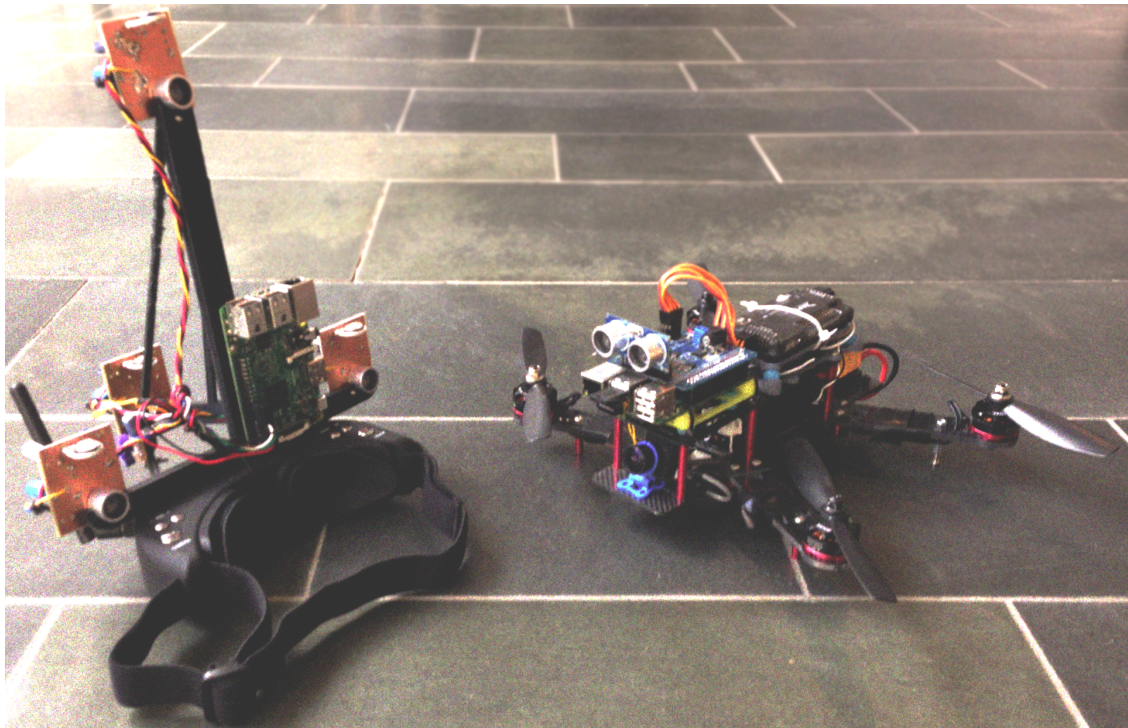




CHALMERS



Tredjepersonsvy i verkligheten

Autonomt styrd drönare med videoströmning i realtid

Kandidatarbete inom Data- och Informationsteknik

JOHANNES ARVIDSSON
ANDREAS FORSBERG
SIMON LINDKVIST
JOHAN SEGERLUND
AUGUST TENFÄLT
DAVID THIRINGER

KANDIDATARBETE

Tredjepersonsvy i verkligheten

Autonomt styrd drönare med videoströmning i realtid

JOHANNES ARVIDSSON
ANDREAS FORSBERG
SIMON LINDKVIST
JOHAN SEGERLUND
AUGUST TENFÄLT
DAVID THIRINGER



CHALMERS

Institutionen för Data- och Informationsteknik
CHALMERS TEKNISKA HÖGSKOLA
Göteborg 2016

The Author grants to Chalmers University of Technology the non-exclusive right to publish the Work electronically and in a non-commercial purpose make it accessible on the Internet. The Author warrants that he/she is the author to the Work, and warrants that the Work does not contain text, pictures or other material that violates copyright law.

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

Third person view in reality
Autonomous drone control with real time video feed

JOHANNES ARVIDSSON
ANDREAS FORSBERG
SIMON LINDKVIST
JOHAN SEGERLUND
AUGUST TENFÄLT
DAVID THIRINGER

© JOHANNES ARVIDSSON, June 2016.
© ANDREAS FORSBERG, June 2016.
© SIMON LINDKVIST, June 2016.
© JOHAN SEGERLUND, June 2016.
© AUGUST TENFÄLT, June 2016.
© DAVID THIRINGER, June 2016.

Examiner: Arne Linde

Chalmers University of Technology
Department of Computer Science and Engineering
SE-412 96 Gothenburg
Sweden
Telephone + 46 (0)31-772 1000

Department of Computer Science and Engineering
Gothenburg, Sweden, June 2016

Abstract

The purpose of this project is to implement the concept of third person view. Third person view is commonly used in video games, where the player sees the controlled character from behind rather than from the characters eyes. The project idea was given by Chalmers University of Technology and was also executed there.

The product is constructed to prove a concept and thus has not been adapted for mass production. The project group has therefore not strived to find the cheapest components or to design and build the product in an aesthetically pleasing way. The product will not be required to identify objects that could get in its way, which means it should be used in open areas.

To solve this task the project uses a drone, a camera and a video display. The video player receives continuous data from a camera mounted on the drone. For the drone to be able to follow the user, a control system has been designed to ensure the drone keeps the same distance and angle from the user at all time. Data about the drones position is gathered by using ultrasound pulses. To measure altitude the drone uses a barometer. The ultrasound pulses which are used for measuring the distance between the user and the drone uses several receivers to calculate at which angle the drone is relative to the user. Calculations are made with the single card computer Raspberry Pi.

The product is not fully implemented as the drone is not able of autonomous control. The different systems that together would be able to make a complete product have been constructed and each one work in themselves, but combining and testing of all subsystems have not been conducted.

Keywords: Drone, third person view, video headset, wireless measurement of position, ultrasound

Sammandrag

Projektets syfte är att implementera konceptet tredjepersonsvy i verkligheten. Tredjepersonsvy är vanligt förekommande i datorspel, där spelaren ser sin karaktär bakifrån istället för att se spelet från karaktärens ögon. Projektet beställdes av Chalmers Tekniska Högskola och är även utfört där.

Produkten har konstruerats för att bevisa ett koncept och har således inte anpassats efter att kunna massproduceras. Projektgruppen har därför inte tagit hänsyn till att finna de billigaste komponenterna eller att utforma och bygga produkten estetiskt tilltalande. Produkten ska inte heller behöva kunna identifiera föremål som kan tänkas förhindra dess framfart, vilket innebär att den bör användas på platser med fri yta.

Projektets uppgift realiseras genom att använda en drönare, en kamera och en videouppspelningsenhet. Videouppspelningsenheten får kontinuerlig data från en kamera på drönaren. För att drönaren ska kunna följa efter användaren byggs ett regelsystem som ser till att drönaren ligger på samma avstånd från användaren hela tiden. Data om drönarens position samlas in genom användning av ultraljudspulser. För att upptäcka höjd över mark använder drönaren sig av en barometer. Ultraljudspulserna som bedömer avståndet mellan användare och drönare använder sig av flera mottagare för att kunna beräkna vilken vinkel drönaren har relativt användaren. Beräkningarna sker med enkortsdatorn Raspberry Pi.

Produkten är inte fullt implementerad då drönaren saknar förmåga att styra sig själv. De olika systemen som kan bidra till en färdig produkt är konstruerade och fungerar var för sig efter uppsatta krav, men en sammanfogning och testning av alla delsystem tillsammans har ej utförts.

Nyckelord: Drönare, tredjepersonsvy, videoheadset, trådlös positionsbestämning, ultraljud.

Erkännanden

Ett stort tack till vår handledare Roger Johansson som har hjälpt till under arbetets gång. Vi vill även tacka Arne Linde och Lars Norén för hjälp med beställning av hårdvara samt tips och råd de givit oss. Tack också till kandidatgruppen som delade lokal med oss för det trevliga sällskap detta inneburit. Till sist vill vi tacka Chalmers bibliotek för deras stöd kring rapportskrivning samt alla personer som har hjälpt till att läsa och granska rapporten.

Författarna, Göteborg, 1 juni 2016

Nomenklatur

Drönare	Ett obemannat luftfartyg.
Enkortsdator	En minidator som oftast används till mindre projekt. Arduino och Raspberry Pi är två exempel på dessa.
ESC	Engelska: <i>Electronic speed controller</i> . Elektronisk varvtalsregulator.
FPV	Engelska: <i>First Person View</i> . Förstapersonsvy. Vyn som syns från ögonperspektiv.
FPV-glasögon	Engelska: <i>First Person View Goggles</i> . Förstapersonsvyglasögon. Glasögonen tar emot video via antenner för direkt uppspelning av videon via inbyggda skärmar.
Frihetsgrad	Beskriver hur många oberoende variabler som behövs för att entydigt ange läget för ett system.
GPIO	Engelska: <i>General-purpose input/output</i> . Samlingsnamn för in- och utportar som saknar specifika användningsområden.
I ² C-buss	Engelska: <i>Inter-Integrated Circuit</i> . En synkron seriell multimasterbuss som används för att koppla låghastighetsenheter till enkortsdatorer, moderkort, mobiltelefoner med mera.
Kvadrokopter	En drönare med fyra rotoror. Rapportens svenska motsvarighet till engelskans <i>quadcopter</i>
PID-regulator	Engelska: <i>Proportional, Integral, Derivative regulator</i> . Regulatortyp som dimensioneras för att ta olika stor hänsyn till det proportionella, integralen av, och derivatan av felet.
Pitch	Engelskans motsvarighet för tippning. Rotation kring tippaxel.
PWM	Engelska: <i>Pulse Width Modulation</i> . Analog representation av digitala värden, åstadkoms genom olika dimensionering av fyrkantsvågor.
Python	Ett programmeringsspråk.
Roll	Engelskans motsvarighet för rullning. Rotation kring längdaxel.
TCP och UDP	Engelska: <i>Transmission Control Protocol (TCP) and User Datagram Protocol (UDP)</i> . Två olika protokoll som används vid digital dataöverföring mellan nätverk.
VR	Engelska: <i>Virtual Reality</i> . Virtuellt verklighet.
Wi-Fi	Trådlös nätverkskommunikation.
Yaw	Engelskans motsvarighet för gir. Rotation kring giraxel.

Materiallista

1N4937	Likriktardiod
ArduPilot	En flygdator för autonom styrning av obemannade drönare.
FPV-kamera	Liten och lätt kamera för användning vid drönarflygning.
HC-SR04	Engelska: <i>Ultrasonic Ranging Module</i> . En ultraljudsmodul för avståndsmätning.
Lf353	Operationsförstärkare, Texas Instruments
Lt1026	Spänningsinverterare
LSM303	Modul med en treaxlig accelerometer och en magnetometer (kompass).
NE12S0A0V06PNFA	Switchregulator för spänning, Delta Electronics
Raspberry Pi	En enkel enkortsdator som brukar användas i projekt av mindre skala.
RS2205 2300kv	Elmotor, EMax Copter
Skyzone	FPV-glasögon som använts under projektet.
X-Rotor	15A ESC, EMax Copter



Innehåll

1	Inledning	1
1.1	Bakgrund	1
1.2	Syfte	2
1.3	Precisering av frågeställningen	2
1.4	Avgränsningar	2
1.5	Metod	3
1.6	Rapportens disposition	3
2	Teoretisk bakgrund	5
2.1	Videoöverföring och videorepresentation	5
2.2	Autonom styrning av drönare	6
2.2.1	Positionering och orientering	6
2.2.2	Styrning av kvadrokopter	7
2.2.3	Reglering av externa styrsignaler	9
3	Teknisk analys	11
3.1	Problemanalys	11
3.2	Avståndsmätning	12
3.3	Trilateration	12
3.4	Orientering	13
3.5	Centrering av kamera med bildigenkänning	13
3.6	Videouppspelning och videoöverföring	13
3.7	Reglering av position	14
3.8	Datakommunikation	14
3.9	Valda lösningar	15
4	Resultat	17
4.1	Systemdesign	17
4.2	Positioneringsdata	18
4.3	Styrning och reglering	20
4.4	Vidorepresentation	21
5	Diskussion	23
5.1	Videosystem	23
5.2	Position	23
5.3	Regulator och Styrning	25
5.4	Ur samhällets synvinkel	26

5.5 Arbetsgång	27
6 Slutsats	29
Referenser	31
A Appendix 1	I
B Appendix 2	III
C Appendix 3	VII

1

Inledning

I detta kapitel ges en övergripande blick av projektets utformning och tillkomst, frågeställningen preciseras och projektets avgränsningar beskrivs.

1.1 Bakgrund

Förstapersonsvy och tredjepersonsvy är koncept som är vanligt förekommande inom dator- och TV-spel. Koncepten kan jämföras med det svenska språkets termer första, andra och tredje person. I språk beskriver första person den som talar, andra person beskriver den tilltalade, och tredje person beskriver andra utomstående aktörer. Tredjepersonsvy kan likt detta avse flera olika fasta eller rörliga åskådarperspektiv så som perspektivet från en övervakningskamera, perspektivet som uppvisas i vanliga TV-serier och filmer, eller det efterföljande perspektivet som är vanligt i dator- och TV-spel. I denna rapport syftar begreppet tredjepersonsvy på det perspektiv som följer målet bakifrån och lätt uppifrån, se Figur 1.1. Förstapersonsvy representeras i Figur 1.2.

Studenter vid Chalmers Tekniska Högskola har tidigare försökt utveckla system som realiserar tredjepersonsvy med hjälp av drönare, men försöken har inte resulterat i fungerande prototyper, då svårigheter uppstått i att mäta drönarens position relativt användaren. Studierna fann även brister i videolänken, då fördröjningen mellan upptagning och uppspelning blev hög [1,2]. Drönare (obemannade flygande farkoster) används i stor utsträckning för fotografering och videofilmning på grund av möjligheten att nå annars otillgängliga perspektiv.



Figur 1.1: *Tredjepersonsvy*



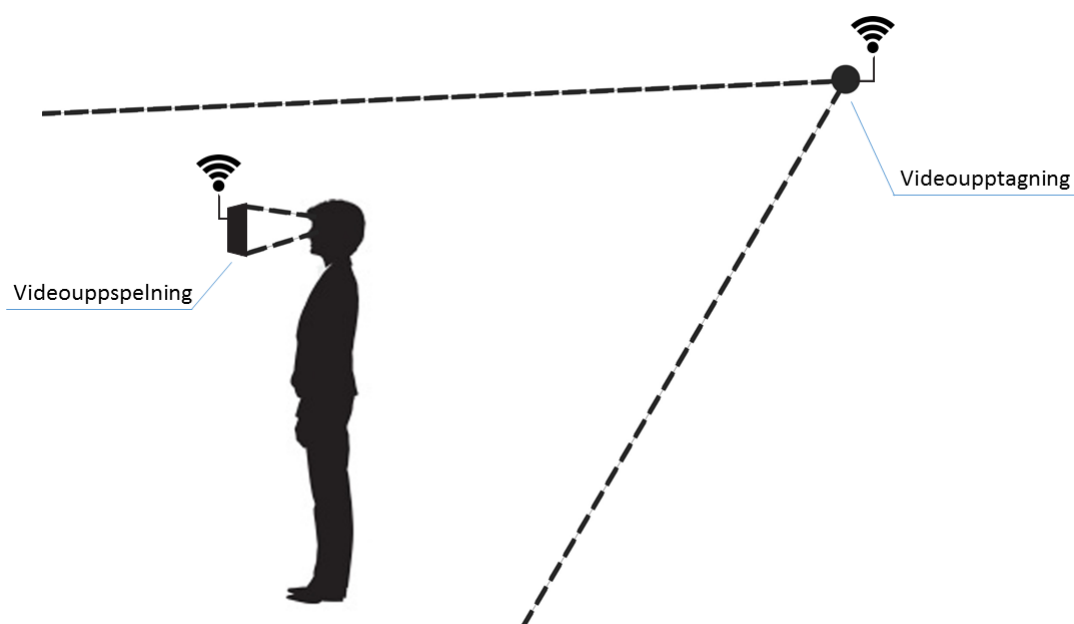
Figur 1.2: *Förstapersonsvy*

1.2 Syfte

Projektet syftar till att tillverka en produkt som realiserar tredjepersonsvy i verkligheten. Uppdraget är beställt av Chalmers Tekniska Högskola och går ut på att bygga en prototyp för att bevisa att konceptet är genomförbart i praktiken.

1.3 Precisering av frågeställningen

Systemet ska i grunden vara uppbyggt med att en drönare flyger bakom användaren och filmar från ett tredjepersonsperspektiv. Med hjälp av egenvalda sensorer, positionerings- och styrlogik ska drönaren styras autonomt för att hålla sin position relativt användaren. Den video som drönaren tar upp ska spelas upp för användaren i realtid, och överföringen av data till användaren ska vara trådlös. Användaren ska få videon uppspelad via ett videoheadset. Konceptet illustreras i Figur 1.3, där användaren ser det som drönaren filmar i realtid.



Figur 1.3: *Konceptöverblick*

Rapporten kommer utvärdera vilka tekniska system och lösningar som är passande för att uppnå ett önskat resultat inom positionsbestämning, styrning, videoupptagning, videouppspelning samt dataöverföring.

1.4 Avgränsningar

Prototypen behöver inte vara uppmärksam på sin omgivning utan ska endast följa och befinna sig på ett bestämt avstånd från användaren. Systemet kan följaktligen inte upptäcka om drönaren är på väg in i en vägg eller andra hinder, och bör därför

endast användas i öppna miljöer. Prototypen behöver inte vara prisvärd eller tilltalande för konsumenter, då den endast tas fram som en prototyp. Vid val av hårdvara och vid systemdesign utvärderas därför inte pris eller estetik. En kvadrokopter användes då denna tillhandahölls av Chalmers Tekniska Högskola. Med bakgrund av detta har inga andra typer av drönare undersökts.

1.5 Metod

Projektet initierades genom att sätta upp vilka mål och prioriteringar som prototypen ska uppfylla. Val av prioriteringar är viktiga eftersom det bidrar till en tydligare väg mot resultatet. Projektet delades in i delsystem; video och datalänk, styrning och reglering samt positionering av drönaren. För dessa sammanställdes en prioriteringslista för prestandaparametrar i ordning av högst prioriterad till lägst prioriterad parameter:

- Fördröjning mellan videoupptagning och videouppspelning
- Upplösning och allmän bildkvalité på videouppspelning
- Tidsåtgång för mätning och uträkning av position
- Tidsåtgång för korrigeringsfel
- Möjlighet att definiera en godtycklig referensposition för drönaren

Utifrån dessa parametrar utfördes en litteraturstudie för att finna lämpliga tekniska lösningar, vilket resulterade i rapportens kapitel *Teknisk analys*. De olika tekniska lösningarna utvärderades och ställdes mot varandra för att sedan komma till ett slutgiltigt val. Efter detta påbörjades genomförandefasen och implementation av de valda lösningarna. Arbetet valdes att utföras på ett iterativt sätt, vilket innebär att lösningarna testades omgående under implementationen och eventuella problem behandlades i efterkommande iteration. Fokus låg på fungerande prototyper och en snabb implementationsfas eftersträvades genom att använda färdig hårdvara så mycket som möjligt, och om nödvändigt modifiera befintliga lösningar i stället för att designa och bygga egna lösningar från grunden.

1.6 Rapportens disposition

Rapportens introduktion följs av ett teorikapitel där en teoretisk grund för projektet presenteras. Efter detta ges en genomgång av olika relevanta teknologier och möjliga lösningar i kapitlet Teknisk analys. I resultatkapitlet presenteras den slutgiltiga lösningen, vald hårdvara och detaljer om systemet. Diskussionskapitlet går igenom vad som varit för- och nackdelar med den valda lösningen och reflektioner kring vad som hade kunnat göras annorlunda. Slutligen ges en sammanfattning och rekommendationer för framtida forskning samt utvärdering av resultat och metodval i kapitlet Slutsats.

2

Teoretisk bakgrund

I detta kapitel läggs teoretisk grund för de teknologier och koncept som används i projektet. Kapitlet utgår från de funktioner som identifierats som nödvändiga för systemet i problemformuleringen.

2.1 Videoöverföring och videorepresentation

För att sända videodata mellan kamera och videouppspelningsenheten så krävs en trådlös datalänk. Trådlös kommunikation kan åstadkommas med teknologier som Bluetooth och Wi-Fi. Bluetooth är en teknologi där enheterna inte behöver vara anslutna till ett nätverk men behöver befinna sig nära varandra, vilket används i bland annat trådlösa hörlurar och högtalare [3]. En annan teknologi är Wi-Fi, där datan skickas mellan två datorer i ett nätverk. Vid överföring i ett Wi-Fi nät så gör datorn om datan till en analog radiosignal som skickas till den andra enheten. Wi-Fi kommunicerar över nätverk med bärfrekvens $2,4GHz$ eller $5GHz$. När mottagardatorn tar emot den analoga signalen så görs den om till en digital signal i datorn, vilket tillåter att videodatan kan redigeras innan den skickas vidare till en uppspelningsenhet.

Data kan även skickas direkt via radiosignaler. Vid användning av radiosignaler så finns konventioner för vilka frekvenser som får lov att användas. Restriktioner av frekvenser finns för att inte störa viktiga signaler. Olika frekvenser ger olika egenskaper för överföring där lägre frekvenser ($900MHz$) går lättare igenom föremål medan högre frekvenser ($5,8GHz$) har längre räckvidd samt kan ha en antenn av mindre storlek [4]. Då sträckan som datan ska skickas över i projektet enbart är ett par meter, samt att det är över en öppen yta, kommer frekvensen enbart bestämmas så den inte stör ut andra system.

Virtuell verklighet, ofta kallat VR (*Virtual Reality*), innebär en datorsimulerad verklighet där en dator genererar en värld som ska upplevas som verklighet. För att övertyga användaren om att datorsimuleringen är verklighet så används VR-glasögon som är en uppspelningsenhet som täcker användarens hela synfält. VR-glasögon kännetecknas utav de inbyggda sensorer de har för att tillåta användaren att se sig om i den virtuella världen. Moderna VR-glasögonen har också möjlighet att spela upp två videoströmmar samtidigt, en för vardera öga, vilket gör det möjligt för användaren att uppleva ett tredimensionellt djup i bilden [5, 6].

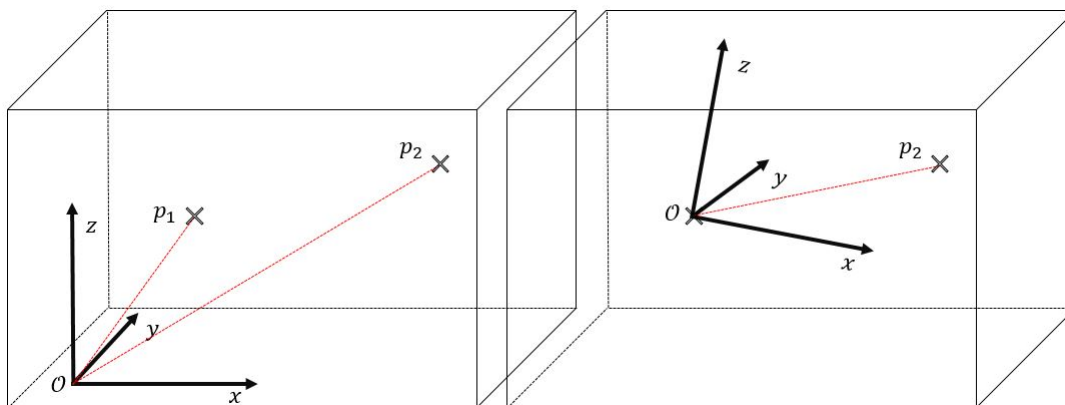
2.2 Autonom styrning av drönare

För att åstadkomma en god tredjepersonsvyupplevelse krävs att drönaren styrs helt autonomt. Detta innebär att drönarens relativa position och orientering behöver bestämmas. Återkoppling används för att jämföra positionerings- och orienteringsdatan med referensvärdena via en regulator, vilket ger de faktiska styrsignalerna för drönaren.

Olika drönartyper så som kvadrokopter, hexakopter med flera har olika typer av intern reglering av styrning. Utöver detta kan externa styrsignalerna så som gas eller rullning regleras innan de skickas till drönarens flygdator. Reglering av externa signaler möjliggör snabbare, stabilare eller mer exakt styrning.

2.2.1 Positionering och orientering

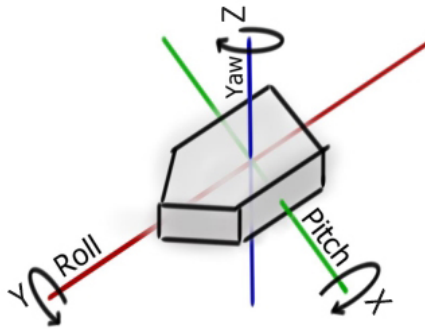
För positionsbestämning av ett eller flera objekt måste ett systemglobalt koordinatsystem definieras. Detta kan vara av olika typer, så som kartesiskt, cylindriskt eller sfäriskt. Det måste även ha en definierad utgångspunkt, origo. I detta projekt hanteras två objekt med var sin position, användaren och drönaren. Då uppstår två sätt att definiera koordinatsystemet: absolut eller relativt. I det absoluta koordinatsystemet definieras koordinatsystemet från ett externt rum, och de båda punkterna är således explicit definierade inom rummet. I ett relativt koordinatsystem definieras koordinatsystemet utifrån punkt 1, och punkt 2 definieras utifrån dess relativa position till punkt 1, se Figur 2.1. Notera även att det relativa koordinatsystemet följer orienteringen av punkt 1 samt att systemet nu är fullkomligt omedvetet om rummet.



Figur 2.1: Absolut respektive relativt koordinatsystem

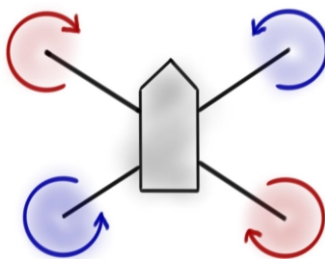
2.2.2 Styrning av kvadrokopter

Styrningen utav en kvadrokopter skiljer sig märkbart från andra typer av drönare. Med dess unika utformning behövs inga komplexa mekaniska styrsystem, istället används fyra motorer för att förlytta drönaren längs med, och rotera runt X-Pitch-, Y-Roll- och Z-Yaw-axlarna, se Figur 2.2.



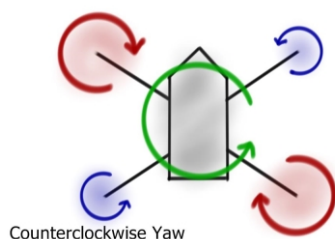
Figur 2.2: Bild på hur X-, Y- och Z-axlarna spänner upp koordinatsystemet. *Koordinatsaxlar* av Scott Hatfield licensierad under [CC BY-NC-SA 4.0](https://creativecommons.org/licenses/by-nc-sa/4.0/)

Genom att motorerna håller en högre rotationshastighet på höger eller vänster sida kommer kvadrokoptern vrida sig runt sin Y-axel på grund utav den ökade lyftkraften. Rotationen runt X-axeln baseras på samma princip genom att öka rotationshastigheten i den främre eller bakre delen av kvadrokoptern. Rotationen runt Z-axeln är mer komplicerad och bygger på Newton's tredje lag [7].



Figur 2.3: Bilden visar hur uppsättningen av motorer roterar. *Motorrotationer* av Scott Hatfield licensierad under [CC BY-NC-SA 4.0](https://creativecommons.org/licenses/by-nc-sa/4.0/)

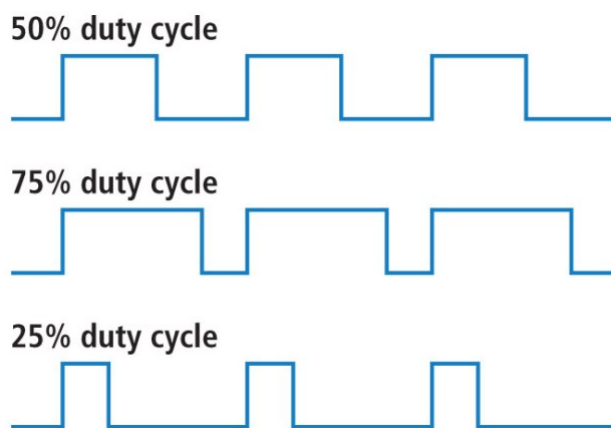
Genom att använda två motsatta motorer som roterar motsols och de resterande roterar medsols påverkas kroppen med lika stort vridmoment från båda hållen och håller sig därmed stabilt runt Z-axeln, se Figur 2.3. För att rotera runt Z-axeln ökas rotationshastigheten på de motorer som roterar i motsatt riktning som önskas, därmed ökas kraften som motorerna påverkar kroppen som vidare får kroppen att rotera vilket illustreras i Figur 2.4 [8].



Figur 2.4: Bilden illustrerar hastigheten av motorrotationerna vid rotation kring Z-axeln. *Rotation runt Z-axeln* av Scott Hatfield licensierad under [CC BY-NC-SA 4.0](#)

En kvadrokopter är på grund av dess utformning ett instabilt system, vilket behöver en kontinuerlig ström av styrningskommandon i hög hastighet för att vara stabil. Dessa interna styrningskommandon sker med hjälp utav ett styrkort som innehåller en accelerometer och ett gyroskop, vilket är två essentiella komponenter för att systemet ska fungera. Accelerometern mäter systemets acceleration längs de tre olika axlarna, medans gyroskopet uppfattar vilken axel den rör sig runt. För förflyttning och rotation kring axlarna analyserar styrkortet den givna insignalen från operatören och beräknar med hjälp av informationen från gyrometern och accelerometern ut den rotationshastighet för motorerna som ger önskad förflyttning.

Insignalerna till styrkortet ges i form av PWM-sigaler (*Pulse Width Modulation*). PWM-sigaler är en metod att läsa av analoga värden genom att skapa en kontinuerligt varierbar effektmatning med hjälp av att slå på och av spänningen snabbare än vad den anslutna apparaten kan urskilja, vilket illustreras i figur 2.5. Det analoga värde som styrkortet läser av bestäms på hur stor del av cykeltiden som spänningen ligger hög. [9].



Figur 2.5: Exempel på PWM-sigaler som representerar olika analoga värden. *PWM-Sigaler* av Sparkfun licensierad under [CC BY-NC-SA 3.0](#)

Utifrån den insignal som styrkortet får vid ett givet tillfälle skickas en utsignal till en varvtalsregulator, även kallad ESC (*Electronic Speed Controller*). Respektive ESC reglerar motorrotationen för att stämma överens med den givna signalen från styrkortet. ESC'n uppfattar inte styrsignalen som en mekanisk rörelse utan som

omkopplingshastigheten av ett nätverk av fälteffekttransistorer, vilket möjliggör en mycket jämnare och precis variation av motorrotationen [10].

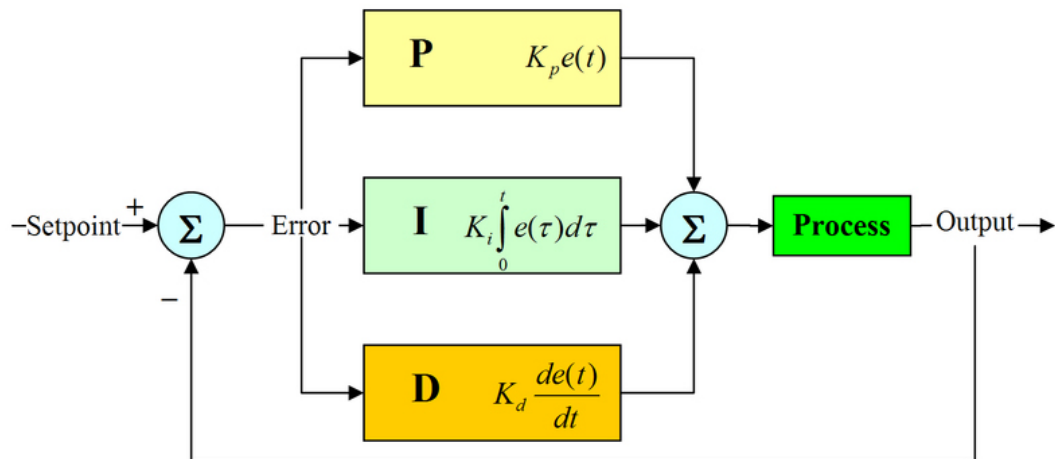
2.2.3 Reglering av externa styrsignaler

Vid autonom styrning används regulatorer för att anpassa styrsignalerna så att systemet blir snabbare, stabilare eller mer exakt. Autonoma system har normalt referensvärden som de utgår ifrån, i detta fall är värdena i form av positions- och orienteringsdata. Styrsignalerna genereras genom styrlogiken utifrån de fel som existerar jämfört referensvärdena, vilket sedan kan regleras för snabbare och jämnare styrning.

Det finns många olika typer av regulatorer av varierande komplexitet och som fungerar olika väl beroende på tillgänglig data. En förutsättning för effektiv reglering är återkoppling av de parametrar som referensvärden valts för, men detaljerade modeller av dynamiken för systemet används också för så kallad framkoppling. Framkoppling innebär att man projicerar resultatet av olika givna styrsignaler, för att på så sätt kunna förutsäga vart systemet är på väg och hur styrsignalerna skall anpassas.

Vid val och dimensionering av regulator kan man antingen ta fram och ställa upp detaljerade modeller för sitt system, utgå från prestandatester av olika dimensionering för sina regulatorer, eller använda en kombination av båda. För bästa möjliga reglering krävs en exakt modell av dynamiken i systemet, men i många fall är detta svårt eller omöjligt på grund av externa faktorer och krafter. Dynamiken för en kvadrokopter är avancerad, och innehåller svårmodellerade komponenter så som luftvirvlar och aerodynamiken för drönarens utformning, som är oregelbundna. För svårmodellerade system används frekvent regulatorer med förenklad framkoppling eller helt utan. En regulator typ med återkoppling men helt utan framkoppling är PID-regulatorer (*Proportional, Integral, Derivative*) [11, 12].

En PID-regulator utgår från ett referensvärde, varifrån man subtraherar ett sensorvärde för att beräkna ett fel. Felet används sedan för att dimensionera en styrsignal så att man uppnår referensvärdet. För att anpassa regulatorn för olika system används olika koefficienter för de olika komponenterna av regulatorn. Den proportionella komponenten är det ögonblickliga felet, den integrerade komponenten är summan av felet över tiden, medans den deriverade komponenten är förändringshastigheten i felet, se Figur 2.6 för illustration av en PID-regulators design.



Figur 2.6: *Klassisk regulatordesign med setpoint (referensvärde), PID-komponenter och negativ återkoppling. PID-regulator av SilverStar licensierad under [CC BY-SA 3.0](#)*

3

Teknisk analys

I följande avsnitt presenteras möjliga lösningar för de delproblem som identifierats. Problemen läggs på en hög abstraktionsnivå för att hålla distansen mellan problemen och på så sätt göra det enklare att särskilja och lösa dem.

3.1 Problemanalys

Det ställs många krav på ett system som ska fungera enligt konceptet tredjepersonsvy. Det är svårt att se en helhetslösning av systemen utan att bryta ner problemet i mindre delar. Resultatet av en nedbrytning är tekniska problem inom flera olika områden. Eftersom slutresultatet önskas ha ett system som relativt användaren navigerar drönaren, har alla delproblem utgått från att systemet har ett relativt koordinatsystem.

Ett problem är hur videoupptagning och videouppspelning ska fungera. För att användaren ska kunna se video av sig själv i realtid, behöver systemet en kamera som fångar upp video ur rätt perspektiv och en annan extern enhet som spelar upp videon för användaren. Det slutgiltiga systemet som ska realiseras kräver även en eller flera trådlösa kommunikationskanaler mellan användare och drönare. Den information som behöver överföras är video från kamera, samt positionerings- och orienteringsdata som beskriver vart drönaren befinner sig.

Position och orientering är två andra problem. Drönaren ska orientera och positionera sig själv relativt användaren och markytan, vilket kräver att drönaren vet vilken riktning och avstånd den har mot användaren. Med riktning menas det vilken vridning som drönarens kameran sida har kring sin egna vertikala axel, giraxeln. Målet är att den ska ha likadan vridning som användaren. Då drönaren ska hålla en bestämd höjd ovanför markytan behövs även en metod för höjdmätning.

Systemet på drönaren ska hantera datan som består av höjd, avstånd till användare och orientering. Detta medför att drönaren behöver läsa av dessa positioneringsvärden, beräkna rätt mängd kraft till sina motorer och därmed förlytta sig till önskad position. Utöver positioneringsdatan som ges av systemet på användaren behöver styrsystemet kontinuerligt läsa av relativiteten och hastigheten kring de olika axlarna, för att reglera och förebygga instabiliteten som skapas av konstruktionen.

3.2 Avståndsmätning

En möjlig metod för att bestämma höjden ovanför markytan är att använda sig utav en barometer. En barometer kan med sin avläsning utav lufttryck avgöra vilken höjd den befinner sig på.

En annan möjlighet för avståndsmätning är att låta systemet använda sig utav en ultraljudsmodul. Modulen kan med ett program användas som ekolod, genom att en ultraljudsvåg sänds och en klocka startas för att sedan stoppas när ljudvågen studsat tillbaka. Tidskillnaden kan användas för att utifrån ljudets hastighet i luft beräkna avståndet till det träffade objektet. Det finns enkla moduler med en träffsäkerhet på tre millimeter och som kan hålla en stabil mätning på upp till fyra meters avstånd [13].

Ytterligare ett alternativ skulle kunna vara mätning med laser. Laser är en av de metoder med högst träffsäkerhet och har dessutom god räckvidd. Jämfört med ultraljudsmodulens träffsäkerhet och räckvidd, så finns det laserenheter, av mindre storlek, som kan mäta med 3mm träffsäkerhet upp till ett avstånd på 30m [14].

3.3 Trilateration

För att beräkna drönarens position relativt en användare är trilateration en möjlighet. Metoden används bland annat av GPS (*Global Positioning System*) [15]. Genom att skicka signaler från en okänd punkt till tre kända punkter, eller vice versa, tillsammans med tidsinformation om när pulsen skickades, kan man med vetskap om signalens propageringshastighet erhålla avståndet mellan samtliga kända punkter och den okända punkten. Med denna informationen kan en tetraeder med den sökta punktens position i en av spetsarna ritas, och positionen är således känd, se Figur 3.1.

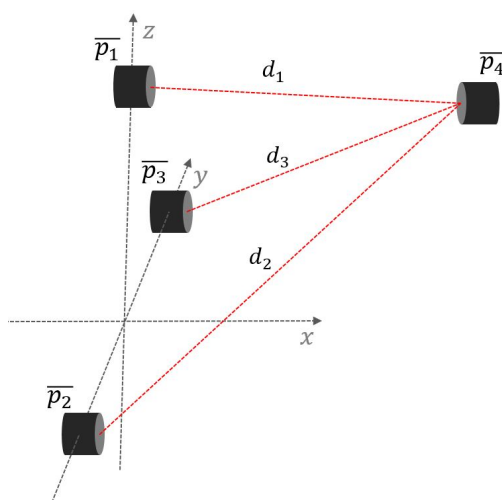


Figure 3.1: Positionen \hat{p}_4 defineras i koordinatsystemet med hjälp av d_1 , d_2 och d_3

Trilateration kan använda olika medier för signaler. Ljud, oftast ultraljud, samt olika frekvenser av elektromagnetiska vågor (radiovågor, infrarött ljus och laser) kan användas.

3.4 Orientering

Drönaren behöver veta och anpassa sig utefter vilken orientering användaren har (vilken riktning användaren tittar åt). Önskat är att drönaren ska orientera sig i samma riktning som användaren. En möjlig lösning för det är att använda sig utav två kompasser, en på drönaren och en på användaren, varav systemet som använder dessa läser av användarens kompass och ser till att drönarens kompass håller sig inom en viss felmarginal genom att skicka styrsignaler till drönaren.

En annan lösning är att använda ett dubbelt och spegelvänt trilaterationssystem, där både användare och drönare är utrustade med en mikrofonarray samt ultraljudspulsmodul.

3.5 Centrerung av kamera med bildigenkänning

För att försäkra att drönarens kamera är riktad och centrerad mot användarens rygg går det att använda en bildhanteringsmetod som på engelska kallas för *Optical Glyph Tracking*, optisk spårning av en bildteckning. Den här metoden tillåter ett system att känna igen ett tvådimensionellt mönster ritat på till exempel en platta och vilken riktning som normalvektorn för planet som plattan spänner ut, är riktad åt [16]. Med den informationen kan drönarens system få nödvändiga styrsignaler för att lokalisera och centrera användaren i bilden.

3.6 Videouppspelning och videoöverföring

En videoström kan överföras trådlöst på olika sätt, möjliga metoder är överföring via Bluetooth, Wi-Fi eller radiosignaler. För att spela upp videoströmmen kan både VR-system och FPV-system (First Person View) användas. Bluetooth och Wi-Fi överför digitala signaler medan radiosignaler överför analoga signaler.

Oculus Rift är ett VR-videoheadset som används vid uppspelning av virtuell verklighet. Oculus Rift behöver digitala insignaler vilket gör att bilddatan måste hanteras på en dator innan den kan spelas upp. Denna hantering av datan kan ge en viss fördröjning. Men att kunna hantera och kontrollera bilddatan ger också möjligheten att bygga system som till exempel kan spåra objekt eller personer med hjälp utav bilddatan. Eftersom Oculus Rift använder sig utav digitala signaler är det fördelaktigt att använda Wi-Fi eller Bluetooth då de också använder sig utav digitala signaler.

Skyzone och liknande FPV-glasögon använder radiosignaler för att ta emot bilddata. Nackdelen med trådlös analog bildöverföring är att delar av en bildruta kan gå förlorad, vilket försämrar bildkvaliteten. Detta är inte lika märkbart om två kameror används. Med två kameror går det att låta vardera av dem skicka likadan video till varsitt öga. Med två videoströmmar från olika kameror kan de två bildrutorna kompensera varandra om den ena skulle sakna en bildruta.

3.7 Reglering av position

En kvadrokopter har sex frihetsgrader, tre rotationella och tre translationella, men endast fyra insignaler i form av sina rotorerna. Dynamiken för systemet är följaktligen högst icke-linjär, vilket gör att regulatorer som LQR (*Linear Quadratic Regulator*) och andra regulatorer för linjära system är opassande.

Klassisk PID-reglering är ett alternativ som inte ställer höga krav på modellering av systemet. Ett annat alternativ är IMC-reglering (*Internal Model Control*), där en modell implementeras för att förutse systemets beteende vid givna signaler. IMC-kontroll av drönare har visats kunna vara kraftfullare än PID-reglering, med snabb och följsam styrning [17].

3.8 Datakommunikation

För att etablera en datakommunikation mellan två enheter går det att använda Bluetooth. Bluetooth är en smidig lösning då enheterna kan anslutas direkt till varandra och de inte är begränsade till en lokal position, utan endast avståndet relativt varandra. En annan metod för att skapa en trådlös anslutning är att ansluta båda enheterna till ett lokalt nätverk av typen Wi-Fi. Enheterna blir då begränsade till nätverkets position och räckvidd. Wi-Fi är oftast bättre än Bluetooth när det kommer till räckvidd, fördröjning och att skicka över data [18, 19]. Wi-Fi kräver dock att enheterna är anslutna till ett nätverk vilket kan ge en del problem. Till exempel måste en ny anslutning göras varje gång enheterna byter nätverk vilket kan vara tidskrävande om enheten inte är ansluten till skärm, tangentbord och mus. Vissa nätverk kan även ge stora fördröjningar med tanke på belastning och säkerhetsprotokoll. Därför är det bättre att låta systemet handha sitt egna nätverk. Detta kan möjliggöras genom att låta en enhet agera som en server och den andra agera som en klient. För att göra enkorts datorn Raspberry Pi till en server går det att konfigurera om dess nätverk till att bli en Wi-Fi accesspunkt [20]. På så sätt går det att skapa ett mobilt Wi-Fi.

Kommunikation mellan de två enheterna, vare sig den görs genom Wi-Fi eller Bluetooth, sker genom att skicka över data med antingen TCP (*Transmission Control Protocol*) eller UDP (*User Datagram Protocol*) som båda är olika protokoll för nätverksanslutningar. UDP skickar data utan att verifiera att datan kommit fram, vilket gör att UDP kan skicka data snabbt. I ett system som är känsligt för fördröjningar

är UDP ett bra val. TCP garanterar istället att datan kommer fram. En rekommendation är att bygga ett system som både har UDP och TCP och låta signaler som är fördröjningskänsliga att gå över UDP.

3.9 Valda lösningar

För avståndsbestämning mellan drönare och användare krävs det en avståndsmätning som fungerar även om personen inte befinner sig rakt framför drönaren. Detta innebär att avståndet med fördel kan mätas i ett vinkelspann, vilket är realiserbart med ultraljud eller vissa typer av laser. Då dessa typer av laser är mycket dyrare än ultraljudsmoduler väljes ultraljud för mätning av avstånd mellan drönare och användare. Då barometer redan finns på drönarens flygdator används detta för avståndsmätning från drönare till marken.

Bildigenkänning så som glyph tracking är möjlig för centrering av kameran, men ska det användas för att bestämma vinkeln som drönaren har mot användaren blir det komplext. Det ställer också höga krav på den kamera som används som upptagningsenhet då den ska ge tillförlitlig positioneringsdata till systemet. Trilateration med ultraljudsmoduler anses vara en enklare lösning för att bestämma vinkeln, och är också lätt att inkorporera då ultraljudsmoduler redan valts för avståndsbestämning. Dubbel spegelvänd trilateration för bestämning av orienteringen löper risk för att ultraljudspulserna stör ut varandra. För systemet väljes trilateration med ultraljud för vinkelbestämning, kompletterat med kompasser på användare och drönare för orientering, då kompasser redan finns inbyggt i videoheadset och i drönarens flygdator.

Då modellering av dynamiken i systemet anses komplex, så väljs regleringen av styrsignalerna till att skötas av en regulator utan framkoppling. Då resurser med kunskap om PID-regulatorer finns inom projektgruppen, väljes detta som regulatortyp för systemet. Dimensionering av regulatorn görs inte heller genom modellering på grund av samma anledning och görs därmed utifrån tester på den färdiga produkten.

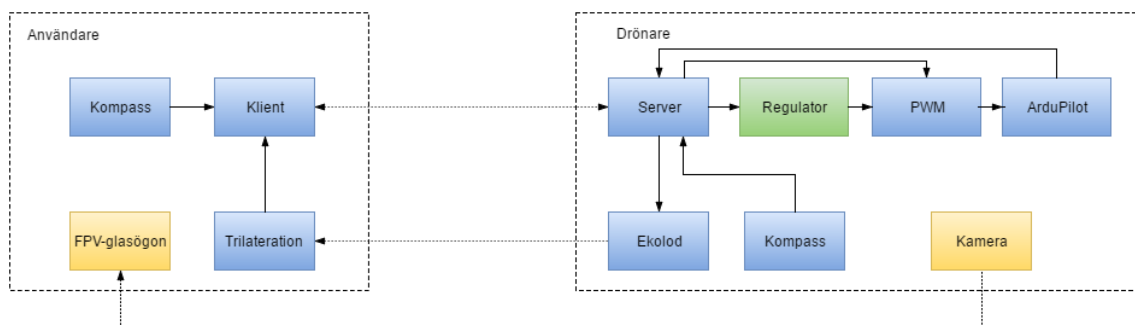
Då Wi-Fi är snabbt och flexibelt, med möjligheten att låta systemet handha sitt eget nätverk, väljes detta som lösning för kommunikationen av positioneringsdata. Då fördröjning mellan upptagning och uppspelning värderas högre än bildkvalitet väljes FPV-glasögon och tillhörande analog kameraupptagning, med enkel direkt radioöverföring av datan. Upptagning med två kameror förkastas då det ställer högre krav på överföring och hårdvara, och anses heller inte vara nödvändigt för användarupplevelsen.

4

Resultat

Kapitlet inleds med en kort överblick över hur hela systemet kommunicerar. Därefter presenteras det i varje delkapitel en djupgående beskrivning för respektive delsystem, samt vilka tekniska lösningar som har valts ut.

4.1 Systemdesign



Figur 4.1: Överblick på kommunikationen mellan delsystem. Pilarna visar i vilken riktning som kommunikation skickas

En överblick av systemet kan ses i figur 4.1. När systemet är aktivt lyssnar trilaterationssystem på ultraljudssignaler som skickas från drönarens ekolod. Trilaterationssystemet skickar data till användarens klient (Raspberry Pi) som beräknar positioneringsdata. Klienten skickar positioneringsdatan samt användarens kompassriktning till servern (drönarens Raspberry Pi) via en Wi-Fi anslutning.

På serversidan hämtas drönarens kompassriktning från kompassmodulen och drönarens aktuella höjd hämtas från ArduPilots barometer. Kombinerat med datan som skickas från klienten kan drönaren beräkna sin position och orientering. Värdena som beräknats hanteras av ett regulatorprogram på servern som skickar styr signaler till PWM-modulen för signalomvandling. PWM-modulen kan även få direkta styr signaler från servern, så som signaler för en uppstartssekvens. Med styr signaler från PWM-modulen kan ArduPiloten styra drönaren. FPV-glasögonen samt kameran kan ses som ett separat system där kameran skickar analog videoström till FPV-glasögonen. Hårdvaruutformning finns mer detaljerat beskriven i Appendix A.

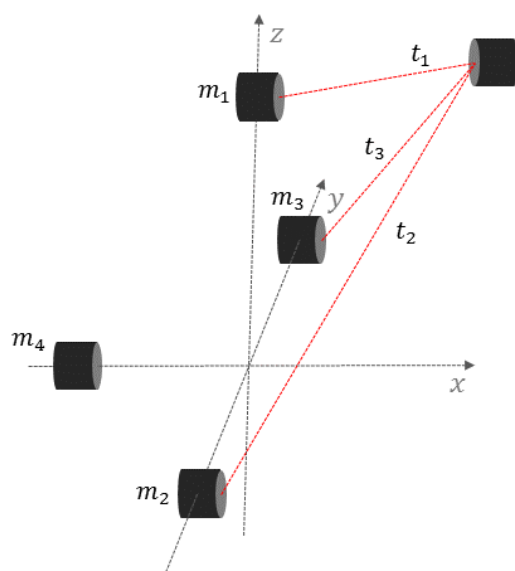
4.2 Positioneringsdata

Positioneringssystemet ska erhålla tre parametrar i sfäriska koordinater för drönaren: dess radiella avstånd (r), dess latitudvinkel (θ) och dess longitudvinkel (ϕ). Koordinaterna är relativa användaren; origo är placerat vid användarens huvud och koordinataxlarna följer användarens huvudrörelser.

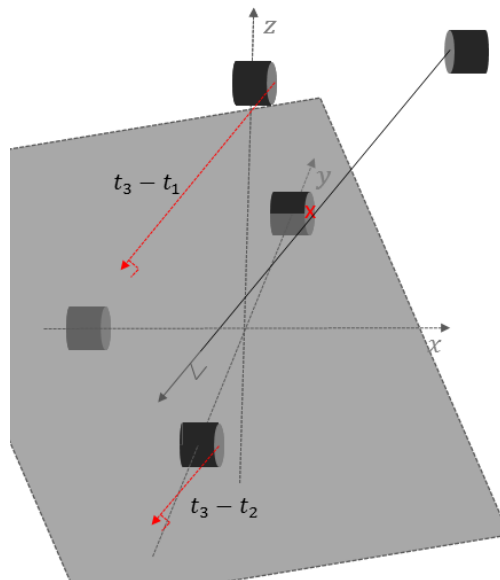
På grund av svårigheter med att synkronisera tidsloggning över två trådlöst sammankopplade system används inte ett fullständigt trilaterationssystem. Systemet kan endast hantera tidsloggning inom de isolerade systemen (användare och drönare). Därmed kan det endast finna vinklar på en riktningsvektor via trilateration och måste finna radiellt avstånd på annan väg.

Det radiella avståndet r erhålles via ultraljudsekolod. Raspberry Pi'n på drönaren är utrustad med en ekolodsmul som genererar en ultraljudspuls på $40kHz$ som sedan studsar på användaren och känns åter igen av modulen. Tiden för pulspropagering samt reflektion loggas i Raspberry Pi'n och avståndet räknas ut med hjälp av vetenskapen om ljudets hastighet i luft.

Ultraljudspulsen som skickas av ekolodet uppskattas till en plan våg. Därmed kan vågfronten ses som ett plan i R^3 vars normalvektor pekar i ljudets propageringsriktning. Fyra mikrofoner på position \hat{m}_1 , \hat{m}_2 , \hat{m}_3 respektive \hat{m}_4 registrerar en puls vid tiderna t_1 , t_2 , t_3 respektive t_4 så att $t_1 < t_2 < t_3 < t_4$. Se Figur 4.2. Vid $t = t_3$ har pulsens plan passerat \hat{m}_1 samt \hat{m}_2 och skär \hat{m}_3 . Se Figur 4.3.



Figur 4.2: Mikrofonuppställning med ljudkälla



Figur 4.3: Vågfront approximerad som plan

Projiceras \hat{m}_1 samt \hat{m}_2 till planet erhåller man två nya punkter \hat{n}_1 och \hat{n}_2 . Dessa två punkter, tillsammans med \hat{m}_3 , spänner planet för pulsen. Vektorerna $\hat{m}_1 - \hat{n}_1$ och

$\hat{m}_2 - \hat{n}_2$ är båda parallella med normalen till planet samt har magnitud $(t_3 - t_1)V_s$ respektive $(t_3 - t_2)V_s$, där V_s är ljudets hastighet i luft. Detta innebär att \hat{m}_3 , \hat{m}_1 och \hat{n}_1 skapar en rätvinklig triangel, så väl som \hat{m}_3 , \hat{m}_2 och \hat{n}_2 samt att $(\hat{m}_1 - \hat{n}_1) \times (\hat{m}_2 - \hat{n}_2) = 0$. Med detta kan följande ekvationer ställas upp:

$$(x_{m3} - x_{n1})^2 + (y_{m3} - y_{n1})^2 + (z_{m3} - z_{n1})^2 = |\hat{m}_3 - \hat{m}_1|^2 - ((t_3 - t_1)V_s)^2 \quad (4.1)$$

$$(x_{m1} - x_{n1})^2 + (y_{m1} - y_{n1})^2 + (z_{m1} - z_{n1})^2 = ((t_3 - t_1)V_s)^2 \quad (4.2)$$

$$(x_{m3} - x_{n2})^2 + (y_{m3} - y_{n2})^2 + (z_{m3} - z_{n2})^2 = |\hat{m}_3 - \hat{m}_2|^2 - ((t_3 - t_2)V_s)^2 \quad (4.3)$$

$$(x_{m2} - x_{n2})^2 + (y_{m2} - y_{n2})^2 + (z_{m2} - z_{n2})^2 = ((t_3 - t_2)V_s)^2 \quad (4.4)$$

$$(y_{n1} - y_{m1})(z_{n2} - z_{m2}) - (y_{n2} - y_{m2})(z_{n1} - z_{m1}) = 0 \quad (4.5)$$

$$(x_{n2} - x_{m2})(z_{n1} - z_{m1}) - (x_{n1} - x_{m1})(z_{n2} - z_{m2}) = 0 \quad (4.6)$$

$$(x_{n1} - x_{m1})(y_{n2} - y_{m2}) - (x_{n2} - x_{m2})(y_{n1} - y_{m1}) = 0 \quad (4.7)$$

Genom numerisk analys kan ekvationsystemet lösas för variablerna x_{n1} , y_{n1} , z_{n1} , x_{n2} , y_{n2} samt z_{n2} . θ och ϕ finnes enligt:

$$n\hat{o}\hat{r}m = (\hat{m}_3 - \hat{n}_1) \times (\hat{m}_3 - \hat{n}_2) \quad (4.8)$$

$$\theta = \cos^{-1}(z_{norm}) \quad (4.9)$$

$$\phi = \tan^{-1}\left(\frac{y_{norm}}{x_{norm}}\right) \quad (4.10)$$

Värt att notera är att ovanstående är lösningen för det specifika fallet $t_1 < t_2 < t_3 < t_4$. För de övriga permutationerna används samma grundkoncept och ekvationer, dock med andra mikrofonpositioner och tidsskillnader. Detta gör att systemet kan identifiera en puls från samtliga vinklar i rummet, och fullkomligt sakna spegelvända lösningar som uppkommer i system med endast tre mikrofoner. Exempel på matlabsript för en permutation finnes i Appendix B.

Mikrofonerna och dess förstärkarkretsar är specifikt utvecklade och designade för applikationen. Förstärkarkretsen bygger på operationsförstärkare, passiva frekvensfilter samt en digitaliserande triggkrets. De ger en digital utsignal, som kan tolkas av

Raspberry Pi, när de detekterar ljudvågor på 40 kHz . Kretschema och kretskortsdesign finns i Appendix C.

Ekvationssystemet är stort och komplicerat. Det är, förutom i vissa specialfall, omöjligt att lösa analytiskt och behöver lösas numeriskt med linjära approximationer. Ekvationssystemet löses i Matlab med hjälp av kommandot `VPAsolve`. På moderna PC-processorer tar lösningen av ekvationssystemet cirka en halv sekund. Det tar avsevärt längre tid på en Raspberry Pi, då enkortsdatorn har en svagare processor samt att det hade behövts implementeras i Python [21]. Beräkningshastigheten står i direkt korrelation till drönarens 'reaktionstid'; den måste först lösa ekvationssystemet för att kunna identifiera sin position. Först efter beräkningen kan felet regleras och drönaren flytta på sig. Fördröjningar som denna kan också bidra till ett instabilt regler-system. För att mitigera detta samplas ett matlabskript; ekvationssystemet för samtliga permutationer löses för samtliga möjliga tidsskillnader i en upplösning som ligger över Raspberry Pi's samplingsfrekvens på dess GPIO-pins (*General-purpose input/output*). Detta görs på en kraftfull dator med hjälp av parallellisering som avsevärt reducerar beräkningstiden. Alla lösningar paras ihop med tillhörande tidsskillnader och sparas i en ".mat"-fil som sedan kan läsas av ett python-skript på en Raspberry Pi med kraftigt reducerad iterationstid.

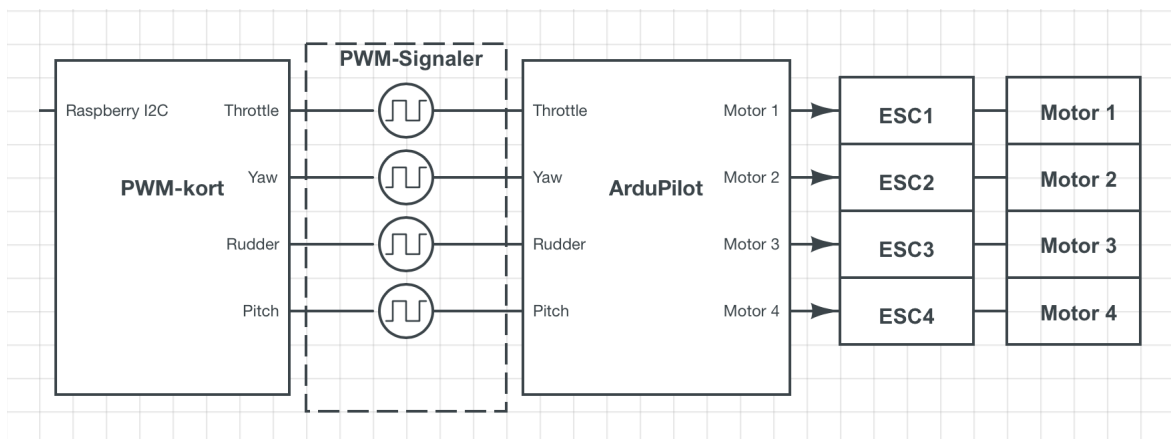
Höjddatan hämtas från ArduPilots inbyggda barometer. Beslutet kom naturligt eftersom många flygdatorer för projektets typ av drönare har en barometer inbyggd, bland annat ArduPilot [22]. Att använda en inbyggd komponent gav fördelen att undvika den extra vikt som skulle tillkommit utav en extern modul, men gav även fördelen att undvika fler program som ska köras och belasta den relativt lågpresterande enkortsdator som drönaren har att tillgå i beräkningskraft. Med barometer undkommer man även problem som inkorrekt höjdmätning på grund utav att drönaren har en lutning. Detta skulle kunna vara ett problem för ultraljuds- eller lasermätning, i och med att ultraljud och laser kräver en vinkelrät träffriktning mot markytan för att ge korrekt höjd.

4.3 Styrning och reglering

Styrsystemets utformning baseras på två olika huvudkomponenter, en Raspberry Pi och en ArduPilot. Raspberry Pi'n agerar som huvudenhet och är det kretskort som reglerar felet i avstånd, latitudvinkel, longitudvinkel och orientering givet från positioneringssystemet. Utifrån dessa fel skapas svarande styrsignaler (roll, yaw, pitch, throttle) för att återgå mot referenspositionen genom att använda ett PWM-kort och en mjukvarubaserad PID-regulator skriven i Python. PWM-kortet kopplas till en Raspberry Pi via I^2C som är ett seriellt protokoll för att koppla samman digitala system [23, 24]. Genom användning av I^2C -bussen kan felregleringen skötas enbart på Raspberry Pi'n. PID-regulatorn används för att undvika översväng, kvarstående fel och långsam styrning vilket möjliggör en stabil svarande styrsignal.

ArduPiloten analyserar den givna styrsignalen av regulatorn och beräknar med hjälp av informationen från gyrometern och accelerometern ut rotationshastigheten för att

efterfölja styrsignalen. Detta styrkort upprätthåller även höjdhållningen med hjälp av ett inbyggt flygläge som genom användning av den inbyggda barometern känner av och bibehåller sin höjd. ArduPiloten beräknar även de interna styrsignalerna, vilket är de justerings signaler som krävs för att systemet ska hålla sig stabilt. Utsignalerna givna av ArduPiloten skickas till en Hobbywing X-Rotor ESC, som fungerar som en sista regulator för motorrotationen och är även den komponent som delar ut ström till motorerna. En större överblick över systemets uppbyggnad ges i Figur 4.4.



Figur 4.4: Överblick av styrsystem

4.4 Videorepresentation

För videoupptagning används kameran AOMWAY 700TVL, då den har en låg fördröjning mellan videoupptagning och att datan skickas. Kameran är monterad på drönaren. Datan skickas analogt med radiosignaler på frekvensen 5,8GHz. Frekvensen valdes för att inte störa nätverkssignalerna (Wi-Fi). Frekvensen är även till fördel för att inte störa signaler från en eventuell handkontroller om drönaren ska styras manuellt. Som videouppspelningsenhet används Skyzone FPV-glasögon. Glasögonen tar emot radiosignalen som kameran skickar ut och spelar upp videon för användaren. Då FPV-glasögon används så spelas samma video upp för båda ögonen. Eftersom samma video spelas upp för båda ögonen så finns det inget 3D-djup i bilden.

5

Diskussion

Kapitlet kommer gå igenom de olika delsystemen som byggts samt utvärdera dess funktionalitet, för att sedan diskutera vilka andra lösningar som hade kunnat utveckla projektet vidare.

5.1 Videosystem

Videosystemet uppfyllde sin funktion med trådlös och direkt överföring med låg fördröjning. Negativt är att bildkvaliteten blev något bristfällig. Detta överensstämmer med prioriteringen där låg fördröjning står över bildkvalitet. Bättre bildkvalitet hade kunnat uppnås med en bättre och förmodligen dyrare kamera. För att skapa ett tredimensionellt djup i videon så behövs dels två kameror samt ett videoheadset som stöder stereoskopisk videouppspelning, vilket innebär att bilden behöver behandlas innan den kan visas upp. Detta hade bidragit till en bättre upplevelse men hade medfört betydligt mer arbete för projektet och hade troligtvis försämrat fördröjningen mellan kamera och uppspelningsenhet. Om mer arbete lades på att minimera fördröjningen i Wi-Fi-nätverket, eller om alternativa trådlösa digitala länkar med låg fördröjning utforskades, samt implementation av en kraftfullare dator vid användaren, hade användarupplevelsen förmodligen ökat. Att ha separata kanaler för olika typer av dataöverföring kan även ha bidragit till en låg fördröjning eftersom positioneringsdata som skickas över Wi-Fi, och videoström som sänder sin data över radiolänk, inte behöver dela på samma trafikutrymme.

5.2 Position

För att bestämma drönarens position valdes en kombination av olika teknologier: ultraljud för avståndsbestämning och trilateration i longitudinell och latitudinell riktning, kompasser för orienteringsdata, och barometer för altitudbestämning. Ultraljudet fungerade väl för avståndsbestämning, vilket den valda hårdvaran var designad för.

Trilaterationssystemet byggdes först med hjälp av ekolodsmodulerna. Dessa skickar ut en puls och registrerar ekot. Tanken var att modifiera dessa moduler så att de endast registrerar en puls. Om en puls då skickas från drönarens ekolod kan en uppsättning av modifierade moduler på så vis logga tidsskillnaderna som behövs för

trilaterationssystemet. Modulerna hade dock en time-out funktion, där de efter ett tag slutar lyssna efter ekot och registrerar en puls, även om en inte finns. Det visade sig väldigt svårt att få en puls att träffa modulerna under tiden då de var aktiva på grund av den trådlösa kommunikationen mellan användare och drönare.

I och med svårigheterna med att registrera en puls, utvärderades möjligheterna för att implementera en form av synkronisering för att få pulsen att träffa inom det korta intervall som ekolodsmodulens mottagare är aktiv. En metod som utvärderades var klocksynkronisering där tanken var att synkronisera operativsystemens klockor på de två enkortsdatorerna som kommunicerar. Om klockorna är synkroniserade går det att säkerställa att sändningen av pulsen sker samtidigt som systemet börjar lyssna efter den. På så vis kommer pulsen garanterat att registreras. En annan metod som utvärderades var radiotriggning där lösningen skulle varit att för att starta en puls, skulle systemet använda en extern radiosändare som skickar en signal från den ena enkortsdatorn till den andra när ekolodsmodulen är redo att ta emot en puls. Även i den här metoden finns en ökad kontroll av tidsstämpeln när puls skickas och avlyssnas. Tyvärr visade det sig vara för tidskrävande för att börja implementera i sent projektstadie och det var inte säkert att de modifierade ekolodsmodulerna skulle fungera stabilt ändå. Därför förkastades synkroniseringsmetoderna helt.

På grund av detta designades till sist egna moduler, utan time-out funktion, från grunden. Trots att detta går emot prioriteringen om att inte konstruera egna system ansågs det vara den bästa lösningen. Båda systemen hade dock problem att registrera pulserna korrekt på grund av mikrofonernas snäva avkänningsvinkel. Teknologier så som bildanalys förkastades då det bedömdes svårt att genomföra eller att det skulle kräva mycket arbete, men ultraljudet blev i praktiken också komplicerat. Med rätt förkunskap kan bildanalys förmodligen vara en mycket effektivare lösning.

Ekolod som metod för det radiella avståndet har brister i och med att reflektionsytan spelar roll för hur tidigt den reflekterade pulsen kan kännas igen. Har, till exempel, användaren en dunjacka på sig kommer ekoloden rapportera ett märkbart större avstånd än om användaren skulle ha en metallskiva på ryggen. Detta ansågs inte som ett större problem eftersom felet är repeterbart mellan ekolodsiterationer och det går genom ett användargränssnitt att kompensera för det.

Kompass- och accelerometerdatan var enkel att implementera. De moduler som köptes var direkt avsedda för användning med Raspberry Pi och fungerade felfritt. Eftersom att både ArduPilot och FPV-glasögonen har inbyggda kompasser och accelerometrar är kompassmodulerna egentligen överflödiga, och att extrahera datan ur den befintliga hårdvaran hade varit en, i slutändan, bättre lösning. Extraheringen av den datan ansågs dock för tidskrävande och svår, så redundans prioriterades. Implementationen för kompassdatan är relativt enkel i och med att den endast jämför de olika kompassvärdena och ser till att drönare och person pekar i samma riktning. För att åstadkomma en tredjepersonsvy där drönaren alltid fokuserar på personen, snarare än att vara vänd i samma riktning som personen, så går det att kombinera orienteringsdatan med drönarens longitudinella positioneringsdata. Detta sågs som

en möjlighet för att kunna definiera punkter för drönaren att befinna sig på som inte är direkt bakom personen.

Höjdbestämning sköttes av barometer internt i ArduPiloten. Detta var enkelt att implementera men gav inte god precision. Det uppstod också ett problem med att barometern inte tar hänsyn till vart marken är, vilket inte är ett problem om en person som ser marken styr drönaren. Ur säkerhetssynpunkt, och för att uppnå målet med ett robust system, så kunde ekolod som pekade nedåt implementeras och en symbios av den datan samt barometerdatan hade kunnat styra höjddreglingen.

Det största problemet för positionsbestämningen var svårigheter att få teknologierna att fungera som tänkt. Ofta var de anpassade för lite andra ändamål och det var utmanande att försöka modifiera dess funktion för att passa kraven. Hade insikten om svårigheterna angående ekolodsmodulerna kommit tidigare hade fokus kunnat läggas på att designa egna moduler från början vilket garanterat hade fört projektet mycket längre. Trots att det är en större tidsinvestering i tidigare skeden så lönar det sig i slutändan och hårdvaran blir bättre anpassad för ändamålet.

En ansats som gjordes tidigt i projektet var att förkasta GPS som metod för positionering. Den ansågs ha för låg precision, vilket är sant för absolut positionering. Det som inte togs i beaktning var att relativ position går att uppnå med en precision på centimeternivå eller under [25]. Detta är på grund av att de två relativa punkterna får nästan exakt samma absoluta positionsfel, som sedan kan släckas ut mot varandra. GPS hade då passat väldigt bra för den relativa positionen, och tillsammans med bildigenkänning bildat ett smidigt och stabilt system. Detta upptäcktes väldigt sent i arbetsprocessen och den befintliga lösningen prioriterades.

5.3 Regulator och Styrning

För att undvika översväng, kvarstående fel eller långsam styrning användes mjukvaruregulatorer för varje individuell parameter hos positioneringsdatan. Då dynamiken för en kvadrokopter är komplex, och faktorer som luftmotstånd och liknande extern påverkan är svårberäknad för en hemmabyggt drönare, ansågs modellering av systemet för utformning av regulatorer för tids- och resurskrävande. Till följd av valet att inte modellera systemet valdes dimensioneringen av regulatorn att göras utifrån prestandatester på det färdiga systemet.

Reglering av positionsfelet kunde aldrig riktigt utvärderas eftersom den kräver datainput från positioneringssystemet som inte blev färdigställt i tid. PID-regulatorer blev dock implementerade på drönarens Raspberry Pi, och de testades med fabrikerat positionsfel. Naturligt blev responsen av drönaren skild från ett verkligt fall i och med att felet, enligt regulatorn, aldrig kompenseras för eftersom det inte kan känna av drönarens positionsändring i verkligheten. Regulatorn betedde sig som förväntat i det experimentet, dock gav det inte så mycket information om hur den skulle bete sig i det riktiga systemet.

En genväg som togs var att använda hårdvara till Raspberry Pi för att generera fyrkantsvågor som sedan kunde skickas till ArduPiloten som rena styr signaler. Det är dock möjligt att injicera kod i ArduPiloten. Då är det också genomförbart att implementera regulatorerna direkt i ArduPiloten genom att endast skicka positionsfelet till den, givet att en sådan lösning lyckas är det också möjligt att använda ArduPilots interna kompass-, accelerometer- och gyroskopdata. På så vis sparas kostnad, vikt och strömförbrukning. Det ansågs dock svårt, osäkert och tidskrävande att införa denna implementation. Således prioriterades redundanta system, som med större säkerhet fungerar.

Att utveckla ett eget styrkort med hjälp av en Arduino sammankopplat med en accelerometer och gyroskop hade varit en möjlig lösning för styrningen. Denna möjlighet diskuterades och framstod som en lovande lösning i och med att all reglering och styrning hade kunnat hanteras på ett och samma kretskort. Det fanns även fler punkter där denna typ av implementation hade givit en fördel gentemot den nuvarande lösningen, vikt och plats hade minskat dramatiskt men även möjligheten att sända styr signaler via seriell kommunikation hade uppstått. Genom att använda seriell kommunikation för signalerna hade räckvidden av kommunikationslänken ökat markant, vilket även hade öppnat upp för nya användningsområden för produkten. Trots fördelarna med att skapa ett eget styrkort valdes denna typ av lösning bort då tiden för att utveckla ett sådant styrkort inte ansågs ligga inom projektets tidsram.

5.4 Ur samhällets synvinkel

Innan en produkt utvecklas och släpps på marknaden, kan det vara nödvändigt att se över eventuella problem som den kan tillföra. Om en produkt för tredjepersonsvy skulle utvecklas till ett såpass stadie att den har en möjlighet att etablera sig på marknaden, uppstår det en hel del frågor om hur produkten möter samhällets lagar samt hur den förhåller sig till de etiska aspekter som existerar. Sett till de lagar som existerar kan konceptet tredjepersonsvy tänkas vara accepterat i större utsträckning än fri drönarflygning med kamera och handkontroll, då det ställs högre krav på drönare som flygs utom synhåll än de som är inom synhåll. För att flyga en drönare utom synhåll krävs det, för att få tillstånd av Transportstyrelsen, att drönaren har ett typcertifikat [26]. En drönare av vikt under 1,5 kg och som avser flygning inom synhåll behöver inget typcertifikat för att få tillstånd, utan behöver endast uppfylla att motorerna maximalt utvecklar en kinetisk energi på 150 J [27]. Det finns även andra hinder för användarna; det ska alltid utses en pilot som är befälhavare för flygningen, innan flygning ska ett flyg- och säkerhetsområde upprättats och loggbok ska föras av flygningsdetaljer [28].

All flygning av drönare med monterad kamera, inom områden som allmänheten har tillträde till, kräver också ett tillstånd utav länsstyrelsen [29]. Att låta en kamera monteras på och filma från en drönare innebär att den omfattas av kameraövervakningslagen, eftersom den då inte är manövrerad direkt av användaren och kan tolkas som en övervakningskamera [29, 30].

Drönarflygning i allmänhet, och autonom flygning i synnerhet, är inte helt frånskiljt risker. De snabbt roterande motorerna kan åsamka stor skada vid kollision med omgivningen, och exempel finns där kollision med människor haft dödlig utgång [31]. Antalet dokumenterade olyckor är få, och studier som jämför olycksfall vid bemannad kontra obemannad flygning visar att obemannad flygning i sig inte är förknippat med olycksfall i högre grad. De olyckor som väl förekommer är vanligen förknippade med bristande kunskap eller ickefungerande utrustning, vilket inom hobbysektorn kan anses vara en stor faktor då ingen licens eller förkunskap krävs för att införskaffa en drönare [32].

Förutom de sociala och juridiska områdena, finns det ett tredje område att beakta om en produkt ska lanseras på marknaden och det är hur produkten befinner sig ur en miljösynpunkt. Som nämnt i rapportens avgränsningar ligger inte fokus på att ta fram en tilltalande produkt för konsumenter, utan att endast ta fram en prototyp. Således har lite eller ingen hänsyn tagits till projektets miljöpåverkan. Skall detta system bli en produkt ämnad för marknaden behöver ytterligare studier göras kring miljöaspekter.

5.5 Arbetsgång

Valet av metodik för val av implementation var till stort hinder för att bli klar med systemet. Att modifiera befintlig hårdvara visade sig otroligt tidsödande för arbetet med trilaterationssystemet då egenskaperna av den införskaffade hårdvaran gjorde att den, när modifierad, inte gav användbar funktionalitet. Metoden att utgå ifrån befintlig hårdvara gav motsatt effekt och gav i detta fallet en längre implementationsfas istället för en kortare.

Metodiken anses fortfarande lämplig, men ytterligare arbete krävs tidigt i arbetsprocessen för att undvika de eventuella problem som metodiken kan medföra. Hårdvara som skall modifieras bör vara väldokumenterad, även kring funktionalitet som anses irrelevant då den kan komma att bli relevant i ett senare skede. Om några som helst antaganden görs kring en hårdvaras funktionalitet bör dessa testas i praktik snarast för att styrka antagandena. Om detta arbetssätt eftersträvas kan förkastlig hårdvara snabbare identifieras och andra möjligheter tidigare utvärderas.

Eftersom tidigare projektgrupper inte lyckats med en fungerande implementation, och projektets svårigheter var tydliga, prioriterades tekniska lösningar som ansågs genomförbara med hjälp av resurserna inom projektgruppen. Att kunna börja jobba tidigt med en implementation som ansågs genomförbar prioriterades över att hitta den absolut bästa och mest skräddarsydda tekniska lösningen. Detta ledde till att mycket forskning övergavs när en kandidat funnits. Således råder det en stor möjlighet att systemutformningen inte är den absolut optimala.

6

Slutsats

Projektets syfte var att implementera tredjepersonsvy i verkligheten. Detta genomfördes med metodiken att lägga så lite tid på implementation som möjligt genom att använda befintliga lösningar och hårdvara, om nödvändigt med modifikation. Detta gjordes för att sedan ha tid till att finslipa systemets funktionalitet. Även om systemet inte blev klart och testat i sin helhet så anses systemet med de valda tekniska lösningarna vara ett realiserbart koncept, då majoriteten av delsystemen blev implementerade och var individuellt funktionella.

Användarupplevelsen förväntas bli bra då fördröjningen av videoöverföringen blev låg och styrsystemet svarade snabbt på insignaler. Då positioneringssystemet inte blev klart så gick det inte att analysera dess funktion. Det som kan nämnas om positioneringssystemet är att ställningen blev otymplig och försämrade potentiellt användarupplevelsen. Regler- och styrsystemet testades i den utsträckning det gick och fungerade enligt förväntningarna. Metodiken att använda befintlig hårdvara, med modifikation, visade sig ineffektiv för vissa delsystem då den modifierade hårdvaran inte gav önskad funktionalitet. Detta hade kunnat lindras genom att snabbare utvärdera funktionaliteten hos hårdvaran eller införskaffa hårdvara med mer omfattande dokumentation.

För att färdigställa prototypen enligt projektets frågeställning krävs fortsatt arbete. En bättre användarupplevelse kan potentiellt erhållas genom att använda GPS samt bildigenkänning för att undvika fysiska begränsningar hos det befintliga positioneringssystemet. Det går även att undersöka om det går att använda videoupptagning med högre bildkvalitet än den som nuvarande kamera ger. Utöver detta så kan drönaren förbättras genom att göra den uppmärksam på sin omgivning eller ge användaren möjlighet att ta kontroll över drönaren med hjälp av en handkontroll.

Referenser

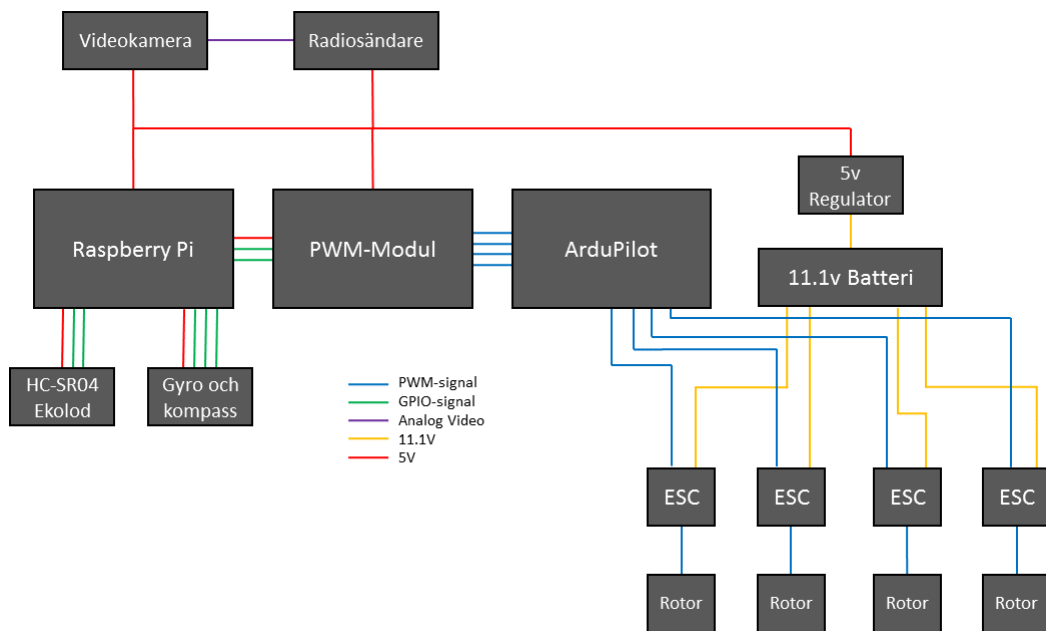
- [1] Allander J, Garpetun A, Hahlin A, Hultén JA, Karlsson J, Käll A. Tredjepersonsvy i verklig miljö Videoströmning från autonom luftfarkost Kandidatarbete inom Data-och Informationsteknik. 2015;.
- [2] Aronsson A, Gashi A, Karlsson B, Olausson T, Patricks M, Wallström A. Autonom kortdistansnavigation i öppen miljö Utveckling av kameraföljningssystem med drönare och Oculus rift för tredjepersonsvy i realtid. 2015;.
- [3] The Story Behind Bluetooth Technology | Bluetooth Technology Website;. Available from: <https://www.bluetooth.com/what-is-bluetooth-technology/bluetooth>.
- [4] Zhu X, Girod B. Video Streaming over Wireless Networks. In: Proc. European Signal Processing Conference (EUSIPCO'07). Eusipco; 2007. p. 1462–1466.
- [5] Rift - Next generation virtual reality; 2016. Available from: <https://www.oculus.com/en-us/rift/>.
- [6] Seminar A. An Introduction to Virtual Reality. Virtual Reality. 2002;p. 1–19. Available from: http://www.fkm.utm.my/~kasim/eng{}_design/vr/Lecture11a-VR0verview.pdf.
- [7] Newtons 3:e lag – Naturvetenskap.org;. Available from: <https://www.naturvetenskap.org/fysik/gymnasiefysik/kraft/newtons-3e-lag/>.
- [8] Martin P, Salaün E. The true role of accelerometer feedback in quadrotor control. In: Proceedings - IEEE International Conference on Robotics and Automation; 2010. p. 1623–1629.
- [9] Jordan D. Pulse-width Modulation - To PWM, or not to PWM, That is the Question [Internet]; 2013. Available from: <https://learn.sparkfun.com/tutorials/pulse-width-modulation>.
- [10] Electronic Speed Control (ESC) | Information & Guide to Electronic Speed Controllers System [Internet]; 2014. Available from: <http://www.rcmodelswiz.co.uk/rc-guides/electric-rc-models-guide/electronic-speed-controllers-esc/>.
- [11] Vilanova R. IMC based Robust PID design: Tuning guidelines and automatic tuning. Journal of Process Control. 2008;18(1):61–70.
- [12] Bouabdallah S, Noth A, Siegwart R, Siegwang R. PID vs LQ control techniques applied to an indoor micro quadrotor. 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat No04CH37566). 2004;3:2451–2456.
- [13] Ultrasonic Ranging Module HC-SR04;. Available from: <http://www.micropik.com/PDF/HCSR04.pdf>.

- [14] AR1000 Laser Distance Sensor | Laser Rangefinders, Laser Measurement Sensors;. Available from: <http://www.acuitylaser.com/products/item/ar1000-laser-distance-sensor>.
- [15] Bajaj R, Ranaweera SL, Agrawal DP. GPS: location-tracking technology. Computer. 2002;35(4):92–94.
- [16] Redlarski G, Pałkowski A, Ambroziak D. Optical glyphs based localization and identification system; 2013. Available from: <https://www.infona.pl/resource/bwmeta1.element.baztech-article-BSW1-0109-0032/content/partContents/2ca0f876-8090-339c-95b5-3e225d728262>.
- [17] Hernandez A, Copot C, De Keyser R, Vlas T, Nascu I. Identification and path following control of an AR.Drone quadrotor. 2013 17th International Conference on System Theory, Control and Computing, ICSTCC 2013; Joint Conference of SINTES 2013, SACCS 2013, SIMSIS 2013 - Proceedings. 2013;p. 583–588.
- [18] Davis TP. Bluetooth versus WiFi - A Comparison; 2004. Available from: <https://www.giac.org/paper/gsec/3941/bluetooth-wifi-a-comparison/106329>.
- [19] Dhawan S. Analogy of promising wireless technologies on different frequencies: Bluetooth, WiFi, and WiMAX. The 2nd International Conference on Wireless Broadband and Ultra Wideband Communications, AusWireless 2007. 2007;(AusWireless).
- [20] Setting up a Raspberry Pi as a Wi-Fi access point | Adafruit Learning System; 2013. Available from: <https://learn.adafruit.com/setting-up-a-raspberry-pi-as-a-wifi-access-point>.
- [21] Raspberry Pi 2 Model B;. Available from: <https://www.raspberrypi.org/products/raspberry-pi-2-model-b/>.
- [22] Ardupilot APM 2.5 and 2.6 Overview;. Available from: <http://ardupilot.org/copter/docs/common-apm25-and-26-overview.html>.
- [23] Ada L. Adafruit 16-Channel PWM/Servo HAT for Raspberry Pi. 2015;.
- [24] Mitchell R, Damouny N, Fenger C, Moelands A. I2C - An integrated serial bus architecture: principles and applications. 1985;p. 687–699.
- [25] Introduction to GPS (Global Positioning System); 1999. Available from: <http://www.utdallas.edu/~aiken/GPSCCLASS/GPSBasics.pdf>.
- [26] Tillstånd obemannade luftfartyg | Kategori 3 UAS;. Available from: <https://www.transportstyrelsen.se/sv/luftfart/Luftfartyg-och-luftvardighet/Obemannade-luftfartyg-UAS/Sok-tillstand-for-UAS/Kategori-3/>.
- [27] Tillstånd obemmanade luftfartyg | Kategori 1B UAS;. Available from: <https://www.transportstyrelsen.se/sv/luftfart/Luftfartyg-och-luftvardighet/Obemannade-luftfartyg-UAS/Sok-tillstand-for-UAS/Kategori-1B/>.
- [28] Tillstånd obemannade luftfartyg | Specifika krav - kategori 1A;. Available from: <https://www.transportstyrelsen.se/sv/luftfart/Luftfartyg-och-luftvardighet/Obemannade-luftfartyg-UAS/Sok-tillstand-for-UAS/Kategori-1A/Fordjupad-information/>.

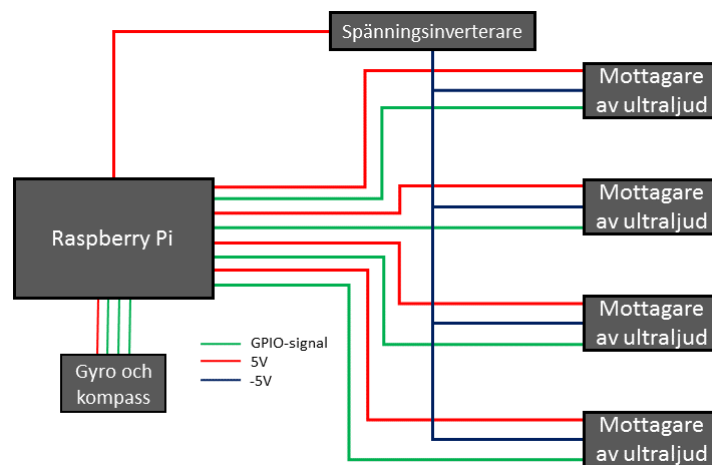
- [29] Kamarrätten: Drönare omfattas av kameraövervakningslagen;. Available from: <http://www.datainspektionen.se/press/nyheter/2015/kamarratten-ja-dronare-omfattas-av-kameraovervakningslagen/>.
- [30] Kameraövervakningslag (2013:460);. Available from: <http://www.riksdagen.se/sv/dokument-lagar/dokument/svensk-forfattningssamling/kameraovervakningslag-2013460{sfs-2013-460}>.
- [31] Man decapitates himself with his own remote-control helicopter | Daily Mail Online;. Available from: <http://www.dailymail.co.uk/news/article-2413231/Roman-Pirozek-Jr-Man-decapitates-remote-control-helicopter.html>.
- [32] Asim M, Ehsan N, Rafique K. Probable Causal Factors in UAV Accidents Based on Human Factor Analysis and Classification System; 2010. Available from: <http://www.icas.org/ICAS{ }ARCHIVE/ICAS2010/PAPERS/492.PDF>.

A

Appendix 1



Figur A.1: Blockdiagram av hårdvara för drönare



Figur A.2: Blockdiagram av hårdvara för användare

B

Appendix 2

```
Matlabscript för permutationen  $t_2 < t_3 < t_1 < t_4$ :  
% Simulated time of signal detection (s)  
t1 = 0.000003;  
t2 = 0.000001;  
t3 = 0.000002;  
t4 = 0.000004;  
  
% Speed of sound (mm/s)  
Vs = 340000;  
  
% Microphone placements in R3 (mm)  
M1 = [-80 0 0];  
M2 = [0 90 0];  
M3 = [0 -90 0];  
M4 = [0 0 180];  
  
% Vector calculations  
M13 = M1-M3;  
M12 = M1-M2;  
  
% Create new points in plane of sound wave front  
% as symbolic vectors  
syms x2 y2 z2;  
syms x3 y3 z3;  
n2 = [x2 y2 z2];  
n3 = [x3 y3 z3];  
  
% Cross product of 'sound delay' vectors  
cr = cross((M3-n3),(M2-n2));  
  
% Pythagoras theorem over triangle of pivotpoint,  
% one new point and one old point  
S1 = eq((M1(1)-n3(1))^2 + (M1(2)-n3(2))^2 + (M1(3)-n3(3))^2,  
(norm(M13)^2)-(((t1-t3)*Vs)^2));  
  
% Length of vector between new and old point  
% is speed of sound times delay  
S2 = eq((M3(1)-n3(1))^2 + (M3(2)-n3(2))^2 + (M3(3)-n3(3))^2,  
(t1-t3)*Vs)^2);
```

B. Appendix 2

```
% Pythagoras theorem over triangle of pivotpoint ,
% one new point and one old point
S3 = eq((M1(1)-n2(1))^2 + (M1(2)-n2(2))^2 + (M1(3)-n2(3))^2 ,
(norm(M12)^2)-(((t1-t2)*Vs)^2));

% Length of vector between new and old point
% is speed of sound times delay
S4 = eq((M2(1)-n2(1))^2 + (M2(2)-n2(2))^2 + (M2(3)-n2(3))^2 ,
((t1-t2)*Vs)^2);

% Sound waves are parallel
S5 = eq(cr(1),0);
S6 = eq(cr(2),0);
S7 = eq(cr(3),0);

% Solve equation system by linear approximation
S = vpasolve([S1,S2,S3,S4,S5,S6,S7]);

% Overwrite symbolic results
x3 = S.x3;
y3 = S.y3;
z3 = S.z3;
x2 = S.x2;
y2 = S.y2;
z2 = S.z2;

% Select correct solution set
if x3(1) <= M3(1) && x2(1) <= M2(1)
    && z3(1) >= M3(3) && z2(1) >= M2(3)
        x3 = S.x3(1);
        y3 = S.y3(1);
        z3 = S.z3(1);
        x2 = S.x2(1);
        y2 = S.y2(1);
        z2 = S.z2(1);
elseif x3(2) <= M3(1) && x2(2) <= M2(1)
    && z3(2) >= M3(3) && z2(2) >= M2(3)
        x3 = S.x3(2);
        y3 = S.y3(2);
        z3 = S.z3(2);
        x2 = S.x2(2);
        y2 = S.y2(2);
        z2 = S.z2(2);
elseif x3(3) <= M3(1) && x2(3) <= M2(1)
    && z3(3) >= M3(3) && z2(3) >= M2(3)
        x3 = S.x3(3);
        y3 = S.y3(3);
        z3 = S.z3(3);
```

```

        x2 = S.x2(3);
        y2 = S.y2(3);
        z2 = S.z2(3);
    else
        x3 = S.x3(4);
        y3 = S.y3(4);
        z3 = S.z3(4);
        x2 = S.x2(4);
        y2 = S.y2(4);
        z2 = S.z2(4);
    end

% Plot microphone placements in R3
scatter3(M1(1),M1(2),M1(3),'m')
axis equal
hold on
scatter3(M2(1),M2(2),M2(3),'r')
scatter3(M3(1),M3(2),M3(3),'r')
scatter3(M4(1),M4(2),M4(3),'r')

% Plot new points in sound plane
scatter3(x2,y2,z2,'g')
scatter3(x3,y3,z3,'g')

% Calculate normal to sound plane
Point = cross([M1(1)-x2,M1(2)-y2,M1(3)-z2],
              [M1(1)-x3,M1(2)-y3,M1(3)-z3]);
Point = Point/norm(Point)

% Plot normal to sound plane, i.e. arrow in sound direction
quiver3(0,0,0,Point(1),Point(2),Point(3),30,
        'black','MaxHeadSize',1.1,'linewidth',1.2)

% Evaluate error due to linear approximation
n2 = [x2 y2 z2]
n3 = [x3 y3 z3]
CloseToZero = dot(n2-n3,[x2-M2(1),y2-M2(2),z2-M2(3)])

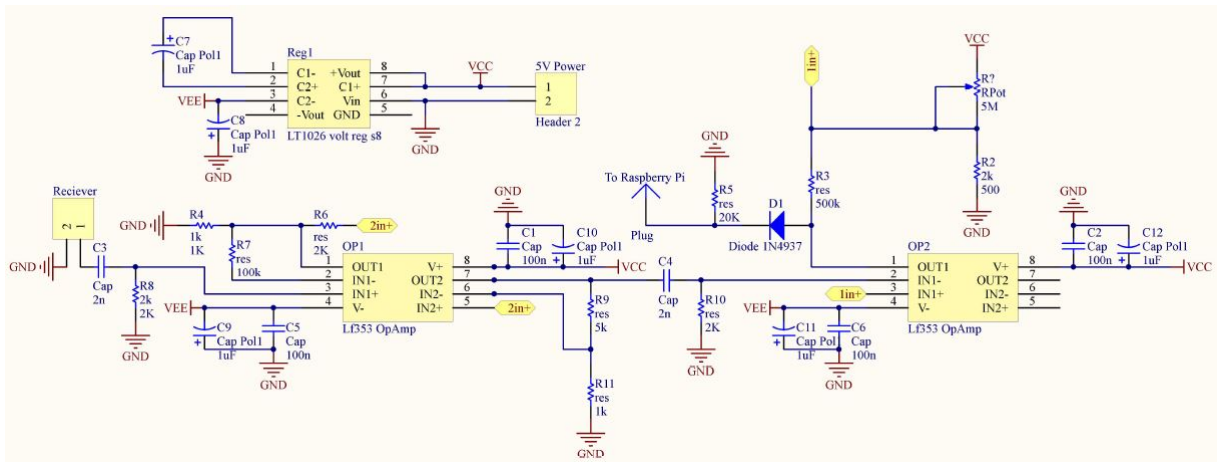
% Calculate theta and phi in spherical coordinates
% for the normal vector to the sound plane
theta = double(acos(Point(3))*360/pi)
phi = double((atan(Point(2)/Point(1)))*360/pi)

end

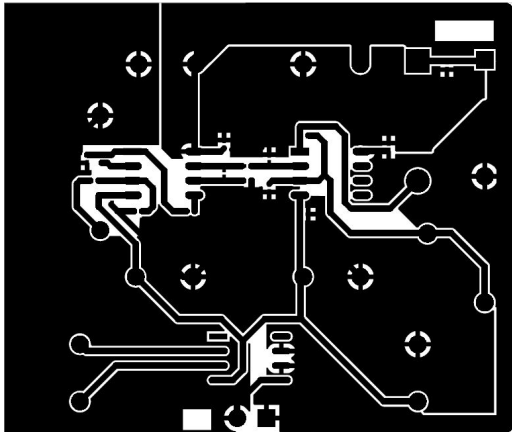
```


C

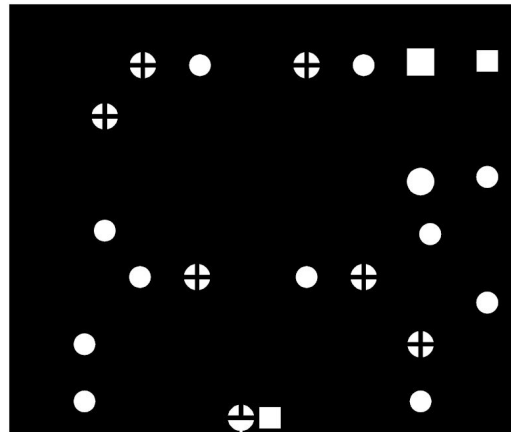
Appendix 3



Figur C.1: Kretsdesign



Figur C.2: Kretslayout framsida



Figur C.3: Kretslayout baksida