# Project 1

## Locomotion planning based on Divergent Component of Motion (DCM)

MICRO-507

Legged robots

---


**Group 09**
Biselx, Michael (283812)
Bumann, Samuel (284704)
Vuilleumier, Loïk (283675)


**13.11.2021**

---


**École Polytechnique Fédérale de Lausanne**

# Table of content

# 1 Introduction

The goal of this project is to control a simulation of the Atlas biped robot produced by Boston Dynamics[1]. The simulation is done by using PyBullet[2]. The control is done by using a trajectory planner that consists of a Center of Mass (CoM) trajectory and a feet trajectory planner.

The resulting positions from the trajectory planner are then converted to the desired joint angles by using an inverse kinematic model. Finally a PD controller with feedback loop is used to drive the different motors of the Atlas robot.

For this project only a Divergent Component of Motion (DCM) trajectory planner needs to be implemented for a Linear Inverted Pendulum (LIP) model. The foot trajectories, inverse kinematic model and the control approach have already been implemented. The DCM trajectory is then used to plan the CoM trajectory.

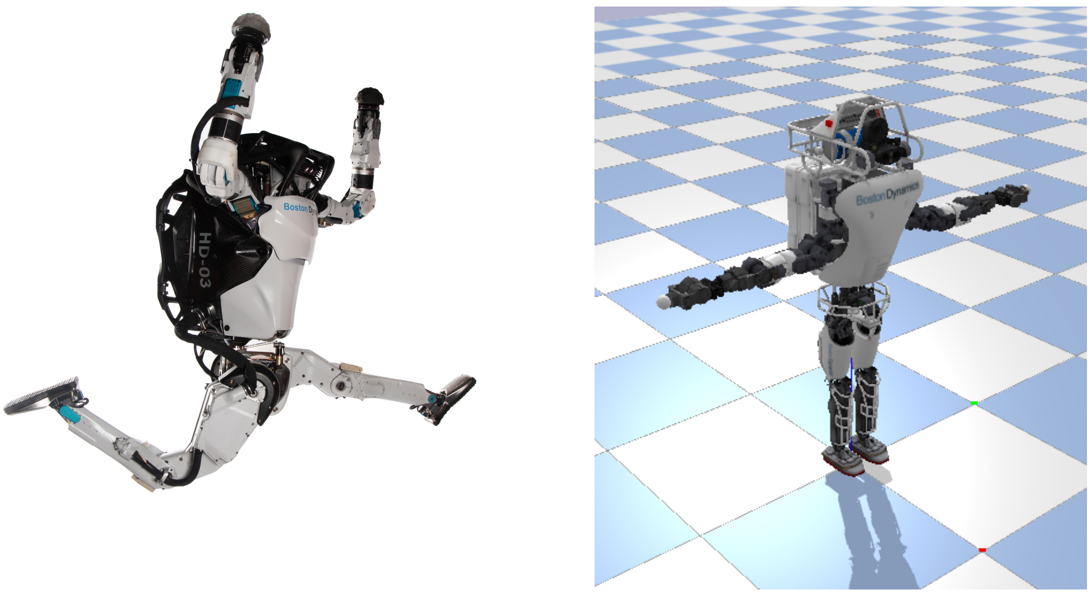

*Figure 1: Atlas robot jumping (left)[1] and simulation of Atlas robot in PyBullet (right)[2]*

[1] Boston Dynamics, Atlas, https://www.bostondynamics.com/atlas (09.11.2021)
[2] PyBullet, TINY DIFFERENTIABLE SIMULATOR, https://pybullet.org/wordpress/ (13.11.2021)

# 2 Method

## 2.1 DCM Model

The Linear Inverted Pendulum (LIP) model gives the following equation:

$$\ddot{x}_c = \omega^2(x_c - r_{\mathrm{CoP}}) \tag{1}$$

From this, the Divergent Component of Motion (DCM) can be defined as follows:

$$\xi = x_c + \frac{\dot{x}_c}{\omega} \tag{2}$$

Where ξ is the DCM, $\omega^2$= g/$z_c$ is the square of the natural frequency of the DCM dynamics and ($x_c$, $y_c$, $z_c$) is the CoM position. The DCM corresponds to the instantaneous capture point, as it is the component of the current movement which is moving away from the current center of pressure (CoP).

From equations (1) and (2) we can derive the dynamics of the CoM and the DCM:

$$\dot{x}_c = \omega(\xi - x_c) \tag{3}$$

$$\dot{\xi} = \omega(\xi - r_{\mathrm{CoP}}) \tag{4}$$

The solution of the differential equation (4) is the following:

$$\xi(t) = r_{\mathrm{CoP}} + (\xi_0 - r_{\mathrm{CoP}})e^{\omega t} \tag{5}$$

## 2.2 DCM Trajectory Planner

Once the CoP positions are defined, the DCM trajectory is calculated in three steps.

Firstly, the DCM for the end of each step is calculated (green points in figure 2). These are calculated recursively from the last position which we set as the last CoP position based on equation (5).

Secondly, the DCM trajectory is calculated between these points by means of equation (5) (green lines in figure 2).

And finally, to avoid instantaneous transitions in the trajectory, the DCM trajectory during the double support phase is obtained by cubic interpolation. This results in the final DCM trajectory (in blue in figure 2).

The CoM trajectory is then obtained by numerically integrating equation (3).

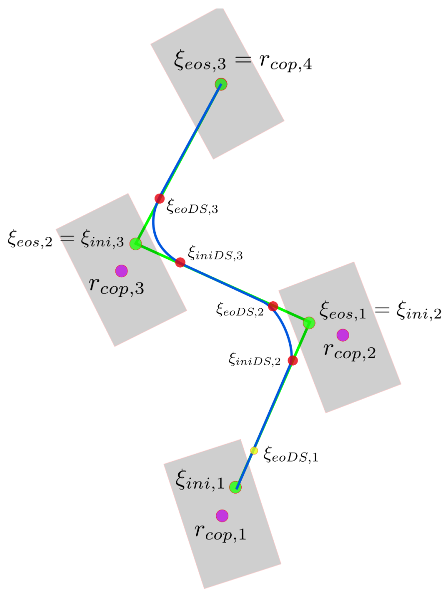


*Figure 2: Diagram of the footsteps, the desired DCM trajectory (in green) and the smoothed DCM trajectory (in blue).*[3]

[3] Legged robots (MICRO-507), Project 1 Description, Locomotion planning based on Divergent Component of Motion (DCM)

## 2.3 Analysis of the Produced Locomotion

A common measure of the efficiency of a gait across different robots is the cost of transport (CoT), which is defined as follows:

$$\mathrm{CoT} = \frac{E}{mgx} = \frac{P}{mgv}$$

During the simulation, in some timesteps, the torque and the angular velocity have opposing signs. This means that the power is negative which means that the actuator has to brake. There are different braking methods for electrical motors[4] :

1. Regenerative braking: The power is fed back to the battery and thus the energy is recovered. If assuming perfect energy recovery we can just deduce the power $P = \tau \cdot \omega$, where $\tau$ is the torque and $\omega$ is the angular velocity of the joint.
2. Dynamic/dissipative braking: The energy is dissipated as heat by connecting a resistor to the actuator. In this case we take $P = min(0, \tau \cdot \omega)$.
3. Plugging: In this case power is injected into the actuator for braking. If we assume that this power is the same as for acceleration we take $P = abs(0, \tau \cdot \omega)$.

We do not know what type of braking the motors on the simulated robot implement, a CoT was calculated for each of the braking scenarios. The CoT in our case was calculated in the following manner:

$$\mathrm{CoT} = \sum_{j}^{joints} \int_0^t \frac{P_j}{mgx} dt = \sum_{j}^{joints} \sum_{i}^{time} \frac{P_{i,j}}{mgx} \cdot \Delta t_{i,j}$$

Where $P_{i,j}$ is the power of joint j at timestep i. $\Delta t_{i,j}$ is the duration of timestep i.

Boston Dynamics' Atlas Robot implements hydraulic actuators, where we can assume that regenerative braking has not been implemented, leaving two possible models (2 and 3 above).

# 3 Results

## 3.1 Normal and Fast Locomotion

The results of the simulation for a normal locomotion can be seen in this video and for a fast locomotion in this video. One can see that the theoretical CoM and foot trajectories correspond with the simulated ones for a normal locomotion. This can also be seen in the almost identical mean theoretical speed (according to the model) and simulated speed (according to the simulation) of 0.0777m/s respectively 0.0778m/s. However the theoretical CoM and foot trajectories differ from the simulated ones for the fast locomotion.

[4] https://www.ambitechbrakes.com/the-different-types-of-braking-in-a-dc-motor/

This difference can also be seen in the difference between theoretical (0.1814m/s) and simulated walking speed (0.1576m/s), which can be attributed to the fact that the theoretical calculation makes some assumptions (such as perfect ground contact, or unconstrained actuators) which are no longer valid in the simulation. Thus one can deduce that the normal locomotion is the better solution.

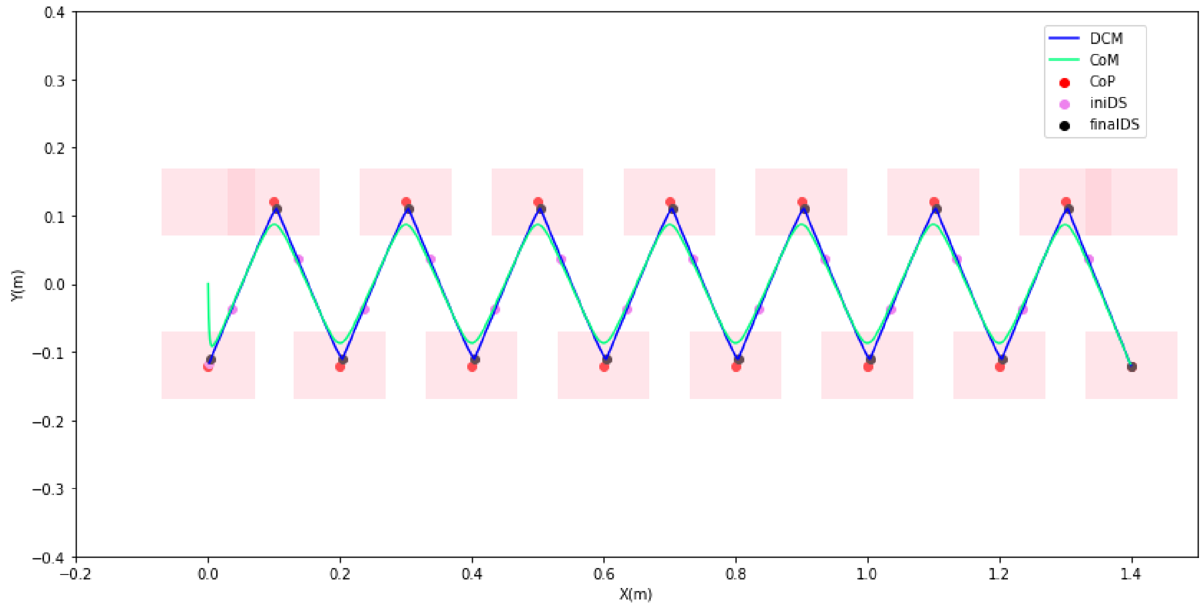


*Figure 3: DCM (blue), CoM (green), CoP (red dot) and feet polygons (light red areas) of robot in x-y plane for normal locomotion*

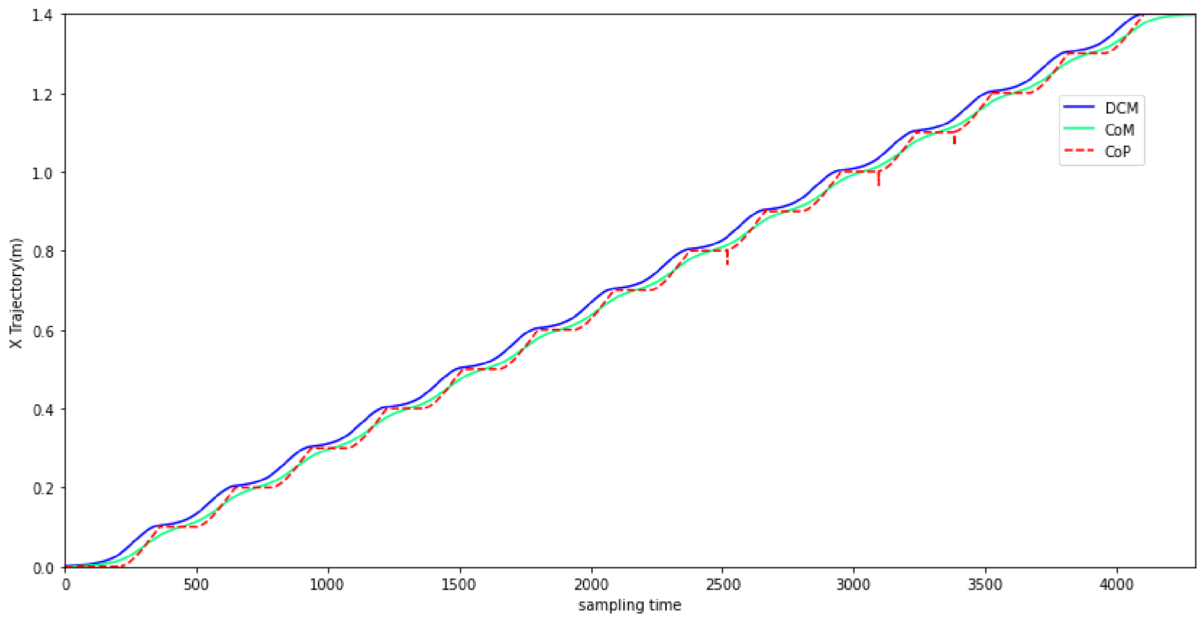


*Figure 4: plot of trajectory in x vs sampling time of DCM (blue), CoM (green) and CoP (red) of robot in for normal locomotion*

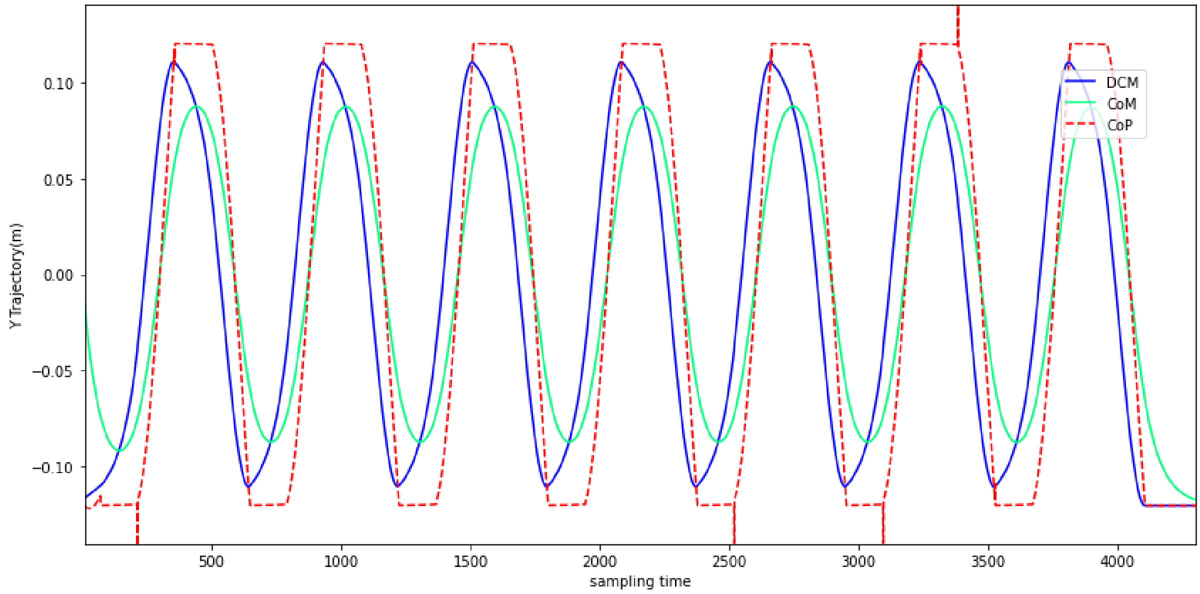


*Figure 5: plot of trajectory in y vs sampling time of DCM (blue), CoM (green) and CoP (red) of robot in for normal locomotion*

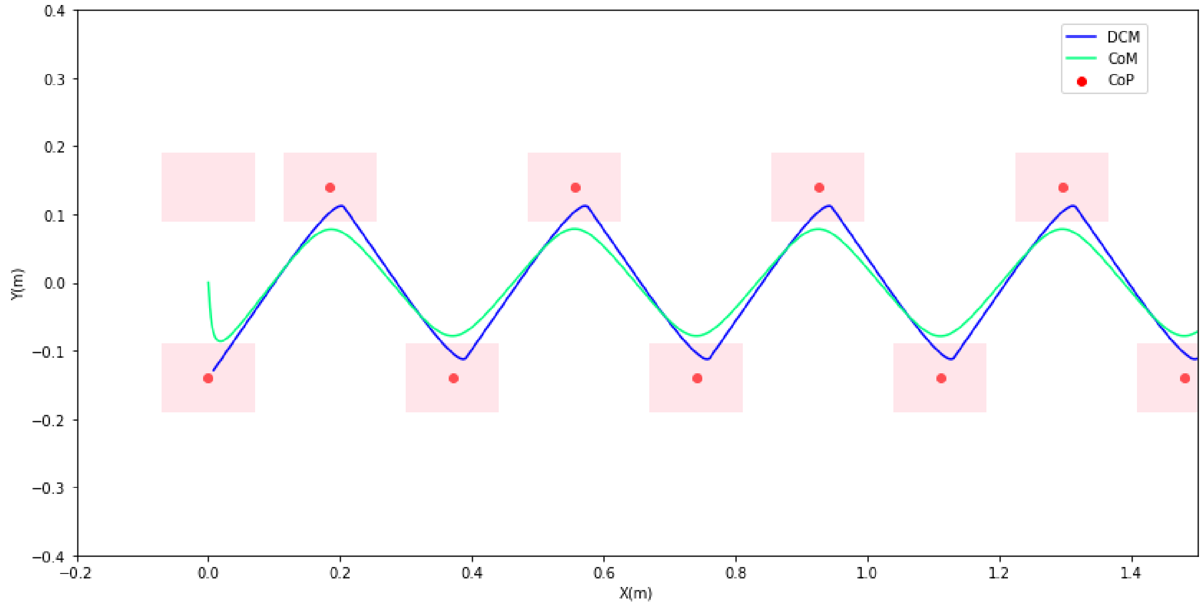


*Figure 6: DCM (blue), CoM (green), CoP (red dot) and feet polygons (light red areas) of robot in x-y plane for faster locomotion*

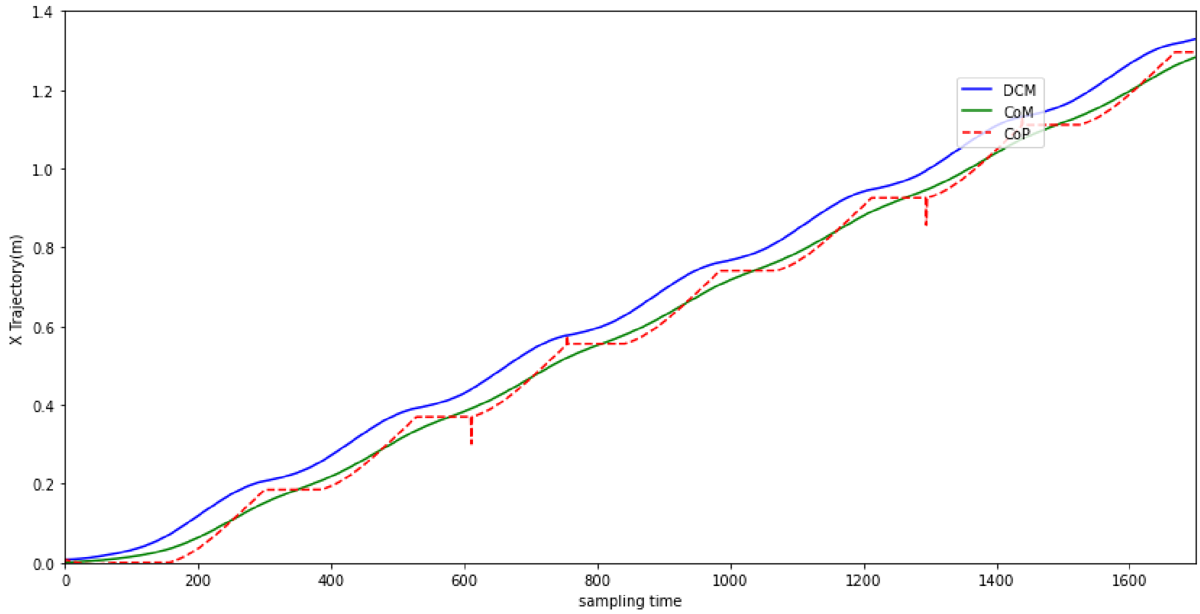


*Figure 7: plot of trajectory in x vs sampling time of DCM (blue), CoM (green) and CoP (red) of robot in for faster locomotion*

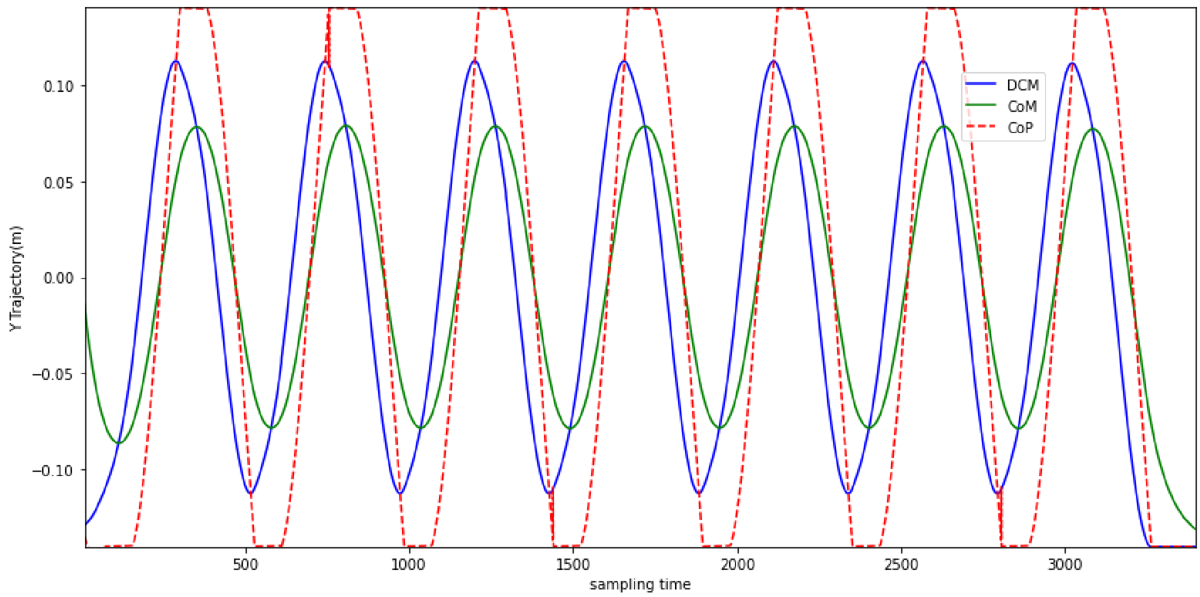


*Figure 8: plot of trajectory in y vs sampling time of DCM (blue), CoM (green) and CoP (red) of robot in for faster locomotion*

## 3.2 Cost of Transport

A comparison between the CoT for the Atlas and the Biped robot from Assignments 2 & 3 can be made. Table 1 shows that the CoT for the Atlas robot is smaller than the CoT for the Biped robot only in the regenerative braking case (which is not realistic due to hydraulic actuation, as discussed in chapter 2.3). For the dynamic/dissipative braking and plugging case the CoT for the Biped is much lower.

| | Atlas (Pybullet simulation) | | Biped (Matlab simulation) |
|---|---|---|---|
| | normal gait | fast gait | optimal gait |
| $\mathbf{CoT_{regen}}$ | 0.024 | 0.072 | 0.045 |
| $\mathbf{CoT_{diss}}$ | 0.261 | 0.440 | 0.066 |
| $\mathbf{CoT_{plug}}$ | 0.497 | 0.809 | 0.088 |

*Table 1: CoT calculated for the different simulations*

# 4 Questions

In this chapter the following questions given in the Jupyter Notebook are answered.

1. **Based on equation (5) which physical parameters will affect the rate of divergence of DCM dynamics?**

The rate of divergence of DCM is given by equation (4):

$$\dot{\xi} = \omega(\xi - r_{\mathrm{CoP}})$$

Thus the parameter $\omega$ and the difference of ξ and $r_{\mathrm{CoP}}$ affect the rate of divergence of DCM dynamics.

2. **In the DCM motion planning how do we guarantee to have a Dynamic Balancing?**

The DCM needs to be in the support polygon over time; meaning it can leave the support polygon but needs to return/converge always back into the support polygon for the system to be dynamically stable.

3. **If we guarantee dynamic balancing, why is the robot not able to walk without parameter tuning?**

While the theoretical model may be stable, it is very much simplified compared to the robot in the simulation, as the model does not consider many of the simulation's limitations such as actuator range and speed limits, range of pivots, etc. Additionally, everything above the hip is simplified to a point-mass in the DCM planning. These many simplifications lead to disturbances which break the stability of the simulated robot.

**4. Which parameters did you find useful for tuning to have a stable locomotion and what's the value of those parameters?**

The following parameters (with their value) were changed to have a stable locomotion:

```
doubleSupportDuration = 0.6
pelvisHeight = 0.8
```

These are the parameters used for the normal gait of table 1, and they result in a theoretical walking speed of 0.0777m/s and a simulated walking speed of 0.0778m/s.

**5. Find the cost of transport for this walking planner and do a comparison with the previous Matlab homework? What is the source of difference?**

For this walking planner, the costs of transport listed in table 1 were found. The differences between the two simulations can be attributed to several factors (such as complexity of robot, the Atlas robot having many more joints than the Biped Walker, etc.. ), but the biggest reason for difference is that in the Biped Walker, the gait was explicitly designed in order to minimize the cost of transport through optimization. Having knees usually reduces the CoT because it enables smaller leg motions, however in this case, it seems that the more joints require more energy to control, which negates the positive effect of knees.

Note that the regenerative braking CoT is not meaningful to compare the two robots because the Atlas robot is hydraulically actuated and thus most likely has no regenerative braking capabilities.

**6. Change the step position and duration in the "Planning Feet Trajectories" section, to have a faster locomotion. What's the fastest walking speed that you have achieved and what are the corresponding parameters?**

The fastest theoretical walking speed of 0.1814m/s and simulated walking speed of 0.1576m/s was achieved with the following parameters (found experimentally):

```
doubleSupportDuration = 0.6
stepDuration = 0.95
pelvisHeight = 0.87
maximumFootHeight = 0.18
stepWidth    = 0.14
stepLength   = 0.185
```

The difference between theoretical and simulated walking speeds are due to actuator torque limits (and other simulation constraints). As one can see in table 1 the CoT for the fast locomotion is almost double compared to the CoT of the normal locomotion speed. Thus, this type of fast walking is not energetically efficient.

# 5 Conclusion

This project gave us the possibility to implement a DCM and CoM trajectory to control a robot in a simulation by using PyBullet. It was interesting to see that the robot in the simulation can fall even if one guarantees dynamic balancing in the DCM motion planning and that heuristic parameter tuning was necessary to make it work.
It was also useful to get an introduction to PyBullet and to see PyBullet's usefulness for mobile robotics.
Overall we realized that control of a biped robot is not a trivial task.