

PID regulering

Fra teori til demonstrasjon

1. Bakgrunn og mål

Formål: Gi en grundig forståelse av PID-regulering (Proportional, Integral, Derivative), samt trening i å forklare og formidle dette videre til andre.

Læringsmål:

- Forklare P-, I- og D-leddets funksjon og rolle i en lukket reguleringsløyfe.
- Demonstrere og visualisere forskjellen mellom de tre leddene praktisk eller via simulering.
- Utvikle pedagogisk materiale – presentasjon og kort video – som viser hvordan PID regulering kan undervises.

2. Praktisk demonstrasjon: Fysisk oppsett eller simulering

Hvis prosjektet kan brukes i undervisningen og blir værende på Fagskolen Rogaland, kan dere søke støtte til å kjøpe inn nødvendig hardware.

Dere kan velge én av følgende, eller en kombinasjon:

Alternativ 1: Fysisk demonstrasjonsenhet

Bygge en enkel fysisk modell som tydelig viser effekten av P, I og D:

- Eksempel: vannnivåregulering i tank, ball på stråle, balanseplattform, DC-motor osv.
- Bruk Arduino eller PLC for styring og observasjon.
- Tuning via Ziegler–Nichols eller manuell metode hvor P, I og D endres stegvis og effekter dokumenteres.

Eksempler finnes for ball-beam-system, invertert pendel, motorstyring osv. For eksempel her:

- [PID Balance+Ball | full explanation & tuning](#)
- [PID brushless motor control tutorial](#)

- [PID temperature controller DIY Arduino](#)
- [Balancing Robot with PID - Mini Robot PCB](#)
- [AC 220V Heater Temperature PID and TRIAC control](#)
- [PID Control Explained: The Ultimate Guide from Basics to Implementation!](#)

Alternativ 2: Simulering

Utfør simuleringer med MATLAB/Simulink, Python eller lignende:

- Illustrer separat effekt av P, PI, PD og PID.
- Vis respons (step-response, overshoot, tid til stabilitet).

Alternativ 3: Kombinasjon

- Simulert og fysisk modell, eller en kombinasjon med bruk hardware-in-the-loop, slik at teori, simulering og praksis kobles.

3. Teoretisk del: Forklare PID-komponenter

Gi en tydelig, lett forståelig forklaring av:

- Proportional (P): reagerer på nåværende feil.
- Integral (I): adresserer akkumulerte, vedvarende feil.
- Derivative (D): forutser fremtidig feil ved å se på endringstakten.
- Beskriv hvordan summen av P, I og D-bidraget danner styringssignal:

$$u(t) = K_p e(t) + K_i \int e(t) dt + K_d \frac{de(t)}{dt}$$

- Diskuter praktiske utfordringer: steady-state feil (P), integrallagring og overshoot (I), støy i D, og tuning av parametere.

4. Innlevering

Det gis bonuspoeng for innlevering på GitHub med readme. Prosjektet kan også leveres som presentasjon, video eller en kombinasjon av disse.

Programmeringsfiler og simuleringsfiler skal leveres sammen med prosjektet.

GITHUB: lever code og forklaringer som readme på Github

- Prosjekter på Github
 - [Her er et video som forklarer hvordan man bruker github](#)
 - [Et annet video som forklarer github](#)
 - [Her er et video som forklarer hvordan man kan lage første prosjektet sitt på Github](#) og inviterer andre i gruppen til å jobbe med samme prosjekt
- Readme on Github:
 - [Her er et Readme template for Github dere kan bruke](#)
 - [Her er et annet template](#)
 - [Her er et eksempel](#)

Presentasjon: PowerPoint eller lignende som viser:

- Teori (P, I, D)
- Metodikk (hvordan demonstrasjonen fungerer)
- Resultater (data, grafer, observasjoner)
- Egen refleksjon: hva var faglig utfordrende, hva fungerte, hva lærte du?

Kort instruksjonsvideo (1–2 minutter):

- Viser oppsettet og demonstrerer forskjellen mellom P, I og D.
- Forklarer i praksis hva som skjer når du justerer hver parameter.

5. Vurderingskriterier

- Forståelse av PID-prinsipper og korrekt teoretisk forklaring.
- Kvalitet på demonstrasjon (tydelighet, funksjonalitet, relevans).
- Refleksjon og selvkritikk vedrørende gjennomføringen og forbedringspunkter.