OPTIMIZATION, ROBOTICS & BIOMECHANICS (ORB)
INSTITUTE OF COMPUTER ENGINEERING (ZITI)
HEIDELBERG UNIVERSITY

Dr. Alexander Schubert,
Giorgos Marinou

Robotics 2 (SS 2022)

Exercise Sheet 3

Presentation during exercises in calendar week 22

Exercise 3.1 – Robotic dynamics as Boundary Value Problem

Robotic motions that are defined by an initial pose and a final pose can be interpreted as BVP. We want to find joint velocities, accelerations or torques that result in a desired end pose when starting from a defined configuration. In this exercise we want to control the humanoid robot iCub using torque control. Therefore we reduced the model to 2D. The provided model has 9 degrees of freedom, of which 6 can be actuated by torques. We want to compute the torques for a sit to stand motion where the robot stands up from a low squat position.

- Use your code from Sheet 1 and compute the start and end pose for iCub. In the starting pose the chest should be 40 cm above the ground. The end pose should be upright with almost stretched legs. Use forward and inverse kinematics to compute these poses and save them to an animation file. Make sure all joints look natural. How do the feet look in the animation?
- Use your results from Sheet 2 and define a rhs function that constraints the two feet. How many points do you need to constrain for a 2D model? Implement a forward simulation.
- Implement a simple single shooting algorithm that iteratively updates the torques for each joint based on the difference to the desired end pose.

$$u_{i+1} = u_i - \lambda * (q - q_{ref}) \tag{1}$$

- This algorithm is only works if you are already close to the solution and is not very robust, so feel free to ask for a good initial guess on the torques.

In the following exercises we solve different Boundary Value Problems (BVP) using the Multi Shooting Implementation Muscod II. **For using MUSCOD you will need to download the VirtualBox interface, where the software is provided.** A skeleton file is provided. You need to define your problem in the source file (SRC/bvpXX.cc) and the DAT file (DAT/bvpXX.dat). Build your files in the build directory and run the code using:
`muscod "name_of_DAT_file"`

Exercise 3.2 – Single Shooting Solve the following ODE for different boundary conditions:

$$\ddot{x} + x = 0 \rightarrow \begin{bmatrix} \dot{x} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} v \\ -x \end{bmatrix} \tag{2}$$

Boundary Condition 1:

$$x(0) = 0$$
$$x(\frac{3\pi}{2}) = 1$$

Boundary Condition 2:

$$x(0) = 0$$
$$x(\pi) = 0$$

Boundary Condition 3:

$$x(0) = 0$$
$$x(\pi) = 1$$

What is the general solution to the ODE? What do you expect for the different boundary conditions? What does the optimization code return?

Exercise 3.3 – Multiple Shooting Solve the following ODE for different boundary conditions:

$$\begin{bmatrix} \dot{x} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} v \\ \lambda_1 \cdot sinh(\lambda_1 \cdot x) \end{bmatrix} \tag{3}$$

Boundary Condition:

$$x(0) = 0$$
$$x(1) = 1$$

Vary the λ_1 parameter in the range of 1 to 20. How does the function change? Is Muscod always able to solve the BVP? How do you need to adjust your DAT file?