Eindhoven University of Technology

MASTER

Control of a hexapodal robot

Willems, J.J.P.A.

Award date:
2011

Link to publication

Disclaimer
This document contains a student thesis (bachelor's or master's), as authored by a student at Eindhoven University of Technology. Student theses are made available in the TU/e repository upon obtaining the required degree. The grade received is not published on the document as presented in the repository. The required complexity or quality of research of student theses may vary by program, and the required minimum study period may vary in duration.

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain

Control of a hexapodal robot

J.J.P.A. Willems

CST 2011.089

Master's thesis

Coach(es):	ir. R.J.M. Janssen dr. ir. M.J.G. van de Molengraft
Supervisor:	prof. dr. ir. M. Steinbuch
Committee:	ir. R.J.M. Janssen dr. I. Gonzalez Alonso dr. ir. M.J.G. van de Molengraft prof. dr. ir. M. Steinbuch

Eindhoven University of Technology
Department of Mechanical Engineering
Control Systems Technology

Eindhoven, November, 2011

Abstract

The Dynamical Systems Design department at the Eindhoven University of Eindhoven is actively researching the design and control of robotic systems. Examples are Amigo, the service robot competing in the RoboCup @home league, the mid-size league soccer robots of Tech United and the Tulip which is the humanoid robot competing in the RoboCup humanoid adult league. One of the new research activities is the hexapodal robot, a legged robot which eventually should participate in the RoboCup Middle Size League. Therefore the final goal in the development of the robot should be to make it run.

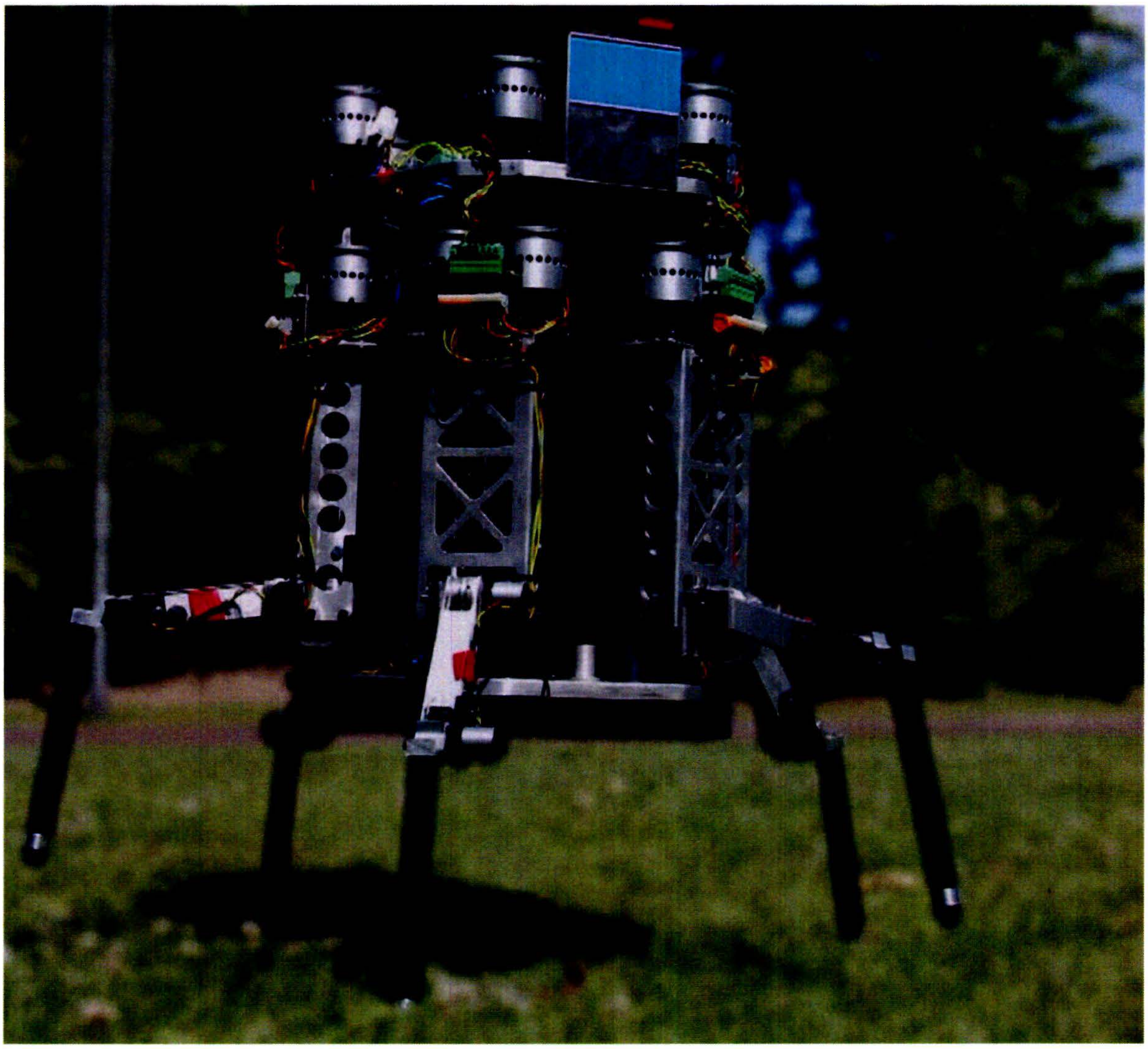
A first prototype is developed that is used as a testbed. The hexapod has 3 degrees-of-freedom per leg where each rotational joint is separately actuated so that the robot has 18 actuators. The torque of the actuator is transmitted to the joint using a torsional spring transmission. The introduced compliance improves the impact safety of the hardware and can decrease energy consumption when correctly exploited. The hexapod is equipped with an inertial measurement unit and each joint and actuator has a position encoder.

A software structure is developed to control the motion of the hexapod. The control is developed in open-source software, mainly in Orocos. To control the hexapod the controller is divided in two main functions. The first controls the leg in joint space when it is positioned in the air by using SISO control techniques. The second controls the body by using linear virtual actuators in combination with the joint space controllers. The resulting forces of the virtual actuators determine the torques on the joints using a force distribution algorithm.

The results of the overall performance as well as of individual components are presented. Although some improvements can still be made, the robot is able to walk. Using more advanced control techniques like learning could improve the walking speed. Exploiting the compliant dynamics and a redesign of the hardware eventually should result in a running hexapod.

Contents

Eindhoven University of Technology Walking Robot



Chapter 1

Introduction

1.1 Motivation and background

For several years walking robots are an active research topic in academic areas [16]. The difference between a legged robot and a wheeled robot is the method of locomotion. A legged robot has several advantages over a wheeled robot. An overview of these advantages is depicted in Fig. 1.1. A legged robot has a better mobility, it can easier bypass obstacles and it can maintain its orientation while moving over irregular terrain. Walking does not have the disadvantage of slippage when moving over soft terrain. The environmental damage is also minimized because it does not leave tracks. On an irregular terrain the speed of a walking robot is superior with respect to a wheeled robot. Walking robots have a lot of possibilities, they can for example be used for the inspection of power plants, exploration of an unknown type of terrain and many more applications [55].

Since 2005 Eindhoven University of Technology is successfully participating in the RoboCup Middle Size League (MSL) [1] with Tech United Eindhoven [2]. Tech United won the unofficial European Championship 3 times and was 4 times in a row runner-up at the World Championship. Since 2011 Tech United expanded its team to compete in the @home league with the service robot called Amigo. In the MSL the research is focused on full autonomy and cooperation at plan and perception level. Therefore robots are used to play autonomous soccer in teams of five. The official goal of RoboCup is
By mid-21st century, a team of fully autonomous humanoid robot soccer players shall win the soccer game, comply with the official rule of the FIFA, against the winner of the most recent World Cup.
At the moment the MSL soccer field is a flat carpet such that wheeled robots can be used. Taking the main goal of RoboCup in mind, the MSL should switch to a real grass field in the (near) future. To respond to this foreseeing a legged robot, a hexapod, is developed within the Control System Technology department at the faculty of Mechanical Engineering.

1.2 Problem statement and objective

The development and control of a walking robot is a complex challenge with a broad spectrum of topics. The legs of a walking robot are connected to one another through the body and also through the ground, forming closed kinematic chains. Forces and moments propagate through the kinematic chain from one leg to another, and therefore dynamic coupling exists. With respect to these issues, there are several aspects in this project that need to be investigated. These aspects

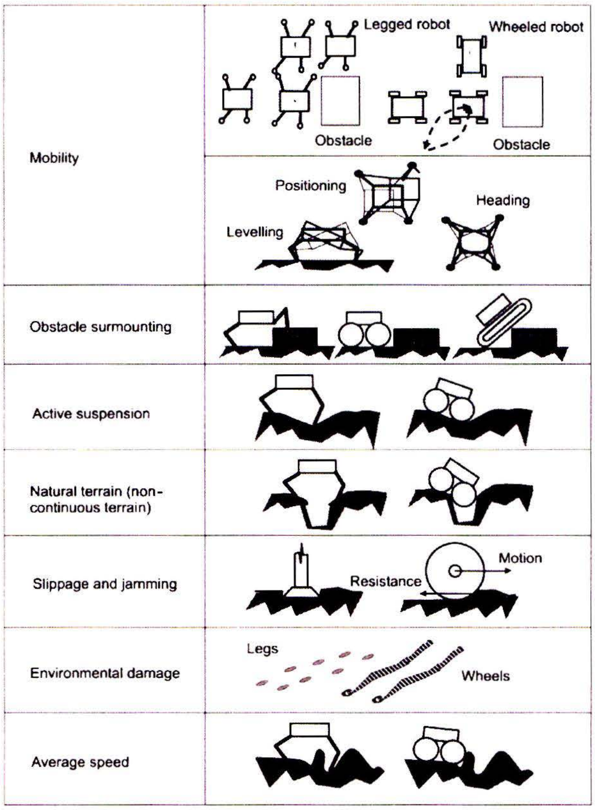


Figure 1.1: Advantages of legged vs wheeled robots [55].

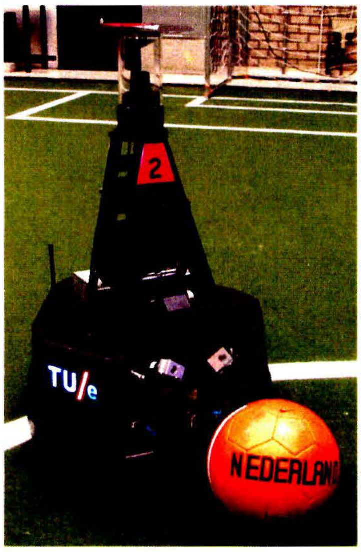


Figure 1.2: TURTLE, Tech United Eindhoven Middle-Size league robot.

are

- the design of a software architecture for the robot,
- the design of a walking pattern, i.e. a gait,
- the control of the legs,
- the control of the body.

Therefore the objective of the work described in this report is

Design and implement a software structure that controls the hexapodal robot in real-time such that it is able to walk.

To fulfill this objective, first the necessary drivers need to be developed to communicate with the hardware. A software design has to be developed, that translates the general objective into practical implementations. In this implementation the encoders of the joints and actuators and the orientation of the hexapod have to be processed. A controller for the legs and body has to be developed and needs to be implemented with a walking procedure. Finally the hexapod has to be tested to validate its ability to walk.

1.3 Outline

In this report first a brief overview of literature about walking robots is given. The hardware design will be discussed in Chapter 3. The modeling of the hexapodal robot will be presented in Chapter 4. In Chapter 5 the used gait will be given. The software architecture, the used software and components, will be discussed in Chapter 6. In Chapter 7 the controller architecture and results will be discussed. In the final chapter the conclusions and recommendations are presented.

Chapter 2

Literature survey

Due to the variety of research topics on walking robots a brief literature survey is performed to get familiar with the possibilities. Research topics that apply are for instance gait generation, force distribution and leg control. Of course there is a lot of overlap between these topics.

2.1 Gait generation and control

Definition 1. *A gait is defined by the time and the location of the placing and lifting of each foot, coordinated with the motion of the body in its six degrees of freedom, in order to move the body from one place to another.*

as it is defined by Song and Waldron [16]. At the moment the research on gaits can be divided in two classes [16] namely (i) periodic and (ii) non-periodic gaits. The periodic gaits can be divided in continuous and discontinuous gaits. The main difference between these gaits is that in the continuous gait the legs and body are constantly moving. With the discontinuous gait the motion is sequential such that the legs are transferred and when all legs are positioned the body is moved forward. The non-periodic walking is often used for terrain negotiating such that it is able to walk on a highly irregular environment.

Periodic gaits The common used gait in the animal world is the periodic-continuous gait. Different type of gaits exist depending on the number of legs. Common gaits for quadrupeds are bounding, trot, pace, gallop. The common gaits for a hexapod are the tripod gait and wave gait.

Non-periodic gaits Non-periodic gaits, also called free gaits, are used on highly irregular terrains. Mostly the algorithms for free gaits use logic and check the stability per step and which leg to move next [33, 16]. Currently a lot of research to non-periodic gaits is on The Little Dog robot [36]. The robot often is used at academic universities for research on rough-terrain locomotion [28, 49]. In the case of Little Dog visual information is used to determine the positioning of the feet.

2.1.1 Control

Due to the large diversity of control structures this subsection only will give a brief overview of the possibilities with respect to control of quadruped and hexapod robots. Most of the studies focus on the combination of control and coordination of the legs because these are closely related.

Central Pattern Generators

The most used method to generate periodic-continuous gaits for multi-legged locomotion is using a Central Pattern Generator (CPG). CPGs are neuronal circuits which can produce rhythmic motor patterns as walking, in the absence of sensory inputs [32]. CPGs are inspired on the animal world, these CPGs are located in the spinal cord in vertebrate animals. Many researchers use the principle of CPGs by using coupled oscillators. In order to adapt to the terrain, sensory feedback is needed [34]. Research on CPG also focuses on adapting the CPG to increase the walking speed or adapt to differences in surface texture. Adaptive gait patterns are created using CPGs including a motor model in [54]. In [46] a passive compliant quadruped robot is used in combination with central pattern generators. The generated trajectories are followed by PID controllers in the RC servo motors. In [17] robust locomotion is obtained using cooperative oscillators. A more extensive survey about CPGs in general can be found in [22].

Force distribution

A walking robot consists of a body with a mass which has to be distributed over several legs. Several researchers investigated the force distribution problem which they applied on different robot configurations. This force distribution problem is applied on a quadruped [59, 30, 18], a hexapod [31], a octopod [57]. The force and moments on the body are related to the forces acting on the tips of the legs or manipulators which are in contact with a surface. Constraints are added such that no slip should occur and the torque of the actuators are limited. Because the solution of the inverse dynamics is not unique an optimization is used. For this optimization, an objective function is defined where most often the required torques and thus the energy consumption is minimized. As stated in [37] for optimization quadratic programming is preferred in these type of cases.

Virtual actuator control

A topic closely related to force distribution is the virtual actuator control introduced by Pratt [41]. Virtual actuator control uses linear virtual components like springs and dampers as the actuator. These components exert a virtual force and moment as function of the error on the body and legs, when not in contact with the ground. Using Pratt's Force Distribution Method [41], an extension of Gardner's Force Distribution Method, the forces acting on the body are partitioned to force contributions of each leg. Using a state machine the controllers are switched between different predefined states. The controller structure is tested in simulation on a walking hexapodal robot [56]. The hexapod was able to walk in different directions. The virtual actuator control is also applied on a bipedal walking robot [40]. In [19] adaptive virtual control is used on a bipedal robot to enhance the robustness of the control.

Control architectures

In cases where a walking robot has to deal with an irregular terrain, more advanced architectures are used. In [15] the walking robot SILO6, used for humanitarian-demining uses 4 levels of control. The basic control level uses a PID controller at each joint. The second level is the reactive control which anticipates on the environment by using the joint positions and foot forces. The third is the deliberative control level which plans the robot motion. The last level, the supervisor control level, selects the type of gait and generates the robot trajectory, i.e. the navigation. In [13] and [24] a control architecture is used for the quadruped robot, Little Dog. The robot is able to walk over a highly irregular terrain. The control used is joint PD control, force P control and inverse dynamics. The planner optimizes the pose of the robot depending on the footholds. Furthermore learning is applied to learn to position the feet on the terrain.

2.2 Compliant walking robots

Nature inspired engineers to use compliant legs because of impact safety. Another advantage of compliant legs is the reduction of energy consumption when the dynamics are exploited correctly. A large diversity of legged robots currently exists, monopods, bipeds, quadrupeds, hexapods and many other configurations, A few are depicted in Fig. 2.2 Marc Raibert, one of the leading experts on walking robots, built one-, two- and four-legged hydraulically actuated robots based on prismatic compliant legs [43, 44]. Using compliant legs, complex dynamical behavior can be achieved by relatively simple control algorithms. First a one-legged robot with a compliant leg was controlled. To use the same algorithms on biped and quadruped robots, Raibert introduced the concept that multiple legs are acting as one leg, i.e. virtual legs are used [43]. A well-known compliant robot is RHEX [47, 26]. RHEX is designed using the principle of a Spring-Loaded Invertible Pendulum (SLIP). SLIP represents locomotion as a mass attached to a spring as depicted in Fig. 2.1. Exploiting the dynamics results in running and hopping robots. SCOUT and PAW [52], two robots from McGill University, use the same principle of SLIP but with a different design. They are designed in simplicity, they exist out of a body with four compliant prismatic legs and have actuated hips. This design limits the walking behavior such that it is only able to bound. Puppy II [20, 21] is a small quadruped robot developed to investigate body dynamics. The robot is actuated at the hip joints and has passive compliant knee joints. The motors are controlled by a simple oscillatory position control. Using this design, Puppy II can achieve a forward speed of 7 leg-length per second. A similar design is used for the Cheetah [46], using a CPG without sensory feedback a forward velocity of 25 centimeter or one body length per second is achieved. As can be noticed that the compliant robots are mainly quadrupeds. Furthermore the compliance is incorporated as a linear spring in the last link.

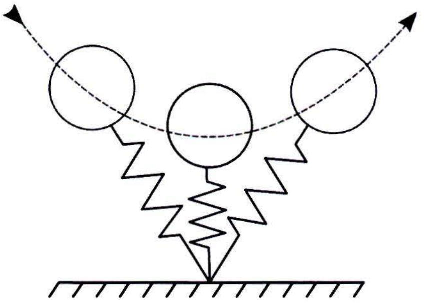
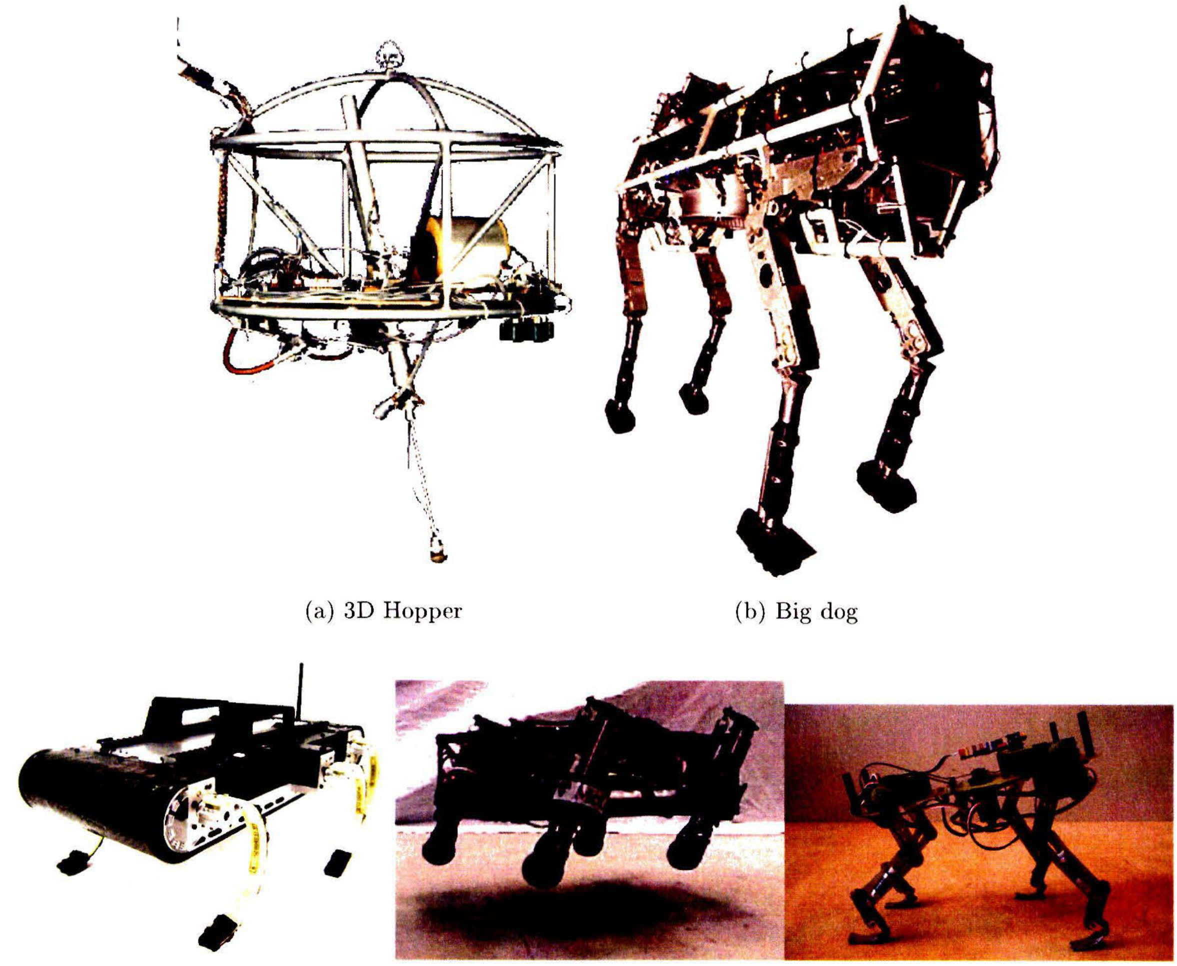


Figure 2.1: Spring-loaded Invertible Pendulum.



(a) 3D Hopper

(b) Big dog

(c) RHEX

(d) PAW

(e) Cheetah

Figure 2.2: Compliant walking robots.

Chapter 3

Hardware design

This chapter describes the first hardware design of Tech United Eindhoven hexapod, a photo is depicted in Fig. 3.1. The hexapod is eventually meant for RoboCup soccer, therefore it is designed with respect to the RoboCup Middle Size League regulations [8]. The following rules are applicable for the design

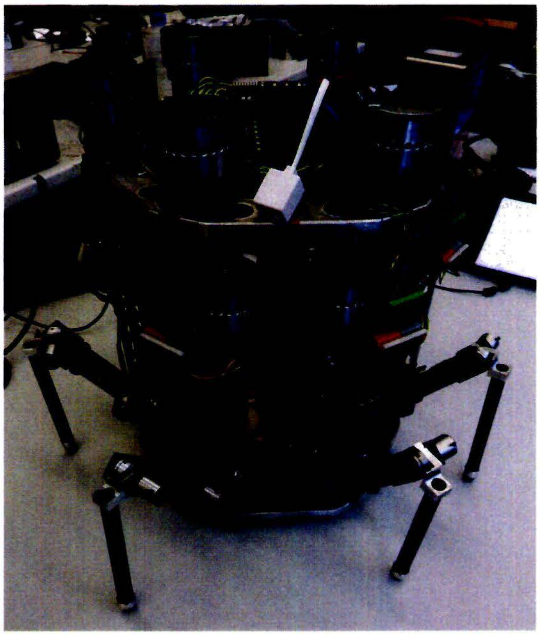


Figure 3.1: Tech United Eindhoven hexapod.

- The projection of the robot's configuration, including its actuators, must fit into a square of at least 30cm × 30cm and at most 52cm × 52cm.
- The robot's height must be at least 40 cm and at most 80 cm.
- The maximum allowable weight of a robot is 40 kg.

The robot consists of a body with six modular legs in a radial symmetrical arrangement. The radial symmetrical arrangement increases the stability with respect to rectangular hexapods, allows them to have a better turning ability and a greater stride length in certain conditions as researched in [14]. The advantage of this configuration is the flexibility of movement direction. During a soccer match the ball is in continuous movement so the robot should be able to alter its direction on the fly.

The hexapod has a mass of 9.48 kg without the (1 kg) battery. The robot measures a diameter of 30 cm, 43 cm in height when the legs are fully retracted and 60 cm when the legs are fully extended. The legs are designed to have three degrees of freedom as can be seen in Fig.4.2b, this is inspired on the leg of an insect. The leg has a weight of 0.9 kg. Each joint is driven by a separate Maxon EC 45 flat motor. The position of the motor shaft is measured through an integrated Hall sensor. The motors are connected to a flexible transmission with GS 38 A Maxon gear heads with a reduction of 60 : 1. The flexible transmission is a rod which acts as a torsional spring, the actuators are positioned at the body and its force transmission to the joints is achieved through these torsional springs. For the second and third joint additional steel cables are used for the transmission. Therefore the system can be regarded as a series-elastic actuator (SEA) [39]. The position of the joints are measured with absolute encoders with a resolution of 4096 counts per revolution.

This design of the legs has several advantages (i) the torsional spring can store and dissipate energy from impact and (ii) the gravitational force acting on the legs can be neglected.

The rotation of the robot is measured with the CHR-6d IMU/Inclinometer [45]. The IMU outputs 3-axis roll rates, 3-axis acceleration and the pitch and roll angles.

For the real-time data acquisition two Beckhoff [10] EtherCAT [23] FB1111-0142 piggyback controller boards are used. The first module is connected to the IMU and the battery. The second is connected to the joint encoders and actuators.

The EtherCAT board can be connected to the on-board PC. However for testing purposes the stacks are wired to an off-board desktop PC. The robot is powered by 8 lithium-ion polymer cells. A schematic overview of the dataflow in the robot can be seen in Fig. 3.2

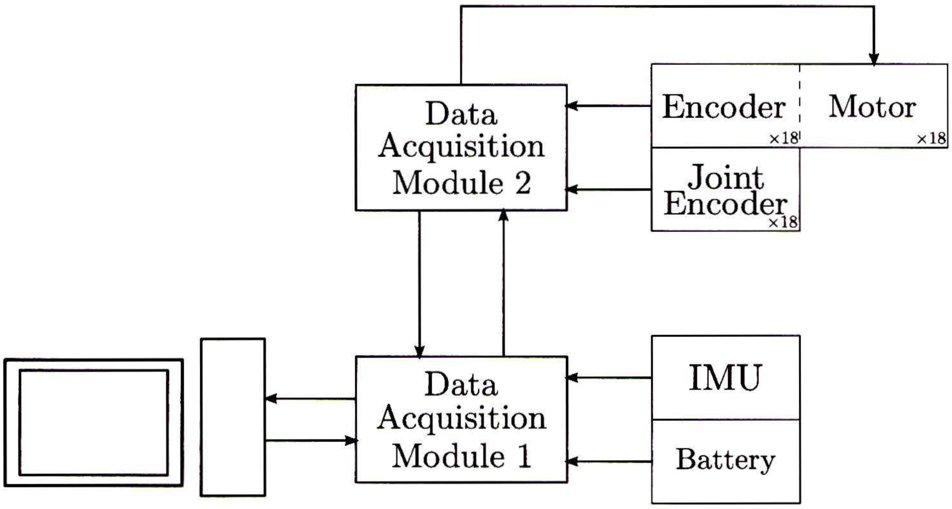


Figure 3.2: Hardware components and dataflow.

Hexapodal speed To compete in the RoboCup MSL the speed of the hexapod is of great importance. The Tech United TURTLEs can achieve a speed of 3.5 m/s.
The actuators have a no load speed of 4300 rpm, after the gearbox with a reduction of $60 : 1$ this results in a rotational speed of 1.2 rotations per second.
A reasonable step size is 0.15 meter, which is equal to a stroke of 45°. A full step consists of two strokes so 90° will take 0.2 seconds, not taking accelerations and decelerations into account. This can result in a speed of 0.75 m/s. However this is under ideal conditions and thus will not be achieved in practice. A solution would be to use legs with different periods and stepping sizes.
It is interesting to note that in nature a gait transition occurs from walking to running when the Froude number is equal to 0.4 [48]. The Froude number is calculated as

$$Fr = \frac{v^2}{gl} \tag{3.1}$$

v denotes the velocity, g the gravitational constant and l the length of the leg.
This implies for the hexapod that at approximately 0.9 m/s this transition occurs.

Chapter 4

System modeling

This chapter describes the system modeling of the hexapodal robot. First the kinematics of the leg will be explained. Then the velocity kinematics and dynamics of the leg will be discussed. The last section will discuss the body dynamics.

4.1 Leg kinematics

When modeling in robotics, generally two spaces can be distinguished namely the operational space and the joint space.

- The operational space, induced by the Cartesian coordinates of the tip of the leg with respect to the center of the body:

$$\boldsymbol{x}_k^{\mathcal{B}} = \boldsymbol{x}_k = \begin{bmatrix} x_k & y_k & z_k \end{bmatrix}^T \tag{4.1}$$

- The joint space, induced by the joint angles:

$$\boldsymbol{q}_k = \begin{bmatrix} \theta_{1,k} & \theta_{2,k} & \vartheta_{3,k} \end{bmatrix}^T \tag{4.2}$$

It should be noted that a distinguishment is made between the kinematic $\theta_{3,i}$ and dynamic $\vartheta_{3,i}$ angle where the dynamic angle is defined as

$$\vartheta_{3,k} = \theta_{2,k} + \theta_{3,k} \tag{4.3}$$

Using this distinguishment the third joint is decoupled from the second joint because the third joint is remotely driven using a steel cable, a figure visualizing this distinction is depicted in Fig. 4.1. The leg consists of three links as can be seen in Fig 4.2a. Using the Denavit-Hartenberg [51, 55] frame assignment, the frames for one leg are obtained as in Fig. 4.2b. The position and orientation of coordinate frame i with respect to the previous coordinate frame $(i-1)$ can be specified by the homogeneous transformation matrix $\mathbf{A}$

$$\mathbf{A}_i^{i-1}(q_i) = \begin{bmatrix} \mathbf{R}_i^{i-1}(q_i) & \boldsymbol{o}_i^{i-1}(q_i) \\ \mathbf{0}^T & 1 \end{bmatrix} \tag{4.4}$$

where $\mathbf{R}_i^{i-1} \in SO(3)$ is a rotation matrix, and the vector $\boldsymbol{o}_i^{i-1} \in \mathbb{R}^3$ denotes the position coordinates from frame i with respect to frame $i-1$. The hexapod legs are identical open kinematic

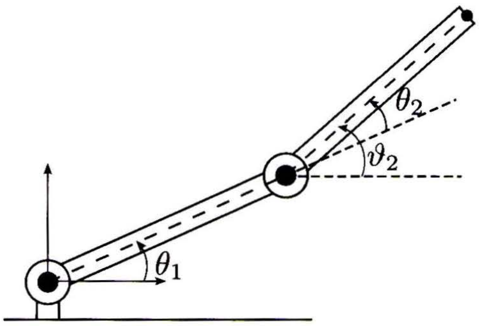
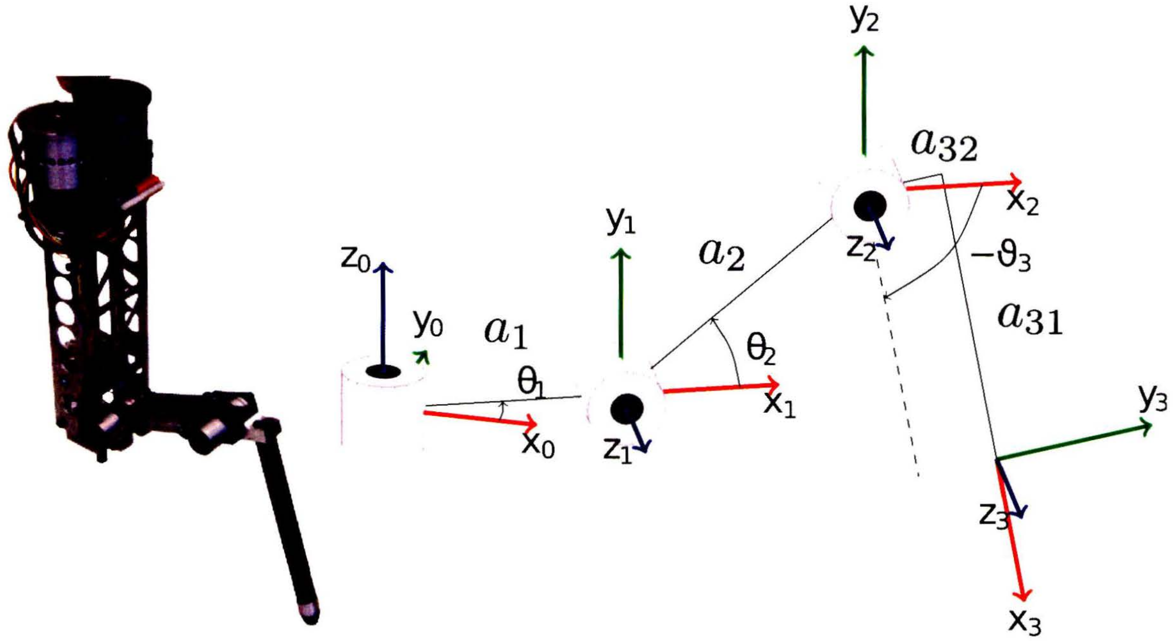


Figure 4.1: Schematic overview of the difference of a geometric θ_2 and dynamic ϑ_2 angle.



(a) Photo leg

(b) Schematic leg

Figure 4.2: Schematic overview of the hexapod leg

chains, normally the forward kinematics can be computed as a product of the homogeneous transformations between the circumjacent coordinate frames. However the last joint of the leg is remotely driven and therefore the homogeneous transformation for a single leg is obtained as

$$\mathbf{T}_3^0(\boldsymbol{q}) = \underbrace{\begin{bmatrix} \mathbf{R}_1^0 & \boldsymbol{o}_2^0 \\ \mathbf{0}^T & 1 \end{bmatrix}}_{\mathbf{A}_2^{0*}} \mathbf{A}_3^2 \tag{4.5}$$

It can be seen that the rotation ($\mathbf{R}_1^0$) of $\mathbf{A}_2^{0*}$ does not depend on θ_2. The derivation of the homogeneous transformation matrices can be found in Appendix A.

The obtained forward kinematics model from the root of the leg to the tip is

$$o_3^0(\boldsymbol{q}) = \begin{bmatrix} a_1\cos(\theta_1) + a_2\cos(\theta_1)\cos(\theta_2) + \cos(\theta_1)\left(a_{31}\cos(\vartheta_3) - a_{32}\sin(\vartheta_3)\right) \\ a_1\sin(\theta_1) + a_2\sin(\theta_1)\cos(\theta_2) + \sin(\theta_1)\left(a_{31}\cos(\vartheta_3) - a_{32}\sin(\vartheta_3)\right) \\ a_2\sin(\theta_2) + a_{32}\cos(\vartheta_3) + a_{31}\sin(\vartheta_3) \end{bmatrix} \tag{4.6}$$

With the use of this, positions of the robot tip in operational space can be calculated from robot configurations in joint space. For example $\boldsymbol{q}_0 = \begin{bmatrix}0 & 0 & 0\end{bmatrix}^T$ corresponds with the configuration when the leg is fully stretched. The corresponding position of the tip of the leg with respect to o_0

$$o_0^3(\boldsymbol{q}_0) = \begin{bmatrix}a_1 + a_2 + a_{31} & 0 & a_{32}\end{bmatrix}^T \tag{4.7}$$

The solution for the inverse kinematics can be found in Appendix A.2.

4.2 Leg velocity kinematics

The velocity kinematic relationship of a serial chain is determined by the Jacobian which is

$$\boldsymbol{\xi} = \mathbf{J}(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{4.8}$$

where

$$\boldsymbol{\xi} = \begin{bmatrix}\boldsymbol{v}_3^0 \\ \boldsymbol{\omega}_3^0\end{bmatrix} \quad \text{and} \quad \mathbf{J} = \begin{bmatrix}\mathbf{J}_v \\ \mathbf{J}_\omega\end{bmatrix} \tag{4.9}$$

$\boldsymbol{v}_3^0$ denotes the linear velocity vector of the tip and $\boldsymbol{\omega}_3^0$ the angular velocity vector. Both are defined with respect to the root of the leg.
The linear velocity of the end effector, i.e. the foot, is $\dot{o}_0^3$ so

$$\mathbf{J}_{v_i} = \frac{\partial o_0^3}{\partial q_i} \tag{4.10}$$

For the angular velocity $\mathbf{J}_\omega$ can be determined by

$$\mathbf{J}_\omega = \begin{bmatrix}\rho_1\mathbf{R}_0^0 k & \rho_2\mathbf{R}_1^0 k & \rho_3\mathbf{R}_2^0 k\end{bmatrix} \tag{4.11}$$

ρ_i is 1 for a revolute joint and 0 for a prismatic joint and $k = \begin{bmatrix}0 & 0 & 1\end{bmatrix}^T$. However in this case, due to the remote driven link, the second column is $\mathbf{0}_{3\times 1}$ because the third link does not depend on the rotation of the second link.
The complete Jacobian is

$$J = \begin{bmatrix} -a_1S_1 - S_1(a_{31}C_3 - a_{32}S_3) - a_2C_2S_1 & -a_2C_1S_2 & -C_1(a_{32}C_3 + a_{31}S_3) \\ a_1C_1 + C_1(a_{31}C_3 - a_{32}S_3) + a_2C_1C_2 & -a_2S_1S_2 & -S_1(a_{32}C_3 + a_{31}S_3) \\ 0 & a_2C_2 & a_{31}C_3 - a_{32}S_3 \\ 0 & 0 & S_1 \\ 0 & 0 & -C_1 \\ 1 & 0 & 0 \end{bmatrix} \tag{4.12}$$

where $\sin(\theta_i) = S_i,\ \cos(\theta_i) = C_i$ except for S_3 and C_3, here ϑ_i is used instead of θ_i.

4.3 Leg dynamics

The dynamics of the leg can be distinguished into two part, (i) the actuator dynamics and (ii) the load dynamics. The complete derivation of the dynamics can be found in Chapter B.

4.3.1 Actuator and transmission dynamics

The transmission of one degree-of-freedom of the leg can be modeled as a mass-spring model as depicted in Fig. 4.3. By applying Newton's law, the equation of motion for the actuator and transmission can be described by

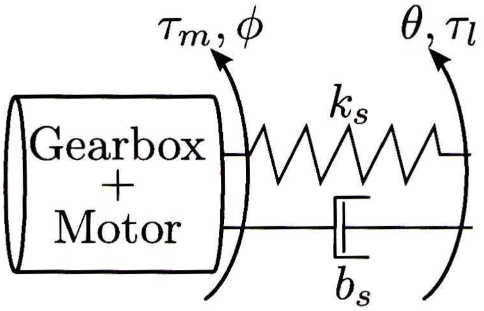


Figure 4.3: Series elastic actuated drive train.

$$J_m\ddot{\phi} - b_s(\dot{\theta} - \dot{\phi}) - \tau_s = \tau_m \tag{4.13}$$

where τ_s is the torque exerted by the torsional springs, which is assumed to be a linear function

$$\tau_s = k_s(\phi, \theta) = k_s(\theta - \phi) \tag{4.14}$$

4.3.2 Mechanical dynamics

The equation for the joints can be obtained as

$$\mathbf{M}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \mathbf{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + \boldsymbol{g}(\boldsymbol{\theta}) + \boldsymbol{\tau}_s = -\mathbf{J}^T\boldsymbol{f} \tag{4.15}$$

$\mathbf{M}$ is the inertia matrix, $\mathbf{C}$ contains the centrifugal and Coriolis terms and $\boldsymbol{g}$ is the gravity vector. When the leg is in the air there is no ground contact thus $\boldsymbol{f} = \mathbf{0}$. However when the leg touches the ground (4.15) becomes undetermined. In this case the equation should be solved in one of the following ways [16]:

- Using Lagrange multipliers to minimize some energy function.
- Modeling foot/terrain interaction.
- Using force sensors to measure f.

A systematic derivation of the Euler-Langrange equations can be found in B. In this case a simplification is used which regards the load as a mass such that the system is a 2-mass-spring-damper model. This results in the following equation of motion

$$J_l\ddot{\theta} + b_s(\dot{\theta} - \dot{\phi}) + \tau_s = 0 \tag{4.16}$$

4.4 Body dynamics

Since there is no relative motion between the hips and the body, it can be considered that the exerted force acts at mass center of the object [29, 37]. The force and moment vector, resulting from gravity and the external force acting on the hexapod, are shown in Fig. 4.4. For simplicity, only the forces on the foot are presented and the torques are neglected.

$\boldsymbol{F} = \begin{bmatrix} F_x & F_y & F_z \end{bmatrix}^T \in \mathbb{R}^3$ and $\boldsymbol{M} = \begin{bmatrix} M_x & M_y & M_z \end{bmatrix}^T \in \mathbb{R}^3$ denote the robot body force vector and moment vector respectively. The force/moment quasi-static equilibrium equation of the robot can be written as [25, 31, 57]

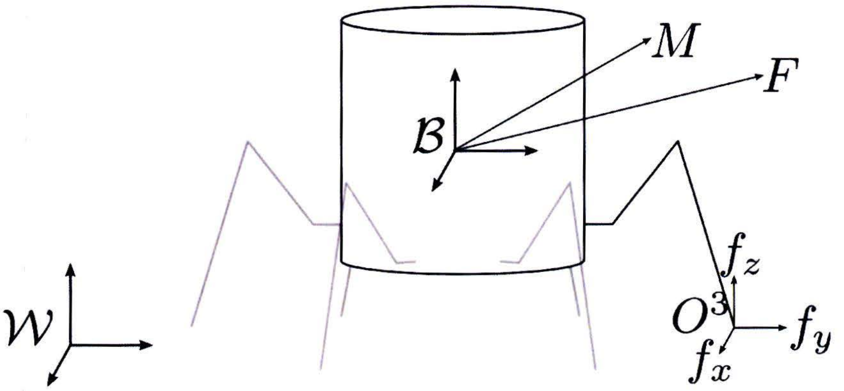


Figure 4.4: Schematic overview of the forces acting on the hexapod.

$$\boldsymbol{F} = \sum_{k=1}^{p} \boldsymbol{f}_k \tag{4.17}$$

$$\boldsymbol{M} = \sum_{k=1}^{p} \boldsymbol{x}_k \times \boldsymbol{f}_k \tag{4.18}$$

Where p is the number of legs on the floor, $\boldsymbol{f}_k$ is the force acting on the tip of leg k and $\boldsymbol{x}_k$ is the position of the tip. Both equations can be rewritten to matrix form as

$$\boldsymbol{F} = \mathbf{J_F}\boldsymbol{f} \tag{4.19}$$

$$= \begin{bmatrix} \mathbf{I}_3 & \dots & \mathbf{I}_3 \end{bmatrix} \begin{bmatrix} \boldsymbol{f}_1 \\ \vdots \\ \boldsymbol{f}_p \end{bmatrix} \tag{4.20}$$

where $\mathbf{J_F} \in \mathbb{R}^{3\times 3p}$ and $\boldsymbol{f} \in \mathbb{R}^{3p}$. The moment vector can be rewritten as

$$\boldsymbol{M} = \mathbf{J_M}\boldsymbol{f} \tag{4.21}$$

$$= \begin{bmatrix} 0 & -x_{z,1} & x_{y,1} & \dots & 0 & -x_{z,p} & x_{y,p} \\ x_{z,1} & 0 & -x_{x,1} & \dots & x_{z,p} & 0 & -x_{x,p} \\ -x_{y,1} & x_{x,1} & 0 & \dots & -x_{y,p} & x_{x,p} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{f}_1 \\ \vdots \\ \boldsymbol{f}_p \end{bmatrix} \tag{4.22}$$

where $\mathbf{J_M} \in \mathbb{R}^{3\times 3p}$. By combining Eqs. (4.20) and (4.22) we obtain

$$\mathbf{J_{FM}}\boldsymbol{f} = \mathbf{W} \tag{4.23}$$

with

$$\mathbf{J_{FM}} = \left[\mathbf{J_F}^T \quad \mathbf{J_M}^T\right]^T \in \mathbb{R}^{6\times 3p}$$
$$\boldsymbol{W} = \left[\mathbf{F}^T \quad \mathbf{M}^T\right]^T \in \mathbb{R}^{6}$$

$\boldsymbol{W}$ denotes the wrench. To determine the requested forces on the tip of the legs $\mathbf{J_{FM}}$ has to brought to the right handside using the pseudo-inverse

$$\boldsymbol{f} = \mathbf{J_{FM}}^{+}\mathbf{W} \tag{4.24}$$

The system is under-determined and the solution of (4.24) is not unique. A solution can be to use optimization techniques with additional constraints to solve the actuator redundancy [38, 37]. The torques on the joints can then be determined by using the transpose of the Jacobian as

$$\boldsymbol{\tau}_k = \mathbf{J}^T\boldsymbol{f}_k \tag{4.25}$$

However because the wrench acting on the body is determined with respect to the world, the force has to transformed to the body using $\mathbf{R}_B^W$ such that the equation becomes

$$\boldsymbol{\tau}_k = \mathbf{J}^T\mathbf{R}_{0,k}^B\mathbf{R}_B^W\boldsymbol{f}_k \tag{4.26}$$

The rotation $\mathbf{R}_{0,k}^B$ transforms the force vector from the body frame to the frame of the root of leg k. This is done because the Jacobian is defined with respect to this frame.

Chapter 5

Walking algorithm

5.1 Gait

A continuous gait is the most used gait by animals. A gait can be characterized by several parameters [16] namely

- the cycle time, t_c, also called the period, is the time it takes for a leg to perform a full step,
- the duty factor of leg k, β_k, is the fraction of the cycle time a leg is on the ground,
- the phase of leg k, Φ_k, is the normalized phase lag with respect to zero lag,
- the leg stroke R, is the distance which a foot is moved relative to the body. In combination with the cycle time this will determine the speed of the robot.

The duty factor and phase determine the type of gait. Several type of gaits exist for an insect such as the tripod gait, which is known as the fastest insect gait. In the tripod gait three legs, the front, back and middle of the other side, support the body and the other three are in the flight phase [11]. The slowest gait is the wave gait at which one leg moves at a time.

5.2 Leg trajectory

The trajectory used is generated for simplicity. It is assumed that every leg will perform the same step in a specified direction. Therefore the trajectory of a leg is generated in a 2D fashion as shown in Fig 5.1. Using a transformation the trajectories of the legs can be altered to move in any direction. The trajectories are generated with respect to the frame of the body ($\mathcal{B}$), this is done by using an offset of the transformed trajectory.

The trajectory of the leg is generated using a sinusoid and a linear function, depending on the state of the leg i.e. lifted in the air or standing on the ground. The result of the generation of this trajectory is that it is not continuous differentiable. However this discontinuity occurs at the transfer between the states thus an impact is expected. The trajectory is created as following, the leg index is omitted for readability

1. The time within the trajectory is determined because the phase per leg can vary.

$$\tilde{t} = \Delta t + (1 - \Phi)t_c \tag{5.1}$$

$$\tilde{t} = \begin{cases} \tilde{t} & \text{if } \tilde{t} \leq t_c \\ \tilde{t} - t_c & \text{if } \tilde{t} > t_c \end{cases} \tag{5.2}$$

2. Then the 2D position of the tip of the leg is determined depending on the state, i.e. is it in the support or flight phase

$$\boldsymbol{\zeta}_x = \begin{cases} \frac{R\tilde{t}}{t_c(1-\beta)} - \frac{R}{2} & \text{if } \tilde{t} \leq t_c(1-\beta) \\ \zeta_h \sin\left(\frac{\pi\tilde{t}}{t_c(1-\beta)}\right) + \zeta_z & \text{else} \end{cases} \tag{5.3}$$

$$\boldsymbol{\zeta}_z = \begin{cases} \frac{R}{2}\cos\left(\frac{\pi(\tilde{t}-t_c(1-\beta))}{\beta t_c}\right) & \text{if } \tilde{t} \leq t_c(1-\beta) \\ \zeta_z & \text{else} \end{cases} \tag{5.4}$$

The trajectory is depicted in left figure of Fig. 5.1. ζ_h denotes the height of the step and ζ_z denotes the offset with respect to the body frame in z direction

3. The x-position $\boldsymbol{\zeta}_x$ is transformed to a x and y position with respect to the body

$$\boldsymbol{x}_x = \boldsymbol{\zeta}_x \cos(\zeta_\gamma) + \zeta_r \cos\left((k-1)\frac{\pi}{3}\right) \tag{5.5}$$

$$\boldsymbol{x}_y = \boldsymbol{\zeta}_x \sin(\zeta_\gamma) + \zeta_r \sin\left((k-1)\frac{\pi}{3}\right) \tag{5.6}$$

$$\boldsymbol{x}_z = \boldsymbol{\zeta}_z \tag{5.7}$$

ζ_γ denotes the angle the trajectory has to be turned around the z-axis. This defines the direction of walking. ζ_r denotes the radius at which the trajectories will be positioned.

This results in the real-time trajectory which can be seen in 5.1. In the left part the 2D-trajectory is shown and in the right the transformed trajectories are shown. The trajectory for turning the robot around its z-axis is an altered version of the continuous gait. The main difference is that the turn angle and turning direction of the leg is specified.

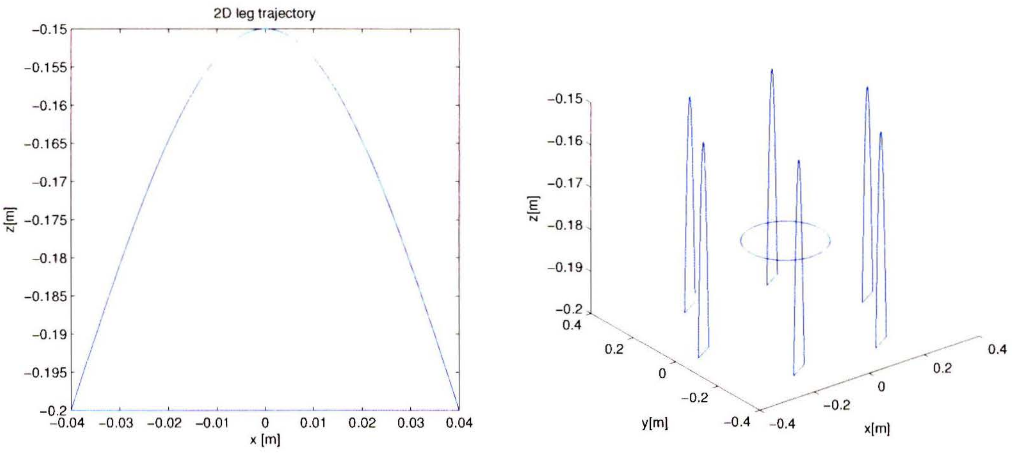


Figure 5.1: Trajectory of the gaits.

It should be noted that no communication or cooperation exist between the trajectories of the legs.

Chapter 6

Software architecture

This chapter describes the software architecture of the hexapod robot. All software used in this project is free and open-source. For the operating system Ubuntu 10.04 LTS was used.

6.1 ROS

The Robot Operating System (ROS) [6] is an open-source framework which provides a structured communication layer [42]. ROS is designed with the following goals in mind: (i) peer-to-peer, (ii) tools-based, (iii) multi-lingual, (iv) thin and (v) free and open-source. The advantage is that at the moment a large community exists and is collaborating on packages or tools for ROS. Due to the communication network these tools can be (re)used by users as requested. In our case several tools are used which are incorporated in ROS, the following will be briefly explained RVIZ, Gazebo and Orocos-RTT.

Orocos-RTT Orocos-RTT stands for the Open RObot COntrol Software - Real-Time Toolkit [53]. Orocos is a C++ framework for constructing complex component based robotic systems. Orocos is developed with the following goals in mind [12]: (i) open source, (ii) modular and flexible, (iii) independent and (iv) component based. This design has the advantage that several computations can be divided over several components keeping the software complexity low. All Orocos components are made configurable such that parameters easily can be altered without editing the source-files, and thus no need to re-compile the code. Orocos is currently used as a soft real-time implementation. A hard real-time implementation could be used by using a real-time operating system.
Furthermore an integration with EtherCAT exists using a wrapper for SOEM, This implies that only a driver needs to be written to get EtherCAT working.

RVIZ RVIZ is a 3D visualization environment for robots using ROS. The joint positions and IMU orientation are published out of Orocos to a ROS-topic. RVIZ is subscribed to this topic such that it is able to visualize the data. If data is recorded it can be used as a tool to replay the data giving a better insight of the state of the hexapod. RVIZ obtains the kinematic structure out of a URDF (Unified Robot Description Format) [7]. URDF is an XML format for representing robot models. This is done by defining the kinematic chains. It is also possible to assign dynamic properties like masses and inertias to the model such that it can be used for dynamic simulation purposes. In Fig. 6.1 the visualizer is shown in the right, in the left the simulation is depicted. Obviously the visualizer also can be used for the real robot.

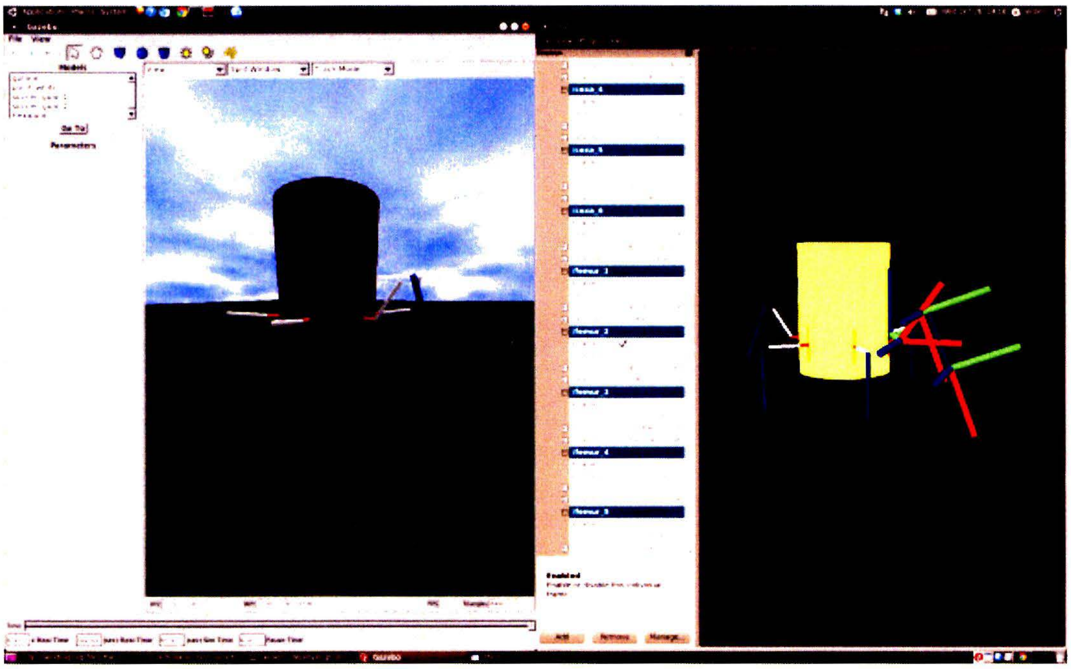


Figure 6.1: Simulator and visualizer of the hexapod.

Gazebo Gazebo [27] is a 3D rigid body simulation package for robots, part of the Player project [3]. Gazebo uses the Open Dynamics Engine (ODE) [5] to simulate physics. ODE consists out of two main components i.e. a rigid-body dynamics and a collision detection engine. Gazebo is developed such that it trades off accuracy for speed. The rendering engine used within Gazebo is Ogre3D [4] which is an Object-Oriented Graphics Rendering Engine.
To communicate with the Orocos layer, Gazebo publishes data to topics which are read by an interface component. This interface component makes the data available for the Orocos components. The interface also reads the torques output of the controller, then the interface publishes the torques to a topic such that they can be used in Gazebo.
As a torsional spring was not present as a transmission this is developed by the ROS-community on request. To verify the transmission a setup of a 2-mass-spring model was designed. The frequency response were measured which satisfied the expected result, a description can be found in Chapter C.

6.2 Implementation

Several Orocos components are written to control the hexapod. The components are structured such that it can be regarded as a standard feedback-control structure as depicted in Fig. 6.2. This structure has the advantage that it is well-known and modular. Therefore several control structures can be easily tested. The components are also designed such that they are configurable and thus parameters can easily be altered. In Fig. 6.2 dashed arrows indicate that the reference generator and controller also use additional information from the hexapod. The component diagram of the hexapod depicted in Fig. 6.3. The communication between the components will be explained in the following subsections.

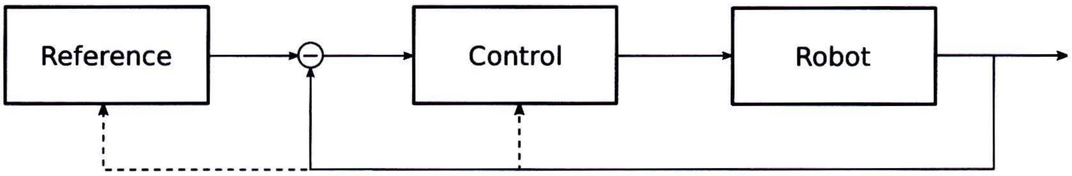


Figure 6.2: Feedback-control scheme.

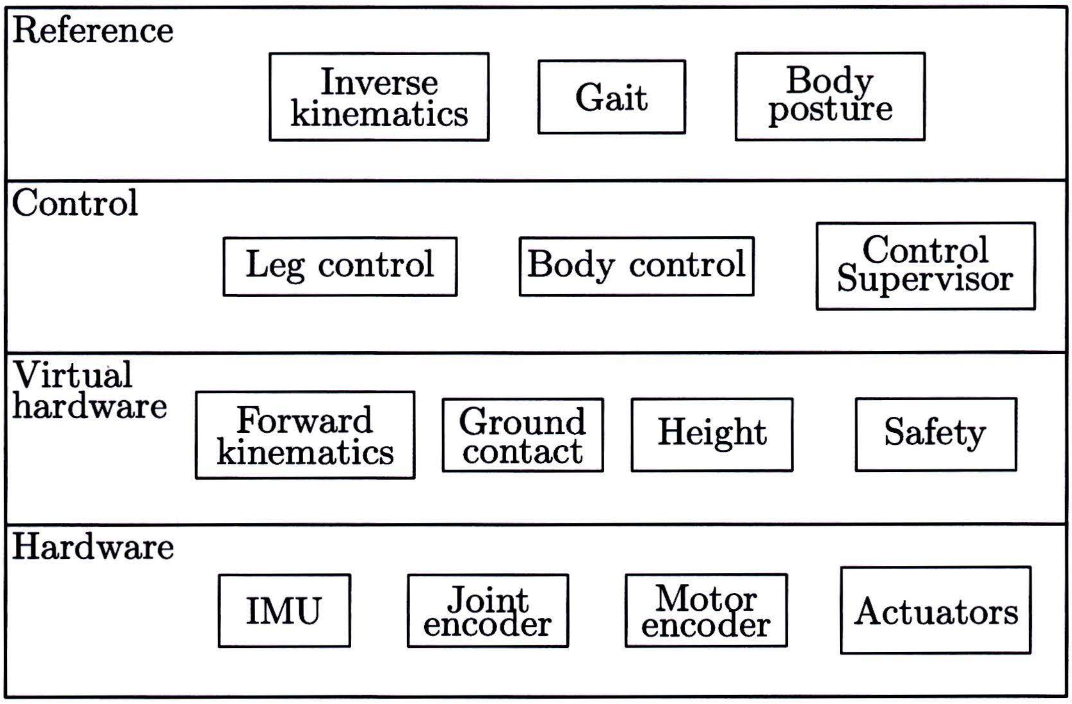


Figure 6.3: Component diagram of the hexapod.

6.2.1 Reference

The top layer as depicted in Fig. 6.3 is the reference layer. The reference layer as in Fig. 6.4 generates the trajectories for the legs. Using inverse kinematics of KDL the joint references are obtained. Besides these outputs the reference layer also generates a reference for the hexapod body in terms of the height, roll and pitch (h, α, β).

Gait

As can be seen in Fig. 6.4 the gait consists out of three components. These component generate the reference for the tip of the leg.

Position This component retrieves the current tip position of the leg and generates a linear trajectory to the requested position within a given duration.

Continuous gait The continuous gait generates a reference for the tip of the leg as described in Chapter 5. Currently two continuous gaits exist, one for walking in a specified direction and one for turning around the hexapod's z-axis. The gait can be altered by editing the configuration parameters.

Inverse kinematics

The inverse kinematics determines the joint angles of the leg from a given position of the tip. The inverse kinematics are using the Kinematics and Dynamics Library (KDL) of the Orocos Project. The implemented algorithm is based on Newton-Raphson iterations. Furthermore joint limits into account and the number of iterations and precision is configurable. To seed the algorithm the previous solution of the inverse kinematics is used to decrease the number of iterations needed.

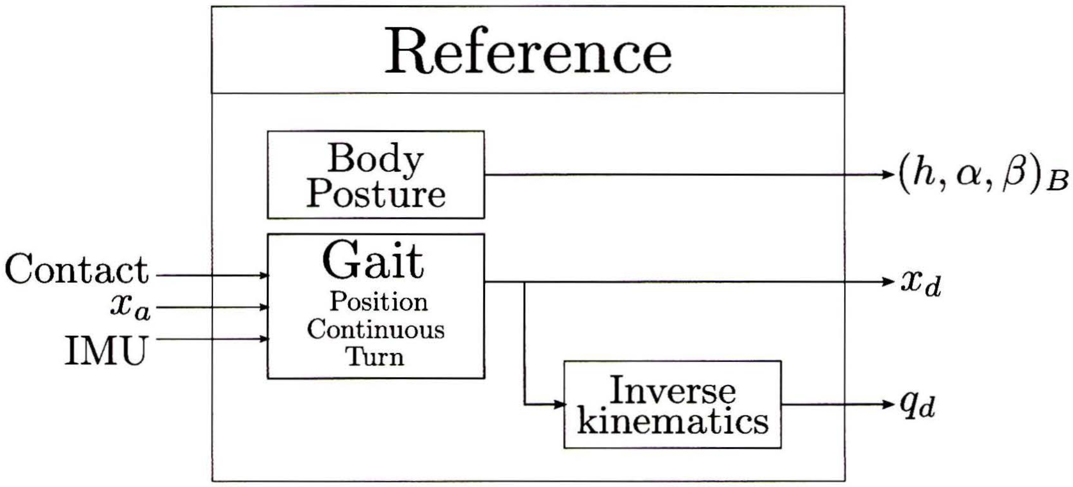


Figure 6.4: Reference structure of the robot.

6.2.2 Control

The second layer is the control layer. The overview of the control structure is depicted in Fig. 6.5. The theory behind the controller part is extensively described in Chapter 7.

Controller components

The controller components developed within the RoboEarth [58] project are used, the description of the components can be found in [35]. The controller components used are PD for the virtual

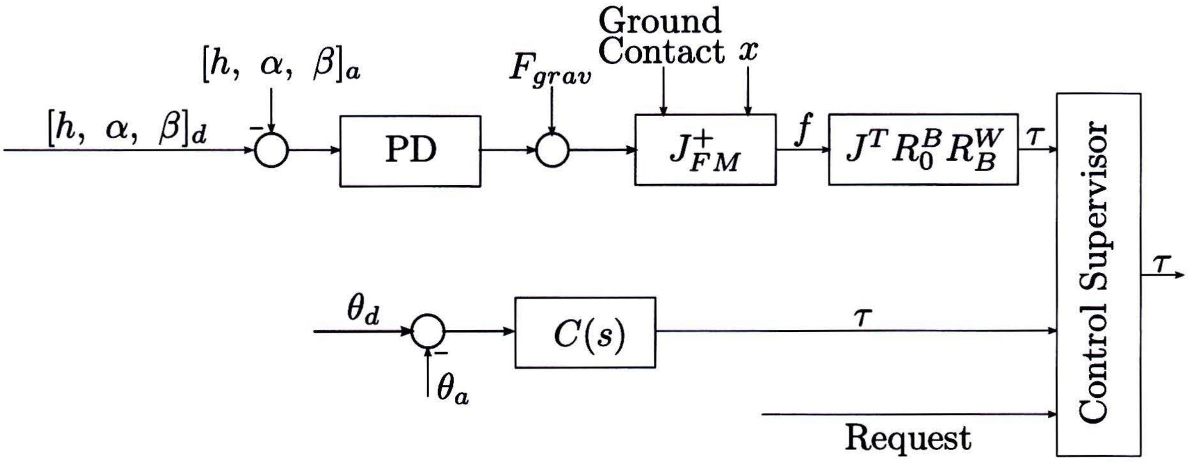


Figure 6.5: Control structure of the robot.

actuators and a lead-filter, low-pass filter for the leg joint controllers.

Control Supervisor

The Control Supervisor is responsible for selecting the appropriate control signals. This depends on the request and state of legs. If there is no particular request the control signals are forwarded depending on their state, i.e. in the air or on the ground. Currently no requests are implemented and the control supervisor simply acts as an addition of both control signals.

Force distribution

The Force distribution distributes the forces that should act on the body to the legs. In Fig. 6.5 this component represented by J^{+}_{FM}. The algorithm used is described in Section. 4.4. To solve these equations the C++ linear algebra library Eigen is used. Because the solution of equation (4.24) is non-unique because the system is under-determined, it is calculated using the singular value decomposition. The returned solution is guaranteed to minimize the Euclidean norm $||\mathbf{J_{FM}}\boldsymbol{f} - \mathbf{W}||$ and thus distributes the body force equally to the leg forces.

Jacobian

The Jacobian component determines the torques on the joints depending on the requested force, the component is represented by $J^T R_0^B R_B^W$ in Fig. 6.5. First the forces with respect to the world frame are transformed to the body frame. Then using the Jacobian, which is implemented as derived in (4.12), these forces are translated to the torques needed for the joints.

6.2.3 Robot

The robot is regarded as the drivers needed to communicate with the robot, a safety component and some virtual encoders. The robot implementation scheme is depicted in Fig. 6.6, this scheme consists of the bottom two layers as shown in Fig. 6.2 i.e. the virtual hardware and hardware layer.

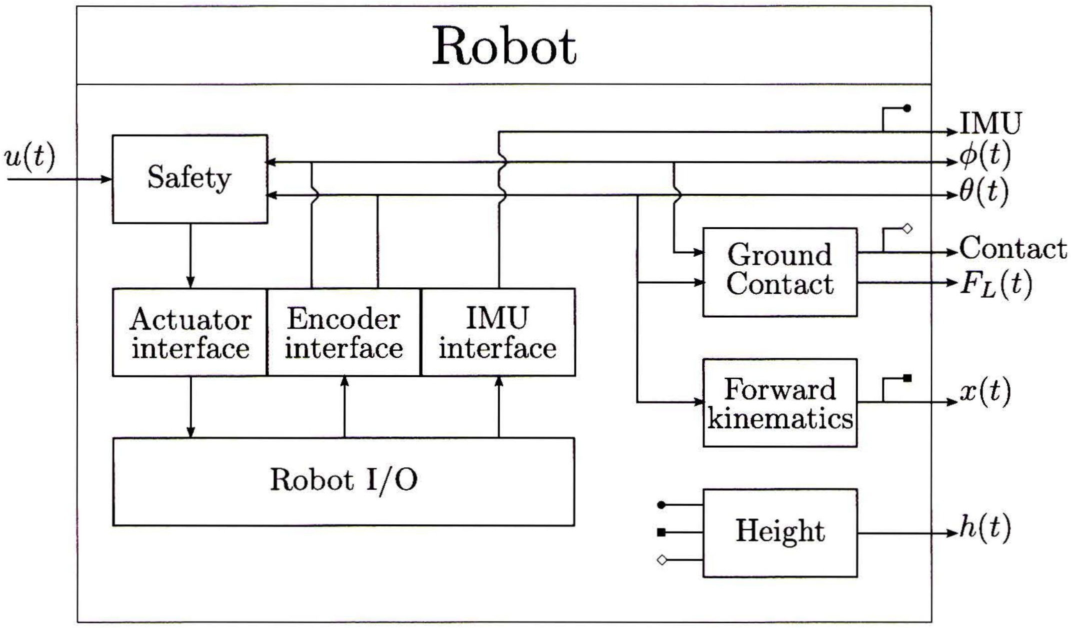


Figure 6.6: Software structure of the robot.

Safety

This component acts as an emergency button on a software level such that the robot does not damages itself. The component can disable actuators and overrule the control signal. It also check the initialization of the actuator encoders such that they do not differ from the joint encoders. Furthermore the safety checks whether a cable is assumed to be broken or that a joint ranges out of its safety limits.

Hardware interfaces

The hardware interfaces, i.e. the actuator, encoder and IMU interface, are responsible for the initialization of the corresponding hardware. Also it converts counts and voltages to usable positions and orientation angles.

Ground contact

The ground contact determines whether a leg is standing on the ground or not. The robot does not have force or touch sensors such that this has to be indicated using a virtual sensor. This virtual sensor is obtained using the difference between the actuator and joint position from the second joint. Then it is checked if the difference exceeds a specified threshold, which is assumed to be caused by a force acting on the leg. In Fig. 6.7 the method is demonstrated, as can be seen at 32.4 seconds the leg touches the ground.

Forward kinematics

The forward kinematics calculates the position of the tip of a leg with respect to body frame, this information is extracted from the URDF of the robot. The forward kinematics are implemented

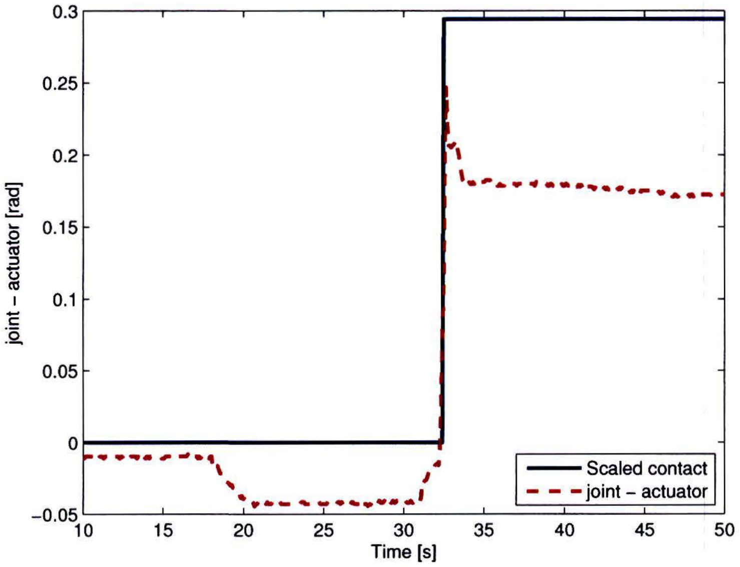


Figure 6.7: Determination of ground contact.

using the Kinematics and Dynamics Library (KDL) of the Orocos Project.

Height

The height of the body is determined under the assumption that the hexapod stands on a flat surface. The height is obtained by averaging the z-position of the tips, transformed to the world coordinate frame.

$$h = -\frac{1}{n}\sum_{i=1}^{p}\left[\mathbf{R}_B^W\left(\boldsymbol{x}_i + \Delta z\right)\right]_z \tag{6.1}$$

Where p is the number of legs on the ground and Δz is the distance from the frame of the body to its bottom. The $[\ldots]_z$ denotes that the z is taken for the summation. If none of the legs are on the ground the height is equal to 0.

Chapter 7

Hexapod control

Two main situations for control can be distinguished namely (i) the leg is lifted in the air and (ii) the leg is standing on the ground. In the first situation the leg can be regarded as a RRR-manipulator, the control objective in this case is to control the position of the tip. The second can be regarded as a situation in which several manipulators are performing cooperative manipulation namely on the body. In this case the control objective changes to the orientation and height of the body.

7.1 Leg control

7.1.1 Identification

To identify the system, the equations of motion ((4.16), (4.13)) in time domain can be rewritten to the frequency domain using Laplace

$$J_m s^2 \Phi = T_m + (k_s + b_s)(\Theta - \Phi) \tag{7.1}$$

$$J_l s^2 \Theta = -(k_s + b_s)(\Theta - \Phi) \tag{7.2}$$

Rewriting both equations results in two transfer functions, one for the collocated case and one for the non-collocated case.

$$H_{\text{non-collocated}} = \Theta / T_m = \frac{b_s s + k_s}{J_m J_l s^4 + (J_m + J_l) b_s s^3 + (J_m + J_l) k_s s^2} \tag{7.3}$$

Each joint is actuated by a motor. Since the third degree of freedom is coupled to the second, this degree of freedom is decoupled using (4.3). Using a frequency response measurement, the system can be identified in closed loop. Indirect identification is performed by measuring the frequency response from $w_i(t)$ to $v_i(t)$. This is the frequency response function (FRF) of the sensitivity function S_i $(= 1/(1 + H_i C_i))$ of the feedback loop in the ith joint. Using the identified sensitivity, for a known controller, the plant may be obtained via

$$H_i(j\omega) = \frac{1 - S_i(j\omega)}{C_i(j\omega) S_i(j\omega)} \tag{7.4}$$

The obtained frequency response functions are depicted in Fig. 7.2. The frequency response of the actuators clearly show a minus 2 slope which indicates a mass system. The FRF of the

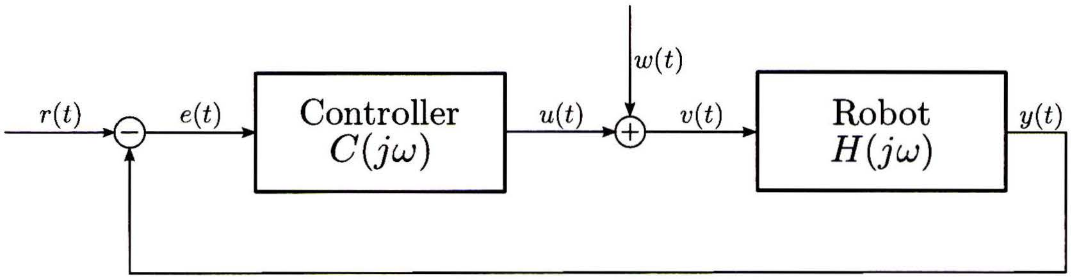


Figure 7.1: Feedback structure for system identification.

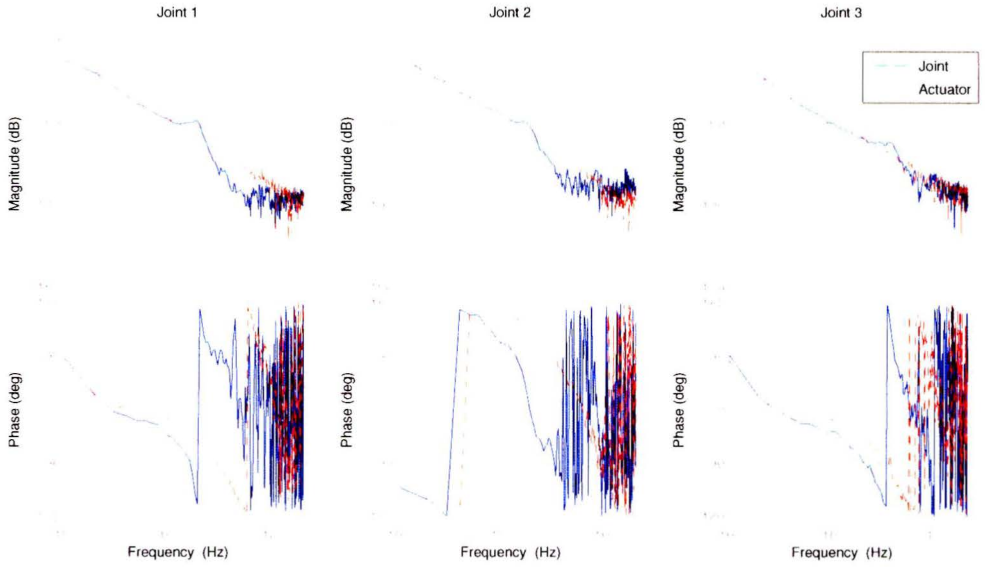


Figure 7.2: Frequency Response Functions.

joints starts with a minus 2 slope, then a resonance occurs and it becomes a minus 4 slope. This indicates a 2-mass-spring-damper model as expected. The FRF diagrams also indicate delays, the delay of the actuators is approximately 9 ms. The cause of the delay is not thoroughly investigated.

7.1.2 Joint control

Using the obtained frequency response of the joints, a feedback controller is designed. A lead-filter is used to obtain phase margin. To suppress the high frequent noise a low-pass filter is added.

The values used for the control filters are depicted in Table 7.1. The open-loop bode diagrams for the joints are shown in Fig. 7.3. A higher bandwidth is not possible due to actuator limitations.

7.1.3 Joint space results

To test the controller two type of experiments are conducted. The first is the step response of the joint. The second experiment is a tracking experiment. The result of the step response of the first joint of leg 1, is depicted in Fig. 7.4.

Table 7.1: Controller parameters

	Joint 1	Joint 2	Joint 3
Gain [-]	-10.0	5.0	-5.0
Lead zero [Hz]	1.7	1.7	1.7
Lead pole [Hz]	15.0	15.0	15.0
Low-pass pole [Hz]	25.0	25.0	25.0
Bandwidth [Hz]	2.6	3.2	3.3
Modulus margin [dB]	4.2	5.9	6.0
Phase margin [deg]	45.9	39.5	36.9
Gain margin [dB]	56.0	7.9	38.5

During the tracking experiment the joint has to follow a second-order trajectory while the leg is in the air. Using a second-order trajectory the velocity and acceleration references are also available to use for feedforward control. The first experiment uses a trajectory which can be related to 1 step per 2 seconds (0.5 Hz). It can be seen in Fig. 7.5 that the trajectory is tracked with an average absolute error of 0.033 rad when only using feedback. Adding feedforward improves the tracking sufficiently with an average absolute error of 0.010 rad. The peaks in the error are caused when the direction of the joint is changed. This can be explained to the compliance between the actuator and joint and the backlash caused by the gearbox. To investigate the maximum stepping size the step frequency is increased to 1.5 steps per second. It can be seen in Fig. 7.6 that the controller is not able to track the trajectory, the actuator even saturates. Using feedforward the tracking is greatly improved as expected and no more saturation occurs. Looking at the control signal it can be seen that the actuator is almost working at its hardware limit (-5 till 5 A) and probably no greater speed can be achieved. This test was conducted outside the continuous operation of the actuator with additional cooling. Noting that the leg was lifted in the air, it can be expected that when the foot is positioned at the ground not the same speed can be achieved.

7.2 Body control

In the second case the principle of virtual actuator control [41] is used. By using virtual components such as spring and dampers the requested forces on the hexapod body are determined. The advantage of using virtual actuators is that a simple linear control structure can be used. Furthermore the positioning of the leg is relatively simple because when standing on an obstacle the tip of the position does not have to be adjusted because the forces are simply altered.
A distinghuisment is made between the hexapod body and world variables. The body variables are the height and orientation of the robot $(h,\ \alpha,\ \beta)$, the following can be used

$$\boldsymbol{W}_{\mathcal{B}} = \begin{bmatrix} 0 \\ 0 \\ k_h(h_d - h_a) - b_h\dot{h}_a + g \\ k_\alpha(\alpha_d - \alpha_a) - b_\alpha\dot{\alpha}_a \\ k_\beta(\beta_d - \beta_a) - b_\beta\dot{\beta}_a \\ 0 \end{bmatrix} \tag{7.5}$$

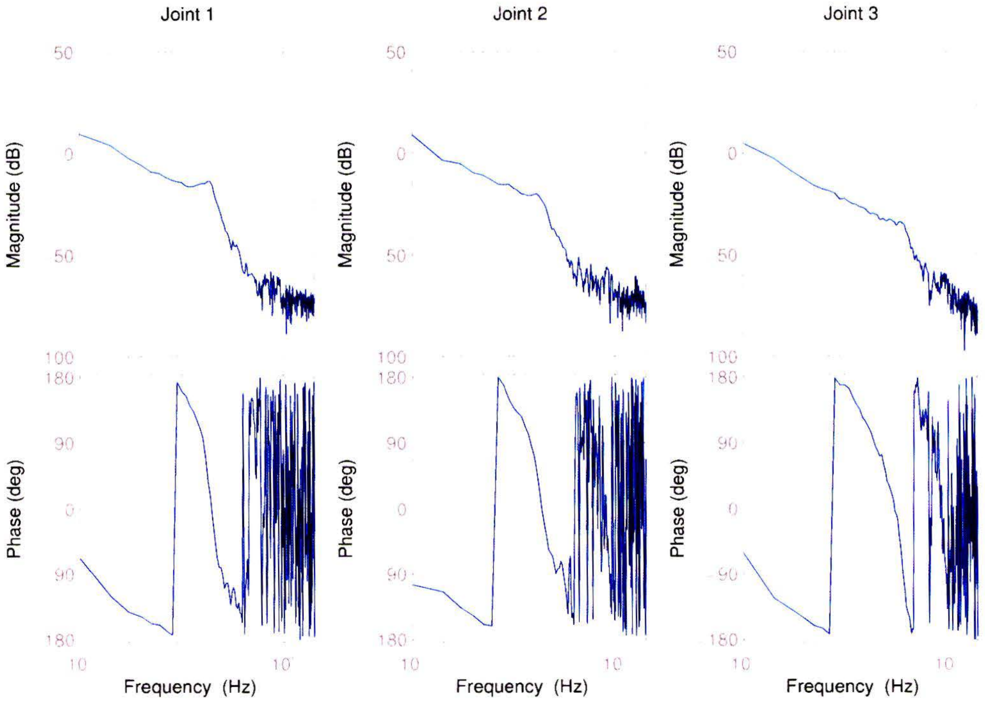


Figure 7.3: Open-loop bode diagram.

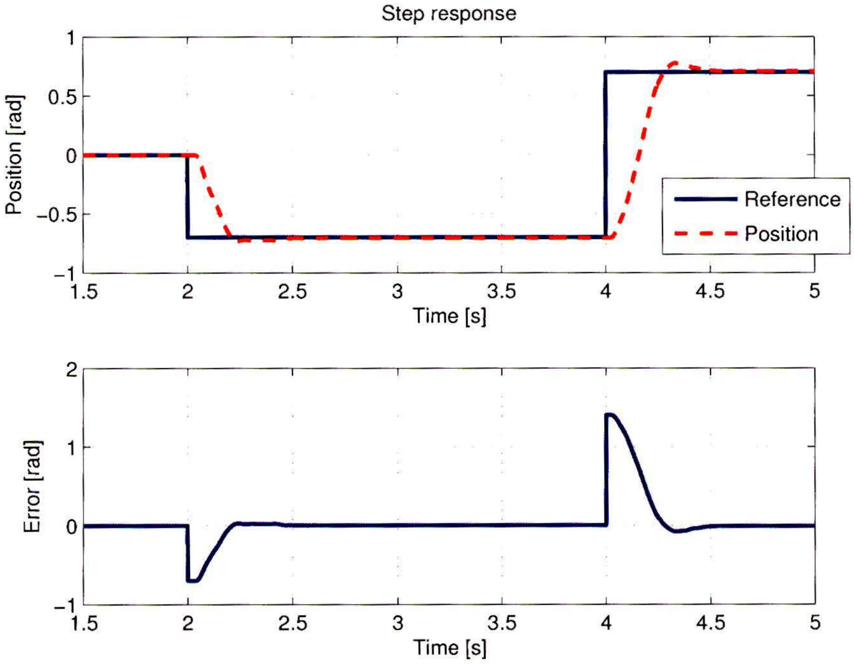


Figure 7.4: Step response of joint 1.

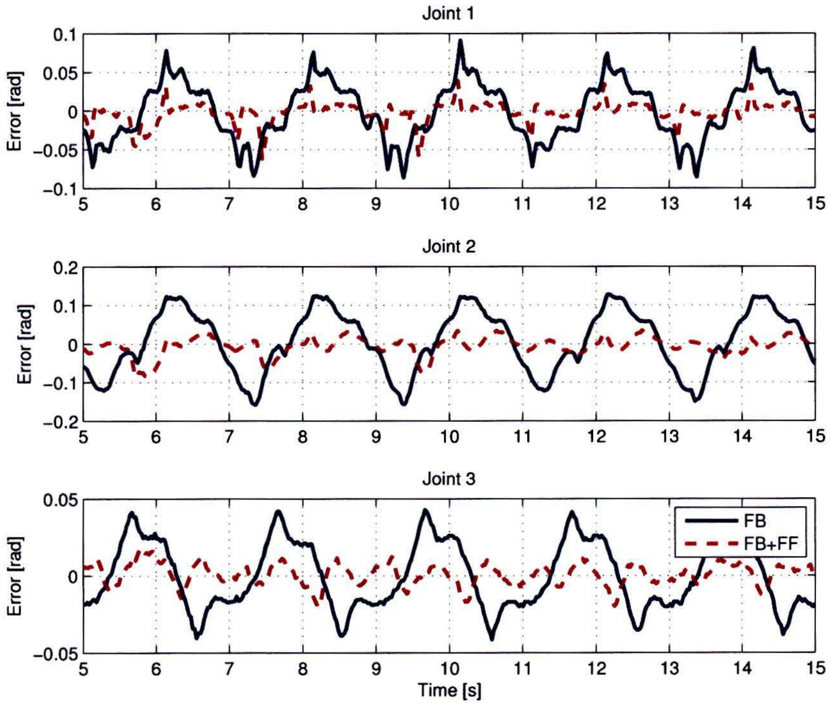


Figure 7.5: Tracking response of a 0.5 Hz trajectory.

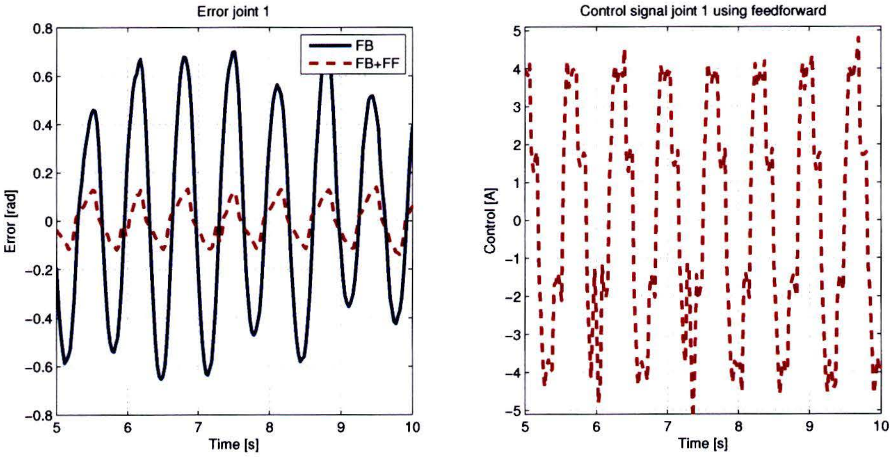


Figure 7.6: Tracking response of joint 1 of a 1.5 Hz trajectory.

$\boldsymbol{W}_{\mathcal{B}}$ is the wrench acting on the body as a function of the body variables. It should be noted that the stiffness and damping of the virtual springs and dampers, and the gravity, should be negative to counteract the gravitational forces. The hexapod world variables $(\dot{x}, \dot{y}, \dot{\gamma})$ correspond to the movement of the hexapod. They induce the following wrench on the body

$$\boldsymbol{W}_{\mathcal{W}} = \begin{bmatrix} b_x(\dot{x}_d - \dot{x}_a) \\ b_y(\dot{y}_d - \dot{y}_a) \\ 0 \\ 0 \\ 0 \\ b_\gamma(\dot{\gamma}_d - \dot{\gamma}_a) \end{bmatrix} \tag{7.6}$$

$\boldsymbol{W}_{\mathcal{W}}$ is the wrench acting on the body depending on the velocities of the hexapod. These velocities should be generated using a trajectory planner. As can be seen $\boldsymbol{W}_{\mathcal{W}}$ does not depend on the position in the world frame. The force distribution algorithm is used to obtain the control signal for the motors. Using the motor torque constant k_t and the gear ratio n the needed current or control signal can be determined by

$$I = \frac{1}{k_t n}\tau_d \tag{7.7}$$

It should be noted that there is no torque feedback in the body controller. Furthermore the torque is applied at the motors and not directly at the joints. However in the static case the torque on the actuators is equal to the torque on the joints.

7.3 Hexapod control

The joint control and body control is combined in one control scheme. This control scheme is used for the locomotion of the hexapod. In the hexapod control the joint control and body control are added together. This results that $\boldsymbol{W}_{\mathcal{W}}$ is not used because the locomotion of the legs should take care of propelling the body forward. The disadvantage of the current method is that the hexapod is only able to walk on a sufficiently flat surface. To overcome obstacles force feedback is needed such that the position control signal is counteracted. Another possibility is to adapt the gait of the leg which is positioned at an obstacle. Using this control scheme the hexapod is able to walk and turn. Low obstacles are counteracted using the posture controller and compliance. As can be seen in Fig. 7.7 the hexapod is not able to walk at the requested height. However it does maintain a constant height. The peaks in the figure indicate that legs are being positioned on the ground, resulting in some additional force. Investigating the roll and pitch during walking shows that the hexapod is able to maintain its orientation. From the body coordinates it can be concluded that the movement is periodic, which can be expected due to the references. Looking at the joint errors during walking, as shown in Fig. 7.8 it can be seen that these are also periodic. The error of the first joint, the joint which is responsible for propelling the body forward, increases when in the standing phase. For the second and third joint peaks in the error occur at the moment of lifting the leg. This is caused by the different goals of the controllers. At the moment the leg needs to be lifted, the body controller still pushes the leg down. If the error becomes large enough, the leg is released due to the leg controller. A possible solution can be implementing a transition in the body controller, such that the force is decreased when it receives a request to release a leg.

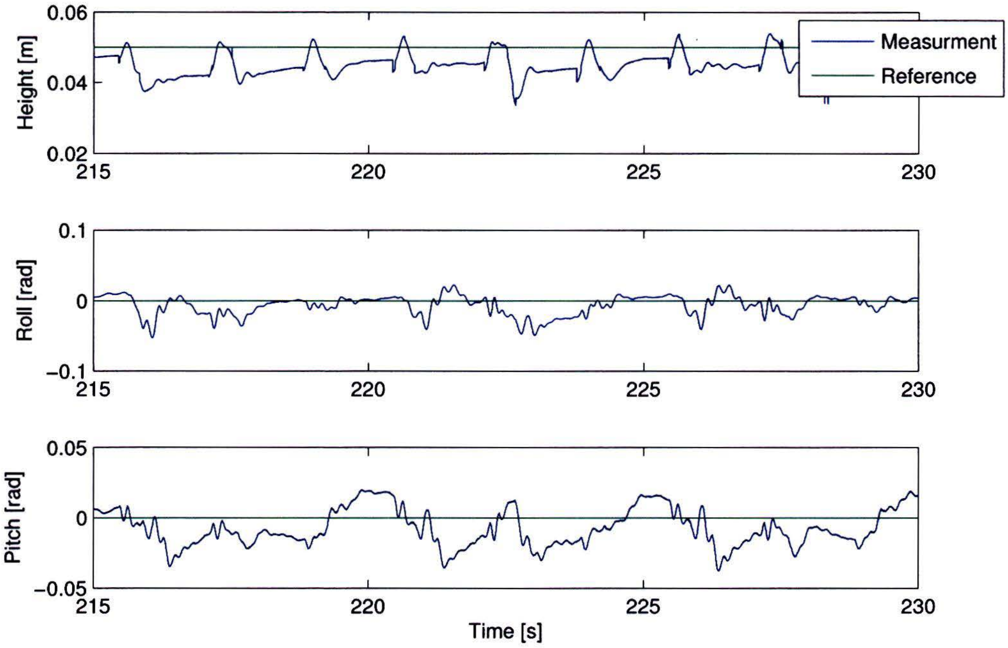


Figure 7.7: Body coordinates during walking.

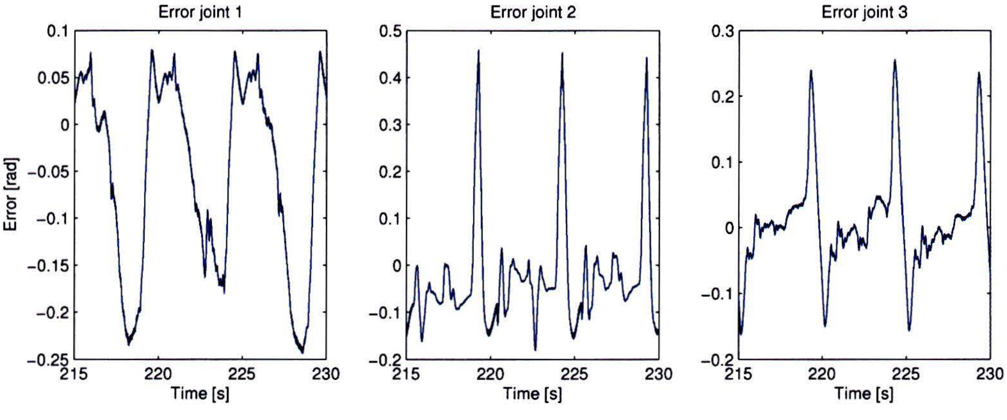


Figure 7.8: Tracking error of leg one.

Switching Using virtual actuator control on the body, i.e. switching between joint and body control was also tested. The advantage is that the tips are not position controlled when standing and thus standing on an object does not influence the body of the hexapod. The downside of switching is that it is difficult to propel the body forward because the velocities of the body were not available, i.e. not implemented. This also results in the case that the position of the tip is not tracked when standing on the ground such that when it needs to be lifted, tracking control is turned on, resulting in a step of the position of the tip. It is questionable if the switching results in a stable system because it may cause undesired impulses, this is however not investigated.

Chapter 8

Conclusions and Recommendations

8.1 Conclusions

The control of a walking robot is more complex then the control of a wheeled robot. Multiple legs should follow a trajectory while keeping the body balanced. A walking robot is sensitive to contact forces with the ground. The effect of the contact forces in the robot are enhanced due to a relative high center of gravity of the hexapod and compliance in the legs.

The goal in this report was to develop and implement a software structure that enables the hexapod to walk. The exact goal that described this project was

Design and implement a software structure that controls the hexapodal robot in real-time such that it is able to walk.

This goal is achieved as the hexapod is able to walk.

- A software architecture was developed containing, drivers, forward and inverse kinematics and many more components,
- a leg controller was developed and implemented,
- a body controller was developed and implemented.

An important aspect is the body controller using a force distribution algorithm in combination with virtual actuators. This enables the hexapod to maintain its orientation while walking. Due to the hexapod design it is sensitive to tilt depending on the foot placement. The force distribution is able to counteract this effect by applying a moment in the opposite direction. During walking the roll and pitch stay within a range of 0.05 [rad]. A challenge remaining is to extend the functionality of the force distribution with respect to phase transitions. The force transition for the leg from swing to stance phase and vice versa could be smoothed. This would decrease the joint errors for the second and third joint and the height of the hexapod would be more constant.

Besides the control the other important aspect is the gait. The gait designed is relative simple but functional. It is able to move in different directions and the parameters, such as the step size and step height, are configurable. However to change these parameters and change the direction the gait first have to finish its step and be reconfigures. So this can not be changed on the fly.

Furthermore the components which create the trajectory for the legs are standalone components. This means that there is no layer on top to coordinate the walking behavior.
The full potential of the robot is not exploited. The robot can walk at a higher speed by decreasing the cylce time of the gait. Also the robot does not make a full step because the leg controller does not provide a sufficient control effort when in the stance phase. An increased effort is needed with respect to the swing phase. This could be achieved using feedforward. Furthermore walking is a repetitive activity, this property also can be exploited by using for example a repetitive controller.

The software structure is designed such that it is flexible. It is component based such that it is easy to modify or replace components to test different types of control. The components are also designed such that they are configurable and thus easy to adjust.
The components and structure designed, can be easily re-used for different hexapods. Also adding new functionalities to the robot such as force control or vision is applicable due to the structure.

8.2 Recommendations

The goal of this project is achieved however there are a lot of possibilities to increase the performance of the hexapod. These recommendations range from several research topics to practical implementations.

8.2.1 Hardware

It is recommended to evaluate the hardware taking in mind that the final goald will be a running hexapod.

Actuation The combination of actuators and gearboxes should be evaluated depending on its goal. As stated in Chapter 3 the speed of the actuators are limited. It can be expected that the first degree of freedom is about speed because that actuator is responsible for propelling the body at a high pace. The second actuator should have sufficient force to maintain the mass of the robot. The the third will be a trade-off between speed and force such that it is able to generate an impulse to launch the body, i.e. the hexapod will jump forward. This aspect is also important for the second joint if eventually a jumping or running hexapod is required.

Compliant legs As already stated complaint legs benefit a walking robot. However it should be evaluated if this is the desired method of obtaining compliance. The main disadvantage of this method is that the height of the hexapod is increased. This results in a relative high center of mass. A possibility would be removing the torsional springs and only add a linear spring in the third link. This is also done in several walking robots like Big Dog and little dog. The disadvantage of this method would be that the position of the tip is unknown but the question is whether this is a problem.

Force sensors The walking robot would also benefit using force sensors. This should be able by using the torsional springs however this did not succeed. Further investigation is required to obtain a force measurement. Using force sensors would enable the advantage of force feedback and thus the robot can better react to obstacles.

8.2.2 Software

The software structure is a first set-up therefore some basic and practical implementations are made. These can be improved by the following recommendations.

Odometry The velocities of the robot are currently not determined. These could be obtained by using the data of the IMU. The IMU returns are very noisy measurement using appropriate filter techniques and sensor fusion of the tip positions accurate velocities can be obtained. Using these velocities, the position of the body also can be defined such that planning can be used.

Dynamics The torsional springs in the system should be exploited to reduce energy consumption, especially when the robot starts to run. As in [9] the passive dynamics of a one-legged compliant robot were used which resulted in energy savings of 95%. The dynamics can be investigated using the prototype leg and additional force sensors.

Gait generation The gait used in this project is a relative simple continuous gait. The gait can be improved especially at the point of touch down and release of the tip of the leg. This transition is currently not smooth resulting in an higher impact then necessary.
Stability margins should also be investigated when walking such that legs can be adapted to maintain stability. Stability margins for future running do not apply.
A practical implementation for the current gaits would be a smooth transition between different gaits.

Bibliography

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

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

[3] Gazebo, the player project. http://playerstage.sourceforge.net/, 2011.

[4] Ogre 3d. http://www.ogre3d.org/, 2011.

[5] Open dynamics engine. http://www.ode.org/, 2011.

[6] Ros.org. http://www.ros.org/wiki/, 2011.

[7] Ros.org urdf package. http://www.ros.org/wiki/urdf, 2011.

[8] MSL Technical Committee 1997-2011. Middle size robot league rules and regulations for 2011. Technical report, RoboCup, December 2010.

[9] M. Ahmadi and M. Buehler. A control strategy for stable passive running. In *Intelligent Robots and Systems 95. 'Human Robot Interaction and Cooperative Robots', Proceedings. 1995 IEEE/RSJ International Conference on*, volume 3, pages 152–157 vol.3, aug 1995.

[10] Beckhoff. *EtherCAT, Ultra high-speed communication*, 2011.

[11] Randall D. Beer, Roger D. Quinn, Hillel J. Chiel, and Roy E. Ritzmann. Biologically inspired approaches to robotics: what can we learn from insects? *Commun. ACM*, 40:30–38, March 1997.

[12] H. Bruyninckx. Open robot control software: the orocos project. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, volume 3, pages 2523–2528 vol.3, 2001.

[13] J. Buchli, M. Kalakrishnan, M. Mistry, P. Pastor, and S. Schaal. Compliant quadruped locomotion over rough terrain. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 814–820, oct. 2009.

[14] S.K.-K. Chu and G.K.-H. Pang. Comparison between different model of hexapod robot in fault-tolerant gait. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 32(6):752–756, nov 2002.

[15] P. Gonzalez de Santos, E. Garcia, and J. Estremera. A control architecture for humanitarian-demining legged robots. In *Proceedings of the 6th International Conference on Climbing and Walking Robots (CLAWAR)*, pages 383–390, September 2003.

[16] Pablo Gonzalez de Santos, Elena Garcia, and Joaquin Estremera. *Quadrupedal Locomotion; An Introduction to the Control of Four-legged.* Springer, 2006.

[17] C. Ferrell. Robust and adaptive locomotion of an autonomous hexapod. In *From Perception to Action Conference, 1994., Proceedings*, pages 66–77, sept. 1994.

[18] Jose A Galvez, Joaquin Estremera, and Pablo Gonzalez de Santos. A new legged-robot configuration for research in force distribution. *Mechatronics*, 13(8-9):907–932, 2003.

[19] Jianjuen Hrr, J. Pratt, Chee-Meng Chew, H. Herr, and G. Pratt. Adaptive virtual model control of a bipedal walking robot. In *Intelligence and Systems, 1998. Proceedings., IEEE International Joint Symposia on*, pages 245–251, may 1998.

[20] F. Iida. Cheap Design Approach to Adaptive Behavior: Walking and Sensing through Body Dynamics. *International Symposium on Adaptive Motion of Animals and Machines*, 2005.

[21] F. Iida, G. Gomez, and R. Pfeifer. Exploiting body dynamics for controlling a running quadruped robot. In *Advanced Robotics, 2005. ICAR '05. Proceedings., 12th International Conference on*, pages 229–235, july 2005.

[22] A. Ijspeert. Central pattern generators for locomotion control in animals and robots: A review. *Neural Networks*, 21(4):642–653, May 2008.

[23] D. Jansen and H. Buttner. Real-time ethernet the ethercat solution. *Computing Control Engineering Journal*, 15(1):16–21, feb.-march 2004.

[24] Mrinal Kalakrishnan, Jonas Buchli, Peter Pastor, Michael Mistry, and Stefan Schaal. Learning, planning, and control for quadruped locomotion over challenging terrain. *The International Journal of Robotics Research*, 30(236), 2010.

[25] C.A. Klein and S. Kittivatcharapong. Optimal force distribution for the legs of a walking machine with friction cone constraints. *Robotics and Automation, IEEE Transactions on*, 6(1):73–85, feb 1990.

[26] Daniel E. Koditschek, Robert J. Full, and Martin Buehler. Mechanical aspects of legged locomotion control. *Arthropod Structure & Development*, 33(3):251–272, 2004. Arthropod Locomotion Systems: from Biological Materials and Systems to Robotics.

[27] N. Koenig and A. Howard. Design and use paradigms for gazebo, an open-source multi-robot simulator. In *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 3, pages 2149–2154 vol.3, sept.-2 oct. 2004.

[28] J. Zico Kolter and Andrew Y. Ng. The stanford littledog: A learning and rapid replanning approach to quadruped locomotion. *The International Journal of Robotics Research*, 30(2):150–174, February 2011.

[29] Woong Kwon and Beom Hee Lee. A new optimal force distribution scheme of multiple cooperating robots using dual method. *J. Intell. Robotics Syst.*, 21:301–326, April 1998.

[30] Hongyi Liu and Bangchun Wen. Force distribution for the legs of a quadruped walking vehicle. *Journal of Robotic Systems*, 14(1):1–8, 1997.

[31] C. Mahfoudi, K. Djouani, S. Rechak, and M. Bouaziz. Optimal force distribution for the legs of an hexapod robot. In *Control Applications, 2003. CCA 2003. Proceedings of 2003 IEEE Conference on*, volume 1, pages 657–663 vol.1, june 2003.

[32] Eve Marder and Dirk Bucher. Central pattern generators and the control of rhythmic movements. *Current Biology*, 11(23):R986–R996, 2001.

[33] Robert B. Mcghee and Geoffrey I. Iswandhi. Adaptive locomotion of a multilegged robot over rough terrain. *Systems, Man and Cybernetics, IEEE Transactions on*, 9(4):176–182, april 1979.

[34] D. Micci-Barreca and H. Ogmen. A central pattern generator for insect gait production. In *From Perception to Action Conference, 1994., Proceedings*, pages 348–351, sept. 1994.

[35] B. Mrkajic. Systems and control library development. Master's thesis, Eindhoven University of Technology, August 2011.

[36] Michael P. Murphy, Aaron Saunders, Cassie Moreira, Alfred A. Rizzi, and Marc Raibert. The littledog robot. *The International Journal of Robotics Research*, 30(2):145–149, February 2011.

[37] M.A. Nahon and J. Angeles. Real-time force optimization in parallel kinematic chains under inequality constraints. *Robotics and Automation, IEEE Transactions on*, 8(4):439–450, aug 1992.

[38] R. Prajoux and L. de S.F. Martins. A walk supervisor architecture for autonomous four-legged robots embedding real-time decision-making. In *Intelligent Robots and Systems '96, IROS 96, Proceedings of the 1996 IEEE/RSJ International Conference on*, volume 1, pages 200–207 vol.1, nov 1996.

[39] G.A. Pratt and M.M. Williamson. Series elastic actuators. *Intelligent Robots and Systems, IEEE/RSJ International Conference on*, 1:399, 1995.

[40] Jerry Pratt, Peter Dilworth, and Gill Pratt. Virtual model control of a bipedal walking robot. In *IEEE Conference on Robotics and Automation*, pages 193–198, 1997.

[41] Jerry Pratt, Ann Torres, Peter Dilworth, and Gill Pratt. Virtual actuator control. In *IEEE International Conference on Intelligent Robots and Systems*, pages 1219–1226, 1996.

[42] Morgan Quigley, Ken Conley, Brian P. Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y. Ng. Ros: an open-source robot operating system. In *ICRA Workshop on Open Source Software*, 2009.

[43] M. Raibert, M. Chepponis, and H. Jr. Brown. Running on four legs as though they were one. *Robotics and Automation, IEEE Journal of*, 2(2):70–82, jun 1986.

[44] Marc H. Raibert. Trotting, pacing and bounding by a quadruped robot. *Journal of Biomechanics*, 23(Supplement 1):79–81, 1990.

[45] CH Robotics. *CHR-6d Digital Inertial Measurement Unit*, product datasheet - rev. 1.4 edition.

[46] S. Rutishauser, A. Sprowitz, L. Righetti, and A.J. Ijspeert. Passive compliant quadruped robot using central pattern generators for locomotion control. In *Biomedical Robotics and Biomechatronics, 2008. BioRob 2008. 2nd IEEE RAS EMBS International Conference on*, pages 710–715, oct. 2008.

[47] Uluc Saranli, Martin Buehler, and Daniel E. Koditschek. Rhex: A simple and highly mobile hexapod robot. *The International Journal of Robotics Research*, 20(7):616–631, 2001.

[48] A. Seyfarth, H. Geyer, R. Blickhan, S. Lipfert, J. Rummel, Y. Minekawa, and F. Iida. Running and walking with compliant legs. In Moritz Diehl and Katja Mombaur, editors, *Fast Motions in Biomechanics and Robotics*, volume 340 of *Lecture Notes in Control and Information Sciences*, pages 383–401. Springer Berlin / Heidelberg, 2006.

[49] Alexander Shkolnik, Michael Levashov, Ian R. Manchester, and Russ Tedrake. Bounding on rough terrain with the littledog robot. *The International Journal of Robotics Research*, December 2010.

[50] Bruno Siciliano and Oussama Khatib, editors. *Springer Handbook of Robotics*. Springer, 2008.

[51] Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, and Giuseppe Oriolo. *Robotics Modelling, Planning and Control*. Advanced Textbooks in Control and Signal Processing. Springer, 2009.

[52] James Andrew Smith. *Galloping, Bounding and Wheeled-Leg Modes of Locomotion on Underactuated Quadrupedal Robots*. PhD thesis, McGill University, November 2006.

[53] P. Soetens. *A software framework for real-time and distributed robot and machine control*. PhD thesis, Katholieke Universiteit Leuven, May 2006.

[54] Yurak Son, Takuya Kamano, Takashi Yasuno, Takayuki Suzuki, and Hironobu Harada. Generation of adaptive gait patterns for quadruped robot with cpg network including motor dynamic model. *Electrical Engineering in Japan*, 155(1):35–43, 2006.

[55] Mark W. Spong, Seth Hutchinson, and M. Vidyasagar. *Robot Modeling and Control*. John Wiley & Sons, Inc., 2006.

[56] Ann Torres. Virtual model control of a hexapod walking robot, December 1996.

[57] R. Vidoni and A. Gasparetto. Efficient force distribution and leg posture for a bio-inspired spider robot. *Robot. Auton. Syst.*, 59:142–150, February 2011.

[58] M. Waibel, M. Beetz, J. Civera, R. D'Andrea, J. Elfring, D. Galvez-Lopez, K. Haussermann, R. Janssen, J.M.M. Montiel, A. Perzylo, B. Schiessle, M. Tenorth, O. Zweigle, and R. van de Molengraft. Roboearth. *Robotics Automation Magazine, IEEE*, 18(2):69–82, 2011.

[59] Debao Zhou, K. H. Low, and Teresa Zielinska. An efficient foot-force distribution algorithm for quadruped walking robots. *Robotica*, 18:403–413, July 2000.

Appendix A

Kinematics

A.1 Forward kinematics

The homogeneous transformation matrices for the leg as depicted in Fig. 4.2b are

$$\mathbf{A}_1^0 = \begin{bmatrix} C_1 & 0 & S_1 & a_1C_1 \\ S_1 & 0 & -C_1 & a_1S_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.1}$$

$$\mathbf{A}_2^1 = \begin{bmatrix} C_2 & -S_2 & 0 & a_2C_2 \\ S_2 & C_2 & 0 & a_1S_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.2}$$

$$\mathbf{A}_3^2 = \begin{bmatrix} C_3 & -S_3 & 0 & a_{31}C_3 - a_{32}S_3 \\ S_3 & C_3 & 0 & a_{31}S_3 + a_{32}C_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.3}$$

where $\sin(\theta_i) = S_i$, $\cos(\theta_i) = C_i$ except for S_3 and C_3, here ϑ_i is used instead of θ_i. Then the homogenous transformation matrices from the root can be derived

$$\mathbf{T}_1^0 = \mathbf{A}_1^0 \tag{A.4}$$

$$\mathbf{T}_2^0 = \mathbf{A}_1^0\mathbf{A}_2^1 \tag{A.5}$$

$$\mathbf{T}_3^0 = \begin{bmatrix} \mathbf{R}_1^0 & o_2^0 \\ 0 & 1 \end{bmatrix} \mathbf{A}_3^2 \tag{A.6}$$

The origin of the frames can be described as follows, note that o_3^0 is not affected by θ_2 because the link is remotely driven.

$$o_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{A.7}$$

$$o_1^0 = \begin{bmatrix} a_1 C_1 \\ a_1 S_1 \\ 0 \end{bmatrix} \tag{A.8}$$

$$o_2^0 = \begin{bmatrix} a_1 C_1 + a_2 C_1 C_2 \\ a_1 S_1 + a_2 S_1 C_2 \\ a_2 S_2 \end{bmatrix} \tag{A.9}$$

$$o_3^0 = \begin{bmatrix} a_1 C_1 + a_2 C_1 C_2 + C_1(a_{31} C_3 - a_{32} S_3) \\ a_1 S_1 + a_2 S_1 C_2 + S_1(a_{31} C_3 - a_{32} S_3) \\ a_2 S_2 + a_{32} C_3 + a_{31} S_3 \end{bmatrix} \tag{A.10}$$

The following for z_i

$$z_0 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T \tag{A.11}$$

$$z_1 = \begin{bmatrix} S_1 & -C_1 & 0 \end{bmatrix}^T \tag{A.12}$$

$$z_2 = \begin{bmatrix} S_1 & -C_1 & 0 \end{bmatrix}^T \tag{A.13}$$

A.2 Inverse kinematics

The inverse kinematics from the tip to the root of the leg is [16]

$$\theta_1 = \operatorname{atan2}(y, x) \tag{A.14}$$

$$\theta_2 = -\operatorname{atan2}(B, A) + \operatorname{atan2}\left(D, \pm\sqrt{A^2 + B^2 - D^2}\right) \tag{A.15}$$

$$\theta_3 = \operatorname{atan2}(z - a_2 S_2, -B - a_2 C_2) - \theta_2 - E \tag{A.16}$$

with

$$A = -z \tag{A.17}$$

$$B = a_1 - (x C_1 + y S_1) \tag{A.18}$$

$$D = \frac{-B^2 + a_3^2 - a_2^2 - z^2}{2 a_2} \tag{A.19}$$

$$E = \operatorname{atan2}(a_{32}, a_{31}) \tag{A.20}$$

$$a_3 = \sqrt{a_{31}^2 + a_{32}^2} \tag{A.21}$$

Note that the E and a_3 terms are caused by the offset from the third joint to the feet of the leg.

Appendix B

Dynamics

Because flexibility has to be taken into account there are $2N$ generalized coordinates with N the number of joints.

$$\boldsymbol{q} = \begin{bmatrix} \boldsymbol{q}_L & \boldsymbol{q}_M \end{bmatrix}^T \tag{B.1}$$
$$= \begin{bmatrix} \theta_1 & \theta_2 & \vartheta_3 & \phi_1 & \phi_2 & \phi_3 \end{bmatrix}^T \tag{B.2}$$

q_L are the generalized coordinates of the joint positions and q_M are the corresponding motor positions. Note the third term ϑ_3, this is due the fact that the link is remotely driven. The Lagrangian [55, 51, 50], which is the difference of the kinetic and potential energy, of a system is given by

$$L(q, \dot{q}, t) = T(q, \dot{q}, t) - U(q, t) \tag{B.3}$$

where T and U respectively denote the kinetic and potential energy. The Euler-Langrange equation is

$$\tau_i = \frac{d}{dt}\frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} \tag{B.4}$$
$$= \frac{d}{dt}\frac{\partial T}{\partial \dot{q}_i} - \frac{\partial T}{\partial q_i} + \frac{\partial U}{\partial q_i} \tag{B.5}$$

where τ_i is the generalized force.

Potential energy

The total potential energy consists out of the gravity and the elastic energy. The gravitational energy can be calculated using

$$U_{grav} = \sum_{i=1}^{n} m_i g^T r_{ci} \tag{B.6}$$

The vector $\boldsymbol{g}$ is the vector giving the direction of gravity in the inertial frame thus $\boldsymbol{g} = \begin{bmatrix} 0 & 0 & g \end{bmatrix}^T$ and the vector $\boldsymbol{r}_{ci}$ gives the coordinates of the center of mass of link i.

$$U_{grav}(\boldsymbol{q}_L) = m_2 g a_{c2} S_2 + m_3 g \left[a_2 S_2 + a_{c32} C_3 + a_{c31} S_3 \right] \tag{B.7}$$
$$= (m_2 a_{c2} + m_3 a_2) g S_2 + m_3 g (a_{c32} C_3 + a_{c31} S_3) \tag{B.8}$$

Using the assumption that the torsional spring behaves linearly, the elastic energy can be calculated as

$$U_{elas} = \frac{1}{2}(\boldsymbol{q}_L - \boldsymbol{q}_M)^T \mathbf{K}(\boldsymbol{q}_L - \boldsymbol{q}_M) \tag{B.9}$$

$$= \frac{1}{2}K_1(\theta_1 - \phi_1)^2 + \frac{1}{2}K_2(\theta_2 - \phi_2)^2 + \frac{1}{2}K_3(\vartheta_3 - \phi_3)^2 \tag{B.10}$$

Kinetic energy

The total kinetic energy consists of contributions of the links and rotors.

$$T = T_{link} + T_{rotor} \tag{B.11}$$

For the links the kinetic energy is

$$T_{link} = \frac{1}{2}\dot{\boldsymbol{q}}_L^T \mathbf{D}_L(\boldsymbol{q}_L)\dot{\boldsymbol{q}}_L \tag{B.12}$$

The matrix $\mathbf{D}_L$ is defined by

$$\mathbf{D}_L(\boldsymbol{q}_L) = \sum_{i=1}^{n} \{m_i J_{v_i}(\boldsymbol{q}_L)^T J_{v_i}(\boldsymbol{q}_L) + J_{\omega_i}(\boldsymbol{q}_L)^T R_i(\boldsymbol{q}_L) I_i R_i(\boldsymbol{q}_L)^T J_{\omega_i}(\boldsymbol{q}_L)\} \tag{B.13}$$

The kinetic energy of the rotors can be described by

$$T_{rotor} = \sum_{i=1}^{N} T_{rotor_i} \tag{B.14}$$

$$= \sum_{i=1}^{N} \left(\frac{1}{2}m_{r_i} v_{r_i}^T v_{r_i} + \frac{1}{2}\omega_{r_i}^T I_{m_i} \omega_{r_i}\right) \tag{B.15}$$

So for the motors we obtain

$$T_{rotor} = \frac{1}{2}I_{m_{1zz}}\dot{\phi}_1^2 + \frac{1}{2}I_{m_{2zz}}\dot{\phi}_2^2 + \frac{1}{2}I_{m_{3zz}}\dot{\phi}_3^2 \tag{B.16}$$

Equations of motion

Using (B.5) the equations of motion can be obtained. For the motors this result in the following

$$\tau_1 = I_{m1zz}\ddot{\phi}_1 - K_1(\theta_1 - \phi_1) \tag{B.17}$$

$$\tau_2 = I_{m2zz}\ddot{\phi}_2 - K_2(\theta_2 - \phi_2) \tag{B.18}$$

$$\tau_3 = I_{m3zz}\ddot{\phi}_3 - K_3(\vartheta_3 - \phi_3) \tag{B.19}$$

The equations of motion for the links are given as the Newton-Euler representation.

$$\boldsymbol{\tau}_e - \mathbf{J}^T \boldsymbol{f} = \mathbf{D}(\boldsymbol{q}_l)\ddot{\boldsymbol{q}}_l + \mathbf{C}(\boldsymbol{q}_l, \dot{\boldsymbol{q}}_l) + \boldsymbol{g}(\boldsymbol{q}_l) \tag{B.20}$$

Where the vector $\boldsymbol{\tau}_e$ are the torsional spring torques, the matrix $\mathbf{C}$ contains the centrifugal and Coriolis terms, the matrix is defined as

$$C_{kj} = \sum_{i=1}^{n} \frac{1}{2}\left\{\frac{\partial d_{kj}}{\partial q_j} + \frac{\partial d_{ki}}{\partial q_j} - \frac{\partial d_{ij}}{\partial q_k}\right\}\dot{q}_i \tag{B.21}$$

Appendix C

Gazebo verification experiment

A test case in Gazebo is used to determine if compliance is incooperated properly. This is done by simulating a 2-mass-spring model. The model is a fourth order model with a resonance at

$$f = \frac{1}{2\pi}\sqrt{\frac{k(I_1 + I_2)}{I_1 I_2}}\ \text{[Hz]} \tag{C.1}$$

Using the parameters shown in Table. C.1 an eigenfrequency of 71 Hz is expected. As can be seen in Fig. C.1 the frequency response corresponds with the expectations. It should be mentioned that the position of the joint x_2 is relative to the position of the actuator x_1.

Table C.1: Simulator parameters

Parameter	Value	Dimension
Inertia motor	0.1	[kg m^2]
Inertia joint	0.1	[kg m^2]
Spring stiffness	10000.0	[Nm/rad]

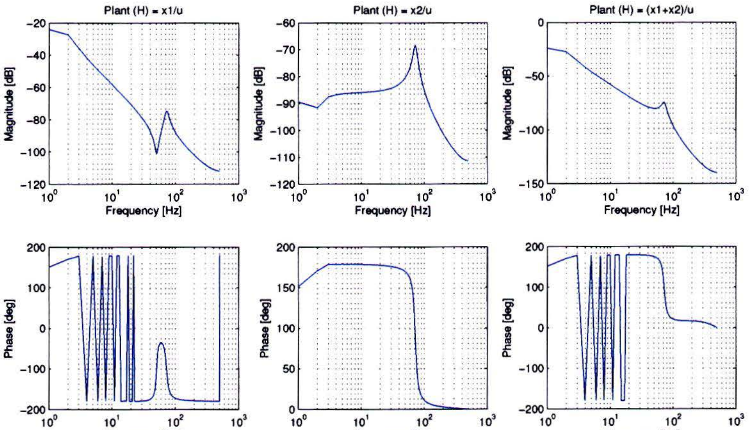


Figure C.1: Frequency response of a 2-mass-spring simulation.