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.

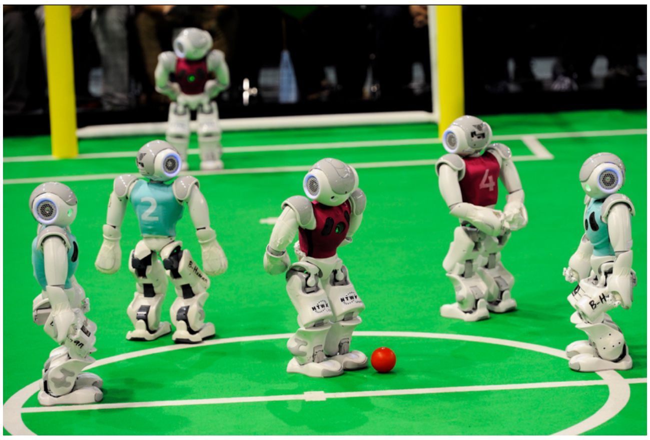

Figure 1.1: Two teams of Nao robots playing against each other in the 2014 edition of RoboCup *[Photo courtesy of RoboCup]*

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
- simulating friction realistically
- incorporating 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

TO DO

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

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[3]

2.2 Available simulators

An integrated simulation tool is preferred over a bare-bones physics engine because :

- time would be lost on creating 3D visualization
- time would be lost on writing code to import model
- time would be lost on debugging

and all that before the actual work could begin.

Blender[1] :

- Uses the Bullet engine
- Scripting via Python, remote control possible through socket
- Modelling tool readily available
- Comment : Hard to use because of obscure simulation options and difficulty to correctly set inertias

Gazebo :

- Can use Bullet, Newton or ODE.
- Scripting via C++
- Uses URDF format for models
- Comment : Hard to use because model must be in URDF format, which no CAD excepted 3dworks exports to. Furthermore, compiled language takes longer to test.

V-Rep:

- Can use Bullet, Newton or ODE.
- Internal scripting in LUA
- Can import 3D collada models.
- Comment : Best tool so far because model can be imported and the inertias are easy to control, simulation options as well.

Matlab:

- Analytical modelling
- Mathcode
- No visualization
- Comment : Not adapted because tedious modelling and no visualization and hard to handle friction and difficult to handle other objects.

2.3 Choice

Out of Gazebo, V-Rep and Blender, V-Rep is chosen as the best tool because

- Gives the choice between 3 engines, something blender cannot do

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

– Makes it easier than Gazebo to create models, because Gazebo uses the URDF format

– Gives better access than blender to the physical options of the simulation (intertias, timestep of engine)

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** $[g]$	**Density** $[g/cm^3]$	**Dimensions** $[mm \times mm \times mm]$
Odroid C-2	40		85.0 x 56.0
Li-Po battery	188	2000	103.0 x 33.0 x 34.0
Mx-28R	72	1150	35.6 x 50.6 x 35.5
LI-USB30-M021C	22	2200	26.0 x 26.0 x 14.7
Frame Fr-07		1200	
Frame Fr-101-H3	7	1200	

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[4].

3.2.4 Springs

Springs are simulated by prismatic and spherical joints.

Chapter 4

Simulation

4.1 Interface (api)

4.2 First simple simulations

4.3 Robot design

4.4 Application : stand up routines

[5]

Chapter 5

Physical validation

5.1 Experimental set-up

The set-up consists in :

– A camera that films the movements of a servo configuration.

– A simulation of that servo configuration in V-Rep.

5.2 Experiments

The first experiment is to test the torque : to that end, a frame is fixed onto a single servo and weighted. The setup is represented on fig. 5.1.

At $12V$, the maximal torque[2] of the servo is supposedly $2.5N.m$. To test this, a weight of $2kg$ is hanged at $12.5cm$ from the center of the servo, because since

$$\begin{aligned} 2.5 - 9.81 \cdot (0.007 \cdot 0.01 + 0.016 \cdot 0.0725) &= x \cdot 0.125 \\ x &= 20N \\ &= 2.03kg \end{aligned}$$

5.3 Servo tuning

5.4 Results

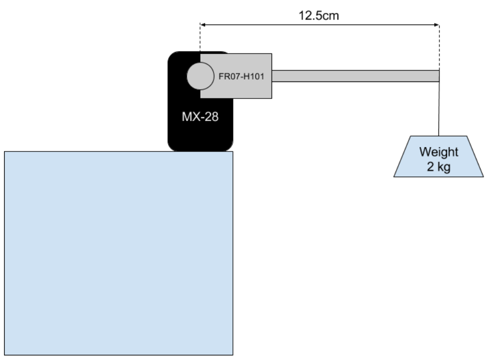

Figure 5.1: Experimental setup for torque testing : A weight of $2kg$ is suspended at $12.5cm$ from the servo, resulting in a torque of $2.5Nm$. The goal is to test whether the servo is able to move the weight upwards from the depicted initial situation.

Chapter 6

Conclusion

Bibliography

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

[2] Dynamixel. Robotis e-manual v1.27.00, 2016. [Online; accessed 18-April-2016].

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

[4] Coppelia Robotics. V-rep user manual 3.3.0, 2016. [Online; accessed 18-April-2016].

[5] Jörg Stückler, Johannes Schwenk, and Sven Behnke. Getting back on two feet: Reliable standing-up routines for a humanoid robot. In *IAS*, pages 676–685, 2006.