



Bachelor Thesis report

Engineering & Computer Science, Degree
project 15hp

Precision i Rörelse

Horisontell Hoppmätning med IMU och Magnetometer

Halmstad Högskola

Mekatronikingenjör 180hp

Halmstad 03-April-2024

Ismail Abuawad

Sammanfattning

Detta examensarbete har genomförts med syftet att utveckla Inno-x företagets system, som är avsett för vardagsidrottare för att mäta neuromuskulära aktiviteter i underkroppen med hjälp av modern teknologi. Systemet omfattar en tröghetsmätningssenheter (IMU) med accelerometer, gyroskop och en EMG-sensor (elektromyografi). Denna konfiguration möjliggör noggrann övervakning av neuromuskulära aktiviteter genom analys av svar på träning. Studiens mål var att identifiera en effektiv sensor för mätning av horisontella hoppavstånd och att utveckla en algoritm som sedan ska integreras i företagets produkt.

Produkten kommer att använda magnetometer och IMU för att tolka mänskliga rörelser och för att förbättra noggrannheten i företagets mätningssystem. Processen inkluderar förbättring av mätningarnas noggrannhet, integration av teknik med biomekaniska principer, utvärdering av kalibreringstekniker för magnetometeravläsningar, kombination av sensorer för rörelseanalys och genomförande av utvärdering med olika åldersgrupper som består av 10 deltagare för att bedöma systemets effektivitet.

Även om ingen av metoderna helt uppnådde den önskade noggrannheten inom ± 5 cm, visade alla metoder god prestanda för olika tillämpningar. Detta antyder att implementeringen av en kalibrerad magnetometer potentiellt kan förbättra systemets noggrannhet vid bestämning av horisontella hoppavstånd, dock endast med en liten marginal, eftersom studien visade att med kalibrerade magnetometer RMSE (Root Mean Square Error) ökat med 0.99 cm.

Ytterligare forskning rekommenderas för att undersöka nya sätt att kalibrera sensorer och integrera dem för mer precisa avläsningar. Dock bör det beaktas att magnetometeravläsningar påverkas av miljöfaktorer. Dessutom är det viktigt att skapa ett användarvänligt gränssnitt som gör det möjligt för idrottare att enkelt spåra och analysera sina prestandadata.

Nyckelord: IMU, magnetometer, orienteringsuppskattning, horisontellt hoppmätningssystem, systemanalys.

Abstract

This thesis has been conducted with the objective of developing the Inno-X company's system, which is intended for everyday athletes to measure neuromuscular activities in the lower body using modern technology. The system includes an Inertial Measurement Unit (IMU) with an accelerometer, gyroscope, and an Electromyography (EMG) sensor. This configuration enables accurate monitoring of neuromuscular activities through the analysis of responses to training. The study's goal was to identify an effective sensor for measuring horizontal jump distances and to develop an algorithm that would then be integrated into the company's product.

The product will use a magnetometer and IMU to interpret human movements and to improve the accuracy of the company's measurement system. The process includes improving the accuracy of measurements, integrating technology with biomechanical principles, evaluating calibration techniques for magnetometer readings, combining sensors for motion analysis, and conducting evaluations with different age groups consisting of 10 participants to assess the system's effectiveness.

Although none of the methods fully achieved the desired accuracy within ± 5 cm, all methods showed good performance for various applications. This suggests that the implementation of a calibrated magnetometer could potentially improve the system's accuracy in determining horizontal jump distances, albeit only by a small margin, as the study showed that with calibrated magnetometers, the Root Mean Square Error (RMSE) increased by 0.99 cm.

Further research is recommended to explore new ways to calibrate sensors and integrate them for more precise readings. However, it should be considered that magnetometer readings are affected by environmental factors. Additionally, it is important to create a user-friendly interface that enables athletes to easily track and analyze their performance data.

Keywords: IMU, magnetometer, orientation estimation, horizontal jump measurement system, system analysis.

Begreppslista

Root mean square error (RMSE)	Är ett mått på spridning av avvikelserna vid jämförelse mot ett givet värde av standardfilen
MEMS-enheter	Är små integrerade enheter som kombinerar mikroelektromekaniska system
STEVAL-MKI109v3	Är ett utvecklingskort som underlättar utvärdering av MEMS- sensorer
STEVAL-MKI217v1	Sensormodulen som har magnetometer, accelerometer och gyro innehåller två MEMS-enheter följande: LIS2MDL: Magnetometer LSM6DSOX: Accelerometer och Gyroscope
EMG	Electromyography är teknik för att registrera medicinska elektriska signaler som erhålls från neuromuskulär aktivitet.
IMU	Inertial measurement unit är en enhet som integrerar fleraxliga accelerometrar, gyroskop för att ge en uppskattning av ett objekts orientering i rummet.
Uppskattning (tröghetsnavigering)	Processen för att bestämma ett objekts position, hastighet, och orientering genom att analysera och integrera data från inertiella sensorer (IMU) osv... Denna teknik används i tröghetsnavigeringssystem för att beräkna objektets rörelse bana och orientering utan behov av externa referenser.
Global rotation	Det är rotationen som inte är relaterad till enhetens lokala rotation, vilken representerar vektorns rotation när enheten befinner sig i nollrotationspunkten (referenspunkt).
MARG	Står för (Magnetic, Angular Rate and Gravity) det är kombination av sensorer som används för att mäta rörelse, orientering i tredimensionellt rum. Typ magnetometer och IMU.
UKF	Unscented Kalman Filter

Förord

Det hade inte varit lätt att slutföra detta projekt och nå detta inspirerande resultat utan de fantastiska ansträngningarna och stödet från många, särskilt Roger Nils Carlson, min handledare. Han har verkligen varit en källa till inspiration och vägledning under hela projektet. Ett stort tack till James Parker från Innowearable AB för deras fulla förtroende och den fantastiska möjligheten att leda detta projekt. Han har gett mig ovärderligt stöd, vilket jag kommer att minnas för alltid.

Innehållsförteckning

1	Introduktion	1
1.1	Syfte och mål:	1
1.2	Krav	2
1.3	Avgränsningar	2
2	Bakgrund	3
2.1	Relaterade projekt	3
2.2	Magnetometerkalibrering för Hard Iron och Soft Iron	3
2.2.1	Hard och Soft iron störningar	4
2.2.2	Visualisering av Hard och Soft iron störningar	4
2.2.3	Hard- och Soft iron -förvrängningar i 3D	5
2.2.4	Eliminering av Hard- och Soft iron -förvrängningar	5
3	Teori	7
3.1	Grundläggande principer för sensorfusion	7
3.2	Mahony-filter	8
3.2.1	Användning av Accelerometer och Gyroskop (IMU)	8
3.2.2	Integration av kvaternioner och accelerometerdata	8
3.2.3	Initiering av global rotation	9
3.2.4	Integration av kvaternioner och gyroskopdata	11
3.2.5	Uppdatering av global rotation	11
3.2.6	Användning av Accelerometer, Gyroskop och Magnetometer (MARG)	12
3.2.7	Kvaternion representation:	13
3.3	Madgwick-filter	14
3.3.1	Grundläggande om kvaternioner	15
3.3.2	Multiplikation av kvaternioner	15
3.3.3	Integration av gyroskop	15
3.3.4	Gradient descent	16
3.3.5	Uppdatering med gradient descent	16
3.4	Horisontellt hoppavstånd:	17
3.4.1	Kvaternion baserad orienteringsoutput	17
3.4.2	Beräkning av gravitationsvektor:	18
3.4.3	Isolering av linjär acceleration	18
3.4.4	Beräkning av förflyttning	18
3.4.5	Avstånd beräkning	19
3.4.6	Kumulativa avstånd	19

4 Metod	21
4.1 Experimentell uppsättning	21
4.2 Hårdvara:	22
4.3 Programvara.....	22
4.4 Datainsamling och analys	22
4.5 Statistisk analys	23
4.5.1 Deskriptiva statistiken.....	23
4.5.2 Statistiska tester	23
4.6 Beräkning och jämförelse	23
4.7 Utvecklingsprocess:.....	24
4.7.1 Korrigering och konvertering av magnetometer.....	25
4.7.2 Orienteringsuppskattning (Mahony - Madgwick)	25
4.7.3 Positionsestimering	26
4.7.4 Visualisering	26
4.7.5 Utvärdering	26
4.8 Matlab statistik	27
4.9 Kodöversikt för beräkning av horisontellt hoppavstånd.....	27
4.9.1 Systemöversikt.....	27
4.9.2 Beskrivning av Python klasser.....	27
4.9.3 Sammanfattning av Python-algoritmen	28
5 Resultat	29
5.1 Utvärdering av Mahony - Madgwick	29
5.2 Utvärdering av (Mahony utan Mag – Mahony med kali- Mag).....	31
5.3 Utvärdering av (Okali mag VS kali mag) med Mahony-filter	32
6 Diskussion	35
6.1 Samhällskrav	35
6.2 RMSE resultat för experimentet	35
6.3 Statistisk analys	35
6.3.1 Mahony vs Madgwick.....	35
6.3.2 Mahony utan Mag vs. Mahony med kali Mag.....	35
6.3.3 Mahony med okali Mag vs. Mahony med kali Mag	36
6.4 Jämförelse med ett relaterat projekt.....	36
6.5 Utmaningar med sensorjustering	36
7 Slutsats	37
8 Litteraturförteckning	39
9 Bilagor	43

9.1	Figurer	43
9.2	Fördelar med en integrerad modul.....	44
9.3	Avläsning av sensorvärden:.....	44
9.4	Teknisk jämförelse mellan olika IMU specifikationer	45
9.5	Magnetometer specifikationer jämförelse	46
9.6	Konfigurering av Hårdvaran:.....	47
9.7	Python algoritmen.....	48
	A – SensorBoard Python Class	48
	B – Dataprocessor Python Class	49
	C – DataVisualizer Python Class	52
	D– MainCalss Python Class.....	53
9.8	Matlab kod.....	54
Figurförteckning		
Figur 1	Figur 1 Inno-X™ befintlig produkt [1]	1
Figur 2	Utan störningar [17].....	4
Figur 3	Hard Iron störningar [17].....	4
Figur 4	Hard och Soft Iron störningar [17]	4
Figur 5	Diagram för Mahony filter [11].....	8
Figur 6	Accelerometer data i varje sampeltid (dt).....	9
Figur 7	Uppdatera q-rotationen baserat på Gyrodata för varje dt	11
Figur 8	Diagram Madgwick filter [11].....	14
Figur 9	Systemöversikt diagram	24
Figur 10	Visar värdena från den okalibrerade magnetometern.....	25
Figur 11	Visar värdena från den kalibrerade magnetometern.....	25
Figur 12	Utvecklingskort STEVAL-MKI109v3 & Modulen STEVAL- MKI217v1 som används under experimentet.....	43
Figur 13	Acceleration värde före och efter verkställda oreinteringsfilter (Madgwick och Mahony)	43
Tabellförteckning		
Tabell 1	Visar olika mätningar för deltagaren samt fel skillnad mellan algoritmer	29
Tabell 2	Statistisk jämförelse av prestanda för filtreringsalgoritmer.....	30
Tabell 3	Jämförelse av felprocent och absolut felvärdeskillnad mellan filtreringsalgoritmer .	30
Tabell 4	Deltagarens hoppmätning Mahony utan Magnetometer – Mahony med kalibrerad Magnetometer.....	31
Tabell 5	Jämförelse av prestanda för Mahony algoritmen med och utan kalibrerade Magnetometer	31
Tabell 6	Bräkning av procentandel felen och den absoluta skillnaden i fel för algoritmer	32
Tabell 7	Deltagarens hoppmätning för okalibrerad magnetometer med Mahony-filter VS kalibrerad magnetometer med Mahony-filter.....	33
Tabell 8	Jämförelse av prestanda för Mahony algoritmen med kalibrerad Mag och okali Mag	33
Tabell 9	Bräkning av procentandel felen och den absoluta skillnaden i fel för algoritmer	34
Tabell 10	(RMSE) på positionsuppskattningarna med råa och filtrerade mätningar visade att filtreringsåtgärden försämrar positionsuppskattningarna 4.1(Roobeek,2017s.53) [35].....	36

1 Introduktion

Detta projekt presenterar utvecklingsprocessen för att förbättra det befintliga InnoWearable-systemet. Systemet har tidigare utformats för att möjliggöra olika mätningar relaterade till idrottsprestanda [1]. Målet med denna iteration är att utöka systemets funktionalitet så att det kan göra noggranna mätningar av horisontella hoppavstånd. Detta kommer att erbjuda användarna värdefull information för övervakning av prestanda och anpassning av träning [5]. Rapporten kommer att fokusera på integrationen av nödvändig hårdvara- och mjukvara för att stödja mätningar av hoppavstånd, inklusive integration av sensorer, insamling och bearbetning av data, med utgångspunkt i Inno-X™-systemet. Dessutom kommer en ny funktion för användarfeedback och rekommendationer att integreras smidigt med den befintliga Inno-X™-appen och sensorerna för att ge användarna en djupgående analys av deras neuromuskulära respons på träning, se figur 1 [1].



Figur 1 Figur 1 Inno-X™ befintlig produkt [1]

System för personliga rekommendationer är designade för att skapa individuellt anpassade tränings- och återhämtningsförslag som baseras på varje användares egna neuromuskulära data. Användarna kommer att få personligt anpassade meddelanden och vägledning som beaktar deras unika träningsbehov och mål. En viktig utveckling inom detta område är förbättring av systemet genom implementeringar av nya komponenter i företagets befintliga produkt, såsom gyroskop-, accelerometer- och magnetometersensorer. Dessa tillägg kommer att integreras i Inno-X företagets produkter för att mäta horisontella hoppavstånd [1]. Denna innovation innebär en stor möjlighet att inte bara förbättra användarupplevelsen utan även öka systemets funktionalitet. Genom att erbjuda exakta och pålitliga mätningar av hoppavstånd, kommer det här uppgraderade systemet att spela en avgörande roll i att främja användarnas idrottsliga utveckling och hjälpa dem att nå sina mål.

1.1 Syfte och mål:

Syftet med projektet är att utveckla en mätalgoritm och en sensor som kan mäta det horisontella hoppavståndet med hjälp av en accelerometer, en magnetometer och ett gyroskop.

Inno-X är ett system som mäter muskelprestanda enbart med en accelerometer och andra sensorer, men för närvarande saknas en magnetometer. Systemet har utformats för att idrottare, tränare och fitnessentusiaster ska kunna följa sina prestationer och optimera sina träningsrutiner [1]. Projektet syftar till att identifiera en effektiv sensor för mätning av det horisontella hoppavståndet och att utveckla en tillhörande mätalgoritm. Denna nyutvecklade mätalgoritm

och sensor planeras att integreras i Inno-X:s produktlinje som en förbättring. Företaget kommer därefter att ansvara för implementeringen av mätagoritmen och sensorn i sin produkt.

1.2 Krav

- Systemet ska kunna mäta en idrottares horisontella hoppdistans med en felmarginal som inte överstiger ± 5 cm, vilket garanterar en hög nivå av precision i bedömningen.
- Systemet måste kunna tillhandahålla beräkningar av hoppavstånd i realtid för att möjliggöra omedelbar feedback.
- De uppmätta resultaten bör vara anpassningsbara för att bedöma idrottares konditionsnivå.

1.3 Avgränsningar

Experimentet är bundet till specifikationerna för accelerometer och gyro (6-axlig IMU) LSM6DSOX samt magnetometer (3-axlig) (LSM6DSOX) från Steval-MKI217v1-modulen [6, 7].

- a) **Sampelfrekvens:** Begränsad till maximala utdatafrekvensen (416 Hz) som dessa sensorer erbjuder.
- b) **Brusnivå:** Interna sensorbrus kan påverka mätnoggrannheten.

Omfattning av rörelseanalys

Algoritmen och metoderna för sensorfusion är optimerade för horisontella hoppmönster. Komplexiteter som introduceras av andra atletiska rörelser, såsom förändringar i höjd, hanteras inte.

Miljöantaganden

Kalibreringar och mätningar baseras på antagandet om ett konstant geomagnetiskt fält och gravitationskraft. Variationer på grund av höjdändringar eller magnetiskt störningar beaktas inte.

Begränsningar vid databehandling

Realtidsdatabehandling prestanda är beroende av Python-programmets beräkningsmässiga effektivitet och gränssnittsförmåga på utvecklingskortet. Gränssnittsfunktionalitet samt processorkraft kan potentiellt påverka prestandan vid snabb realtidsberäkningen.

Tidsmässiga begränsningar

Systemet är optimerat för kortvariga hoppaktiviteter. Långvariga eller snabba efterföljande hopp kan introducera problem med avdrift eller feljustering som inte beaktas i den nuvarande algoritmen.

2 Bakgrund

2.1 Relaterade projekt

Användningen av bärbar teknologi för övervakning av idrottsprestationer har blivit allt mer populär. För idrottare och idrottsentusiaster har prylar som träningsmätare, smartklockor och specialiserade idrottssensorer blivit en standardutrustning [2]. Hjärtfrekvens, GPS-spårning, rörlighetsanalys och andra typer av data kan alla samlas in med hjälp av dessa teknologier [18, 2].

Bedömning av hoppförmåga är en kritisk del i idrottskonditionering och träning. För att mäta hoppets höjd, kraft och avstånd har forskare och praktiker använt olika tekniker, inklusive kraftplattor, optiska spårningssystem och accelerometrar. Dessa mätvärden är till hjälp för att bedöma en idrottares smidighet och explosiva styrka [3].

I en studie som syftar till att förbättra noggrannheten i spårningen av spelarrörelser inom sport med hjälp av GPS- och IMU-enheter, framför allt för att höja skadeförebyggande åtgärder och prestationsförbättringar. Studien fokuserar på att utvärdera träningsrörelser och arbetsbelastning i kontexten av specifika sporter, där sensorfusionsteknik används för att uppskatta rörelse. Speciellt undersöks effektiviteten hos olika algoritmer för att beräkna spelaracceleration. Detta innebär en omfattande analys och utvärdering av olika filter, såsom unscented Kalmanfilter (UKF) och Madgwickfilter, utförs för att bestämma det mest lämpligt för projektets syften [35].

Studien beskriver användningen av prisvärda och lättillgängliga GPS- och IMU-sensorer i en bärbar enhet för att spåra idrottares rörelser. Till skillnad från andra studier som huvudsakligen förlitar sig tungt på simuleringar och robotteknik, utforskar denna forskning sensorfusions teknik i en sportmiljö. Resultaten kan bidra till idrottsvetenskapen, särskild när det gäller att optimera träning och förebygga skador. Detta nya angreppssätt skiljer studien från andra som främst fokuserar på självkörande fordon eller robotteknik [35].

Detta experiment tar sig an de praktiska utmaningarna som existerar inom tekniken för spårning av idrottare. I motsats till tidigare forskning, som ofta är teoretiska eller baserad på simulerade scenarier, tar denna studie hänsyn till sensorernas begränsningarna och svårigheterna med att noggrant filtrera bort brus för att exakt bestämma spelares tillstånd. Projektets huvudmål är att överkomma verkliga hinder inom idrottsspårningsteknik och därmed erbjuda konkreta fördelar för idrottsvetenskap, särskild inom områdena skadeförebyggande och prestationsförbättring. Denna inriktning på praktiska tillämpningar gör att detta projekt skiljer sig från relaterade forskningsarbeten som i huvudsak undersöker teknologins utveckling och validering av teknologier i kontrollerade miljöer [35].

2.2 Magnetometerkalibrering för Hard Iron och Soft Iron

Magnetometern är en sensor som upptäcker styrkan och riktningen av lokala magnetfält. Denna information kan jämföras med modeller av jordens magnetfält för att bestämma systemets orientering i förhållande till den magnetiska nordpolen [17]. I praktiska tillämpningar kan dock mätningar av magnetfält förvrängas av närliggande föremål, vilket resulterar i magnetiska störning [22]. För att uppnå noggranna mätningar av riktningen måste dessa störningar

reduceras. Konstanta interna magnetiska störningar kan identifieras och korrigeras genom kalibrering av både Hard iron och Soft iron. Artikeln framhåller att det är möjligt att kompensera för interna magnetiska störningar som inte varierar över tid med hjälp av kalibreringar för Hard iron och Soft iron [17].

2.2.1 Hard och Soft iron störningar

Permanenta förvrängningar i magnetiska mätningar klassificeras som Hard iron eller Soft iron störningar. Dessa uppstår på grund av närheten till föremål eller andra enheter som genererar magnetiska fält, till exempel högtalare, och de uppkommer när det magnetiska materialet är fast i samma referensram. Soft iron störningar anses vara förändringar i det omgivande magnetfältet och orsakas av metaller som järn eller nickel om de är placerade i närheten av magnetometern. Hard iron störningar bidrar vanligtvis mer än Soft iron till det totala okorrigerade felet än Soft iron störningar [22, 17].

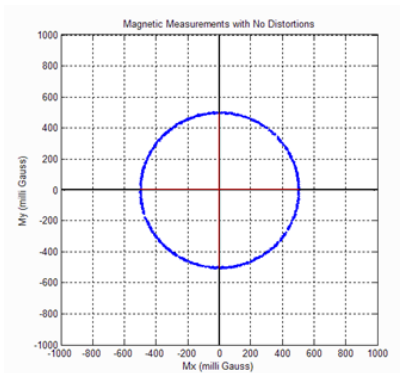
2.2.2 Visualisering av Hard och Soft iron störningar

Visualisering är en metod för att korrigera förvrängningar orsakade av både från Hard iron och Soft iron. En av dessa metoder är att plotta magnetometerns data i en 2D-graf. Figur 2 representerar de magnetiska mätningarna i ett 2D-diagram medan enheten långsamt roteras runt z-axeln. Som tidigare nämnts finns det flera scenarier för magnetometrar, vilken kan indelas i följande kategorier:

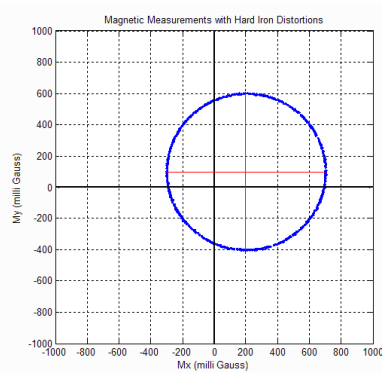
För det första, när det inte finns några förvrängningar från magnetometern, vilket klart kan ses i figur 2. Detta kännetecknas av att cirkelns centrum är i origo i förhållande till både axlar x och y. Cirkelns radie representerar storleken på det magnetiska fältet [22, 17].

I det andra scenariot beaktas närvaron av Hard iron förvrängningar, vilka orsakar en konstant förskjutning i de magnetiska mätningarna. Detta visas i figur 3 där en 2D-plot illustrerar dessa förvrängningen. Cirkelns centrum är förskjutet, vilket tydliggör bias från Hard iron [17].

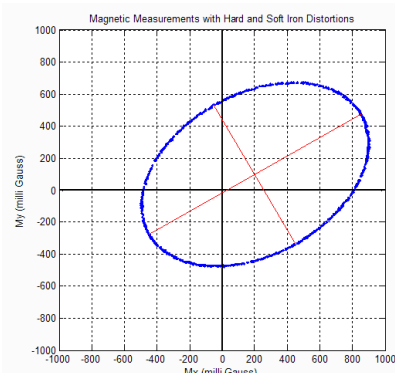
Att använda ett 2D-diagram för att visualisera förvrängningar från både Soft iron och Hard iron gör det enklare att observera effekterna. För Hard iron förvrängningar karaktäriseras effekten av att cirkelns centrum förskjuts från origo utan att påverka på cirkelns form. Å andra sidan uppstår elliptiska formen uppstår som ett resultat av Soft iron förvrängningar, vilka påverkar de



Figur 2 Utan störningar [17]



Figur 3 Hard Iron störningar [17]



Figur 4 Hard och Soft Iron störningar [17]

befintliga magnetfälten genom att modifiera magnetometers utdata. Denna effekt framgår tydligt i figur 4 [22, 17].

2.2.3 Hard- och Soft iron -förvrängningar i 3D

Processen med att visualisera 2D-förvrängningar är att klargöra de effekterna som påverkar magnetisk mätning genom att visa skillnaderna mellan Hard och Soft iron förvrängningar. Ursprungligen utgörs magnetisk mätning av en tredimensionell vektor. Utan några av de tidigare nämnda påverkningarna formar denna mätningsvektor en fullständig sfär. I jämförelse med 2D-visualiseringar visar den att Hard iron- förvrängningar förskjuter sfärens centrum från nollpunkten, medan Soft iron-förvrängningar deformerar sfären och gör den till en ellipsoid [17].

2.2.4 Eliminering av Hard- och Soft iron -förvrängningar

Kalibreringen av magnetiska mätinstrument är en kritisk del av projektet för att eliminera de Hard och Soft iron förvrängningar som påverkar magnetometers utdata. Dessa förvrängningar kan orsaka signifikanta mätfel, vilket är särskilt problematiskt i experiment där precisionen i orienteringsuppskattningar är avgörande del. Metoden innebär att skapa en ellipsoid som representerar både biasen och den förvrängda formen på en sfär som är centrerad kring nollpunkten. Att visualisera i 3D är en viktig del av projektet för att tydligare förstå hur det magnetiska mätinstrumentet rör sig under rörelse. Denna tredimensionella visualisering tas rummens dimensioner i beaktande för att uppnå så exakta resultat som möjligt. Denna process kräver att det magnetiska mätinstrumentet roterar runt alla axlar (360°) för att täcka hela mätområdet. För att uppskatta kompensationsparametrarna för magnetometern används metoder som är tillämpliga på både Hard och Soft iron [17]. I processen för att kalibrera magnetometrar och korrigera för Hard och Soft iron-förvrängningar används en ekvation för att beräkna de kompensationsparametrarna. Formel (1) är som följer:

$$m_c = S_I(\tilde{m} - b_{HI}) \quad (1)$$

Där m_c är den kalibrerade magnetometeravläsningen, S_I matrisen för Soft iron- korrigering, \tilde{m} är den råa magnetometeravläsningen, och b_{HI} representerar bias från Hard iron [17]. Den kalibrerade avläsningen i vektorkomponentform ges av följande formel (2):

$$\begin{bmatrix} m_{cx} \\ m_{cy} \\ m_{cz} \end{bmatrix} = \begin{bmatrix} C_{00} & C_{01} & C_{02} \\ C_{10} & C_{11} & C_{12} \\ C_{20} & C_{21} & C_{22} \end{bmatrix} \begin{bmatrix} \tilde{m}_x - b_{H0} \\ \tilde{m}_y - b_{H1} \\ \tilde{m}_z - b_{H2} \end{bmatrix} \quad (2)$$

Denna ekvation är grundläggande för att korrigera magnetometerdata och för att säkerställa att avläsningarna blir så noggranna som möjligt. För en fullständig förklaring av denna kalibreringsprocess och ekvation, se [17].

3 Teori

Sensorfusion är ett tvärvetenskapligt område som är viktigt för tekniska applikationer såsom autonom navigering och dataanalys. Det omfattar metoder för att kombinera data från flera källor för att förbättra kvaliteten på information. Denna text fokuserar på Mahony- och Madgwick-filter, som är centrala för att stabilisera data från accelerometrar, gyroskop och magnetometrar. Dessa filter bidrar till utvecklingen av precisionsnavigering och rörelseanalys [21, 20]. Genom att undersöka dessa filter, deras teori och användning, belyser texten sensorfusionens förmåga med att öka prestanda och noggrannhet i tekniska system, vilket lägger grunden för framtida forskning och innovation inom området [20].

3.1 Grundläggande principer för sensorfusion

Sensorfusion är processen att kombinera data från olika sensorer för att få en mer tillförlitlig och omfattande förståelse av sensormiljön. Denna metodik tar sig an utmaningen med sensorernas begränsade precision och osäkerhet genom att utnyttja styrkorna hos varje enskild sensortyp. Genom att sammanföra och analysera data från flera källor kan sensorfusionssystem erbjuda förbättrad informationskvalitet jämfört med vad som är möjligt med enskilda datakällor [20, 21].

En av de största utmaningarna inom sensorfusion är att korrekt integrera och tolka sensordata från olika källor, som kan ha varierande noggrannhetsnivåer, upplösningar - det vill säga hur exakt dessa sensorer kan mäta och uppdateringsfrekvenser [21]. För att hantera dessa utmaningar används avancerade algoritmer och filter. Bland dessa är Mahony-filter och Madgwick-filter särskilt framstående för deras förmåga att effektivt integrera data från accelerometrar, gyroskop och magnetometrar [15, 16].

Mahony-filter

Mahony-filter är en populär teknik inom sensorfusion som används för att uppskatta orientering och rörelse genom att integrera data från accelerometrar, gyroskop och, i vissa fall, magnetometrar. Det är känt för sin låga beräkningskostnad och goda prestanda i realtidssystem, vilket gör det idealiskt för inbäddade applikationer och mobila enheter [15].

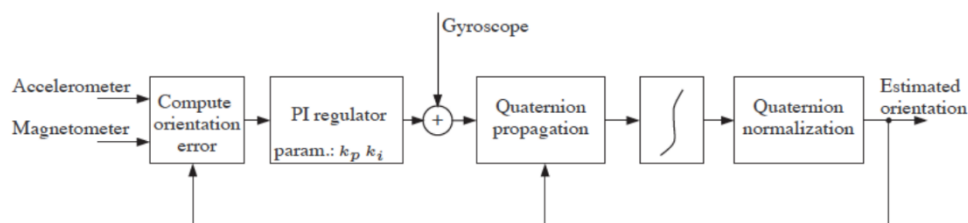
Madgwick-filter

Madgwick-filter erbjuder, å andra sidan, en sofistikerad lösning för att beräkna orientering genom att använda en optimeringsalgoritm som minimerar fel mellan uppskattad och faktisk orientering. Detta filter är särskilt värdefullt för dess förmåga att leverera högkvalitativa resultat även under dynamiska förhållanden, vilket gör det till ett föredraget val för applikationer som kräver hög precision och pålitlighet [16].

Båda dessa filter spelar en kritisk roll i sensorfusion genom att erbjuda robusta metoder för att hantera osäkerhet och brus i sensordata, vilket möjliggör skapandet av avancerade navigationssystem och rörelsedetekteringstekniker. Genom att förstå och tillämpa dessa grundläggande principer och tekniker kan vi utveckla system som är kapabla att noggrant tolka och reagera på sin omgivning, vilket är avgörande för framstegen inom områden som autonomi, robotik och interaktiv teknologi [21, 16, 15].

3.2 Mahony-filter

I sammanhanget av att mäta av horisontellt hoppavstånd inom sportbiomekanik, spelar Mahony-filter en nyckelroll för att noggrant följa och analysera idrottarnas rörelser genom teknik för sensorfusion [15]. Utöver standard-IMU-sensorer (accelerometrar och gyroskop), används detta filter för att integrera data från tröghetsmätenheter (IMU:er) och från mer avancerade system, såsom MARG (magnetometer, accelerometer, gyroskop). Figur 5 visar filtrets blockdiagram [15, 11].



Figur 5 Diagram för Mahony filter [11]

3.2.1 Användning av Accelerometer och Gyroskop (IMU)

Denna studie bidrar till förståelsen av hur avancerade sensorfusions- och filtreringsalgoritmer kan förbättra noggrannheten i orienteringssystem. Detta har direkt tillämpning på rörelseanalys inom sportbiomekanik. Forskningen ger en teoretisk grund för att noggrant spåra och analysera idrottares rörelser med hjälp av Mahony-filter kombinerat med IMU-data i Mahony-filterprocessen. Accelerometern ger data om linjär acceleration, vilket inkluderar både enhetens rörelse och gravitationskraften. Denna information är avgörande för att bestämma enhetens orientering i förhållande till gravitationsfältet. Gyroskopet, däremot, mäter enhetens rotationshastighet, vilket ger en direkt indikation på hur snabbt och i vilken riktning enheten roterar [15, 11].

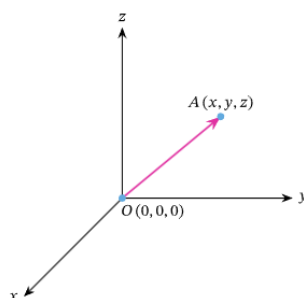
3.2.2 Integration av kvaternioner och accelerometerdata

Att omvandla accelerometerdata till kvaternioner innebär att utföra en serie beräkningar för att noggrant beskriva en enhets orientering. Accelerometerns rådata representeras i 3-D med en vektor $A(a_x, a_y, a_z)$ som utgår från origo och sträcker sig i rummet, vilket illustreras i figur 7. Kvaternioner, som utgör ett avancerat matematiskt verktyg, erbjuder en lösning för att undvika de begränsningar och problem som är förknippade med Euler-vinklar, såsom singulariteter och gimbal lock [26]. Denna metodik är särskilt fördelaktig eftersom den, möjliggör en smidig och kontinuerlig representation av orientering utan de ovannämnda problemen [15, 10]. Genom att kontinuerligt utföra dessa beräkningar varje gång enheten används, och genom att integrera data från accelerometern, kan man effektivt uppskatta enhetens orientering i förhållande till omgivningen. Dessa beräkningar, som är anpassade för tredimensionella miljöer, tar hänsyn till både gravitationens inverkan och eventuella rörelser som upptäcks av accelerometern. Användningen av kvaternioner erbjuder mer pålitlig och konsekvent orienteringsinformation, vilket är avgörande när man behöver ha precisa mätningar i rörelse där tillförlitliga orienteringsdata är kritisk [15, 10].

3.2.3 Initiering av global rotation

Normalizing av accelerometer data

För att erhålla en enhetsvektor som tydligt indikerar riktningen mot gravitationen är det nödvändigt att normalisera avläsningar av accelerometern [25]. Denna normaliseringsprocess är av stor betydelse, eftersom det inte primärt är avläsningarnas storlek som är av intresse, utan deras riktning i förhållande till gravitationens. Genom att normalisera avläsningarna isoleras den riktning som pekar mot jordens tyngdpunkt, vilket är avgörande många tillämpningar inom analys av sensoriska data och rörelsespårning. Nedan presenteras formel (3), som detaljerat beskriver den matematiska processen för normaliseringen. Detta möjliggör en konsekvent och korrekt omvandling av accelerometers mätningar till enhetsvektorer.



Figur 6 Accelerometer data i varje sampeltid (dt)

För givna accelerometer värden $[a_x, a_y, a_z]$ enligt figur 6, vid varje sampeltid (dt), är den normaliserade accelerometer-vektorn \vec{a} visas i formel 3 nedan [30]:

$$\vec{a} = \frac{[a_x, a_y, a_z]}{\|[a_x, a_y, a_z]\|} \quad (3)$$

Variabler representerar följande:

\vec{a}	normaliserade vektorn
$[a_x, a_y, a_z]$	Komponenterna som mäts av accelerometer
$\ [a_x, a_y, a_z]\ $	betecknar den euklidiska normen, som ger längden av vektorn i rummet.

Beräkning av roll- och pitchvinklar:

De viktiga rådata som erhålls från accelerometer kan hjälpa för att bestämma enhetens orientering i rummet. Genom analys av rådata beräknas rollvinkeln (ϕ) och pitchvinkeln (θ) från den normaliserade accelerometervektorn [18,2]. Denna beräkningsprocess använder komponenterna i accelerometerdata (a_x, a_y, a_z) för att beskriva enhetens orientering i rummet vid varje sampeltid (dt). Deras kombination ger en fullständig bild av enhetens riktning och lutning i förhållande till gravitationen. När enheten är stationär, och både a_x och a_y är lika med noll kommer roll och pitch att vara noll enligt ekvationerna (4, 5).

Rollvinkeln, som är runt x-axeln, indikerar enhetens lutning åt sidan och beräknas i radianer med enligt formeln (4). Denna beräkning och dess tillämpning inom olika kontrollsystem, inklusive sportteknologi, stöds väl av källor som diskuterar användningen av accelerometerdata för att bestämma roll- och pitchvinklar [27, 2]. Även om data ger insikt om roll och pitch, krävs ytterligare referenspunkter, som magnetiskt norr för att fullständigt bestämma yaw-vinkeln, rotationen runt z-axeln.

$$\phi = \arctan2(a_y, a_x) \quad (4)$$

Pitchvinkeln, som är runt y-axeln, och indikerar enhetens lutning framåt eller bakåt, beräknas i radianer med följande formel (5) se [27]:

$$\theta = \arctan2(-a_x, \sqrt{a_y^2 + a_z^2}) \quad (5)$$

Det är viktigt att förstå att den specifika sekvensen av rotationer påverkar beräkningen av Euler-vinklar, vilket understryker rotationssekvensens betydelse för bestämningen av Euler-vinklar enligt förklaringar på Physics Stack Exchange [36]. Denna insikt betonar rotationssekvensens betydelse för exakt bestämning av Euler-vinklar. Vid beräkningen av rollvinkel (ϕ) och pitchvinkel (θ), baseras ekvationerna på accelerationen mätt längs x, y och z axlarna vid varje sampeltid (dt), angiven i (m/s^2). Här representeras a_x , a_y och a_z respektive accelerationer, där a_z inkluderar gravitations effekt när enhetens är stilla. Funktionen $\arctan2$ används för att bestämma vinklarna i intervallet $-\pi$ till π , mätt i radianer (rad), vilket tillhandhåller en precisa representation av enhetens orientering i rummet.

Konvertera roll och pitch vinklar till kvaternion:

När roll- och pitchvinklarna har beräknats omvandlas de till kvaternioner för att representera enhets orientering i 3D. Yaw-komponenten, som representerar rotationen runt z-axeln, kan inte bestämmas med enbart accelerometerdata och antas därför vara noll. Kvaternionen som representerar rotationen visas nedan i formel (6) som beskrivits i [24], kapitel C:

$$q = [q_0, q_1, q_2, q_3] \quad (6)$$

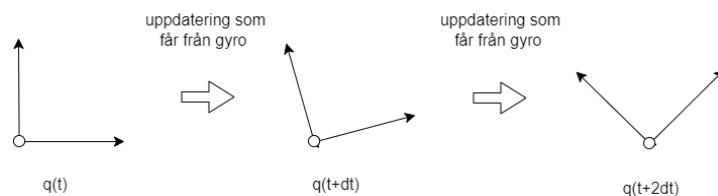
Där kvaternionens komponenter ges av:

- $q_0 = \cos(\frac{\theta}{2})\cos(\frac{\phi}{2})$
- $q_1 = \sin(\frac{\theta}{2})\cos(\frac{\phi}{2})$
- $q_2 = \cos(\frac{\theta}{2})\sin(\frac{\phi}{2})$
- $q_3 = \sin(\frac{\theta}{2})\sin(\frac{\phi}{2})$

Kvaternion, q , representerar enhetens orientering i 3D-rummet i förhållande till ett fast koordinatsystem, under antagandet att det inte finns någon initial rotation.

3.2.4 Integration av kvaternioner och gyroskopdata

Genom att kombinera kvaternion orienteringar med gyroskopdata uppnås hög noggrannhet vid uppskattning av enhetens rumsliga orientering. Detta uppnås genom att använda mätningar av vinkelhastigheten från gyroskopet. Figur 7 illustrerar hur kvaternion rotationen kontinuerligt uppdateras baserat på gyroskopdata över tidsintervaller.



Figur 7 Uppdatera q -rotationen baserat på Gyrodata för varje dt

3.2.5 Uppdatering av global rotation

Vinkelhastighet från gyroskop:

Vinkelhastigheten runt var och en av de tre axlarna i koordinatsystemet representeras av vinkelhastighetsvektorn $\omega = [\omega_x, \omega_y, \omega_z]$, som tillhandahålls av gyroskopet. Dessa uttrycks vanligtvis i radianer per sekund. Med denna data kan enhetens rörelse uppskattas med hög noggrannhet.

Representation av kvaternioner:

Som tidigare nämnt att kvaternion ger en robust orientering i 3D - rummet. En kvaternion $q = [q_0, q_1, q_2, q_3]$ representerar objektets orientering i rummet, där q_0 den skalära komponenten och $[q_1, q_2, q_3]$ vektordelen. Skalärkomponenten representerar rotationsvinkeln medan vektordelen representerar rotationsaxeln. Kvaternioner löser problemet med gimbal-lock, som kan uppstå med andra orienteringsrepresentationer som Euler-vinklar [26].

Derivata av kvaternioner:

Uppdateringar av orientering i realtid är avgörande för tillämpningar som robotteknik, flyg, samt virtuell och förstärkt verklighet [22]. Ekvation (14) i [24, 27] beskriver användning av vinkelhastighetsvektorn för att beräkna förändringshastigheten hos en kvaternion (eller kvaternion derivatan). Formel (7) visar kvaternion derivatan. Där \dot{q} är tidsderivatan med enhet (1/s), och $\frac{1}{2}$ är en konstant faktor som säkerställer att vinkelhastigheterna inverkan korrekt beräknas och appliceras på kvaternionen.

$$\dot{q} = \frac{1}{2} q \otimes \omega' \quad (7)$$

Där ω' är vinkelhastighets kvaternionen, som är $\omega' = [0, \omega_x, \omega_y, \omega_z]$, och \otimes betecknar kvaternion multiplikationen. Det första elementet är noll eftersom ω' indikerar en ren rotation utan en skalär komponent.

Multiplikation av Kvaternioner matris:

Efter bestämning av \dot{q} , används kvaternion multiplikation för att uppdatera kvaternionens nya värde genom att kombinera den nuvarande orienteringen med nya rotationsdata. Detta utförs enligt formel (8) se [30]:

$$q \otimes \dot{w} = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ w_x \\ w_y \\ w_z \end{bmatrix} = \begin{bmatrix} -q_1\omega_x - q_2\omega_y - q_3\omega_z \\ +q_0\omega_x + q_2\omega_z - q_3\omega_y \\ +q_0\omega_y - q_1\omega_z + q_3\omega_x \\ +q_0\omega_z + q_1\omega_y - q_2\omega_x \end{bmatrix} \quad (8)$$

Kvaternion uppdatering:

För att erhålla enhets nya orientering vid tiden $t + \Delta t$, integreras dess derivata över tiden för att uppdatera kvaternionen. Där Δt är tidssteget mellan uppdateringarna. En enkel Eulers integrationsmetod kan användas enligt detta formel 9 se [27,28]:

$$q(t + \Delta t) = q(t) + \dot{q}(t) \cdot \Delta t \quad (9)$$

Normalisering:

Kvaternionen bör normaliseras efter varje uppdatering för att bevara dess enhetslängd, då numeriska fel kan orsaka avvikelse. Normaliseringsprocess visas i formel 10, [27,28, 30]:

$$q_{normalized} = \frac{q}{\|q\|} \quad (10)$$

Här $\|q\|$ betecknar Kvaternion norm.

3.2.6 Användning av Accelerometer, Gyroskop och Magnetometer (MARG)

I avsnitt 3.1 diskuterats vikten av sensorfusion och dess roll i verkliga tillämpningar såsom flyg- och robotteknik, där precision och orienteringsnoggrannhet är avgörande. I projektet används sensorfusion av accelerometer, gyroskop och magnetometer (MARG) för att uppskatta orientering. Genom att kombinera data från varje sensor kan ett objekts orientering i rummet förstås fullständigt. Den fullständiga matematiska förklaringen är följande:

Data från sensorerna:

- **Accelerometer:** Beräknar linjär acceleration, inklusive gravitation. Ger accelerationsvektorn $\vec{a} = [ax, ay, az]$.
- **Gyroskop:** Beräknar rotationshastighet eller vinkelhastighet. Ger vinkelhastighetsvektorn $\vec{\omega} = [\omega_x, \omega_y, \omega_z]$.
- **Magnetometer:** Bestämmer intensiteten och riktning av magnetfältet. Varje komponent i denna vektor $\vec{m} = [mx, my, mz]$ representerar styrka av magnetfält längs x, y, z i tredimensionellt koordinatsystem.

Normalisering av sensordata:

För att omvandla avläsningar från varje sensor till en enhetsvektor normaliseras de enligt formel 11 se [27]:

$$\vec{a}_{norm} = \frac{\vec{a}}{\|\vec{a}\|}, \vec{m}_{norm} = \frac{\vec{m}}{\|\vec{m}\|} \quad (11)$$

Där variabler (m) för magnetometer och (a) för accelerometer

Initiala orienteringsuppskattning:

Processen för att använda data från accelerometer och magnetometer för att bestämma enhets initiala orienteringen.

a. Från accelerometer

Utnyttja data från accelerometer för att bestämma enhets lutning i förhållande till jordens gravitation är avgörande. Detta uppnås genom att mäta accelerationen längs enhetens tre axlar (x, y, z). När enheten är stilla motsvarar dess acceleration gravitationskraften $9.81m/s^2$, vilket innebär att den pekar direkt mot jordens centrum. Accelerometer kan inte skilja mellan acceleration som orsakad av enhetens egna rörelser eller av gravitationskraften. Genom att noggrann analys av dessa mätvärden kan vi dock bestämma pitch (θ) och roll (ϕ) med hjälp av formel (4, 5).

b. Från magnetometer:

Givet en global rotation (q) som representerar orienteringen av en enhetsrotation i rummet och en normaliserad magnetometers mätvektor $\vec{m} = [m_x, m_y, m_z]$, roterar denna metod \vec{m} från den lokala ramen (enhetsrotation) till den globala ramen (Earth Frame) med formeln som ekvation 45 i [28]:

$$h = q \times [0, m_x, m_y, m_z] \times q^* \quad (12)$$

Rotationen tillämpas genom att multiplicera den rena kvaternionen med q och sedan med konjugatet av q (betecknad som q^*) Detta är en standardmetod för att rotera en vektor med en kvaternion.

3.2.7 Kvaternion representation:

Skapa en kvaternion genom att konvertera roll-, pitch- och yaw-Eulervinklar:

Bestäm Halvvinklarna:

Eftersom kvaternionoperationer effektivt dubblar rotationsvinkeln, används halvvinklarna för roll, pitch och yaw vilka är $\frac{\phi}{2}$, $\frac{\theta}{2}$ och $\frac{\psi}{2}$, respektive.

Beräkna Komponenterna för Kvaternioner:

Följande formler 13 kan användas för att beräkna kvaternionen, som beskrivits i kapitel C i [24].

$$q = [q_0, q_1, q_2, q_3]:$$

$$q_0 = \cos(\phi/2) \cdot \cos(\theta/2) \cdot \cos(\psi/2) + \sin(\phi/2) \cdot \sin(\theta/2) \cdot \sin(\psi/2) \quad (13)$$

$$\begin{aligned}
q_1 &= \sin(\phi/2) \cdot \cos(\theta/2) \cdot \cos(\psi/2) - \cos(\phi/2) \cdot \sin(\theta/2) \cdot \sin(\psi/2) \\
q_2 &= \cos(\phi/2) \cdot \sin(\theta/2) \cdot \cos(\psi/2) + \sin(\phi/2) \cdot \cos(\theta/2) \cdot \sin(\psi/2) \\
q_3 &= \cos(\phi/2) \cdot \cos(\theta/2) \cdot \sin(\psi/2) - \sin(\phi/2) \cdot \sin(\theta/2) \cdot \cos(\psi/2)
\end{aligned}$$

Dessa formler är resultatet av att använda Euler-formeln för att konvertera Eulervinklar till kvaternioner, samtidigt som rotationsordningen och specifika konventionen som (ZYX eller XYZ) beaktas för att undvika felaktiga resultat och säkerställa korrekt rotation.

Kvaternion uppdatering med gyroskop:

Använd kvaternion integration för att uppdatera kvaternionen med gyroskopdata enligt formel (7, 9). Dessa beskriver hur kvaternionen $q(t)$ uppdateras med Δt genom att använda kvaternion derivatan $\dot{q}(t)$ och ω' , som är kvaternion vinkelhastighet som i ekvation 45 och 48b i [15].

Sensorfusioneringsalgoritm:

Använd data från accelerometern och magnetometern för att justera kvaternionen från gyroskopet med en sensorfusioneringsalgoritm, som Mahony- eller Madgwick-filtret. Detta steg säkerställer noggrann och stabil orienteringsskattning över tid genom att kompensera för eventuella drift.

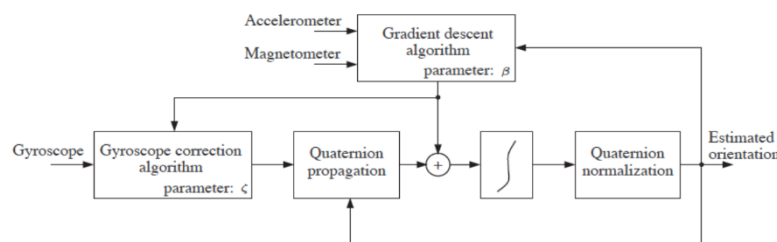
Normalisering av Utdata:

Efter att kvaternionen har uppdaterats bör den normaliseras för att bevara dess enhetslängd, vilket motsvara numeriska fel. Slutlig orienteringsskattningen ges av enligt formel 10 som tidigare nämnts [27,28, 30].

Observera att noggrannheten i MARG-baserad orienteringsskattning beror på kvaliteten på sensorerna och effektiviteten hos sensorfusioneringsalgoritmen. Fusioneringsalgoritmen spelar en kritisk roll i att korrigera drift, framför allt från gyroskopet, vilket gör det möjligt att uppnå en fullständig 3D-orienteringsskattning genom att effektivt kombinera fördelarna från varje sensor samtidigt som deras nackdelar minimeras.

3.3 Madgwick-filtret

En annan populär algoritm för att uppskatta orientering genom sensorfusion är Madgwick-filtret. Figur 8 nedan visar flödesschemat för sensorfusion algoritmen. Det integrerar effektivt information från accelerometrar, magnetometrar och gyroskop för att beräkna en sensors orientering i tre dimensioner [10,11].



Figur 8 Diagram Madgwick filter [11]

3.3.1 Grundläggande om kvaternioner

Tredimensionella rotationer och orienteringar kan representeras av kvaternioner med formeln $q = [q_w, q_x, q_y, q_z]$, där q_w är den skalära komponenten och $[q_x, q_y, q_z]$ är vektorkomponenten. Detta ger ett effektivt och tydligt sätt att beräkna objektens orientering i tredimensionellt.

3.3.2 Multiplikation av kvaternioner

För att kombinera enskilda rotationsrörelser till en enda övergripande rotation, vilket är viktigt för att noggrant uppdatera orienteringsberäkningar baserat på kontinuerligt sensordata, används kvaternion multiplikation. Två kvaternioner, p och q , komponeras genom följande operation visar i matris 14 [31]:

$$q \otimes p = \begin{bmatrix} p_w q_w - p_x q_x - p_y q_y - p_z q_z \\ p_w q_x + p_x q_w - p_y q_z - p_z q_y \\ p_w q_y - p_x q_z + p_y q_w + p_z q_x \\ p_w q_z + p_x q_y - p_y q_x + p_z q_w \end{bmatrix} \quad (14)$$

3.3.3 Integration av gyroskop

Gyroskopdata ger information om hur enhetens orientering förändras över tiden. Denna data integreras för att kontinuerligt uppdatera och förbättra uppskattningen av enhetens orientering [31]. Processen är avgörande för att spåra dynamiska rörelser och justera orienteringen i realtid. Orienteringsförändringar i vinkelhastighet, $\omega = [\omega_x, \omega_y, \omega_z]$, tillhandahålls av gyroskopet. Kvaternionderivata, baserad på ekvation 12 i [28], presenteras som formel 7. Variabeln \dot{q} representerar derivatan av kvaternionen q , vilken beskriver den nuvarande orienteringen. Vektorn $[0, \omega_x, \omega_y, \omega_z]$ innehåller skalära komponenterna samt den reella delen, med ω_x, ω_y och ω_z som representerar vinkelhastighet kring de tre axlarna i rummet. Den första delen (noll) indikerar den skalära delen av en vektor som används för att beskriva en ren rotation utan några skalära (eller "reella") komponenter.

Målet med sensorfusion är att anpassa den uppskattade orientering till magnetometer och accelerometeravläsningar för jordens magnetfält respektive gravitation. Accelerometermatris $f_{acc}(q)$ som visas i formel 15 och magnetometermatris $f_{mag}(q)$ enligt formel 16, som visas i ekvation 25 och 31 i [28], används för detta syfte. Variabler q_w, q_x, q_y, q_z representerar komponenterna i kvaternionen q och a_x, a_y, a_z representerar mätvärden från accelerometer i tre axlar.

$$f_{acc}(q) = \begin{bmatrix} 2(q_x q_z - q_w q_y) - a_x \\ 2(q_w q_x - q_y q_z) - a_y \\ 2(0.5 - q_x^2 - q_y^2) - a_z \end{bmatrix} \quad (15)$$

$$f_{mag}(q) = \begin{bmatrix} 2b_x(0.5 - q_y^2 - q_z^2) + 2b_z(q_x q_z - q_w q_y) - m_x \\ 2b_x(q_x q_y - q_w q_z) + 2b_z(q_w q_x + q_y q_z) - m_y \\ 2b_x(q_w q_y + q_x q_z) + 2b_z(0.5 - q_x^2 - q_y^2) - m_z \end{bmatrix} \quad (16)$$

Dessa formler är avgörande för noggrann och långsiktig orienteringsskattning genom att korrigera eventuell drift, särskilt från gyroskopet, och möjliggöra fullständig 3D-orienteringsskattning genom en omfattande metod som utnyttjar varje sensors styrkor samtidigt som deras brister minimeras.

3.3.4 Gradient descent

För att beräkna gradienten av målfunktionen $f(q)$ enligt ekvationerna 15 och 16 med avseende på kvaternionen används gradient descent för att minska fel i mätningar från accelerometer och magnetometer. Kvaternionen justeras enligt denna gradient för att uppnå en mer exakt orientering som stämmer överens med faktiska sensordata. Gradienten pekar mot den riktning där målfunktionen ökar mest, så genom att röra sig i motsatt riktning kan felet minimeras [24].

Jakobian för Accelerometerdata

Det är avgörande att förstå hur varje kvaternionens komponent påverkar felet i accelerometer data. Matrisen 17 ger partiella derivator av målfunktionen $f(q)$ med avseende på kvaternionens komponenter se ekvation 26 i [28], vilket visar hur små justeringar av kvaternionen påverkar den beräknade orienteringen. För $f_{acc}(q)$ med avseende på $q = [qw, qx, qy, qz]$ [31]:

$$J_{acc}(q) = \begin{bmatrix} -2q_y & 2q_z & -2q_w & 2q_x \\ 2q_x & 2q_w & 2q_z & 2q_y \\ 0 & -4q_x & -4q_y & 0 \end{bmatrix} \quad (17)$$

Jakobian för magnetometerdata

Liknande accelerometerdata illustrerar denna matris kvaternionens inverkan på målfunktionen $f(q)$ baserat på magnetometerdata. Den visar de partiella derivatorna för varje komponent i q , vilket understryker hur förändringar i kvaternionen påverkar orienteringen i förhållande till jordens magnetfält. Med avseende till kvaternionen q visar matrisen 18 nedan Jakobian med magnetometer del, detta visar matrisen 32 i [28] som beskrivs genom att kombinera mätningar från gravitations- och magnetfält är det möjligt att uppskatta en viss orientering av en sensor.

$$J_{mag}(q) = \begin{bmatrix} -2b_z q_y & 2b_z q_z & -4b_x q_y - 2b_z q_w & -4b_x q_z + 2b_z q_x \\ -2b_x q_z + 2b_z q_x & 2b_x q_y + 2b_z q_w & 2b_x q_x + 2b_z q_z & -2b_x q_w + 2b_z q_y \\ 2b_x q_y & 2b_x - q_z - 4b_z q_x & 2b_x q_w - 4b_z q_y & 2b_x q_x \end{bmatrix} \quad (18)$$

Gradient descent för optimering

För att minska felet i uppskattningen av ett objekts orientering implementeras en algoritm för gradient descent. Med en initial approximation och en variabel steglängd (μ) justerar algoritmen iterativt den roterande representationen (kvaternionen) som beskrivs i ekvation (35) i [28]. Det önskade orienterings resultat uppnås genom att minska avvikelserna definierad av målfunktionen f och dess Jakobian (J). Vilket är avgörande för att uppnå optimal orientering.

3.3.5 Uppdatering med gradient descent

Uppdateringssteget för gradient descent utförs med hjälp av Jakobimatriser för att kontinuerligt förbättra orienteringsskattning baserat på sensordata enligt följande steg:

För accelerometersuppdatering:

Gradienten $\nabla f_{acc}(q)$ bestäms genom att multiplacera den transponerade Jakobmatrisen för accelerometerdata $J_{acc}(q)$ med funktionen $f_{acc}(q)$, som visas i matris 19. Den nya kvaternionen q_{new} kan beräknas genom att subtrahera produkten av β och gradient från den aktuella kvaternionen q , enligt ekvation 33 och 34 visar i [28] och i koden [31].

$$\begin{aligned}\nabla f_{acc}(q) &= J_{acc}(q)^T \cdot f_{acc}(q) \\ q_{new} &= q - \beta \nabla f_{acc}(q)\end{aligned}\tag{19}$$

För magnetometersuppdatering [31,27]:

Liknande accelerometersuppdateringen bestäms gradienten $\nabla f_{mag}(q)$ genom att multiplacera den transponerade Jakobmatrisen för magnetometerdata, $J_{mag}(q)$ med $f_{mag}(q)$:

$$\begin{aligned}\nabla f_{mag}(q) &= J_{mag}(q)^T \cdot f_{mag}(q) \\ q_{new} &= q - \beta \nabla f_{mag}(q)\end{aligned}\tag{20}$$

Formlerna (19) och (20) illustrerar hur uppdateringen med data från genomförs med data från både accelerometer och magnetometer. Gradient descent metoden tillämpas för båda uppdateringarna, där β en justeringsparameter som styr konvergenshastigheten. Efter varje uppdatering normaliseras den nya kvaternionen q_{new} för att behålla den som en enhetskvaternion. Denna process leder till en pålitlig orienteringsskattning i tre dimensioner och effektivt minskar felet mellan den beräknade och den faktiska orienteringen som upptäcks av sensorerna, som anges i ekvation 32 i [28].

Normalisering

Efter varje uppdatering ska kvaternionen normaliseras för att säkerställa att den förblir en enhetskvaternion, som föreskrivs i formel 10 [28].

Kvaternion uppdatering

Gradient descent stegen, tillsammans med gyroskopintegration, används för att uppdatera kvaternionen, där β fungerar som en förstärkningsfaktor som styr inverkan av korrigeringen från accelerometern och magnetometern. Med Δt som tidssteget mellan mätningarna, visar koden i [31] en serie steg som realiserar principen bakom följande formel (21):

$$q_{updated} = q + \Delta t(\dot{q} - \beta \nabla f(q))\tag{21}$$

3.4 Horisontellt hoppavstånd:

Detta avsnitt beskriver metoder för att beräkna det horisontella hoppavståndet. Detta görs genom analys av senornas orientering, beräkning av gravitationsvektor, isolering av linjär acceleration, beräkning av förskjutning och avståndsberäkning.

3.4.1 Kvaternion baserad orienteringsoutput

Användning av kvaternion baserad orienteringsoutput från Mahony- och Madgwick filter är en effektiv metod för att bestämma sensorns orientering i tredimensionellt. Representerade som formel 6 [31, 30].

3.4.2 Beräkning av gravitationsvektor:

I det standardiserade gravitationsfältet är värdena för x och y axlarna noll, medan z axelns värde är 9.81 m/s^2 . Därför endast z-komponenten tas, som multipliceras med gravitationsvärdet, efter att den har roterats enligt enhetens rotation. Formeln 22 nedan visar hur man använder elementen i en kvaternion för att rotera en standard gravitationsvektor \vec{g} till sensorns orientering. En standard gravitationsvektor \vec{g} pekar vanligtvis nedåt längs z-axeln med en storlek som motsvarar ungefär 9.81 m/s^2 vid jordensytan. Resultatet är den uppskattade gravitationsvektor sett från sensorns nuvarande orientering, \vec{g}' , vilket kan ses i bilaga B och beskrivas i ekvation 2.30 i [33].

$$\vec{g}' = \begin{bmatrix} 2(q_1q_3 - q_0q_2) \\ 2(q_0q_1 + q_2q_3) \\ q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \times g \quad (22)$$

Där:

- $2(q_1q_3 - q_0q_2)$: bestämmer x-komponenten för den roterade gravitationskraften.
- $2(q_0q_1 + q_2q_3)$: bestämmer y-komponenten för den roterade gravitationskraften.
- $q_0^2 - q_1^2 - q_2^2 + q_3^2$: bestämmer z-komponenten för den roterade gravitationskraften.

3.4.3 Isolering av linjär acceleration

Detta steg är viktigt för att kunna nyttja den linjär acceleration samt för att få korrekt förstå objektens rörelse, särskild vid rörelseanalys. Man måste subtrahera gravitationskraften från accelerometer råa data, vilket görs genom att filtrera bort gravitationens konstanta effekt (ren av jordens tyngdkraft). Följande formel 23 beräknar den verkliga linjära acceleration \vec{a}_{linear} som enheten upplever. Som beskrivs i bilaga B under dataprocessklassen för både Mahony-processen och Madgwick-processen, visar detta [31]

$$\vec{a}_{linear} = \vec{a}_{raw} - \vec{g} \quad (23)$$

Där \vec{a}_{linear} är den acceleration efter borttagning av gravitation (m/s^2), \vec{a}_{raw} representerar den råa data som samlas in från accelerometern och inkluderar både linjära acceleration och tyngdkraftens acceleration (m/s^2), och \vec{g} det är tyngdkraftvektorn, som riktat mot jordens centrum m/s^2 .

3.4.4 Beräkning av förflyttning

Bilaga B beskriver implementeringen av denna process, antagande att den linjära accelerationen är konstant över ett litet tidsintervall Δt . Då beräknas förflyttningen \vec{d} i varje axel med den kinematiska ekvationen 24:

$$\vec{d} = \frac{1}{2} \vec{a}_{linear} \Delta t^2 \quad (24)$$

Detta resulterar i en förflyttningsvektor $\vec{d} = [d_x, d_y, d_z]$ där varje komponents beräknas enligt följande formler. Att räkna ut förflyttning i varje riktning under en tidsperiod ger oss en komplett bild av objektens rörelse enligt formel 25:

$$(25)$$

$$d_x = \frac{1}{2} \vec{a}_{linear,x} \Delta t^2, d_y = \frac{1}{2} \vec{a}_{linear,y} \Delta t^2, d_z = \frac{1}{2} \vec{a}_{linear,z} \Delta t^2$$

3.4.5 Avstånd beräkning

För att få ett enda mått på det totala avståndet, oberoende av rörelseriktning, beräknas det skalära avståndet genom att omvandla vektorförflyttningen till ett skalärt värde med hjälp av följande formel 26:

$$d = \sqrt{d_x^2 + d_y^2 + d_z^2} \quad (26)$$

Där d representerar den totala avstånd, och $d_x^2 + d_y^2 + d_z^2$: Representerar förflyttning vektorn längst respektive axel.

3.4.6 Kumulativa avstånd

Beskrivning av att ackumulera avstånd här menas att summera avstånd för varje intervall för att få en hel sträcka över tiden. Det slutliga resultatet blir kumulativ och representerar det sammantagna avstånd eller sträckan som enhetens har färdats, beräknat från den linjära accelerationen över tiden.

Rörelseekvation:

För att få sträckan (s) som en funktion av tiden (t) används derivatan av ekvationen ($\frac{1}{2} at^2$) genom integration. Från definitionen av acceleration och integrerar den två gånger. Acceleration (a) är hastighetsändringen (v) i förhållande till tiden (t), och hastigheten är förändringen av förskjutning (s) i förhållande till tiden som formel 27 - 29 visar, det visa i ekvation 15 och 16 i [29].

$$s = \int v dt = \iint a dt dt \quad (27)$$

Integrera hastighet för att hitta sträckan

$$v = \int a dt = at + C$$

$$s = \int v dt = \int at dt = \frac{1}{2} at^2 + D \quad (28)$$

Bestämning av konstanten C sker genom att anta start hastighet (v_0) vid tiden (t_0) som motsvarar startpunkt. Vilket blir följande

$$C = v_0 = 0$$

Vilket blir $v = at$. Där D är också konstanten av integrationen. Bestämning av konstanten D sker genom att anta startposition (s_0) vid tiden ($t = 0$) $D = 0$ motsvarar startpunkt från origo. Vilket blir följande:

$$D = s_0 = 0$$

$$s = \frac{1}{2} at^2 \quad (29)$$

4 Metod

För att uppnå projektets mål krävs specifik hårdvara som InnoX har tillhandahållit. Dessa komponenter är avgörande för mätning, datahantering och analys av horisontella hoppavstånd data. Det säkerställer att den nödvändiga utrustningen och teknologin finns tillgänglig för att utföra ett effektivt uppdrag och undersöka om användning av endast dessa utrustningar kan uppfylla projektets mål.

Informationen som samlats in i detta avsnitt kommer att bidra till att definiera ytterligare forskningsmetoder och tillvägagångssätt för vårt projekt. Det innefattar att etablering av lämpliga metoder och vägledning för slutsatser i fortsatt forskning. Denna information kommer även att bidra till att vara vägledande för projektets riktning och understryka dess relevans och betydelse inom område för idrottsteknologi och prestationsövervakning. Utvecklingen av mätalgoritmen för rörelseanalys baseras på data från (IMU) och magnetometer. För att beräkna det horisontella hoppavståndet, följs dessa steg:

- **Förbättring av mätningssystemet:**

Utveckla en enkel metod för att mäta horisontella hopp. Identifiera möjliga förbättringar i de nuvarande analysmetoderna som används i Inno-x produkten inom idrottsområdet.

- **Integration av teknik och biomekanik:**

Illustrera hur integrationen av sensorer, såsom MEMS-enheter, kan förbättra förståelsen för mänskliga rörelser. Främja samverkan mellan teknik och sportvetenskap genom att beskriva hur dessa sensorer kan integreras med biomekanik.[5, 6].

- **kalibreringsteknikers effektiviteten:**

Utvärdera hur kalibreringsmetoder för Hard Iron och Soft Iron påverkar noggrannheten i magnetometeravläsningar och beskriv hur dessa metoder kan förbättra systemets precision [22].

- **Kombination av sensorer för rörelseanalys:**

Förklara fördelarna med att använda olika sensorer (accelerometer, gyroskop, magnetometer) för att analysera rörelser. Ge en översikt över hur kombinationen av olika sensordata kan ge insikter om mänskliga rörelser.

- **Utvärdering:**

Genomföra enkla tester med en grupp individer från olika åldersgrupper för att utvärdera systemets effektivitet.

4.1 Experimentell uppsättning

Deltagaren

Prototypen ska placeras strax över fotknölen på den nedre delen av foten innan deltagarna utför ett horisontellt hopp. Denna placering är vald för att effektivitet samla informationen om idrottarens rörelsemönster under hoppet.

Mätning av hoppsträcka

Den faktiska hoppsträckan mäts med en tumstock, parallellt med insamlingen av sensordata. Detta möjliggör jämförelse mellan de uppmätta avstånden och de avstånd som beräknats baserat på sensordata.

4.2 Hårdvara:

Sensormodul

Projektet använder en STEVAL-MKI217V1 sensormodul (9-axlig), som innehåller två MEMS-enheter: en LSM6DSOX (accelerometer och gyroskop) och en LIS2MDL (magnetometer). Dessa enheter ger en omfattande bild av rörelser i tre dimensioner (x-, y-, och z-axlarna). Den 9-axliga kapaciteten underlättar insamling av data som täcker både acceleration och rotation samt orientering i rummet, vilket är kritiskt för analys och beräkning av horisontella hoppavstånd. Se figur 12 i bilaga 9.1 för en översikt över den sensormodul som används under arbetet [6, 7].

Utvecklingskort

För att förbereda experimentet, placeras STEVAL-MKI217V1-sensormodulen på ett STEVAL-MKI109V3-utvecklingskort. Denna placering underlättar enkel integration och anslutning till datorn via USB-kabel, som visas figur 12 i bilaga 9.1. Etableringen av seriell kommunikation mellan sensorn och PC:n möjliggör smidig dataöverföring och realtidsläsning vid utförande av horisontella hopp, med sensorns frekvens inställd på 416Hz för optimal datainsamling [9].

4.3 Programvara

Datainsamling och analys utförs med hjälp av Python-skript och program i Visual Studio Code (VSCode), en mångsidig kodredigerare som stöder flera programmeringsspråk och verktyg, inklusive Unico-GUI [19, 8]. MATLAB-programmet, specialiserat på matematisk analys och beräkning, spelar en viktig roll i projektet genom att underlätta statisk analys av slutliga hoppvärden. Detta bidrar till utvärderingen av algoritmernas prestanda som används i projektet [37].

Mjukvarufilter Mahony- och Madgwick-algoritmerna används för att genomföra sensorfusion, vilket leder till robust och tillförlitlig beräkning av sensorns orientering i tre dimensioner. Att integrera data från accelerometer, gyroskop, och magnetometer, uppskattar dessa algoritmer orienteringen på ett sätt som är mindre känsligt för störningar från sensorerna. Detta bidrar indirekt till förbättrad noggrannhet och tillförlitlighet i hoppresultaten genom att erbjuda en mer konsekvent och korrekt tolkning av rörelsedata [11, 15, 16].

4.4 Datainsamling och analys

Seriell Kommunikation

Sensorn och den personliga datorn (PC) använder ett seriekommunikationsprotokoll för att kommunicera med varandra, med dataöverföring på 9600 bitar per sekund (baudrate). Denna hastighet balanserar effektiv dataöverföring med systemets förmåga att processa informationen.

4.5 Statistisk analys

Syftet med den statistiska analysen är att identifiera signifikanta skillnader i hur olika algoritmer estimerar hoppavstånd baserat på data från en grupp av 10 individer. Varje individ bidrog med data från ett hopp, med resultat från både Mahony- och Madgwick-filtrerna.

Denna process kräver ett kraftfullt verktyg som Matlab för att noggrant bearbeta och utvärdera de insamlade datan, för att identifiera signifikanta skillnader i algoritmernas prestanda för hoppavståndsestimering [37].

4.5.1 Deskriptiva statistiken

Med hjälp av Matlab beräknas medelvärdesdifferens, P-värde, och T-statistik för att ge insikt i datans huvudsakliga egenskaper. Denna grundläggande analys hjälper till att förstå datans fördelning och eventuella tydliga mönster, vilket lägger grunden för en mer ingående jämförelse av filtrernas prestanda [23].

4.5.2 Statistiska tester

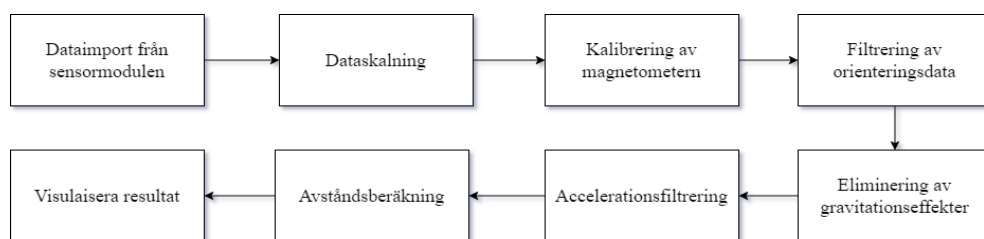
Analysen avser att statistiskt utvärdera och jämföra prestandan mellan Mahony jämfört med Madgwick, samt olika användningsfall av magnetometern (Mag) inom Mahony-filter. För detta ändamål används Matlabs inbyggda ttest-funktion [37], ett parat t-test, som är väl lämpat för jämförelser mellan två beroende dataset. Det parat t-testet är särskilt relevant i detta experiment eftersom det omfattar situationer där varje deltagare utför ett enskilt hopp, och därmed erhålls jämförbara dataset för både Mahony- och Madgwick-filtrerna från just detta hopp. På så sätt kan varje deltagares resultat från de två olika filtrerna betraktas som direkt jämförbara, parvisa observationer. Det parat t-testet används för att bedöma om de observerade skillnaderna i mätfel mellan de två filtreringsinställningarna är statistiskt signifikanta, eller om de kan anses som slumpmässiga variationer. Ett p-värde under 0.05 anses indikera att skillnaderna i filterprestanda är statistiskt signifikanta, vilket innebär att de inte endast är ett resultat av slumpmässiga variationer [23].

4.6 Beräkning och jämförelse

För att verifiera mätningarnas noggrannhet jämförs de beräknade hoppavstånden, som erhållits från sensordata, med de faktiska hoppavstånden mätta med tumstock. Denna jämförelse gör det möjligt att identifiera avvikelser eller mönster som kan avslöja biomekaniska skillnader bland deltagarna, vilket i sin tur ger värdefulla insikter om deras hoppresultat.

4.7 Utvecklingsprocess:

Figur 9 presenterar en omfattande översikt över de centrala stegen i utvecklingsprocessen. Detta illustrativa diagram fungerar som en vägledning för läsaren genom projektets olika steg och underlättar förståelsen för hur varje del bidrar till att uppnå det önskade slutresultatet.



Figur 9 Systemöversikt diagram

Bearbetning samt datasamling

Rådata hämtas från accelerometer, gyroskop, och magnetometer med hjälp av UNICO-GUI-programvaran [8]. Därefter lagras dessa data för vidare bearbetning och visualisering i Python.

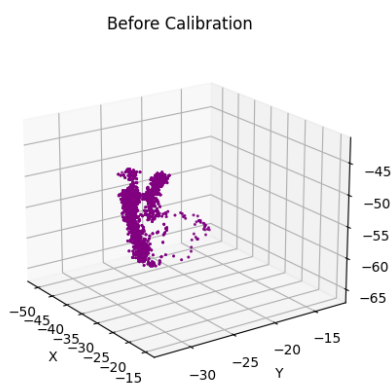
Kalibrering av magnetometern

Kalibrering av magnetometern är en kritisk process för att korrigera magnetiska störningar i sensorerna. Metoderna för Hard iron- och Soft iron-kalibrering har detaljerats. Dessa metoder korregerar fel i magnetiska sensorer, såsom de som finns i integrerade enheter, för att säkerställa att IMU-enheter levererar exakta data. Korrekta avläsningar är beroende av denna kalibrering, särskilt för navigation och orientering. Hard iron-kalibrering riktar sig mot att korrigera statiska magnetfält, medan Soft iron-kalibrering fokuserar på att justera för förvrängningar i magnetfältet som orsakas av närliggande material eller objekt. Dessa steg är avgörande för att upprätthålla sensoravläsningarnas noggrannhet och för att undvika felaktiga resultat orsakade av magnetiska interferenser [22].

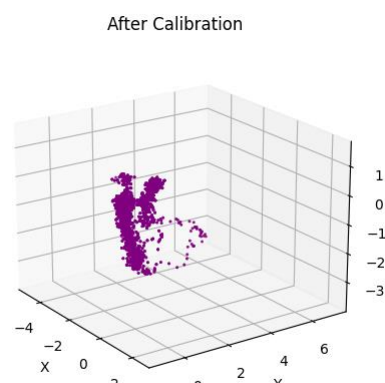
Kalibreringsprocessen för magnetometer sker genom Hard Iron-kalibrering följt av Soft Iron-kalibrering. Först plottas magnetometeravläsningarna i tre dimensioner för att identifiera datapunkternas centrum, vilket visar Hard Iron-förvrängningens offsetvärden för axlarna X, Y, och Z. Dessa värden subtraheras från avläsningarna för att korrigera potentiella fel. I det andra steget, beräknas den genomsnittliga radien från centrum till datapunkterna längs de tre axlarna för att upptäcka avvikelser. Avslutningsvis justeras avläsningarna genom beräkning och tillämpning av skalfaktorer för varje axel baserat på dessa avvikelser för att åtgärda Soft Iron-effekten.

4.7.1 Korrigering och konvertering av magnetometer

För att garantera en korrekt datainsamlingsprocess och få tillförlitliga mätdata från sensorn, krävs kalibrering av magnetometern. Detta innefattar korrigering av störningar och avvikelser i sensoravläsningarna som kan uppstå under hopp, särskilt vid riktningssuppskattningar. I detta fall behöver inga andra sensorer än magnetometern kalibreras, då den används enskilt för att uppskatta orientering. Magnetometern kan även användas för att bestämma geografiska positioner och som en del i navigationssystem för att ange riktning i förhållande till jordens magnetfält. Kalibreringen minskar fel orsakade av olika miljöfaktorer, vilket ökar noggrannheten i sensorernas avläsningar [22]. Magnetometerdata kombineras med data från IMU-sensorer, och nedan följer en detaljerad förklaring med en illustration av denna kalibreringsprocess.



Figur 10 Visar värdena från den okalibrerade magnetometern



Figur 11 Visar värdena från den kalibrerade magnetometern

Den tredimensionella plotten, som figur 10 illustrerar magnetometer-värden längs X-, Y- och Z-axlarna, med uppenbara förskjutningar och förvrängningar orsakade av både Soft och Hard iron effekter, vilka påverkar mätdataens kvalitet. Hard iron-förvrängningar indikeras av att datapunkterna inte är centrerade runt origo, och Soft iron-förvrängningar leder till den icke-sfäriska formen på datapunktmolnet. Sensorernas orientering i rummet återspeglas inte korrekt i denna rådata, som är i 3D-kartesiska koordinater.

Figur 11 illustrerar en 3D plot där en tydlig förbättring i sensordata syns. Datapunkterna är koncentrerade runt origo, vilket visar på en minskning förskjutningarna som fanns före kalibrering som i figur 10. Det syns att effekterna av 'Soft iron', som orsakar skalningsfel och förvrängningar som orsakades av olika anledningar typ av systematiska fel, har korrigerats. Dessutom visar den minskning av datapunkter att förvrängningar från Hard iron, som orsakar förskjutningar i sensorsdata har minskat. Resultatet är en bättre datamängd som ger en mer pålitlig representation av sensorns riktiga orientering och rörelse i rummet. som beskrivs i underavsnitt 2.2.4.

4.7.2 Orienteringsuppskattning (Mahony - Madgwick)

Efter att ha korrigerat accelerometer-, gyroskop- och magnetometervärdena som erhållits från sensormodulen, inklusive skalning och konvertering för acceleration och gyrodata, samt kalibrering för magnetometerdata, matas de in i ett orienteringsfilter. Detta filter tar först accelerationens vektor (som representerar den gravitationella vektorn i en statisk position) för att beräkna den globala orienteringen. Sedan används gyroskopdata (vinkelhastighet) vid varje

steg för att uppdatera rotationen av den globala orienteringsvektorn (q). Som figur 13 i bilaga 9.1 illustrerar, visar ett exempel på accelerationsdata före och efter tillämpningen av orienteringsfilter. Gravitationen elimineras från accelerationsdata genom att rotera gravitationsvektorn med den globala rotationen (q). Dessutom filterades accelerations data med hjälp av låg-passfiltret (LPF) som beskrivs i underavsnitt 4.9.3 och i bilaga B för kodimplementering.

4.7.3 Positionsestimering

Följande steg är grunderna i beräkningsprocessen för det horisontella hoppavståndet med hjälp av moduldata här refererar till de råa mätvärdena som erhålls från sensorerna på enheten, inklusive accelerometer, gyroskop och magnetometer. Beskriver hur gravitationsvektorn hanteras och beräkning av linjär acceleration, och slutligen, samla in det totala avståndet över alla samplingstider.

- Steg 1:** Få den globala rotationen (q) av modulen med antingen Mahony- eller Madgwick filter.
- Steg 2:** Roter gravitationsvektorn enligt q -vektorn enligt formel 22 i (underavsnitt 3.4.2).
- Steg 3:** Eliminera den roterade gravitationsvektorn från accelerations data som mottagits från modulen för att få den linjära accelerationen av formel 23 i (underavsnitt 3.4.3).
- Steg 4:** Beräkna förflyttning för varje axel (x, y, z) enligt formel 24 och formel 25 för att beräkna förflyttningen i de specifika axlarna (x, y, z). Den totala skalära distansen beräknas med formel 26.
- Steg 5 och 6:** Beräkning och ackumulering av den totala distansen över tiden. Kumulativt avstånd beräknas genom att ackumulera avstånd för varje intervall (dt) för att kunna få den totala sträckan övertiden enligt avsnitt 3.5.

4.7.4 Visualisering

Med hjälp av Matplotlib, ett Python-bibliotek, visualiseras resultaten genom att plotta förändringar i global acceleration jämfört med förändringar i lokal acceleration och gyroskopvärden. Detta ger en djupare förståelse för rörelsedynamiken.

4.7.5 Utvärdering

Tester (Mahony-Madgwick) har utförts med att använda UNICO-GUI-programmet [8] för att spara all data från magnetometer, accelerometer och gyro under horisontella hopp i en textfil för att analyseras med hjälp av orienteringsuppskattning och positionsuppskattningsalgoritmer.

Utvärderingsfas omfattar flera steg:

- Deltagaren utför ett horisontellt hopp, vilket innebär att hoppsträckan inte är förutbestämd för att utvärdera enhetens prestanda vid olika avstånd och olika hoppstyrkor. Hoppstyrka här tyder på den kraft som används vid ett hopp för deltagarna, vilket innebär att ett längre hopp kräver större ansträngning jämfört med ett kortare hopp.
- Hoppavstånd registreras i en textfil med hjälp av Uncio-GUI [8].
- RMSE beräknas genom att jämföra dessa beräknade värden baserade på sensorernas data efter nödvändig bearbetning, med manuellt uppmätta data.

- Nödvändiga jämförande analyser genomförs, där skillnader mellan olika scenarier utforskas, såsom skillnader mellan Mahony- och Madgwick-filer.

4.8 Matlab statistik

Definiera data

- A, B: innehåller beräknade hoppavstånd som algoritmerna visade för tio deltagare. Varje elements i vektor representerar en individ hoppavstånd i centimeter.

Utföra parat t-test

- Skriptet använder en MATLAB inbyggd funktion för ” ttest (A, B)” för att utföra ett parat t-test mellan de olika algoritmkonfigurationer. Ett parat t-test används i detta experiment för att jämföra mellan två uppsättningar (algoritmer) av relaterade data (det verkliga avståndet) för samma individer, se bilaga (avsnitt 9.8).

4.9 Kodöversikt för beräkning av horisontellt hoppavstånd

Syftet med Python-koden är att använda data från IMU och magnetometer för att bestämma det horisontella hoppavståndet. Detta innefattar läsning av data från sensorerna, filtrering för att uppskatta orienteringen, och beräkning av förflyttningen horisontellt.

4.9.1 Systemöversikt

Datasamling genomförs genom insamling av data från accelerometern, gyroskopet (IMU) och magnetometern. För att hantera datan används Mahony- och Madgwickfilter för sensorfusion, vilket hjälper till att uppskatta orientering genom att eliminera gravitationskomponenten från accelerometerdata. Avståndsberäkning utförs med bearbetade accelerometerdata för att bestämma den horisontella hoppdistansen. Slutligen sker visualisering av data genom användning av olika plottning verktyg för att analysera sensordata över tid.

4.9.2 Beskrivning av Python klasser

- **Sensorplattan klass:**
Denna klass handlar om att konfigurera och hantera kommunikationen mellan programmet och en sensorkort. Den underlättar datautbyte, konfiguration och initiering av sensorkortet, vilket säkerställer korrekt interaktion och funktion av systemet se Bilaga A.
- **Dataprocessor-klass:**
För att initiera Mahony- och Madgwick-filer, bearbetas inkommande data genom dessa filter för att sedan beräkna gravitationsvektorn. Denna vektor subtraheras från accelerometerdata för att erhålla linjär acceleration. Därefter tillämpas ett lågpasfilter på den linjära accelerationen. På basen av den filtrerade accelerationen beräknas avståndet, och slutligen kontrolleras om hoppet är slutfört (Threshold). Denna process beskrivs i Bilaga B.
- **Data Visualizer-klass:**
Tillhandahåller metoder för att plotta sensordata såsom accelerometer, gyroskopdata för mer beskrivning för kod se Bilaga C.

- **MainApp-klass:**

Hanterar programmets flöde, inklusive att läsa data och anropa metoder för bearbetning och visualisering som i Bilaga D.

4.9.3 Sammanfattning av Python-algoritmen

Sensorfusionsalgoritmer:

Mahony och Madgwick algoritmer kombinerar data från accelerometrar, gyroskop och magnetometer för att ge en stabil och noggrann uppskattning av sensors orientering, vilket är avgörande för att korrekt eliminering av gravitationen från accelerometerdata.

Beräkning av linjär acceleration:

Gravitationskomponenten subtraheras från den totala accelerometer data för att få linjär acceleration. Detta steg är nödvändigt för att isolera rörelsen från gravitationens statiska kraft.

Lågpassfiltrering:

Ett lågpassfilter appliceras på data för linjär acceleration för att minska brus och jämna ut signalen, vilket förbättrar noggrannheten i efterföljande beräkningar av avstånd som koden visar i Bilaga B. Det beskrivna steget i följande:

- a) Filtret uppdaterar iterativt accelerationens x-, y- och z-komponenter.
- b) En större andel av den nya avläsningen används om skillnaden i acceleration mellan på varandra följande avläsningar är större än en förbestämd tröskel-värde.
- c) Om förändringen är liten, jämnar filtret effektivt ut snabba förändringar och bevarar mer av det tidigare värdet.
- d) En extra kontroll sätter extremt låga accelerationsvärden till noll för att förhindra drift i avståndsberäkningen orsakad av minimalt sensorbrus.

Beräkning för förflyttning

Detaljerad beskrivning av metoden för att beräkna horisontell hoppdistans från accelerationsdata, med beaktande av gravitationens eliminering. Processen omfattar diskretisering och integration av acceleration över tidsintervaller samt tillämpning av tekniker för att fastställa avslutet på ett hopp se bilaga B.

5 Resultat

Detta kapitel presenterar de experimentella resultaten som erhållits med hjälp av de tidigare beskrivna teknikerna. Projektet innefattar flera uppgifter för att effektivt kunna uppnå sina mål. I denna del analyseras resultaten från användningen av Mahony- och Madgwick-filtrer, inverkan av magnetometerdata på noggrannheten i Mahony-filtret samt skillnaderna mellan kalibrerad och okalibrerad magnetometerdata. I själva verket syftar projektet inte till att granska algoritmerna i sig, utan snarare att mäta avståndet för horisontella hopp och avgöra vilket av dessa filter som ger det mest tillförlitliga resultatet. För att säkerställa resultaten kommer en statistisk analys av alla data att genomföras för att avgöra om resultaten är signifikanta eller slumpmässiga. Denna analys innefattar beräkningen av p-värden och t-värden för att fastställa algoritmernas signifikans.

5.1 Utvärdering av Mahony - Madgwick

I detta avsnitt utförs en utvärdering av filtreringsalgoritmerna Mahony och Madgwick, med data från accelerometer och gyroskop. Studiens mål är att utvärdera dessa algoritmers noggrannhet i mätningar av horisontella hopp. Deltagarna genomförde hopp-tester, och de faktiska hoppavstånden registrerades. Tabell 1 jämför dessa faktiska avstånd med de avstånd som uppmätts med hjälp av Mahony- och Madgwick-filtreringsalgoritmerna.

Tabell 1 Visar olika mätningar för deltagaren samt fel skillnad mellan algoritmer

Ålder (år)	verkliga hoppavstånd (cm)	Mahony (cm)	Fel (Mahony) (cm)	Madgwick (cm)	Fel Madgwick (cm)
44	97	99.11	+2.11	99.94	+2.94
32	112	111.39	-0.61	111.87	-0.13
69	42	49.13	+7.13	49.15	+7.15
27	91	91.64	+0.64	92.16	+1.16
37	125	131	6	132	7
37	75	78.46	3.46	78.47	3.47
38	80	86.12	6.12	86.37	6.37
30	115	111.73	-3.27	112.28	-2.72
30	108	91.16	-16.84	91.34	-16.66
32	95	93.85	-1.15	95.15	0.15

RMSE

Mahony: 6.61 cm

Madgwick : 6.69 cm

Tabell 2 Statistisk jämförelse av prestanda för filtreringsalgoritmer.

Statistiska värde	Mahony	Madgwick
Hypotes	1	
P-värde	0.0041355	
T-statistik	-3.8128	
Medelvärdesdifferens	-0.514	

Båda filtreringsalgoritmerna presterade något likt, med hög precision. Mahony algoritmen hade lägre RMSE-värde på 6.61 cm jämfört med Madgwicks RMSE på 6.69 cm, vilket indikerar marginella skillnader i precision. För att säkerställa resultaten genomfördes ytterligare en test i form av statistisk analys, vilken visade att deras prestationer skiljer sig åt på ett statistiskt signifikant sätt. Med en medelvärdesdifferens på -0,514, ett p-värde på 0,0041355 och en T-statistik på -3,8128, indikerar analysen att Mahony-algoritm statistiskt sett presterar något bättre än Madgwick- algoritmen. Denna signifikanta skillnad, stödd av ett lågt p-värde, bekräftar att skillnaden i prestanda inte är slumpmässig utan beror på egenskaperna hos de algoritmerna.

Tabell 3 sammanfattar felprocenten och den absoluta skillnaden i felvärden mellan de två filtreringsalgoritmer, vilket möjliggör jämförelse och utvärdering av deras respektive prestanda. Jämförelsen syftar till att belysa hur varje algoritm presterar i olika situationer, vilket möjliggör en noggrann analys av deras precision och frekvens av beräkningsfel. Genom att presentera de kvantitativa skillnaderna i prestanda identifieras situationer där en algoritm kan vara överlägsen den andra.

Tabell 3 Jämförelse av felprocent och absolut felvärdeskillnad mellan filtreringsalgoritmer

Ålder (År)	Fel Mahony (%)	Fel Madgwick (%)	$ abs $ Skillnad fel (Mah - Mad) (cm)
44	2.18	3.03	0.83
32	-0.54	-0.12	0.48
69	16.98	17.02	0.02
27	0.70	1.27	0.52
37	4.80	5.60	1.00
37	4.61	4.63	0.01
38	7.65	7.96	0.25
30	-2.84	-2.37	0.55
30	-15.59	-15.43	0.18
32	-1.21	0.16	1.3

I de flesta fall som tabell 3 uppvisar har Mahony-filter en lägre felprocent jämfört med Madgwick-filter, vilket indikerar en potentiellt högre precision i orienteringsuppskattningen. Skillnaden i fel mellan Mahony och Madgwick varierar över olika åldersgrupper och verkar inte visa något direkt koppling till ålder, vilket antyder att andra faktorer än ålder (t.ex., individuell rörelsedynamik), kan ha påverkat filterprestanda. Trots vissa mindre skillnader i

felprocenten, visar dessa små absoluta felmarginalerna att båda algoritmerna generellt sett presterar likvärdigt.

5.2 Utvärdering av (Mahony utan Mag – Mahony med kali- Mag)

I sammanhanget av deltagarnas hopp utvärderar effektiviteten hos Mahony-filtret för orienteringsuppskattning, med särskilt fokus på magnetometerdataens påverkan. För detta ändamål jämfördes två olika konfigurationer av Mahony-filtret en som enbart använder data från gyroskop och accelerometer, och den andra som inkluderar data från en kalibrerad magnetometer. Deltagare av olika åldrar utförde ett hopp enligt tidigare beskrivna metoder, med målet att fastställa om noggrannheten i orienteringsuppskattningen påverkas avsevärt av att ta bort magnetometeravläsningar från Mahony-filtret. Tillägget av magnetometer till systemet visade sig påverka resultaten i vissa fall, som representeras i tabell 4.

Tabell 4 Deltagarens hoppmätning Mahony utan Magnetometer – Mahony med kalibrerad Magnetometer

Ålder	Den verkliga hoppavstånd (cm)	Mahony utan Mag. (cm)	Fel (Mahony utan Mag.) (cm)	Mahony med Kali Mag. (cm)	Fel Mahony med Kali Mag (cm)
44	97	99.11	+2.11	95.44	-1.56
32	112	111.39	-0.61	105.83	-6.17
69	42	49.13	+7.13	49.00	+7.00
27	91	91.64	+0.64	88.21	-2.79
37	125	131	6.00	130.88	5.88
37	75	78.46	3.46	76.87	1.87
38	80	86.12	6.12	85.48	5.48
30	115	111.73	-3.27	110.93	-4.07
30	108	91.16	-16.84	91.29	-16.71
32	95	93.85	-1.15	105.86	10.86

RMSE

Mahony utan Mag: 6.61 cm

Mahony med kali Mag: 7.6 cm

Tabell 5 Jämförelse av prestanda för Mahony algoritm med och utan kalibrerade Magnetometer

Statistiska värde	Mahony utan Mag	Mahony med kali Mag
Hypotes	0	
P-värde	0.80574	
T-statistik	0.25329	
Medelvärdesdifferens	0.38	

Baserat på RMSE-värde kan man observera att användningen av en kalibrerad magnetometer i Mahony-algoritmen resulterade i en ökning av RMSE från 6.61 cm till 7.6 cm, vilket visar på en försämring i noggrannheten. Enligt statistiska analyser i Tabell 5, som utgår från en nollhypotes (H0) om att det inte finns signifikant skillnad, vilket bekräftar detta resultat av ett p-värde på 0.80574, en T-statistik på 0.2532 och medelvärdesdifferens på 0.38 dessa värden tyder på att skillnaden i noggrannhet mellan dessa algoritm konfigurationer, är ganska lite och

visar inte statistiskt signifikant. I tabell 6, sammanfattas felprocenten och den absoluta skillnaden i felvärden mellan Mahony utan Mag och Mahony med Kali Mag, för att kunna jämföra och utvärdera deras respektive prestanda.

Tabell 6 Bräkning av procentandel felen och den absoluta skillnaden i fel för algoritmer

Ålder	Fel (Mahony utan Mag.) (%)	Fel Mahony med Kali Mag (%)	 abs Skillnad fel (Mahmed kali mag) – (Mah utan mag) (cm)
44	2.18	-1.61	3.67
32	-0.54	-5.51	5.56
69	16.98	16.67	0.13
27	0.70	-3.07	3.43
37	4.80	4.70	0.12
37	4.61	2.49	1.59
38	7.65	6.85	0.64
30	-2.84	-3.54	0.80
30	-15.59	-15.47	0.13
32	-1.21	11.43	12.01

Tabell 6 presenterar att tillägget av en kalibrerad magnetometer till Mahony-filter, resulterar betydande förändringar i felprocenten i vissa fall. I några situationer minskar felen (t.ex., åldern 44 och 32 år), medan en tydlig ökning ses hos personen som är (32 år gammal) med en skillnad på ca 12 cm. Detta visar att inkluderingen av en kalibrerad magnetometer kan ha en signifikant positiv effekt på mätningsnoggrannhet för vissa fall eller rörelsetyper, men kan också leda till ökade fel under andra omständigheter, typ för snabba eller komplexa rörelse kan leda till tidsfördröjning i uppdatering av magnetometerdata. Det understryker behovet av att kalibrera systemet baserat på specifika användningsfall.

5.3 Utvärdering av (Okali mag VS kali mag) med Mahony-filter

Detta avsnitt analyserar prestandan hos Mahony-filtret vid användning av både kalibrerad och okalibrerad magnetometerdata. Användningen av en kalibrerad magnetometer kan generellt förbättra noggrannheten i Mahony-algorithmens mätningar, även om det fortfarande observeras betydande variationer i felmarginalerna. Som visas i tabell 7 förbättras noggrannheten märkbart med kalibrering i vissa fall; till exempel, för en person på 37 år med ett verkligt hoppavstånd på 75 cm, reduceras mätfel med kalibrering till +1.87 cm, jämfört med +25 cm utan kalibrering. Kalibreringen leder till minskade fel marginaler i vissa fall, vilket indikerar en förbättrad förmåga hos algoritmerna att noggrant uppskatta hoppavstånd.

Tabell 7 Deltagarens hoppmätning för okalibrerad magnetometer med Mahony-filter VS kalibrerade magnetometer med Mahony-filter

Ålder	Den verkliga hoppavstånd (cm)	Mahony med okali Mag (cm)	Fel (Mahony med okali Mag) (cm)	Mahony med Kali Mag (cm)	Fel Mahony med Kali Mag (cm)
44	97	106.67	+9.67	95.44	-1.56
32	112	113.87	+1.87	105.83	-6.17
69	42	49.13	+7.13	49.00	+7
27	91	94.29	+3.29	88.21	-3.43
37	125	135.14	10.14	130.88	5.88
37	75	100	25	76.87	1.87
38	80	87.98	7.98	85.48	5.48
30	115	121.81	6.18	110.93	-4.07
30	108	109.70	1.7	91.29	-16.71
32	95	98.44	3.44	105.86	10.86

RMSE

Mahony med okali Mag : 10.02 cm

Mahony med kali Mag: 7.64 cm

Tabell 8 Jämförelse av prestanda för Mahony algoritm med kalibrerade Mag och okali Mag

Statistiska värde	Mahony med okali Mag	Mahony med kali Mag
Hypotes	1	
P-värde	0.021972	
T-statistik	2.7639	
Medelvärdesdifferens	7.724	

RMSE visar för Mahony med okalibrerad magnetometer på 10.02 cm, vilket minskar till 7.64 cm en lägre felmarginal när en kalibrerad magnetometer används. Tabell 8 visar en statistisk jämförelse av prestanda för Mahony-algoritmen med och utan kalibrerad magnetometer, där ett p-värde på 0.021972 och en T-statistik på 2.7639 indikerar en statistiskt signifikant skillnad i noggrannhet mellan de algoritmerna. Samt med Medelvärdesdifferens på 7.724 detta innebär att "Mahony med okali Mag", i genomsnitt, har ett högre värde jämfört med Mahony med kali Mag. Denna signifikans bekräftar hypotesen om att användning av en kalibrerad magnetometer kan påverka positivt till förbättring av algoritmens prestanda. Tabell 9 sammanfattar felprocenten och den absoluta skillnaden i felvärden mellan Mahony med okali Mag och Mahony med kali Mag, för att få mer detaljerad uppfattning av deras respektive prestanda.

Tabell 9 Bräkning av procentandel felen och den absoluta skillnaden i fel för algoritmer

Ålder	Fel Mahony med okali Mag. (%)	Fel Mahony med Kali Mag (%)	 abs Skillnad fel (Mahony med okali Mag) – (Mahony med Kali Mag) (cm)
44	9.67	-1.61	11.23
32	1.66	-5.50	8.04
69	16.97	16.67	0.13
27	3.61	-3.06	6.72
37	8.11	4.70	4.26
37	33.33	2.49	23.13
38	9.98	6.85	2.50
30	5.93	-3.54	10.25
30	1.57	-15.46	18.41
32	3.62	11.43	7.42

Skillnaderna i felprocent som visas i tabell 9 erbjuder en jämförelse i användningen av en okalibrerad och en kalibrerad magnetometer med Mahony-filter som belyser vikten på kalibreringen av magnetometern vilket är kritisk för att förbättra orienteringsuppskattningens noggrannhet. Dessa förbättringar i felprocent sågs när en kalibrerad magnetometer användes, vilket tyder att korrekt kalibrering kan ha en märkbar positiv effekt på systemets prestanda. Framgår av tabellen uppvisar den 37-årig deltagaren en signifikant förbättring i felmarginalen, med absolut skillnad på 23.13 cm. Detta indikerar en ännu mer betydande inverkan av kalibreringen av magnetometern på algoritmens felmarginal.

6 Diskussion

6.1 Samhällskrav

Idrottsträningen har genomgått en revolution tack vare modern teknik. Tidigare beroende av manuell anteckning och videoanalys har ersatts av realtidsdata från bärbara sensorer, GPS och mobiltelefoner. Dessa tekniker erbjuder omedelbar feedback, förbättrar rörelsebehärskning, underlättar kommunikation och bidrar till minskning av skador, vilket gynnar tränare, idrottare och fitnessentusiaster. Därmed möjliggörs ett effektivare sätt att förbättra idrottsprestanda samtidigt som tid och pengar sparas.

6.2 RMSE resultat för experimentet

Experimentet visade att den algoritm för positionsuppskattning som baseras på rörelseformeln inte uppnår den önskad precision när den enbart förlitar sig på IMU och magnetometer för avståndsmätning. Dessutom bidrog inte tillägget av en magnetometer märkbart till mätningarnas precision men orsakade svårigheter vid sensorjustering på grund av dess känslighet för externa magnetfält från källor som högtalare eller mobiltelefoner, vilket kan påverka noggrannheten negativt. Studien visade att användning av en kalibrerad magnetometer resulterade i en RMSE på 7.64 cm, jämfört med en okalibrerad magnetometer som uppvisade en RMSE på 10.02 cm för Mahony-algoritmen, vilket innebär en minskning med 2.38 cm som tyder på en viss förbättring med kalibrerad magnetometer. Att använda enbart data från gyroskopet och accelerometern för orienteringsuppskattning för att förenkla processen visade inte så mycket förbättringar, med en RMSE på 6.61 cm för Mahony och 6.69 cm för Madgwick. Ingen av metoderna uppfyller helt målet på RMSE ± 5 cm noggrannhet, men alla metoderna ligger inom en rimlig gräns för många tillämpningar.

6.3 Statistisk analys

Denna analys betonar vikten av noggrann validering och kalibrering av sensorer och algoritmer vid biomekaniska mätningar. Resultaten ger insikter i hur tekniska val påverkar datakvaliteten och, i förlängningen, noggrannheten i biomekaniska analyser.

6.3.1 Mahony vs Madgwick

Mahonys algoritm visade signifikant bättre prestanda än Madgwicks i orienteringsuppskattning enligt resultaten i tabell 2, med ett P-värde på 0.0041355 och ett negativt T-värde på -3.8128. Denna statistiskt signifikanta skillnad understryker vikten av algoritmval för noggrann rörelseanalys och dess potentiella påverkan på träningsoptimering och utveckling av bärbar teknologi.

6.3.2 Mahony utan Mag vs. Mahony med kali Mag

Inga statistiskt signifikanta skillnader noterades med ett P-värde på 0.80574, vilket indikerar att en kalibrerad magnetometer inte signifikant förbättrade Mahony-algorithmens prestanda i detta specifika fall. Detta pekar på att andra faktorer än magnetometerens kalibrering kan vara avgörande för orienteringsuppskattningens noggrannhet.

6.3.3 Mahony med okali Mag vs. Mahony med kali Mag

En statistiskt signifikant förbättring i prestanda observerades när magnetometer kalibrerades, med ett P-värde på 0.021972. Detta betonar kalibreringens betydelse för noggrannheten i orienteringsuppskattning med Mahony-filter.

Slutligen Mahony-algoritmen överträffar Madgwick i orienteringsuppskattning, varierar nyttan av att inkludera en kalibrerad magnetometer beroende på kontext och användningsfall. Vilket understryker vikten av ytterligare forskning för att optimera algoritmer och konfigurationer för specifika applikationer och system.

6.4 Jämförelse med ett relaterat projekt

Studien utvärderar effektiviteten av olika sensorfusion algoritmer i spårning av idrottares rörelse. Projektet syftade till att replikera och utvärdera algoritmernas förmåga att noggrant övervaka position, hastighet och acceleration under rörelser liknande de som ses i fältsporter. Experimenten använde både GPS- och IMU-data för att ge en grundlig bedömning av varje algoritms prestanda över olika rörelsemönster, inklusive gång, joggning, löpning och sprint [35]. För att bedöma hur väl ett filter fungerade användes RMSE som huvudindikator. Nedan följer RMSE-värden som mättes för att uppskatta position när filtret ställdes in på olika sätt som presenteras i tabell 10 nedan:

Tabell 10 (RMSE) på positionsuppskattningarna med råa och filtrerade mätningar visade att filtreringsåtgärden försämrar positionsuppskattningarna 4.1(Roobeek,2017s.53) [35].

filter configuration	RMSE [m]
raw measurements	1.03 (0.04)
Kraft + UKF	1.21 (0.32)
Madgwick + UKF	1.23 (0.29)
Kraft + KF	1.21 (0.32)
Madgwick + KF	1.22 (0.29)

I översikt visade filtreringsalgoritmerna, inklusive Kalmanfilter, generellt inte på en förbättring av precisionen i positionsuppskattningar jämfört med ofiltrerade GPS-data. Faktum är att filtreringsprocessen ofta resulterade i en lätt minskning av noggrannheten. Som konstruerades för detta projekt beror på detta främst på svårigheterna med att exakt uppskatta och justera fel i sensorernas orientering beräkningar, vilket avsevärt påverkar precisionen i position uppskattningarna [35].

6.5 Utmaningar med sensorjustering

Sensorn som används under experimentet kan inte ge ett exakt värde på hoppavståndet, utan endast ge ett ungefärligt värde. Detta beror på att sensorn saknar en referenspunkt. Därför har en tumstock även använts för att ge ett exakt värde som har jämförts med resultatet av sensorn.

Vid ett hopp kan även brus uppstå på grund av oregelbundna rörelser luften som uppkommer antingen vid hoppet eller vid landning. Processen av att justera low-pass filter under programmering påverkar precisionen av mätningen av det horisontella hoppavståndet. Därför blir noggrannheten i positionsberäkningarna påverkade.

7 Slutsats

Med endast en (IMU) och en magnetometer syftade detta projekt till att noggrant mäta avståndet för ett horisontellt hopp. Strategier utvecklades för att maximera noggrannheten, trots sensorernas begränsningar och känslighet för fel.

De största problemen var de skakiga avläsningarna och felen i rörelsesensorerna (IMU), samt magnetometer känslighet för omgivningen. Problemen tacklades genom att använda filtrerings- och kalibreringstekniker för minimera problemen. Detta möjliggjorde skapandet av ett enklare system som kunde användas i verkligheten, och gav en möjlighet att utforska hur teknikerna skulle kunna användas inom sport och för spårning av fysisk aktivitet.

I och med att enheten var tvungen att vara kopplad till en dator och till foten uppstod utmaningar för deltagarna. Det blev svårt för deltagarna att kontrollera hoppet i och med att de var kopplade till en sladd, vissa deltagare lyckades även dra ur kabeln ur datorn vid hoppet. För att undvika detta hade en trådlös sensor varit önskvärd.

Att lägga till en magnetometer för att öka noggrannheten tilläts ett extra kalibreringssteg för varje hopp, baserat på deltagarnas geografiska plats och det minskade noggrannhet. Det observerades även att externa enheter, exempelvis mobiltelefoner, inte bara påverkade magnetometern, utan även accelerometern genom att förändra den kombinerade dataanalysen.

Resultaten på RMSE på 10.02 cm för Mahony utan kalibrerad magnetometer och 7.64 cm med kalibrerad magnetometer. Å andra sidan visade mätningar med enbart accelerometer och gyroskop ett RMSE på 6.61 cm för Mahony filtret och 6.69 cm för Madgwick filtret. Detta visar på att mätsystemet inte uppfyller målet på $RMSE \pm 5$ cm noggrannhet, men att alla metoderna ligger inom en rimlig gräns. Det betyder att även om ingen metod är perfekt, kan de vara tillräckligt bra som ett resultat.

8 Litteraturförteckning

1. Innowearable. Inno-x [Internet]. 2021 [citerad 13 oktober 2023]. Tillgänglig från: <https://innowearable.se/innox/>
2. Aroganam G, Manivannan N, Harrison D. Review on Wearable Technology Sensors Used in Consumer Sport Applications. *Sensors*. 2019. [Åtkomstdatum: 15 november 2023]. Review on Wearable Technology Sensors
3. Carbonaro N, Tognetti A, redaktörer. *Wearable Technologies*. Basel: MDPI - Multidisciplinary Digital Publishing Institute; 2019 [citerad 15 november 2023].
hh.Wearable Technologies
4. Karlsson V. Maximal och uthållighetsstyrka hos kvinnliga handbollsspelare. Linnéuniversitetet. 2017 [citerad 15 november 2023]. Tillgänglig från: <https://lnu.diva-portal.org/smash/record.jsf?pid=diva2%3A1098112&dswid=-7721>
5. Raytelligence. Raytelligence acquires the sports electronics company Innowearable AB [Internet]. [citerad 28 november 2023]. Tillgänglig från: <https://www.raytelligence.com/en/raytelligence-acquires-the-sports-electronics-company-innowearable-ab/>
6. STMicroelectronics. LSM6DSOX-Datablad [Internet]. 2019 [citerad 1 oktober 2023]. Tillgänglig från: <https://www.st.com/resource/en/datasheet/lsm6dsox.pdf>
7. STMicroelectronics. LIS2MDL-Datablad [Internet]. 2018 [citerad 1 oktober 2023]. Tillgänglig från: <https://www.st.com/resource/en/datasheet/lis2mdl.pdf>
8. STMicroelectronics. Unico GUI [Internet]. [citerad 1 oktober 2023]. Tillgänglig från: https://www.st.com/en/development-tools/unico-gui.html#st_description_sec-nav-tab
9. STMicroelectronics. STEVAL-MKI109v3 utvecklingskort datasheet [Internet]. 2022 [citerad 1 oktober 2023]. Tillgänglig från: [user_manual/um2116-stevalmki109v3-professional-mems-tool-motherboard-for-mems-adapter-boards-stmicroelectronics](https://www.st.com/resource/en/user_manual/um2116-stevalmki109v3-professional-mems-tool-motherboard-for-mems-adapter-boards-stmicroelectronics)
10. Al-Fahoum AS, Abadir MS. Design of a Modified Madgwick Filter for Quaternion-Based Orientation Estimation Using AHRS. *Int J Comput Electr Eng*. 2018;10(3).
Madgwick modified-filter
11. Ludwig SA. Optimization of Control Parameter for Filter Algorithms for Attitude and Heading Reference Systems. Congress on Evolutionary Computation (CEC); 08-13 juli 2018; Rio de Janeiro, Brasilien. IEEE; 2018. [citerad 29 oktober 2023].
Tillgängligt från: Optimization of Control Parameter for Filter Algorithms (Mahony-Madgwick)
12. STMicroelectronics. STEVAL-MKI189V1 - LSM6DSM adapter board for a standard DIL24 socket. Tillgänglig från: <https://www.st.com/en/evaluation-tools/steval-mki189v1.html>. [citerad 1 oktober 2023].

13. STMicroelectronics. STEVAL-MKI137V1 - LIS3MDL adapter board for a standard DIL24 socket. Tillgänglig från: <https://www.st.com/en/evaluation-tools/steval-mki137v1.html>. [citerad 1 oktober 2023].
14. Guzmits M, Schnabl C, Eiwen M, Marton-Lindenthal S, Rohrer S, Lampalzer A. Distance Measurement using a Surface Micro-machined Accelerometer. [citerad 30 dec 2023]. Tillgänglig från: Distance-Measurement-using-a-Surface-Micro-machined-Accelerometer
15. Robert Mahony, Tarek Hamel, and Jean-Michel Pflimlin. Nonlinear Complementary Filters on the Special Orthogonal Group. IEEE Transactions on Automatic Control, Institute of Electrical and Electronics Engineers, 2008, 53 (5), [citerad 29 oktober 2023]. Tillgänglig från: Mahony
16. AHRS. Madgwick Orientation Filter — AHRS 0.3.1 documentation [Internet]. AHRS [citerad 30 oktober 2023]. Tillgänglig från: Madgwick
17. Tawil Y. Magnetometer Soft Iron and Hard Iron Calibration: Why and How. Atadiat. 2022 Dec 15 [citerad 2024 jan 17]. Tillgänglig från: Why & How-Magnetometer Soft Iron and Hard Iron Calibration
18. Goodrich R. Accelerometers: What They Are & How They Work. Live Science. October 2, 2013. Tillgänglig från: accelerometers
19. Ó Tuama D. What is VS Code? (Visual Studio Code) [Internet]. Dublin: Code Institute; 2021 Mar 10 [citerad 2023 okt 2]. Tillgänglig från: <https://codeinstitute.net/se/blog/what-is-vs-code/>
20. Laurell A, Karlsson E, Naqqar Y. GPS and IMU Sensor Fusion to Improve Velocity Accuracy: Independent Project Work in Electrical Engineering, 15 hp. [Internet]. Stockholm: KTH Royal Institute of Technology; 2022 Jun. Tillgänglig från: <https://www.diva-portal.org/smash/get/diva2:1682155/FULLTEXT01.pdf>
21. Sasiadek JZ. Sensor Fusion in Aerospace Applications [Internet]. Ottawa (ON): Carleton University. Tillgänglig från: Sensor Fusion
22. Analog Devices. FAQ: Hard & Soft Iron Correction for Magnetometer Measurements. [Internet]. Norwood, MA: Analog Devices; [citerad 2024 Jan 30]. Tillgänglig från: faq-hard-soft-iron-correction-for-magnetometer-measurements
23. Kenton W. Statistically Significant [Internet]. Investopedia. 2023 Sep 28 [cited 2024 Feb 25]. Tillgänglig från: statistically_significant.
24. Graf B. Quaternions and Dynamics. February 2007. Tillgänglig från: Quaternions And Dynamics.
25. Sebastian O.H. Madgwick, Andrew J.L. Harrison, Ravi Vaidyanathan. June 29 - July 1, 2011.]. Tillgänglig från: Estimation-of-IMU-and-MARG-orientation-using-a-gradient-descent-algorithm.pdf.

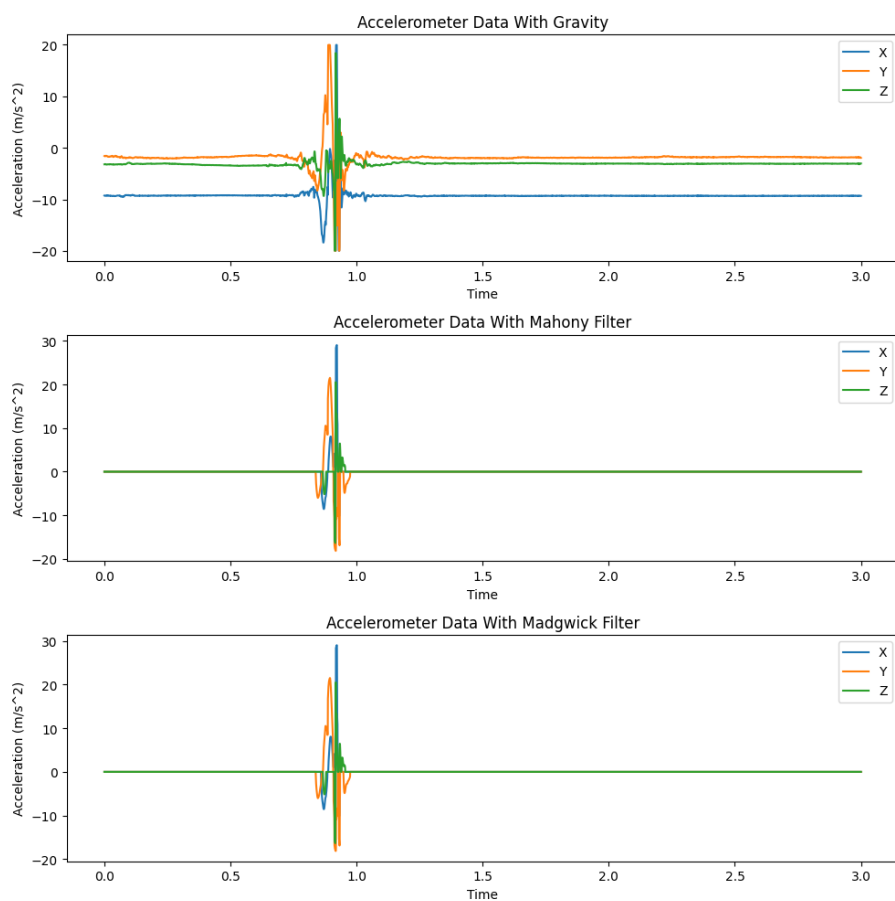
26. Understanding Gimbal Lock and how to prevent it. Xsens Knowledge Base. [Internet]. 2023 Mar 23, 10:52 AM. [citerad 2024 Feb 15].
Tillgänglig från: Understanding-Gimbal-Lock-and-how-to-prevent-it.
27. Garcia M. Attitude from gravity (Tilt) [Internet]. AHRS 0.3.1 documentation. [citerad Okt 12, 2024]. Available from: <https://ahrs.readthedocs.io/en/latest/filters/tilt.html>
28. Madgwick SOH. An efficient orientation filter for inertial and inertial/magnetic sensor arrays [Internet]. 2010 Apr 30. Tillgänglig från: madgwick_internal_report.pdf. [citerad 2024 Feb 15].
29. hi Y, Fang L, Guo D, Qi Z, Wang J, Che J. Research on distance measurement method based on micro-accelerometer. [cited 2024 Feb 14]. Tillgänglig från:
Research_on_distance_measurement_method_based_on_micro-accelerometer
30. Madgwick SOH. MahonyAHRS.c. 2011. [Internet]. Tillgänglig från: MahonyAHRSL83 [cited Okt 12, 2023].
31. SOH Madgwick. MadgwickAHRS.c. [Internet]. 2011. Tillgänglig från: MadgwickAHRS . [cited Okt 12, 2023].
32. Voler Systems. Motion Control: Accelerometers, Gyroscopes, and Magnetometers [Internet]. Voler Systems; [citerad 2024 Feb 15]. Tillgänglig från: motion-control-accelerometers-gyroscopes-magnetometers
33. Henriksson M. Estimation of heading using magnetometer and GPS [Internet]. 16 September 2013 [citerad 2024 Feb 15]. Tillgänglig från: <https://www.diva-portal.org/smash/get/diva2:648760/FULLTEXT01.pdf>
34. Gibbs K. Equations of motion [Internet]. 2016 [citerad 2024 Feb 15]. Tillgänglig från: Equations_of_motion
35. Roobeek M. Motion tracking in field sports using GPS and IMU [Internet]. University of Technology; 2023 [citerad 2024 Feb 15]. Tillgänglig från: Master of Since Theises Motion tracking in field sports.pdf
36. Spirine. Determining the three Euler angles from the acceleration [Internet]. Physics Stack Exchange. 2015 May 25 [citerad 2024 Feb 17]. Tillgänglig från: physics.stackexchange.com/questions/123456/determining-the-three-euler-angles-from-the-acceleration
37. The MathWorks, Inc. (1994-2024). ttest2. [Online]. Tillgänglig: Statistics- ttest. [citerad 18 mars 2024].
38. A guide to statistical tools in qualitative research [Internet]. Career development. 2023 Mar 30 [citerad 2024-03-24]. Tillgänglig: statistical-tools-in-qualitative-research.
39. FasterCapital. Analyzing Root Mean Squared Error (RMSE). [Internet]. Dubai: FasterCapital; [citerad 2024-03-24]. Tillgänglig: analyzing-root-mean-squared-error-(rmse)

9 Bilagor

9.1 Figurer



Figur 12 Utvecklingskort STEVAL-MKI109v3 & Modulen STEVAL- MKI217v1 som används under experimentet



Figur 13 Acceleration värde före och efter verkställda oreorienteringsfilter (Madgwick och Mahony)

9.2 Fördelar med en integrerad modul

Integrerade sensorer: Modulen integrerar både LSM6DSOX (IMU) och LIS2MDL (en högupplöst magnetometer). Detta innebär att modulen på egen hand erbjuder 3D-acceleration, 3D-rotationshastighet och 3D-magnetfältmätningar, och effektivt kombinerar kapabiliteter hos två separata moduler.

Kompakthet: Att ha båda sensorerna på en enda modul minskar den totala ytan för vår hårdvaruinstallation, vilket leder till en mer kompakt och effektiv design.

Enhetlig Datahantering: Med båda sensorerna på ett enda kort blir data-synkronisering och fusion inneboende enklare, vilket säkerställer sammanhängande och tidiga dataströmmar.

Ekonomiska överväganden: Genom att anta sensormodul som har två MEMS-enheter eliminerar behovet för av två separata utvecklingskort effektivt. Detta leder till kostnadsbesparingar inte bara när det gäller hårdvaruanskaffning, utan även när det gäller utvecklingstid och potentiella felsökningsinsatser.

Minskad Komplexitet: Att använda en enda modul minskar komplexiteten i mellananslutningar, potentiella signalstörningar och den övergripande systemintrikaten.

Effektivt Energiutnyttjande: Att hantera och reglera energi för en modul leder vanligtvis till bättre energieffektivitet jämfört med att hantera separata kraftledningar och regulatorer för två olika kort

Enkel Integrering: En enhetlig lösning innebär färre komponenter att integrera i det bredare systemet. Det accelererar prototypfasen och förenklar iterationer.

Konsekvent Miljöpåverkan: Med båda sensorerna på samma modul.

9.3 Avläsning av sensorvärden:

Första steget i processen är att utveckla ett skript i Python för att initiera kommunikationen med utvecklingskortet.

- **Upprättande av Anslutning:**
Anslut utvecklingskort till datorn med en USB-kabel. Kortet bör normalt registreras som en virtuell COM-port.
- **Initialisering:**
Öppna en seriell anslutning till kretskortet med rätt COM-port och en baudhastighet på 9600. Använd start- och stoppkommandon enligt datasheeten för utvecklingskortet, till exempel `"*setdb217v1"` och `"*start,"` för att inleda anslutningen. För att avsluta anslutningen, använd kommandot `"*stop"[9]`.
- **Sända Kommandon:**
För att initiera en specifik sensor (t.ex. sensorn på utvecklingskortet), sänd kommandot `*setdb217v1`. Detta kommando ställer in gränssnittet för kommunikation med sensormodellen. Om det behövs för att nollställa sensorutgångarna kan vi sända kommandot `*zoff [9]`.

- **Läsning av Data:**

Efter initialisering sänds specifika kommandon för att läsa från utvalda register på sensorn (t.ex., '*wAADD'). Ett citat från databladet, som beskriver hur kommandot används i koden, lyder: ”Detta kommando skriver innehållet i accelerometerregistren på demonstrationskortet. De hexadecimala stora bokstäverna AA och DD representerar adressen till registret och datan som ska skrivas, respektive. Till exempel skriver *w20C7 värdet 0xC7 till registret på adress 0x20” (Datablad, sida 21, underavsnitt 4.2.9) [9].

I detta projekt används kommandot (*w106C) för att initiera läsningen av accelerationsdata från IMU-sensorn och kommandot (*w116C) för att starta gyroskopläsningen av IMU-sensorn. Användning och konfiguration av sensorns registerbaserad på informationen på sida 56 i databladet för LSM6DSOX, och ytterligare detaljer finns på sida 57 i databladet för sensor. Dock 6C i hexadecimal vilket betyder att frekvensen för acceleration och gyroskop är 416 Hz (högprestandaläge) för att dra nytta av noggrannheten och få pålitliga realtidsvärden [6].

9.4 Teknisk jämförelse mellan olika IMU specifikationer

SENSORMODUL kan testas tillsammans och prestanda utvärderas och ger en mer heltäckande bild av hur enheterna interagerar. Modul fördelar följande:

LSM6DSOX som tillhör modulen är en avancerad IMU med en 3D-accelerometer och ett 3D-gyroskop. Den har funktioner som gör den värdefull i rörelseanalys.

Funktioner i LSM6DSOX:

- ✓ **Maskininlärningskärna:** Mönsterigenkänning som minskar behovet av efterbearbetning.
- ✓ **Snabb datainsamling:** Kan fånga snabba rörelser som hopp.
- ✓ **Brusreduktion:** Förbättrade filter och algoritmer.
- ✓ **Data-buffer:** Lagring ombord säkerställer att ingen data går förlorad under snabba rörelser.

Feature/Characteristic	LSM6DSOX(STEVAL-KI217v1)	LSM6DSM (STEVAL-MKI189V1)
Type (DoF)	6DoF (3D accelerometer + 3D gyroskop)	6DoF (3D accelerometer + 3D gyroskop)
Maskininlärningskärna	Finns(för grundläggande klassificeringar)	Saknas
Inbyggda funktioner	Lutning, fritt fall, uppvakning, 6D/4D-orientering	Stegräknare, lutningsdetektering, betydande rörelsedetektering
FIFO Buffer	Finns	Finns
Koppling	I2C, SPI, I3C	I2C, SPI
Temperaturesensor	Inbyggd	Inbyggd
Brusnivå	Mycket låg (upp till 3,3 mg för acc, upp till 0,005 dps för gyro)	låg
ODR (Utgångsdatafrekvens)	Upp till 6.66kHz for gyroskop, 3.33kHz för accelerometer	Upp till 3.33kHz för både gyro och accelerometer
Särskilda funktioner	Stegräknare, knackigenkänning, handledslutning	Kompensation för Hard/Soft iron för extern magnetisk sensor
Strömförbrukning	Energisparlägen finns	Energisparlägen finns
Primärt användningsområde	Wearables, konsumentelektronik	Wearables, konsumentelektronik, allmänna rörelsetillämpningar

9.5 Magnetometer specifikationer jämförelse

LIS2MDL som tillhör modulen är en högupplöst magnetisk sensor. Även om den främst är designad för applikationer som kompassning och magnetiskt responsiva funktioner, är det viktigt att utforska dess potential inom andra områden som sportanalys.

LIS2MDL Egenskaper:

- **Hög noggrannhet:** LIS2MDL ger precisa magnetiska mätningar.
- **Låg strömförbrukning:** Lämplig för bärbara enheter.
- **Kompakt:** Lätt att integrera i bärbara sportanalyslösningar.

Funktion/Specifikation	LIS2MDL(STEVAL-MKI217v1)	LIS3MDL STEVAL-MKI 137v1
Typ	3-axlig magnetometer	3-axlig magnetometer
Driftspänning	1.71 V till 3.6 V	1.9 V till 3.6 V
Känslighet	1,5 mG/LSB (Ultra-Högpresterande läge)	0.6 mG/LSB
Utgångsdatafrekvens (ODR)	10 Hz till 100 Hz	0.625 Hz till 1,000 Hz
Helt skalintervall	±50 Gauss	±4/±8/±12/±16 Gauss
Gränssnitt	I ² C, SPI	I ² C, SPI
Strömförbrukning	50 µA (typvärde vid 50 Hz)	108 µA (maxvärde vid 80 Hz)
Brusnivå	3 mG rms/√Hz (i Ultra-lågeffektläge)	1 mG rms/√Hz (i högupplöst läge)
Temperaturområde	-40°C till +85°C	-40°C till +85°C
Inbyggda funktioner	Temperaturkompensation, självttest	Temperatursensor, självttest
Avbrott	Avbrottsgenerator för magnetfält	Avbrottsgenerator för magnetfält
Driftlägen	Strömsparläge, enkel konvertering, kontinuerlig	Strömsparläge, enkel konvertering, kontinuerlig
Förpackning	LGA-12 (2x2x0.7 mm)	LGA-16 (3x3x1 mm)
Tillämpningar	E-kompass, positionsspårning	E-kompass, gestigenkänning, positionsspårning

9.6 Konfigurering av Hårdvaran:

Korrekt installation och anslutning av hårdvarukomponenter är avgörande för systemets övergripande funktionalitet och prestanda. Följande steg säkerställer att sensorn är korrekt ansluten och redo att kommunicera med datorn:

- a- Anslut IMU-sensorn och magnetometersensorn på ST utvecklingskort.
- b- Kontrollera att ström- och kommunikation är korrekt anslutna.
- c- Koppla utvecklingskort till datorn via USB-kabel.

9.7 Python algoritmen

A – SensorBoard Python Class

```
class SensorBoard:
    """
    This class handles communication and initialization with the sensor board.
    """
    def __init__(self, port='COM4', baud_rate=BAUD_RATE):
        self.serial_port = serial.Serial(port, baud_rate)
        self.initialize_board()

    def initialize_board(self):
        # sending commands to the evaluation board
        commands = ["*stop\r\n", "*zon\r\n", "*setdb217v1\r\n", "*zoff\r\n", "*w106C\r\n", "*w116C\r\n", "*start\r\n", "*debug\r\n"]
        for cmd in commands:
            self.serial_port.write(cmd.encode('utf-8'))
            time.sleep(0.2)

    def read_line(self):
        return self.serial_port.readline().decode('utf-8').strip()
```

B – Dataprocessor Python Class

```
class DataProcessor:
    """
    Processes sensor data using Mahony, Madgwick.
    """
    def __init__(self):
        self.mahony = None
        self.madgwick = None
        self.is_initialized = False
        self.dt = 1.0 / BAUD_RATE

        self.accel_fulldata_mahony = []
        self.accel_fulldata_madgwick = []

        self.linear_acc_mahony = [0.0, 0.0, 0.0]
        self.linear_acc_madgwick = [0.0, 0.0, 0.0]

        self.distance_mahony = 0.0
        self.distance_madgwick = 0.0

        self.finished_bool = False
        self.finished_cnt = 0
        self.end_bool = False

        self.q_mahony = None
        self.q_madgwick = None

    def initialize_filters(self, initial_accel):
        # execute just in the first execution loop
        q0 = acc2q(initial_accel)
        self.mahony = Mahony(q0=q0)
        self.madgwick = Madgwick(q0=q0)
        self.is_initialized = True
        self.q_mahony = q0
        self.q_madgwick = q0

    def process_data(self, accel, gyro, dt):
        #after getting the acceleration and gyro data from the module,
        #it make the update on the global orientation to subtract gravity vector
        if not self.is_initialized:
            raise Exception("Filters not initialized")

        # Process data using Mahony filter
        self.process_mahony(accel, gyro, dt)

        # Process data using Madgwick filter
        self.process_madgwick(accel, gyro, dt)
```

```

def process_data(self, accel, gyro, dt):
    # Process data using Madgwick filter
    self.process_madgwick(accel, gyro, dt)

def process_mahony(self, accel, gyro, dt):
    #update the global orientation q
    self.q_mahony = self.mahony.updateIMU(q=self.q_mahony, acc=accel, gyr=gyro)

    #compute gravity vector according to the updated q orientation
    gravity = self.compute_gravity_vector(self.q_mahony)

    #subtract the gravity from the current acceleration
    linear_acc = self.subtract_gravity(accel, gravity)

    #appling the low pass filter to get ride off the noise
    self.linear_acc_mahony = self.apply_low_pass_filter(self.linear_acc_mahony, linear_acc)

    #Accumulate the distance that calculated according to the dt time in each step
    self.distance_mahony += self.calculate_distance(self.linear_acc_mahony, dt)

    #save the current acceleration value for visualization purpose
    self.accel_fulldata_mahony.append(copy.deepcopy(self.linear_acc_mahony))

    #check if the acceleration is still zero for a 1000 reading loop (if the man is finish the jumping session)
    self.check_finish_step(self.linear_acc_mahony)

def process_madgwick(self, accel, gyro, dt):...

def end_episode(self, dt):
    """after finishing the jump, it detects the acceleration jumping data from the noise data and return the jump distance"""
    mahony_reads = self.separate_acceleration_data(self.accel_fulldata_mahony)
    madgwick_reads = self.separate_acceleration_data(self.accel_fulldata_madgwick)
    return self.cal_dis_from_reading_list(mahony_reads, dt), self.cal_dis_from_reading_list(madgwick_reads, dt)

def check_finish_step(self, linear_acc):
    for i in range(3):
        if linear_acc[i]>8.0:
            self.finished_bool = True

    if self.finished_bool and linear_acc == [0.0, 0.0, 0.0]:
        self.finished_cnt += 1
    else:
        self.finished_cnt = 0

    if self.finished_cnt > 1000:
        self.finished_cnt = 0
        self.end_bool = True

```

```

def separate_acceleration_data(self, data):
    separated_data = []
    current_list = []

    for point in data:
        if point == [0.0, 0.0, 0.0]:
            if current_list:
                separated_data.append(current_list)
                current_list = []
            else:
                current_list.append(point)

    # Add the last list if it's not empty
    if current_list:
        separated_data.append(current_list)

    return separated_data

def cal_dis_from_reading_list(self, outlist, dt):
    #return the maximum distance from multiple acceleration lists
    dis_MH_final = []
    for list in outlist:
        dis_MH = 0.0
        for linear_acc_mahony in list:
            xx = 0.5 * linear_acc_mahony[0] * dt**2 * 1000
            yy = 0.5 * linear_acc_mahony[1] * dt**2 * 1000
            zz = 0.5 * linear_acc_mahony[2] * dt**2 * 1000

            dis_MH += np.sqrt(xx**2 + yy**2 + zz**2)
        dis_MH_final.append(copy.deepcopy(dis_MH))
    return max(dis_MH_final)

def compute_gravity_vector(self, q):
    return [
        2 * (q[1] * q[3] - q[0] * q[2]) * 10,
        2 * (q[0] * q[1] + q[2] * q[3]) * 10,
        (q[0]**2 - q[1]**2 - q[2]**2 + q[3]**2) * 10
    ]

def subtract_gravity(self, accel, gravity):
    return [ai - gi for ai, gi in zip(accel, gravity)]

```

```

def subtract_gravity(self, accel, gravity):
    return [ai - gi for ai, gi in zip(accel, gravity)]

def apply_low_pass_filter(self, previous, current, threshold=10):
    for i in range(3):
        if abs(current[i])<2:
            current[i] = 0.0
        if abs(current[i] - previous[i]) > threshold:
            previous[i] = 0.1 * previous[i] + 0.90 * current[i]
        else:
            previous[i] = 0.2 * previous[i] + 0.8 * current[i]

        if abs(previous[i])<2: #Second step threshold
            previous[i] = 0.0

    return previous

def calculate_distance(self, linear_acc, dt):
    displacement = [0.5 * a * dt**2 * 1000 for a in linear_acc]
    return np.sqrt(sum(d**2 for d in displacement)) # Adjust unit conversion as needed

```

C – DataVisualizer Python Class

```
class DataVisualizer:
    """visualization of sensor data."""
    def __init__(self):
        pass

    @staticmethod
    def plot_sensor_data(time, sensor_data, title, x_label, y_label):
        plt.figure(figsize=(10, 4))
        plt.plot(time, [data[0] for data in sensor_data], label='X-axis')
        plt.plot(time, [data[1] for data in sensor_data], label='Y-axis')
        plt.plot(time, [data[2] for data in sensor_data], label='Z-axis')
        plt.title(title)
        plt.xlabel(x_label)
        plt.ylabel(y_label)
        plt.legend()
        plt.grid(True)
        plt.tight_layout()
        plt.show()

    @staticmethod
    def visualize_accelerometer_data(time, accel_data, FilterName):
        """
        Visualizes accelerometer data.
        time: Time data.
        accel_data: Accelerometer data.
        """
        DataVisualizer.plot_sensor_data(time, accel_data, f'{FilterName} Accelerometer Data', 'Time (s)', 'Acceleration (m/s^2)')

    @staticmethod
    def visualize_gyroscope_data(time, gyro_data):
        """
        Visualizes gyroscope data.
        """
        DataVisualizer.plot_sensor_data(time, gyro_data, 'Gyroscope Data', 'Time (s)', 'Angular Rate (rad/s)')
```

D– MainClass Python Class

```
class MainApp:
    """
    Flow of the application.
    """
    def __init__(self):
        self.board = SensorBoard()
        self.processor = DataProcessor()
        self.boolState = [False, False] # 0->accel 1->Gyro
        self.start_time = time.time()

        self.run()

    def run(self):
        previous_time = time.time()
        while True:
            try:
                #read data from module
                line = self.board.read_line()
                if "AX" in line:
                    self.boolState[0] = True
                    accel = self.extract_values(line)
                    accel = self.fix_accel(accel)
                    # print(accel)

                elif "GX" in line:
                    self.boolState[1] = True
                    gyro = self.extract_values(line)
                    gyro = self.fix_gyro(gyro)

                elif "MX" in line:
                    mag = self.extract_values(line)

            except:
                pass

            if not self.processor.is_initialized and self.boolState[0]:
                # execute in the first loop
                self.processor.initialize_filters(accel)

            if all(self.boolState) and self.processor.is_initialized:
                #true when new acceleration (and) gyro is available after initializing the board
                self.boolState = [False, False]

            current_time = time.time()
            dt = current_time - previous_time

            self.processor.process_data(accel, gyro, dt)
            print("Mahony:" +str(self.processor.distance_mahony)+ " - Madgwick: " +str(self.processor.distance_madgwick)+ " dt: " +str(dt))

            previous_time = time.time()

            if(self.processor.end_bool):
                #Final step
                final_dis_mahony, final_dis_madgwick = self.processor.end_episode(dt)
                print("Final_Mahony:" +str(final_dis_mahony)+ " - Final_Madgwick: " +str(final_dis_madgwick))

                #visualizing results
                time2 = np.linspace(0,time.time()-self.start_time,len(self.processor.accel_fulldata_mahony))
                DataVisualizer.visualize_accelerometer_data(time2, self.processor.accel_fulldata_mahony, "Mahony")
                DataVisualizer.visualize_accelerometer_data(time2, self.processor.accel_fulldata_madgwick, "Madgwick")

            return

    @staticmethod
    def extract_values(line):
        """Extracts values from a string."""
        parts = line.split()
        return float(parts[2]), int(parts[5]), int(parts[8])

    @staticmethod
    def degree_to_radian(deg):
        return deg * math.pi / 180

    @staticmethod
    def fix_accel(accel):
        '''Scale the acceleration data'''
        return [accel[0]/400, accel[1]/400, accel[2]/400]

    def fix_gyro(self, gyro):
        return [self.degree_to_radian(gyro[0]), self.degree_to_radian(gyro[1]), self.degree_to_radian(gyro[2])]
```

9.8 Matlab kod

```
1 - clear;
2 - clc;
3 - % Mata in data från olika algoritm konfigurationer som finns i tabellerna 1, 4 och 7 i resultaten.
4 - A = [106.67,113.87,49.13,94.29,135.14,100,87.98,121.81,109.70,98.44];
5 - B = [95.44,105.83,49.00,88.21,130.88,76.87,85.48,110.93,91.29,105.86];
6
7 - % Beräkna skillnaderna mellan mätningarna
8 - differences = A - B;
9
10 - % Utför ett parat t-test
11 - [h, p, ci, stats] = ttest(A,B);
12
13 - % Visa resultatet
14 - disp(['T-test resultat: hyp = ' num2str(h)])
15 - disp(['T-statistik: ' num2str(stats.tstat)]);
16 - disp(['P-värde: p = ' num2str(p)]);
17 - disp(['Medelvärdesdifferens: ' num2str(mean(differences))]);
```