Control and optimization of a three-link 2D biped

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

Introduction

In this exercise, we will implement our first controller and then learn how to use optimization methods to determine a set of parameters that achieve different types of gait.

Exercise 4.1: implement a controller

There are many controllers that one could implement with the three-link 2D biped model. Here, we will demonstrate the use of the virtual constraint. We will define two constraints y_1 and y_2 that are a function of the model's coordinates. They represent the conditions that would help in producing a stable gait. Therefore, we will ensure that they become zero by closing the loop using proportional derivative (PD) control. In the following lectures, we will introduce methods such as zero dynamics, task-space and CPG-based control[1].

The robot has one degree of under-actuation because it has two actuators to control the three degrees of freedom. Thus, it is necessary to choose which degrees are controlled and which are free to evolve naturally. With u_1, we can control both the leaning angle of the torso and the stance foot, whereas with u_2 we can control the swing foot and the torso. We can define u_1 such that it directly controls the torso and indirectly help in the propulsion of the stance foot. Then, we can define u_2 to control only the swing foot. In that case, q_1 is considered "free" to follow the dynamics of the model. The analytical expressions of the virtual constraints are shown below:

$$\begin{aligned} y_1 &= q_3 - \alpha \\ y_2 &= -q_2 - q_1 \end{aligned} \tag{1}$$

where α is the desired lean angle. If we drive the first constraint to zero, then we ensure that $q_3 = \alpha$. With the second constraint, essentially we require that $q_2 \approx -q_1$. Please note that even if the two angles are equal through u_2, it is u_1 that can also disturb the equilibrium. Finally, to design the PD controller differentiates the constraints with respect to time

$$\begin{aligned} \dot{y}_1 &= \dot{q}_3 \\ \dot{y}_2 &= -\dot{q}_2 - \dot{q}_1 \end{aligned} \tag{2}$$

It follows that the PD controller has the following expression:

$$\begin{aligned} u_1 &= k_1^P y_1 + k_1^D \dot{y}_1 \\ u_2 &= k_2^P y_2 + k_2^D \dot{y}_2 \end{aligned}$$

where k_i^P and k_i^D denote the proportional and derivative gains. This controller has five hyper-parameters. Following the theoretical description, please complete the function *control.m*. Note that we have imposed a lower and upper bound of 30 Nm on the control $\boldsymbol{u}$. Default parameters are provided in *control_hyper_parameters.m*.

[1] Presents possible controllers: `http://web.eecs.umich.edu/~grizzle/biped_book_web/`

Exercise 4.2: gait metrics and evaluation

Evaluate your controller using the *analyse.m* function, considering:

- min-max velocities reached
- plots of the angles vs time
- velocity of the robot vs time
- displacement vs step number
- step frequency vs step number
- torques vs time
- normalized mean effort, defined as follows ($U_{max} = 30Nm$):

$$\text{effort} = \frac{1}{2TU_{max}} \sum_{i}^{T} (u_1^2(i) + u_2^2(i))$$

- cost of transport, defined as follows:

$$\text{CoT} = \frac{\text{effort}}{x_{hip}(T) - x_{hip}(1)}$$

- plot of $\dot{\boldsymbol{q}}$ vs $\boldsymbol{q}$ for all three angles

Exercise 4.3: optimize controller's hyper-parameters

The goal of this part is to learn how to determine the controller's parameters that achieve different types of gaits. To do so, we can construct an optimization problem. We must define the objective criterion that is a function of the model parameters. The optimizer[2] evaluates the objective function multiple times to determine the "optimal" set of parameters that minimize its value. The objective function must do the following (i) simulate the model using the input parameters, (ii) evaluate the quality of the gait, and (iii) return a value that indicates the "performance". Mathematically speaking, we are solving an unconstrained optimization

$$\underset{p}{\text{minimize}} f(p)$$

where p denotes the decision variable and $f(p)$ the objective function. In our case, the objective function can be defined as follows:

$$f(p) = w_1 |\dot{x}_{hip} - \bar{\dot{x}}_{hip}(p)| + w_2 \text{CoT}(p) \tag{3}$$

where $\dot{x}_{hip}$ denotes the desired horizontal hip velocity, $\bar{\dot{x}}_{hip}(p)$ the simulated mean velocity, and w_i are the weights. These weights influence the relative importance of each term, in the sense that they might be competing. For example, if we try to minimize the effort through CoT, then we would expect that the simulated velocity should be lower. On the contrary, if we want to obtain faster gaits, then the CoT should be higher.

One final point is that sometimes the optimizer will determine gaits that are inappropriate. For example, the model can take a very short step, which can make the CoT large (denominator). Also, the model might reach the target velocity, but the resulting gait can be unstable. It is our job to handle these corner cases.

To solve this task, please complete *optimization_fun.m* and *run_optimization.m*. Experiment with the design of the objective function Eq. 3. Obtain a different set of parameters that lead to different gaits (e.g., different speeds). Compare the two using the metrics defined in the previous exercise. **Please submit your code and your report for the three exercises before the deadline and also include these in your final report.**

[2] You can begin with *fminsearch*: `https://www.mathworks.com/help/matlab/ref/fminsearch.html`