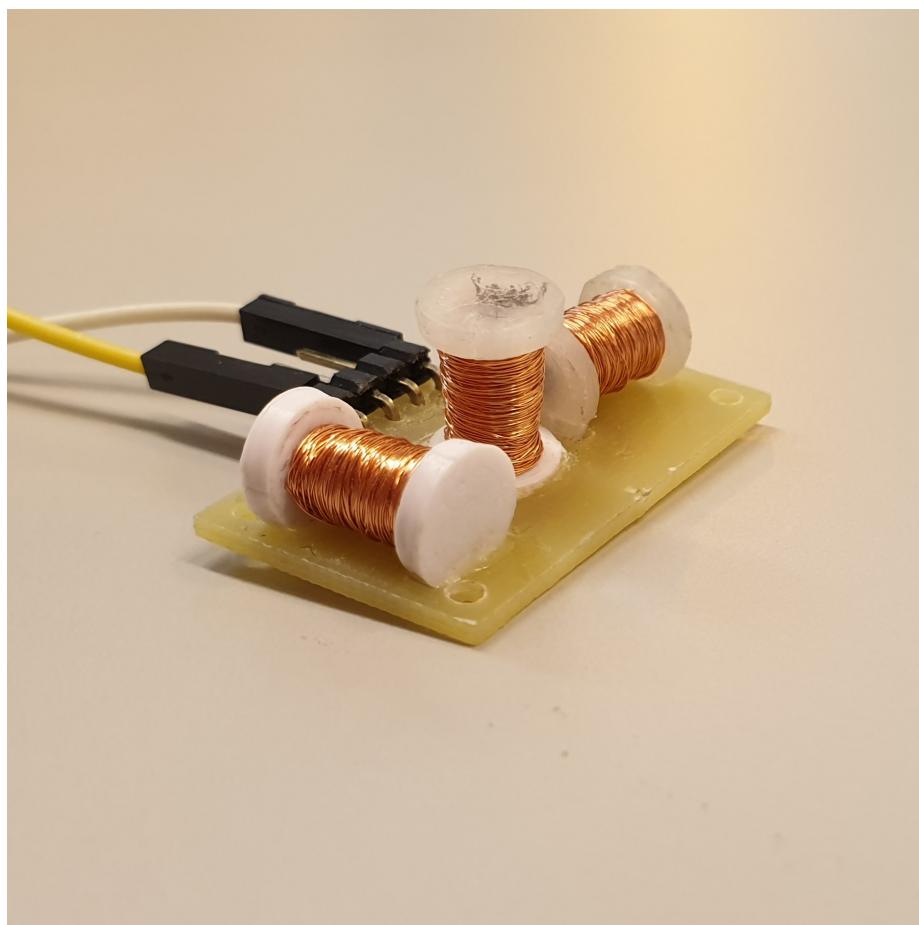


Gymnasiearbete Handroid

Spårning och representation av fingerrörelser.

Gabriel Calota
Jonathan Damsgaard Falck
William Johansson



Lärosäte: ABB Gymnasiet

Klass: 190S

Handledare: Andreas Jillram, ABB Gymnasiet

Sammanfattning

Det finns ett ökat intresse och en ökad efterfrågan på olika sätt för människan att interagera med datorer och robotar. Bland annat hur människokroppen kan användas för inmatning till olika datorprogram och maskiner. Den här tekniken är som mest utvecklad inom virtual reality och augmented reality eftersom det är inom dessa områden som tekniken i målget har störst användning. Tekniken är dock fortfarande relativt begränsad och använder mestadels grövre motorik som inmatning och förlorar den precision som kan ges av finmotoriska rörelser. Den här rapporten beskriver ett gymnasiearbete utfört av tre elever på ABB-Gymnasiet som har undersökt hur människokroppen kan användas som inmatningsverktyg till digitala system. Projektet har framförallt fokuserat på hur människans hand och fingerrörelser kan läsas av och representeras i en digital miljö. Ursprungligen syftade projektet att läsa av människans fingerrörelser för att sedan representera det med en fysisk robothand som skulle imitera de rörelser som människan utförde. Projektet bytte sedan inriktning och valde att representera fingerrörelserna digitalt i spelmotorn Unity istället. Tekniken som undersöktes för att läsa av handrörelser var egenbyggda sensorer som mätte av magnetfält.

Abstract

In recent time the interest and demand for different ways for the human to interact with computers and robots has increased. For example the possibility of using the human body as a tool to generate inputs for computer programs and different machines. This technology is most developed in virtual and augmented reality since it is within these sectors the technology today has most use. The technology is still relatively limited and uses mostly gross motor control as input and loses the precision that can be achieved with fine motor skill movements. This report describes a degree project executed by three student at ABB-Gymnasiet that has been investigating how the human body can be used as an input tool for digital systems. The project's main focus has been how the human hand and finger motions can be read and represented in a digital environment. Initially the projekt aimed at reading human finger motions and representing them with a mechanical model of a human hand that would imitate the finger motions carried out by the human using the sensor. The project then changed directions and decided to represent the finger motions digitally in the game engine Unity. The technology used to read the hand and finger movements was self-built sensors that utilised and measured magnetic fields.

Due to time limitations and other unexpected issues the measurement part of the project was not able to be completed to a point where a finger could be measured, but tests were made to see if the method is feasible. Tests to send and receive wireless AC signals from one inductor to an array of three inductors in different orientations showed that the relation between the signals of the three inductor receivers varied when moving and rotating the transmitting inductor.

Innehåll

1 Inledning	1
1.1 Syfte	1
1.2 Frågeställning	1
1.3 Bakgrund	1
1.3.1 Computer vision	1
1.3.2 Inertial measurement unit (IMU)	1
1.3.3 Flexsensor	1
1.3.4 Magnetfält	2
2 Teori	3
2.1 Elektromagnetism	3
2.1.1 Magnetfält	3
2.1.2 Induktion	3
2.2 Elektronik	3
2.2.1 Spolar	3
2.2.2 Mätning av spolar	3
2.2.3 Kondensatorer	4
2.2.4 Operationsförstärkare	4
2.2.5 Filter	4
2.2.6 Oscillatorer	5
3 Metod och material	6
3.1 Process 1 - Sensor	6
3.1.1 Val av metod	6
3.1.2 Spolar	6
3.1.3 Montering på hand	8
3.1.4 Filter	8
3.1.5 Oscillator	9
3.2 Process 2 - Representation	9
3.2.1 Fysisk hand	9
3.3 Arduino Nano 33 BLE	11
3.4 Bluetooth Low Energy	12
3.5 Unity	12
3.6 Digital visualisering	12
4 Resultat	13
4.1 Oscillator	13
4.1.1 Spolar	13
4.1.2 Oscillatorkrets	13
4.2 Filterkrets	15
5 Diskussion och Slutsats	16
5.1 Spårning	16
5.1.1 Oscillator	16
5.1.2 Filter	17
5.2 Visualisering	17
5.3 Planering	17

5.3.1	Undersökning och val av metod	17
5.3.2	Dimensionering av kretsar	18
5.3.3	Tid	18
5.3.4	Framtida undersökningar	18
5.4	Slutsats	19
6	Källor	20
	Bilagor	22
	A Kod Spolsnurrare	22
	B Filterkrets	24
	C Oscillatorkretsar	25
	D Servokrets	26
	E Kod Arduino Peripheral	27
	F Kod Rendering	29

1 Inledning

I denna del följer en beskrivning av projektets mål och syfte, samt en bakgrund för att se hur syftet uppnås med olika metoder.

1.1 Syfte

Syftet med projektet är att undersöka hur människan finmotoriska rörelser kan detekteras, hur rörelserna såsom fingerrörelse kan användas som inmatning till olika system och utveckla en prototyp som kan både detektera och återskapa fingerrörelser från en människohand. De finmotoriska rörelser kan återskapas antingen med en digital representation av rörelsen eller med en fysisk robothand.

1.2 Frågeställning

Frågeställningen i projekt Handroid är om det går att mäta finmotoriska kroppsrörelser såsom fingerrörelser med magnetfält, och i så fall hur stor precision metoden har.

1.3 Bakgrund

För att få bättre uppfattning av hur fingerspårningstekniken ser ut idag beskrivs här olika fingerspärningsmetoder som undersöktes under projektets planering.

1.3.1 Computer vision

Computer vision är en idag vanlig metod för att spåra hand och fingerrörelser. Med hjälp av en kamera filmas handen in och filmen kan sedan användas som indata till ett fingerspärningsprogram. För att ta fram fingerrörelserna från filmen används olika computer vision-verktyg tillsammans med maskininlärning. Metoden är välbeprövad, men kräver att händerna är inom synhåll för kameran och i rätt vinkel för att fungera. Det här innebär att lösningen endast fungerar i kontrollerade miljöer. Ett exempel som använder sig utav computer vision är projektet *On-Device, Real-Time Hand Tracking with MediaPipe* av Google Research [1].

1.3.2 Inertial measurement unit (IMU)

Inertial measurement unit är en slags sensor som mäter rörelse. Genom att placera sensorer på de olika fingersegmenten på handen kan rörelser i tre dimensioner spåras. Precis spårning kräver minst 16 sensorer placerade på handen. Det finns även lösningar med IMU-sensorer som använder 7 sensorer där där övriga positioner antas utifrån de positioner som läses in.[2]

1.3.3 Flexsensor

En flexsensor är en sensor som ändrar resistans proportionerligt med hur mycket den är böjd [3]. Flexsensorer kan användas för att mäta hur mycket ett finger är böjt och de har använts i projekt och produkter för att spåra böjning, till

exempel genom att integrera flexsenorer i en handske för att styra olika applikationer. Ett exempel är datahandsken [4], som används bland annat för att känna igen teckenspråk.

1.3.4 Magnetfält

Magnetfält kan användas för att mäta rotation och avstånd av fingrar och händer. Denna metod är något ovanligare inom industrin, men många exempel kan hittas som examensprojekt på olika högskolor i flera länder. Många av dessa projekt liknar detta gymnasiearbete, med liknande syfte och grundläggande metod, men med lite annorlunda tillvägagångssätt. Ett projekt använder sig av maskininlärning för att bättre kunna förutspå fingerposition [5]. Projektet *AuraRing* [6] använder sig av en ring runt fingret som skickar ut magnetfält, som sedan kan plockas upp av fem spolar på handleden för att få en bättre bild av fingrets position. De använder sig av både maskininlärning och en mer traditionell iterativ metod för att uppnå sin mätning. Ett annat projekt, *MagIK* [7], använder spolar på fingrarna samt en rad spolar på handryggen för att kunna räkna ut position utan AI, men använder sig av kamera för att verifiera den uppmätta positionen. Detta är alltså något som inte är vanligt inom industrin, och en metod som kan tillämpas och utforskas på olika sätt.

2 Teori

För att få en bättre bild av hur en robothand kan skapas och styras med hjälp av olika sensorer undersöktes andra, liknande projekt. Projektet drog inspiration från en kandidatexamen från två KTH-studenter [8] med ett liknande syfte, samt idéer för hur fingrar kan roteras och hur en tumme kan kontrolleras på en fysiska hand kom från ett projekt som hette *Etho Hand* [9] vars syfte var att utveckla en hand som kunde utföra komplexa rörelser.

2.1 Elektromagnetism

Eftersom detta projekt utnyttjar magnetfält och elektromagnetiska egenskaper förklaras här några koncept som bör känna till för att förstå hur spårningen med hjälp av magnetfält fungerar.

2.1.1 Magnetfält

Magnetfält är fält som rör sig i en riktning från nord- till sydpolen på en magnet eller runt en strömförande elektrisk ledare. Den magnetiska flödestätheten, ekvivalent med magnetfältets *styrka*, kring en ledare med ström, avtar med avstånd från ledaren. [10]

2.1.2 Induktion

Inducerad spänning är spänning som alstras när en ledare befinner sig i ett magnetfält, på grund av att de laddade partiklarna hamnar på motsatta sidor av ledaren. Det skapar en skillnad i laddning, vilket korrelerar med spänning. Enligt Lenz lag ger denna spänning upphov till en ström, i en sluten krets, och att strömmens riktning motverkar förändringen av det magnetiska flödet. Denna princip kan användas för att alstra ström ur ett magnetfält, exempelvis i en generator. Induktion fungerar också åt motsatt håll: ström och spänning i en krets ger upphov till magnetfält runt ledaren. [10]

2.2 Elektronik

Här följer en beskrivning av de viktigaste elektriska komponenterna och kretarna som används i det här projektet.

2.2.1 Spolar

En spole är en ledare som lindats i varv på ett sådant sätt att magnetfältet som skapas när ström flödar igenom har en nord- och en sydpol. Spolar kan lindas runt ett material, en *kärna* som förstärker magnetfältet, eller bara med luft i mitten. På grund av att spolen inducerar spänning kommer den motverka strömmen i kretsen den ingår, enligt Lenz lag, vilket bland annat ger en förskjutning mellan spänning och ström, i en växelströmskrets. [10]

2.2.2 Mätning av spolar

För att dimensionera en krets kan det vara viktigt att veta vilka värden ens komponenter har. Till resistorer och kondensatorer är detta oftast enkelt, då de

flesta moderna multimeterna kan mäta resistans och kapacitans. Verktyg för att mäta spolars induktans (som mäts i enheten 1 Henry) är sällsyntare, och därför finns olika metoder för att själv mäta spolar utan specialverktyg. För detta kan exempelvis en funktionsgenerator och ett oscilloskop användas, och genom att utnyttja spolens upp- och urladdningsförmåga kan man approximera spolens induktans. En sinusformad spänning med amplituden 0,5 V utan något annat inkopplat kommer minska när en spole kopplas parallellt med mätverktyget. Genom att ändra frekvensen kan man uppnå signalen med 0,5 V amplitud. Denna frekvens f tillsammans med resistansen R , som mäts med multimeter när ingen spänning ligger över den, kan användas för att räkna ut induktansen L enligt formel 1. [11]

$$L = \frac{\sqrt{\frac{2500+100R-3R^2}{3(2\pi)^2}}}{f} \quad (1)$$

2.2.3 Kondensatorer

Kondensatorn är en elektrisk komponent som kan lagra elektrisk energi, genom att laddas upp och laddas ur med spänning, vilket ger upphov till en förskjutning mellan ström och spänning, i en växelströmskrets. [10]

2.2.4 Operationsförstärkare

En operationsförstärkare är en förstärkare som förstärker spänningsskillnaden mellan operationsförstärkarens ingångar enligt formel 2 där A_{OL} är förstärkningen i en öppen loop. [12]

$$V_{out} = A_{OL}(V_+ - V_-) \quad (2)$$

2.2.5 Filter

Ett elektroniskt filter används för att filtrera bort oönskade frekvenser från signaler. [13] En typ av elektroniskt filter är aktiva filter, där aktiva komponenter som operationsförstärkare används för att förbättra filtrets prestanda och förutsägbarhet, samt för att sänka priset. Operationsförstärkaren förhindrar också komponenter efter filtret från att påverka filtrets egenskaper. [14] För att filtrera alla signaler förutom signaler med en frekvens inom ett visst intervall kan ett band-pass filter användas. Bandpassfilter har en bandbredd som är skillnaden mellan den övre frekvensen och lägre frekvensen som motsvarar en förstärkning på -3 dB. [15]

$$Q = \frac{f_c}{BW} \quad (3)$$

$$R_1 = \frac{Q}{2\pi f_c C A_f} \quad (4)$$

$$R_2 = \frac{Q}{2\pi f_c C (2Q^2 - A_f)} \quad (5)$$

$$R_3 = \frac{Q}{\pi f_c C} \quad (6)$$

$$A_f = \frac{R_3}{2R_1} \quad (7)$$

$$A_f < 2Q^2 \quad (8)$$

$$R'_2 = R_2 \left(\frac{f_c}{f'_c} \right)^2 \quad (9)$$

Ett bandpassfilter kan dimensioneras genom att välja en centerfrekvens f_c , en bandbredd BW och ett värde på två kondensatorer C . Tre resistorer kan sedan dimensioneras med formel 4, 5 och 6 där formel 3 bestämmer filtrets kvalitetsfaktor. Amplifikationen vid centerfrekvensen kan räknas ut med formel 7 som dessutom måste uppfylla villkoret 8. För att välja nya centerfrekvenser kan formel 9 användas. [16]

2.2.6 Oscillatörer

En oscillator är en elektrisk krets eller en del av en elektrisk krets där växelström, ström som regelbundet byter riktning, skapas. På grund av spolars och kondensatorers upp- och urladdningsförmåga kan dessa användas, i olika kombinationer för att göra en krets där strömmen och spänningen svänger (blir svagare/starkare, och byter riktning) med en viss frekvens. Oscillatörer är användbara om man vill skapa växelström med en likströmskälla.

En sorts oscillator är en Wien-bryggeoscillator (*Wien Bridge Oscillator*) [17] där resistorer, kondensatorer och en operationsförstärkare används för att skapa en vågformad växelströmssignal, där frekvensen bestäms av resistorernas och kondensatorernas värden. En annan sorts oscillator är en 555-oscillator [18], som bygger på ett chip med en integrerad krets (*IC*) vid namn 555. 555-chippet kan ge fyrkantsformade utsignaler om de sätts i ett visst värde, och där frekvensen bestäms med hjälp av restistorer och kondensatorer. Det går också att filtrera signalen så att den liknar en sinusvåg, med hjälp av en spole och kondensatorer.

3 Metod och material

Projektet började med att undersöka hur spårning av handrörelser har gjorts tidigare. Där följde undersökningar i hur en människohand fungerar och hur en fysisk hand kan konstrueras. Projektet delades sedan in i två processer, en process för att skapa ett sensorsystem som kunde användas för att spåra fingerrörelser, och en annan process för att skapa en representation av den data som mäts in av sensorsystemet, som arbetades på parallellt.

3.1 Process 1 - Sensor

De egenbyggda sensorerna bestod av spolar, oscillatorkretsar, och filter, som skräddarsyddes för varje finger.

3.1.1 Val av metod

Metoden som valdes var att spåra fingrarna med hjälp av spolar på fingertopparna som sänder ut ett magnetfält som genom att läsas av skulle kunna ge fingrarnas rotation och position. Valet av metod stod mellan tre huvudkandidater. Avläsning av magnetfält från spolar på fingertopparna, flexsensorer för att ta fram fingrarnas rotation, eller flertalet IMU-sensorer som genom att mäta rörelserna på olika delar av handen kan ge data på hur hela handen rör på sig.

Metoden med accelerometrar valdes bort tidigt eftersom det var en väldigt avancerad och dyr metod som skulle kräva minst 7 stycken sensorer för att fungera och minst 16 stycken sensorer för att kunna spåra hela handen. Då metoden krävde mycket för att komma igång bedömdes den icke-lämplig för projektet.

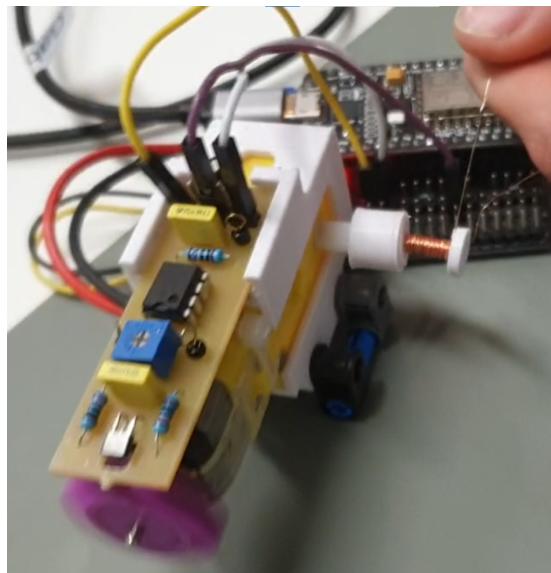
Metoden med flexsensorer var den metod som bedömdes som lättast att genomföra. Den valdes bort på grund av att flexsensorer ansågs för begränsande då flexsensorerna endast fanns i specifika storlekar som gjorde att vissa positioner och rotationer skulle vara svåra att mäta. Flex-sensorerna som sensorer var även väldigt dyra och metoden bedömdes vara aningen enkel och hade redan gjorts många gånger tidigare. Dessutom kan en flexsensor endast mäta rörelse i en riktning vilket innebar att precis mätning av alla fingrars rörelser skulle kräva ett stort antal sensorer.

Slutligen valdes magnetfält eftersom det verkade som en lovande metod. Magnetfält används till största delen för att spåra grövre rörelser men bedömdes ha potential att mäta även finare rörelser. Metoden var även den som var minst beprövad vilket innebar att gruppen skulle behöva ägna sig åt mera testning och utforskning.

3.1.2 Spolar

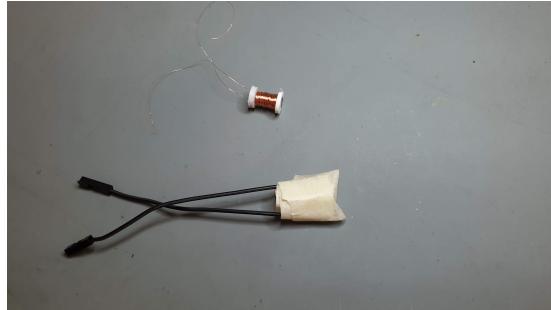
Spolar användes för att skapa och läsa av magnetfältet som används för att läsa av fingrets position och rotation. En spole består av en järnkärna, 0,1 mm koppartråd samt 3D-printade hattar. För att få lämplig storlek och funktion från spolarna valdes ett varvtal på 1000 varv. Spolarna konstruerades genom

att limma fast hattar i ändarna på järnkärnan. Sedan lindades spolarna med koppartråd. Lindningen gjordes genom att fästa den olindade spolen i en motor som genom att snurra samtidigt som koppartråd var fäst i den olindade spolen gjorde att järnkärnan lindades med koppartråden. Motorn var kopplad till en Arduino mikrokontroller som med hjälp av en varvmätare kunde läsa av hur många varv koppartråd som hade lindats runt spolen. När mikrokontrollern räknat att motorn roterat 1000 varv stängdes motorn automatiskt av. Se figur 1. Kod för spolsnurraren finns i bilaga A



Figur 1: Bild från när en spole lindades med hjälp av Spolsnurraren.

De färdiga spolarna mättes med oscilloskop och funktionsgenerator för att säkerställa dess induktans, så att den kan användas i oscillatorkretsen med rätt komponenter. Den första spolen som färdigställdes jämfördes också med spolar som används i utbildningssyfte inom fysik, för att säkerställa att de hade samma funktion. Detta genom att använda två fysik-spolar för att skicka och ta emot signaler, där en spole var kopplad i serie med en funktionsgenerator som genererade en sinusformad signal, och där den andra var kopplad till ett oscilloskop för att läsas av. En av fysik-spolarna byttes sedan ut mot den egensnurrade spolen, för att se om signalen kunde skickas och tas emot av dessa. För att få bättre kontakt med andra kablar och komponenter valdes ett antal spolar ut för att lödas fast i kablar av större tjocklek. Jämförelsen mellan en spole med fastlödd kabel och en spole utan fastlödd kabel kan ses i figur 2.



Figur 2: En spole med endast hatt, kärna och lindning (överst) och en spole med kablar fastlödda (underst).

3.1.3 Montering på hand

För att montera spolar på handen monterades de tre mottagarspolarna på ett kretskort som agerade spolhubb. De tre spolarna monterades i varsin riktning så att de kan ta upp fingerrörelser i tre dimensioner. Kretskortet modellerades i Ultiboard och frästes sedan ut på skolan kretskortsfräs. På kretskortet monterades spolarna genom att limmas fast i kretskorten och kopparträden löddes fast på kortets undersida. För att kunna skicka vidare signalen som spolarna plockar upp löddes även en stiftlist fast på kretskortet.

3.1.4 Filter

För att särskilja på olika fingrar skulle varje finger ha en egen frekvens med ett bestämt avstånd från andra fingrar, för att göra det möjligt att filtrera bort alla förutom ett finger. Filtret ska filtrera frekvenser som inte är inom ett visst frekvensområde och ett aktivt bandpassfilter valdes som filtertyp. Filterkretsen designades genom att räkna ut värden på resistorer genom att välja en bandbredd, värde på kondensatorerna och en centerfrekvens. Efter rådgivning med elektronikläraren Daniel Åkerlund bestämdes att varje finger skulle ha ett avstånd på cirka 15 kHz till nästa finger och att det första fingrets centerfrekvens skulle vara 100 kHz. Se bilaga B för kretsdesign.

Tabell 1: Värden på resistorer som räknats ut.

f_c (kHz)	R_1 ($k\Omega$)	R_2 (Ω)	R_3 ($k\Omega$)
100	49.7	11.0	198.9
115	49.7	8.0	198.9
133	49.7	5.7	198.9
147	49.7	4.3	198.9
162	49.7	3.3	198.9

Kretsen testades sedan i simulering för att säkerställa att kretsens fungerar som förväntat, dock användes en ideal operationsförstärkare under simuleringen. När kretsen sedan simulerades med icke-ideal operationsförstärkare kunde kretsen inte längre filtrera signalen. Det visade sig att dimensioneringen av resistorerna krävde en ideal, eller i alla fall näst intill ideal, operationsförstärkare

för att kretsen skulle fungera. Kretsen designades sedan om med hjälp av ett filterdesignverktyg av Texas Instruments eftersom verktyget också räknade med icke-ideala operationsförstärkare. Avståndet mellan varje finger ökades också.

Tabell 2: Värden på resistorer som räknats ut med hjälp av Texas Instruments.

f_c (kHz)	R_1 ($k\Omega$)	R_2 (Ω)	R_3 ($k\Omega$)
100	270	316	806
117	220	220	806
141	180	150	806
170	150	100	806
203	120	68	806

3.1.5 Oscillator

För de olika fingrarna bestämdes att oscillatorerna skulle ha frekvenser från 100 kHz, med ungefär 15 kHz mellanrum för att ge utrymme för filtrernas bandbredd. Fem oscillatorer behövdes, ett för varje finger. Ursprungligen planerades att använda oscillatorer som bygger på wienbryggor (Wienbryggessoncillator). Till detta skulle behövas resistorer, kondensatorer och operationsförstärkare. Resistorer och kondensatorer dimensionerades och kretsen simulerades i program för att kontrollera att allt stämde. Efter förslag om att integrera spolarna i kretsen som en resonanskälla byttes oscillatorna mot 555-oscillatorer. Här dimensionerades istället två olika kretsar i en: en med resistorer och kondensatorer, och en med kondensatorer och spolen.

Två oscillatorer byggdes i verkligheten: en med frekvensen 100 kHz och en med frekvensen 117 kHz. I 117 kilohertz-kretsen användes ett 556-chip, som har samma funktioner som två stycken 555:or; en på varje sida av komponenten, och som drar mindre effekt. Se bilaga C för kretsschema.

3.2 Process 2 - Representation

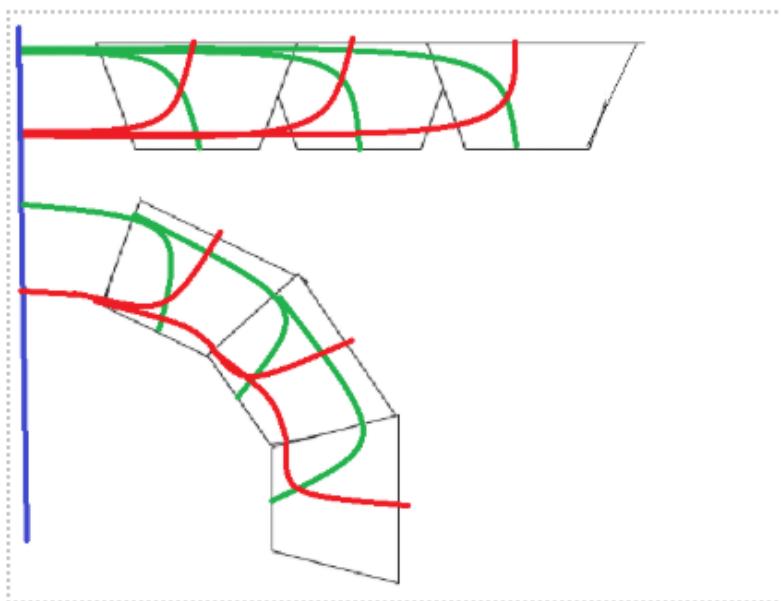
För att verifiera att sensorsystemet fungerar och för att kunna visa att systemet går att använda till någon form av applikation. Målet var att visualisera sensordata på ett sätt som gör det lätt att se om systemet fungerar och för att sedan kunna utvärdera hur väl sensorerna har lyckats mäta fingerrörelser.

3.2.1 Fysisk hand

Ursprungligen beslöts att en fysisk hand skulle designas för att visualisera data från sensorerna. Den starkaste motiveringen för valet att använda en fysisk hand var att en fysisk hand fungerar väldigt bra som ett utställningsobjekt och visar att fingerspårning inte enbart kan användas i digitala applikationer utan också användas för att kontrollera något fysiskt. När detta beslut togs var projektet dessutom inriktat på att använda flexsensorer, som tidigare nämnts valdes bort, vilket på grund av den begränsade möjligheten att en flexsensor endast kan mäta en rotation skulle innebära att endast ett fåtal rotationer

mätes. Konstruktionen skulle på grund av detta bli betydligt enklare än att perfekt rekonstruerar en hand eftersom endast ett fåtal rotationer skulle mäts kunde konstruktionen förenklas genom att enbart tillåta likformig rotation av varje finger i endast en riktning.

Designen bestod av fingersegment som kunde skrivas ut med en 3D-skrivare och 7 servon som med hjälp av trådar skulle böja fingrarna. Se bilaga D för kretsschema. Till varje finger används två trådar. Trådarna dras i kanaler på motsatta sida som de fäst. När snörena sedan sätts i spänn kommer de att förhindra rotation åt varsitt håll och genom att förlänga en snöret och förkorta andra snöret lika mycket kommer fingrets rotation förändras. Se figur 3. Tummens böjning skulle på ett liknande sätt kontrolleras men skulle också med hjälp av ett kullager och två par snören kontrollera rotation kring tummens fäste.



Figur 3: Skiss av flexion och extension av finger med hjälp av två snören.

Själva handen designades i segment liknande hur människohanden ser ut och skulle sedan monteras ihop med stänger i lederna. Se figur 4.



Figur 4: Bild på utskrivet finger (ringfinger) bestående av tre segment och tre böjleder.

Att bygga en fysisk hand valdes sedan bort, i och med att en annan spårningsmetod valdes, delvist eftersom konstruktionen blir mer komplicerad när fler riktningar skulle mäts men framförallt eftersom sensor metod inte är lika beprövad som flexsensorer och att eliminera möjlig felkällor med den fysiska handen och istället skapa en digital hand underlättar testning av sensorsystemet.

3.3 Arduino Nano 33 BLE

Mikrokontrollen Arduino Nano 33 BLE används för att kommunicera med renderingsprogrammet. Arduino Nano 33 BLE är en mikrokontroller baserad på kretsen Nordic nRF52480 producerad av företaget Arduino. Arduino Nano 33 BLE är en fysiskt liten mikrokontroller. Nordic nRF52480 är bestyckad med en Cortex M4F processor och en NINA B306 BLE-modul som möjliggör kommunikation med Bluetooth Low Energy. Arduino Nano 33 BLE har 8 analoga ingångar och kan drivas av spänning mellan 4,5 och 21 volt. Annan viktig hårdvaruinformation om mikrokontrollern kan ses i tabell 3. [19]

Mikrokontrollern används för att läsa av mätvärden från sensorerna och för att skicka vidare mätvärdena till datorn där renderingen sker. Den programmeras i C++ med hjälp av platformIO, en tillbyggnad i utvecklingsmiljön Visual Studio Code som möjliggör programmering av mikrokontrollers i C++.

Tabell 3: Viktiga specifikationer Arduino Nano 33 BLE [19]

Mikrokontroller	nRF52840
Driftspänning	3,3V
Inspänning (min)	4,5V
Inspänning (max)	21V
Digitala ingångar/utgångar	14 st
Analoga ingångar	8 st
Längd	45 mm
Bredd	18 mm

Mikrokontrollen använder bluetooth för att skicka iväg data, först väntar mikrokontrollen på att renderingsprogrammet ska ansluta via Bluetooth Low Energy och läser sedan av värdena från de olika sensorerna och publicerar sensorvärdena. För att åstadkomma kommunikation används biblioteket ArduinoBLE. Kod för mikrokontrollens BLE kommunikation finns i bilaga E.

3.4 Bluetooth Low Energy

Bluetooth är ett protokoll för att trådlöst skicka data mellan olika enheter. Bluetooth Low Energy liknar traditionell Bluetooth men är mer energieffektiv. [20]

3.5 Unity

Unity är en programvara som möjliggör skapandet av real-time 3d-miljöer för spel, film etc. Programmet används framförallt för att skapa spel och kan skapa 2D, 3D, VR och AR spel. [21]

3.6 Digital visualisering

Ursprungligen skulle ett C# program skapas som kunde kommunicera med mikrokontrollen som sedan också skulle rendera fingerrörelserna. Istället för att skapa ett eget renderingsprogram beslutades det att spelmotorn Unity skulle användas. Kommunikationen mellan mikrokontrollen och renderingsprogrammet sköttes därav istället av biblioteket BleWinrtDll som är går att direkt integrera med Unity och en färdig konstruerad modell från Ultraleaps Unity Plugin användes för att representera fingerrörelse. Unity-programmet skulle anslutas till mikrokontrollen som sedan skulle översätta värdena från sensorerna till rotationer i fingrarnas leder. Programmet testades genom att skicka två specifika värden från mikrokontrollen till Unity-programmet som roterade ett fingers leder till två olika rotationer beroende på vilket värde som skickats. Kod för test av BLE kommunikation finns i bilaga F.

4 Resultat

Resultaten av de olika processerna som beskrivits redovisas här objektivt, och diskussion av resultatet följer i nästa del.

4.1 Oscillator

En del av sensorerna bestod av oscillatorerna, som skulle skapa och skicka en signal genom en spole.

4.1.1 Spolar

De egentillverkade spolarna hade samma funktion som andra fabrikstillverkade spolar, enligt den jämförelse som gjordes. Signalen var något förändrad vad gäller amplitud, men hela signalen kunde tas emot, och signalens amplitud växte och krympte beroende på avstånd från sändar-spolen. Störningar från omgivningen var dock något större, enligt resultaten som kunde ses i oscilloskop.

Första mätningen av de första tre spolarnas induktans gav en genomsnittlig induktans på 3,3 mH vilket användes vid dimensionering av oscillatorer. Vid en andra omgång mätningar mättes fem andra spolar, där den genomsnittliga induktansen var runt 7,0 mH, vilket ledde till att oscillatorkretsarna dimensionerades om. Se tabell 4.

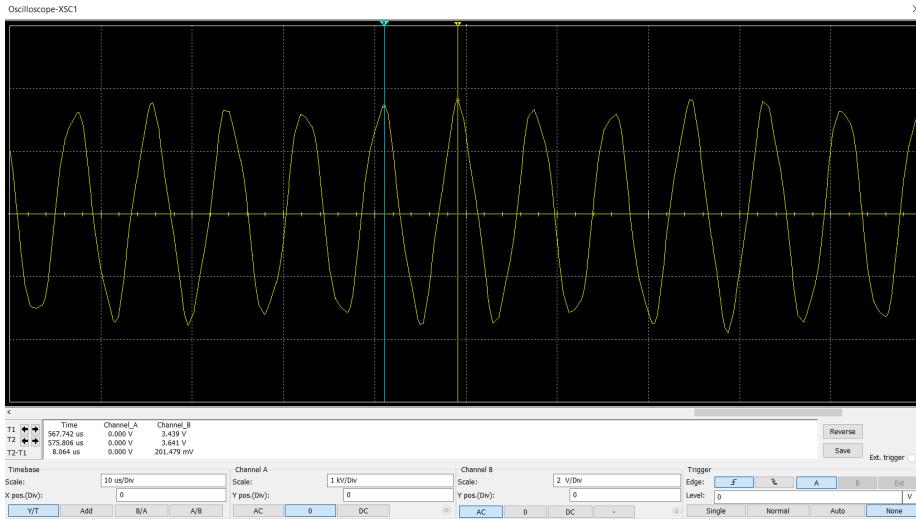
Tabell 4: Uppmätta induktanser i egensurrade spolar. Värdena från första mätningarna är runt 3,3 mH och de från andra mätningen är runt 7,0 mH.

Spolnummer	Induktans (mH)
1	3,3
2	3,3
3	3,53
4	6,86
5	7,03
6	7,12
7	7,03
8	7,23

Under testning gick vissa spolar sönder, och lödningen släppte på två av fyra lödda spolar under laborationer och tester.

4.1.2 Oscillatorkrets

I simuleringar gav oscillatorkretsarna signaler med amplituder (maxvärdet) från 3 V till 5 V. Se figur 5.

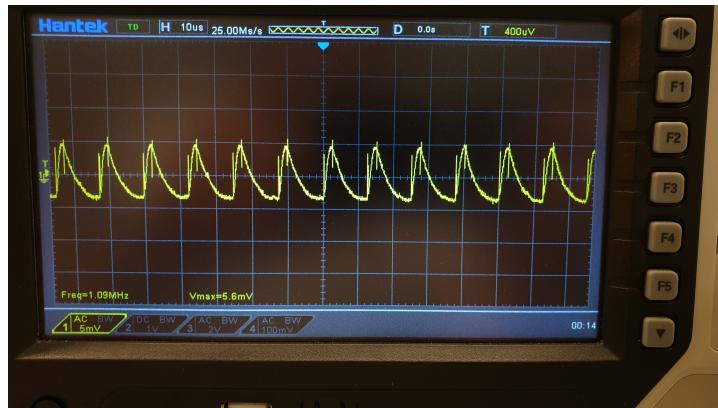


Figur 5: Bild av simulerad oscillator med dimensionerad frekvens 117 kHz, tagen i Multisim.

Kretsen som dimensionerades för 100 kHz, som använde ett 555-chip, gav signaler som endast skilje sig från omgivningens störningar med någon millivolt, och vars frekvens rimligtvis kunde antas vara ungefär 10 gånger större än 100 kilohertz. Då spolen som ingick i kretsen gick sönder kunde fler tester inte utföras.

Kretsen som skulle ge 117 kHz, med 556-chippet, gav en signal med amplitud på cirka 5,5 mV, och frekvens som varierade mellan 115 kHz och 130 kHz. Signalen var inte helt sinusformad men oscillerade mellan positiv och negativ spänning i sågtandsliknande vågor. Se figur 6.

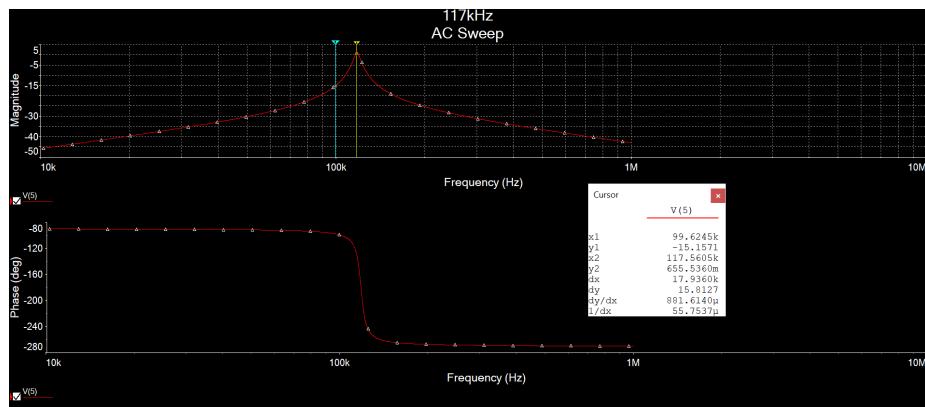
De komponenter som användes hade inte exakta värden enligt mätningar som gjordes vid kopplingen. Resistorerna hade felmarginaler mellan 1 och 5%, medan kondensatorerna hade felmarginaler från 10 till 30%.



Figur 6: Oscilloskopbild av signal från 117 kHz-oscillatorkretsen. Nere till vänster kan man se spänningens maxvärde, och hur stor frekvens signalen har (med störningar).

4.2 Filterkrets

I simuleringar i Multisim filtrerade filtret bort signaler 17 kHz ifrån center frekvensen med cirka -15 dB. Se figur 7.



Figur 7: Bild av simulerad filter med dimensionerad frekvens 117 kHz, tagen i Multisim.

Filterkretsen gav vid test i verkligheten en oigenkännbar signal.

5 Diskussion och Slutsats

Metoden och resultaten kommer här utvärderas och diskuteras, för att försöka komma till en slutsats.

5.1 Spårning

Att storleken på signalen från spolarna berodde på dess avstånd från spolen som sände ut magnetfältet tyder på att metoden kan användas för att mäta avstånd. Detta kan användas för att mäta förhållande mellan de olika mottagarspolarnas signalers storlek (amplitud). Om detta kan användas för att få fram fingrarnas rotation har ännu inte testats, och inget matematiskt samband har hittats ännu. För att få reda på detta skulle behövas fler tester och fler färdiga delar av spårningen.

5.1.1 Oscillator

Oscillatorerna fungerade inte som planerat, och gav andra resultat än de som hade simulerats. 100 kHz-oscillators svaga och knappt urskiljbara signaler var svåra att tolka, då de påminde om kondensatorers upp- och urladdningscykler när systemet var av (ingen matningsspänning) och blev mer fyatkantsformade när kretsen hade matningsspänning. Den svaga signalen antogs kunna filtrerats och sedan förstärkts, när filterkretsen väl hade blivit byggd. Felet kan ha berott på många faktorer, exempelvis att komponenterna var tvungna att ha mer exakta värden. I simuleringar behövde kondensatorerna vara väldigt exakta för att spänningen i kretsen skulle ge upphov till resonans och bli starkare. Att kondensatorerna hade så stora felfmarginaler kan ha gjort att kretsen inte fungerade som den skulle, men enligt nya simuleringar bör det inte ha påverkat resultatet så mycket.

En annan felkälla kan vara att spolarnas induktans mättes felaktigt. Resultaten från mätningarna har ännu inte verifierats med andra verktyg eller metoder, vilket innebär att mätningarna kan vara felaktiga. Detta upptäckte vi redan vid den andra gångens mätningar, där induktansen drastiskt ändrades från 3,3 mH till 7 mH, trots att spolarna var mycket lika i varvtal, material och storlek. Om den första mätningen var fel finns en större risk att andra mätningen också är fel, och det skulle innebära att kretsens andra komponenter var feldimensionerade.

Ännu en potentiell felkälla som kan förklara signalens låga maxvärdens jämfört med simuleringarna kan vara att respektive 555 och 556 ger för svaga spänningar. Enligt databladet, de guider som följdes och simuleringarna som gjordes skulle 555:an ha en utspänning runt 4 till 5 V, medan den här låg på bara några millivolt. Vad detta beror på är för tillfället okänt, men det är tydligt att 555:an och 556:an har haft stor inverkan på projektets resultat, och speciellt oscillatordelens resultat.

5.1.2 Filter

Filterkretsens syfte var att isolera signalen från ett finger samtidigt som den skulle reducera störningar. Vid första testet visades det sig att filtret introducerade flera störningar till signalen istället för att filtrera ut störningar. Signalen som passerade genom filtret var oigenkännbar och kunde inte kopplas till den insignal som filtret tog emot. En felkälla som kunde kopplas till de ökade störningarna var att vi hade en instabil matningsspänning som störde operationsförstärkaren. Försök att stabilisera matningsspänningen gjordes genom att byta strömförsörjning och koppla in frikopplingskondensatorer. Märkbara förbättringar observerades men signalen var fortfarande inte tillräckligt bra.

5.2 Visualisering

Visualiseringen var så pass färdigställd att det enda som behövdes var att sensor-system skulle ge någon utsignal samt att skapa en översättning från sensordata till handposition.

5.3 Planering

En stor del av projektet under de första månaderna bestod av planering, som undersökningen av de olika andra metoderna som används idag, och den övergripande tidsplaneringen som gjordes för att dela upp arbetsuppgifter och tid.

5.3.1 Undersökning och val av metod

Som beskrivet tidigare gick vi i grupp igenom olika metoder för att mäta och spåra fingerrörelser, samt hur man kan representera rörelserna på en fysisk eller digital hand. Undersökningen gav gruppen insikt om vad som är möjligt, hur svåra metoderna verkade vara, och vad som används idag. Vi valde till slut att använda magnetfält, och bygga våra egna sensorer för att kunna mäta handen, och att representera rörelserna med en hand i Unity, men som beskrivet i metod var den ursprungliga planen att bygga en fysisk hand, och mycket tid i början av projektet tillägnades åt att planera handens funktion och delar, vilket exempelvis kan ses i figur 4. Brist på tid är något genomgående i detta projekt, då många delar tog längre tid än planerat eller, som i den fysiska handens fall, inte användes för projektet, annat än i testsyfte.

Valet av metod kan också diskuteras, då planen var att börja med att använda flexsensorer. Denna plan ändrades fort när vi fick förslag om att försöka använda oscillerande magnetfält och filtrera ut deras signaler istället. Alternativet verkade väldigt spännande för oss i gruppen eftersom denna metod inte var lika beprövad som de andra metoderna, vilket gav projektet en ny riktning: istället för att bygga en robothand som kan styras med egna handrörelser, något som hobbyister gjort i flera år, skulle vi utforska om det är möjligt att mäta finare rörelser, såsom fingrar, med magnetfält. Visualiseringen blev mindre viktig, men fanns fortfarande kvar för att kunna visa resultatet och hjälpa oss komma till slutsatser om metoden. Eftersom detta var något som inte hade några tutorials eller guider online att följa innebar detta också att vi i gruppen fick använda det vi lärt oss under vår gymnasieutbildning

angående elektronik, programmering, samt källkritik. Från ett visst perspektiv kan detta beslut ha varit lite förhastat, just eftersom metoden är så pass okänd och outnyttjad av industrin. Det innebar att vi riskerade att inte hinna klart projektet i tid, eller att konceptet inte skulle fungera alls. Hade vi i gruppen haft ett annat syfte i åtanke, till exempel bara att styra en hand, hade vi valt en annan, mer beprövad metod, för att få ett bra resultat. Att projektets syfte ändrades gjorde dock att detta inte var lika viktigt, även om gruppen fortfarande jobbade mot att få ett resultat i tid till projektets slut.

5.3.2 Dimensionering av kretsar

Flera kretsar dimensionerades under projektet, och de kretsar som arbetades mest på var filter- och oscillatorkretsarna. Som tidigare diskuterat dimensionerades flera olika oscillatorkretsar, först wienbryggor, sen 555 med induktanser runt 3,3 mH, och till sist 555 med induktanser runt 7 mH. Problem med dimensioneringen där handlade om att hitta rätt kondensator och resistorvärdet för att kretsen skulle uppnå resonans, vilket tog mycket tid att räkna ut. När spolarnas induktans korrigeras dimensionerades alla 5 kretsar om med helt nya värden, vilket i sin tur tog mycket tid, som i annat fall kunde använts för att bygga kretsarna på riktigt för att hitta och korrigera fel. Detta till trots lyckades alla kretsar dimensioneras, men väldigt mycket senare än planerat.

Filterkretsarna hade också sina problem och svårigheter. Att hitta ett filter som samtidigt kan förstärka signalen den filtrerar var svårt, men lyckades efter mycket kort tid. Något som upptäcktes förhållandevise sent in i projektets gång var, som tidigare nämnt, att den operationsförstärkare som används i kretsen varit ideal, och att hela kretsen endast fungerar med en ideal operationsförstärkare. Även här lyckades vi ändå lösa problemet genom Texas Instruments egna verktyg, och kunde utgå från kretsen vi fick därifrån istället.

Lärdomar vi drar från dessa kretsdimensioneringar är att inte använda sig av ideala komponenter, och försöka säkerställa att dimensioneringen också fungerar med icke-ideala komponenter. Att också se till att ens komponenter har rätt värden, genom att göra flera separata mätningar kan spara mycket tid, så att man snabbt kan gå vidare till att utföra det man planerat, så att man har tid att åtgärda problem.

5.3.3 Tid

Mycket mer tid gick åt det här projektet än planerat, på grund av oväntade fel eller för att vi bytt metod och syfte. Att planera in mer tid för att lösa problem behövs när man gör projekt som anger teknik, speciellt när det kommer till något som inte gjorts av andra. Ändå är projektet på god väg att bli färdigt, även om det inte är helt klart.

5.3.4 Framtida undersökningar

Om det här projektet skulle tas vidare skulle man fortsätta att arbeta för att få en helhetslösning som kan spåra rörelser för ett finger. För att uppnå detta skulle filter och oscillatorkrets behöva förbättras. Komponenter med

bättre precision skulle behöva användas för att eliminera flera störningar. Om projektet ska föras vidare bör även spolarna som agerar sändare optimeras för att skapa en så stark och störningsfri signal som möjligt.

När signalerna från sensorerna är ordnade skulle de behöva översättas från en spänning till data som kan representeras av den digitala handen i Unity-programmet. Metoden är generellt skalbar vilket innebär att om spårning av ett finger fungerar kan det enkelt skalas upp till spårning av samtliga fingrar. Om fullständig spårning av en hand ska uppnås bör även system för att spåra handledens rörelser. Exempelvis rotation.

5.4 Slutsats

Projektets mål att kunna läsa av hand och fingerrörelser uppnåddes inte. Däremot har en grund lagts där en stor del av teorin för att kunna läsa av fingerrörelser finns. Deltester och laborationer visar på att metoden att mäta magnetfält kan vara en möjlig metod för att läsa in fingerrörelserna. Flera tester och en fungerande prototyp krävs för att dra en avgörande slutsats.

6 Källor

- [1] F. Z. Valentin Bazarevsky, *On-Device, Real-Time Hand Tracking with MediaPipe*, (Hämtad 2022-05-31), Google, 2019. URL: <https://ai.googleblog.com/2019/08/on-device-real-time-hand-tracking-with.html>.
- [2] Wikipedia, *Finger tracking — Wikipedia, The Free Encyclopedia*, <http://en.wikipedia.org/w/index.php?title=Finger%20tracking&oldid=1041223705>, (Hämtad 2022-05-31), 2022.
- [3] Wikipedia, *Flex sensor — Wikipedia, The Free Encyclopedia*, <http://en.wikipedia.org/w/index.php?title=Flex%20sensor&oldid=1080644716>, (Hämtad 2022-05-31), 2022.
- [4] Wikipedia, *Wired glove — Wikipedia, The Free Encyclopedia*, <http://en.wikipedia.org/w/index.php?title=Wired%20glove&oldid=1088757158>, (Hämtad 2022-05-31), 2022.
- [5] N. Golestani och M. Moghaddam, ”Wearable magnetic induction-based approach toward 3D motion tracking”, *Sci Rep*, årg. 11, nr 18905, 2021.
- [6] F. S. Parizi, E. Whitmire och S. Patel, ”AuraRing: Precise Electromagnetic Finger Tracking”, *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, årg. 3, nr 4, dec. 2019. DOI: 10.1145/3369831. URL: <https://doi-org.offcampus.lib.washington.edu/10.1145/3369831>.
- [7] F. Santoni, A. De Angelis, A. Moschitta och P. Carbone, ”MagIK: A Hand-Tracking Magnetic Positioning System Based on a Kinematic Model of the Hand”, *IEEE Transactions on Instrumentation and Measurement*, årg. 70, s. 1–13, 2021. DOI: 10.1109/TIM.2021.3065761.
- [8] M. Kazi och M. Bill, *Robotic Hand Controlled by Glove Using Wireless Communication*, 2020.
- [9] C. Konnaris, C. Gavriel, A. A. Thomik och A. A. Faisal, ”EthoHand: A dexterous robotic hand with ball-joint thumb enables complex in-hand object manipulation”, i *2016 6th IEEE International Conference on Bio-medical Robotics and Biomechatronics (BioRob)*, 2016, s. 1154–1159. DOI: 10.1109/BIOROB.2016.7523787.
- [10] A. Kullander Sjöberg, S. Nilsson, L. Boström och B. Ekstig, *Fysik för gymnasieskolan 1 och 2 Digital*. Natur & Kultur, 2019.
- [11] R. Dekker, *A Simple Method to Measure Unknown Inductors*, <https://www.dos4ever.com/inductor/inductor.html>, (Hämtad 2022-05-31).
- [12] Wikipedia contributors, *Operational amplifier — Wikipedia, The Free Encyclopedia*, https://en.wikipedia.org/w/index.php?title=Operational_amplifier&oldid=1086669232, (Hämtad 2022-05-31), 2022.
- [13] Wikipedia contributors, *Electronic filter — Wikipedia, The Free Encyclopedia*, https://en.wikipedia.org/w/index.php?title=Electronic_filter&oldid=1052998098, (Hämtad 2022-05-31), 2021.
- [14] Wikipedia contributors, *Active filter — Wikipedia, The Free Encyclopedia*, https://en.wikipedia.org/w/index.php?title=Active_filter&oldid=1014412101, (Hämtad 2022-05-31), 2021.

- [15] Wikipedia contributors, *Band-pass filter* — Wikipedia, The Free Encyclopedia, https://en.wikipedia.org/w/index.php?title=Band-pass_filter&oldid=1079536663, (Hämtad 2022-05-31), 2022.
- [16] jojo, *Band Pass Filter*, <https://www.circuitstoday.com/band-pass-filters>, (Hämtad 2022-05-31), 2011.
- [17] Electronics Tutorials, *The Wien Bridge Oscillator*, https://www.electronics-tutorials.ws/oscillator/wien_bridge.html, (Hämtad 2022-05-31).
- [18] Learning about Electronics, *How to Build a Sine Wave Generator with a 555 Timer Chip*, <http://www.learningaboutelectronics.com/Articles/Sine-wave-generator-circuit-with-a-555-timer.php>, (Hämtad 2022-05-31).
- [19] *Arduino® Nano 33 BLE*, ABX00030, Rev. 1, Arduino, april 2021.
- [20] Bluetooth, *Bluetooth Wireless Technology*, <https://www.bluetooth.com/learn-about-bluetooth/tech-overview/>, (Hämtad 2022-05-31), 2022.
- [21] Unity, *Wondering what Unity is? Find out who we are, where we've been and where we're going*, <https://unity.com/our-company>, (Hämtad 2022-05-31), 2022.

Bilaga A

Kod Spolsnurrare

```
#include <Arduino.h>

#define hastighetssensor 12 //D6
#define motorPinDir 0 //D2
#define motorPinSpeed 5 //D1

unsigned long n_pulses = 0;
unsigned long time1 = 0;
unsigned long time2 = 0;
int n_turns = 0;
bool one = false;

ICACHE_RAM_ATTR void sensorCallback() {
    n_pulses++;
}

void setup() {
    // put your setup code here, to run once:
    Serial.begin(115200);

    attachInterrupt(digitalPinToInterrupt(hastighetssensor),
                    sensorCallback, RISING);

}

void loop() {

    if (n_pulses % 96 == 0 && one) {
        n_turns++;
        one = false;
        Serial.print(n_turns);
        Serial.print(" Since last: 0.");
        Serial.print((millis() - time1));
        Serial.print(" ms ETA: ");
        Serial.print(((millis() - time1) *
                     (1000-n_turns))/(1000*60));
        Serial.print(" min ");
        Serial.print(((millis() - time1) *
                     (1000-n_turns))/(1000)%60);
        Serial.println(" s");
        time1 = millis();
    }
}
```

```
if (n_pulses % 96 != 0) {
    one = true;
}

if (n_turns >= 1000) {

    digitalWrite(motorPinDir, 1);
    analogWrite(motorPinSpeed, 0);
}
else {
    digitalWrite(motorPinDir, 1);
    analogWrite(motorPinSpeed, 512);
}

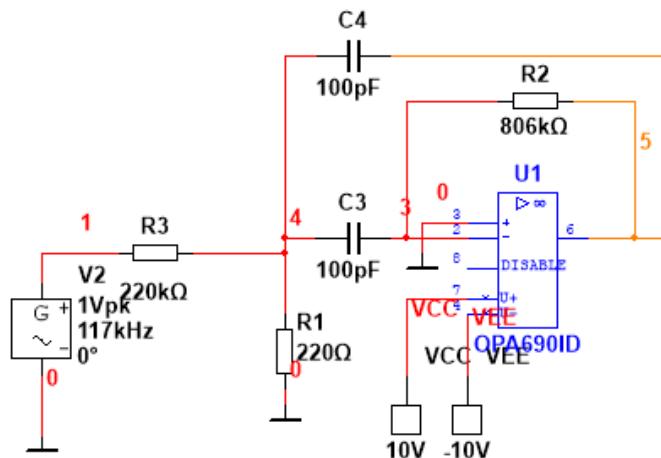
// put your main code here, to run repeatedly:

}
```

Bilaga B

Filterkrets

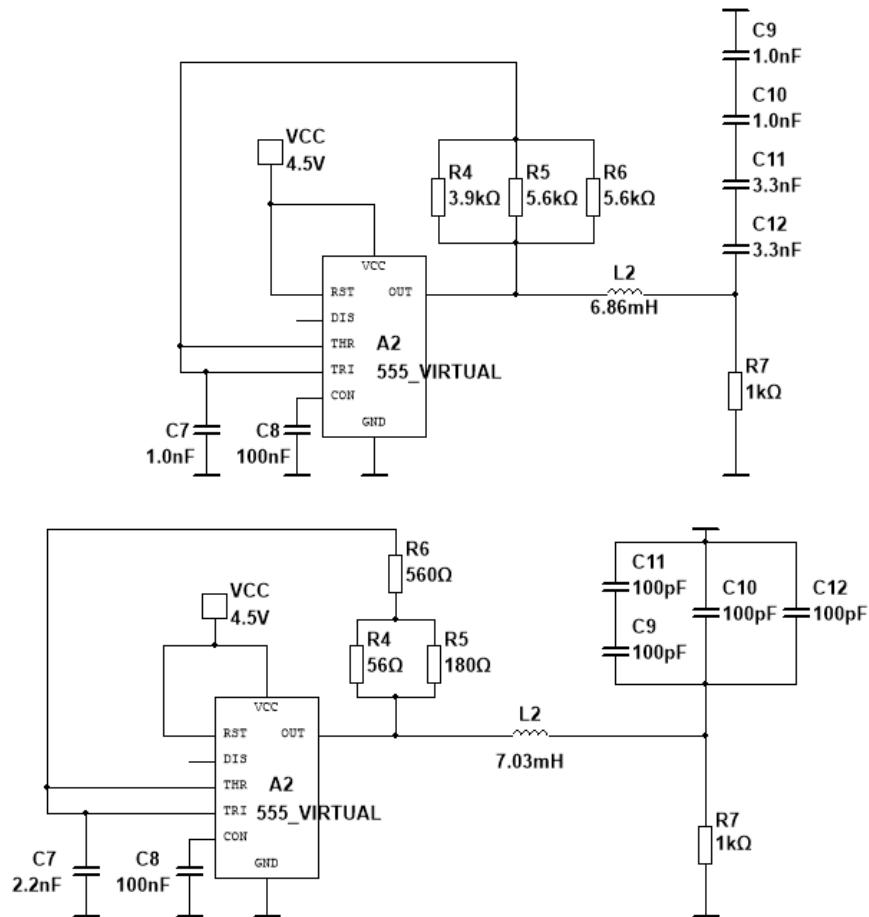
Krets för filter 117kHz. Kretsen är gjord i Multisim.



Bilaga C

Oscillatorkretsar

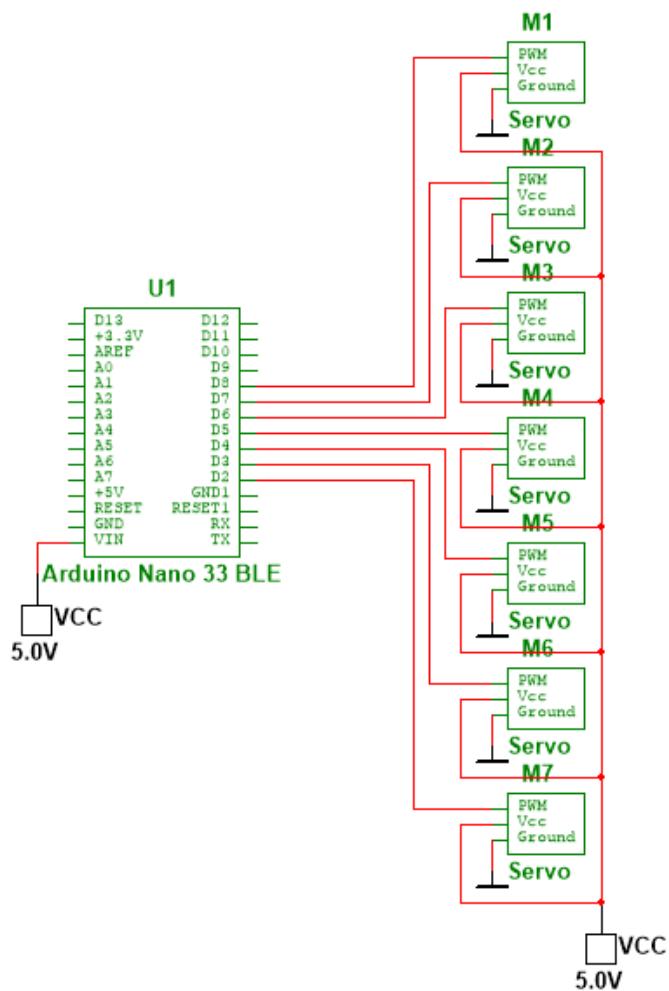
Kretsar för 100 kHz (överst) respektive 117 kHz (underst). Kretsarna är gjorda i Multisim.



Bilaga D

Servokrets

Krets för servo. Kretsen är gjord i Multisim.



Bilaga E

Kod Arduino Peripheral

```
#include <Arduino.h>
#include <ArduinoBLE.h>
#include <functions.h>

const char *deviceServiceUuid = "190H";
const char *deviceServiceCharacteristicUuid = "190F";

String value;

String val = "0100,0100,0100";
unsigned long prevTime;
int advertising;

BLEService handService(deviceServiceUuid);
BLEStringCharacteristic
→ fingerCharacteristic(deviceServiceCharacteristicUuid, BLERead
→ | BLEWrite | BLENotify, 14);

void setup()
{
    Serial.begin(9600);
    while (!Serial)
        ;
    if (!BLE.begin())
    {
        Serial.println("- Starting BLE module failed!");
        while (1)
            ;
    }
    BLE.setLocalName("Arduino Handroid");
    BLE.setAdvertisedService(handService);
    handService.addCharacteristic(fingerCharacteristic);
    BLE.addService(handService);
    fingerCharacteristic.writeValue(val);
    BLE.advertise();

    Serial.println("Nano 33 BLE (Peripheral Device)");
    Serial.println(" ");
}

void loop()
{
    BLEDevice central = BLE.central();
```

```

Serial.println("- Discovering central device...");

if (central)
{
    Serial.println("* Connected to central device!");
    Serial.print("* Device MAC address: ");
    Serial.println(central.address());
    Serial.println(" ");

    prevTime = millis();
    while (central.connected())
    {
        writeValues();
    }

    Serial.println("* Disconnected to central device!");
}
}

void writeValues()
{
    fingerCharacteristic.writeValue(val);
    if (millis() - prevTime > 5000)
    {
        if (val == "0100,0100,0100")
        {
            val = "1000,1000,1000";
        }
        else
        {
            val = "0100,0100,0100";
        }
        prevTime = millis();
    }
}

```

Bilaga F

Kod Rendering

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using System;

public class handController : MonoBehaviour
{

    public Demo bluetoothScript;
    float speed = 0.1f;
    int[] orientation;

    public GameObject index_a;
    public GameObject index_b;
    public GameObject index_c;

    void Start()
    {
        GameObject bluetoothGameObject = GameObject.Find("Demo");
        bluetoothScript =
→ bluetoothGameObject.GetComponent<Demo>();
    }

    void Update()
    {
        orientation = Array.ConvertAll<string,
→ int>(bluetoothScript.subscribeText.text.Split(','),int.Parse);

        // Test to verify that Arduino Nano 33 BLE can control
→ Unity hand model
        if (orientation[0] == 100)
        {
            Quaternion toRotation = Quaternion.Euler(0, 0, 0);
            index_a.transform.localRotation =
→ Quaternion.Lerp(index_a.transform.localRotation, toRotation,
→ Time.time * speed);
            index_b.transform.localRotation =
→ Quaternion.Lerp(index_b.transform.localRotation, toRotation,
→ Time.time * speed);
            index_c.transform.localRotation =
→ Quaternion.Lerp(index_c.transform.localRotation, toRotation,
→ Time.time * speed);
        }
        if (orientation[0] == 1000)
        {
```

```
        Quaternion toRotation = Quaternion.Euler(0, 0, -85);
        index_a.transform.localRotation =
    ↵ Quaternion.Lerp(index_a.transform.localRotation, toRotation,
    ↵ Time.time * speed);
        index_b.transform.localRotation =
    ↵ Quaternion.Lerp(index_b.transform.localRotation, toRotation,
    ↵ Time.time * speed);
        index_c.transform.localRotation =
    ↵ Quaternion.Lerp(index_c.transform.localRotation, toRotation,
    ↵ Time.time * speed);
    ↵ }

}
}
```