MASTER THESIS

The Mechanics and Control of Leaning to Lift Heavy Objects

carried out at
Information Technology and Systems Management
at Salzburg University of Applied Sciences
and
the Robotics Institute of
Carnegie Mellon University, Pittsburgh

written by:
Fabian Sonnleitner, BSc

FH Salzburg
Informationstechnik &
System-Management

Program Director:	FH-Prof. DI. Dr. Gerhard Jöchtl
Supervisor (CMU):	Prof. Ralph Hollis
Supervisor (FHS):	FH-Prof. DI. Dr. Thomas Kurz

Pittsburgh, August 2018

Declaration of Authorship

I, Fabian Sonnleitner, born on the 16th July, 1994 in Salzburg, confirm that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

Pittsburgh, PA on August 26, 2018

Fabian SONNLEITNER

Matriculation Number

Acknowledgements:

I would like to thank Prof. Ralph Hollis for the opportunity to work with him and his amazing inventions. I really enjoyed working on the ballbot as well as the great trip to the air show.

My deepest gratitude goes to Roberto Shu, who did not just help me with all my technical questions but also included me in his circle of friends. You made this six months to fantastic six months. I will definitely not forget all the fun adventures we had and I am looking forward to seeing you in Austria.

I would like to thank all my friends at Carnegie Mellon University for making the six months I stayed here unforgettable. I would also like to thank Zhongyu Li and Sebastian Schoendorfer for their great help in conducting the ballbot experiments. Additionally, I would like to thank Thomas Kurz for the great support and advise from overseas.

Furthermore, I want to thank my family and my friends for supporting me all the time from back home.

General Information

First Name, Surname:	Fabian Sonnleitner
University:	Fachhochschule Salzburg GmbH
Degree Program:	Informationstechnik & System-Management
Title of Thesis:	The Mechanics and Control of Leaning to Lift Heavy Objects
Keywords:	Ballbot, dynamic stability, balancing mobile robots, mobile manipulator, human interaction, transportation
Supervisor:	FH-Prof. DI. Dr. Thomas Kurz

Abstract

To be truly useful, mobile robots need to be able to lift and transport objects to collaborate with humans in work and home environments. This thesis presents how to successfully detect, lift and transport a heavy object with a dynamically stable mobile robot. Heavy repetitive lifting is still a major health risk for workers. Further, present mobile manipulators are either too weak to carry heavy objects $>$10 kg or they have wide bases that make them difficult to navigate in narrow work or home environments. Therefore, ballbot's small footprint, strong arms and the capability to exert forces by inducing a lean angle make it a fitting robot to take over those demanding lifting tasks. The goal of this work is to develop a control algorithm to enable the CMU ballbot with its 2 DOF arms to detect a heavy object, navigate to it, lift it and transport it to a desired location semi-autonomously. Previous works have successfully demonstrated two-wheeled dynamically stable mobile manipulator robots transporting heavy objects. The novelty of this thesis presents the use of CMU ballbot as the first single-wheel dynamically stable omnidirectional robot to achieve such task. A successful semi-autonomous lift and transport of a 15 kg heavy box was achieved using a combination of feedforward and feedback control laws based on a quasi-static center of mass computation. Ballbot's pan and tilt sensor turret tracked fiducial markers on the box. Ballbot-to-human and human-to-ballbot exchange of a heavy object was achieved while dynamically balancing.

Contents

Acknowledgements . . . ii

Abstract . . . iii

List of Figures . . . viii

List of Tables . . . ix

1 Introduction 1

1.1 The CMU Ballbot . . . 1

1.2 Motivation . . . 1

1.3 Goal and Approach . . . 3

2 The CMU Ballbot 4

2.1 History . . . 4

2.2 Hardware and Electronics . . . 5

2.3 Software and Controls Overview . . . 10

2.4 Arm controller . . . 11

2.5 Balance controller . . . 11

2.5.1 Balancing . . . 11

2.5.2 Station keeping . . . 12

2.5.3 Outer Loop Control . . . 12

2.5.4 Differential Flatness Path Planning . . . 13

2.6 System Identification . . . 14

2.7 Dynamic model of the ballbot . . . 15

3 Related work 17

3.1 Object detection . . . 17

3.1.1 Object detection with ARUCO fiducial tracking . . . 17

3.1.2 Object detection using a 2D lidar and a camera . . . 18

3.1.3 Markerless visual servoing on unknown objects for humanoid robot platforms . . . 18

3.2 Lifting . . . 19

3.2.1 Human lifting motion . . . 19

3.2.2 Human interaction and feedback . . . 20

3.2.3 Humanoid robots . . . 21

3.2.4 Two wheeled dynamically stable robots . . . 22

3.2.5 Mobile manipulators . . . 24

4 Approaches 26
4.1 Human behaviour 26
4.1.1 Object detection 26
4.1.2 Lifting and transporting 27
4.2 High level approach 27
4.3 Object detection using Fiducial tracking 31
4.4 Motion planning 32
4.4.1 Path Planning and Yawing 32
4.4.2 Arm movement 34
4.5 Object Mass detection 36
4.5.1 Calculation 36
4.5.2 Polynomial estimation 37
4.5.3 Detection through object identification 37
4.5.4 Control based lean angle compensation 38
4.6 Center of mass feed forward control 39

5 Results 41
5.1 2D Simulation 41
5.2 Object detection 43
5.3 Arm controller 45
5.4 Weight detection and limits 45
5.4.1 Maximum liftable mass with the current setup 46
5.4.2 The accuracy of the mass detection 47
5.5 Lean angle compensation 50
5.6 Lifting and carrying tasks 51
5.6.1 Delivering an object to a human 52
5.6.2 Receiving an object from a human operator 53
5.6.3 Transporting an object between two different locations 54
5.6.4 Lifting vs. Grabbing 55
5.6.5 Heavy weight transport 56

6 Conclusion and outlook 57
6.1 Conclusion 57
6.2 Future Work 58
6.2.1 Lidar 58
6.2.2 Object detection 58
6.2.3 Lifting and path planning 58

Bibliography 60

Appendices **66**
A Inertia measurement 67
B Ballbot transformation tree 69

List of Figures

1.1 The CMU ballbot 2

2.1 CMU Ballbot balancing with one arm raised 6
2.2 Pan and tilt turret with RGBD camera and 2D lidar 6
2.3 CAD model of the four-motor inverse-mouse ball drive with included yaw drive. (Courtesy: Ralph Hollis; Appeared in [1]) 7
2.4 Ballbot's 2-DOF arms. 8
2.5 Different attachable hands for ballbot's 2 DOF arms. 9
2.6 Software and PC overview 10
2.7 Balance controller overview 12
2.8 Simplified Ballbot model (Courtesy: Ralph Hollis; Appeared in [2]) 13
2.9 Ballbot's configuration shown in a two dimensional model of ballbot with arms (Courtesy: Ralph Hollis; Appeared in [3, 4]) 16

4.1 Human lifting approach. 28
4.2 Comparison of the ballbot simulation lifting a object with 20 kg 29
4.3 Step by step approach 30
4.4 Object detection and localization with ARUCO fiducial tracking 32
4.5 Path planing visualization in the Rviz simulation 33
4.6 Teach and Play comparison: command vs.real angle 35
4.7 Arm y movement for lifting a box with handles 35
4.8 Calibration trajectories generated in Matlab^{TM} 37
4.9 Determination of the polynomial function using Matlabs^{TM} curve fitting tool. 38
4.10 Center of mass compensation and lean angle control overview 39

5.1 2D Matlab ^{TM} simulation of station keep mode at three different times, while raising its arms. 41
5.2 Plot of lean angle, arm angle and ground position over time. 42
5.3 2D Matlab ^{TM} simulation of station keep mode at three different times, while raising its arms with 50% less detected weight. 42
5.4 Plot of lean angle, arm angle and ground position over time with 50% less detected weight. 43
5.5 Results: Object detection and localization with ARUCO fiducial tracking . 44
5.6 Left plot: arm command vs. real angles while lifting 2kg on each hand. Right plot: arm position error at a certain angle while performing a lifting movement. 45

5.7 Drawing of arm construction using Series Elastic Actuators. (Courtesy: Ralph Hollis; Appeared in [4]) 47
5.8 Object mass calculation while lifting. Mass used for test was 2kg per hand. 48
5.9 Object mass estimation while lifting. Mass used for test was 2kg per hand. . 49
5.10 Lifting of 2kg per hand. 50
5.11 Lean angle compensation without knowing or detecting the lifted mass. Lifting duration of 10 and 20 seconds 51
5.12 Selected screen-shots: Human receiving a 10 kg heavy object from the ballbot. 52
5.13 Selected screen-shots of the video: Test person gives the ballbot a object to transport while balancing. 53
5.14 Selected screen-shots of the video: Transporting an object from A to B. . . 54
5.15 Lifting vs. Grabbing 55
5.16 Maximum experimentally detected lean angle 56

List of Tables

2.1 I-Beam Parameters 14
2.2 Ballbot Parameters 15

5.1 Average error while preforming a lifting to 90° and back. The weight is attached to each arm. 46
5.2 Weight estimation by polynomial function 49

1 Introduction

Mobile robots which can tidy up the house, help in the kitchen or transport heavy objects for humans are almost available. At least this is the general sentiment of people. But in reality, there are only very few robots which can detect and transport objects while being fast, agile and able to be used in a human environment [5–7]. Most of the existing mobile manipulation robots rely on wheels or tracks to travel. Therefore they need a big base to support center of mass shift. Additionally, most of those robots are not able to drive into any commanded direction without adjusting their yaw angle first [8–11].

In 2005 a new type of balancing robots was invented by Ralph Hollis [2, 12, 13]. Instead of using wheels, feet or tracks this new type featured a single point of contact to the ground using a sphere. This new robot, called the ballbot, set new goals and possibilities in robotics and is the robot used in this theses. Due to its single point of contact, it is omnidirectional, and can, therefore, drive in any direction at any time. Additionally, the ballbot has a very small base, despite being as tall as a human.

This thesis will investigate how the ballbot is able to detect, lift and transport objects of unknown weight. Further, the maximum lifting capabilities of this robot shall be investigated. This concludes to the main research question: How is a single wheeled, dynamically stable, mobile robot, such as the ballbot able to successfully lift and transport a heavy object?

1.1 The CMU Ballbot

Unless otherwise specified, the use of ballbot refers to the CMU ballbot during the entire thesis. The ballbot is located at the Robotics Institute of Carnegie Mellon University. Due to the newly invented inverse mouse-ball drive, the ballbot was the first dynamically stable robot with an omnidirectional drive. This means, that the ballbot can drive in any direction without needing to adjust its position prior. This is a big benefit compared to other robots which use wheels or tracks to maneuver [2, 8–13]. Furthermore, this innovative drive and balance mechanism allows a human height while still having a very small, human-like, footprint [14].

In 2012 a pair of two-degree-of-freedom (2 DOF) arms were added [3, 4]. This upgrade enabled the ballbot to help a person to perform a sit-to-stand maneuver in 2015 [15]. Figure 1.1 shows the CMU ballbot in its dynamically stable balancing state.

1.2 Motivation

This thesis shall help to develop the ballbot from a professionally supervised helper, one step closer to an autonomous, self-surviving and acting system. Mobile robots, such as

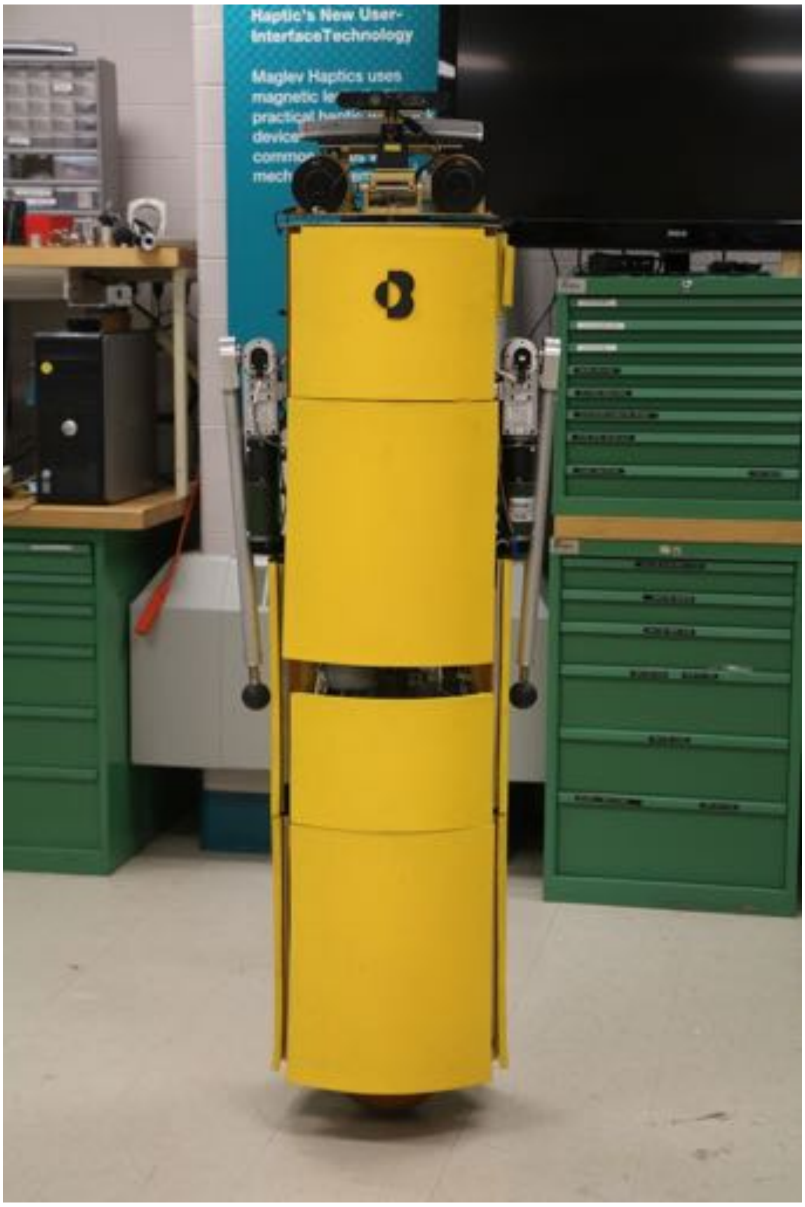


Figure 1.1: The CMU ballbot

the ballbot need to be able to interact with an object and their environment to be truly useful. While over the last decades many robot bases, which allow payloads to be placed on them, were developed, it is important to add arms to those robots and enable them to manipulate different objects. The ballbot is an ideal platform, as it is, compared to most robotic platforms, omnidirectional and has a very small footprint while having a human size.

Till now, the CMU ballbot showed, that it is able to use its arm for leading and following tasks. This ability works in both directions, so either the robot can follow a human by "lead it by the hand" or the other way round. Using its 2 DOF arms, the ballbot is able to assist a human operator in dynamic carrying tasks. Additionally, it was presented, that the ballbot can assist a human to stand up from a chair. During this so-called, sit-to-stand maneuver, the ballbot leans back while the human holds onto the ballbot's arms. This introduces a momentum which helps the human to stand up.

The ballbot shall now receive the ability to detect and transport an object without the need of a human interaction. According to our knowledge, this would be the first time, that a ballbot type robot is able to independently carry objects.

An additional benefit to the small footprint, human size, and omnidirectional maneuverability is the human interaction. As the human size robot needs to lean back during the lifting maneuver, an external observer is able to estimate the object's weight.

1.3 Goal and Approach

The goal of this thesis is to investigate the ballbot's ability of detecting, grabbing and transporting objects while staying gracefully balanced. It is important that the ballbot will still have its graceful balancing during all steps of the lifting motion, as humans feel uncomfortable around robots with high jerk motions. Humans tend to associate those non-smooth or ungraceful motions as panic motions [16].

Chapter 2 will provide a good understanding of the current ballbot system. It will show what tasks the ballbot can already master and which parts of a successful object detection and transport are missing.

Chapter 3 will focus on the related work for the two main task, the object detection, and localization as well as the actual lifting and transport. The risks and issues with humans performing repetitive lifting task will be assessed. Additionally, this chapter will provide an overview of the capabilities of current mobile manipulation robots. Further, statically stable robots, which do not need to adjust to the transported mass, will be discussed to explore different grabbing and lifting approaches.

The learning's and ideas from previous steps will be used in Chapter 4 to generate the ideal approaches for the ballbot. The implementations for the object detection and localization, as well as the motion planning will be presented. Furthermore, this chapter will show, how the ballbot can detect the weight of the object and calculate the corresponding lean angle.

Chapter 5 will compare the implementations form the previous chapter against each other and present their benefits and drawbacks. After selecting the best solutions for each task a variety of experiments is performed to see the new real-world applications for the ballbot.

The last chapter will summarize the new abilities of the ballbot, discuss conquered issues and will give an outlook for future improvements.

2 The CMU Ballbot

This chapter will provide a better understanding of the ballbot and its components. The first part will give some information about the ballbot's history and other robots which are balancing on a sphere around the world. The following section shall provide an overview of CMU ballbot's hardware and electronic system. The different sensors, actuators, and controllers will be described in detail and how they are connected to each other. The next few sections present the current software and control stack. An overview of the full setup as well as a deeper dive into the needed controls for the arms and the balancing will be presented. The last section of this chapter will focus on the physical properties and the dynamic model of the ballbot. This is needed as many components were added or changed since the ballbot was fully measured the last time in 2009 [17].

2.1 History

The CMU ballbot, which is located at the Microdynamics Systems Laboratory in the Robotics Institute of Carnegie Mellon University, Pittsburgh, PA, USA, was invented by the founder and director of this Laboratory Prof. Ralph Hollis. The full concept of the dynamically stable mobile robot is described in [2, 12, 13] and mainly contains an inverse mouse-ball drive, an inertia measurement unit, a control PC, and batteries. The individual components, their purpose, and their connections will be described in more detail in the following chapter. The CMU ballbot is human size and can withstand hard and soft pushes by humans [18]. It showed as well in [19, 20] that it can navigate in human environments and avoid obstacles.

Shortly after the successful balancing of the Ballbot, Schearer started simulations on the ballbot's behavior without and with arms[14]. These results were used to develop a real pair of arms for the ballbot. The 2 DOF arms [3, 4, 21], introduced in 2011, opened more possibilities for the ballbot. The 2013 presented shape space planer used those arms to navigate. Furthermore, this planer enables arm movement while keeping its position with dummy weights attached to the arms [21]. This thesis will extend those capabilities to lift an object with unknown masses and transport them. In 2015 the ballbot successfully showed, that it can assist humans to get up from a chair. For this so-called sit to stand maneuver a sitting human holds onto ballbot's arms. The ballbot then starts leaning backwards which introduces an upwards and forwards momentum for the human [15]. At the same time, Vaidya developed a center of mass offset compensation which enabled the ballbot to navigate on slopes of up to 4° [22].

After introducing the CMU ballbot many different adaptions of ball-balancing robots were developed and tested. The following paragraph will showcase a selection of them. The prototype presented in [23] was able to balance on a sphere, but could not be navigated to a certain location. At the same time in 2005, Endo presented an omnidirectional vehicle on a Basketball. This vehicle can be considered as some kind of a wheelchair, as it is designed to transport a human [24]. In 2008 Kumagai and Ochiai presented two different robots. The first is a robot which can balance on a roll, similar to a 2D ballbot. The second robot called BallIP is a small ballbot of approximately 0.5 m height [25, 26]. The ballbot named Rezero from ETH Zurich measures approximately 1 m and was presented in 2013 [27]. In 2014 this robot was used to develop a non-linear model predictive motion control [28]. Pouya Asgari published 2013 and 2014 two papers [29, 30] in which a simulation of a ballbot with a 3 DOF manipulator is presented. It presents the center of mass offset problem while extending the manipulator and proposes a pendulum and a manipulation controller to avoid unwanted drifting of the simulated ballbot.

2.2 Hardware and Electronics

This section will provide a detailed understanding of ballbot's individual components and their connections. The CMU ballbot is designed with three aluminum channels angled 120° to each other and several decks to mount the components, as it can be seen in Figure 2.1. Each channel contains one leg which allows the ballbot to have a stationary stable state while its power is shut down. During balancing the legs are retracted and barely visible.

Power

The CMU ballbot is a 48 V system. It has four 12 V lead acid batteries in series which can be charged by the onboard charger. This ensures that ballbot can be charged at any wall outlet without any additional equipment. As not all components are able to take the 48 V a power distribution board converts it to lower voltages for each component. An emergency stop switch located near the top of the ballbot cuts the power to the ball motor amplifiers and signals its status to the QNX PC. With this current battery setup, the ballbot is able to balance about 2 hours. If critical low voltages are reached an audio alert will be played and the ballbot transitions from balancing to a stationary mode with its legs down.

Turret

The turret on top of the ballbot contains a lidar, an RGBD camera, two speakers and a microphone array. As it can be seen in Figure 2.2 the lidar and the camera are mounted on top of a 90° tilt mechanism. This and the pan, with a range of ±330°, allows both sensors an unobstructed view around the ballbot. The speakers are used to announce the next movements and warn in case of a detected issue.

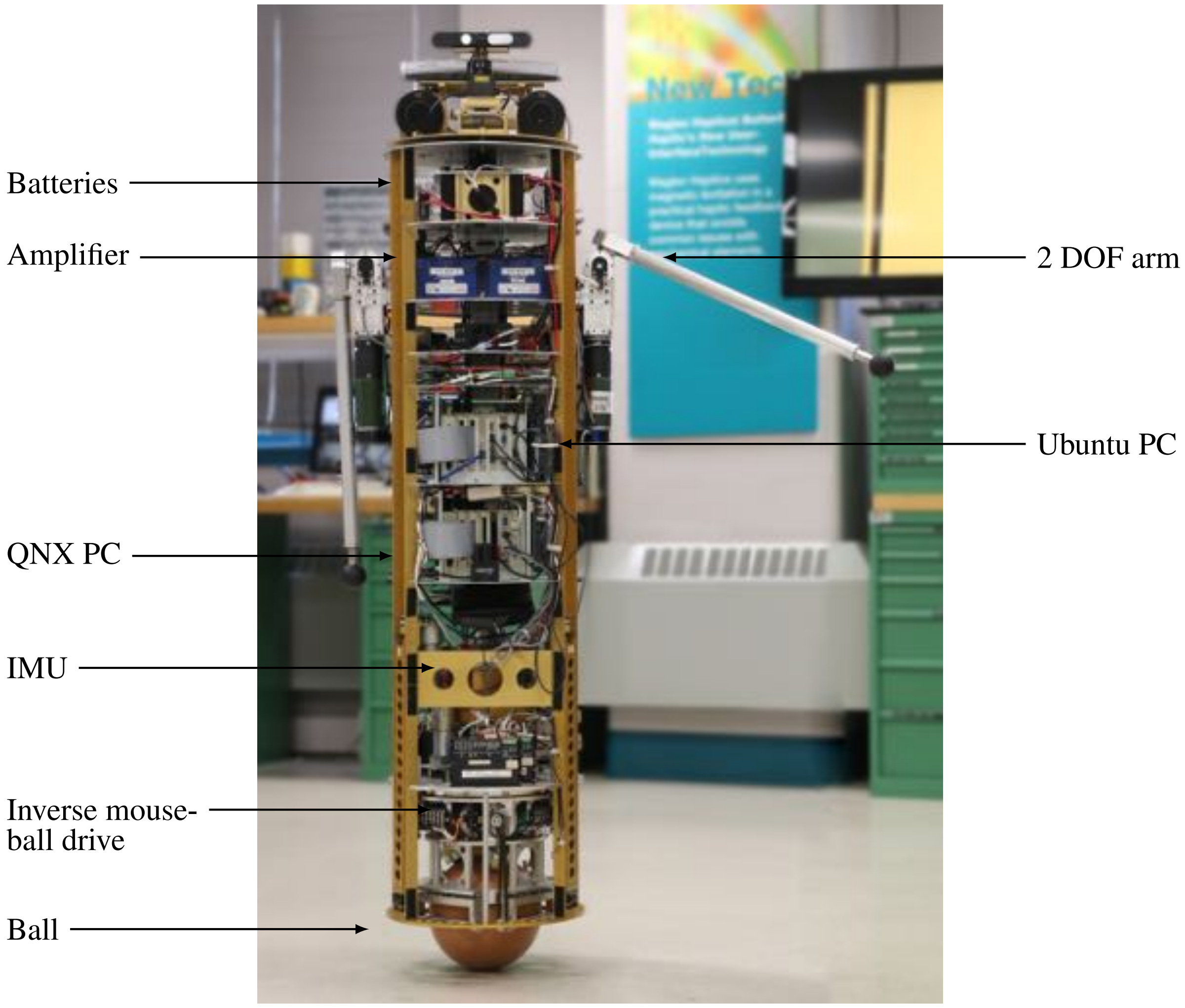


Figure 2.1: CMU Ballbot balancing with one arm raised

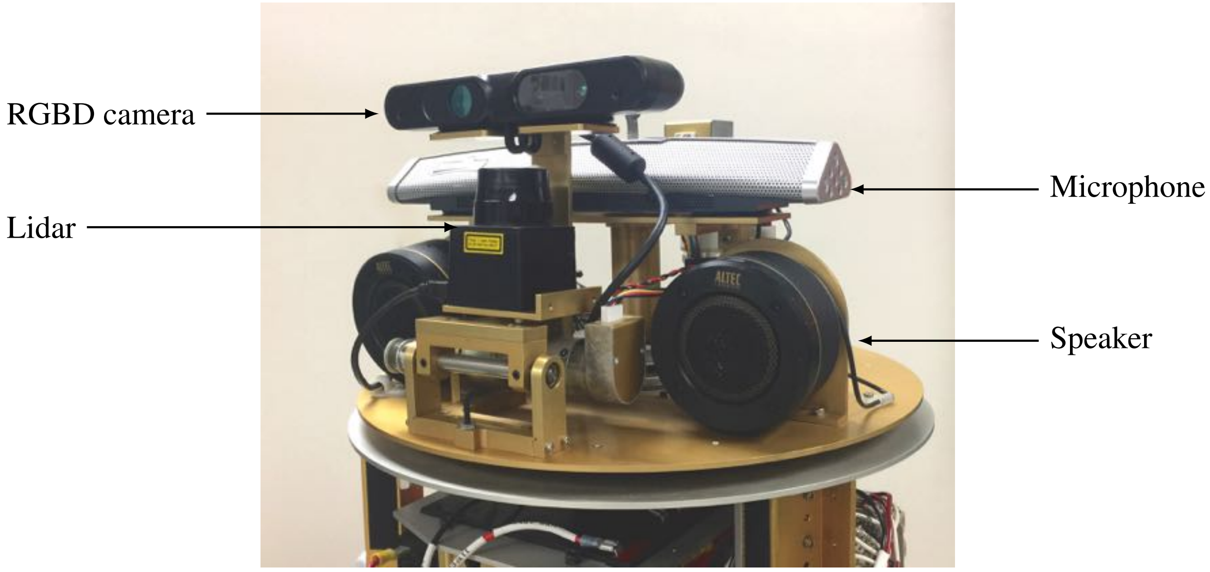


Figure 2.2: Pan and tilt turret with RGBD camera and 2D lidar

Computing

The ballbot consists out of two on-board computers. One is running a real-time QNX operating system, the other one is running Linux Ubuntu. The PCs are connected via Ethernet and can be reached over WiFi from a third off-board computer. The QNX runs the low-level balancing controller as well as the arm and turret control. The Ubuntu PC reads both lidars, one on the turret, one on ballbot's front-side, the camera and controls the speakers. The third, off-board, PC is used to send commands, via a graphical interface, and visualize the received data from the ballbot using Rviz and Rqt.

Inverse mouse-ball drive

The inverse mouse-ball drive lets the ballbot drive and balance. It features 4 motors which drive rollers using belts. These drive rollers introduce the force to the ball in one direction but let the ball rotate freely in the other direction. As it can be seen in Figure 2.3 the first deck above the inner mouse ball drive contains the yaw drive and the motor amplifier.

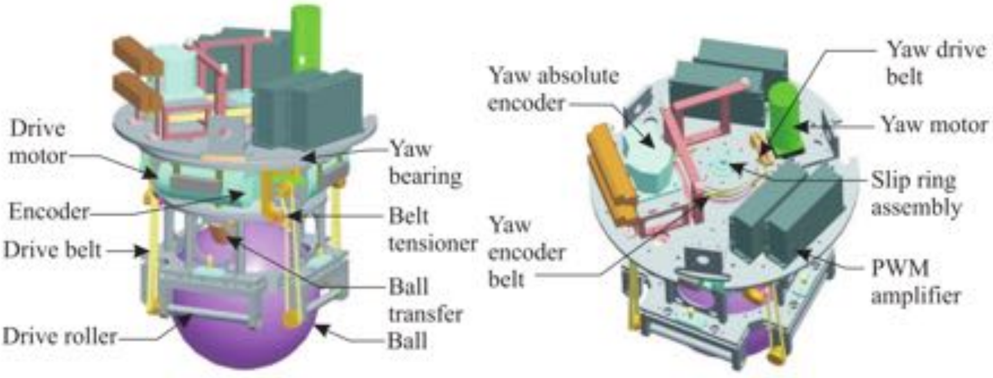
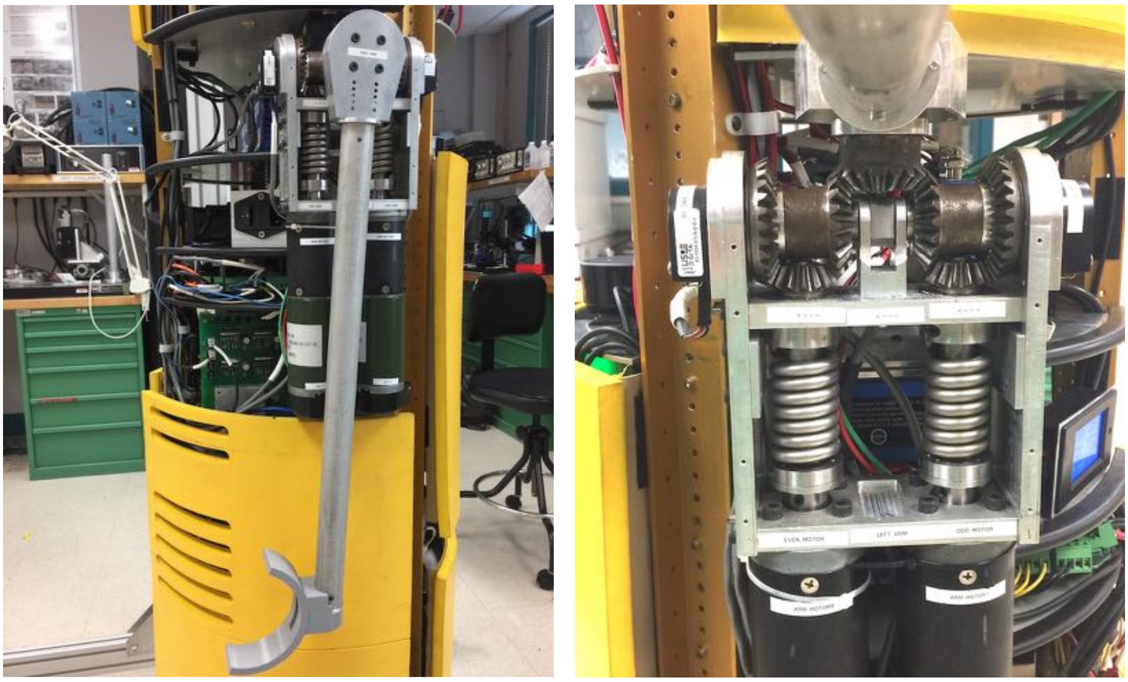


Figure 2.3: CAD model of the four-motor inverse-mouse ball drive with included yaw drive. (Courtesy: Ralph Hollis; Appeared in [1])

Arms

In 2011 the ballbot's abilities got extended by the addition of 2 DOF arms, as presented in Figure 2.4a. This is the first and so far only ballbot with arms [3, 4, 21]. They were designed by Ralph Hollis and Byungjun Kim and consist out of Series Elastic Actuators, are used in robotics since the mid-1990s. According to Pratt [31] the benefits of those are more accurate and stable force control, a lower reflected inertia and a higher shock tolerance. A 1999 published paper by Robinson used successfully Series Elastic Actuators as legs for a Biomimetic Walking Robot [32]. A very recent paper from Roozing [33] provides a deeper understanding of the selection of the stiffness. This may help for future modifications to improve the capabilities of the current arms.



(a) Ballbot's 2-DOF arm with series-elastic actuators.

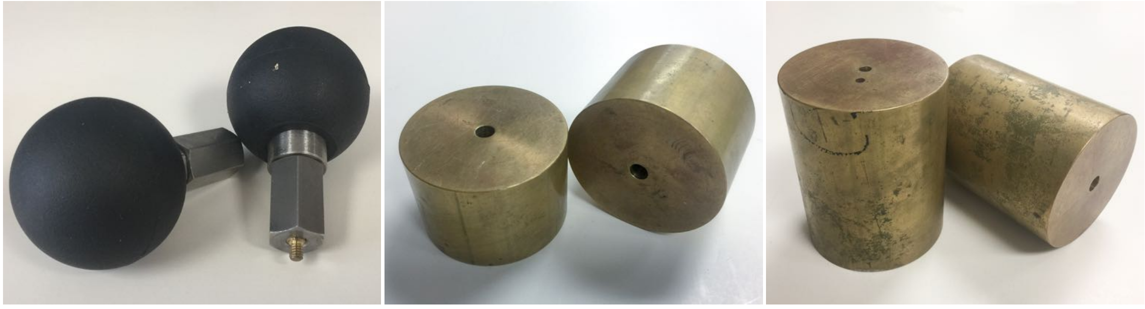
(b) Close up, of the series-elastic actuators and bevel gears.

Figure 2.4: Ballbot's 2-DOF arms.

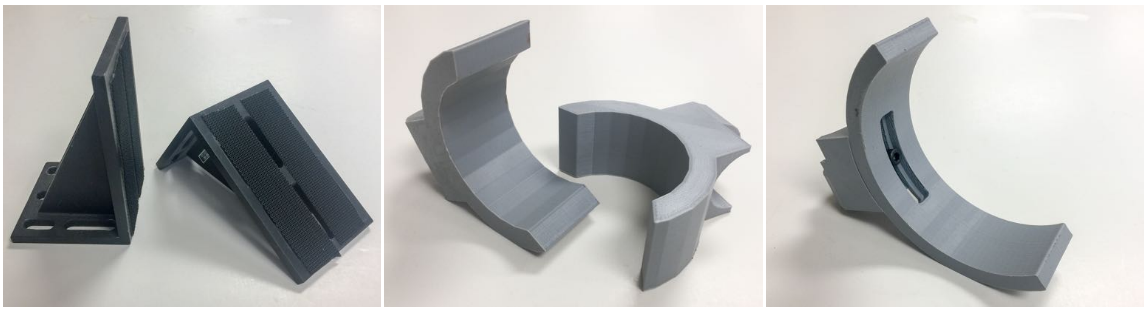
The arms are mounted in a height of 1.3 m symmetrically around the y-axis of the ballbot. Figure 2.4b shows a close- up of the arm assembly. Each arm consists of two electric motors with a torque of 0.12 Nm at 3000 rpm, motor amplifiers running at 48 V with a maximum operating current of 10 A, two springs, four encoders and 2:1 beveled gears. To supply more torque a planetary gear train with a factor of 127:1 was added to the motors. The helical spring was custom designed with a torsion coefficient of 16.37 Nm/rad. The torque of each motor can be calculated by the springs coefficient and the optical encoder with 1024 steps on each side of the spring. The arms can be easily mounted and removed by disconnecting the power and data connectors and removing four screws. The beveled gear design presented in Figure 2.4b allows the arms to move in 2 DOFs, but with the current 0.56 m long arms it is not possible to bring them together in front of the ballbot. To compensate this issue an angled arm could be introduced.

Hands

A variety of different hands can be attached to the arms for different tasks. The attachments used in this thesis are presented in Figure 2.5. Figure 2.5a shows a knob which is used for robot-human interaction. It provides a good grip due to a soft rubber surface. A haptic sensor can be added to detect if a human is holding the hand. 2.5b and Figure 2.5c are showing dummy weights with 1kg respectively 2kg. These weights are used to calibrate the torque values provided by the springs and encoders. Furthermore, these dummy weights were used to test the correct lean angle compensation, while moving the arms. The hands displayed in Figure 2.5d-2.5f are used in different grabbing and lifting scenarios. While the hands, shown in Figure 2.5d, are used to grab an object by pressing the hands against each other, the hands in Figure 2.5e and Figure 2.5f are used to lift boxes with provided handles. Figure 2.5d shows a hand with a Velcro$^{\text{TM}}$ surface. This surface can either be used to lift a box with the opposite Velcro$^{\text{TM}}$ or to attach a rubbery surface to it, to increase the friction. Due to the physical properties of the previously described 2 DOF arms, there is only a very limited space in which the arms can press against each other in front of the ballbot. The hands relay on the properties of the box, especially the correct size and surface is needed.



(a) Knobs for human interaction (b) 1kg dummy weight (c) 2kg dummy weight



(d) Hands with Velcro$^{\text{TM}}$ (e) Lifting hands version 1 (f) Lifting hand version 2

Figure 2.5: Different attachable hands for ballbot's 2 DOF arms.

The hand, shown in Figure 2.5f, is a newer version of the hands displayed in Figure 2.5e. Tests showed, that a bigger radius for the u-shape, helps to increase the probability to pick up the object. Additionally, it is possible to adjust the angle of the u-shape in this version by up to 45°, depending on the task.

2.3 Software and Controls Overview

Ballbot's software is running on two different onboard systems which are connected via Ethernet. As it is presented in Figure 2.6, these two onboard computers are connected to an onboard router which allows additional computers to connect wireless via Wi-Fi to the ballbot.

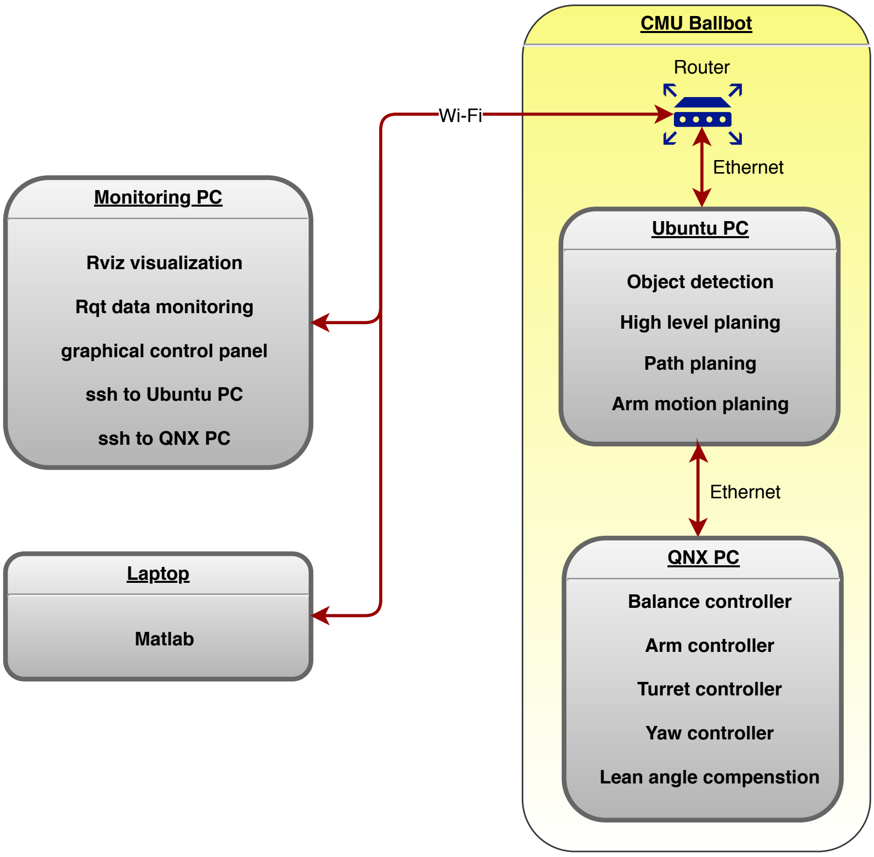


Figure 2.6: Software and PC overview

The Ubuntu PC is configured as the ROS [34] master. It computes the high-level planning such as the object detection, the path planning, and the arm trajectories. The second computer is running a real-time QNX operating system. It is used to run the balancing routine, as well as the arm, yaw, leg and turret controllers. This QNX system connects to the motor amplifier via an extended IO card which contains the digital to analog converters. Furthermore, this system will calculate the object mass and the needed lean angle to compensate the external forces.

The external monitoring computer is used to command and monitor the ballbot via the ROS network. It contains a graphical user interface to toggle the basic functions of the ballbot and connects via ssh to the onboard computers. The Rviz and RQT software are used to visualize, send, and receive data from the ballbot. Additional devices such as a laptop can be added via Wi-Fi to send and receive ROS messages using Matlab^{TM}.

2.4 Arm controller

The arm controller is a PID controller running on the QNX computer. It receives the desired arm angles via ROS messages. Those ROS messages are either sent by the high-level control script running on the Ubuntu computer or from Matlab$^{\text{TM}}$ running on a Laptop connected via Wi-Fi.

There are three main arm modes implemented on the ballbot. The modes are a *stationary mode*, a *compliant mode*, and a *position mode*. In stationary mode, the arm amplifiers are commanded to supply no current. Therefore the arms do not move due to the friction of the system. The compliant mode enables the operator to move ballbot's arms freely. During this mode, the arm controller computes the torques needed to compensate gravity. In position control mode the PID controller tracks the needed torques to reach the desired position or follow the desired trajectories.

As both controllers compute the desired torques, only one routine is needed to calculate the corresponding currents for the arm amplifiers. The amplifier receives a 0-10 V signal from the digital to analog converter of the QNX system, where 0 V corresponds to a current of -5 A and 10 V correspond to a current of 5 A. In stationary mode, 5 V is sent to the amplifiers to disable the current output.

2.5 Balance controller

This chapter will present the key function of the ballbot. The three different balance modes and their controllers will be presented. The last section will take a closer look at the differential flatness path planning, which is used to navigate. The balance controller and its subsystems are running on the real-time QNX computer and send their commands via digital to analog converter to the motor amplifiers.

Figure 2.7 presents an overview of how the individual components of the balance controller interact with each other. Depending on the selected mode, some components may not be active. The following sections will present, which components are used in which mode and why this setup is needed.

2.5.1 Balancing

The balancing mode, presented by [17], is the most basic mode of the ballbot. Its sole purpose is to keep ballbot's lean angle $\phi = 0°$ (see Figure 2.8). To achieve this, the PID balancing controller, in the blue box of Figure 2.7, receives only the real body angles, presented in green, and the manual compensation, presented in the yellow box. Neither the station keep compensation nor the arm compensation is active. The manual compensation,

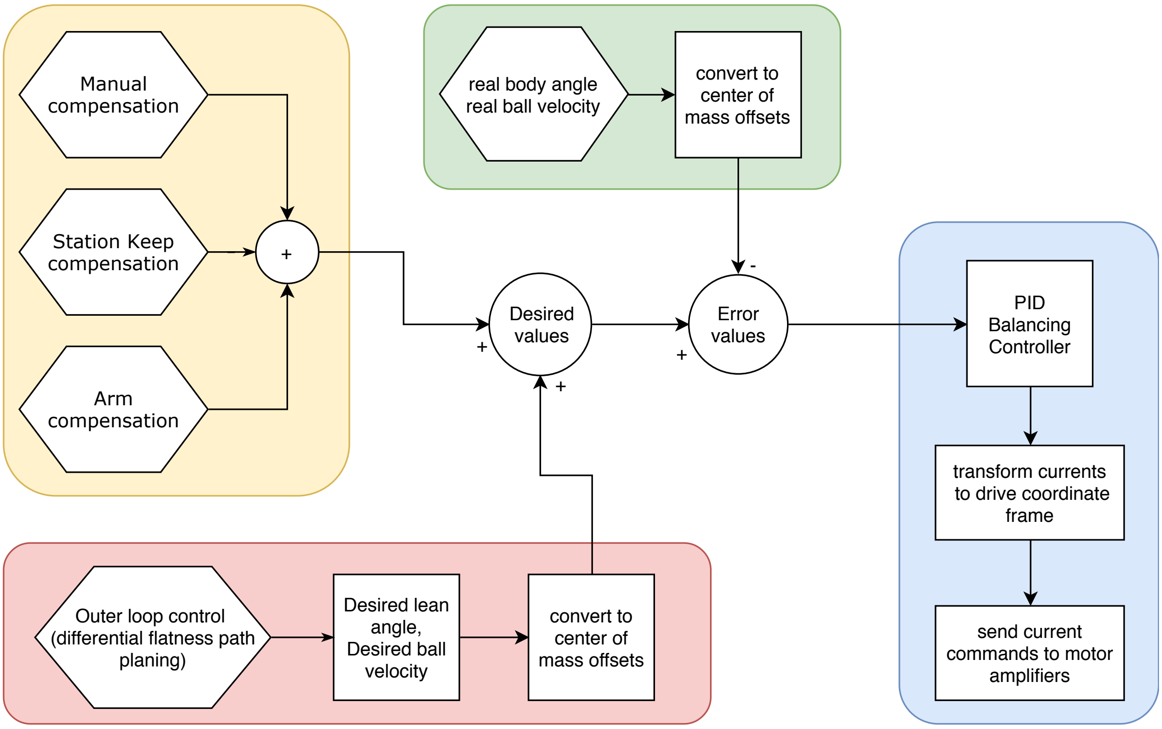

Manual compensation
Station Keep compensation
Arm compensation
+
real body angle real ball velocity
convert to center of mass offsets
-
Desired values
+
+
Error values
+
PID Balancing Controller
transform currents to drive coordinate frame
send current commands to motor amplifiers
Outer loop control (differential flatness path planing)
Desired lean angle, Desired ball velocity
convert to center of mass offsets

Figure 2.7: Balance controller overview

lets the operator adjust the desired lean angle if an in-balance of the system is known. As there is no other compensation, ballbot's position is expected to drift. To avoid this, a station keep compensation was implemented and will be presented in the following section.

2.5.2 Station keeping

The station keep mode, implemented by [17], tries to keep the ball angle $\theta = 0^\circ$ (see Figure 2.8). As the ball is the only contact point between the ballbot and the ground, this ball angle translates to ballbot's position. Additionally, to the manual compensation, which was presented in the previous section, the station keep compensation and the arm compensation are active. The station keep compensation will adjust the desired lean angles to reduce the ball angle, as this compensation is not enough if heavy objects are attached to the arms, the work presented in this thesis implemented the arm compensation. This compensation adds an additional lean angle to counteract the force introduced by lifting and transporting objects. It will be presented in greater detail in Chapter 4.

2.5.3 Outer Loop Control

The outer loop control adds another component to the balance control architecture. Figure 2.7 shows, that the red outer loop control block is separated from the other desired lean angles. The outer loop control receives the desired lean angles and ball velocities from the differential flatness path planner, presented in the following section. It is used to perform

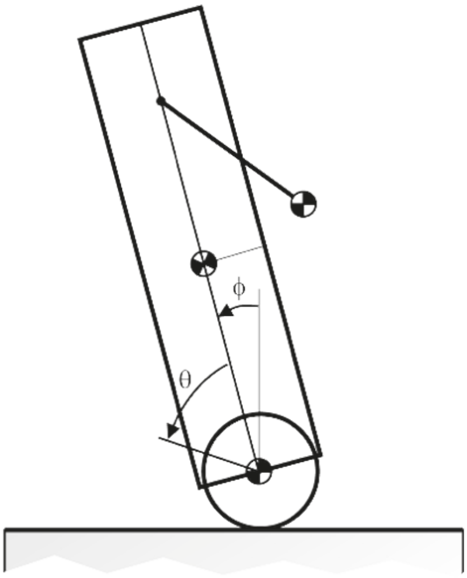


Figure 2.8: Simplified Ballbot model (Courtesy: Ralph Hollis; Appeared in [2])

point-to-point movements. The separation is due to the fact that the planer is designed to work with lean angles between $-5^\circ < \phi < 5^\circ$. Therefore the other compensation angle is added to the desired values of the outer loop control in a separate step.

2.5.4 Differential Flatness Path Planning

The differential flatness path planning presented and implemented by Michael Shomin [35] is the path planning algorithm used for all navigation tasks. Prior to this path planner was implemented, the ballbot was navigating with a set of precalculated motions [19, 20, 36, 37]. While this worked well for basic navigation tasks, the differential flatness path planning enabled a wider spectrum of navigation including on-board planed abort or recover trajectories.

The presented differential flatness path planning was developed to achieve a fast onboard computation for feasible trajectories for the underactuated ballbot. This was achieved by linearizing the system to find flat outputs. This was only possible due to the fact that ballbot's lean angle ϕ can be assumed to be bound $-5^\circ < \phi < 5^\circ$. While ballbot's acceleration is dependent on the lean angle, the position and therefore the trajectory depends on the ball angle θ. To enable a point-to-point motion with an initial and final condition a ninth order polynomial is needed. This polynomial enables the path planner to generate feasible trajectories for θ and ϕ [35].

Additionally, this work enabled ballbot to recover from initial lean angles as well as initial velocities. In an experimental test, where an initial lean angle of 2° was introduced, the ballbot managed to recover six out of six times successfully.

2.6 System Identification

To achieve the desired graceful movement of the ballbot a good understanding of its properties such as the inertia, the center of gravity and its weight was needed. As described in [17] there is a procedure to measure all needed parameters. Those measured values do not match with the current ballbot anymore as over the last five years the hardware of the ballbot was strongly modified. The overall weight of the ballbot increased due to the added turret, a new IMU unit, and the 2 DOF arms. Therefore the current the ballbot was measured again.

As shown in appendix A a stiff torsion wire was mounted perpendicular to the ballbot, similar to the approach showed by [17]. By moving the torsion wire up and down on the body of the ballbot the center of gravity could be determined. With the torsion wire mounted exactly at the center of gravity a force was introduced to on end of the ballbot. The angular velocity trajectory was recorded with the internal IMU. The same experiment was redone with a steel beam with parameters shown in Table 2.1. To calculate its inertia, it was split into three rectangles rotating around a perpendicular axis as presented in Equation 2.1 and 2.2. The result of $I_{I-beam} = 5.962\ \mathrm{kgm}^2$ was as well verified by a 3D model created in Creo Parametric.

Table 2.1: I-Beam Parameters

Parameter	Symbol	Value
Length	l	1.746 m
Width	w	10 cm
Height	h	14.3 m
Mass	m	23.4 kg
Steel depth	d	0.5 cm

$$I_{rectangle} = \frac{1}{12} \cdot m \cdot (w^2 + l^2) \tag{2.1}$$

$$I_{I-beam} = I_{rectangle_1} + I_{rectangle_1} + I_{rectangle_1} = 5.962\ \mathrm{kgm}^2 \tag{2.2}$$

The frequencies presented in Equation 2.3 are the angular velocity frequencies of the I-beam and the ballbot. A plot of those recorded frequencies can be seen in appendix A.

$$\begin{aligned} \omega_{I-beam} &= 0.36\ \mathrm{Hz} \\ \omega_{Ballbot} &= 0.23\ \mathrm{Hz} \end{aligned} \tag{2.3}$$

As the inertia of the steel beam is known the inertia of the ballbot can be calculated using the Equation 2.4. A similar inertia measurement unit was used to measure the angular velocity trajectory of the steel beam and the ballbot.

$$I_{Ballbot} = I_{I-beam}\frac{\omega^2_{I-beam}}{\omega^2_{Ballbot}} = 14.6073\ \mathrm{kgm}^2 \tag{2.4}$$

The setup shown in appendix A provided the pitch and roll inertia I^b_{xx} and I^b_{yy}. To obtain the yaw inertia I^b_{zz}, the ballbot was mounted upright in the fixture. Again a force was introduced and the results were compared to the measured and calculated results from the steel beam. The results of this measurement, as well as all other parameters needed for the dynamic model, are shown in Table 2.2. Additionally, this table shows that the measured values are different from the old values measured by Nagarajan in 2009 [17], due to many modifications of ballbot's setup.

Table 2.2: Ballbot Parameters

Parameter	Symbol	Value	Value from [17]
Ball radius	r	0.1058 m	0.106 m
Ball mass	m_{ball}	2.44 kg	2.44 kg
Ball moment of inertia	I_{ball}	0.0174 kgm^2	0.0174 kgm^2
Body mass	m_b	64.86 kg	51.66kg
Body COM along z-axis from ball center	l_b	0.71 m	0.69 m
Body roll moment of inertia about COM	I_{xx}	14.6073 kgm^2	12.59 kgm^2
Body pitch moment of inertia about COM	I_{yy}	14.6073 kgm^2	12.48 kgm^2
Body yaw moment of inertia about COM	I_{zz}	0.66 kgm^2	0.66 kgm^2
Arm mass (each)	m_a	7.03 kg	-
Arm length	l_a	0.56 m	-

The overall weight of the ballbot increased from 51.66 kg to 81.65 kg with the 2 DOF arms.

2.7 Dynamic model of the ballbot

The ballbot simulation and the differential flatness path planner need a dynamic model of the robot. Nagarajan presented the following model in his master thesis [3]. The model assumes, that there is no yaw motion on the body, therefore only a planar version needs to be calculated. Additionally, the model does not include any slip between the ball and the drive rollers or the ball and the floor.

The model uses four configuration variables which each have an X and Y component. The 3D model is represented by $q = [\theta, \alpha^l, \alpha^l, \phi]$, where θ is the ball angle, α^l and α^r are the arm angles for each arm and ϕ is the body lean angle. While θ, α^l, and α^r are actuated, ϕ is dependent on the other variables. Figure 2.9 presents the exact locations of those angles.

Using the inertia matrix $M(q)$, the Coriolis and centrifugal matrix $C(q, \dot{q})$ and the gravity vector $G(q)$ presented in Equation 2.5-2.7, the Euler-Lagrange equations can be written as

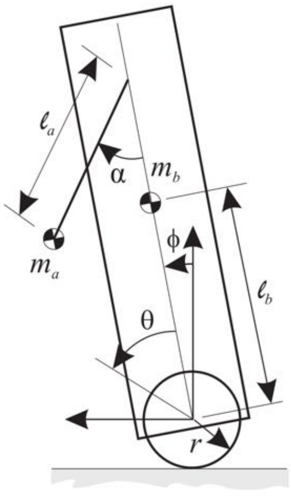


Figure 2.9: Ballbot's configuration shown in a two dimensional model of ballbot with arms (Courtesy: Ralph Hollis; Appeared in [3, 4])

a matrix presented in Equation 2.8. In the result of this equation, $\tau = [\tau_\theta, \tau_{\alpha^l}, \tau_{\alpha^r}]$ is the vector of the combined forces.

$$M(q) = \begin{bmatrix} M_{\theta\theta} & M_{\theta\alpha^l}(q_s) & M_{\theta\alpha^r}(q_s) & M_{\theta\phi}(q_s) \\ M_{\alpha^l\theta}(q_s) & M_{\alpha^l\alpha^l}(q_s) & M_{\alpha^l\alpha^r}(q_s) & M_{\alpha^l\phi}(q_s) \\ M_{\alpha^r\theta}(q_s) & M_{\alpha^r\alpha^l}(q_s) & M_{\alpha^r\alpha^r}(q_s) & M_{\alpha^r\phi}(q_s) \\ M_{\phi\theta}(q_s) & M_{\phi\alpha^l}(q_s) & M_{\phi\alpha^r}(q_s) & M_{\phi\phi}(q_s) \end{bmatrix} \tag{2.5}$$

$$C(q,\dot{q}) = \begin{bmatrix} 0 & C_{\theta\alpha^l}(q_s,\dot{q_s}) & C_{\theta\alpha^r}(q_s,\dot{q_s}) & C_{\theta\phi}(q_s,\dot{q_s}) \\ 0 & C_{\alpha^l\alpha^l}(q_s,\dot{q_s}) & 0 & C_{\alpha^l\phi}(q_s,\dot{q_s}) \\ 0 & 0 & C_{\alpha^r\alpha^r}(q_s,\dot{q_s}) & C_{\alpha^r\phi}(q_s,\dot{q_s}) \\ 0 & C_{\phi\alpha^l}(q_s,\dot{q_s}) & C_{\phi\alpha^r}(q_s,\dot{q_s}) & C_{\phi\phi}(q_s,\dot{q_s}) \end{bmatrix} \tag{2.6}$$

$$G(q) = \begin{bmatrix} 0 \\ G_{\alpha^l(q_s)} \\ G_{\alpha^r(q_s)} \\ G_{\theta^l(q_s)} \end{bmatrix} \tag{2.7}$$

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \begin{bmatrix} \tau \\ 0 \end{bmatrix} \tag{2.8}$$

The derivation of those matrices and all elements of the sub-matrices are presented in more detail in Appendix A of [3].

3 Related work

Similar to a human, which needs to localizes the object and compensate its weight while transporting, the ballbot needs to conquer those tasks. Therefor, this chapter will give an overview of the related work for object detection and localization. It will present different methods to identify the target object, its location, and its properties. Additionally, the related work for lifting motions is presented. Different approaches and risks for human, humanoid, balancing and stable robots will be discussed. This chapter shall provide a better understanding of object manipulation, so the best fitting approaches can be implemented and tested on the ballbot.

3.1 Object detection

The following section will present three different approaches for object detection and pose estimation. While the first one will rely on, in robotics commonly used, fiducial markers, the second approach shall detect objects without markers using the same sensor types as ballbot. The third section will show how this sensor data can be used to generate grasping trajectories.

3.1.1 Object detection with ARUCO fiducial tracking

Fiducial markers are commonly used in robotic for localization and object detection, but as Garrido-Jurado describes in [38] those existing marker tracking systems cannot keep up with the high resolutions and frame rates published by modern cameras. Therefore, they proposed a new multi-scale strategy to improve the speed of the detection. They experimentally showed that the system is able to process over 1000 frames per second in 4k video streams. This remarkable improvement is achieved by applying the following preprocessing to each image. First, a locally adaptive image threshold gets applied on the original camera stream. Using this filtered image a contour detection is performed. As fiducial markers are rectangular, irrelevant polygons are removed. This leaves the relevant markers, which are transformed to the original square shape to compute the correct pose. Additionally, the markers uses its cells for bit assignment. This bit assignment gives each marker a unique identification number, which can be later used to distinguish between objects. Each marker consists out of a 7 by 7 cell array, with a two cell thick black boarder. The inner 5 by 5 square is used for the identification, where each row symbolizes a five bit word. To improve the robustness, only two of those five cells are used as data bits. The purpose of the remaining three is a failure detection using Hamming Code. This five rows with two binary cells provide 1024 unique marker configurations.

A new version of the ARUCO fiducial detection presented by Romer-Ramirez et al. [39] in 2018 shows, that the speed of the fiducial tracking can be even improved replacing the adaptive thresholding with a global thresholding method. Additionally resizing the areas with markers helps to increase the processing speed, as smaller images can be processed quicker. The ARUCO framework is able to work with a wide variety of image streams assuming the camera is calibrated.

3.1.2 Object detection using a 2D lidar and a camera

Kwak et al. [40] proposed in 2014 an object detection algorithm using the combination of a 2D lidar and an RGB camera. While the camera provides good visual information, it lacks in terms of scale and depth. On the other hand, the 2D can provide this missing geometric information. In 2011 Kwak presented in [41] a procedure for an extrinsic calibration. Till then it was difficult to generate a robust mapping between the pixels provided by the camera and the corresponding features of the lidar data. Kawk proposes to us two planes which are in an angle attached to each other. Both planes have a visible line on its outer sides and a line where they are attached. This enables the calibration algorithm to match the camera, which sees the lines, and the lidar, which can detect the border and the angled attachment of the planes.

With this calibration in place, Kwak et al. were able to introduce an object segmentation with matches camera and lidar feature and then selects the best matching feature pairs to differentiate between objects. Experiments showed, that the combination of those two sensors provides a better segmentation than using only one sensor. Additionally, they achieved to track different moving and stationary objects while moving the sensor array [40].

3.1.3 Markerless visual servoing on unknown objects for humanoid robot platforms

A very recent paper by Fantacci [42] proposes a new framework for markerless object detection and grasping. According to Fantacci, a key factor is, to understand the object shape and pose as well as the exact location and orientation of the end-effector. Similar to the approach of Vezzani [43], where the shape of the object gets modeled and simplified using stereoscopic vision, Fantacci uses stereo vision to detect the volume of the object. The desired grasping position is computed by a formulated non-linear constrained optimization problem. Image-based control commands are then sent to the end-effector to reach the grasping position. In an experimental test, the proposed framework showed significant improvements in Root Mean Square Error in respect to the provided direct kinematics.

3.2 Lifting

This chapter focuses on different approaches and motions for lifting objects with manipulators. While the first part will take a closer look at the issues and risks found in human lifting and why robotic lifting is therefore needed, the following parts will look at humanoid and other dynamically stable robots which are able to transport objects. The last section will present how not balancing mobile manipulators interact with objects.

3.2.1 Human lifting motion

As Hsiang explains in [44] there is a lot of research in human lifting motions and the resulting injuries and problems such as low back pain. According to the work of Hsiang, there are a lot of studies on the risks of repetitive liftings, such as the study of Antwi-Afari [45]. But it is not fully understood how and if proper training and special lifting techniques help to avoid or reduce the risks. The study of Antwi-Afari [45] performed a biomechanical analysis on the risks for musculoskeletal disorders due to repetitive heavy lifting. They concluded, that the main risk factors for musculoskeletal disorders are the lifting duration, weights, and postures.

Additionally, Hsiang explains in [44], that back injuries due to repetitive heavy lifting are a big cost factor for workers compensation costs in companies. The presented work could not find a representative correlation between different lifting techniques and low back pain. Therefore they conclude, that even if the safety officials of companies will provide lifting training, the workers will still continue with carrying heavy objects and therefore suffer from the risks.

As presented back issues due to heavy lifting are a big problem for workers and companies. Therefore it was just a matter of time that devices were built which should assist workers to lift and transport heavy objects. These personal lift-assist devices can range from passive straps which help the workers to keep a correct position to fully active exoskeletons which reduce the needed force applied by the worker. The study of Graham [46] investigated the use of a passive lifting assisting devices for workers in an automotive manufacturing facility. The used passive exoskeleton is mounted at the hip of the worker. Additionally, it attaches to the chest and the front side of the thigh. These three points of contact shall reduce the strain of the lower back muscles. By measuring surface EMG data while the worker performed their usual lifting task with and without the device, Graham could conclude, that the device reduced the thoracic and lumbar erector activity. Eighth out of ten workers confirmed, that they would use the device with small modifications on a daily base.

A similar study presented by Bosch [47] concluded, that passive exoskeleton can reduce the muscle activity by about one third. Additionally, these devices can extend the endurance of the workers by almost three times. The system used in this study reduced the back issues but increased the discomfort in the chest region and is therefore not a good solution.

These presented studies show, that repetitive lifting task are bad for the human well being, even with proper training, a good lifting position and additional assisting devices. Therefore it is necessary to develop fast and agile human-size robots, which can take over those carrying task.

The paper published in 2004 by Matsunaga presents a kinematic analysis of human lifting motion. It compares different setups such as the location and the weight of the object. To determine the exact position of each human joint colored markers were attached at the ankle, knee, hip, elbow, shoulder, and hand. Additionally, the object was tracked by a similar marker using the QuickMagIV motion analysis system. As the ballbot cannot reach the ground with its arms, only the experiments done at desk height are relevant. Depending on the mass, the center of gravity is shifted during the lifting motion backward. The human in this experiment accomplished this center of mass shift by moving his hip joint further back for heavier weights [48]. This concludes that the ballbot needs to shift its center of gravity as well during the lifting motion. As the ballbot can not flex his body similar to a human, the whole body angle needs to be adjusted.

3.2.2 Human interaction and feedback

Sciutti presented in 2014 how the object weight can be detected from human and humanoid lifting movements. The human detects the weight of the object either by observation or by experimenting with the object. If the human was not able to observe an interaction with the object, the mass gets detected by the pressure applied to the fingertips and hands. Therefore a human is able to counterbalance very fast. Already in the early childhood, humans learn to estimate the weight of an object just by observing another human interacting with it. On the other side, it is difficult to estimate the weight of an object, if it gets lifted by a robot, as they generally do not need to lean or flex its muscles. According to Sciutti, it is important to provide this valuable information to enable a good human to robot interaction. The paper proposes that the humanoid robots need to perform human-like arm motions to provide this additional information [49].

As the human-size ballbot needs to lean back during the lifting maneuver, an external observer is able to differentiate if the lifted object is light or heavy. In Chapter 4.1.2 the compensation angle of a human is compared to the compensation angle of the ballbot while performing a similar lifting and transporting task.

3.2.3 Humanoid robots

Over the last decade, a variety of different humanoid robots were presented. This chapter will take a closer look at a selection of them and analyze how they are able to detect and localize objects. Additionally, this chapter will present how those robots lift and transport objects while keeping their balance. It is expected, that some of these approaches are transferable to ballbot's expected behavior. To give a basic understanding of how capable the ballbot is compared to other robots the maximum transported weight of each robot will be outlined. This covers only the published transported weight and the real capabilities might be higher, but it will give a reference point.

Dynamic Lifting Motion of Humanoid Robots using HRP-2

A paper presented 2007 by Arisumi proposes different lifting motions for humanoid robots. The first presented approach uses preliminary motion. By rapidly shifting the center of gravity of the robot a higher kinetic force can be applied to lift certain objects. The second approach tries to keep the center of percussion aligned with the support polygon. They were able to verify their approaches using simulation and the HRP-2 robot. HRP-2 is about 1.5 m tall and has a mass of 58 kg. The repeatable tests showed, that it can lift a 4.5 kg heavy box which is about 0.5 m away from the robot. This paper focuses only on the lifting motion and does not cover any experiments on transporting the box to a different place. Screen-shots of their simulation and experiment show that HRP-2 is leaning slightly forward to increase the momentum while raising the box and leaning back. While HRP-2 is holding the object it needs to compensate the additional forces by leaning backward [5].

Toro (DLR)

Toro, the humanoid robot developed at the german DLR, showed in a publication by Henze in 2016 that it is able to lift a box filled with bottles while keeping its balance on two legs. The weight of the box is 12.2 kg. Similar to HRP-2 described in the previous section Toro did only perform a lifting maneuver without actually transporting the box to a different end location. According to the experimental results, Toro's hip orientation did only shift by 1°. Due to the leg and foot design, Toro is able to counteract the measured additional forces of 62 N at the hands. The front part of the foot increases its pressure and avoids therefor an otherwise needed lean angle to compensate for the mass of the object. The Hands are controlled in the Cartesian world frame and raised during the lifting task by 0.1m [6].

Atlas (Boston Dynamics)

The humanoid robot Atlas, designed by Boston Dynamics, is specialized for mobile manipulation. It is able to walk and even run on different surfaces[7]. In 2016 Boston Dynamics published a video [50] which showed how, the 180 lbs heavy, Atlas can interact with 4.5 kg (10 lbs) heavy boxes. It was able to pick them up from the ground by applying inwards pressure and place them on a shelf, which is located 90° tilted to Atlas. The hands have a flat surface which is just pressed against the box. This introduced friction is enough to carry the box. During the lifting motion, Atlas body shifts back to compensate the additional forces. A second test in the same video, showed Atlas trying to lift a box which gets moved by the operator as soon as Atlas is near. The box is covered with fiducial markers. As the box gets moved way, Atlas adjusts its position and tries to lift the box again.

It is assumed, that those fiducial markers are used to detect the location of the box using onboard vision system of Atlas.

3.2.4 Two wheeled dynamically stable robots

Humanoid robots have a big advantage while performing stationary lifting tasks. They can use their feet to counteract the introduced torques during a lifting motion. Therefore the following sections will focus on robots which need to actively keep their balance. As there are currently no robots which balance on a ball and lift objects, the following robots are all balancing on two wheels.

Segway Robotic Mobility Platform

The Segway Robotic Mobility Platform presented by Nguyen in 2014 is a dynamically stable drive system. It consists out of two individually controllable wheels. By leaning forward the wheels try to counteract this lean angle and the system drives forward. The platform allows different payload or manipulators to be mounted on top of it. It can rotate on its point and carry a payload of up to 45 kg while keeping its balance. The National Aeronautics and Space Administration (NASA) mounted their humanoid torso of Robonaut on this platform to test the coordinated control of dexterous limbs. Using the Segway Robotic Mobility Platform, Robonaut was able to use tools and open doors while keeping its balance. The Massachusetts Institute of Technology (MIT) used its 5-DOF arm on this platform. Similar to ballbot's arms MIT's arms use Series Elastic Actuators to achieve the desired compliance [51]. The robot is able to successfully open a door with its arm while keeping its balance. Additionally, two spring-loaded kickstands were added to prevent the robot from damage whenever a fault is detected [52].

uBot-4

The UMass uBot-4, developed at the University of Massachusetts Amherst, is already the fourth generation in the uBot series. The uBot-4 balances similar to the Segway Robotic Mobility Platform, described in the previous section, on two wheels. To interact with its environment it has a rotary torso and a pair of 4-DOF arms. uBot-4 is able to lift up to 1.85 kg with its arms fully extended, while the robot weighs just 11.5 kg. This is a lifting capability of 16% of its own weight. Due to its small size of 0.542 m and long arms of 0.502 m, it is able to pick up and transport objects from the floor [53].

Golem Krang

Golem Krang is a dynamically stable robot which is balancing similar to the previous section on two wheels. It has a torso with two arms mounted to the drive base. In [8] Stilman describes how these additional actuators can be used to improve the maneuverability of the two-wheeled platform. It is shown that the torso can extend and collapse itself while keeping the robot balanced. In [54] Stilman presents, that it can carry 40 kg mounted to the top of the torso and transition from a static sitting to a dynamic balancing autonomously. During this transition, the lean angle was constantly adjusted to compensate for the changing center of mass. As it can be seen in the video published by Georgia Tech [9] the robot is able to withstand external forces such as pushes, pulls and even kicks by a human operator. Furthermore, it presented its ability to use its arms for carrying and lifting tasks while driving and keeping its balance. The published video shows, that Golem Krang lifts and transports a simple chair. The lifting motion is performed by fully extending the arm while driving close to the chair. As soon as the arm reaches through one of the chairs holes, the robot's torso leans back and raises therefor the arm. Even though the transported chair swings freely and introduces additional disturbances, no effects on the balancing performance can be detected [9].

Handle (Boston Dynamics)

Handle is currently the newest robot from Boston Dynamics. It balances on two legs which have wheels instead of feet. Therefor Handle does not walk, it drives. Additionally, it has two arms to interact with its surroundings. In the video [10], published by Boston Dynamics in 2017, Handle showed that it can lift a 100 lbs heavy object while balancing. Furthermore, it was able to transport the same object. To lift the object special hands with small claws were used. These claws hook into the box while it gets lifted. Handle has a big body above its arms. This body can move to counteract the forces introduced by the lifting motion. The video description states, that they are using similar dynamics, balance and mobile manipulation as in their other robots, such as the previously presented humanoid Atlas. Handle does not use any fiducial markers to detect the object, so the box has either a preset location or Handle uses other sensors to detect it.

3.2.5 Mobile manipulators

This last lifting section will present different lifting and grasping techniques. Even though the presented robots are not balancing while performing the lifting task, they can provide valuable information on how to position correctly in front of the object and how to plan the arm trajectories before, during and after the lifting tasks.

Centauro

The Centauro robot is a project of the European Union's Horizon 2020 Programme. It is designed to perform tasks, so far, only humans could perform. Especially it shall be used in disaster scenarios which are too dangerous for human workers. The 93 kg heavy robot uses 4 legs as a base and an anthropomorphic upper body. Due to the use of aluminum, magnesium and titanium alloys the researchers at IIT-Istituto Italiano di Tecnologia achieved to reduce the weight of the arm to 10.5 kg while keeping its single arm payload capacity at 11 kg. To detect its location and objects RGBD cameras and a spherical lidar are used. Additionally, the arms provide a high fidelity torque feedback and a thermal state monitoring [55].

In a recently published video [56], by the IIT-Istituto Italiano di Tecnologia, Centauro is shown in different disaster scenarios. The robot presented, that it can lift and transport a 6kg heavy concrete block and remove obstacles such as a wooden stick from its path. To lift the concrete block the robot lowers its body by bending its four legs. Centauro proceeds with extending its arms to each side of the object. The arms secure the object by applying forces in opposite directions. The hands have a flat soft surface to increase the friction. After releasing the object at a different location, the arms are moved back to Centauros torso. During this trajectory, one hand touches the object and causes it to tip over.

Therefore it is important to track the object even after a successful transport and plan the arm trajectories to avoid further undesired interactions.

Fetch and Freight

The two robots Fetch and Freight presented by Wise et al. in [11] are designed to be robust, low-cost mobile manipulators. They shall be used for research and commercial applications. While the Freight robot is a differential drive mobile base frame, which can carry an object which is placed on its top, Fetch has additionally a variety of sensors and a manipulation arm with 7 degrees of freedom. With its arm, it is able to handle up to 6 kg. Similar to the ballbot it features a pan and tilt head which contains a short-range rgbd camera. Fetch features a ROS package called simple grasping which includes a tabletop perception system and a path planner for its mobile arm. The perception system is able to distinguish between graspable objects and surfaces where it can place objects. Compared to other approaches, the simple grasping does not try to model the objects. It learns with a trial and error approach in simulation and real life how to grasp different objects.

4 Approaches

The following chapter will present how a human will handle the task which was given to Ballbot. By using the OpenPose skeleton tracking [57], the movements of a human can be compared with the arm and lean angles of ballbot. The learned movements of a human lifting and transporting approach will then be used to compile a high-level approach with respect to ballbot's abilities and constraints. This created step by step plan will then be further analyzed step by step. Different approaches and implementation for each step will be presented. Additionally, this chapter will present how the correct lean angle while lifting and transporting an object is calculated.

4.1 Human behaviour

To verify the presented ideas in Chapter 3 a human subject will be given the task of detecting, localizing, lifting and transporting an object. The human behaviour on detection and locating the object shall provide a better understanding of which approach the ballbot should use to find the object fast and reliable. During lifting and transporting the human will be recorded with a camera. The body lean angle and the arm movement will be tracked with the OpenPose skeleton tracking [57]. It shall be investigated how humans position in front of the box to navigate from one place to another. Additionally, the arm movements while setting the box down will be recorded. As the ballbot has a human-like size and weight, the lean angle and arm angles are expected to be compare able. To ensure a similar behavior, the used object for the human tests will be the same as for the following experiments with the ballbot. During the lifting and transporting the object will contain a 10 kg heavy dummy weight.

4.1.1 Object detection

To analyze how humans try to find a certain object in a room following test was conducted. A person was asked to close their eyes while several objects were placed around the human at different heights and distances. The person was then asked to open his eyes and find a certain object, without leaving its position. It was detected that the person looks straight ahead with a slightly lowered line of sight. Afterwards, both sides of the room were checked. The last step was to look directly at the feet of the person. As the ballbot is able to pan and tilt its camera like a human can move its head, a similar approach will be implemented. Ballbot will start looking for objects directly in front of him and the camera angle will be tilted downwards by 10°, as no objects are expected to be located higher than the ballbot. The camera will be moved to both sides of the room prior to the area near to the ballbot will be investigated by lowering the tilt.

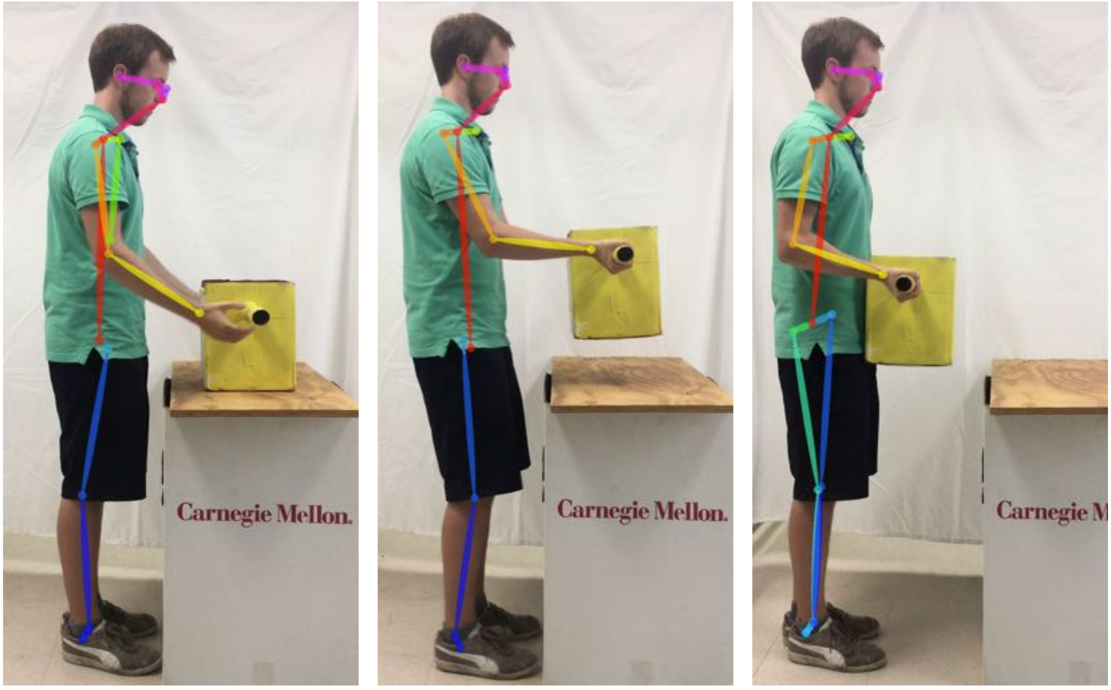
4.1.2 Lifting and transporting

Ballbot shall be used in a human environment and work together with humans. To achieve a predictable motion, the ballbot needs to perform the lifting task as similar as possible compared to a human. Therefore a test person was asked to lift the same box which was used for most of the experiments with the ballbot. During the lifting motion, the person was recorded. Using the skeleton detection of OpenPose [57] the joint angles and positions could be extracted from the video. Figure 4.1a shows the human position with the augmented reality skeleton overlay prior to lifting the box. It can be seen, that the human is located as close to the box as possible and extended his arms below the handles of the box. Figure 4.1b provides an understanding of the lean angle during the initial phase of the lifting. The weight of the box during the entire experiment was 10 kg. A lean angle of 3.4° was measured between the ankle and the shoulder joint. The lean angle of the ballbot performing the same task was measured at 4.4°. This difference is due to the different weight, height, and arm lengths. Additionally, it can be seen, that the human test person did bend his arms, which is currently not possible on the ballbot. Figure 4.1c shows, that the box is brought close to the body as soon as the table is cleared by stepping back. This reduces the needed compensation angle. Ballbot shall perform a similar motion after driving away from the table to ensure the box will not interfere with it while lowering the arms. Bringing the box close to the body does not just reduce the needed lean angle, it reduces as well the inertia while yawing.

As it was discussed in Chapter 3.2.2, it is important for a human to robot interaction to understand the weight of the carried object. This ensures, that the human is prepared for the weight if he receives the object from the ballbot. Due to the human size and weight of the ballbot, it is expected that humans are able to estimate the carried load. Further research is needed to confirm this hypothesis.

4.2 High level approach

Similar to the human lifting approach presented in the previous section, the ballbot needs to detect the object first. Ballbot has a variety of different sensor which will help detect the box. There are two lidars and one RGBD camera currently installed on the robot. The camera can be used to detect predefined shapes or colors. Due to the fact, that the installed lidars are only 2D lidars, a object detection is more complicated. The main issue is to distinguish between the object which shall be carried and other objects in the room with similar shapes. The paper of Kwak [40] publish in 2014 describes a technique where camera and lidar readings are combined to distinguish between objects. A different approach is the use of a fiducial marker. Fiducial markers are squares shaped patterns



(a) Hands are positioned below box prior lifting

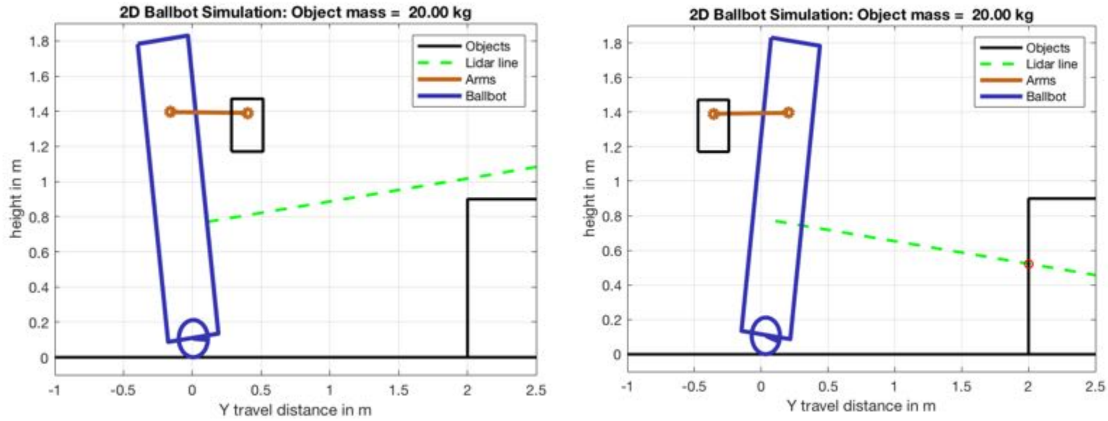
(b) High lean angle during initial lifting

(c) Reduced lean angle, Object near body

Figure 4.1: Human lifting approach.

printed on a flat surface with a very high contrast. They are commonly used in robotics for navigation, localization, and detection. A deeper look into the benefits and drawback between those possible object detection solutions will be performed in the following chapter.

Assuming a successful object detection, the robot needs to plan a path to align its arms with the object, so that a lifting motion can be started. Ballbot is using its lower lidar, which is located 0.77 m above ground for its navigation. To ensure that this lidar will not be blocked by the object or the table it rests on, the ballbot is supposed to lift from its backside. As displayed in figure 4.2 this approach provides another benefit. While lifting and transporting the ballbot will constantly have a lean angle to compensate the additional mass which caused a shift in the center of mass. By lifting the box on the opposite side of the lidar, the 2D beam will be tilted down. Figure 4.2 shows the simulation of the ballbot lifting an object with 20 kg. In Figure 4.2a the box is lifted in front of the ballbot. The dashed green line shows the beam of the lidar. A desk with a typical height of 0.9 m would not be detected if it is further away than two meters. This issue can be avoided by turning the ballbot 180° and lifting on its back side as seen in Figure 4.2b. The lidar will still be tilted, but it will take more than 10 m to detect the ground. This range is enough to provide a solid localization in the testing environment while transporting.



(a) In front of its body: Lidar can not detect a 0.9 m high object more than 2 m in front of him.

(b) On its back side: Lidar is able to detect objects up to 10 m prior it hits the ground.

Figure 4.2: Comparison of the ballbot simulation lifting a object with 20 kg

Prior lifting an object, the ballbot needs to navigate near the object and adjust its position accordingly. This is done in six steps as it can be seen in figure 4.3. During step 1 the ballbot performs a so-called balance and legs up maneuver. This maneuver lets the ballbot transition from its resting position with all legs down to a balancing state [58]. This transition movement, originally designed by Ak Mampetta in 2006, was adapted to ensure the ballbot won't start drifting in one direction while raising its legs. As soon as the legs leave the ground the ballbot was using balance mode, but with the modified version it will transition 0.5 seconds after the legs left the ground to station keep mode, which means the ballbot detects the center of mass offsets and adjusts its lean angle so the body stays stationary. The delay of 0.5 s ensures, that the legs are high enough and won't touch the floor even if the station keep mode needs to adjust the lean angle. In step 2 the ballbot looks for the object and calculates the relative position to its body. In step 3 this calculated position is used by the differential flatness path planner to drive 0.3m in front of the box. The 0.3 m offset is calculated in the frame of the box, so ballbot's end position is directly in front of the box. Step 4 adjusts the yaw angle according to the angle of the box. Additionally, a 180° yaw is performed to ensure ballbot's back is aligned with the box. In step 5 the arms will be raised according to the position of the handles of the box. This ensures, that the arms won't hit the table the box is sitting on. The last step prior to the lifting movement is driving closer to the object. During this step, the ballbot drives backwards with its hands partially raised. The correct distance is again calculated by the current box position. Using the current box position ensures, that even if the box was moved during the approach the ballbot will reach an end position where a successfully lifting can be performed.

Depending on the object which needs to be transported, the ballbot has a variety of different hands as shown in Chapter 2.2. Furthermore, there are as well different lifting motions. The

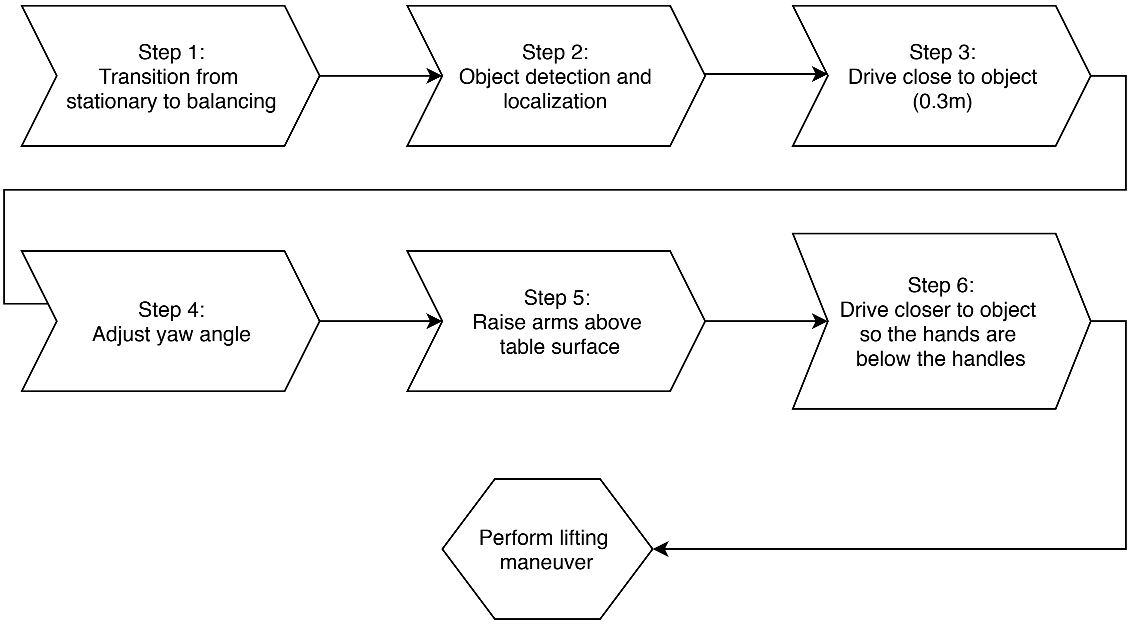


Figure 4.3: Step by step approach

two main arm motions for picking up an object used in this thesis are grabbing the object by applying inward pressure on to the object and lifting the object with its handles. Due to the setup of the ballbot's 2 DOF arms, the objects need to have a certain shape to be picked up. Chapter 4.4.2 will provide more insight into the benefits and drawback of those approaches. A successful lifting can either be detected by the additional torque needed by the motors, or with the tracking of the fiducial markers. Using the fiducial markers is preferred, as it works even with very light objects which are otherwise hard to detect. If the fiducial markers move in the same direction as the hands, a successful lifting is assumed. If the lifting failed, the pose of the object corresponding to the ballbot's position needs to be reevaluated and again adjusted.

During the lifting motion, it is essential to calculate or estimate the center of mass offset permanently. Without knowing the center of mass offset the ballbot cannot keep its position, which will result in an undesired drift of the position while lifting. Chapter 4.5 will show different approaches to how this mission-critical offset can be detected. Chapter 4.6 will then present how this offset will be added to the path planning commands.

After a successfully lift the ballbot will drive away from the pickup location to have enough space to lower the object until it touches the body of the ballbot. This approach is similar to the human behavior and ensures, that the center of mass offset while transporting is as small as possible. Furthermore, a third contact point between the ballbot and the object avoids, that the object acts like a pendulum. By knowing the object's properties, the fiducial tracking can be used to detect how far the object can be lowered. As the box is lifted on the back side of the ballbot there is no need to check if the box blocks the lidar, which could be the case on the front side.

To transport the object to a different location the previously calculated center of mass shift is combined with the trajectory planning angle from the differential flatness path planner. Due to the ballbot's omnidirectional maneuverability the ballbot can drive in any direction, but it is preferred to drive into the lidar facing direction to avoid obstacles. The end position of this transport is again set 0.3 m in front of the actual end position. After lifting the box above the height of the table and yawing to the correct angle, the ballbot can drive to its actual target and slowly lower the box onto the table. This step is again critical, as the center of mass will change very fast as soon as the object touches the table. A successful set down can be detected in a similar way as a successful lifting. Either by using the needed torques of the arms or by detecting, that the box is in a stable position while the arms are still moving down. To avoid, that the arms hit the table, the arm movement is stopped after a confirmed set down.

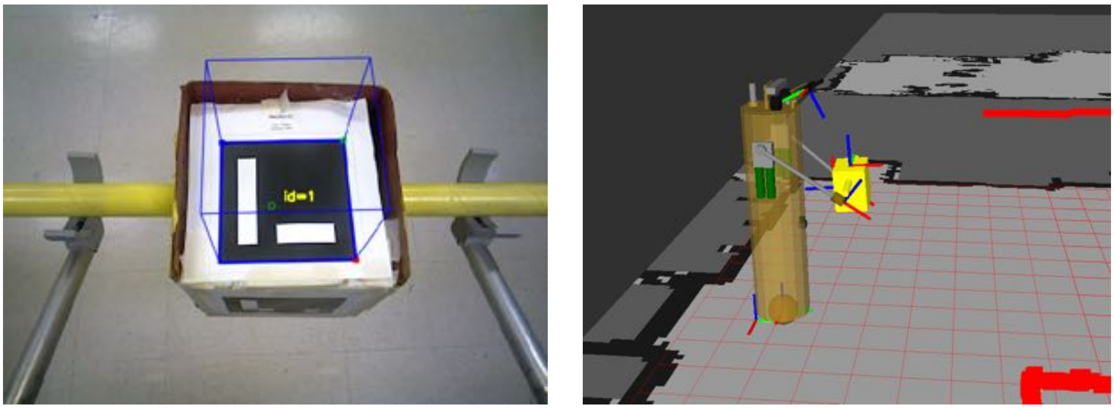
Ballbot will then drive 0.3 m away from the desk and lower its arms to the zero position. After this successful transport, the ballbot will use the differential flatness path planner to drive to its home position and lower its legs to achieve a stationary stable position while waiting for new commands.

4.3 Object detection using Fiducial tracking

This chapter will focus on the detection and localization of object which should be transported. As the main part of this thesis is focusing on the actual lifting and transporting task, only one of the presented object detection approaches was implemented. As the first presented approach, which uses fiducial markers, is more reliable than the second which suggested a sensor fusion between the RGB image of the camera and the 2D lidar, the marker approach was implemented. Both methods are able to scan 360° around the ballbot using the pan and tilt turret.

The implemented solution uses the ARUCO framework [38, 39] to track fiducial markers. To enable the ballbot's onboard camera to work with the fiducial detection and tracking system it was necessary the calibrate the camera. This is done by recording a chessboard pattern with a known dimension using the ROS camera calibration software. The Asus Xtion pro camera publishes a raw RGB image to the ROS network using the Openni driver. This constant video stream is then analyzed by ARUCO. As soon as ARUCO detects a valid fiducial marker it publishes its position as a frame in the ballbot's transformation tree. It can publish up to 2 different fiducial marker positions simultaneously. A python script running on the Ubuntu on board pc will constantly check if the markers and frames are still valid.

As long as they are valid, a PD controller will keep the markers in the center of the camera frame using the pan and tilt of the turret. If no valid marker is found, the camera will be moved to scan the ballbot's surroundings. The centered marker can be seen in Figure 4.4a while the ballbot is lifting the box. Furthermore, this Figure shows, that ARUCO detects the ID of the marker. This ID can be used to uniquely identify a certain object and therefore its physical properties such as weight, size or preferred pickup motion. Another possibility would be to use the ID as an address for the objects target location.



(a) Openni camera view with augmented reality overlay of ARUCO fiducial detection

(b) View in Rviz visualization:Box location generated by fiducial tracking

Figure 4.4: Object detection and localization with ARUCO fiducial tracking

Figure 4.4b shows the ballbot and the detected box in the Rviz visualization. To detect a successful lifting, the hand frame and the box frames can be compared. The detected location of the box will be used in the following chapter to navigate to the object and lift it.

4.4 Motion planning

This chapter will present the different steps to align the ballbot with the object and the needed arm motions. The first part will cover the approach near the box and how the body will be correctly aligned. The second part will cover the needed arm trajectories to lift, carry and release different objects at different locations.

4.4.1 Path Planning and Yawing

As described in the high-level approach the ballbot needs to navigate to several different locations. To plan the navigation path the differential flatness path planning is used. It receives the target goal as a Posestamped message via the ROS network and provides the balancing controller with the needed lean angles and ball velocities.

The main control script which is running on the Ubuntu computer sends those goal messages to the path planner. After detecting and localizing the box, the control script calculates a position 0.3 m in front of the front side of the box. This is done by using the frame of the fiducial marker to align perpendicular to the correct side of the box. The next step is yawing the ballbot so that its back-side is aligned with the object. The yawing is discussed at the end of this section. As soon as the alignment is done, the ballbot drives closer to the box using again the frame of the fiducial marker. After a successful lifting maneuver the ballbot navigates away from the desk using its body frame. The navigation to the destination and home position is carried out in the map frame.

Additionally to the autonomous driving, the target position can be set manually to cover a wider variety of tasks. The manual goal can be set, by either placing the goal on the Rviz map or specifying Cartesian coordinates in a certain frame. The available frames of the ballbot's transformation tree can be seen in Appendix B. Setting a goal on the map is especially useful if the ballbot shall navigate to a human, as humans are visible as an object on the map. Figure 4.5 shows a screen-shot of this approach. The target is visible as the big coordinate system on the left side. The green line on the floor is the planned path for the ballbot. The black markings on the floor are objects and the light gray area is where the ballbot is able to drive.

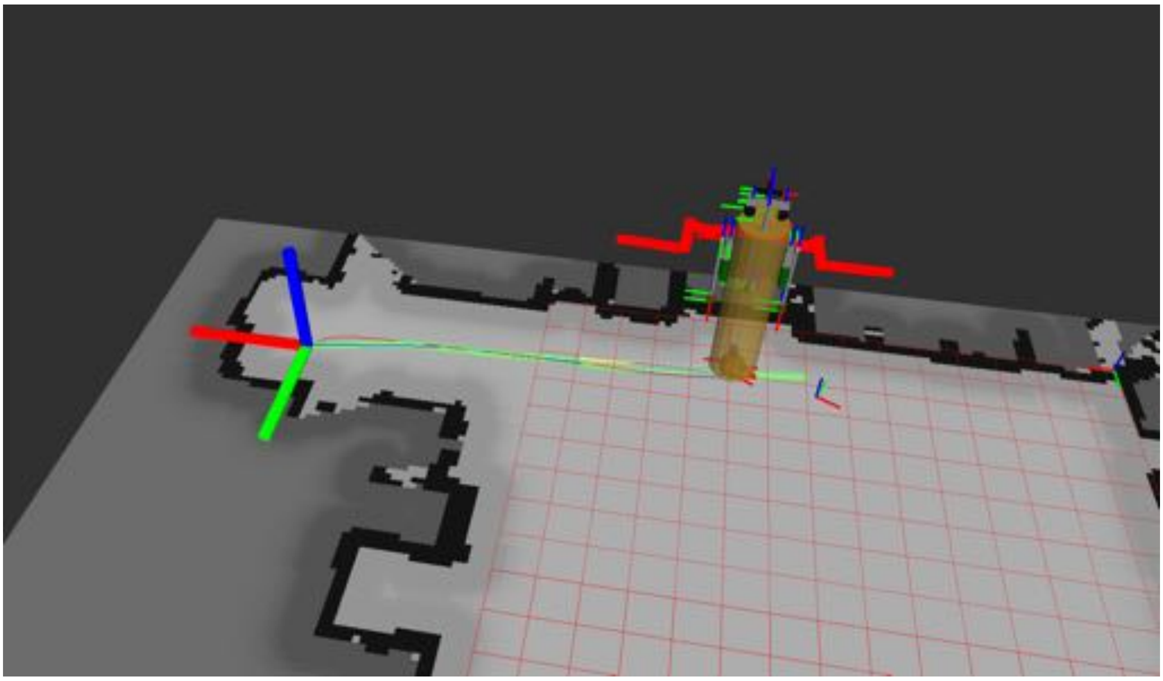


Figure 4.5: Path planing visualization in the Rviz simulation

Without transporting objects the ballbot is able to receive step commands for the yaw motion. The yaw controller will then turn the ballbot to the desired direction. As the inertia for the yaw axis changes, while transporting objects, the controller might under or overshoot, as it is not tuned for this inertia. To perform a smooth yaw movement a ramp with a predefined speed is sent to the yaw controller.

4.4.2 Arm movement

The arm movement needs to be split into two different lifting approaches. The first one is used for objects without a handle. The object gets lifted by applying inward pressure, as the arms press against the box. The second movement can be used for objects with handles.

To lift an object without handles the hands with the flat Velcro$^{\text{TM}}$, presented in Figure 2.5d needs to be attached to the arms. Depending on the outside surface of the object either the Velcro$^{\text{TM}}$ surface or the soft rubber surface is used. While the Velcro$^{\text{TM}}$ provides more grip during lifting and carrying, it is difficult to release the arms after a successful transport safely. Additionally to this issue the box need the be fitted as well with the opposing Velcro$^{\text{TM}}$ surface. The rubber surface is more versatile compared to the Velcro$^{\text{TM}}$, but lacks in terms of friction.

Lifting an object with handles is easier for the ballbot's 2 DOF arms, as they do not have to apply inwards pressure. To lift these objects special hands were designed and manufactured using a 3D printer. The hands presented in Figure 2.5e and Figure 2.5f have both a U-shaped opening to accommodate the handle. The later ones are bigger and allow the angle of the U-shape to be adjusted. This was a necessary modification to ensure a safe transportation even if the box is pressed against the body.

To get a better understanding of the needed trajectories a teach and play function was implemented. It has two different modes, recording of an arm trajectory and performing a prerecorded trajectory. During recording the arms are set to compliant mode. This allows the operator to move the arms with a minimal torque effort. The position and acceleration values of each arm are recorded at either 10 Hz or 50 Hz. After a successful recording, the trajectories are saved locally as comma separated value (CSV) files. The second mode lets the operator select one of those recorded files and sends the corresponding commands to the arms. To achieve a higher precision, these CSV files can be generated as well with other applications such as Matlab$^{\text{TM}}$.

Figure 4.6 shows the performance of the teach and play mode. This figure shows the arm angles in X and Y direction during a grabbing movement. The blue line is the recording, where the arms were moved manually in compliant mode. The red line shows the replay of that recorded movement. The positions match with a mean square error of less than 5%. The X-axis shows the time in 0.1 s steps, while the Y-axis displays the arm angles in radians.

Whereas the teach and play mode works well if the objects location and properties stay the same, a more general approach is needed to conquer a bigger variety of situations. Therefore, the objects position and ID, acquired by the fiducial tracking, are used to position

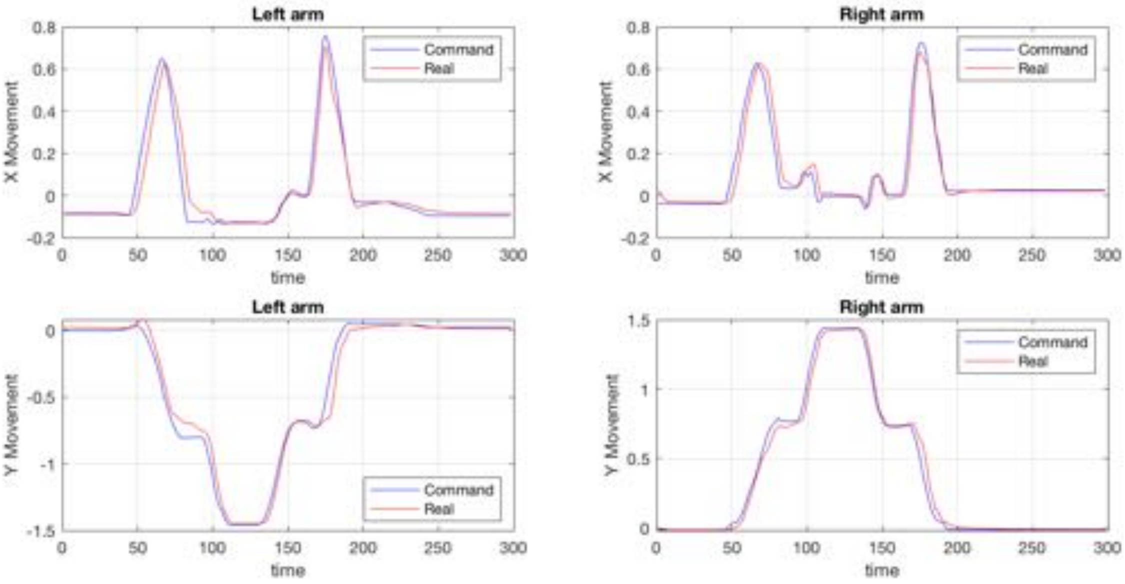


Figure 4.6: Teach and Play comparison: command vs.real angle

the arms accordingly. Depending on the ID one of the two different lifting approaches is selected. The height of the box, relative to the ground determines how high the arms need to be raised. Figure 4.7 shows, that the arms will be raised in two steps. First, the arms are lifted above the table top, but below the handle of the box. After readjusting the ballbot's position, the second step of the lifting is performed. In this step the actual lifting takes place. As it can be seen in Figure 4.7 the velocities and accelerations during the second step are slower than in the previous step. This slower movement helps the ballbot to determine the correct weight of the box and adjust its lean compensation angle. The small red circles in the angle plot are the calculated endpoints, which ensure a successful lifting.

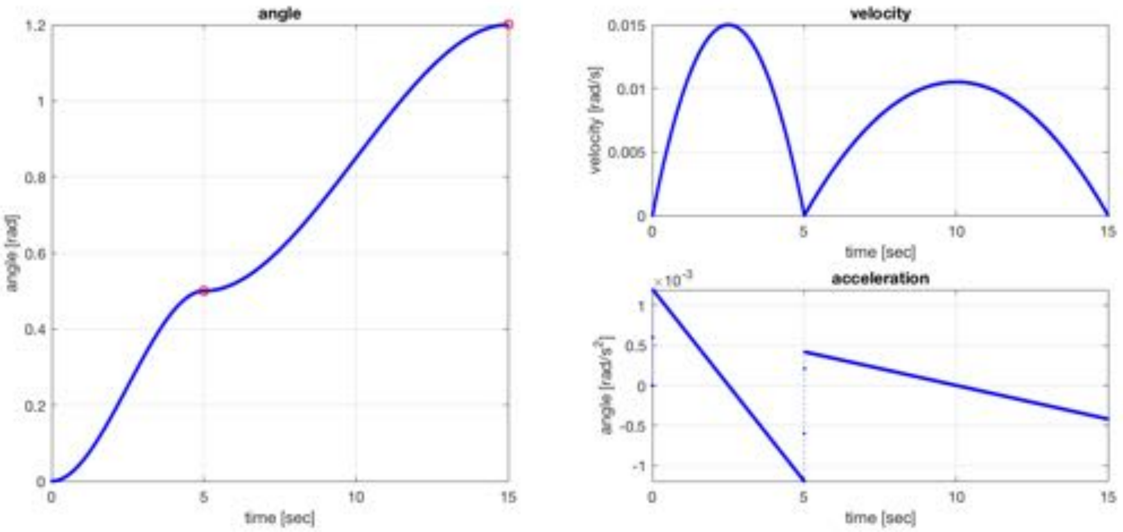


Figure 4.7: Arm y movement for lifting a box with handles

After moving away from the table the box properties are used again to calculate how far the arms can be lowered. To achieve the least disturbing forces introduced by the box, it needs to be lowered to the ballbot's hull as close as possible.

4.5 Object Mass detection

To keep the ballbot stable while interacting with objects, it is necessary to understand the external forces introduced by them. As the ballbot shall lift and carry objects the main interfering force is the added torque on the arms introduced by the mass of the object. This chapter will show four different solutions how the mass of the object and therefore the additional forces can be detected. These results will help to calculate the needed lean angle to compensate for the interactions.

The first two solutions rely on the series elastic actuators of the 2 DOF arms. The third presented solution uses the object identification provided by the fiducial marker tracking system. The last solution assumes that neither of those torques or identification information is available and uses a control based approach.

4.5.1 Calculation

The first presented solution uses the torques received from the Series Elastic Actuators in the arms. The torques can be calculated using Equation 4.1, where the difference between the two encoders is multiplied by the spring coefficient.

$$\begin{aligned} Spring_{coefficient} &= 16.37 \text{ Nm/rad} \\ Gear_{ratio} &= 127 \\ Torque &= \frac{Encoder_{top} \cdot Gear_{ratio}}{Encoder_{bottom}} \cdot Spring_{coefficient} \end{aligned} \tag{4.1}$$

Each arm has two springs and therefore two spring torques. $Torque_{Spring0}$ is the torque of the front side spring $Torque_{Spring1}$ is the torque of the back side spring. The Equations 4.2 use those spring torques to calculate the Mass per Arm in X and Y direction using the Arm_{length} of 0.56 m. The additional multiplication by two is due to the 2:1 ratio of the bevel gears.

$$\begin{aligned} Mass_X &= \frac{\frac{2 \cdot (Torque_{Spring0} - Torque_{Spring1})}{Arm_{length}}}{9.81 \frac{\text{m}}{\text{s}^2}} \\ Mass_Y &= \frac{\frac{2 \cdot (Torque_{Spring0} + Torque_{Spring1})}{Arm_{length}}}{9.81 \frac{\text{m}}{\text{s}^2}} \end{aligned} \tag{4.2}$$

4.5.2 Polynomial estimation

This solution for the mass detection uses the raw spring torsion. Each arm publishes its spring torsion and arm angles to a ROS topic. A separate laptop computer running Matlab^{TM} can read and record this topic via Wi-Fi. Additionally, a Matlab^{TM} script was implemented which can send arm trajectories to the the ballbot using another ROS topic. Matlab^{TM} was used to generate arm calibration trajectories. These generated trajectories are shown in Figure 4.8. Each arm is moved five times from 0° to 90° with a weight of up to 2kg per arm. For higher loads, the range is limited between 40° and 90°, as this is the range where the ballbot can successfully transport an object without interfering with its body. During those movements, the spring values, as well as the corresponding arm angles and attached weights, were recorded. Figure 4.9 shows the curve fitting tool of Matlab^{TM} was used to fit a polynomial curve into the dataset. The X and Y axis shows the arm angle and the torque values from the spring, while the Z axis shows the correlating mass. Different mass levels are visual, as the mass was increased in 1kg steps during calibration. This procedure was done for each arm individually to get the best results.

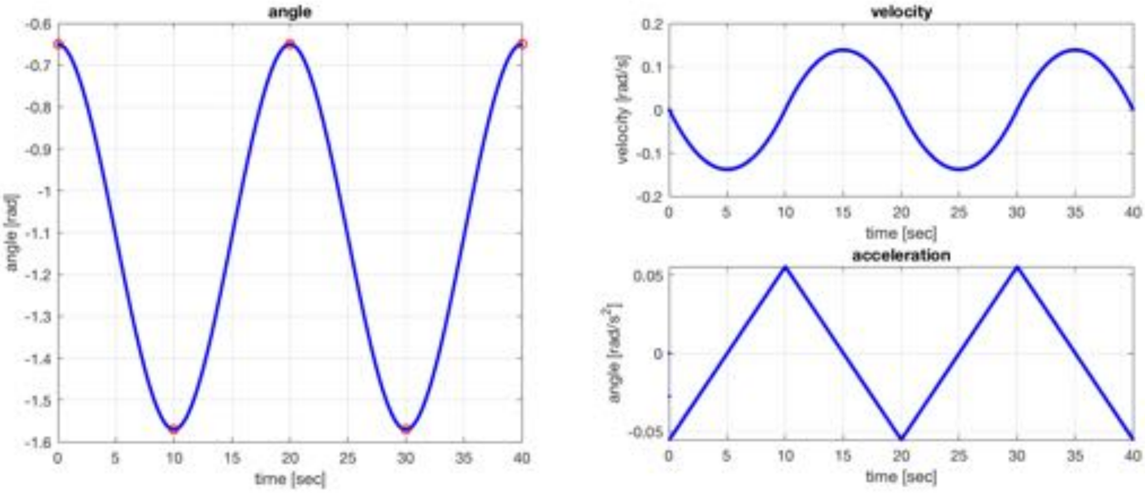


Figure 4.8: Calibration trajectories generated in Matlab^{TM}

4.5.3 Detection through object identification

Whereas the two previously presented approaches use the Series Elastic Actuators, this method does not need any information from the arms. This approach uses the object identification and a database which provides the accurate weight for each object. Using the ARUCO fiducial detection it is trivial to distinguish between different objects. Each object will be outfitted with a unique fiducial marker ID. This ID will then be used to identify the object and receive the correct weight of it from a provided database. As there is no need for a torque detection, the Series Elastic Actuators located in the arm assembly could be

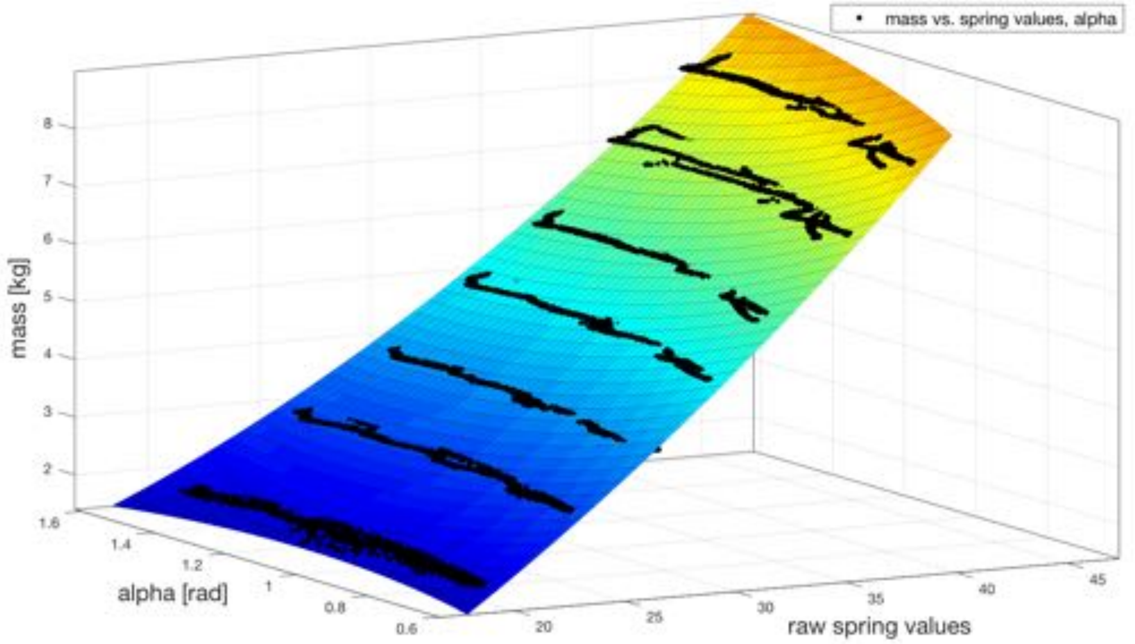


Figure 4.9: Determination of the polynomial function using Matlabs^{TM} curve fitting tool.

changed to static actuators, which would enable the ballbot to lift higher weights, as it does not need to stay below the protection limits of the spring elements. Whereas the benefit of this method would be a very accurate object weight and therefore a precisely matched lean angle compensation, as well as an increased lifting capability of the ballbot, there are some drawbacks which need to be considered. Using only this method the ballbot would not be able to detect if the object has a different weight than provide by the database and compensate for a wrong weight. The second disadvantage of this method is, that sudden changes in the weight of the object would not be recognized. Therefore a human interaction, where an operator receives an object from the ballbot would not be feasible.

4.5.4 Control based lean angle compensation

The following approach neither uses the torque measured by the Series Elastic Actuators in the arms nor predefined object weights. It relies on the center of mass compensation performed by the station keep controller which was explained in detail in Chapter 2.5.2. The station keep mode adjusts the lean angle of the ballbot to keep it at the position where station keep was engaged. If the controller detects external forces such as introduced during a lifting motion the center of mass offsets gets adjusted. Using the inverse center of mass calculation the force at the end of the arm and therefore the mass can be calculated. As this approach only relies on the station keep controller a very slow lifting motion is needed. Even with a 50% slower lifting motion, the station keep results are not as good as with the mass detection, as discussed in detail in Chapter 5.5.

4.6 Center of mass feed forward control

Adjusting the lean angle of the ballbot to keep it gracefully balanced is the main task for the real-time QNX computer located on the ballbot. The Figure 4.10 shall help to get a better understanding of the different compensation methods and desired lean angles. While the upper part of the flowchart shows three different center of mass compensation routines, the lower part shows the chain of command for the ballbot's lean angle in outer loop control. Prior to sending the odometry data to the outer loop path planner, the combined center of mass offsets are subtracted. Therefore the outer loop planer, which is, in this case, the differential flatness path planer, receives a zeroed lean angles without the center of mass offsets. After calculating the desired lean angle and ball velocity, those offsets are added to resemble the real values needed for the ballbot. This needs to be done to avoid the need of constantly updating the dynamic model of the ballbot used for the path planning. The PID balancing controller tries to minimize the error between the real lean angle, the ball velocity, and the desired values.

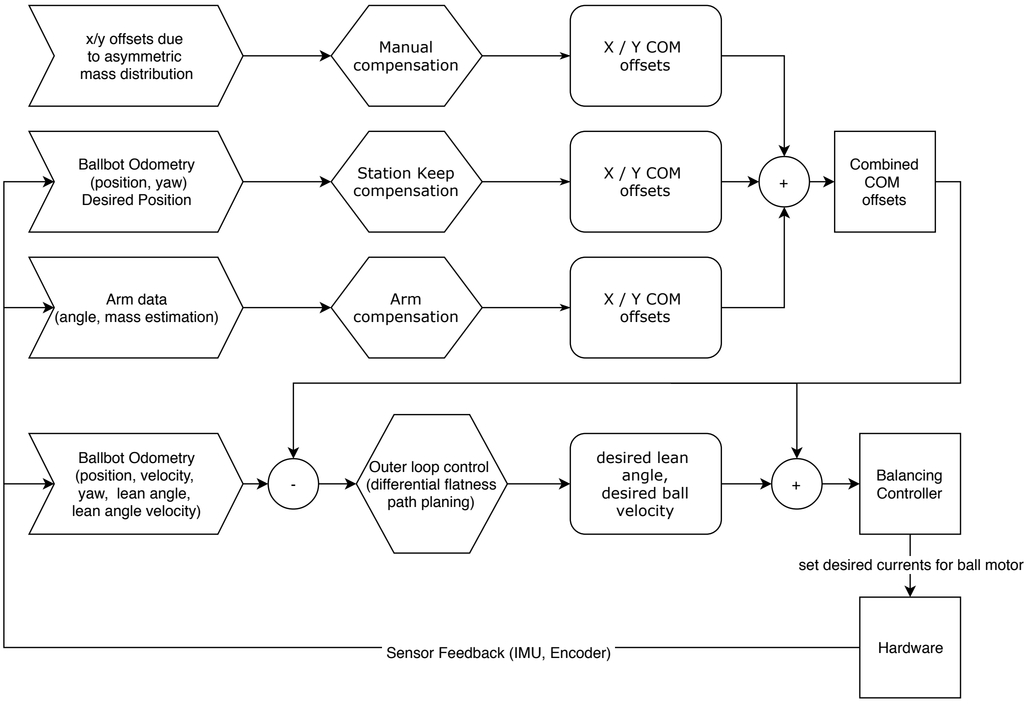

x/y offsets due to asymmetric mass distribution
Manual compensation
X / Y COM offsets
Ballbot Odometry (position, yaw) Desired Position
Station Keep compensation
X / Y COM offsets
+
Combined COM offsets
Arm data (angle, mass estimation)
Arm compensation
X / Y COM offsets
Ballbot Odometry (position, velocity, yaw, lean angle, lean angle velocity)
-
Outer loop control (differential flatness path planing)
desired lean angle, desired ball velocity
+
Balancing Controller
set desired currents for ball motor
Hardware
Sensor Feedback (IMU, Encoder)

Figure 4.10: Center of mass compensation and lean angle control overview

The combined center of mass offsets consists out of three different lean angle compensation branches. The manual compensation is static and can be used if the in-balance due to not symmetrically arranged components is known. It can be set prior the ballbot starts balancing and should not be changed afterwards. The second branch operates only in station keep mode. After the ballbot transitions from a stable position with all three legs down to balancing, station keep mode is engaged to avoid the ballbot drifting in one direction if the manual compensation is not exactly accurate. The station keep compensation uses the odometry data and tries to avoid changes the ball angle and therefore keeps the position stationary. These computed center of mass offsets are saved and used, even if the balancing state changes to outer loop control, where the desired lean angles are commanded by the path planner. The third branch is the newly introduced arm compensation, which will be presented in greater detail.

As described in the previous chapter the mass of the object attached to the arms can be detected. Therefore the arm compensation can compute the needed lean angle to counteract these additional forces. To calculate the shifted center of mass, the body center of mass position and the arm center of mass position are calculated separately. Equation 4.3 calculated the center of mass for the body using $L_{com} = 0.71$ m as the body center of mass along the z-axis and ϕ as the ballbot's lean angle.

$$\begin{aligned} Body_{comx} &= -L_{com} \cdot \sin(\phi) \\ Body_{comy} &= L_{com} \cdot \cos(\phi) \end{aligned} \tag{4.3}$$

To calculate the center of mass for the arms, Equation 4.4 is used, where $L_{armjoint} = 1.3m$ is the shoulder joint height, $L_{arm} = 0.56m$ is the arm length and α is the arm angle. As the lifting is either performed on the front or back side of the ballbot only a 2D model needs to be considered. The arms itself are assumed to be weightless.

$$\begin{aligned} Arm_{comx} &= L_{armjoint} \cdot \sin(\phi) - L_{arm} \cdot \sin(\phi - \alpha) \\ Arm_{comy} &= L_{armjoint} \cdot \cos(\phi) - L_{arm} \cdot \cos(\phi - \alpha) \end{aligned} \tag{4.4}$$

The body and arm center of mass offsets are combined to the new center of mass of the whole system in Equation 4.5. $M_{body} = 81.65kg$ is the mass of the ballbot and M_{arm} is the weight of the object received by the object mass estimation.

$$\begin{aligned} System_{comx} &= \frac{Body_{comx} \cdot M_{body} + Arm_{comx} \cdot M_{arm}}{M_{body} + M_{arm}} \\ System_{comy} &= \frac{Body_{comy} \cdot M_{body} + Arm_{comy} \cdot M_{arm}}{M_{body} + M_{arm}} \end{aligned} \tag{4.5}$$

Using the Equation 4.6 the needed compensation angle can be calculated and added to the combined center of mass offset angles, which enables a gracefully balancing while lifting objects, as it can be seen in the following chapter.

$$CompensationAngle = \text{atan} \frac{System_{comx}}{System_{comy}} \tag{4.6}$$

5 Results

The following chapter will discuss the key elements of a successful performed lifting and transportation routine. Furthermore, the experimental results of different approaches are shown. The key elements consist of a basic feasibility study which was performed in Matlab ™, the object detection, localization and tracking abilities, a precise arm movement, a robust object weight detection and the most important part a correct lean angle adjustment.

5.1 2D Simulation

To ensure the center of mass feedforward term described in detail in Chapter 4.6 will work as expected, it was necessary to implement it first in a simulation. Using the simulation prior to experimenting on the actual hardware can partially prevent that the ballbot loses its balance and gets damaged. Therefore an already existing 2D simulation of the ballbot, written in Matlab ™, was used. As the task of lifting an object with the 2 DOF arms of ballbot is symmetrical there was no need for a much more complex 3D simulation. Working approaches in the 2D simulation can be used on ballbot's sagittal and coronal plane to achieve a 3D lean angle compensation. The existing simulation did not take additional loads on the arm into account and was therefore modified. The simulation uses the properties of ballbot received from the system identification process described in detail in Chapter 2.6. The dynamic model used by this Matlab ™ simulation was presented in Chapter 2.7.

Figure 5.1 shows three time steps in during a simulated lifting movement. To provide a better visualization of the lean angle an object weight of 20 kg was chosen. In this case, it was assumed, that the mass was detected by the object ID and therefore accurate. This results in no lateral movement of the ballbot as the body lean angle is always adjusted correctly. Figure 5.2 shows the lean angle, arm angle and ground position over time for this test. While the arms raise to a horizontal position, the body of the ballbot leans back.

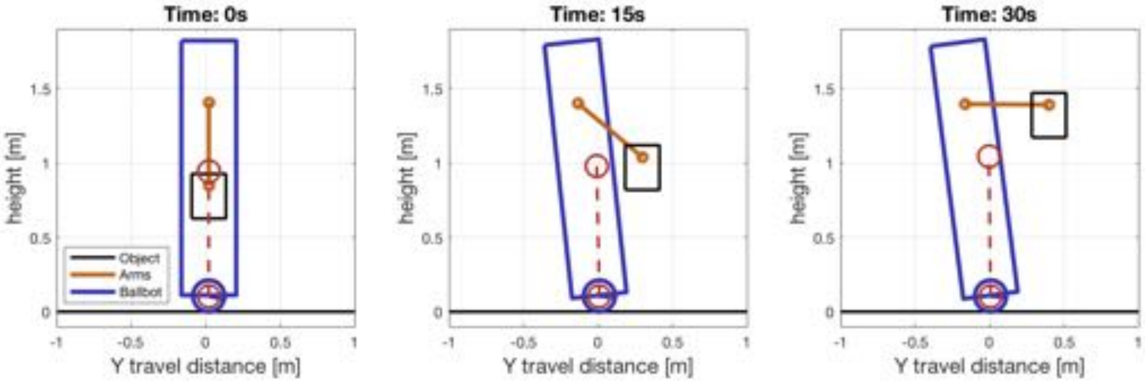


Figure 5.1: 2D Matlab ™ simulation of station keep mode at three different times, while raising its arms.

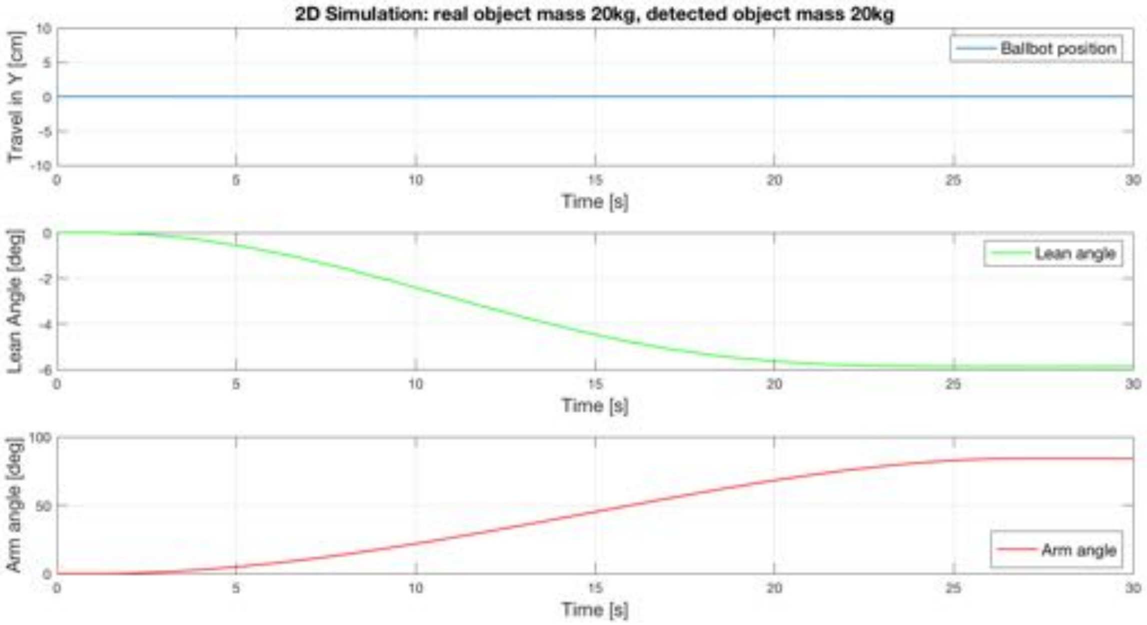


Figure 5.2: Plot of lean angle, arm angle and ground position over time.

By applying noise or offsets to the arm mass, the expected behavior with other, less accurate, weight detection methods could be predicted. Figure 5.3 shows the same lifting motion as the first test, but in this case, the mass detection was set to estimate 50% less. This resulted in a stable end position of the ballbot, but with a drift of 0.8 m in the direction of the lift. A visualization of the lean angle, arm angle and ground position over time is presented in Figure 5.4. These results from the 2D simulation matches with the results of the real ballbot presented in 5.5. A drifting location during lifting can be detected in both experiments.

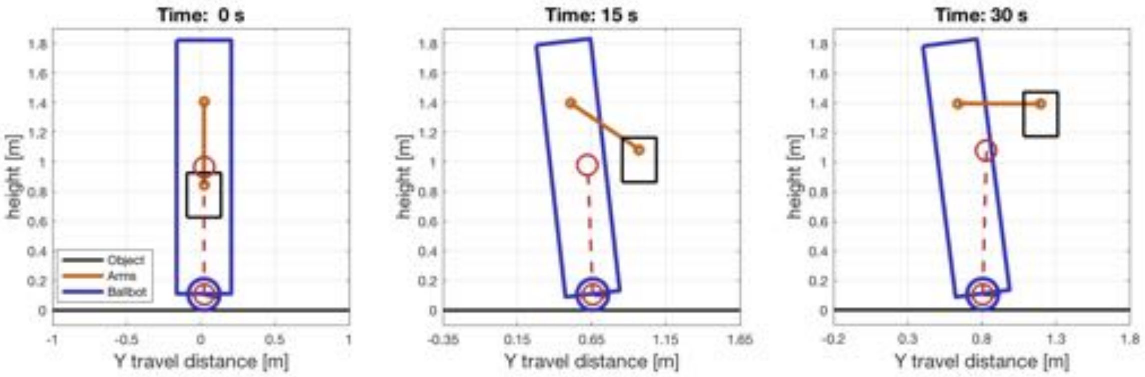


Figure 5.3: 2D Matlab ™ simulation of station keep mode at three different times, while raising its arms with 50% less detected weight.

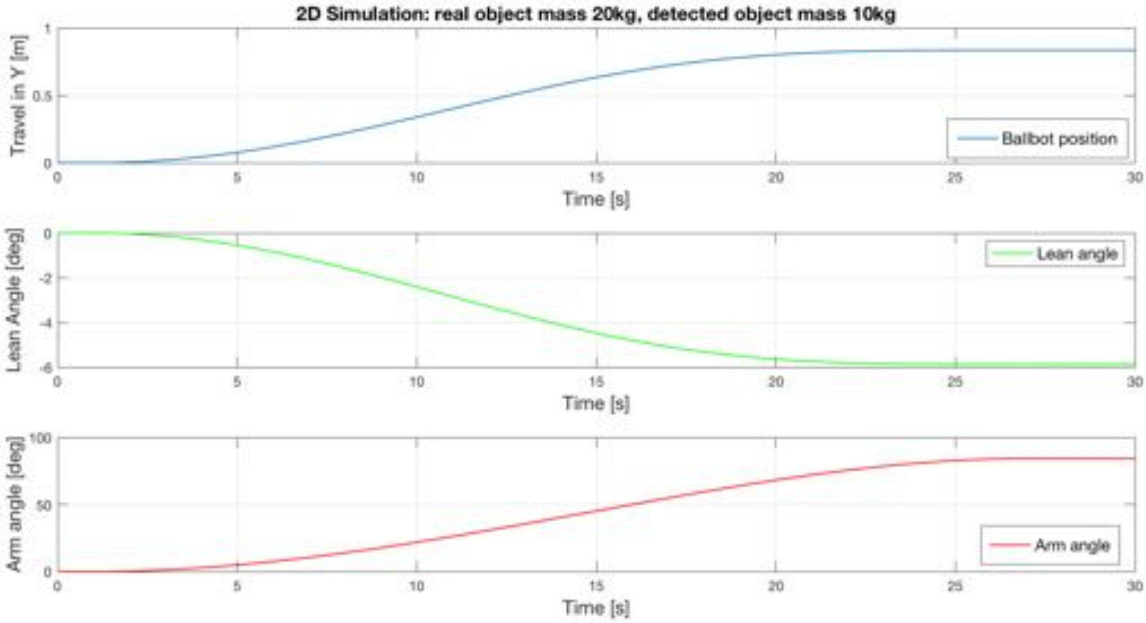


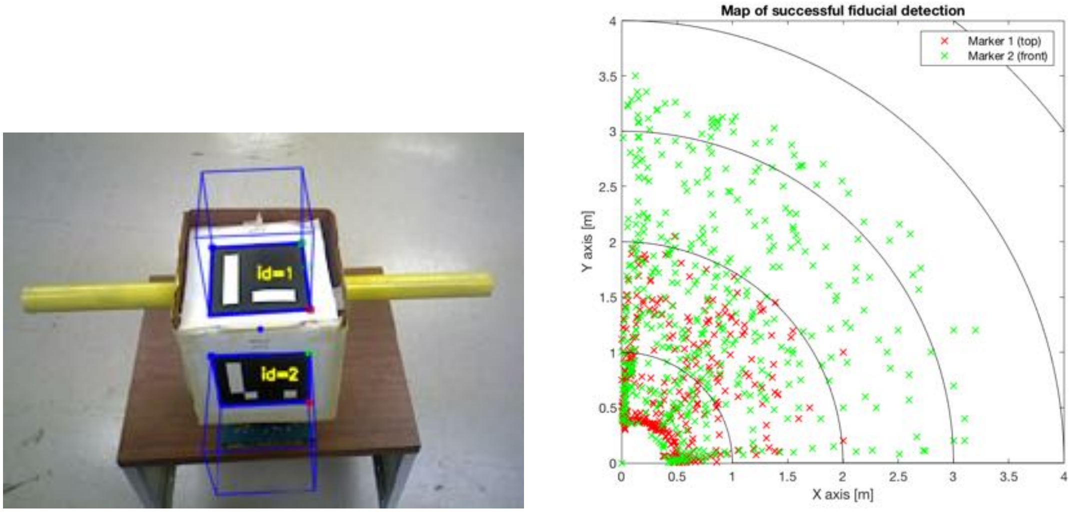
Figure 5.4: Plot of lean angle, arm angle and ground position over time with 50% less detected weight.

Concluding, the proposed lean angle compensation works in the 2D simulation. Even when the mass is not estimated fully correct, the ballbot will keep its balance. The first real-world test on the robot shall be done with low masses attached to the arms. Furthermore, the mass should be hard-coded for the first tests to ensure the compensation angle calculation is working as expected.

5.2 Object detection

Prior the object mass detection is needed, the object needs to be localized. This chapter will cover the implementation to detect and locate the object which shall be transported. Due to the highest reliability the ARUCO fiducial tracking [38, 39] presented in chapter 3.1.1 is used. The Asus Xtion pro live camera has a 58° horizontal and 45° vertical field of view. Due to the used Openni driver, a maximum resolution of 640 by 480 is available. The camera is mounted on a turret which can independently pan an tilt, as it can be seen in Chapter 2.2. This ensures a 360° coverage around the ballbot.

As soon as the camera stream and tracker are launched, the turret starts to pan and tilt the camera. Similar to humans it looks for fiducial markers in its range of sight. After detecting a marker a proportional controller keeps the marker centered in the frame. A test proved, that the tracking works during, station keeping, navigation and yawing. If more than one marker is detected, the marker with the lowest ID will be tracked in the center of the camera frame. The position of up to two markers will be published to the ROS network, where other nodes, such as the arm and position controller, will use those position and orientation information to interact with the object.



(a) The box contains to fiducial markers. The marker with ID = 1 on the top of the box and the marker with ID = 2 on the front side.

(b) Map of successful object detection and localization

Figure 5.5: Results: Object detection and localization with ARUCO fiducial tracking

The object was fitted with two fiducial markers with a size of 15 by 15 cm each. As it can be seen in Figure 5.5a the marker on the top of the box has the ID 1 and the marker in the front has the ID 2. Two markers on different sides of the box were needed, two ensure a detection at a distance, as well as in close proximity. Figure 5.5b presents the results of a benchmark test, where the object with the two fiducial markers was placed on a table in 0.9 m height. The marker with the ID 2 was facing the ballbot. During the test, the box was moved around randomly between ballbot's front and 90° to the right side. The benchmark script was running at 5 Hz. Each time a valid pose of the object could be detected it was marked on the map. The red crosses symbolize a detection of the top marker, the green crosses symbolize the location of the front marker. The box can not get closer than 0.3 m, as it would interfere with ballbot's body. The top marker is detected to a maximum of 2m radius from the ballbot. The marker in the front cloud be detected up 3.5 m away. As the turret can rotate 360°, the result received from this 90° benchmark test can be expected all around the ballbot.

Modifying the marker size might increase the range, as it is dependent on the camera's resolution. A test with the resolution reduced to 320 by 240 QVGA at 30 Hz showed that the detection range was as well reduced to 2.5 m. This shows, that with a greater resolution objects further away could be detected. The newest generation of ARUCO presented in 2018 is able to process 4K images at a rate of over 1000 fps [39].

5.3 Arm controller

As the previous section showed, the ballbot is now able to reliably detect and localize the target object. Prior the objects mass can be estimated, the performance of the arm controller needs to be evaluated. ballbot's arm controller is implemented on the real-time QNX system. The arm controller has three different modes. In the stationary mode, the arm motors are not supplied with current. This mode is set during the initialization of the ballbot and can be used to move the arms manually to the desired position. The second mode is called compliant mode, it is controlled by a PID controller which tracks the torques generated by the motors. Its goal is to keep the needed force it takes to move the arm as low as possible and compensate for the gravity component. The mode used for all lifting tasks is called position mode. In position mode, the PID controller tracks the desired X and Y angle trajectories. Figure 5.6 shows the performance of this controller while moving the arms from their start position to a 90° angle in front of the ballbot and back to start. The black dashed line is the command, in this case, generated by Matlab $^{\text{TM}}$, and the blue and red lines show the right and left arm angles. As it can be seen in the second plot of Figure 5.6 the position error get less, the higher the arm gets raised.

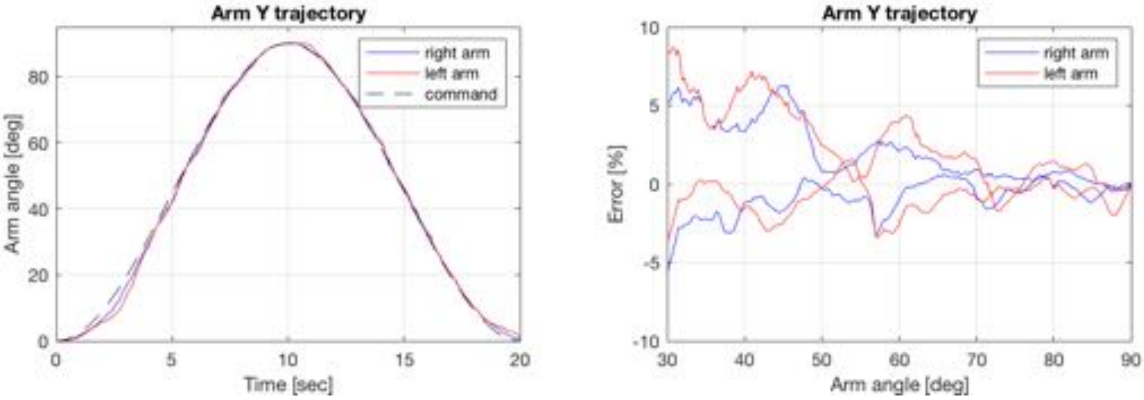


Figure 5.6: Left plot: arm command vs. real angles while lifting 2kg on each hand. Right plot: arm position error at a certain angle while performing a lifting movement.

Tests with different loads attached to the arms showed, that this controller performs well over the needed arm mass range, which will be further discussed in the following chapter. As Table 5.1 shows the average error stays continuously below 5%. Due to the less important slack, the position error decreases by increasing the weight.

5.4 Weight detection and limits

As the ballbot is now able to detect the box and move its arms reliable, the weight of the lifted object needs to be determined. Detecting the exact mass of the lifted object is critical to adjust the lean angle by the correct amount. But it is equally important to ensure, that the

Attached weight per arm	average error
no end-effector	2.75%
hand used for lifting	1.4%
1 kg	3.56%
2 kg	2.43%
3 kg	2.73%
4 kg	1.81%
5 kg	1.32%
6 kg	0.83%
7 kg	0.9%

Table 5.1: Average error while preforming a lifting to 90° and back. The weight is attached to each arm.

lifting won't damage any parts of the ballbot. Therefore, this chapter will start focusing on the maximum ratings prior taking a look into the different methods of weight detection.

5.4.1 Maximum liftable mass with the current setup

This section will present the maximum liftable mass for each arm with the current setup to prevent any damaged to the springs. As it can be seen in Figure 5.7 each arm gets driven by two motors with connected gearboxes. Each of those gearboxes is connected to a spring with a constant of 16.37 Nm/rad [4]. Using this spring constant, the custom spring dimensions and the spring calculation tool of the spring manufacturer Planetspring [59], a maximum possible torque $Spring_{maxQ} = 15.648$ Nm and a maximum save travel angle of 54.76° was calculated. These are the theoretical absolute maximum values and shall never be exceeded to prevent the spring from permanent deformation.

Using the Equation 5.1, the maximum arm torques in X and Y direction are calculated. The results are higher than the actual spring ratings due to the 2:1 gear ratio of the bevel gears and the load separation on both springs. To lift objects in front or on the backside of the ballbot the Y direction of the arms is mainly used.

$$Torque = 2 \cdot (Spring_{max_q} + Spring_{max_q}) = 62.592\,\text{Nm} \tag{5.1}$$

The equation 5.2 shows, that each arm is capable of lifting $11.39kg$. This results in an absolute maximum object mass of $22.78kg$ with two arms.

$$Mass_{perArm} = \frac{\frac{Torque}{Arm_{length}}}{9.81\frac{m}{s^2}} = 11.39\text{ kg} \tag{5.2}$$

The current setup of the ballbot is limited by the springs. If the maximum torque gets

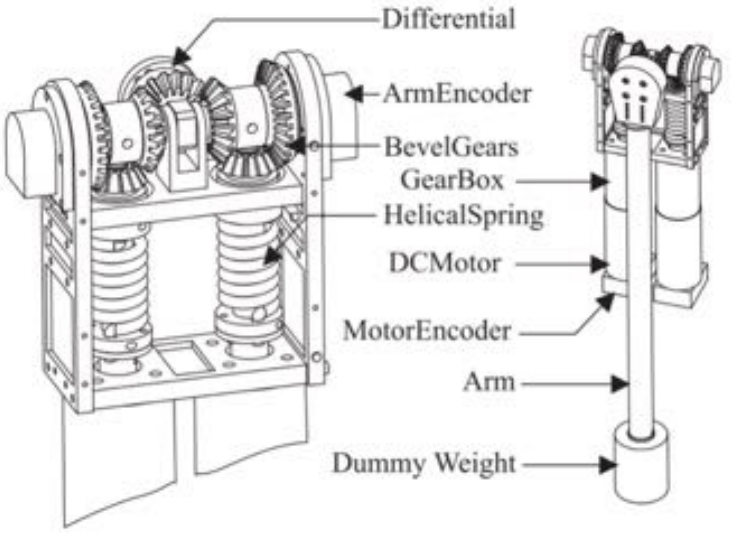


Figure 5.7: Drawing of arm construction using Series Elastic Actuators. (Courtesy: Ralph Hollis; Appeared in [4])

exceeded, they deform permanently and need to be replaced. As the springs are custom made, this needs to be avoided in any case. To ensure these values won't be exceeded by accident an additional safety function was introduced to the system. The detected weight and spring torques will be constantly broadcasted to the ROS network. So other applications can easily detect if they applied force on each of the two arms is too high. Furthermore, the ballbot will constantly monitor these values and give an audio alert, using its speakers. If 75% of the maximum mass and or torque is detected on one of its arms, the ballbot's robotic voice will tell which percentage on which arm was reached. Using this audio alert, provides an additional safety level for the human-robot interaction outside of a laboratory environment.

5.4.2 The accuracy of the mass detection

It is very important to know the accurate weight of the carried object to compensate for the lean angle accordingly. In chapter 4.5, four different methods were presented to get the correct center of mass offset. The first three relay on detecting the mass of the object, while fourth, control based approach, which will be presented in the following chapter.

The easiest way to detect the object's mass is, by externally measuring it and providing this weight in a database. Each object needs to have a fiducial marker with a unique ID and the corresponding weight. Whereas this approach would provide the best results during transport, it has some critical flaws. If the weight changes due to any reason, may it be a wrong entry in the database, a wrong ID or suddenly changed weight during transport, an incorrect compensation will lead to undesired movements.

The second proposed solution is calculating the mass using the two springs, which act as torque sensors, in each arm. As the spring coefficient and the angle of torsion is known, the applied rotational torque can be calculated. As the arm angle and length are known as well, the applied gravitational force at the end of the arm can be calculated. As shown in Figure 5.8 the results received by this method are not ideal for a good angle calculation. The two different lines result from separately measured torques during raising and lowering the arms. The two main reasons for this wrong results are, that the springs in the arms are already about 6 years old and the torque coefficients might have changed over the time due to a lot of use. The second reasons, which introduces some error is the backlash in the arm joints due to bevel gears and normal wear and tear of the hardware components.

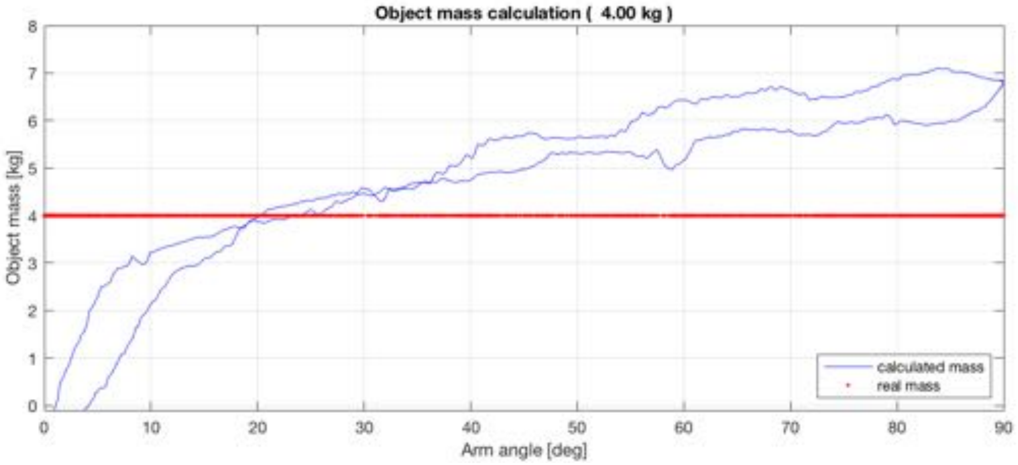


Figure 5.8: Object mass calculation while lifting. Mass used for test was 2kg per hand.

The third presented solution uses a polynomial function, derived by the arm angles and torque values matched to certain masses. To determine that function each arm was moved two times from 0° to 90° with different weights attached. The used weights ranged from an empty lifting up to a 10 kg heavy box. The recorded torques, attached weights, and arm angles were used to fit a polynomial curve with Matlab $^{\text{TM}}$s curve fitting tool into the data set. For each arm, a separate function was generated, due to differences between the two arms of up to 20%. The Equations 5.3 were used, where τ is the torque in Y direction and α is the arm angle.

$$\begin{aligned} Mass_{left} &= -1.482 + 0.1099\tau - 0.8986\alpha + 0.001651\tau^2 - 0.05282\tau\alpha + 0.8132\alpha^2 \\ Mass_{right} &= -4.582 - 0.1858\tau + 0.8223\alpha + 0.0002303\tau^2 - 0.05588\tau\alpha + 1.111\alpha^2 \end{aligned} \tag{5.3}$$

Figure 5.9 shows the performance of this approach while lifting 2kg per hand. Again the two different lines result from individually measured torques during raising and lowering the arms. This solution has issues with angles below 20° as the measured torques are very small. Due to the shape of the ballbot's body only angles above 30° are important for a lifting motion, as the body would otherwise interfere with the objects.

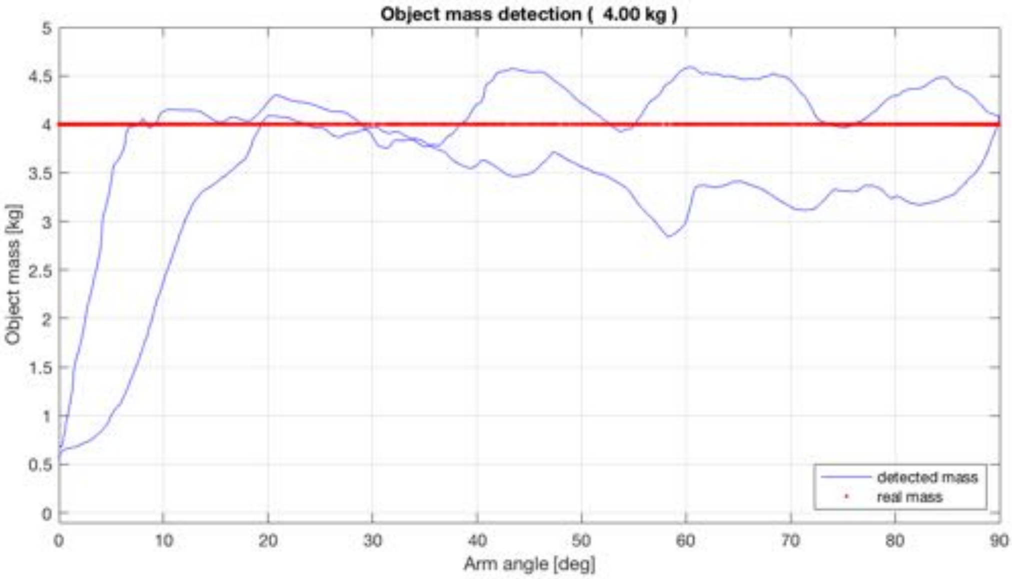


Figure 5.9: Object mass estimation while lifting. Mass used for test was 2kg per hand.

The following Table 5.2 shows the performance of the third solution with different weights and arm angles. As the estimations are in close proximity to the real weights and the weights are continuously estimated, this approach was chosen for all lifting experiments. Even though the maximum lifting limit of the ballbot is about 22 kg, the maximum chosen weight for this tests was 10.52 kg. This prevents forces above the spring limits even if the weight is fully supported by just one arm.

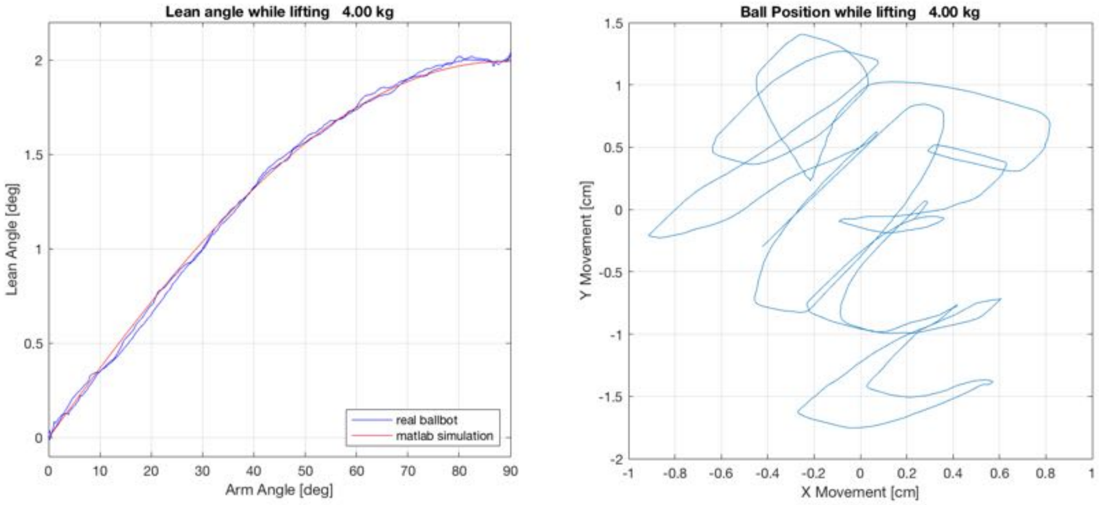
Mass	45 deg	60 deg	80 deg	avg error
0,78 kg	0,81 kg	0,82 kg	0,76 kg	2,14%
1,92 kg	1,91 kg	1,91 kg	1,89 kg	-0,87%
2,78 kg	2,87 kg	2,97 kg	2,76 kg	3,12%
4,01 kg	3,99 kg	4,05 kg	3,98 kg	-0,08%
4,96 kg	5,01 kg	4,92 kg	4,97 kg	0,13%
6,10 kg	5,99 kg	6,01 kg	6,05 kg	-1,37%
7,24 kg	7,21 kg	7,15 kg	7,30 kg	-0,28%
8,50 kg	8,47 kg	8,52 kg	8,60 kg	0,35%
10,52 kg	10,43 kg	10,56 kg	10,59 kg	0,06%
avg error	0,41%	1,06%	-0,39%	0,36%

Table 5.2: Weight estimation by polynomial function

This chapter presented, that the object mass detection works well the whole range the arms can support. The following Chapter will present a comparison of the lean angle compensation with and without mass detection.

5.5 Lean angle compensation

While transporting an object the ballbot must always keep a certain lean angle to compensate those additional forces. This lean angle offset is used during station keep and outer loop control mode. The previous chapter provided information on how the object's weight can be determined, but this chapter will provide results on how good this offset can be detected without knowing or estimation the mass of the object. Figure 5.10a shows a comparison of the simulated and real lean angle. The expected angle in red and the real angle in blue match very closely as the mass was known for this test. Therefore the maximum lateral movement was below 2 cm in Y direction and below 1 cm in X direction, as it can be seen in Figure 5.10b.



(a) Comparisson of the ballbot's lean angle while lifting 4kg vs. Matlab ^{TM} Simulation

(b) Lateral movement of the ballbot while lifting 2kg per hand

Figure 5.10: Lifting of 2kg per hand.

Without knowing the mass a movement of up to 55 cm in Y direction could be detected. As Figure 5.11 shows this movement can be reduced by lowering the speed of the lifting. By doubling the raising time from 10 s to 20 s the lateral movement could be reduced to below 30 cm in the Y direction and in X direction it achieved similar results as the test with a known mass.

These tests show that a successful lifting is possible if the lifting speed is slow enough. An ideal lifting speed could not be determined, as it is highly dependent on the mass of the object, which is unknown in that case.

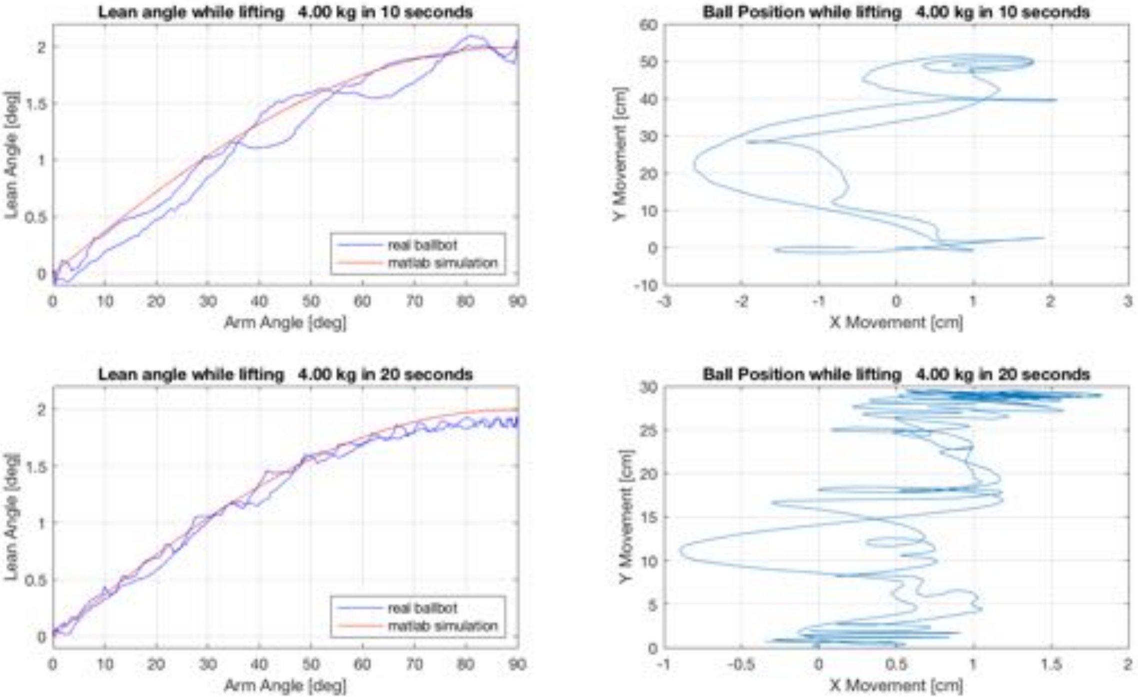


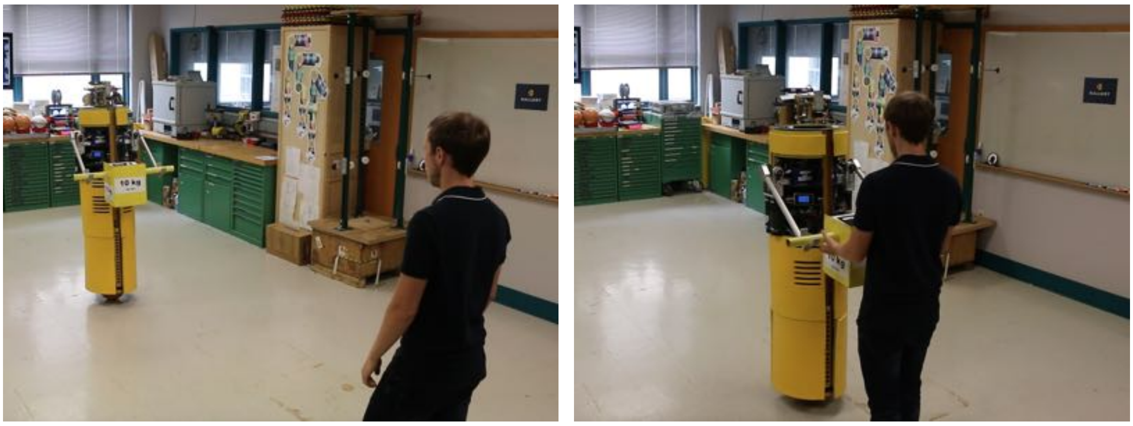
Figure 5.11: Lean angle compensation without knowing or detecting the lifted mass. Lifting duration of 10 and 20 seconds

5.6 Lifting and carrying tasks

This section will give further insight into the real world applications for the new capabilities of the ballbot. It showed in the previously presented individual tests, that it can reliably detect objects using a fiducial marker and calculate their relative position to the ballbot. Additionally the ballbot showed, that it can use its arms depending on the object geometry to perform different lifting or grabbing movements adjusted to the position of the box. To adjust ballbot's lean angle it was a key factor to detect the mass of the object. Various tests showed the different approaches and benefits and drawbacks of each solution, resulting in a good compensation angle calculation. This calculated angle can be used either in station keep while performing a stationary task like an interaction with the object or yawing, or in outer loop control where the ballbot will navigate between different locations while carrying a load. These presented abilities are be combined in the following experiments.

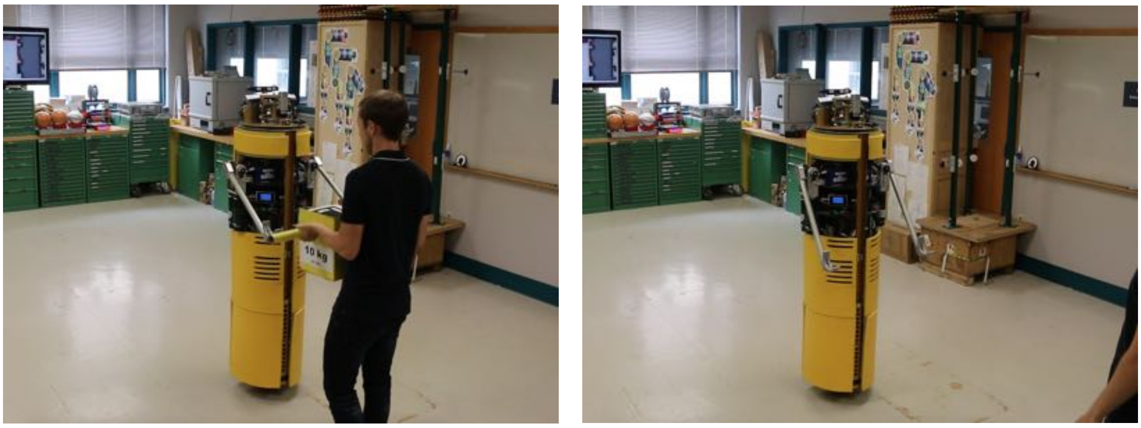
5.6.1 Delivering an object to a human

The first experiment shows the ballbot's abilities to deliver an object with an unknown mass to a selectable target position. In this case, the ballbot is carrying a 10 kg heavy object with its arms raised to 60°. Figure 5.12a shows, the ballbot transporting the object to its target position, which is 0.3 m in front of a test person. The path planning is done by the differential flatness path planning as explained in Chapter 4.4.1. Figure 5.12b shows that target position near the human was reached. The handing over of the object takes less than 5 seconds. In Figure 5.12b the ballbot has no contact to the box anymore. This load change on the arms is detected and the ballbot starts lowering its arms and drive to its home position as is can be seen in Figure 5.12d. This experiment presented the ballbot's ability to deliver an object with an unknown mass and adjust the lean angle fast enough to keep its position while interacting with a human. This test was performed with the maximum load of 10 kg as this high loads introduce the highest lean angle and therefore are more difficult to compensate.



(a) Ballbot transporting object to human (Time: 0:18)

(b) The human is picking up the object (Time: 0:36)



(c) The human has received the object (Time: 0:41)

(d) Ballbot is lowering its arms and driving back to its home position (Time: 0:48)

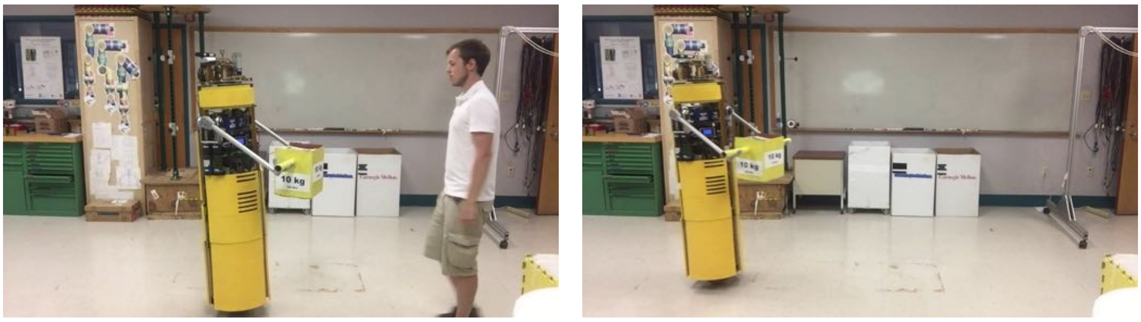
Figure 5.12: Selected screen-shots: Human receiving a 10 kg heavy object from the ballbot.

5.6.2 Receiving an object from a human operator

In the second experiment, the ballbot is receiving a 10 kg heavy object from a human test person. Figure 5.13a shows the ballbot using the differential flatness path planner to drive close the test person. In Figure 5.13b the object is placed on the robots extended arms. It immediately adjusts its lean angle to stay in the same position. Further, the ballbot tells via its speakers, that it detected an object and how heavy it is. Ballbot then continues to drive back to a predefined location, as it can be seen in Figure 5.13d. This experiment was repeated four times to ensure the repeatability. Each time the ballbot was able to counterbalance the additional forces and report the weight of the object by ± 0.5 kg.

(a) Ballbot drives close to the test person (Time: 0:06)

(b) The test person places the object on ballbot's raised arms (Time: 0:17)



(c) Ballbot detected the object and reported the weight (Time: 0:23)

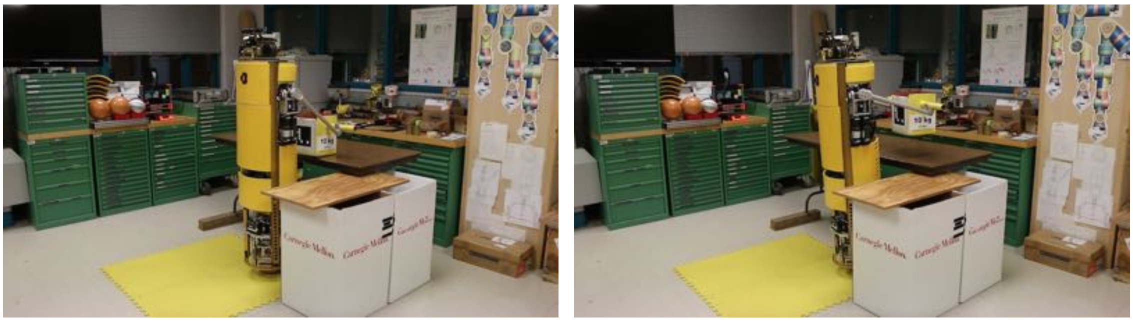
(d) Ballbot transports the object back to its home position (Time: 0:37)

Figure 5.13: Selected screen-shots of the video: Test person gives the ballbot a object to transport while balancing.

5.6.3 Transporting an object between two different locations

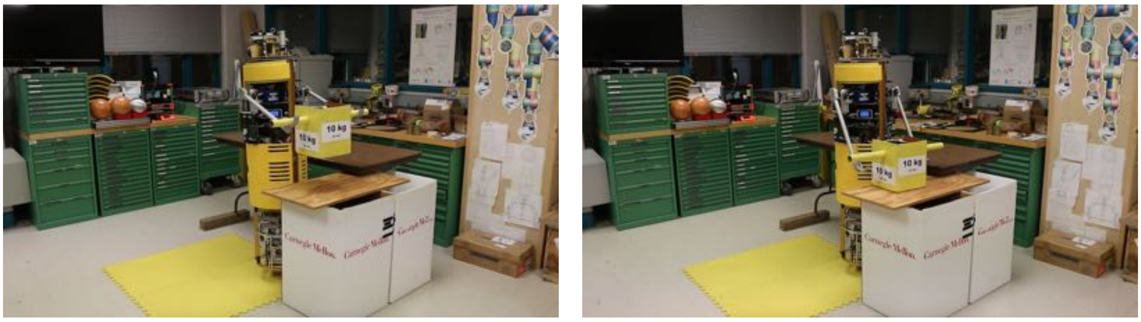
This experiment shall present the ballbots ability to semi-autonomously lift and transport an object between two different locations. Ballbot was positioned in front of the box prior to the start of the experiment. During the whole test, it was balancing gracefully. As it can be seen in Figure 5.14a the ballbot adjusted its arms slightly below the handles of the box and starts lifting the box with its 2 DOF arms. As soon as the target arm angles of 80° were reached, the ballbot started to yaw to its right side by 90°, as it can be seen in Figure 5.14b. The Figure 5.14c shows, that the ballbot yawed to the correct position while transporting the 10 kg heavy object. The final Figure 5.14d shows, that the end position, with an object successfully placed on the table, is reached 47 s after the lifting command was sent. This experiment was repeated five times in a row with similar results. A similar experiment without performing the yaw motion was as well repeated five times successfully.

A transport as explained in Chapter 4.2 was not repeatably achievable. It was not possible to reliably navigate to the object within the very tight margin of ± 4 cm, needed for the hands. Therefore, a successfully lift was not guaranteed due to the imperfect starting position.



(a) Ballbot is lifting the 10 kg heavy object (Time: 0:10)

(b) Ballbot started to yaw to its right (Time: 0:28)



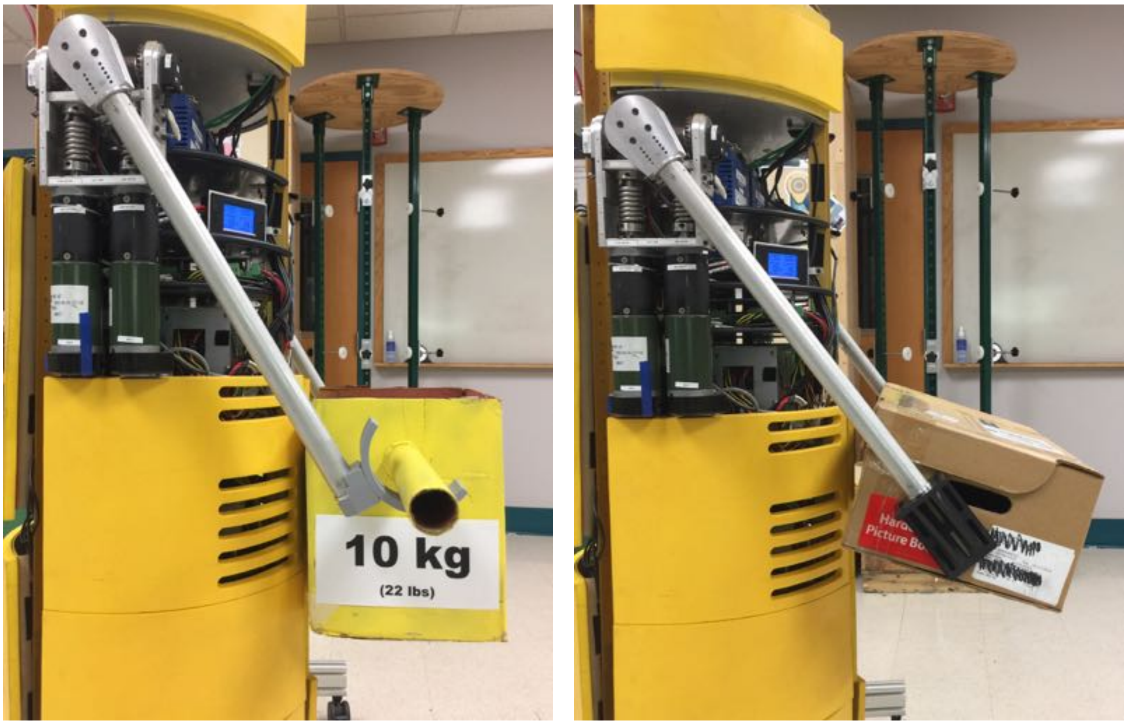
(c) Ballbot yawed to the target position (Time: 0:42)

(d) Ballbot sets the object down at its target location(Time: 0:57)

Figure 5.14: Selected screen-shots of the video: Transporting an object from A to B.

5.6.4 Lifting vs. Grabbing

As discussed in Chapter 4.4.2 there are two main motions for the 2-DOF arms. This experiment will show the benefits and issues of each motion. The first presented motion uses 3D printed hands with which the ballbot is able to lift objects with handles. Figure 5.15a shows the ballbot lifting a object with handles. The benefit of this lifting motion is, that even heavy objects can be transported very securely. The second motion, shown in Figure 5.15b, relies on the generated friction between the object and the arms, which are pushing against it. The higher the friction is, the heavier the objects can be transported. While Velcro^{TM} would provide more friction over a soft rubber surface, the rubber is preferred, as the objects do not need to be modified with Velcro^{TM} and experiments showed, that it is difficult to release the hands from the Velcro^{TM} surface after a successful transport. Due to the design of the 2 DOF arms only objects which are as wide as the ballbot can be lifted securely. As the comparison in Figure 5.15 shows, the object with handles can rotate freely. This allows the ballbot to bring it closer to its body compared to the rotated object in 5.15b. As the object with handles can be transported more securely and close to the body this box was used in all transporting experiments.



(a) Ballbot using its 3D printed hands to lift a object with handles.

(b) Ballbot using its flat hands to lift a object without handles.

Figure 5.15: Lifting vs. Grabbing

5.6.5 Heavy weight transport

This final experiment presents the maximum lifting and transport capability of the current setup of the ballbot. If there are tasks where higher limits would be needed, the Series Elastic Actuators in the arms need to be replaced. During this tests, the spring loads were closely monitored at all times to protect them from permanent deformation. Ballbot can transport higher loads when the arms are less extended. For this test, the arms were extended to 50°. At this angles the box touches the body slightly. The U-shape angle of the hands was adjusted as well to the maximum. This ensures that the box won't slip out of ballbot's hands. Figure 5.16 shows the setup of the arm and hands, as well as the body lean angle. The weight of the box was measured at 15 kg. This introduced a lean angle of 6.2° for the ballbot.

Even with that 15 kg heavy object the ballbot showed that it can keep its position stationary in station keep mode, yaw around its axis while keeping its position stationary and drive from point to point in outer loop control using the differential flatness path planning.

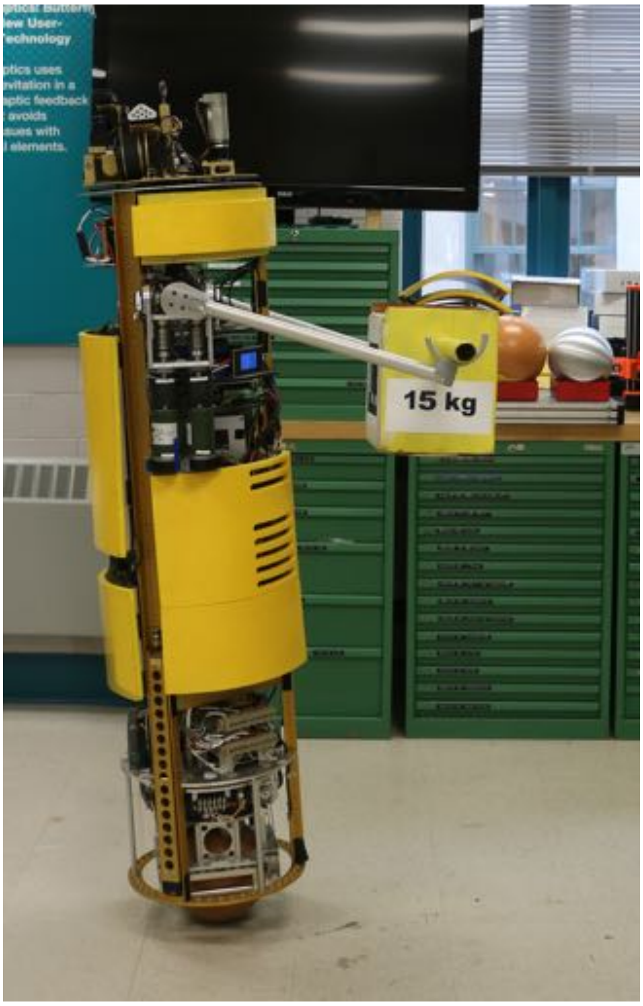


Figure 5.16: Maximum experimentally detected lean angle

6 Conclusion and outlook

This final chapter will conclude the work and results presented in this thesis. It will discuss which methods were used to reach the goal and provide insight into the future developments of the ballbot to enable a greater variety of interactive functions and abilities.

6.1 Conclusion

The main research question of this thesis was: How is a single wheeled, dynamically stable, mobile robot, such as the ballbot able to successfully lift and transport a heavy object? Answering this question should bring the ballbot from the from professional supervised robot, working in a lab environment, one step closer to a fully autonomous and self-sustained helper. It should be investigated how much weight the ballbot can successfully lift and transport while balancing gracefully. Additionally the ballbot needed to be able to detect an object and find its accurate location.

As seen in the previous chapter the ballbot is now able to detect and locate objects using fiducial markers using its onboard RGB camera and the ARUCO fiducial detection. Furthermore, the ballbot can autonomously transition from its statically stable position with all three legs down to a dynamically stable position while keeping its balance. As the lifting and carrying tasks showed, the ballbot was able to successfully lift and transport objects with varying weights. Additionally, it was shown, that the ballbot is able to collaborate with humans by delivering or receiving surprisingly heavy objects directly. The ballbot provides an audio feedback to help the interacting humans to understand the ballbots next movements and actions. Additionally the ballbot is able to adapt dynamically if the weight of the box changes. This ensures, that even if the boxes weight changes suddenly, the robot will keep its balance.

The one of biggest challenges to achieve these results was the correct weight estimation. The better the weight could be estimated, calculated or otherwise detected, the better the compensation could be performed. With this challenge in mind the following section will display the future work which needs to be done to improve ballbot's abilities. Additionally, the precision of the differential flatness path planer was a big challenge. Due to the narrow opening of the hands and the relatively short arms, the position in front of the box needs to be very precise.

6.2 Future Work

The ballbot is now able to detect, lift and transport different objects. As a household robot the ballbot lifting and transporting abilities need to cover a bigger variety of objects and situations. This final section will present which parts of the ballbot need to be improved to cover a wider range of possible tasks.

6.2.1 Lidar

The ballbot relies on a good lidar detection to locate itself in mapped environments. While the ballbot transports heavy objects the stationary mounted lidar tilts at the same angle as the body. This angle resulted in difficulties with a correct localization. As the ballbot shall be able to carry heavier object in the future, the lean angle will increase. Therefore a different solution for the localization needs to be implemented. A possible solution could either be the same lidar mounted on a gimbal to compensate the lean angle or a spherical lidar. Additionally, a visual localization and mapping algorithm could be added to improve the navigation.

6.2.2 Object detection

With the current camera setup, the ballbot was able to detect objects with fiducial markers at a distance of up to 3.5m. While this works well in a controlled environment, new object detection methods need to be added to prepare the ballbot for a real-life situation. A first step would be to increase the resolution of the camera, this would provide an extended range to detect smaller markers or marker at a greater distance. The second step would be to transition from a marker detection to an object detection solution where no additional things need to be added to the objects.

6.2.3 Lifting and path planning

Assuming the object detection is solved and the ballbot can detect the object's shape and location, the next step is to grasp and lift the object. The current 2-DOF arms are restricted to very basic movements. Additionally, the springs used in the Series Elastic Actuators limit the ballbot to objects below 15kg. The presented approaches to compensate external forces can be used for arms with more domains of freedom. By adding more domains of freedom a highly improved object manipulation is expected. Currently, a new pair of 7-DOF arms for the ballbot is in development and will be presented to the public soon. This set of new arms is expected to enable a new level of mobile manipulation on the dynamically stable system.

The results showed, that the current path planner tracks long trajectories very well, but lacks in the precision of exactly reaching the target location. As a precise position, prior lifting is essential, an additional more precise path planning method needs to be implemented. This needs to be done especially for small position adjustments near the object or target location.

By considering those results and conclusions, the planned extension of the CMU ballbot platform with 7 degree-of-freedom (7 DOF) arms will enable a wide range of possible jobs for ballbot robot types. Using more flexible arms and grippers increases the spectrum of transportable objects widely.

Bibliography

[1] U. Nagarajan, G. Kantor, and R. Hollis, “The ballbot: An omnidirectional balancing mobile robot,” *International Journal of Robotics Research*, vol. 33, no. 6, pp. 917–930, 2014.

[2] T. Lauwers, G. Kantor, and R. Hollis, “One is enough!” *Proc. Int’l. Symp. for Robotics Research*, vol. October, 2005.

[3] U. Nagarajan, “Fast and Graceful Balancing Mobile Robots,” MS Thesis, Carnegie Mellon University, 2012.

[4] U. Nagarajan, B. Kim, and R. Hollis, “Planning in high-dimensional shape space for a single-wheeled balancing mobile robot with arms,” *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 130–135, 2012.

[5] H. Arisumi, J. R. Chardonnet, A. Kheddar, and K. Yokoi, “Dynamic lifting motion of humanoid robots,” *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 2661–2667, 2007.

[6] B. Henze, M. A. Roa, and C. Ott, “Passivity-based whole-body balancing for torque-controlled humanoid robots in multi-contact scenarios,” *International Journal of Robotics Research*, vol. 35, no. 12, pp. 1522–1543, 2016.

[7] Boston Dynamics, “Atlas, The Next Generation,” 2016. [Online]. Available: https://youtu.be/rVlhMGQgDkY (Accessed 2018-06-01).

[8] M. Stilman, “Optimized Control Strategies for Wheeled Humanoids and Mobile Manipulators,” *IEEE International Conference on Humanoid Robotics*, vol. December, 2009.

[9] Georgia Tech, “Golem Krang, The strongest humanoid robot,” 2010. [Online]. Available: https://youtu.be/7TjKxjQOzVw (Accessed 2018-05-05).

[10] Boston Dynamics, “Introducing Handle,” 2017. [Online]. Available: https://youtu.be/-7xvqQeoA8c (Accessed 2018-06-01).

[11] Melonee Wise, M. Ferguson, D. King, E. Diehr, and D. Dymesich, “Fetch & Freight: Standard Platforms for Service Robot Applications,” *Workshop on Autonomous Mobile Service Robots, held at the 2016 International Joint Conference on Artificial Intelligence*, pp. 2–7, 2016.

[12] T. B. Lauwers, G. A. Kantor, and R. L. Hollis, “A dynamically stable single-wheeled mobile robot with inverse mouse-ball drive,” *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2006, no. 3, pp. 2884–2889, 2006.

[13] R. L. Hollis, “Ballbots,” *Scientific American*, vol. 295, no. 4, pp. 72–77, 2006.

[14] E. M. Schearer, “Modeling dynamics and exploring control of a single-wheeled dynamically stable mobile robot with arms,” MS Thesis, Carnegie Mellon University, 2006.

[15] M. Shomin, J. Forlizzi, and R. Hollis, “Sit-to-stand assistance with a balancing mobile robot,” *Proceedings - IEEE International Conference on Robotics and Automation*, no. June, pp. 3795–3800, 2015.

[16] B. Rohrer, S. Fasoli, H. I. Krebs, R. Hughes, B. Volpe, W. R. Frontera, J. Stein, and N. Hogan, “Movement smoothness changes during stroke recovery,” *The Journal of Neuroscience: The Official Journal of the Society for Neuroscience*, vol. 22, no. 18, pp. 8297–8304, 2002.

[17] U. Nagarajan, A. Mampetta, G. a. Kantor, and R. L. Hollis, “State transition, balancing, station keeping, and yaw control for a dynamically stable single spherical wheel mobile robot,” *2009 IEEE International Conference on Robotics and Automation*, vol. 2, pp. 998–1003, 2009.

[18] U. Nagarajan, G. Kantor, and R. L. Hollis, “Human-robot physical interaction with dynamically stable mobile robots,” *Proceedings of the 4th ACM/IEEE international conference on Human robot interaction - HRI ’09*, vol. March, pp. 281–282, 2009.

[19] U. Nagarajan, “Dynamic Constraint-based Optimal Shape Trajectory Planner for Shape-Accelerated Underactuated Balancing Systems.” *Robotics: Science and Systems*, 2010.

[20] U. Nagarajan, G. Kantor, and R. Hollis, “Hybrid control for navigation of shape-accelerated underactuated balancing systems,” *Proceedings of the IEEE Conference on Decision and Control*, pp. 3566–3571, 2010.

[21] U. Nagarajan and R. Hollis, “Shape space planner for shape-accelerated balancing mobile robots,” *International Journal of Robotics Research*, vol. 32, no. 11, pp. 1323–1341, 2013.

[22] B. Vaidya, M. Shomin, R. Hollis, and G. Kantor, “Operation of the ballbot on slopes and with center-of-mass offsets,” *Proceedings - IEEE International Conference on Robotics and Automation*, no. June, pp. 2383–2388, 2015.

[23] L. Havasi, "ERRO Sphere : an Equilibrator Robot," *Intl. Conf. on Control and Automation*, pp. 971–976, 2005.

[24] T. Endo and Y. Nakamura, "An omnidirectional vehicle on a basketball," *2005 International Conference on Advanced Robotics, ICAR '05, Proceedings*, pp. 573–578, 2005.

[25] M. Kumagai and T. Ochiai, "Development of a robot balancing on a ball," *International Conference on Control, Automation and Systems*, pp. 433–438, 2008.

[26] M. Kumaga and T. Ochiai, "Development of a robot balanced on a ball," *2009 IEEE International Conference on Robotics and Automation*, pp. 4106–4111, 2009.

[27] L. Hertig, D. Schindler, M. Bloesch, C. D. Remy, and R. Siegwart, "Unified state estimation for a ballbot," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 2471–2476, 2013.

[28] M. Neunert, F. Farshidian, and J. Buchli, "Adaptive Real-time Nonlinear Model Predictive Motion Control," *IROS 2014 Workshop on Machine Learning in Planning and Control of Robot Motion*, 2014.

[29] P. Asgari, P. Zarafshan, and S. A. A. Moosavian, "Dynamics modelling and stable motion control of a Ballbot equipped with a manipulator," *International Conference on Robotics and Mechatronics, ICRoM 2013*, no. February, pp. 259–264, 2013.

[30] ——, "Manipulation control of an armed Ballbot with stabilizer," *Proceedings of the Institution of Mechanical Engineers. Part I: Journal of Systems and Control Engineering*, vol. 229, no. 5, pp. 429–439, 2015.

[31] G. Pratt and M. Williamson, "Series Elastic Actuators–Pratt," *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, vol. 1, pp. 399–406, 1995.

[32] D. Robinson, J. Pratt, D. Paluska, and G. Pratt, "Series elastic actuator development for a biomimetic walking robot," *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pp. 561–568, 1999.

[33] W. Roozing, J. Malzahn, N. Kashiri, D. G. Caldwell, and N. G. Tsagarakis, "On the Stiffness Selection for Torque-Controlled Series-Elastic Actuators," *IEEE Robotics and Automation Letters*, vol. 2, no. 4, pp. 2255–2262, 2017.

[34] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Mg, "ROS: an open-source Robot Operating System," *Icra*, 2009.

[35] M. Shomin and R. Hollis, “Differentially flat trajectory generation for a dynamically stable mobile robot,” *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 4467–4472, 2013.

[36] U. Nagarajan, G. Kantor, and R. Hollis, “Trajectory Planning and Control of a Dynamically Stable Single Spherical Wheeled Mobile Robot,” *IEEE International Conference on Robotics Automation ICRA 2009*, pp. 3743 – 3748, 2009.

[37] ——, “Integrated planning and control for graceful navigation of shape-accelerated underactuated balancing mobile robots,” *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 136–141, 2012.

[38] S. Garrido-Jurado, R. Muñoz-Salinas, F. Madrid-Cuevas, and M. Marín-Jiménez, “Automatic generation and detection of highly reliable fiducial markers under occlusion,” *Pattern Recognition*, vol. 47, no. 6, pp. 2280–2292, 2014.

[39] F. J. Romero-Ramirez, R. Muñoz-Salinas, and R. Medina-Carnicer, “Speeded up detection of squared fiducial markers,” *Image and Vision Computing*, vol. 76, no. June, pp. 38–47, 2018.

[40] K. Kwak, J.-S. Kim, J. Min, and Y.-W. Park, “Unknown multiple object tracking using 2D lidar and video camera,” *Electronics Letters*, vol. 50, no. 8, pp. 600–602, 2014.

[41] K. Kwak, D. F. Huber, H. Badino, and T. Kanade, “Extrinsic calibration of a single line scanning lidar and a camera,” *IEEE International Conference on Intelligent Robots and Systems*, pp. 3283–3289, 2011.

[42] C. Fantacci, G. Vezzani, U. Pattacini, V. Tikhanoff, and L. Natale, “Markerless visual servoing on unknown objects for humanoid robot platforms,” *IEEE-RAS International Conference on Robotics and Automation*, 2018.

[43] G. Vezzani, U. Pattacini, and L. Natale, “A grasping approach based on superquadric models,” *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 1579–1586, 2017.

[44] S. M. Hsiang, G. E. Brogmus, and T. K. Courtney, “Low back pain (LBP) and lifting technique - A review,” *International Journal of Industrial Ergonomics*, vol. 19, no. 1, pp. 59–74, 1997.

[45] M. F. Antwi-Afari, H. Li, D. J. Edwards, E. A. Pärn, J. Seo, and A. Y. Wong, "Biomechanical analysis of risk factors for work-related musculoskeletal disorders during repetitive lifting task in construction workers," *Automation in Construction*, vol. 83, no. June, pp. 41–47, 2017.

[46] R. B. Graham, M. J. Agnew, and J. M. Stevenson, "Effectiveness of an on-body lifting aid at reducing low back physical demands during an automotive assembly task: Assessment of EMG response and user acceptability," *Applied Ergonomics*, vol. 40, no. 5, pp. 936–942, 2009.

[47] T. Bosch, J. van Eck, K. Knitel, and M. de Looze, "The effects of a passive exoskeleton on muscle activity, discomfort and endurance time in forward bending work," *Applied Ergonomics*, vol. 54, pp. 212–217, 2016.

[48] N. Matsunaga and Y. Okimura, "Kinematic Analysis of human lifting movement for biped robot control," *AMC 2004 - Kawasaki, Japan*, pp. 421–426, 2004.

[49] A. Sciutti, L. Patanè, F. Nori, and G. Sandini, "Understanding object weight from human and humanoid lifting actions," *IEEE Transactions on Autonomous Mental Development*, vol. 6, no. 2, pp. 80–92, 2014.

[50] Boston Dynamics, "Getting some air Atlas," 2018. [Online]. Available: https://youtu.be/vjSohj-Iclc (Accessed 2018-06-01).

[51] H. G. Nguyen, J. Morrell, K. D. Mullens, A. B. Burmeister, S. Miles, N. Farrington, K. M. Thomas, and D. W. Gage, "Segway robotic mobility platform," *SPIE Proc. 5609: Mobile Robots XVII*, no. October, 2004.

[52] R. Brooks, L. Aryananda, A. Edsinger, P. Fitzpatrick, C. C. Kemp, U. O'Reilley, E. Torres-Jara, P. Varshavskaya, and J. Weber, "Sensing and Manipulating Built-for-Human Environments," *International Journal of Humanoid Robotics*, vol. 01, no. 01, pp. 1–28, 2004.

[53] P. Deegan, B. J. Thibodeau, and R. a. Grupen, "Designing a Self-Stabilizing Robot For Dynamic Mobile Manipulation," *Proceedings of Robotics: Science and Systems Workshop on Manipulation for Human Environments*, 2006.

[54] M. Stilman, J. Olson, and W. Gloss, "Golem Krang: Dynamically stable humanoid robot for mobile manipulation," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 3304–3309, 2010.

[55] Istituto Italiano di Tecnologia - IIT, “The Centauro : A new disaster response robot to assist rescue workers to operate safely,” pp. 1–2, 2018. [Online]. Available: https://phys.org/news/2018-07-centauro-disaster-response-robot-workers.html (Accessed 2018-08-01).

[56] ——, “The Centauro,” 2018. [Online]. Available: https://www.youtube.com/watch?v=L7JssknlCvw (Accessed 2018-08-01).

[57] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, “Convolutional pose machines,” in *CVPR*, 2016.

[58] A. Mampetta, “Automatic transition of ballbot from statically stable state to dynamically stable state,” MS Thesis, Carnegie Mellon University, 2006.

[59] Planetspring, “Torsion Spring Calculator,” 2018. [Online]. Available: https://www.planetspring.com/pages/torsion-spring-calculator-torsion-spring-calculation.php?id=torsion (Accessed 2018-05-04).

Appendices

A Inertia measurement

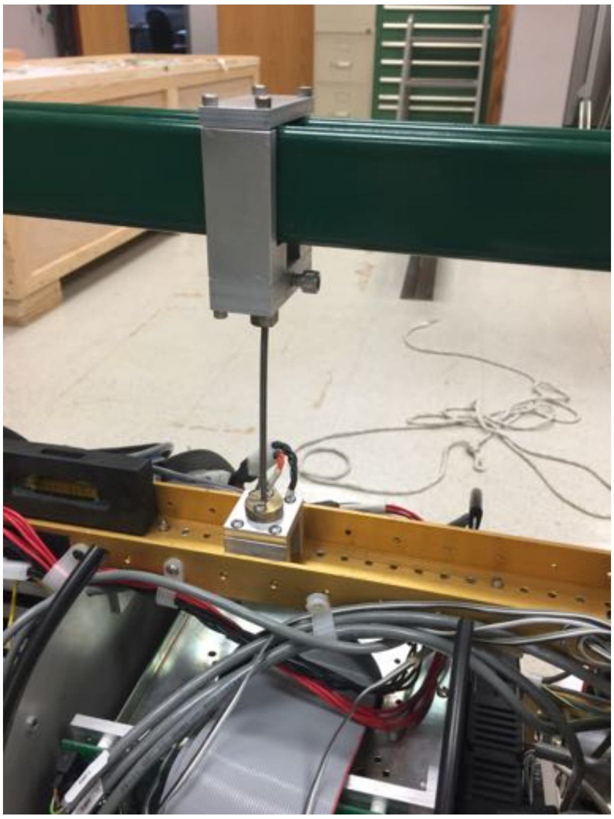


Figure 1: Ballbot mounted to the inertia measurement frame at its center of gravity

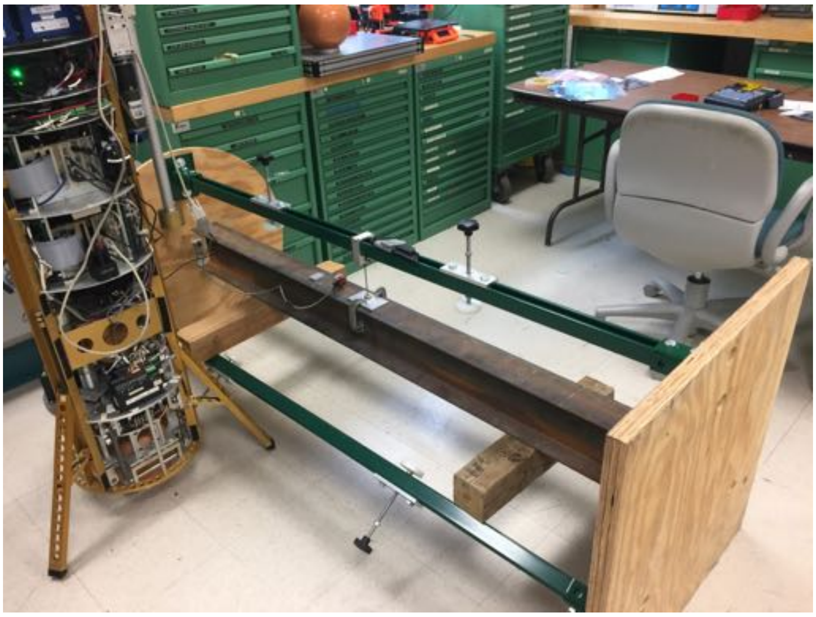


Figure 2: I-Beam mounted to the inertia measurement frame at its center of gravity

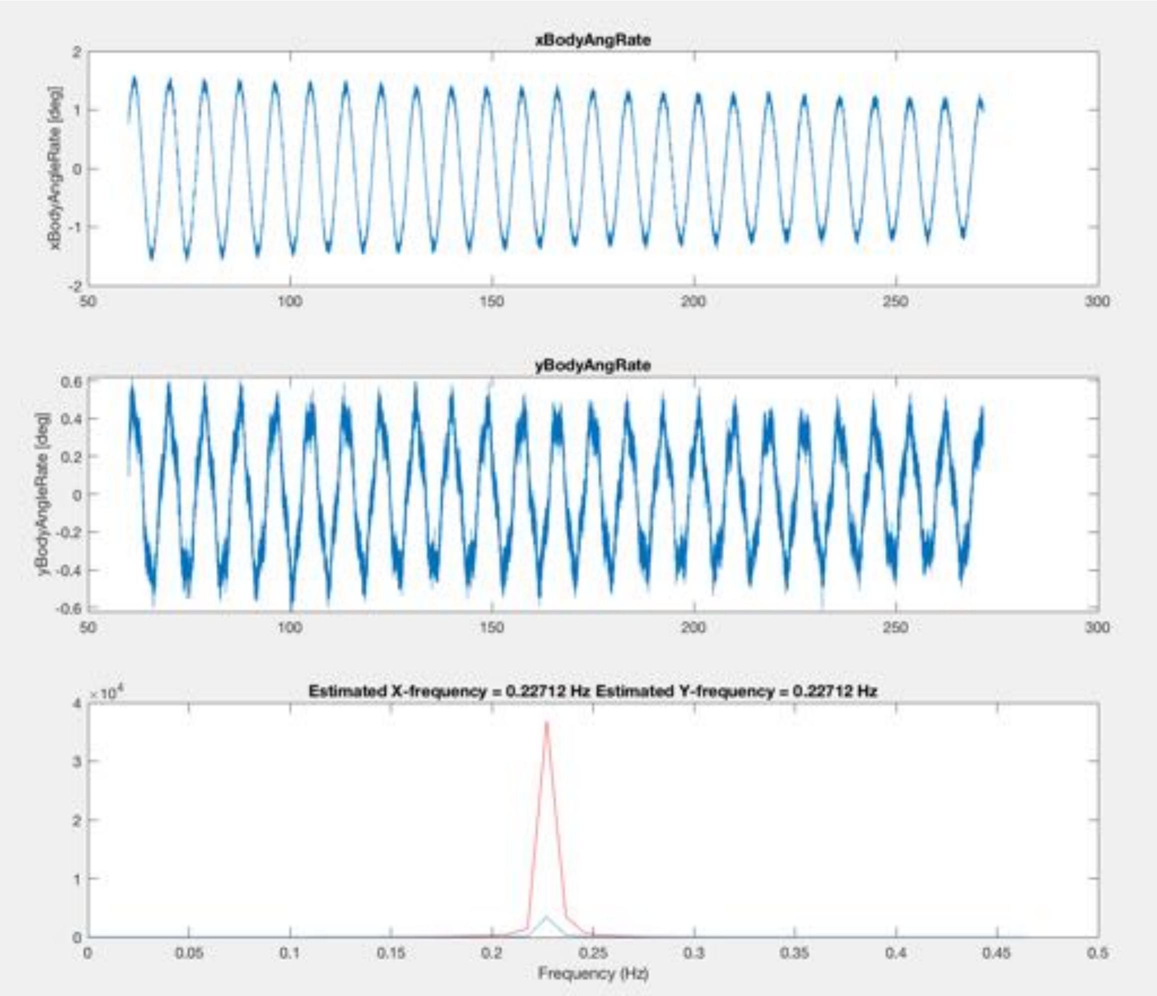
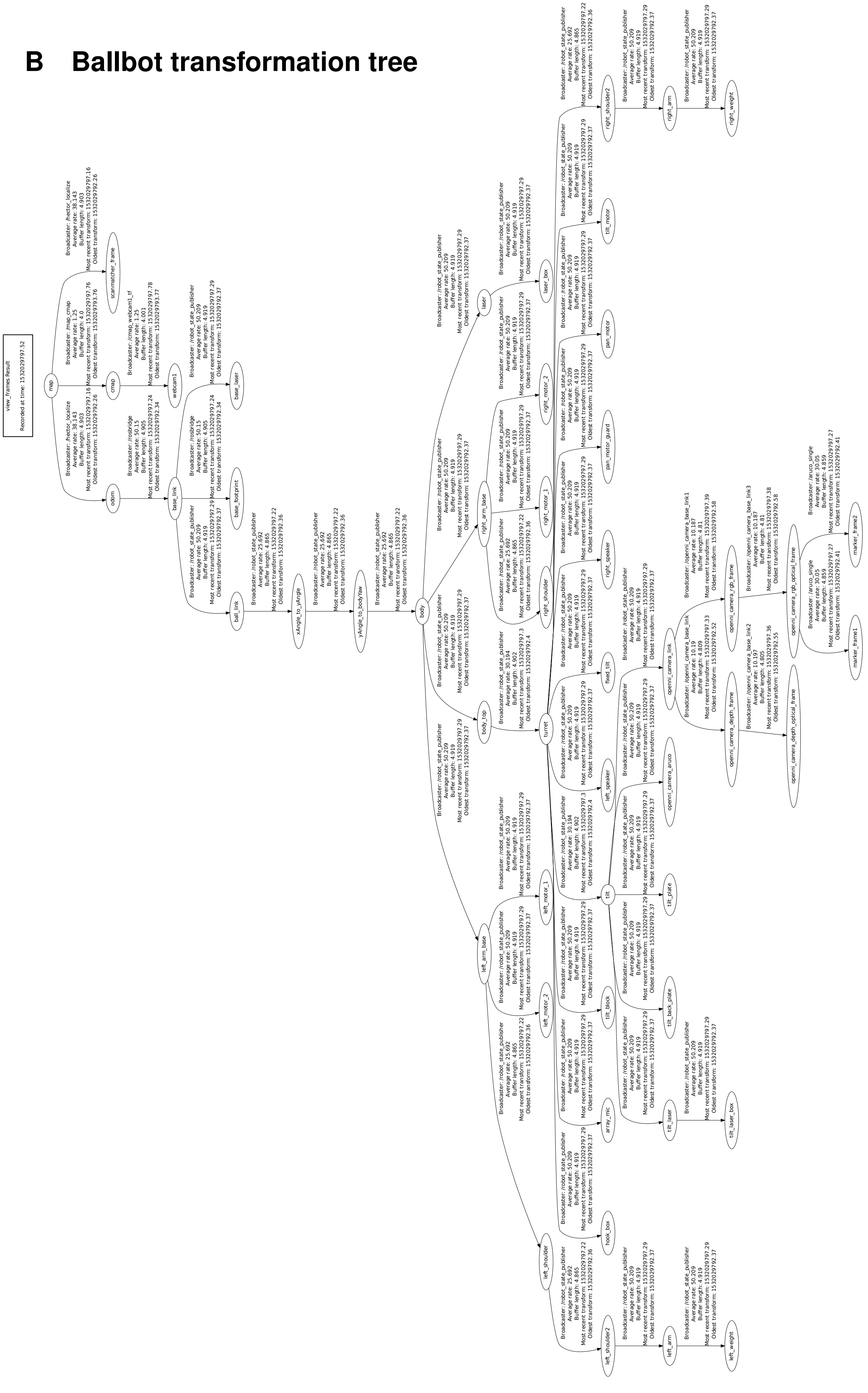


Figure 3: Ballbots angular acceleration and the average frequency

B Ballbot transformation tree


view_frames Result
Recorded at time: 1532029797.52
map
odom
cmap
scanmatcher_frame
webcam1
base_link
base_footprint
base_laser
ball_link
xAngle_to_yAngle
yAngle_to_bodyYaw
body
laser
laser_box
right_arm_base
right_motor_1
right_motor_2
right_shoulder
right_shoulder2
right_arm
right_weight
body_top
turret
tilt_motor
pan_motor
pan_motor_guard
right_speaker
fixed_tilt
left_speaker
tilt
openni_camera_link
openni_camera_aruco
openni_camera_rgb_frame
openni_camera_depth_frame
openni_camera_rgb_optical_frame
openni_camera_depth_optical_frame
marker_frame1
marker_frame2
tilt_plate
tilt_block
tilt_back_plate
tilt_laser
tilt_laser_box
array_mic
hook_box
left_arm_base
left_motor_1
left_motor_2
left_shoulder
left_shoulder2
left_arm
left_weight