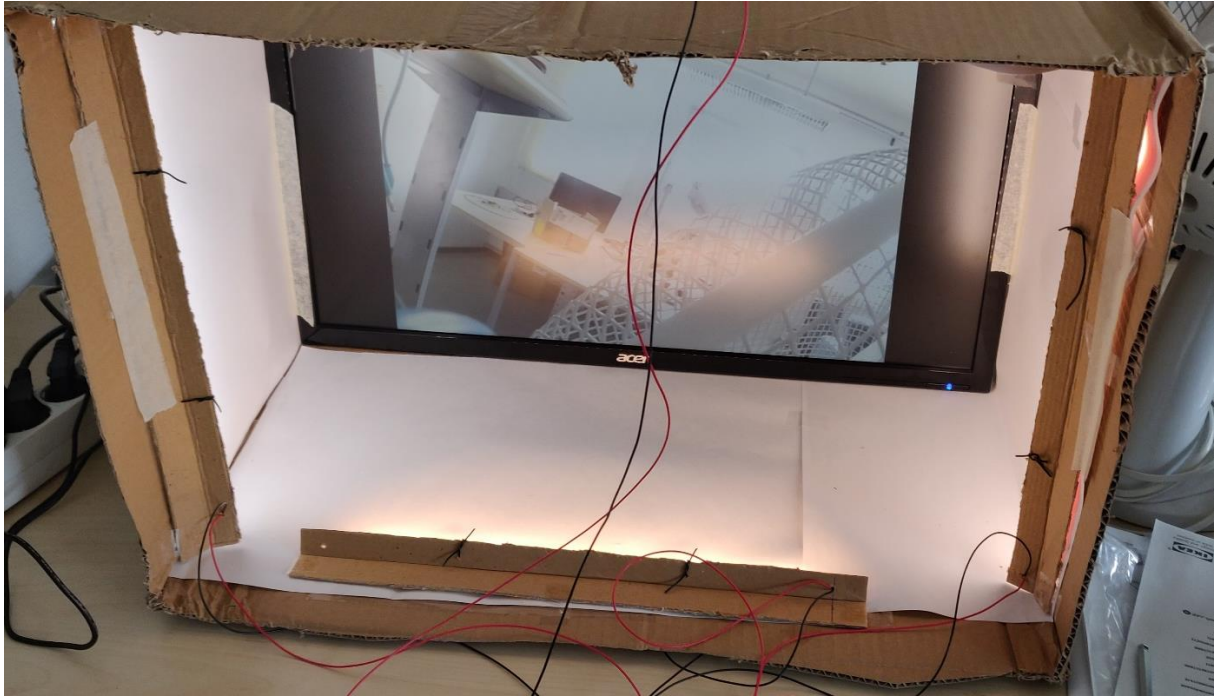




# CHALMERS



## Elektroniskt virtuellt fönster

Examensarbete inom högskoleingenjörsprogrammet Mekanik

SVANTE KJELLBERG



## FÖRDORD

Detta examensarbete omfattar 15hp och skrevs på Institutionen för elektroteknik på Chalmers tekniska högskola. Det utfördes av Svante Kjellberg på Mekanikingenjörsprogrammet 180hp hos Broccoli Engineering AB.

Tack till Björn Bergholm på Broccoli Engineering för idén av ämnet och möjligheten att utföra examensarbetet hos er.

Tack till Bertil Thomas på Chalmers för all hjälp som jag fått under den här tiden. Det har varit välbehövt och är väldigt uppskattat.

A handwritten signature in black ink, consisting of two parts: 'Svante' on the left and 'Kjellberg' on the right. The signature is fluid and cursive.

## SAMMANFATTNING

Virtuella fönster är ett koncept med syfte att simulera ett fönster för att ge belysning och utblick framförallt i utrymmen där vanliga fönster inte skulle vara möjliga. Det kan handla om att det finns bärande väggar i vägen eller att rummet befinner sig under jorden. Tanken är att filma en vy, från exempelvis ett vanligt fönster och visa det på en skärm. Skärmen som kan vara en datorskärm eller liknande, ska regleras på ett sådant sätt att det efterliknar referensfönstret så mycket som möjligt. Det finns många olika effekter för att ett virtuellt fönster ska efterlikna ett vanligt fönster som; ljusstyrka, färg, utstrålningsvinkel, perspektiv för betraktaren med mera.

Projektet gick ut på att utveckla en prototyp av ett virtuellt fönster som med begränsad utrustning skulle efterlikna ett referensfönster. Prototypen som byggdes fokuserade främst på att reglera ljusstyrkan. För att reglera ljusstyrkan användes externa LED-moduler för att lysa upp istället för att ändra skärmen i sig. Fotoresistorer användes för att mäta ljusstyrkan vid ett referensfönster som även filmades av en kamera. Det kameran filmade visades på en datorskärm som agerade som det faktiska virtuella fönstret. Regleringen programmerades i python med en enkortsdator som styrde LED-modulernas ljusstyrka. Prototypen har potential för utveckling då den i nuläget endast reglerar ljusstyrkan i ett ganska begränsat intervall. Prototypen kan fungera som ett inredningsalternativ men bör i nuläget inte användas för att ersätta belysningen från ett vanligt fönster.

## ABSTRACT

The concept of virtual windows is to simulate the appearance, light and feel of a real window. The application for such a window would be in the situation where a real window wouldn't be possible to install. This could be for example underground or in a room with no walls connected to the outside. The idea is to film, for example, a regular window and display it on a screen. The screen could be a computer screen or something similar. The display is regulated so that it mimics the characteristics of the reference window as much as possible. Characteristics such as; brightness, colour, angle of the light, perspective for the viewer and more.

The project was about developing a prototype of a virtual window that, with limited equipment, would emulate a reference window. The prototype focused mainly on regulating the brightness level. To adjust the brightness, it used external LED modules to control the light instead of controlling the screens brightness level. Photoresistors were used to measure the brightness of the reference window and the virtual window to give feedback to the system. A camera was used to film the reference window and display it on the virtual window screen. A single board computer was programmed with python to be used as the regulator of the system and control the brightness level of the LED modules.

There is much room for development. The brightness level of the virtual window is restricted to a limited interval and could be improved with better equipment. The prototype could be used as decoration but shouldn't be considered to replace the lighting of a regular window as of now.

# Innehåll

1.	INLEDNING.....	1
1.1	Bakgrund.....	1
1.2	Syfte.....	1
1.3	Mål.....	1
1.4	Avgränsningar.....	1
1.5	Specificering av frågeställningen.....	2
2.	TEORETISK OCH TEKNISK BAKGRUND.....	3
2.1	Ljus och hälsa.....	3
2.2	Tillgängliga lösningar.....	3
2.3	Egenskaper hos fönster och skärmar.....	3
2.4	Raspberry Pi 3 Model B.....	6
2.5	Runcam 2.....	6
2.6	A/D-omvandlare.....	6
2.7	Optokopplare.....	6
2.8	Fotomotstånd.....	7
2.9	LED-modul.....	7
2.10	Pulse Width Modulation.....	7
2.11	Stegsvarsidentifiering.....	7
2.12	Ziegler-Nichols svängningsmetod.....	9
3.	METOD.....	10
3.1	Hårdvara.....	10
3.2	Beräkning och mjukvara.....	11
3.2.1	Stegsvarsidentifiering.....	11
3.2.2	PID – reglering.....	13
3.2.3	Filtrera störningar.....	16
3.2.4	Flödesschema av programmet.....	17
4.	RESULTAT.....	18
5.	SLUTSATS OCH DISKUSSION.....	19
5.1	Vidare utveckling.....	19
	REFERENSER.....	i
	BILDREFERENSER.....	ii
	APPENDIX.....	iii
	Programkod.....	iii
	Simulinkmodeller.....	vi



## BETECKNINGAR

PWM – Pulse width modulation

A/D – Analog/Digital

RbPi – Raspberry Pi

PID – Proportionell, integrerande och deriverande

ZN – Ziegler-Nichols

LED – Light Emitting Diode



# 1. INLEDNING

I vissa byggnadskonstruktioner finns det begränsade möjligheter att montera fönster där man vill ha in dagsljus. Det kan vara på grund av saker som att konstruktionen av en vägg inte tillåter ett fönster eller att rummet är beläget på en plats där ett fönster varit värdelöst till exempel en källare under marknivå.

## 1.1 Bakgrund

Det är allmänt känt att brist på dagsljus påverkar människor negativt i form av humör, sömnrhythmer med mera. Detta kan bli ett större problem med tanke på hur mycket tid som många människor spenderar inomhus numera samt att elektriskt ljus än så länge inte kan ersätta dagsljus. Om det är möjligt att återskapa dagsljus till den grad att det kan ersättas med elektriskt ljus öppnar det upp för väldigt många möjligheter. Då kan det skapas ett elektroniskt "fönster" med ljus som är likt det ljus som går igenom ett vanligt fönster som man kan ha som referensfönster. Detta kan implementeras i utrymmen där ett vanligt fönster inte är möjligt att implementera. Det öppnar upp möjligheten att bygga lägenheter och kanske till och med hela städer under jorden.

## 1.2 Syfte

Syftet med projektet är att undersöka hur väl det går att återskapa ljus från ett fönster med hjälp av billiga komponenter i form av LED-moduler, fotoresistorer, en datorskärm, en kamera och en enkortsdator.

## 1.3 Mål

Målet med projektet är att bygga ett elektroniskt "fönster" som ska regleras på sådant sätt att ljuset anpassas efter ett referensfönster, visar det som syns genom referensfönstret och efterliknar det så bra som möjligt utifrån de begränsningar som sätts. Fotoresistorerna ska läsa av ljusstyrkan hos referensfönstret och genom reglering via enkortsdatorn styra LED-modulerna som simulerar ljuset från referensfönstret. Datorskärmen ska representera själva fönstret och kameran ska spegla det som ses genom fönstret i realtid till skärmen

## 1.4 Avgränsningar

- I ett vanligt fönster förändras det man ser genom det beroende på vilken position man har relativt fönstret vilket inte kommer att behandlas i detta examensarbete.
- Det är begränsat av den utrustning som kommer att vara tillgänglig.
- Det kommer behandla framförallt ljusintensitet. I mån av tid finns det optioner som skulle kunna behandla färgtemperatur och annat för att göra det mer realistiskt.
- Skärmen används endast för att visa bilden från referensfönstret och regleras inte i sig.
- Tidsramen för detta projektet är ca 10 - 12 veckor.

## 1.5 Specificering av frågeställningen

- I. Hur väl kan begränsad utrustning simulera ett fönsters ljusintensitet?
- II. Kan en datorskärm upplevas som ett fönster med hjälp av en kamera?
- III. Hur väl kan LED-lampor efterlikna sin färg efter dagsljus?

## 2. TEORETISK OCH TEKNISK BAKGRUND

Detta kapitlet kommer att presentera information kring olika komponenter och metoder som används i projektet.

### 2.1 Ljus och hälsa

Det är bevisat och numera allmänt känt att dagsljus har positiva hälsoeffekter på människor och att brist på det kan orsaka olika typer av hälsoproblem. Det kan påverka humöret, dagsrytmen, stressregleringen och sömnrutmen. [1] Eftersom vi i dagens samhälle spenderar mycket tid inomhus innebär det att det finns en ökad risk för att få brist på dagsljus. Elektriskt ljus har inte lika brett spektrum som dagsljus vilket kan innebära att färgåtergivningen blir annorlunda. Elektriskt ljus är också ofta statiskt och innehåller ingen information om väder och tid. Det har inte heller samma hälsoeffekter som dagsljus har.

### 2.2 Tillgängliga lösningar

CoeLux är ett företag som erbjuder några versioner av ett takmonterat virtuellt fönster. De har fokuserat på att återskapa realistisk belysning genom att skapa ett filter till sin belysning som ska motsvara atmosfären komprimerad till några millimeters tjocklek vilket ger en väldigt övertygande känsla av faktiskt solljus. [2] En av deras mindre produkter simulerar ett fönster som är ca 1,2 m x 0,6 m och kan lysa upp ca 20 kvadratmeter men väger ungefär 200 kg. Priset på deras fönster är också ganska högt: ca 34 000 \$ totalt (2016).

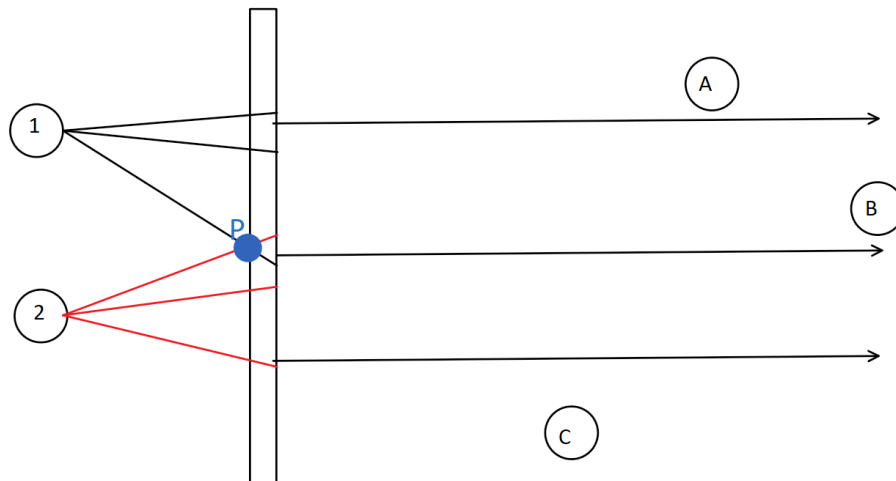
Atmosph erbjuder ett "fönster" för utblick och vyer istället för belysning. Det är en datorskärm som spelar upp högupplösta videor på vyer med ljud. [3] Den kan kontrolleras via en applikation för smartphones och kostar mellan 700 \$ - 900 \$. Den väger ca 5 kg och är 64 cm x 38 cm.

### 2.3 Egenskaper hos fönster och skärmar

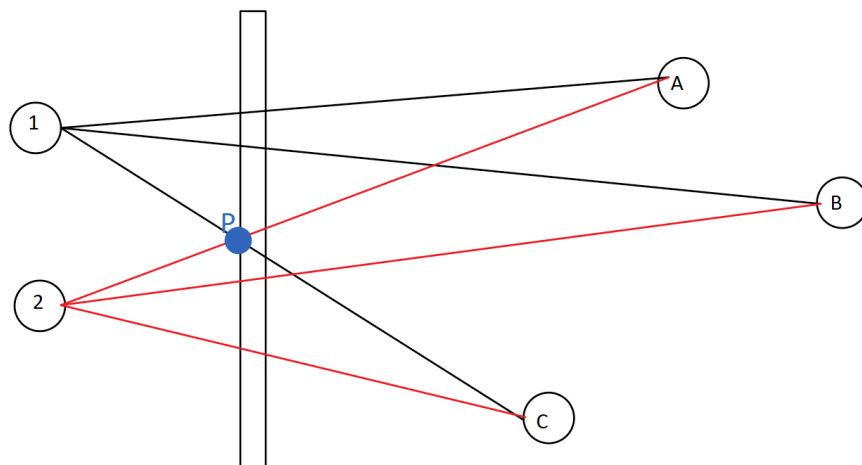
För att kunna återskapa ett fönster så väl som möjligt är det viktigt att ta upp vilka egenskaper ett fönster har jämfört med en skärm.

En datorskärm drivs av elektricitet vilket drar en del energi, men ett fönster kan uppbringa energiförluster genom att det har sämre värmeisolering än väggen som det är monterat i vilket gör att värmeenergi går förlorad ur byggnaden. En datorskärm kräver också mer varierat material för att bygga jämfört med ett fönster. Skärmens kvalitet kan göra väldigt stor skillnad. Upplösningen och färgerna kan variera väldigt mycket och påverkar upplevelsen oerhört.

I Figur 2.1 och Figur 2.2 demonstreras en av de största skillnaderna mellan ett fönster och en skärm vilket är att bilden projiceras på skärmen.



Figur 2.1 Perspektiv på skärm



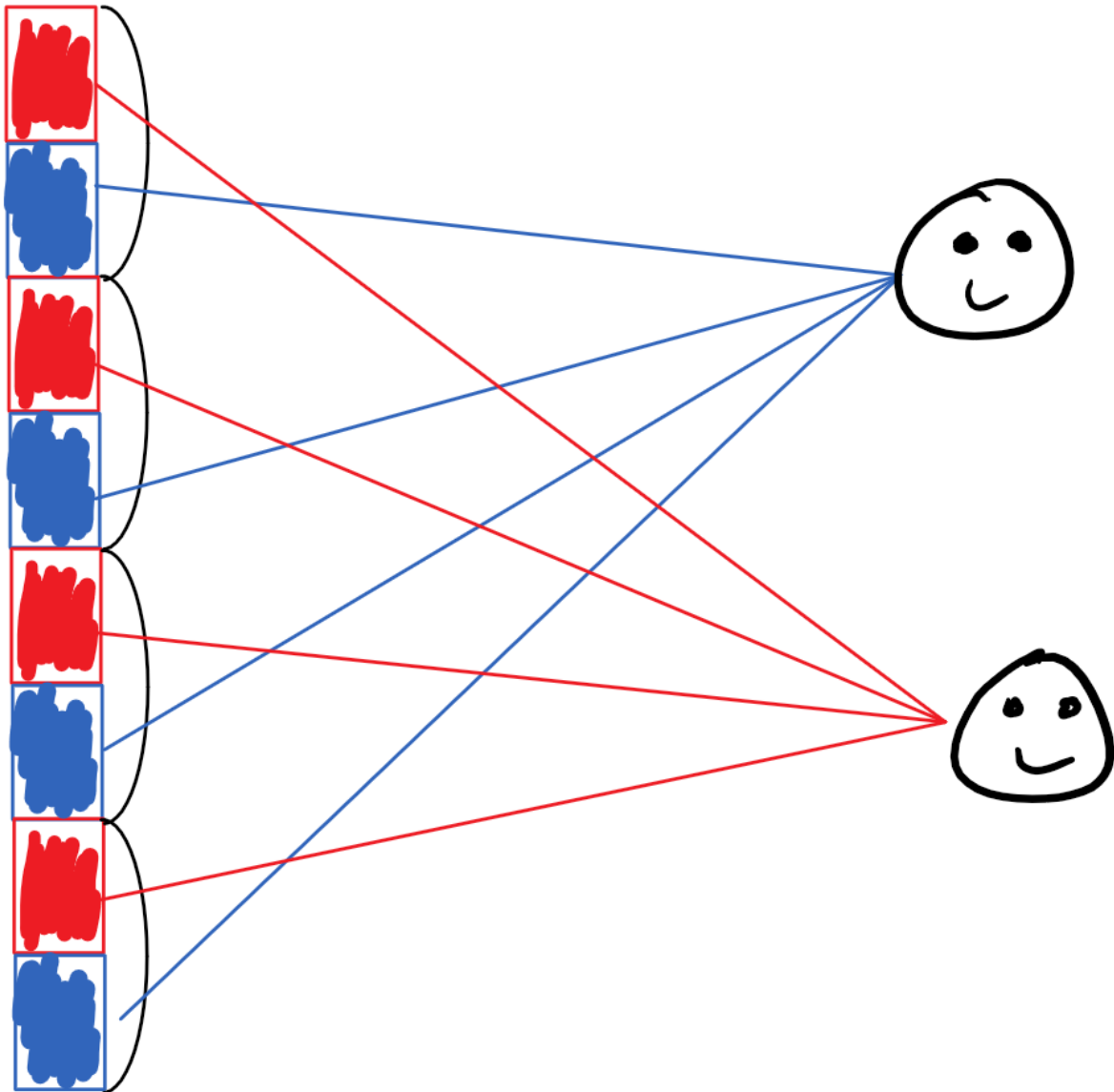
Figur 2.2 Perspektiv genom fönster

Eftersom bilden projiceras på datorskärmen gör det att bilden ser likadan ut oavsett vilken placering som betraktaren (1 och 2) har. I ett fönster så förändras bilden och placeringen av objekten (A, B och C) som syns genom det beroende på avstånd till objektet och betraktarens position. Om betraktaren står i position 1 och tittar på punkten P på fönstret så kommer betraktaren se objektet C, men om betraktaren står i position 2 och tittar på punkten P kommer betraktaren att se objektet A. På samma sätt kan inte heller en skärm simulera solljus då det bara kommer att projiceras på skärmen istället för att stråla in i den vinkel som det kom från.

Ett exempel för att skärmen ska ändra perspektivet beroende på betraktarens position är att sätta upp flera kameror som filmar motivet i olika vinklar för att skapa en panoramabild. Därefter kan man med hjälp av sensorer av valfritt slag identifiera betraktarens position och anpassa bilden efter det. Det skulle dock inte fungera med fler betraktare samtidigt.

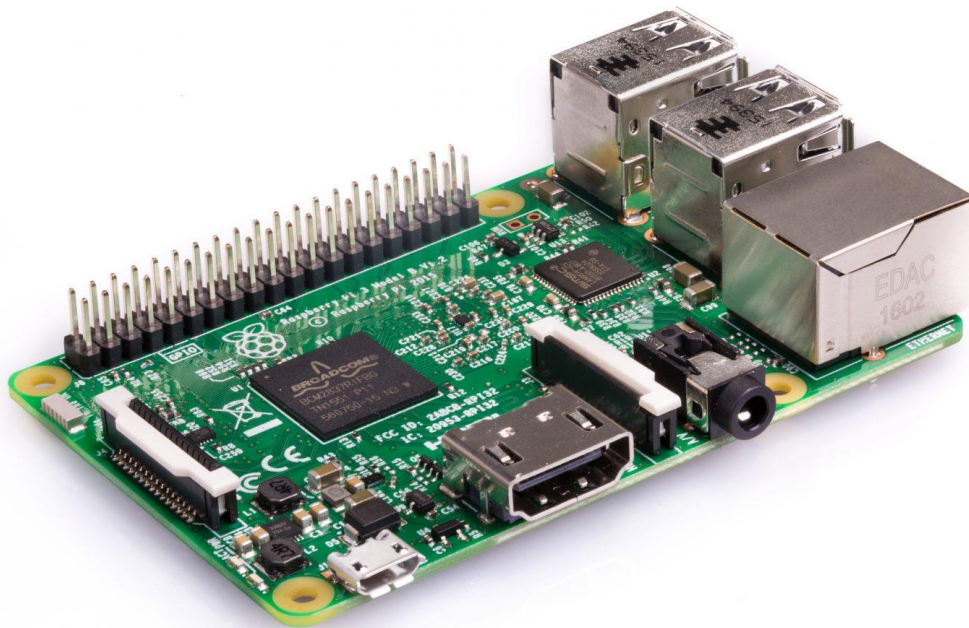
Lentikulärt tryck är när två bilder är uppdelade i strimlor där strimlorna delas in i par med en lins som bryter ljuset så att ena bilden visas från ett perspektiv och den andra bilden visas från ett annat perspektiv. Se Figur 2.3. Med detta skulle man kunna visa ett panoramamotiv som är anpassat för specifika vinklar. Står man mellan dessa vinklar kommer man att se bilderna

blandat med varandra vilket gör att bilden blir suddig och otydlig. Då kan det vara flera betraktare men de måste stå i rätt vinkel i förhållande till skärmen för att bilden ska bli bra.



*Figur 2.3 Lentikulärt tryck, olika perspektiv visar olika bilder*

## 2.4 Raspberry Pi 3 Model B



Figur 2.4. Raspberry Pi 3 Model B [4]

Raspberry Pi 3 Model B är en enkortsdator. Den har 40 stycken GPIO-pinnar (General Purpose Input Output) med möjlighet för PWM-styrning, digitala ingångar och utgångar med mera. [4] Den har även HDMI-utgång, USB-portar och ethernet-utgång. Raspberry Pi är en väldigt vanlig dator inom hobby och hemmaprojekt.

## 2.5 Runcam 2

Runcam 2 är en kamera som är designad för att kunna användas främst till drönare och andra hobbyprojekt. [5] Den är liten och behändig med liten fördröjning vilket eftersöktes då bilden ska anpassa sig så snabbt som möjligt till det som syns genom fönstret. Runcam 2 har en pixelupplösning på 4 megapixel och kan filma i 1080p upplösning och 60 bilder per sekund.

## 2.6 A/D-omvandlare

En A/D-omvandlare omvandlar analoga signaler till digitala. I detta projektet användes en MCP3008 vilket är ett microchip med åtta analoga ingångar och som har en upplösning på 10 bitar. [6] Den konverterar den analoga signalens spänning till en digital 10 bitars serie som beskriver spänningen mellan 0 - 5 V i siffror mellan 0 - 1023.

## 2.7 Optokopplare

Optokopplare är en komponent som separerar strömmar i olika system med hjälp av en LED och en fototransistor. När ström går genom LED aktiverar den fototransistorn så att ström går

genom fototransistorn. Detta gör det möjligt att till exempel styra en 24 V signal med en 5 V signal utan att det finns någon risk att den ena spänningen kommer över i andra kretsen vilket skulle kunna skada utrustningen.

## 2.8 Fotomotstånd

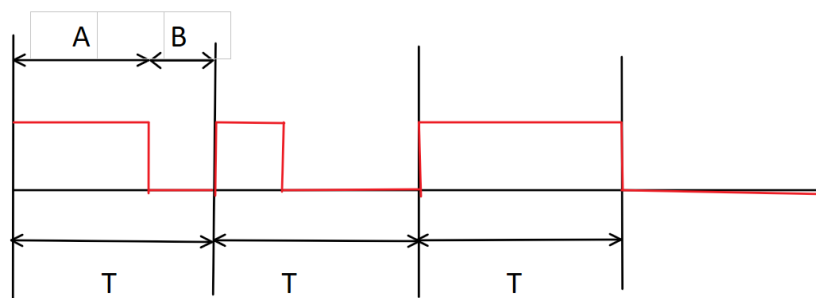
Ett fotomotstånd är ett motstånd som varierar i resistans beroende på hur mycket ljus som faller på den. När den utsätts för mer ljus så minskar resistansen.

## 2.9 LED-modul

I projektet användes tre LED-moduler (LRAML SW800) som på varje modul har 28 små LED-lampor. [7] De har en driftspänning på 24V och drar 3W. Modulerna har alla olika ljstemperaturer, nämligen 3000K, 4000K och 5000K. De kan dimmas med hjälp av PWM signaler.

## 2.10 Pulse Width Modulation

Pulse Width Modulation (PWM) eller pulsbreddsmodulering är en pulsbaserad signal vars bredd på pulsen kan ändras till önskat värde. Det innebär att man kan bestämma hur länge en signal ska vara aktiv under en period med bestämd tid. Se Figur 2.5.



Figur 2.5 Pulsbreddsmodulering

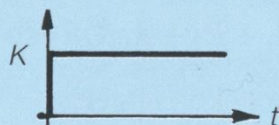
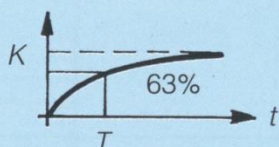
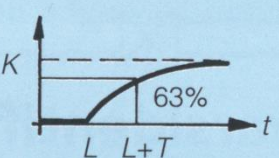
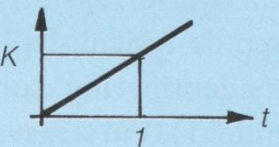
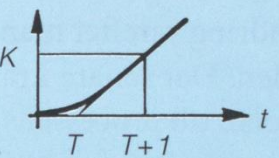
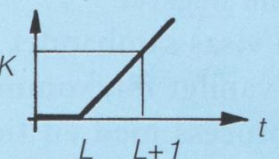
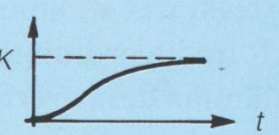
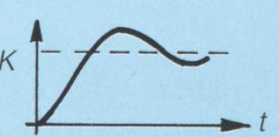
A = Aktiv längd

B = Inaktiv längd

T = Periodlängd

## 2.11 Stegvarsidentifiering

En metod för att ta fram överföringsfunktioner för processer är stegvarsidentifiering. Det är en experimentell metod som går ut på att skicka en stegformad insignal till processen och analysera utsignalen.[8] Utsignalen kan sedan jämföras med en tabell över typiska överföringsfunktionstyper se Figur 2.6. När en graf som liknar ens egen har valts så är det bara att ta fram konstanterna för tillhörande överföringsfunktion.

Typ av process	Överföringsfunktion	Stegsvar (vid enhetssteg)
Process med P-verkan	$G(s) = K$	
En tidskonstant	$G(s) = \frac{K}{1+Ts}$	
En tidskonstant + dödtid	$G(s) = \frac{Ke^{-Ls}}{1+Ts}$	
Integration	$G(s) = \frac{K}{s}$	
Integration + en tidskonstant	$G(s) = \frac{K}{s(1+Ts)}$	
Integration + dödtid	$G(s) = \frac{K \cdot e^{-Ls}}{s}$	
Två tidskonstanter (se även s. 117–118)	$G(s) = \frac{K}{(1+T_1s)(1+T_2s)}$	
Andra ordningens process med översväng (se även s. 120)	$G(s) = \frac{K}{as^2 + bs + 1}$	

Figur 2.6. Samband mellan överföringsfunktioner och stegsvar [8]



## 2.12 Ziegler-Nichols svängningsmetod

ZN-svängningsmetod är en metod för att dimensionera PID-regulatorer och dess parametrar ( $K, T_I, T_D$ ). [8] Den går ut på att öka förstärkningen på regulatorn som ställs in som en P-regulator tills systemet börjar självsvänga varpå förstärkningen  $K_0$  och periodtiden  $T_0$  för svängningarna mäts upp. De uppmätta värdena ska sedan användas till parametrarna enligt följande tabell.

Regulatortyp	Parametrar		
	$K$	$T_I$	$T_D$
P-regulator	$0,5 K_0$	–	–
PI-regulator	$0,45 K_0$	$0,85 T_0$	–
PID-regulator	$0,6 K_0$	$0,5 T_0$	$0,125 T_0$

Där  $K_0$  = Den förstärkning som gör att ett system med P-reglering börjar självsvänga

$$K_0 = \frac{1}{|G_p(\omega_\pi)|}$$

och

$T_0$  = Periodtiden för svängningarna som uppstår

$$T_0 = \frac{2\pi}{\omega_\pi}$$

## 3. METOD

Arbetet började med att undersöka vilken typ av utrustning som skulle användas. Därefter testades olika alternativ som det kunde användas på. När det var klart vilket alternativ som skulle användas skaffades de sista delarna av utrustningen och allt kopplades upp på behörigt sätt.

Under projektets gång testades en del python-kod som kunde användas vilket blev grunden till ett program som kunde testa all utrustning så att det fungerade som tänkt. Med alla delar testade så programmerades ett program som skulle bli det slutgiltiga programmet för projektet. När programmets struktur var klar påbörjades undersökningen om hur det skulle regleras. Stegsvår mättes upp för den identifierade processen och med hjälp av Matlab och Simulink kunde en modell sättas upp och regleras enligt ZN-tumregelmetod.

### 3.1 Hårdvara

RbPi:en är regulatorn i systemet och är kopplad till alla reglerande delar men inte till kameran och skärmen. Dessa kopplades till en annan tillgänglig enkorts dator för att bespara RbPi:en välbehövad processorkraft och speglade endast kamerans bild på skärmen.

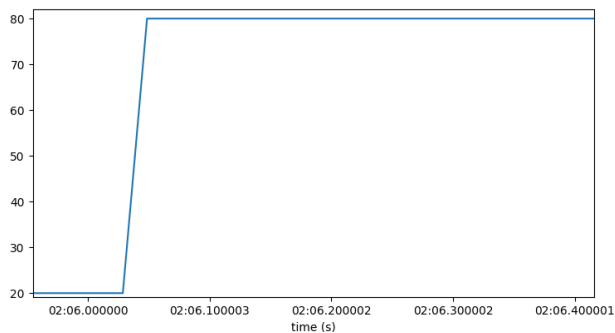
Eftersom LED-modulerna inte kan uppnå ljusstyrkan av direkt solljus anpassas inte systemet för att hantera det utan fokuset hamnar på omgivningens ljusstyrka. Eftersom fotomotståndet reagerar väldigt kraftigt på om den utsätts för direkt belysning jämfört med omgivningens belysning är LED-modulerna monterade i riktning från sensorn mot skärmen så att ljuset reflekteras på omgivningen innan det når fotomotståndet. Detta ger något mindre intensiv belysning på sensorn men ger mycket mer realistisk upplevelse av ljuset. De tre LED-modulerna har också olika färgtemperaturer men monteringen gör att det inte märks på samma nivå vilket kan vara en fördel eftersom färgtemperaturen inte regleras i det här projektet.

Sensormodulerna är uppbyggda av en fotoresistor som är kopplad till +5V och till en resistor och sen till jord. Signalen mäts mellan resistorn och fotoresistorn. Signalen som fotomotstånden avger är i analog form vilket inte RbPi:en kan hantera utan måste göras om till en digital signal. Detta sker med hjälp av MCP3008 som kopplas till ett antal GPIO-portar på RbPi:en och översätter den analoga signalen till en digital signal som RbPi:en kan hantera. Därefter skickar RbPi:en en PWM signal till en optokopplare som styr LED-modulernas ljusstyrka.

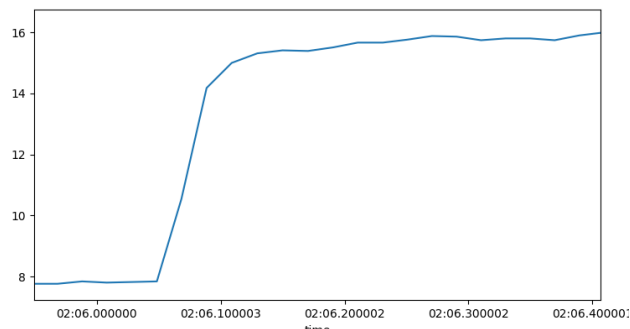
## 3.2 Beräkning och mjukvara

### 3.2.1 Stegvarsidentifiering

Genom att skicka ett steg till systemet och sedan använda sig av stegvarsidentifiering av grafer i python och modellering i simulink togs det fram en överföringsfunktion som representerar processen på en bra nivå. Figur 3.1 visar styrvärdet och Figur 3.2 visar utsignalen från processen.



Figur 3.1. Styrvärdet vid steg

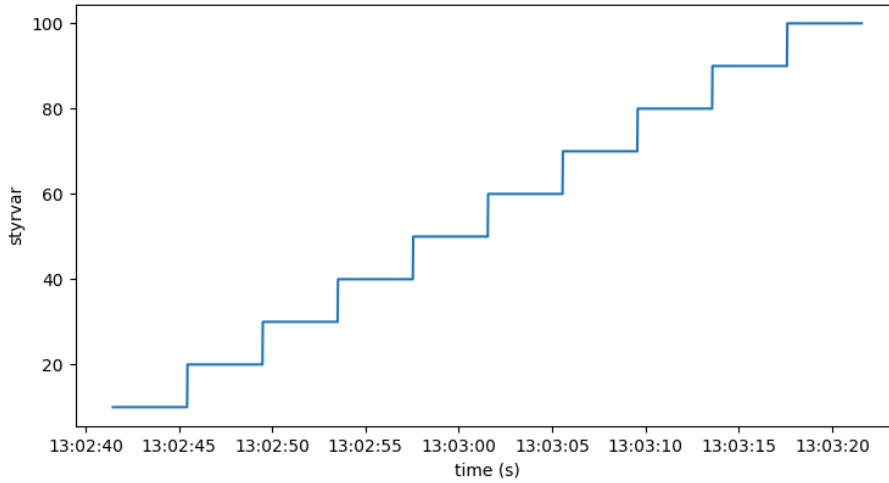


Figur 3.2. Ärvärdet vid steg

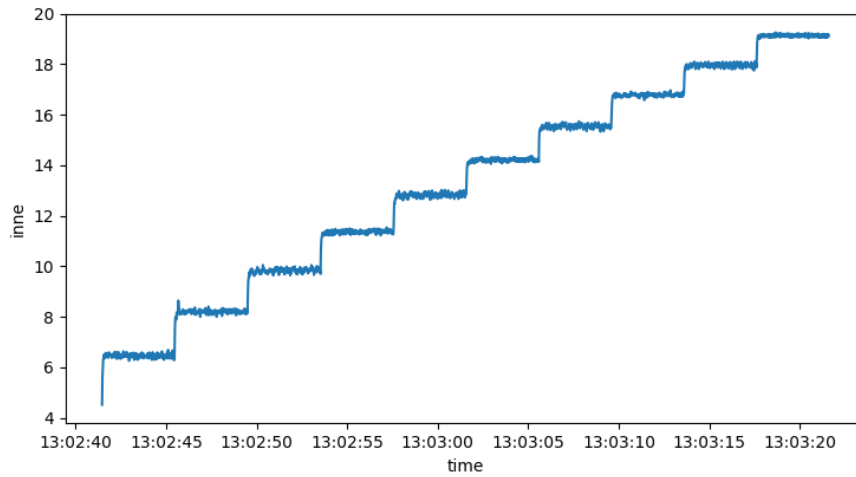
Överföringsfunktionen som togs fram med hjälp av stegvarsanalys enligt metodiken som presenteras i 2.11. Den tredje av graferna i Figur 2.6 är den som är mest lik, vilket innebär att det är den överföringsfunktionen som kommer att användas. Värdena är uppmätta från grafen i Figur 3.2.

$$G(s) = \frac{K * e^{-Ls}}{1 + Ts} = \frac{0,133 * e^{-0,0197s}}{1 + 0,0331s}$$

Överföringsfunktionen stämde inte på alla punkter när olika steg testades se Figur 3.3 och Figur 3.4 då K-värdet blev annorlunda för de olika stegen. Slutsats att processens K-värde inte var konstant drogs.

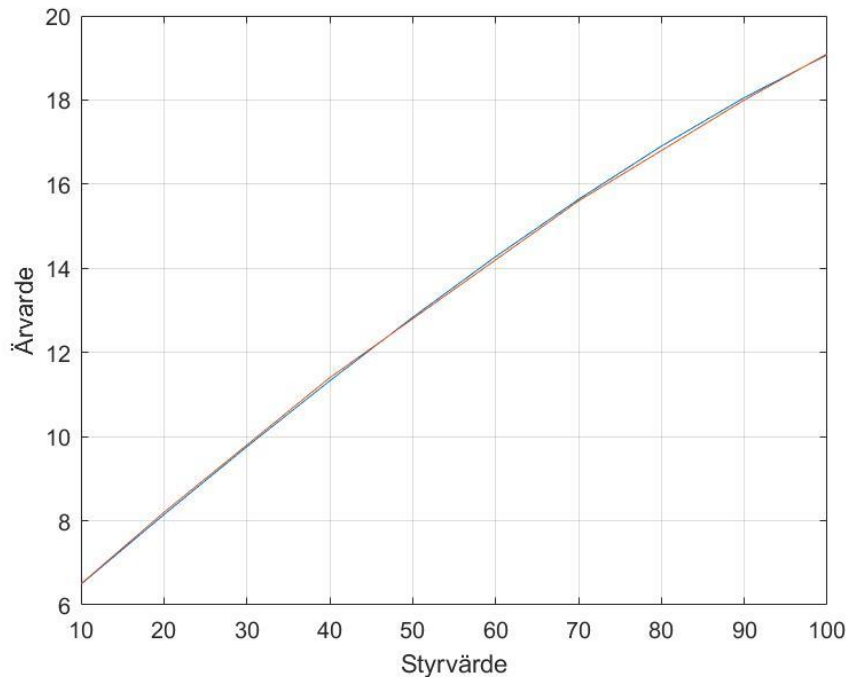


Figur 3.3. Styrvärdet vid olika steg



Figur 3.4. Ärvärdet vid olika steg

För att beskriva processens K-värde anpassades en formel till de uppmätta mätvärdena genom att approximera i Matlab som matchade mätpunkterna tillräckligt väl. Se Figur 3.5.



Figur 3.5. Graf över K-värdet, formel jämfört med uppmätt

Formeln som togs fram matchar förstärkningen där felet går upp till max 0.1 i skillnad från den uppmätta förstärkningen.

$$K(u) = 5,8 + 0,1875 * u - e^{0,017u}$$

Genom att implementera formeln för K-värdet med resten av överföringsfunktionen i simulink skapades en bra modell av processen. Se Simulinkmodeller i APPENDIX.

### 3.2.2 PID – reglering

För att reglera systemet på ett bra och utförligt sätt behövs det en regulator. PID-reglering var väldigt lämpligt trots att deriverande reglering ibland kan vara slitsamt för de styrdon som används då de måste göra många ändringar ofta. I detta projektet blir det endast RbPi:ens GPIO-pinnar som skickar PWM-signalen och en optokopplare som påverkas av det vilket gör att det inte blir ett stort problem.

PID-regulatorns parametrar bestämdes med hjälp av ZN-tumregelmetod då det är ett effektivt sätt att reglera ett färdigt system med goda resultat. I efterhand ändrades parametrarna lite grann genom testning för att göra det något bättre.

Parametrarna från ZN enligt kapitel 2.12:

$$\begin{aligned} P &= 18 \\ T_I &= 0,033 \\ T_D &= 0,00825 \end{aligned}$$

Slutgiltiga parametrarna:

$$\begin{aligned} P &= 10 \\ T_I &= 0,033 \\ T_D &= 0,005 \end{aligned}$$

Med parametrarna satt så behövdes regleringen formateras till en formel för styrvärdet som ska skickas som en PWM signal. Formeln beskrivs tidsdiskret då styrvärdet uppdateras enligt programmets flöde. Samplingsintervallet ( $h$ ), uppskattades till 0,01 från graferna.

$$\text{styrvärde} = P * (\text{felvärde} + (1/T_I) * \text{Ivärde} + T_D * \text{Dvärde})$$

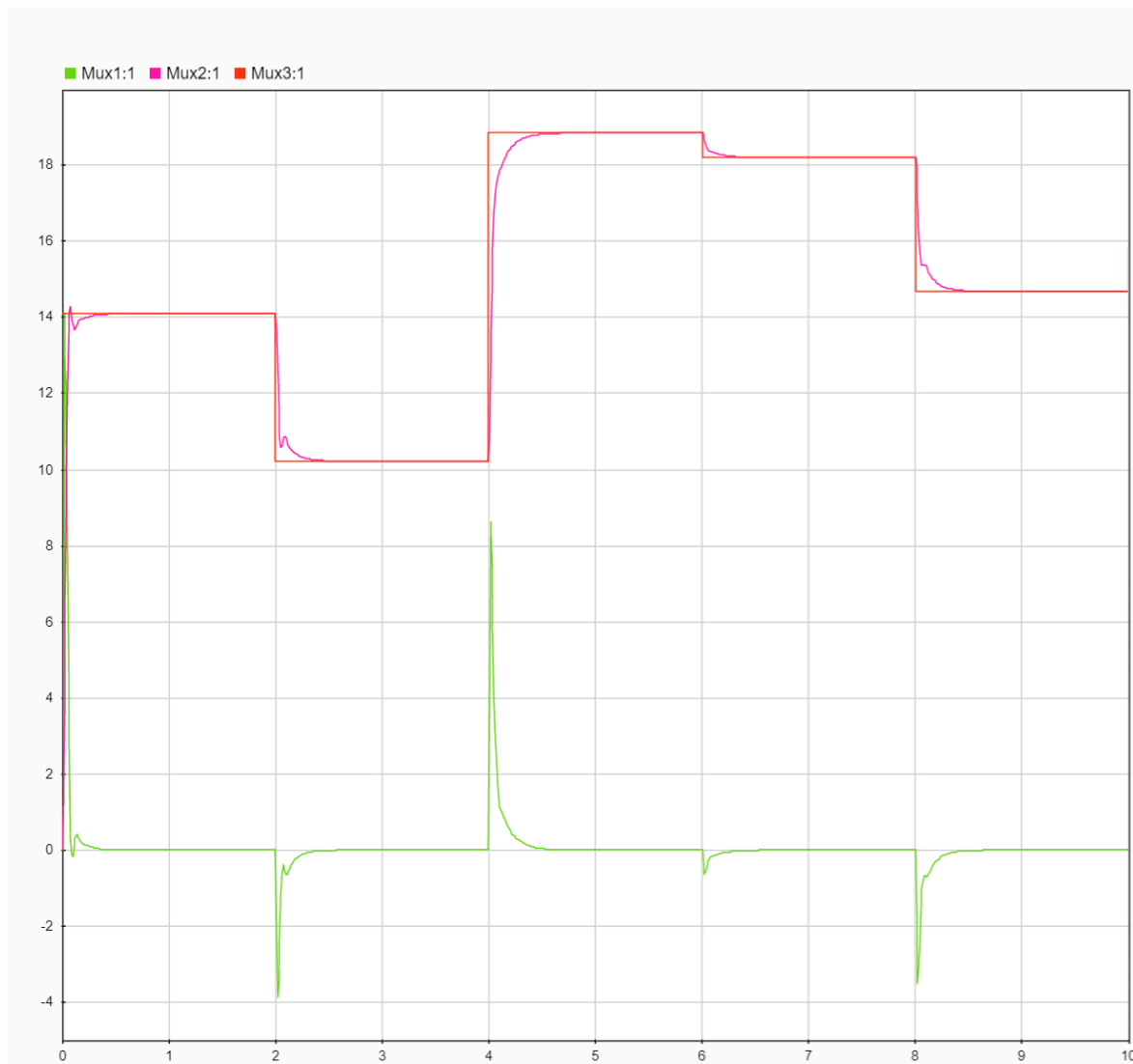
Där

$$\text{felvärde}(k) = \text{börvärde}(k) - \text{ärvärde}(k)$$

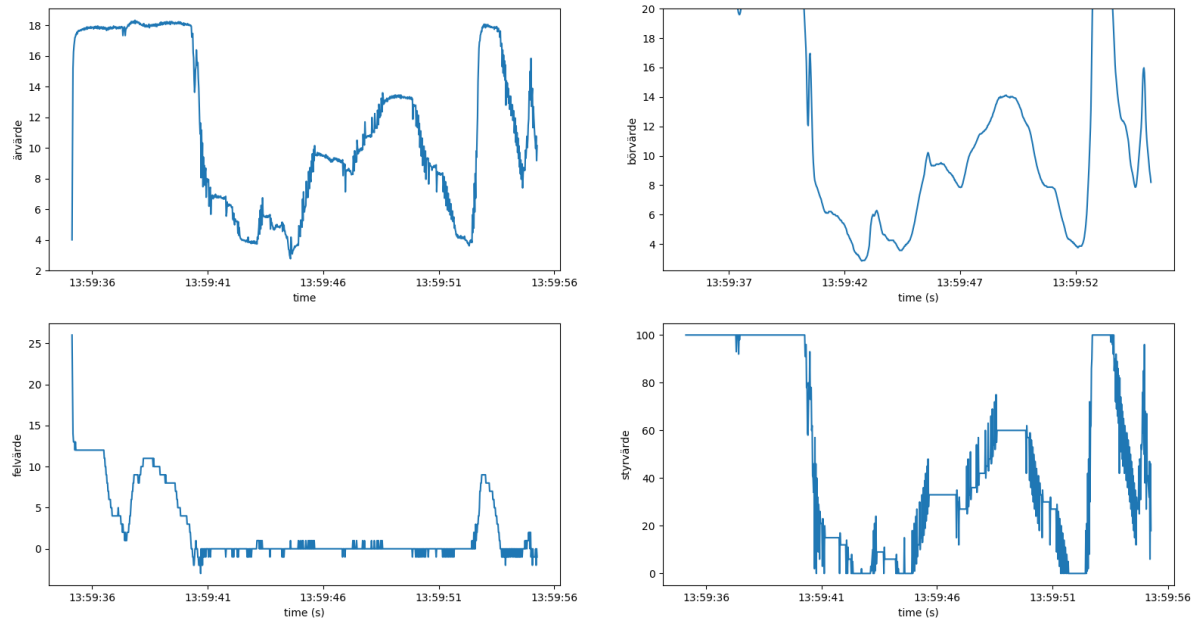
$$\text{Ivärde}(k) = \text{Ivärde}(k - 1) + (h * \text{felvärde}(k))$$

$$\text{Dvärde}(k) = \frac{\text{felvärde}(k) - \text{felvärde}(k - 1)}{h}$$

( $k$ ) är nuvarande värde och ( $k - 1$ ) är föregående värde. I Figur 3.6 visas simulinkmodellens reglering och i Figur 3.7 visas systemets faktiska reglering.



Figur 3.6. PID-regleringen enligt simulinkmodell Mux1 = felvärdet, Mux2 = ärvärdet, Mux3 = börvärdet



Figur 3.7. PID-regleringen i riktiga systemet

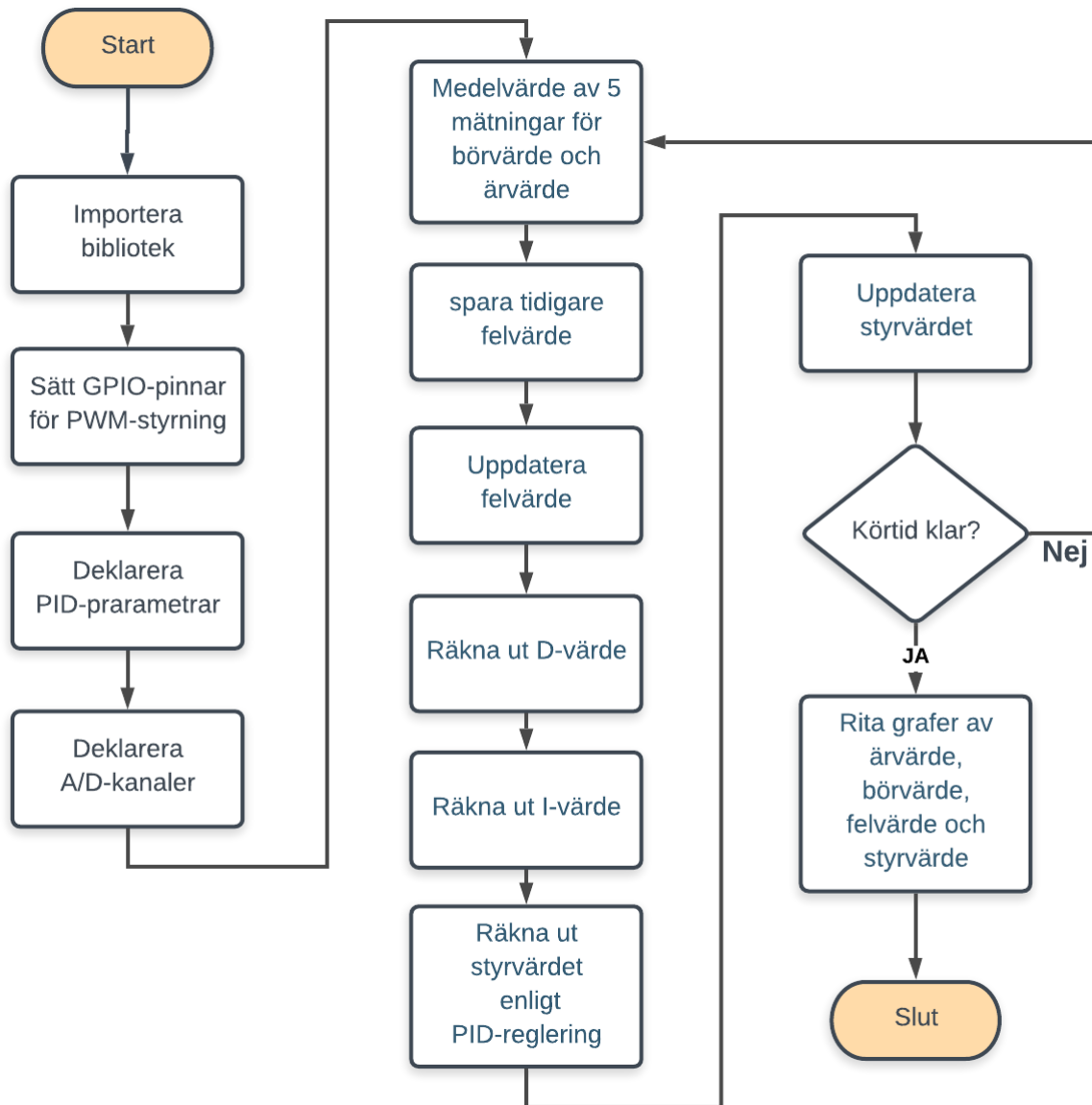
### 3.2.3 Filtrera störningar

Det blev en del störningar av omgivningsljus, framförallt av lysrör i rummet. Fotoresistorerna hinner reagera kraftigt på lysrörens blinkningar vilket skapar stora störningar som var tvungna att filtreras bort. Genom att ta medelvärdet på fem mätningar så försvann en stor del av störningarna utan att påverka systemet något avsevärt.



### 3.2.4 Flödesschema av programmet

Figur 3.8 beskriver programmets flöde enligt en bestämd körtid.



Figur 3.8. Flödesschema av regleringsprogrammet

## 4. RESULTAT

Resultatet blev en prototyp med en datorskärm försluten i en låda som fungerar som simuleringsrum där datorskärmen bli det virtuella fönstret. Lådan är klätt i vitt papper på insidan för att representera ett rum med vita väggar och reflektera ett liknande ljus. Datorskärmen har så låga ljusinställningar som möjligt för att det inte ska störa resten av systemet när det blir mörkt då skärmens ljusstyrka inte regleras. Utanpå lådan sitter en kamera och en fotoresistor för att mäta ljusstyrkan från referensfönstret.

I Figur 3.7 visas regleringen av systemet tillsammans med PID-regleringen. Det har ett begränsat intervall det kan reglera inom då LED-lamporna får matas med max 3W. Detta gör att systemet endast kan reglera ett relativt litet intervall från ca 3–4 % till ca 18 % av vad fotoresistorerna reagerar på. Detta förklarar också felvärdeskurvan som ibland blir tydligt större när det är ljusare än vad systemet kan tillföra. I sitt begränsade intervall så reglerar PID-regulatorn dock systemet bra. Ärvärdeskurvan följer börvärdeskurvan väldigt tydligt och så länge börvärdet är inom det reglerbara intervallet så är felvärdet nästan alltid noll.

På grund av den begränsade ljusstyrkan från LED-modulerna och att färgtemperaturen inte beaktas blev "Fönstret" inte helt övertygande. Det kan dock fortfarande vara en trevlig prydnad men kräver mer arbete om det ska kunna ersätta ett fönster eller användas i kommersiellt syfte.

När det finns möjlighet till ett vanligt fönster är det en klar rekommendation jämfört med ett virtuellt fönster ur miljösynpunkt. Ett vanligt fönster drar ingen ström och kräver inte lika mycket material för att sätta upp men släpper dock igenom en hel del energi jämfört med en vägg. Tanken är att det virtuella fönstret ska sättas upp där vanliga fönster inte är möjliga att ha. Om det virtuella fönstret blir tillräckligt bra för det så skulle det kunna bli mycket mer attraktivt att bo under jorden då man skulle kunna få utsikter över helt andra områden än det som skulle vara utanför ett vanligt fönster. Detta skulle kunna leda till mer utrymme ovan jord för skog och jordbruk.

## 5. SLUTSATS OCH DISKUSSION

Ett av de största problemen i projektet var att LED-lamporna inte kan leverera mer än ca 20% av vad fotoresistorn kan ta upp jämfört med dagsljus som kan leverera mer än 100% av vad fotoresistorn kan visa. Detta skulle kunna förhindras med andra typer av sensorer och eventuellt flera olika typer av fotoresistorer som hanterar olika ljusstyrkor. Perspektivproblemet är ett problem som skulle ta mycket tid och energi att lösa. Hur det ska lösas med flera personer som betraktar fönster samtidigt hade krävt någon mycket mer avancerad lösning. Solljus infaller genom ett fönster i olika vinklar på olika tider på dygnet vilket inte heller beaktades i projektet men som hade klart påverkat upplevelsen.

Det testades en del olika sätt att ändra på ljusstyrkan. Försökte med en annan enkortsdator än RbPi (Odroid XU4) att ändra på skärmens inställningar för att ändra i färg och ljusintensitet men märkte att det skulle kräva mer tid och arbete än vad som fanns till förfogande.

Ett annat sätt var att med en mediaspelare spela upp det kameran filmade och ändra automatiskt i mediaspelarens effekter som "saturation", "gamma", "contrast", "brightness", osv. Detta testades med VLC och FFplay. Detta kanske skulle kunna ha fungerat men även här kändes det som att det skulle ta för mycket tid och arbete då det blir lite för långt ifrån mekatronikämnet för att jag ensam ska kunna lösa det inom rimlig tid.

Det virtuella fönstret från detta projektet blev mer likt produkterna från Atmoph än de från Coelux i avsnitt 2.2, på det sätt att fönstret mer blir en prydnad för utblick än i att det ger effektiv belysning. I nuläget är detta projektet klart billigare än både Atmoph och Coelux då en vanlig datorskärm kan skaffas till klart lägre pris. En RbPi kostar några hundralappar. LED-modulerna kan man köpa olika typer av, dessa kostade ca 100 kr per modul, men om man köper individuella LED-lampor och inte en modul med flera blir det billigare. Fotoresistorerna kostar bara någon krona stycket.

### 5.1 Vidare utveckling

Det finns mycket som kan göras för att ljuset ska bli mer likt dagsljus. Dagsljus kan variera i många olika färgtemperaturer, framförallt vid gryning och skymning. Detta hade gått att uppnå med hjälp av fler olika färgtemperaturer på lamporna och styra varje lampa var för sig och då använda sig av färgsensorer som skulle kunna mäta ungefär vilka våglängder av ljus var starkast och reglera utifrån det. Med lite annan utrustning, framförallt starkare och bättre lampor hade större intervall för ljusstyrka uppnåtts. Skärmens inställningar med ljusstyrka, kontrast och färg hade nog varit väldigt viktigt att ändra och reglera på samma sätt som med lamporna för att upplevelsen ska bli någorlunda verklig.

Att utveckla så att skärmen anpassade perspektivet för betraktaren hade man kunnat göra genom att ha exempelvis flera kameror bredvid varandra som skapar en panoramabild som systemet sedan visar delar av för betraktaren beroende på betraktarens position. Positionen på betraktaren hade man kunnat identifiera med en värmesensor för att se var i rummet betraktaren är. Detta skulle förstås då inte fungera om det finns fler betraktare eller om det finns något annat i rummet som skulle kunna misstas för betraktaren om det har samma

temperatur som betraktaren. En rörelsesensor med det hade kunnat hjälpa till att utesluta det som inte har rört på sig.

Med lentikulärt tryck kan det fås en bild som visar olika bilder beroende på betraktarens position genom att linser över bilderna bryter ljuset så att endast en av bilderna syns från den position betraktaren har. Hade en skärm fungerat på samma sätt så hade flera betraktare kunnat se från olika vinklar samtidigt och se genom det virtuella fönstret i deras respektive korrekta vinkel. Huruvida effekten hade gett ett naturligt utseende eller inte är svårt att säga, många bilder med lentikulärt tryck enligt egen erfarenhet ger konstiga bilder om de inte syns från rätt vinkel. Att ta fram en datorskärm som var anpassad efter detta hade tagit väldigt mycket tid och energi men det skulle kunna vara ett alternativ för vidare utveckling.

Ljusets infallsvinkel genom fönstret hade man kunnat lösa genom att vinkla LED-lamporna i den vinkel som ljuset ska gå genom att ha en stegmotor eller liknande som vrider dem som tänkt. De måste bara monteras på ett sätt så att de inte blockerar eller blir blockerade av datorskärmen vilket skulle kunna bli ett problem.

## REFERENSER

- [1] Folkhälsomyndigheten, "folkhalsomyndigheten.se," 21 12 2017. [Online]. Available: <https://www.folkhalsomyndigheten.se/publicerat-material/publikationsarkiv/l/ljus-och-halsa/>. [Använd 27 03 2019].
- [2] J. Jacobson, "cepro.com," 11 03 2016. [Online]. Available: [https://www.cepro.com/article/coelux\\_the\\_40000\\_fake\\_skylight\\_everyone\\_will\\_want](https://www.cepro.com/article/coelux_the_40000_fake_skylight_everyone_will_want). [Använd 28 03 2019].
- [3] Atmoph, "Atmoph.com," Atmoph, [Online]. Available: <https://atmoph.com/en/>. [Använd 28 03 2019].
- [4] RASPBERRY PI FOUNDATION, "RaspberryPi.org," RASPBERRY PI FOUNDATION, [Online]. Available: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>. [Använd 29 11 2018].
- [5] R. T. Co, "Runcam," Runcam Technology Co, [Online]. Available: <https://shop.runcam.com/runcam2/>. [Använd 18 10 2018].
- [6] M. T. Inc, "MCP3004/3008 2.7V 4-Channel/8-Channel 10-Bit A/D Converters with SPI Serial Interface," 2008. [Online]. [Använd 06 12 2018].
- [7] I. m. GmbH, "LINEAR RIGID ADVANCE LRAML-SW800-24V-28S103-20-IC," 10 2017. [Online]. [Använd 16 11 2018].
- [8] B. Thomas, Modern Reglerteknik, Stockholm: Liber, 2016.

## BILDREFERENSER

Figur 2.1 Perspektiv på skärm .....	4
Figur 2.2 Perspektiv genom fönster .....	4
Figur 2.3 Lentikulärt tryck, olika perspektiv visar olika bilder.....	5
Figur 2.4. Raspberry Pi 3 Model B [4].....	6
Figur 2.5 Pulsbreddsmodulering .....	7
Figur 2.6. Samband mellan överföringsfunktioner och stegsvar [8].....	8
Figur 3.1. Styrvärdet vid steg.....	11
Figur 3.2. Ärvärdet vid steg .....	11
Figur 3.3. Styrvärdet vid olika steg.....	12
Figur 3.4. Ärvärdet vid olika steg.....	12
Figur 3.5. Graf över K-värdet, formel jämfört med uppmätt.....	13
Figur 3.6. PID-regleringen enligt simulinkmodell Mux1 = felvärdet, Mux2 = ärvärdet, Mux3 = börvärdet.....	15
Figur 3.7. PID-regleringen i riktiga systemet .....	16
Figur 3.8. Flödesschema av regleringsprogrammet.....	17

# APPENDIX

## Programkod

```
from gpiozero import MCP3008
import RPi.GPIO as GPIO
import time
import datetime as dt
import matplotlib.pyplot as plt
import matplotlib.animation as animation
import numpy as np
import math
from scipy import signal

#setup pins
GPIO.setmode(GPIO.BCM)
GPIO.setup(4, GPIO.IN)
GPIO.setup(12, GPIO.OUT)

p = GPIO.PWM(12, 100) # channel=12 frequency=100Hz
p.start(0)

#styrvarde = duty cycle 0 - 100%
styrvarde = 20

ivarde = 0
gammalivarde = 0
felvarde = 0

#PID parametrar
P = 10
Ti = 1/30
Td = 0.005
h = 0.01

# Skapa en figur för plotting
fig = plt.figure()
ax = fig.add_subplot(1, 1, 1)
xs = []

ysbor = []
ysar = []
es0 = []
ss0 = []

s0ut = MCP3008(channel=3, device=0) #TSL250 ute
s1ut = MCP3008(channel=4, device=0) #TSL250 ute
s2ut = MCP3008(channel=5, device=0) #fotoresistor ute

s0in = MCP3008(channel=0, device=0) #TSL250 inne
s1in = MCP3008(channel=1, device=0) #TSL250 inne
s2in = MCP3008(channel=2, device=0) #fotoresistor inne

for t in range(0, 1000):
```

```

#Filtrera störningar genom att ta medelvärdet av 5 mätningar
borvarde=0
arvarde=0

for i in range(0,5):
    borvarde = borvarde + s2ut.value*100
    arvarde = arvarde + s2in.value*100

borvarde = borvarde/5
arvarde = arvarde/5

gammalfelvarde = felvarde #spara e(k-1)
felvarde = borvarde - arvarde #uppdatera e(k)
felvarde = math.trunc(felvarde)
dvarde = (felvarde - gammalfelvarde) / h #räkna ut Deriveringsvärdet

ivarde = gammalivarde + (h * felvarde) #uppdatera I(k)
if styrvarde == 100:
    ivarde = gammalivarde
gammalivarde = ivarde #spara I(k-1)

styrvarde = P * (felvarde + (1/Ti)*ivarde + Td*dvarde) #uppdatera
styrvärdet

if styrvarde >= 100: #begränsa styrvärdet så att det inte kan gå över
100 eller under 0
    styrvarde=100
if styrvarde <= 0:
    styrvarde=0

p.ChangeDutyCycle(styrvarde)

# Add x and y to lists
xs.append(dt.datetime.now())

ysbor.append(borvarde)
ysar.append(arvarde)
es0.append(felvarde)
ss0.append(styrvarde)
#es0.append(ivarde) #om du vill kolla ivardet

# Draw plot
plt.title('Light intensity')

plt.subplot(2, 2, 1)
plt.plot(xs,ysar)
plt.xlabel('time')
plt.ylabel('ärvärde')

plt.subplot(2, 2, 2)
plt.plot(xs,ysbor)
plt.xlabel('time (s)')

```



```
plt.ylabel('börvärde')
```

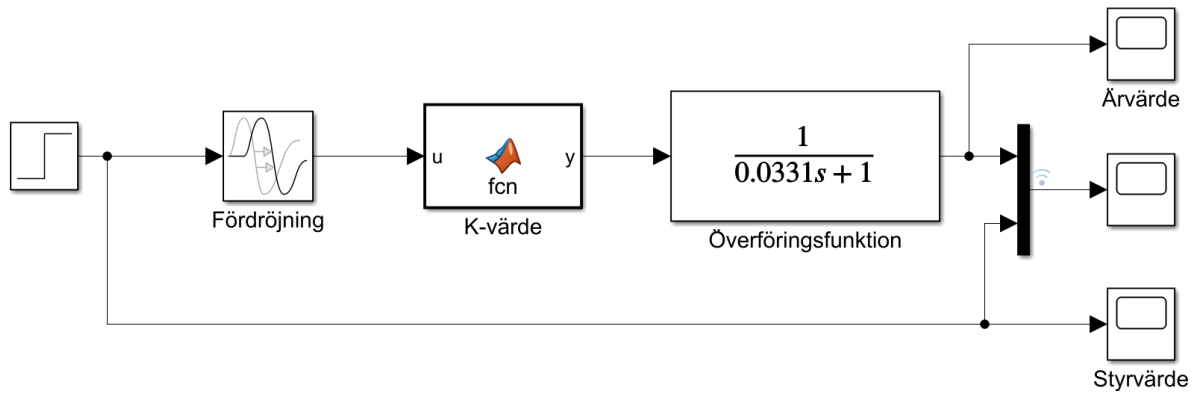
```
plt.subplot(2, 2, 3)  
plt.plot(xs, es0, '-')  
plt.xlabel('time (s)')  
plt.ylabel('felvärde')
```

```
plt.subplot(2, 2, 4)  
plt.plot(xs, ss0, '-')  
plt.xlabel('time (s)')  
plt.ylabel('styrvärde')
```

```
# Draw the graph  
plt.show()
```

## Simulinkmodeller

Modell för överföringsfunktionen av processen:



Modell över hela systemet med process och PID-regulator:

