Exercise 3: Legged Robot Control

Prof. Marco Hutter
Teaching Assistants: Dario Bellicoso, Jan Carius*

November 13, 2018

Abstract

In this exercise you will learn how to implement control algorithms for a legged robot. All kinematic and dynamic quantities are already implemented, alongside with a simulation and visualization. You will implement controllers which make the robot follow a motion reference in the body while respecting contact constraints and applying a force with its end-effector. The partially implemented MATLAB scripts, as well as the visualizer are available.

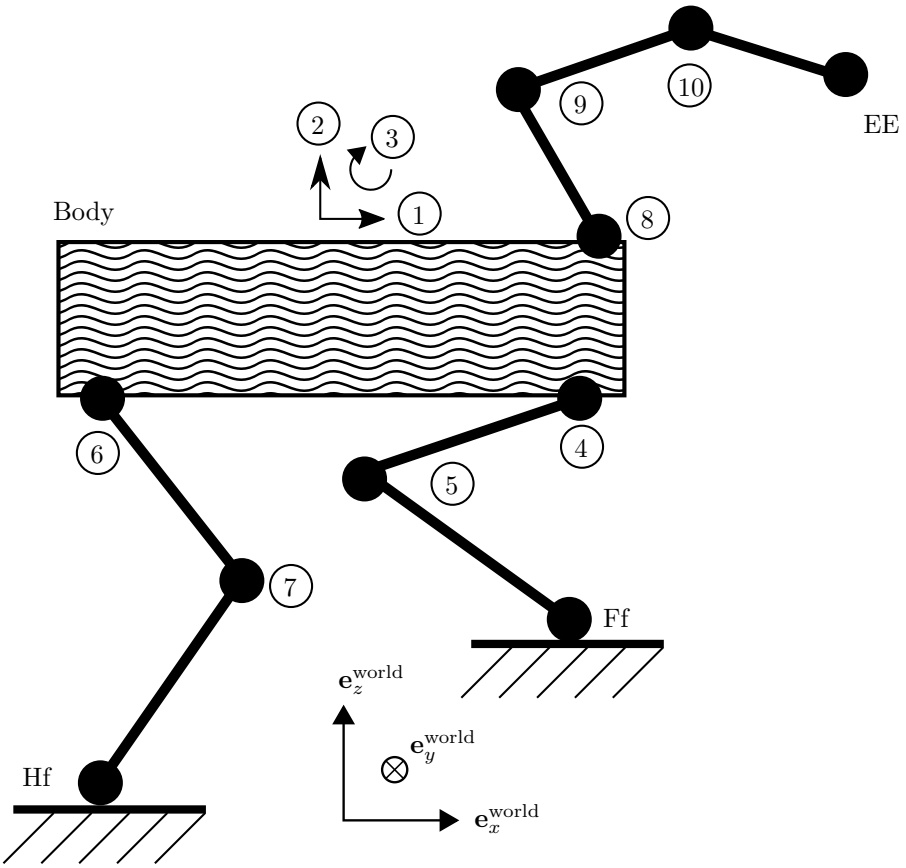

Figure 1: Schematic of a quadruped in two dimensions. The circled number indicate the ten degrees of freedom of this system. The extremities are called end-effector (EE), front foot (Ff), and hind foot (Hf).

*bellicoso@mavt.ethz.ch, jan.carius@mavt.ethz.ch

1 Introduction

The legged robot is schematically drawn in Fig. 1. All kinematic and dynamic quantities are already pre-computed and can be considered given for your implementation of the controllers. To initialize your workspace, run the `init_workspace.m` script.

Dynamics The dynamics of this free-floating system can be written as

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{b}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) = \mathbf{S}^\top \boldsymbol{\tau} + \mathbf{J}_{\mathrm{Ff}}(\mathbf{q})^\top \mathbf{f}_{\mathrm{Ff}} + \mathbf{J}_{\mathrm{Hf}}(\mathbf{q})^\top \mathbf{f}_{\mathrm{Hf}} + \mathbf{J}_{\mathrm{EE}}(\mathbf{q})^\top \mathbf{f}_{\mathrm{EE}} \ , \quad (1)$$

which is a system of ten equations. The actuation torques $\boldsymbol{\tau} \in \mathbb{R}^7$ cannot directly influence the body's degrees of freedom, therefore the selection matrix reads

$$\mathbf{S}^\top = \begin{bmatrix} \mathbf{0}_{3\times 7} \\ \mathbb{I}_{7\times 7} \end{bmatrix} \ . \quad (2)$$

Owing to the planar nature of our problem, the task space only has three degrees of freedom, namely x and z translation as well as rotation around the y axis.
For a more conventient notation, we also define the constraint Jacobian, $\mathbf{J}_c \in \mathbb{R}^{4\times 10}$, as the stacked linear parts of the foot Jacobians:

$$\mathbf{J}_c := \begin{bmatrix} \mathbf{J}_{\mathrm{Ff,\ xz}} \\ \mathbf{J}_{\mathrm{Hf,\ xz}} \end{bmatrix} \ , \quad (3)$$

with the four corresponding constraint forces defined as

$$\mathbf{f}_c := \begin{bmatrix} \mathbf{f}_{\mathrm{Ff,\ xz}} \\ \mathbf{f}_{\mathrm{Hf,\ xz}} \end{bmatrix} \ , \quad (4)$$

where the xz subscript refers to the linear components in x and z directions.

2 Control

In this section, you will write two model-based controllers for different tracking tasks. We will use hierarchical control formulated as quadratic problems (see script Section 3.10.4). Furthermore, we will use dynamic model-based control, hence our control commands are on torque level.
The hierarchical optimization procedure is already implemented for you. Your task involves formulating the optimization problem. Using the notation from the lecture, each objective i is written in the form

$$\min_{\mathbf{x}} ||\mathbf{A}_i\mathbf{x} - \mathbf{b}_i|| \ , \quad (5)$$

hence you need to specify numerical values for $\mathbf{A}_i$ and $\mathbf{b}_i$. For each optimization problem, you may also add additional constraints of the form

$$\mathbf{C}_i\mathbf{x} \leq \mathbf{d}_i \ . \quad (6)$$

2.1 Constraint Consistent Body Motion

Exercise 2.1

In this exercise, you will implement the controller `floating_body_control.m` which moves the body according to a provided control reference (position, orientation) while keeping the feet on the ground. The decision variables are $\mathbf{x} = [\ddot{\mathbf{q}}^\top, \mathbf{f}_c^\top, \boldsymbol{\tau}^\top]^\top$. You may assume that $\mathbf{f}_{\text{EE}} \equiv \mathbf{0}$, and that the floor does not introduce reaction moments, i.e. $\mathbf{f}_{\text{Ff},\,\theta} = \mathbf{f}_{\text{Hf},\,\theta} = \mathbf{0}$. The equations of motion for this particular setup can thus be written as

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{b}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) = \mathbf{S}^\top \boldsymbol{\tau} + \mathbf{J}_c(\mathbf{q})^\top \mathbf{f}_c \,. \tag{7}$$

The following hierarchy of tasks is to be implemented

1. Fulfill equations of motion
2. Ensure feet remain stationary on the ground
3. Track the desired body motion

Hint: Task 2. and 3. can be formulated as task space control with the task space acceleration, $\dot{\mathbf{w}}_{task}^{des}$, to be designed for the specific task.

$$\dot{\mathbf{w}}_{\text{task}}^{\text{des}} = \mathbf{J}_{\text{task}}\ddot{\mathbf{q}} + \dot{\mathbf{J}}_{\text{task}}\dot{\mathbf{q}} \tag{8}$$

You can test your controller by running `simulation.m` .

```
function [ tau ] = floating_body_control(model, t, state)
% Implement a controller that controls the floating body of the ...
    legged system
% Outputs:
%   tau : [7x1] torques [tau_F, tau_H, tau_A]'
%       tau_F : [2x1] torques [hip, knee]' (front leg)
%       tau_H : [2X1] torques [hip, knee]' (hind leg)
%       tau_A : [3x1] torques [shoulder, elbow, wrist]' (arm)
%
% Inputs:
%   model : object containing kinematic and dynamic quantities
%   t : Current time [s]
%   state : Current state of the system (position, velocity)
```

2.2 Hybrid Force-Motion Control

In the same framework of hierarchical quadratic programming, we can also use hybrid force-motion control.

Exercise 2.2

Similar to the previous task, we now want to use the end-effector to additionally push against a wall at a fixed spot. We therefore extend our decision variable to $\mathbf{x} = [\ddot{\mathbf{q}}^\top, \mathbf{f}_c^\top, \boldsymbol{\tau}^\top, \mathbf{f}_{\text{EE}}^\top]^\top$. You should place your code in `hybrid_force_control.m` and test it by running `simulation_with_wall.m` .

With the addition of end-effector forces the equations of motion become

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{b}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) = \mathbf{S}^\top \boldsymbol{\tau} + \mathbf{J}_c(\mathbf{q})^\top \mathbf{f}_c + \mathbf{J}_{\text{EE}}(\mathbf{q})^\top \mathbf{f}_{\text{EE}} \,. \tag{9}$$

Consider the following hierarchy of tasks

1. Fulfill equations of motion
2. Ensure feet remain stationary on the ground

3. Apply a force against the wall in the x direction (other end-effector force components to remain zero)
4. Keep the end-effector at a fixed position on the wall in the z direction
5. Move the body according to the reference

```
1 function [ tau ] = hybrid_force_control(model, t, state)
```