

CZECH TECHNICAL UNIVERSITY IN PRAGUE

Faculty of Electrical Engineering

BACHELOR THESIS



Vojtěch Janů

Motion control for a humanoid

Department of Cybernetics

Thesis supervisor: **RNDr. Miroslav Kulich, Ph.D.**

BACHELOR PROJECT ASSIGNMENT

Student: Vojtěch Janů
Study programme: Cybernetics and Robotics
Specialisation: Robotics
Title of Bachelor Project: Motion Control for a Humanoid

Guidelines:

1. Get acquainted with ROS (Robot Operating System, <http://ros.org>) and Gazebo (gazebo.org).
2. Make a literature review of algorithms for motion control of a humanoid robot.
3. Choose (in cooperation with the supervisor) an appropriate type of a humanoid and create/import its model into the Gazebo environment.
4. Implement the chosen algorithm for humanoid control.
5. Verify experimental features and behaviour of the implemented control algorithm in the Gazebo environment and the chosen robot.
6. Suggest improvements of the implemented algorithm based on experience made during implementation.

Bibliography/Sources:

- [1] Jung-Yup Kim, Ill-Woo Park, Jun-Ho Oh: Experimental realization of dynamic walking of the biped humanoid robot KHR-2 using zero moment point feedback and inertial measurement, *Advanced Robotics*, Vol. 20, Iss. 6, 2006.
- [2] Hashimoto, K.; Takezaki, Y.; Motohashi, H.; Otani, T.; Kishi, T.; Hun-ok Lim; Takanishi, A.: "Biped walking stabilization based on gait analysis," *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, vol., no., pp.154,159, 14-18 May 2012.
- [3] Qiang Huang; Yokoi, K.; Kajita, S.; Kaneko, K.; Arai, H.; Koyachi, N.; Tanie, K.: "Planning walking patterns for a biped robot," *Robotics and Automation, IEEE Transactions on*, vol.17, no.3, pp.280,289, Jun 2001
- [4] P. Sardain and G. Bessonnet: 2004. Forces acting on a biped robot. Center of pressure-zero moment point. *Trans. Sys. Man Cyber. Part A* 34, 5 (September 2004), 630-637.
- [5] Kajita, S.; Kanehiro, F.; Kaneko, K.; Fujiwara, K.; Harada, K.; Yokoi, K.; Hirukawa, H.: "Biped walking pattern generation by using preview control of zero-moment point," *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on*, vol.2, no., pp.1620,1626 vol.2, 14-19 Sept. 2003.

Bachelor Project Supervisor: RNDr. Miroslav Kulich, Ph.D.

Valid until: the end of the summer semester of academic year 2015/2016

L.S.

doc. Dr. Ing. Jan Kybic
Head of Department

prof. Ing. Pavel Ripka, CSc.
Dean

Prague, January 20, 2015

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: Vojtěch Janů
Studijní program: Kybernetika a robotika (bakalářský)
Obor: Robotika
Název tématu: Řízení pohybu humanoidního robota

Pokyny pro vypracování:

1. Seznamte se s prostředím ROS (Robot Operating System, <http://ros.org>) a Gazebo (gazebo.org).
2. Proveďte rešerši algoritmů řízení pohybu humanoidního robota.
3. Ve spolupráci s vedoucím vyberte vhodný typ humanoidního robota a vytvořte/importujte jeho model do prostředí Gazebo.
4. Implementujte vybraný algoritmus řízení.
5. Proveďte experimentální ověření vlastností implementovaných algoritmů řízení v systému Gazebo a na vybraném robotu.
6. Na základě získaných poznatků navrhněte vylepšení stávajících algoritmů.

Seznam odborné literatury:

- [1] Jung-Yup Kim , Ill-Woo Park , Jun-Ho Oh: Experimental realization of dynamic walking of the biped humanoid robot KHR-2 using zero moment point feedback and inertial measurement, Advanced Robotics, Vol. 20, Iss. 6, 2006.
- [2] Hashimoto, K.; Takezaki, Y.; Motohashi, H.; Otani, T.; Kishi, T.; Hun-ok Lim; Takanishi, A.: "Biped walking stabilization based on gait analysis," Robotics and Automation (ICRA), 2012 IEEE International Conference on , vol., no., pp.154,159, 14-18 May 2012.
- [3] Qiang Huang; Yokoi, K.; Kajita, S.; Kaneko, K.; Arai, H.; Koyachi, N.; Tanie, K.: "Planning walking patterns for a biped robot," Robotics and Automation, IEEE Transactions on, vol.17, no.3, pp.280,289, Jun 2001
- [4] P. Sardain and G. Bessonnet: 2004. Forces acting on a biped robot. Center of pressure-zero moment point. Trans. Sys. Man Cyber. Part A 34, 5 (September 2004), 630-637.
- [5] Kajita, S.; Kanehiro, F.; Kaneko, K.; Fujiwara, K.; Harada, K.; Yokoi, K.; Hirukawa, H.: "Biped walking pattern generation by using preview control of zero-moment point," Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on , vol.2, no., pp.1620,1626 vol.2, 14-19 Sept. 2003.

Vedoucí bakalářské práce: RNDr. Miroslav Kulich, Ph.D.

Platnost zadání: do konce letního semestru 2015/2016

L.S.

doc. Dr. Ing. Jan Kybic
vedoucí katedry

prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 20. 1. 2015

Declaration of author of the thesis

I hereby declare that I have completed this thesis independently and that I have used only the sources listed in the enclosed bibliography and that I did it in accordance with the Methodological instructions about adhering to ethical principles when preparing university theses.

Prohlášení autora práce

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne

.....

Podpis autora práce

Acknowledgements

I would like to thank and express my sincere appreciation to my supervisors RNDr. Miroslav Kulich, Ph.D., for his support and provided assistance wherever it was needed.

Next I would like to thank to my family and friends for their support, inspiration and motivating discussions.

Abstrakt

Cílem této bakalářské práce je návrh algoritmu chůze pro humanoidního robota. V průběhu práce je zvolen robot a vhodný algoritmus chůze pro aplikaci. Následně je sestaveno simulační a vývojové prostředí. Na robotu je provedena regulace motorů a aplikován výpočet a zobrazení referenčního bodu "Zero moment point". Pro robot je navržen algoritmus pro zahájení chůze pomocí snížení těžiště a rozkývání robota v laterálním směru s frekvencí rovnou vlastní frekvenci. Poté je aplikována chůze robota na místě.

Klíčová slova: řízení, robot, chůze, humanoid, dvounohý robot

Abstract

Goal of this bachelor thesis is to design algorithm for a walking of a humanoid robot. During the thesis, the robot and the algorithm for the walking is chosen and a simulation and a development environment is set. The regulation of motors of the humanoid is tuned up and the zero moment point is computed and displayed. Starting sequence of the humanoid robot is designed with usage of a lowering and a swinging of pelvis. Therefore, walking on one spot is applied for a humanoid robot.

Keywords: motion control, robot, walk, gait, humanoid, biped

Contents

1	Introduction	1
2	Terminology	2
3	State of the art	5
3.1	Application of walk	5
3.2	Humanid robotics	6
4	Algorithm	8
4.1	Simple inverted pendulum	8
4.2	Walking pattern	9
4.3	Walking stabilization	9
4.3.1	Damping controller	10
4.3.2	ZMP compensator	10
4.3.3	Soft landing controllers	13
4.3.4	Pelvis swing amplitude controller	13
4.3.5	Torso pitch/roll controller	14
4.3.6	Tilt over controller	14
4.3.7	Landing position controller	14
5	Simulation environment	16
5.1	Robot REEM-C	16
5.2	ROS	16
5.3	Gazebo	17
5.4	Rviz	18

CONTENTS

6	Realization	19
6.1	Robot selection	19
6.2	Setup of the development environment	20
6.2.1	Installation of Ubuntu	20
6.2.2	Matlab	21
6.3	Application of walking	22
6.3.1	ZMP node	22
6.3.2	Control of motors	23
6.3.3	Leg movements	25
7	Conclusions	27

List of Figures

2.1	Phases of human or humanoid walk where red is right leg and blue is left leg . . .	3
2.2	Phases of human or humanoid walk where red is right leg and blue is left leg . . .	4
4.1	Block diagram of damping control	11
4.2	Block diagram of ZMP compensator	12
4.3	ZMP and projection of COM displayed at support polygon of stance feet . . .	13
4.4	At figure a) is inside tilt over, at figure b) is outside tilt over. First move is shown red and second is blue	15
5.1	REEM-C robot (Source: http://pal-robotics.com/static/palrobotics/img/home/slider/reemc_hello.png)	17
6.1	The required environment connection	21
6.2	ZMP nodes structure	22
6.3	The comparison of new and original arm control in response to a step	23
6.4	The comparison of new and original leg control in response to a step both with new control of arms running	23
6.5	The comparison of new and original torso control in response to a step both with new control of legs and arms running	24
6.6	Lateral displacement of pelvis. On the left are the data of algorithm, on the right are the real data of simulation	25
6.7	Walk of REEM-C	26

List of Tables

4.1	Schedule of conrollers running during stages. SSP is single support phase and DSP is double support phase	10
5.1	REEM-C: Sensors and cameras	18
6.1	REEM-C: Sensors and cameras	20
6.2	Table of values of original and new PIDs	24
7.1	CD Content	31

Chapter 1

Introduction

Walking is one of the most interesting and most difficult ways of transportation in a rough terrain and in comparison with driving it has many advantages and disadvantages that are mostly caused by construction and complexity of a walker. The greatest disadvantages in complexity are caused by the number of degrees of freedom (DOF) of each leg (each human leg has 7DOF) in comparison with the wheels (each wheel has 1DOF plus 1DOF for all car for turning) and cause difficulty to control whole walking process. Lower stability and difficulty to achieve higher speeds are associated with complexity and structure too. On the other hand the greatest advantages of walking are its almost unlimited maneuverability and possibility of overcoming obstacles that can be seen at parkour runners. This complexity of walking fascinated me and the challenge of understanding all processes that enable me to create a functional walking algorithm attracted me.

Within the thesis I try to realize a walking algorithm thanks to which it will be possible to search unknown environment with humanoid, transport objects, carry loads etc. This algorithm allows motion on inclined or uneven terrain thanks to its controllers. The walking algorithm is written in Robot Operating System (ROS) so it could be easily transferred to another humanoid with similar structure.

Structure of this thesis is as follows. Used terminology are defined in chapter 2. In chapter 3 are summarized the algorithms for motion control and most famous humanoid robots are described. In chapter 4 is described algorithm chosen for implementation. Simulation and development environment is described in chapter 5. The chapter 6 describes realization of chosen algorithm. Evaluation of the results is in the chapter 7.

Chapter 2

Terminology

There are many terms and definitions used in designing humanoid robots and developing algorithms for a walk. The following list provides a summary of these definitions that are used in this thesis.

Stance phase. *Stance phase is a phase in which the leg supports the weight of a body. Through this phase feet must be in contact with the ground and when friction is too high for the feet to stay at the same point for the whole phase.*

Swing phase. *Swing phase is a phase in which the leg is not in contact with the ground. In this phase during human walk leg is moving forward to the point in which a feet touches the ground and a swing leg became a stance leg.*

Single support phase. *Single support is the part of human walk in which the weight of a human body is supported by only one leg. The second leg may not have to provide any contact with the ground.*

Double support phase. *Double support phase of human walk is phase in which body weight is supported by both legs but is not necessary to have full feet contact with the ground.*

Convex hull. *Convex hull of a set of points is a hull of a minimal convex set containing all given points.*

Support polygon. *Support polygon is a convex hull of all contact points on the ground.*

Frontal (Coronal) plane. *Frontal or Coronal plane is the plane perpendicular to x axis and it is parallel to xy plane.*

Siggital (Median) plane. *Siggital or Median plane is the plane perpendicular to y axis and is parallel to xz plane.*

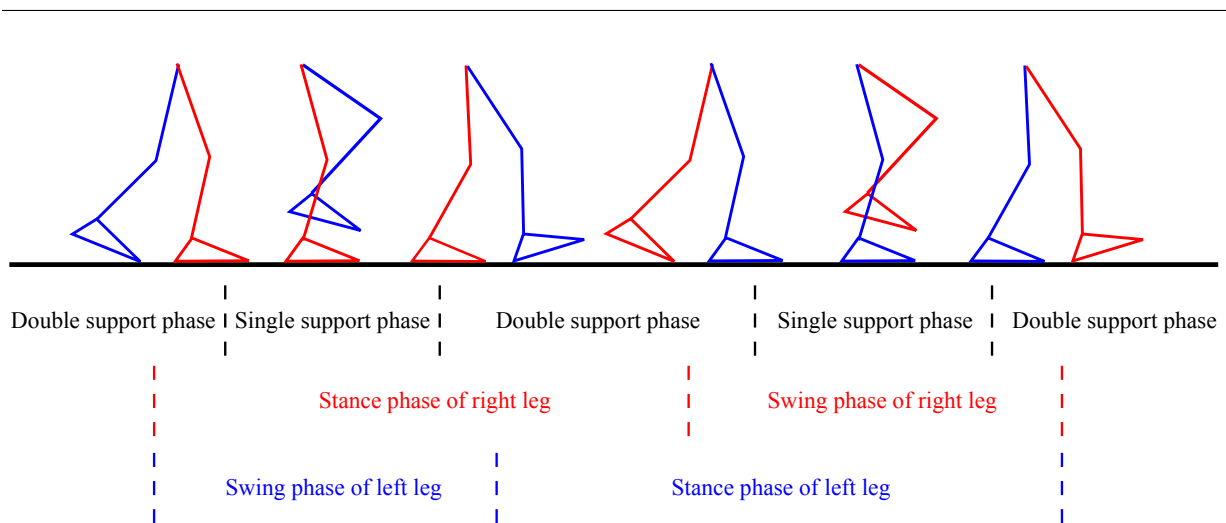


Figure 2.1: Phases of human or humanoid walk where red is right leg and blue is left leg

Transverse (Horizontal) plane. *Transverse or Horizontal plane is the plane perpendicular to z axis and it is parallel to yz plane.*

Lateral movement. *Lateral movement is a movement in y axis direction (in the Coronal plane).*

Zero moment point. *Zero moment point is dynamic criterion. It is defined as the point on the ground about which the sum of all the moments of the active forces equals to zero.*

Degrees of freedom. *degrees of freedom (DOF) is number of independent parameters that determine state of mechanical system. Number of DOF can be calculated using Gruebler's Equation [1] (equation 2.1).*

$$nDOF = 3(n - 1) - 2 \cdot l - h \quad (2.1)$$

Where $nDOF$ is the resulting number of DOF, n is number of links, l is number of lower pairs and h is number of higher pairs.

Inverse kinematics (IK). *Inverse kinematics use kinematics equations of a robot to determine the joint parameters from known position of the end-effector.*

Forward kinematics. *Forward kinematics use kinematics equations of a robot to determine the position of the end-effector from the joint parameters.*

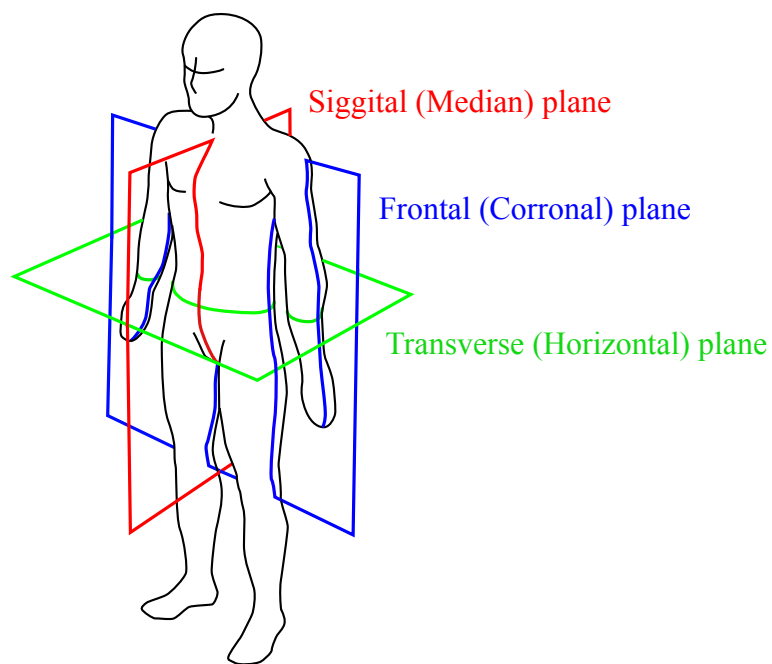


Figure 2.2: Phases of human or humanoid walk where red is right leg and blue is left leg

Chapter 3

State of the art

3.1 Application of walk

Human has been interested in walking since the first studies of anatomy. The biggest boom of the discipline was reached during the Second World War when the world was in need of a good enough artificial limb which could replace a healthy leg and enable soldiers with amputation to walk. Since simple prosthesis, evolution of technology enables a humankind to create much more sophisticated mechanisms. With this progress, a human tries to reproduce walking process, the ideal way to transport over an uneven terrain, not only on the human with use of prostheses but also on a self-driven mechanisms that could walk independently. In this reason many sophisticated mechanisms were discovered and many algorithms for realization of stable walking on biped mechanisms were developed to these days. These existing algorithms are still improved and new algorithms are created to achieve faster, more stable and smoother gait on various types of surfaces and to avoid collisions.

Some of these algorithms are based on particle swarm optimization using human joint data [2] or based on a sensor-driven neuronal controller and real-time on-line learning [3] which enables a robot to move without a position or trajectory tracking control algorithm. Very common are algorithms based on a simple inverted pendulum model [4, 5, 6, 7, 8] which use Zero Moment Point (ZMP) as stability criterion. But ZMP is not only one existing criterion for stability and floor projection of Center Of Mass(COM) or Center Of Pressure(COP) [9] can be and are used too. Unlike projection of COM, ZMP reflects effects of dynamics on a robot and is based on dynamics equilibrium. ZMP is defined as the point on the ground about which the sum of all the moments of the active forces equals zero [10].

During the walking, many factors that could affect the stability and cause a fall may occur. Even adult healthy people sometimes fall on slippery or unpaved surfaces. Therefore use of control is necessary to avoid these factors. Because walking is complicated repeated unstable movement based on repetitive tilt over is necessary to choose the correct walking speed and the correct stride length to avoid fall in sigittal plane. There is simple a rule that can be used to avoid from falling forward or backward during walking.

"You will never fall forward if you put your swing leg fast enough in front of your stance leg. In order to prevent falling backward the next step, the swing leg shouldn't be too far in front." [11]

It says that step a must be so short that it enables the robot to use inertia force move through a single support phase and so fast that the robot steadily lends and can go through a double support phase. But stabilization must be carried out in the coronal plane too.

Many algorithms can be used to improve walking by modifying gait. One of possible modifications is the footstep planning algorithm [12] that enables robot to select a position of lend of a foot. Another possible modification is for example an algorithm for compensation forces acting on a robot [13] that could be caused by interaction with a human.

3.2 Humanid robotics

Similarly to the interest in walking, human has been always fascinated by the systems that look and behave like a living beings especially those who look like the humans. One of the first who was fascinated so much that he designed a humanoid automaton was Leonardo Da Vinci. Since then many people tried to create more and more sophisticated mechanism. Two of them were Pierre Jaquet-Droz and Jacques de Vaucanson.

With the development of an information technologies and discovery of an integrated circuits attempts to build a computer-controlled mechanism were realized. In 1973 first full-scale anthropomorphic computer-controlled humanoid mechanism was created by the Ichiro Kato's team at Waseda University in Tokyo. Subsequently many other humanoid robots were created in this university. One of the last ones is Wabian-2R a 150cm tall robot that weighs 64Kg have 41DOF and is equipped with a passive toe joint that makes its foot able to move in more human-like movements.

However, the Asimo is much more famous humanoid robot than Wabian tha was created in Japan. This robot that in its latest version from 2014 is 130 cm tall, weighs 55 Kg and is able to move in 57DOF [14]. Its development began in 1986 in the Honda Motor Company by E series, in 1993 followed by P series that was equipped with a head and hands and first robot of ASIMO series was introduced in 2000. The ASIMO robot is in its latest version able to walk upstairs and downstairs, is able to footstep planning [12] to avoid collision or to run at speeds of 7 km/h and movements of this robot are one of the smoothest and the most human-like.

Also in Japan, National Institute of Advanced Industrial Science and Technology (AIST) in cooperation with Kawada Industries formed the Humanoid Robotics Project (HRP). The latest robots made in this project are HRP-4C that is 158 cm tall gynoid (female-looking humanoid robot) that can utilize motors for facial expressions to look more like humans and 151 cm tall HRP-4 with 34 DOF.

The child-size robot HUBO was created at KAIST university in Korea as a direct competitor to the robot ASIMO. The weight, height and total degrees of freedom of HUBO-2 version made

3.2. HUMANID ROBOTICS

in 2009 are 45 Kg, 125 cm and 41DOF [15]. This robot is able to walk and run and it will participate in DARPA Robotics Challenge [16]. DARPA Robotics Challenge is challenge where 25 robotics organizations around the world will compete from 5th to 6th June 2015 in disaster simulation for \$3.5 million.

Robot ATLAS is another robot participating in DARPA Robotics Challenge and it is developed by Boston Dynamics currently owned by Google. Boston Dynamics is well known for their robot Cheetah that is the fastest legged robot in the world or LS3 that was created for carrying heavy load through a rough terrain. The last version of ATLAS robot for search and rescue purpose from 2015 is 188 cm tall, weighs 156 Kg and is equipped with a new battery pack which makes it able to work for one hour on a "mixed mission" [17].

Among Europe's most famous humanoid robots there is NAO invented by Aldebaran based in Paris. This 58 cm tall robot was selected in 2008 for RoboCup Soccer League as the successor to Sony AIBO. Since 2009 Aldebaran company has been developing a new robot Romeo that is to act as Robot Staff Assistant.

Representative of European humanoid robots is also REEM-C made by Spanish PAL Robotics that is used in this paper.

Chapter 4

Algorithm

In this paper one of existing algorithms was chosen with regard to the objective to apply walking to a humanoid robot. The algorithm is described in [4] and is designed to imitate a human gait. The algorithm is partially developed experimentally and partially theoretically and as many other algorithms stabilization in a single support phase is based on a model of an inverted pendulum and the criterion of stability is based on the zero moment point. The advantage of this algorithm, among other, is its successful realization at the humanoid robot KHR-2, its modularity and simplicity. On the other hand as disadvantage of the algorithm can be considered its age when more complex and robust algorithms exist these days.

4.1 Simple inverted pendulum

At the beginning of walk, the robot is set to the walking ready pose in which the robot has slightly bend knees to avoid singularity in inverse kinematics computing. During robot's walk, center of mass (COM) is repeatedly in single support phase and in this phase the robot can be approximated as a simple inverted pendulum. The natural frequency of the inverted pendulum can be computed as:

$$f = \frac{1}{2\pi} \sqrt{\frac{g}{l}} \quad (4.1)$$

where g is gravitational acceleration and l is the high of the center of mass. This frequency is used to determined sideways tilt frequency (frequency of a move of COM in the coronal plane during transferring weight) of a walking robot to utilize energy of a falling pendulum during weight shift from one leg to the other. Thanks to use of natural frequency, energy consumption during walking is reduced.

4.2 Walking pattern

There are three basic factors to design and setup walking pattern.

1. Walking period (frequency)
2. Double-support ratio
3. Lateral swing amplitude of the pelvis(swing in coronal plane)

The period is set to the value computed from frequency of simple inverted pendulum (equation 4.1) in this algorithm. Double-support ratio is set experimentally. A portion of double-support phase in walking cycle of human is between 10% and 20% whereas for a robot is this portion lower because of absence of a toes. The lateral swing amplitude is set experimentally based on a human amplitude. In the case of an adult human the lateral swing amplitude is about 6cm during walk with the frequency 1 step per second. When a walking speed is increases the swing amplitude is decreases. This is caused by higher frequency of putting feet to the floor in which a center of mass has shorter time to fall.

According to the previous factors, trajectories for a pelvis and a feet were designed. Pelvis trajectory was created using a cosine function to achieve a smooth path without discontinuity of velocity that could produce destabilization of the robot. At the maximum and minimum displacement is time delay to perform better stability. Elevation of both feet (in z direction) is generated with use of cosine function too. Cycloid function is used for absolute trajectory of both feet.

4.3 Walking stabilization

To perform smooth and sable movement all motors are PD regulated. To achieve higher efficiency five stages of walk are designed according to which controllers are run.

1. Lift the first leg to its maximum flexion and height defined by the algorithm.
2. Lower the first leg until it makes a complete contact with the ground.
3. Lift the second leg to its maximum flexion and height defined by the algorithm.
4. Lower the second leg until it makes a complete contact with the ground.
5. Follows after Stage 1. or 3. and brings the robot to the stop pose with both legs landed on the ground.

The first four stages are cycling during walking and the fifth stage runs during the standstill pose. Table 4.1 shows which controllers run during the particular stages.

4.3. WALKING STABILIZATION

Controllers	Stage 1	Stage 2		Stage 3	Stage 4		Stage 5	
		SSP	DSP		SSP	DSP	SSP	DSP
Damping controller	Yes	Yes	no	Yes	Yes	no	Yes	no
ZMP compensator	Yes	Yes	no	Yes	Yes	no	Yes	no
Soft landing controllers	no	no	Yes	no	no	Yes	no	Yes
Pelvis swing amplitude controller	no	no	Yes	no	no	Yes	no	no
Torso pitch/roll controller	no	no	Yes	no	no	Yes	no	Yes
Tilt over controller	Yes	no	no	Yes	no	no	no	no
Landing position controller	no	Yes	Yes	no	Yes	Yes	Yes	Yes

Table 4.1: Schedule of controllers running during stages. SSP is single support phase and DSP is double support phase

4.3.1 Damping controller

The damping controller is designed to eliminate oscillation in the single support phase. Those oscillations came from a force/torque sensor of the stance leg that is installed in an ankle joint and from the swing leg itself. The model for this controller is designed as a simple inverted pendulum with a compliant joint. The control block diagram is shown in figure 4.1 where u is the reference joint angle, θ is the actual joint angle, K is stiffness of leg, T is the measured torque, g is gravitational acceleration k_d is gain of damping control and u_c is a compensated joint angle. α and β are defined as follows:

$$\alpha = \frac{K}{ml^2} - \frac{g}{l} \quad (4.2)$$

$$\beta = \frac{K}{ml^2} \quad (4.3)$$

where m is mass of the pendulum and l is a length of pendulum.

4.3.2 ZMP compensator

The ZMP compensator is designed to eliminate fluctuations of a humanoid body during the single support phase of walking and is used because damping control itself is not efficient enough. As controlled variable pelvis displacement is used in x and y axes, moreover ZMP as a reference is employed. ZMP allows to control static and dynamic influence of fluctuations. Mathematical models of KHR-2 in presence of damping control are fourth-order systems that

4.3. WALKING STABILIZATION

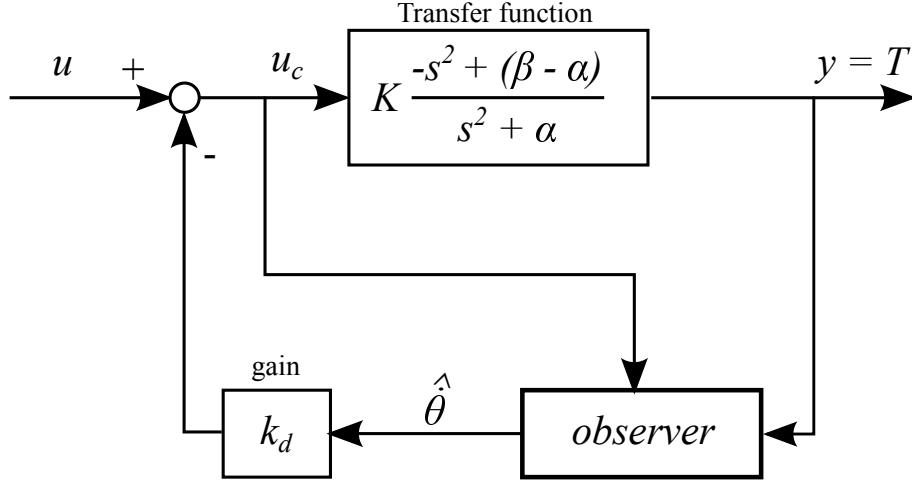


Figure 4.1: Block diagram of damping control

are derived by experimental frequency response analysis and are shown in equations 4.4 and 4.5 where ZMP_x and ZMP_y are x and y components of ZMP and $pelvis_x$ and $pelvis_y$ are x and y components of pelvis displacement.

$$\frac{ZMP_y}{pelvis_y} = \frac{28541.65}{(s^2 + 4.59s + 131.52)^2} \quad (4.4)$$

$$C_x(s) = \frac{6668.34}{(s^2 + 3.68s + 106.77)^2} \quad (4.5)$$

Compensators of regulators of both axes are designed by using a pole placement technique and in the case of KHR they are fifth-order compensators. Each compensator has an integrator that helps to prevent steady-state error and improves continuity of the input. These compensators are shown in equations 4.6 and 4.7.

$$C_y(s) = \frac{-34.94s^4 - 1396s^3 - 18287.5s^2 - 185137.6s - 607515.1}{s^5 + 80s^4 + 2831.7s^3 + 54573.2s^2 + 530148.6s + 1593487.3} \quad (4.6)$$

$$C_x(s) = \frac{-62.065s^4 - 3020.8s^3 - 35673.5s^2 - 355565.5s - 1196033.3}{s^5 + 70.44s^4 + 2255.6s^3 + 40240.4s^2 + 376617.8s + 1118129.2} \quad (4.7)$$

The block diagram of the ZMP compensator is on the figure 4.2 where $C(s)$ is compensator, $G(s)$ is a transfer function of the system, u_{pelvis} is displacement of the uncontrolled algorithm, $u_{compens}$ is compensatory change of pelvis displacement, ZMP_{ref} is a reference ZMP criterion point and y is compensator output.

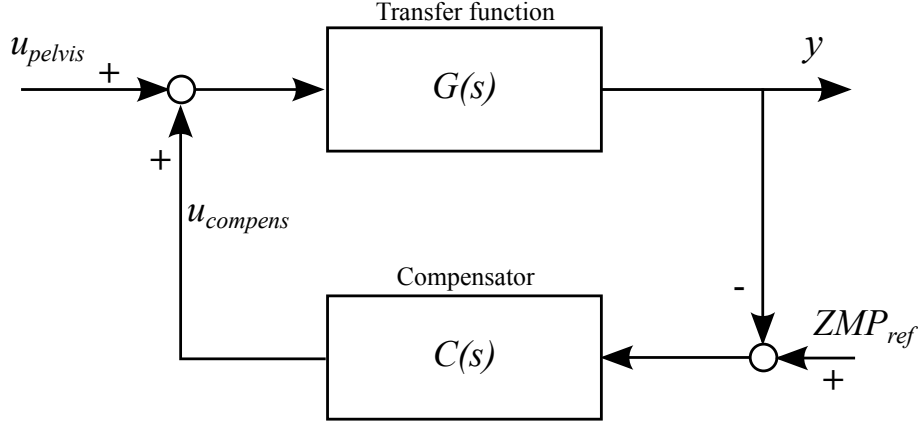


Figure 4.2: Block diagram of ZMP compensator

Zero Moment Point

The Zero moment point(ZMP) is a dynamic equivalent of projection of the center of mass. Projection of COM consider only static forces unlike ZMP which considers both static and dynamic forces. Consequently projection of COM is sufficient criterion only in situation where dynamic forces are negligible otherwise ZMP must be used. The Zero moment point is defined as a point on the ground, where total sum of inertial and gravity moments are equal to zero. This criterion was first defined by Mimir Vukobratović in 1972. [18]

This two criteria are shown in figure 4.3 where M is mass of the whole body, $F_{gravity}$ is gravity force acting on the center of mass, $F_{inertia}$ is the sum of all inertia forces acting on center of mass, ZMP is the zero moment point and COM is the floor projection of center of mass. Moment of force M about any point P may be expressed as:

$$M_P = \vec{PG} \times m\vec{g} - \vec{PG} \times m\vec{a}_G - \dot{\vec{H}}_G \quad (4.8)$$

where G is the position of COM, m is mass of COM, \vec{g} is gravity acceleration, \vec{a}_G is acceleration of COM and $\dot{\vec{H}}_G$ is the rate of angular momentum at the COM. When we assumed that friction of feet is so high that any sliding could be neglected ZMP could be determined form a force/torque sensor for x and y direction as:

$$ZMP_x = \frac{Torque_y}{Force_z} \quad (4.9)$$

$$ZMP_y = \frac{Torque_x}{Force_z} \quad (4.10)$$

where $Torque_x$ and $Torque_y$ are torques around x and y axis of the force/torque sensor and $Force_z$ is z component of force acting on the force/torque sensor.

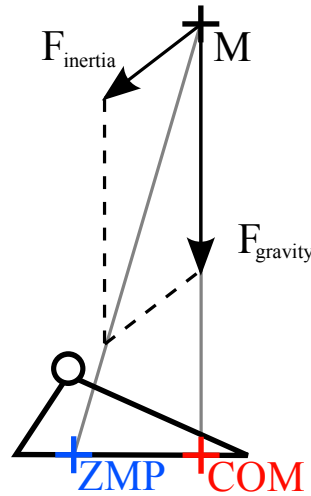


Figure 4.3: ZMP and projection of COM displayed at support polygon of stance feet

4.3.3 Soft landing controllers

This control consists of two parts: of a timing controller and an orientation controller. The timing controller interrupts walking cycles whenever the foot does not land on the ground to the end of the 2. or 4. stage and wait until the foot stably land on the ground. The orientation controller adapts an ankle joint to the ground surface to achieve stable contact trough measure torques. The process of orientation control is described in equation 4.11 where u_c is compensated angle of th ankle, u is input of an uncontrolled joint angle, C_L is a damping coefficient and T is a measured torque.

$$u_c = u + \frac{Ts}{C_Ls + K_L} \quad (4.11)$$

4.3.4 Pelvis swing amplitude controller

Due to a decreasing amplitude with increasing walking speed Pelvis swing amplitude controller is used to adjust the pelvis amplitude according to move of ZMP during each walking cycle. The principle is simple.

1. Calculate the average values of the positive (around maximum of lateral swing) and negative (in maximum of lateral swing) ZMP during the n th walking cycle.
2. Derive the n th averaged ZMP amplitude (equation 4.13).
3. Modify the lateral pelvis swing amplitude in the $(n + 1)$ th walking cycle by adding the compensatory amplitude using a PI controller (equation 4.12).

4.3. WALKING STABILIZATION

$$A_{comp}^{pelvis}(n+1) = A_{pelvis} + ZMP_{err}(n) \cdot \left(k_p + \frac{k_I}{s}\right) \quad (4.12)$$

$$ZMP_{err} = ZMP_{y_{ref}} - \frac{1}{T_{step}} \left(\int_{Stage_3}^{Stage_4} ZMP_y dt - \int_{Stage_1}^{Stage_2} ZMP_y dt \right) \quad (4.13)$$

Where A_{comp}^{pelvis} is compensated pelvis swing amplitude, A_{pelvis} is uncompensated pelvis amplitude, k_p is proportional gain of PI controller, k_I is integral gain of PI controller, $ZMP_{y_{ref}}$ is averaged ZMP amplitude, ZMP_y is y component of ZMP, ZMP_{err} is deviation that is regulated and $Stage_1, Stage_2, Stage_3$ and $Stage_4$ are boundary of integral that starts at the beginning of $Stage_3$ respectively $Stage_1$ and ends at the end of $Stage_4$ respectively $Stage_2$.

4.3.5 Torso pitch/roll controller

This controller is useful when the robot walks on an uneven surface by compensating an angle of inclination to avoid tilting over during the movement. The controller integrates an angular position of the torso (from inertial sensors) in each single support phase and then calculates differences between the right and left single support phase and modifies pelvis center position in the next walking cycle by move the pelvis to the opposite direction from the inclination.

4.3.6 Tilt over controller

This controller is designed to avoid fall in the lateral direction. External forces or unevener can occurred fall in two direction. In one case the robot is falling on the swing leg and it is necessary to transfer weight on the swinging leg to avoid fall. In the second case the robot falls outside and is necessary for stabilization to transfer weight and perform a step because the leg are crossing each other. Both cases are shown in figure 4.4. The control law for this controller is as follows:

$$\text{If } \int_{Stage_{1,s}}^{Stage_{1,\tau}} \theta_r^t \text{ or } s \text{odt} > V_{TOout} \text{ or } \int_{Stage_{1,s}}^{Stage_{1,\tau}} \theta_r^t \text{ or } s \text{odt} < V_{TOin} \quad (4.14)$$

$$\text{Then } \theta_{m,r}^{R-ankle}(t) = \theta_r^{R-ankle}(t) + A_R(M_R) \sin(2\pi \frac{t_c}{\tau'}) (deg) (0 \leq t_c \leq \frac{\tau'}{2})$$

Where V_{TOout} and V_{TOin} are the experimentally set thresholds values, $M_R = \int_{Stage_{1,s}}^{Stage_{1,\tau}} dt$ (deg), τ' is tilt over control duration, θ_R^{torso} the rolling angle of the torso, A_R is amplitude of compensation and $\theta_r^{R-ankle}$ is the rolling angle of the ankle.

4.3.7 Landing position controller

The landing position controller is based on a mechanism of adjusting walking trajectories to step towards the falling direction. Thanks to this controller robots steps are all not the same. The following definition (equation 4.15) is a control law for this controller.

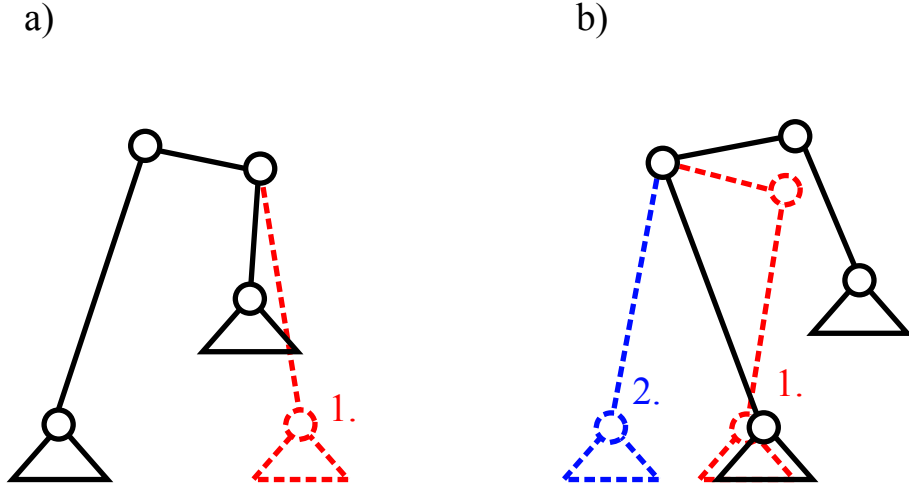


Figure 4.4: At figure a) is inside tilt over, at figure b) is outside tilt over. First move is shown red and second is blue

$$\begin{aligned} \tilde{P}_m^{foot}(n) &= \tilde{P}^{foot}(n) + k_p \tilde{\theta}_{err}^{torso} \\ \tilde{P}^{foot} &= \begin{pmatrix} P_x^{foot} \\ P_y^{foot} \end{pmatrix} \\ \tilde{\theta}_{err}^{torso} &= \begin{pmatrix} \dot{\theta}_{p,avg}^{torso} - \dot{\theta}_{p,stable}^{torso} \\ \dot{\theta}_{r,avg}^{torso} - \dot{\theta}_{r,stable}^{torso} \end{pmatrix} \end{aligned} \quad (4.15)$$

Where \tilde{P}^{foot} is an unmodified landing position vector, $\tilde{P}_m^{foot}(n)$ is a modified landing position vector of n-th step, $\dot{\theta}_{p,avg}^{torso}$ and $\dot{\theta}_{r,avg}^{torso}$ are average angular velocities of the torso in pitch and roll direction, $\dot{\theta}_{p,stable}^{torso}$ and $\dot{\theta}_{r,stable}^{torso}$ are the threshold of a stable angular velocity of the torso in pitch and roll direction and k_p is a proportional control gain.

Chapter 5

Simulation environment

5.1 Robot REEM-C

REEM-C is a member of PAL Robotics family. This company from Barcelona, founded in 2004 produces three robots, a full-size humanoid service robot REEM that uses undercarriage for its transport, automatic inventory robot StockBot and REEM-C. REEM-C robot that is in figure 5.1 is 165 cm tall, weighs 80 Kg. (7DOF in each arm ,7DOF in each Hand, 6DOF in each leg, 2DOF in neck and 2DOF in waist)Thanks to 44DOF this robot is able to imitate a full range of human movement as squatting, sitting, walking, waving hands and is able to carry 1 Kg payload in each hand. The computing core of the robot consists of two Intel core i7 computers with 4 GB RAM and 60 GB Solid-state drive in each. One computer is designed as a control computer running Real time OS using Xenomai core and the second one is designed as a media computer running Ubuntu 12.04 LTS. Sensors and cameras of REEM-C are described in table 6.1. The main advantage of this robot in comparison to its competitors is its full compatibility with ROS and existing simulation setting containing controllers, URDF model, transform frames connected with joints and sensors, set Gazebo and Rviz environment. Disadvantages of this robot are almost no information about and almost no documentation (only brief info at official pages and three tutorials [19] at ros pages and few comments in code) in comparison with for example Aldebaran robots [20]. Another disadvantage is REEM-Cs weight which is almost twice as big as HRP-4Cs with seven extra centimeters in height. This extra weight could cause greater energy consumption and worse mobility.

Controllers of REEM-C robot run at frequency of 50Hz which is the same as joint states and a force/torque sensors.

5.2 ROS

The Robot Operating System (ROS) is an open-source framework for writing software for robots. It is a complex ecosystem which consists of many libraries and tools that can be used



Figure 5.1: REEM-C robot

(Source: http://pal-robotics.com/static/palrobotics/img/home/slider/reemc_hello.png)

with a wide variety of robotic platform and provides a wide variety of services, including hardware abstraction, low-level device control, implementation of commonly-used functionality, message-passing between processes, and package management.

The basic Computation Graph concepts of ROS are nodes that perform tasks and computation. Master that provides naming and registration services, Parameter Server where nodes can store and retrieve parameters at runtime, messages which are used to communicate between nodes, services that provide message structure for request and reply, topics that identify the content of the message, and bags that is a mechanism for storing message data.

5.3 Gazebo

Gazebo is a robot simulation toolbox that has been develop since 2002 and is designed for testing algorithms through simulation in complex indoor and outdoor environments. It is developed by Open Source Robotics Foundation and it is primary a simulation environment in ROS. Gazebo has four possible physical engines. Open source Open Dynamic Engine(ODE) that is present since version 1.9 and Bullet physic library, Simbody a multibody physics API and Dynamic Animation and Robotics Toolkit (DART) that are present since version 3.0. Thanks to the open-source graphics rendering engine OGRE Gazebo provides high-quality rendering of environments, including lighting, textures and shadows.

5.4. RVIZ

Type of sensor	Quantity	Position	Description
Force/Torque	2	Ankle	6 axis F/T sensor
Sonar	4	Torso, Head	Range: 0.03 m – 4 m Resolution: 0.01m
Laser	2	Feet	Outreach: 5.6 m Pitch angle: 0.36° Frequency: 10 Hz
IMU	1	Body	5G acceleration Max turn: 450 deg/s
Microphone	4	Body	omnidirectional microphone
Front Stereo Camera	2	Head	Sensor type: CCD progressive 1/2" Max frame rate: 119 fps Resolution: 659x493 px Optics: 4.5 mm mount C lens
Back Camera	1	Head	Sensor type: CMOS global shutter 1/3" Max frame rate: 90 fps Resolution: 752x480 px Optics: 2.5 mm mount M12 lens

Table 5.1: REEM-C: Sensors and cameras

5.4 Rviz

Rviz is a 3D visualization toolbox of ROS that enables to visualize the robot and all data like point clouds, basic shapes, points, lines, etc. in a 3D space. In addition to visualization Rviz enables to interact with markers in real time to affect a simulation.

Chapter 6

Realization

After literature research it was clear that Robot Operating System (ROS) will be used for this work because it is a multi-platform and the most widespread robotic framework for robotic software development. Thanks to utilization of ROS, newly created packages can be transferred to another humanoid robot then the used one so ROS compatibility becomes a requirement for choose of humanoid robot.

6.1 Robot selection

Not much humanoid robots had the ROS compatible simulation model so the selection was not too difficult. I have found three robots that could be used and then I tried to find out which one to choose. Those robots were HUBO, REEM-C and Nao. Well done documentation and its often used all around the world speaks in favor of Nao. Another advantages were many successfully applied walking algorithms one of which is algorithm documented in paper [21]. Against Nao speaks its size thanks to it is much more difficult to overcome even small obstacle and unevenness and is not able to carry Kinect sensor that is widely used in search environment tasks.

HUBO was studied as the next. It is a child size robot with great mobility, that is able to hold Kinect in its hand. It is able to pass through uneven terrain as evidenced by participation in The DARPA Robotics Challenge. After importation into V-REP simulator it has been found that HUBOs simulation model [22] does not contain sensor and actuators. Without this sensory devices, the algorithm would have to be driven by data from the simulator and so it would not be applicable to the real robot.

The last studied robot was REEM-C. It is full size humanoid robot equipped with many sensory devices as a lasers or sonars. The big advantage of this robot is its full compatibility with ROS and its ability to hold Kinect or another device. The greatest disadvantage is almost nonexistent documentation except of a short tutorial and a few comments in code on GitHub [23].

6.2. SETUP OF THE DEVELOPMENT ENVIRONMENT

After careful examination of the information, REEM-C was selected for this work. NAO was not chosen because of its size and HUBO because of absence of sensors in its model.

Name	Nao	Hubo	Reem-C
Degrees of freedom	25	41	44
Height [cm]	58	125	165
Weight [kg]	4.3	45	80
Simulator	Gazebo Part of R-viz	Gazebo (without sensors) R-viz (without sensors)	Gazebo
Pros	A lot of researches A lot of documentation	A lot of researches Its size	ROS based Its size
Cons	Its size	Model without sensors	No documentation

Table 6.1: REEM-C: Sensors and cameras

6.2 Setup of the development environment

After selection of the robot, there was a need to select a simulator that will be used. Two possibilities come into consideration, Gazebo and V-REP. Both are great simulators but there are some differences. V-REP is able to show a customizable user interface and plot data streams. Gazebo have a lot of tutorials and it is a part of the ROS toolkit so it communicate with it natively. Because REEM-C model runs Gazebo, Gazebo was chosen for this work. My goal in setting the development environment, was to connect Rviz, Gazebo, ROS and Matlab and run REEM-C simulation on it. Simulation environment connection is shown in figure 6.1.

At the beginning I try to run REEM-C simulation at Ubuntu 14.04 with ROS Indigo Igloo but it didn't work and I failed to adjust it so I switched to the recommended version of Ubuntu 12.04 with ROS Hydro.

6.2.1 Installation of Ubuntu

After Installation of Ubuntu 12.04 and ROS hydro I found that Gazebo didn't run properly. It ran so slowly that it didn't allow to work. I did a research and found that it is caused by Optimus (Switch between two graphic cards) which Ubuntu 12.04 can not handle properly. I found posts with a similar issues but most of advices were to install latest Ubuntu (13.04 and newer) which has the problem solved but I needed to run Ros hydro and it does not support the newer Ubuntu then 12.04. I found on the forums that some people are helped to install a

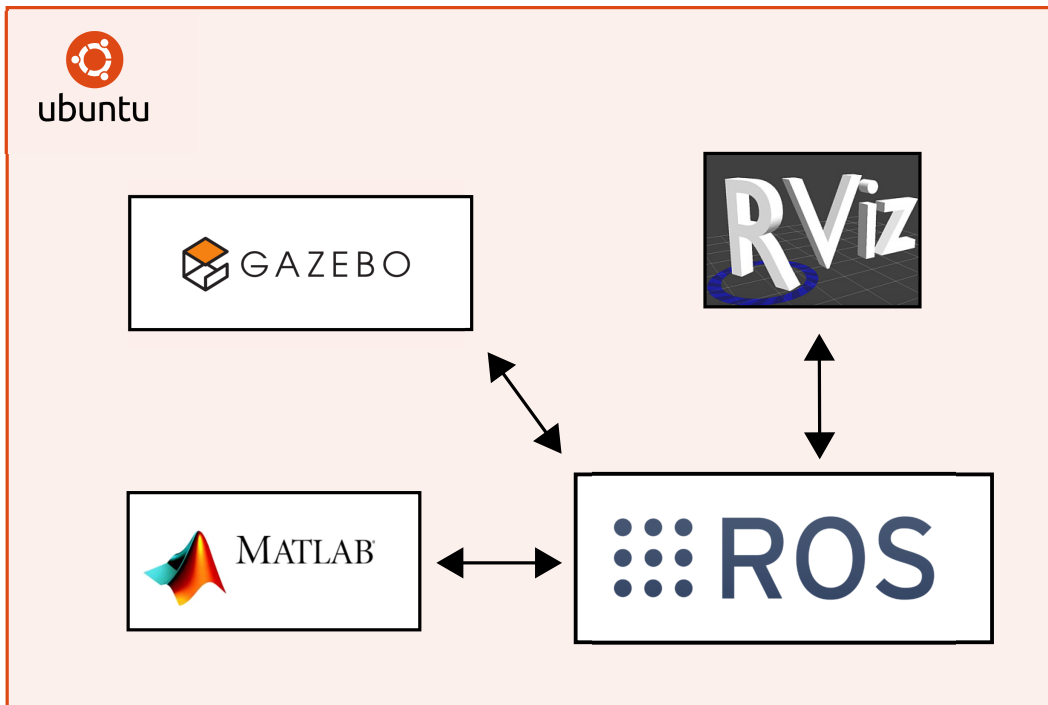


Figure 6.1: The required environment connection

newer driver from Canonical repository or from Nvidia website but nothing of these worked for me. Then I found that there are two alternatives for Optirun on Linux, Bumblebee and Nvidia Prime, that can manage integrated and dedicated graphics card.

I tried the Bumblebee as a first. After installation according to instructions on Ubuntu website [24], simulation ran tolerably but at the third to twentieth attempt (graphic card configuration errors were shown during start up of Gazebo) and when it successfully started, after a while it fell. Then I tried Nvidia Prime and it worked well but until the reboot of system. After reboot, X Server didn't start up and when I tried to start it up manually, the system reported that it found no monitor. After a while I realized that the problem is in `xorg.conf` file that was automatically rewritten by Nvidia Prime and I found that unrepaired bug of Prime which was reported . As a last chance I tried the Bumblebee once again (with `nvidia_304_updates` driver) and it works. The problem was the Bumblebee UI plugin that I installed previously on the first attempt and that caused the errors and falls.

6.2.2 Matlab

After the moment that I was able to ran Ubuntu with Bumblebee, I tried to set the simulation environment. Gazebo in version 1.9 and Rviz in version r1.10.19 are part of ROS hydro installation and they are both ready to communicate with ROS. The last missing part of the simulation environment was Matlab, the toolbox for design, simulation and analyses of systems

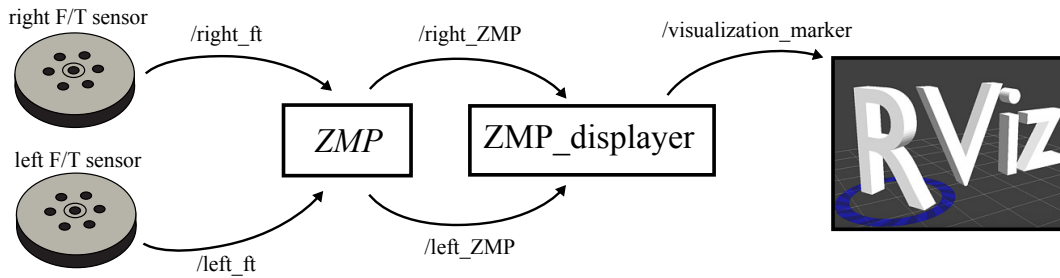


Figure 6.2: ZMP nodes structure

and controllers. In autumn 2014, a plugin for connection Matlab to ROS was freely available. Since then MathWorks published the paid Robotics System Toolbox [25] that replaced plugin and that was not a part of the university Matlab instalation. When I tried to find the plugin I failed and alternatives didn't works for me. For example IPC-Bridge [26] supports natively only a short list of messages. Therefore, I lost the possibility to use Matlab. Matlab was partly replaced by standard ROS tools for examble rqt_plot to generate graphs.

6.3 Application of walking

After familiarization with the ROS, the Rviz and the Gazebo I decided to start with a computation of the ZMP that will be needed later.

6.3.1 ZMP node

REEM-C is equipped with a force/torque (FT) sensors so the ZMP could be computed from theirs data. Because I didn't found any documentation for this robot I started by looking for the FT sensor node in the ROS or for the topic to which the sensor data were published. This topic was found after while. Topics for the right and the left FT sensors are named `/right_ft` and `/left_ft` and both are `geometry_msgs/WrenchStamped` type. Based on this information I created ZMP a node that subscribe the FT topics and that are publishing `/left_ZMP` and to the `/right_ZMP` topics mesages of `geometry_msgs/PointStamped` type. These messages contains x , y and z values of ZMP where the z value is always zero. The computation of the ZMP is described in a section 4.3.2, equations 4.9 and 4.10.

Then I created ZMP_displayer a node that subsribed `right_ZMP` and `right_ZMP` and were publishing to `/visualization_marker` topic that enables in Rviz to display the positions of ZMP points on the REEM-C model. Whole structure is shown in the figure 6.2.

6.3. APPLICATION OF WALKING

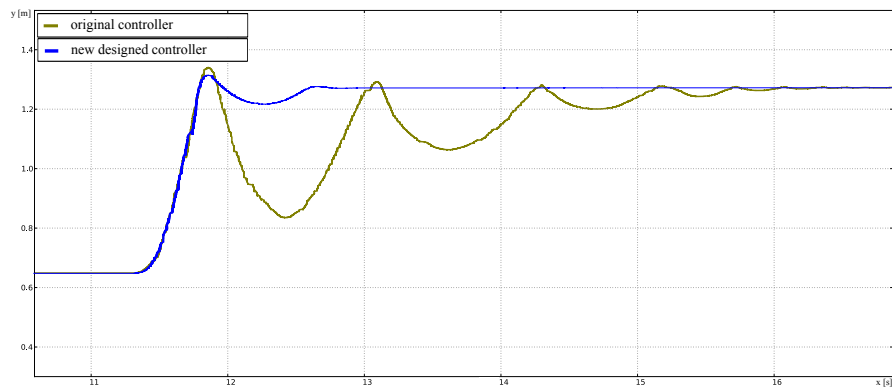


Figure 6.3: The comparison of new and original arm control in response to a step

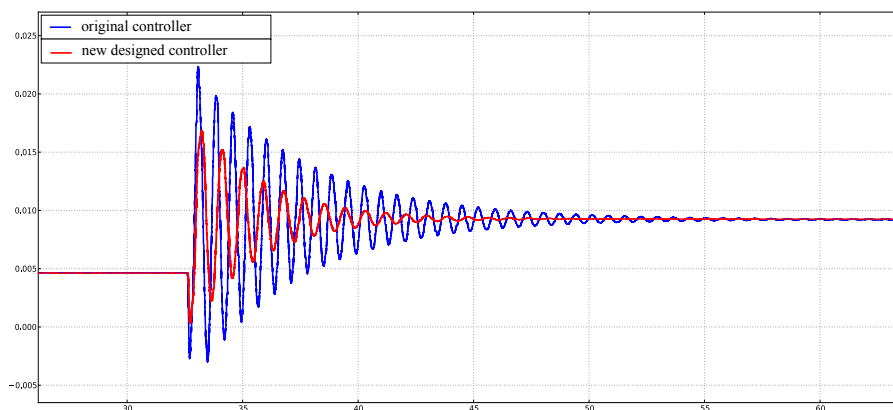


Figure 6.4: The comparison of new and original leg control in response to a step both with new control of arms running

6.3.2 Control of motors

For control of REEM-C is used `ros_control` package that enables a controller management and include a control toolbox. In the original algorithm all motors were PD controlled. I tried PD and PID and for REEM-C robot were better results with PID controller (lower amplitude of oscillation).

First, I set the PIDs of all joints of both arms to the same values and tune them all together manually. Then I tried to change the values of a single controllers little bit to achieve better results and then applied the same approach to the legs and to the torso controllers. Controllers of arms were tuned up with original values of controllers in legs, torso, fingers and head. Controllers of legs were tuned with the new values in the arm controllers and the original in rest of the body. Controllers of torso were tuned with new values in arm and leg controllers and original in rest of body. I managed to get better settings of controllers on arms, legs and torso than were original ones that were set by default by PAL Robotics. Controllers of a fingers and head were let in their default values without change because they have no influence on the robot stability.

6.3. APPLICATION OF WALKING

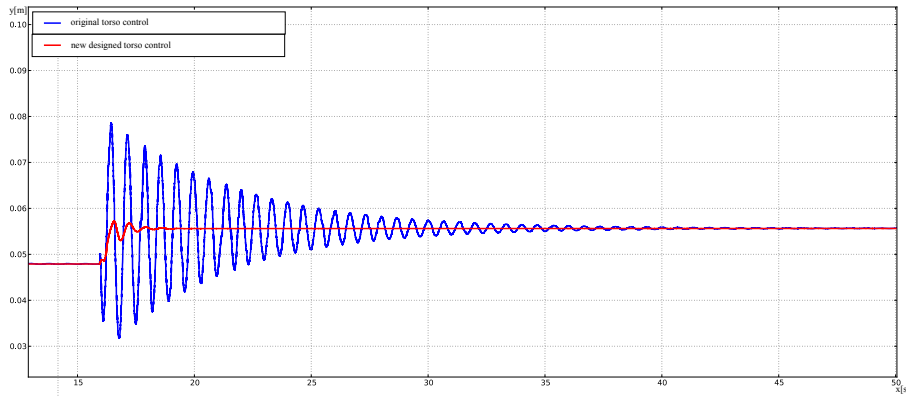


Figure 6.5: The comparison of new and original torso control in response to a step both with new control of legs and arms running

The comparison of the original and the new arm, leg and torso control are shown in graphs 6.3, 6.4 and 6.5. Values of both, original and new controllers are shown in table 6.2.

Joints	New			Original		
	P	D	I	P	D	I
left/right leg 1st joint	1700	20	0.001	3000	10	1
left/right leg 2nd joint	1700	20	0.001	3000	10	1
left/right leg 3rd joint	1700	20	0.001	3000	10	1
left/right leg 4th joint	1900	21	0.001	3000	10	1
left/right leg 5th joint	1900	21	0.001	3000	10	1
left/right leg 6th joint	1900	20	0.001	3000	10	1
torso 1st joint	200	35	0.001	10000	10	1
torso 2nd joint	200	35	0.001	10000	10	1
left/right arm 1st joint	350	15	2	3000	5	1
left/right arm 2nd joint	350	15	2	3000	5	1
left/right arm 3rd joint	350	15	2	3000	5	1
left/right arm 4th joint	350	15	2	3000	5	1
left/right arm 5th joint	500	1	5	500	2	0.1
left/right arm 6th joint	500	1	20	500	2	0.1
left/right arm 7th joint	500	1	20	500	2	0.1

Table 6.2: Table of values of original and new PIDs

6.3. APPLICATION OF WALKING

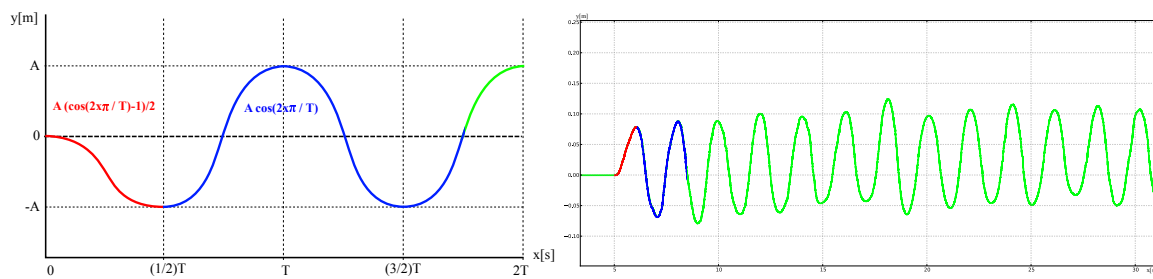


Figure 6.6: Lateral displacement of pelvis. On the left are the data of algorithm, on the right are the real data of simulation

6.3.3 Leg movements

To avoid singularity of inverse kinematics, bending the knees and lowering of pelvis was needed. For this propose, I lower the pelvis of robot about 1.5 cm. From the equation 4.1 for calculating the natural frequencies of the inverted pendulum was calculated natural frequency of the robot which was approximately 0.5 Hz. For REEM-C the portion of the double-support phase was experimentally set to 30%. The lateral swing amplitude of the the pelvis was experimentally set to 11 cm. For comparison, values of HUBO robot were 0.526 Hz natural frequency, 5% double-support phase portion and 6 cm lateral displacement.

For fluent walk of robot, create starting sequence that enables dynamically smooth animation of COM was needed to. This sequence was created with use of cosine function and is shown in figure 6.6 where A is length of amplitude and T is period. Starting sequence is composed form two function. First is cosine function with length of amplitude equals 5.5 cm shifted to the negative (shown red). Second part is cosine with length of amplitude equals to 11 cm (shown blue). Real data from a graph of the simulation are deformed. That is caused by interventions of the controllers.

The starting sequence is followed by walking sequence itself. Legs are raised up during this sequence with experimentally set double-support portion. To perform smooth movements are legs raised up with cosine function too. Whole motion is shown step by step in figure 6.7.

The rest of the algorithm I could not implement due to lack of time caused by complications in setting up the simulation environment.

6.3. APPLICATION OF WALKING

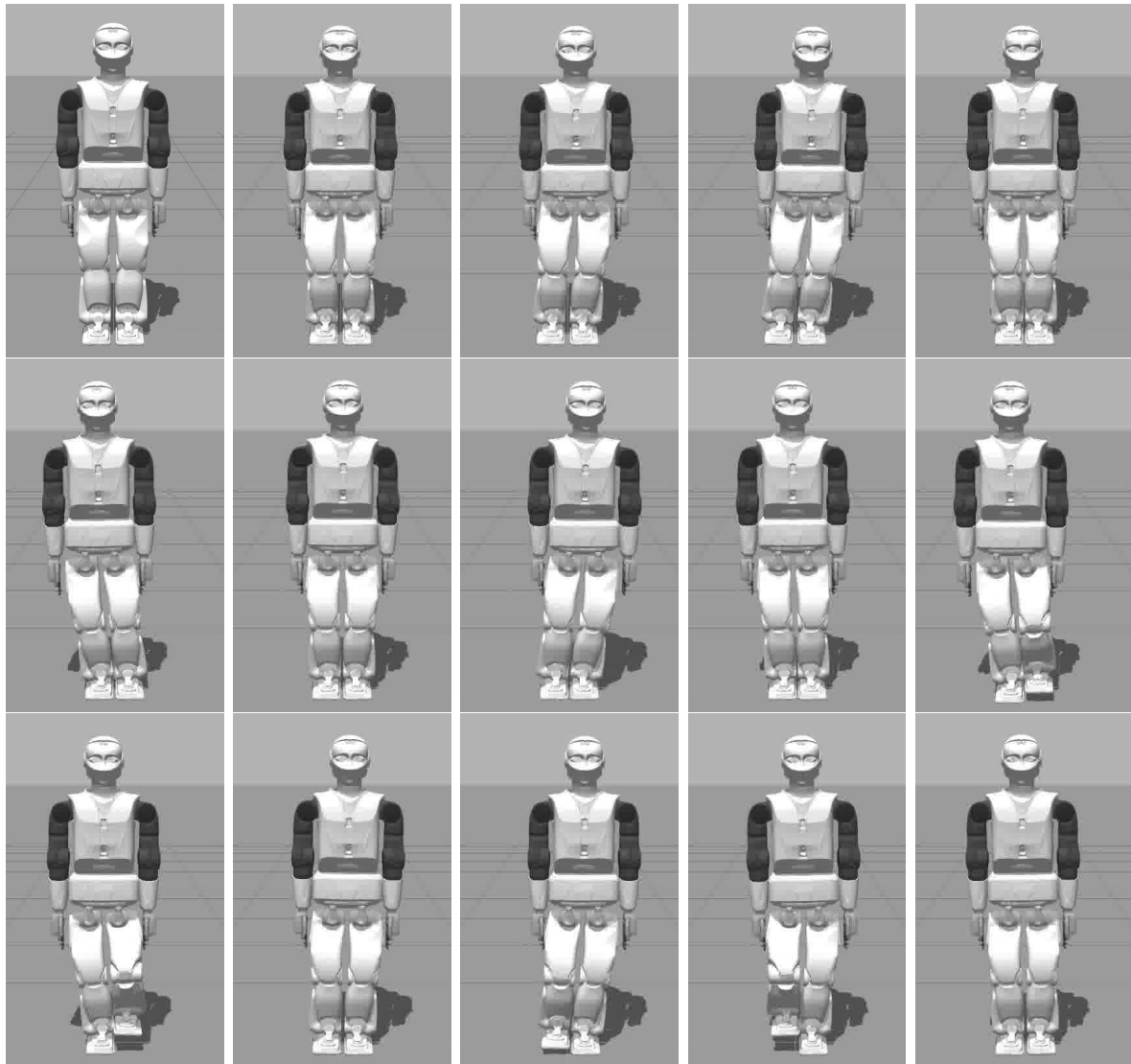


Figure 6.7: Walk of REEM-C

Chapter 7

Conclusions

The robot chosen for this work was a REEM-C made by PAL Robotics. The robot was chosen because of its size, degrees of freedom, an ability to hold and carry the Kinect and because of an ability to run the Robotic Operating System. As an algorithm for implementation, approach designed by Jung-Yup Kim, Ill-Woo Park and Jun-Ho Oh [4] was chosen. Working simulation setup of OS for computer with Nvidia Optimus was found too. This setup is Ubuntu 12.04 with `nvidia_304_updates` driver and Bumblebee installed without additional plugins. Simulation environment for Ubuntu 14.04 with ROS Indigo Igloo was issued recently. In the environment Nvidia Optimus works fine. ROS Hydro with Gazebo as a simulator and Rviz for displaying of the ZMP were used in the development environment.

For the REEM-C were designed PID controllers to achieve better stability in standing pose. These controllers were designed for motors of legs, arms and torso. In figures 6.3, 6.4 and 6.5 there is shown that the newly designed controller provide better control then the original ones designed by PAL Robotics. Furthermore the algorithm for a calculation and display of ZMP was successfully applied. For chosen robot was designed the starting sequence that enables to COM dynamically smooth motion. Then walking sequence with use of cosine function was designed. It enables to the robot stable walking on the spot. The rest of the algorithm (stabilization and walk forward) was not implemented due to lack of time caused by complications in setting up the development environment.

Next steps should be design of controllers that are described in chapter 4. Thanks to these controllers stable walking on an uneven surfaces could be achieved. The most important controller is the ZMP controller that uses the ZMP as a reference. After the algorithm application an ability of the robot can be extended by applying turning, walking backwards or by climbing stairs. Robot would be able to search environments or transfer stuffs with these abilities.

Bibliography

- [1] Introduction to Mechanisms of Carnegie Mellon University. <https://www.cs.cmu.edu/~rapidproto/mechanisms/chpt4.html>. Accessed: 2015-05-05.
- [2] Jong Wook Kim. Online joint trajectory generation of human-like biped walking. *International Journal of Advanced Robotic Systems*, 11:1–12, 2014.
- [3] T Geng, B Porr, and F Worgotter. Fast biped walking with a sensor-driven neuronal controller and real-time online learning. *International Journal of Robotics Research*, 25, March 2006.
- [4] Jung-Yup Kim, Ill-Woo Park, and Jun-Ho Oh. Experimental realization of dynamic walking of the biped humanoid robot KHR-2 using zero moment point feedback and inertial measurement. *Advanced Robotics*, 20(6):707–736, 2006.
- [5] Qiang Huang, Kazuhito Yokoi, Shuuji Kajita, Kenji Kaneko, Hirohiko Aral, Noriho Koyachi, and Kazuo Tanie. Planning walking patterns for a biped robot. *IEEE Transactions on Robotics and Automation*, 17(3):280–289, 2001.
- [6] Shuuji Kajita, Mitsuharu Morisawa, Kanako Miura, Shin Nakaoka, Kenji Kaneko, Fumio Kanehiro, and Kazuhito Yokoi Aist. Biped Walking Stabilization Based on Linear Inverted Pendulum Tracking. In *The 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 2, Taipei, Taiwan, October 2010.
- [7] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa. Biped walking pattern generation by using preview control of zero-moment point. *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, 2:1620–1626, 2003.
- [8] Kemalettin Erbatır and Okan Kurt. Natural ZMP trajectories for biped robot reference generation. *IEEE Transactions on Industrial Electronics*, 56(3):835–845, 2009.
- [9] P. Sardain and G. Bessonnet. Forces acting on a biped robot. Center of pressure-zero moment point. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 34(5):630–637, 2004.

BIBLIOGRAPHY

- [10] MIOMIR VUKOBRATOVIĆ and BRANISLAV BOROVIAC. Zero-moment point — thirty five years of its life. *International Journal of Humanoid Robotics*, 02(02):225–227, 2005.
- [11] Martijn Wisse, Arend L. Schwab, Richard Q. van der Linde, and F. C T van der Helm. How to keep from falling forward: Elementary swing leg action for passive dynamic walkers. *IEEE Transactions on Robotics*, 21(3):393–401, 2005.
- [12] Joel Chestnutt, Manfred Lau, German Cheung, James Kuffner, Jessica Hodgins, and Takeo Kanade. Footstep planning for the Honda ASIMO humanoid. *Proceedings - IEEE International Conference on Robotics and Automation*, 2005:629–634, 2005.
- [13] Sang Ho Hyon, Joshua G. Hale, and Gordon Cheng. Full-body compliant human-humanoid interaction: Balancing in the presence of unknown external forces. *IEEE Transactions on Robotics*, 23:884–898, 2007.
- [14] ASIMO. <http://en.wikipedia.org/wiki/ASIMO>. Accessed: 2015-02-02.
- [15] Ill-Woo Park, Jung-Yup Kim, Jungho Lee, Min-Su Kim, Baek-Kyu Cho, and Jun-Ho Oh. *Development of biped humanoid robots at the humanoid robot research center, korea advanced institute of science and technology (KAIST)*, chapter 3. InTech, June 2007.
- [16] The DARPA Robotics Challenge official website. <http://www.theroboticschallenge.org/>. Accessed: 2015-02-02.
- [17] Article about new ATLAS. <http://spectrum.ieee.org/automaton/robotics/military-robots/atlas-drc-robot-is-75-percent-new-completely-unplugged>. Accessed: 2015-02-02.
- [18] M VUKOBRATOVIĆ and J Stepanenko. On the stability of anthropomorphic systems. *Mathematical Bioscience*, 15:1–37, October 1972.
- [19] REEM-C tutorials at official ros website. <http://wiki.ros.org/Robots/REEM-C/Tutorials/>. Accessed: 2015-20-03.
- [20] Official Aldebaran documentation of its robots. <http://doc.aldebaran.com/2-1/index.html>. Accessed: 2015-20-03.
- [21] David Gouaillier, Cyrille Collette, and Chris Kilner. Omni-directional closed-loop walk for NAO. *2010 10th IEEE-RAS International Conference on Humanoid Robots, Humanoids 2010*, pages 448–454, 2010.
- [22] GitHub of HUBO model. <https://github.com/wmhilton/hubo-urdf>. Accessed: 2015-01-05.
- [23] GitHub of PAL Robotics. <https://github.com/pal-robotics/pal-ros-pkg/>. Accessed: 2015-01-05.

BIBLIOGRAPHY

- [24] Bumblebee installation guide. <http://wiki.ubuntu.cz/bumblebee>. Accessed: 2015-01-05.
- [25] Robotics System Toolbox of Matlab. <http://www.mathworks.com/products/robotics/>. Accessed: 2015-01-05.
- [26] IPC-Bridge between ROS and Matlab. <https://alliance.seas.upenn.edu/~meam620/wiki/index.php?n=Roslab.IpcBridge#Installationn>. Accessed: 2015-05-05.

Appendix

CD Content

In table 7.1 are listed names of all root directories on CD

Directory name	Description
sim	Simulation environment folder
Thesis.pdf	Text of Bachelor thesis
Readme.txt	Readme file

Table 7.1: CD Content