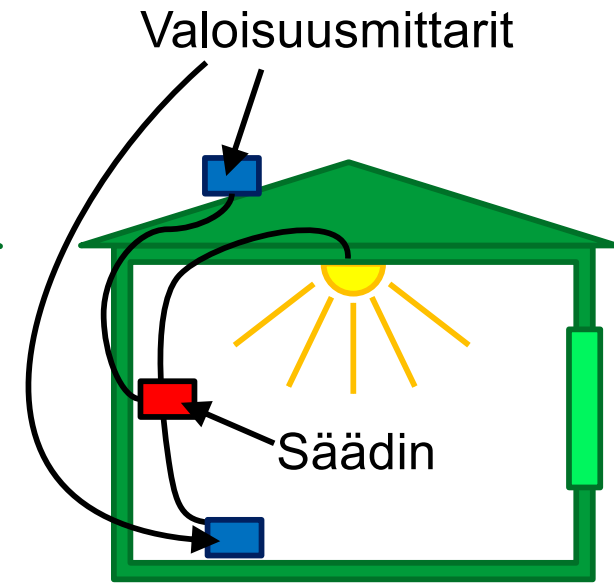
## Avoin järjestelmä



## Suljettu järjestelmä

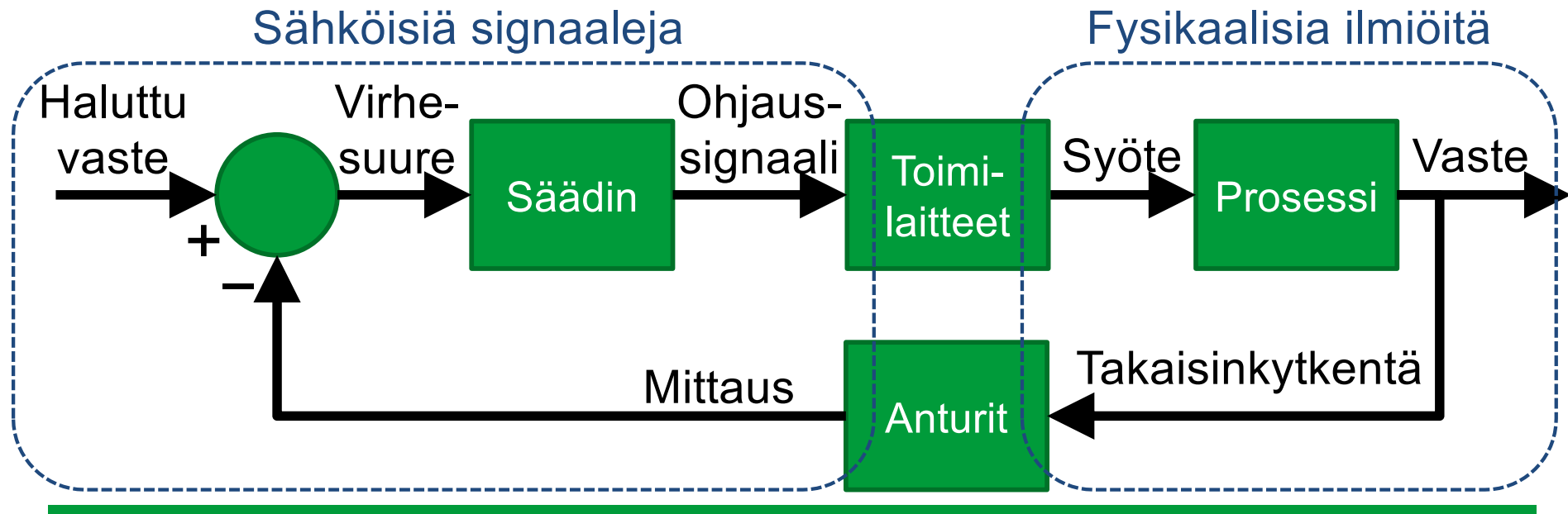


## Hybridi



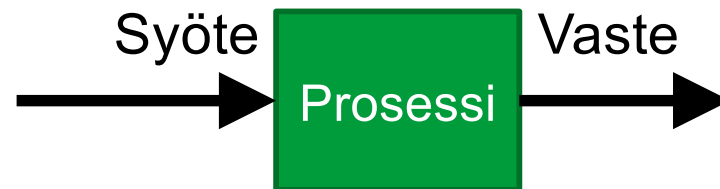
# Takaisinkytkentä ja säätö, avoin/suljettu järjestelmä, perusperiaate

- Mittaus ohjattavasta prosessista
- Mitataan ohjauksen aiheuttamaa reaktiota
- Takaisinkytkentä (negatiivinen)

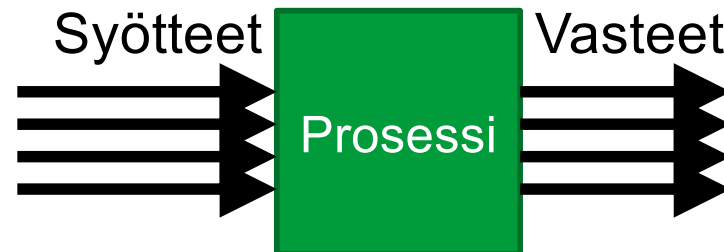


# SISO ja MIMO

- SISO: Single Input Single Output



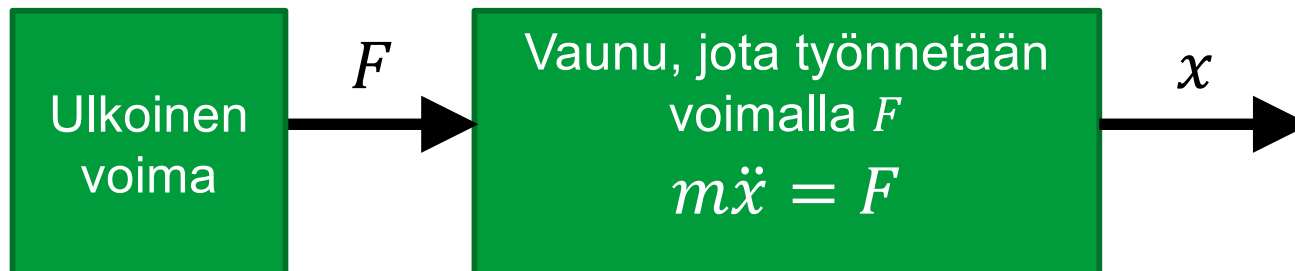
- MIMO: Multiple Input Multiple Output



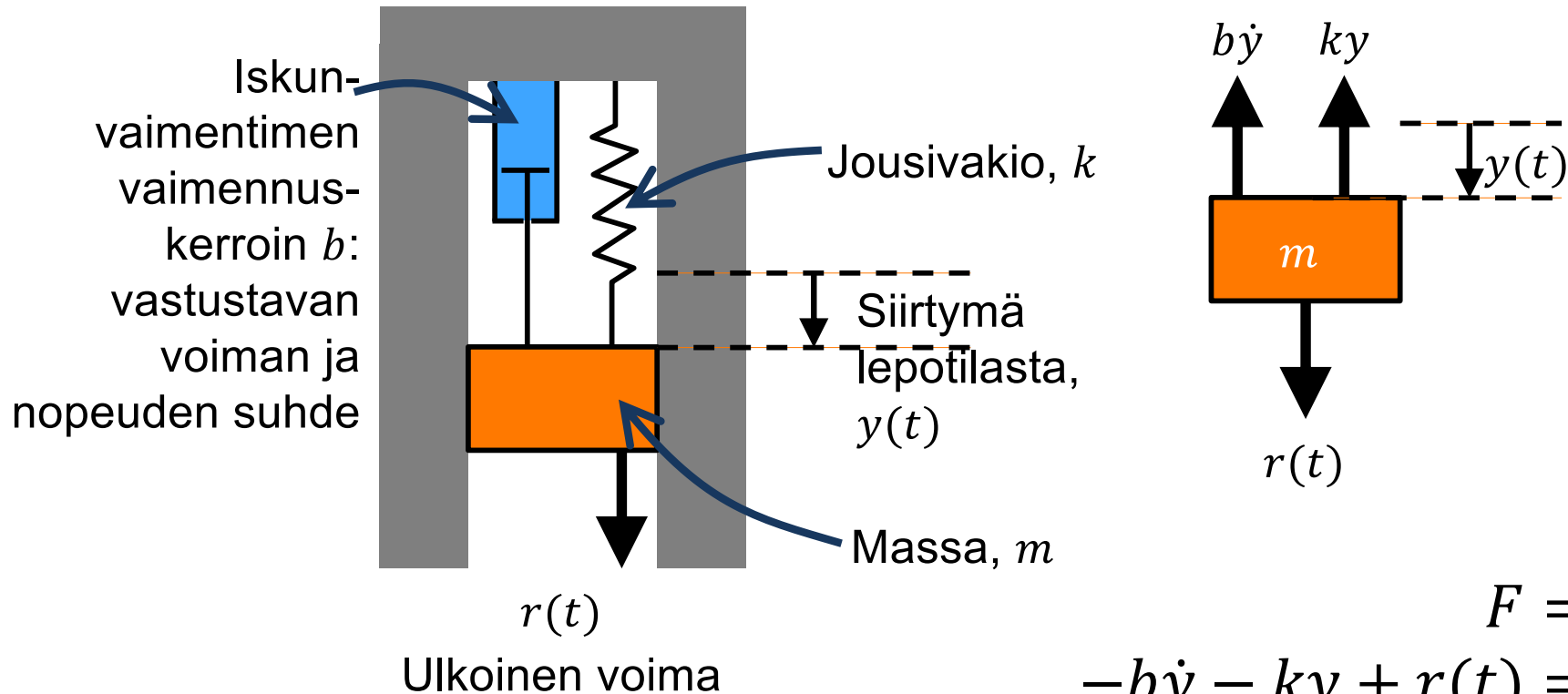


# Dynaamisen systeemin mallintaminen, lohkokaavioesitys

- Systemillä syötteet ja vasteet
- Systemin dynaamiset ominaisuudet kuvataan differentiaaliyhtälöillä
  - Perustuvat fysikaaliseen malliin tms. sääntöihin
- Lohkokaavioilla voidaan visualisoida systeemin osien vuorovaikutuksia, esim:



# Jousi-massa-vaimennin -järjestelmä

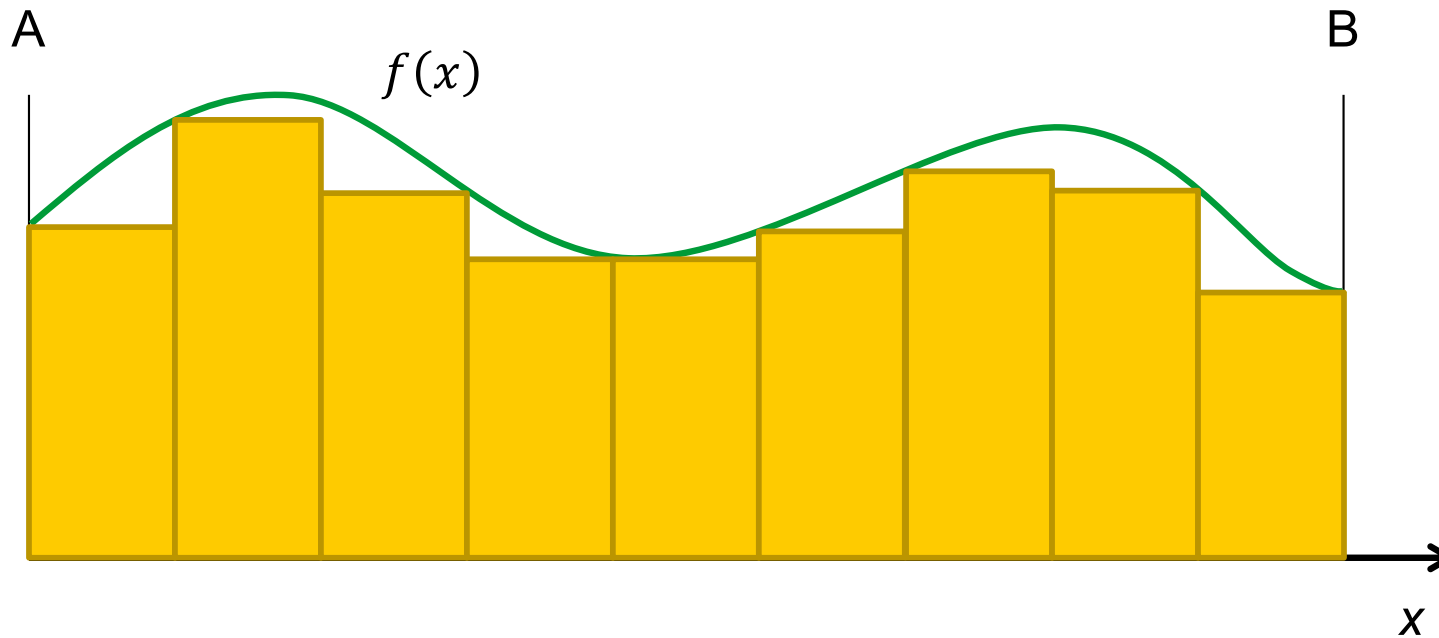


$$F = ma$$
$$-b\dot{y} - ky + r(t) = m\ddot{y}$$
$$m\ddot{y} + b\dot{y} + ky = r(t)$$

Toisen kertaluvun lineaarinen vakiokertoiminen differentiaaliyhtälö (ODE)

# Numeerinen integrointi

$$\int_A^B f(x) dx \approx \sum \min_{x_1 \leq x \leq x_2} f(x) (x_2 - x_1)$$



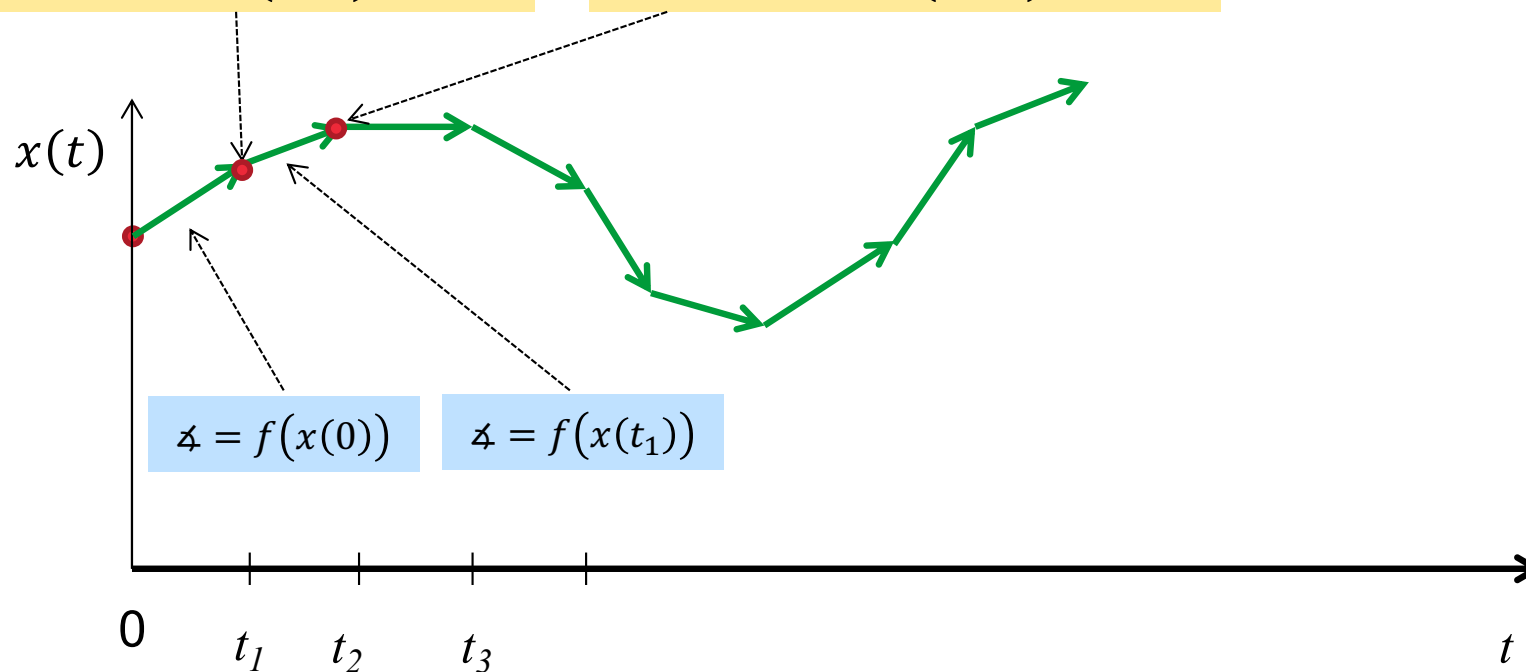
# Numeerinen integrointi (ODE)

esim.  $\dot{x}(t) = 9x(t) + 4 + \sin t$

yl.  $\dot{x}(t) = f(x(t), t)$

$$x(t_1) \approx x(0) + f(x(0))(t_1 - 0)$$

$$x(t_2) \approx x(t_1) + f(x(t_1))(t_2 - t_1)$$



Tavallisen differentiaaliyhtälön ratkaisu numeerisesti: Eulerin menetelmä.  
Voidaan käyttää tarkempia menetelmiä, esim. Runge-Kutta menetelmää.

# Toisen asteen numeerinen integrointi (ODE)

esim.  $\ddot{x}(t) = 3\dot{x}(t) + 9x(t) + 4 + \sin t$

yl.  $\ddot{x}(t) = f(\dot{x}(t), x(t), t)$

Miten voidaan integroida?

Käytetään kahta muuttujaa  $x(t)$  ja  $x_2(t) = \dot{x}(t)$  yhden sijaan:

$$\dot{x}(t) = x_2(t)$$

$$\dot{x}_2(t) = f(x_2(t), x(t), t) = 3x_2(t) + 9x(t) + 4 + \sin t$$

Eulerin menetelmälle saadaan päivitykset:

$$x(t_1) \approx x(0) + x_2(0)(t_1 - 0)$$

$$x(t_2) \approx x(t_1) + x_2(t_1)(t_2 - t_1)$$

$$x_2(t_1) \approx x_2(0) + f(x_2(0), x(0), 0)(t_1 - 0)$$

$$x_2(t_2) \approx x_2(t_1) + f(x_2(t_1), x(t_1), t_1)(t_2 - t_1)$$

→ toisen asteen integrointi muistuttaa ensimmäisen asteen integrointia, mutta käytetään lisämuuttujaa ensimmäisen asteen derivaatalle

# Numeerinen integrointi (ODE)

- Numeerinen ODE integrointi Pythonilla
  - Voidaan käyttää olemassa olevia ohjelmakirjastoja
    - Esim. `scipy.integrate.solve_ivp`
  - Tai tehdään luuppi missä lasketaan integraatio jokaisella ajanhetkellä manuaalisesti tai käyttäen valmista funktiota
- Mahdollista toteuttaa
  - vakio-askelpituuden integrointimenetelmiä (fixed step)
  - muuttuva-askelpituuden integr. menetelmiä (variable step)
- Fixed step
  - monissa yksinkertaisissa tapauksissa ok
  - jos simulaatiossa on sekä hyvin hitaita ja hyvin nopeita yhtä aikaa, on tarpeen miettiä tarkkaan mikä menetelmä toimii

# Kertaustehtävä 1

- Järjestelmän dynamiikkaa kuvaa yhtälö  
 $\ddot{y} - 3\dot{y} + 2y - u = 0$  alkuarvoiksi oletetaan  $y(0s) = 0m$ ,  
 $\dot{y}(0s) = 1m/s$ ,  $\ddot{y}(0s) = 1m/s^2$
- Kirjoita kaava sellaiseen muotoon, että voidaan tehdä numeerinen integrointi.  $u(t)$  on vapaa muuttuja
- Laske Eulerin menetelmällä numeerinen integrointi ajanhetkellä  $t = \Delta t$ , kun  $\Delta t = 0.1s$

# Kertaustehtävä 1 - malli

- Lisätään uusi muuttuja  $y_2(t)$  ensimmäisen asteen derivaatalle:  
 $\dot{y}(t) = y_2(t)$
- Saadaan  $\dot{y}_2(t) = \frac{3}{s} y_2(t) - \frac{2}{s^2} y(t) + u(t)$ . (lisätty yksiköt)
- Eulerin menetelmällä saadaan päivitykset
- $t_1 = t_0 + \Delta t = 0s + 0.1s = 0.1s$
- $y(t_1 = 0.1s) \approx y(t_0 = 0s) + y_2(t_0)(t_1 - t_0) = 0m + 1m/s(0.1s - 0s) = 0.1m$
- $\dot{y}(t_1) = y_2(t_1) \approx y_2(t_0) + \dot{y}_2(t_0)(t_1 - t_0)$   
 $= y_2(t_0) + (3y_2(t_0) - 2y(t_0) + u(t_0))(t_1 - t_0)$   
 $= \frac{0m}{s} + \left( \frac{3}{s} \left( \frac{1m}{s} \right) - \frac{2}{s^2} (0m) + u(t_0) \right) (0.1s - 0s) = \frac{0.3m}{s} + u(t)(0.1s)$



# Mallin kalibrointi ja oppiminen näytteistä

# Yleisiä virhelähteitä mittauksissa

- *Kohina*: voi johtua esim. lämpökohinasta anturin elektroniikassa
- Ei tunneta mittaussmallia  $y = f(x)$
- Vakiovirhe  $a$  (bias):  $y = f(x) + a$
- Saturaatio: mittaus saavuttaa minimi- tai maximirajan
- Ajautuminen (drift): mitattu vaste muuttuu ajan ylin riippumatta todellisesta vasteesta. Voi johtua jostakin havaitsemattomasta fysikaalisesta muutoksesta anturissa
- Anturin tila vaikuttaa mittaukseen
- Kvantiointivirheet kun muunnetaan analoginen signaali digitaaliseksi

# Käytännön mittaukset

- Mitä jos mitataankin vain sijainti  $x(t)$ , mutta ei nopeutta  $\dot{x}(t)$  tai kiihtyvyyttä  $\ddot{x}(t)$  erillisillä antureilla?

- $\dot{x}(t)$  on sijainnin derivaatta. Derivaatta on

$$\dot{x}(t) = \lim_{\Delta t \rightarrow 0} \left( \frac{x(t+\Delta t) - x(t)}{\Delta t} \right)$$

- Approksimoidaan derivaattaa käyttämällä

$$\dot{x}(t) \approx \left( \frac{x(t_{n+1}) - x(t_n)}{t_{n+1} - t_n} \right)$$

- Approksimaatio on tarkka jos  $t_{n+1} - t_n$  on tarpeeksi pieni

- Samalla tavalla voidaan laskea kiihtyvyys

$$\ddot{x}(t) \approx \left( \frac{\dot{x}(t_{n+1}) - \dot{x}(t_n)}{t_{n+1} - t_n} \right)$$

# Mallin parametrien estimointi näytteistä: esimerkkinä jousi-massa-vaimennin

- Mitä jos ei tiedetä jousivakiota  $k$ , eikä vaimennuskerrointa  $b$ , eikä massaa  $m$ ? Jousi-massa-vaimennin:  $m\ddot{x} + b\dot{x} + kx = r$
- Mitataan näyte  $N$ :llä ajanhetkellä:  $D = (x(t_1), x(t_2), \dots, x(t_N), \dot{x}(t_1), \dot{x}(t_2), \dots, \dot{x}(t_N), \ddot{x}(t_1), \ddot{x}(t_2), \dots, \ddot{x}(t_N))$
- Käytetään tuntemattomille parametreille vektoria  $\theta = (m \ b \ k)$  .

Saadaan

$$m\ddot{x}(t_0) + b\dot{x}(t_0) + kx(t_0) = r(t_0) \Rightarrow \\ (\ddot{x}(t_0) \ \dot{x}(t_0) \ x(t_0))\theta^T = r(t_0)$$

Monella näytteellä saadaan vastaavasti

$$\begin{aligned} (\ddot{x}(t_1) \ \dot{x}(t_1) \ x(t_1))\theta^T &= r(t_1) \\ (\ddot{x}(t_2) \ \dot{x}(t_2) \ x(t_2))\theta^T &= r(t_2) \\ &\dots = \dots \\ (\ddot{x}(t_N) \ \dot{x}(t_N) \ x(t_N))\theta^T &= r(t_N) \end{aligned}$$

# Mallin parametrien estimointi näytteistä: jousi-massa-vaimennin jatkuu

- Kirjoitetaan matriisimuotoon:

$$X\theta^T = R$$

missä  $X$  on  $N \times 3$  matriisi, jossa on  $(\ddot{x}(t) \dot{x}(t) x(t))$  näyte jokaisella rivillä ja  $R$  on  $N \times 1$  matriisi (vektori), jossa on  $r(t)$  arvot

- Jos yritetään ratkaista  $\theta$  minimoimalla pienin neliövirhe ("minimize mean squared error") saadaan

$$\theta^T = (X^T X)^{-1} X^T R$$

- Yleisesti, *lineaarisen* systeemin  $x\theta^T = r$  parametrit  $\theta$  voidaan yrittää ratkaista keräämällä näytteet matriiseihin  $X$  ja  $R$  ja laskemalla  $\theta^T = (X^T X)^{-1} X^T R$

# Mallin oppiminen



Hukkafunktio (“loss function”)

- Mitataan  $N$  näytettä  $D = (y(t_1), y(t_2), \dots, y(t_N))$
- Sovitetaan malli  $f(x, \theta)$  näytteisiin
- Pienimmän neliövirheen sovitus valitsee parametrit  $\theta$ , jotka minimoivat virheen

$$L(\theta) = \sum_n (f(x_n, \theta) - y(t_n))^2$$

- Muitakin virheitä voidaan minimoida, esim.  
 $\sum_n |f(x_n, \theta) - y(t_n)|$   
mutta neliövirhe yleisin

# Lineaarinen ja epälineaarinen malli

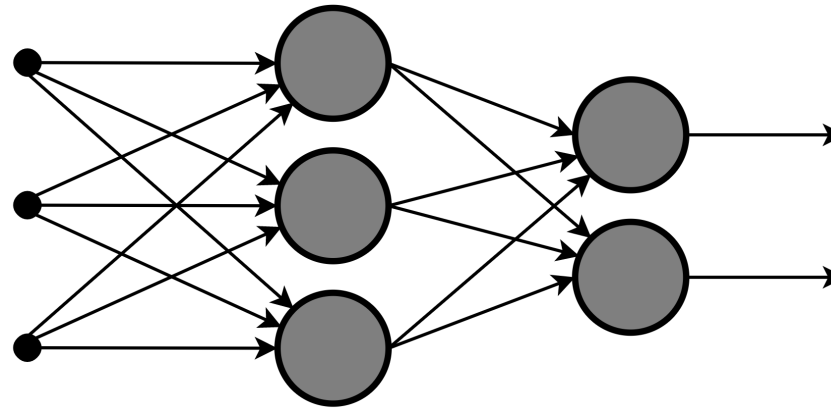
- Lineaarisisä mallissa  $f(x, \theta) = \sum_i \theta_i \phi_i(x)$  sovitetaan lineaariset parametrit  $\theta$ . Funktiot  $\phi_i(x)$  voivat olla mitä tahansa funktioita

- Epälineaarisisä mallissa optimoitavat parametrit vaikuttavat epälineaarisisä. Esim. mallissa

$$f(x, \theta) = f(x, \hat{\theta}, \mu_1, \dots, \mu_M, \Sigma_1, \dots, \Sigma_M) = \sum_i \hat{\theta}_i e^{-\sigma_i(x-\mu_i)^T(x-\mu_i)}$$

malli on epälineaarinen *suhteessa parametreihin*  $\mu_i$  ja  $\sigma_i$

# Neuroverkko



- Vahvuudet:
  - Pystyy mallintamaan monimutkaisia prosesseja
  - Pystyy periaatteessa extrapoloimaan
  - Tehokas python koodi erilaisten neuroverkkojen opettamiseen on valmiina
- Heikkoudet:
  - Neuroverkon toimintaa hankala ymmärtää. Mitä neuroverkko antaa vasteeksi tietylle syötteelle?
  - Opetus voi vaatia paljon näytteitä



# Mallin sovitus

- Minimoidaan neliövirhe  $L(\theta) = \sum_n (f(x_n, \theta) - y(t_n))^2$
- Gradienttimenetelmä seuraa gradienttia  $\frac{\delta L(\theta)}{\delta \theta}$
- Valitaan  $\theta_0$ , vektori joka sisältää parametrialkuarvot
- Gradienttimenetelmässä parametrit päivitetään kaavalla

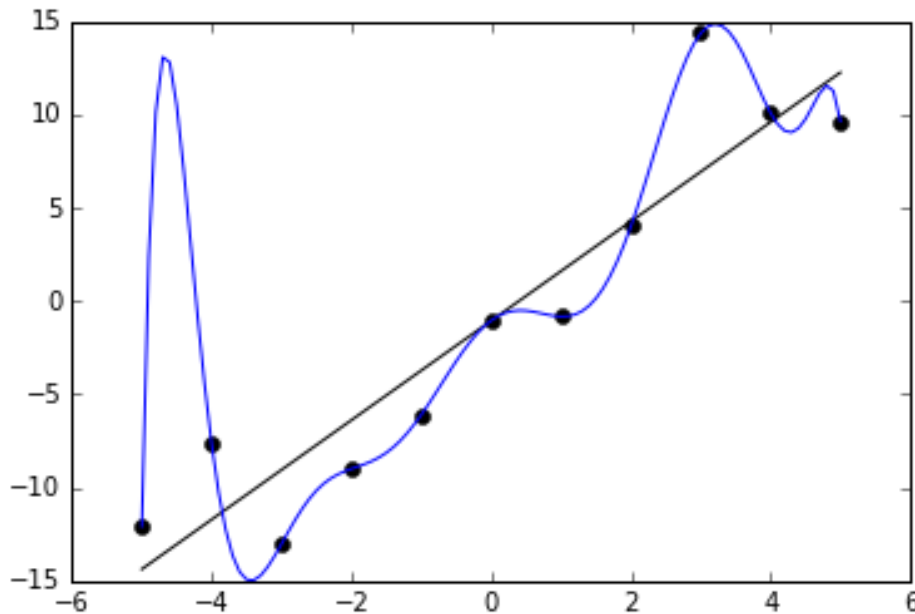
$$\theta^{k+1} = \theta^k - \alpha \frac{\delta L(\theta)}{\delta \theta}$$

- Kun  $f(x, \theta)$  on lineaarinen, löydetään globaali minimi hukkafunktiolle  $L(\theta)$



# Epälineaarinen malli: mallin sovittaminen

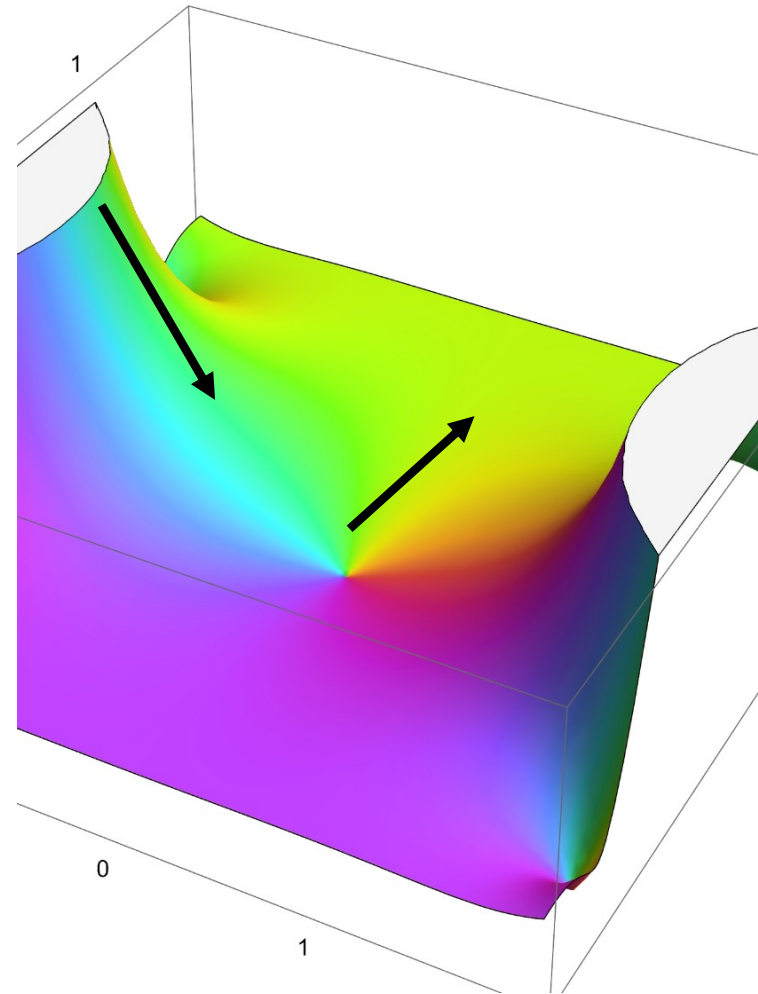
1. Lokaaleja minimejä on useampi
2. Malli voi ylisovittaa ("overfit") näytteisiin



Tällä kurssilla käytetään valmiita python-funktioita mallin sovittamiseen, mutta periaatteet hyvä ymmärtää.

# Epälineaarinen malli käytännössä

- Valitaan malli, joka pystyy kuvaamaan prosessia riittävällä tarkkuudella
- Kerätään tarpeeksi näytteitä
- Käytetään valmista python funktiota sovittamaan mallin parametrit näytteisiin



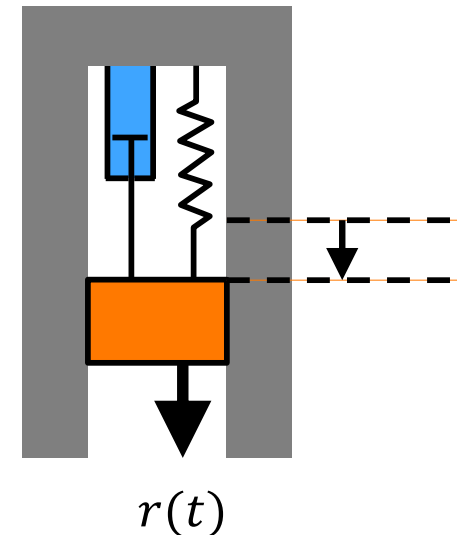
# Jousi-massa-vaimennin: epätarkka malli

- Jousi-massa-vaimentimelle on valmis malli

$$m\ddot{x}(t_n) + b\dot{x}(t_n) + kx(t_n) = r(t_n)$$

$$f(x, \theta) = m\ddot{x}(t_n) + b\dot{x}(t_n) + kx(t_n)$$

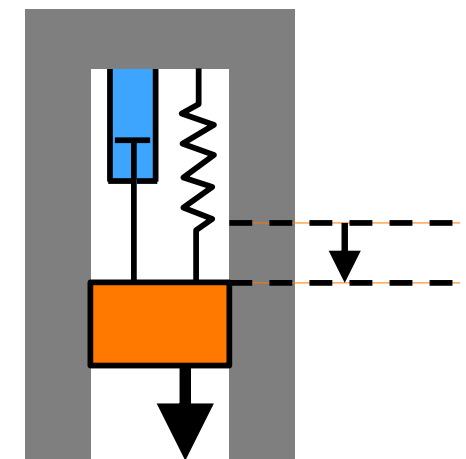
- Mutta entä kitka ja muut tuntemattomat asiat mitkä voivat vaikuttaa prosessiin?
- Jos  $f(x, \theta)$  ei ole riittävän tarkka, pitäisikö oppia yleinen malli näytteistä valmiin mallin tilalle?
- Koko prosessin mallintaminen voi vaatia paljon näytteitä



# Tuntemattoman ja valmiin mallin yhdistäminen

- Opitaan yleinen malli  $g(x, \theta_g)$  vain tuntemattomalle osalle: vaatii vain vähän näytteitä
- Kokonaismalli on  $\hat{f}(x, \theta) = f(x, \theta_f) + g(x, \theta_g)$ , missä  $\theta$  koostuu valmiin mallin parametreista  $\theta_f$  ja yleisen mallin parametreista  $\theta_g$
- $\theta_g$  löydetään minimoimalla neliövirhe
$$L(\theta_g) = \sum_n \left( f(x_n, \theta_f) + g(x_n, \theta_g) - y(t_n) \right)^2$$
- Minimointiin voidaan käyttää gradienttimenetelmää

Systemi esimerkiksi jousi-massa-vaimennin, jossa tunnettu osuus koostuu jousesta, massasta, ja vaimentimesta ja tuntematon osuus kitkasta



$$y(t) = r(t)$$

$\theta_f$  pidetään vakiona

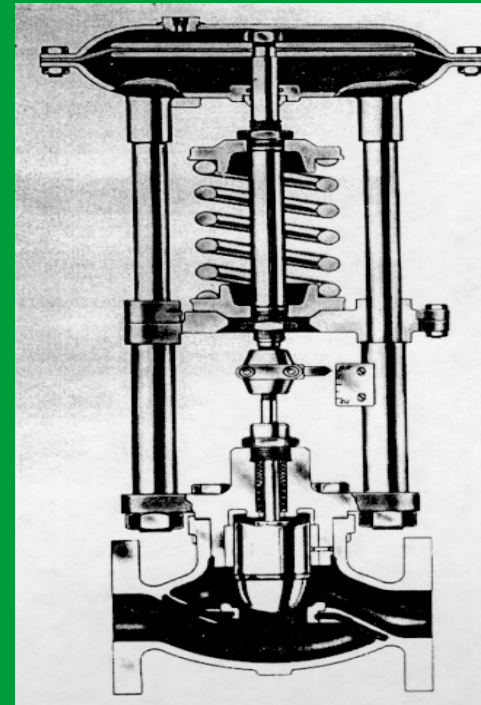
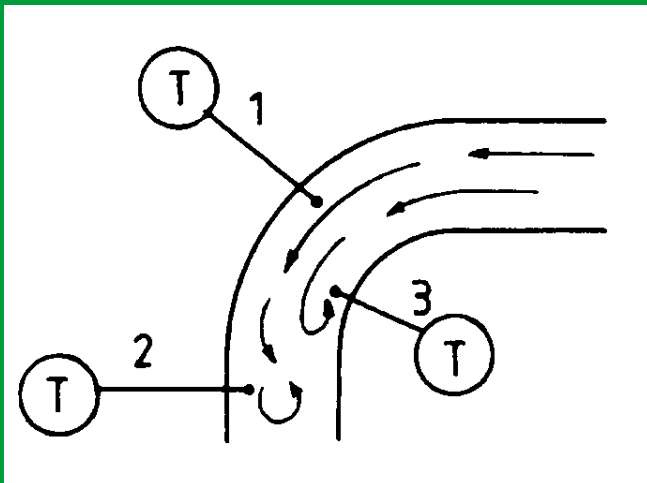
## Kertaustehtävä 2

- Systemille on annettu dynamiikka malli  $r(t) = ay^2(t) - abu(t)$ , missä  $a$  ja  $b$  ovat mallin määrittäviä muuttujia
- Onko malli lineaarinen vai epälineaarinen?
- On kerätty näytteitä muuttujille  $r(t)$ ,  $y(t)$ , ja  $u(t)$  eri ajanhetkillä  $t$ . Johda päivityskaava vakioille  $a$  ja  $b$  olettaen että minimoidaan neliövirhettä

## Kertaustehtävä 2 - malli

- Malli on epälineaarinen, koska siinä on tulo  $ab$
- Pienimmän neliövirheen minimointi perustuu  $L(a, b) = \sum_t (ay^2(t) - abu(t) - r(t))^2$  minimointiin
- Lasketaan osittaisderivaatat  $a$ :lle ja  $b$ :lle
- $\frac{\delta L(a,b)}{\delta a} = \sum_t 2(y^2(t) - bu(t))(ay^2(t) - abu(t) - r(t))$
- $\frac{\delta L(a,b)}{\delta b} = \sum_t 2(-au(t))(ay^2(t) - abu(t) - r(t))$
- Gradienttimenetelmässä saadaan päivityksiksi
- $a^{k+1} = a^k - \alpha \frac{\delta L(a,b)}{\delta a}$  ja  $b^{k+1} = b^k - \alpha \frac{\delta L(a,b)}{\delta b}$
- Lyhyemmin merkittävä  $\theta^{k+1} = \theta^k - \alpha \frac{\delta L(\theta)}{\delta \theta}$ ,  $\theta = (a, b)$

# Anturit ja toimilaitteet





# Anturit, eri anturityyppejä

- Lämpötilan mittaus
  - Vastusanturi
  - Termoelementti
  - Pyrometri ja infrapuna-anturi
  - Kuituanturi
- Virtausmittaus
  - Kuristuslaippa-anturi ja venturiputki
  - Myös rotametri, turbiini, magneettinen virtausmittaus
  - Vähemmän käytettyjä: vortex, äänikaikumittaus

# Anturit, eri anturityyppejä

- Paineen mittaus
  - Kalvorasia
  - paljeputki
  - painekaari (Bourdon-putki)
  - MEMS- ja MST-anturit
- Pinnankorkeuden mittaus
  - uimurityyppiset mittaukset
  - paine-eroon palautuvat mittaukset
  - kapasitiiviset mittaukset
  - vastuslankamittaukset
  - tutkaperiaatteella toimivat mittaukset
  - äänikaiku
  - valopulssin kulku aika
  - säteilymittaukset

# Anturit, eri anturityyppejä

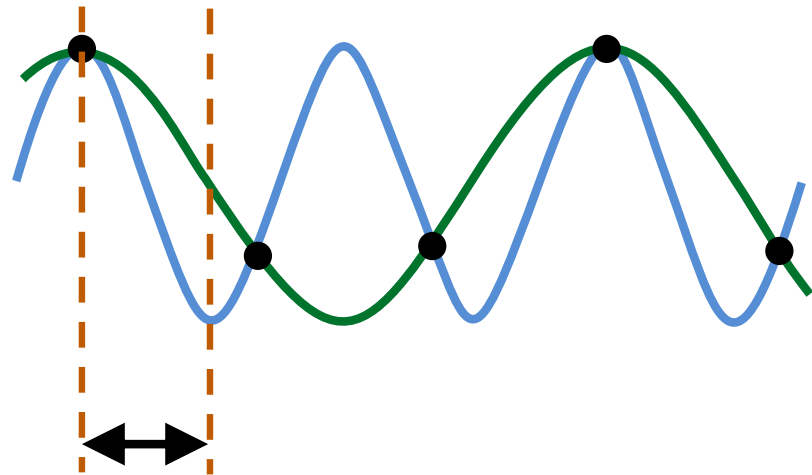
- Asento, asema, pyörimisnopeus
  - Potentiometrit
  - Kosketinanturit
  - Venymäliuskat
  - Puolijohdinsauvat
  - Kapasitiiviset anturit
  - Induktiiviset anturit
- Optiset mittaukset
  - Valokytkimet
  - Optiset enkooderit kulmamittauksissa
  - Laser käyttö etäisyyden mittauksessa
  - Laserin ja kameran yhdistelmät paikannuksissa ja muotojen mittauksissa

# Anturit, eri anturityyppejä

- Kiertokulma
  - Optinen enkooderi
  - Kiertopotentiometri
  - Resolveri
- Lineaariliike
  - Magnetostriktiivinen anturi
  - Vaijerianturi
  - Magneettinauha-anturit
- Nopeus
  - Takogeneraattori
  - Pulssilaskija
  - Optiset menetelmät
  - Doppler-tutka

# Antureiden dynaamiset ominaisuudet

- Näytteenotto signaalista
  - Jatkuvasta signaalista muodostetaan numerojono
  - Liian alhainen näytteenottoväli aiheuttaa ongelmia
  - Maksimi näytteenottoväli = puolet korkeataajuisimman signaalin aallonpituudesta
  - Korkeammat taajuudet laskostuvat alemmille
  - Laskostumisen estäminen:
    - Riittävän korkea näytteenottotaajuus (väh. 2x signaalin korkein taajuus)
    - Signaalin alipäästösuodattaminen ennen näytteenottoa

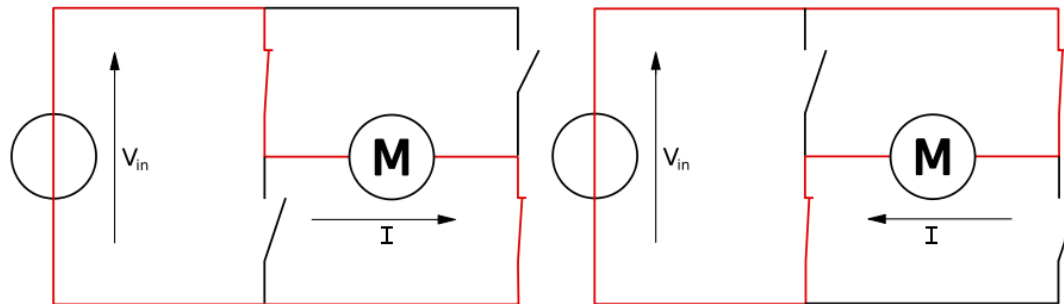


# Toimilaitteet, toimilaitteiden dynaamiset ominaisuudet

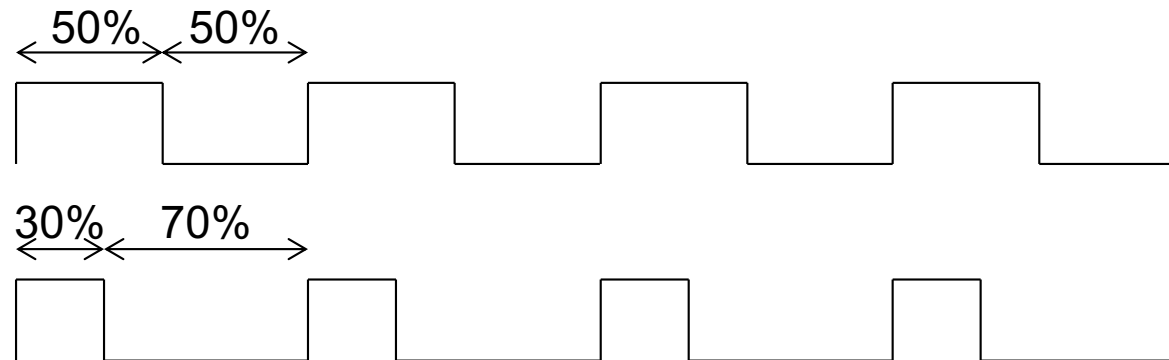
- Prosessiventtiilit
  - Tyyppejä
    - Yksi- ja kaksi-istukkaventtiilit
    - Kaksi- ja kolmitieventtiilit
    - Lämpäventtiilit
    - Palloventtiilit
  - Venttiilin läpivirtaus riippuu paineesta
  - Venttiiliä ohjaavalla moottorilla dynamiikka
    - Kuvataan siirtofunktiolla
- Hydrauliikkaventtiilit
  - Ohjaa hydraulisia sylintereitä
  - Koko systeemin dynamiikalla siirtofunktio

# Toimilaitteet, toimilaitteiden dynaamiset ominaisuudet

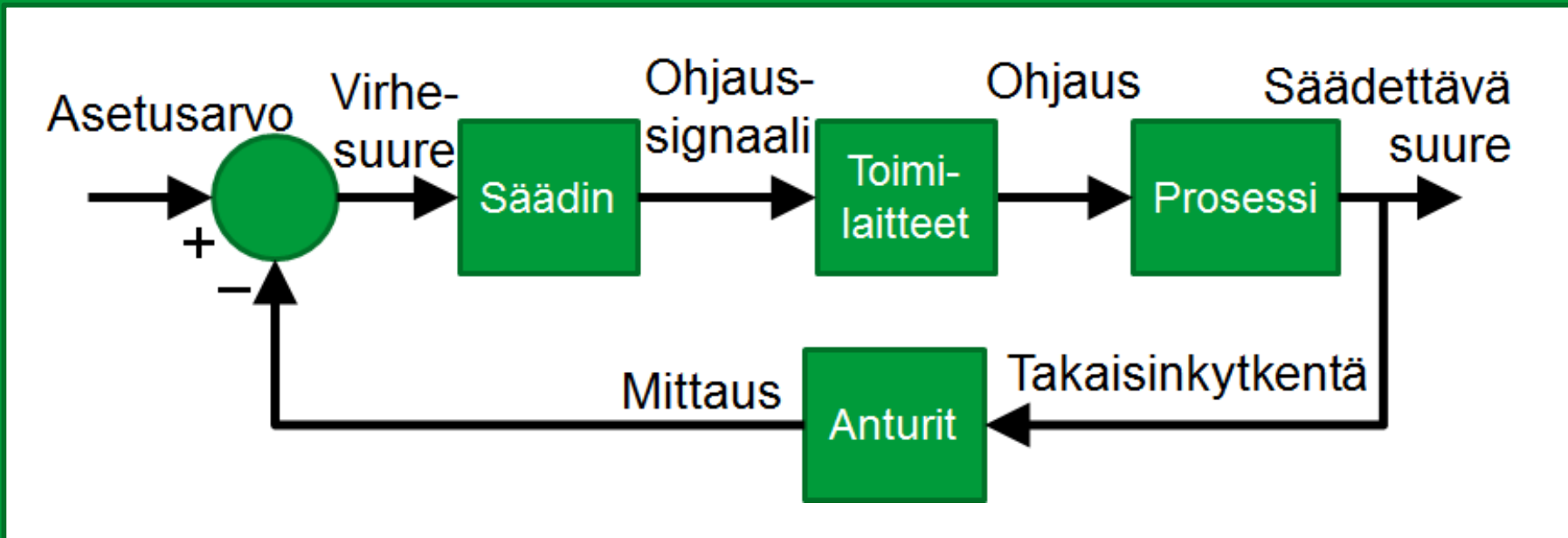
- Sähkökäytöt (moottoriohjaimet)
  - H-siltakytkennällä ohjataan moottorin pyörimissuuntaa



- Jännitteen pulssileveysmodulaatiolla ohjataan nopeutta

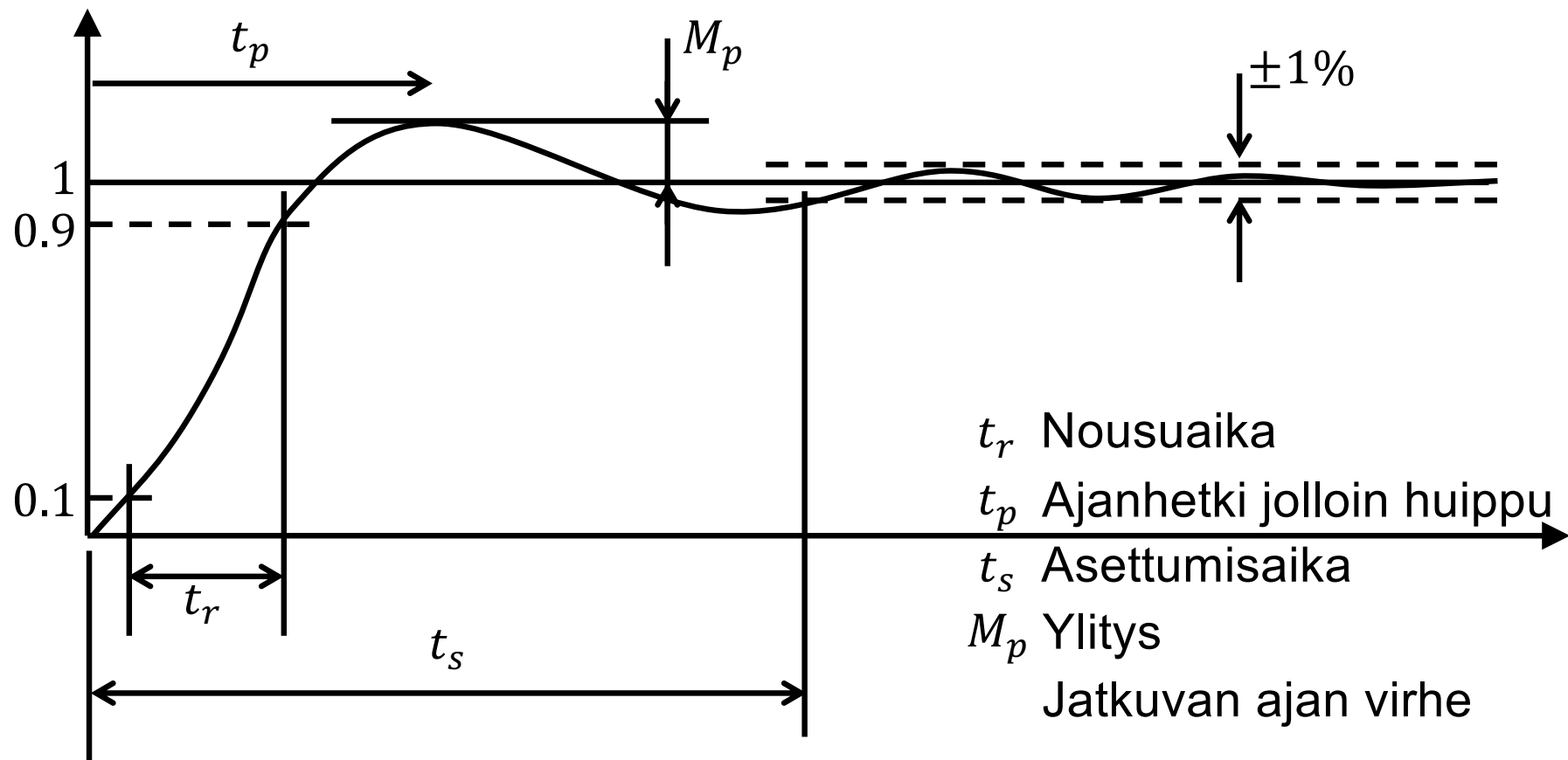


# Takaisinkytketty säätö

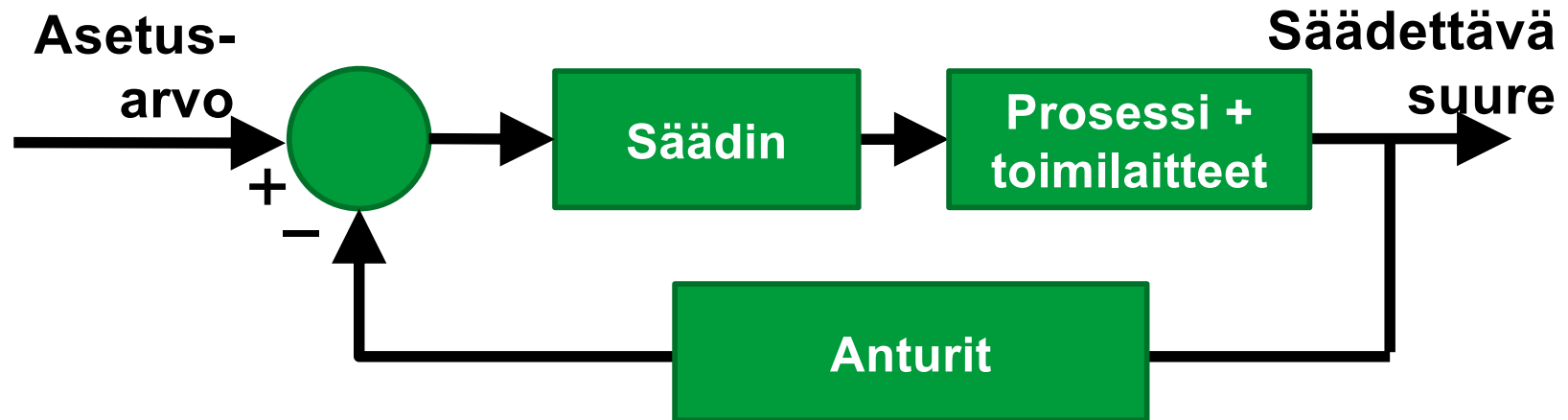




# Dynaamiset ominaisuudet aikatasossa



# Takaisinkytkentä ja säätö yleisesti

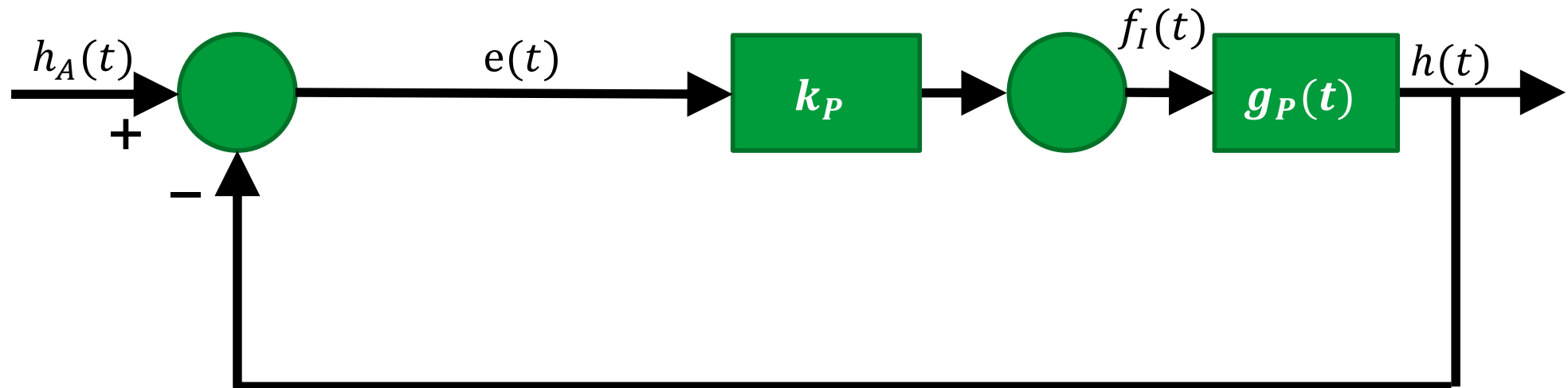


- Asetusarvosäätö
  - Asetusarvo ei muutu, säätimen tehtävä on kompensoida häiriöt
  - Esim. kemianteollisuuden valmistusprosessit
- Servosäätö
  - Prosessin halutaan seuraavan asetuseron muutoksia tarkasti
  - Esim. moottorilla tehtävä asennon säätö

# P/PI/PD/PID -säätimet

- P-säädin ohjaa prosessia skaalaamalla asetusravon ja mittauksen erotusta vakio kertoimella  $k_P$ :

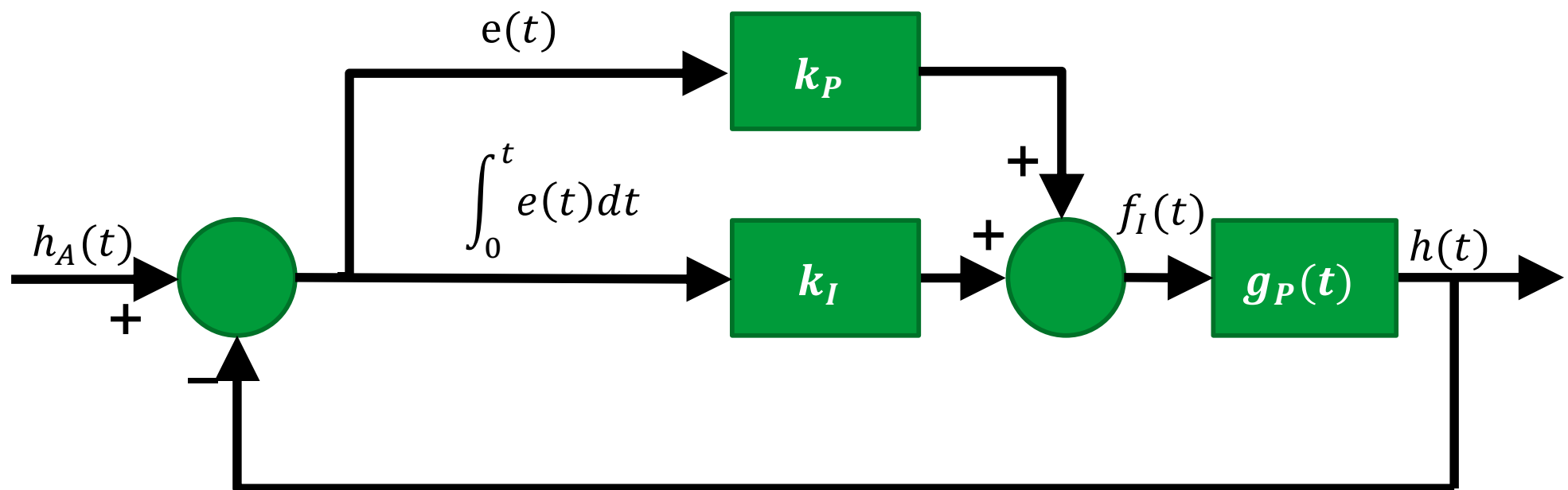
$$f_I(t) = k_P e(t), \text{ missä } e(t) = h_A(t) - h(t)$$



# P/PI/PD/PID -säätimet

- PI-säätimessä P-säätimen rinnalla integraattori
  - Poistaa pysyvän poikkeaman

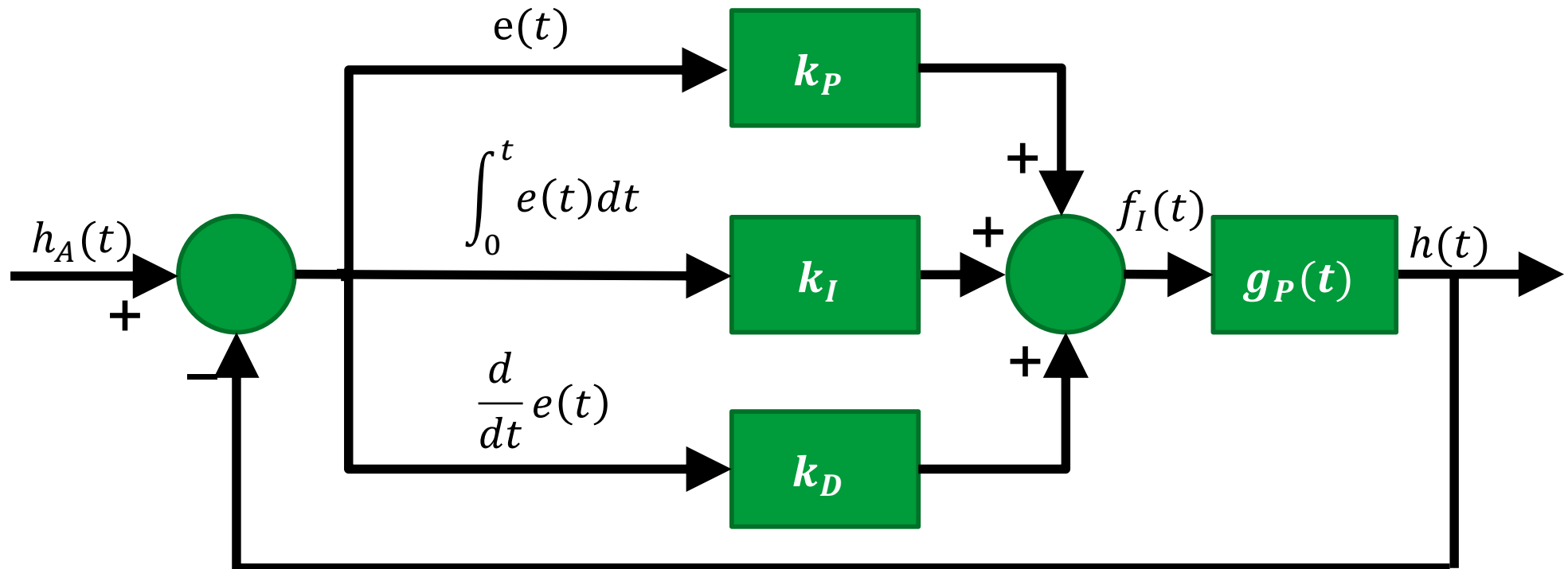
$$f_I(t) = k_P e(t) + k_I \int_0^t e(t) dt$$



# P/PI/PD/PID -säätimet

- PID-säätimessä P- ja I-säätimien lisäksi derivaattori
  - Rauhoittaa nopeita muutoksia

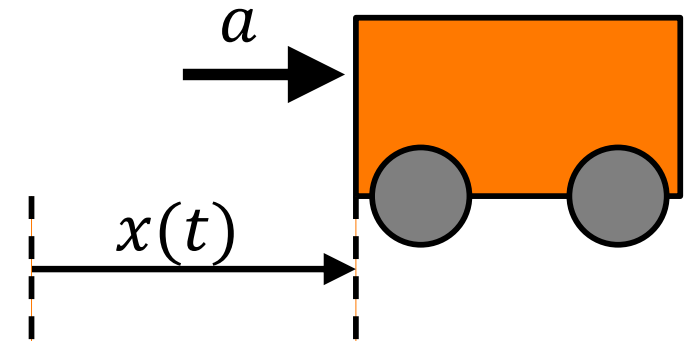
$$f_I(t) = k_P e(t) + k_I \int_0^t e(t) dt + k_D \frac{d}{dt} e(t)$$



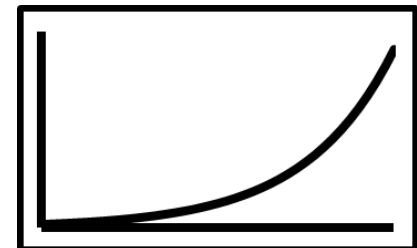
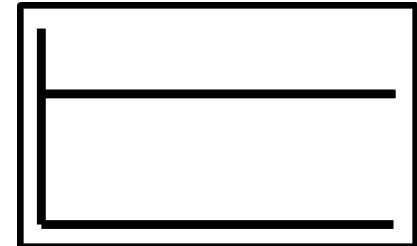
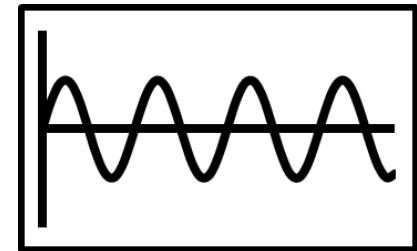
# P/PI/PD/PID -säätimet

- Integroiva prosessi
  - Jos prosessissa itsessään integraattori (esim. täyttyvä säiliö)
- Voidaan käyttää PD-säädintä
  - PID-säädin, jossa I-osan vahvistus  $\rightarrow 0$

# Stabiilius, värähtely: P-säädin kärriesimerkissä

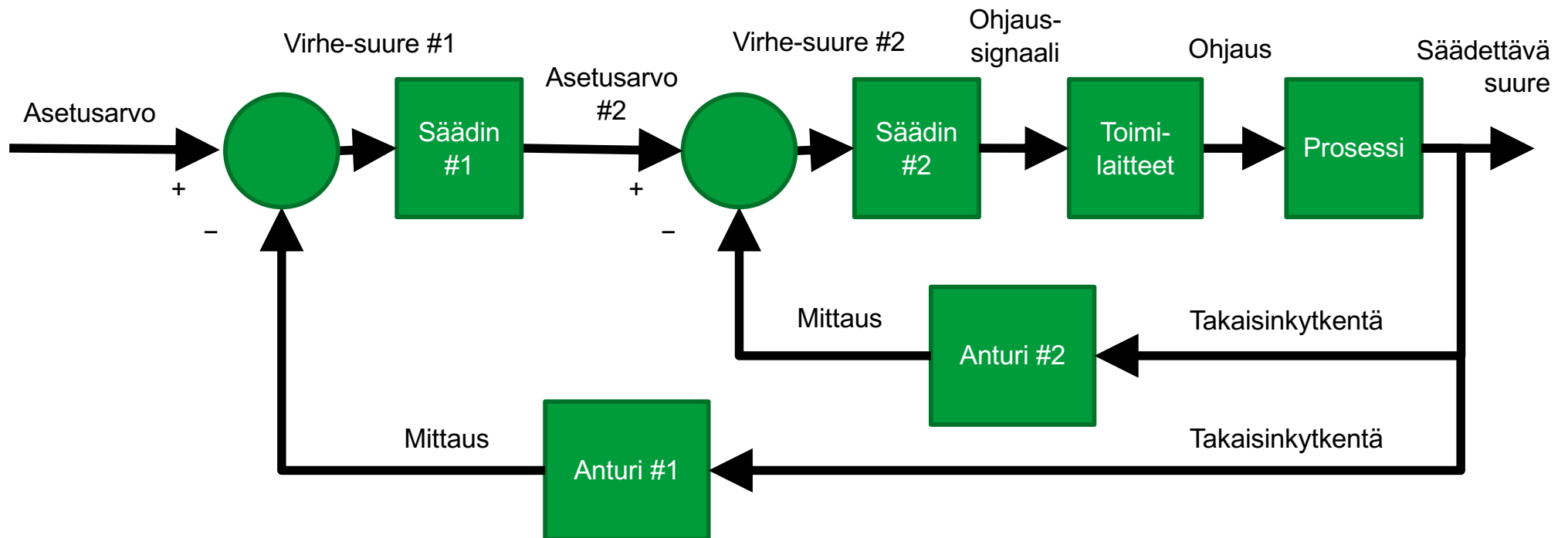


- Kun lähtönopeus  $\dot{x}(0) = 0$ ,  
saadaan  $x(t) = x(0) \cos(\sqrt{k_P}t)$
  - Kun  $k_P > 0$ ,  $x(t)$  **värähtelee / oskilloi**
  - Kun  $k_P = 0$ ,  $x(t)$  on **vakio**
  - Kun  $k_P < 0$ ,  $x(t) =$   
 $x(0) \cos(t\sqrt{-k_P}i) = \frac{x(0)}{2} \left( e^{-\sqrt{-k_P}t} + e^{\sqrt{-k_P}t} \right)$   
→  $x(t)$  kasvaa eksponentiaalisesti
- Systemi on **epästabiili**.



# Kaskadisäätö – systeemikaavio

- Kaskadisäätö: takaisinkytketty säätö sarjassa
- Yleensä ”sisempi” säätösilmukka toimii korkeammalla taajuudella kuin ”ulompi”
- Tässä kaksi säätösilmukkaa, mutta voi olla useampia





# Miksi kaskadisäätö?

- Voidaan testata ja kehittää osia erillisinä:
  - Kun kohteena on säiliön pinnankorkeus, voidaan kehittää ja testata lähtövirtauksen ja pinnankorkeuden säätö erikseen
  - Sähkömoottoreilla voi olla omat sisäiset säätimensä: annetaan asetusarvona esimerkiksi pyörimisnopeus
- Voidaan käyttää ”sisempää” säädintä nopealla korkeammalla taajuudella kuin ”ulompaa”. Usein suositellaan 3x tai 5x nopeampaa sisäistä säädintä
- Sisempi säätösilmukka voi usein olla aggressiivisempi ja pelkkä P-säädin voi riittää, jos ulompi säädin on PID-säädin, joka poistaa poikkeamat ja kohinan

# Kertaustehtävä 3

- P-säädetyin järjestelmän differentiaaliyhtälö on  $\dot{g}(t) = 2 + 3 u(t)$ , asetusarvo on  $g_A(t) = 0$
- Millä arvoilla järjestelmä värähtelee tai on epästabiili?
- Onko järjestelmässä pysyvää poikkeamaa ja jos on mikä se on?

# Kertaustehtävä 3 - malli

- Differentiaaliyhtälö on

$$\dot{g}(t) = 2 + 3u(t) = 2 + 3k_p(g_A(t) - g(t)) = 2 - 3k_p g(t)$$

- Koska meillä on ensimmäisen asteen differentiaaliyhtälö saadaan

- $g(t) = c_1 e^{-3k_p t} + c_2$  ja  $\dot{g}(t) = -3k_p c_1 e^{-3k_p t}$

- Asetetaan  $g(t)$  ja  $\dot{g}(t)$  differentiaaliyhtälöön ja saadaan

- $-3k_p c_1 e^{-k_p t} = 2 - 3k_p c_1 e^{-k_p t} - 3k_p c_2 \Rightarrow c_2 = 2/(3k_p)$

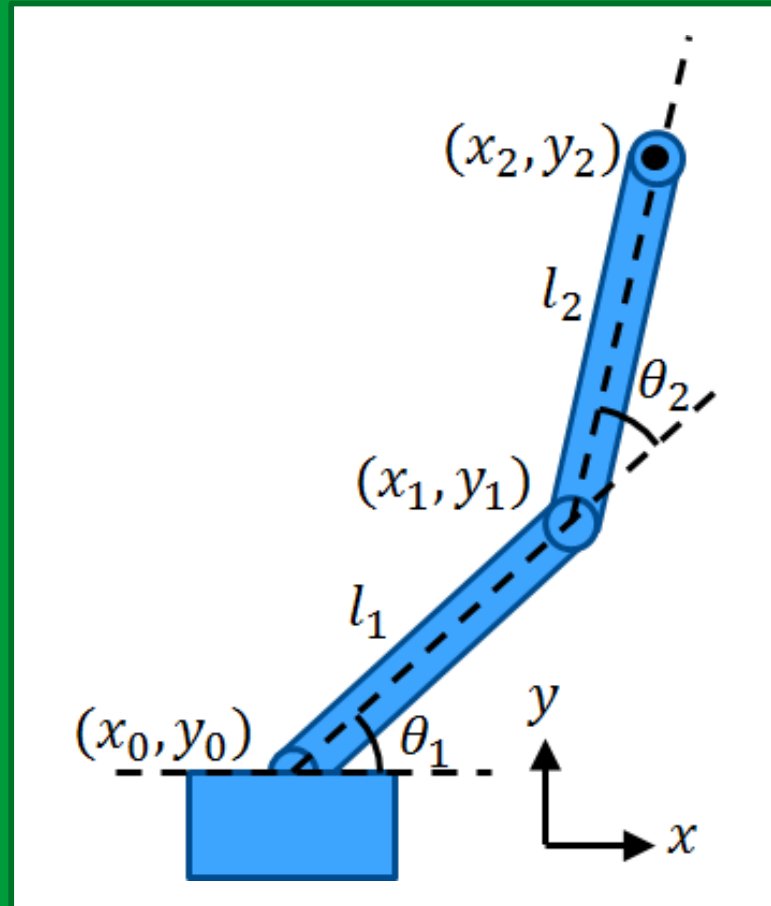
- $c_1$  saadaan  $g(t=0)$  :n avulla:

- $g(0) = c_1 e^0 + 2/(3k_p) \Rightarrow c_1 = g(0) - 2/(3k_p)$  ja saadaan

- $g(t) = (g(0) - \frac{2}{3k_p})e^{-3k_p t} + \frac{2}{3k_p}$

- Järjestelmä on stabiili kun  $k_p > 0$  , jolloin eksponenttifunktion eksponentti on negatiivinen. Värähtelyä ei ole. Systemissä on pysyvä poikkeama  $\frac{2}{3k_p}$ .

# Robottiikka ja kinematiikka

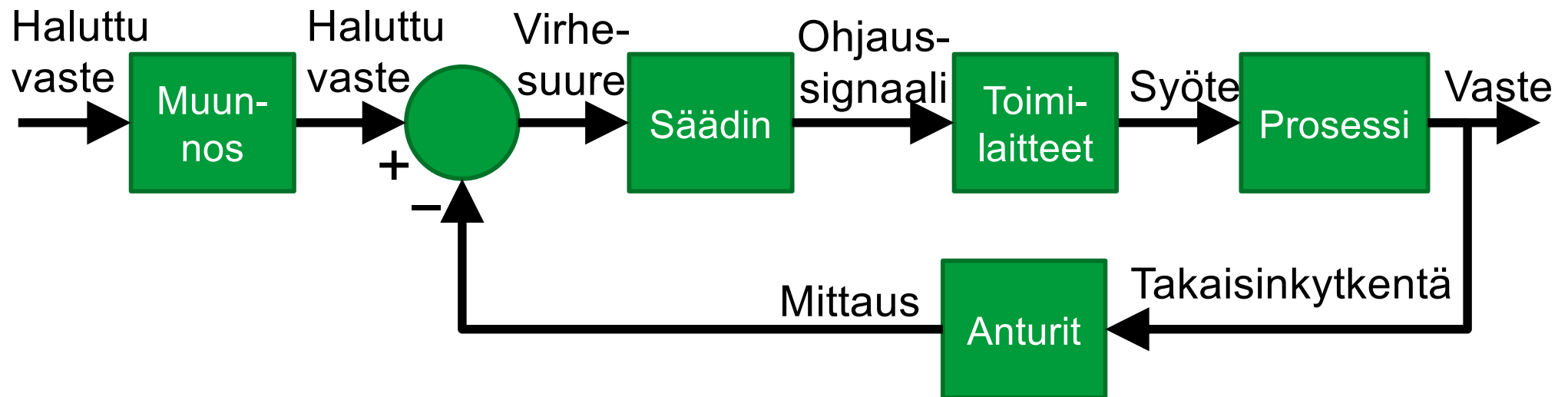


# Roboteista yleisesti

- Robotti on vuorovaikutuksessa ympäristön kanssa liikuttamalla ja aistimalla
- Robottikäsivarret yleisiä teollisuudessa
- Siirtymät koordinaattijärjestelmien välillä
  - Robotin työkalupisteen paikka ja asento
- Kinematiikka
  - Liikkeiden laskenta ilman vaikuttavien voimien käsittelyä
- Vapausasteet = itsenäisten paikkamuuttujien määrä
  - Yleensä nivelten määrä
- Robotin työalue: millä alueella työkalupiste voi sijaita

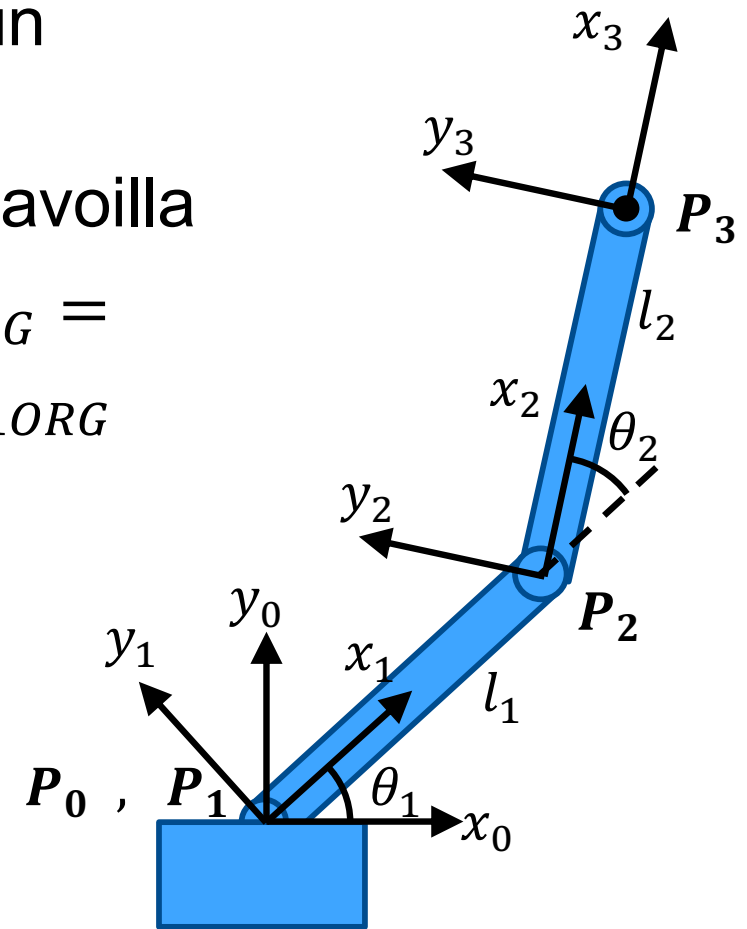
# Robotin kinematiikka

- Aina ei ohjata suoraan kiinnostavaa suuretta
- Esim. robotti:
  - Käytännön sovelluksissa kiinnostavat suuret: XYZ-koordinaatit
  - Säädettävät suuret: nivelten kulmat



# Robotin suora kinematiikka

- Työkalupisteen paikan laskenta, kun nivelten parametrit tiedetään
- Voidaan laskea trigonometrisilla kaavoilla
- Käyttämällä matriisiesitystä:  ${}^0P_{3ORG} = {}^0R_1 ({}^1R_2 {}^2P_{3ORG} + {}^1P_{2ORG}) + {}^0P_{1ORG}$ 
  - $P$  on siirtymä koordinaatistojen välillä
  - $R$  on rotaatiomatriisi
- Homogeenisilla koordinaateilla:  
 ${}^0T_3 = {}^0T_1 {}^1T_2 {}^2T_3$ 
  - $T$  sisältää siirtymän ja rotaation



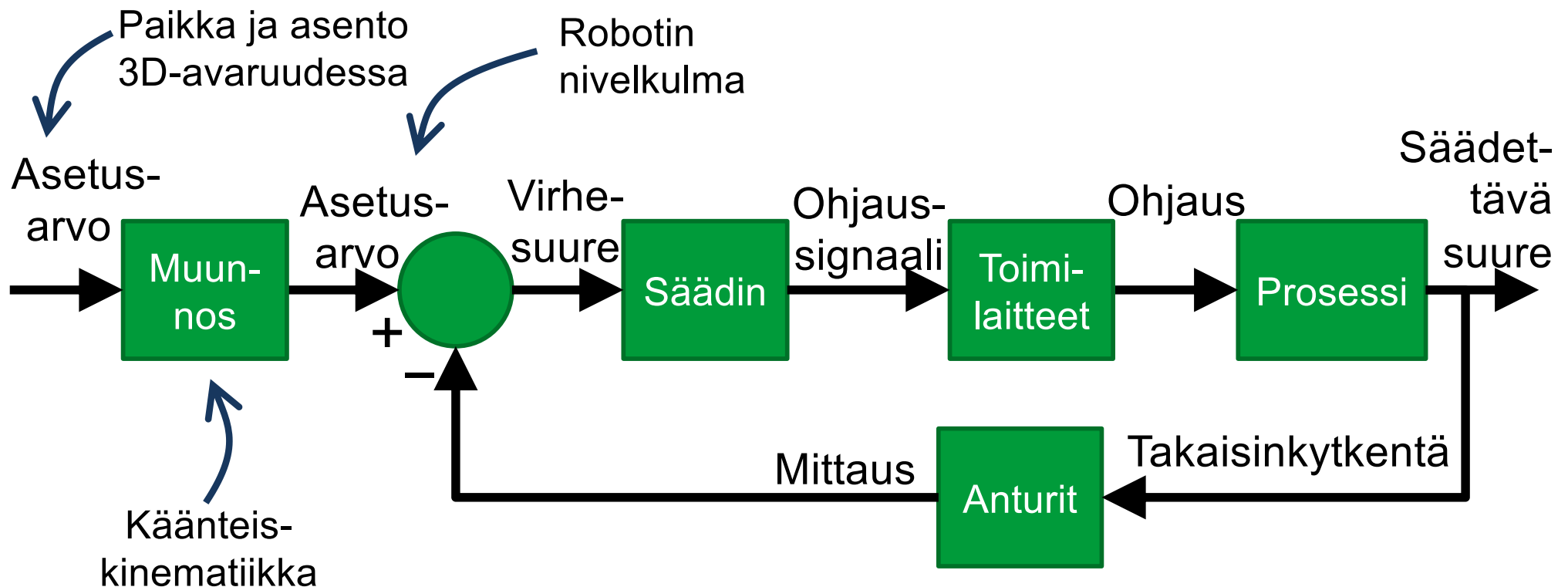
# Robotin käänteinen kinematiikka

- Määrittää robotin kulmat, kun työkalupiste halutaan saada tiettyyn paikkaan
- Hankalampaa kuin suoran kinematiikan laskenta
  - Ratkaisua ei välttämättä löydy suljetussa muodossa
- Ratkaisuja voi olla enemmän kuin 1
- Tyypillisesti ratkaistaan geometrinen ongelma
  - Kolmiolaskentaa



# Robotin käänteinen kinematiikka

- Robotin kinemaattisen muunnoksen huomioiminen



# Robotin liikekinematiikka

- Ratkaisee robotin nivelkulmien muutokset
- Jacobin matriisin avulla saadaan laskettua muutos karteesisessa koordinaatistossa, kun tiedetään muutos nivelkulmissa

$$\dot{P} = J(\theta)\dot{\theta}$$

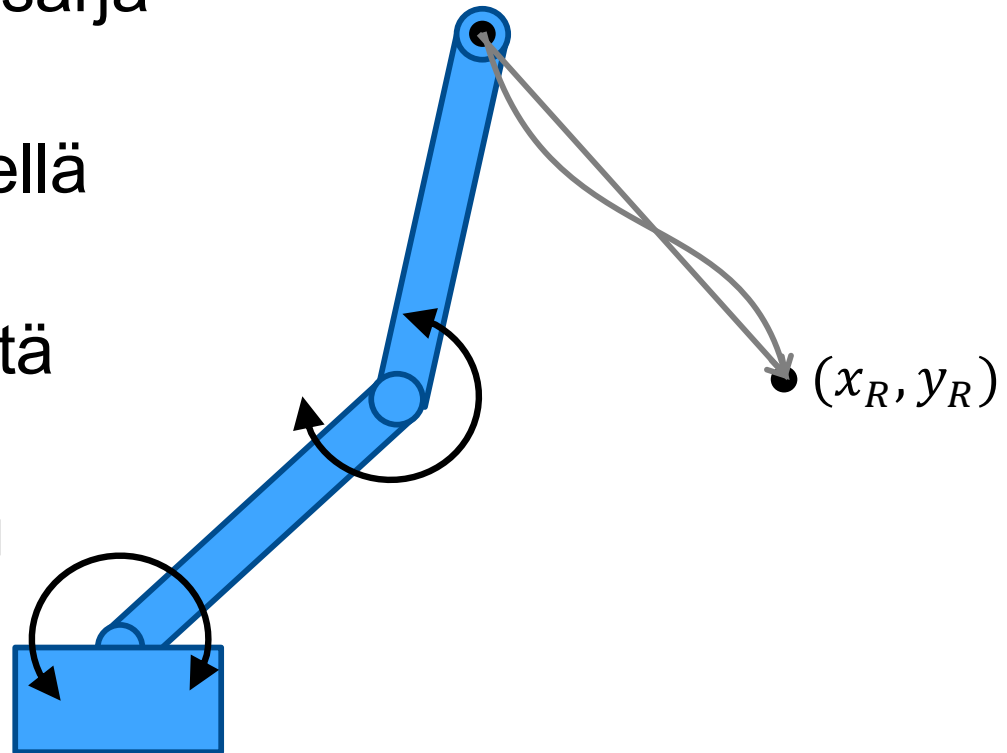
- Jacobin matriisi on suoran kinematiikan derivaattamatriisi
- Käänteisellä Jacobin matriisilla nivelnopeus, kun tiedetään, kuinka nopeasti työkalupiste liikkuu (X,Y,Z):ssa

$$\dot{\theta} = J^{-1}(\theta)\dot{P}$$

- Jos käänteismatriisia ei ole olemassa, nopeutta ei voi toteuttaa  
→ singulariteetti

# Robotin nivelkulmien säätö

- Robottiohjelma yleensä sarja peräkkäisiä paikkoja
- Halutut kulmat käänteisellä kinematiikalla
- Säädetään kutakin niveltä PD-säätimellä
- Jokainen nivel hakeutuu mahdollisimman nopeasti tavoiteasentoon

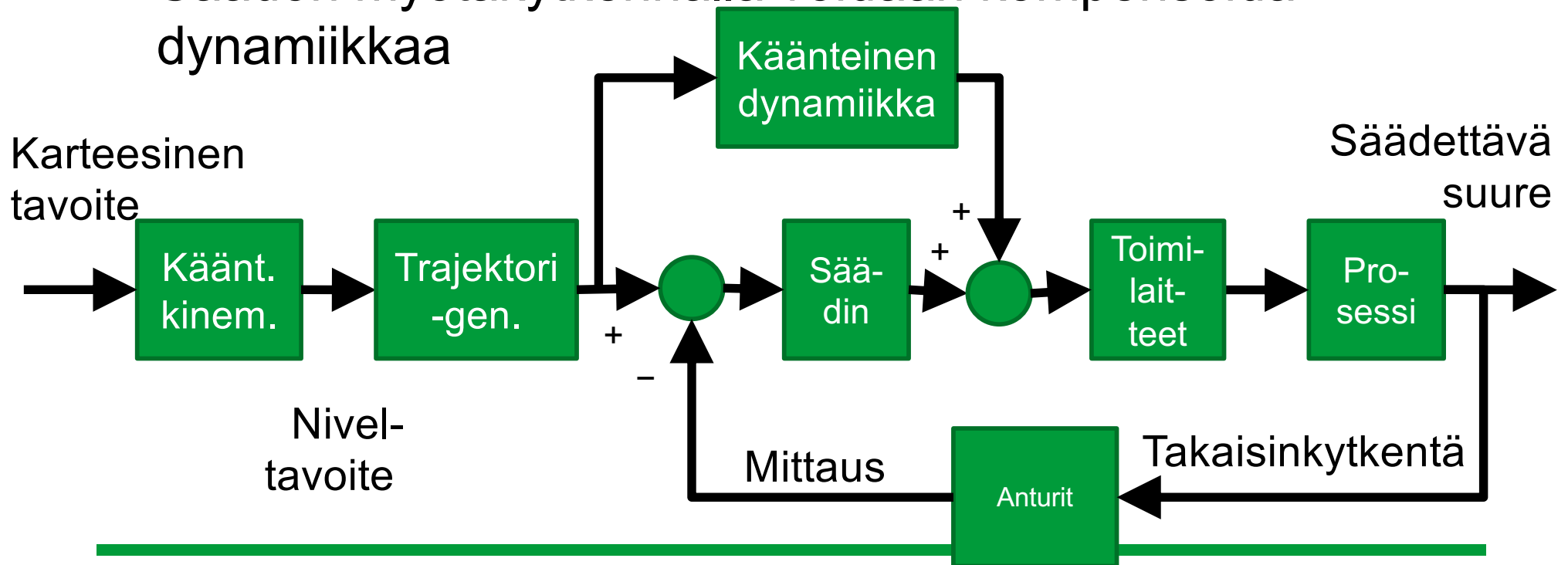


# Robotin liiketrajektorin säätö

- Trajektori määrittää liikkeen kahden pisteen välillä ajan funktiona
- Nopein liike saadaan ajamalla niveliä maksiminopeudella (*nivel jonka liikuttavana oleva matka on suurin määrää liikkeeseen kuluvan kokonaisajan -> kaikkien nivelten liike alkaa ja päättyy yhtä aikaa, eli kyseessä on nivelkoordinaatiston suhteen lineaarisesti interpoloitu liike* )
  - Työkalupisteen liikerata mielivaltainen
  - Vapaille liikkeille ok (ei lähellä esteitä)
- Suorakulmaisen koordinaatiston suhteen kontrolloitu liike: lineaariset trajektorit karteesisissa avaruudessa
  - Hitaampi, turvallisempi
  - Rajoitetuille liikkeille (esteiden lähellä, tarvitaan tarkkaa liikettä)
  - Reittipisteiden välillä ajo
    - Päivitetään esim. 100x / sekunti tai 10 millimetrin välein

# Robotin säädön myötäkytkentä, dynamiikan huomiointi

- Robotin dynamiikkaan vaikuttavat monet tekijä, mm. gravitaatio, hitausmomentti, yms.
- Säädön myötäkytkennällä voidaan kompensoida dynamiikkaa



# Kertaustehtävä 4

- Tasossa liikkuva manipulaattori sisältää kaksi niveltä, varsien pituus on 1m ja kulmat ovat  $\theta_1 = 2\pi$  ja  $\theta_2 = \frac{9\pi}{4}$
- Laske homogeenisillä muunnosmatriiseilla työkalupisteen sijainti
- Työkalupiste liikkuu ylöspäin 1m/s. Mitkä ovat nivelten kulmanopeudet?

## Kertaustehtävä 4 – malli 1/2

$$\bullet T_{01} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\bullet T_{12} = \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 1 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\bullet T_{23} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\bullet P_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$\bullet P_0 = T_{01} T_{12} T_{23} P_3 = \begin{bmatrix} 1 + \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

# Kertaustehtävä 4 – malli 2/2

- $\dot{\mathbf{p}} = [\mathbf{0} \ \mathbf{1}]^T$
- $\begin{bmatrix} \omega_1 \\ \omega_2 \end{bmatrix} = J(\boldsymbol{\theta})^{-1} * \begin{bmatrix} \mathbf{0} \\ \mathbf{1} \end{bmatrix}$

$$\mathbf{A}^{-1} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{\det \mathbf{A}} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

$$J(\boldsymbol{\theta}) = \begin{bmatrix} -l_1 \sin \theta_1 - l_2 \sin(\theta_1 + \theta_2) & -l_2 \sin(\theta_1 + \theta_2) \\ l_1 \cos \theta_1 + l_2 \cos(\theta_1 + \theta_2) & l_2 \cos(\theta_1 + \theta_2) \end{bmatrix}$$

- $$J^{-1}(\boldsymbol{\theta}) = \frac{1}{(-s\theta_1 - s(\theta_1 + \theta_2)) * c(\theta_1 + \theta_2) + s(\theta_1 + \theta_2) * (c\theta_1 + c(\theta_1 + \theta_2))} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

$$= \frac{1}{\left(-\frac{1}{\sqrt{2}}\right) * \left(\frac{1}{\sqrt{2}}\right) + \left(\frac{1}{\sqrt{2}} * \left(1 + \frac{1}{\sqrt{2}}\right)\right)} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ -1 - \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ -\sqrt{2} - 1 & -1 \end{bmatrix}$$

- $$\begin{bmatrix} \omega_1 \\ \omega_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ -\sqrt{2} - 1 & -1 \end{bmatrix} * \begin{bmatrix} \mathbf{0} \\ \mathbf{1} \end{bmatrix} = \begin{bmatrix} \mathbf{1} \\ -\mathbf{1} \end{bmatrix}$$



# Kurssitentti

# Kurssitentti 16.4.2024

- Tentti on livetentti ja järjestetään tiistaina 16.4.2022 klo 9:00 – 11:30
- Tenttipaikkana toimii TUAS-talon luentosalit AS2 ja TU2 (tarkempi salijako ilmoitetaan ennen tenttiä)
- Tenttitehtävät ovat kurssin harjoitusten tyylisiä lasku- ja selitystehtäviä
- Tehtävät ovat suomeksi ja ruotsiksi
- Tentissä saa olla mukana laskin ( CAS myös sallittu )
- Tentissä ei jaeta varsinaista kaavakokoelmaa vaan tenttiin saa tehdä itselleen muistiinpanot ( tarkemmat tiedot tulevat Tentti-osion alle )
- HUOM! Jos sinulla on tenttiä koskevia erityisjärjestelypyyntöjä (esim. lisäaika tai erillinen tila) toimitathan tiedot niistä sähköpostitse pääassistentille mahdollisimman pian ellet ole jo sitä tehnyt.