

# Oppstartsveka: PID og Linjefølgning

---

## Innleiing

Denne veka er delt inn i tre hovuddelar som bygger på kvarandre, og målet er at studentane skal lære praktisk PID-regulering gjennom simulering, sanntids datainnsamling, og fysisk linjefølgning med robotbil. Dei skal gå frå å forstå PID-blokka visuelt, til å bruke sensorar og filtrere data, før dei set saman alt i ei konkurransebasert oppgåve med linjefølgning.

Heile opplegget legg vekt på å bygge intuisjon, vise sanntidsresultat, og halde det engasjerande med visuell tilbakemelding og lett gamification. Det er også viktig at oppgåvene gir mening i ein større samanheng, og at studentane får følelsen av å lage noko som faktisk fungerer.

## Del 1 – Lære PID

Vi har stjelt truck-simulasjonen frå TTK4111 og gjort nokre små justeringar. Simulasjonen viser no farta til bilen og har ein knapp for å skru av animasjonen (fordi den er treig – men det er kjekt for studentane å sjå noko anna enn ein graf).

Denne modellen er laga for å kunne svinge med kaskade-regulerin, vi har gjømt den indre pidden ferdig tuna inne i truckblokka, så dei trur dei kun ser på rattets vinkel- Skal jo ikkje skremme dei.

### Oppgåve 1 – Køyre til fartsgrensa

**Mål:** Skape forståing for kva som skjer inni ein PID-regulator ved å vise bidraga frå  $P$ ,  $I$ , og  $D$ -leddet eksplisitt. Bygg ein intuisjon.

#### Innhald:

- Blokkskjema av PID med eigen modul for kvar komponent
- Plotting av pådrag frå kvar komponent
- Animert og Grafisk framstilling av fartsregulering

#### Score:

- Me måler tidskonstant, poeng der etter
- Null poeng viss dei bryt fartsgrensa

### Oppgåve 2 – Forbikøyring med 2 PID

**Mål:** Lære å bruke Simulink sin ferdigbygde PID-blokk og få erfaring med å tune styring. Sjå at systemets dynamikk har noko å sei når man tune. Fartspidden vil ha store verdia, men denne PIDen vil kreve verdia lågare enn 1. Når dei skal tude Zumobilen vil dei nok trenge ein mellomting.

#### Innhald:

- Samme PID-verdi som i oppgåve 1 brukt til fart
- Tune ny og meir sensitiv PIS
- Visuell framstilling av forbikjøring

**Scenario:**

1. Start med referanse  $p =$  høyre kjørebane og  $v =$  fartsgrense
2. Når  $v$  er nådd  $\rightarrow$  bytt  $p$  til venstre fil,  $v = wroooooom$
3. Når ny  $v$  og  $p$  er nådd  $\rightarrow$  gå tilbake til original bane og fart

**Score:**

- Raskast gjennomføringstid
- 

## Del 2 – Lese sensordata og filter

**Mål:** Forstå sensorane på bilen og korleis rådata kan filtrerast for å få mening.

**Innhald:**

- Lese gyroskop og akselerometer i sanntid - filtrere/kombinere det og sjå forskjellen (slik som i ståhjulinglabben)
- Lese encodrar medan dei kjører zumo bilen manuelt
- Konvertere encoderdata til estimert fart
- *Stretch goal1:* Bruke encoderdata til å plote korleis bilen trur han har kjørt
- *Stretch goal2:* UAV-plot ved å lese imu data

**Test i simulasjon:**

- Rådata gir mykje støy
- Filtring (t.d. lavpass) nødvendig for stabilt estimat

**Score:**

- Kanskje: beste filtrerte kurve, eller kartlegging av bevegelse
- Vi må teste litt før vi landar noko konkret

**Intensjon:** Få studentane til å føle at dei har starta på eit kult og viktig studie (og det har dei jo).

---

## Del 3 – Linjefølgging med PID

**Mål:** Bruke kunnskapen frå del 1 og 2 til å lage ein fungerande linjefølgjar med tilhøyrande regulering.

**Innhald:**

- Tune PID for linjefølger
- Bruke data frå sensordelen til å:
  - Plotta fart
  - Estimere bevegelse (encoder/UAV-plot) (git stretch-måla)
- Tracke avvik frå linja over tid - i konkurranse medfører det straff å få stort avvik

**Konkurranse:**

- Rundetid

- Tidstraff ved for stort samla avvik (bilen blir spunnen i 5 sekund før han får køyre vidare)
- Avviksummen nullstillast etter straff

### Score:

- Raskaste rundetid med minst straff
- Lavt avvik belønnes indirekte gjennom færre straffar

## Oppsummering

Del	Tema	Mål	Score
1	Lære PID	Forstå blokkene og bygge intuisjon	Tidskonstant, unngå oversving
2	Sensor og filtrering	Lære om filtrering og sjå noko kult (gitt stretch 1 og 2)	Beste databruk / bevegelsesestimat
3	Linjefølgning med PID	Bruke alt i ein praktisk konkurranse	Raskast rundetid + avviksbasert tidsstraff

Kvar av oppgåvene kjem som eigne Simulink-filer. Studentane skal i stor grad trekke linjer mellom ferdigplasserte blokker og justere parameter. Undervegs blir det introdusert små «Simulink-triks» som er nyttige å kjenne til gjennom studiet — ting eg sjølv har lært meg gjennom prøving og feiling, og ofte har tenkt: «*Kvifor var det ingen som viste meg dette tidlegare?*»

Døme på slike element:

- **GoTo** ↔ **From**: for å rydde i signalnettverket
- **(De)Mux**: for å forstå og handtere arrays i signalbanar
- **Scope**, **Display** og liknande for sanntidsvisning
- Potensielt fleire nyttige verktøy som dukkar opp undervegs