

Professional Drone, Hybrid Power Pack - Timebox 5

Team 2
Aarhus Universitet, Herning.

28. februar 2019

Stud. nr: 201602094	Navn: Søren Holm Korsgaard
Stud.nr.: 201607563	Navn: Jacob Gustafsson
Stud.nr.: 201704859	Navn: Jonas Buus
Stud.nr.: 20084327	Navn: Simon Rasmussen
Stud.nr.: 201704483	Navn: Thomas Dueholm Jensen

Indhold

1	Strategy and planning (Jonas)	2
2	Motorstyring(Simon)	3
3	Analyse	3
3.1	Kravsspecifikation	3
3.1.1	Overordnet mål og baggrund	3
3.2	Testsetup	4
4	Elstarter	5
4.1	Analysis (Thomas)	5
4.2	Verifikationsbeskrivelse (Thomas)	5
4.3	Implementering (Søren)	6
5	Deployment	7

1 Strategy and planning (Jonas)

I timebox 4 erfarede vi, at vi ikke havde verificeret et vigtigt krav ift. ensretteren. Amplituden på indgangssignalet skal være på minimum $9V_p$ (jf. datablad for lt4320¹). I vores udregninger hidtil, havde vi kun taget højde for spændingsniveauer ved max. rpm på forbrændingsmotoren. Vi har ikke målt, hvilke spændingsniveauer motoren genererer ved tomgangsdrift. Derfor har vi valgt, at stoppe arbejdet på ensretteren, indtil vi har målt spændingsniveauerne genereret ved tomgang.

Ensretteren har bidraget med et uventet stort tidsforbrug, som gør at vi må revurdere vores development plan for de sidste 5 timeboxes.

Motivationen for revurderingen, skyldes også, at vi er blevet i tvivl om hvorvidt vor nuværende teststand vil fungere i praksis. Motoren larmer enormt meget, producerer kraftige rystelser i hele teststanden (hvorpå måleudstyr befinder sig), og sikkerhedsaspektet er tvivlsomt, taget afskærmning i betragtning i forhold til den mængde energi motoren producerer. Udstødningsgasser er endnu en væsentlig faktor, eftersom vi er nødt til at opholde os meget tæt på motoren, på nuværende tidspunkt.

Udsigten til, at skulle bruge adskillige timer på test med forbrændingsmotoren virker ikke hensigtsmæssig, og måske endda sikkerhedsmæssigt uforsvarlig. Dette har bragt os ind i en diskussion om at designe en ny teststand, hvor forbrændingsmotoren erstattes af en AC-motor, eller hvor teststanden bygges om, så motor er afskærmet, adskilt fra måleudstyr mv.

I denne timebox arbejder Simon videre med at opstille kravs-specifikation, samt verifikation, for motorstyringen, og med at afslutte analysen. Endvidere kommer Simon ind på hvordan denne kan optimeres gennem test.

Thomas afslutter analysen vedr. elstarteren, og opstiller samtidig en verifikationsbeskrivelse for realisering af krav-nr. 2.1.1.1.

Afslutningsvist vil Søren beskrive implementering af den kode, som bevirker elstarterens funktionalitet, og samtidig beskrive softwarens flow, så denne kan verificeres i næste timebox.

Rapporten er samlet af Jonas.

¹Linear Technologies, 2019

2 Motorstyring(Simon)

3 Analyse

Målet med dette afsnit er at opstille en kravsspecifikation for motorstyringen samt at færdiggøre analysen af motorstyringen og lægge op til hvordan en test af motorstyring kan udformes for at præcisere PID-algoritme samt at opfylde krav.

3.1 Kravsspecifikation

Med baggrund i Fjare²

- Overshoot skal ikke være mere en 197 rpm.
- Justeringstiden må max være 8,8 sekunder.
- Ifm. et step respons skal 90 % af målet være opnået i mindre end 3 sekunder.
- Det skal være muligt at vedligeholde en konstant hastighed over længere tid (dvs. mere end 5 minutter).

3.1.1 Overordnet mål og baggrund

Endeligt formål er at finde de optimale værdier af K_P , K_i og K_d i PID-reguleringen. For et givent sæt af K-værdier, udvælges en række forskellige typer af trinvis ændringer i hastigheden. Der måles på hvor lang tid det tager for ændringen at indfinde sig, samt hvor store udsving dette gør sig ud i.

Som udgangspunkt er det tidligere vist⁽³⁾ at $K_p = 0,065$ og $K_i = 0,000005$ har været optimale, samt at K_d helt blev droppet eftersom det ikke var en gavnlig parameter.

²Paul D. Fjare, 2014

³Paul D. Fjare, 2014

Fra Fjare⁴ kunne et eksempel på et program være:

```
void velPID ( ) {

    lowpassSpeed = alpha * lastLowpassSpeed + (1-alpha ) * measuredSpeed;
    K1 = kp * setpointWeight * (setpointSpeed - lastSetpointSpeed) + kp * (lastMeasuredSpeed - lowpassSpeed);
    K2 = ki * (setpointSpeed - lowpassSpeed);
    K3 = kd * (2 * lastMeasuredSpeed - lowpassSpeed - % lastLastMeasuredSpeed);
    output = lastOutput - K1 - K2 - K3;
    throttlePos = floor(output + 0.5);

    if(throttlePos<throttleopen){
        output = (double) throttleopen;
        throttlePos=throttleopen;
    }

    if(throttlePos>throttlesafe){
        output = ( double )throttlesafe;
        throttlePos=throttlesafe;
    }

    lastLowpassSpeed = lowpassSpeed;
    lastLastMeasuredSpeed = lastMeasuredSpeed;
    lastMeasuredSpeed = lowpassSpeed;
    lastSetpointSpeed = setpointSpeed;
    lastOutput = output;
    throttle.writeMicroseconds(throttlePos);
}
```

Figur 1: PID-program. Fjare, 2014.

3.2 Testsetup

Det skal være muligt at

- servomotoren kan reguleres
- motoren kan startes
- omdrejningerne kan måles

Et testprogram kan køres via microcontroller som foretager PID-reguleringen og optage motorens omdrejninger ved forskellige skift mellem hastigheder. Disse gentages for forskellige værdier af K_p og K_i .

⁴Paul D. Fjare, 2014

4 Elstarter

4.1 Analysis (Thomas)

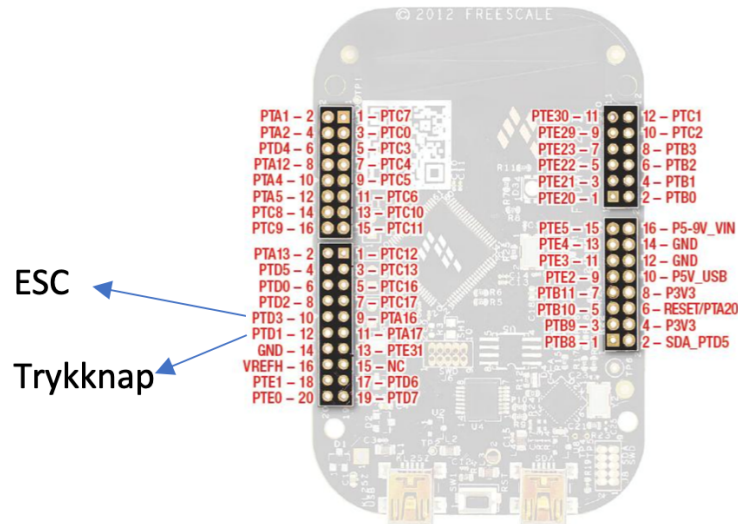
I denne del af timebox 5 skal kravet om el-start af forbrændingsmotoren (krav-nr. 2.1.1.1 fra Launch fasen) realiseres og sendes til deployment. Realiseringen vil blive udført vha. et software program udviklet i programmeringssproget C. Programmet skal kunne generere et 50 Hz PWM signal med en duty-cycle på 1.25 ms. PWM signalet vil starte BLDC generatoren, som derved genererer nok omdrejninger til at forbrændingsmotoren kan starte.

Inden elstarten kunne testes, var det nødvendigt at sikre, at forbrændingsmotoren var i stand til at starte. Derfor blev der foretaget grundindstillinger af karburator, hall-sensor og tændingsmodul, hvorefter det var muligt at starte forbrændingsmotoren med et PWM signal genereret af et oscilloskop fra el-lab, AU Herning.

4.2 Verifikationsbeskrivelse (Thomas)

Herunder er oplistet, hvordan krav-nr. 2.1.1.1 verificeres.

1. Start mcuXpresseIDE
2. Forbind MCU, FRDM-KL25Z, til computeren via USB-porten
3. Forbind KL25Z pin #76 til PWM-signal ledningen (orange) på ESC'en (Se figur 3)
4. Forbind KL25Z pin #74 til ønsket trykknop (Analog Discovery static IO eller fysisk trykknop på breadboard)
5. Tryk på build, og derefter Debug i mcuXpresso for at loade programmet til KL25Z. Afslut med F8 for at aktivere programmet
6. Aktivér og hold den valgte trykknop nede, indtil forbrændingsmotoren er startet
7. Konstatér at forbrændingsmotor er startet vha. elstart.



Figur 2: PIN-diagram over FRDM-KL25Z.

4.3 Implementering (Søren)

I launch fasen har vi beskrevet et krav om at kunne starte benzinmotoren med en el-starter. I timebox 3 viste vi hvordan et pwm-signal kunne drive motoren. Vi viste, at et signal på 50 Hz, med en duty-cycle på 1,25 ms ville generere 500 rpm (+/- 100) på benzinmotoren, mens en duty-cycle på 1 ms ville stoppe opstartsprocessen. Denne funktionalitet er implementeret i nedstående kode (Se figur 3). Selve opsætning af pwm, er vist i timebox 4, da den har samme opsætning som servo-motoren.

Flow for opstart af motor, ved funktionen start():

1. Startknap initialiseres
2. Afventer at brugeren trykker på knappen
3. En timer opsættes, som tæller vores 'i' værdi når der er gået 500 ms.
4. Så prøver motoren at starte benzinmotoren, og det vil den gøre så længe start knappen er trykket ned (dog i maks. 15 sek)
5. Efter 15 sek vil opstartsprocessen afbrydes.

```

2 void init_start() {
3     // set clock to PORTD
4     SIM->SCGC5 |= SIM_SCGC5_PORTD_MASK;
5     // set pin too gpio
6     PORTD->PCR[START_BTN] |= (1UL << 8);
7     // set pin to input
8     PTD->PDDR = (0UL << START_BTN);
9 }
10
11 void start() {
12     init_start();
13     while (1) {
14         if (PTD->PDIR & (1UL << START_BTN)) {
15             // Lave et interrupt for 500 ms
16             initSysTickTmr(1500000);
17             //Sætter vores pwm signal med en puls på 1,25ms, dette genere 500 rpm +- 100 rpm
18             TPM0->CONTROLS[PWM_CH_ESC].CnV = START;
19             // Den skal starte så længe knappen er trykket ned og må maks cranke i 15 s
20             while (i != 30 && PTD->PDIR != (0UL << START_BTN))
21                 ;
22             __disable_irq();
23             i = 0;
24             //Sætter vores pwm signal puls til 1 ms, hvilket stopper Starteren
25             TPM0->CONTROLS[PWM_CH_ESC].CnV = MIN_VAL_SERVO;
26             break;
27         } else
28             continue;
29     }
30 }
31
32
33
34

```

Figur 3: Kodeudsnit: Opstartsprocess for motor.

5 Deployment

Hermed godkender kunderne, Morten Opprud Jakobsen og Jan Møller Nielsen, ovenstående i time-box 5.

Mandag d. 18/2-2019

Morten Opprud Jakobsen

Jan Møller Nielsen