Legged Robots: Assignment 4
Designing a Walking Controller for the Three-link Biped

Group №6 : Darko Lukic, Leonardo Cencetti

June 7, 2019

1 Introduction

This report provides information regarding the bipedal walking controller developed during the mini-project. This report will pay particular attention to the controlling aspects of the model, as the dynamics and kinematics have been already addressed in previous sessions. All the information provided will be supported by tables, figures and plots to better explain the topic.

2 The model

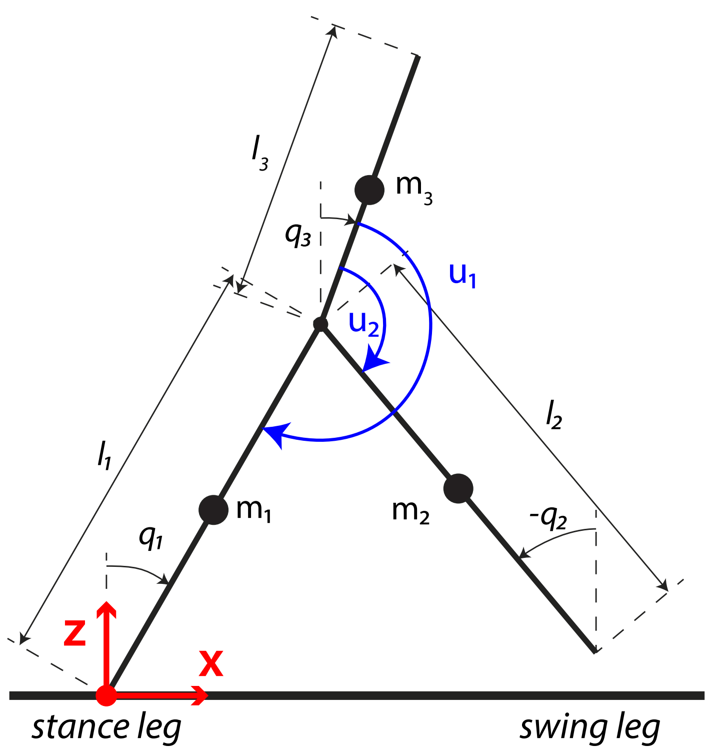

Figure 1: Illustrative representation of the three-link biped model used in the project.

The model used for this project is the Three-Link Bipedal robot, a simplified humanoid model, constrained to the 2D space. The kinematics and dynamics of the model account for both gravity (pendulum effect) and impact with the ground's surface. However, this model requires constant contact with the ground and therefore does not allow gaits with a flight phase (e.g. running). The model consists in three limbs, namely the swing leg, the stance leg and the torso, each one with its mass, represented as a point mass in the geometric center of the limb. The legs are of length $l1 = l2 = 0.5m$ and mass $m1 = m2 = 7kg$, while the torso has length $l3 = 0.35m$ and mass $m3 = 17kg$. The angles $q1$, $q2$ and $q3$ are taken with respect to the Z axis, and positive in the clockwise direction. The torques $u1$ and $u2$ are considered positive in the clockwise direction according to the right-hand rule. The joint connecting the three limbs will be referred to as the *hip* of the robot, while the extremities of the swing and stance legs will be referred to as *swing foot* and *stance foot* respectively. The equation of motion of the model were obtained with Lagrangian mechanics.

3 Trajectory

The model has one degree of under-actuation therefore two actuators are controlled. A multiple trajectories are designed and evaluated and in the report trajectory with the best result[1] is presented. The first objective is to keep an angle of the torso equal to some given angle (t_t):

$$q_3 - t_t = 0 \tag{1}$$

The second objective is to determine a trajectory of swing leg. For this purpose virtual constrain is used:

$$q_2 + t_{sw} * tanh(pi * q_1/t_{sw}) = 0 \tag{2}$$

A purpose of *tanh* function is to keep a swing leg away from the ground.

In order to use a controller with D property a derivative of the given trajectory (Equation 2) is required:

$$q_2 + pi * (sech(pi * q_1/t_{sw}))^2 = 0 \tag{3}$$

A visual representation of the virtual constraints (Equation 2 and 4) is presented in figure 2.

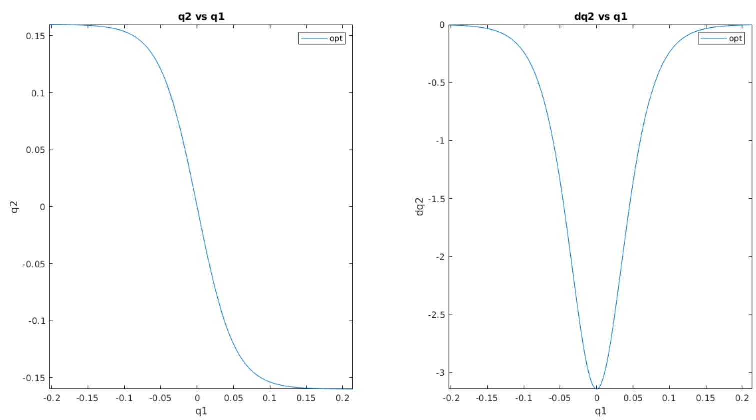

Figure 2: Trajectory of swing leg (q_2).

4 Controller

By using the equations of trajectory the following controller obtained:

$$u_1 = kp_s * (q_2 + t_{sw} * tanh(pi * q_1/t_{sw})) + kd_s * (q_2 + pi * (sech(pi * q_1/t_{sw}))^2) \tag{4}$$

[1]Trajectory with which we were able to keep torques under 30 Nm, but still to achieve speed of around 1 m/s

$$u_2 = kp_t * (q_3 - t_t) + kd_t * q_3 \quad (5)$$

The controller is implemented and the following parameters are observed:

- the angles vs time,
- velocity of the robot vs time,
- displacement in each step vs step number,
- step frequency vs step number,
- torques vs time,
- cost of transport and
- plots of q vs dq for all three angles.

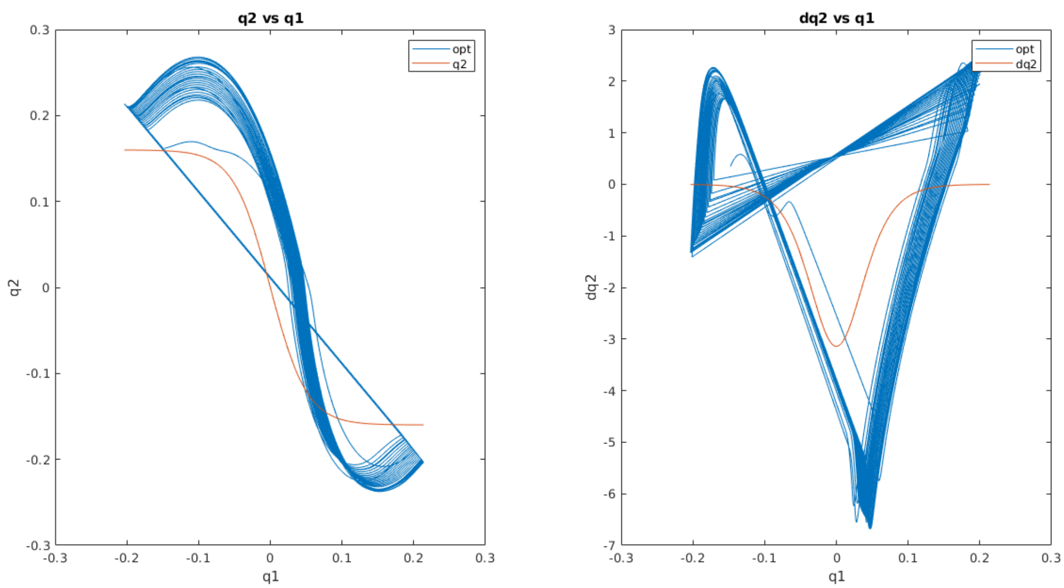

Figure 3: **Left plot** shows a dependency between angles of swing leg (q_2) and stance (q_1) legs. A desired trajectory is specified with red color. The blue lines represent actual dependencies between the angles in 20 steps. **Right plot** gives a dependency between velocities of swing leg (q_2) and stance (q_1) legs. Just like for the angles, red color specifies required trajectory whereas blue lines present actual trajectories after in 20 steps.

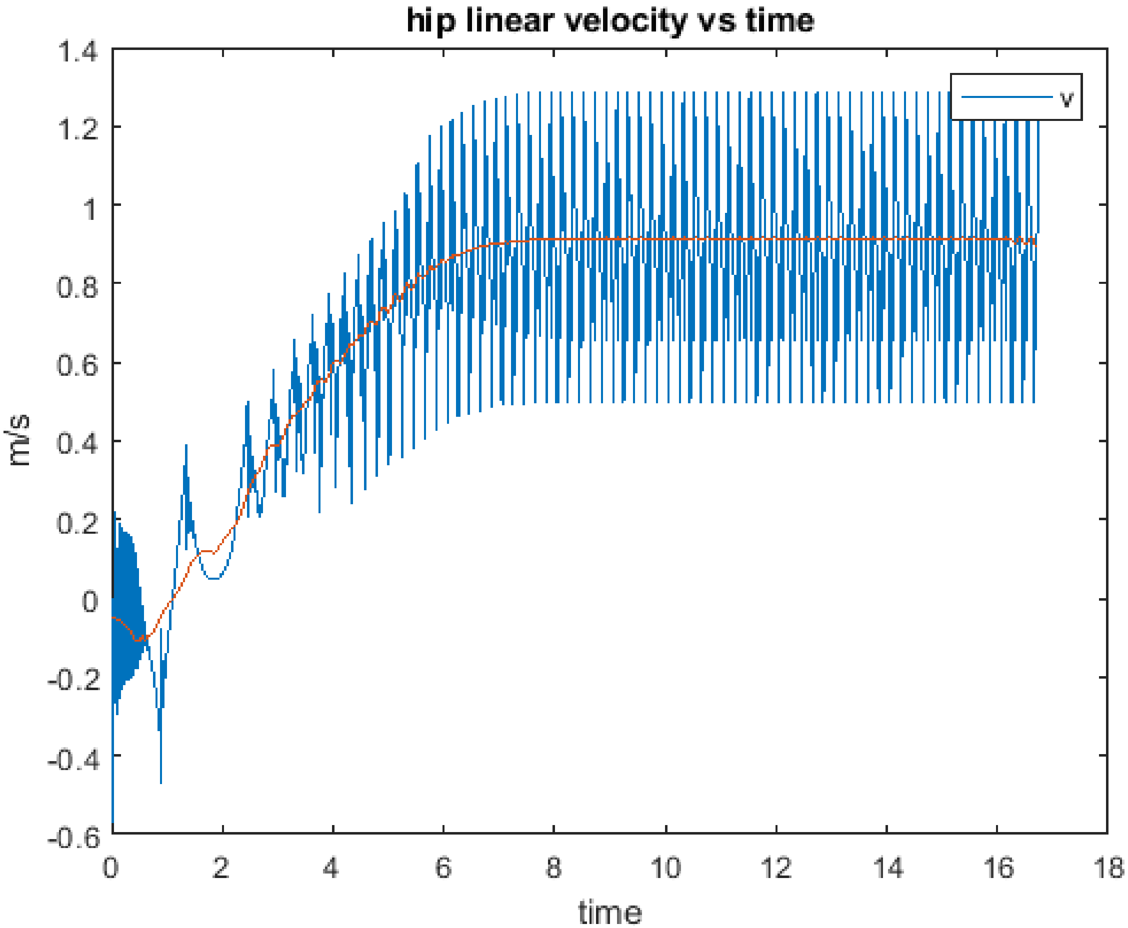

Figure 4: Velocity of the robot in the time

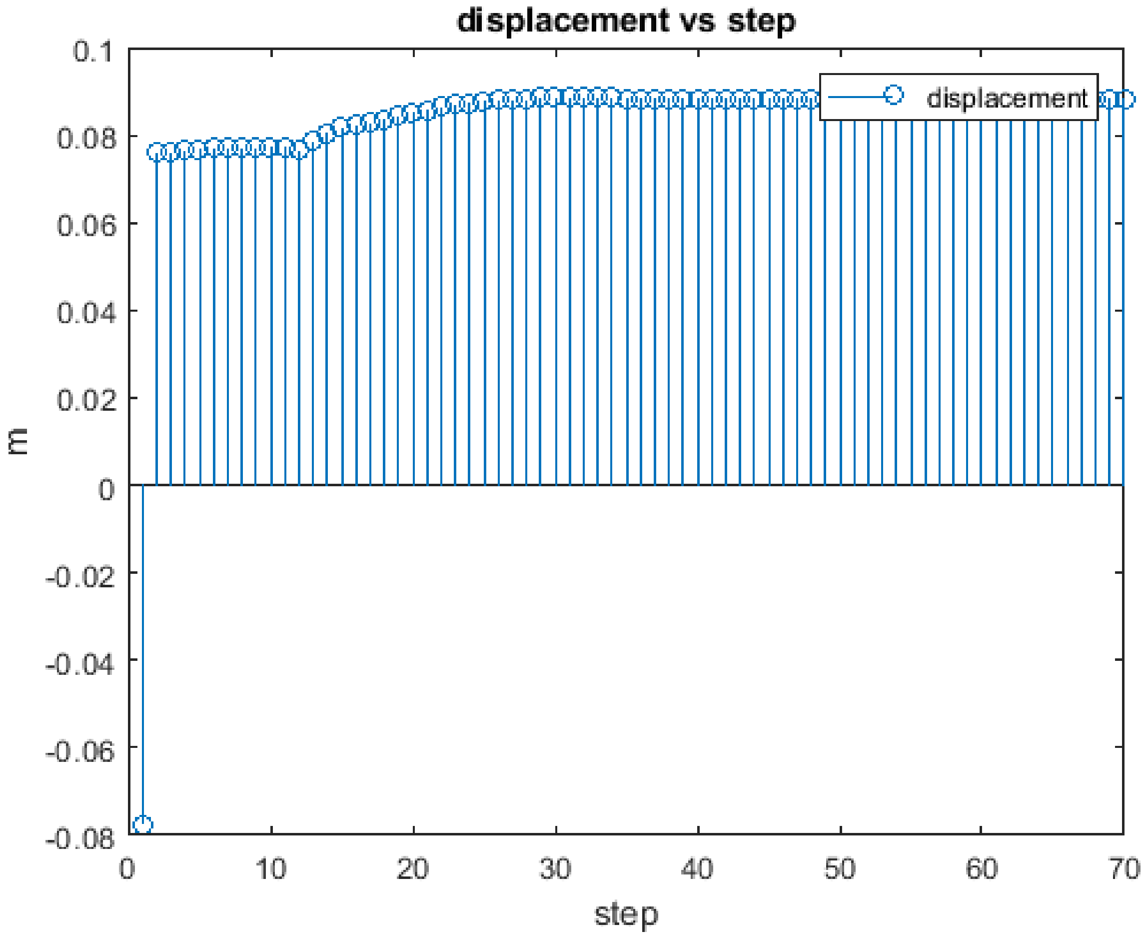

Figure 5: Displacement of each step. Step numbers are provided on abscissa and displacement (in meters) on ordinate

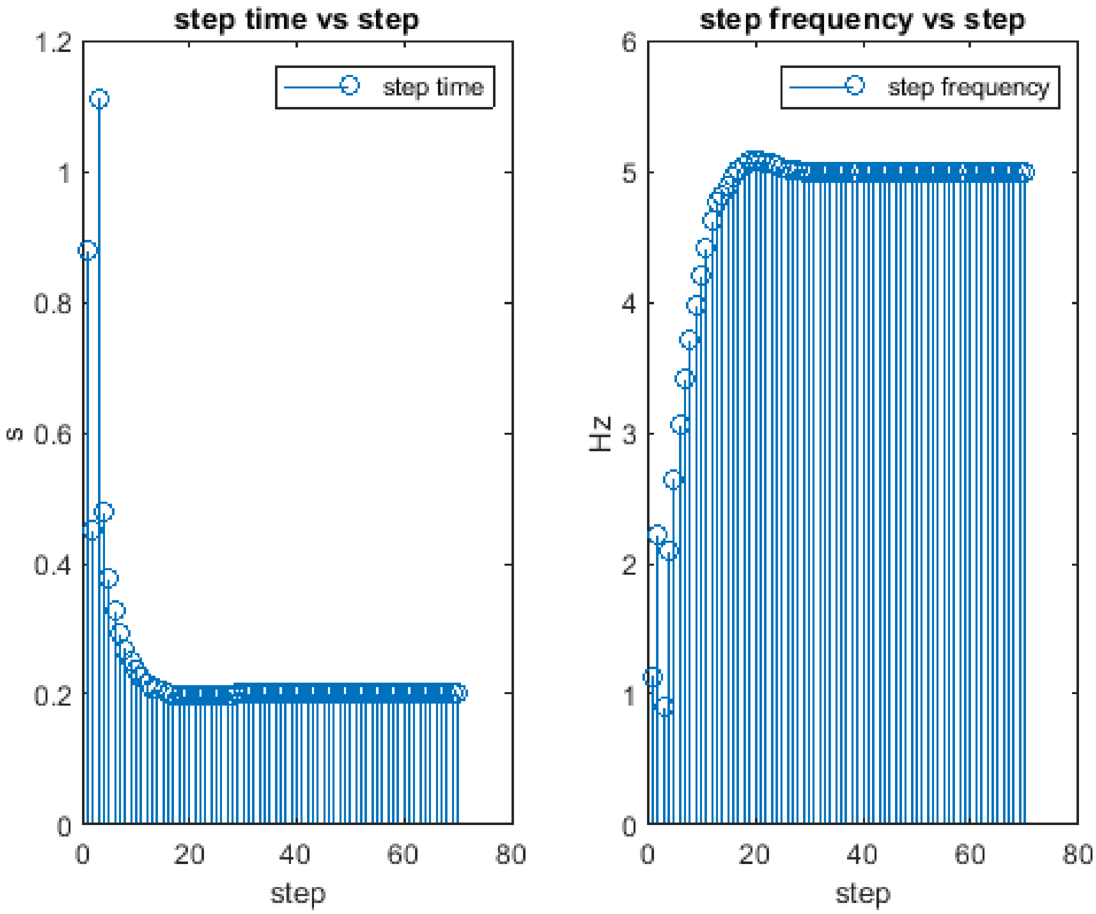

Figure 6: Time between two steps is shown in the left plot whereas in the right plot a frequency sampled in each step is plotted

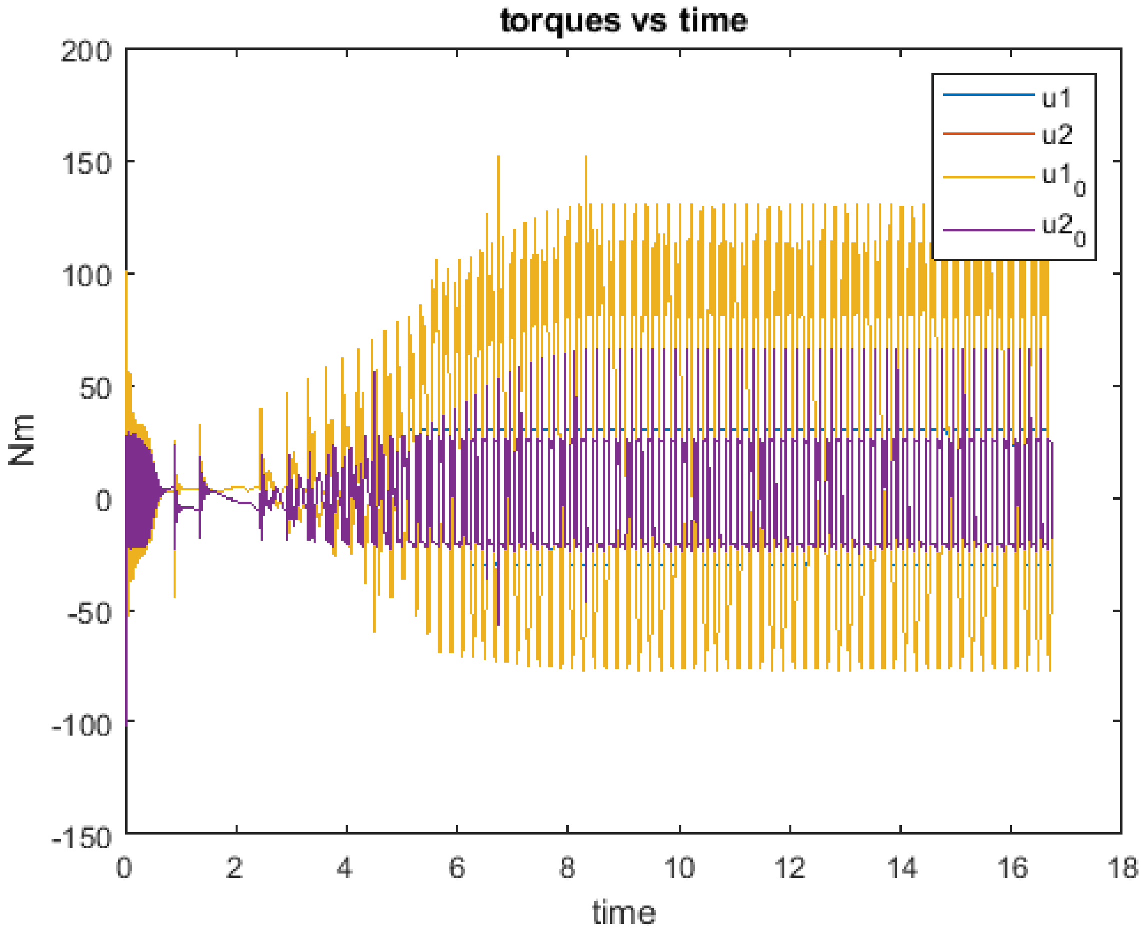

Figure 7: Forces applied to each controller (q_2 and q_1) in time

As in real robotic systems high torques are very hard to achieve, torques are limited to 30 Nm.

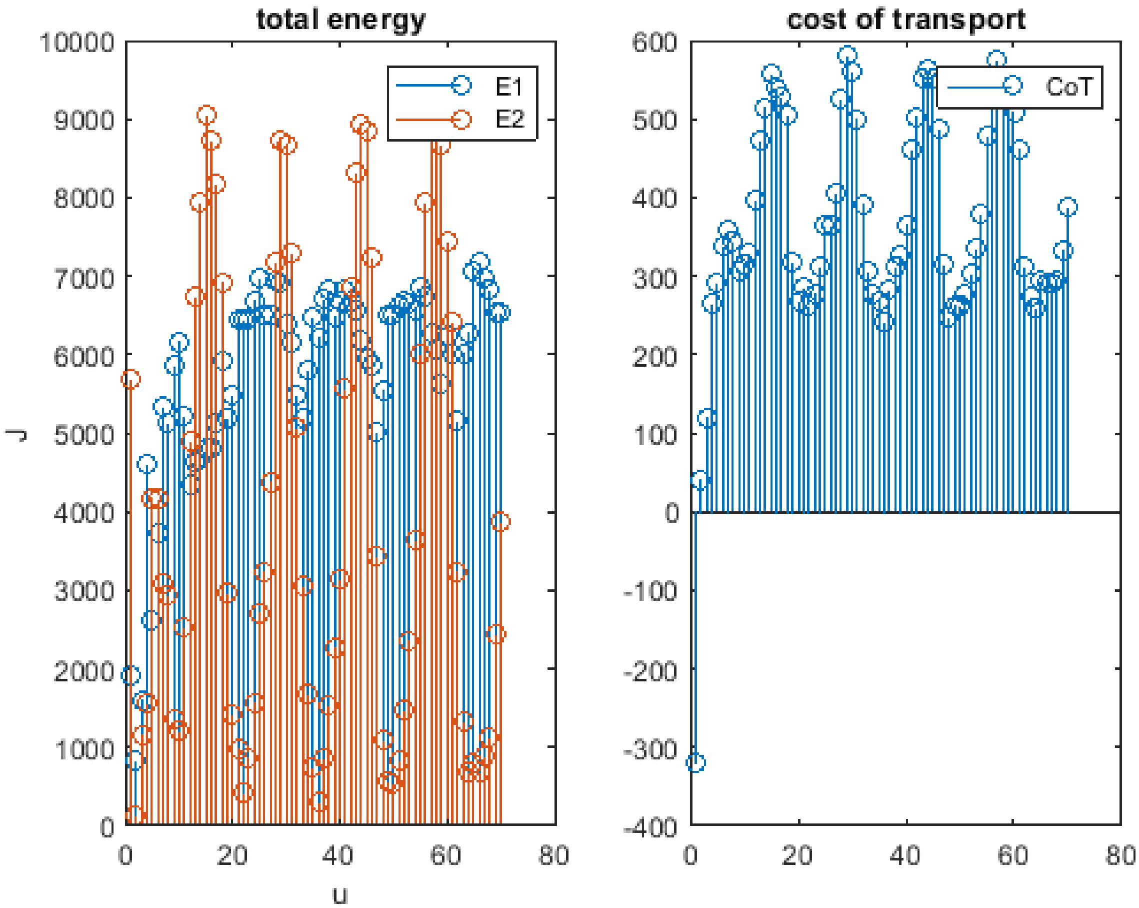

Figure 8: Cost of transport (CoT)

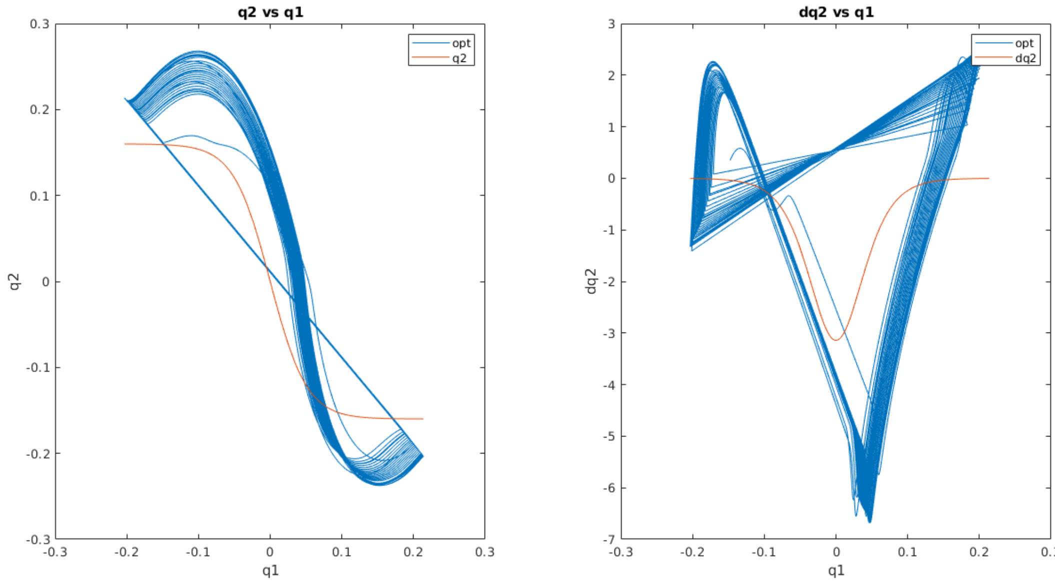

Figure 9: This plot shows a dependency between angle and velocity of the corresponding angle through time (q_i vs dq_i)

5 Optimization

The choice of the optimal hyper-parameters was performed with the help of the Global Optimization toolbox. Specifically, a genetic algorithm was used to find the optimal combination of values. The following variables were optimized:

- k_{pt}: the P parameter for the torso
- k_{dt}: the D parameter for the torso
- k_{ps}: the P parameter for the swing leg
- k_{ds}: the D parameter for the swing leg
- t_{t}: the target angle of the torso
- t_{sw}: the target angle of the swing foot (the maximum step angle)

6 Conclusion

During this work we learned how to design trajectories, apply virtual constraints and develop a controller. In order to fulfill required constraints different trajectories were designed and evaluated for the swing leg, as well for the torso. Also, in order to speed up the tweaking of hyper-parameters, reinforcement learning was applied with genetic algorithm.

Unfortunately, we couldn't develop a controller which can handles perturbations and produces torques without spikes above 30 Nm. The controller is able handle perturbations, but without 30 Nm limit.