

University of Twente

EEMCS / Electrical Engineering
Control Engineering



Modeling and Gait Control of a Quadruped Robot

Sebastiaan van Loon

M.Sc. Thesis

Supervisors

prof.dr.ir. J. van Amerongen

dr.ir. S. Stramigioli

ir. V. Duindam, dr. M. Poel

October 2005

Report nr. 036CE2005

Control Engineering

EE-Math-CS

University of Twente

P.O. Box 217

7500 AE Enschede

The Netherlands

Summary

Walking quadruped robots can easily stand in a stable position, the difficulty lies within the walking algorithm. Four legged animals have different gaits for different speeds, e.g. crawl, trot, pace and gallop. The generation of fast and stable walking motions requires algorithms that can take into account the dynamical effects and stability issues. A dynamical model of a quadruped robot can be used for generating new motions, with advanced algorithms, e.g. reinforcement learning or evolutionary algorithms. These algorithms need multiple test runs. These test runs on a practical robot have several disadvantages. A model does not suffer from the disadvantages of a robot, which makes it suitable for multiple test runs. In this report a method is presented to develop and verify a multi-body dynamical model of a quadruped robot.

The robot, used in this report, is the ERS-7 Aibo from Sony. It is a four legged robot and consists of 20 actuated joints and 19 body parts. First the design and implementation of the model is presented. Different kinds of contact models with the floor are discussed and a choice is made to use a compliant contact model. The motor specifications are not publicly available, only the speed limits are known and have been included into the model. Finally the mechanical end stops of the joints are modeled by limiting the input signals of the controller.

Some dynamical parameters of the used robot are not publicly available. Several methods are discussed to obtain the missing data, one estimation method is used. The estimation method uses a pendulum model. The model is fitted on measurement data, which is obtained from the real robot. The results, that were obtained using an extensive pendulum model, gave the best estimation of the missing parameters.

The dynamical model is verified with the real robot, to see if the behaviors are comparable. A stable walk is implemented in the model as well as in the real robot. The result shows that the model represents the real robot gait accurately.

Preface

‘It always ends with a new beginning’

This thesis brings my period as a student to an end and a new period starts with challenging opportunities. I would like to thank the people who supported me and made it possible to succeed in ending this wonderful period:

- I would like to thank Mannes Poel who lend us the ERS-7 Aibo. Stefano Stramigioli for his enthusiasm and offering me this great challenge. I also would like to thank Vincent Duindam for his support, the numerous discussions and interest in this project.
- Controllab Products B.V. for their help and support with 20-sim, especially Frank Groen for his help with the new 3D mechanics toolbox.
- The Dutch Aibo Team, in particular Niek Wijngaards. Although my work did not contribute to the actual soccer performance of the Aibo, I was very pleased with all the good advice and practical input.
- The technical support staff, Marcel Schwirtz and Gerben te Riet o/g Scholten for their support with all my technical problems. Carla Gouw-Banse for her help with the problems nobody else could solve.
- The students at the department for the coffee breaks, lunches, interesting talks and laughs.
- My roommates at ‘Campuslaan 67’ and ‘AAAAAAA-Bovenaar’, you were all great!
- Specially I would like to thank all my friends who supported me all this time. I hope our friendship continues with this new beginning!
- Las but not least, I would like to thank my family, for their interest in my study progress, Christmas dinners and support. My father and mother for their endless support, love and good advice.

Sebastiaan van Loon,
Enschede, September 2005

Contents

1	Introduction and Problem Definition	1
1.1	Introduction	1
1.2	Problem Definition	2
1.3	Outline	2
2	Modeling of a Quadruped Robot	3
2.1	Simulation Software	3
2.2	Kinematics	3
2.3	Dynamics	4
2.3.1	Contact Model	5
2.3.2	Joint Limits and Motor Speeds	7
2.4	Controlling Motions	8
3	Parameter Estimation of the Aibo	9
3.1	Measuring Method	9
3.2	Estimation using a Simple Pendulum Model	10
3.2.1	Experimental Setup	10
3.2.2	Experimental Results	11
3.3	Estimation using an Extensive Pendulum Model	13
3.3.1	Experimental Setup	14
3.3.2	Experimental Results	14
3.4	Choice of Leg Parameters	15
4	Verification of the Aibo Model: using a Walking Pattern	17
4.1	Walking Gaits	17
4.2	Test Case: Walking Aibo	18
4.2.1	Setup Test Case	18
4.2.2	Results Test Case	19
4.3	Aibo Model versus Real Aibo	22
5	Conclusions and Recommendations	23
5.1	Conclusions	23
5.2	Recommendations	24
	Bibliography	28
A	Kinematical and Dynamical Properties of the ERS-7 Aibo	29
B	Sensor and Controller Specifications	33
B.1	Measurements with Position Sensors	33

C	Walking Results	37
C.1	Long Simulation Walk	37
C.2	Walking without the Floor	38
C.3	Higher Frequency Walking Patterns	38
C.4	Other Integration Methods	38

Chapter 1

Introduction and Problem Definition

1.1 Introduction

Robots are used for specific repetitive motions in many different areas and in different environments, e.g. assembly and manufacturing plants, explorations in seas and outer space. Robots outperform humans on speed, accuracy and flexibility. Their disadvantages are the lack of capability to think independently, to collaborate with other robots, to learn from mistakes and to adapt to changes in their surroundings. Many walking robots are currently only capable of walking on a level floor with a known surface. Any changes to the surface will often make the robot unstable or incapable to walk, unless it adapts its walking gait. Robots should be able to adapt their walking pattern and direction, because of external changes like wear of the motors, the texture of the ground and moving objects.

To develop new walking algorithms, a walking quadruped robot is used from Sony Corporation [Sony Corporation, 2005b], called the ERS-7 Aibo. Aibo means ‘Friend’ in Japanese and is a commercial product, built to entertain people in day-to-day life. Aibo is a mature toy with many different sensors and actuators, as can be seen in Figure 1.1. It is used in many different research areas such as vision, robotics, artificial intelligence and embedded systems. The Aibo is an ideal platform to test new algorithms and ideas, because Sony is actively promoting the standard interface software “OPEN-R” [Sony Corporation, 2005a] for programming the robot. To compare these new algorithms and ideas, research groups around the world are validating their work in RoboCup’s 4-Legged Soccer Leagues [The Robocup Foundation, 2005]. The goal of this competition is to exchange knowledge and to promote robotics and artificial intelligence research. Another goal is to let the Aibo play soccer as realistically as real human soccer. This means that the robots play in a real environment, with e.g. real ground texture, wear of the motors and collisions with other robots.

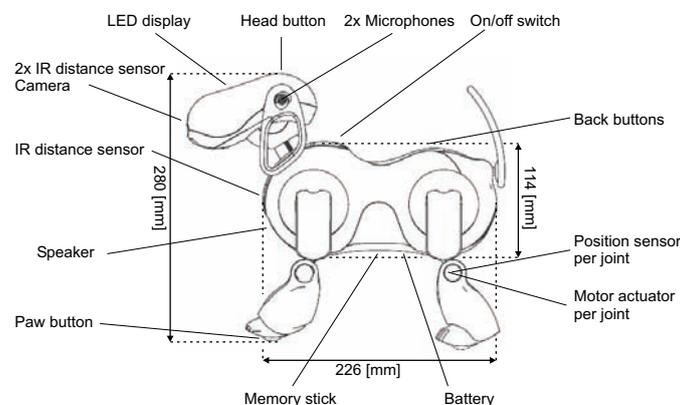


Figure 1.1: The physical ERS-7 Aibo with sensors and actuators [Sony Corporation, 2005a]

One of the teams that participates in the RoboCup is the Dutch Aibo Team [Dutch Aibo Team, 2005]. This group is a cooperation between the Decis-Lab and the universities of Amsterdam, Eindhoven, Delft, Groningen, Twente, Saxion Hogescholen and Utrecht. It was founded in 2004 and its main focus is on modeling and building collaborative autonomous intelligent systems. This report is part of the Dutch Aibo Team.

1.2 Problem Definition

The difficulty with walking robots is the creation and optimization of the walking pattern. The many degrees of freedom makes it difficult to create a walking pattern that is stable and fast. Implementing a walking pattern can be done in different ways. It can be tuned by hand, which is time consuming and the result is stable but slow. Another option is to generate walking patterns with algorithms that are able to adapt to changes of the environment and can be optimized to be stable and fast. In order to test and develop such algorithms there are two options. The algorithm can be implemented on the real Aibo, or the algorithm can be tested on a computer, using a model of the Aibo that simulates its behavior.

The implementation on the Aibo has the advantage of the direct usage of the real environment and the dynamics of the Aibo. Developing new algorithms is a slow process, since the software should be compiled and put on the memory stick. The interface of the Aibo is limited with some leds and a speaker, and is difficult to use for debugging. A limiting factor for optimization problems is the onboard CPU, which makes it difficult to find the solution within a reasonable time. Other disadvantages are the limited power of the battery and the failures that can occur due to the hardware (jams of the legs) and the software. Successful studies have been done on generating motions with evolutionary algorithms calculated on a real Aibo [Röfer, 2004] [Düffert and Hoffmann, 2006].

The other option is to optimize the walking patterns based on a multi-body dynamical model, like Stelzer et al. [2003]. With a dynamical model of the Aibo, it is possible to design new motions on a computer, without the use of a real Aibo and its disadvantages. The model can be used with advanced learning algorithms to generate robust walking patterns. For the RoboCup competition, the model could be used to implement new motions for handling the ball and creating efficient kicks and defence moves.

The goal of this thesis is to develop a multi-body dynamical model of the ERS-7 Aibo, which can be used to develop stable motions and optimal walking gaits. The model has to give a realistic representation of the real Aibo motions and will need the same dynamical parameters. The model must be verified with the real Aibo. The objectives of this research are to:

- Design a multi-body dynamical model of the ERS-7 Aibo
- Estimate the missing dynamical parameters
- Verify the model

1.3 Outline

The outline of this report is as follows. Chapter 2 describes the dynamical model of the ERS-7. First the kinematics are shown and the dynamical factors are given as well as the influences of the environment on the robot. Several parameters in the created model have to be estimated and two methods are used to obtain these values in Chapter 3. The model is then tested with a walking motion in Chapter 4. It is shown that the dynamical model of the Aibo is walking just like the real Aibo. Finally in Chapter 5, conclusions of this research are drawn with recommendations for further research.

Chapter 2

Modeling of a Quadruped Robot

Modeling the Aibo has been done in different studies for different purposes, see for example Stelzer et al. [2003], Hengst et al. [2002] and Hardt and von Stryk [2002]. These models are based on the older version of the Aibo, the ERS-210. The new version, now used in the RoboCup's 4-Legged Soccer Leagues, is the ERS-7. This new robot is bigger, the CPU is faster and its motors are improved with respect to the ERS-210. The models in previous studies were developed to optimize the gait of the ERS-210 and the implementations of the floor contact were different. Some studies used a paw motion, where the leg was attached to the floor without friction [Hengst et al., 2002]. Other studies modeled the ground contact with inelastic collisions, resulting in instantaneous jumps in the state velocities [Hardt and von Stryk, 2002] [Stelzer et al., 2003]. The approach in this report will describe the floor influences in a continuous manner, as well as the new kinematic layout of the ERS-7.

In Section 2.1, the choice of the simulation software is discussed. Then, the kinematics of the Aibo are given in Section 2.2, followed by the dynamics in Section 2.3. In the last section, the controller is shown with its sensors and actuators.

2.1 Simulation Software

The dynamics of a quadruped robot model needs simulation software that can model the kinematics, dynamics and the constraints. There should be an option to easily optimize the parameters as well as the possibility to animate and show the results of the simulation. Finally, the resulting model should be exported to other software packages, in C-code or Matlab-code.

Modeling of dynamical structures is possible in many programs. For working with movies and 3D animations, Blender [Blender, 2005] is a good option and can be linked with the programming language Python. For simulation with rigid bodies the Open Dynamics Engine [Open Dynamics Engine, 2005] is a good program and has the advantage that it is open source. For fast prototyping and simulation the commercial software package Webots [Webots, 2005] could be used, which is linked with the Open Dynamics Engine. For simulations and dynamics the simulation package 20-sim [Control Lab Products, 2005] can also be used. This program has a 3D mechanics toolbox for modeling multi-body mechanisms as port-Hamiltonian systems. The model can be exported to Matlab-code or C-code. The choice was made to use 20-sim for modeling the Aibo, because the support for modeling multi-body mechanisms is very intuitive and the implementation of constraints is easily done. The software uses real-time code generation and the correctness of the results are verifiable. The software is developed at the University of Twente, which gives a good support for debugging and implementing the models.

2.2 Kinematics

The ERS-7 Aibo is a quadruped robot. As can be seen in Figure 2.1, it has three degrees of freedom per leg, three degrees of freedom for its head movement, two degrees of freedom for its tail and one degree

of freedom for its mouth. Each ear has two joints of which one is actuated and the other is passive. In total, the dynamical model of the ERS-7 has 19 body parts and 20 actuated joints.

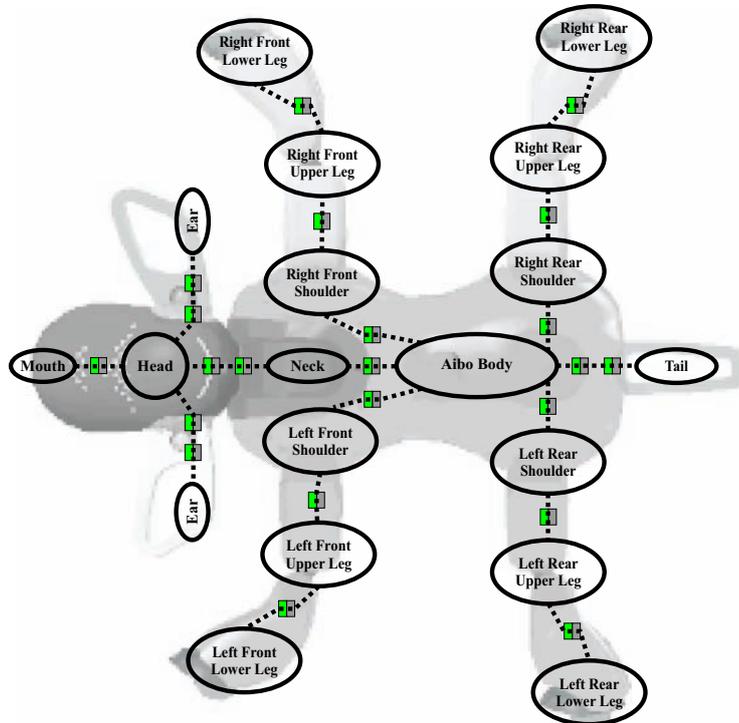


Figure 2.1: Sketch of the body parts and joints of the ERS-7 Aibo

The Aibo kinematics are given in the manual from the OPEN-R SDK [Sony, 2004] together with the kinematic model for Java, as Light Wave Objects, also provided by Sony. This documentation is used to make a model in the 3D Mechanics Toolbox of 20-sim. This toolbox can generate a numerical model that can be used in the simulation software of 20-sim.

In Appendix A several tables are given with the specific data. The relative positions between the bodies are given in Table A.1. The physical properties of each body part are shown in Table A.2. The lengths are measured and verified with the data from OPEN-R, whereas the masses and inertias are estimated as given in Chapter 3.

2.3 Dynamics

The kinematics of the Aibo only provides information about the geometrical relation between the body parts. The 3D Mechanics Toolbox also includes the direction of the gravitational force, the inertias and masses. Still, this model is not sufficient to describe a walking behavior, since the interaction with the floor and mechanical end stops are not implemented and the joints are not actuated. These external parameters directly influence the dynamical behavior of the model.

The numerical model produced by the toolbox is used in 20-sim. The interaction between body parts and environment is calculated within the numerical model by using screw theory [Stramigioli and Bruyninckx, 2001]. The inputs and outputs of the numerical model are shown in Figure 2.2. The steering of the joints as well as the floor contact are defined with bond graphs. The bond graph language describes the power flowing from one element to another. In this case the joints are steered with a controller that provides a torque. The masses of the bodies attached to the joint will influence the angular velocity. The forces and torques from the floor contact are combined as a variable called a wrench. The velocities

(linear and rotational) are combined in a six dimensional variable, called a twist. The position and orientation of each body part are given as a homogeneous transformation matrix (H-matrix). The H-matrix represents a coordinate transformation between two coordinate frames, ψ_i and ψ_j , denoted by H_j^i . Further information on bond graphs can be found in Breedveld and van Amerongen [1994] and information about wrenches, twists and H-matrices can be found in Stramigioli [2001].

The interface of the real Aibo allows the user only to set the joint angle instead of the torque, as shown in Figure 2.2. This angle is used as a setpoint for a controller. The joint angle is measured with the sensors inside the joints. This implies that the overall model will also include controllers. This topic is further discussed in Section 2.4.

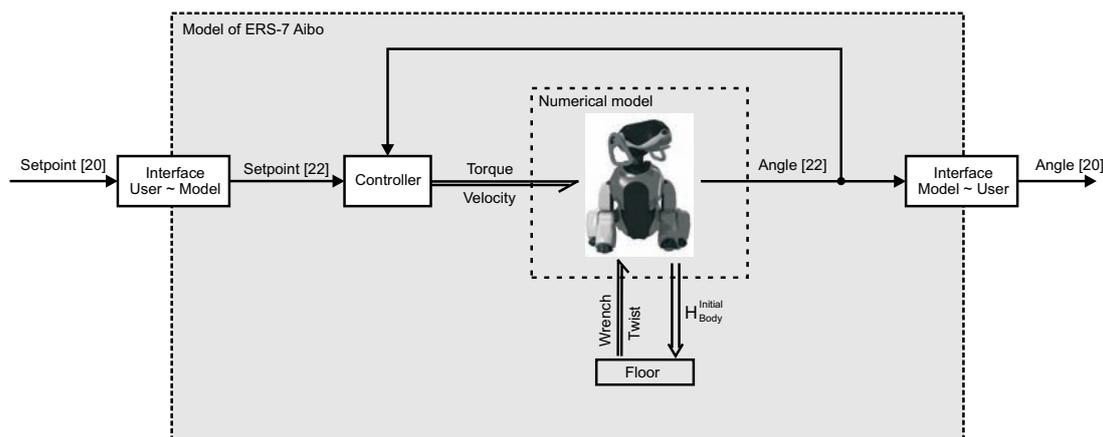


Figure 2.2: Block diagram of the model of the Aibo with the kinematics, dynamics and constraints

The movement of the Aibo is restricted by its environment in several ways. First of all the gravitational force pushes the robot downwards, while the floor prevents it from falling. The joints are limited by mechanical end stops, which constrain the joint from reaching an arbitrary angle. The friction of the floor prevents the Aibo from sliding and makes it possible to let the Aibo walk. All these constraints have to be included in the model for a realistic representation of the Aibo. In the following section these constraints are discussed.

2.3.1 Contact Model

The interaction with the floor is important for walking. The stability and speed of the walking pattern is influenced by the floor, depending on how uneven, sticky or soft the material is. For modeling the floor, the characteristics should be defined and the contact area of the Aibo with the floor should be specified. In the RoboCup the Aibo walks on a carpet, but the characteristics of the floor are not specified and differ per event. Every team has its own use of the legs, which gives a wide variety of contact areas on the feet. Some teams use the front legs of the Aibo flat on the floor to maintain stability and the rear legs to move around, others have all four feet flat on the floor. Sony uses the legs upright, which is more unstable and looks more doglike. These qualities should be taken into account when designing a dynamical model for making new walking patterns.

To model a standing pose, a power interaction needs to be modeled between the floor and the legs. In Figure 2.3 an illustration is given with the forces and torques that can interact with a body. The collision can occur on a fixed contact point on the body or the contact point can move over the body depending on its orientation.

Compliant or rigid contact models

The contact dynamics can be modeled as a compliant or rigid contact. Compliant contacts assume that when two objects collide, the contact phase is not instantaneous. First the objects will touch each other,

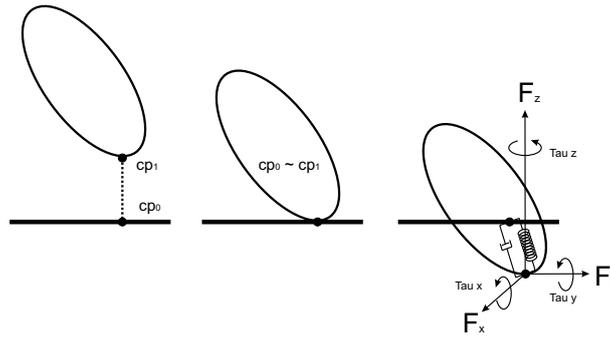


Figure 2.3: On a collision of a body and the floor, forces act on the contact points. The normal force is modeled with a spatial spring and damper

then deform a bit and then bounce off each other, as described in Duindam and Stramigioli [2003]. This can be modeled with a spring and damper acting on the foot when it touches the floor. This spring is based on a spatial spring (as discussed in Stramigioli and Duindam [2002]) and makes a distinction between rolling and sliding. Rigid contacts, as described in Duindam [2004], means that there is an instant contact. This model implies that on impact some kinematic energy is instantaneously lost and the momentum on impact changes stepwise.

For the contact model of the Aibo, the use of a rigid contact model is hard to implement, because the model has multiple contact points. This diversity in contact states needs a lot of book-keeping and recalculations of the momentum on impact. The compliant contact model can be used in a continuous and easy way, which makes it a good option for the implementation of the contact model.

Implementation of a compliant contact model

When an object collides, there is an exchange of energy. This energy can be shown using a bond graph, with a force as an effort and a velocity of the object as a flow. In a three-dimensional space the force can be divided in three forces and three torques. Combining this in a vector results in a wrench. A wrench acting on frame ψ_i can be represented by a six-dimensional vector:

$$W^i = [F_x, F_y, F_z, \tau_x, \tau_y, \tau_z] \quad (2.1)$$

where F_x , F_y and F_z [N] are the forces and τ_x , τ_y and τ_z [Nm] the torques. The flow is called a twist and includes the velocities of frame ψ_i with respect to frame ψ_j :

$$T_i^{i,j} = [v_x, v_y, v_z, \omega_x, \omega_y, \omega_z] \quad (2.2)$$

where v_x , v_y and v_z [m/s] are the velocities in the x-, y- and z-direction and ω_x , ω_y and ω_z [rad/s] are the angular velocities around the x-, y- and z-axis. More information about the use of wrenches and twists are given in [Stramigioli, 2001].

The compliant contact model is implemented in the box marked ‘Floor’ in Figure 2.2. To apply the forces, first the moment when the body hits the floor should be known. This is checked with the distance between the contact points of the body (cp_1) and the floor (cp_0), given in Figure 2.3. The numerical model can give the H-matrix, H_{body}^{floor} , which gives information about the distance between the body and the floor. With this H-matrix, the shape of the body and the assumption that the floor is flat, the distance between cp_0 and cp_1 is known. If the contact point hits the floor, it will apply forces as a wrench to the body part that makes contact.

The forces that act on the contact point of the body are divided into a normal force and two tangential forces. The normal force (F_n) is calculated with the Hunt-Crossley model given by Hunt and Crossley [1975]. The model is nonlinear and suitable for contacts with both stiff and compliant objects. It prevents physical inconsistencies on impact and sticky forces at the moment of load removal that occur with a linear

contact model like the Kelvin-Voigt model (also given in Hunt and Crossley [1975]). The Hunt-Crossley model is formulated as:

$$F_n(t) = \begin{cases} -K_p x^n(t) - K_d x^n(t) \dot{x}(t) & x \geq 0 \\ 0 & x < 0 \end{cases} \quad (2.3)$$

where x [m] is the penetration depth, \dot{x} [m/s] is the speed of the body in the vertical z-direction, n is a real number, depending on the shape and material of the object. n is usually close to the unity. K_d [Ns/m] is a stiff damper and K_p [N/m] is a stiff spring, both of which are applied when the body hits the floor.

When walking, the floor also exerts a force in the horizontal plane. This can be modeled with different friction models. The options to use as a damper are Coulomb, viscous, static and Stribeck friction. Coulomb friction is an offset depending on the direction of the motion. The viscous friction is a linear relation between the velocity and the friction force. The static friction opposes the direction of motion when the sliding velocity is zero. Stribeck friction decreases continuously with increasing velocities. More about friction models can be found in Armstrong-Héouvy et al. [1994]. As for the friction of the feet, a friction model with all four friction models is chosen. The friction force (F_{fr}) at the contact point depends on the normal force and the tangential speed (v_{tr}) of the contact point. v_{tr} is a combination of the velocity v_x and v_y in the x and y direction:

$$v_{tr} = \sqrt{v_x^2 + v_y^2}$$

$$F_{fr} = -F_n \left(\left(\mu_c + (\mu_{st} |\tanh(s v_{tr})| - \mu_c) e^{-\left(\frac{v_{tr}}{v_{st}^2}\right)} \right) \text{sign}(v_{tr}) + \mu_v v_{tr} \right) \quad (2.4)$$

where μ_c [m] is the Coulomb friction coefficient, μ_{st} [m] is the static friction coefficient, μ_v [s/m] is the viscous friction coefficient, v_{st} [m/s] is the characteristic Stribeck velocity and s [s/m] is the steepness of the Coulomb friction curve. The force F_{fr} needs to be distributed over the F_x and F_y axis and the total force that acts on the contact point, becomes:

$$\begin{aligned} F_x &= F_{fr} \left(\frac{v_x}{v_{tr}} \right) \\ F_y &= F_{fr} \left(\frac{v_y}{v_{tr}} \right) \\ F_z &= F_n \end{aligned} \quad (2.5)$$

Rolling of an object is modeled with a pure rotation around the contact point. The rotations around the x- and y-axis are frictionless, while the roll around the z-axis provides a torque, which is calculated in the same way as above but depends on the rotational speed (ω_z):

$$\begin{aligned} \tau_x &= 0 \\ \tau_y &= 0 \\ \tau_z &= -F_n \left(\left(\mu_c + (\mu_{st} |\tanh(s \omega_z)| - \mu_c) e^{-\left(\frac{\omega_z}{v_{st}^2}\right)} \right) \text{sign}(\omega_z) + \mu_v \omega_z \right) \end{aligned} \quad (2.6)$$

2.3.2 Joint Limits and Motor Speeds

The joint ranges of the ERS-7 are limited, because they have mechanical end stops. These end stops need to be implemented into the model. This can be done with a nonlinear one-dimensional spring-damper element for each joint, in much the same way as the compliant contact model described in Section 2.3.1. Using a compliant contact model as a mechanical end stop for all 20 actuated joints, means an enormous increase in simulation time due to the resulting stiff equations. Another option is to limit the input values from the controller. This saves simulation time, but will not give a realistic representation when the controllers are turned off. The last option is chosen because it is assumed that the controllers are always on. The limits from the mechanical joints of the Aibo are given in Appendix A in Table A.3, together with the software limitations built in OPEN-R software.

The Aibo motors are limited by their power and maximal accelerations. The model should have a protection or warning system when these limits are crossed. OPEN-R provides only the angular velocity limits (also given in Table A.3 in Appendix A), which can be used in the model. The model generates a warning when a joint exceeds its speed limit, but it cannot check if any acceleration or power limits are exceeded, since the limits are not available.

In the model the actuator characteristics are not included, since the data were not freely available, as stated on the website of OPEN-R [Sony Corporation, 2005a]. The output of the controller in this model is converted to a torque (see Figure 2.4, ‘MSe’), which implies a current steering signal. The friction in the joint is modeled with a dissipative element, depicted as ‘R’ in Figure 2.4. The specific friction model is estimated in Chapter 3.

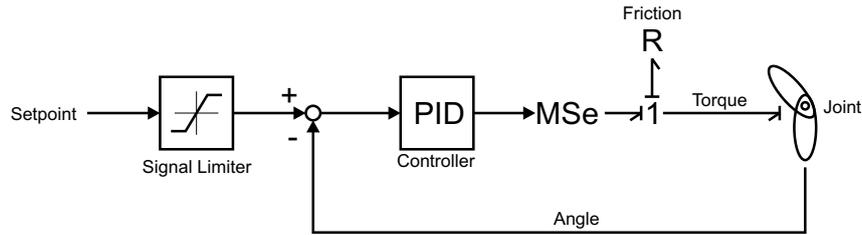


Figure 2.4: Control for a signal joint, within the Aibo model

2.4 Controlling Motions

The joints of the Aibo model are steered with bond graphs. The effort is given as the torque and the flow as angular velocity. In the ERS-7 the joints are controlled by PID (proportional, integral and differential) controllers, with the angle given as setpoint. In Figure 2.4 an overview is given of a controller for a signal joint. The Aibo works with time frames of 8 [ms] each. In each frame, a motion can be set and the controller reads the sensor values.

The setpoint is given by the software, this signal is limited to angles within the hardware limits as explained in Section 2.3.2. The actual angle is measured with a position sensor. With this measurement an error signal is created and is used by a PID controller. The specifications of these sensors are not given by OPEN-R, but some tests have been done to estimate the accuracy of the sensor (see also Appendix B). The step size of the sensor is approximately 0.006 [rad] and has a variation of 0.012 [rad]. The step size indicates that the position sensor is a 10 bits AD converter. The sampling time is 8 [ms], more information about the sensors is given in Table B.1 in Appendix B.

The PID controllers are embedded into the hardware of the Aibo. It is possible to change the proportional, integral and differential gain of the controller with the software. There are some limitations within OPEN-R, because of the limited precision floating point numbers. Each gain is split up into a gain (g) and a shift (s) component, in the following way:

$$\text{gain} = \frac{g}{2^{16-s}} \quad (2.7)$$

More information can be found on the website of Tekkotsu [Carnegie Mellon University, 2005]. Default values are given in Table B.2 in Appendix B. The shift values are the same for each gain of each controller. This relation of the gains makes it impossible to adjust separate gains independently. The gains can only be chosen within a certain range, fixed by s .

Chapter 3

Parameter Estimation of the Aibo

The model that is constructed in the previous chapter combines the capabilities and limitations of the Aibo. A lot of information is known about the ERS-7 Aibo, e.g. the kinematics and its total mass [Sony Corporation, 2005a]. Several parameters are not publicly available from Sony, due to commercial reasons. This disclosed information contains masses, center of mass, inertias and motor characteristics.

To develop a realistic dynamical model, it is important to have insight in these dynamical parameters. The Aibo is a fully controlled robot, which means that all its joints are actively actuated (except some of the ear joints). There is no need to estimate the exact values: an indication of the real values is enough to construct the model. In the next sections an experimental procedure to provide the missing parameters is explained.

3.1 Measuring Method

The Aibo is used as a scientific platform and has several possibilities to get data from its environment and physical states. The Aibo has position sensors, acceleration sensors, a camera and switch buttons to obtain data. The position sensors have a sampling time of 8 [ms], while the movements are restricted by the motors and do not exceed a couple of Hertz. Therefore the sampling frequency is fast enough to use for controlling the walking gaits.

The Aibo has a hardware layer that Sony does not reveal. The hardware is available through the OPEN-R SDK, which is a cross development environment. It is able to work with different hardware modules (ERS-X), it is easy to include new software parts and it has network support.

On top of OPEN-R, several platforms have been developed to develop new software. There are some developed by Sony, e.g. R-CODE SDK and AIBO Remote Framework. These programs are easy to use, but lack the possibilities to exploit the hardware of the Aibo. Therefore several RoboCup teams decided to develop their own environment, e.g. Tekkotsu [Carnegie Mellon University, 2005], URBI [Baillie, 2005] and the German Team Code [German Team, 2005].

Tekkotsu is different, because it is designed to develop new software instead of focussing on playing soccer. It works with parts of software that can be added to the platform. When carrying out tests on the Aibo it has the advantage of OPEN-R and the functionalities of a platform on a higher software level. Tekkotsu has the functionalities to test motions, positions and has a wireless interface to measure data and choose between different programs. For these reasons, Tekkotsu is used in this project as a measurement platform.

3.2 Estimation using a Simple Pendulum Model

The best way to measure parameters like masses, inertias and friction, is to measure each of the body parts separately or by using the design drawings. It was not possible to carry out those two options¹, so another option was chosen to estimate these parameters. The masses were roughly estimated first with a scale, to get an indication of the proportions. The center of mass per body part was chosen in the middle of every body part, since it was not possible to measure them separately. The moments of inertia can be calculated when the masses of each body part are known. The inertias are found by assuming the body parts to be perfect ellipsoids and by using Weisstein [2005]:

$$\begin{aligned} I_x &= \frac{1}{5} M (y^2 + z^2) \\ I_y &= \frac{1}{5} M (x^2 + z^2) \\ I_z &= \frac{1}{5} M (x^2 + y^2) \end{aligned} \quad (3.1)$$

where M [kg] is the mass of the body, I_x , I_y and I_z [kg m²] the moments of inertia around the x-, y- and z-axis, and x , y , z [m] are the radii of the ellipsoid. The mass per body part and the friction per joint can be estimated using a pendulum model. The first attempt is to estimate them with a simple model of a pendulum, only taking a point mass and linear friction. The experimental setup and the obtained results are discussed in the next sections.

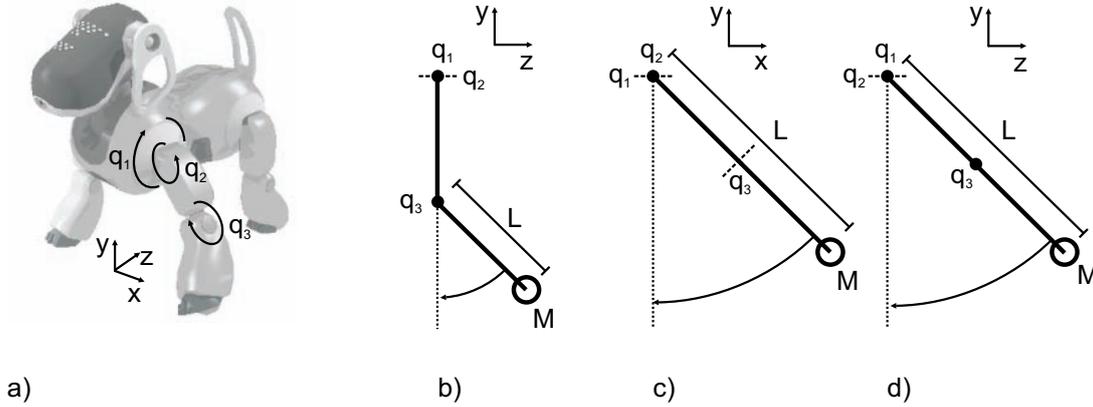


Figure 3.1: The leg of the Aibo can be seen as different pendulum models: a) Overview of the left front leg of the Aibo, with the positions of the joints. b) Pendulum model of the lower leg around joint q_3 . Joints q_1 and q_2 are fixed c) Pendulum model of the upper leg, combined with the lower leg, rotating around joint q_2 . Joints q_1 and q_3 are fixed. d) Pendulum model of the whole leg (shoulder, upper leg and lower leg), rotating around joint q_1 . Joints q_2 and q_3 are fixed

3.2.1 Experimental Setup

The masses and friction are estimated using the possibility to turn the controller gains off. Then only the gravitation force acts on the joints and the movement is compared with a simple model of a pendulum as given in Figure 3.1 a), b) and c). Every joint in the leg is tested. The controller gains are turned off when the joint is set to a specific angle. The leg swings as a pendulum. The positions of the joints are measured with the position sensors inside the joints. The same test is done with a simple pendulum model. This model represents a point mass at a certain distance of a joint. The model is fitted to the measurements by adjusting the coefficient $\frac{B}{ML^2}$, given in the pendulum equation:

$$\ddot{\theta} + \frac{B}{ML^2} \dot{\theta} + \frac{g}{L} \sin(\theta) = \frac{1}{ML^2} \tau \quad (3.2)$$

¹For this thesis a ERS-7 Aibo was made available from the department of Computer Science of the University of Twente. This Aibo could not be disassembled, because it was on loan. Sony was approached by several RoboCup teams to share the design drawing or missing data, but they turned down the requests.

where M [kg] is the mass of the pendulum, L [m] is the length of the pendulum from the joint till the center of mass, B [Ns/m] is the friction coefficient, g [m/s²] is the gravity constant, τ [Nm] is the external force from the motor and θ [rad] is the angle of the pendulum. Angle θ can be measured with the position sensors, the gravity constant g is known as well as the estimations of the lengths to the center of mass L . The mass M and the friction coefficient B are the only unknown factors. With two different ratios of $\frac{B}{ML^2}$ the two unknown parameters can be solved. To get an extra ratio an extra mass is added to the lower leg of the real Aibo as well as in the model. The friction will not change and the mass M of the pendulum itself is still unknown.

The coefficients are estimated by fitting the pendulum model to the measurements. This results in two different values, V_1 and V_2 , for the ratio $\frac{B}{ML^2}$, given as:

$$V_1 = \frac{B}{(M + M_1)L^2} \quad (3.3)$$

$$V_2 = \frac{B}{(M + M_2)L^2} \quad (3.4)$$

where M_1 and M_2 are the known extra weights added to the leg (M_1 or M_2 is zero if no extra weight is added). This gives two equations, with two unknown parameters, M and B , which can be solved as:

$$M = \frac{M_1V_1 - V_2M_2}{V_2 - V_1} \quad (3.5)$$

$$B = V_1L^2V_2 \frac{M_2 - M_1}{V_1 - V_2} \quad (3.6)$$

Unfortunately, the lower leg of the Aibo does not move at all with only the gravitational force. This is due to the friction within the joint q_3 . Therefore the first test is done with an extra mass added to the lower leg (which makes M_1 unequal to zero).

The measurement results are used to estimate the ratios $\frac{B}{ML^2}$ and $\frac{g}{L}$ of Equation 3.2. The best fit is found when the error between the measurement and the model is minimal. The criterion that is minimized is:

$$J(t) = \int_0^t (y_{real}(s) - y_{model}(s))^2 ds \quad (3.7)$$

where t [s] is the time, $y_{real}(s)$ [rad] is the measured angle of the joint and $y_{model}(s)$ [rad] is the simulated angle of the joint.

3.2.2 Experimental Results

The pendulum model is fitted on the experimental results by minimizing the criterion of Equation 3.7. The experimental results and the fitted pendulum model are shown in Figure 3.2. The fitted model is not exactly the same due to the simplified model of the pendulum that is used. The model of the pendulum contains a viscous (linear) friction (B) and has masses, but no inertias. In reality also the static, Coulomb and Stribeck friction are involved and the swing depends on the inertias of several ellipsoids connected to each other. These simplifications can inhibit a good estimation if the difference between the model and the real Aibo is too large.

The minimum of Equation 3.7 is found by changing parameter $\frac{B}{ML^2}$. Two known masses are used, one of 0.220 [kg] and one of 0.402 [kg]. The results of the estimation of $\frac{B}{ML^2}$ is given in Table 3.1. The estimated friction is approximately 0.024 [Ns/m], but some of the estimated masses are negative, which is physically impossible. This result is explicable, because Equation 3.5 is sensitive to small deviations in the estimation, as can be seen in Figure 3.3. Only a small region provides a positive mass, so a measurement error or estimation fault will have a major influence on the calculated mass. The estimation of the friction cannot be negative because M_2 and V_1 are bigger than M_1 and V_2 .

This result cannot be used and shows that another approach should be taken. First of all the pendulum model is too simplistic. The point mass should be replaced by several ellipsoids with inertias and the friction model should be made more extensive. In the next paragraph another approach is used to find the missing parameters.

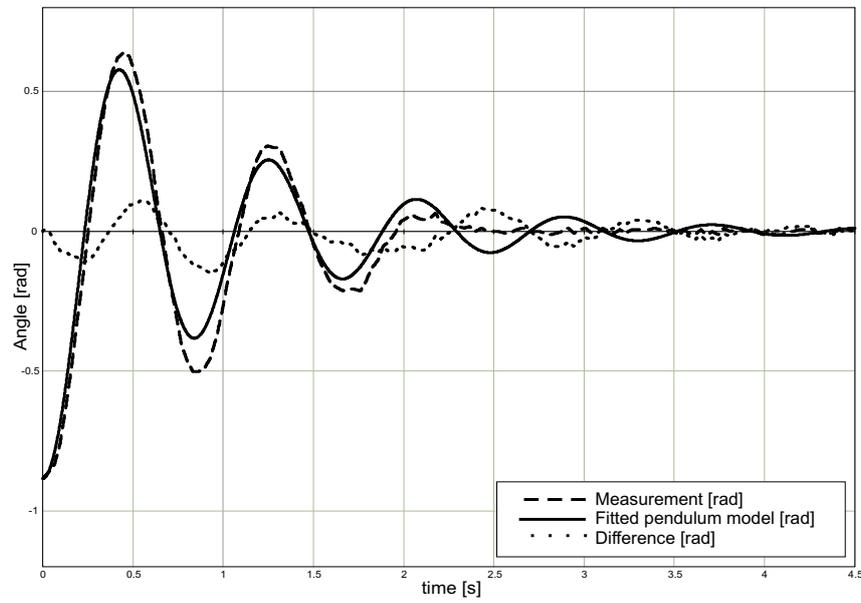


Figure 3.2: Measurement data of the right front shoulder and a fitted pendulum model

Table 3.1: Results after estimation of the masses and frictions of several joints, using Equations 3.5 and 3.6

joint	Test 1			Test 2			Estimated	
	Mass [kg]	$\frac{B}{ML^2}$ [Ns/kgm ²]	$\frac{g}{L}$ [s ⁻²]	Mass [kg]	$\frac{B}{ML^2}$ [Ns/kgm ²]	$\frac{g}{L}$ [s ⁻²]	Mass [kg]	Friction [Ns/m]
q_1	0.220	3.283	60	0.402	1.975	60	0.056	0.024
q_2	0.220	4.005	60	0.402	2.135	60	-0.012	0.022
q_3	0.220	8.387	75	0.402	4.058	75	-0.049	0.025

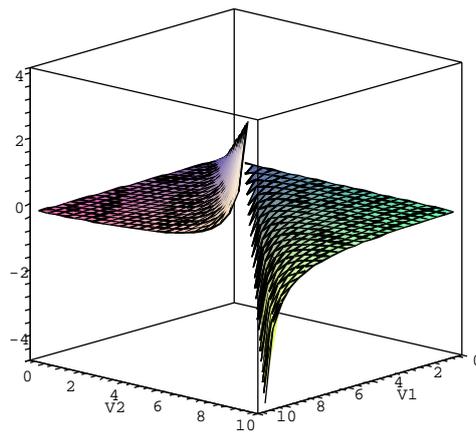


Figure 3.3: Plot of Equation 3.5. Region where mass is positive is limited and very sensitive

3.3 Estimation using an Extensive Pendulum Model

In the previous section the missing masses and frictions were estimated with a simple pendulum model and with the use of ratios. The real Aibo leg is not equal to a point mass without inertias. The leg is equivalent to three ellipsoids which have masses and moments of inertia. Also the extra mass, added to the lower leg in order to give it a better swing, can be modeled as a hollow cylinder. In Figure 3.4 a schematic overview is given of the shoulder (S), upper leg (U), lower leg (L) and extra mass (E). The inertias are given by:

$$\begin{aligned} I_{S,x} &= \frac{1}{5} M_S (y^2 + z^2) \\ I_{U,x} &= \frac{1}{5} M_U (y^2 + z^2) \\ I_{L,x} &= \frac{1}{5} M_L (y^2 + z^2) \\ I_{E,x} &= \frac{1}{2} M_E R^2 + \frac{1}{12} M_E L_E^2 \end{aligned} \quad (3.8)$$

where $I_{S,x}$, $I_{U,x}$, $I_{L,x}$ and $I_{E,x}$ [kg m²] are the inertias in the direction of the x-axis of, respectively, the shoulder, the upper leg, the lower leg and the extra mass added to the leg. M_S , M_U , M_L and M_E [kg] are the masses of the ellipsoids, y and z [m] are the radii of the ellipsoids, R [m] is the radius of the extra mass and L_E [m] is the length of the extra mass. On the real Aibo the joint q_2 turns around the z-axis, but for this model it does not matter which axis is used since the bodies are symmetrical. Therefore q_2 in the model turns over the x-axis.

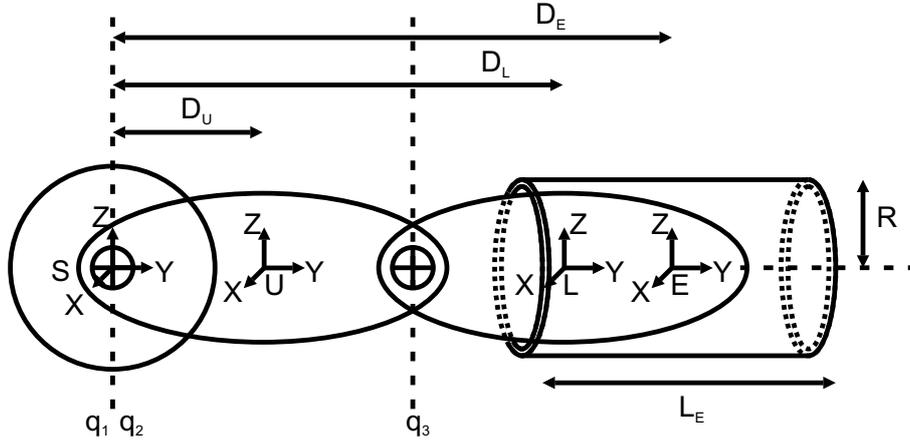


Figure 3.4: Schematics of the Aibo leg, given with ellipses and cylinders

The origin of the rotation axes of the ellipses differ from the centers of mass of the ellipses. Therefore the inertias are recalculated to the rotation axis with the parallel axis theorem. This can be given as:

$$\begin{aligned} I_{U,j} &= I_{U,x} + M_U D_U^2 \\ I_{L,j} &= I_{L,x} + M_L D_L^2 \\ I_{E,j} &= I_{E,x} + M_E D_E^2 \end{aligned} \quad (3.9)$$

where $I_{U,j}$, $I_{L,j}$ and $I_{E,j}$ [kg m²] are the new inertias recalculated for the rotation axis in the joint. D_U , D_L and D_E [m] are the distances between the center of mass of an ellipsoid and the rotational axis at the joint. Only the inertia of the shoulder has not been changed, because the turning axis is already through the center of mass.

These inertias are summed as one moment of inertia that acts on the joint as well as the total mass, which is also shifted:

$$\begin{aligned} I_{joint}^* &= I_{S,j} + I_{U,j} + I_{L,j} + I_{E,j} \\ M_{joint}^* &= M_S D_S + M_U D_U + M_L D_L + M_E D_E \end{aligned} \quad (3.10)$$

where I_{joint}^* [kg m²] is the total inertia acting on the joint, M_{joint}^* [kg m] is the total mass that acts on the joint. For each joint the same tests are performed as in Section 3.2.1 to estimate the mass of the body part(s) and the friction in the joint. The distances D_L and D_E can be different per joint.

The friction in the joint is modeled in the same way as Equation 2.6, given as:

$$F_{friction}(\dot{\theta}) = F_n \left(\left(\mu_c + (\mu_{st} |\tanh(s\dot{\theta})| - \mu_c) e^{-\left(\frac{\dot{\theta}}{v_{st}}\right)} \right) \text{sign}(\dot{\theta}) + \mu_v \dot{\theta} \right) \quad (3.11)$$

where $\dot{\theta}$ [rad/s] is the angular velocity of the joint and F_n [N] is the normal force chosen unity to keep the friction model of the bearing relative simple. The new model of the damped pendulum model now looks like:

$$I_{joint}^* \ddot{\theta} + F_{friction}(\dot{\theta}) + gM_{joint}^* \sin(\theta) = \tau \quad (3.12)$$

Or without the motor force:

$$\ddot{\theta} + \frac{F_{friction}(\dot{\theta})}{I_{joint}^*} + \frac{gM_{joint}^*}{I_{joint}^*} \sin(\theta) = 0 \quad (3.13)$$

3.3.1 Experimental Setup

The model of the Aibo leg is implemented in 20-sim with the unknown parameters. These parameters are estimated by fitting the model to the measurements. The used parameters are the masses (of the shoulder, the upper leg, the lower leg and the extra mass), the distance of the extra mass and the friction coefficients.

The criterion of Equation 3.7 is optimized by changing the parameter values with the Davidson Fletcher Powel gradient search [Bazaraa et al., 1990](included in 20-sim). First, the measurements with only the lower leg are used to estimate the mass of the lower leg and the friction parameters in joint q_3 . Then the same is done with the upper and lower leg turning around q_2 , but with a fixed mass of the lower leg. Finally, the masses of the lower leg and upper leg are fixed and the shoulder mass and the friction parameters of joint q_1 are estimated.

3.3.2 Experimental Results

The tests to estimate the shoulder mass, the upper leg mass and the friction coefficients in the joints (q_1 and q_2) are performed with and without different masses added to the lower leg. Only the estimation of the lower leg mass and the friction coefficients in joint q_3 could not be tested without the use of an extra mass, because the friction is too high. During the numerical estimation some parameters have none or little effect on the estimation. The friction model only needs the viscous and Coulomb friction to fit the model on the test results. One of the results is shown in Figure 3.5 where the fitting of the model is improved in comparison with the other method, shown in Figure 3.2.

The experiments with the shoulder joint q_1 gives some unsatisfactory results. The mass of the shoulder has no or little effect on the estimation, since it is located at the rotation axis. Another effect is the difference between the estimated friction parameters in q_1 and q_2 with or without the extra mass attached to the lower leg. To fit the pendulum model with the measurements, the tests without the extra mass estimate the Coulomb friction very low (around 10^{-3}) and with the mass much higher (around $40 \cdot 10^{-3}$). This difference can be caused by the normal force of the extra mass, which acts on the bearing in the joint. The normal force is taken as unity in Equation 3.11. The choice is made to use the estimations of the friction parameters without the use of the extra mass, because in normal cases the Aibo has no extra mass added to the leg. For joint q_3 no tests could be done without the extra mass, so the tests with the lowest extra mass is used.

The chosen values for the masses and friction parameters are averages of four tests. For q_1 and q_2 only the four tests without an extra mass are used. For joint q_3 the four test with the lowest mass are used. The results of the chosen masses are given in Table 3.2 and the chosen friction parameters in Table 3.3.

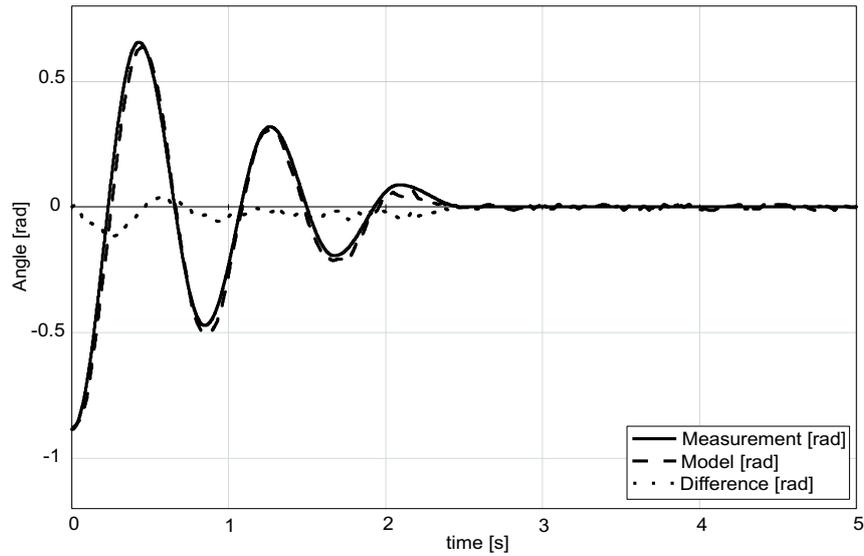


Figure 3.5: Measurement data of the right front shoulder and fitted with the pendulum model of Equation 3.13

Table 3.2: Results from estimation with an extensive model: Mass estimation

Body	Mass [kg]
Shoulder	0.04
Upper Leg	0.052
Lower Leg	0.036

Table 3.3: Results from estimation with an extensive model: Friction estimation

joint	μ_{st} [m]	μ_c [m]	μ_v [ms/rad]	v_{st} [rad/s]	s [s/rad]
q_1	0	0.001	0.017	0	0
q_2	0	0.007	0.019	0	0
q_3	0	0.020	0.01	0	0

3.4 Choice of Leg Parameters

The model parameters have been estimated using two methods, one with a simplified model and one with a more extensive model. This first method used ratios to calculate the values for the missing masses and friction parameter, but the masses were sometimes negative due to the sensitivity of the method.

The results from the more extensive pendulum model are more satisfying. They give a good estimation of the masses and show that the friction in the joints depends mostly on the viscous and Coulomb friction. Therefore a friction model is added in all the legs of the Aibo model, with this equation:

$$F_{jointfriction} = \mu_c \text{sign}(\dot{\theta}) + \mu_v \dot{\theta} \quad (3.14)$$

where μ_c [m] and μ_v [ms/rad] are the Coulomb and viscous friction coefficients and $\dot{\theta}$ [s/rad] the rotational speed of the joint.

The friction as well as the inertias and masses are added to the Aibo model. In Table 3.4 the chosen values of the friction coefficients are given and in Table 3.2 the chosen masses are given. For detailed

information about the chosen masses see Appendix A Table A.2. A comparison is made with the modified Aibo model. The result is compared with real measurements, which is shown in Figure 3.6 for the right front shoulder joint q_1 . The drop is the same, only the steady state position of the joint is different due to small differences in the kinematics between the model and the real Aibo. This difference is not critical as the controllers will actively hold the joints on their position.

Table 3.4: Chosen frictions for the Aibo model

	μ_c [m]	μ_v [ms/rad]
Shoulder	0.001	0.017
Upper leg	0.007	0.019
Lower leg	0.010	0.010

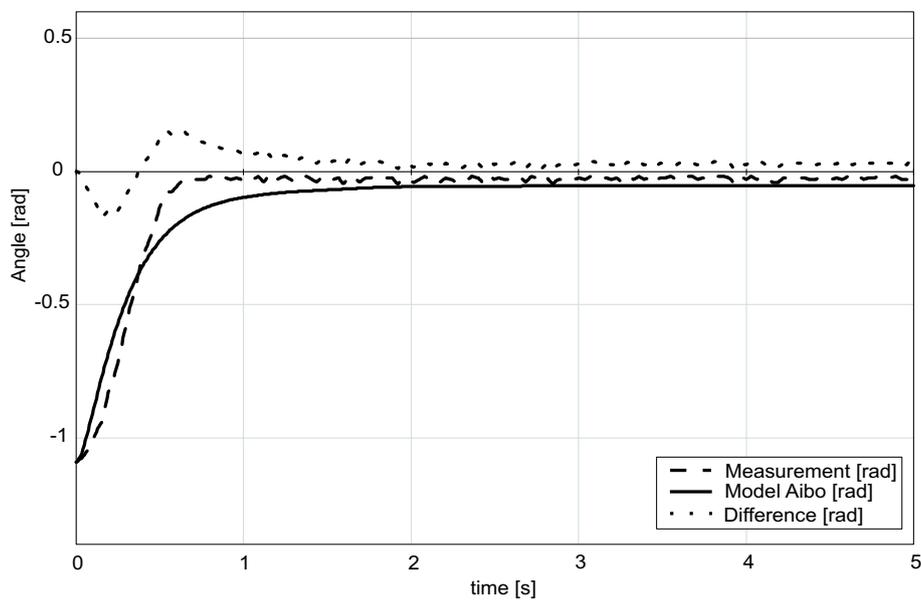


Figure 3.6: Drop test around joint q_1 of the model compared to a measurement from the real Aibo

Chapter 4

Verification of the Aibo Model: using a Walking Pattern

The dynamic model of the Aibo from Chapter 2, with the estimated parameters from Chapter 3, can now be used for simulation. This chapter shows the walking capabilities of the model. A walking pattern sets the setpoints of the controllers at a certain time, making the robot move forwards, backwards, turn or move sideways. There are many different ways to implement gaits with different speeds, stabilities and accuracy. In this chapter several walking gaits are shown and one simple walking algorithm is implemented in the model as a test case. The same walking algorithm is also implemented on the real Aibo and the results are compared.

4.1 Walking Gaits

The dynamical properties of a moving robot limit the possible movements and walking gaits. The speed of a walk also influences the gait pattern. Animals with four legs have several walking patterns, e.g. crawl, trot and pace as shown in Figure 4.1. The crawl operates by moving each leg in turn, lifting it one at a time. It keeps the center of gravity inside the triangle described by the legs that touch the ground. This gives it a stable but slow characteristic. The trot gait lifts two diagonal legs at the same time. The robot balances on the other two diagonal legs. The pace simultaneously lifts both legs on the same side of the body, alternating between the sides. This is the least stable gait, as it is easy for the robot to lose its balance.

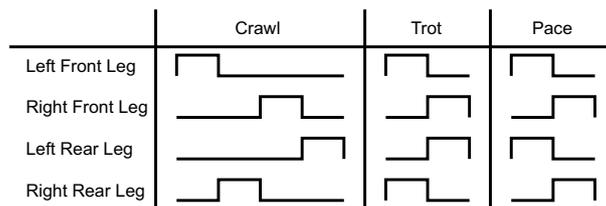


Figure 4.1: Walking gaits for a quadruped robot [Hengst et al., 2002]

The choice of a gait depends on the task at hand. Speed can be a criterion, but in other cases switching between walks or combining different walks can be more important, as with the RoboCup. Sometimes accuracy is important for localization or ball handling. In other cases a stable movement of the head is desirable for sharp camera images. In all cases it is important that the robot does not become unstable. The model that is used in this report can explore the boundaries of the possible motions. To test these limits, a stable trot is chosen, shown in Figure 4.2. The trot is a simplification of an earlier study from Stelzer et al. [2003] and re-defined to implement it in Tekkotsu. The dynamical model itself

uses the frame definition of OPEN-R, but Tekkotsu uses the Denavitt-Hartenberg method explained in Craig [1989].

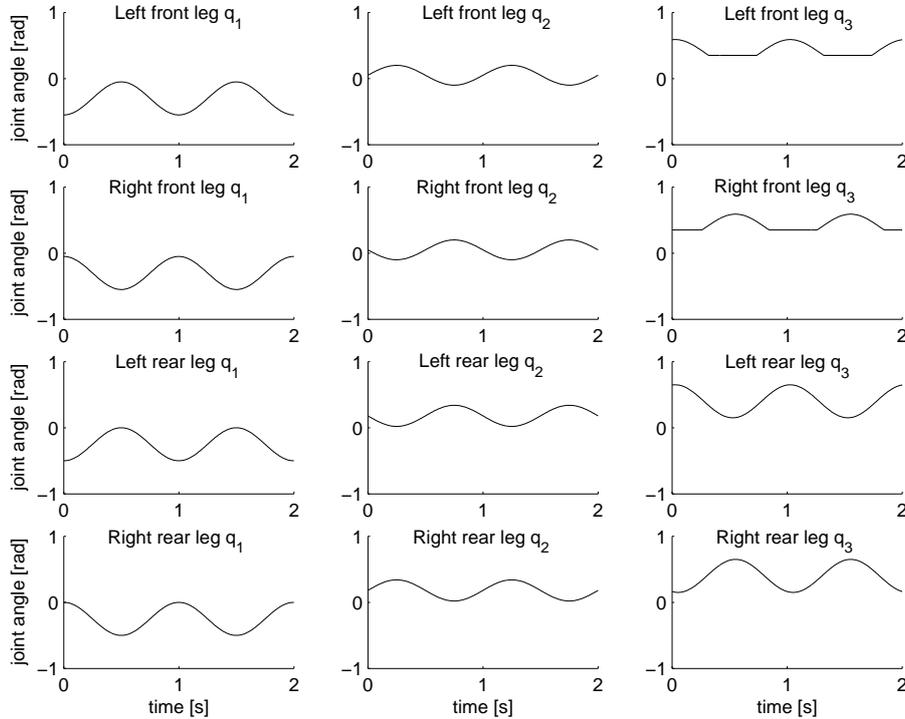


Figure 4.2: Implemented setpoints (Denavitt-Hartenberg definition) of the walking gait, which is used as a test walk in the dynamic model and on the real Aibo

4.2 Test Case: Walking Aibo

To compare the model with the real Aibo, a test walk is implemented to see if the model behaves according to the real Aibo. The result of this section will be the difference of the measured angles of the real Aibo compared to the simulated output angles of the model, as well as the walking distance.

4.2.1 Setup Test Case

The test case is the implementation of a stable walking gait. The walking gait is implemented in 20-sim and Tekkotsu. The possibilities of Tekkotsu are limited by the hardware of the Aibo and this limits the testing possibilities.

The walking gait is first implemented in the model, within the 20-sim environment. The algorithm consists of continuous movements of the joints as given in Figure 4.2. The simulation is done with different integration methods for a duration of ten seconds on a AMD Athlon(tm) XP 1800+ with 512 MB of RAM. The output is the given setpoints, the simulated joint angles and the forward displacement. The results of the setpoints and measured angles can be exported to a text file. With this text file a readable motion file is made with Matlab, which can be used within the Tekkotsu platform.

To implement a walking algorithm in OPEN-R or Tekkotsu, an option is to write a C++ file. Unfortunately debugging the code is time consuming and the walking algorithm is frequently changed during tests. Fortunately Tekkotsu has a functionality to play motions that are defined in a text file. In the text file setpoints for specific joints are given on specific times. The disadvantage is the sampling time in the text file, which cannot be lower than 128 [ms]. Programming the gait in C++ allows a sampling time

of 8 [ms]. Using the motion text file allows changing the motions files during the tests, while changing the C++ code means restarting the Aibo for every test. In this stadium the frequency limits are not explored and therefore the choice is made to use the fast motion files.

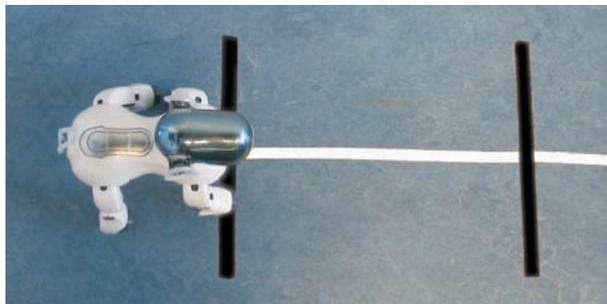


Figure 4.3: Test setup of the Aibo on the graduated ruler

The Tekkotsu platform is used to send the measured joint angles through wireless LAN to a computer, where the data is logged and stored. The displacement of the Aibo is not measured with the sensors of the Aibo, but with a graduated ruler on the floor, as shown in Figure 4.3. The results from the measured joint angles are compared with the simulated results in Matlab and the displacement of the real Aibo is measured by hand and compared with the simulation.

4.2.2 Results Test Case

Simulation results

Simulations with the walking gait shows that the Aibo can move forward, using the friction of the floor. Figure 4.4 gives an animation of the walking gait. The walk does not look very natural, because the knees are not bending much. The goal here is not to optimize this gait, but to show that the real Aibo will also walk like the animated model. As long as the gait is stable and the real Aibo does not fall, it can be used as a test case.

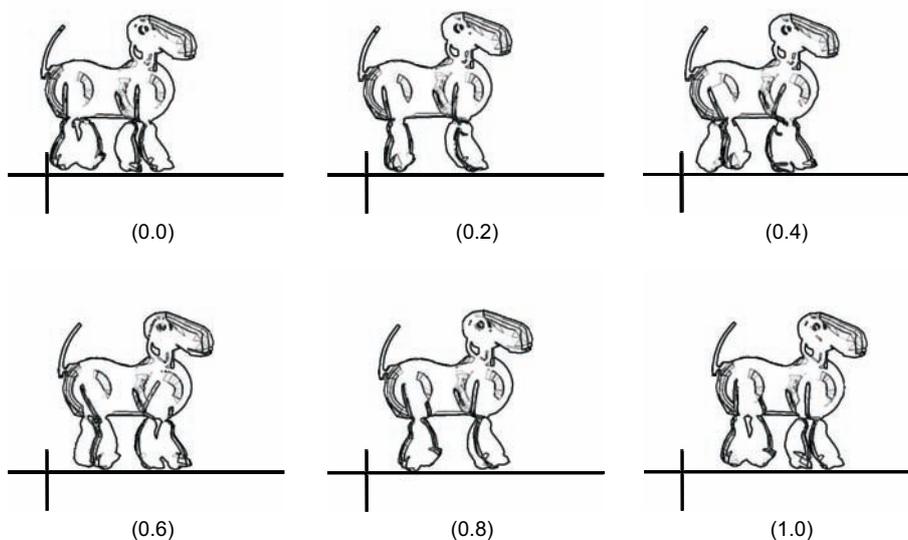


Figure 4.4: Animation of the simulated Aibo gait for one step cycle

In Figure 4.5, the displacement of the model in the forward direction is shown. To initiate the contact model, the Aibo is dropped from 0.02 [m] in order to prevent the feet from touching the floor. After two seconds the effects from the drop have faded and the walk starts for eight seconds. The Aibo has a displacement of 0.51 [m], which is a speed of 0.064 [m/s]. The sideways displacement from the left to the right as well as the vertical movement are stable and also shown in Figure 4.5. The walk is also simulated for five minutes, to see the drift or even unstable behavior. The result (given in Figure C.1 in Appendix C) shows that when the Aibo drops, it gets an initial offset to the right. The walk stays stable with a constant speed.

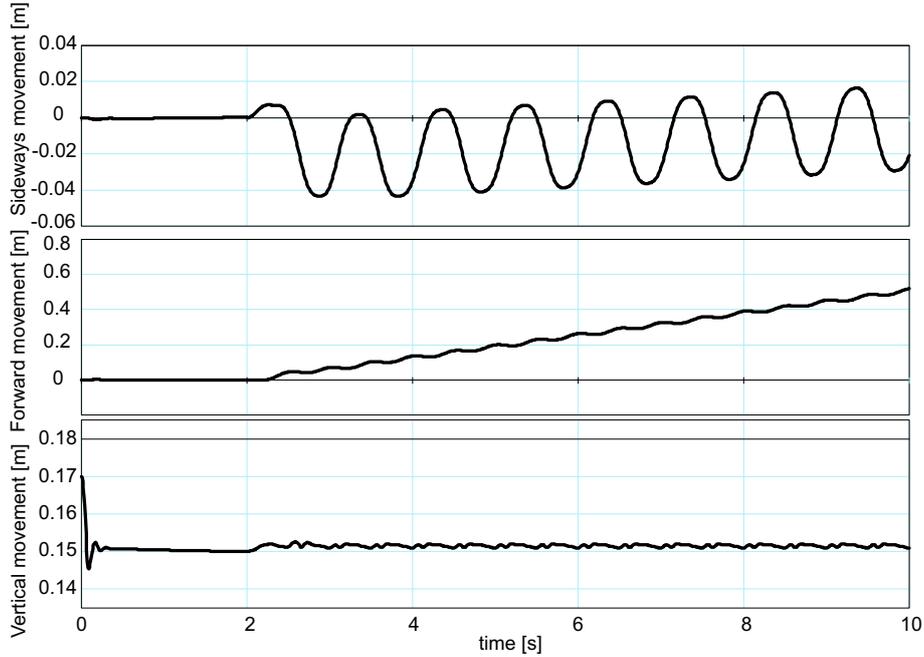


Figure 4.5: Forward movement of the simulated Aibo

Several integration methods can be used to simulate the model. The chosen stepsize can influence the result, because it introduces little differences in the contact force calculations. Smaller stepsizes will make the simulation slower. For several integration methods and different stepsizes tests are done and shown in Table 4.1. The calculation time and displacement of the robot differs per integration method. The table shows that several integration methods have an end displacement around 0.51 [m]. The fastest integration method to reach this result is achieved with the Runge-Kutta 4 method, with a stepsize of 10^{-5} [s].

Measurement results from the real Aibo

The setpoints of the walking gait from the simulation are modified into a motion file for Tekkotsu and used on the real Aibo. The robot waits for two seconds and walks for eight seconds, just like in the simulation. First a test is carried out without the feet touching the floor. This allows to see if the motors are capable of following the reference signal. The measured joint angles (given in Figure C.2 in Appendix C) follow the reference signal without any delay or amplitude difference.

The influence of the floor is tested, to see if the walking gait really displaces the real Aibo as the simulation estimates. The measurements of the joint angles during the walk are given in Figure 4.6 as well as the reference signal and the simulation results. In the figure some signals are enlarged for two periods of the walk. The result shows that the simulation follows the reference signal very well, but the shoulder joints q_2 of the real Aibo seem to have difficulties pushing the feet on the floor. The motors are not strong enough to overcome the floor friction, while the simulation has no trouble with it.

Table 4.1: Integration methods tested on the model

Integration method	Parameters integration			Result		
	Step size [s]	Absolute	Relative	Calculation time [s]	Calculations Number	Displacement [m]
Runge-Kutta 4	5.00E-04	.	.	175.172	80459	0.607
	1.00E-04	.	.	837.813	400962	0.513
	1.00E-05	.	.	7516.94	4000831	0.508
	1.00E-06	.	.	75493.9	40000670	0.517
Modified Backward - Differentiation Formula	.	1.00E-05	1.00E-05	2802.91	1450251	0.511
	.	1.00E-06	1.00E-06	2239.44	1130894	0.508
Vode Adams	.	1.00E-06	1.00E-06	18926.3	9789172	0.542
	.	1.00E-05	1.00E-05	43585.1	22956984	0.608
Runge-Kutta 2	5.00E-05	.	.	767	400887	0.535
	1.00E-05	.	.	3820.52	2000926	0.517

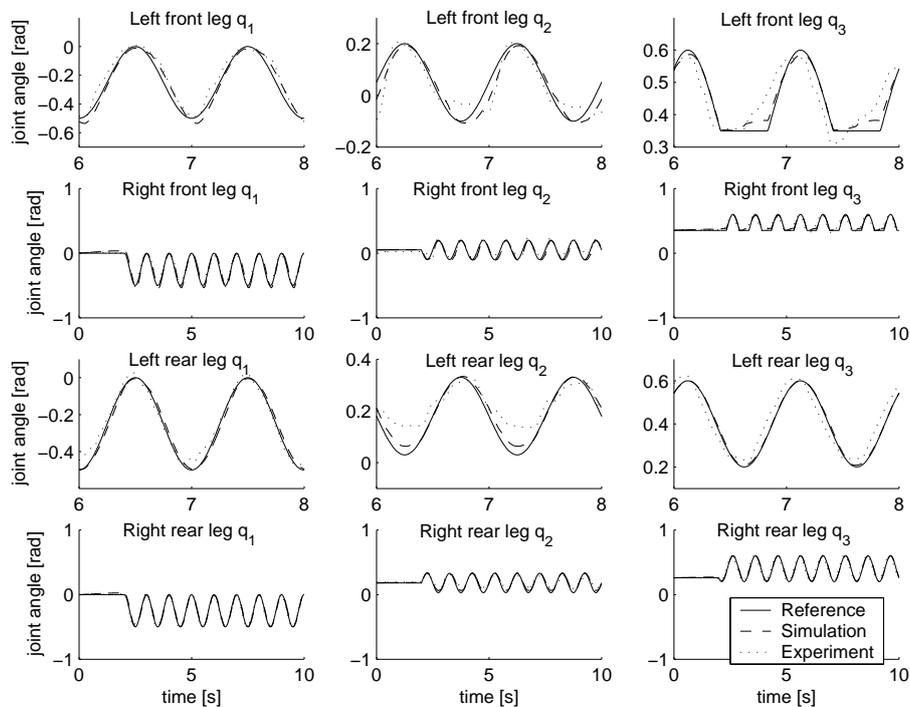


Figure 4.6: The reference signal is a 1 Hz walking gait. The results from the simulation and the real Aibo are shown next to the reference signal. Some signals are enlarged to show the difference between the simulation and the real measurements

Using a higher frequency of the gait is tested, but the use of the motion files limits this frequency. The number of setpoints per period is too small with a sampling of 128 [ms], as can be seen in Appendix C Figure C.3. If a higher frequency is desirable, then the walking algorithm should be implemented as a motion behavior in C++.

The real displacement of the Aibo differs per test. The initial position is very important for the walking direction of the Aibo, as well as in the simulation. The displacement is between 0.49 to 0.54 [m] on a smooth lab floor and is comparable with the simulation results.

4.3 Aibo Model versus Real Aibo

The dynamic model of the Aibo has been tested with a simple walking algorithm. The model was capable of walking with a stable gait, following the given setpoints accurately. The real Aibo walked the same distance on a smooth surface, but some joints were not able to follow the given setpoints due to the friction of the floor and the limitations of the motors. The maximum frequency of the walking gait could not be explored, while the used method had a limited sampling time. The test case showed that the motions of the model are a good estimation of the real Aibo and gives a good prediction what the displacement and speed will be of the real Aibo. This model can be used to optimize the walk or other motions, without the use of the real Aibo. When the motion is optimized, it can be transformed to C++ code and implemented into the real Aibo. The disadvantage is that not all robots are the same, due to different wear of the motors and play in the joints, therefore the performance can differ per Aibo.

Chapter 5

Conclusions and Recommendations

5.1 Conclusions

The main objective of this Master's project was to design and verify a dynamical model of an ERS-7 Aibo. This study was divided into the design of a multi-body dynamical model, the estimation of the dynamical parameters and the verification of the model with the real Aibo.

Design a multi-body dynamical model of the ERS-7 Aibo

The dynamics of the model were implemented in the 3D Mechanics Toolbox of 20-sim, using the data provided by Sony. The kinematical implementation gave a good resemblance of the real Aibo. For the contact model with the floor a compliant model was chosen, instead of a rigid contact model. The compliant model has the advantage that it can be used in a continuous manner, without a lot of book-keeping.

The motor characteristics were unknown and not implemented in the model. This made the dynamical model not suitable to check the power consumption or energy efficiency of the walking gait. Only the maximal motor speeds are given and have been implemented in the model. The model gives a warning during the simulation if the boundaries are exceeded.

Estimate the missing dynamical parameters

Some dynamical parameters of the Aibo were missing, because they were not publicly available. These parameters were the mass, center of mass and moment of inertia for every body part as well as the friction coefficients in the joints. Several experiments were done to estimate these missing dynamical parameters of the real Aibo. The body parts were taken as ellipsoids with the center of mass in the center of the body. The masses and friction parameters were estimated using a pendulum model that was fitted on the measurement data. Only the shoulder mass could not be estimated accurately, while this mass is located at the rotational axis of the joint. The friction estimation showed that the friction in the joints is properly described by Coulomb and viscous friction.

Verify the model

To compare the dynamical model with the real Aibo, a test case was made. The test case was an implementation of a stable walk within the model and the real Aibo. The measurement setup in Tekkotsu made it easy to test different motions, but limited the walking frequency to one step per second.

The result showed that the joint angles of the model corresponded with the joint angles of the real Aibo during the walk. Also the displacement in the forward direction of the Aibo was comparable. The simulation time of ten seconds took about 13 minutes (on an AMD Athlon(tm) 1800+, 512 MB), while the real test only took one or two minutes. This difference becomes smaller if longer tests are needed,

because the disadvantages of testing with the real Aibo (battery, CPU, compilation time) will increase the testing time.

General conclusions

In general, the model is capable of simulating the same behavior as the real Aibo. Not all aspects have been tested and the simulation time is still high. The advantages are the removal of practical limitations during long tests with the Aibo. The model is especially useful in combination with optimization and learning algorithms that needs multiple runs and has to change several parameters of the motion. The model gives a good overview of the robot and all internal parameters can be viewed.

5.2 Recommendations

The purpose of the dynamic model of the Aibo is to create stable and fast walking patterns. This model is one step in that direction, as it can simulate the Aibo behavior, but cannot generate new motions. In this thesis the foundations are laid on which optimization and learning algorithms can be built to use and explore the boundaries of the Aibo. Therefore there are still several recommendations to give:

Design a multi-body dynamical model of the ERS-7 Aibo

- The simulation time of the model is six times higher than a test with the real Aibo. The model in this thesis is not fully optimized for the whole code. Most of the simulation time is due to the compliant contact model. The spatial spring and damper are very stiff in order to prevent the Aibo from bouncing on the floor, but the drawback is the increasing simulation time. For tests with a short duration or in combination with other applications, a faster simulation time would be preferred. Therefore a simplification or optimization of the code could be an option. It could also be possible to simplify the dynamical model to only a kinematical model combined with a contact model, but this requires another kind of contact model.
- The characteristics of the motors have not been implemented. If the parameters of the motor are estimated, they can be included into the model. This information will make it possible to look at the forces and speeds of the motor and might be used to create stable and energy efficient walking patterns.

Estimate the missing dynamical parameters

- The missing parameters from the Aibo were estimated using several tests and fitted with a pendulum model. A better method would be to obtain the data directly from measurements of each body part separately. Then it is possible to measure the center of mass, the real mass and the moment of inertia in an accurate and direct way. However, the influence on the performance of the model will be minimal, while all its joints are actively actuated.

Verify the model

- The model has not been tested with other motions, like turning, pace or kicking a ball. The main focus was on the model and the contact force with the ground. The model becomes more accurate, if different motions are tested and the result is used to improve the model. The use of unstable motions can be an indication for the accuracy of the estimated dynamical parameters. With the use of stable motions it is shown that the model has the same walking pattern as the real Aibo. When the Aibo becomes unstable from a walk, the moment it falls down should be the same as in the simulation. This will indicate that the model behaves very accurately compared to the real Aibo.

- The measurement of the displacement in this thesis is performed by hand. This makes the measurement less accurate and is only useful with a forward or backward walk. The measurement could be performed with the onboard sensors of the Aibo, like the camera in combination with the acceleration sensors. If rocky walking patterns are used, this method could become less accurate. Measurements of the displacement with the camera has been used before in the research of Duffert and Hoffmann [2006].
- The implementation of the walking algorithm is in practice always embedded in the software of the Aibo. Only the parameters of the walk can be changed. The advantage of the walking algorithm embedded in the software is the high sampling rate of 8 [ms]. The disadvantage is that the code needs to be re-compiled with a new walking algorithm. This can be a time consuming process. When an optimal walking pattern is found with the model, a translation should be made to implement it on the Aibo. The walking pattern should have enough adjustable parameters during a walk to adapt to changes in the environment.

General recommendations

A combination of a software platform and the model of the Aibo would provide a good support of developing new robotic walking algorithms. There are simulators available to test the Aibo software without the use of the Aibo hardware. The dynamical model can be implemented into such a software simulator to simulate the software and hardware simultaneously.

The model should be used in combination with optimization and learning algorithms to look for new motions or walking patterns (other than crawl, trot and pace). A comparison should be made with results from other researches, which use the real Aibo instead of a model. This can directly show the benefits and the disadvantages of using a multi-body dynamical model.

Bibliography

- Armstrong-Héouvy, B., Dupont, P., and de Wit, C. C. (1994). A survey of models, analysis tools and compensation methods for the control of machines with friction. *Automatica*, 30(7):1083–1138.
- Baillie, J.-C. (2005). Universal Robotic Body Interface website. <http://www.urbiforge.com>.
- Bazaraa, M., Sherali, H., and Shetty, C. (1990). *Nonlinear Programming, Theory and Algorithms*. John Wiley and Sons Inc. New York.
- Blender (2005). Blender website. <http://www.blender.org/>.
- Breedveld, P. C. and van Amerongen, J. (1994). *Dynamische Systemen, modelvorming en simulatie met bondgrafen - Deel. 1, 2 en 3*. Open Universiteit, Heerlen.
- Carnegie Mellon University (2005). Tekkotsu website. <http://www.tekkotsu.org/>.
- Control Lab Products (2005). 20-sim version 3.6. <http://www.20sim.com/>.
- Craig, J. J. (1989). *Introduction to Robotics: Mechanics and Control*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.
- Düffert, U. and Hoffmann, J. (2006). Reliable and precise gait modeling for a quadruped robot. In *RoboCup 2005: Robot Soccer World Cup IX*. Springer. To appear.
- Duindam, V. (2004). Impact modeling for 3d contact of rigid mechanisms. Personal communication.
- Duindam, V. and Stramigioli, S. (2003). Modeling the kinematics and dynamics of compliant contact. *International Conference on Robotics and Automation*, pages 4029–4034.
- Dutch Aibo Team (2005). Dutch Aibo Team website. <http://www.dutchaiboteam.nl/>.
- German Team (2005). German Team website. <http://www.germanteam.org/>.
- Hardt, M. and von Stryk, O. (2002). The role of motion dynamics in the design, control and stability of bipedal and quadrupedal robots. In *RoboCup 2002: International Symposium, Fukuoka, Japan*, pages 206–223.
- Hengst, B., Ibbotson, D., Pham, S. B., and Sammut, C. (2002). Omnidirectional locomotion for quadruped robots. In *RoboCup 2001: Robot Soccer World Cup V*, pages 368–373, London, UK. Springer-Verlag.
- Hunt, K. H. and Crossley, F. (1975). Coefficient of restitution interpreted as damping in vibroimpact. *ASME Journal of Applied Mechanics*, 42(2):440–445.
- Open Dynamics Engine (2005). Open Dynamics Engine website. <http://ode.org/>.
- Röfer, T. (2004). Evolutionary gait-optimization using a fitness function based on proprioception. In *RoboCup 2004: International Workshop, Lisbon, Portugal*, pages 310–322.

- Sony (2004). *OPEN-R SDK, Model Information for ERS-7*. Sony Corporation, <http://openr.aibo.com/> (download section of the registered area).
- Sony Corporation (2005a). OPEN-R website. <http://openr.aibo.com/>.
- Sony Corporation (2005b). Sony Aibo website. <http://www.sony.net/Products/aibo/index.html>.
- Stelzer, M., Hardt, M., and von Stryk, O. (2003). Efficient dynamic modeling, numerical optimal control and experimental results for various gaits of a quadruped robot. *CLAWAR: International Conference on Climbing and Walking Robots, Catania, Italy*, pages 601–608.
- Stramigioli, S. (2001). *Modeling and IPC Control of Interactive Mechanical Systems - A Coordinate-free Approach*. Springer-Verlag.
- Stramigioli, S. and Bruyninckx, H. (2001). *Geometry and screw theory for robotics*. Tutorial during ICRA 2001.
- Stramigioli, S. and Duijndam, V. (2002). A novel lumped spatial model of tire contact. *International Conference on Intelligent Transportation Systems*, pages 600–605.
- The Robocup Foundation (2005). The Robocup Foundation website. <http://www.robocup.org/>.
- Webots (2005). Webots website. <http://www.cyberbotics.com/>.
- Weisstein, E. W. (2005). Eric Weisstein's World of Physics. <http://scienceworld.wolfram.com/physics/MomentofInertiaEllipsoid.html>.

Appendix A

Kinematical and Dynamical Properties of the ERS-7 Aibo

The ERS-7 Aibo has several body parts as can be seen in Figure 2.1, for the model 19 parts are used. These parts are linked together with joints. The connection positions of the joints are given in Table A.1. The body of the Aibo has 6 body parts linked to it, the neck, tail and four shoulders. The positions of the joints are defined with the frame of the Aibo body, which is located in the middle of the body. OPEN-R defines its frames in a LightWave 5.6 format. The frames of all the body parts are orientated with the z-axis to the rear end of the Aibo, the x-axis point to the right side of the Aibo and the y-axis is upwards. Others define it with the Denavit-Hartenberg method [Craig, 1989] of computing kinematic parameters. The z-axis of each link is determined by the axis of rotation for revolute joints. For this model the method of OPEN-R is used. The legs and neck have an initial offset of 30 or -30 degrees depending on their location (front/rear/left/right). Some link connections are measured and are in italics in the table, because they were not specified by OPEN-R.

The physical dimensions of the bodies are given in Table A.2. The lengths are given for the same frames used before. The lengths are given as the diameter of the body. Masses and inertias are estimated, since Sony does not provide any information about this. The estimation is discussed in Chapter 3. The hardware and software limitations of the joints are given in Table A.3, as well as the speed limits of the joints.

Table A.1: Connections of the joints of the ERS-7 Aibo, used in the 3D Mechanical Toolbox from 20-sim

Joint Name	x [mm]	y [mm]	z [mm]	φ_x [deg]	φ_y [deg]	φ_z [deg]	Link Name
Aibo Body							
Neck Tilt 1	0	19.5	-67.5	0	0	0	Neck
Tail Tilt	0	33	102	0	0	0	Tail 1
Left Front q_1	0	0	-65	0	0	0	Left Front Shoulder
Right Front q_1	0	0	-65	0	0	0	Right Front Shoulder
Left Rear q_1	0	0	65	0	0	0	Left Rear Shoulder
Right Rear q_1	0	0	65	0	0	0	Right Rear Shoulder
Neck							
Neck Tilt 1	0	-40	0	0	0	0	Aibo Body
Neck Pan	0	40	0	0	0	0	Neck Ghost
Neck Ghost							
Neck Pan	0	0	0	0	0	0	Neck
Neck Tilt 2	0	0	0	-30	0	0	Head
Head							
Neck Tilt 2	0	1.75	21.4	0	0	0	Neck Ghost
Left Ear Tilt	-42	16.25	32.4	0	0	0	Left Ear 1
Right Ear Tilt	42	16.25	32.4	0	0	0	Right Ear 1
Mouth Tilt	0	-17.5	-40	0	0	0	Mouth
Left Ear 1							
Left Ear Tilt	0	0	0	0	0	0	Head
Left Ear Pan	0	0	0	0	0	0	Left Ear 2
Left Ear 2							
Left Ear Pan	0	30	0	0	0	0	Left Ear 1
Right Ear 1							
Right Ear Tilt	0	0	0	0	0	0	Head
Right Ear Pan	0	0	0	0	0	0	Right Ear 2
Right Ear 2							
Right Ear Pan	0	30	0	0	0	0	Right Ear 1
Mouth							
Mouth Tilt	0	0	10	0	0	0	Head
Tail 1							
Tail Tilt	0	0	0	0	0	0	Aibo Body
Tail Pan	0	0	0	0	0	0	Tail 2
Tail 2							
Tail Tilt	0	0	10	0	0	0	Tail 1
Left Front Shoulder							
Left Front q_1	50	0	0	0	0	0	Aibo Body
Left Front q_2	-12.5	0	0	0	0	0	Left Front Upper Leg
Left Front Upper Leg							
Left Front q_2	0	35	0	0	0	0	Left Front Shoulder
Left Front q_3	-4.7	-35	-9	30	0	0	Left Front Lower Leg
Left Front Lower Leg							
Left Front q_3	0	45	0	0	0	0	Left Front Upper Leg
Left Front Lower Leg Contact	0	0	0	0	0	0	-
Right Front Shoulder							
Right Front q_1	-50	0	0	0	0	0	Aibo Body
Right Front q_2	12.5	0	0	0	0	0	Right Front Upper Leg
Right Front Upper Leg							
Right Front q_2	0	35	0	0	0	0	Right Front Shoulder
Right Front q_3	4.7	-35	-9	30	0	0	Right Front Lower Leg
Right Front Lower Leg							
Right Front q_3	0	45	0	0	0	0	Right Front Upper Leg
Right Front Lower Leg Contact	0	0	0	0	0	0	-
Left Rear Shoulder							
Left Rear q_1	50	0	0	0	0	0	Aibo Body
Left Rear q_2	-12.5	0	0	0	0	0	Left Rear Upper Leg
Left Rear Upper Leg							
Left Rear q_2	0	35	0	0	0	0	Left Rear Shoulder
Left Rear q_3	-4.7	-35	9	-30	0	0	Left Rear Lower Leg
Left Rear Lower Leg							
Left Rear q_3	0	45	0	0	0	0	Left Rear Upper Leg
Left Rear Lower Leg Contact	0	0	0	0	0	0	-
Right Rear Shoulder							
Right Rear q_1	-50	0	0	0	0	0	Aibo Body
Right Rear q_2	12.5	0	0	0	0	0	Right Rear Upper Leg
Right Rear Upper Leg							
Right Rear q_2	0	35	0	0	0	0	Right Rear Shoulder
Right Rear q_3	4.7	-35	9	-30	0	0	Right Rear Lower Leg
Right Rear Lower Leg							
Right Rear q_3	0	45	0	0	0	0	Right Rear Upper Leg
Right Rear Lower Leg Contact	0	0	0	0	0	0	-

Table A.2: Physical dimensions of the ERS-7 Aibo

	Diameter size			Mass [kg]	Inertia		
	x [m]	y [m]	z [m]		I_{xx} [kg.m ²]	I_{yy} [kg.m ²]	I_{zz} [kg.m ²]
Aibo Body	0.0993	0.1141	0.226	0.9132	2.93E-03	2.78E-03	1.04E-03
Neck	0.0268	0.08	0.0268	0.05	1.78E-05	3.59E-06	1.78E-05
Neck Ghost	0.01	0.01	0.01	0	0.00E+00	0.00E+00	0.00E+00
Head	0.0752	0.0565	0.1328	0.18	1.87E-04	2.10E-04	7.96E-05
Mouth	0.04	0.007	0.04	0.02	1.65E-06	3.20E-06	1.65E-06
Left Ear 1	0.01	0.01	0.01	0	0.00E+00	0.00E+00	0.00E+00
Left Ear 2	0.01	0.06	0.04	0.0045	1.17E-06	3.83E-07	8.33E-07
Right Ear 1	0.01	0.01	0.01	0	0.00E+00	0.00E+00	0.00E+00
Right Ear 2	0.01	0.06	0.04	0.0045	1.17E-06	3.83E-07	8.33E-07
Tail 1	0.01	0.01	0.01	0	0.00E+00	0.00E+00	0.00E+00
Tail 2	0.04	0.08	0.01	0.0071	2.31E-06	6.04E-07	2.84E-06
Left Front Shoulder	0.02	0.0695	0.0695	0.04	1.93E-05	1.05E-05	1.05E-05
Right Front Shoulder	0.02	0.0695	0.0695	0.04	1.93E-05	1.05E-05	1.05E-05
Left Rear Shoulder	0.02	0.0695	0.0695	0.04	1.93E-05	1.05E-05	1.05E-05
Right Rear Shoulder	0.02	0.0695	0.0695	0.04	1.93E-05	1.05E-05	1.05E-05
Left Front Upper Leg	0.0368	0.07	0.0328	0.052	1.55E-05	6.32E-06	1.63E-05
Right Front Upper Leg	0.0368	0.07	0.0328	0.052	1.55E-05	6.32E-06	1.63E-05
Left Rear Upper Leg	0.0368	0.07	0.0328	0.052	1.55E-05	6.32E-06	1.63E-05
Right Rear Upper Leg	0.0368	0.07	0.0328	0.052	1.55E-05	6.32E-06	1.63E-05
Left Front Lower Leg	0.042	0.09	0.045	0.036	1.82E-05	6.82E-06	1.78E-05
Right Front Lower Leg	0.042	0.09	0.045	0.036	1.82E-05	6.82E-06	1.78E-05
Left Rear Lower Leg	0.046	0.0922	0.046	0.036	1.91E-05	7.62E-06	1.91E-05
Right Rear Lower Leg	0.046	0.0922	0.046	0.036	1.91E-05	7.62E-06	1.91E-05
Total mass Aibo	.	.	.	1.6913	.	.	.

* The sizes of all the links have been measured and the masses and inertias are estimated.

Table A.3: Hardware and software limitations of the joints

	Hardware Limitations		Software limitations		
	min degree	max degree	min degree	max degree	speed
	[rad]	[rad]	[rad]	[rad]	[rad/s]
Left Front Leg q_1	-2.0944	2.3562	-2.0071	2.2689	4.865
Left Front Leg q_2	-0.2618	1.6232	-0.1745	1.5359	5.280
Left Front Leg q_3	-0.5236	2.2166	-0.4363	2.1293	5.280
Right Front Leg q_1	-2.0944	2.3562	-2.7053	2.2689	4.865
Right Front Leg q_2	-0.2618	1.6232	-0.1745	1.5359	5.280
Right Front Leg q_3	-0.5236	2.2166	-0.4363	2.1293	5.280
Left Rear Leg q_1	-2.3562	2.0944	-2.2689	2.0071	4.865
Left Rear Leg q_2	-0.2618	1.6232	-0.1745	1.5359	5.280
Left Rear Leg q_3	-0.5236	2.2166	-0.4363	2.1293	5.280
Right Rear Leg q_1	-2.3562	2.0944	-2.2689	2.0071	4.865
Right Rear Leg q_2	-0.2618	1.6232	-0.1745	1.5359	5.280
Right Rear Leg q_3	-0.5236	2.2166	-0.4363	2.1293	5.280
Neck Tilt 1	-1.3963	0.0524	-1.3090	0.0000	3.185
Neck Pan	-1.6232	1.6232	-1.5359	1.5359	10.057
Neck Tilt 2	-0.3491	0.8727	-0.2618	0.7854	5.781
Tail Tilt	0.0000	1.1345	0.0873	1.0472	15.163
Tail Pan	-1.0472	1.0472	-0.7854	0.7854	15.163
Mouth	-0.9599	-0.0524	-0.9599	-0.0524	10.145
Left Ear Tilt	0.0000	0.2618	0.0000	0.0175	0
Left Ear Pan	-0.7854	0.7854	.	.	0
Right Ear Tilt	0.0000	0.2618	0.0000	0.0175	0
Right Ear Pan	-0.7854	0.7854	.	.	0

Appendix B

Sensor and Controller Specifications

The specifications of the sensors in Table B.1 are extracted from the on-line manual [Sony, 2004] and from the Tekkotsu source code [Carnegie Mellon University, 2005]. For the specific gains of the PID controllers, Table B.2 gives an overview of the values.

Table B.1: Specifications of the sensors inside the ERS-7

	Unit	Range	
		minimal	maximal
Head IR Distance (near)	$10^{-6} [m]$	50000	500000
Head IR Distance (far)	$10^{-6} [m]$	200000	1500000
Chest IR Distance	$10^{-6} [m]$	100000	900000
Front-back Acceleration	$10^{-6} [m/s^2]$	-19613300	19613300
Right-left Acceleration	$10^{-6} [m/s^2]$	-19613300	19613300
Up-down Acceleration	$10^{-6} [m/s^2]$	-19613300	19613300
Power remaining	ratio	0	1
Power thermometer	degrees Celcius	.	.
Power Capacity	milli-amp hours	.	.
Power Voltage	volts	.	.
Power Current	milli-amp	.	.

B.1 Measurements with Position Sensors

The position sensor characteristics are not given by OPEN-R. This is why some tests are performed to get an insight in the precision and accuracy of the sensors. As a test case the position sensor is used from the left front shoulder of the ERS-7 Aibo. The joint is set to a specific position and from that position slowly changed to measure only the precision error and its accuracy, without taking speed into account.

The left front shoulder can move from -2.007 to 2.27 radians, 312 measurements have been done in the range of -2 to 2 [rad] and are plotted in Figure B.1. The measured angles are close to the reference line, but at the borders the measured angles differ from the setpoints. Another test is performed in the interval [-0.05 to 0.05] [rad] with, again, 312 measurements. Results are shown in Figure B.2. It shows the stepsize of the sensor and the variation around the setpoints. The quantization error of the sensor is the difference between the setpoint and the measured angle, as given in Figure B.3, and gives a better insight of the stepsize and variation. The minimal stepsize is 0.006 [rad] and the variation of the signal around 0.012 [rad]. The measured values are the same or higher than the given setpoint.

Table B.2: Specifications of the PID gains

	P gain			I gain			D gain		
	g	s	gain	g	s	gain	g	s	gain
Left Front Leg q_1	20	14	5	4	2	2.44E-04	1	15	0.5
Left Front Leg q_2	28	14	7	8	2	4.88E-04	1	15	0.5
Left Front Leg q_3	28	14	7	8	2	4.88E-04	1	15	0.5
Right Front Leg q_1	20	14	5	4	2	2.44E-04	1	15	0.5
Right Front Leg q_2	28	14	7	8	2	4.88E-04	1	15	0.5
Right Front Leg q_3	28	14	7	8	2	4.88E-04	1	15	0.5
Left Rear Leg q_1	20	14	5	4	2	2.44E-04	1	15	0.5
Left Rear Leg q_2	28	14	7	8	2	4.88E-04	1	15	0.5
Left Rear Leg q_3	28	14	7	8	2	4.88E-04	1	15	0.5
Right Rear Leg q_1	20	14	5	4	2	2.44E-04	1	15	0.5
Right Rear Leg q_2	28	14	7	8	2	4.88E-04	1	15	0.5
Right Rear Leg q_3	28	14	7	8	2	4.88E-04	1	15	0.5
Neck Tilt 1	10	14	2.5	4	2	2.44E-04	2	15	1
Neck Pan	8	14	2	2	2	1.22E-04	4	15	2
Neck Tilt 2	8	14	2	8	2	4.88E-04	2	15	1
Tail Tilt	10	14	2.5	4	2	2.44E-04	4	15	2
Tail Pan	10	14	2.5	4	2	2.44E-04	4	15	2
Mouth	8	14	2	0	2	0.00E+00	4	15	2

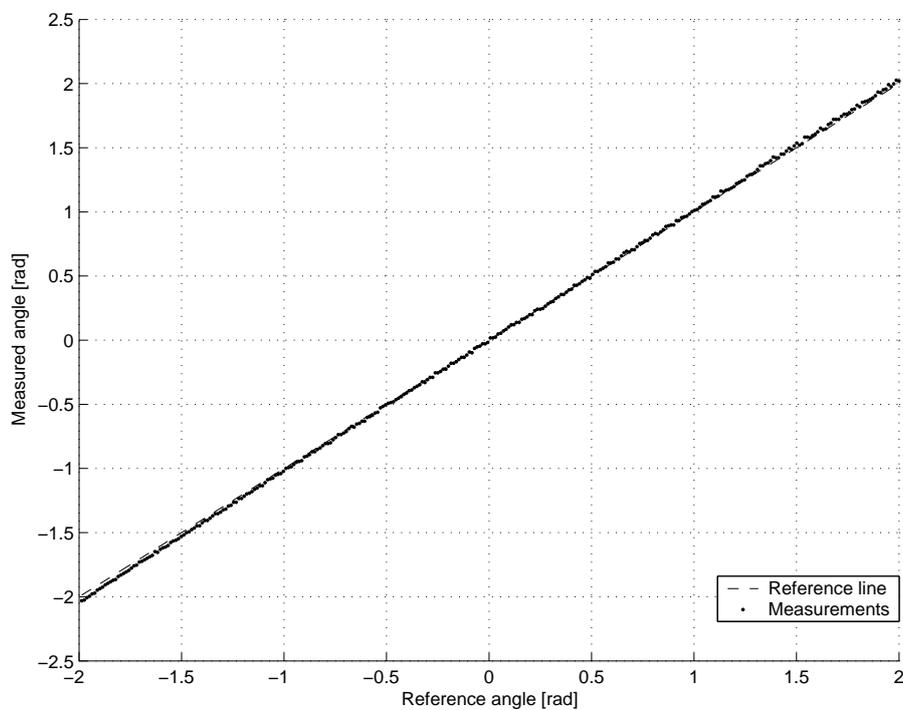


Figure B.1: Sensor range from -2 till 2 [rad], with 312 measurement points

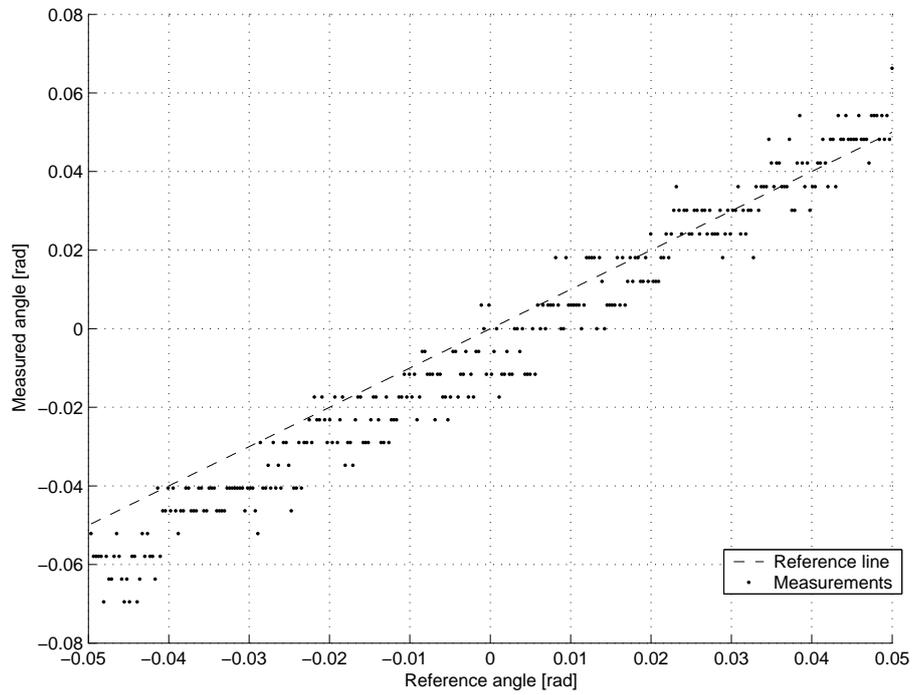


Figure B.2: Sensor range from -0.05 till 0.05 [rad], with 312 measurement points

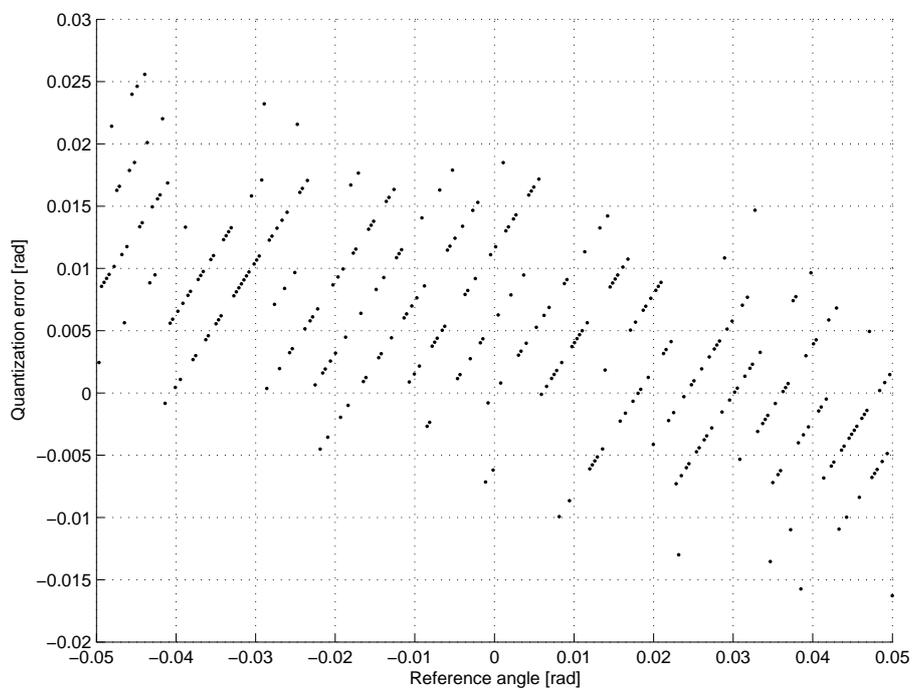


Figure B.3: Sensor quantization error

Appendix C

Walking Results

The dynamic model of the ERS-7 Aibo is tested with a walking pattern. The limits are tested and shown in this appendix. Some tests have been done without the floor, other tests checked the maximal frequency or stability for a longer testing time.

C.1 Long Simulation Walk

To investigate if The stability of the simulated walk is investigated for a longer period than ten seconds. The simulation duration is five minutes, where the Aibo makes 298 steps as shown in Figure C.1. The drop to initialize the contact model give the Aibo an offset, but stays constant. The vertical movement of the robot is also constant and the forward movement has a constant speed. Unfortunately this test has not been performed on the real Aibo, because the motion file would be too large.

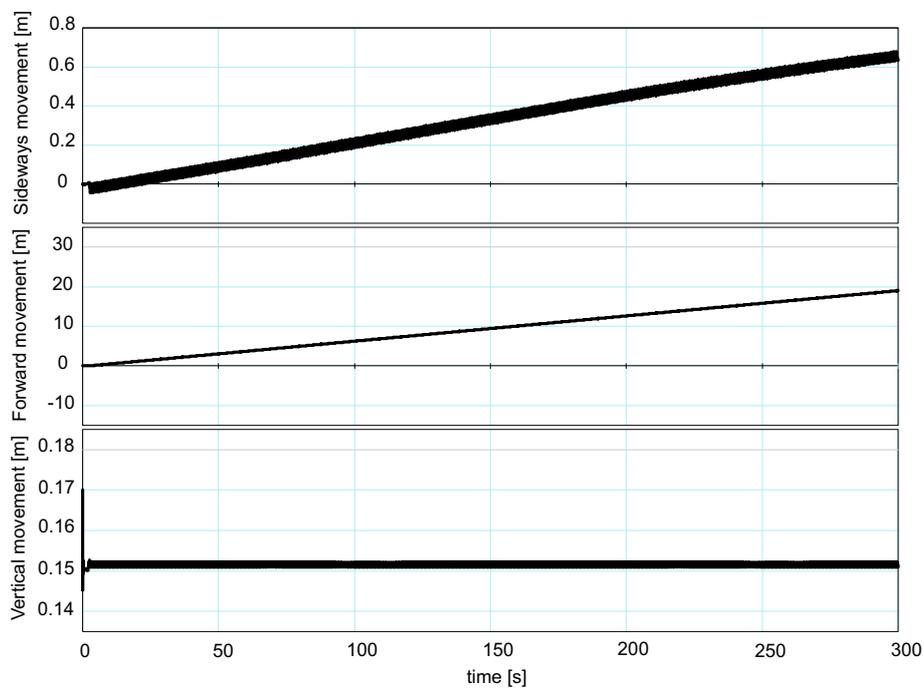


Figure C.1: Simulation of the Aibo model for 5 minutes. The sideways movement has an initial offset

C.2 Walking without the Floor

The influence of the floor on the real robot is tested. The Aibo is fixed in the air without touching the floor with its feet. In Figure C.2 the result is shown with the reference signal, the results from the simulation and the results of the measurements on the real Aibo. The result shows that the motor of the real Aibo and the simulated Aibo can handle the speed and amplitude of the reference signal.

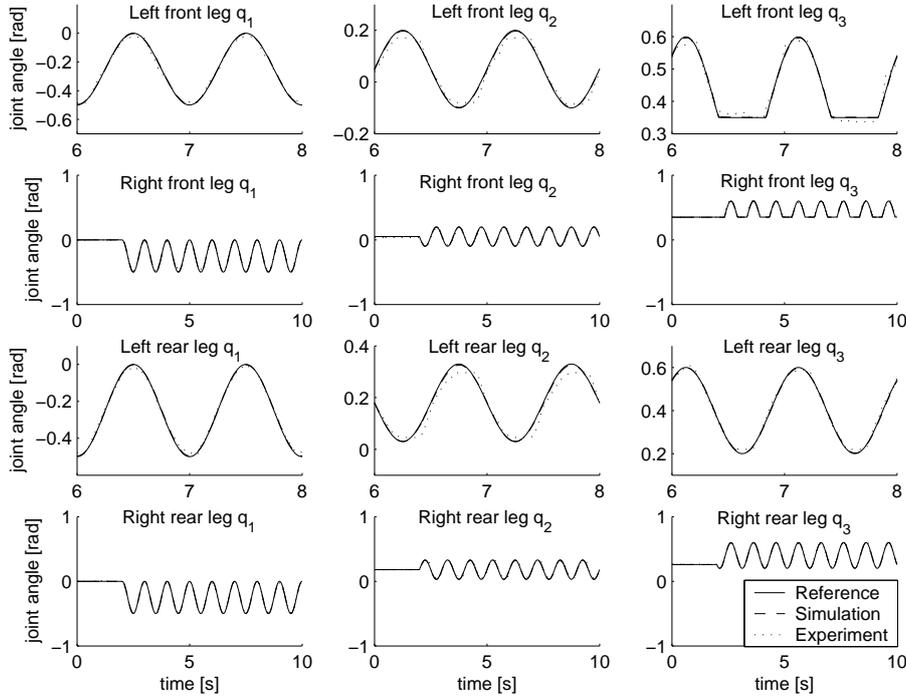


Figure C.2: Simulation and real results compared. The reference signal is an one Hertz walking gait. The simulations and real measurements are done without the influence of the floor on the feet

C.3 Higher Frequency Walking Patterns

The frequency of the walking algorithm is changed from one to two Hertz and tested, to see the effect on the real Aibo. The setpoints in the motion file can only handle a sampling frequency of 128 [ms], which is not enough to a continuous reference signal. The real Aibo stumbles several times during the walk due to this effect and a plot is made in Figure C.3 to show the measured joint angles.

The sampling time of the motion file is decreased from 128 to 96 [ms], which is under the advised sampling time, the Aibo crashes every time during the loading of the motion file. When the walking pattern or algorithm is implemented in a C++ file, the sampling frequency can be 8 [ms], but is not tested in this project.

C.4 Other Integration Methods

Different integration methods are used to test the robustness of the model. The results of the methods should be the same in order to proof that the end result of each integration is correct. Runge-Kutta 4 is used with a fixed stepsize, while there are other methods that are more flexible. The model is tested with the Modified Backward Differentiation Formula (MBDF) and the result is shown in Figure C.4. The integration method Vode Adams is also used and this result is in Figure C.5. The results from the

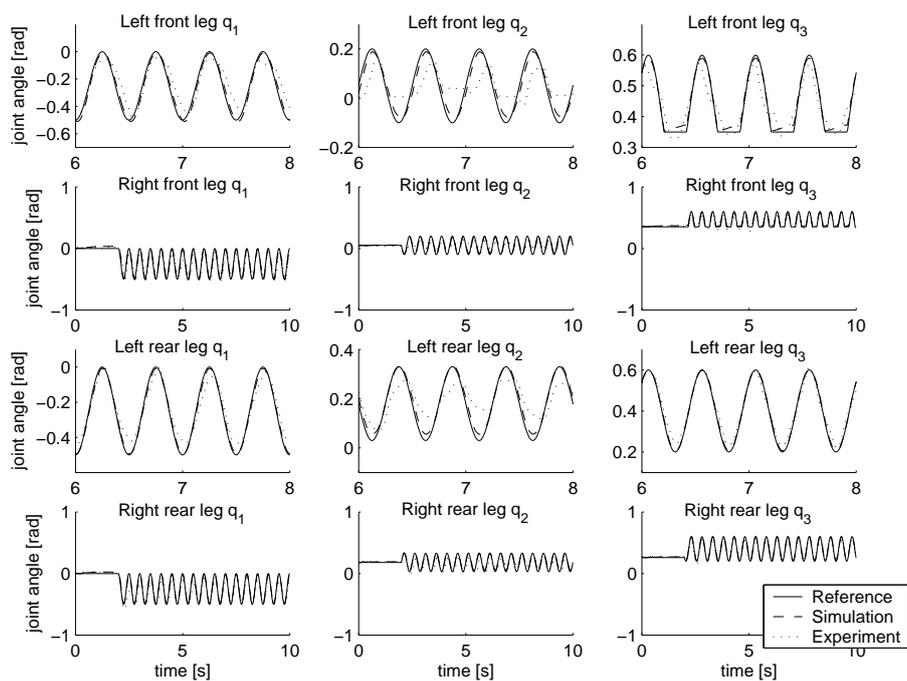
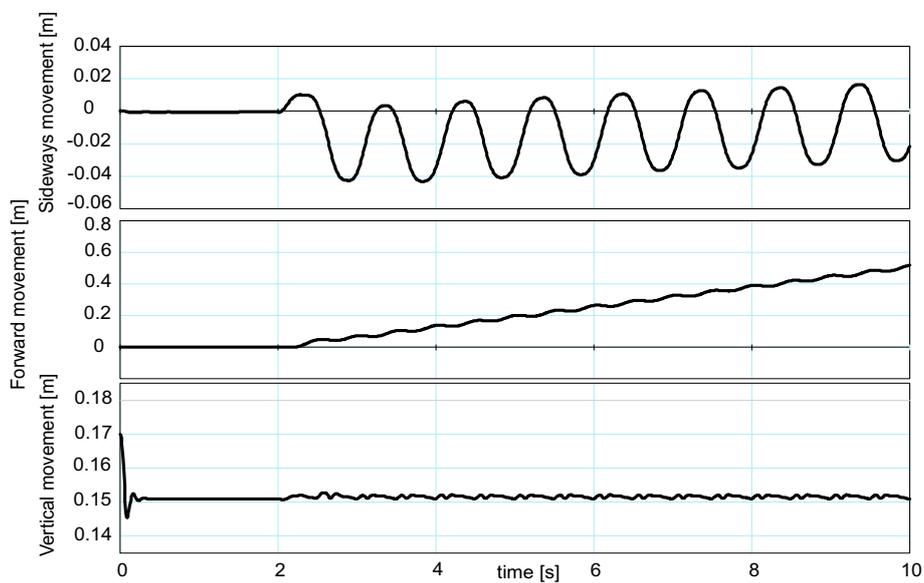


Figure C.3: Simulation of the Aibo model with a two Hertz walk

different integration methods show the same movement of the Aibo. This shows that the integration method does not influence the result, only the integration stepsize must be small enough for the contact model of the floor.

Figure C.4: MBDF integration method (10^{-5}) for ten seconds with a displacement of 0.517 [m]

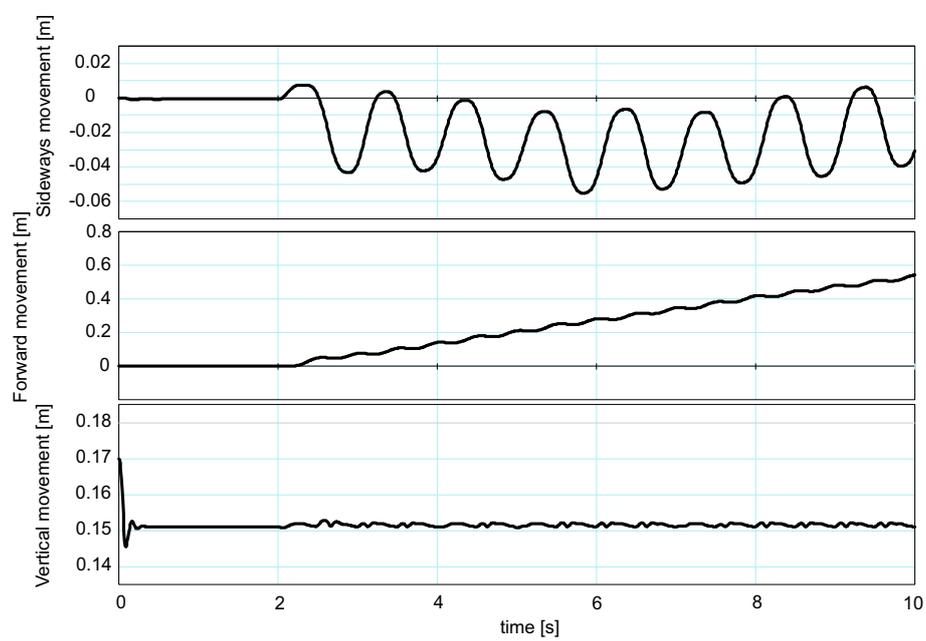


Figure C.5: Vode Adams integration method (10^{-6}) for ten seconds with a displacement of 0.542 [m]