University of Liège - Faculty of engineering

Master thesis

Simulation of complex actuators

Author : Hubert Woszczyk
Promotor : Prof. Bernard Boigelot

Master thesis conducted for obtaining the Master's degree in *Electrical Engineering* by Hubert Woszczyk

Academic year 2015-2016

Simulation of complex actuators

Hubert Woszczyk, under the supervision of Prof. Bernard Boigelot

Academic year 2015-2016
Faculty of Applied Sciences
Electrical Engineering

Abstract

The word *robot* has been crafted by Czech writer Karel Čapek in the beginning of the XXth century and is derived from the slavic world *robota* which means *work*, as in *there is work to be done*. Much changed since those days, and this master thesis is about robots who prefer to play football rather than work in factories. This manuscript is the result of the planned participation of a team of students to the contest *RoboCup*. A team of humanoids robots need to be build and with no prior knowledge in that subject it would be difficult to build a functioning prototype on the first try. To avoid time expensive real-life experiments, a simulation tool is needed. We will take a survey of the existing simulators and choose the one that fits our needs best before using it to perform some basic simulations on the model of our robot. These simulations are used to detect design flaws that made the robot unable to stand up from a prone position or unable to walk. The end result of this work is the design of a robot that is able to stand and stand up when toppled over. Furthermore, this report serves as confirmation of various design decisions that were made beforehand such as the choice of the MX-28R as the servo to be used in the joints.

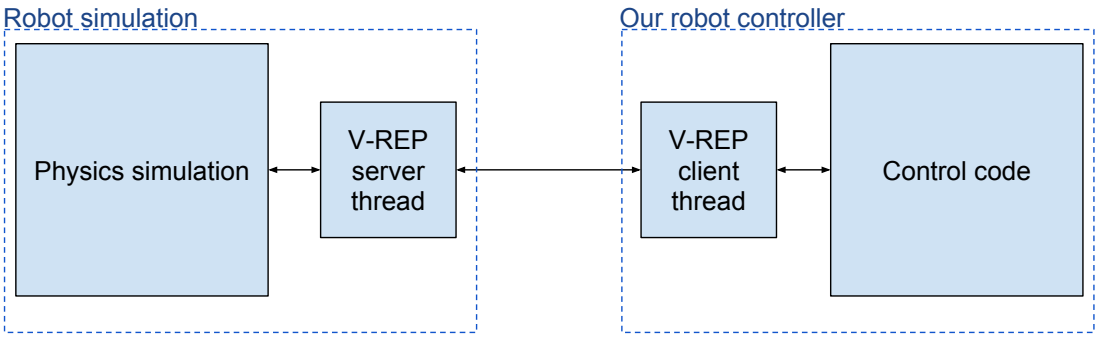
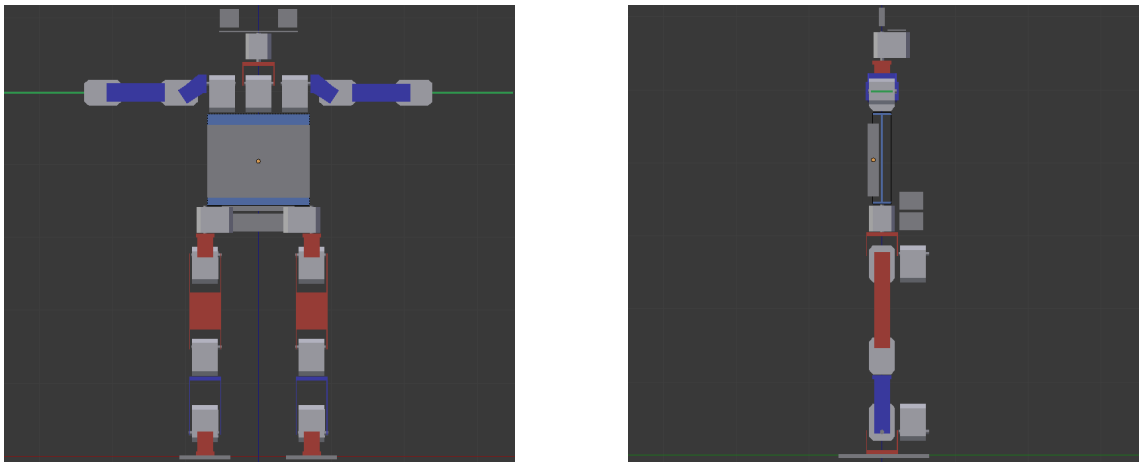


Figure 1: Representation of the architecture of the simulation setup. While the physics simulation is done in a simulator (V-Rep) along with the local control of the servos, the higher level control algorithms are executed outside. The communication between the simulator and high level control code is based on a TCP socket.



(a) Front view of the robot

(b) Side view of the robot

Figure 2: Front and side views of the final robot design this master thesis produced. The grey servos are connected together through different types of frames. At the top, two cameras are discernible and at the centre, the electronics and batteries can be seen.

Acknowledgements

My first thanks go to Prof. Bernard Boigelot who made it possible for numerous students, including me, to work in the passionate field of robotics. I also wish to thank him for his guidance, help and accessibility.

I am deeply grateful to my friends Elodie and Laurine for reading and correcting this manuscript. I also want to thank fellow students Grégory Di Carlo and Guillaume Lempereur with whom I had the pleasure of working together one last time.

Finally, I would like to express my sincere thanks to all those who helped me complete this master thesis.

Contents

List of Figures

List of Tables

Chapter 1

Conclusion

1.1 Conclusion

In this work we applied rigid body simulation to the problem of designing a robust humanoid robot. Our work produced a model of a robot that respects the rules of the kidsize league of the RoboCup Soccer competition. Our simulations prove that it is able to stand up from a lying position, be it prone or supine.

1.2 Problems encountered

A master thesis is a major endeavour and these are rarely devoid of obstacles. During this year several elements obstructed the completion of this work :

- V-Rep is a fine tool but the lack of a proper internal modelling tool was a major thorn in the side as every major modification meant that the whole model had to be modified in Blender and re-imported into V-Rep. This created a lot of overhead work which contributed nothing of interest to this work.

  This drawback is generalized amongst all the simulators that we surveyed at the beginning of this report and we feel it should be addressed quickly by their creators. Nevertheless, we understand that a modelling tool such as Blender took years to create so we would not expect simulators to catch up any time soon.

- We also learned of the importance of studying mechanisms carefully before using them. Though things might appear simple at first, subtle implementation details might change everything. A fine example of that is us melting the core of the motor inside a MX-28R servo during our tests because of our trust in the announced safety mechanisms. Needless to say, they proved insufficient and we should have examined the documentation more closely.

- Choosing a physics engine was difficult because the field is quite fragmented. On one hand there exist well established commercial solutions, but they are focused on games and make some significant shortcuts whenever possible in order to be as fast as possible. On the other hand there exist a quantity of open-source physics engine but they are usually the work of one man and are

poorly documented. It was hard to motivate the choice of Newton Dynamics on any other basis than 'it worked best'.

1.3 Future work

1.3.1 Modelling

While the model is in a usable state it could still be bettered and we suggest to begin with the items listed hereafter:

- **Springs.** Springs also need some work, as of now they are just there as a proof of concept but their parameters will need to be tuned.
- **Inertias.** As of now, the model uses simplified inertias, in the belief that a controller should be able to correct minor differences in behaviour between the model and the actual robot. If these inertias need to be made more accurate, we suggest to use Meshlab[1] to compute the inertias of those objects.
- **Model format.** It is still uncertain if Blender shall continue to support the COLLADA format, as explained mentioned in the development roadmap ([Ble15]). In the negative the choice should be made whether to continue using the COLLADA format and find another modelling software that supports it or to move on to another format (URDF for example, now that the robot model is defined).

1.3.2 Routines

Now that we have a simulator and a complete model of the robot, more routines can be created.

- **Standing up from a supine position.** Even though the robot can roll from a supine to a prone lying position and use the standing from prone routine it would be faster to be able to stand from a supine position directly.
- **Walking.** Being able to walk is the basic requirement for a robot to compete in RoboCup. A walking sequence is the last proof needed to be able to tell that the robot we designed is able to compete.
- **Shooting a ball.** As soon as the robot is able to walk, the next step should be testing if it can shoot a soccer ball.

1.3.3 Online simulation

In parallel or after creating the routines aforementioned, the simulator should be used to test the high level control code of the robot. The interaction between the simulator and the control code will be the same but the control code will be much more complex than just a static sequence of orders.

[1] http://meshlab.sourceforge.net/

Bibliography

[Ble15] Blender. 2.8 project developer kickoff meeting notes, 2015. https://code.blender.org/2015/11/the-2-8-project-for-developers/ [Accessed 01-May-2016].

[Rob15] Robocup. Robocup soccer humanoid league rules and setup, 2015.

Appendix A

Rules

The robots that participate in the kidsize competition must respect the following characteristics[Rob15]:

1. $40cm \leq H_{top} \leq 90cm$.
2. Maximum allowed weight is $20kg$.
3. Each foot must fit into a rectangle of area $(2.2 \cdot H_{com})^2/32$.
4. Considering the rectangle enclosing the convex hull of the foot, the ratio between the longest side of the rectangle and the shortest one, shall not exceed 2.5.
5. The robot must fit into a cylinder of diameter $0.55 \cdot H_{top}$.
6. The sum of the lengths of the two arms and the width of the torso at the shoulder must be less than $1.2 \cdot H_{top}$. The length of an arm is defined as the sum of the maximum length of any link that forms part of the arm. Both arms must be the same length.
7. The robot does not possess a configuration where it is extended longer than $1.5 \cdot H_{top}$.
8. The length of the legs H_{leg}, including the feet, satisfies $0.35 \cdot H_{top} \leq H_{leg} \leq 0.7 \cdot H_{top}$.
9. The height of the head H_{head}, including the neck, satisfies $0.05 \cdot H_{top} \leq H_{head} \leq 0.25 \cdot H_{top}$. H_{head} is defined as the vertical distance from the axis of the first arm joint at the shoulder to the top of the head.
10. The leg length is measured while the robot is standing up straight. The length is measured from the first rotating joint where its axis lies in the plane parallel to the standing ground to the tip of the foot.

Appendix B

Design guidelines

For a dynamic simulation several design restrictions must be considered :

- use pure convex as much as possible, they are much more stable and faster to simulate. When a more complex shape is used, approximate it with several convex shapes.
- use reasonable sizes, neither not too small nor too big. Thin shapes may behave strangely.
- when using joints, keep the ratio of the masses below 10. Otherwise, the joint may have large orientation/position errors.

Appendix C

Control code of the servo

```
1  if not PID_P then
2      PID_P=0.1
3      PID_I=0
4  end
5
6  if init then
7      pidCumulativeError=0
8  end
9  ctrl = errorValue*PID_P
10
11 if PID_I ~=0 then
12     pidCumulativeError = pidCumulativeError+errorValue*
           dynStepSize
13 else
14     pidCumulativeError=0
15 end
16
17 ctrl = ctrl + pidCumulativeError*PID_I
18
19 velocityToApply = ctrl/dynStepSize
20 if (velocityToApply > velUpperLimit) then
21     velocityToApply = velUpperLimit
22 end
23 if (velocityToApply < -velUpperLimit) then
24     velocityToApply = -velUpperLimit
25 end
26 forceOrTorqueToApply = maxForceTorque
27
28 return forceOrTorqueToApply, velocityToApply
```