Exploration of control strategies for 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 the previous assignments, we build the necessary tool to model, simulate, control, and optimize the controller's hyper-parameters to obtain different gaits. The goal of this exercise is to design your own controller and explore its characteristics. Here, you will extrapolate on what you have learned, be creative, and try new things autonomously. **The role of the TAs would be to brainstorm with you, give advice, raise questions, but not to provide you with the solution.** Real-life problems are ambiguous and sometimes there is not a unique or definite solution. Therefore, we think that an important trait is to be able to navigate ambiguity and come up with unique solutions.
Please upload your code, videos of your controller (could be funny as well), and the report before 11.12.2020.

Part 1: theory

The first part of the report should present the work you did in the previous assignments. Try not to include snippets of code in the report. **Also, make sure to include the questions and answers of the previous assignments.** Below you will find an indicative structure:

- Section 1.1: kinematics
- Section 1.2: dynamics (Lagrange, equations of motion, B)
- Section 1.3: impact map (do not forget the questions in the code)
- Section 1.4: model validation (tests) and animation
- Section 1.5: numerical integration (do not forget the additional questions)
- Section 1.6: control and optimization (controller, gait metrics, optimization criteria)

Part 2: implement your own controller

In this part, you will be asked to implement your own controller. **Please make sure that the controller is different from the one presented in the class. You can discuss your proposition with the TAs to check if the controller meets this requirement.** In the report, present the concept and theory of your implementation.

You will be asked to compare the performance of your controller with respect to the one proposed in the previous assignment. In particular, you should explore the following gait conditions on both controllers and obtain different hyper-parameters using optimization:

- the maximum number of steps that indicate stability (also use phase plots)
- the minimum and maximum steady-state gait velocity
- optimize for different step lengths (short, self-selected, and long)
- optimize for different step frequencies

- examine the robustness of the controller against external perturbations
- examine the robustness of the controller against internal perturbations (sensory noise)

In order to apply perturbations, you can modify the control function. For example, you can apply a torque directly on $\boldsymbol{\tau}$ on top of your controller (*control.m*). Determine the maximum perturbation (negative and positive) that the model can withstand for each joint independently. For example, the magnitude of the perturbation can be included as a decision-variable in your optimization. Since, the maximum value will depend on the gait phase, try to apply the perturbations when the model has reached steady-state and during early, mid, and late swing. For the internal perturbation, you can add noise to the sensory input $\boldsymbol{q}, \dot{\boldsymbol{q}}$. Similarly, determine the min and maximum bounds.
Hints:

- Since this robot does not have feet, clearly, ZMP-based approaches would fail here. In contrast, trajectory-based approaches or the methods discussed in the lecture might be helpful.
- Note that you have only two actuators and three degrees of freedom. Hence, the robot has one degree of under actuation. It is important to decide which variables or combination of variables you would like to fully control. Think about, which one of these seems more intuitive to directly control? Which one can be regarded as a “free variable”?
- In case you decide to continue with virtual constraints or trajectory control, you can consider using splines or Bezier polynomials for designing your virtual constraints (or trajectories).
- When designing your controller, be careful with the convention for the positive and negative signs.

Part 3: compete and reflect

In this part, **each group will briefly present their controller (1–2 minutes) to the rest of the students on 15.12.2020**. Each group will fill a common spreadsheet in advance using the values of the performance metrics that we defined in part 2. The ultimate goal is to reflect on the different control policies, understand why some controllers did well or not for a particular challenge, and have fun.