Case 4W078

N.C.H. Hoogerwerf - 0734333

Prediction method for lob shot trajectories

Bachelor End Project

Supervisor: dr. ir. René v.d. Molengraft

Coach: Bob van Ninhuijs MSc.

Technische Universiteit Eindhoven
Department of Mechanical Engineering
Control Systems Technology Group

Eindhoven, June 29, 2014

Summary

During gameplay it often occurs that a robot and/or keeper stand between the ball and the goal. That means that a lob shot can be the solution to shoot at goal. However, the trajectory of a lob shot is highly depending on the shooting effort, ball pressure, voltage of the capacitor, etc. To determine a more robust shooting effort, a model is required of the shooting system of the robot, so that a better ball trajectory prediction can be provided.

In this project the capacitor voltage is included to the prediction model. This is a part of the prediction model that predicts the trajectory of the ball during a lob shot. Before the start of this project only the relation between the duty cycle and the distance of the lob shot is known.

First the electrical circuit is simplified to a model to find the linear relation for the current through the coil versus the capacitor voltage. The relation between the force of the solenoid and the current is quadratic. With the assumption that the distance is linear dependent on the force, because of the duty cycle versus the distance is linear, the relation between the capacitor voltage and the distance should be quadratic.

The relation between the capacitor voltage and the distance is experimentally determined. The software that controls the shooting system has been modified so that the experiment could be executed. Besides the relation, the behavior of the capacitor is analyzed. The capacitor voltage against the time is investigated and the voltage drops during a shot are also analyzed. The trajectories of the balls have shown that there is difference between the shots with the same duty cycle and capacitor voltage. Therefore some recommendations are made to give a more accurate result.

Finally, the relation for the duty cycle dependent on the distance and the capacitor voltage is found. A low level control algorithm need to make sure that the duty cycle increases when the capacitor is not fully charged.

List of Symbols

Symbol	Quantity	Unit
D	Distance	[m]
DC	Duty cycle	[-]
F	Force	[N]
I	Current	[A]
L	Inductance	[H]
RCV	Ratio Capacitor Voltage	[-]
t	Time	[s]
V	Voltage	[V]

Contents

1 Introduction

In this project the capacitor voltage is included to the prediction model for lob shot trajectories for the TURTLE of Tech United Eindhoven. The project comes from the Control Systems Technology group of the department Mechanical Engineering of TU Eindhoven.

1.1 Tech United Eindhoven

This paragraph contains a description of Tech United Eindhoven. The first part is about RoboCup. The second part is about the organization of Tech United Eindhoven and their robots. The last paragraph gives more information about the soccer robot.

1.1.1 RoboCup

RoboCup is an international research and education initiative, established in 1993. It fosters intelligent robotics research by providing a standard problem where a wide range of technologies can be integrated and examined. Society can benefit from these major innovations. [1]

The main focus of the RoboCup competitions is the game of soccer. The cooperation between multiple robotic systems in a highly dynamic environment requires a lot of new technologies.

1.1.2 Organization

The first dutch team, named Clockworks Orange, was founded in 2000 and consisted of a composed team form several universities. In 2005, when Clockworks Orange decided to quit, a new team was established by TU Eindhoven en TU Delft, ′Tech United′. This team participated in the Dutch Open Championships Roboludens. After that, TU Delft retreated and TU Eindhoven decided to continue the project. ′Tech United′ became ′Tech United Eindhoven′ in 2006, and since 2007 Tech United Eindhoven belongs to the top of robot soccer. In 2012 they have become world champion. [2]

The robots of Tech United Eindhoven [3] can be divided in three different categories: Soccer(TURTLEs), Care(AMIGO) and Humanoid(TUlip). The three robots can be seen in figure 1.1.

- TURTLE is the name of the soccer robot. The robotic soccer team consists of six TURTLEs, one goalkeeper and five field players. The TURTLEs takes part in the RoboCup Soccer League. In section 1.1.3 more information can be found about the TURTLEs.
- AMIGO is the care robot of Tech United Eindhoven. AMIGO means Autonomous Mate for IntelliGent Operations. Elderly people can function independenly for a longer time with robots like AMIGO. AMIGO learns from own events and other robots by exchanging information on a platform.
- TUlip is a humanoid robot that can walk on two legs. This robot takes also part in the RoboCup Soccer League, but he plays in the adult size against other humanoid robots. In this games the robots only take penalties.

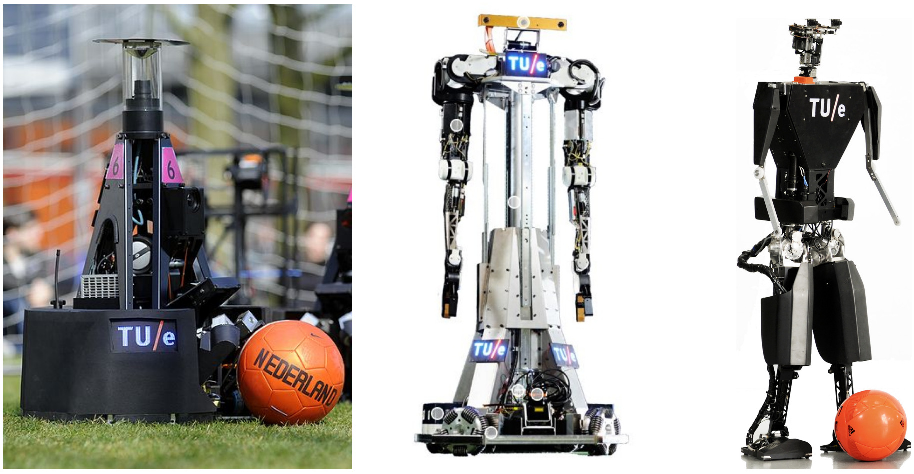

Figure 1.1: *The robots of Tech United Eindhoven; TURTLE, AMIGO and TUlip respectively.*

1.1.3 TURTLE

As mentioned before, the name of the soccer robot is TURTLE. TURTLE is an abbreviation for Tech United Robocup Team Limited Edition. The robot plays in a team with five players, four field players and one goalkeeper. The goalkeeper looks a little different, because it has a rack around itself. With this rack the keeper has more opportunity to stop the ball. The TURTLEs takes part in the middle size league. The last four year they reached the finals of the world championship. In 2012 they won the world championship. Also they won the European championship several times. [4]

A detailed description of the TURTLEs can be found on the wiki of Tech United. The hardware and software of the TURTLEs are described on this site. Also the contact information of the team can be found. [5]

1.2 Project objective

The objective of this project is to improve the trajectory of the lob shot. The problem definition of this project is described below.

During gameplay it often occurs that a robot and/or keeper stand between the ball and the goal. This means a flat shot is not an option, however, a lob shot can be the solution. However, a lob shot should land exactly between the keeper and the goal which is highly depending on the shooting effort, ball pressure, voltage of the capacitor, etc. Currently, the shooting effort is obtained utilizing a linear fit without incorporating the aforementioned variables. To determine a more robust shooting effort, a model is required of the shooting system of the robot, so a better ball trajectory prediction can be provided. [6]

The assignment is to develop software that predicts the trajectory of the ball during lob shots. The assignment has the following subtasks that are already set in the project description [6]:

- Identification of the shooting system.
- Modeling of the shooting system and experimental verification.
- Integration of the model in the software.

Other subtasks that can made by the assignment are:

- The relation between the voltage of the capacitor and the current through the solenoid.
- The relation between the voltage of the capacitor and the time.

- The relation between the voltage of the capacitor and distance of the ball.
- The function for the duty cycle dependent on the distance and the capacitor voltage.

1.3 Report structure

This report has the following structure. First the background for this project is given in the second chapter. The shooting system is described and the reason for lob shots is explained.

In the third chapter the examined relations are introduced and explained. The relation between the current and the voltage of the capacitor are examined. After that the relation between the duty cycle and the distance is described. Lastly, the behavior of the capacitor is examined. Hereby the relation of the capacitor voltage against the time and against the distance of the ball are introduced and explained.

Next the software is explained. The fourth chapter contains an explanation of the function that calculates the capacitor voltage ratio and the code that is used for the experiment to determine the relation between the capacitor voltage and the distance.

The fifth chapter contains the experiments to determine the relation between the capacitor voltage versus the time and between the capacitor voltage versus the distance.

In the last chapter the results of the experiments are shown. First, the relation between the capacitor voltage and the time are determined. After that, the main relation between the capacitor voltage and the distance is determined. Next, the voltage drops during a shot are compared. At last, the ball trajectories are compared to analyze the difference in the shots.

At last, this will all be concluded in Chapter 7 and recommendations will be made in Chapter 8.

2 Background

This chapter contains the background for this project. First the shooting system is described. This consists of the shooting mechanism, the hardware, and the driving circuit, the electrical system of the TURTLE. After that the reason for lob shots is explained.

2.1 Shooting system

The shooting mechanism of the TURTLE is the basis for a soccer robot. With the mechanism the mobile soccer robot can shoot the ball with a variable speed and over a variable distance. The shooting mechanism was already designed and realized [7]. In this research the actuator is designed and validated for a soccer robot.

2.1.1 Shooting mechanism

The actuator that is selected for the shooting mechanism is a plunger type reluctance actuator. This type of actuator is chosen because the application requires force in only one direction. A capacitor is used to buffer energy from the battery of the robot. Through an IGBT, the energy is transferred from the capacitor to the actuator. The force applied by the actuator can be adjusted to applying pulse width modulation(PWM) to enable a variable shooting power. The mover of the actuator is connected to a lever which is attached to the ball. By moving this lever up and downwards, the contact point with the ball can be varied. This provides the ability that the robot can shoot with a different angle with respect to the ground. There is choice between a flat shot on the ground with a angle of 0° or a lob shot with a maximum angle of 38°. In figure 2.1 the actuator including the shooting mechanism are shown.

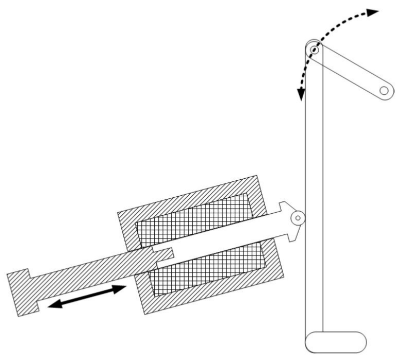

Figure 2.1: *The actuator including the shooting mechanism of the soccer robot. [7]*

2.1.2 Driving circuit

As mentioned before, the capacitor is used to buffer energy from the batteries. The robot is equipped with two batteries of $24V$. When the robot will shoot, the solenoid requires a high peak power. A $4.7mF$ capacitor is selected for this reason. The capacitor which is rated for a voltage up to $450V$ is charged from the batteries using a DC-DC converter up to $435V$. The force produced by the solenoid can be adjusted by using a PWM signal to switch the IGBT. With the IGBT the current through the coil can be adjusted. The capacitor is discharged over the solenoid using the circuit shown in figure 2.2. The solenoid is represent with the inductance L and resistance R. To avoid voltage spikes over the IGBT when switching off, a flyback diode is placed parallel with the solenoid.

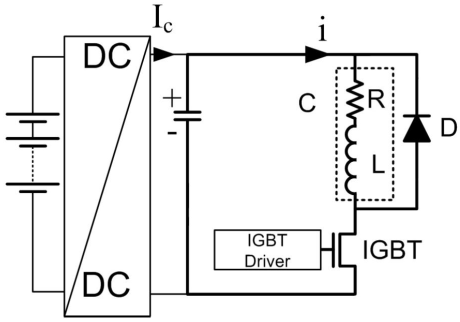

Figure 2.2: *The circuit used to drive the actuator. [7]*

The maximum allowable current through the circuit is $100A$. The coil resistance is 2.4Ω. The PWM duty cycle is a ratio between 0.0 and 1.0. A simplified model is made to determine the inductance. The response of the current is determined on the basis of the circuit of figure 2.2 and with the assumptions that the duty cycle is 1.0, the maximum capacitor voltage is $435V$ and the resistance is 2.4Ω. The model and the response of this circuit are shown in figure 2.3. The inductance that is used to get this response is $26mH$. In reality the inductance vary, but with this approximation the relation between the current and the voltage of the capacitor can be determined.

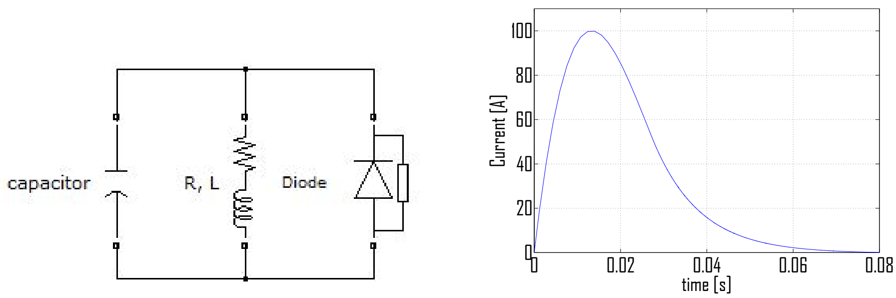

Figure 2.3: *The model and the response(time vs current) of the circuit with duty cycle* 1.0.

2.2 Lob shots

As mentioned before, a more robust shooting effort is needed to shoot more accurate. It is desired that the robot shoots just under the goal crossbar. This means that the gradient of the lob trajectory should be the largest on the goal line. For that reason a model is required of the shooting system of the robot. Before this project only the duty cycle with respect to the distance is known. This relation is determined in the validation of the shooting actuator [7] and is explained in more detail in section 3.2.

3 Analysis

In this chapter the examined relations are introduced and explained. First the relation between the current and the voltage of the capacitor is examined. After that the relation between the duty cycle and the distance is described. This relation was examined earlier, but is an important relation for this project. Lastly, the behavior of the capacitor is examined. Hereby the relation of the capacitor voltage against the time and against the distance of the ball are explained and introduced.

3.1 Current vs. voltage

To determine the relation between the voltage of the capacitor and the current through the solenoid, the model in section 2.1.2 is used. In figure 3.1 the current through the solenoid can be seen for different voltages. As can be seen the current peak decreases when the voltage decreases. In figure 3.2 is shown that it is a linear relation,

$$I = 0.23V \tag{3.1}$$

where I is the current and V is the voltage.

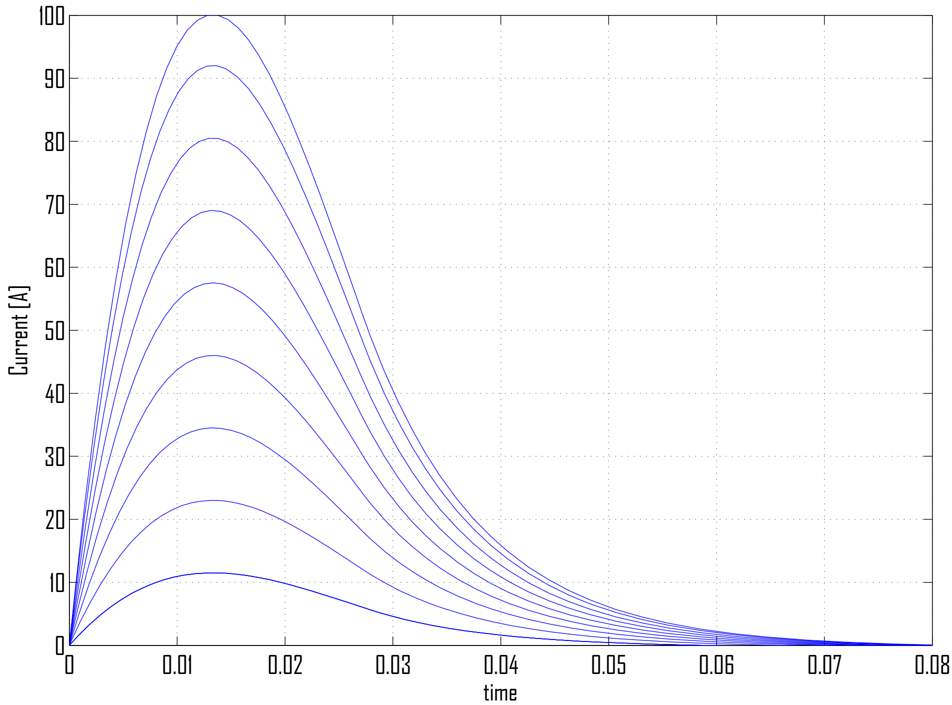

Figure 3.1: *Current through the coil for different voltages of the capacitor.*

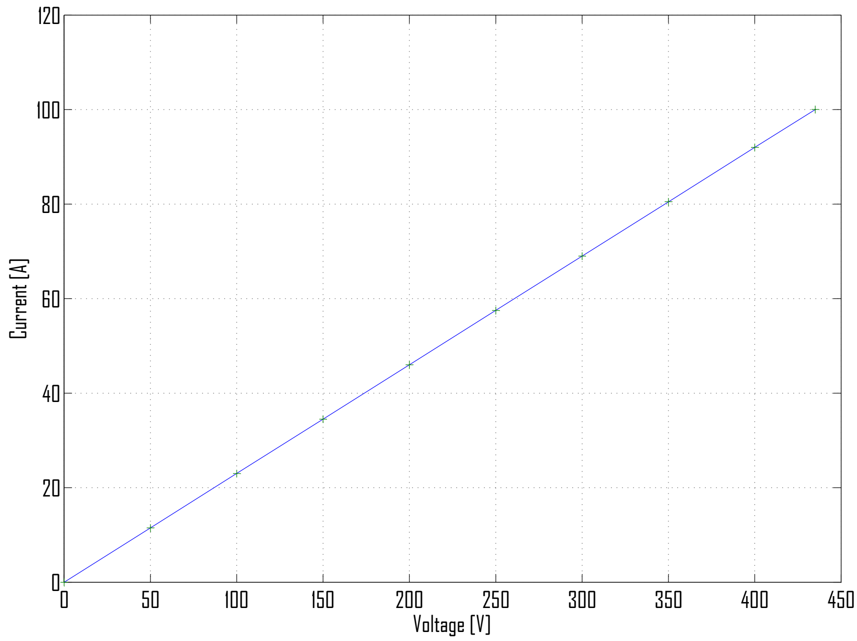

Figure 3.2: *Relation between voltage of the capacitor and the current peaks.*

3.2 Duty cycle vs. distance

As mentioned before, the force produced by the solenoid can be adjusted by using a pulse width modulation signal to switch the IGBT. The PWM duty cycle is a ratio between 0.0 and 1.0. With a duty cycle of 1.0 the robot shoots as hard as possible. If the capacitor is fully charged and the duty cycle is 1.0, the distance where the ball hits the ground during a lob shot is 12.2 meter according to earlier experiments. [7]

In table 3.1 the results of the measurement obtained for different duty cycles are listed. The distance represents the distance from the center of the TURTLE to the point where the ball hits the ground for the first time. The relation between the distance of the ball and the duty cycle is linear. However it is very dependent on the ball, because the mass and the pressure of the ball are of major influence on the movement of the ball. [7]

Table 3.1: *Shooting performance [7]*

Duty cycle	Distance (m)
0.2	2.5
0.3	3.7
0.4	5.1
0.5	6.0
0.6	7.5
0.7	8.9
0.8	10.1
0.9	11.2
1.0	12.2

3.3 Charge time capacitor

The force that the solenoid can produce, is dependent on the voltage of the capacitor. With a fully charged capacitor the force will be higher than when the capacitor is not fully charged. The time it will takes to charge the capacitor can be measured with the TURTLE. A general model for charging the capacitor can be made. The ratio of the capacitor voltage against the time can be described with,

$$RCV(t) = \alpha * exp(\beta * t) + \gamma \tag{3.2}$$

where the parameters α, β and γ of this equation will be determined in section 6.
Another important effect is the amount of discharging of the capacitor during a shot. The implementation in the software to measure the capacitor voltage will be explained in section 4.

3.4 Capacitor voltage vs. distance

The distance of the ball during a lob shot is also dependent on the capacitor voltage. When the relation between the distance and the capacitor voltage is known, a relation can be made where the distance of the ball is dependent on the duty cycle and the capacitor voltage. With the new relation the distance of the lob shot should be the same with different voltages of the capacitor. So when the capacitor is not fully charged, the duty cycle should become higher to reach the same distance.

It is assumed that the relation between the distance of the ball and the duty cycle is linear. [7] Because of this relation it is considered that the force versus the distance is also linear. With,

$$F = \frac{I^2(t)}{2} \frac{\delta L(z)}{\delta z} \tag{3.3}$$

where F is the force, I is the current, L is the inductance and z is the movement of the solenoid, the quadratic relation between the force and the current, and the relation that the current versus the voltage is linear the expected fit can be made. The expectation is that the relation for the capacitor voltage versus the distance of the lob shot is a quadratic fit. The relation between the distance and the capacitor voltage is experimentally determined.

4 Software

This chapter contains an explanation of the algorithm that calculates the capacitor voltage ratio. In the second subsection the code is explained that is used for the experiment to determine the relation between capacitor voltage versus the distance of the lob shot.

4.1 Algorithm capacitor voltage

The shooting system is controlled with the code of *kick.c*. [7] One of the functions in *kick.c* calculates the capacitor voltage. This function, named *GetCapacitorState*, returns the current voltage as ratio of the maximum value between 0.0 and 1.0.

The function starts with a filter. In this filter the sensor values are stored. The function also continuously searching for a maximum capacitor voltage. There are three checks before the function returns the ratio. It checks if it is measured in the real maximum range, if it is measured above a certain threshold and if it is measured long enough. If one of the checks is not true, the function returns the value -1. In figure 4.1 a schematic overview of this function is shown. The c-code of the function *GetCapacitorState* can be found in Appendix B.

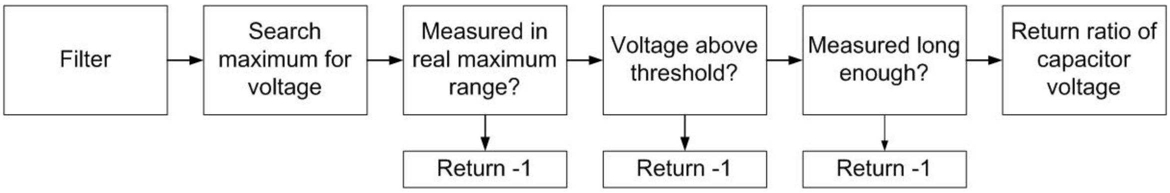

Figure 4.1: *Schematic representation of the function that calculates the capacitor voltage in kick.c.*

It is important to plot the ratio returned by the function versus the time. In figure 4.2 two plots can be seen. The dash line represents the ratio versus the time and the full line represents the capacitor voltage. The sensor values of the capacitor voltage is up to about a ratio of 3.6. This could be different for other actuators. Therefore the ratio is calculated to make the capacitor state for each TURTLE the same. As can be seen figure 4.2 the ratio is initially -1 as expected. When the capacitor voltage is above the threshold 3.5 and measured long enough over five seconds, than the ratio will be calculated.

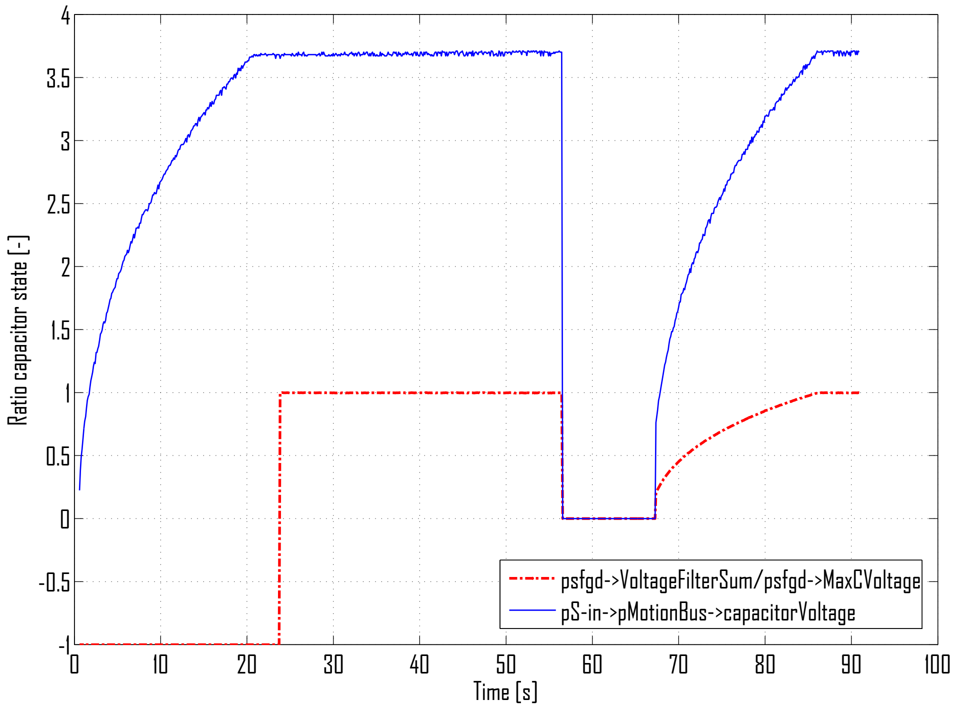

Figure 4.2: *Ratio returned by the function and capacitor voltage versus the time.*

4.2 Algorithm for experiment voltage vs. distance

The experiment capacitor voltage versus the distance of the lob shot can only be done when the voltage of the capacitor is known. The ratio returned by the function *GetCapacitorState* can be used to shoot the ball when the capacitor has a certain voltage ratio.

The code *kick.c* that is used for this experiment is explained in Appendix A. As can be seen in this appendix a lot of cases are used to control the shooting system. For this experiment an extra requirement is added to shoot with different voltages of the capacitor in the case *wait for shot*. The requirement ensures that the TURTLE can only shoot when the ratio of the capacitor voltage is above a certain value. The extra requirement is added to the if-statement. A part of the case *wait for shot* can be seen in the following code. The new requirement is displayed in bold.

```
        case WAIT_FOR_SHOT: /* Wait for aim */
            WaitSamples = (int)(MOTION_SAMPLE_RATE*S_in.pMotionTuningbus->...
                K_wait_time_during_aim_in_s);


            if ((IsAimed(S, &S_in) || (S_in.pTRCbus->joystickOnOff == 1)) && ...
                GetCapacitorState(S, &S_in) > (S_in.pMotionTuningbus->...
                Capacitor_voltage_threshold)){  /* aimed! */
                psfgd->CounterWaitForAim    = 0;
                psfgd->State                = SHOOT;
                ...
                ...
```

As can be seen in the code the value of this threshold added to the *pMotionTuningbus* in the bus manager. The tunable parameter *Capacitor_voltage_threshold* has to be added to the *pMotionTuningbus* before this can be used.

Another parameter that should be changed in the bus manager is the tunable parameter *K_wait_time_during_aim_in_s*. This parameter is initially set to 3.0. This ensures that the TURTLE does not aim too long. But for the experiment capacitor voltage versus distance of the lob shot, the time-out for aiming should be long enough until the TURTLE shoots the ball.

The data of the capacitor voltage and the time are written to a text-file. In appendix C the c-code is explained. Also the Matlab file used to plot the data can be found in this appendix.

5 Experiments

This chapter contains the experiments to determine the behavior of the capacitor and to determine the relation between the capacitor voltage and the distance of the lob shot. The way of experimenting is in the following subsections explained.

5.1 Capacitor state

The first experiment to determine the behavior of the capacitor is the capacitor voltage versus the time. With this experiment the parameters of equation 3.2 will be determined. The experiment is done on the following way:

- Start the TURTLE and software as described on the WIKI. [5]
- Wait 30 seconds until the capacitor is fully charged and the ratio can be calculated.
- Push on the emergency button and wait 30 seconds. The capacitor is now fully discharged.
- Push on the reset button on the TURTLE.
- Analyze the results as described in appendix C.

With the second experiment the voltage drops of the capacitor will be investigated. The five points that are described above are the same for this experiment. The only difference is that the TURTLE shoot a ball with a fixed duty cycle and capacitor voltage, after the TURTLE is reset.

5.2 Capacitor voltage vs. distance

The goal of this experiment is to determine the relation between the capacitor voltage and the distance of the ball. The experiment is done on the following way:

- Start the TURTLE and software as described on the WIKI. [5]
- Ensure that the ball is 0.6 bar.
- Set the capacitor voltage threshold on the desired value in the bus manager.
- Wait 30 seconds until the capacitor is fully charged and the ratio can be calculated.
- Push on the emergency button and wait 30 seconds. The capacitor is now fully discharged.
- Push on the reset button on the TURTLE.
- Pull the ball into the ball handling.
- Give the lob shot command d in manual mode.
- Wait until the capacitor reached the desired voltage and the TURTLE shot.
- Measure the distance.
- After the measurement stop the manual mode, set the lever into the capacitor and start the manual mode again. The lever will be initialized again.

This experiment will be done for different capacitor voltage thresholds. The used voltage thresholds are the ratios 1.0, 0.9, 0.8, 0.7, 0.6 and 0.5.

6 Results

In this chapter the results of the experiments are shown. First, the relation between the capacitor voltage and the time are determined. After that, the main relation between the capacitor voltage and the distance is determined. Next, the voltage drops during a shot are compared. At last, the ball trajectories are compared to analyze the difference in the shots.

6.1 Charge time capacitor

To analyze the results of the first measurement, a Matlab script is used to plot the right curves. In figure 6.1 the capacitor voltage ratio is plotted against the time. The best fitted curve through the points is described with,

$$RCV(t) = \alpha * exp(\beta t) + \gamma \tag{6.1}$$

where $\alpha = -0.8975$, $\beta = -0.08569$ and $\gamma = 1.074$.

As can be seen in figure 6.1 it takes 26 seconds to fully charge the capacitor.

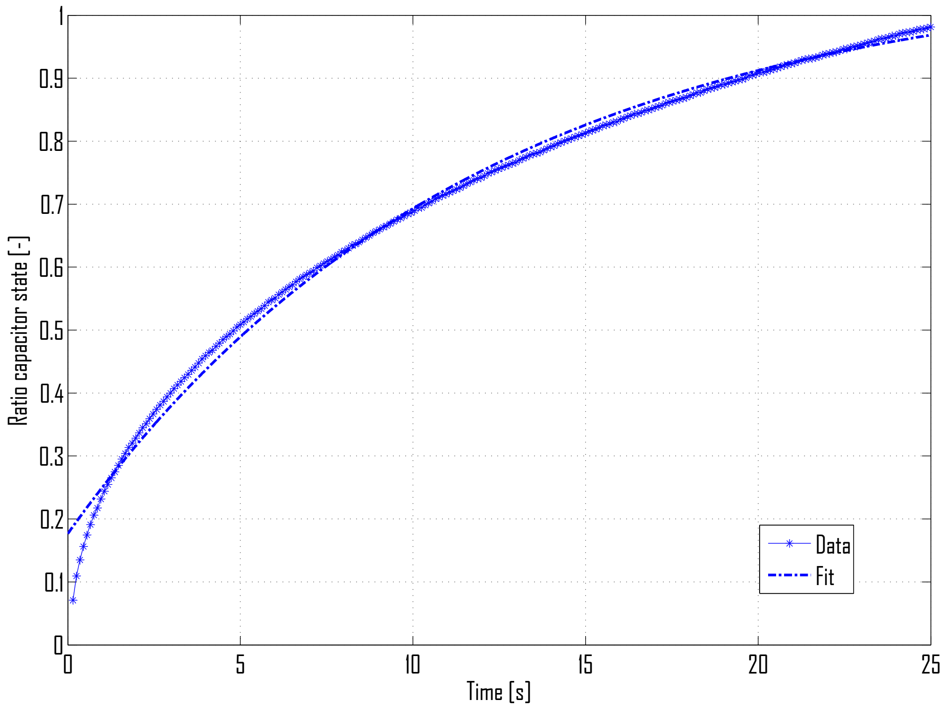

Figure 6.1: *Charging the capacitor.*

6.2 Relation capacitor voltage vs. distance

Each shot is filmed by a high resolution camera that took 300 frames per second. These frames are loaded in a Matlab script and every five frames the center of the ball is clicked. The Matlab script can be found in appendix E.1. The data with all the centers of the ball will give the trajectories of the ball for each shot. In figure 6.2 the trajectories of each shot can be seen.

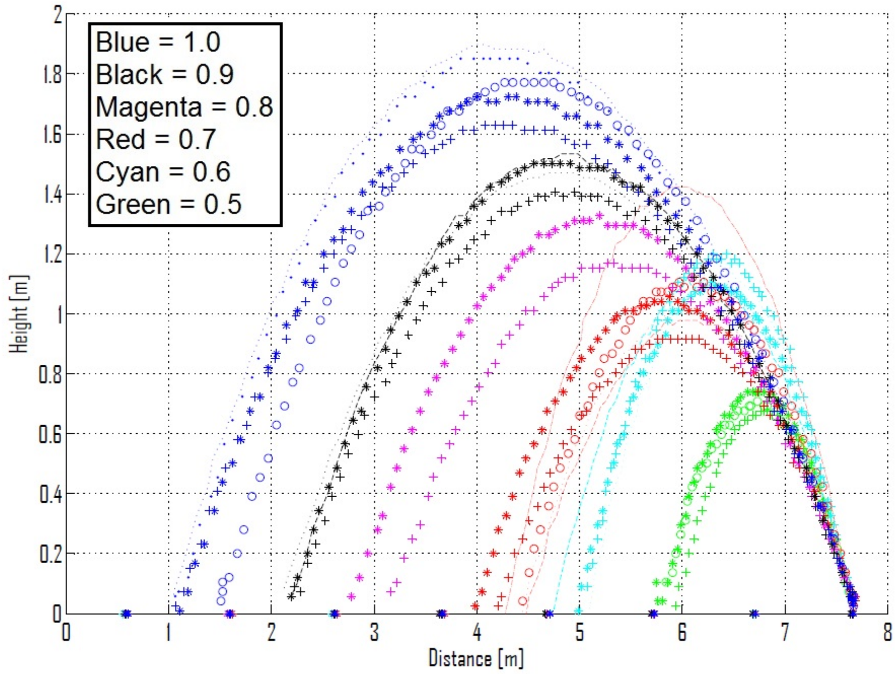

Figure 6.2: *Trajectories of the ball with duty cycle* 40% *and for different capacitor voltage ratios with TURTLE 6.*

Eight markers are shown in the plot that represent the meters. With this markers the amount of pixels for one meter can be calculated. One meter is 63 pixels in the plot, so the distances of the lob shots are known. The distances of the lob shots are calculated and can be found in Appendix E.2. The distance of the lob shots can be plotted against the capacitor voltages. The data of table E.1 is plotted in figure 6.3.

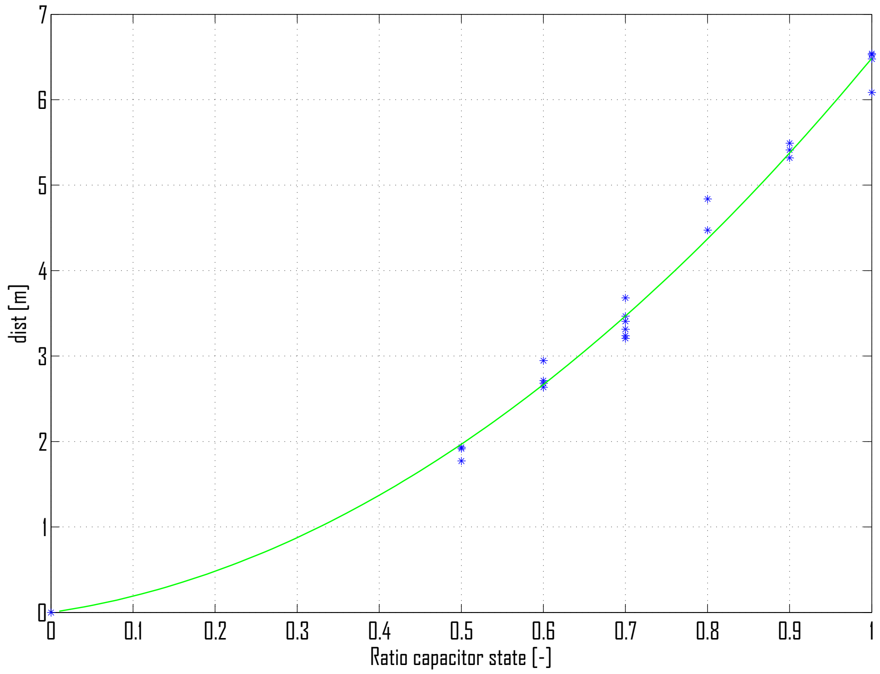

Figure 6.3: *Data and fit of the relation between the capacitor voltage ratio and the distance of the lob shots.*

In section 3.4 is determined that the relation between the capacitor voltage and distance should be quadratic. With this relation a fit can be made through the points of the data. The relation that represent the fit is described with,

$$DC = aV^2 + bV \tag{6.2}$$

where $a = 5.093$ and $b = 1.390$

Finally, the relation between distance, capacitor voltage and duty cycle can also be made. First the relation for the distance dependent on the linear duty cycle and and the capacitor voltage is determined with,

$$D = \frac{aV^2 + bV}{d} * c * DC \tag{6.3}$$

Where $a = 5.093$, $b = 1.390$, $c = 16.209$ and $d = 6.483$.

This relation can be also be determined for the duty cycle dependent on the distance and the capacitor voltage with,

$$DC = \frac{dD}{c(aV^2 + bV)} \tag{6.4}$$

6.3 Voltage drops

To determine the variation in the distances of the shots, the voltage drops during a lob shot are investigated. The first six shots are executed without discharging the capacitor between two shots. After that four shots with discharging. At last, four shots again without discharging. The measured data is plotted against the time and can be seen in figure 6.4.

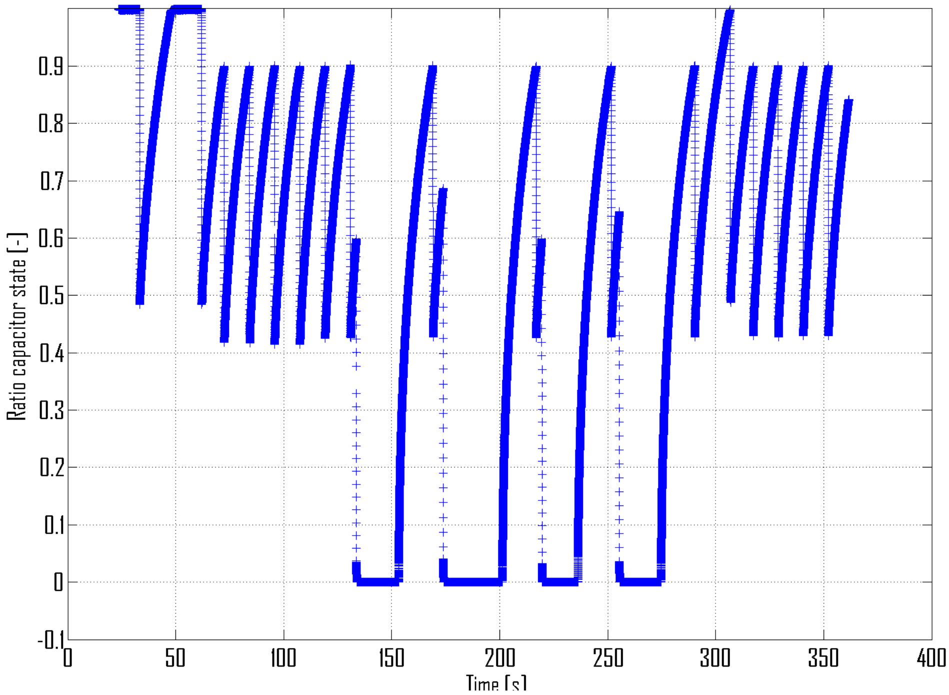

Figure 6.4: *Voltage drops of the capacitor with a* 50% *duty cycle and a capacitor voltage threshold of* 0.9*.*

Each voltage drop is analyzed to calculate the difference of the highest and lowest value of the capacitor voltage ratio. In appendix D the table with this data can be found. The average of the voltage drops is 0.4756. The minimum and maximum difference is respectively 0.4701 and 0.4854. This is a maximum error of 3%. This is calculated with,

$$\epsilon = \frac{max - min}{min} 100\% \tag{6.5}$$

where ϵ = the maximum error, max = 0.4854 and min = 0.4701.

6.4 Ball trajectories

In figure 6.2 can be seen that the trajectories of the shots, with the same capacitor voltage ratio, are not comparable. For example, the trajectories of the shots with a capacitor voltage ratio of 1.0 are analyzed, see figure 6.5.

The trajectories vary in height and in distance. The maximum error in height is $0.26m$. So the maximum error is 16%, calculated with

$$\epsilon = \frac{max - min}{min} 100\% \tag{6.6}$$

where ϵ = the maximum error, max = 1.888 and min = 1.628.

The angles of the trajectories with respect to the ground vary and also initial velocity of the ball vary. In section 2.1.1 was given that the maximum angle is 38°, but after analyzing the trajectories this angle is 45°.

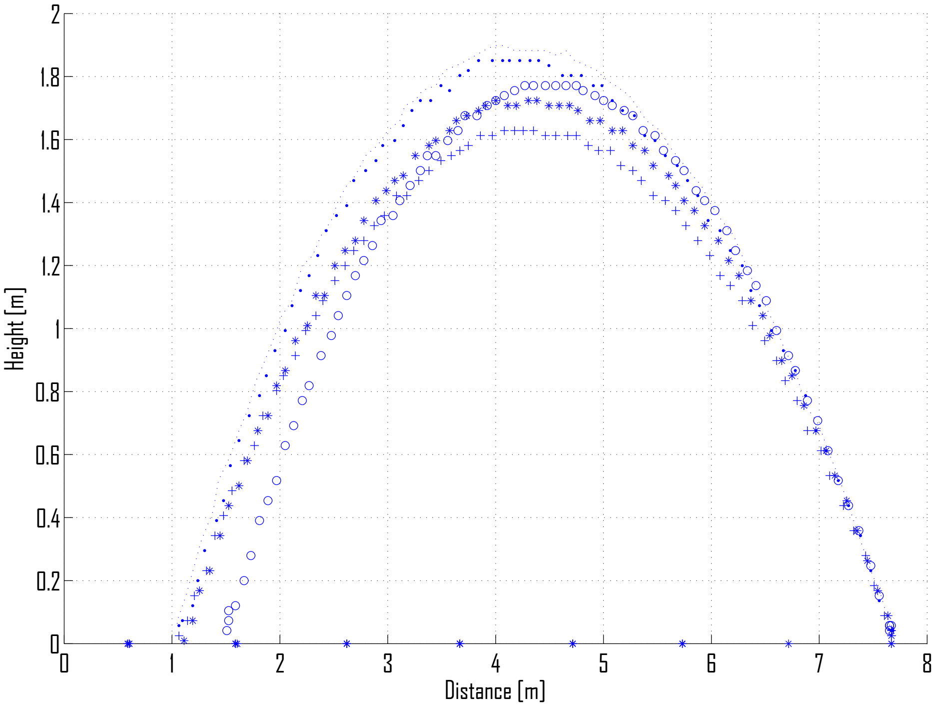

Figure 6.5: *Trajectories of the ball with a capacitor voltage ratio of* 1.0.

7 Conclusion

The project was set up for developing software that predicts the trajectory of the ball during a lob shot. The trajectory of a lob shot is highly depending on the shooting effort, ball pressure, voltage of the capacitor, etc. To determine a more robust shooting effort, a model is required of the shooting system of the robot, so a better ball trajectory prediction can be provided.

First, the background for this project has been researched. The shooting system has been described. This consists of the shooting mechanism and the driving circuit. To determine the unknown inductance in the model of the electrical circuit, a simplified model has been made. The inductance, found in the model, is $26mH$.

The software that controls the shooting system and the function that calculates the capacitor voltage ratio has been explained. Afterwards the changes made in the S-function *kick.c* has been described. The changes in the code were necessary to execute the experiments.

With the simplified model a linear relation between the current through the coil and the voltage of the capacitor has been found. Besides the linear relation between the duty cycle and the distance has been described. Because of this relation the assumption has been made that the force versus the distance is also linear. A third relation has been found that the force versus the current is quadratic. With all these relation the conclusion has been made that the relation between the capacitor voltage and the distance should be quadratic. Besides this relation, the behavior of the capacitor has been examined.

The goals of the experiments are, examine the behavior of the capacitor and find the relation between the capacitor voltage and the distance. The result of the first relation between the capacitor voltage ratio and the time is that it takes 26 seconds to fully charge the capacitor. Second, the relation between the capacitor voltage and the distance has been found. Also the relation has been found for the duty cycle dependent on the distance of the lob shot and the capacitor voltage. The constants a, b, c and d are dependent on the calibration of the shooting system.

During the experiment the lob shot trajectories has been record with a camera and analyzed with a Matlab script to determine the distances of each lob shot. After analyzing the trajectories of the lob shots, the conclusion has been made that distances of the lob shots are different. Therefore the voltage drops during a lob shot are investigated. The variation in the voltage drops, for a shot with the same duty cycle and capacitor voltage, is 3%. The trajectories vary also in height with a maximum error of $0.26m$, so a percentage of 16%. The last conclusion that can be made about the trajectories is that the angle of the ball is not 38°, but ranges around 45°.

8 Recommendation

In this chapter the next steps for a more accurate lob shot are described for the future.

First repeat the experiments for other duty cycles and with other TURTLEs to determine a more accurate relation between the capacitor voltage and the distance. In this project the experiment was done with only TURTLE 6 and with a duty cycle of 40%.

With more duty cycles, the relation between the duty cycle and the distance with a fully charged capacitor can be determined. When this relation is exactly known this relation can be implemented for the relation of the duty cycle dependent on the distance and the capacitor voltage.

The experiments are time consuming. It will be better to make the experiments more automatically. That means the lever will automatically be initialized before the TURTLE shoots the ball. Also the capacitor will automatically be fully discharged before the capacitor charges to a certain voltage.

At last, other factors should be investigated, such as the ball handling, the lever angle during a shot and the angle of the ball with respect to the ground. At this moment the trajectories of the lob shots are not comparable. So with more factors, the prediction model for lob shot trajectories should be more accurate.

In the future the new model should be implemented in the software. A low level control algorithm need to make sure that the duty cycle increases when the capacitor is not fully charged.

References

[1] Robocup. http://www.robocup.org/.

[2] Beginning of tech united eindhoven. http://www.techunited.nl/nl/ontstaan.

[3] Tech united eindhoven website. http://www.techunited.nl/.

[4] Turtle of tech united eindhoven. http://www.techunited.nl/nl/turtle.

[5] Wiki of tech united eindhoven. http://www.techunited.nl/wiki/index.php?title=Main_Page.

[6] dr. ir. René v.d. Molengraft B. van Ninhuijs MSc. Project description: Prediction method for lob shot trajectories, 2014.

[7] E.A. Lomonova K.J. Meessen, J.J.H. Paulides. A football kicking high speed actuator for a mobile robotic application. Proceedings of the 36th Annual Conference of the IEEE Industrial Electronics Society, IECON 2010, 7-10 November 2010, Glendale, Arizona. (pp. 1659-1664).

[8] M. van 't Klooster. Human - robot interaction. 2014. p. 39 - 44.

[9] Struct (c programming language). http://en.wikipedia.org/wiki/Struct_%28C_programming_language%29#typedef.

A Appendix kick.c

The general lay out of the S-function *kick.c* is described in this appendix. The general lay out of a S-function is earlier described. [8] This appendix should give a clear structure of *kick.c*.

The script of a S-function can be divided in three subsections:

- Defines and includes
- Callback method implementations
- Simulink product interfaces

A.1 Defines and includes

The defines and includes part can be divided in several parts:

- **Describe S-function**. This part includes the name of the S-function and the version of Simulink.
- **Libraries**. Date of other S-function can be used in another S-function, but only when the file is in the top of the function file. The libraries are necessary to write in C-language.
- **Define ports**. Data can come in and leave a S-function, but it can only when ports are described.
- **Define states and parameters**. The states for the callback method are defined with integers. Also parameters are defined into this part. The parameter definitions part should be placed between the Input and Output definitions and the Input and Output properties.
- **Define structs**. A struct is used to define a physically grouped list of variables to be placed under one name in a block of memory [9]. There are defined three types of structs, the input, output and global data struct. The input struct contains the variables that are defined at the input ports. This is also applies to the output struct. The global data struct contains the variables that are only used in the script *kick.c*.

A.2 Callback method implementations

The callback method implementations part can also be divided in several parts:

- **mdlInitialize Sizes**. In this section first the S-function parameters field is cleared, so no parameter mismatches will occur. After that, every port which is defined previously has several properties which are described below.
- **mdlInitialize Sample Times**. In this part the definition of the sample time is described on which the S-function operates.
- **mdlInitialize Conditions**. In this section the definition of the initialize conditions is described on which the S-function operates. In the last part of this section *my_id* gets the right agent variable if *MATLAB_MEX_FILE* is not defined.

After the standard functions the following functions are defined to describe standard actions. In the mdlOutputs part these functions can be used.

- **Initialize Lever**.
- **Set Angle**.
- **Determine Lever Angle**.
- **Is Aiming**.

- **Get Capacitor State**.
- **mdlOutputs**. The section mdlOutputs compute the S-functions outputs at the current time step and store the results in the S-functions output signal arrays. It starts with getting pointers to global data. After that, the function describing the output variables and receiving the input variables. And it initially set all outputs to zero. Some parameters are also initialized and set. At last, the main part of *kick.c* is described that is used to control the shooting system. That is done with cases. With the switch-statement the different cases are separated. The following cases are defined:
  - **Reset**
  - **Initialize**
  - **Homing**
  - **Set lever to preferred angle**
  - **Wait for aiming**
  - **Prepare shot**
  - **Wait for shot**
  - **Shoot**
  - **Check shot**
  - **Push ball**
  - **Wait after shot**
  - **Default**

  The default case can be used for performing a task when none of the cases is true. More explanation about the actions in these cases is given in *kick.c* itself.
- **mdlTerminate**. Perform any actions required at termination of the simulation.

A.3 Simulink product interfaces

This part should always be included in the script. This ensures that the Simulink model can be compiled. It checks if the file is being compiled as a MEX-file. It used the MEX-file interface mechanism *simulink.c* and the code generation registration function *cg_sfun.h*.

B Appendix GetCapacitorState

In this appendix the c-code for the function *GetCapacitorState* is shown. [7]
This function is used to calculate the current ratio as ratio of the maximum value. The ratio is used to determine the relation between the capacitor voltage and the distance of a lob shot.

```c
static real_T GetCapacitorState(SimStruct *S, InputStruct_t* pS_in)
{
    psfun_global_data psfgd = (psfun_global_data) ssGetRWork(S);

    /* first do the filter thing */
    psfgd->ibufVoltageFilter++;
    /* retract oldest value from sum */
    psfgd->VoltageFilterSum -= psfgd->VoltageFilter[psfgd->ibufVoltageFilter % ...
        MA_FILT_SIZE];
    /* add new value to sum */
    psfgd->VoltageFilterSum += pS_in->pMotionBus->capacitorVoltage;
    /* add new value to filter */
    psfgd->VoltageFilter[psfgd->ibufVoltageFilter % MA_FILT_SIZE] = pS_in->...
        pMotionBus->capacitorVoltage;
    /* continue to search for maximum capacitor voltage */
    psfgd->MaxCVoltage = dmax(psfgd->MaxCVoltage,psfgd->VoltageFilterSum);
    /* check if we already measured in the real maximum range */
    if(!psfgd->MaxVoltageFoundFlag){
        /* check we have a voltage above the threshold */
        if(pS_in->pMotionBus->capacitorVoltage > THRESHOLD_C_VOLTAGE){
            /* check if we measured long enough */
            if((ssGetT(S) - psfgd->StartTimeMaxVoltageMeasurement) > MAX_VAL_TIME){
                psfgd->MaxVoltageFoundFlag = 1;
            }
        }else{
            psfgd->StartTimeMaxVoltageMeasurement = ssGetT(S);
        }
        return -1;
    }
    /* return the current voltage as ratio of the maximum value between 0 and 1! */
    return (psfgd->VoltageFilterSum/psfgd->MaxCVoltage);

}
```

C Appendix plot data

This appendix contains the procedure to write and analyze data from the TURTLE. First the c-code is shown that is added to the S-function *kick.c*. After that the commands in the terminal are given. At last, the Matlab script that is used to read and plot the data.

```
    /* Write every 0.1 second the capacitor state, time and resettime to ...
        capacitor_state.txt*/
    if(psfgd->CounterCapacitor == freq/savefreq){

        FILE *fptr = NULL;
        /* Open the log file, "a" means that previous text in file is not ...
            overwritten */
        fptr = fopen("capacitor_state.txt", "a");
        /* Check if the file was opened */
        if (fptr != NULL)
        {
         /* Log time and capacitor state to text file */
         fprintf(fptr, "%f, %f\n", ssGetT(S), GetCapacitorState(S, &S_in));
         fclose(fptr);
        }

        psfgd->CounterCapacitor = 0;
    }
    else{
        psfgd->CounterCapacitor++;
    }
```

With the following commands in the terminal the file can be copied from a TURTLE to a devpc.

```
>> sudo su
>> ssh turtle[number of TURTLE]
>> sudo su
>> cd /home/robocup/
>> ls
>> scp [name of file] robocup@devpc[number of devpc]:[destination location of file]
```

For example, [number of TURTLE]=6, [name of file]=capacitor_state, [number of devpc]=5 and [destination location of file]=/home/robocup/svn/trunk/dev/Nick/Exp1/capacitor.txt.

With the following Matlab script the data of capacitor.txt can be read and plotted.

```
% Capacitor state versus time

%Init
clear all; close all; clc;

% Load file
fid = fopen('capacitor.txt');      %Open the txt file

%Scan text
C = textscan(fid, '%f%f', 'delimiter', ',');
fclose(fid);

%Make arrays
t = C{1};                          %time
cap_state = C{2};                  %capacitor state

%Plot
figure()
plot(time, cap_state, '*')
title('Capacitor state');
xlabel('Time [s]');
ylabel('Ratio capacitor state [-]');
```

D Appendix voltage drops

To determine the variation in the distances of lob shots, the voltage drops during a shot are investigated. Each voltage drop is analyzed to calculate the difference of the highest and lowest value of the capacitor voltage ratio. In table D.1 the values are shown.

Table D.1: *Voltage drops*

Highest value	Lowest value	Difference
0.9003	0.418	0.4823
0.9	0.4169	0.4831
0.9005	0.4151	0.4854
0.9003	0.4146	0.4857
0.9002	0.4252	0.475
0.9017	0.4264	0.4753
0.9012	0.427	0.4742
0.9	0.4264	0.4736
0.9003	0.4277	0.4726
0.9007	0.428	0.4727
0.9	0.4299	0.4701
0.9005	0.4275	0.473
0.9002	0.4294	0.4708
0.9007	0.4294	0.4713

E Appendix voltage vs. distance

E.1 Trajectory tool

The following Matlab scripts are made to load the movie, clicked on the centers of the balls and make the data sets. The function *drawcircle.m* ensures for a circle around the arrow, so the center of the ball can be found easier.

```
% clickballs.m

% Load movie
mov = mmreader('0.5/shot3.avi');

nFrames = mov.NumberOfFrames;
height = mov.Height;
width = mov.Width;

ballPositions = NaN(nFrames,2);

i = 1;

fg = figure(1);
h = imshow(zeros(height,width,3));
hold on;
c = plot(0,0, 'r');
hold off;
global m x y;
m = 0;
set(fg,'WindowButtonMotionFcn',{@drawCircle,c,7,height});
set(fg,'WindowButtonDownFcn',{@clicked, height});

while i < nFrames
    cdata = read(mov, i);
    set(h, 'CData', cdata);
    title(['Frame #' num2str(i)]);
    while(m == 0)
        pause(0.05);
    end
    if(m == 1)
        ballPositions(i,1) = x;
        ballPositions(i,2) = height-y;
        i = i+5;
    end
    if(m == 2)
        i = i+10;
    end
    if(m == 3)
        i = i-10;
    end
    if(m == 4)
        i = nFrames;
    end
    m = 0;
end

close all;

% Make data sets
balls05_3(:,1) = ballPositions(find(¬isnan(ballPositions(:,1))),1);
balls05_3(:,2) = ballPositions(find(¬isnan(ballPositions(:,2))),2);
```

```matlab
% drawcircle.m

function drawCircle(handles, ¬, h, r, height)
    handle = get(handles);
    child = get(handle.Children);
    childPos = child.Position;
    xy = handle.CurrentPoint;
    position = handle.Position;
    size = position(3:4);
    marginX = childPos(1)*size(1);
    marginY = childPos(2)*size(2);

    x = xy(1)-marginX;
    y = height - (xy(2)-marginY);
    pts = linspace(0,2*pi,20);
    set(h, 'XData', x+r*cos(pts));
    set(h, 'YData', y+r*sin(pts));
end
```

```matlab
% clicked.m

function clicked(handles, ¬, height)
    handle = get(handles);
    child = get(handle.Children);
    childPos = child.Position;
    xy = handle.CurrentPoint;
    position = handle.Position;
    size = position(3:4);
    marginX = childPos(1)*size(1);
    marginY = childPos(2)*size(2);

    global x y;
    x = xy(1)-marginX;
    y = height - (xy(2)-marginY);

    button = get(gcf,'SelectionType');

    global m;
    if(strcmp(button, 'normal'))
        m = 1;
    end
    if(strcmp(button, 'extend'))
        m = 2;
    end
    if(strcmp(button, 'alt'))
        m = 3;
    end
    if(x < 0)
        m = 4;
    end

end
```

E.2 Data voltage vs distance

The distances of the lob shots at different capacitor voltage ratios are calculated and are shown in the table below.

Table E.1: *Capacitor performance*

Capacitor voltage ratio [-]	Distance [m]
0.5	1.93
	1.91
	1.77
0.6	2.71
	2.95
	2.64
	2.68
0.7	3.68
	3.47
	3.41
	3.31
	3.24
	3.21
0.8	4.84
	4.47
0.9	5.49
	5.41
	5.41
	5.32
1.0	6.54
	6.48
	6.53
	6.08