

Programmering av en självreglerande 2DOF plattform för marina tillämpningar

Programming the Control of a 2DOF Platform for Marine Applications

Examensarbete inom högskoleingenjörsprogrammet Mekanik

ALFRED FRIMAN
EDUARD KAPOUN

Institutionen för signaler och system
Avdelningen för reglerteknik, automation och mekatronik
CHALMERS TEKNISKA HÖGSKOLA
Göteborg, Sverige, 2012

Förord

Denna rapport är en del av vårt examensarbete vid Chalmers tekniska högskola inom Mekanikingenjörsprogrammet (180hp) för uppdragsgivaren Satcube AB. Arbetet löpte över sommaren 2012 med början i mars och avslut i augusti och motsvarar 15 högskolepoäng.

Handledare på Satcube AB var VD Jakob Kallmér. Handledare på institutionen Signaler & System på Chalmers var Göran Hult.

Vi skulle vilja ge ett stort tack till Jakob Kallmér och andra affilierade till projektet som ej nämns, för att främst ha givit oss förtroendet att genomföra ett projekt av detta slag men som också låtit oss ta del av denna otroligt lärorika resa med resulterande erfarenhet. Tack Jakob för att Du har stött oss genom projektets gång och varit en utomordentlig handledare samt peppat oss då projektet tidvis visat sig frustrerande.

Vi vill även ge ett stort tack till Lars Olsson på Satcube AB för hans programmerings- och hårdvarutekniska råd och expertis, utan dem hade vi förmodligen inte kommit lika långt i både projektet och våra slutsatser. Vi uppskattar den tid Du tagit att svara på mejl och telefonsamtal när vi kört fast samt den erfarenhet du haft att dela med dig gällande programmering på hårdvarunivå.

Ännu ett stort tack skulle vi vilja ge P.O. Bäckman på Satcube AB som tillsammans med Lars Olsson och Jakob Kallmér gav oss lite mer insikt i hur företaget fungerar utanför kontoret samt det parallella arbetet som utfördes på verkstaden i Karlstad.

Sist skulle vi vilja ge ett stort tack till vår handledare på Chalmers, Göran Hult, för att ha varit en så enastående och pedagogisk lärare i föregående kurser att vi klarade en stor del av projektets tekniska aspekter utan större hjälp. Trots detta skulle vi ändå vilja tacka för de konsultationstiderna som varit och den insikt Du givit oss inom elektronik och programmering genom din egen erfarenhet. Vi skulle även vilja säga tack för den viktiga utrustning vi fått låna av Chalmers via Dig då den hjälpt oss en stor del under arbetsgången.

Tack så mycket!

*Alfred Friman
Eduard Kapoun*

Sammanfattning

Navigering och kommunikation till sjöss kräver stabila, precisa och oftast dyra komponenter. Ett system av komponenter består oftast av en mekaniskt rörlig mottagare för specifik trådlös kommunikation. Sådana mottagare är monterade på en plattform som verkställer rörelser i flera axlar för att behålla pekriktning och därmed en stabil kommunikationslänk. I detta arbete utreds en plattform som möjliggör precisionspekning för en antenn mot en satellit i den geostationära banan.

Konstruktioner av detta slag varierar väldigt mycket i storlek och prestanda och är oftast utvecklade efter en viss typ av fartygsstorlek eller driftsförhållande. Satcube AB vill därför studera möjligheten att utveckla en plattform, oberoende av målfartyg, med hjälp av betydligt billigare elektronisk hårdvara än vad som vanligtvis används i liknande konstruktioner. Detta för att minimera kostnad men ändå försöka behålla en tillfredställande prestanda jämte konkurrenter med en pekprecision på $0,1^\circ$. Pekprecisionen åstadkoms med hjälp av programvara vilket tas fram under arbetets gång samt en inklinometer vilken är positionerad på plattformen och mäter dess lutning.

Av studien har det framgått att det inte är möjligt att utveckla en produkt med önskad prestanda och pekprecision jämte konkurrenter med inklinometern som avsetts för arbetet. Detta resultat hämmar en fullskalig produktion då främst prestanda på vinkelmätning är undermålig för ändamålet och gör därför produkten i sin helhet obrukbar i kommersiella sammanhang.

Resultatet har nåtts med en del avgränsningar av bland annat vindlaster, vibrationer och bristen på tillgång till en realistisk sjösimulator. Studien har därför begränsats till tester och experiment vilka inte speglar verkligheten på ett realistiskt vis.

Trots avgränsningarna som gjorts och slutsatserna som dragits har det också utförts en sidostudie på möjligheterna att inkludera bland annat ett gyroskop i konstruktionen som riktlinjer för framtida förbättringar.

Abstract

Navigation as well as communication at sea requires stable, precise, and usually expensive components. Systems of several components mainly consist of an ambulating receiver for specific airborne communication. Such receivers are mounted on platform-mechanisms which execute the necessary multi-axis movements to maintain the correct pointing direction and thus a stable communication link. This study investigates a platform-mechanism that enables precision pointing of a satellite antenna to a satellite receiver situated in the Earth's geostationary orbit.

Mechanical constructions of this type vary greatly in size and performance and are usually developed for a particular type of vessel size or marine conditions. Satcube AB would therefore like to evaluate the feasibility of developing a mechanical platform, targeting a broad range of vessels, using significantly inexpensive electronic hardware than is commonly used in existing solutions. This is to minimize cost while catering to a broader vessel range and maintaining a satisfactory pointing precision of 0.1° . The movement of the platform is accomplished partly by means of software, produced as the study goes along, interfacing with a tilt meter measuring the platforms angular incline.

The study has shown that it is not possible to develop a product with the target performance and pointing precision while taking into consideration requirements set by competing solutions in the industry. This inhibits a full-scale production since the performance of the tilt meter proved inadequate for this purpose and therefore renders the product unusable for commercial use.

Although the result had been reached with restraints to the scope of the project, they did not prove to mitigate the limiting factors of the tilt meter. Constraints included wind loads, vibrations generated by the ship hull and a lack of access to a sea-state simulator. Therefore, the study had been through testing and experimentation that did not correctly reflect reality in a realistic and scientific way.

Despite the boundaries made and conclusions drawn, a smaller investigation on the possibilities to include a gyroscope to accompany the tilt meter was conducted in hopes of laying the foundation for future improvements in the responsiveness and pointing precision of the system.

INNEHÅLLSFÖRTECKNING

Beteckningar	1
1 INLEDNING.....	2
1.1 Bakgrund	2
1.2 Syfte.....	2
1.3 Avgränsningar.....	2
1.4 Precisering av arbetsuppgiften	2
2 TEKNISK BAKGRUND	3
2.1 Positionssystem.....	3
2.2 Hårdvara.....	4
2.2.1 Olimex PINGUINO-OTG med PIC32 Processor	4
2.2.2 Lägesgivare.....	5
2.2.3 Inklinometer.....	5
2.2.4 Stegmotor.....	6
3 METODIK	8
4 FÖRSTUDIE.....	9
4.1 Havets rörelser	9
4.2 Rörelsemönster av farkoster till sjöss	9
4.3 Marknadsundersökning av aktuella lösningar	11
4.4 Fastställande av lämpligt fartyg för arbetet.....	12
4.5 Slutgiltiga kravbilden.....	13
5 UTFÖRANDE.....	15
5.1 Principiella idén.....	15
5.2 Begränsningar i mjukvaran.....	15
5.3 Kommunikation inom systemet	18
5.4 Plattformens mekanik	22
5.5 Proqrambeskrivning och tester med en sensor	24
5.5.1 Informationskortet och motorkortens programflöden.....	24
5.5.2 Bestämning av pulsfrekvens, hastighet och dess samband	24
5.5.3 Framtagande av regulatoralgoritm	26
5.5.4 Test av systemets prestanda	28
5.6 Tankar och tester med en extra sensor	29
6 SLUTSATSER OCH RESULTAT	31
6.1 Inklinometern.....	31

6.2	Stegmotorerna	32
6.3	Utvecklingskorten	33
6.4	Programkod.....	33
6.5	Ekonomiska ramar.....	33
6.6	Möjliga förbättringar.....	33
6.7	Genomförbarhet	35
7	KÄLLFÖRTECKNING.....	36
7.1	Figurer	36
7.2	Text.....	37
7.3	Personreferenser.....	37

Beteckningar

DOF	Degree of Freedom
ESV	Earth Station on Vessel
PIC	Peripheral Interface Controller
GPS	Global Positioning System
LCD	Liquid Crystal Display
LAN	Local Area Network
IMU	Inertial Measurement Unit
OTG	On-The-Go
USB	Universal Serial Bus
ICSP	In Circuit Serial Programming
UEXT	Universal EXTension
LiPo	Lithium Polymer
LED	Light Emitting Diode
UART	Universal Asynchronous Receiver/Transmitter
TTL	Transistor-Transistor Logic
ANSI	American National Standards Institute
CDC	Communications Device Class
COM	COMmunications port
RS-232	Recommended Standard 232 (Seriell kommunikation)
I²C	Inter-Integrated Circuit
SPI	Serial Peripheral Interface bus
IDE	Integrated Development Environment
VSAT	Very Small Antenna Terminal
SOTM	SATCOM On The Move
COTM	Communications On The Move

1 INLEDNING

1.1 Bakgrund

Satcube AB är ett ingenjörsföretag som utvecklar innovativa satellitkommunikationslösningar. Företaget bildades 2006 och bygger ursprungligen på en konsultbaserad inriktning med rötter i svenska satcom-industrin, däribland SWE-DISH Satellite Systems och SAAB Ericsson Space.

Satcube AB siktar på att utvecklas från ett rent konsultföretag till att ta sig in på marknaden med en egenutvecklad produktfamilj. Flaggskeppet i produktlinjen blir en ESV-terminal som konkurrerar i både prestanda och pris med nuvarande tillgängliga system.

1.2 Syfte

Huvudsyftet med arbetet är att ta fram programmeringskod till den delen av ESV-terminalen som består av en 2-DOF plattform med uppgift att reglera sig parallellt med horisonten i båda axlar. Detta ska möjliggöra precisionspekning mot en satellit av ett fartyg med en påmonterad VSAT i anslutning till en specifik kravbild anpassad för en specifik pekprecision samt olika sjögångar och fartyg.

1.3 Avgränsningar

På grund av den begränsade tiden för arbetets utförande bortsågs påverkningar från bland annat vindlaster och vibrationer på plattformen. Då det inte var möjligt att tillgå en sjösimulator fick naturliga havsrörelser avgränsas likaså, som istället simulerades i ett labb med mekaniska domkrafter.

1.4 Precisering av arbetsuppgiften

Utöver programmeringen ska även systemets komponenter, som är inklinometer, utvecklingskort, induktiva ändlägesgivare samt stegmotorer och dess drivkretsar kopplas samman på ett fungerande sätt. Det ingår också praktiska labbmoment och experimentering med framtagen data i diverse simulerings- och beräkningsprogram.

Frågor som besvaras framöver:

- Är en inklinometer tillräcklig för uppgiften?
- Är stegmotorer tillräckliga för uppgiften?
- Är det möjligt att uppnå en pekprecision på $0,1^\circ$?
- Är resultatet inom rimliga ekonomiska ramar?
- Vad skulle kunna förbättras?
- Är arbetet genomförbart?

2 TEKNISK BAKGRUND

I detta avsnitt behandlas teoretiska motiveringar och förklaringar till den praktiska delen av arbetet, samt förklaringar till de komponenter som använts och till viss del hur de fungerar. För en illustration över plattformens utseende, dess delar och var följande komponenter i detta kapitel sitter monterade på plattformen se Bilaga D. s.D1.

2.1 Positionssystem

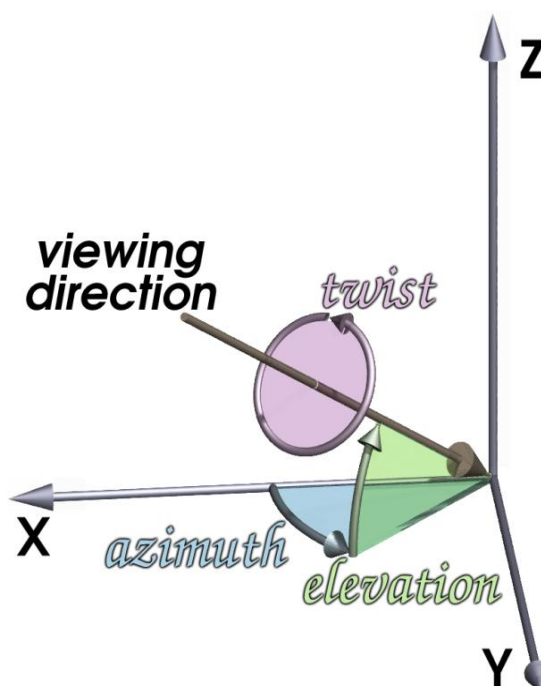
En Earth Station on Vessel förkortat ESV består av en VSAT (Very Small Antenna System) monterad på ett flertal rörliga axlar som tillsammans arbetar för att upprätthålla en stabil signal mellan antensystem och satellit. ESV-terminaler används mest på fartyg eller större båtar men kan också monteras på andra farkoster i rörelse.

Eftersom systemen används på farkoster måste antensystemet till skillnad från vanliga fasta TV satellitmottagare ställa in sig kontinuerligt allteftersom farkosten rör sig i förhållande till satelliten den pekar mot. Utöver själva positionsförändringen hos till exempel ett fartyg måste också faktorer motverkas av ESV-terminalen såsom vågor, vindlaster och vibrationer där vågor är den största faktorn att ta hänsyn till.

Inom positionssystem och andra tillämpningar gällande pekprecision används det olika lösningar för att dynamiskt korrigera system från yttre störningar och upprätthålla precisionen som söks. En del av dessa anordningar kan bestå av flera motordrivna axlar som tillsammans korrigerar ett fel. Axlarna benämns azimuth, elevation samt polarisation (twist).

I detta fall utreds en anordning bestående av en tvåaxlig plattform vars uppgift är att ta hand om elevationens påverkan genom att reglera båda motoraxlarnas delplattformar mot horisontalplanet. Detta säkerställer på så sätt att precisionen av en påmonterad VSAT mot en viss punkt upprätthålls oberoende av vilken lutning plattformen utsätts för. För att plattformen ska kunna fungera som ett reglerande system används kringutrustning som inklinometrar, induktiva givare, utvecklingskort med processor av modell PIC32 samt stegmotorer med tillhörande drivkretsar.

I systemet finns tre utvecklingskort som alla har var sin uppgift. Det första utvecklingskortet tar emot data från inklinometern vilket består av momentanvinkeln för plattformen. Dessa data behandlas och skickas sedan vidare till de två andra utvecklingskorterna som styr respektive axels motor. I det andra och tredje utvecklingskortet behandlas det mottagna vinkeldata till motsvarande styrsignal som stegmotorn ska ta för att korrigera vinkelförändringen av respektive axel.



Figur 2.1. Avbildning av axlarna azimuth, elevation och polarisation (twist), samt deras rörelsebanor [1]

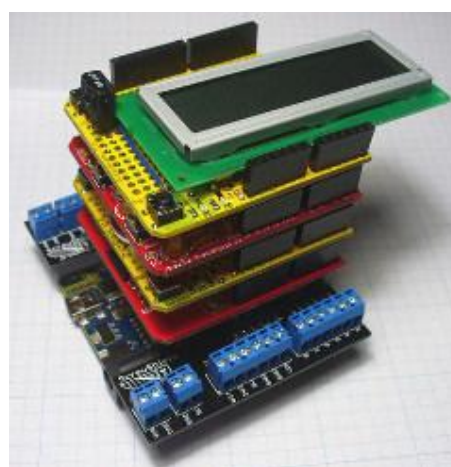
2.2 Hårdvara

2.2.1 Olimex PINGUINO-OTG med PIC32 Processor

Pinguino-projektet påbörjades 2008 av en grupp frivilliga mikrocontrollerentusiaster som ville sprida fröet Arduino hade satt inom hobbyelektronik genom att förmedla glädjen av att skapa egna mekatroniska projekt enkelt och billigt i sitt hem.

Idag växer Arduino-plattformen otroligt fort i takt med att fler hobbyister och nybörjare fattar intresse för denna väldigt enkla plattform för projekt som annars kräver en hel del insyn och kunskap för att programmera, vilket naturligtvis också gäller Pinguino för de hobbyister som kräver mer kraftfull hårdvara. Pinguino-projektet olik Arduino är baserad på antingen en 8-bit eller 32 bitars PIC processor medan Arduino begränsar sig till en 8-bitars ATmel processor. Skillnaderna i prestanda är stora då det i Pinguino fordras mer programminne, RAM-minne samt en betydligt snabbare klockfrekvens än motsvarigheterna i Arduino.

Pinguino har ännu inte fått helt fungerande stöd för så kallade "shields" som finns för Arduino, men planerna enligt utvecklarerna är att det i slutändan ska vara fullständigt kompatibelt. "Shields" är externa kretskort med egna komponenter som går att köpa separat relativt billigt.



Figur 2.2. Flera shields modulärt kopplade till varandra, [2]

Några exempel på komponenter som finns att tillgå på diverse shields är GPS, WiFi, LAN, Radioantennor, diverse IMU sensorer och givare, LCD-Displayer, knappar och en mängd fler som inte nämns här. De shields som går att köpa kopplas enkelt till baskortet på ett modulärt vis med tillhörande stiftkontakter som passar i baskortets stiftgångar. Utöver detta kan flera shields kombineras samtidigt vilket markant ökar valfriheten och möjligheterna till vilka projekt som kan skapas.

Utvecklingskortet är en kommersiellt tillverkad "Pinguino-OTG" från Olimex LTD. som drivs av en PIC32MX440F256H processor. Den ursprungligen tänkta användningen av kortet är i samband med en gratis men något begränsad programmeringsmiljö kallad Pinguino. Olimex utvecklingskort i samband med den valda processorn tillåter en mängd olika periferienheter att kopplas in, bland annat genom USB, en SD/MMC plats, miniICSP för programmering, strömkontakter för både LiPo batterier och en extern strömkälla samt en universell UEXT-kontakt.

Utöver dessa kontakter finns det en mängd digitala samt analoga pinnar att tillgå som är enkla att komma åt och mäta på. Inbyggt finns också en reset-knapp, en användardefinierad knapp samt två programmerbara LED-lampor. Ett utvecklingskort liknande den som används i denna utredning går att bygga på egen hand med ritningar som finns att tillgå gratis då Pinguino plattformen likt Arduino som den härstammar från är Open Source.

Processorn har en klockfrekvens på 80MHz, en intern oscillator på 8MHz och därtill programminne på 256KB och 32KB RAM vilket ter sig tillräckligt för denna studie. Det finns fem timers att tillgå, möjligheten för upp till 96 avbrottskällor som alla kan ha varierande prioritet samt två UART-kanaler på TTL-nivå.

Utöver detta klarar utvecklingskortet av en spänningsmatning på allt mellan +9V till +30V och temperaturer mellan -25°C till +85°C vilket lämpar sig särskilt bra för industriella applikationer av varierande storlek och miljöer. Däremot klarar de flesta av ingångspinnarna endast av att ta emot spänningar på högst +3,3V från periferienheter då högre spänningar riskerar skada processorn, men ett antal högspänningsingångar på +5V finns att tillgå.



Figur 2.3. Olimex Pinguino-OTG, [3]

Eftersom Pinguino plattformen fortfarande till stor del är under uppbyggnad används utvecklingskortet istället som ett labbkort med MikroElektronika MikroC Pro for PIC32 v2.1 som utvecklingsmiljö körande programmeringsspråket C.

2.2.2 Lägesgivare

För att säkerställa att plattformen inte rör sig utanför sina fysiskt möjliga gränser finns lägesgivare installerade som i samband med programvara ser till att plattformen stannar när den når motsvarande lägesgivare. Lägesgivarna är induktiva vilket innebär att de reagerar på metalliska föremål som samtidigt är magnetiska.



Figur 2.4. Contrinex DW-AD-603-M8 lägesgivare, [4]

Då ett sådant föremål befinner sig högst 1,5mm ifrån givaren skickar den ifrån sig en positiv spänning som signalerar att föremålet är för nära. Finns inget föremål i närheten visar inte givaren något utslag. Givaren kan drivas med alla spänningar mellan +9V till +30V och ger en utsignalsspänning på samma nivå som inspänningen. Fördelen med en induktiv givare är att det inte finns några mekaniska delar som måste underhållas, bytas ut eller som kan gå sönder.

2.2.3 Inclinometer

En viktig förutsättning för att systemet ska kunna fungera är att man kan mäta lutningen på plattformen. Detta görs med hjälp av en inclinometer som fästs centralt på den inre axeln av plattformen och mäter på så sätt vinkeln av hela konstruktionen. Inclinometern har förmågan att mäta på två horisontella axlar samtidigt (X och Y), med möjligheten att med ett externt program ta emot kommandon för att välja utmatning av antingen X, Y eller både X och Y värden som mätts upp.

Det finns även möjlighet att skicka kommandon som bestämmer ifall den kontinuerligt ska mata ut uppmätta vinkelvärden och vilken baud-rate den ska arbeta i beroende på vilken utmatningshastighet som önskas för processen. Uppdateringsfrekvensen av inclinometern är däremot 20Hz och går inte att ändra.

Inklinometern fungerar med hjälp av en kapacitiv mikro-pendel som med jordens gravitation ger olika stora utslag på pendeln och därmed spänning vid vinkelförändringar. Tillsammans med fem inbyggda filter och integrerad programvara omvandlas denna spänning till ett vinkelvärde som matas ut genom RS232-protokollet i form av hexadecimala tal med tillhörande kontrollbitar och kommandoord. Eftersom inklinometern fungerar med en mekanisk mikro-pendel påverkas dess mätningar av stötar och horisontella translationsrörelser.

Tabell 2.1. Utvalda tekniska data för SCA126T-30, [5]

Parametrar	SCA126T-30
Mätområde	$\pm 30^\circ$
Mätaxel	X och Y
Temperaturdrift vid $-25^\circ\text{C} \sim 85^\circ\text{C}$	$\pm 0,008^\circ/\text{C}$
Responströskel	$\pm 0,5\%$
Uppdateringsfrekvens	20Hz
Upplösning vid 5Hz	$0,01^\circ$
Noggrannhet	$0,03^\circ$
Känslighet (maximalt)	3%
Stötsäkring	20000g på 0,5ms



Figur 2.5. RION Technology Co. Ltd. SCA126T-30, [5]

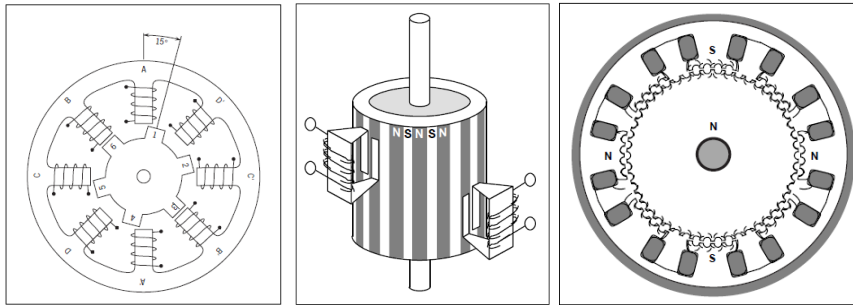
2.2.4 Stegmotor

En stegmotor är en elektromekanisk motor som fått sitt namn från att den tar in elektriska pulser som får dess axel att rotera i diskreta steg, en puls är alltså lika med ett steg. Hastigheten regleras således genom att ändra frekvensen på pulserna. [15]

Det finns tre olika typer av stegmotorer:

1. Variabel reluktansmotor är den enklaste modellen och består av en tandad rotor i järn som påverkas av ett magnetfält skapat av spolar i statorn. Rörelse uppstår beroende på vilka spolar som är strömförande och rotorn kommer då ställa in sig så att reluktansen som är det magnetiska motståndet blir så liten som möjligt. På grund av rotorns grova stegindelning blir ett steg oftast tiotals grader vilket ger dålig upplösning och precision.
2. Permanentmagnetiserad stegmotor har till skillnad från reluktansmotor en helt cylindrisk rotor som är permanentmagnetiserad med skiftande nord och syd poler parallellt med motorns axel. Precis som innan används spolar för att få rotorn att röra sig. På grund av den ökade magnetsiska flödet som skapas med permanentmagneten så kommer denna typ av stegmotor att ha ett högre vridmoment än reluktansmotorn. Typiska stegvinklar varierar mellan $7,5^\circ$ till 15° .

3. Hybrid stegmotorn är en kombination av reluktansmotor och permanentmagnetiserad stegmotor och har utöver spolarna i statorn en tandad rotor som även är permanentmagnetiserad med skiftande nord och syd poler. Detta ger större vridmoment och en finare stegupplösning från $0,9^\circ$ till $3,6^\circ$.



Figur 2.6. Från vänster; Variabel reluktansmotor, permanentmagnetmotor, hybridstegmotor [6]

Spolarna i stegmotorn lindas oftast unipolärt eller bipolärt. För att byta rotationsriktning på axeln delas den unipolära stegmotors lindningar upp i två lindningshalvor med motsatta lindningsriktningar för att kunna drivas av en spänning med endast en polaritet. Den bipolära stegmotorn är bara lindad åt ett håll och kräver då att spänningens polaritet ändras för att byta rotationsriktning. Unipolär lindning är betydligt enklare än bipolär men ger sämre moment eftersom bara en lindningshalva kan användas åt gången gentemot en bipolär som använder sig utav hela lindningen.

Fördelar:

- Motorn har fullt vridmoment vid statiskt tillstånd förutsatt att lindningarna är strömförande.
- Noggrann positionering eftersom bra stegmotorer har en träffsäkerhet på 3-5% på varje steg och att det felet inte ackumuleras.
- Utmärkt respons för start/stopp och riktningförändring.
- Lång livslängd.
- Kräver ingen återkoppling för att veta motorns position eftersom det räcker att hålla koll på antalet pulser som skickats till den, vilket blir både billigare och enklare.
- Många olika hastigheter kan antas eftersom hastigheten är proportionell mot frekvensen på inkommande pulser.

Nackdelar:

- Resonans kan uppstå vid olika hastigheter vilket medför tappat vridmoment och kan leda till missade steg och dålig synkronisering.
- Stegmotorer kan bli svåra att handskas med vid höga varvtal.

Stegmotorn som användes i projektet var en Nema34 av hybridtyp med steglängd på $1,8$ grader. Motorn kunde kopplas både bipolärt och unipolärt där det förstnämnda valdes för att få ut maximalt vridmoment, $8,5$ Nm gentemot $6,0$ Nm unipolärt. Till motorn användes drivkretsen CW6060AC som gav möjligheten att dela upp varje steg i så kallade mikrosteget, det vill säga att den ursprungliga steglängden på $1,8^\circ/\text{steg}$ kunde delas upp till maximalt 256 delar för att uppnå en maximal upplösning på $0,0039^\circ/\text{steg}$.

3 METODIK

Huvuddelen av arbetet utfördes på Satcubes kontor i Göteborg där det även fanns möjlighet till löpande diskussion med handledare. Kontinuerlig kontakt med kontoret i Karlstad genomfördes också både via mail och telefon för utbyte utav källkod, diverse information, få handledning och uppdatera samtliga inblandade om projektets framsteg.

Projektet inleddes med ett par veckors litteratursökning där en förståelse för havets rörelser och dess påverkan på diverse fartygs sjögång införskaffades. Under denna tid studerades även konkurrerande systems utförande och prestanda. En del av målen med förstudien var att få en förståelse för diverse fartygs rörelser i sjögång för att senare ta hjälp av denna kunskap i programutvecklingen. Ett annat delmål var även att väga samman den insamlade informationen om olika fartygs rörelser och konkurrerande system till en slutgiltig kravbild för produkten.

Under förstudien bestämdes och beställdes också utvecklingskort och nödvändiga komponenter som behövdes för arbete på plattformen. Efter denna del av förstudien påbörjades hopkoppling av direkt tillgängliga komponenter och laboration med enbart inklinometer och stegmotor påbörjades inför laborationerna på den fysiska plattformen.

När konstruktionen av plattformen blev klar skickades den till Satcubes kontor i Göteborg varpå inkoppling av utvecklingskort och komponenter samt modifiering av programkoden genomfördes. Många olika kodstrukturer och upplägg testades under arbetets gång, bland annat möjligheten att använda två inklinometrar istället för bara en. Den ena inklinometern skulle då sträva efter att befinna sig i nollplanet precis som innan medan den andra skulle mäta systemets hastighet för att få en jämnare hastighet på delplattformarnas rörelser och på så sätt uppnå en bättre reglering.

4 FÖRSTUDIE

4.1 Havets rörelser

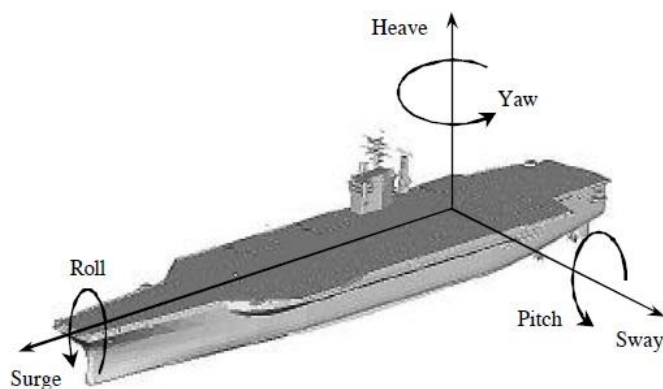
Naturligtvis lyder samtliga rörelser och vågor i havet de naturliga och fysikaliska lagarna som teoretiskt gör det möjligt att räkna på, men komplexiteten är vad som sätter gränsen till vad det idag är möjligt att räkna fram. Att sammanfoga ett oändligt antal varierande rörelser, likt vitt brus, till en enda ekvation som förklarar havets tillstånd vid en given tidpunkt; dåtid, nutid eller framtid, har ännu inte arbetats fram.

Idag finns det komplicerade men ändå förenklade matematiska tillståndsmodeller av vågor och rörelser i havet som till viss del förklarar verkligheten beroende på tillämpning. Dessa modeller kan lämpa sig för diverse användningsområden inom tekniken och gör det på så sätt möjligt att laborera med.

En tillståndsmodell som är särskilt trivial att använda sig av är approximationen av havets vågor till sinusformade vågor av olika amplitud- och frekvenskaraktistik [19]. Med en sådan tillståndsmodell blir det genast mycket enklare att utföra beräkningar på nödvändiga data som det annars inte hade varit möjligt att räkna fram eller testa på grund av brist på resurser eller utrustning för lämpliga experiment.

En del data som kan räknas fram från den sinus-förenklade modellen är till exempel våghastighet, acceleration, läge, frekvens, amplitud och period.

4.2 Rörelsemönster av farkoster till sjöss

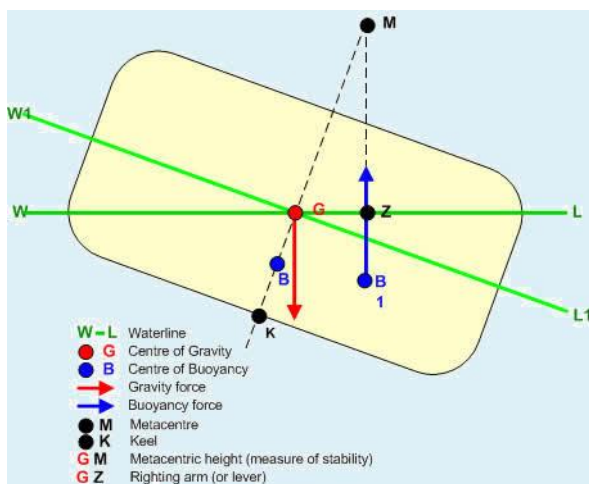


Figur 4.1. Förekommande rörelseriktningar för fartyg, [7]

Det är oundvikligt att havets rörelser påverkar en kropp som befinner sig i mediet. Med fartyg som exempel finns ett oändligt antal rörelsemönster som kan antas beroende på fysiska egenskaper som storlek och vikt. De vanligast kategoriserade rörelseriktningarna är heave (translation i höjddled), surge (translation fram/bak), sway (translation i sidled), roll (rotation i sidled), pitch (rotation i höjddled) och yaw (rotation i horisontalplanet). En kombination av dessa rörelser beskriver på så sätt fartygets tillstånd till sjöss.

Generellt gäller att i likadana vattentillstånd rör sig mindre fartyg mer än större fartyg. Däremot kan ett större fartyg röra sig mer än ett mindre fartyg på ett oroligt hav, då det mindre fartyget har större chans att ”åka med” vågorna än det stora fartyget som istället påverkas av dess kraft. Fartygens olika rörelsemönster beror således helt på havets tillstånd samt fartygets fysiska karakteristik. Vid närmare utredning blir det snabbt många icke-statiska variabler som spelar in vilka tillsammans bildar ett problem som ökar exponentiellt i svårighetsgrad att lösa.

Med hjälp av tidigare nämnda förenklade tillståndsmodeller samt en del konstanta variabler på diverse fartygs fysiska karakteristik är det möjligt att utföra beräkningar



med tillfredställande resultat. En viktig variabel som är konstant men som förändras beroende på fartygets fysiska egenskaper så som vikt, bredd, höjd och masscentrum är den metacentriska höjden [16].

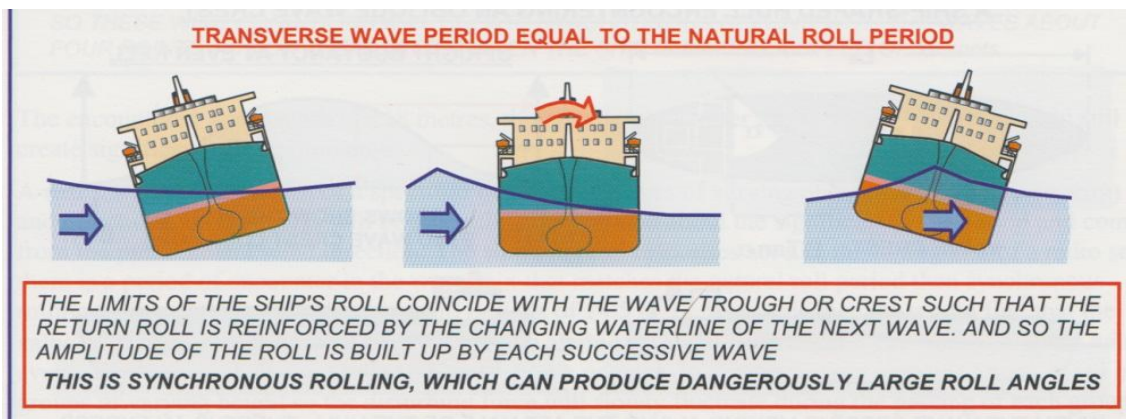
I figur 4.2 syns symbolerna G, M, B, B1 och Z. G är tyngpunkten för fartyget medan B är flytkraften för det undanträngda vattnet. B1 är flytkraftens angreppspunkt vid lutning som samtidigt bestämmer vridmomentet Z på fartyget.

Figur 4.2. Den metacentriska höjden illustrerad som avståndet mellan 'G' och 'M' [8].

Metacentrum (M) bestäms där normalen till tyngdpunkten av fartyget (G) korsar den vertikala kraftriktningen

av flytkraftens angreppspunkt (B1). Då M befinner sig ovanför G kommer fartyget alltid försöka förbli flytande eftersom det bildas en positiv hävarm till vridmomentet för att rätta upp fartyget. Om M hamnar under G finns det en mycket stor risk att fartyget kapsejsar då hävarmen till det upprätande vridmomentet blir för litet eftersom skrovet då har en tyngdpunkt långt ovanför vattenytan. En positiv metacentrisk höjd (avstånd \overline{GM} i figur 4.2.) är alltså alltid att föredra vid skeppsbyggande.

En av de centrala punkterna till att förstå fartygets rullande rörelsemönster är skillnaden i metacentrisk höjd då den påverkar rörelserna avsevärt. Det blir samtidigt stora skillnader i maximala roll-vinklar som fartyget kan utsättas för. Ett exempel på detta är en segelbåt som har maximala roll-vinklar medan en gummibåt har minimala sådana, bortsett från att skrovets fysiska egenskaper påverkar denna egenskap.



Figur 4.3. Illustration av vågperioder och naturliga roll-perioder i en kombination som kan leda till kapsejsbildande roll-vinklar, [9].

Vidare behöver roll-vinklar inte endast bestämmas av metacentrisk höjd, de bestäms oftast av en kombination av våghöjd, -period, -frekvens samt den metacentriska höjden på fartyget. Studier och experiment pekar på att vissa kombinationer av riktning, infallsvinklar på vågor samt frekvens har stor betydelse för hur labil flytstabiliteten för ett fartyg blir till sjöss. [17]

Från figur 4.3. ses en kombination av en vågperiod som sammanträffar med ett fartygs naturliga periodtid vilken kan leda till extrema rollvinklar. Den typiska gränsen för rollvinklar som frambringar ett fartyg att kapsejsa är ca. 30°.

4.3 Marknadsundersökning av aktuella lösningar

Med förstudien bestående av rörelsemönster hos fartyg i minnet krävs det ett lämpligt ramverk för produkten att utgå från. Då produkten i fråga strävar efter att ersätta nuvarande lösningar på marknaden både prestanda- och prismässigt är det lämpligt att göra en jämförelse mot dessa. Bland de största aktörerna inom branschen finns SeaTel, Orbit, Thrane & Thrane och KVH.

Många tillverkare delar liknande tekniska specifikationer med varandra vilket ger en bra vägledning till hur utvecklingen av produkten ska ske bättre. Det finns däremot också avvikande tekniska specifikationer beroende på vilka fartyg företaget byggt produkten för. Som ovan nämnt är det möjligtvis mer lämpligt att ha en lägre vinkelhastighet på plattformen för stora fartyg medan det krävs en snabbare vinkelhastighet för mindre fartyg. Andra fartyg ändrar inte sin kurs särskilt mycket och därför inte heller satellit för kommunikationslänk, medan andra ska kunna ha tillgång till alla tillgängliga satelliter och ändrade kursrutter vilket direkt ändrar på vinkelhastighetskrav och toleranser på pekprecision.

Tabell 4.1. Exempel på relevanta tekniska specifikationer från större aktörer

Tillverkare/Modell	Pitch max/min	Roll max/min	Yaw max/min	Hast.	Accel.
SeaTel 4009-33	$\pm 14^\circ$ @ 6-12 sec	$\pm 24^\circ$ @ 8-12 sec	$\pm 8^\circ$ @ 15-20 sec	12°/s	15 °/s ²
Orbit AL-7103-Ku Mk II	$\pm 15^\circ$ @ 6 sec	$\pm 30^\circ$ @ 8 sec	$\pm 8^\circ$ @ 15 sec	12°/s	-
Thrane SAILOR 900 VSAT	$\pm 15^\circ$	$\pm 30^\circ$	$\pm 10^\circ$	15°/s	15 °/s ²
Intellian v110	$\pm 15^\circ$	$\pm 25^\circ$	$\pm 8^\circ$	12°/s	5 °/s ²

Tabell 4.1. visar en grundläggande idé om vilka tekniska specifikationer och krav studien ska kunna rikta sig in på och om möjligt utföra bättre. Grundidén innan marknadsundersökningen har varit pitch- och rollvinklar på $\pm 25^\circ$ som i marknadsundersökningen visar en tydlig skillnad när det gäller pitchvinklar.

Valet av $\pm 25^\circ$ beror på att systemintegratören i framtiden kan installera systemet i en riktning som är fördelaktig för det fartyg VSAT installeras på. Detta medför att pitch- och rollaxlarna förblir olika beroende på fartyg och installatör. En kravbild på $\pm 25^\circ$ i både pitch- och roll är därför fördelaktig för att motverka framtida konflikter med installationsriktning och resulterande prestanda.

En anledning till att konkurrenter har lägre toleranser på pitchvinklar än vad studien avser använda kan vara för att samtliga tillverkare utformat sina plattformar för fartyg och hav som ej riskerar utsättas för högre vinklar än $\pm 15^\circ$. Samtidigt kan det observeras en ganska likartad vinkelstandard för rollvinklar i marknadsundersökningen jämfört med den grundläggande vinkelspecifikationen för studien, då $\pm 30^\circ$ anses vara den allmänna gränsvinkeln till att fartyg börjar kapsejsa.

Förklaringen till skillnaderna mellan roll- och pitchvinklarna och varför de inte är likadana i kravspecifikationer hos tillverkare leder tankarna tillbaka till flytångreppspunkter på fartyget. På ett fartyg som är cirkulärt och sfäriskt underifrån kommer det finnas precis lika många flytångreppspunkter på alla sidor av fartyget vilket säkerställer att fartyget kan vinkla sig precis lika många grader åt olika håll då flytkraftshävarmen för vridmomentet blir precis lika lång på alla sidor.

På ett fartyg som å andra sidan liknar ett konventionellt fartyg med cylindriskt och avlångt skrov kommer mängden flytångreppspunkter och längden på flytkraftshävarmen för vridmomentet i pitchvinkel vara så pass många respektive stora att fartyget får väldigt svårt att vinkla sig mer än $\pm 15^\circ$. Däremot kommer hävarmen och mängden flytångreppspunkter i rollvinkel att vara mycket mindre, vilket innebär en större frihet i mängden grader fartyget kan luta sig. Som bekant kapsejsar ett fartyg oftast aldrig fram eller bak, den kapsejsar oftast alltid åt sidan.

4.4 Fastställande av lämpligt fartyg för arbetet

Den enklaste metoden att bestämma en lämplig kravbild för ett visst fartyg är att förhållas till en särskild storlek vilket det utformas en kravbild kring, eftersom den

överenskomna storleken på fartyget alltid under fysiskt normala omständigheter kommer befinna sig inom ett visst referensområde gällande roll- och pitchvinklar samt rullnings- och vinkelhastigheter.

Samtidigt som valet av en fartygstyp praktiskt är enklare att bestämma en teknisk kravbild kring begränsas automatiskt den tillgängliga marknaden för produkten till endast den del som innehar ett fartyg en plattform utformats för. Därför är det fördelaktigt att tänka bredare gällande kravbilden och tekniska specifikationen för att nå ut till så mycket av marknaden som möjligt och plocka så stora marknadsandelar som möjligt.

Att tänka bredare kan i stora drag innebära att det bestäms en ”övre” eller ”undre” gräns på fartygsegenskaper som det sedan blir ett mål att sträva efter. I studien har det däremot visat sig att det blev mest fördelaktigt att bestämma en ”undre” gräns för ett fartyg som kravspecifikationen skulle utformas kring eftersom målet var att plattformen skulle kunna klara av så många olika förhållanden på hav som möjligt.

Den minsta fartygsstorleken som skulle tas i beaktning var av en storlek motsvarande Stridsbåt 90. Eftersom reglersystemet hade en gräns till hur bra den skulle fungera var det lättare att försöka anpassa möjliga fartygsstorlekar till reglersystemet än tvärtom.

Med det minsta fartyget i beaktning hade projektet således ett stort spann av fartyg och därmed marknad att tillgodose.



Figur 4.4. Illustration av Stridsbåt 90, [10]

4.5 Slutgiltiga kravbilden

Att bestämma sig för ett fartyg och sedan tillämpa ett reglersystem var inte det lättaste när olika väderförhållanden och hav spelade roll eftersom det fanns ett oändligt antal kombinationer av varje som i så fall fick tas hänsyn till. Som tidigare nämnt visade det sig betydligt enklare att först titta på andra konkurrenter och deras produkters tekniska specifikationer och sedan försöka tillämpa ett reglersystem som skulle överträffa deras med samtidig hänsyn till extrema väder- och havsförhållanden som det tänkta fartyget möjligen skulle utsättas för.

Trots detta blev det svårt att bestämma en konkret kravbild eftersom tekniska data på diverse fartygs fysiska rörelser och periodtider på olika hav och väderförhållanden inte fanns tillgängliga. Det blev därför lättast i slutändan att utgå ifrån möjligheterna för simulering av plattformens rörelser i labbet eftersom det var enkelt att simulera en realistiskt snabb rörelse för ett fartyg för att sedan se ifall reglersystemet hann ikapp och kunde stabilisera sig. Syftet med detta tillvägagångssätt var att redan från början försöka simulera plattformen för en väldigt liten och rörelserik båt som möjligt som en grund för det framtida arbetet.

Till en början åstadkoms rörelsesimuleringen av plattformen med handkraft som med enkelhet kunde uppnå 40-50 grader/sekund. Denna vinkelhastighet var långt över konkurrenternas specifikationer men ställde även stora krav på reglersystemets respons. En begränsning av den simulerade vinkelhastigheten till en rimligare men utmanande nivå var att föredra för att ligga steget före konkurrenterna men också för att utveckla ett reglersystem som passade så stor mängd fartyg, hav, väder samt system inom SOTM/COTM segmenten som möjligt.

Samtidigt var det viktigt att ha komponenterna som användes i åtanke då kravbilden kunde begränsas eller försämrats beroende på de olika sensorernas och givarnas prestanda. Den centrala komponenten som kunde utgöra den största begränsningen var inklinometern. Eftersom den hade en uppdateringsfrekvens på 20Hz och peknoggrannheten behövde vara ca $0,1^\circ$ var det lätt att räkna ut hur snabb vinkelhastigheten behövde vara för att behålla peknoggrannheten vid en viss sampelfrekvens.

Den maximala vinkelhastigheten med hänsyn till inklinometerns uppdateringsfrekvens och peknoggrannheten som önskades räknades fram till 2 grader/sekund vilket direkt var mycket mindre än konkurrenternas tekniska specifikationer. Våra tidigare tester som hade uppnått 40-50 grader/sekund var därför både korrekta men samtidigt felaktiga eftersom peknoggrannheten försämrades avsevärt då många avläsningar inte togs i beaktning på grund av den låga samplingsfrekvensen av inklinometern. I skuggan av denna begränsning stod det klart att reglersystemet skulle behöva fungera väldigt bra på många tappade sampel och ändå behålla den önskade peknoggrannheten.

Den slutgiltiga kravbilden resulterade i en marginell höjning av konkurrenternas tekniska specifikationer, givet inklinometerns uppmätta prestanda. Värt att notera är att den bestämda vinkelhastigheten i kravbilden är omöjlig att uppnå med tanke på inklinometerns uppdateringsfrekvens, men med ett bra reglersystem var det tänkt att uppnå en sådan vinkelhastighet med tappade sampel i beaktning.

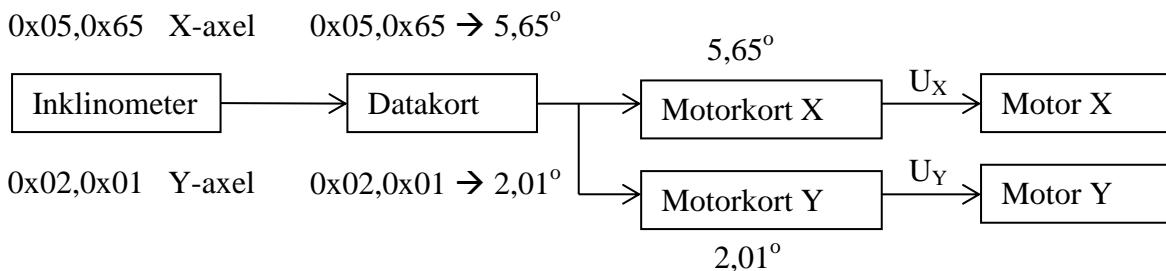
Tabell 4.2. Den slutgiltiga kravbilden på plattformens prestanda

Tillverkare	Pitch	Roll	Yaw	Hast. [deg/s]	Acc. [deg/s²]
Satcube AB	$\pm 25^\circ$	$\pm 25^\circ$	0°	13	12

5 UTFÖRANDE

5.1 Principiella idén

Plattformen är tänkt att reglera sig i båda sina motoraxlar mot horisontalplanet inom ett intervall av $\pm 25^\circ$. Vinkelvärden från inklinometern tas in i datakortet kontinuerligt via UART med hjälp av en avbrottsrutin som har i uppgift att behandla all inkommande data och sedan skicka denna information till en lämplig subrutin i programmet för lämplig formatering. Slutligen sänds den formaterade informationen återigen via UART till respektive motor-kort där en avbrottsrutin tar vid och läser in data. Dessa data görs sedan om till lämplig styrsignal för respektive motor.



Figur 5.1. En schematisk och övergripande bild av idén till programmets struktur och flöde.

5.2 Begränsningar i mjukvaran

Programmeringsspråket i Pinguino är en förenklad version av C där utvecklarna skrivit sina egna rutiner och bibliotek som går att använda. Att språket förenklats och egna bibliotek implementerats gör programmeringen mycket enklare och mottaglig för den genomsnittlige användaren, vilket i andra fall skulle kräva en kunskap om processor-nära kommandon, register, konfigurering och bitsättning.

Även fast Pinguino och Arduino riktar sig mot amatörer och hobbyister har en del spektakulära idéer skapats under projektens livstid vilket förstärker det faktum att ett förenklat programmeringsspråk och kommersiellt konstruerade kort inte begränsar komplexiteten av projekten som går att uppfinna vilket många underskattar.

Det är naturligtvis möjligt att utan problem programmera kortet i en annan utvecklingsmiljö som inte behandlar Pinguino-språket. Det går följaktligen att programmera kortet i ANSI C, men väljs denna metod för programmering finns inte de förenklade språkfunktionerna tillgängliga.

Skillnaderna mellan Pinguino och ANSI C är inte särskilt stora men språken är trots allt väldigt olika. Den största skillnaden är att Pinguino använder sig av inbyggda bibliotek som främst gör bakomliggande ANSI-C kod mycket lättare att förstå. I andra fall hjälper biblioteken till att korta ner kod som annars hade varit flera rader lång till en enda rad.

Ett exempel på detta kan visas genom en enkel kodrad som har i uppgift att tända en LED-lampa. I Pinguino används följande kodrad:

```
pinMode(LEDpin, HIGH);
```

Medan det i ANSI C (MikroC, MPLAB) behöver skrivas:

```
LATDSET = 0x05;
```

Alt.

```
LATDbits.LATD5 = 1;
```

I detta fall var Pinguinokoden inte kortare men betydligt lättare att förstå än dess motsvarighet i ANSI-C.

Båda kodexempel utför samma operation men kan uppfattas som helt olika till en utomstående. Det går dock tydligt att se syftet av Pinguinos kodrad i jämförelse med kodraden i ANSI-C för att uppnå samma resultat. I ANSI-Cs kodexempel förutsätts det att programmeraren har kunskap om register i processorn som skall modifieras, vad registerna heter samt var i registret det ska ändras utan att påverka andra bitar i samma register. Syftet framgår alltså inte lika tydligt och kan därför ta ett tag för en utomstående nybörjare att sätta sig in i tänket och komma igång. Pinguino hjälper på så vis programmeraren på traven.

En fördel med att skriva kod i ANSI-C är att det skrivs enligt ANSI-standard. Det begränsar alltså inte förståelsen för utomstående programmerare som ska ta del av koden som möjligtvis inte bekantat sig med Pinguino, samt att koden i sin helhet är precis likadan i alla kompilatorer som väljer att stödja ANSI-C vilket gör det möjligt att enkelt byta utvecklingsmiljö utan att behöva skriva om stora delar av koden för en ny kompilator.

En tredje fördel är att programmeraren betydligt mer kontroll över programkoden på hårdvarunivå vilket tillåter modifiering av register det annars inte hade varit möjligt att ändra på vid användning av Pinguino. Projekt av större slag, alternativt professionella projekt, skrivs med fördel i ANSI-C mest på grund av dessa tre fördelar.

Vinsterna med Pinguino är att det är nybörjarvänligt för mindre projekt och tillåter programmeraren att koncentrera sig mer på idén och programflödet än tekniska buggar, viktiga ”kom-ihåg” fakta om aktuella processorn eller hur processorn fungerar i detalj på hårdvarunivå. Koden blir också väldigt logisk och lättförståelig för nybörjare samt att mjukvaruplattformen i sig bidrar till en bra kunskapsbas för den som vill lära sig från grunden och sedan ta sig vidare till med komplicerade projekt.

I början av studien användes Pinguinos egna utvecklingsmiljö och språk. Experiment med stegmotorn samt processen för att bekanta sig med den omkringliggande hårdvaran genomfördes också i det förenklade gränssnittet med bra resultat. Däremot började små problem att uppenbara sig på vägen.

För att utvecklingskortet körades Pinguino skulle kommunicera med en dator använde den sig av sin USB anslutning. Datorer som kommunicerar med mikrocontrollers med hjälp av USB-Seriell adapters använder ett särskilt gränssnitt för detta som kallas CDC.

Med CDC möjliggör datorn att mikrocontrollern kan kommunicera med en emulerad COM-port, dvs. en virtuell seriell port likt RS-232 men helt emulerad i programkod.

Med hjälp av CDC är det möjligt att i vanliga fall skriva/läsa information till/från en terminal på datorn som avlyssnar motsvarande virtuella COM-port med hjälp av följande kommando i Pinguino:

```
CDC.printf("Denna rad skickas till datorn");
```

Motsvarande kod i ANSI-C hade sträckt sig flera A4-sidor långt då CDC är ett stort och omfattande bibliotek som möjliggör USB kommunikation. Kodraden ovan är ett väldigt bra exempel på hur Pinguino underlättar för programmerare genom att korta ner kod från några A4 sidor till en rad och därmed minska på projektets storlek och komplexitet.

En stor del av programmeringen i början gick ut på att skriva värden och information till terminalen för att fastställa att ett korrekt program skrivits som matade ut förväntade data. Att skriva strängar som exemplet ovan var således inga problem men det var först när information i form av siffror och decimaltal behövde skrivas ut som Pinguino började visa sina brister.

Som exempel går det att skriva en siffra sparad i variabeln "Siffra" till datorn med följande kommando i Pinguino:

```
CDC.printf("%d", Siffra);
```

Där "%d" är en symbol som betecknar att det endast skrivs siffror och ersätts därmed med värdet av variabeln "Siffra". Eventuella decimaler försummas och heltalet rundas alltid nedåt, t.ex. talet 1.67 sparad i "Siffra" skrivs istället till datorn som talet 1.

Beräkningarna som gjordes i programmet för att uppnå en bra reglering var däremot inte alltid utformade att resultera i jämna tal på grund av stegantalet på motorn och microstepping-inställningarna på drivkretsen till motorn.

På grund av detta behövdes ett sätt att skriva ut decimaltal till datorn. I vanliga fall fungerar detta med motsvarande taltyp i samma kodsekvens som ovan:

```
CDC.printf("%f", Decimaltal);
```

Där "%f" betecknar flyttal som ersätts med värdet av variabeln "Decimaltal" med eventuella decimalplatser inkluderade.

Efter mycket testande var slutsatsen att Pinguino-utvecklarna ännu inte implementerat printf() funktionen i CDC-biblioteket för Pinguino vilket var väldigt viktigt för projektets fortskridande. Utan funktionen var det väldigt svårt att se ifall programmet fungerade som förväntat.

Med detta stora och projektbegränsande tillkortakommande var ett byte av utvecklingsmiljö oundvikligt. Ett byte innebar att börja om på programmeringsdelen av projektet, att lära oss en ny miljö och hur dess kompilator fungerade samt vilka makron som fanns för att underlätta programmeringen. Det var på så vis i början en betydande omställning att byta från Pinguinos enkla gränssnitt, utvecklingsmiljö och språk till den

mer invecklade utvecklingsmiljön med ANSI-C som språkstandard som var MikroElektronika MikroC Pro for PIC32.

Trots omställningen visade sig MikroC vara en omfattande och fulländad utvecklingsmiljö med debugging-funktioner och en egen terminal som alla var långt mycket bättre och funktionella än i Pinguinos motsvarigheter. Den stora omställningen visade sig vara i det som tidigare sköttes bakom kulisserna av Pinguino som istället behövde konfigureras manuellt, som diverse konfigurationsbitar och register, men i slutändan var det en välbehövlig omställning.

5.3 Kommunikation inom systemet

Kommunikation till och från inklinometern sker seriellt över RS-232 genom en 9 pin DSUB kontakt. Olika kommandon är möjliga att skicka till inklinometern för att ha möjlighet att ändra baud-rate och automatisk utmatning av avlästa vinkelvärden bland andra kommandon. Kommandona skickas seriellt i form av hexadecimalt formaterade och fördefinierade kommandoord av storleken 1 byte. Vid varje mottagande av information från inklinometern mottages tillhörande kontrollbitar, kommandoord och checksumbitar alla kodade i hexadecimala siffror med storleken 1 byte.

Mottagning av information från inklinometern sker med en uppdateringshastighet av 20Hz och en baud-rate vilken är förinställd på 9600bps. De avlästa vinkelvärdena skickas från inklinometern seriellt i form av hexadecimala tal som är kodade så att ett avläst vinkelvärde på t.ex. 32° skickas ut som 0x32 hexadecimalt. De hexadecimala talen motsvarar således inte deras decimala motsvarigheter.

För programförståelsens skull var det betydligt enklare att förmedla information mellan korten i decimala tal, men för att kunna tillämpa idén behövdes de avlästa hexadecimala värdena först konverteras till decimala tal med hjälp av strängvariabler.

De inlästa värdena från inklinometern sparades därför i en strängvariabel vilken är definierad till att vara lika stor som antalet bytes inklinometern skickar ut vid varje sampelfrekvens (inklusive diverse kontrollbitar och kommandoord), i detta fall 14 bytes. En sådan variabel kan deklarerars enligt nedanstående kodrad där samtliga platser i strängen är från början nollställda:

```
char c[14]={0,0,0,0,0,0,0,0,0,0,0,0,0,0};
```

X-Axis: +020.10 ° Y-Axis: -005.25°

68 0D 00 84 00 20 10 10 05 25 00 00 00 FB

Figur 5.2. Ett exempel på ett datautskick från inklinometer vid en viss vinkel för X och Y, [11]

I figur 5.2 visas en rad med hexadecimala tal som skrivs ut från inklinometern vid varje vinkelförfrågan. De första fyra bytes består av två par fixa tal "68" och "0D", sedan adressen på sensorn "00" samt kommandoordet som i detta fall är "84".

Följande tre bytes består av "00 20 10" vilka motsvarar först en byte för tecken där positivt motsvarar "00" och negativt "10", sedan två par bytes för själva vinkelvärde; en byte för heltal och en byte för ett tvåsiffrigt decimaltal. I exemplet ovan betecknas alltså en positiv vinkel på 20.1° för X-axeln som "00 20 10". Analogt för Y-axeln gäller att en vinkel på -5.25° betecknas som "10 05 25."

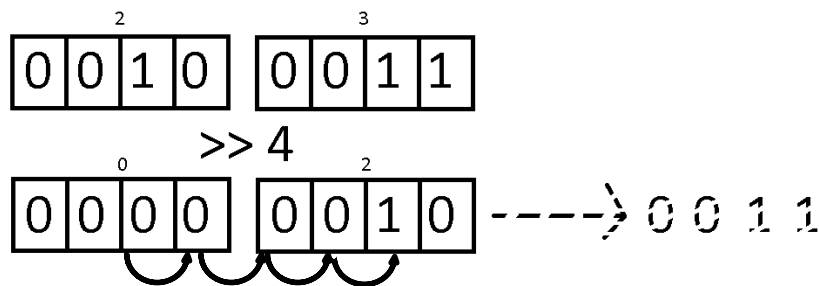
Efter de bytes som beskriver de avlästa vinklarna följer tre bytes vilka är förinställda av tillverkaren att alltid visa "00." Den sista byten är en checksumma som summerar alla föregående bytes.

Både heltal och decimaltal från inklinometern behövde behandlas för att korrekt kunna konvertera värdena till sina decimala motsvarigheter. Nedan följer en enkel kodrad som behandlar heltalen:

```
heltal = ((c[5]>>4)*10+((c[5]) & 0x0f));
```

Variabeln "c" är strängvariabeln som används vid inläsning av strängen inklinometern skickar ut (se figur 5.2). Talet inom hakparenteser berättar vilket index den befinner sig på i strängvariabeln "c." Eftersom index inom C-programmering alltid börjar på noll befinner sig det inlästa hexadecimala heltalet från inklinometern på plats 5 i strängvariabeln "c." En snabb jämförelse med figur 5.2 visar att detta stämmer om databyten "68" har index noll.

Operationen som utförs sedan betecknas med symbolen ">>" som är en bitvis högerskift.

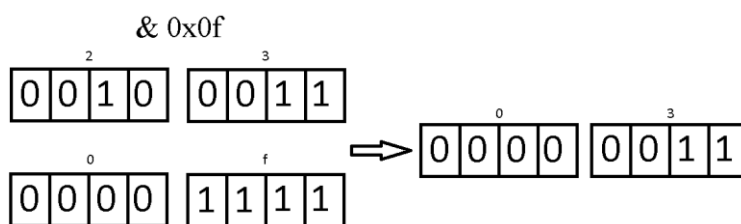


Figur 5.3. Illustration av en bitvis högerskift med fyra steg,

I figur 5.3 visas hur tiotalet i det hexadecimala talet "0x23" plockas ut genom en högerskift av samtliga bitar. Tioalet får således värdet av ett ental, i detta fall "2," och resten av bitarna försummas.

Nästa operation i kodraden multiplicerar entalet med tio för att slutligen få en representation av det ursprungliga hexadecimala talet i dess decimala form. Talet "0x20" i hex blir därför "20" decimalt och omvandlingen av tiotalet är klart.

Den sista delen av kodraden utför en AND-operation för att plocka ut de hexadecimala entalen som rakt av motsvarar sina decimala värden.



Figur 5.4. Illustration av en logisk AND-operation för att isolera ental

I figur 5.4 illustreras hur entalet "3" plockas ur det hexadecimala talet "0x23" med hjälp av en mask. Varje bit i masken (0x0f) multipliceras med motsvarande bit i det hexadecimala talet "23". Resultatet blir entalet "3".

Den fullständiga omvandlingen av heltalet i vinkelavläsningen från inklinometern är således klar. Vad gäller decimaler i vinkelvärde tillämpas samma princip, varpå det kompletta och omvandlade heltalet i ovanstående exempel och de omvandlade decimaltalen adderas för att få en komplett representation av det avlästa vinkelvärde i decimalform.

Dessa omvandlade tal skickas sedan vidare till övriga utvecklingskort med hjälp av en särskild kommunikation. Utvecklingskorten har flera olika möjligheter att kommunicera med periferienheter på; bland annat två UART kanaler, I²C och SPI. Det enklaste och mest kompatibla med annan hårdvara är seriell kommunikation genom UART vilket därför var det självklara valet av kommunikation mellan korten sinsemellan och dator.

Det seriella kommunikationsprotokollet kan arbeta på olika spänningsstandarder som RS-232- och TTL-nivå. Skillnaden mellan standarderna är att de uppfattar logiska nollor och ettor olika beroende på spänningsmängd. TTL-standarderna tolkar logiska nollor som

spänningar mellan +0V till +0,8V och logiska ettor som spänningar mellan +2,0V och +5V, medan RS-232 standarden tolkar dem som -3V till -25V respektive +3V till +25V. Eftersom utvecklingskortets processor körs på så låga spänningar som +3,3v är det TTL nivå som processorn kan kommunicera på utan att skada sina interna kretsar av överspänning.

För att kommunikationen mellan korten skulle vara tillfredställande behövde den bli synkroniserad. Synkroniseringen av informationen sköttes av kontrollbitar som programmerades in i början på datasträngarna som skickades. En kontrollbit i detta fall bestod av en bestämd symbol med ett ASCII-kodat tecken, i detta fall "C". Dessa kontrollbitar används för att berätta för nästa utvecklingskort att en ny sträng är redo att tas emot och som samtidigt säkerställer att rätt data hamnar på rätt platser i minnet.

Samma typer av kontrollbitar används vid inläsning från inklinometer då det är viktigt att datasträngen läses in endast då datasträngen i sin helhet kan lagras. Strängvariablerna senare i programflödet beror direkt på motsvarande index i den inkommande datasträngen för att kunna behandla informationen rätt och vidarebefordra denna i ett synkroniserat tillstånd.



Figur 5.5. En RS232-TTL nivåomvandlare av Sparkfun Electronics, [12]

För att systemet skulle ha möjlighet att kommunicera på det beskrivna sättet åstadkoms detta med en mellanhand i form av extra hårdvara eftersom olika system och hårdvara är sällan anpassade till samma spänningar och strömmar. Det behövdes följaktligen en adapter mellan utvecklingskortet och inklinometrarna i form av en RS-232 ↔ TTL-nivåomvandlare som kunde skala upp TTLs låga spänningsnivåer till RS-232s högre motsvarighet och tvärtom.

På grund av tidigare problem med CDC biblioteket inom Pinguinos IDE krävdes också ett nytt sätt att kommunicera med datorn på som inte krävde ett CDC bibliotek. Detta löstes genom att återigen kommunicera seriellt men krävde då samtidigt en RS-232 ↔ USB adapter eftersom datorerna som användes inte hade en seriell port att tillgå. Ibland var vissa komponenter producerade med samma kön på kontakterna varpå en könbytare var nödvändig men även då krävdes lite modifiering för att få kommunikationen rätt eftersom Rx- och Tx-linjerna behövde korsas för att fullborda kommunikation vid inkoppling av hona-hona eller hane-hane.

Förutom spänningsanpassningen mellan RS-232 och TTL fordrades en form av spänningsanpassning för kommunikation mellan utvecklingskort och lägesgivare då de endast fungerade på spänningar mellan +9V och +30V och matade en utspänning som var direkt proportionell mot inspänningen. Ingångarna som användes på utvecklingskortet hade som begränsning en inspänning på +3,3V där alla spänningar över den nämnda riskerade skada motsvarande pinnar på processorn. Spänningsanpassningen för lägesgivarna bestod följaktligen av en enkel spänningsdelning med användning av 2,5kΩ och 800kΩ resistorer och en fast inspänning på +13V för att få en utspänning på ca +3V.

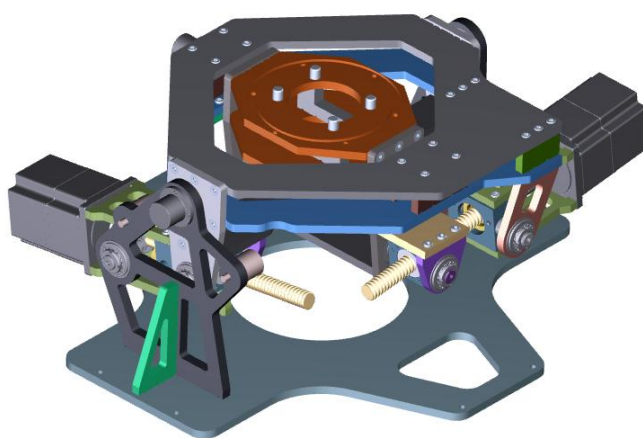
Utöver kommunikation behövdes ett lämpligt sätt att skicka steg till motorn. Då pricksäkerheten för plattformen som skulle uppnås var 0,1° var den ursprungliga

steglängden på $1,8^\circ/\text{steg}$ allt för hög, därför användes en drivkrets med microstepping-funktionalitet av modell CW6060. På så sätt var det möjligt att dramatiskt minska antalet grader/steg för att uppnå en efterfrågad vinkelupplösning på $0,1^\circ$ och slutligen en mycket jämnare reglering. (Se Bilaga A. s.A1 för kopplingsschema)

Slutligen då det tidigare blev en övergång från Pinguinos utvecklingsmiljö och bootloader till MikroC var det inte möjligt att programmera kortet genom USB som tidigare hade gjorts. Det fick därför köpas till en programmerare av modell PICKit 2 som kunde använda sig av Pinguinokortets mini-ICSP kontakt för programmering av kortet.

5.4 Plattformens mekanik

Plattformens huvuddelar består av en inre och en yttre delplattform som drivs av var sin stegmotor via kulsruvar. Den inre delplattformen sitter monterad med sin tillhörande stegmotor i den yttre och kan därför rotera både i x- och y-led. Det är på den inre delplattformen som inklinometern sitter och strävar efter att ligga i nollplanet, det är också här resten av ESV-terminalen kommer monteras.

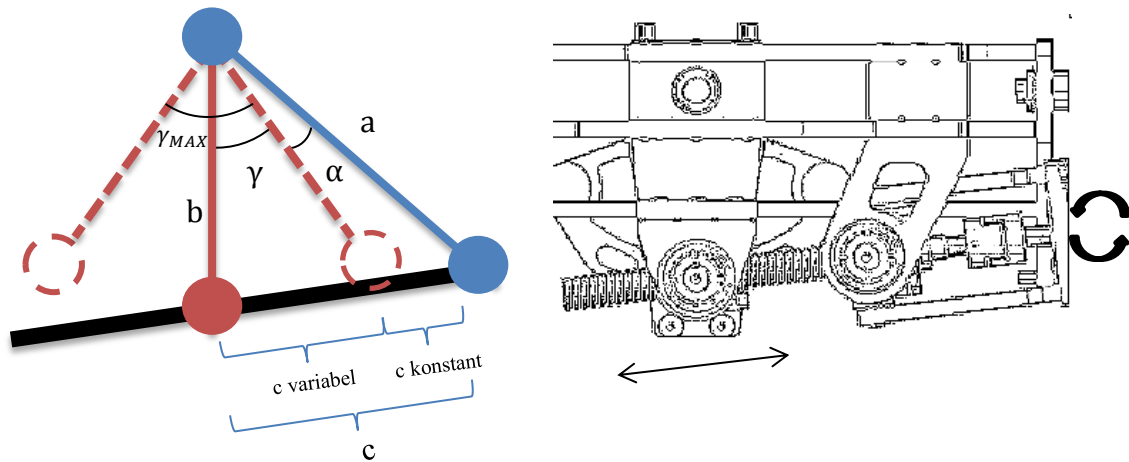


Figur 5.1. Översiktsbild av plattformens konstruktion och utseende, [13]

På grund av plattformens konstruktion kommer en olinjär rörelse att uppstå på de båda delplattformarnas vinkeländringar.

Vid analys av den inre delplattformen ses att det finns tre punkter varav två är fasta. Med cosinussatsen och avståndet mellan dessa punkter kan denna olinjäritet beskrivas.

Från figur 5.2. illustreras att sidorna **a** och **b** har fasta längder medan sida **c** är kulsruven ändrar längd beroende på delplattformens lutning. Sida **c** har en konstant längd i början som är den minsta längden på kulsruven som kan uppstå innan plattformen tar i sig själv, denna längd ger upphov till minsta vinkeln mellan **a** och **b** som är α . Efter den konstanta längden på kulsruven kommer den del som varierar för att få delplattformen att vinkla sig. Den varierande delen på kulsruven ger upphov till vinkeln γ och delplattformens rörelse. De två röstreckade linjerna utgör intervallet som delplattformarna kan röra sig inom.



Figur 5.2. Plattformens mekaniska rörelse illustrerad med vinklar och längder, jämfört med en skiss av den verkliga konstruktionen med tillhörande rörelseriktning, [14]

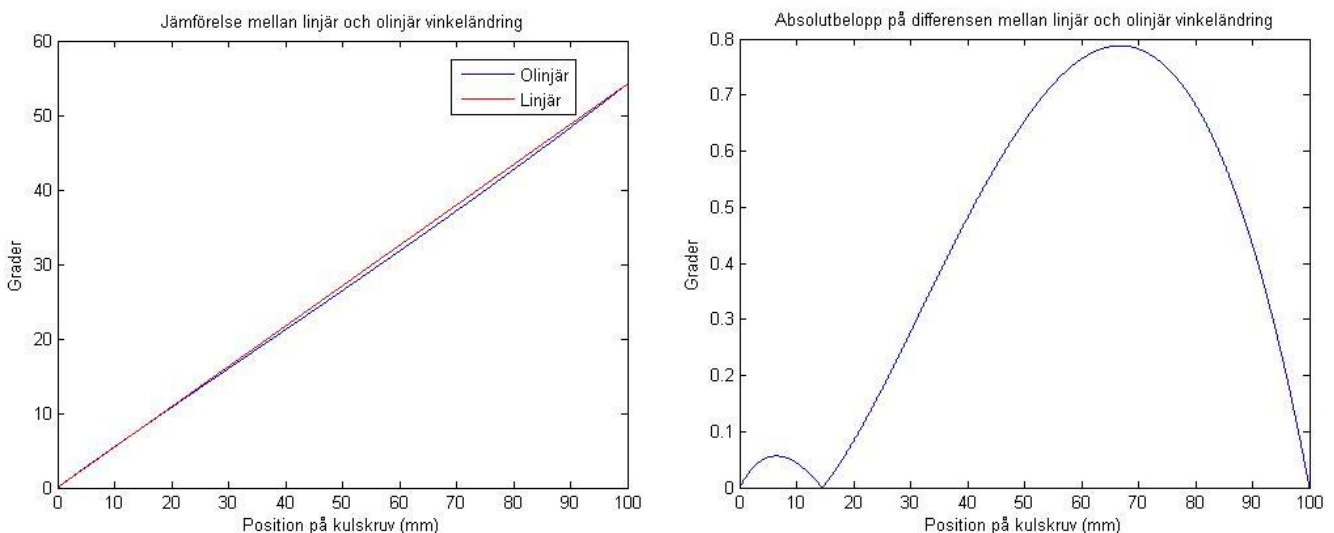
För att få ut förändringen på vinkel γ när längden c varierar används cosinussatsen:

$$c^2 = a^2 + b^2 - 2ab \cos \gamma \quad (5-1)$$

Där γ löses ut:

$$\gamma = \cos^{-1} \frac{(a^2 + b^2 - c^2)}{(2ab)} \quad (5-2)$$

Med hjälp utav Matlab jämförs en linjär vinkeländring med den verkliga olinjära vinkeländringen på inre delplattformens tidigare beskrivna rörelseintervall. I figurerna nedan ses att en skillnad på hastigheten uppstod som kunde ha påverkan i ett senare skede utav projektet.



Figur 5.3. Illustration av den olinjära rörelsen i två motsvarande grafer

Analogt gäller för den yttre delplattformen men ett annat resultat erhålls på grund av längre avstånd på sida a (se Bilaga B. s.B1).

5.5 Programbeskrivning och tester med en sensor

5.5.1 Informationskortet och motorkortens programflöden.

Ett utav de tre utvecklingskortet är avsett till att ta in och behandla informationen från sensorn och sin tur skicka ut denna till de två motorkorten. Här görs också de konfigureringar på sensorn så att den skickar data i både X och Y led och att detta sker kontinuerligt. Informationen som läses in via avbrott på UART görs om så att de inlästa värdena blir till decimaltal som tidigare beskrivits. När behandlingen utav informationen är färdig skickas den vidare återigen via UART. (se Bilaga C. s.C1)

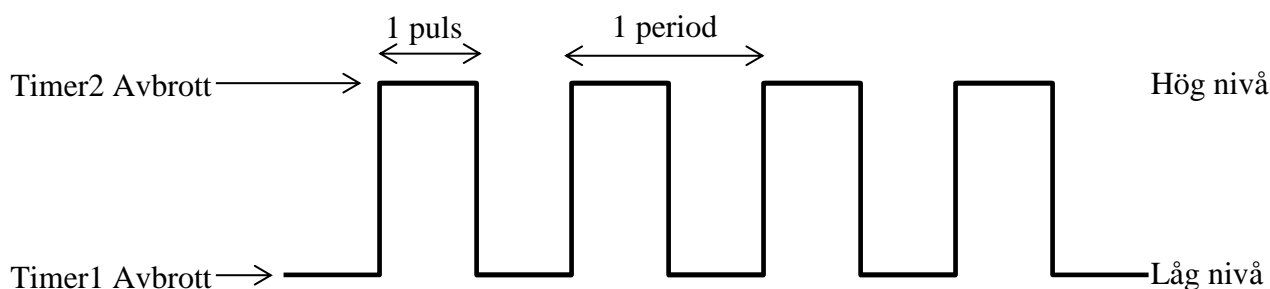
I de båda motorkorten tas respektive X- och Y-axelns vinkelvärden skickade från informationskortet in kontinuerligt via avbrott på UART och ses där som fel i grader gentemot nollplanet. Dessa fel ska försöka elimineras genom att driva stegmotorerna i rätt hastighet, riktning och antal steg. Programet till motorkorten är uppbyggt så att efter inläsningen utförs jämförelsen med en brusnivå för att bedöma om plattformen verkligen är i rörelse eller om det bara är brus från inklinometern som tas emot.

När brusnivån överstigs bestäms en rotationsriktning på stegmotorn beroende på positivt eller negativt fel och körning utförs tills felet är eliminerat och då börjar processen om. Under tiden körningen äger rum räknas den aktuella hastighet ut på en förutbestämd rampkurva, denna rampkurva hjälper motorn att accelerera och komma upp i högre hastigheter. (se Bilaga C. s.C2).

Själva drivningen utav stegmotorerna sker via avbrott på Timer1 och Timer2 genom att signalen till motorerna sätts till hög i Timer2 och låg i Timer1 om vartannat, detta skapar ett pulståg som driver motorerna. Utöver att sätta motorsignalen låg i Timer1 utförs en uträkning på förändringshastigheten också här för att senare kunna användas till D-verkan i regleringen i Timer2 (se Bilaga C. s.C3). Det är sedan i Timer2 som regulatoralgoritmen finns och frekvensen på pulserna till stegmotorerna bestäms för att ändra hastighet genom rampning. Ändlägesgivarna kollas kontinuerligt för att säkerställa att delplattformarna befinner sig inom tillåtna intervallet (se Bilaga C. s.C4).

5.5.2 Bestämning av pulsfrekvens, hastighet och dess samband

För att ändra rotationshastigheten på stegmotorn måste som tidigare nämnts frekvensen på pulserna till motorn ökas och minskas. Eftersom en puls skickas varje gång Timer2 får ett avbrott så är det alltså tätheten på dessa avbrott som styr frekvensen och därmed hastigheten. Det register som bestämmer efter hur många klockcykler ett avbrott inträffar är PR2 registret. Detta medför att ett minskat värde i PR2 ger snabbare avbrott som betyder högre frekvens på pulserna till stegmotorn som i sin tur får en högre hastighet. Tvärtom gäller vid minskning utav hastigheten där värdet i PR2 då ökas. Timer1 har ett motsvarande register till PR2 som heter PR1, detta ställs in kontinuerligt så att avbrott på Timer1 sker mittemellan Timer2 avbrotten för att få jämna pulser.



Figur 5.4. Illustration över tidsförlopp gällande de motordrivande avbrottsrutinerna Timer1 och Timer2

Ett problem som uppstår med ovan beskrivna pulståget och den hastighet på stegmotorn som erhålls från detta är att en linjär ändring utav värdena i PR2 registret inte ger en linjär ändring i hastigheten på tillhörande delplattform. Därför utformades den tidigare nämnda rampkurvan så att värdena i PR2 registret skulle motsvara en linjär hastighetsförändring på delplattformarna (se Bilaga B. s.B2).

Hastigheten och dess maximala värde beror också på vilken steglängd drivkretsen till stegmotorn är inställd på då en mindre steglängd erfordrar ett högre antal pulser för att åstadkomma ett varvs rotation. Det ökade värdet på antalet pulser per varv kräver därför alltså en ökning utav avbrotten på Timer2 och därmed en minskning utav PR2 registret. Värdet i PR2 registret kan bara minskas till en viss gräns innan avbrott sker för ofta och resten utav programmet påverkas negativt. En högre upplösning ger alltså lägre maximal hastighet men gör det samtidigt möjligt till bättre precision och jämnare rörelse på motorns rotation.

Tabell 5.1. Exempel på möjliga stegupplösningar

Upplösning	Steglängd (grader)	Steg/varv
1/1	1,8	200
1/2	0,9	400
1/4	0,45	800
1/8	0,225	1600
1/16	0,1125	3200
1/32	0,05625	6400

Genom projektet användes upplösningen 1/32 i störst utsträckning eftersom denna inställning bedömdes ge den bästa kombinationen utav precision och maximal hastighet.

För att kunna erhålla rampkurvan för PR2 registrets värden som motsvarade en linjär hastighetsökning på delplattformarna arbetades ett samband mellan dessa fram:

B' = vinkelhastighet (grader/s)

$steg_{MAX}$ = antal steg i intervallet

grader = antal grader i intervallet

$steg_{varv}$ = antal steg/varv

tp = tid i sekunder mellan intervallgränserna

sys_clk = klockfrekvensen på PIC (80 MHz)

n = varvtal

PR2 = klockcykler/steg (avbrott)

$$tp = \frac{grader}{B'} \quad (5-3)$$

$$n = \frac{steg_{MAX}/steg_{varv}}{tp} \quad (5-4)$$

$$PR2 = \frac{sys_clk}{n*steg_{varv}} \quad (5-5)$$

$$\Rightarrow PR2 = \frac{sys_clk * grader}{B' * steg_{MAX}} \quad (5-6)$$

Detta samband validerades med mätningar där ett intervall på $28,5^\circ$ användes till att mäta tiden tp . Ekvation 5-1 användes för att räkna ut mätt B' och ekvation 5-4 för att framräknat B' .

PR2	Räknat B' (°/s)	tp (s)	Mätt B' (°/s)	Räknat B' -Mätt B' (°/s)
60000	0,552727	51,5	0,553398	-0,00067
50000	0,663273	43,0	0,662791	0,000482
40000	0,829091	34,4	0,828488	0,000603
30000	1,105455	25,7	1,108949	-0,00349
20000	1,658182	17,2	1,656977	0,001205
10000	3,316364	8,6	3,313953	0,00241
5000	6,632727	4,3	6,627907	0,00482

Även om differensen mellan räknat och mätt värde på B' är väldigt liten måste en tanke läggas vid att de uppmätta B' värdena endast är medelvärden över intervallet på $28,5^\circ$ och vidare att intervallet just är $28,5^\circ$ och därför inte omfattar hela intervallet på 50° . Det finns alltså en chans att differensen blir större vid olika tidpunkter beroende på delplattformarnas olinjära rörelser som tidigare beskrevs i kapitel 5.4. Denna faktor bedöms inte spela någon större roll eftersom sambandet mellan PR2 och B' endast används till att få fram en riktlinje för hastighetsbestämningen och behöver då inte vara exakt.

5.5.3 Framtagande av regulatoralgoritm

Valet av regulatorn utgick från en PID-regulator men eftersom systemet anses vara självintegrerande, det vill säga inget kvarstående fel uppstår valdes I-verkan bort och kvar blev då en PD-regulator. Parametrarna bestämdes sedan med Ziegler-Nichols självsvängningsmetod som bygger på att studera systemet vid självsvängning. Först

ställdes PD-regulatorn in som en ren P-regulator och förstärkningen K ökades successivt tills systemet precis hamnade i självsvängning.

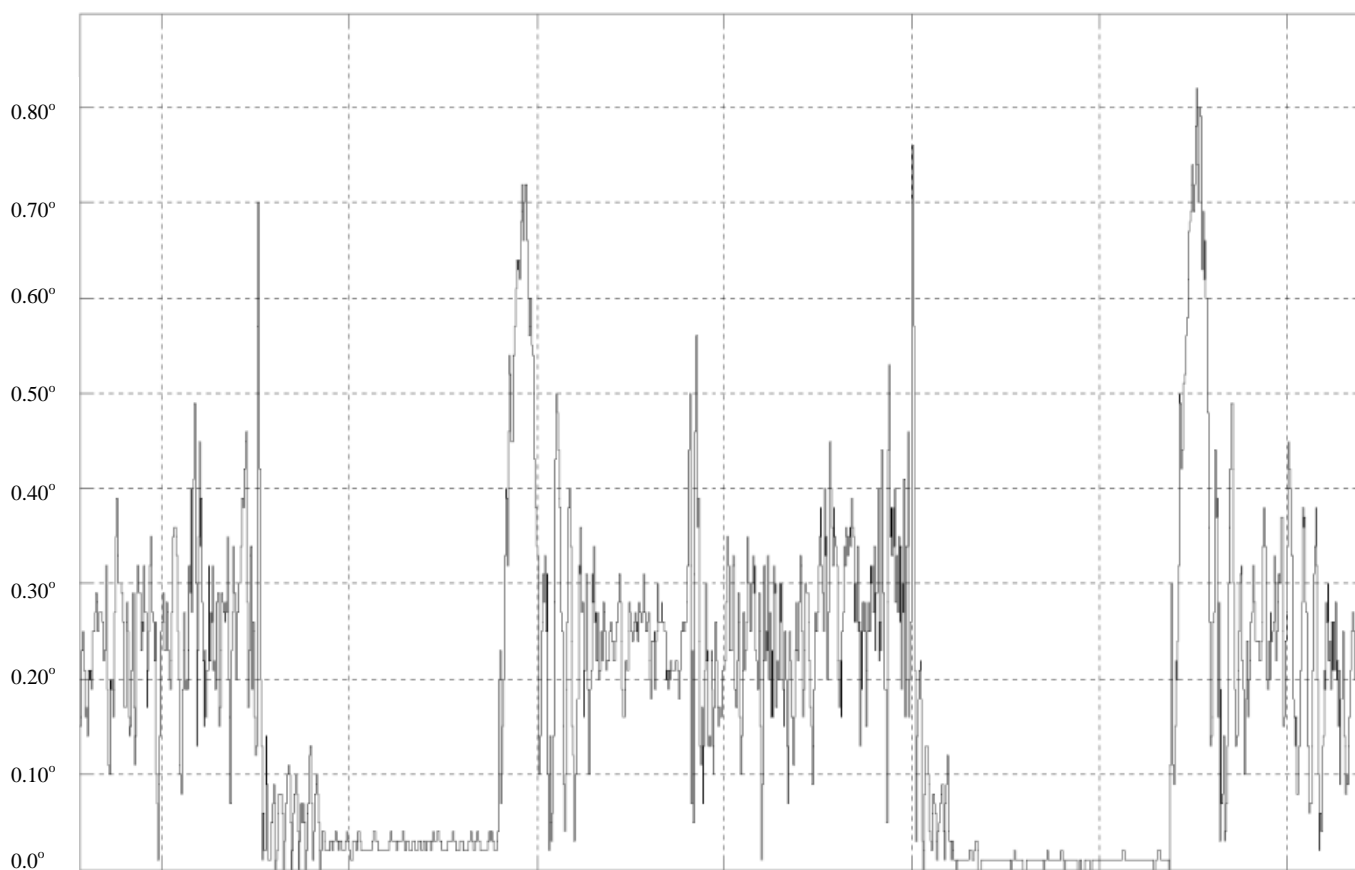
Utsignalen från inklinometern registrerades och vid granskning uppvisade den en oscillerande rörelse, från denna signal bestämdes en genomsnittlig periodtid T_0 . Med periodtiden T_0 och förstärkningen K_0 kunde parametrarna räknas fram med hjälp av en tabell.

Tabell 5.2. Hjälpstabell för bestämning av reglertekniska parametrar vid Ziegler-Nichols svängningsmetod med användning av framräknade T_0 - och K_0 -värden [18].

Regulatortyp	Parametrar		
	K	T_I	T_D
P-regulator	$0,5K_0$	-	-
PI-regulator	$0,45K_0$	$0,85T_0$	-
PID-regulator	$0,6K_0$	$0,5T_0$	$0,125T_0$

Eftersom inte parameterbestämningen till en ren PD-regulator fanns att tillgå bestämdes K och T_D från PID-regulatorns värden. På grund av detta och med tanke på att Ziegler-Nichols metod generellt ger en ganska grov parameterinställning behövdes en del finjusteringar göras för att få så bra resultat som möjligt.

5.5.4 Test av systemets prestanda



Figur 5.5. Graf över utmatad vinkeldata från systemet i körning, med illustration över systemets respons och reglertekniska egenskaper vid flera starter och stopp samt insvängningsförlopp.

Ovan visas resultatet av reglersystemets prestanda för reglering i Y-led, likvärdigt resultat uppnås vid test i X-led. Testet utfördes under 50 sekunder och hela plattformens hastighet varierade mellan 0 – ca 2°/s. Y-axeln representerar felet i grader och X-axeln är tidsspännet på 50 sekunder. Rörelsen på plattformen åstadkoms med tre elektriska domkrafter vilket medförde ryckiga start/stopp och begränsad hastighetsbestämning.

Från resultatet ses inverkan utav dessa ryckiga start/stop före och efter det att systemet hamnat i vila runt nollplanet. Det är då de största felet på ca 0,7° – 0,8° uppvisas, anledningen till detta är som tidigare beskrivits att inklinometern är känslig för de stötar som uppstår och visar därför på en “falsk vinkel”. Utöver de pikar som uppstår befinner sig delplattformen ca 0,1° – 0,4° ifrån nollplanet vid körning, detta är följderna av att både regleringen och hastigheten på delplattformarna beror på felet utgående från en hastighet på 0°/s, vilket medför att vid körning kommer delplattformarna alltid ligga efter. Denna eftersläpning kommer sedan att växa i samband med att hastigheten på hela plattformen ökar. När väl plattformen kommer i vila hittas nollplanet snabbt och med bra precision.

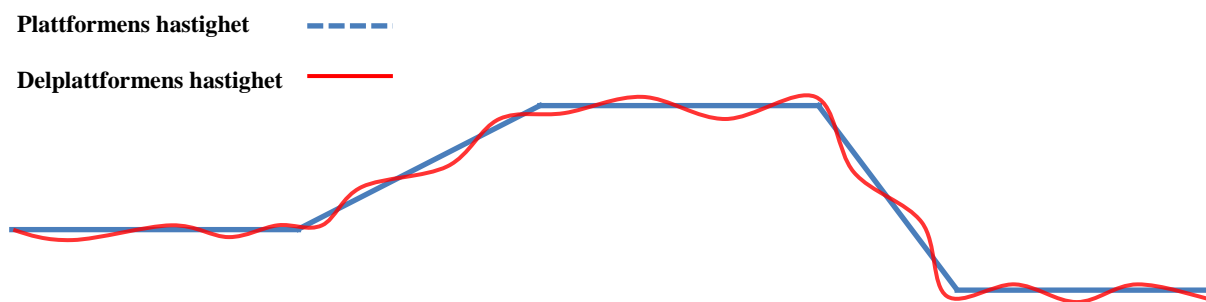
Resultatet som erhöles var inte godtagbart för även om nollplanet hittades bra när hela plattformen befann sig i vila är det ett scenario som sällan inträffar. Med dessa resultat i

hand behövdes ett nytt upplägg och tänk men insikten att inklinometern var undermålig blev också tydlig.

5.6 Tankar och tester med en extra sensor

Om en tillbakablick görs till förstudien av hur ett fartyg påverkas av havets rörelser och hur dess hastighet påverkar plattformen växer en ny tanke fram om hur systemet borde utformas. När ett fartyg går i vågor kommer en naturlig ökning och minskning av hastigheten att uppstå när fartyget vaggar från sida till sida. Precis samma hastighet kommer plattformen att utsättas för och därför skulle vetskapen om denna hastighet kunna utnyttjas till förbättrad reglering.

Det nya upplägget utgick från att använda en extra sensor som då skulle sitta på plattformens bas och mäta dess hastighet. Detta skulle hjälpa till att lösgöra regleringen och hastighetsbestämningen på delplattformarna från varandra och enbart låta regleringen bero på felet. Delplattformarnas hastighet skulle alltså då bestämmas utgående från hastighetsmätningen på hela plattformens rörelse. Med detta upplägg skulle den tidigare problematiska eftersläpningen kunna elimineras och nollplanet hittas även när plattformen är i rörelse.



Figur 5.6. Schematisk bild över systemets- och delplattformens hastigheter.

Med en extra sensor blev det också möjligt för delplattformarna att ligga runt nollplanet och reglera sig oberoende av vilken sida nollan de låg. Tidigare kunde de bara närma sig nollan innan de bytte riktning eller stannade, men med den extra sensorn kunde det avgöras om plattformen var i vila och om i så fall delplattformarna skulle kunna hamna i vila också.

Ny programkod togs fram och en till likadan inklinometer monterades på plattformens bas för att mäta vinkelförändringen så att dess hastighet kunde räknas ut i grader per sekund.

$$B' = \frac{(|\text{En gammal vinkel} - \text{Nuvarande vinkel}|)}{\text{Uppdateringsfrekvens}} \quad (5-7)$$

Gränser på maximala och minimala hastigheter beroende på plattformens hastighet togs fram så ett rörligt tillåtet hastighetsintervall fanns för att undvika orimligt stora hastigheter på delplattformarna när hela plattformen har liten hastighet eller tvärtom. Alla tänkbara värden på intervallet och även på rampkurvan räknas ut och läggs i tabeller innan huvudprogrammet tar vid.

Dessa värden hämtas sedan genom indexering i huvudprogrammet vilket kräver mindre processorkraft än att räkna ut värdena kontinuerligt. I fallet med en sensor behövdes bara en uträkning till rampkurvan och då kunde en kontinuerlig uträkning utföras utan problem men när ett tillhörande intervall ska räknas ut är det fördelaktigt att göra allt detta innan. (se Bilaga C. s.C5 till C8)

Vid testning märktes det att användandet av en inklinometer för att bestämma plattformens hastighet inte var att föredra på grund av dess känslighet mot stötar och ryck, ett byte av sensortyp till en som inte påverkas utav stötar och ryck som till exempel ett gyroskop skulle troligen åtgärda problemet. Annars visade testerna på positiva resultat med potential att jobba vidare på.

6 SLUTSATSER OCH RESULTAT

Studien har uppgett några resultat som lett till delvis oväntade slutsatser. Här förklaras svaren under respektive rubrik till frågorna som ställdes i preciseringen av arbetsuppgiften innan studien påbörjades.

6.1 Inklinometern

Initialt skulle en mindre sidostudie kring en lämplig inklinometer utföras med hänsyn till plattformens kravbild. För att projektet inte skulle dra ut på tiden fick sidostudien utelämnas och en godtycklig inklinometer valdes istället.

Frågan som ställdes handlade därmed om ifall en lågkostnadsinklinometer var tillräcklig för att systemet skulle fungera som tänkt. De initiala tankegångarna och hypoteserna verkade stämma om att den skulle vara tillräcklig, men fler experiment och tester under projektets gång som utfördes i första hand för att försöka förbättra systemets prestanda i helhet med bland annat diverse programkod, kopplingar och specifika ändringar i reglersystemet visade allihop att inklinometern inte räckte till.

Anledningarna till att inklinometern var otillräcklig är ett par. Den främsta anledningen är uppdateringsfrekvensen som är specificerad till 20Hz vilket betyder att för varje 50ms som passerar skickar inklinometern ut en signal, en sampel, av den aktuella vinkeln med en noggrannhet på 0,03°. Responstiden har en direkt koppling till vinkelhastigheten i kravbilden eftersom det är sampeltiden som tillsammans med den önskade precisionen visar ifall responstiden är tillräckligt snabb för att undvika tappade sampel vid den önskade vinkelhastigheten.

Som tidigare nämnt räcker en enkel kalkylering för att visa att inklinometern inte räcker till. Den önskade precisionen i grader dividerar man på responstiden i sekunder och resultatet blir den maximala vinkelhastigheten inklinometern kan utsättas för utan tappade sampel.

$$v = \frac{P}{R} \quad (6-1)$$

Resultatet med värdena $R = 0,05s$, $P = 0,1^\circ$ ger $V = 2^\circ/s$. Jämfört med kravbilden som tidigare fastställdes till $13^\circ/s$ var detta direkt en väldigt stor begränsning mot konkurrenterna. Det visade sig väldigt svårt att av just denna anledning programmera ett tillfredställande reglersystem som arbetade på väldigt många tappade sampel vid väldigt hög precision.

En annan anledning till att inklinometern inte räckte till var själva funktionen av det interna mätinstrumentet. Problemet visade sig i att den var mekanisk och var därför en aning långsam innan den ställde in sig. Den var också otroligt känslig mot horisontella translaterande rörelser. Eftersom inklinometern var tvåaxlig kunde den mätas i både X- och Y-led. Om det gjordes en mätning i Y-led genom att luta inklinometern och samtidigt röra inklinometern i X-led över bordet gav den utslag i båda axlar trots att det endast var Y-axeln som var påverkad av lutning.

Detta problem försvårade utformningen av ett reglersystem ytterligare då plattformen under den dåvarande simuleringsanordningen (elektriska domkrafter) antog tvära start/stopp moment med konstanta och förutbestämda hastigheter. De snabba förändringarna i hastighet från startläge till stoppläge gav stora utslag i translaterande krafter på inklinometern vilka gav falska vinkelvärden till utvecklingskorten.

Det blev därför en del problem med att reglersystemet försökte korrigera på en vinkel som plattformen inte befann sig i som i sin tur gav upphov till en del pendling kring nollpunkten den skulle stabilisera mot. Men även vid en mjukare simulering (handkraft) visade sig reglersystemet otillräckligt med en samtidig antydning till den ovan nämnda diskussionen om inklinometerens uppdateringsfrekvens.

Gällande frågan om pekprecision är det fullt möjligt att uppnå en precision på $0,1^\circ$ men det beror på om det ska vara stillastående eller rörlig precision. Med inklinometern som fanns att tillgå bedömdes det inte möjligt att uppnå en rörlig precision på $0,1^\circ$ utan väldigt stora variationer vid körning samt start och stopp. Det visade sig däremot möjligt att ställa in sig på, som närmast, ca $0,2^\circ$ - $0,3^\circ$ från körning till stillastående. Efter några sekunder ställde sig plattformen slutligen in på $0,1^\circ$ men vid det laget, i realistiska förhållanden på ett fartyg i sjögång, hade fartyget hunnit vinkla sig flera tiotal grader.

Med föregående beräkningar på inklinometerens maximala vinkelhastighet i åtanke hade det varit möjligt att uppnå en rörlig pekprecision på $0,1^\circ$ om vinkelhastigheten på fartyget ej skulle överstiga $2^\circ/s$ samt om inklinometern inte hade påverkats av horisontella och vertikala translaterande rörelser, men eftersom havet påverkar samtliga fartyg i sjögång för betydligt större vinkelhastigheter var en så låg vinkelhastighet på inklinometern ej tillräcklig.

Slutligen är det diskutabelt ifall programmet, reglersystemet, inklinometern eller en annan del av systemet var ett bidrag till problemet. Det är också möjligt att flera komponenter eller flera aspekter kring mekaniken av plattformen eller den fysikaliska teorin kring elektroniken och störningar emellanåt bidragit till en sämre prestanda än förväntat. Dessvärre var dessa aspekter kring studien utanför dess omfång att analysera och experimentera med men bidrar trots det med råd om möjliga förbättringar.

6.2 Stegmotorerna

Under studiens gång visade sig aldrig några problem med stegmotorernas hastighet eller precision. Givetvis skulle precisionen på stegmotorerna ha varit undermålig om de hade körts med sin ursprungliga stegvinkel på $1,8^\circ/\text{steg}$ då detta var alldeles för mycket för att uppnå en precision på $0,1^\circ$ från inklinometern vilket var därför en drivkrets till motorerna användes som hade microstepping-funktionalitet. Med drivkretsen kunde vinklar som var väldigt mycket mindre än $0,1^\circ$ uppnås vid tillfredställande hastigheter.

Ett problem med stegmotorer är att de inte kan accelereras alltför fort vilket innebär att man får anpassa reglersystem och övrig programkod till en rampfunktion. Rampfunktionen får då även fungera vid hastighetsminskning eftersom motors lindningar bidrar till en induktiv last vilket agerar som tröghetsmoment. Det blir därför lätt att motorn stannar ett antal steg utöver programmets förväntan, som även det bidrar

till felmarginaler i vinkelavläsning och onödig korrigeringsreglersystemet som slutligen påverkar pekprecisionen.

6.3 Utvecklingskortet

PIC32MX440F256H på ett serietillverkat utvecklingskort av Olimex LTD. visade sig vara utomordentlig i kapacitet för uppgiften. Det var fördelaktigt att redan från början på studien tänka noga på att avlasta utvecklingskortet från onödiga beräkningar för att inte sakta ner systemets beräkningsförmåga och därmed reglering- och inläsningsförmåga vilket var en anledning till varför tre utvecklingskort användes för att dela på arbetet.

Teoretiskt hade antalet utgångar och ingångar på ett enskilt kort räckt för att driva två stegmotorer och samtidigt läsa in från en inklinometer, men arbetsbördan på processorn hade varit alldeles för stor med tanke på programmets längd, omfattning och komplexitet. Utöver detta hade en del konflikter förmodligen stötts på i programmeringen då programmet använder sig av ett fixt antal timers som i hela systemet med tre kort överstiger antalet timers ett utvecklingskort själv har att tillgå.

6.4 Programkod

Det finns ett väldigt stort antal sätt att uttrycka en rad kod, därför blir i princip alla människoskrivna program unika. Grundidén och planeringen för ett program som ska vara så snabbt och effektivt som möjligt är också svårt att bemästra vid första anblick. Med det sagt finns det en rad olika åtgärder på programkoden som skulle vara nyttiga för att optimera processen.

Inom perspektivet programmeringsspråk lämpar sig det åldrande C väldigt bra för ändamål som denna studie inriktat sig på. Med dagens teknik finns det snabbare och dyrare utvecklingskort vilka kör på högnivåspråk som samtidigt är betydligt mer optimerade och hade lämpat sig bra, om inte bättre, för denna studie. Trots detta har slutsatsen blivit att C visat sig vara ett tillräckligt effektivt, enkelt och snabbt språk att både programmera och låta processorn räkna på. Dessutom finns fördelen att språket används väldigt brett inom industrin och reglertekniska sammanhang vilket underlättar fortsatt arbete på projektet.

6.5 Ekonomiska ramar

Eftersom inklinometern visade sig undermålig i prestanda var det därför inte aktuellt att framföra en ekonomisk plan för utvecklingen av plattformen och jämföra kostnader mot konkurrenter. Det är således också svårt att bestämma en eventuell prisplan ifall det i framtiden bestäms att bättre hårdvara ska ersätta den som studien haft att tillgå.

6.6 Möjliga förbättringar

Den första självklara förbättringen till systemet är att byta ut inklinometern mot en annan som har högre precision, bättre felmarginal och är betydligt snabbare i uppdateringsfrekvens.

En högre precision med ca fyra decimaler istället för två skulle medföra fler möjligheter med lättare filtrering av inläsningsresultat som skulle ge en mjukare inläsning av vinkelvärden för behandling. En mjukare inläsning skulle sedan medföra en bättre reglering då värdena inte ändrar sig sporadiskt vilket med den nuvarande inklinometern medför till väldigt små ryck på ca $0,05^\circ$. En inklinometer med högre precision har med all sannolikhet en känsligare avläsningsmekanism som i alla fall skulle ge en bättre och mer noggrann avläsning på två decimaler än vad den nuvarande inklinometern presterar med samma antal decimaler. En mindre felmarginal på inklinometers mätning skulle ge mer precisa vinkelavläsningar rent generellt vilket automatiskt förbättrar regleringen.

En snabbare uppdateringsfrekvens skulle hjälpt väldigt mycket för att korrigera för den mängd sampel som uteblev med den nuvarande inklinometern. En snabbare samplingsfrekvens innebär att fler vinkelvärden kan avläsas vid en given tidpunkt vilket ger en väldigt precis avläsning på plattformens tillstånd. Med en väldigt precis tillståndsbild blir det mycket enklare för regulatorn att göra de nödvändiga korrigeringarna. Kortare intervall mellan mätningarna bidrar också till en minskad risk för att plattformen pendlar kring nollan då nollan ”upptäcks” snabbare och tillåter därmed regleringen att minska eller stoppa styrsignaler till motorerna vid rätt tidpunkt.

För att ge perspektiv på hur mycket snabbare uppdateringsfrekvens en bättre inklinometer behöver ha för att klara en vinkelhastighet enligt kravbilderna på $13^\circ/s$ medan den behåller en pekprecision på $0,1^\circ$ görs en beräkning med ekvation (6-1). Svaret blir en uppdateringsfrekvens på minst $R = 147\text{Hz}$. En ännu snabbare uppdateringsfrekvens möjliggör såklart en ännu högre vinkelhastighet den kan utsättas för, men brusfaktorn för avläsningarna blir högre vilket tvingar till en filtrering som i slutändan kan påverka regleringen beroende på utförande.

En annan förbättring till systemets generella prestanda och respons är påbyggnaden av ett riktigt gyroskop till konstruktionen. Tidigare utfördes tester med en extra inklinometer på basen av konstruktionen som skulle mäta hela konstruktionens lutning och från den lutningen beräkna en vinkelhastighet. Till en viss utsträckning var den metoden till hjälp då systemets prestanda och respons blev aningen bättre, men problemen som upplevdes med inklinometern så som uppdateringsfrekvens, känslighet för sidorörelser och brusig vinkeldata upplevdes ändå.

Ett ordentligt gyroskop hade istället bidragit till direkta mätningar på konstruktionens vinkelhastighet varpå extra kod för att manuellt utföra en hastighetsberäkning från avlästa vinkeldata kan undvikas. Detta hade sparat på processorkraft för beräkning och mängden kod i programmet vilket hade påskyndat reglerprocessen något. Ett gyroskop är dessutom specifikt utformad för sin uppgift medan en inklinometer istället är utformad att mäta statiska vinkeldata. Gyroskopet har därför bättre förmåga att mäta mer precist, ha snabbare uppdateringsfrekvens ($>20\text{Hz}$) och i det hela förbättra processen avsevärt med rätt implementering i reglersystemet.

Som en tredje förbättring skulle det vara fördelaktigt att tillämpa en IMU på systemet som kombinerar en inklinometer, accelerometer och gyroskop i ett enda chip. Kostnaden för en IMU med tillfredställande prestanda är däremot väldigt mycket högre än att köpa komponenterna som ingår separat. Däremot är en IMU en komplett och smidig lösning som tar väldigt lite plats och som ger bäst utsignal då de tre sensorernas mätvärden kombineras med diverse filter.

Vibrationer har varit en signifikant störning i vinkel- och hastighetsmätningar. Någon form av vibrationsdämpning på antingen sensorer eller konstruktion är att föredra.

En sista förbättring är programkoden som kan optimeras i både det principiella tänket och koden i helhet. Som tidigare nämnt finns det väldigt många olika sätt att skriva ett program på som utför samma handling, men vissa sätt är kortare och mer optimerade än andra. Det är svårt att komma till en slutsats angående programkoden i detta projekt på den fronten, men det finns alltid plats för förbättring.

6.7 Genomförbarhet

Med den begränsade inklinometern i baktanke har slutsatsen blivit att det inte är möjligt att genomföra projektet med den prestanda på plattformen som efterfrågas. En stor del av tiden på projektet spenderades med att försöka kompensera för inklinometerens nackdelar i hopp om att utforma ett reglersystem som mötte kraven. Att endast en inklinometer skulle vara tillräcklig för uppgiften är svårt att bestämma då en inklinometer med bättre prestanda hade behövt testas. I teorin skulle plattformens prestanda bli bättre om endast inklinometern byttes ut mot en bättre men i samband med t.ex. ett gyroskop är utsikterna för ett lyckat projekt med tillfredställande prestanda väldigt goda.

7 KÄLLFÖRTECKNING

7.1 Figurer

1: BRL-CAD. (24/03/2008) Azimuth, Elevation, Twist. *BRL-CAD Image Gallery*.
<http://brlcad.org/gallery/s/diagrams/diagram.png.html> (9/08/2012)

2: Boxall, J. Stacked Arduino Shields. *Arduino Shield List*.
<http://shieldlist.org/templates/images/stacked-shields.jpg> (26 Jun. 2012)

3: Olimex LTD. PIC32-PINGUINO-OTG INDUSTRIAL GRADE PINGUINO MAPLE ARDUINO LIKE DEVELOPMENT BOARD. *Olimex Development Boards and Tools*. <https://www.olimex.com/dev/pic32-pinguino-otg.html> (9/08/2012)

4: CONRAD. Contrinex inductive sensor DW-AD-603-M8. *Conrad Electronics*.
<http://www.conrad-uk.com/ce/en/product/155814/Contrinex-inductive-sensor-DW-AD-603-M8>. (9/08/2012)

5: RION-Tech., Digital Type Dual Axis Inclinometer. *RION Sensor Technology*.
<http://www.rion-tech.net/en/showproduct.asp?ID=7> (9/08/2012)

RION-Tech, Digital Type Dual Axis Inclinometer SCA126T-30 Datasheet (9/08/2012)

6: Eriksson, F. (6/03/1998) Stepper Motor Basics. *Industrial Circuits Application*.
6/03/1998. Ss. 1. <http://www.solarbotics.net/library/pdflib/pdf/motorbas.pdf> (9/08/2012)

7: Love, L.J. (2003) *Compensation of Wave-Induced Motion and Force Phenomena for Ship-Based High Performance Robotic and Human Amplifying Systems*. [Elektronisk].
Ss. 5. Oak Ridge, Tennessee: Oak Ridge National Laboratory.

8: Holt, T. (2012) Stability - Introduction. *Holtcraft*.
http://www.holtcraft.com/tip_stability1.php. (2012-03-28).

9: Clark, I.C.. (2005) A Ship's Motion in a Seaway. I *Ship Dynamics for Mariners*, red. Ian Clark, ss. 221. London: The Nautical Institute.

10: Wikipedia. Stridsbåt 90. *Wikipedia – Den fria encyklopedin*.
http://sv.wikipedia.org/wiki/Stridsb%C3%A5t_90. (29/6-2012)

11: Datablad för RION-Tech. Digital Dual Axis Type Inclinometer model SCA126T-30, *RION Inclinometer Data Format* (9/08/2012)

12: Sparkfun Electronics, RS232 Shifter SMD. *Sparkfun Electronics*.
<http://shieldlist.org/templates/images/stacked-shields.jpg>. (7/8/2012)

13: Satcube AB, *Datablad för Plattform*, (9/08/2012)

14: (Höger) Satcube AB, *Datablad för Plattform*, (9/8/2012)

7.2 Text

15: Eriksson, F. (6/03/1998) Stepper Motor Basics. *Industrial Circuits Application*. 6/03/1998. <http://www.solarbotics.net/library/pdflib/pdf/motorbas.pdf> (13/08/2012)

16: PRINCIPLES OF STABILITY, *Federation of American Scientists*, <http://www.fas.org/man/dod-101/navy/docs/swos/dca/stg4-01.html> (14/08/2012)

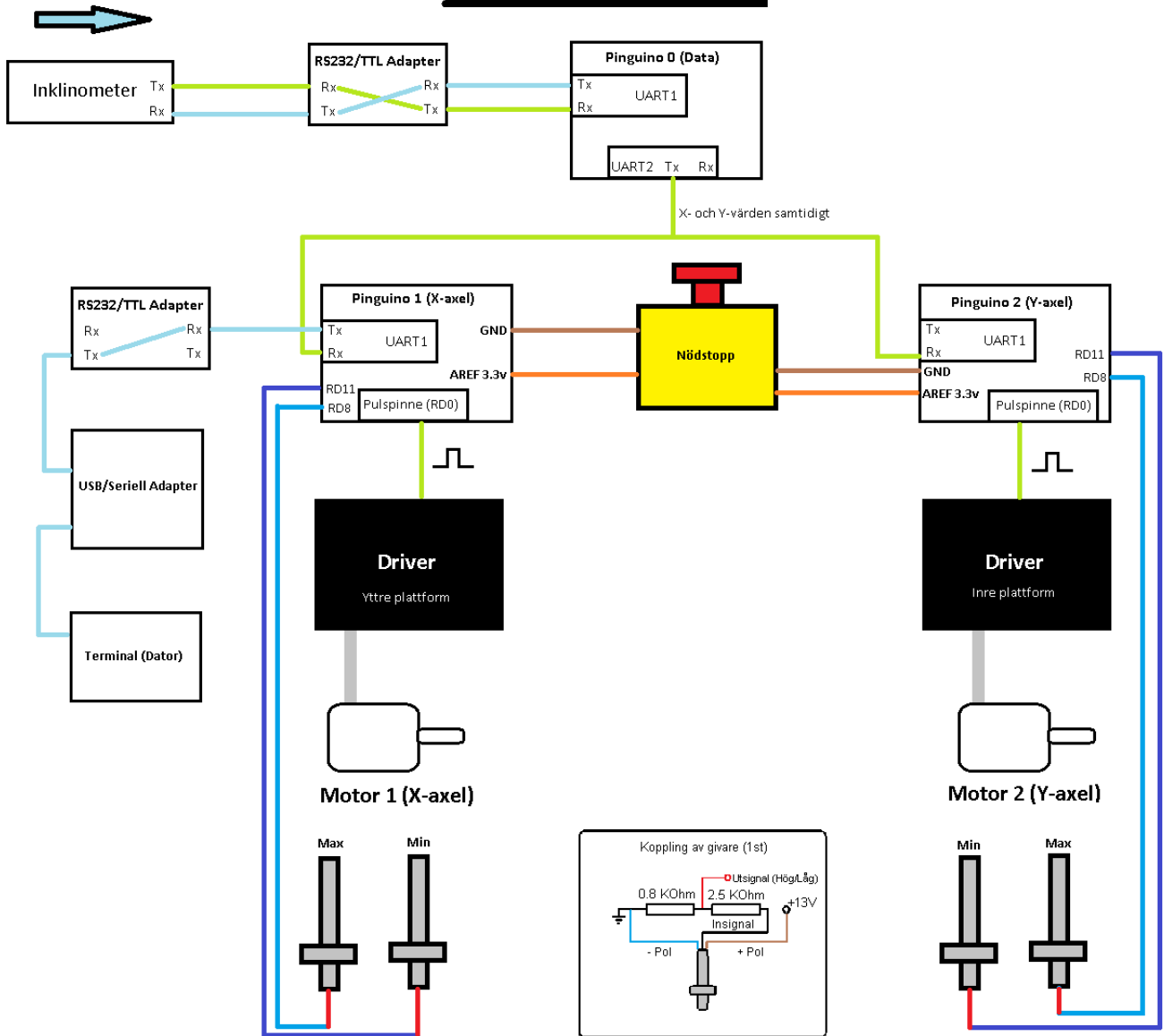
17: Clark, I.C.. (2005) A Ship's Motion in a Seaway. I *Ship Dynamics for Mariners*, red. Ian Clark, ss. 217-224. London: The Nautical Institute.

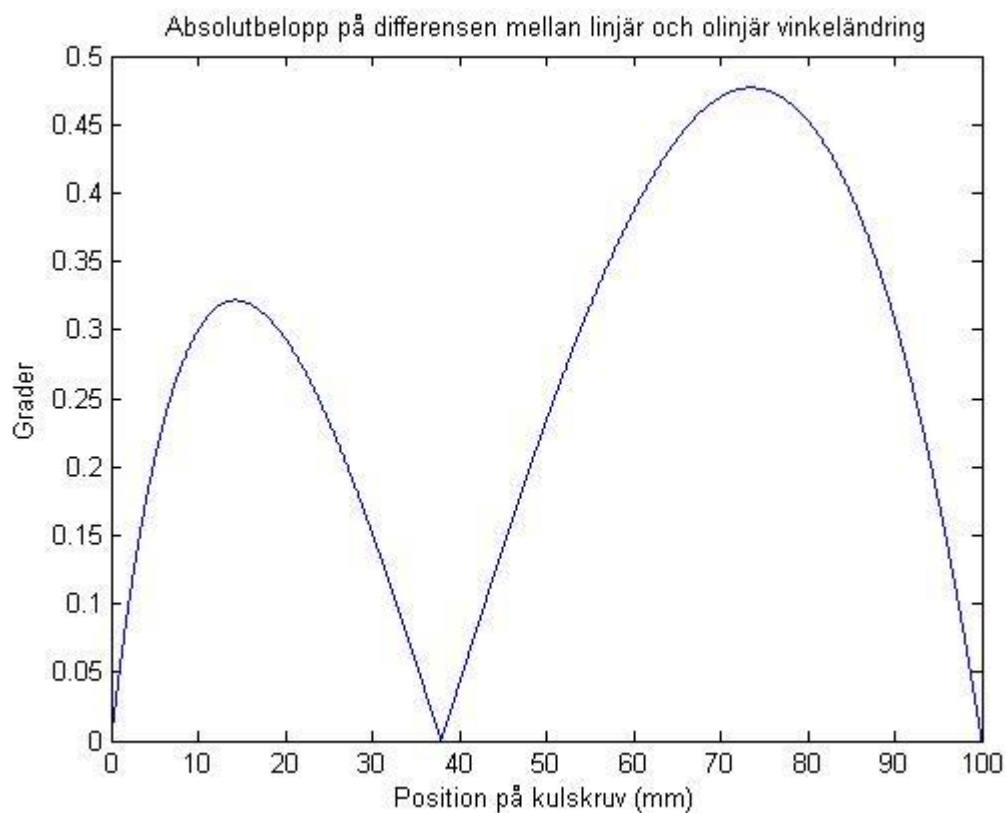
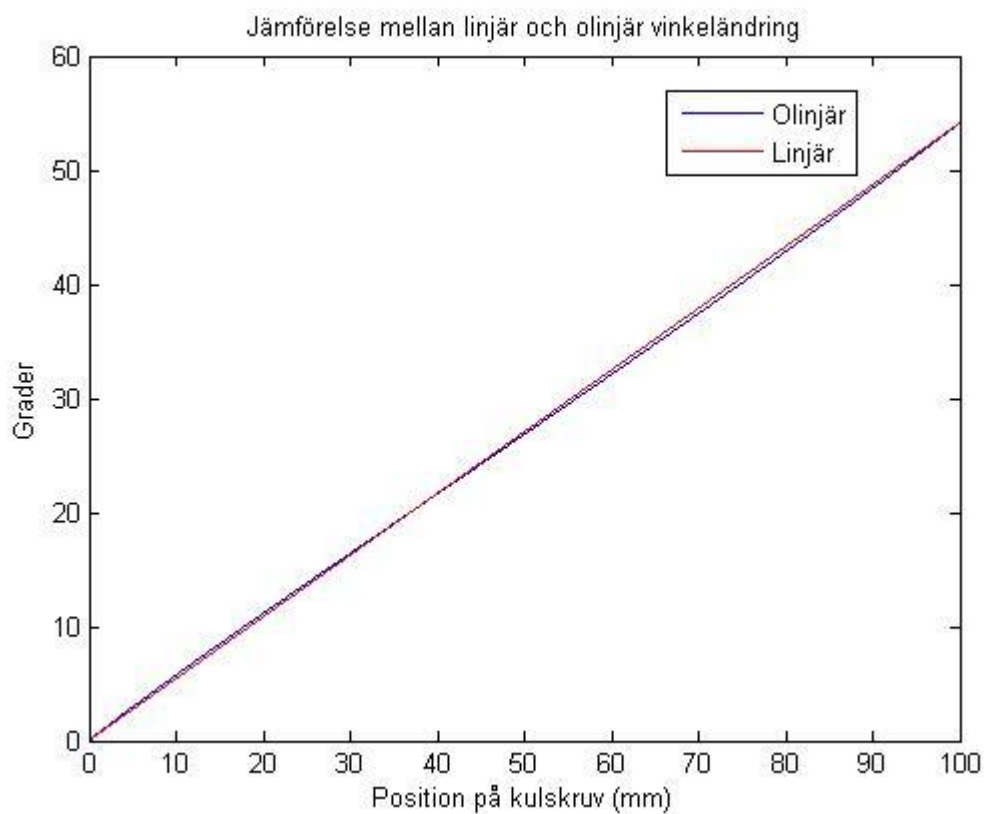
18: Thomas, B. (2008) Dimensionering av analoga reglersystem. I *Modern Reglerteknik*, red. Björn Magnusson, s. 191. Stockholm: Liber.

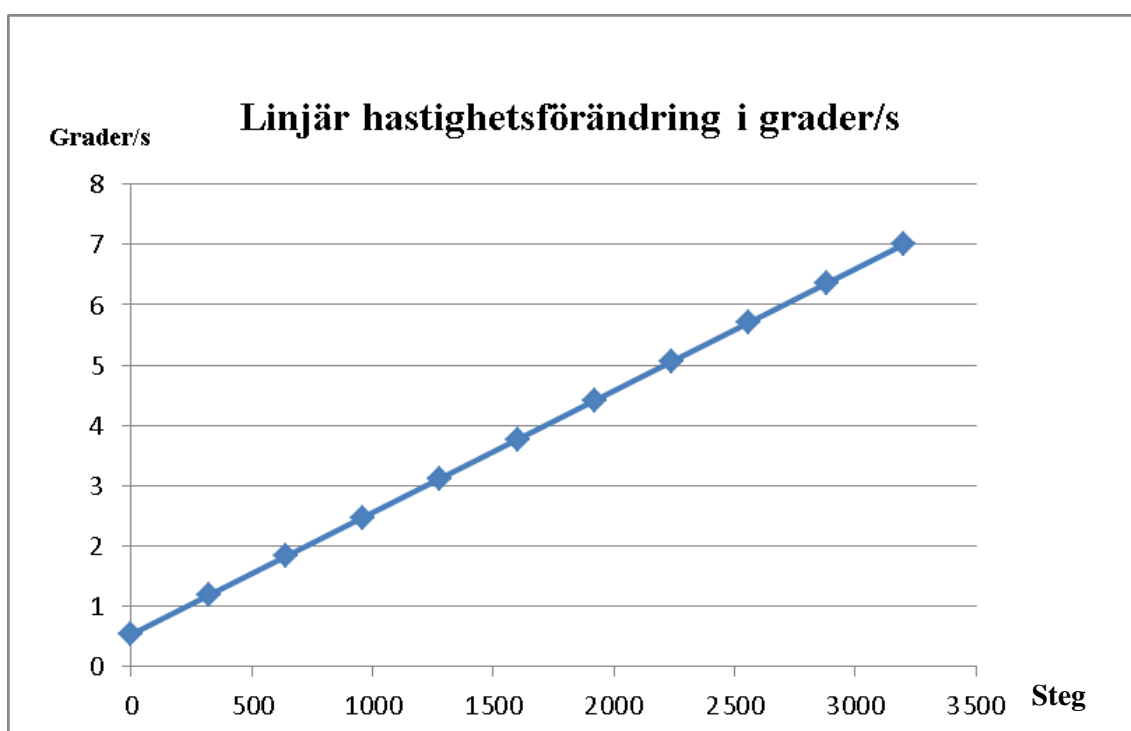
7.3 Personreferenser

19: Martin Schreuder, LightHouse Maritime Competence Center, <http://www.lighthouse.nu>

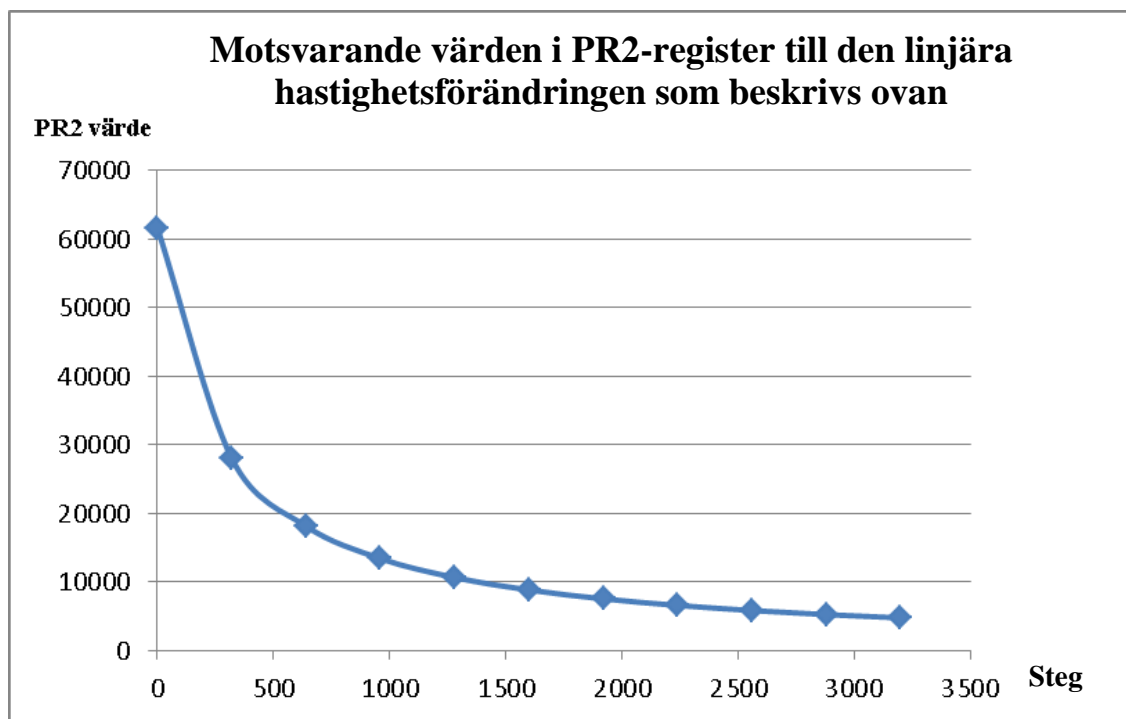
Kopplingschema



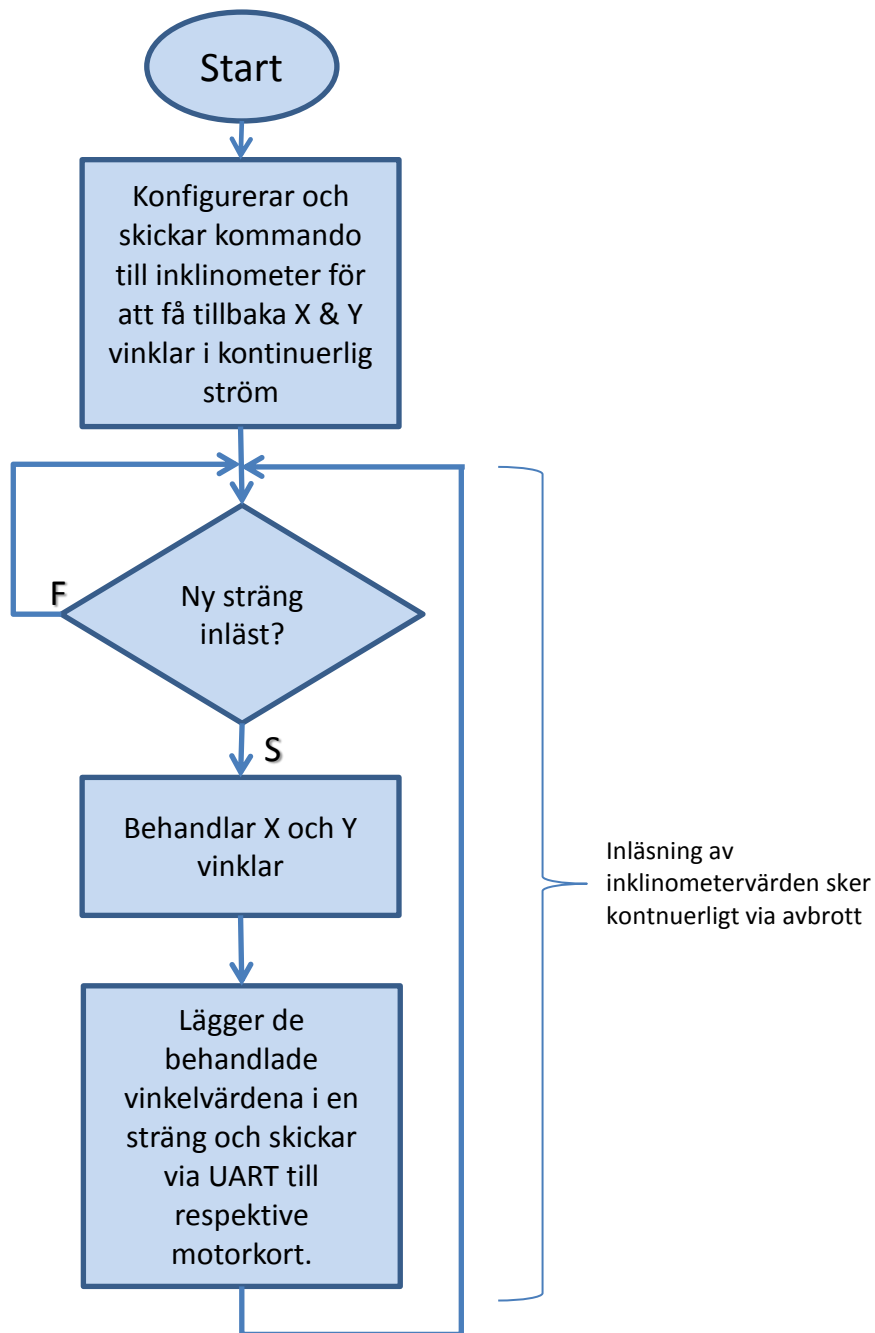




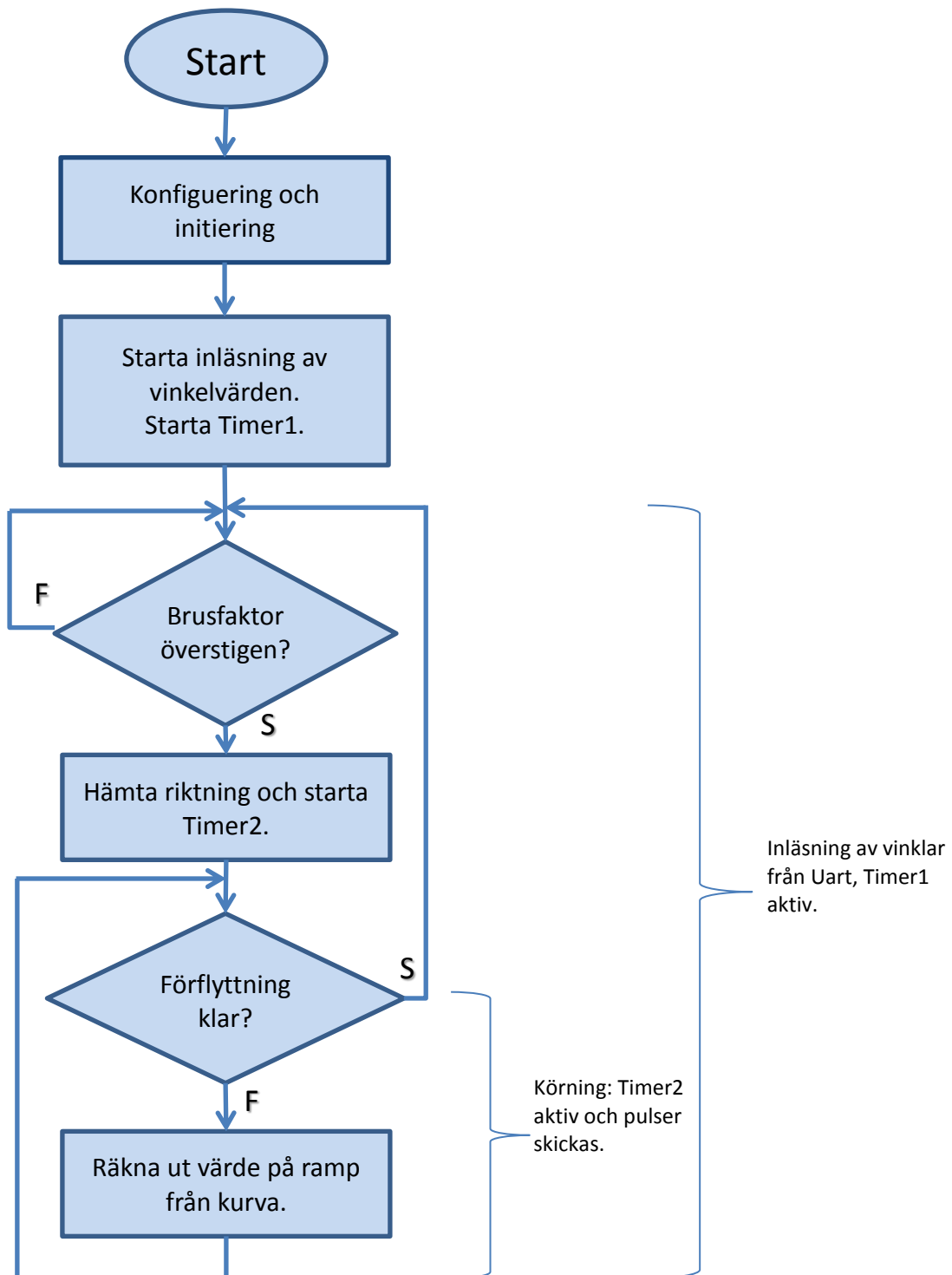
Figur 3.1. Illustration över den linjära hastighetsförändringen. För att motorn ska uppnå den maximala hastighet som motsvarar 7 grader/s på delplattformen måste den stega 3200 steg.



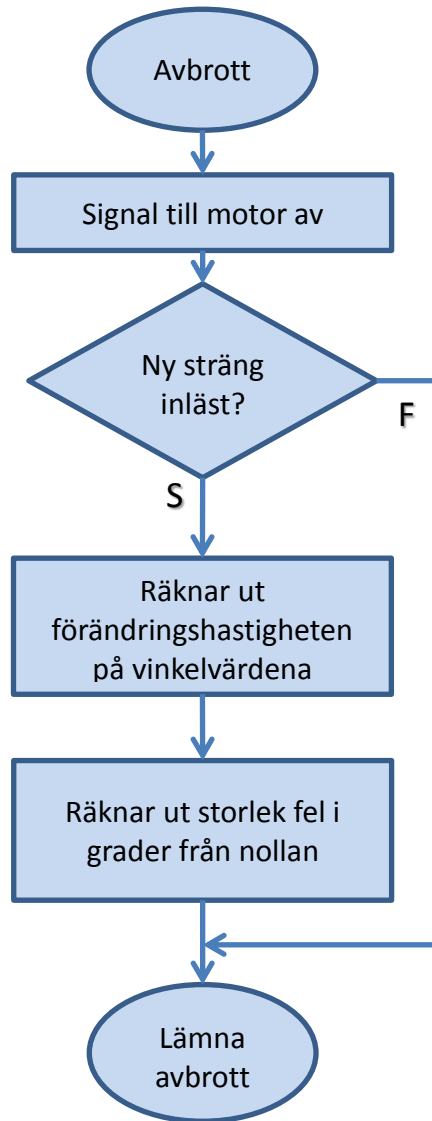
Figur 3.2. Illustration över de motsvarande värdena i PR2-registret för den linjära hastighetsförändringen ovan. Här visas hur värdet i PR2-registret ändrar sig i takt med att stegen ökar. Desto lägre värde i PR2-registret desto snabbare skickas pulser till motorn där varje puls motsvarar ett steg. Den maximala hastigheten på 7 grader/s på delplattformen motsvarar således ett värde på ca 5000 klockcykler/steg i PR2-registret. En uppräknig i TIMER2-registret motsvarar en klockcykel.



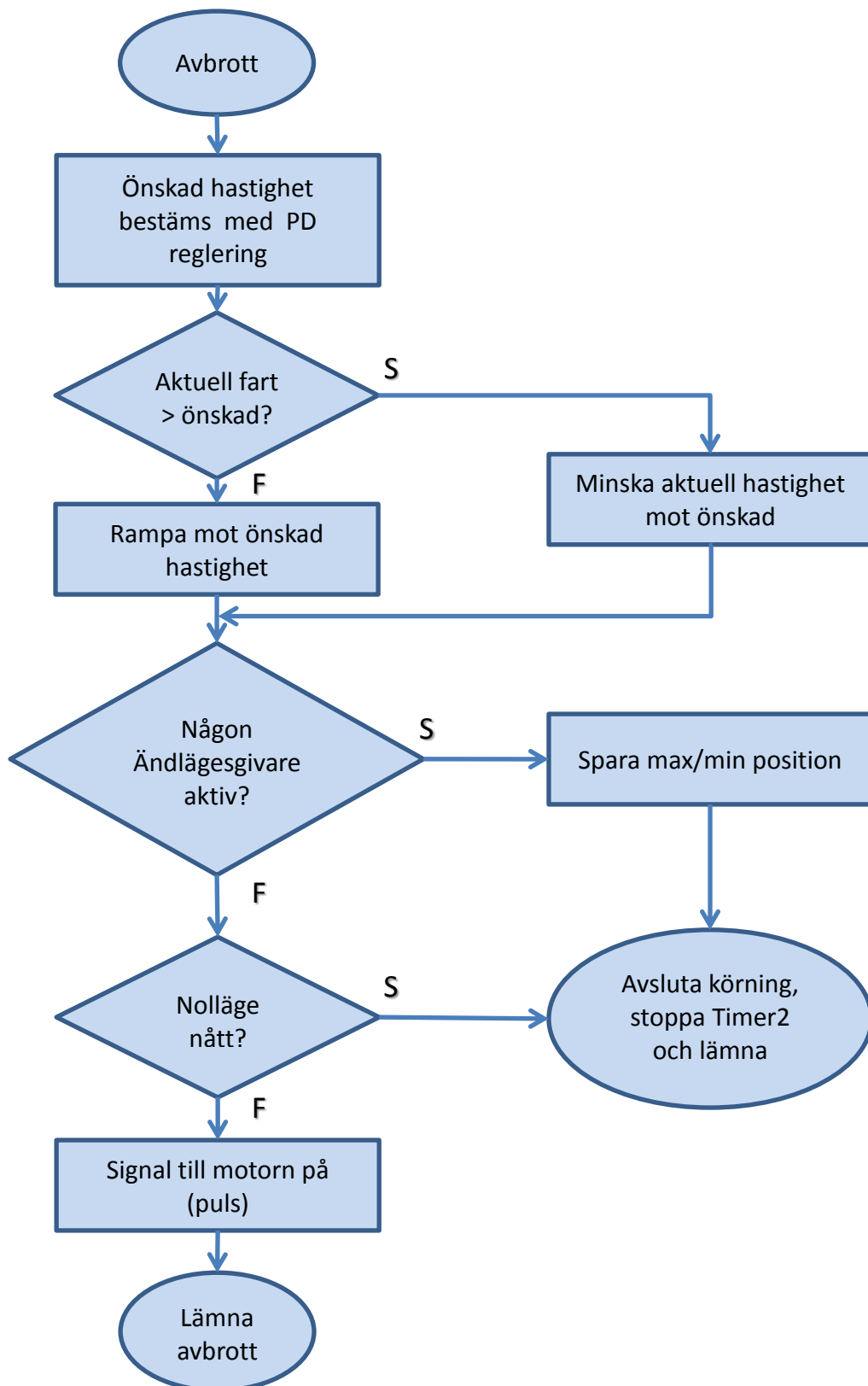
Main() i motorkort - BILAGA C. Sid C2



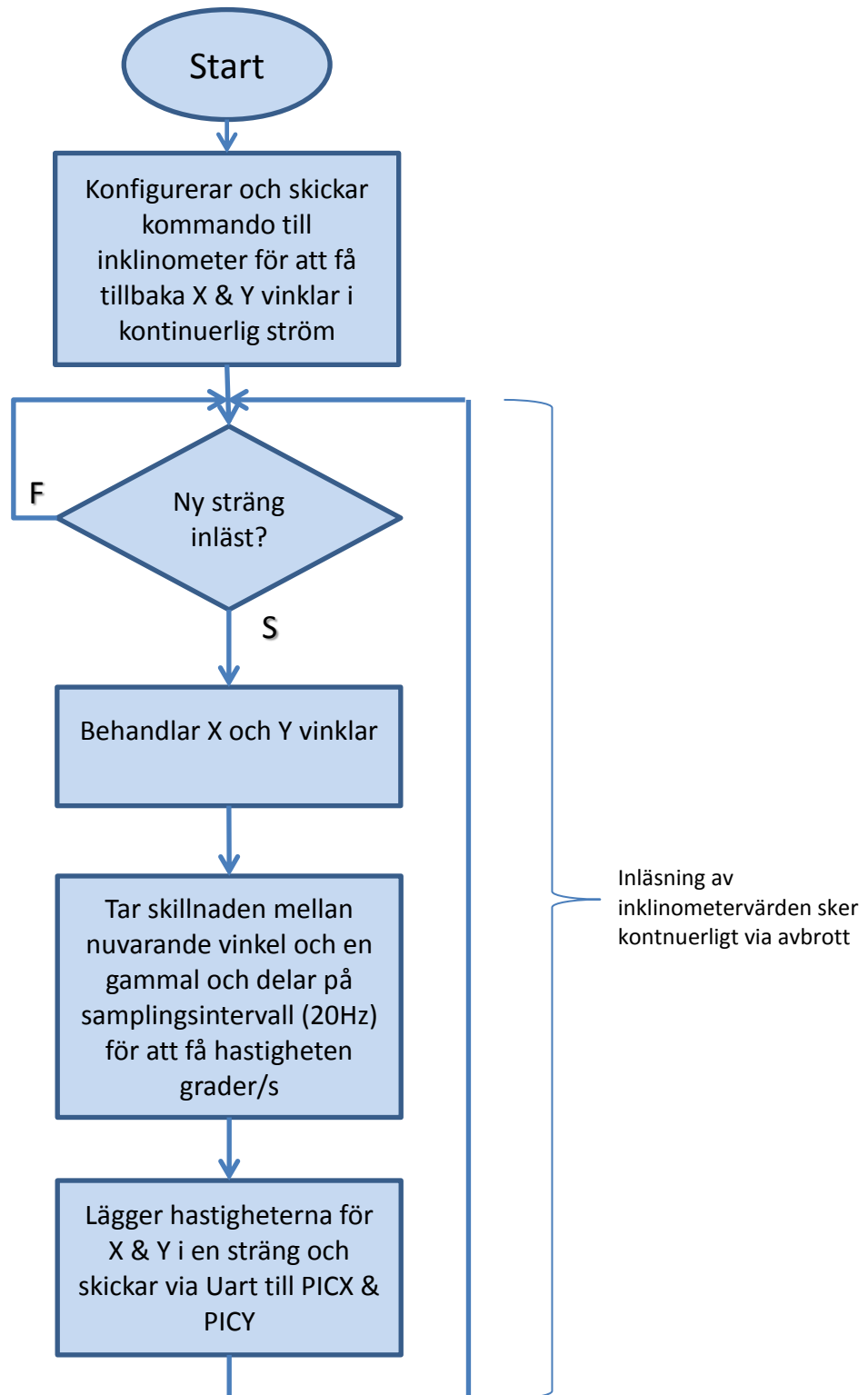
Timer1-avbrott i motorkort - BILAGA C. Sid C3



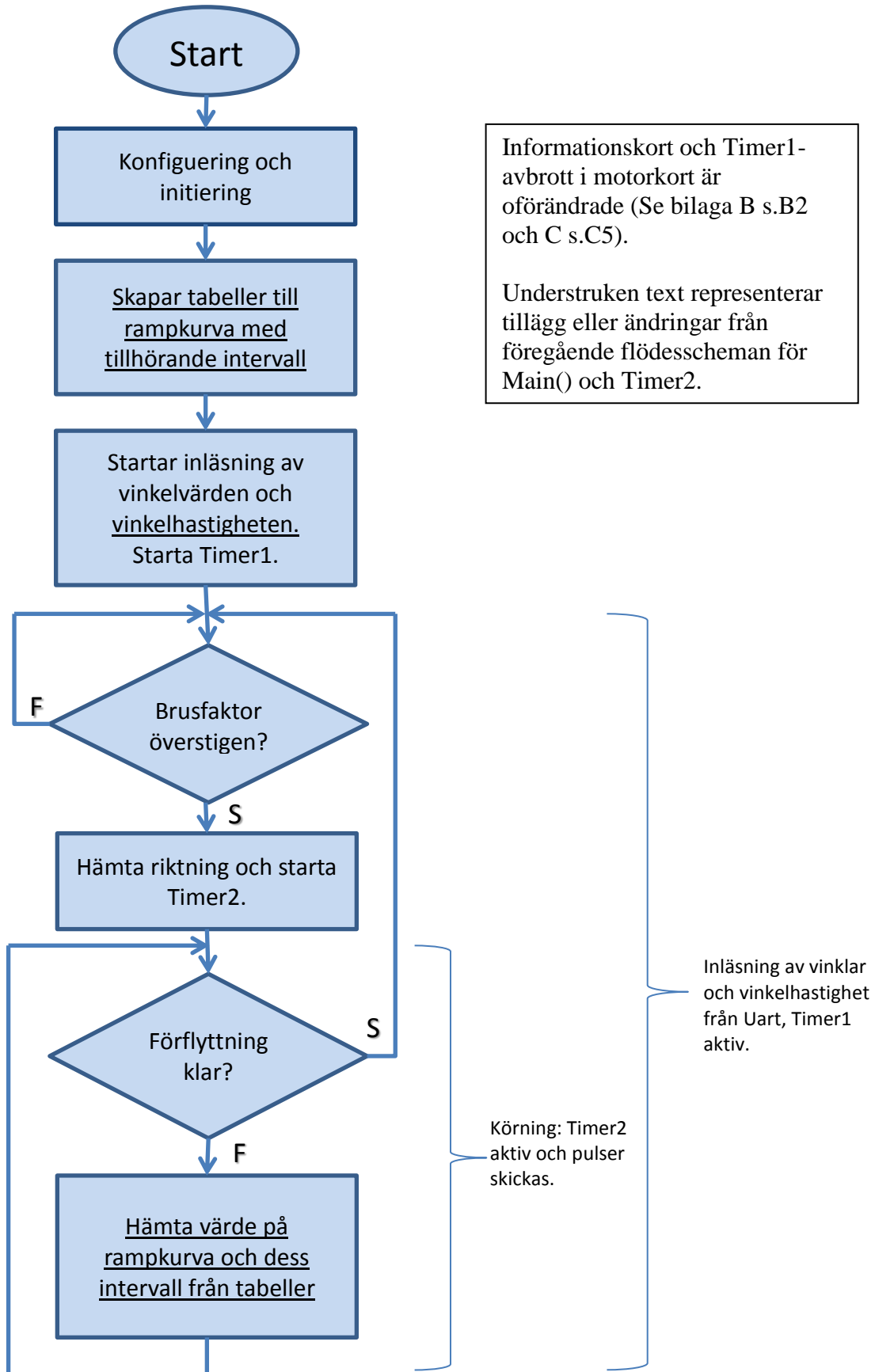
Timer2-avbrott i motorkort - BILAGA C. Sid C4



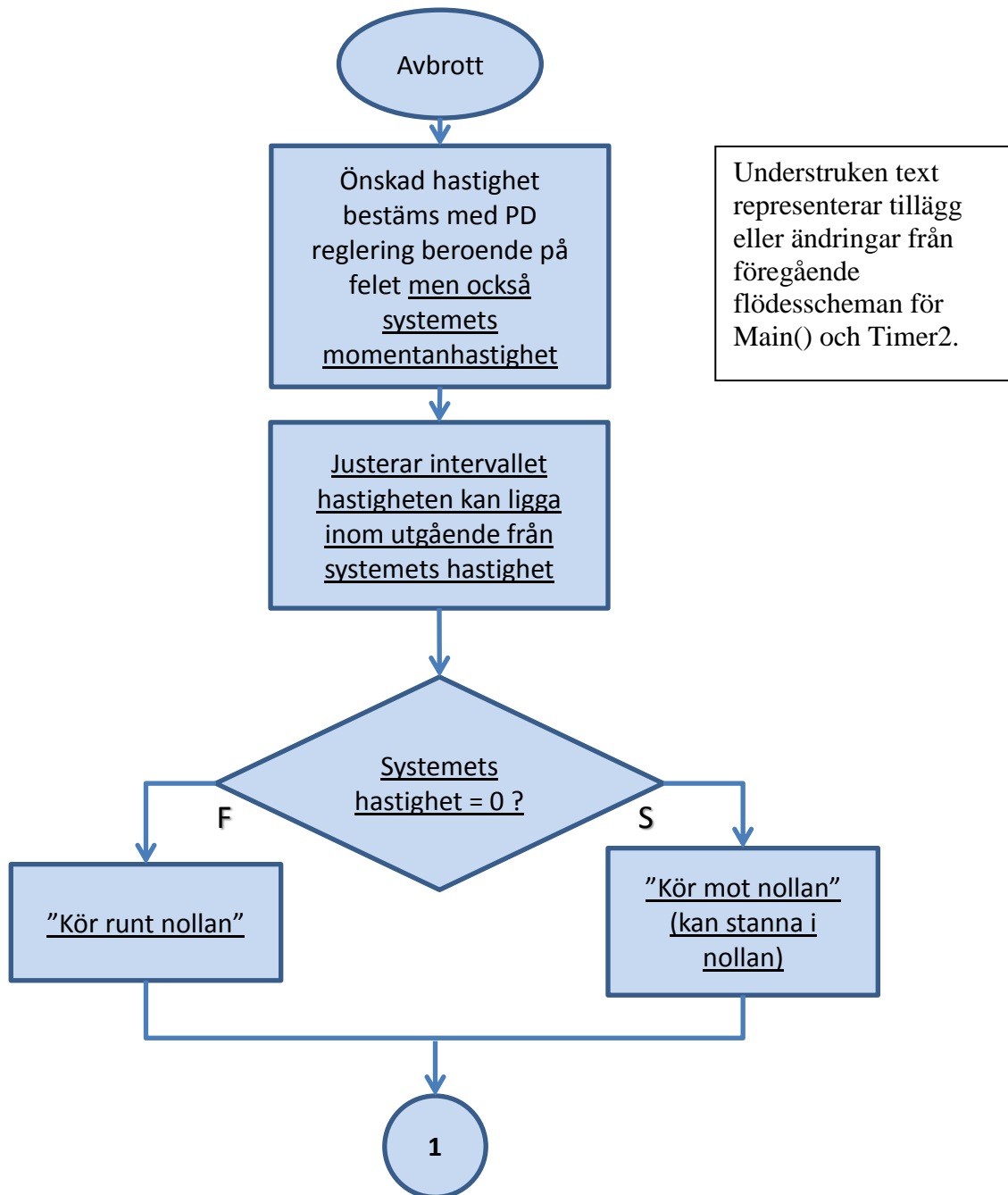
Hastighetsbestämning - BILAGA C. Sid C5

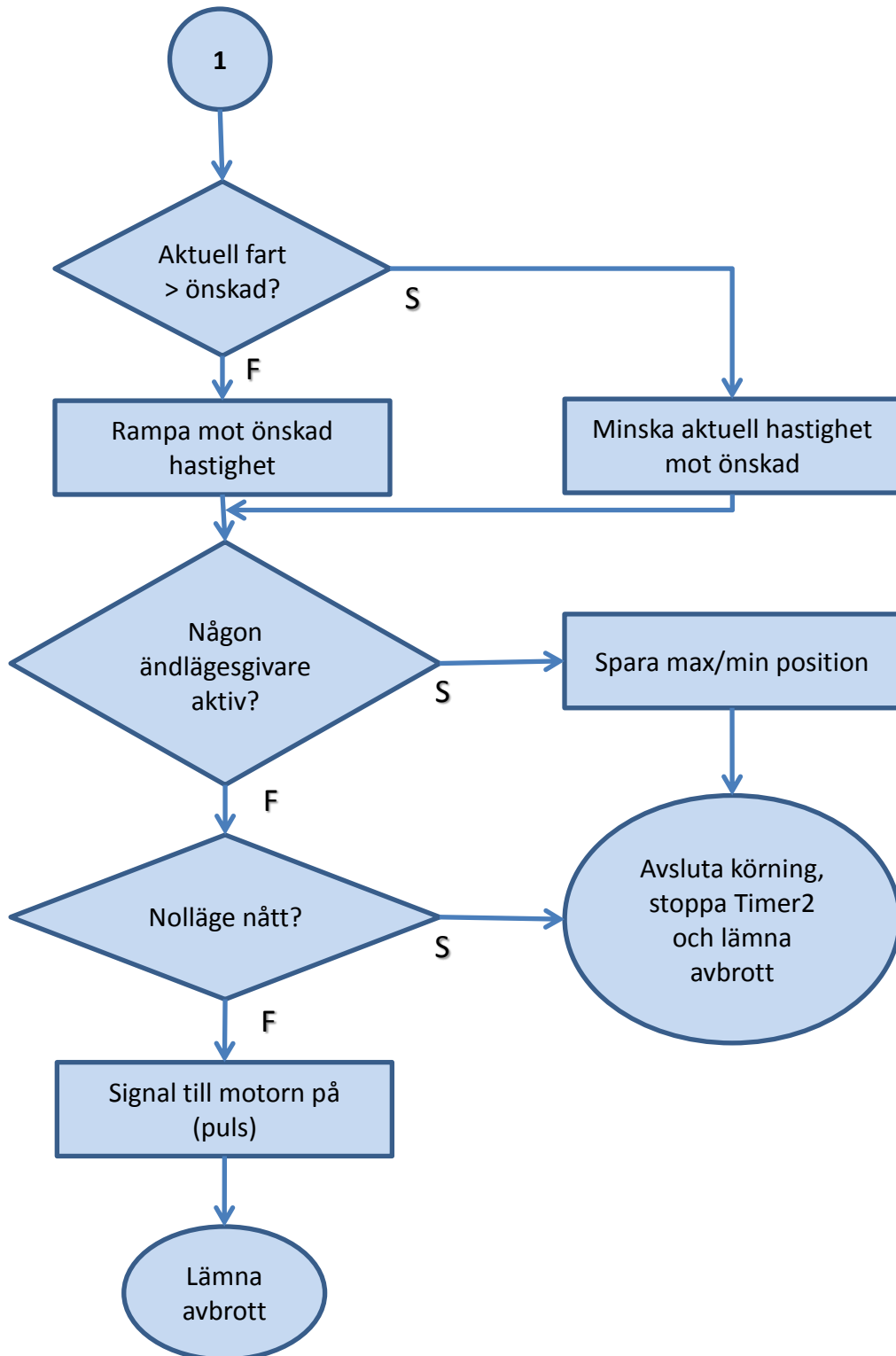


Main() i motorkort (två sensorer) - BILAGA C. Sid C6



Timer2-avbrott (två sensorer) - BILAGA C. Sid C7





Översikt av plattformens olika delar - BILAGA D. Sid D1

