Univ Lyon, Université Claude Bernard Lyon 1, Univ Gustave Eiffel, IFSTTAR, LBMC UMR_T9406

EquiSim
User's Manual

Toufik BENTALEB

February 24, 2021

Abstract

This manual is intended to help users to run and to know how and where they can perform any modification in the EquiSim software tool. This software tool was developed to allow human balance prevention and recovery to be performed using MATLAB and using linear model predictive control (LMPC) algorithm. A "Quick Start" approach to using this software will be presented, along with a detailed section containing full explanations and examples for using this tool.

Contents

1 Introduction

The aim of this document is to help users to use the MATLAB software tool for human/robot balance prevention and recovery. It is intended to provide all necessary information in a simple way. A "Quick Start" approach to using this tool will be presented, along with a detailed. The EquiSim user's manual contains a thorough description of all files in the software tool.

2 System Requirements

In order to get the code to execute properly, the following programs are required:

1. The program has been developed for use in the MATLAB environment and it is running perfectly on `MATLAB 2017b` (`MATLAB 9.03.0`).

2. To solve the quadratic programming problem using `quadprog` function the `OPTIMIZATION TOOLBOX` is required.

3 How files are organised

The folder `EquiSim_folder` contains all files necessary of the software tool, were divided into many categories:

- The main Matlab script file `main_EquiSim.m` to run the program.

- The folder `core_test` contains `classdef`, `function`, and `script` folders because the software was built using Object-Oriented (OO) techniques.

- The folder `core_MPC` contains `classdef`, `function`, and `script` folders.

- The folder `core_physical_model` contains `classdef`, `function`, and `script` folders.

- The folder `results` contains results graphs, for the test performed, in format of `*.JPG`, `*.EPS` and `*.PDF`.

4 Getting started with software

This section will take you through one example in order to get you started with the MATLAB software tool for human/robot balance prevention and recovery.

4.1 The main function

The function `main_EquiSim.m` is the only function visible in the file `EquiSim_folder` to run the software, and you can call it from the command line. Shown below (Fig. 1) is a part of the the main function and their parameters are described in the Table ?? and Table ??.

```
clear all, close all, clc,
path(pathdef); %clear the pathdef of include library

%addpath script/
addpath core_physical_model/function/ core_physical_model/classdef/ ...
    core_physical_model/script/
addpath core_test/function/ core_test/classdef/
addpath core_MPC/classdef/ core_MPC/classdef/linear_trajectories/ ...
    core_MPC/function/ core_MPC/script/

robot_type='human';
phase_duration_type='phase_duration_01';
walking_type=4;
cop_ref_type='ankle_center';
polyhedron_position='waist_center';
kinematic_limit='hexagonTranslation';
COM_form='comPolynomial';

run('core_test_all_axis/script/script_constant.m')
run('core_test_all_axis/script/script_init_storage_qp_result.m')
run('core_physical_model/script/script_init_storage_physical_model.m')
run('core_physical_model/script/script_init_storage_sensor_dynamics.m')
.
.
.
```

Figure 1: File saved as main_EquiSim.m

File/Variable/String	Description
robot_type	Name of a m-file script that contains the physical properties of a human or robot bodies: robot_type="human" $\rightarrow$ human model. robot_type="hrp4" $\rightarrow$ hrp4 robot model. robot_type="hrp2" $\rightarrow$ hrp2 robot model. robot_type="human" was chosen as a case study. The "human" string is used to call the m-file script human.m. The default Center of Mass (CoM) height h_com. COM height limits to the floor with respect to h^{com} are h^{com}_{max} and h^{com}_{min}. Initial standing state default with release angle θ, $\omega_0 = \sqrt{(g/(h^{com} \cdot \cos(\theta)))}$ and $\zeta_0 = 1/\omega_0^2$ where g is the gravity. The parameters in the file are: $x_0^{com} = [h^{com} \cdot \sin(\theta); 0; ((h^{com} \cdot \sin(\theta))/\zeta_0)]$; $y_0^{com} = [0; 0; 0]$; $z_0^{com} = [h^{com} \cdot \cos(\theta); 0; 0]$; The feet initial positions: $x^{step}_{r,0}$, $y^{step}_{r,0}$,$x^{step}_{l,0}$, and $y^{step}_{l,0}$.
phase_duration_type	Name of a script file that contains the duration and sampling time of the phases: phase_duration_r$\rightarrow$ Swing phase of the right foot. phase_duration_l$\rightarrow$ Swing phase of the left foot. phase_duration_b$\rightarrow$ Double support phase. phase_duration_RT$\rightarrow$ Reaction time. phase_duration_APA$\rightarrow$ Anticipatory postural adjustments (APA) phase. phase_duration_start$\rightarrow$Starting phase. phase_duration_stop$\rightarrow$ Stop phase. N_r$\rightarrow$ Right foot swing phase sampling time. N_l$\rightarrow$ Left foot swing phase sampling time. N_b$\rightarrow$ Double support phase sampling time. N_RT$\rightarrow$ sampling time of the reaction time. N_APA$\rightarrow$ Anticipatory postural adjustments (APA) phase sampling time.

Table 1: The constant parameters of the simulation case (part 1)

File/Variable/String	Description
`walking_type`	to select which type of walking area `walking_type`→ walking flat. `walking_type`→ walking airbus stairs. `walking_type`→ walking flat quick. `walking_type`→ walking walking flat fixed foot step positions. `walking_type`→ walking airbus stairs fixed foot step positions.
`cop_ref_type`	translate the step position to the `cop_ref_type`→ `'ankle_center'` Center of pressure reference centered on the ankle. `cop_ref_type`→ `'foot_center'` Center of pressure reference centered on the middle of the foot.
`polyhedron_position`	Polyhedron centered: `polyhedron_position`→ `'ankle_center'` on the ankle. `polyhedron_position`→ `'foot_center'` hexagon kinematic limits. `polyhedron_position`→ `'hexagonTranslation'` hexagon kinematic limits with translation.
`kinematic_limit`	The kinematic limit: `kinematic_limit`→ `''` very simple polyhedron. `kinematic_limit`→ `'hexagon'` on the middle of the foot. `kinematic_limit`→ `'waist_center'` on the middle of the waist.
`COM_form`	COM trajectory form: `COM_form`→ `'comPolynomial'` COM with piece-wise jerk. `COM_form`→ `'comExponential'` ZMP with piece-wise velocity. `COM_form`→ `'comPolyExpo'` COM with polynomial of exponential.
`firstSS`	First-foot stepping: `firstSS`→ `'r'` Right foot. `firstSS`→ `'l'` Left foot. `firstSS`→ `'b'` Both feet.
`nb_foot_step`	Number of steps.

Table 2: The constant parameters of the simulation case (part 2)

5 Script files

In the beginning, the main function main_EquiSim calls many other scripts to load human model parameters, creating the experiment test, preparation of storage data from tree type of physical models such as the model used in MPC controller, the physical model without noise, and the physical model with noise.

5.1 script_constant

First, this script m-file calls the object-oriented script classdef_create_robot to define the human model parameters

Parameters	**Description**
h_com	CoM height
nb_foot_step	Number of steps.

Table 3: The constant parameters classdef_create_robot

5.2 script_init_storage_qp_result

5.3 script_init_storage_physical_model

5.4 script_init_storage_sensor_dynamics

6 The simulation case

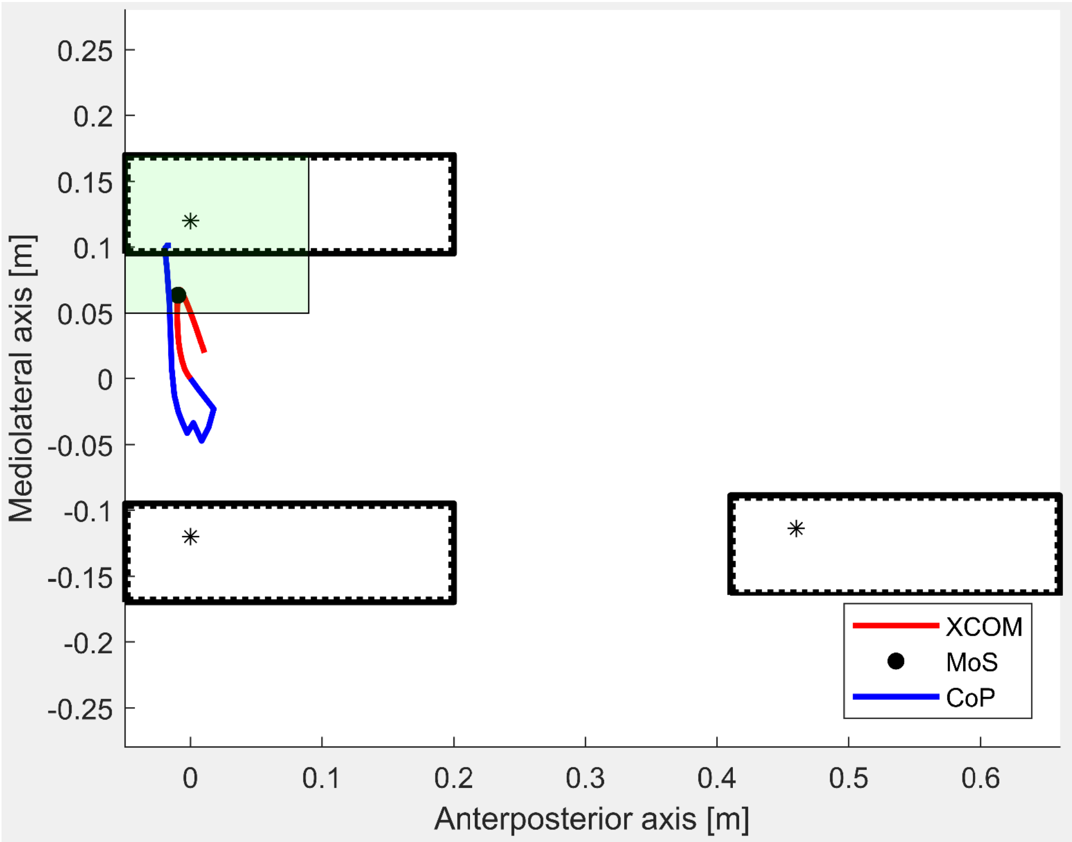

Figure 2: bode diagram