VIETNAMESE-GERMAN UNIVERSITY

Electrical Engineering and Information Technology Department

PRE-THESIS PROJECT

Robotic Arm: Simulation on Linux Using Gazebo and ROS

Author:
Vu Hoi NGUYEN

Supervisor:
Prof. Peter NAUTH
Lab.Eng. Dinh Than LE

May 29, 2015

Contents

List of Figures

Abstract

This document is written as a requirement of the course "Elective project" for senior year in Vietnamese-German University. It reports the content of the project, my working process, the result and discuss the problems which are frequently occurs in the process.

My project studies the controlling methods providing by Robotics Operating System (ROS) and simulation sequence using Gazebo on Linux (specifically distribution Ubuntu).

The goal of the project is to successfully create a humanoid arm inside Gazebo and control its virtual joints using ROS.

To do this, a virtual-physical model need to be coded using a marking language (xml). A robot is divided in to single "links" (ex. upper arm, forearm, hand, ...), connected by "joints" (ex. elbow, wrist, ...). These links are constructed by the simplest 3D shapes like spheres or boxes for faster and easier simulating. This model also holds the physical properties of the links such as dimensions of the parts, their mass, initials; type of the links (revolve, linear, ...) and relationship between the links and the joints. Gazebo then takes care of the relationships and simulate the behavior of the robot in the virtual environment.

Furthermore, this model must have an appearance which is friendly to users (actual look of the robot, not just basic shapes). This "real" appearance is designed in another software, then exported (and imported into Gazebo) in form of a .stl file. Gazebo, again, take care of this change and display the new appearance of the robot into its 3D environment.

As the last part, "virtual" motors are attached to the joints of the model and controlled by ROS. Although these joints can be controlled directly inside Gazebo, using ROS have a critical importance, as standardized ROS controlling methods can as well be used in other environments (including real life). That way, ROS will significantly reduce the future work if one attends to apply this simulation to a real robot, which is obviously the purpose in doing simulations and will be surely done.

1 Introduction

1.1 Reasons for simulation

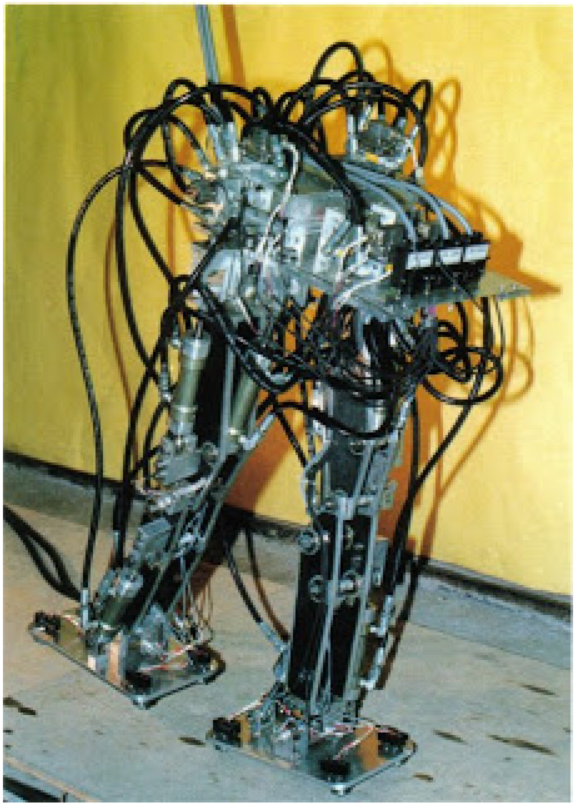

Figure 1: WL12RIII biped robot in 1989

1.2 Why Linux, Gazebo, ROS?

2 Working with ROS

2.1 Basic parts of ROS

2.2 Publishers and Subscribers

3 Gazebo Simulation

Abstract

4 Control Gazebo by ROS

Abstract

5 Project operating timeline

Abstract

6 Conclusion

Abstract

7 Reference

Introduction to Robotics: Mechanics and Control, John J. Craig, third edition https://www.sharelatex.com/learn/

```
// Hello.cpp
#include javax.swing.JApplet;
import java.awt.Graphics;

public class Hello extends JApplet {
    public void paintComponent(Graphics g) {
        g.drawString("Hello, world!", 65, 95);
    }
}
```