Department of Mechanical Engineering
University of Technology Eindhoven
Bachelor End Project

Calibration of a shooting mechanism using a kinect camera

Author: P. van Dooren
Coordinator: J.P.F. Senden,MSc
Supervisor: dr.ir. M.J.G. van de Molengraft

Eindhoven, July 2nd, 2017

Contents

1 Introduction

The Robocup Federatio was founded to promote robotics and AI research by organising a series of publicly appealing yet challenging challenges. One of these challenges is the middle size league, where teams of five fully autonomous robots play soccer on a full size FIFA field [1]. No external sensors may be used. However the robots are allowed to communicate with one another. The robots use a pc as their main computer, most of the code runs on this.

This report will focus on the shooting mechanism of such robots. The robots use this mechanism to shoot the ball, either for a shot at the opponents goal or to pass to a team mate. The robot calculates a desired trajectory for the ball and must then provide the right inputs to the shooting mechanism to realise this trajectory. Therefore the relation between inputs and ball trajectories must be established. In other words, the shooting mechanism must be calibrated. This report will investigate the possibility of doing this calibration using a Kinect camera on the robots.

In many teams the shooting mechanism is executed as a shooting lever combined with a linear actuator. The latter is a coil powered by a high voltage capacitor. A plunger is placed inside the coil. When the capacitor is discharged the plunger will shoot out, providing the force used in the shot. The capacitor is discharged using pulse width modulation (PWM). The duty cycle with which the capacitor is discharged can be adjusted. The plunger is attached to the shooting lever, which is a rod with a pin attached to its end. This pin will make contact with the ball. The height of the shooting lever can be adjusted using a motor. This will change the contact location on the ball.
The duty cycle with which the capacitor is discharged K and the position of the shooting lever L are the two main inputs of the shooting mechanism. The trajectory of the ball can be characterised by three parameters, the initial velocity of the ball v_0, the initial angle of the trajectory with respect to the ground α_0 and the initial angle of the trajectory with respect to the yz plane of the robot β_0, see Figure 1.

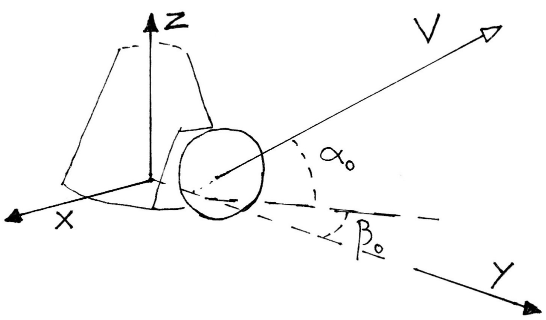

Figure 1: Conventions used throughout this report.

Tech united is a team of students and engineers of the University of Technology in Eindhoven, who participate in the middle size league. Their robots, called the Tech United Robocup Team Limited Edition (TURTLEs), will be used in this project.

In the past the relations $v_0(K, L)$ and $\alpha_0(K, L)$ have been determined using measurements from an external camera [3] and a model of the shooting mechanism [4]. However both of these approaches take a long time to execute. A much faster method to calibrate the shooting mechanism is wanted, preferably one which does not require external equipment.

Many robots in the MSL are also equiped with a kinect camera to detect the ball in 3D. This camera could be used to track the ball when it is shot by the robot allowing to measure the trajectory of the ball. This method is faster than [3] and [4] as the calibration does not require an elaborate test setup. The only equipment required is the robot itself, a ball and a field. When this method is used to measure shots with various values for K and L the relations $v_0(K, L)$ and $\alpha_0(K, L)$ can be set up in a relatively short amount of time.

The goal of this project is to determine whether or not the kinect is a viable option for calibrating the shooting mechanism. Two main questions will be discussed in this report.

1. Can the v_0 and α_0 be determined using the kinect?
2. How many shots are necessary to establish $v_0(K, L)$ and $a_0(K, L)$?

In Chapter 2 the working of the shooting mechanism will be explained in more detail as well as the working of the kinect camera. Next the method to process the kinect data to determine the ball trajectory is presented in Chapter 3. An experiment was carried out to determine the accuracy of this processing method, this is described in Chapter 4. Next the amount of measurement points necessary for the calibration is determined in Chapter 5. A final experiment was carried out to determine whether or not the calibration works properly. The results of this experiment are presented in Chapter 6

List of Symbols

Symbols

Symbol	Meaning	Unit
v_0	Initial speed of the ball	Meters per second $[\frac{m}{s}]$
α_0	Initial angle of the ball w.r.t the ground	Degrees [°]
β_0	Initial angle of the ball w.r.t the yz plane of the robot	Degrees [°]
K	Duty cycle	-
L	Lever angle	-
x	Ball position in the x direction of the robot	Meters $[m]$
y	Ball position in the y direction of the robot	Meters $[m]$
z	Ball position in the z direction of the robot	Meters $[m]$
y_{end}	y position where the ball touches the ground	Meters $[m]$
g	Gravitational acceleration constant	Meters per second squared $[\frac{m}{s^2}]$
d	Distance	Meters $[m]$
r	Pixel to Meter ratio	Meters per pixel $[m]$

2 Shooting Mechanism and Kinect

In this chapter the components that are featured in this report will be explained in further detail.

2.1 Shooting Lever

The shooting lever of the TURTLEs is an aluminium rod whose height can be adjusted using an upper rod. This upper rod is connected by wires to a motor situated behind the lever. The rotation of the motor can be read using an encoder. The wire is kept in tension by a reel on a spring. The shooting lever can be seen in Figure 2

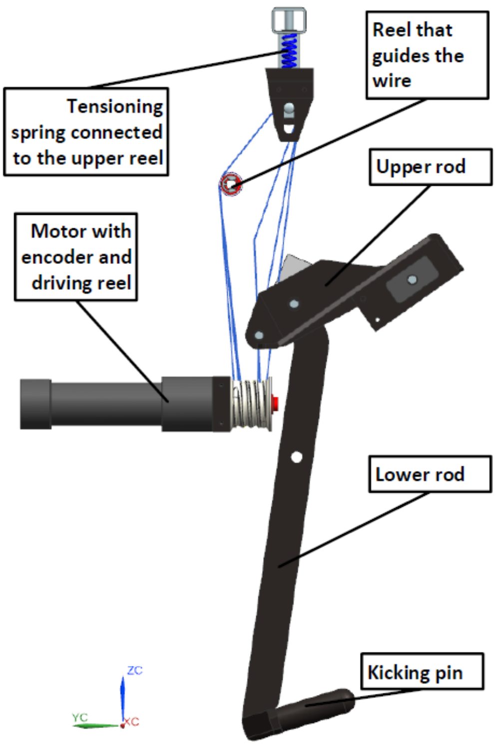

Figure 2: Shoot lever and its components [2]

Every time when the TURTLE is started it calibrates the shooting lever. It does this by raising the shooting lever until it reaches a stop. It then lowers the lever until it reaches another stop. The robot saves the encoder values for these two positions. The lever position L is defined as 0 when the lever is in the upper position and 1 when the lever is in the lower position. All other values of L are found by interpolating between the encoder values for the upper and lower positions.

Previous research has determined the variance of the shooting lever to be below 0.002. This results in a variation of end position of at most 2cm [2].

2.2 Linear actuator

The power of a shot comes from the linear actuator. This consists of a coil powered by a high voltage capacitor and a plunger which is placed inside the coil. Half of the plunger is made of a ferrous metal while the other half is made of non ferrous metal. The capacitor can be discharged to create a strong magnetic field around the coil. The ferrous part of the plunger will then be pulled into the coil. This provides the force used for the shot.
The capacitor can be charged up to 450V. It is discharged using pulse width modulation (PWM). A transistor opens and closes at a very fast rate. The duty cycle is the percentage of the period that the transistor is open. This value can range from 0 (always closed) to 1 (always open).

The robot has a sensor which measures the voltage over the capacitor. It does not measure the voltage directly. Previous research has found a relation between the voltage measured by this sensor and the actual voltage over the capacitor[2]. The relation between the duty cycle and the voltage drop over the capacitor after a shot was also established. The same research has found that the variation in duty cycle results in a maximum variation of end position of the ball of 30cm. The error increases with increasing duty cycle.

2.3 Kinect 2 camera

A kinect camera is mounted on the front end of the robot. It takes two types of images, a colour image and a depth image using infra red light. The images from the camera are not processed on the main computer of the robot. Instead a Jetson board is used to find the ball and the position data is sent to the main computer.

The kinect has multiple applications on the TURTLEs however the one used in this project is the detection of balls. The code on the Jetson board detects objects which have the same colour and shape as the ball. Because the distance to an object is known from the depth image the amount of pixels a ball would theoretically have in the colour image can be calculated. A confidence value is determined by dividing the amount of pixels by the theoretical amount. If the confidence is higher than 0.9 the object is believed to be the ball and saved. The object with the highest confidence value is communicated to the main computer. If no object has a confidence above 0.9 the Jetson will send position $[0, 0, 0]^T$ and confidence 0. There is a possibility to send more than one ball candidates however it is not currently used. In the remainder of this report the position data of the ball as seen by the kinect will be referred to as kinect balls or just balls.

3 Shot analysis algorithm

In this chapter the final version of the shot analysis script will be elaborated on.

3.1 Kinect data

In figure 3 the kinect balls detected after a shot with K=1 and L=1 is shown. Only seven balls are detected after a shot. This is because the Jetson board only sends data when the confidence of the ball is higher than 0.9. During a shot the ball has a lower confidence due to the motion blur on the ball. The code on the Jetson board was therefore changed to send balls with a confidence of 0.3 or higher because in practice an object with a confidence above 0.3 will always be found. The confidence guard was made higher in the analysis method itself. The results of this change can be seen in Figure 4, where the kinect balls detected after a shot with the same inputs is shown. A much larger portion of the trajectory is communicated to the main computer of the robot. Due to the lowered confidence guard the kinect gives some false positives. These will have to be filtered out to create an accurate estimate.

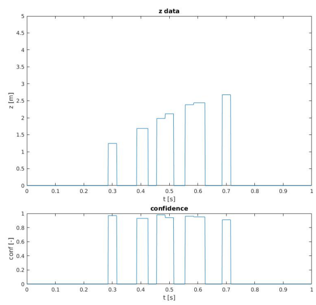

Figure 3: Data received from the Jetson board with the confidence guard set on 0.9

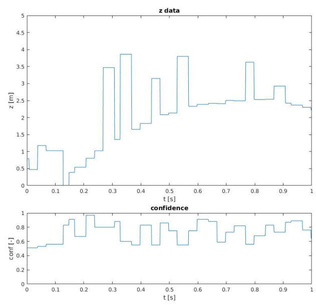

Figure 4: Data received from the Jetson board with the confidence guard set on 0.3

3.2 Analysis algorithm

The data send by the kinect must be analysed to determine the initial conditions of the ball, v_0, α_0 and β_0. The algorithm also needs to distinguish between accurate data and false positives. Furthermore it should stop the analysis at a proper moment. This could be when the ball goes out of range of the kinect or when the ball bounces on an obstacle or the floor.

The shot analysis algorithm uses an extended Kalman filter to estimate the initial conditions of the ball. The state of the Kalman filter is

$$x = \begin{bmatrix} v_0 & \alpha_0 & \beta_0 & t_{offset} & y_{offset} & x_{offset} & t \end{bmatrix}^T \tag{1}$$

where v_0 is the initial velocity of the ball, α_0 is the initial angle of the ball with respect to the ground, β_0 is the initial angle of the ball with respect to the yz plane of the robot, t_{offset} is the time delay between when the analysis starts and the ball is shot off, y_{offset} is the initial y position of the ball, x_{offset} is the initial x position of the ball and t is the current time.

The initial covariance matrix is used to tune the response of the filter. The largest covariances are those corresponding to the state variables which are expected to vary the most. These are the covariances of v_0, α_0 and β_0. The state variables y_{offset} and x_{offset} should be constants, however to accommodate slight calibration errors they have been made state variables to ensure a proper fit. Their covariance values are chosen to be very small to reflect the fact that they should be constant. The variable t_{offset} can also be considered a constant for high duty cycles. For lower duty cycles the t_{offset} varies dependant on the lever position.

The predicted measurement position p_{est} is calculated using a model of a ball travelling through the air. It is assumed that air friction can be neglected [4]. Therefore gravity is the only significant force acting on the ball.

$$h(x) = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} = \begin{bmatrix} \cos(\alpha_0)\sin(\beta_0)v_0(t - t_{offset}) + x_{offset} \\ \cos(\alpha_0)\cos(\beta_0)v_0(t - t_{offset}) + y_{offset} \\ \sin(\alpha_0)v_0(t - t_{offset}) - \frac{1}{2}g(t - t_{offset})^2 \end{bmatrix} \tag{2}$$

The extended Kalman filter also estimates the variance of the estimate. This feature is used to filter out false positives. The variance of the estimate is added to the variance of the noise on the measurements and the total is used to determine the expected standard deviation of the next measurement. Detected balls that fall outside the 95% confidence interval in any coordinate are deemed as false positives and rejected. This error bound can be seen in Figure 5. Detected balls with a confidence below 0.6 are also filtered out as the majority of balls in the trajectory have a confidence of 0.6 or above.
The algorithm stops analysing the shot when half a second has passed without new balls being found. In case of a lob shot the measurement will also stop when the predicted z coordinate of the ball falls below ground level.
When the analysis is stopped the algorithm checks how many balls were found. If this number is lower than 6 the measurement is not trusted and the analysis returns a confidence of 0.

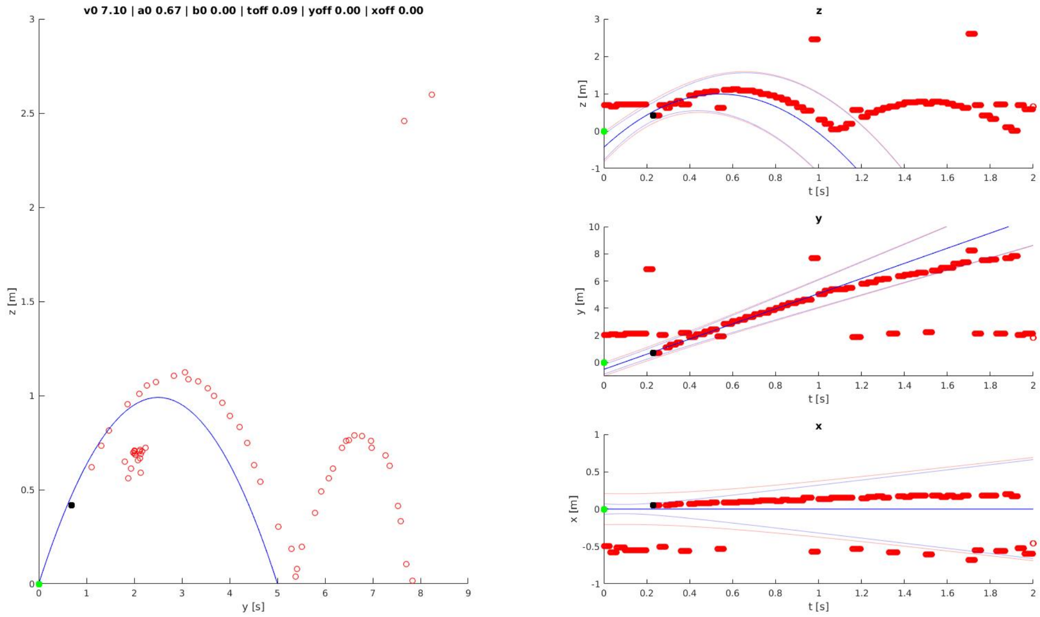

Figure 5: Visualisation of the shot analysis algorithm. Left: ball trajectory in the yz plane. Right: ball coordinates as a function of time. Blue line: current estimate, red dots: kinect balls, faint red line: error bound.

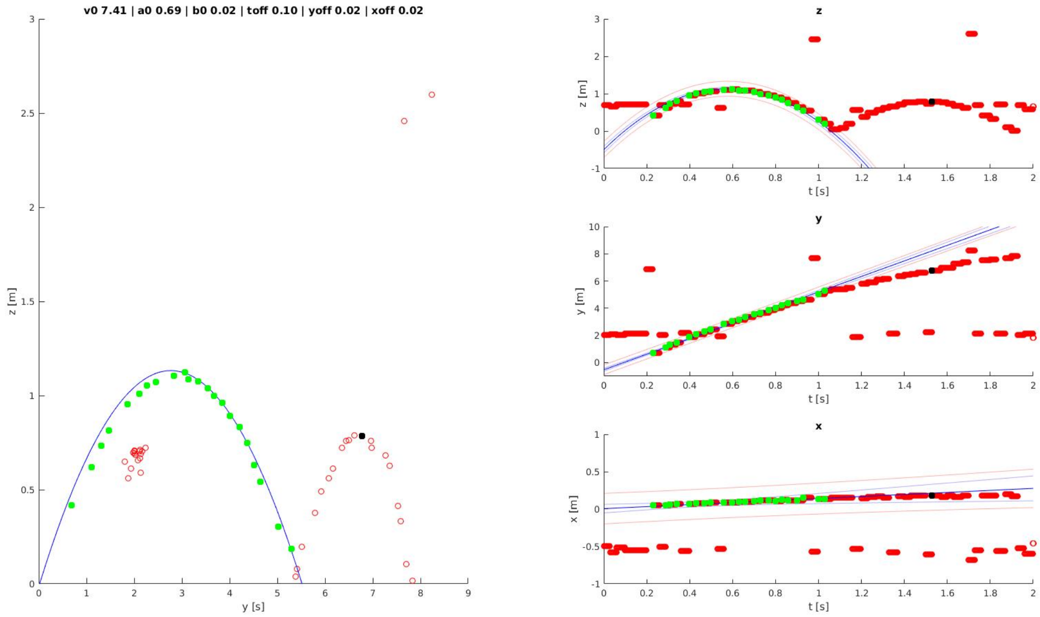

Figure 6: Visualisation of the shot analysis algorithm. Left: ball trajectory in the yz plane. Right: ball coordinates as a function of time. Blue line: current estimate, red dots: kinect balls, faint red line: error bound, green dots: accepted balls.

One downside of using this method is that it needs an initial estimate of the ball conditions for the initial state of the Kalman filter. If this estimate is off by too much the analysis will fail as the detected balls will fall outside of the error bounds and be incorrectly filtered out.

The analysis algorithm correctly filters out most balls correctly. Because the entries of the covariance matrix become smaller as more balls are found the error bounds used to filter out false positives shrink. This can be seen in Figure 6. This improves the performance of the algorithm for filtering out false positives.

4 Validation

In order to verify the accuracy of the shot analysis algorithm an experiment was carried out. A TURTLE took shots which were recorded by both the kinect and a side camera placed perpendicular to the ball path. These two data sets were then compared to each other to examine whether or not the kinect can accurately estimate the position of a ball travelling through the air. The data from the side camera was also processed to calculate the initial speed and angle of the ball. This was then compared to the initial conditions estimated by the shot analysis algorithm. In total 46 shots were taken. Shots were taken at duty cycles of 0.02 0.1 0.26 0.50 0.74 and 1.00. The shots taken at low duty cycles were used to determine the range of shots that could be analysed. The lever positions tested were 0 0.2 0.4 0.6 0.8 and 1.0. The combinations of these duty cycles and lever positions resulted in 36 shots which cover the range of duty cycles and lever positions. Furthermore for lever positions of 0.4 and 1.0 five additional shots were taken for a repeatability test, both with a duty cycle of 1.0.

4.1 Ball positions

The ball positions recorded by the kinect and by the side camera were compared. Three examples of these comparisons can be seen in the figures below. When comparing the positions of the ball, there is a difference in some of the shots. In Figure 7 the kinect detects the ball closer to the robot than it actually is. whereas in Figure 8 the kinect detects it further away.

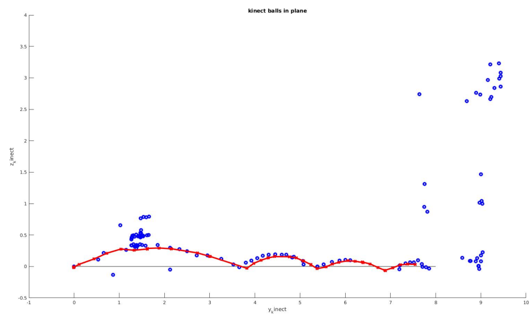

Figure 7: comparison between the data from the side camera (red) and the kinect(blue)

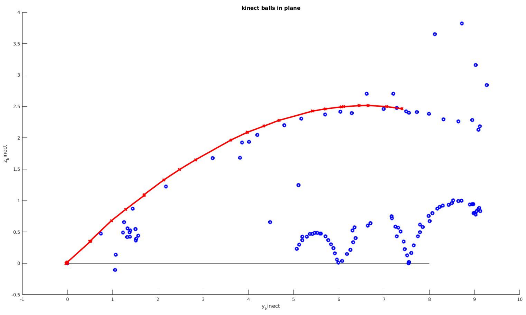

Figure 8: comparison between the data from the side camera (red) and the kinect(blue)

This appears to be an error in the y direction. This could indicate that the depth image is not properly calibrated. This was tested by placing the ball at known distances from the TURTLE and comparing this distance with the y coordinate of the kinect balls. The kinect data matched the actual distances measured.

The error in y direction could also be caused by a de-synchronisation between the colour and depth images of the kinect. This would imply an offset that scales linearly with the speed of the ball in the y direction. To test this hypothesis, the offset in y was plotted against the initial conditions of the ball. The offset remains below 5cm and does not correlate with v_0 as initially thought but instead it correlates with α_0. When the data from the kinect and the side camera was compared again after taking into account the offset the problems of Figures 7 and 8 still appeared. It is therefore unlikely that the error is due to de-synchronisation.

Another possible explanation is that some pixels of the depth image which measure the background are included when calculating the distance to the ball. There is a filter on the Jetson board which should prevent this, however it is possible that it does not work for a ball traveling with high speed through the air. This inclusion of background pixels could explain why the ball is seen further away in Figure 8 however it does not explain why the ball is seen closer in Figure 7. Therefore this is also unlikely to be the cause.

4.2 Initial conditions

To determine the initial conditions from the side camera data the data was fitted against the ball model of equation 2. If β_0 is assumed to be 0 the second and third index of H can be combined into

$$z = \tan(\alpha_0)v_0 - \frac{1}{2}\frac{g}{\cos^2(\alpha_0){v_0}^2}y^2 \tag{3}$$

a 2^{nd} degree polynomial was fitted through the y and z data. The initial conditions were then calculated using

$$z = c_1y^2 + c_2y + c_3 \tag{4}$$

$$\alpha_0 = \arctan(c_2) \tag{5}$$

$$v_0 = \sqrt{-\frac{1}{2} g \frac{1}{c_1 \cos(\alpha_0)^2}} \tag{6}$$

The data of shots with the lowest duty cycles could not be processed due to the trajectory of the ball being very short.

The data of shots with a lever position of 0 were processed by fitting the time and y data of the shots against the second index of H. This was done using a linear fit. The time of the side camera data was determined by assuming a frame rate of 30Hz. α_0 and β_0 were assumed to be 0. v_0 was then determined using

$$y = c_1 y + c_2 \tag{7}$$

$$v_0 = c_1 \tag{8}$$

The initial conditions calculated from the side camera data were compared to the initial conditions estimated by the shot analysis algorithm. The results can be seen in Tables 1 and 2.

Table 1: initial speed of the ball

K	0.02			0.1			0.26		
L	v0 camera	v0 kinect	error v0	v0 camera	v0 kinect	error v0	v0 camera	v0 kinect	error v0
0	1.9583	1.6001	-0.3583	3.0669	2.8523	-0.2146	5.8643	7.2134	-0.7852
0.2	-	-	-	-	-	-	8.1653	7.2722	-0.8931
0.4	-	-	-	-	4.5225	-	7.5900	7.3034	-0.2866
0.6	-	-	-	4.3250	4.2184	-0.1066	7.2811	7.1151	-0.1660
0.8	-	-	-	4.5758	4.4539	-0.1219	7.3527	7.2539	-0.0987
1.0	2.1018	-	-	4.3467	4.3678	0.0211	7.1219	7.0983	-0.0236
K	0.5			0.74			1.0		
L	v0 camera	v0 kinect	error v0	v0 camera	v0 kinect	error v0	v0 camera	v0 kinect	error v0
0	8.9755	7.2134	-1.7622	9.5934	7.9413	-1.6522	10.4746	8.3070	-2.1677
0.2	8.9755	9.6828	-0.9385	12.3959	10.6527	-1.7431	11.1232	10.8879	-0.2353
0.4	10.4839	10.1043	-0.3797	11.5664	11.4555	-0.1110	12.3073	11.8838	-0.4235
0.6	10.1970	10.1054	-0.0916	12.0899	11.6978	-0.3921	12.7089	12.4646	-0.2443
0.8	9.7868	9.7734	-0.0134	11.4316	11.3461	-0.0856	11.8072	11.8363	0.0291
1.0	9.6006	9.8591	0.2586	10.7259	10.9344	0.2085	11.1463	11.4586	0.3123

Table 2: Initial angle of the ball

K	0.02			0.1			0.26		
L	a0 camera	a0 kinect	error a0	a0 camera	a0 kinect	error a0	a0 camera	a0 kinect	error a0
0	0	0.0576	0.0576	0	-0.0702	-0.0702	0	0.0011	0.0011
0.2	-	-	-	-	-	-	11.8691	14.2339	2.3648
0.4	-	-	-	-	18.4034	-	18.8075	19.8972	1.0897
0.6	-	-	-	25.7644	27.8680	2.1035	27.2603	28.0171	0.7569
0.8	-	-	-	32.2627	34.7905	2.5278	34.8021	35.4646	0.6625
1.0	33.2931	-	-	43.5634	42.1690	-1.3944	43.7586	42.8066	-0.9520
K	0.5			0.74			1.0		
L	a0 camera	a0 kinect	error a0	a0 camera	a0 kinect	error a0	a0 camera	a0 kinect	error a0
0	0	0.1322	0.1322	0	0.1587	0.1587	0	0.1750	0.1750
0.2	11.4994	13.3319	1.8325	9.7499	11.5004	1.7505	10.9252	11.4936	0.5684
0.4	19.8551	20.9773	1.1222	20.0720	20.4573	0.3854	18.9133	19.9724	1.0591
0.6	28.2126	28.9816	0.7690	28.4712	29.6386	1.1674	27.9456	28.3105	0.3649
0.8	37.3647	37.5612	0.1965	37.1150	37.0606	-0.0544	36.4159	36.0250	-0.3909
1.0	40.7281	39.1396	-1.5885	41.3601	40.0137	-1.3464	39.8267	37.7041	-2.1226

The initial conditions calculated from both methods match closely for certain shots. v_0 can be measured up to 0.3m/s accurate and α_0 can be measured up to an accuracy of 2°. Some shots however display larger errors. The largest errors appear to occur for low duty cycles and low lever positions. For low duty cycles only a small portion of the trajectory is visible to the kinect. Which may cause only 6 data points used in the analysis. The same is true for small lever positions. Large errors also occur at higher values of duty cycles and lever positions, this is thought to be caused by the inaccurate ball positions of the kinect as described in 4.1.

At low lever positions the error becomes greater. This is due to a limited portion of the ball trajectory being visible to the kinect.

There is also a large error in the initial velocity measured for passes. However the processing of the camera may not be entirely accurate if the frame rate was not exactly 30 Hz. Therefore no conclusion can be drawn from these results in regard to passes.

The end position of the ball can be predicted from the initial conditions using

$$y_{end} = \frac{\sin(\alpha_0)\cos(\alpha_0)v_0^2}{\frac{1}{2}g} \tag{9}$$

This was be used to determine the error in end position of the ball, which can be seen in Table 8.

Table 3: end position of the ball

K	0.02			0.1			0.26		
L	y camera	y kinect	error y	y camera	y kinect	error y	y camera	y kinect	error y
0	0	0.0005	0.0005	0	-0.0020	-0.0020	0	0.0001	0.0001
0.2	-	-	-	-	-	-	2.7359	2.5696	-0.1663
0.4	-	-	-	-	1.2491	-	3.5842	3.4801	-0.1042
0.6	-	-	-	1.4928	1.4991	0.0063	4.4007	4.2800	-0.1208
0.8	-	-	-	1.9268	1.8951	-0.0317	5.1654	5.0695	-0.0959
1.0	0.4132	-	-	1.9235	1.9352	0.0117	5.1656	5.1211	-0.0445
K	0.5			0.74			1.0		
L	y camera	y kinect	error y	y camera	y kinect	error y	y camera	y kinect	error y
0	0	0.0245	0.0245	0	0.0356	0.0356	0	0.0430	0.0430
0.2	4.4931	4.2889	-0.2042	5.2285	4.5201	-0.7084	4.6941	4.7192	0.0251
0.4	7.1584	6.9578	-0.2006	8.7921	8.7610	-0.0311	9.4691	9.2429	-0.2262
0.6	8.8308	8.8243	-0.0065	12.4877	11.9911	-0.4967	13.6321	13.2251	-0.4071
0.8	9.4190	9.4106	-0.0084	12.8199	12.6220	-0.1980	13.5779	13.5861	0.0082
1.0	9.2914	9.7019	0.4106	11.6328	12.0036	0.3708	12.4586	12.9525	0.4939

For many shots the end positions can be measured up to 30cm accurately. However for the same range of shots mentioned above the errors are large. The highest being 0.5 metres off.

4.3 Repeatability

The repeatability of the processing was tested by shooting five shots with the same inputs. The analysis should be able to measure a similar initial condition for each shot. This was done for two sets of inputs, lever position 0.4 and duty cycle 1 and lever position 1 and duty cycle 1. The results of this can be seen in Table 4.

Table 4: Repeatability validation

	L=0.4		L=1.0	
	kinect	camera	kinect	camera
y	9.1805	9.3459	13.4038	13.2420
σ_y	0.2996	0.3116	0.2099	0.3396
v_0	11.9730	12.1666	11.6109	11.5016
σ_{v_0}	0.0768	0.2849	0.0851	0.1516
α_0	19.4578	19.4578	38.6256	39.5403
σ_{α_0}	0.4697	0.600	0.2991	0.2269

There is a significant difference between the standard deviation of the shots as measured by the kinect and the deviation measured by the side camera. The deviation measured by the kinect is in most cases smaller than the actual deviation, which could be due to the initial estimate of the kalman filter which does not change much between shots. It can therefore be concluded that the shot analysis method influences the measured standard deviation.

5 Calibration points

A good calibration should not take long. Therefore it is important to estimate how long a calibration will take. To do this the repeatability of shots has to be analysed. This may be a function of K and L. Furthermore it is important to know how many measurements are necessary to reduce the expected error of the calibration to an acceptable level. The end position of the ball should be predicted up to 10 cm accurate.

5.1 Analysis of the repeatability of shots

To estimate the repeatability 5 shots were taken at various K and L. The standard deviation of the measured initial conditions was calculated. Although the analysis algorithm is not very accurate for some values of K and L, it is assumed that the repeatability can still be estimated. The results of the repeatability analysis can be seen in the tables below.

Table 5: Repeatability of v_0

L \K	0.2	0.4	0.6	0.8	1.0
0	0.0975	0.0616	0.1662	0.1767	0.1747
0.2	0.0521	0.2103	0.2496	0.1838	0.1412
0.4	0.1200	0.2907	0.3489	0.1423	0.2886
0.6	0.1636	0.2439	0.2154	0.3740	0.2053
0.8	0.0760	0.0623	0.1658	0.1706	0.2421
1.0	0.1633	0.0737	0.1398	0.1570	0.2989

Table 6: Repeatability of α_0

L \K	0.2	0.4	0.6	0.8	1.0
0	0.1055	0.0542	0.0426	0.0913	0.0594
0.2	0.3365	0.4725	0.7970	0.6265	0.4818
0.4	0.4897	0.5962	0.5003	0.6951	0.6118
0.6	1.2771	0.5999	0.8642	1.2015	1.2067
0.8	0.7449	0.7288	0.7064	0.5769	1.4356
1.0	0.7210	0.6928	0.8622	1.1244	2.8313

The estimated v_0 and α_0 are not independent due to the analysis method. Therefore the variation in the end position of the ball cannot be calculated from the variations in v_0 and α_0. Instead the end position for each measurement is calculated using 9. This data is then used to calculate the standard deviation of the end position. The results of this can be seen in Table 7

Table 7: Repeatablity y_{end}

L \K	0.2	0.4	0.6	0.8	1.0
0	0.0050	0.0054	0.0045	0.0136	0.0095
0.2	0.0451	0.1142	0.1909	0.2058	0.1112
0.4	0.0755	0.2594	0.4599	0.1281	0.5656
0.6	0.1446	0.3036	0.4741	0.7002	0.3966
0.8	0.0499	0.0822	0.2765	0.3304	0.3797
1.0	0.2065	0.1268	0.2655	0.2980	0.5039

The average standard deviation in the end position is 27cm and the maximum is 70cm. This is a significant deviation and much larger than predicted based on the deviations in K and L from [2].

There is not a clear relation between the data in Tables 5, 6 and 7 and the duty cycle and lever angle. This is most likely because 5 measurement points is a small number for calculating the standard deviation. Nevertheless an estimate of the relation was made. 4^{th} Degree polynomials were fitted through the data, which can be seen in Figure 9, 10 and 11. These relations will be used to simulate the calibration later on in this chapter.

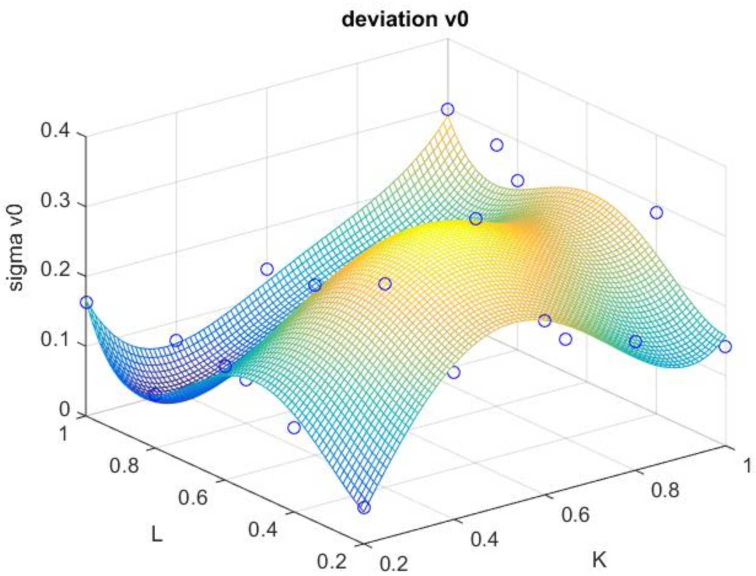

Figure 9: The standard deviation of v_0

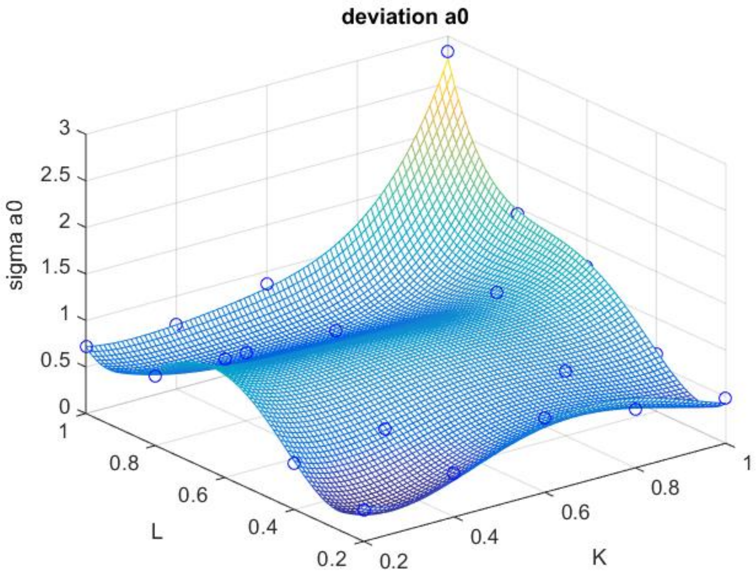

Figure 10: The standard deviation of α_0

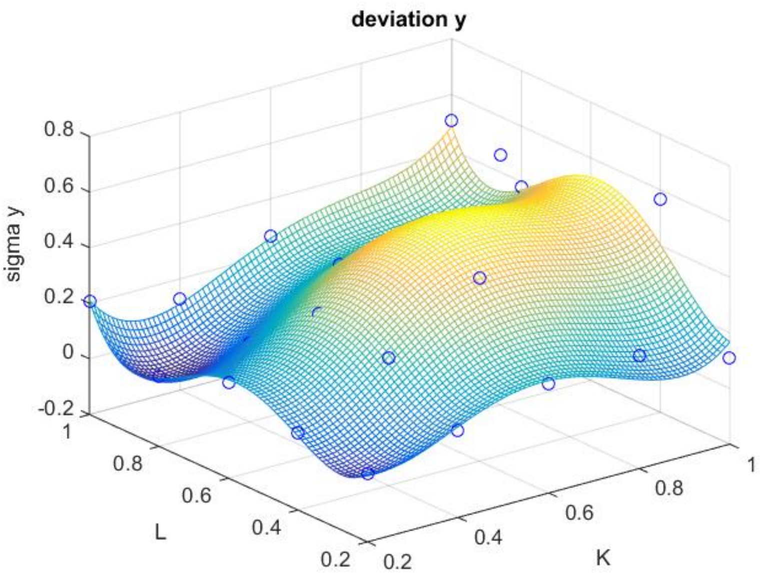

Figure 11: The standard deviation of y_{end}

5.2 Optimising calibration points

The most straightforward choice of calibration points is having them evenly spaced over the range of K and L. However for values of K and L where the deviations in $v0$ and α_0 are small require fewer calibration points than values of K and L where the deviations are large. The calibration could be done more efficiently by strategically choosing the points used for the calibration.
When the standard deviation on the independent variables is negligible the underlying curve $v_0(K, L)/\alpha_0(K, L)$ does not influence the standard deviation of the estimate. This is the case for the shooting mechanism as proven by [2]. Therefore the deviation of the estimate is only a function of the deviation curve and the choice of measurement points.

Because the math involved in calculating the standard deviation curve after a polynomial fit is complicated a numerical simulation was chosen over an analytical solution. As a proof of concept a simulation was made in one dimension. Three distributions of calibration points were tested: Evenly spaced points, points placed near the edges of the calibrated range and a distribution with more points near the greatest deviation. Figures 12, 13 and 14 show the standard deviation of the curve after the polynomial fit and Figure 15 compares the different distributions.

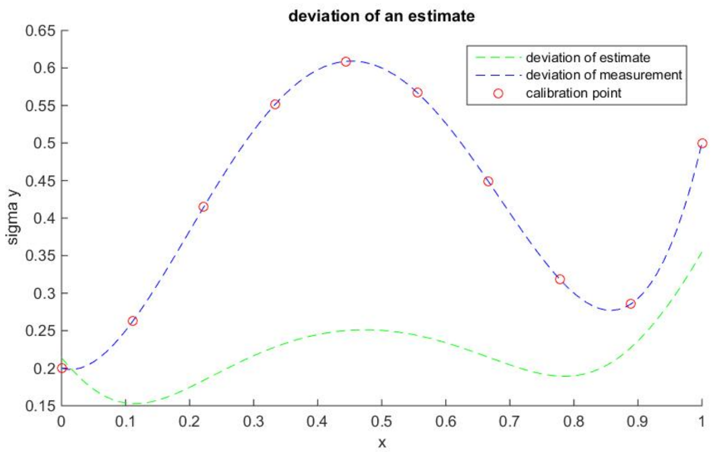

Figure 12: The standard deviation of the estimate of y with calibration points evenly spaced.

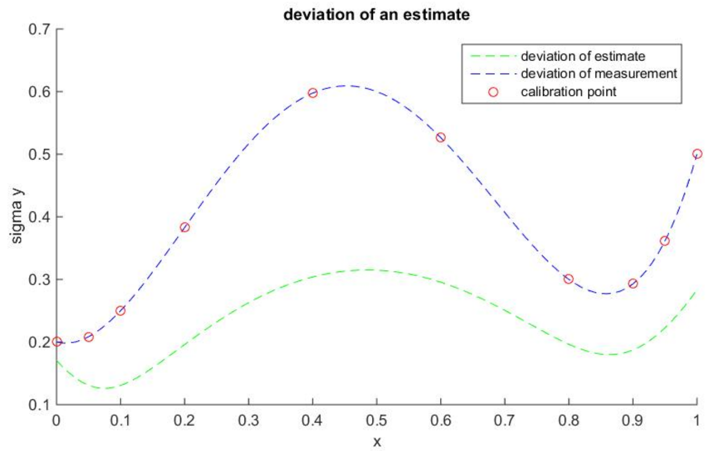

Figure 13: The standard deviation of the estimate of y with calibration points concentrated near the edges of the domain.

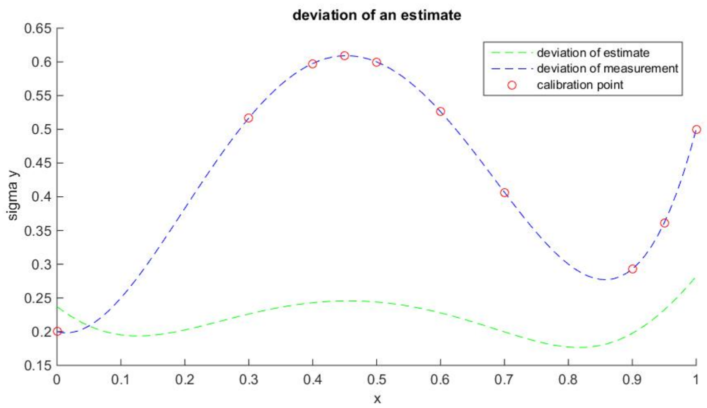

Figure 14: The standard deviation of the estimate of y with calibration points consentrated where the deviation of the measurements is largest.

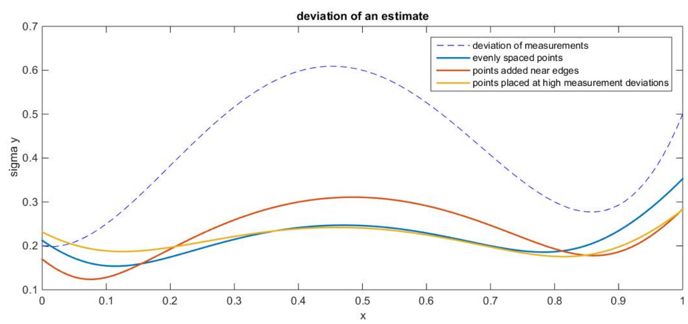

Figure 15: The standard deviation of the estimate of y.

The standard deviation of the estimate is definitely affected by the choice of calibration points. Choosing the calibration points concentrated around areas of high deviation can be used to make the deviation of the estimate more even. However the improvement compared with the evenly spaced points is small. The median deviation for estimates in figures 12, 13 and 14 are 0.2148, 0.2369 and 2.132 respectively. Therefore the median deviation for the estimate does not improve significantly.

5.3 Number of calibration points necessary

Besides the placement of the calibration points another important factor is the number of points. The number should be large enough so that the standard deviation of the estimate is sufficiently small. However it should also be small enough so that the calibration can be performed in a reasonable amount of time.

The simulation described in Section 5.2 was expanded to two dimensions. The calibration points were chosen to be evenly spaced. The simulation was run for 25, 49 and 100 points. These calibrations would take about 15 minutes, 30 minutes and 60 minutes respectively. In Figures 16, 17 and 18 the standard deviation of the estimated end position is shown.

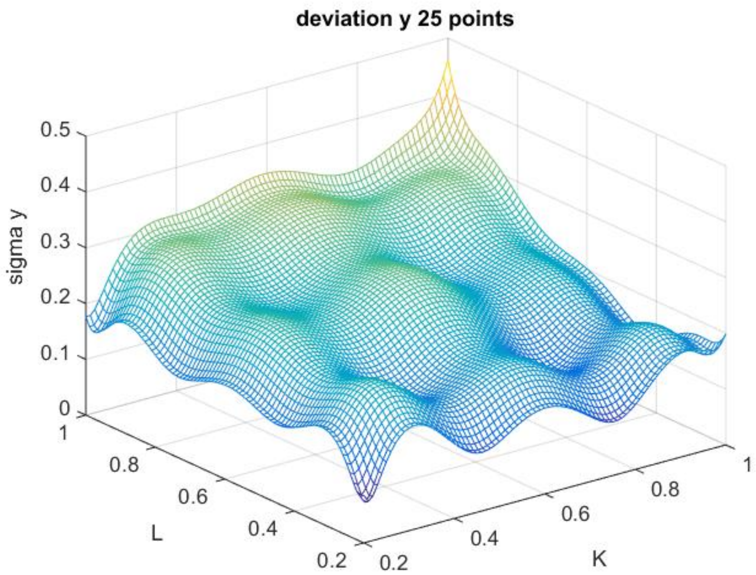

Figure 16: The standard deviation of the estimate of y after a calibration with 25 points.

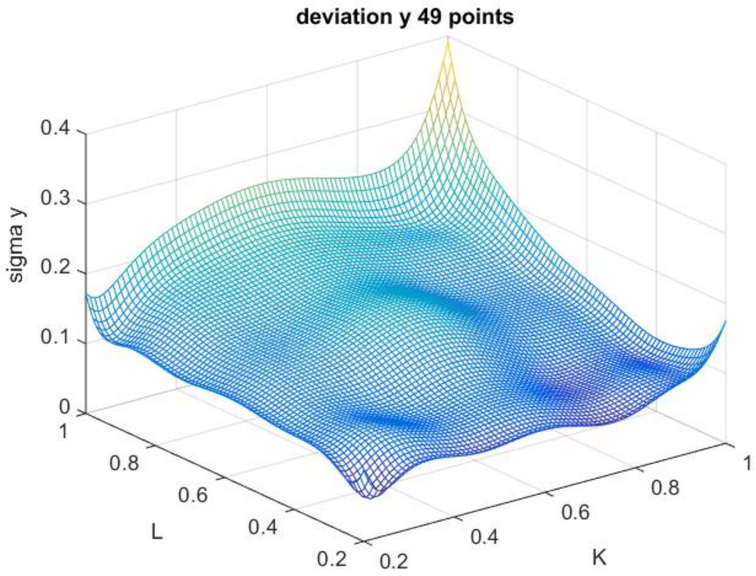

Figure 17: The standard deviation of the estimate of y after a calibration with 49 points.

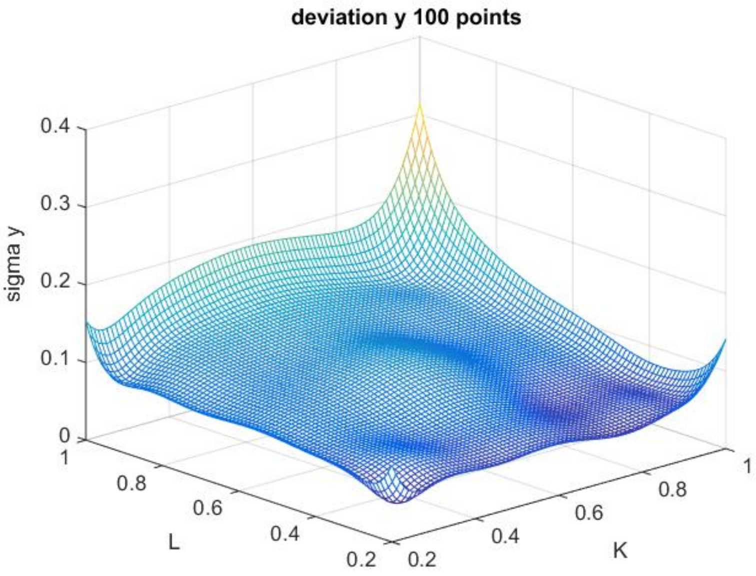

Figure 18: The standard deviation of the estimate of y after a calibration with 100 points.

The median deviation for 25 points is 0.2292 m, for 49 points 0.1391m and for 100 points 0.0951m. More calibration points improve the accuracy of the

estimate. 25 Calibration points will also cause the deviation of the estimate to assume a wavy pattern because the polynomial fit has 15 coefficients. More calibration points smooth out the deviation curve.
It takes around 100 calibration points to reduce the standard deviation of the end position below 10cm. This will take an hour every time a robot needs to be calibrated. This is too long as five robots will have to be calibrated each time. To make the calibration more time-efficient the repeatability of the shots could be improved. This way fewer points will be necessary to achieve the desired accuracy. Another possibility is to design the calibration in such a way that portions of the curves can be calibrated one at a time, for example using another extended Kalman filter. This would allow to calibrate a smaller range which would also require fewer points.

6 Experiment

To test whether or not the calibration of the shooting mechanism provides sufficient accuracy an experiment was carried out. The shooting mechanism was calibrated using 210 shots spread out over a range of duty cycles and lever positions. A fourth degree polynomial was fitted through the measured velocities and angles. The fit had an adjusted R^2 of 0.9854 so it is assumed to be accurate. For six different combinations of duty cycle and lever angle the location where the ball will land on the ground was predicted. For each combination five shots were taken. A camera was placed near the predicted landing spot to measure the place of impact. The actual landing location of the ball was then compared to the predicted one. These shots were also analysed using the algorithm described in 3. Based on the measured v_0 and α_0 another estimate of the landing position was obtained. The results of this experiment can be seen in Table 8.

Table 8: Predicted and actual landing positions

K	L	$y_{predicted}$	y_{Video}	σ_{yVideo}	y_{Kinect}	$\sigma_{yKinect}$
-	-	m	m	m	m	m
0.5	0.65	7.28	8.616	0.0843	8.6076	0.0842
0.34	0.9	5.74	6.4975	0.0962	6.5392	0.0937
0.74	0.42	6.82	7.1175	0.2465	6.8890	0.2310
1	0.42	8.79	8.8675	0.0758	8.6848	0.1205
0.28	0.42	2.76	3.5975	0.0742	3.3700	0.1788
0.4	0.6	5.6	6.3475	0.1710	6.1876	0.2042

The predicted landing locations were not reached in the experiment. The robot shot further than predicted. However the analysis of these shots was able to correctly estimate the landing positions. This would suggest that the situation during the calibration was different from the situation during the experiment. During the calibration the ball was shot from the penalty dot, which has a slightly different texture from the rest of the soccer field. This may have resulted in a difference in the calibration. However when comparing the calibration done for the experiment against other calibrations done over the course of this project the difference found did not explain the large differences in landing location. Therefore the experiment was most likely not carried out properly. However from the fact that the analysis algorithm was able to predict the landing positions it is believed that the experiment would have been successful if the conditions were the same as during the calibration.

7 Conclusion

The possibility of using the kinect camera to calibrate the shooting mechanism was investigated. A method was developed to measure the ball trajectory and a prediction was made on how many measurement points would be required for a calibration.

The method developed to measure the trajectory of a ball shot by a soccer robot uses a Kalman filter to estimate the initial velocity and angles of the ball. The covariance matrix of the filter is used to calculate an error bound which is used to filter out false positives.

Experiments have proven that the method is accurate for certain combinations of duty cycle and lever position. It is able to predict the landing position of the ball with an accuracy of 30cm. For shots with low duty cycles and lever positions the analysis will give larger errors due to a smaller portion of the trajectory being visible to the kinect. For larger duty cycles and lever position the error will also increase. The reason for this is that the kinect camera is not able to accurately detect balls moving at a higher velocity. The exact cause of this problem is not yet found.

The deviation of shots was used to determine the amount of measurement points necessary for a calibration. In order to reach a median deviation of less than 10cm about 100 measurement points are needed. This is a lot considering five robots will need to be calibrated. This could be solved by improving the reproducibility of the shots. However it is also possible to calibrate the robots more efficiently if only a portion of the v_0 and α_0 curves can be calibrated.

The experiment to validate the calibration was not executed properly. The predicted end positions were not reached by a large margin, however the analysis method run on the same shots was able to predict the end position accurately.

The kinect can be used to calibrate the shooting mechanism. However the accuracy is currently limited by the ability of the kinect to detect fast moving balls. It also takes about one hour to calibrate a robot. However this could be improved either by improving the reproducibility or by calibrating only a portion of the curves.

7.1 Recommendations

The kinect camera is not able to detect balls moving at a high velocity. The kinect also gives a much lower confidence for a ball in the air compared to a ball placed on the floor. The origin of these problems should be found and the ball detection should be improved. This will improve the accuracy of the ball detection algorithm. The y_offset and x_offset can then be made constant in the Kalman filter.

The reproducibility of shots is now thought to be mainly caused by the variation in the capacitors voltage [2]. If this is improved the calibration could be done much faster.

The calibration of the shooting mechanism now relies on gathering a large data set and then fitting a polynomial through the points. This could be made better by implementing a Kalman filter, which would not require data to be saved. This would also allow to calibrate only a portion of the curves.

The shot analysis algorithm could still be improved upon. The algorithm is not able to detect all the faulty measurements using only a guard on the number of balls detected. The accuracy of the calibration should also be validated again as the experiment to determine this was not executed properly.

References

[1] www.Robocup.org/leagues/6 *RoboCupSoccer - Middle Size*, 2017.

[2] Sam Lousberg, Jordy Senden and René van de Molengraft *Analyzing and improving the reproducibility of shots taken by a soccer robot*, Eindhoven, 2017.

[3] Antoine de Bock, Ferry Schoenmakers and René van de Molengraft *Designing and implementing a skill for shooting at 3D targets for Tech United Eindhoven soccer robots*, Eindhoven, 2015.

[4] Jordy Senden, Yanick Douven and René van de Molengraft *A Model-Based Approach to Reach a 3D Target with a Soccer Ball, Kicked by a Soccer Robot*, Eindhoven, 2016.

A Experiment setup

An experiment was carried out to validate the accuracy of the shot analysis method. In this section the setup of that experiment is discussed in detail.

A.1 Experiment 1

In the first experiment the TURTLE was placed at the edge of the soccer field. Its orientation was chosen so that it would shoot along the length of the field. Opposite the field a camera was placed on a tripod perpendicular to the ball trajectory. The kinect data from the TURTLE was logged and the camera opposite the field recorded the shots. Figure 19 is a snapshot of this side camera.

Figure 19: Experiment setup as seen from the side camera

A scale was added to the frame of the camera by two pieces of tape spaced 5 metres apart. This was used to determine the pixel to meter ratio. The sizes of the signs behind the robot were also measured to verify this. There was a significant difference between the pixel to meter ratios obtained from the pieces of tape and the signs on the background. Therefore the ratio used was interpolated using

$$r = \frac{d_{r,b}}{d_{t,b}} r_{tape} + \frac{d_{r,t}}{d_{t,b}} r_{signs} \tag{10}$$

where r is the pixel to meter ratio used, r_{tape} the pixel to meter ratio based on the distance between the two pieces of tape, r_{signs} the pixel to meter ratio based on the signs in the background, $d_{r,t}$ the distance between the center of the robot and the white line on which the tape is placed, $d_{r,b}$ the distance between the center of the robot and the background on which the signs are hung and $d_{t,b}$ the distance between the white line and the background.

The data logged by the TURTLE was processed using the shot analysis algorithm described in Chapter 3. The videos from the camera were processed using an algorithm developed in Matlab by [4]. This algorithm provided the coordinates of the ball in y and z for every frame of the video. The processing algorithm occasionally made errors in detecting the ball. By visually comparing the data found by the algorithm to the video of the ball, these errors were removed from the dataset. This process is illustrated in Figure 20

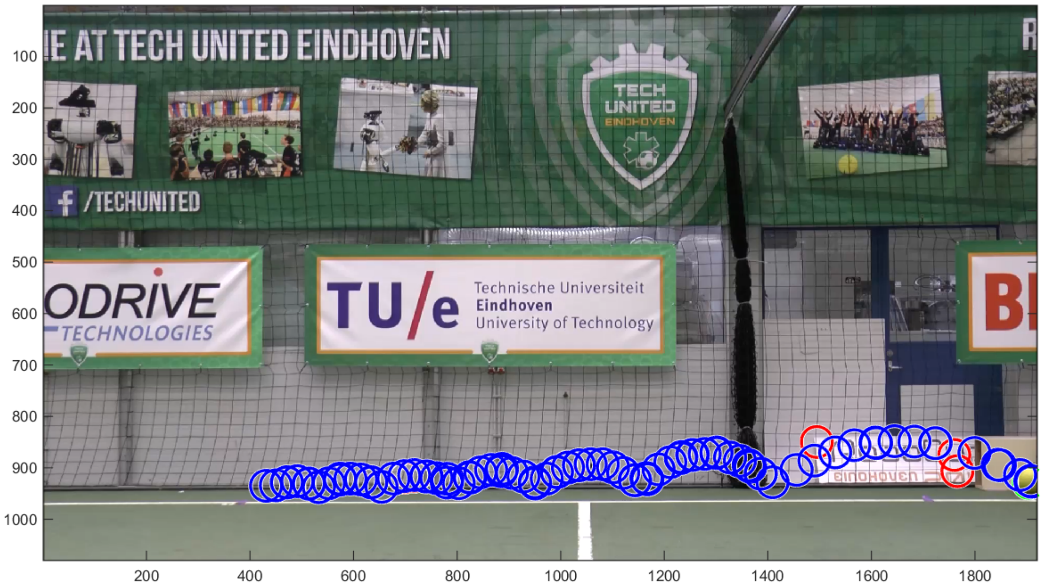

Figure 20: Balls found by the video processing algorithm. Red balls have been removed manually.

A.2 Experiment 2

In the second experiment the end position of the ball was predicted using the calibration. To verify the accuracy of these predictions the landing positions had to be measured. This was done using a similar setup to experiment 1. However this time the side camera was placed much closer to the path of the ball. It was placed so that the predicted landing position of the ball was in the middle of the frame. A striped board was made to be used as a scale. The stripes on this board were 5cm wide. On the soccer field lines were taped 1m apart from each other to position the smaller board in the right place.

The TURTLE was placed one meter from the first line. To confirm that the TURTLE was placed exactly one meter from this line a calibration was performed. The kinect data was logged while the ball was placed on the lines. From this data it was determined that the TURTLE was 1 metre and 13cm away from the first line. All the measured end locations were corrected for this. From the videos of each shot the frame where the ball touched the ground was selected, see Figure 21. The end position of the ball was then read from the scale. For videos where the landing of the ball was in between frames the closest frame was taken.

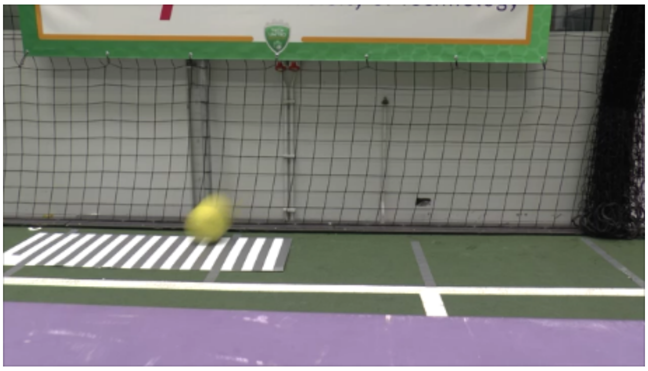

Figure 21: Experiment setup