CMPEN 497 – Humanoid Robotics
Spring 2023

Instructors:
Dr. Yanxi Liu
Dr. Robert Collins

Project #4
Simulated Motion Design

Released on Friday, April 7, 2023
Dance file Due on Friday, April 21, 2023
Final submission Due on Friday, April 28, 2023

Addison Petro **amp6809@psu.edu**

Contents

1 Introduction

In this project, you will put your knowledge about humanoid robots and stability to use in a simulated environment. You will be creating an original dance that meets a certain set of specifications (listed later in the document), and you will be performing stability analysis using foot pressure sensor data. In addition, you will have to present your approach to this project in class. We will also be seeing if your dance behaves properly on the physical robot.

2 Software

For this project, you will be using Webots to make and test your dance. You will be using Python for the stability analysis, and the PyGLM library is required for vector math. Links to documentation for these tools are included below.

NOTE: For this project you must use Webots version 2021a!

Webots: `https://cyberbotics.com/`
Webots User Guide: `https://cyberbotics.com/doc/guide/index`
Webots Reference Manual: `https://cyberbotics.com/doc/reference/index`
Python: `https://www.python.org/`
PyGLM: `https://pypi.org/project/PyGLM/`

3 Project Overview

3.1 Starter Code

You are given a Webots world and controller code that outputs center of mass and center of pressure data. You are also given the Python programs required to do CoM to CoP analysis. Lastly, you are given a sample dance file that you can use as a reference for creating your own dance.

3.2 The Dance - Due Fri. April 21

You are to create a dance that meets the following specifications:

- Dance must be stable for at least 20 seconds
- Dance must be stable for at least 25 poses
- Before becoming unstable, each motor must move at least 10° (0.1745 rad) from the default pose
- Some time after the 20 second and 25 pose minimum, the robot must become unstable and fall over
- **Avoid self-intersection**

To create the dance, you should edit any of the .motion files provided by replacing, adding, and removing values as needed. These files can be opened in your text editor of choice.

The table below shows the limits of the robot joint angles in radians. Note that some combinations of angles cause the robot to intersect with itself, so you will need make sure sure self-intersection doesn't happen during your dance.

Motor	Min	Max	Default
RShoulderPitchMotor	0.00	3.14	1.57
RShoulderYawMotor	0.00	3.14	2.44
RElbowYawMotor	0.00	3.14	2.88
LShoulderPitchMotor	0.00	3.14	1.57
LShoulderYawMotor	0.00	3.14	0.70
LElbowYawMotor	0.00	3.14	0.26
RHipYawMotor	0.00	3.14	1.57
RHipPitchMotor	0.00	3.14	1.05
RKneePitchMotor	0.00	3.14	1.33
RAnklePitchMotor	0.00	3.14	1.92
RAnkleYawMotor	1.05	3.14	1.57
LHipYawMotor	0.00	3.14	1.57
LHipPitchMotor	0.00	3.14	2.09
LKneePitchMotor	0.00	3.14	1.81
LAnklePitchMotor	0.00	3.14	1.22
LAnkleYawMotor	0.00	2.09	1.57
NeckRollMotor	0.79	2.35	1.57

Table 1: Webots Model Joint Limits

3.3 Stability Analysis

You are given starter code for computing the CoM to CoP stability metric from only CoM and raw foot pressure data. These programs assume that there is at least some pressure, meaning that at least one of the pressure sensors is always in contact with the ground. However, when your robot falls over, this may not be the case. You must modify these programs to account for that. You are free to make your own design choices for handling these cases. You will document how you handle these cases in your writeup.

3.4 Presentation - In Class, Mon. April 24 / Wed. April 26

You will give a short presentation (roughly 5-10 minutes) on your project. This presentation should include details about your thought process for designing your dance(s) and your plans for computing stability (handling the cases described above). If you have already completed the stability computation part of this project, please present your results. After your presentation, I will show you how the actual robot behaves when running your dance. You will submit the files used in your presentation (PowerPoint, images, etc.) in your final submission.

3.5 Writeup

Like in previous projects, you will need to provide a writeup detailing your work. In your writeup, you should include information about any design decisions you made and any problems you faced. You will also need to provide information about how to use your program.

4 Requirements (150 Points)

Dance Submission - Fri. April 21

- Working Webots world, controller, and dance file (75 points)

- When submitting these, please use a .zip file as specified below

Presentation - In Class, Mon. April 24 / Wed. April 26

- 5-10 minute presentation with visuals (25 points)

Final Submission

- Include presentation materials (5 points)
- Stability analysis + working code (25 points)
- Writeup (20 points)

5 Submission Instructions

For your submission, you should create a zipped directory that contains working code and simulation files for your project (please verify that they work) as well as any other files you used throughout the project. Additionally, include a writeup that describes your work and any relevant information about design decisions. Please mention any extra credit you did in your writeup.

Dance submission files should be named: *LastName_FirstName_P4_Dance.zip*
Final submission files should be named: *LastName_FirstName_P4.zip*

Minimum included files in final submission:

- Webots world
- Controller code
- Dance file
- Generated runtime data (CoM, raw pressure)
- Stability code (modified starter code)
- Writeup
- Presentation files

6 Extra Credit (Max 20 points)

Possible extra credit ideas:

- 10 points per extra dance + stability computation (Extra dances can be completed after the presentation, just include them in the final submission)