EINDHOVEN UNIVERSITY OF TECHNOLOGY

Model evaluation of the humanoid TUlip

Bachelor Thesis

A.T. van Rijn

7/1/2013

Coaches: P. W. M. van Zutven, H. Nijmeijer

TABLE OF CONTENTS

1. INTRODUCTION

The humanoid TUlip has been modeled in several ways over the course of the years. Last in this series is the model in the simulator Gazebo, which originated as a part of the Robot Operating System (ROS) [1]. T. Assman has used this model in his master thesis to simulate the sideward fall of TUlip [2]. As a part of this, he optimized the model to fit the experimental data. The goal of this bachelor thesis is to investigate the accuracy of this model in other movements such as walking forward, walking sideways and point turning.

Having a functional model of TUlip would improve the development of new walking strategies. Since the model does not break when it falls over, parameters in new walking strategies can be acquired with less work and less risk to the robot. In the future, TUlip might even teach himself how to walk in the simulator using a self-learning algorithm [3].

In order to compare the behavior of TUlip in the simulator with his behavior in the real world, the data that the robot gathers during an experiment are also gathered in the simulation. These data consist of the angle of TUlip's joints, the forces under the different corners of his feet and the angle of the torso and are further explained in Chapter 2. To determine the conformity of the simulation with the reality, these data are compared. The two are said to match well when the maximum error does not exceed 20% of the stroke and the root-mean-square (RMS) value does not exceed 10% of the stroke. When these values stay below the 10% and 5% respectively, the simulation is said to be accurate.

In the first Chapter, the robot TUlip will be explained, so its possibilities and limitations become clear. After that, the simulator Gazebo and the Matlab scripts that are used are explained. In Chapter 4, the data will be compared and the differences analyzed. In Chapter 5 the conclusions from the report are gathered and in Chapter 6, the points of discussion are elaborated on.

2. THE ROBOT TULIP

In this chapter, the robot TUlip and its workings are explained.

The humanoid soccer robot TUlip was developed by Dutch Robotics for research in the recreation of human walking. It is also allowed to play in the humanoid adult size league of the RoboCup [4]. TUlip is fitted with incremental encoders on each of his joints, the leg joints are shown in Figure 2.1b. The small orange box that is visible in Figure 2.1a is an Inertial Measurement Unit (IMU) by Xsens. On each corner of both feet, TUlip has a Tekscan FlexiForce force sensor, for a total of 8 force sensors. For vision, TUlip uses two cameras, mounted to the torso with two Dynamixel servo motors, allowing for rotations in yaw and pitch directions.

The cameras are only used to orientate and find the ball and are not relevant for this thesis. The force sensors were meant to tell whether a corner of the foot was in the air or on the ground and saturate at 120 N [2,5]. This means an added inaccuracy, making the force sensors too unreliable for accurate measurements. The IMU uses an accelerometer, a gyroscope and a compass to determine its orientation.

TUlip has 16 joints, allowing for movement. The two joints in the neck are already described and there is a joint in each shoulder that isn't used for walking. Then there are six joints in each leg, as shown in Figure 2.1b.

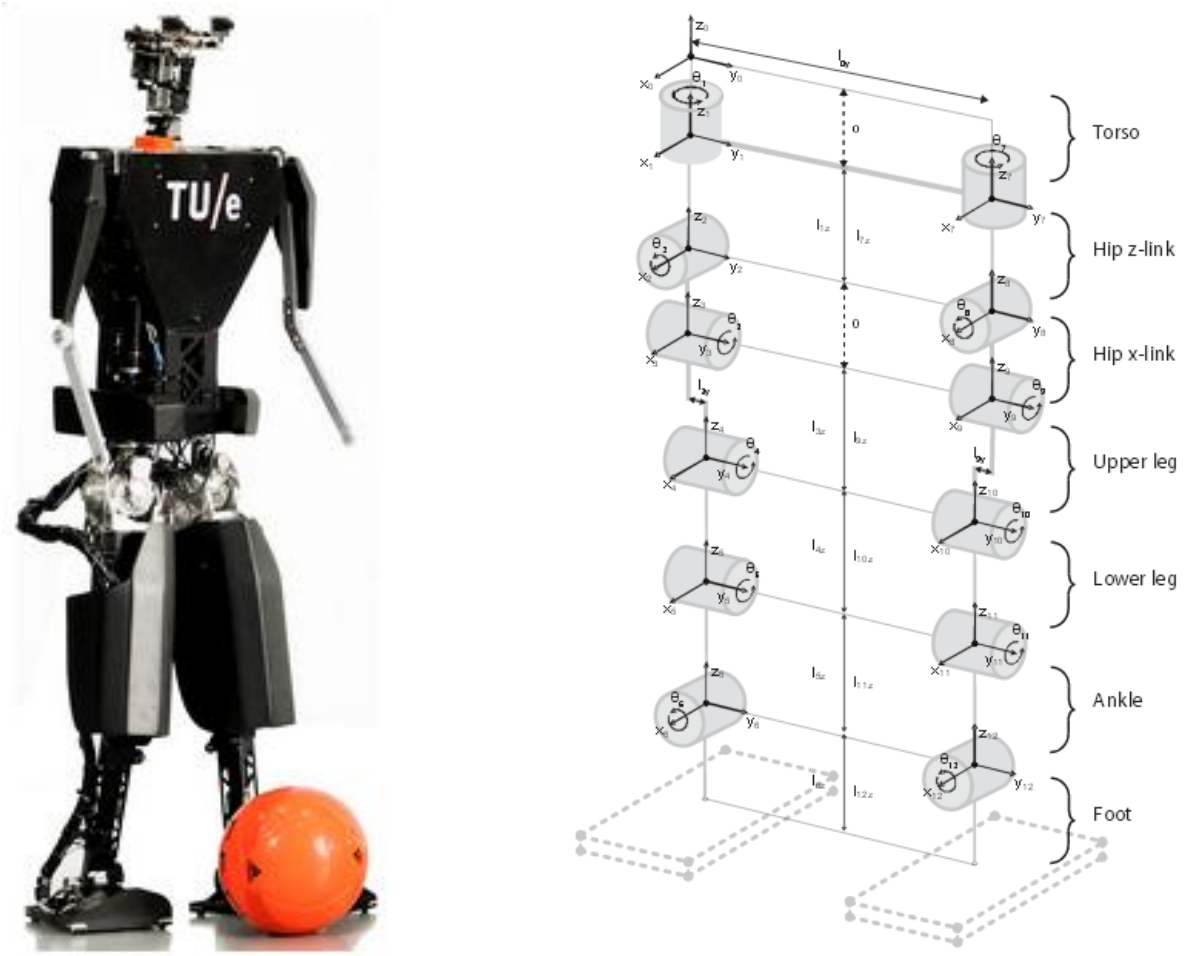


FIGURE 2. 1A. THE HUMANOID ROBOT TULIP WITH COVERS [6]. B. A SCHEMATIC REPRESENTATION OF THE JOINTS IN TULIPS' LEGS [7].

3. SOFTWARE

In this chapter, the used software and scripts are further explained.

3.1 GAZEBO

Gazebo is a multi-robot simulator, capable of simulating robots, sensors and objects, created at the University of Southern California in 2002 [8]. Gazebo uses multiple physics engines including Open Dynamics Engine (ODE) and Bullet. In Gazebo, many parameters of these physics engines can be adjusted. As was determined in [2], the number of iterations used should be 3000 in order to achieve convergence in the solution.

The model in Gazebo consists of a collection of shapes. Each one of these shapes has a given weight, center of gravity and moment of inertia around the different axes. The values of these parameters were estimated in [7]. These shapes are then connected with joints. Each joint has a certain friction and damping. The friction and damping values are fitted to experimental data in [2].

3.2 MATLAB

Matlab is a mathematical software package that is often used in scientific applications. Matlab can process most kinds of data and therefore is a good meeting point for the experimental and simulated data. T. Assman wrote a GUI for comparing the experiments and simulations graphically, as described in [2]. Figure 3.1 is a detail from one of these graphs and will be used to explain the different line types.

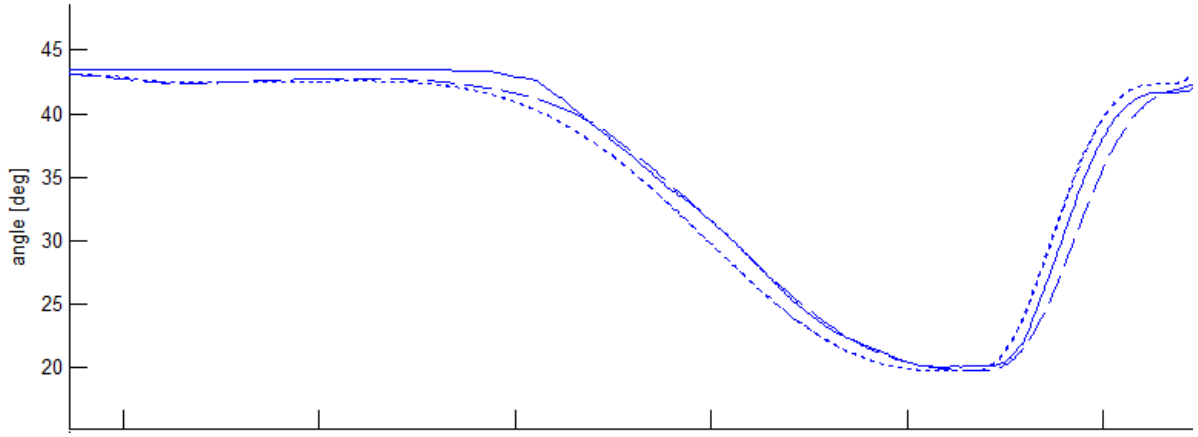


FIGURE 3. 1 DETAIL FROM ONE OF THE JOINTS IN GRAPHICAL REPRESENTATION

The dotted lines are the desired joint angles, stored in an array $\boldsymbol{Y}_{desired}$. Since this is the same for both the experiment and the simulation, the lines should overlap exactly. Due to the discrete time, this is only possible with an accuracy of 0.05 seconds. The dashed line is the simulated 'measured' value of the joint angle and the solid line is the measured joint angle of the experiment. These two lines should overlap if the model perfectly represents the reality.

Another script was written for this bachelor thesis, using components of the aforementioned GUI. This script semi-automatically synchronizes the experiment data with the simulation data and calculates the difference between the realized angles. For every joint, this results in an array of errors $\boldsymbol{E}$. Then, the stroke of that joint is calculated using formula 3.1.

$$S = max(\boldsymbol{Y}_{desired}) - min(\boldsymbol{Y}_{desired}) \tag{3.1}$$

With this stroke S, a dimensionless measure for the total error is calculated using formula 3.2.

$$RMS = \frac{\sqrt{\frac{1}{N}\sum_{i=1}^{N} \boldsymbol{E}_i^2}}{S} \tag{3.2}$$

This formula consists of the formula for the RMS value, taken from [9], in which N stands for the number of data points, divided by the stroke to create a dimensionless RMS value. This allows joints with a larger stroke to have a larger error and demands a higher accuracy from the smaller strokes The maximum error is also made dimensionless using formula 3.3.

$$m = \frac{max(\boldsymbol{E})}{S} \tag{3.3}$$

Since the absolute size of the strokes is not as important as their relation to each other, an array is made containing all the strokes from all the joints. Using the largest stroke in that array, all strokes are normalized from zero to one. This way, the relative size, and thereby importance, of these strokes is much clearer.

For the data provided by the IMU, a similar code is included. The only difference is that there is no desired value available. This could be calculated since the state of every joint is known, but instead the largest stroke of either the experiment or the simulation is used. The used script can be found in Appendix A.

Based on these data, the dimensionless RMS value, the dimensionless maximum error and the normalized stroke size, any alterations to the model will be judged to be an improvement or not in Chapter 4. This quantification of the error also allows for statements about the absolute accuracy of the model.

4. ANALYSIS

In this chapter, differences between the reality and the simulation will be investigated and an improvement to the model will be attempted. The different movements are forward stepping, sideward stepping and point turning.

4.1 FORWARD STEPPING

Both TUlip and the model performed a forward stepping experiment of which the joint angles are plotted over time in Figure 4.1. To clarify the patterns in the graph, the stance leg is named. As can be seen, the joint angles of the stance leg hardly change.

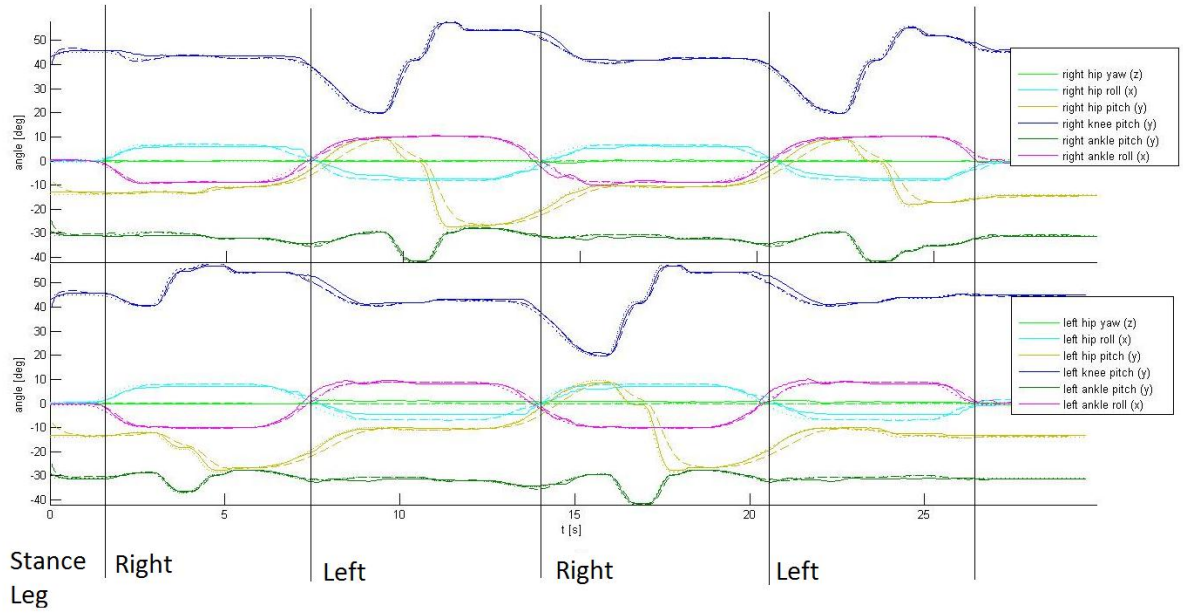


FIGURE 4. 1 JOINT ANGLES IN A FORWARD STEPPING EXPERIMENT

Even before a careful analysis of the data was made, it became clear that there was a fault in the model. When a leg was moved forwards, the toe of that foot came much closer to the ground in the simulation than in the real experiment. Since this points to a fault in the pitch, the joint angles of the pitching joints in the hip, knee and ankle where examined. The greatest error was found in the hip pitch joint angle. In Figure 4.2, a detail of 4.1 is shown where the difference between the green dashed line, the simulation, and the green solid line, the experiment, is clearly visible.

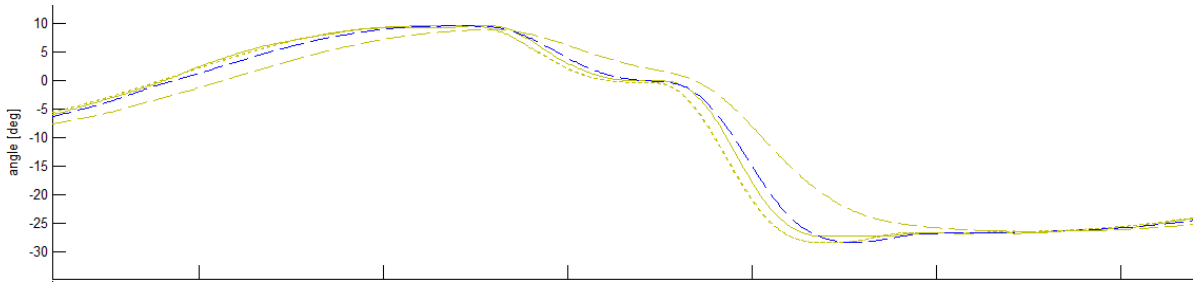


FIGURE 4. 2 DETAIL OF THE HIP PITCH JOINT ANGLE, IN GREEN THE INITIAL SITUATION, IN BLUE THE IMPROVED SITUATION

The simulation fails to match the gradient of the desired angle, thereby causing the error in angular position. Since the gradient in this graph represents the angular velocity of the joint and the velocity can be decreased by the damping in that joint, the damping values of the joints in the leg were examined. By reducing the

damping in the hip pitch joint, the simulation now followed the blue line in Figure 4.2, which is a clear improvement. The damping value now also fell in the same range as the other joint damping values which were determined by T. Assman in [2].

To quantify the error in the model, thereby allowing an objective way of assessing changes to the model, the script described in Chapter 3.2 was used. In order to reduce the chance of large measurement errors, the experiments where done twice. Since the RMS value varied less than 1% and the maximum error less than 2% between these experiments, only the data of one of these experiments is shown. The other experiment and the numerical data that was used for these representations can be found in Appendix B. In Figure 4.3, the dimensionless RMS value, maximum error and normalized stroke are plotted in a bar graph.

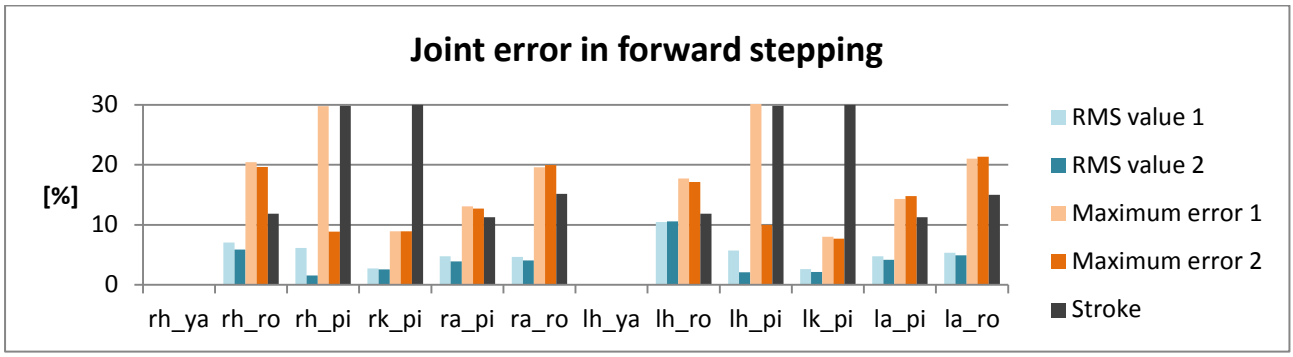


FIGURE 4. 3 ERROR IN THE JOINT ANGLES (0-30)

The abbreviations underneath every group of bars, describe which joint is meant. The first letter stands for 'left' or 'right', the second for 'hip', 'knee' or 'ankle' and the two letters after the underscore stand for 'yaw', 'roll' or 'pitch'. In the walking strategies that don't use the yaw joint in the hip, this group is left out. Because of the small stroke in this joint, small errors would be overly present reducing the readability of the graph. The blue bars give the dimensionless RMS value, the orange bars represent the dimensionless maximum error and the dark grey bar gives the normalized size of the stroke. For visibility, the strokes are normalized on a scale of 0-30 instead of the 0-1 scale described in Chapter 3.2. The scale of normalization can always be found in the caption of the graph. In this graph, the lighter version of the color represents situation 1, which is the initial situation, and the darker version of the color represents situation 2, with the altered damping in the hip pitch joint. Because the walking strategy is still the same, the stroke doesn't change and is only plotted once.

When looking at Figure 4.3, the drastic increase in accuracy of the hip pitch joints is immediately noticed. Beside this, the accuracy in the hip roll joints, the knee joints, right ankle pitch and left ankle roll joint improves. For the left ankle pitch joint, the RMS value decreases, but the maximum error increases. This is because the new simulation matches the desired angle slightly better, as is visible in Figure 4.4.

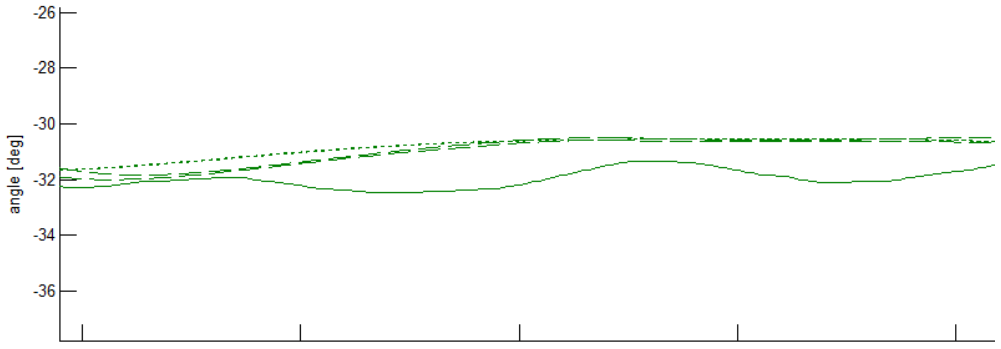


FIGURE 4. 4 MAXIMUM ERROR OF THE ANKLE PITCH JOINT

Compared to the other joints, the roll joints in the hip and ankle do rather poorly. Except for the left hip roll joint, they barely make the 5% RMS value mark, set in Chapter 1 and none of the joints manage to stay below the 10% maximum error. In Figure 4.5, as closer look is taken to the joint angles of the left hip and ankle roll joints. The ankle roll follows the desired angle quite well, but then starts deviating when weight is put on the foot. This behavior is also seen in the right ankle. Because the joint angle in the experiment oscillates towards the desired joint angle, this offset is probably caused by some elastic element within the ankle of the robot. At this point it is unclear if and how this could be modeled. The roll in the hip shows a much simpler error. It doesn't reach the desired joint angle and follows the desired angle at a certain angular lag. This appears to be some sort of backlash effect. This will be further discussed in Chapter 4.2.

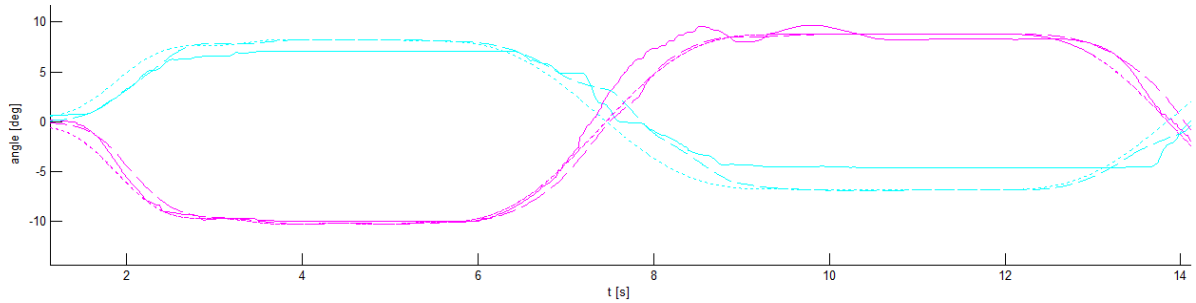


FIGURE 4. 5 DEATIL SHOWING THE LEFT HIP AND ANKLE ROLL JOINTS

After this analysis of the joint angles, the data from the IMU was examined. As described in Chapter 3.2, the script works mostly the same for the IMU as for the joint angles, with the exception that the stroke can vary. Therefore, in Figure 4.6 both strokes are plotted as well.

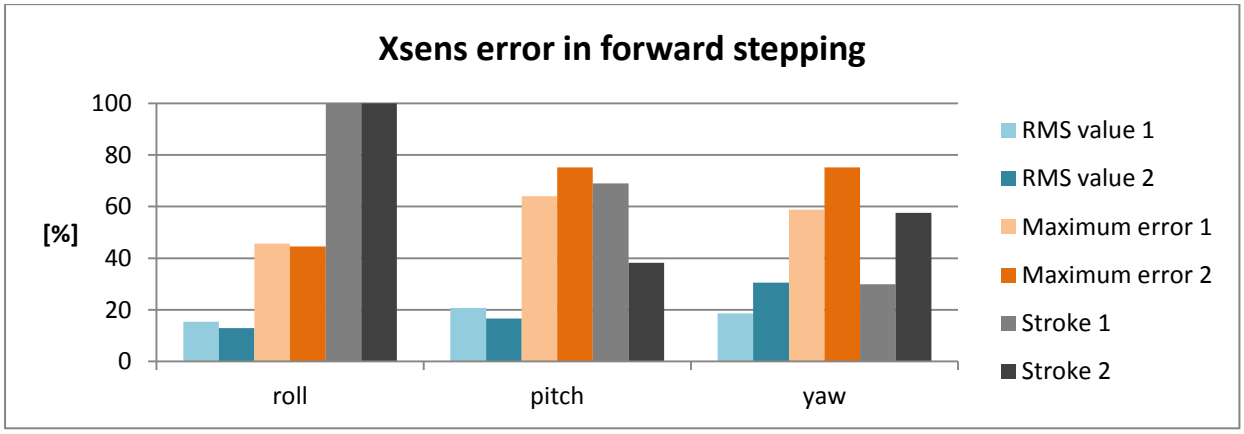


FIGURE 4. 6 DIMENSIONLESS ERROR IN UPPERBODY ANGLE

The roll continues to have the largest error of the three angles, although it did decrease, but both the RMS value and the maximum error decrease in situation 2, with decreased damping in the hip pitch joint. The stroke of the pitch decreases substantially and although the maximum error increases somewhat, the RMS value shrinks. The stroke of the yaw appears to increase, but this is because of the smaller size of the maximum stroke, the roll.

In Figure 4.7, the angles recorded by the IMU during the experiment and gathered in the simulation are plotted against time.

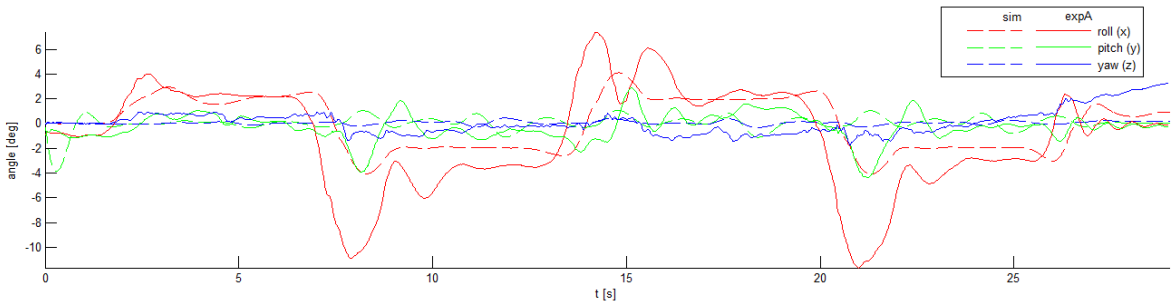


FIGURE 4. 7 DATA FROM THE IMU DURING FORWARD STEPPING

From this graph, several relations can be deducted. Firstly, the yaw, plotted in blue, coincides fairly well until the last 5 seconds of the experiment. At this point the yaw starts to wander. In the other experiment, this occurred after 10 seconds. This is either caused by a fault in the IMU or by a random event in the surrounding, such as a bump in the floor causing the foot to land a little early or friction force from the data cable. Either way, it is impossible to model the wandering of the yaw. The roll, plotted in red, shows a similar movement, a deflection followed by a damped vibration, but with a different amplitude and damping. Also, the experimental value appears to have an offset of -0.5 degrees. This could be caused by an asymmetrical assembly of either the IMU or the entire robot. Since the offset is so small and could be different or gone when TUlip is reassembled, this will not be modeled. The error in amplitude and damping are most likely caused by the errors in the hip and ankle roll joints. The elastic element in the ankle roll joint could cause the more prominent vibration, and the error in the hip roll joint could cause the larger angle. The pitch, plotted in green, shows no clear correlation between the experimental and simulated data. On a side note, this angle stays within the ±4 degree range, making exact predictions difficult.

4.2 SIDEWARD STEPPING

The most important movement in sideward stepping is de roll of the legs, made possible by the roll joints in the hip and ankle. As became clear in Section 4.1, the angle in the hip roll joint doesn't always reach the desired angle and follows with a certain lag. Figure 4.8 shows the joint angles during the side stepping experiment plotted against time. Figure 4.9 shows the described lag in a detail from the hip roll joint angles.

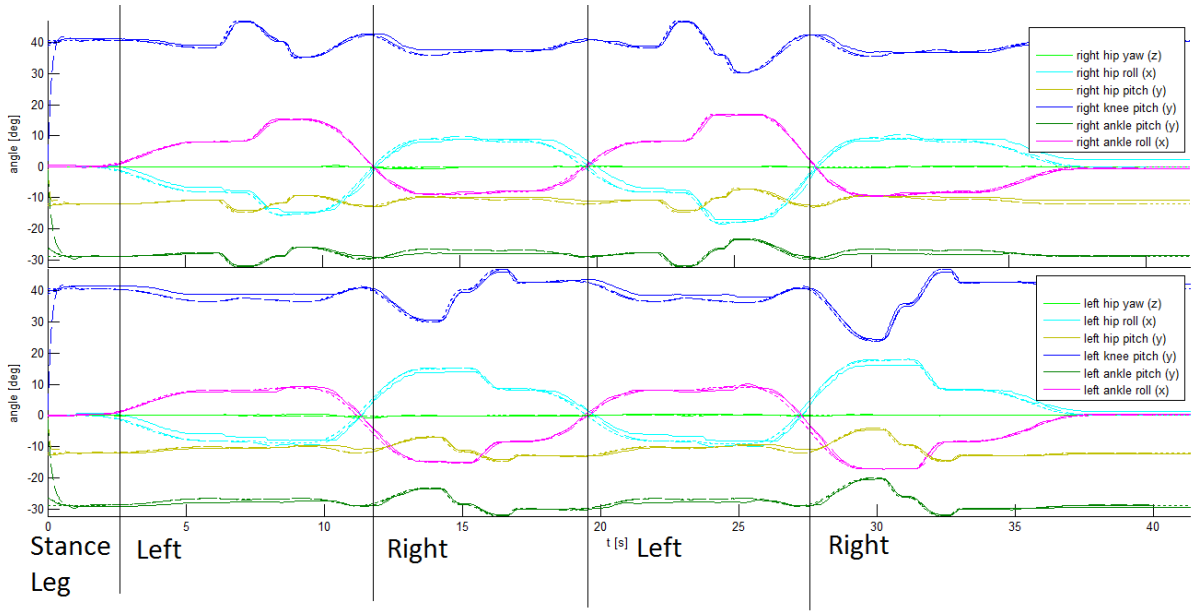


FIGURE 4. 8 JOINT ANGLES IN SIDE STEPPING EXPERIMENT

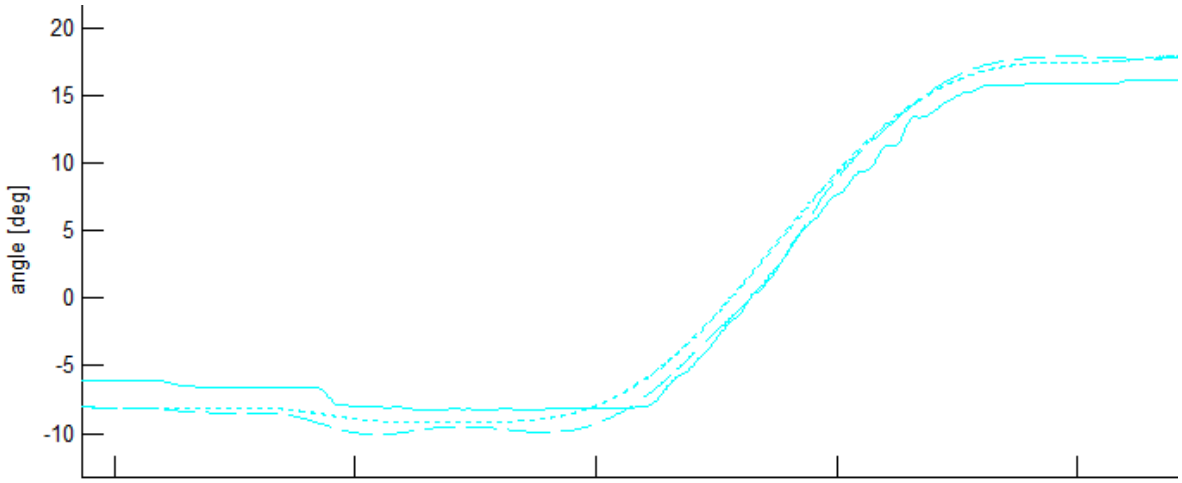


FIGURE 4. 9 DETAIL FROM THE HIP ROLL JOINT ANGLE

Because the joint angle remains constant and then follows with a certain lag when the slope of the desired angle changes direction, backlash was initially suspected. This was modeled by placing a joint between the hip roll joint and the hip pitch joint. This joint had the standard friction, no damping and a maximum range of one degree in both directions as suggested in [2].

After implementing this joint, the robot in the simulation of the side stepping experiment destabilized after 25 seconds. The experiment data in which the robot is falling or on the ground, is ignored so these experiments can still be analyzed. The robot doesn't fall in the forward stepping experiment.

In Figures 4.10 and 4.11 the errors in the forward stepping and sideward stepping experiment are shown respectively. The general trend is one of decreasing accuracy, only the left hip roll joint improves its performance.

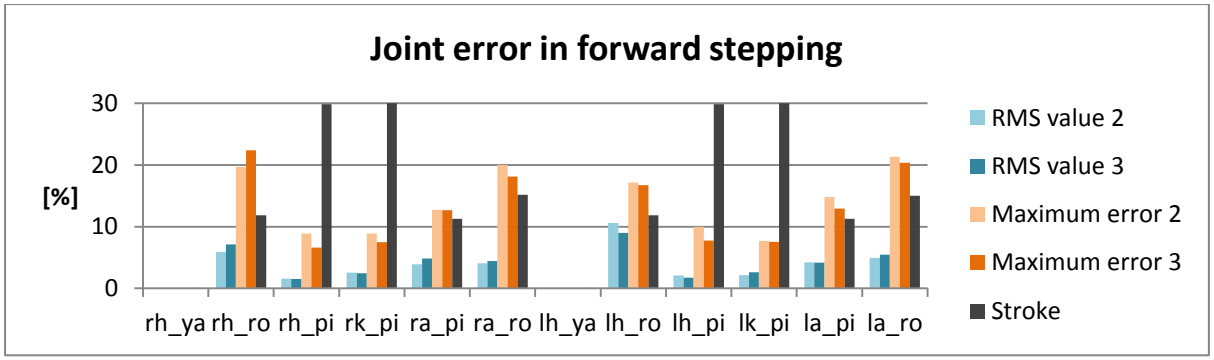


FIGURE 4. 10 ERROR IN THE JOINT ANGLES (0 – 30)

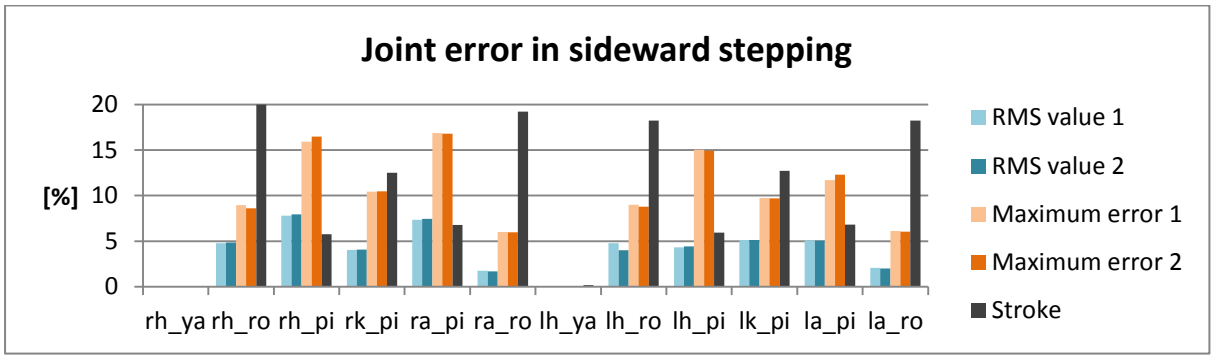


FIGURE 4. 11 ERROR IN THE JOINT ANGLES (0 – 20)

After this unsuccessful alteration of the model, the error was reinvestigated. The balance of moments on the hip roll joints shows that backlash should in fact cause a sharper than desired angle and that the cross over point should be around the neutral line, since that is where the actuator changes its direction.

A new theory involves the friction in the hip roll joints. It states that the friction in the hip roll joints is so high, that an extra stimulus from the feedback is needed to move it. This would explain the lag in following the desired joint angle. At the stages with constant angle, the feedback might not deliver enough moment to overcome the friction. This could explain the offset from the desired angle.

As an initial attempt to simulate the effect of high friction, every joint was given a high friction, over five times the original friction. In Figure 4.12 the results of this change in parameters is shown. The results are mixed, both roll joints improve their accuracy, but all other joints either increase their RMS value or their maximum error. This means that every joint has a certain friction that gives a certain RMS value and maximum error and, depending on the preference, has an optimal friction that can be determined. In Chapter 6, it will be further discussed whether or not this should be done. When other friction models are included in the simulator, this becomes more difficult because both the friction model and its parameters could be optimized.

Since the roll joints in both the hip and the ankle did improve on all fronts with a higher friction, an attempt was made to alter the model so the friction could be adjusted per joint. So far, this attempt has not been successful.

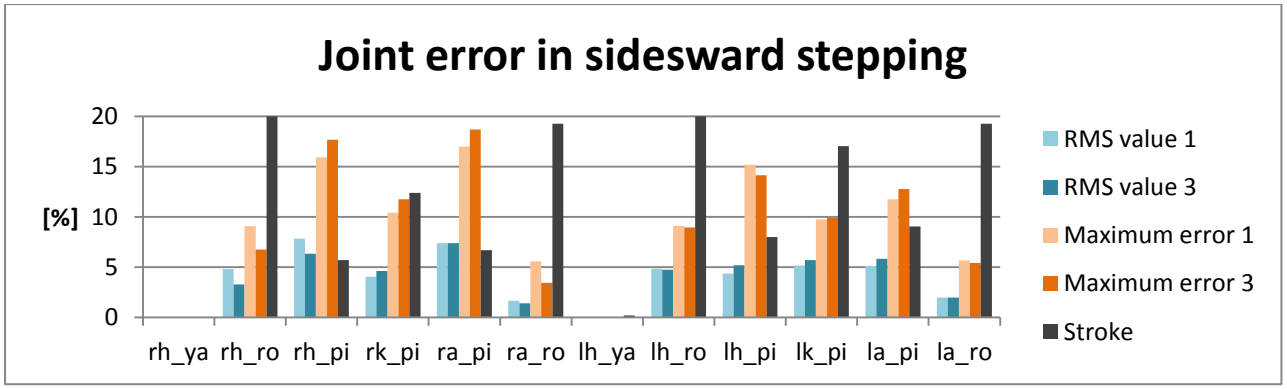


FIGURE 4. 12 ERROR IN THE JOINT ANGLES (0 – 20)

Finally, the IMU data are analyzed. In Figure 4.13 the IMU data of a sidestepping experiment are plotted against time. Once again, the yaw wanders showing the largest slope when the right foot lands, see Figure 4.8. This could be caused by backlash in the left hip yaw joint, either hip roll joints or any of the reasons named in Chapter 4.1. The roll shows a similar shape, but with an offset and different amplitude as was explained in Chapter 4.1. In the simulation, the pitch varies less than 1 degree and the IMU pitch varies about 1.5 degrees. The relevance of explaining the misalignment of these two has therefore limited relevance.

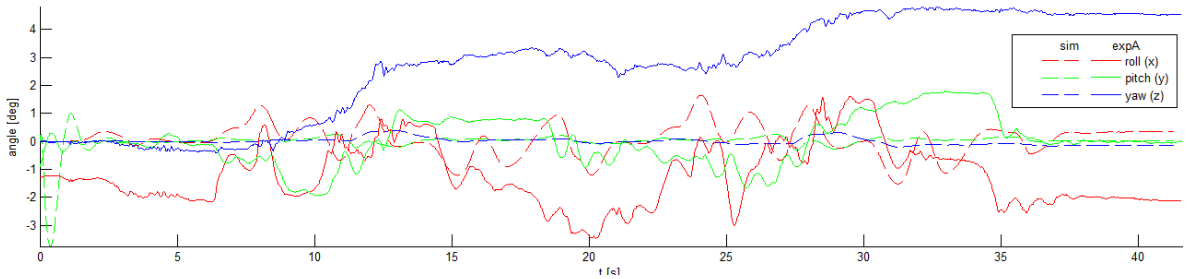


FIGURE 4. 13 IMU DATA DURING THE SIDEWARD STEPPING EXPERIMENT

4.3 POINT TURN

When plotting the joint angles of this behaviour, as is done in Figure 4.14, the error in the hip yaw joint immediately grabs the attention. This joint is not actively used in any of the previous experiments, which explains why this substantial error wasn't encountered earlier.

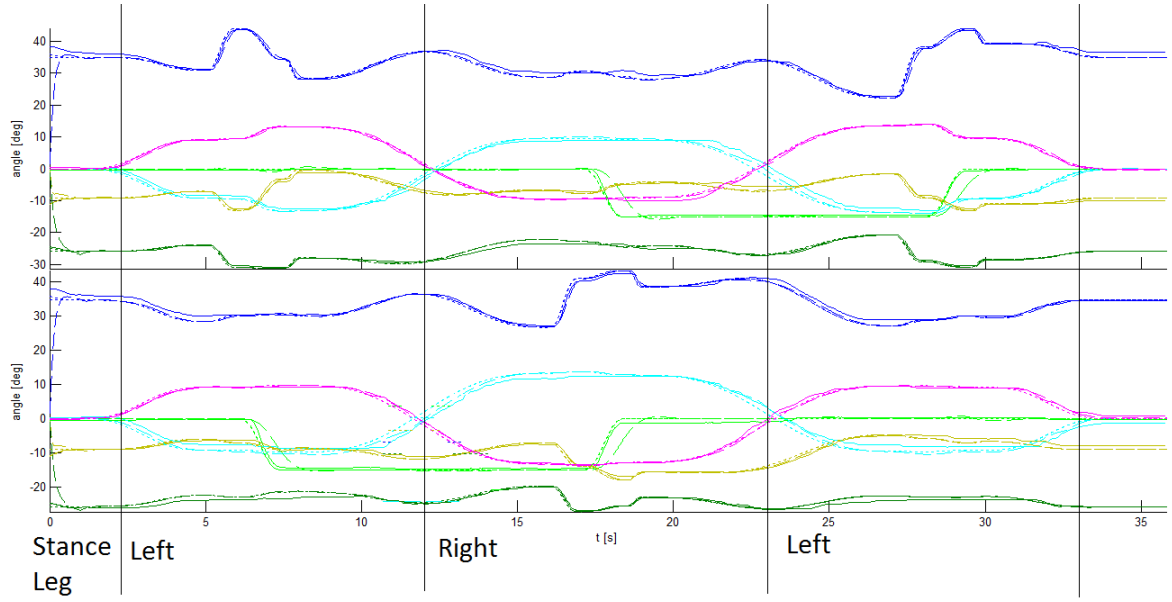


FIGURE 4. 14 JOINT ANGLES IN THE POINT TURN EXPERIMENT

In Figure 4.15, the error is shown in more detail. It shows how the simulation has a certain time lag, but then follows the desired angle with the correct slope and reaches the desired angle. Because the slope is correct, the main cause lies not with the damping of the hip yaw joint. Because the final angle is correct, backlash cannot explain this behavior. A possible explanation for this behavior is a too high rotational inertia in the model. When damping in the joint is reduced, the damped vibration at the end of the movement becomes more clearly visible, indicating significant rotational inertia. Although the parameters for the model were extensively examined in [7], J. Baelemans pointed out in his discussion that the determination of the dynamical parameters was still somewhat unsure. In Figure 4.14 it is shown that the hip yaw joint of the stance leg rotates, which means that every rotational inertia value of the model is involved. Small errors in every value could add up to show this behavior.

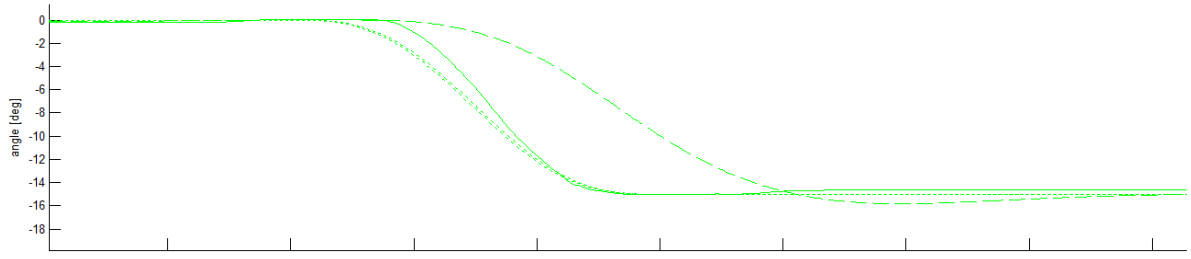


FIGURE 4. 15 DETAIL OF THE ERROR IN THE HIP YAW JOINT ANGLE

However, after numerous simulations in both the newest and the older version of Gazebo, no reduction of this error could be accomplished. Simulations done include: lowered izz values for every part of the leg, lowered izz values for the torso, lowered mass of the torso, lowered mass of the foot, lowered mass of the upper leg, lowered damping in the hip yaw joint and combinations of the above. The fact that this error still occurs means that either the simulation or the theory on the origin of the error is incorrect.

When investigating the quantified error of the point turning experiment, it therefore shows that the yaw joints do not achieve a high accuracy as can be seen in Figure 4.16. More unexpected is the poor performance of the left ankle joint. A closer examination of the left ankle pitch can be seen in Figure 4.17.

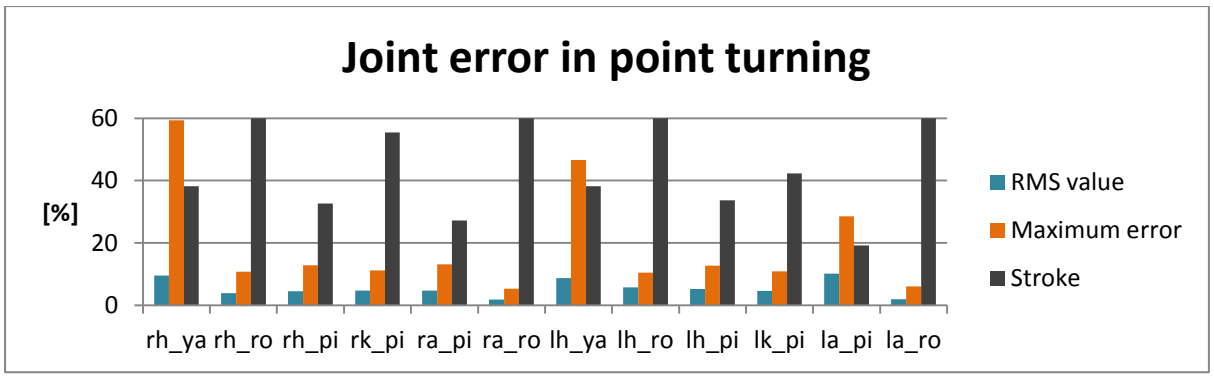


FIGURE 4. 16 ERROR IN JOINT ANGLE (0 – 60)

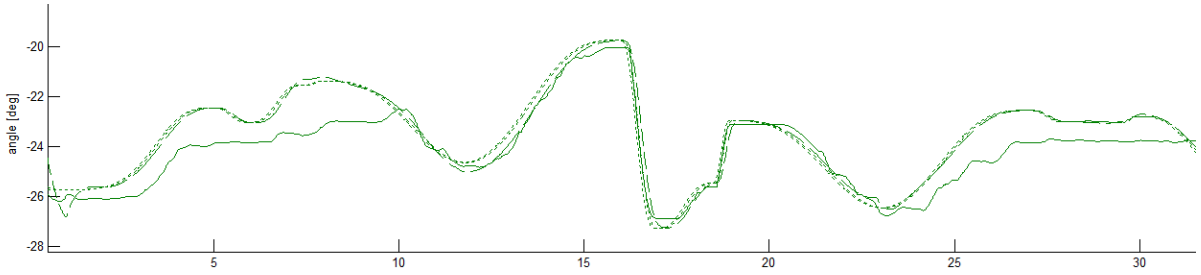


FIGURE 4. 17 DETAIL OF THE ERROR IN THE LEFT ANKLE PITCH JOINT ANGLE

When Figure 4.17 is compared to Figure 4.14, it becomes clear that this large error occurs when the weight of the robot is on that foot. Investigation of Figures 4.1 and 4.8 reveals that this is a consistent phenomenon throughout the walking strategies that also occurs to some extend in the knee pitch joint. Possible ways of modeling this error include an elastic element in the ankle pitch direction or backlash in the ankle pitch joint with a very high friction. Neither possibility has been further investigated due to a lack of time.

The reason that this error only comes to light at this point, is that it is an error of 2 degrees at most, which means that it is easily suppressed. Only in the point turn movement, is the area where the error occurs, when the leg is the stance leg, large enough to be significant. This is because the point turn movement is the only movement that has an imbalance in the stance legs; twice left and once right. Therefore, the part where the left leg is the stance leg covers about two thirds of the total experiment time, leaving a significant impact.

Finally, the IMU data is analyzed again. In Figure 4.18, the IMU data is plotted against time.

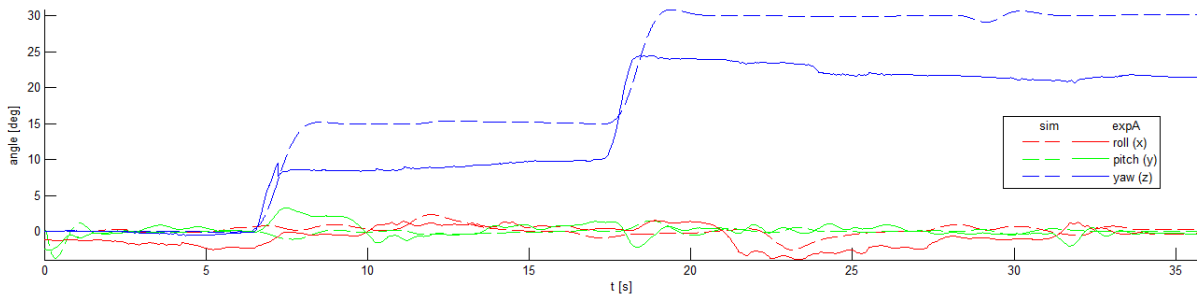


FIGURE 4. 18 IMU DATA FROM THE POINT TURN EXPERIMENT

The most obvious error occurs in the yaw of the robot. Although a similar trend is visible, the IMU immediately falls behind. Besides that, it never reaches the target value of 30 degrees. It consistently misses a third of the actual rotation and besides that wanders up to 4 degrees in the last 15 seconds, where the robot shouldn't be rotating. TUlips joints actually make a 15 degree movement twice and a 5 degree slip per step seems unlikely since the entire weight of the robot is on the stance foot. Therefore, a likely conclusion is that the IMU yaw has a relatively low accuracy. Because of the large amplitude of the yaw, the pitch and roll are difficult to observe, therefore Figure 4.19 shows a detail where the pitch and roll are clearly visible.

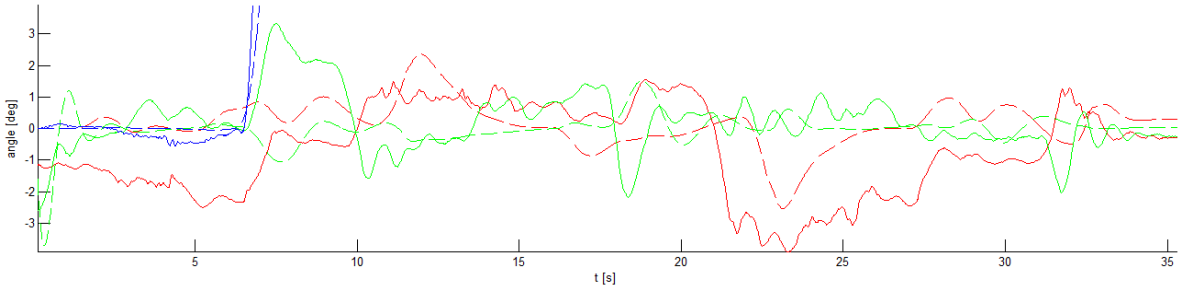


FIGURE 4. 19 DETAIL OF THE PITCH AND ROLL FROM THE IMU DATA

From this, the familiar conclusions about the roll can be made; there is a deviation from the center line, the general shape of the experiment and the simulation are compatible but the experiment has a larger amplitude, possibly caused by backlash in the hip and ankle roll joints. The pitch has a more distinctive shape than in previous experiments, allowing for more detailed conclusions. One of the observations made, is that the experiment and simulation appear to be in antiphase. When the IMU shows a notable peak, the simulation peaks in the opposite direction. This is an unexpected observation that will be further discussed in chapter 6.

5. CONCLUSION

The goal of this project was to determine the accuracy of the model of TUlip in the Gazebo simulator. To achieve this, three walking strategies have been conducted both on the real robot and in the simulator. For the comparison of the results a script was written that calculates the difference between the simulator and experiment results. From this difference, the RMS value and maximum error is calculated and made dimensionless by dividing by the total stroke of that joint. It is said that the model is accurate when the dimensionless RMS value does not exceed 5% and the dimensionless maximum error does not exceed 10% and that the model is adequate when these values stay below 10% and 20% respectively.

In the hip pitch joints, a major difference between the experiment and the simulation was found. It was caused by the damping in that joint which was modeled too high. In the ankle roll joints, some sort of elastic element was identified but not modeled. However, modeling a high friction in this joint improved the accuracy. In the hip roll joint a backlash like effect was discovered and successfully modeled by simulating a high friction in that joint. Assigning that friction only to the ankle and hip joints has not succeeded. The ankle pitch shows a small deviation in the stance leg, this might be modeled with high friction backlash or an elastic component, but this has not been tested. The hip yaw joint shows a serious time delay in the simulation. This was thought to be caused by high rotational inertia, but this could not be proven. The actual cause is still unknown.

The IMU data doesn't match well between the simulation and the experiment. The yaw is found to be poorly measured by the IMU in the point turn experiment. The roll matches quite nicely, although the robot has a larger amplitude because of backlash in the roll joints. The pitch has not been clearly visible in any of the experiments, but there was very little correlation to be found between the IMU data and the simulation.

The forward stepping is simulated adequately, according to the previously postulated boundaries. However, when only the most important joints for this movement are examined, the hip and knee pitch joints, the movement is simulated accurately.

The sideward stepping is simulated adequately, however when only looking at the most important joints, the hip and ankle roll, the movement is simulated accurately.

The point turn is simulated inadequately because of a large error in the simulation of the hip yaw joint movement and a smaller error in the ankle pitch joint. Since all of the hip yaw error originates from the time delay, solving this error would instantly make the simulation adequate.

6. DISCUSSION

In this chapter, the side notes that come with the results from the conclusion are made.

Over the course of this project, attempts have been made to model the measured results as accurately as possible. This might suggest that these results are considered to be 'the true state of the robot'. However, the robot is a real world entity, changing by nature. No experiments have been conducted to examine the amount of change per unit of time for it would be difficult to determine an adequate sample time. During the winter recess the robot will hardly change except for aging, during the RoboCup the effects of wear will be significant and after maintenance an entire part of the robot could be new. The only sound conclusion is that the robot changes over time.

Since this is the case, it raises the question of how much effort should be put into fine tuning the model to exactly match the experiments conducted at a certain time. For instance, one could determine the optimal friction values for every joint or determine the amount of backlash in a joint, but these values would have to be updated every now and then. A balance must be found between the amount of time put into improving the model and the accuracy gained from these improvements. This choice could still be 'the model should be very accurate', but this decision should be made knowing that this does not mean a onetime only time investment.

Furthermore, the way of making the RMS value and maximum error dimensionless can be questioned. This method, using the stroke of the joint, was chosen over using the desired angle value to make every spot dimensionless before using statistics and using one value for all the joints. This is because this seemed to give the most useful information. However, it also forces the small movements to be very accurate. This is unrealistic since the same joint that performs a large stroke in one movement, performs a small stroke in another movement. For instance, if a joint error is largely caused by backlash, than the accuracy of the joint will probably not change but the appreciation will. In this thesis, this occurred in the ankle pitch joints. In both the forward and sideward stepping, it had a significant stroke, suppressing the small error of 2 degrees or less that did occur. In the point turn, however, the stroke decreased thusly that this small error became a large dimensionless error. In my opinion, this dimensionless error provides the most relevant information, but its limitations should be kept in mind when evaluating the results it produces.

Combining these two points, it can be concluded that the model functions very well. Even in the point turn, the time delay hardly has any effect on the larger picture at all. This doesn't mean that the model shouldn't be further improved, but that the effect of this extra effort should be monitored closely, since the model already functions. If the movement strategies evolve to rely on fast, dynamical effects and the current model ceases to suffice, improvements to the model have a high efficiency and should receive attention. The adequacy of this model with regards to fast, dynamical effects has not been investigated in this thesis, so this statement is hypothetical.

The IMU is not modeled very accurately for multiple reasons. Firstly, it involves small angles that are determined by all joints in the robot, thus accumulating all errors that are present in the actual robot, but not in the model. Secondly, the IMU has a measurement uncertainty. Combined, it is not surprising that the model and the experiment don't match very well. The question is, whether this is important or not. As long as the IMU and the simulation show a similar shape, the movement of the robot agrees and it fulfills a task as double check. However, showing the same trend is important and therefore a short follow up study might be conducted into the pitch sensing of the IMU by artificially creating large pitch angles, rotating the robot, and observing the behavior of the IMU data.

BIBLIOGRAPHY

[1] “Homepage of the ROS” http://www.ros.org

[2] T.M. Assman, Coaches: P.W.M. van Zutven, H. Nijmeijer, Humanoid push recovery stepping in experiments and simulations, 114, DCT 2012.059, Internal Report (2012)

[3] “Self-learning Robots using Evolutionary and Genetic Algorithms” http://www.idt.mdh.se/kurser/ct3340/ht11/MINICONFERENCE/FinalPapers/ircse11_submission_22.pdf

[4] “Homepage of the humanoid league in Robocup” http://www.tzi.de/humanoid/bin/view/Website/WebHome

[5] Tekscan, Flexiforce sensors user manual." http://www.tekscan.com/pdf/FlexiForce-Sensors-Manual.pdf

[6] “Homepage of TUlip group” http://www.techunited.nl/nl/tulip

[7] J.A.J. Baelemans, Coaches: P.W.M. van Zutven, H. Nijmeijer, Parameter estimation of humanoid robots using the center of pressure, 106, DCT 2013.023 (2013)

[8] “Homepage of the Gazebo simulator” http://www.gazebosim.org

[9] “Explanation of the root-mean-square value on the Wolfram Alpha website” http://www.wolframalpha.com/input/?i=root%E2%80%90mean%E2%80%90square&lk=1&a=ClashPrefs_*MathWorld.Root!-Mean!-Square-

APPENDICES

APPENDIX A

This appendix contains the Matlab script written for and used in this bachelor thesis.

```
function [errorVals,errorValx] = errorfind(simname,expname,x)
%x=1 for forward stepping, x=2 for side stepping and x=3 for pointturn
double step
%simname and expname only need to contain the folder in which the
experiment is logged or the simulation is saved. The datafiles from the
Groovy stack are used automatically

exp = load_exp(expname);

%% Joints
joint    = simload_joints(
['../../tue_tulip_ros_groovy_stack/tulip_controller/datafiles/' simname
'/datafile_joints.rtp']);

N = [3,21;6,21;3,17];       %deduced parameters for finding the starting
index

startInd = find(exp.jointsens.q{N(x,1)}.desiredAngle(700:end) >=
joint{N(x,1)}.desired_position(700),1,'first') + N(x,2);       %Finding
start index of experiment by aligning the peak at t = 7.00 s, skipping
first 7 s of experimental data

errorVals = zeros(3,12);
maxRange = 0;

for c = 1:12

Size = size(joint{c}.time,1);
%Size = 2500;       %For a subset measurement

interVal = interp1q(exp.jointsens.time(startInd:startInd+Size)-
exp.jointsens.time(startInd),
exp.jointsens.q{c}.jointAngle(startInd:startInd+Size),joint{c}.time(1:Size)
);

%Plotting
% figure(c)
% plot(joint{c}.time,joint{c}.position)
% hold on
% plot(joint{c}.time,interVal,'r')

rangeVal = max(joint{c}.desired_position(1:Size)) -
min(joint{c}.desired_position(1:Size));     %Determining total range of
motion to make the result dimensionless
errorVal = joint{c}.position(100:Size) - interVal(100:Size);
%Calculating the difference between the simulated and experimental motion

if rangeVal > 0.01
    errorVals(:,c) = [100*sqrt(mean(errorVal.^2))/rangeVal
100*max(abs(errorVal))/rangeVal rangeVal]';     %Array containing standard
RMS of error [%], maximum absolute error [%] and the range [-]
```

```
        %find(abs(errorVal)==max(abs(errorVal)))        %uncomment to print
the array location of the maximum error
else
    errorVals(:,c) = [NaN NaN rangeVal]';
end

if rangeVal > maxRange
    maxRange = rangeVal;
end
end

errorVals(3,:) = errorVals(3,:)/maxRange;
fprintf(' rh yaw     rh roll    rh pitch  rk pitch  ra pitch   ra roll   lh
yaw   lh roll   lh pitch  lk pitch  la pitch  la roll\n')

%% Xsens
clear maxRange interVal rangeVal
xsens    = simload_xsens(
['../../tue_tulip_ros_groovy_stack/tulip_controller/datafiles/' simname
'/datafile_xsens.rtp']);

errorValx = zeros(3);
maxRange = 0;

startIndx = find(exp.xsens.time>=exp.jointsens.time(startInd),1,'first');
endIndx =
find(exp.xsens.time>=exp.jointsens.time(startInd+Size),1,'first');
xsens.time = xsens.time/1e9;

interVal = {interp1q(exp.xsens.time(startIndx:endIndx)-
exp.xsens.time(startIndx),exp.xsens.northframe.roll.rad(startIndx:endIndx),
xsens.time);
            interp1q(exp.xsens.time(startIndx:endIndx)-
exp.xsens.time(startIndx),exp.xsens.northframe.pitch.rad(startIndx:endIndx)
,xsens.time);
            interp1q(exp.xsens.time(startIndx:endIndx)-
exp.xsens.time(startIndx),exp.xsens.northframe.yaw.rad(startIndx:endIndx),x
sens.time)};

interVal{3} = interVal{3} - interVal{3}(1);         %Correcting initial
offset in yaw

%plotting
% figure(c+1)
% plot(xsens.time,xsens.roll,xsens.time,interVal{1},'r')
% figure(c+2)
% plot(xsens.time,xsens.pitch,xsens.time,interVal{2},'g')
% figure(c+3)
% plot(xsens.time,xsens.yaw,xsens.time,interVal{3},'c')


rangeVal = [max(max(interVal{1}) - min(interVal{1}),max(xsens.roll(50:end))
- min(xsens.roll(50:end)));     %Determining total range of motion to make
the result dimensionless
            max(max(interVal{2}) -
min(interVal{2}),max(xsens.pitch(50:end)) - min(xsens.pitch(50:end)));
%Determining total range of motion to make the result dimensionless
```

```
            max(max(interVal{3}) - min(interVal{3}),max(xsens.yaw(50:end))
- min(xsens.yaw(50:end)))];     %Determining total range of motion to make
the result dimensionless

errorVal = {xsens.roll(50:Size) - interVal{1}(50:Size);
%Calculating the difference between the simulated and experimental motion
            xsens.pitch(50:Size) - interVal{2}(50:Size);
%Calculating the difference between the simulated and experimental motion
            xsens.yaw(50:Size) - interVal{3}(50:Size)};
%Calculating the difference between the simulated and experimental motion

for k = 1:3

    if rangeVal(k) > 0.01
        errorValx(:,k) = [100*sqrt(mean(errorVal{k}.^2))/rangeVal(k)
100*max(abs(errorVal{k}))/rangeVal(k) rangeVal(k)]';     %Array containing
the RMS of error [%], maximum absolute error [%] and the range [-]
    else
        errorValx(:,k) = [NaN NaN rangeVal(k)]';
    end

    if rangeVal(k) > maxRange
        maxRange = rangeVal(k);
    end
end

errorValx(3,:) = errorValx(3,:)/maxRange;

%% Footsensors
footsens =
simload_footsens(['../../tue_tulip_ros_groovy_stack/tulip_controller/datafi
les/' simname '/datafile_footsens.rtp']);

startIndf =
find(exp.footsens{1}.time>=exp.jointsens.time(startInd),1,'first');
endIndf =
find(exp.footsens{1}.time>=exp.jointsens.time(startInd+Size),1,'first');
footsens.time = footsens.time/1e9;

%plotting
f = 2;
for e = 1:2
    for d = 1:4
        ind = round((4/3)*d^3 - 10*d^2 + (65/3)*d - 10);
        figure(d+(e-1)*4)
        plot(footsens.time(500:end),footsens.f{(e-
1)*4+ind}.val(500:end),exp.footsens{f}.time(startIndf:endIndf)-
exp.footsens{e}.time(startIndf),120*(exp.footsens{f}.f{d}.val(startIndf:end
Indf)-min(exp.footsens{f}.f{d}.val(startIndf:endIndf))),'r')
        title(footsens.f{(e-1)*4+ind}.name)
    end
    f = f - 1;
end
```

APPENDIX B

In this appendix, all quantified errors that have been calculated are presented numerically.

The first row shows the RMS value, the second row shows the maximum error and the third row contains the stroke, normalized from 0-1.

Forward stepping situation 1, experiment 1

rh_ya	rh_ro	rh_pi	rk_pi	ra_pi	ra_ro	lh_ya	lh_ro	lh_pi	lk_pi	la_pi	la_ro
NaN	7,5128	5,9964	2,7051	4,5241	4,476	NaN	9,7758	5,4122	2,4804	4,5156	4,9588
NaN	19,4263	29,0954	8,7207	11,3077	19,212	NaN	17,7841	30,5072	7,4169	12,393	20,1729
0,0011	0,3954	0,9947	0,9998	0,3756	0,5052	0,0011	0,395	0,9947	1	0,3756	0,4998

roll	pitch	yaw
14,3975	19,5708	48,4934
41,5993	63,0448	96,9009
1	0,7681	0,6291

Forward stepping situation 1, experiment 2

rh_ya	rh_ro	rh_pi	rk_pi	ra_pi	ra_ro	lh_ya	lh_ro	lh_pi	lk_pi	la_pi	la_ro
NaN	7,0513	6,1679	2,7218	4,7934	4,6616	NaN	10,4797	5,7061	2,6636	4,7662	5,3703
NaN	20,4514	29,7945	8,9105	13,0914	19,6035	NaN	17,7553	31,1139	8,0208	14,3276	21,0608
0,0011	0,3954	0,9947	0,9998	0,3756	0,5052	0,0011	0,395	0,9947	1	0,3756	0,4998

roll	pitch	yaw
15,4208	20,6803	18,594
45,6977	63,9887	58,8581
1	0,6897	0,2996

Forward stepping situation 2, experiment 1

rh_ya	rh_ro	rh_pi	rk_pi	ra_pi	ra_ro	lh_ya	lh_ro	lh_pi	lk_pi	la_pi	la_ro
NaN	6,1091	1,4149	2,5594	3,6422	3,8821	NaN	10,4667	1,8794	2,0397	4,152	4,8229
NaN	17,0006	7,4706	7,663	11,5941	19,5489	NaN	16,3207	8,9263	6,7493	12,9618	20,382
0,0011	0,3954	0,9947	0,9998	0,3756	0,5052	0,0011	0,395	0,9947	1	0,3756	0,4998

roll	pitch	yaw
12,1017	17,927	47,6857
40,3776	73,1379	93,4135
1	0,382	0,9188

Forward stepping situation 2, experiment 2

rh_ya	rh_ro	rh_pi	rk_pi	ra_pi	ra_ro	lh_ya	lh_ro	lh_pi	lk_pi	la_pi	la_ro
NaN	5,8655	1,5785	2,5648	3,922	4,0787	NaN	10,595	2,0881	2,1521	4,2103	4,9202
NaN	19,651	8,8605	8,9053	12,7304	19,9864	NaN	17,1318	9,9883	7,6991	14,8225	21,357
0,0011	0,3954	0,9947	0,9998	0,3756	0,5052	0,0011	0,395	0,9947	1	0,3756	0,4998

roll	pitch	yaw
12,8968	16,5924	30,521
44,6052	75,1389	75,2269
1	0,3823	0,5756

Forward stepping with backlash, experiment 1

rh_ya	rh_ro	rh_pi	rk_pi	ra_pi	ra_ro	lh_ya	lh_ro	lh_pi	lk_pi	la_pi	la_ro
NaN	7,2547	1,5337	2,5309	4,6028	4,422	NaN	8,8412	1,6969	2.4424 3	0,7785	5,2833
NaN	19,1286	6,5565	7,5609	11,7003	17,2427	NaN	16,8191	8,0573	6,8486	13,0349	19,8164
0,0012	0,3954	0,9947	0,9998	0,3756	0,5054	0,0012	0,395	0,9947	1	0,3756	0,4998

roll	pitch	yaw
15,4827	22,4187	48,933
38,879	67,6986	95,0718
1	0,382	0,3197

Forward stepping with backlash, experiment 2

rh_ya	rh_ro	rh_pi	rk_pi	ra_pi	ra_ro	lh_ya	lh_ro	lh_pi	lk_pi	la_pi	la_ro
NaN	7,1226	1,5374	2,4306	4,8343	4,4023	NaN	8,9693	1,7195	2,5962	4,1851	5,4766
NaN	22,3939	6,6266	7,4856	12,687	18,1372	NaN	16,708	7,7587	7,5236	12,9385	20,3441
0,0012	0,3954	0,9947	0,9998	0,3756	0,5054	0,0012	0,395	0,9947	1	0,3756	0,4998

roll	pitch	yaw
15,6726	18,3731	32,9768
43,5318	68,5694	80,5134
1	0,3822	0,1385

Sidestepping

rh_ya	rh_ro	rh_pi	rk_pi	ra_pi	ra_ro	lh_ya	lh_ro	lh_pi	lk_pi	la_pi	la_ro
NaN	4,7832	7,7858	4,0447	7,3495	1,7413	NaN	4,7899	4,3278	5,1296	5,0873	2,0687
NaN	8,9704	15,9072	10,4188	16,8621	6,0173	NaN	9,0082	15,0293	9,74	11,6871	6,1129
0,0011	1	0,2849	0,6196	0,3347	0,9623	0,0084	1	0,3989	0,8509	0,452	0,9621

roll	pitch	yaw
38,2023	21,3139	64,2042
80,9836	84,3627	97,5685
0,7508	0,6987	1

Side stepping with backlash

rh_ya	rh_ro	rh_pi	rk_pi	ra_pi	ra_ro	lh_ya	lh_ro	lh_pi	lk_pi	la_pi	la_ro
NaN	4,8308	7,9498	4,0784	7,4277	1,678	NaN	3,9973	4,4114	5,1245	5,0547	1,978
NaN	8,5947	16,4685	10,4651	16,7833	5,9661	NaN	8,7753	14,9253	9,7063	12,3064	6,0476
0,001	1	0,2849	0,6196	0,3347	0,9623	0,0084	1	0,3989	0,8509	0,452	0,9621

roll	pitch	yaw
33,8713	21,4211	63,317
69,8385	84,6671	95,6302
0,7508	0,6955	1

Side stepping with friction 400

rh_ya	rh_ro	rh_pi	rk_pi	ra_pi	ra_ro	lh_ya	lh_ro	lh_pi	lk_pi	l
NaN	3,2572	6,3215	4,5943	7,3788	1,3996	NaN	4,735	5,1796	5,6758	
NaN	6,7447	17,6686	11,7317	18,6742	3,4151	NaN	8,94	14,1319	9,9096	
0,001	1	0,2853	0,6205	0,3352	0,9622	0,0079	0,9828	0,3995	0,8521	

roll	pitch	yaw
31,9665	24,9996	51,0981
72,2623	63,1082	88,9275
0,783	0,5959	1

Point turn double step with backlash, experiment 1

rh_ya	rh_ro	rh_pi	rk_pi	ra_pi	ra_ro	lh_ya	lh_ro	lh_pi	lk_pi	la_pi	la_ro
9,8837	4,0502	4,7134	4,7489	5,0252	2,5885	9,0445	4,6377	5,5665	4,8965	10,0875	1,7682
60,8801	11,4	13,0003	8,8508	15,0269	5,626	48,208	10,9734	12,7424	11,9734	24,9591	6,5962
0,6366	0,9993	0,5442	0,9246	0,4533	0,9993	0,6358	1	0,5619	0,7051	0,3197	1

roll	pitch	yaw
27,6487	15,557	92,359
74,963	54,8118	102,1998
0,0152	0,0164	1

Point turn double step with backlash, experiment 2

rh_ya	rh_ro	rh_pi	rk_pi	ra_pi	ra_ro	lh_ya	lh_ro	lh_pi	lk_pi	la_pi	la_ro
10,8096	3,9248	4,5968	5,192	5,0633	2,6469	10,0131	4,49	6,1444	4,8402	10,2611	2,0512
64,9423	9,846	11,9662	15,187	15,1332	6,0728	52,4173	9,0444	12,5009	15,5227	26,353	8,3683
0,6366	0,9993	0,5442	0,9246	0,4533	0,9993	0,6358	1	0,5619	0,7051	0,3197	1

roll	pitch	yaw
28,346	16,8812	92,4591
71,5283	56,8218	102,1706
0,0156	0,0164	1

Point turn double step, experiment 1

rh_ya	rh_ro	rh_pi	rk_pi	ra_pi	ra_ro	lh_ya	lh_ro	lh_pi	lk_pi	la_pi	la_ro
9,5657	3,9284	4,5502	4,6976	4,7656	1,7981	8,6888	5,7777	5,2747	4,6385	10,1693	1,9205
59,3167	10,8211	12,7842	11,1962	13,1369	5,358	46,5803	10,4817	12,7379	10,8386	28,5124	6,0489
0,6366	0,9993	0,5442	0,9246	0,4533	0,9993	0,6358	1	0,5619	0,7051	0,3197	1

roll	pitch	yaw
26,9709	17,6712	92,3857
67,6213	73,6163	102,1873
0,0152	0,0164	1

Point turn double step, experiment 2

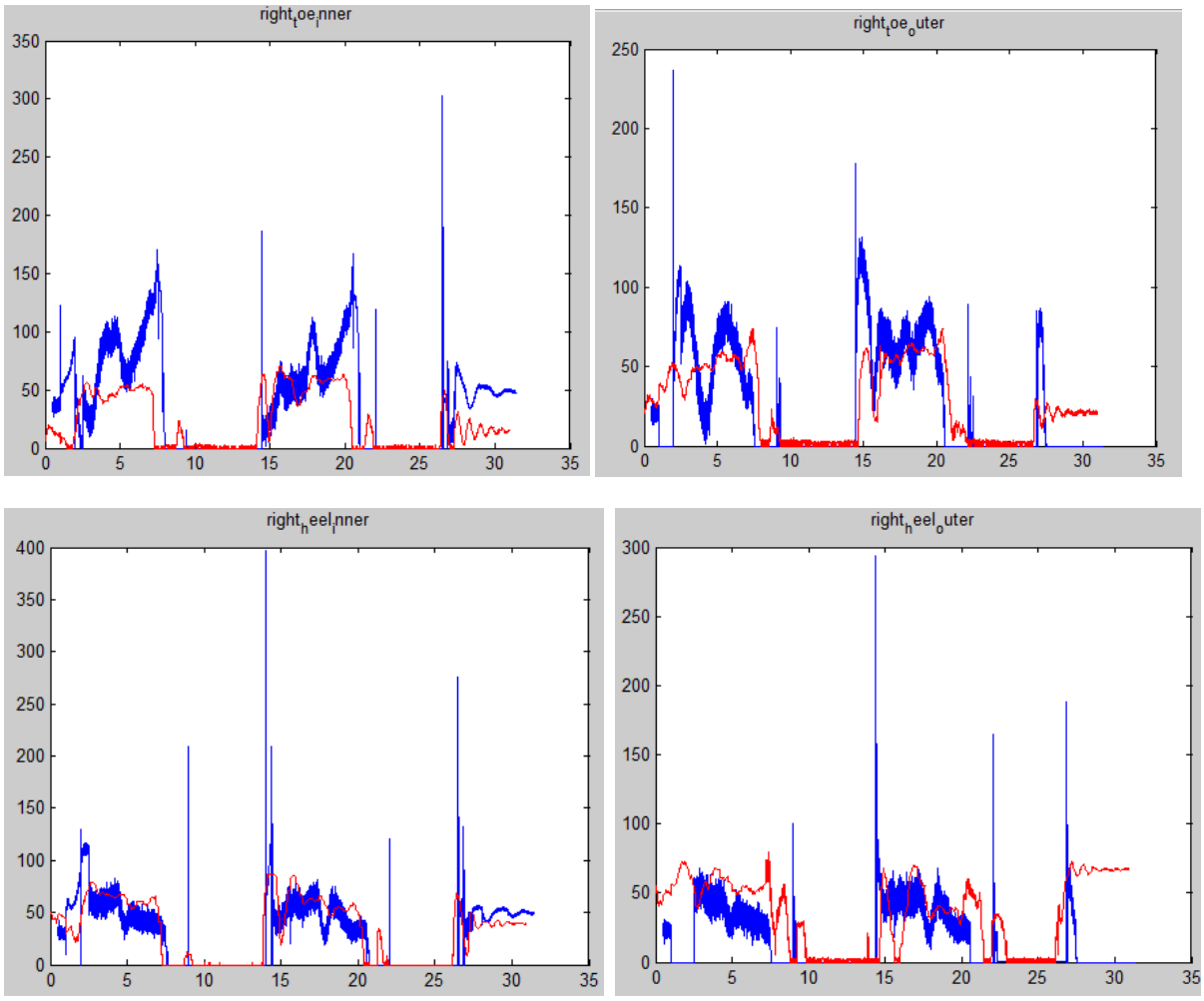
rh_ya	rh_ro	rh_pi	rk_pi	ra_pi	ra_ro	lh_ya	lh_ro	lh_pi	lk_pi	la_pi	la_ro
10,4945	3,7284	4,3893	5,1291	4,7675	2,0313	9,6838	5,682	5,8261	4,5902	10,3324	2,2441
63,5135	9,7999	11,8755	14,8924	13,0231	5,3598	50,8654	9,5999	12,4732	15,3046	29,8156	7,6904
0,6366	0,9993	0,5442	0,9246	0,4533	0,9993	0,6358	1	0,5619	0,7051	0,3197	1

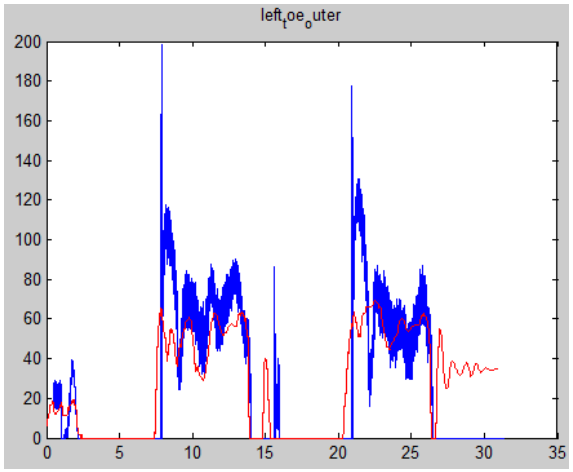
roll	pitch	yaw
28,0009	18,8026	92,4863
69,6722	75,2847	102,2318
0,0156	0,0164	1

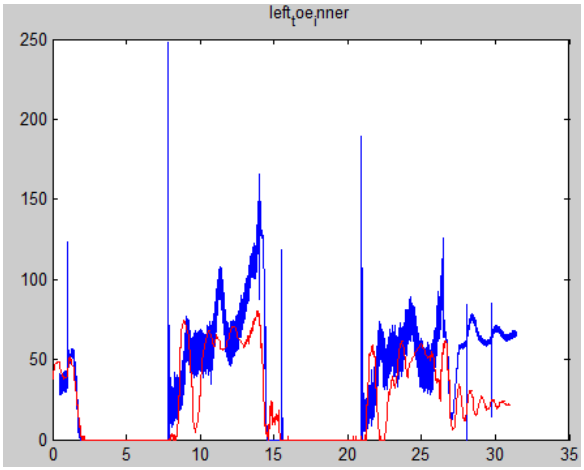
APPENDIX C

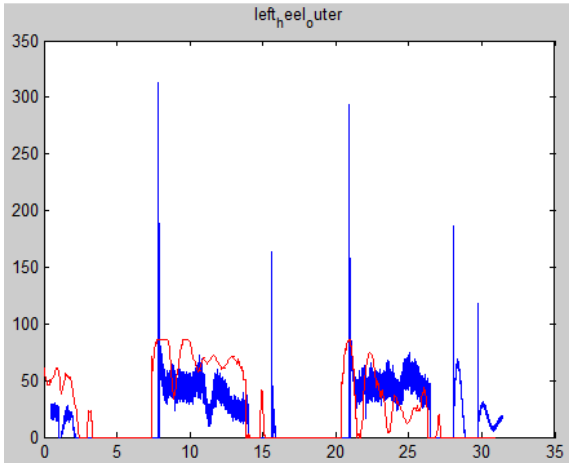
In this appendix, the foot sensor data of the three experiments are shown. The scaling is done approximately. The blue line depicts the simulation, the red line the measurement.

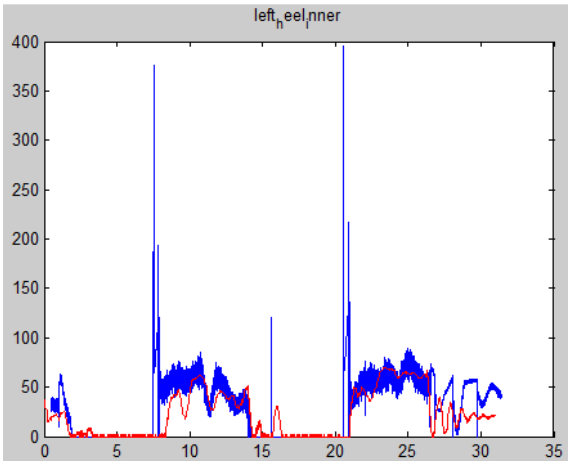
Forward stepping



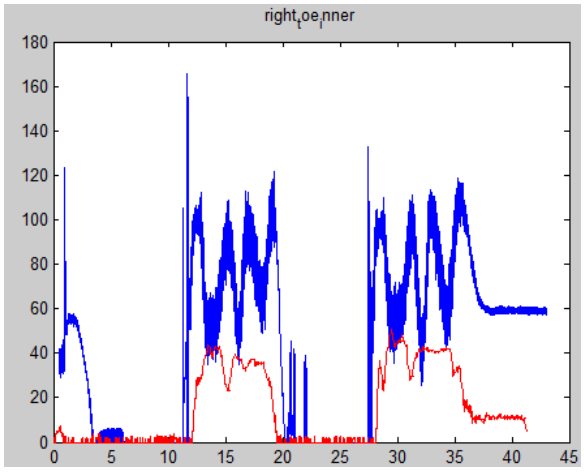


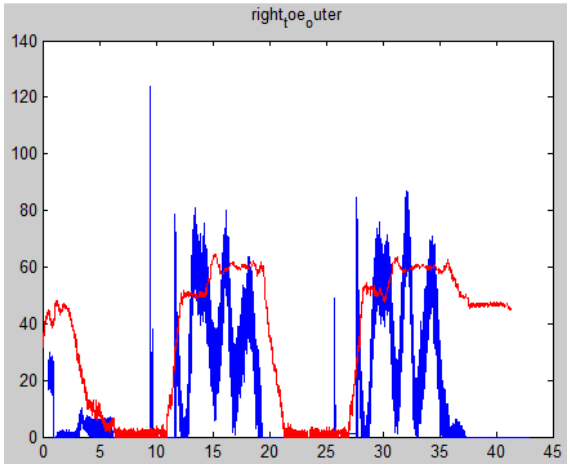


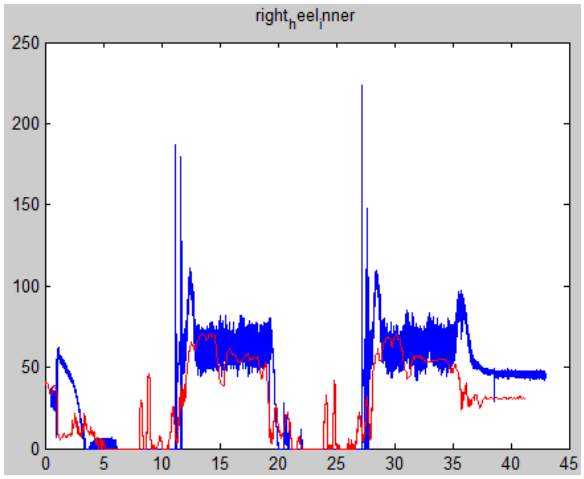


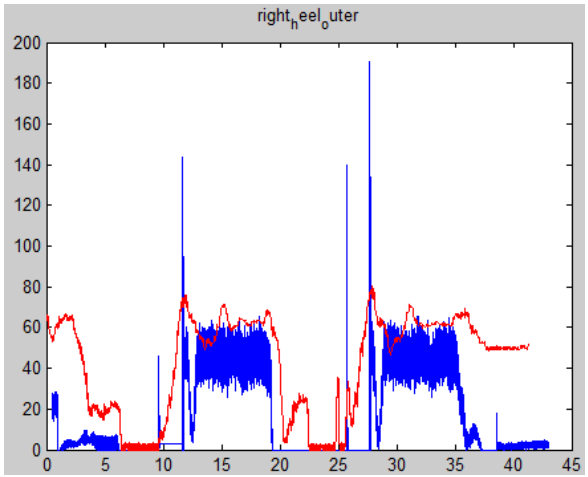


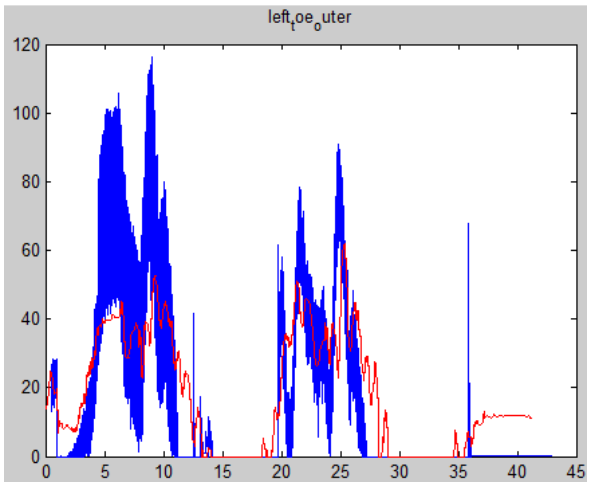
Sideward stepping

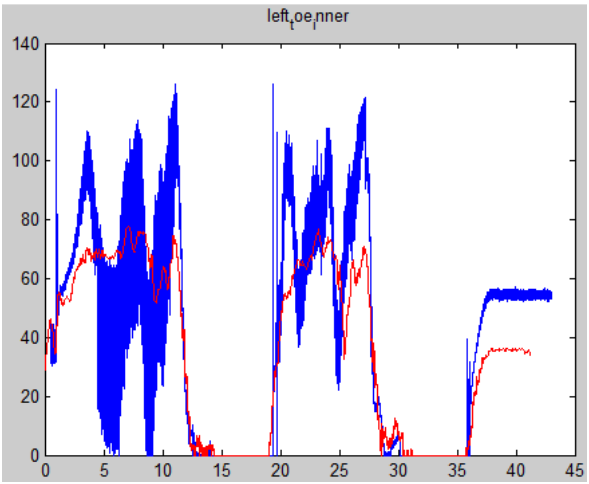


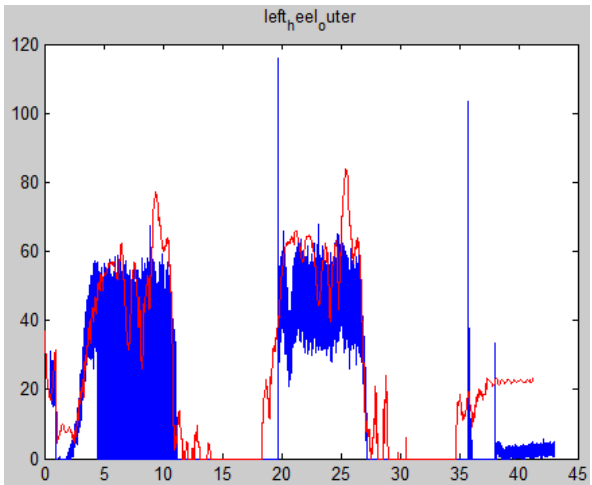


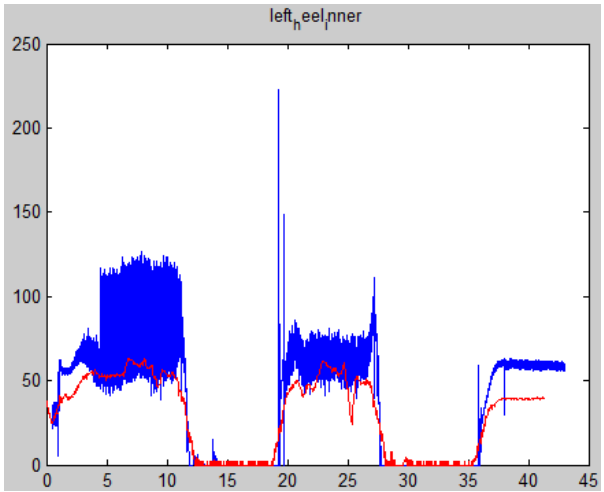

right_heel_inner


right_heel_outer

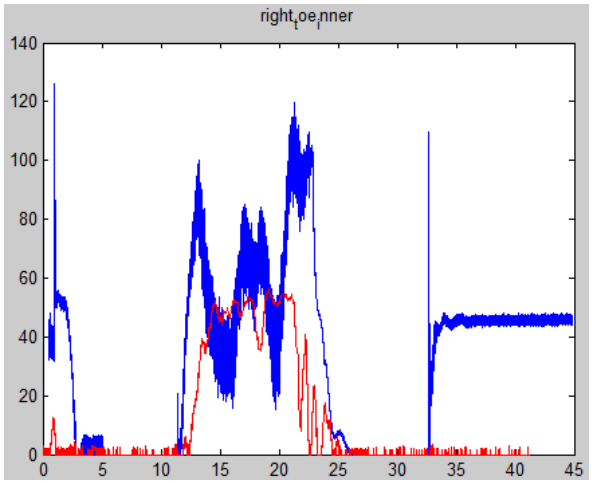

left_toe_outer

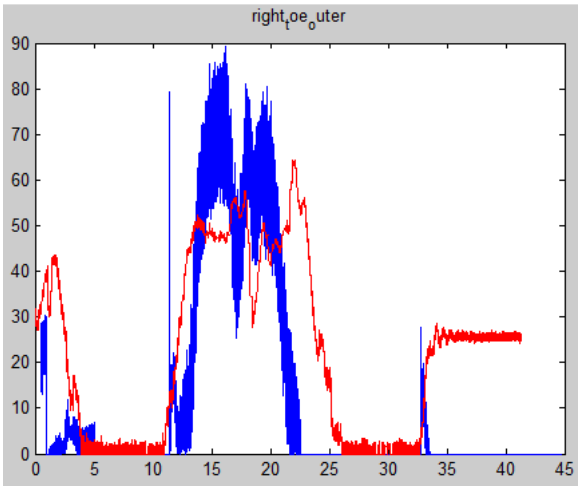

left_toe_inner

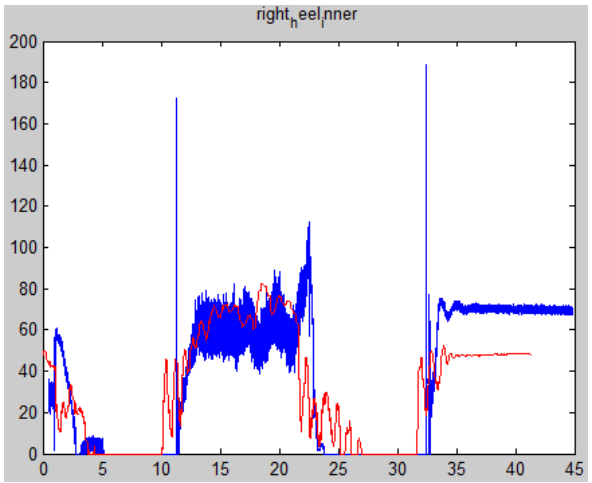

left_heel_outer

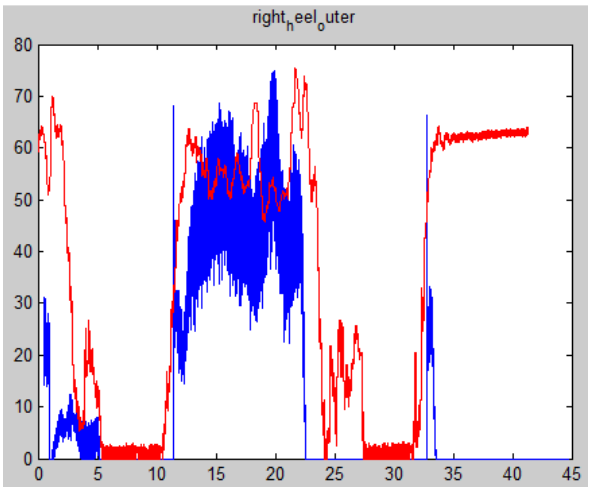

left_heel_inner

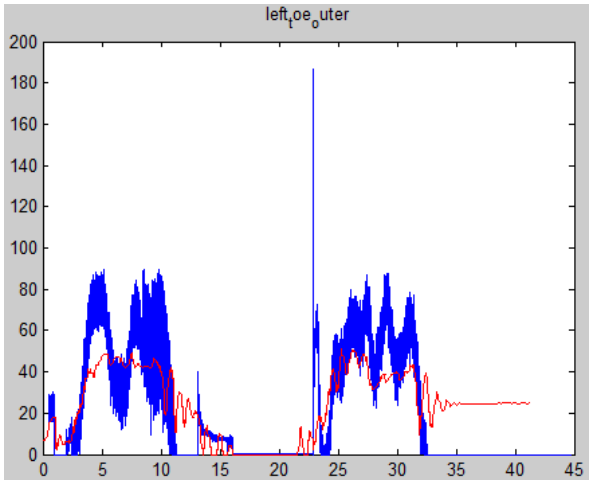
Point turning

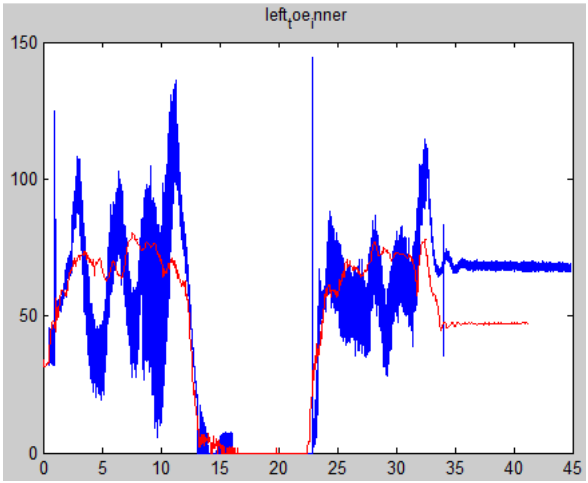




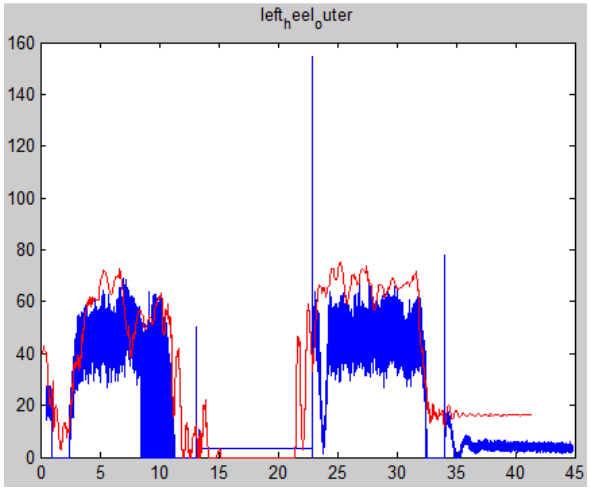


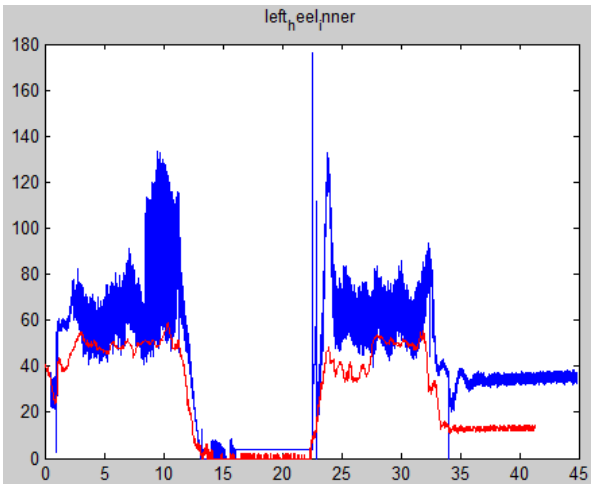






X


left_heel_outer
160
140
120
100
80
60
40
20
0
0
5
10
15
20
25
30
35
40
45


left_heel_inner
180
160
140
120
100
80
60
40
20
0
0
5
10
15
20
25
30
35
40
45