CHALMERS

Reglering av ett UAV-system

En initial prototyp för ett styrsystem till en Quadcopter

Examensarbete inom Högskoleingenjörsprogrammet i Elektroteknik

Mikael Bengtsson
Niclas Carlström

Institutionen för data- och informationsteknik
CHALMERS TEKNISKA HÖGSKOLA
Göteborg, Sverige 2014

Reglering av ett UAV-system
En initial prototyp för ett styrsystem till en Quadcopter

Mikael Bengtsson, Niclas Carlström

Institutionen för data- och informationsteknik
Chalmers tekniska högskola
412 96 Göteborg
Tel: 031-772 1000
Fax: 031-772 3663

Institutionen för data- och informationsteknik
Göteborg, 2014

Sammanfattning

Denna rapport behandlar utvecklandet och konstruktionen av ett digitalt styrsystem till en autonom UAV(Unmanned Aerial Vehicle)-farkost. Detta i form av en quadcopter, en typ av helikopter med fyra propellrar som styrs individuellt utifrån önskat flygmönster. Idén till projektet väcktes av ett intresse för komplexiteten kring quadcopterns styrsystem samt de utmaningar ett reglersystem av denna typ för med sig. Målet med arbetet är att utifrån en specifik hårdvara undersöka om stabil flygning i rummet kan uppnås. Det primära målet är att utveckla ett styrsystem som balanserar quadcoptern vinkelrät mot z-axeln i rummet då detta är ett avgörande första steg för vidare utveckling av styrningen. En omfattande del i arbetet behandlar modellering av systemet och att utifrån detta konstruera det reglersystem som ska uppnå målet.

I rapporten redovisas olika metoder för hur teoretiska modeller utvecklas med hänsyn till farkostens delsystem. En serie experiment utfördes på den fysiska modellen för att samla in mätdata som användes till modelleringen. I metoden utfördes även simuleringar på modellerna för att testa funktioner på ett kontrollerat sätt. Följaktligen testades funktionerna på hårdvaran och resultaten jämfördes. En omfattande del i metoden är utvecklingen av mjukvaran som hela systemet är beroende av. Arbetet omfattar även framtagandet av den hårdvara som krävs för att realisera hela konstruktionen.

Projektet resulterade i en initial prototyp för ett kontrollsystem till höjd- och vinkelkontroll med tillhörande teoretiska modeller för quadcopterns dynamik. För höjdkontrollen behandlades detta strikt teoretiskt och i simuleringsmiljö medan vinkelkontrollen även applicerades på systemet och utvärderades. Samtidigt utvecklades den hård- och mjukvara som var nödvändig för att realisera konstruktionen. Vidare utvecklades även de simuleringsmiljöer som legat till grund för dimensioneringen av styrsystemet.

Abstract

This report discusses the development and construction of a digital control system for an autonomous UAV (Unmanned Aerial Vehicle). The vehicle in this project is a quadcopter which is a type of helicopter with four rotors that are controlled individually according to the desired flight pattern. The idea for the project was evoked by an interest in the complexity of this type of control system, and what challenges might arise in the developing process. The goal of the project is to investigate whether a stable flight, with specific components, can be achieved. The primary objective is to develop a control system that balances the quadcopter perpendicular to the z-axis of the room. This is a crucial first step for further development of the control system. A major part of the report deals with modeling of the quadcopter and its subsystems. Consequently, simulation of these models constitutes a considerable part of the report and its results.

The report presents different methods of how theoretical models are developed, with reference to the quadopter's subsystems. A series of experiments was conducted on the physical model in order to collect the measurement data used for modeling. The method also included simulations on models for testing different functions in a controlled environment. Finally, the functions were implemented on the hardware and the results were evaluated and compared to those of the simulation. Another substantial part of the method is the development of software which constitutes the control system. Furthermore, the report also describes the construction of the hardware required to realize the entire system.

The project resulted in an initial prototype of a control system for height and angle control with related theoretical models for the system. For height control, this was treated theoretically and solely in a simulation environment. The angle control was treated in the same manner, except this was also applied to the hardware and evaluated. In addition, the project resulted in developed hardware and software environments for the design. The results also contain the simulation environments that formed the basis for the design of the control system.

Beteckningar

Yaw – Rotation kring z-axel
Pitch – Rotation kring y-axel
Roll – Rotation kring x-axel

ESC – Electronic Speed Controller
ISR – Interrupt Service Routine
SMD – Surface Mounted Device
CAD – Computer-Aided Design
UAV – Unmanned Aerial Vehicle
BLDC - Brushless DC

Innehållsförteckning

1. Inledning

1.1 Bakgrund

Elektriska styrsystem utgör idag en stor del i dagens tekniksamhälle. Allt från styrsystemet i ett fartyg till styrning inom industriell produktion är uppbyggda kring dessa system. Ett avancerat styrsystem kan exempelvis avsevärt öka precisionen vid navigering eller öka noggrannheten och snabbheten i en tillverkningsprocess. Generellt minskar också omkostnaderna om ett effektivt och tillförlitligt styrsystem utför arbetsmoment som en operatör annars hade haft.

Historiskt sett har de dyrare och mer avancerade styrsystemen främst nyttjats av företag med en omfattande omsättning inom industribranschen. Detta kan förklaras med att endast dessa företag har haft kapital och intresse i att investera i dessa. Men i takt med att den underliggande tekniken bakom avancerade styrsystem blir mer vanlig och att priset på snabba mikroprocessorer sjunker blir också utvecklings- och tillverkningskostnaderna mindre för företag. Detta gör att elektriska system med komplicerade styralgoritmer ofta dyker upp i allt mer vardagliga produkter så som leksaker, hemelektronik m.m. Att yngre generationer vuxit upp med en ständig närvaro av avancerad teknik är också ett argument för att det finns, och kommer finnas en lönsam marknad för företag att tillverka tekniskt avancerade produkter även för kommersiella marknader.

1.2 Syfte

Syftet är att undersöka möjliga tillvägagångssätt vid utveckling av styrsystemet till en autonom UAV (Unmanned Aerial Vehicle)-farkost i form av en quadcopter. Vidare att analysera om målet är genomförbart med de tänkta metoder som presenteras i rapporten. Den tänkta metoden är att utveckla ett styrsystem bestående av, till största del, separata och för ändamålet outvecklade komponenter. Detta medför att en betydande del av tiden för utvecklingen av hela systemet ägnas till konstruktion och konfiguration av enskilda komponenter.

Processorn som används i projektet är en 32-bitars ARM processor. Dessa används idag i stor utsträckning i flertalet applikationer inom modern teknik, ofta inom konsumentelektroniken på grund av dess strömsnåla egenskaper och relativt låga pris. År 2013 utgjorde ARM processorer ca 90 % av hela marknaden för processorerna i smartphones och även en betydande del i marknaden för notebooks [1]. En ökad kunskap inom arkitekturen och syntax för denna typ av processor ger därför en hög relevans till uppgiften och nyttiga erfarenheter i framtida arbetslivet.

1.3 Mål

Projektets mål är att utveckla ett initialt styrsystem till en quadcopter samt att undersöka om ett tillfredställande resultat kan uppnås utifrån specifik hårdvara. Intentionen är att det utvecklade styrsystemet ska utgöra en prototyp och kan beskrivas som ett första steg till en mer avancerad slutprodukt. Detta på grund av begränsningar i form av tid och resurser som föreligger. Därför

har vissa avgränsningar gjorts inom projektets ram, se avsnitt 1.4 nedan. Följaktligen kommer förslag på vidare utveckling av styrsystemet att behandlas i rapportens slutsats.

Vidare är målet att med experimentella metoder utforma teoretiska modeller av systemet. Om systemets olika delar beskrivs i form av överföringsfunktioner kan en simuleringsmiljö utarbetas och önskade funktioner testas på ett kontrollerat sätt. Avsikten är att implementera simulerade funktioner på hårdvaran för att sedan utvärdera och jämföra respektive resultat. Då en tillfredställande höjdkontroll är starkt beroende av att quadcoptern är stabil och parallell mot marken ställs fokus på att utveckla en fungerande algoritm för att stabilisera farkosten vinkelrätt mot z-axeln i rummet, alltså parallellt mot marken. Mätdata från sensorer kommer att behandlas i en processor som i sin tur reglerar varvtalet på motorerna individuellt för att uppnå stabilitet. Målsättningen är att med två olika sensorer; gyroskop och accelerometer uppnå stabiliseringen. Sammanfattningsvis kan frågeställningen formuleras till att undersöka om önskad funktion kan uppnås med den tänkta metoden samt de specifika komponenter som systemet innehåller.

1.4 Avgränsningar

Det primära målet är utveckla och undersöka det elektroniska styrsystemet till quadcoptern. Med anledning av detta kommer vissa enheter som inte direkt är associerade med styrsystemet införskaffas färdigutvecklade. Anpassningskretsen till motorerna, ESC (Electronic Speed Controller), beställs färdigutvecklade då dessa är för tidskrävande att utveckla inom projektet. Ett komplett chassi med tillhörande BLDC (Brushless DC) motorer och propellrar beställs även det färdigt och kommer inte utvecklas inom projektet. En detaljerad teori bakom motorerna kommer inte heller att behandlas. För att rama in projektet har vissa avgränsningar på styrsystemet gjorts. Dock är avsikten att dessa ska vidareutvecklas utanför projektets ram. Höjdkontroll kommer inte implementeras i hårdvaran, dock kommer dynamiken samt regulator för höjdkontroll beräknas och simuleras i rapporten. Vidare kommer kontrollsystemet för vinkeln *yaw* inte behandlas i rapporten, se avsnitt 2.1.2. Dessa två kontrollprinciper ses som en vidareutveckling av det arbete som detta projekt resulterar i.

2. Teknisk beskrivning

En quadcopter består av fyra propellrar som styrs individuellt för att få farkosten att flyga stabilt och på önskad höjd. Propellrarna är monterade på varsin borstlös likströmsmotor som via en ESC styrs av mikrokontrollern. Detta med en PWM signal vars duty cycle varieras för att reglera gaspådrag till motorerna. Följande kapitel beskriver teorin bakom de mest betydande delarna i arbetet.

2.1 Modellering

I denna rapport skildras begreppet *systemet* som quadcoptern och dess dynamik. Med andra ord används begreppet för att beteckna farkosten vars egenskaper som ska undersökas. Under arbetets gång undersöks de delsystem som återför hela systemets egenskaper. Ett delsystem kan till exempel reduceras till farkostens dynamik runt en specifik axel eller till lyftkraft för en enskild motor.

En modell kan definieras som något som beskriver ett system utan att användaren behöver experimentellt utföra tester på systemet [2]. De matematiska modeller som denna rapport behandlar beskriver samband mellan storheter och anges som matematiska relationer. I litteraturen beskrivs två grundprinciper för hur modeller konstrueras, fysikaliskt modellbygge samt identifiering. I rapporten används båda principerna. Den första principen, fysikaliskt modellbygge, använder de naturlagar som beskriver delsystemen. Denna metod används i framtagandet av modell för höjddynamiken, se avsnitt 2.1.1. Den andra principen använder observationer från systemet för att sedan anpassa modellens egenskaper till systemet. Med andra ord utförs experiment på systemet och dess delsystem där modellen konstrueras utifrån mätdata, ett exempel på detta är stegsvarsanalys [2]. Identifiering är den princip som används i samband med modellen för vinkeldynamiken.

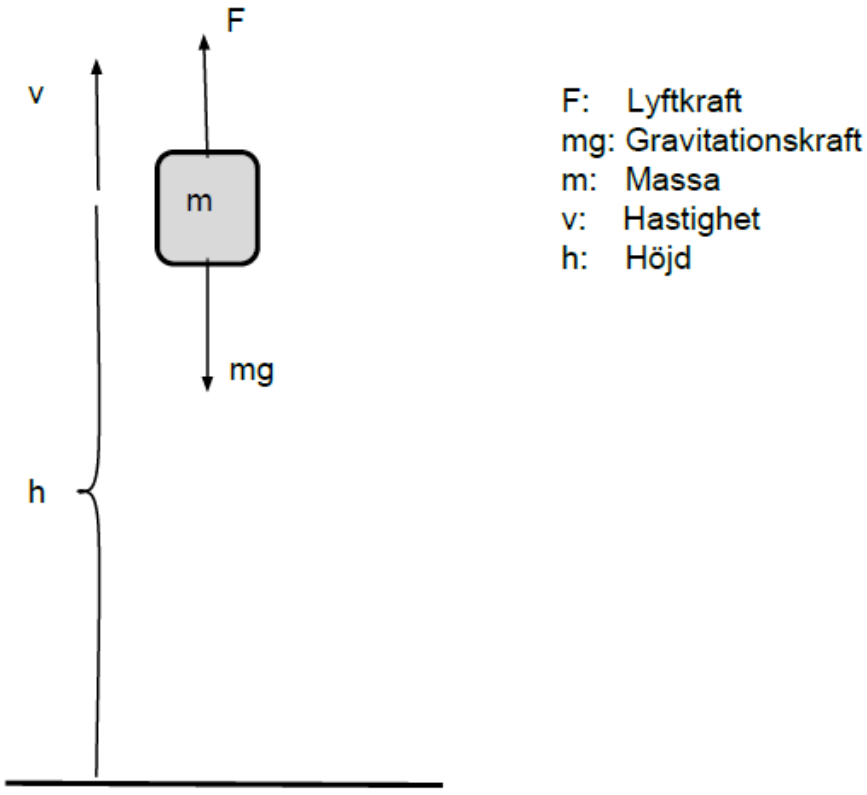
Strävan är alltså att utforma en modell som efterliknar systemet så väl som möjligt. I metodkapitlet, avsnitt 4.4, beskrivs de olika metoder som används. Där används bland annat verktyget System Identification Toolbox i Matlab för att ta fram olika modeller. En utmaning i modelleringen är de olinjäriteter som uppstår bland annat från de aerodynamiska effekter som propellrarna producerar [3]. I de ovan nämnda verktyget finns standardmodeller som kan hantera denna typ av systemdynamik. Exempel på de standardmodeller som finns att använda i verktyget är modeller som ARX och Hammerstein Wiener [4]. Då arbetet med att lösa ut de fysikaliska sambanden för de olinjära effekterna är allt för tidskrävande för detta projekt används endast linjära modeller för att beskriva systemet.

En stor fördel med att experimentellt ta fram en modell för en process är att man då kan dimensionera en regulator med önskad stigtid och undertrycka eventuella översvängningar. Kännedom om processens överföringsfunktion gör det också möjligt att simulera systemet i en

dator och finjustera parametrar i regulatorn vid behov. För att förenkla identifieringen av systemets dynamik särskiljs höjd- och vinkeldynamik. Dessa två delsystem beskrivs mer ingående i kommande avsnitt. De beskriver den teori som ligger bakom de modelleringsmetoder som används i avsnitt 4.4. Resultaten av modelleringen redovisas i avsnitt 5.2.

2.1.1 Höjddynamik

Om inte quadcoptern är precis parallell mot marken när motorerna är aktiva kommer den att driva sidledes åt det håll systemet är vinklat mot. För att överkomma detta problem kan en testrigg konstrueras som hindrar farkosten från att driva åt något håll och endast röra sig vertikalt. Då det saknas resurser och tid för att konstruera en sådan testrigg används ett strikt teoretiskt resonemang istället för en experimentell stegsvarsanalys. Figur 2.1 nedan beskriver de fysikaliska krafter som verkar på systemet.



Figur 2.1 Illustration över härledningen till överföringsfunktionen från verkande kraft till höjd.

För att quadcoptern ska öka i höjd krävs att lyftkraften från propellrarna är större än gravitationskraften som verkar på systemet. Om systemet stiger i höjd kan den resulterande kraften som påverkar höjden beskrivas enligt Newtons andra lag som ekvation (2.1). Löses accelerationen ut ur detta samband och skrivs ut enligt dess definition fås differentialekvationen (2.2).

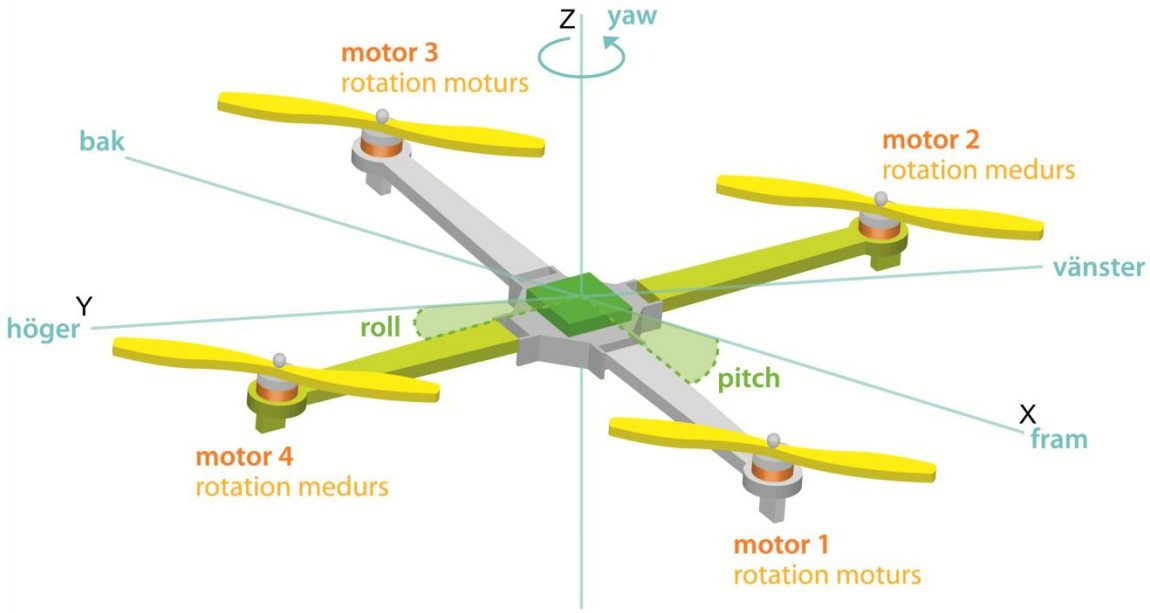
$$F - mg = m \cdot a \qquad (2.1)$$

$$a = \frac{dv}{dt} = \frac{\partial^2 h}{\partial t^2} = \frac{F-mg}{m} \qquad (2.2)$$

En Laplacetransformation av differentialekvationen (2.2) och omskrivning av uttrycket leder till överföringsfunktionen (2.3) som beskriver quadcopterns höjd.

$$h = \frac{F-mg}{ms^2} \qquad (2.3)$$

2.1.2 Vinkeldynamik



Figur 2.2 Illustration över definieringen av systemet.

Som figur 2.2 illustrerar beror systemets vinkel på lyftkraft hos motstående motorer. Lyftkraft från motor ett och tre bestämmer vinkel *pitch* och lyftkraft från motor två och fyra bestämmer vinkel *roll*. Så som systemet är definierat innebär en negativ *pitch* att systemet rör sig framåt och en negativ *roll* att systemet rör sig åt höger. På grund av att motorerna både generar en lyftkraft och ett vridmoment måste motstående motorer rotera i samma riktning och de andra två i motsatt riktning. Detta för att dessa vridmoment ska eliminera varandra och således hindra systemet från att bli instabilt och rotera okontrollerat [5]. Om lyftkraften från två motstående motorer är större än de andra två kommer inte dessa vridmoment eliminera varandra och systemet kommer att rotera kring z-axeln, denna vinkel är definierad som *yaw*. Modeller och regulatorer för vinkelkontroll tas fram via experimentella tester på vinkeln *pitch*. Det antas att systemets dynamik är lika för både *pitch* och *roll* och att dessa kan beskrivas med samma modell och därmed regleras med samma parametrar i deras separata regulatorer. Som avgränsningarna i avsnitt 1.4 nämner kommer inte reglering och modellering av vinkeln *yaw* behandlas i denna rapport.

2.2 Sensorer

2.2.1 Gyroskop

Gyroskop är fysiska sensorer som detekterar och mäter vinkelrörelse hos ett föremål i förhållande till en tröghetsreferensram [6]. Gyroskopet som användes i detta projekt mäter data i tre riktningar. Denna sensor ger en digital utsignal uttryckt i grader per sekund, alltså hur snabbt farkosten roterar kring en eller flera axlar vid tidpunkten då mätningen utförs. Om gyroskopet

endast registrerar ett positivt värde i sitt "X-register" betyder detta att farkosten endast roterar moturs kring x-axeln i ett kartetiskt koordinatsystem med en viss hastighet. Denna komponent beställdes separat och behövde ytmonteras på ett kretskort, se appendix C. Med data från gyroskopet kan vinklarna bestämmas enligt ekvation (2.4), (2.5) och (2.6) nedan. Där $Gyro_{index}\ (t_i)$ är data från gyroskopet vid sampel nummer i, t_i är tiden vid aktuellt sampel och t_s är samplingstiden.

$$pitch_{Gyro} = \int_{t_0}^{t_i} Gyro_y\ dt \approx \sum_{t_0}^{t_i} Gyro_y(t_{i-1}) \cdot t_s \qquad (2.4)$$

$$roll_{Gyro} = \int_{t_0}^{t_i} Gyro_x\ dt \approx \sum_{t_0}^{t_i} Gyro_x(t_{i-1}) \cdot t_s \qquad (2.5)$$

$$yaw_{Gyro} = \int_{t_0}^{t_i} Gyro_z\ dt \approx \sum_{t_0}^{t_i} Gyro_z(t_{i-1}) \cdot t_s \qquad (2.6)$$

2.2.2 Accelerometer

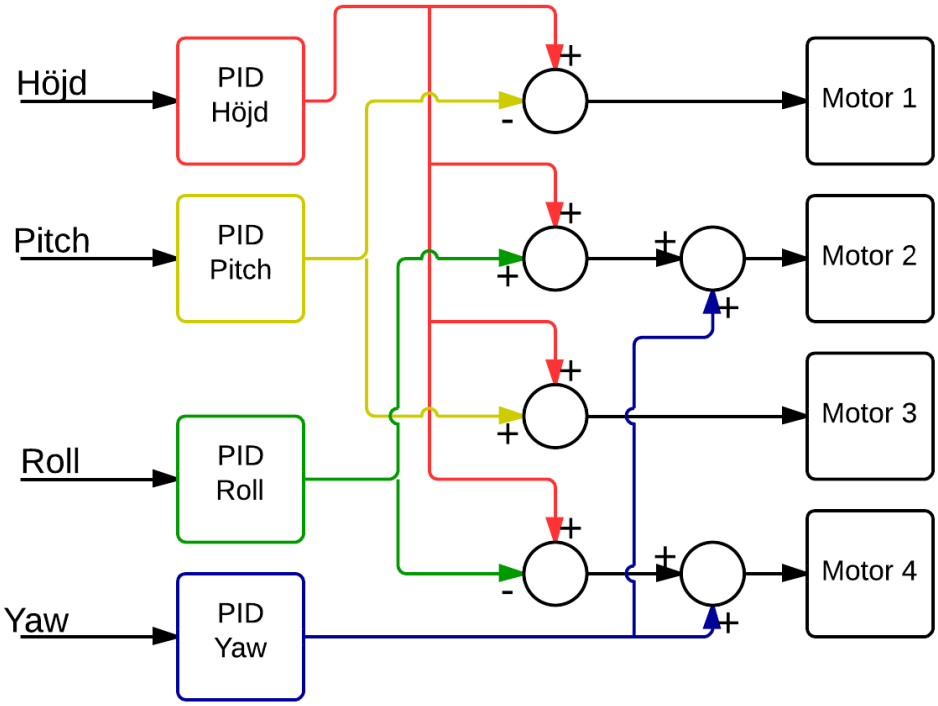
En accelerometer är en givare som ger ifrån sig en signal proportionell mot hastighetsförändringen eller accelerationen givaren utsätts för [7]. Accelerometern mäter kraft i tre riktningar och skickar ut en digital signal uttryckt i g-kraft. I ett kartetiskt koordinatsystem kan detta beskrivas som att accelerometern mäter kraft i x-, y- och z-led. Om accelerometern endast registrerar en positiv kraft i x-led betyder detta att farkosten endast förflyttar sig i negativ x-led riktning. Det faktum att gravitationen påverkar accelerometern konstant kan utnyttjas för att beräkna vinkel *pitch* och *roll* [8]. Med data från accelerometern kan vinklarna bestämmas enligt ekvation (2.7) och (2.8) nedan.

$$pitch_{Acc} = \arctan\left(\frac{\mathrm{Acc_x}}{\sqrt{\mathrm{Acc_y^2}+\mathrm{Acc_z^2}}}\right) \qquad (2.7)$$

$$roll_{Acc} = arctan\left(\frac{Acc_y}{\sqrt{Acc_x^2+Acc_z^2}}\right) \qquad (2.8)$$

2.3 Regulatorer

I kontrollsystemet används ett flertal regulatorer beroende på vilket delsystem som regleras. Figur 2.3 nedan beskriver övergripande hur ett slutgiltigt kontrollsystem kan se ut. Som figuren illustrerar producerar höjdkontrollen en styrsignal till alla motorer och kan ses som ett basvärde för den totala styrsignalen till respektive motor. De återstående styrsignalerna från *pitch, roll* och *yaw* kan ses som ändringar i basvärdet beroende på önskat flygmönster.



Figur 2.3 Beskrivning av samverkan av systemets olika regulatorer.

Som tidigare nämnts antas vinkel *pitch* samt *roll* inneha identiska dynamiska egenskaper, vilket även medför identiska parametrar för respektive regulator. Därav konstrueras och testas endast regulatorn för vinkeln *pitch* gällande vinkelkontrollen. Regulatorn för höjdreglering kommer därtill att konstrueras och simuleras, denna kommer dock inte implementeras på hårdvaran som tidigare nämnts.

I avsnitt 4.5.2 presenteras två olika metoder för att dimensionera regulatorn för vinkeln *pitch*, en modellbaserad dimensionering och en tumregelmetod. Dessa benämns som *Regulator 1* respektive *Regulator 2.* Den modellbaserade dimensioneringen bygger på principen att ta fram en total överföringsfunktion över hela systemet med önskad stigtid samt ett monotont insvängningsförlopp utan översväng. Därefter, utifrån den totala överföringsfunktionen, analytiskt bestämma en överföringsfunktion för regulatorn [9]. Den analytiskt bestämda överföringsfunktionen för regulatorn omvandlas sedan till det tidsdiskreta Z-planet och skrivs om som en differensekvation för att kunna programmeras i processorn. Regulatorns

överföringsfunktion, G_r, kan beräknas enligt ekvation (2.9) där G_p är processen och G_{tot} är systemets totala överföringsfunktion i ett återkopplat system [9].

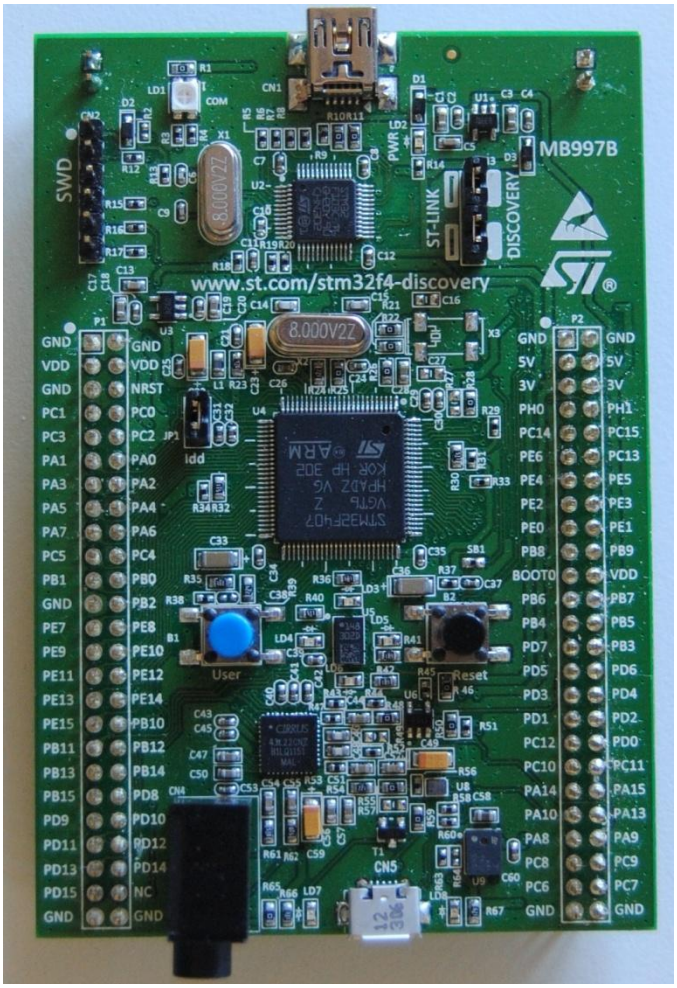
$$G_r = \frac{G_{tot}}{G_p(1-G_{tot})} \qquad (2.9)$$

Tumregelmetoden som användes är Ziegler-Nichols svängningsmetod och utfördes enligt arbetsmetodik presenterad i litteraturen [9]. Metoden bygger på att sätta systemet som ska regleras i självsvängning genom att öka den proportionella förstärkningen för en P-regulator i systemet. Med uppmätta värden på självsvängningarnas periodtid samt värdet på den proportionella förstärkningen beräknas *P*, *I* och *D* parametrarna i PID-regulatorn enligt metoden [9].

3. Systembeskrivning

Följande kapitel behandlar tekniska specifikationer för de mest signifikanta komponenter i hårdvaran som används i systemet.

3.1 STM32F4 Discovery Board



Figur 3.1 Utvecklingskortet STM32F4-Discovery.

STM32F4 Discovery Board är ett utvecklingskort tillverkat av STMicroelectronics. Kortet är baserat kring mikrokontrollern STM32F407VGT6 och innefattar även flera användbara enheter så som en digital accelerometer, ST MEMS audio sensor, integrerade tryckknappar m.m. [10]. Mikrokontrollern består av en 32-bitars ARM Cortex-M4F processor med en maximal klockfrekvens på 168MHz. Processorn programmeras i programspråket C.

Utvecklingskortet accepterar sin spänningsmatning via en extern källa eller via kortets USB-port. Vid användning av en extern spänningskälla kräver utvecklingskortet en spänning på 5 V. Kortet har även 3 V samt 5 V anslutningar för matning till externa applikationer [10]. Utvecklingskortet valdes för dess höga prestanda och låga kostnad samt att det finns öppen programvara till processorn för programmering. Vidare värderades det faktum att den typen av processor i kortet utgör en betydande del av marknaden för konsumentelektronik [1].

3.2 Sensorer

För att uppnå målet med automatiserad stabilisering i rummet användes diverse sensorer till styrsystemet. Det som initialt används i projektet är ett gyroskop och en accelerometer för den vinkelkontroll som ska realiseras. Utvecklingskortet har en integrerad accelerometer som användes i projektet. Gyroskopet implementeras på ett separat kretskort. Denna komponent beställdes enskilt som en SMD (Surface Mounted Device) krets. För att praktiskt kunna använda komponenten med resten av systemet tillverkades ett mönsterkort där komponenten ytmonterades, metoden för detta redovisas i appendix C. Detta moment bidrar till praktiska erfarenheter kring tillverkning av mönsterkort samt tillför mer elektronikrelaterade uppgifter till projektet. Nedan följer specifikationer för accelerometern samt gyroskopet.

Gyroskop L3G4200D
Tillverkare: STMicroelectronics
Specifikationer [11]:

- 2.4 V - 3.6 V driftspänning
- 1.8 V kompatibla I/O portar
- Tre valbara skalor (250/500/2000 dps)
- Stödjer I2C/SPI protokoll
- 16 bitars utdata register
- Integrerat låg- och högpassfilter med valbar bandbredd

Accelerometer LIS302DL
Tillverkare: STMicroelectronics
Specifikationer [12]:

- 2.16 V - 3.6 V driftspänning
- 1.8 V kompatibla I/O portar
- Två valbara skalor (2/8 g)
- Stödjer I2C/SPI protokoll
- 8 bitars utdata register
- Integrerat högpassfilter

Det ska förtydligas att skalan 2/8 g avser enheten g-kraft.

3.3 Motorer och ESC

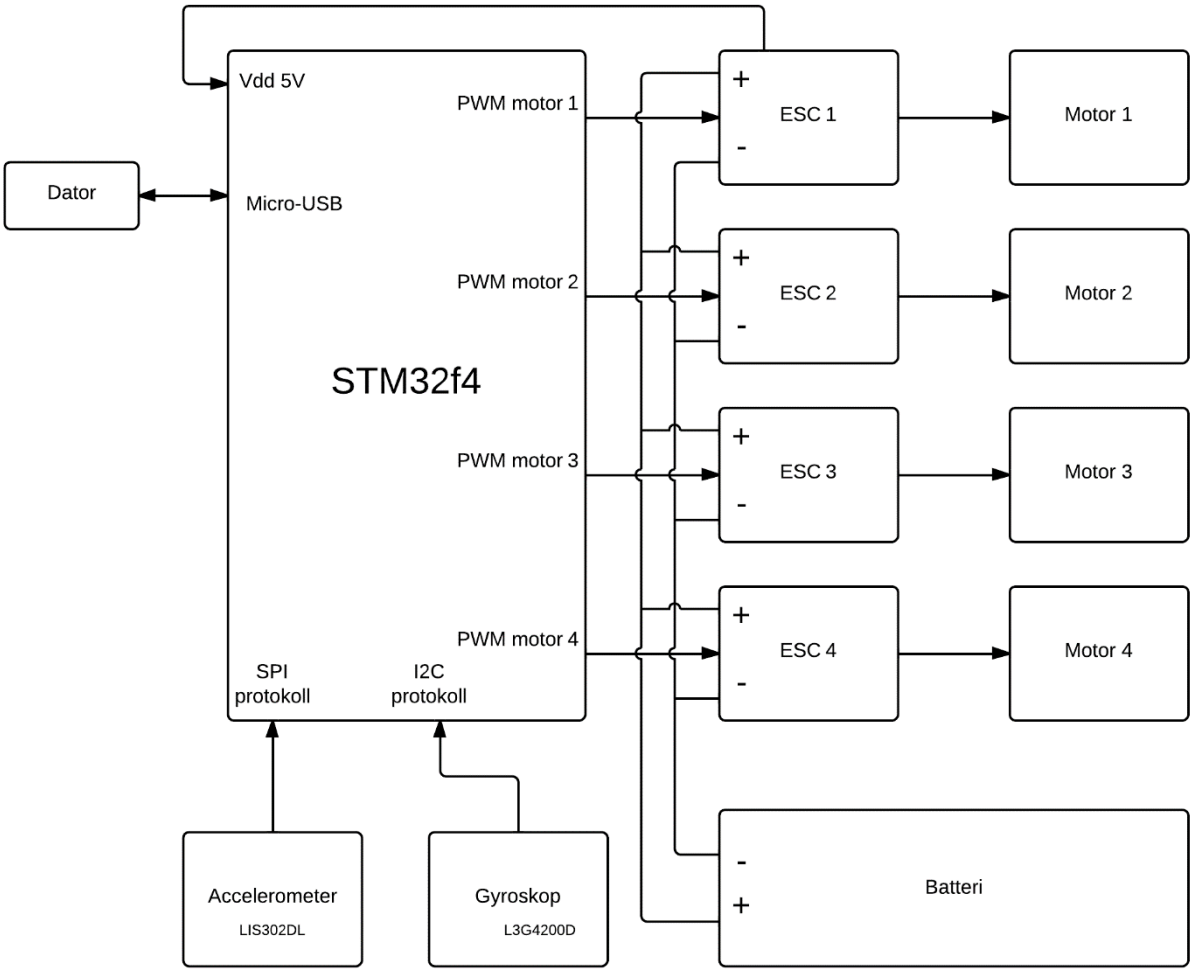
Efterforskning visade att ESC:erna har en tendens att inte klara lika stor strömstyrka som tillverkarna hävdar. Av denna anledning beställdes ESC:er som med god marginal skulle klara av den maximala strömstyrkan till motorerna. Systemet matas med ett 3-celligt LiPo-batteri. I slutet av detta avsnitt presenteras även ett principiellt kopplingsschema av systemet i figur 3.1.

ESC HK-SS30A
Tillverkare: Hobbyking
Specifikationer [13]:

- Vikt: 22 g
- Storlek: 24x52x6 mm
- Max strömstyrka 25 A
- Motortyp: Borstlösa likströmsmotorer

ST2210 Brushless Motor
Specifikationer [14]:

- Vikt: 80 g
- Diameter: 28 mm
- Höjd: 25 mm
- KV: 1050 rpm/V
- Strömkonsumtion olastad: 0.5 A
- Max strömstyrka: 10.2 A



Figur 3.1 Principiellt kopplingsschema över systemet.

4. Metod

Detta kapitel behandlar de metoder som använts under arbetet för att realisera en tillförlitlig vinkelreglering samt höjdkontroll av systemet. Som tidigare nämnts begränsas höjdkontrollen till simuleringar och testas inte på hårdvaran. Kapitlet behandlar konstruktionen av hårdvara, mjukvara samt de teoretiska metoderna för att konstruera modeller och regulatorer. Kapitlet avslutas med en redogörelse av de simuleringar och tester som utförts på framtagna modeller, regulatorer och utvecklad hårdvara. Detta ligger till grund för de resultat som redovisas i kapitel 5. Med tanke på det primära målet, att stabilisera quadcoptern vinkelrätt mot z-axeln i rummet, var en stor del av tester och simuleringar på systemet fokuserat kring vinkelkontroll. Mjukvaran Matlab Simulink används genomgående i projektet för simuleringar samt för kommunikationen mellan utvecklingskortet och dator.

4.1 Hårdvaruutveckling

Gyroskopet i systemet monteras separat eftersom utvecklingskortet saknar denna typ av sensor. Komponenten beställdes som en SMD krets vilket medför att storleken på de fysiska in- och utgångarna är mycket liten. För att praktiskt kunna använda komponenten tillsammans med det övriga systemet producerades ett enkelsidigt mönsterkort med storleksmässigt mer anpassade in- och utgångar. Gyroskopet ytmonterades därefter på det tillverkade mönsterkortet. Ytterligare komponenter kring gyroskopet för filtrering m.m. monterades på samma mönsterkort. För designarbetet av mönsterkortet användes programvaran Cadsoft Eagle Light Edition. Den fysiska tillverkningen av mönsterkortet gjordes på Elektrosektionens Teletekniska Avdelning på Chalmers Tekniska Högskola. I appendix C följer en detaljerad beskrivning kring konstruktionen av mönsterkortet samt montering av komponenter. Beskrivningen tar till största vikt upp designarbetet i mjukvaran Eagle och mindre på den fysiska framställningen av mönsterkortet.

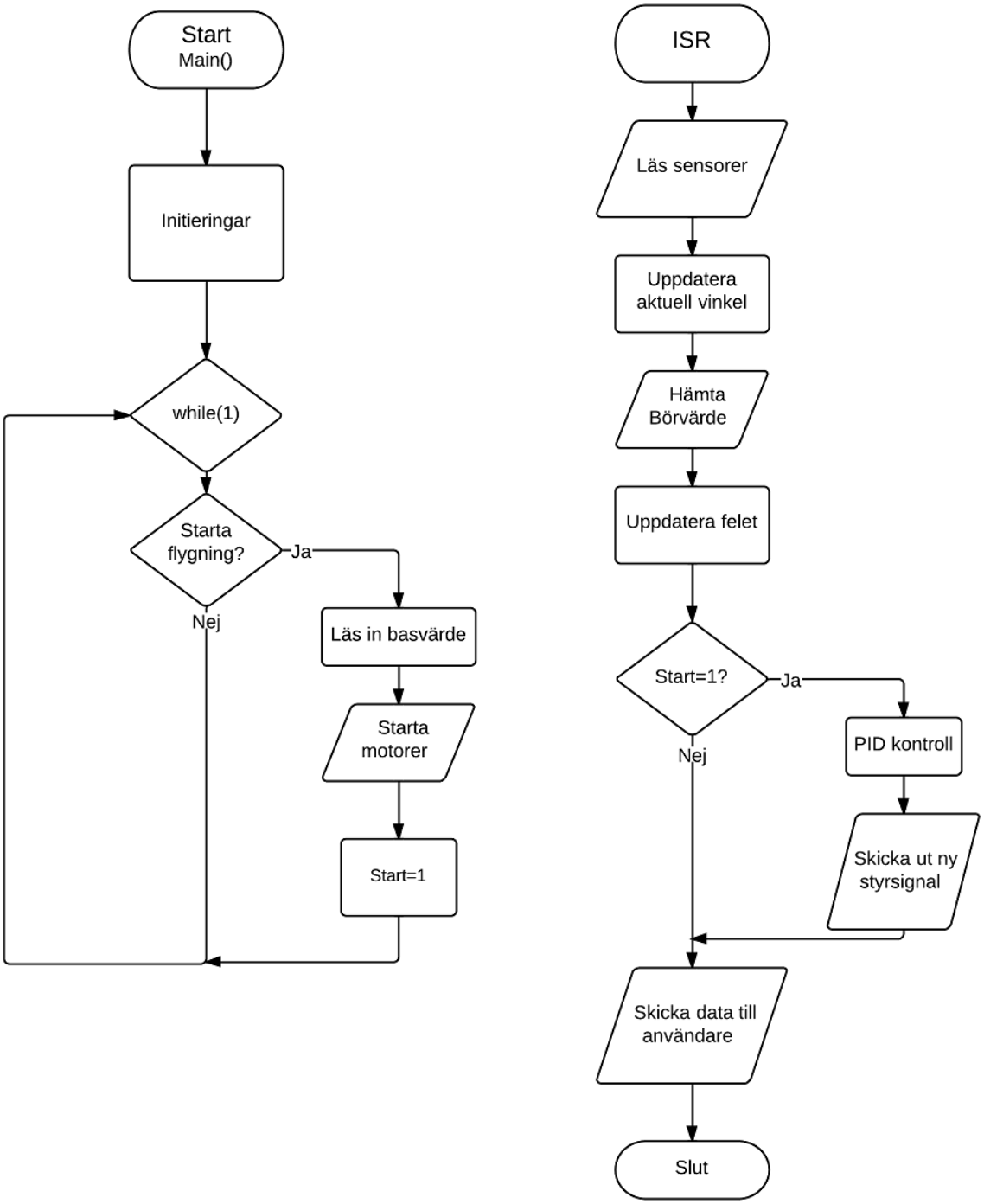
4.2 Mjukvaruutveckling

4.2.1 Programstruktur

Figur 4.3 nedan visar programstrukturen för de experiment som utförts i samband med vinkelkontrollen. Programmet består av en primär loop samt en avbrottsrutin, ISR (Interrupt Service Routine). Innan den primära loopen startar initieras nödvändiga applikationer så som sensorer, timer, PWM parametrar m.m. Därefter väntar programmet på ett startkommando från användaren som innebär att motorer aktiveras till det basvärde som nämns i avsnitt 2.4. Denna sekvens startar då en tryckknapp på utvecklingskortet trycks ned. Då regulatorerna är digitala och beräknar derivata- och integralapproximationer på aktuella fel relativt börvärdet är det viktigt att detta sker under konstanta tidsintervall. Därmed används en dedikerad avbrottsrutin för att läsa data från sensorerna och reglera felen. Avbrottsrutinen programmerades att exekveras efter 0.02 sekunder vilket ger en samplingsfrekvens på 50 Hz. Denna samplingsfrekvens valdes

för att säkerställa att sensorerna skulle hinna registrera ett nytt mätvärde innan programmet försöker läsa data från dem igen.

Avbrottsrutinen hämtar börvärdet över seriell kommunikation från Simulink. Varefter det aktuella felet uppdateras och en kontroll om startsekvensen är initierad utförs. Avslutningsvis sänds data från systemet till Simulink för visuell observation och behandling. Detta kan till exempel vara aktuell vinkel, styrsignaler m.m.



Figur 4.3 Flödesschema över programstrukturen.

Processorns systemklocka valdes genom en konfiguration av dess interna register till 168 MHz vilket är processorns maximala klockfrekvens. Detta för att säkerställa att processorn ska hinna bearbeta eventuella framtida krävande funktioner. Läsaren hänvisas även till avsnitt 4.3.1 och 4.3.2 där konfiguration av sensorerna behandlas.

4.2.2 Utvecklingsmiljö

All programmering av processorn utfördes i programmet CooCox CoIDE. Utvecklingsmiljön bygger på det öppna programmeringsspråket C. Integrerat i CoIDE finns CoBuilder som är en plattform speciellt framtagen för processorer av typen ARM Cortex M. I CoBuilder finns ett stort antal funktioner och bibliotek som förenklar bland annat startrutiner och skrivning av data till diverse konfigurationsregister i processorn. I CoIDE finns också ett felsökningsverktyg, CoDebugger, vilket gör det möjligt för användaren att följa programmet när det exekveras i processorn genom klassiska felsökningsfunktioner som breakpoints, step och övervakning av variabler [15]. Programmet överförs till processorn via en USB anslutning.

4.3 Läsning av sensorer

För att uppnå stabil flygning är det kritiskt att mäta farkostens vinkel noggrant. De tester som utfördes på sensorerna för undersökning och behandling av mätdata utfördes i samband med Experiment Pitch i avsnitt 4.4.2. Gyroskopet som användes ger data i enheten dps (Degrees per Second), alltså hur fort sensorn roterar kring respektive axel i rummet. För att beräkna farkostens faktiska vinkel i rummet används en elementär integralapproximation. Nackdelen med gyroskopet är att det är mycket känsligt och experiment visade att gyroskopet registrerade rörelser även när gyroskopet låg helt stilla. Detta fenomen är ett känt problem vid vinkelmätning och om en integralapproximation utförs på denna felaktiga data kommer även den faktiska vinkelberäkningen att bli felaktig. Vilket kallas att gyroskopet driftar [16]. Som beskrivet i avsnitt 2.2 kan farkostens vinkel även beräknas utifrån data från accelerometern. Fördelen med att mäta vinkel med accelerometern är att den inte utsätts för någon drift. Nackdelen med att bestämma vinkeln från accelerometerns data är att farkosten utsätts för starka vibrationer när motorerna är aktiva. Dessa vibrationer gör att data från accelerometern blir brusig och att endast beräkna pålitliga vinklar från accelerometern är omöjligt. En kombination av gyroskopet och accelerometerns egenskaper är alltså önskvärd och realiseras med ett komplementfilter, se avsnitt 4.3.3.

4.3.1 Konfiguration av gyroskop

Nedan följer en punktlista på hur gyroskopet konfigurerades följt av en förklarande text.

- ODR: 100 Hz
- Högpassfilter brytfrekvens: 8 Hz
- Skala: 2000 dps (70 mdps/digit)
- Kommunikation: I2C protokoll

ODR är en förkortning för 'Output Data Rate' och beskriver med vilken frekvens gyroskopet registrerar ny mätdata. Denna frekvens måste således vara större än samplingsfrekvensen 50 Hz för att samplade värden från gyroskopet ska vara tillförlitliga. Brytfrekvensen på 8 Hz, som är det största valbara värdet på gyroskopets interna högpassfilter, valdes för att filtrera bort en del av gyroskopets drift. Skalan på 2000 dps valdes för att gyroskopet inte skulle registrera små förändringar, även här i syftet att minimera gyroskopets drift. Kommunikationen mellan gyroskopet och processorn ställdes att följa standard I2C protokoll.

4.3.2 Konfiguration av accelerometer

Nedan följer en punktlista på hur accelerometern konfigurerades följt av en förklarande text.

- ODR: 100 Hz
- Högpassfilter: Ofiltrerad
- Skala: 2 g (18 mg/digit)
- Kommunikation: SPI protokoll

Då det är önskvärt att mäta långsamma förändringar i mätdata från accelerometern för att beräkna vinkel ställdes accelerometern att inte högpassfiltrera sin utsignal. Skalan på accelerometern ställdes till 2 g då accelerometern i syfte att mäta vinkel inte behöver kunna registrera starkare krafter. Kommunikationen mellan accelerometern och processorn ställdes att följa SPI protokoll.

4.3.3 Komplementfilter

Ett komplementfilter bygger på principen att låg- och högpassfiltrera mätdata från två sensorer som mäter samma sak på olika sätt för att få en mer noggrann mätning. Vinkeln, $\propto$, ut från ett komplementfilter beskrivs enligt ekvation (4.1) där g_1 och g_2 är uppmätt vinkel från sensor ett och två och a är filterkoefficienten. Filterkoefficienten kan bestämmas enligt ekvation (4.2) där τ är tidskonstanten för filtret och $t_s$är samplingstiden [17].

$$\propto = a \cdot g_1 + (1 - a) \cdot g_2 \qquad (4.1)$$

$$a = \frac{\tau}{\tau + t_s} \qquad (4.2)$$

Som beskrivet i avsnitt 4.3 har gyroskopet en lågfrekvent drift och accelerometern ett högfrekvent brus på utsignalen. Därmed högpassfiltrerades utsignalen från gyroskopet och utsignalen från accelerometern lågpassfiltrerades.

Då experiment visade att gyroskopet registrerade värden av ett medelvärde på ungefär 1.7 grader/s när gyroskopet låg helt stilla bör tidskonstanten vara under en sekund för att filtrera bort detta värde som ger upphov till gyroskopets drift. Dock kan inte en tidskonstant bestämmas med bara detta resonemang då det kan leda till att den mycket brusiga mätdata från accelerometern påverkar komplementfiltrets vinkelberäkning för mycket. En avvägning av dessa resonemang och testning av beräknade filterkoefficienter är alltså nödvändigt för att hitta den mest lämpliga tidskonstanten. Efter experimentella tester fastslogs värdet på filterkonstanten som gav en vinkel fri från drift och högfrekventa störningar, se figur 5.1 i avsnitt 5.1. Komplementfiltrets utsignal kunde då beskrivas enligt ekvation (4.3).

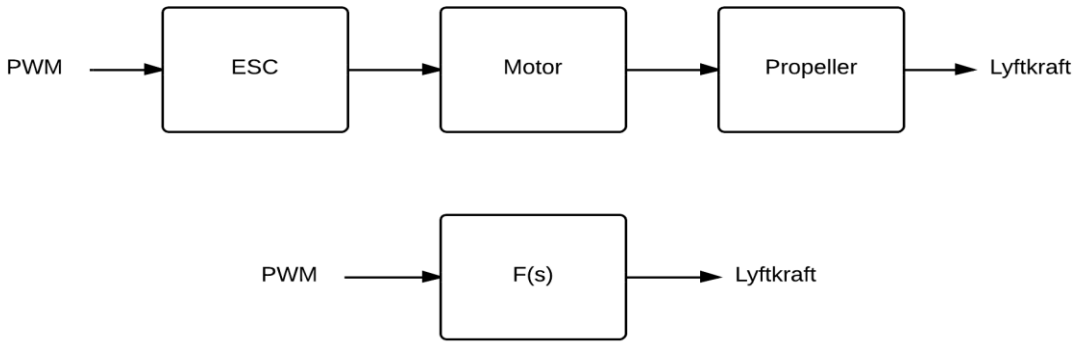
$$pitch = 0.98 \cdot (pitch_{Gyro} \cdot t_s) + 0.02 \cdot pitch_{Acc} \qquad (4.3)$$

4.4 Modellering av det dynamiska systemet

För att få en ökad förståelse för hur systemet beter sig under olika villkor har flertalet modeller tagits fram under arbetets gång. Detta för att kunna simulera vissa scenarion under en kontrollerad miljö vilket underlättar testförfarandet och dimensionering av regulatorer. I kapitlet undersöks olika metoder för modellering beroende på vilken del av systemet som modellerades. Centralt i metoden relaterade till vinkeln *pitch* var att mätdata från experiment inhämtades via seriell överföring till programvaran Simulink. Kapitlet inleds med de två experiment som varit centrala i modelleringen. Experimenten resulterade i två modeller. En för höjddynamiken samt en för vinkeldynamiken. Dessa modeller utgör därefter underlaget till dimensioneringen av regulatorer för systemet.

4.4.1 Experiment Lyftkraft

För att få ett matematiskt samband mellan lyftkraft från propellern och styrsignal från processorn utfördes ett experimentellt test på processen. I detta fall utgörs processen av ESC-Motor-Propeller, se figur 4.4.



Figur 4.4 Illustration över delprocesserna som ska representeras av en process.

Stigtiden för hur snabbt motorerna når det faktiska varvtalet efter att styrsignalen skickats mättes med en digital tachometer kopplad till ett oscilloskop. Denna stigtid varierar dock beroende på hur stor skillnaden är mellan motorns aktuella varvtal och varvtalet motorn ska nå upp till. Mätningar visade att en uppskattad stigtid på 0.3 sekunder är ett befogat värde att använda i simuleringen i avsnitt 4.5. En överföringsfunktion (4.4) med denna stigtid togs fram i Matlab.

$$G(s) = \frac{1}{0.1s+1} \qquad (4.4)$$

En av motorerna med propellern monterad fästes vid en finskalig digital våg och mikrokontrollern programmerades till att generera en PWM signal som med ett knapptryck

ökade signalens duty cycle och därmed effekten till motorn. Polariteten på motorn växlades så att lyftkraften från propellern blev riktad nedåt mot vågen. En tångamperemeter användes för att mäta strömmen till motorn vid de olika PWM signalerna. Motorerna tål en maximal strömstyrka på 10.2 A och testet avbröts när strömmätningen visade att denna gräns var uppnådd.

Uppmätta värden behandlades i Matlab och en linjär regression med minsta kvadratmetoden utfördes, se figur 5.2 i avsnitt 5.2.1. Den linjära regressionen resulterade i följande samband, ekvation (4.5), där variabeln *pwm* är insignal till processen i form av en PWM signal (0-255). Variabeln *M* är utslaget på vågen uttryckt i gram.

$$M = 5.1293 \cdot pwm - 343.4452 \qquad (4.5)$$

För att teorin presenterad i avsnitt 2.1 ska gälla krävs att lyftkraften från motorerna uttrycks i Newton. En multiplikation med antalet motorer, fyra, av sambandet måste även utföras för att beskriva systemets totala lyftkraft beroende på styrsignal.

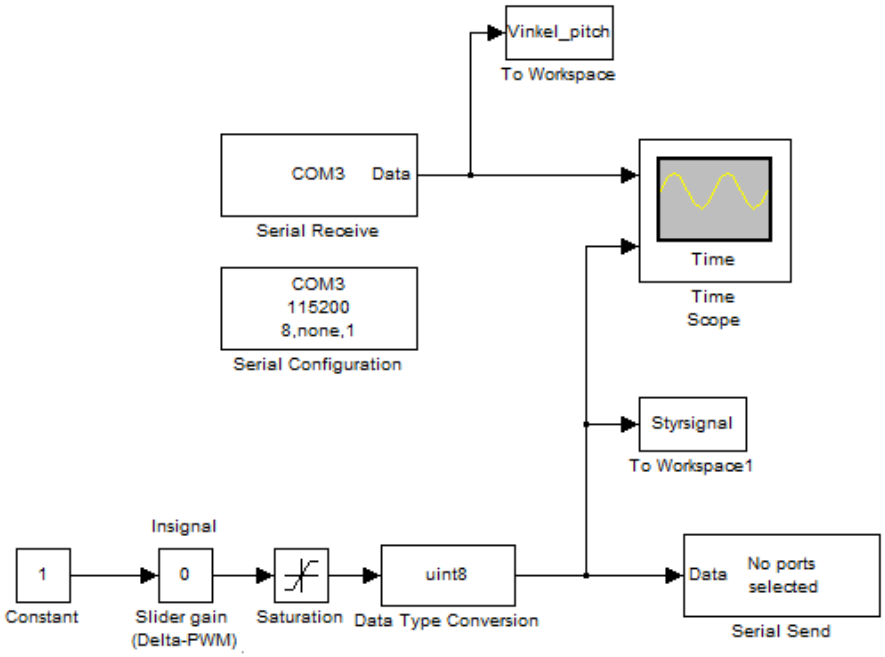
Omvandling till lyftkraft uttryckt i Newton för 4 motorer, g är gravitationskonstanten.

$$F = \frac{5.1293 \cdot pwm - 343.4452}{1000} \cdot g \cdot 4 = 0.2010 \cdot pwm - 13.4724 \qquad (4.6)$$

4.4.2 Experiment Pitch

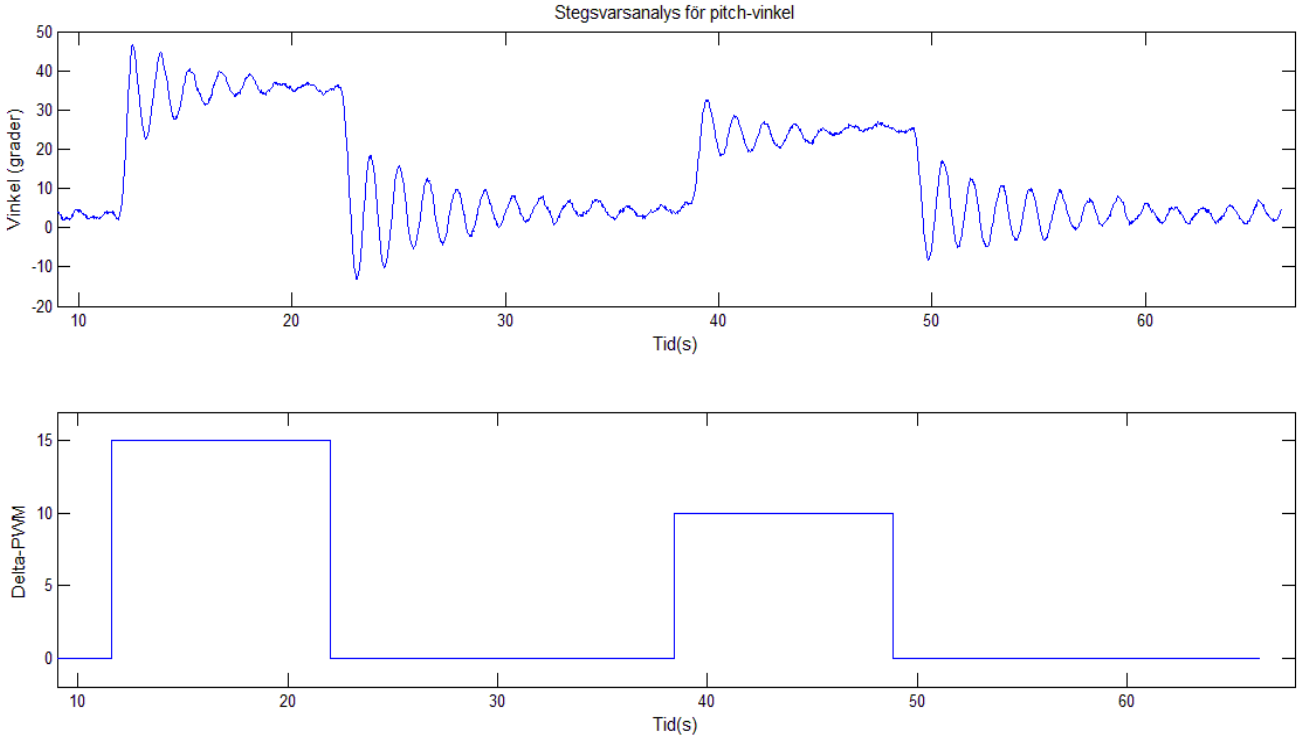
Målet med detta experiment var att hitta en överföringsfunktion för sambandet mellan skillnad i lyftkraft från motstående motorer och vilken vinkel i *pitch* detta medför att quadcoptern stabiliserar sig vid. Målet var även att kontrollera önskad funktion av sensorer och komplementfilter. Under detta experiment fästes quadcoptern i en testrigg som hindrade den från att lyfta samt att den endast kunde rotera kring en axel, rotation kring y-axeln. Följaktligen användes endast två motorer monterade vid detta experiment, motor 1 och motor 3. Som beskrivet i avsnitt 2.1.2 avgör skillnaden i varvtal hos de två motstående motorerna vilken vinkel quadcoptern antar. Som beskrivet i avsnitt 2.3, figur 2.3, utgörs alla styrsignaler till motorerna av ett basvärde som bestäms av höjdkontrollen. I detta experiment applicerades denna styrsignal till ca 55 % gaspådrag till respektive motor. Motiveringen för detta värde är att det ligger inom det intervall som motorerna antas arbeta i under en verklig situation.

Experimentet övervakades i realtid genom seriell kommunikation mellan processorn och Simulink. I figur 4.5 nedan visas hur programmet var utformat samt en förklaring av det.



Figur 4.5 Simulinkprogram för uppmätning av stegsvar för vinkeln *pitch*.

Mätdata erhålls och sänds i blocken Serial Receive respektive Serial Send. Nödvändiga konfigurationer för överföringen sker i blocket Serial Configuration. Mottagen data presenterades i ett Time Scope för visuell visning och skickades även till Matlab Workspace för vidare behandling i System Identification Toolbox. Vid användarens kommando skickas en insignal i form av en förändring till de PWM signaler i processorn som kontrollerar motorerna, detta värde benämns som *Delta-PWM* i figur 4.5 samt figur 4.6. Följaktligen resulterar en ändring i insignalen till en varvtalsändring hos de två aktuella motorerna. Processorn var programmerad att direkt addera samt subtrahera insignalen på styrsignalen till respektive motor och därmed utsätta systemet för en stegformad förändring i vinkeln *pitch*. Ett antal stegformade ändringar av insignalen gjordes med varierande amplitud, se figur 4.6. Figuren visar även den resulterande vinkel som systemet antog.



Figur 4.6 Uppmätt data från stegsvarsexperimentet.

I verktyget System Identification Toolbox användes data som experimentet resulterade i för att modellera processen där variablerna *Vinkel_pitch* är utsignal och *Styrsignal* är insignal, enligt figur 4.5. I verktyget användes en linjär modell för att beskriva processen. Den linjära modellen approximerades till en överföringsfunktion med fyra poler och ett nollställe i S-planet. I ekvation (4.7) nedan presenteras den resulterande överföringsfunktionen.

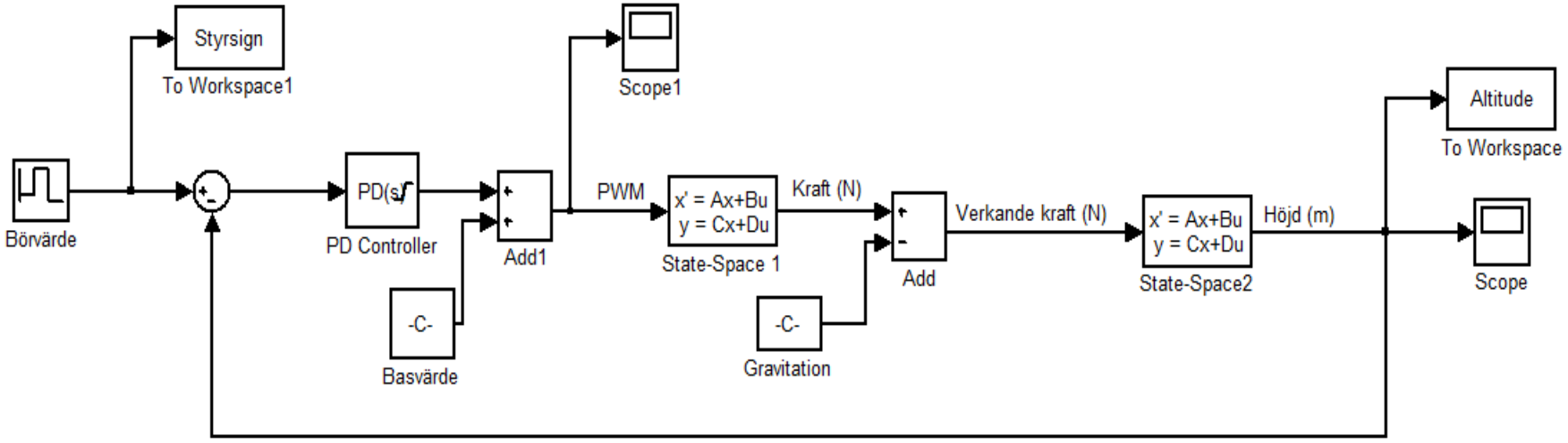
$$G_p = \frac{99.67s + 1.702}{s^4 + 2.607s^3 + 21.84s^2 + 47.21s + 0.5606} \qquad (4.7)$$

Resultatet från simuleringar på modellen redovisas i avsnitt 5.2.2 under kapitlet Modelleringsresultat. I figur 5.3 i avsnitt 5.2.2 jämförs hur väl modellen överensstämde med den fysiska processen.

4.5 Regulatordimensionering

4.5.1 Höjdreglering

Med en tillämpning av teorin i avsnitt 2.1 och det beräknade sambandet mellan lyftkraft och styrsignal presenterat i avsnitt 2.1.1 användes programmet Simulink för att dimensionera en höjdregulator. För att utföra en realistisk simulering var det nödvändigt att omvandla överföringsfunktionerna till tillståndsmodeller då systemet kräver begynnelsevillkor för att förhindra en negativ höjd under motorernas stigtid. Nedan presenteras en figur, figur 4.7, över hur simuleringen utfördes i Simulink följt av en förklaring av de olika blocken. De mest relevanta blocken förklaras med tillhörande rubrik i kursiv text. Resultatet av simuleringen behandlas i avsnitt 5.



Figur 4.7 Illustration över hur simuleringen av höjdregulatorn utfördes i Simulink.

PD Controller och Basvärde

Då överföringsfunktionen (2.3) presenterad i avsnitt 2.1.1 är av kvadratisk integrerande karaktär används endast en PD-regulator för att motverka den negativa fasvridning detta ger upphov till [6]. Konstanten Basvärde som adderas med styrsignalen från PD-regulatorn är det värdet på styrsignalen som ger upphov till jämvikt, alltså ingen förändring i höjd. Denna addering resulterar i att PD-regulatorn reglerar styrsignalen kring jämviktsnivån.

State Space 1

Detta block representerar lyftkraft från motorerna beroende på styrsignal. Styrsignalen betecknad PWM i figuren multipliceras med den linjära skillnaden i sambandet, ekvation (4.6), och för att simulera motorernas stigtid multipliceras detta med överföringsfunktionen (4.4) från avsnitt 2.1.1. Överföringsfunktionen som omvandlades till en tillståndsmodell är alltså överföringsfunktionen (4.8).

$$F_{upp} = \frac{0.2010}{0.1s+1} \cdot pwm \qquad (4.8)$$

Gravitation

Som beskrivet i avsnitt 2.1 verkar gravitationen konstant på quadcoptern och denna kraft måste subtraheras från lyftkraften från motorerna. Eftersom ekvation (4.6) presenterad i avsnitt 4.4.1 inte går genom origo måste även lyftkraften subtraheras med den negativa konstanten i ekvation (4.6). Quadcoptern väger 800 gram och g är gravitationskonstanten, detta block representerar således ekvation (4.9).

$$F_{ned} = 0.8 \cdot g + 13.4724 = 21.3178 \qquad (4.9)$$

State Space 2

Detta block representerar omvandlingen från verkande kraft till resulterande höjd, alltså är det överföringsfunktionen (2.3) presenterad i avsnitt 2.1.1 som här har omvandlats till en tillståndsmodell.

$$h(s) = \frac{1}{0.8s^2} \qquad (2.3)$$

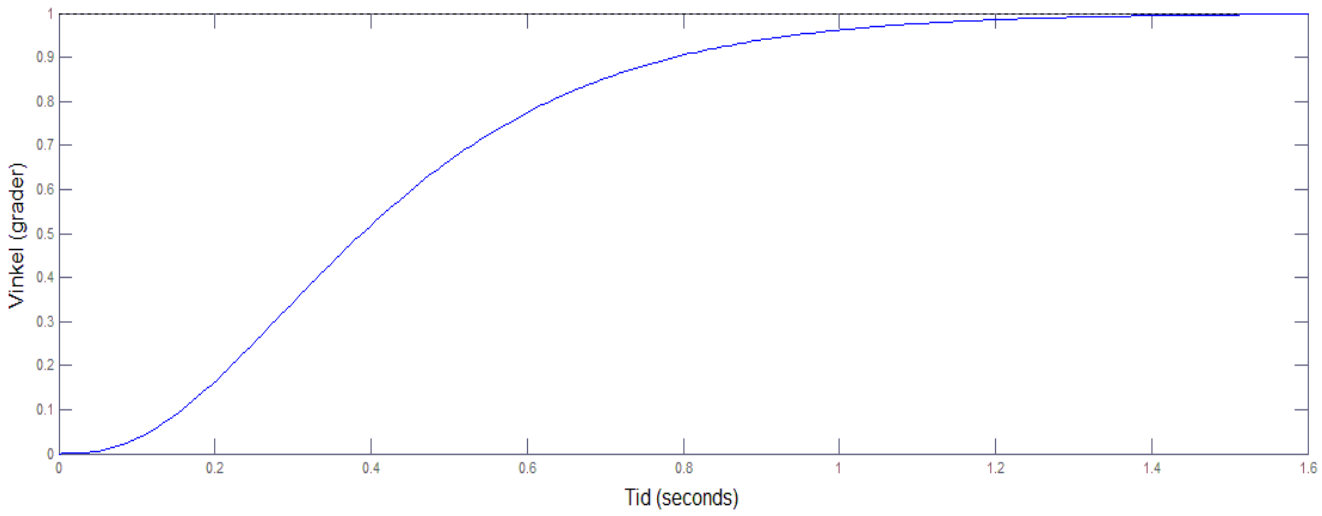
4.5.2 Pitchreglering

Följande avsnitt presenterar de två metoder som användes för att dimensionera två olika PID-regulatorer för *pitch* kontroll, en modellbaserad dimensionering samt en tumregelmetod. Dessa kommer att benämnas som *Regulator 1* respektive *Regulator 2* i resterande delar av rapporten. Teorin för dessa metoder presenterades i avsnitt 2.4. Båda de framtagna regulatorerna simulerades i Simulink innan implementering i processorn för test på hårdvaran. Simuleringsresultaten jämfördes också vilket kan ses i avsnitt 5.3 i figur 5.5. I dimensioneringen och simuleringen användes den linjära modellen för processen G_p, ekvation (4.7), beskriven i avsnittet 4.4.2 ovan. Denna modell kommer i resten av kapitlet att betecknas G_p då det är denna modell som uteslutande användes i simuleringarna för vinkelkontroll.

Regulator 1 - Modellbaserad Dimensionering

En total överföringsfunktion (4.10) för hela systemet med önskade egenskaper togs fram analytiskt, figur 4.8 illustrerar ett stegsvar för denna överföringsfunktion. Denna överföringsfunktion togs fram approximativt enligt den metod som presenteras i litteraturen [9]. Karakteristiken på stegsvaret konstruerades med hänsyn till stabilitet snarare än snabbhet hos systemet.

$$G_{tot}(s) = \frac{1}{0.0029s^3+0.0626s^2+0.4406s+1} \quad (4.10)$$



Figur 4.8 Stegsvar för önskad total överföringsfunktion, överföringsfunktion (4.10).

Med ekvation (2.9) presenterad i avsnitt 2.4 beräknades en överföringsfunktion för *Regulator 1* i Matlab vilket resulterade i överföringsfunktion (4.11).

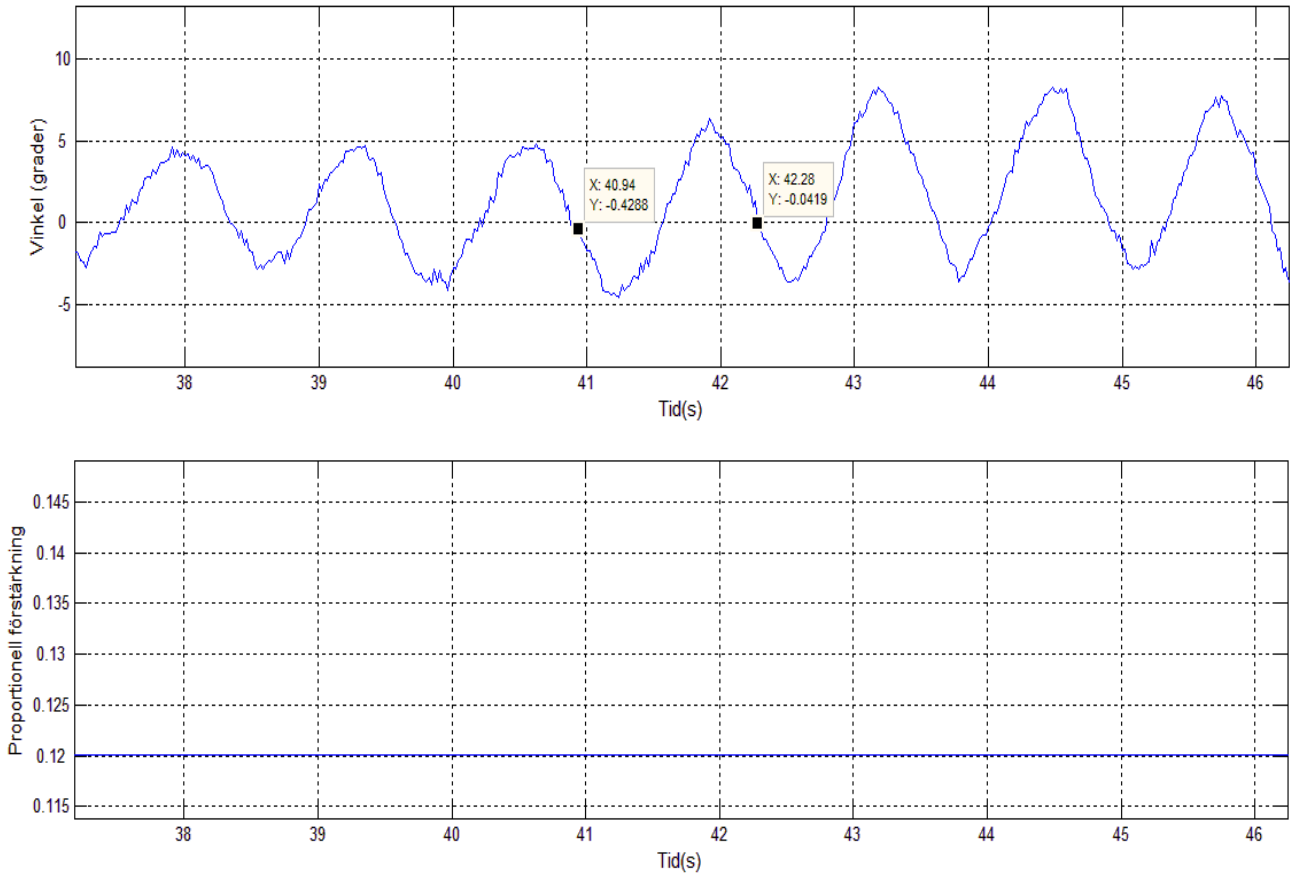
$$G_r = \frac{G_{tot}}{G_p(1-G_{tot})} = \frac{3.46s^4 + 9.019s^3 + 75.56s^2 + 163.3s + 1.94}{s^4 + 21.6s^3 + 152.3s^2 + 2.594\,s} \qquad (4.11)$$

För att implementera regulatorn i processorn måste den uttryckas som en differensekvation. En diskretisering med Tustins metod av överföringsfunktionen ovan utfördes i Matlab med samplingstiden 0.02 sekunder. Efter omskrivning resulterar detta i differensekvationen (4.12). Beräkningar för detta redovisas i appendix A.

$$u(k) = 2.889e(k) - 11.39e(k-1) + 16.85e(k-2) - 11.09e(k-3) + 2.743e(k-4) + 3.6u(k-1) - 4.848u(k-2) + 2.898u(k-3) - 0.6491u(k-4) \qquad (4.12)$$

Regulator 2 - Ziegler-Nichols Metoden

Den tumregelmetod som användes för dimensioneringen av denna regulator var Ziegler-Nichols metod. Ett Simulinkprogram där aktuell vinkel *pitch* samplades samtidigt som användaren skickar en styrsignal i form av proportionell förstärkning, K_0, till processorn konstruerades för ändamålet. I figur 4.9 nedan presenteras den självsvängning som K_0 gav upphov till. Som beskrivet i teorin bakom metoden i avsnitt 2.4 noterades periodtiden, T_0, för självsvängning samt den aktuella proportionella förstärkningen. Dessa presenteras nedan tillsammans med de beräknade parametrarna för PID-regulatorn.



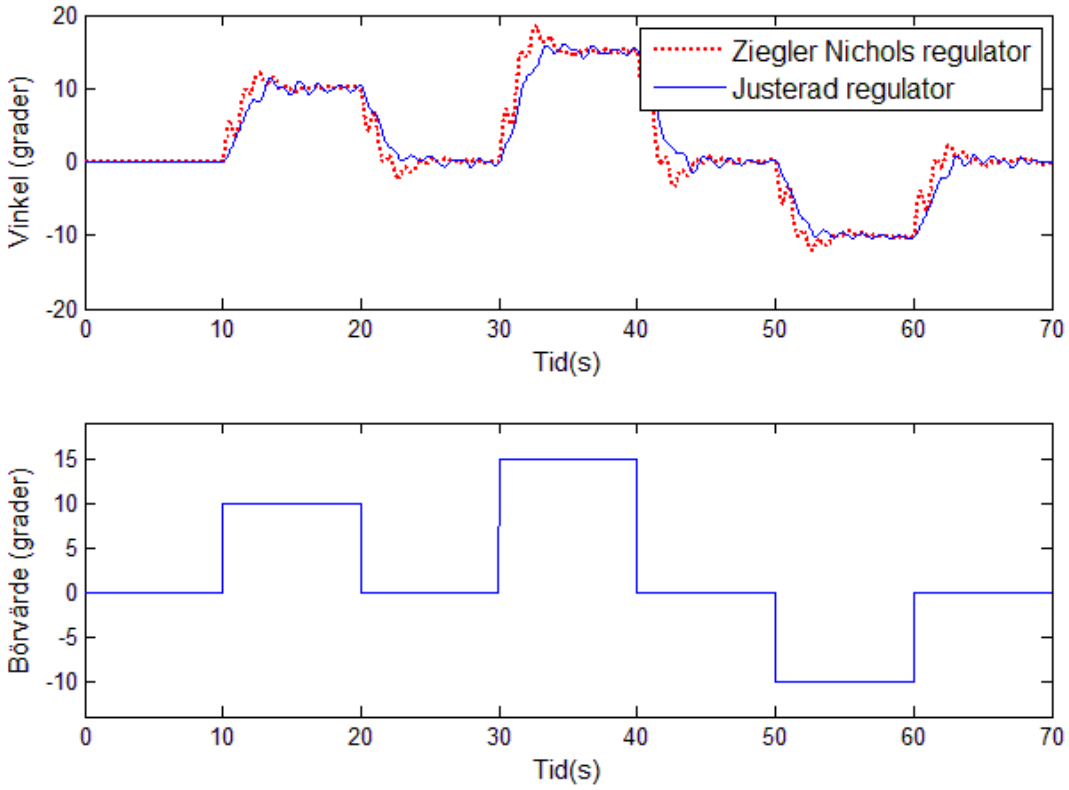
Figur 4.9 Graf över data som användes vid Ziegler-Nichols metod.

Uppmätta värden: $K_0 = 0.12,\ T_0 = 1.34$

Beräknade parametrar: $P = 0.072, I = 0.67,\ D = 0.1675$

Ziegler-Nichols metoden kan dock ibland resultera i ett system med dålig stabilitet [9] och simulering av regulatorn visade att ytterligare justeringar på parametrarna *P*, *I* och *D* behövdes. Resultatet av simuleringarna redovisas i figur 4.10 nedan där en jämförelse av den ojusterade samt den justerade regulatorn illustreras. Den justerade regulatorn resulterade i följande parametrar.

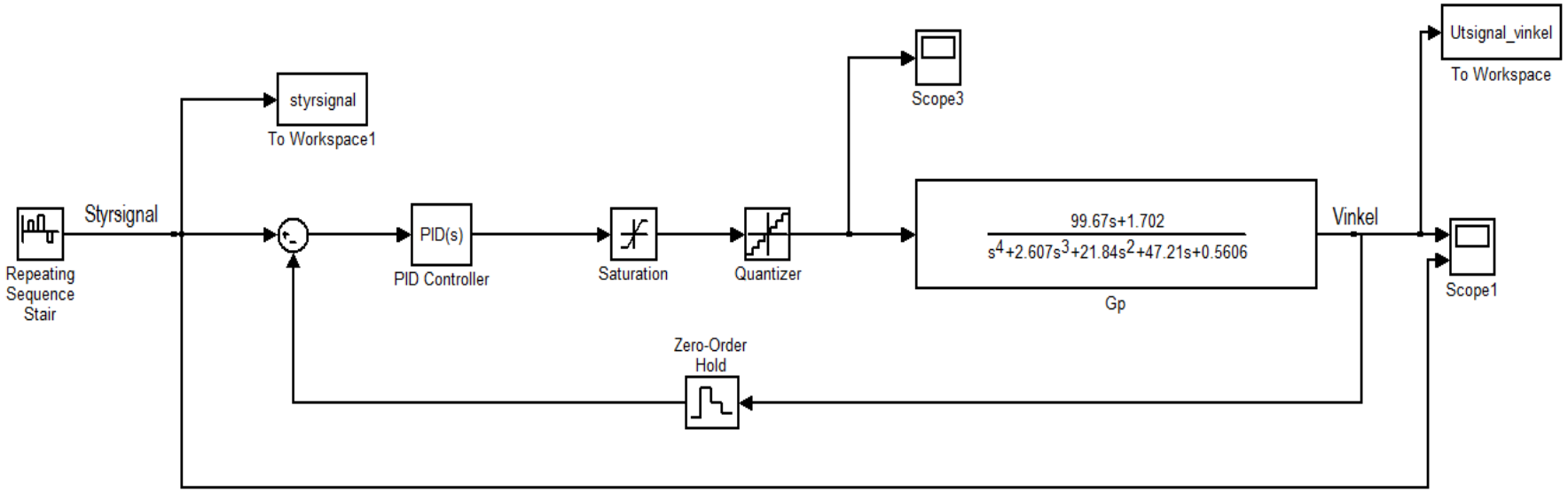
Regulator 2: $P = 0.0033,\ I = 0.33,\ D = 0.02$



Figur 4.10 Illustration över insvängningsförloppet för den ojusterade och den justerade regulatorn.

4.6 Simulering och Tester

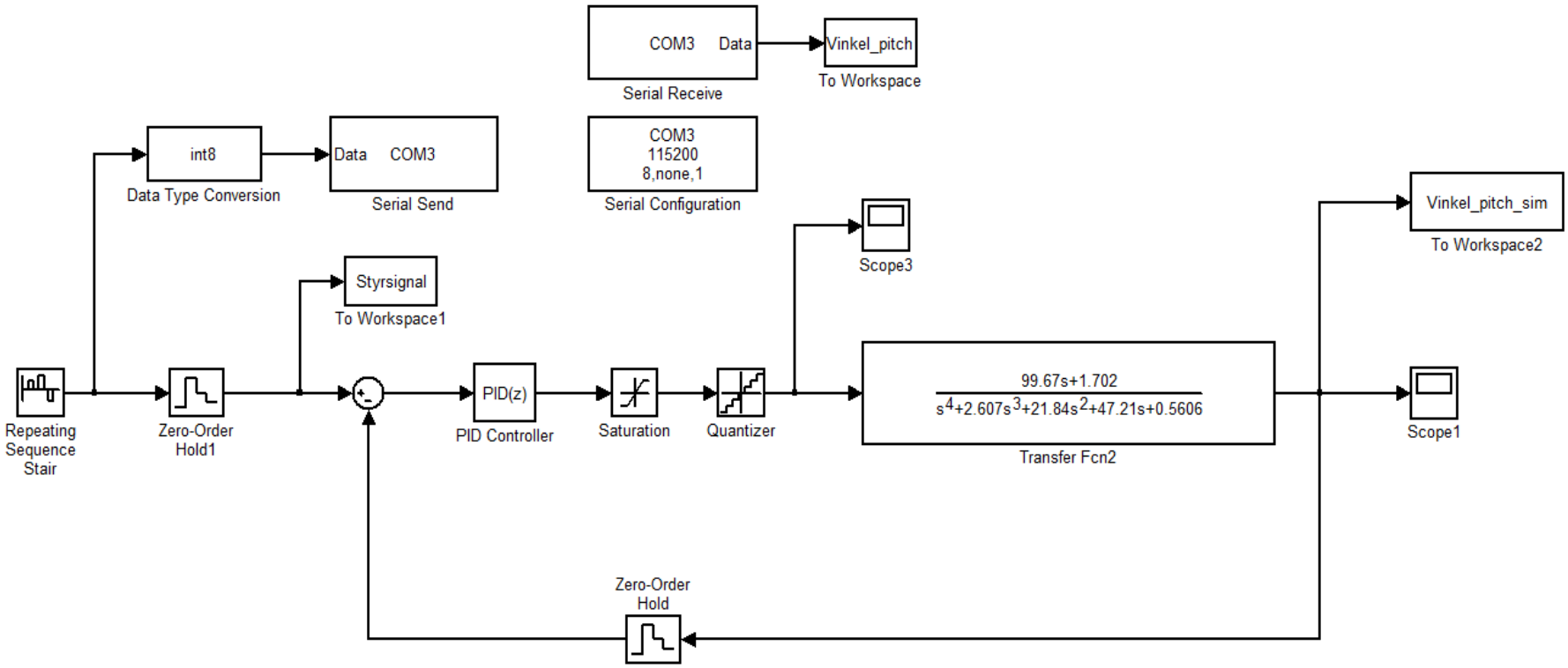
Som inledningen på kapitlet nämnde begränsas testförfarandet på hårdvaran till endast vinkelkontroll. De två regulatorer som konstruerades för detta i avsnitt 4.5 ovan simulerades i Simulink. Processen, G_p, som användes i simulationen var ekvation (4.7) som utarbetades i avsnitt 4.4.2. Figur 4.11 nedan beskriver hur simulationerna utfördes. Två simuleringar utfördes, en för varje regulator, och dessa var identiskt programmerade utöver vilken regulator som implementerades i blocket PID Controller. Resultatet samt en jämförelse av regulatorerna i simuleringsmiljön redovisas i avsnitt 5.3 i figur 5.5.



Figur 4.11 Simulering av PID-regulatorer för kontroll av *pitch* vinkel.

De båda regulatorerna implementerades därefter i hårdvaran. Strukturen på programkoden skiljer sig för respektive regulator. *Regulator 1* tillämpas i mjukvaran med differensekvationen beskriven i ekvation (4.12). I appendix B redovisas programkoden för denna regulator. *Regulator 2* appliceras med konventionell aritmetik utifrån de justerade parametrarna *P*, *I* och *D*. Programkoden för *Regulator 2* är även den redovisad i appendix B.

Ett slutgiltigt test utfördes för att undersöka hur väl det verkliga systemet stämde överens med regulator samt modell, G_p, i simuleringsmiljön. Ett simulinkprogram konstruerades där programmet skickar styrsignal till både det fysiska systemet, quadcoptern, samt till simuleringsmodellen med tillhörande regulator. Se figur 4.12 nedan. Ett antal stegformade ändringar på styrsignalen utfördes med ett intervall av 10 sekunder under en simuleringstid på totalt ca 65 sekunder. Uppmätt vinkel från det fysiska systemet samplades och jämfördes med den vinkel som simuleringen resulterade i. Resultatet av detta redovisas i figur 5.3 i avsnitt 5.3. I testet implementerades *Regulator 2* som regulator till vinkelkontrollen. Anledningen till valet av denna regulator beror på att tester av systemet där *Regulator 1* var implementerad visade att systemet betedde sig felaktigt och okontrollerat. En mer omfattad diskussion kring orsaken till detta redogörs i kapitel 6.



Figur 4.12 Simulinkprogram för att samla in data och jämföra simulerat och verkligt insvängningsförlopp.

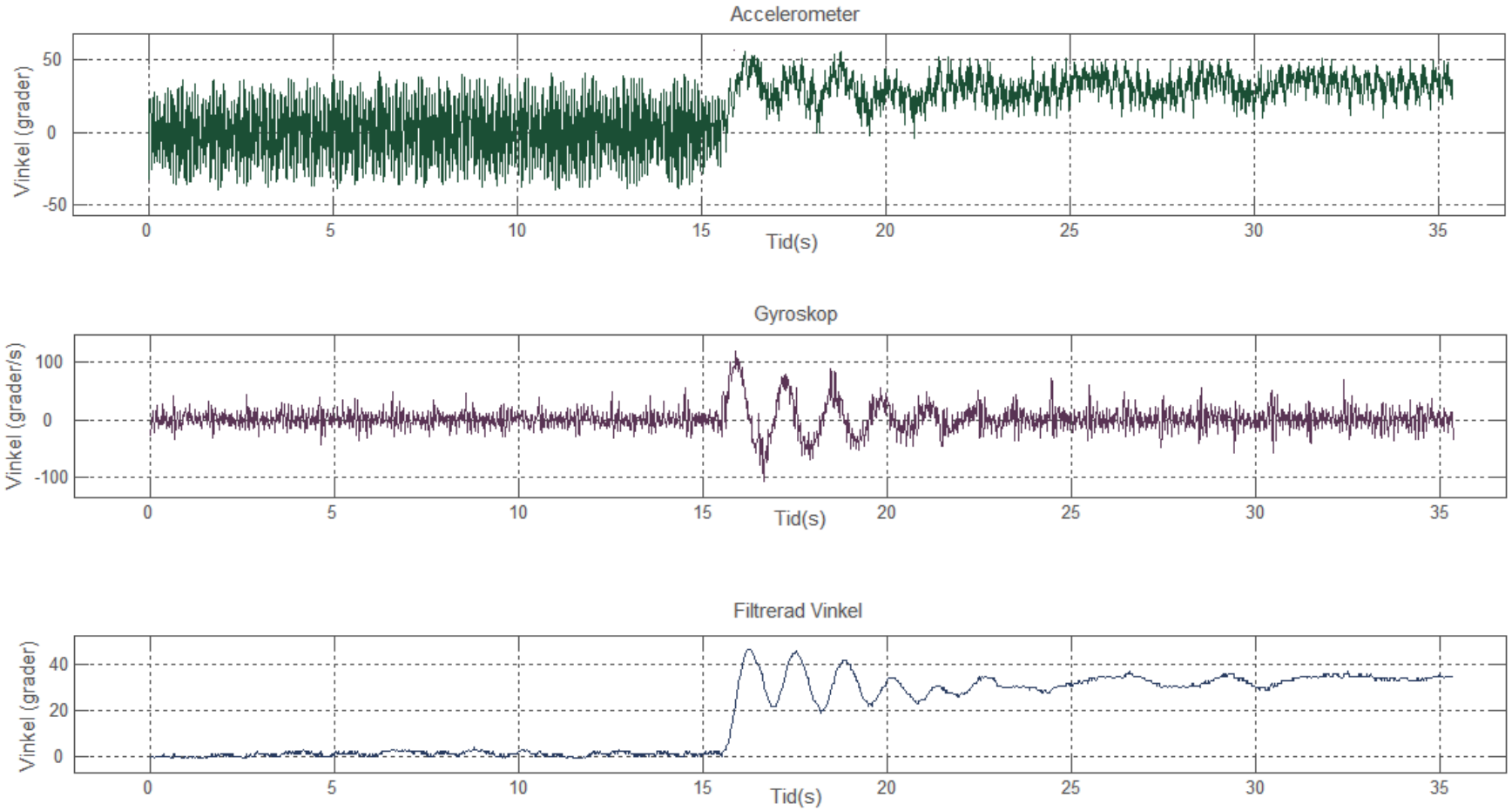
5. Resultat

I detta avsnitt presenteras resultaten från de experiment och simuleringar som utförts i metodkapitlet i rapporten. Kapitlet är uppdelat i avsnitt som behandlar resultat från de mest relevanta delarna av rapporten. Mätdata och simuleringar redovisas och förklaras samt diskuteras.

5.1 Mätdata från sensorer

I samband med experimentet för *pitch* vinkel i avsnitt 4.4.2 utfördes även tester av sensorerna. Figur 5.1 nedan illustrerar mätdata från accelerometern, gyroskopet samt komplementfiltrerad data vid en stegformad ändring på insignalen. I denna figur syns det tydligt att accelerometerns data innehöll mycket brus på grund av vibrationer från motorerna. I mätdata från gyroskopet syns även där relativt mycket brus trots att dess interna filter är aktiverade. Det är detta brus som ger upphov till gyroskopets drift efter integralapproximationen.

Den filtrerade vinkeln är däremot relativt störningsfri och uppfyller de krav som ställs på vinkelmätningen. Att den filtrerade vinkeln oscillerar något kring nollnivån innan den stegformade förändringen sker är ingen störning ur ett mätperspektiv. Detta uppstod då systemet inte var helt stabilt i utgångsläget. De olinjäriteter som systemet innehar samt testriggens inflytande på experimentet anses vara orsaken till detta. Total jämvikt under en längre tid mellan motor ett och tre var alltså svårt att realisera i experimentet. Då systemet utsattes för en stegformad förändring var det också väntat att systemet skulle oscillera en viss tid innan det stabiliserades någorlunda kring en ny vinkel. Sammanfattningsvis vittnar resultaten om att sensorerna med tillhörande komplementfilter fungerar önskvärt och att ytterligare utveckling i nuläget inte är nödvändigt.



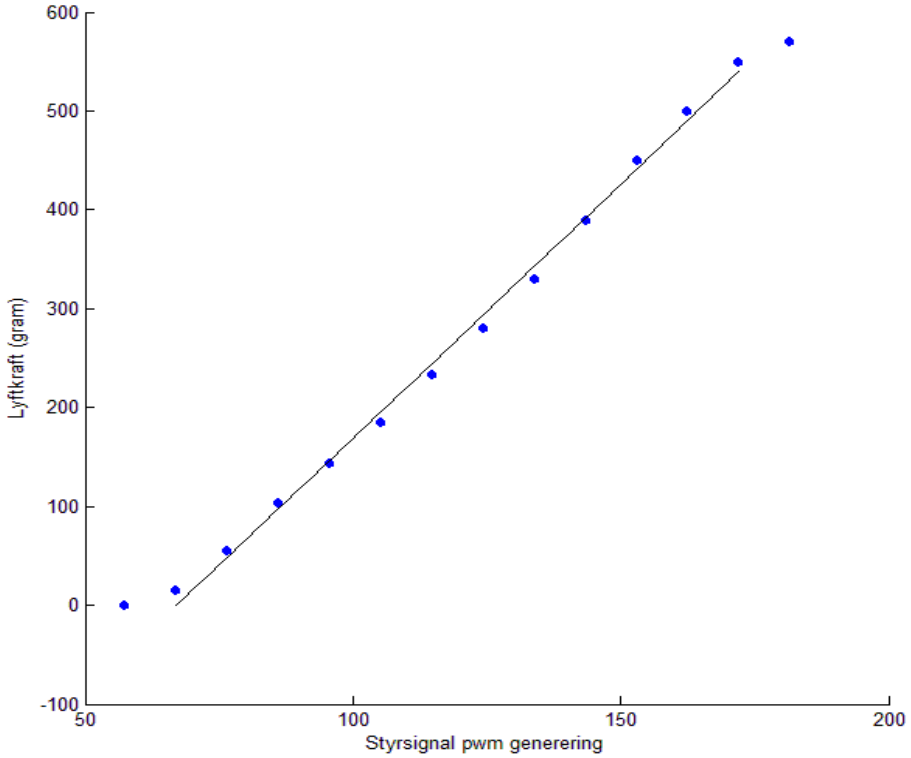
Figur 5.1 Grafer över beräknad vinkel från accelerometer, gyroskop samt den komplementfiltrerade vinkeln.

5.2 Modelleringsresultat

Nedan presenteras resultatet av metoderna för modellering beskrivna i kapitel 4.

5.2.1 Experiment 1: Lyftkraft

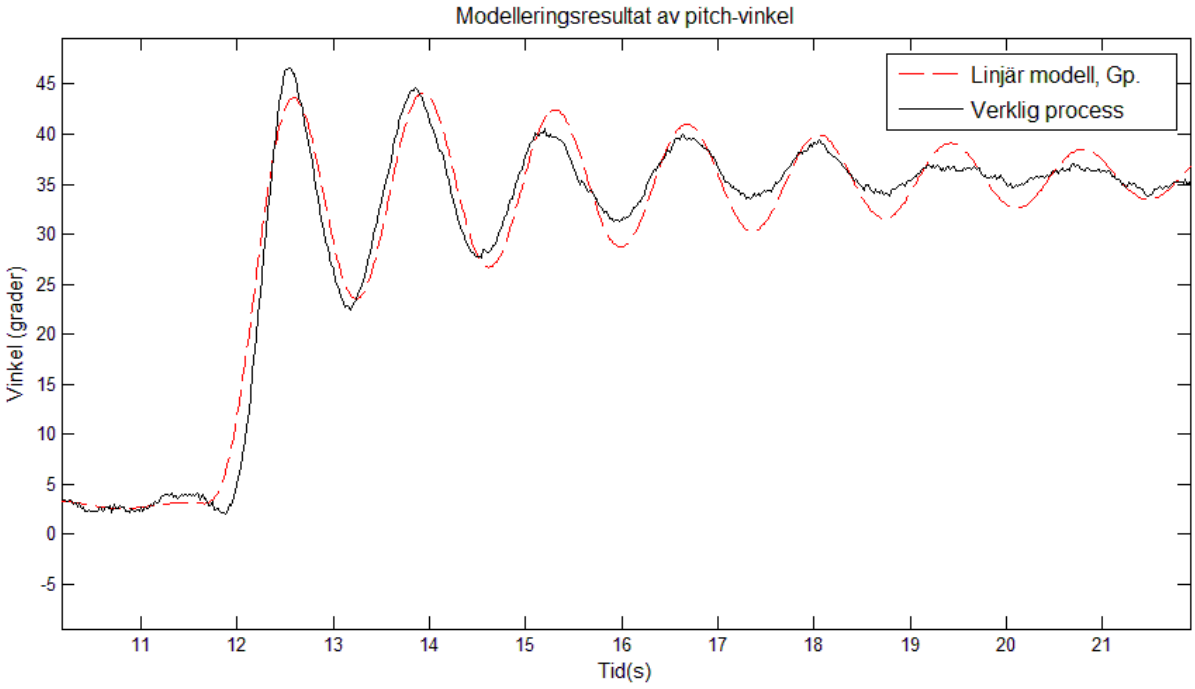
Figur 5.2 nedan illustrerar mätpunkter och det linjära sambandet som togs fram genom minsta kvadratmetoden i Matlab. Då sambandet inte är strikt linjärt samt att quadcoptern under flygning sällan kommer utsättas för max- eller minimala styrsignaler uteslöts den första och sista mätpunkten ur regressionen för att få en mer precis approximation i det intervall styrsignalerna oftast kommer att arbeta i.



Figur 5.2 Graf över mätpunkter samt det approximerade linjära sambandet mellan lyftkraft och styrsignal.

5.2.2 Experiment 2: Pitch

I figur 5.3 jämförs den linjära modellen med den verkliga processen. Den linjära modellen togs fram med System Identification Tool i Matlab och verktyget beräknade att modellen stämde överens med den verkliga processen med 85.3 %.

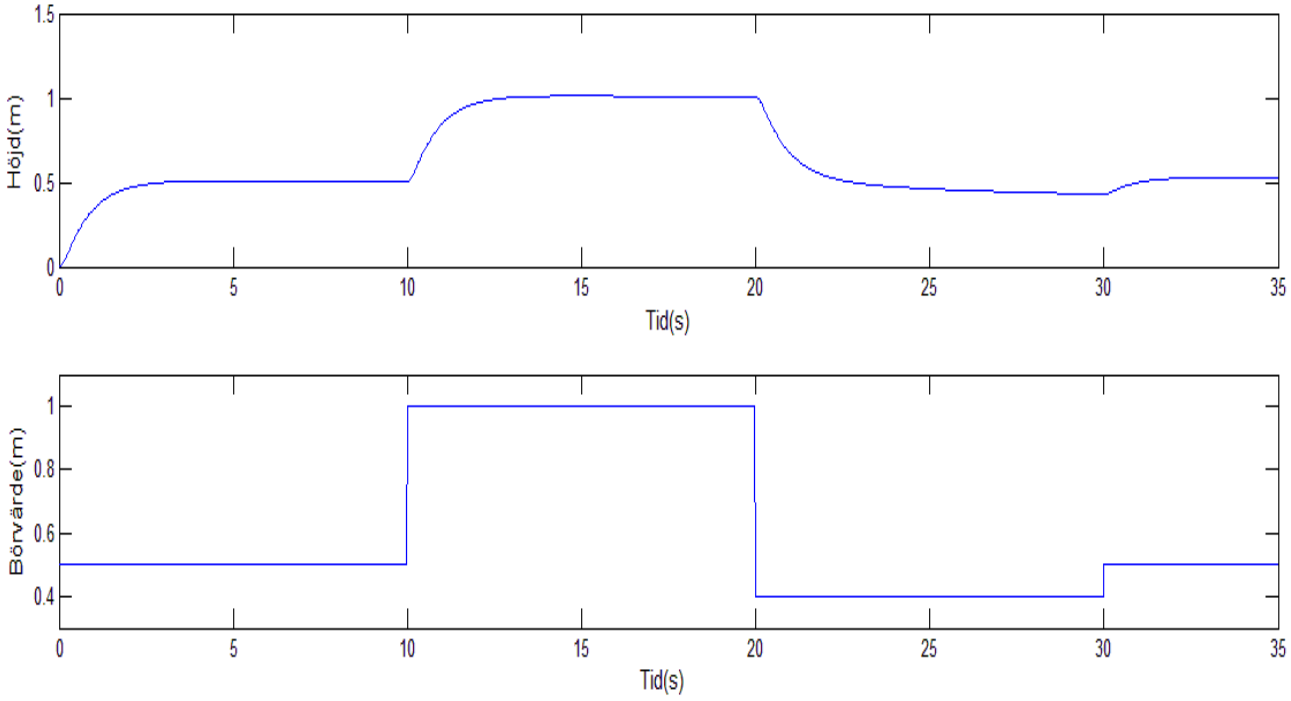


Figur 5.3 Jämförelse över modellerad och verklig process.

5.3 Simulering och test av regulator

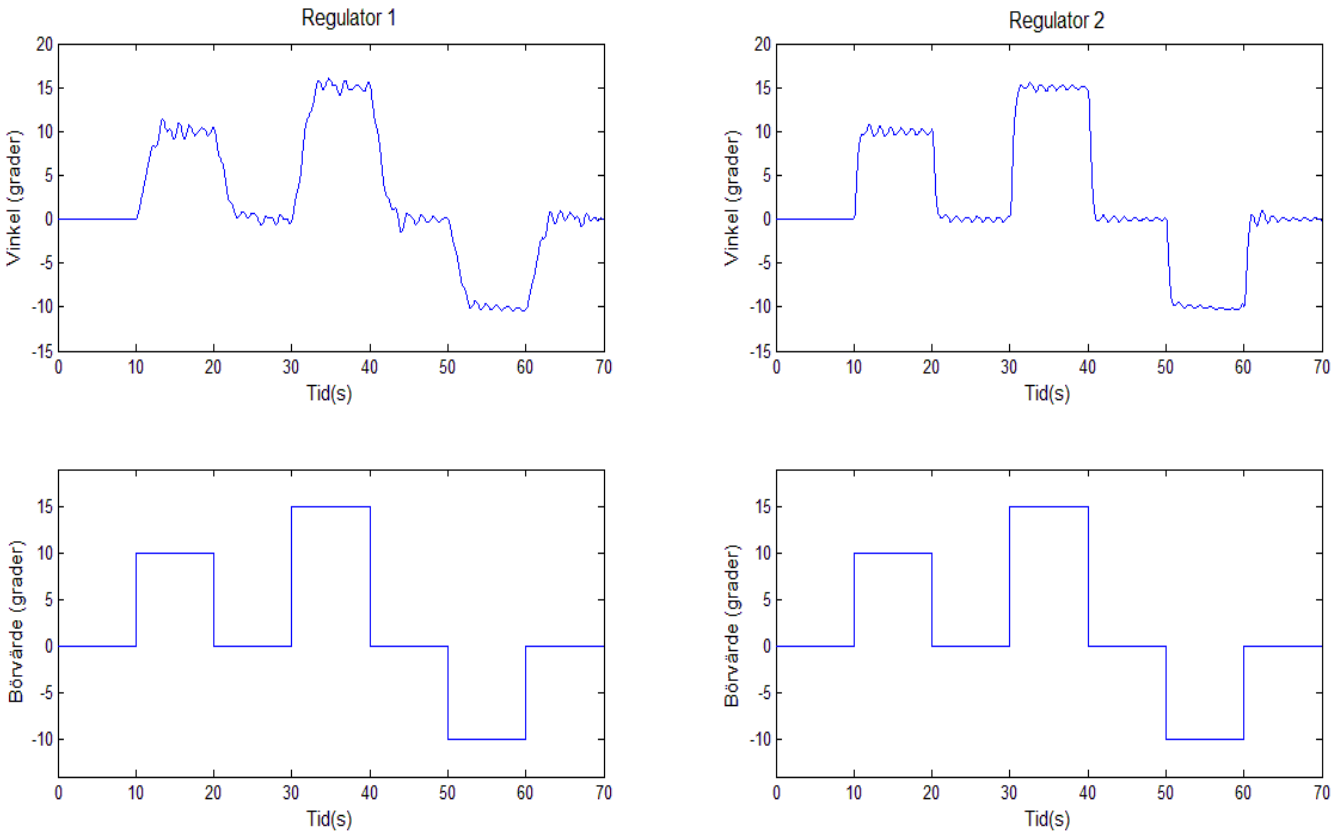
Nedan presenteras resultaten av de simuleringar och tester som utfördes i kapitel 4. Inledningsvis framförs resultatet från höjdkontrollen, därefter presenteras resultatet från vinkelkontrollen.

I figur 5.4 nedan visas resultatet av den simulering som utfördes på höjdkontrollen i avsnitt 4.5.1. Att höjdkontrollen inte har precis samma insvängningsförlopp då systemet stiger i höjd som när systemet minskar i höjd beror på att PD-regulatorn endast kan styra systemets lyftkraft och den enda kraften som kan påverka systemet att minska i höjd är den konstanta gravitationskraften. PD-regulatorn kan således styra systemet att generera lyftkraft och därmed stiga i höjd snabbare än vad gravitationskraften kan få systemet att minska i höjd. Då regulatorn ska fungera som önskat för både en stigning och en minskning i höjd måste därför parametrarna i PD-regulatorn justeras så att insvängningsförloppet för både stigning och minskning är så lika som möjligt.



Figur 5.4 Simuleringsresultat av systemets insvängningsförlopp vid stegformade börvärdesändringar i höjd.

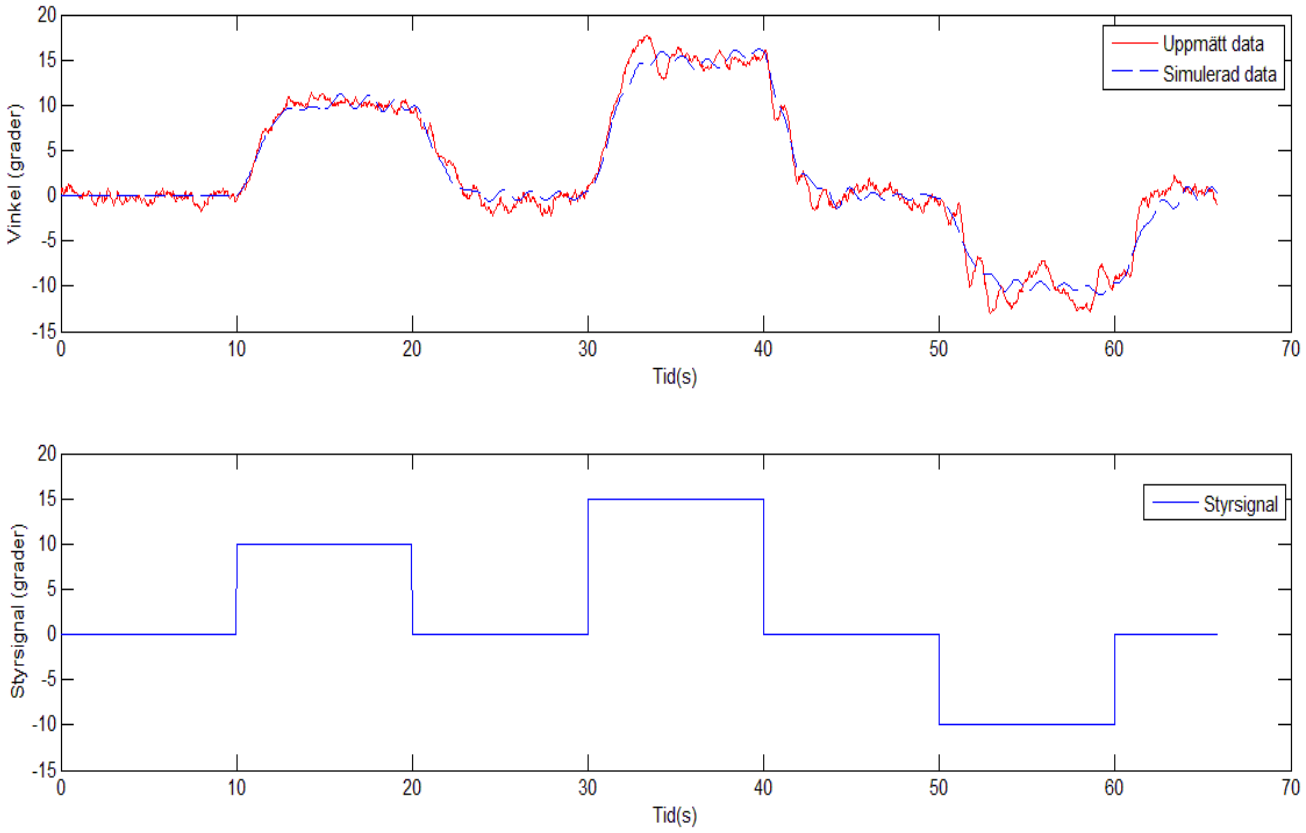
I figur 5.5 nedan presenteras resultatet från simuleringen i avsnitt 4.6 där de båda regulatorerna för vinkelkontroll jämfördes. Som figuren visar ger båda regulatorerna upphov till en liten oscillation kring börvärdet vid simulering men då dessa är relativt små ansågs regulatorerna vara tillräckligt tillförlitliga för att implementeras i processorn och testas på det verkliga systemet i testriggen.



Figur 5.5 Jämförelse av simuleringsresultat av de två regulatorerna för vinkelkontroll.

Efter implementering av regulatorerna i hårdvaran utfördes enligt figur 4.12 i avsnitt 4.6 ett test för att undersöka hur väl simuleringsmiljön stämmer överens med det verkliga systemet. Den modellbaserade regulatorn, *Regulator 1*, visade sig i test på det verkliga systemet inte följa simuleringsresultatet. Systemet reagerade visserligen på börvärdesändringar men med kraftiga översvängar som ledde till att systemet aldrig stabiliserade sig på börvärdet. Anledningen till detta berörs mer utförligt under Diskussion i kapitel 6. Därmed användes enbart *Regulator 2* i detta test och följaktligen presenteras resultatet nedan enbart för denna.

I figur 5.6 har simulerings- och systemets verkliga resultat ritats för att visuellt jämföra resultaten. Denna figur visar att det verkliga systemet har en tendens att oscillera mer än vad simuleringsresultaten visade men även att systemet svarar på börvärdesändringar och ställer in sig efter dessa. Dock visade det sig att systemet inte lyckades stabilisera sig lika tillförlitligt och snabbt då en yttre störning applicerades på systemet, alltså när systemets vinkel ändrades för hand.



Figur 5.6 Jämförelse mellan simulerat och verkligt insvängningsförlopp.

6. Diskussion

Resultatet av rapporten tyder på att styrsystemet för vinkelkontroll kan realiseras med de metoder samt den hårdvara som användes. Detta kan ses som en utvärdering av resultatet utifrån den frågeställningen som ställdes; att undersöka om önskad funktion kan uppnås med den tänkta metoden samt de specifika komponenter som systemet innehåller.

På grund av projektets ingångsvinkel, att mycket av arbetet med styrsystemets komponenter fick utföras från grunden, gick mycket av tiden i början av projektet åt att bekanta sig med dessa. En tidig utmaning i arbetet med hårdvaran var att få en förståelse över utvecklingsmiljön för mikrokontrollern. Vid eftertanke kanske en något mer tillförlitlig och användarvänlig utvecklingsmiljö hade varit önskvärd. Den använda utvecklingsmiljön anses dock vara fullt gångbar med tanke på den kostnadsfria aspekten. Ytterligare felsökningstid gick åt då problem uppstod med den seriella kommunikationen mellan utvecklingskortet och Simulink. För insamling av data skulle alternativa program kunna vara exempelvis Labview. För simuleringar samt modellering är dock Matlab och Simulink väsentliga i arbetet.

Anledningen till att den modellbaserade *Regulator 1* inte fungerade tillförlitligt i det verkliga systemet kan bero på att modellen av processen inte stämde överens tillräckligt bra med den verkliga processen. På grund av tidsbrist fanns det inte utrymme att vidare felsöka detta felaktiga beteende hos regulatorn för att bestämt fastslå denna anledning. Observationer av styrsignalerna från denna regulator visade dock på mycket kraftiga styrsignaler precis när en börvärdesändring skedde vilket förmodligen förklarar de kraftiga översvängningarna som uppstod. Eftersom denna regulator var uppbyggd kring en differensekvation var det heller inte möjligt att finjustera parametrarna i denna på samma sätt som är möjligt i en traditionell PID-regulator. Beräkningarna bakom denna differensekvation kontrollräknades och gjorde det möjligt att utesluta felaktiga beräkningar som en källa till problemet. En felsökning av programkoden för regulatorn visade initialt inte några konkreta fel. Dock fanns det, av tidsbrist, inte möjlighet för fortsatt felsökning av programkoden där eventuella strukturfel skulle kunna utgöra problemet.

En viktig faktor som starkt påverkar hur systemet beter sig är vart systemets tyngdpunkt ligger. Anledningen till att detta inte behandlats i rapporten är att det beställda chassit minimerar möjligheterna att manipulera denna tyngdpunkt då det av fysiska skäl endast är möjligt att montera batteriet, som påverkar tyngdpunkten mest, på två ställen på chassit. Alla experiment utfördes med batteriet monterat på undersidan av chassit vilket ledde till att systemet fick en låg tyngdpunkt. En låg tyngdpunkt innebär att systemet naturligt stabiliserar sig vinkelrätt mot z-axeln vid total jämvikt mellan motstående ändar på systemet. Det krävs dock ett större arbete för systemet att ställa in sig och bibehålla en icke vinkelrät vinkel mot z-axeln med denna tyngdpunkt. Som figur 4.6 i avsnitt 4.4.2 illustrerar är tiden för oscillationerna större kring vinkeln noll och avtar något snabbare då systemet antar en lutning. Detta kan förklaras med att

ett visst motstånd påverkar systemet i detta läge vilket kan tänkas dämpa oscillationerna. Att systemet oscillerar kring nollnivån redan innan den stegformade styrsignalen kommer beror på att total jämvikt mellan motstående motorer var svår att realisera i testriggen. På grund av tidsbrist fanns det inte utrymme att flytta tyngdpunkten då detta hade lett till en annan modell av processen och således hade det krävts att göra om både experiment 2: Pitch och dimensionera nya regulatorer för denna process.

Frågan om vilken inverkan en flygande autonom farkost kan ha på vårt globala samhälle är en svår fråga att besvara. Tekniken kan tänkas tillämpas i en mängd olika områden. En kamera kan monteras för att utföra kostnadseffektiv flygfotografering och i framtiden kan den användas för att leverera paket inom budbranschen [18]. Att människor finner glädje genom hobbyverksamhet är också ett tillämpningsområde som det redan idag finns en stor kommersiell marknad för. Tekniken kan även tillämpas i mer etiskt ifrågasatta områden. Istället för en kamera kan ett vapen monteras och farkosten kan då användas för ett militärt syfte med potentialen att både förstöra egendom och släcka människoliv. Den som styr en sådan farkost kan argumentera att tekniken används för att beskydda demokrati, mänskliga rättigheter eller hävda någon annan "legitim" anledning. Vart gränsen ska dras för vad som är etiskt korrekt eller inte är en politisk fråga och bör därför behandlas som en sådan.

7. Slutsats

Slutsatsen från projektet är att det resulterade i en initial prototyp för ett kontrollsystem till höjd- och vinkelkontroll med tillhörande teoretiska modeller för systemets dynamik. Samtidigt betonas målet som var att konstruera en första modell till ett slutgiltigt komplett styrsystem. Initialt i projektet var det svårt att bedöma hur det slutgiltiga resultatet skulle se ut. Mycket på grund av det som nämnts i diskussionen ovan, att många delar i projektet utfördes från grunden och den osäkerhet detta medför. Med detta i åtanke samt de utmaningar som ställts under arbetets gång anses projektet uppnå de mål som angavs. I appendix D presenteras och kommenteras den preliminära tidplan som togs fram vid projektets start.

7.1 Förbättringar

En betydande del i de förbättringar som kan göras på systemet anses vara regulatorn för vinkelkontrollen. En önskad funktion nåddes visserligen, dock observerades brister vid t.ex. yttre störningar och snabba börvärdesändringar i systemet. Ytterligare undersökningar och utveckling behöver därför göras gällande regulatorn för vinkelkontroll. Förslagsvis kan andra typer av regulatorer implementeras och testas. Litteraturen nämner ett flertal av dessa, exempelvis Fuzzy Control. En regulator med två frihetsgrader har också diskuterats och skulle kunna vara en möjlig lösning.

För att ytterligare förbättra noggrannheten vid vinkelmätning skulle ett lågpass filter kunna appliceras på accelerometern för att reducera en del av bruset. Efterforskning har även visat att ett Kalmanfilter, som är ett alternativ till komplementfiltret och bygger på mer avancerad matematik, kan leda till en mer exakt vinkelmätning. Dock anses det nuvarande komplementfiltret vara fullt gångbart för det nuvarande ändamålet [19].

En högre upplösning på ESC:erna borde resultera i en mer precis reglering då varvtalet på motorerna och därmed lyftkraften från dem då kan styras med en större noggrannhet. Att flytta upp batteriet till den andra möjliga positionen resulterar i att systemet får en något högre tyngdpunkt och skulle eventuellt göra systemet enklare att reglera.

7.2 Vidareutveckling

För att möjliggöra höjdkontroll krävs det sensorer för att mäta quadcopterns höjd. Ett förslag på sensorer som kan implementeras för detta ändamål är MA40S4S och MA40S4R från tillverkaren Murata Manufacturing som tillsammans utgör en avståndssensor. MA40S4S skickar en ultraljudsvåg mot marken och MA40S4R känner av när denna ultraljudsvåg reflekterats tillbaka mot marken och således kan tidsskillnaden mellan skickad och mottagen signal användas för att beräkna quadcopterns aktuella höjd. För att quadcoptern ska kunna röra sig fritt måste även trådlös kommunikation upprättas mellan användare och system. Detta skulle kunna realiseras

med en fjärrkontroll som sänder instruktioner till quadcoptern via radiovågor eller via bluetooth. En anslutning via GPRS, 3G, 4G eller WiFi är också ett alternativ som skulle göra det möjligt att styra quadcoptern via en smartphone eller dator.

Referenser

[1] P. Whytock, "ARM Strengthens Grip On Phone Processor Market," *Electronic Design,* Augusti 2013. [Online]. Tillgänglig: http://electronicdesign.com/digital-ics/arm-strengthens-grip-phone-processor-market [Hämtad: 3 juni, 2014].

[2] L. Ljung och T. Glad, *Modellbygge och simulering*, andra upplagan, Lund: Studentlitteratur, 2004.

[3] V. Sikiric, "Control of Quadrocopter," KTH Royal Institute of Technology, Stockholm, 2008.

[4] The Mathworks, Inc., ”System Identification Toolbox,” *The Mathworks, Inc*, [Online]. Tillgänglig: http://www.mathworks.se/products/sysid/description4.html. [Hämtad: 3 juni, 2014].

[5] Morar, I och Nascu, I, "Model simplification of an unmanned aerial vehicle, i *Automation Quality and Testing Robotics (AQTR), 2012 IEEE International Conference on* , vol., no., pp.591,596, 24-27 May 2012. [Online]. Tillgänglig: IEEE Xplore, doi: 10.1109/AQTR.2012.6237779 [Hämtad: 3 juni, 2014].

[6] A.Trusov, ”Overview of MEMS Gyroskopes: History, Principles of Operations, Types of Measurements,” University of California, Irvine, USA, maj 2011. [Online]. Tillgänglig: http://www.alexandertrusov.com/uploads/pdf/2011-UCI-trusov-whitepaper-gyros.pdf [Hämtad: 13 juni, 2014].

[7] L.Bengtsson, *Elektriska mätsystem och mätmetoder*, Lund: Studentlitteratur, 2012.

[8] STMicroelectronics, ” AN3182 Application note: Tilt measurement using a low-*g* 3-axis accelerometer,” STMicroelectronics, April 2012. [Online]. Tillgänglig: http://www.st.com/web/en/resource/technical/document/application_note/CD00268887.pdf [Hämtad: 3 juni, 2014].

[9] B. Thomas, *Modern Reglerteknik*, fjärde upplagan, Stockholm: Liber AB, 2008.

[10] STMicroelectronics, "STM32F4 Discovery", *STMicroelectronics*, 2014. [Online]. Tillgänglig: http://www.st.com/web/catalog/tools/FM116/SC959/SS1532/PF252419 [Hämtad: 3 juni, 2014].

[11] STMicroelectronics, ”L3G4200D: MEMS motion sensor ultra-stable three-axis digital output gyroscope,” STMicroelectronics, Dec 2010. [Online]. Tillgänglig: http://www.st.com/st-web-ui/static/active/en/resource/technical/document/datasheet/CD00265057.pdf [Hämtad: 3 juni, 2014].

[12] STMicroelectronics, ”LIS302DL: MEMS motion sensor 3-axis - ± 2g/± 8g smart digital output “piccolo” accelerometer,” STMicroelectronics, Okt 2008. [Online]. Tillgänglig: http://www.st.com/st-web-ui/static/active/en/resource/technical/document/datasheet/CD00135460.pdf [Hämtad: 4 juni, 2014].

[13] HobbyKing, “HobbyKing SS Series 25-30A ESC,” *HobbyKing.com*, [Online]. Tillgänglig: *http://hobbyking.com.au/hobbycity/store/uh_viewitem.asp?idproduct=6458* [Hämtad: 4 juni, 2014].

[14] Hobbylord, “ST2210 Brushless Motor,” *Hobbylord Multi-Rotors Aircraft,* [Online]. Tillgänglig: http://www.hobbylord.com/en/content/?240.html [Hämtad: 4 juni, 2014].

[15] CooCox, “CooCox CoIDE,” *CooCox Free/Open ARM Cortex MCU Development Tools*,[Online]. Tillgänglig: http://www.coocox.org/CooCox_CoIDE.htm [Hämtad: 13 juni, 2014].

[16] Jose C,et al.,” Stability control of a Quad-Rotor Using a PID Controller,” i *Brazilian Journal of Instrumentation and Control.”* [Online]. Tillgänglig: http://revistas.utfpr.edu.br/pg/index.php/bjic/article/download/1656/1093. [Hämtad: 3 juni, 2014].

[17] S Colton, ”The Balance Filter: A Simple Solution for Integrating Accelerometer and Gyroscope Measurements for a Balancing Platform,” Juni 2007, [Online]. Tillgänglig: https://googledrive.com/host/0B0ZbiLZrqVa6Y2d3UjFVWDhNZms/filter.pdf [Hämtad: 15 juni, 2014].

[18] S Anthony, ”Amazon unveils 30-minute Prime Air quadcopter delivery service, but it’s completely impractical,” *ExtremeTech*, Dec 2013. [Online]. Tillgänglig: http://www.extremetech.com/extreme/171879-amazon-unveils-30-minute-prime-air-quadcopter-delivery-service-but-its-completely-impractical [Hämtad: 3 juni, 2014].

[19] T Buchberger, ”Kalman Filtering,” *quadcopter-NextGen*, [Online]. Tillgänglig: http://quad-nextgen.hagenberg.servus.at/content/kalman-filtering [Hämtad: 3 juni, 2014].

Appendix

Appendix A - Beräkningar

Överföringsfunktion (4.11) från avsnitt 4.5.2:

$$G_r(s) = \frac{3.46s^4 + 9.019s^3 + 75.56s^2 + 163.3s + 1.94}{s^4 + 21.6s^3 + 152.3s^2 + 2.594\,s}$$

Diskretisering i Matlab ger följande där U är utsignal och E är insignal.

$$G_r(z) = \frac{2.889z^4 - 11.39z^3 + 16.85z^2 - 11.09z + 2.743}{z^4 - 3.6z^3 + 4.848z^2 - 2.898z + 0.6491} = \frac{U}{E}$$

Omskrivning till negativ representation ger:

$$G_r(z) = \frac{2.889 - 11.39z^{-1} + 16.85z^{-2} - 11.09z^{-3} + 2.743z^{-4}}{1 - 3.6z^{-1} + 4.848z^{-2} - 2.898z^{-3} + 0.6491z^{-4}} = \frac{U}{E}$$

Omskrivning till differensekvation ger:

$$2.889e(k) - 11.39e(k-1) + 16.85e(k-2) - 11.09e(k-3) + 2.743e(k-4) = u(k) - 3.6u(k-1) + 4.848u(k-2) - 2.898u(k-3) + 0.6491u(k-4)$$

Slutligen kan den aktuella styrsignalen beskrivas enligt:

$$u(k) = 2.889e(k) - 11.39e(k-1) + 16.85e(k-2) - 11.09e(k-3) + 2.743e(k-4) + 3.6u(k-1) - 4.848u(k-2) + 2.898u(k-3) - 0.6491u(k-4) \qquad (4.12)$$

Appendix B - Programkod

Globala variabler

```
//Global variables
uint8_t start=0;

//Accelerometer
int8_t x_acc=0, y_acc=0, z_acc=0;
int8_t x_acc_offset=4;
float acc_pitch=0.0, acc_roll=0.0;

//Gyroskop
int16_t x_gyro=0;
int16_t y_gyro=0;
int16_t z_gyro=0;
int8_t gyro_offset=0;

float gyro_roll=0.0, gyro_pitch=0.0, gyro_z_angle=0.0;

// Real angles
double real_pitch=0.0, real_roll=0.0, test=0.0;
int8_t real_pitch_round;

//int8_t pitch_delta=0;
float pwm_bas_varde=40.0;                    //0-255
float m1=0.0,m2=0.0,m3=0.0,m4=0.0;           //Styrsignaler till motorer

// Regulatorer
int8_t set_pitch=0;
float pe=0, pe1=0, pe2=0, pe3=0, pe4=0, pu=0, pu1=0, pu2=0, pu3=0, pu4=0;

double p_term,i_term,d_term;
double error=0.0;
double styrsign=0.0;
PID regulator;


//Serial send/receive
uint8_t send_buffer[5];
uint8_t rec_buffer;
```

Main loop

```
int main(void)
{

	LED_init();
	button_init();
	Init_All();
	delay_nms(2000);
	GPIO_SetBits(GPIOD, GPIO_Pin_15); //Indikeringsled att initieringar är klara

	// Sätter PID parametrar
	regulator.windup_guard=30.0;
	regulator.proportional_gain=0.0033;
	regulator.integral_gain=0.33;
	regulator.derivative_gain=0.02;


    while (1){

	   // Väntar på att användaren trycker start
	   if (GPIO_ReadInputDataBit(GPIOA, GPIO_Pin_0) == 1) {

				m1=pwm_bas_varde;
				m3=pwm_bas_varde;

				PWM_SetDC(1,&m1);
				PWM_SetDC(4,&m3);

				delay_nms(1500);
				start=1;
			}

    }

} //End of main loop
```

Avbrottsrutin

```
void TIM3_IRQHandler()
{

        if (TIM_GetITStatus(TIM3, TIM_IT_Update) != RESET)
                  {
                     TIM_ClearITPendingBit(TIM3, TIM_IT_Update);
                     GPIO_ToggleBits(GPIOD, GPIO_Pin_13);   //Indikator LED

                     //Läs sensorer och omvandla datan till vinkel
                     Update_Acc();
                     Acc_Convert_Angle();
                     Update_Gyro();
                     Gyro_Convert_Angle();

      //Komplementfilter
      real_pitch=(0.98*(real_pitch+(gyro_pitch)*0.02))+(0.02*acc_pitch);

                     VCP_get_char(&rec_buffer);      //Hämta Börvärde från Simulink
                     Set_pitch=(int8_t)rec_buffer;


                     if(start==0){
                           regulator.prev_error=set_pitch-real_pitch;
                     }

                     error=set_pitch-real_pitch; //Uppdatera aktuellt fel

                     if(start==1){

                           pid_update(&regulator, error, 0.02);

                    //Addera och subtrahera styrsignalen till motstående motorer
                           m1=(pwm_bas_varde+styrsign);
                           m3=(pwm_bas_varde-styrsign);

                           // Uppdatera duty cycle till ESC
                           PWM_SetDC(1, &m1);
                           PWM_SetDC(4, &m3);
                     }

               }
}
```

Regulator 1

```
void discrete_reg(){

	extern uint8_t set_pitch;
	extern double real_pitch;

	//Gamla fel och styrsignaler
	extern float pu,pu1,pu2,pu3,pu4,pe,pe1,pe2,pe3,pe4;

	// Uppdatera aktuellt fel
	pe=set_pitch-real_pitch;

	// Uppdatera styrsignalen
	pu=(2.889*pe)-(11.39*pe1)+(16.85*pe2)-(11.09*pe3)+(2.743*pe4)+(3.6*pu1)-
(4.848*pu2)+(2.898*pu3)-(0.6491*pu4);

	// Spara gamla fel och styrsignaler
	pu4=pu3;
	pu3=pu2;
	pu2=pu1;
	pu1=pu;
	pe4=pe3;
	pe3=pe2;
	pe2=pe1;
	pe1=pe;
}
```

Regulator 2

```
typedef struct {
          double windup_guard;
          double proportional_gain;
          double integral_gain;
          double derivative_gain;
          double prev_error;
          double int_error;
          double control;
      } PID;

void pid_update(PID* pid, double curr_error, double dt) {
          double diff;
          extern double p_term;
          extern double i_term;
          extern double d_term;

          // integration med windup guard
          pid->int_error += (curr_error * dt);
          if (pid->int_error < -(pid->windup_guard)){
             GPIO_SetBits(GPIOD, GPIO_Pin_12);
              pid->int_error = -(pid->windup_guard);
             }
          else if (pid->int_error > pid->windup_guard){
             GPIO_SetBits(GPIOD, GPIO_Pin_12);
              pid->int_error = pid->windup_guard;
             }

          // Derivata
          diff = ((curr_error - pid->prev_error) / dt);

          // Multiplikation med parametrar
          p_term = (pid->proportional_gain * curr_error);
          i_term = (pid->integral_gain     * pid->int_error);
          d_term = (pid->derivative_gain   * diff);

          // Summering
          pid->control = p_term + i_term + d_term;
          styrsign=pid->control;

             // Spara gammalt fel
          pid->prev_error = curr_error;
}

void pid_zeroize(PID* pid) {
          // Nollställ PID
          pid->prev_error = 0;
          pid->int_error = 0;
      }
```

Funktioner rörande gyroskop

```
void Gyro_Config()
{
    I2C_write(L3G4200D_ADDR, CTRL_REG1, 0x0F);
    I2C_write(L3G4200D_ADDR, CTRL_REG2, 0x00);
    I2C_write(L3G4200D_ADDR, CTRL_REG3, 0x00);
    I2C_write(L3G4200D_ADDR, CTRL_REG4, 0x30);
    I2C_write(L3G4200D_ADDR, CTRL_REG5, 0x01);
}

void Update_Gyro(void){
      // Om gyroskopet har registrerat mätvärden
      if((I2C_readreg(L3G4200D_ADDR,STATUS_REG)&0x08)==0x08)
            {
                  extern int16_t x_gyro;
                  extern int16_t y_gyro;
                  extern int16_t z_gyro;
                  uint8_t a1,a2;

                  a1=I2C_readreg(L3G4200D_ADDR,OUT_X_L);
                  a2=I2C_readreg(L3G4200D_ADDR,OUT_X_H);
                  x_gyro=((a2<<8) | a1);

                  a1=I2C_readreg(L3G4200D_ADDR,OUT_Y_L);
                  a2=I2C_readreg(L3G4200D_ADDR,OUT_Y_H);
                  y_gyro=((a2<<8) | a1);

                  a1=I2C_readreg(L3G4200D_ADDR,OUT_Z_L);
                  a2=I2C_readreg(L3G4200D_ADDR,OUT_Z_H);
                  z_gyro=((a2<<8) | a1);
          }
}

void Gyro_Convert_Angle(void){

      extern int16_t x_gyro;
      extern int16_t y_gyro;
      extern int16_t z_gyro;
      extern float gyro_roll;
      extern float gyro_pitch;
      extern int8_t gyro_offset;

      // Gyrot ställt i 2000 dps -> 70 mdps/digit
      gyro_pitch=(y_gyro-gyro_offset)*0.07;
}
```

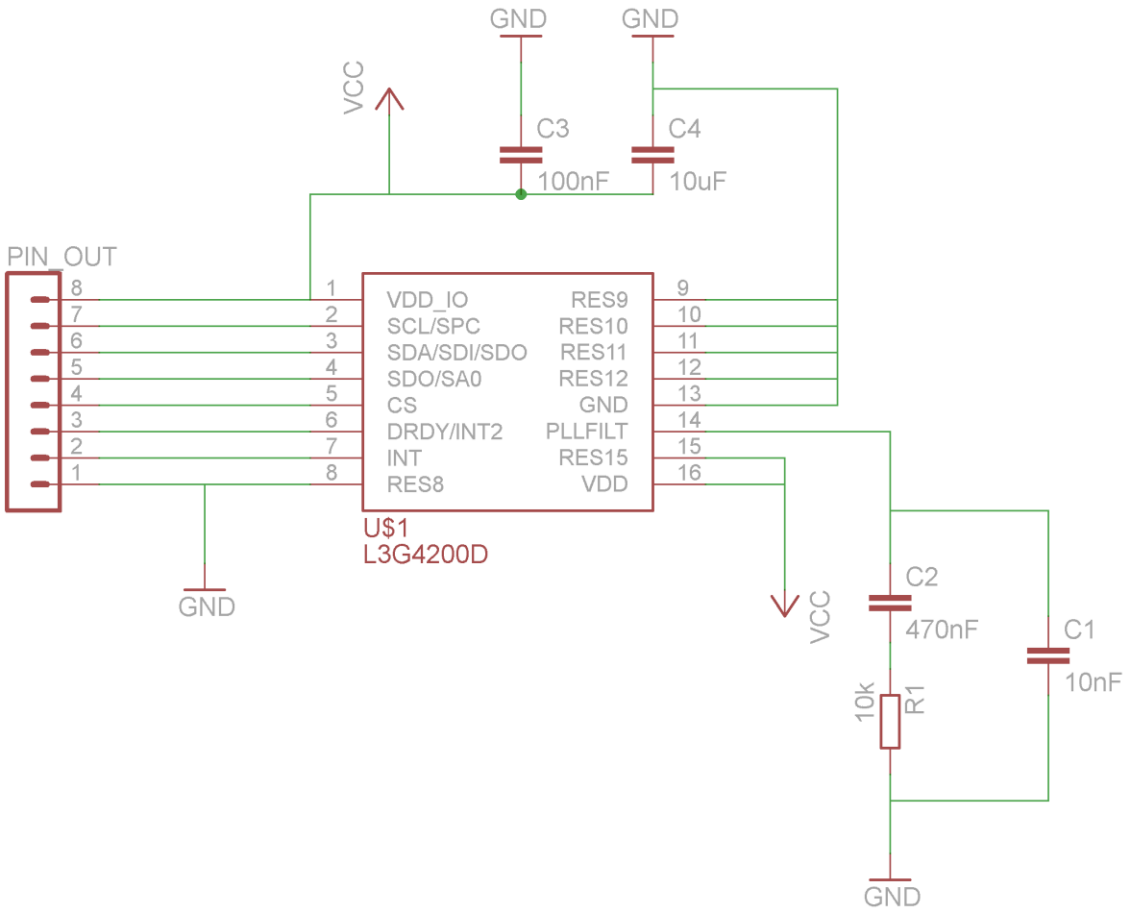
Funktioner rörande accelerometer

```
void Acc_Config(void){
	ACC_SPI_SendData(0x20, 0x40);
}

void Update_Acc(void){

	extern int x_acc,y_acc, z_acc;

	x_acc=ACC_SPI_GetData(0x29);			//Read x-value. Address 0x29
	y_acc=ACC_SPI_GetData(0x2B);			//Read y-value. Address 0x2B
	z_acc=ACC_SPI_GetData(0x2D);			//Read z-value. Address 0x2D

}

void Acc_Convert_Angle(void){

	extern int8_t x_acc, y_acc, z_acc;
	extern float acc_pitch, acc_roll;
	float temp,x,y,z;

	x=x_acc*0.018;
	y=y_acc*0.018;
	z=z_acc*0.018;

	temp=sqrt(pow(y, 2) + pow(z, 2));
	acc_pitch=atan2(x, temp)*(180/PI);

	temp=sqrt(pow(x, 2) + pow(z, 2));
	acc_roll=atan2(y, temp)*(180/PI);
}
```

Appendix C – Konstruktion av mönsterkort

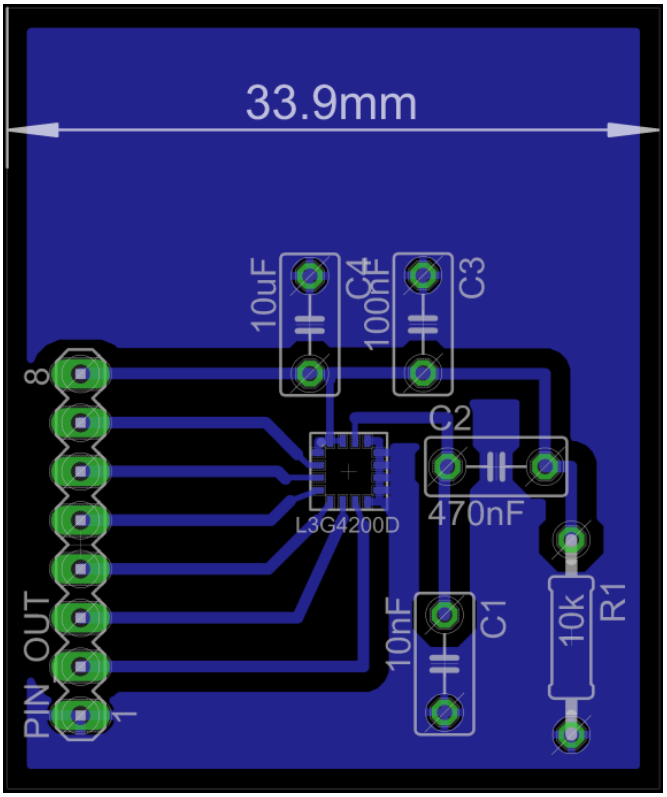
Specifikationerna som ställs på mönsterkortet är i princip bara den storleksbegränsning som finns med tanke på monteringen på farkostens ram. Av detta skäl bestämdes att mönsterkortets dimensioner får maximalt vara 60x60mm. Designarbetet av mönsterkortet utfördes i mjukvaran Cadsoft Eagle Light Edition.

Inledningsvis ritades ett kretsschema upp i mjukvaran. För att gyroskopet ska fungera som planerat krävs diverse kringliggande komponenter så som avkopplingskondensatorer, filter m.m. I databladet för L3G4200D föreslår tillverkaren kopplingsscheman för filter och värden på kondensatorer. Av praktiska skäl användes hålmonterade komponenter till detta. I figur C.1 nedan är kretsschemat för hela mönsterkortet med gyroskopet L3G4200D och övriga komponenter inritade. Komponenten vars namn PIN_OUT är en stiftlist med standardmått för att underlätta kopplingen mellan mönsterkortet och STM32F4-kortet.



Figur C.1 Kretsschema för gyroskopet med tillhörande komponenter.

Mjukvaran låter därefter användaren designa layouten av det fysiska mönsterkortet via ett CAD (Computer-Aided Design) gränssnitt. Detta görs utifrån de kopplingar som är definierade i kretsschemat i figur 4.1. I denna fas får användaren bestämma de fysikaliska specifikationerna till mönsterkortet så som kortets storlek och placering av komponenter och ledare. Följaktligen måste hänsyn tas till parametrar så som strömstyrka, isolationsavstånd och jordplan m.m. I figur C.2 nedan visas resultatet av layoutarbetet. I figuren syns undersidan av mönsterkortet med dess kopparbanor, jordplan, lödpaddar, och borrhål för hålmonterade komponenter. I figuren syns även ett lager som beskriver vissa av komponenternas placering, namn och värden samt dimensioner till mönsterkortet. Detta lager kommer inte att synas vid framställningen av mönsterkortet utan utgör endast ett visuellt stöd för läsaren.



Figur C.2 Layout för mönsterkortet. Blått lager motsvarar kopparledningar och kopparplan.

Som tidigare nämnts måste användaren fastställa vissa parametrar som kan påverka mönsterkortets funktion så som val av ledningsbredd, isolationsbredd och placering av komponenter. Initialt framgick det att storlekskravet på maximalt 60x60mm ej medförde några begränsningar till dessa parametrar. Av detta skäl kunde de väljas relativt fritt och med goda marginaler för att säkerställa önskad funktion. Följaktligen placerades avkopplingskondensatorerna C4 och C3 så nära gyroskopet som möjligt för att minimera eventuella störningar. Vidare valdes lednings- och isolationsbredd främst med hänsyn till att produktionen av det fysiska mönsterkortet försvåras då dessa är för små. Därför

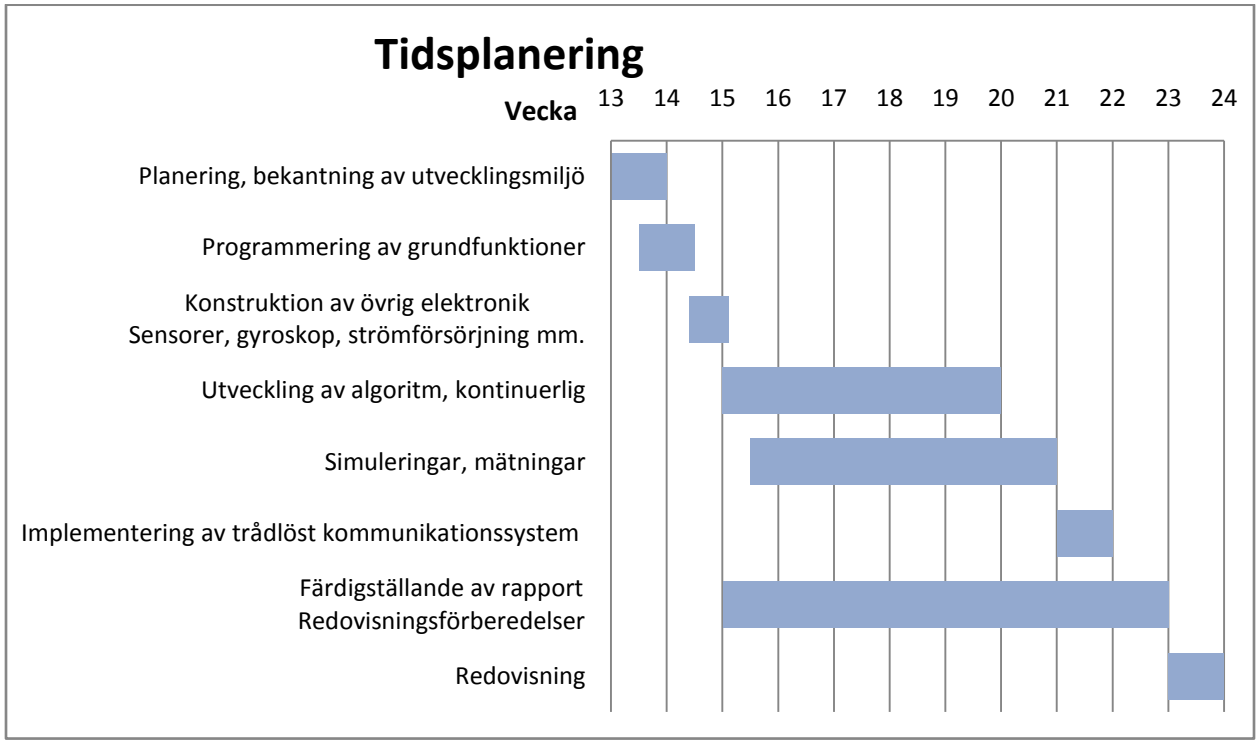
valdes väl tilltagna storlekar på respektive parameter. De strömstyrkor och spänningar som flyter i kretsen är små vilket medför att man kan bortse från deras inverkan på val av lednings- samt isolationsbredd i detta fall.

Avslutningsvis framställs det fysiska mönsterkortet utifrån de definitioner som användaren bestämt i samband med layoutarbetet. Layouten i figur C.2 skrivs ut på OH-blad varefter det belyses med UV-ljus på ett kretskortslaminat med fotoresist. Mönsterkortet utsätts därefter för kemiska processer där koppar etsas bort på de ställen som layouten definierat. Då gyroskopet utgörs av en ytmonterad komponent vars avstånd mellan benen är liten, ca 0.3mm, fick denna ytmonteras med hjälp av speciell utrustning. Övriga komponenter hålmonterades med handlödning. Ett första mätningstest utfördes då konstruktionen var klar för att säkerställa att inga kortslutningsfel mellan ledningar och komponenter inträffat.

Matningsspänningen, Vcc, till gyroskopet är 3V och detta togs direkt från STM32F4-kortet då denna har en inbyggd utgång med 3V för extern matning. För mjukvarukonfiguration av gyroskopet hänvisas läsaren till avsnitt 4.3.1 - Konfiguration av gyroskop.

Appendix D - Tidplan

I figur D-1 nedan presenteras den preliminära tidplan som fastslogs vid projektets start. Där följen även kommentarer kring hur den slutgiltiga tiden disponerades i projektet.



Figur D-1 Planerad tidplan för projektet.

Inledningsvis i projektet följdes tidplanen i figur D-1 relativt väl. De första tre kategorierna tog dock längre tid än väntat och pågick så långt som fram till vecka 18-19. Detta på grund av det omfattande arbetet som krävdes för att bekanta sig med processorn, sensorerna och utvecklingsmiljön. På grund av att större delen av hårdvaran i projektet var en ny erfarenhet för oss som utvecklare var det svårt att förutse hur lång tid denna fas i projektet skulle pågå. Dock anser vi att det extra arbete som krävdes var väl investerad tid med tanke på de lärdomar som det förde med sig. En stor del av syftet med projektet kretsade dessutom kring de nya erfarenheter som den specifika hårdvaran medförde.

Skrivandet av rapporten blev något lidandes med tanke på det som nämns ovan. Intentionen var att inleda arbetet med rapporten i ett tidigt skede. Detta skedde till viss del, men försvårades på grund av det tidsödande arbete med hårdvaran. I denna period var det därför svårt att få en uppfattning om rapportens struktur och vilka delar som skulle ingå. Sammanfattningsvis uppnådde dock projektet den sluttid som planerades med en fullständig rapport med tillhörande presentation.