

Metrics for Bipedal Walking of Humanoid Robots in RoboCup Standard Platform League

Project Thesis

submitted by
Maik Marius Rebaum

Institute of Embedded Systems
Hamburg University of Technology

April 2024

1st Examiner: Prof. Dr. Goerschwin Fey
Supervisor: Maximilian Schmidt, M. Sc.

Abstract

The RoboCup is a robotics focused research initiative with several leagues focused on different competitive fields. One of these is the Standard Platform League[1] in which teams compete in soccer matches using multiple fully autonomous robots. The NAO is the humanoid robot platform used within the Standard Platform League, and has been used to bring about various advances in the field of humanoid walking, including omnidirectional walking[2] or planning on-the-fly step adjustments[3]. Walking, amongst other environmental interactions, is still one of the most difficult aspects of humanoid robotics[4][5]. It is also a highly critical component of the Standard Platform League competition, and the RoboCup provides the perfect environment to procure and test research advances.

Finding optimal parameters or testing modified walking engine software, often requires hours of manual labor, and improvements in walking quality are judged through pure guesswork. It is thus vital to find metrics that can be used to quantitatively judge the quality of walking without requiring human estimations of what is better or worse. This thesis discusses the application of six main categories of walking quality in robotics research:

- Robustness
- Speed
- Energy consumption
- Heat production
- Slipping
- Double support time

To gather data on this, while adhering to limitations of the RoboCup Standard Platform League (SPL), software is developed which accesses the raw sensor data of the NAO. This allows for the comparison of the B-Human, Nao Devils, HULKs and FastHULKs walking engines, the last of which are identical in software but differ in the parameters used. Metrics based on torso stability, energy consumption, heat production and number of falls of the robot are defined and applied to the collected data. It is found that although the Nao Devils maintain the highest torso stability while consuming the least energy and producing the least amount of heat, it is the team which also has the most number of falls. The comparison of HULKs, FastHULKs, and B-Human shows no absolute ordering, indicating that they may lie on different points of a Pareto-optimal scale of walking quality. The limited number of metrics defined in this thesis are not enough to gain a comprehensive objective understanding of what exemplifies high quality walking and finding a Pareto-optimal walking quality requires further research.

Contents

List of Figures	viii
List of Tables	ix
Abbreviations	xi
1 Introduction	1
1.1 Problem Statement	1
1.2 Sketch of Contribution	2
1.3 Structure of Thesis	3
1.4 Usage of Electronic Tools	3
2 Preliminaries	5
2.1 Basic Notation	5
2.2 Unix Sockets	5
2.3 Hardware Description of the NAO Robot	5
2.4 NAO Software Basics	8
2.5 Bipedal Walking	9
2.6 HULKs Software	11
2.6.1 HULKs Walking Engine	11
2.6.2 Walking Engine Black-Box	13
2.7 RoboCup Standard Platform League Competition	13
3 State of the Art	15
3.1 Robustness	15
3.1.1 Static and Dynamic Stability	15
3.1.2 Camera Stability	17
3.1.3 External Forces	17
3.1.4 Terrain	18
3.1.5 Hardware	18
3.2 Speed	18
3.3 Energy Consumption	19
3.4 Heat Production	19
3.5 Slipping	20
3.6 Double Support Time	20
3.7 Applicability	21
3.7.1 RoboCup SPL Context	21
3.7.2 Black-box Evaluation	22
3.7.3 Addressing Ambiguity	22

4 Method	23
4.1 Data Gathering Method	23
4.1.1 Delays in Data	24
4.1.2 Data Conversion from MessagePack to CSV	25
4.2 Experimental Setup	25
4.2.1 Testing Environment	26
4.2.2 Testing Scenarios	26
5 Data Presentation and Post Processing	29
5.1 Collected Data	29
5.2 Coordinate Systems in this Thesis	30
5.3 Ground Contact Estimation	31
5.4 Support Foot Estimation	32
5.4.1 Force Sensitive Resistor (FSR)	32
5.4.2 Kinematics	32
5.4.3 Calculation of z_d	34
5.5 Center of Mass (CoM), and Zero Moment Point (ZMP)	34
5.6 Support Polygon Calculation	35
5.7 Feature Extraction	36
5.7.1 Cropping the Start	38
5.7.2 Removing of Falls	39
5.8 Issue In Data Gathering	40
6 Data Analysis	41
6.1 Falling	41
6.2 Center of Mass (CoM)	42
6.3 Zero Moment Point (ZMP)	44
6.4 Double Support Phase Length	45
6.5 Energy Consumption	46
6.6 Heat Production	48
6.7 Torso Stability	50
6.8 Summary	52
7 Conclusion	55
Bibliography	61
Declaration	63

List of Figures

2.1	NAO Joints Overview[13]	6
2.2	Joint Angles of Left Leg and Head[13]	6
2.3	NAO Dimensions Overview[13]	7
2.4	NAO Feet with FSR Sensors[13]	7
2.5	NAO Inertial Measurement Unit (IMU)[13]	8
2.6	NAO Software as based on Softbank Robotics Documentation[13]	9
2.7	Bipedal walking cycle	9
2.8	Single support phase inverted pendulum	10
2.9	Possible Step Splines	11
2.10	HULKs Walking Engine	12
3.1	Zero Moment Point (ZMP)	16
4.1	Hulahoop execution steps	24
4.2	JSON excerpt of one frame	25
4.3	Field Setup for Data Gathering Purposes	27
5.1	Robot coordinate system	30
5.2	Angled support foot	33
5.3	Convex hull of left sole in <i>Lsole</i> Coordinates	33
5.4	Support foot using two methods from HULKs-NAO4	34
5.5	Example of the support of HULKs-NAO4 in one frame	36
5.6	Knee- and Hip-pitches of B-Human-NAO1	37
5.7	Sections of NaoDevils-NAO3 to be removed	37
5.8	Irrelevant data from FastHULKs-NAO3 before walking start	38
5.9	Section from FastHULKs-NAO3 where robot fell	39
6.1	Comparison of CoM of the walking engines on NAO1	42
6.2	Heatmap of CoM of the walking engines on NAO1	43
6.3	Comparison of CoM of the HULKs walking engine on NAO1-4	43
6.4	Number of frames (W) where the ZMP was outside support polygon of both feet	45
6.5	Average Double Support Phase Length	46
6.6	Metric 2: Average battery current	47
6.7	Metric 3: Peak battery current	47
6.8	Temperatures of motors from BHUMAN-NAO1	48
6.9	Temperatures of motors from FastHULKs-NAO3	49
6.10	Metric 4: Average temperature gain in $^{\circ}\text{C}$ / minute of joints in Ω	49
6.11	Metric 5: Average absolute gyroscope x value	51
6.12	Metric 6: Average absolute accelerometer y value compensated for gravity	52
6.13	Metric 7: Average absolute accelerometer z value compensated for gravity	52

List of Tables

2.1	Basic notation	5
5.1	An excerpt of B-Human-NAO1 CSV with abbreviated headers	29
5.2	Sensor Values	30
5.3	Hysteresis Estimate	32
6.1	Metric 1: Number of falls, *based on less than 10 minutes of data	41
6.2	Number of frames (W) where the ZMP was outside support polygon of both feet	44
6.3	Average Double Support Phase Length	46
6.4	Metric 2: Average battery current	47
6.5	Metric 3: Peak battery current	47
6.6	Metric 4: Average temperature gain in $^{\circ}\text{C}$ / minute of joints in Ω	49
6.7	Maximum temperature in $^{\circ}\text{C}$ of the hottest joint in Ω	50
6.8	Metric 5: Average absolute gyroscope x value	51
6.9	Metric 6: Average absolute accelerometer y value compensated for gravity	51
6.10	Metric 7: Average absolute accelerometer z value compensated for gravity	52

Abbreviations

CoM Center of Mass

DOF Degrees of Freedom

FSR Force Sensitive Resistor

FZMP Fictitious Zero Moment Point

HAL Hardware Abstraction Layer

HULA HULKs Level Abstraction

IMU Inertial Measurement Unit

LoLA Low Level Abstraction

SPL Standard Platform League

URDF Universal Robotic Description Format

ZMP Zero Moment Point

1 Introduction

Bipedal humanoid robots are a growing field in the domain of robotics, as their compact form factor and flexibility in movement allows them to work side by side with humans[6][7]. The RoboCup Standard Platform League (SPL) intends to advance humanoid robotics through the coordination and administration of competitive autonomous soccer matches. SPL means that all teams compete using the same robotics hardware, the NAO. As the hardware model is identical between teams, it is necessary to develop highly performant robotics code in order to gain a competitive advantage in the RoboCup.

There are various aspects to robotics code in the context of the SPL, including computer vision, behavioral strategy, and inter-agent communication. One of the main aspects is walking, the quality of which determines a team's success in the RoboCup. Walking quality is determined through various evaluation categories, such as stability, energy consumption, heat production, and speed. Robots that are able to walk faster across a field, while remaining stable, are able to get to the ball quicker than the other team, and in turn score more goals. To effectively and efficiently improve the quality of walking, teams use metrics to judge the quality of the existing walking software and its parameters. Upon making a change to their software and parameters, teams evaluate their metrics and are able to identify whether the changes made have improved their walking quality.

This thesis is written in the context of the *HULKs*[8] SPL team. For the HULKs to make effective changes to their software and parameters, it is important to understand how the walking quality of their robots compares to that of other teams. Additionally, evaluating a set of metrics solely for the HULKs, only sparsely demonstrates the applicability of these metrics in the more general context of the RoboCup and humanoid robotics. This thesis therefore compares the HULKs walking to two other SPL teams, namely *B-Human*[9] and the *Nao Devils*[10].

HULKs, B-Human, and the Nao Devils have Glicko ratings[11] of 1601, 2431, and 1663, respectively[12]. Similar to an ELO rating in chess, a higher number in the Glicko system indicates a stronger performance in competitions. B-Human was chosen for this thesis as it is the team with the internationally highest Glicko rating. The Nao Devils are chosen as they have a Glicko rating closest to that of the HULKs, which indicates a similarity of performance. However, these teams are also chosen for comparison and evaluation based on empirical evidence of their walking performance. Experience during RoboCup competitions shows that B-Human walks the most stable of any team, and a robot falling is a rare occasion. The walking of the Nao Devils robots is fast, but often leads to instability and robots regularly falling during a game.

1.1 Problem Statement

The HULKs have no metrics by which to analyze the quality of walking that their software and parameters produce. Software changes and calibration are implemented based on subjective evaluation. While tuning parameters, stability is judged by visual inspection, and temperature by how hot the robot's joints feel to the touch. These are subjective and depend on the previous experience of the person performing the adjustments. Other aspects of walking quality, like

energy consumption, are not even considered. This individualized tuning of robots often takes hours and wastes valuable time before a competition match. Additionally, as stated above, the comparison to other teams is also only based on visual comparison.

To this end there are four key challenges this thesis overcomes:

1. To the author's knowledge, no prior work exists which analyzes multiple facets of walking quality for the context of the NAO and the RoboCup. The HULKs thus have no all-encompassing understanding of how robustness, speed, heat production, energy consumption, slipping, and double support time, are used in research of bipedal robots.
2. There is a lack of understanding of how parameters are interconnected with each other and with the hardware of the robot. For example, increasing the forward acceleration of a robot from standstill, may lead to the robot tilting backwards too much, and falling over. Under this circumstance, to compensate for the acceleration, it is necessary to make adjustments which make the robot tilt more forward. The correlation of parameters, which reflect this particular scenario and others like it, is unknown. It also depends on the hardware for which parameter tuning is performed. Some robots have joint gears that are more worn out and therefore less precise. The amount the robot should tilt, so that it can compensate for the whiplash created by acceleration during walking, may differ depending on the deterioration of the hardware. Evaluating the walking quality through metrics makes it possible to quantify the effect a parameter change has on each robot.
3. Parameter tuning usually takes place a few hours or even minutes before a match, where individual team members select whatever field space is available to test each robot. In this context human judgment is currently the quickest way to transform information from visual inspection of the robot's walking into parameter changes. Time and space constraints do not allow for the setup of external measuring tools, such that tuning is done with only a robot and laptop in hand.
4. Comparison of the walking quality of different teams is difficult. Each team uses their own software, which in some cases is entirely unique from other teams, including both structure and programming language. Human analysis of the code does not provide sufficient metrics to judge walking quality by. The difference in software also presents a challenge as the format in which each team processes data is individualized and does not provide easily comparable data. Attempting to modify B-Human's and Nao Devils' software, such that data about each team's walking is structured the same way, is out of the scope of this thesis. The software and parameters of the other teams must essentially only be treated as a black-box for testing purposes.

1.2 Sketch of Contribution

The first part of this thesis delves into state-of-the-art walking analysis categories and metrics used in current research and explores their applicability to the NAO and the challenges in this thesis. To collect and analyze data, a non-intrusive sensor data collection software is developed, along with the groundwork for a data processing pipeline. Qualitative analysis is then performed to examine the walking quality difference exhibited by HULKs, B-Human, and Nao Devils. Metrics are defined and evaluated to provide a quantitative evaluation of the variations among the walking engines employed by these teams.

1.3 Structure of Thesis

Chapter 2 explains preliminary information relevant to understanding the context of this thesis. Chapter 3 discusses existing state-of-the-art walking quality analysis of legged robots and discusses how these are applied to the NAO in scope of this thesis. Chapter 4 explains the chosen method of data gathering as well as the testing scenarios used to gather data. Chapter 5 provides an overview of the collected data and provides calculations for secondary data that are calculated from the collected data. Chapter 6 compares the walking quality of the HULKs, B-Human, and the Nao Devils both qualitatively and quantitatively using metrics. Chapter 7 provides a conclusion for the discovered results and discusses future work in this topic.

1.4 Usage of Electronic Tools

This thesis was written in *LaTex*. *ChatGPT3.5* was employed in this thesis as a search engine to provide the correct syntax of LaTex formatting commands. The text, images, tables, and software used in this thesis contain no usage of artificial intelligence.

2 Preliminaries

This chapter contains necessary preliminaries in understanding the foundational concepts on which this thesis is built. First is a summary of basic notations used in the rest of this thesis, as well as a brief description of Unix sockets. Then follows a discussion of what is needed to make a NAO robot walk. It is first necessary to discuss the hardware and underlying software of the NAO as it is provided by *Aldebaran*[13]. From this the process of humanoid walking is explained along with a description of the walking engine the HULKs team uses to accomplish this. This chapter concludes with a brief summary of the competitions in the RoboCup SPL,

2.1 Basic Notation

All variables and notations used are defined within context. Table 2.1 contains the most important notation that is used throughout the thesis.

Symbol	Meaning
g	Gravity constant, $9.807m/s^2$
k	Frame Number
K	Total number of frames
L,R	Subscripts referring to the left or right foot
$\mathcal{R},\mathcal{P},\mathcal{G}$	Coordinate Frames
P	A 2D/3D Point
T_O^U	Homogeneous transformation matrix from O to U
t	Time
X, Y, Z	Axes
x,y,z	Coordinates within the axes
$\ddot{x},\ddot{y},\ddot{z}$	Acceleration along the axes

Table 2.1: Basic notation

2.2 Unix Sockets

Unix sockets are a standard component of POSIX operating systems[14]. They provide inter-process communication on the same host operating system. One type of Unix socket is a stream socket. A stream socket provides a sequenced data stream between two applications. A file descriptor is used to name the socket, which needs to be identical for the socket host and user when initializing the connection.

2.3 Hardware Description of the NAO Robot

The NAO, as used in this thesis, refers to the sixth generation of humanoid robot developed by Aldebaran, and serves primarily as an education and research tool in the field of humanoid

robotics[13]. It features 25 Degrees of Freedom (DOF) provided by 5 different types of motors that make up its various joints (see Fig. 2.1). Figure 2.1 also includes the labeling for the rotations of each motor such that changes in roll, pitch, and yaw are rotations about the X, Y, and Z axes respectively of each joint. This means for example that a rotation of the leg forward using the hip pitch motor is a negative rotation about the Y axis (see Fig 2.2).

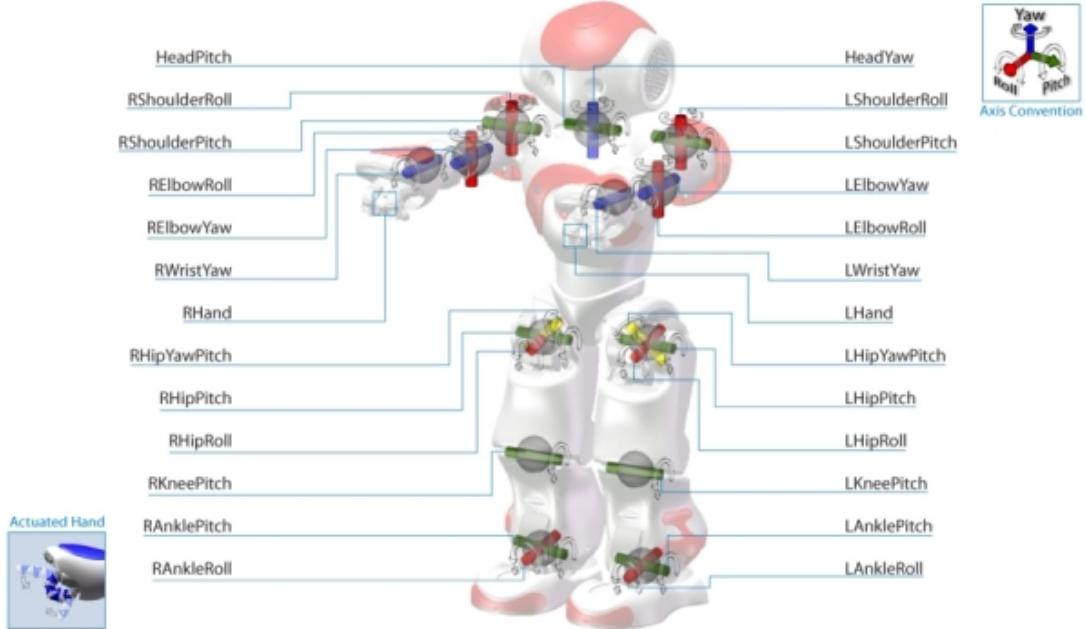


Figure 2.1: NAO Joints Overview[13]

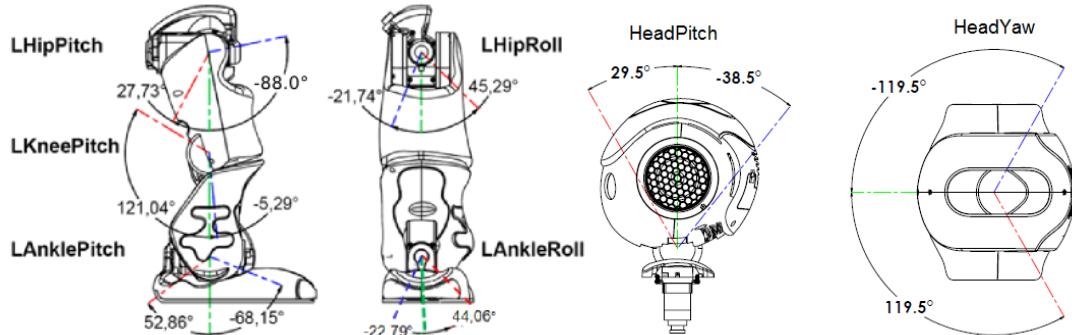


Figure 2.2: Joint Angles of Left Leg and Head[13]

Aldebaran also provides a Universal Robotic Description Format (URDF)[15] file for simulations, which contains masses and dimensions of the limbs of the NAO. The dimensions are shown in Figure 2.3. The masses and dimensions are required to perform accurate kinematics calculations of the robot.

The NAO contains several sensors. First, there are 8 FSRs located at the bottom of the feet which provide pressure information in order to determine ground contact (Fig. 2.4). Second is the IMU made up of a 3-axis gyroscope and a 3-axis accelerometer. This sensor sits central to the robots' torso (see Fig. 2.5). Besides the raw data from the gyroscope and accelerometer, the IMU also provides a filtered ‘AngleX’ and ‘AngleY’ which determines the radians by which

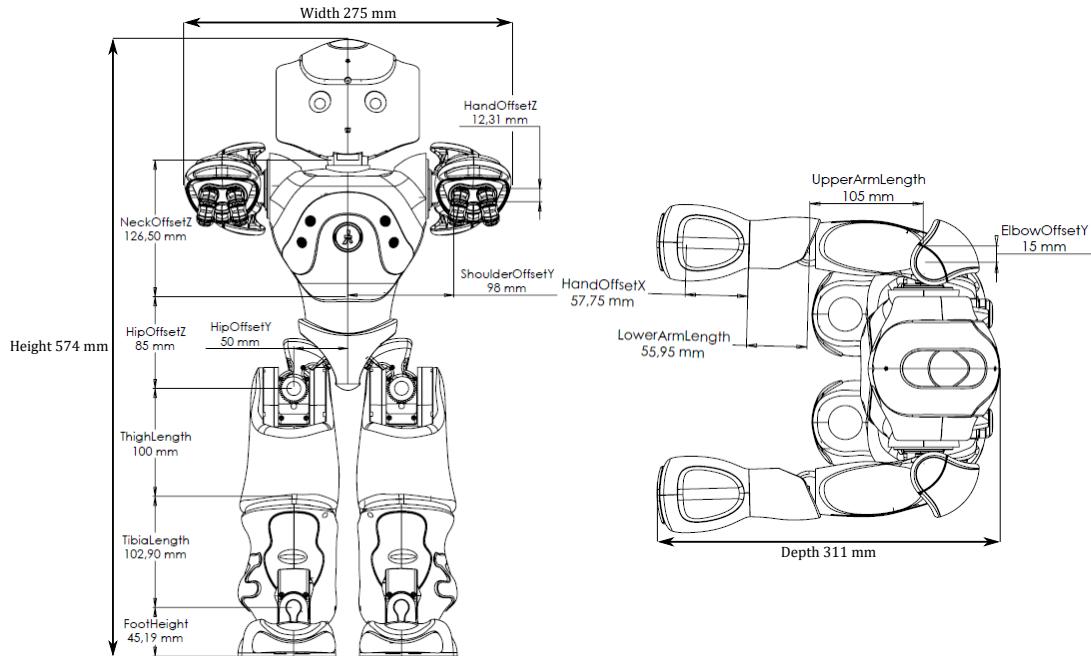


Figure 2.3: NAO Dimensions Overview[13]

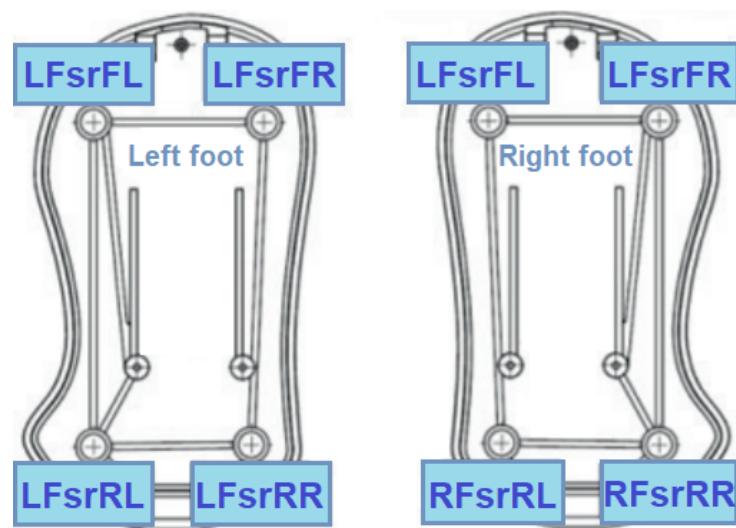


Figure 2.4: NAO Feet with FSR Sensors[13]

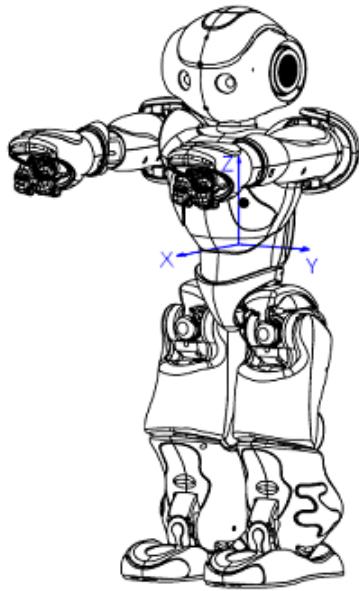


Figure 2.5: NAO Inertial Measurement Unit (IMU)[13]

the torso is tilted left/right or forward/backward from its upright position in which the Z-axis opposes the direction of gravity. Each of the 25 motors contains joint encoders to read out the angle of the joints with 0.1° accuracy. Finally, there are sensors to provide information regarding the electrical current of all 25 motors and the battery.

2.4 NAO Software Basics

As introduced in Chapter 1 the SPL is a RoboCup league where the teams play with identical hardware, but each team develops their own software usually referred to as robotics code. Robotics code is the software that is used to take sensor and camera data and calculate motor control commands.

The NAO runs two underlying processes which enable the programmed robotics code to control the robot (see Fig. 2.6). At the lowest level is the Hardware Abstraction Layer (HAL) which manages the communication with the electronics boards of the robotics hardware. Connected to HAL is the Low Level Abstraction (LoLA) process. LoLA and HAL exchange data every 12ms which in the direction of HAL to LoLA consists of sensor values and in the reverse is the motor commands. In order for robotics code to read sensor data and give control commands, LoLA provides a Unix stream socket that the robotics code can connect to.

Information is sent in both directions in the form of *MessagePacks*[16]. MessagePack is described as an ‘efficient binary serialization format’ [16] which allows for the rapid data exchange between multiple applications. It is similar to JSON, which stores data in a *key* and *value* pairing such that values can be accessed and written using the key. MessagePack also uses strings as keys, but it encodes the values to significantly reduce the total object size.

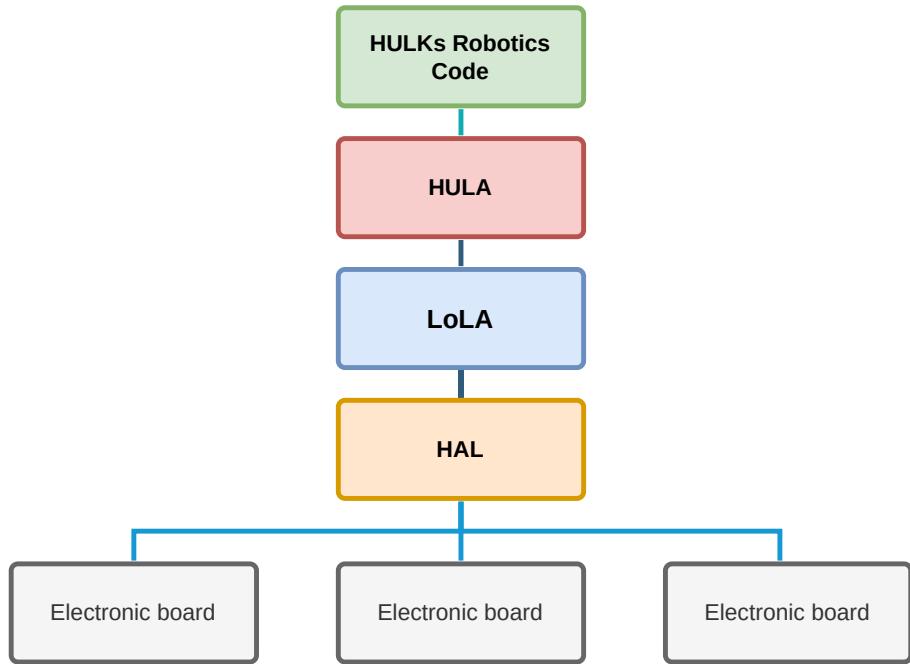


Figure 2.6: NAO Software as based on Softbank Robotics Documentation[13]

In the case of the HULKs team, the connection to LoLA is made by the HULKs Level Abstraction (HULA). HULA converts the MessagePack data into structs usable by the rest of the HULKs robotics code. An overview of this is shown in Figure 2.6.

2.5 Bipedal Walking

A walking cycle of a bipedal robot refers to the complete sequence of movement that encompasses both feet being lifted and placed once. When a foot is in contact with the ground and supports the weight of the robot it is considered to be the *support foot*. The respective other foot which is moving relative to the ground is considered the *swing foot*. Figure 2.7 shows one walk cycle, in which it is assumed that the robot is walking in a straight line. One complete cycle of a

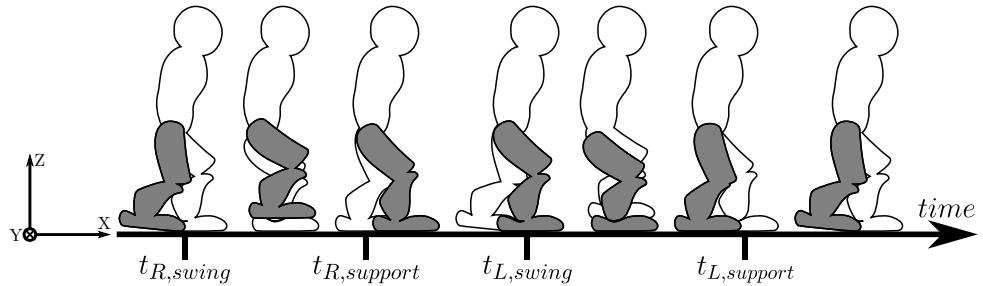


Figure 2.7: Bipedal walking cycle

bipedal walk can be partitioned into two steps, usually symmetric for the left and right foot[4]. The timestamps for the right foot are marked in Figure 2.7. It starts at $t_{R,swing}$, the instant at which the right foot leaves the ground and becomes the swing foot. The robot then proceeds through the single support phase as the right foot swings to be in front of the left foot where

it is placed back on the ground at time $t_{R,support}$. This is followed by the double support phase during which both feet are in contact with the ground and the robot continues to move forward while shifting its weight from the left foot onto the right foot which becomes the support foot. The step using the right foot finishes as the left foot leaves the ground and becomes the swing foot at time $t_{L,swing}$.

For the purposes of understanding how the NAO is enabled to walk it is considered as an open kinematic chain. This means the limbs of the robot are successively linked to each other in a chain with the terminal limbs are not attached to a fixed point. Although the arm may also be used for walking, the main consideration is the torso along with the two leg sub-chains. The point that represents the location of the average mass of all links in the kinematics chain is known as the Center of Mass (CoM) and is calculated by:

$$P_{CoM} = \frac{\sum^i P_i m_i}{\sum^i m_i} \quad (2.1)$$

where P_i and m_i are respectively the location and mass of each limb i . Forces acting equally on all parts of the kinematic chain can be represented as a single force acting on the CoM.

The single support phase provides a unique challenge to balancing the robot as the whole kinematic chain essentially becomes an inverted pendulum. The CoM of the robot may be in a position that is not over any point of contact of the current support foot (Fig. 2.8). Assuming no acceleration of a robot in this instance leads to it inherently becoming unstable as the force of gravity creates a moment $\tau = \Delta x \cdot F_g$ around the X axis that causes the robot to fall as $F_g = F_N$. Where F_N is the normal force of the ground acting on the robot and F_g is the force of gravity. This necessitates that the current swing foot be placed in manner that allows the robot to catch itself before it falls over. Dynamic walking like this essentially is controlled falling. The

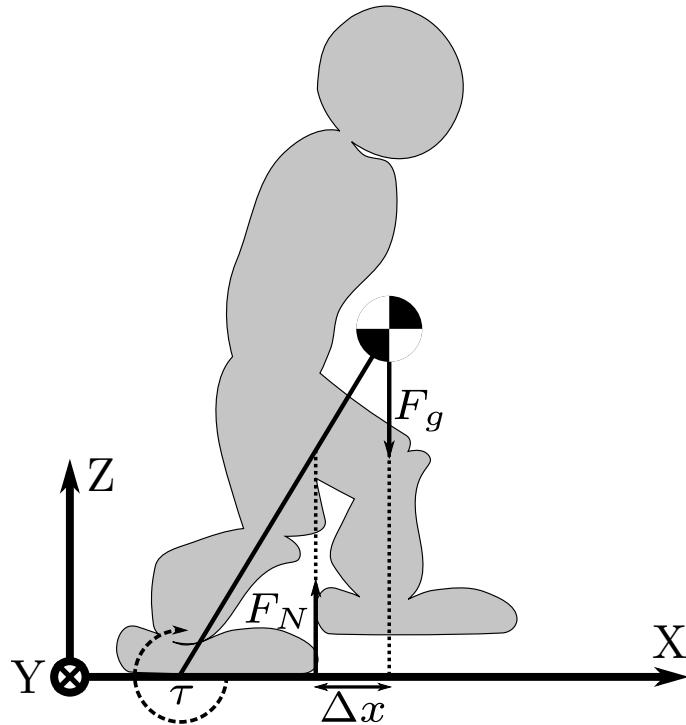


Figure 2.8: Single support phase inverted pendulum

placement and movement of the feet is known as a gait or gait pattern. This includes both the positioning of the feet relative to each other, and the spline shape of the movement of the swing foot. Examples of these splines for the HULKs team are shown in Figure 2.9, for a forward and a sideward step respectively. A forward step follows a symmetric curve and reach its apex at the middle of the step, while the support limb is fully stretched. Meanwhile, a side step has its apex early in its movement, in order to prevent the swing foot from intersecting with the ground as the robot tilts in the direction of the swing foot.



Figure 2.9: Possible Step Splines

2.6 HULKs Software

Software that is successfully applied in a RoboCup soccer match has many components. In Figure 2.6 the sum of these is simply summarized as the “HULKs Robotics Code”. These include capturing images from the camera for the purpose of localization and ball detection, dealing with audio signals such as the whistle at the start of a game, and communicating with other robots in the team. The HULKs team has a software framework that handles all of these robotics code processes. While it is running, this framework receives data from LoLA via HULA at some time t_k and after performing necessary calculations sends motor commands back to LoLA via HULA at some time t_{k+e} where k is the frame number and e is a value between 0 and 1. This means that motor commands are sent back to LoLA before the next set of sensor data is sent by LoLA. The interval between t_k and t_{k+1} is considered a frame and constitutes 12 ms, meaning motor commands are only sent as often as LoLA is able to provide data.

2.6.1 HULKs Walking Engine

The walking engine is the software component which handles the calculation of motor position commands from sensor information. Figure 2.10 shows the schematic representation of the software implementation of concepts introduced in Section 2.5. The goal of the walking engine

is to take sensor data and behavioral targets coming from other components of the HULKs robotics code in order to calculate the necessary motor positions needed to make the robot reach these targets. The data flow following the description in Figure 2.10 for a frame in which the robot walks is as follows:

1. Behavioral goal: The behavior code determines a target location, such as the ball, and a path to reach that location. This is the motion command that enters the walking engine.
2. Decide the best course of action:
 - a) The step planner calculates necessary foot placements needed in order for the robot to follow this path
 - b) At the same time the motion selector determines whether walking is the best course of action for the robot. If the robot is beyond a point of falling where it can save itself it may better to perform a different action to protect the hardware. It also might be the case that it is not safe for the robot to transition into walking from another motion.
3. Calculation of motor positions: The above data cumulates into a walk command, which is used to calculate the motor positions for the robot in the current frame. For this the current motor positions and other sensor data are used along with the walking engine parameters to calculate the target joint positions based on a desired gait pattern.
4. Sending of motor commands: The joint position commands are sent to LoLA via HULA before the end of the frame.

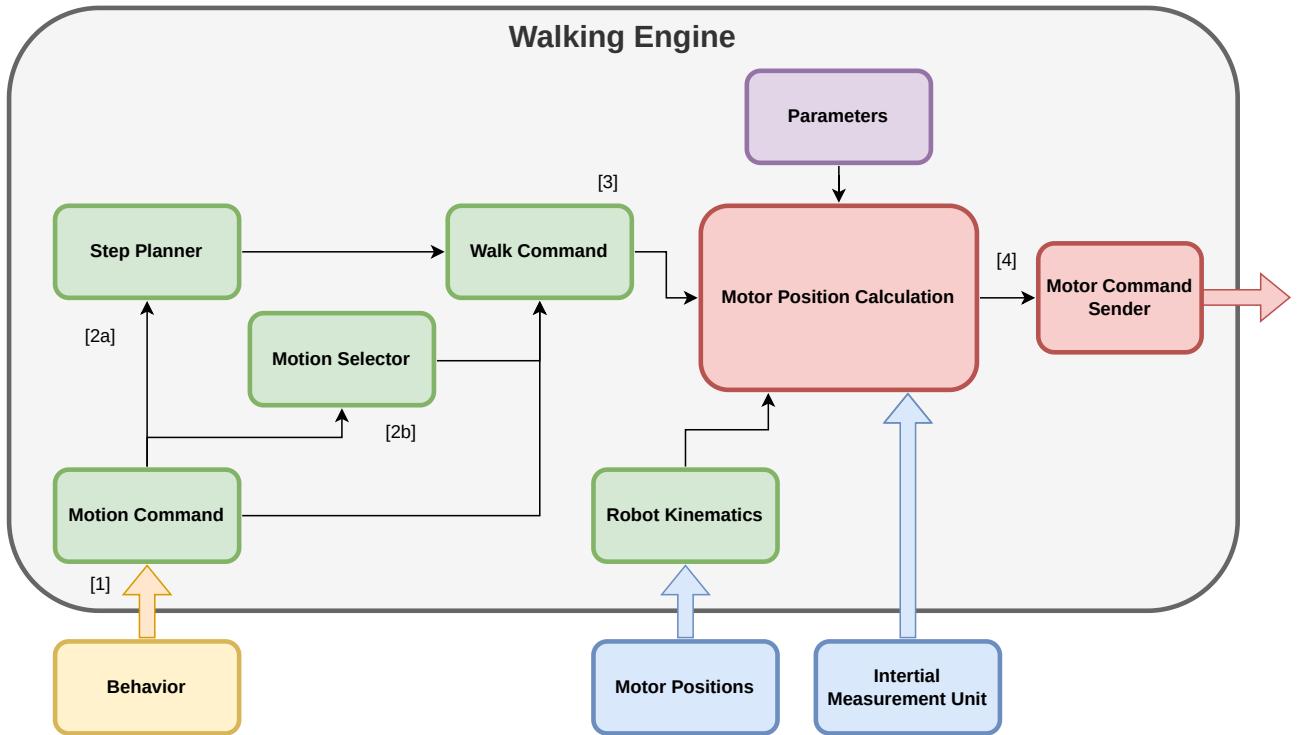


Figure 2.10: HULKs Walking Engine

2.6.2 Walking Engine Black-Box

In this thesis anything contained within the walking engine box (see Fig. 2.10) is considered part of the walking engine. In short this is both the software which makes the final decisions and calculations for the joint commands, as well as the list of parameters used for these calculations. The parameters will be treated as a black-box for the remainder of this thesis.

The motivation for this is the removal of human guesswork in analyzing walking quality. Each team's exact logical setup and definition of the walking engine is unique, and a comparison of the walking engines provides only a qualitative analysis of software implementations when considering the individual intricacies of each. Another motivation of this thesis is to perform quantitative comparisons of the walking engines of other teams. The differences in walking engines are most like the difference in gait patterns that each team uses and the performance of these are what the metrics in this thesis quantitatively analyze.

2.7 RoboCup Standard Platform League Competition

A RoboCup SPL match consists of two teams of autonomous NAO robots playing soccer against each other. HULKs, B-Human, and Nao Devils compete in the *Champions Cup*, with games being played with 7 robots from each team. There is also a *Challenger Shield* where teams play with 5 robots each, as well as simpler rules for the purpose of beginner friendliness. The match is played on a 9 by 6 meter field with markings closely resembling those of a standard soccer field.

A competition match has a length of 20 minutes, split into two 10 minute halves. Before the beginning of the match all the robots are placed onto the sidelines of the field. Once the match starts no human intervention by members of the teams is permitted until after the respective 10-minute half has concluded. This means that during the game the robots must act autonomously in object detection, locomotion, and multiagent decision-making.

RoboCup SPL tournaments consist of multiple matches over the course of several days. On a competition day a team plays 2-3 matches and has time in between these to make adjustments to software and parameters.

As mentioned in Chapter 1 the NAO model is identical between teams. The SPL rules ensure this by disallowing hardware modifications to the robot.

3 State of the Art

The walking quality of legged and specifically humanoid robots is analyzed through various evaluation categories, such as speed and energy consumption. Depending on the research focus, metrics are defined, that are then used in a closed or open loop manner. Closed loop evaluation of metrics is targeted towards redesigning and improving the robot through hardware or software changes. Open loop evaluation seeks to present the success of a robot without further iterating on design changes. While the goal of this thesis is to evaluate metrics that are suitable for iterating on software and parameter changes in a closed loop manner, this thesis does not actually propose specific changes, essentially treating these metrics as open loop.

A metric for an evaluation category is a single value which represents an objective summary of the success of the robot in that category. The definition of a metric thus requires not only the calculation of a summary value, but also the distinction whether this value should ideally be high or low. For example, when analyzing energy consumption, one choice of metric is the average energy consumption, which should be low. Conversely, when analyzing a robot for its ability to walk up steps, the higher the step size, the better the performance of the robot. The choice of metric depends on the application of the metric in the robot design process.

This chapter thus dives into various metrics and evaluation categories used in the research of legged robotics. It also includes an analysis on applicability to the NAO based on the problem statement and limitations presented in Section 1.1.

3.1 Robustness

Robustness in software deals with the capability of a system to operate correctly even in abnormal, such as unexpected or adverse, conditions. The same applies to the walking of a robot. One of the main issues when dealing with bipedal walking is primarily the concept of an inadvertently unstable system as discussed in Section 2.5. Often dealt with as an inverted pendulum, keeping the upright stability of a robot requires constant adjustments and stabilization. Generally speaking, falling is considered detrimental to the performance of a bipedal robot, and thus the number of times a robot falls will be used as a simple indicator of walking quality. However, assuming a robot manages to stay upright, there are several components to understanding walking performance in terms of robustness. These are static and dynamic stability, camera stability, resistance to external forces, and the ability to walk on various types of terrain. Hardware robustness also plays a role in this, as a robot must not damage itself during normal operation. This includes situations such as the robots limbs colliding with each other, and the robot falling over.

3.1.1 Static and Dynamic Stability

The support polygon of a robot is determined by its support foot or supporting feet. It is defined as the convex hull of the points of the feet which are in contact with the ground. A robot is statically stable when its CoM projected to the ground is within its support polygon. Walking algorithms based on static stability have the limitation that the robot shifts its CoM from

the supporting polygon of one foot to that of the other, exclusively during the double support phase[17]. Additionally, static stability does not properly represent stability of the robot while it is in motion.

Because of limitations of the CoM the Zero Moment Point (ZMP)[18] is introduced, which takes into account the dynamics of the CoM to give a better representation of stability during dynamic movement of the robot. The concept of the ZMP asserts that for a dynamically stable robot there exists a point $P := (P_x, P_y)$ as the location of the normal force acting on the robot, such that the sum of forces acting on the robot create no moment of inertia about any axis parallel to the ground. This includes the gravitational and normal forces as well as acceleration caused by the robot itself or external forces (see Fig. 3.1).

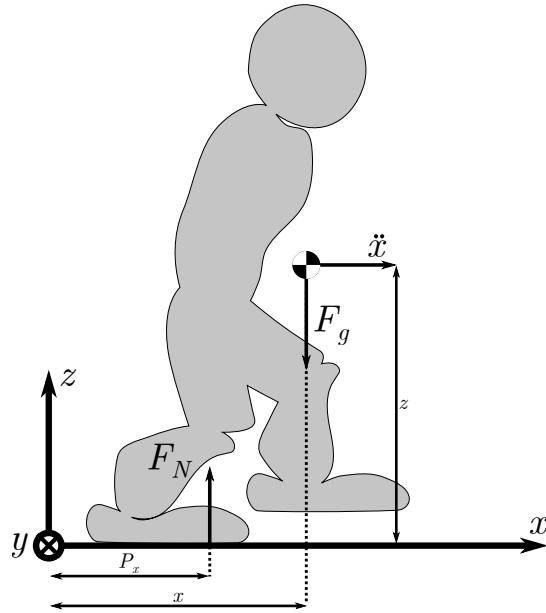


Figure 3.1: Zero Moment Point (ZMP)

Based on the work of Kajita et al.[19] the moment equations thus require:

$$0 = gx + \ddot{x}z - gP_x \quad (3.1)$$

$$0 = gy + \ddot{y}z - gP_y \quad (3.2)$$

where g is the gravitational constant and \ddot{x}, \ddot{y} are the acceleration of the CoM. Additionally, x, y, z are the positions of the CoM, and P the position of the normal force based on an arbitrary coordinate system as long as the X and Y axis are parallel to the ground. The components of the ZMP P can then be calculated by:

$$P_x = \frac{gx + \ddot{x}z}{g} = x + \frac{\ddot{x}z}{g} \quad (3.3)$$

$$P_y = \frac{gy + \ddot{y}z}{g} = y + \frac{\ddot{y}z}{g} \quad (3.4)$$

For dynamic stability the location of the ZMP must still be within the support polygon. To this end the dynamic stability of the robot as a metric is determined by whether the ZMP always remains within the support polygon.

If the ZMP is allowed to fall outside the support polygon, it is known as a Fictitious Zero Moment Point (FZMP). When this occurs the normal force acting on the robot is applied at some point at the edge of the support foot. If the robot instantaneously stops walking, the positioning of this normal force does not prevent the robot from falling. If the normal force acts through the FZMP instead, a fall is prevented. Countless walking algorithms[3][17][6] use the FZMP to allow for a smoother walk. Moving the ZMP to outside the support polygon of the support foot during walking allows for a transfer of balance toward the next support foot much earlier in the walk cycle, as opposed to exclusively doing this during the double support phase.[18].

Evaluating the FZMP is more difficult, as movement of the point outside the support polygon only implies momentary instability which may be desired as the robot smoothly shifts its weight mid-step. To combat this issue most research[3][20][21][19] uses a reference trajectory of the ZMP based on the desired motion of the robot. The deviation of the resultant ZMP or FZMP from this plan is then used as a metric. For reading clarity this thesis only uses the term ZMP for the remainder of the text, with full acknowledgement that this may refer to a FZMP should it lie outside the support polygon.

Additionally, this is only one method of calculating the ZMP. Other methods use the movement of the individual limbs to obtain a more accurate ZMP. However, this requires either sensors on the individual limbs of the robot, or an accurate model for how the dynamic motion of the limbs based on a measured acceleration of the motors attached to each limb[18]. Because of the limitations of the NAO hardware, the simpler method shown in Equations 3.3 and 3.4 is used in this thesis.

3.1.2 Camera Stability

Another component of robustness is stability of the cameras or camera of a robot. A stable camera provides a clear image feed that allows for the tracking of objects while navigating and localization of the robot within an area[22]. In the RoboCup this is critical for the NAO to determine its location on the field as well as detection the ball and other robots. Optimally, camera movements relative to the environment are only the result of planned actions. Movements such as walking which may result in unpredictable shifts of the camera are to be compensated for. For this purpose robots are designed with hardware stabilization of the camera in mind. This can take the form of hydraulics, which Bazeile et al.[22] designed for a quadruped robot intended for terrain mapping. The performance of the stabilization was evaluated through the accuracy of the generated terrain maps. Another stabilization method involves a motor for each eye, which Habra et al.[23] used to compensate for the head movements of a humanoid robot. The performance of their stabilization algorithm was evaluated through pixel accuracy of the resulting images.

3.1.3 External Forces

A robot's robustness against external forces or resistance to being pushed deals with both a robot's ability to recover from a push and the magnitude and duration of a push[24][25]. Faraji et al.[25] measure the recovery time both in seconds and in number of steps the robot needs to take before it is able to resume normal walking operation. The magnitude of the push is both be analyzed in newtons[24][26] or in newton-meters[25], which uses the positional data of the push relative to some reference frame to calculate the moment created by a push. Robustness to external forces is also used in conjunction with an analysis of the ZMP[27]. If the ZMP is

located centrally to the support polygon, a larger force can be applied to change the acceleration of the CoM before the ZMP leaves the support polygon and the robot becomes instable.

The RoboCup competitions are situations in which the NAOs feel the influence of external forces, as robots have the tendency to bump into each other over the course of a game. An analysis of robustness against external forces is therefore very valuable.

3.1.4 Terrain

A robots walking robustness in respect to terrain is analyzed in research in one of several ways. One option is to assign a passing or failing value to the robot and its software for a specific set of obstacles or terrain definitions. An example of this is the bipedal robot SLIDER, for which Wang et al.[26] designed a step algorithm specifically for being able to walk on slopes. SLIDER successfully walks on 5°, 10° and 15° slopes. Another option is a step height analysis, one that is extensively surveyed by Ke et al.[24]. It involves the analysis of step height of the robot for the purpose of walking on terrain with disturbances. This is evaluated as the absolute achievable step height or as a percentage relative to the robots leg height. At the time of survey the robot ATRIAS[28] was able to successfully step both up and down 30 cm steps, and the robot MABEL[29] was able to achieve a step height of 20% of its leg height.

3.1.5 Hardware

Hardware robustness means that a robot should not experience mechanical or electrical failures during its operation. During normal operation this means deterioration caused by the planned movement of a robot should be minimized. To specifically combat deterioration during prolonged parameter optimization Hwangbo et al.[30] proposed a novel optimization algorithm which “outperformed current state-of-the-art algorithm in all tasks by a factor of three or more” [30]. Unfortunately Hwangbo et al. only seek the reduction of hardware “wear and tear” as a goal, but do not propose a method of quantifying hardware deterioration.

Hardware robustness is also studied in the context of humanoid robotics by Kakuichi et al.[31]. Kakuichi et al. analyze the methods for preventing mechanical and electrical damage to a robot during falls, specifically through the addition of hard points to absorb impact shocks.

3.2 Speed

Speed or velocity can be considered general performance metrics in many robotic systems. Most research specifically uses velocity as a metric, as it includes a direction component. However, when trying to judge a robot’s ability to traverse a specific path in a certain amount of time, only the time and the path’s length effectively matter, which evaluates to an average speed. Thus going forward these two terms are used somewhat interchangeably.

Speed is especially critical for robots for which traversing a large distance in a given amount of time is indicative of success in a task. An example is search and rescue robots, such as Boston Dynamics’ Spot[32] in which walking speed of the robot may determine the survival rate of victims of a disaster. In other scenarios it leads to higher throughput such as for robots that may transport goods in a warehouse. In the context of the RoboCup increased speed leads to higher competitiveness as there is a clear advantage over other teams if the ball may be reached faster than the other team[33].

For research speed is often used in conjunction with other metrics or evaluation categories. Research by Roy and Pratihar[34] focused the analysis of a six-legged robot and analyzed the

relationship between speed and energy consumption. They used a target desired velocity as an input parameter to calculate the needed gait pattern and timings that would be needed to achieve such a velocity. Their paper however makes no mention of whether the robot was able to reach that target speed. Research by Weingarten et al. [35] specifically attempts to maximize speed in a six-legged robot while using gait adjustments to improve energy efficiency.

3.3 Energy Consumption

There are two main categories for analyzing energy consumption of a robot. First, the peak energy consumption E_p is required during the design of the robot to ensure that the battery providing the energy can withstand such a desired power draw. Second, the energy efficiency and consumption must be known in order to understand robot lifetime between charging cycles and in many robotics industries the energy efficiency is an important financial consideration[36].

The work of Shamsuddin et al.[37] in 2011 hopes for advancements in increased energy efficiency of humanoid robots. However, the 2018 work by Kashiri et al.[38] summarizes that this aspect of humanoid robotics still lags far behind other advancements such as walking stability improvements. A walking engine which has a similar performance in other aspects but outperforms another walking engine in terms of energy efficiency is thus for all intents and purposes a better walking engine. Several RoboCup SPL teams have made improvements to their robotics code in order to improve energy efficiency [9][39], but no clear metrics have been set to analyze energy usage of the robots.

As mentioned in Section 3.2 the work of Roy and Pratihar[34] focuses on energy consumption. Their work also looked at the specific energy consumption E_s which is defined as follows:

$$E_s = \frac{E}{mgd} \quad (3.5)$$

where E is the energy required by a robot having mass m to travel distance d . The specific energy consumption is a dimensionless quantity and is used in their research as an index of energy efficiency. Roy and Pratihar find that in their six-legged robot an increased velocity decreases the specific energy consumption, meaning that a higher velocity leads to a more efficient robot.

3.4 Heat Production

All electrical systems produce heat due to resistances in components converting electricity into heat. This is a problem as sustained high temperatures can damage components of the robot over time. Primarily this occurs through the demagnetization of the permanent magnets in the motors at high operating temperatures, but can also lead to burnout of the motor wire windings. There are several methods that have been developed for humanoid robots to combat heat in joint motors, which are evaluated for success based on the resulting heat loss. One method involves using either water-cooling[40] or thermoelectric[41] cooling modules on the motors. The success of these methods was evaluated based on motor temperature and achievable torque.

Another method is the cooling of joints through current control. One such example is the two resistor thermal model designed by Urata Et al.[42], which allows for the estimation of a motors core temperature based on electrical current. This temperature estimate is then used to lower the temperature over longer periods of time by limiting current.

The NAOs hardware and software provides for a similar setup. The motors of the NAO do not have thermistors and there is no actual temperature measurement conducted within the

robot. Rather, joint temperatures are “a simulated one, using [the] electric current value of the motor” [13], although the exact model used for this is not published.

The NAO’s electric boards implement a temperature limit to protect the motor when the temperature of a joint is deemed too hot. This is realized by reducing the joint’s stiffness. The stiffness is a value which indicates a motor’s ability to resist movement induced by an applied torque, 0 being none and 1 being the maximum torque the motor is rated for. In practice when this occurs, even momentarily, it results in the robot simply collapsing on the field if, for example, the ankle joints have their stiffness reduced to 0. It is thus of interest to keep the operating temperature low by limiting the amount of current required by the motors. Work by Mellmann[39] has already analyzed that above a specific threshold current supplied to the motors the joints heat up under stationary conditions.

3.5 Slipping

Slipping deals with the concept of a robot’s limb moving while it is the supporting limb, usually in the direction opposite of the robot’s target direction of travel. Slipping leads to inefficiencies in the locomotion of the robot, as the distance covered by each step is reduced, meaning the energy cost of traversing a distance increases [43]. It also leads to the robot falling down as the location of the ground reaction forces change, and they are no longer able to counteract the force of gravity.

Respectively to these two analysis methods there are two main ways to measure slip. The first is based on the average velocity achieved by the robot over some distance[43], which for a robot with slip is decreased compared to one that does not slip. The other method involves measuring the ground reaction forces in the tangential direction along the ground using an angled force sensor [44].

3.6 Double Support Time

As discussed in Section 2.5 the walking cycle of a robot is split into two components, the single support phase and the double support phase. The double support phase provides unique challenges when developing the control algorithm of a bipedal robot[45], as it must deal with the process of transferring the CoM of the robot from one foot to the other while continuing to provide forwards motion. The double support time is calculated as part of a step cycle using:

$$t_{DST} = (t_{L1}^{up} - t_R^{down}) + (t_R^{up} - t_L^{down}) \quad (3.6)$$

where $t_{L1}^{up}, t_R^{down}, t_R^{up}, t_L^{down}$ are defined as the moments in time when left and right foot are placed down on or picked up from the ground as per Figure 2.7. The proportional length of the double support phase with respect to the walk cycle is then:

$$\delta_{DSP} = \frac{t_{DST}}{t_{L1}^{up} - t_L^{up}} \quad (3.7)$$

The robots Flame and TULip[46] specifically take advantage of an increased double support time in order to regain energy for the next step using a system of springs in the foot. The robot Mabel[29] was specifically designed to have an instantaneous double support phase i.e. zero double support time made possible by the guarantee that the robot’s feet do not experience slipping.

3.7 Applicability

The applicability of the evaluation categories to this thesis depend on heavily on the third and fourth challenges presented in Section 1.1. Those are the context of the RoboCup SPL, both while performing parameter tuning and during a match, and the requirement that each team's robotics code be treated as a black-box. Also, an additional challenge is that there exists ambiguity in research in the definitions of metrics for the double support time and hardware deterioration.

Due to these limitations this thesis only defines metrics for these categories:

- Energy consumption and heat production, as the data for these are easily obtainable from the robot sensors.
- Camera stability, judged by the rotation and acceleration of the robot torso as opposed to the pixel velocity discussed in Section 3.1.2. Although camera stability also depends on head movement, keeping the torso steady ensures the camera can be steady if the robot is not moving its head.

The static and dynamic stability based on CoM and the ZMP, as well as the double support time, are used to provide a qualitative comparison between walking engines.

The rest of the walking quality analysis categories are not used. The following sections describe the limitations the challenges from Section 1.1 create for this thesis.

3.7.1 RoboCup SPL Context

The goal of this thesis is to find metrics which can be evaluated for a robot while parameter tuning takes place before a match. As explained in Section 1.1 this means no external tools are required. This prevents both *speed* and *robustness to external forces* from being applicable as metrics in this thesis as one requires the measurement of distances and time, while the other requires a standardized tool for applying a force to a robot repeatably. SPL rules prohibit the modification of robot hardware, which prevents the introduction of angled force sensors or other external tools to determine *slipping*.

The terrain of a RoboCup field does not contain any slopes or large terrain disturbance. Recent research on the topic of terrain in the RoboCup primarily deals with the motion control concepts needed to walk on artificial grass at all[47][48] but does not focus on the finer details behind it. From experience by the HULKs the terrain variation at different events comes through minor differences in the type of artificial grass used on the field, which may differ in length and rigidness depending on the event organization. This provides a challenge in studying terrain robustness in the context of this thesis. The teams walking engine is in general designed to be able to walk on artificial grass. The HULKs software will thus always be assigned a passing mark for any field which matches the RoboCup requirements. Analyzing how well the team can walk on each field must then be based on another metric, such as one of the stability metrics mentioned in Subsection 3.1.1. In this case the field terrain becomes an input parameter and the stability is judged as a metric dependent on the field. From this the walking engine's ability to adapt its parameters dynamically to each field is judged as an average of the individual performances of each field. A walking engine which adapts its parameters will perform better in such a metric than one which whose parameters are fitted to a specific field. However, even though the goal of this thesis is to quantify walking such that parameter tuning prior to games may be made easier, the scope of this thesis does not allow for the setting up of several unique fields to analyze data from.

3.7.2 Black-box Evaluation

Treating each team's robotics code as a black-box means that there is no access to motion planning information, nor to camera data or robot localization. An accurate analysis of the *dynamic stability* of the robot requires the knowledge of a reference ZMP which would stem from motion planning information. Analyzing *camera stability* using pixel velocity as described in Section 3.1.2 would require access to the camera information, and a knowledge of the planned movement of the head.

The location of a robot is known through its localization algorithm, but this is also contained within the black-box of each team's robotics code. This makes calculating the *speed* based on internal measurements of the robot impossible.

Due to the fact that location and speed information of the robot are not available, two other metrics cannot be analyzed. One is the efficiency loss due to *slipping* as it depends on knowledge of the robot's actual speed vs. target speed. The other is the *specific energy* from Eq. 3.5 which requires knowledge of the distance traveled by the robot.

3.7.3 Addressing Ambiguity

The double support time of a robot's walk isn't universally applicable as a metric, unless specific applications of a robot are considered as in Section 3.6. So although the double support time can be measured in this thesis it is not reasonable to apply it as a metric because no work exists to support either argument that a lower or higher double support time leads to higher quality walking.

There are two issues when dealing with *hardware robustness* in the context of this thesis. First, there are no clear definitions of how to measure hardware deterioration of a robot(see Sec. 3.1.5). Reichenberg[49] analyzes the backlash of specific motors of a NAO robot in an attempt to understand the effects of hardware deterioration on the robot's ability to perform certain motions. However, the work focuses on designing a controller that is able to deal with backlash, and does not further quantify differences in hardware across robots. Second, for the HULKs changes in hardware of the NAO through deterioration, besides obviously broken gears and the like, have only made themselves noticeable over the course of several months or years. Any metric that measures how the walking engine affects hardware, in a way that may lead to eventual long term failure, is likely to show no results. Thus, this thesis does not attempt to define one.

4 Method

The method of gathering data involves both the software used to collect data and the experimental setup. For the software aspect, a simple script is designed which fits into the limitations of this thesis by solely relying on the raw MessagePacks intercepted from between LoLA and the robotics code. This data is converted from MessagePacks into a human-readable CSV format. The experimental setup closely approximates what a robot experiences in half of a RoboCup SPL match.

4.1 Data Gathering Method

The best way to gather data on the robot’s walking is to collect the data directly provided by LoLA. There are several reasons for this. First, this adheres to the limitations put forth in this thesis as presented in Section 1.1 as it requires no external data sources. Second, it provides all the information necessary for the evaluation of heat, temperature, as well as gyroscopic and accelerometer based stability as described in Section 3.7. Finally, gathering data in such a way does not require the modification of the code from other teams, another limitation presented in Section 1.1.

As discussed in Section 2.4 LoLA only allows for one incoming socket connection, which is taken up by the data handling process of each team. As shown in Figure 2.6, this is the HULA process in the case of the HULKs. Nao Devils also have a similar abstraction layer, while B-Human connects to LoLA directly with their main binary. For simplicity the process which usually connects to LoLA is simply be called the “main” service.

The main service connects to LoLA via the Unix socket identified by `/tmp/robocup` as defined by Aldebaran. The fact that this is identically defined for all teams is advantageous. As previously mentioned, it is possible to move connected Unix sockets by simply renaming them, such that the connection persists, but the file handle has changed. This brings about the design of *hulahoop* which essentially exists as an intermediary between LoLA and the main service. Hulahoop stores the MessagePack, and the timestamp in milliseconds when the data was received, as bytes in a file. It consists of the hulahoop binary which handles the transfer and storage of data, and the *hulahoop.service* files which handle the arrangement of connections to LoLA and each team’s respective main service. Summarized for the context of the HULKs code, hulahoop and *hulahoop_hulks.service* execute the following steps:

1. `hulahoop_hulks.service` stops HULA and the HULKs robotics code which leaves the `/tmp/robocup` socket available.
2. `hulahoop_hulks.service` then starts hulahoop which connects to LoLA
3. hulahoop moves the connected `/tmp/robocup` socket to a new file handles `/tmp/lolasocket`
4. hulahoop opens a new Unix stream socket at `/tmp/robocup`

5. `hulahoop_hulks.service` restarts HULA which is once again able to connect to `/tmp/robocup` even though this is now being provided by hulahoop and not by LoLA. `hulahoop_hulks.service` also restarts the robotics code

Hulahoop then continues running, receiving messages both from the LoLA socket and the main service socket sides and forwards these to the respective other socket. An overview of these steps is seen in Figure 4.1. This allows the data that LoLA is sending and receiving to be logged, while still forwarding it to and from the main service running the robotics code. Hulahoop resolves the single connection limitation to LoLA's Unix stream socket.

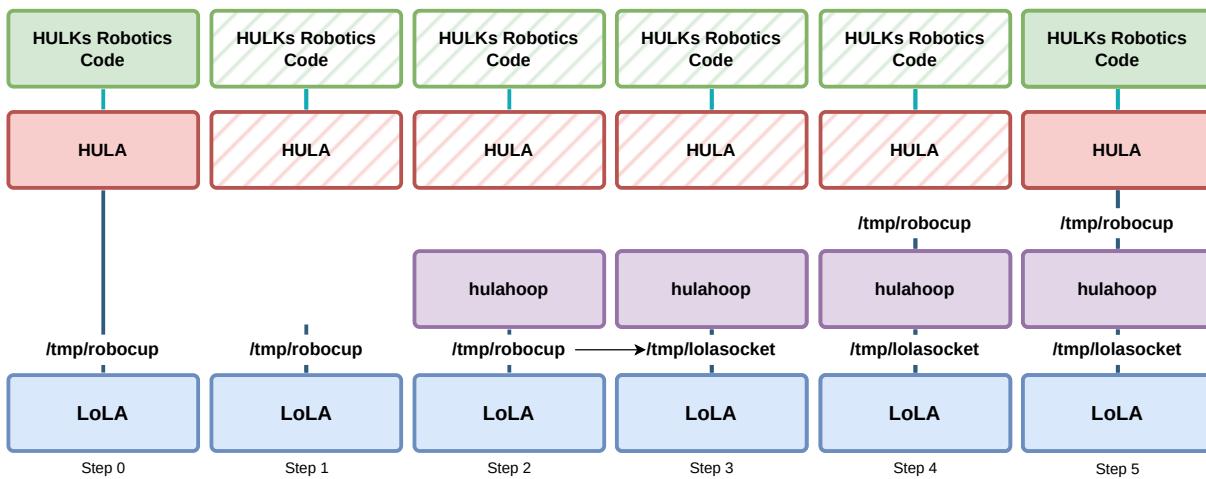


Figure 4.1: Hulahoop execution steps

4.1.1 Delays in Data

In any robotics system it is never possible to know the exact values of sensor data in a specific time point, because the processing and transmission of data through software will inherently cause delays. Böckmann[50] analyzed the delays between motion commands to when the sensors showed motor movement on the 5th generation NAO and found this to be 30 ms. Reichenberg[51] updated this information for the 6th generation NAO, as used in this thesis, and found this to be 36 ms, equivalent to 3 frames. The source of the delay is either HAL, which delays motor control by some time, filtering of sensor noise, which delays the measurement of motor positions, or some combination of the two. Richter-Klug[52] specifically analyzed the IMU data of the NAO and discovered that the gyroscope and accelerometer data is not updated every frame, rather LoLA alternates in sending new values for each frame. This means that every other frame the IMU data has at least a 12 ms delay.

Hulahoop now introduces a new delay as there is some processing time to write the MessagePack data into a file when the MessagePack is forwarded between LoLA and the main service. To test this a small script which mimics the behavior of LoLA is used such that the time delay between data being sent to hulahoop and the same data being received from hulahoop is measured. While this brief test did not constitute perfect benchmarking of hulahoop, it did show that the delays caused by hulahoop are only up to 100 microseconds. Given the magnitude of delays already existent in the data transfer to the electronic boards of the motor and from the various sensors, the delay from hulahoop is, for this thesis, considered insignificant.

4.1.2 Data Conversion from MessagePack to CSV

As described in Section 4.1, the data is stored as raw bytes from the received MessagePacks. This is to ensure no further delays occur due to data processing while data is being collected. The MessagePacks sent by LoLA are all identically 896 bytes in size, meaning that every 896 bytes read out of the binary file can be treated as a MessagePack again in post-processing. The MessagePack format keeps the keys as human-readable strings while the values are encoded. Because of this, the MessagePacks are first decoded into JSON (see Fig. 4.2). From there the tree structure of the JSON is flattened and written as a single row into a CSV file where the columns match the key names.

```
{
    "Accelerometer": {
        "x": 3.4296681880950928,
        "y": 0.5748047232627869,
        "z": -7.012617588043213
    },
    "Angles": {
        "x": 0.04314344748854637,
        "y": 0.0613592304289341
    },
    ...
    "Position": {
        ...
        "left_hip_pitch": -0.49083805084228516,
        "left_hip_roll": 0.0123138427734375,
        "left_knee_pitch": 0.8666679859161377,
        ...
    },
    "Received_at": 59614.0,
    ...
    "Temperature": {
        ...
        "left_hip_pitch": 28.0,
        "left_hip_roll": 27.0,
        "left_knee_pitch": 28.0,
        ...
    },
    ...
}
```

Figure 4.2: JSON excerpt of one frame

4.2 Experimental Setup

The HULKs parameter tuning is usually done in smaller bursts, where the robot walks for a minute or two, after which some adjustments are made and the process repeats. However, any parameter changed must perform well for the length of a RoboCup game, so metrics in this thesis are analyzed across a 10-minute time frame. However, in order to make the setup as simple as possible, the robot’s behavior does not depend on a certain role assignment or player position as might occur in a real game. Instead, the robot is brought into a *playing* state in which its only focus is to chase the ball and kick it into the opponent goal. This forces the robot to continuously “chase” the ball as it would in a real match, including any forward locomotion,

side steps, or turning the walking engine generates. While not all robots would be walking continuously in a normal game, testing like this ensures representation of the most extreme case that a robot may have to endure.

4.2.1 Testing Environment

The experimental setup includes a scaled down version of a standard SPL field, with dimensions seen in Figure 4.3. The following steps are then followed:

1. Hulahoop is started
2. The robot is placed standing up on the sideline at position [S] and a ball is placed on the penalty marker at position [F].
3. The robot is brought into its *stiff* state through a push of the chest button. In this state the robot stands upright with its arms at its sides.
4. The robot is brought into the *initial* and then playing states through another two pushes of the chest button.
5. The robot is given time to walk onto the field towards the ball until it reaches position [A].
6. The ball is then rolled by a human tester to positions [B] followed by [C] and [D], at a pace which keeps the robot within roughly 1 m of the ball.
7. The ball is then rolled back to [E], [F] and finally [A] in order to make the robot turn around and walk back to the other side of the field.
8. The previous two steps are then repeated for a duration of 10 minutes as time by a stopwatch. If the robot falls at any point the stopwatch is paused and only resumed after the robot has continued walking. This is to ensure that 10 minutes worth of data on just walking can be collected.

The line connecting points [A]-[F] represents the approximate path the robots walked across the field.

4.2.2 Testing Scenarios

There are two main variables that should be tested: hardware and walking engine. As described in Section 1.1, the first variable, comparing hardware, is used to evaluate walking metrics on different robots in order to understand how the deterioration of hardware affects walking performance. For this purpose four HULKs robots were selected, which are identified in the rest of this thesis as NAO1, NAO2, NAO3, and NAO4. The other goal is to compare the walking engine of the HULKs with the walking engines of B-Human and Nao Devils. Section 1.1 also pointed out that using human judgment it is difficult to understand how the change of one parameter affects another. For this purpose, previously manually tuned faster walking parameters, are treated as an additional walking engine, here called FastHULKs. The difference in the parameters is primarily exemplified by quicker steps and a tilt of the torso so that the robot does not fall over. The FastHULKs parameters are not currently used by the HULKs as by the current method of guessing these parameters seem to make the robot unstable and produce too much heat.

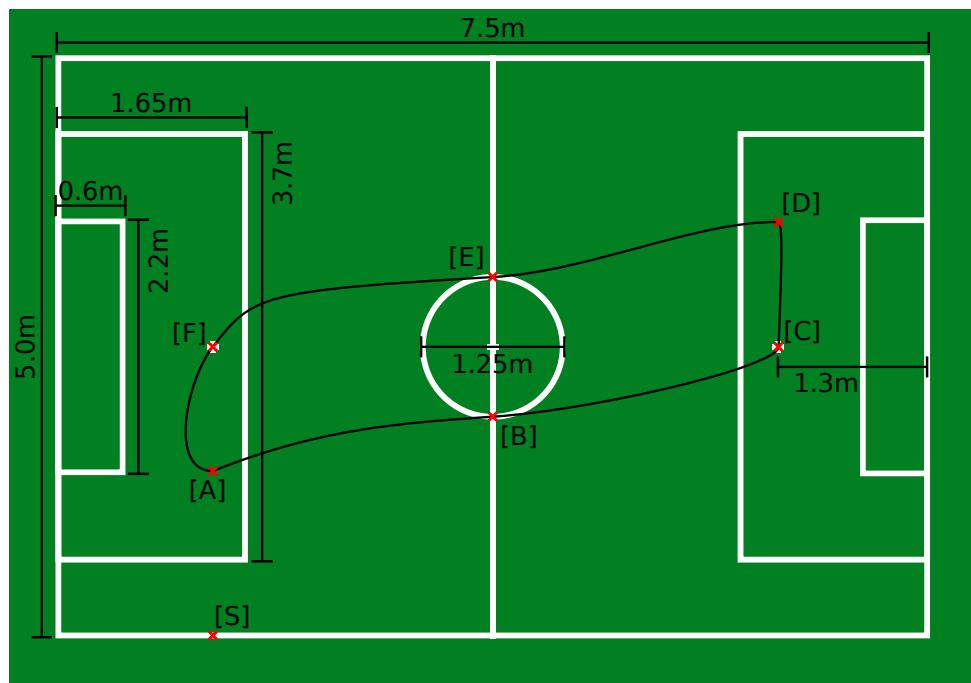


Figure 4.3: Field Setup for Data Gathering Purposes

This thesis numerically checks this assumption for validity. The four robots along with the four walking engines leads to 16 total scenarios for which data is collected. In this thesis the specific scenario from which data was collected is identified by a combination of the walking engine and the robot number, i.e. HULKs-NAO3.

5 Data Presentation and Post Processing

After the data is collected it must be processed in order to obtain comparable data points. The rest of this chapter explains important secondary values relevant to the analysis categories from Chapter 3 that can be calculated from the raw data. Additionally, during the data collection, some robots fell and had to stand back up to continue walking. These falls must be counted and then removed from the data.

5.1 Collected Data

As stated in Section 4.1.2 the data of each frame is written as a single row into a CSV file. An excerpt of this data is shown in Table 5.1, the last row of which corresponds to the JSON excerpt shown in Figure 4.2. The whole table includes the sensor values listed in Table 5.2 with their corresponding units.

The values for the accelerometer and gyroscope are given based on the axes relative to the torso, while the angle values show the rotation of the torso relative to its upright position. As shown in Figure 2.2 the positions of each motor are given in its local coordinate system.

Accelerometer.x	Angles.y	Position.left_hip_pitch	P...left_knee_pitch	Received_at (ms)
-1.6190	0.1072	-0.5200	1.0830	59433
-1.6190	0.1116	-0.5384	1.0860	59445
-1.8202	0.1137	-0.5537	1.0753	59457
-1.8202	0.1110	-0.5568	1.0446	59469
-0.7089	0.1049	-0.5537	0.9971	59481
-0.7089	0.0989	-0.5415	0.9342	59493
0.0575	0.0951	-0.5399	0.8759	59505
0.0575	0.0922	-0.5430	0.9020	59517
3.2189	0.0909	-0.5445	0.8743	59529
3.2189	0.0901	-0.5476	0.8667	59541
5.6906	0.0849	-0.5491	0.8513	59553
5.6906	0.0761	-0.5445	0.8452	59565
1.1688	0.0665	-0.5338	0.8483	59578
1.1688	0.0585	-0.5246	0.8559	59589
3.4297	0.0535	-0.5062	0.8621	59602
3.4297	0.0431	-0.4908	0.8667	59614

Table 5.1: An excerpt of B-Human-NAO1 CSV with abbreviated headers

The CSV table also includes the timestamp of when the data was received in milliseconds. To process the data and perform computations the CSV is imported into a *Pandas*[53] *dataframe* using *Python*. The dataframe contains the same table of values as the CSV, but is easier to perform operations on. Secondary values that are calculated from the sensor data are added to the dataframe as a new column. This thesis uses the names of the headers of each column to

identify the sensor values. For example, the rotational position of the pitch motor of the left is identified as *Position.left_hip_pitch*. These labels are considered to be self-explanatory as to which sensor value is used.

Sensor Category	Number of values	Unit
Accelerometer	3: X, Y, Z axes	$\frac{m}{s^2}$
Gyroscope	3: X, Y, Z axes	$\frac{rad}{s}$
Angles	2: X, Y	rad
Battery Charge	1	%
Battery Current	1	ampere
Battery Temperature	1	$^{\circ}C$
FSR	8: 4 on each foot	kg
Motor Current	25: for each motor	ampere
Motor Position	25	rad
Motor Stiffness	25	unitless
Motor Temperature	25	$^{\circ}C$

Table 5.2: Sensor Values

5.2 Coordinate Systems in this Thesis

As stated LoLA provides the gyroscope and accelerometer values relative to the robot's torso. To deal with this the robot coordinate system \mathcal{R} is introduced. This is located centrally at the base of the robot torso, 221.5 mm below the robot neck and the Z axis is always aligned with the torso (see Fig. 5.1). Values from the IMU are provided by LoLA in this coordinate system.

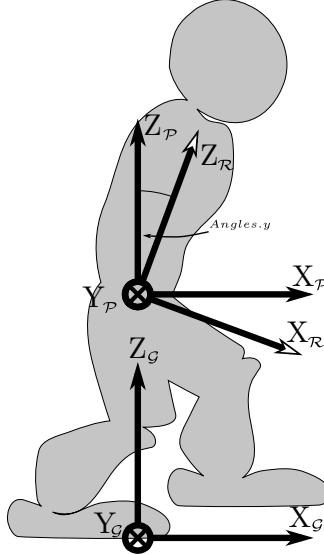


Figure 5.1: Robot coordinate system

A point $P_j := (x, y, z, 1)$ in the homogenous coordinate system j can be represented in coordi-

nates in \mathcal{R} by:

$$P_{\mathcal{R}} = T_j^{\mathcal{R}} \cdot P_j \quad (5.1)$$

Where $T_j^{\mathcal{R}}$ is the homogenous coordinate system transformation from coordinate system j to coordinate system \mathcal{R} .

For each limb the kinematics calculations from the HULKs robotics code are used to determine the transformation $T_j^{\mathcal{R}}$ of each limb j . Note that $T_j^{\mathcal{R}}$ depends on the position and rotation of the respective links in the kinematic chain connecting the limb j to the \mathcal{R} coordinate system.

As the robot walks and the torso rotates relative to the ground so does the \mathcal{R} coordinate system. To combat this the *parallel* coordinate system \mathcal{P} is introduced, which is located at the position of \mathcal{R} but always parallel to the ground.

$$P_{\mathcal{P}} = T_{\mathcal{R}}^{\mathcal{P}} \cdot P_{\mathcal{R}} \quad (5.2)$$

$$T_{\mathcal{R}}^{\mathcal{P}} = \begin{pmatrix} \cos \theta_y & 0 & \sin \theta_y & 0 \\ \sin \theta_x \sin \theta_y & \cos \theta_x & -\sin \theta_x \cos \theta_y & 0 \\ -\cos \theta_x \sin \theta_y & \sin \theta_x & \cos \theta_x \cos \theta_y & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (5.3)$$

Points in \mathcal{R} are calculated in \mathcal{P} using $T_{\mathcal{R}}^{\mathcal{P}}$ which is a homogenous rotation based on $\theta_x = \text{Angles.x}$ and $\theta_y = \text{Angles.y}$ (see Eq. 5.3).

Finally, the position of \mathcal{P} depends on the movement of the robot. However, to calculate the ZMP using the equations 3.3 and 3.4 a corresponding coordinate system parallel to the ground must be used. For this coordinate system \mathcal{G} is used, for which the X and Y axis are parallel to those of \mathcal{P} , while the origin of the Z axis sits on the ground plane. The transformation from \mathcal{P} to \mathcal{G} is:

$$T_{\mathcal{P}}^{\mathcal{G}} = \begin{pmatrix} 1,0,0,0 \\ 0,1,0,0 \\ 0,0,1,z \\ 0,0,0,1 \end{pmatrix} \quad (5.4)$$

where z_d is the vertical position of \mathcal{P} relative to \mathcal{G} and may change over time depending on the robot's motion.

5.3 Ground Contact Estimation

In order to calculate the single and double support time of walking, it must be known whether each foot is in contact with the ground. To determine this, the FSR values are used. The weight of each foot on the ground is the sum of its FSR values.

$$F_{L,k} = \sum_{i \in \Omega_1, j \in \Omega_2} FSR.left_foot_i_jk \quad (5.5)$$

$$F_{R,k} = \sum_{i \in \Omega_1, j \in \Omega_2} FSR.right_foot_i_jk \quad (5.6)$$

where $\Omega_1 = \{front, rear\}$ and $\Omega_2 = \{left, right\}$ as in Figure 2.4

According to the NAO documentation of the FSRs "the returned value is approximate" [13] and best practice is to take into account the change in value rather than the absolute value when performing calculations. For this purpose a hysteresis estimate shown in Table 5.3 is introduced

S_{k-1}	S_k
0	0 if $\alpha - \beta > -\gamma$ else 1
1	0 if $\alpha - \beta > \gamma$ else 1

Table 5.3: Hysteresis Estimate

which is used to determine boolean of some variable S_k . True and false are represented as 0 and 1 in S_k , and are based on S_{k-1} , the value of the variable in the previous frame.

Using $S_0 = 0$, $\beta = 0.6$ and $\gamma = 0.2$ and $\alpha = F_{L,k}, F_{R,k}$. From this it can be sequentially determined for each frame whether each foot is in contact with the ground. These are referred to as *left.contact* and *right.contact* in this thesis.

5.4 Support Foot Estimation

In order for a robot to determine that it is safe for it to take the next step in its walking cycle, it must know that its support foot has changed. One method for this again uses the FSRs in the feet, while the other is based on the kinematic chain. The main application of the support foot in this thesis is calculating z_d , which is needed to calculate the transformation $T_{\mathcal{P}}^{\mathcal{G}}$ (see Eq. 5.4).

5.4.1 FSR

Rather than comparing *left.contact* and *right.contact* the HULKs determine the support foot by comparing $F_{L,k}$ and $F_{R,k}$. For this the hysteresis estimate from Table 5.3 is again used, with the following parameters:

- $\alpha = F_{L,k}$
- $\beta = F_{R,k}$
- $\gamma = 0.2$
- $S_0 = 0$

The resulting boolean value *Left_is_support* is equal to 1 if the left foot is the support foot and 0 if the right foot is the support foot. As the FSR hardware model is the same between teams this calculation is used regardless of walking engine.

5.4.2 Kinematics

The definition of coordinate system \mathcal{P} implies that the position of the robot which has the minimum Z axis component must be in contact with the ground. Therefore, the support foot of the robot is determined by using the positions of the feet relative to \mathcal{P} . A simple approximation uses the point at the origin of the soles of the robot and the kinematics chain to calculate the position of each foot in \mathcal{P} (see Fig 2.4). The origin of the sole in the homogenous coordinates of the left sole is a point $P_{sole} = (0,0,0,1)$. Then it holds:

$$P_{\mathcal{P}} = (x_{sole}, y_{sole}, z_{sole}, 1) = T_{sole}^{\mathcal{P}} \cdot P_{sole} \quad (5.7)$$

where z_{sole} is the location of the sole along the Z axis relative to \mathcal{P} and $|z_{sole}|$ is the absolute distance between \mathcal{P} and the ground. However, this is only used under the assumption that the

foot is always placed flat on the ground while being the support foot, which is not always the case.

If a robot is standing on obstacle, such as the foot of another robot, the distance of the foot relative to \mathcal{P} based on the above equation, may be shorter than the actual distance of any part of the foot relative to \mathcal{P} . This is shown in Figure 5.2 with [1] indicating the distance based on Eq. 5.7, and [2] the actual distance to the ground from \mathcal{P} . To compensate for this such a situation, the vertical position of all points of the foot should be calculated and compared. For this purpose a set Ω_L of 33 points, representing the convex hull of the bottom of the left foot, was extracted from the NAOs URDF file. Set Ω_L is plotted in Figure 5.3 together with an outline of the shape of the foot for reference. Only a convex hull is considered as points fully contained in the convex hull are unable to make contact with the ground.

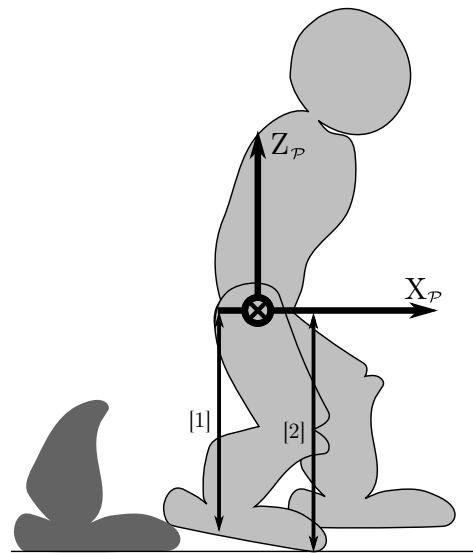


Figure 5.2: Angled support foot

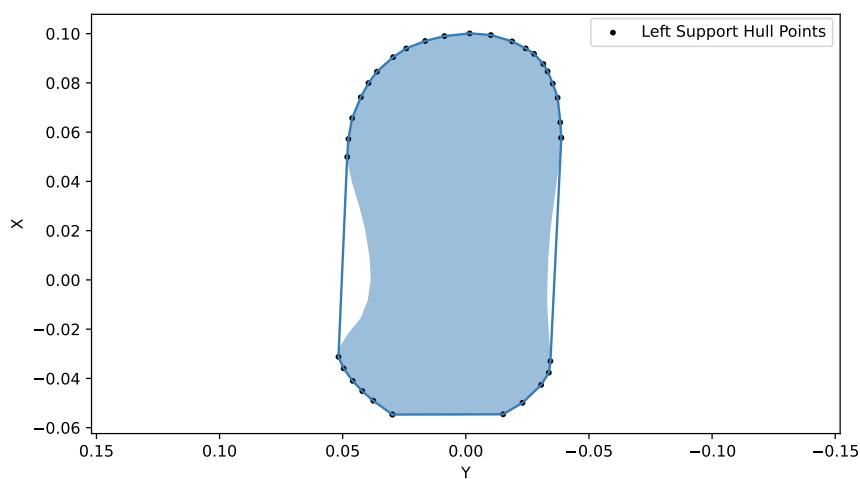


Figure 5.3: Convex hull of left sole in *Lsole* Coordinates

Let $P_{L,l} = (x,y,0,1) \in \Omega_L$ be a point of the convex hull in homogenous coordinates of the left sole. Then the corresponding point $P_{R,r} = (x, -y, 0, 1) \in \nabla$, where Ω_R is the set of points in the convex hull of the right foot. The Z axis position of these points relative to coordinate system \mathcal{P} is given by:

$$z_i \in (x_i, y_i, z_i) = P_{\mathcal{P},i} = T_{sole}^{\mathcal{P}} \cdot P_{sole,i} \forall P_{sole,i} \in \Omega_L, \Omega_R \quad (5.8)$$

Let $z_L := \{z_i\} | P_{sole,i} \in A$ and $z_R := \{z_i\} | P_{sole,i} \in R$. From here the support foot is determined by:

$$left_is_support_{\kappa} = \begin{cases} 1 & \text{if } \min\{z_L\} > \min\{z_R\} \\ 0 & \text{otherwise} \end{cases} \quad (5.9)$$

where κ is used to differentiate between calculation based on FSR and kinematics data. In short, the foot which has a single point further towards the ground than any point in the other foot is considered the support foot.

5.4.3 Calculation of z_d

A comparison of the two above named methods is shown in Figure 5.4. The estimation of the support foot using kinematics is delayed in time compared to the method using FSR, where the width of the shaded area represents the delay. The average value of this was evaluated for the HULKs-NAO4 scenario and was found to be 30.8 ms. This approximately correlates with the delays of the motor position sensors discussed in Section 4.1.1. As it is important for the robotics code to know the value of the support foot as soon as possible it makes more sense to use the values provided by the FSR sensors. This thesis thus uses $|z_d|$ based on:

$$z_d = \begin{cases} |\min\{z_L\}| & \text{if } left_is_support = 1 \\ |\min\{z_R\}| & \text{otherwise} \end{cases} \quad (5.10)$$

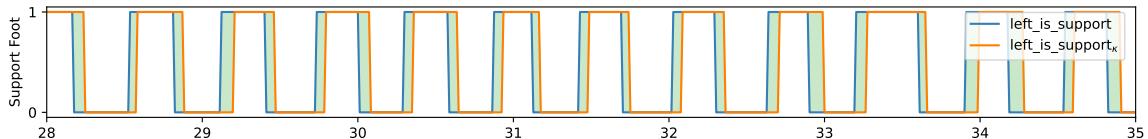


Figure 5.4: Support foot using two methods from HULKs-NAO4

5.5 CoM, and ZMP

The position of the CoM in the homogenous ground coordinate system is calculated by

$$P_{CoM,\mathcal{G}} = (x_{CoM}, y_{CoM}, z_{CoM}, 1) = \frac{\sum_{j \in \psi} T_j^{\mathcal{G}} \cdot m_j \cdot c_j}{\sum_{j \in \psi} m_j} \quad (5.11)$$

where m_j is the mass of limb j , c_j is the location of the CoM of limb j in local limb coordinates, and ψ is the set of limbs of the robot.

To calculate the ZMP the values of \ddot{x} and \ddot{y} must be known in ground coordinates as well. For this define the homogenous vector $a_{\mathcal{R}} = (Accelerometer.x, Accelerometer.y, Accelerometer.z, 0)$ representing IMU acceleration in robot coordinates as provided by the IMU. Then:

$$a_{\mathcal{G}} = (\ddot{x}, \ddot{y}, \ddot{z}, 0) = T_{\mathcal{P}}^{\mathcal{G}} \cdot T_{\mathcal{R}}^{\mathcal{P}} \cdot a_{\mathcal{R}} \quad (5.12)$$

The values $\ddot{x}, \ddot{y}, x_{CoM}, y_{CoM}$, and z_{CoM} are then used to calculate P_{ZMP*} based on equations 3.3 and 3.4.

The position of the P_{ZMP} depends heavily on the acceleration of the robot, including any noise this sensor may have, which must be filtered[54]. For this purpose a running average with a window size of 8 is used, such that

$$P_{ZMP,k} = \frac{1}{8} \sum_{i=0}^7 P_{ZMP*,k-i} \quad (5.13)$$

The accuracy of the filter cannot be compared to a reference ZMP, as is done in most research. The window size is arbitrarily chosen to be lower than that of both Nagasawa et al.[55] and Ven der Noot et al.[54], which had window sizes 10 and 100 times the sampling rate respectively.

5.6 Support Polygon Calculation

The support polygon as introduced in Section 3.1.1 is the area of the foot or feet which is in contact with the ground. As this may not always be flat (see Fig 5.2) a 2-D projection onto a flat plane in \mathcal{G} is used. This makes it useful in analyzing the ZMP as the coordinates for these are also in ground coordinates.

Equation 5.8 already calculates the location of all supporting points, such that the projection onto the ground simple requires a projection matrix M :

$$M = \begin{pmatrix} 1,0,0,0 \\ 0,1,0,0 \\ 0,0,0,0 \\ 0,0,0,0 \end{pmatrix} \quad (5.14)$$

Then:

$$\text{leftsupport}_G := \{P_{\mathcal{P},l} \cdot M | l \in \text{lefthull}\} \quad (5.15)$$

$$\text{rightsupport}_G := \{P_{\mathcal{P},r} \cdot M | r \in \text{righthull}\} \quad (5.16)$$

Where lefthull and righthull are the set of the points $P_{\mathcal{P},i}$ (see Eq. 5.8) of the convex hull of the feet in the parallel coordinate system. The total support polygon of the robot is

$$\text{support}_{\mathcal{G}} := \begin{cases} \text{leftsupport}_{\mathcal{G}} & \text{if } \text{left.contact} = 1, \text{right.contact} = 0 \\ \text{rightsupport}_{\mathcal{G}} & \text{if } \text{left.contact} = 0, \text{right.contact} = 1 \\ \text{conv}(\text{leftsupport}_{\mathcal{G}} \cup \text{rightsupport}_{\mathcal{G}}) & \text{if } \text{left.contact} = 1, \text{right.contact} = 1 \\ \{\emptyset\} & \text{otherwise} \end{cases} \quad (5.17)$$

However, performing an analysis of the CoM and ZMP with respect to $\text{support}_{\mathcal{G}}$ requires, as stated in Section 3.7.2, knowledge about the robot's planned motion. An example of this is shown in Figure 5.5 It shows the support polygon of the left foot along with that of the right foot, if it was on the ground. The large green area highlights the convex hull of the total potential support of the feet. Notably both the CoM and ZMP lie outside the support polygon of the left foot. However, this is because the right foot of the robot is becoming the swing foot and the weight of the robot is being shifted from the left to the right foot mid-step.

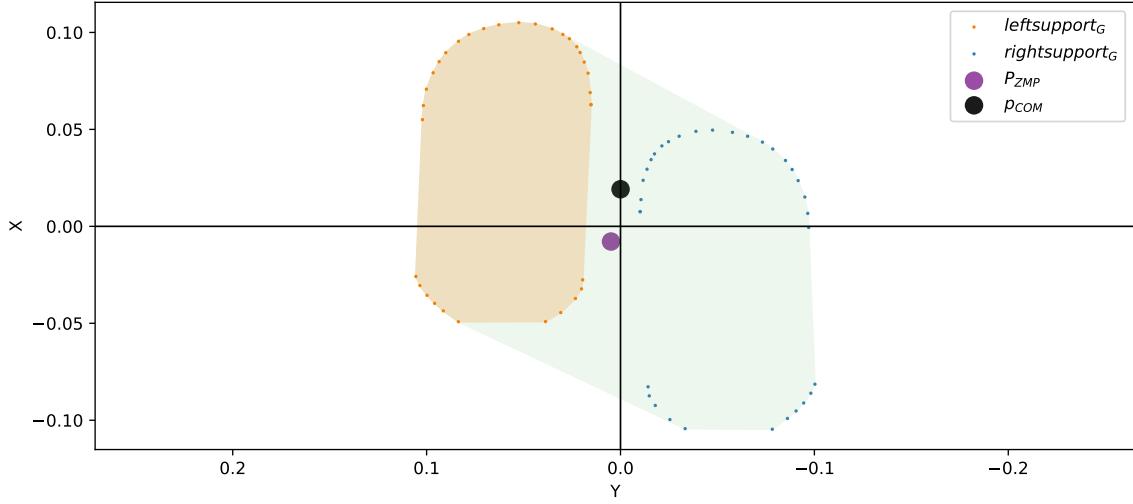


Figure 5.5: Example of the support of HULKs-NAO4 in one frame

At this moment the robot is technically falling, but because it is planning on catching itself with the right foot, it cannot necessarily be considered unstable.

To this end this thesis uses the convex hull of the union of both feet as the support polygon. The assumption is, that if the CoM and ZMP stay within this polygon, even if both feet aren't placed on the ground, that the robot can prevent itself from falling by simply placing the swing foot down.

5.7 Feature Extraction

Hulahoop is started before the NAO is placed on the field (see Sec. 4.2.2) and isn't stopped until after the testing is concluded. This causes the data to contain sections of frames which are irrelevant to the analysis in this thesis. This occurs both before and after the sections during which the robot walks. While the number of falls itself is a metric, the data from these moments must be removed from the rest of the analysis, as the interest in this thesis lies in the robot walking and not the process of standing back up.

Under normal walking circumstances the position of the robot's knee and hip pitch motors move as shown in Figure 5.6. The figure shows regular bending of the knees and movements of the hip joints. The highlighted area corresponds to the values from Table 5.1. To demonstrate how this section of data correlates with regular walking, the frame $t_k = 59541$ ms as been marked with a dashed line within the highlighted area. During this frame the left hip has a large negative angle which indicates it is rotated such that the foot moves to the front of the body (see Fig. 2.2). Meanwhile, the right one has a lesser negative angle meaning it is rotated to be more in line with the body than the left leg. This frame approximately correlates to the time $t_{L,support}$ in Figure 2.7. Such features can be identified everywhere in the regular walking cycle.

In irregular situations these cyclical patterns are broken. Figure 5.7 shows these three such situations. These areas are exemplified by irregular positions of the legs and rotation of the torso. The left and right highlighted areas are before and after the robot is placed on the field

for testing and the middle contains an instance in which the robot fell. The information of these irregular positions of the torso and legs is used to remove these sections from the data.

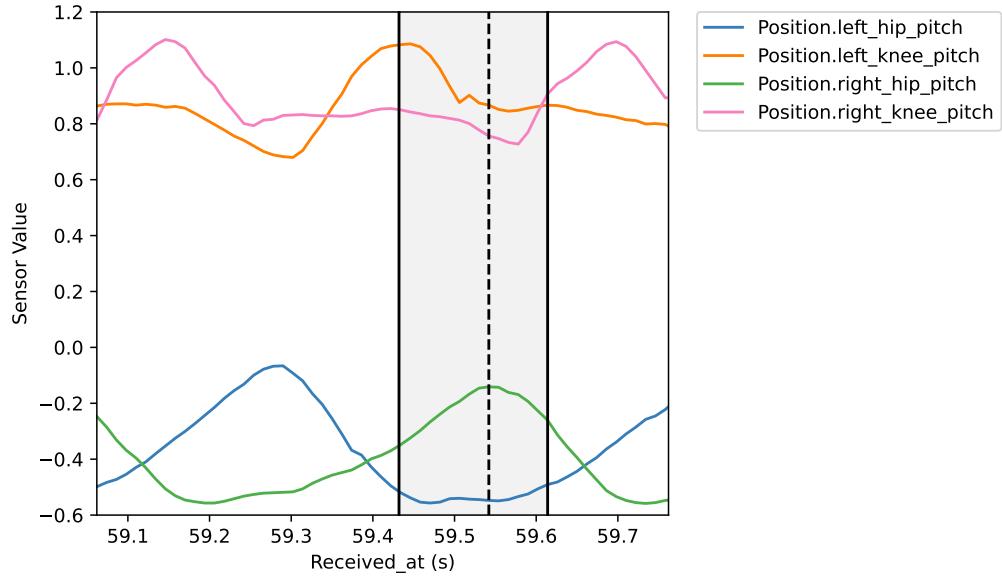


Figure 5.6: Knee- and Hip-pitches of B-Human-NAO1

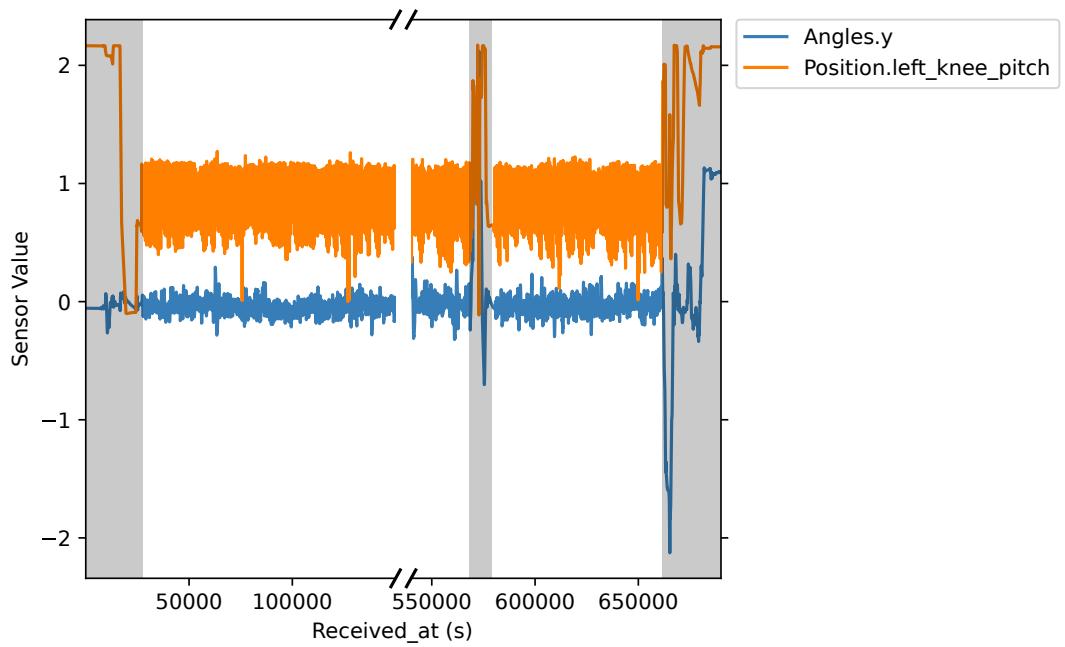


Figure 5.7: Sections of NaoDevils-NAO3 to be removed

5.7.1 Cropping the Start

To crop the data to the walking start, the timestamp at which the robot starts walking is found. The steps for this are based on the fact that HULKs, B-Human, and Nao Devils have a *look-around* motion as part of their robotics code which is completed before the robot starts walking. During look-around motion in which the robot looks to the left and right of itself before turning its head back to the center. To find start of walking following markers are found within the data under the constraint that they occur after each other in time:

1. k_{stiff} , the frame when $Stiffness.left_knee_pitch_k > 0$, which occurs when the robot is brought into the stiff state (see Sec. 4.2.2)
2. $k_{l1} > k_{stiff} + 60$, when $|Position.head_yaw_k| > 1$. The absolute value is used so that the order of directions in which the robot turns the head does not matter.
3. $k_{c1} > k_{l1}$, when $|Position.head_yaw_k| < 0.1$ which indicates the robot turning its head back to the center.
4. This is repeated for $k_{l2} > k_{c1}$ and $k_{c2} > k_{l2}$ to mark points where the robot turns its head in the other direction.

These timestamps are marked in Figure 5.8

All data from the start of hulahoop at t_0 until $t_{k_{c2}}$ is then removed from the dataframe. The timestamps of the data are remapped by $t_k = t_k - t_{k_{c2}}$ so that the “Received_at” value of the first frame of the remaining starts at 0.

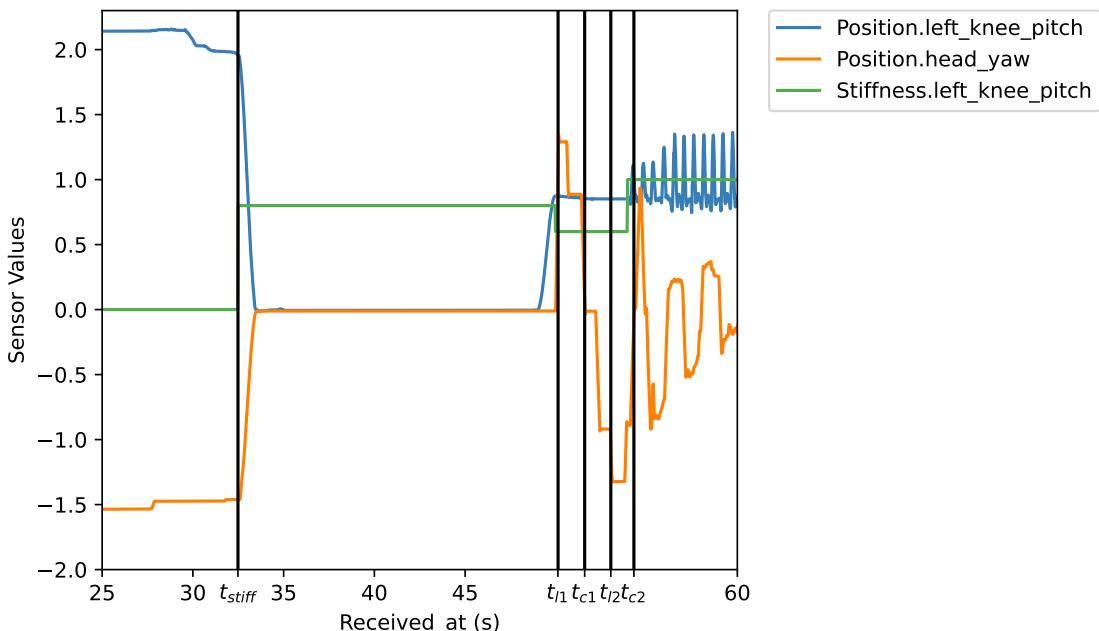


Figure 5.8: Irrelevant data from FastHULKs-NAO3 before walking start

5.7.2 Removing of Falls

Falls are found in the data based on the Angle.y of the torso and the pressure of the feet on the ground. After finding a fall it is also necessary to find the time when the robot has sat up and then stood up again and continued walking. Since one testing scenario may have multiple falls it is necessary to go through the data iteratively. The procedure is as follows:

1. Initialize $\text{fallcount} = 0$
2. Find k_{fall} where $\text{left.contact}_k, \text{right.contact}_k = 0$ and $|\text{Angle.y}_k| > 1$
3. If $k_{\text{fall}} > 600000$ or nonexistent skip to Step 9. Any fall more than 10 minutes since the start is irrelevant.
4. Find $k_{\text{sitting}} > k_{\text{fall}}$ where $\text{left.contact}_k, \text{right.contact}_k = 1$
 $\text{Position.left_knee_pitch}_k > 2$
5. Find $k_{\text{walking}} > k_{\text{sitting}}$ where $\text{Position.left_knee_pitch}_k < 1$
6. Remove the frame range $[k_{\text{fall}}, k_{\text{walking}}]$ from the dataframe, $\text{fallcount}+ = 1$.
7. For $t_k > t_{k_{\text{walking}}}$ remap with $t_k = t_k - t_{k_{\text{walking}}}$ so that the timestamps remain continuous.
8. Repeat steps 2-7.
9. Remove all data with $t_k > 600000$ from the dataframe. Store the value of fallcount .

This process ensures that exactly $K = 50000$ frames equaling 10 minutes of data are left in the dataframe. Figure 5.9 shows a section of data where a robot fell with the relevant timestamps marked.

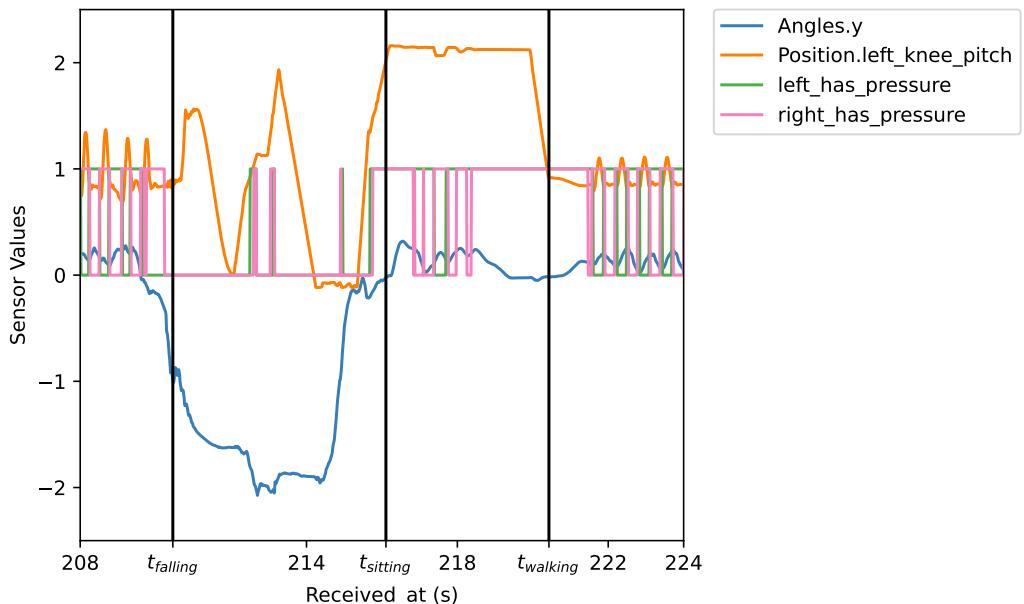


Figure 5.9: Section from FastHULKs-NAO3 where robot fell

5.8 Issue In Data Gathering

It is important at this point that the data for FastHULKs-NAO2 and NaoDevils-NAO2 do not include data for 10 minutes of walking, or 50000 frames. Over the course of three attempts, each these combinations of walking engine and robot always lead to instability and falls, from which the robot was not able to or simply did not stand up from again. Therefore, the attempts with the longest time frame of information are used here. For FastHULKs-NAO2 this was 48056 frames or roughly 9 minutes and 37 seconds, and for NaoDevils-NAO2 this was only 27147 frames or 6 minutes and 26 seconds. For calculations K is adjusted accordingly.

6 Data Analysis

This chapter consists of a qualitative analysis of the various evaluation categories discussed in Section 3.7, with regard to the collected data. It discusses interesting findings in regard to the relationship of evaluation categories to each other and to the hardware and walking engines that were tested. Finally, it also mathematically defines and calculates metrics with which the final evaluation of the walking engines is made. The metrics used, all assume an optimal value of 0 or close to it, meaning the larger the number the worse the walking quality. This is not further specified in each metric definition.

As mentioned in Section 5.8 the results of FastHULKs-NAO2 and NaoDevils-NAO2 are based on incomplete data. These two scenarios are still analyzed in this thesis, but for all intents and purposes the walking performance of these combinations of robots and walking engines is considered atrocious. In the tables in this chapter the corresponding values are marked with a * as a reminder to this fact.

6.1 Falling

As stated in Section 3.1, falling is generally detrimental to a robot's performance. Any time that it takes for a robot to stand back up during a RoboCup match, is time spent that it isn't actively attempting to score goals.

$$\text{Metric 1: } fallcount \quad (6.1)$$

The results of this metric are shown in Table 6.1

	HULKs	FastHULKs	B-Human	Nao Devils
NAO1	0	2	0	5
NAO2	0	1*	0	1*
NAO3	0	1	0	1
NAO4	0	0	0	9

Table 6.1: Metric 1: Number of falls, *based on less than 10 minutes of data

As mentioned in Section 4.2.2 the parameters that make up FastHULKs are not used by the HULKs team as they lead to instability. In comparing HULKs and FastHULKs, the results are thus as expected, with HULKs showing zero falls for any robot and FastHULKs showing some. In the case of FastHULKs-NAO2, the single fall that occurred resulted in the testing scenario having to be ended early.

Also expected is the regular falls of the Nao Devils. As stated in Chapter 1 the Nao Devils were chosen for this thesis based on the general observation that their robots regularly fall. Even though it is known that Nao Devils robots fall during games, the robot falling 9 times during the NaoDevils-NAO4 testing scenario was unexpectedly high.

6.2 Center of Mass (CoM)

Although the static stability of the robot is not analyzed in this thesis, the position of the CoM relative to the robot still provides interesting insights into the differences of the walking engines. Figure 6.1 shows scatter plots of the X and Y axes position of the CoM in ground coordinates for each walking engine. There is a distinct similarity between HULKs and FastHULKs, which

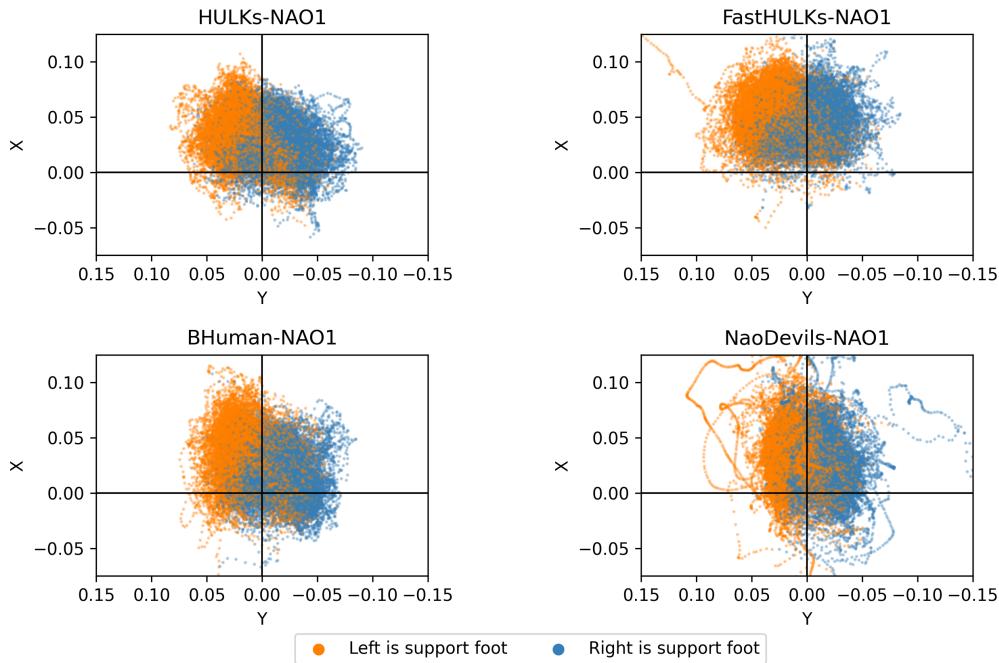


Figure 6.1: Comparison of CoM of the walking engines on NAO1

stems from the fact that the software of these two walking engines is the same. Notably though the CoM of FastHULKs is shifted further forward than HULKs, which is to be expected from the torso tilt adjustments that are part of the FastHULKs parameters. The overall shape of the HULKs walking engine is similar to that of B-Human, with large clusters (see Fig. 6.2) for the left and right foot at approximately ± 5 cm along the Y axis. Nao Devils looks drastically different, the CoM being clustered much closer to $y = 0$ while being spread out further along the X axis.

Additionally, several outlying trails of points can be seen outside the main cluster (see Fig. 6.1). While the few outliers seen in B-Human are all situations in which the robot was able to recover, demonstrated by B-Human having 0 falls, the outliers seen in the Nao Devils scatter plot may indicate points in time when the robot was extremely unstable and close to falling. Notably these patterns occur regardless of the robot chosen, and are approximately equal for each walking engine. A similar trail of outliers is seen for FastHULKs, which is verified to be the few frames before one of the falls that the robot experienced in this scenario.

When comparing the CoM for the HULKs walking engine with different robots the first differences in robot hardware are visible. This is shown in Figure 6.3. NAO2, which was stated to be considered the robot with the most hardware deterioration int Section 4.2.2, shows the most visible outliers of these four robots. This may be the result of loose joints causing more backlash

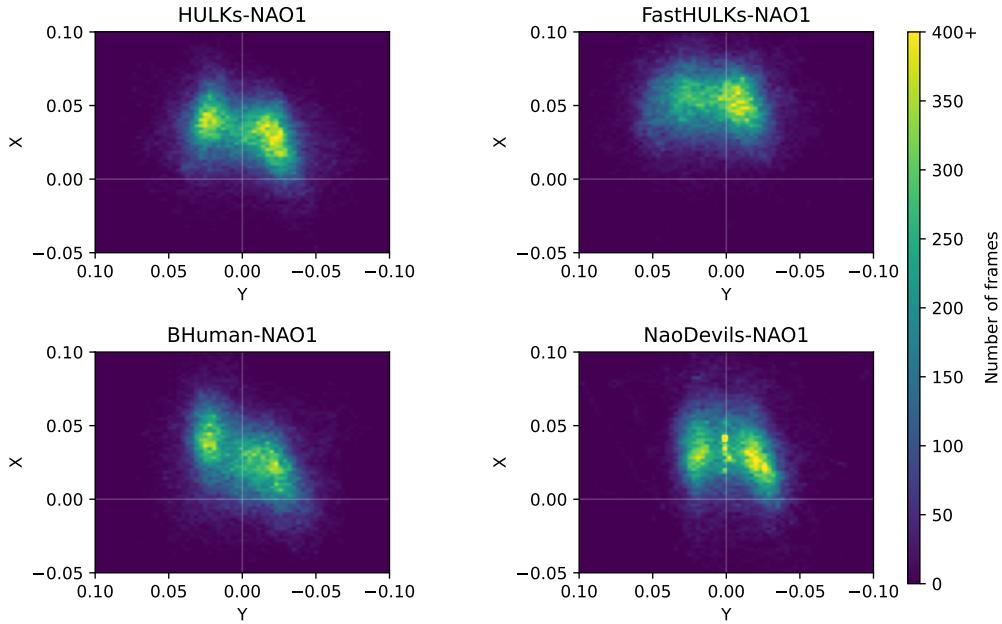


Figure 6.2: Heatmap of CoM of the walking engines on NAO1

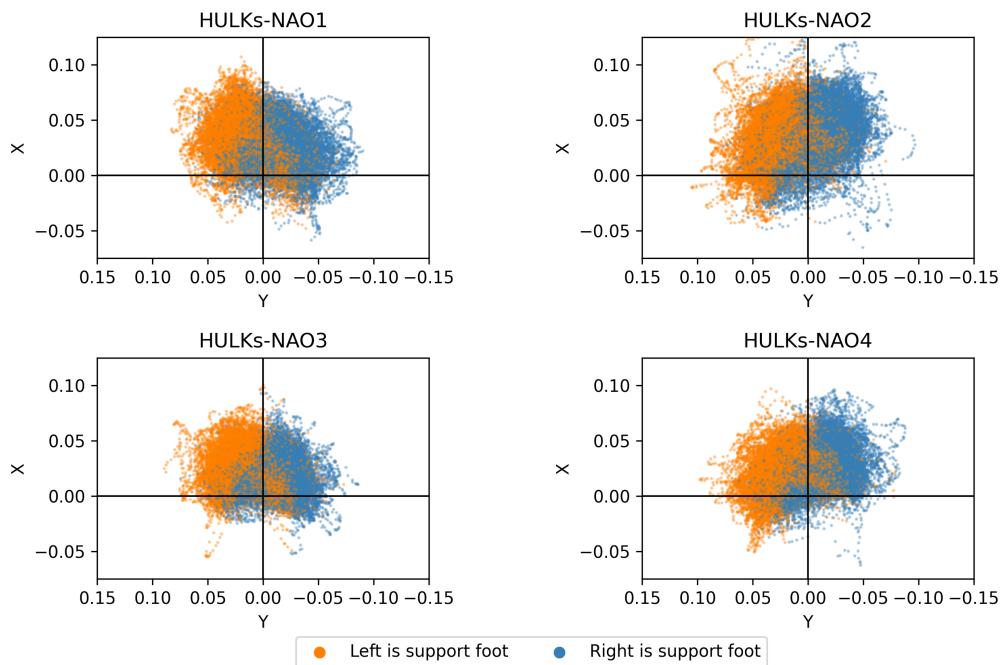


Figure 6.3: Comparison of CoM of the HULKs walking engine on NAO1-4

in the robot as it walks, leading to the CoM shifting unpredictably. However, while NAO3 does not have a lot of outliers, NAO4 does, even though these two robots were purchased at the same time. This indicates that in even the few months since their purchase date these robots have received uneven usage and the hardware has deteriorated in unequal ways. A comparison of the CoM of two brand-new robots must be conducted to see if the shape of the CoM scatter plot is a direct result of the hardware or whether other factors play into it.

6.3 Zero Moment Point (ZMP)

As stated in Section 5.5 this thesis analyzes the ZMP with regard to the support polygon of both feet. An initial approach to finding summarizing value for this involves determining the average distance of the ZMP to the support polygon, with the distance equaling 0 if the ZMP is inside the support polygon. The equations for that are as follows:

$$D_{avg} = \frac{1}{W} \sum_{k \in K} D_k \quad (6.2)$$

$$D_k = \begin{cases} 0 & \text{if } P_{ZMP,k} \in support_{G,k} \\ \min(\text{dist}(P_{ZMP,k}, \overline{AB})) & \text{otherwise} \end{cases} \quad (6.3)$$

$$W = \sum_{k \in K} i \quad \text{where } i = \begin{cases} 0 & \text{if } D_k = 0 \\ 1 & \text{otherwise} \end{cases} \quad (6.4)$$

Here D_{avg} is the average distance of the ZMP outside the support polygon, calculated using the distance D_k to the support polygon and W the number of frames that the ZMP was outside the support polygon. D_k in turn is the minimum distance of the ZMP to any line \overline{AB} in the boundary of $support_{G,k}$ (see Eq. 5.17). However, this initial approach provides values which are too similar between testing scenarios to offer any interesting insight. The average distance for each scenario is on the order of a few micrometers.

Instead, only the values of W are considered, the number of frames during which the ZMP is outside the convex hull of the support polygon of both feet. The results of this are shown in Table 6.2 and Figure 6.4. These values indicate the number of frames during which the ZMP

	HULKs	FastHULKs	B-Human	Nao Devils
NAO1	120	820	356	889
NAO2	233	1169*	216	489*
NAO3	23	156	36	212
NAO4	32	83	10	886

Table 6.2: Number of frames (W) where the ZMP was outside support polygon of both feet

was not contained in the convex hull of the support polygon of both feet. As was stated in Section 3.1.1, the ZMP represents the point where, if the robot stops instantaneously walking, a normal force must be applied to prevent it from falling over. Under the assumption that the robot is able to place its swing foot down on the ground, the instance before it stops walking, it means that as long as the ZMP is contained within the support polygon of both feet, the robot does not fall over. However, the values from Table 6.2 show instances where the ZMP is outside the support polygon. This means that even if the robot were able to instantaneously place its swing foot down, it would not be able to stabilize itself if it instantaneously stopped walking.

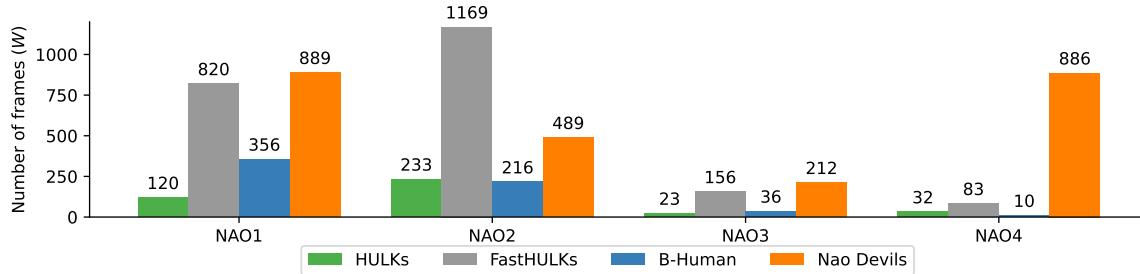


Figure 6.4: Number of frames (W) where the ZMP was outside support polygon of both feet

While this is by no means a perfect analysis of dynamic stability, it does provide some valuable insight into the differences between the walking engines.

For the case of FastHULKs and Nao Devils this data approximately correlates with the number of falls the robot's experienced, with an increased value of W indicating unstable walking and thus more falls. The pattern breaks down when looking at B-Human-NAO1, B-Human-NAO2, and HULKs-NAO2, which each had more frames of instability than NaoDevils-NAO3, yet in these scenarios the robots never fell. This is most likely due to an early enough detection of instability of the robot by the walking engine and the ability to properly compensate with stabilizing steps. Further testing with intentional pushing of the robot would provide more evidence to each walking engine's ability to compensate for instability.

Finally, it is interesting to note that NAO3 and NAO4 consistently, except for the case of NaoDevils-NAO4, fewer frames of the ZMP being outside the support polygon than NAO1 and NAO2. This could be due to differences in hardware deterioration, since NAO1 and NAO2 are approximately 2 years older than NAO3 and NAO4, and have thus been used for more testing and competitions. Further testing might provide evidence to support that tracking the value of W over the course of the robot's usage may be a way to objectively determine that the hardware has deteriorated. Another valuable comparison would be to thoroughly calculate the backlash of the joints of the robots and determine if this has a correlation with the instability as determined by the ZMP.

6.4 Double Support Phase Length

As stated in Section 3.7 the length of the double support phase cannot be analyzed objectively with respect to some optimal value as the optimal length of the double support phase is unknown for the NAO robot and the different walking engines.

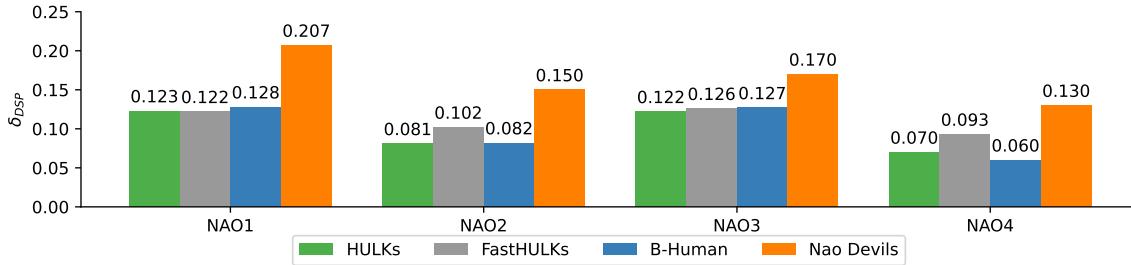
Rather than calculating the double support phase as a fraction of each walk cycle, the average double support phase length over the entire data set is considered. The equation for this is:

$$\delta_{DSP} = \frac{1}{K} \sum_{k \in K} 1 - |left.contact_k - right.contact_k| \quad (6.5)$$

which calculates the proportion of frames where both feet are in contact with the ground. For this the boolean values from Section 5.3 are used, such that the calculation inside the sum evaluates to 1 if both feet are in contact with the ground and 0 if only one foot is in contact with the ground. The results for this are shown in Table 6.3.

The length of the calculated double support phase relies entirely on data accuracy from the FSR sensors. The actual length of the double support could in reality be identical between each

	HULKs	FastHULKs	B-Human	Nao Devils
NAO1	0.123	0.122	0.128	0.207
NAO2	0.081	0.102*	0.082	0.150*
NAO3	0.122	0.126	0.127	0.170
NAO4	0.070	0.093	0.060	0.130

Table 6.3: Average Double Support Phase Length

Figure 6.5: Average Double Support Phase Length

robot for a walking engine. However, if the FSRs of one robot are overall more sensitive than those of another, the pressure needed to consider a foot supporting is reached earlier, and the double support phase becomes longer. The responsiveness or accuracy of the FSRs were not tested in this thesis, so this correlation between hardware and double support time based on FSR values is not proven.

The most interesting result from a comparison of the double support phase between walking engines is the comparison of the Nao Devils to the other three. The Nao Devils have a longer double support phase than any of the other walking engines for any given robot.

6.5 Energy Consumption

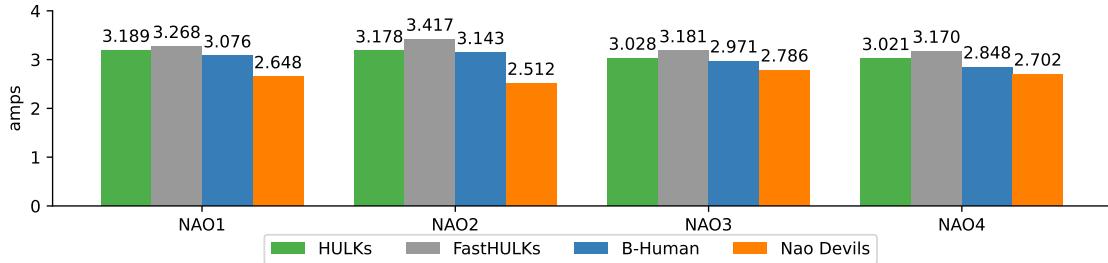
Due to the fact that the batteries of the NAOs all have the same nominal voltage, calculating the average energy consumption in Watts would only result in a scaled version of the average current. Therefore, the metric for the average energy consumption is given in terms average current of the battery. The current is positive while the robot is charging and negative while it is not. Since the battery is not being charged while the robot is walking, the negative of the current is used, such that this metric continues to have an optimal value of zero or close to.

$$\text{Metric 2: } \frac{1}{K} \sum_{k \in K} -\text{Battery.current}_k \quad (6.6)$$

The results of this are shown in Table 6.4 and Figure 6.6

For every robot the highest average energy consumption was from FastHULKs, followed in decreasing order by HULKs, B-Human, and NaoDevils. It means that the walking engine and parameters of the HULKs software can possibly be improved in order to reduce energy consumption. However, there also seems to be a limit to how much can be gained from trying to limit the energy consumption of the robot. Although the Nao Devils have the lowest current draw of all the teams, they also have the highest number of falls. Instead, the difference in HULKs and B-Human shows a more realistic comparison of the energy that can be saved. Both

	HULKs	FastHULKs	B-Human	Nao Devils
NAO1	3.189	3.268	3.076	2.648
NAO2	3.178	3.417*	3.143	2.512*
NAO3	3.028	3.181	2.971	2.786
NAO4	3.021	3.170	2.848	2.702

Table 6.4: Metric 2: Average battery current

Figure 6.6: Metric 2: Average battery current

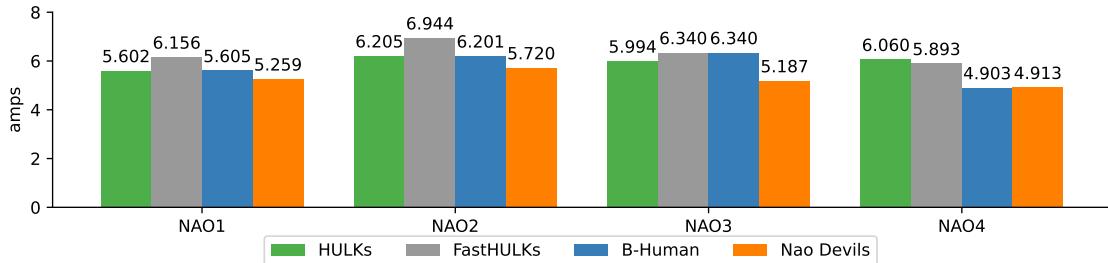
of these walking engines suffered from 0 falls, but B-Human managed to do this while consuming less energy, which is considered a higher quality of walking.

Additionally, the peak energy consumption is of interest. This is simply calculated by:

$$\text{Metric 3: } \max(-\text{Battery.current}_k) \forall k \in K \quad (6.7)$$

The results are shown in Table 6.5 and Figure 6.7. The values approximately follow the same trend as those of metric 2, with a minor distinction that the scenario B-Human-NAO4 had the lowest peak energy consumption.

	HULKs	FastHULKs	B-Human	Nao Devils
NAO1	5.602	6.156	5.605	5.259
NAO2	6.205	6.944	6.201	5.720
NAO3	5.994	6.340	6.340	5.187
NAO4	6.060	5.893	4.903	4.913

Table 6.5: Metric 3: Peak battery current

Figure 6.7: Metric 3: Peak battery current

Critical is that both the peak energy consumption and average energy consumption should stay within the limits of what the battery is capable of handling. While no walking engine triggers the surge protection of the NAO's battery, it is notable that all teams have an average and peak energy consumption above the 2 amps that the NAO battery is rated for.

6.6 Heat Production

The joints of the NAO do not heat up equally. This is seen in Figure 6.8 and 6.9, which show that specifically the hip-, knee-, and ankle-pitch motors of both the left and the right leg heat up the most. In the case of BHUMAN-NAO1, a few other motors heat up some amount, namely the pitch motors of the left and right shoulder and the head pitch. For FastHULKs-NAO3 these also heat up, along with the *left_hip_yaw_pitch* motor and the left and right ankle roll motors. However, the temperature of most of these after 10 minutes of walking is well under 48°C. For

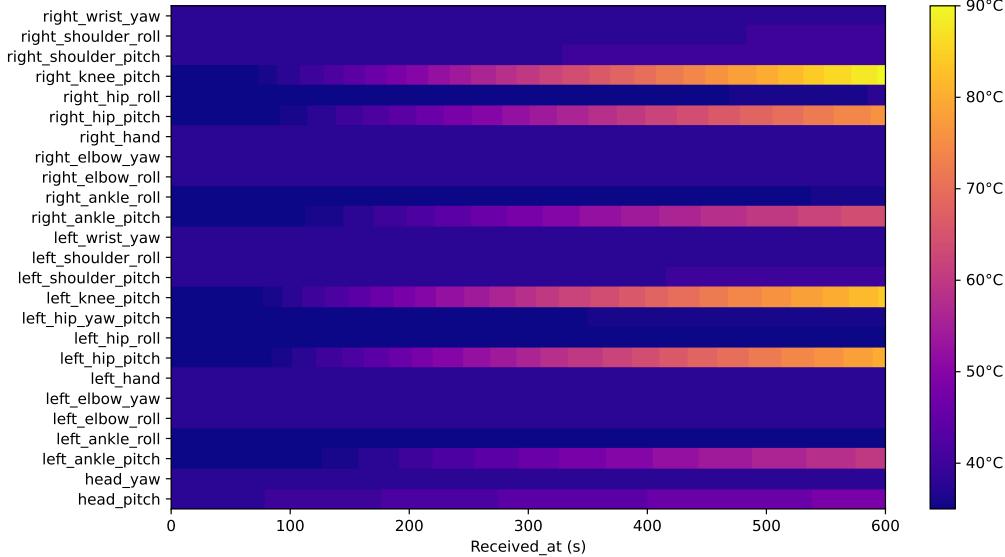


Figure 6.8: Temperatures of motors from BHUMAN-NAO1

all test scenarios it was the case that the pitch motors named above heated up the most. These are the most relevant in walking, especially in stepping forwards as the kinematic chain formed by these limbs are what allow the placement of the feet far in front and behind the robot torso. Therefore, a metric is defined which considers only these motors:

$$\text{Metric 4: } \frac{5000}{|\Omega|K} \sum_{i \in \Omega} \text{Temperature}.i_K - \text{Temperature}.i_1 \quad (6.8)$$

$$\Omega = \{\text{left_hip_pitch}, \text{right_hip_pitch}, \\ \text{left_knee_pitch}, \text{right_knee_pitch}, \\ \text{left_ankle_pitch}, \text{right_ankle_pitch}\} \quad (6.9)$$

where K is the total number of frames, such that $\text{Temperature}.i_L$ is the ending temperature of the respective joint. The factor of 5000 is the number of frames in one minute. Equation 6.8

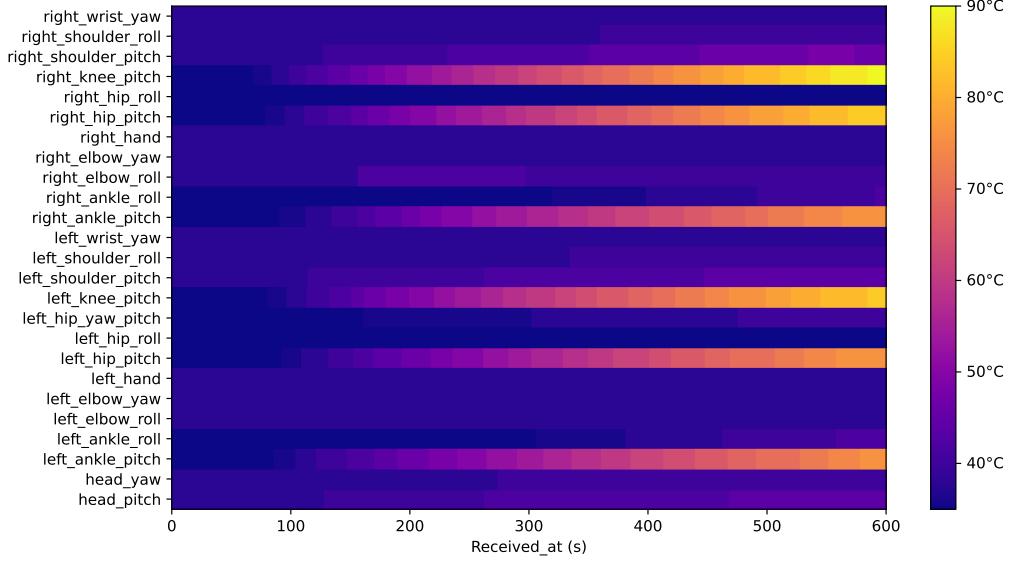


Figure 6.9: Temperatures of motors from FastHULKs-NAO3

thus gives the average temperature gain in $^{\circ}\text{C} / \text{minute}$ of joints in Ω . This metric is defined like this to allow for a comparison of heat gain even for the two scenarios where less than 10 minutes of data were gathered. The results are shown in Table 6.6 and Figure 6.10. The trend

	HULKs	FastHULKs	B-Human	Nao Devils
NAO1	5.10	5.23	4.87	3.67
NAO2	4.97	5.31*	4.90	3.20*
NAO3	5.27	5.43	4.88	4.20
NAO4	5.03	5.00	4.53	3.93

Table 6.6: Metric 4: Average temperature gain in $^{\circ}\text{C} / \text{minute}$ of joints in Ω

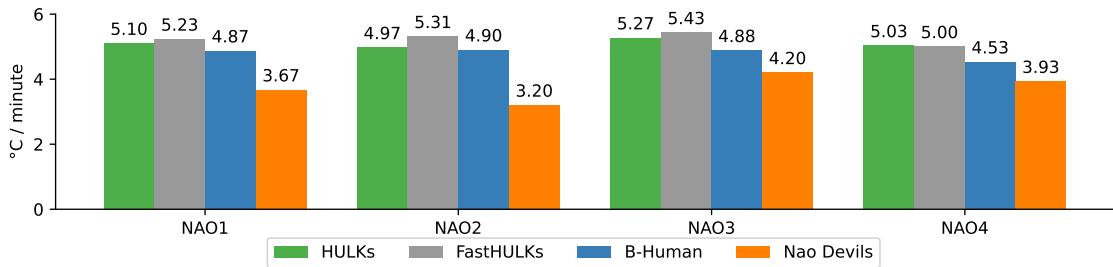


Figure 6.10: Metric 4: Average temperature gain in $^{\circ}\text{C} / \text{minute}$ of joints in Ω

of the metric 4 generally follows that of metric 2, i.e. that FastHULKs is the worst and Nao Devils is best performing walking engine. This makes sense as the temperature values of the NAO are simulated based on the current consumption as explained in Section 3.4. An outlier

to this trend is FastHULKs-NAO4, although the reason why this robot heats up less, is not determinable from the data without a deeper analysis of the walking patterns of the robot.

While B-Human has an average temperature gain for the 6 leg joints that is lower than HULKs, for all except NAO4 the maximum temperature of any single leg joint is equal to or higher than that of both hulks and FastHULKs (see Tab. 6.7). It was expected that the robots with high joint temperatures would eventually collapse due to HAL potentially reducing the stiffness in the motors, however, this did not occur. It is unknown exactly when and how HAL reduces the stiffness in the motors, and there exists no data to show that this would be reflected in the sensor data from LoLA. For the case of BHuman-NAO3 either HAL didn't reduce the stiffness in the 104°C motor, or the B-Human walking engine was able to compensate any momentary loss of stiffness through balance adjustments of the whole robot. The maximum temperature

	HULKs	FastHULKs	B-Human	Nao Devils
NAO1	90	88	90	82
NAO2	92	94*	94	66*
NAO3	98	90	104	90
NAO4	90	94	84	84

Table 6.7: Maximum temperature in °C of the hottest joint in Ω

of NaoDevils-NAO2 was 66°C when the test was concluded due to the robot's inability to get back up. During the testing the assumption was that the robot had simply gotten too hot. However, given that this temperature is well below that of other testing scenarios, it is unclear what caused this combination of walking engine and robot to repeatedly fail. Finding this source of the issue would require a deeper investigation into the Nao Devils software along with tests on other hardware to see if this is reproducible in some way.

6.7 Torso Stability

As stated in Section 3.7 the camera stability is judged through the stability of the torso, where better stability is defined by a decrease in the movement of the torso. The three main aspects of this are the acceleration in the Y and Z axes, and the rotation around the X axis of the robot. Rotations around the Y and Z axes can be compensated by the head's pitch and yaw motors respectively. Acceleration of the torso along the X axis does not result in the camera changing the direction and rotation of its field of view and is considered less detrimental than motion blur caused by a rotating torso or lateral and vertical acceleration. Thus, the following metrics are defined:

$$\text{Metric 5: } \frac{1}{K} \sum_{k \in K} |Gyroscope.x_k| \quad (6.10)$$

$$\text{Metric 6: } \frac{1}{K} \sum_{k \in K} |Accelerometer.y_k - y_{\mathcal{R}}| \quad (6.11)$$

$$\text{Metric 7: } \frac{1}{K} \sum_{k \in K} |Accelerometer.z_k - z_{\mathcal{R}}| \quad (6.12)$$

where $y_{\mathcal{R}}$ and $z_{\mathcal{R}}$ are the acceleration due to gravity in the robot coordinate system calculated using a homogeneous vector based on g :

$$(x_{\mathcal{R}}, y_{\mathcal{R}}, z_{\mathcal{R}}, 0) = T_{\mathcal{P}}^{\mathcal{R}} * (0, 0, -g, 0) \quad (6.13)$$

Metric 5 is the rotational speed of the torso around the X axis, metrics 6 and 7 are the absolute acceleration of the robot torso in the Y and Z axis compensated with gravitational acceleration. The results of these metrics are shown in Tables 6.8,6.9 and 6.10.

There is no clear trend when comparing the different robots and walking engines for the three metrics that make up the torso stability. The only exception to this are the values of the Nao Devils, which, like was the case for energy consumption and heat production, are lower and therefore better than those of the other three walking engines. This comes as a surprise, given the general instability observed for the Nao Devils and the number of falls the robots using the Nao Devils walking engine experienced. However, what this clearly shows, is that torso stability does not correlate to overall stability or the ability to not fall.

For NAO2,NAO3, and NAO4 metrics 4 and 5 saw an improvement and metric 6 got worse when comparing HULKs to FastHULKs. This indicates that although FastHULKs consumes more energy and heats up the robot more, it is possible to gain some camera stability from the set of parameters that make up FastHULKs.

	HULKs	FastHULKs	B-Human	Nao Devils
NAO1	0.420	0.425	0.474	0.269
NAO2	0.431	0.428*	0.383	0.217*
NAO3	0.383	0.381	0.388	0.224
NAO4	0.433	0.404	0.392	0.262

Table 6.8: Metric 5: Average absolute gyroscope x value

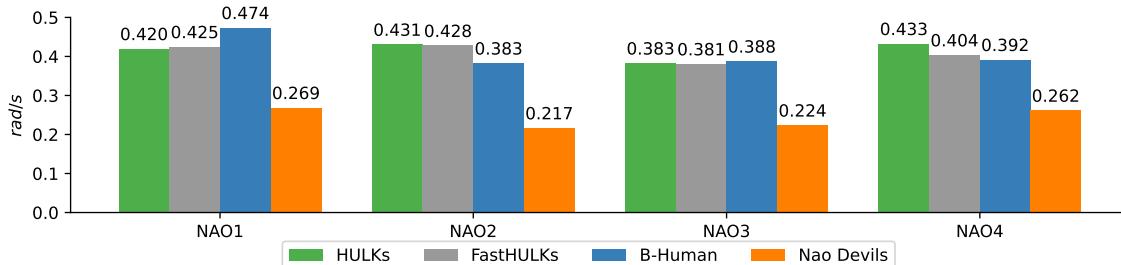


Figure 6.11: Metric 5: Average absolute gyroscope x value

	HULKs	FastHULKs	B-Human	Nao Devils
NAO1	2.067	2.097	2.153	1.752
NAO2	2.064	2.072*	2.122	1.541*
NAO3	2.022	2.068	2.096	1.794
NAO4	1.966	1.960	2.018	1.740

Table 6.9: Metric 6: Average absolute accelerometer y value compensated for gravity

When analyzing torso stability for HULKs, FastHULKs, and B-Human the difference in hardware are once again visible. For all situations, except NAO4-HULKs, NAO3 and NAO4 consistently showed more stability than NAO1 and NAO2. This points again to differences in the looseness of joints leading to different amount of backlash in the robot. Nao Devils is an outlier, as the highest torso stability, i.e. the lowest rotation and acceleration of the torso, were demon-

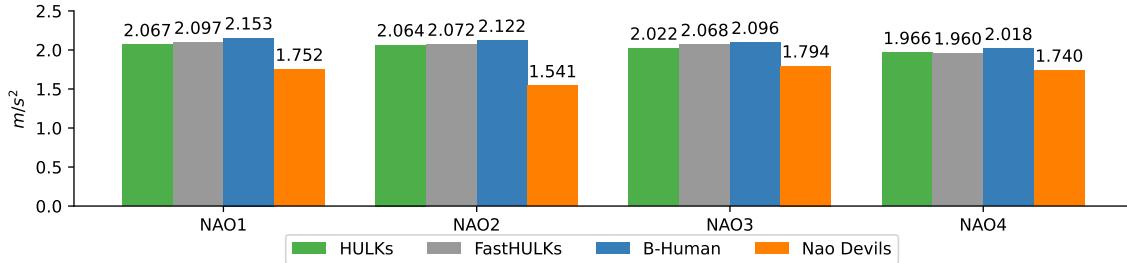


Figure 6.12: Metric 6: Average absolute accelerometer y value compensated for gravity

	HULKs	FastHULKs	B-Human	Nao Devils
NAO1	2.430	2.430	2.455	2.157
NAO2	2.517	2.548*	2.499	1.815*
NAO3	2.106	2.253	2.320	1.969
NAO4	2.237	2.314	2.276	1.987

Table 6.10: Metric 7: Average absolute accelerometer z value compensated for gravity

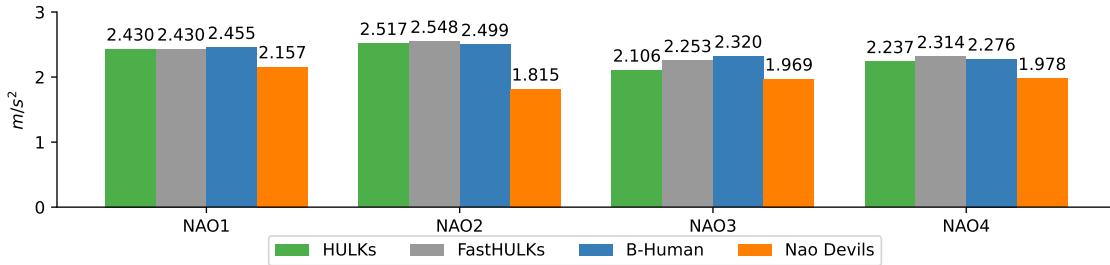


Figure 6.13: Metric 7: Average absolute accelerometer z value compensated for gravity

strated by NAO2, the robot considered to have the most hardware deterioration. This would indicate that the Nao Devils walking engine may include a stabilization specifically for worn out hardware. A code analysis or further testing on other robots could be used to verify this.

6.8 Summary

Overall, based on the metrics defined, save for the number of falls, the walking engine of the Nao Devils outperforms the other walking engines in every metric. On average, it keeps the torso the steadiest to allow for the highest potential camera stability, while consuming less energy and producing less heat than the walking engines of HULKs and B-Human, regardless of the robot used. However, as pointed out, the Nao Devils testing scenarios also had the highest total number of falls by far. As was discussed in the introduction, these regular falls are an expected occurrence, as the Nao Devils have a much more stop-and-go approach to their walking. Robots will either walk slow and steady, or speed up to a point of instability, and fall over.

The increased energy consumption, heat production, and number of falls are evidence of why the parameters of FastHULKs are not employed by the HULKs team. The heat production and number of falls especially have an impact on hardware deterioration in the long run. However,

the comparison of these two scenarios shows that the metrics in this thesis are usable in analyzing the effects of parameter changes in a way that subjective human observation of walking can not. It is possible to quantify torso stability using this metric which replaces human judgement of the same.

As for the walking quality of the walking engines overall, when taking into account all metrics including the number of falls, no Pareto-optimal combination of walking engine and hardware is determinable. No single robot or walking engine consistently outperforms the rest. It is clear that in order to assign a metrics to the walking quality as a whole, a cost function which combines the individual metrics is needed. One example of this is a weighted sum of the individual metrics. However, in this first attempt at analyzing walking quality such the weights would only be arbitrarily chosen, thus a combined metric is not calculated in this thesis.

7 Conclusion

This thesis lays the groundwork for an intensive study of walking quality of a NAO robot in the context of the RoboCup. It provides a review of state-of-the-art categories for analyzing walking quality, and under the limitations set forth, defines and calculates metrics for a portion of these. An important finding of this thesis is that the metrics defined and evaluated in this thesis are not enough to obtain an absolute objective ranking of the walking engines of the three teams studied. The Nao Devils consistently have the lowest and therefore the best value for all metrics except the number of falls. B-Human had higher maximum joint temperatures than HULKs, but managed to consume less energy and have a lower average heat gain of the walking critical joint motors in the legs of the robot. While these metrics do cover important aspects of potential camera stability by means of torso stability, heat production, and energy consumption, other aspects critical to the RoboCup, such as speed and robustness against pushing, were not studied. It is clear that all aspects of walking quality must be used to gain a fully comprehensive understanding of walking.

The HULKs gain several insights into walking quality from this research. First, the summary of research in this field provides a concise reference point for further research into walking quality. Second, this thesis establishes an initial set of metrics for analyzing walking quality, along with a method of gathering the required data, both for the HULKs walking engine and for that of other teams. It also means that changes in walking quality can now be analyzed when changes in walking engine software occur. Third, while differences in hardware deterioration are not consistently noticeable, differences in the robot stability calculated through the ZMP and the torso stability do seem to show that older hardware inherently becomes unstable. This should be a focus point for further investigation. Finally, the evaluation of the metrics for the four walking engines shows that there is a range of walking engines and parameters that still allow for successful walking. This means that the HULKs should try out other ranges of parameters to further improve their walking quality and evaluate the metrics proposed in this thesis for these changes. This may allow for a finding of an entirely unique set of parameters that improve the walking quality drastically, for example by reducing energy consumption.

Outlook

Based on the gained insights there are many applications of this research which future work can pursue.

First, to gain an even deeper understanding of the differences in the walking engines, a larger data gathering setup must be designed which is not limited to the restrictions of the parameter tuning environment of the RoboCup. Using a camera based robot tracking system allows for the identification of the robot's position and speed without requiring access to the robotics code. Such a setup optimally also includes the ability to track each limb of the robot to determine slipping, and a system through which the robots can be pushed repeatably, so that the robustness against external forces is tested. New metrics based on the collected values can be defined which allow for an even more in-depth comparison of walking engines.

Second, the scope should be limited to just the HULKs team, as the motivation is for the HULKs to improve their walking. When no longer comparing the HULKs to other teams, the requirement of performing black-box analysis becomes obsolete and allows for a deeper analysis of the walking quality through access to the robotics code. This means that the position of the ZMP can be compared to the planned motion of the robot and deviations can be determined, even when only considering a single support foot. The location of the robot can be tracked by using the internal localization algorithms which allows for the calculation of speed and inefficiencies due to slipping.

Third, based on the scope which focuses on just the HULKs team, large amounts of data using different parameters can be gathered through simulations. This also allows for the testing on slightly different fields and simulating external forces, such that information is gathered about all aspects of walking quality. Through this enough data can be gathered to the point that an optimization algorithm can be applied to the gathered data, and sets of parameters which optimize the individual metrics are found. With enough data it may be possible to approximate gradients for the changes in parameters, which allows for the applicability of stochastic gradient descent. If a combined cost function for all metrics is defined, this method allows for the finding of a set of parameters which optimizes all aspects of the walking quality.

Bibliography

- [1] *Robocup Standard Platform League*. Website. (Accessed: October 23, 2023). URL: <https://spl.robocup.org>.
- [2] Bernhard Hengst. *rUNSWift Walk2014 Report - Robocup Standard Platform League*. Tech. rep. rUNSWift, 2014. URL: <https://cgi.cse.unsw.edu.au/~robocup/2014ChampionTeamPaperReports/20140930-Bernhard.Hengst-Walk2014Report.pdf>.
- [3] Oliver Urbann, Ingmar Schwarz, and Matthias Hofmann. “Flexible Linear Inverted Pendulum Model for cost-effective biped robots.” In: *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*. 2015, pp. 128–131. DOI: 10.1109/HUMANOIDS.2015.7363525.
- [4] Tadeusz Mikolajczyk et al. “Recent Advances in Bipedal Walking Robots: Review of Gait, Drive, Sensors and Control Systems.” In: *Sensors* 22.12 (2022). ISSN: 1424-8220. DOI: 10.3390/s22124440. URL: <https://www.mdpi.com/1424-8220/22/12/4440>.
- [5] Eiichi Yoshida. “Robots that look like humans: A brief look into humanoid robotics.” In: *Metode Science Studies Journal* 0.9 (2019), pp. 143–151. ISSN: 2174-9221. DOI: 10.7203/metode.9.11405. URL: <https://ojs.uv.es/index.php/Metode/article/view/11405>.
- [6] Jenna Reher and Aaron D. Ames. “Dynamic Walking: Toward Agile and Efficient Bipedal Robots.” In: *Annual Review of Control, Robotics, and Autonomous Systems* 4.1 (2021), pp. 535–572. DOI: 10.1146/annurev-control-071020-045021. URL: <https://doi.org/10.1146/annurev-control-071020-045021>.
- [7] Sujan Warnakulasooriya et al. “Bipedal Walking Robot-A Developmental Design.” In: *Procedia Engineering* 41 (Dec. 2012), pp. 1016–1021. DOI: 10.1016/j.proeng.2012.07.277.
- [8] HULKs. *HULKs - Robocup SPL team TUHH, Germany*. Website. (Accessed: October 23, 2023). 2023. URL: <https://hulks.de/>.
- [9] Thomas Röfer et al. “B-Human 2021 – Playing Soccer Out of the Box.” In: *RoboCup 2021: Robot World Cup XXIV*. Ed. by Rachid Alami et al. Cham: Springer International Publishing, 2022, pp. 302–313. ISBN: 978-3-030-98682-7.
- [10] Nao Devils. *Nao Devils TU Dortmund - RoboCup Standard Platform Soccer Team*. Website. (Accessed: October 23, 2023). URL: <https://naodevils.de>.
- [11] Mark E. Glickman. *The Glicko system*. Website. (Accessed: October 23, 2023). URL: <http://www.glicko.net/glicko.html>.
- [12] Arne Hasselbring. *RoboCup-SPL Ratings*. Database. (Accessed: October 23, 2023). 2023. URL: <https://github.com/RoboCup-SPL/Ratings>.
- [13] SoftBank Robotics Europe. *NAO - Documentation*. Website. (Accessed: October 23, 2023). 2021. URL: http://doc.aldebaran.com/2-8/home_nao.html.

Bibliography

- [14] “IEEE Standard for Information Technology—Portable Operating System Interface (POSIX) Base Specifications, Issue 7.” In: *IEEE Std 1003.1-2017 (Revision of IEEE Std 1003.1-2008)* (2018), pp. 1–3951. DOI: 10.1109/IEEESTD.2018.8277153.
- [15] Open Robotics. *URDF*. Website. (Accessed: October 23, 2023). 2023. URL: <http://wiki.ros.org/urdf>.
- [16] Sadayuki Furuhashi. *MessagePack*. Website. (Accessed: October 23, 2023). 2021. URL: <https://msgpack.org/index.html>.
- [17] Carlos André Dias Bezerra and Douglas Eduardo Zampieri. “Biped Robots: The State of Art.” In: *International Symposium on History of Machines and Mechanisms*. Ed. by Marco Ceccarelli. Dordrecht: Springer Netherlands, 2004, pp. 371–389. ISBN: 978-1-4020-2204-3.
- [18] Miomir Vukobratovic and Branislav Borovac. “Zero-Moment Point - Thirty Five Years of its Life.” In: *I. J. Humanoid Robotics* 1 (Mar. 2004), pp. 157–173. DOI: 10.1142/S0219843604000083.
- [19] Shuuji Kajita et al. “Biped Walking Pattern Generator allowing Auxiliary ZMP Control.” In: Nov. 2006, pp. 2993–2999. DOI: 10.1109/IROS.2006.282233.
- [20] Kemalettin Erbatur and Okan Kurt. “Natural ZMP Trajectories for Biped Robot Reference Generation.” In: *IEEE Transactions on Industrial Electronics* 56.3 (2009), pp. 835–845. DOI: 10.1109/TIE.2008.2005150.
- [21] Nicolas Van der Noot and Allan Barrea. “Zero-Moment Point on a bipedal robot under bio-inspired walking control.” In: *MELECON 2014 - 2014 17th IEEE Mediterranean Electrotechnical Conference* (2014), pp. 85–90. URL: <https://api.semanticscholar.org/CorpusID:12987843>.
- [22] Stéphane Bazeille et al. “Active camera stabilization to enhance the vision of agile legged robots.” In: *Robotica* 1 (Nov. 2015), pp. 1–19. DOI: 10.1017/S0263574715000909.
- [23] Timothée Habra et al. “Multimodal gaze stabilization of a humanoid robot based on reafferences.” In: *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*. 2017, pp. 47–54. DOI: 10.1109/HUMANOIDS.2017.8239536.
- [24] Ye Xie et al. “A Review: Robust Locomotion for Biped Humanoid Robots.” In: *Journal of Physics: Conference Series* 1487.1 (Mar. 2020), p. 012048. DOI: 10.1088/1742-6596/1487/1/012048. URL: <https://dx.doi.org/10.1088/1742-6596/1487/1/012048>.
- [25] Salman Faraji, Hamed Razavi, and Auke J. Ijspeert. “Bipedal walking and push recovery with a stepping strategy based on time-projection control.” In: *The International Journal of Robotics Research* 38.5 (2019), pp. 587–611. DOI: 10.1177/0278364919835606. eprint: <https://doi.org/10.1177/0278364919835606>. URL: <https://doi.org/10.1177/0278364919835606>.
- [26] Ke Wang, Hengyi Fei, and Petar Kormushev. “Fast Online Optimization for Terrain-Blind Bipedal Robot Walking With a Decoupled Actuated SLIP Model.” In: *Frontiers in Robotics and AI* 9 (2022). ISSN: 2296-9144. DOI: 10.3389/frobt.2022.812258. URL: <https://www.frontiersin.org/articles/10.3389/frobt.2022.812258>.
- [27] Albertus Hendrawan Adiwahono et al. “Push recovery controller for bipedal robot walking.” In: *2009 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*. 2009, pp. 162–167. DOI: 10.1109/AIM.2009.5230022.

- [28] Quan Nguyen et al. “Dynamic Walking on Randomly-Varying Discrete Terrain with One-step Preview.” In: July 2017. DOI: 10.15607/RSS.2017.XIII.072.
- [29] J.W. Grizzle et al. “MABEL, a new robotic bipedal walker and runner.” In: July 2009, pp. 2030–2036. DOI: 10.1109/ACC.2009.5160550.
- [30] Jemin Hwangbo et al. “ROCK* Efficient black-box optimization for policy learning.” In: *2014 IEEE-RAS International Conference on Humanoid Robots*. 2014, pp. 535–540. DOI: 10.1109/HUMANOIDS.2014.7041414.
- [31] Yohei Kakiuchi et al. “Development of life-sized humanoid robot platform with robustness for falling down, long time working and error occurrence.” In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2017, pp. 689–696. DOI: 10.1109/IROS.2017.8202226.
- [32] Boston Dynamics. *Spot to the Rescue*. Website. (Accessed: October 23, 2023). 2023. URL: <https://bostondynamics.com/blog/spot-to-the-rescue/>.
- [33] T. Hemker et al. “Efficient Walking Speed Optimization of a Humanoid Robot.” In: *The International Journal of Robotics Research* 28.2 (2009), pp. 303–314. DOI: 10.1177/0278364908095171. eprint: <https://doi.org/10.1177/0278364908095171>. URL: <https://doi.org/10.1177/0278364908095171>.
- [34] Shibendu Shekhar Roy and Dilip Kumar Pratihar. “Dynamic modeling, stability and energy consumption analysis of a realistic six-legged walking robot.” In: *Robotics and Computer-Integrated Manufacturing* 29.2 (2013), pp. 400–416. ISSN: 0736-5845. DOI: [http://doi.org/10.1016/j.rcim.2012.09.010](https://doi.org/10.1016/j.rcim.2012.09.010). URL: <https://www.sciencedirect.com/science/article/pii/S0736584512001135>.
- [35] J.D. Weingarten et al. “Automated gait adaptation for legged robots.” In: *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*. Vol. 3. 2004, 2153–2158 Vol.3. DOI: 10.1109/ROBOT.2004.1307381.
- [36] Mohsen Soori, Behrooz Arezoo, and Roza Dastres. “Optimization of energy consumption in industrial robots, a review.” In: *Cognitive Robotics* 3 (2023), pp. 142–157. ISSN: 2667-2413. DOI: <https://doi.org/10.1016/j.cogr.2023.05.003>. URL: <https://www.sciencedirect.com/science/article/pii/S2667241323000174>.
- [37] Syamimi Shamsuddin et al. “Humanoid robot NAO: Review of control and motion exploration.” In: *2011 IEEE International Conference on Control System, Computing and Engineering*. 2011, pp. 511–516. DOI: 10.1109/ICCSCE.2011.6190579.
- [38] Navvab Kashiri et al. “An Overview on Principles for Energy Efficient Robot Locomotion.” In: *Frontiers in Robotics and AI* 5 (2018). ISSN: 2296-9144. DOI: 10.3389/frobt.2018.00129. URL: <https://www.frontiersin.org/articles/10.3389/frobt.2018.00129>.
- [39] Heinrich Mellmann et al. *Berlin United - Nao Team Humboldt Team Report 2019*. Tech. rep. Adaptive Systeme, Institut für Informatik, Humboldt-Universität zu Berlin, Unter den Linden 6, 10099 Berlin, Germany, 2019. URL: <https://www2.informatik.hu-berlin.de/~naoth/docs/publications/technical/naoth-report19.pdf>.
- [40] Junichi Urata et al. “Design of high torque and high speed leg module for high power humanoid.” In: *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2010, pp. 4497–4502. DOI: 10.1109/IROS.2010.5649683.

Bibliography

- [41] Kazuma Morikawa and Seiichiro Katsura. “Thermoelectric Cooling Application to Motors for High-Power Operation.” In: *IEEJ Journal of Industry Applications* 12 (Dec. 2022). DOI: [10.1541/ieejjia.22004623](https://doi.org/10.1541/ieejjia.22004623).
- [42] Junichi Urata et al. “Thermal control of electrical motors for high-power humanoid robots.” In: *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2008, pp. 2047–2052. DOI: [10.1109/IROS.2008.4651110](https://doi.org/10.1109/IROS.2008.4651110).
- [43] Wen-Loong Ma, Yizhar Or, and Aaron D. Ames. “Dynamic Walking on Slippery Surfaces : Demonstrating Stable Bipedal Gaits with Planned Ground Slippage.” In: *2019 International Conference on Robotics and Automation (ICRA)*. 2019, pp. 3705–3711. DOI: [10.1109/ICRA.2019.8793761](https://doi.org/10.1109/ICRA.2019.8793761).
- [44] J.A. Vazquez-Santacruz and M. Velasco-Villa. “Experimental Estimation of Slipping in the Supporting Point of a Biped Robot.” In: *Journal of applied research and technology* 11 (June 2013), pp. 348–359. DOI: [10.1016/S1665-6423\(13\)71545-1](https://doi.org/10.1016/S1665-6423(13)71545-1).
- [45] Gustavo Medrano-Cerda, Houman Dallali, and and Brown. “Control of a Compliant Humanoid Robot in Double Support Phase: A Geometric Approach.” In: *International Journal of Humanoid Robotics* vol.1 (Mar. 2012). DOI: [10.1142/S0219843612500041](https://doi.org/10.1142/S0219843612500041).
- [46] Daan Hobbelin, Tomas De Boer, and Martijn Wisse. “System overview of bipedal robots Flame and TULip: Tailor-made for Limit Cycle Walking.” In: Oct. 2008, pp. 2486–2491. DOI: [10.1109/IROS.2008.4650728](https://doi.org/10.1109/IROS.2008.4650728).
- [47] Mochamad Ayuf Basthomí et al. “Walking Balance Control for Humanoid Soccer Robot on Synthetic Grass.” In: *2020 International Electronics Symposium (IES)*. 2020, pp. 213–218. DOI: [10.1109/IES50839.2020.9231879](https://doi.org/10.1109/IES50839.2020.9231879).
- [48] Dimas Pristovani Riananda et al. “Walking Trajectory Optimization Algorithm For Robot Humanoid on Synthetic Grass.” In: *EMITTER International Journal of Engineering Technology* 6.1 (July 2018), pp. 35–61. DOI: [10.24003/emitter.v6i1.229](https://doi.org/10.24003/emitter.v6i1.229). URL: <https://emitter2.pens.ac.id/ojs/index.php/emitter/article/view/229>.
- [49] Philip Reichenberg. *Entwicklung eines dynamischen Aufstehens unter Vermeidung inkorrektener Bewegungszustände*. BA thesis. 2020. URL: <https://b-human.de/downloads/theses/master-thesis-preichenberg.pdf>.
- [50] Arne Böckmann. “Entwicklung einer dynamischen Schussbewegung mit dem humanoiden Roboter NAO.” MA thesis. Universität Bremen, 2015.
- [51] Philip Reichenberg. “Sturzverminderung von humanoiden FuSSballrobotern unter Entwicklung eines passiven Zweikampfes.” MA thesis. Universität Bremen, 2021.
- [52] Jesse Richter-Klug. “Visuelle Odometrie in der RoboCup Standard Platform League.” MA thesis. Universität Bremen, 2018.
- [53] Inc. NumFOCUS. *pandas*. Website. (Accessed: November 1, 2023). 2023. URL: <https://pandas.pydata.org/>.
- [54] Nicolas Van der Noot and Allan Barrea. “Zero-Moment Point on a bipedal robot under bio-inspired walking control.” In: *MELECON 2014 - 2014 17th IEEE Mediterranean Electrotechnical Conference* (2014), pp. 85–90. URL: <https://api.semanticscholar.org/CorpusID:12987843>.

- [55] Toshiyuki Nagasawa, Yuta Tawaki, and Toshiyuki Murakami. “Comparison of Filtering Methods in Measuring Human ZMP Using Kinect Sensor.” In: *IECON 2022 48th Annual Conference of the IEEE Industrial Electronics Society*. 2022, pp. 1–6. DOI: 10.1109/IECON49645.2022.9969023.

Declaration

Hereby, I declare that I produced the present work myself only with the help of the indicated aids and sources. The thesis in its current or a similar version has not been submitted to an auditing institution before.

Hamburg, April 24, 2024

Place, Date

Maik Marius Rebaum