Geregelde RC-DAC

Practicumverslag BARS2 – Intro Regeltechniek

Naam: Jurre Lievaart

Vak: BARS2 – Regeltechniek
Opleiding: TIS Delft
Datum: 2 oktober 2026

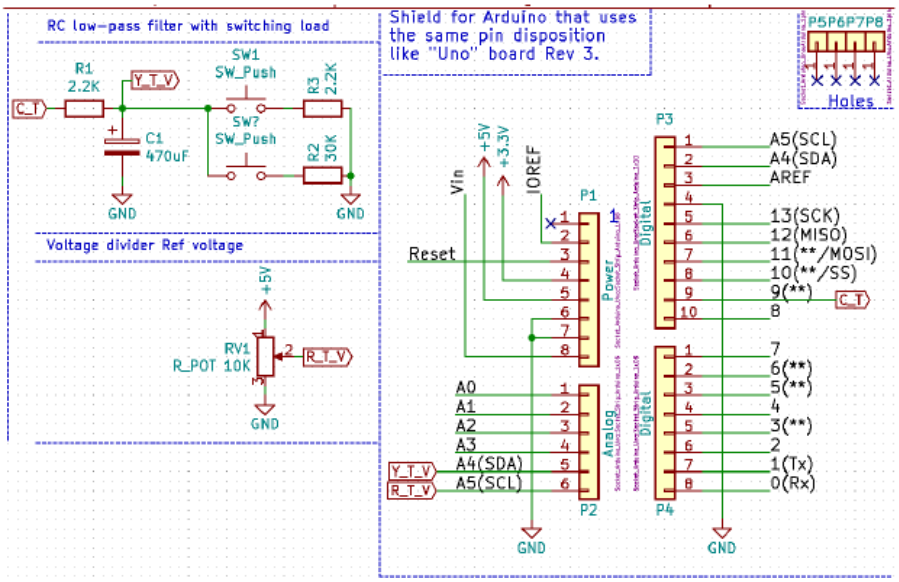
1. Inleiding

Een Arduino kan geen echte analoge spanning maken, alleen een PWM-signaal van 0 of 5 V. Met een RC-laagdoorlaatfilter wordt dat signaal gemiddeld tot een gelijkspanning tussen 0 en 5 V: een eenvoudige DAC (Digitaal naar analoog converter). In dit practicum is deze RC-DAC gebouwd en daarna voorzien van een regelaar, zodat de uitgang ook onder belasting het setpoint volgt.

Het verslag behandelt de vier deelopdrachten: de opbouw in open-loop (1), het verwerken van meetdata (2), het sluiten van de regellus met een P-regelaar (3) en het tunen van de regelaar (4).

2. Opstelling

De schakeling is opgebouwd volgens figuur 1 uit de opdracht. Pin 9 geeft het PWM-signaal c(t) via R1 = 2,2 kΩ naar C1 = 470 µF. De spanning over C1 is de uitgang y(t) en wordt gemeten op A4. Het setpoint r(t) komt van een potmeter (10 kΩ) op A5. Met drukknop SW1 wordt last 1 (R3 = 2,2 kΩ) aangesloten, met SW2 last 2 (R2 = 30 kΩ).

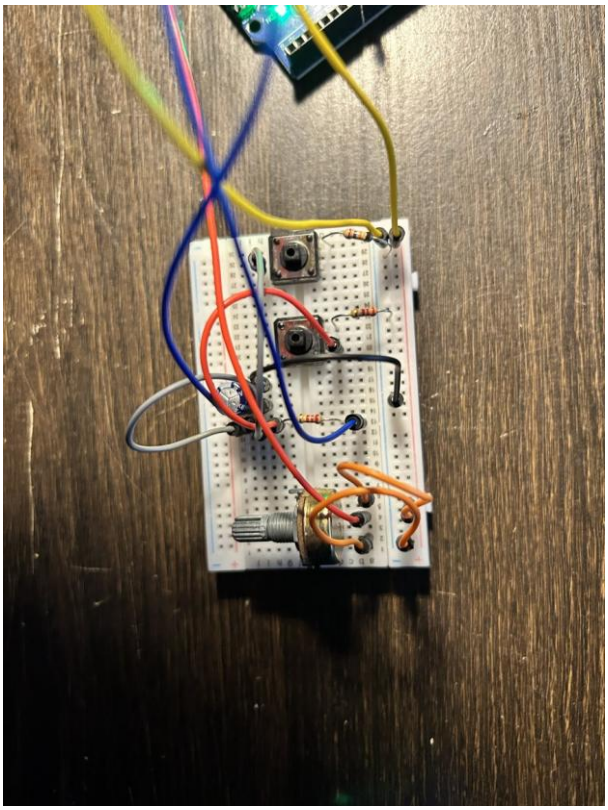


Figuur 1 – Aansluitschema RC-DAC (uit de opdracht)

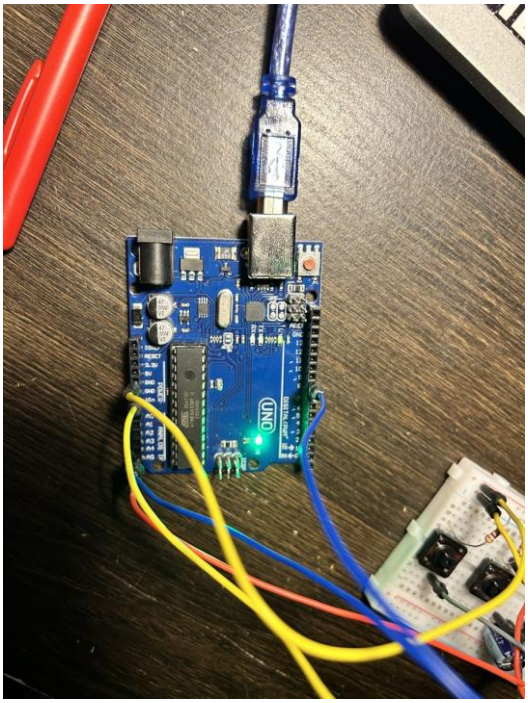
Theorie. De tijdconstante van het filter is τ = R1·C1 = 2200 Ω · 470 µF ≈ 1,0 s. Na ongeveer 5τ ≈ 5 s staat de uitgang op zijn eindwaarde. De PWM-frequentie is 490 Hz (periode 2 ms), veel sneller dan τ, dus de rimpel op y is verwaarloosbaar.

Een last vormt met R1 een spanningsdeler. Dat verlaagt de statische versterking Kss = y/c:

Situatie	Last	Kss	Max. uitgang (c = 5 V)
Geen last	—	1,00	5,00 V
Last 2 (SW2)	30 kΩ	0,93	4,66 V
Last 1 (SW1)	2,2 kΩ	0,50	2,50 V



Figuur 2 – Opbouw op het breadboard



Figuur 3 – Aansluitingen op de Arduino Uno

Foutzoeken. Tijdens de opbouw bleef y steeds hangen op 1,75 V. Door onderdelen één voor één los te halen bleek de condensator kapot. Na vervanging werkte de schakeling.

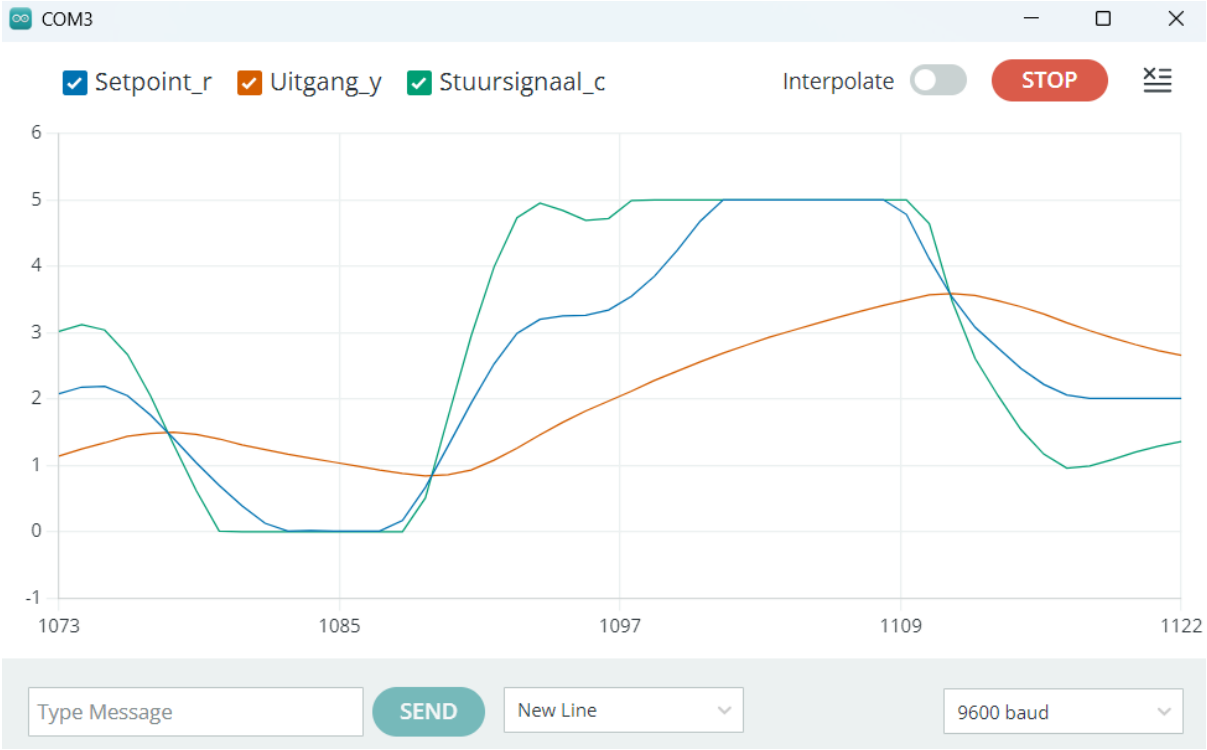
3. Deelopdracht 1 – Hardware en software

In de code van de docent zijn vijf regels ingevuld: de afgeleiden van r en y, de fout, de integraal en de regelwet.

```
r_t_diff = (r_t_v - r_t_past) / dt;           // afgeleide setpoint
y_t_diff = (y_t_v - y_t_past) / dt;           // afgeleide uitgang
e_t = r_t_v - y_t_v;                          // fout
if (Ki != 0.0) sum_e_t = sum_e_t + e_t * dt;  // integraal (spaarpot)
c_t = (Kc * r_t_v) - Kp * y_t_v + Kd_r * r_t_diff - Kd_y * y_t_diff + Ki * sum_e_t;
```

- **Fout:** e = r − y. Is de uitgang te laag, dan is e positief en stuurt de regelaar harder.
- **Afgeleide:** nieuw − oud, gedeeld door dt. Dat geeft de verandering in V/s.
- **Integraal:** elke ronde komt e·dt erbij. Zolang er een fout is, groeit de som.
- **Regelwet:** de P-actie werkt op y met een minteken (negatieve terugkoppeling). De compensator Kc = Kp + 1 zorgt dat y zonder last precies op r uitkomt.
- **Omrekenen:** ADC 0–1023 naar volt met y = waarde · 5,0/1023, en volt naar PWM 0–255 met c · 255/5,0. Zonder ".0" wordt 5/1023 een deling van gehele getallen en is de uitkomst 0. map() is niet gebruikt.

In open-loop (OPENLOOP 1) is Kp = Ki = 0 en Kc = 1, dus c = r. De uitgang volgt dan de potmeter met een vertraging van ongeveer τ. Dit is gedemonstreerd en afgetekend.

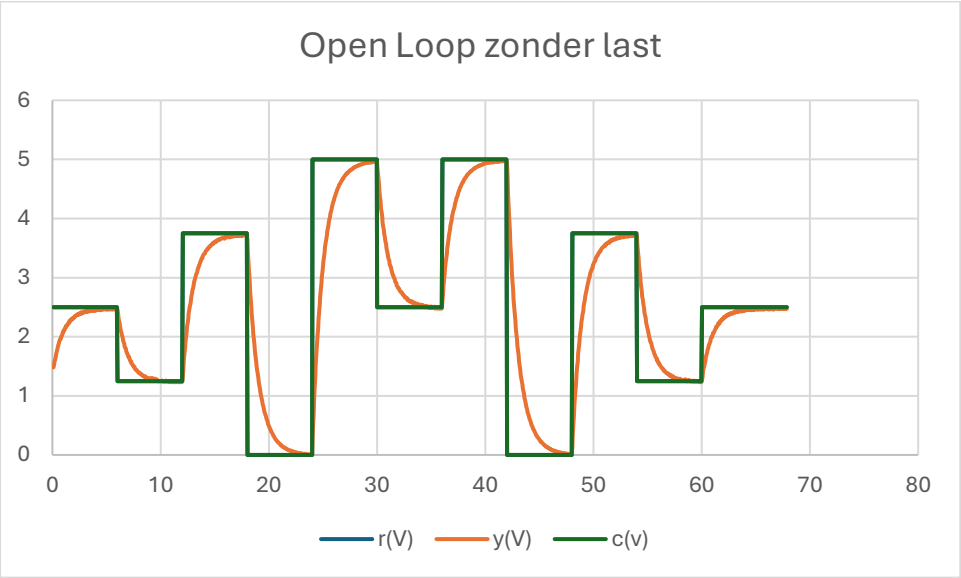


4. Deelopdracht 2 – Meetdata in open-loop

De testserie is in de code vastgelegd, met 6 s per stap (> 5τ):

2,5 → 1,25 → 3,75 → 0 → 5 → 2,5 → 5 → 0 → 3,75 → 1,25 → 2,5 V

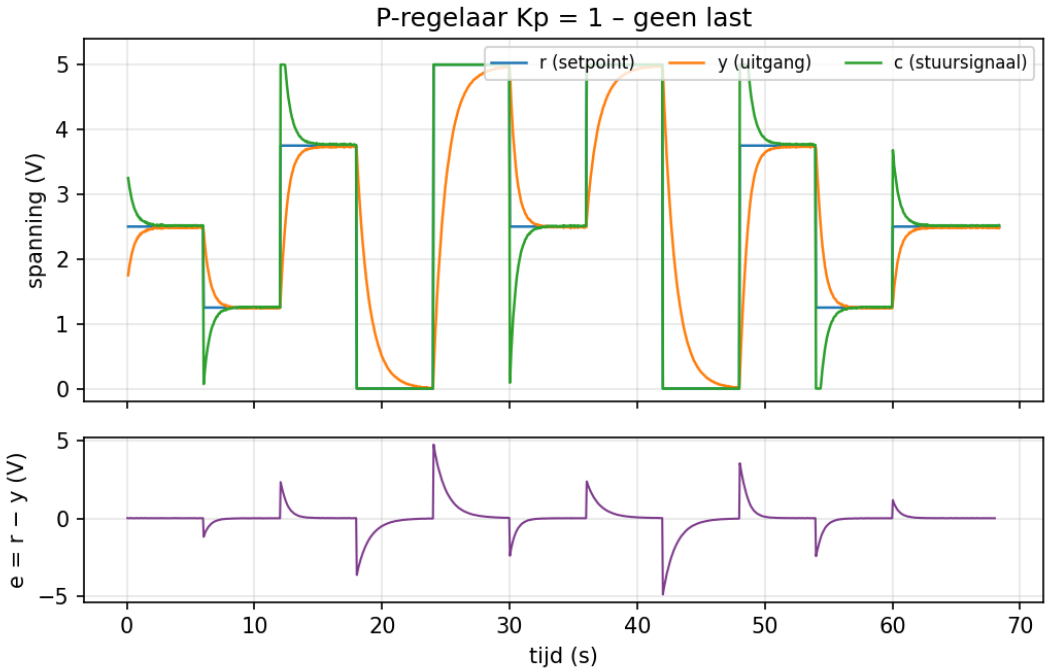
Hiermee worden beide series uit de opdracht (omhoog en omlaag, met 0, 1,25, 2,5, 3,75 en 5 V) in één meting doorlopen. De data is via de seriële poort met PuTTY gelogd en in Excel verwerkt.



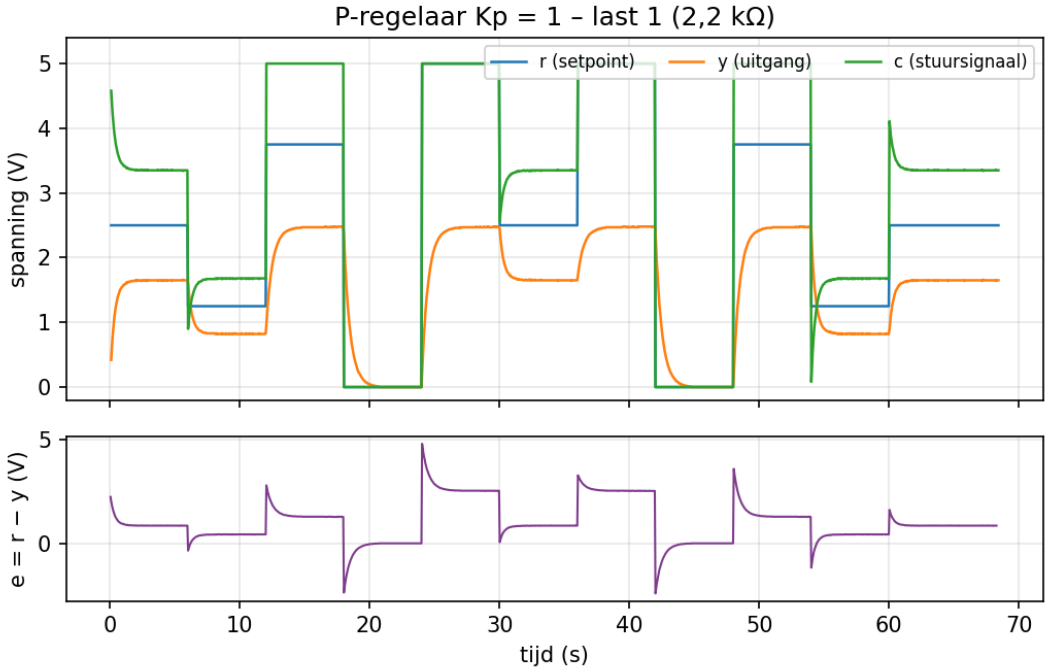
In open-loop is c gelijk aan r. De uitgang y volgt elke stap volgens een eerste-orde laadcurve: na ongeveer 1 s is 63% van de stap bereikt, na ongeveer 5 s de eindwaarde. Zonder last komt y op r uit.

5. Deelopdracht 3 – Sluiten van de regellus

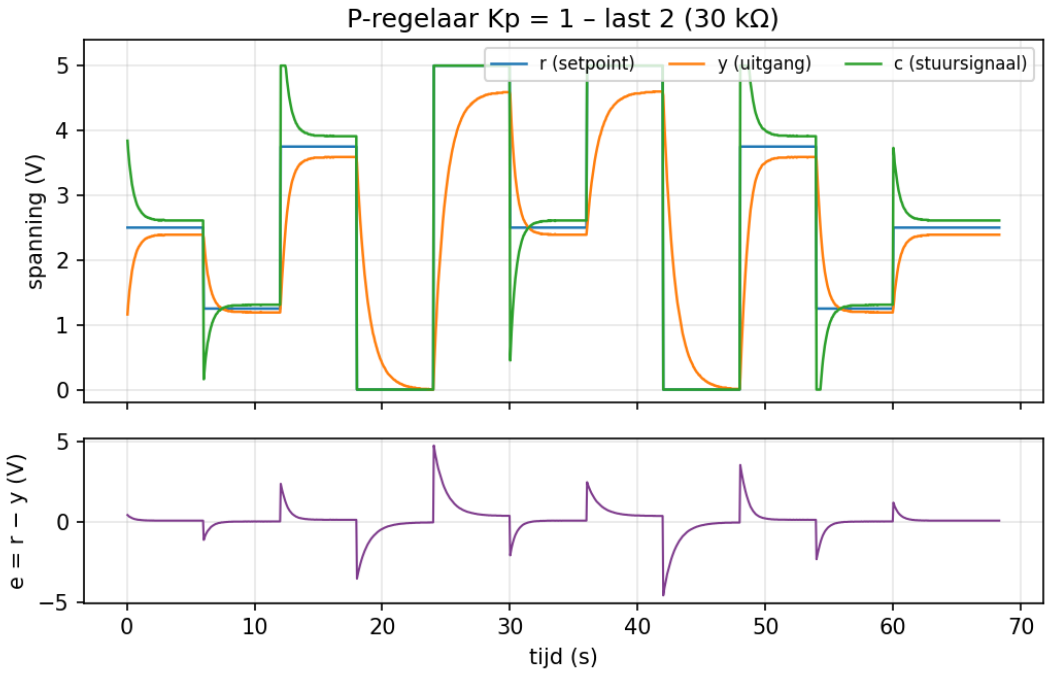
Instellingen: OPENLOOP 0, Kp = 1, Ki = 0. De software zet de compensator automatisch op Kc = 1 + Kp·Kss = 2 (Kss = 1). Dezelfde testserie is drie keer gemeten: zonder last, met last 1 en met last 2.



Figuur 4 – P-regelaar zonder last



Figuur 5 – P-regelaar met last 1 (2,2 kΩ)



Figuur 6 – P-regelaar met last 2 (30 kΩ)

Uitgang y aan het eind van elke stap, met de offset ten opzichte van r:

Situatie	r = 1,25 V	r = 2,50 V	r = 3,75 V	r = 5,00 V	< 15%?
Geen last	1,24 (0,8%)	2,49 (0,4%)	3,74 (0,3%)	4,96 (0,8%)	Ja
Last 2 (30 kΩ)	1,19 (4,8%)	2,39 (4,4%)	3,59 (4,3%)	4,59 (8,2%)	Ja
Last 1 (2,2 kΩ)	0,82 (34%)	1,65 (34%)	2,48 (34%)	2,48 (50%)	Nee

Zonder last volgt y het setpoint binnen 1%. Na een stap stuurt de regelaar eerst extra hard (c schiet omhoog of omlaag) en bereikt y het setpoint binnen ongeveer 2 s, sneller dan in open-loop. Bij de stappen naar 5 V en 0 V is c verzadigd.

Onder last blijft een offset over, omdat een P-regelaar alleen bijstuurt zolang er een fout is. Bij last 2 is dat ongeveer 4–5%, en 8% bij 5 V. Bij last 1 is het 34%, en boven 2,5 V komt y niet uit: R1 en R3 vormen een spanningsdeler van 2,2 kΩ / 2,2 kΩ, dus ook met c = 5 V is y maximaal 2,5 V.

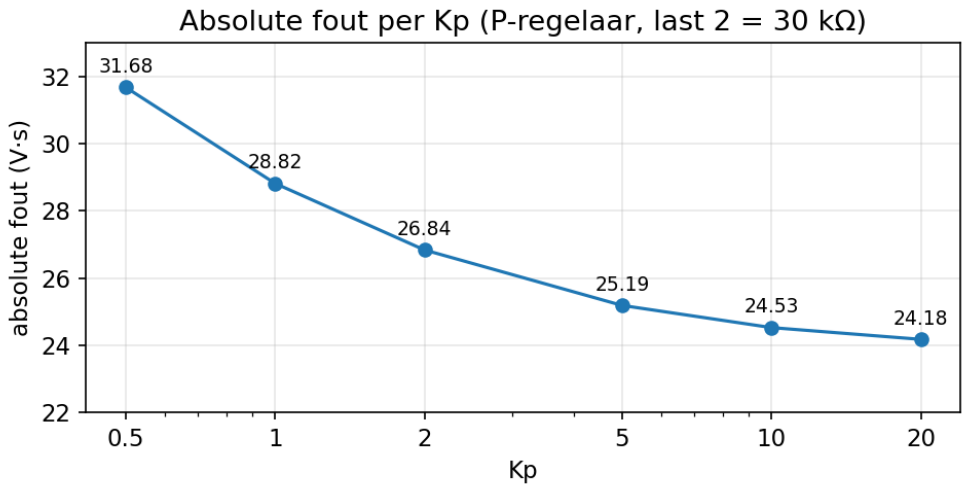
Oordeel: de P-regelaar met Kp = 1 voldoet zonder last en bij last 2 (binnen 15%), maar niet bij last 1.

6. Deelopdracht 4 – Regelaar tunen

Met aangepaste code test de Arduino meerdere instellingen automatisch na elkaar. Per test wordt de absolute fout over de hele testserie opgeteld: Σ |r − y| · dt. Hoe kleiner, hoe beter. Alle metingen zijn met last 2 (30 kΩ) gedaan.

P-regelaar

Kp	0,5	1	2	5	10	20
Absolute fout (V·s)	31,68	28,82	26,84	25,19	24,53	24,18



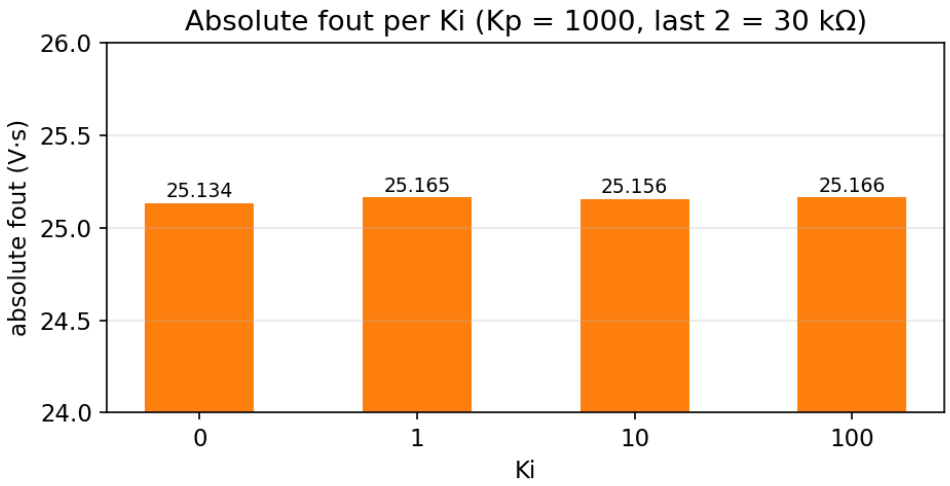
Figuur 7 – Absolute fout per Kp

Hoe hoger Kp, hoe kleiner de fout, maar de winst wordt steeds kleiner. Bij een hoge Kp staat c na een stap direct op 0 of 5 V; sneller kan de condensator niet laden. Het RC-circuit is een eerste-orde systeem zonder vertraging en wordt daardoor niet instabiel. In overleg met de docent is verder gegaan met Kp = 1000.

PI-regelaar

Bij Ki ≠ 0 is in de code Kc = Kp gezet, zodat c = Kp·(r − y) + Ki·∫e dt. Met Kc = 1 werkt de regelaar bij Kp = 1000 niet, omdat de P-actie y dan bijna naar 0 V trekt.

Ki (Kp = 1000)	0	1	10	100
Absolute fout (V·s)	25,134	25,165	25,156	25,166



Figuur 8 – Absolute fout per Ki bij Kp = 1000

Het verschil tussen de Ki-waarden is 0,03 V·s (0,1%) en valt binnen de meetonnauwkeurigheid. Met Kp = 1000 is de offset al kleiner dan één ADC-stap (5 mV), dus de I-actie heeft niets meer weg te regelen. De Ki-reeks begint elke test met een lege condensator, de Kp-reeks niet; de twee reeksen zijn daarom niet exact vergelijkbaar.

Keuze

- **Beste P-regelaar: Kp = 1000.** Laagste fout, stabiel, offset kleiner dan meetbaar.
- **Beste PI-regelaar: Kp = 1000, Ki = 10.** Laagste fout van de metingen met Ki ≠ 0; het verschil is verwaarloosbaar.

Bij dit systeem voegt Ki niets toe, omdat een hoge Kp al volstaat. Ki is nuttig als Kp laag moet blijven, bijvoorbeeld bij een systeem met vertraging dat anders gaat slingeren. Bij Kp = 1 (deelopdracht 3) bleef onder last 1 nog 34% offset over; die zou de I-actie wel wegwerken.

7. Conclusie

Met een PWM-signaal en een RC-filter (τ ≈ 1 s) is een werkende DAC gemaakt die in open-loop het setpoint volgt. Een P-regelaar met Kp = 1 maakt het systeem sneller en houdt zonder last de offset onder 1%, maar onder zware last (2,2 kΩ) blijft 34% offset over. Een hogere Kp verkleint de fout steeds verder; bij dit eerste-orde systeem kan Kp zeer hoog worden (1000) zonder instabiliteit. De I-actie voegt bij die Kp niets meetbaars toe. Boven 2,5 V is last 1 fysiek niet te regelen, omdat de PWM dan al op 100% staat.