Modeling of a three-link 2D biped

Dimitar Stanev, Andrea Di Russo, Laura Paez, Alireza Manzoori Mohammad Askari,
Auke Ijspeert
{dimitar.stanev/andrea.dirusso/laura.paez/ali.manzoori/mohammad.askari/auke.ijspeert}@epfl.ch

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**:

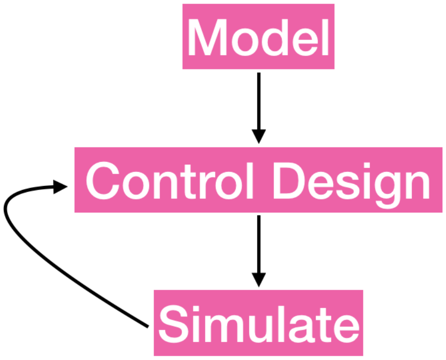

Figure 1: **Control Design Pipeline**

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 Figure 2. The project objectives can be divided into three main parts:

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

Each week, students should submit their answer to the tasks so that we can follow their progress and help them solve their issues during the practical sessions. At the end of the course, you will be asked to submit a final report, your code and demonstration videos related to the topics that will be covered during the course. It is advised to structure your final report according to the following sections:

- Introduction
- Methods

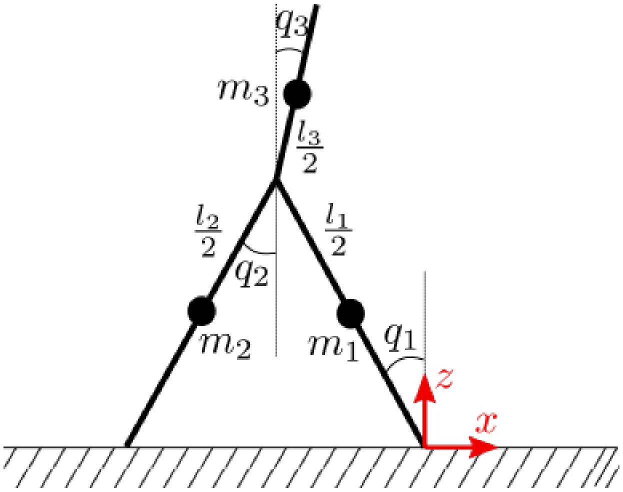

Figure 2: **Three Link Biped**

- Results
- Discussion
- Conclusion

In this and in the next lecture 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 kinematics 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 $\boldsymbol{q} = [q_1; q_2; q_3]$ as shown in Figure 2. Pay attention to the positive direction of the angles and the inertial coordinate system at the stance foot (see Figure 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_G.m*, *eval_B.m* (in the "dynamics" folder)

By the end of this task, you have calculated the formulas for the matrices $\boldsymbol{M}$, $\boldsymbol{C}$, $\boldsymbol{G}$, and $\boldsymbol{B}$ in the equations of motion:

$$\boldsymbol{M}(\boldsymbol{q})\ddot{q} + \boldsymbol{C}(\boldsymbol{q}, \dot{q})\dot{q} + \boldsymbol{G}(\boldsymbol{q}) = \boldsymbol{B}\boldsymbol{u} \tag{1}$$

where, $\boldsymbol{u} = [u_1; u_2]$ 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_G.m*, *eval_B.m* (in the "dynamics" folder). Those functions can be then used to calculate the matrices $\boldsymbol{M}$, $\boldsymbol{C}$, $\boldsymbol{G}$, and $\boldsymbol{B}$.

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*. 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.