



CHALMERS

System för bestämning av positionsförändringar mellan dragfordon och efterfordon

En undersökning av hur statiska magnetfält kan
användas för att bestämma inbördes
positionsförändringar

Examensarbete inom Data- och Informationsteknik

Anton Andrén

EXAMENSARBETE

**System för bestämning av positionsförändringar
mellan dragfordon och efterfordon**

En undersökning av hur statiska magnetfält kan
användas för att bestämma inbördes
positionsförändringar

Anton Andrén

Institutionen för Data- och Informationsteknik
CHALMERS TEKNISKA HÖGSKOLA

Göteborg 2016

System för bestämning av positionsförändringar mellan dragfordon och efterfordon

En undersökning av hur statiska magnetfält kan användas för att bestämma inbördes positionsförändringar

Anton Andréén

© Anton Andréén, 2016

Examinator: Peter Lundin

Institutionen för Data- och Informationsteknik
Chalmers Tekniska Högskola
412 96 Göteborg
Telefon: 031-772 1000

The Author grants to Chalmers University of Technology and University of Gothenburg the non-exclusive right to publish the Work electronically and in a non-commercial purpose make it accessible on the Internet.

The Author warrants that he/she is the author to the Work, and warrants that the Work does not contain text, pictures or other material that violates copyright law.

The Author shall, when transferring the rights of the Work to a third party (for example a publisher or a company), acknowledge the third party about this agreement. If the Author has signed a copyright agreement with a third party regarding the Work, the Author warrants hereby that he/she has obtained any necessary permission from this third party to let Chalmers University of Technology and University of Gothenburg store the Work electronically and make it accessible on the Internet.

Institutionen för Data- och Informationsteknik
Göteborg 2016

Sammanfattning

Denna rapport undersöker hur man kan använda statiska magnetfält för att skapa system vilka kan användas för att bestämma inbördes rörelse mellan dragfordon och efterfordon. Uppdragsgivaren VBG Truck Equipment AB arbetar med att utveckla kopplingssystem mellan dragfordon och efterfordon. Det finns en potential att systemet som rapporten utreder kan användas till stödsystem för chauffören. Under projektet har ett system utvecklats som använder en magnetometer för att läsa av ett statiskt magnetfält genererat av en magnet. När magneten förflyttas i förhållande till magnetometern ger detta förändringar i sensordata som kan användas för att beräkna en lägesförändring mellan magneten och magnetometern. Systemet innehåller även en beräkningsenhet som kontinuerligt läser av magnetometern och omvandlar sensordata till tillståndsvariabeln. Vidare har en kortfattad utredning gjorts för hur tillståndsvariabeln kan skickas vidare via en CAN-buss för att det ska gå att integrera i befintliga fordon. Hela systemet är utvecklat för en enkel teststation och är endast kompatibel med denna teststation som består av en enkel konfiguration av magnetiskt material som innehåller en bom vilken ger möjlighet att variera en tillståndsvariabel. Tillståndsvariabeln beräknas approximativt med en uppslagstabell.

Nyckelord: Magnetometer, statiskt magnetfält, sensor, fordonskombination

Abstract

This report investigates how magnetostatic fields can be used together with a sensor to track internal movements in vehicle combinations. The client VBG Truck Equipment AB develops hitches for trucks and there is potential for such sensors to be used in systems aiding the driver. A system has been developed that uses a magnetometer to read the magnetostatic field supplied by a magnet. When the magnet is moved in relation to the magnetometer the magnetostatic field changes. This can be used to calculate a state variable that describes the relation between the magnet and the magnetometer. The system also contains a unit that continuously reads the magnetometers values and converts them to the state variable. Furthermore, it has been investigated how the state variable can be transmitted to another node using a CAN-bus in order to be able to implement the system in existing vehicles. The system as a whole has been constructed for a test station containing a setup of magnetic material with a bar that allows a state variable to be changed. The system is only working for this test station and the specific state variable is calculated. The state variable is calculated approximatively through a lookup table.

Keywords: Magnetometer, Magnetostatic field, sensor, vehicle combination

Förord

Denna rapport är ett resultat av ett examensarbete på 15hp som utförts på dataingenjörsprogrammet (180hp) vid Chalmers Tekniska Högskola. Arbetet har utförts genom ett samarbete mellan två parter på arbetsmarknaden, VBG Truck Equipment AB som är uppdragsgivaren och EUC i Västra Götaland som har bidragit med handledning och övrigt stöd.

Jag vill tacka både de ovan nämnda parterna samt Chalmers Tekniska Högskola för möjligheten att genomföra projektet samt all hjälp jag har mottagit.

Innehållsförteckning

Sammanfattning	iii
Abstract	iv
Förord	v
Begrepp och förkortningar	1
1 Introduktion	2
1.1 Bakgrund	2
1.2 Syfte	2
1.3 Avgränsningar	2
1.4 Frågeställningar	2
2 Teknisk bakgrund	4
2.1 verktyg som använts	4
2.2 Controller Area Network (CAN)	4
2.3 Teststation	6
2.4 Utvecklingsystem	6
3 Metod	8
3.1 Arbetsprocess	8
3.1.1 Utvärdera utvecklingssystem	8
3.1.2 Bestämma sambandet mellan mätdata och vinkel	8
3.1.3 Bestämning av can struktur	10
3.1.4 Undersökning av påverkan vid vridning eller tilt av bommen	10
4 Genomförande	11
4.1 Undersökning av utvecklingssystem	11
4.2 Sveg test för att bestämma utvecklingssystem	11
4.3 Tester för att bestämma position för magneten	12
4.4 Undersökning av matematiskt samband mellan sensordata och vinkel	14
4.5 Kurvanpassning av dataserie	14
4.6 Uppslagstabell för att bestämma vinkeln	15
4.7 Extrahera data från utvecklingssystemet	15
4.8 Bearbeta kommunikationsloggen för att hitta önskad data	17
4.9 Insamling av mätdata till uppslagstabell	22
4.10 Konstruktion av den mjukvara som översätter sensordata till vinkel	22
4.11 Förberedning av data för sändning över can-buss	23
4.12 Slutgiltig version av mjukvara	23
4.13 Undersökning om hur vrid och tilt påverkar det genererade vinkelutslaget	24
5 Resultat	25
5.1 Bäst lämpade utvecklingssystemet för utveckling av sensorn	25

5.2 Matematiskt samband mellan mätdata och vinkel	25
5.3 CAN protokoll för att sända vinkelutslag till lastbil	27
5.4 Påverkan av vrid och tilt av bommen	27
6 Slutsats och kritisk diskussion	29
6.1 Resumé.....	29
6.2 Kritisk diskussion av resultat	29
6.3 Planering kontra genomförande	31
6.4 Vidareutveckling.....	31
Referenser	33
Bilagor.....	36

Begrepp och förkortningar

VBG	Uppdragsgivaren VBG Truck Equipment AB
Beräkningsenhet	Avser i denna rapport en dator som kör windows 8
Nätverksinterface	Ett gränssnitt med hård- och mjukvarulösningar som möjliggör för kommunikation mellan två eller flera noder.
Embedded Programming	Programkod som skrivs för en specifik mikrokontroller.
Datapak	En struktur för hur data ska skickas och tas emot.
Controller Area Network (CAN)	Ett nätverksgränssnitt.
Utvecklingssystem	Avser en kombination av ett eller flera kretskort som innehåller en magnetometer samt ett gränssnitt för att anslutas till en dator. Även medföljande mjukvara anses vara en del av utvecklingssystemet.
Mikrokontroller	En krets med en beräkningsenhet, minnen och potentiell övrig funktionalitet.
Sensor 2GO	Ett av utvecklingssystemen som undersökts.
RD4247MAG3110	Ett av utvecklingssystemen som undersökts.
FRDM-KL25Z	En mikrokontroller.
FRDM-FXS-MULT2-B	Ett kretskort som har tre sensorer kopplade till sig.
MAG3110	En magnetometer som finns på RD4247MAG3110 såväl som FRDM-FXS-MULT2-B.
KL25Z	Ett av utvecklingssystemen som undersökts. Detta system är en kombination av korten FRDM-KL25Z och FRDM-FXS-MULT2-B.
Magnetometer	En magnetfältssensor.
Dynamisk bredd	Utfallsrummet mellan högsta och lägsta avlästa värde hos magnetometern.

1 Introduktion

Detta kapitel ger en bakgrund till rapportens tillkomst, syfte och vilka frågeställningar den ämnar besvara.

1.1 Bakgrund

En lastbils kombination består av en lastbil och ett eller flera efterfordon. VBG Truck Equipment AB arbetar med att utveckla innovativa, tillförlitliga och säkra kopplingssystem som kopplar samman lastbil och efterfordon. Utvecklingen av dessa kopplingssystem går mot mer komplexa kombinationer. Det betyder att det ställs högre krav på chauffören och lastbils kombinationen i sin helhet. Det är önskvärt att utveckla system som underlättar och stödjer chauffören i olika moment. Exempel på ett sådant moment är backning.

En magnetometer är en sensor som läser av magnetfält [1]. Magneten avger ett magnetfält som kan avläsas av magnetometern. Beroende på var denna magnet fästes i en fordonskombination är det möjligt att magnetfältet förändras när fordonens relativa position förändras. Detta kan användas för att skapa system som producerar en önskad tillståndsvariabel.

1.2 Syfte

Studien utreder möjligheten att utnyttja statiska magnetfält för att skapa system som kan användas för bestämning av inbördes rörelser mellan dragfordon och efterfordon. Sådana system kommer att vara viktiga delar i stödsystem för att underlätta för chauffören.

Syftet är att ta fram en prototyp av ett system med en magnetometer, en magnet, en beräkningsenhet inklusive mjukvara och ett nätverksinterface som kan användas vid framtida produktutveckling av liknande system.

1.3 Avgränsningar

Ingen hårdvara konstrueras under projektet, istället kommer tre stycken utvecklingssystem innehållande varsin magnetometer att utvärderas. De utvecklingssystem som utvärderas är specificerade av uppdragsgivaren och kommer endast att utvärderas mot varandra.

Nätverket ska vara av typen CAN [2]. Då det inte finns någon hårdvara att testa nätverket mot kommer endast lämpliga befintliga protokoll för datapaketens struktur undersökas teoretiskt. Om ingen lämplig struktur existerar ska ett förslag på en ny struktur tas fram.

Den beräkningsenhet som kommer att användas är en vanlig persondator med operativsystemet Windows 8. Det finns inga krav på att programmet skall vara portabelt till andra operativsystem.

Systemet kommer att utvecklas för en teststation som simulerar ett kopplingssystem och kommer inte att anpassas för eller testas på något annat kopplingssystem. Systemet kommer även endast testas med en och samma permanentmagnet.

1.4 Frågeställningar

Den första frågeställningen som ska beaktas är vilket av de tre utvecklingssystemen som lämpar sig bäst för uppgiften. Det vill säga vilken magnetometer är bäst lämpad. Den primära aspekten som bör beaktas är vilket av korten som ger störst dynamisk bredd. En annan aspekt

som bör tas i beaktning men som inte är primär är vilket utvecklingssystem som påverkas minst av yttrestörningar.

Den andra frågeställningen som ska besvaras är hur man med hjälp av mätdata, som samlas in under projektet, kan producera tillförlitliga värden på tillståndsvariabeln för fordonskombinationen. För denna studie används vinkeln mellan magnet och magnetometer i olika konfigurationer som en tillståndsvariabel. Den väljes för att den är lätt att utvärdera. Ett program ska utvecklas för beräkningsenheten som kontinuerligt läser av och omvandlar data från utvecklingssystemet till ett aktuellt vinkelutslag.

Den tredje frågeställningen är hur man ska strukturera datapaketen som senare ska kunna skickas ut på en CAN-buss i en lastbil. Datapaketen ska innehålla den beräknade tillståndsvariabeln.

Den sista uppgiften är att utvärdera störningskänslighet. Detta görs genom att inom ett mindre intervall variera vinklar kring axlar ortogonala till den axel kring vilken sensorns huvudfunktion studeras.

2 Teknisk bakgrund

Detta kapitel redogör för de olika tekniska verktyg samt den hård- och mjukvara som använts.

2.1 verktyg som använts

Uppdragsgivaren har bestämt att en magnetometer ska användas istället för andra typer av sensorer. En magnetometer är en sensor som läser av magnetfält. Alla de magnetometrar som använts är 3-axlade magnetometrar som mäter av magnetfältets kraftvektorer i tre ortogonala riktningar x, y och z. SI-Enheten för magnetisk fältstyrka är Tesla (T) [1].

I projektet används en permanentmagnet som har ett statiskt magnetfält [3].

N
S

Figur 2.1 Skiss av magneten med polerna utmärkta.

RS232 protokollet används för den seriell kommunikation mellan utvecklingssystem och beräkningsenhet. För att den seriella kommunikationen ska fungera måste båda enheterna skicka och läsa information lika snabbt som varandra, den hastighet som information skickas och mottages i heter baud rate [4].

Termite används för att läsa av och skriva ut kommunikation som från en port på datorn [5].

Ett oscilloskop används för att läsa av spänningsförändringar över tid på en kabel eller krets [6].

C är det programspråk som används för att konstruera programvara. Programspråket C valdes då det tillåter att man arbetar nära hårdvaran och kan skriva direkt till register och portar [7]. Den utvecklingsmiljö som används är Visual Studio 2015 [8].

MATLAB används för beräkningar och visualisering av ekvationer och andra matematiska samband [9].

En uppslagstabell är ett fält som används för att ersätta beräkningar. I datorprogram används uppslagstabeller ofta istället för krångliga beräkningar för att spara processortid eller när ett värde ska matchas mot ett annat [10].

Excel används för att sammanställa data och mätserier under projektet samt för att generera grafer [11].

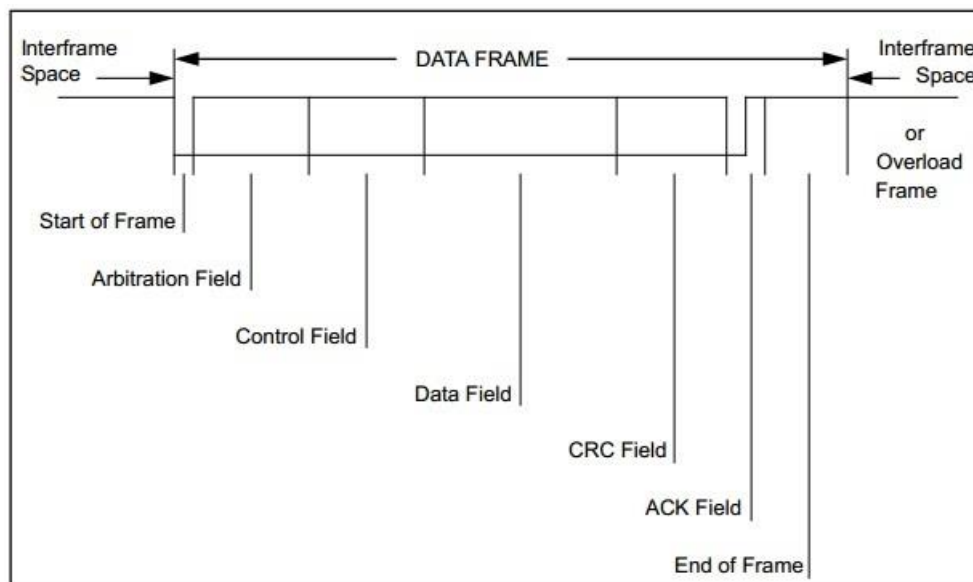
2.2 Controller Area Network (CAN)

CAN är ett protokoll för databussar som möjliggör för två eller flera noder att kommunicera mellan noderna. CAN utvecklades för fordonsindustrin men har spridit sig till andra områden [2].

Denna rapport behandlar endast CAN-meddelande i basformatet. Ett CAN-meddelande innehåller en rad ettor och nollor som tolkas av mottagaren. De olika fälten i meddelandet i ordning från första till sista är:

- Start bit – Markerar starten av meddelandet och är alltid en logisk nolla.

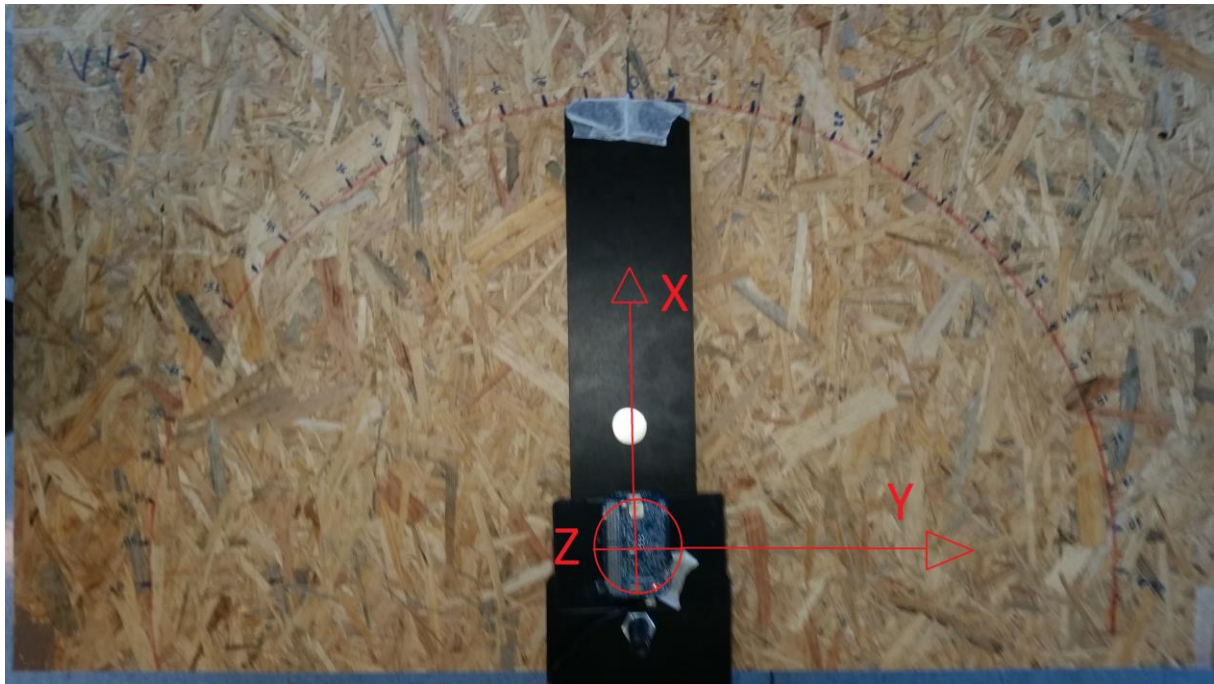
- Identifierare – Används för att beskriva typen av meddelande internt i systemet samt anger prioritet. Storleken på fältet är 11 bit.
- RTR bit – Anger om meddelande avser att sända data eller om meddelandet är en förfrågan till en annan nod.
- IDE bit – Är alltid en logisk nolla i basformatet.
- Reserverad bit – Är alltid en logisk nolla i basformatet.
- Storleken på datafältet – Anger hur stort datafältet är i bytes. Storleken på fältet är 4 bit.
- Datafält – Det fält som innehåller data som ska skickas. Storleken på fältet är noll till åtta bytes.
- CRC – Cyclic redundancy check är en uträkning på alla de föregående fälten som används för att fastställa att data inte blir korrupt vid överföring. Fältet är av storleken 15 bit.
- CRC avgränsnings bit – Är alltid en logisk etta.
- ACK bit – Är alltid en logisk etta för avsändaren. Kan sättas till logisk nolla av mottagaren.
- ACK avgränsnings bit – Är alltid en logisk etta.
- Slut bitar – Sju logiska ettor som markerar slutet av meddelandet.



Figur 2.2 CAN-meddelande struktur. Hämtat från Bosch CAN specification Version 2.0, Robert Bosch 1991.

Utöver detta utförs så kallad bit stuffing vilket innebär att om fem stycken logiska ettor eller nollor sänds i rad infogas en bit av motsatt värde förutom för CRC avgränsnings biten, ACK biten och Slut bitarna [2][12].

2.3 Teststation



Figur 2.3 Teststationen med magnetometers ortogonala axlar x , y och z utmärkta. Författarens egen bild.

Den teststation som syns i figur 2.2 har använts för att göra samtliga mätningar består av ett vanligt träbord med en skiva ovanpå. En enkel konfiguration av magnetiskt material (stål) är förankrad i skivan med en kort bom i stål vilket ger möjlighet att variera vinkelutslag. På skivan finns en halvcirkel markerad med punkter som motsvarar en förflyttning av bommen med 5 grader. De markerade punkterna går från -90 grader till +90 grader. Bommen flyttas manuellt till det önskade vinkelutslaget varpå data från utvecklingssystemet kan läsas av. På så vis kan man jämföra data från utvecklingssystemet med ett aktuellt vinkelutslag. Utvecklingssystemet kommer att placeras ovanpå den fasta delen på en plastskiva som monterats för att utvecklingssystemen ska stå stabilt. Utvecklingssystemen placeras så att y -vektorns värde förväntas vara noll då bommen är rak. En viss off-set kan förekomma om magnetometern inte sitter centrerad på utvecklingssystemet.

2.4 Utvecklingssystem

Ett utvecklingssystem utgörs av ett eller flera kretskort med tillhörande sensorer och mjukvara. Uppdragsgivaren har specificerat tre stycken utvecklingssystem som ska utvärderas under projektet. När de har utvärderats ska det system som är mest lämpligt användas under resterande delen av projektet. Alla utvecklingssystem är bestyckade med en magnetometer och ett färdigt program för Windows som kan läsa av och föra logg på sensordata vilket användes vid de första testerna. Gemensamt för alla tre är även att de kan anslutas till en dator via USB.

Det första utvecklingssystemet är Sensor 2GO. Sensor 2GO har den 3-axlad magnetometer TLV493D-A1B6 monterat på kretskortet. Utvecklingssystemet är även utrustat med mikrokontrollen XMC1100 [13].

Det andra utvecklingssystemet är RD4247MAG3110 Sensor Toolbox Tilt-Compensated eCompass Kit. Systemet innehåller den 3-axlade magnetometern. Utöver magnetometern sitter en accelerometer på kortet vilken inte kommer att användas. Bärarkortet är NXPs LFSTBUSB [14].

Det sista utvecklingssystemet är en kombination av två olika kretskort. FRDM-KL25Z är bärarkortet. FRDM-KL25Z är utrustad med mikrokontrollen MKL25Z128VLK4 MCU [15]. FRDM-KL25Z är utrustad med två portar vilka kan anslutas till en dator. Endast OpenSDA porten som använder sig av RS232 protokollet för att kommunicera via USB kommer att användas [16]. FRDM-FXS-MULT2-B är ett sensorkort med flera sensorer [17]. Magnetometern är samma som på RD4247MAG3110, alltså MAG3110. Det finns totalt tre sensorer i kombinationen men accelerometern och gyroskopet kommer inte att användas. Kombinationen av de båda kretskorten kommer härnäst att endast benämnas KL25Z.

3 Metod

Detta kapitel redogör för de metoder och hypoteser som ställts upp till grund för rapporten.

3.1 Arbetsprocess

Arbetsprocessen är indelad i fyra faser vilka motsvarar de frågeställningar projektet är grundat på. De fyra faserna är ordnade så att den viktigaste genomförs först och den minst viktiga sist.

3.1.1 Utvärdera utvecklingssystem

Under den första fasen ska de tre utvecklingssystemen utvärderas mot varandra. Samtliga tre utvecklingssystem kommer med programvara som gör det möjligt att kontinuerligt spara ner data från magnetometers x, y och z register i en logg. En rad tester kommer att utföras då loggningsfunktionalitet utnyttjas för att sedan kunna jämföra loggarna mot varandra för att hitta den bäst lämpade sensorn för fortsättningen av projektet.

Utvecklingssystem kommer ett i taget att placeras på plastbrickan på teststationen. Därefter kommer magneten att placeras på bommen rakt under utvecklingssystemet. Fyra mätningar per utvecklingssystem kommer att göras där bommen i en svepande kontinuerlig rörelse förs från ett utslag på +90 grader till -90 grader. Eftersom 90 grader är maximalt utslag i både positiv och negativ riktning på teststationen kommer hela utfallsrummet att registreras. När varje utvecklingssystem har genomgått fyra mätningar kommer magneten att flyttas två centimeter längre ut på bommen och fyra nya mätningar görs. Totalt testas fyra olika positioner för magneten, varje gång förflyttas den två centimeter ut på bommen.

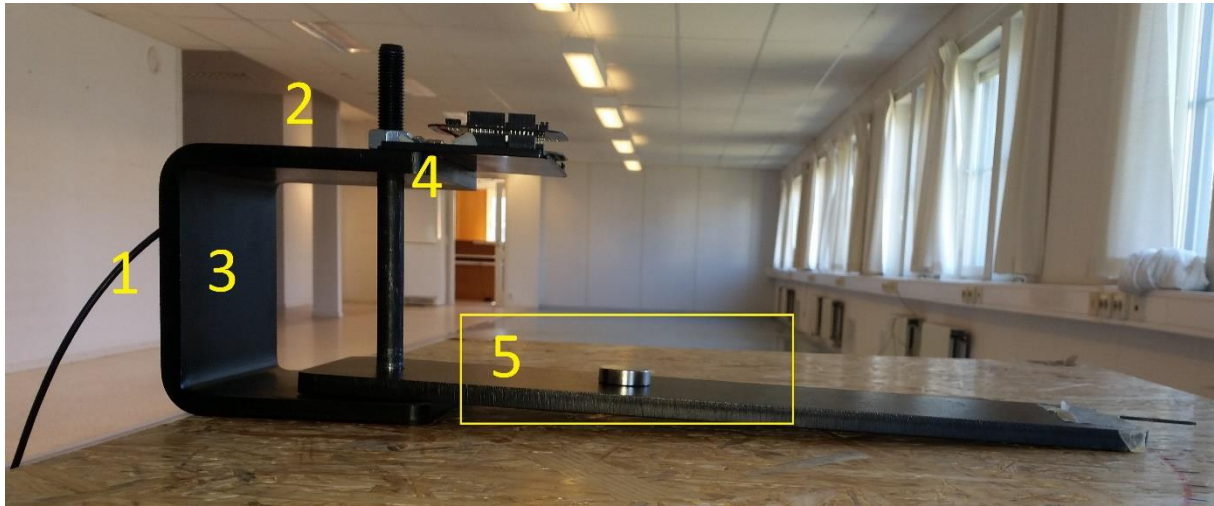
Anledningen till att testet använder en svepande rörelse är för att sensorerna måste kunna anpassa sig snabbt om fältstyrkan och dess riktning ändras, vilket inte simuleras om man gjorde avläsningar när bommen är stationär. Anledningen till att fyra tester görs på varje distans för magneten är för att minimera inverkan av den mänskliga faktorn då svepet utförs.

När alla sexton tester utförts för varje sensor kommer mätdata analyseras och sensorn med störst dynamisk bredd väljas till vidareutvecklingen av systemet. De övriga två sensorerna kommer efter detta inte användas.

Om testerna inte visar vilket utvecklingssystem som är bäst lämpat kommer en teoretisk jämförelse att göras där specifikationerna för varje system jämförs och det kraftfullaste i egenskap av dynamisk bredd och processorkraft väljs för vidareutveckling.

3.1.2 Bestämma sambandet mellan mätdata och vinkel

Efter att det mest lämpliga utvecklingssystemet har utsetts krävs det att mer noggranna mätningar görs för att bestämma sambandet mellan mätdata och vinkel. Dessa nya mätningar görs genom att bommen placeras stilla på ett av de på teststationen angivna vinkelutslag och sedan läses värdena från sensorn av. Dessa nya mätserier kommer innehålla motsvarande sensorvärden för vinkelutslag mellan -90 grader till +90 grader med 5 graders intervall.



Figur 3.1 Magnetens position. Siffrorna motsvarar de i listan över olika positioner. Författarens egen bild.

Dessa nya mätningar kommer att göras med magneten placerad på olika positioner som illustreras i figur 3.1 för att få bästa dynamik. Följande positioner kommer testas

1. Centrerat på baksidan av den fasta delen.
2. Bakom bulten på ovansidan av den fasta delen.
3. På insidan av den fasta delen, på den bakre väggen
4. På insidan av den fasta delen, rakt under utvecklingssystemet
5. På bommen rakt under utvecklingssystemet, och sedan flyttas den två centimeter ut på bommen per test. Detta upprepas fyra gånger tills att den ligger som högst åtta centimeter ifrån ursprungspositionen.

Positionerna är valda för att i någon mån kartlägga vilka konfigurationer som kan vara användbara i framtida produktupplägg.

För varje position görs det två mätserier för att återigen minska den mänskliga faktorns inverkan.

Om det visar sig att bäst dynamik uppnås om magneten ligger på bommen ska ytterligare ett test göras då magneten ligger mitt emellan de två bästa resultaten. Det bästa resultatet av dessa tre väljs för vidare utveckling.

När positionen har valts görs även ett test då hela teststationen vrids 90 grader relativt sin ursprungsposition. Detta för att undersöka hur stor inverkan jordens magnetfält har på mätningarna.

Ett enkelt matematiskt samband för sensordata och vinkel har ställts upp av uppdragsgivaren. Ekvation 3.1 beskriver sambandet där α är vinkeln i grader för bommen, F_y och F_z är y-vektorn respektive z-vektorn från magnetometern vilka anges i enheten uT.

$$\alpha \sim \arctan\left(\frac{F_y}{F_z}\right)$$

Ekvation 3.1 formel för vinkeln av resultant vektorn

Om ekvation visar sig felaktig eller om den producerar resultat med dålig tillförlitlighet bör andra möjligheter utforskas. En sådan möjlighet är att en uppslagstabell kan användas för att

omvandla sensordata till vinkel. Teststationen är endast utrustad för att mäta av vinklar med fem graders intervall vilket gör att uppslagstabellen blir approximativt. För att försöka få större noggrannhet kan en kurvanpassning av mätdata göras. Om kurvanpassningen är tillförlitlig kan den sedan användas för att producera en uppslagstabell med mindre intervall än de ovan nämnda fem graderna.

När behandlingen av mätdata är klar konstrueras mjukvara som kontinuerligt läser av data från utvecklingssystemet och omvandlar det till ett aktuellt vinkelutslag.

Programspråket C kommer att användas vid konstruktion av mjukvaran för att med enkelhet kunna läsa och skriva direkt till den port som utvecklingssystemet är ansluten till [8].

3.1.3 Bestämning av can struktur

En teoretisk undersökning ska genomföras där befintliga protokoll och standarder för CAN-buss kommunikation utreds. Om en lämplig paketstruktur för CAN-meddelandet existerar bör denna användas för att det ska vara enklare att integrera i en lastbils befintliga system.

Om det inte finns någon lämplig paketstruktur för meddelandet fastställs en ny tillsammans med uppdragsgivaren VBG.

3.1.4 Undersökning av påverkan vid vridning eller tilt av bommen

Eftersom det redan är etablerat vilken position magneten skall sitta på i systemet krävs det nu att man gör nya mätserier där man varierar vridning och tiltning av metall bommen. Eftersom aktuella utslag är under sex grader bör det göras tester för en, två, tre, fyra och fem grader. Detta ska göras i båda riktningarna för både vridning och tiltning. Två mätserier per fall mäts upp.

4 Genomförande

Detta kapitel beskriver arbetsgången för projektet samt hur svaren till rapportens frågeställningar har producerats.

4.1 Undersökning av utvecklingssystem

Utvecklingssystemen kräver att vissa drivrutiner finns installerade på datorn till vilken de är anslutna till för att kommunikationen mellan de båda enheterna ska fungera. Vissa av drivrutinerna installerades automatiskt. Andra kunde hittas via leverantörernas hemsidor. Även den medföljande programvaran till varje sensor installerades. Detta gick problemfritt för Sensor 2GO och RD4247MAG3110 systemen. KL25Z fungerade däremot inte och det undersöktes om det var drivrutinerna på datorn som inte fungerade. Efter att både OpenSDA v1.x och OpenSDA v2.x hade prövats konstaterades det att felet låg hos utvecklingssystemet. Fokus flyttades till utvecklingssystemet och efter en uppdatering till den senaste versionen av mjukvara till FRDM-KL25Zs samt att drivrutinen för OpenSDA v1.x installerats på datorn fungerade samtliga utvecklingssystem.

När de första testerna gjordes visade det sig att Sensor 2GO och RD4247MAG3110 loggade data i ett format Excel har stöd för. KL25Z loggade data i .csv format. Excel stödjer också .csv formatet. Men i det senare fallet var kolumnerna kommaseparerade och komma användes även för att ange decimaler vilket medförde att kolumnerna hamnade snett. För att fortfarande kunna läsa ut väsentliga data som fanns i dessa .csv filer ställdes det manuellt in att Excel skulle separera kolumner efter komma genom att öppna filen i texteditor och lägga till "sep=," på en egen rad högst upp. Problemet kvarstod dock att även decimalerna blev separerade. Istället för att spendera tid på att lösa problemet lades magneten nära sensorn för att generera ett kraftigt utslag som enkelt kunde identifieras i Excel även om kolumnerna hade fått felaktigt separerade decimaler.

De intressanta värdena som fanns i loggarna var de x, y och z värden som motsvarade magnetometerns avlästa värden.

4.2 Slep test för att bestämma utvecklingssystem

När alla tre utvecklingssystemen fungerade tillsammans med sin medföljande mjukvara gjordes de första testen vid teststationen. Testet sattes upp som planerat genom att utvecklingssystem ett i taget placerades på plastskivan med magneten liggande på bommen. Men istället för att placera ut magneten och sedan testa alla sensorer innan den flyttades gjordes alla tester för en sensor i taget. Detta för att slippa flytta på utvecklingssystemen mellan tester vilket spara tid och säkerställde att utvecklingssystemen stod i samma position under alla test.



Figur 4.1 Bild av testuppställningen med KL25Z. Bommen sveptes från vänster till höger sett ur det här perspektivet. Författarens egen bild.

Magnetens position mättes ut med en linjal. Utvecklingssystemen placerades i mitten av plastskivan vilket kontrollmättes med en linjal. När utvecklingssystemet och magneten var i position märktes positionerna där de båda befann sig med en penna för att kunna återskapa testet ifall något skulle gå fel. Markeringen för magneten användes senare när den första serien tester var färdig för att garantera att den befann sig i samma position som under föregående test.

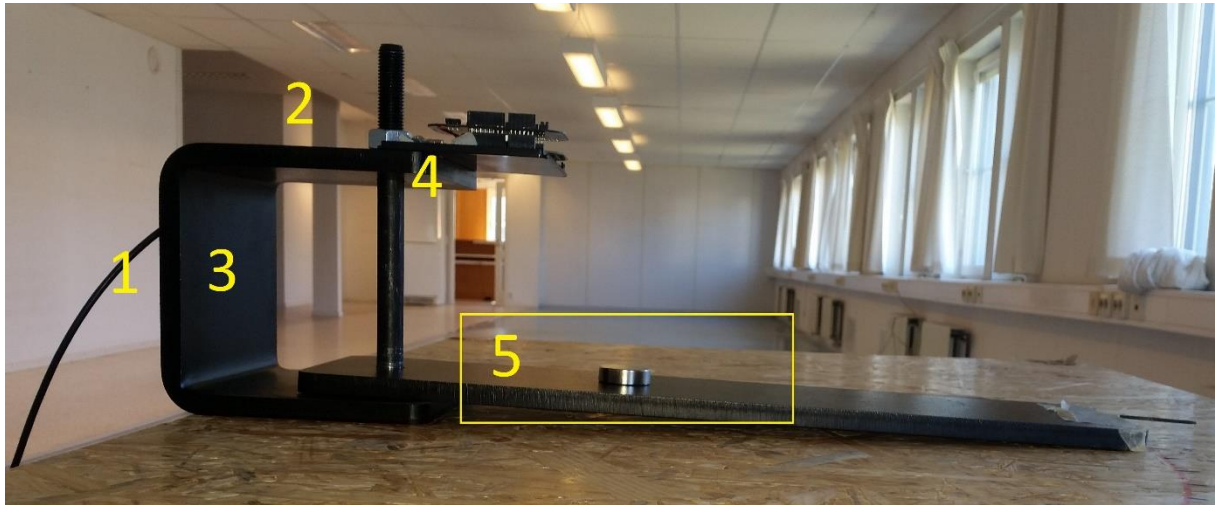
Eftersom bommen vrids manuellt under svepande rörelsen från +90 grader till -90 grader är det inte troligt att testerna utfördes i exakt samma hastighet. För att försöka vrids bommen i samma hastighet varje gång och därigenom ge de olika utvecklingssystemen samma förutsättningar användes musik med en stadig takt.

4.3 Tester för att bestämma position för magneten

Efter att KL25Z visat sig vara den sensor som gav störst dynamisk bredd behövde nya mätserier göras. Den position som bestäms baseras på när y vektorn får bästa dynamik.

Mätserierna utfördes med punktavläsningar istället för att göra en svepande rörelse. Bommen placerades på ett känt gradtal och sedan lästes värdena för magnetfältets vektorer ut visuellt ur den programvara som följt med utvecklingssystemet och antecknades manuellt. Det uppstod här ett problem eftersom det fanns yttre störningar som störde magnetometern vilket gjorde att den decimal som värdena angavs med varierade. Detta kunde orsaka att värdet på sensordata pendlade mellan två olika heltal och decimalen var omöjlig att läsa av. För att få så pålitlig data som möjligt beslutades det att ignorera decimalerna och om det pendlade mellan två heltal skulle det alltid avrundas uppåt. Eftersom att värdena varierar från minus några

hundra uT till upp över 700 uT så påverkar inte decimalen utfallet något avsevärt oavsett. Detta gjordes konsekvent i alla mätserierna.



Figur 4.2 Magnetens position. Siffrorna motsvarar de i listan över olika positioner. Författarens egen bild.

Efter att det bekräftats att det fanns små yttre störningar vid mätningarna gjordes mätningar med magneten i de positioner som illustreras i figur 4.2 i enighet med metodkapitlet 3.1.2:

1. Magnetens placerades på baksidan av den fasta delen av teststationen.
2. Magnetens placerades ovanpå den fasta delen av teststationen, men bakom utvecklingssystemet och bulten som håller fast bommen.
3. Magnetens placerades på insidan av den fasta delen av teststationen på den bakre väggen.
4. Magnetens placerades på insidan av den fasta delen av teststationen under utvecklingssystemet.
5. Magnetens placerades på bommen på så vis att den var på mitten av bommen på bredden och utflyttad sex respektive åtta centimeter ut på bommen från den punkt som är rakt under utvecklingssystemet. Detta motsvara de tidigare tester som gjorts för att bestämma vilket utvecklingssystem som var bäst lämpat.

För test 1-4 visade sig att dynamiken var för dålig och att signalen påverkades för mycket av brus för att det skulle vara möjligt att gå vidare med de uppställningarna.

Snittvärden för alla vektorerna x , y och z för varje position användes för att minimera potentiell mänsklig inverkan på mätdata. Testen med magneten liggande på bommen gav alla fina resultat med god dynamik för såväl x som y vektorn. Det visade sig att x och z vektorernas kurvform efterliknade varandra och därför togs ett beslut att använda den position där y vektorn får bäst dynamik eftersom att ekvationen från uppdragsgivarens då kan undersökas för både resultanten av y och x vektorn såväl som y och z vektorn.

Bäst dynamik hade testet där magneten var utflyttade sex eller åtta centimeter på bommen från den punkt som var rakt under utvecklingssystemet.

Ytterligare ett test gjordes där magneten placerades i en position mitt mellan de två föregående bästa mätningarna, alltså sju centimeter utflyttade längs bommen. Detta test visade sig producera det bästa resultatet.

Det togs ett beslut att inte prova att hitta bättre dynamik genom att minska avståndet som magneten flyttades ut på bommen till mindre intervall än en centimeter av två anledningar. Den första är att positionerna mätts ut för hand med en linjal och det skulle möjligen gå att krympa intervallet till en halv centimeter men efter det blir det alldeles för små marginaler om intervallet blir mindre än så. Den andra anledningen var tidsbrist.

Det gjordes ett kontroll test med hela teststationen roterat 90 grader till moturs gentemot sin ursprungsposition. Testet genomfördes för att kontrollera om systemet påverkas av rotationen så att det kan beaktas vid framtida prototypupplägg.

Vid granskning av insamlad data från kontrolltestet upptäcktes att alla tre vektorer påverkats av att teststationen hade roterats. Påverkan var som störst för x vektorn.

4.4 Undersökning av matematiskt samband mellan sensordata och vinkel

Uppdragsgivarens föreslagna ekvation beräknar vinkeln av resultanten mellan y-vektorn och z-vektorn. Eftersom att z-vektorn och x-vektorn har liknande beteende testades även att använda x-vektorn istället för z-vektorn i formeln. Även om sensorn centrerats i förhållande till vinkeln noll grader på teststationen kan magnetometern vara något sned i förhållande till teststationen. Eftersom att detta borde ge en stadig förskjutning av värden producerade av ekvationen ignorerades detta.

Beräkningen genomfördes på en serie där snittet av de fyra mätserier sammanställts för att minska inverkan av den mänskliga faktorn. Ekvationen för hur vinkeln kan beräknas är felaktig och felmarginalen är upp till 149 grader då y och x vektorerna används och 43 grader då y och z vektorerna användes. Orsaken till felmarginalerna har inte kunnat fastställas. Eftersom att beräkningarna inte gav tillfredställande resultat beslutades det att en uppslagstabell ska användas eftersom att detta genererar en mindre felmarginal på upp till fem grader.

4.5 Kurvanpassning av dataserie

För att få mer precis data till uppslagstabellen kan man mäta upp nya mätserier med mindre intervall mellan gradtalen. Eftersom att det är väldigt tidskrävande samt att teststationen endast är utmärkt med fem graders intervall gjordes en kurvanpassning i MATLAB för mätserien för att hitta ett polynom för att minska felmarginalen. Polynomet ger sensorvärdet beroende på vinkeln. Detta möjliggör att mindre intervall kan användas för att bestämma sensorvärdet vilket ger en mer noggrann uppslagstabell.

MATLAB har de fördefinierade funktionerna polyval och polyfit vilka används för kurvanpassning data. Polyfit använder sig av Vandermonde matrisen för att bestämma polynoms koefficienter [18]. Polyval används för att utvärdera polynomet i varje punkt

specificerad [19]. Y-vektorn och z-vektorn användes för att göra approximation. X-vektorn lämnades utanför eftersom att den påverkats mest när teststationen roterats.

```
// d and d2 represent the angle
d = -90:5:90;
d2 = -90:1:90;

// Enter the dataserie
x = [-128 ... -126];

// Find the polynomial's coefficients (the number is the grade and can be changed for a better fit)
px = polyfit(d,x,10);

// Evaluate the poly with more assertion spots
px2 = polyval(px, d2);
```

figur 4.3 MATLAB kod för kurvanpassning av en dataserie för x. Samma metod kan användas för y och z.

Kurvanpassningen gav inga bra resultat och därför fick de uppmätta värdena med fem graders intervall användas till uppslagstabellen.

4.6 Uppslagstabell för att bestämma vinkeln

En uppslagstabell används där data från sensorn direktöversätts till ett vinkelutslag med ett statistiskt fält som innehåller en översättning. Det finns dock en komplikation eftersom att vissa värden för z-vektorn motsvarar två vinklar. Detta löses genom att kombinera y och z vektorernas data och två olika uppslagstabeller som matchas motvarandra för att hitta det korrekta utfallet. Detta kan göras då y-vektorn har värdet 0 uT i samma punkt som z-vektorn når sin maxpunkt. Utfallsrummet för z-vektorn delades in i två uppslagstabeller där det ena motsvarar positiva vinklar och det andra negativa i förhållande till att bommen är rak.

4.7 Extrahera data från utvecklingssystemet

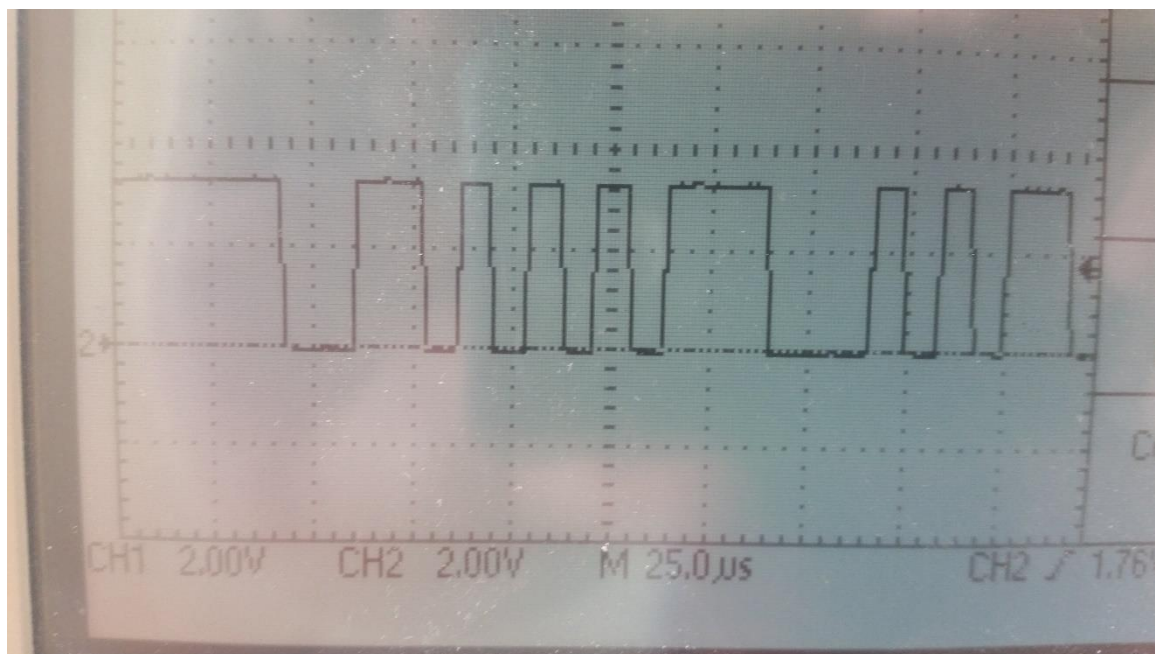
För att kunna använda sensordata i systemet måste den kunna läsas av en extern enhet. Eftersom att utvecklingssystemet skickar data med RS232 protokollet måste ett program utvecklas som kan läsa av den kontinuerligt. För att ett sådant program ska fungera måste baud rate, hur många stop bitar som används och information om paritet vara känt eftersom detta är parametrar för den seriella kommunikationen. Portläsnings programmet Termite användes för att undersöka om utvecklingssystemet skickade data i ASCII format genom att

alla olika standard inställningarna iterativt undersöktes men utan att ett läsbart meddelande kunde hittas.



Figur 4.4 Termite skärmdump

Då data inte skickas i ASCII format är det troligt att det skickas som rådata och då måste parametrarna för den seriella kommunikationen vara kända för att man ska kunna tolka dataströmmen. Ett oscilloskop kopplades till USB porten och ett meddelande fångades upp som syns i figur 4.5. Ur meddelandet kunde baud rate utläsas till 115200 Hz, storleken på meddelandet utläsas till 1 byte, det används en stop bit och meddelandet skickades utan paritetsbit.



figur 4.5 Meddelande fångat med oscilloskop. Författarens egen bild.

Nu kunde ett C program utformas för att spara mätdata i en text fil. Noterbart är att USB porten som används för den seriella kommunikationen öppnas som en fil [20].


```

// Handle block; open the com port stream as a "file"
HANDLE handle;
handle = CreateFile(L"COM8",
    GENERIC_READ | GENERIC_WRITE,
    0,
    0,
    OPEN_EXISTING,
    FILE_ATTRIBUTE_NORMAL,
    0);
if (handle == INVALID_HANDLE_VALUE) {
    if (GetLastError() == ERROR_FILE_NOT_FOUND) {
        //serial port does not exist. Inform user.
        printf("Error 1: Serial port does not exist / is not
            conected\n");
    }
    //some other error occurred. Inform user.
    printf("Error 2:\n");
}

```

figur 4.6 Kod för att öppna porten som en fil. Den port som används i figuren är COM8.

Sedan sätts parametrarna för den seriella kommunikationen, vilket illustreras i figur 4.7 [21].

```

// DCB block; set parameters for the communication
DCB dcb = { 0 };
dcb.DCBlength = sizeof(dcb);

if (!GetCommState(handle, &dcb)) {
    //error getting state
    printf("Error 3: error getting state \n");
}

dcb.BaudRate = CBR_115200;
dcb.ByteSize = 8;
dcb.StopBits = ONESTOPBIT;
dcb.Parity = NOPARITY;
if (!SetCommState(handle, &dcb)) {
    //error setting serial port state
    printf("Error 4: error setting serial port state\n");
}

```

Figur 4.7 C kod för att sätta kommunikations parametrar.

ReadFile används för att läsa av porten och fprintf tillsammans med en filpekare används för att spara ner kommunikationsloggen i ett textdokument [22][23]. Loggen sparade data hexadecimalt då varje byte alltid representerar två tecken vilket gör det mer överskådligt.

4.8 Bearbeta kommunikationsloggen för att hitta önskad data

När mätdata kunde sparas ner i en logg fanns ett mönster som upprepade sig frekvent. Det var förekomsten "7e 7e" följt av 01-07 som gjorde att paketstrukturen snabbt kunde identifieras.

```

7e 7e 01 07 80 f8 ce 17 d8 ff 14 01 e4 20 2a ff 4b fe dc ff ff
ff fe ff 00 00 f3 3d c1 00 c7 01 74 63 08 01 7e 7e 02 08 f4 01
89 8e 7e 7e 03 09 80 f8 ce 17 03 00 fd ff 03 00 7e 7e 04 0a 80
f8 ce 17 12 00 02 00 89 04 7e 7e 05 0b 80 f8 ce 17 ca b4 ff ff
10 09 7e 7e 06 0c 1b 00 10 27 af fe 00 00 00 00 00 00 7e 7e 07
0d 3d 00 d8 ff 01 00 05 00 fd ff 00 00 41 00 e1 ff fb ff 05 00
fe ff 00 00 da ff 27 00 f5 00 f5 fd 12 00 04 00 3c ff ff 7f ff
7f 00 80

```

Figur 4.8 Osorterat meddelande från seriellport

```

7e 7e 01 07 80 f8 ce 17 d8 ff 14 01 e4 20 2a ff 4b fe dc ff ff
ff fe ff 00 00 f3 3d c1 00 c7 01 74 63 08 01

7e 7e 02 08 f4 01 89 8e

7e 7e 03 09 80 f8 ce 17 03 00 fd ff 03 00

7e 7e 04 0a 80 f8 ce 17 12 00 02 00 89 04

7e 7e 05 0b 80 f8 ce 17 ca b4 ff ff 10 09

7e 7e 06 0c 1b 00 10 27 af fe 00 00 00 00 00 00

7e 7e 07 0d 3d 00 d8 ff 01 00 05 00 fd ff 00 00 41 00 e1 ff fb
ff 05 00 fe ff 00 00 da ff 27 00 f5 00 f5 fd 12 00 04 00 3c ff
ff 7f ff 7f 00 80

```

Figur 4.9 Sorterat meddelande från seriellport

Efter det kördes utvärderingsprogrammet som följde med KL25Z för att i tur och ordning ställa in mikrokontrollen på FRDM-KL25Z kortet att bara skicka information från en av tre sensorer åt gången. Ett mönster visade sig snabbt och det konstaterades att magnetometer informationen skickades på totalt sex byte. Detta verifierades ytterligare genom att en magnet fördes närma sensorn var på dessa sex bytes varierade kraftigt i värde.

Det visade sig även att paketstrukturen varierade vilket gjorde att förskjutningen mellan starten på meddelandet och magnetometerdata varierade med ett byte. Detta löstes genom att signalen för att ställa in mikrokontrollen på att bara skicka data från magnetometern lästes av och återskapades. På så vis kunde de sex byte som motsvarar data från en annan sensor, vilka alltid nu motsvaras av värdet 00, användas som ett mönster för att hitta magnetometerdata.

7e 7e 01 0e c0 94 cf 17	e4 ff 10 01 b0 20	2c ff 53 fe e8 ff	01 00 01
00 ff ff	f1 3d cc 00 d1 01 75 63 08 01		
7e 7e 02 0f f4 01 1f 8a			
7e 7e 03 10 c0 94 cf 17 03 00 09 00 02 00			
7e 7e 04 11 c0 94 cf 17 13 00 02 00 89 04			
7e 7e 05 12 c0 94 cf 17 cc b4 ff ff 10 09			
7e 7e 06 13 1b 00 10 27 ae fe 00 00 00 00 00 00			
7e 7e 01 01 c0 57 36 89	00 00 00 00 00 00	30 ff 48 fe d3 ff	00 00 00
00 00 00	b2 3e 00 00 00 00 01 63 06 01		
7e 7e 02 02 f4 01 82 07			
7e 7e 03 03 c0 57 36 89 00 00 00 00 00 00			
7e 7e 04 04 c0 57 36 89 00 00 00 00 81 04			
7e 7e 05 05 c0 57 36 89 d0 bc ff ff a4 09			
7e 7e 06 06 11 00 10 27 c0 ff 0a ff 70 fe 2a 00			

Figur 4.10 Magnetometerdata isolerat i meddelande från seriellport när de båda andra sensorerna är på i den övre rutan och av i den nedre. Magnetometerdata markerat med grönt och data från de andra sensorerna med blått. 30 ff motsvarar x-vektorn, 48 fe motsvarar y-vektorn och d3 ff motsvarar z-vektorn.

Eftersom att storleken på varje datapaket som skickas från utvecklingssystemet varierar sparas tillräckligt många bytes undan i en buffert för att garantera att data från magnetometern samt de mönster som används för att isolera den fångas in. Detta för att garantera att magnetometerdata kan hittas vid varje genomsökning av bufferten.

Genom tester kunde de bestämmas vilken data som tillhörde vilken variabel x, y och z. Testet gjordes genom att data med ett känt värde lästes av.

```

while (i_bytebuffer_idx < 150) {
    dword_Bytes_Read = 0;

    // Read up to 150 bytes from the COM port and store them in bytes_read
    i_read = ReadFile(handle, bytes_read, 150, &dword_Bytes_Read, NULL);

    // If the stream was read successfully
    if (i_read)
    {
        i_bytes_read_idx = 0;

        // Loop through the bytes_read and store them in a buffer
        // untill 150 bytes are stored.
        // This is done to ensure that the magnetometer data is
        // captured
        while (dword_Bytes_Read)
        {
            bytearray[i_bytebuffer_idx] =
            bytes_read[i_bytes_read_idx];

            i_bytes_read_idx++;
            i_bytebuffer_idx++;
            --dword_Bytes_Read;
        }
    }
}

```

Figur 4.11 C kod som sparar tillräckligt med data i en buffer för att garantera att magnetometerdata kan hittas

Sedan söks bufferten igenom och data från magnetometern isoleras och kan behandlas. Magnetometerdata skickades som totalt sex byte, två för varje vektor x, y och z. Men ingen primitiv variabeltyp av storleken två byte motsvarar ett decimal tal. Ett test gjordes där värdet på varje fältvariabel skrevs ut och det visade sig snabbt att det inte var en variabeltyp som hanterar decimaler utan en short som divideras med tio för att åstadkomma decimalen. För att omvandla indata från rådata till en short lagras två bytes för att sedan med logisk OR och bitskiftning skapa en short. Vilken bit som var den mest respektive minst signifikanta biten fastställdes via provning mot kända värden.

```

// Loop through the databuffer to find the desired magnetometer data
// The loop condition ensures that the program doesn't read out of bounds data
for (i_bytebuffer_idx = 0; i_bytebuffer_idx < 125; i_bytebuffer_idx++)
{
    // Find the message containing the magnetometer data
    // The message is identified by the pattern 0x7e 0x7e 0x01
    if (bytebuffer[i_bytebuffer_idx] == (char)0x7e &&
        bytebuffer[i_bytebuffer_idx + 1] == (char)0x7e &&
        bytebuffer[i_bytebuffer_idx + 2] == (char)0x01 )
    {
        // Find the pattern for the accelerometer sensor and the gyroscope
        // The 6 byte pattern is allways 0x00 0x00 0x00 0x00 0x00 0x00 for both of
        // the above sensors
        if (bytebuffer[i_bytebuffer_idx + 8] == (char)0x00 &&
            bytebuffer[i_bytebuffer_idx + 9] == (char)0x00 &&
            bytebuffer[i_bytebuffer_idx + 10] == (char)0x00 &&
            bytebuffer[i_bytebuffer_idx + 11] == (char)0x00 &&
            bytebuffer[i_bytebuffer_idx + 12] == (char)0x00 &&
            bytebuffer[i_bytebuffer_idx + 13] == (char)0x00 &&
            bytebuffer[i_bytebuffer_idx + 20] == (char)0x00 &&
            bytebuffer[i_bytebuffer_idx + 21] == (char)0x00 &&
            bytebuffer[i_bytebuffer_idx + 22] == (char)0x00 &&
            bytebuffer[i_bytebuffer_idx + 23] == (char)0x00 &&
            bytebuffer[i_bytebuffer_idx + 24] == (char)0x00 &&
            bytebuffer[i_bytebuffer_idx + 25] == (char)0x00 )
        {
            //Read the corresponding magnetometer data into short x, y, z
            ...
            // The magnetometer data has been found. Proceed with the
            // next reading by canceling the loop.
            i_bytebuffer_idx = 150;
            tmp_flag = 1;
        }
        // If the pattern wasn't found try finding it one byte
        // further into the message
        else if (bytebuffer[i_bytebuffer_idx + 9] == (char)0x00 &&
            bytebuffer[i_bytebuffer_idx + 10] == (char)0x00 &&
            bytebuffer[i_bytebuffer_idx + 11] == (char)0x00 &&
            bytebuffer[i_bytebuffer_idx + 12] == (char)0x00 &&
            bytebuffer[i_bytebuffer_idx + 13] == (char)0x00 &&
            bytebuffer[i_bytebuffer_idx + 14] == (char)0x00 &&
            bytebuffer[i_bytebuffer_idx + 21] == (char)0x00 &&
            bytebuffer[i_bytebuffer_idx + 22] == (char)0x00 &&
            bytebuffer[i_bytebuffer_idx + 23] == (char)0x00 &&
            bytebuffer[i_bytebuffer_idx + 24] == (char)0x00 &&
            bytebuffer[i_bytebuffer_idx + 25] == (char)0x00 &&
            bytebuffer[i_bytebuffer_idx + 26] == (char)0x00 )
        {
            // Read and convert the corresponding magnetometer data into
            // short x, y, z
            ...
            // The magnetometer data has been found. Proceed with the
            // next reading by canceling the loop.
            i_bytebuffer_idx = 150;
            tmp_flag = 1;
        }
    }
}

```

Figur 4.12 C kod som isolerar magnetometerdata I bufferten

```
mag_x = (bytebuffer[i_bytebuffer_idx + 14] | bytebuffer[i_bytebuffer_idx + 15] << 8);  
mag_y = (bytebuffer[i_bytebuffer_idx + 16] | bytebuffer[i_bytebuffer_idx + 17] << 8);  
mag_z = (bytebuffer[i_bytebuffer_idx + 18] | bytebuffer[i_bytebuffer_idx + 19] << 8);
```

Figur 4.13 C kod som konverterar 2 bytes till en short. Koden motsvarar de tre punkterna i if satsen i figur 4.12. Samma teknik används men med annan offset för punkterna i if else satsen.

4.9 Insamling av mätdata till uppslagstabell

Vid tidigare datainsamlingar hade endast mjukvaran som följde med utvecklingssystemen använts vilket gör det svårt att avgöra om någon kalibreringsalgoritm körs på datorn. Därför gjordes två sista mätserier som ska ligga till grund för uppslagstabellen med den mjukvara som utvecklats under projektet. För att minska felmarginalen används snittvärdet av de två tabellerna.

4.10 Konstruktion av den mjukvara som översätter sensordata till vinkel

Eftersom mjukvaran konstruerats så att magnetometerdata sparas ner i motsvarande variabler `mag_x`, `mag_y` och `mag_z` som motsvarar vektorerna x , y och z kunde lösningen med uppslagstabellen implementeras. Lösningen innebär att värdet på y -vektorn undersöks och om det är positivt anropas en uppslagstabell som översätter z -vektorns värde till ett gradtal, om y -vektorns värde är negativt anropas en annan uppslagstabell med samma funktion. För att hitta motsvarande vinkel för ett värde på z -vektorn har ett fält skapats där värdet på z -vektorn kan användas som index för att hitta det önskade värdet.

För att programmet inte ska få oönskat beteende kan inte värden för z -vektorn som ligger utanför intervallet $-90 < \text{grader} < 90$ användas. Detta säkerställs genom att värdet på z -vektorn undersöks innan programmet översätter det med uppslagstabellen. Om värdet inte ligger inom det tillåtna intervallet görs ingen översättning.

Programmet representerar det aktuella vinkel utslaget med två variabler. Den första variabeln anger den lägsta gränsen för det intervall vinkeln ligger inom och den andra anger den högsta. Eftersom att värdena är uppmätta på samma sätt som programmet bör detta intervall som högst motsvara en felmarginal på <5 grader.

```

// Calculate the angle using the mag_y and mag_z data
// Reset the values
angle_start = 0;
angle_end = 0;

// Check that the value for mag_z is in an allowed interval
if (((mag_y < 0 && (mag_z / 10) > 93) || (mag_y > 0 && (mag_z / 10) > 134))
    && (mag_z / 10) < 650)
{
    // If mag_y < 0 the corresponding lookup table is lookup_negative_angle
    // which returns the start of the
    // interval for the angle. Add 5 to receive the end of the interval.
    if (mag_y < 0)
    {
        angle_start = lookup_negative_angle(mag_z / 10);
        angle_end = angle_start + 5;
    }
    // If mag_y > 0 the corresponding lookup table is lookup_positive_angle
    // which returns the end of the
    // interval for the angle. Subtract 5 to receive the start of the interval.
    else
    {
        angle_end = lookup_positive_angle(mag_z / 10);
        angle_start = angle_end - 5;
    }
}
else
{
    // VALUE NOT IN THE VALID RANGE
}

```

Figur 4.14 C kod som anropar rätt uppslagstabell om z-vektorn som avlästs från KL25Z har ett tillåtet värde.

4.11 Förberedning av data för sändning över can-buss

Den paketstruktur som finns för hur meddelanden ska skickas över CAN-buss lämpar sig väl för att skicka de båda variablerna av typen short som innehåller gränserna för intervallet för den beräknade vinkeln. Det fält som bestämmer storleken för data som överförs skall sättas till den binära representationen av fyra, 0100, eftersom att de två variabler av typen short som ska skickas är på två bytes vardera. Anledningen till att variabeltypen short valdes är för att man i framtiden ska kunna anpassa meddelandet till att visa en decimals noggrannhet istället för att sakna decimaler som det är i nuläget. Datafältet fylls med de två variablerna där den lägre gränsen skickas först följt av den övre. Övriga fält behandlas av CAN-interfacet.

4.12 Slutgiltig version av mjukvara

Den mjukvara som utvecklats testades på testtriggen genom att bommens position varierades samtidigt som mjukvaran skrev ut det uppskattade värdet på en skärm. Testet uppvisade önskat beteende när bommen fördes mellan gradangivelserna. Det upptäcktes nu att systemet är väldigt känsligt då komponenter flyttas och en manuell kalibrering med tre olika vinklar behövde göras. Kalibreringen genomfördes genom att bommen placerades på en av de tre vinklarna i taget varpå magneten flyttades tills att alla tre vinklar gav önskade värden för z-vektorn. Inga övriga tester utfördes.

4.13 Undersökning om hur vrid och tilt påverkar det genererade vinkelutslaget

Den första metoden som bestämdes för att beräkna påverkan vid vrid och tilt av bommen baserades på att mätserier skulle göras utvecklades innan lösningen där vinkeln presenteras som ett intervall framtagits. Det bedöms nu att det endast behöver testas hur stor påverkan systemet får för tre respektive sex graders tilt och vrid av bommen. Detta eftersom att det är troligt att felmarginalen uppslukas av marginalen som intervallet bidrar till.

När mätserierna skulle utföras upptäcktes ett problem med programvaran som gjorde att värdena för de tre vektorerna hoppade och lade sig på nya nivåer. Värdena återgick aldrig till de värden som mätts upp tidigare då bommen placerades i sin vanliga orientering. Däremot kunde värdena till synes slumpartat hoppa mellan olika nivåer flera gånger och vid flera olika orienteringar av bommen. Det gick inte heller att fastställa att samma vridning eller tiltning av bommen ledde till samma felmarginal utan värdena varierade utan sammanhang.

Mjukvaran felsöktes utan att problemet kunde hittas. Utförliga tester gjordes då bommen inte tiltades eller vreds men mjukvaran agerade då åter felfritt.

Inga test med den medföljande mjukvaran för utvecklingssystemet gjordes.

5 Resultat

Resultat kapitlet ämnar ge svar åt de fyra frågeställningar som ställts upp vid början av projektet. Nedan följer en kortfattad sammanfattning av dessa:

1. Vilket av utvecklingssystemen Sensor 2GO, RD4247MAG3110 och KL25Z är bäst lämpat för projektet?
2. Vad finns det för matematiskt samband mellan mätserier med magnetometerdata och vinkeln i grader? Detta samband ska användas vid mjukvaruutveckling av ett program som kontinuerligt läser av utvecklingssystemets sensordata och omvandlar det till ett vinkelutslag.
3. Hur ska den data som produceras av mjukvaran som utvecklats skickas på CAN-buss?
4. Påverkas det beräknade vinkelutslaget om bommen tiltas eller vrids?

5.1 Bäst lämpade utvecklingssystemet för utveckling av sensorn

För att avgöra vilket utvecklingssystem som var bäst lämpat att användas i sensorsystemet gjordes en serie tester där bommen på teststation sveptes från höger till vänster. Det gjordes fyra tester per sensor och position för magneten som varierades på fyra distanser ut med bommen. Ur de loggar som genererats under testerna fastställdes vilken dynamik utvecklingssystemen fick för var och en av x-, y- och z-vektorer. Data för sensor 2go har inte fastställts eftersom den inte genererade pålitliga värden.

	Dynamisk bredd för X vektorn (uT)				Dynamisk bredd för Y vektorn (uT)				Dynamisk bredd för Z vektorn (uT)			
KL25Z	Serie 1	Serie 2	Serie 3	Serie 4	Serie 1	Serie 2	Serie 3	Serie 4	Serie 1	Serie 2	Serie 3	Serie 4
Grundposition	228	228	221	218	409	407	403	408	589	579	575	587
Grundposition + 2cm	535	520	528	528	523	523	522	522	349	346	346	347
Grundposition + 4cm	237	235	237	234	205	205	206	205	37	37	39	37
Grundposition + 6cm	69	69	67	69	53	54	53	55	21	22	20	21

Figur 5.1 Dynamisk bredd för KL25Z

	Dynamisk bredd för X vektorn (uT)				Dynamisk bredd för Y vektorn (uT)				Dynamisk bredd för Z vektorn (uT)			
RD4247MAG3110	Serie 1	Serie 2	Serie 3	Serie 4	Serie 1	Serie 2	Serie 3	Serie 4	Serie 1	Serie 2	Serie 3	Serie 4
Grundposition	223	218	217	212	287	288	288	288	366	368	363	366
Grundposition + 2cm	415	413	403	422	433	434	434	435	241	239	237	239
Grundposition + 4cm	195	197	199	198	190	190	190	190	34	34	33	34
Grundposition + 6cm	76	75	75	76	70	69	69	69	23	23	23	22

Figur 5.2 Dynamisk bredd för RD4247MAG3110

Den bästa dynamiken i testserien hade KL25Z när magneten var placerad två centimeter ut på bommen från sin ursprungsposition rakt under sensorn. Ingen utvärdering av yttrepåverkan behövde göras för att fastställa vilket system som var bäst lämpat. Med det som grund anses KL25Z vara bäst lämpad av de tre utvecklingssystemen.

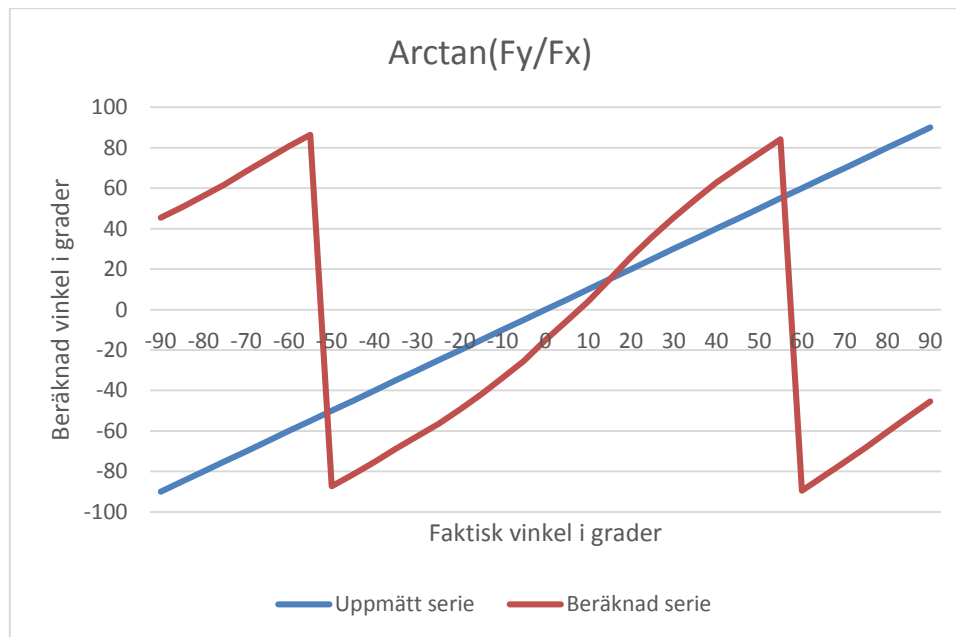
5.2 Matematiskt samband mellan mätdata och vinkel

Innan uppdragsgivarens förslag till ekvation som innebär att z- och y-vektorns resultant kan användas för att bestämma vinkeln behövdes mätserier där vinkelns förhållande till sensordata var känt. Det gjordes flera mätserier även för att fastställa detta men även för att hitta den positionen av magneten som gav bäst dynamik. När magneten var placerad på den fasta delen av teststationen uppmättes väldigt dålig dynamik och mycket brus vilket föranledde att dessa mätserier inte kunde användas. Då magneten låg på bommen uppmättes bra dynamik. Den position av magneten som valdes var den som gav bäst dynamik för y-vektorn. Detta eftersom att x- och z-vektorerna uppvisade liknande beteende när mätserierna genomfördes vilket ligger till grund för att ekvationen även testades för y- och x-vektorernas resultant.

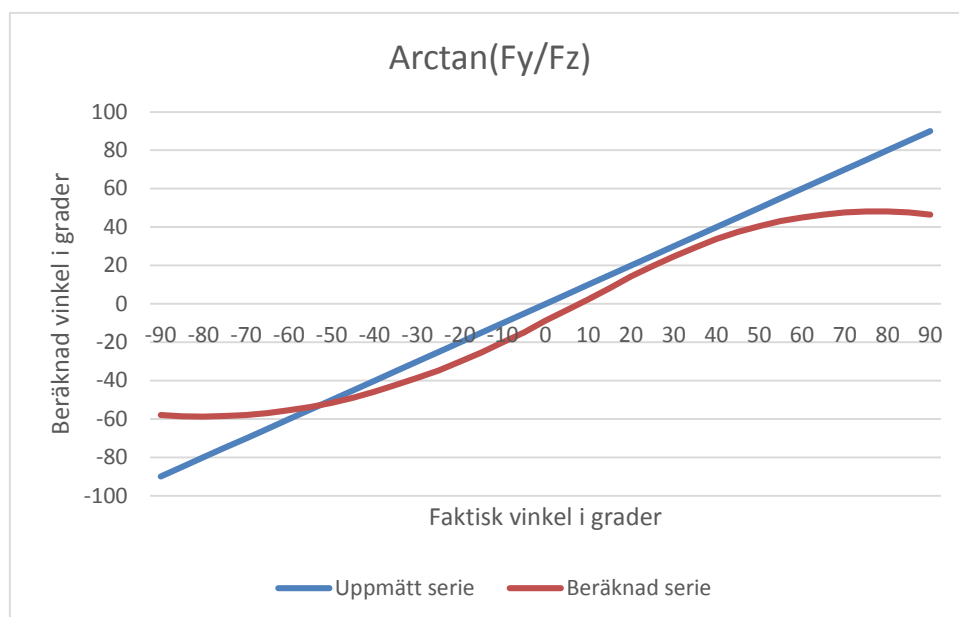
	Dynamisk bredd		
	X (uT)	Y(uT)	Z(uT)
Grundposition	230	415	601
Grundposition + 2	370	553	706
Grundposition + 4	486	628	717
Grundposition + 6	559	632	656
Grundposition + 7	520	632	610
Grundposition + 8	584	622	567

Figur 5.3 Dynamisk bredd för KL25Z vid andra omgången med tester. Värdena är avrundade till närmaste heltal.

En sammanställning som beskriver hur mycket det beräknade värdet för vinkeln skiljer sig från det faktiska återges i figur 5.4 och figur 5.5. Det framgår då att ekvationen inte producerar önskade värden eftersom att skillnaden mellan det beräknade och det faktiska värdet varierar kraftigt. Om skillnaden hade varit konstant hade det varit möjligt att det var testuppställningen som gjorde att värdena differerade. I figur 5.4, så byter x vektorn tecken vid ca -55 och ca 55 grader vilket förklarar de kraftiga förändringarna som syns i grafen.



Figur 5.4 Graf för faktiska vinkeln kontra den med ekvationen beräknade vinkeln för resultant vektorn för y och x vektorerna



Figur 5.5 Graf för faktiska vinkeln kontra den med ekvationen beräknade vinkeln för resultant vektorn för y och x vektorerna

En kurvanpassning med mjukvaran MATLAB gjordes sedan för att kunna skapa en uppslagstabell med bättre noggrannhet än de fem graders intervall som teststationen tillåter. Men det krävdes att polynomapproximationen använde sig av ordningsnummer över 10 för att kurvanpassningen skulle likna resultatet. Ett polynom med ett högt ordningsnummer är inte tillförlitligt och därför bestämdes att de värden som mätts upp var tillräckliga eftersom att de ger en relativt liten felmarginal på upp till fem grader.

Uppslagstabellen implementeras så att den beräknade vinkeln beskrivs som ett intervall där den övre och undre gränsen är kända från uppmätta dataserier. Intervalllets felmarginal påverkas av yttrefaktorer som att teststationen roteras.

Den mjukvara som utvecklats i programspråket C läser kontinuerligt av sensordata från utvecklingssystemet och konverterar det till motsvarande vinkel.

Systemet upptäcktes vara känsligt för när sensorn och magneten flyttades. När de skulle återplaceras krävdes en enkel kalibrering där tre vinklar med kända värden för z-vektorn användes för att komponenterna skulle kunna återplaceras på samma ställe.

5.3 CAN protokoll för att sända vinkelutslag till lastbil

Den basstruktur som finns för kommunikation på CAN-buss bedöms som tillräcklig för projektet eftersom att endast fyra bytes av data behöver skickas per meddelande. De fyra bytes som skickas motsvarar två variabler av typen short där den första innehåller värdet för den nedre gränsen av intervallet för vinkeln och den andra innehåller den övre.

Ingen faktisk tillämpning har kunnat implementeras på grund av brist på en faktisk CAN-buss samt information om meddelandena som skickas i en lastbil.

5.4 Påverkan av vrid och tilt av bommen

Systemet påverkas kraftigt redan vid små justeringar av vrid och tilt (under två grader) vilket gör det oanvändbart. Sensorvärdena hoppar mellan olika nivåer på ett slumpartat sätt vilket

gör att systemet endast är tillförlitligt när bommen befinner sig i den orientering som mjukvaran är utvecklad för. Ingen anledning till felet kunde fastställas.

6 Slutsats och kritisk diskussion

Detta kapitel beskriver författarens tankar om resultatet, metoden som använts samt vad som måste göras för att utveckla framtida produktionsupplägg.

6.1 Resumé

Med en magnetometer och en permanentmagnet kan en tillståndsvariabel uppskattas. Det gick inte att fastställa ett matematiskt samband mellan sensordata och vinkel vilket ligger till grund för att en approximation utförs med en uppslagstabell. Mjukvara har utvecklats som läser av sensordata och med uppslagstabellen översätts det till ett intervall på fem grader inom vilken vinkeln befinner sig. En felmarginall existerar för programmet som påverkas av faktorer som jordens magnetfält och potentiellt annan störning. En annan faktor som påverkar systemet kraftigt är om bommen tiltas eller vrids.

Den basstruktur för CAN-meddelanden som existerar kan användas för att skicka vidare värdet för tillståndsvariabeln.

6.2 Kritisk diskussion av resultat

Det system som har utvecklats kan användas för att mäta och approximera lämplig tillståndsvariabel. Systemet borde användas med försiktighet då det endast bidrar med en approximation av tillståndsvariabeln. Eftersom att systemet har en känslighet för yttre påverkan i egenskap av att det roteras respektive jordens magnetfält bör ytterligare studier göras innan en robust sensor kan skapas för praktisk tillämpning. Inflytandet av övrigt magnetiskt material i omgivningen till en sensor byggd på det studerade konceptet har inte studerats. Detta måste göras innan skarp implementation.

Det studerade konceptet kan inom sina begränsningar användas till såväl dynamiska tillståndsvariabler som till varningsfunktioner för vissa tillstånd. Dessa varningsfunktioner kan i sin tur leda till mindre olyckor i trafiken vilket är gynnsamt för miljön eftersom att fordonsskrot innehåller miljöfarliga ämnen. Det är även gynnsamt för samhället i sin helhet eftersom att det kan förhindra personskador.

När den position av magneten som gav bäst dynamik skulle bestämmas efter att KL25Z utsetts till den mest lämpade sensorn bestämdes positionen utifrån den bästa dynamiken för y-vektorn. I efterhand upptäcktes det att y-vektorn fick lika bra dynamik för positionen då magneten var utflyttad sex centimeter längst bommen men att x och z vektorerna fick mycket bättre dynamik då. Alltså borde förmodligen en annan position av magneten valts men på grund av hur uppslagstabellen är implementerad har detta en minimal inverkan på resultatet. Inte heller den ekvation som ställts upp av uppdragsgivaren skulle gynnas nämnvärt av att den bättre positionen för magneten använts. Anledningen till att systemet inte korrigerades till att använda den bättre positionen är för att det upptäcktes när projektet nästan var färdigt och det inte fanns tid för en sådan ändring.

Eftersom att det inte fanns tillgång till någon fysisk CAN-buss eller information om meddelanden och noder som ska dela samma CAN-buss som systemet kunde det endast fastställas att paketstrukturen som existerar för CAN-meddelanden lämpar sig väl.

Det är även en brist att programmet endast kan köras på en Windows dator eftersom det är möjligt att andra operativsystemet är mer önskvärda vid framtida implementationer. Dock kan stora delar av programmet överföras till andra enheter som inte kör Windows om man

anpassar koden för att läsa av USB-porten till att bli kompatibel med det nya systemet. I fallet att man väljer att utveckla en framtida variant av systemet där man låter mikrokontrollen själv utföra omvandlingen från sensordata till tillståndsvariabel är inte detta nödvändigt eftersom att sensordata då kan läsas direkt ur sensorns register.

En noggrann analys av vad som är orsaken till det brus som upplevts under projektet borde utförts. På samma sätt borde en noggrann analys gjorts av hur sensorn påverkats av att bommen hastigt förflyttas samt att storleken på den påverkan som upplevs då teststationen roteras. Detta gjordes inte delvis på grund av bristande kunskap och delvis på grund av tidsbrist.

Det problem som uppkommer när bommen utsätts för vrid eller tilt kunde inte identifieras och åtgärdas. Detta gör att systemet endast kan användas efter att felet har hittats och åtgärdas eller om bommens orientering kan garanteras vara konstant. Det finns två teser för vad som kan vara orsaken till felet. Den första är att KL25Z skickar felaktig data då magnetfältet påverkas av att magneten förflyttas i höjddled. Den andra är att läsningen av KL25Z blir felaktig till följd av en oväntad ändring av meddelandestrukturen. Ingen av dessa teser kunde testas eller verifieras till följd av tidsbrist.

Hade man gjort tester för att bestämma påverkan vid vrid och tilt av bommen med den mjukvara som följde med utvecklingssystemet hade man troligen kunnat fastställa hur stor denna påverkan var. Detta gjordes dock inte eftersom att det endast ansågs värdefullt att genomföra testet för det egna systemet.

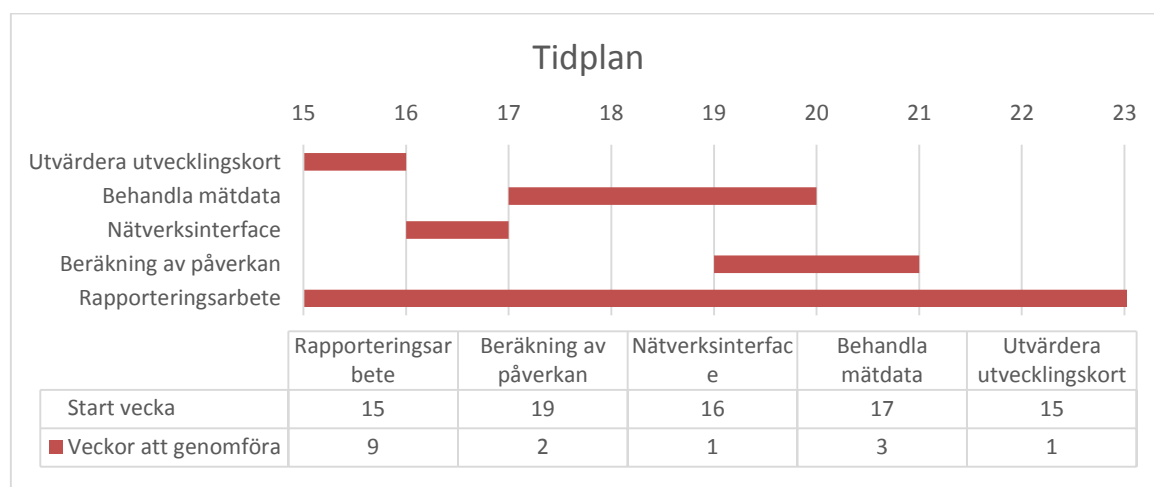
När data skulle extraheras från utvecklingssystemet behövde protokollet samt parametrarna för överföringen bestämmas manuellt eftersom att denna information inte fanns tillgänglig för den färdiga programvaran för mikrokontrollen på KL25Z. Den information som fanns var förkompilerade binärfiler som kan laddas in i mikrokontrollen. Detta föranledde att den manuella utredningen genomfördes. Det finns dock information och guider för hur man kan utveckla egen programvara för mikrokontrollen vilket förmodligen är anledningen till att den färdiga programvaran saknar dokumentation.

Att en magnetometer användes specificerades av uppdragsgivaren men liknande system skulle kunna konstrueras med andra sensortyper. Exempelvis skulle en optisk sensor med en kamera kunna användas för att bestämma en tillståndsvariabel genom att bilden från den optiska sensorn behandlas av en beräkningsenhet. Det är möjligt att en optisk sensor kan producera mer noggranna resultat och därför borde detta utvärderas innan man för projektet vidare. Något som också behöver beaktas är vilka nackdelar de olika sensortyperna har. En optisk sensor är känslig för ljus samt skuggor medan en magnetometer är känslig för magnetism. Detta betyder att de är olika svårigheter som måste behandlas

Det finns även givare på marknaden som på ett effektivt sätt kan producera en tillståndsvariabel som motsvarar ett vinkelutslag. Dessa givare har väldigt hög upplösning men den stora nackdelen är att de måste sitta i anslutning till en rörlig del. Detta begränsar användningsområdet. Dessutom så är dessa givare begränsade i att de endast kan användas för att beräkna en specifik tillståndsvariabel medan det utvecklade systemet kan anpassas för att beräkna en tillståndsvariabel för en vinkel eller, med vissa förändringar, en distans.

6.3 Planering kontra genomförande

Innan projektet påbörjades togs en planering fram som beskrivs med Gantt-schemat i figur 6.1.



Figur 6.1 Gantt-schema över planeringen innan projektet påbörjats.

Efter att projektet påbörjats tog det inte lång tid innan planeringen frångicks eftersom det tog två veckor utvärdera utvecklingssystemen för att bestämma vilket som var bäst lämpat. Detta till följd av komplikationer med att få KL25Z kortet att leverera data till Windows datorn och att det inte fanns tillgång till lokalerna där sensorerna fanns de första två dagarna. Vidare gjorde en veckas sjukdom att projektet blev ytterligare försenat. När den fjärde veckan påbörjades låg projektet två veckor efter planeringen.

Detta gjorde att resterande delen av projektet blev lidande och det kunde bara spenderas ytterst lite tid på att utvärdera andra möjligheter än de två som rapporten diskuterar för hur sensordata kan omvandlas till ett värde för vinkeln.

Trots detta har ett fungerande system utvecklats och de frågeställningar som ställts upp besvarats.

6.4 Vidareutveckling

För att föra projektet vidare borde en kravspecifikation utvecklas för vad magnetometer måste ha för egenskaper. Det är möjligt att en känsligare magnetometer kan leda till bättre resultat.

Sedan måste systemet anpassas för en verklig tillämpningssituation som mycket väl kan visa andra egenskaper gällande både dynamik och hur magneten och utvecklingssystemet kan placeras.

Programvaran som har utvecklats läser av och konverterar data från magnetometern på KL25Z. Om programvaran körs på en dator med en lägre processorfrekvens är det möjligt att det får ett oönskat beteende. Detta har inte utvärderats eller testats vilket borde göras ifall det ska ligga till grund för ett framtida produktupplägg.

Vidare måste CAN-meddelandet anpassas för den CAN-buss det sänds på. Det som behöver anpassas är vilken identifierare som ska användas samt att man bör utreda om vilka övriga meddelanden som systemet ska dela buss med. Vidare måste det undersökas vilken

paketstruktur som används och vilka andra system som är implementerade. Troligtvis ska meddelandet anpassas efter J1939 standarden.

Det behöver göras tester för att fastställa hur yttre störningar påverkar systemet. Eftersom att fordonskombinationer utsätts för en rad olika situationer som inte kunde testas via den teststation som används. Några av dessa faktorer som kan påverka systemet är:

1. Temperatur
2. Fukt
3. Andra fordon som kör i närheten av fordonskombinationen
4. Fordonskombinationers påverkan då dessa har flera magnetiska egenskaper som att de är gjorda i metall och har flera elektriska komponenter
5. Hur jordens magnetfält förändras då fordonet färdas mellan olika städer

Referenser

- [1] J. L. Crassidis, K.-L. Lai, and R. R. Harman, "Real-Time Attitude-Independent Three-Axis Magnetometer Calibration," *J. Guid. Control. Dyn.*, vol. 28, no. 1, pp. 115–120, Jan. 2005.
- [2] W. Voss, *A comprehensible guide to controller area network*. 2005.
- [3] E. Furlani, *Permanent magnet and electromechanical devices: materials, analysis, and applications*. 2001.
- [4] L. E. Frenzel, *Principles of electronic communication systems*. McGraw-Hill, 2008.
- [5] Riemersma Thiadmer (CompuPhase), "Termite: RS232 Terminal." [Online]. Available: http://www.compuphase.com/software_termite.htm. [Accessed: 01-May-2016].
- [6] N. Kularatna and Institution of Electrical Engineers., *Digital and analogue instrumentation : testing and measurement*. Institution of Electrical Engineers, 2003.
- [7] U. Bilting and J. Skansholm, *Vägen till C*. Studentlitteratur, 2011.
- [8] Microsoft, "Welcome to Visual Studio 2015." [Online]. Available: <https://msdn.microsoft.com/en-us/library/dd831853.aspx>. [Accessed: 01-Jun-2016].
- [9] The MathWorks, "MATLAB The Language of Technical Computing." [Online]. Available: <http://se.mathworks.com/products/matlab/>. [Accessed: 01-Jun-2016].
- [10] C. Wilcox, M. M. Strout, and J. M. Bieman, "An optimization-based approach to lookup table program transformations," *J. Softw. Evol. Process*, vol. 26, no. 6, pp. 533–551, Jun. 2014.
- [11] Microsoft, "Excel 2016." [Online]. Available: <https://products.office.com/en/excel>. [Accessed: 01-Jun-2016].
- [12] M. Di Natale, H. Zeng, P. Giusto, and A. Ghosal, *Understanding and Using the Controller Area Network Communication Protocol*. New York, NY: Springer New York, 2012.
- [13] Infineon Technologies AG, "Sensors 2GO." [Online]. Available: <http://www.infineon.com/cms/en/product/promopages/sensors-2go/>. [Accessed: 18-Apr-2016].
- [14] Freescale Semiconductor, "RD4247MAG3110 Sensor Toolbox Tilt-Compensated eCompass Kit - User's Guide." [Online]. Available:

http://cache.nxp.com/files/sensors/doc/user_guide/MAG3110UG.pdf.
[Accessed: 18-Apr-2016].

- [15] NXP Semiconductors, “FRDM-KL25Z: Freedom Development Platform for Kinetis KL14, KL15, KL24, KL25 MCUs.” [Online]. Available: <http://www.nxp.com/products/software-and-tools/hardware-development-tools/freedom-development-boards/freedom-development-platform-for-kinetis-kl14-kl15-kl24-kl25-mcus:FRDM-KL25Z>. [Accessed: 18-Apr-2016].
- [16] Freescale Semiconductor, “OpenSDA User’s Guide,” 2012. [Online]. Available: http://cache.nxp.com/files/32bit/doc/user_guide/OPENSDAUG.pdf?fp=1&WT_TYPE=UsersGuides&WT_VENDOR=FREESCALE&WT_FILE_FORMAT=pdf&WT_ASSET=Documentation&fileExt=.pdf. [Accessed: 01-May-2016].
- [17] NXP Semiconductors, “FRDM-FXS-MULT2-B: Freedom Development Platform for NXP Sensors with Bluetooth®.” [Online]. Available: <http://www.nxp.com/products/software-and-tools/hardware-development-tools/freedom-development-boards/frdm-fxs-mult2-b-freedom-development-platform-for-freescale-sensors-with-bluetooth.:FRDM-FXS-MULT2-B>. [Accessed: 18-Apr-2016].
- [18] The MathWorks, “polyfit.” [Online]. Available: <http://se.mathworks.com/help/matlab/ref/polyfit.html>. [Accessed: 20-May-2016].
- [19] The MathWorks, “polyval.” [Online]. Available: <http://se.mathworks.com/help/matlab/ref/polyval.html>. [Accessed: 20-May-2016].
- [20] Microsoft, “CreateFile function.” [Online]. Available: [https://msdn.microsoft.com/en-us/library/windows/desktop/aa363858\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/aa363858(v=vs.85).aspx). [Accessed: 14-May-2016].
- [21] Microsoft, “DCB structure.” [Online]. Available: [https://msdn.microsoft.com/en-us/library/windows/desktop/aa363214\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/aa363214(v=vs.85).aspx). [Accessed: 14-May-2016].
- [22] Microsoft, “ReadFile function.” [Online]. Available: [https://msdn.microsoft.com/en-us/library/windows/desktop/aa365467\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/aa365467(v=vs.85).aspx). [Accessed: 14-May-2016].

- [23] Microsoft, “fprintf, _fprintf_l, fwprintf, _fwprintf_l.” [Online]. Available: <https://msdn.microsoft.com/en-us/library/xkh07fe2.aspx>. [Accessed: 14-May-2016].

Bilagor

Bilaga 1. Sid 1(2)

Tabell för faktiska vinkeln kontra den med ekvationen beräknade vinkeln för resultant vektorn för y och x vektorerna.

Faktisk vinkel i grader	Arctan(Fy/Fx)	Differans
-90	45,44	-135,44
-85	50,64	-135,64
-80	56,23	-136,23
-75	61,90	-136,90
-70	68,31	-138,31
-65	74,54	-139,54
-60	80,66	-140,66
-55	86,42	-141,42
-50	-87,28	37,28
-45	-81,50	36,50
-40	-75,24	35,24
-35	-68,82	33,82
-30	-62,67	32,67
-25	-56,45	31,45
-20	-49,29	29,29
-15	-41,84	26,84
-10	-33,65	23,65
-5	-25,31	20,31
0	-15,08	15,08
5	-5,57	10,57
10	3,94	6,06
15	14,81	0,19
20	26,01	-6,01
25	36,04	-11,04
30	45,36	-15,36
35	54,22	-19,22
40	62,74	-22,74
45	70,10	-25,10
50	77,28	-27,28
55	84,09	-29,09
60	-89,48	149,48
65	-82,36	147,36
70	-75,32	145,32
75	-67,94	142,94
80	-60,47	140,47
85	-52,87	137,87
90	-45,34	135,34

Tabell för faktiska vinkeln kontra den med ekvationen beräknade vinkeln för resultant vektorn för y och z vektorerna.

Faktisk vinkel i grader	Arctan(Fy/Fz)	Differans
-90	-57,92	-32,08
-85	-58,51	-26,49
-80	-58,81	-21,19
-75	-58,36	-16,64
-70	-57,85	-12,15
-65	-56,83	-8,17
-60	-55,38	-4,62
-55	-53,76	-1,24
-50	-51,57	1,57
-45	-48,91	3,91
-40	-45,79	5,79
-35	-42,28	7,28
-30	-38,56	8,56
-25	-34,67	9,67
-20	-30,09	10,09
-15	-25,33	10,33
-10	-20,12	10,12
-5	-14,89	9,89
0	-8,69	8,69
5	-3,17	8,17
10	2,22	7,78
15	8,20	6,80
20	14,24	5,76
25	19,66	5,34
30	24,64	5,36
35	29,36	5,64
40	33,72	6,28
45	37,39	7,61
50	40,44	9,56
55	43,11	11,89
60	45,00	15,00
65	46,52	18,48
70	47,57	22,43
75	48,21	26,79
80	48,19	31,81
85	47,66	37,34
90	46,50	43,50