Locomotion Planning Based on Divergent Component of Motion (DCM)

Legged Robotics, Group 22

Vincent Bruno Jean-Louis Gherold, Théo Yvan Pugin-Bron, Benjamin Noah Lugon-Moulin

October 27, 2022

1 Introduction

This project has the aim of planning the Center of Mass (CoM) trajectory of a biped robot by applying the concept of Divergent Component of Motion, for locomotion on flat terrain. By simulating the locomotion of the Atlas robot, we analyze the effect of different parameters on the performance of our DCM and CoM-based foot step planning.

2 DCM and CoM Motion Planning

In order to plan the foot steps of the robot, we must first characterize the leg dynamics of the robot, which can be well described by the Linear Inverted Pendulum model. By setting the momentum to be zero around the Center of Pressure (CoP), we obtain the following equation :

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

where x_c is the horizontal position of the Center of Mass (CoM) of the robot (which corresponds to the robot's pelvis), $\omega = \sqrt{\frac{g}{z_c}}$ is the natural frequency of the leg dynamics, and r_{CoP} corresponds to the Center of Pressure. Next, we define the Divergent Component of Motion (DCM) as follows :

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

Monitoring the DCM will enable us to plan the foot steps while preventing the robot from falling.

From these two equations we derive the CoM and DCM dynamics:

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

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

Solving the DCM dynamics equation yields the following :

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

with $\xi_0 = \xi(0)$. The timestep t takes values $t \in [0, T]$ for each step, with T being the duration of the step. This equation gives us the desired DCM trajectory from given constant CoPs, and therefore we will base ourselves on this equation to plan the DCM and CoM motion of the robot.

Next, the DCM and CoM motion planning is implemented in the following steps:

1. Based on the kinematic and dynamic constraints of the robot, as well as the its desired velocity, we select the footstep positions and step duration.

2. Next, in order to guarantee dynamic balance during locomotion, we place the desired CoP inside the footprints.

3. Capturability Constraint : The last DCM is placed on the last CoP ($\xi_{eos,N-1} = r_{CoP,N}$, eos standing for end-of-step). As a result, we order the robot to come to a stop after reaching the final foot position.

4. For each step, we calculate the desired DCM locations at the end of the step via recursion using equation (5):

$$\xi_{eos,i} = r_{CoP,i} + (\xi_{ini,i} - r_{CoP,i})e^{\omega T} \tag{6}$$

$$\xi_{ini,i} = r_{CoP,i} + (\xi_{eos,i} - r_{CoP,i})e^{-\omega T} \tag{7}$$

$$\xi_{eos,i-1} = \xi_{ini,i} \tag{8}$$

with $\xi_{ini,i}$ being the ith initial desired DCM.

5. From (5) and (7) we compute the reference trajectories for the DCM position for each step:

$$\xi_i(t) = r_{CoP,i} + (\xi_{ini,i} - r_{CoP,i})e^{\omega(t-T)} \tag{9}$$

with the internal step t being reset at the beginning of each step, $t \in [0, T]$.

Finally, from the full DCM trajectory, we substitute the current DCM and CoM into equation (3) to find the CoM trajectory, and then use numerical integration (such as the Euler method) to find the CoM position.

3 Simulation Results

For our motion planning implementation, we have the following desired footstep trajectories:

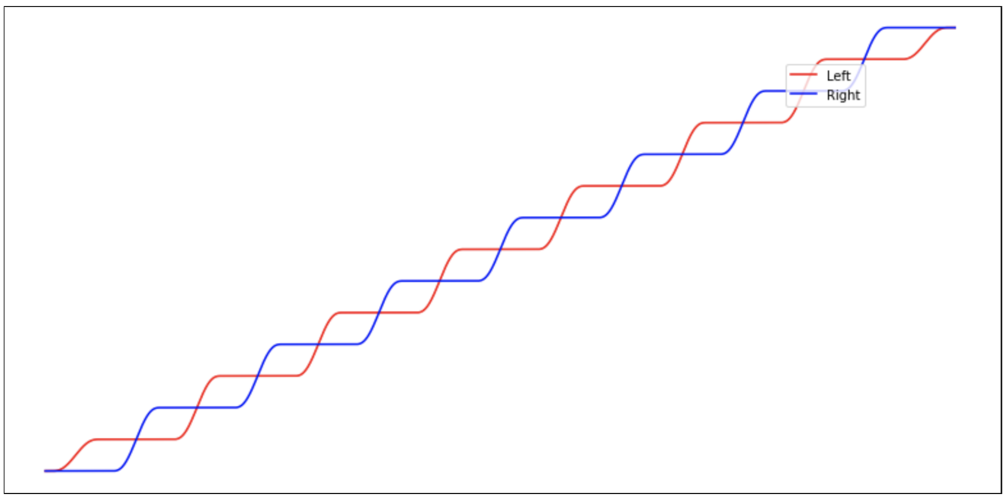

Figure 1: Horizontal position of each foot as a function of time.

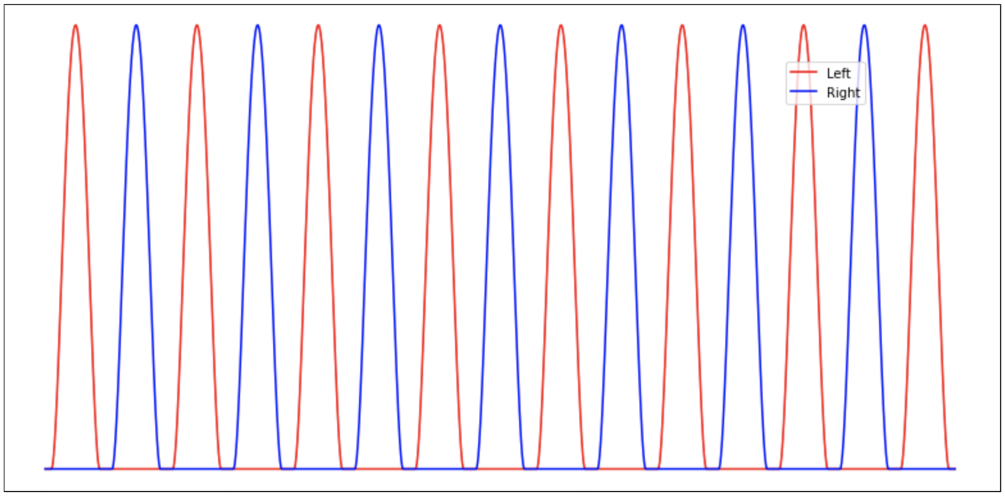

Figure 2: Vertical position of each foot as a function of time. We observe that as this is a walking gait, there is always at least one foot touching the ground. Additionally there is a short phase during which both feet are in contact with the ground : this is the double support phase, whose duration has been set to 20% of the total step duration. Note however that for the actual implementation of the equations for the DCM motion planning, we assume there is no double support phase.

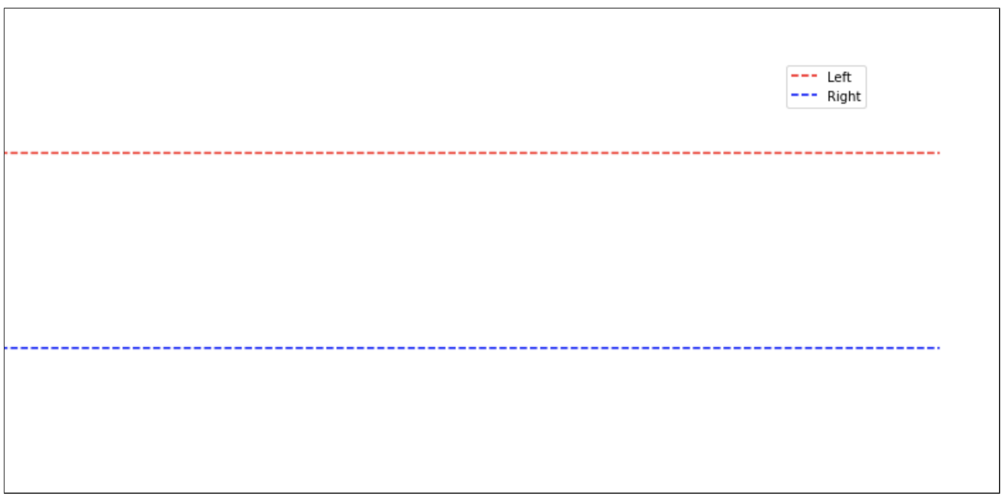

Figure 3: Horizontal y-position of each foot as a function of time. This shows us that the robot is moving forward in a straight line.

Given these feet trajectories, we obtain the following DCM and CoM trajectories:

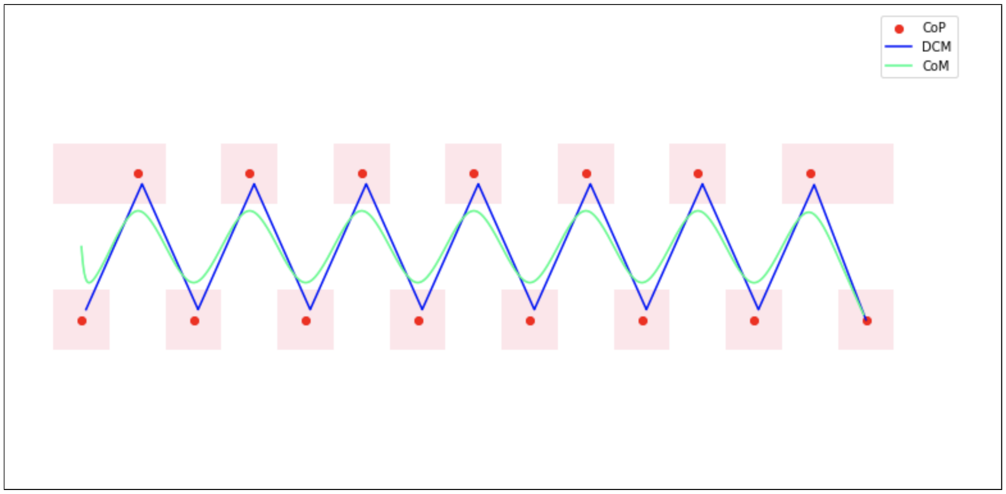

Figure 4: Top view of the DCM, CoM, CoP and foot positions. We can observe that the CoM doesn't have to be within the footprint at the end of each step to guarantee stability. As will be explained later, stability is guaranteed by placing the CoP for each footstep within each footprint. We may also notice how for the last step the DCM coincides with the CoP, which leads to the robot making a full stop.

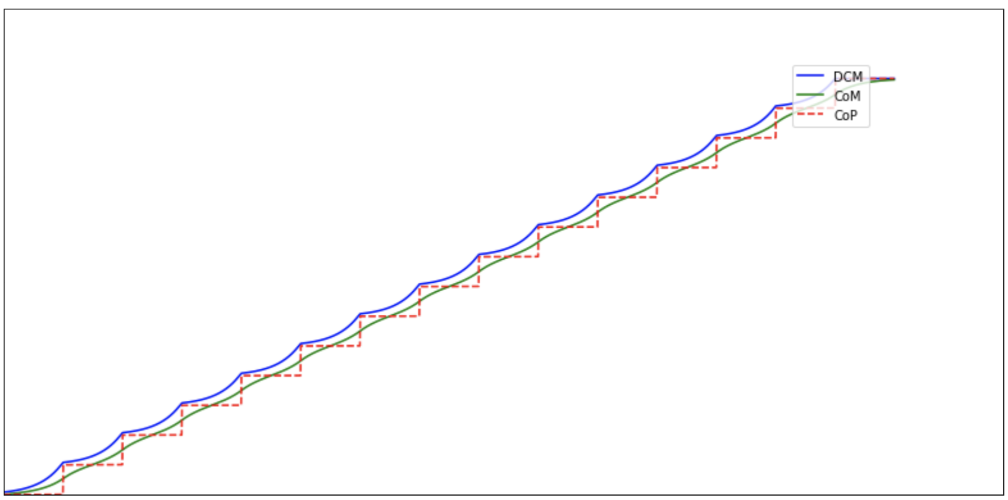

Figure 5: Evolution of DCM, CoM and CoP positions with time.

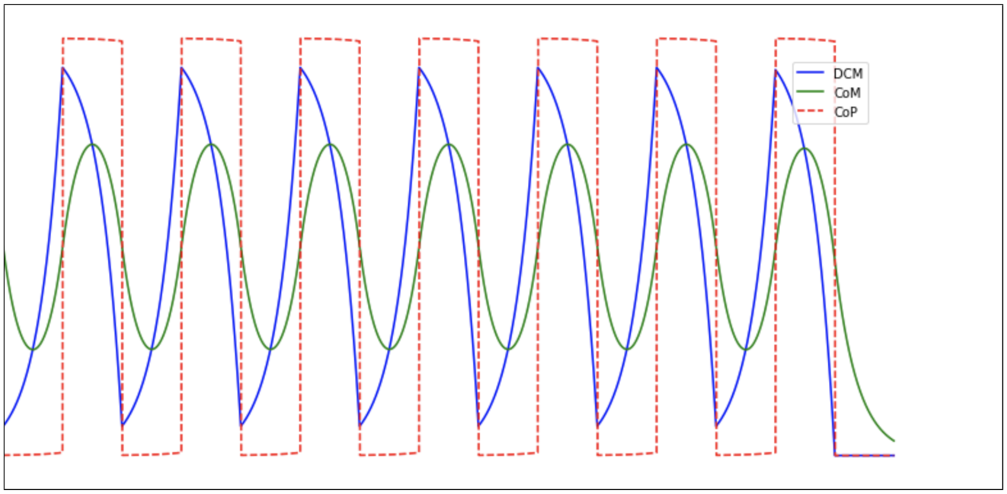

Figure 6: DCM, CoM and CoP trajectories. Here we can more clearly observe how the DCM evolves as an exponential function : $\xi_i(t) = r_{CoP,i} + (\xi_{ini,i} - r_{CoP,i})e^{\omega(t-T)}$

4 Question Answers

4.1 Physical parameters affecting rate of divergence of the DCM dynamics

Recall the solution for the DCM Dynamics (equation (5)) : $\xi(t) = r_{CoP} + (\xi_0 - r_{CoP})e^{\omega t}$

From this equation we can observe that the natural frequency $\omega = \sqrt{\frac{g}{z_c}}$ depends on both the gravity g and height of the center of mass z_c. Decreasing g (for example by having moon gravity) and increasing z_c (having longer legs) leads to a smaller ω and therefore a smaller term $e^{\omega t}$, and as a result the rate of divergence is much smaller (this makes it easier to keep the system stable).

4.2 How to guarantee dynamic balancing conditions

In order to guarantee dynamic balancing conditions, we must make sure that for each footstep, the ZMP (Zero Moment Point, defined as "the point at which the horizontal torques due to ground reaction forces are zero"[1]) remains within each footprint over time. This is done by setting the CoP in a fixed desired location inside the footprint. Note that the ZMP is the same as the CoP when the ZMP is right under one foot, or in the support polygon.

4.3 Why is the robot unable to walk without parameter tuning

The robot is in open loop, everything is planned before, and only joint torques are applied to the robot without any feedback on how the robot has behaved. That means if in the simulation the robot is not exactly the same as in the mathematical model, we can have instabilities.

4.4 Parameter tuning

In order to achieve stable locomotion, we choose the following parameters:

- a double support duration of 0.14 s
- a pelvis height of 0.72 m
- a maximum foot height of 0.05 m
- a step width of 0.12 m
- a step length of 0.14 m
- a step duration of 0.7 s

- a constant CoM height of 0.72 m
- a timestep of 1/240 s
- a number of step of 14
- gravity constant is (of course) 9.81 $m \cdot s^{-2}$

We obtained the following result:

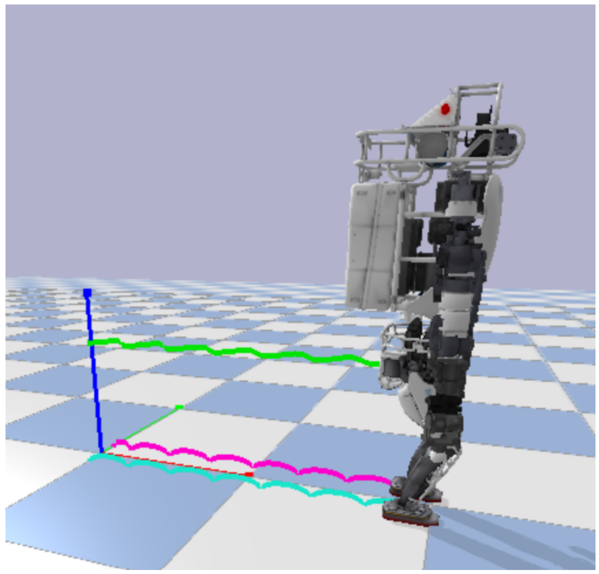

Figure 7: Stable locomotion with the parameters described above

Using Pybullet, we computed also the mean walking speed of the robot for the parameters. We obtain of value of 0.17381 $m \cdot s^{-1}$

4.5 Calculation and plotting of the real ZMP

To plot the ZMP, we used the function getContactPoints() from Pybullet which is described here:

getContactPoints will return a list of contact points. Each contact point has the following fields:

contactFlag	int	reserved
bodyUniqueIdA	int	body unique id of body A
bodyUniqueIdA	int	body unique id of body B
linkIndexA	int	link index of body A, -1 for base
linkIndexB	int	link index of body B, -1 for base
positionOnA	vec3, list of 3 floats	contact position on A, in Cartesian world coordinates
positionOnB	vec3, list of 3 floats	contact position on B, in Cartesian world coordinates
contactNormalOnB	vec3, list of 3 floats	contact normal on B, pointing towards A
contactDistance	float	contact distance, positive for separation, negative for penetration
normalForce	float	normal force applied during the last 'stepSimulation'

Figure 8: getContactPoints function description

We used the following formula to compute the ZMP:

$$ZMP_x = \frac{\sum_{j=1}^{N} p_{jx} f_{jz}}{\sum_{j=1}^{N} f_{jz}}$$

$$ZMP_y = \frac{\sum_{j=1}^{N} p_{jy} f_{jz}}{\sum_{j=1}^{N} f_{jz}}$$

The f_{jz} is the vertical component of contact force at contact point j, and p_{jx} is the contact position of contact point j.

By computing the ZMP for each time step, we obtain the following graph :

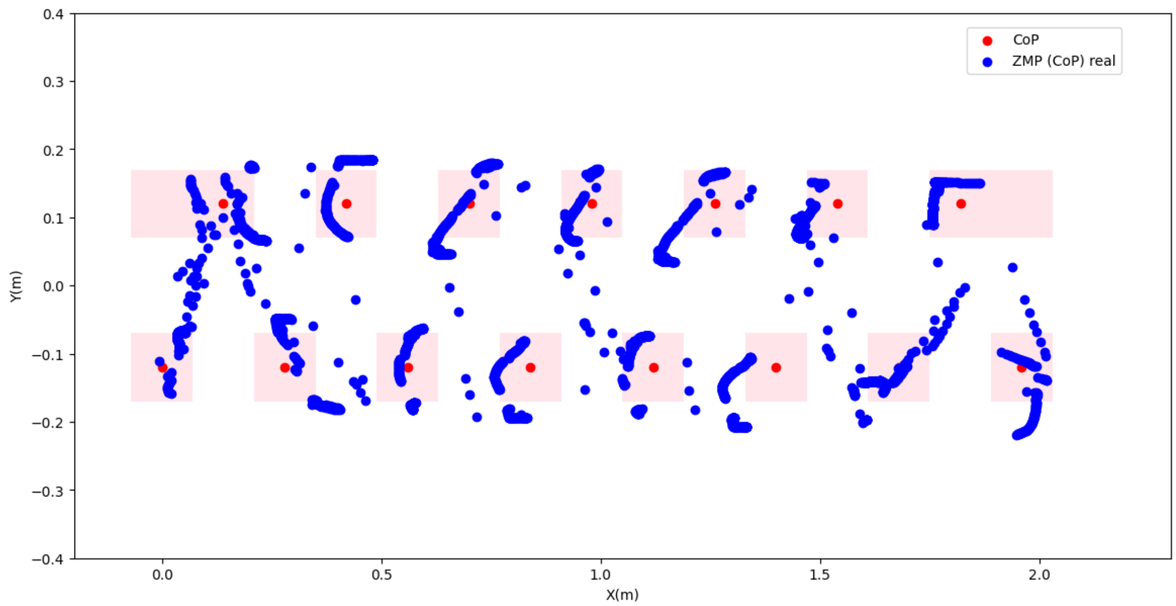

Figure 9: ZMP graph with the planned CoP (in red) and the real CoP (in blue)

We can see that the ZMP computed in blue roughly matches our target CoP in red.

4.6 Tuning DCM parameters to achieve faster locomotion

After several iteration, fortunately not done by hand, the faster robot (average speed of $0.559m \cdot s^{-1}$, see video *'Atlas on Earth'*), is obtained by tuning the following parameters :

- Step duration : 0.30475 s
- Step length : 0.212 m
- Maximum foot height : 0.082 m
- Step width : 0.128 m
- Pelvis height : 0.62667 m

To achieve this speed, we tried several combinations of parameters (automatically) and tested whether the robot fell or stood stable. Note that we also changed the pelvis height, despite the fact that this isn't a DCM parameter.

4.7 Considering moon gravity

Here we consider that we use a moon exploration Atlas, the gravity is hence 1.62 $m \cdot s^{-2}$. After several iterations, the faster robot, see video *'Lunar explorer'*, is obtained by tuning the following parameters :

- Step duration : 0.51 s
- Step length : 0.4 m
- Maximum foot height : 0.13 m
- Step width : 0.12 m
- Pelvis height : 0.72 m

The speed of the robot can be obtained using the getBaseVelocity() function of Pybullet. The average speed, from first to last step, of this Lunar explorer is around 0.628 $m \cdot s^{-1}$. However, as we can see in the second part of the video, the Atlas does not go straight ahead. Indeed during the first steps, which correspond to the acceleration time, the feet slip on the moon surface especially for the right foot, resulting in a slight deviation from the initial path.

4.8 Effect of considering double support during DCM planning

In the implementation of the equations for the DCM motion planning equations, we assumed there was no double support phase during each step. In other words, the transitions between left and right support phases are instantaneous, and both feet are never in contact with the ground at the same time. As a result, the external forces applied on each supporting foot are discontinuous, and this leads to the DCM trajectory having sharp edges for the times corresponding to the transition between left and right leg support (as can be seen in Figure 4). [2]

By considering the double support in our equations for DCM planning, the external forces acting on each foot would no longer be discontinuous and the DCM trajectory edges would be rounded. This smoother DCM trajectory can be observed in the implementation done by A Vedadi, K. Sinaei, P. Abdolahnezhad, S. Aboumasoudi, A. Yousefi-Koma in *Bipedal Locomotion Optimization by Exploitation of the Full Dynamics in DCM Trajectory Planning*[1], who obtained the DCM trajectory below :

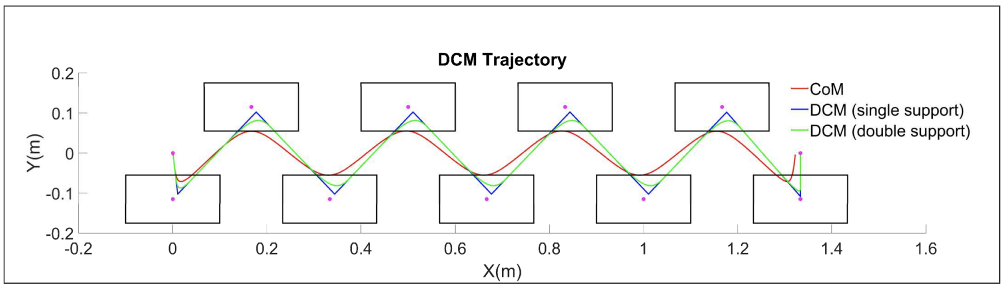

Figure 10: Planned trajectories of CoM (red), DCM with continuous double support (green) and without continuous double support (blue)[1].

5 Conclusion

In this project we have successfully modeled and implemented motion planning of the Center of Mass of the simulated biped Atlas robot, by using the concept of Divergent Component of Motion. By analyzing the equations for the CoM and DCM dynamics as well as tuning the different parameters of the robot, we were able to gain a better understanding on how a biped robot's trajectory can be modeled and adjusted.

Despite the CoP of the robot being set inside each footprint of each step to guarantee stability, we observed that parameter tuning is still necessary to keep the robot from falling, as the robot operates without any feedback. Initially achieving a speed of 0.17381 $m \cdot s^{-1}$, by tuning parameters such as step duration, step length and maximum foot height, we managed to reach a speed of $0.559 m \cdot s^{-1}$. When considering a different value for gravity such as that on the moon, parameters needed to be tuned again, and the robot achieved a top speed of 0.628 $m \cdot s^{-1}$. Finally, we commented on the effect of considering the double support phase in our DCM equations, which would smoothen the sharp edges of our DCM trajectory.

References

[1] A Vedadi, K. Sinaei, P. Abdolahnezhad, S. Aboumasoudi, A. Yousefi-Koma. "Bipedal Locomotion Optimization by Exploitation of the Full Dynamics in DCM Trajectory Planning". In: *Proceedings of the 9th RSI International Conference on Robotics and Mechatronics (ICRoM 2021)* (2021). DOI: https://arxiv.org/pdf/2208.00221.pdf.

[2] Englsberger, Johannes ; Ott, Christian ; Albu-Schaffer, Alin. "Three-Dimensional Bipedal Walking Control Based on Divergent Component of Motion". In: *New York: IEEE;IEEE transactions on robotics* VOL.31.NO.2 (2021), pp. 355–368.