University of Liège - Faculty of engineering

Master thesis

Simulation of complex actuators

Author : Hubert Woszczyk
Promotor : Pr. Bernard Boigelot

Master thesis submitted for the degree of
Msc in Electrical Engineering

Academic year 2015-2016

Simulation of complex actuators

Hubert Woszczyk

Abstract

Lorem ipsum dolor...

Acknowledgements

I would like to express my sincere thanks to all those who provided me the possibility to complete this thesis.

First of all, I thank the professor B. Boigelot who made this thesis possible and helped me on various occasions.

I also wish to thank Grégory Di Carlo and Guillaume Lempereur, my fellow students who also worked on the robot.

Contents

List of Figures

Chapter 1

Introduction

1.1 Context

This thesis sprung from the participation of a team of students to the the ”Robocup” contest. Robocup is a robotic contest in which robots from all around the world compete in a game of football. There are various categories but our team will compete in the kidsize competition.

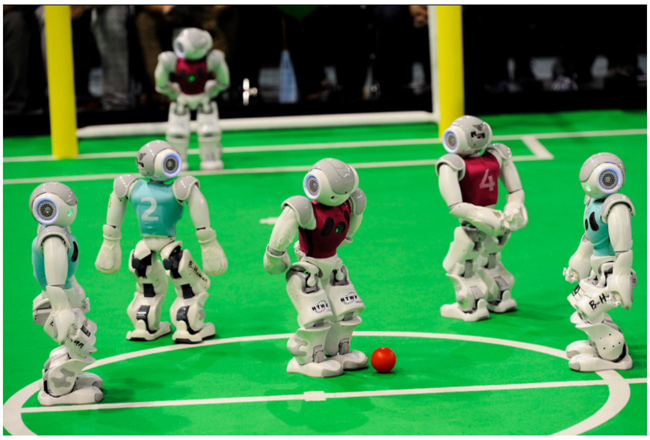

Figure 1.1: Two teams of Nao robots playing against each other in the 2014 edition of RoboCup *[Photo courtesy of RoboCup]*

1.2 Scope

The scope of this thesis is to provide the team with a simulation tool and a model able of :

- simulating realistic inertias
- receiving orders at approximately 300Hz
- simulate friction realistically
- incorporate springs and dampers
- visualization

The model should also receive the same orders as the real robot.

Chapter 2

Simulator

In this chapter we discuss the choice of V-rep as the simulation tool for this project. We begin by explaining the basics of rigid body dynamics simulation, take a survey of some of the existing simulators and finally test some of them.

2.1 Simulation of rigid body dynamics

2.2 Available simulators

The basis of the simulation is the ability to simulate the dynamics of the robot and the friction with the soil. So, something bare-bone as Matlab or bullet could be used,

[1] The list of physics simulating engines is quite long, but the most popular ones are, in no particular order :

1. Bullet
2. ODE
3. DART
4. Simbody
5. PhysX
6. Havok

Bullet was chosen because while it does not distinguish itself when it comes to pure physical simulation [2], a 3D modelling application called Blender is built atop of it, providing excellent tools for fast and easy robot modelisation. Blender also provides access to Bullet through a well document Python API.

2.3 Tested simulators

Blender is originally a 3D modelling and rendering software. It uses the Bullet physics engine to simulate realistic physics in animations. It has a Python interface which can be used to script a simulation.

Engine	License	Coordinates	Origin	Editor	Solver type
Bullet	Free	Maximal	Games	Blender	Iterative
ODE	Free	Maximal	Simplified robot dynamics, games		Iterative
DART	Free	Generalized	Computer graphics, robot control		
Simbody	Free	Generalized	Biomechamics		
PhysX	Proprietary	Maximal	Games		
Havok	Proprietary	Maximal	Games		

Table 2.1: Features comparison

Simulator	License	Physics engine(s)	Integrated editor	Modelling
Blender	Free	Bullet	Fully fledged	Internal
V-REP	Free (educational license)	Bullet, ODE, Newton, Vortex(10s limit)	Limited	Can import .COLLADA
Gazebo	Free	Bullet, ODE, Simbody, DART	Limited	SDF format
Webots	Proprietary	ODE	None	SDF format
Matlab	Proprietary	None	None	Mathematical

Table 2.2: Comparison of simulators

V-rep integrates a simulator that can run on different physics engines and supports remote operation.

2.4 Choice

Chapter 3

Modelling tools

This chapter covers the tools used in order to create a model of the robot, from the placement of the servos and joints to the incorporation of accelerometers.

3.1 Blender

The first stage of the modelling is done in Blender which is a lot more suited to this kind of work than V-Rep. Blender is used to do the following :

- place the servos, hinges and other elements in place.
- simplify the servos, hinges into simple convex shapes with a low vertex count.
- place position markers for the joints to be placed in V-Rep.

The model is finally exported in the COLLADA format.

3.2 V-REP

The model is finalized by :

- defining the mass and inertia of each piece(compiled in table 3.1) and enabling them for dynamic simulation.
- adding joints between servos. For 2DOF joints, hinges are used as intermediates.
- adding scripts to simulate sensors (COG, accelerometers).
- adding springs on the legs through the use of prismatic and spheric joints.

3.2.1 Servos

Servos are simulated by joints.

Module	Weight [g]	Dimensions [mm x mm x mm]
Odroid C-2	40	85.0 x 56.0
Li-Po battery	188	103.0 x 33.0 x 34.0
Mx-28R	72	35.6 x 50.6 x 35.5
LI-USB30-M021C	22	26.0 x 26.0 x 14.7
Frame Fr-07		
Frame Fr-101-H3	7	

Table 3.1: Weights and dimensions of the pieces of the robot

3.2.2 Joints

Spherical joint : 3DOF angular.

Prismatic joint : 1DOF linear.

Revolute joint : 1DOF angular.

3.2.3 Sensors (accel, cog)

The COG is computed through a script inside V-Rep, attached to a piece of the model and made available through the remote interface.

3.2.4 Springs

Chapter 4

Simulation

4.1 Interface (api)

4.2 First simple simulations

4.3 Robot design

4.4 Application ! stand up routines

Chapter 5

Physical validation

5.1 Mode expérimental

5.2 Experiments

5.3 Servo tuning

5.4 Results

Chapter 6

Conclusion

Bibliography

[1] Herman Bruyninckx. Blender for robotics and robotics for blender. *Dept. of Mechanical Engineering, KU Leuven, Belgium*, 2004.

[2] Tom Erez, Yuval Tassa, and Emanuel Todorov. Simulation tools for model-based robotics: Comparison of bullet, havok, mujoco, ode and physx.