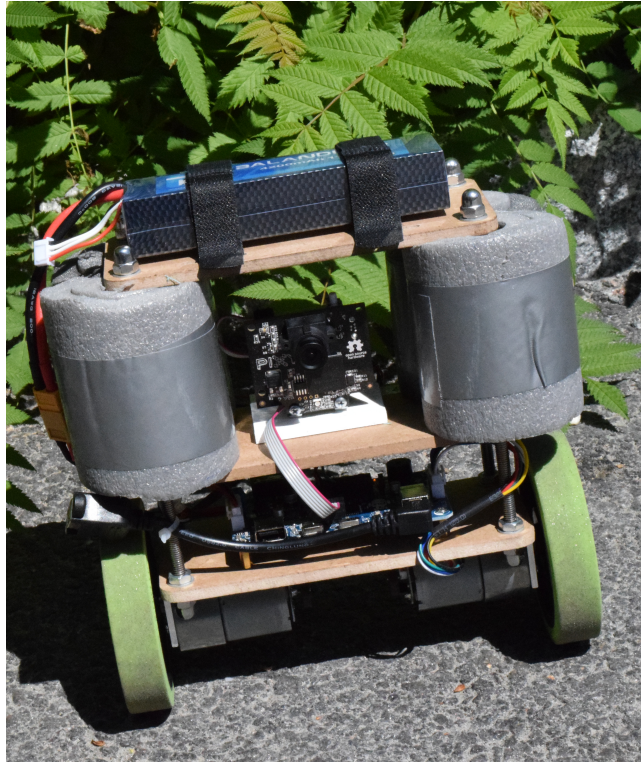




CHALMERS



Utveckling av autonoma robotar för fotbollsspel

Ett kandidatarbete på Chalmers tekniska högskola, SSYX02-16-83

Joel Andersson
Jonas Alexandersson
Jonathan Kammerland
Simon Sundell
Tobias Bergström

Förord

Rapporten är skriven på svenska och arbetet har utförts vid Chalmers tekniska högskola i Göteborg. Den är resultatet av ett kandidatarbete på 15 högskolepoäng under Institutionen för Signaler och System. Vi tackar vår handledare Jonas Fredriksson för hans assistans. Ett speciellt tack även till de anställda på Prototyplabbet för deras behjälplighet under konstruktion och produktion av olika komponenter.

Sammandrag

Grunden till projektet Robotfotboll ligger i en starkt ökande trend av hemautomation. Robotdammsugare, robotgräsklippare och monteringslinjer med mera blir ett allt vanligare inslag i vardagen. Fotboll som spelas av robotar förekommer över hela världen och internationella tävlingar arrangeras årligen vilket bidrar till att utvecklingen inom området fortsätter framåt. Arbetet i det här projektet har till största delen bestått av programmering, men även produktion av ett antal detaljer för sammansättningen av slutprodukten, exempelvis kamerafäste och fotbollsplan.

Projektet använder och begränsas av färdiga komponenter bestående av robotar och kameror för att skapa ett lag om tre självgående robotar som kan spela i en fotbollsmatch. Matchen spelas mot ett lag av samma typ av robotar från ett annat projekt med samma mål, i syfte att utvärdera de autonoma robotarnas förmåga att utföra en mänsklig uppgift. Varje robot styrs av en egen mikroprocessor med begränsad beräkningskapacitet som dock avlastas av kameran och dess kretskort vilket sköter all bildbehandling.

Genom standardiserade prestandatester som utgår ifrån kravspecifikationen kan projektet visa resultat där komponenterna fungerar väl individuellt och till viss del tillsammans.

Det största problemet för att robotarna skulle vara autonoma visade sig som tydligast under själva fotbollsmatchen vilket var kamerans ljuskänslighet samt att färgerna uppfattas olika vid olika ljussättningar. På grund av den relativt billiga hårdvaran i kameran kunde inte robotarna tillräckligt väl lokalisera alla objekt på planen oavsett hur noggrant kalibreringen utfördes. Detta påverkade kraftigt alla robotarnas prestanda under matchen och därmed utgången av den.

Avslutningsvis dras flertalet slutsatser, där fotbollsmatchen är huvudfokus. På grund av den bristande identifieringen hos kameran presterade även resten av komponenterna sämre. Det var väldigt tydligt att robotarna inte på egen hand kan genomföra en fotbollsmatch utan ingripande från människor, eftersom omkullkörningar är väldigt frekventa. Med en förbättrad igenkänning av objekt får mjukvaran mer pålitlig data som den efter bearbetning skickar till resten av roboten.

Abstract

The basis of the project Robot Soccer lies in the rapidly increasing trend of home automation. Robotic vacuum cleaners, lawn mowers, assembly lines and such are to a greater extent a casual element in many peoples everyday life. The work of this project consits mostly of programming but also the production of some parts for the assembly of the final product, such as camera brackets and the soccer field.

The project utilizes and is limited by complete components including robots and cameras to create a team of three autonomous robots which can play a game of soccer. The game will be played against a rival team of the same type of robots from another project with the same goal. The purpose of this is to evaluate the autonomous robots' ability to perform a human task. Each of them is controlled by their own small microprocessor with limited computing capacity. The computer is relieved by the camera and its circuit board which is handling the analysis of incoming pictures.

The biggest problem was most clear while playing the actual game of soccer which was the photosensitivity of the camera and the different perception of colors caused by variations in illumination. The relatively cheap hardware in the camera was one of the reasons the robots could not locate all the different objects well enough despite how carefully the calibration was executed. This affected the robots' performance considerably during the game and thus also the outcome of it.

Finally several conclusions were made, in which the performance of the camera was the main focus. Because of its deficient recongnition of objects the rest of the components performed worse aswell. With improved identification the software would recieve more reliable data which, after processing, is sent to the rest of the robot.

Beteckningar

- **Fleragentsystem** - Ett system av samverkande intelligenta agenter inom en begränsad miljö.
- **USB-dongel** - Universal Serial Bus-dongel. Individuell elektronisk enhet som använder gränssnittet USB för att kommunicera med en värddator.
- **Falsk positiv** - Ett positivt utfall från en avläsning där det sökta ämnet inte finns.
- **SONAR** - En teknik som använder sig av ultraljud för att mäta avstånd till objekt i omgivningen.
- **Färgkod** - Ett sätt för Pixy att identifiera objekt som har tydliga färger. Kan förekomma i färgkombinationer, där flera färger läggs i följd för att skapa en unik färgkod för ett visst objekt.
- **Master** - En enhet som inom Bluetooth-tekniken agerar mittpunkt för upp till 7 slave-enheter.
- **Slave** - Den enhet som kommunicerar med en master-enhet.

Innehåll

1	Inledning	1
1.1	Syfte	2
1.2	Mål	2
1.3	Kravspecifikation	3
2	Matematiska modeller	4
2.1	Modell för avståndsmätning	4
2.2	Modell för styrning i cirkelbana	5
2.3	Modell för reglering	6
2.4	Kalmanfiltrering	7
2.4.1	Kalmanfilter för lutning av robot	7
2.4.2	Kalmanfilter för relativ positionsbestämning	8
3	Hårdvarudesign	9
3.1	Balduino	9
3.2	Pixy	10
3.3	Mikrofon - Analog Sound Sensor SKU: DFR0034	11
3.4	Bluetooth-modul	11
3.5	Kameraplattform	11
3.6	Boll	12
3.7	Rörisolering	12
3.8	Sarg för fotbollsplan	12
3.9	Målbur för fotbollsplan	13
3.10	Lagtröjor	13
4	Mjukvarudesign	13
4.1	Spelstrategier	14
4.1.1	Målvakt	15
4.1.2	Försvarare	16
4.1.3	Anfallare	16
4.1.4	Generella strategier	17
5	Testresultat och analys	18
5.1	Avståndsmätning med Pixy	18
5.2	Precision vid körning av en specifik distans	19
5.3	Körning i en åttaformad bana	20
5.4	Topphastighet raksträcka	21
5.5	Rotation kring egen axel	22
5.6	Körning i en böjd bana	22
5.7	Stabilisering efter störning av robot	23
5.8	Mätning av riktning på objekt relativt robot	25
5.9	Identifikation av objekt med Pixy	26
6	Resultat	27

6.1	Fotbollsmatch mot annat lag	27
6.2	Träningsmatch	28
6.3	Robotens prestanda	29
7	Diskussion	29
7.1	Fotbollsmatch	29
7.2	Robotens prestanda	30
7.3	Analys av mjukvara	31
7.3.1	Balans	31
7.3.2	Målvakt	31
7.3.3	Försvarare	31
7.3.4	Anfallare	32
7.4	Analys av hårdvara	32
7.4.1	Balanduino	32
7.4.2	Pixy	32
7.4.3	Utredning av trådlös kommunikation inom robotlaget	33
7.4.4	Mikrofon	33
7.5	Produktion av övriga komponenter	34
7.5.1	Design av kameraplattform	34
7.5.2	Produktion av fotbollsplan	34
8	Slutsatser	35
8.1	Autonom fotbollsmatch	35
8.2	Precision i rörelsemoment för Balanduino	35
8.3	Objektidentifiering med hjälp av Pixy	36
8.4	Trådlös kommunikation	36
8.5	Förbättra mikrofonens funktionalitet	36
8.6	Projektet som pedagogiskt hjälpmedel	36
9	Vidareutveckling	36
9.1	Trådlös kommunikation	36
9.2	Förslag på sensorer	37
A	Bluetooth	1
A.1	Protokoll	1

1 Inledning

I dagens samhälle blir användandet av robotar allt mer synligt. Robotarna utvecklas konstant och anpassas för att hjälpa människan med att ta sig förbi hinder, öka säkerhet och underlätta vardagen. De har en otrolig potential till att utföra arbeten som för människan hade varit farliga, monotona eller oerhört ansträngande. Robotar äger inte heller samma svagheter som människan, så som den mänskliga faktorn, och arbeten kan därför bli noggrannare utförda vilket utnyttjas mer och mer inom sjukvården, industrin och trafiken. Även hos privatpersoner används autonoma robotar i större utsträckning där robotdammsugare och robotgräsklippare är tydliga tecken på. Denna tillväxt i efterfrågan av autonoma robotar leder till att kraven kunderna kan ställa ökar. Därmed är robotik ett område där utvecklingsmöjligheterna och framtidsutsikterna är goda. För att upprätthålla samt öka intresset för robotiken används flera olika metoder varav en av dessa är robotfotboll.

Den första roboten som kunde spela fotboll togs fram och byggdes av Alan Mackworth, en professor inom datavetenskap vid University of British Columbia [5], tillsammans med sin forskargrupp år 1992. Denna robot låg till grund för utvecklingen av robotfotboll som en av de främsta globala plattformarna för fleragentsystem. Robotfotboll används idag som en standardmiljö för testning av forskningsteorier enligt Mackworth [6], inom överlappande forskningsområden genom att det kan användas som ett medium för utvecklingen av nya fleragentsystem. En fotbollsspelande robot är en autonom, intelligent agent som löser problem och utför uppgifter via samarbete och kommunikation med andra intelligenta agenter och ingår därmed i ett fleragentsystem. Vid Vienna University of Technology har man använt kunskaper inom robotfotboll för utveckling av industriella applikationer. I artikeln "Industrial Applications of Robotsoccer" från konferensen Proceedings 5th MATHMOD [11] beskrivs det hur en robot som kallas för "Robot-Space" är direkt hämtad från området robotfotboll. På grund av att robotfotboll är en relativt billig plattform för testning och utveckling inom flera tekniska områden så som till exempel kollisionsdetektion, bildbehandling och positionskontroll är det ett betydande utvecklingsområde. Den kunskap som fåtts via arbetet med fotbollsspelande robotar, enligt Industrial Applications of Robotsoccer, omsattes till utveckling av robotar för industriella applikationer. Solkraft från rymden är ett sådant projekt vilket använder en vida-reutveckling av en robot som spelar fotboll.

År 1997 grundades FIRA, Federation of International Robot-soccer Association. Syftet med förbundet är, enligt FIRAs hemsida [7], att sprida vetenskapen och tekniken inom robotik till den yngre generationen samt lekmännen. För att lyckas med sitt mål försöker de samla duktiga studenter och forskare som har olika teknisk bakgrund där verkningsområdena kan vara till exempel robotik, kommunikation, intelligent kontroll, bildbehandling, data och IT samt artificiell intelligens. Som framkommer i stadgarna till FIRA [8] sågs robotfotboll som en bra tillämpning för forskning inom dessa områden och genom att få dessa personer att arbeta mot ett gemensamt mål som består av flera olika tekniska aspekter främjas utvecklingen av bland annat fleragentsystem. Den första internationella turneringen inom spor-

ten robotfotboll hölls år 1996 och sedan dess hålls evenemanget årligen. Idag finns det under FIRA ett flertal olika klasser som deltagarna i turneringen kan tävla i där främst konstruktionen skiljer sig åt mellan klasserna. Detta projekt utgår ifrån robotar av klassen RoboSot, i vilken varje lag består utav en till tre stycken robotar. Enligt reglerna för klassen RoboSot [10] ska robotarna vara helt autonoma, dock är det tillåtet med en semi-autonom robot där en värddator får användas för att processa information som hämtas in ifrån de kameror och/eller sensorer som robotarna är utrustade med. Eftersom robotarna är autonoma ställs det höga krav på bland annat hårdvarudesign, kommunikation och sensorer.

Det här projektets huvudfokus ligger på att hantera styrning av robotar helt autonomt, vilket betyder att robotarna ska utföra uppgifter på egen hand och inte vara beroende av en kontrollenhet. Den enskilda robotens styrning grundar sig i det klassiska reglertekniska problemet med en inverterad pendel, se [9]. En inverterad pendel utgör ett instabilt system när det inte aktivt regleras. Det vill säga att pendeln inte kan balanseras upprätt och faller helt enkelt ner när pendelns rörelse inte kompenseras. Mer detaljerad förklaring fås av artikeln ”Inverted Pendulum: System Modeling” [14]. En vanlig pendel utgör ett stabilt system då dess upphängningspunkt befinner sig ovanför pendelns massa, medan en inverterad pendel istället har sin massa ovanför sin upphängningspunkt. Som man kan se i en rapport från Kungliga Tekniska Högskolan [1] så krävs det att upphängningspunktens position anpassas i förhållande till vinkeln mellan pendeln och jämviktsläget för att den inverterade pendeln ska kunna balansera.

Problemet som ska lösas består inte endast av den tekniska utmaningen att styra en sådan robot, utan består även av svårigheten att koppla samman den med annan teknik för robotens interaktion med en boll, spelplan, mål och andra robotspelare. För detta krävs någon form av sensorer som kan läsa av och hämta information från omgivningen. Här finns flertalet alternativ så som radar, sonar, laser och kamera m.m. Projektets robotar kommer utrustas med en optisk sensor i form av kameran Pixy [12]. På grund av att fokus ligger på reglering, styrning och utformning av samspel mellan robotar kommer en färdig robot att användas, en så kallad Balanduino [3]. Denna robot fungerar som en inverterad pendel som nämnt ovan och kräver därmed kontinuerlig reglering för att kunna balansera.

1.1 Syfte

Rapporten beskriver utveckling och utvärdering av autonoma robotars förmåga att spela fotboll.

1.2 Mål

Projektets mål är att spela en fotbollsmatch med ett lag bestående av tre autonoma robotar.

1.3 Kravspecifikation

Den framtagna kravspecifikationen syftar till att tydliggöra hur väl roboten ska fungera. De uppsatta kraven specificerar i första hand hur väl roboten ska kunna utföra moment som anses nödvändiga för att uppnå det uppsatta målet.

Krav:

1. Roboten ska ha en topphastighet på minst 1 m/s när den kör rakt framåt.
Syfte: Att minimera den tid målvakten har på sig att blockera ett skott mot mål.
2. Roboten ska vid körning rakt framåt ha en avvikelse på högst ± 10 cm ifrån måldistansen.
Syfte: Kunna orientera sig på planen och möjliggöra strategisk positionering.
3. Roboten ska kunna utföra en rotation kring sin egen axel med högst 10° absolut avvikelse ifrån önskad rotation vid en rotation på maximalt 360° .
Syfte: Roboten ska kunna köra i given riktning för strategisk positionering utan yttre kalibrering.
4. Roboten ska kunna köra i en böjd bana med given svängradie med högst 10% relativ avvikelse.
Syfte: Att behålla hastighet under positionering istället för att åka rakt, bromsa och svänga.
5. Roboten ska stabilisera till stillastående position efter en störning på högst 1 sekund.
Syfte: Ska snabbt kunna återgå till vad den ska utföra efter att den blivit påverkad av en störning.
6. Roboten ska kunna stabilisera sig efter en störning på en lutning av 10° orsakad av knuff.
Syfte: Kunna klara av mindre kollisioner utan att ramla.
7. Med hjälp av Pixy ska roboten kunna identifiera 6 stycken olika objekt samtidigt på ett avstånd av 2 meter.
Syfte: För att roboten med hjälp av kameran ska kunna identifiera boll, sarg, 2 olika mål och lagspelare.
8. Roboten ska kunna mäta avstånd till boll med högst 5 cm felmarginal vid ett avstånd på 1 m.
Syfte: Det ska vara möjligt planera manöver utifrån bollens position när den är i närheten.
9. Roboten ska kunna avgöra riktningen på objekt relativt roboten med en avvikelse på högst 10° med hjälp av Pixy.
Syfte: Det är viktigt att roboten ska kunna skapa sig en rimlig uppfattning om sin omgivning. Ett exempel är att kunna undvika sarg men ändå kunna spela nära den.

Önskemål:

1. Det bör finnas någon form av kommunikationsmöjlighet mellan robotarna.
Syfte: För att kunna utbyta information och skapa ett mer effektivt lagarbete.
2. Roboten bör kunna identifiera boll och mål och därefter skjuta bollen i målets riktning.

Syfte: Detta är av grundidéerna för att uppnå någon form av smart spelstrategi.

3. Roboten bör kunna identifiera det egna målet, positionera sig framför det samt skjuta bort bollen om den kommer nära.

Syfte: Det är en enkel strategi för att hindra motståndarna ifrån att göra mål när vi tappar offensiven.

2 Matematiska modeller

I detta avsnitt presenteras modeller av olika problem som ska lösas för att roboten ska kunna spela fotboll. Dessa modeller innefattar analys av den data som Balanduinion tar emot från Pixy. Filter och felhantering av data från sensorer samt styrning och balansering av Balanduino roboten.

2.1 Modell för avståndsmätning

Pixy kan mäta storlek och position på ett kalibrerat föremål i dess synfält. En rektangel fås ut som höjd och bredd med (x,y) position mitt i objektet relativt nedre vänstra hörnet.

Avståndet till olika objekt kan då ganska enkelt beräknas approximativt. Kamerans bildplan blir en projektion av rummet i synfältets riktning där ett identifierat föremål tar upp en "area"/del av synfält. Låt detta område kallas Ω . Om samma föremål då finns på olika områden i rummet med ytan S_1 & S_2 , sett från samma godtyckligt origo ska de ha samma area.

$$Area(S_1) = \iint_{S_1} dS_1 = \iint_{S_2} dS_2 = Area(S_2) \quad (1)$$

Antag då att denna yta kan approximeras som ett ursnitt på en sfärisk yta under villkoret att snittet nästan blir en platt yta. Det innebär att snittet måste ta upp en liten rymdvinkel¹ Ω sett från origo, alltså $\Omega \ll 4\pi$. I rympolära koordinater blir då uttrycket

$$r_1^2 \Omega_1 = \iint_{\Omega_1} r_1^2 \sin\theta d\phi d\theta \approx \iint_{\Omega_2} r_2^2 \sin\theta d\phi d\theta = r_2^2 \Omega_2 \quad (2)$$

Genom att kalibrera hur stor area eller del av synfältet Ω_1 samma föremål tar upp vid avstånd r_1 kan ett avstånd r_2 bestämmas enligt

$$r_2 \approx r_1 \sqrt{\frac{\Omega_1}{\Omega_2}} \quad (3)$$

om Ω_2 mäts. Felet kommer sig av skillnaden i konvexitet på ytorna då arean antas vara som den på ett stycke av en sfär. Analogt blir mätning i en dimension $r_2 \approx r_1 \frac{\theta_1}{\theta_2}$,

¹ Rymdvinkel är 2D vinkel som kan visualiseras som hur bred en kon är. Som referens är alltså fulla rymdvinkeln 4π eller mantelarean på enhetsfären.

där θ är utsträckning på synfältet längs en axel. Hur väl detta fungerar återfinns i testet i kapitel 5.1.

2.2 Modell för styrning i cirkelbana

För att få roboten att köra i en krökt bana används olika cirkelbanor med varierande radie. Längden på cirkelbågen fås fram genom det enkla sambandet:

$$L = r\theta \quad (4)$$

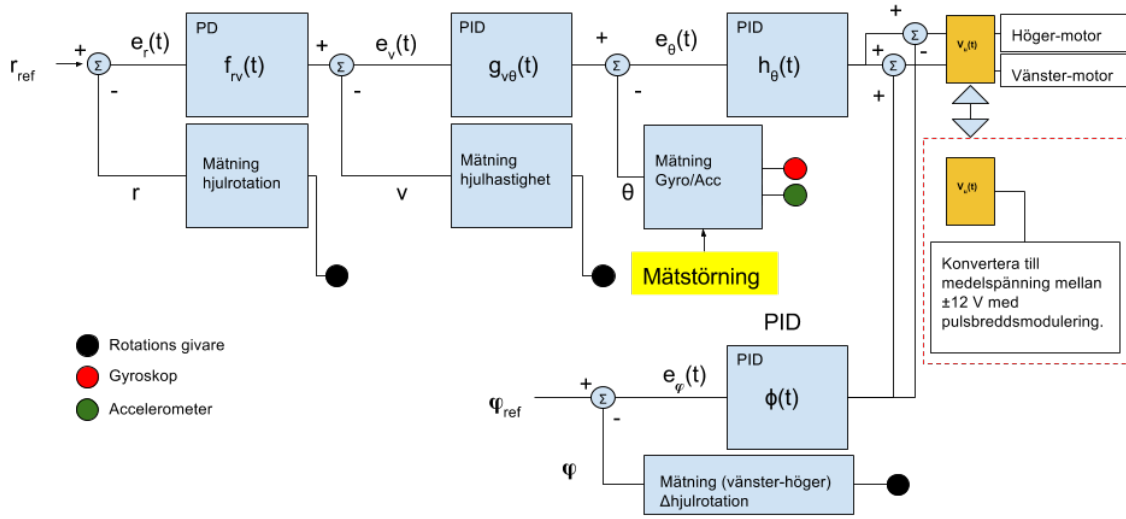
Där L är längden av cirkelbågen, r är cirkelns radie och θ är vinkeln roboten vridit ifrån startpositionen. Genom att bestämma en fix radie och mäta hur långt roboten har kört L kan vi beräkna vinkeln θ som roboten ska stå i enligt $\theta = L/r$. Om roboten kan ställa in sig efter denna vinkel följer den en cirkel. Implementation finns i avsnitt *genomförande* och precision i rörelse i *resultat*.

$$Length(C) = \int_C dl = \int_{\theta_0}^{\theta} r(\phi)d\phi = r\Delta\theta \quad (5)$$

om C är en cirkel (konstant radie r).

2.3 Modell för reglering

Vid mätning av position, hastighet och vridning används rotationsensorer, se *hårdvarudesign*, på båda hjulen. Lutningen och förändringen i lutning av pendeln mäts med hjälp av en accelerometer och ett gyroskåp. Båda måste användas för att accelerometern är för brusig att mäta direkt på och gyroskopet driver under körning. Med accelerometern ställs en offset för gyroskopet in med jämna mellanrum. På grund av mätstörningar filtreras storheterna θ och $\dot{\theta}$ med ett kalmanfilter. Representationen av styrningen kan ses i blockschemat i figur 1.



Figur 1: Övergripande reglertekniska designen för styrning. Där kaskadkopplingen av regulatorerna f_{rv} , $g_{r\theta}$ och h_θ reglerar storheterna position(r), hastighet(v) och acceleration/lutning(θ) i radiell riktning respektive. Notera då att referenssignal propagerar framåt i schemat där en regulator sätter börvärdet till nästa. Reglerfelet $e_X(t)$ fås med hjälp av givare för storheterna r, v, θ genom $e_X(t) = X_{ref} - X(t)$. Parallellt ligger regulatorn $\Phi(t)$ som styr rotation kring vertikal axel genom att mäta skillnaden mellan vänster och höger hjulrotation. Utsignalen från Φ lägger på en skillnad mellan höger och vänster motor med syfte att vrida roboten. Mätstörningen i accelerometern är stor nog att hänsyn måste tas till den genom att filtrera signalen med ett kalmanfilter.

Styrningsprocessen i figur 1 använder tidsdiskreta *PID*-regulatorer enligt:

$$P(t_k) = K_p e(t_k) \quad (6)$$

$$I(t_{k+1}) = I(t_k) + K_I h e(t_k) \quad (7)$$

$$D(t_k) = \frac{1}{T_f + h} (T_f D(t_{k-1}) + K_d (e(t_k) - e(t_{k-1}))) \quad (8)$$

$$utsignal(t_k) = P(t_k) + I(t_k) + D(t_k) \quad (9)$$

Där h är tidssteget mellan iterationer, T_f filtertiden, e reglerfelet och K_X PID parametrarna.

2.4 Kalmanfiltrering

Robotpositionering, balans och avståndsmätning är alla påverkade bristfällig mätutrustning och yttre störningar. Ett av de svåraste problemen är att hantera osäkerheten som finns i given mätdata. Om roboten inte vet var exempelvis bollen befinner sig med tillräcklig noggrannhet blir det extra svårt att fatta bra beslut. Kalmanfiltret är en statistisk metod som kombinerar information från olika källor för att skapa en bättre approximation av tillståndet på det man försöker mäta, se [16]. I ett diskret kalmanfilter kan man skriva filteralgoritmen till en process på följande sätt, enligt [2, s. 137].

Förutse nästa:

$$\hat{x}_{k|k-1} = F_k \hat{x}_{k-1|k-1} + B u_k \quad \text{Tillstånd}$$

$$P_{k|k-1} = F_k P_k F_k^T + Q_k \quad \text{Tillståndets kovarians}$$

Uppdatera:

$$y_k = z_k - H_k \hat{x}_{k|k-1} \quad \text{Mätning residual}$$

$$S = H_k P_{k|k-1} H_k' + R \quad \text{Mätningens kovarians}$$

$$K_k = P_{k|k-1} H_k' S^{-1} \quad \text{Optimal kalman förstärkning}$$

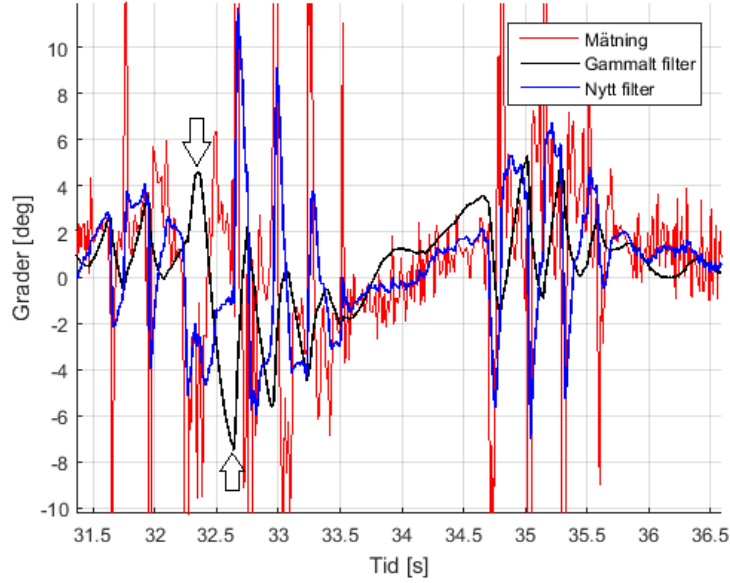
$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k y_k \quad \text{Tillstånd}$$

$$P_{k|k} = (I - K_k H_k) P_{k|k-1} \quad \text{Tillståndets kovarians}$$

Där $x_k|\hat{k} - 1$ är tillståndsvektorn i iteration k givet föregående mätning $k-1$, u är en kontrollvektor för processen, F och B är avbildningsmatrisen till tillståndsrummet för \hat{x} och u respektive. Mätningen z_k av en variabel antas vara normalfördelad kring det sanna värdet x_k enligt $z_k = H_k x_k + v_k$, där v_k är mätstörningen och H_k är avbildningen från tillståndsrummet till det observerade rummet. Exempelvis om man söker att mäta positionen till något har man tillståndsvektorn $[x_1, x_2]'$. Då skulle kontrollvektorn u_k vara hur mycket en robot rört sig i tillståndsrummet mellan mätningar. Q och R är process- och mätvariansen man måste anpassa för sin modell. För en mer utförlig beskrivning och bevis av ekvationerna, se [2].

2.4.1 Kalmanfilter för lutning av robot

Vid reglering av robotens vinkel, θ , från jämviktläget behöver man filtrera signalen från accelerometern då den är brusig, se figur 1. Det implementerade filtret som kom med roboten har problem att hänga i snabba ändringar och tenderar att hamna ur fas när man ställer regulatorerna särskilt aggressivt, se figur 2. I figur 2 filtrerar vi tillståndsvektorn $\hat{x} = [\theta, \dot{\theta}]'$ genom att kombinera accelerometer- och gyroskopdata. Man låter i båda fallen vinkeln man kan lösa ur accelerometern vara den variabel vi mäter, z_k . Gyroskopvärdet används som kontroll u_k . Skillnaden ligger i implementation och val av andra parametrar Q, R och H .



Figur 2: Olika kalmanfilter på samma mätdata. Notera att mätdata är framtagen då roboten matas med det gamla filtret. Det gamla filtret hänger inte med i snabba förändringar och verkar därför hamna ur fas med mätdatan, se pilar.

2.4.2 Kalmanfilter för relativ positionsbestämning

Ortsvektorn till ett föremål från roboten fås genom planpolära koordinater enligt:

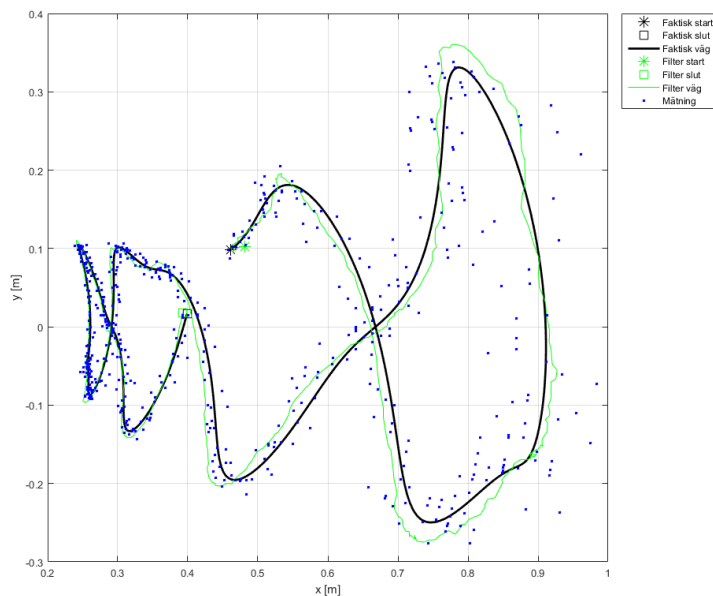
$$\vec{r} = r\hat{r} = r\cos(\phi)\hat{x} + r\sin(\phi)\hat{y} \quad (10)$$

där r är avståndet i meter till föremål, ϕ vinkeln relativt robot, och \hat{x}, \hat{y} och \hat{r} är Ortsvektorer i kartesiska respektive radiella koordinater. Det roboten kan mäta med kameran är avstånd och vinkel till föremål, samt var den själv kör. Tillstånden som filtreras är då $[r, \dot{r}]$ och $[\phi, \dot{\phi}]$ parallellt i olika filter. För att sedan konvertera till kartesiska koordinater som i figur 3. Exempel på ett sämre filter då en boll rullas för hand i en åtta finns i figur 17 i bilagor.

Man kan då få hastighetvektorn mellan två mätpunkter enligt

$$\vec{v} = \frac{\Delta r\hat{r} + \Delta x\hat{\phi}}{\Delta t} \quad (11)$$

om stationärt origo antas. Där Δr är skillnaden i distans i den radiella riktingen \hat{r} och Δx hur långt föremålet har rört sig på synfältet i horisontella riktingen $\hat{\phi}$ vinkelrätt mot \hat{r} .



Figur 3: Simulerad kalman filtrering av position hos boll relativt robot. Roboten befinner sig i origo och ser längs med x-axeln. Notera att process- och mätvarians är uppskattad från riktiga mätningar.

3 Hårdvarudesign

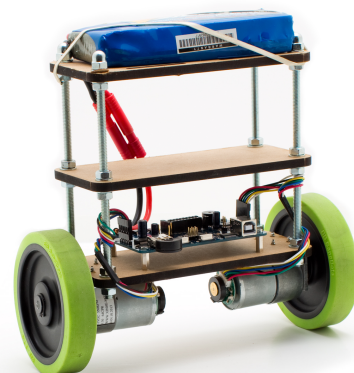
Detta kapitel kommer att beskriva de olika komponenter som ingår i det slutgiltiga systemet. De olika typer av komponenter som kommer att ingå har genomgått en demokratisk diskussion som baseras på de krav som fastställts för vad robotarna ska kunna göra i sitt slutgiltiga stadium. Vid projektets start bestämdes det att robotarna ska fungera som en inverterad pendel och då det fanns tillgång till flera Balanduinorobotar föll valet snabbt på dessa. De väsentliga komponenter som vi anser viktiga för att få robotarna att interagera med varandra och omgivningen för att möjliggöra en match i fotboll beskrivs därför i det här kapitlet.

3.1 Balanduino

Robotarna som används är så kallade Balanduinorobotar vilka beskrivs i [4], se även figur 4. Källkoden är öppen och den är kompatibel med Arduino.

Specifikationer:

- Dimensioner: 241 mm × 216 mm × 98 mm
- Vikt: ~1350g
- Mikrokontroller: Atmel 8-bit ATmega1284P
 - 128 kB Flash
 - 16 kB SRAM
 - 4 kB EEPROM
 - 1x UART
 - 4x PWM utgångar
 - 1x Avbrott
 - 5x Analoga ingångar
 - 1x I2C gränssnitt
- MPU6050 6-axlig sensor
 - 3-axlig accelerometer
 - 3-axligt gyroskop
- FT230X USB Serial chip - möjliggör programmering av robot via USB-port
- Maxim MAX3421E USB Host controller
- 2 st 12V DC-motorer
- 4200 mA 11.1 V 3-cell LiPo battery



Figur 4: *Balanduino robot av TKJ electronics.*

3.2 Pixy

För identifikation av objekt används en kamera som heter Pixy [12], se även figur 5. Den är enkel att använda och har öppen källkod och programvara. Det finns färdiga program för bildanalys och kameran identifierar objekt med hjälp av färger. Pixy har en uppdateringsfrekvens på 50 Hz, vilket innebär att bilden uppdateras var 20:e millisekund. Den lär sig nya objekt med endast en knapptryckning då ett objekt hålls framför linsen och lagrar detta som en färgsignatur. Sju stycken unika signaturer kan upptäckas samtidigt och för identifikation av fler objekt används färgkodning. Med hjälp av en algoritm för sammansatta komponenter kan Pixy känna igen olika färgkombinationer och identifiera dessa som olika objekt. De sju unika färgsignaturerna kan därmed kopplas samman till en mängd olika färgkombinationer.

Specifikationer:

- Dimensioner: $53,3 \text{ mm} \times 50,8 \text{ mm} \times 35,6 \text{ mm}$
- Vikt: 27 g
- Processor: NXP LPC4330, 204 MHz, tvåkärnig
- Bildsensor: Omnivision OV9715, 1/4", 1280x800
- Linstyp: Standard M12
- Tillgängliga datautgångar: UART serial, SPI, IC2, USB, digital, analog
- RAM: 264 kB
- Primärminne (Flash): 1 MB
- Strömförbrukning: 140 mA
- Matningsspänning: USB inmatning (5V), oreglerad ingång (6V-10V)
- Arduino-kompatibel



Figur 5: En Pixy från Charmed-Labs.

3.3 Mikrofon - Analog Sound Sensor SKU: DFR0034

En mikrofon används för att skicka signaler till robotarna så de kan starta eller stoppa spelet, se figur 23. Den mäter ljudnivån i omgivningen och passar bra till Arduino vilket gör att den fungerar bra att koppla in i mikrokontrollern på Balanduino.

Specifikationer:

- Matningsspänning: 3,3V-5V
- Gränssnitt: Analogt
- Dimensioner: $22 \text{ mm} \times 30 \text{ mm}$

3.4 Bluetooth-modul

I den utrustning som följer med Balanduino-roboten ingår en Bluetooth-modul i form av en USB-dongel, se figur 20. Denna används för att utreda huruvida robotarna kommer använda sig av trådlös kommunikation för att göra fotbollsspelet mer effektivt.

Specifikationer:

- Dimensioner: $22 \text{ mm} \times 19 \text{ mm} \times 7 \text{ mm}$

3.5 Kameraplattform

För att fästa Pixy och mikrofon på en Balanduino behöver en plattform konstrueras. Detta gjordes genom att skissa en model i CATIA för att sedan skrivas ut med hjälp av en 3D-skrivare. Plattformen som konstruerades tillåter Pixy att fästas i tre olika

lägen, det första med plan (horisontell) fästning och de andra två med en lutning framåt som tillåter kameran att se objekt som ligger närmre bättre. Bakom de tre möjliga lägena för Pixy finns en plan yta för fästning av mikrofon. För en fullständig ritning över kameraplattformen, se figur 18 i appendix.

Specifikationer:

- Kameralägen: 0° , 10° och 15°
- Dimensioner: $60\text{ mm} \times 55\text{ mm} \times 10\text{ mm}$

3.6 Boll

Bollen är av typen softboll. Den väger inte mycket vilket är bra för roboten som skjuter den med hjälp av en knuff, och den har en stark klar orange färg som förklarar identifiering för Pixy.

Specifikationer:

- Skumgummiboll
- Färg: Orange
- Diameter: 150 mm
- Slitstarkt yttermaterial

3.7 Rörisolering

För att skydda robotens hårdvara från kraftiga stötar eller omkullkörningar används rörisolering i två lager som monteras runt robotens metallstänger. De två lagren är en kombination av två olika storlekar av rörisolering, se figur 19.

Specifikationer:

- Material: Polyetenskumplast
- Tjocklek: 13 mm (skumplast)
- Diameter: 42 mm respektive 65 mm

3.8 Sarg för fotbollsplan

Sargen är tillverkad för att det blir mer flyt i spelet då bollen kommer att hålla sig innanför planens gränser. Den består av 12 långa och 8 korta träskivor som binds samman med hjälp av bockade metall-förbindelser på både över och undersida, se figur 21.

Specifikationer:

Långa träskivor:

- Antal: 12 st.
- Material: Plywood
- Dimensioner: $1200\text{ mm} \times 100\text{ mm} \times 10\text{ mm}$

Korta träskivor:

- Antal: 8 st.

- Material: Plywood
- Dimensioner: $600\text{ mm} \times 100\text{ mm} \times 10\text{ mm}$

Raka metall-förbindelser:

- Antal: 22 st.
- Material: Stål & aluminium
- Dimensioner: $12\text{ mm} \times 12\text{ mm} \times 200\text{ mm}$

Metall-förbindelser hörn på 45°:

- Antal: 16 st.
- Material: Stål
- Dimensioner: $12\text{ mm} \times 12\text{ mm}$ (Höjd och bredd, längden varierar)

3.9 Målbur för fotbollsplan

Målburen är konstruerad med stolpar, en ribba och två fötter som hjälper till att hålla målet stående. Det som är noterbart för målburen är dess ribba, vilket är en rektangulär platta som tillåter färger att placeras så att Pixy kan identifiera målburarna på långa avstånd, se figur 22.

Specifikationer:

- Dimensioner målbur: $1000\text{ mm} \times 600\text{ mm} \times 300\text{ mm}$
- Dimensioner för enbart ribba: $1000\text{ mm} \times 400\text{ mm}$

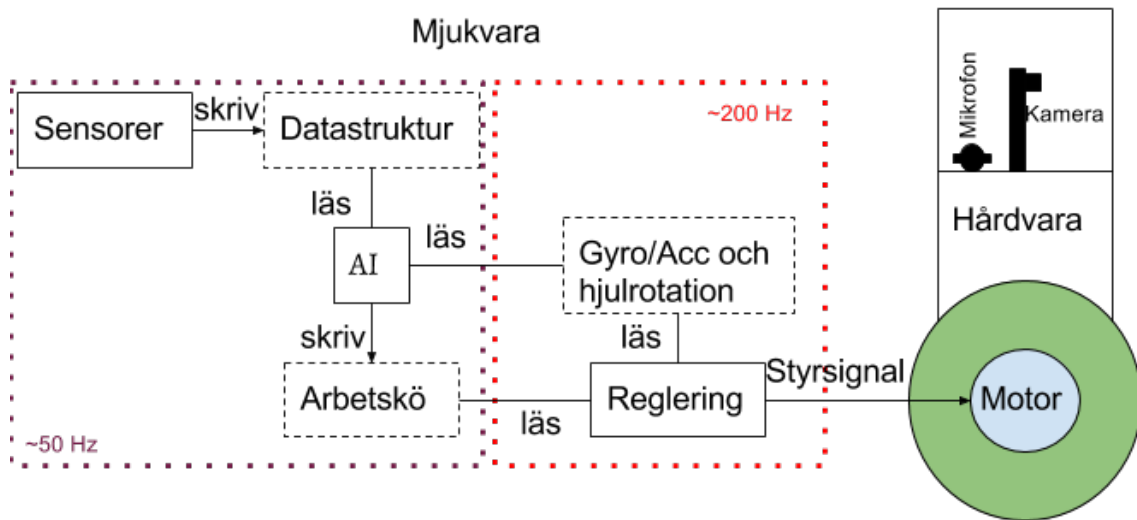
3.10 Lagtröjor

De lagtröjor som robotarna kläs med består av tygremсор som binds hela vägen runt skyddet på vardera sida om kameran. Fördelen med detta är flexibiliteten i att byta färg eller lägga till extra färger om det blir problem, så som svårigheter att identifiera lagkamrater genom Pixy.

4 Mjukvarudesign

I detta avsnitt presenteras implementering av den mjukvara som styr robotarna. All mjukvara är skriven i C/C++ och exekveras på det Arduinobaserade Balanduino kortet, se *hårdvarudesign*.

Även om programmet exekveras sekventiellt är designen av flödet parallellt. Den parallella strukturen är en metod för att implementera de olika delsystem i programmet på ett logiskt sätt. Därför att delsystem som reglering och informationsinhämtning måste arbeta kontinuerligt utan att blockeras av andra instruktioner. Om regleringen blockeras för mycket blir balanseringen av pendeln instabil och slår i marken. Designen av programflödet kan ses i figur 6 och för psuedokod se algoritm 1.



Figur 6: Programflöde för en robot. Regleringen samplar hela tiden för att roboten ska kunna balansera och manövrera. Arbetskön innehåller data till insignalerna för regulatorerna med syfte att utföra olika manöversekvenser. Vi uppdaterar datastrukturen för de olika synliga spelobjekten med kamerans uppdateringsfrekvens 50 Hz. *AI* komponenten är hjärtat i programmet och avgör vilka uppgifter som behöver utföras utifrån given information i datastrukturen.

4.1 Spelstrategier

Likt en riktig fotbollsmatch finns det mängder av möjliga strategier att tillämpa, men lämpliga strategier är begränsade av hårdvarans och den övriga mjukvarans prestanda. Följande strategier har tagits fram med syfte att genom uppdelade spelroller arbeta effektivt som ett lag.

```

Data: game object positions, sound level, robot states, job queue, current
        job, strategy
Result: change job queue.
if job is done then
    | if queue is not empty then
    | | job = get next job in queue;
    | | load job;
    | else
    | | execute strategy;
    | end
end
if close to collision AND is moving then
    | abort moving job;
    | push turn to front of queue;
end
if big tilt disturbance then
    | abort job;
    | push stabilize to front of queue;
end
check strategy break conditions;
execute job;

```

Algorithm 1: AI rutin. Huvuduppgiften är att exekvera satt *strategy* när *job queue* är tom och avbryta exekvering om något går fel.

4.1.1 Målvakt

För att få till ett bra fotbollsspel är det nödvändigt med en agent vars uppgift primärt är att hindra motståndarna ifrån att göra mål. En utav robotarna ska därmed befinna sig inom ett visst avstånd ifrån det egna målet och hindra bollen från att rulla in i det egna målet. I sin artikel, se [17], presenterar vinnarna av RoboCup år 1998, 2000 och 2001 CS Freiburg lösningar på olika problem och sina egna strategier. Deras målvakt har en annan hårdvarulösning än resterande robotar. Kameran är vriden 90° åt ett håll för att enklare kunna röra sig i sidled utmed mållinjen. Detta var ej aktuellt för detta projekt då Balanduinorobeten inte kan röra sig i sidled och därmed försvinner nyttan av att ha en kamera monterad i sidled. CS Freiburgs målvakt använder sig av flera olika färdigheter för att hindra bollen från att gå i mål och den är stationerad till att befinna sig inom ett visst målområde. Då målvakten inte kan se bollen placerar den sig i mitten av målet för att rotera lite åt höger respektive vänster för att söka efter bollen. Rör sig bollen sakta mot målvakten blockerar den bollen genom att stå i vägen. Om bollen däremot rör sig i en hög hastighet fångas bollen upp av målvakten där den beräknas passera mållinjen. Befinner sig bollen vid någon sida av målet vänder sig roboten åt det hörn där bollen befinner sig för att hålla bollen inom synfältet.

Målvakten är den enklaste av de strategier som implementerats och bygger på att agenten alltid befinner sig mellan bollen och det egna målet. Därför att bollen går åt sidan från eget mål när målvakten skjuter. Se figur 7 för positionering och algoritm 2 för strategin.

Data: game object states, job queue
push follow goal to back of queue;
push follow ball to back of queue;
push strike to back of queue;

Algorithm 2: Strategi för målvakt. *Follow* är en funktion som gör att roboten följer efter ett objekt. Notera att sista jobbet *follow ball* gör att om bollen är utanför robotens räckvidd för *strike* så kommer den enbart rotera för att ha bollen centrerat i sitt synfält. Om detta görs under en längre period nås en timeout i jobbet som gör att det avslutas. I det läget åker agenten ut ur målet mot bollen med *strike* jobbet en kort sträcka. Då kan roboten kalibrera om mot målet framifrån vid omexekvering.

4.1.2 Försvarare

Försvararens uppgift är främst att försvara det egna målet genom att förebygga att bollen tar sig in på egen planhalva. Detta sker lite längre fram i planen jämfört med målvakten, se figur 7 för positionering. Strategin presenteras i algoritm 3

Data: game object states, job queue
if *own goal AND ball is visible* **then**
 | find random way around ball;
 | return;
end
if *own goal is visible AND friendly player is not visible* **then**
 | execute goalkeeper;
 | return;
else
 | push follow ball to back of queue;
 | push strike to back of queue;
 | push follow own goal to back of queue;
end

Algorithm 3: Strategi för försvarare. Notera att roboten kommer stå vänd mot det egna målet vid ett visst avstånd i slutet av sekvensen eftersom *follow own goal* läggs till sist i kön. Därmed kan den avgöra om den bör rotera in mot målet som målvakt eller undvika bollen.

4.1.3 Anfallare

Anfallarens primära uppgift är att göra mål och den söker mestadels efter bollen. Roboten strävar efter att hamna i linje med boll och motståndarmål för att kunna

utföra ett skott mot målet. Detta görs med hjälp av olika metoder beroende på övriga förhållanden i spelet. Enligt algoritm 4 kommer agenten som agerar anfallare söka möjlighet för att slå bollen mot motståndarlagets mål eller söka sig mot försvarsposition vid behov.

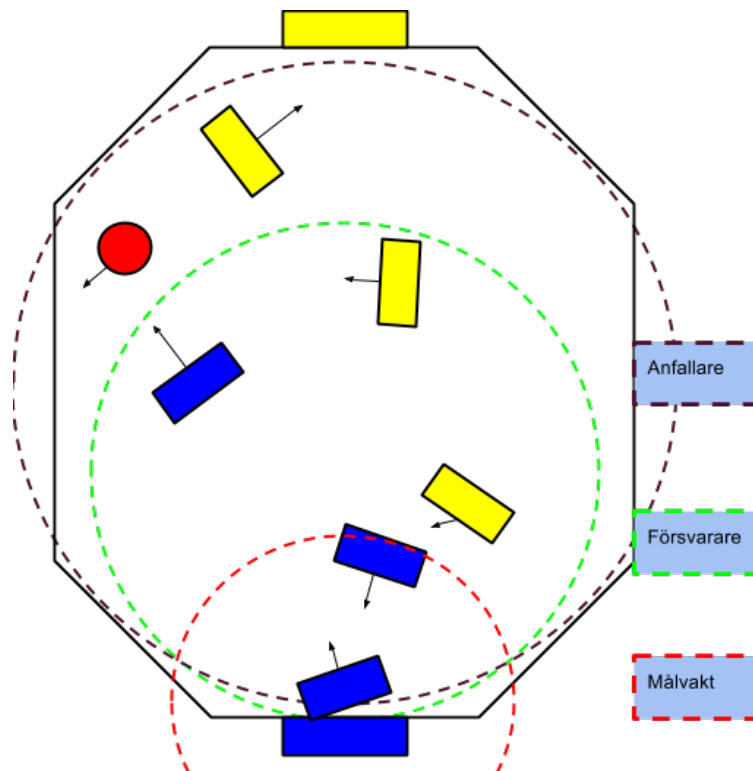
```
Data: game object states, job queue
if own goal AND Ball is visible then
  | walk random;
  | return;
end
if own goal is visible AND friendly player is not visible then
  | execute defender;
  | return;
else
  | push any align for ball and goal or strike to back of queue;
end
```

Algorithm 4: Strategi för anfallare. Notera att det finns slumpmässighet med *walk random* och *any align*. Anfallaren kan rotera och blir försvarare om den inte ser försvar hemmåt.

4.1.4 Generella strategier

Alla robotar har vissa strategier som är gemensamma trots att de har olika spelarroller. Kollisionsdetektion är en sådan strategi som är väldigt viktig för att undvika onödiga stopp i spelet på grund av att robotarna välter. Den bygger på att så fort roboten upptäcker ett hinder i form av sarg, motspelare eller medspelare inom ett visst avstånd ifrån roboten styr den undan beroende på vilken vinkel objektet befinner sig i gentemot roboten. Det finns även en funktion som bygger på ett slumpmässigt rörelsemönster vilken kallas för *walk random*. Den används bland annat för att ta sig ur situationer där roboten fastnar och inte vet vad den ska göra. Detta är en funktion som används av etablerade robotfotbollslag, se [17].

Det finns även möjlighet för robotarna att byta spelarroll under spelets gång. Upp-täcker försvararen att det inte finns någon robot vid det egna målet övertar denna rollen som målvakt för en stund. Särskilt kan anfallaren söka sig nedåt till att bli försvarare eller målvakt om det krävs. Syftet är att ge laget fler möjligheter att skjuta bort bollen från egen planhalva genom att flytta in agenter mellan boll och eget mål. Alla strategier är tidsbegränsade vilket innebär att robotarna återfår sina ursprungliga standardroller efter en viss tid.



Figur 7: En exempelmatch, cirklarna visar det blå lagets möjliga positionering för de olika spelarrollerna.

5 Testresultat och analys

För att verifiera de krav som är uppsatta i kravspecifikationen har standardiserade tester utförts, vilka återfinns i Bilaga 1. Syftet med dessa tester är att få en bättre vetskap om robotens fysiska begränsningar och på så sätt verifiera kraven. Genom dessa tester är det även möjligt att undersöka hur väl den framtagna koden fungerar för att styra roboten genom att se om det förväntade beteendet uppnås. Resultaten diskuteras sedan med en mer ingående analys som kan förklara eventuella avvikande resultat.

5.1 Avståndsmätning med Pixy

Detta test utfördes för att verifiera hur väl avståndsmätningen med Pixy tillsammans med den framtagna mjukvaran fungerar. Detta test används för att undersöka hur väl krav 8 i kravspecifikationen uppfylls.

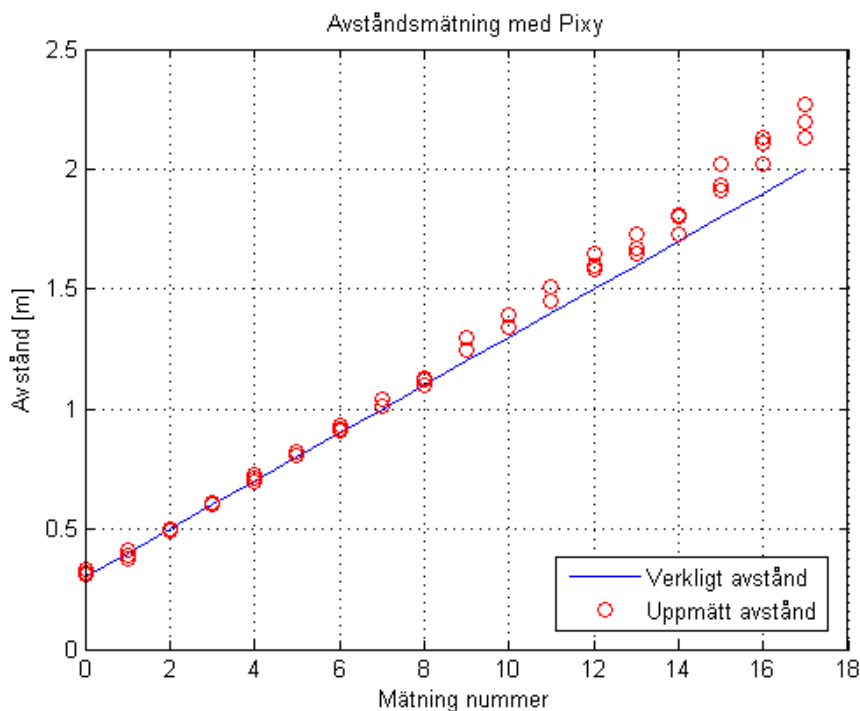
Resultat

Se figur 8 för distansmätning med hjälp av Pixy genom att mäta vertikal utsträckning på kamerans synfält. Felet är mindre än 5 cm upp till ca 1.2 - 1.3 meter vilket innebär att krav nummer 8 uppfylls. På avstånd större än detta uppnås inte längre kraven för att mäta avståndet. Vid över 3 meters avstånd till bollen tappar Pixy den helt och hållet ibland. Objekt av större storlek har Pixy större sannolikhet att

upptäcka. De exakta värdena återfinns i testet "Ävståndsmätning med Pixy" i Bilaga 1.

Analys

Att kameran har problem att få en bra bild av det den försöker mäta är ett stort problem i en verklig miljö. Detta försöks lösas mjukvarumässigt genom rimlighetsbedömning och filtrering. Nämnvärt är att små störningar i vertikal utsträckning ger små skillnader i distans på nära avstånd men på stort avstånd ger samma störning stora distansskillnader. Detta kan tydligt ses i figur 8. Ju större avståndet till bollen är desto längre bort antas den vara vid beräkningen av distansen.



Figur 8: Distansmätning till boll med Pixy. Mätningen har utförts med ett avstånd mellan 30 och 200 cm via 18 olika punkter där 3 värden har tagits vid varje punkt. De röda ringarna i grafen representerar de uppmätta värdena från Pixy och den blå linjen representerar det verkliga avståndet. Notera att mätningarna stämmer bra överens med det verkliga avståndet fram till cirka 100 cm.

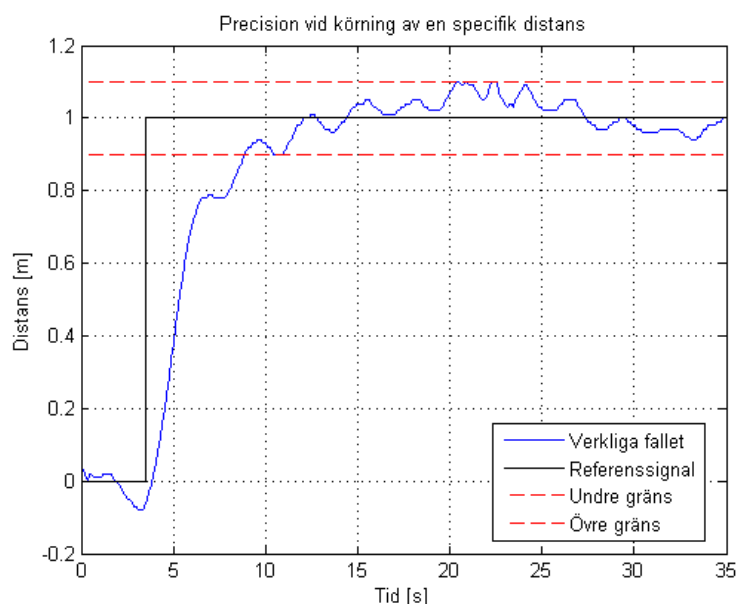
5.2 Precision vid körning av en specifik distans

Detta test utfördes för att verifiera hur väl positionsregleringen av roboten fungerar via att undersöka om krav 2 uppfylls.

Resultat

Utfallet av testet är att roboten klarar av att köra en angiven sträcka med en avvikelse på maximalt ± 10 cm, vilket verifierar krav nummer 2, dvs. att roboten

oscillerar kring den önskade positionen inom en sträcka på 10 cm åt båda hållen. Testet visas i sin helhet i Bilaga 1.



Figur 9: Visar principen för det utförda testet angående precisionen för körning av en viss distans. Den svarta linjen visar referenssignalen vilket är den önskade ändringen av distans. Den blå kurvan visar den verkliga förändringen i distans för roboten. De två rödaktiga streckade linjerna visar en övre och undre gräns för vilket oscillationen av roboten ska hålla sig inom för att hålla det uppsatta kravet. Notera att endast ett resultat ifrån detta test har plottats vilket vilket visas här. Resterande resultat har fått på samma sätt och återfinns i det fullständiga testet precision vid körning av en specifik distans i Bilaga 1.

Analys

Ett intressant resultat är att roboten har i princip samma avvikelse från den önskade positionen oavsett om den färdats en, två eller tre meter. Detta innebär att positionsregleringen fungerar som planerat. Det är oundvikligt att få bort avvikelsen från målpositionen helt och hållet eftersom roboten måste vara i rörelse för att kunna balansera vilket gör att den oscillerar kring sin nuvarande position.

5.3 Körning i en åttaformad bana

Detta test utfördes för att testa hur väl arbetskon fungerar då den åttaformade banan består av fyra olika moment, men även för att testa robotarnas rörlighet när sekvenser av rörelser ska utföras.

Resultat

Alla tre robotar klarar av att köra i en åttaformad bana, dock med stor avvikelse ($\approx 50\text{cm}$) från utgångspunkten efter ett varv för två utav de tre robotarna, medan

den tredje kör med en mycket låg avvikelse ($\approx 3\text{cm}$). Utfallet av testet visas i tabell 1.

Test	Robot	Hastighet	Varvtid	Avvikelse
Test 1	nr 37	1.13m/s	11.85s	3cm
Test 2	nr 42	1.12m/s	11.96s	47cm
Test 3	nr 59	1.12m/s	11.98s	51cm

Tabell 1: Här visas resultatet av testet när robotarna kör i en åttaformad bana. Notera hur stor skillnaden i avvikelse ifrån startpositionen är mellan de olika robotarna är.

Analys

Testet gav väldigt olika resultat beroende på vilken robot som användes. Robot nummer 37 klarade att köra i en åttaformad bana med en avvikelse från startpunkten på endast 3 cm efter ett varv, medan robot nummer 42 och nummer 59 båda hade en betydligt större avvikelse på cirka 50 cm efter ett kört varv. Detta test är består av en sekvens i fyra olika delar vilka är, en raksträcka, en del av en cirkelbana, en raksträcka igen och slutligen en del av en cirkelbana åt det andra hållet. För att köra fler varv i åttan upprepas denna sekvens. Varje delmoment utförs med en specifik tolerans vilket betyder att när roboten befinner sig inom denna börjar den processera nästa delmoment. Den satta toleransen under detta test var 1 cm för varje delmoment. Hamnar roboten inte exakt i rätt position vid slutet av varje delmoment fortplantar sig det felet vidare i de kommande delmomenten vilket kan leda till en stor avvikelse i slutändan.

Robotarna har heller inget positionssystem att regleras efter utan deras bana beräknas helt och hållet ifrån rotationssensorer då hjulen rör sig. För att åstadkomma rörelsen i cirkelbanan i åttan roteras hjulen olika mycket. Detta sätt att styra roboten är känsligt för yttre påverkan i form av att hjulen till exempel sladdar. Är underlaget dammigt eller halt är risken för sladd stor. Sladdar roboten med ett hjul kommer den ändra sin riktning utan att rotationssensorerna känner av detta vilket leder till att roboten hamnar ur kurs. På grund av att fyra olika delmoment utförs påverkas efterföljande delmoment kraftigt då roboten hamnar ur kurs i ett tidigare moment. Detta kan förklara att två av robotarna får en så pass stor avvikelse ifrån den åttaformade banan. En annan orsak kan vara konstruktionstekniska bitar eller att robotarna har fått en stöt som har påverkat till exempel hjulens fixering eller gyrots funktionalitet.

5.4 Topphastighet raksträcka

Detta test utfördes för att ta fram robotens maxhastighet vid körning rakt framåt och undersöka hur väl krav 1 uppfylls.

Resultat

Roboten har en maxhastighet på cirka 1.68 m/s vid körning av en raksträcka, vilket uppfyller krav nummer ett. Utfallet av testet visas i Bilaga 1.

Analys

Vid detta test finns det en del osäkerhet kring mätvärdena. Tidtagning skedde manuellt med hjälp av en timer och därmed påverkades mätvärdena av den mänskliga faktorn. Tidtagningen startade när roboten passerat en specifik markering och stoppades när den passerat en annan markering två meter längre bort. Testet utfördes fem gånger och ett medelvärde ifrån mätningarna togs fram för att minimera mätsäkerheten.

På grund av begränsat utrymme vid testet kunde den totala sträckan roboten körde maximalt flyttas till fem meter. Roboten beräknades ha uppnått maxhastighet efter 1,5 meter och starta retardationen först efter den passerat slutmarkeringen och därmed ha 1,5 meter kvar av den totala sträckan. Detta bekräftades endast med hjälp av en visuell kontroll. I kravspecifikationen är det specificerat att roboten skall klara av en topphastighet på 1 m/s vilket den tydligt gör. Den uppmätta topphastigheten till $1,68 \text{ m/s}$ visar på att trots något felaktiga mätvärden skulle roboten uppnå detta krav med en god marginal.

5.5 Rotation kring egen axel

Detta test utfördes för att undersöka noggrannheten för rotation kring sin egen axel och därmed verifiera hur väl krav 3 uppfylls.

Resultat

Det största utfallet blev en absolut avvikelse på 10° vilket uppfyller krav nummer 3. Utfallet av testet visas i Bilaga 1.

Analys

Det kommer vara essentiellt att robotarna klarar av att rotera snabbt för att följa objekt, exempelvis en boll i rörelse. Med ett mer dynamiskt beteende kan mer avancerade beteenden återskapas vid spel av en fotbollsmatch. Det märktes att ju mindre rotationsgrader som roboten ska vrida sig desto högre blir noggrannheten i precision. Roterar roboten mer än 360 grader får den ofta en större översläng och måste därmed rotera i motsatt riktning efter den utförda rotationen för att nå den önskade positionen. Dock tros detta inte bli något problem ty roboten väldigt sällan kommer behöva rotera mer än ett halvt varv i själva spelet.

5.6 Körning i en böjd bana

Detta test utfördes för att undersöka noggrannheten vid körning i en böjd bana. Kravet som undersökts är krav 4 i kravspecifikationen.

Resultat

Resultatet visar att roboten som högst har en relativ avvikelse på cirka 4 % vilket gör att roboten uppfyller krav nummer 4. Utfallet av testet visas i Bilaga 1.

Analys

Detta test utfördes för att verifiera att roboten kan köra i en böjd bana vilket kommer vara till stor nytta då roboten till exempel ska ta bollen, men en annan spelare står i vägen så att roboten ej kan köra rakt framåt. Programmeringsmässigt utnyttjas cirkelbanor med olika radier för att få roboten att köra i en sväng. För att roboten ska svänga mindre kraftigt används en bana längs en cirkelbåge med en stor radie. Som testet visar klarar roboten av kravet på 10 % relativ avvikelse från målpositionen väl då den som högst var cirka 4 %.

5.7 Stabilisering efter störning av robot

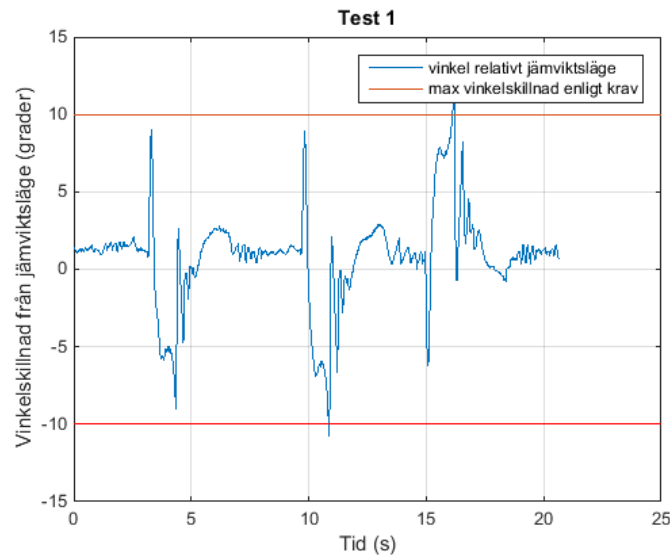
Detta test genomfördes för att kontrollera hur väl krav nummer 5 och 6 uppfylls genom att undersöka hur stora störningar roboten klarar av i form av en simulerad kollision.

Resultat

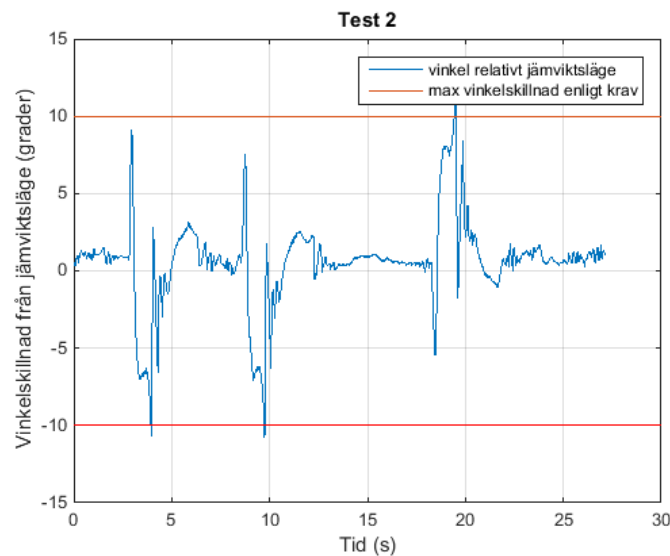
Vid den första testomgången, se figur 10, var störningen av vinkeln gentemot jämviktsläget större än 10° vid två tillfällen och det högsta uppmätta värdet på vinkeln var $11,51^\circ$. Vid den andra testomgången, se figur 11, var störningen av vinkeln gentemot jämviktsläget större än 10° vid varje knuff av roboten. Den största vinkeln uppmättes till $12,61^\circ$. Detta innebär att krav nummer 6 uppfylls. Av figur 10 och 11 kan det avläsas att det tar ca 4 sekunder för roboten att stabilisera sig efter en störning på omkring 10 grader. Detta innebär att roboten inte uppfyller krav nummer 5.

Analys

Det är svårt att bestämma hur stor störning som roboten klarar av när endast en knuff med handkraft används som störning. Då en större kraft användes för att knuffa roboten tippade den över vid ett par tillfällen. Därmed anses de uppmätta störningarna vara goda uppskattningar på den maximala tillåtna störningen. Krav nummer 5 som handlar om att roboten ska stabilisera sig inom en sekund efter en störning är svårt att verifiera. Är störningen tillräckligt liten klarar roboten av detta krav, men resultatet blir inte särskilt användbart. Därmed uppskattades tiden det tar för roboten att stabilisera sig när den utsättes för en störning på över 10° vilket innebär att roboten inte uppnår krav nummer 5. För att få till ett bra flyt när sedan robotarna ska spela fotboll är det viktigare att roboten klarar av att stabilisera sig efter en stor störning i form av till exempel en krock än att det går väldigt fort. Detta på grund av att om roboten ramlar omkull måste den resas upp med hjälp av en människa.



Figur 10: Test nr 1 av stabilisering efter störning. Notera att vinkelförändringen är större än 10° vid två av de tre knuffarna av roboten. Märk även att den största störningen vid varje knuff inträffar när roboten bromsar in och försöker stabilisera sig och inte vid själva knuffen.



Figur 11: Test nr 2 av stabilisering efter störning. Notera här att vinkelförändringen vid varje störning orsakad av knuff av robot är större än 10° vid varje knuff samt att den största vinkelförändringen fås när roboten bromsar in för att balansera till en stillastående position och inte vid själva knuffen.

Det är noterbart att vinkeln vid mätningen inte rör sig runt värde 0 utan att roboten har en något positiv vinkel mätt ifrån jämviktsläget. Detta kan innebära att felaktiga mätvärden fås. Då inga andra störningar i form av uppenbart felaktiga mätvärden av vinkeln ficks under testet är troligtvis orsaken en annan. Det troligaste är att robotens tyngdpunkt inte befinner sig exakt i mitten av konstruktionen. I och med

att skydd i form av rörisolering har monterats kan detta påverka. Även batteriets placering kan bidra till att tyngdpunkten flyttar sig. Då krävs istället en liten vinkel för att roboten ska kunna balansera, jämviktsläget hamnar alltså vid en vinkel skild från noll.

5.8 Mätning av riktning på objekt relativt robot

För att se hur väl roboten kan mäta vilken vinkel olika objekt har i förhållande till sig själv utfördes detta test. Målet beskrivs i krav 9 i kravspecifikationen.

Resultat

Krav 9 i kravspecifikationen uppfylls av roboten som med god precision kan räkna ut vinkeln gentemot andra objekt relativt sig själv. Felet är i genomsnitt $2^\circ - 3^\circ$ och den största uppmätta felmarginalen under testet var 3.5° . Det uppsatta kravet specificerar att felmarginalen högst får vara 10° vilket därmed uppnås med god marginal. Mätserien vid testet visas i Bilaga 1.

Analys

Det finns en del mätosäkerheter kring testet både vid mätningen och beräkningen vid den manuellt beräknade vinkeln och den uppmätta vinkeln vilken fås ut med hjälp av Pixy-kameran. För att få exakta mätvärden krävs det att kameran står vänd vinkelrätt mot bollens mittpunkt. Detta har endast uppskattats via ögonmått. För att kompensera för små störningar och hållas sig stående kan Balanduinoroboten ej heller stå helt stilla. Därmed är avståndet a , mellan boll och robot, ej konstant utan rör sig kring det uppmätta avståndet på 1 m vilket påverkar resultatet av vilken vinkel Pixy läser av. Det finns även mätosäkerhet kring de uppmätta avstånden a och b som används för att räkna ut vinkeln x . Dessa mätosäkerheter anses dock vara relativt små och påverkar inte resultatet nämnvärt. Krav 9 uppfylls som tidigare nämnt med en stor marginal så även om det finns fel i mätvärdena kommer dessa ej göra att felosäkerheten kring vinkeln på bollen relativt roboten överstiger 10° .

5.9 Identifikation av objekt med Pixy

Detta test utfördes för att undersöka hur väl objektidentifieringen med hjälp av Pixy fungerar.

Resultat

Pixy känner igen sex olika objekt med storleken $21\text{cm} \times 30\text{cm}$ och kan identifiera dessa på ett avstånd på två meter vilket gör att krav 7 uppfylls.



Figur 12: Figuren visar hur de färgade papperna uppfattas i Pixy. Notera att hela storleken av vissa objekt inte upptäcks. Rektanglar som indikerar en signatur ändras hela tiden i storlek på grund av att färgigenkänningen inte är helt stabil vilket inte kan visas i bilden.

Analys

Det finns en hel del faktorer som påverkar utfallet av detta test. Pixy är väldigt ljuskänslig så för ett optimalt resultat krävs det ett väl upplyst rum med en neutral bakgrund samt att inga andra objekt som kan störa identifikationen finns i rummet. Detta är svårt att uppnå så därmed har testet påverkats av de rådande förhållandena vid testtillfället. Som figur 12 visar ser Pixy de sex färgade papperna som sex skilda signaturer vilket är så det ska fungera. Dock ser Pixy inte alla objekt fullt ut, utan de olika rektanglarna skiljer sig ifrån papprets storlek. Detta kan utgöra ett problem när roboten med Pixys hjälp ska mäta avståndet till ett visst objekt. Är rektangeln för signaturen mindre än den verkliga storleken på objektet kommer objektet antas befinna sig längre bort än det faktiskt gör. För att uppnå det givna resultatet vid detta test var det en del arbete för att få en bra ljussättning över objekten. Eftersom objekten i detta fallet utgjordes av pappersark föll det ej några större skuggor över objekten då ett ljus lyser från ett visst håll. Detta har annars

upptäckts vara ett problem när objekten istället är saker som till exempel en boll eller liknande. Skuggningar och reflektioner på ytan kan då antingen uppfattas som en del av objektet eller orsaka att endast en del av objektet upptäcks. Vid testen är även objekten och kameran fixerade vid en viss position och flyttas inte runt. Det har upptäckts att det blir problem om samma färger flyttas till en annan del i rummet med en annan ljussättning. Då kan Pixy inte se vissa färger eller så misstar den en färg för en annan närliggande färg. För att få till en så bra fotbollsmatch som möjligt är det därför väldigt viktigt att ljussättningen över hela planen är så jämn som möjligt för att Pixy ska kunna känna igen färgsignaturerna på flera olika positioner på planen.

6 Resultat

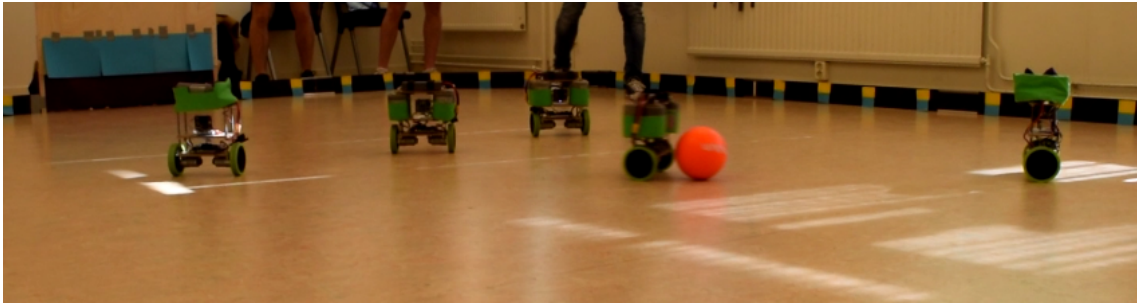
Nedan visas resultat från de två fotbollsmatcher som spelades under projektets gång vilket kan länkas samman med det mål som sattes upp vid projektets start. Förutom det behandlar kapitlet även prestanda för robotarna och Pixy samt benämner kort vilka funktioner som fungerade bättre och vilka funktioner som fungerade sämre.

6.1 Fotbollsmatch mot annat lag

Enligt projektets mål skulle en fotbollsmatch bestående av tre robotar per lag spelas och så blev även fallet. Matchen hade en total speltid på cirka femton minuter med ett avbrott någonstans mitt i för att kalibrera Pixy. Som signaturer i matchen användes färgkombinationer istället för en enskild färg för varje objekt. Detta resulterar i mindre falska positiva, men gör det samtidigt svårare för Pixy att finna objekten. Prestandamässigt så presterade projektgruppens robotar bäst i första delen av matchen och sämre fram mot slutet. Slutresultatet blev förlust med 1-2 efter en spännande match, se tabell 2 för fullständig statistik.

Fotbollsmatch	Projektlaget	Motståndarlaget
Resultat	1	2
Självsmål	1	0
Skott på mål	2	4
Skott på eget mål	1	1
Skott	13	17
Passningar	3	2
Tacklingar på motståndare	2	4
Tacklingar på medspelare	3	3
Omkullkörning	27	24

Tabell 2: Tabellen visar statistiken från den fotbollsmatch som spelades mot ett lag från annat projekt.



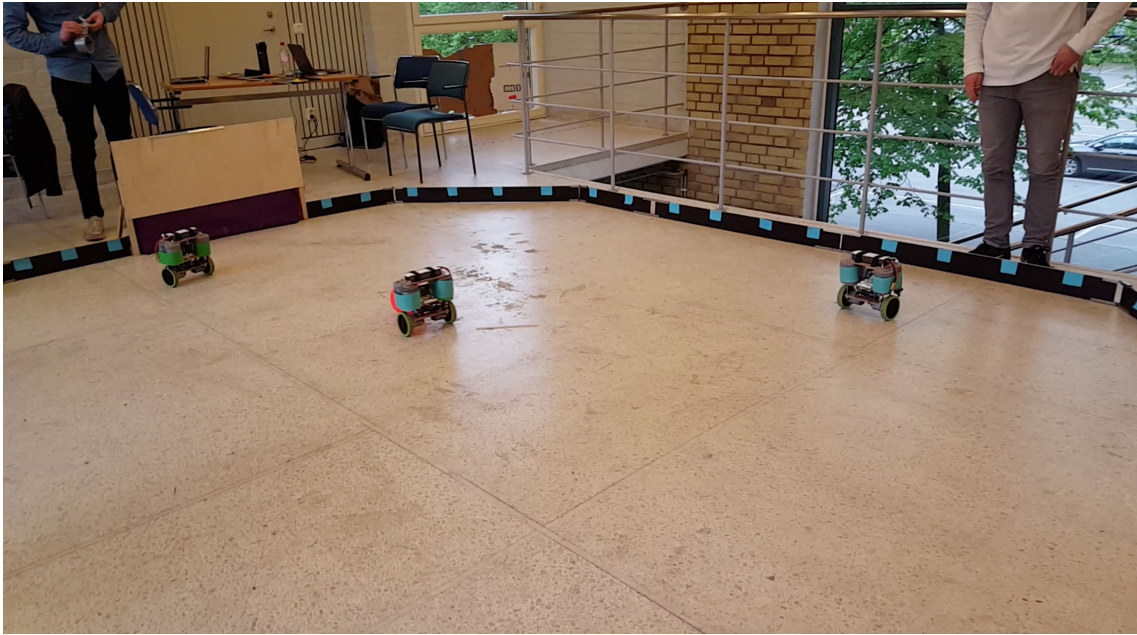
Figur 13: Bild tagen under match mot lag från annat projekt.

6.2 Träningsmatch

För att få fram ett bättre resultat och få en chans att åtgärda de saker som fungerade sämre under fotbollsmatchen så sattes en egen match ihop med två robotar i vardera lag. Denna gången användes enskilda färger som signaturer för att göra det lättare för Pixy att hitta alla objekt. Även om Pixy fungerade bättre under denna matchen var det fortfarande problem med objektidentifieringen. Dock lyckades några av de funktioner som krånglade under den föregående matchen att fungera under denna match, så som målvaktsspelet och funktionen som skulle positionera robotarna i en linje så att boll och mål var rakt framför dem. För fullständig statistik från träningsmatchen se tabell 3.

Träningsmatch	Lag Grön	Lag Blå
Resultat	4	0
Självsmål	0	0
Skott på mål	4	1
Skott på eget mål	0	0
Skott	8	6
Passningar	1	2
Tacklingar	2	2
Tacklingar på medspelare	1	0
Omkullkörning	22	19

Tabell 3: Tabellen visar statistiken från den interna träningsmatchen som projektgruppen spelade. Klara vinnare blev Lag Grön.



Figur 14: Bild tagen under träningsmatch.

6.3 Robotens prestanda

Roboten uppfyller majoriteten av de uppsatta kraven i kravspecifikationen. Det är endast krav nummer fem, vilket säger att roboten ska kunna stabilisera sig inom en sekund efter en störning, som ej uppfylls. Roboten kan utföra enskilda sekvenser väl och håller sig inom noggrannheten för de specificerade kraven. Dock har problem upptäckts vid integrationen mellan Pixy och Balanduinorobot vid olika situationer i matchen. Fungerar inte objektidentifieringen blir informationen till styrningen lidande. Då flera sekvenser utförs i en följd fortplantas fel i noggrannhet vilket leder till att roboten ej alltid hamnar på den tilltänkta positionen.

7 Diskussion

Kapitlet ämnar att gå in djupare på de moment som fungerade bra och dåligt under projektets gång samt utreda varför robotarna presterade som de gjorde. Det tas även upp mer analys angående hårdvaran och mjukvara som har använts samt hur saker som fotbollsplan och kameraplattform har tagits fram.

7.1 Fotbollsmatch

Den fotbollsmatch som utspelades gav en tydligare bild för hur väl alla funktioner fungerar tillsammans. Vad som kunde uppmärksammas var att Pixys objektidentifiering fungerar sämre än vad som förväntats och även att en del funktioner, som hade fungerat när de testats enskilt, fungerade sämre. Att Pixys igenkänning inte var lika bra som tidigare kan bero på att just färgkombinationer istället för enskilda färger användes som signaturer för objektidentifiering. Vid tidigare tester har mestadels enskilda färger använts som signaturer men för att få bort falska positiva

användes istället färgkombinationer till viss del. Problemet med färgkombinationer är att även om de minskar risken för falska positiva gör de även det svårare för Pixy att hitta det korrekta objektet. Pixy behöver då känna igen minst två kombinerade färger vilket blir svårt när ljusförhållandena skiftar. Projektgruppen tror därför att med bra kalibrering och enskilda färger som signaturer hade fotbollsmatchen givit ett bättre resultat. Ett tecken på detta är att igenkänningen för bollen, som var en enskild färg, var det som fungerade bäst.

Resultatet från den första fotbollsmatchen visade på att det var en väldigt liten del utav skotten som faktiskt gick på mål. Det var dessutom mycket omkullkörningar och tacklingar som inträffade. Den andra matchen visade på ett bättre resultat men det finns dock fortfarande en del saker som kan komma att förbättras. För att spelet skall kunna bli helt autonomt så behöver robotens kollisionsundvikande förbättras. Ett exempel är specialfallet när bollen ligger framför sargen och roboten gör bedömningen att sparka på bollen oavsett om sarg kan identifieras, och det här beror på brist i strategierna eftersom de inte tar hänsyn till dessa specialfall. Möjligtvis kan även sargen, som orsakade många omkullkörningar, behöva förändras.

7.2 Robotens prestanda

Det är viktigt att balanseringen av roboten fungerar bra för att ge en bra grund till att kunna utföra de olika momenten som finns i en fotbollsmatch. För att verifiera att roboten faktiskt klarar av specifika moment har det utarbetats ett flertal tester vilka kan hittas i Bilaga 1 i sitt fulla utförande. Dessa tester har använts för att undersöka om de uppsatta kraven i kravspecifikationen har uppfyllts. De uppsatta kraven gäller främst robotens prestanda för att utföra enkla moment av styrningen av roboten, men även en del krav för Pixy angående prestandan för objektidentifiering.

Testerna visar att roboten uppfyller alla krav för Pixy och alla krav för robotens styrning förutom kravet angående att stabilisera sig inom en sekund efter en störning. En sak som är viktig att tänka på är att de utförda testerna har undersökt hur väl en viss rörelse eller ett visst moment har fungerat enskilt. Görs flera sekvenser av rörelser i en följd kommer den uppnådda positionen skilja sig betydligt mer från den önskade då felen fortplantar sig mellan varje ny sekvens. Detta kan ses i testet: Körning i en åttaformad bana. Där utförs fyra olika sekvenser av rörelser i en följd som sedan upprepas ett flertal gånger. Det kan tydligt ses att felet för varje sekvens fortplantar sig då två av de tre robotarna hamnar en bit ifrån den önskade positionen efter ett varv i åttan trots att de använder sig av samma programkod. Orsaken till denna skillnad i precision hos robotarna kan bero på små skillnader i hårdvara mellan robotarna eller olika yttre påverkan på dem. Det är omöjligt att tillverka två identiska robotar utan det kommer alltid finnas vissa skillnader i till exempel motorerna och deras tröghetsmoment, positionering av olika delar eller gyroskopet och accelerometerns funktionalitet. Orsaken kan även vara att en robot har ramlat och ett hjul har blivit snedställt eller att en robot har sämre grepp på golvet. Sedan finns det en möjlighet att skillnaden beror på mjukvaruproblem, då distanser som

roboten skall köra sätts med en tolerans så finns alltid risken att roboterna samplar olika och slutför arbetet vid olika ställen inom toleransen.

De utförda testerna av Pixy har gjorts i en så bra miljö som möjligt med hänsyn till val av färger, ljusförhållanden och kalibrering. Vi har märkt att Pixy är väldigt känslig för dessa saker och resultatet förändras i en stor grad om förhållandena inte är optimala, eftersom Pixy är väldigt ljuskänslig och är dålig vid omställning av exponering. Färger påverkas av ljuset och den observerade färgen av ett objekt påverkas därför väldigt mycket i förhållande till den kalibrerade färgen. Därmed är det inte troligt att objektidentifieringen fungerar lika bra när roboten rör sig runt på planen. Det är även troligt att de olika kamerorna fungerar olika även fast de har samma programvara. Detta har visat sig genom att olika kameror har haft olika inställningar för kalibreringen. Att kalibrera en Pixy och sedan föra över dess parametrar för objektidentifiering till en annan Pixy har inte gett önskat resultat utan varje Pixy måste kalibreras individuellt.

7.3 Analys av mjukvara

Följande avsnitt analyserar de olika strategierna för att bedöma hur väl de fungerar samt programstrukturen i sin helhet.

7.3.1 Balans

Särskilt att ha en bra balans är viktig för robotens prestanda. Då man gärna vill kunna klara stora störningar och ändra börvärden mycket utan att få en vältande robot. Gamla filtret i figur 2 visar ett problem som blir när robot vinkeln oscillerar mycket. Detta är en sak som kan förbättras mer.

7.3.2 Målvakt

När målvakten testades enskilt fungerade den relativt väl. Roboten lyckades hitta det egna målet, vända sig om för att möta upp en eventuell boll som kom rullande och sedan återgå till sin position framför målet. Dock var resultatet sämre när den användes i den första fotbollsmatchen. Roboten hade då svårt att överhuvudtaget ta sig till målet. Man kunde se att den bara stod och sökte efter målet genom att rotera fram och tillbaka, alltså att den inte kunde se målet alls. I den andra matchen fungerade målvaktsspelet bättre, roboten kunde nästan alltid ta sig till målet förutom någon robot som hade en dålig kalibrering. Roboten kunde lätt hitta bollen och möta upp. Då ingen kod hade förändrats mellan den första och den andra matchen bekräftar det att det enbart är objektidentifieringen som krånglar vid målvaktsspelet och inte själva koden för robotarna.

7.3.3 Försvarare

Försvarare och målvakt liknar varandra. Ända skillnaden är att en försvarare kan följa upp längre ut på mittfältet och spela längre ut generellt. Då det alltid är väldigt värdefullt att ha en målvakt i fotbollsmatcherna visar sig försvarsfunktionen vara

väldigt användbar i matcher med ett mindre antal robotar. Den kan där ge laget en ökad flexibilitet genom att låta en robot agera både utespelare och målvakt. Försvaren visade sig vara bra på att rotera in till mål under träningsmatchen och kunde även återgå till mittfältet.

7.3.4 Anfallare

Anfallaren är konstruerad för ett mer offensivt spel än de andra strategierna. Identifierar roboten det egna målet och bollen samtidigt kör den runt och letar annat läge. Den offensiva strategin försöker beräkna fram en position som roboten ska ta sig till för att positionera sig på en linje så att ett rakt skott på mål kan utföras. Under den första fotbollsmatchen blev allt ganska slumpmässigt för att den i princip var blind. Däremot under den andra matchen var det mycket lättare att identifiera målburar, vilket gjorde anfallaren mer relevant då den kunde utföra fler moment för att skjuta på motståndarlagets mål. Särskilt kollisionsdetektion behöver förbättras som att ta med sin egen hastighet i bedömning av risk för kollision. För att annars hinner den kanske inte bromsa in även om den ser hinder i tid. Samt inte slå bollen rakt mot sargen när den nära den.

7.4 Analys av hårdvara

Projektet bygger på en del hårdvara som ska användas, men utöver dessa kan ytterligare komponenter införskaffas som kan bidra till en bättre slutprodukt. I följande avsnitt diskuteras dessa komponenter utifrån hur väl de fungerar.

7.4.1 Balanduino

Ett alternativ vore att använda sig av robotar som har fler än två hjul eftersom det hade blivit ett stabilt system där reglering för balanseringen hade eliminerats. Däremot hade systemet blivit allt mer komplext med fler hjul och motorer att ta hänsyn till vid reglering. Fastän det kanske inte är ett felaktigt alternativ, blev det överraskande när Balanduinorobotarna navigerade sig framåt med sådan bra precision, vilket är till störst del på grund av den goda programmeringsstrukturen. Det var lättare att använda sig av ett instabilt system än förväntat, där dessa är av typen inverterad pendel. Det innebär att roboten har bra förutsättningar att snabbt rotera åt olika håll kring sin egen axel.

7.4.2 Pixy

Vi fann att Pixy har större svårigheter att identifiera objekt med god säkerhet, mest på grund av dess ljuskänslighet. Då en signatur sätts i Pixy kalibreras färgen efter olika tröskelvärden för vilka nyanser av färgen som ska uppfattas. Detta orsakar ofta problem då ljusförhållandena förändras, på grund av den dåliga omställningen av exponering. Är tröskelvärdet av färger för lågt känner Pixy inte igen färgen och är tröskelvärdet satt för högt kan andra färger ses som den satta färgsignaturen i ett annat ljus. Det har därför varit svårt att kalibrera Pixy att känna igen objekt i lokaler där ljusförhållandena är dåliga och objekt i bakgrunden har haft en chans

att bli uppfattade som de objekt som är i spel. I lokaler med fönster där solljus lyser in uppstår ofta ytterligare svårigheter för Pixys identifiering av objekt från olika vinklar eftersom solljuset från en sida medför att objekt med en viss färg behöver en högre tröskel för att identifieras, dock innebär det att många fler falska positiva kan förekomma. De ideala förhållandena skulle vara en klar vit lokal som har jämn belysning över alla objekt från alla vinklar. Det innefattar lampor som ger vitt ljus och vita väggar som inte gör att Pixy identifierar objekt i bakgrunden som falska positiva, och inga blanka golvytor som kan reflektera objekt och orsaka felmätningar. De här är de största anledningar till de svårigheter som förekom under fotbollsmatchen som spelades mot det andra laget.

7.4.3 Utredning av trådlös kommunikation inom robotlaget

Medföljande de färdiga Balanduino-robotarna kom en Bluetooth-modul i form av en USB-ansluten enhet. Enligt hemsidan för Balanduino, se [3], kan denna användas för att kontrollera roboten med diverse olika kontroller annars avsedda för TV-spel. I projektets tidiga skede antogs därför att dessa enheter även borde kunna anslutas mellan varandra och därmed möjliggöra en trådlös kommunikation mellan alla robotar tillhörande projektet. Dock misslyckades varenda försök att skapa en sådan anslutning. Vid försöken användes programkod som utgick ifrån exempelkod skriven av Kristian Lauszus [13] vid TKJ Electronics. Denna kör ett protokoll som ska simulera en seriell anslutning, se appendix A. Problemet grundades i att den USB-enhet som användes bara kan agera slave. Detta innebär att en anslutning mellan två av dessa USB-enheter inte är möjlig, vilket ledde undersökningen till att leta fram annan hårdvara. Eftersom det inte ingår i projektets mål och krav att kommunikation skulle finnas utan endast är satt som ett önskemål beslutades det att undersökningen skulle avbrytas. Det blev mer tidskrävande än vad som rymdes inom tidsplanen.

7.4.4 Mikrofon

Idéen för att starta och stoppa spel kom upp halvvägs in i projektet. Därför införskaffades mikrofoner för att det skulle hjälpa till att smidigare starta och stoppa ett pågående spel. Det ideala vore om en signal i form av ett visslande ljud till exempel från en visselpipa skulle få alla robotar på spelplanen att helt stoppa sitt spel och balansera på plats tills en ny signal i form av ytterligare en vissling uppfattas. Dock märktes det under förberedelserna inför matchen mot det andra laget att det inte alls fungerade som planerat. Det resulterande beteendet av funktionen blev istället att någon enstaka robot stannade medans andra fortsatte att spela. Dessutom kunde robotar stanna till av andra ljud som kommer från omgivningen, exempelvis personer som talar högt eller släpar stolar längs golvet som ger ifrån sig ljud. Roboten kunde även stänga av sig själv på grund av ljudet från motorn vid snabb körning. Matchen fortsatte med att funktionen stängdes av efter en stund, och därifrån fortsatte spelet utan funktionen och förlitade sig enbart på manuell start och stopp för spel. Vid planeringsfasen för hur start och stopp av spelet antogs robotarna inte ramla omkull så ofta som de faktiskt gör. Därmed är robotarna tvungna att resas manuellt väldigt ofta och funktionen av en mikrofon tappar i värde.

7.5 Produktion av övriga komponenter

Förutom den hårdvara som var bestämd från projektets start och de detaljer som köptes in behövdes en del andra komponenter tas fram på egen hand. En fotbollsplan samt kameraplattform skapades och tillvägagångssättet presenteras i de kommande avsnitten.

7.5.1 Design av kameraplattform

För att kunna fästa Pixy på Balanduino behövdes någon typ av plattform. De valmöjligheter som fanns är att antingen köpa en färdig plattform eller skapa en egen. Med motivationen att en egen plattform inte blir allt för svår att göra, med faktumet att det även blir billigare och att den färdiga plattformen dessutom innehöll funktioner som är överflödiga, så beslutades det att en egen plattform ska konstrueras. Arbetet började med att skissa fram en prototyp i CATIA med hjälp av mått som finns tillgängliga på Pixys hemsida. I den första prototypen tog vi hänsyn till att en tilt-funktion, vilket innebär att kameran kan vinkla sig i vertikalt led, kanske skulle användas och plats för en eventuell servomotor togs med i skisserna. Det blev totalt sett tre versioner av plattformen, som alla togs fram i och med att vi kom längre in i projektet och fler komponenter tillkom eller togs bort. Idéen med att ha en tilt-funktion togs bort för den slutgiltiga versionen av plattformen, vilket innebar att det finns ledigt utrymme. Skruvhål skapades där denna lediga yta fanns för att kunna fästa mikrofonen på plattformen i samband med Pixy.

7.5.2 Produktion av fotbollsplan

För att en möjlighet till fotbollspel av robotar skulle kunna ske var det nödvändigt att en spelplan konstruerades för robotarna att spela på. Tidigt i projektet blev det gemensamma möten tillsammans med den andra projektgruppen i robotfotboll för att bestämma generella regler och diskutera format för spelplanen. Till en början var det bestämt att en plan med endast linjer på golvet, som med tydliga markeringar och färger, skulle användas eftersom robotarna skulle ha möjligheten att identifiera och spela innanför spelplan. Något som kom upp längre fram var att denna typ av spelplan skulle skapa en hel del avbrott i spelet då bollen skulle skjutas utanför spelplanen alldeles för ofta. Det var anledningen till att vi beslutade att använda oss av en sarg i spelplanen för att få mer kontinuitet i spelet. För att robotarna sedan i sin tur ska veta att det finns en sarg och kunna planera sin bana utifrån planens gränser används färgkoder utspridda på sargen. För att planen skall vara smidig att förflytta samt kunna variera i storlek konstruerades sargen i moduler. För att kunna montera ihop modulerna klipptes plåtbitar ut som sedan bockades, se figur 21. Dessa kunde sedan smidigt monteras samt avmonteras på modulernas över- och undersida för att länka samman hela planen.

Den andra komponenten som behövdes för att fotbollsplanen skulle bli funktionell var en målbur. Eftersom den var en del av det gemensamma arbetet med den andra projektgruppen i robotfotboll har även målburen genomgått en del olika koncept. Första konceptet handlade om att bara använda sig av enklare stolpar för att robo-

ten skulle hitta målburen. Lite längre in i projektet diskuterades det huruvida det kan bli svårt för Pixy att identifiera den här typen av målbur på längre distanser eftersom stolparna var smala. Därifrån kom andra konceptet som byggde på att ha en vägg längst bak i målburen som kunde färgläggas för färgkodning i form av färgkombinationer. Det skulle bli mycket lättare att identifiera på avstånd med en större yta. Efter ytterligare diskussion beslutades att det inte skulle bli så bra som det först var tänkt med det här konceptet heller, mest på grund av den extra distansen som blir bakåt (djupet inåt i målburen) från sargen och hur Pixy identifierar målburen från olika vinklar, t.ex. om sargen täcker delar av den nedre halvan av målet för Pixy. Till sist kom vi till det koncept som sedan konstruerades, då vi flyttade den bakre regionen av målburen fram så den fick bli ribba, se figur 22. Då prototypplabbet har ett begränsat urval av trä plockas det ihop olika träbitar för att få ihop de två målburarna. Dessa bitar kapades till för att få rätt längd och skruvades sedan enkelt ihop med varandra. Under själva processen var det framförallt två stycken saker som det lades fokus på. Det första var att få målytan tillräckligt stor för att bollen skulle kunna rulla igenom målet och det andra var att ribban i sin tur skulle bli stor nog för att den skulle synas tydligt på längre avstånd.

8 Slutsatser

I det här kapitlet presenteras de slutsatser som har tagits fram utifrån de resultat som producerats.

8.1 Autonom fotbollsmatch

Det är fullt möjligt att spela en fotbollsmatch med hjälp av robotar, vilket påvisas i resultatet. Vad som är tydligt är dock att robotarna inte skulle klara av att spela en match på egen hand utan ingripande av människor efter en omkullkörning eller tackling. Hela syftet med att skapa autonoma robotar är att de ska vara självgående under hela matchen och klara sig helt utan hjälp. Robotarnas uppgift i detta projekt är att spela en match, dock inte helt utan ingripande av människor. Det finns områden som kan förbättras och byggas vidare på för att förbättra spelet så det blir säkrare, mer avancerat och självgående.

8.2 Precision i rörelsemoment för Balanduino

När robotarna skulle utföra rörelsemoment för att köra raksträckor, cirkelbanor eller rotationer upplevdes precisionen förvånansvärt bra. Det kan konstateras att robotar som baseras på det klassiska instabila systemet av en inverterad pendel är en bra metod för att basera ett mobilt system på. Detta på grund av den enkelhet kring utnyttjandet av dess balansering, som konstant måste regleras, för att styra den framåt och bakåt. Att roboten enbart har två hjul gör att den även enkelt kan roteras kring.

8.3 Objektidentifiering med hjälp av Pixy

Det blev tydligt att Pixy som enda sensor inte var tillräckligt för att återskapa en kontinuerlig fotbollsmatch med säkra mätningar och beslutstaganden, vilket hade bidragit till att förminska antalet avbrott under matchens gång avsevärt. Det förekom alldeles för många felaktiga signaler för objektidentifikationen från Pixy, som resulterar i felmätningar och sedan felbedömningar.

8.4 Trådlös kommunikation

Efter undersökning konstanterades att problemet låg i hårdvaran. De Bluetooth-enheter i formen av en USB-dongel var förprogrammerade till att agera slave-enheter de visade sig inte vara open-source. Detta innebar att de bara kan upprätta en anslutning med en master-enhet, i detta fallet skulle det vara en av flera olika TV-spelskontroller eller en mobiltelefon, och inte mellan varandra. Följaktligen behöver ett projekt av denna typen en modul som fritt går att programmera.

8.5 Förbättra mikrofonens funktionalitet

Det fastställdes tidigt under matchspel att funktionaliteten för mikrofonen skulle stängas av, detta på grund av hur ofta det förekom att robotar avbröt sitt spelande när de uppfattade signaler från omgivningen. För att åtgärda och få en bättre kontroll över denna funktionalitet behöver tröskeln för registrering av ljud höjas en del och med det behövs även en mer högljudd och distinkt signal, så som en visselpipa, användas för att utlösa start eller stopp under en fotbollsmatch.

8.6 Projektet som pedagogiskt hjälpmedel

Att utveckla och programmera robotar har visat sig vara väldigt lärorikt på många plan. Genom att använda delvis färdiga robotar och användarvänliga utvecklingsmiljöer kan man skapa stora inlärningsmöjligheter för breda målgrupper. Eftersom man direkt kan exempelvis prova programkod och upptäcka att den fungerar öppnar projektet upp för skapandet av pedagogiska plattformar för elever från lågstadiet upp till gymnasiet.

9 Vidareutveckling

De tankar och idéer som presenteras i det här kapitlet är till för att ge förslag på förbättringar till framtida projekt inom området.

9.1 Trådlös kommunikation

Ett möjligt mål eller delmål för framtida projekt kan vara att bygga ett trådlöst nätverk mellan robotarna och skapa effektiva spelstrategier med mera. För att göra detta togs nedanstående förslag fram som möjlig ersättning av de medföljande USB-donglarna:

- HC-05 trådlös Bluetooth-modul för seriell kommunikation för Arduino.
- ESP8266 WiFi-modul för Arduino.
- Radiosändare och -mottagare likt de som används för radiostyrda leksaksbilar.

Om man använder Bluetooth-teknologin kan exempelvis en värddator placerad utanför planen användas som master-enhet för att ta emot, bearbeta och skicka ut data från robotarna, alternativt utses en robot till ”lagkapten” och får ansvaret att på liknande sätt hantera datan från övriga robotar.

9.2 Förslag på sensorer

En sak som var tydlig mot slutet av projektet var att en Pixy vid bristande ljusförhållanden inte klarade av att skicka tillräckligt säker information. Är denna information instabil kan felaktiva signaler skickas till Balanduino-kortet och därmed påverka styrsignalen till roboten. Det kunde ge väldigt oönskade beteenden, eller om Pixy får allmänna svårigheter att identifiera objekt står roboten bara still.

Det som rekommenderas för framtida projekt är att tillföra fler sensorer, vilket kan vara i princip vad som helst. Vad vi diskuterade tidigt i projektet var om vi skulle använda oss av SONAR, vilket är en teknik som bygger på att ljudvågor skickas och sedan återkommer för att beräkna avstånd till förekommande objekt i omgivningen. Det är en mycket vanlig teknik som används under vatten för att navigera, men skulle även ha en möjlig användning i det här projektet, dessutom finns det att köpa en billig designad för Arduino, Ultrasonic Sensor HC-SR04.

En annan tänkbar lösning på problemen med väldigt ljuskänsliga Pixy är helt enkelt att denna ersätts av en motsvarande kamera med bättre hårdvara. Nackdelen är att kostnaden för dessa kameror snabbt stiger när kvaliteten på den bearbetade bilddatan ökar och ljuskänsligheten minskar. Dock med färdiga, fotbollsspelande robotar, fotbollsplan m.m. ser vi att det med stor sannolikhet kan rymmas, i en budget motsvarande detta projektets, en kamera av högre kvalitet.

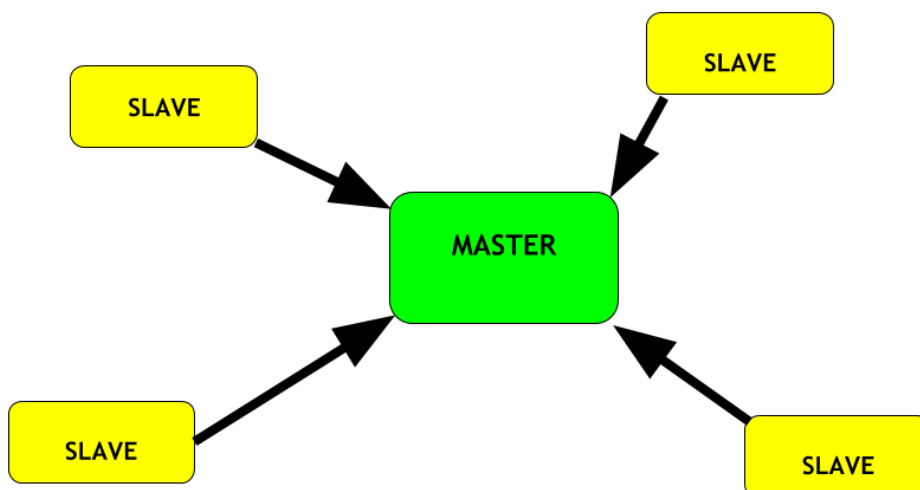
Referenser

- [1] Peter Bergström, Karim Gabe och Anders Meurk. ”Den inverterande pendeln med oscillerande fästpunkt”. I: (). URL: <http://www.mech.kth.se/~hanno/InvPendelOscKandExJobb.pdf>.
- [2] Donald E. Catlin. *Estimation, Control and the Discrete Kalman Filter*. Amhers, MA 01003: Springer-Verlag New York Inc, 1989.
- [3] TKJ Electronics. *Balduino - Balancing Robot Kit*. April 2016. URL: <http://balduino.net/about-balduino>.
- [4] TKJ Electronics. *Overview - Balduino*. Mars 2014. URL: http://wiki.balduino.net/Overview#Frame_design.
- [5] Pooyan Fazli. *Alan Mackworth Biography*. 2013. URL: <https://www.cs.ubc.ca/~mack/Biography.htm>.
- [6] Pooyan Fazli. *Alan Mackworth RobotSoccer*. 2013. URL: <https://www.cs.ubc.ca/~mack/RobotSoccer.htm>.
- [7] FIRA. *About FIRA*. April 2016. URL: http://www.fira.net/contents/sub01/sub01_1.asp.
- [8] FIRA. *FIRA Bylaws*. April 2016. URL: http://www.fira.net/contents/sub01/sub01_7.asp.
- [9] Tom Harris. ”How Robots Work”. I: (april 2002). URL: <http://science.howstuffworks.com/robot.html>.
- [10] Jong-Hwan Kim m. fl. *FIRA RoboSot Competition - Version 1.0.0*. Aug. 2006. URL: <http://www.fira.net/contents/data/RoboSot.pdf>.
- [11] P Kopacek. ”Industrial applications of Robotsoccer”. I: *Proceedings 5th MATH-MOD* (febr. 2006). URL: <https://www.engineeringvillage.com/search/doc/detailed.url?SEARCHID=d8040c25M5cb6M441eMada4M914fa3dcc016&DOCINDEX=1&database=2&pageType=quickSearch&searchtype=Quick&dedupResultCount=&format=quickSearchDetailedFormat>.
- [12] Jean-Philippe Lang. *Overview - CMUcam5 Pixy - CMUcam: Open Source Programmable Embedded Color Vision Sensors*. April 2016. URL: <http://cmucam.org/projects/cmucam5>.
- [13] Kristian Lauszuz. *USB Host Shield 2.0*. Mars 2015. URL: https://github.com/felis/USB_Host_Shield_2.0/blob/master/examples/Bluetooth/SPP/SPP.ino.
- [14] University of Michigan, Carnegie Mellon University och University of Detroit Mercy. ”Inverted Pendulum: System Modeling”. I: (okt. 2015). URL: <http://ctms.engin.umich.edu/CTMS/index.php?example=InvertedPendulum§ion=SystemModeling>.
- [15] Microsoft. *What is a Bluetooth personal area network (PAN)?* 2016. URL: <http://windows.microsoft.com/en-us/windows/what-is-bluetooth-personal-area-network#1TC=windows-7>.

- [16] Rudy Negenborn. "Robot Localization and Kalman Filters On finding your position in a noisy world". Examensarb. 3508 TB Utrecht The Netherlands: Utrecht University, sept. 2003.
- [17] Thilo Weigel m.fl. "CS Freiburg: coordinating robots for successful soccer playing". I: (okt. 2002). URL: <http://ieeexplore.ieee.org.proxy.lib.chalmers.se/xpls/icp.jsp?arnumber=1067991>.

A Bluetooth

Bluetooth (populärt kallat blåttand på svenska) är en standard för att utföra kommunikation mellan olika enheter trådlöst. Den använder sig av höga frekvenser, 2.45 GHz, och fungerar således bara på korta avstånd (<100 meter). Med hjälp av en mängd olika protokoll går det via Bluetooth skapa snabb och energisnål kommunikation mellan enheter fysiskt nära varandra och då även skapa så kallade PANs, Personal Area Networks [15]. Dessa nätverk kallas även piconets (se figur 15, där en enhet antingen kan vara en master- eller en slave-enhet. Upp till 7 aktiva slave-enheter kan vara anslutna till en master-enhet samtidigt.

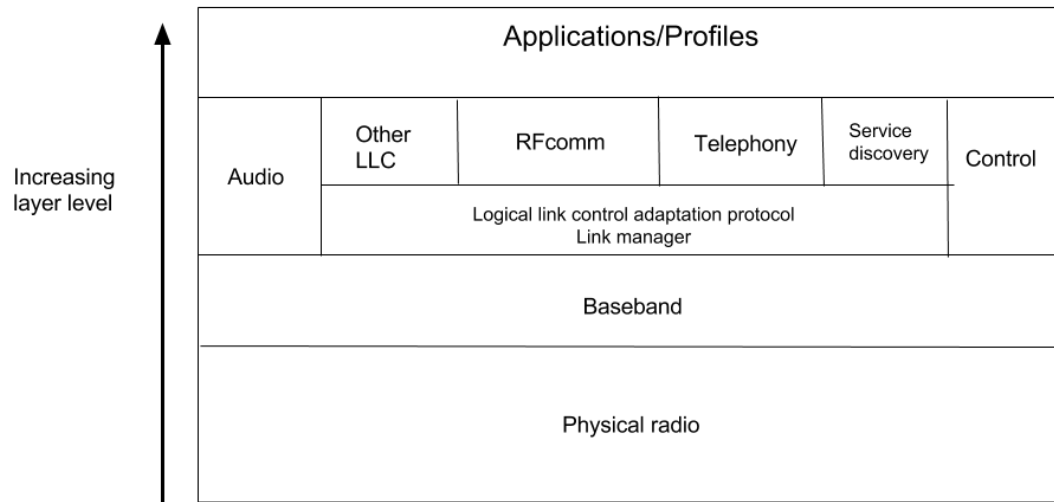


Figur 15: Figur över ett s k piconet. MASTER-enheten kan vara kopplad med upp till 7 aktiva SLAVE-enheter samtidigt och ännu fler satta i viloläge.

A.1 Protokoll

För att kommunikationen ska fungera bra måste vissa specifika regler följas, definierade för den typ av anslutning som ska upprättas. Dessa regler samlas i ett protokoll som implementeras i mjukvaran. Internet, lokala datornätverk och WiFi är fler exempel på där protokoll som dessa används. Inom Bluetooth, som är en familj av protokoll, existerar en mängd som alltid måste finnas med, s k "core protocols". Utöver dem läggs applikationsprotokoll till. Hela lagerstrukturen kan ses i figur 16. En enkel applikationsprofil är Serial Port Profile (SPP) som använder sig av protokollet

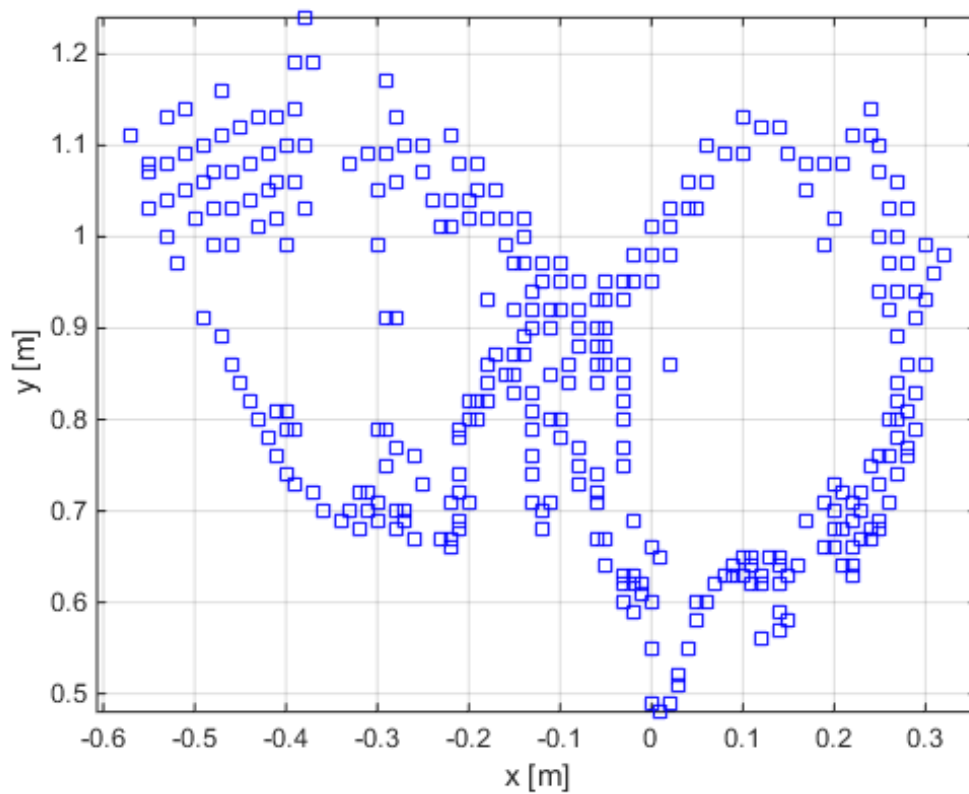
Bluetooth Protocol Stack



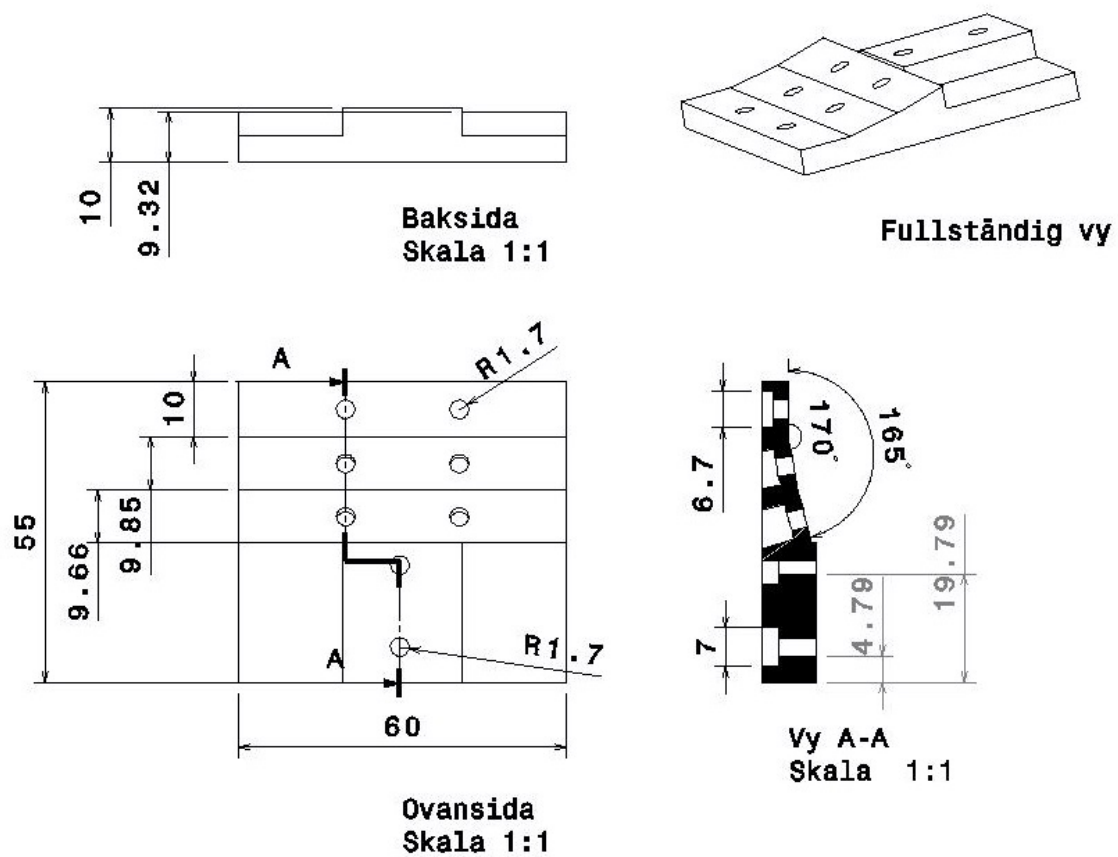
Figur 16: Illustration av lagerstrukturen som används av Bluetooth

RFcomm för att emulera en seriell port. Exempel på protokoll som kan användas av applikationsprofiler är:

- Telephony Control Protocol Specification (TCS)
- Audio/video control transport protocol (AVCTP)
- Radio frequency communication (RFCOMM)



Figur 17: Dåligt filtrering av position hos en boll relativt robot i origo. Man kan se konturen av en brusig åttaformad bana.



Figur 18: Ritning över kameraplattformen för att fästa Pixy samt mikrofon. Alla mått presenteras i millimeter.



Figur 19: Skydd för Balanduino gjort av två lager rörisolering.



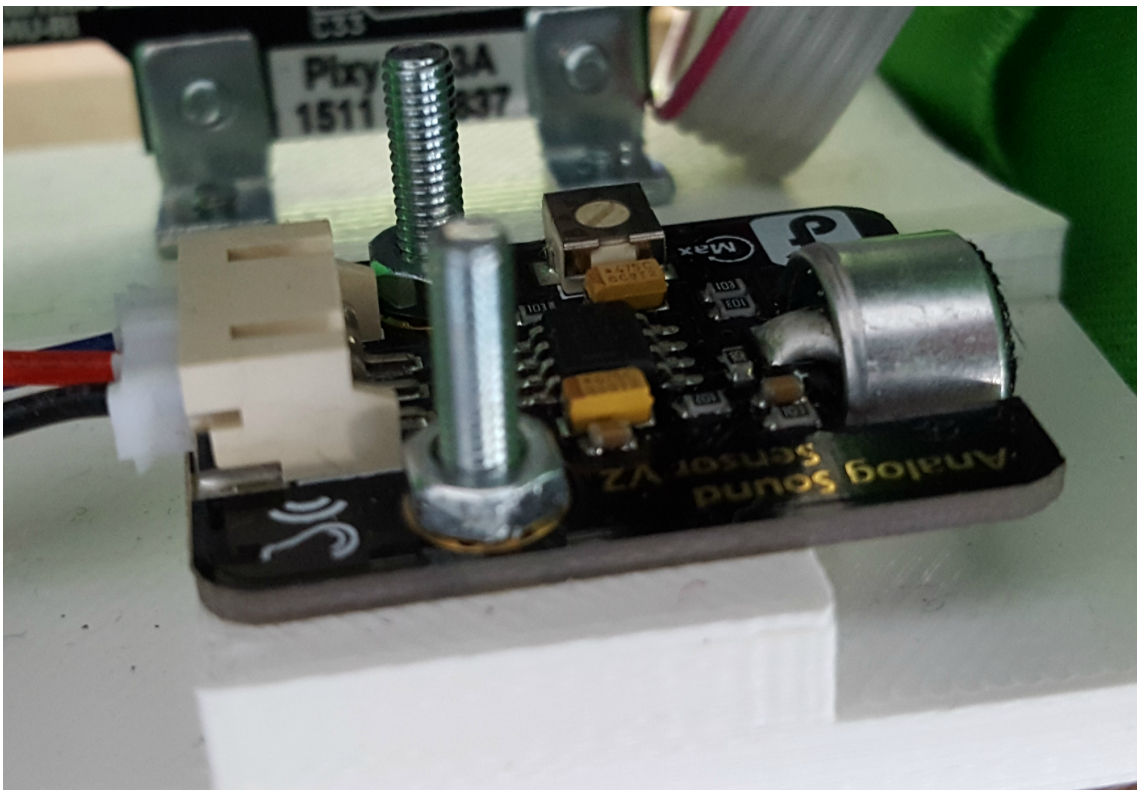
Figur 20: Bild på USB-dongeln för Bluetooth-kommunikation som följer med Balanduino-roboten.



Figur 21: Metallstängerna som binder samman sargen.



Figur 22: Bild på konstruerad målbur för fotbollsspel.



Figur 23: Bild på mikrofon för att starta och stoppa spel med hjälp av ljudsignal.

Bilaga 1

Precision vid körning av en specifik distans

Syfte

Syftet med testet är att mäta noggrannheten för angiven distans.

Kravsbeskrivning

- Roboten ska vid körning rakt framåt ha en avvikelse på högst ± 10 cm ifrån måldistansen.

Genomförande

Med markeringar på golvet mäter vi upp olika distanser och sedan mäter vi felet mellan positionen där roboten stabiliseras och markeringen för måldistansen. Då roboten fungerar som en inverterad pendel kan den aldrig vara helt stilla. Därmed mäts insvängningstiden tills roboten positionerar sig inom den önskade avvikelsen på ± 10 cm och ej går utanför dessa markeringar. Avvikelsen från den önskade positionen mäts som den maximala avvikelse från angiven position under en tidsperiod på 30 s. Denna tid startas då roboten nått den önskade positionen.

Resultat

Utfallet av testet visas i tabell 4 för de olika distanserna.

Test	Distans	Avvikelse	Insvängningstid
Test 1	1 m	± 10 cm	9.17 s
Test 2	1 m	± 9 cm	9.67 s
Test 3	2 m	± 9 cm	11.84 s
Test 4	2 m	± 8 cm	12.57 s
Test 5	3 m	± 7 cm	19.56 s
Test 6	3 m	± 9 cm	16.98 s

Tabell 4: Här visas de olika testfallen för precision av körning av en specifik distans. Notera att den maximala avvikelsen är 10 cm från den körda distansen vilken uppkommer då roboten oscillerar kring dess position.

Slutsats

Roboten klarar av det uppsatta kravet på att nå en position med en avvikelse på maximalt ± 10 cm.

Körning i en åttaformad bana

Syfte

Köra roboten i en åttaformad bana för att se hur väl de framtagna algoritmerna för att köra i cirkelbana samt raksträcka fungerar.

Vad testet vill uppnå

- Utan avbrott köra en bana i form av en åtta.

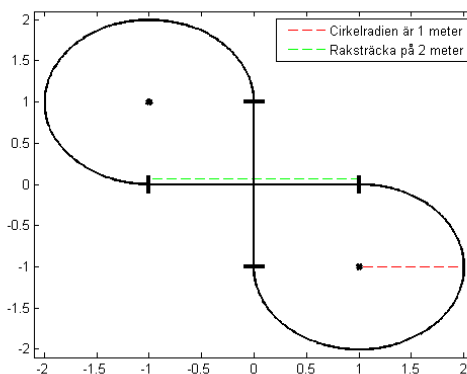
Genomförande

För att utföra testet markeras de punkter på golvet där roboten byter arbete, vilka plockas ut från arbetskön och ser ut som följande:

1. Raksträcka
2. Cirkelbana (vänster)
3. Raksträcka
4. Cirkelbana (höger)

Denna arbetslista fyller sedan på arbetskön så fort kön blir tom, roboten åker därför ändlöst i en åtta.

Tidtagning kommer göras per varv/åtta, i godtyckligt antal, samt mätning hur mycket roboten avviker från utgångspunkten då den utfört alla fyra arbeten (ett varv).



Figur 24: Överblick av den planerade banan.

Resultat

Resultatet visas i tabell 5 och visar hur väl testet fungerar för tre olika robotar.

Test	Robot	Hastighet	Varvtid	Avvikelse
Test 1	nr 37	1.13m/s	11.85s	3cm
Test 2	nr 42	1.12m/s	11.96s	47cm
Test 3	nr 59	1.12m/s	11.98s	51cm

Tabell 5: Resultatet för testet med körning i en åttaformad bana. Notera hur avvikelsen från startpositionen skiljer sig åt i en stor grad för de olika robotarna.

Slutsats

Alla tre robotar klarar av att köra i en åttaformad bana, dock med stor avvikelse ($\approx 50cm$) från utgångspunkten efter ett varv för två utav de tre, medan den tredje kör med en mycket låg avvikelse ($\approx 3cm$).

Topphastighet raksträcka

Syfte

Syftet med testet är att mäta den maximala hastighet som roboten kan köra i på en raksträcka.

Kravspecifikation

- Roboten ska ha en topphastighet på minst 1 m/s när den kör rakt framåt.

Genomförande

Markerar upp en sträcka på två meter och mäter hur lång tid roboten tar på sig att köra denna sträcka. För att roboten ska hålla en konstant fart under denna sträcka accelereras roboten upp till toppfart innan den första markeringen och börjar retardationen efter slutmarkeringen. Hastigheten v räknas ut med följande formel:

$$v = \frac{s}{\Delta t}$$

där $s = 2 \text{ m}$ och Δt är tiden det tar för roboten att köra sträckan s .

Resultat

Medelvärde av topphastigheten är 1.68 m/s, de olika testfallen visas i sin helhet i tabell 6

Test	Robot	Distans	Tid	Hastighet
Test 1	nr 37	2 m	1.26 s	1.59 m/s
Test 2	nr 37	2 m	1.19 s	1.68 m/s
Test 3	nr 37	2 m	1.17 s	1.71 m/s
Test 4	nr 37	2 m	1.16 s	1.72 m/s
Test 5	nr 37	2 m	1.19 s	1.68 m/s

Tabell 6: Här visas de uppmätta topphastigheterna för olika testfall.

Slutsats

Roboten klarar med god marginal det uppsatta kravet på en topphastighet på 1 m/s.

Rotation kring egen axel

Syfte

Syftet med testet är att mäta noggrannheten för robotens rotation kring sin egen axel.

Kravspecifikation

- Roboten ska kunna utföra en rotation kring sin egen axel med högst 10° absolut avvikelse ifrån önskad rotation vid en rotation på maximalt 360° .

Genomförande

Roboten roterar 360° respektive 180° kring sin egen axel. Den nya positionen läses av och den eventuella avvikelsen från den önskade rotationen utläses med hjälp av en gradskiva. Gradskivan fästs på roboten och roboten placeras så att skillnaden i grader före och efter rotationen kan läsas av via en markerad linje på golvet.

Resultat

I tabell 7 visas utfallet av testet.

Test	Rotationsgrader	Rotationsriktning	Avvikelse
Test 1	180°	höger	5°
Test 2	180°	höger	3°
Test 3	180°	vänster	2°
Test 4	180°	vänster	4°
Test 5	360°	höger	4°
Test 6	360°	höger	7°
Test 7	360°	vänster	8°
Test 8	360°	vänster	10°

Tabell 7: Här visas avvikelsen i rotationsgrader vid både rotation åt vänster och höger för 180° respektive 360°

Slutsats

Roboten klarar av det uppsatta kravet på maximalt 10° absolut avvikelse för rotation kring egen axel.

Körning i en böjd bana

Syfte

Syftet med testet är att mäta noggrannheten för robotens körning längs en böjd bana.

Kravspecifikation

- Roboten ska kunna köra i en böjd bana med given svängradie med högst 10% relativ avvikelse.

Genomförande

Roboten kommer köra längs en cirkelbåge med radien 1 m respektive 2 m. Efter avslutad sväng kommer roboten ha roterat 90° gentemot startpositionen. Testet kommer utföras både med vänstersväng och högersväng.

Resultat

Utfallet av testet visas i tabell 8

Test	Svängradie	Riktning sväng	Avvikelse	Relativ avvikelse
Test 1	1 m	höger	4 cm	4 %
Test 2	1 m	höger	1 cm	1 %
Test 3	1 m	vänster	2 cm	2 %
Test 4	1 m	vänster	3 cm	3 %
Test 5	2 m	höger	3 cm	1,5%
Test 6	2 m	höger	2 cm	1 %
Test 7	2 m	vänster	4 cm	2 %
Test 8	2 m	vänster	5 cm	2,5 %

Tabell 8: Här visas noggrannheten för körning i en böjd bana med olika radie samt åt olika håll.

Slutsats

Roboten klarar av att köra i en böjd bana med mindre än 10% relativ avvikelse.

Stabilisering efter störning av robot

Syfte/Hypotes

Syftet med testet är att mäta robotens förmåga att reglera och hålla sig i god balans efter rimlig tid samt hur stora störningar den klarar av.

Kravspekifikation

- Roboten ska kunna stabilisera sig efter en störning på en lutning av 10° orsakad av knuff.
- Roboten ska stabilisera till stillastående position efter en störning på högst 1 sekund.

Genomförande

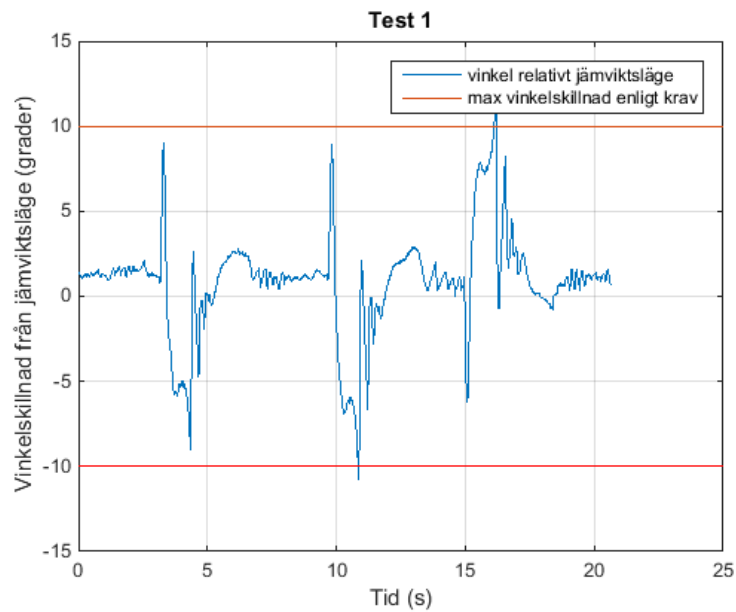
Då det intressanta med testet är att se hur väl roboten klarar av att återfå balansen efter en störning i form av en knuff kopplas positionsregulatorn bort. Detta för att vi inte vill ha tiden det tar för roboten att återfå positionen utan endast den tid det tar för den att stabilisera sig efter en störning. Genom att skriva ut tid och vinkelskillnad ifrån jämviktsläget kontinuerligt samlas data in för vidare analys via Matlab. Roboten får knuffar medan mätvärdena inhämtas.

Resultat

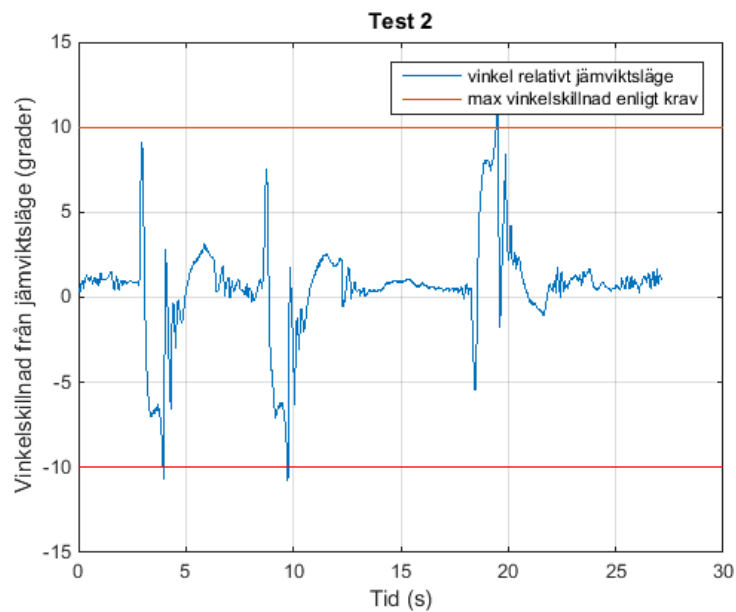
Vid den första testomgången, se figur 25, var störningen av vinkeln gentemot jämviktsläget större än 10° vid två tillfällen och det högsta uppmätta värdet på vinkeln var $11,51^\circ$. Vid den andra testomgången, se figur 26, var störningen av vinkeln gentemot jämviktsläget större än 10° vid varje knuff av roboten. Den största vinkeln uppmättes till $12,61^\circ$. Av figur 25 och 26 kan det avläsas att det tar ca 4 sekunder för roboten att stabilisera sig efter en störning på omkring 10 grader.

Slutsats

Roboten klarar av kravet att kunna stabilisera sig efter en störning på 10° grader, men den klarar inte av det andra kravet med hänseende på att stabilisera sig inom en sekund. Kravet definierar inte hur stor störningen ska vara och därmed är det möjligt att roboten kan stabilisera sig fortare om störningen är i mindre skala.



Figur 25: Test nr 1 av stabilisering efter störning.



Figur 26: Test nr 2 av stabilisering efter störning.

Mätning av riktning på objekt relativt robot

Syfte

Syftet med testet är att se hur väl det fungerar att mäta riktningen på objekt relativt roboten med hjälp av Pixy.

Kravspecifikation

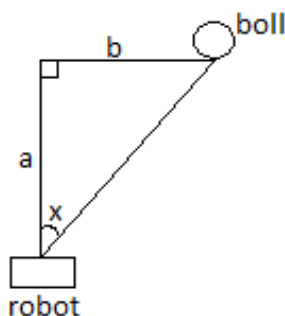
- Roboten ska kunna avgöra riktningen på objekt relativt roboten med en avvikelse på högst 10° med hjälp av Pixy.

Genomförande

Via programmeringen har en metod tagits fram för att avgöra hur stor vinkel ett objekt befinner sig ifrån en centrerad linje mitt i Pixy-kamerans synfält. Denna vinkel jämförs sedan med den uppmätta vinkeln x i figur 27 vilken tas fram genom formeln:

$$x = \arctan\left(\frac{b}{a}\right)$$

Vid testet används en konstant sträcka på 1 m för sträcka a medan b varieras inom ett intervall på 0-0.75 m med steg på 0.15 m.



Figur 27: Illustration av test för riktning av objekt relativt robot

Resultat

Tabell 9 nedan visar resultaten och det kan konstateras att vinkeln som fås via roboten skiljer sig med ca 2-3 grader mot den uppmätta vinkeln.

Test	Vinkel från Pixy	Uppmätt vinkel	Skillnad
Test1	0°	0°	0°
Test2	5°	8.5°	3.5°
Test3	14°	16.7°	2.7°
Test4	22°	24.2°	2.2°
Test5	29°	31°	2°
Test6	34°	36.9°	2.9°

Tabell 9: Tabellen visar den uppmätta vinkeln med Pixy respektive den verkliga vinkeln till bollen för olika vinklar.

Slutsats

Roboten uppfyller krav nummer 9 vilket är att den ska avgöra riktningen på objekt relativt roboten med en felmarginal på mindre än 10° . Detta krav uppfylls med god marginal då testen visar att felmarginalen ligger på ca 2-3 grader i genomsnitt.

Identifikation av objekt med Pixy

Syfte

Testet syftar till att se hur väl objektidentifieringen fungerar med Pixy-kameran.

Kravspecifikation

- Med hjälp av Pixy ska roboten kunna identifiera 6 olika objekt samtidigt på ett avstånd av 2 meter.

Genomförande

Då Pixy lär sig att känna igen olika objekt via färger som sparas som signaturer används sex olikfärgade A4-papper ($21\text{cm} \times 30\text{cm}$) som objekt vid detta test. Dessa fästs mot en vit bakgrund med ett visst mellanrum mellan varandra. Därefter kalibreras Pixy efter de rådande förhållandena och sedan testas hur väl Pixy känner igen färgsignaturerna på ett avstånd på två meter.

Resultat

Pixy upptäcker de 6 olika objekten vid ett avstånd på två meter, vilket visas i figur 28.



Figur 28: Figuren visar hur de färgade papperna uppfattas i Pixy.

Slutsats

Krav nummer 7 uppfylls då Pixy kan identifiera 6 olika objekt samtidigt vid ett avstånd på 2 m. Resultatet beror dock mycket på de rådande förhållandena vid testet. Saker som ljus och kalibrering är faktorer som påverkar resultatet i en hög utsträckning.

Avståndsmätning med Pixy

Syfte

Syftet med testet är att verifiera hur väl avståndsmätningen med Pixy stämmer överens med verkligheten.

Kravspecifikation

- Roboten ska kunna mäta avstånd till boll med högst 5 cm felmarginal vid ett avstånd på 1 m.

Genomförande

För att robotarna ska kunna navigera på fotbollsplanen har en funktion tagits fram för att mäta avstånd till olika objekt utifrån teorin i avsnittet matematiska modeller, se 2.1. Vid detta test mättes avståndet till bollen vid 18 olika distanser mellan 0.3 m och 2 m där den uppmätta sträckan jämfördes med den verkliga sträckan.

Resultat

Nedanför i tabell 10 visas ett snitt på den uppmätta distansen vid varannan mät-punkt för mätt avstånd samt det verkliga avståndet.

Test	Avstånd från Pixy	Uppmätt avstånd	Skillnad
Test1	0.32 m	0.3 m	0.02 m
Test2	0.50 m	0.5 m	0 m
Test3	0.71 m	0.7 m	0.01 m
Test4	0.92 m	0.9 m	0.02 m
Test5	1.12 m	1.1 m	0.02 m
Test6	1.36 m	1.3 m	0.06 m
Test7	1.61 m	1.5 m	0.11 m
Test8	1.78 m	1.7 m	0.08 m
Test9	2.09 m	1.9 m	0.19 m

Tabell 10: Testresultat för avståndsmätning med Pixy. Visar de uppmätta avståndet respektive det riktiga avståndet samt skillnaden mellan dessa.

Slutsats

Krav nummer 8 uppfylls då den uppmätta sträckan stämmer väl överens med den verkliga sträckan upp till 1 m. Däremot börjar den uppmätta sträckan skilja sig en hel del mot den verkliga distansen efter 1 m. Detta beror på att ju längre bort ifrån Pixy som objektet är, desto större påverkar små störningar av höjden det beräknade avståndet.