Developing a Walking Controller for a Three-link 2D Biped

Jessica Lanini, Andrea Di Russo, Dimitar Stanev, Laura Paez,
Vivek Ramachandran, Mehmet Mutlu and Auke Ijspeert
{jessica.lanini - andrea.dirusso - dimitar.stanev - laura.paez - ramachandran.vivek - mehmet.mutlu}@epfl.ch

Start: 24/09/2019 - Deadline: 20/12/2019

Introduction

I am sure by now you have seen many legged robots. The question is, if I give you a legged robot how would you make it walk? Where do you start? Of course, there is an interface where you can receive sensory data and send commands, but will you start by sitting and sending commands and see what happens? No. So, what is the starting point? The starting point is to model, then design control and finally simulate. The whole point of the mini-project is to realize the following **Control Design Pipeline**:

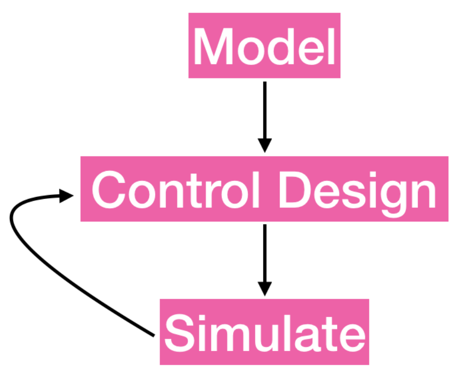

Figure 1: **Control Design Pipeline**

In order to understand the main principles behind the control design, we will work with a simple model: the **Three-link 2D Biped**, as represented in Fig. 2.

The project will be divided into three main parts:

- Modelling and visualization of the 3-link;
- Solving the equations of motion of the 3-link biped (simulation);
- Design of two different walking controllers, evaluate the resultant gaits and compare the performances.

You are asked to **develop these three parts and report your methods, results and observations in a final report**.

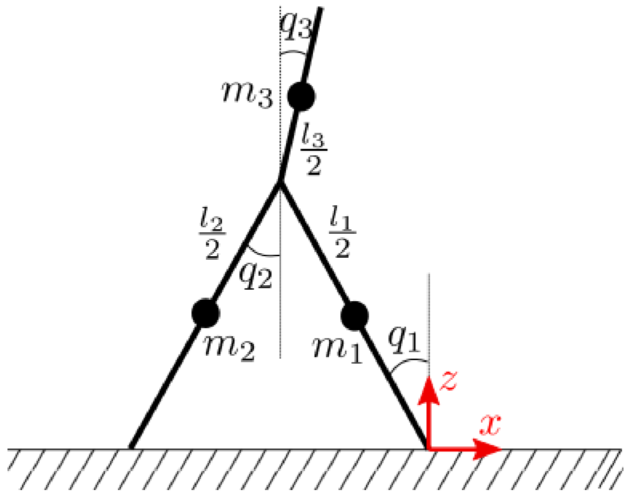

Figure 2: **Three Link Biped**

You are asked to structure your report according to the following sections which will be evaluated with the corresponding percentages:

- Introduction : 3%
- Methods : 5%
- Results : 40% (kinematics and dynamics: 10%; controllers: 30%)
- Discussion : 50%
- Conclusion : 2%

In this lecture and in the next you will complete the first part: *Modelling and visualization of the 3-link.*

Exercise 2.1: Kinematic

In this first part you are asked to complete the following matlab files:

- generate_kinematics.mlx (in the generate_model" folder)
- visualize.m

In order to implement the kinematic of the 3-link model please refer to the Exercise 1.1.
The function **visualize.m** should plot a schematic of the 3-link biped, and it is used to help you verify your code. Later this function will be used for making an animation of the 3-link biped simulation. To generate the kinematics use the generalized coordinates q=[q1;q2;q3] as shown in Fig. 2. Pay careful attention to the positive direction of the angles and the inertial coordinate system at the stance foot (see Fig. 2).

Exercise 2.2: Dynamic

In this second part you are asked to complete the following matlab files:

- generate_dynamics.mlx (in the "generate_model" folder)
- eval_M.m, eval_C.m, eval_M.m, eval_G.m, eval_B.m (in the "dynamics" folder)

In order to implement the dynamics of the 3-link model please refer to the Exercise 1.1.
By the end of this task, you have calculated the formulas for the matrices M, C, G, and B in the equations of motion:

$$M\ddot{q} + C\dot{q} + G(q) = Bu \tag{1}$$

where, u=[u1;u2] is the control vector. By running the generate_dynamics.mlx you should see new files in the dynamics folder, named x_tmp.m. From those files you can implement eval_M.m, eval_C.m, eval_M.m, eval_G.m, eval_B.m (in the "dynamics" folder). Those functions can be then used to calculate the matrices M,C,G and B.

Exercise 2.3: Impact map

This exercise is based on the explanation of the impact map analysis we did in class. You are asked to complete the following matlab files:

- generate_impact_map.mlx (in the "generate_model" folder)
- eval_A_m.m, eval_A_p.m (in the "dynamics" folder)
- impact.m (in the "dynamics" folder)

In the generate_impact_map.mlx file you can find two lines of code which allow you to convert your result in a matlab files (in particular matlabFunction(A_m, 'File', '../dynamics/Am_tmp.m'); matlabFunction(A_p, 'File', '../dynamics/Ap_tmp.m');). These tmp files will be useful for you to complete eval_A_m.m, eval_A_p.m.

NB: At the end of generate_impact_map.mlx you can find some questions that you should address in the final report! To answer those questions you may need to complete the eval_energy.m file.

Exercise 2.4: Test your results!

Make sure that your results for the previous parts are correct!
To this end test the kinematics and the dynamics implementation through the *test_dynamics.m* function in the "test" folder.