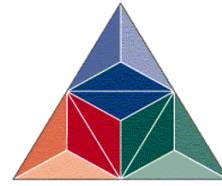


HØGSKOLEN
i Buskerud

Dokumentoversikt

Prosjektbeskrivelse	1
Etteranalyse	2
Forprosjekt	3
Prosjektplan	4
Kravspesifikasjon	5
Testplan	6
Testspesifikasjon	7
Testlogg	8
Testrapporter	9
Brukermanual	10
Multikopter teori	11
Komponentdokumentasjon	12
Økonomi	13
Designdokument reguleringsystem	14
Designdokument testbenk	15
Designdokument programvare regulert	16
Implementasjon programvare	17
Foreningsdokument	18
Testbenk	19
Matematisk modell	20
Motormodell	21
Parameterfremstilling	22
Sensorfusjon	23
Simuleringsdokument	24
Seriell protokoll	25
Iterasjonsplaner	26
Iterasjonsrapporter	27
Vedlegg	28



HØGSKOLEN
i Buskerud

Oppgavetittel: Hovedprosjekt - Prosjektbeskrivelse - v3.0		
Fag(nr./navn): SFHO 3200 / Prosjektstyring med hovedprosjekt		
Gruppemedlemmer, signatur:		
_____ Vegard Torkelsen	_____ Michael Odden	_____ Herå Rørvik
_____ Trine Lindberg	_____ Luke Carambot	_____ Patrick Bjørum
Intern veileder: Dag Samuelsen Intern sensor: Olaf Hallan Graven Ekstern veileder og sensor: Ragnvald Otterlei		
Dato: 2011-04-06		

Innholdsfortegnelse

Dokumenthistorie.....	1
Innledning.....	1
1 Hensikt, mål.....	2
2 Delprosjekter.....	2
3 Prosjektbeskrivelse.....	2
3.1 Multikopteret.....	2
3.1.1 Reguleringsystemet.....	2
3.1.2 Sensorer.....	2
3.1.3 Mikrokontroller.....	3
3.1.4 Databehandling.....	3
3.1.5 Støy.....	3
3.1.6 Fjernkontroll.....	3
3.1.7 Matematisk modellering.....	3
3.1.8 Konstruksjon.....	3
3.1.9 Ønskelige funksjoner.....	3

Dokumenthistorie

Versjon	Beskrivelse	Når
1.0	Dokument opprettet	
2.0	Rettet skrivefeil	
3.0	Oppdatert etter riktig mal, rettet gjenstående skrivefeil	2011-04-06

Tabell 1: Dokumenthistorie

Innledning

Dette prosjektet har blitt definert av gruppen og Simicons representant Ragnvald Otterlei gjennom et møte tirsdag 14.09.10.

Prosjektet går ut på å utvikle og konstruere et fire rotors "helikopter" som skal være stabilt og lett å fly via en fjernkontroll. Multikopteret skal stabiliseres ved hjelp av et reguleringsystem i tre akser, roll, pitch og yaw, samt ved hjelp av nødvendige sensorer som gyroer og akselerometere. Reguleringen og stabiliseringen gjøres via en mikrokontroller.

1 Hensikt, mål

Målet med dette prosjektet er å demonstrere et firerotors multikopter som flyr stabilt, og som er enkelt å fly via en fjernkontroll.

Hensikten er å lage et lett styrlig fjernstyrt fartøy som skal kunne brukes i praktiske sammenhenger, for eksempel kan et kamera eller IR-kamera tilkobles for søk etter savnede personer i vanskelig terreng.

2 Delprosjekter

Det vil være hensiktsmessig å dele opp oppgaven i flere deloppgaver slik som:

- Design og valg av komponenter
- Innkjøp av ramme, motorer, sensorer, etc.
- Utvikling av reguleringssystem og payload
- Bygging og sammenstilling av plattformen
- Test av delsystemer - styring, payload og plattform
- Systemintegrasjon og systemtester
- Demonstrasjon

Opgaven krever god planlegging og risikostyring. Mange av deloppgavene er avhengig av at andre deloppgaver er fullført før de kan startes.

3 Prosjektbeskrivelse

Prosjektgruppen består av tre kybernetikkstudenter, to mekatronikkstudenter og en datastudent. Alle disse fagområdene vil inngå i oppgaven.

3.1 Multikopteret

3.1.1 Reguleringssystemet

Motorhastighetene må reguleres slik at fartøyet holder seg så stabilt som mulig, samtidig som den har såpass lav reaksjonstid at man kan navigere den på en fornuftig måte i et fornuftig tempo. Her må vi vurdere om vi skal bruke et vanlig PID system med tilbakekobling, eller om det er andre metoder som kan vise seg å være bedre for denne typen anvendelser.

3.1.2 Sensorer

Det finnes en rekke sensorer som kan passe inn i dette prosjektet. Vårt arbeid blir da å finne de sensorene som kan anvendes slik at fartøyet flyr så stabilt som mulig, uten å bruke unødvendige

sensorer som øker vekt, kostnader og kompleksitet. Det kan også være praktisk å konstruere et sensorbrett, slik at vi får samlet alle sensorene på ett og samme kretskort.

3.1.3 Mikrokontroller

Vi må også finne en mikrokontroller som er så billig og primitiv som mulig, for å holde kostnadene nede, men som har nok innganger og utganger til sensorer og motorer samt høy nok prosessorhastighet.

3.1.4 Databehandling

Vi har en rekke data som må behandles av mikroprosessen.

3.1.5 Støy

Sensorene som skal brukes kommer til å inneholde støy som sannsynligvis må filtreres bort. Vi må også kunne bruke et filter som klarer å identifisere den verdien som er nærmest den egentlige bevegelsen av fartøyet. For eksempel, hvis vi bruker to sensorer til å finne vinkelen på roll, må vi anvende et filter som klarer å tilnærme den egentlige vinkelen så nære som mulig. Her har vi en rekke muligheter som må undersøkes, slik som *weighted average*, *kalman filter*, *bayesian networks* og *Dempster-Shafer filtre*.

3.1.6 Fjernkontroll

Signalkvaliteten fra fjernkontrollen må undersøkes og vi må se om vi trenger å behandle dette signalet slik at mikrokontrolleren enkelt klarer å identifisere de dataene den mottar fra fjernkontrollen.

3.1.7 Matematisk modellering

Vi må utvikle en matematisk modell over systemet, slik at vi bedre kan forutse hvordan fartøyet oppfører seg i luften. Den matematiske modellen vil også hjelpe oss i å utvikle et fungerende reguleringssystem, samtidig som den kan hjelpe oss å bestemme en rekke andre faktorer som størrelse, tregghetsmoment, kritisk vinkel på roll og pitch. Med kritisk vinkel mener vi den maksimale vinkelen fartøyet kan ha mens det fortsatt klarer å opprettholde stor nok løftekraft i forhold til tyngdekraften. Den matematiske modellen kan også brukes til å utvikle en enkel simulator, slik at vi kan se hvordan justeringer i reguleringssystemet påvirker flydynamikken.

3.1.8 Konstruksjon

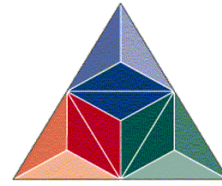
Siden konstruksjon ikke er fagfeltet til noen i gruppa, kommer vi til å kjøpe et ferdig byggesett for et multikopter og gjøre de nødvendige endringene for å tilpasse det til vårt bruk.

3.1.9 Ønskelige funksjoner

Det vil også være ønskelig å kunne konstruere en nødlandingsfunksjon ("failsafe") på fartøyet slik at hvis det mister signalet fra fjernkontrollen vil den utføre en funksjon som fører til en automatisk

landing.

I tillegg til dette vil det også være ønskelig å gi fartøyet en praktisk funksjon som f.eks et kamera.



HØGSKOLEN
i Buskerud

Oppgavetittel: Hovedprosjekt - Etteranalyse - v1.0		
Fag(nr./navn): SFHO 3200 / Prosjektstyring med hovedprosjekt		
Gruppemedlemmer, signatur:		
_____ Vegard Torkelsen	_____ Michael Odden	_____ Herå Rørvik
_____ Trine Lindberg	_____ Luke Carambot	_____ Patrick Bjørum
Intern veileder: Dag Samuelsen Intern sensor: Olaf Hallan Graven Ekstern veileder og sensor: Ragnvald Otterlei		
Dato: 2011-05-28		

Innholdsfortegnelse

Dokumenthistorie	1
Definisjoner og forkortelser.....	1
Innledning.....	1
1 Måloppfyllelse.....	2
1.1 Prosjektresultat.....	2
1.1.1 Krav og testing.....	2
1.2 Kostnader.....	3
1.3 Evalueringer av produktet.....	3
1.3.1 Gyroregulert modus.....	3
1.3.2 Vinkelregulert modus.....	4
1.3.3 Simulering.....	4
1.3.4 Dragonfly Testbench GUI.....	4
1.3.5 Generell brukernytte av produktet.....	4
2 Prosjektgjennomføring.....	5
2.1 Arbeidsmetoder.....	5
2.2 Samarbeid inn- og utad.....	5
2.2.1 Samarbeid og miljø i prosjektgruppen.....	5
2.2.2 Oppdragsgiver, ekstern veileder og sensor.....	5
2.2.3 Intern veileder.....	6
2.2.4 Sykdom og fravær.....	6
2.3 Prosjektadministrasjonen.....	6
2.3.1 Ansvarsfordeling.....	6
2.3.2 Møter og rapportering.....	6
2.4 Timebruk.....	7
2.4.1 Vurdering av timesrapportene.....	7
2.4.2 Timesrapport per medlem.....	8
2.4.3 Timesrapport per aktivitet.....	9
2.4.4 Timesrapport per sak.....	10
Konklusjoner.....	17
Referanser.....	18

Dokumenthistorie

Versjon	Beskrivelse	Når
1.0	Første offisielle versjon.	2011-05-28
1.1	Noe tekst lagt til, korrekturlest	2011-05-30

Tabell 1: Dokumenthistorie

Definisjoner og forkortelser

PID	Proporsjonal-integral-derivat regulator
Pitch	Rotasjon rundt bredden av fartøyet
Roll	Rotasjon rundt lengden av fartøyet
Yaw	Rotasjon rundt sin egen z-akse
P	Roll - vinkelhastighet
Q	Pitch - vinkelhastighet
R	Yaw - vinkelhastighet

Tabell 2: Definisjoner og forkortelser

Innledning

Dette dokumentet er en etteranalyse av vår hovedoppgave - Project: Dragonfly - ved HiBu 2011, og beskriver hvordan prosjektet har gått og hvorfor ting har blitt gjort som det er gjort.

Målet med dette prosjektet var å bygge og demonstrere et lettstyrt, fjernstyrt firerotors multikopter. Stabiliseringen skulle gjøres ved bruk av et reguleringsystem og sensorer.

1 Måloppfyllelse

1.1 Prosjektresultat

Generelt er resultatet av prosjektet er at vi har nådd de viktige målene og oppfylt de viktigste kravene for prosjektet. I de følgende kapitlene spesifiseres dette nærmere.

1.1.1 Krav og testing

Tester har blitt utført etter testspesifikasjonen[2] for å sikre at alle kravene gitt i kravspesifikasjonen[1] skulle bli oppfylt. Ikke alle kravene ble oppfylt på grunn av tiden det tok å designe og implementere A og B kravene. C og D kravene var de mindre viktige kravene, og disse ble nedprioritert og derfor ikke godkjent.

Oppsummert oversikt over godkjente krav:

- Alle A krav
- Nesten alle B krav
- Ingen C og D

Første mål i prosjektet var å konstruere systemet slik at fartøyet kunne fly uten regulering. Poenget var å få systemet til å fungere først, og deretter bygge på med det litt mer komplekse, som regulatorer med tilbakekobling.

En god del tid ble brukt på å få fartøyet flyvbart uten regulering. Vi kom til slutt frem til at det er umulig for en person å reagere raskt og jevnt nok til å holde fartøyet kontrollert i luften. En nærmere undersøkelse kunne muligens ha klarlagt dette allerede i forstudiefasen. Hvis så hadde vært gjort, kunne vi muligens fått implementert alle A-krav på et tidligere stadie og på den måten mest sannsynlig vært foran tidsplanen ved dette tidspunktet.

Etter kravet (KB502) og prosjektplanen ble reguleringssystemet bygget for vinkelregulering, med unntak av yaw-aksen. Under implementering kom vi frem til at systemet vi hadde laget var ustabil. Derfor ble det videre i prosjektet brukt ressurser på å finne hovedårsaken og løsningen på dette.

Løsningen vi kom frem til var rotasjonsregulering i tillegg til vinkelregulering, og med denne løsningen fikk vi et fartøy som er flyvbart for de som har erfaring med å fly vanlige radiostyrte helikoptere. Vi kom frem til at den delen av løsningen som har med vinkelregulering å gjøre oppfører seg korrekt nok til å godkjenne de B-kravene som gikk på dette. Det andre målet ble dermed oppfylt med forbehold om at oppførselen enda ikke er optimal med de reguleringsparameterene vi har prøvd.

Vi kommer til å bruke den resterende tiden mellom innleveringen og fremføringen til å jobbe med finjusteringer av regulatorene for å heve kvaliteten på stabiliteten og dermed gjøre det enklere å fly.

1.2 Kostnader

I utgangspunktet hadde vi to budsjetter, et for komponenter og det andre for administrativt. Senere satt vi de i inn i samme budsjett[3] for enkelthets skyld. Etter avtale med Simicon kunne vi legge ut for deler og annet for inntil 7000,- som de skal tilbakebetale når produktet er levert. Gruppen ga derfor 1000,- hver til økonomi-ansvarlig som har tatt ansvar for innkjøp av deler. Denne ordningen har fungert utmerket.

Et lite problem vi støtte på under bestilling var at noen av komponentene var utsolgt. Det var derfor stor fordel at de valgte komponentene var tilgjengelig fra andre nettbutikker til samme pris.

De totale kostnadene ble noe mindre enn det som var budsjettert, dette til tross for at det var komponenter som måtte kjøpes ekstra grunnet skader. Årsaken til at dette var at en del komponenter ble kjøpt og sendt enkeltvis fra utenlandske nettbutikker, noe som førte til at vi unngikk fortolling da de enkelte forsendingene var under gitte importgrenser.

1.3 Evalueringer av produktet

Noen stikkord som beskriver produktet er

- Rimelig
- Enkelt (i fysisk oppbygging)
- Rotasjonsregulering - gjør at fartøyet motstår endringer i vinkelhastighet, og gjør det mulig å fly som et vanlig helikopter
- Vinkelregulering – skal gjøre fartøyet helt stabilt, er implementert og oppfører seg i hovedsak som det skal, men er ikke stabilt ennå

1.3.1 Gyroregulert modus

Tre regulatorer regulerer vinkelhastighetene i alle de tre aksene. Regulatorene får referanser direkte fra radioen. Med denne reguleringen implementert ble oppførselen til fartøyet omtrent slik som hos vanlige fjernstyrte helikoptre. Selv om denne modusen ikke er så stabil som planen er for fartøyet, er den en viktig del av den vinkelregulerte modusen.

De PID verdiene vi har kommet frem til ved testing er listet i tabell 3.

	P	I	D
P	-0.9	0	0
Q	-0.9	0	0
R	1.5	0	0

Tabell 3: Gyro PIDene

Merk: Pitch og Roll har negative verdier fordi positive målinger i vinkelhastighet skal *motvirkes*.

1.3.2 Vinkelregulert modus

Den vinkelregulerte modusen inneholder ytterligere to PID-regulatorer i tillegg til de tre fra gyroreguleringen, dermed har vi tilsammen fem PID-regulatorer i denne modusen. De to PID-ene fra vinkelreguleringen er plassert mellom radioen og gyroreguleringen.

Vinkelregulatorerne til Roll og Pitch får referanser fra radioen, og tilbakekoblingen til disse kommer fra en vinkelestimator som gir vinkelhastighet-referanser til gyroregulatorerne for P og Q.

Det er ingen regulering av vinkel for Yaw, kun regulering av vinkelhastighet.

De så langt beste PID-verdiene til vinkelregulatorerne er oppgitt i tabell 4. Stabiliteten er ennå ikke på ønsket nivå, i henhold til vårt overordnede mål. Slik det er nevnt i Prosjektresultat-avsnittet ble tiden for knapp, og vi hadde derfor ikke mer tid til å teste og tune parametere til vinkeregulatorerne.

	P	I	D
Roll	1.5	0.2	0
Pitch	1.5	0.2	0

Tabell 4: Vinkel PIDene

Merk: Systemet har ikke den beste stabilitet ennå, og disse verdiene er å anse som midlertidige.

1.3.3 Simulering

Simuleringen gav oss et dypere innblikk i fartøyets oppførsel og dynamiske egenskaper. Målet med simuleringen var å finne tilnærmede PID verdier som skulle gjøre selve PID tuningen enklere. Problemet var at det ikke var tid eller kompetanse til å implementere alle de dynamiske egenskapene til fartøyet i simuleringen, slik som avanserte aerodynamiske effekter. Resultatet av dette var at det var en god del enklere å få det simulerte fartøyet stabilt enn det det var å få det virkelige systemet stabilt. Simuleringen antydte et stabilt fartøy over et større område med PID verdier, men det viste seg ved testing å være anderledes i virkeligheten.

1.3.4 Dragonfly Testbench GUI

Vårt egetutviklede testbenk-software og GUI, Dragonfly Testbench, viste seg å være et svært nyttig verktøy når vi kom til tuning av reguleringssystemene, til endring av filtre og andre faktorer. Ved å bruke dette programmet kunne en enkelt oppdatere parametere, resette sensorer og lese ut data, uten å måtte endre kildekoden. Dette førte til uvurderlige tidsbesparelser.

1.3.5 Generell brukernytte av produktet

Slik produktet er nå, kan det i hovedsak sees på som en utviklingsmodul. Mikrokontrolleren har fortsatt mye ledig kapasitet til å videreutvikle programvaren, samt flere innganger og utganger ledige slik at andre sensorer og utstyr kan kobles til. Ellers er fartøyet flyvbart med en tilkoblet fungerende kameramodul slik at fartøyet kan brukes til filming, selv om bildekvaliteten med det kameraet vi benytter ikke er spesielt god med tanke på oppløsning og frekvens.

2 Prosjektgjennomføring

2.1 Arbeidsmetoder

Arbeidsmetodikken vi har tatt utgangspunkt i er den iterative metoden Rational Unified Process (RUP). Vi har tillatt oss å gjøre noen tilpasninger da vi har et flerfaglig prosjekt og RUP i hovedsak er beregnet for programvareutvikling.

Dette var en arbeidsmetode som viste seg å være svært passende for vårt prosjekt da vi med en iterativ tilnærming og fokus på analyse og design kunne fokusere på å implementere element for element og ha god oversikt over fremdriften. Den ga mulighet til å kunne kontrollere det nødvendige tidsforbruket på en enkel og effektiv måte.

Vi definerte hele forprosjektet som én iterasjon, noe som er litt misvisende da den i RUP-terminologi omhandlet både *Inception* og *Elaboration*. Videre var iterasjonene 2 til 7 *Construction*-iterasjoner hvor vi ved slutten av hver iterasjon hadde utarbeidet artefakter som brakte oss nærmere vårt ferdige produkt. *Transition*-fasen, i den grad vi kan si å ha en slik en, er iterasjon 9 hvor vi i dette prosjektet ferdigstiller alle dokumenter, utfører slutt-tester, har den avsluttende presentasjonen og avleverer produkt til kunden.

Iterasjon 8 ville også vært en konstruksjonsfase med fokus på C og D-krav, men ble nedprioritert da vi så behovet for ekstra iterasjoner knyttet mot A og B-kravene.

Takket være denne arbeidsmetodikken avdekket vi i god tid at vi kom til å måtte gå noe utenfor de opprinnelige planene, og dette tillot oss å enkelt gjøre slike tilpasninger.

2.2 Samarbeid inn- og utad

2.2.1 Samarbeid og miljø i prosjektgruppen

Prosjektgruppen mener at det har vært et bevisst forhold til samarbeid og miljø i gruppen. Med en gruppe bestående av seks ganske forskjellige personer med ulike bakgrunner, kompetanse og fagområder, har gruppen hatt et bredt spekter av forskjellige egenskaper, og dette har vært positivt for samarbeid og miljø i gruppen. Alt i alt har vi opplevd et veldig bra samarbeid, og alle er fornøyde med den generelle arbeidsfordelingen.

2.2.2 Oppdragsgiver, ekstern veileder og sensor

Simicon AS

Ragnvald Otterlei

Prosjektgruppen mener samarbeidet med ekstern veileder fungerte særdeles bra. Helt fra første møte har prosjektgruppen blitt godt mottatt, og det har vært en god tone. Vi har hatt møter hvor veileder har stilt seg til rådighet til å svare på spørsmål, og også gitt oss gode og nyttige tilbakemeldinger.

2.2.3 Intern veileder

Dag Samuelsen

Prosjektgruppen mener samarbeidet med intern veileder har fungert veldig bra. Det har vært holdt ukentlige møter hvor det har blitt gitt konstruktive råd og tilbakemeldinger på det arbeidet som er gjort, og det har vært mulighet for å stille spørsmål og få gode svar.

2.2.4 Sykdom og fravær

I løpet av prosjektperioden har det vært lite sykdom og tilnærmet null fravær. Vi tror det gode samarbeidet og miljøet i gruppen har hatt positiv innvirkning når det gjelder dette.

2.3 Prosjektadministrasjonen

Tidlig i prosjektet fant vi ut at vi trengte noen verktøy for tidsplanlegging, timeføring og dokumenthåndtering. Vi kom frem til at datastudenten på gruppen kunne sette opp et prosjektstyringsverktøy på nett som inneholdt alt det vi trengte, og dette ble lagt inn som en passordbeskyttet del av nettsiden vår. Verktøyet har vært til stor hjelp og har blant annet hjulpet oss med:

- Holde orden på aktiviteter
- Direkte oppdateringer av Gantt-diagram
- Timelogging
- Wiki, forum – deling av informasjon og nyheter
- Lagring og back-up av dokumenter

I tillegg har vi brukt Dropbox og Google Docs for å forenkle samarbeidet ved deling av informasjon og arbeid med dokumenter.

2.3.1 Ansvarsfordeling

For å forsøke å opprettholde en jevn arbeidsbelastning har hver enkelt person fått ansvar for bestemte aktiviteter eller områder, og det var da tenkt at denne skulle være hovedressurs, men at gjennomføring av oppgavene kunne delegeres videre til ledige ressurser ved behov. Ved videredelegering har ansvarshavende forblitt samme person, så denne var fortsatt ansvarlig for å sjekke at oppgaven faktisk ble gjort. Denne metoden har fungert greit, men arbeidsbelastningen ble allikevel noe ujevn, og videredelegering og ansvarsfordeling kunne kanskje vært gjort bedre.

2.3.2 Møter og rapportering

- Faste møter og fast rapportering har fungert veldig bra
- Rapportering av tester i testspesifikasjon har gått bra
- Rapportering av tester gjort for research har ofte kommet i etterkant der det har vist seg at de

kunne være av interesse å kunne gjenskape. I etterkant ser vi at slike tester også burde vært rapportert når de gjøres.

2.4 Timebruk

Dette delkapittelet inneholder en oversikt over timeforbruket til hvert av medlemmene totalt, timebruk på hver aktivitet, og på hver sak.

Forskjellen i tidsbruk per medlem kan kommenteres da det er relativt store forskjeller. Noen har brukt noe mere tid generelt, og enkelte har hatt færre/flere andre fag og dermed mulighet til å jobbe mer/mindre. Men noe forskjell kommer også av at noen medlemmer begynte å logge timer aktivt før andre, og det har også vært litt uklarheter om hva man skulle logge timer for og ikke når det gjaldt research og lesing. Vi på gruppen har vært enige om kun å logge tid for effektivt arbeid, så vi har forsøkt å være strenge på timeføring.

Det vil også logges timer i perioden frem til og med den avsluttende presentasjonen. Timelistene er eksportert fra vår interne prosjektside 2011-05-27.

Aktiviteter som tok mer tid en planlagt

- Reguleringsstruktur
- Radioavlesing
- Minnehåndtering
- Matematisk modell (kropp og driftssystem)
- Vanskeligheter med tuning

2.4.1 Vurdering av timesrapportene

I starten var det ikke opprettet godt nok definerte aktivitetsposter på den interne prosjektsiden, og en mengde timer ble feil-logget i den sammenheng. I hovedsak kan man anse posten "Annet" som analyse-relatert arbeid.

Vi ser også at rapportene bekrefter våre inntrykk og intensjoner om et jevnt arbeidstrykk gjennom prosjektperioden, med noen unntak i forkant av presentasjonene hvor det naturlig har vært lagt inn litt ekstra innsats. Det kan være noe missvisende at posten "Utvikling" også har blitt benyttet for utvikling av presentasjonene, samt i starten utvikling av dokumenter før vi fikk en egen post for nettopp "Dokumentasjon" i Februar.

2.4.2 Timesrapport per medlem

Medlem	2010-10	2010-11	2010-12	2011-1	2011-2	2011-3	2011-4	2011-5	Totalt
Vegard Torkelsen	4.00	15.00	34.00	79.00	67.50	86.00	57.25	100.75	443.50
Luke Carambot		9.00	81.50	91.00	66.00	77.00	73.00	111.50	509.00
Trine Lindberg	10.00		49.00	55.50	32.00	67.00	59.00	82.50	355.00
Herå Rørvik		8.00	25.00	56.90	36.50	59.50	58.00	77.50	321.40
Patrick Bjørum			15.00	75.50	36.50	57.00	40.00	84.50	308.50
Michael Odden	9.00	10.50	14.00	78.50	43.50	64.00	55.50	70.50	345.50
Totalt	23.00	42.50	218.50	436.40	282.00	410.50	342.75	527.25	2282.90

Tabell 5: Timer per medlem

2.4.3 Timesrapport per aktivitet

Aktivitet	2010-10	2010-11	2010-12	2011-1	2011-2	2011-3	2011-4	2011-5	Totalt
Design		7.00	8.00	33.40	17.50	60.50	21.00	20.50	167.90
Utvikling	6.00	11.50	106.50	98.50	85.50	77.50	60.50	141.00	587.00
Testing				49.50	66.50	13.00	16.50	91.00	236.50
Feilsøking			3.50		30.50	9.00		23.00	66.00
Analyse		5.00		5.00	1.00	33.00	10.00	8.00	62.00
Møtevirksomhet	7.00	5.00	14.50	28.00	40.00	34.00	24.75	37.00	190.25
Annet	10.00	14.00	82.00	216.00	27.50	139.00	73.00	9.00	570.50
Administrasjon			4.00	6.00	6.00	10.50	17.00	35.25	78.75
Dokumentasjon					7.50	34.00	120.00	162.50	324.00
Totalt	23.00	42.50	218.50	436.40	282.00	410.50	342.75	527.25	2282.90

Tabell 6: Timer per aktivitet

2.4.4 Timesrapport per sak

Utskriften har sitt opphav i prosjekt-websiden og dermed er nummereringen av IDene ulikt de i prosjektplan_4.0.pdf. Se [3] for kryssreferanser.

Sak	2010-10	2010-11	2010-12	2011-1	2011-2	2011-3	2011-4	2011-5	Totalt
Aktivitet #3: Undersøkelse rundt matematisk modell		12.00		6.00	1.00	52.50			71.50
Funksjon #4: Oppsett av prosjektstyrings-side	6.00		2.00						8.00
Aktivitet #5: Kravspesifikasjon		10.00	44.50	7.00			3.00		64.50
Aktivitet #8: Interne møter	5.00	7.00	5.00	14.00	14.00	20.00	22.00	25.50	112.50
Aktivitet #12: Redmine-veiledning		1.50	4.00	1.00					6.50
Aktivitet #14: Forstudierapport.	2.00	2.00	8.00						12.00
Aktivitet #15: Prosjektorganisering				37.40	3.00	3.50	2.00	22.00	67.90
Aktivitet #16: Møter		3.00	10.50	14.00	27.00	13.50	3.75	13.00	84.75
Aktivitet #20: Prosjektplan			5.00						5.00
Aktivitet #21: Oppretting av maler, standardisering av dokumenter eller kode		3.00	6.50	13.50					23.00
Aktivitet #22: Økonomi			13.00						13.00
Aktivitet #23: Hjemmeside				5.00					5.00
Aktivitet #25: Research				5.00		2.00			7.00
Aktivitet #28: Prosjektplan, første versjon		4.00	43.50	21.00					68.50
Aktivitet #30: Presentasjon 1			8.00	81.50					89.50

Sak	2010-10	2010-11	2010-12	2011-1	2011-2	2011-3	2011-4	2011-5	Totalt
Aktivitet #32: Presentasjon 3 (Siste presentasjon!)								12.50	12.50
Aktivitet #35: Testspesifikasjon			33.00	24.00	1.00		4.00		62.00
Funksjon #36: Hindre forelder-saker i å vises på Gantt-diagram				1.00					1.00
Aktivitet #37: Prosjektbeskrivelse/idèdokument			5.00						5.00
Aktivitet #39: Prosjektplakat								4.50	4.50
Aktivitet #40: Oppfølgingsdokumenter				2.00	3.50	4.50	1.50	2.50	14.00
Aktivitet #41: Signalbehandling					5.00				5.00
Aktivitet #42: Reguleringsystem, research				4.00	25.50	39.50	7.00	1.00	77.00
Aktivitet #44: Multikopterteori				2.50	2.00		0.50		5.00
Aktivitet #46: Forstudie	10.00		9.00						19.00
Aktivitet #47: Etteranalyse								18.00	18.00
Aktivitet #50: Teknologidokument			3.00						3.00
Aktivitet #53: Reguleringsystem								7.00	7.00
Aktivitet #54: Implementasjon				2.00					2.00
Aktivitet #56: Prototype til testing				60.00	2.00				62.00
Aktivitet #57: Testbenk				1.50	2.00				3.50
Aktivitet #58: Ferdigstilling til innlevering								8.00	8.00

Sak	2010-10	2010-11	2010-12	2011-1	2011-2	2011-3	2011-4	2011-5	Totalt
Aktivitet #59: Testing				39.50	32.50	5.00	22.00	7.50	106.50
Aktivitet #60: Testplan			18.50	0.50					19.00
Aktivitet #61: Testrapport				2.50		3.00			5.50
Aktivitet #62: Testbenk, design				8.50	2.00				10.50
Aktivitet #64: Design av programvare for uregulert fartøy				11.00	2.00				13.00
Aktivitet #65: Design av programvare for regulert fartøy					4.00	13.00	5.00		22.00
Aktivitet #66: Utvikling av programvare for uregulert fartøy				13.00	23.00				36.00
Aktivitet #67: Utvikling av programvare for regulert fartøy						39.00		4.00	43.00
Aktivitet #68: Testlogg				4.50	4.50	5.00			14.00
Aktivitet #69: Iterasjonsrapport 2				2.00					2.00
Aktivitet #70: Iterasjonsrapport 3					3.00				3.00
Aktivitet #71: Iterasjonsrapport 4						1.00			1.00
Aktivitet #72: Iterasjonsrapport 5							1.50		1.50
Aktivitet #73: Iterasjonsrapport 6								1.50	1.50
Aktivitet #74: Iterasjonsrapport 7								2.00	2.00
Aktivitet #77: Design av koblingsskjema				3.00					3.00
Aktivitet #79: Feilsøking og forbedring								6.00	6.00

Sak	2010-10	2010-11	2010-12	2011-1	2011-2	2011-3	2011-4	2011-5	Totalt
Aktivitet #80: Design av matematisk modell				6.00	19.00	42.50			67.50
Aktivitet #81: Design av signalbehandling						12.50			12.50
Aktivitet #83: Matematisk modell				2.00		11.00	32.50		45.50
Aktivitet #84: Simulering av reguleringsystem							42.00	34.75	76.75
Aktivitet #85: Implementering av signalbehandling						4.00			4.00
Aktivitet #86: Implementering av reguleringsystem						18.00			18.00
Aktivitet #87: Testing av prototype				8.00	30.00				38.00
Aktivitet #88: Testing av reguleringsystemet								73.50	73.50
Aktivitet #91: Prototype				8.00	6.50				14.50
Aktivitet #92: Design av reguleringsystem						13.00		0.50	13.50
Aktivitet #93: Testdesign				10.50					10.50
Aktivitet #94: Oppsett av tester				2.00					2.00
Aktivitet #96: Oppdatere prosjektplanen				3.00					3.00
Aktivitet #100: Regnskap					1.00				1.00
Aktivitet #102: Iterasjonsplan iterasjon 2				2.00					2.00
Aktivitet #103: Iterasjonsplan iterasjon 3				4.00					4.00
Aktivitet #104: Iterasjonsplan iterasjon 4					2.00	2.00			4.00

Sak	2010-10	2010-11	2010-12	2011-1	2011-2	2011-3	2011-4	2011-5	Totalt
Aktivitet #105: Iterasjonsplan iterasjon 5							1.00		1.00
Aktivitet #106: Iterasjonsplan iterasjon 6							2.00		2.00
Aktivitet #107: Iterasjonsplan iterasjon 7								1.50	1.50
Funksjon #108: Enkle visningsbilder for blogg-innlegg							1.50		1.50
Aktivitet #109: Programvare for testbenk				4.00	61.50		7.00	2.50	75.00
Aktivitet #110: Presentasjon 2						28.50	65.50	2.00	96.00
Aktivitet #111: Utbedre signalproblemer					5.00	7.00			12.00
Aktivitet #112: Designe testbenk-utvidelse for regulert fartøy						14.00			14.00
Aktivitet #113: Utbedre testbenk for uregulert fartøy						10.00			10.00
Aktivitet #114: Design av nye tester						5.00			5.00
Aktivitet #115: Ombygging for lavere COG							2.00		2.00
Aktivitet #116: Vedlikehold og oppdateringer av ProsjektWeb						7.00			7.00
Aktivitet #117: Oppdatere kravspesifikasjon						0.50	3.00		3.50
Aktivitet #118: Oppdatere testspesifikasjon						9.00			9.00
Aktivitet #119: Designdokument reguleringssystem						4.00		2.00	6.00
Aktivitet #120: Vi deler informasjon						13.00		4.00	17.00
Aktivitet #121: Unit-tester						4.00			4.00

Sak	2010-10	2010-11	2010-12	2011-1	2011-2	2011-3	2011-4	2011-5	Totalt
Aktivitet #122: Sensor filtrering						4.00	7.00	3.00	14.00
Aktivitet #123: Redesigne håndtering av radio-input							5.00	9.00	14.00
Aktivitet #124: Ferdigstilling av dokumenter							80.00		80.00
Aktivitet #125: Implementering av reguleringsystem							7.50		7.50
Aktivitet #126: Implementere redesignet radio-input							2.00	24.00	26.00
Aktivitet #127: Parameterfremstilling for matematisk modell							6.50		6.50
Aktivitet #128: Implementering av testbenk-protokoll i regulert fartøy								12.00	12.00
Aktivitet #129: Forbedring av matematisk modell							5.00	5.00	10.00
Aktivitet #130: Forbedringer av programvare ifm ferdigstilling								37.00	37.00
Aktivitet #131: Lagring av tuning-data mellom omstarter								2.00	2.00
Aktivitet #132: Forbedring av testbenk								14.50	14.50
Funksjon #133: Påbygging av reguleringsystemet								10.00	10.00
Aktivitet #134: Iterasjonsplan iterasjon 9								1.00	1.00
Aktivitet #135: Tuning og testing av reguleringsystem								29.50	29.50
Aktivitet #136: Ferdigstilling av dokumenter								110.50	110.50
Aktivitet #137: Testing av kameramodul								6.00	6.00
Aktivitet #138: Brukermanual								1.00	1.00

Sak	2010-10	2010-11	2010-12	2011-1	2011-2	2011-3	2011-4	2011-5	Totalt
Aktivitet #139: Implementasjonsdokument, programvare								6.00	6.00
Aktivitet #140: Iterasjonsrapport 9								1.00	1.00
Totalt	23.00	42.50	218.50	436.40	282.00	410.50	342.75	527.25	2282.90

Tabell 7: Timer per sak

Konklusjoner

Prosjektet er nå fullført og vi er veldig fornøyde og stolte av produktet vårt og det vi har fått til. Samtidig har vi tilegnet oss store mengder nyttig kompetanse og erfaring som vi kan ta med oss videre ut i arbeidslivet. Vi har utviklet oss mye både med tanke på faglig kompetanse og kjennskap til generelt prosjektarbeid og samarbeid. Prosjektet har ført oss langt utenfor våre pensum og fagområder, noe som har vist seg å være både utfordrende, spennende og lærerikt.

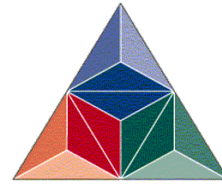
Vi har nådd våre mål og utviklet et flyvbart fartøy med fungerende reguleringsystem og med gode muligheter og kapasitet for videreutvikling.

Referanser

[1] kravspesifikasjon_5.0.pdf

[2] testspesifikasjon_3.0.pdf

[3] id-oversikt_1.0.pdf



HØGSKOLEN
i Buskerud

Oppgavetittel: Hovedprosjekt - Forprosjekt - v4.0		
Fag(nr./navn): SFHO 3200 / Prosjektstyring med hovedprosjekt		
Gruppemedlemmer, signatur:		
_____ Vegard Torkelsen	_____ Michael Odden	_____ Herå Rørvik
_____ Trine Lindberg	_____ Luke Carambot	_____ Patrick Bjørum
Intern veileder: Dag Samuelsen Intern sensor: Olaf Hallan Graven Ekstern veileder og sensor: Ragnvald Otterlei		
Dato: 2011-05-22		

Innholdsfortegnelse

Dokumenthistorie.....	1
Innledning.....	2
1 Kort beskrivelse av prosjektet.....	2
2 Prosjektgruppen.....	2
3 Hjemmeside.....	4
4 Oppdragsgiver.....	5
5 Sensorer og veiledere.....	5
5.1 Intern veileder.....	5
5.2 Intern sensor.....	5
5.3 Ekstern veileder og sensor.....	6
6 Målet med prosjektet.....	6
7 Prosjektomgivelser og relaterte prosjekter.....	7
8 Beskrivelse av alternative løsninger.....	7
8.1 Reguleringsmetoder.....	8
9 Begrensninger og utfordringer.....	9
10 Risiko.....	9
11 Utviklingsmetodikk.....	10
12 Aktiviteter.....	10
13 Tidsplaner.....	12
14 Detaljert plan, resurser og aktiviteter.....	12
Referanser.....	15
Vedlegg 1.....	16

Dokumenthistorie

Versjon	Beskrivelse	Når
1.0	Opprettet dokument med forslag til overskrifter, kopiert over relevant info fra idèdokument. Lagt inn innholdet fra samarbeidsdokumentet(google docs). Rettlest, fjernet skrivefeil	2010-10-11
2.0	Flyttet kontaktinfo til under beskrivelse av gruppe medlemmene Endret forsiden til å inneholde signaturfelt Oppdatert risikomomenter Korrigert beskrivelse av bedrift Korrigert prioriteter Oppdater tidsplan og gantt, endret dato for forprosjekt og milepæl 2.	2010-11-23
3.0	Oppdatert dokumentmal og korrekturlest	2011-04-05
4.0	Oppdateringer til dokumentmal, pirking	2011-05-22

Tabell 1: Dokumenthistorie

Innledning

Dette dokumentet gir en grov oversikt over prosjektet. Det inneholder også informasjon om prosjektgruppen og oppdragsgiver.

1 Kort beskrivelse av prosjektet

Prosjektet går ut på å utvikle og konstruere et fire rotors helikopter, heretter referert til som fartøyet, som skal være stabilt og lett å fly via en fjernkontroll. Fartøyet skal stabiliseres ved hjelp av et reguleringsystem i tre akser, roll, pitch og yaw, samt ved hjelp av nødvendige sensorer som gyroer og akselerometere. Reguleringen og stabiliseringen skal gjøres via en mikrokontroller.



Figur 1: [1]



Figur 2: [2]

2 Prosjektgruppen



Vegard Torkelsen - Elektroingenør - Kybernetikk - Prosjektleder.

Kort beskrivelse: 23 år, opprinnelig fra Hønefoss, men flyttet til Kongsberg etter førstegangstjenesten i 2007 for å studere ved HiBu. Høsten 2008 hadde jeg et semester ved University of Woollongong i Australia. Etter endt bachelor ønsker jeg å ta en master i kybernetikk ved enten UiS eller NTNU. Fritidssysler er bl.a gitar, tv-spill og surfing.

Styrker: Blir veldig engasjert i prosjektene jeg er del av og jeg er ikke redd for å ta på meg en utfordring.

Svakheter: Kan fort begynne å stresse, slik at det kan "gå litt fort i svingene". Kan også bli noe utålmodig når det kommer til ting jeg ikke har noe makt over.

Kontaktinfo: tlf: 9953582, e-post: vegard_3@hotmail.com

Michael Odden - Dataingeniør - Simulering og spillutvikling



Kort beskrivelse: 25 år, opprinnelig fra Sarpsborg men bor nå i Drammen. Studerer til dataingeniør (Simulering & Spillutvikling) ved HiBu. Har fra før fagbrev i IKT-driftsfag med lærlingtid ved Luftforsvarets Skolesenter Kjevik. Ønsker etter endt bachelor å jobbe innen serious gaming med vekt på simulering.

Styrker: Har god forståelse for programmering generelt, webutvikling og utvikling av desktop-programvare spesielt og mye erfaring med prosjektarbeid i arbeidssammenheng. Har også god forståelse for

matematikk og fysikk.

Svakheter: Setter noen ganger for store forventninger til meg selv og har noe lett for å ta på meg for mange oppgaver.

Kontaktinfo: tlf: 95293737, e-post: me@michaelodden.com

Herå Rørvik - Elektroingeniør - Kybernetikk



Kort beskrivelse: 32 år, opprinnelig fra Iran. Kom til Norge i 2000 og bor nå i Drammen med min mann og to barn. Har hjelpepleier fagbrev, og har jobbet på sykehuset i Drammen siden 2006. Jobber der fremdeles ved siden av studiene.

Styrker: Trives med å jobbe i gruppe og ønsker alltid å gjøre mitt beste.

Svakheter: Blir fort stresset når ting ikke går etter planen.

Kontaktinfo: tlf: 91517631, e-post: fratjan2001@yahoo.com

Trine Lindberg - Elektroingeniør - Kybernetikk



Kort beskrivelse: 22 år, opprinnelig fra Fredrikstad men bor nå i Vestfossen mens jeg studerer i Kongsberg. Etter endt videregående i 2007 jobbet og reiste jeg et år før jeg startet på bachelorstudiet i kybernetikk. Tenker muligens å ta en master i kybernetikk ved UiO etter endt bachelor.

Styrker: Er pliktoppfyllende og liker utfordringer. Jobber like godt i gruppe som individuelt.

Svakheter: Liker ikke gi ifra meg ting som ikke er 100%, noe som kan føre til at jeg bruker mye tid på slutførelsen av ting.

Kontaktinfo: tlf: 47077385, e-post: trine.lindberg88@gmail.com

Luke Carambot - Elektroingeniør - Mekatronikk



Kort beskrivelse: 28 år, opprinnelig fra Oregon USA, er gift og bor i Kongsberg. Var ferdig med videregående i år 2000 og har jobbet mest som vaktmester/bygger inntil jeg begynte på bachelorstudiene på Høgskolen i Buskerud i 2007.

Styrker: Kan sette meg godt inn i et oppgave. Løsningsorientert og liker å gjøre ting riktig. Jeg er veldig analytisk. Synes det er underholdende å være kreativ og designe/bygge ting. Jeg er nysgjerrig og er ikke redd for å prøve meg fram.

Svakheter: Ikke den beste i å planlegge før jeg begynner med noe. Bruker mye tid for å få ting riktig/rett. Vil gjerne utsette ting for å få det riktig gjort.

Kontaktinfo: tlf: 41045413, e-post: lukecarambot@gmail.com

Patrick Bjorum - Elektroingeniør - Mekatronikk



Kort beskrivelse: 25 år, opprinnelig fra Oslo, men bor nå i Lier. Var ferdig med videregående i 2004 og utførte førstegangstjenesten samt jobbet før jeg begynte på bachelorstudiene hos Høgskolen i Buskerud i 2007. Jeg har drevet med R/C helikopter på fritiden i 3 år.

Styrker: Har en god del teknisk innsikt i dette temaet pga. hobbyen min og klarer dermed lett å se helheten når vi gjør ting. Kan være perfeksjonist, så jeg liker at ting blir gjort ordentlig og gir meg ikke før det er slik jeg vil ha det. Er veldig sosial og liker godt å jobbe i

grupper.

Svakheter: Etersom jeg kan være perfeksjonist så kan ting ta litt lengre tid enn det burde gjort noen ganger. Hvis ting går tregt kan jeg trenge "et spark bak" for å komme i gang igjen.

Kontaktinfo: 92054151, patrick.bjorum@gmail.com

3 Hjemmeside

Vi har opprettet en hjemmeside for prosjektet:

www.projectdragonfly.no

Hjemmesiden vil være vår front for informasjon rundt prosjektet, samt tjene som inngang til vårt prosjekthåndteringsystem. Siden består dermed også av et passordbeskyttet område hvor vi deler intern informasjon.

4 Oppdragsgiver

Ragnvald Otterlei – Simicon AS

Kontor:

*Dyrmyrgata 35
3611 Kongsberg*

Post:

*PB 787
3606 Kongsberg*

epost: *info@simicon.no*

Simicon AS er et lite firma som utvikler og selger tjenester/systemer i tilknytning til ubemannede luftfartøyer.

Fra år 2009 har de jobbet innenfor maritime og arktiske anvendelser av UAS. De har utviklet et UAS system for datafangst i forhold til meteorologi og båttrafikk. De har også gjort studier og konseptutvikling på et fly med VTOL (ta av og lande vertikalt) egenskaper, dette flyet er en hybrid mellom et fly og et helikopter, slik at det bruker VTOL egenskapene til helikopteret, mens det etter letting kan bruke flyets vinger for å oppnå høy hastighet og rekkevidde.

Bedriften er privateid. De leverer multifunksjonelle produkter som kan utstyres med en rekke sensorer slik som optiske kameraer, infrarøde kameraer, ladar (laser direction and range), GPS, IMU, altimeter og mikrofoner. I tillegg tilbyr de rådgivningstjenester innen UAS-anvendelser. [7]

5 Sensorer og veiledere

5.1 Intern veileder

Dag Samuelsen

E-post: dag.samuelsen@hibu.no

Mobil: 92058406

5.2 Intern sensor

Olaf Hallan Graven

E-post: Olaf.Hallan.Graven@hibu.no

Telefon: 32869556

5.3 Ekstern veileder og sensor

Ragnvald Otterlei

E-post: ragnvald@simicon.no

Telefon: 32728370 / 92030141

6 Målet med prosjektet

Målet med dette prosjektet er å demonstrere et fartøy som flyr stabilt, og som er enkelt å fly via en fjernkontroll.

Fartøyet skal stabiliseres slik at det skal være enkelt å lette, fly og lande. Fartøyet skal også utstyres med en kameramodul slik at man kan fly selv om fartøyet er utenfor syns-rekkevidde. Det ferdige fartøyet skal kunne brukes til senere utvidelser uten større inngrep i designet.

Prioritet 1

Vi må oppnå kommunikasjon mellom kontrollen, mikro-kontrolleren og motorene slik at fartøyet kan fly.

Prioritet 2

Fartøyet skal reguleres i pitch, roll og yaw, slik at det kan hovre. Fartøyet skal kunne fly i "autopilot", slik at den kan hovre med liten input fra fjernkontrollen, denne "autopiloten" skal kunne skrues av og på. Med hover, mener vi at fartøyet holder seg i et visst område, slik at den ikke driver vekk. Vi må også oppnå kommunikasjon mellom fartøyets kamera og bakkestasjon.

Prioritet 3

Det er ønskelig at fartøyet har en fail-safe funksjon, slik at fartøyet lander automatisk hvis kommunikasjonen mellom kontrollen og fartøyet blir brutt.

Det er også ønskelig å utvikle en metode for å advare brukeren når batteriet når et gitt minimumsnivå, eller hvis batteriet når et enda lavere batterinivå, vil fartøyet lande automatisk.

Prioritet 4

Fartøyet bør selv kunne regulere høyden slik at man kan øke stabiliteten til fartøyet, spesielt når det hovrer.

7 Prosjektomgivelser og relaterte prosjekter

Prosjektet vårt er ikke unikt siden det er laget flere lignende fartøyer før. Et av de mest spennende eksemplene er multirotor-helikopter som er utviklet ved University of Pennsylvania. De har utviklet et fartøy som kan gjøre en rekke ekstremt komplekse bevegelser, slik som saltoer, fly igjennom vinduer og andre hindringer, lande på vertikale og nær vertikale vegger vha borrelås, samt at den også er ekstremt stabil og klarer å gjenoprette sin stabilitet selv etter at den blir kastet opp i luften. Dette prosjektet foregår i UoP's GRASP lab[3], og blir styrt vha en rekke kameraer inne i labben. Fartøyet er derfor kun funksjonabel inne i selve testområdet i labben.

NASA har også opprettet et forskningsteam for studenter som har arbeidet med å utvikle et multirotor-helikopter som har som mål å kunne arbeide i sverm[6]. Det vil si at det skal være mulig å operere med flere fartøyer i luften samtidig, hvor disse fartøyene skal kunne kommunisere sammen ved å utveksle bl.a data om sin posisjon for å unngå kollisjon.

Det har også blitt utviklet en rekke plattformer for kommersiell bruk, et godt eksempel på dette er md4 serien som er utviklet av microdrones[4]. Denne brukes i hovedsak til overvåkning for å enkelt kunne dekke større områder enn man kunne gjort fra bakken. Den har også blitt brukt for å gi brannpersonell bedre oversikt over ulykkessteder.

Mindre og billigere utgaver har også blitt utviklet, som f.eks AR.DRONE[5]. Dette er i hovedsak et "leketøy" og har med sin lave flytid på 12 minutter og sin lave flyhøyde på 6m ingen god praktisk bruk. Det som er spesielt med denne er at den kan kobles opp til iPhone eller iPad og deretter styres direkte fra disse.

8 Beskrivelse av alternative løsninger

- **Bruke GPS til å identifisere fartøyets posisjon i X, Y, og Z.**

Fordeler: Er passelig tilgjengelig og gir absolutt posisjon

Ulemper: Unøyaktig i seg selv, spesielt i høyde målinger.

- **Lage en referanse GPS for å minske unøyaktigheten til GPSen på fartøyet.**

Fordeler: Unøyaktigheten til GPS blir løst og blir veldig nøyaktig. GPS signalet er tilgjengelig i det fleste steder. Gir relativ posisjon i forhold til en bakkestasjon.

Ulemper: Krever en mikroprosessor for å differensialvisere GPS signalene. Krever GPS mottakere på både fartøyet og en bakkestasjon. Krever ekstra kommunikasjonstyr mellom fartøyet og bakkestasjonen.

- **Bruke bevegelses-sensorer til å kunne regne ut fartøyets vinkel i roll, pitch og yaw. (akselerometere og gyro)**

Fordeler: Enkelt å bruke. Kan integrere for å regne ut relativt hastighet og posisjon. Krever ikke absolutt posisjon. Ganske nøyaktig

Ulemper: Spesielt gyroen kan “flyte” og dermed ikke gi riktig måling. Kan ikke regulere posisjon med nøyaktighet på grunn av unøyaktighet i målinger. Kan ikke regne ut høyden hvis bakken ikke er vannrett.

- **Bruke altimeter for å identifisere fartøyets høyde.**

Fordeler: IR sensor vil gi svært nøyaktige målinger ved lave høyder. Ved bruk av barometer har den i praksis ingen maks høyde.

Ulemper: IR sensoren har bare en rekkevide på 5-6m. Barometerets målinger kan variere kraftig av lokale metrologiske variasjoner, siden den måler trykkforskjeller i luften. Barometerets målinger regnes ut i meter over havet, mens IR sensorens målinger gir meter over bakken.

- **Bruke kamera og et bilde-gjennkjenningsprogram for å identifisere fartøyets posisjon i X,Y og Z.**

Fordeler: Kan være svært nøyaktig hvis systemet er sofistikert nok.

Ulemper: Kan bli vanskelig å bruke siden fartøyet er beregnet for utendørsbruk og det kan bli vanskelig for systemet å identifisere fartøyet når avstanden mellom fartøyet og kameraet øker. Det krever også svært kompleks kode.

8.1 Reguleringsmetoder

- **Direkte regulering utifra målte verdier (regulering av det som er målt):**

Fjernkontrollen gir mål-verdier som blir sammenlignet med de målte verdiene, dette gir en feil som regulatoren styres etter. For eksempel, om fjernkontrollen er på 50% fart i x retningen, og akselerometeret måler 20% vil regulatoren styre etter en 30% feil. Regulatoren ville skape forandringer i det fysiske systemet for å få 50% akselerasjon i x retningen.

Fordeler: Krever ikke kjente absoluttverdier. Enklere å bygge.

Ulemper: Ikke så enkelt å anvende til andre styringsmåter. Uønskede påvirkninger ville komme i fra de ukjente ytre påvirkninger.

- **Modellere systemet(matematisk) i rommet for å få ideelle bevegelser å regulere mot:**

I stedet for at fjernkontrollen styrer motorene direkte og dermed påvirke bevegelsene direkte, skal den påvirke bevegelsene av det modellert systemet. Feilen mellom det modellerte og det “virkelige” gjennom forskjellige sensorer blir det reguleringssystemet styrer etter.

Fordeler: Ytre påvirkninger, som i praksis er ukjente, blir løst ved at når det modellerte systemet står i ro vil reguleringssystemet holde den fysisk i ro uansett ytre påvirkninger som for eksempel vind. Enklere å bytte styringstype, som f.eks. bytte ut fjernkontrollen med en programmert bane eller GPS veipunkter.

Ulemper: Krever at absolutt posisjon er kjent, noe som fører til mer kompleksitet. Hastigheten til regulatoren kan bli en begrensning. Om det blir store forskjeller mellom det modellerte og det fysiske systemet vil en oppdage en bestemt treghet i responsene til styringen gjennom fjernkontrollen.

9 Begrensninger og utfordringer

Hovedbegrensningene vi ser nå tidlig i prosjektet er at de ulike sensorene i vår prisklasse har en viss unøyaktighet som må taes med i betraktning. Eksempelvis vil det være mulig å benytte både IR-sensor og barometer til høydemåling, men der IR-sensoren ikke er pålitelig over visse høyder er et barometer på generell basis unøyaktig.

En annen utfordring vi møter i dette prosjektet er at all form for kommunikasjon mellom sensorer og mikrokontroller, må foregå på et dataspråk som er forståelig for mikrokontrolleren. Vi må også få mikrokontrolleren til å forstå dataen den mottar.

En annen begrensning er da selvsagt tiden. Dette prosjektet setter store krav til sporbarhet, dokumentasjon og sikre løsninger som vil kreve mye tid som igjen vil begrense våre muligheter med tanke på funksjonalitet.

Vi konkluderer med at, selv om prosjektet byr på store utfordringer, har vi kunnskap og arbeidslyst nok til å kunne gjennomføre prosjektet innen tidsfristen.

10 Risiko

Risikofaktorer i prosjektet:

Bestilte deler kommer ikke frem i tide eller er utsolgt og lignende – for å unngå dette må vi bestille i god tid og finne flere alternative leverandører.

Skade på fartøyet under testing– for å unngå lang ventetid på nye deler etter en eventuell krasj bestiller vi ekstra sett av de mest utsatte delene. Videre bør vi utvikle en fail-safe funksjon som lander fartøyet av seg selv hvis den skulle miste kommunikasjon med senderen.

Skade på personell under testing – når propellene roterer med 3-4000RPM kan mye uønsket skje. Derfor vil vi simulere mest mulig før vi i det hele tatt prøver å lette. Deretter vil fartøyet bli festet til en test-rig for skikkelig testing før vi til slutt tester den i fritt rom.

Gruppemedlemmer/veiledere blir syke eller liknende – bør tenke igjennom hva vi kan gjøre for å minimere forsinkelse av prosjektet hvis dette skulle skje.

11 Utviklingsmetodikk

RUP (Rational Unified Process) er en generell og iterativ utviklingsmodell hovedsaklig beregnet på utvikling av programvare. Vi ønsker å benytte RUP som utviklingsmodell da denne virker å også være egnet til prosjektstyring mer generelt. Denne tillater oss å forbedre prosessen etterhvert som går gjennom prosjektet i tillegg til at den legger stor vekt på leveranse av del-produkter som igjen lar oss måle fremdrift.

Vi vurderte hovedsaklig to andre utviklingsmodeller, henholdvis **Scrum** og **vannfallsmodellen**. Vannfallsmetoden kommer til kort - slik vi ser det - ved at den levner for små muligheter til tilpasninger underveis, noe som raskt vil kunne være tilfellet da dette prosjektet er av en type og størrelse vi ikke har jobbet med tidligere. Den taper også mot RUP ved at den gir en mindre klar indikasjon på hvor man er i prosjektet og man vil i verste tilfall risikere å ikke ha noen deler av produktet klart.

Når det gjaldt Scrum anså vi denne som mest egnet for rene programvare-prosjekter samt at vi oppfattet "rammene" som vanskelige å forholde seg til generelt i et slikt tverrfaglig prosjekt.

12 Aktiviteter

Aktivitetsnummer	Aktivitet	Mål
1xx	Administrasjon og prosjektstyring	
101	Prosjektorganisering	God prosjektflyt og godt samarbeid
102	Møter	Offentlig beslutninger og oppdateringer
103	Idédokument	Dokumentasjon
104	Prosjektplan	Dokumentasjon
105	Standardisering av dokumenter/kode	Fastsette standard på dokumenter og kode
106	Økonomi	Holde orden på økonomien
107	Hjemmeside	Opprettelse og oppdatering av hjemmesiden
108	Timelister	Organisering av timelister
109	Prosjektplakat	Dokumentasjon
110	Internmøter	Offentlige beslutninger og oppdateringer
111	Presentasjoner	Presentere prosjektet for sensorer og veiledere
2xx	Research	
201	Signalbehandling	Tilegne seg nødvendig kunnskap

Aktivetsnummer	Aktivitet	Mål
202	Reguleringsystem	Tilegne seg nødvendig kunnskap
203	Matematisk modellering	Tilegne seg nødvendig kunnskap
3xx	Krav/analyse	
301	Forstudie	Dokumentasjon
302	Kravspesifikasjon	Dokumentasjon
303	Analyse	Dokumentasjon
4xx	Design/teknologi	
401	Design	Stabil og solid plattform
402	Designokument	Dokumentasjon
403	Teknologidokument	Dokumentasjon
404	Programmering	Lage software
405	Bestilling av komponenter	Bestille riktige komponenter
5xx	Implementasjon	
501	Konstruksjon	Gjøre klar hver komponent
502	Prototype	Sette sammen komponentene
503	Ferdigstilling	Finjustere prototype
6xx	Test	
601	Testing	Angir om kravene blir nådd
602	Testspesifikasjon	Retningslinjer til testing
604	Testrapport	Dokumentasjon
606	Brukerveiledning	Brukerveiledning til kunde for drift og vedlikehold av fartøyet

Tabell 2: Aktiviteter

13 Tidsplaner

Gantt-diagram: se vedlegg 1.

Oppgave	Startdato	Sluttdato
Idemyldring	25.08.10	01.10.10
Forprosjekt	19.09.10	20.10.10
Milepæl 1		20.10.10
Etablere webside	25.09.10	12.11.10
Krav- og testspesifikasjon	20.10.10	14.12.10
Milepæl 2		14.12.10
Prosjektplan	01.11.10	04.01.11
Milepæl 3		04.01.11
Hovedprosjekt	29.01.11	01.05.11
Rapport	29.01.11	11.05.11
Milepæl 4		11.05.11
Design plakaten	05.05.11	20.05.11
Levering av sluttrapport	08.05.11	01.06.11

Tabell 3: Tidsplan

14 Detaljert plan, resurser og aktiviteter

Aktivitetsnr.	Beskrivelse	Avhengighet	Når	Ressurser
101	Denne aktiviteten dekker den generelle prosjektorganiseringen		Hele prosjektperioden	Alle i prosjektgruppa
102	Møter med interne veiledere og med oppdragsgiver/ekstern veileder. Møter innad i gruppa er ikke med i denne aktiviteten.	Innkalling og agenda. Tid	Etter nyttår: En gang i uka med intern veileder og hver andre eller tredje uke med ekstern veileder	Alle i prosjektgruppa, interne og eksterne veiledere
103	Idèdokument er levert i slutten av august	Må være godkjent før noen andre aktiviteter kan begynne	August 2010	Alle i prosjektgruppa
104	Prosjektplanen må hele	De fleste andre	Hele	Alle i

	tiden oppdateres	aktiviteter avhenger av prosjektplanen	prosjektperioden	prosjektgruppa
105	Oppretting av maler, standardisering av dokumenter eller kode		Starten av prosjektperioden	Michael
106	Oppretting av budsjetter, betaling av regninger, føre regnskap o.l.			Patrick
107	Oppretting og vedlikehold av hjemmesiden		Hele prosjektperioden	Michael, Webserver
108	Eksportere timelister fra hjemmesiden og lignende	107	Med fast mellomrom hele prosjektperioden	Michael, Webserver
109	Opprette en plakat med resultater fra prosjektet	103	I slutten av prosjektperioden	Alle i prosjektgruppa
110	Innadmøter av gruppemedlemmene	Innkalling og agenda	Hele prosjektperioden	Alle i prosjektgruppa
111	Alt arbeid rundt presentasjonene		Fra nov. og ut prosjektperioden	Alle i prosjektgruppa
201	Research knyttet til signalbehandling	103	Hele prosjektperioden	Patrick og Herå
202	Research knyttet til reguleringsystem	103	Hele prosjektperioden	Trine og Vegard
203	Research knyttet til matematisk modellering	103		Michael og Luke
301	Arbeid med forstudiedokument, rapporten skal inneholde en detaljert beskrivelse av prosjektet, frem til prosjektplanen er ferdig.	103	Frem til desember 2010	Alle i prosjektgruppa
302	Avklare krav med oppdragsgiver, lage dokument	103	Hele prosjektperioden	Alle, Herå er ansvarlig
303	Risikoanalyser og lignende, må oppdateres underveis		Hele prosjektperioden	Alle
401	Design av software	201, 202 og 203	Hele prosjektperioden	Michael

402	Dokument om design	401		
403	Dokument med tekniske spesifikasjoner og forklaringer	302	Hele prosjektperioden, oppdateres jevnlig	
404	Programmering	401, 201, 202 og 203	Siste halvdel av prosjektperioden	Michael
405	Bestilling av komponenter	201, 202 og 203	Siste halvdel av prosjektperioden	Patrick
501	Fysisk konstruksjon, sammensetting av delsystemer	405	Siste del av prosjektperioden	
502	Prototyping, sette sammen de ulike delene til å fungere sammen	405	Siste del av prosjektperioden	
503	Ferdigstilling, overlevering til oppdragsgiver	Alle krav i kravspesifikasjon må være møtt. 502	Når prosjektet er ferdig	
601	Testing av de ulike delene og prototype	502	Siste del av prosjektperioden	Luke
602	Opprette og oppdatere testspesifikasjon	302	Hele prosjektperioden	Luke og Herå
604	Skrive rapport etter testing	601	Etter testing	Luke
606	Opprette dokument for andre brukere	502		Alle

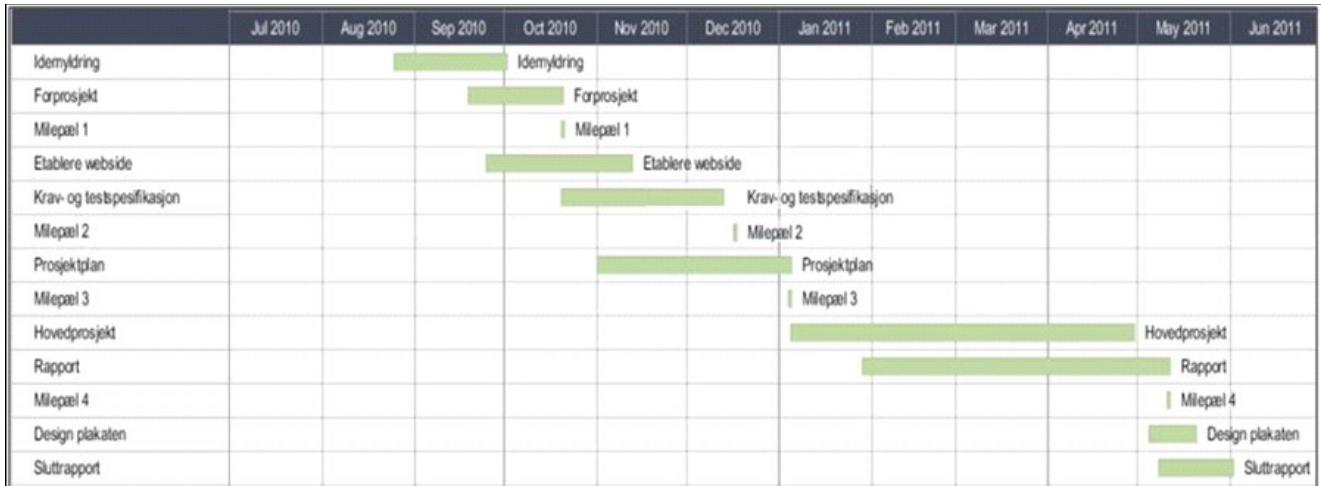
Tabell 4: Detaljert plan

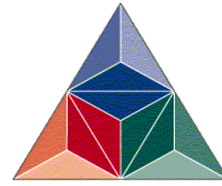
Referanser

- [1] <http://www.kkmulticopter.com/index.php/classifieds?view=annonce&id=3> (2010-10-19)
 - [2] <http://www.mictronics.de/projects/my-mikrokoetter/> (2010-10-19)
 - [3] <http://www.grasp.upenn.edu/> (2010-10-19)
 - [4] http://www.microdrones.com/en_home.php (2010-10-19)
 - [5] <http://ardrone.parrot.com/parrot-ar-drone/usa> (2010-10-19)
 - [6] <http://www.nasa.gov/centers/langley/multimedia/iotw-quadcopter.html> (2010-10-19)
 - [7] <http://simicon.no/> (2010-10-19)
-

Vedlegg 1

Gantt diagram





HØGSKOLEN
i Buskerud

Oppgavetittel: Hovedprosjekt - Prosjektplan - v4.0		
Fag(nr./navn): SFHO 3200 / Prosjektstyring med hovedprosjekt		
Gruppemedlemmer, signatur:		
_____ Vegard Torkelsen	_____ Michael Odden	_____ Herå Rørvik
_____ Trine Lindberg	_____ Luke Carambot	_____ Patrick Bjørum
Intern veileder: Dag Samuelsen Intern sensor: Olaf Hallan Graven Ekstern veileder og sensor: Ragnvald Otterlei		
Dato: 2011-05-22		

Innholdsfortegnelse

Dokumenthistorie.....	1
Definisjoner og forkortelser	1
Innledning	2
Hensikt med dokumentet.....	2
Kort beskrivelse av prosjektet.....	2
Illustrasjonsbilder	2
1 Prosjektgruppen.....	3
2 Hjemmeside.....	5
3 Oppdragsgiver	5
4 Sensorer og veiledere	6
5 Målet med prosjektet	6
6 Avgrensninger og forutsetninger	7
6.1 Avgrensninger.....	7
6.2 Forutsetninger.....	7
7 Arbeidsoppgaver	7
7.1 Liste over arbeidsoppgaver.....	7
7.2 Oppgavebeskrivelser.....	8
7.2.1 Komponentdokumentasjon.....	8
7.2.2 Bestilling av deler.....	8
7.2.3 Koblingsskjema.....	8
7.2.4 Kontroll av komponenter.....	8
7.2.5 Komponenter settes sammen i henhold til koblingsskjemaet.....	8
7.2.6 Bakkestasjon klargjøres for kameramodul.....	8
7.2.7 Første prototype ferdigstilles.....	8
7.2.8 Kontroll av første prototype.....	8
7.2.9 Feilsøking og testing av første prototype.....	9
7.2.10 Matematisk modell må utvikles.....	9
7.2.11 Reguleringsystemet må utvikles.....	9
7.2.12 Matematisk modell og reguleringsystem må simuleres.....	9
7.2.13 Implementering av reguleringsystem.....	9
7.2.14 Konstruksjon av testbenk.....	9
7.2.15 Testing av reguleringsystemet.....	9
7.2.16 Feilsøking/forbedring av reguleringsystem.....	9
7.2.17 Utvikling av nødlandingsfunksjon for kommunikasjonssvikt.....	9
7.2.18 Testing av nødlandingsfunksjon for kommunikasjonssvikt.....	10
7.2.19 Utvikling av nødlandingsfunksjon for kritisk batterinivå.....	10
7.2.20 Testing av nødlandingsfunksjon for kritisk batterinivå.....	10
7.2.21 Testing av sensorsignal, og implementering av eventuell signalbehandling.....	10
8 Aktiviteter og ansvar.....	10
8.1 Generell ansvarsfordeling.....	10
8.2 Ansvar fordelt utover oppgaver.....	11
8.3 Aktiviteter.....	12
9 Tidsplaner.....	13
10 Kostnadsbudsjett	13
11 Dokumentoversikt	14
12 Referanser	15
13 Relevante dokumenter	15
14 Vedlegg	15

Vedlegg 1: Aktiviteter.....	16
Vedlegg 2: Gantt-diagrammer.....	20
Vedlegg 3: Budsjetter.....	22

Dokumenthistorie

Versjon	Beskrivelse	Når
1.0	Lagt til budsjett, oppdatert ansvarsområder. Endret milepæler etter iterasjonene, Omformulert aktivitetene 501 og 502 Lagt til aktivitetene 112, 204, 205, 406, 603 og 605	2011-01-03
2.0	Korrekturlest . Lagt til aktivitetene 407-411, 504, 505, 701, 702 Oppdatert ansvarsfordeling og kryssreferanser mot aktiviteter	2011-01-18
3.0	Lagt til aktivitetene 113 og 114, rettskrevet og ferdigstilt	2011-04-05
4.0	Oppdatert gantt-diagrammer Korrektur, dokument rettet til dokumentmal	2011-05-22

Tabell 1: Dokumenthistorie

Definisjoner og forkortelser

RUP	Rational Unified Process
UAS	Unmanned Aircraft System
Roll, Pitch, Yaw	Rotasjon om de respektive aksene
VTOL	Vertical Take Off and Landing

Tabell 2: Definisjoner og forkortelser

Innledning

Hensikt med dokumentet

Hensikten med prosjektplanen er å gi en detaljert beskrivelse av arbeidsoppgavene vi kommer til å ha i løpet av hovedprosjektet. Prosjektplanen skal beskrive hva som skal gjøres, når det gjøres og av hvem det gjøres. Den skal også beskrive hvilke arbeidsoppgaver som er avhengige av hverandre. Arbeidsoppgavene kommer til å utføres i iterasjoner, slik det er beskrevet i arbeidsmetoden vi har valgt.

Kort beskrivelse av prosjektet

Prosjektet går ut på å utvikle og konstruere et firerotors helikopter, heretter referert til som fartøyet, som skal være stabilt og lett å fly via en fjernkontroll.

Prosjektet er definert av prosjektgruppen, og gjøres i samarbeid med firmaet Simicon som var villige til å sponse og veilede prosjektet.

Fartøyet skal stabiliseres ved hjelp av et reguleringsystem i tre akser, roll, pitch og yaw, samt ved hjelp av nødvendige sensorer som gyroer og akselerometere. Reguleringen og stabiliseringen skal gjøres via en mikrokontroller.

Illustrasjonsbilder



Figur 1: [1]



Figur 2: [2]

1 Prosjektgruppen



Vegard Torkelsen - Elektroingeniør - Kybernetikk - Prosjektleder.

- *Kort beskrivelse:* 23 år, opprinnelig fra Hønefoss, men flyttet til Kongsberg etter førstegangstjenesten i 2007 for å studere ved HiBu. Høsten 2008 hadde jeg et semester ved University of Woollongong i Australia. Etter endt bachelor ønsker jeg å ta en master i kybernetikk ved enten UiS eller NTNU. Fritidssysler er bl.a gitar, tv-spill og surfing.
- *Styrker:* Blir veldig engasjert i prosjektene jeg er del av og jeg er ikke redd for å ta på meg en utfordring.
- *Svakheter:* Kan fort begynne å stresse, slik at det kan “gå litt fort i svingene”. Kan også bli noe utålmodig når det kommer til ting jeg ikke har noe makt over.
- *Kontaktinfo:* tlf: 9953582, e-post: vegard_3@hotmail.com



Michael Odden - Dataingeniør - Simulering og spillutvikling

- *Kort beskrivelse:* 25 år, opprinnelig fra Sarpsborg men bor nå i Drammen. Studerer til dataingeniør (Simulering & Spillutvikling) ved HiBu. Har fra før fagbrev i IKT-driftsfag med lærlingtid ved Luftforsvarets Skolesenter Kjevik. Ønsker etter endt bachelor å jobbe innen serious gaming med vekt på simulering.
- *Styrker:* Har god forståelse for programmering generelt, webutvikling og utvikling av desktop-programvare spesielt og mye erfaring med prosjektarbeid i arbeidssammenheng. Har også god forståelse for matematikk og fysikk.
- *Svakheter:* Setter noen ganger for store forventninger til meg selv og har noe lett for å ta på meg for mange oppgaver.
- *Kontaktinfo:* tlf: 95293737, e-post: me@michaelodden.com



Herå Rørvik - Elektroingeniør - Kybernetikk

- *Kort beskrivelse:* 32 år, opprinnelig fra Iran. Kom til Norge i 2000 og bor nå i Drammen med min mann og to barn. Har hjelpepleier fagbrev, og har jobbet på sykehuset i Drammen siden 2006. Jobber der fremdeles ved siden av studiene.
- *Styrker:* Trives med å jobbe i gruppe og ønsker alltid å gjøre mitt beste.
- *Svakheter:* Blir fort stresset når ting ikke går etter planen.
- *Kontaktinfo:* tlf: 91517631, e-post: hera.rorvik@gmail.com



Trine Lindberg - Elektroingeniør - Kybernetikk

- *Kort beskrivelse:* 22 år, opprinnelig fra Fredrikstad men bor nå i Vestfossen mens jeg studerer i Kongsberg. Etter endt videregående i 2007 jobbet og reiste jeg et år før jeg startet på bachelorstudiet i kybernetikk.
- *Styrker:* Er pliktoppfyllende og liker utfordringer. Jobber like godt i gruppe som individuelt.
- *Svakheter:* Liker ikke gi ifra meg ting som ikke er 100%, noe som kan føre til at jeg bruker mye tid på slutførelsen av ting.
- *Kontaktinfo:* tlf: 47077385, e-post: trine.lindberg88@gmail.com



Luke Carambot - Elektroingeniør - Mekatronikk

- *Kort beskrivelse:* 28 år, opprinnelig fra Oregon USA, er gift og bor i Kongsberg. Var ferdig med videregående i år 2000 og har jobbet mest som vaktmester/bygger inntil jeg begynte på bachelorstudiene på Høgskolen i Buskerud i 2007.
- *Styrker:* Kan sette meg godt inn i et oppgave. Løsningsorientert og liker å gjøre ting riktig. Jeg er veldig analytisk. Synes det er underholdende å være kreativ og designe/bygge ting. Jeg er nysgjerrig og er ikke redd for å prøve meg fram.
- *Svakheter:* Ikke den beste i å planlegge før jeg begynner med noe. Bruker mye tid for å få ting riktig/rett. Vil gjerne utsette ting for å få det riktig gjort.
- *Kontaktinfo:* tlf: 41045413, e-post: lukecarambot@gmail.com



Patrick Bjørum - Elektroingeniør - Mekatronikk

- *Kort beskrivelse:* 25 år, opprinnelig fra Oslo, men bor nå i Lier. Var ferdig med videregående i 2004 og utførte førstegangstjenesten samt jobbet før jeg begynte på bachelorstudiene hos Høgskolen i Buskerud i 2007. Jeg har drevet med R/C helikopter på fritiden i 3 år.
- *Styrker:* Har en god del teknisk innsikt i dette temaet pga. hobbyen min og klarer dermed lett å se helheten når vi gjør ting. Kan være perfeksjonist, så jeg liker at ting blir gjort ordentlig og gir meg ikke før det er slik jeg vil ha det. Er veldig sosial og liker godt å jobbe i grupper.
- *Svakheter:* Etersom jeg kan være perfeksjonist så kan ting ta litt lengre tid enn det burde gjort noen ganger. Hvis ting går tregt kan jeg trenge "et spark bak" for å komme i gang igjen.
- *Kontaktinfo:* 92054151, patrick.bjorum@gmail.com

2 Hjemmeside

Vi har opprettet en hjemmeside for prosjektet:

<http://projectdragonfly.no/>

Hjemmesiden vil være vår front for informasjon rundt prosjektet, samt tjene som inngang til vårt prosjekthåndteringssystem. Siden består også av et passordbeskyttet område hvor vi deler intern informasjon. Vi har også en facebookside som er mer uformell, og som oppdateres litt oftere.

3 Oppdragsgiver

Ragnvald Otterlei – Simicon AS

Kontor:

Dyrmyrgata 35

3611 Kongsberg

Post:

PB 787

3606 Kongsberg

epost: *info@simicon.no*

Simicon AS er et lite firma som utvikler og selger tjenester og systemer i tilknytning til ubemannede luftfartøyer. Selskapet ble etablert i 1999 og holder til i Kongsberg. De senere årene har de jobbet mest innenfor maritime og arktiske anvendelser av UAS. De har blant annet utviklet et UAS system for datafangst i forhold til meteorologi og båttrafikk. [3]

De har også gjort studier og konseptutvikling på et fly med VTOL (ta av og lande vertikalt) egenskaper, dette flyet er en hybrid mellom et fly og et helikopter, slik at det bruker VTOL egenskapene til helikopteret, mens det etter letting kan bruke flyets vinger for å oppnå høy hastighet og rekkevidde. [4]

Bedriften er privateid. De leverer multifunksjonelle produkter som kan utstyres med en rekke sensorer slik som optiske kameraer, infrarøde kameraer, ladar (laser direction and range), GPS, IMU, altimeter og mikrofoner. I tillegg tilbyr de rådgivningstjenester innen UAS-anvendelser.

4 Sensorer og veiledere

Intern veileder:

Dag Samuelsen

E-post: dag.samuelsen@hibu.no

Mobil: 92058406

Intern sensor:

Olaf Hallan Graven

E-post: Olaf.Hallan.Graven@hibu.no

Telefon: 32869556

Ekstern veileder og sensor:

Ragnvald Otterlei

E-post: ragnvald@simicon.no

Telefon: 32728370/92030141

5 Målet med prosjektet

Målet med dette prosjektet er å demonstrere et fartøy som flyr stabilt, og som er enkelt å fly via en fjernkontroll. Fartøyet skal stabiliseres slik at det skal være enkelt å lette, fly og lande. Fartøyet skal også utstyres med en kameramodul slik at man kan fly selv om fartøyet er utenfor syns rekkevidde. Det ferdige fartøyet skal kunne brukes til senere utvidelser uten større inngrep i designet.

Prioritet 1

Vi må oppnå kommunikasjon mellom kontrollen, mikrokontrolleren og motorene slik at fartøyet kan fly.

Prioritet 2

Fartøyet skal reguleres i pitch, roll og yaw, slik at det kan hovre. Fartøyet skal kunne fly i "autopilot", slik at den kan hovre med liten input fra fjernkontrollen, denne "autopiloten" skal kunne skrues av og på. Med hover, mener vi at fartøyet holder seg i et visst område, slik at den ikke driver vekk. Vi må også oppnå kommunikasjon mellom fartøyet kamera og bakkestasjon.

Prioritet 3

Det er ønskelig at fartøyet har en fail-safe funksjon, slik at fartøyet lander automatisk hvis kommunikasjonen mellom kontrolleren og fartøyet blir brutt.

Det er også ønskelig å utvikle en metode for å advare brukeren når batteriet når et gitt minimumsnivå, og hvis batteriet når et enda lavere batterinivå ønsker vi at fartøyet skal lande automatisk.

Prioritet 4

Fartøyet bør selv kunne regulere høyden slik at man kan øke stabiliteten til fartøyet, spesielt når det hovrer.

6 Avgrensninger og forutsetninger

6.1 Avgrensninger

For å avgrense prosjektet, har vi delt opp prosjektet i flere forskjellige prioriteter. Vi konsentrerer oss først om hovedprioritetene, og fortsetter så med de andre hvis det blir tid til dette. Tiden blir nok vår største begrensning.

6.2 Forutsetninger

Det er flere forutsetninger for at vi skal kunne komme i mål med prosjektet, blant annet at delene vi bestiller kommer frem i tide. Vi bestiller delene tidlig for at dette ikke skal bli noe problem, men uventede ting kan likevel oppstå, så vi bør ha en nødløsning hvis noe uventet skulle hende.

Andre viktige forutsetninger er at prosjektgruppen yter maksimalt og holder alle tidsfrister, samt at vi klarer å sette teorien ut i praksis. Det er også en forutsetning at vi klarer å planlegge arbeidet på en slik måte at vi har god oversikt over hvem som skal gjøre hva, til hvilken tid. Når vi gjør dette er det viktig at vi ikke planlegger for store arbeidsoppgaver, for da vil det bli vanskelig å se om vi klarer å holde tidsfristen eller ikke.

7 Arbeidsoppgaver

7.1 Liste over arbeidsoppgaver

Her kommer en liste over arbeidsoppgaver som må gjøres, denne listen er bare for å få en oversikt over de arbeidsoppgavene og utfordringene vi står ovenfor. Den kommer ikke til å inneholde informasjon om tidspunkt/ansvar e.l, da dette kommer i kapitlet Arbeidsoppgaver og ansvar. Detaljert beskrivelse av oppgavene kommer i neste delkapittel.

- Komponentene må bestilles.
- Komponentene må hentes.
- Sette opp koblingskjema for komponentene.
- Kontrollere at alle komponentene fungerer.
- Komponentene settes sammen iht. Koblingskjema.
- Bakkestasjonen klargjøres slik at den kan fungere med kamera.
- Første prototype av fartøyet testes for sine hovedfunksjoner, uregulert.
- Feilsøking/Forbedring av hovedfunksjoner, uregulert.
- Matematisk modell må utvikles
- Reguleringsystem må utvikles.
- Matematisk modell må simuleres.
- Reguleringsystem må simuleres.
- Reguleringsystem må implementeres.
- Konstruksjon av testbenk.
- Testing av reguleringsystem.
- Feilsøking/forbedring av reguleringsystem/matematisk modell.
- Utvikling av nødlandingsfunksjon for kommunikasjonssvikt, C krav.
- Testing av nødlandingsfunksjon for kommunikasjonssvikt.
- Utvikling av nødlandingsfunksjon for kritisk batterinivå, C krav.
- Testing av nødlandingsfunksjon for kritisk batterinivå.

7.2 Oppgavebeskrivelser

7.2.1 Komponentdokumentasjon

Før vi bestiller noen av komponentene, må vi forsikre oss om og dokumentere at disse komponentene er de vi har behov for og at de er kompatible med hverandre. Dette dokumenteres i en komponentdokumentasjon.

7.2.2 Bestilling av deler

De delene som er satt opp på budsjettet og som er godkjent av Simicon skal bestilles. Gruppemedlemmene vil legge ut for bestillingen og alle fakturaer og kvitteringer tas vare på, slik at vi får tilbakebetalt av Simicon. Hvis noen av delene ikke er på lager, må leverandøren kontaktes slik at vi får en oversikt over når delene er tilgjengelige. Hvis dette viser seg å være vanskelig må vi finne alternative leverandører eller eventuelt alternative deler.

7.2.3 Koblingsskjema

Det skal settes opp et detaljert koblingsskjema for fartøyet. Her må vi forsikre oss om at alle komponenter er koblet sammen korrekt, at alle komponentene er koblet til med riktig spenning, og at de ikke drar for høye eller for lave strømmer. Vi må forsikre oss om at alle komponenter er i stand til å kommunisere med hverandre. Vi må også velge fornuftige innganger/utganger på mikroprosessen slik at det blir plass til alle komponentene, og slik at kretskortet blir så kompakt som mulig.

7.2.4 Kontroll av komponenter

Alle komponentene må kontrolleres, slik at vi er sikre på at vi ikke mottar deler som er defekte, og at vi forsikrer oss om at vi forstår virkemåten korrekt, og at de oppfører seg i henhold til datablad.

7.2.5 Komponenter settes sammen i henhold til koblingsskjemaet

Etter at komponentene er kontrollert og koblingsskjemaet er ferdigstilt skal alle komponentene settes sammen, slik at det er klart til programmering av mikroprosessen. Her er det viktig at vi dokumenterer hvilke eventuelle mangler vi har, slik som festeskruer, spacere, koblinger osv. Det er også viktig at vi er klar over at vi kan bli nødt til å gjøre endringer på oppsettet, slik at vi ikke lodder fast, eller permanent fester komponenter unødvendig.

7.2.6 Bakkestasjon klargjøres for kameramodul

Bakkestasjonen må kunne fungere med kameramodulen, slik at vi kan få livestream av opptaket. Vi må da benytte en bærbar datamaskin som vi kan koble mottakeren på. Vi må her også undersøke rekkevidden på kameramodulen.

7.2.7 Første prototype ferdigstilles

Etter at alle delkomponenter er koblet sammen, må mikroprosessen programmeres, slik at vi kan få et uregulert system på fartøyet. Dette vil si at stikkene styrer motorene direkte ved hjelp av et enkelt program. Den første prototypen er helt uregulert.

7.2.8 Kontroll av første prototype

Når første prototypen er ferdigstilt, må denne kontrolleres. Vi må her se at fartøyet oppfører seg korrekt når vi justerer stikkene, og kunne se at vi får styrt fartøyet i alle akser, roll, pitch og yaw, samt at vi får korrekt gasspådrag.

7.2.9 Feilsøking og testing av første prototype

Etter at den første prototypen er kontrollert, må vi sørge for at vi blir kvitt alle feil og at systemet optimaliseres, slik at vi får så liten og effektiv kode som mulig. Vi må også forbedre eventuelle mekaniske feil på fartøyet. Her må vi også se om vibrasjonsdemperene fungerer som planlagt.

7.2.10 Matematisk modell må utvikles

Når den første prototypen er godkjent, må vi begynne å utvikle et reguleringsystem for fartøyet. Da må vi starte med en detaljert matematisk modell som kan forutsi fartøyets oppførsel når det flyr. Det vil si at vi må kunne lage en modell for fartøyets bevegelser i roll, pitch og yaw, samt gasspådrag. Denne matematiske modellen vil da brukes med reguleringsystemet.

7.2.11 Reguleringsystemet må utvikles

Når den matematiske modellen er ferdig må den implementeres sammen med et reguleringsystem som vil bestå av tre separate PID-er med tilbakekobling. Disse vil da regulere vinklene i roll, pitch og yaw. Gasspådraget vil styres manuelt siden vi ikke har noe måte å måle høyde på.

7.2.12 Matematisk modell og reguleringsystem må simuleres

Før den matematiske modellen og reguleringsystemet implementeres må vi simulere disse, slik at vi forsikrer oss om at vi får kvittet oss med de groveste feilene.

7.2.13 Implementering av reguleringsystem.

Neste trinn blir å implementere reguleringsystemet. Det vil si at vi må programmere inn PID-ene i mikroprosessorern.

7.2.14 Konstruksjon av testbenk

For å kunne teste kravene må vi konstruere en testbenk. Vi kommer til å behøve to forskjellige oppsett, et hvor vi kan måle roll/pitch og en hvor vi kan måle yaw. Disse testbenkene vil brukes til å måle at kravene er møtt og til å justere PID-ene.

7.2.15 Testing av reguleringsystemet

Etter at reguleringsystemet er implementert og testbenkene konstruert, må vi teste reguleringsystemet. Her må vi måle at vi møter de vinklene som korresponderer til utslag på stikkene. Vi må også måle hvor fort systemet reagerer, og hvor mye oversving vi får.

7.2.16 Feilsøking/forbedring av reguleringsystem

Etter at reguleringsystemet er testet må vi bruke denne informasjonen til å eliminere eventuelle feil, og forbedre reguleringsystemet, slik at vi får så rask respons med så lite oversving som mulig.

7.2.17 Utvikling av nødlandingsfunksjon for kommunikasjonssvikt

Når denne arbeidsoppgaven er nådd, vil vi være ferdig med A, og B kravene. De siste arbeidsoppgavene vil inneholde ønskelige funksjoner, altså C krav. Denne funksjonen skal utvikles for å øke sikkerheten til fartøyet. Hvis fartøyet mister kommunikasjon med kontrolleren, skal mikroprosessorern gå inn i en egen løkke som vil få fartøyet til å stanse i luften og deretter redusere høyden sakte.

7.2.18 Testing av nødlandingsfunksjon for kommunikasjonssvikt

Nødlandingsfunksjonen må testes, dette vil gjøres i hovedsak i testbenkene. Vi må der se hvordan fartøyet oppfører seg hvis kontrolleren blir avslått, altså når vi mister kommunikasjonen. Vi burde også klare å se om fartøyet hovrer før den begynner å lande.

7.2.19 Utvikling av nødlandingsfunksjon for kritisk batterinivå

Det skal i tillegg utvikles en nødlandingsfunksjon som slår inn hvis batterinivået blir kritisk lavt. Denne nødlandingsfunksjonen skal oppføre seg likt som den for kommunikasjonssvikt. Det vil si at fartøyet overstyrer kontrolleren og utfører en nødlanding.

7.2.20 Testing av nødlandingsfunksjon for kritisk batterinivå

Denne nødlandingsfunksjonen må også testes, den testes på samme måte som funksjonen for kommunikasjonssvikt. Vi må her sjekke at når batteriet når kritisk nivå, vil mikrokontrolleren overstyre kontrolleren og gå inn i nødlandingsfunksjonene.

7.2.21 Testing av sensorsignal, og implementering av eventuell signalbehandling

Signalet som kommer fra sensorene må undersøkes, og vi må gjøre en vurdering om det vil være nødvendig å behandle dette signalet på noen måte. Hvis vi bestemmer oss for å gjøre dette, må vi da også implementere denne signalbehandlingen.

8 Aktiviteter og ansvar

8.1 Generell ansvarsfordeling

Vegard
<ul style="list-style-type: none">• Prosjektledelse• Kommunikasjon• Signalbehandling
Luke
<ul style="list-style-type: none">• Overholdning av dokumentmal• Matematisk modell
Trine
<ul style="list-style-type: none">• Korrekturlesing• Reguleringsystem

Herå
<ul style="list-style-type: none"> • Testing • Analyse
Michael
<ul style="list-style-type: none"> • Hjemmeside • Programmering • Utvikling av dokumentmaler • Simulering
Patrick
<ul style="list-style-type: none"> • Bestilling av komponenter • Sammensetting av komponenter • Økonomi

Tabell 3: Generell ansvarsfordeling

8.2 Ansvar fordelt utover oppgaver

Oppgave	Ansvar	Ref aktivitet
1. Komponentdokumentasjon	Patrick	205
2. Bestilling av deler	Patrick	405
3. Koblingsskjema	Patrick	407
4. Kontroll av komponenter	Vegard	
5. Sammensetning av komponenter.	Patrick	501
6. Klargjøring av bakkestasjon.	Patrick	
7. Ferdigstilling av første prototyp.	Patrick	503
8. Kontroll av første prototyp.	Patrick	
9. Feilsøking og testing av første prototyp.	Herå	601
10. Utvikling av matematisk modell.	Luke	408
11. Utvikling av reguleringssystem.	Trine	406
12. Simulering av matematisk modell og reguleringssystem.	Michael	701, 702

Oppgave	Ansvar	Ref aktivitet
13. Implementering av reguleringsystem.	Trine	505
14. Konstruksjon av testbenk.	Luke	502
15. Testing av reguleringsystem.	Herå	601
16. Feilsøking/forbedring av reguleringsystem.	Trine	406
17. Utvikling av nødlandingsfunksjon for kommunikasjonssvikt.	Vegard	410
18. Testing at nødlandingsfunksjon for kommunikasjonssvikt.	Herå	601
19. Utvikling av nødlandingsfunksjon for batterisvikt.	Vegard	410
20. Testing av nødlandingsfunksjon for batterisvikt.	Herå	601
21. Testing av sensorsignal, og evt utvikling og implementering av signalbehandling.	Vegard	601, 504

Tabell 4: Ansvar for oppgaver

8.3 Aktiviteter

Se vedlegg 1 for liste over alle aktiviteter med beskrivelse av hver aktivitet, samt hvem som har ansvar for de forskjellige aktivitetene.

9 Tidsplaner

Tidsplanene våre er delt opp i iterasjoner. Dette for blant annet enkelt å kunne følge fremdriften, samt at det tillater oss å kunne fokusere på de til enhver tid viktige aspektene av prosjektet. Hver iterasjon er en komplett gjennomgang fra planlegging til leveranse av et produkt som er relevant i dette stadiet.

Det vil være realistisk å anta at vi ved nærmere gjennomganger av planene vil inkorporere mer spesifikke under-iterasjoner for de enkelte milepælene.

Ved å benytte en iterativ tilnærming til gjennomføring av prosjektet får vi som sagt en god oversikt over hva som til enhver tid må gjøres og hva som er gjort. Dette i tillegg til at vi har brutt ned problemstillingen på en slik måte at vi relativt enkelt kan gå tilbake til tidligere iterasjoner og utføre korrigeringer dersom noe skulle vise seg å være utilstrekkelig.

Man vil også se at de tidlige iterasjonene har en større vekt på definering av prosjektets innholds og styremåte mens de senere har fokus på gjennomføring og utvikling av selve produktet.

Nedenfor følger en oversikt over milepælene i prosjektet, se vedlegg 3 for detaljerte gantt-diagrammer.

Milepæl 1

Milepæl 1 har blitt plassert etter fullføring av planlegging av prosjektet, det vil si etter iterasjon 1.

Milepæl 2

Vi har valgt å sette milepæl 2 etter fullføring av uregulert system, dette er fordi dette er et av de største målene med prosjektet.

Milepæl 3

Milepæl 3 har blitt satt etter fullføring av det regulerte systemet, dette er fordi dette er et av de største målene med prosjektet.

Milepæl 4

Milepæl 4 er ferdigstilling av hele prosjektet.

10 Kostnadsbudsjett

Det har blitt satt opp to budsjetter for prosjektet, et for deler til fartøyet og et for administrasjonsmaterialer.

Gruppen kommer til å legge ut penger for innkjøp av deler og få dette tilbakebetalt av Simicon. For å spare på utgiftene blir alle delene kjøpt på internett fra forskjellige internasjonale nettsider. På grunn av mulig lang leveringstid ble delene bestilt allerede i desember slik at alle delene er tilgjengelige når vi skal starte prosjektet i januar.

Budsjettet for administrasjonsmaterialene er bare et estimert budsjett over hva utgiftene blir for papir og trykkmateriell, samt kaffe og kaker til de tre presentasjonene. Se vedlegg 4 for budsjetter.

11 Dokumentoversikt

- Prosjektbeskrivelse
- Forstudierapport
- Kravspesifikasjon
- Prosjektplan
- Iterasjonsdokumenter og rapporter
- Brukerveiledning prosjektverktøy/webseite
- Møteinnkallinger
- Møtereferater
- Oppfølgingsdokumenter
- Dokumentmaler
- Kodestandard
- Testspesifikasjon
- Testplan
- Designdokument
- Analysedokument
- Multikopterteori
- Testlogg
- Testrapport
- Etteranalyse

12 Referanser

- [1] <http://www.kkmulticopter.com/index.php/classifieds?view=annonce&id=3> 19.10.2010
- [2] <http://www.mictronics.de/projects/my-mikrokoetter/> 19.10.2010
- [3] <http://simicon.no/> 15.12.2010
- [4] <http://www.ntnu.no/gemini/2001-06E/20-21.htm> 15.12.2010

13 Relevante dokumenter

- Prosjektbeskrivelse
- Forprosjektrapport
- Kravspesifikasjon
- Testspesifikasjon

14 Vedlegg

- 1: Aktiviteter
- 2: Gantt diagrammer
 - NB! Oppdatert gantt diagram finnes på <http://git.projectdragonfly.no:3000/projects/dragonfly/issues/gantt>
- 4: Budsjett deler og administrasjon

Vedlegg 1: Aktiviteter

Aktivetsnr	Aktivitet	Beskrivelse	Ansvar
1xx	Administrasjon og prosjektstyring		
101	Prosjektorganisering	Denne aktiviteten dekker den generelle prosjektorganiseringen	Vegard
102	Møter	Møter med interne veiledere og med oppdragsgiver/ekstern veileder. Møter innad i gruppa er ikke med i denne aktiviteten.	Vegard
103	Idèdokument	Idèdokument er levert i slutten av august	Trine
104	Prosjektplan	Prosjektplanen må hele tiden oppdateres	Trine
105	Standarisering av dokumenter/kode	Oppretting av maler, standarisering av dokumenter eller kode	Michael
106	Økonomi	Oppretting av budsjetter, betaling av regninger, føre regnskap o.l.	Patrick
107	Hjemmeside	Oppretting og vedlikehold av hjemmesiden	Michael
108	Timelister	Eksportere timelister fra hjemmesiden og lignende	Michael
109	Prosjektplakat	Opprette en plakat med resultater fra prosjektet	Michael
110	Internmøter	Internmøter av gruppemedlemmene	Vegard

Aktivitetsnr	Aktivitet	Beskrivelse	Ansvar
111	Presentasjoner	Alt arbeid rundt presentasjonene	1. presentasjon: Michael 2. presentasjon: Vegard 3. presentasjon: Trine
112	Oppfølgingsdokumenter	Skriving av oppfølgingsdokumenter	Michael
113	Iterasjonsplaner	Skriving av iterasjonsplaner	Michael
114	Iterasjonsrapporter	Skriving av iterasjonsrapporter	Michael
2xx	Research		
201	Signalbehandling	Research knyttet til signalbehandling	Vegard
202	Reguleringsystem	Research knyttet til reguleringsystem	Trine
203	Matematisk modellering	Research knyttet til matematisk modellering	Luke
204	Multikopterteori	Skriving av teoridokument	Vegard
205	Komponentdokumentasjon	Dokumentasjon av komponenter	Patrick
3xx	Krav/analyse		
301	Forstudie	Arbeid med forstudiedokument, rapporten skal inneholde en detaljert beskrivelse av prosjektet, frem til prosjektplanen er ferdig.	Trine
302	Kravspesifikasjon	Avklare krav med oppdragsgiver, lage dokument	Herå
303	Analyse	Risikoanalyser og lignende, må oppdateres underveis	Luke
4xx	Design/teknologi		

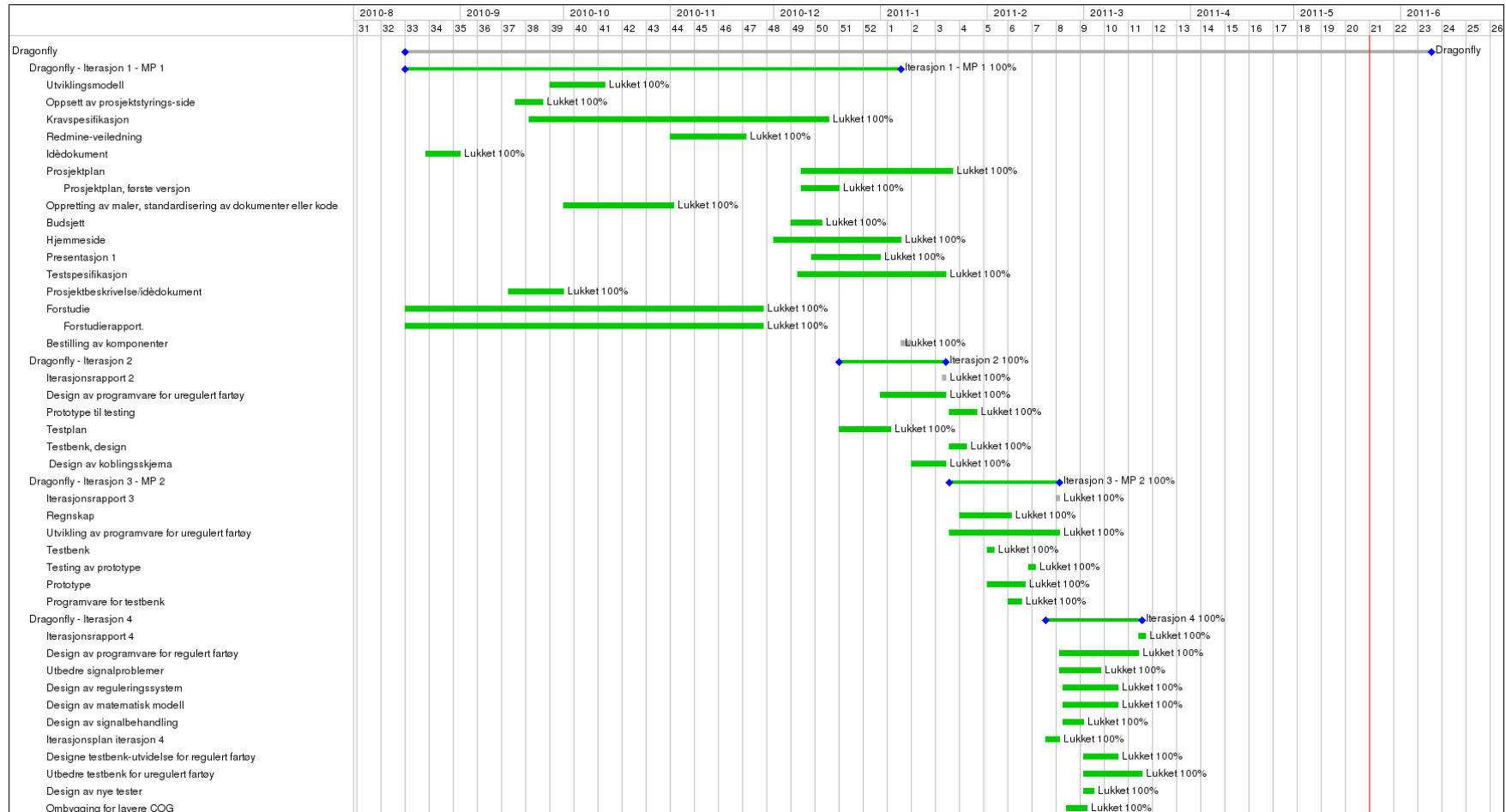
Aktivitetsnr	Aktivitet	Beskrivelse	Ansvar
401	Design av programvare	Design av software	Michael
402	Designokument	Dokument om design	Trine
403	Teknologidokument	Dokument med tekniske spesifikasjoner og forklaringer	Vegard
404	Programmering	Programmering	Michael
405	Bestilling av komponenter	Bestilling av komponenter	Patrick
406	Reguleringssystem	Design og tuning av reguleringssystem	Trine
407	Design av koblingsskjema	Design av koblingsskjema	Michael
408	Design av matematisk modell	Design av matematisk modell	Luke
409	Design av signalbehandling	Design av signalbehandling	Vegard
410	Design av testbenk	Design av testbenk	Luke
411	Utvikling av ekstrarfunksjoner	Utvikling av nødlandingsfunksjoner	Vegard
5xx	Implementasjon		
501	Konstruksjon av prototype	Fysisk konstruksjon, sammenetting av delsystemer	Patrick
502	Testbenk	Konstruksjon av testbenk	Luke
503	Ferdigstilling	Ferdigstilling, overlevering til oppdragsgiver	Vegard
504	Implementering av signalbehandling	Implementering av signalbehandling	Vegard
505	Implementering av reguleringssystem	Implementering av reguleringssystem	Trine
6xx	Test		
601	Testing	Testing av de ulike delene og prototype	Herå

Aktivitetsnr	Aktivitet	Beskrivelse	Ansvar
602	Testspesifikasjon	Opprette og oppdatere testspesifikasjon	Luke
603	Testplan	Skrive testplan	Luke
604	Testrapport	Skrive rapport etter testing	Luke
605	Testbenk	Design av testbenk	Luke
606	Bruerveiledning	Opprette dokument for andre brukere	Vegard
7xx	Simulering		
701	Matematisk modell	Simulering av matematisk modell	Luke
702	Reguleringssystem	Simulering av reguleringssystem	Trine

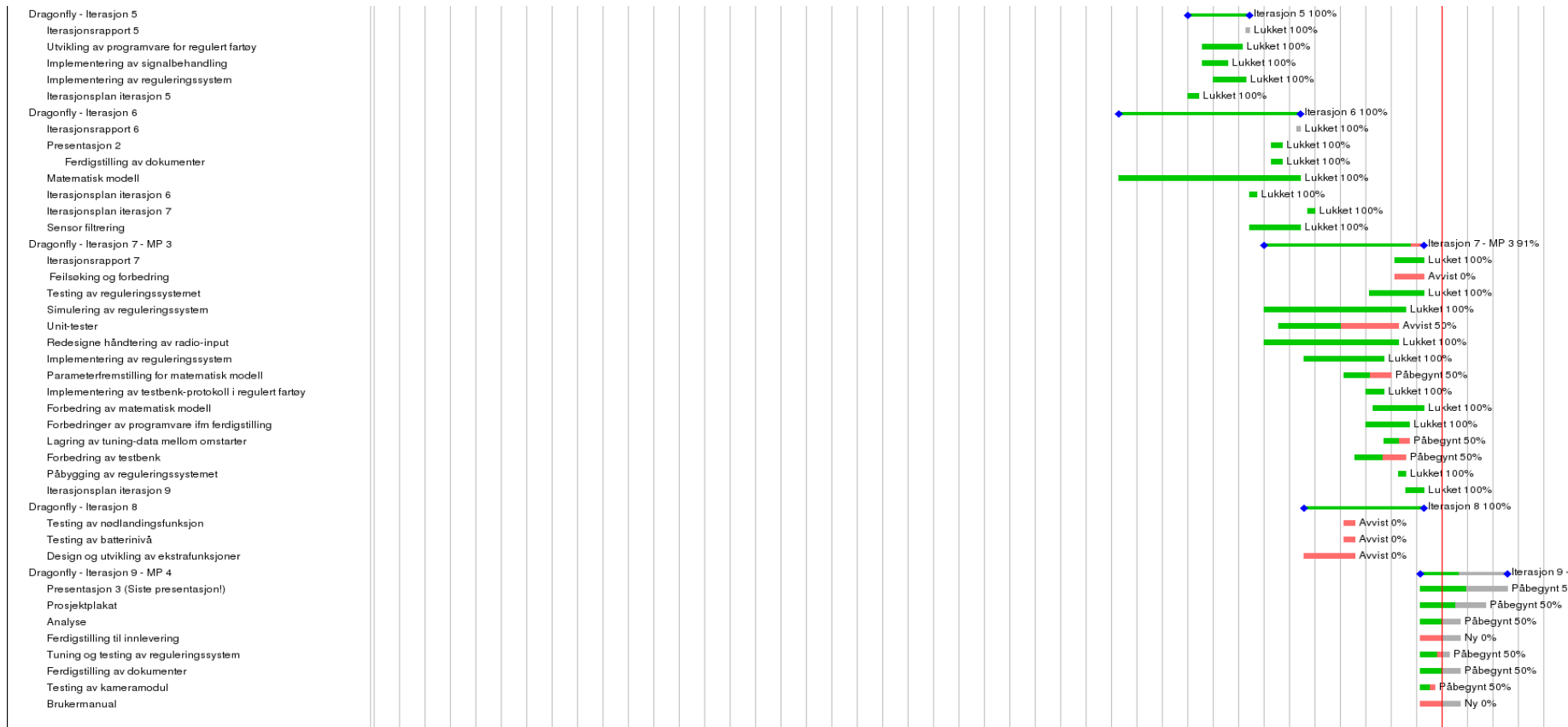
Tabell 5: Aktiviteter

Vedlegg 2: Ganttogrammer

NB! Oppdateres fortløpende, nyeste finnes på <http://git.projectdragonfly.no:3000/projects/dragonfly/issues/gantt> (krever innlogging)



Figur 3: Oppdatert ganttogram (2011-05-22) del 1



Figur 4: Oppdatert ganttogram (2011-05-22) del 2

Vedlegg 3: Budsjetter**Budsjett over deler til fartøyet**

Produkt Navn	Produktbeskrivelse	Pris	Valuta
ITG 3200 Triple-Axis Gyro	3-akse gyro	50,00	\$
AeroQuad Shield v2.0	Koblingsbrett	30,00	\$
Stackable Headers - Mega Pack	Koblingspins	12,00	\$
Arduino MEGA 2560	Mikrokontroller brett	65,00	\$
2.1mm DC Power Connector for Arduino	Kontakt/plugg	1,75	\$
APC 10x4.7 Slow Flyer Pusher Propellers (3 stk)	Propell	15,00	\$
APC 10x4.7 Slow Flyer Propellers (3 stk)	Propell	10,50	\$
Estimert frakt		35,00	\$
Reciever Cables (8stk)	Servo/mottaker kabler	10,00	\$
Connection Headers	Rad med koblingspunkter	2,25	\$
Angle pin headers (2stk)	45 graders koblingspunkter	4,00	\$
Estimert frakt		3,00	\$
BMA180 Triple Axis Accelerometer	3-aksers akselerometer	30,00	\$
Estimert frakt		8,50	\$
KDA20-22L Hacker style brushless outrunner (4 stk)	Motor	59,00	\$
Zippy Flightmax 4000mAh 3S1P 20C	Batteri	20,00	\$
TR_P18a Plush 18amp ESC (5stk)	Spennings kontroller	59,50	\$
Estimert frakt		20,00	\$

Project: **Dragonfly**

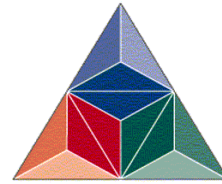
Produkt Navn	Produktbeskrivelse	Pris	Valuta
Futaba R617FS 7-Channel 2.4GHz FASST Receiver	Mottaker	89,00	\$
Estimert frakt		21,00	\$
KKquadrocopter Airframe	Helikopter kropp	89,00	\$
Estimert frakt		16,00	\$
Maxpro Battery Monitor	Batterialarm	6,00	\$
Estimert frakt		7,00	\$
Estimert fortolling		45,00	\$
Total \$		708,50	\$
Valuta (per 16.12.2010)		5,97	\$
Total NOK		4229,75	NOK
MVA		1,25	%
Total NOK inkl MVA		5287,18	NOK

Tabell 6: Budsjett over deler

Budsjett over diverse og administrasjonskostnader

Produkt/Aktivitet	Produkt/aktivitet beskrivelse	Pris	Valuta
Dokumentasjon	Papir og kopi	1000,00	NOK
Plakat	Plakater til å henge opp	150,00	NOK
Presentasjon 1	Kaffe og kaker/frukt	250,00	
Presentasjon 2	Kaffe og kaker/frukt	450,00	NOK
Presentasjon 3	Kaffe og kaker/frukt	450,00	NOK
Total		2300,00	NOK

Tabell 7: Budsjett over diverse administrasjonskostnader



HØGSKOLEN
i Buskerud

Oppgavetittel: Hovedprosjekt - Kravspesifikasjon - v5.0		
Fag(nr./navn): SFHO 3200 / Prosjektstyring med hovedprosjekt		
Gruppemedlemmer, signatur:		
_____ Vegard Torkelsen	_____ Michael Odden	_____ Herå Rørvik
_____ Trine Lindberg	_____ Luke Carambot	_____ Patrick Bjørum
Intern veileder: Dag Samuelsen Intern sensor: Olaf Hallan Graven Ekstern veileder og sensor: Ragnvald Otterlei		
Dato: 2011-05-15		

Innholdsfortegnelse

Dokumenthistorie.....	1
Definisjoner og forkortelser	1
Referanser	1
Innledning.....	2
Kravspesifikasjon	2
Oppgaven	2
Prioritet 1	2
Prioritet 2	2
Prioritet 3	2
Prioritet 4	3
1 Beskrivelse av systemet	3
2 Krav	3
2.1 Klassifisering av krav.....	3
2.1.1 Type klassifisering	4
2.1.2 Prioritet klassifisering	6
2.1.3 Fremstilling klassifisering	6
2.1.4 Tilfredsstillelse klassifisering	7
2.1.5 Rolle klassifisering	7
2.2 Identifisering av krav	7
3 Krav – oversikt	8
3.1 A Krav	8
3.2 B Krav	9
3.3 C Krav	10
3.4 D Krav	10
4 Krav – detaljer	11
4.1 Dokumentasjon	12
KA401	12
KB402	12
KB403	13
4.2 Reguleringsystem	14
KB501	14
KB502	14
KB503	15
KB504	15
KB505	16
KB506	16
KC507	17
KD508	17
4.3 Maskinvare	18
KA601	18
KA602	18
KA603	19
KB604	19
KB605	20
4.4 Fartøy	21
KA701	21
KA702	21

4.5 Brukergrensesnitt	22
KA801	22
KA802	22
KA803	23
KA804	23
4.6 Spesialfunksjoner	24
KA901	24
KB902	24
KB903	25
KC904	25
KC905	26

Dokumenthistorie

Versjon	Beskrivelse	Når
1.0	Dokument opprettet	2010-11-23
2.0	Fylt ut informasjon om x og y. Fjernet overflødige krav, sette opp iht dokumentmal	2010-11-25
3.0	Gjort om dokumentet Gjort ferdig kryssreferansene til testspesifikasjon.	2010-11-25
4.0	Ikke systemkrav har blitt trukket ut. Nå ha "Underkategoriene" erstattet prosjekt, system og prosess som kategorier. Feltet "Underkategori" ble fjernet. En ny kategori, Spesialfunksjoner, ble lagt til. Alle kravene ble omnummerert for å speile kategoriene. Nødvendige deler av innledningen ble fjernet. Korrekturlest Kommentert gamle kravIDer fra v3.0 Rettskrevet og ferdigstilt	2011-04-08
5.0	Rette opp feil kryssreferansene til v3.0	2011-05-15

Tabell 1: Dokumenthistorie

Definisjoner og forkortelser

HIBU	Høyskolen i Buskerud
PDF	Portable Document Format
Pitch	Rotasjon rundt lengden av «vingene» til fartøyet
Roll	Rotasjon rundt lengden av fartøyet
Word	Microsoft Word
Yaw	Rotasjon rundt aksene som ville være høydemålingen til fartøyet

Tabell 2: Definisjoner og forkortelser

Referanser

[1] Strøm, t., & Hallan Graven, O.(2010) *Prosjekthåndbok*: Kongsberg.

[2] Dragonfly (2011) *Forstudierapport*: Kongsberg.

[3] Glinz M. (2007) *On Non-Functional Requirements*: Internet (2010-12-14).

<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.94.9254&rep=rep1&type=pdf>

Innledning

Kravspesifikasjon

Dette dokumentet definerer kravspesifikasjonene til produktet. Her er alle kravene gitt av Simicon AS som oppdragsgiver og prosjektgruppen. Oppdragsgiveren selv har bestemt prioriteten til sine krav. Nesten alle krav foreslått av prosjektgruppen har lavere prioritet.

Kapittel 2 tar for seg klassifisering og identifisering av kravene. Ekstra tid er brukt på å finne en mer robust klassifisering av kravene, siden den vanlige oppdelingen i funksjonelle og ikke-funksjonelle krav ikke er veldig godt definert. Klassifiseringen som er tatt i bruk vil forhåpentligvis gjøre oppdelingen mer oversiktlig og danne bedre grunnlag til videre utvikling. Hovedpunktene er:

- Kravene klassifiseres etter [5 aspekter](#).
- En metode for å klassifisere kravtype.
- Ikke-funksjonelle krav er av typen Ytelse, Egenskap eller Begrensning.
- Det er fremstillingen til et krav som bestemmer testingen av kravet.
- Mer nøyaktig mål på tilfredsstillelse av kravene.

Kapittel 3 gir en rask oversikt over alle kravene sortert etter prioritet og deretter om de er funksjonelle eller ikke.

Oppgaven

Det skal konstrueres et fjernstyrt fire rotors helikopter (heretter beskrevet som fartøyet) og utvikles et reguleringsystem som skal gjøre det enklere å styre fartøyet. Fartøyet skal være fjernkontrollert av standard radioutstyr. Det er fire prioriteter i prosjektet [2] og kravene til systemet utledes ifra disse.

Prioritet 1

Det må oppnås kommunikasjon mellom fjernkontrollen, mikrokontrolleren og motorene slik at fartøyet kan fly.

Prioritet 2

Fartøyet skal reguleres i pitch, roll og yaw, slik at det kan hovre. Fartøyet skal kunne fly i “autopilot”, slik at den kan hovre med liten input fra fjernkontrollen, denne “autopiloten” skal kunne skrues av og på fra fjernkontrollen. Med hover, menes det at fartøyet holder seg i ro innenfor et visst område, slik at den ikke driver vekk. Det må også oppnås kommunikasjon mellom fartøyets kamera og bakkestasjon.

Prioritet 3

Det er ønskelig at fartøyet har en fail-safe funksjon, slik at fartøyet lander automatisk hvis kommunikasjonen mellom fjernkontrollen og fartøyet blir brutt. Det er også ønskelig å utvikle en metode for å advare brukeren når batteriet når et gitt minimumsnivå, eller hvis batteriet når et enda lavere batterinivå, er det ønskelig at fartøyet skal lande automatisk.

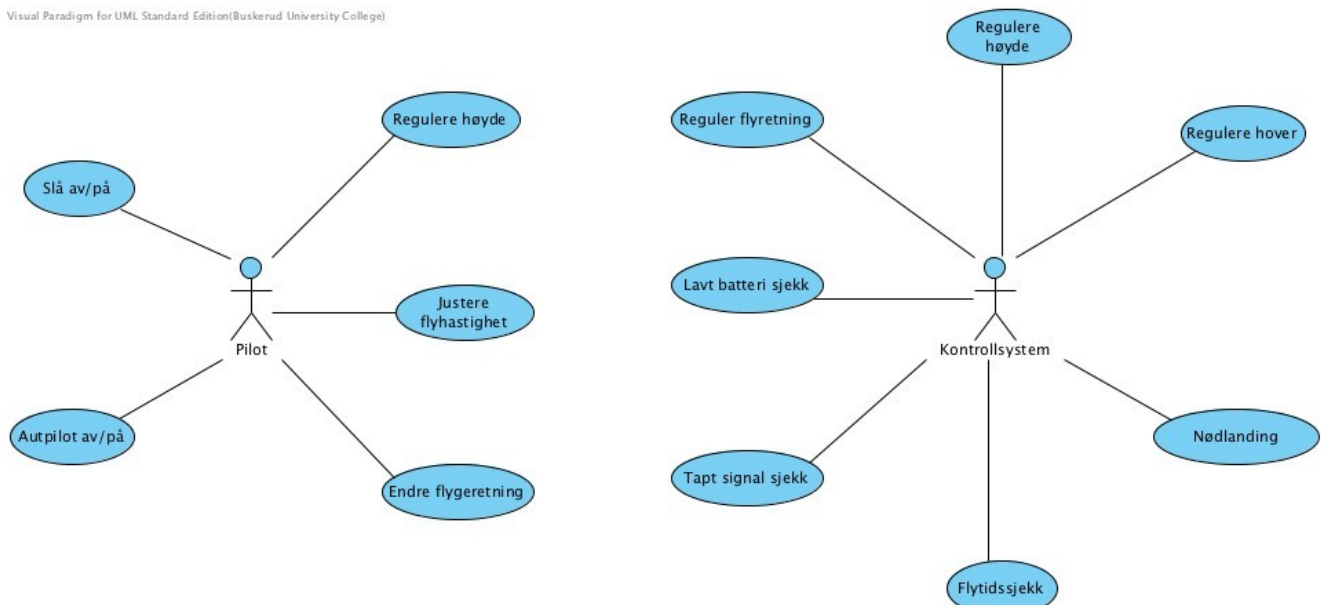
Prioritet 4

Fartøyet skal selv kunne regulere høyden slik at man kan øke stabiliteten til fartøyet, spesielt når det hovrer.

1 Beskrivelse av systemet

På det stadiet vi er nå kan vi se på systemet som styrt fra to hold, på den ene siden har vi de funksjonene som vil være direkte styrt av brukeren, og på den andre siden har vi de som kontrollsystemet tar hånd om. Vi vil i første omgang fokusere på å få fartøyet flygbart uten de reguleringene som til slutt skal gjøre fartøyet så stabilt vi ønsker.

Visual Paradigm for UML Standard Edition(Buskerud University College)



Figur 1: figur x - Overordnet Use Case-diagram for systemet. "Pilot" er personen som styrer fjernkontrollen, "Kontrollsystem" er programvaren som kjører på mikrokontrolleren

2 Krav

2.1 Klassifisering av krav

Det vanlige er å klassifisere krav som funksjonelle og ikke-funksjonelle. Dette er også blitt observert i mange kravspesifikasjoner fra tidligere hovedprosjekter på HIBU. Vi har ikke funnet noen klar

dokumentasjon på hvorfor kravene er delt inn i disse klassene, og den eneste metoden gitt for å bestemme om et krav er funksjonelt eller ikke er å definere et funksjonelt krav som noe som beskriver systemets funksjoner eller oppførsel. Ikke-funksjonelle krav er da definert som egenskapene til systemet.

Denne metoden kommer fort til kort. Ved å omformulere et krav kan det byttes om på klassifiseringen. Et eksempel; fartøyet skal kunne hovre (holde seg i ro) er et funksjonelt krav, mens at fartøyet skal kunne hovre innen en 3 meters radius er et ikke-funksjonelt krav. I følge [3] er det relativt god enighet i industrien, og også dokumentert for, at funksjonelle krav beskriver systemets oppførsel. Men ingen slike enigheter finnes for ikke-funksjonelle krav.

Kort oppsummert ligger det en svakhet i å si at krav som beskriver funksjoner og oppførsel er funksjonelle krav, og at krav som beskriver egenskaper og hvordan systemet er, er ikke-funksjonelle krav. Denne klassifiseringen kan synes å bli for grov.

Derfor er det er ønskelig å dele kravene inn i følgende 5 aspekter;

- type
- prioritet
- fremstilling
- tilfredsstillelse
- rolle

2.1.1 Type klassifisering

Krav kan klassifiseres etter hva de handler om. Et krav kan gjelde en funksjon eller data, dette som tradisjonelt kalles et funksjonelt krav. Videre kan krav beskrive ytelse i hastighet, volum, strøm, osv. Den tredje typen krav er de som beskriver bestemte egenskaper, som stabilitet og sikkerhet. De siste er krav som gir begrensinger, f.eks. at kostnadene på produktet ikke må overstige X antall kroner.

Type	Beskrivelse
Funksjonell	En funksjon som et system skal kunne gjøre, eller en data bit eller data struktur som skal være en del av systems tilstand
Ytelse	En krav som gjelder tid (punkter i tid, reaksjonstid, tidsintervaller), hastighet, volum, eller strømning (volum per tidsenhet)
Egenskap	Egenskaper som gjelder både produktets bruk (stabilitet, enkelt å bruke) og administrasjon (vedlikehold, brukervennlighet)
Begrensing	En designbeslutning eller designbegrensing innført av en investor (f.eks. momenter de som utvikler systemet ikke kan påvirke)

Tabell 3: Klassifiseringer av kravtyper

Etter tradisjonell tankegang blir den første typen de funksjonelle kravene, og de siste tre utgjør ikke-funksjonelle krav. I kapittel 3 Krav – oversikt er kravene listet opp, først etter prioritet og deretter om de er funksjonelle eller ikke-funksjonelle. Tabell 5 gir en metode for å bestemme typen til et vilkårlig krav.

nr.	Spørsmål Er dette kravet nevnt fordi vi trenger å beskrive...	Resultat
1.	...noe av systemets oppførsel, data, input eller reaksjoner til stimulus – uansett den måten dette blir gjort?	Funksjonell
2.	...begrensinger om timing, behandling, eller reaksjonstid, datavolum eller strømning?	Ytelse
3.	...en bestemt egenskap som systemet eller komponenten skal ha?	Egenskap
4.	...noen andre begrensinger på hva systemet skal gjøre, hvordan det skal gjøres eller en noe beskrevet løsning eller delløsning?	Begrensning

Tabell 4: Kravtypemetoden

2.1.2 Prioritet klassifisering

Utifra forstudierapporten har systemet fire prioriteter, som må taes med når kravene utledes. Disse er nummerert som 1, 2, 3 og 4. Se tabell 6.

Prioritet	Beskrivelse
1.	Første prioritet
2.	Andre prioritet
3.	Tredje prioritet
4.	Fjerde prioritet

Tabell 5: Prioritering av krav

2.1.3 Fremstilling klassifisering

Det er fire typiske fremstillingsformer til krav. Disse er bestemt av måten kravet blir kontrollert eller testet. Det er stort sett de operative og deklorative fremstillingene som bør kommenteres.

Krav som beskriver hva systemet forventes å gjøre blir typisk fremstilt på en operativ form, funksjoner å utføre, data å fremstille, tilstander å være i, osv. Disse kravene må granskes, testes eller få godkjenning.

Deklarative krav beskriver en plikt. Ofte står krav om data eller begrensinger på denne formen. Typiske «slik er det bare» krav, og godkjenningen er å se over resultatet.

Fremstillingen	Type kontrollering
Operativ	Granske, teste eller formelle godkjenninger
Kvantitativ	Måle (tallskala)
Kvalitativ	Ingen direkte godkjenning. Kan bli bestemt av investorers subjektive vurderinger av det utplasserte systemet.
Deklarativ	Gjennomgå, kontrollere

Tabell 6: Fremstilling av krav bestemmer kontrolleringsmåte

2.1.4 Tilfredsstillelse klassifisering

Når det gjelder vurderingen av tilfredsstillelse av krav kan disse klassifiseres. Enten er det slik at kravet er tilfredsstilt eller ikke tilfredsstilt, dette kalles hard. Hvis kravet er gradvis tilfredsstilt og graden til tilfredsstillelse kan måles numerisk kalles dette myk.

2.1.5 Rolle klassifisering

Det er to roller som krav kan ha. Fastsettende krav er de som gjelder bare systemet, prosjektet eller prosessen. I praksis betyr det at systemet (prosjektet eller prosessen) har full kontroll og/eller ansvar.

Kravene som er avhengig av informasjon utenfor systemet er normative krav. Ta f.eks. et krav at fartøyet skal være fjernstyrt. Prosjektet må følge reglene som beskriver hvilke frekvenser som er tillatt å bruke til fjernstyring.

Rollen	Type kontrollering
Fastsettende	Gjelder bare systemet
Normativ	Normer i systemets miljø

Tabell 7: Klassifisering av krav roller

2.2 Identifisering av krav

Alle krav får egen identifikasjonskode. Kodene er basert på tabell 8.

Kravprefiks:	Prioritet:	Kategorier:	Nummerering:
K - Markerer at identifiseringen er et krav	A – 1. prioritet B – 2. prioritet C – 3. prioritet D – 4. prioritet S – Slettet	4 – Dokumentasjon 5 – Reguleringsystem 6 – Maskinvare 7 – Fartøy 8 – Brukergrensesnitt 9 – Spesialfunksjoner	01-99

Tabell 8: Identifisering av krav

Eksempel på identifiseringer: KB499, KA501, KD533, KC422 eller KS605

3 Krav – oversikt

3.1 A Krav

KravID	Beskrivelse	UseCase	TestID
Funksjonelle			
KA401	Budsjettet skal være godkjent av Simicon AS.		TK501
KA601	Selve kroppen til fartøyet skal kjøpes ferdigutviklet.		TK405
KA602	De innkjøpt delene skal testes for å bekrefte at de fungerer som de skal.		TK106, TK406, TK107, TK108, TK109, TK110, TK407, TK201, TK112
KA603	Fartøyet skal konstrueres av innkjøpte deler. Delene som evt. utvikles må begrunnes.		TK408, TI422
KA701	Fartøyet skal fjernstyres med standard radioutstyr.		TK404
KA702	Fartøyet skal fly ved hjelp av fjernkontrollen. (dvs. manuell styring uten regulering)		TK601, TI701, TI702, TI703, TI704, TI709, TS105, TS801
KA801	Opp og ned på venstre stikke skal styre Thrust.		TS101
KA802	Side til side på venstre stikke skal styre Yaw.		TS102
KA803	Opp og ned på høyre stikke skal styre Pitch.		TS103
KA804	Side til side på høyre stikke skal styre Roll.		TS104
KA901	Det skal implementeres en «lavt-batteri-alarm». Batteriet kan eksplodere eller ta fyr om den blir tømt helt, eller blir brukt feil.		(TK606), TI707

Tabell 9: Alle A kravene

3.2 B Krav

KravID	Beskrivelse	UseCase	TestID
Funksjonelle			
KB402	Multikopterteori skal beskrives		TK401
KB403	Reguleringssystemet og simuleringer skal dokumenteres som en rapport.		TK401
KB501	Sett opp en matematisk modell for å vurdere reguleringsmetoder.		TK401
KB502	Det skal utvikles og implementeres et reguleringssystem.		TK602, TI705, TI710, TS113, TS801
KB503	Når reguleringen er på skal det alltid være mulig å styre med fjernkontrollen.		TS113, TS801
KB504	Reguleringssystem skal kunne slås av/på fra fjernkontrollen.		TS114
KB505	Reguleringssystemet skal være stabilt.		TS113, TK401
KB506	Signalene fra sensorene må undersøkes, og evt. signalbehandles slik at det er brukbart for reguleringen.		TS113, TK401,
KB605	Vibrasjonsdempere skal benyttes mellom sensorer og ramme.		TS419
KB902	Fartøyets skal utstyres med en kameramodul for slik at man kan observere flyvningen gjennom kameraet.		TS420
Ikke-funksjonelle			
KB604	Bena på fartøyet skal farges slik at det er enkelt å se hva slags orientering fartøyet har i luften.		TK421
KB903	Fartøyets videostream skal kunne lagres på en datamaskin.		TS423

Tabell 10: Alle B kravene

3.3 C Krav

KravID	Beskrivelse	UseCase	TestID
Funksjonelle			
KC507	Ved tap av signal i en periode over ett sekund skal fartøyet utføre en nødlanding.		TK604, TK605, TI711, TS117, TS801
KC904	Det skal implementeres en flytidsalarm. Tidspunktet til alarmen skal være bestemt utifra testresultater på gjennomsnittlig flytid.		TK603, TI713, TS118, TS801
KC905	Om fartøyet er i luften 20 sekunder etter at «lavtbatterialarmen» har gått skal reguleringen ta over med en nødlandingsfunksjon for å sikre at batteriet ikke blir tappet helt ut.		TK604, TK606, TI707, TI712, TS119, TS801

Tabell 11: Alle C kravene

3.4 D Krav

KravID	Beskrivelse	UseCase	TestID
Funksjonelle			
KD508	Fartøyet skal kunne regulere høyden.		TI708, TS120, TS801

Tabell 12: Alle D kravene

4 Krav – detaljer

Her kommer en rekke tabeller som inneholder kravene med detaljer. Først et eksempel.

Identifisering	
ID	[K/A/B/C/D/S/XXX]
Beskrivelse	[Beskrivelse]
Kategori	[Kategori navn]
Kryssreferanser	
UseCase	[UsecaseID]
TestID	[TestID]
Informasjon	
Opphav til krav	[Kilden til kravet]
Dato	[åååå-mm-dd]
Type	[Funksjonelt/Ytelse/Egenskap/Begrensning]
Prioritet	[A/B/C/D/S]
Fremstilling	[Operativt/Kvantitativt/Kvalitativt/Deklarativt]
Tilfredsstillelse	[Hard/Myk]
Rolle	[Fastsettende/Normativt]
Status	[Foreslått/Mottatt/Utgått]
Referanser	[Lenker til relevante dokumenter]
Kommentarer	[Tekst]

Tabell 13: Eksempelmal

4.1 Dokumentasjon**KA401**

Identifisering	
ID	KA401
Beskrivelse	Budsjettet skal være godkjent av Simicon AS.
Kategori	Dokumentasjon
Kryssreferanser	
UseCase	
TestID	TS501
Informasjon	
Opphav til krav	Simicon AS - Ragnvald Otterlei
Dato	14.09.2010
Type	Egenskap
Prioritet	A
Fremstilling	Deklarativt
Tilfredsstillelse	Hard
Rolle	Fastsettende
Status	Mottatt
Referanser	
Kommentarer	Var KA122 i v3.0

KB402

Identifisering	
ID	KB402
Beskrivelse	Multikopterteori skal beskrives
Kategori	Dokumentasjon
Kryssreferanser	
UseCase	
TestID	TK401
Informasjon	
Opphav til krav	Prosjektgruppen
Dato	2010-11-02
Type	Funksjonell
Prioritet	A
Fremstilling	Deklarativt
Tilfredsstillelse	Hard
Rolle	Fastsettende
Status	Foreslått
Referanser	
Kommentarer	Var KB120 i v3.0

KB403

Identifisering	
ID	KB403
Beskrivelse	Reguleringssystemet og simuleringer skal dokumenteres som en rapport.
Kategori	Dokumentasjon
Kryssreferanser	
UseCase	
TestID	TK401
Informasjon	
Opphav til krav	Prosjektgruppen
Dato	02.11.2010
Type	Funksjonell
Prioritet	B
Fremstilling	Deklarativt
Tilfredsstillelse	Myk
Rolle	Fastsettende
Status	Foreslått
Referanser	
Kommentarer	Var KB121 i v3.0

Identifisering	
ID	KB501
Beskrivelse	Sett opp en matematisk modell for å vurdere reguleringsmetoder.
Kategori	Reguleringsystem
Kryssreferanser	
UseCase	
TestID	TK401
Informasjon	
Opphav til krav	Simicon AS - Ragnvald Otterlei
Dato	08.11.2010
Type	Funksjonelt
Prioritet	B
Fremstilling	Deklarativ
Tilfredsstillelse	Myk
Rolle	Fastsettende
Status	Mottatt
Referanser	
Kommentarer	Var KB208 i v3.0

Identifisering

ID	KB502
Beskrivelse	Det skal utvikles og implementeres et reguleringsystem.
Kategori	Reguleringsystem
Kryssreferanser	
UseCase	
TestID	TS113
Informasjon	
Opphav til krav	Simicon AS - Ragnvald Otterlei
Dato	08.11.2010
Type	Funksjonelt
Prioritet	B
Fremstilling	Operativt
Tilfredsstillelse	Myk
Rolle	Fastsettende
Status	Mottatt
Referanser	
Kommentarer	Var KB209 i v3.0

Identifisering	
ID	KB503
Beskrivelse	Når reguleringen er på skal det alltid være mulig å styre med fjernkontrollen.
Kategori	Reguleringssystem
Kryssreferanser	
UseCase	
TestID	TS113
Informasjon	
Opphav til krav	Prosjektgruppen
Dato	2010-11-02
Type	Funksjonelt
Prioritet	B
Fremstilling	Operativ
Tilfredsstillelse	Hard
Rolle	Fastsettende
Status	Foreslått
Referanser	
Kommentarer	Var KB210 i v3.0

Identifisering	
ID	KB505
Beskrivelse	Reguleringssystemet skal være stabilt.
Kategori	Reguleringssystem
Kryssreferanser	
UseCase	
TestID	TK401, TS113
Informasjon	
Opphav til krav	Prosjektgruppen
Dato	2010-11-02
Type	Egenskap
Prioritet	B
Fremstilling	Deklarativ
Tilfredsstillelse	Hard
Rolle	Normativ
Status	Foreslått
Referanser	
Kommentarer	Var KB212 i v3.0

Identifisering	
ID	KB506

Beskrivelse	Signalene fra sensorene må undersøkes, og evt. signalbehandles slik at det er brukbart for reguleringen.
Kategori	Reguleringssystem
Kryssreferanser	
UseCase	
TestID	TK401, TS113
Informasjon	
Opphav til krav	Prosjektgruppen
Dato	02.11.2010
Type	Funksjonelt
Prioritet	B
Fremstilling	Kvalitativt
Tilfredsstillelse	Myk
Rolle	Fastsettende
Status	Foreslått
Referanser	
Kommentarer	Var KB213 i v3.0

Identifisering	
ID	KC507
Beskrivelse	Ved tap av signal i en periode på over ett sekund, skal fartøyet utføre en nødlanding.
Kategori	Reguleringssystem
Kryssreferanser	
UseCase	
TestID	TS117
Informasjon	
Opphav til krav	Prosjektgruppen
Dato	2010-11-02
Type	Funksjonelt
Prioritet	C
Fremstilling	Operativ
Tilfredsstillelse	Hard
Rolle	Fastsettende
Status	Mottatt
Referanser	
Kommentarer	Var KC203 i v3.0

Identifisering	
ID	KB508
Beskrivelse	Fartøyet skal kunne regulere høyden.
Kategori	Reguleringssystem
Kryssreferanser	
UseCase	
TestID	TS120
Informasjon	
Opphav til krav	Prosjektgruppen
Dato	02.11.2010
Type	Funksjonelt
Prioritet	D
Fremstilling	Kvalitativt
Tilfredsstillelse	Myk
Rolle	Fastsettende
Status	Mottatt
Referanser	
Kommentarer	Var KD214 i v3.0

Identifisering	
ID	KA601
Beskrivelse	Selve kroppen til fartøyet skal kjøpes ferdigutviklet.
Kategori	Maskinvare
Kryssreferanser	
UseCase	
TestID	TK405
Informasjon	
Opphav til krav	Simicon AS - Ragnvald Otterlei
Dato	2010-06-14
Type	Begrensing
Prioritet	A
Fremstilling	Deklarativ
Tilfredsstillelse	Hard
Rolle	Fastsettende
Status	Mottatt
Referanser	
Kommentarer	Var KA204 i v3.0

Identifisering	
ID	KA602
Beskrivelse	Innkjøpte delene skal testes for å bekrefte at de fungerer som de skal.
Kategori	Maskinvare
Kryssreferanser	
UseCase	
TestID	TK106, TK406, TK107, TK108, TK109, TK110, TK407, TK201, TK112, TK120
Informasjon	
Opphav til krav	Prosjektgruppen
Dato	2010-11-02
Type	Egenskap
Prioritet	A
Fremstilling	Deklarativ
Tilfredsstillelse	Hard
Rolle	Fastsettende
Status	Foreslått
Referanser	
Kommentarer	Var KA205 i v3.0

Identifisering	
ID	KA603
Beskrivelse	Fartøyet skal konstrueres av innkjøpte deler. Delene som evt. utvikles må begrunnes.
Kategori	Maskinvare
Kryssreferanser	
UseCase	
TestID	TK408
Informasjon	
Opphav til krav	Simicon AS - Ragnvald Otterlei
Dato	2010-09-14
Type	Egenskap
Prioritet	A
Fremstilling	Deklarativ
Tilfredsstillelse	Myk
Rolle	Fastsettende
Status	Mottatt
Referanser	
Kommentarer	Var KA206 i v3.0

Identifisering	
ID	KB604
Beskrivelse	Bena på fartøyet skal farges slik at det er enkelt å se hva slags orientering fartøyet har i luften.
Kategori	Maskinvare
Kryssreferanser	
UseCase	
TestID	TK421
Informasjon	
Opphav til krav	Prosjektgruppen
Dato	2010-11-02
Type	Begrensning
Prioritet	B
Fremstilling	Deklarativt
Tilfredsstillelse	Myk
Rolle	Fastsettende
Status	Foreslått
Referanser	
Kommentarer	Var KB207 i v3.0

Identifisering	
ID	KB605
Beskrivelse	Vibrasjonsdempere skal benyttes mellom sensorer og ramme.
Kategori	Maskinvare
Kryssreferanser	
UseCase	
TestID	TS419
Informasjon	
Opphav til krav	Prosjektgruppen
Dato	2010-11-02
Type	Funksjonelt
Prioritet	B
Fremstilling	Kvalitativt
Tilfredsstillelse	Myk
Rolle	Fastsettende
Status	Foreslått
Referanser	
Kommentarer	Var KB217 i v 3.0

Identifisering	
ID	KA701
Beskrivelse	Fartøyet skal fjernstyres med standard radioutstyr.
Kategori	Fartøyet
Kryssreferanser	
UseCase	
TestID	TK404
Informasjon	
Opphav til krav	Simicon AS - Ragnvald Otterlei
Dato	2010-09-14
Type	Begrensing
Prioritet	A
Fremstilling	Deklarativ
Tilfredsstillelse	Hard
Rolle	Fastsettende
Status	Mottatt
Referanser	
Kommentarer	Var KA201 i v3.0

Identifisering	
ID	KA702
Beskrivelse	Fartøyet skal fly bemannet ved hjelp av fjernkontrollen. (dvs. manuell styring uten regulering)
Kategori	Fartøyet
Kryssreferanser	
UseCase	
TestID	TS105
Informasjon	
Opphav til krav	Simicon AS - Ragnvald Otterlei
Dato	2010-09-14
Type	Funksjonelt
Prioritet	A
Fremstilling	Operativ
Tilfredsstillelse	Myk
Rolle	Fastsettende
Status	Mottatt
Referanser	
Kommentarer	Var KA202 i v3.0

Identifisering	
ID	KA801
Beskrivelse	Opp og ned på venstre stikke skal styre Thrust.
Kategori	Brukergrensesnitt
Kryssreferanser	
UseCase	
TestID	TS101
Informasjon	
Opphav til krav	Simicon AS - Ragnvald Otterlei
Dato	08.11.2010
Type	Funksjonelt
Prioritet	A
Fremstilling	Deklarativt
Tilfredsstillelse	Hard
Rolle	Fastsettende
Status	Mottatt
Referanser	
Kommentarer	Var KA220 i v3.0

Identifisering	
ID	KA802
Beskrivelse	Side til side på venstre stikke skal styre Yaw.
Kategori	Systemkrav
Kryssreferanser	
UseCase	
TestID	TS102
Informasjon	
Opphav til krav	Simicon AS - Ragnvald Otterlei
Dato	2010-11-08
Type	Funksjonelt
Prioritet	A
Fremstilling	Deklarativt
Tilfredsstillelse	Hard
Rolle	Fastsettende
Status	Mottatt
Referanser	
Kommentarer	Var KA221 i v3.0

Identifisering	
ID	KA803
Beskrivelse	Opp og ned på høyre stikke skal styre Pitch.
Kategori	Brukergrensesnitt
Kryssreferanser	
UseCase	
TestID	TS103
Informasjon	
Opphav til krav	Simicon AS - Ragnvald Otterlei
Dato	08.11.2010
Type	Funksjonelt
Prioritet	A
Fremstilling	Deklarativt
Tilfredsstillelse	Hard
Rolle	Fastsettende
Status	Mottatt
Referanser	
Kommentarer	Var KA222 i v3.0

Identifisering	
ID	KA804
Beskrivelse	Side til side på høyre stikke skal styre Roll.
Kategori	Brukergrensesnitt
Kryssreferanser	
UseCase	
TestID	TS104
Informasjon	
Opphav til krav	Simicon AS - Ragnvald Otterlei
Dato	2010-11-08
Type	Funksjonelt
Prioritet	A
Fremstilling	Deklarativt
Tilfredsstillelse	Hard
Rolle	Fastsettende
Status	Mottatt
Referanser	
Kommentarer	Var KA223 i v3.0

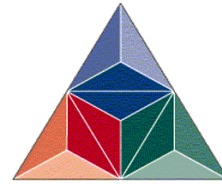
Identifisering	
ID	KA901
Beskrivelse	Det skal implementeres en «lavt-batteri-alarm». Batteriet kan eksplodere eller ta fyr om den blir tappet helt ut.
Kategori	Spesialfunksjoner
Kryssreferanser	
UseCase	
TestID	TS116
Informasjon	
Opphav til krav	Simicon AS - Ragnvald Otterlei
Dato	2010-11-08
Type	Funksjonelt
Prioritet	A
Fremstilling	Operativt
Tilfredsstillelse	Hard
Rolle	Normativt
Status	Mottatt
Referanser	
Kommentarer	Var KA216 i v3.0

Identifisering	
ID	KB902
Beskrivelse	Fartøyets skal utstyres med en kameramodul for slik at man kan observere flyvningen gjennom kameraet.
Kategori	Spesialfunksjoner
Kryssreferanser	
UseCase	
TestID	TS420
Informasjon	
Opphav til krav	Prosjektgruppen
Dato	2010-11-02
Type	Funksjonelt
Prioritet	B
Fremstilling	Operativt
Tilfredsstillelse	Hard
Rolle	Fastsettende
Status	Foreslått
Referanser	
Kommentarer	Var KB218 i v3.0

Identifisering	
ID	KB903
Beskrivelse	Fartøyets videostream skal kunne lagres på en datamaskin.
Kategori	Spesialfunksjoner
Kryssreferanser	
UseCase	
TestID	TS420
Informasjon	
Opphav til krav	Prosjektgruppen
Dato	02.11.2010
Type	Egenskap
Prioritet	B
Fremstilling	Operativt
Tilfredsstillelse	Myk
Rolle	Fastsettende
Status	Foreslått
Referanser	
Kommentarer	Var KB219 i v3.0

Identifisering	
ID	KC904
Beskrivelse	Det skal implementeres en flytidsalarm. Tidspunktet til alarmen skal være bestemt utifra testresultater på gjennomsnittlig flytid.
Kategori	Spesialfunksjoner
Kryssreferanser	
UseCase	
TestID	TS118
Informasjon	
Opphav til krav	Prosjektgruppen
Dato	2010-11-02
Type	Funksjonelt
Prioritet	C
Fremstilling	Operativt
Tilfredsstillelse	Hard
Rolle	Fastsettende
Status	Foreslått
Referanser	
Kommentarer	Var KC215 i v3.0

Identifisering	
ID	KC905
Beskrivelse	Om fartøyet fremdeles er i luften 20 sekunder etter «lavt-batteri-alarmeren» har gått skal reguleringen ta over med en nødlandingsfunksjon for å sikre at batteriet ikke blir utladet.
Kategori	Spesialfunksjoner
Kryssreferanser	
UseCase	
TestID	TS119
Informasjon	
Opphav til krav	Prosjektgruppen
Dato	2010-12-21
Type	Funksjonelt
Prioritet	C
Fremstilling	Operativ
Tilfredsstillelse	Hard
Rolle	Fastsettende
Status	Foreslått
Referanser	
Kommentarer	Var KC224 i v3.0



HØGSKOLEN
i Buskerud

Oppgavetittel: Hovedprosjekt - Testplan - v2.0		
Fag(nr./navn): SFHO 3200 / Prosjektstyring med hovedprosjekt		
Gruppemedlemmer, signatur:		
_____ Vegard Torkelsen	_____ Michael Odden	_____ Herå Rørvik
_____ Trine Lindberg	_____ Luke Carambot	_____ Patrick Bjørum
Intern veileder: Dag Samuelsen Intern sensor: Olaf Hallan Graven Ekstern veileder og sensor: Ragnvald Otterlei		
Dato: 2011-01-03		

Innholdsfortegnelse

Dokumenthistorie.....	1
Definisjoner og forkortelser.....	1
Innledning.....	1
1 Elementene som skal testes	2
1.1 Prosjekt.....	2
1.1.1 Dokumentasjon.....	2
1.1.2 Presentasjoner.....	2
1.2 System	2
1.2.1 Fartøyet	2
1.2.2 Maskinvare	3
1.2.3 Reguleringsystemet	3
1.3 Prosess	3
2 Teststrategi	4
3 Generelle Testemetoder	4
3.1 Black-box testing.....	4
3.2 White-box testing	4
3.3 Grensesnitt testing	4
4 Testnivåer	5
5 Typer	6
5.1 Statisk testing	6
5.1.1 Målingstest	6
5.1.2 Vurderingstest	6
5.1.3 Kontrollering.....	6
5.1.4 Godkjenningstest	6
5.1.5 Kodeinspeksjon	6
5.2 Dynamisk testing	6
5.2.1 Funksjonstest	6
5.2.2 Regresjonstest	6
5.2.4 Grensesnitttest	6
5.2.5 Målingstest	6
5.2.6 Vurderingstest	7
6 Test dokumentasjon	7
6.1 Testspesifikasjon	7
6.2 Testlogg	7
6.3 Testrapporter	7
7 Oppgaver rundt testing	7
7.1 Testbenk til fartøyet	7
7.2 Testrom til fartøyet	8
8 Ansvar	8
8.1 Testansvarlig	8
8.2 Testdesign	8
8.3 Testing	8
8.4 Testanalyse	8
9 Resurser	8
9.1 Personell	8
9.2 Programvare	8
9.3 Maskinvare	9

9.4 Verktøy	9
9.5 Fasiliteter	9
9.6 Data	9
10 Begrensing og risiko	9
10.1 Begrensinger	9
10.2 Risikoer	9
11 Referanser	10

Dokumenthistorie

Versjon	Beskrivelse	Når
1.0	Første offisielle versjon	2011-01-03
2.0	Lagt til mal	2011-05-27

Tabell 1: Dokumenthistorie

Definisjoner og forkortelser

ESC	Estimated Speed Controller
I2C	Inter-Integrated Circuit
IDE	Integrated Development Environment
Prototype 1.	Fartøyet uten regulering
Prototype 2.	Fartøyet med regulering
Prototype 3.	Fartøyet oppgradert med spesielle funksjoner

Tabell 2: Definisjoner og forkortelser

Innledning

Dette er testplanen til Project Dragonfly. Dokumentet gir en oversikt over hvordan testing er planlagt. Dokumentet danner også grunnlaget til testspesifikasjonen som er en leveranse i første iterasjon. Hovedpoenget er å identifisere elementene som skal testes under prosjektet, legge ned teststrategi og -metoder, og å si litt om de forskjellige testtypene som skal brukes for å kontrollere at alle kravene i kravspesifikasjonen er dekket. I tillegg nevnes testnivåer for å kunne dele opp testingen på en mer oversiktlig måte i testspesifikasjonen. Testdokumentasjon som dette dokumentet gir grunnlag til er tatt opp i kapittel 5. De siste fire kapitlene tar opp ekstraordinære forberedelser til noe tester, ansvarsroller, resurskrav og litt om begrensinger og risikoer tilknyttet testing.

Det som er interessant i dette dokumentet er:

- Elementene som skal testes i [kapittel 1](#)
- Teststrategien i [kapittel 2](#)
- Kapitlene 4 og 6 om [Testnivåer](#) og [Dokumentasjon](#)
- Begrensinger og risikoer i [kapittel 10](#)

1 Elementene som skal testes

1.1 Prosjekt

1.1.1 Dokumentasjon

- Visjonsdokument
- Forstudierapport
- Kravspesifikasjon
- Prosjektplan
- Testplan
- Testspesifikasjon
- Iterasjonsplaner
- Iterasjonsrapporter
- Analysedokument
- Designdokument
- Testresultater
- Etteranalyse
- Kontrakt
- Kodestandard
- Budsjett
- Multikopterteori
- Innkalling til presentasjonene
- Språk
- Digital formater
- Dokument form
- Hjemmesiden

1.1.2 Presentasjoner

- Presentasjon 1
- Presentasjon 2
- Hovedpresentasjon

1.2 System

1.2.1 Fartøyet

- Fjernkontrollen
- Manuell styring av fartøyet (uten regulering)
- Nødlandingsfunksjon ved tapt styringssignal
- Responsen ved bevegelsene på stikkene på fjernkontrollen (uten og med regulering)
 - Venstrestikken og Høyrestikken
 - Side til side

- Opp og ned
- Nullstilling
- Mikrokontroller funksjoner
 - Omformingen av 4 innganger til 4 utganger (stikkeposisjoner til motorhastigheter)
- Flytidsalarm
 - Gjennomsnittsflytid
 - Alarmen

1.2.2 Maskinvare

- Innkjøpte deler
 - Chassi
 - Motorer
 - ESCene
 - Gyro
 - Akselerometer
 - Mikrokontroller
 - Batteri
- Lavtbatterialarm
- Montering av kamera
- Lagring av videostream

1.2.3 Reguleringsystemet

- Matematisk modell
- Reguleringsystem
- Stabiliteten til reguleringsystemet
- Av/på funksjonen til reguleringsystemet
- Sensorsignalene (evt. behandlede signaler)

1.3 Prosess

- Oppfølgingsdokumenter
 - Ukentlig
 - 4 ukers versjon
- Møter
 - Møterom
 - Innkalling
 - Møtereferater
 - Veiledermøter
- Timelister
 - Timelister

2 Teststrategi

Det er flere teststrategier å velge mellom [1]. Strategiene som forklares her er de som er planlagt å bruke i prosjektet.

Hovedstrategien blir inkrementell testing som starter med bottom-up testing. De kritiske, eller mest grunnleggende elementene testes først. Så langt det er mulig blir modulene testet alene. Deretter blir det testing på hele systemet ettersom moduler blir testet og lagt til.

Stresstesting går ut på å belaste systemet og observere hvordan det håndterer belastningen. Kravspesifikasjonen hittil gir ikke noen behov for stresstesting. Back-to-back testing kan brukes der det er like deler som skal testes. Alle delene kan da gis samme input, og resultatene kan sammenlignes. Et eksempel på når dette kan brukes er testing av motorene.

3 Generelle Testemetoder

Mesteparten av testene kommer til å bruke black-box metoden, white-box metoden eller begge deler. Grensesnitt-testing kommer til å bli lite brukt, fordi nesten alle grensesnittene i dette prosjektet er definert fra før av.

3.1 Black-box testing

Testene etter denne metoden er laget direkte fra kravspesifikasjonen. Disse er også kalt funksjonell testing fordi det som er forventet på utgangen av testelementet er et resultat av kjent data på inngangen. Meningen med black-box er at oppbygningen til elementet er ukjent. Bestemt data inn skal gi noe bestemt data ut.

3.2 White-box testing

Denne metoden brukes når oppbygningen til elementet er kjent. Her er det mulig å analysere testelementet for å bestemme nøyaktig hvordan testen skal bygges. Nødvendigvis må de som lager testene ved denne metoden ha den kunnskap om oppbygningen av testelementet.

3.3 Grensesnitt testing

Grensesnitt testing er å teste koblingene mellom modulene i et system. Systemer med ulike komponenter vil ofte ha et grensesnitt mellom komponentene for å sikre kommunikasjon. Disse testes for feil ved denne metoden.

4 Testnivåer

Det er ønskelig å dele testene i tre testnivåer. Disse nivåer kommer til å gi bedre oversikt til testspesifikasjonen.

Testnivå	Beskrivelse
Komponent	Testene er white-box tester og bekrefter funksjonaliteten eller kvaliteten av komponenten.
Integrasjon	Typiske tester er grensesnitttester, men også alle andre tester laget for å teste at to eller flere komponenter fungerer sammen.
System	Testing av systemet i dets helhet, at det dekker kravspesifikasjonen.

Tabell 3: Testnivåer

Alle tre testnivåer kommer til å utføres gjennom iterasjonene fra og med iterasjon 2.

Iterasjon 2:

- Komponenttesting av alle enkeltdeler

Iterasjon 3:

- Integrasjonstesting av alle delene til prototype 1. (fjernkontroll-mottaker, mottaker-mikrokontroller, mikrokontroller-motorene, batteri-koblingsbrettet, osv.)
- Systemtesting på prototype 1.

Iterasjon 5:

- Komponenttesting av alle enkeltdeler (matematisk modell, reguleringsstystem, osv.)
- Integrasjonstesting av delene lagt sammen til prototype 2. (mikrokontroller-sensorene)
- Systemtesting på prototype 2.

Iterasjon 6:

- Komponenttesting av alle enkeltdeler
- Integrasjonstesting av oppgraderingsfunksjoner lagt til prototype 3.
- Systemtesting på prototype 3.

Iterasjon 7:

- Systemtesting på prosjektet.

5 Typer

5.1 Statisk testing

Statistiske tester er en strukturert gjennomgang av testelementet hvor det blir sjekket mot nødvendige spesifikasjoner og dokumentasjon. Disse testene utføres i nesten alle fasene av prosjektet.

5.1.1 Målingstest

Testen innebærer å måle noe ved tallverdier. Den statistiske versjonen ville bety at testen kontrolleres mot dokumentasjon. Black-box metoden brukes.

5.1.2 Vurderingstest

Testene som ikke kan måles ved tallverdier og må vurderes mot kriterier. Metoden kan være både black- og white-box.

5.1.3 Kontrollering

Kontrollere at en bekreftelse eller visse egenskaper er tilstede, eller at noe eksisterer. Metoden kan være både black- og white-box.

5.1.4 Godkjenningstest

Kontrollere at det finnes bekreftelse på godkjenning fra noen uten for prosjektet.

5.1.5 Kodeinspeksjon

Statisk testing av laget kode. Det vil si at den som tester skal gjennomgå koden og se etter feil i koden. Testen bruker white-box metoden.

5.2 Dynamisk testing

Dynamisk testing har med det dynamiske delen av systemet/produktet å gjøre. Det må være mulig å kjøre/bruke/simulere produktet for å få gjennomført testene. Disse type testene er stort sett brukt i de senere fasene i prosjektet hvor store nok deler av systemet har blitt laget.

5.2.1 Funksjonstest

Testen brukes for å teste at noe gjør det det skal. Her brukes black-box metoden.

5.2.2 Regresjonstest

Test om funksjoner som har fungert har slutet å fungere på grunn av oppdatering. Testene er også etter black-box metoden.

5.2.4 Grensesnitttest

Testing av overgang mellom to eller flere delsystemer. Testene skal være etter white-box metoden.

5.2.5 Målingstest

Testen innebærer å måle noe ved tallverdier. Den dynamiske versjonen er en test hvor systemet blir testet/brukt/simulert for å få målingen. Testene vil stort sett basere seg på black-box metoden.

5.2.6 Vurderingstest

Testene som ikke kan måles ved tallverdier og må vurderes mot kriterier. Den dynamiske versjonen krever at systemet blir testet/brukt/simulert for å få et resultat å vurdere. Metoden kan være både black- og white-box.

6 Test dokumentasjon

6.1 Testspesifikasjon

Det dokumentet som skal inneholde alle testscenarier for å dekke testing av elementene i testplanen mot kravspesifikasjonen. Det skal gi detaljene utover testplanen slik at testen kan utføres.

6.2 Testlogg

En testlogg skal dokumentere hver enkelt test som blir utført. Den skal inneholde hvem som har gjort den, når den ble gjort, hva resultat ble, samt en vurdering.

6.3 Testrapporter

Testene som utgjør en modul skal rapporteres i en testrapport. Disse rapporter skal skape en oversikt over alle testene ved å oppsummere testene i en modul. Modulene er listet i tabellen nedenfor:

Prosjekt	System	Prosess
<ul style="list-style-type: none"> • Dokumentasjon • Presentasjoner 	<ul style="list-style-type: none"> • Fartøyet • Maskinvare • Reguleringsystem 	<ul style="list-style-type: none"> • Oppfølgingsdokumenter • Møter • Timelister

Tabell 4: Testrapporter

7 Oppgaver rundt testing

Noen tester krever forberedelse ut over testprosedyren. Dette gjelder først og fremst systemkravene i prosjektet. Det vil si at det kreves et spesielt oppsett før det er klart til å kjøres en test. Dette kapittelet er et forsøk på å nevne disse situasjonene.

7.1 Testbenk til fartøyet

Til endelig implementasjonstesting av mikrokontrolleromformingene (med og uten reguleringsystem) og av reguleringsystemet er det behov for en testbenk for begrenning i bevegelsene til fartøyet. Selv om tester på kvaliteten på disse mikrokontroller-transformasjonene er utført, er det nødvendig med en begrenning i bevegelsene som en sikkerhetstiltak. Vi kan aldri være helt sikre på hva som kan skje når de godkjente signalene fra mikrokontrolleren faktisk styrer propellene, med høye turtall. Denne testbenken må derfor konstrueres.

7.2 Testrom til fartøyet

Fartøyet trenger plass når det er klart til å fly. Vi trenger et sikkert sted å fly slik at fartøyet ikke skader noe eller noen. Stedet må finnes og ordnes.

8 Ansvar

8.1 Testansvarlig

En person kommer til å ha hovedansvar for testing. Denne må:

- Sørge for at testene er planlagt og alt som trengs er ordnet.
- Sørge for at alle testrapporter er laget
- Holde all testdokumentasjon oppdatert og i orden

8.2 Testdesign

Den som har ansvar for testdesign skal:

- Identifisere testelementer
- Beskrive testframgang
- Dokumentere det forventede resultatet

Det kan ofte være nødvendig og jobbe sammen med den som utvikler delen som skal testes slik at man får oversikten og kunnskapen som kreves for å fullføre oppgaven.

8.3 Testing

Ansvar for testing kan taes av nesten hvem som helst. Oppgavene er:

- Følge testprosedyrer for å utføre en test
- Logg resultater

8.4 Testanalyse

Noen skal samle opp testresultatene som tilhører en modul (del av systemet) og analyse dem. Analysen skal føres til en testrapport.

9 Resurser

Alle resursene som trengs til testing.

9.1 Personell

- Testere

9.2 Programvare

- Matlab
- Simulink
- Office (Microsoft eller OpenOffice)
- Arduino 0021 IDE

9.3 Maskinvare

- Datamaskin
- Spenningskilde
- Spenning-/strømmåler
- Oscilloskop
- Batterilader

9.4 Verktøy

- til konstruksjon av testbenk
- (hjelm og kroppsbeskyttelse, livsforsikring) ;)

9.5 Fasiliteter

- Rom til flytesting

9.6 Data

- I2C protokoll
- § 21. Fjernstyring og § 8. Mobil videolink (lovdata.no)
- Processing/Wiring programmeringspråk

10 Begrensing og risiko

10.1 Begrensinger

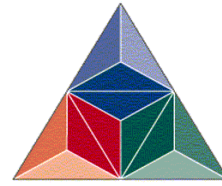
- Tid/arbeidskraft
- Penger
- Kunnskap/erfaring

10.2 Risikoer

- Ødelegge komponenter
- Skade person/gjenstander
- LiPo batterier
- Mangel på tid
- Utilstrekkelig planlegging
- Utilstrekkelig dokumentasjon

11 Referanser

- [1] Strøm, t., & Hallan Graven, O.(2010) Prosjekthåndbok: Kongsberg.
- [2] Dragonfly (2011) Forstudierapport: Kongsberg.
- [3] Dragonfly (2011) Prosjektplan: Kongsberg.
- [4] (2005) Forskrift som gir generelle tillatelser til bruk av frekvenser: Internet (03.01.2011).
<http://www.lovdatab.no/cgi-wift/ldles?doc=/sf/sf/sf-20050620-0715.html#21>
- [5] Processing hjemmeside: Internet (03.01.2011).
<http://www.processing.org/>
- [6] Wiring hjemmeside: Internet (03.01.2011).
<http://wiring.org.co/>



HØGSKOLEN
i Buskerud

Oppgavetittel:

Hovedprosjekt - Testspesifikasjon - v3.0

Fag(nr./navn):

SFHO 3200 / Prosjektstyring med hovedprosjekt

Gruppemedlemmer, signatur:

Vegard Torkelsen

Michael Odden

Herå Rørvik

Trine Lindberg

Luke Carambot

Patrick Bjørum

Intern veileder: Dag Samuelsen

Intern sensor: Olaf Hallan Graven

Ekstern veileder og sensor: Ragnvald Otterlei

Dato: 2011-05-19

Innholdsfortegnelse

Dokumenthistorie.....	1
Definisjoner og forkortelser.....	2
Referanser.....	2
Innledning.....	2
1 Om tester.....	3
1.1 Type tester	3
1.2 Identifisering av tester	3
2 Tester - oversikt	4
2.1 Komponenttesting.....	4
2.2 Integrasjonstesting	6
2.3 Systemtesting	7
3 Tester - detaljer	8
3.1 Funksjonstester	9
TS101.....	9
TS102	10
TS103.....	11
TS104.....	12
TS105.....	13
TK106.....	14
TK107.....	15
TK108.....	16
TK109.....	17
TK110.....	18
TK111.....	19
TK112.....	20
TS113.....	21
TS114.....	23
TS115.....	24
TS116.....	25
TS117.....	26
TS118.....	27
TS119.....	28
TK120.....	29
3.2 Målingstester	30
TK201.....	30
3.3 Vurderingstester.....	31
3.4 Kontrolleringstester.....	31
TK401.....	31
TK404.....	32
TK405	33
TK406.....	34
TK407.....	35
TK408.....	36
TS419.....	37
TS420.....	38
TK421.....	39
TI422.....	40

TK423.....	41
3.5 Godkjenningstester.....	42
TK501.....	42
3.6 Kodeinspeksjon.....	43
TK601.....	43
TK602.....	44
TK603.....	45
TK604.....	46
TK605.....	47
TK606.....	48
3.7 Grensesnittester.....	49
TI701.....	49
TI702.....	50
TI703.....	51
TI704.....	52
TI705.....	53
TI706.....	54
TI707.....	55
TI708.....	56
TI709.....	57
TI710.....	58
TI711.....	60
TI712.....	61
TI713.....	62
3.8 Regresjonstester.....	63
TS801.....	63

Dokumenthistorie

Versjon	Beskrivelse	Når
1.0	<p>Opprettet dokument</p> <p>Lagt til alle testene som trengs for å dekke kravene</p> <p>Fylt ut mange av oversiktstabellene</p> <p>Gjort ferdig kryssreferansene til kravspesifikasjonen</p> <p>Første offisiell versjon</p>	2011-01-03
2.0	<p>Flyttet alle tester som hadde med dokumenter/prosjektarbeid</p> <p>En del korrekturlesing</p> <p>Oppdatert TK106, TK107, TK108, TK109, TK110, TK111, TK112, TK113, TK114 og TK116, TK404, TK405 og TK406.</p> <p>Oppdatert TK118, TI701, TI702, TI703, TI704, TI705, TI706, TI707, TI709 og TI710.</p> <p>Opprettet TK423</p> <p>Lagt til ny målingstest TK202</p> <p>Ny test TK203, TK204</p> <p>Oppdatert TS101, 102, 103, 104</p> <p>Lagt til TK401 og oppdatert referansene til kravspesifikasjonen</p> <p>Oppdatert TS113, lagt til ny test TS106, TK205, TS201</p> <p>Lagt til ny test TK205-2</p> <p>Rettskrevet og ferdigstilt</p>	2011-04-08
3.0	<p>Slettet testene som ikke har direkte med krav å gjøre (TK202 ,TK203 ,TK204 ,TK205 ,TK205-2, TS106,TS201 som er målingstester inn i mellom)</p> <p>Oppdatert kryssreferanser</p> <p>Satt til mal</p>	2011-05-27

Tabell 1: Dokumenthistorie

Definisjoner og forkortelser

ESC	Electronic Speed Controller
Pitch	Rotasjon rundt lengden av «vingene» til fartøyet
Roll	Rotasjon rundt lengden av fartøyet
Yaw	Rotasjon rundt akselen som ville være høydemålingen til fartøyet

Tabell 2: Definisjoner og forkortelser

Referanser

- [1] Strøm, t., & Hallan Graven, O.(2010) Prosjekthåndbok: Kongsberg.
- [2] Dragonfly (2011) Kravspesifikasjon: Kongsberg.
- [3] Dragonfly (2011) Testplan: Kongsberg.
- [4] Dragonfly (2011) Prosjektplan: Kongsberg.

Innledning

Hensikten med dette dokument er å dokumentere alle testene som skal utføres. Det gir en konkret fremgangsmetode for å teste at alle kravene er oppfylt. Dokumentet er utledet i fra kravspesifikasjonen og testplanen.

Kapittel 1 starter med identifisering av testene og hvordan de er delt opp i dokumentet. En oversikt over alle testene, delt opp i testnivåer, er gitt i kapittel 2. Testprosedyrene og alle detaljene finnes i kapittel 3.

1 Om tester

Testtypene fra testplanen blir her nevnt en gang til, deretter en forklaring på test identifikasjon.

1.1 Type tester

- Funksjonstest - Teste at noe gjør det det skal
- Målingstest - Test som kan måles ved tall
- Vurderingstest - Test som ikke kan måles ved tall og må vurderes mot et kriterie
- Kontrollering - Sjekke at visse egenskaper er til stede, eller at noe eksisterer.
- Godkjenningstest - Sjekke at de finnes bekreftelse på godkjenning fra noen utenfor prosjektet
- Kodeinspeksjon - Statisk testing av laget kode
- Grensesnitttest - Testing av overgangen mellom to eller flere systemer
- Regresjonstest - Teste om funksjoner som har fungert har sluttet å fungere på grunn av oppdatering.

1.2 Identifisering av tester

Testprefiks	Nivå	Type	Nummerering
T - Markerer at identifiseringen er en test	K - Komponenttesting I - Integrasjonstesting S - Systemtesting	1 - Funksjon 2 - Målingstest 3 - Vurderingstest 4 - Kontrollering 5 - Godkjenningstest 6 - Kodeinspeksjon 7 - Grensesnitttest 8 - Regresjonstest	01 - 99

Tabell 3: Identifisering av tester

Eksempel identifiseringer: TK205, TS101, og TI811

2 Tester - oversikt

2.1 Komponenttesting

TestID	Beskrivelse	UseCase	KravID
TK106	Innkjøpte motorer test.		KA602
TK107	Innkjøpte ESC test.		KA602
TK108	Innkjøpte gyro test.		KA602
TK109	Innkjøpte akselerometer test.		KA602
TK110	Innkjøpte mikrokontroller test.		KA602
TK111	Fjernkontroll (sender - mottaker) test.		KA702, (KA602)
TK112	Teste lavt-batteri-alarmen.		KA602, KA901
TK120	Innkjøpt høydemåler test.		KD508
TK201	Innkjøpte batteri test.		KA602
TK401	Kontrollere at dokumentet [dokumentnavn] er skrevet.		KB402, KB403, KB501, KB505, KB506
TK404	Kontrollere standard radioutstyr.		KA701
TK405	Kontrollere kjøpt chassi.		KA601
TK406	Kontrollere innkjøpte propeller.		KA602
TK407	Kontrollere koblingsbrett.		KA602
TK408	Kontroller at fartøyet er konstruert av innkjøpte deler.		KA603
TK421	Kontroller at bena er farget slik at orienteringen til chassi er enkelt å se.		KB604
TK423	Testing av kameramodul og bakkestasjon.		KB902
TK501	Kontrollere godkjent budsjettet av Simicon AS.		KA401

TestID	Beskrivelse	UseCase	KravID
TK601	Kontroller transformeringen av fjernkontrollsignalene. Kodeblokken tar 4 innganger («3 rotasjoner og 1 retning» fra fjernkontrollen) og transformerer til pådrag på de 4 motorene.		KA702
TK602	Kontroller kode til reguleringsystemet.		KB502
TK603	Kontroller kode til flytids-alarmer.		KC904
TK604	Kontroller kode til nødlandingsfunksjonen.		KC507, KC905
TK605	Kontroller kode til tapt-signal-alarmer.		KC507
TK606	Kontroller kode til lavt-batteri-alarmer.		(KA901), KC905

Table 4: Komponenttesting

2.2 Integrasjonstesting

TestID	Beskrivelse	UseCase	KravID
TI422	Kontrollerer konstruksjonen av fartøyet, at alle deler er på plass og ferdig montert.		KA603
TI701	Teste integrasjon av fjernkontrollen og mottakeren.		KA702
TI702	Teste integrasjon av mottakeren og mikrokontrolleren.		KA702
TI703	Teste integrasjon av mikrokontrolleren og motorene (via ESCene).		KA702
TI704	Teste integrasjon av batteriet og koblingsbrettet.		KA702
TI705	Teste integrasjon av gyroen og mikrokontrolleren.		KB502
TI706	Teste integrasjon av akselerometeret og mikrokontrolleren.		KB502
TI707	Teste integrasjon mellom mikrokontrolleren og funksjonen for sjekk av batterinivå.		KA901, KC905
TI708	Teste integrasjon mellom mikrokontrolleren og høydemåleren.		KD508
TI709	Teste integrasjon av mikrokontrollerens transformering av fjernkontrollsignalet.		KA702
TI710	Teste integrasjon av mikrokontrollerens reguleringssystem. Signalene til motorene skal kontrolleres på oscilloskopet.		KB502
TI711	Teste integrasjon av mikrokontrollerens nødlandingsfunksjon og tapt fjernkontrollsignal.		KC507
TI712	Teste integrasjon av mikrokontrollerens nødlandingsfunksjon og lavt-batteri-nivå.		KC905
TI713	Teste integrasjon av mikrokontrollerens flytids-alarm.		KC904

Tabell 5: Integrasjonstester

2.3 Systemtesting

TestID	Beskrivelse	UseCase	KravID
TS101	Thrust test.		KA801
TS102	Yaw test.		KA802
TS103	Pitch test.		KA803
TS104	Roll test.		KA804
TS105	Fly test av fartøyet.		KA702
TS106	Fly test av fartøyet med lavere CG.		KA702
TS113	Teste reguleringsstytemet.		KB502, KB503, KB505, KB506
TS114	Teste reguleringsstytemets av/på funksjon fra fjernkontrollen.		KB504
TS115			
TS116	Teste visning og lagring av videostreamen fra fartøyet på en datamaskin.		KB902,KB903
TS117	Teste nødlandingsfunksjon ved tapt fjernkontrollsignal.		KC507
TS118	Teste flytids-alarmer.		KC904
TS119	Teste nødlandingsfunksjon, 20 sekunder etter lavt-batteri-alarmer.		KC905
TS419	Kontroller at vibrasjonsdempere brukes mellom sensorene og chassi.		KB605
TS420	Kontroller at kameramodulen er montert på fartøyet.		KB902, KB903
TS801	Flytest. Bekrefte at fartøyet fungerer som tidligere testet etter oppgradering av mikrokontroller programvare.		KA702, KB502, KB503, KC507, KC904, KC905, KD508

Tabell 6: Systemtester

3 Tester - detaljer

Identifisering	
ID:	[TK/I/SXXX]
Beskrivelse:	[Kort beskrivelse av testen]
Nivå:	[Test nivå]
Type test:	[Test typen]
Kryssreferanser	
UseCase:	[UseCaseID til UseCasene som inngår i testen]
KravID:	[KravID til kravene som inngår i testen]
Testdesign	
Kriterier:	<ul style="list-style-type: none"> [Avhengighet til testen]
Testbeskrivelse:	1. [Hvordan testen skal utføres]
Forventet Resultat:	[Forventet testresultat]
Informasjon	
Laget av:	[testdesignernavn] [åååå-mm-dd]
Kommentar:	[Tekst]

Tabell 7: Eksempelmal

3.1 Funksjontester

TS101

Identifisering	
ID:	TS101
Beskrivelse:	Thrust test
Nivå:	Systemtesting
Type test:	Funksjon
Kryssreferanser	
UseCase:	[UseCaseID til UseCasene som inngår i testen]
KravID:	KA801
Testdesign	
Kriterier:	<ul style="list-style-type: none"> Fartøyet skal kobles opp til testbenken for sikkerhets skyld. Fartøyet og fjernkontrollen er på. (Venstre stikken på fjernkontrollen skal være helt nede.)
Testbeskrivelse:	<ol style="list-style-type: none"> Rolig bevege bare venstre stikken oppover. Etter at fartøyet har lettet, senk venstre stikke litt slik at fartøyet har akkurat nok oppdrift til å holde seg nøytralt i luften. Ta venstre stikken helt ned. (rolig)
Forventet Resultat:	<ol style="list-style-type: none"> Fartøyet skal få mer thrust og evt. lette (rett oppover) Fartøyet skal opprettholde nøytral oppdrift og holde seg oppe uten å gå oppover eller nedover Fartøyet skal miste thrust og komme ned, testen over.
Informasjon	
Laget av:	Luke Carambot 2010-12-22, oppdatert av Patrick Bjørum 2011-02-23
Kommentar:	

TS102

Identifisering	
ID:	TS102
Beskrivelse:	Yaw test
Nivå:	Systemtesting
Type test:	Funksjon
Kryssreferanser	
UseCase:	[UseCaseID til UseCasene som inngår i testen]
KravID:	KA802
Testdesign	
Kriterier:	<ul style="list-style-type: none"> • Godkjent TS101 slik at fartøyet kan opprettholde nøytral oppdrift. • Fartøyet skal kobles opp til testbenken for sikkerhets skyld. • Fartøyet og fjernkontrollen er på. • (Venstre stikken på fjernkontrollen skal være helt nede.)
Testbeskrivelse:	<ol style="list-style-type: none"> 1. Rolig bevege venstre stikke oppover for å oppnå nøytral oppdrift. 2. I hover, beveg venstre stikken til høyre. 3. I hover, beveg venstre stikken til venstre. 4. Ta fartøyet ned. Testen ferdig.
Forventet Resultat:	<ol style="list-style-type: none"> 1. Fartøyet letter og oppnår nøytral oppdrift. 2. Fartøyet vrir kun rundt høydeaksen med klokka (sett ovenfra) når venstre stikken beveges til høyre. 3. Fartøyet vrir kun rundt høydeaksen mot klokka (sett ovenfra) når venstre stikken beveges til venstre. 4. Fartøyet lander, testen over.
Informasjon	
Laget av:	Luke Carambot 2010-12-22, oppdatert av Patrick Bjørum 2011-02-23
Kommentar:	

TS103

Identifisering	
ID:	TS103
Beskrivelse:	Pitch test
Nivå:	Systemtesting
Type test:	Funksjon
Kryssreferanser	
UseCase:	[UseCaseID til UseCasene som inngår i testen]
KravID:	KA803
Testdesign	
Kriterier:	<ul style="list-style-type: none"> • Godkjent TS101 slik at fartøyet kan opprettholde nøytral oppdrift. • Fartøyet skal kobles opp til testbenken for sikkerhets skyld. • Fartøyet og fjernkontrollen er på. • (Venstre stikken på fjernkontrollen skal være helt nede.)
Testbeskrivelse:	<ol style="list-style-type: none"> 1. Rolig bevegelse av venstre stikke oppover for å oppnå nøytral oppdrift. 2. I hover, høyre stikke bevegelse rolig framover. 3. I hover, høyre stikke bevegelse rolig bakover. 4. Venstres stikk bevegelse rolig nedover, fartøyet landes, testen er ferdig.
Forventet Resultat:	<ol style="list-style-type: none"> 1. Fartøyet letter og oppnår hover. 2. Fartøyet beveger fronten nedover og drar framover. 3. Fartøyet beveger fronten oppover og drar bakover. 4. Fartøyet lander, testen over.
Informasjon	Informasjon
Laget av:	Luke Carambot 2010-12-22, oppdatert av Patrick Bjørum 2011-02-23
Kommentar:	

TS104

Identifisering	
ID:	TS104
Beskrivelse:	Roll test
Nivå:	Systemtesting
Type test:	Funksjon
Kryssreferanser	
UseCase:	[UseCaseID til UseCasene som inngår i testen]
KravID:	KA804
Testdesign	
Kriterier:	<ul style="list-style-type: none"> • Godkjent TS101 slik at fartøyet kan opprettholde nøytral oppdrift. • Fartøyet skal kobles opp til testbenken for sikkerhets skyld. • Fartøyet og fjernkontrollen er på. • (Venstre stikken på fjernkontrollen skal være helt nede.)
Testbeskrivelse:	<p>5. Rolig bevegelse av venstre stikke oppover for å oppnå nøytral oppdrift.</p> <p>6. I hover, høyre stikke bevegelse rolig til høyre.</p> <p>7. I hover, høyre stikke bevegelse rolig til venstre.</p> <p>8. Venstres stikk bevegelse rolig nedover, fartøyet landes, testen er ferdig.</p>
Forventet Resultat:	<p>5. Fartøyet letter og oppnår hover.</p> <p>6. Fartøyet beveger høyre siden nedover og drar til høyre.</p> <p>7. Fartøyet beveger venstre siden nedover og drar til venstre</p> <p>8. Fartøyet lander, testen over.</p>
Informasjon	
Laget av:	Luke Carambot 2010-12-22, oppdatert av Patrick Bjørum 2011-02-23
Kommentar:	

TS105

Identifisering	
ID:	TS105
Beskrivelse:	Fartøy flytest
Nivå:	Systemtesting
Type test:	Funksjon
Kryssreferanser	
UseCase:	[UseCaseID til UseCasene som inngår i testen]
KravID:	KA702
Testdesign	
Kriterier:	<ul style="list-style-type: none"> • Fartøyet og fjernkontrollen er på. • Godkjent TS101, TS102, TS103 og TS104. • Har et rom å teste fartøyet i for å unngå skader på testperson, gjenstander og fartøyet • (Venstre stikken skal være helt ned)
Testbeskrivelse:	<ol style="list-style-type: none"> 1. Rolig prøve seg frem på aksene til hver stikke. Start med retning oppover på venstre stikke. 2. Hvis steg 1. gikk bra, test alle posisjoner til høyre stikke. Ta det forsiktig i tilfelle andre reaksjoner kommer av kombinerte stikkebevegelser. 3. Fly fartøyet rundt i luften i 5 minutter, for å kontrollere at det ikke utvikler seg noen feil over tid. 4. Testen er ferdig
Forventet Resultat:	Fartøyet oppfører seg som forventet ved justeringer på hver stikke om gangen. Fartøyet skal bevege seg bestemt etter styringen ifra fjernkontrollen. Det er mulig å styre fartøyet i luften i 5 minutter.
Informasjon	
Laget av:	Luke Carambot 2010-12-22
Kommentar:	

TK106

Identifisering	
ID:	TK106
Beskrivelse:	Innkjøpte motorer test
Nivå:	Komponenttesting
Type test:	Funksjon
Kryssreferanser	
UseCase:	[UseCaseID til UseCasene som inngår i testen]
KravID:	KA602
Testdesign	
Kriterier:	<ul style="list-style-type: none"> • Har et batteri eller spenningskilde (maks 11V) • Et program har blitt utviklet på mikrokontrolleren for å kjøre motoren gjennom ESCen. • Mikrokontrolleren er programmert med testprogram.
Testbeskrivelse:	<ol style="list-style-type: none"> 1. Koble motoren til ESCen (tre tykke ledninger) 2. Koble den tynne hvite ledningen til pin9 på mikrokontrolleren 3. Koble den tynne svarte ledningen til gnd på mikrokontrolleren 4. Koble den tynne røde ledningen til Vin på mikrokontrolleren 5. Koble opp batteriet (eller 11V spenning) til ESCen
Forventet Resultat:	Motoren går rolig fra stillestående til høy hastighet, deretter rolig ned igjen til den står stille. (programmet kjører i løkke)
Informasjon	
Laget av:	Luke Carambot 2011-01-14
Kommentar:	

TK107

Identifisering	
ID:	TK107
Beskrivelse:	Innkjøpte ESC test
Nivå:	Komponenttesting
Type test:	Funksjon
Kryssreferanser	
UseCase:	[UseCaseID til UseCasene som inngår i testen]
KravID:	KA602
Testdesign	
Kriterier:	<ul style="list-style-type: none"> • Har en fungerende motor. • Et program har blitt utviklet på mikrokontrolleren for å teste en ESC vha. en motor.
Testbeskrivelse:	<ol style="list-style-type: none"> 1. Koble motoren til ESCen (tre tikke ledninger) 2. Koble den tynne hvite ledningen til pin9 på mikrokontrolleren 3. Koble den tynne svarte ledningen til gnd på mikrokontrolleren 4. Koble den tynne røde ledningen til Vin på mikrokontrolleren 5. Koble opp batteriet (eller 11V spenning) til ESCen
Forventet Resultat:	Motoren går rolig fra stillestående til høy hastighet, deretter rolig ned igjen til den står stille. (programmet kjører i løkke)
Informasjon	
Laget av:	Luke Carambot 2011-01-14
Kommentar:	

TK108

Identifisering	
ID:	TK108
Beskrivelse:	Innkjøpte gyro test
Nivå:	Komponenttesting
Type test:	Funksjon
Kryssreferanser	
UseCase:	[UseCaseID til UseCasene som inngår i testen]
KravID:	KA602
Testdesign	
Kriterier:	<ul style="list-style-type: none"> • Mikrokontrolleren er koblet opp til aeroquadbrettet • Et testprogram har blitt utviklet og programmert på mikrokontrolleren
Testbeskrivelse:	<ol style="list-style-type: none"> 1. Utviklingsbrettet skal legges flat. 2. Koble mikrokontrolleren til USB. 3. Vent 10 sekunder. 4. Beveg aeroquadbrettet i hver sin akse.
Forventet Resultat:	Data programmet speiler bevegelsene i vinkelhastighet (grader/sek).
Informasjon	
Laget av:	Luke Carambot 2011-01-17
Kommentar:	

TK109

Identifisering	
ID:	TK109
Beskrivelse:	Innkjøpte akselerometer test
Nivå:	Komponenttesting
Type test:	Funksjon
Kryssreferanser	
UseCase:	[UseCaseID til UseCasene som inngår i testen]
KravID:	KA602
Testdesign	
Kriterier:	<ul style="list-style-type: none"> • Mikrokontrolleren er koblet opp til aeroquadbrettet • Et testprogram har blitt utviklet og programmert på mikrokontrolleren
Testbeskrivelse:	<ol style="list-style-type: none"> 1. Koble mikrokontrolleren til USB. 2. Vent 10 sekunder. 3. Med brettet liggende bestem for deg tre akser X, Y og Z 4. Notere utslaget på datamaskin når du: <ol style="list-style-type: none"> 1. Beveg akselerometeret i positiv X retning 2. Beveg akselerometeret i negativ X retning 3. Beveg akselerometeret i positiv Y retning 4. Beveg akselerometeret i negativ Y retning 5. Beveg akselerometeret i positiv Z retning 6. Beveg akselerometeret i negativ Z retning 5. Koble ut USB kabelen.
Forventet Resultat:	Datamaskinen skal vise tall ulikt fra null iht. bevegelsene i hver retning.
Informasjon	
Laget av:	Luke Carambot 2011-01-19
Kommentar:	

TK110

Identifisering	
ID:	TK110
Beskrivelse:	Innkjøpt mikrokontroller test
Nivå:	Komponenttesting
Type test:	Funksjon
Kryssreferanser	
UseCase:	[UseCaseID til UseCasene som inngår i testen]
KravID:	KA602
Testdesign	
Kriterier:	<ul style="list-style-type: none"> • Arduino IDE er installert på datamaskin • Mikrokontrolleren kobles til datamaskin ved USB.
Testbeskrivelse:	<ol style="list-style-type: none"> 1. Alle porter testes for input/output. 2. Et enkelt program kjøres for å teste mikrokontrollerens hovedfunksjoner.
Forventet Resultat:	Alle portene fungerer iht til datablad. Programmet utfører alle grunnleggende funksjoner, som å sette utganger/innganger høy, gjøre enkle addisjoner, multiplikasjoner osv.
Informasjon	
Laget av:	Vegard Torkelsen 2010-12-25
Kommentar:	

TK111

Identifisering	
ID:	TK111
Beskrivelse:	Fjernkontroll (sender-mottaker) test
Nivå:	Komponenttesting
Type test:	Funksjon
Kryssreferanser	
UseCase:	[UseCaseID til UseCasene som inngår i testen]
KravID:	KA702, (KA602)
Testdesign	
Kriterier:	<ul style="list-style-type: none"> • Godkjent TK106 og TK107 • Batteriet til senderen og batteripakken er laddet
Testbeskrivelse:	<ol style="list-style-type: none"> 1. Koble opp 4 motorer til 4 ESC. 2. Koble ESCene kanalene 1-4 på mottakeren. 3. Slå senderen på. 4. Still venstre stikken på null, dvs. helt nede. 5. Koble batteripakken til ESCene. 6. Vent til ESCene er ferdige å pipe. 7. Bevege stikkene i alle retninger.
Forventet Resultat:	Til hver stikke og retning er det minst en motor som kjører.
Informasjon	
Laget av:	Luke Carambot 2011-01-19
Kommentar:	

TK112

Identifisering	
ID:	TK112
Beskrivelse:	Teste lavt-batteri-alarmen
Nivå:	Komponenttesting
Type test:	Funksjon
Kryssreferanser	
UseCase:	[UseCaseID til UseCasene som inngår i testen]
KravID:	KA602, KA216
Testdesign	
Kriterier:	<ul style="list-style-type: none"> Batterialarmen kobles til en variabel spenningsforskyning.
Testbeskrivelse:	<ol style="list-style-type: none"> Sett spenning til høyere en 11.1V Sett spenning mellom 11.1 - 10.2V Sett spenning mellom 10.2 - 9.9V Sett spenning til lavere enn 9.9V
Forventet Resultat:	<p>Ved > 11.1V, alarmens LED er blå. Ved 11.1V - 10.2V, alarmens LED er blå og blinker. Ved 10.2V - 9.9V, alarmens LED lyser rødt Ved < 9.9V, alarmens LED blinker rødt og alarmen kan høres.</p>
Informasjon	
Laget av:	Vegard Torkelsen 2011-12-25
Kommentar:	

TS113

Identifisering	
ID:	TS113
Beskrivelse:	Teste reguleringsystemet
Nivå:	Systemtesting
Type test:	Funksjon
Kryssreferanser	
UseCase:	[UseCaseID til UseCasene som inngår i testen]
KravID:	KB502, KB503, KB505, KB506
Testdesign	
Kriterier:	<ul style="list-style-type: none"> • Reguleringsystemet er utviklet og implementert. • Fartøyet settes i testtrigg for sikkerhets skyld.
Testbeskrivelse:	<ol style="list-style-type: none"> 1. Slå kontrolleren på og stille venstre stikken helt nede. 2. Slå på fartøyet. 3. Vent til ESCene er ferdig å pipe. 4. Beveg venstres stikke oppover til 75% utslag, hold posisjon. 5. Beveg venstres stikke til venstre til 25% utslag, hold posisjon. 6. Beveg venstre stikke til høyre til 50% utslag, hold posisjon. 7. Beveg venstres stikke til venstre 50% utslag, hold posisjon. 8. Beveg høyre stikke oppover til 50% utslag, hold posisjon. 9. Beveg høyre stikke nedover til 50% utslag, hold posisjon. 10. Beveg høyre stikke til venstre til 50% utslag, hold posisjon. 11. Beveg høyre stikke til høyre til 50% utslag, hold posisjon
Forventet Resultat:	<ol style="list-style-type: none"> 4. Pådraget når det nivået som tilsvarer 75% påslag. Pådraget holder seg stabilt i form av underdempet eller bedre. 5. Rotasjonen i yaw skal holde den rotasjonshastigheten som tilsvarer 25% utslag, fartøyet skal rotere mot klokken. Rotasjonshastigheten skal holde seg stabilt i form av underdempet eller bedre. 6. Rotasjonen i yaw skal holde den rotasjonshastigheten som tilsvarer 50% utslag, fartøyet skal rotere med klokken. Rotasjonshastigheten skal holde seg stabilt i form av underdempet eller bedre. 7. Rotasjonen i yaw skal holde den rotasjonshastigheten som tilsvarer 50% utslag, fartøyet skal rotere mot klokken. Rotasjonshastigheten skal holde seg stabilt i form av underdempet eller bedre. 8. Rotasjonen i pitch skal holde den vinkelen som tilsvarer 50% utslag, nesen skal gå nedover. Vinkelen skal holde seg stabil i form av

	<p>underdempet eller bedre.</p> <p>9. Rotasjonen i pitch skal holde den vinkelen som tilsvarer 50% utslag, nesen skal gå oppover. Vinkelen skal holde seg stabil i form av underdempet eller bedre.</p> <p>10. Rotasjonen i roll skal holde den vinkelen som tilsvarer 50% utslag, rotasjonen langs lengderetning skal gå mot klokken. Vinkelen skal holde seg stabil i form av underdempet eller bedre.</p> <p>11. Rotasjonen i roll skal holde den vinkelen som tilsvarer 50% utslag, rotasjonen langs lengderetning skal gå med klokken. Vinkelen skal holde seg stabil i form av underdempet eller bedre.</p>
Informasjon	
Laget av:	Vegard Torkelsen 2011-01-03, oppdatert av Patrick Bjørum 2011-03-16
Kommentar:	

TS114

Identifisering	
ID:	TS114
Beskrivelse:	Teste reguleringsystemets av/på funksjon fra fjernkontrollen.
Nivå:	Systemtesting
Type test:	Funksjon
Kryssreferanser	
UseCase:	[UseCaseID til UseCasene som inngår i testen]
KravID:	KB504
Testdesign	
Kriterier:	<ul style="list-style-type: none"> • Reguleringsystemet er utviklet og implementert. • Mikrokontrolleren kobles til datamaskin.
Testbeskrivelse:	<ol style="list-style-type: none"> 1. Start med bryteren (kanal 5) står i stilling for aktivert reguleringsystem og venstre stikken helt nede. 2. Slå på senderen. 3. Slå på fartøyet. 4. Sett bryteren (kanal 5) i stilling for deaktivert reguleringsystem. 5. Sett bryteren (kanal 5) i stilling for aktivert reguleringsystem.
Forventet Resultat:	<ol style="list-style-type: none"> 3. Når programmet starter går det rett inn i løkka for aktivert reguleringsystem. 4. Når bryteren settes i stilling for deaktivert, går programmet inn i løkka for uregulert system. 5. Når brytere settes tilbake i stilling for aktivert reguleringsystem, går programmet inn i løkka for aktivert reguleringsystem.
Informasjon	
Laget av:	Vegard Torkelsen 2011-01-03
Kommentar:	

TS115

Identifisering	
ID:	TS115
Beskrivelse:	
Nivå:	
Type test:	
Kryssreferanser	
UseCase:	[UseCaseID til UseCasene som inngår i testen]
KravID:	
Testdesign	
Kriterier:	
Testbeskrivelse:	
Forventet Resultat:	
Informasjon	
Laget av:	[testdesignernavn] [åååå-mm-dd]
Kommentar:	Har blitt fjernet.

TS116

Identifisering	
ID:	TS116
Beskrivelse:	Teste visning og lagring av videostreamen fra fartøyet på en datamaskin
Nivå:	Systemtesting
Type test:	Funksjon
Kryssreferanser	
UseCase:	[UseCaseID til UseCasene som inngår i testen]
KravID:	KB902,KB903
Testdesign	
Kriterier:	<ul style="list-style-type: none"> • Koble opp video og mottaker til datamaskin
Testbeskrivelse:	<p>Skru på kameraet og mottaker.</p> <ol style="list-style-type: none"> 1. Skru på nødvendig programvare for å se videofeeden. 2. Skru på opptaksfunksjon.
Forventet Resultat:	<ol style="list-style-type: none"> 1. Ser videofeeden på datamaskinen. 2. Filen fra opptaksfunksjonen inneholder det samme som vi så på datamaskinen.
Informasjon	
Laget av:	Vegard Torkelsen 2011-01-03
Kommentar:	

TS117

Identifisering	
ID:	TS117
Beskrivelse:	Teste nødlanding ved tapt fjernkontrollsignal
Nivå:	Systemtesting
Type test:	Funksjon
Kryssreferanser	
UseCase:	[UseCaseID til UseCasene som inngår i testen]
KravID:	KC507
Testdesign	
Kriterier:	<ul style="list-style-type: none"> • Godkjent TI711 og TI713
Testbeskrivelse:	<ol style="list-style-type: none"> 1. Slå fartøyet og fjernkontrollen på 2. DEL 1 <ol style="list-style-type: none"> 1. Bruk venstre stikken å få fartøyet ca. 1m oppe i luften 2. Slå av fjernkontrollen 3. DEL 2 <ol style="list-style-type: none"> 1. Bruk venstre stikken å få fartøyet ca. 1m oppe i luften 2. Slå reguleringsystemet på 3. Slå av fjernkontrollen
Forventet Resultat:	Fartøyet lander av seg selv (både del 1 og 2)
Informasjon	
Laget av:	Luke Carambot 2011-01-03
Kommentar:	

TS118

Identifisering	
ID:	TS118
Beskrivelse:	Teste flytidsalarmen
Nivå:	Systemtesting
Type test:	Funksjon
Kryssreferanser	
UseCase:	[UseCaseID til UseCasene som inngår i testen]
KravID:	KC904
Testdesign	
Kriterier:	<ul style="list-style-type: none"> Ferdig beregnet gjennomsnittlig flytid.
Testbeskrivelse:	<ol style="list-style-type: none"> Stille venstre stikken på senderen helt nede og slå den på. Start stoppeklokken med det samme du slår på fartøyet. Mål tid til flytidsalarmen slår ut.
Forventet Resultat:	Flytidsalarmen slår ut ved korrekt tidspunkt.
Informasjon	
Laget av:	Vegard Torkelsen 2011-01-03
Kommentar:	Flytidsalarmen er et resultat av gjennomsnittlig flytid og gjennomsnittlig landingstid.

TS119

Identifisering	
ID:	TS119
Beskrivelse:	Teste at automatisk nødlanding skjer ca 20 sekunder etter lavt-batteri-alarmer.
Nivå:	Systemtesting
Type test:	Funksjon
Kryssreferanser	
UseCase:	[UseCaseID til UseCasene som inngår i testen]
KravID:	KC905
Testdesign	
Kriterier:	<ul style="list-style-type: none"> • Godkjent TI711 og TI712
Testbeskrivelse:	<ol style="list-style-type: none"> 1. Kjøre fartøyet til til lavt-batteri-alarmer går. 2. Ta tiden til automatisk nødlanding skjer. 3. Avslutt test etter 30 sekunder.
Forventet Resultat:	Automatisk nødlanding skjer 20 sekunder etter at alarmer har gått.
Informasjon	
Laget av:	Herå Rørvik 2011-01-03
Kommentar:	

TK120

Identifisering	
ID:	TK120
Beskrivelse:	Innkjøpt høydemåler test.
Nivå:	Komponenttesting
Type test:	Funksjon
Kryssreferanser	
UseCase:	[UseCaseID til UseCasene som inngår i testen]
KravID:	KD508
Testdesign	
Kriterier:	<ul style="list-style-type: none"> • Koblet høydemåler til oscilloskop og korrekt spenningsforsyning.
Testbeskrivelse:	1. Mål et sett med høyder, 1,5 og 10m
Forventet Resultat:	Signalene stemmer i henhold til datablad
Informasjon	
Laget av:	Vegard Torkelsen 2011-01-03
Kommentar:	

3.2 Målingstester

TK201

Identifisering	
ID:	TK201
Beskrivelse:	Innkjøpt batteri test
Nivå:	Komponenttesting
Type test:	Funksjon
Kryssreferanser	
UseCase:	[UseCaseID til UseCasene som inngår i testen]
KravID:	KA602
Testdesign	
Kriterier:	<ul style="list-style-type: none"> Batteriet er mottatt
Testbeskrivelse:	<ol style="list-style-type: none"> Lade opp batteriet. Mål spenningen på fullt oppladet batteri. Måle hver celle individuelt
Forventet Resultat:	Batteriet holder 12.6V fullt oppladet. Alle de tre cellene har samme spenning
Informasjon	
Laget av:	Vegard Torkelsen 2011-01-03
Kommentar:	

3.3 Vurderingstester

[Ingen tester]

3.4 Kontrolleringstester

TK401

Identifisering	
ID:	TK401
Beskrivelse:	Kontrollere at dokumentet [dokumentnavn] er skrevet.
Nivå:	Systemtesting
Type test:	Kontrollering
Kryssreferanser	
UseCase:	[UseCaseID til UseCasene som inngår i testen]
KravID:	KB402, KB403, KB501, KB505, KB506
Testdesign	
Kriterier:	<ul style="list-style-type: none"> • Har [dokumentnavn] i fra kravspesifikasjon, eller • Har det som skal dokumenteres i fra kravspesifikasjon.
Testbeskrivelse:	1. Kontroller at dokumentet er opprettet og dvs. eksister.
Forventet Resultat:	Dokumentet er opprettet og eksister.
Informasjon	
Laget av:	Luke Carambot 2010-12-21
Kommentar:	

TK404

Identifisering	
ID:	TK404
Beskrivelse:	Kontrollere standard radioutstyr.
Nivå:	Komponenttesting
Type test:	Kontrollering
Kryssreferanser	
UseCase:	[UseCaseID til UseCasene som inngår i testen]
KravID:	KA701
Testdesign	
Kriterier:	<ul style="list-style-type: none"> • Fjernkontrollen (sender) er fysisk tilgjengelig, eller • Databladet til enheten er tilstede.
Testbeskrivelse:	<ol style="list-style-type: none"> 1. Kontroller frekvensen enheten bruker ved å se på den, eller 2. Finn frem dokumentasjon til enheten og se etter frekvensen brukt. 3. Kontroller at frekvensen er lovlig i norge på § 21. Fjernstyring: http://www.lovddata.no/cgi-wift/ldles?doc=/sf/sf/sf-20050620-0715.html#21
Forventet Resultat:	Frekvensen er enten en av de i tabellen «til alle typer fjernstyring» eller en av de i tabellen «til fjernstyring av modellfly».
Informasjon	
Laget av:	Luke Carambot 2010-12-21
Kommentar:	

TK405

Identifisering	
ID:	TK405
Beskrivelse:	Kontrollere innkjøpte chassi.
Nivå:	Komponenttesting
Type test:	Kontrollering
Kryssreferanser	
UseCase:	[UseCaseID til UseCasene som inngår i testen]
KravID:	KA601
Testdesign	
Kriterier:	<ul style="list-style-type: none"> • Chassi har blitt bestilt og mottatt.
Testbeskrivelse:	1. Kontrollere at bekreftelse for handelen er tilgjengelig
Forventet Resultat:	Kvittering finnes.
Informasjon	
Laget av:	Herå Rørvik 2011-01-03
Kommentar:	

TK406

Identifisering	
ID:	TK406
Beskrivelse:	Kontrollere innkjøpte propellere
Nivå:	Komponent
Type test:	Kontrollering
Kryssreferanser	
UseCase:	[UseCaseID til UseCasene som inngår i testen]
KravID:	KA602
Testdesign	
Kriterier:	<ul style="list-style-type: none"> • Propeller har blitt bestilt og mottatt
Testbeskrivelse:	<ol style="list-style-type: none"> 1. Visuelt se at propellene er uskadet. 2. Bekrefte dimensjonene på pakken er riktig.
Forventet Resultat:	<ol style="list-style-type: none"> 1. Propellene er uskadde. 2. Dimensjonene på pakkene er 10 X 4.7 3. At halvparten er "Pushers"
Informasjon	
Laget av:	Vegard Torkelsen 2011-01-03
Kommentar:	

TK407

Identifisering	
ID:	TK407
Beskrivelse:	Kontrollerer innkjøpte koblingsbrett
Nivå:	Komponenttesting
Type test:	Kontrollering
Kryssreferanser	
UseCase:	[UseCaseID til UseCasene som inngår i testen]
KravID:	KA602
Testdesign	
Kriterier:	<ul style="list-style-type: none"> • Koblingsbrettet er bestilt og mottatt.
Testbeskrivelse:	1. Kontrollerer at koblingsbrettet er helt og uten skader, kontrollerer at koblingsbrettet er av samme type som vi bestilte.
Forventet Resultat:	1. Koblingsbrettet skal være helt og uten skader, brettet skal være det vi bestilte.
Informasjon	
Laget av:	Vegard Torkelsen 2011-01-03
Kommentar:	

TK408

Identifisering	
ID:	TK408
Beskrivelse:	Kontroller at fartøyet er konstruert av innkjøpte deler.
Nivå:	Komponenttesting
Type test:	Kontrollering
Kryssreferanser	
UseCase:	[UseCaseID til UseCasene som inngår i testen]
KravID:	KA603
Testdesign	
Kriterier:	<ul style="list-style-type: none"> • Vi har alle delene tilgjengelig.
Testbeskrivelse:	1. Kontrollere fartøyet med komponentdokumentasjon, at den nevner alle innkjøpte deler og i tilfellene en del er utviklet at den er dokumentert.
Forventet Resultat:	Komponentdokumentasjon stemmer med konstruksjonen.
Informasjon	
Laget av:	Luke Carambot 2010-12-22
Kommentar:	

TS419

Identifisering	
ID:	TS419
Beskrivelse:	Kontroller at vibrasjonsdempere brukes mellom sensorene og chassi.
Nivå:	Systemtesting
Type test:	Kontrollering
Kryssreferanser	
UseCase:	[UseCaseID til UseCasene som inngår i testen]
KravID:	KB605
Testdesign	
Kriterier:	<ul style="list-style-type: none"> • Har fartøyet fysisk tilgjengelig
Testbeskrivelse:	1. Kontrollere visuelt at vibrasjonsdempere er på plass mellom koblingsbrettet og chassi.
Forventet Resultat:	Vibrasjonsdempere er til stedet.
Informasjon	
Laget av:	Luke Carambot 2010-12-22
Kommentar:	

TS420

Identifisering	
ID:	TS420
Beskrivelse:	Kontroller at kameramodulen er montert på fartøyet.
Nivå:	Systemtesting
Type test:	Kontrollering
Kryssreferanser	
UseCase:	[UseCaseID til UseCasene som inngår i testen]
KravID:	KB902, KB903
Testdesign	
Kriterier:	<ul style="list-style-type: none"> • Har fartøyet fysisk tilgjengelighet
Testbeskrivelse:	1. Kontrollere visuelt at kamera modulen er montert på fartøyet.
Forventet Resultat:	Kameramodulen er på plass.
Informasjon	
Laget av:	Luke Carambot 2010-12-22
Kommentar:	

TK421

Identifisering	
ID:	TK421
Beskrivelse:	Kontroller at bena er farget slik at orienteringen til chassi er enkelt å se.
Nivå:	Komponenttesting
Type test:	Kontrollering
Kryssreferanser	
UseCase:	[UseCaseID til UseCasene som inngår i testen]
KravID:	KB604
Testdesign	
Kriterier:	<ul style="list-style-type: none"> • Chassi er fysisk tilgjengelig
Testbeskrivelse:	<ol style="list-style-type: none"> 1. Kontrollere visuelt at bena er farget og at orienteringen er enkel å se. 2. Notere hvor lett det er å se denne markering.
Forventet Resultat:	Bena er farget og det er enkelt å se det.
Informasjon	
Laget av:	Luke Carambot 2010-12-22
Kommentar:	

TI422

Identifisering	
ID:	TI422
Beskrivelse:	Kontrollerer konstruksjonen av fartøyet, at alle deler er på plass og ferdig montert.
Nivå:	Integrasjonstesting
Type test:	Kontrollering
Kryssreferanser	
UseCase:	[UseCaseID til UseCasene som inngår i testen]
KravID:	KA603
Testdesign	
Kriterier:	<ul style="list-style-type: none"> • Godkjent TK106, TK107, TK110
Testbeskrivelse:	1. Kontrollerer visuelt og fysisk at alle komponenter er på plass og er festet korrekt.
Forventet Resultat:	Alle komponenter er på plass og festet korrekt.
Informasjon	
Laget av:	Vegard Torkelsen 2011-01-03
Kommentar:	

TK423

Identifisering	
ID:	TK423
Beskrivelse:	Testing av kameramodul og bakkestasjon.
Nivå:	Komponenttest
Type test:	Kontrollering
Kryssreferanser	
UseCase:	[UseCaseID til UseCasene som inngår i testen]
KravID:	KB902
Testdesign	
Kriterier:	Ingen
Testbeskrivelse:	<ol style="list-style-type: none"> 1. Koble først kameraet til spenningsforsyning på 12V vha spenningsregulatoren. 2. Koble videoutgangen til et visningsmedium, TV, prosjektor e.l 3. Etter at Kameraet er bekreftet funksjonabelt, inkluder sender/mottaker, og repeter trinn 2 med dette. 4. Etter sender/mottaker er bekreftet funksjonabel må overgang til laptop testes, overgangen kobles til laptopen og korrekt programvare installeres. 5. Se at videooverføringen fungerer.
Forventet Resultat:	Kameramodulen skal klare å vise bilde via en sender/mottaker, og vi skal kunne motta denne. Modulen skal fungere med både 5 og 12 volt.
Informasjon	
Laget av:	Herå Rørvik 2011-01-21
Kommentar:	Kameramodulen har innebygd 12v til 5v regulator, men vi har dette også på mikrokontrolleren.

3.5 Godkjenningstester

TK501

Identifisering	
ID:	TK501
Beskrivelse:	Kontrollere godkjent budsjettet av Sinicon AS.
Nivå:	Komponenttesting
Type test:	Godkjenning
Kryssreferanser	
UseCase:	[UseCaseID til UseCasene som inngår i testen]
KravID:	KA401
Testdesign	
Kriterier:	<ul style="list-style-type: none"> Budsjettet har blitt levert til Simicon AS for godkjenning.
Testbeskrivelse:	1. Se at godkjenning av budsjettet er dokumentert.
Forventet Resultat:	Godkjenningsdokumentet finnes.
Informasjono	
Laget av:	Luke Carambot 2011-01-03
Kommentar:	

3.6 Kodeinspeksjon

TK601

Identifisering	
ID:	TK601
Beskrivelse:	Kontrollere transformeringen av fjernkontrollsignalene. Kodeblokken tar 4 innganger («3 rotasjoner og 1 retning» fra fjernkontrollen) og transformerer til pådrag på 4 motorer.
Nivå:	Komponenttesting
Type test:	Kodeinspeksjon
Kryssreferanser	
UseCase:	[UseCaseID til UseCasene som inngår i testen]
KravID:	KA702
Testdesign	
Kriterier:	<ul style="list-style-type: none"> • Koble opp mikrokontroller til datamaskin.
Testbeskrivelse:	<ol style="list-style-type: none"> 1. Kontroller koden for syntax feil ved kompilering. 2. Kontroller koden for oppførselsfeil.
Forventet Resultat:	Koden er uten syntax feil, og oppførselsfeil.
Informasjon	
Laget av:	Vegard Torkelsen 2011-01-03
Kommentar:	

TK602

Identifisering	
ID:	TK602
Beskrivelse:	Kontrollerer kode til reguleringssystemet.
Nivå:	Komponenttesting
Type test:	Kodeinspeksjon
Kryssreferanser	
UseCase:	[UseCaseID til UseCasene som inngår i testen]
KravID:	KB502
Testdesign	
Kriterier:	<ul style="list-style-type: none"> • Koden til reguleringssystemet er utviklet.
Testbeskrivelse:	<ol style="list-style-type: none"> 1. Kontrollere kode i henhold til kodemal 2. Kontrollere for syntax feil ved å compilere. 3. Kontrollere for oppførselsfeil ved automatiserte tester.
Forventet Resultat:	<ol style="list-style-type: none"> 1. Koden er strukturert i henhold til kodemal. 2. Koden inneholder ikke noen syntax feil. 3. Koden oppfører seg som planlagt.
Informasjon	
Laget av:	Vegard Torkelsen2011-01-03
Kommentar:	

TK603

Identifisering	
ID:	TK603
Beskrivelse:	Kontrollerer kode til flytidsalarmen.
Nivå:	Komponenttesting
Type test:	Kodeinspeksjon
Kryssreferanser	
UseCase:	[UseCaseID til UseCasene som inngår i testen]
KravID:	KC904
Testdesign	
Kriterier:	<ul style="list-style-type: none"> • Kode er utviklet
Testbeskrivelse:	<ol style="list-style-type: none"> 1. Kontrollere kode i henhold til kodemal 2. Kontrollere for syntax feil ved å compilere. 3. Kontrollere for oppførselsfeil ved automatiserte tester.
Forventet Resultat:	<ol style="list-style-type: none"> 1. Kode er strukturert i henhold til kodemal. 2. Kode inneholder ingen syntax feil. 3. Kode trigger funksjon for alarm etter tilstrekkelig flytid.
Informasjon	
Laget av:	Michael Odden 2011-01-03
Kommentar:	

TK604

Identifisering	
ID:	TK604
Beskrivelse:	Kontrollerer kode til nødlandingsfunksjonen.
Nivå:	Komponenttesting
Type test:	Kodeinspeksjon
Kryssreferanser	
UseCase:	[UseCaseID til UseCasene som inngår i testen]
KravID:	KC507, KC905
Testdesign	
Kriterier:	<ul style="list-style-type: none"> • Kode er utviklet
Testbeskrivelse:	<ol style="list-style-type: none"> 1. Kontrollere kode i henhold til kodemal 2. Kontrollere for syntax feil ved å compilere. 3. Kontrollere for oppførselsfeil ved å aktivere automatisk nødlandingsfunksjon før oppstart.
Forventet Resultat:	<ol style="list-style-type: none"> 1. Kode er strukturert i henhold til kodemal. 2. Kode inneholder ingen syntax feil. 3. Fartøy lander uten å skade seg selv eller omgivelser
Informasjon	
Laget av:	Michael Odden 2011-01-03
Kommentar:	

TK605

Identifisering	
ID:	TK605
Beskrivelse:	Kontrollerer kode til funksjonen som sjekker for et fjernkontrollsignal. Funksjonen brukes for å sjekke om mottakeren har mistet kontakt med senderen.
Nivå:	Komponenttesting
Type test:	Kodeinspeksjon
Kryssreferanser	
UseCase:	[UseCaseID til UseCasene som inngår i testen]
KravID:	KC507
Testdesign	
Kriterier:	<ul style="list-style-type: none"> • Kode for sjekk etter tap av signal er utviklet. • Kode er uten syntax-feil
Testbeskrivelse:	<ol style="list-style-type: none"> 1. Skruer av fjernkontroll 2. Venter 1s sekunder iht ønsket grense
Forventet Resultat:	<ol style="list-style-type: none"> 1. Oppdager at signalet er mistet og iverksetter handling
Informasjon	
Laget av:	Michael Odden 2011-01-03
Kommentar:	

TK606

Identifisering	
ID:	TK606
Beskrivelse:	Kontrollerer kode til batterinivåsjekkfunksjonen.
Nivå:	Komponenttesting
Type test:	Kodeinspeksjon
Kryssreferanser	
UseCase:	[UseCaseID til UseCasene som inngår i testen]
KravID:	(KA901), KC905
Testdesign	
Kriterier:	<ul style="list-style-type: none"> • Kode for batterinivå-sjekk er utviklet • Kode er uten syntax-feil
Testbeskrivelse:	<ol style="list-style-type: none"> 1. Starter fartøyet 2. Lar fartøyet stå på-slått inntil batterinivået bryter kritisk grense, 9.7V.
Forventet Resultat:	<ol style="list-style-type: none"> 1. Lavt batterinivå blir oppdaget og ønsket handling iverksettes.
Informasjon	
Laget av:	Michael Odden 2011-01-03
Kommentar:	

3.7 Grensesnittester

TI701

Identifisering	
ID:	TI701
Beskrivelse:	Teste integrasjon av fjernkontrollen og mottakeren.
Nivå:	Integrasjonstesting
Type test:	Grensesnittester
Kryssreferanser	
UseCase:	[UseCaseID til UseCasene som inngår i testen]
KravID:	KA702
Testdesign	
Kriterier:	<ul style="list-style-type: none"> Godkjent TK106 og TK107
Testbeskrivelse:	<ol style="list-style-type: none"> Koble motoren opp til ESCen. Koble ESCen til kanal 3 på mottakeren. Slå senderen på og stille venstre stikken helt ned. Koble opp batteriet til ESCen og vent til den er ferdig å pipe. Beveg venstre stikken opp over. Null stille venstre stikken. Koble ut batteriet.
Forventet Resultat:	Motoren var mulig å styre gjennom fjernkontrollen.
Informasjon	
Laget av:	Luke Carambot 2011-01-21
Kommentar:	Det kan hende at "throttle", som skal være venstre stikken, går på en annen kanal enn 3. Dette må justeres, enten ved å teste alle stillingene på begge stikkene for å få motoren til å kjøre eller koble ESCen på en annen kanal på mottakeren og prøve testen på nytt.

TI702

Identifisering	
ID:	TI702
Beskrivelse:	Teste integrasjon av mottakeren og mikrokontrolleren.
Nivå:	Integrasjonstesting
Type test:	Grensesnittester
Kryssreferanser	
UseCase:	[UseCaseID til UseCasene som inngår i testen]
KravID:	KA702
Testdesign	
Kriterier:	<ul style="list-style-type: none"> • Godkjent TI701 • Et program har blitt utviklet til testen.
Testbeskrivelse:	<ol style="list-style-type: none"> 1. Slå senderen på og stille venstre stikken helt nede. 2. Koble batteriet til fartøyet. 3. Beveg alle stikkene hver for seg.
Forventet Resultat:	<ol style="list-style-type: none"> 1. Hver av stikkene og bryterne gir identifiserbare signaler over hvert sitt respektive bevegelsesområde.
Informasjon	
Laget av:	Luke Carambot 2011-01-21
Kommentar:	

TI703

Identifisering	
ID:	TI703
Beskrivelse:	Teste integrasjon av mikrokontrolleren og motorene (via ESCene).
Nivå:	Integrasjonstesting
Type test:	Grensesnittester
Kryssreferanser	
UseCase:	[UseCaseID til UseCasene som inngår i testen]
KravID:	KA702
Testdesign	
Kriterier:	<ul style="list-style-type: none"> • Har en batteri eller spenningskilde (maks 11V) • Et program har blitt utviklet på mikrokontrolleren for å kjøre motoren gjennom ESCen. • Mikrokontrolleren er programmert med testprogram.
Testbeskrivelse:	<ol style="list-style-type: none"> 1. Koble motoren til ESCen (tre tykke ledninger) 2. Koble den til hvit ledning til pin9 på mikrokontrolleren 3. Koble den til svart ledning til gnd på mikrokontrolleren 4. Koble den til rød ledning til Vin på mikrokontrolleren 5. Koble opp batteriet (eller 11V spenning) til ESCen
Forventet Resultat:	Motoren går rolig fra stillestående til høy hastighet, deretter rolig ned igjen til den står stille. (programmet kjører i løkke)
Informasjon	
Laget av:	Luke Carambot 2011-01-14
Kommentar:	

TI704

Identifisering	
ID:	TI704
Beskrivelse:	Teste integrasjon av batteri og koblingsbrettet.
Nivå:	Integrasjonstesting
Type test:	Måling
Kryssreferanser	
UseCase:	[UseCaseID til UseCasene som inngår i testen]
KravID:	KA702
Testdesign	
Kriterier:	<ul style="list-style-type: none"> • Godkjent TK407 og TK201
Testbeskrivelse:	<ol style="list-style-type: none"> 1. Koble opp et ladet batteri til koblingsbrettet 2. Mål 5V kretsen 3. Mål 3.3V kretsen
Forventet Resultat:	Spenningene skal være 5V og 3.3V
Informasjon	
Laget av:	Luke Carambot 2011-01-03
Kommentar:	

TI705

Identifisering	
ID:	TI705
Beskrivelse:	Teste integrasjon av gyro og mikrokontrolleren.
Nivå:	Integrasjonstesting
Type test:	Grensesnittester
Kryssreferanser	
UseCase:	[UseCaseID til UseCasene som inngår i testen]
KravID:	KB502
Testdesign	
Kriterier:	<ul style="list-style-type: none"> • Mikrokontrolleren er koblet opp til aeroquadbrettet • Et testprogram har blitt utviklet og programmert på mikrokontrolleren
Testbeskrivelse:	<ol style="list-style-type: none"> 1. Utviklingsbrettet skal legges flatt. 2. Koble mikrokontrolleren til USB. 3. Vent 10 sekunder. 4. Beveg aeroquadbrettet i hver sin akse.
Forventet Resultat:	Data programmet speiler bevegelsene i vinkelhastighet (grader/sek).
Informasjon	
Laget av:	Luke Carambot 2011-01-21
Kommentar:	Dette er den samme som TK108

TI706

Identifisering	
ID:	TI706
Beskrivelse:	Teste integrasjon av akselerometer og mikrokontrolleren.
Nivå:	Integrasjonstesting
Type test:	Grensesnittester
Kryssreferanser	
UseCase:	[UseCaseID til UseCasene som inngår i testen]
KravID:	KB502
Testdesign	
Kriterier:	<ul style="list-style-type: none"> • Mikrokontrolleren er koblet opp til aeroquadbrettet • Et testprogram har blitt utviklet og programmert på mikrokontrolleren
Testbeskrivelse:	<ol style="list-style-type: none"> 1. Koble mikrokontrolleren til USB. 2. Vent 10 sekunder. 3. Med brettet liggende bestem for deg tre akser X, Y og Z 4. Notere utslaget på datamaskin når du: <ol style="list-style-type: none"> 1. Beveg akselerometeret i positiv X retning 2. Beveg akselerometeret i negativ X retning 3. Beveg akselerometeret i positiv Y retning 4. Beveg akselerometeret i negativ Y retning 5. Beveg akselerometeret i positiv Z retning 6. Beveg akselerometeret i negativ Z retning 5. Koble ut USB kableen.
Forventet Resultat:	Datamaskinen skal vise tall ulikt fra null iht. bevegelsene i hver retning.
Informasjon	
Laget av:	Luke Carambot 2011-01-21
Kommentar:	Dette er den samme som TK109

TI707

Identifisering	
ID:	TI707
Beskrivelse:	Teste integrasjon av mikrokontrolleren funksjon for sjekk av batterinivå.
Nivå:	Integrasjonstesting
Type test:	Grensesnittester
Kryssreferanser	
UseCase:	[UseCaseID til UseCasene som inngår i testen]
KravID:	KA901, KC905
Testdesign	
Kriterier:	<ul style="list-style-type: none"> • TK110 • Mikrokontrolleren er programmert med testprogrammet.
Testbeskrivelse:	<ol style="list-style-type: none"> 1. Mål spenningen over batteriet ved en voltmeter. 2. Koble batteriet til mikrokontrolleren gjennom GND og VIN pinnene 3. Koble en motstand (høye verdier er best) mellom VIN og A8.
Forventet Resultat:	Spenningen målt ved voltmeteren skal stemme med spenningen vist på dataskjermen.
Informasjon	
Laget av:	Luke Carambot 2011-01-21
Kommentar:	

TI708

Identifisering	
ID:	TI708
Beskrivelse:	Teste integrasjon av mikrokontrolleren og høydemåleren.
Nivå:	Integrasjonstesting
Type test:	Grensesnittester
Kryssreferanser	
UseCase:	[UseCaseID til UseCasene som inngår i testen]
KravID:	KD508
Testdesign	
Kriterier:	<ul style="list-style-type: none"> • Koble opp mikrokontroller og høydemåler
Testbeskrivelse:	<ol style="list-style-type: none"> 1. Beveg høydemåleren sakte oppover. 2. Beveg høydemåleren tilbake til originalposisjon.
Forventet Resultat:	1. Det er mulig for mikrokontrolleren å lese signalet fra høydemåleren.
Informasjon	
Laget av:	Vegard Torkelsen 2011-01-03
Kommentar:	

TI709

Identifisering	
ID:	TI709
Beskrivelse:	Teste integrasjon av mikrokontrollers transformering av fjernkontrollsignalet.
Nivå:	Integrasjonstesting
Type test:	Grensesnittester
Kryssreferanser	
UseCase:	[UseCaseID til UseCasene som inngår i testen]
KravID:	KA702
Testdesign	
Kriterier:	<ul style="list-style-type: none"> • Godkjent TI701, TI702 og TI703 • Mikrokontrolleren er programmert med testprogrammet. • Mottakeren er koblet til mikrokontrolleren (første 6 kanalene).
Testbeskrivelse:	<ol style="list-style-type: none"> 1. Kjør mikrokontroller program. 2. Slå på fjernkontrollen. 3. Beveg <ol style="list-style-type: none"> 1. Venstre stikken <ol style="list-style-type: none"> 1. Opp 2. Ned 3. Venstre 4. Høyre 2. Høyre stikken <ol style="list-style-type: none"> 1. Opp 2. Ned 3. Venstre 4. Høyre
Forventet Resultat:	<p>Ved</p> <ul style="list-style-type: none"> • 3.1.1 Signalene til motorene skal øke alle sammen • 3.1.2 Signalene til motorene skal senke alle sammen • 3.1.3 Motor 1 og 3 senker, motor 2 og 4 skal øke • 3.1.4 Motor 1 og 3 øker, motor 2 og 4 skal senke • 3.2.1 Motor 1 skal senke, motor 3 skal øke • 3.2.2 Motor 1 skal øke, motor 3 skal senke • 3.2.3 Motor 2 skal øke, motor 4 skal senke

	<ul style="list-style-type: none"> 3.2.4 Motor 2 skal senke, motor 4 skal øke
Informasjon	
Laget av:	Luke Carambot 2011-01-21
Kommentar:	

TI710

Identifisering	
ID:	TI710
Beskrivelse:	Teste integrasjon av mikrokontrollers reguleringsystem.
Nivå:	Integrasjonstesting
Type test:	Grensesnittester
Kryssreferanser	
UseCase:	[UseCaseID til UseCasene som inngår i testen]
KravID:	KB502
Testdesign	
Kriterier:	<ul style="list-style-type: none"> Godkjent TK602 Mikrokontrolleren er programmert med testprogrammet
Testbeskrivelse:	<ol style="list-style-type: none"> Slå fartøyet og fjernkontrollen på Still venstre stikken på ca. 50% Beveg høyre stikken helt til venstre <ol style="list-style-type: none"> Nullstill stikken og vent 5 sekunder Beveg høyre stikken helt til høyre <ol style="list-style-type: none"> Nullstill stikken og vent 5 sekunder Beveg høyre stikken helt opp <ol style="list-style-type: none"> Nullstill stikken og vent 5 sekunder Beveg høyre stikke helt ned <ol style="list-style-type: none"> Nullstill stikken og vent 5 sekunder
Forventet Resultat:	Signalene blir like for punktene 3, 4, 5 og 6
Informasjon	
Laget av:	Luke Carambot 2011-01-03
Kommentar:	

TI711

Identifisering	
ID:	TI711
Beskrivelse:	Teste integrasjon av mikrokontrollerens nødlandingsfunksjon og tapt fjernkontrollsignal.
Nivå:	Integrasjonstesting
Type test:	Grensesnittester
Kryssreferanser	
UseCase:	[UseCaseID til UseCasene som inngår i testen]
KravID:	KC507
Testdesign	
Kriterier:	<ul style="list-style-type: none"> • Godkjent TK604 og TK605 • Fartøyet er koblet til testbenken og propellene er av
Testbeskrivelse:	<ol style="list-style-type: none"> 1. Slå fartøyet og fjernkontrollen på 2. Beveg venstre stikke opp ca. 50% 3. Slå av fjernkontrollen
Forventet Resultat:	Nødlandsingsfunksjon kjøres (hører at motorene forandrer hastighet og stopper opp)
Informasjon	
Laget av:	Luke Carambot 2011-01-03
Kommentar:	

TI712

Identifisering	
ID:	TI712
Beskrivelse:	Teste integrasjon av mikrokontrollerens nødlandingsfunksjon og lavt-batteri-nivå.
Nivå:	Integrasjonstesting
Type test:	Grensesnittester
Kryssreferanser	
UseCase:	[UseCaseID til UseCasene som inngår i testen]
KravID:	KC905
Testdesign	
Kriterier:	<ul style="list-style-type: none"> • Godkjent TK604 og TK606 • Fartøyet er koblet til testbenken og propellene er av • Batteriet er erstattet med en variabel spenningskilde
Testbeskrivelse:	<ol style="list-style-type: none"> 1. Still spenningskilden på 12V 2. Slå fartøyet og fjernkontrollen på 3. Beveg venstre stikk opp ca. 50% 4. Skru spenningen ned til 9.9V 5. Juster spenningen sakte ned 6. Notere når nødlandingsfunksjon skjer (hører at motorene forandrer hastighet og stopper opp)
Forventet Resultat:	Nødlandingsfunksjonen skjer ved 9.7V
Informasjon	
Laget av:	Luke Carambot 2011-01-03
Kommentar:	

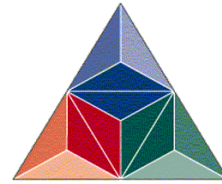
TI713

Identifisering	
ID:	TI713
Beskrivelse:	Teste integrasjon av mikrokontrollerens flytidsalarm.
Nivå:	Integrasjonstesting
Type test:	Grensesnittester
Kryssreferanser	
UseCase:	[UseCaseID til UseCasene som inngår i testen]
KravID:	KC904
Testdesign	
Kriterier:	<ul style="list-style-type: none"> • Godkjent TS105 og TK603 • Flytiden er kjent fra mikrokontroller-programvaren
Testbeskrivelse:	<ol style="list-style-type: none"> 1. Notere klokkeslett 2. Slå på fartøyet 3. Se etter flytidsalarmen når klokketiden er rundt flytiden
Forventet Resultat:	Alarmen går på når flytiden nåes.
Informasjon	
Laget av:	Luke Carambot 2011-01-03
Kommentar:	

3.8 Regresjonstester

TS801

Identifisering	
ID:	TS801
Beskrivelse:	Siste flytest for å bekrefte at fartøyet fungerer som tidligere testet etter oppgraderingene av mikrokontroller programvare.
Nivå:	Systemtesting
Type test:	Regresjonstesting
Kryssreferanser	
UseCase:	[UseCaseID til UseCasene som inngår i testen]
KravID:	KA702, KB502, KB503, KC507, KC904, KC905, KD508
Testdesign	
Kriterier:	<ul style="list-style-type: none"> • Skal være siste testen på slutten av prosjektet • Testene listet her utføres kun om de har blitt utført tidligere
Testbeskrivelse:	<ol style="list-style-type: none"> 1. Om mulige, kjøre TS105 på nytt 2. Om mulige, kjøre TS113 på nytt 3. Om mulige, kjøre TS117 på nytt 4. Om mulige, kjøre TS118 på nytt 5. Om mulige, kjøre TS119 på nytt 6. Om mulige, kjøre TS120 på nytt
Forventet Resultat:	Testene blir godkjent som før
Informasjon	
Laget av:	Luke Carambot 2011-01-03
Kommentar:	



HØGSKOLEN
i Buskerud

Oppgavetittel:

Hovedprosjekt - Testlogg - v4.0

Fag(nr./navn):

SFHO 3200 / Prosjektstyring med hovedprosjekt

Gruppemedlemmer, signatur:

Vegard Torkelsen

Michael Odden

Herå Rørvik

Trine Lindberg

Luke Carambot

Patrick Bjørum

Intern veileder: Dag Samuelsen

Intern sensor: Olaf Hallan Graven

Ekstern veileder og sensor: Ragnvald Otterlei

Dato: 2011-08-04

Innholdsfortegnelse

Dokumenthistorie.....	1
Innledning.....	1
1 Oversikt.....	2
2 Testene.....	5
TK112.....	6
TK110.....	7
TK111.....	7
TK111-2.....	8
TK201.....	8
Ingen.....	9
Ingen.....	10
Ingen.....	10
TK404.....	11
TK406.....	12
TK421.....	13
TK405.....	13
TK501.....	14
TK601.....	14
TK106 og TK107 og TI703.....	15
TI701.....	15
TI702.....	16
TK407.....	17
TK408.....	17
TK111.....	18
TK108 & TK109.....	19
TK423.....	20
TI422.....	20
TI704.....	21
TI422.....	21
TK423.....	22
TS103.....	23
TS104.....	24
TS105.....	25
TI705.....	25
TI706.....	26
TS419.....	26
TS114.....	27
TI710.....	28
TK602.....	29
TS113.....	30

Dokumenthistorie

Versjon	Beskrivelse	Når
1.0	Dokument opprettet Første offisielle versjon, gjort TK110, TK112, TK201, TK111, TK106, TK107, TK111-2, TK404, TK405, TK406, TK421 og TK501, TI703, TK407, TK408, TK108, TK109	2011-01-13
2.0	Stokket om noen innlegg slik at de forekommer etter dato og gjort til andre offisielle versjon.	2011-01-26
3.0	Utført TS116, TK423, TI422, TI704, TI701, TI702, TI709, TS101, TS102, TS103, TS104, TS105, oppdatert mal	2011-04-08
4.0	Utført TI705, TI706, TS419, TS114, TI710, TS113 og TK602	2011-05-26

Tabell 1: Dokumenthistorie

Innledning

Dette dokument er et arbeidsdokument. I dette dokumentet ligger alle testene som er utført. Dokumentasjonen har dannet grunnlag til senere testrapporter som skal ta for seg grupper av tester.

1 Oversikt

TestID	Beskrivelse	Test utført av	Status
TK106	Innkjøpte motorer test.	Vegard Torkelsen	Godkjent
TK107	Innkjøpte ESC test.	Vegard Torkelsen	Godkjent
TK108	Innkjøpte gyro test.	Vegard Torkelsen	Godkjent
TK109	Innkjøpte akselerometer test.	Vegard Torkelsen	Godkjent
TK110	Innkjøpte mikrokontroller test.	Luke Carambot, Michael Odden	Godkjent
TK111	Fjernkontroll (sender - mottaker) test.	Patrick Bjørum	Godkjent
TK111-2	Fjernkontroll(sender - mottaker) test	Patrick Bjørum	Godkjent
TK112	Teste lavt-batteri-alarmer.	Patrick Bjørum	Godkjent
TK120	Innkjøpt høydemåler test.		utgår
TK201	Innkjøpte batteri test.	Vegard Torkelsen	Godkjent
<i>ingen</i>	Testing av løftkraft	Vegard Torkelsen	Godkjent
<i>ingen</i>	Testing av turtall	Vegard Torkelsen	Ikke Godkjent
<i>ingen</i>	Testing av turtall (2)	Vegard Torkelsen	Godkjent
TK404	Kontrollere standard radioutstyr.	Luke Carambot, Herå Rørvik	Godkjent
TK405	Kontrollere kjøpt chassi.	Herå Rørvik	Godkjent
TK406	Kontrollere innkjøpte propeller.	Herå Rørvik	Godkjent
TK407	Kontrollere koblingsbrett.	Patrick Bjørum	Godkjent
TK408	Kontroller at fartøyet er konstruert av innkjøpte deler.	Patrick Bjørum	Godkjent
TK421	Kontroller at bena er farget slik at orienteringen til chassi er enkelt å se.	Herå Rørvik	Godkjent
TK423	Testing av kameramodul og bakkestasjon	Vegard Torkelsen	Godkjent
TK501	Kontrollere godkjent budsjettet av Simicon AS.	Luke Carambot	Godkjent

TestID	Beskrivelse	Test utført av	Status
TK601	Kontroller transformeringen av fjernkontrollsignalene. Kodeblokken tar 4 innganger («3 rotasjoner og 1 retning» fra fjernkontrollen) og transformerer til pådrag på de 4 motorene.	Vegard Torkelsen	Godkjent
TK602	Kontroller kode til reguleringssystemet.	Michael Odden	Godkjent
TK603	Kontroller kode til flytids-alarmen.		utgår
TK604	Kontroller kode til nødlandingsfunksjonen.		utgår
TK605	Kontroller kode til tapt-signal-alarmen.		utgår
TK606	Kontroller kode til lavt-batteri-alarmen.		utgår
TI422	Kontroller konstruksjonen av fartøyet, at alle deler er på plass og ferdig montert.	Herå Rørvik	Godkjent
TI701	Teste integrasjon av fjernkontrollen og mottakeren.	Patrick Bjørum	Godkjent
TI702	Teste integrasjon av mottakeren og mikrokontrolleren.	Michael Odden	Godkjent
TI703	Teste integrasjon av mikrokontrolleren og motorene (via ESCene).	Vegard Torkelsen	Godkjent
TI704	Teste integrasjon av batteriet og koblingsbrettet.	Herå Rørvik	Godkjent
TI705	Teste integrasjon av gyroen og mikrokontrolleren.	Luke Carambot	Godkjent
TI706	Teste integrasjon av akselerometeret og mikrokontrolleren.	Luke Carambot	Godkjent
TI707	Teste integrasjon mellom mikrokontrolleren og funksjonen for sjekk av batterinivå.		utgår
TI708	Teste integrasjon mellom mikrokontrolleren og høydemåleren.		utgår
TI709	Teste integrasjon av mikrokontrollerens transformering av fjernkontrollsignalet.	Patrick Bjørum	Godkjent
TI710	Teste integrasjon av mikrokontrollerens reguleringssystem.	Trine Lindberg	Godkjent

TestID	Beskrivelse	Test utført av	Status
TI711	Teste integrasjon av mikrokontrollerens nødlandingsfunksjon og tapt fjernkontrollsignal.		utgår
TI712	Teste integrasjon av mikrokontrollerens nødlandingsfunksjon og lavt-batteri-nivå.		utgår
TI713	Teste integrasjon av mikrokontrollerens flytids-alarm.		utgår
TS101	Thrust test.	Patrick Bjørum	Godkjent
TS102	Yaw test.	Patrick Bjørum	Godkjent
TS103	Pitch test.	Patrick Bjørum	Godkjent
TS104	Roll test.	Patrick Bjørum	Godkjent
TS105	Fly test av fartøyet.	Patrick Bjørum	Ikke Godkjent
TS113	Teste reguleringssystem.	Vegard Torkelsen	Godkjent
TS114	Teste reguleringssystemets av/på funksjon fra fjernkontrollen.	Luke Carambot	Godkjent
TS115	<i>tomt</i>		
TS116	Teste visning og lagring av videostreamen fra fartøyet på en datamaskin.	Vegard Torkelsen	Godkjent
TS117	Teste nødlandingsfunksjon ved tapt fjernkontrollsignal.		utgår
TS118	Teste flytids-alarmer.		utgår
TS119	Teste nødlandingsfunksjon, 20 sekunder etter lavt-batteri-alarmer.		utgår
TS419	Kontroller at vibrasjonsdempere brukes mellom sensorene og chassi.	Luke Carambot	Ikke godkjent
TS420	Kontroller at kameramodulen er montert på fartøyet.	Herå Rørvik	Godkjent
TS801	Regresjons flytest. Bekrefte at fartøyet fungerer som tidligere testet etter oppgradering av mikrokontroller programvare.		<i>Ingen behov</i>

Tabell 2: Alle tester fra testspesifikasjon_3.0

2 Testene

Malen under skal kopieres og fylles ut av testperson ved testing.

Identifisering	
TestID:	[TestID fra testspesifikasjon]
Testing	
Ressurser:	[Ressurser brukt under testing]
Forventet Resultat:	[Forventet testresultat fra testspesifikasjon]
Faktisk Resultat:	[Faktisk testresultat]
Konklusjon:	[Godkjent/Ikkegodkjent]
Informasjon	
Utført av:	[Testernavn] [åååå-mm-dd]
Kommentar:	[Tekst]

Tabell 3: Loggmal

TK112

Identifisering	
TestID:	TK112
Testing	
Ressurser:	Variabel spenningsforsyning
Forventet Resultat:	Ved > 11.1V, alarmens LED er blå. Ved 11.1 - 10.2V, alarmens LED er blå og blinker. Ved 10.2 - 9.9V, alarmens LED lyser rødt Ved < 9.9V, alarmens LED blinker rødt og alarmen kan høres.
Faktisk Resultat:	>11.1 LED lyser blått 11.1V - 10.2V LED blinker blått 10.2V - 9.9V LED Lyser rødt, alarm piper sakte. <9.9 LED blinker rødt , alarm piper raskt.
Konklusjon:	Godkjent
Informasjon	
Utført av:	Patrick Bjørum 2011-01-13
Kommentar:	Var uventet at alarmen pep mellom 10.2 - 9.9V, men det er bare bra!

TK110

Identifisering	
TestID:	TK110
Testing	
Ressurser:	Datamaskin USB kabel Arduino IDE v22 Koblingsbrettet, LEDer og ledninger
Forventet Resultat:	Alle portene fungerer iht til datablad. Programmet utfører alle grunnlegende funksjoner, som
Faktisk Resultat:	Portene 2-13 OK digital inn/ut Power pinnene OK Alle testede inn- og utganger fungerer som forventet
Konklusjon:	Godkjent
Informasjon	
Utført av:	Michael Odden og Luke Carambot 2011-01-13
Kommentar:	Portene 0-1 fungert ikke som digital ut ganger (mikrokontrollertest3)

TK111

Identifisering	
TestID:	TK111
Testing	
Ressurser:	Ingen
Forventet Resultat:	Til hver stikke og retning er det minst en motor som kjører.
Faktisk Resultat:	Mottakeren kunne ikke finne senderen (ID feil).
Konklusjon:	Ikke godkjent
Informasjon	
Utført av:	Patrick Bjørum 2011-01-17
Kommentar:	Skal se etter en annen sender (fjernkontroll).

TK111-2

Identifisering	
TestID:	TK111-2
Testing	
Ressurser:	Ny fjernkontroll
Forventet Resultat:	Til hver stikke og retning er det minst en motor som kjører.
Faktisk Resultat:	Til hver stikke og retning er det minst en motor som kjører. Signalene fra stikkene er lineære.
Konklusjon:	Godkjent
Informasjon	
Utført av:	Patrick Bjørum 2011-01-19
Kommentar:	Har lånt en fjernkontroll som fungerer.

TK201

Identifisering	
TestID:	TK201
Testing	
Ressurser:	Voltmeter
Forventet Resultat:	Alle de tre cellene i batteriet skal ha samme spenning, totalspenningen burde ligge rundt 12V, avhengig av hvor oppladet batteriet er.
Faktisk Resultat:	Alle de 3 cellene ligger på 3.85V, den totale spenningen blir da 11.55V
Konklusjon:	Godkjent
Informasjon	
Utført av:	Vegard Torkelsen 2010-01-17
Kommentar:	

Ingen

Identifisering	
TestID:	<i>ingen</i>
Testing	
Ressurser:	Vekt, lodd, motorer, chassi, mikrokontroller, ESC
Forventet Resultat:	Vi forventer at vi oppnår den informasjonen vi trenger for å se hva løftekraften på hver motor er, vi forventet at det var noe produksjonsavvik.
Faktisk Resultat:	Motorene hadde betydelig forskjellig løftekraft, og vi oppnådde den informasjonen vi behøvde for å få en god oversikt over variasjonene i denne kraft.
Konklusjon:	Godkjent
Informasjon	
Utført av:	Vegard Kleppe Torkelsen 2011-02-01
Kommentar:	Motorene er relativt like ved rundt 50% turtall, men variere noe ved lavt og høyt turtall. Se Vedlegg: http://git.projectdragonfly.no:3000/documents/42

Ingen

Identifisering	
TestID:	<i>ingen</i>
Testing	
Ressurser:	motorer, ESC, mikrokontroller, stroboskop
Forventet Resultat:	Vi forventer å motta den informasjonen vi trenger for å kunne bestemme sammenhengen mellom pulsbredde og RPM
Faktisk Resultat:	Vi fikk den ikke veldig godt resultat og valgte derfor og opprette test TK204
Konklusjon:	Godkjent
Informasjon	
Utført av:	Vegard Kleppe Torkelsen 2011-02-02
Kommentar:	se vedlegg : http://git.projectdragonfly.no:3000/documents/45

Ingen

Identifisering	
TestID:	<i>ingen</i>
Testing	
Ressurser:	motorer, ESC, mikrokontroller, mikrofon, laptop
Forventet Resultat:	Vi forventer å motta den informasjonen vi trenger for å kunne bestemme sammenhengen mellom pulsbredde og RPM
Faktisk Resultat:	Vi fikk nødvendig informasjon om turtallet.
Konklusjon:	Godkjent
Informasjon	
Utført av:	Vegard Kleppe Torkelsen 2011-02-02
Kommentar:	se vedlegg : http://git.projectdragonfly.no:3000/documents/44

TK404

Identifisering	
TestID:	TK404
Testing	
Ressurser:	Internett
Forventet Resultat:	Frekvensen er enten en av de i tabellen «til alle typer fjernstyring» eller en av de i tabellen «til fjernstyring av modellfly».
Faktisk Resultat:	Senderfrekvens 2,4 GHz finnes ikke i tabellene på lovdata.no. Fjernkontrollen selges i Norge på http://elefun.no/rc/radioutstyr.aspx .
Konklusjon:	Godkjent
Informasjon	
Utført av:	Luke Carambot og Herå Rørvik 2011-01-19
Kommentar:	Siden http://www.lovdata.no er ikke oppdatert, måtte vi bruke http://elefun.no/rc/radioutstyr.aspx . for å bekrefte at senderen er standard utstyr.

TK406

Identifisering	
TestID:	TK406
Testing	
Ressurser:	Ingen
Forventet Resultat:	<ol style="list-style-type: none">1. Propellene er uskadde.2. Dimensjonene på pakkene er 10 X 4.73. At halvparten er "Pushers"
Faktisk Resultat:	Mottatt uskadde propeller og riktig dimensjoner på pakken.
Konklusjon:	Godkjent
Informasjon	
Utført av:	Herå Rørvik 2010-01-19
Kommentar:	

TK421

Identifisering	
TestID:	TK421
Testing	
Ressurser:	Ingen
Forventet Resultat:	Bena er farget og det er enkelt å se det.
Faktisk Resultat:	Bena er farget og det er enkelt å se det.
Konklusjon:	
Informasjon	
Utført av:	Herå Rørvik 2011-01-19
Kommentar:	

TK405

Identifisering	
TestID:	TK405
Testing	
Ressurser:	Ingen
Forventet Resultat:	Bekreftelse for kjøpt chassi er tilgjengelig og det finnes kvittering av handelen.
Faktisk Resultat:	Kvitteringen finnes på http://git.projectdragonfly.no:3000/attachments/203/Kvittering_7.jpg og Chassi er mottatt.
Konklusjon:	Godkjent
Informasjon	
Utført av:	Herå Rørvik 2011-01-19
Kommentar:	

TK501

Identifisering	
TestID:	TK501
Testing	
Ressurser:	Ingen
Forventet Resultat:	Godkjenning dokumentet finnes.
Faktisk Resultat:	Bekreftelse finnes på http://git.projectdragonfly.no:3000/attachments/196/Godkjent_budsjett.pdf .
Konklusjon:	Godkjent
Informasjon	
Utført av:	Luke Carambot 2011-01-19
Kommentar:	

TK601

Identifisering	
TestID:	TK601
Testing	
Ressurser:	Ingen
Forventet Resultat:	Vi forventet at mikrokontrolleren oversatte kontrollersignalet over til motorligningene på korrekt måte, slik at de sendte ut riktig pulsbredde slik at riktige motorene fikk pådrag ved riktig stikkebevegelser
Faktisk Resultat:	Resultatet var som forventet
Konklusjon:	Godkjent
Informasjon	
Utført av:	Vegard Torkelsen 2011-01-27
Kommentar:	

TK106 og TK107 og TI703

Identifisering	
TestID:	TK106 og TK107 og TI703
Testing	
Ressurser:	Mikroprosessor, batteri, motorer og ESCer
Forventet Resultat:	Motorene og ESCene fungerte iht datablad.
Faktisk Resultat:	Motorene og ESCene fungerte fint, både ved lavt og høyt turtall.
Konklusjon:	Godkjent
Informasjon	
Utført av:	Vegard Kleppe Torkelsen 2011-01-19
Kommentar:	Disse komponentene ble testet sammen, siden det blir vanskelig og manuelt generere signalet som motoren bruker.

TI701

Identifisering	
TestID:	TI701
Testing	
Ressurser:	Fjernkontroll, radiomottaker, ESC, motor
Forventet Resultat:	Motoren var mulig å styre gjennom fjernkontrollen.
Faktisk Resultat:	Motoren var mulig å styre gjennom fjernkontrollen.
Konklusjon:	Godkjent
Informasjon	
Utført av:	Patrick Bjørum 2011-01-21
Kommentar:	(loggført av Michael Odden)

TI702

Identifisering	
TestID:	TI702
Testing	
Ressurser:	Fjernkontroll, radiomottaker, mikrokontroller
Forventet Resultat:	Hver av stikkene og bryterne gir identifiserbare signaler over hvert sitt respektive bevegelsesområde.
Faktisk Resultat:	Hver av stikkene og bryterne gav identifiserbare signaler over hvert sitt respektive bevegelsesområde. <ul style="list-style-type: none"> • Elevation: 1079 - 1926 • Aileron: 1096 - 1904 • Thrust: 1102 - 1915 • Yaw: 1095 - 1938
Konklusjon:	Godkjent
Informasjon	
Utført av:	Michael Odden 2011-01-21
Kommentar:	Kontrollen var konfigurert til å sende signaler lineært med stikkenes bevegelse, dette gjenspeilte seg også i de mottatte verdiene.

TK407

Identifisering	
TestID:	TK407
Testing	
Ressurser:	Ingen
Forventet Resultat:	Koblingsbrettet skal være helt og uten skader, brettet skal være det vi bestilte.
Faktisk Resultat:	Koblingsbrettet er helt og uten skader, brettet er det vi bestilte.
Konklusjon:	Godkjent
Informasjon	
Utført av:	Patrick Bjørum 2011-01-24
Kommentar:	

TK408

Identifisering	
TestID:	TK408
Testing	
Ressurser:	Ingen
Forventet Resultat:	Komponentdokumentasjon stemmer med konstruksjonen.
Faktisk Resultat:	Komponentdokumentasjon stemmer med konstruksjonen.
Konklusjon:	Godkjent
Informasjon	
Utført av:	Patrick Bjørum 2011-01-24
Kommentar:	Må bare bytte ut "huset"

TK111

Identifisering	
TestID:	TK111
Testing	
Ressurser:	Ingen
Forventet Resultat:	Til hver stikke og retning er det minst en motor som kjører.
Faktisk Resultat:	Til hver stikke og retning er det minst en motor som kjører.
Konklusjon:	Godkjent
Informasjon	
Utført av:	Patrick Bjørum 2011-01-24
Kommentar:	

TK108 & TK109

Identifisering	
TestID:	TK108 & TK109
Testing	
Ressurser:	Mikrokontroller, AeroQuad Configurator v2.6, AeroQuad software v2.1
Forventet Resultat:	Sensorene fungerer i henhold til datablad
Faktisk Resultat:	Sensorene fungerer i henhold til datablad
Konklusjon:	Godkjent
Informasjon	
Utført av:	Vegard Kleppe Torkelsen 2011-01-25
Kommentar:	<p>Brukte software som er laget av et forum kalt AeroQuad.</p> <p>Se vedlegg: http://git.projectdragonfly.no:3000/documents/43</p> <p>Vi hadde misforstått funksjonaliteten til gyroen og akselerometeret. Gyroen måler rotasjonshastighet, mens akselerometeret bruker tyngdens akselerasjons til å måle vinkel.</p>

TK423

Identifisering	
TestID:	TK423
Testing	
Ressurser:	Kamera, sender, mottaker, laptop, video-til-usb overgang, VLC video software og mikrokontroller.
Forventet Resultat:	Kameramodulen skal klare å vise bilde via en sender/mottaker, og vi skal kunne motta denne. Modulen burde fungere med både 5 og 12 volt.
Faktisk Resultat:	Som forventet
Konklusjon:	Godkjent
Informasjon	
Utført av:	Vegard Kleppe Torkelsen 2011.01.27
Kommentar:	Kameramodulen har innebygd 12v til 5v regulator, men vi har dette også på mikrokontrolleren.

TI422

Identifisering	
TestID:	TI422
Testing	
Ressurser:	Ingen
Forventet Resultat:	Alle komponenter er på plass og festet korrekt
Faktisk Resultat:	Alt er på plass, men noe skal festes bedre.
Konklusjon:	Godkjent
Informasjon	
Utført av:	Herå Rørvik [2011-02-08]
Kommentar:	

TI704

Identifisering	
TestID:	TI704
Testing	
Ressurser:	Multimeter, koblingsbrett, batteri
Forventet Resultat:	Spenningsene skal være 5V og 3.3V
Faktisk Resultat:	Spenningsene er 5V og 3.3V
Konklusjon:	Godkjent
Informasjon	
Utført av:	Herå Rørvik 2011-02-08
Kommentar:	

TI422

Identifisering	
TestID:	TI422
Testing	
Ressurser:	Ingen
Forventet Resultat:	Alle komponenter er på plass og festet korrekt
Faktisk Resultat:	Alt er på plass, men noe skal festes bedre.
Konklusjon:	Godkjent
Informasjon	
Utført av:	Herå Rørvik [2011-02-08]
Kommentar:	

TK423

Identifisering	
TestID:	TK423
Testing	
Ressurser:	Fartøy, testbenk, fjernkontroll
Forventet Resultat:	<ol style="list-style-type: none"> 1. Fartøyet letter og oppnår nøytral oppdrift. 2. Fartøyet vrir kun rundt høydeaksen med klokka (sett ovenfra) når venstre stikken beveges til høyre. 3. Fartøyet vrir kun rundt høydeaksen mot klokka (sett ovenfra) når venstre stikken beveges til venstre. 4. Fartøyet lander, testen over.
Faktisk Resultat:	Fartøyet roterer begge retningene vi ønsker med letthet.
Konklusjon:	Godkjent
Informasjon	
Utført av:	Patrick Bjørum 2011-02-23
Kommentar:	

TS103

Identifisering	
TestID:	TS103
Testing	
Ressurser:	Fartøy, testbenk, fjernkontroll
Forventet Resultat:	<ol style="list-style-type: none">1. Fartøyet letter og oppnår nøytral oppdrift.2. Fartøyet beveger fronten nedover og og drar framover.3. Fartøyet beveger fronten oppover og drar bakover.4. Fartøyet lander, testen over.
Faktisk Resultat:	Fartøyet drar begge retningene vi ønsker med letthet
Konklusjon:	Godkjent
Informasjon	
Utført av:	Patrick Bjørum 2011-02-23
Kommentar:	

TS104

Identifisering	
TestID:	TS104
Testing	
Ressurser:	Fartøy, testbenk, fjernkontroll
Forventet Resultat:	<ol style="list-style-type: none">1. Fartøyet letter og oppnår hover.2. Fartøyet beveger høyre siden nedover og drar til høyre.3. Fartøyet beveger venstre siden nedover og drar til venstre4. Fartøyet lander, testen over.
Faktisk Resultat:	Fartøyet drar begge retningene vi ønsker med letthet
Konklusjon:	Godkjent
Informasjon	
Utført av:	Patrick Bjørum 2011-02-23
Kommentar:	

TS105

Identifisering	
TestID:	TS105
Testing	
Ressurser:	Fartøy, testbenk, fjernkontroll
Forventet Resultat:	Fartøyet oppfører seg som forventet ved justeringer på hver stikke om gangen. Fartøyet skal bevege seg bestemt etter styringen ifra fjernkontrollen. Det er mulig å styre fartøyet i luften i 5 minutter.
Faktisk Resultat:	Fartøyet justerer seg i de retningene man velger med justeringer, men den uten input drar den vilt rundt i alle retninger og det er i praksis umulig å holde den rolig i luften
Konklusjon:	Ikke godkjent
Informasjon	
Utført av:	Patrick Bjørum 2011-02-23
Kommentar:	Dette kan komme av uliniærhet på motorkurvene samt for høyt tyngdepunkt. Vi skal utarbeide disse manglene og teste på nytt senere.

TI705

Identifisering	
TestID:	TI705
Testing	
Ressurser:	Fartøy og Arduino IDE
Forventet Resultat:	Data programmet speiler bevegelsene i vinkelhastighet (grader/sek).
Faktisk Resultat:	Vinkelhastighetene kommer frem i hver sine akse.
Konklusjon:	godkjent
Informasjon	
Utført av:	Luke Carambot 2011-05-20
Kommentar:	

TI706

Identifisering	
TestID:	TI706
Testing	
Ressurser:	Fartøy og Arduino IDE
Forventet Resultat:	Datamaskinen skal vise tall ulikt fra null iht. bevegelsene i hver retning.
Faktisk Resultat:	Akselerasjon ble målt i hver sine akse og retning.
Konklusjon:	Godkjent
Informasjon	
Utført av:	Luke Carambot 2011-05-20
Kommentar:	

TS419

Identifisering	
TestID:	TS419
Testing	
Ressurser:	Ingen
Forventet Resultat:	Vibrasjonsdempere er til stede.
Faktisk Resultat:	Ingen vibrasjonsdempere er til stede.
Konklusjon:	Ikke godkjent
Informasjon	
Utført av:	Luke Carambot 2011-05-20
Kommentar:	Har valgt å ikke ta med vibrasjonsdempere

TS114

Identifisering	
TestID:	TS114
Testing	
Ressurser:	Fartøyet, fjernkontroll
Forventet Resultat:	<p>1. Når programmet starter går det rett inn i løkka for aktivert reguleringssystem.</p> <p>4. Når bryteren settes i stilling for deaktivert, går programmet inn i løkka for uregulert system.</p> <p>5. Når brytere settes tilbake i stilling for aktivert reguleringssystem, går programmet inn i løkka for aktivert reguleringssystem.</p>
Faktisk Resultat:	<p>(et grønt LEDlys er brukt for å avgjøre hvilke løkke programmet er i)</p> <p>1. Når programmet starter er LEDlyset grønt.</p> <p>4. Når bryteren settes i stilling for deaktivert, går lyset av.</p> <p>5. Når brytere settes tilbake i stilling for aktivert reguleringssystem, går lyset på igjen.</p>
Konklusjon:	Godkjent
Informasjon	
Utført av:	Luke Carambot 2011-05-20
Kommentar:	Egentlig er dette en bytte mellom reguleringsmodusene, siden det er umulig å fly fartøyet uten noen form for regulering.

TI710

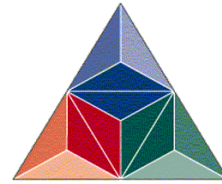
Identifisering	
TestID:	TI710
Testing	
Ressurser:	Fartøyet, fjernkontroll, fysisk testbenk
Forventet Resultat:	Signalene blir like, dvs fartøyet går tilbake til nullpunktet når man slipper stikka uansett hvilken retning den har vært i, og med samme respons og tid.
Faktisk Resultat:	Får omtrent samme respons og resultat når vi slipper stikka uansett hvilken retning den har stått i.
Konklusjon:	Godkjent
Informasjon	
Utført av:	Trine Lindberg 2011-05-24
Kommentar:	Testen er gjort på litt annen måte enn spesifisert i spesifikasjonen. Siden stabil modus ikke er ferdig tunet er det ikke helt stabilt ennå, men godt nok til å godkjenne denne testen.

TK602

Identifisering	
TestID:	TK602
Testing	
Ressurser:	Fartøyet, fjernkontroll, fysisk testbenk
Forventet Resultat:	<ol style="list-style-type: none"> 1. Kode er strukturert i henhold til kodemal 2. Kode uten syntax-feil og kompilierbar 3. Kode verifisert mht oppførsel
Faktisk Resultat:	<ol style="list-style-type: none"> 1. Koden er strukturert i henhold til kodemal. 2. Koden inneholder ikke noen syntax feil. 3. Programmet oppfører seg som planlagt.
Konklusjon:	Godkjent
Informasjon	
Utført av:	Michael Odden 2011-05-25
Kommentar:	<p>Test utført på revisjon: 18df4c7c</p> <p>Funksjonell kode har ikke vært egnet for automatiserte tester. Tester er derfor gjort ved kjøring mot testbenk og vha radio-kontroll.</p> <p>Funksjoner bekreftet:</p> <ul style="list-style-type: none"> * Inn/ut av regulert modus - bekreftet via LED og testbenk * Inn/ut av armert modus - bekreftet via LED og testbenk * Radio leses - bekreftet via testbenk * Motorsignaler sendes ut til ESCer - bekreftet via observasjoner samt testbenk * Seriell inn/ut fungerer - testbenken fungerer ved å sende og lese parametere * Reguleringsystemet kjører - testet vha observasjoner <p>Merk: Det er ikke verifisert at reguleringsystemet faktisk gjør fartøyet flygbart - det anses ikke som relevant for denne testen.</p>

TS113

Identifisering	
TestID:	TS113
Testing	
Ressurser:	Fartøyet, fjernkontroll, fysisk testbenk
Forventet Resultat:	<ol style="list-style-type: none"> 1. Pådraget når det nivået som tilsvarer 75% påslag. Pådraget holder seg stabilt i form av underdempet eller bedre. 2. Rotasjonen i yaw skal holde den rotasjonshastigheten som tilsvarer 25% utslag, fartøyet skal rotere mot klokken. Rotasjonshastigheten skal holde seg stabilt i form av underdempet eller bedre. 3. Rotasjonen i yaw skal holde den rotasjonshastigheten som tilsvarer 50% utslag, fartøyet skal rotere med klokken. Rotasjonshastigheten skal holde seg stabilt i form av underdempet eller bedre. 4. Rotasjonen i yaw skal holde den rotasjonshastigheten som tilsvarer 50% utslag, fartøyet skal rotere mot klokken. Rotasjonshastigheten skal holde seg stabilt i form av underdempet eller bedre. 5. Rotasjonen i pitch skal holde den vinkelen som tilsvarer 50% utslag, nesen skal gå nedover. Vinkelen skal holde seg stabil i form av underdempet eller bedre. 6. Rotasjonen i pitch skal holde den vinkelen som tilsvarer 50% utslag, nesen skal gå oppover. Vinkelen skal holde seg stabil i form av underdempet eller bedre. 7. Rotasjonen i roll skal holde den vinkelen som tilsvarer 50% utslag, rotasjonen langs lengderetning skal gå mot klokken. Vinkelen skal holde seg stabil i form av underdempet eller bedre. 8. Rotasjonen i roll skal holde den vinkelen som tilsvarer 50% utslag, rotasjonen langs lengderetning skal gå med klokken. Vinkelen skal holde seg stabil i form av underdempet eller bedre.
Faktisk Resultat:	Alle bevegelsene var etter forventet resultat
Konklusjon:	Godkjent
Informasjon	
Utført av:	Vegard Torkelsen 2011-05-26
Kommentar:	Denne testen ble gjort før PIDene i stabilregulert modus var ferdig tunet. Dette pga tidsmangel. Selv om reguleringssystemet ikke var så stabil som vi ønsker, kunne vi klart se at stikkebevegelsene resulterte i en endring i vinkel i alle akser. Rotasjonshastighetene i yaw var også som forventet.



HØGSKOLEN
i Buskerud

Oppgavetittel:

Hovedprosjekt - Testrapport iterasjon 2 - v2.0

Fag(nr./navn):

SFHO 3200 / Prosjektstyring med hovedprosjekt

Gruppemedlemmer, signatur:

Vegard Torkelsen

Michael Odden

Herå Rørvik

Trine Lindberg

Luke Carambot

Patrick Bjørum

Intern veileder: Dag Samuelsen

Intern sensor: Olaf Hallan Graven

Ekstern veileder og sensor: Ragnvald Otterlei

Dato: 2011-05-18

Innholdsfortegnelse

Dokumenthistorie.....	1
Innledning.....	1
1 Oversikt over testene i iterasjon 2.....	2
2 Rapporten.....	3
2.1 Forberedelse til testene.....	3
2.2 De enkle.....	3
2.3 Utfordringene.....	3
Konklusjoner.....	4
Referanser.....	5

Dokumenthistorie

Versjon	Beskrivelse	Når
1.0	Første offisielle versjon	2011-01-31
2.0	Forandret tittelen fra testrapport 2	2011-05-18

Tabell 1: Dokumenthistorie

Innledning

Dette rapport tar for seg de testene som ble gjennomført under iterasjon 2. Mest parten av testene var komponenttester. Alle de innkjøpte dele skulle testes for å sikre at alt er i orden før vi begynner å bygge.

1 Oversikt over testene i iterasjon 2

TestID	Beskrivelse
TK106	Innkjøpte motorer test.
TK107	Innkjøpte ESC test.
TK108	Innkjøpte gyro test.
TK109	Innkjøpte akselerometer test.
TK110	Innkjøpte mikrokontroller test.
TK111	Fjernkontroll (sender - mottaker) test.
TK112	Teste lavt-batteri-alarmer.
TK201	Innkjøpte batteri test.
TK404	Kontrollere standard radioutstyr.
TK405	Kontrollere kjøpt chassis.
TK406	Kontrollere innkjøpte propeller.
TK407	Kontrollere koblingsbrett.
TK408	Kontroller at fartøyet er konstruert av innkjøpte deler.
TK421	Kontroller at bena er farget slik at orienteringen til chassis er enkelt å se.
TK501	Kontrollere godkjent budsjettet av Simicon AS.
TI701	Teste integrasjon av fjernkontrollen og mottakeren.
TI702	Teste integrasjon av mottakeren og mikrokontrolleren.
TI703	Teste integrasjon av mikrokontrolleren og motorene (via ESCene).

Tabell 2: Testene til iterasjon 2 [2]

2 Rapporten

2.1 Forberedelse til testene

Noen av testene skulle kjøres gjennom mikrokontrolleren. Til disse testene måtte testprogrammet programmeres. Testene til å teste mikrokontrolleren, ESCene, motorene og de to sensorene, gyroen og akselerometeret krevde et testprogram. Disse programmene ble laget til testene.

2.2 De enkle

Testene som var enkelt å fullføre uten problemer var

- Lavt-batteri-alarmen test
- Batteri test
- Chassis test
- Propell test
- Koblingsbrett
- At fartøyet er konstruert av innkjøpte deler
- Godkjent budsjett
- Integrasjon av mottakeren og mikrokontrolleren
- Integrasjon av mikrokontrolleren og motorene

2.3 Utfordringene

Komponent testene av fjernkontrollen og mottakeren ble vanskelig å teste hver for seg og ble til den samme test som integrasjon av fjernkontrollen og mottakeren. I første omgang ble de ikke godkjent. Mottakeren fant ikke senderen (fjernkontrollen) og de ble en par dager til en sender ble lånt som ble oppdaget av mottakeren. Testene ble godkjent til slutt.

Den neste utfordringen var med de to sensorene. Det var usikkerhet rundt det data som testprogrammet ga tilbake. For å komme seg videre ble noen ferdig utviklet program brukt på mikrokontrolleren slik at komponenttestene kunne fullføres. Testene fikk godkjent.

Det siste test som kan kommenteres er testen på standard utstyret til fjernstyring. Det viste seg at lovdata på Internettet er utdatert. Frekvensen på senderen fantes ikke i tabellene på de lovlige fjernstyringsfrekvensene. Testen ble godkjent ved å dokumentere at senderen faktisk selges i Norge. Den er antatt til å være standard og lovlig utstyr.

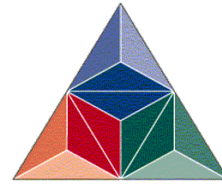
Konklusjoner

Curabitur sollicitudin pulvinar euismod. Sed molestie est eget dolor imperdiet egestas. Sed sit amet lacus at massa bibendum lobortis. Nullam leo nulla, iaculis sit amet condimentum eget, sollicitudin in quam. Fusce eu nibh augue, at dignissim lectus. Nunc imperdiet lacinia nisi quis pulvinar. Vestibulum vitae lectus felis, ac cursus ipsum. Suspendisse ac eros tellus. Pellentesque quis neque diam. Nam ut porttitor nunc.

Med andre ord, alle testene er godkjent [1].

Referanser

- [1] Dragonfly (2011) Testlogg_2.0: Kongsberg
- [2] Dragonfly (2011) Testspesifikasjon_1.0: Kongsberg



HØGSKOLEN
i Buskerud

Oppgavetittel: Hovedprosjekt - Testrapport iterasjon 3 - v2.0		
Fag(nr./navn): SFHO 3200 / Prosjektstyring med hovedprosjekt		
Gruppemedlemmer, signatur:		
_____ Vegard Torkelsen	_____ Michael Odden	_____ Herå Rørvik
_____ Trine Lindberg	_____ Luke Carambot	_____ Patrick Bjørum
Intern veileder: Dag Samuelsen Intern sensor: Olaf Hallan Graven Ekstern veileder og sensor: Ragnvald Otterlei		
Dato: 2011-05-18		

Innholdsfortegnelse

Dokumenthistorie.....	1
Innledning.....	1
1 Oversikt over testene i iterasjon 3.....	2
2 Rapporten.....	3
2.1 Forberedelse til testene.....	3
2.2 De enkle.....	3
2.3 Løftekraft og turtallstester.....	3
2.4 Videomodul.....	6
2.5 De siste 5 testene.....	7
2.5.1 Bevegelses testene.....	7
2.5.2 Flytesten.....	7
Konklusjoner.....	7
Referanser.....	8

Dokumenthistorie

Versjon	Beskrivelse	Når
1.0	Første offisielle versjon	2011-03-02
2.0	Forandret tittelen fra testrapport 3	2011-05-18

Tabell 1: Dokumenthistorie

Innledning

Denne rapporten tar for seg de testene som ble gjennomført under iterasjon 3.

1 Oversikt over testene i iterasjon 3

TestID	Beskrivelse
TK202	Testing av løftekraft.
TK203	Testing av turtall.
TK204	Testing av turtall(2).
TK423	Kontroll av videomodul.
TK601	Kontroller transformeringen av fjernkontrollsignalene.
TI422	Kontroller konstruksjonen av fartøyet, at alle deler er på plass og montert.
TI704	Teste integrasjonen av batteriet og koblingsbrettet.
TI709	Teste integrasjon av mikrokontrollerens transformering av fjernkontrollsignal
TS101	Thrust test
TS102	Yaw test
TS103	Pitch test
TS104	Roll test
TS105	Fly test av fartøyet

Tabell 2: Testene til iterasjon 3 [2]

2 Rapporten

2.1 Forberedelse til testene

For å få fullført disse testene måtte vi låne litt forskjellig utstyr:

Til turtall testene brukte vi først et stroboscop, så lånte vi en høy-presisjons mikrofon koblet til et digitalt oscilloskop.

For å teste løftkraft brukte vi en vanlig kjøkkenvekt og et tungt objekt liggende oppå som motorene prøvde å løfte.

Til thrust, yaw, pitch, roll og flytesten forelenget vi armene og satt på et snøre slik fire personer kunne holde i hver sin ende sånn at fartøyet var trygt fastspent mens en prøvde å manøvrere det.

Til kameratesten brukte vi en prosjektor med vanlig video-inn inngang for å se bildet.

2.2 De enkle

Testene som var enkelt å fullføre uten problemer var:

- Kontroller konstruksjonen av fartøyet, at alle deler er på plass og montert
- Teste integrasjonen av batteriet og koblingsbrettet
- Teste integrasjon av mikrokontrollerens transformering av fjernkontrollsignal

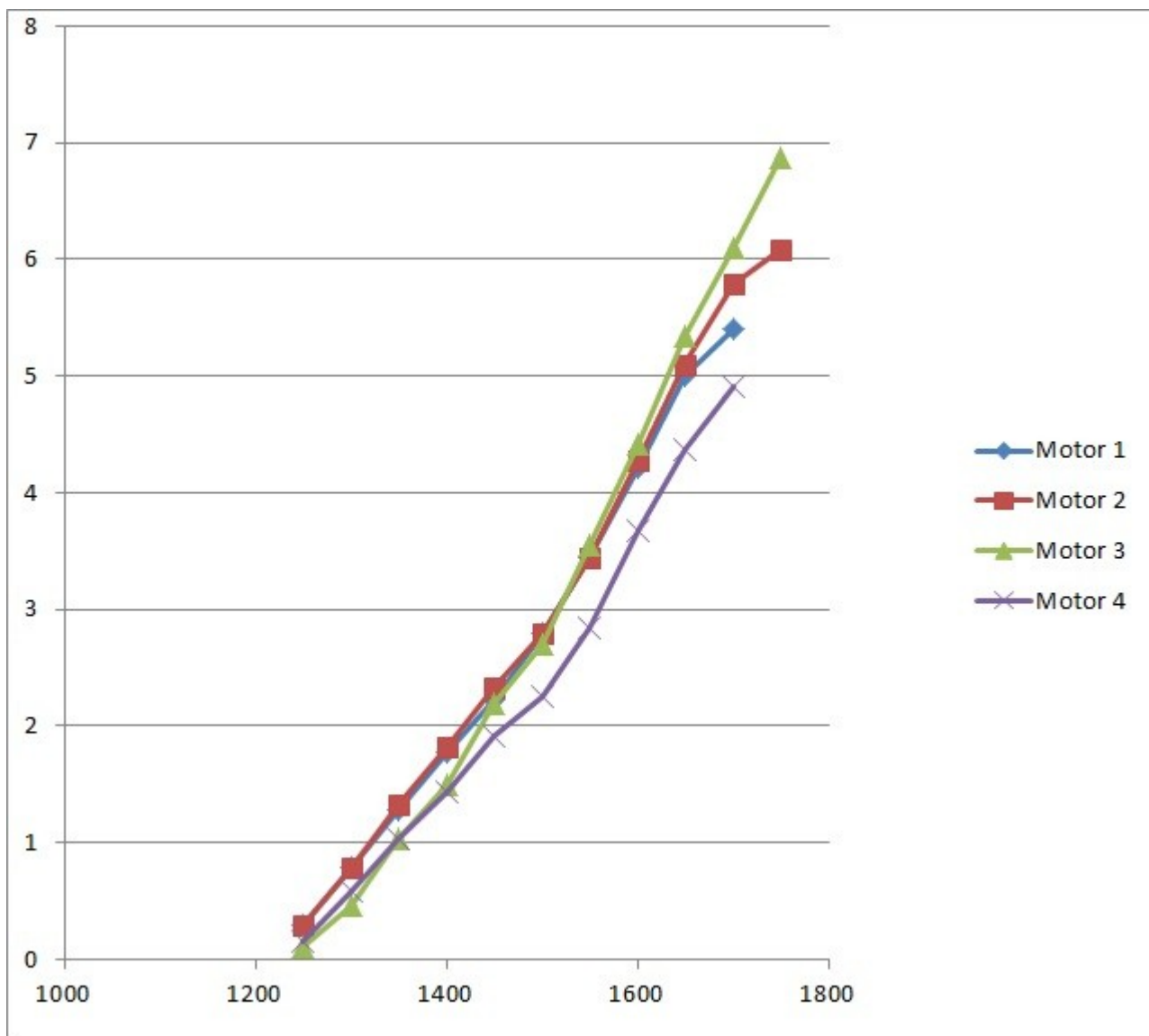
2.3 Løftkraft og turtallstester

Under utviklingen av det uregulerte systemet, la vi merke til at motorene ikke oppførte seg likt. Vi ønsket derfor å utføre en test for å undersøke hvordan de enkelte motorene oppførte seg, slik at vi kan identifisere og rette opp eventuelle produksjonsavvik.

Den første testen som ble utført var en test som undersøkte løftkraften til hver motor.

Her ble fartøyet hengt opp slik at det kunne rotere fritt rundt en akse. Deretter festet vi et lodd liggende oppå en kjøkkenvekt i en snor til den motorarmen som skulle testes. Pulsbredden til ESCen ble deretter økt med et fast nivå, fra minimum til maksimum.

Loddet lå på en kjøkkenvekt slik at vi kunne se hvor mange gram motoren løftet. Dette ble deretter satt opp i en tabell[3] og regnet om til Newton. Se grafen på neste side.

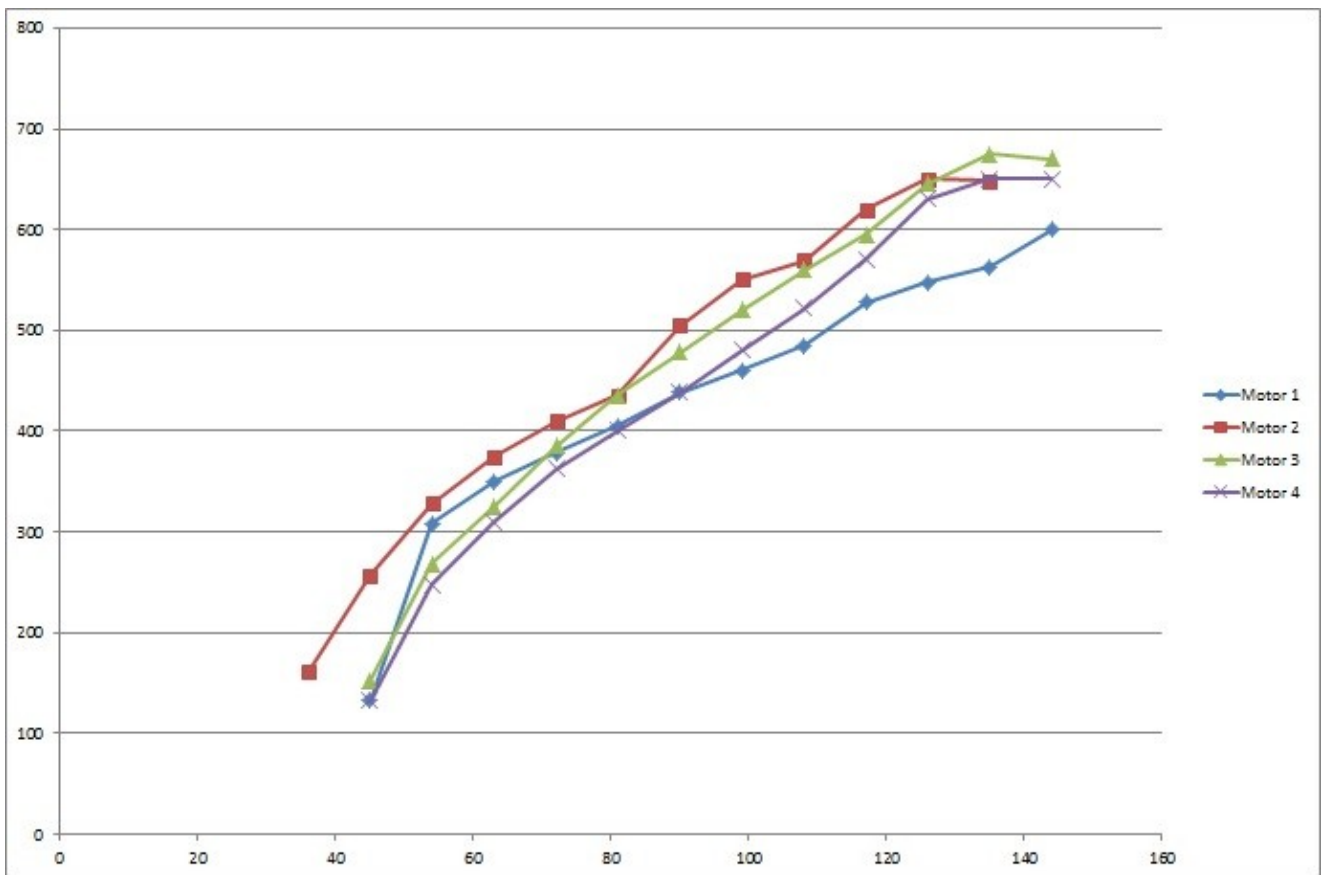


Figur 1: Løftekraft per motor (pulsbredde [us] vs kraft [N])

Denne testen viste å by seg på visse problemer, turbulens fra motoren hadde påvirket vekten og vi hadde vanskeligheter med å gjøre nøyaktige målinger, som man kan se ut i fra grafen førte dette til en spredning i dataene.

Vi valgte derfor å utføre en turtallstest, siden løftekraften er et produkt av turtallet.

Turtallstesten ble gjort ved hjelp av et stroboskop. Stroboskop sender ut blinkende lys som blinker ved en gitt variabel frekvens. Dette gjør at vi kan justere stroboskopets frekvens til det når samme frekvens som motorens rotasjon og vi kan da lese av turtallet. Resultatene ble igjen oppført i en tabell [4]. Se figur på neste side.

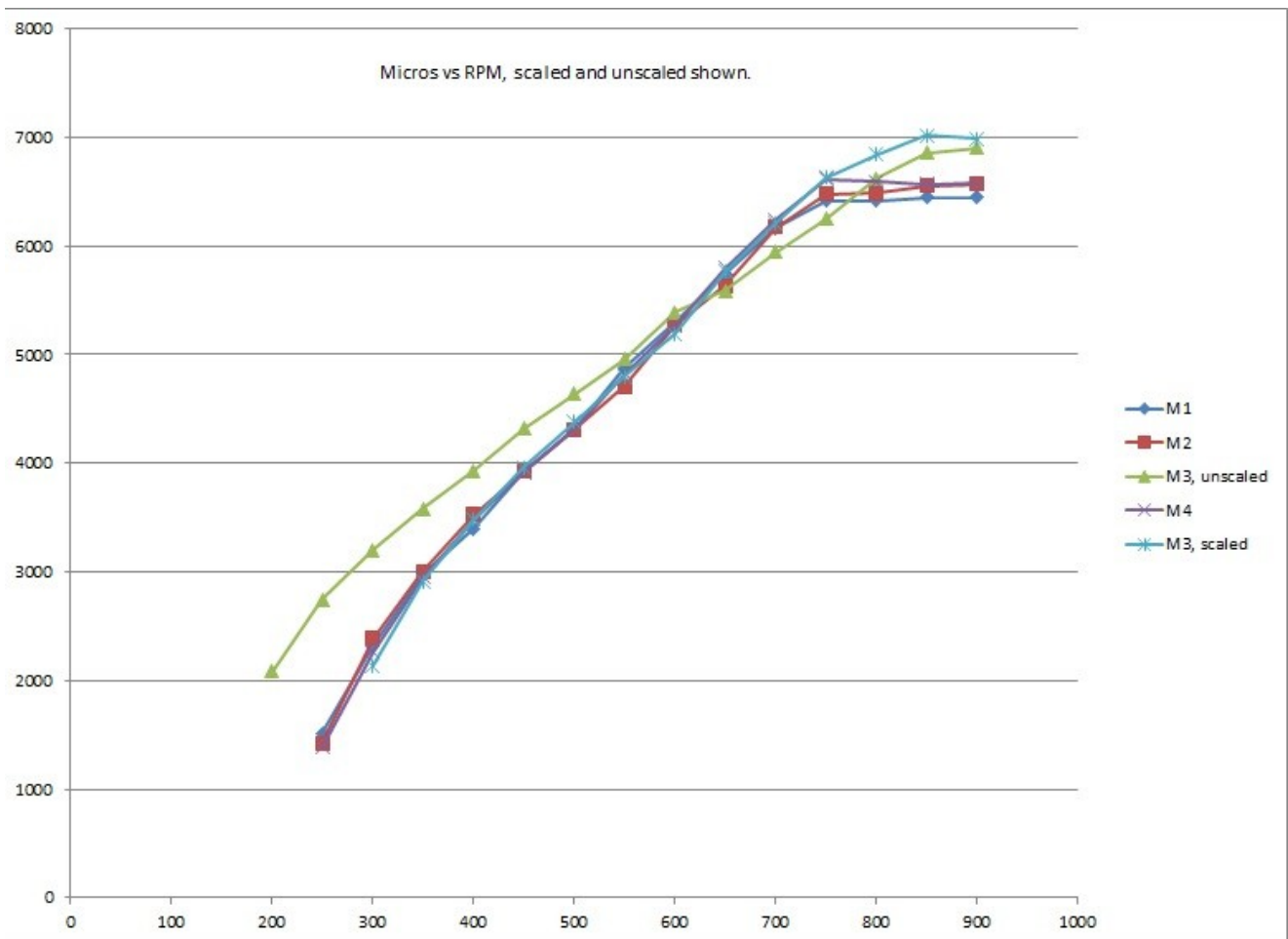


Figur 2: Turtall ved stroboskopet (grader [rad] vs frekvens [Hz])

Igjen hadde vi problemer med å få nøyaktige målinger, det var vanskelig å få stroboskopet til å treffe frekvensen til motorene nøyaktig. Videre var det snakk om så høye turtall at det ikke gikk an å lese av nøyaktig hastighet. Resultatet av denne testen førte til at vi valgte å gjøre enda en turtalls test.

Den andre turtalls testen ble gjort ved hjelp av en høy-precisjons mikrofon som ble plassert et par cm over rotorbladet. På denne måten vil mikrofonen fange opp luftstrømmen fra rotorbladet i det den passerer under mikrofonen. Mikrofonene ble koblet opp mot en laptop, hvor et oscilloscop-program kunne registrere frekvensen. Resultatene ble igjen ført inn i en tabell [5]

Resultatet sees i grafen på neste side.



Figur 3: Turtall ved frekvens analyse (pulsbredde (minus 1000us) [us] vs RPM [omdrengninger per minutt])

Her ser vi at resultatene er mye mer konsise, vi ser at motor en, to og fire sine frekvenser er like ved like pulsbreder, mens motor tre har tydelige produksjonsavvik og behøver skalering for å matche de andre motorene.

2.4 Videomodul

Videomodulen ble testet i to trinn, en komponenttest og en integrasjonstest.

Vi testet først kameraet for seg selv ved å koble det opp til en spenningskilde og en prosjekter, for deretter å bekrefte at prosjektoren korrekt viste kamerabildet.

Integrasjonstesten ble gjort ved å koble kameraet og senderen opp mot en spenningskilde, mottakeren ble koblet opp mot laptop vha en usb overgang. Riktig programvare[6,7] for visning av bildet måtte også lokaliseres, samt driver for modulen.

Etter at alt var satt opp kunne vi bekrefte at vi kunne se videostreamen på laptopen.

2.5 De siste 5 testene

2.5.1 Bevegelses testene

Det var forventet å oppnå en stabil hover på dette stadium og testene var beskrevet slik at bevegelsene, thrust, yaw, roll og pitch skulle ta utgangspunkt i en stabil hover. Det viste seg at det var umulig å få til en stabil hover manuelt. Dette er blandt annet fordi kreftene fra de ulike motorene er ulike. Det var blitt prøvd å skalere motorsignalene for å balansere kreftene, men uten suksess av den graden testene forventet. Derfor ble testene omskrevet siden hensikten med dem var å bekrefte at bevegelsene (grovt) stemte med justeringene på fjernkontrollen. Testene ble deretter godkjent.

2.5.2 Flytesten

Flytesten ble ikke godkjent. Selv om transformeringsfunksjonen omformer signalene fra fjernkontrollen riktig, og de testene ble godkjent, er det for mange andre faktorer som foregår som gjør det umulig å få fartøyet å hovre stabilt. Tydeligvis er kreftene ikke balanserte nok for å hovre. Om det går ann å hovre fartøyet ville det kreve veldig mye ressurser av selv en erfaren piloten bare for å holde fartøyet stabil, for ikke snakke om å få gjort noen oppgaver.

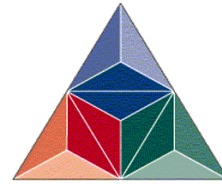
Konklusjoner

Dette er første gang vi kom til slutten av en iterasjon uten å være fornøyd med systemet da vi hadde store forventninger på at fartøyet kunne fly manuelt. Det viste seg at dette er nesten umulig på grunn av ulikeheter mellom motorene.

Vi har undersøkt litt på dette og ser at mange andre også mener det er umulig å fly manuelt. Vi har også fått ulike tips om å justere tyngdepunktet til fartøyet for å øke stabiliteten - da ved å senke dette.

Referanser

- [1] Dragonfly (2011) Testlogg_2.0: Kongsberg
- [2] Dragonfly (2011) Testspesifikasjon_1.0: Kongsberg
- [3] <http://git.projectdragonfly.no:3000/documents/42>
- [4] <http://git.projectdragonfly.no:3000/documents/45>
- [5] <http://git.projectdragonfly.no:3000/documents/44>
- [6] <http://www.nchsoftware.com/index.html>
- [7] <http://www.videolan.org/vlc/>



HØGSKOLEN
i Buskerud

Oppgavetittel: Hovedprosjekt - Testrapport iterasjon 5 - v2.0		
Fag(nr./navn): SFHO 3200 / Prosjektstyring med hovedprosjekt		
Gruppemedlemmer, signatur:		
_____ Vegard Torkelsen	_____ Michael Odden	_____ Herå Rørvik
_____ Trine Lindberg	_____ Luke Carambot	_____ Patrick Bjørum
Intern veileder: Dag Samuelsen Intern sensor: Olaf Hallan Graven Ekstern veileder og sensor: Ragnvald Otterlei		
Dato: 2011-05-18		

Innholdsfortegnelse

Dokumenthistorie.....	1
Innledning.....	1
1 Oversikt over testene i iterasjon 5.....	2
2 Rapporten.....	2
2.1 Forberedelse til testene.....	2
2.2 Ubelastet turtallstest.....	2
Konklusjoner.....	3
Referanser.....	4

Dokumenthistorie

Versjon	Beskrivelse	Når
1.0	Dokument opprettet, rettskrevet og ferdigstilt.	2011-04-07
2.0	Forandret tittelen fra testrapport 5	2011-05-18

Tabell 1: Dokumenthistorie

Innledning

Dette dokumentet tok for seg testene som ble gjennomført i iterasjon 5.

1 Oversikt over testene i iterasjon 5

I denne iterasjonen ble det i hovedsak utført tester som ikke var direkte koblet opp noen krav, testene var for å kunne identifisere parametere i den motor modellen.

2 Rapporten

2.1 Forberedelse til testene

For å kunne utføre disse testene trengte vi noe utstyr.

Vi utførte en turtallstest hvor motorene var ubelastede, for dette trengte vi et stroboskop.

2.2 Ubelastet turtallstest

I de tidligere turtallstestene utførte vi testen med propeller montert på motorene. For å kunne identifisere motorparametere behøvde vi i tillegg en turtallstest hvor motorene var ubelastede.

Denne turtallstesten ble gjennomført som de tidligere testene; motorene ble merket slik at vi kunne kjenne igjen posisjonen til motoren. Deretter ble pulsbredden inn til ESCen økt med jevne mellomrom samtidig som vi leste av rotasjonshastigheten.

Vi målte også spenningen over ESCen ved hver måling, samt strømmen igjennom den.

Resultatet er som vist under

Vs	Vs%	Micros	RPM	Amps	Rad/s	kb
12.41	3.1025	1250	3650	0.21	382.22	0.008117
	3.723	1300	7750	0.4	811.58	0.004587
	4.3435	1350	9677	0.52	1013.37	0.004286
	4.964	1400	10500	0.58	1099.55	0.004515
	5.584	1450	11200	0.62	1172.86	0.004761
	6.205	1500	11425	0.64	1196.42	0.005186
	6.8255	1550	11550	0.66	1209.51	0.005643
	7.446	1600	11700	0.67	1225.22	0.006077
	8.0665	1650	11800	0.68	1235.69	0.006528
	8.687	1700	11950	0.68	1251.40	0.006942

	9.3075	1750	12100	0.69	1267.10	0.007345
	9.928	1800	12100	0.69	1267.10	0.007835
	10.5485	1850	12100	0.69	1267.10	0.008325
	11.169	1900	12100	0.7	1267.10	0.008815
	11.7895	1950	12200	0.7	1277.58	0.009228
	12.41	2000	12200	0.7	1277.58	0.009714

Tabell 1. Måleresultater ubelasted turtallstest.

Formelen for beregning av kb stammer fra [1]

Utifra målingene hadde vi håpet å kunne regne ut parameteren kb, dette er en parameter som skal være konstant, sett utifra resultatet stemmer ikke dette med våre beregninger.

Dette kan være grunnet to ting.

1. Vi har gjort målinger eller beregninger feil.
2. Antagelsen at vi kan anvende beregninger og formler for vanlig DC motor stemmer ikke.

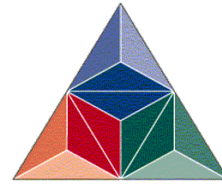
Diskusjon rundt denne testen vil bli tatt med intern veileder så snart vedkommende er tilbake fra et utenlandsopphold.

Konklusjoner

Testens resultat ble ikke slik vi hadde forventet og vi må arbeide videre må å finne den ønskede konstanten kb.

Referanser

[1] Modelling and high performance control of electrical machines, Chiasson John.



HØGSKOLEN
i Buskerud

Oppgavetittel: Hovedprosjekt - Testrapport iterasjon 9 - v1.0		
Fag(nr./navn): SFHO 3200 / Prosjektstyring med hovedprosjekt		
Gruppemedlemmer, signatur:		
_____ Vegard Torkelsen	_____ Michael Odden	_____ Herå Rørvik
_____ Trine Lindberg	_____ Luke Carambot	_____ Patrick Bjørum
Intern veileder: Dag Samuelsen Intern sensor: Olaf Hallan Graven Ekstern veileder og sensor: Ragnvald Otterlei		
Dato: 2011-05-25		

Innholdsfortegnelse

Dokumenthistorie.....	1
Innledning.....	1
1 Oversikt over testene i iterasjon 9.....	2
2 Rapporten.....	2
2.1 Forberedelse til testene.....	2
2.2 Sensor integrasjonstest, TI705 og TI706.....	2
2.3 Integrasjonstest på reguleringssystemet TI710.....	2
2.4 Vibrasjonsdempere TS419.....	2
2.5 Av/på bryter for reguleringssystem TS114.....	2
2.6 Testing av reguleringssystemets stabilmodus TS113.....	3
Konklusjoner.....	3
Referanser.....	4

Dokumenthistorie

Versjon	Beskrivelse	Når
1.0	Dokument opprettet	2011-05-25

Tabell 1: Dokumenthistorie

Innledning

Dette dokumentet tok for seg testene som ble gjennomført i iterasjon 9.

1 Oversikt over testene i iterasjon 9

I denne iterasjonen ble det i hovedsak gjort intergrasjonstester av sensorer, testing av reguleringsystem og flytester.

2 Rapporten

2.1 Forberedelse til testene

For å kunne utføre disse testene trengte vi noe utstyr.

For å kunne teste reguleringsystemet, ble det konstruert en testbenk som begrenser aksebevegelsene til fartøyet, slik at vi kunne undersøke en akse om gangen.

2.2 Sensor integrasjonstest, TI705 og TI706

Disse testene[1] var for å testet at implementasjonen på mikrokontrolleren fungerer som forventet. Begge sensorene gir ut data etter bevegelser i fartøyet. På gyroen var det interessant å bekrefte at retning (fortegnet) og aksene stemte med bevegelser i fartøyet. Det tilsvarende måtte testes for akselerometeret. I tillegg til rådata, målt i Newton meter per sekund i andre, er den brukt for å regne ut en vinkel. Det er vinklene fra akselerometeret som blir brukt i reguleringsystemet. Integrasjon av sensorene ble godkjent.

2.3 Integrasjonstest på reguleringsystemet TI710

Som en integrasjonstest er målet å bekrefte at implementasjonen av reguleringsystemet fungerer som forventet på mikrokontrolleren. Utover testen har implementasjonen av reguleringsystemet blitt bekreftet gjennom testbenk GUIen. Det er klart at motorsignalene kommer tilbake til nullstillingen etter at stikkene er nullstilt.

2.4 Vibrasjonsdempere TS419

Vi har valgt å ikke bruke vibrasjonsdempere, derfor er testen ikke godkjent.

2.5 Av/på bryter for reguleringsystem TS114

Vi har testet funksjonaliteten til av/på bryteren til reguleringsystemet.

Denne funksjonen har blitt noe endret etter at det ble bekreftet at fartøyet var umulig å manøvrere helt uregulert. Denne funksjonen vil derfor veksle mellom gyroregulert og vinkelregulert modus.

Vi kunne klart se at knappen aktiverte de to modusene korrekt. Ved bryteren av (grønt led lys av) førte stikkebevegelsene til en endring i fartøyets rotasjonshastighet, mens ved bryteren på (grønt led lys på)

ga stikkebevegelsene en endring i vinkel.

2.6 Testing av reguleringsystemets stabilmodus TS113

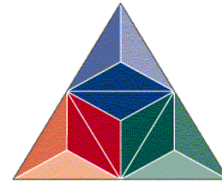
Systemtesten ble utført i testbenken. Vi har valgt å godkjenne denne testen selv om reguleringsystemets PIDer ikke er helt tunet. Vi kunne klart se at hvis vi slapp stikken som styrte den frie akselen i testbenken, returnert fartøyet tilbake til 0-punktet. Og bevegelser på stikker fører til en endring i vinkelen.

Konklusjoner

De testene som ble gjort hadde forventede resultater, sensorene og reguleringsystemet og av/på bryter fungerer som forutsett. Ingen vibrasjonsdempere er tilstede.

Referanser

[1] Testspesifikasjon_3.0



HØGSKOLEN
i Buskerud

Oppgavetittel: Hovedprosjekt - Brukermanual - v1.0		
Fag(nr./navn): SFHO 3200 / Prosjektstyring med hovedprosjekt		
Gruppemedlemmer, signatur:		
_____ Vegard Torkelsen	_____ Michael Odden	_____ Herå Rørvik
_____ Trine Lindberg	_____ Luke Carambot	_____ Patrick Bjørum
Intern veileder: Dag Samuelsen Intern sensor: Olaf Hallan Graven Ekstern veileder og sensor: Ragnvald Otterlei		
Dato: 2011-05-18		

Innholdsfortegnelse

Dokumenthistorie.....	1
Innledning.....	1
1 Opplasting av kode.....	2
1.1 Utviklingsmiljø.....	2
1.1 Opplasting.....	2
2 Fjernkontrollen.....	3
3 Tilkobling.....	5
3.1 Forhåndsregler.....	5
3.2 Tilkoblinger.....	5
4 Første flyvning.....	9
5 Testbenk GUI.....	9
Konklusjoner.....	11
Referanser.....	12

Dokumenthistorie

Versjon	Beskrivelse	Når
1.0	Opprettet dokument Rettskrevet og lagt til flere bilder, lagt til GUI	2011-05-24

Tabell 1: Dokumenthistorie

Innledning

Dette dokumentet er en brukermanual for fartøyet, slik at en lett kan bli kjent med fartøyets funksjoner og anvende disse.

1 Opplasting av kode

1.1 Utviklingsmiljø

For å kunne kompilere og laste opp programmet til mikrokontrolleren behøves et sett med verktøy. Disse finnes samlet under Arduinos utviklingsmiljø, under navnet "Arduino". Her får man et enkelt utviklingsmiljø med teksteditor, kompilator, opplastningsfunksjon og seriell monitor.

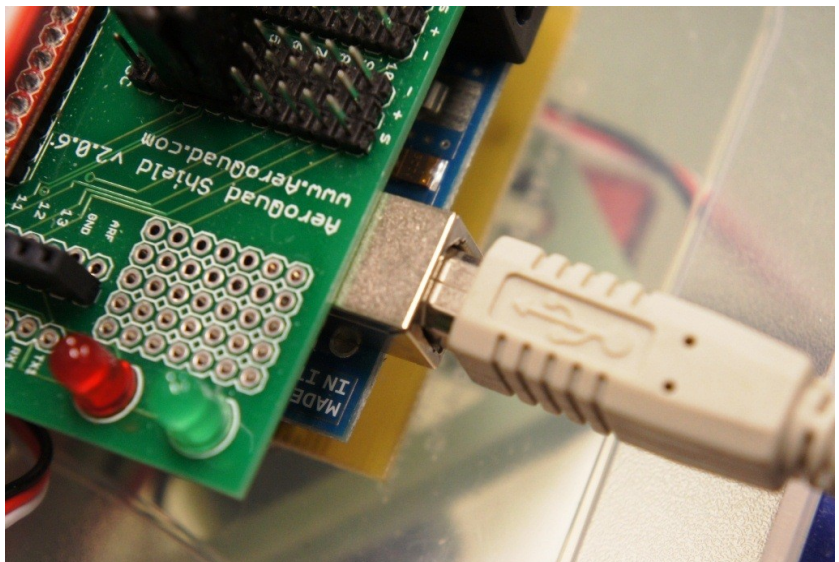
Arduino finnes for Windows, OS X (Mac) og Linux og kan lastes ned fra [1].

1.1 Opplasting

Når Arduino er installert, må filen `regulated.pde` åpnes. Naviger deg frem til filen, høyreklikk og velg "open with". Finn deretter kompilatoren og velg denne for å åpne.

Når denne filen åpnes vil alle tilhørende kodefiler automatisk åpnes sammen med denne og være tilgjengelig via fane-menyen over tekst-området.

Etter at dette er gjort, må PCen og mikrokontrolleren kobles sammen ved hjelp av en USB kabel. Se bildet under.



Figur 1: Tilkobling via USB

Når dette er gjort er man klar for opplasting. Trykk på opplastingsknappen, eller trykk `ctrl+U`.



Figur 2: Opplasting

Etter noen øyeblikk kommer meldingen "Done uploading" i kompilatorvinduet. Dette betyr at programvaren har blitt installert på mikrokontrolleren.

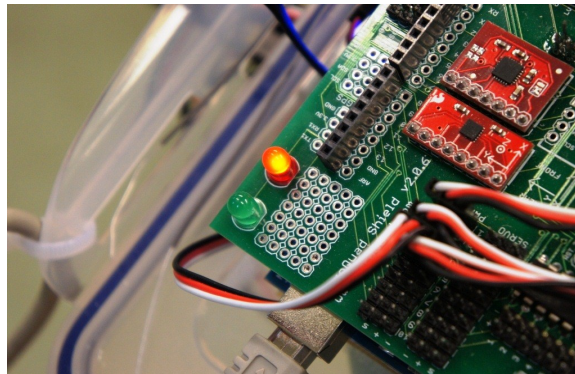
2 Fjernkontrollen



Figur 3: Fjernkontrollen

Før man kobler til batteriet, er det viktig at fjernkontrollen er slått på. Dette er for å sikre at mikrokontrolleren mottar riktige signaler fra mottakeren. Under kan man også se de forskjellige funksjonene til fjernkontrollen.

1. Denne knappen er en sikkerhetsbryter, når denne er i øvre posisjon (vekk fra brukeren) vil ikke motorene kunne rotere. Vi sier da at fartøyet ikke er armert. Hvis bryteren står i nedre posisjon (mot brukeren) vil motorene rotere. Denne brukes hvis fartøyet skal fraktes for hånd, mens batteriet er tilkoblet, eller hvis fartøyet kommer ut av kontroll. Hvis motorene er armerte vil det også lyse et rødt LED lys på koblingsbrettet øverst på fartøyet.



Figur 4: Armert fartøy

2. Knapp nummer to på bildet lar brukeren velge mellom gyro-regulert eller stabil modus. Hvis bryteren står i øvre posisjon er fartøyet i gyro-regulert modus, og hvis den står i nedre posisjon vil fartøyet være i stabil modus. Hvis fartøyet er i stabil modus, vil det lyse et grønt LED lys på koblingsbrettet øverst på fartøyet.



Figur 5: Indikator for reguleringsmodus

3. Venstre stikke lar brukeren styre thrust og yaw. Beveg stikken oppover/nedover for å endre thrust, og side til side for å styre fartøyet i yaw. Når stikken går mot høyre vil fartøyet rotere med klokken og venstre blir da mot klokken.
4. Høyre stikke lar brukeren styre roll og pitch. Beveg stikken til høyre/venstre for å styre fartøyet i roll, og opp eller ned for å bevege fartøyet i pitch.
5. De små knappene på sidene av hver stikke lar brukeren justere trim. Det vil si at hvis fartøyet er i stabil modus, og ikke regulerer seg til en stabil hover, men regulerer seg mot en annen vinkel kan denne brukes for å endre setpunktet slik at reguleringen blir korrekt. Dette kan ofte

oppstå hvis fartøyet har blitt initialisert på en flate som ikke er horisontal.

6. Dette er av/på bryteren på fjernkontrollen.

3 Tilkobling

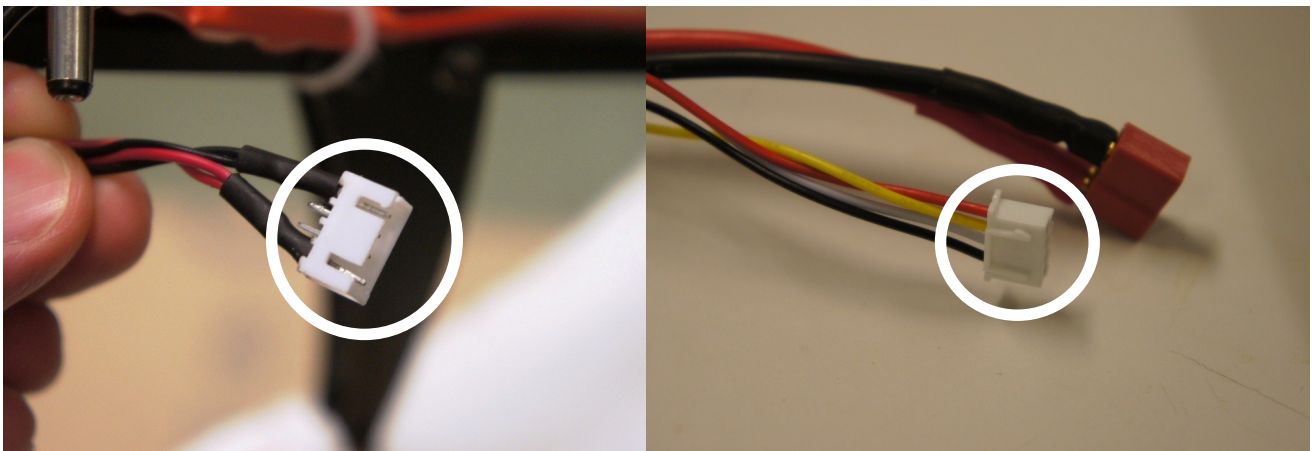
Når brukeren har lastet opp koden og fått kjennskap til hvordan kontrollen fungerer, skal batteriet, batterialarm og kameraet kobles til.

3.1 Forhåndsregler

Før batteriet kobles til må brukeren være sikker på at motorene **IKKE** er armerte, og at venstre stikke (3) er **HELT NEDE**. Når motorene roterer kan de føre til alvorlige skader på brukeren, personer rundt, fartøyet eller andre gjenstander. Hvis man er usikker på om motorene er armerte, ta av propellene før oppstart for sikkerhets skyld.

3.2 Tilkoblinger

Når brukeren er klar kobles batterialarmen til batteriet, se figuren under. De er da de to hvite pluggene som skal kobles sammen.

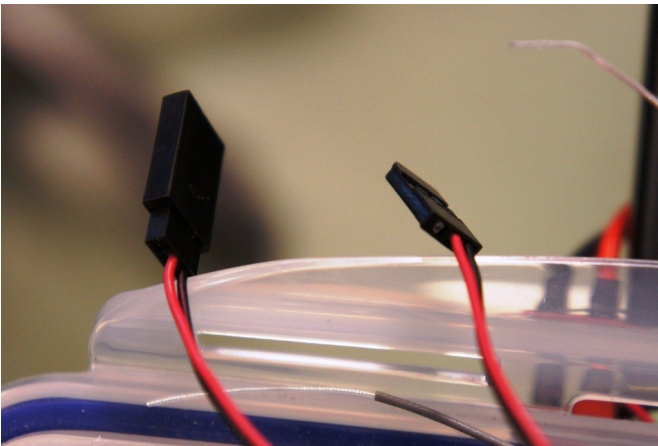


Figur 6: Batterialarmen

Batterialarmen vil da signalisere at den måler spenningen ved å gi et lydsignal. Batterialarmen har et lysende blått LED-lys hvis spenningen på batteriet er akseptabelt.

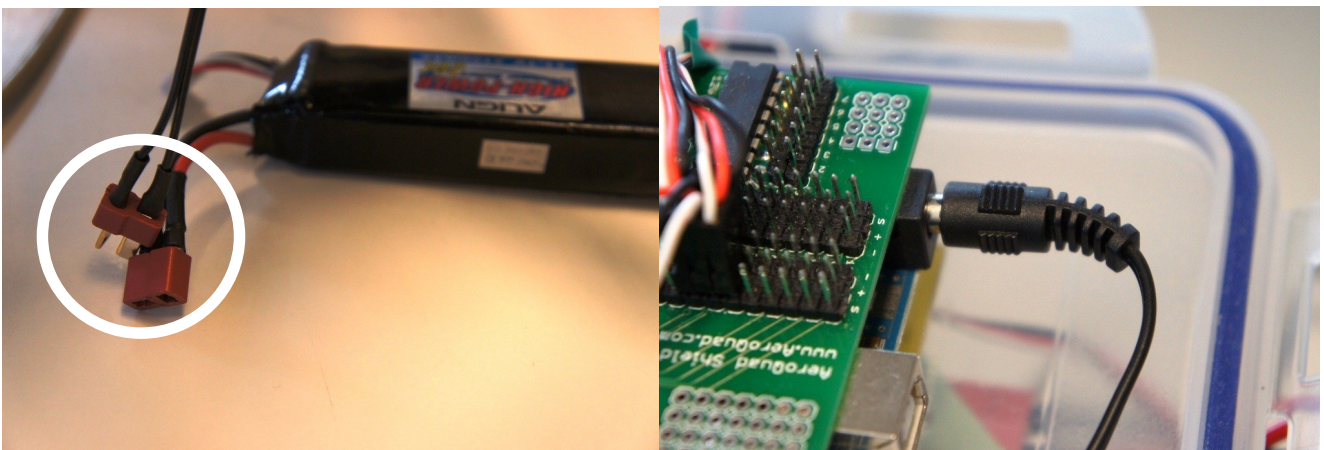
Når batterialarmen er koblet til, kan brukeren koble til spenningen til mikrokontrolleren. Her finnes det to metoder som kan brukes avhengig av om man skal bruke et eller to batterier.

Metode 1, ett batteri: Sett sammen to kablene som vist på figur 7, Se figuren under.



Figur 7: Spenningsforsyning til mikrokontroller

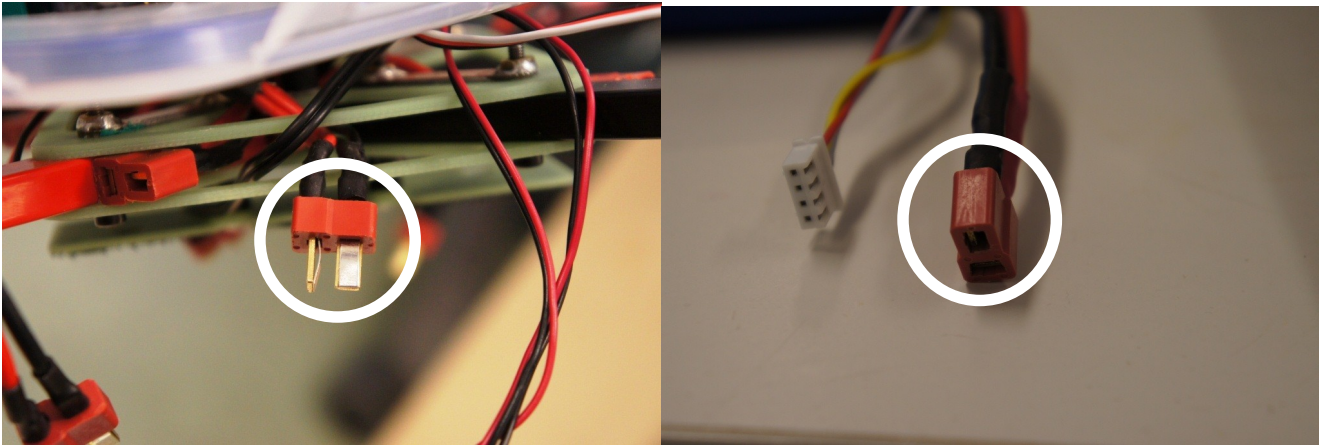
Metode 2, to batterier: Først kobles sekundær-batteriet (det minste) til, se figur under. Deretter kobles pluggen til mikrokontrolleren.



Figur 8: Sekundær batteri

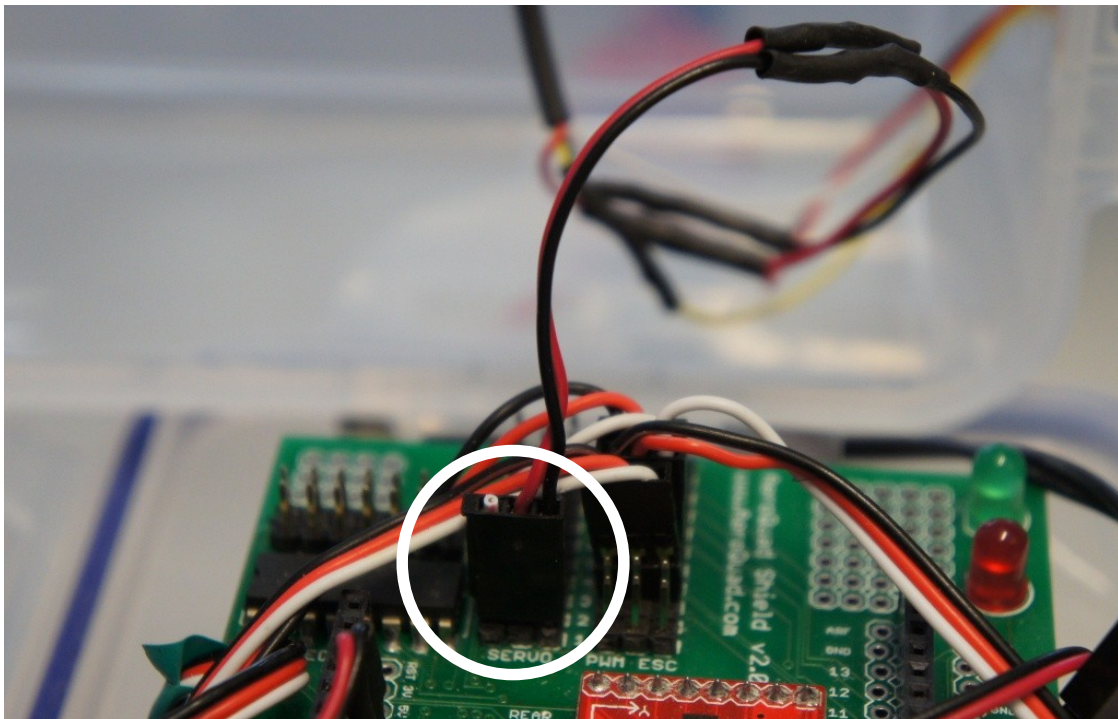
Det er viktig at fartøyet står i ro på en horisontal flate når denne kobles til. Dette er fordi fartøyet oppstarts rutine vil begynne og sensorene vil nullstille seg.

Etter dette kan batteriet kobles til, se figuren på neste side.



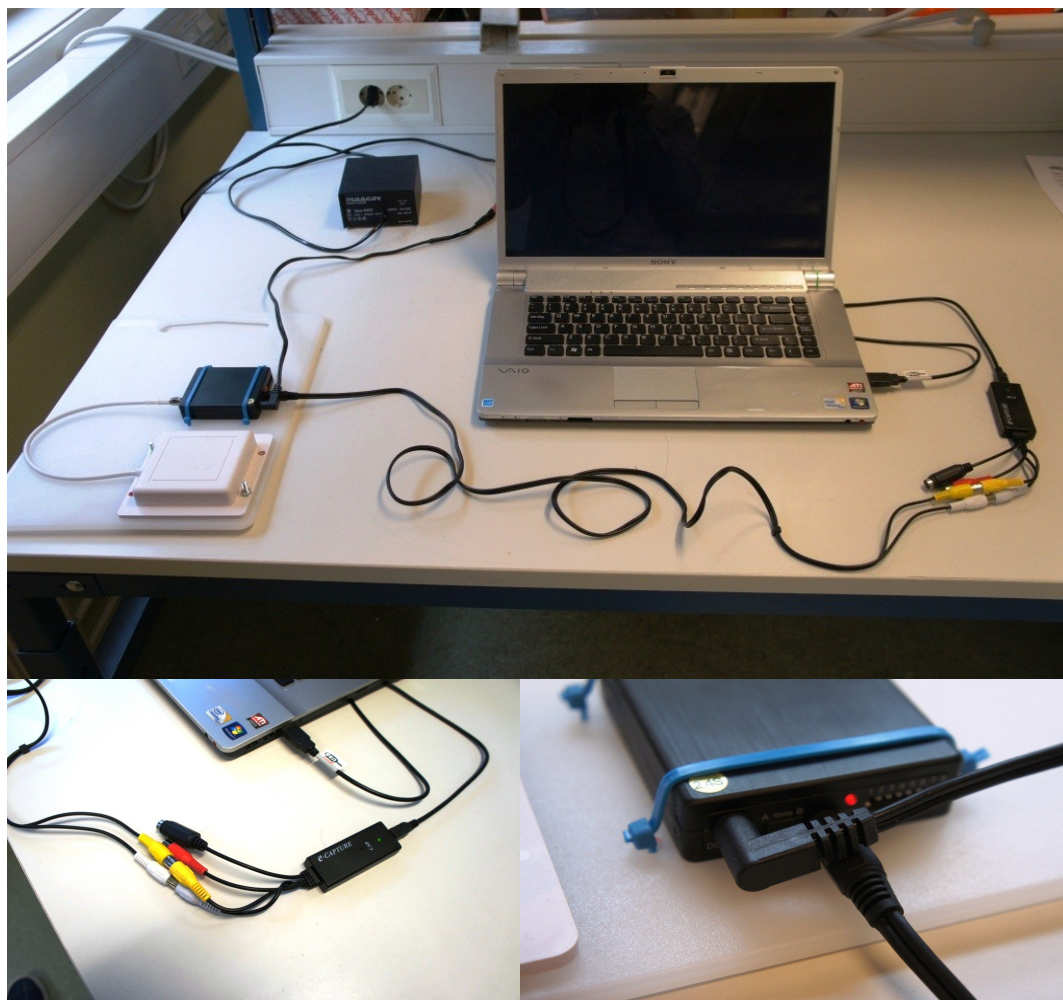
Figur 9: (Primær) Batteri

Det siste som kobles til er kameramodulen. Kameramodulens spenningstilkobling ligger i beskyttelseslokket til fartøyet, der kameraet er montert. Denne kobles til servo pin 1 på koblingsbrettet, pass på polariteten, her er det mulig å koble feil! Se figur 10.



Figur 10: Kameramodul

Når dette er gjort, er fartøyet klart. Da gjenstår det bare å koble opp mottakeren til kameramodulen til bakkestasjonen. Antennen kobles til mottakerenheten, spenningsforsyningen kobles til mottakerenheten. Deretter kobles USB overgangen til en PC og mottakerenheten. Se figuren under.



Figur 11: Bakkestasjon

Når dette er koblet opp må brukeren åpne et passende videoprogram, vi brukte "Debut Video Capture Software". Dette fordi vi opplevde en stor tidsforsinkelse på videooverføringen da vi brukte andre programmer. Dette programmet er gratis og kan lastes ned gratis på [2].

Etter at programmet er installert og åpnet, må brukeren endre noen innstillinger for å få opp korrekt video.

Gå til Options, velg Webcam/ Capture Device. Under Name må brukeren velge USB2.0 ATV. Deretter velg Device Settings og huk av PAL/BDGHI og Composite Video.

Trykk OK på denne skjermen og på neste. Fartøyets videostream skal nå sees på skjermen.

Hvis brukeren ønsker å ta opp videostreamen, må formatet velges som .avi, trykk så på record knappen, dette gjøres nederst, under videostreamen.

Når det er ønskelig å avslutte opptaket, trykk på stop knappen.

Hvis brukeren ønsker å se opptaket, velg Recordings over videostreamen, da skal alle opptak dukke opp og kan spilles av.

4 Første flyvning

Hvis brukeren har lite erfaring med å fly slike fartøyet anbefales det å fly i omgivelser med så lite vind som mulig og i et område hvor man ikke kan sette andre personer i fare. Brukeren burde først øve på å åpne en stabil hover i hofte høyde. Det er også viktig at brukeren har den fremre delen av fartøyet vekk fra seg selv, slik at stikkebevegelsene samsvarer med bevegelsene på fartøyet. Se figuren under.



Figur 12: Bruker i forhold til fartøyets posisjon, rødt ben indikerer frem

Hvis fartøyet kommer ut av kontroll, kan det ofte være bedre å skru av motorarmeringen slik at fartøyet umiddelbart faller istedet for å prøve å gjenoppta kontrollen. Fly gjerne i et område hvor en dårlig landing ikke vil påføre fartøyet store skader, slik som ute over gress eller lignende.

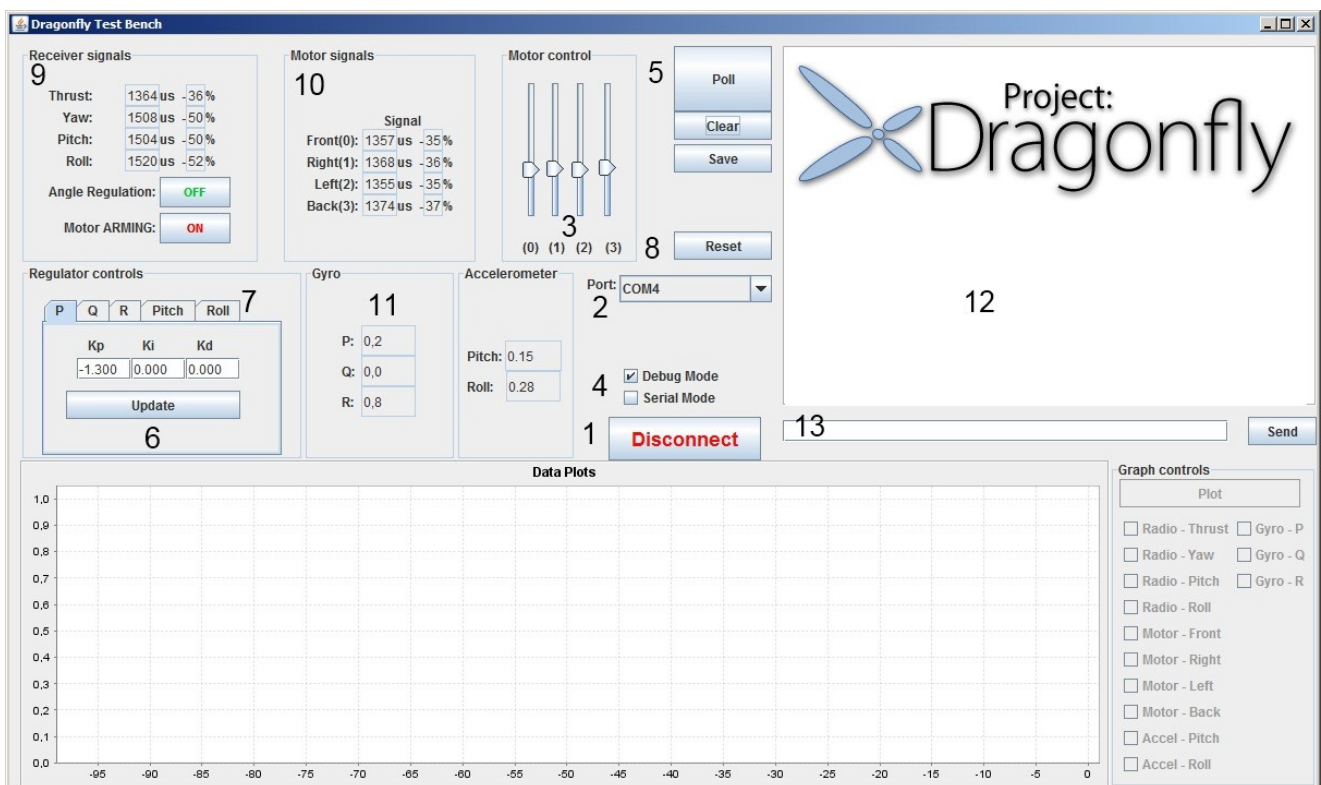
5 Testbenk GUI

Hvis brukeren ønsker å endre parametere i programvaren, slik som PID parametere, anbefales det å bruke testbenk-programvaren. Denne lar brukeren enkelt koble fartøyet opp mot programvaren, slik at det ikke blir nødvendig å last opp ny programvare for hver gang.

1. Koble fartøyet til pc, slik som ble gjort under programvare opplasting.
2. Åpne DragonflyTestbench.jar
3. Velg riktig port fra "port" vinduet
4. Trykk "connect"

Brukeren er da klar til å bruke programmet.

I figuren under er en oversikt over programmets funksjoner.



Figur 13: Testbenk GUI

1. Connect/Disconnect, kobler fartøyet til eller fra programvaren
2. Port. Her velger brukeren porten som fartøyet er koblet til datamaskinen.
3. Motor Controls. Her kan brukeren overstyre fjernkontrollen, her kan brukeren sende signaler rett til motorene, de står i rekkefølgen motor 1,2,3 og 4. For å bruke denne funksjonen må "Serial mode" være aktivert.
 - Ved aktivert debug mode vil GUIen motta oppdaterte verdier på motorer, radio og sensorer i sanntid.
 - Ved aktivert serial mode så kan brukeren overstyre radiosignalene fra kontrollen.
4. Serial mode og debug mode. Her kan brukeren velge å aktivere/deaktivere disse modusene.

5. Poll, Ved å trykke på denne henter programmet ut verdiene fra PIDene på fartøyet og sjekker om fartøyet står i regulated mode eller ikke.
6. Oppdater, denne knappen sender de PID verdiene brukeren har tastet inn i programmet til fartøyet.
7. Her kan brukeren velge hvilken PID er som skal justeres, P,Q og R er roll pitch og yaw på gyro-regulert, mens Pitch og Roll er tilsvarende på vinkelregulert.
8. Reset, denne knappen nullstiller sensorene.
9. Denne tabellen viser signalene fra radioen, den lar deg også overstyre armeringsknappen og modusknappen fra fjernkontrollen.
10. Denne tabellen viser signalene mikrokontrollen sender ut til motorene grafen til høyre oppdateres også.
11. Her vises sensorinformasjonen i sanntid
12. her kan brukeren observere alle kommandoene som er blitt sendt til/fra GUIen
13. Her kan brukeren manuelt taste inn kommandoer

Graf funksjonen er ikke implementert i programvaren, men er klar for videre utvikling.

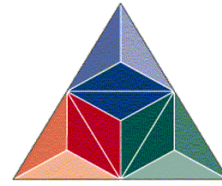
For en full oversikt over kommandoene se[3]

Konklusjoner

Sikkerhet er svært viktig under oppkobling og flyvning, propellene roterer i svært høye hastigheter og kan medføre alvorlige skader. Uerfarne brukere må lære seg å hovre før de prøver seg på mere avanserte bevegelser, det anbefales å øve i en simulator.

Referanser

- [1] <http://arduino.cc/en/Main/Software> (2011-05-19)
- [2] <http://www.nchsoftware.com/capture/index.html> (2011-05-20)
- [3] testbenk_1.0.pdf



HØGSKOLEN
i Buskerud

Oppgavetittel: Hovedprosjekt - Multikopter teori - v1.0		
Fag(nr./navn): SFHO 3200 / Prosjektstyring med hovedprosjekt		
Gruppemedlemmer, signatur:		
_____ Vegard Torkelsen	_____ Michael Odden	_____ Herå Rørvik
_____ Trine Lindberg	_____ Luke Carambot	_____ Patrick Bjørum
Intern veileder: Dag Samuelsen Intern sensor: Olaf Hallan Graven Ekstern veileder og sensor: Ragnvald Otterlei		
Dato: 2011-01-31		

Innholdsfortegnelse

Dokumenthistorie.....	1
Innledning.....	1
1 Kapittel.....	2
1.1 Bakgrunnshistorie.....	2
1.2 Manøvrering.....	2
2 Kapittel.....	3
2.1 Forskjellige prosjekter.....	3
Referanser.....	5

Dokumenthistorie

Versjon	Beskrivelse	Når
1.0	Dokument opprettet Dokument ferdigstilt	2011-04-06

Tabell 1: Dokumenthistorie

Innledning

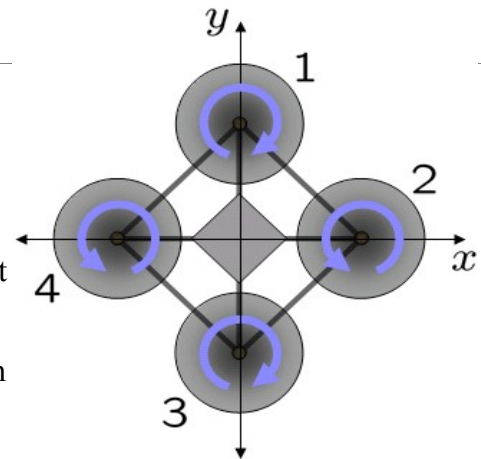
Vi skal skrive litt bakgrunnshistorie og teori om multi-koptere.

1 Kapittel

1.1 Bakgrunnshistorie

Betegnelsen Quadroter er et helikopter som løftes av fire forskjellige propeller. De er klassifisert som rotorcraft som er det samme som vanlige helikopter. En ting som skiller de to (Bortsett fra antall propeller) er at quadroteren har fast-vinklet blader.

Quadroteren var en av de første konstruksjonene som kunne ta en suksessfull VTOL (Vertical Take Off and Landing) med et en eller flere passasjerer. Senere har den i all hovedsak blitt brukt til UAVs (Unmanned Aerial Vehicles) hvor den er lett å styre pga flere elektroniske sensorer til å stabilisere.



Figur 1: Multikopter akser

Noen av årsakene til at man foretrekker multikopter framfor vanlige helikoptre til UAV er fordi man blant annet slipper å ha en mekanisk kobling som justerer vinkelen på bladene som gjør at man kan forenkle designen og spare inn på kostnader. I tillegg er det mye mindre kinetisk kraft i hver lille rotor som gjør at eventuelle skader blir mye mindre ved en ulykke.

1.2 Manøvrering

Quadroteren (Multi-kopteret) styres på følgende måte:

Hver rotor produserer både en skyvekraft og et dreiemoment i senteret for rotasjonen.

Hvis alle rotorene roterer med samme vinkelhastighet, med rotor 1 og 3 roterende med klokken og rotor 2 og 4 mot klokken, så er det totale aerodynamiske dreiemomentet, og dermed også vinkel akselerasjonen rundt z-aksen nøyaktig null (Hvorav z-aksen er aksens i sentrum innover i bildet). Dette betyr at man ikke trenger noen ekstra stabilisering i z-aksen som vanlige helikoptre bruker halerotoren til.

Styringen av z-aksen her blir altså kontrollert av differansen fra rotor-parene som roterer i hver sin retning, så ved å øke kraft på rotor 1 og 3 mens man minker kraften proporsjonalt på 2 og 4 så vil fartøyet rotere mot klokken, mens man fortsatt beholder samme oppdrift.

Vinkel akselerasjon rundt x og y-aksen kan oppnås separat uten å ha noen innvirkning på z-aksen. Hvert rotor-par som roterer i samme retning styrer en akse, som betyr at man kan drive den framover (roterer over x-aksen) eller til siden (roterer over y-aksen). Det gjøres ved at man øker skyvekraften på den ene rotoren og minker på proporsjonalt den andre slik at man får en ubalansert kraft over enten x eller y-aksen, men beholder samme vinkel akselerasjon rundt z-aksen, samt samme oppdrift.

2 Kapittel

2.1 Forskjellige prosjekter.

Vi er langt ifra de eneste som driver med multi-koptere ute i verden, og det finnes mange imponerende ting som foregår. Et par eksempler:

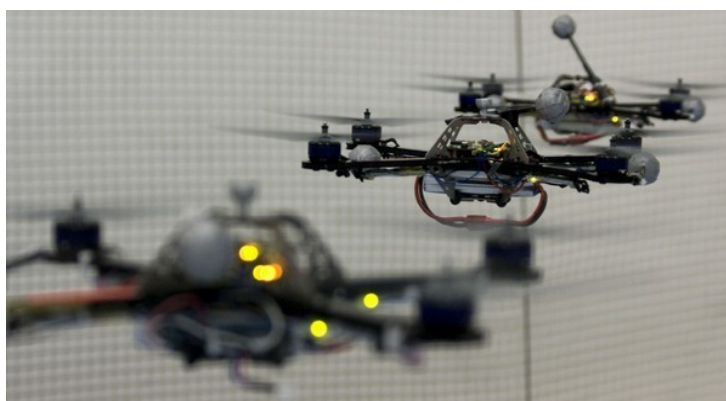
AR.Drone fra Parrot var de første i verden med et multi-kopter som kunne styres med bare iPhone via wifi. I tillegg til å kunne styre den, så får man også opp live video feed på iPhonens skjerm fra et av de to kameraene som sitter på fartøyet.

De oppdaterer stadig programvaren og nå har det kommet mulighet for to personer å fly sammen og konkurrere mot hverandre ved at man skyter på hverandres fartøy med virtuelle raketter på iPhonen.



Figur 2: AR Drone

På universitetet ETH i Sveits har de noe som de kaller Flying Machine Arena som er et område på 10x10x10m hvor de satt opp åtte bevegelses følsomme kameraer som gir informasjon om posisjon og bevegelse i rommet til multi-kopterene deres.



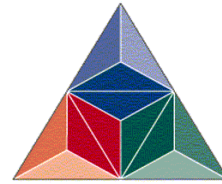
Figur 3: Multikoptere fra ETH

Der har de utført en rekke spennende demonstrasjoner hvor blant annet multi-kopterene kan sprette ping-pong baller i luften, danse

koordinert sammen, ekstrem akrobatikk, spille piano og med samarbeid av tre stykker; løfte tyngre objekter helt koordinert. Alt gjøres med samspill av kameraer og programvare, helt uten input fra en bruker med fjernkontroll.

Referanser

- [1] <http://en.wikipedia.org/wiki/Quadcopter>
- [2] <http://ardrone.parrot.com/parrot-ar-drone/en/>
- [3] http://www.idsc.ethz.ch/Research_DAndrea/FMA



HØGSKOLEN
i Buskerud

Oppgavetittel: Hovedprosjekt - Komponentdokumentasjon - v3.0		
Fag(nr./navn): SFHO 3200 / Prosjektstyring med hovedprosjekt		
Gruppemedlemmer, signatur:		
_____ Vegard Torkelsen	_____ Michael Odden	_____ Herå Rørvik
_____ Trine Lindberg	_____ Luke Carambot	_____ Patrick Bjørum
Intern veileder: Dag Samuelsen Intern sensor: Olaf Hallan Graven Ekstern veileder og sensor: Ragnvald Otterlei		
Dato: 2011-04-05		

Innholdsfortegnelse

Dokumenthistorie.....	1
Innledning.....	1
1 Motorer og tilhørende komponenter.....	1
1.1 DC motorer.....	1
1.2 Propeller.....	2
1.2.1 Puller propeller.....	2
1.2.2 Pusher propeller.....	2
1.3 ESC-Estimated Speed Controller.....	2
2 Sensorer.....	3
2.1 Gyro.....	3
2.2 Akselerometer.....	3
3 Mikrokontroller.....	4
4 Koblingsbrett.....	4
5 Batteri.....	5
6 Chassi.....	5
Konklusjon.....	6
Referanser.....	6

Dokumenthistorie

Versjon	Beskrivelse	Når
1.0	Dokument opprettet	2010-12-14
2.0	Lagt til bilder og lagt til tilleggsinformasjon. Standardisert	2011-01-03
3.0	Rettet skrivefeil, standardisert iht ny mal.	2011-04-05

Tabell 1: Dokumenthistorie

Innledning

Dette dokumentet er for å beskrive de forskjellige komponentene vi har valgt å bruke til fartøyet. Dokumentet tar også for seg hvorfor vi har gått for akkurat disse komponentene. Informasjonen er i all hovedsak hentet fra forhandleren de selges fra.

1 Motorer og tilhørende komponenter

1.1 DC motorer



Figur 1: Motor

DC motorene som skal bestilles, er av typen hacker børsteløs 20-22L.[1] Denne motoren har en maks spenning på 11V, og et kontinuerlig arbeidsområde mellom 6-14A (maks 17A). Batteriet vi skal bestille er beregnet for bruk innenfor radiostyrte prosjekter, og er derfor et naturlig valg for vårt prosjekt. Motorene er også relativt små, med en total lengde på 4.6cm og en ytre diameter på 2.8 cm. Dette fører til at motorene er lette i vekt, bare 56gram hver. Lette komponenter er ønskelige for å forlenge flyve-tiden.

1.2 Propeller



Figur 2: Propell

Propellene vi bruker, består av to par propeller, et par pusher propeller og et par puller propeller. Dette betyr at den ene typen gir løftekraft når den går med klokken, og den andre gir løftekraft når den går mot klokken. Det er nødvendig med disse to typene propeller for å kunne styre fartøyets vinkel i yaw.

1.2.1 Puller propeller

Puller propellene vi har valgt, er av typen APC10x4.7 Puller propellers.[2] Dette er en “vanlig” propell som er beregnet for medium til større fartøyer og passer perfekt til 20-22L motorene.

1.2.2 Pusher propeller

Pusher propellene vi har valgt er av typen APC10x4.7 Pusher propellers.[3] Denne propellen er identisk som pusher propellene, bortsett fra bladene står i reversert vinkel, slik at man oppnår løftekraft når de roteres i motsatt retning.

1.3 ESC-Estimated Speed Controller



Figur 3: ESC

Motorene behøver styring, og dette gjøres vha en estimated speed controller, eller estimert hastighetskontroller. Hastighetskontrolleren fungerer som en Digital/analog konverter, hvor et digitalt signal fra mikro-kontrolleren styrer hvor mye spenning som går over motoren. Det behøves en ESC for hver motor, og vi har valgt ESC av typen turnigy plush 18amp speed controller[4]. Vi kan se utifra databladet til kontrolleren at den kan ha en kontinuerlig strøm på maks 18A og har en lineær utgangskurve. Dette er en ESC som er beregnet for hobbymotorer og faller som et naturlig valg for våre motorer.

2 Sensorer

For å kunne regulere fartøyets bevegelser behøver vi sensorer som kan måle bevegelsene, slik at vi har noe å regulere mot.

2.1 Gyro



Figur 4: Gyro

Gyroen vi har valgt er av typen ITG 3200 tripple-axis gyro[5] Denne gyroen vil måle vinkelhastighetene i alle tre akser. Gyroen inneholder en 16-bits analog-digital-konverter, slik at vi får oversatt signalet på en slik måte at mikro-kontrolleren kan forstå det. Den har i tillegg en fleksibel spenningsforsyning, fra 2.1V til 3.6V. Dette gjør det enda enklere å gi implementere den inn i systemet vårt.

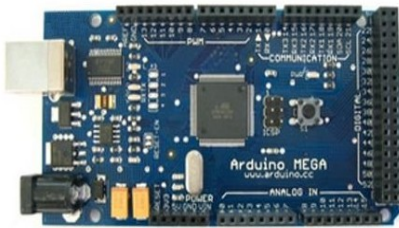
2.2 Akselerometer



Figur 5: Akselerometer

Vi vil i tillegg bruke et akselerometer for å kunne måle akselerasjonen til fartøyet i de tre aksene. Vi har valgt en av typen BMA180 triple axis accelerometer.[6] Dette akselerometeret gir oss et utgangssignal i form av 14bits digitalt signal. Den inneholder også en rekke filtre vi kan bruke for å enkelt kunne behandle utgangssignalet. I tillegg har den en funksjon som detekterer støt, denne kan brukes til å undersøke om fartøyet har landet. Måleområdet til sensoren er fra $\pm 1g$ til $\pm 16g$.

3 Mikrokontroller



Figur 6: Arduino MEGA 2560

For å kunne bearbeide inngangssignalene og oversette de til signaler som motorene kan bruke, må vi anvende en mikrokontroller. Vi har valgt å bruke en av typen Arduino MEGA /w Atmega 2560[7]. Dette er en mikrokontroller som anvender åpen kildekode. Den er utstyrt med 54 digitale input/output porter og 14 PWM porter. Den bruker 7-12V spenningskilde, og har en 16MHZ klokkehastighet. Dette er en kraftig, men billig mikrokontroller som passer vårt behov perfekt. Den tilbyr oss nok I/O porter i tillegg til at den er relativt liten og lett.

4 Koblingsbrett



Figur 7: Koblingsbrett

For å enkelt kunne koble mikro-kontrolleren og sensorene sammen, bruker vi et koblingsbrett. Vi kjøper derfor et ferdiglaget koblingsbrett[8], som enkelt lar oss koble gyroen og akselerometere sammen med mikro-kontrolleren. Koblingsbrettet inneholder også plasser for flere sensorer, slik at produktet enkelt kan videreutvikles. Her finnes det også plasser for å koble sammen hastighetskontrollerene med mikro-kontrolleren.

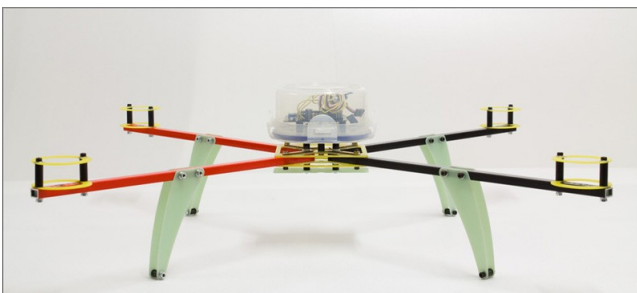
5 Batteri



Figur 8: Batteri

Naturligvis behøver alle komponentene våres en spenningsforsyning. Vi velger zippyflightmax 4000mAh 20C batteri[9] Dette er et tre cellers batteri, slik som motorene og ESC'ene våre krever. Batteriet har en spenning på 11.1V og en kapasitet på 4ampertimer. 20C betyr hvor stor konstant utladningshastighet den kan ha, som man regner ut ved å bare gange med kapasiteten: $4Ah \times 20C = 80A$ (holder i massevis til våre motorer). Batteriet veier totalt 306gram.

6 Chassi



Figur 9: Chassi

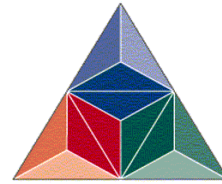
Vi behøver et chassis som holder alle komponentene sammen. Siden vi ikke har noen maskinstudenter på gruppen har vi valgt å kjøpe et ferdigkonstruert chassi Vi har valgt en som er konstruert av aluminium og fiberglass, som gjør det både stivt og lett.[10] Dette chassiet gir oss en plattform hvor vi kan montere komponentene på en enkel og trygg måte. I tillegg er det konstruert med beskyttelse på hver arm som passer på motorene ved evt. stygg landing.

Konklusjon

Etter å ha undersøkt de forskjellige komponentene, kan vi se at de er kompatible med hverandre.

Referanser

- [1] http://hobbyking.com/hobbyking/store/uh_viewItem.asp?idProduct=4700 (2010-12-16)
- [2] <https://www.aeroquadstore.com/ProductDetails.asp?ProductCode=LP10047> (2010-12-16)
- [3] <https://www.aeroquadstore.com/ProductDetails.asp?ProductCode=LP10047SFP> (2010-12-16)
- [4] http://hobbyking.com/hobbyking/store/uh_viewItem.asp?idProduct=4312 (2010-12-16)
- [5] <https://www.aeroquadstore.com/ProductDetails.asp?ProductCode=SEN-09801> (2010-12-16)
- [6] <https://www.aeroquadstore.com/ProductDetails.asp?ProductCode=SEN-09723> (2010-12-16)
- [7] <https://www.aeroquadstore.com/ProductDetails.asp?ProductCode=ARDUINO-04> (2010-12-16)
- [8] <https://www.aeroquadstore.com/ProductDetails.asp?ProductCode=AQ2-000> (2010-12-16)
- [9] http://hobbyking.com/hobbyking/store/uh_viewItem.asp?idProduct=7634 (2010-12-16)



HØGSKOLEN
i Buskerud

Oppgavetittel: Hovedprosjekt - Økonomi - v1.0		
Fag(nr./navn): SFHO 3200 / Prosjektstyring med hovedprosjekt		
Gruppemedlemmer, signatur:		
_____ Vegard Torkelsen	_____ Michael Odden	_____ Herå Rørvik
_____ Trine Lindberg	_____ Luke Carambot	_____ Patrick Bjørum
Intern veileder: Dag Samuelsen Intern sensor: Olaf Hallan Graven Ekstern veileder og sensor: Ragnvald Otterlei		
Dato: 2011-05-27		

Innholdsfortegnelse

Dokumenthistorie.....	1
Definisjoner og forkortelser.....	1
Innledning.....	1
1 Budsjet.....	2
2 Regnskap.....	3
Referanser.....	4

Dokumenthistorie

Versjon	Beskrivelse	Når
1.0	Opprettet dokument	2011-05-27

Tabell 1: Dokumenthistorie

Definisjoner og forkortelser

\$	Amerikanske Dollar
NOK	Norske kroner

Tabell 2: Definisjoner og forkortelser

Innledning

Dette dokumentet er for å samle de økonomiske excel arkene på et dokument.

1 Budsjett

budsjett_v.4.0				
Produkt Navn	Produktbeskrivelse	Pris	Valuta	Butikk
ITG 3200 Triple-Axis Gyro	3-akse gyro	50,00		\$Aeroquad Store
AeroQuad Shield v2.0	Koblingsbrett	30,00		\$Aeroquad Store
Stackable Headers - Mega Pack	Koblingspins	12,00		\$Aeroquad Store
Arduino MEGA 2560	Mikrokontroller brett	65,00		\$Aeroquad Store
2.1mm DC Power Connector for Arduino	Kontakt/ plugg	1,75		\$Aeroquad Store
APC 10x4.7 Slow Flyer Pusher Propellers (3 stk)	Propell	15,00		\$Aeroquad Store
APC 10x4.7 Slow Flyer Propellers (3 stk)	Propell	10,50		\$Aeroquad Store
Estimert frakt		35,00		\$Aeroquad Store
Receiver Cables (8stk)	Servo/ mottaker kabler	10,00		\$diydrone.com
Connection Headers	Rad med koblingspunkter	2,25		\$diydrone.com
Angle pin headers (2stk)	45 graders koblingspunkter	4,00		\$diydrone.com
Estimert frakt		3,00		\$diydrone.com
BMA180 Triple Axis Accelerometer	3-aksers akselerometer	30,00		\$Sparkfun.com
Estimert frakt		8,50		\$Sparkfun.com
KDA20-22L Hacker style brushless outrunner (4 stk)	Motor	59,00		\$Hobbyking.com
Zippy Flightmax 4000mAh 3S1P 20C	Batteri	20,00		\$Hobbyking.com
TR_P18a Plush 18amp ESC (5stk)	Spennings kontroller	59,50		\$HobbyKing.com
Estimert frakt		20,00		\$Hobbyking.com
Futaba R617FS 7-Channel 2.4GHz FASST Receiver	Mottaker	89,00		\$Flying-Hobby.com
Estimert frakt		21,00		\$Flying-Hobby.com
KKquadcopter Airframe	Helikopter kropp	89,00		\$kkmulticopter.com
Estimert frakt		16,00		\$kkmulticopter.com
Maxpro Battery Monitor	Batterialarm	6,00		\$HeliDirect.com
Estimert frakt		7,00		\$HeliDirect.com
Estimert fortolling		45,00		\$
Utskrift av plakat		100,00		NOKHiBu
Utskrift av hovedprosjekt		1050,00		NOKHiBu
Total \$		708,50		\$
Valuta (per 16.12.2010)		5,97		\$
Total NOK		4229,75		NOK
MVA		1,25		%
Total NOK inkl MVA		6437,18		NOK

Figur 1: Budsjett [1]

Budsjettet er et estimat over hva delene koster inkl. Fortolling. Prisene er hentet fra diverse nettbutikker som har de delene vi trenger.

2 Regnskap

regnskap_v.3.0				
Produkt Navn	Produktbeskrivelse	Pris	Valuta	Butikk
ITG 3200 Triple-Axis Gyro	3-akse gyro	50,00		\$Aeroquad Store
AeroQuad Shield v2.0	Koblingsbrett	30,00		\$Aeroquad Store
Stackable Headers - Mega Pack	Koblingspins	12,00		\$Aeroquad Store
Arduino MEGA 2560	Mikrokontroller brett	65,00		\$Aeroquad Store
2.1mm DC Power Connector for Arduino	Kontakt/ plugg	1,75		\$Aeroquad Store
APC 10x4.7 Slow Flyer Pusher Propellers (3 stk)	Propell	15,00		\$Aeroquad Store
APC 10x4.7 Slow Flyer Propellers (3 stk)	Propell	10,50		\$Aeroquad Store
Frakt		35,50		\$Aeroquad Store
Receiver Cables (8stk)	Servo/ mottaker kabler	10,00		\$diydrones.com
Connection Headers	Rad med koblingspunkter	2,25		\$diydrones.com
Angle pin headers (2stk)	45 graders koblingspunkter	4,00		\$diydrones.com
Frakt		3,00		\$diydrones.com
BMA180 Triple Axis Accelerometer	3-aksers akselerometer	30,00		\$Sparkfun.com
Frakt		8,50		\$Sparkfun.com
KDA20-22L Hacker style brushless outrunner (4 stk)	Motor	55,00		\$Hobbyking.com
Zippy Flightmax 4000mAh 3S1P 20C	Batteri	20,00		\$Hobbyking.com
TR_P18a Plush 18amp ESC (5stk)	Spennings kontroller	50,00		\$HobbyKing.com
Frakt		20,00		\$Hobbyking.com
Futaba R6L7FS 7-Channel 2.4GHz FASST Receiver	Mottaker	89,00		\$Flying-Hobby.com
Frakt		23,00		\$Flying-Hobby.com
KKquadrocopter Airframe	Helikopterkropp	89,00		\$kkmulticopter.com
Frakt		16,00		\$kkmulticopter.com
Maxpro Battery Monitor	Batterialarm	6,00		\$HeliDirect.com
Frakt		7,00		\$HeliDirect.com
KDA20-22L Hacker style brushless outrunner (2 stk)	Motor	29,50		\$Hobbyking.com
Frakt		3,50		\$HobbyKing.com
Diverse kabler og koblingspunkter		15,00		\$HiBu
Fortolling og MVA		42,00		\$
Total \$		742,50		\$
Valuta (per 16.12.2010)		5,97		\$
Total NOK		4432,73		NOK
Spennings forsyning	Spenning til kameramottaker			Lånt av Simicon
BlackWidowAV KX131 Cased camera	Kamera			Lånt av Simicon
BlackWidowAV Receiver	Mottaker til kamera			Lånt av Simicon
BlackWidowAV BWAV240500	Sender til kamera			Lånt av Simicon
Futaba 6EX 2.4GHz	Sender til fjernstyring			Lånt av Patrick Bjørnum

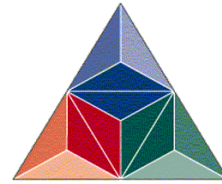
Figur 2: Regnskap [2]

Regnskapet er en oversikt over hva de egentlige kostnadene ble. Total kostnad er lavere enn budsjettet blant annet fordi flere av delene trengte ikke fortolling. (Verdi under 200 kroner)

Referanser

[1] Budsjett_4.0

[2] Regnskap_3.0



HØGSKOLEN
i Buskerud

Oppgavetittel: Hovedprosjekt - Designdokument reguleringsystem - v2.0		
Fag(nr./navn): SFHO 3200 / Prosjektstyring med hovedprosjekt		
Gruppemedlemmer, signatur:		
_____ Vegard Torkelsen	_____ Michael Odden	_____ Herå Rørvik
_____ Trine Lindberg	_____ Luke Carambot	_____ Patrick Bjørum
Intern veileder: Dag Samuelsen Intern sensor: Olaf Hallan Graven Ekstern veileder og sensor: Ragnvald Otterlei		
Dato: 2011-05-23		

Innholdsfortegnelse

Dokumenthistorie.....	1
Definisjoner og forkortelser.....	1
Innledning.....	1
1 Valg av regulator.....	2
2 Kort beskrivelse av PID regulator.....	2
2.1 Pseudokode.....	3
3 Reguleringsystemet.....	3
3.1 Blokkdiagram.....	4
3.1.1 Vinkel regulator.....	5
3.1.2 Rotasjons regulator	5
Konklusjoner.....	6
Referanser.....	7

Dokumenthistorie

Versjon	Beskrivelse	Når
1.0	Første offisielle versjon	2011-04-11
2.0	Lagt til nytt design, påbygg av reguleringsystemet Lagt inn blokkdiagrammer Lagt til definisjoner og forkortelser Lagt til tekst under «reguleringsystemet»	2011-05-23

Tabell 1: Dokumenthistorie

Definisjoner og forkortelser

PID	Proporsjonal-integral-derivat regulator
Pitch	Rotasjon rundt lengden av fartøyet
Roll	Rotasjon rundt bredden av fartøyet
Yaw	Rotasjon rundt sin egen akse

Tabell 2: Definisjoner og forkortelser

Innledning

Dette er et designdokument for reguleringsystemet til fartøyet. Dokumentet tar for seg hva vi har lagt til grunn for valg av regulator, samt nærmere beskrivelse av det valgte reguleringsystemet.

1 Valg av regulator

For valg av regulator la vi følgende til grunnlag:

- God regulering
- Enkelt å designe
- Enkelt å simulere
- Enkel implementering
- Enkel tuning
- Vår kunnskap

Vi måtte så finne en regulator som både vil regulere godt nok og som samtidig ikke er for kompleks. Ut ifra dette kom vi frem til at vi i første omgang går for PID regulering. Dette blant annet fordi dette er den regulatoren vi kan mest om, og som også er blant de enkleste og likefrem både å designe, simulere, implementere og tune. Vi har også sett at mange andre liknende prosjekter bruker PID regulering.

2 Kort beskrivelse av PID regulator

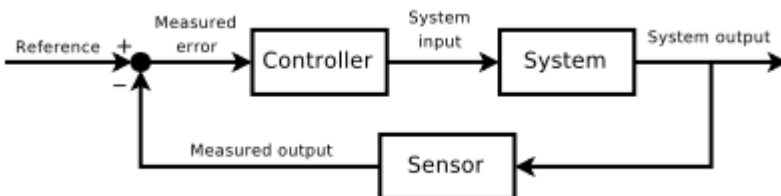
En proporsjonal-integral-derivat kontroller (PID) er den mest vanlige tilbakekoblingskontrolleren. En PID-regulator beregner feilen mellom målt og ønsket(referanse) verdi. Kontrolleren forsøker å minimere feilen ved å justere utgangen fra PIDen.

Den matematiske likningen for en PID:

$$u(t) = MV(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{d}{dt} e(t)$$

hvor u er utgangen fra PIDen, K_p , K_i og K_d er pådrag til P, I og D leddene, og e er forskjellen mellom målt og ønsket verdi - "error".

Nedenfor vises et generelt blokkdiagram for en tilbakekoblingskontroller, Controller-blokken er hos oss en PID:



Figur 1: Kontrollstruktur

2.1 Pseudokode

Her er en enkel software loop som implementerer PID-algoritmen i den ideelle parallelle form:

```
previous_error = 0
integral = 0
start:
    error = setpoint - actual_position
    integral = integral + (error*dt)
    derivative = (error - previous_error)/dt
    output = (Kp*error) + (Ki*integral) + (Kd*derivative)
    previous_error = error
    wait(dt)
    goto start
```

3 Reguleringsystemet

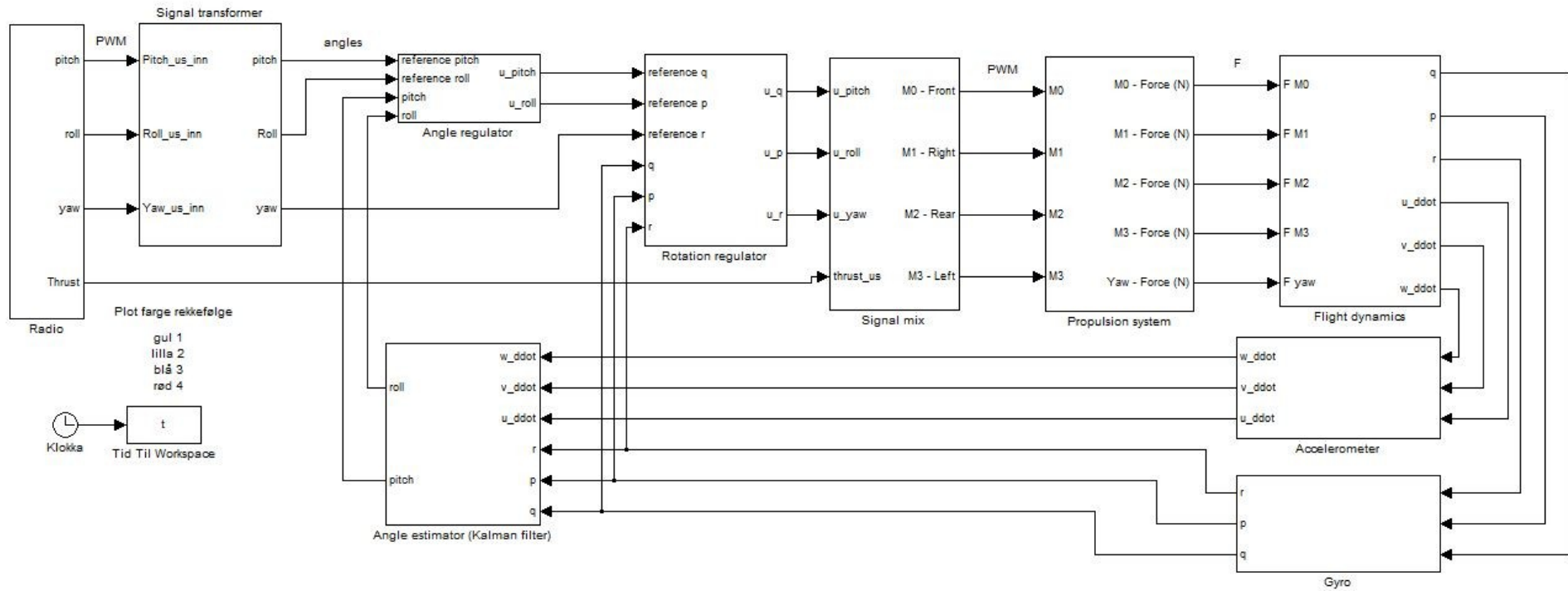
Vi ønsker å bruke fem separate PIDer, tre PIDer for å regulere mot vinkelhastighet i retningene roll, pitch og yaw, og to PIDer til å regulere mot vinkler i roll og pitch.

Referansesignalene til roll og pitch i vinkelreguleringen er radiosignalet fra fjernkontrollen, mens referansesignalet til roll og pitch i rotasjonsreguleringen er lik utgangen fra vinkelregulatorene hvis disse er koblet til, hvis ikke kommer også disse fra fjernkontrollen. Referansesignalet for rotasjonsreguleringen av yaw kommer direkte fra fjernkontrollen.

Tilbakekoblingssignalene til vinkelreguleringen får vi ved å bruke et kalman-filter på vinklene fra gyroen og akselerometeret, mens vinkelhastigheten til rotasjonsregulatorene hentes direkte fra gyroen.

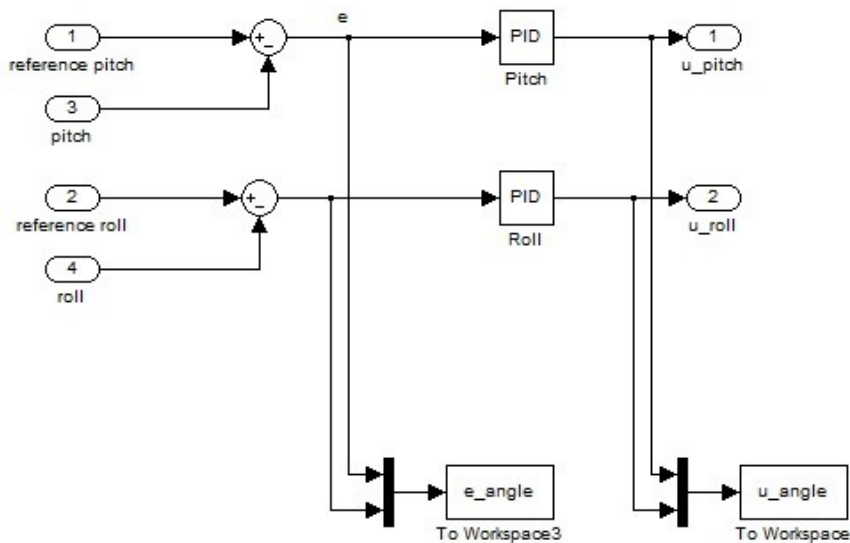
Vi starter først med kun rotasjonsregulering, da hentes tilbakekoblingssignalene direkte fra gyroen, så dette kaller vi gyro-modus. I denne modusen er det mulig å fly, men det er ikke enkelt for en uerfaren flyger. Derfor implementerer vi også vinkelreguleringen, som skal gjøre fartøyet enkelt å fly for hvem som helst, man skal til og med kunne slippe stikkene, og fartøyet skal da gå i stabil hover av seg selv. Denne reguleringsmodusen kaller vi stabil modus.

3.1 Blokkdiagram



Figur 2: Blokkdiagram

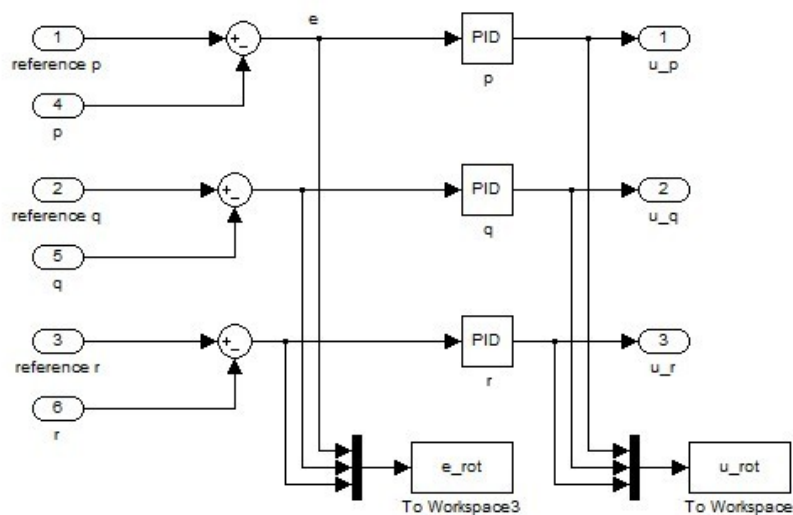
3.1.1 Vinkel regulator



Figur 3: Angle regulator

Vinkelregulatoren sammenligner referansevinklene fra fjernkontrollen i roll og pitch med vinklene fartøyet har ifølge sensorene (gjennom Kalman filteret). Den inneholder to PIDer som gir pådrag etter hvor stor forskjellen er.

3.1.2 Rotasjons regulator



Figur 4: Rotation regulator

Rotasjonsregulatoren bruker utgangene fra vinkelregulatoren som referanser for vinkelhastighet i roll

og pitch. Referansen til yaw kommer direkte fra fjernkontrollen. Vinkelhastighetene sammenlignes med verdiene fra gyroen i hver av aksene, og PIDene gir pådrag dersom vi har en forskjell i en referanse og målt gyroverdi.

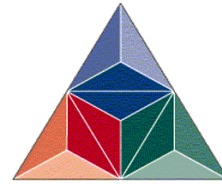
Systemdesignet er tegnet i Simulink, se også eget simuleringsdokument [1].

Konklusjoner

Vi har designet et reguleringssystem bestående av 5 PIDer. Systemet skal først simuleres, så implementeres, testes og tunes.

Referanser

[1] simuleringsdokument_1.0



HØGSKOLEN
i Buskerud

Oppgavetittel: Hovedprosjekt - Designdokument testbenk - v3.0		
Fag(nr./navn): SFHO 3200 / Prosjektstyring med hovedprosjekt		
Gruppemedlemmer, signatur:		
_____ Vegard Torkelsen	_____ Michael Odden	_____ Herå Rørvik
_____ Trine Lindberg	_____ Luke Carambot	_____ Patrick Bjørum
Intern veileder: Dag Samuelsen Intern sensor: Olaf Hallan Graven Ekstern veileder og sensor: Ragnvald Otterlei		
Dato: 2011-04-07		

Innholdsfortegnelse

Dokumenthistorie.....	1
Innledning.....	1
1 Krav.....	2
1.1 Fysisk.....	2
1.2 Programvaren.....	2
2 Design.....	2
2.1 Serial Protocol.....	2
2.1.1 Syntax.....	2
2.1.2 Funksjoner (func).....	3
2.1.3 Parametre.....	3
2.1.4 Modus.....	3
2.1.5 Returverdier.....	4
2.1.6 Eksempler.....	4
2.2 Illustrasjoner.....	5

Dokumenthistorie

Versjon	Beskrivelse	Når
1.0	Første versjon, lagt til tegningen	2011-01-26
2.0	Integrert wiki side Seriel protocol , lagt til bilde av GUI	2011-04-12
3.0	Endret tittelen fra testbenk til testbenk_designdokument	2011-05-20

Tabell 1: Dokumenthistorie

Innledning

Dette dokument er å dokumentere krav og planen til en testbenk.

1 Krav

Her er kravene til testbenken. Disse kan deles opp i to grupper, den fysiske delen og programvare delen.

1.1 Fysisk

Denne delen skal begrense bevegelsene til fartøyet.

1. Fartøyet skal ikke kunne komme over 45°
2. Fartøyet skal ikke kunne komme over 1,5m i høyde
3. I alle mulige stillingene skal propellene ikke komme bort i noe (noen)
4. Det skal være mulig å koble en USB kabel fra data til fartøyet
5. Begrense bevegelsene ned til en akse (for testing)

1.2 Programvaren

Denne skal gjøre det mulig å få vite hva i praksis skjer i mikrokontrolleren.

1. Til prototype 1 skal det kunne leses av
 1. RCMottakkersignaler (4x)
 2. Motorsignalene
 3. Gyro data (sekundær)
 4. Akselerometer (sekundær)
2. Til prototype 2 skal det kunne leses av
 1. RCMottakkersignaler (4x)
 2. Motorsignalene
 3. Gyro data
 4. Akselerometer
 5. Parametere til regulatoren
3. Testbenkprogrammet skal overføre data til/fra mikrokontrolleren gjennom USB.
4. Mottatt data skal lagres til analyse.
5. Presentasjon av sanntidsdata, eventuelt skriving til mikrokontrolleren skal skje gjennom et brukergrensesnitt på datamaskin.

2 Design

2.1 Serial Protocol

Hensikten er å kartlegge kommunikasjonsmetodene og kravet ifm testbenkens lesing og skriving til fartøyet ved testing.

Data er overført serielt. Det er hensiktsmessig å ha størrelsen på data så små som mulig. I tråd med dette brukes de to første bitene til func og param h.h.v. Etter disse to, skal resten av bitene beskrive value.

2.1.1 Syntax

Alle kommandoer vil være i følgende format:

[<func>] <param> [<value>]

2.1.2 Funksjoner (func)

Når kommandoer er sent til fartøyet skal <func> være sett. Tillatte verdier er 0-1 og sier om fartøyet skal skrive eller lese. Data fra fartøyet skal ikke ha <func> med men starter med <param> biten - se underkapittelet "Eksempler".

Func	Ascii	Navn	Resultat
>	62	set	Parameteret skal skrives i fartøyet
<	60	get	Parameteret skal leses av fartøyet

Tabell 2: Kommandofunksjoner

2.1.3 Parametre

Dette er de kommandoene vi skal kunne sende til, og motta fra, fartøyet. Tillatte verdier er 0-127.

Param	Ascii	Navn	Tillatte verdier	Resultat
=	61	mode	0-7	Setter fartøyet i valgt modus. Dette er sum av flere modus - se Modus
#	35	regulation	0/1	Deaktiverer/aktiverer regulering
:	58	camera	0/1	Deaktiverer/aktiverer kamera
^	94	gyro	0/1	Deaktiverer/aktiverer gyro
_	95	acc	0/1	Deaktiverer/aktiverer akselerometer
A,B,C,D	65+<N>	engine_thrust_<N>	1000-2000	Overstyrer pulsbredden sendt til motorens ESC (mikrosekunder)
E,F,G,H	69+<N>	engine_factor_<N>	0.0-1.0	Setter en "permanent" skalering på pulsbreddene sendt til motorens ESC
a,b,c,d,e,f	96+<I>	radio_<I>	1000-2000	Lar deg simulere input via radio for de ulike kanalene (mikrosekunder)
o,p,q,r,s,t	110+<I>	radio_factor_<I>	0.0-1.0	Setter en "permanent" skalering på pulsbreddene som signaler (I < 7)

Tabell 3: Kommandoene

2.1.4 Modus

Da det vil være hensiktsmessig å dele opp mikrokontrollerens oppførsel i ulike modus ønsker vi å kunne angi disse. Ved noen moduser er det enkelte kommandoer som ikke vil fungere. Dette utdypes nærmere.

Moduser kan summeres bitvis.

Verdi	Modus	Binær verdi	Beskrivelse
1	debug	001	Aktiverer debug-modus, dvs sender all data til seriell utgang
2	regulated	010	Aktiverer regulert system. Dvs leser input også fra gyro og acc
4	serialflight	100	Fullstendig styrt via testbenk

Tabell 4: Moduser

Modus "debug"

Settes LSB høy vil debug-modus aktiveres og all log-data sendes direkte over den serielle forbindelsen.

Modus "regulated"

Settes andre bit lav aktiveres uregulert modus hvor fartøyet ikke vil ta hensyn til akselerometer eller gyroskop, bare radio-input. Settes derimot bit'en høy vil regulert modus aktiveres.

Modus "serialflight"

Settes MSB høy vil radio-input ignoreres til fordel for parametere sendt fra testbenken.

2.1.5 Returverdier

Vi må bestemme format på data returnert fra fartøyet.

Eks:

```
// Fra testbenken:
"<C" // Les av motor2signalet

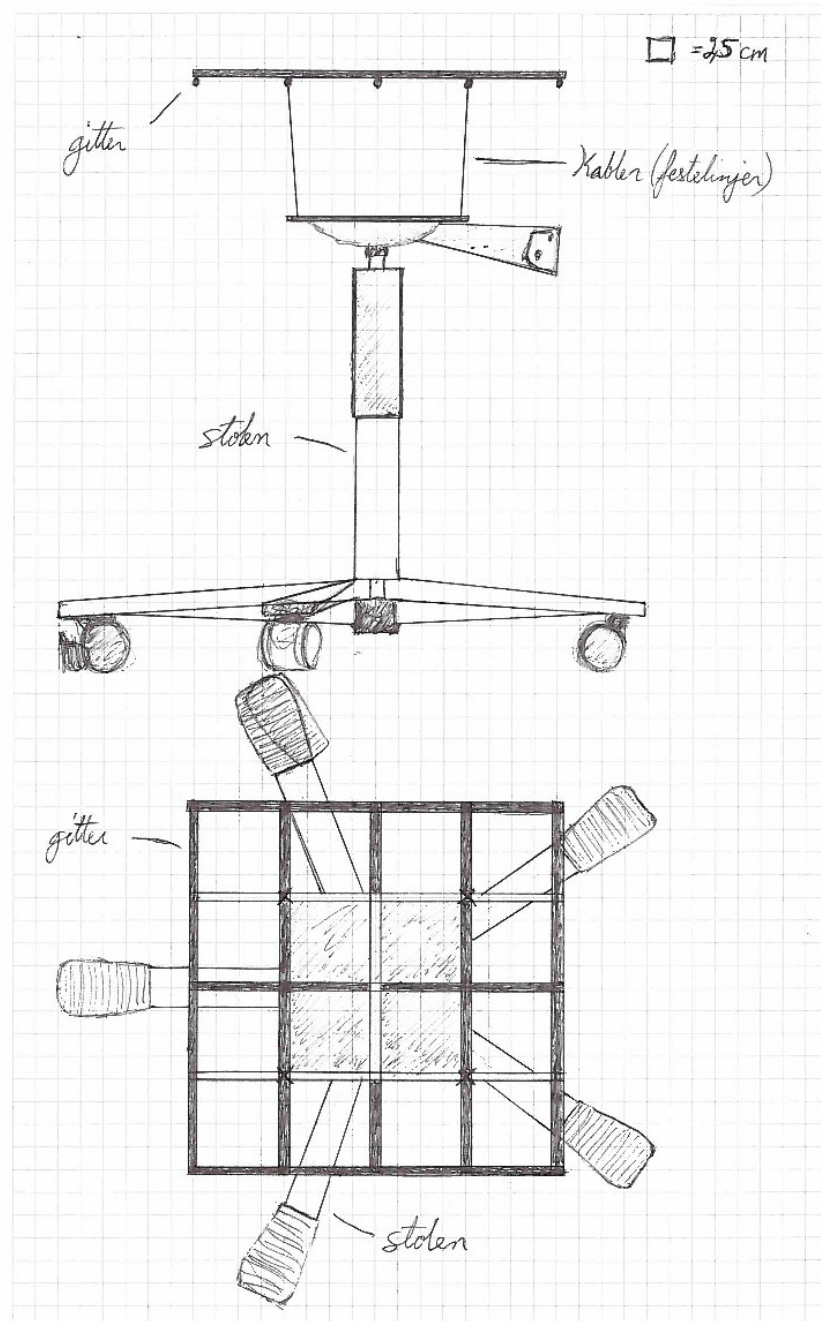
// Fra fartøyet:
"C1329" // motor2 er på 1329
```

2.1.6 Eksempler

```
">^1" // Gyro på
">C1500" // Setter motor2 til 1500 (ca. 50%)
"<=" // Returnerer hvilket modus fartøyet er i
">=3" // Aktiverer regulerte-debug modus
```

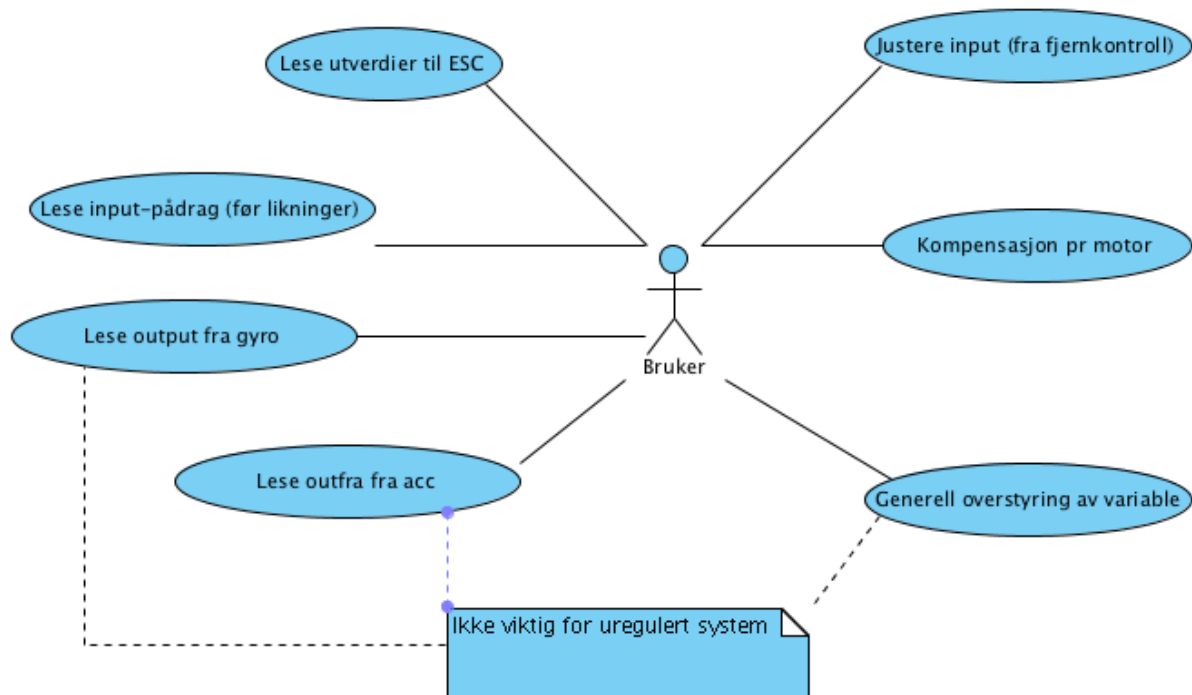
2.2 Illustrasjoner

Under er et forslag til design av den fysisk testbenken.

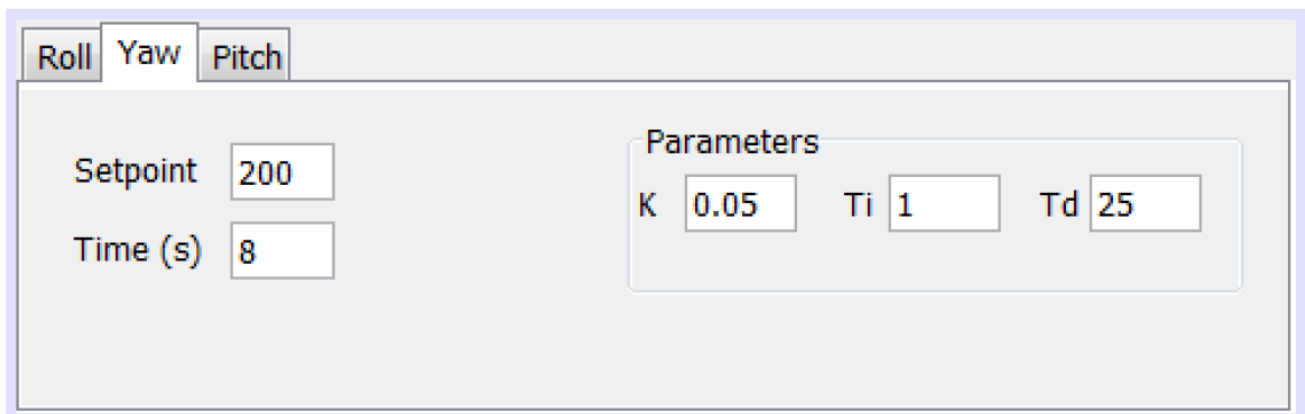


Figur 1: Testbenkdesign

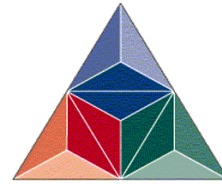
Under er en UseCase for å illustrere programmet.



Figur 2: Programdesign



Figur 3: reguleringssystemets GUI



HØGSKOLEN
i Buskerud

Oppgavetittel: Hovedprosjekt - Design av programvare for regulert fartøy - v2.0		
Fag(nr./navn): SFHO 3200 / Prosjektstyring med hovedprosjekt		
Gruppemedlemmer, signatur:		
_____ Vegard Torkelsen	_____ Michael Odden	_____ Herå Rørvik
_____ Trine Lindberg	_____ Luke Carambot	_____ Patrick Bjørum
Intern veileder: Dag Samuelsen Intern sensor: Olaf Hallan Graven Ekstern veileder og sensor: Ragnvald Otterlei		
Dato: 2011-04-12		

Innholdsfortegnelse

Dokumenthistorie.....	1
Innledning.....	1
1 Overordnede diagram.....	2
1.1 Komponentdiagram.....	2
1.2 Deploymentdiagram.....	3
2 Teknologi.....	4
2.1 Relevante teknologier.....	4
2.2 Valg, begrensninger og gitte arkitekturvalg.....	4
2.2.1 setup().....	4
2.2.2 loop().....	4
2.2.3 ESC.....	4
3 Funksjonalitet.....	5
3.1 Use Case diagrammer.....	5
3.1.1 Det regulerte fartøyet.....	7
3.1.2 Testbenk.....	8
3.2 Oppstart.....	9
3.3 Avslutning.....	9
3.4 Autopilot av/på.....	9
3.5 Flyvning.....	10
3.6 Regulering.....	12
3.7 Nedprioriterte krav / funksjoner.....	13
3.7.1 Batterisjekk.....	13
3.7.2 Tapt signal-sjekk og nødlanding.....	13
3.7.3 Regulering av høyde.....	13
4 Alternative designvalg.....	14
4.1 Benytte hardware-interrupts til lesing av radio-input.....	14
Referanser.....	15

Dokumenthistorie

Versjon	Beskrivelse	Når
1.0	Første verifiserte versjon	2011-04-08
2.0	Gjort endringer i henhold til mal samt korrigert en link	2011-05-27

Tabell 1: Dokumenthistorie

Innledning

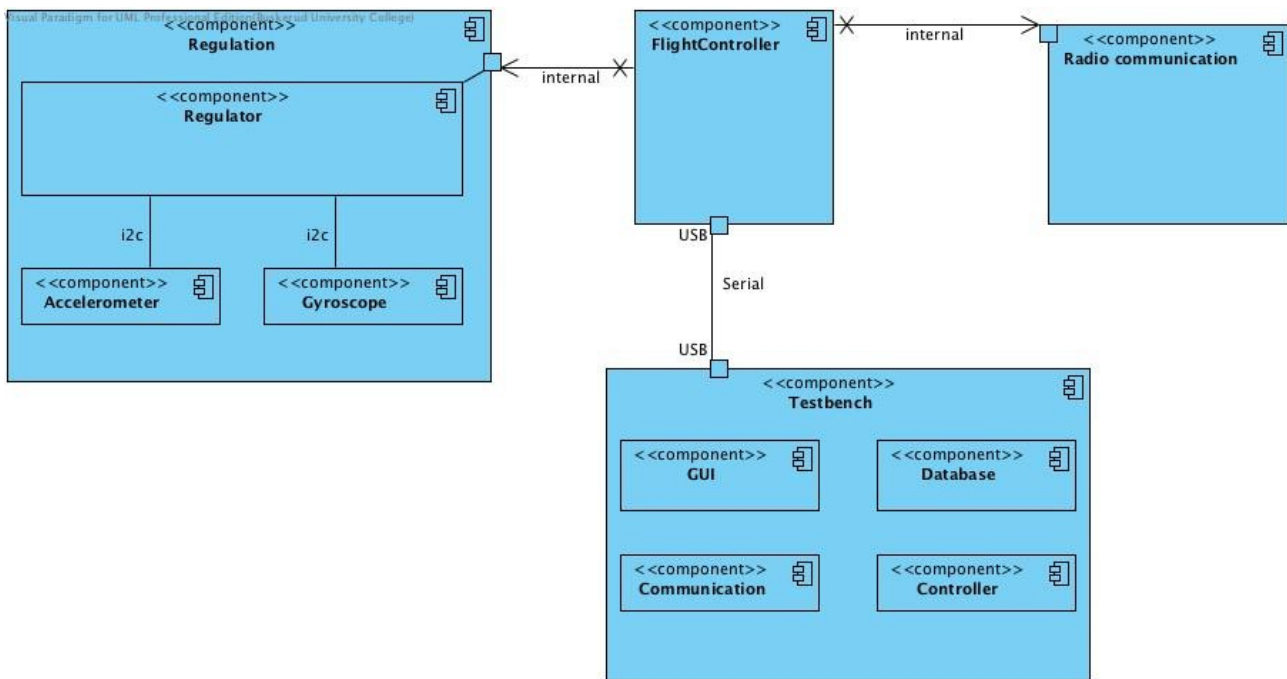
Et veltenkt og veldefinert design av programvaren er viktig for å tidlig avdekke feil, samt fremme nøyaktig tenking rundt valg av arkitektur og programflyt.

Vi har valgt å bruke UML [1] for å analysere, designe og - så langt det går - implementere programvaren til vårt fartøy og våre støttesystemer. Du vil her finne deployment-, komponent-, use case-, klasse- og sekvensdiagrammer for de tilfellene som er relevante i forhold til det regulerte fartøyet. I et sanntidssystem som dette kan det også være nyttig å ha med tilstandsdiagram, men det viste seg gjennom designet av denne fasen at det så langt ikke var nødvendig da de diskrete tilstandene var meget enkle, og de kontinuerlige ikke modellerbare på en måte som ville forsvart tidsbruken.

1 Overordnede diagram

1.1 Komponentdiagram

Dette diagrammet viser en overordnet representasjon av systemet og dets selvstendige deler slik vi har identifisert dette. Kapittelet om "Funksjonalitet" vil gå nærmere inn på hvordan dette fungerer sammen og er her kun ment for å gi et visst overblikk over systemet.



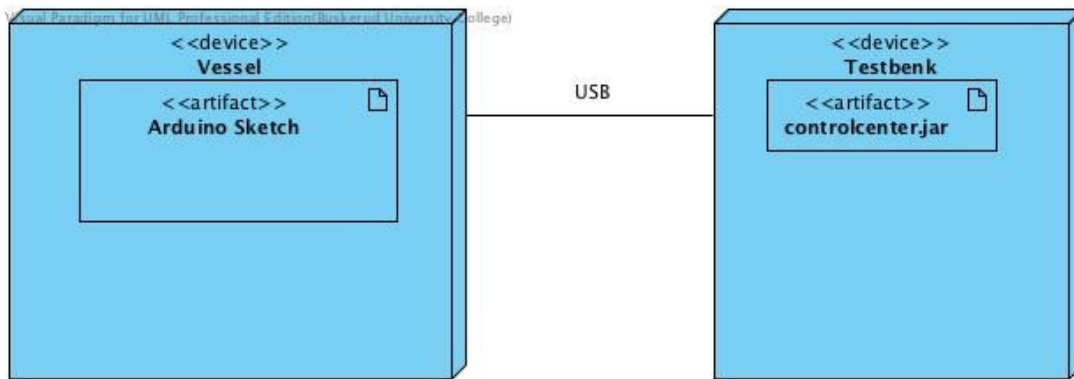
Figur 1: Overordnet komponentdiagram

Årseken til denne oppdelingen på fartøyet - jeg går ikke inn på testbenken i dette dokumentet - ligger i at vi har identifisert tre forsåvidt selvstendige deler som kan være ønskelig å behandle selvstendig. F.eks ligger hovedforskjellen mellom det regulerte og det uregulerte systemet ligger i at "Regulation"-komponenten i det uregulerte systemet ikke gjør noen innvirkning på systemet. Når det kommer til radio-kommunikasjonen kan det være hensiktsmessig å skille den i en selvstendig komponent som håndterer sine egne lese- og filtrerings-rutiner upåvirket av resten av systemet.

Kjernen av systemet ligger da i "FlightController"-komponenten som har ansvaret med å fordele oppgavene mellom de ulike komponentene, kombinere resultatene og til sist påvirke motorene med pådrag.

1.2 Deploymentdiagram

Vi har i utgangspunktet to steder hvor våre programmer vil kjøre. Dette er da fartøyet, her identifisert ved "Vessel", og datamaskinen som skal kjøre testbenk-programvaren.



Figur 2: Deploymentdiagram

2 Teknologi

2.1 Relevante teknologier

Vi har valgt å benytte oss av en Arduino Mega 2560[4] mikrokontroller levert av Arduino[4]. Denne kjører binære Arduino "Sketcher" som vi kan kompilere til fra C++ og laste opp via USB.

Arduino har et eget Java-basert IDE som lar oss på en enkel måte redigere filer, kompilere og laste opp til kontroller samt opprette en seriell kommunikasjonskanal over USB.

2.2 Valg, begrensninger og gitte arkitekturvalg

Ved å benytte ovennevnte mikrokontroller er vi også pålagt et par krav til hvordan vi kjører programmet, disse vil vi begrunne i de følgende del-kapitlene.

2.2.1 setup()

Mikrokontrolleren vil etter initiell initiering først kjøre *funksjonen* "void setup()", som vi må opprette. Denne kjøres én gang og er det naturlige stedet å legge kode som initierer vårt fartøy.

Alle våre kontroll-klasser har derfor en tilsvarende setup-metode som har som oppgave å ta seg av oppsett som er avhengig av at mikrokontrolleren er ferdig initialisert. Dette til forskjell fra konstruktorene som kjøres øyeblikkelig ved objekt-opprettelse og hvor vi da ikke kan - når vi har globale objekter - benytte oss av flere av Arduinos standard funksjoner og føre til at systemet kompilerer men ikke starter.

2.2.2 loop()

Etter at setup() har blitt kjørt vil mikrokontrolleren kjøre funksjoner "void loop()" - som også må opprettes av oss - så ofte som mulig. Dette blir da stedet hvor vi kjører all repeterende kode slik som lesing av radio-signaler og sensor-verdier samt sender hastigheter til motorene.

2.2.3 ESC

Til å kontrollere motorene benytter vi ESCer (Electronic Speed Controller) som tar pulsbredder til input og kontrollerer hver sin respektive motor. For hva programvaren angår er dette alt vi trenger å vite.

Lovlige verdier for pulsbreddene er fra ca 960-2000 μ s, men siden motorene ikke reagerer før tidligst 1030 μ s har vi valgt å definere pulsbredde-intervallene til å være fra 1000-2000 μ s da dette gir oss en bredde på 1000 μ s som igjen er veldig enkelt å jobbe videre med - både programvaremessig og manuelt.

Arduino har en Servo-klasse som opererer med utsending av pulsbredder og vi har valgt å benytte denne da den tillater oss å sende mikrosekunder innen gitte rammer og virker å være effektiv nok.

3 Funksjonalitet

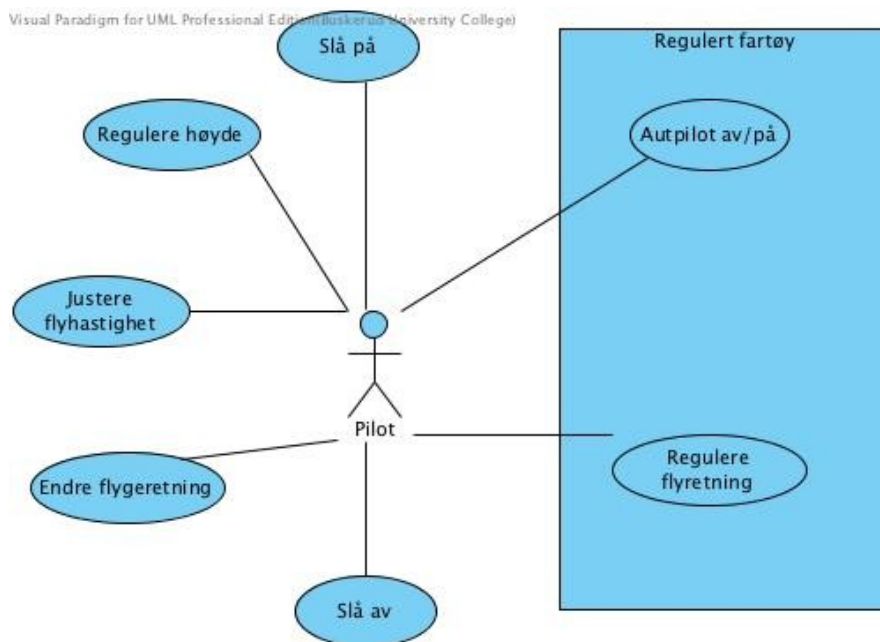
Vi har designet et sett med diagrammer hvilke har som mål å oppsummere kravene kunden og gruppen stiller til fartøyet og hvordan vi har sett for oss å løse disse. Dette dokumentet fokuserer på fartøyet men vi har også valgt å inkludere noe av den overordnede funksjonaliteten til testbenken da dette også angår nettopp fartøyet.

Se eget dokument for testbenken [2].

3.1 Use Case diagrammer

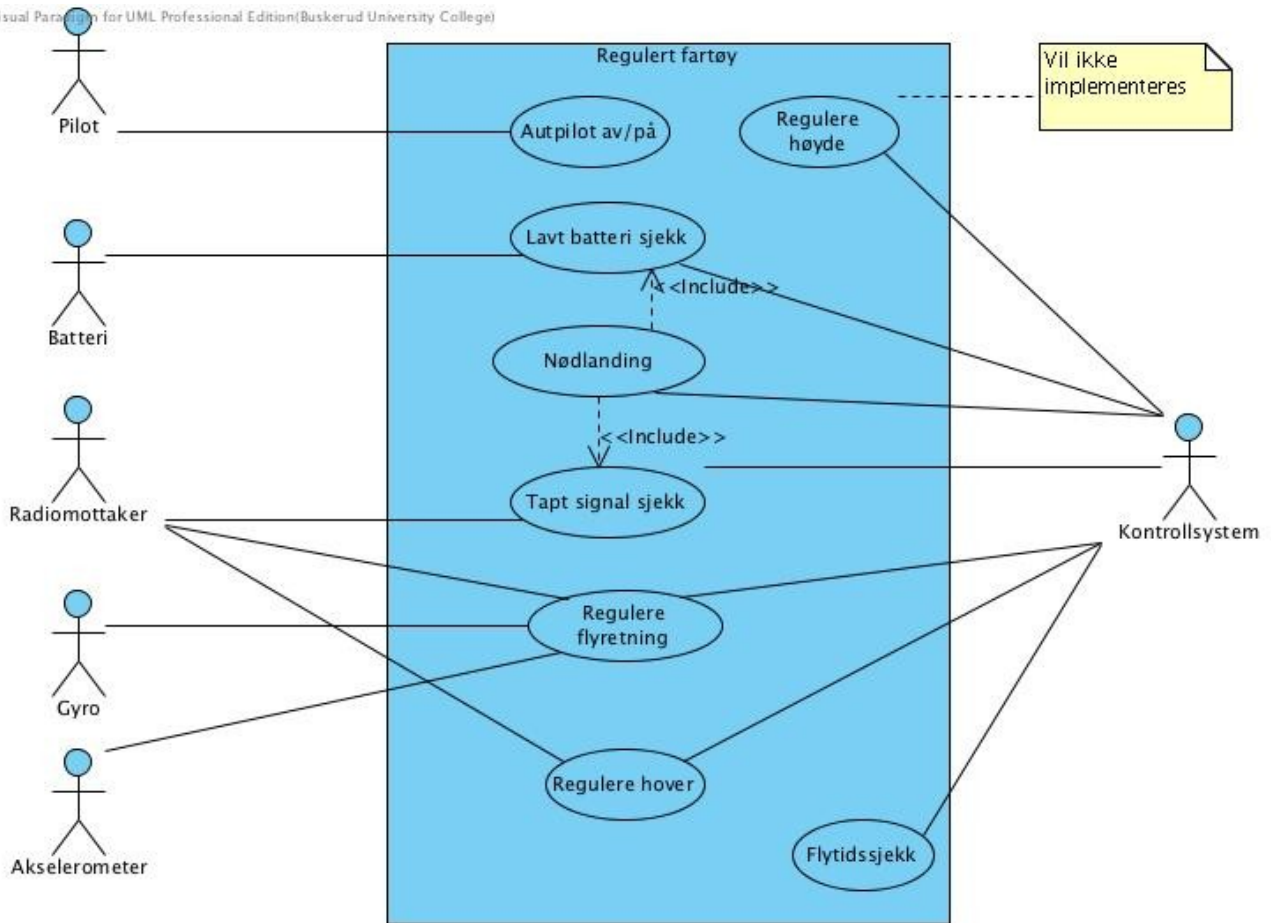
Med hensyn til oversiktighet har vi delt opp use case-diagrammene for det regulerte systemet i to. Det første tar for seg piloten (vedkommende som fører fartøyet og radiokontrolleren) og dennes påvirkninger på systemet mens neste er hvordan dette ser ut fra det regulerte fartøyets side.

Use case-diagrammene har som hensikt å visualisere på en oversiktig måte de ulike kravene / brukstilfellene det er satt til fartøyet.



Figur 3: Use case-digram som sett fra pilotens side

Visual Paradigm for UML Professional Edition (Buskerud University College)



Figur 4: Use case-digram som sett fra fartøyets side

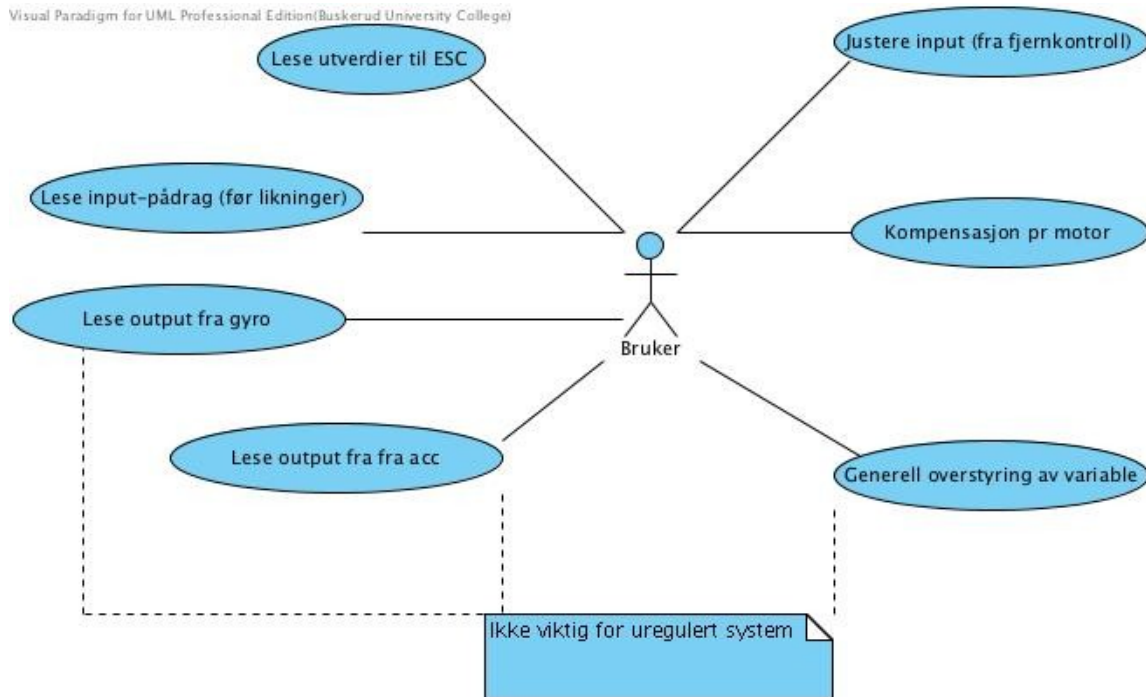
3.1.1 Det regulerte fartøyet

Vi har identifisert et sett med aktører som påvirker vårt system:

Aktør	Beskrivelse	Relaterte use-caser
Pilot	Dette er personen som kontrollerer fartøyet via en egen radio-kontroller, samt radio-kontrolleren selv da dette - hva systemet angår - er èn enhet.	Slå på, Slå av, Endre flygeretning, Justere flygehastighet, Regulere høyde, Autopilot av/på
Kontrollsystem	Initiativtakeren på fartøyet. Ingen egentlig aktør men representerer gjerne operativsystemet på mikrokontrolleren.	Reguler flyretning, Regulere høyde, Lavt batteri sjekk, Autopilot av/på, Nødlanding, Tapt signal sjekk, Regulere hover, Flytidssjekk
Gyroskop	Sensor, 3-akset gyroskop, i vårt tilfelle en ITG3200.	Reguler hover, Regulere flyretning
Akselerometer	Sensor, 3-akset akselerometer, i vårt tilfelle en BMA180.	Reguler hover, Regulere flyretning
Batteri	Selve batteriet og den fysiske batteri-alarman vi har koblet til denne.	Lavt batteri sjekk
Radiomottaker	Radio-mottaker på fartøyet, tar imot input fra pilotens radio-kontroller og sender dette som pulsbredder til mikrokontrolleren. 6 kanaler.	Tapt signal sjekk, Regulere hover, Regulere flyretning

Tabell 2: Aktører som påvirker systemet

3.1.2 Testbenk



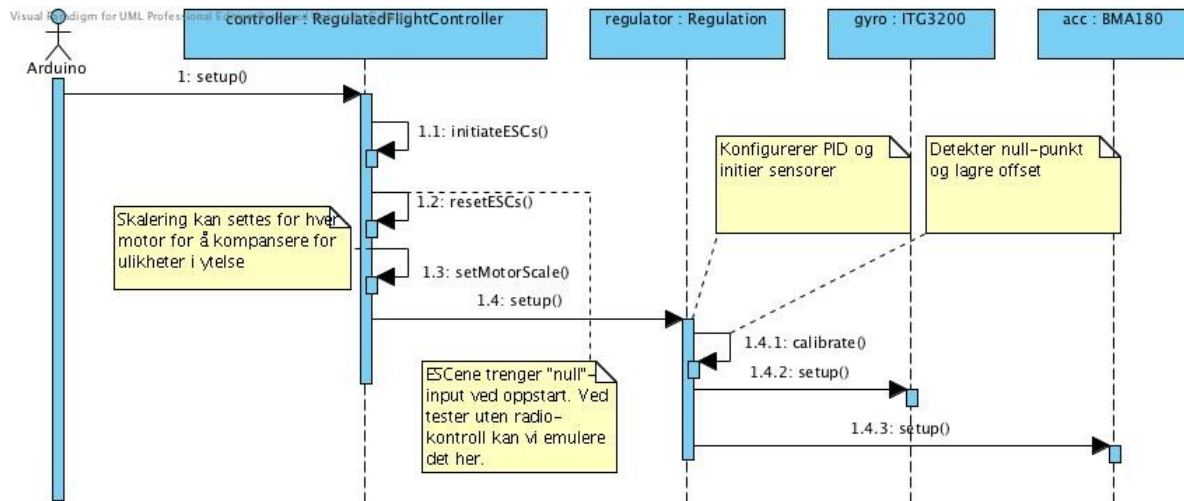
Figur 5: Use case-digram for testbenken

Se eget dokument for testbenken [2], den vil ikke bli diskutert noe mer i dette dokumentet da det er en selvstendig enhet av systemet.

3.2 Oppstart

Ved oppstart av fartøyet er det en viss mengde arbeid som må gjøres:

- Opprette nødvendige klasser og koblinger mellom disse
- Gjøre nødvendige kalibreringer og nullstillinger mtp sensorer, ESCer og radio



Figur 6: Oppstartsrutine

3.3 Avslutning

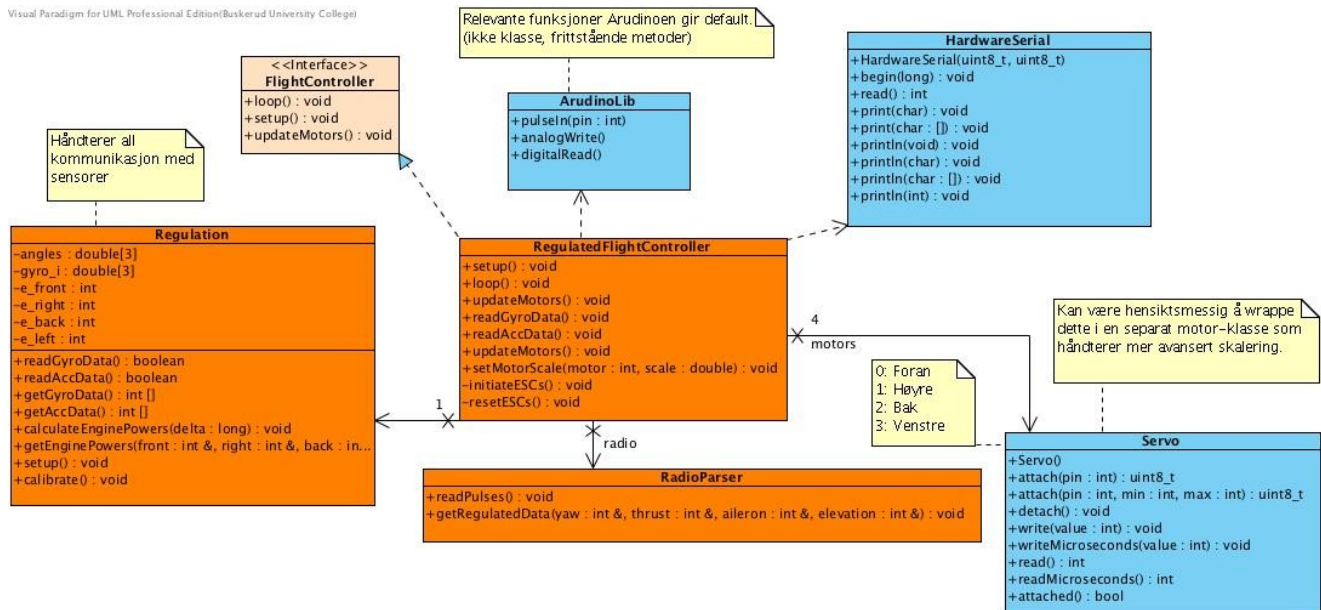
Vi avslutter / skrur av fartøyet ved å trekke ut strømmen og det er derfor valgt å ikke knytte en programvare-rutine til dette. Det er heller ikke avdekket noen behov ved dette tidspunktet angående dette.

3.4 Autopilot av/på

Det er et ønske om å kunne deaktivere reguleringssystemet, "Autopiloten", på en enkel måte f.eks hvis en sensor skal vise seg å sende ugyldige data under flyvning og på den måten ha en viss mulighet til å redde fartøyet og begrense skadeomfang.

Dette vil kun basere seg på lesing av et radio-signal og å initiere regulert eller uregulert rutine i FlightController.

3.5 Flyvning

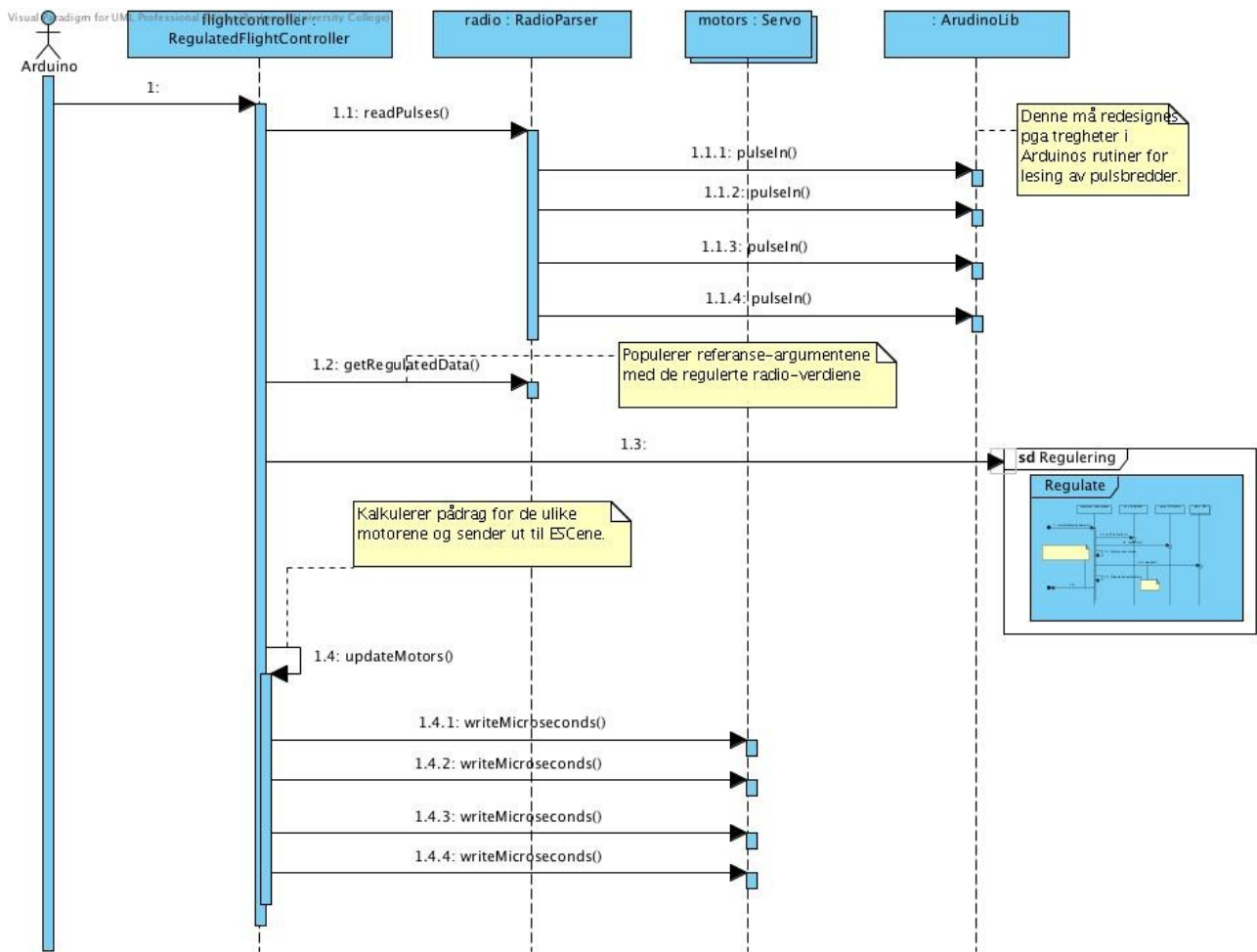


Figur 7: Klassediagram for regulert flyvning

Siden det vil være vanskelig og/eller i beste fall ugunstig å beskrive de ulike bevegelsene separat har vi valgt å putte alt som har med retning, høyde og hastighet å gjøre i en større beskrivelse, "Flyvning".

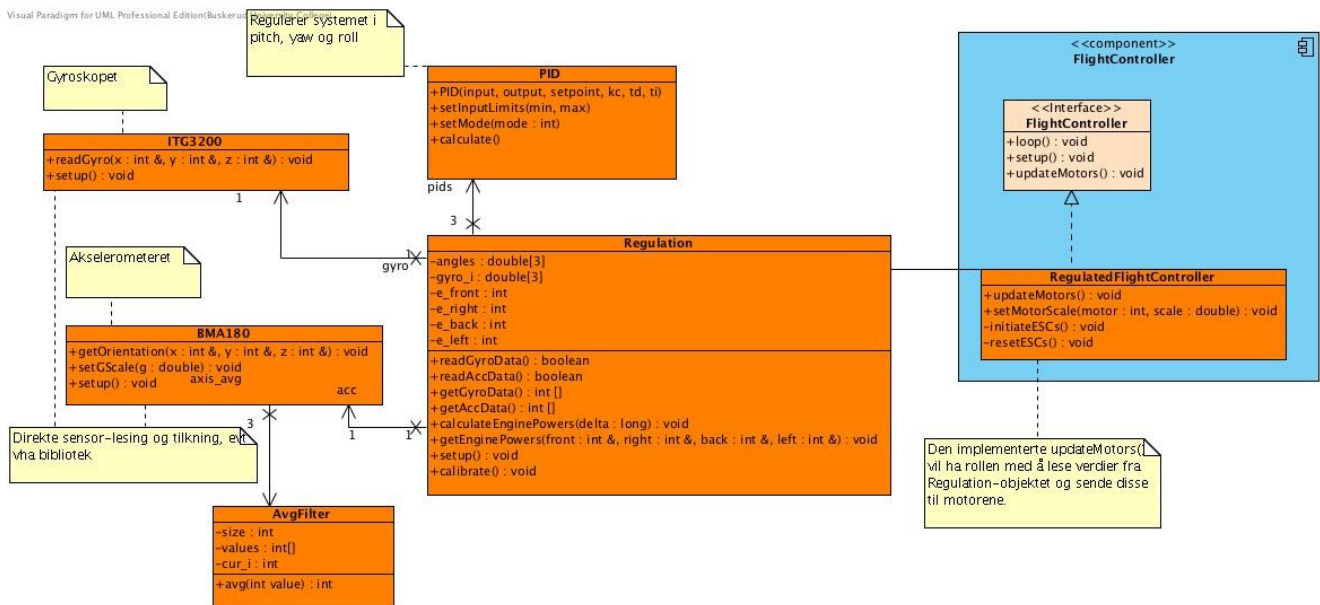
Dette er naturligvis den mest omfattende delen av systemet vårt, og inkluderer et sett med mindre arbeidsoppgaver:

1. Lesing fra radio
2. Lesing fra sensorer
3. Filtrering av signaler
4. Automatisert regulering og sending av verdier til ESCene.



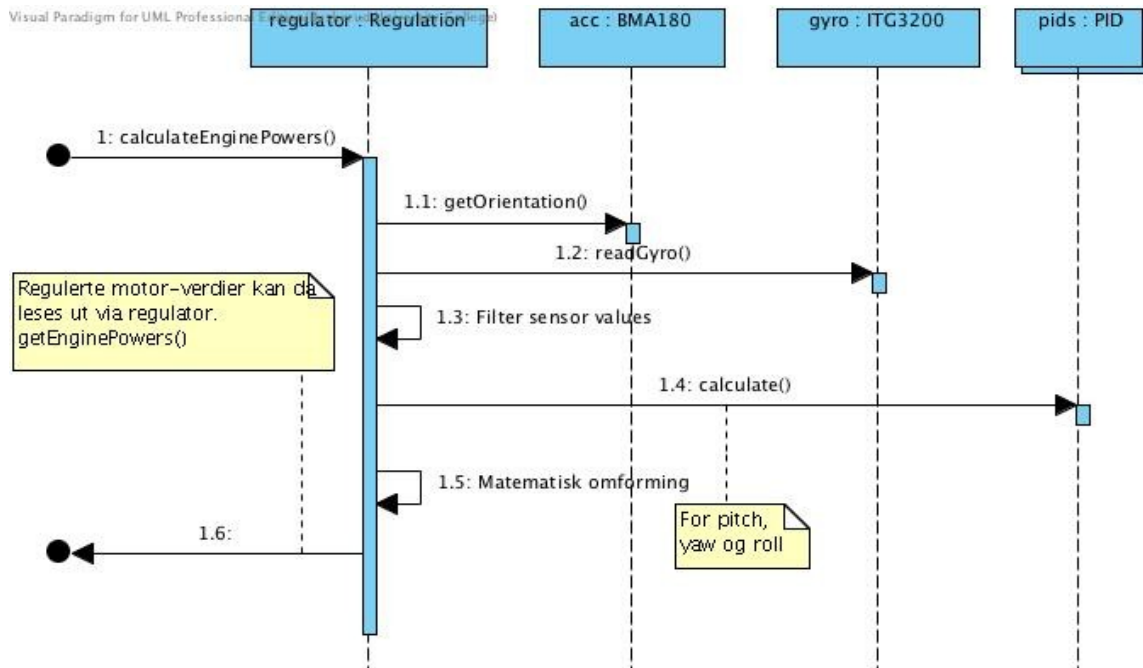
Figur 8: Rutine for lesing av input og regulering

3.6 Regulering



Figur 9: Klassediagram for reguleringsystemet

Reguleringsystemet skal ta hensyn til tyngdefeltet som lest fra akselerometeret og vinkelhastigheter lest fra gyroskopet. Disse vil filtreres mot hverandre med et komplementærfilter [3] som vektlegger dataene som kommer fra disse ulikt. Dette fordi akselerometerverdiene blir feilaktige når fartøyet akselererer, samt at de integrerte vinkelhastighetene opplever en viss "drift" pga unøyaktigheter vi ikke klarer å eliminere.



Figur 10: Sekvensdiagram for reguleringsystemet

3.7 Nedprioriterte krav / funksjoner

På grunn av oppgavens omfang og de mange mulighetene som ligger rundt disse har vi valgt i første omgang å prioritere et fartøy som flyr så stabilt som mulig ved hjelp av et robust reguleringsystem for deretter å fokusere på ekstra-funksjoner etter dette er oppnådd. Her følger en liten oversikt over hvilke funksjoner vi ikke har prioritert i denne omgang.

3.7.1 Batterisjekk

Dette er en nedprioritert funksjon som vil bli dekket i en annen iterasjon, men har likevel valgt å dekke et foreløpig utkast av denne funksjonaliteten på dette stedet da den anses som en del av det regulerte systemet.

3.7.2 Tapt signal-sjekk og nødlanding

Dette er også funksjonalitet som er nedprioritert inntil videre og som eventuelt vil håndteres i siste design/implementasjons-iterasjoner. Arkitekturen til programvaren er ellers lagt opp med med hensyn på at det ikke skal aktivt hindre dette på noen måte.

3.7.3 Regulering av høyde

Høyderegulering har vi ansett som et "kjekt og ha hvis tid"-krav og vil ikke bli prioritert i denne oppgaven men er nevnt i dette dokumentet for tydelighetens skyld.

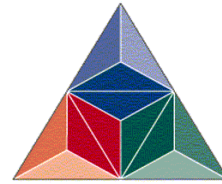
4 Alternative designvalg

4.1 Benytte hardware-interrupts til lesing av radio-input

Det kan vise seg at å benytte Arduinos standard-rutiner for lesing av pulsbredder ikke er effektiv nok i et slikt sanntidssystem som er såpass avhengig av hurtig tilpasning til respons. Alternativt kan vi her legge opp våre egne rutiner mot mikrokontrollerens interrupts for analog input. Dette krever noe ekstra tid, men kan være påkrevd.

Referanser

- [1] UML infrastructure, <http://www.omg.org/spec/UML/2.0/Infrastructure/PDF/> (2011-04-08)
- [2] Designdokument, testbenk. testbenk_1.0.pdf (nå designdokument_testbenk_3.0.pdf)
- [3] Dokument om fusjon av sensordata. sensorfusjon_1.0.pdf
- [4] Arduino webside, <http://arduino.cc/> (2011-05-27)



HØGSKOLEN
i Buskerud

Oppgavetittel: Hovedprosjekt – Implementasjon av regulert programvare – v1.0		
Fag(nr./navn): SFHO 3200 / Prosjektstyring med hovedprosjekt		
Gruppemedlemmer, signatur:		
_____ Vegard Torkelsen	_____ Michael Odden	_____ Herå Rørvik
_____ Trine Lindberg	_____ Luke Carambot	_____ Patrick Bjørum
Intern veileder: Dag Samuelsen Intern sensor: Olaf Hallan Graven Ekstern veileder og sensor: Ragnvald Otterlei		
Dato: 2011-05-27		

Innholdsfortegnelse

Dokumenthistorie.....	1
Innledning.....	1
1 Overordnede diagram.....	2
1.1 Komponentdiagram og klassediagram.....	2
1.2 Aktiviteter.....	3
1.2.1 Seriell kommunikasjon.....	3
1.2.1.1 Klassediagram.....	3
1.2.1.2 Sekvensdiagram.....	3
1.2.2 Reguleringsystemet.....	5
1.2.2.1 Klassediagram.....	5
1.2.2.2 Sekvensdiagram.....	6
1.2.3 Radioavlesning.....	7
1.2.4 Sekvensdiagram for håndtering av radio interrupt.....	7
1.2.4.1 Vurdering.....	8
1.2.5 Minnehåndtering.....	8
2 Platformspesifikke hensyn.....	9
2.1 Begrensninger.....	9
2.2 C++-operatorene "new" og "delete".....	9
3 Kode ikke utviklet av gruppen.....	10
3.1 Arduino biblioteksklasser.....	10
3.1.1 Servo.....	10
3.2 Kalman filter: Kalman.h.....	10
3.3 Avlesing fra gyroskop: ITG3200.cpp / .h.....	10
4 Konklusjoner.....	11
Referanser.....	12

Dokumenthistorie

Versjon	Beskrivelse	Når
1.0	Første ferdigstilte versjon.	2011-05-27

Tabell 1: Dokumenthistorie

Innledning

I overgangen fra et designet system til det er ferdig implementert vil det nødvendigvis være enkelte modifikasjoner av mindre sort som må til og i noen tilfeller redesign av enkeltkomponenter. Dette dokumentet vil omhandle vår programvare slik den fremstår i ferdig versjon.

Relasjoner til andre dokumenter

Dette dokumentet er å anse som et tillegg til designdokumentet for det regulerte fartøyet og vil hovedsaklig ta for seg de aspektene av programvaren som skiller seg fra det som er nevnt der.

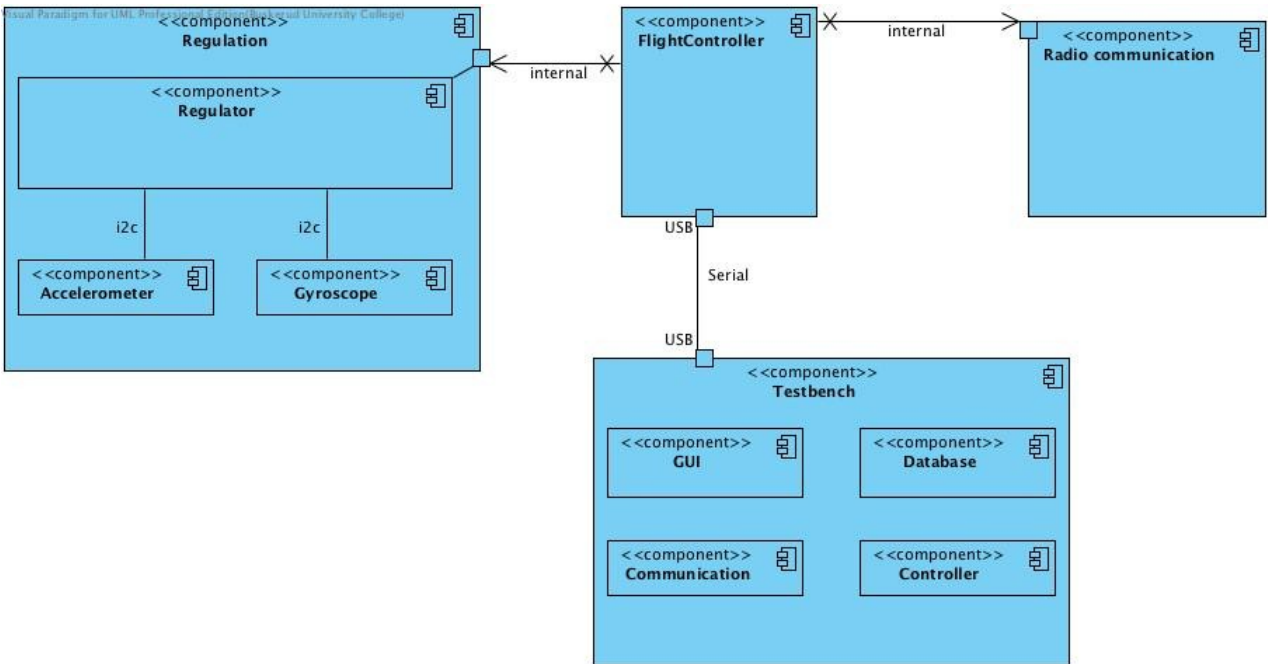
	Dokument	Kommentar
[1]	programvare-design_regulert_2.0.pdf	Gjeldende dokument er ment som en utvidelse av design dokumentet.
[2]	testbenk_1.0.pdf	Utdypende om testbenken og dens programvare
[5]	seriellprotokoll_1.0.pdf	Beskriver de rent formelle detaljene rundt protokollen for kommunikasjon mellom fartøy og testbenk.

Tabell 2: Relaterte dokumenter - nummerering er i henhold til referanser i slutten av dokumentet

1 Overordnede diagram

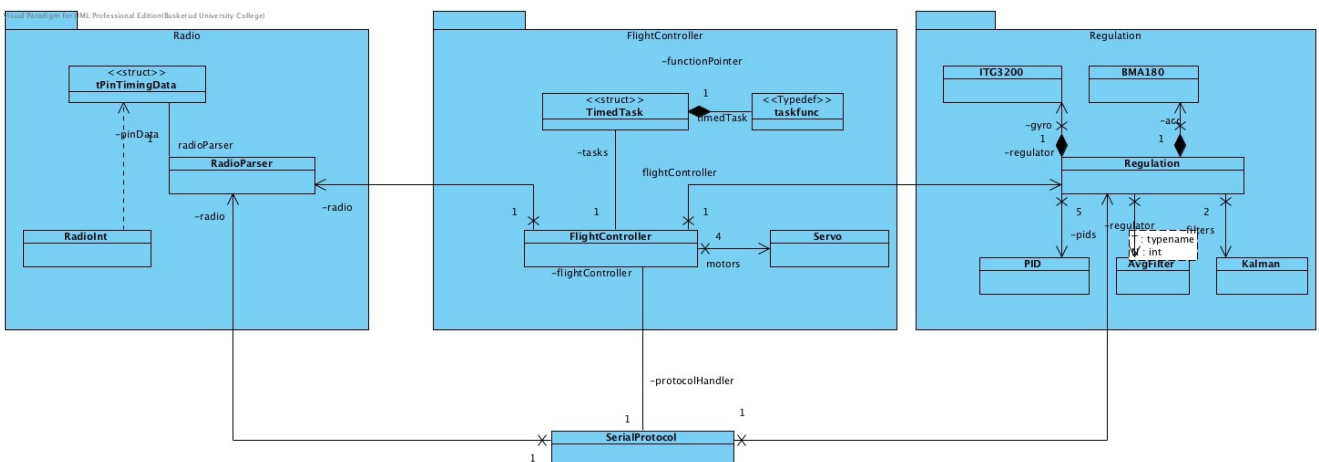
1.1 Komponentdiagram og klassediagram

Den overordnede strukturen av programmet er beholdt fra det opprinnelige designet.



Figur 1: Komponentdiagram for hele systemet

Dette dokumentet vil fokusere på programvaren slik den er på fartøyet. For informasjon om de testbenk-relaterte elementene refereres til testbenk-dokumentet [2].



Figur 2: Oversiktsbilde over klassene

1.2 Aktiviteter

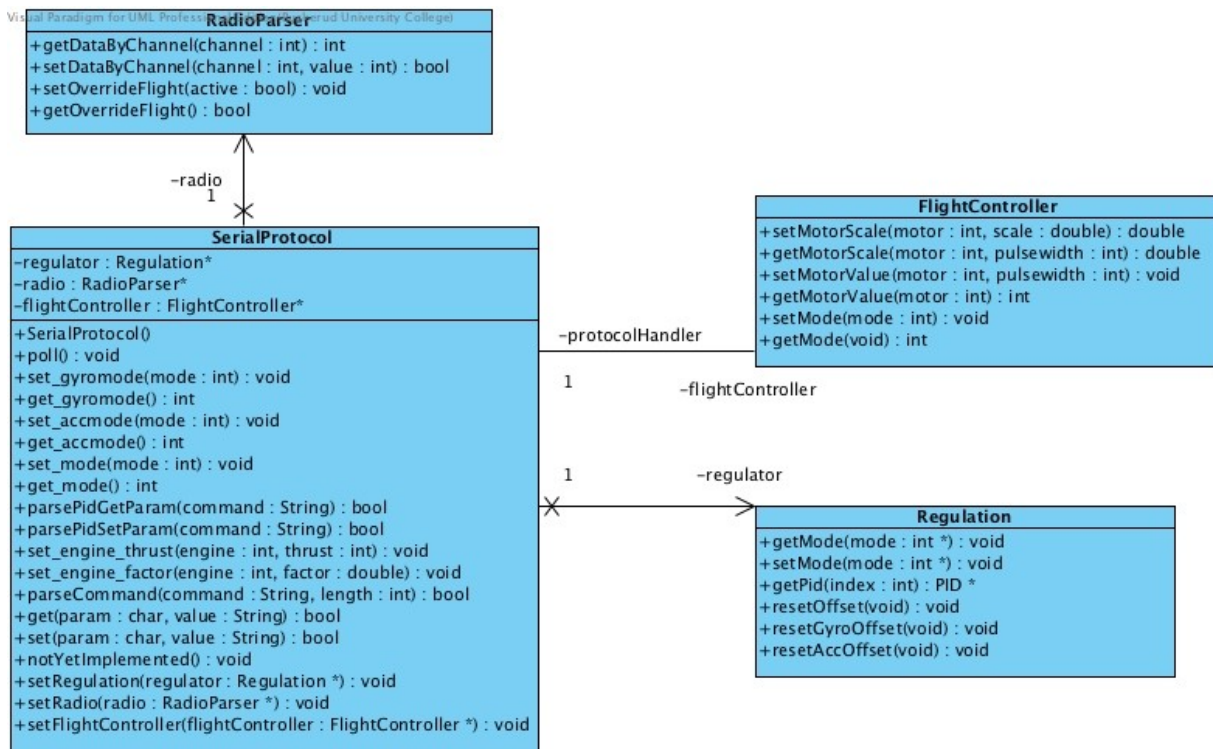
1.2.1 Seriell kommunikasjon

I det opprinnelige designet definerte vi støtten for seriell kommunikasjon og behandling av vår serielle protokoll som et grensesnitt ("interface") som ble realisert av FlightController-klassen. Dette skapte unødvendig komplisert kode og miksing av ansvarsområder da FlightController hovedsaklig skal være en tilrettelegger og et knutepunkt i programmet.

Løsningen ble å implementere dette som en separat klasse, SerialProtocol med en poll-metode som leser inn ventende data og deretter behandler denne.

Den serielle kommunikasjonen er nærmere beskrevet i testbenk-dokumentet [2] og dokumentet om seriell protokoll [5].

1.2.1.1 Klassediagram



Figur 3: Klassediagram for den serielle kommunikasjonen

1.2.1.2 Sekvensdiagram

FlightController::loop() vil med jevne mellomrom gå innom SerialProtocol::poll() for å sjekke om det er kommet data over den serielle forbindelsen. Hvis dette er tilfellet vil den splitte opp kommandoen i følgende deler:

1. Funksjon: set / get

2. Parameter

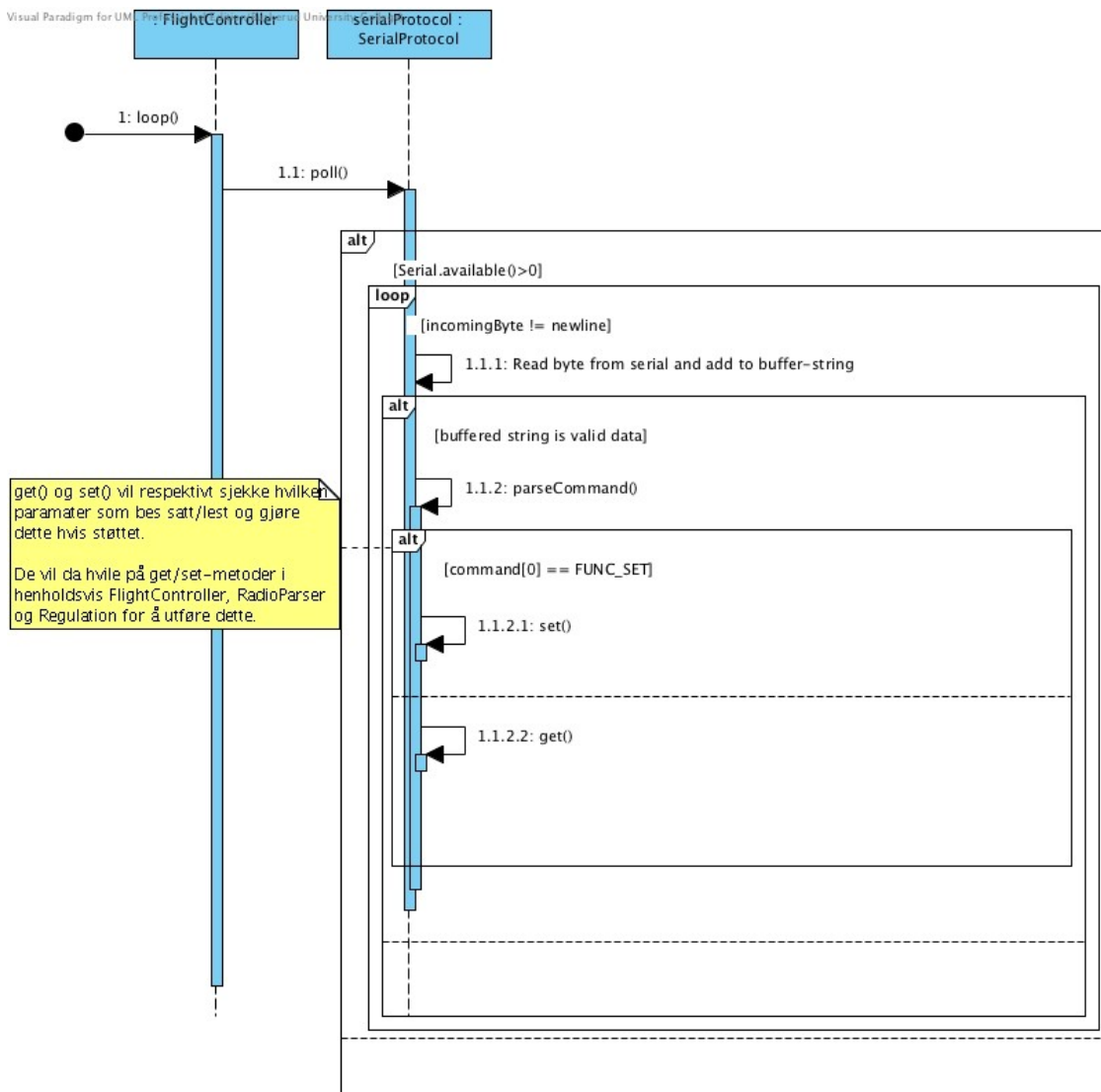
- Eksempelvis modus, radio-signaler, motor-pådrag, PID-parametere.

3. Verdi (kun hvis funksjon=set)

4. Deretter vil get/set-metodene til SerialProtocol respektivt håndtere sine henvendelser og dispatche det videre i programmet.

- Er det en get-kommando vil verdien det spørres etter hentes fra den relevante klassen og deretter skrives tilbake til testbenken via Serial.print().
- Er det en set-kommando vil den sende med verdien i riktig format til riktig set-metode i riktig klasse avhengig av hvilken parameter som skal endres.

Protokollen for seriell kommunikasjon mellom testbenk og fartøy står nærmere definert i [3] og [5].



Figur 4: Sekvensdiagram for håndtering av seriell kommunikasjon

1.2.2 Reguleringsystemet

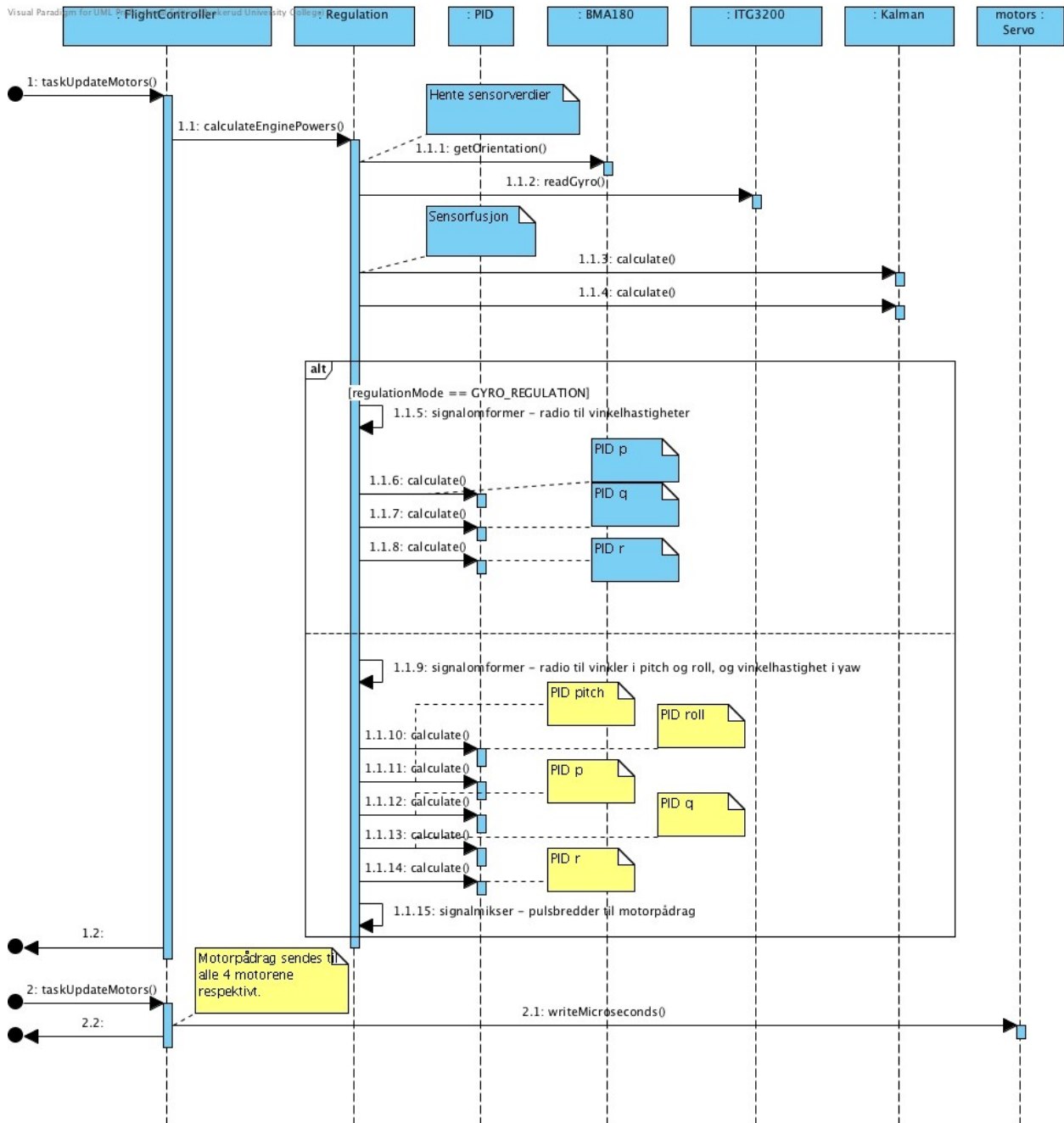
Reguleringsystemet var helt klart det området av programmet som har vært utsatt for mest justeringer underveis i implementasjonen. De mest vesentlige endringene her er knyttet mot filtrering av sensor-signaler og samkjøringen av de ulike PIDene.

1.2.2.1 Klassediagram



Figur 5: Klassediagram for reguleringsystemet

1.2.2.2 Sekvensdiagram



Figur 6: Sekvensdiagram for reguleringsystemet

Reguleringsystemet får innsendt radio-signaler via metodekallet, disse blir brukt som referansesignaler. Videre blir sensor-verdier hentet fra akselerometer og gyroskop og filtrert mot hverandre for mer nøyaktig gjengivelse av orientering. Deretter - avhengig om man er i gyro-regulert modus, eller vinkel-regulert modus - sendes disse signalene inn til et sett PIDs som over tid regulerer systemet ved å gi nødvendige pådrag i yaw, pitch og roll som til slutt blir kalkulert om til pådrag for de enkelte motorene.

1.2.3 Radioavlesning

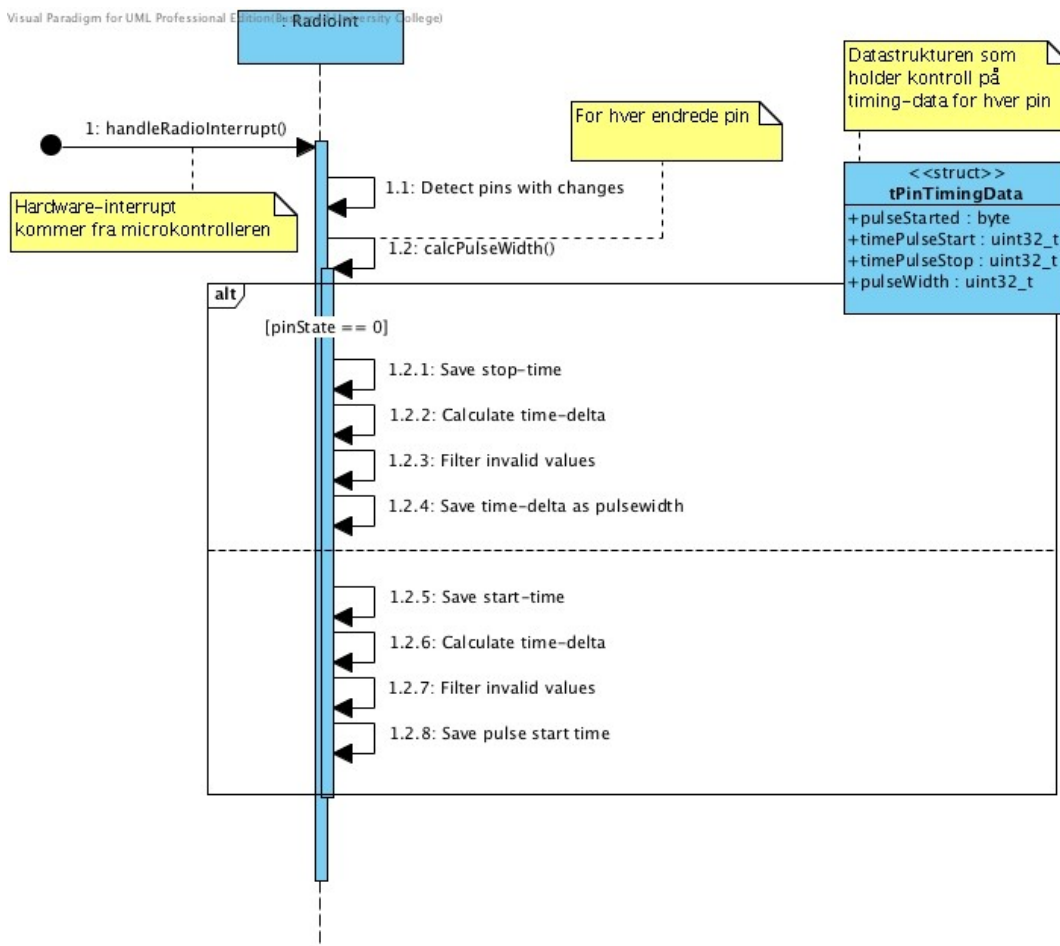
I det opprinnelige design-dokumentet la vi opp til å benytte Arduino's medfølgende "pulseIn"-funksjoner noe som viste seg å innføre betraktelige tregheter i systemet.

Løsningen på dette var å implementere en egen rutine som reagerer på hardware-avbrudd på de gitte portene og kalkulerte pulsbredder etterhvert som de ble motatt av systemet. På denne måten kan systemet ha verdiene klare når resten av programmet trenger disse.

Konseptet i seg selv er veldig enkelt og krevde ingen større design-endringer da hovedutfordringen lå i å finne ut de korrekte registerverdiene og -områdene for å aktivere og styre dette.

Dette er implementert i RadioInt-klassen som tilbyr tilgang til et array med disse verdiene via metoden RadioInt::getPinTimingData(void).

1.2.4 Sekvensdiagram for håndtering av radio interrupt



Figur 7: Sekvensdiagram for håndteringen av radio interrupt

Ved å implementere funksjonen "ISR(PCINT2_vect)" kan vi dirigere håndtering av interruptene til den statiske metoden RadioInt::handleRadioInterrupt(). De relevante registrene man må jobbe mot for Arduinos interrupt-rutiner står beskrevet i databladet [3] på sidene 113-117.

1.2.4.1 Vurdering

Ved å implementere denne formen for radioavlesning i forhold til Arduino's medfølgende pulseIn-funksjon opplevde vi en økning i programfrekvens fra 16Hz til 150Hz. Dette var en enorm forbedring og helt klart et nødvendig steg mot å gjøre dette til et mer flygbart fartøy da reguleringsystemet oftere kan gjøre tilpasninger på motor-pådragene.

1.2.5 Minnehåndtering

Da mikrokontrolleren er meget begrenset med tanke på minnestørrelse og CPU-hastighet har vi så langt det har vært mulig, praktisk og uten å innføre potensielle feilkilder basert oss på å sende pekere til de ulike variablene og objektene vi har i minnet. Dette innførte ingen markante design-endringer, men var heller ikke noe eksplisitt diskuterte i designfasen.

Minnebruken har blitt monitorert over tid og det har ikke dukket opp noen tegn på noen minnelekasjer.

2 Platformspesifikke hensyn

2.1 Begrensninger

Mikrokontrolleren vi benyttet var en Arduino Mega2560. Denne har en Atmega2560 CPU som er klokket til 16MHz og 256KB minne i tillegg til at den ikke hadde noen implementasjon for håndtering av tråder gav oss et sett med utfordringer knyttet mot ytelse og samtidighet.

Da de ulike elementene av programmet ikke trenger å kjøre i nødvendigvis hver eneste programsyklus implementerte vi en enkel task-håndtering.

Dette er gjort i `FlightController::loop()` og det er da skilt mellom følgende hovedaktiviteter/tasker:

- Lesing og håndtering av seriell data
- Lesing av radio-input
- Kjøring av reguleringssystemet og oppdatering av motorpådrag

Da seriell kommunikasjon er det minst viktige og kun relevant i test-sammenheng kjøres denne kun hver 10. syklus. Med utgangspunkt i at programmet kjører i omtrentlig 150Hz så vil dette si 15 ganger i sekundet.

Radio-input er langt mer viktig enn seriell input, men da dette skal styres av mennesker så er dette begrenset til menneskers observasjonsevne og rekasjonstid. Vi har derfor valgt å kjøre denne kun hver andre syklus, da tilsvarende 75 ganger i sekundet. Denne kan antakeligvis trygt reduseres til det halve av dette også men vi har ikke testet dette tilstrekkelig.

Kjøring av reguleringssystemet og oppdatering av motorpådrag er den viktigste oppgaven og kjøres hver eneste syklus slik at programvaren raskest mulig kan korrigere for feil mellom referanser og målinger.

2.2 C++-operatorene "new" og "delete"

Det viste seg at kompilatoren ikke tok med seg implementasjon av operatorene *new* og *delete*, noe som førte til kompilasjonssykninger da vi trengte å være sikre på at objekter vi hadde opprettet ikke ble dupliserte, samt at minneområdene hvor de lå ikke ble overskrevet. Vi hadde opprinnelig enkelte komplikasjoner knyttet til dette da vi sendte rundt referanser til objekter som var blitt lagt på stack og i enkelte tilfeller ble overskrevet når programmet gikk ut av "scopet" de ble opprettet i og dermed gav udefinert oppførsel.

Løsningen på dette ble å implementere operatorene "new" og "delete" respektivt - dette er gjort i filen *CppFix.h*.

3 Kode ikke utviklet av gruppen

3.1 Arduino biblioteksklasser

3.1.1 Servo

Vi benytter Servo-klasser for å kommunisere mot de respektive ESCene. Vi valgte å benytte disse av bekvemmelighet da det var kode som er godt testet fra Arduino's side, samt at gav oss enkel fleksibilitet mot saturering og valgfri sending av pulsbredder som millisekunder eller grader. Alternativt kunne vi benyttet Arduino's analogWrite-funksjon men var ingen øyeblikkelige fordeler og funksjonaliteten var allerede testet til å være tilfresstillende.

3.2 Kalman filter: Kalman.h

Basisen for kildekoden til vårt Kalman-filter var gjort tilgjengelig via hjemmesidene til X-Firm som i hovedsak er Patrik Olsson's blogg. Koden er distribuert både på hans blogg og ulike fora og anses som tilgjengelig basert på hans kommentarer samt det faktum at algoritmen for Kalman-filtre er fritt tilgjengelig..

Koden er hentet fra http://www.x-firm.com/?page_id=191 og endret for økt fleksibilitet i form av objektorientering samt at passe vår struktur

3.3 Avlesing fra gyroskop: ITG3200.cpp / .h

Koden er frigitt under lisensen "GNU LESSER GENERAL PUBLIC LICENSE" [4] som gir oss tillatelse til å redistribuere den videre i eget arbeid ved behold av lisensen.

Prosjektet er tilgjengelig på: <http://code.google.com/p/itg-3200driver/>

4 Konklusjoner

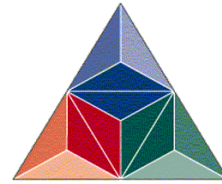
Det overordnede designet viste seg å i hovedsak stå sin test med unntak av håndteringen av radio-input. Problemene knyttet mot radio-input var noe vi ble klar over under enkelte prototyper i forbindelse med utarbeiding av av designet men som vi besluttet å håndtere senere da vi hadde et mer helhetlig system å teste mot.

Endringene som har blitt innført i reguleringsystemet henger sammens med oppdatering i de matematiske modellene og anses ikke som rene design-mangler men heller en naturlig evolusjon under en slik iterativ tilnærming.

Med den leverte koden (versjon 1.0) har vi et robust system som håndterer radio-input, seriell kommunikasjon mot testbenk, lesing av sensordata og et reguleringsystem som jobber for å rotere fartøyet etter de referansesignalene mottatt fra radio-mottakeren.

Referanser

- [1] Designdokument programvare regulert fartøy. programvare-design_regulert_2.0.pdf
- [2] Dokument om testbenk. testbenk_1.0.pdf
- [3] Datablad for Arduino Mega 2560,
http://www.atmel.com/dyn/resources/prod_documents/doc2549.PDF (2011-05-27)
- [4] GNU LGPL *<http://www.gnu.org/licenses/lgpl.html> (2011-05-27)*
- [5] Seriell protokoll dokument. seriellprotokoll_1.0.pdf



HØGSKOLEN
i Buskerud

Oppgavetittel: Hovedprosjekt - Foreningsdokument - v1.0		
Fag(nr./navn): SFHO 3200 / Prosjektstyring med hovedprosjekt		
Gruppemedlemmer, signatur:		
_____ Vegard Torkelsen	_____ Michael Odden	_____ Herå Rørvik
_____ Trine Lindberg	_____ Luke Carambot	_____ Patrick Bjørum
Intern veileder: Dag Samuelsen Intern sensor: Olaf Hallan Graven Ekstern veileder og sensor: Ragnvald Otterlei		
Dato: 2011-05-27		

Innholdsfortegnelse

Dokumenthistorie.....	1
Innledning.....	1
1 Komponenter, begreper og definisjoner.....	2
1.1 Komponenter.....	2
2 Kombinerte diagram.....	3
2.1 Struktur.....	3
2.1.1 Komponentdiagram.....	3
2.1.2 Klassediagram for programvaren.....	4
2.1.3 Kombinert diagram.....	5
2.2 Programflyt.....	6
2.2.1 Blokkdiagram for simulering.....	6
2.2.2 Sekvensdiagram for programvare.....	7
2.2.3 Kombinert diagram.....	8
3 Konklusjoner.....	10
Referanser.....	10

Dokumenthistorie

Versjon	Beskrivelse	Når
1.0	Første ferdigstilte dokument.	2011-05-27

Tabell 1: Dokumenthistorie

Innledning

Dette dokumentet har som mål å gi leseren en klar formening om sammenhengene mellom programvaren, elektronikken og simulerings-dokumentet.

Det opprinnelige målet var å forene disse i et felles diagram men fant dette til å bli for vanskelig å gjøre forståelig. Løsningen falt da heller på å gjøre dette med et sett diagrammer fra ulike nivåer og med en tekstlig beskrivelse.

Vi vil her fokusere på de komponentene og hendelsene som foregår i/på selve fartøyet. Testbenk og radio-kontroller er ikke prioritert her. Det vil heller ikke være fokus på hvordan oppgavene løses internt i de respektive områdene, men heller på sammenhengene mellom disse.

Relasjoner til andre dokumenter

Dette dokumentet forsøker å trekke litt sammenlikninger mellom følgende dokumenter, og er dermed også dokumentene å lese videre i for dypere forståelse om de respektive feltene.

	Dokument	Kommentar
[1]	programvare-implementasjon_1.0	Hvordan koden er implementert og endringer gjort fra designet.
[2]	programvare-design_regulert_2.0	Designet av programvaren.
[3]	simuleringsdokument_1.0	Om simuleringene gjort for å hjelpe til å finne parametere til det fysiske systemet.
[4]	komponentdokumentasjon_3.0	De ulike komponentene brukt i systemet.
[5]	koblingsskjema_1.0	Skjema som viser koblingene mellom de ulike elektriske komponentene i systemet.

Tabell 2: Relaterte dokumenter

1 Komponenter, begreper og definisjoner

1.1 Komponenter

Elektrisk komponent	Programvare	Simuleringsbegrep	Hensikt
BMA180, accelerometer	BMA180.cpp	Akselerometer	Gi oss målinger av tyngdefeltet nødvendig for å kunne kalkulere orientasjonen til et objekt i rommet.
ITG3200, gyroscope	ITG3200.cpp	Gyroskop	Gi oss målinger av vinkelhastighet - dvs hastigheten sensoren roterer rundt sin egen akse.
Futaba R617FS, receiver	RadioParser.cpp, RadioInt.cpp	Radio	Gi oss pulsbredden i samsvar med pilotens posisjoner av joystickene på radio-kontrollen.
Turnigy Plush - 18A (ESC)	Servo.cpp (Arduino biblioteksklasse)	Fremdriftssystem (ESCene pluss motorene de styrer)	ESCene tar imot pulsbredden fra mikrokontrolleren / programvaren og regulerer hastighetene på sine respektive motorer utifra disse.

Tabell 3: Oversikt over komponenter, begreper og definisjoner

For mer informasjon om de ulike komponentene henvises det til komponentdokumentasjonen [4].

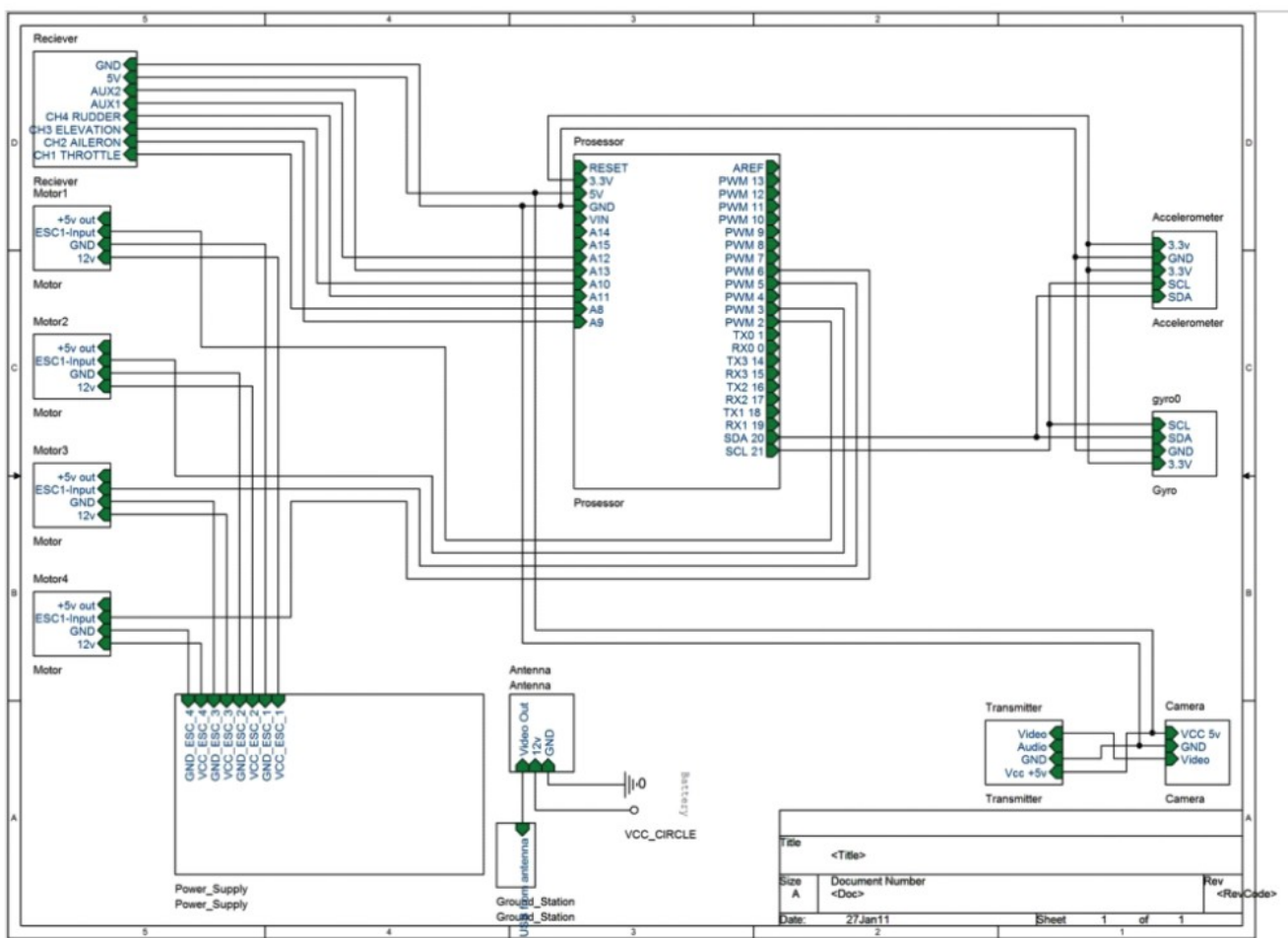
2 Kombinerte diagram

Det er i hovedsak to måter å se på systemet vårt på som vi vil fokusere på i dette dokumentet. Det er *struktur* og det er *programflyt*. Strukturen omhandler oppbygning og ansvarsfordeling, mens programflyten handler om hva som skjer hvor i hvilken rekkefølge for å løse de gitte oppgavene med de gitte ressursene.

2.1 Struktur

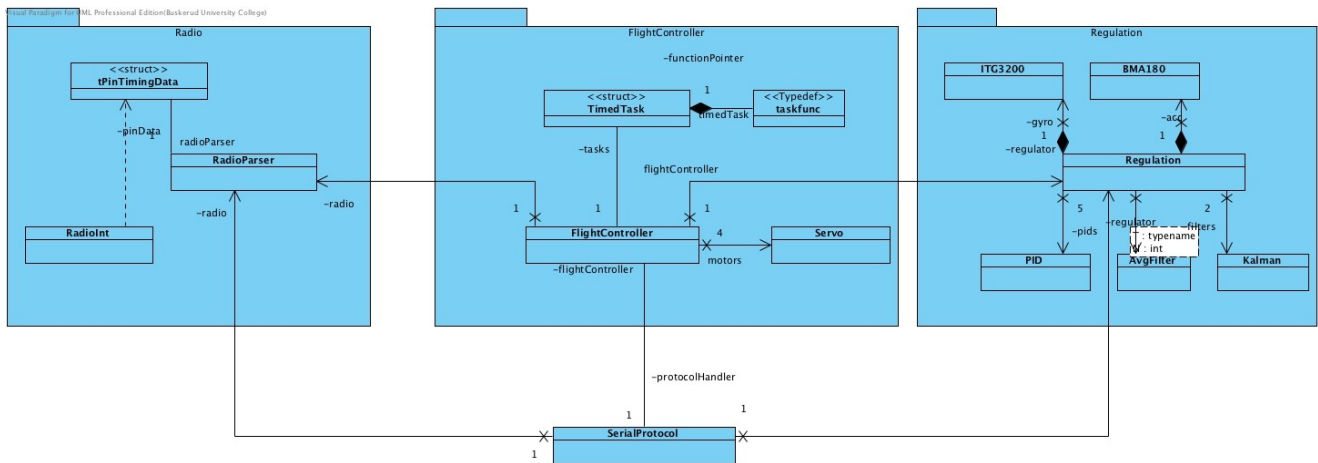
I ytterpunktene av programvaren har vi klasser som opererer som grensesnitt mot de elektriske komponentene vi ønsker å lese/skrive data fra/til.

2.1.1 Komponentdiagram



Figur 1: Komponentdiagram som viser kobling mellom de ulike komponentene [5]

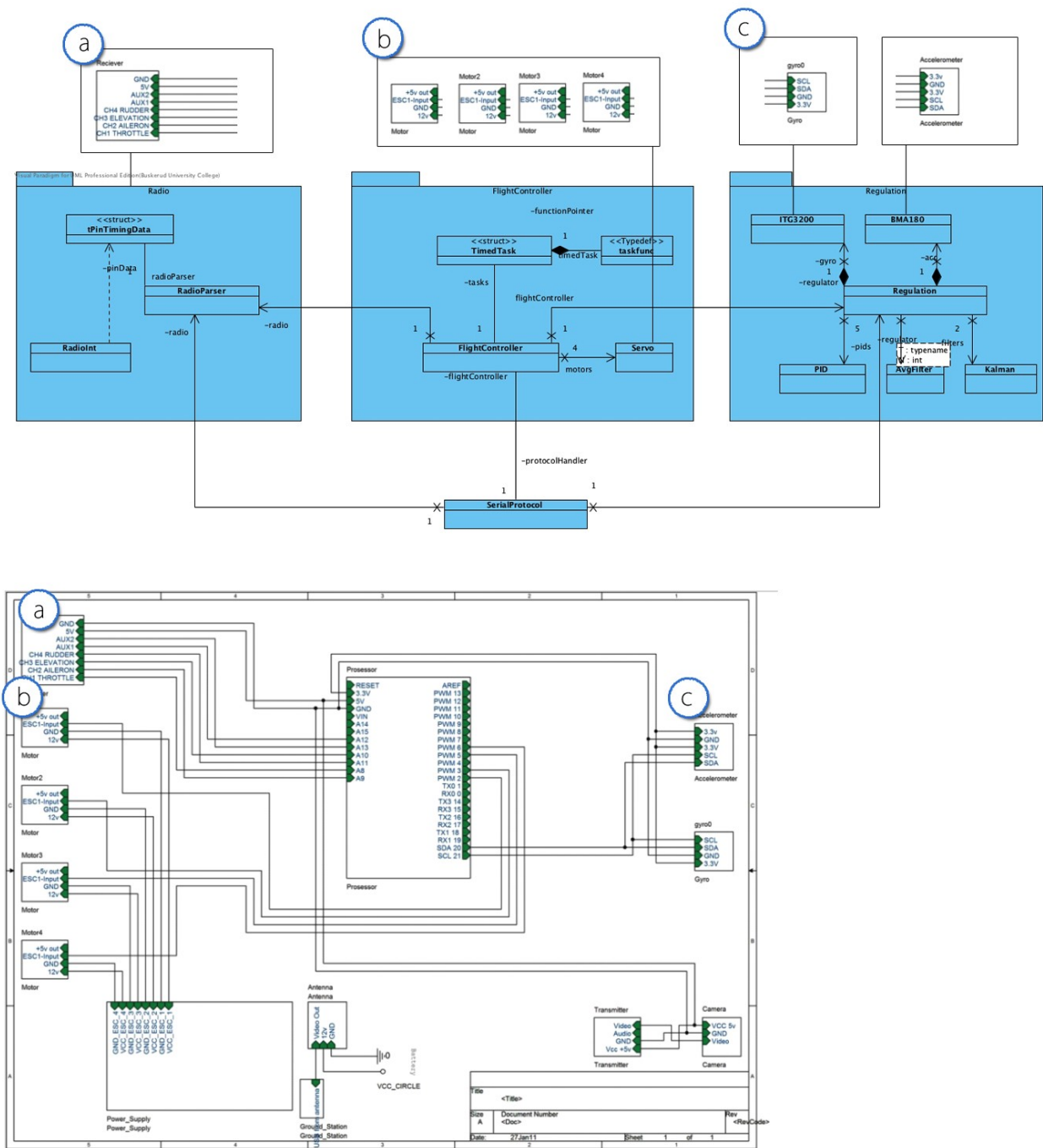
2.1.2 Klassediagram for programvaren



Figur 2: Forenklet pakke og klassediagram for fartøyets programvare

For mer detaljert beskrivelse om de ulike klassene og deres ansvarsområder henvises det til dokumentene for design [2] og implementasjon [1] av programvaren.

2.1.3 Kombinert diagram

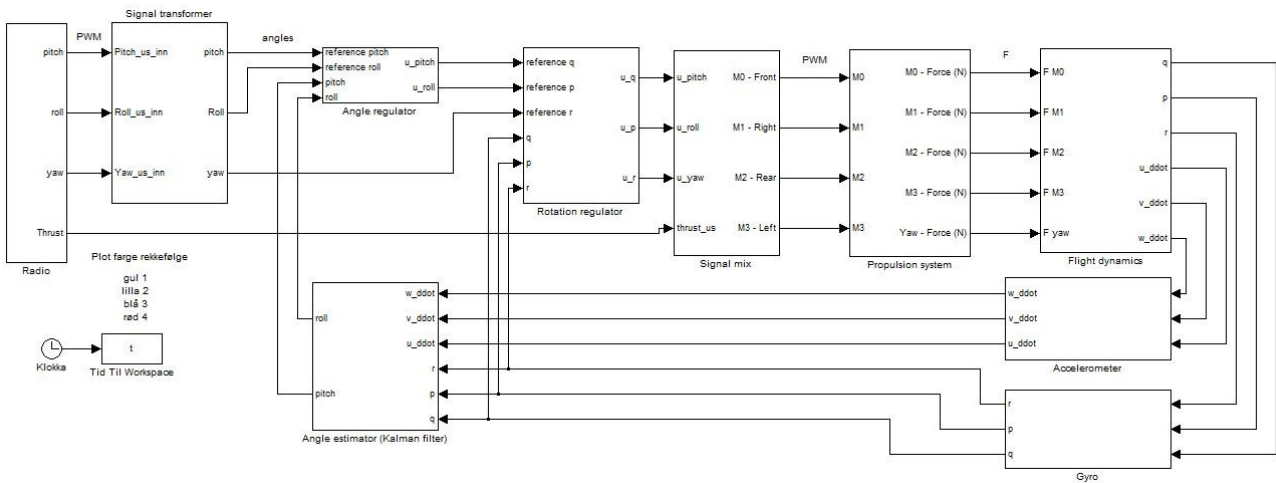


Figur 3: Kombinert diagram som viser relasjoner mellom komponentdiagrammet og programvarens klassediagram.

2.2 Programflyt

Med bakgrunn i sammenhengen mellom elektriske komponenter og programvare-klasser som beskrevet i 3.1 kan vi videre fokusere på sammenhengen mellom simulerings-diagrammet og programvarens sekvensdiagram.

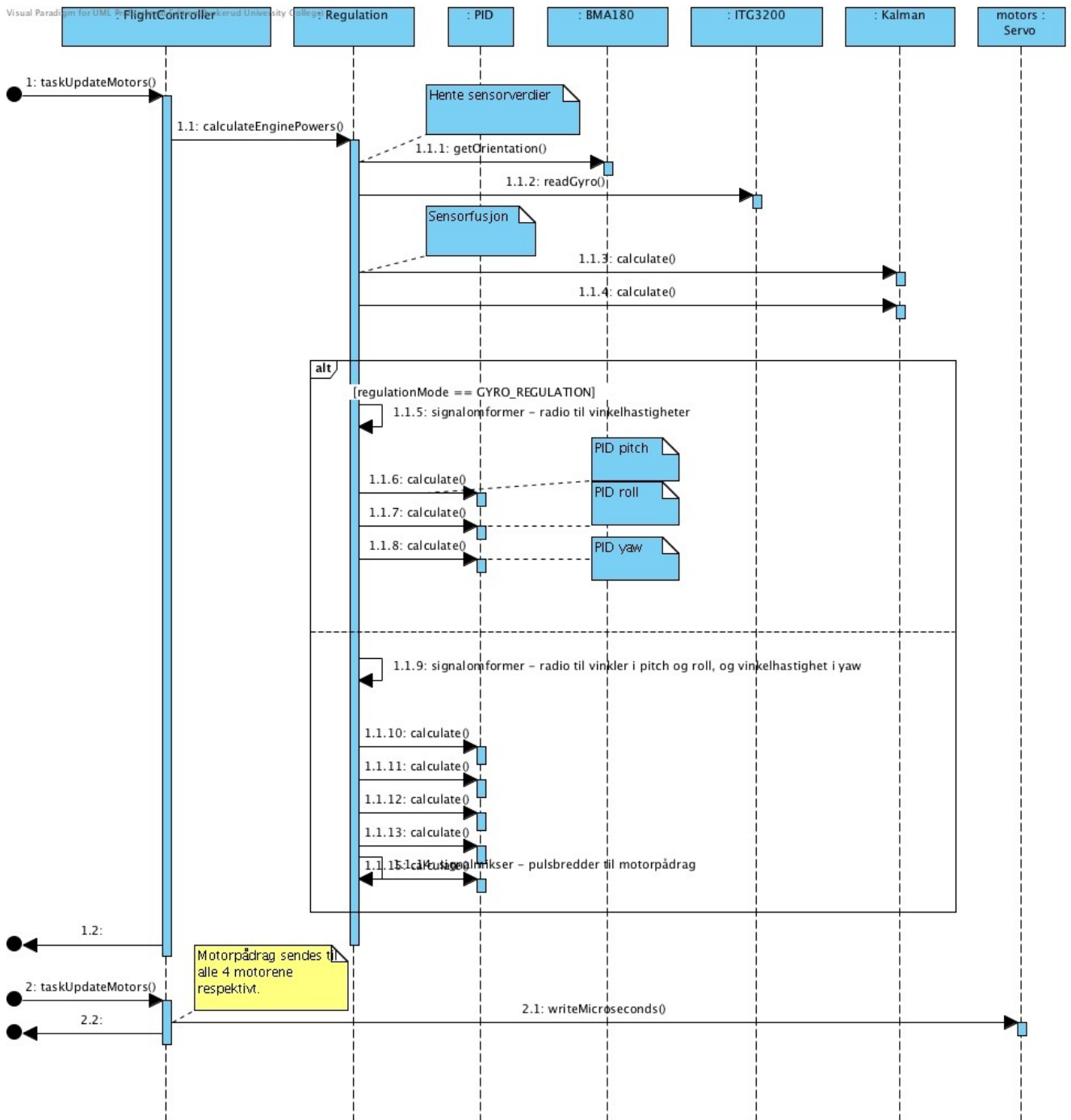
2.2.1 Blokkdiagram for simulering



Figur 4: Blokkdiagram for simulering

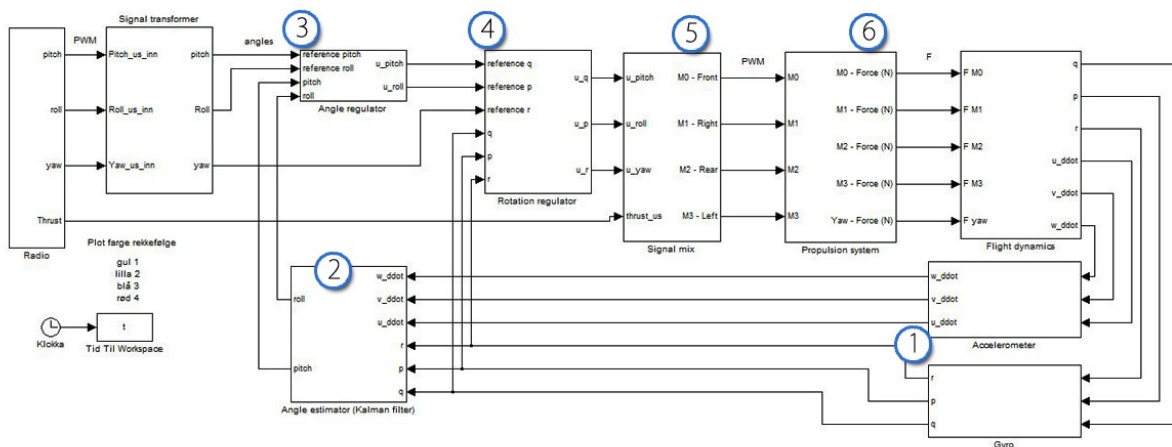
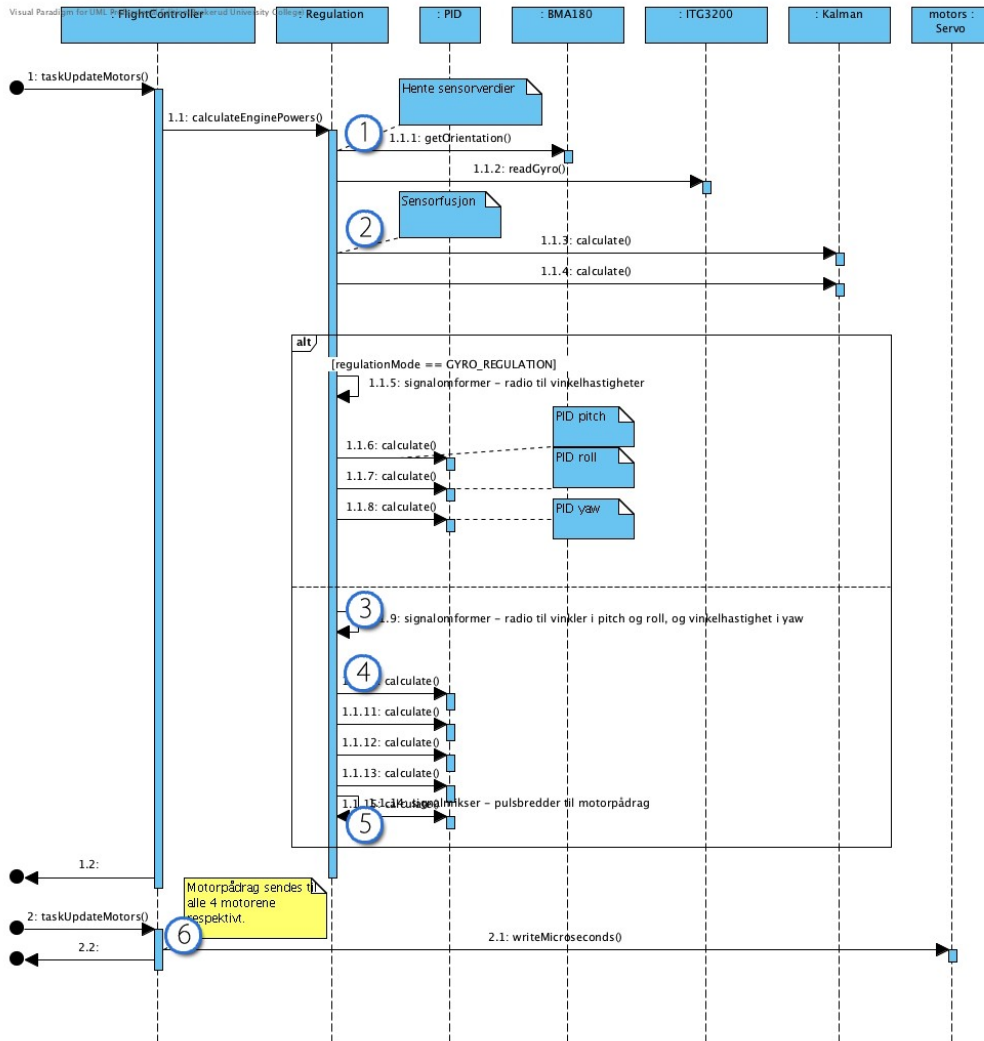
For utfyllende informasjon om simuleringene henvises til simuleringsdokumentet [3].

2.2.2 Sekvensdiagram for programvare



Figur 5: Sekvens for reguleringsystemet slik det er implementert i programvaren.

2.2.3 Kombinerert diagram



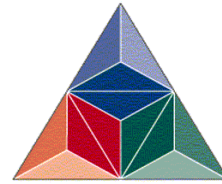
Figur 6: Kombinert diagram som viser relasjoner mellom de ulike hendelsene i simuleringen og programvaren.

3 Konklusjoner

Dette dokumentet er kun ment for å gi leserene en sammenheng mellom begreper og diagrammer brukt i de ulike fagretningene representert i prosjektet.

Referanser

- [1] Implementasjon av programvare regulert fartøy. programvare-implementasjon_1.0.pdf
- [2] Designdokument programvare regulert fartøy. programvare-design_regulert_2.0.pdf
- [3] simuleringsdokument_1.0.pdf
- [4] komponentdokumentasjon_3.0.pdf
- [5] koblingskjema_1.0.pdf



HØGSKOLEN
i Buskerud

Oppgavetittel: Hovedprosjekt - Testbenk - v1.0		
Fag(nr./navn): SFHO 3200 / Prosjektstyring med hovedprosjekt		
Gruppemedlemmer, signatur:		
_____ Vegard Torkelsen	_____ Michael Odden	_____ Herå Rørvik
_____ Trine Lindberg	_____ Luke Carambot	_____ Patrick Bjørum
Intern veileder: Dag Samuelsen Intern sensor: Olaf Hallan Graven Ekstern veileder og sensor: Ragnvald Otterlei		
Dato: 2011-04-07		

Innholdsfortegnelse

Dokumenthistorie.....	1
Innledning.....	1
1 Programvare.....	2
1.1 Kjøbling.....	3
1.2 Modusene.....	3
1.3 Manuell kommando.....	4
1.4 Statistiske data.....	6
1.5 Dynamiske data – overstyring.....	6
1.6 Ikke implementert.....	6
2 Fysikk testbenker.....	7
2.1 Krav til testbenk.....	7
2.2 Første versjon "stolen".....	7
2.3 Håndholdt benk.....	8
2.4 Aksebegrenset benk.....	9
Referanser.....	10

Dokumenthistorie

Versjon	Beskrivelse	Når
1.0	Opprettet dokumentet	2011-05-20

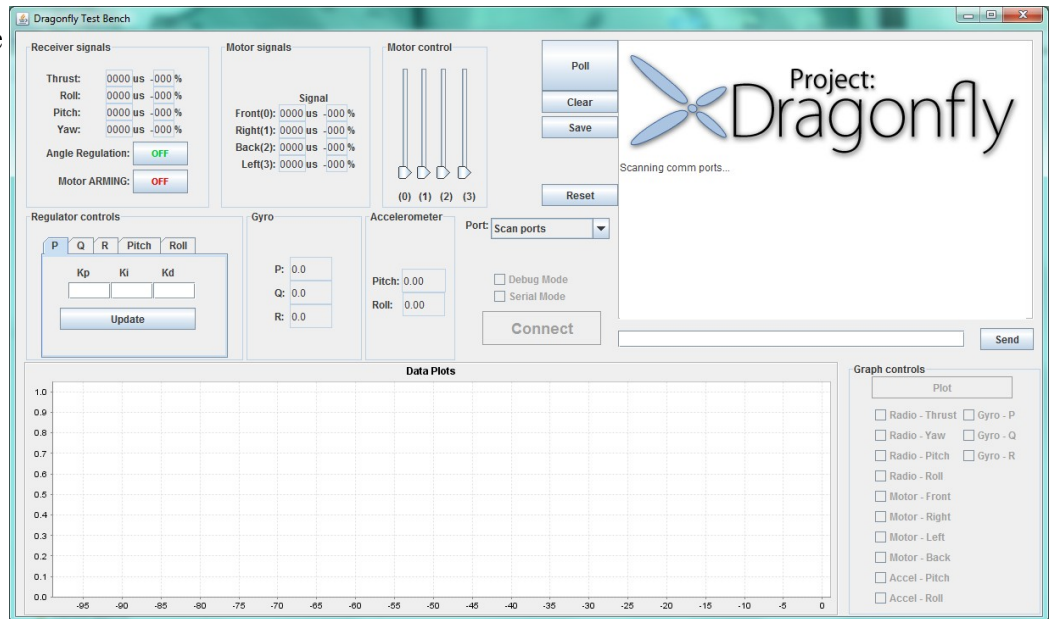
Tabell 1: Dokumenthistorie

Innledning

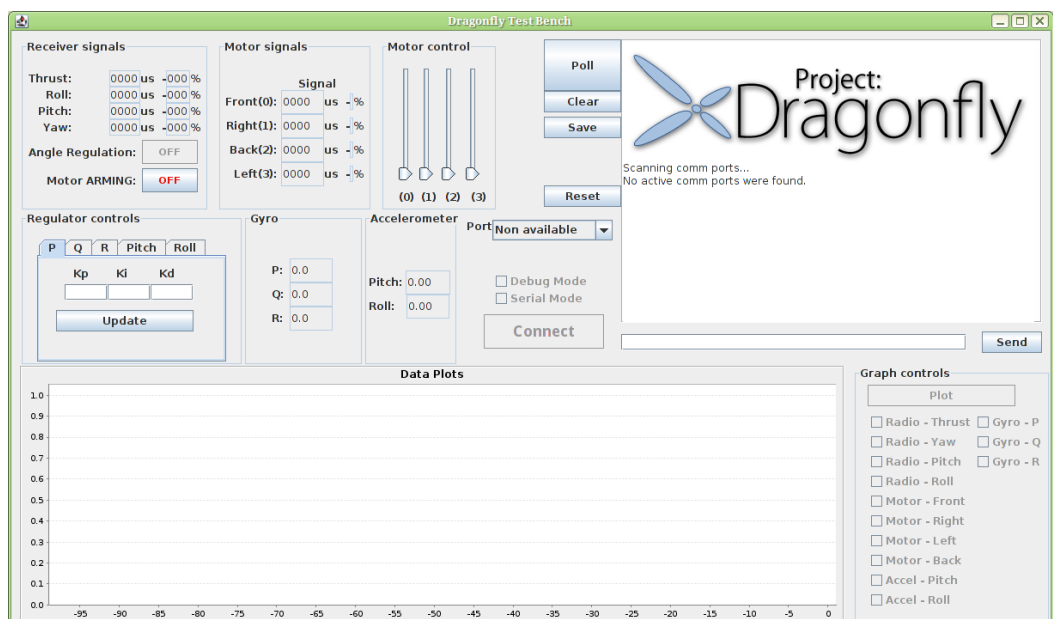
Dette dokumentet er for å dokumentere implementasjon av testbenker i prosjektet.

1 Programvare

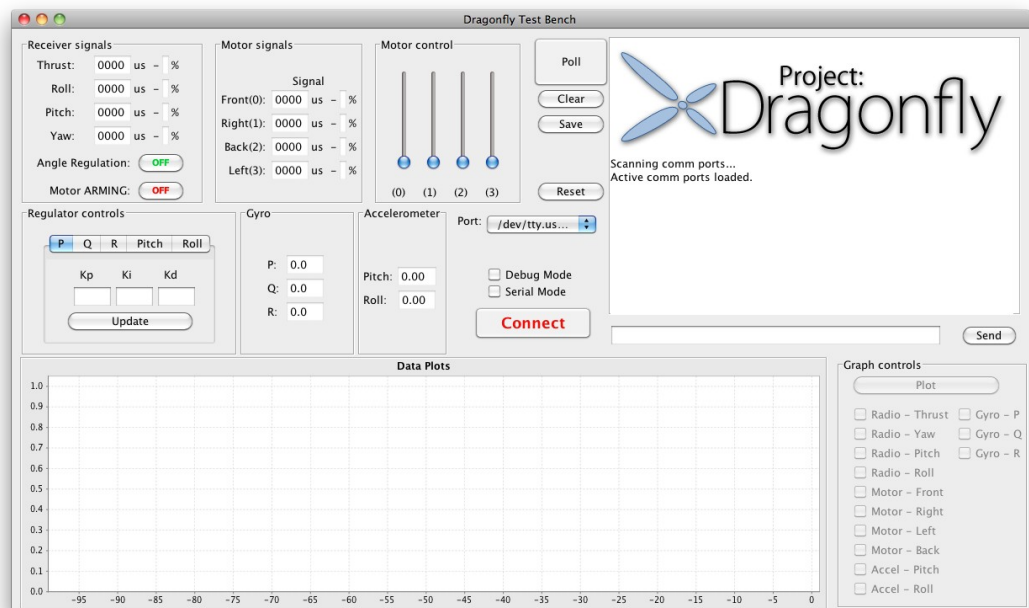
Hensikten med testbenk som programvare er å ha en måte til å endre vise parametere til reguleringsystemet på fartøyet uten å omprogrammere mikrokontrolleren. Det var spesielt denne delen av testbenken som ble brukt for å oppnå sine mål. De delene som ikke ble implementert var på grunn av mangel på tid. Spesielt datalagring og sanntidsplotting viste seg å kreve mer tid enn var tillatt å bruke på testbenken. Figurene under viser programmet på Windows, Ubuntu og OSX.



Figur 1: Dragonfly Test Bench i Windows



Figur 2: Dragonfly Test Bench i Ubuntu



Figur 3: Dragonfly Test Bench i OS X

1.1 Kobling

Dragonfly Test Bench bruker det samme seriell biblioteket som arduino IDE 22 bruker. Biblioteket heter rxtx og kan hentes på nett [1]. Når Dragonfly Test Bench kjøres blir en komboboks fylt med alle oppdagede seriellportene på PCen. Når brukeren har valgt porten som er koblet til fartøyet og klikket Connect-knappen, vil noe tekst komme på status boksen for å bekrefte at forbindelsen er suksessfull.

1.2 Modusene

Det fins to forskjellige moduser som kan hakes av over Connect-knappen.

1. Debug Mode er i midten og får fartøyet til å kontinuerlig oppdatere Dragonfly Test Bench GUIen ved diverse data som går gjennom mikrokontrolleren. Alle fanene blir oppdatert.
2. Serial Mode er en modus hvor nesten alt, inkludert radiosignalene overstyres av Dragonfly Test Bench. Det går an å styre de første 4 radiokanalene og alle motorsignalene direkte fra programmet. Det som ikke kan styres er sensorene.

1.3 Manuell kommando

Under prosjektlogoen er det et tekstfelt. Under det er det enda en tekstfelt og en Send-knapp. Alle kommandoene som Dragonfly Test Bench sender til fartøyet kan brukes også her. Kommandoer kan ha en av to funksjoner. En funksjon er at kommandoen setter verdier. Oppgitte verdier er overført til fartøyet og oppdatert hos programvaren på det. Den andre funksjonen er at kommandoen skal lese av verdiene fra fartøyet. Resultatet skal komme på tekstfeltet under logoen og de tilhørende elementene på formen oppdateres seg. Tabell 3. gir en oversikt på alle kommandoene[2].

Kommandofunksjonen, om den leser eller skriver en verdi, er bestemt av den første tegn i kommando teksten. Tabell 2. viser tegnene.

Tegn	Beskrivelse
<	Les variabelen
>	Sett variabelen

Table 2: Kommandofunksjon[2]

Tegn	Tillatte verdier	Beskrivelse
=	0 - 7	Trebit flagg som er Test Bench modusen. 1 – Debug Mode, 2 – Regulated Mode, 3 – Serial Mode
*	<i>ingen</i>	Les av alle verdier i ei linje (til Debug modus)
#	0,1	Vinkelregulator av/på
!	0,1	Motorarmering av/på
^	<i>ingen</i>	Les av gyrodata
_	<i>ingen</i>	Les av akselerometerdata (som vinkel i kun 2-akser)
A	1000-2000	Motorsignal – foran
B	1000-2000	Motorsignal – høyre
C	1000-2000	Motorsignal – venstre
D	1000-2000	Motorsignal – bak
a	1000-2000	Radiosignal – kanal 1

Tegn	Tillatte verdier	Beskrivelse
b	1000-2000	Radiosignal – kanal 2
c	1000-2000	Radiosignal – kanal 3
d	1000-2000	Radiosignal – kanal 4
e	1000, 2000	Radiosignal – kanal 5 (av/på)
f	1000, 2000	Radiosignal – kanal 6 (av/på)
z	0-3	Reset - 0: full reset. 1: resetter gyro offset. 2: resetter acc-offset.
Pp	Reelle tall	Proporsjonalleddet til P-PID
Pq	Reelle tall	Proporsjonalleddet til Q-PID
Pr	Reelle tall	Proporsjonalleddet til R-PID
Ps	Reelle tall	Proporsjonalleddet til Pitch-PID
Pt	Reelle tall	Proporsjonalleddet til Roll-PID
Pi	Reelle tall	Integralleddet til P-PID
Pj	Reelle tall	Integralleddet til Q-PID
Pk	Reelle tall	Integralleddet til R-PID
PI	Reelle tall	Integralleddet til Pitch-PID
Pm	Reelle tall	Integralleddet til Roll-PID
Pd	Reelle tall	Derivatleddet til P-PID
Pe	Reelle tall	Derivatleddet til Q-PID

Tegn	Tillatte verdier	Beskrivelse
Pf	Reelle tall	Derivatleddet til R-PID
Pg	Reelle tall	Derivatleddet til Pitch-PID
Ph	Reelle tall	Derivatleddet til Roll-PID

Table 3: Kommandoer[2]

1.4 Statiske data

Statiske data betyr at det er umulig å overstyre datastrømmen. Strengt tatt er alle feltene statiske data og kun noen få kan være dynamiske, som kan overstyres. Det er kun i Debug Mode at alle feltene blir oppdatert. Regulator verdiene kan hentes kun ved Poll-knappen og kan gjøres i alle modusene.

1.5 Dynamiske data – overstyring

Dynamiske data skjer når programmet er i Serial Mode. Da er det mulig å styres visse verdier direkte fra Dragonfly Test Bench. Radio (mottaker) signalene og motorsignalene kan settes ved å endre tekstfeltene. Verdiene er oppgitt i mikrosekunder. Motorsignalene kan også styres vha. fire glide brytere i "Motor control" -fanen.

Et unntak for at data er dynamisk kun i Serial Mode er regulator verdiene. Alle fanene til de fem regulatorene kan oppdateres kun vha. Update knappen. Dette kan gjøres i alle modusene.

1.6 Ikke implementert

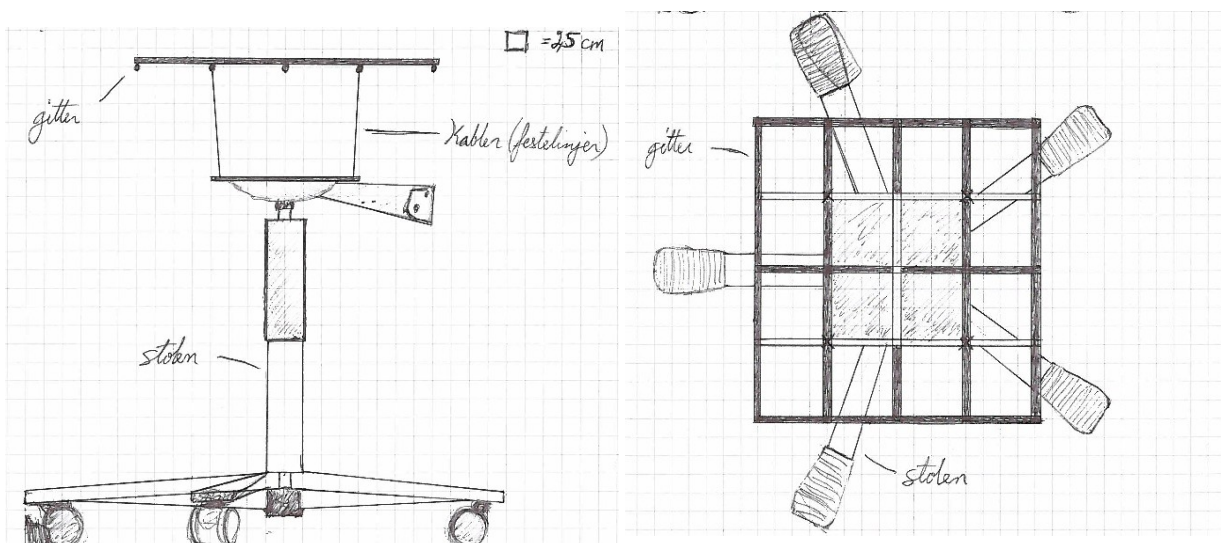
Det var ønsket å plote ut datastrømmen i sanntid. På grunn av mangel på tid til de som hadde kompetanse ble ikke dette videreutviklet. Det var valgt å bruk jFreeChart som plotting verktøy og mye står klart i bibliotekene, men sanntidsplotting er noe som ikke ennå er utviklet i jFreeChart. Jobben for å få dette til ble ned derfor prioritert.

2 Fysikk testbenker

2.1 Krav til testbenk

1. Fartøyet skal ikke kunne komme over 45°
2. Fartøyet skal ikke kunne komme over 1,5m i høyde
3. I alle mulige stillingene skal propellene ikke komme bort i noe (noen)
4. Det skal være mulig å koble en USB kabel fra data til fartøyet
5. Begrense bevegelsene ned til en akse (for testing)

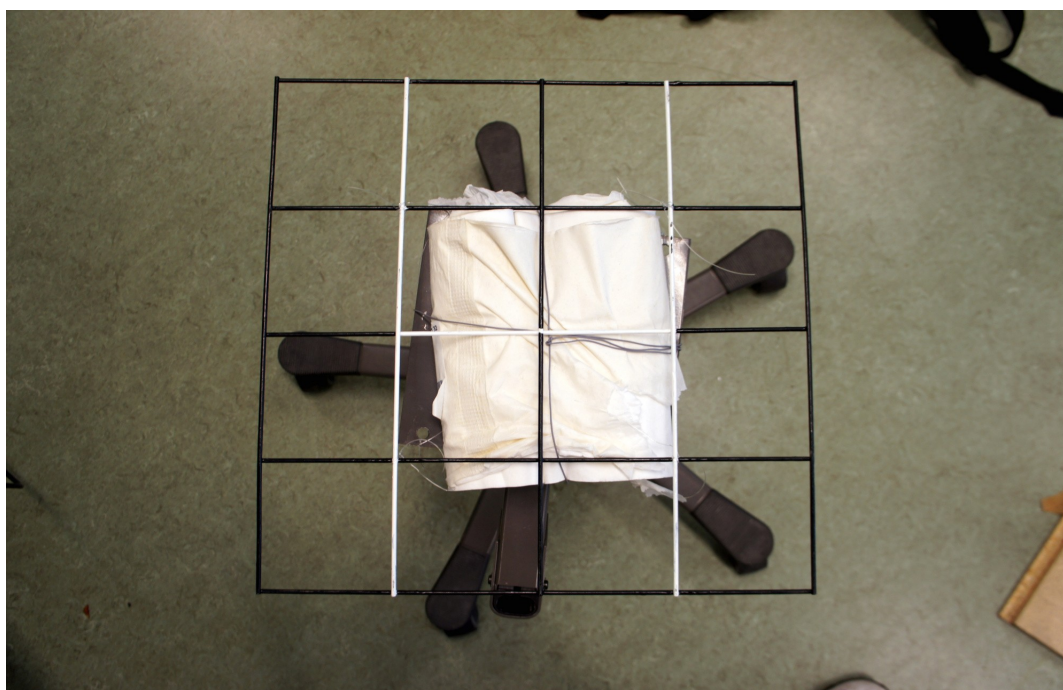
2.2 Første versjon "stolen"



Figur 4: Design av første testbenk

Vi brukte en vanlig kontorstol til å lage testbenken hvor vi fjernet setet og ryggen og festet et gitter til toppen med fiskesnører. Ideen var at fartøyet skulle festes rett på gitteret for sikker testing.

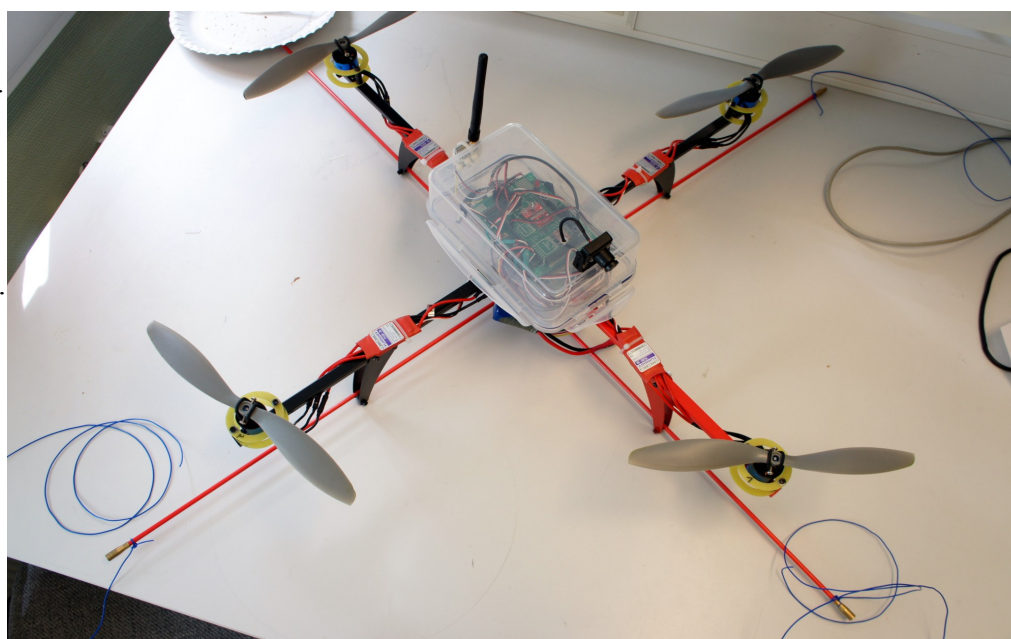
I praksis viste det seg at det var umulig å få testet hvordan fartøyet oppførte seg med de forskjellige parameterne fordi fiskesnørene måtte være så korte. Det førte til at fartøyet ikke fikk nok spillerom til å kunne korrigere sine egne feil. Det var i tillegg vanskelig å få begrenset bevegelsene til en akse.



Figur 5: Første testbenk

2.3 Håndholdt benk

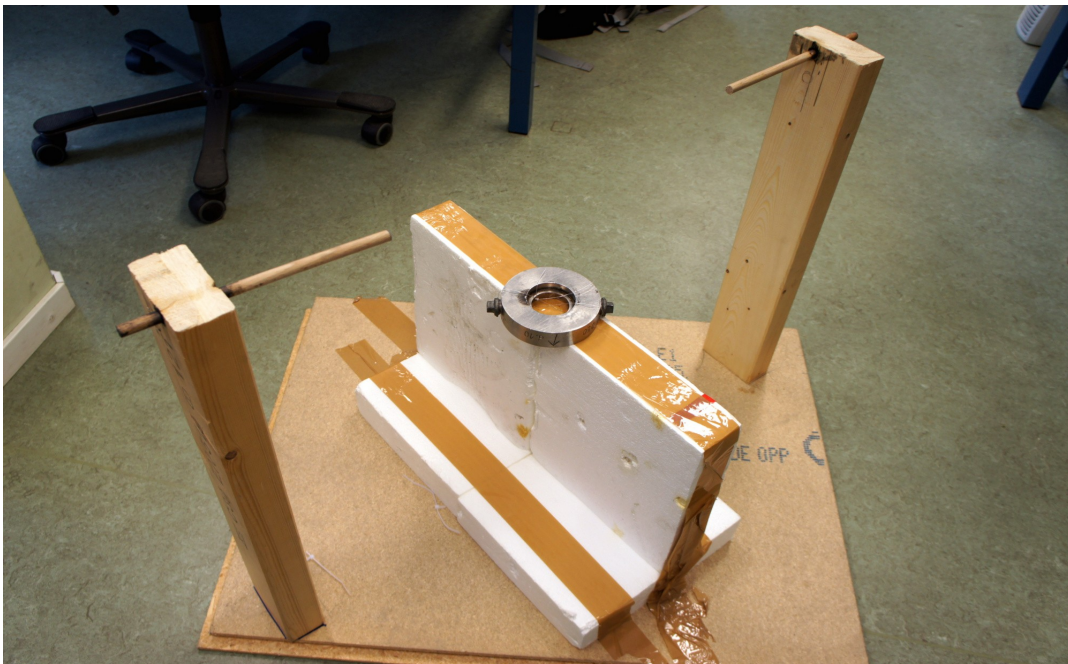
Andre versjon var mye enklere. Her forlenget vi armene med en stang som vi festet med strips. Det gjorde at propellene ikke ville komme ned i bakken hvis den skulle lande litt skjevt. Videre så festet vi en tråd til hver ende av forlengelsen slik at fire personer kunne holde i hver sin arm. Dermed kunne man teste fartøyet i sikkerhet uten fare for skade på personell eller fartøy. Det fungerte forholdsvis bra, men også her var det vanskelig å begrense bevegelsene til en akse. I tillegg er det problematisk å få noen konkrete data på forskjeller med forskjellige parametere.



Figur 6: Håndholdt benk

2.4 Aksebegrenset benk

Tredje og siste versjon bygget vi fra bunnen av med tanke på at fartøyet kun skal bevege seg i en akse, og skal være lett å se konkret hva som skjer ved forskjellige parametere. Konstruksjonen består av en sponplate som bunn, to planker opp på hver siden for å holde fartøyet oppe, og to pinner som festes langs samme akse, men på hver sin side av fartøyet. Til slutt la vi isopor på bakken langs den andre akselen. Fartøyet kan derfor kun bevege seg i en akse, og den vil ikke kunne bevege seg mer enn 45° i hver retning. Dermed er det mye lettere å se hvilke påvirkninger de forskjellige parameterne har når vi tester.

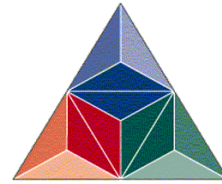


Figur 7: Aksebegrenset benk

Referanser

[1] <http://rxtx.qbang.org>

[2] seriellprotokoll_1.0



HØGSKOLEN
i Buskerud

Oppgavetittel: Hovedprosjekt - Matematisk modell - v2.0		
Fag(nr./navn): SFHO 3200 / Prosjektstyring med hovedprosjekt		
Gruppemedlemmer, signatur:		
_____ Vegard Torkelsen	_____ Michael Odden	_____ Herå Rørvik
_____ Trine Lindberg	_____ Luke Carambot	_____ Patrick Bjørum
Intern veileder: Dag Samuelsen Intern sensor: Olaf Hallan Graven Ekstern veileder og sensor: Ragnvald Otterlei		
Dato: 2011-05-18		

Innholdsfortegnelse

Dokumenthistorie.....	1
Innledning.....	1
1 Variabler og begreper.....	2
1.1 Variabler.....	2
1.2 Begreper.....	3
2 Referansesystemer.....	3
3 Rotasjoner.....	4
4 Posisjon og orientering.....	6
5 Kinematikk.....	6
5.1 Lineær hastighet.....	6
5.2 Vinkelhastighet.....	7
5.3 Totalhastigheten til partiklene.....	8
6 Dynamikk.....	8
6.1 Kraft og moment.....	8
6.2 Lineær bevegelsesmengde.....	9
6.3 Vinkel bevegelsesmengde.....	10
7 Total modell.....	12
8 Faserommet og konfigurasjonsrommet.....	12
9 Forutsetningene.....	13
Konklusjoner.....	14
Referanser.....	15

Dokumenthistorie

Versjon	Beskrivelse	Når
1.0	Dokument opprettet	2011-04-13
2.0	Korrekturlest og ferdigstilt	2011-05-20

Tabell 1: Dokumenthistorie

Innledning

Modellen utledet i dette dokumentet har jorda som referanse. Dette gjør modellen ganske kompleks, men bedre anvendbar i praktiske oppgaver, som å holde seg i en bestemt retning eller bevege seg fra punkt til punkt i rommet.

Dette dokumentet er laget i sammenheng med en Bachelor-oppgave som innebærer å regulere orienteringen til et multirotor-kopter slik at det kan holde seg stabilt i luften og gi operatøren mulighet til enkelt å styre fartøyet med en fjernkontroll. Strengt tatt kreves det ingen ytre referanse for å oppnå dette, og bare en del av den komplette modellen utviklet i dette dokumentet er nødvendig til denne oppgaven. Denne delen er beskrevet i kapittel 6.3 og er summert i likning (31).

En grunn til at en mer komplett og kompleks modell ble utledet var fordi vi på forhånd ikke hadde oversikt over hvilke deler av modellen vi kom til å trenge. Men denne komplette modellen kan likevel komme godt med ved en eventuelt videre utvikling av systemet.

1 Variabler og begreper

1.1 Variabler

$R_E^B()$ = Rotasjon (matrise) fra det faste koordinatsystemet E til kroppens koordinatsystem B

$R_B^E()$ = Rotasjon (matrise) fra B til E

$E = [\hat{x}, \hat{y}, \hat{z}]^T$ Inertialsystem (rom) nord, øst og ned

x = Inertialposisjon (nord) av fartøyet langs \hat{x} i E

y = Inertialposisjon (øst) av fartøyet langs \hat{y} i E

z = Inertialposisjon (høyde) av fartøyet langs $-\hat{z}$ i E

$r_E = [x, y, z]^T$ Inertialposisjonsvektor til fartøyet i E

$B = [\hat{i}, \hat{j}, \hat{k}]^T$ Kropslig koordinatsystem (bevegelig)

$b_B = [i, j, k]^T$ En vilkårlig vektor i kroppens koordinatsystemet B

u = Hastighet til kroppen målt langs \hat{i} i B

v = Hastighet til kroppen målt langs \hat{j} i B

w = Hastighet til kroppen målt langs \hat{k} i B

$v_B = [u, v, w]^T$ Hastighetsvektor til fartøyet i B

v = Hastighetsvektor til fartøyet i E

θ = Vinkel (roll) om \hat{i}

ϕ = Vinkel (pitch) om \hat{j}

ψ = Vinkel (yaw) om \hat{k}

$\Lambda = [\theta, \phi, \psi]^T$ Vinkelvektor

p = Vinkelhastighet (roll) målt om \hat{i} i B

q = Vinkelhastighet (pitch) målt om \hat{j} i B

r = Vinkelhastighet (yaw) målt om \hat{k} i B

$\omega_B = [p, q, r]^T$ Vinkelhastighetsvektor målt i B , i radianer/sek

ω = Vinkelhastighetsvektor i E , i radianer/sek

$\eta = [x, y, z, u, v, w, \theta, \phi, \psi, p, q, r]^T$ Tilstands variabel

m = Masse

M = Totalmasse

R = Tyngdepunktet

g = Akselerasjon fra jorden

p = Bevegelsesmengde

L = Bevegelsesmengde (roterende)

I = Trehetsmoment (fartøyet)

Ω_i = Vinkelhastighet til den i-ende propell)

b = Løftekonstant

d = Motstand av luft til rotasjon av propell

l = Avstand fra origo til midten av motor/propell

J = Trehetsmoment (propell)

L = Lagrangian

τ = Moment
 T = Kinetiskenergi
 U = Potensialenergi
 U_i = Inngangsvariabler
 F_i = Løftkraft den i-ende motor
 K = Totalkraft om $\hat{\mathbf{k}}$

1.2 Begreper

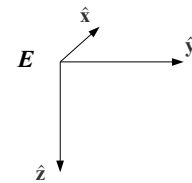
Generaliserte koordinater: Koordinatsystemer istedet for Kartesisk.

Konservativ mengde: En funksjon av de avhengige variablene som er konstant.

Faserom: Det sett av mulige posisjoner og bevegelsesmengder til et system i et tidspunkt.

2 Referansesystemer

Det brukes et ortogonalt koordinatsystem for å beskrive kroppen. Origo er valgt til å være tyngdepunktet til fartøyet, dette forenkler modellen på visse områder. Dette referansesystemet er i bevegelse. Det blir nødvendig å beskrive bevegelsene til kroppen med hensyn til et annet referansesystem som ikke er i bevegelse. Her er det naturlig å velge jorden til dette referansesystemet, og vi kaller dette (NED) koordinatsystemet til fartøyet for et inertialsystem.



Figur 1: Inertialsystem med det vanlige (NED) koordinatsystemet til fartøyet

Siden alt beskrives i rommet må det være tre akser. La

- det bevegelige kropps-referansesystemet B har koordinatene $\hat{\mathbf{i}}, \hat{\mathbf{j}}, \hat{\mathbf{k}}$
- det fast inertialsystemet E ha koordinatene $\hat{\mathbf{x}}, \hat{\mathbf{y}}, \hat{\mathbf{z}}$,

Koordinatsystemene har en høyre-hånds orientering og det vanlig at et fartøy bruker positiv $\hat{\mathbf{i}}$ framover, positiv $\hat{\mathbf{j}}$ til høyre for fartøyet (da en ser framover) og positiv $\hat{\mathbf{k}}$ i ned-retningen. Det er også vanlig å kalle disse aksene $\hat{\mathbf{i}}, \hat{\mathbf{j}}$ og $\hat{\mathbf{k}}$ for roll, pitch og yaw henholdsvis.

3 Rotasjoner

Det er to måter å oppfatte en rotasjon på, passiv eller aktiv (figur 2.). Aktive rotasjoner transformerer (roterer) basisen, vektorene beholder sine koordinater i den originale basisen mens de får nye koordinater i den transformerte basisen. Passive rotasjoner transformerer kun vektorene og dermed holder basisen seg uendret.

En aktiv rotasjon er brukt for å rotere et referansesystem og er illustrert i to-dimensjoner i figur 3. En rotasjon til den nye basisen er gitt av

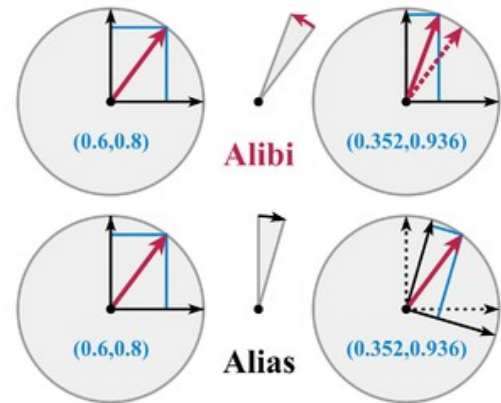
$$\begin{aligned}\hat{x} &= \hat{i} \cdot \cos \theta + \hat{j} \cdot \sin \theta \\ \hat{y} &= -\hat{i} \cdot \sin \theta + \hat{j} \cdot \cos \theta\end{aligned}\quad (1)$$

og på matriseform

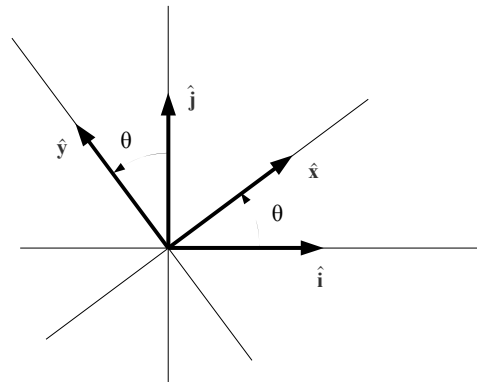
$$\begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} \hat{i} \\ \hat{j} \end{bmatrix}\quad (2)$$

hvor rotasjonsmatrisen er

$$R(\theta) = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix}\quad (3)$$



Figur 2: Aktive (Alias) vs Passive (Alibi) rotasjoner [3]



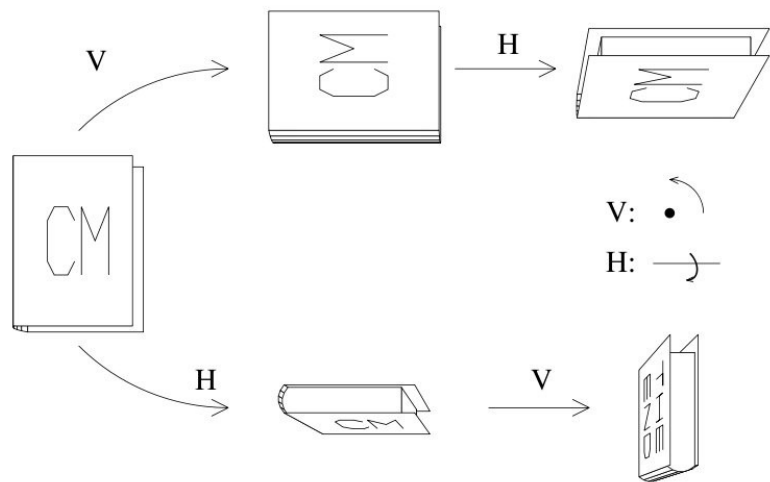
Figur 3: Rotasjon i 2D

I tre dimensjoner er det igjen to måter å tenke på en rotasjon på. En måte er at en gitt akse og en gitt vinkel beskriver en rotasjon. Den andre er rotasjonen som kommer av en rekke rotasjoner om forskjellige akser.

Så er det valgt å definere rotasjon som en rekke rotasjoner om forskjellige akser (basisene).

Det neste som må nevnes er at en rekke rotasjoner ikke er kommutative. Det vil si at

$VH \neq HV$ hvor V er å rotere et objekt om den vertikalsaksen og H er å rotere om den horisontalaksen. Figur 4. viser dette grafisk.



Figur 4: Illustrasjon at multiplekasjon ikke er kommutativ for rotasjonsmatriser [1]

Rotasjonsmatrisene til rundt hver akse av **B**:

$$R(\hat{\mathbf{k}}, \psi) \triangleq \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4)$$

$$R(\hat{\mathbf{j}}, \phi) \triangleq \begin{bmatrix} \cos \phi & 0 & -\sin \phi \\ 0 & 1 & 0 \\ \sin \phi & 0 & \cos \phi \end{bmatrix} \quad (5)$$

$$R(\hat{\mathbf{i}}, \theta) \triangleq \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & \sin \theta \\ 0 & -\sin \theta & \cos \theta \end{bmatrix} \quad (6)$$

Det finnes flere måter å legge sammen rotasjoner på. Det som er sagt til å være vanlig for fartøy er Tait-Bryan metoden [5]. Hovedtrekket er at det roteres om hver akse, i motsetning til at det er flere rotasjoner på én akse. Den sammensatte rotasjonsmatrisen fra inertialsystemet **E** til kroppreferansesystemet **B** er:

$$\begin{aligned} R_E^B(\theta, \phi, \psi) &= R(\hat{\mathbf{i}}, \theta) R(\hat{\mathbf{j}}, \phi) R(\hat{\mathbf{k}}, \psi) \\ &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & \sin \theta \\ 0 & -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} \cos \phi & 0 & -\sin \phi \\ 0 & 1 & 0 \\ \sin \phi & 0 & \cos \phi \end{bmatrix} \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} c\phi c\psi & c\phi s\psi & -s\phi \\ s\theta s\phi c\psi - c\theta s\psi & s\theta s\phi s\psi + c\theta c\psi & s\theta c\phi \\ c\theta s\phi c\psi + s\theta s\psi & c\theta s\phi s\psi - s\theta c\psi & c\theta c\phi \end{bmatrix} \quad (7) \end{aligned}$$

hvor $c\theta \triangleq \cos\theta$ og $s\theta \triangleq \sin\theta$.

Siden alle matrisene i regnestykkene er ortogonale blir den inverse til rotasjonsmatrisen den samme som den transponerte. Så fra \mathbf{B} til \mathbf{E} blir:

$$R_B^E(\theta, \phi, \psi) = R_E^B(\theta, \phi, \psi)^T = \begin{bmatrix} c\phi c\psi & s\theta s\phi c\psi - c\theta s\psi & c\theta s\phi c\psi + s\theta s\psi \\ c\phi s\psi & s\theta s\phi s\psi + c\theta c\psi & c\theta s\phi s\psi - s\theta c\psi \\ -s\phi & s\theta c\phi & c\theta c\phi \end{bmatrix} \quad (8)$$

4 Posisjon og orientering

En stiv kropp, slik som fartøyet, er en samling av partikler. De er faste i kroppen, men kroppen selv kan roteres og beveges. La punktet fartøyet kan rotere om være tyngdepunktet til kroppen og anta at det er det geometriske midtpunktet som er origo $(0,0,0)$ i \mathbf{B} . Posisjon til dette punktet i rommet skal være gitt ved \mathbf{r}_E . La $R_B^E()$ være en rotasjon om tyngdepunktet til fartøyet slik at orienteringen til en vektor \mathbf{b} fast i kroppens koordinatsystem blir gitt av $R_B^E()\mathbf{b}$. Så i et vilkårlig punkt i kroppen, sett i rommet, er resultatet av først å rotere punktet om origo og deretter forflytte det lik:

$$\mathbf{r} = R_B^E()\mathbf{b} + \mathbf{r}_E. \quad (9)$$

En fysisk tolkning på $R_B^E()$ er at den gir retningene til kroppens akser i \mathbf{E} (rommet). Så

$$\hat{\mathbf{i}}_E = R_B^E()\hat{\mathbf{i}}_B, \quad \hat{\mathbf{j}}_E = R_B^E()\hat{\mathbf{j}}_B \quad \text{og} \quad \hat{\mathbf{k}}_E = R_B^E()\hat{\mathbf{k}}_B.$$

5 Kinematikk

Med \mathbf{r}_E og $R_B^E()$ gitt som posisjonen og orientering for enkelhets skyld, er forandringer over tid det neste å se på. Behovet er uttrykkene for $\dot{\mathbf{r}}_E$ og $\dot{R}_B^E()$.

5.1 Lineær hastighet

Siden \mathbf{r}_E er tyngdepunktet til kroppen er lineærhastigheten gitt ved,

$$\mathbf{v} = \dot{\mathbf{r}}_E = (\dot{x}, \dot{y}, \dot{z})^T \quad (10)$$

Når disse hastighetene er gitt ved sensorer på kroppen er forholdet mellom hastighetene $(\dot{x}, \dot{y}, \dot{z})^T$ i \mathbf{E} og hastighetene $(u, v, w)^T$ i \mathbf{B} gitt ved:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix} = R_B^E(\theta, \phi, \psi) \begin{pmatrix} u \\ v \\ w \end{pmatrix}$$

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix} = \begin{bmatrix} c\phi c\psi & s\theta s\phi c\psi - c\theta s\psi & c\theta s\phi c\psi + s\theta s\psi \\ c\phi s\psi & s\theta s\phi s\psi + c\theta c\psi & c\theta s\phi s\psi - s\theta c\psi \\ -s\phi & s\theta c\phi & c\theta c\phi \end{bmatrix} \begin{pmatrix} u \\ v \\ w \end{pmatrix}. \quad (11)$$

5.2 Vinkelhastighet

I følge Euler, finnes det en akse som går igjennom rotasjonspunktet når kroppen roterer (anta ingen lineære bevegelser av punktet) og la aksene være gitt ved ω . $\dot{R}_B^E()$ ikke er ω . Så hva er forholdet mellom $R_B^E()$ og ω ?

Anta at en vektor \mathbf{b} roterer med kroppen med vinkelhastighet ω . En dekomponering av \mathbf{b} gir at \mathbf{b} har en komponent som er parallell med ω og en komponent som er rettvinklet til ω , slik at $\mathbf{b} = \mathbf{b}_{\parallel} + \mathbf{b}_{\perp}$. Videre er endringer i \mathbf{b} gitt ved $\dot{\mathbf{b}} = \omega \times \mathbf{b}_{\perp}$. Men siden $\omega \times \mathbf{b}_{\parallel} = 0$, er det mulig å skrive,

$$\dot{\mathbf{b}} = \omega \times \mathbf{b}. \quad (12)$$

som er generelt for roterende vektorer.

$$\text{Derved er } \dot{R}_B^E() = \omega \times R_B^E(). \quad (13)$$

Men $R_B^E()$ er en matrise. Kryssproduktet på matriseform kan skrives som,

$$\mathbf{a} \times \mathbf{b} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \quad (14)$$

Så endringer i orientering over tid er lik,

$$\dot{R}_B^E() = \omega \times R_B^E() = \begin{bmatrix} 0 & -\dot{\psi} & \dot{\phi} \\ \dot{\psi} & 0 & -\dot{\theta} \\ -\dot{\phi} & \dot{\theta} & 0 \end{bmatrix} \begin{bmatrix} c\phi c\psi & s\theta s\phi c\psi - c\theta s\psi & c\theta s\phi c\psi + s\theta s\psi \\ c\phi s\psi & s\theta s\phi s\psi + c\theta c\psi & c\theta s\phi s\psi - s\theta c\psi \\ -s\phi & s\theta c\phi & c\theta c\phi \end{bmatrix} \quad (15)$$

Det som er vesentlig i forhold til oppgaven, som er å regulere orientering til fartøyet, er forholdet mellom vinkelhastighetene målt på fartøyet og vinklene (θ, ϕ, ψ) i rommet. Å regne ut forholdet er ikke så rett fram siden disse vinklene kommer får forskjellige rotasjoner. Ved små vinkelendringer (hastigheter $[\dot{\theta}, \dot{\phi}, \dot{\psi}]^T$) ville de enkelte rotasjonene nærme seg identitetsmatrisen I [6], som blir da

$$\begin{aligned} \begin{pmatrix} p \\ q \\ r \end{pmatrix} &= R(\hat{\mathbf{i}}, \dot{\theta}) \begin{pmatrix} \dot{\theta} \\ 0 \\ 0 \end{pmatrix} + R(\hat{\mathbf{i}}, \theta) R(\hat{\mathbf{j}}, \dot{\phi}) \begin{pmatrix} 0 \\ \dot{\phi} \\ 0 \end{pmatrix} + R(\hat{\mathbf{i}}, \theta) R(\hat{\mathbf{j}}, \phi) R(\hat{\mathbf{k}}, \dot{\psi}) \begin{pmatrix} 0 \\ 0 \\ \dot{\psi} \end{pmatrix} \\ &= \begin{pmatrix} \dot{\theta} \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & \sin \theta \\ 0 & -\sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} 0 \\ \dot{\phi} \\ 0 \end{pmatrix} + \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & \sin \theta \\ 0 & -\sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} \cos \phi & 0 & -\sin \phi \\ 0 & 1 & 0 \\ \sin \phi & 0 & \cos \phi \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ \dot{\psi} \end{pmatrix} \\ &= \begin{pmatrix} 1 & 0 & -s\phi \\ 0 & c\theta & s\theta c\phi \\ 0 & -s\theta & c\theta c\phi \end{pmatrix} \begin{pmatrix} \dot{\theta} \\ \dot{\phi} \\ \dot{\psi} \end{pmatrix}. \end{aligned}$$

Det ønskede forholdet kommer frem ved å invertere og dette gir likning 16 på neste side.

$$\begin{pmatrix} \dot{\theta} \\ \dot{\phi} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} 1 & \sin \theta \cdot \tan \phi & \cos \theta \cdot \tan \phi \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta \cdot \sec \phi & \cos \theta \cdot \sec \phi \end{pmatrix} \begin{pmatrix} p \\ q \\ r \end{pmatrix}. \quad (16)$$

Legg merke til at likningene (11) og (16) gir en omforming av kroppens lineære -/vinkelhastigheter til rommets lineære -/vinkelhastigheter. Kravene til første del av reguleringen er å regulere de absolutte vinklene i rom og dermed er likningen (16) av spesiell betydning.

5.3 Totalhastigheten til partiklene

Legg også merke til at en partikkel i kroppen, sett i rom, er gitt ved

$$\mathbf{r} = \mathbf{r}_E + R_B^E(\cdot) \mathbf{b} \quad (9)$$

Nå er det mulig å derivere dette og det gir

$$\dot{\mathbf{r}} = \mathbf{v} + \boldsymbol{\omega} \times R_B^E(\cdot) \mathbf{b} \quad (17)$$

En omskriving ved $\mathbf{b} = \mathbf{r} - \mathbf{r}_E$ fører til

$$\dot{\mathbf{r}} = \mathbf{v} + \boldsymbol{\omega} \times R_B^E(\cdot) (\mathbf{r} - \mathbf{r}_E) \quad (18)$$

som beviser at hastigheten til partiklene i kroppen har to komponenter, en lineær komponent (\mathbf{v}) og en vinkelkomponent, $\boldsymbol{\omega} \times R_B^E(\cdot) (\mathbf{r} - \mathbf{r}_E)$.

6 Dynamikk

6.1 Kraft og moment

Aktuatorene (motorene med propell) ligger fast på kroppen. Kraft og moment fra disse er gitt i \mathbf{B} , men ytre krefter som vind og tyngdefeltet er gitt i \mathbf{E} .

Moment er forskjellige fra kraft i det at det er avhengig av posisjonen til tyngdepunktet, l . To propeller går motsatte retninger for å skape balanse i moment. Løftekreftene er vinkelrett på xy-planet til fartøyet.

Dette gir følgende likninger

$$U_1 = (F_1 + F_2 + F_3 + F_4) \quad \text{Løftkraft} \quad (19)$$

$$U_2 = l(F_4 - F_2) \quad \text{Moment om } \hat{\mathbf{i}} \quad (20)$$

$$U_3 = l(F_3 - F_1) \quad \text{Moment om } \hat{\mathbf{j}} \quad (21)$$

$$U_4 = K \quad \text{Moment om } \hat{\mathbf{k}} \quad (22)$$

hvor

$$F_i = b \Omega_i^2 \text{ og}$$

$$K = d (\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2)$$

Det siste kraften som er tilstede er tyngdekraften.

6.2 Lineær bevegelsesmengde

Lineær bevegelsesmengde er gitt ved:

$$m \frac{d \mathbf{r}_E}{d t_E} = \mathbf{p}$$

hvor $\frac{d \mathbf{r}_E}{d t_E}$ er hastigheten sett i E . Likningen (17) blir brukt siden kroppen kan roterer og fører til

$$m \frac{d \mathbf{r}_E}{d t_E} = m (\mathbf{v} + \boldsymbol{\omega} \times R_B^E(\cdot) \mathbf{b}) = \mathbf{p}$$

Dette kan forenkles ved bruk av tyngdepunktet som er i origo i B , og dermed er summen av alle \mathbf{b} vektorene lik null. Ved \mathbf{b} lik null blir likningen som kjent slik at alle partiklene kan tenkes samlet i ett punkt og vi har

$$M \dot{\mathbf{r}}_E = M \mathbf{v} = \mathbf{p} \quad (23)$$

Det som er viktig er at likningen ikke lenger er avhengig av rotasjonen. Videre får vi at

$$\mathbf{v} = \frac{\mathbf{p}}{M}$$

og sammenhengen mellom lineær bevegelsesmengde og kraft blir

$$\dot{\mathbf{p}} = \frac{\dot{\mathbf{p}}}{M} = \frac{\mathbf{F}_{total}}{M}$$

Da er lineær bevegelsesmengde (og hastigheten) konstant når kreftene er lik null.

Sett i kroppen på fartøyet er totalkraften lik tyngdekraft pluss løftekraften og blir

$$\dot{\mathbf{p}}_b = \frac{1}{M} (\mathbf{F}_g + \mathbf{F}_L) \quad (24)$$

Tyngdekraften er gitt i E og må roteres inn på kroppen.

$$\dot{\mathbf{p}}_b = \frac{1}{M} \left(R_E^B(\cdot) \begin{pmatrix} 0 \\ 0 \\ M\mathbf{g} \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ -\mathbf{F} \end{pmatrix} \right)$$

$$\begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{pmatrix} = \begin{pmatrix} -g \sin \phi \\ g \cos \phi \sin \theta \\ g \cos \phi \cos \theta \end{pmatrix} + \frac{1}{M} \begin{pmatrix} 0 \\ 0 \\ -\mathbf{F} \end{pmatrix} \quad (25)$$

6.3 Vinkel bevegelsesmengde

Motivasjonen for å jobbe med noe så lite intuitivt som vinkel bevegelsesmengde er at den er en porsjonert mengde og dermed forenkler likningene fremfor å bruke vinkelhastighet, som ikke er en porsjonert mengde. Dette betyr at vinkel bevegelsesmengde er uavhengig av translatoriske bevegelser.

Vinkel bevegelsesmengde er gitt ved,

$$\mathbf{L} = \mathbf{I} \boldsymbol{\omega} \quad (26)$$

hvor \mathbf{I} er treghetsmoment. Siden modellen skal beskrive fartøyet i rommet blir treghetsmoment en tensor (generalisert vektor). Kroppen til fartøyet er symmetrisk om hver akse. Dette betyr at produktene av treghetsmomentene mellom aksene blir neglisjerbare. Slik sett blir treghetsmomentstensorene diagonale og er forenklet til treghetsmomentene av hver akse og gir de kjente Euler likningene.[1] Disse aksene kalles prinsipielle akser til rotasjonen, og er valgt til å være aksene til kroppens referansesystem.

Så under disse forutsetningene er treghetsmomentet lik

$$\mathbf{I} = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}. \quad (27)$$

Ved å ta hensyn til at kroppens roterende koordinatsystemet er

$$\frac{d\mathbf{L}}{dt_E} = \frac{d\mathbf{L}}{dt_B} + \boldsymbol{\omega} \times \mathbf{L}$$

og ved innsetting av (26) blir dette

$$\boldsymbol{\tau} = \dot{\mathbf{L}} = (\mathbf{I} \dot{\boldsymbol{\omega}}) + \boldsymbol{\omega} \times (\mathbf{I} \boldsymbol{\omega}) \quad (28)$$

Vinkelakselerasjonene kommer til å være de ukjente under reguleringen, og likningen kan omformes slik at vinkelakselerasjonen er lik summen av det som kommer fra vinkelhastighetene og momentene fra driftsystemet. Løser likning (28) med hensyn på vinkelhastigheten og får da

$$\dot{\boldsymbol{\omega}} = \mathbf{I}^{-1}(\boldsymbol{\omega} \times (\mathbf{I} \boldsymbol{\omega}) + \boldsymbol{\tau}) \quad (29)$$

$$\begin{pmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{pmatrix} = \begin{bmatrix} \frac{1}{I_{xx}} & 0 & 0 \\ 0 & \frac{1}{I_{yy}} & 0 \\ 0 & 0 & \frac{1}{I_{zz}} \end{bmatrix} \left(\begin{bmatrix} 0 & r & -q \\ -r & 0 & p \\ q & -p & 0 \end{bmatrix} \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \begin{pmatrix} p \\ q \\ r \end{pmatrix} + \boldsymbol{\tau} \right)$$

$$\begin{pmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{pmatrix} = \begin{bmatrix} \frac{1}{I_{xx}} & 0 & 0 \\ 0 & \frac{1}{I_{yy}} & 0 \\ 0 & 0 & \frac{1}{I_{zz}} \end{bmatrix} \left(\begin{bmatrix} 0 & r & -q \\ -r & 0 & p \\ q & -p & 0 \end{bmatrix} \begin{pmatrix} I_{xx} p \\ I_{yy} q \\ I_{zz} r \end{pmatrix} + \boldsymbol{\tau} \right)$$

$$\begin{pmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{pmatrix} = \begin{bmatrix} \frac{1}{I_{xx}} & 0 & 0 \\ 0 & \frac{1}{I_{yy}} & 0 \\ 0 & 0 & \frac{1}{I_{zz}} \end{bmatrix} \begin{pmatrix} (I_{yy} - I_{zz})qr \\ (I_{zz} - I_{xx})pr \\ (I_{xx} - I_{yy})pq \end{pmatrix} + \boldsymbol{\tau}$$

$$\begin{pmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{pmatrix} = \begin{pmatrix} \frac{(I_{yy} - I_{zz})}{I_{xx}}qr \\ \frac{(I_{zz} - I_{xx})}{I_{yy}}pr \\ \frac{(I_{xx} - I_{yy})}{I_{zz}}pq \end{pmatrix} + \begin{pmatrix} \frac{\tau_i}{I_{xx}} \\ \frac{\tau_j}{I_{yy}} \\ \frac{\tau_k}{I_{zz}} \end{pmatrix} \quad (30)$$

Dette er en omforming av Euler's likningene.

Andre leddet i likningen (30) er summen av de ubevarte momentene som kommer av driftsystemet og gyroeffektene til propellene.

$$\tau_i = U_2 + q\mathbf{J}(-\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4)$$

$$\tau_j = U_3 + p\mathbf{J}(\Omega_1 - \Omega_2 + \Omega_3 - \Omega_4)$$

$$\tau_k = lK$$

$$\begin{pmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{pmatrix} = \begin{pmatrix} \frac{(I_{yy} - I_{zz})}{I_{xx}}qr \\ \frac{(I_{zz} - I_{xx})}{I_{yy}}pr \\ \frac{(I_{xx} - I_{yy})}{I_{zz}}pq \end{pmatrix} + \begin{pmatrix} \frac{U_2 + q\mathbf{J}(-\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4)}{I_{xx}} \\ \frac{U_3 + p\mathbf{J}(\Omega_1 - \Omega_2 + \Omega_3 - \Omega_4)}{I_{yy}} \\ \frac{lK}{I_{zz}} \end{pmatrix} \quad (31)$$

7 Total modell

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix} = R_B^E(\theta, \phi, \psi) \begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{bmatrix} c\phi c\psi & s\theta s\phi c\psi - c\theta s\psi & c\theta s\phi c\psi + s\theta s\psi \\ c\phi s\psi & s\theta s\phi s\psi + c\theta c\psi & c\theta s\phi s\psi - s\theta c\psi \\ -s\phi & s\theta c\phi & c\theta c\phi \end{bmatrix} \begin{pmatrix} u \\ v \\ w \end{pmatrix} \quad (11)$$

$$\begin{pmatrix} \dot{\theta} \\ \dot{\phi} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} 1 & \sin\theta \cdot \tan\phi & \cos\theta \cdot \tan\phi \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta \cdot \sec\phi & \cos\theta \cdot \sec\phi \end{pmatrix} \begin{pmatrix} p \\ q \\ r \end{pmatrix} \quad (16)$$

$$\begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{pmatrix} = \begin{pmatrix} -g \sin\phi \\ g \cos\phi \sin\theta \\ g \cos\phi \cos\theta \end{pmatrix} + \frac{1}{M} \begin{pmatrix} 0 \\ 0 \\ -F \end{pmatrix} \quad (25)$$

$$\begin{pmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{pmatrix} = \begin{pmatrix} \frac{(I_{yy} - I_{zz})}{I_{xx}} qr \\ \frac{(I_{zz} - I_{xx})}{I_{yy}} pr \\ \frac{(I_{xx} - I_{yy})}{I_{zz}} pq \end{pmatrix} + \begin{pmatrix} \frac{U_2 + qJ(-\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4)}{I_{xx}} \\ \frac{U_3 + pJ(\Omega_1 - \Omega_2 + \Omega_3 - \Omega_4)}{I_{yy}} \\ \frac{IK}{I_{zz}} \end{pmatrix} \quad (31)$$

8 Faserrommet og konfigurasjonsrommet

Fartøyet har 6 frihetsgrader. Tre mulige retninger å forflytte seg og tre mulige rotasjoner rundt disse retningene. Et faserom er rommet av alle mulige posisjoner og bevegelsesmengder. Faserrommet til dette system er et 12-dimensjonalt rom. Når det bare er posisjoner som skal håndteres kalles det et konfigurasjonsrom hvor dimensjonene er like antall frihetsgrader til systemet.

$$\boldsymbol{\eta} = [x, y, z, u, v, w, \theta, \phi, \psi, p, q, r]^T$$

Posisjonene til fartøyet (x, y, z) er gitt i inertialsystemet \mathbf{E} . Hastighetene (u, v, w) og vinkelhastighetene (p, q, r) er målt av sensorer som er på kroppen og dermed er gitt i \mathbf{B} . Tait-Bryan vinklene (θ, ϕ, ψ) er gitt i middeltidlige referansesystemer bestemt av rekkefølgen til den sammensatte rotasjonsmatrisen fra \mathbf{E} til \mathbf{B} .

9 Forutsetningene

- Vi ønsker at de ikke-diagonale elementene i treghetsmoment-matrisen skal være så små som mulig slik at de kan regnes lik null og derved linearisere vinkelmomentlikningene.
 - Kroppens koordinatsystem har tyngdepunktet i origo.
 - Fartøyet er symmetrisk om tre ortogonale akser (kalles også prinsipielle akser) og koordinatsystemet til kroppen har disse som akser.

- Få de uekte kreftene forenklet.

$$m \mathbf{a} = \mathbf{F} - \underbrace{m \ddot{\mathbf{r}}}_{\text{aksel.}} - \underbrace{2m \boldsymbol{\omega} \times \dot{\mathbf{u}}}_{\text{Corioluskraft}} - \underbrace{m \dot{\boldsymbol{\omega}} \times \mathbf{u}}_{\text{Eulerkraft}} - \underbrace{m \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{u})}_{\text{sentrifugalkraft}}$$

Disse kommer av at vi har et bevegelig referansesystem hvor \mathbf{u} er en vilkårlig vektor.

- Når \mathbf{r}_E (vektoren til fartøyet i \mathbf{E}) ligger i origo til \mathbf{E} ($\mathbf{r}_E = [0,0,0]$) blir akselerasjonskreftene $m \ddot{\mathbf{r}} = 0$.
- Kroppen er antatt stiv og dermed blir $\dot{\mathbf{u}} = 0$ (vektorene til partiklene endrer seg ikke) som fører til at Coriolus-leddet også blir lik null.
- Forenkle gyro effektene som kommer av rotasjon av kroppen og rotasjonen av propellene
 - Kroppen kommer ikke til å rotere fort nok til at disse betyr noe særlig i vår oppgave med å holde fartøyet i ro; det blir tilnærmet ingen gyroeffekter.
 - Gyroeffektene av propellene er også ment å være ubetydelig i forhold til kreftene propellene skaper når de drar.

Konklusjoner

Den dynamiske modellen som er nødvendig for å regulere kroppens orientering i luften kommer av likning (31). Spaltet opp i hver akse blir det:

$$\begin{aligned}
 I_{xx} \dot{p} &= \underbrace{(I_{yy} - I_{zz}) qr}_{\text{gyroeffekten av kroppen}} + \underbrace{U_2}_{\text{moment om } x\text{-akse}} + \underbrace{q J (-\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4)}_{\text{gyroeffekten av propellene}} \\
 I_{yy} \dot{q} &= \underbrace{(I_{zz} - I_{xx}) pr}_{\text{gyroeffekten av kroppen}} + \underbrace{U_3}_{\text{moment om } y\text{-akse}} + \underbrace{p J (\Omega_1 - \Omega_2 + \Omega_3 - \Omega_4)}_{\text{gyroeffekten av propellene}} \\
 I_{zz} \dot{r} &= \underbrace{(I_{xx} - I_{yy}) pq}_{\text{gyroeffekten av kroppen}} + \underbrace{l(K)}_{\text{moment om } z\text{-akse}}
 \end{aligned}$$

Og når fartøyet skal holde seg i ro og gyroeffektene er sett som null kan disse forenkles til:

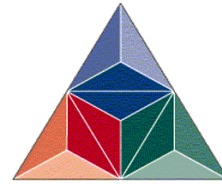
$$\dot{p} = \frac{U_2}{I_{xx}}$$

$$\dot{q} = \frac{U_3}{I_{yy}}$$

$$\dot{r} = \frac{K}{I_{zz}}$$

Referanser

- [1] Joel A. Shapiro, *Classical Mechanics*: Rutgers, the State University of New Jersey, 2010
(hentet 2011-03-23 fra <http://www.physics.rutgers.edu/~shapiro/507/book.pdf>)
- [2] John Chiasson, *Modeling and High-performance Control of Electric Machines*: John Wiley & Sons, Inc., 2005
- [3] Wikipedia, *Rotation matrix*
(hentet 2011-04-06 fra http://en.wikipedia.org/wiki/Rotation_matrix#Ambiguitie)
- [4] MIT OpenCourseWare , *2.017J Design of Electromechanical Robotic Systems* , 2009
(hentet 2011-04-06 fra http://ocw.mit.edu/.../MIT2_017JF09_ch09.pdf)
- [5] Bouabdallah, S., A. Nuechter, and C. Stachniss, *Design and control of quadrotors with application to autonomous flying* :Lausanne Polytechnic University, 2007
- [6] Randal W. Beard , *Quadrotor Dynamics and Control* : Brigham Young University , 2008



HØGSKOLEN
i Buskerud

Oppgavetittel: Hovedprosjekt - Motormodell - v3.0		
Fag(nr./navn): SFHO 3200 / Prosjektstyring med hovedprosjekt		
Gruppemedlemmer, signatur:		
_____ Vegard Torkelsen	_____ Michael Odden	_____ Herå Rørvik
_____ Trine Lindberg	_____ Luke Carambot	_____ Patrick Bjørum
Intern veileder: Dag Samuelsen Intern sensor: Olaf Hallan Graven Ekstern veileder og sensor: Ragnvald Otterlei		
Dato: 2011-03-16		

Innholdsfortegnelse

Dokumenthistorie.....	1
Innledning.....	1
1 Matematiskmodell.....	2
2 Parameterne.....	2
2.1 Likningene til disse motorene.....	2
2.2 Parameter R.....	3
2.3 Parameter f.....	4
Konklusjoner.....	5
Referanser.....	6
Vedlegg 1.....	7

Dokumenthistorie

Versjon	Beskrivelse	Når
1.0	Dokument opprettet	2011-03-16
2.0	Lagt til treghetsmoment og korrekturlest. Forandret del kapitel 2.3	2011-03-17
3.0	Lagt til beregninger av R. Skrev om konklusjon.	2011-05-20

Tabell 1: Dokumenthistorie

Innledning

Motorene som brukes på fartøyet er trefase BLDC (børsteløs DC) motorer. BLDC motorer er av typen PMSM (permanentmagnet synkronisk maskiner) som er AC maskiner med trapesoidisk bak-EMF [1]. Forskjellen fra en vanlig DC motor er at rotoren har permanente magneter og en elektronisk kommentator som sørger for at spenningen kommer på riktig statorvikling på riktig tidspunkt. For at kommentator skal greie dette må den vite rotor posisjon til bestemte tidspunkter. Denne styringskretsen er så nødvendig at de sees som en del av motoren og med den på plass oppfører motoren som en DC motor. Det vil si at hastigheten er proporsjonal med spenningen og moment er proporsjonal med strømmen. Dette er grunnen til at en enklere DC motormodell kan brukes på denne pakken.

Motorene som har blitt kjøpt er uten styringskretsen. Det er vanlig at en bruker én ESC (estimated speed controller) til å styre motoren til RC-modeller. Hoved oppgaven til ESCene er å styre statorspenningene som kommentator. Det er tre vanlige måter å få rotorposisjonen til statorstyringen, enten vha. hallsensorer (går på det magnetiske feltet), enkodere (som vanligvis bruker en lysdiode og en lyssensor) eller bak-EMF. Siden våre motorene har kun tre ledninger til oppkobling til ESCen er det tydelig at motoren har ingen ekstra sensorer og derfor må bruke bak-EMF metoden for å styre kommentatoren. Det er til all tid en statorvikling som ikke er aktiv og kan derfor brukes til å måle av bak-EMFen.

Utifra å se på strømforbruk under kjøring har det blitt testet og bekreftet at motorene er momentstyrt av ESCene, siden strømforbruk gikk opp under økning i belastningen.

1 Matematiskmodell

Om en DC modell blir valgt til å modellere en ESC/motor pakke, er det slik at:

$$L \frac{di(t)}{dt} = U_s(t) - R_A i(t) - K_b \omega_R(t) \quad [\text{V}] \quad (1)$$

$$J \frac{d\omega(t)}{dt} = K_T i(t) - f \omega_R(t) - \tau_L(t) \quad [\text{Nm}] \quad (2)$$

$$\frac{d\theta(t)}{dt} = \omega_R(t) \quad [\text{rad/s}] \quad (3)$$

Definisjoner:

L = rotorinduktans

i = tilført rotorstrøm

U_s = tilført rotorspenning

R_A = rotormotstand (ledere)

K_b = rotasjonsindusertspenning konstant

ω_R = rotorhastighet

J = rotortrehetsmoment

K_T = moment konstant

f = rotorfeltmotstand

τ = rotorlastmoment

θ = rotorposisjon (angular)

2 Parameterne

2.1 Likningene til disse motorene

Ved å se nærmere på de egenskapene til motorene i prosjektet kan det noteres følgende:

- Ettersom motorene har permanente magneter i stator er $K_T = K_b$.
- Utifra «databladet» kan K_b regnes ut til å være lik $\frac{1}{30,8} \left[\frac{Vs}{rad} \right]$
- I stasjonær drift er derivatleddene i (1) og (2), lik null.
- L er lik null (Se vedlagt «datblad»)
- Motorene er av «outrunner» type og betyr at rotor (med magneter) ligger på utside og spinner rundt statorviklingene. Trehetmomentet J har blitt regnet ut til å være lik $5,26 \cdot 10^{-5} [kgm^2]$.

Da kan likningene (1) og (2) reduseres til:

$$0 = U(t) - Ri(t) - K_b \omega_R(t) \quad [\text{V}] \quad (1b)$$

$$0 = K_b i(t) - f \omega_R(t) - \tau_L(t) \quad [\text{Nm}] \quad (2b)$$

De siste parameterne som skal regnes ut er R og f .

2.2 Parameter R

Motstanden i motorpakken kan regnes ut vha. 1b.

$$0 = U(t) - Ri(t) - K_b \omega_R(t) \quad [\text{V}]$$

$$R = \frac{U(t)}{i(t)} - \frac{K_b \omega_R(t)}{i(t)} \quad \Omega \quad (3)$$

Det var mulig å kjøre noen tester hvor motor strømmen ble målt under drift. Spenningen over batteriet var også målt. Hastighetene var målt og motoren var ubelastet. Tabell 2 viser data.

Vs	Vs%	Micros	RPM	Amps	Rad/s
12.41	3.1025	1250	3650	0.21	382.22
	3.723	1300	7750	0.4	811.58
	4.3435	1350	9677	0.52	1013.37
	4.964	1400	10500	0.58	1099.55
	5.584	1450	11200	0.62	1172.86
	6.205	1500	11425	0.64	1196.42
	6.8255	1550	11550	0.66	1209.51
	7.446	1600	11700	0.67	1225.22
	8.0665	1650	11800	0.68	1235.69
	8.687	1700	11950	0.68	1251.40
	9.3075	1750	12100	0.69	1267.10
	9.928	1800	12100	0.69	1267.10
	10.5485	1850	12100	0.69	1267.10
	11.169	1900	12100	0.7	1267.10
	11.7895	1950	12200	0.7	1277.58
	12.41	2000	12200	0.7	1277.58

Tabell 2: Motortest uten last

Testen gir nok opplysninger til å regne ut R ved likning (3) gitt K_b .

$$R = \frac{U(t)}{i(t)} - \frac{K_b \omega_R(t)}{i(t)} \quad \Omega$$

Setter inn data inn i likningen gir R som:

R
-44,3201917
-56,5675
-54,9194680
-52,9925437
-52,4126938
-50,9996956
-49,1580382
-48,2595077
-47,1372135
-46,9748090
-46,1334933
-45,2342180
-44,3349426
-42,8151577
-42,4148145
-41,5283859

Tabell 3:
Beregnet R

2.3 Parameter f

Denne parameter er en del av (2b). Om motoren er kjørt uten last kan likningen forkortes til

$$0 = K_b i - f \omega_R$$

$$f = \frac{K_b i}{\omega_R} \left[\frac{\text{Nms}}{\text{rad}} \right] \quad (4)$$

Ved bruk av den samme data i tabell 2 og K_b kan likningen (4) brukes til å regne ut f .

$$f = \frac{K_{bi}}{\omega_R} \left[\frac{\text{Nms}}{\text{rad}} \right]$$

Konklusjoner

Resultatet på R ser ikke riktig ut. De kan ikke være minus. Derfor blir ingen beregninger gjort på f. Det viser seg vanskelig å finne parameterne til modellen siden direkte kontroll og måling av spenningen er låst i ESCene, og er dermed grunnen til at beregningene på testdata (til og med selv testdata) ikke er riktige.

Andre løsninger må undersøkes.

Referanser

[1] John Chiasson , *Modeling and High-performance Control of Electric Machines* : John Wiley & Sons, Inc. , 2005

Vedlegg 1

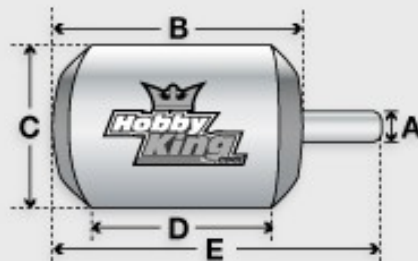
Databladet til motorene



Dimensions: 28x32mm
 Rating: 924kv
 Battery Config: 3 Lipo Cells
 Shaft: 3.175mm
 Weight: 56gr
 Standard Current: 6-14A
 Max Current: 17A
 Suggested Prop: 11x4.7

Product Config Table

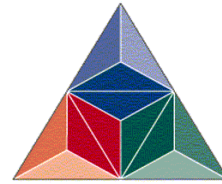
Kv (rpm/v)	924
Weight (g)	56
Max Current (A)	17
Resistance (mh)	0
Max Voltage (V)	11
Power(W)	0
Shaft A (mm)	-
Length B (mm)	32
Diameter C (mm)	28
Can Length D (mm)	17
Total Length E (mm)	46



Try our comparison system.

[COMPARE](#)

[Update/Add my own data](#)
[Customer Data](#)



HØGSKOLEN
i Buskerud

Oppgavetittel:

Hovedprosjekt - Parameterfremstilling - v3.0

Fag(nr./navn):

SFHO 3200 / Prosjektstyring med hovedprosjekt

Gruppemedlemmer, signatur:

Vegard Torkelsen

Michael Odden

Herå Rørvik

Trine Lindberg

Luke Carambot

Patrick Bjørum

Intern veileder: Dag Samuelsen

Intern sensor: Olaf Hallan Graven

Ekstern veileder og sensor: Ragnvald Otterlei

Dato: 2011-05-26

Innholdfortegnelse

Dokumenthistorie.....	1
Innledning.....	1
1 Sammenheng mellom pulsbredde og rotasjonshastighet.....	2
2 Sammenheng mellom pulsbredde og løftekraft.....	3
3 Sammenheng mellom pulsbredde og yaw-kraft.....	4
4 Motoraksellerasjonstest.....	5
4.1 Stroboskop.....	5
4.1.1 Stroboskop program.....	5
4.2 Mikrofon.....	5
4.2.1 Mikrofon program	6
4.3 Resultater.....	6
4.3.1 Stroboskop.....	6
4.3.2 Mikrofon.....	6
4.4 Andvendelse.....	6
5 Skalering av motor 3 – ikke lenger i bruk.....	7
6 Treghetsmoment til fartøyet.....	9
6.1 Vekt og dimensjoner.....	9
6.2 Grunnlegende formelen.....	9
6.2.1 Treghetsmoment for kube.....	9
6.2.2 Treghetsmoment for sylinder.....	9
4.2.3 Treghetesmoment for stav.....	10
6.2.4 Parallell-akse teoremet.....	10
6.3 Treghetsmoment om x-aksen.....	10
6.3.1 Delberegning 1 – Batteri.....	10
6.3.2 Delberegning 2 – Motor 1 og 3.....	10
6.3.3 Delberegning 3 – Motor 2 og 4.....	11
6.3.4 Delberegning 4 – Arm 1 og 2.....	11
6.3.5 Delberegning 5 – Arm 2 og 4.....	11
6.3.6 Delberegning 6 – Boks.....	12
6.3.7 Delberegning 7 – Totalt.....	12
6.4 Treghetsmoment om y-aksen.....	12
6.5 Treghetsmoment om z-aksen.....	12
6.5.1 Delberegning 1 – Batteri.....	12
6.5.2 Delberegning 2 – Motorer.....	13
6.5.3 Delberegning 3 – Armer.....	13
6.5.4 Delberegning 4 – Boks.....	14
6.5.5 Delberegning 5 – Totalt.....	14
6.6 Motorens treghetsmoment.....	14
6.6.1 Delberegning 1 – Treghetsmoment for rotor.....	15
6.6.2 Delberegning 2 – Treghetsmoment for aksel.....	15
6.6.3 Delberegning 3 – Treghetsmoment for propsaver.....	15
6.6.4 Delberegning 4 – Treghetsmoment for propsaver skruer.....	15
6.6.5 Delberegning 5 – Treghetsmoment for propell.....	15
6.6.6 Delberegning 6 – Totalt.....	15
7 Pitch ϕ	16
8 Roll θ	17
Referanser.....	18

Dokumenthistorie

Versjon	Beskrivelse	Når
1.0	Opprettet dokument, lagt inn motorkskalering	2011-02-02
2.0	Oppdatert iht mal	2011-04-08
3.0	Akselerasjonstest, yawtest og treghetsmoment for motor, skrevet innledning og kommentert motorskalering	2011-05-26
4.0	Utfyllende akselerasjonstest	2011-05-27

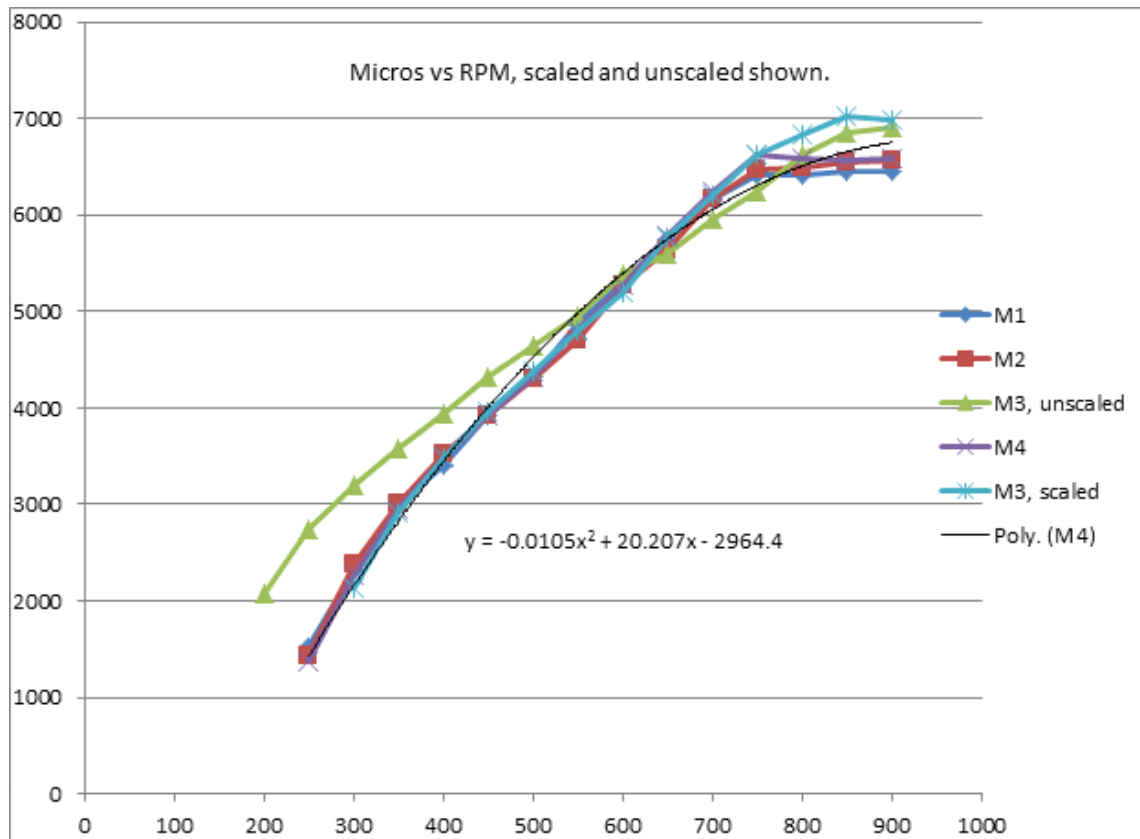
Tabell 1: Dokumenthistorie

Innledning

Dette dokumentet forklarer hvordan gruppen kom frem til verdier til forskjellige parametere som har blitt brukt i simuleringen av reguleringsystemet sammen med den matematiske modellen.

1 Sammenheng mellom pulsbredde og rotasjonshastighet

Gjennom en serie tester[1], har vi målt rotasjonshastigheten til propellene. Disse testene fører til at vi kan utlede en sammenheng mellom pulsbredden inn til ESCene og rotasjonshastigheten til motorene ved hjelp av en grafisk estimering.



Figur 1: Rotasjonshastighet

Kurven som ble estimert er gitt ved:

$$RPM = -0.0105\Delta^2 + 20.207\Delta - 2964.4$$

Her er delta pulsbredden inn til ESCen. For å oversette RPM til rotasjonshastighet gjorde vi følgende.

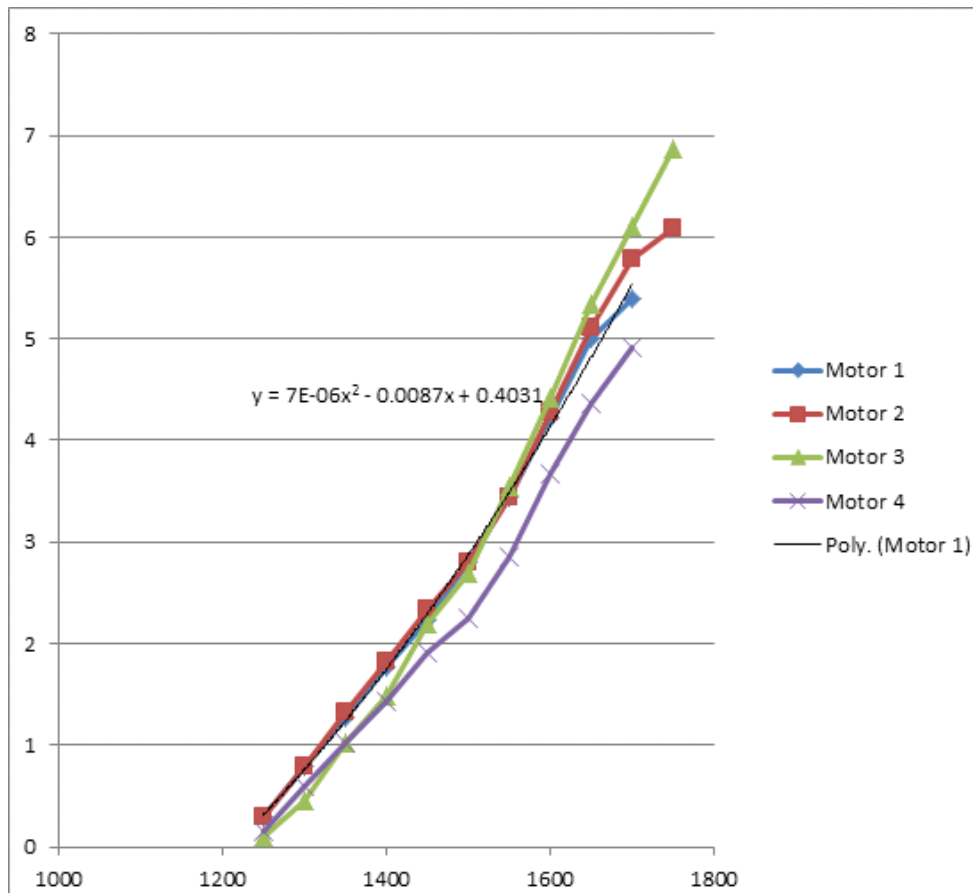
$$r = \frac{rpm}{60 * 2\pi}$$

$$r = \frac{-0.0105\Delta^2 + 20.207\Delta - 2964.4}{60 * 2\pi}$$

Legg merke til at motor 3 har blitt skalert i grafen over, for nærmere informasjon om denne skaleringen, se kapittel 3.

2 Sammenheng mellom pulsbredde og løftekraft

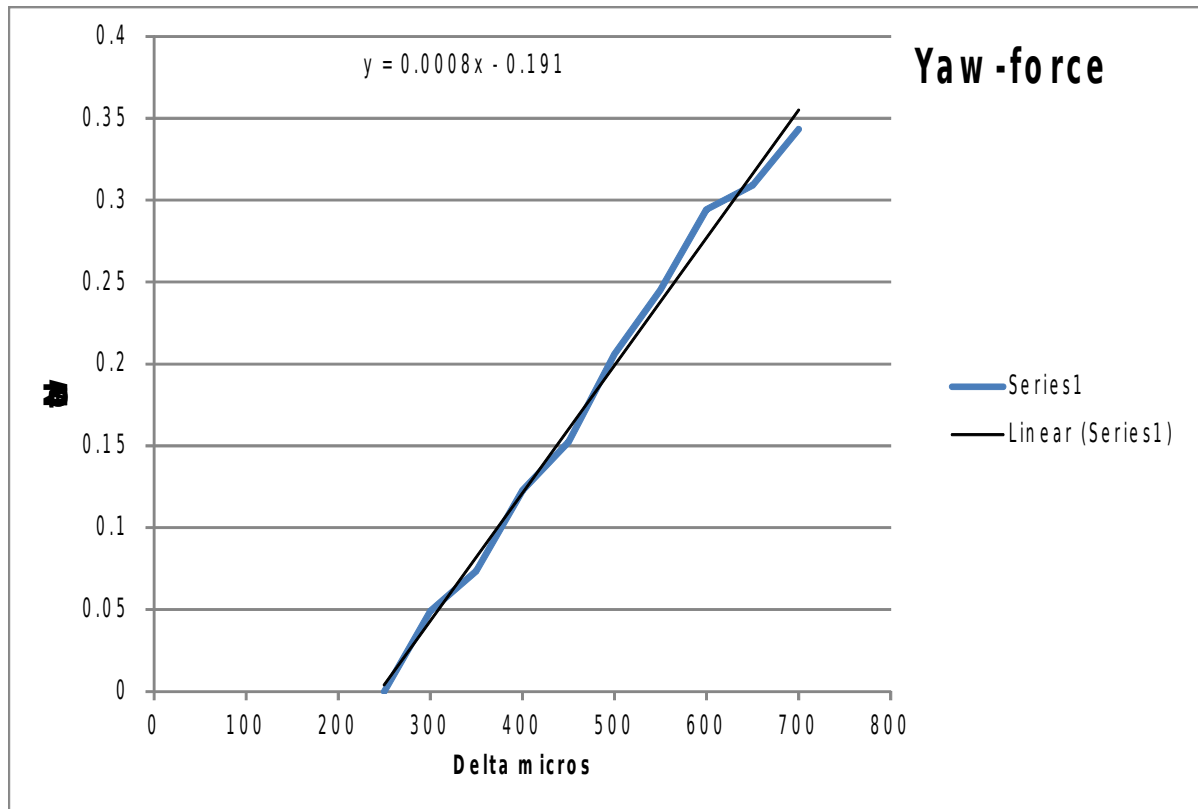
Gjennom en serie tester[2] har vi målt løftekraften i forhold til pulsbredden inn til ESCene. Som i forrige kan vi utlede en sammenheng mellom disse faktorene gjennom en grafisk estimering.



Figur 2: Pulsbredde til løftekraft (x akse er pulsbredde, y aksen er Newton)

3 Sammenheng mellom pulsbredde og yaw-kraft

Slik som tidligere har vi gjort tester[5] for å finne denne kraften.



Figur 3: Pulsbredde til kraft i yaw

Denne figuren viser den kraften en motor skaper i yaw. Her er x-aksen micros over 1000 og y-aksen er Newton. Dette gir oss en ligning vi kan bruke videre, altså;

$$Kraft = 0.0008x - 0.191$$

4 Motoraksellerasjonstest

4.1 Stroboskop

	Beskrivelse
1	Still inn Stroboskopet på frekvensen til slutthastighet (hastigheten etter spranget)
2	Kjør testprogrammet (kjører motoren opp i en bestemt hastighet og venter på knapp for sprang)
3	Trykk knapp (gir sprang i hastigheten) og starte stoppeklokken samtidig
4	Stopp stoppeklokken i det propellen står nesten stille (sett med stroboskop)

Tabell 2: Stroboskopprosedyre

4.1.1 Stroboskop program

La x_1 være startsignalet gitt i mikrosekunder.

La x_2 være signalet etter et sprang, gitt i mikrosekunder.

Programmet deles opp i to deler. Det første er bare for å stille inn stroboskopet på slutthastigheten. Programmet kjører motoren sakte opp til ønsket sprangsignal x_2 . Når stroboskopet er innstilt (så propellen visuelt står stille) kan programmet avsluttes.

Den andre delen er selve testprogrammet. Programmet kjører motorene sakte opp til en bestemt hastighet, x_1 , og holder x_1 inntil et signal inn på port 2 (trykker på knapp) gjør at vi får et sprang til en høyere hastighet, x_2 . Ved å trykke på knappen en gang til stopper programmet.

4.2 Mikrofon

	Beskrivelse
1	Koble opp mikrofon og oscilloscop, samt gjør klar kamera
2	Start testprogrammet, testprogrammet kjører motoren opp til en bestemt hastighet
3	Start filming av scope
4	Trykk knapp, testprogrammet gir et sprang i hastigheten
5	Leser av pulsbreddene på oscilloscopet og regner om til frekvens i excel.

Tabell 3: Mikrofonprosedyre

4.2.1 Mikrofon program

Bruker samme testprogrammet som for stroboskoptesten.

4.3 Resultater

4.3.1 Stroboskop

Det viste seg å være veldig vanskelig å måle tiden ved hjelp av stoppeklokke da akselerasjonstiden er svært liten. Det blir derfor stor feil på grunn av reaksjonsevne. Den eneste konklusjonen vi fikk ut av denne testen er at akselerasjonen må være mindre enn den vi fant av stroboskoptesten.

4.3.2 Mikrofon

Ved å bruke mikrofon var testen lettere å utføre, og det var enklere å måle tiden. Ved å undersøke filmen frame for frame fikk vi greie resultater som vi kan bruke videre. Et høyhastighet kamera kunne gitt enda bedre resultater.

Den resulterende akselerasjonen ble tilnærmet til å være 1400 rad/s^2

4.4 Andvendelse

Denne akselerasjonen skulle brukes videre i simuleringen av reguleringsystemet. Et problem var da at vi ikke hadde mulighet til å strupe akselerasjonen til motorene. Vi måtte derfor bruke dette resultatet for å finne ut hvor fort kreftene som motorene genererer kan endre seg, dette er fordi simuleringen oversetter en pulsbredde til en kraft og ikke en rotasjonshastighet.

For å komme frem til et svar måtte vi anta at kraften som en propell lager ved en viss rotasjonshastighet, er den samme når hastigheten er konstant og når den endrer seg.

Denne antagelsen kommer til å gi svaret vårt en viss feilmargin, denne feilmarginen oppstår pga de aerodynamiske effektene fra rotorbladet, når rotorbladet aksellerer har luftgjennomstrømningen en annen gjennomstrømningsrate en når hastigheten er konstant. Selv om målingen ikke er optimal gir den oss ivhertfall en estimert verdi som vi kan bruke i simuleringen.

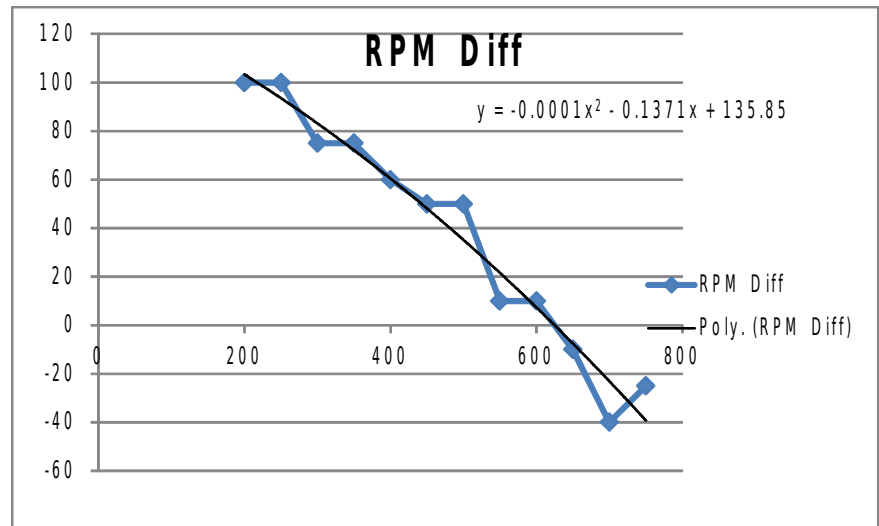
For å komme frem til denne, andvendte vi den tidligere turtalstesten for å finne en estimert pulsbredde ved alle målepunktene under akselerasjonen. Deretter brukte vi disse pulsbreddene sammen med kraft-testene, slik at vi fikk plottet propellens kraft under akselerasjon. På denne måten kunne vi finne kraftakselerasjonen til propellen som ble 15 N/s^2 . [1]

5 Skalering av motor 3 – ikke lenger i bruk

Ved de første testene la vi merke til at motor 3 oppførte seg anderledes en de andre motorene. Vi måtte derfor finne en funksjon som lar oss sørge for at motor 3 har lik rotasjonshastighet som de andre motorene ved likt pådrag.

Vi tok utgangspunkt i grafen og så hvilken pulsbredde motor 3 burde ha for å få lik rotasjonshastighet som de andre motorene. Se figur 4. til høyre.

Under kolonnen Current i tabell 4, ser man den pulsbredden som går inn i ESCen, under kolonnen Optimal, er pulsbredden vi må ha for at motorene skal ha lik rotasjonshastighet, mens Diff er forskjellen mellom de to.



Figur 4: Rotasjonshastighet

Deretter tegnet vi en graf som sammenlignet Current og Diff, Figur 4.

Current	Optimal	Diff
200	100	100
250	150	100
300	225	75
350	275	75
400	340	60
450	400	50
500	450	50
550	540	10
600	590	10
650	660	-10
700	740	-40
750	775	-25
800	775	25
850	775	75
900	775	125

Tabell 4: Skalering

Fra denne grafen i Figur 4 kunne vi estimere en funksjon som gir oss den rotasjonshastigheten vi ønsker.

Når funksjonen var utviklet implementerte vi den og foretok en ny måling.

Tabell 3. viser at skaleringen fungerer slik som sett i Figur 1.

Denne skaleringen er ikke lenger i bruk, da vi fikk identifisert faktoren som forårsaket forskjellen i turtall på motorene. Etter å ha redefinert ESCenes øvre og nedre inngangsverdier oppførte alle motorene seg likt.

Motor 3		
micros	frekvens	RPM
200		
250		
300	71.2	2136
350	97	2910
400	116	3480
450	132	3960
500	146	4380
550	160	4800
600	173	5190
650	192	5760
700	207	6210
750	221	6630
800	228	6840
850	234	7020
900	233	6990

Tabell 1: Skalert motor

6 Trehetsmoment til fartøyet

Det er nødvendig og finne trehetsmomentet for fartøyet rundt de gitte aksene. Dette er fordi trehetsmomentet gir en sammenheng mellom den påtrykte kraften og fartøyet bevegelser.

For å kunne beregne dette trenger man en mengde data.

6.1 Vekt og dimensjoner

	Bredde (m)	Høyde(m)	Lengde(m)	Radius(m)	Vekt(kg)
Batteri	0.085	0.018	0.085		0.306
Motor		0.046		0.014	0.077
Armer	0.012	0.008	0.325		0.034
Boks	0.125	0.07	0.195		0.0856

Tabell 2: Vekt og dimensjoner på fartøyet

6.2 Grunnleggende formelen

For å kunne beregne trehetsmomentene skaffer vi oss først en oversikt over de formlene vi kommer til å bruke.

6.2.1 Trehetsmoment for kube

$$I_h = \frac{1}{12} m(w^2 + d^2)$$

$$I_w = \frac{1}{12} m(h^2 + d^2)$$

$$I_d = \frac{1}{12} m(w^2 + h^2)$$

6.2.2 Trehetsmoment for sylinder

$$I_z = \frac{mr^2}{2}$$

$$I_x = I_y = \frac{1}{12} m(3r^2 + h^2)$$

4.2.3 Treghetesmoment for stav

$$I_{end} = \frac{mL^2}{3}$$

6.2.4 Parallell-akse teoremet

$$I_d = I + md^2$$

6.3 Treghetsmoment om x-aksen

x-aksen er den aksen som er normal på fremover/bakover retning. Ved å bruke dataene og formlene ovenfor kan vi beregne treghetsmomentet for enkeltkomponentene og deretter finne det totale treghetsmomentet rundt denne aksen.

6.3.1 Delberegning 1 – Batteri

$$I_w = \frac{1}{12}m(h^2 + d^2)$$

$$I_w = \frac{1}{12}0.306(0.018^2 + 0.085^2)$$

$$\underline{\underline{I_w = 0.0001925 \text{ kgm}^2}}$$

6.3.2 Delberegning 2 – Motor 1 og 3

$$I_x = \frac{1}{12}m(3r^2 + h^2)$$

$$I_x = \frac{1}{12}0.077(3 * 0.014^2 + 0.046^2)$$

Siden motorene sitter et stykke unna massesenteret må vi anvende parallell akse teoremet.

$$I_d = I + md^2$$

$$I_d = \frac{1}{12}0.077(3 * 0.014^2 + 0.046^2) + 0.077 * 0.325^2$$

$$\underline{\underline{I_x = 0.00815 \text{ kgm}^2}}$$

6.3.3 Delberegning 3 – Motor 2 og 4

$$I_x = \frac{1}{12} m(3r^2 + h^2)$$

$$I_x = \frac{1}{12} 0.077(3 * 0.014^2 + 0.046^2)$$

$$\underline{I_x = 1.73 * 10^{-5} \text{ kgm}^2}$$

6.3.4 Delberegning 4 – Arm 1 og 2

$$I_w = \frac{1}{12} m(h^2 + d^2)$$

$$I_w = \frac{1}{12} 0.034(0.008^2 + 0.325^2)$$

Igjen må vi anvende parallell akse teoremet.

$$I_w = \frac{1}{12} 0.034(0.008^2 + 0.325^2) + 0.034 \left(\frac{0.325}{2} \right)^2$$

$$\underline{I_w = 0.000898 \text{ kgm}^2}$$

6.3.5 Delberegning 5 – Arm 2 og 4

$$I_w = \frac{1}{12} m(h^2 + d^2)$$

$$I_w = \frac{1}{12} 0.034(0.008^2 + 0.012^2)$$

$$\underline{I_w = 5.89 * 10^{-7} \text{ kgm}^2}$$

6.3.6 Delberegning 6 – Boks

$$I_w = \frac{1}{12} m(h^2 + d^2)$$

$$I_w = \frac{1}{12} 0.856(0.07^2 + 0.125^2)$$

$$\underline{I_w = 0.00146 \text{ kgm}^2}$$

6.3.7 Delberegning 7 – Totalt

Nå når vi har beregnet treghetsmomentet for hver komponent kan vi til slutt legge de sammen og finne det totale treghetsmomentet om x-aksen.

$$I_x = 0.0001925 \text{ kgm}^2 + 2 * 0.00815 \text{ kgm}^2 + 2 * 1.73 * 10^{-5} \text{ kgm}^2 + 2 * 0.000898 \text{ kgm}^2 \\ + 2 * 5.89 * 10^{-7} \text{ kgm}^2 + 0.00146 \text{ kgm}^2 = 0.00146 \text{ kgm}^2$$

$$\underline{I_x = 0.2517 \text{ kgm}^2}$$

6.4 Treghetsmoment om y-aksen

Vi gjør de samme beregningene som for Y-aksen, bare med en endring i dimensjonene på boksen og kommer frem til

$$\underline{I_y = 0.0268 \text{ kgm}^2}$$

6.5 Treghetsmoment om z-aksen**6.5.1 Delberegning 1 – Batteri**

$$I_h = \frac{1}{12} m(w^2 + d^2)$$

$$I_h = \frac{1}{12} 0.306(0.085^2 + 0.085^2)$$

$$\underline{I_h = 0.000368 \text{ kgm}^2}$$

6.5.2 Delberegning 2 – Motorer

$$I_z = \frac{mr^2}{2}$$

$$I_z = \frac{0.077 * 0.014^2}{2}$$

Siden motorene sitter et stykke unna massesenteret må vi anvende parallell akse teoremet.

$$I_z = \frac{0.077 * 0.014^2}{2} + 0.077 * 0.325^2$$

$$\underline{I_z = 0.00814 \text{ kgm}^2}$$

6.5.3 Delberegning 3 – Armer

$$I_d = \frac{1}{12}m(w^2 + h^2)$$

$$I_d = \frac{1}{12}0.034(0.008^2 + 0.325^2)$$

Parallell akse teoremet.

$$I_d = \frac{1}{12}0.034(0.008^2 + 0.325^2) + 0.034 \left(\frac{0.325}{2}\right)^2$$

$$\underline{I_d = 0.0012 \text{ kgm}^2}$$

6.5.4 Delberegning 4 – Boks

$$I_d = \frac{1}{12}m(w^2 + h^2)$$

$$I_d = \frac{1}{12}0.856(0.125^2 + 0.195^2)$$

$$\underline{\underline{I_d = 0.00382 \text{ kgm}^2}}$$

6.5.5 Delberegning 5 – Totalt

Vi kan igjen legge sammen alle faktorene og få det totale treghetsmomentet for z-aksen.

$$I_z = 0.000368 \text{ kgm}^2 + 4 * 0.00814 \text{ kgm}^2 + 4 * 0.0012 \text{ kgm}^2 + 0.00382 \text{ kgm}^2$$

$$\underline{\underline{I_z = 0.0415 \text{ kgm}^2}}$$

6.6 Motorens treghetsmoment

I et forsøk på å identifisere noen av motorens parametere slik som kb, var det nødvendig å regne ut treghetsmomentet til motoren.

Del	Måling	SI
Rotorens ytre diameter	27.8	mm
Rotor veggens tykkelsen	2.4	mm
Rotor høyde	19.9	mm
Aksel diameter	3	mm
Aksel lengde	46.45	mm
Aksel vekt	0.003	kg
Rotor vekt	0.026	kg
Total vekt	0.029	kg
Propsaver diameter	10.4	mm
Propsaver høyde	8	mm

propsaver indre diameter	3	mm
Propsaver vekt	0.001	kg
Skruens diameter	2.9	mm
Skruens lengde	11.5	mm
Vekt, propsaver og skurer	0.0005	kg
Propellens lengde	25	cm
Propellens bredde	2.5	cm
Propellens vekt	0.01	kg

Tabell 3: Vekt og dimensjoner for motor

6.6.1 Delberegning 1 – Trehetsmoment for rotor

$$(1/2)*0.026*((25.4*10^{-3})^2)+((27.8*10^{-3})^2)=\underline{0.000018434\text{ kgm}^2}$$

6.6.2 Delberegning 2 – Trehetsmoment for aksel

$$\frac{(0.003*(1.5*10^{-3})^2)}{2}=\underline{3.375*10^{-9}\text{ kgm}^2}$$

6.6.3 Delberegning 3 – Trehetsmoment for propsaver

$$\frac{1}{2}*0.001*((1.5*10^{-3})^2+(5.2*10^{-3})^2)=\underline{1.46*10^{-8}\text{ kgm}^2}$$

6.6.4 Delberegning 4 – Trehetsmoment for propsaver skruer

$$\frac{(0.0005*(11.5*10^{-3})^2)}{3}+0.0005*(1.5*10^{-3})^2=\underline{2.316*10^{-8}\text{ kgm}^2}$$

6.6.5 Delberegning 5 – Trehetsmoment for propell

$$\left(\frac{1}{12}\right)*0.01*(0.25^2+0.025^2)=\underline{5.26*10^{-5}\text{ kgm}^2}$$

6.6.6 Delberegning 6 – Totalt

$$I_{motor}=1.8434*10^{-5}+3.375*10^{-9}+1.46*10^{-8}+2.316*10^{-8}+5.26*10^{-5}=\underline{\underline{5.26*10^{-5}\text{ kgm}^2}}$$

7 Pitch ϕ

Pitch er forårsaket av en forskjell i løftkraft mellom den fremre motoren (motor 1) og den bakre motoren (motor 3). Her skal vi utlede en formel for vinkelakselerasjon i pitch.

Sammenhengen mellom rotasjonsakselerasjon(α), treghetsmoment (I) og dreimoment(τ) er forklart ved følgende formel. [3]

$$\tau = I\alpha$$

$$\tau = Fs$$

$$\tau = (F_{M1} - F_{M3})s$$

$$(F_{M1} - F_{M3})s = I\alpha$$

$$\frac{(F_{M1} - F_{M3})s}{I} = \alpha$$

Ved å bruke denne sammen med de tidligere utregnede verdiene kommer vi frem til en samlet formel.

$$\frac{((0.000007\Delta_{M1}^2 - 0.0087\Delta_{M1}) - (0.000007\Delta_{M3}^2 - 0.0087\Delta_{M3}))0.325}{0.0268} = \alpha_\phi$$

$$\frac{(0.000007\Delta_{M1}^2 - 0.000007\Delta_{M3}^2 - 0.0087\Delta_{M1} + 0.0087\Delta_{M3})0.325}{0.0268} = \alpha_\phi$$

$$((0.000007(\Delta_{M1}^2 - \Delta_{M3}^2) - 0.0087(\Delta_{M1} - \Delta_{M3}))12.1268 = \alpha_\phi$$

Siden vi ønsker en balanse i den totale oppdriften vet vi at vi må ha en sammenheng mellom de to motorene. Dette vil si at hvis vi ønsker å øke motor 1 med f.eks 100 μ s må vi senke motor 3 med den samme verdien. Dette gir oss en mulighet til å eliminere en av motorene slik at vi kun sitter igjen med en input og en output.

Siden Pitch er nøytral ved 1500 μ s får vi følgende likning for denne balansen.

$$\Delta_{M3} = 3000 - \Delta_{M1}$$

Vi kan da sette denne inn i den tidligere formelen.

$$((0.000007(\Delta_{M1}^2 - (3000 - \Delta_{M1})^2) - 0.0087(\Delta_{M1} - (3000 - \Delta_{M1})))12.1268 = \alpha_\phi$$

Ved å forenkle denne likningen ender vi opp med.

$$\ddot{\phi} = 0.298\Delta_{M1} - 447$$

8 Roll θ

Kalkuleringen for roll vil være den samme som for pitch, bare med noen små endringer i treghetsmomentet.

$$((0.000007(\Delta_{M2}^2 - \Delta_{M4}^2) - 0.0087(\Delta_{M2} - \Delta_{M4}))16.422 = \alpha_\theta$$

Vi kan igjen eliminere en av motorene slik som vi gjorde i pitch.

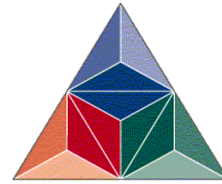
$$((0.000007(\Delta_{M2}^2 - (3000 - \Delta_{M2})^2) - 0.0087(\Delta_{M2} - (3000 - \Delta_{M2})))16.422 = \alpha_\theta$$

Dette forenkles til:

$$\ddot{\theta} = 0.404\Delta_{M2} - 606$$

Referanser

- [1] Testresultater_turtallstest_audio.xlsx, turtallstest.xlsx og turtallstest2.xlsx
- [2] motor_test_2.0.xlsx
- [3] Young and Freedman, University Physics
- [4] MotorAksellerasjonstest.xlsx
- [5] yawTest.xlsx



HØGSKOLEN
i Buskerud

Oppgavetittel: Hovedprosjekt - Sensorfusjon - v2.0		
Fag(nr./navn): SFHO 3200 / Prosjektstyring med hovedprosjekt		
Gruppemedlemmer, signatur:		
_____ Vegard Torkelsen	_____ Michael Odden	_____ Herå Rørvik
_____ Trine Lindberg	_____ Luke Carambot	_____ Patrick Bjørum
Intern veileder: Dag Samuelsen Intern sensor: Olaf Hallan Graven Ekstern veileder og sensor: Ragnvald Otterlei		
Dato: 2011-04-05		

Innholdsfortegnelse

Dokumenthistorie.....	1
Innledning.....	1
1 Komplementærfilteret.....	2
1.1 Teoretisk testdata.....	2
1.2 Fysisk Testdata.....	3
2 Kalman filter.....	3
2.1 Kalman filter, fysisk testdata.....	4
Konklusjoner.....	4
Referanser.....	5

Dokumenthistorie

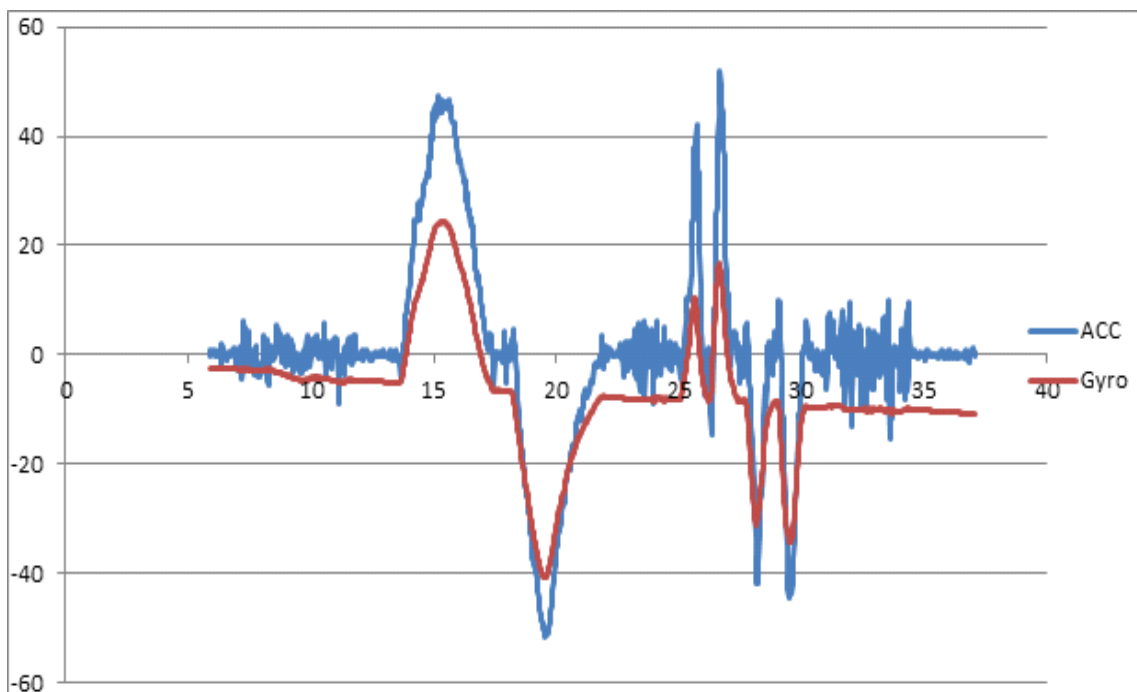
Versjon	Beskrivelse	Når
1.0	Dokument opprettet Rettskrevet og ferdigstilt	2011-04-07
2.0	Lagt til kalmanfilter	2011-05-04

Tabell 1: Dokumenthistorie

Innledning

En av utfordringene i oppgaven er å skaffe pålitelige målinger til vinklene i pitch og roll, problemet er at akselerometeret er kun pålitelig ved rolige bevegelser og det er veldig ømfintlig til støy. Samtidig vil gyroen gi oss en rotasjonshastighet som vi må integrere for å komme frem til vinkelen. Denne integrasjonen fører til at alle avrundingsfeil blir ført videre til neste beregning og akkumulerer. Dette er også kjent som gyro drift.

Vi måtte finne en måte å slå sammen disse målingene slik at vi fikk målinger med så litt støy som mulig og så lite drift som mulig.



Figur 1: Ubehandlet måledata

1 Komplementærfilteret

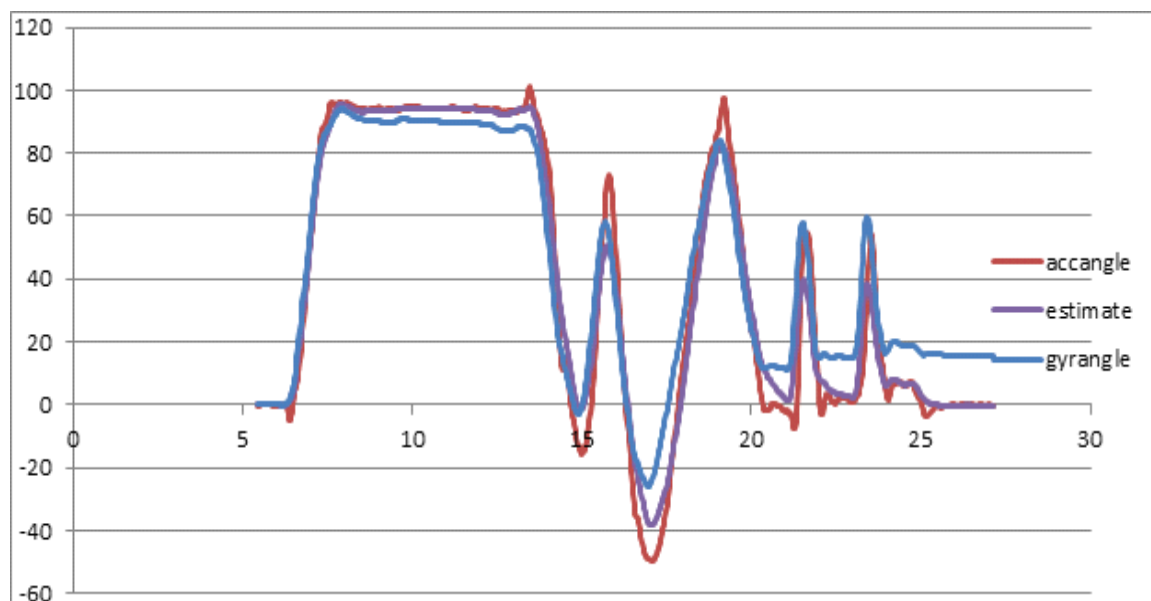
Et av filterne vi har testet er et komplementærfilter.

$$\text{estimertvinkel} = \frac{(w * (\text{estimertvinkel} + \text{gyrorate} * dt))}{\text{Høy pass filter av gyro vinkel}} + \frac{((1 - w) * \text{accvinkel})}{\text{lav pass filter av acc vinkel}}$$

Filteret består av et høypassfilter og et lavpassfilter. For å få det ønskede resultatet sender vi dataen fra akselerometeret gjennom et lavpassfilter slik at vi får kvittet oss med uønsket støy. Vi sender også målingene fra gyroen gjennom et høypassfilter slik at får dempet gyroens drift. Vektingen lar oss finne en balanse mellom støy og drift, høy vekting på akselerometeret gir mer støy men mindre drift, mens en høy vekting på gyroen gir oss mindre støy men mere drift.

1.1 Teoretisk testdata

Før implementeringen ønsket vi å se den teoretiske estimerte vinkelen vi fikk fra filteret. Vi brukte da sensorene for å måle gyroens vinkelhastigheten og akselerometerets vinkel, samt at vi hentet ut tiden mellom hver måling. Med vekten w som 0.96 fikk vi følgende resultat. [1]



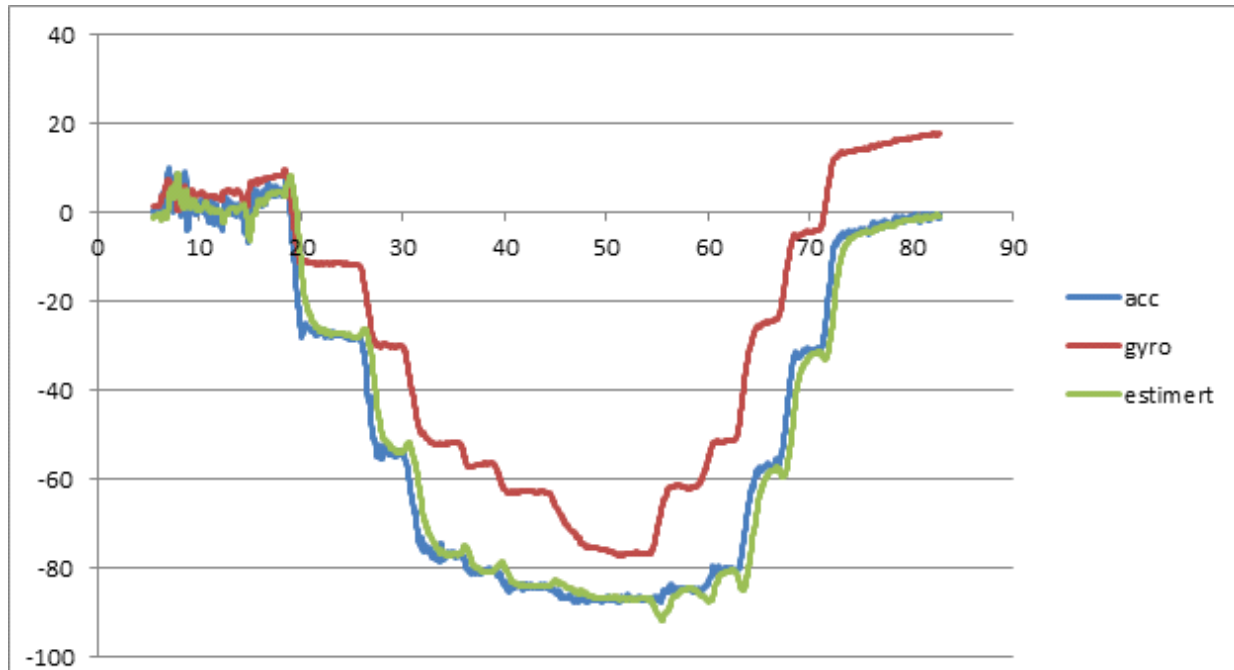
Figur 2: Teoretisk testdata, komplementærfilter

Her ser vi akselerometerets vinkel som den røde linjen, vinkelen fra gyroens integrerte rotasjonshastighet som den blå linjen og den estimerte vinkelen som den lilla linjen.

Legg merke til at den estimerte vinkelen inneholder lite støy og tilsynelatende ingen drift.

1.2 Fysisk Testdata

Etter at filterets funksjon var bekreftet implementerte vi filteret i mikrokontrolleren og gjorde en ny test. [1]



Figur 3: Fysisk testdata, komplementærfilter

Her ser man akselerometers vinkel som den blå linjen, gyroens vinkel fra den integrerte rotasjonshastigheten som den røde linjen og den estimerte vinkelen som den grønne linjen. Vi ser tydelig at filteret gir oss en estimert vinkel med kraftig redusert støy og tilnærmet ingen drift.

En ulempe er at filteret skaper en liten tidsforsinkelse, men dette er en feil det er vanskelig å komme seg unna. En annen ulempe vi ser er at den estimerte verdien gjør noen relativt små, oversving ved hurtige endringer i vinkelen. Grunnet disse feilene ønsket vi i tillegg å teste et kalman-filter

2 Kalman filter

Kalman filteret er et avansert filter som bruker forrige estimerte verdi og den estimerte feilen til denne verdien, for å finne den neste verdien.[2]

$$\hat{X}_k = K_k * Z_k + (1 - K_k) * (X_{k-1})$$

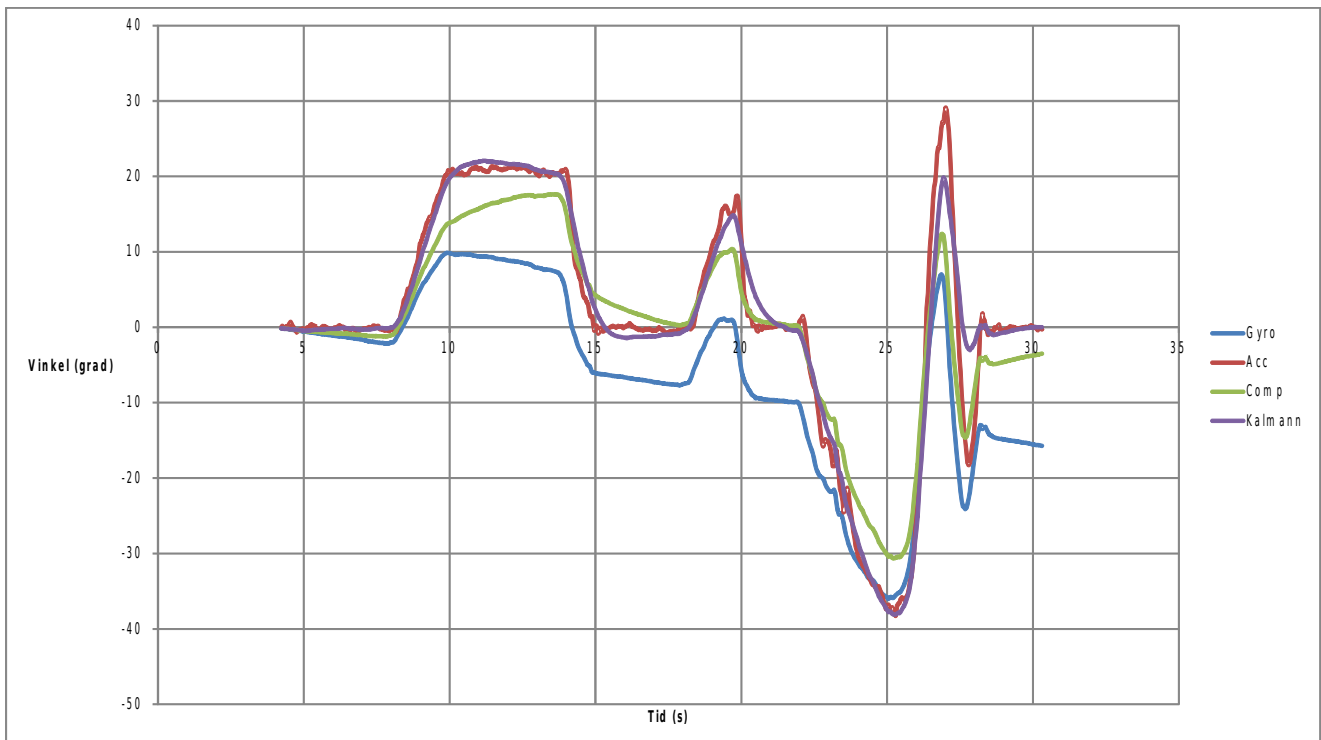
- X_k : Estimert vinkel
- K_k : Kalman Gain
- Z_k : Målt verdi

- X_{k-1} : Forrige estimerte verdi

K_k , kalman gain finnes ved hjelp av avanserte matriseregninger. Vi har ingen forkunnskaper innen dette filteret, men vi fant en kode som viste seg å fungere bra med vårt prosjekt.[3]

2.1 Kalman filter, fysisk testdata.

Vi gjorde en ny test hvor vi brukte både komplementærfilteret og kalmanfilteret samtidig, slik at vi enkelt kunne sammenligne de.



Figur 4: Kalmanfilter, komplementærfilter, gyrocvinkel og accellerometervinkel

I grafen over ser vi de nødvendige dataene for å kunne se effekten av filteret. Hvis vi sammenligner komplementærfilter (grøn linje) og kalmanfilteret (lilla linje) kan vi se at kalman filteret er både hurtigere og inneholder mindre drift enn det komplementærfilteret gjør. Støynivået på de to filterene er tilnærmet like. Se [1] for fullstendig testdata.

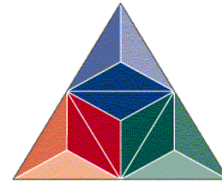
En ulempe med kalmanfilteret er at det bruker noe mer prosessorhastighet enn komplementærfilteret, men måledataens kvalitet er uten tvil verdt dette.

Konklusjoner

De viser seg at komplementærfilteret gir relativt gode resultater, men ikke ved sammenligning med kalmanfilteret. Vi velger derfor å implementere og bruke kalmanfilteret i reguleringssystemet.

Referanser

- [1] <http://git.projectdragonfly.no:3000/documents/56>
- [2] <http://bilgin.esme.org/BitsBytes/KalmanFilterforDummies.aspx>
- [3] http://www.x-firm.com/?page_id=191



HØGSKOLEN
i Buskerud

Oppgavetittel: Hovedprosjekt - Simulering - v1.0		
Fag(nr./navn): SFHO 3200 / Prosjektstyring med hovedprosjekt		
Gruppemedlemmer, signatur:		
_____ Vegard Torkelsen	_____ Michael Odden	_____ Herå Rørvik
_____ Trine Lindberg	_____ Luke Carambot	_____ Patrick Bjørum
Intern veileder: Dag Samuelsen Intern sensor: Olaf Hallan Graven Ekstern veileder og sensor: Ragnvald Otterlei		
Dato: 2011-05-25		

Innholdsfortegnelse

Dokumenthistorie.....	1
Definisjoner og forkortelser.....	1
Innledning.....	1
1 Simulering i MatLab.....	2
1.1 Blokkdiagrammer.....	2
1.1.1 Hele systemet.....	2
1.1.2 Radio.....	4
1.1.3 Signal transformer.....	4
1.1.4 Angle regulator.....	6
1.1.5 Rotation regulator.....	7
1.1.6 Signal mix.....	7
1.1.7 Propulsion system.....	9
1.1.8 Flight dynamics.....	9
1.1.9 Accelerometer.....	11
1.1.10 Gyro	11
1.1.11 Angle estimator(Kalman filter).....	11
1.2 m-fil.....	12
2 Resultater.....	13
2.1 Gyro modus.....	13
2.1.1 Pitch/Roll - oversving:	13
2.1.2 Pitch/Roll – oscillasjoner.....	13
2.1.3 Pitch/Roll – best:.....	14
2.1.4 Yaw:.....	14
2.2 Stabil modus.....	15
2.2.1 Pitch/Roll angle regulator:	15
Konklusjoner.....	16
Referanser.....	17

Dokumenthistorie

Versjon	Beskrivelse	Når
1.0	Dokument opprettet Nytt design lagt til Lagt til definisjoner og forkortelser, oppdatert figurer Oppdatert simuleringene og grafene Korrekturlest	2011-05-25

Tabell 1: Dokumenthistorie

Definisjoner og forkortelser

PID	Proporsjonal-integral-derivat regulator
Pitch	Rotasjon rundt lengden av fartøyet
Roll	Rotasjon rundt bredden av fartøyet
Yaw	Rotasjon rundt sin egen akse
ESC	Estimated Speed Controller

Tabell 2: Definisjoner og forkortelser

Innledning

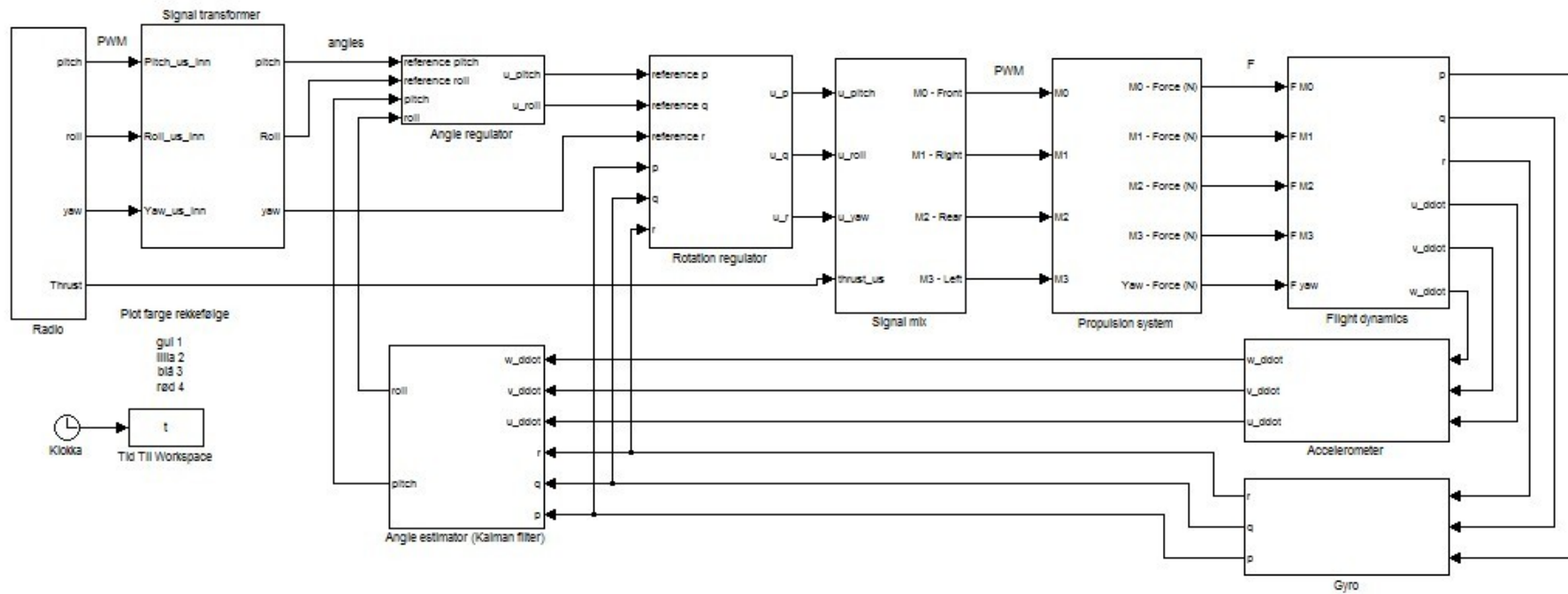
Dette er et dokument om simulering av reguleringsystemet. Simulering er gjort for å finne PID-parametrene som et utgangspunkt til tuning av regulatorene.

1 Simulering i MatLab

1.1 Blokkdiagrammer

1.1.1 Hele systemet

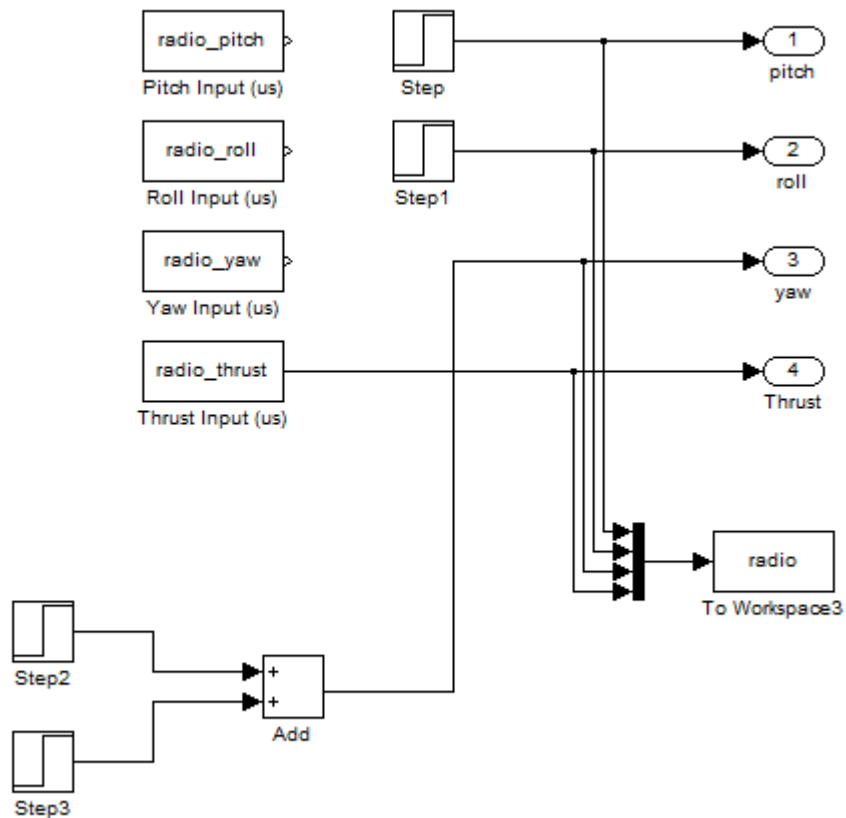
Figuren på neste side viser det overordnede blokkdiagrammet for systemet. Her er systemet delt opp i 10 blokker, hver av blokkene forklares nærmere i de følgende kapitlene. Systemet er tegnet i simulink i MatLab[1].



Figur 1: Blokkdiagram

1.1.2 Radio

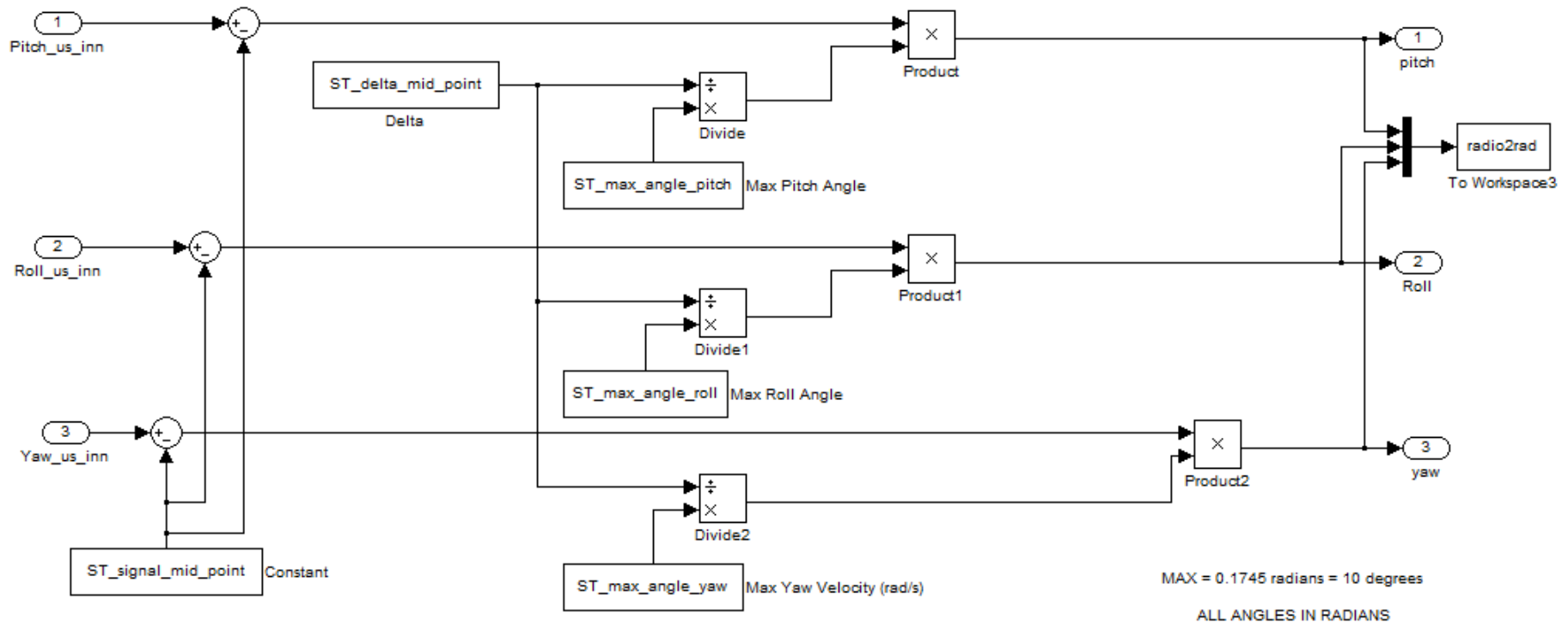
Radioblokka brukes for å simulere input fra radioen. Kan sette forskjellige inngangssignaler som for eksempel et step og analysere hvordan fartøyet reagerer.



Figur 2: Radio

1.1.3 Signal transformer

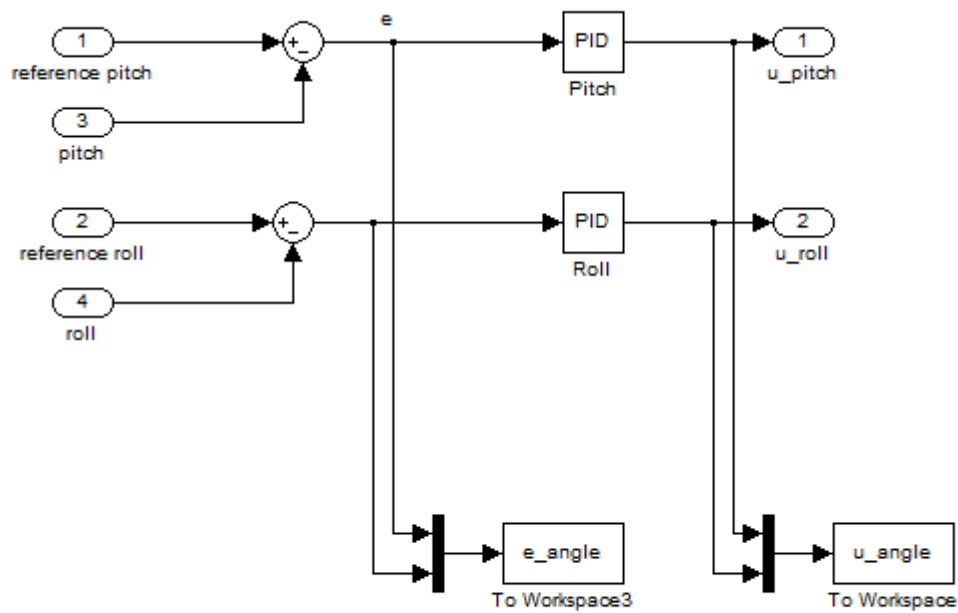
Signal transformeren oversetter inngangssignalenes pulsbredde til en vinkel eller vinkelhastighet i radianer. Dette gjør vi fordi vi får radianer fra sensorene. Her velges også den maksimale og minimale vinkelen vi ønsker i pitch og roll, samt den maksimale vinkelhastigheten vi ønsker i yaw.



Figur 3: Signaltransformer

1.1.4 Angle regulator

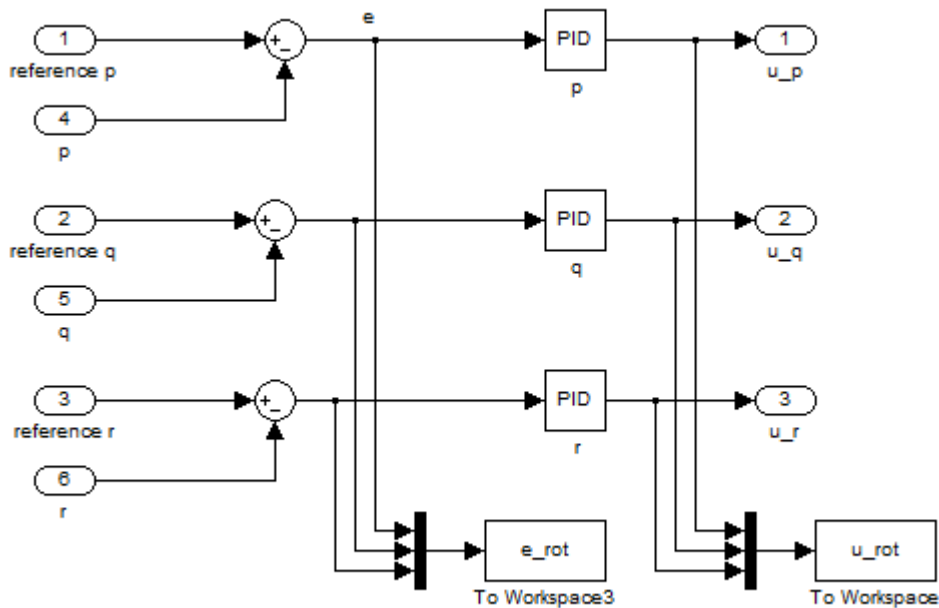
Vinkelregulatoren sammenligner referansevinklene fra fjernkontrollen i roll og pitch med vinklene fartøyet har ifølge sensorene (gjennom Kalman filteret). Den inneholder to PIDer som gir pådrag etter hvor stor forskjellen er.



Figur 4: Angle regulator

1.1.5 Rotation regulator

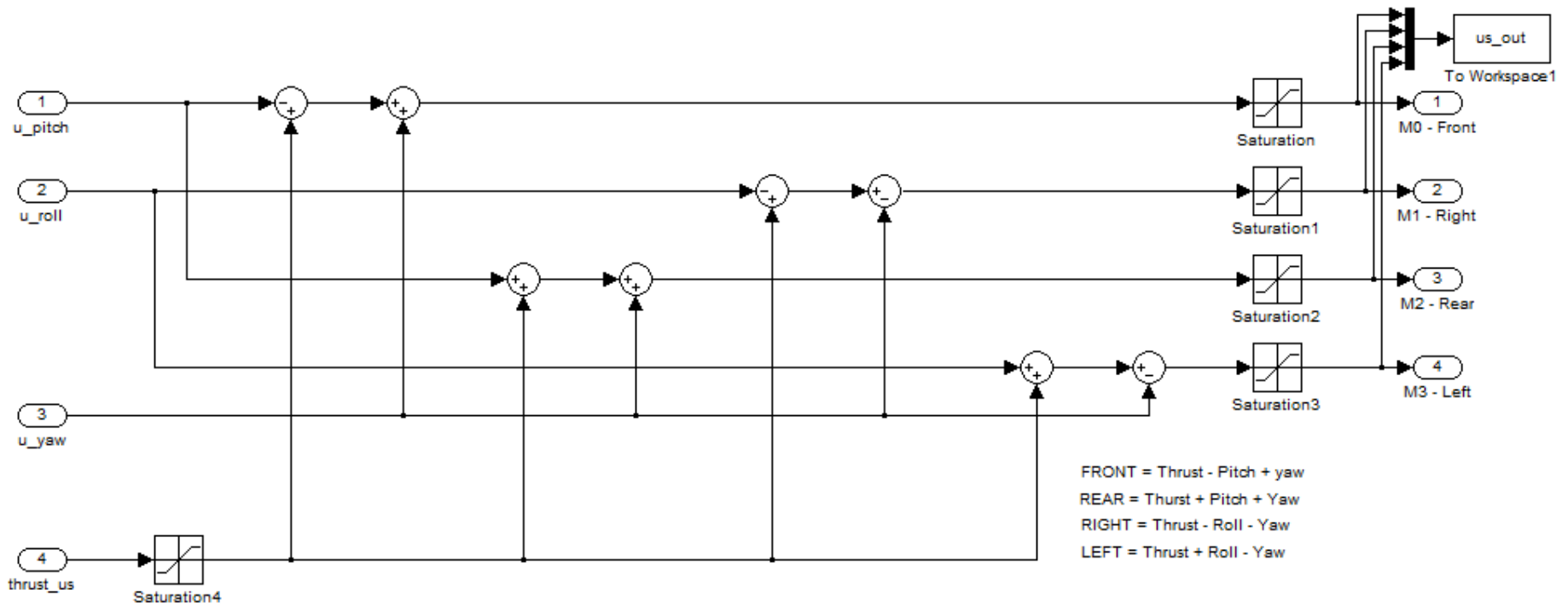
Rotasjonsregulatoren bruker utgangene fra vinkelregulatoren som referanser for vinkelhastighet i roll og pitch. Referansen til yaw kommer direkte fra fjernkontrollen. Vinkelhastighetene sammenlignes med verdiene fra gyroen i hver av aksene, og PIDene gir pådrag dersom vi har en forskjell i en referanse og målt gyroverdi.



Figur 5: Rotation regulator

1.1.6 Signal mix

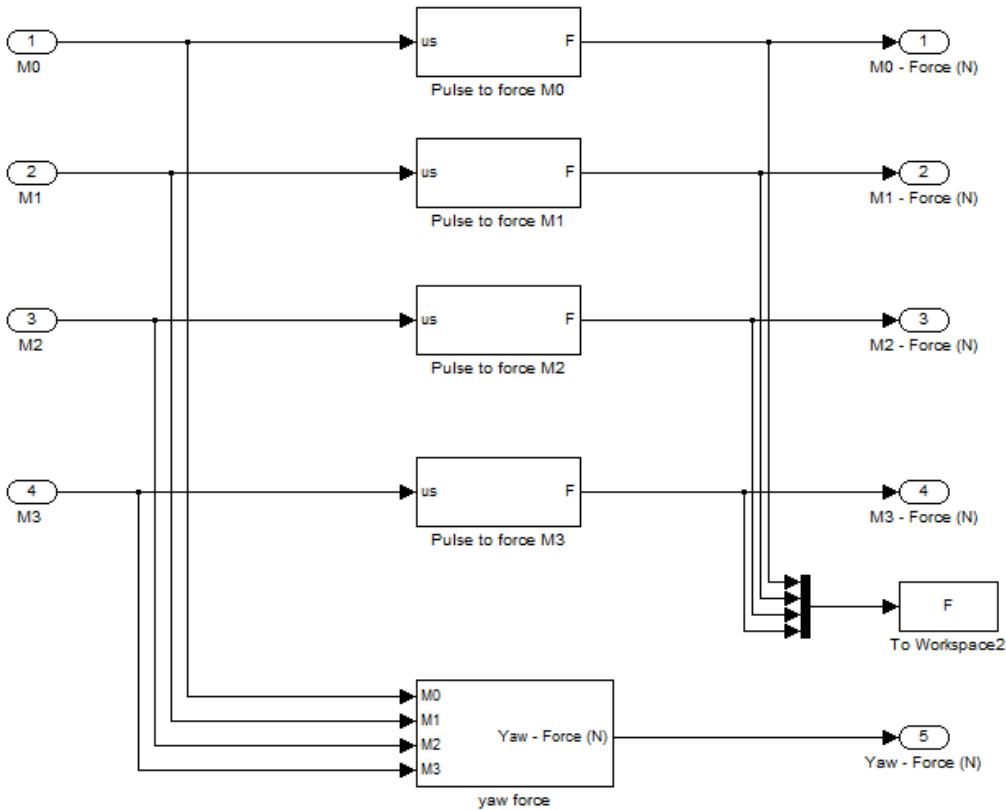
Signal mix blokk inneholder motorligningene som sørger for at riktige motorer får riktig pådrag, slik at vi får de bevegelsene vi ønsker i roll, pitch og yaw. Denne blokk inneholder saturation-blokker før hver av motorene siden ESCene kun opererer i et gitt signalområde. Den inneholder også en saturation på thrust, slik at vi er sikre på at alle motorene ikke går til maks og vi da ikke har mulighet til å endre vinkel. Dette sikrer at PIDene alltid har mulighet for å korrigere vinkelen.



Figur 6: Signal mix

1.1.7 Propulsion system

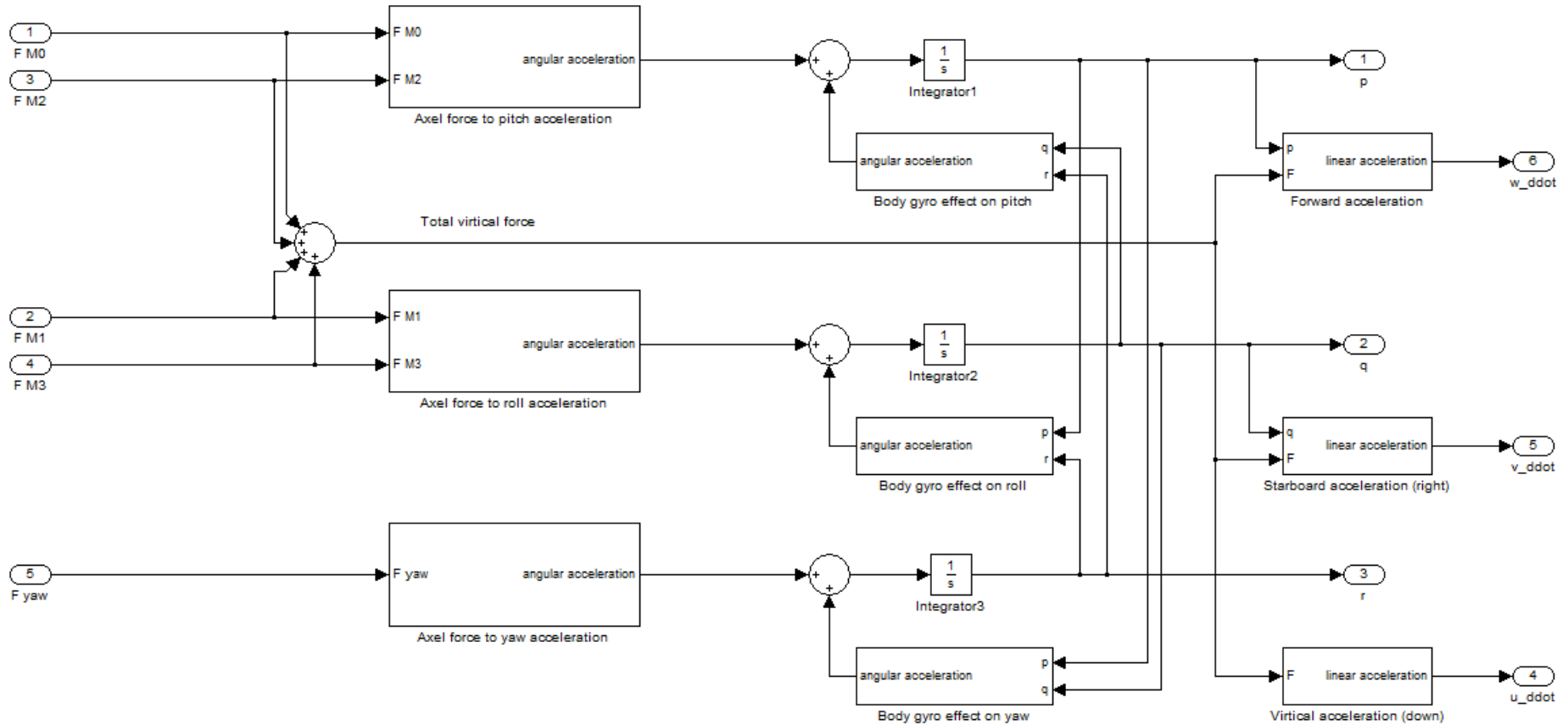
Propulsion system blokk oversetter pulsbredder inn til ESCene til en kraft ut på motorene. Underblokkene i systemet er bare matematiske beregninger, likningene er funnet ved testing, se[1]. Underblokkene inneholder også «Rate limiter» for å begrense akselerasjonen til motorene.



Figur 7: Propulsion system

1.1.8 Flight dynamics

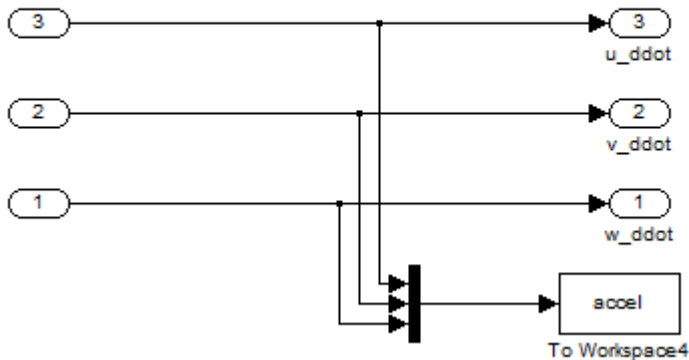
Flight dynamics blokk oversetter motorkraftene til fartøyets bevegelser slik at vi får vinklene til roll og pitch samt vinkelhastighetene til roll, pitch og yaw i følge likningen (31) i [4] med unntak av at gyroeffektene fra propellene er sett som null. Blokken tar for seg gyroeffekter som ellers nærmere seg null når fartøyet står stille i luften.



Figur 8: Flight dynamics

1.1.9 Accelerometer

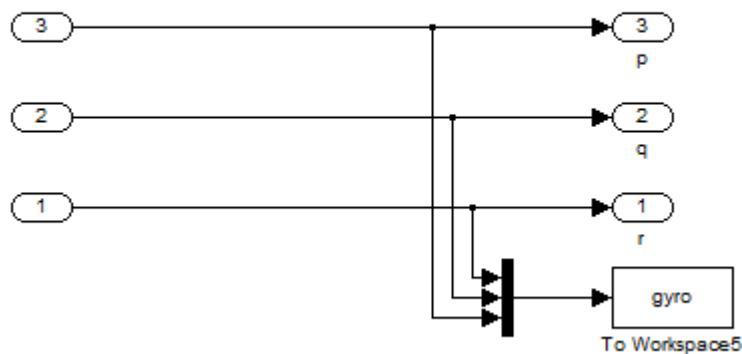
Akselerometer blokka simulerer sensorens signaler til mikrokontrolleren, her bruker vi akselerasjonen vi får fra flight dynamics og er brukt av Kalman filteret for å estimere vinklene til vinkelregulerings PIDene.



Figur 9: Accelerometer

1.1.10 Gyro

Gyro blokka simulerer sensorens signaler til mikrokontrolleren, her bruker vi vinkelhastighetene vi får fra flight dynamics og sender tilbake til PIDene i rotasjonsreguleringen.

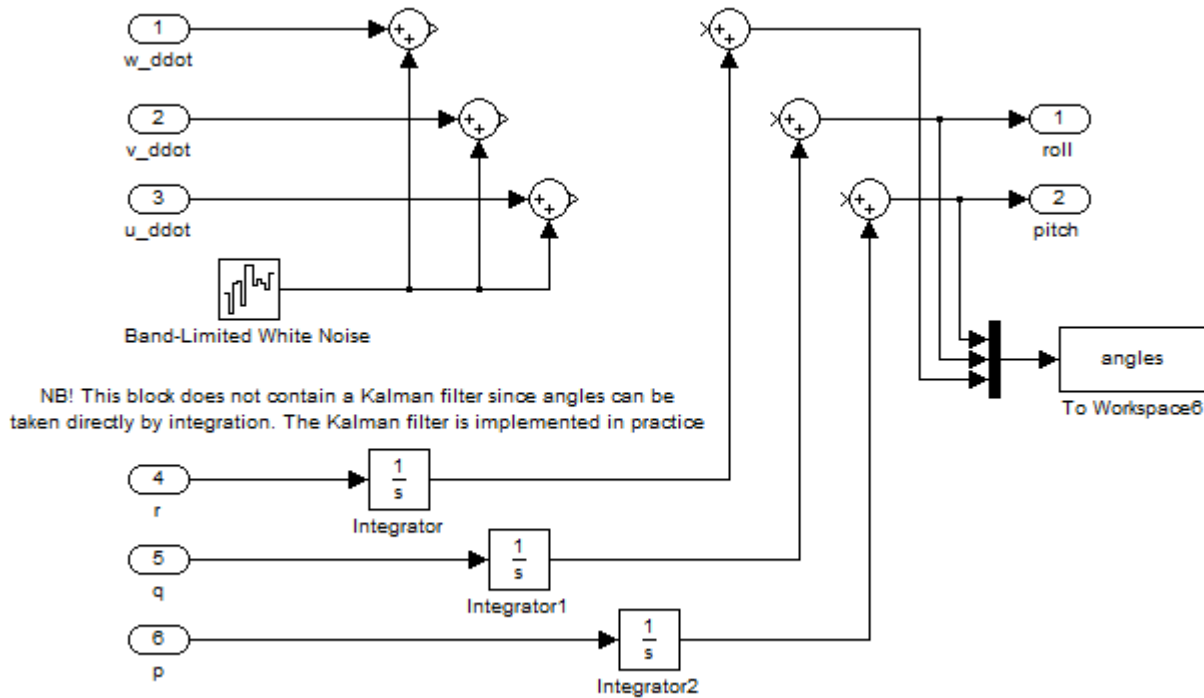


Figur 10: Gyro

1.1.11 Angle estimator(Kalman filter)

Kalman filteret er ikke implementert i simuleringen da vi får vinkler ved integrasjon. Blokken er til

for å illustrere oppkoblingen av systemet og at Kalman filteret bruker data fra både gyroen og akselerometeret for å estimere orientering.



Figur 11: Angle estimator

1.2 m-fil

For å kjøre simuleringen er det laget en m-fil[2], der man kan justere og hente/printe ut ønskede variabler.

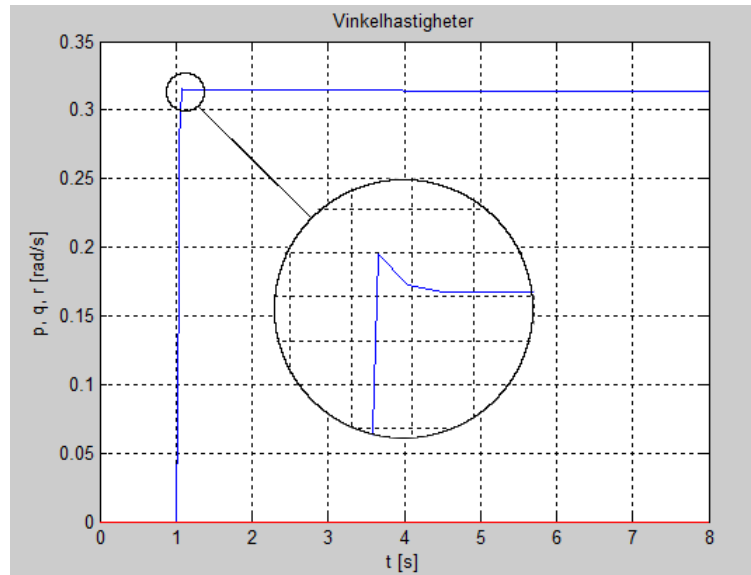
2 Resultater

2.1 Gyro modus

2.1.1 Pitch/Roll - oversving:

P	I	D
2,1	0	0

Det er et lite oversving i responsen ved en verdi på 2,1 i proporsjonalleddet. Ellers er systemet stabilt.

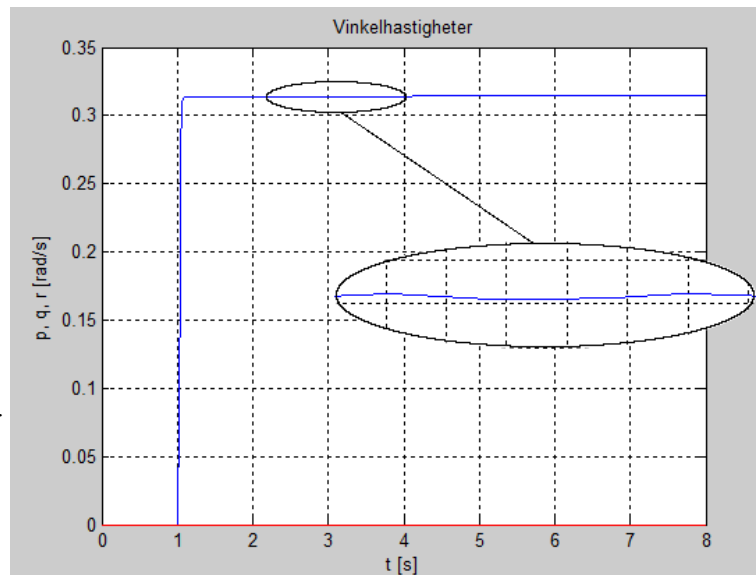


Figur 12: Pitch/Roll respons på step - oversving

2.1.2 Pitch/Roll – oscillasjoner:

P	I	D
2	0	0

Ved 2,0 i proporsjonalleddet er oversvinget borte. Når en ser nærmere nok er det mulig å se oscillasjoner i begynnelsen av responsen. Disse ligger rundt 3-5Hz i frekvens.

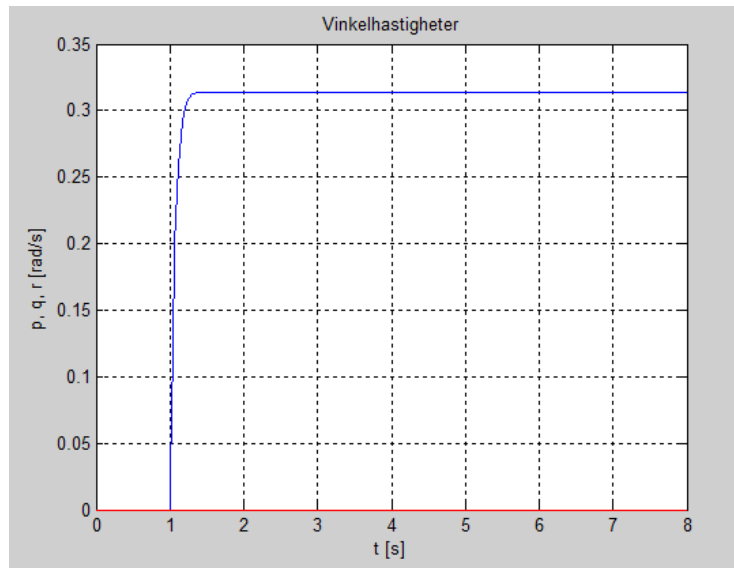


Figur 13: Pitch/Roll respons på step - oscillasjoner

2.1.3 Pitch/Roll – best:

P	I	D
0,5	0	0

Den beste responsen får vi ved en verdi på 0,5 i proporsjonalleddet. Da er det ingen oscillasjoner i responsen.



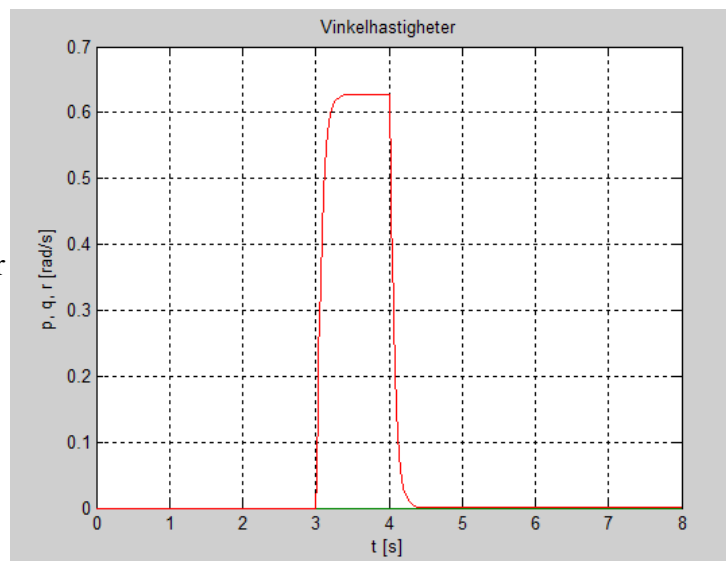
Figur 14: best Pitch/Roll respons på step

2.1.4 Yaw:

P	I	D
12	0	0

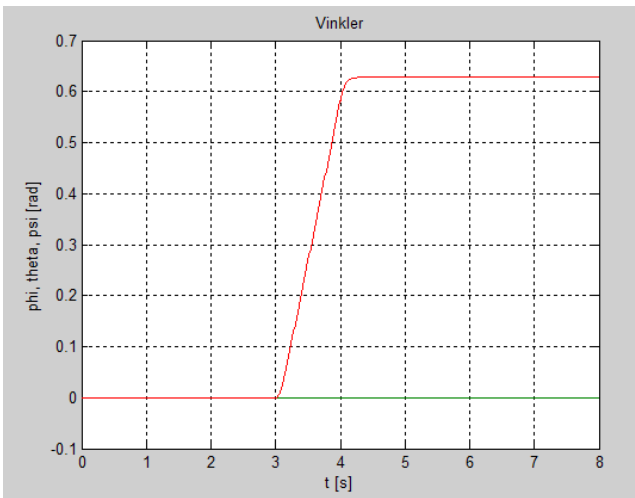
Momentet fartøyet klarer å lage om z-aksen er lite. Dette er grunnen til at proporsjonalleddet skal være vesentlig større i yaw enn i pitch og roll.

En pulsrespons er tatt i bruk fordi det er ønskelig å se at rotasjonene stopper opp når referansen fra radioen er tilbake til null. Dette er tilfellet i figur 15.

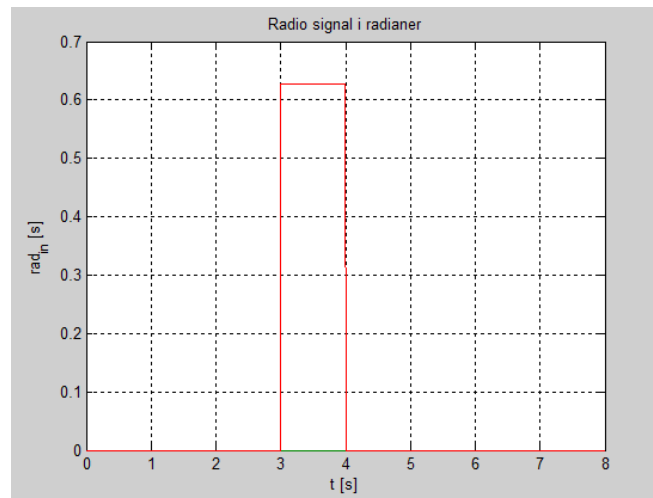


Figur 15: Yaw respons på puls - vinkelhastighet

De to siste grafene nedenfor er for vinkelposisjon, og referansesignalet kommer da fra radioen.



Figur 16: Yaw respons på step - vinkelposisjon



Figur 17: Radioreferanse i yaw

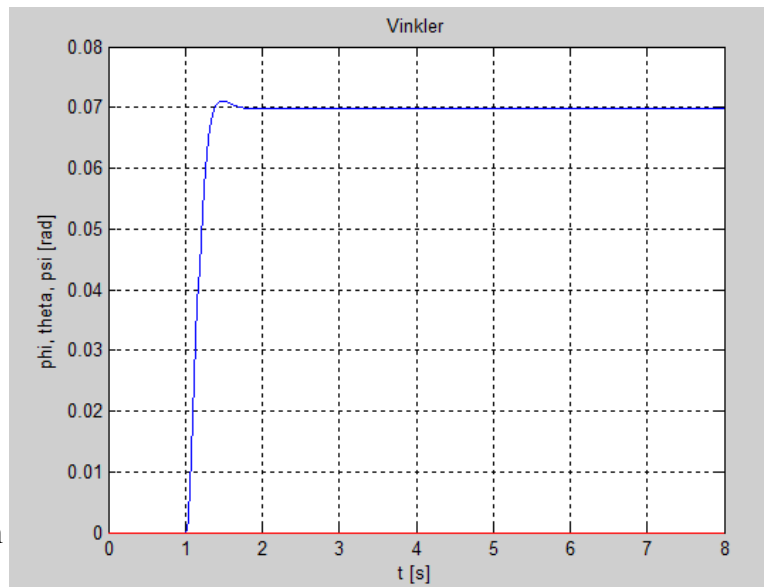
2.2 Stabil modus

2.2.1 Pitch/Roll angle regulator:

P	I	D
0,12	0	0

På den endelige vinkelreguleringen er det beste resultatet ved 0,12 i proporsjonalledet.

For å få hurtigere responser må parametrene i rotasjonsregulatorene økes, men det er da mulig at det oppstår oscillasjoner. P-leddet på vinkelregulatoren kan da økes fram til 0,2.



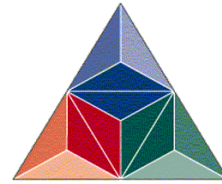
Figur 18: Vinkelregulator respons på step

Konklusjoner

Simuleringene er gjort på bakgrunn av en del forenklinger, slik at parametrene som fungerer i simuleringen ikke nødvendigvis er de optimale i praksis. Parametrene vil allikevel gi et godt utgangspunkt når vi skal tune regulatoren.

Referanser

- [1] toRegSYS.mdl
- [2] steptest_script.m
- [3] parameterfremstilling_3.0 (kapittel 2)
- [4] matematisk_modell_2.0



HØGSKOLEN
i Buskerud

Oppgavetittel: Hovedprosjekt – Seriell protokoll - v1.0		
Fag(nr./navn): SFHO 3200 / Prosjektstyring med hovedprosjekt		
Gruppemedlemmer, signatur:		
_____ Vegard Torkelsen	_____ Michael Odden	_____ Herå Rørvik
_____ Trine Lindberg	_____ Luke Carambot	_____ Patrick Bjørum
Intern veileder: Dag Samuelsen Intern sensor: Olaf Hallan Graven Ekstern veileder og sensor: Ragnvald Otterlei		
Dato: 2011-05-27		

Innholdsfortegnelse

Dokumenthistorie.....	1
Begreper og forkortelser.....	1
Innledning.....	1
1 Syntax.....	2
2 Funksjoner (func).....	2
3 Parametere (param).....	3
3.1 PID-parametere.....	3
4 Modus.....	4
4.1 Modus "debug".....	4
4.2 Modus "regulated".....	4
4.3 Modus "serialflight".....	4
5 Returverdier.....	5
6 Eksempler.....	5
7 Kjente begrensninger.....	5
Konklusjoner.....	6

Dokumenthistorie

Versjon	Beskrivelse	Når
1.0	Første versjon	2011-05-27

Tabell 1: Dokumenthistorie

Begreper og forkortelser

Begrep	Beskrivelse
N	Angir en av de fire motorene/ESCene, 0-3 hvor 0 er rett frem og ellers stigende med klokken.
I	Angir en av de ulike kanalene radio-signalene kommer inn på 1-6
MSB	Most Significant Bit
LSB	Least Significant Bit

Tabell 2: Begreper og forkortelser

Innledning

Data er overført serielt mellom en datamaskin og fartøyet via en USB-kabel. Det er da hensiktsmessig å ha størrelsen på data så små som mulig for å minske belastning og tid brukt ved lesing og skriving. I tråd med dette brukes de to første elementene til henholdsvis funksjon og parameter. Etter disse to skal resten av kommandoen beskrive verdi.

1 Syntax

Alle kommandoer vil være i følgende format:

`[<func>] <param> [<value>]`

<func> og <value> er ikke alltid påkrevd - dette blir forklart nærmere videre i dokumentet.

2 Funksjoner (func)

Når kommandoer er sent til fartøyet skal <func> være sett. Tillatte verdier er ">" og "<" (ASCII 62 og 60 respektivt) og sier om det er en parameter fartøyet skal endre verdi på eller returnere den gjeldende verdien. Data fra fartøyet skal ikke ha <func> med men starter med <param>-biten - se underkapittelet "Eksempler".

Func	ASCII	Navn	Resultat
>	62	set	Parameter skal skrives til fartøyet
<	60	get	Parameter skal leses og returneres av fartøyet

Tabell 3: Kommandofunksjoner

3 Parametere (param)

Dette er de kommandoene vi skal kunne sende til, og motta fra, fartøyet. Tillatte verdier er 0-127.

Param	Ascii	Navn	Tillatte verdier	Resultat
=	61	mode	0-7	Setter fartøyet i valgt modus. Dette er sum av flere modus - se Modus
#	35	regulation	0/1	Deaktiverer/aktiverer regulering
!	33	arm	0/1	Setter fartøyet i armert/uarmert modus
A,B,C,D	65+<N>	engine_thrust_<N>	1000-2000	Overstyrer pulsbredden sendt til motorens ESC (mikrosekunder)
a,b,c,d,e,f	96+<I>	radio_<I>	1000-2000	Lar deg simulere input via radio for de ulike kanalene (mikrosekunder)
z	121	reset	0-3	0: full reset. 1: resetter gyro offset. 2: resetter acc-offset. Kan summeres. (Obs! Ikke endelig bekreftet)
P	80	PID	Se eget delkapittel	Lesing / setting av PID-konstanter
^	94	gyro	ingen	Prefiks ved mottak av gyro-verdier (ingen get/set kommando pr nå)
-	95	acc	ingen	Prefiks ved mottak av acc-verdier (ingen get/set kommando pr nå)
*	42	multi-param	ingen	Hvis gitt i starten av en linje mottatt fra fartøyet vil det si at det kommer et sett med verdier adskilt med semi-kolon

Tabell 4: Kommandoparametere

3.1 PID-parametere

Det er definert et eget subsett med parametere for de ulike PIDene i systemet. Hver PID har tre parametere som kan settes: K_p, K_i, K_d.

Syntaks for å sende/lese en PID-verdi er:

>P<parameter><evt verdi>

Param	ASCII	Navn	Resultat
p+<PID-index>	112+index	K _p	Setter p-faktor for gitt PID
i+<PID-index>	105+index	K _i	Setter i-faktor for gitt PID
d+<PID-index>	100+index	K _d	Setter d-faktor for gitt PID

Tabell 5: PID-parametere

```
>Pp3.500 // Setter p-faktoren på første PID til 3.500
>Ph1.300 // Setter d-faktoren på femte PID til 1.300
```

PID-index er i intervallet: [0,4].

Index	PID
0	P
1	Q
2	R
3	Pitch
4	Roll

Tabell 6: PID-index

4 Modus

Da det vil være hensiktsmessig å dele opp mikrokontrollerens oppførsel i ulike modus ønsker vi å kunne angi disse. Ved noen moduser er det enkelte kommandoer som ikke vil fungere. Dette utdypes nærmere.

Moduser kan summeres bitvis.

Verdi	Modus	Binær verdi	Beskrivelse
1	debug	001	Aktiverer debug-modus, dvs sender all data til seriell utgang
2	regulated	010	Aktiverer regulert system. Dvs leser input også fra gyro og acc
4	serialflight	100	Fullstendig styrt via testbenk

Tabell 7: Modus

4.1 Modus "debug"

Settes LSB høy vil debug-modus aktiveres og all log-data sendes direkte over den serielle forbindelsen.

4.2 Modus "regulated"

Settes andre bit lav aktiveres uregulert modus hvor fartøyet ikke vil ta hensyn til akselerometer eller gyroskop, bare radio-input. Settes derimot bit'en høy vil regulert modus aktiveres.

4.3 Modus "serialflight"

Settes MSB høy vil radio-input ignoreres til fordel for parametere sendt fra testbenken.

5 Returverdier

Returverdier fra fartøyet er i formatet "<param><verdi>" - dvs at det funksjons-argumentet ikke skal være med.

Eks:

```
// Fra testbenken:  
"<C" // Les av motor2signalet  
  
// Fra fartøyet:  
"C1329" // motor2 er på 1329
```

6 Eksempler

```
">c1500" // Setter radiokanal 3 til 1500 (ca. 50%)  
"<=" // Returnerer hvilket modus fartøyet er i  
">=3" // Aktiverer regulerte-debug modus  
"<a" // Returnerer den gjeldende leste radio-verdien for  
kanal 1
```

7 Kjente begrensninger

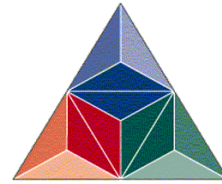
Vi har fokusert på å holde mengden data som blir sendt frem og tilbake mellom testbenk og fartøy så liten som mulig mens man fortsatt opprettholder muligheten til å skrive og forstå dette manuelt. For å opprettholde dette har vi valgt å benytte et format bestående av vanlige tegn og bokstaver.

Konsekvensene av dette er todelt:

1. Datamengdene kunne vært enda mindre ved å enkode det i et mer komprimert format men dette ville igjen ført til at formatet ikke hadde vært direkte lesbart og/eller komponerbart.
2. Parameterene består av maks to bokstaver og det gjør at det kan være noe vanskelig å huske og forstå samtlige kommandoer direkte - men det er ansett som fortsatt mulig å relativt enkelt komponere og forstå det.

Konklusjoner

Med forbehold om de begrensningene nevnt under "Kjente begrensninger" anser vi oss som fornøyde med protokollen og den har vist seg svært fungerende for våre behov.



HØGSKOLEN
i Buskerud

Oppgavetittel: Hovedprosjekt - Iterasjonsplan iterasjon 2 - v1.0 <i>Periode: 2011-01-07 - 2010-01-19</i>												
Fag(nr./navn): SFHO 3200 / Prosjektstyring med hovedprosjekt												
Gruppemedlemmer, signatur: <table><tr><td>_____</td><td>_____</td><td>_____</td></tr><tr><td>Vegard Torkelsen</td><td>Michael Odden</td><td>Herå Rørvik</td></tr><tr><td>_____</td><td>_____</td><td>_____</td></tr><tr><td>Trine Lindberg</td><td>Luke Carambot</td><td>Patrick Bjørum</td></tr></table>	_____	_____	_____	Vegard Torkelsen	Michael Odden	Herå Rørvik	_____	_____	_____	Trine Lindberg	Luke Carambot	Patrick Bjørum
_____	_____	_____										
Vegard Torkelsen	Michael Odden	Herå Rørvik										
_____	_____	_____										
Trine Lindberg	Luke Carambot	Patrick Bjørum										
Intern veileder: Dag Samuelson Intern sensor: Olaf Hallan Graven Ekstern veileder og sensor: Ragnvald Otterlei												
Dato: 2011-01-06												

Innholdsfortegnelse

Dokumenthistorie.....	1
Innledning.....	1
1 Mål.....	2
2 Aktiviteter.....	2
3 Ressurser.....	3

Dokumenthistorie

Versjon	Beskrivelse	Når
1.0	Første ferdige versjon	2011-01-06

Tabell 1: Dokumenthistorie

Innledning

En iterasjonsplan er en detaljert oversikt over målene, aktivitetene og andre relevante opplysninger rundt en gitt iterasjon. Du vil i denne iterasjonsplanen finne det overordnede samt delmål og de aktivitetene som anses nødvendige for å nå disse. Aktivitetsoversikten viser til hvem som er ansvarlige for de ulike aktivitetene, hvilke ressurser som er nødvendige og hvilke artifakter som skal være de målbare resultatene av disse. Da dette er en iterasjon med hovedfokus på design vil det ikke være fokus på brukstilfeller (use cases) mtp aktivitetene som skal gjennomføres.

1 Mål

Målet med iterasjonen er å ha et ferdig-designet uregulert fartøy. Ved avsluttet iterasjon skal alle elementer som trengs for å konstruere dette uregulerte fartøyet være klart.

1. Bekrefte at vi har alle nødvendige komponenter
2. Testing av alle relevante komponenter
3. Måling av alle nødvendige data
 1. Data sendt/mottatt av sender, mottaker og mikrokontroller
4. Design av chassi-konstruksjon
5. Design av koblingsskjema
6. Design av programvare

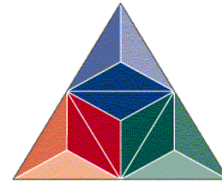
2 Aktiviteter

Aktivitet	Ansvarlig	Frist	Ressurser	Artifakter	Aktivitets ID
Design av koblingsskjema	Patrick Bjørum	2011-01-15	PC, Patrick Bjørum, Vegard Torkelsen	Koblingsskjema	407
Design av programvare for uregulert fartøy	Michael Odden	2011-01-19	PC, Michael Odden, Mikrokontroller	UML-diagrammer <ul style="list-style-type: none"> ○ Use case ○ Klasse ○ Sekvens ○ Tilstand 	401-1
Testbenk	Luke Carambot	2011-01-15	Fartøyet, verktøy, Luke Carambot, Trine Lindberg,	Gjenstand: Testbenk	410
Testplan	Luke Carambot	2011-01-13	Luke Carambot, Herå Rørvik, Trine Lindberg	Dokument: Testplan	603
Konstruksjon av prototype	Patrick Bjørum	2011-01-17	Fartøyet, Patrick Bjørum, Vegard Torkelsen	Gjenstand: Prototype	501
Iterasjonsrapport	Michael Odden	2011-01-20	PC, Michael Odden	Dokument: Iterasjonsrapport	101-1

Tabell 2: Aktivitetsoversikt

3 Ressurser

- Personer
 - Alle gruppemedlemmene
- Utstyr
 - Personlig PC/laptop
 - Alminnelig relevant verktøy, loddeapparat, ledninger mm.
- Komponenter
 - Mikrokontroller
 - Chassi
 - Fjernkontroll
 - Radiomottaker
- Lokaler
 - Grupperom, C161 (Høgskolen i Buskerud avd Kongsberg)



HØGSKOLEN
i Buskerud

Oppgavetittel:

Hovedprosjekt - Iterasjonsplan iterasjon 3 - v2.0
Periode: 2011-01-20 - 2011-02-21

Fag(nr./navn):

SFHO 3200 / Prosjektstyring med hovedprosjekt

Gruppemedlemmer, signatur:

Vegard Torkelsen

Michael Odden

Herå Rørvik

Trine Lindberg

Luke Carambot

Patrick Bjørum

Intern veileder: Dag Samuelson

Intern sensor: Olaf Hallan Graven

Ekstern veileder og sensor: Ragnvald Otterlei

Dato: 2011-01-27

Innholdsfortegnelse

Dokumenthistorie.....	1
Innledning.....	1
1 Mål.....	2
2 Risikoanalyse.....	2
2.1 Ødeleggelse av komponenter.....	2
2.2 Sykdom.....	2
2.3 Programvare-komplikasjoner.....	3
3 Aktiviteter.....	3
4 Aktiviteter, detaljert.....	4
404-1: Utvikling av programvare.....	4
501-2: Forbedring av prototype.....	4
601-1: Testing av prototype.....	4
106-2: Regnskap.....	5
101-2: Iterasjonsrapport 3.....	5
502: Testbenk.....	5
5 Aktiviteter pr person.....	6
6 Ressurser.....	6
Referanser.....	8

Dokumenthistorie

Versjon	Beskrivelse	Når
1.0	Fylt ut disposisjon, beskrevet alle aktivitetene i større detalj og trukket ut oppgaver pr person.	2011-01-27
2.0	Lagt til avsnitt om risikoanalyse og korrigert skrivefeil.	2011-01-27

Tabell 1: Dokumenthistorie

Innledning

En iterasjonsplan er en detaljert oversikt over målene, aktivitetene og andre relevante opplysninger rundt en gitt iterasjon. Du vil i denne iterasjonsplanen finne det overordnede samt delmål og de aktivitetene som anses nødvendige for å nå disse. Aktivitetsoversikten viser til hvem som er ansvarlige for de ulike aktivitetene, hvilke ressurser som er nødvendige og hvilke artifakter som skal være de målbare resultatene av disse. Den overordnede aktivitetsoversikten er etterfulgt av en mer detaljert gjennomgang.

Vær obs på at personen ansvarlig for den enkelte aktivitet skal sørge for at:

- All nødvendig informasjon er samlet inn
- Progresjon oppdateres på ProsjektWeb
- Problemer taes opp med relevante personer og/eller gruppen
- Delegerer oppgaver hensiktsmessig til andre gruppemedlemmer
- Aktiviteten er gjennomført i tide
 - Skulle ikke dette la seg gjøres skal det dokumenteres og ny tidsfrist skal defineres så kjapt som mulig.
- Om noen aktiviteter er blitt i overkant store vil det være hensiktsmessig å dele den opp i sub-aktiviteter for bedre å måle fremgang.

1 Mål

Målet med iterasjonen er å ha et ferdig-konstruert uregulert fartøy. Ved avsluttet iterasjon skal fartøyet være flyvbart vha fjernkontroll.

1. Implementere programvare for styring
2. Fullføre prototypen
3. Fullføre testbenk
 1. Fysisk enhet
 2. Programvare for utlesing og enkel justering av verdier/parametre
4. Utføre tester iht testplanen [3]

2 Risikoanalyse

2.1 Ødeleggelse av komponenter

Det er en kontinuerlig risiko at komponenter kan bli ødelagt ved test og bruk. Det oppfordres til forsiktighet rundt håndteringen, samt påpasselighet med tanke på statisk elektrisitet, kortslutning, klipping mm.

Videre så vil det i denne iterasjonen være en del tester i forhold til flyveegenskaper og dette kan føre til skader spesielt på propeller, chassi og omgivelser. Vi har enkel tilgang til ekstra sett med propeller. Alle andre komponenter vil innebære en viss ekstra kostnad og leveringstid.

Lengst leveringstid er det for deler fra AeroQuadStore[4] (mikrokontroller og sensorer) som var på 2-3 uker, og høyeste enkeltkostnad er knyttet til mottaker (\$89), chassi (\$89) (sett som en enhet) og mikrokontroller (\$65).

2.2 Sykdom

Alle aktiviteter har minst to personer tilknyttet seg som ressurspersoner. Skulle den ansvarlige for aktiviteten bli syk eller av andre grunner ikke kunne fullføre sine oppgaver vil den neste ressurspersonen ta over dette ansvaret og dermed oppgaven med å evt delegere dette videre.

Designene fra forrige iterasjon (koblingskjema og programvare) er såpass godt at det skal være mulig for de andre å implementere dette.

2.3 Programvare-komplikasjoner

De største utfordringene ved denne iterasjonen vil mest sannsynlig være ifm programvaren. Denne må være robust og nøyaktig for å gjøre den tilstrekkelig flyvbar samt forhindre skader på fartøyet og omgivelsene. Programvaren bør dermed så langt det lar seg gjøre testes i isolerte former og deretter mot enkeltkomponenter før de endelige forsøkene. Dette i henhold til testplanen [3].

3 Aktiviteter

Aktivitet	Ansvarlig	Frist	Ressurser	Artifakter	ID
Utvikling av programvare	Michael Odden	2011-02-19	PC, Michael Odden, Luke Carambot, Mikrokontroller	Arduino sketch (c++)	404-1
Forbedring av prototype	Vegard Torkelsen	2011-02-11	Fartøyet, PC, Vegard Torkelsen, Patrick Bjørum, Trine Lindberg, Herå Rørvik	Prototype	501-2
Testing av prototype	Herå Rørvik	2011-02-21	Fartøyet, PC, testbenk, Herå Rørvik, Luke Carambot, Patrick Bjørum	Testrapport / logg	601-1
Regnskap	Patrick Bjørum	2011-02-07	PC, Patrick Bjørum	Dokument over regnskapet	106-2
Iterasjonsrapport 3	Michael Odden	2011-02-23	PC, Michael Odden, Trine Lindberg	Dokument: Iterasjonsrapport	101-2
Testbenk	Luke Carambot	2011-02-xx	Luke Carambot, Michael Odden	Testbenk, Kontrollprogram	502

Tabell 2: Nærmere detaljer rundt aktivitetene finnes i prosjektplanen[1] samt på prosjektweb[2].

4 Aktiviteter, detaljert

404-1: Utvikling av programvare

Ansvarlig: Michael Odden

Antatt tidsbruk: 40t

Krav: KA702, KA801- KA804

Frist: 2011-02-19

Artifakter: Kildekode for uregulert fartøy. Arduino skecth (c++)

Ressurser: PC, mikrokontroller m/ og uten fartøy.

Med utvikling av programvare gjelder det her den programvaren som skal kjøre på mikrokontrolleren ombord i fartøyet. Denne skal håndtere input fra radio-kildo og omforme dette til pulsbredde-signaler til de 4 ESCene som igjen vil styre kreftene til de respektive motorene.

501-2: Forbedring av prototype

Ansvarlig: Vegard Torkelsen

Antatt tidsbruk:

Krav: KA901, i tillegg til kodekrav og tester

Frist: 2011-02-11

Artifakter:

- Forbedret prototype

Ressurser: Eksisterende prototype, PC, Alle komponenter

En prototype er allerede konstruert i forbindelse med forrige iterasjons tester. Denne skal nå ferdigkonstrueres slik at den med programvaren (404-1) fullfører iterasjonsmålet om det uregulerte fartøyet.

601-1: Testing av prototype

Ansvarlig: Herå Rørvik

Antatt tidsbruk:

Tester: TI422, TI704, TI709, TS101-TS105

Frist: 2011-02-16

Artifakter:

- Testlogg

Ressurser: Forbedret prototype, PC, testbenk

Alle utestående tester i forhold til det uregulerte fartøyet skal gjennomføres og loggføres.

106-2: Regnskap

Ansvarlig: Patrick Bjørum

Antatt tidsbruk: 2t

Frist: 2011-02-21

Artifakter:

- Regnskap

Ressurser: PC, budsjett

Regnskapet skal gi et oversiktlig bilde over hvilke utgifter prosjektet har hatt - og hvor disse har gått. Denne bør sees i sammenheng med budsjettet. Dette vil selvsagt oppdateres mot slutten av prosjektet, men det antas nå at hovedinnkjøpene er gjort og dette bør dokumenteres. Det skal også fremkomme hvilke komponenter vi har fått låne - og av hvem.

101-2: Iterasjonsrapport 3

Ansvarlig: Michael Odden

Antatt tidsbruk: 2t

Frist: 2011-02-22

Iterasjonsrapporten skal skrives når den relaterte iterasjonen er fullført, men det kan være hensiktsmessig å hente inn informasjon fra aktivitetene fortløpende. Den skal evaluere alle aspekter av iterasjonen.

502: Testbenk

Ansvarlig: Luke Carambot

Antatt tidsbruk: 20t

Krav: Ingen direkte krav.

Frist: 2011-02-20

Artifakter:

1. Fysisk testbenk (i hovedsak ferdig)
2. Programvare for utlesing av informasjon over USB (Arduino Sketch, C++)

3. Programvare for konfigurering av systemet over USB (Arduino Sketch, C++)

Ressurser: Luke Carambot, Michael Odden, Prototype

Testbenken har som oppgave å muliggjøre og forenkle innhenting av data knyttet til de ulike komponentene samt del-systemene og systemet i sin helhet. Dette vil være enkel lesing av input, output og relevante nøkkelverdier innad i programvaren, samt måling av de genererte kreftene.

5 Aktiviteter pr person

Her følger en forenklet oversikt over hvilke personer som anses som ressurs-personer ihht hvilke aktiviteter. Den må sees i sammenheng med aktivitetsoversikten ovenfor.

Person	Aktiviteter
Vegard Torkelsen	<ul style="list-style-type: none">• 501-2: Forbedring av prototype• 601-1: Testing av prototype
Patrick Bjørum	<ul style="list-style-type: none">• 106-2: Regnskap• 501-2: Forbedring av prototype
Trine Lindberg	<ul style="list-style-type: none">• 501-2: Forbedring av prototype• 101-2: Iterasjonsrapport
Herå Rørvik	<ul style="list-style-type: none">• 601-1: Testing av prototype• 501-2: Forbedring av prototype
Luke Carambot	<ul style="list-style-type: none">• 502: Testbenk• 404-1: Utvikling av programvare
Michael Odden	<ul style="list-style-type: none">• 404-1: Utvikling av programvare• 101-2: Iterasjonsrapport• 502: Testbenk

Tabell 3: De ulike aktivitetene hver person skal bidra til

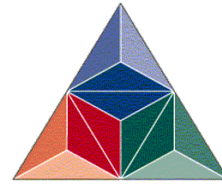
6 Ressurser

- Personer
 - Alle gruppelemmene
- Utstyr
 - Personlig PC/laptop
 - Alminnelig relevant verktøy, loddeapparat, ledninger mm.
 - Testbenk (vil utvikles over iterasjonen)

- Komponenter
 - Mikrokontroller
 - Chassi
 - Sensorer: Akselerometer, gyroskop
 - Batteri
 - Fjernkontroll
 - Radiomottaker
 - Lokaler
 - Grupperom, C161 (Høgskolen i Buskerud avd Kongsberg)
-

Referanser

- [1] Prosjektplan - prosjektplan_4.0.pdf
- [2] Veikart fra ProsjektWeb - veikart_1.0.pdf
- [3] Testplan - testplan_2.0.pdf
- [4] AeroQuad Store, <https://www.aeroquadstore.com/> - 2011-01-30



HØGSKOLEN
i Buskerud

Oppgavetittel: Hovedprosjekt - Iterasjonsplan iterasjon 4 - v1.0 <i>Periode: 2011-02-22 - 2010-03-16</i>		
Fag(nr./navn): SFHO 3200 / Prosjektstyring med hovedprosjekt		
Gruppemedlemmer, signatur:		
_____ Vegard Torkelsen	_____ Michael Odden	_____ Herå Rørvik
_____ Trine Lindberg	_____ Luke Carambot	_____ Patrick Bjørum
Intern veileder: Dag Samuelson Intern sensor: Olaf Hallan Graven Ekstern veileder og sensor: Ragnvald Otterlei		
Dato: 2011-02-22		

Innholdsfortegnelse

Dokumenthistorie.....	1
Innledning.....	1
1 Mål.....	2
2 Risikoanalyse.....	2
2.1 Ødeleggelse av komponenter.....	2
2.2 Sykdom.....	2
2.3 Programvare-komplikasjoner.....	2
2.4 Tilgang til ressurser.....	2
3 Aktiviteter.....	3
4 Aktiviteter, detaljert.....	4
5 Aktiviteter pr person.....	4
6 Resurser.....	5
Referanser.....	6

Dokumenthistorie

Versjon	Beskrivelse	Når
1.0	Første offisielle versjon	2011-02-22

Tabell 1: Dokumenthistorie

Innledning

En iterasjonsplan er en detaljert oversikt over målene, aktivitetene og andre relevante opplysninger rundt en gitt iterasjon. Du vil i denne iterasjonsplanen finne det overordnede samt delmål og de aktivitetene som anses nødvendige for å nå disse. Aktivitetsoversikten viser til hvem som er ansvarlige for de ulike aktivitetene, hvilke ressurser som er nødvendige og hvilke artifakter som skal være de målbare resultatene av disse. Den overordnede aktivitetsoversikten er etterfulgt av en mer detaljert gjennomgang.

Vær obs på at personen ansvarlig for den enkelte aktivitet skal sørge for at:

- All nødvendig informasjon er samlet inn
- Progresjon oppdateres på ProsjektWeb
- Problemer taes opp med relevante personer og/eller gruppen
- Delegerer oppgaver hensiktsmessig til andre gruppemedlemmer
- Aktiviteten er gjennomført i tide
 - Skulle ikke dette la seg gjøres skal det dokumenteres og ny tidsfrist skal defineres så kjapt som mulig.
- Om noen aktiviteter er blitt i overkant store vil det være hensiktsmessig å dele den opp i sub-aktiviteter for bedre å måle fremgang.

1 Mål

Målet med iterasjon 4 er å designe det regulerte systemet. Dette er den antakeligvis tyngste iterasjonen teoretisk sett. Vi vil også under denne iterasjonen lage vår andre presentasjon.

Resultatet av denne iterasjonen vil blant annet være:

- Design-dokumenter og UML-diagrammer for programvare
- Dokumenter om signalbehandling, matematiske modeller og reguleringsystemets design
- Presentasjon 2

2 Risikoanalyse

2.1 Ødeleggelse av komponenter

Det er en kontinuerlig risiko at komponenter kan bli ødelagt ved test og bruk. Det oppfordres til forsiktighet rundt håndteringen, samt påpasselighet med tanke på statisk elektrisitet, kortslutning, klipping mm.

I denne iterasjonen vil hovedfokuset være på design og det vil ikke være langt mindre andel flygetester enn forrige og neste iterasjon.

2.2 Sykdom

Alle aktiviteter har minst to personer tilknyttet seg som ressurspersoner. Skulle den ansvarlige for aktiviteten bli syk eller av andre grunner ikke kunne fullføre sine oppgaver vil den neste ressurspersonen ta over dette ansvaret og dermed oppgaven med å evt delegere dette videre. Designene fra forrige iterasjon (koblingsskjema og programvare) er såpass godt at det skal være mulig for de andre å implementere dette.

2.3 Programvare-komplikasjoner

Det ligger ikke så mye utvikling av programvare for fartøyet under denne iterasjonen, men derimot mer for testbenken da vi er avhengig av å kunne gjøre en mengde målinger under utvikling av reguleringsystemet.

2.4 Tilgang til ressurser

Prosjektets mest fremtredende flaskehals er fartøyet da vi kun har ett av dette og samtlige av oss vil på ulike tidspunkt trenge tilgang til dette for å utføre ulike tester. Det er dermed viktig av vi er effektive under den tiden vi bruker fartøyet og lar andre slippe til i de periodene man kan.

3 Aktiviteter

Aktivitet	Ansvarlig	Frist	Ressurser	Artifakter	ID
Presentasjon 2	Herå Rørvik	2011-03-17	PC, Fartøyet, Alle medlemmer,	Presentasjon, Perm med dokumenter, CDer med alle dokumenter	111-2
Design av programvare for regulert fartøy	Michael Odden	2011-03-16	PC, Luke Carambot, Fartøyet,	UML-diagrammer, overordnet design-dokument	401-2
Iterasjonsrapport 4	Michael Odden	2011-03-18	PC	Iterasjonsrapport	101-3
Design av matematisk modell	Patrick Bjørum	2011-03-10	PC, Luke Carambot,	Dokumentasjon av modell	408
Design av signalbehandling	Vegard Torkelsen	2011-02-28	PC, Fartøyet	Designdokument for signalbehandling	409
Matematisk modell	Luke Carambot	2011-03-10	PC, Patrick Bjørum,	Matematisk modell	701
Design av reguleringsystem	Trine Lindberg	2011-03-10	PC, Fartøyet,	Designdokument for reguleringsystem	406-1
Simulering av reguleringsystem	Trine Lindberg	2011-03-16	PC, Designdokument for reguleringsystem (406-1),	Simulator	702
Iterasjonsplan Iterasjon 5	Michael Odden	2011-03-17	PC, Vegard Torkelsen	Iterasjonsplan	<i>Ikke definert</i>
Utbedre testbenk for uregulert fartøy	Luke Carambot	2011-03-17	PC	Oppdatert testbenk-programvare	502-2
Design testbenk-utvidelse for regulert fartøy	Herå Rørvik	2011-03-10	PC	Designdokument for utvidelser	410-1
Utbedre signalproblemer	Michael Odden	2011-03-05	PC, Fartøyet	Oppdatert programvare	404-3
Design av nye tester	Patrick Bjørum	2011-03-03	PC		601-7
Ombygging for lavere COG	Patrick Bjørum	2011-03-01	PC, Fartøyet	Ombygd prototype	501-2

[tabell 2 - Nærmere detaljer rundt aktivitetene finnes i prosjektplanen[1] samt på prosjektweb[2].]

4 Aktiviteter, detaljert

For utfyllende informasjon om de ulike aktivitetene refereres det til ProsjektWeb [2].

5 Aktiviteter pr person

Her følger en forenklet oversikt over hvilke personer som anses som ressurs-personer iht hvilke aktiviteter. Den må sees i sammenheng med aktivitetsoversikten ovenfor.

Person	Aktiviteter
<Samtlige>	<ul style="list-style-type: none"> • Presentasjon 2
Vegard Torkelsen	<ul style="list-style-type: none"> • Design av signalbehandling • Design av reguleringsystem • Utbedre testbenk for uregulert fartøy • Iterasjonsplan, iterasjon 5
Patrick Bjørum	<ul style="list-style-type: none"> • Design av signalbehandling • Design av matematisk modell • Matematisk modell • Design av nye tester
Trine Lindberg	<ul style="list-style-type: none"> • Design av reguleringsystem • Simulering av reguleringsystem • Design av signalbehandling
Herå Rørvik	<ul style="list-style-type: none"> • Design av nye tester • Design av reguleringsystem • Simulering av reguleringsystem • Designe testbenk-utvidelse for regulert fartøy
Luke Carambot	<ul style="list-style-type: none"> • Utbedre testbenk for uregulert fartøy • Matematisk modell • Design av matematisk modell • Design av programvare for regulert fartøy
Michael Odden	<ul style="list-style-type: none"> • Utbedre signalproblemer • Design av programvare for regulert fartøy • Utbedre testbenk for uregulert fartøy • Iterasjonsrapport, iterasjon 4 • Iterasjonsplan, iterasjon 5

[tabell 3 - De ulike aktivitetene hver person kan bidra til]

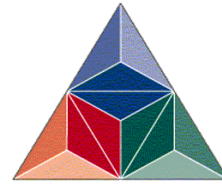
6 Ressurser

- Personer
 - Alle gruppelemmene
- Utstyr
 - Personlig PC/laptop
 - Alminnelig relevant verktøy, loddeapparat, ledninger, lader mm.
 - Testbenk (vil utvikles etter behov gjennom iterasjonen)
- Komponenter - mikrokontroller, chassi, sensorer (gyro, akselerometer), batteri, fjernkontroll, radiomottaker
- Lokaler
 - Grupperom, C161 (Høgskolen i Buskerud avd Kongsberg)

Referanser

[1] Prosjektplan - prosjektplan_4.0.pdf

[2] Veikart fra ProsjektWeb - veikart_1.0.pdf



HØGSKOLEN
i Buskerud

Oppgavetittel:

Hovedprosjekt - Iterasjonsplan iterasjon 5 - v1.0
Periode: 2011-03-18 - 2010-03-30

Fag(nr./navn):

SFHO 3200 / Prosjektstyring med hovedprosjekt

Gruppemedlemmer, signatur:

Vegard Torkelsen

Michael Odden

Herå Rørvik

Trine Lindberg

Luke Carambot

Patrick Bjørum

Intern veileder: Dag Samuelson

Intern sensor: Olaf Hallan Graven

Ekstern veileder og sensor: Ragnvald Otterlei

Dato: 2011-03-18

Innholdsfortegnelse

Dokumenthistorie.....	1
Innledning.....	ii
1 Mål.....	3
2 Risikoanalyse.....	3
2.1 Ødeleggelse av komponenter.....	3
2.2 Sykdom og annet fravær.....	3
2.3 Programvare-komplikasjoner.....	3
2.4 Tilgang til ressurser.....	3
3 Aktiviteter.....	4
4 Aktiviteter, detaljert.....	4
5 Aktiviteter pr person.....	4
6 Resurser.....	5
Referanser.....	6

Dokumenthistorie

Versjon	Beskrivelse	Når
1.0	Første offisielle versjon	2011-03-18

Tabell 1: Dokumenthistorie

Innledning

En iterasjonsplan er en detaljert oversikt over målene, aktivitetene og andre relevante opplysninger rundt en gitt iterasjon. Du vil i denne iterasjonsplanen finne det overordnede samt delmål og de aktivitetene som anses nødvendige for å nå disse. Aktivitetsoversikten viser til hvem som er ansvarlige for de ulike aktivitetene, hvilke ressurser som er nødvendige og hvilke artifakter som skal være de målbare resultatene av disse. Den overordnede aktivitetsoversikten er etterfulgt av en mer detaljert gjennomgang.

Vær obs på at personen ansvarlig for den enkelte aktivitet skal sørge for at:

- All nødvendig informasjon er samlet inn
- Progresjon oppdateres på ProsjektWeb
- Problemer taes opp med relevante personer og/eller gruppen
- Delegerer oppgaver hensiktsmessig til andre gruppemedlemmer
- Aktiviteten er gjennomført i tide
 - Skulle ikke dette la seg gjøres skal det dokumenteres og ny tidsfrist skal defineres så kjapt som mulig.
- Om noen aktiviteter er blitt i overkant store vil det være hensiktsmessig å dele den opp i sub-aktiviteter for bedre å måle fremgang.

1 Mål

Målet med iterasjon 5 er å implementere systemet som designet i iterasjon 4.

Resultatet av denne iterasjonen vil hovedsaklig være:

- Oppdatert programvare med enkelt reguleringsystem
- Test-data fra implementert system til analyse for forbedret design

Ved fullført iterasjon skal vi ha opparbeidet nok kunnskap om hva som er problemområdene slik at disse kan analyseres ved behov og deretter designe en løsning i iterasjon 6 for så igjen å implementere dette i iterasjon 7.

2 Risikoanalyse

2.1 Ødeleggelse av komponenter

Det er en kontinuerlig risiko at komponenter kan bli ødelagt ved test og bruk. Det oppfordres til forsiktighet rundt håndteringen, samt påpasselighet med tanke på statisk elektrisitet, kortslutning, klipping mm.

I denne iterasjonen vil hovedfokuset være på implementasjon og testing og det vil da være en viss risiko for komponenter. Det er dog ikke planlagt noen større flygetester.

2.2 Sykdom og annet fravær

Alle aktiviteter har minst to personer tilknyttet seg som ressurspersoner. Skulle den ansvarlige for aktiviteten bli syk eller av andre grunner ikke kunne fullføre sine oppgaver vil den neste ressurspersonen ta over dette ansvaret og dermed oppgaven med å evt delegere dette videre. Designene fra forrige iterasjon (koblingskjema og programvare) er såpass godt at det skal være mulig for de andre å implementere dette.

2.3 Programvare-komplikasjoner

I denne iterasjonen legges det stor vekt på programvaren, og komplikasjoner knyttet til denne kan fort ta mye kostbar tid. Hvis designet er veloverveid nok vil det hjelpe til med å begrense risikoen, men vi innfører også peer-reviews hvor en annen person alltid vil overse den enkeltes kode.

2.4 Tilgang til ressurser

Prosjektets mest fremtredende flaskehals er fartøyet da vi kun har ett av dette og samtlige av oss vil på ulike tidspunkt trenge tilgang til dette for å utføre ulike tester. Det er dermed viktig av vi er effektive under den tiden vi bruker fartøyet og lar andre slippe til i de periodene man kan.

3 Aktiviteter

Aktivitet	Ansvarlig	Frist	Ressurser	Artifakter	ID
Programvare for regulert fartøy	Michael Odden	2011-04-28	PC, fartøy	Oppdatert kildekode.	404-2
Implementering av signalbehandling	Vegard Torkelsen	2011-03-24	PC, fartøy	Klasser: BMA180, ITG3200	504
Implementering av reguleringsystem	Trine Lindberg	2011-03-29	PC, fartøy	Klasser: Regulator, PID	505
Testing av reguleringsystem	Herå Rørvik	2011-03-30	PC, fartøy	Testdokumenter etter behov	601-2
Iterasjonsrapport 5	Michael Odden	2011-04-30	PC	Dokument, iterasjonsrapport	114-5

Tabell 2: Nærmere detaljer rundt aktivitetene finnes i prosjektplanen[1] samt på ProsjektWeb[2].

4 Aktiviteter, detaljert

For utfyllende informasjon om de ulike aktivitetene refereres det til ProsjektWeb [2].

5 Aktiviteter pr person

Her følger en forenklet oversikt over hvilke personer som anses som ressurs-personer ihht hvilke aktiviteter. Den må sees i sammenheng med aktivitetsoversikten ovenfor.

Person	Aktiviteter
<Samtlige>	<ul style="list-style-type: none"> • Ved ledig tid: <ol style="list-style-type: none"> 1. Tilby assistanse til de andre medlemmene 2. Begynn å forbered dokumenter for presentasjon 3. Se på fremtidige aktiviteter som kan påbegynnes
Vegard Torkelsen	<ul style="list-style-type: none"> • Signalbehandling • Programvare
Patrick Bjørum	<ul style="list-style-type: none"> • Testing av reguleringsystem • Signalbehandling
Trine Lindberg	<ul style="list-style-type: none"> • Reguleringsystem • Programvare
Herå Rørvik	<ul style="list-style-type: none"> • Testing av reguleringsystem • Reguleringsystem
Luke Carambot	<ul style="list-style-type: none"> • Testing av reguleringsystem • Generell jobbing med matematiske modeller

Michael Odden	<ul style="list-style-type: none">• Programvare• Iterasjonsrapport
---------------	---

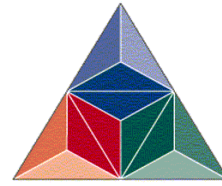
Tabell 3: De ulike aktivitetene hver person kan bidra til

6 Ressurser

- Personer
 - Alle gruppemedlemmene
 - Intern veileder: Dag Samuelsen
- Utstyr
 - Personlig PC/laptop
 - Alminnelig relevant verktøy, loddeapparat, ledninger, lader mm.
 - Testbenk (vil utvikles etter behov gjennom iterasjonen)
- Komponenter - mikrokontroller, chassi, sensorer (gyro, akselerometer), batteri, fjernkontroll, radiomottaker
- Lokaler
 - Grupperom, C161 (Høgskolen i Buskerud avd Kongsberg)

Referanser

- [1] Prosjektplan - prosjektplan_4.0.pdf
- [2] Veikart fra ProsjektWeb - veikart_1.0.pdf
- [3] Møterefertat veiledermøte - møterefertat_veileder_2011-03-10.pdf



HØGSKOLEN
i Buskerud

Oppgavetittel:

Hovedprosjekt - Iterasjonsplan iterasjon 6 - v1.0
Periode: 2011-03-31 - 2010-04-13

Fag(nr./navn):

SFHO 3200 / Prosjektstyring med hovedprosjekt

Gruppemedlemmer, signatur:

Vegard Torkelsen

Michael Odden

Herå Rørvik

Trine Lindberg

Luke Carambot

Patrick Bjørum

Intern veileder: Dag Samuelson

Intern sensor: Olaf Hallan Graven

Ekstern veileder og sensor: Ragnvald Otterlei

Dato: 2011-03-18

Innholdsfortegnelse

Dokumenthistorie.....	1
Innledning.....	1
1 Mål.....	2
2 Risikoanalyse.....	2
2.1 Ødeleggelse av komponenter.....	2
2.2 Sykdom og annet fravær.....	2
2.3 Programvare-komplikasjoner.....	2
2.4 Tilgang til ressurser.....	2
3 Aktiviteter.....	3
4 Aktiviteter, detaljert.....	3
5 Aktiviteter pr person.....	3
6 Resurser.....	4
Referanser.....	5

Dokumenthistorie

Versjon	Beskrivelse	Når
1.0	Første offisielle versjon	2011-03-30

Tabell 1: Dokumenthistorie

Innledning

En iterasjonsplan er en detaljert oversikt over målene, aktivitetene og andre relevante opplysninger rundt en gitt iterasjon. Du vil i denne iterasjonsplanen finne det overordnede samt delmål og de aktivitetene som anses nødvendige for å nå disse. Aktivitetsoversikten viser til hvem som er ansvarlige for de ulike aktivitetene, hvilke ressurser som er nødvendige og hvilke artifakter som skal være de målbare resultatene av disse. Den overordnede aktivitetsoversikten er etterfulgt av en mer detaljert gjennomgang.

Vær obs på at personen ansvarlig for den enkelte aktivitet skal sørge for at:

- All nødvendig informasjon er samlet inn
- Progresjon oppdateres på ProsjektWeb
- Problemer taes opp med relevante personer og/eller gruppen
- Delegerer oppgaver hensiktsmessig til andre gruppemedlemmer
- Aktiviteten er gjennomført i tide
 - Skulle ikke dette la seg gjøres skal det dokumenteres og ny tidsfrist skal defineres så kjapt som mulig.
- Om noen aktiviteter er blitt i overkant store vil det være hensiktsmessig å dele den opp i sub-aktiviteter for bedre å måle fremgang.

1 Mål

Målet med iterasjon 6 er å forbedre designet på de punktene oppdaget som nødvendige under implementeringene i iterasjon 5. Disse vil da re-implementeres i iterasjon 7. Vi flyttet også Presentasjon 2 fra iterasjon 4 til denne da vi gjorde oppdateringer på tids-planen.[3]

Resultatet av denne iterasjonen vil hovedsaklig være:

- Spesialiserte design-dokumenter samt oppdaterte overordnede design-dokumenter der dette inngår.
- Presentasjon 2

2 Risikoanalyse

2.1 Ødeleggelse av komponenter

Det er en kontinuerlig risiko at komponenter kan bli ødelagt ved test og bruk. Det oppfordres til forsiktighet rundt håndteringen, samt påpasselighet med tanke på statisk elektrisitet, kortslutning, klipping mm.

I denne iterasjonen vil hovedfokuset være på design og det vil ikke være en langt mindre andel flygetester enn forrige og neste iterasjon.

2.2 Sykdom og annet fravær

Alle aktiviteter har minst to personer tilknyttet seg som ressurspersoner. Skulle den ansvarlige for aktiviteten bli syk eller av andre grunner ikke kunne fullføre sine oppgaver vil den neste ressurspersonen ta over dette ansvaret og dermed oppgaven med å evt delegere dette videre. Designene fra forrige iterasjon (koblingsskjema og programvare) er såpass godt at det skal være mulig for de andre å implementere dette.

2.3 Programvare-komplikasjoner

Det ligger ikke så mye utvikling av programvare for fartøyet under denne iterasjonen, men derimot mer for testbenken da vi er avhengig av å kunne gjøre en mengde målinger under utvikling av reguleringssystemet.

2.4 Tilgang til ressurser

Prosjektets mest fremtredende flaskehals er fartøyet da vi kun har ett av dette og samtlige av oss vil på ulike tidspunkt trenge tilgang til dette for å utføre ulike tester. Det er dermed viktig av vi er effektive under den tiden vi bruker fartøyet og lar andre slippe til i de periodene man kan.

3 Aktiviteter

Aktivitet	Ansvarlig	Frist	Ressurser	Artifakter	ID
Presentasjon 2	Herå Rørvik	2011-04-15	PC, Fartøyet, Alle medlemmer,	Presentasjon, Perm med dokumenter, CDer med alle dokumenter	111-2
Iterasjonsrapport 6	Michael Odden	2011-04-13	PC	Iterasjonsrapport, pdf	101-5
Iterasjonsplan 7	Michael Odden	2011-04-16	PC	Iterasjonsplan, pdf	
Sensor filtrering	Vegard Torkelsen	2011-04-13	PC, fartøyet	Sensorfiltreringsdokument	409-1
Forbedret radio-input (design)	Michael Odden	2011-04-08	PC, fartøyet	Utbedret design for regulert fartøy	
Matematisk modell	Luke Carambot	2011-04-08	PC	Dokument for matematisk modell	701

Tabell 2: Nærmere detaljer rundt aktivitetene finnes i prosjektplanen[1] samt på prosjektweb[2].

4 Aktiviteter, detaljert

For utfyllende informasjon om de ulike aktivitetene refereres det til ProsjektWeb [2].

5 Aktiviteter pr person

Her følger en forenklet oversikt over hvilke personer som anses som ressurs-personer ihht hvilke aktiviteter. Den må sees i sammenheng med aktivitetsoversikten ovenfor.

Person	Aktiviteter
<Samtlige>	<ul style="list-style-type: none"> Presentasjon 2 - alle som ikke har noen andre overhengende aktiviteter skal bidra til denne.
Vegard Torkelsen	<ul style="list-style-type: none"> Forbedret radio-input Iterasjonsplan 7
Patrick Bjørum	<ul style="list-style-type: none"> Presentasjon 2 Forbedret radio-input
Trine Lindberg	<ul style="list-style-type: none"> Presentasjon 2
Herå Rørvik	<ul style="list-style-type: none"> Presentasjon 2
Luke Carambot	<ul style="list-style-type: none"> Matematisk modell
Michael Odden	<ul style="list-style-type: none"> Iterasjonsrapport 6 Iterasjonsplan 7

	<ul style="list-style-type: none">• Forbedret radio-input
--	---

Tabell 3: De ulike aktivitetene hver person kan bidra til

6 Ressurser

- Personer
 - Alle gruppemedlemmene
- Utstyr
 - Personlig PC/laptop
 - Alminnelig relevant verktøy, loddeapparat, ledninger, lader mm.
 - Testbenk (vil utvikles etter behov gjennom iterasjonen)
- Komponenter - mikrokontroller, chassi, sensorer (gyro, akselerometer), batteri, fjernkontroll, radiomottaker
- Lokaler
 - Grupperom, C161 (Høgskolen i Buskerud avd Kongsberg)

Referanser

[1] Prosjektplan - prosjektplant_4.0.pdf

[2] Veikart fra ProsjektWeb - veikart_1.0.pdf

[3] Møtereferat fra veiledermøte - møtereferat_veileder_2011-03-10.pdf



HØGSKOLEN
i Buskerud

Oppgavetittel:

Hovedprosjekt - Iterasjonsplan Iterasjon 7 - v1.0
Periode 2011-04-14 - 2011-05-17

Fag(nr./navn):

SFHO 3200 / Prosjektstyring med hovedprosjekt

Gruppemedlemmer, signatur:

Vegard Torkelsen

Michael Odden

Herå Rørvik

Trine Lindberg

Luke Carambot

Patrick Bjørum

Intern veileder: Dag Samuelsen

Intern sensor: Olaf Hallan Graven

Ekstern veileder og sensor: Ragnvald Otterlei

Dato: 2011-05-04

Innholdsfortegnelse

Dokumenthistorie.....	1
Innledning.....	1
1 Mål.....	2
2 Risikoanalyse.....	2
2.1 Ødeleggelse av komponenter.....	2
2.2 Sykdom og annet fravær.....	2
2.3 Programvare-komplikasjoner.....	2
2.4 Tilgang til ressurser.....	2
3 Aktiviteter	3
4 Aktiviteter, detaljert.....	3
5 Aktiviteter pr person.....	4
6 Resurser.....	4
Referanser.....	6

Dokumenthistorie

Versjon	Beskrivelse	Når
1.0	Første offisielle versjon	2011-05-04

Tabell 1: Dokumenthistorie

Innledning

En iterasjonsplan er en detaljert oversikt over målene, aktivitetene og andre relevante opplysninger rundt en gitt iterasjon. Du vil i denne iterasjonsplanen finne overordnede mål, samt delmål og de aktivitetene som anses nødvendige for å nå disse. Aktivitetsoversikten viser til hvem som er ansvarlige for de ulike aktivitetene, hvilke ressurser som er nødvendige og hvilke artifakter som skal være de målbare resultatene av disse. Den overordnede aktivitetsoversikten er etterfulgt av en mer detaljert gjennomgang.

Vær obs på at personen ansvarlig for den enkelte aktivitet skal sørge for at:

- All nødvendig informasjon er samlet inn
- Progresjon oppdateres på ProsjektWeb
- Problemer tas opp med relevante personer og/eller gruppen
- Delegerer oppgaver hensiktsmessig til andre gruppemedlemmer
- Aktiviteten er gjennomført i tide
 - Skulle ikke dette la seg gjøre skal det dokumenteres og ny tidsfrist skal defineres så kjapt som mulig.
- Om noen aktiviteter er blitt i overkant store vil det være hensiktsmessig å dele den opp i sub-aktiviteter for bedre å måle fremgang.

1 Mål

Målet med iterasjon 7 er å implementere det forbedrede designet fra iterasjon 6. På slutten av denne iterasjonen skal det være mulig å fly det regulerte fartøyet!

Resultatet av denne iterasjonen vil hovedsaklig være:

- Simulert fartøy i Simulink
- Fungerende testbenk-programvare
- Flygbart regulert fartøy

2 Risikoanalyse

2.1 Ødeleggelse av komponenter

Det er en kontinuerlig risiko at komponenter kan bli ødelagt ved test og bruk. Det oppfordres til forsiktighet rundt håndteringen, samt påpasselighet med tanke på statisk elektrisitet, kortslutning, klipping mm.

I denne iterasjonen vil det bli en del flytesting, så det er viktig å være obs på dette.

2.2 Sykdom og annet fravær

Alle aktiviteter har minst to personer tilknyttet seg som ressurspersoner. Skulle den ansvarlige for aktiviteten bli syk eller av andre grunner ikke kunne fullføre sine oppgaver vil den neste ressurspersonen ta over dette ansvaret og dermed oppgaven med å evt. delegere dette videre. Designene fra forrige iterasjon (koblingsskjema og programvare) er såpass godt at det skal være mulig for de andre å implementere dette.

2.3 Programvare-komplikasjoner

Det ligger ikke så mye utvikling av programvare for fartøyet under denne iterasjonen, men derimot mer for testbenken da vi er avhengig av å kunne gjøre en mengde målinger under utvikling av reguleringssystemet.

Da de medfølgende bibliotekene til Arduino har ingen klart definerte rutiner for håndtering av hardware-interrupts for de analoge inngangene. Dette kan føre til komplikasjoner og økt tidsbehov ifm undersøkelser, feilsøking og korrigeringer.

2.4 Tilgang til ressurser

Prosjektets mest fremtredende flaskehals er fartøyet da vi kun har ett av dette og samtlige av oss vil på ulike tidspunkt trenge tilgang til dette for å utføre ulike tester. Det er dermed viktig av vi er effektive

under den tiden vi bruker fartøyet og lar andre slippe til i de periodene man kan. Hvis noen blir sittende å vente kan de starte på oppgaver som ligger i neste iterasjon.

3 Aktiviteter

Aktivitet	Ansvarlig	Frist	Ressurser	Artifakter	ID
Re-designe håndtering av radio-input	Michael	2011-05-10	Michael, Luke, PC		
Unit-tester	Michael	2011-05-10	Michael, PC	Testdokument	
Testing av reguleringssystemet	Herå	2011-05-17	Gruppemedlemmene, PC, fartøyet	Testdokument	601
Simulering av reguleringssystem	Trine	2011-05-12	Trine, Luke, Vegard, PC, Matlab	Simuleringsdokument	702
Feilsøking og forbedring	Trine	2011-05-17	Gruppemedlemmene, PC, fartøyet		406
Forbedring av matematisk modell	Luke	2011-05-17	Luke	Matematisk modell dokument	701
Implementering av reguleringssystem	Trine	2011-05-06	Trine, Vegard, Michael, PC		505
Parameterfremstilling for matematisk modell	Vegard	2011-05-08	Gruppemedlemmene, PC, testutstyr, fartøyet	Matematisk modell dokument	
Implementering av testbenk-protokoll i regulert fartøy	Michael	2011-05-06	Michael, PC		
Iterasjonsrapport 7	Michael	2011-05-17	Herå, PC	Iterasjonsrapport	114

Tabell 2: Nærmere detaljer rundt aktivitetene finnes i prosjektplanen[1] samt på prosjektweb[2].

4 Aktiviteter, detaljert

For utfyllende informasjon om de ulike aktivitetene refereres det til ProsjektWeb [2].

5 Aktiviteter pr person

Her følger en forenklet oversikt over hvilke personer som anses som ressurs-personer ihht hvilke aktiviteter. Den må sees i sammenheng med aktivitetsoversikten ovenfor.

Person	Aktiviteter
<Samtlige>	<ul style="list-style-type: none"> • Feilsøking og forbedring • Testing av regulerings-system
Vegard Torkelsen	<ul style="list-style-type: none"> • Parameterfremstilling for matematisk modell • Simulering av regulerings-system • Implementering av regulerings-system
Patrick Bjørum	<ul style="list-style-type: none"> • Parameterfremstilling for matematisk modell • Testing av regulerings-system
Trine Lindberg	<ul style="list-style-type: none"> • Simulering av regulerings-system • Implementering av regulerings-system
Herå Rørvik	<ul style="list-style-type: none"> • Testing av regulerings-system • Iterasjonsrapport 7
Luke Carambot	<ul style="list-style-type: none"> • Re-designe håndtering av radio-input • Simulering av regulerings-system • Forbedring av matematisk modell
Michael Odden	<ul style="list-style-type: none"> • Re-designe håndtering av radio-input • Unit-tester • Forbedret radio-input • Implementering av regulerings-system • Implementering av testbenk-protokoll i regulert fartøy

Tabell 3: De ulike aktivitetene hver person kan bidra til

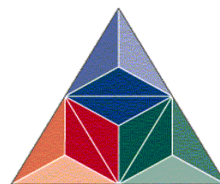
6 Ressurser

- Personer
 - Alle gruppemedlemmene
- Utstyr
 - Personlig PC/laptop
 - Alminnelig relevant verktøy, loddeapparat, ledninger, lader mm.
 - Testbenk (vil utvikles etter behov gjennom iterasjonen)
- Komponenter - mikrokontroller, chassis, sensorer (gyro, akselerometer), batteri, fjernkontroll, radiomottaker
- Lokaler
 - Grupperom, C161 (Høgskolen i Buskerud avd Kongsberg)

Referanser

[1] Prosjektplan - prosjektplan_4.0.pdf

[2] Veikart fra ProsjektWeb - veikart_1.0.pdf



HØGSKOLEN
i Buskerud

Oppgavetittel:

Hovedprosjekt - Iterasjonsplan Iterasjon9 - v1.0
Periode: 2011-05-17- 2011-06-09

Fag(nr./navn):

SFHO 3200 / Prosjektstyring med hovedprosjekt

Gruppemedlemmer, signatur:

Vegard Torkelsen

Michael Odden

Herå Rørvik

Trine Lindberg

Luke Carambot

Patrick Bjørum

Intern veileder: Dag Samuelsen

Intern sensor: Olaf Hallan Graven

Ekstern veileder og sensor: Ragnvald Otterlei

Dato: 2011-05-19

Innholdsfortegnelse

Dokumenthistorie.....	2
Innledning.....	4
Mål.....	5
Risikoanalyse.....	5
Ødeleggelse av komponenter.....	5
Sykdom og annet fravær.....	5
Programvare-komplikasjoner.....	5
Tilgang til ressurser.....	5
Aktiviteter	6
Aktiviteter, detaljert.....	6
Aktiviteter pr person.....	6
Ressurser.....	7
Referanser.....	8

Dokumenthistorie

Versjon	Beskrivelse	Når
1.0	Første offisielle versjon	2011-05-19

Tabell 1: Dokumenthistorie

Innledning

En iterasjonsplan er en detaljert oversikt over målene, aktivitetene og andre relevante opplysninger rundt en gitt iterasjon. Du vil i denne iterasjonsplanen finne overordnede mål, samt delmål og de aktivitetene som anses nødvendige for å nå disse. Aktivitetsoversikten viser til hvem som er ansvarlige for de ulike aktivitetene, hvilke ressurser som er nødvendige og hvilke artifakter som skal være de målbare resultatene av disse. Den overordnede aktivitetsoversikten er etterfulgt av en mer detaljert gjennomgang.

Vær obs på at personen ansvarlig for den enkelte aktivitet skal sørge for at:

- All nødvendig informasjon er samlet inn
- Progresjon oppdateres på ProsjektWeb
- Problemer taes opp med relevante personer og/eller gruppen
- Delegerer oppgaver hensiktsmessig til andre gruppemedlemmer
- Aktiviteten er gjennomført i tide
 - Skulle ikke dette la seg gjøres skal det dokumenteres og ny tidsfrist skal defineres så kjapt som mulig.
- Om noen aktiviteter er blitt i overkant store vil det være hensiktsmessig å dele den opp i sub-aktiviteter for bedre å måle fremgang.

1 Mål

Målet med iterasjon 9 er å ferdigstille fartøyet og prosjektet. På slutten av denne iterasjonen skal det være enkelt å fly det regulerte fartøyet i stabil modus, og all dokumentasjon skal være levert. Presentasjon 3 skal også planlegges, og iterasjonen slutter med selve presentasjonen, og da har vi også nådd den siste milepælen i prosjektet.

Resultatene av denne iterasjonen vil hovedsaklig være:

- Ferdigstilt, enkelt flygbart fartøy
- Ferdigstilt dokumentasjon
- Plakat
- Presentasjon 3

2 Risikoanalyse

2.1 Ødeleggelse av komponenter

Det er en kontinuerlig risiko at komponenter kan bli ødelagt ved test og bruk. Det oppfordres til forsiktighet rundt håndteringen.

I denne iterasjonen vil det bli en del flytesting, så det er viktig å være obs på dette.

2.2 Sykdom og annet fravær

Alle aktiviteter har minst to personer tilknyttet seg som ressurspersoner. Skulle den ansvarlige for aktiviteten bli syk eller av andre grunner ikke kunne fullføre sine oppgaver vil den neste ressurspersonen ta over dette ansvaret og dermed oppgaven med å evt delegere dette videre.

2.3 Programvare-komplikasjoner

Det meste av programvaren er nå ferdig, og tid er brukt på å gjøre koden robust. Skulle likevel komplikasjoner oppstå blir det satt inn ressurser til å feilsøke.

2.4 Tilgang til ressurser

Prosjektets mest fremtredende flaskehals er fartøyet da vi kun har ett av dette, men i denne iterasjonen er det også mye dokumentarbeid og planlegging av presentasjon 2 så ressursene kan fordeles ut over flere oppgaver så ikke alle planlegger å holde på med fartøyet samtidig.

3 Aktiviteter

Aktivitet	Ansvarlig	Frist	Ressurser	Artifakter/resultater	ID
Presentasjon 3	Trine	2011-06-09	Alle på gruppa, PC, prosjektor	Presentasjon 3	111-3
Prosjektplakat	Michael	2011-06-06	Michael, Patrick, PC	Prosjektplakat	109
Analyse	Luke	2011-05-27	Luke, PC	Analysedokument	303
Ferdigstilling til innlevering	Vegard	2011-05-27	Vegard, fartøyet, PC	Ferdigstilt fartøy og prosjekt	503
Tuning og testing av reguleringsystem	Luke	2011-05-24	Gruppemedlemmene, PC, fartøyet	Ferdig tunet reguleringsystem	503-1
Ferdigstilling av dokumenter	Herå	2011-05-27	Gruppemedlemmene, PC	Ferdige dokumenter	503-2
Testing av kameramodul	Patrick	2011-05-06	Patrick, Vegard, PC, kameramodul	Fungerende kamera med live overføring	505
Bruerveiledning	Vegard	2011-05-27	Vegard, Patrick, PC	Brukermanual-dokument	606

Tabell 2: Nærmere detaljer rundt aktivitetene finnes i prosjektplanen[1] samt på prosjektweb[2].

4 Aktiviteter, detaljert

For utfyllende informasjon om de ulike aktivitetene refereres det til ProsjektWeb [2].

5 Aktiviteter pr person

Her følger en forenklet oversikt over hvilke personer som anses som ressurs-personer ihht hvilke aktiviteter. Den må sees i sammenheng med aktivitetsoversikten ovenfor.

Person	Aktiviteter
<Samtlige>	<ul style="list-style-type: none">• Testing og tuning av reguleringsystemet• Ferdigstilling av dokumenter• Presentasjon 3
Vegard Torkelsen	<ul style="list-style-type: none">• Ferdigstilling til innlevering• Brukerveiledning• Testing av kameramodul
Patrick Bjørum	<ul style="list-style-type: none">• Testing av kameramodul• Prosjektplakat• Brukerveiledning• Presentasjon 3: Film
Trine Lindberg	<ul style="list-style-type: none">• Presentasjon 3
Herå Rørvik	<ul style="list-style-type: none">• Ferdigstilling av dokumenter
Luke Carambot	<ul style="list-style-type: none">• Analysedokument• Tuning og testing av reguleringsystem
Michael Odden	<ul style="list-style-type: none">• Prosjektplakat• Presentasjon 3: Promo-plakat

Tabell 3: De ulike aktivitetene hver person kan bidra til

6 Ressurser

- Personer
 - Alle gruppe-medlemmene
- Utstyr
 - Personlig PC/laptop
 - Alminnelig relevant verktøy, loddeapparat, ledninger, lader, multimeter mm.
 - Testbenk (vil utvikles etter behov gjennom iterasjonen)
- Komponenter - mikrokontroller, chassi, sensorer (gyro, aksellerometer), batteri, fjernkontroll, radiomottaker
- Lokaler
 - Grupperom, C161 (Høgskolen i Buskerud avd Kongsberg)
 - Brukersenteret

Referanser

[1] Prosjektplan - prosjektplan_4.0.pdf

[2] Veikart fra ProsjektWeb - veikart_1.0.pdf



HØGSKOLEN
i Buskerud

Oppgavetittel:

Hovedprosjekt - Iterasjonsrapport iterasjon2 - v2.0
Periode: 2011-01-07 - 2011-01-20

Fag(nr./navn):

SFHO 3200 / Prosjektstyring med hovedprosjekt

Gruppemedlemmer, signatur:

Vegard Torkelsen

Michael Odden

Herå Rørvik

Trine Lindberg

Luke Carambot

Patrick Bjørum

Intern veileder: Dag Samuelsen

Intern sensor: Olaf Hallan Graven

Ekstern veileder og sensor: Ragnvald Otterlei

Dato: 2011-04-06

Innholdsfortegnelse

Dokumenthistorie.....	1
Innledning.....	1
1 Aktiviteter fra iterasjonsplan.....	2
2 Overholdelse.....	2
3 utfordringer og tiltak.....	3
Konklusjon.....	4

Dokumenthistorie

Versjon	Beskrivelse	Når
1.0	Første versjon	2011-01-23
2.0	Endret tittel på "Problemer og lærepenger" til "Utfordringer og tiltak"	2011-04-06

Tabell 1: Dokumenthistorie

Innledning

En iterasjonsrapport oppsummerer en avsluttet iterasjon og skal analysere prosessen dit. I denne rapporten har vi lagt med aktivitets-oversikten slik den fremkom i iterasjonsplanen, samt at vi tar for oss problemer vi har hatt, hvor godt vi har overholdt planen og eventuelle justeringer videre.

1 Aktiviteter fra iterasjonsplan

Aktivitet	Ansvarlig	Frist	Ressurser	Artifakter	ID
Design av koblingsskjema	Patrick Bjørum	2011-01-15	PC, Patrick Bjørum, Vegard Torkelsen	Koblingsskjema	407
Design av programvare for uregulert fartøy	Michael Odden	2011-01-19	PC, Michael Odden, Mikrokontroller	UML-diagrammer <ul style="list-style-type: none"> ○ Use case ○ Klasse ○ Sekvens ○ Tilstand 	401-1
Testbenk	Luke Carambot	2011-01-15	Fartøyet, verktøy, Luke Carambot, Trine Lindberg,	Gjenstand: Testbenk	410
Testplan	Luke Carambot	2011-01-13	Luke Carambot, Herå Rørvik, Trine Lindberg	Dokument: Testplan	603
Konstruksjon av prototype	Patrick Bjørum	2011-01-17	Fartøyet, Patrick Bjørum, Vegard Torkelsen	Gjenstand: Prototype	501
Iterasjonsrapport	Michael Odden	2011-01-20	PC, Michael Odden	Dokument: Iterasjonsrapport	101-1

Tabell 2: Oversikt over aktiviteter for iterasjonen.

2 Overholdelse

Vi har i denne iterasjonen produsert følgende artifakter:

- Koblingsskjema (<Vegard>)
- UML-diagrammer (umldesign_1.0.pdf)
 - Use Case-diagram
 - Klassediagram
 - Sekvensdiagram
 - Tilstandsdiagram
- Testbenk
 - Dette er en grov versjon som vil forbedres utover i prosessen.

- Testplan (testplan_1.1.pdf)
- Prototype (fysisk gjenstand)
 - Vil forbedres inntil den er vårt ferdige produkt.
- Iterasjonsrapport (dette dokumentet: iterasjonsrapport_iterasjon2_1.0.pdf)

3 utfordringer og tiltak

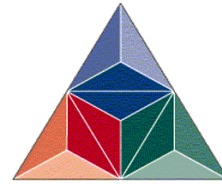
Da vi begynte med 2. iterasjon hadde vi mer å lære angående strukturering av en iterasjon og tok ganske pent på hvordan dette skulle gjennomføres. Dette førte til enkelte situasjoner hvor enkelte ikke visste hva (om noe) de skulle ta seg til. For å unngå at dette skjer fremover har vi kommet frem til å ha en åpnere dialog oss imellom, samt at de neste iterasjonsplanene i større grad vil gå innpå dette. Det vil naturligvis være perioder med ulik belastning på den enkelte, men dette bør isåfall komme bedre frem.

Pga ovennevnte uvisshet så førte det også til at noe av fokuset var litt feil mtp at man heller jobbet med det morsomme (bygging, prototyping) istedet for det mest nødvendige og et par av design-planene ble ferdige kort tid etter iterasjonens slutt-dato.

Konklusjon

Når vi nå går over i Iterasjon 3 føler vi at vi har en klar formening om hvordan vi skal gå frem videre. Vi har en ganske klar oversikt over komponentene, deres funksjon og fartøyets design.

Som nevnt under "Problemer og lærepenge" vil vi formulere de neste iterasjonsplanene mer nøyaktig for å unngå misnøye og bedre treffsikkerheten mtp tidsfrister. Vi er ellers i god rute tidsmessig.



HØGSKOLEN
i Buskerud

Oppgavetittel:

Hovedprosjekt - Iterasjonsrapport iterasjon 3 - v2.0
Periode: 2011-01-21 - 2011-02-21

Fag(nr./navn):

SFHO 3200 / Prosjektstyring med hovedprosjekt

Gruppemedlemmer, signatur:

Vegard Torkelsen

Michael Odden

Herå Rørvik

Trine Lindberg

Luke Carambot

Patrick Bjørum

Intern veileder: Dag Samuelson

Intern sensor: Olaf Hallan Graven

Ekstern veileder og sensor: Ragnvald Otterlei

Dato: 2011-04-06

Innholdsfortegnelse

Dokumenthistorie.....	1
Innledning.....	1
1 Aktiviteter fra iterasjonsplan.....	2
2 Overholdelse.....	2
3 Utfordringer og tiltak.....	3
Konklusjon.....	4
Referanser.....	5

Dokumenthistorie

Versjon	Beskrivelse	Når
1.0	Første offentlige versjon	2011-02-22
2.0	Mindre endringer på aktivitetsliste og ny overskrift for "lærepenge"-kapittelet.	2011-04-06

Tabell 1: Dokumenthistorie

Innledning

En iterasjonsrapport oppsummerer en avsluttet iterasjon og skal analysere prosessen dit. I denne rapporten har vi lagt med aktivitets-oversikten slik den fremkom i iterasjonsplanen, samt at vi tar for oss problemer vi har hatt, hvor godt vi har overholdt planen og eventuelle justeringer videre.

Hovedmålet med denne iterasjonen har vært å implementere det uregulerte fartøyet, samt implementere støttesystemer via testbenken.

1 Aktiviteter fra iterasjonsplan

Aktivitet	Ansvarlig	Frist	Ressurser	Artifakter	ID
Utvikling av programvare	Michael Odden	2011-02-19	PC, Michael Odden, Luke Carambot, Mikrokontroller	Arduino sketch (c++)	404-1
Forbedring av prototype	Vegard Torkelsen	2011-02-11	Fartøyet, PC, Vegard Torkelsen, Patrick Bjørum, Trine Lindberg, Herå Rørvik	Prototype	501-2
Testing av prototype	Herå Rørvik	2011-02-21	Fartøyet, PC, testbenk, Herå Rørvik, Luke Carambot, Patrick Bjørum	Testrapport / logg	601-1
Regnskap	Patrick Bjørum	2011-02-07	PC, Patrick Bjørum	Dokument over regnskapet	106-2
Iterasjonsrapport 3	Michael Odden	2011-02-23	PC, Michael Odden, Trine Lindberg	Dokument: Iterasjonsrapport	101-2
Testbenk	Luke Carambot	2011-02-xx	Luke Carambot, Michael Odden	Testbenk, Kontrollprogram	502

Tabell 2: Nærmere detaljer rundt aktivitetene finnes i prosjektplanen[1] samt på ProsjektWeb[2].

2 Overholdelse

Vi har i denne iterasjonen produsert følgende artifakter:

- Foreløpig regnskap - Regnskap_V1.0.ods, Denne vil oppdateres gjennom prosjektet.
- Forbedret prototype. Fysisk konstruksjon fullført.
- Programvare for testbenk.
- Programvare for uregulert system ihht design
 - commit: 3f14cb1acd1423b4584cf6e90fe250f73ee9f29c
 - Denne må forbedres på enkelte punkter, men det vil bli ytterligere aktiviteter.
- Iterasjonsrapport
 - Dette dokumentet: iterasjonsrapport_iterasjon3_1.0.pdf

Andre vesentlige resultater/produkter:

- Arbeidsdokument for testbenk og seriell-kommunikasjon på Wiki [9],[10]
- Innhentet testdata ifm turtall [8] og løftkraft [7]
- Utarbeidet matematisk modell for roll og pitch - ikke ferdigstilt dokument

3 utfordringer og tiltak

Mot slutten av iterasjonen dukket det opp problemer med at input-signaler tilsynelatende ikke blir sendt raskt nok ut til ESCene, dette må vi utforske som egen del-aktivitet i neste iterasjon. I tillegg har det vært en del problemer knyttet opp mot serielltilkoblingen mellom ulike OS og fartøyet - dette har vært OS-spesifikke problemer som vi nå har løst takket være spesial-kompilerte utgaver av RXTX-biblioteket [6].

Vi har også tittet på muligheter for å senke tyngdepunktet en anelse for å gjøre det lettere å fly og har i den forbindelse montert et ekstra batteri under fartøyet.

Det har vist seg å være vanskelig å få til pålitelige tester siden alle forsøk på test-rigger har negative konsekvenser da vi alltid må tenke sikkerhet og dermed begrenser presisjon for testing av flyegegenskaper.

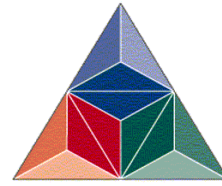
Konklusjon

Vi er nå i mål med en prototype som implementerer et uregulert system hvor vi kun ved statiske skaleringer av motor-krefter og radio-input har gjort den - i teorien - flyvbar. Problemet med et slikt uregulert fartøy er at den er meget vanskelig å kontrollere og krever stor plass som sådan for å kunne utføre reguleringen manuelt.

Nå går vi over i iterasjon 4 hvor vi skal designe det regulerede systemet, hvilket har som mål å benytte data fra akselerometer og gyroskop for å oppnå en langt mer stabil hover og gjøre den enklere å manøvrere.

Referanser

- [1] Prosjektplan - prosjektplan_4.0.pdf
- [2] Veikart fra ProsjektWeb - veikart_1.0.pdf
- [3] Testplan - testplan_2.0.pdf
- [4] AeroQuad Store, <https://www.aeroquadstore.com/> - 2011-01-30
- [5] iterasjonsplan_iterasjon3_1.1.odt
- [6] RXTX, <http://users.frii.com/jarvi/rxtx/> - 2011-02-20
- [7] Thrusttest av motorer
- [8] Turtallstest for motorer
- [9] Wiki Testbenk, <http://git.projectdragonfly.no:3000/projects/dragonfly/wiki/Testbenk> - 2011-02-21
- [10] Wiki Serial Protocol, http://git.projectdragonfly.no:3000/projects/dragonfly/wiki/Serial_Protocol - 2011-02-21



HØGSKOLEN
i Buskerud

Oppgavetittel: Hovedprosjekt - Iterasjonsrapport iterasjon 4 - v1.0 <i>Periode: 2011-02-22 - 2011-03-17</i>		
Fag(nr./navn): SFHO 3200 / Prosjektstyring med hovedprosjekt		
Gruppemedlemmer, signatur:		
_____ Vegard Torkelsen	_____ Michael Odden	_____ Herå Rørvik
_____ Trine Lindberg	_____ Luke Carambot	_____ Patrick Bjørum
Intern veileder: Dag Samuelson Intern sensor: Olaf Hallan Graven Ekstern veileder og sensor: Ragnvald Otterlei		
Dato: 2011-03-17		

Innholdsfortegnelse

Dokumenthistorie.....	1
Innledning.....	1
1 Aktiviteter fra iterasjonsplanen.....	2
2 Overholdelse.....	3
3 Utfordringer og tiltak.....	3
Konklusjon.....	4
Referanser.....	5

Dokumenthistorie

Versjon	Beskrivelse	Når
1.0	Første offentlige versjon	2011-03-17

Innledning

En iterasjonsrapport oppsummerer en avsluttet iterasjon og skal analysere prosessen dit. I denne rapporten har vi lagt med aktivitets-oversikten slik den fremkom i iterasjonsplanen, samt at vi tar for oss problemer vi har hatt, hvor godt vi har overholdt planen og eventuelle justeringer videre.

Hovedmålet med denne iterasjonen har vært å designe foreløpig utgave av det regulerte fartøyet. Ved slutten av denne iterasjonen var målet å ha inne et enkelt regulerings-system samt avdekke de viktigste områdene vi må jobbe videre med.

1 Aktiviteter fra iterasjonsplanen

Aktivitet	Ansvarlig	Frist	Ressurser	Artifakter	ID
Presentasjon 2	Herå Rørvik	2011-03-17	PC, Fartøyet, Alle medlemmer,	Flyttet til iterasjon 6	111-2
Design av programvare for regulert fartøy	Michael Odden	2011-03-16	PC, Luke Carambot, Fartøyet,	UML-diagrammer, overordnet design-dokument	401-2
Iterasjonsrapport 4	Michael Odden	2011-03-18	PC	Iterasjonsrapport	101-3
Design av matematisk modell	Patrick Bjørum	2011-03-10	PC, Luke Carambot,	Dokumentasjon av modell	408
Design av signalbehandling	Vegard Torkelsen	2011-02-28	PC, Fartøyet	Designdokument for signalbehandling	409
Matematisk modell	Luke Carambot	2011-03-10	PC, Patrick Bjørum,	Matematisk modell	701
Design av reguleringsystem	Trine Lindberg	2011-03-10	PC, Fartøyet,	Designdokument for reguleringsystem	406-1
Simulering av reguleringsystem	Trine Lindberg	2011-03-16	PC, Designdokument for reguleringsystem (406-1),	Simulator	702
Iterasjonsplan Iterasjon 5	Michael Odden	2011-03-17	PC, Vegard Torkelsen	Iterasjonsplan	113-5
Utbedre testbenk for uregulert fartøy	Luke Carambot	2011-03-17	PC	Oppdatert testbenk-programvare	502-2
Design testbenk-utvidelse for regulert fartøy	Herå Rørvik	2011-03-10	PC	Designdokument for utvidelser	410-1
Utbedre signalproblemer	Michael Odden	2011-03-05	PC, Fartøyet	Oppdatert programvare	404-3
Design av nye tester	Patrick Bjørum	2011-03-03	PC		601-7
Ombygging for lavere COG	Patrick Bjørum	2011-03-01	PC, Fartøyet	Ombygd prototype	501-2

[tabell 2 - Nærmere detaljer rundt aktivitetene finnes i prosjektplanen[1] samt på prosjektweb[2].]

2 Overholdelse

Vi har i denne iterasjonen produsert følgende artifakter:

- Designdokumenter:
 - Programvare
 - Reguleringsystem
 - Matematisk modell
- Programvare
 - Utbedret signalproblemer: *Revisjon 54190d40*
 - Utbedret testbenk: *Revisjon 31157a88*
- Komponenter
 - Lett ombygd fartøy med ekstra batteri for nyttevekt og lavere COG.
- Iterasjonsrapport
 - Dette dokumentet: `iterasjonsrapport_iterasjon4_1.0.pdf`

3 Utfordringer og tiltak

Underveis oppdaget vi at det ville være litt for "vannfallsmodell" å designe og implementere hele det regulerte systemet i én vending (les: to iterasjoner), spesielt med tanke på at det fortsatt er en del faktorer vi ikke har helt overblikk over enda og trenger å teste og prototype noe mer før vi kan komme til noen endelig løsning. Vi er likevel avhengig av å implementere deler av det regulerte systemet, slik som PID, enkelte filtre mm for å teste tilstrekkeligheten av bl.a. den matematiske modellen og reguleringsystemet.

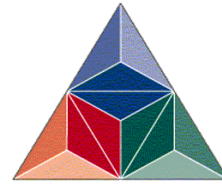
Konklusjon

Vi har nå designet et fartøy som skal implementere et relativt enkelt reguleringssystem. Vi avdekket underveis at det kunne være behov for å gjøre design og implementasjon av det regulerte systemet i to omganger hvor første runde med design og implementasjon hadde som oppgave å avdekke svakheter og potensielle problemstillinger da det var et par nøkkelfaktorer vi ikke hadde nok informasjon om uten å utføre en del testing og prototyping først. Vi har dermed forskjøvet de to siste iterasjonen[4] og lagt til en ny iterasjon 6 og 7 som respektivt er forbedret design basert på oppdagelsene i iterasjon 5, og iterasjon 7 er da implementasjon av disse.

Vi går nå over i iterasjon 5 hvor vi skal implementere dette så langt det går med den hensikt å avdekke områder vi må redesigne og re-implementere i iterasjon 6 og 7. Iterasjonen har ellers vært uten store problemer.

Referanser

- [1] Prosjektplan - prosjektplan_4.0.pdf
- [2] Veikart fra ProsjektWeb - veikart_1.0.pdf
- [3] iterasjonsplan_iterasjon4_1.0.pdf
- [4] møtereferat_veileder_2011-03-10.pdf



HØGSKOLEN
i Buskerud

Oppgavetittel: Hovedprosjekt - Iterasjonsrapport iterasjon 5 - v1.0 <i>Periode: 2011-03-18 - 2011-03-30</i>		
Fag(nr./navn): SFHO 3200 / Prosjektstyring med hovedprosjekt		
Gruppemedlemmer, signatur:		
_____ Vegard Torkelsen	_____ Michael Odden	_____ Herå Rørvik
_____ Trine Lindberg	_____ Luke Carambot	_____ Patrick Bjørum
Intern veileder: Dag Samuelson Intern sensor: Olaf Hallan Graven Ekstern veileder og sensor: Ragnvald Otterlei		
Dato: 2011-03-30		

Innholdsfortegnelse

Dokumenthistorie.....	1
Innledning.....	1
1 Aktiviteter fra iterasjonsplanen.....	2
2 Overholdelse.....	2
3 utfordringer og tiltak.....	2
Konklusjon.....	3
Referanser.....	4

Dokumenthistorie

Versjon	Beskrivelse	Når
1.0	Initiell versjon	2011-03-30

Tabell 1: Dokumenthistorie

Innledning

En iterasjonsrapport oppsummerer en avsluttet iterasjon og skal analysere prosessen dit. I denne rapporten har vi lagt med aktivitets-oversikten slik den fremkom i iterasjonsplanen, samt at vi tar for oss problemer vi har hatt, hvor godt vi har overholdt planen og eventuelle justeringer videre.

Hovedmålet med denne iterasjonen har vært å implementere en foreløpig utgave av det regulerte fartøyet. Ved slutten av denne iterasjonen hadde vi implementert en oppdatert programvare med et enkelt reguleringssystem som leser fra akselerometer og gyroskop samt regulerer disse verdiene via PID-regulatorer for hver akse.

Disse reguleringene er på ingen måte tilstrekkelige og det var heller ikke målet med iterasjonen. Vi har gjennom dette avdekket hvilke områder vi må legge mest vekt på fremover.

1 Aktiviteter fra iterasjonsplanen

Aktivitet	Ansvarlig	Frist	Ressurser	Artifakter	ID
Programvare for regulert fartøy	Michael Odden	2011-04-28	PC, fartøy	Oppdatert kildekode.	404-2
Implementering av signalbehandling	Vegard Torkelsen	2011-03-24	PC, fartøy	Klasser: BMA180, ITG3200	504
Implementering av reguleringsystem	Trine Lindberg	2011-03-29	PC, fartøy	Klasser: Regulator, PID	505
Testing av reguleringsystem	Herå Rørvik	2011-03-30	PC, fartøy	Testdokumenter etter behov	601-2
Iterasjonsrapport 5	Michael Odden	2011-04-30	PC	Dokument, iterasjonsrapport	114-5

Tabell 2: Nærmere detaljer rundt aktivitetene finnes i prosjektplanen[1] samt på ProsjektWeb[2].

2 Overholdelse

Vi har i denne iterasjonen produsert følgende artifakter:

- Programvare
 - Klassene: BMA180, ITG3200, PID, AvgFilter, Regulation
- Iterasjonsrapport
 - Dette dokumentet: iterasjonsrapport_iterasjon5_1.0.pdf
- Dokumenter
 - Testrapport for iterasjon 5 [4].

3 Utfordringer og tiltak

Det har ikke vært noen utfordringer av størrelse i forhold til gjennomføringen av iterasjon 5.

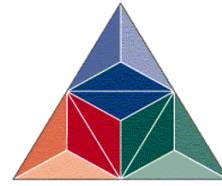
Konklusjon

Under denne iterasjonen har vi avdekket at hovedproblemene vi må fokusere på videre er:

- Samkjøring av sensordata fra akselerometer og gyroskop da begge har problemer hver for seg:
 - Akselerometerets målinger blir feil når selve fartøyet akselererer
 - Gyroskopets integrerte vinkelhastighet drifter over tid
- Arduino's pulseIn()-funksjon skaper for store forsinkelser

Referanser

- [1] Prosjektplan - prosjektplan_4.0.pdf
- [2] Veikart fra ProsjektWeb - veikart_1.0.pdf
- [3] iterasjonsplan_iterasjon5_1.0.pdf
- [4] testrapport_5_1.0.pdf



HØGSKOLEN
i Buskerud

Oppgavetittel: Hovedprosjekt - Iterasjonsrapport iterasjon 6 - v1.0 <i>Periode: 2011-03-31 - 2011-04-13</i>		
Fag(nr./navn): SFHO 3200 / Prosjektstyring med hovedprosjekt		
Gruppemedlemmer, signatur:		
_____ Vegard Torkelsen	_____ Michael Odden	_____ Herå Rørvik
_____ Trine Lindberg	_____ Luke Carambot	_____ Patrick Bjørum
Intern veileder: Dag Samuelson Intern sensor: Olaf Hallan Graven Ekstern veileder og sensor: Ragnvald Otterlei		
Dato: 2011-05-04		

Innholdsfortegnelse

Dokumenthistorie.....	1
Innledning.....	1
1 Aktiviteter fra iterasjonsplanen.....	2
2 Overholdelse.....	2
3 Utfordringer og tiltak.....	2
Konklusjon.....	3
Referanser.....	4

Dokumenthistorie

Versjon	Beskrivelse	Når
1.0	Første offentlige versjon	2011-05-04

Tabell 1: Dokumenthistorie

Innledning

En iterasjonsrapport oppsummerer en avsluttet iterasjon og skal analysere prosessen dit. I denne rapporten har vi lagt med aktivitets-oversikten slik den fremkom i iterasjonsplanen, samt at vi tar for oss problemer vi har hatt, hvor godt vi har overholdt planen og eventuelle justeringer videre.

Målet med iterasjon 6 var å forbedre designet på de punktene oppdaget som nødvendige under implementeringene i iterasjon 5. Disse vil da re-implementeres i iterasjon 7.

Viktigste resultater av denne iterasjonen:

- Spesialiserte design-dokumenter samt oppdaterte overordnede design-dokumenter der dette inngår.
- Presentasjon 2

1 Aktiviteter fra iterasjonsplanen

Aktivitet	Ansvarlig	Frist	Ressurser	Artifakter	ID
Presentasjon 2	Herå Rørvik	2011-04-15	PC, Fartøyet, Alle medlemmer,	Presentasjon, Perm med dokumenter, CDer med alle dokumenter	111-2
Iterasjonsrapport 6	Michael Odden	2011-04-13	PC	Iterasjonsrapport, pdf	101-5
Iterasjonsplan 7	Michael Odden	2011-04-16	PC	Iterasjonsplan, pdf	
Sensor filtrering	Vegard Torkelsen	2011-04-13	PC, fartøyet	Sensorfiltreringsdokument	409-1
Forbedret radio-input (design)	Michael Odden	2011-04-08	PC, fartøyet	Utbedret design for regulert fartøy	
Matematisk modell	Luke Carambot	2011-04-08	PC	Dokument for matematisk modell	701

Tabell 2: Nærmere detaljer rundt aktivitetene finnes i prosjektplanen[1] samt på prosjektweb[2]

2 Overholdelse

Vi har i denne iterasjonen produsert følgende artifakter:

- Designdokumenter:
 - Reguleringsystem
 - Matematisk modell
 - Sensorfiltrering

Andre vesentlige resultater/produkter:

- Presentasjon 2
 - alle dokumentasjon oppdatert til innlevering.

3 Utfordringer og tiltak

Det viste seg å være en utfordring å finne alle parametre til den matematiske modellen, så arbeidet med denne fortsetter ut i neste iterasjon. Presentasjon 2 tok en del tid, så iterasjonsrapport for iterasjon 6 og -plan for iterasjon 7 ble nedprioritert og utsatt og delegert videre til ledige ressurser i neste iterasjon.

Konklusjon

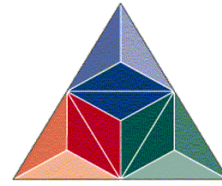
Vi har nå designet og skal implementere et forbedret reguleringsystem, samt sensorfiltrering og forbedret radio-input. Vi har merket at tiden begynner å bli knapp, og vi har derfor valgt å utvide iterasjon 7, og kutte ut iterasjon 8 som kun gjelder ekstra-funksjoner og c-krav. Milepæl 3 er da også flyttet, til sluttdatoen for iterasjon 7. Iterasjon 9 og milepæl 4 er også forskjøvet til absolutt siste innleveringsfrist som er 27.mai.

Hovedmålene med denne iterasjonen er oppfylt, og vi går nå over i iterasjon 7 hvor vi skal implementere det vi har designet.

Referanser

[1] Prosjektplan - prosjektplan_4.0.pdf

[2] Veikart fra ProsjektWeb - veikart_1.0.pdf



HØGSKOLEN
i Buskerud

Oppgavetittel: Hovedprosjekt - Iterasjonsrapport iterasjon 7 - v1.0 <i>Periode: 2011-04-13 - 2011-05-17</i>		
Fag(nr./navn): SFHO 3200 / Prosjektstyring med hovedprosjekt		
Gruppemedlemmer, signatur:		
_____ Vegard Torkelsen	_____ Michael Odden	_____ Herå Rørvik
_____ Trine Lindberg	_____ Luke Carambot	_____ Patrick Bjørum
Intern veileder: Dag Samuelson Intern sensor: Olaf Hallan Graven Ekstern veileder og sensor: Ragnvald Otterlei		
Dato: 2011-01-06		

Innholdsfortegnelse

Dokumenthistorie.....	1
Innledning.....	1
1 Aktiviteter fra iterasjonsplanen.....	1
2 Overholdelse.....	2
3 Utfordringer og tiltak.....	3
Konklusjon.....	4
Referanser.....	5

Dokumenthistorie

Versjon	Beskrivelse	Når
1	Første offisielle versjon	2011-05-19

Tabell 1: Dokumenthistorie

Innledning

En iterasjonsrapport oppsummerer en avsluttet iterasjon og skal analysere prosessen dit. I denne rapporten har vi lagt med aktivitets-oversikten slik den fremkom i iterasjonsplanen, samt at vi tar for oss problemer vi har hatt, hvor godt vi har overholdt planen og eventuelle justeringer videre.

Målet med iterasjon 7 var å implementere det forbedrede designet fra iterasjon 6. På slutten av denne iterasjonen skulle det være mulig å fly det regulerte fartøyet!

Viktigste resultater av denne iterasjonen:

- Simulert fartøy i Simulink
- Fungerende testbenk-programvare og GUI
- Bedre radio-input avlesning
- Flygbart regulert fartøy

1 Aktiviteter fra iterasjonsplanen

Aktivitet	Ansvarlig	Frist	Ressurser	Artifakter	ID
Re-designe håndtering av radio-input	Michael	2011-05-10	Michael, Luke, PC		
Unit-tester	Michael	2011-05-10	Michael, PC	Testdokument	
Testing av reguleringsystemet	Herå	2011-05-17	Gruppemedlemmene, PC, fartøyet	Testdokument	601
Simulering av reguleringsystem	Trine	2011-05-12	Trine, Luke, Vegard, PC, Matlab	Simuleringsdokument	702
Feilsøking og forbedring	Trine	2011-05-17	Gruppemedlemmene, PC, fartøyet		406
Forbedring av matematisk modell	Luke	2011-05-17	Luke	Matematisk modell dokument	701
Implementering av reguleringsystem	Trine	2011-05-06	Trine, Vegard, Michael, PC		505
Parameterfremstilling for matematisk modell	Vegard	2011-05-08	Gruppemedlemmene, PC, testutstyr, fartøyet	Matematisk modell dokument	
Implementering av testbenk-protokoll i regulert fartøy	Michael	2011-05-06	Michael, PC		
Iterasjonsrapport 7	Michael	2011-05-17	Herå, PC	Iterasjonsrapport	114

Tabell 2: Nærmere detaljer rundt aktivitetene finnes i prosjektplanen[1] samt på prosjektweb [2].

2 Overholdelse

Stort sett har vi overholdt tidsfristene etter planen. Den matematiske modellen har fått noe lavere prioritet i denne iterasjonen da reguleringsystem, GUI og testing har krevd alle ressurser.

Vi har i denne iterasjonen produsert følgende artifakter:

- Dokumenter:
 - Oppdatert designdokumentet for reguleringsystem
 - Simuleringsdokument
 - Sensorfusjon – Kalman filter dokument
 - Iterasjonsrapport 7
 - Iterasjonsplan 9

Andre vesentlige resultater/produkter:

- Fikset bedre radio-avlesning
- Påbygd reguleringsystemet og implementert dette
- Simulert fartøyet i Simulink
- Fungerende testbenk-programvare og GUI
- Bedre sensorsignaler – implementert Kalman filter
- Flygbart fartøy

3 utfordringer og tiltak

Å finne parametre til den matematiske modellen ble utsatt sist, og har blitt nedprioritert også i denne iterasjonen da vi har ansett det som viktigere å få reguleringsystemet i orden. Det kommer til å tas opp igjen i neste iterasjon hvis vi får tid.

Når det gjaldt sensorfiltrering fant vi ut at det filteret vi hadde ikke fjernet støy tilstrekkelig. Vi prøvde derfor å implementerte et Kalman filter og, og det viste seg å gi mye bedre resultater.

Angående reguleringsystemet oppdaget vi at vinkelregulering alene ikke er tilstrekkelig for å få fartøyet stabilt, så vi tok en ny research-runde, og fant ut at vi måtte regulere mot vinkelhastigheter i første omgang, og deretter i tillegg mot vinkler i roll og pitch. Vi bygde derfor på reguleringsystemet, først testet vi kun med rotasjonsregulering(vinkelhastigheter) hvor tilbakekoblingen kommer kun fra gyroen. Dette kaller vi gyro-modus. Vi tunet PIDene i roll og pitch i denne modusen, og fikk til å fly ganske greit med en erfaren flyger bak kontrollen! Deretter implementerte vi vinkelregulering i roll og pitch. Etter litt trøbbel med implementering av vinkelregulering(stabil modus) kunne vi til slutt starte å tune disse PIDene.

Konklusjon

Vi har nå bygd på designet av reguleringsystemet og implementert dette, samt fått til en bedre sensorfiltrering og forbedret radio-input. Vi har også fått til gode simuleringer, og har en godt fungerende testbenk-programvare med GUI.

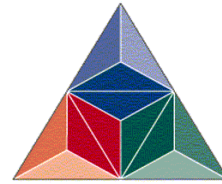
Vi har et flygbart fartøy, men noe testing og tuning av reguleringsystemet gjenstår siden uventede utfordringer oppsto. Dette tar vi med oss over i iterasjon 9 og fullfører i starten av denne.

Hovedmålene med denne iterasjonen er oppfylt, og vi går nå over i iterasjon 9 hvor vi skal ferdigstille og fullføre prosjektet!

Referanser

[1] Prosjektplan - prosjektplan_4.0.pdf

[2] Veikart fra ProsjektWeb - veikart_1.0.pdf



HØGSKOLEN
i Buskerud

Oppgavetittel: Hovedprosjekt - Iterasjonsrapport iterasjon 9 - v2.0 <i>Periode: 2011-05-17- 2010-06-09</i>		
Fag(nr./navn): SFHO 3200 / Prosjektstyring med hovedprosjekt		
Gruppemedlemmer, signatur:		
_____ Vegard Torkelsen	_____ Michael Odden	_____ Herå Rørvik
_____ Trine Lindberg	_____ Luke Carambot	_____ Patrick Bjørum
Intern veileder: Dag Samuelson Intern sensor: Olaf Hallan Graven Ekstern veileder og sensor: Ragnvald Otterlei		
Dato: 2011-05-24		

Innholdsfortegnelse

Dokumenthistorie.....	1
Innledning.....	1
1 Aktiviteter fra iterasjonsplanen.....	2
2 Overholdelse.....	2
3 Utfordringer og tiltak.....	3
Konklusjon.....	3
Referanser.....	4

Dokumenthistorie

Versjon	Beskrivelse	Når
1.0	Første offisielle versjon	2011-05-23
2.0	Noen omformuleringer	2011-05-24

Tabell 1: Dokumenthistorie

Innledning

En iterasjonsrapport oppsummerer en avsluttet iterasjon og skal analysere prosessen dit. I denne rapporten har vi lagt med aktivitets-oversikten slik den fremkom i iterasjonsplanen, samt at vi tar for oss problemer vi har hatt, hvor godt vi har overholdt planen og eventuelle justeringer videre.

Målet med iterasjon 7 var å implementere det forbedrede designet fra iterasjon 6. På slutten av denne iterasjonen skulle det være mulig å fly det regulerte fartøyet!

Viktigste resultater av denne iterasjonen:

- Ferdigstilt, enkelt flygbart fartøy
- Ferdigstilt dokumentasjon
- Plakat
- Presentasjon 3

1 Aktiviteter fra iterasjonsplanen

Aktivitet	Ansvarlig	Frist	Ressurser	Artifakter/resultater	ID
Presentasjon 3	Trine	2011-06-09	Alle på gruppa, PC, prosjektor	Presentasjon 3	111-3
Prosjektplakat	Michael	2011-06-06	Michael, Patrick, PC	Prosjektplakat	109
Etteranalyse	Luke	2011-05-27	Luke, PC	Etteranalysedokument	303
Ferdigstilling til innlevering	Vegard	2011-05-27	Vegard, fartøyet, PC	Ferdigstilt fartøy og prosjekt	503
Tuning og testing av reguleringsystem	Luke	2011-05-24	Gruppemedlemmene, PC, fartøyet	Ferdig tunet reguleringsystem	503-1
Ferdigstilling av dokumenter	Herå	2011-05-27	Gruppemedlemmene, PC	Ferdige dokumenter	503-2
Testing av kameramodul	Patrick	2011-05-06	Patrick, Vegard, PC, kameramodul	Fungerende kamera med live overføring	505
Brukerveiledning	Vegard	2011-05-27	Vegard, Patrick, PC	Brukermanual-dokument	606

Tabell 2: Nærmere detaljer rundt aktivitetene finnes i prosjektplanen[1] samt på prosjektweb[2]

2 Overholdelse

Denne rapporten skrives halvveis ut i iterasjonen, siden frist for levering av dokumentasjon er en stund før iterasjonsslutt. Men så langt i denne iterasjonen ligger vi godt an etter planen, og mener at siste del av iterasjonen også vil gå etter planen dersom ikke store uventede problemer skulle dukke opp.

Vi har i denne iterasjonen produsert følgende artifakter:

- Oppdatert simuleringsdokument
- Etteranalysedokument (under utvikling)
- Brukermanual (under utvikling)
- Promo plakat
- Prosjektplakat (under utvikling)
- Presentasjon 3 (under utvikling)

Andre vesentlige resultater/produkter:

- Funnet PID parametre til gyro modus som fungerer
- Kameramodul fungerer
- Godt flygbart fartøy med live kamera overføring!

3 utfordringer og tiltak

Å finne riktige PID parametre har vært en utfordring, og testing av parametre har tatt mye tid. Vi har funnet gode parametre i Gyro-modus, men en del tuning gjenstår i stabil modus for å få fartøyet så stabilt som vi ønsker det. Vi har noen dager igjen, og ledige ressurser vil bruke tid på tuning, men i første omgang prioriteres nå ferdigstilling av dokumenter til innlevering, samt forberedelser til siste presentasjon. Tiden er knapp, og vi må fordele ressurser godt og gjøre prioriteringer.

Konklusjon

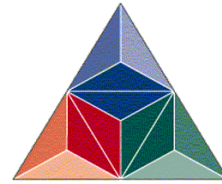
Vi har et godt fungerende flygbart fartøy, men noe tuning av reguleringssystemet gjenstår, og tiden er knapp. Vi gjør så godt vi kan for å tune fartøyet ved ledig tid, men prioriterer innlevering av dokumenter og siste presentasjon.

Hovedmålene med denne iterasjonen er delvis oppfylt, og vi regner med å oppfylle resten innen iterasjonsslutt, og ferdigstille og fullføre prosjektet!

Referanser

[1] Prosjektplan - prosjektplan_4.0.pdf

[2] Veikart fra ProsjektWeb - veikart_1.0.pdf



HØGSKOLEN
i Buskerud

Oppgavetittel: Hovedprosjekt - Veikart - v1.0		
Fag(nr./navn): SFHO 3200 / Prosjektstyring med hovedprosjekt		
Gruppemedlemmer, signatur:		
_____ Vegard Torkelsen	_____ Michael Odden	_____ Herå Rørvik
_____ Trine Lindberg	_____ Luke Carambot	_____ Patrick Bjørum
Intern veileder: Dag Samuelsen Intern sensor: Olaf Hallan Graven Ekstern veileder og sensor: Ragnvald Otterlei		
Dato: 2011-05-30		

Dokumenthistorie

Versjon	Beskrivelse	Når
1.0	Første offisielle versjon	2011-05-30

Tabell 1: Dokumenthistorie

Innledning

Dette dokumentet er kun en eksport av veikartet slik det fremstår i prosjektsiden og er kun ment for å ha et statisk dokument å referere til fra andre dokumenter.

Det gjøres obs på at aktivitets-IDene som referert til i oversikten er de interne IDene som benyttet i prosjekt-sidene og ikke de angitt i prosjektplanen. Se [1] for oversikt.

1 Iterasjon 1 - MP 1

Forprosjekt, analyse og første presentasjon

- Aktivitet #5: Kravspesifikasjon
- Aktivitet #12: Redmine-veiledning
- Aktivitet #14: Forstudierapport.
- Aktivitet #17: Idèdokument
- Aktivitet #20: Prosjektplan
- Aktivitet #21: Oppretting av maler, standardisering av dokumenter eller kode
- Aktivitet #23: Hjemmeside
- Aktivitet #28: Prosjektplan, første versjon
- Aktivitet #30: Presentasjon 1
- Aktivitet #35: Testspesifikasjon
- Aktivitet #37: Prosjektbeskrivelse/idèdokument
- Aktivitet #46: Forstudie
- Aktivitet #52: Bestilling av komponenter
- Aktivitet #75: Budsjett

2 Iterasjon 2

Design av uregulert fartøy

- Aktivitet #56: Prototype til testing
- Aktivitet #60: Testplan
- Aktivitet #62: Testbenk, design
- Aktivitet #64: Design av programvare for uregulert fartøy
- Aktivitet #69: Iterasjonsrapport 2
- Aktivitet #77: Design av koblingskjema

3 Iterasjon 3 - MP2

Implementasjon av uregulert fartøy.

- Aktivitet #57: Testbenk
- Aktivitet #66: Utvikling av programvare for uregulert fartøy
- Aktivitet #70: Iterasjonsrapport 3
- Aktivitet #87: Testing av prototype
- Aktivitet #91: Prototype
- Aktivitet #100: Regnskap
- Aktivitet #109: Programvare for testbenk

4 Iterasjon 4

Første design av regulert fartøy

- Aktivitet #65: Design av programvare for regulert fartøy
- Aktivitet #71: Iterasjonsrapport 4
- Aktivitet #80: Design av matematisk modell
- Aktivitet #81: Design av signalbehandling
- Aktivitet #92: Design av reguleringsystem
- Aktivitet #104: Iterasjonsplan iterasjon 4
- Aktivitet #111: Utbedre signalproblemer
- Aktivitet #112: Designe testbenk-utvidelse for regulert fartøy
- Aktivitet #113: Utbedre testbenk for uregulert fartøy
- Aktivitet #114: Design av nye tester
- Aktivitet #115: Ombygging for lavere COG

5 Iterasjon 5

Implementasjon av regulert fartøy

- Aktivitet #67: Utvikling av programvare for regulert fartøy
- Aktivitet #72: Iterasjonsrapport 5
- Aktivitet #85: Implementering av signalbehandling
- Aktivitet #86: Implementering av reguleringsystem
- Aktivitet #105: Iterasjonsplan iterasjon 5

6 Iterasjon 6

Tilpasset design av regulert fartøy

- Aktivitet #73: Iterasjonsrapport 6
- Aktivitet #83: Matematisk modell
- Aktivitet #106: Iterasjonsplan iterasjon 6
- Aktivitet #107: Iterasjonsplan iterasjon 7
- Aktivitet #110: Presentasjon 2
- Aktivitet #122: Sensor filtrering
- Aktivitet #124: Ferdigstilling av dokumenter

7 Iterasjon 7

Forbedret implementasjon av uregulert fartøy

- Aktivitet #74: Iterasjonsrapport 7
- Aktivitet #79: Feilsøking og forbedring
- Aktivitet #84: Simulering av reguleringsystem
- Aktivitet #88: Testing av reguleringsystemet
- Aktivitet #121: Unit-tester
- Aktivitet #123: Redesigne håndtering av radio-input
- Aktivitet #125: Implementering av reguleringsystem
- Aktivitet #127: Parameterfremstilling for matematisk modell

- Aktivitet #128: Implementering av testbenk-protokoll i regulert fartøy
- Aktivitet #129: Forbedring av matematisk modell
- Aktivitet #130: Forbedringer av programvare ifm ferdigstilling
- Aktivitet #131: Lagring av tuning-data mellom omstarter
- Aktivitet #132: Forbedring av testbenk
- Aktivitet #134: Iterasjonsplan iterasjon 9

8 Iterasjon 8

Design og implementasjon av C-krav

- Aktivitet #89: Testing av nødlandingsfunksjon
- Aktivitet #90: Testing av batterinivå
- Aktivitet #95: Design og utvikling av ekstrarfunksjoner

9 Iterasjon 9

Milepæl 4 - ferdigstilling av prosjektet.

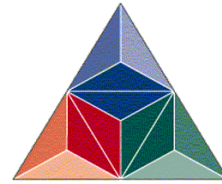
- Aktivitet #32: Presentasjon 3 (Siste presentasjon!)
- Aktivitet #39: Prosjektplakat
- Aktivitet #47: Etteranalyse
- Aktivitet #58: Ferdigstilling til innlevering
- Aktivitet #135: Tuning og testing av reguleringsystem
- Aktivitet #136: Ferdigstilling av dokumenter
- Aktivitet #137: Testing av kameramodul
- Aktivitet #138: Brukermanual
- Aktivitet #139: Implementasjonsdokument, programvare
- Aktivitet #140: Iterasjonsrapport 9

Konklusjon

Dette dokumentet er kun ment som et referansedokument.

Referanser

[1] ID-Oversikt - id-oversikt.pdf



HØGSKOLEN
i Buskerud

Oppgavetittel: Hovedprosjekt - ID-oversikt - v1.0		
Fag(nr./navn): SFHO 3200 / Prosjektstyring med hovedprosjekt		
Gruppemedlemmer, signatur:		
_____ Vegard Torkelsen	_____ Michael Odden	_____ Herå Rørvik
_____ Trine Lindberg	_____ Luke Carambot	_____ Patrick Bjørum
Intern veileder: Dag Samuelsen Intern sensor: Olaf Hallan Graven Ekstern veileder og sensor: Ragnvald Otterlei		
Dato: 2011-05-30		

Dokumenthistorie

Versjon	Beskrivelse	Når
1.0	Første offisielle versjon	2011-05-30

Tabell 1: Dokumenthistorie

Innledning

Dette dokumentet er kun ment som en koblingsoversikt mellom aktivitets-IDer slik de er definert i prosjektplanen kontra hvordan de er representert internt i ProsjektWeb'ens database.

1 Oversikt

AktivitetsID	ProsjektWeb ID	Emne
100	24	Administrasjon og prosjektstyring
101	15	Prosjektorganisering
102	16	Møter
103	37	Prosjektbeskrivelse/idèdokument
104	20	Prosjektplan
105	21	Oppretting av maler, standardisering av dokumenter eller kode
106	22	Økonomi
107	23	Hjemmeside
108	19	Timelister
109	39	Prosjektplakat
110	8	Interne møter
111	29	Presentasjoner
112	40	Oppfølgingsdokumenter
113	101	Iterasjonsplaner
200	25	Research
201	41	Signalbehandling
202	42	Reguleringssystem, research
203	3	Undersøkelse rundt matematisk modell
204	44	Multikopterteori
205	45	Komponentdokumentasjon
300	26	Krav/Analyse

301	46	Forstudie
302	5	Kravspesifikasjon
303	47	Etteranalyse
400	27	Design/Teknologi
401	48	Design av programvare
402	49	Designdokument
403	50	Teknologidokument
404	51	Programmering
405	52	Bestilling av komponenter
406	53	Reguleringssystem
407	77	Design av koblingsskjema
408	80	Design av matematisk modell
409	81	Design av signalbehandling
410	62	Testbenk, design
411	95	Design og utvikling av ekstrarfunksjoner
500	54	Implementasjon
501	56	Prototype til testing
502	57	Testbenk
503	58	Ferdigstilling til innlevering
504	85	Implementering av signalbehandling
505	86	Implementering av reguleringssystem
600	55	Test
601	59	Testing
602	35	Testspesifikasjon

603	60	Testplan
604	61	Testrapport
606	138	Brukermanual
606	63	Brukerveiledning
700	82	Simulering
701	83	Matematisk modell
702	84	Simulering av reguleringsystem
103-1	17	Idèdokument
104-1	28	Prosjektplan, første versjon
104-2	96	Oppdatere prosjektplanen
106-1	75	Budsjett
106-2	100	Regnskap
107-1	116	Vedlikehold og oppdateringer av ProsjektWeb
111-1	30	Presentasjon 1
111-2	110	Presentasjon 2
111-3	32	Presentasjon 3 (Siste presentasjon!)
113-2	102	Iterasjonsplan iterasjon 2
113-3	103	Iterasjonsplan iterasjon 3
113-4	104	Iterasjonsplan iterasjon 4
113-5	105	Iterasjonsplan iterasjon 5
113-6	106	Iterasjonsplan iterasjon 6
113-7	107	Iterasjonsplan iterasjon 7
113-8	134	Iterasjonsplan iterasjon 9
114-2	69	Iterasjonsrapport 2

114-3	70	Iterasjonsrapport 3
114-4	71	Iterasjonsrapport 4
114-5	72	Iterasjonsrapport 5
114-6	73	Iterasjonsrapport 6
114-7	74	Iterasjonsrapport 7
114-8	140	Iterasjonsrapport 9
301-1	14	Forstudierapport.
302-1	119	Designokument reguleringsystem
302-1	117	Oppdatere kravspesifikasjon
401-1	64	Design av programvare for uregulert fartøy
401-2	65	Design av programvare for regulert fartøy
404-1	66	Utvikling av programvare for uregulert fartøy
404-2	67	Utvikling av programvare for regulert fartøy
404-3	111	Utbedre signalproblemer
406-1	92	Design av reguleringsystem
406-2	79	Feilsøking og forbedring
409-1	122	Sensor filtrering
410-1	112	Design testbenk-utvidelse for regulert fartøy
501-1	91	Prototype
501-2	115	Ombygging for lavere COG
502-1	109	Programvare for testbenk
502-2	113	Utbedre testbenk for uregulert fartøy
503-1	135	Tuning og testing av reguleringsystem
503-2	136	Ferdigstilling av dokumenter

505-1	125	Implementering av reguleringsystem
601-1	87	Testing av prototype
601-2	88	Testing av reguleringsystemet
601-3	89	Testing av nødlandingsfunksjon
601-4	90	Testing av batterinivå
601-5	94	Oppsett av tester
601-6	93	Testdesign
601-7	114	Design av nye tester
601-7	68	Testlogg
602-1	118	Oppdatere testspesifikasjon
701-1	129	Forbedring av matematisk modell
412	139	Implementasjonsdokument, programvare
607	137	Testing av kameramodul
505-1	133	Påbygging av reguleringsystemet
502-1	132	Forbedring av testbenk
505-2	131	Lagring av tuning-data mellom omstarter
505-3	130	Forbedringer av programvare ifm ferdigstilling
505-4	128	Implementering av testbenk-protokoll i regulert fartøy
408-1	127	Parameterfremstilling for matematisk modell
505-6	126	Implementere redesignet radio-input
503-3	124	Ferdigstilling av dokumenter
413	123	Redesigne håndtering av radio-input
111-1	120	Vi deler informasjon

Konklusjon

Dette dokumentet er kun ment som et referansedokument.