



CHALMERS

Självcentrerande kamera gimbal för träning av Artificiell Intelligens

Examensarbete inom högskoleingenjörsprogrammet Mekatronik

ANDREAS LJUNG MONTEIRO
DENISE SAMUELSSON FRIBERG

INSTITUTIONEN FÖR ELEKTROTEKNIK

CHALMERS TEKNISKA HÖGSKOLA

Göteborg, Sverige 2024

www.chalmers.se

BACHELOR THESIS 2024

Self-centering camera gimbal for training of Artificial Intelligence

ANDREAS LJUNG MONTEIRO
DENISE SAMUELSSON FRIBERG



CHALMERS

Department of Electrical Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2024

Självcentrerande kamera gimbal för träning av Artificiell Intelligens

Examensarbete inom högskoleingenjörsprogrammet Mekanik

Andreas Ljung Monteiro och Denise Samuelsson Friberg

© ANDREAS LJUNG MONTEIRO, 2024

© DENISE SAMUELSSON FRIBERG, 2024

Handledare: Christoffer Weber, Wiretronic AB

Examinator: Bertil Thomas, System- och reglerteknik

Examensarbete 2024

Institutionen för elektroteknik

Chalmers tekniska högskola

SE-412 96 Göteborg, Sverige

Telefon: +46 (0)31-772 10 00

Förord

Detta examensarbete har genomförts under år 2024 av två mekatronikstudenter vid institutionen för Elektroteknik på Chalmers tekniska högskola. Arbetet omfattar 15 högskolepoäng och är utfört i uppdrag för företaget Wiretronic AB.

Arbetet grundas av en begränsning som företaget hade med deras produktfotografering. Vi är tacksamma för möjligheten att arbeta med och utveckla en lösning till problemet. Det har varit lärorikt och utmanande att jobba med projektet och spännande att få vara en del i deras utveckling.

Vi vill även tacka Wiretronic som har försett oss med delar till gimbalen samt för stöttningen under projektets gång. Förhoppningsvis är detta början på ett projekt som de kan arbeta vidare på i framtiden och önskar de lycka till.

Andreas Ljung Monteiro & Denise Samuelsson Friberg, Göteborg, 2024

Sammanfattning

Fotografering av olika objekt till olika e-handelsproduktidor och webbshoppar används i stor utsträckning i dagens samhälle, inklusive 360° bilder. Företaget Wiretronic har en mindre studio för den typen av fotografering, däremot med begränsning till mindre objekt. Där fanns då en efterfrågan till att fotografera större objekt till samma kvalitet. Vilket leder till projektets grund, om det är möjligt att fotografera större objekt med likvärdigt resultat som i en mindre studio.

Utifrån efterfrågan utvärderades möjligheterna till en lösning på problemet. Gimbalen var det mest självklara för att fotografera och filma större objekt då de är stabila vid rörelse. En prototyp designades och testade fram en fungerande lösning. Utöver designen undersöktes även de komponenter som behövdes för systemet. De flesta komponenterna kunde sedan fås med hjälp av Wiretronic med en annan motor än den som var uttänkt från början.

Designen skapades med hjälp av CAD för att sedan enkelt skriva ut gimbalen med en 3D-skrivare. En systemarkitektur planerades med de komponenter som krävs tillsammans med ett styrsystem. När de fysiska delarna var färdiga gjordes en mjukvaruutveckling samt PID-reglering för självcentreringen.

Tester av gimbal-design i kombination med beräkningar gav ett lovande resultat med en gimbal som kan självcentrera större objekt för produktfotografering. Systemet har däremot mindre begränsningar i prestanda som går att utveckla vidare.

Nyckelord: Gimbal, CAD, programmering, systemarkitektur, Arduino, mikrokontroller, planetväxel

Abstract

Photography of various objects for various e-commerce product pages and online shops is widely used in today's society, including 360° images. The company Wiretronic has a smaller studio for that type of photography, however, with a limitation to only smaller objects. There was then a demand to photograph larger objects to the same quality. Which leads to the basis of the project, if it is possible to photograph larger objects with equivalent results as in the smaller studio.

Based on that demand, the possibilities for a solution to the problem was evaluated. A gimbal seemed to be the most obvious thing to photograph larger objects as they are stable when in motion. A prototype was then designed and tested for a solution that would fit the expectations. In addition to the design, the components used for the system were also examined. Most of the components could be obtained with the help of Wiretronic with a different engine than the one originally conceived.

The design was created with CAD to then print the gimbal with a 3D printer. A system architecture was planned with components that are required along with a control system. When the physical parts were finished, software development and PID regulation for the self-centering were done.

Tests of the gimbal design combined with calculations yielded a promising result with a gimbal capable of self-centering larger objects for product photography. However, the system has minor performance limitations that can be further developed.

The report is written in Swedish.

Keywords: Gimbal, CAD, programming, system architecture, Arduino, microcontroller, planetary gear

Innehållsförteckning

Terminologi/Förkortningar	1
1. INLEDNING	2
1.1 Bakgrund.....	2
1.2 Syfte.....	2
1.3 Avgränsningar.....	2
1.4 Precisering av frågeställning.....	2
2. TEORETISK/TEKNISK BAKGRUND.....	4
2.1 Hållfasthetsberäkningar för konstruktion med PLA.....	4
2.2 Gimbal princip.....	5
2.3 Hårdvarukomponenter	5
2.3.1 Teensy 3.6 mikrokontroller	6
2.3.2 ams-OSRAM AS5600	6
2.3.3 Texas Instruments DRV8313	6
2.3.4 TDK InvenSense MPU6050	6
2.3.5 BLDC motor	6
2.4 3D-skrivning.....	7
2.5 Mjukvara och PID kontroller.....	7
2.5.1 PlatformIO	7
2.5.2 Arduino	7
2.5.3 PID kontroller	7
2.5.3 I ² C kommunikation.....	8
3. METOD	9
3.1 Komponentval.....	9
3.2 Systemarkitektur.....	9
3.3 Algoritm utveckling.....	10
3.3.1 Algoritmens struktur och flöde.....	11
3.3.2 Signalbehandling och feedback-loop.....	11
3.4 Motorstyrning	11
3.4.1 Styrning med sinusoidala signaler	11
4. KONSTRUKTION	13
4.1 CAD Design.....	13
4.2 3D-skrivning & materialval.....	13
5. GENOMFÖRANDE.....	14
5.1 BLDC-motor.....	14
5.2 Prestandamätningar.....	14
5.2.1 Planetväxel.....	14
5.2.2 PID-reglering	16

5.3 Programmering	17
5.3.1 Kodstruktur och organisation	17
5.3.2 PID-reglering	18
5.3.3 Biblioteksanvändning	19
5.3.4 Sensordata.....	19
5.4 Konstruktion	19
5.4.1 Montering	20
5.5 Elektronik koppling	21
6. RESULTAT	23
6.1 PID kontroller	23
6.2 Planetväxel.....	23
6.3 Självcentrering	24
7. DISKUSSION OCH SLUTSATS	25
7.1 Sammanfattning av resultatet	25
7.2 Prestanda och effektivitet hos gimbalen	25
7.3 Tekniska begränsningar och utmaningar	25
7.4 Miljöpåverkan.....	26
7.5 Förslag till förbättringar.....	26
7.6 Slutsatser och framtida arbete.....	27
Referenser	28

Terminologi/Förkortningar

- PLA:** “Polylaktid” är en naturligt nedbrytbar, termoplastisk polyester som kan användas i en 3D-printer.
- PWM:** “Pulse width modulation”, pulsbreddsmodulering är en modulerande styrsignal för att få ut analoga resultat med digitala medel.
- I/O-pinnar:** Input/Output-pinnar kan programmeras för att antingen driva eller läsa av en yttre elektrisk nivå.
- PID:** PID-regulator (proportional, integral, derivative) är styrenheter som använder ett återkopplingssystem för att styra processvariabler.
- EMK:** Elektromotorisk spänning eller elektromotorisk kraft är ett mått på den elektriska spänningen mellan elektroder.
- IMU-sensor:** “Inertial measurement unit”, tröghetsmätenhet är en elektronisk enhet som mäter och anger en kropps specifika kraft samt vinkelhastighet genom en kombination av accelerometer och gyroskop.
- CAD:** “Computer aided design” syftar till digitalt baserad design och skapande av tekniska ritningar främst inom konstruktionsverksamhet och arkitektur.
- DC-motor:** Likströmsmotor, som är en av de vanligaste typerna av elektrisk motor som är baserad på likström som matas och omvandlas till rörelseenergi.
- BLDC-motor:** Borstlös likströmsmotor (brushless direct current) är en elektrisk motor.
- I²C:** “Inter-integrated circuit” är en seriell kommunikationsbuss som möjliggör dataöverföring mellan integrerade kretsar med hjälp av endast två ledare, en för klocksignalen (SCL) och en för data (SDA).
- SDA/SCL:** SCL (Serial Clock) är linjen som överför klocksignalen för att synkronisera dataöverföringen, medan SDA (Serial Data) är linjen som överför själva data mellan enheter i ett I²C-system.
- DSLR-kamera:** “Digital spegelreflexkamera”, en kamera som använder en spegelmekanism för att ge en exakt optisk sökarebild och möjliggör byte av objektiv, vilket ger hög bildkvalitet och flexibilitet i fotografering.

1. INLEDNING

Fotografering av objekt används till stor del i dagens samhälle för att visa upp produkter. På Wiretronic har de en 360° produktfotografering studio för att filma mindre objekt automatiskt. Denna teknologi möjliggör att objekt fotograferas med en stabil och skarp bild på hela objektet då det roterar på en platta. Kameran i studion, som är stationär, läser av QR-koder för att läsa plattans position och utifrån det roterar den för att få en helhetsbild av objektet. Detta används för att hitta rätt reservdelar till det inskannade objektet hos kunder.

Produktfotografering i 360° används främst på webbshoppar och e-handelsproduktidor. Det ger mer inlevelse för produkterna som visas och är till för att fånga konsumenternas intresse. Till skillnad från stillbenfotografering ger det mer engagemang och en bättre helhetsbild [1].

1.1 Bakgrund

Fotografering av stora objekt innebär en utmaning, eftersom det ofta krävs omfattande fysiskt arbete för att korrekt positionera kamerorna och säkerställa att objekten förblir centrerade i bildramen. För Wiretronic är begränsningen att större objekt inte får plats i deras studio och då inte går att filmas på samma sätt som mindre objekt. De vill kunna skapa en automatiserad process där man kan åstadkomma samma resultat som med studion utan att vara begränsad till storleken på produkten.

En gimbal är en konstruktion som gör det möjligt för ett objekt, som en kamera, att bibehålla stabilitet och balans, även när den som håller eller rör på gimbalen rör sig [2]. En gimbal erbjuder en lösning på problemet för företaget genom att automatisera processen för att centrera kameran på objektet, vilket avsevärt förenklar arbetsflödet för fotografering.

1.2 Syfte

Projektets syfte är att utveckla en självcentrerande gimbal anpassad för fotografering av stora objekt. Genom att automatiskt justera position och vinkel baserat på objektets storlek och form, syftar gimbalen till att förenkla processen för att skapa konsistenta och centrerade bilder. Dessa bilder ska sedan användas för att träna ett AI-system för bildigenkänning, vilket förbättrar systemets förmåga att noggrant identifiera och katalogisera produkter.

1.3 Avgränsningar

Projektet kommer att fokusera på utveckling och konstruktion av gimbalen, inklusive dess styrsystem och integration med fotograferingsutrustning. Testning och optimering av AI-systemet för bildigenkänning, samt analys av bildmaterialets påverkan på AI-systemets prestanda, ligger utanför projektets räckvidd.

1.4 Precisering av frågeställning

Arbetet grundades i att skapa ett hjälpmedel för fotografering av större objekt än de som redan är möjligt hos Wiretronic och då ansågs en gimbal vara en lösning som skulle undersökas. Därmed inkluderar projektet de följande centrala frågeställningarna:

- Hur kan en gimbal konstrueras för att automatiskt anpassa sig och centrera stora objekt för fotografering, i syfte att skapa enhetliga och högkvalitativa bilder?

- Vilka tekniska lösningar krävs för att gimbalen ska kunna hantera självpositionering och självstabilisering?
- På vilket sätt kan gimbalen's styrsystem utformas för att effektivt stabilisera kamerautrustning för kvalitativa bilder?

2. TEORETISK/TEKNISK BAKGRUND

Teoriavsnittet innefattar en teoretisk bakgrund för en djupare förståelse i de komponenter och beräkningar som använts inom projektet.

2.1 Hållfasthetsberäkningar för konstruktion med PLA

Vid konstruktion av gimbalen är det viktigt att säkerställa att alla delar är tillräckligt starka för att motstå de krafter som uppstår under drift, särskilt när tunga komponenter som en digital spegelreflexkamera (DSLR-kamera) ska hållas stabila. Polylaktid (PLA) är ett lättillgängligt och relativt starkt plastmaterial som används flitigt inom 3D-utskrift och presenteras som ett praktiskt val för prototyper och delar som utsätts för måttliga krafter. För att beräkna hållfastheten hos de olika komponenterna som skrivits ut i PLA-plast används flera grundläggande hållfasthetsformler. Formlerna används för att förstå hur materialet kommer att bete sig under olika belastningar och för att säkerställa att designen är tillräckligt robust.

För att analysera hur komponenterna böjer sig under belastning används formeln för tröghetsmomentet I [m^4] för ett cirkulärt tvärsnitt:

$$I = \frac{\pi \cdot (D^4 - d^4)}{64}$$

$$N_r = N_s + 2 \cdot N_p$$

Där D är ytterdiametern [m] och d är innerdiametern [m].

Tröghetsmomentet beskriver hur massan fördelas kring en axel och påverkar hur mycket en komponent kommer att böjas när den utsätts för en kraft. Ett högre tröghetsmoment innebär att komponenten är mer motståndskraftig mot böjning.

För att beräkna böjspänningen σ [MPa] i en komponent används böjmomentet M [Nm] och tröghetsmomentet I :

$$\sigma = \frac{M \cdot y}{I} \quad ; \quad y = \frac{D}{2}$$

vilket kan skrivas som:

$$M = \frac{\sigma \cdot I}{y}$$

Där M är böjmomentet och y är avståndet från det neutrala planet till yttersta fibern, vilket för ett cirkulärt tvärsnitt är halva diametern $D/2$.

PLA-materialet som används i konstruktionen har draghållfasthet på 30 ± 5 MPa [3]. För att inkludera en säkerhetsmarginal sätts den tillåtna draghållfastheten till 20 MPa vilket kan uttryckas som:

$$\sigma_t = 20 \cdot 10^6 \text{ Pa}$$

För att analysera krafter på en fastspänd balk används formeln för böjmomentet:

$$M = \frac{W \cdot L}{8} \rightarrow M = \frac{\sigma \cdot I}{y} \rightarrow \frac{W \cdot L}{8} = \frac{\sigma \cdot I}{y}$$

Där W är den applicerade kraften och L [m] är längden på balken.

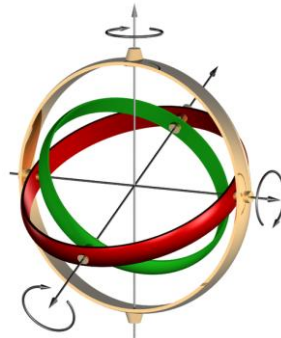
Därefter kan vi omforma ekvationen för att uttrycka W som:

$$W = \frac{8 \cdot \sigma \cdot I}{L \cdot y}$$

Dessa formler används för att bestämma de maximala krafter som komponenterna kan tåla utan att materialet deformeras eller går sönder. Genom att tillämpa dessa principer på gimbalen's design säkerställs att konstruktionen är både stark, stabil och kan hantera de krafter som uppstår under normal användning.

2.2 Gimbal princip

En gimbal är en mekanisk anordning som möjliggör stabilisering och rotation av ett objekt längs flera axlar, vanligtvis tre: yaw (gir), pitch (tippning) och roll (rullning). Genom att tillåta fri rotation kring dessa axlar kan en gimbal hålla ett objekt, såsom en kamera eller sensor, stabilt oavsett rörelser i den omgivande miljön eller plattformen.



Figur 1. Bilden visar de tre axlarna en gimbal kan rotera kring: gir, tippning och rullning. [4] Bilden används med tillstånd enligt Creative Commons-licensen CC BY-SA 4.0. Licensvillkoren finns tillgängliga på:

<https://creativecommons.org/licenses/by-sa/4.0/deed.sv>

Moderna gimbalsystem integrerar sensorer och motorer för aktiv stabilisering. Inertiala mätenheter (IMU), som kombinerar gyroskop och accelerometer, används för att mäta rotationshastigheter och accelerationer längs de tre axlarna. Denna sensordata bearbetas i realtid av en mikrokontroller som implementerar styralgoritmer, exempelvis PID-reglering. Algoritmerna beräknar nödvändiga korrigeringar och skickar signaler till borstlösa likströmsmotorer (BLDC-motorer), vilka justerar positionen av gimbalen's ramar för att motverka oönskade rörelser.

I fotografering av stora objekt, där exakt positionering är viktig, förenklar gimbalen arbetsflödet genom att automatisera kamerans centrering på objektet. Detta är särskilt användbart i maskininlärningsapplikationer där högkvalitativa och stabila dataset är avgörande.

2.3 Hårdvarukomponenter

Följande kapitel beskriver kortfattat de hårdvarukomponenter som använts under projektet för att sammanställa gimbalen.

2.3.1 Teensy 3.6 mikrokontroller

Teensy 3.6 är en mikrokontroller baserad på en 32-bitars ARM Cortex-M4-processor som körs vid 180 MHz, vilket gör den lämplig för komplexa applikationer som kräver snabb exekvering. Den har 34 digitala Input/Output-pinnar (I/O-pinnar) och 21 analoga ingångar, vilket ger flexibilitet för att ansluta olika sensorer och perifera enheter. Med inbyggt USB-stöd underlättas programmering och kommunikation. Dessutom erbjuder Teensy 3.6 flera I²C-bussar (upp till tre), vilket möjliggör kommunikation med ett flertal olika periferienheter [5].

2.3.2 ams-OSRAM AS5600

AS5600 är en magnetisk rotationspositionssensor med 12-bitars upplösning och analog eller PWM-utgång. Den använder en diametralt magnetiserad cylindrisk magnet för att mäta exakta vinklar. Komponenten är speciellt designad för potentiometerapplikationer vilket gör den robust mot externa magnetfält. Sensorn är enkel att programmera då den använder I²C-gränssnitt. Det intervall som är vanligast för utporten är 0 till 360 grader, men går även att programmera mindre intervall genom att identifiera en startposition och slutposition. För att ändra polariteten på utporten beroende på rotationsriktning ansluts en inport till jord [6].

2.3.3 Texas Instruments DRV8313

Texas Instruments DRV8313 är en integrerad motorstyrkrets designad för att kontrollera och driva trefasiga borstlösa DC-motorer. Den kan leverera upp till 2,5 A kontinuerlig ström per fas och har inbyggda skyddsfunktioner mot överspänning, överström och övertemperatur, vilket gör den särskilt användbar i applikationer som kräver hög precision och effektivitet [7].

2.3.4 TDK InvenSense MPU6050

För att få information om hur kameran rör sig och kunna motverka dessa rörelser för att hålla den stabil användes en accelerometer och ett gyroskop. Det specifika chippet som kombinerade detta var MPU6050 som är en integrerad sexaxlig rörelsesensor som kombinerar en 3-axlig gyroskop och en 3-axlig accelerometer. Kommunikationen sker via ett I²C-gränssnitt och har inbyggda digitala lågpasfilter för att minimera brus och förbättra mätningarnas noggrannhet. MPU6050 har också en inbyggd Digital Motion Processor (DMP) som kan användas för att avlasta beredningsarbetet från mikrokontrollern [8].

2.3.5 BLDC motor

Borstlösa likströmsmotorer (BLDC-motorer) fungerar genom att aktivera statorns elektromagnetiska spolar i en bestämd sekvens för att skapa en roterande magnetisk kraft. Denna kraft interagerar med rotorns permanenta magneter och genererar ett vridmoment som får rotorn att rotera. För att uppnå korrekt rotation måste spolarna aktiveras med exakta elektriska fasförskjutningar, vilket styrs av en specialiserad motorstyrkrets eller drivrutin.

Till skillnad från borstade likströmsmotorer, där kommuteringen sker mekaniskt med borstar och kommutatorer, sker kommuteringen i BLDC-motorer elektroniskt. Detta eliminerar friktion och slitage som uppstår med borstar, vilket resulterar i högre effektivitet, längre livslängd och tystare drift.

BLDC-motorer är effektivare än borstade likströmsmotorer under samma belastning och hastigheter. För att säkerställa att drivrutinen aktiverar spolarna i rätt sekvens, krävs kännedom om rotorns exakta position vid varje tidpunkt. Detta uppnås genom sluten styrning, ofta med hjälp av Hall-effektsensorer eller genom

att detektera mot-elektromotorisk spänning (EMK). Rotationshastigheten kan därefter styras noggrant genom att justera sekvensens hastighet. [9]

2.4 3D-skrivning

3D-skrivning är en additiv tillverkningsmetod där tredimensionella objekt skapas genom att successivt lägga lager av material ovanpå varandra baserat på en digital modell. Denna teknik möjliggör snabb och kostnadseffektiv produktion av komplexa former som annars skulle vara svåra eller omöjliga att tillverka med traditionella metoder. Genom att använda olika typer av material som plast, metall, och kompositmaterial, erbjuder 3D-skrivning stor flexibilitet i design och funktionalitet. Tekniken används inom många områden, från prototypstillverkning och medicinsk utrustning till flyg- och rymdindustrin, och har en viktig roll i att minska produktionskostnader samt öka innovationstakten. [10]

2.5 Mjukvara och PID kontroller

I detta avsnitt beskrivs mjukvaran samt reglerteknik som använts under projektet.

2.5.1 PlatformIO

PlatformIO (PIO) är ett plattformsoberoende-utvecklingsverktyg för inbyggda system som integrerar en utvecklingsmiljö (IDE), ett kraftfullt byggsystem, och verktyg för debugging, enhetstester och automatisk kodanalys. Till skillnad från traditionella verktyg, som ofta kräver specifika IDE och komplicerade inställningar för varje plattform, erbjuder PlatformIO en flexibel och användarvänlig lösning som stödjer ett brett spektrum av mikrokontroller och utvecklingsramverk. Med automatiserad hantering av bibliotek och beroenden samt enkel integration, underlättar PlatformIO utvecklingsprocessen och stödjer ett stort utbud av utvecklingskort och mikrokontroller arkitekturer [11].

2.5.2 Arduino

Arduino är en open-source plattform för elektronik som kombinerar enkel hårdvara och mjukvara för att möjliggöra snabb prototyputveckling. Plattformen använder Arduino-programmeringsspråket, baserat på Wiring, och Arduino Software (IDE), baserat på Processing, för att programmera och styra mikrokontrollern på kortet. Ursprungligen utvecklad vid Ivrea Interaction Design Institute för att underlätta prototypframställning för studenter utan elektronik- och programmeringsbakgrund, har Arduino vuxit till att bli en global plattform som används i allt från enkla projekt till komplexa vetenskapliga instrument.

Utöver sina egna utvecklingskort och Arduino IDE så erbjuder också Arduino miljön en plattform och ett starkt ekosystem av bibliotek och API för andra Arduino-kompatibelt utvecklingskort [12].

2.5.3 PID kontroller

En PID-kontroller (Proportionell-Integrerande-Deriverande kontroller) är en regleralgoritm som används för att justera och stabilisera dynamiska system. Den fungerar genom att kontinuerligt beräkna felet $e(t)$, vilket är skillnaden mellan det önskade värdet (börvärdet) och det faktiska mätvärdet (ärvärdet). Kontrollerutgången $u(t)$ bestäms genom summan av tre termer enligt PID-ekvationen:

$$u(t) = K_p \cdot e(t) + K_i \cdot \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt}$$

Där K_p är den proportionella förstärkningen, K_i är den integrerande förstärkningen och K_d den deriverande förstärkningen.

Den proportionella (P) delen tar det aktuella felet och multiplicerar det med en proportionell konstant (K_p), vilket ger en direkt proportionell respons mot felets storlek.

Den integrerande (I) delen summerar tidigare fel över tid, vilket hjälper till att eliminera kvarstående fel genom att summan av dessa fel ökar så länge de kvarstår.

Den deriverande (D) delen beräknar hastigheten av felets förändring genom derivatan $d(t)$ och multiplicerar den med K_d . Detta bidrar till att dämpa snabba förändringar och förbättrar systemets stabilitet.

Dessa tre delar, P, I och D, summeras sedan för att producera en styrsignal, ofta benämnd som $u(t)$. Denna styrsignal används för att styra ett ställdon, till exempel en motor, där motorns hastighet justeras beroende på PID-kontrollerns output; ett lågt värde resulterar i långsam körning medan ett högt värde resulterar i snabbare körning [13].

2.5.3 I²C kommunikation

I²C är ett seriellt datakommunikationsprotokoll och använder en seriell datalinje (SDA) och en seriell klockcykel linje (SCL). I²C protokollet utvecklades 1982 av Philips Semiconductor (dagens NXP Semiconductor) och används ofta för att koppla ihop periferienheter (t.ex. sensorer) med andra processorer som mikrokontrollers.

I²C-protokollet fungerar genom att en 'master'-enhet styr kommunikationen och en eller flera 'slave'-enheter svarar. Varje enhet på bussen har en unik adress, och 'master'-enheten använder dessa adresser för att rikta kommunikationen. Data skickas som paket som innehåller en startbit, en adress, en läsbara/skrivbara bitar, själva datan, och en stoppbit.

Protokollet stödjer olika hastighetslägen, från standard mode på 100 kbps till ultra-fast mode på 5 Mbps, vilket gör det flexibelt för olika applikationer. I²C möjliggör även anslutning av flera enheter på samma buss, där en 'master'-enhet kan kommunicera med flera 'slave'-enheter, som alla har unika adresser [14].

3. METOD

För att utveckla en kamera gimbal för fotografering av stora objekt, som effektivt kan samla in data, följdes en metodologisk ansats. Projektet inleddes med en behovsanalys och kravspecifikation som identifierade de kritiska funktionella och tekniska kraven för gimbalen. Kraven som ställdes var att gimbalen ska kunna användas likt den studio Wiretronic redan har och utifrån det kravet skapa ett koncept.

En del av behovsanalysen var att ta reda på i vilka vinklar som kameran skall kunna roteras, konstruktion och materialval för att kunna hantera kamerans vikt, motorstyrkan för att få nog med vridmoment för att kameran skall kunna roteras samt vilka komponenter som bör kombineras för att driva motorn och integrera en PID kontroller.

3.1 Komponentval

Teensy 3.6 valdes som den huvudsakliga mikrokontrollern på grund av dess tillgänglighet och höga klockhastighet. Systemet kräver en snabb mikrokontroller (180 MHz ARM Cortex-M4-processor) för att möjliggöra responsiv motorstyrning och hantering av flera uppgifter i realtid, såsom PID-kontroll och generering av PWM-signaler. Dessutom stöder Teensy 3.6 I²C-protokollet, vilket möjliggör snabb dataöverföring mellan sensorer (MPU6050, IMU och AS5600 magnetometrar) och mikrokontrollern, vilket minimerar latens och säkerställer korrekt motorstyrning.

DRV8313 BLDC-motordrivkretsar valdes för att kontrollera de borstlösa DC-motorerna (BLDC) på gimbalen's axlar. DRV8313 ger möjlighet till sinusoidal styrning, vilket möjliggör mjuk och exakt rörelse, vilket är viktigt för att minimera vibrationer och bibehålla en stabil kameravinkel. DRV8313 har ett spänningsintervall på 8–60 V vilket ger flexibilitet på vilka olika batterier och motorer som kan användas [7].

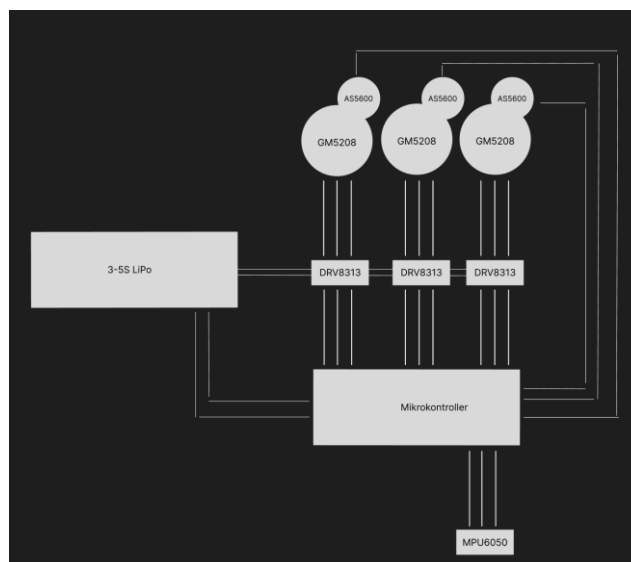
En MPU6050 IMU-sensor (Inertial Measurement Unit) valdes för dess förmåga att tillhandahålla både accelerometer- och gyroskopdata, vilket ger omfattande information om kamerans rörelsebeteende. Sensorn är också ett kostnadseffektivt val som är enkel att tillhandahålla då det finns möjlighet att köpa den med ett breakout-kort.

AS5600-magnetometern valdes eftersom den är en kostnadseffektiv och lättillgänglig komponent för att mäta rotationspositionen hos en BLDC-motor. Den finns tillgänglig som ett breakout-kort, vilket underlättar integrationen i systemet via I²C-kommunikation utan behov av extra komponenter.

Ett batteri med en spänning på 11,1 till 18,5 V valdes för att försörja hela systemet med ström. Detta spänningsintervall möjliggör användning av olika batterityper, vilket ger flexibilitet beroende på tillgänglighet. Dessutom är det ett lämpligt intervall för samtliga komponenter, då motordrivaren har ett rekommenderat spänningsintervall på 8–60 V [7] och den valda mikrokontrollern har en maximal inspanning på 21 V.

3.2 Systemarkitektur

Den föreslagna systemarkitekturen för gimbalen är uppbyggd kring en kombination av hårdvarukomponenter och styrsystem för att säkerställa stabilisering och självcentrering av kameran.



Figur 2. Översiktlig systemarkitektur för gimbalen.

Systemarkitekturen är utformad för att möjliggöra noggrann och dynamisk kontroll av kamerans position och rörelser i tre dimensioner genom att använda en integrerad kombination av sensorer, mikrokontroller, motorer och motorstyrkretsar. Varje av de tre BLDC-motorerna är utrustad med en AS5600-magnetometer som mäter motoraxelns rotationsvinkel och rotationshastighet i realtid. Dessa sensorer ger kontinuerlig feedback om motorernas position till systemet, vilket är avgörande för att säkerställa precisionsstyrning.

MPU6050 IMU-sensorn, placerad på kamerafästet, mäter kamerans lutning och position längs alla tre axlar och skickar denna data till Teensy 3.6 mikrokontrollern. All sensordata samlas in och behandlas av mikrokontrollern via I²C-kommunikationsbussen, där den genomgår signalbehandling för att filtrera bort brus och förbättra mätningarnas noggrannhet. Den filtrerade data från IMU-sensorn kombineras med positionsdata från magnetometrarna för att beräkna kamerans aktuella vinkel och rörelsetillstånd.

Mikrokontrollern kör PID-algoritmer för varje motoraxel separat. PID-algoritmen använder positions- och rörelsedata från sensorerna för att dynamiskt justera motorstyrningen och minimera avvikelser från den önskade kameravinkeln. Detta görs genom att skicka justerade PWM-signaler till de tre DRV8313 BLDC-motordrivkretsarna, som är anslutna till respektive BLDC-motor. Dessa motordrivare omvandlar PWM-signalerna till de nödvändiga elektriska strömmarna för att styra motorernas hastighet och riktning, vilket säkerställer att kameran hålls stabil och centrerad oavsett yttre rörelser eller störningar.

Denna integrerade systemarkitektur gör det möjligt att upprätthålla en exakt och stabil kameravinkel genom att kontinuerligt övervaka och justera motorernas rörelse i realtid baserat på sensorfeedback och kontrollalgoritmer.

3.3 Algoritmutveckling

Mjukvaran för mikrokontrollern Teensy 3.6 utvecklades med hjälp av PlatformIO i kombination med Arduino-kärnan inom Visual Studio Code. Användningen av PlatformIO möjliggjorde en flexibel och effektiv utvecklingsmiljö med automatiserad hantering av bibliotek och beroenden. Detta förenklade utvecklingsprocessen avsevärt och möjliggjorde snabbare testning och felsökning.

Koden är skriven i Arduino/C++ och strukturerad i modulära komponenter för att hantera funktioner som sensordatahantering, motorstyrning och algoritmkontroll. En central del av mjukvaran är en PID-kontroller som används för att justera och stabilisera gimbalen genom att kontinuerligt reglera motorernas rörelser baserat på sensordata.

3.3.1 Algoritmens struktur och flöde

Algoritmen är centrerad kring en feedback-loop som kontinuerligt samlar in data från sensorer (AS5600-magnetometrar och MPU6050 IMU), bearbetar dessa data och genererar styrsignaler för motorerna. I varje iteration hämtas data från IMU-sensorn via I²C-bussen, inklusive information om kamerans acceleration och rotationshastighet, samtidigt som vinkeldata samlas in från AS5600-magnetometrar för att mäta motoraxlarnas position.

Denna sensordata filtreras för att reducera brus och oönskade signalvariationer. Efter filtrering används den bearbetade datan i PID-beräkningarna för att bestämma lämpliga styrsignaler till motorerna.

3.3.2 Signalbehandling och feedback-loop

PID-kontrollern, implementerad i C++, justerar kontinuerligt motorernas hastighet och riktning genom att beräkna ett felvärde — skillnaden mellan den önskade och faktiska kamerans position. En realtids feedback-loop använder IMU-sensors data för att identifiera plötsliga rörelser eller vibrationer och justerar därefter PWM-signalerna som skickas till DRV8313-motordrivkretsarna för att kompensera för dessa rörelser.

Vid varje iteration av huvudloopen skickas korrigerade styrsignaler till motorerna, vilket säkerställer en snabb respons och smidig justering för att hålla kameran stabil och centrerad i bildramen.

3.4 Motorstyrning

En 3-fas BLDC-motor fungerar genom att tre faser (eller lindningar) styrs för att skapa ett roterande magnetfält, vilket i sin tur driver en permanent magnet i rotorn. Varje fas i motorn fungerar som en elektromagnet som attraherar eller repellerar den magnetiska rotorn. Genom en växlande sekvens av att slå på och av strömmen i dessa faser, roterar rotorn kontinuerligt.

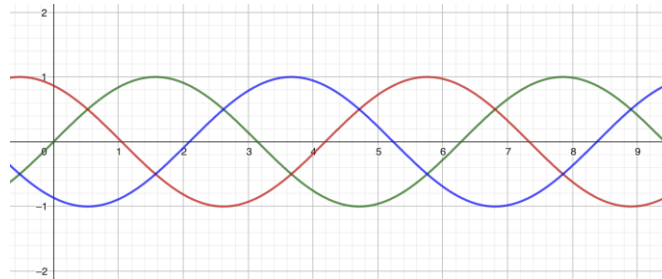
Processen fungerar så här:

1. **Initial aktivering:** Ström tillförs till två av de tre faserna, vilket skapar ett magnetfält som attraherar rotorn till en specifik position.
2. **Växling av faser:** När rotorn når denna position, ändras strömtillförseln så att strömmen byts till nästa par av faser, vilket driver rotorn vidare.
3. **Kontinuerlig rotation:** Genom att växla vilka faser som är aktiva, skapas ett roterande magnetfält som hela tiden tvingar rotorn att följa efter och därmed rotera kontinuerligt.

3.4.1 Styrning med sinusoidala signaler

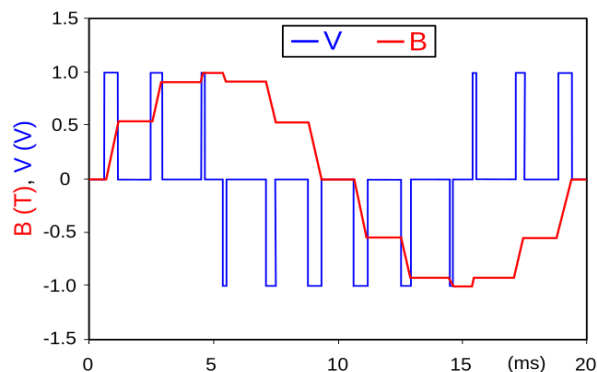
Vid sinusoidal styrning används sinusformade signaler för att styra strömmarna i de tre faserna. Varje fas får en sinusformad spänning som är förskjuten med 120 grader i förhållande till de andra faserna. Detta skapar ett jämnt roterande magnetfält, vilket ger en mjuk rotation av motorn.

1. **Sinusformade spänningar:** De tre faserna matas med sinusformade spänningar som är förskjutna 120 grader från varandra. Detta säkerställer att det magnetiska fältet som roterar runt rotorn är jämnt och konstant i styrka.
2. **Mjuk rotation:** Eftersom spänningarna är sinusformade, blir kraften som appliceras på rotorn mjukare, vilket resulterar i en mer stabil och effektiv rotation med mindre vibrationer och ljud.



Figur 3. Sinusoidal kontrollsignal för BLDC motor.

För att använda sinusoidal styrning med en mikrokontroller omvandlas de sinusformade signalerna till digitala PWM-sigaler. Genom pulsbreddsmodulering (PWM) kan dessa sinusvågor återskapas genom att variera pulsbredden, vilket styr mängden kraft som tillförs varje fas. Genom att justera pulsbredden kan den effektiva spänningen regleras, vilket i sin tur möjliggör kontroll över motorns hastighet och vridmoment.



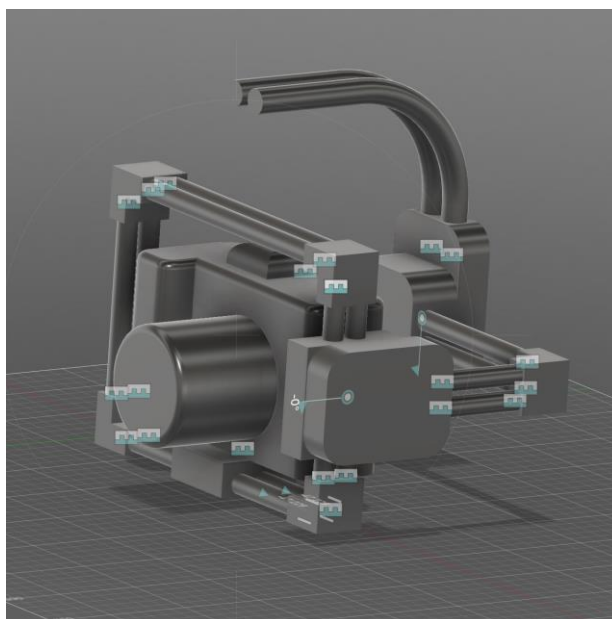
Figur 4. Exempel på hur PWM signaler konverteras till sinusoidala signaler där $V(V)$ är pulser och $B(T)$ är resultatet.[15] (Zureks, n.d.). Bilden används med tillstånd enligt Creative Commons-licensen CC BY-SA 3.0. Licensvillkoren finns tillgängliga på: <https://creativecommons.org/licenses/by-sa/3.0/>

4. KONSTRUKTION

I följande kapitel beskrivs konstruktionen och materialet av gimbalen för att uppnå det önskade resultatet. För att skapa konstruktionen har tillgängligheten av både program och material varit i stor vikt.

4.1 CAD Design

För att designa gimbalkonstruktionen användes CAD-programvaran Fusion 360. Valet av Fusion 360 motiverades av dess flexibilitet för att fungera på olika operativsystem som Windows och macOS, till skillnad från andra CAD-program som har ett begränsat antal av kompatibla operativsystem. Genom att använda CAD-modellering kan konstruktionen visualiseras och optimeras för att se till att alla komponenter integreras.



Figur 5. Gimbal-designen i Fusion360.

4.2 3D-skrivning & materialval

För att möjliggöra en kostnadseffektiv och snabb produktion av prototypen valdes 3D-skrivning som tillverkningsmetod för gimbalen's komponenter. Valet av 3D-utskrift motiverades av dess flexibilitet vid designiterationer och förmåga att snabbt producera komponenter utan behov av specialverktyg eller formar. Detta gör det möjligt att snabbt justera designen baserat på testresultat och feedback.

Som material för 3D-utskriften valdes PLA (polylaktid), en bioplast som är både hållbar och enkel att arbeta med. PLA valdes specifikt på grund av dess höga tillgänglighet, kostnadseffektivitet och relativt höga draghållfasthet (cirka 30 MPa) [3] och har en tillräcklig styvhet för att klara av de mekaniska krav som ställs på gimbalen under normal användning.

Materialvalet och dimensioneringen av komponenterna anpassades för att säkerställa att gimbalen kunde bära vikten av en DSLR-kamera och motstå de krafter och påfrestningar som uppstår under rörelse.

5. GENOMFÖRANDE

Detta kapitel beskriver genomförandet, vilket innefattar uppbyggnaden av gimbalen och mätningar som gjorts för att färdigställa systemet.

5.1 BLDC-motor

På grund av tidsbrist kunde den ursprungligen planerade motorn inte införskaffas, vilket resulterade i att en mindre och svagare BLDC-motor användes i stället. Trots att denna motor har ett lägre vridmoment än den avsedda, var den tillräcklig för att validera konceptet av gimbalen. Däremot påverkas prestandan av att motorn inte kunde leverera det planerade vridmomentet.

Den nya motorn var dock fortfarande en 3-fas BLDC-motor, vilket valdes för att bibehålla konstruktionens modularitet. Detta innebär att en starkare 3-fas BLDC-motor kan implementeras i framtiden utan att omfattande ändringar i konstruktionen behöver göras. Motorn hade en rekommenderad spänning på 6V - 8V [16], detta innebär att den valda motordrivaren blev kompatibel med den nya motorn, men att batteriet behövde ersättas med ett som hade lägre spänning.

5.2 Prestandamätningar

Vid de initiala testerna av motorn och dess förmåga att rotera gimbalkonstruktionen presterade motorn som förväntat när den endast behövde hantera konstruktionen. Men när en kamera monterades på gimbalen visade det sig att motorn inte hade tillräckligt med kraft för att effektivt rotera både konstruktionen och kameran. Detta visade ett behov av ökat vridmoment för att kunna hantera den extra belastningen.

För att kunna rotera konstruktionen med en monterad kamera behövdes motorns vridmoment ökas. Därför beslutades det att inkludera en planetväxel som en lösning för att uppnå detta. Planetväxeln valdes inte bara för sin förmåga att effektivt öka vridmomentet, utan också för sin kompakta storlek, vilket möjliggjorde en relativt smidig design. Dessutom erbjuder planetväxeln hög effektivitet och jämn kraftöverföring, vilket säkerställer en stabil och pålitlig drift utan att öka konstruktionens totala vikt och volym avsevärt.

5.2.1 Planetväxel

En planetväxel är en typ av växellåda som består av tre huvudsakliga komponenter: ett centralt solhjul (solväxel), ett eller flera planetväxlar som roterar runt solhjulet, och en ringväxel som omger planetväxlarna. Planetväxeln är känd för sin kompakta design och förmåga att överföra stora mängder vridmoment med hög effektivitet och stabilitet. Den används ofta i applikationer där utrymme är begränsat men där hög vridmomentöverföring krävs.

Den matematiska formeln för vridmomentutväxling i en planetväxel är:

$$\text{Vridmoment ut} = \text{vridmoment in} \cdot \text{utväxlingsförhållande}$$

Där utväxlingsförhållandet beräknas som förhållandet mellan antalet tänder på ringväxeln N_r och antalet tänder på solhjulet N_s :

$$\text{utväxlingsförhållande} = 1 + \frac{N_r}{N_s}$$

Dessutom gäller att:

$$N_r = N_s + 2 \cdot N_p$$

Detta innebär att genom att öka antalet tänder på ringväxeln relativt till solhjulet kan ett högre utväxlingsförhållande uppnås och därmed ett högre utgående vridmoment.

Det beräknades att en 3x utväxling skulle behövas för att öka vridmomentet, då det efterfrågade vridmomentet var cirka 1000 g och motorn initialt kunde generera endast cirka 350 g i vridmoment.

Med hjälp av ovanstående utväxlingsformel och målet att tredubbla vridmomentet kan förhållandet mellan solhjulet och ringväxeln beräknas:

$$3 = 1 + \frac{N_r}{N_s} \rightarrow 2 = \frac{N_r}{N_s}$$

Detta innebär att ringväxeln bör ha två gånger fler tänder än solhjulet. Antalet tänder för planetväxlarna kan sedan bestämmas genom formeln:

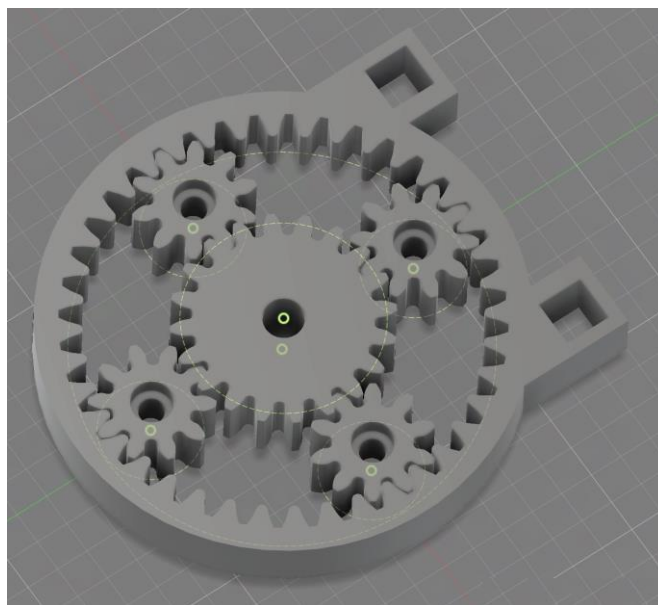
$$N_r = N_s + 2 \cdot N_p \rightarrow N_p = \frac{N_r - N_s}{2}$$

För att uppfylla samtliga villkor fastställdes måtten på planetväxlarna i enlighet med ovanstående beräkningar, vilket säkerställer rätt utväxling och därmed det nödvändiga vridmomentet. För att uppnå en 3x utveckling i vridmoment blir relationen mellan kugghjulen som följande:

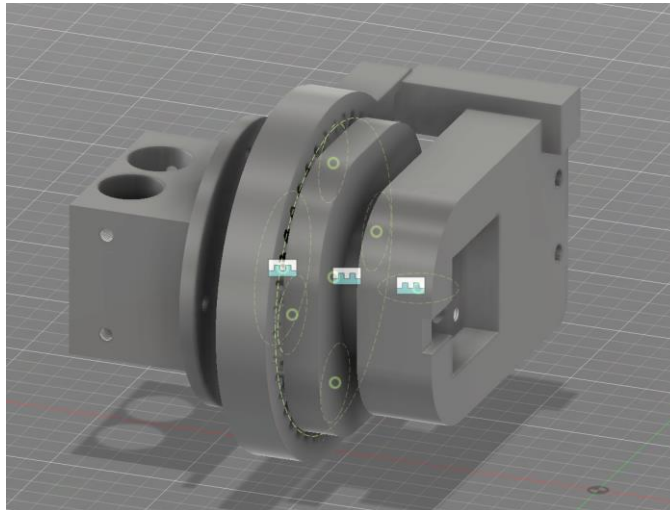
Planethjulet: 10 tänder

Solhjulet: 20 tänder

Ringhjulet: 40 tänder



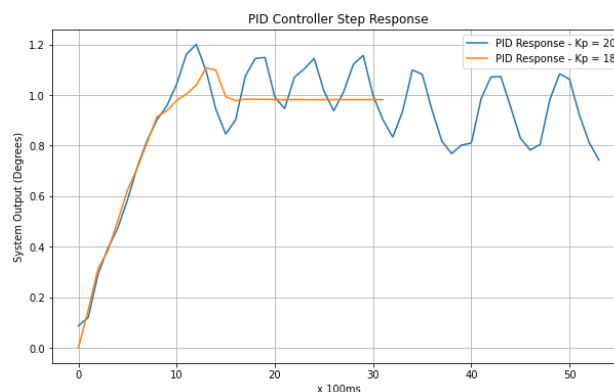
Figur 6. Insidan av planetväxel från Fusion360.



Figur 7. Planetväxel konstruktion designad i Fusion360.

5.2.2 PID-reglering

Med den uppgraderade planetväxeln som tillåter ett starkare vridmoment för att rotera kameran, kunde justeringen av PID-regulatorn påbörjas. För att justera PID-regulatorn användes Ziegler-Nichols-metoden. Denna metod kräver att systemet först sätts i självsvängning. Justeringen utfördes genom att sätta börvärdet till 0, manuellt justera gimbalen till en 90-gradersposition, och sedan låta gimbalen självcentrera tillbaka till 0 grader. Under denna process mättes positioneringsdata och hastighet. Vinkeldatan registrerades var 100 millisekund under självcentreringen.



Figur 8. Självcentrering med olika P-värden, med självstängning och utan självsvängning.

Vid ett P-värde på 20 observerades översvängningar med en svängningsperiod på cirka 0,5 sekunder. När P-värdet sänktes till 18 observerades fortfarande en översvängning, men systemet stabiliserades därefter.

För att justera PID-regulatorn enligt Ziegler-Nichols-metoden noterades att oscillationstiden var cirka 0,5 sekunder, vilket ger en kritisk period, $T_u = 0,7$ sekunder [17]. Detta ger en integraltid T_i enligt:

$$T_i = \frac{T_u}{2} = \frac{0,7}{2} = 0,35s$$

Metoden använder följande formler för att bestämma PID-konstanterna:

- Proportionalitetskonstanten K_p beräknas som

$$K_p = 0,6 \cdot K_u = 0,6 \cdot 20 = 12$$

- Integrationskonstanten K_i ges av K_p / T_i , vilket ger

$$K_i = \frac{K_p}{T_i} = \frac{12}{0,35} = 34$$

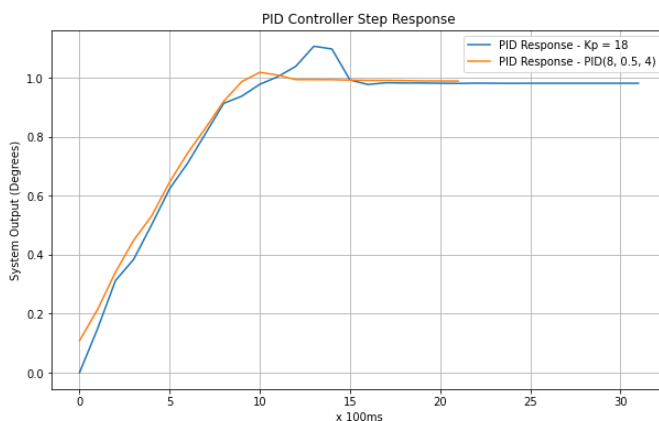
- Derivationskonstanten K_d ges av formeln $K_d = 0,6 \cdot K_u \cdot T_u$ vilken kan beräknas med tidigare värden och ger

$$T_d = \frac{T_u}{8} = \frac{0,7}{8} = 0,0875$$

$$K_d = K_p \cdot T_d = 12 \cdot 0,0875 = 1,05$$

Med Ziegler-Nichols justering skulle PID-värdena bli: $K_p=12$, $K_i=34$ och $K_d=1,05$. Dessa värden kan dock vara olämpliga eftersom ett högt K_i -värde innebär en stark vikt på det ackumulerade felet, medan ett lågt K_d -värde skulle innebära att hastigheten i förändringen av kamerans position har liten påverkan på korrigeringen, vilket kan resultera i instabilitet och stora kvarvarande fel.

För att motverka detta justerades PID-värdena under testning tills en bra balans uppnåddes mellan snabb respons och minimala kvarstående fel. En effektiv kombination som uppnåddes genom praktisk testning var $K_p=8$, $K_i = 0,5$ och $K_d = 4$. Resultaten av denna inställning kan observeras i figur 8.



Figur 9. Självcentrering med olika P -värden, med självstängning och utan självsvängning.

5.3 Programmering

Mikrokontrollerns programmering och implementeringen av PID-regleringen utfördes i Visual Studio Code, med PlatformIO som utvecklingsmiljö och Arduino-kärnan som bas. Programmeringen skedde i Arduino/C++, där centrala funktioner som PID-reglering, motorkontroll och sensordata hantering implementerades.

5.3.1 Kodstruktur och organisation

För att förbättra hanteringen och överskådligheten delades koden upp i separata filer, där varje fil ansvarade för en specifik funktionalitet inom projektet. Detta möjliggjorde en modulär uppbyggnad och underlättade felsökning och vidareutveckling.

main.cpp: Huvudfilen som fungerar som ingångspunkt för programexekveringen.

Sensorhantering: Separata filer skapades för hanteringen av accelerometern och magnetometern, vilket inkluderade insamling och bearbetning av data.

Motorkontroll: En dedikerad fil ansvarade för att styra motorerna baserat på sensordata och PID-regleringen. Motorstyrningen implementerades genom att skicka sinussignaler till DRV8313-motordrivaren, som i sin tur drev BLDC-motorn. För att aktivera motorn användes funktionerna `moveMotor()` och `moveMotorAntiClockWise()`, vilka körde motorn i en förbestämd riktning beroende på behov. Motorhastigheten och styrkan justerades dynamiskt baserat på PID-regulatorns felvärde (errorvärde), vilket säkerställde att motorn reagerade korrekt på förändringar i systemets tillstånd.

För att undvika blockering av exekveringen, som kan orsakas av användning av funktioner som `sleep()`, kalibrerades körfunktionerna för att kunna exekveras i varje cykel på mikrokontrollern vid behov. Detta gjorde att motorkontrollen kunde ske i realtid utan att påverka systemets responsivitet.

PID-reglering: En egen fil hanterade logiken för PID-regleringen, som justerade motorernas respons för att hålla kameran stabil och centrerad.

5.3.2 PID-reglering

PID-kontrollern implementerades i filen `pid.cpp` som kontinuerligt blev matad sensordata från MPU6050 IMU sensorn. Börvärdet var satt till 0 grader (kameran på plan nivå) där ärvärdet inmatades från sensordata i form av data.

Implementeringen av PID-regulatorn i C++ är som följande kodsnuitt:

```
float PID::compute(float current_value, float target_value)
{
    unsigned long current_time = millis();
    float time_change = (current_time - last_time) / 1000.0; // konvertering från ms till s

    float error = target_value - current_value;

    integral += error * time_change;
    float derivative = (error - previous_error) / time_change;

    float output = Kp * error + Ki * integral + Kd * derivative;

    previous_error = error;
    last_time = current_time;

    return output / 1000.0; // skalning för utsignal till motorn
}
```

Figur 10. Kodutdrag från PID-algoritm implementerad i C++.

I koden representerar `current_value` det aktuella värdet från sensorn och `target_value` det önskade målvärdet. För att beräkna tidsskillnaden som behövs för den integrerande (I) och deriverande (D) delen, räknas först tidsintervallet mellan nuvarande och föregående tidpunkt ut, omvandlat till sekunder.

Felvärdet (`error`) beräknas sedan som skillnaden mellan målvärdet (`target_value`) och det aktuella värdet (`current_value`). Den integrerande delen (`integral`) beräknas genom att multiplicera felvärdet med tidsintervallet, och adderas kontinuerligt så länge ett fel finns. För att undvika "integraluppbyggnad" (ackumulering av felet över tid när ett konstant fel kvarstår), sätts integralvärdet till noll när målvärdet och det aktuella värdet är lika och inget fel kvarstår.

Den deriverande delen (*derivative*) beräknas genom att ta skillnaden mellan nuvarande fel och föregående fel, dividerat med tidsintervallet. Detta visar hur snabbt felet ändras över tid; ju snabbare felet ändras (dvs. ju snabbare skillnaden mellan mål- och aktuellt värde förändras), desto större blir den deriverande delen, vilket påverkar storleken på PID-utgångsvärdet.

PID-utgången (*output*) beräknas genom att summera de proportionella (P), integrerande (I), och deriverande (D) delarna, multiplicerade med sina respektive konstanter, K_p , K_i , och K_d . Slutligen skalas utgången för att passa signalen till motorn.

5.3.3 Biblioteksanvändning

För att hantera kommunikationen och sensordata från MPU6050-accelerometern/gyrometern användes ett externt bibliotek [18] som tillhandahöll API. Dessa API inkluderade funktioner för initial kontakt med sensorn via I²C-kommunikationsprotokollet, korrigering av felvärden, samt användning av interrupt-porten för att kompensera för drift och motverka eventuella felvärden. Användningen av biblioteket underlättade integrationen med MPU6050-sensorn genom att tillhandahålla nödvändiga funktioner för bearbetning och tolkning av rådata.

5.3.4 Sensordata

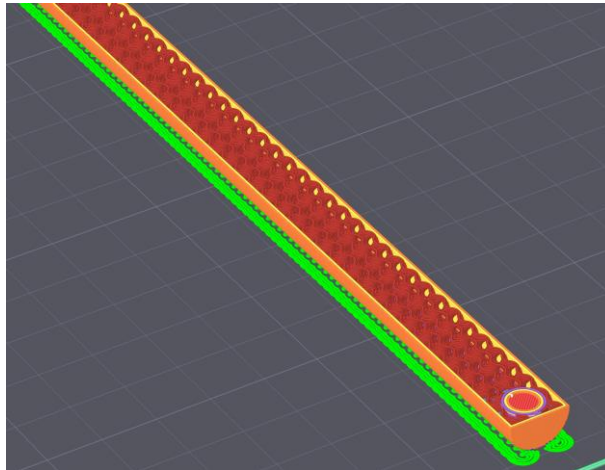
Sensordata kunde ibland uppvisa plötsliga hopp eller vibrationer, även när kamerafästet, där sensorn var monterad, var stilla. Detta berodde på brus eller felaktiga mätningar från sensorn, vilket ledde till att PID-kontrollen reagerade på felaktiga data och försökte justera motorerna i enlighet med dessa. Som ett resultat började kameran röra sig oförutsägbart, trots att den borde ha förblivit stationär. Detta problem kunde skapa störningar i bildstabiliteten och minska systemets övergripande prestanda och tillförlitlighet.

För att hantera dessa rörelser togs ett medelmått av de senaste 5 värdena i ett försök att jämna ut signalerna och motverka eventuella störningar. Detta resulterade tyvärr i en lite fördröjning av invärdet till PID-regulatorn.

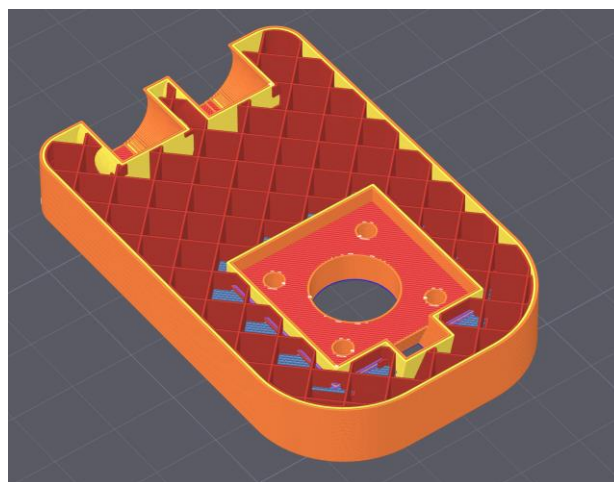
5.4 Konstruktion

Gimbalkonstruktionen tillverkades med hjälp av en 3D-skrivare, där alla komponenter, inklusive gimbalpinnarna, huset för BLDC-motorn och kamerans monteringsfäste, skapades i PLA-plast. Gimbalens konstruktion designades i CAD-mjukvaran Fusion360, där en digital modell skapades för att visualisera designen och säkerställa dess funktionalitet. Varje komponent designades separat och sammanfogades därefter i modellen för att simulera funktionen. De färdiga CAD-filerna exporterades i .stl-format till en så kallad slicer-mjukvara för att förbereda dem för 3D-utskrift. Programmet som användes, BambuStudio, är en slicer-mjukvara utvecklad av 3D-skrivartillverkaren BambuLab, som även tillverkade den 3D-skrivare samt PLA-filament som användes för att skriva ut komponenterna till gimbalen.

För 3D-utskriftarna användes PLA-plast. De flesta komponenterna printades med standardinställningar som inkluderade 15 % infill med ett "grid"-mönster, vilket minskade materialåtgången för delar som inte hade en avgörande roll i konstruktionens integritet. För komponenter som hade en kritisk strukturell funktion, som kopplingsdosor och pelare, justerades inställningarna för att öka hållfastheten. Kopplingsdosorna printades med 30 % infill med ett grid-mönster, medan pelarna också printades med 30 % infill men med dubbla ytterlager och ett "gyroid"-mönster för att ytterligare stärka konstruktionen.



Figur 11. En gimbal pelare med gyroid mönster och 25% infillning som förbereds för 3D-skrivning i slicermjukvaran.



Figur 12. Motorfästet för gimbalen med grid mönster och 15% fyllning som förbereds för 3D-utskrivning.

För att optimera hållbarheten och minska risken för brott i pelarna 3D-printades gimbal-pinnarna diagonalt. Genom att orientera varje lager längs den horisontella axeln minskades risken för att konstruktionen skulle försvagas. Tidigare försök, där pinnarna printades i en annan riktning, resulterade i avsevärt reducerad styrka och gjorde dem mer benägna att gå sönder under förväntad belastning, eftersom sammanhållningen då främst berodde på plastens ihopsmältning snarare än materialets inneboende styrka.

5.4.1 Montering

För att sammanfoga de 3D-printade komponenterna i PLA-plast användes "press and shrink fit"-tekniken. Genom att designa delarna med små toleranser kunde en stark presspassning uppnås vilket minimerar rörelser mellan komponenterna och säkerställer en stabil struktur. Vissa komponenter kylde ned innan montering för att lättare passa ihop, medan andra värmdes upp, vilket underlättade monteringen och förstärkte fogen när delarna återgick till rumstemperatur.

För att möjliggöra säker montering och demontering utan att skada de 3D-printade delarna, användes mässingsinsatser (brass heat inserts) för att skapa hållbara gängade infästningspunkter. Mässingsinsatserna värmdes upp och pressades in i de förformade hålen på de 3D-printade komponenterna. När plasten svalnade och härdades kring insatserna, skapade detta en stark och slitstark infästning som klarar av repetitiva monterings- och demonteringscykler utan att förlora sin hållfasthet.

5.5 Elektronik koppling

För att mäta kamerans lutning och säkerställa korrekt vinkelposition, integrerades en MPU6050 IMU-sensor på bottenplattan där kameran är monterad. Sensorns huvudsyfte är att registrera lutningsvinklar och identifiera eventuella avvikelser från den önskade positionen.

MPU6050-breakoutkortet anslöts till Teensy-mikrokontrollern på följande sätt:

1. **Strömtillförsel:** En likspänning på 3,3 V tillfördes från Teensy-mikrokontrollern till MPU6050, som därmed fungerade som strömkälla för sensorn.
2. **Kommunikation:** För dataöverföring kopplades SDA-porten på MPU6050 till SDA0 på Teensy, och SCL-porten till SCL0. Detta etablerade en I²C-kommunikation mellan sensorn och mikrokontrollern, vilket möjliggör snabb och effektiv dataöverföring.
3. **Interrupt-funktion:** För att förbättra mätnoggrannheten och reducera fel som orsakas av driften i accelerometerns värden, kopplades INT-porten (interrupt) från MPU6050 till en digital I/O-port på Teensy-mikrokontrollern. Denna anslutning möjliggjorde användning av avbrotts signaler, vilket förbättrar systemets reaktionsförmåga och mätnoggrannhet.

För att mäta vridningen på BLDC-motorn användes en AS5600-magnetometer, som var fäst vid motorn och ansluten till mikrokontrollern. Denna magnetometer är avgörande för att övervaka och justera motorns rotationsvinkel med hög precision.

AS5600-breakoutkortet anslöts till Teensy-mikrokontrollern enligt följande:

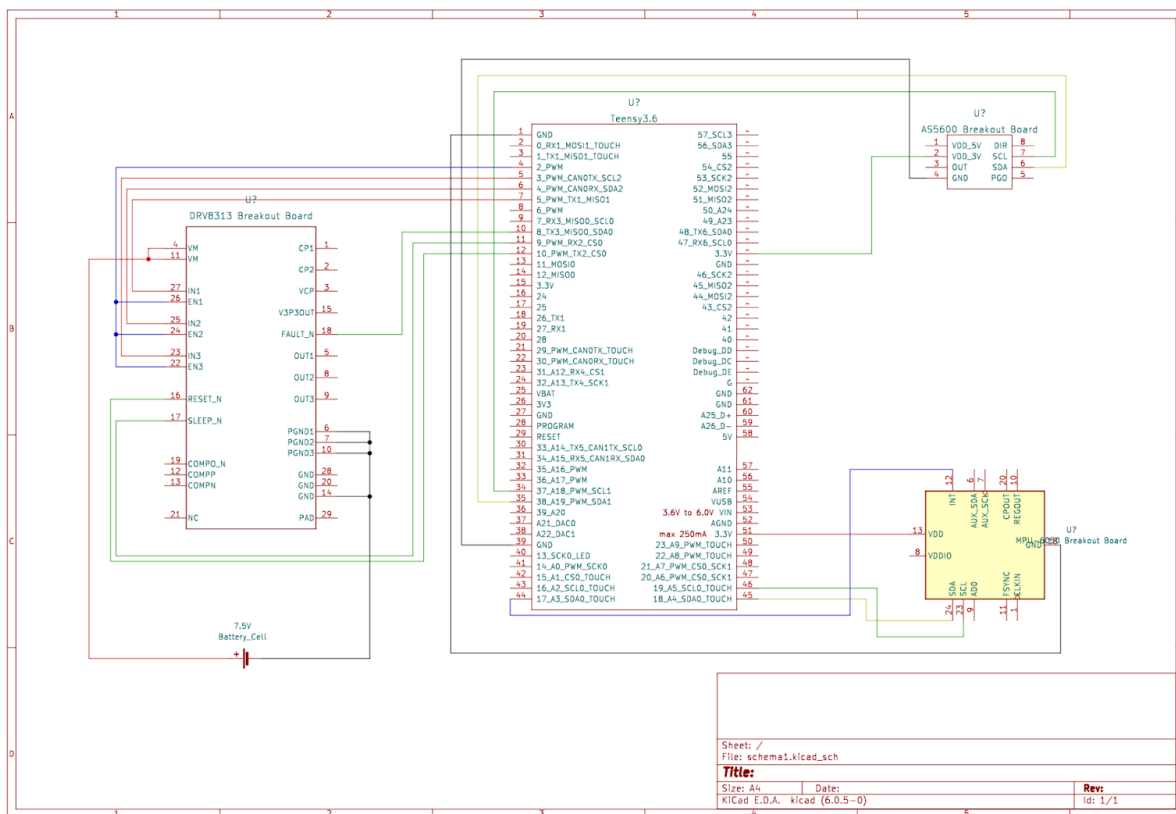
1. **Strömtillförsel och jordning:** En likspänning på 3,3 V tillfördes från Teensy-mikrokontrollern till AS5600 för att driva sensorn. Samtidigt etablerades jordning för att säkerställa en stabil och korrekt funktion av systemet.
2. **Kommunikation:** För att möjliggöra dataöverföring mellan magnetometern och mikrokontrollern kopplades SDA-porten på AS5600 till SDA1 och SCL-porten till SCL1 på Teensy-mikrokontrollern. Denna koppling skapade en I²C-kommunikationskanal, vilket möjliggjorde en effektiv och pålitlig överföring av mätdatan.

För att driva BLDC-motorn anslöts ett batteripaket till Texas Instruments DRV8313-motordrivaren, vilket säkerställde tillräcklig spänning och ström för att driva motorn. DRV8313-motordrivaren kopplades även till mikrokontrollern för att möjliggöra styrning av motorn via programvara.

Anslutningen mellan DRV8313-breakoutkortet, Teensy-mikrokontrollern och BLDC-motorn gjordes enligt följande:

1. **Strömförsörjning:** Batteripaketet anslöts till DRV8313-motordrivaren för att tillhandahålla den nödvändiga spänningen och strömmen för att driva BLDC-motorn effektivt.

- Motorstyrning:** ENABLE-pinnen på DRV8313, som är en gemensam koppling för alla ENABLE-pinnar på kretskortet, anslöts till PWM-pinne 2 på Teensy-mikrokontrollern. Denna anslutning aktiverade motorstyrningen genom att möjliggöra styrning av motorns faser.
- Sinusoidal Styrning:** IN1, IN2 och IN3 på DRV8313 anslöts till PWM-portar på Teensy-mikrokontrollern. Dessa anslutningar tillät sinusoidal styrning av motorn genom att generera PWM-signaler från Teensy.
- Övervakning och Kontroll:** FAULT-, SLEEP- och RESET-pinnarna på DRV8313 anslöts till digitala I/O-portar på Teensy-mikrokontrollern. Dessa anslutningar möjliggjorde övervakning och hantering av DRV8313-kretskortets status och säkerställde att motordrivaren fungerade korrekt under drift.



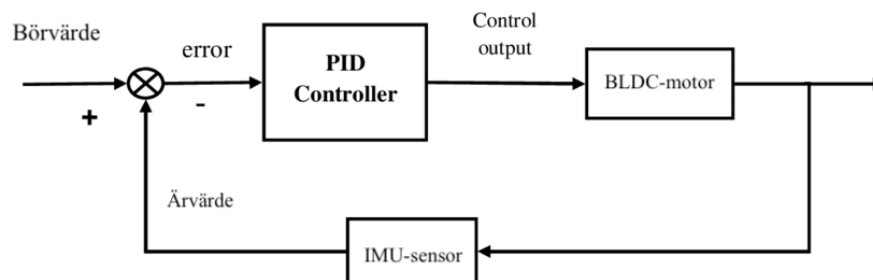
Figur 13. Kretsschema över koppling mellan mikrokontroller och sensorgivare.

6. RESULTAT

I detta kapitel presenteras det resultat som framställts utifrån de förutsättningar som framförts.

6.1 PID kontroller

En PID-kontroller (Proportionell, Integrerande, Deriverande) valdes på grund av dess förmåga att ge exakt kontroll över systemets rörelser genom att kontinuerligt justera insatsvärdena baserat på avvikelser från önskat läge. Implementeringen av PID-kontrollen utfördes i C++, vilket möjliggjorde en flexibel och direkt integrering med den övriga systemkoden.



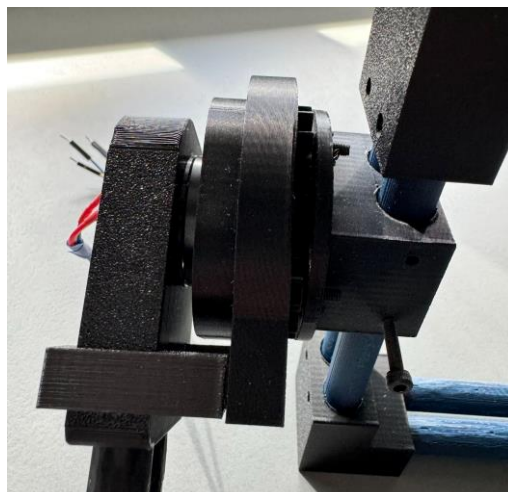
Figur 14. Design av PID kontroller för centrering av gimbal.

På grund av den implementerade planetväxeln och det ökade vridmomentet uppstod en förlust i hastighet. Eftersom motorn inte längre kunde uppnå samma hastighet blev det omöjligt för PID-regulatorn att generera översvängningar. Detta gjorde det svårare att justera systemet och hitta de optimala P-, I- och D-parametrarna för att uppnå önskad prestanda.

Kvaliteten på den 3D-printade planetväxeln ledde också till ökad friktion mellan kugghjulen, vilket ytterligare bidrog till att sakta ner rotationen.

6.2 Planetväxel

Med den implementerade planetväxel kunde kameran roteras i 360 grader, i vilken riktning som var nödvändig, men som ett resultat med högre vridmoment var gimbalen inte lika snabb och tog längre tid på sig för att reagera till ändringar och störningar till börvärdet.



Figur 15. Bild på planetväxeln.

6.3 Självcentrering

Med den implementerade planetväxeln kunde gimbalen rotera som förväntat, men med en lägre hastighet än planerat. Genom att använda data från accelerometern och gyrometern kunde mikrokontrollern exakt bestämma kamerans vinkel. Denna information matades in i PID-regulatorn, vilket möjliggjorde justeringar när den faktiska vinkeln avvek från det önskade värdet, som initialt var inställt på 0 grader.

På så sätt kunde gimbalen effektivt centrera kameran på 0 grader och automatiskt korrigera för eventuella avvikelser, vilket säkerställer att kameran förblev stabilt centrerad.

7. DISKUSSION OCH SLUTSATS

Följande kapitel ger en mer ingående analys över resultatet och hur det kan utvecklas i framtiden.

7.1 Sammanfattning av resultatet

Projektet syftade till att utveckla en självcentrerande kameragimbal för att underlätta fotografering av stora objekt och effektivisera datainsamling för maskininlärningsapplikationer. Målet att uppnå automatisk centrerung av objektet i kamerans bildram uppnåddes genom gimbalen's konstruktion. Med integration av objektspårningsteknik i den monterade kameran kan gimbalen även följa rörliga objekt och hålla dem centrerade i bildramen.

Under projektets gång identifierades dock vissa prestandabegränsningar. Den ursprungliga motorn hade inte tillräckligt med vridmoment för att stabilisera kameran effektivt, vilket krävde införandet av en planetväxel för att öka motorns vridmoment. Detta ledde till en minskning av justeringshastigheten, vilket innebär att gimbalen uppvisade en viss fördröjning vid snabba rörelser innan den kunde åter centrera objektet.

Självcentrerungen fungerade tillfredsställande för applikationer där objektet huvudsakligen är stillastående men kan röra sig vid olika tillfällen. För sådana applikationer kan gimbalen uppnå och upprätthålla centrerungen av objektet effektivt.

7.2 Prestanda och effektivitet hos gimbalen

Vid bedömningen av gimbalen's prestanda visade sig vissa styrkor, särskilt i dess förmåga att bibehålla kamerans centrerung vid statiska och långsamma rörelser. Systemets precision i att hålla objektet centrerat under långsamma rörelser uppfyllde de förväntade kraven, vilket gör det väl lämpat för tillämpningar där objektets position inte förändras snabbt.

Trots dessa fördelar observerades en begränsning i systemets justeringshastighet vid plötsliga rörelser. Gimbalen visade en märkbar fördröjning innan den kunde åter centrera objektet, vilket kan vara otillräckligt i scenarier som kräver snabb respons. Denna begränsning påverkar särskilt applikationer där objektet ofta ändrar position snabbt, såsom sportfotografering eller andra dynamiska miljöer.

Gimbalen presterade bäst i kontrollerade miljöer och förhållanden där objektets rörelser är långsamma eller förutsägbara. Dess förmåga att upprätthålla stabilitet och centrerung gör den användbar för specifika tillämpningar, men ytterligare optimeringar krävs för att förbättra responsiviteten i mer krävande miljöer.

7.3 Tekniska begränsningar och utmaningar

Utvecklingen av gimbalen stötte på flera tekniska utmaningar som påverkade systemets prestanda och användbarhet. En av de mest framträdande begränsningarna var otillräcklig kraft hos den ursprungliga motorn, vilket föranledde implementeringen av en planetväxel för att öka vridmomentet. Även om detta ökade gimbalen's förmåga att hantera tyngre laster, påverkade det samtidigt systemets justeringshastighet negativt, vilket resulterade i långsammare återcentrerung vid snabba rörelser.

En annan teknisk utmaning var den ökade friktionen i den 3D-printade planetväxeln. Materialvalet (PLA-plast) och precisionen i de utskrivna komponenterna bidrog till en förhöjd friktion mellan kugghjulen, vilket inte bara minskade justeringshastigheten utan även ökade energiförbrukningen. För att förbättra effektiviteten skulle ett mer lämpligt material med lägre friktion och högre hållbarhet, såsom nylon eller metall, kunna användas för planetväxeln.

Slutligen påverkas prestandan av sensorernas begränsade noggrannhet. Den använda accelerometern och gyroskopet var tillräckliga för grundläggande rörelseupptäckt men uppvisade drift och brus som ledde till minskad precision över tid. Att använda mer avancerade sensorer med högre upplösning och bättre stabilitet skulle kunna minska dessa problem och förbättra systemets responsivitet och noggrannhet.

7.4 Miljöpåverkan

Den skanningen av produkterna som Wiretronic gör syftar till att underlätta för kunder att lokalisera rätt reservdelar till olika produkter. Gimbalen, som möjliggör inskanning av större objekt, vidgar då utbudet att kunna erbjuda samma möjlighet för fler produkter. I ett hållbarhetsperspektiv bidrar gimbalen till att förse kunder med reservdelar för att kunna reparera något befintligt i stället för att köpa nytt.

Gimbalen tillverkades i PLA-plast, ett fossilfritt och biologiskt nedbrytbart material som gör det till ett miljövänligt alternativ jämfört med många andra plaster. Tillverkningen skedde med hjälp av 3D-utskrift, vilket minimerade materialspill och bidrog till en mer hållbar användning av resurser.

7.5 Förslag till förbättringar

För att förbättra gimbalen's prestanda och effektivitet identifierades flera potentiella åtgärder för framtida iterationer av projektet. En viktig förbättring skulle vara att uppgradera den nuvarande motorn till en starkare BLDC-motor med högre vridmoment. Detta skulle förbättra gimbalen's responsivitet och minska fördröjningen vid justeringar, vilket skulle möjliggöra bättre hantering av tyngre kameror och snabbare rörelser. Dessutom skulle en optimering av drivsystemet, såsom införandet av mer avancerade styrmetoder som sinusformad styrning, kunna bidra till mjukare och mer exakta rörelser.

Vidare skulle en optimering av mjukvarualgoritmerna kunna förbättra gimbalen's stabilitet och responsivitet. Justering av PID-parametrarna genom ytterligare tester kan hjälpa till att finjustera systemets reaktionsförmåga och minska översvängningar eller fördröjningar vid plötsliga rörelser. Dessutom kan adaptiva algoritmer som dynamiskt justerar PID-värdena baserat på rörelsens hastighet och typ implementeras för att ytterligare förbättra systemets prestanda.

För att öka noggrannheten och stabiliteten kan mer avancerade sensorer övervägas. Användningen av högupplösta IMU-sensorer, och eventuellt fler sensorer för att förbättra precisionen i rörelse- och positioneringsmätningarna samt ha fler mätpunkter för att kunna detektera felaktiga avvikelser i sensordatan.

En annan viktig förbättring skulle vara att designa gimbalen med en mer modulär arkitektur, vilket skulle underlätta framtida uppgraderingar och komponentbyten. Detta kan inkludera användning av standardiserade monteringspunkter och anslutningar, vilket gör det möjligt att snabbt anpassa gimbalen för olika användningsområden eller uppdatera komponenter utan att behöva göra omfattande förändringar i designen.

7.6 Slutsatser och framtida arbete

Sammanfattningsvis uppfyller projektet sitt huvudsakliga mål att utveckla en fungerande prototyp av en självcentrerande kamera gimbal för fotografering av stora objekt. Medan flera av de tekniska utmaningarna adresserades framgångsrikt, finns det fortfarande utrymme för förbättringar, särskilt vad gäller motorprestanda och sensorsystemens precision. Framtida arbete bör fokusera på att optimera gimbalen's design och implementera högpresterande komponenter för att ytterligare förbättra stabiliteten och prestandan, samt utföra mer omfattande tester för att säkerställa att gimbalen uppfyller kraven under olika fotograferingsförhållanden.

Referenser

- [1] PhotoRobot, "360° Produktfotografering - Innehållsformat och produktion", 2022. [Online]. Tillgänglig: <https://sv.photorobot.com/blog/360-product-photography> [Hämtad: Sep. 2, 2024].
- [2] J. Hellerman, "What is a gimbal?", *No film school*, [Online], Jun. 10, 2024. Tillgänglig: <https://nofilmschool.com/what-is-a-gimbal-definition> [Hämtad: Sep. 2, 2024].
- [3]. Bambu Lab, "PLA-Matte Technical Data Sheet." [Online]. Tillgänglig: https://cdn.shopify.com/s/files/1/0645/5876/0155/files/Bambu_PLA_Matte_Technical_Data_Sheet.pdf?v=1695182428. [Hämtad: Sep. 12, 2024].
- [4] GlobalSpec, "Gimbals information", [Online]. Tillgänglig: https://www.globalspec.com/learnmore/motion_controls/positioning_stages_slides_guides/gimbals [Hämtad: Dec. 22, 2024].
- [5]. PJRC, "Tensy 3.6 Development Board." [Online]. Tillgänglig: <https://www.pjrc.com/store/teensy36.html>. [Hämtad: Sep. 12, 2024].
- [6]. OSRAM, "AS5600 Magnetic Rotary Position Sensor Data Sheet." [Online]. Tillgänglig: <https://look.ams-osram.com/m/7059eac7531a86fd/original/AS5600-DS000365.pdf>. [Hämtad: Sep. 12, 2024].
- [7]. Texas Instruments, "DRV8313 Three-Phase Motor Driver Data Sheet," 2023. [Online]. Tillgänglig: https://www.ti.com/lit/ds/symlink/drv8313.pdf?ts=1717090047728&ref_url=https%253A%252F%252Fwww.google.com%252F. [Hämtad: Sep. 12, 2024].
- [8]. InvenSense, "MPU-6000 and MPU-6050 Product Specification," TDK, 2013. [Online]. Tillgänglig: https://product.tdk.com/system/files/dam/doc/product/sensor/motion-inertial/imu/data_sheet/mpu-6000-datasheet1.pdf. [Hämtad: Sep. 12, 2024].
- [9]. Renesas Electronics Corporation, "What are Brushless DC Motors," Renesas Electronics Corporation, Sep. 2024. [Online]. Tillgänglig: <https://www.renesas.com/en/support/engineer-school/brushless-dc-motor-01-overview>. [Hämtad: Sep. 23, 2024].
- [10]. "Hubs Guide to 3D Printing," Hubs, Sep. 2024. [Online]. Tillgänglig: <https://www.hubs.com/guides/3d-printing/>. [Hämtad: Sep. 23, 2024].
- [11]. PlatformIO, "What is PlatformIO?" [Online]. Tillgänglig: <https://docs.platformio.org/en/latest/what-is-platformio.html>. [Hämtad: Sep. 12, 2024].
- [12]. Arduino, "Introduction to Arduino." [Online]. Tillgänglig: <https://www.arduino.cc/en/Guide/Introduction>. [Hämtad: Sep. 12, 2024].
- [13]. MathWorks, "PID Control: An Introduction to Proportional, Integral, and Derivative Control." [Online]. Tillgänglig: <https://se.mathworks.com/discovery/pid-control.html>. [Hämtad: Sep. 12, 2024].
- [14]. Texas Instruments, "I²C: Understanding I²C and the I²C Communication Protocol," 2023. [Online]. Tillgänglig: <https://www.ti.com/lit/an/sbaa565/sbaa565.pdf?ts=1724733938304>. [Hämtad: Sep. 12, 2024].

- [15]. Zureks, "Pulse-width modulation graph," *Pulse-width modulation*, Wikipedia. [Online]. Tillgänglig: https://en.wikipedia.org/wiki/Pulse-width_modulation. [Hämtad: Sep. 12, 2024].
- [16]. Electrokit, "BLDC Motor: DC Brushless Gimbal Motor 3 Phase Datasheet." [Online]. Tillgänglig: https://www.electrokit.com/upload/product/41019/41019421/OT-EM3215_DC_Brushless_Gimbal_Motor_3_Phase_Datasheet.pdf. [Hämtad: Sep. 12, 2024].
- [17] J. G. Ziegler and N. B. Nichols, "Optimum settings for automatic controllers," *Trans. ASME*, vol. 64, no. 8, pp. 759-768, Nov. 1942.
- [18]. GitHub, "I²Cdevlib-MPU6050 Code Library." [Online]. Tillgänglig: <https://registry.platformio.org/libraries/jrowberg/I2Cdevlib-MPU6050>. [Hämtad: Sep. 12, 2024].



CHALMERS