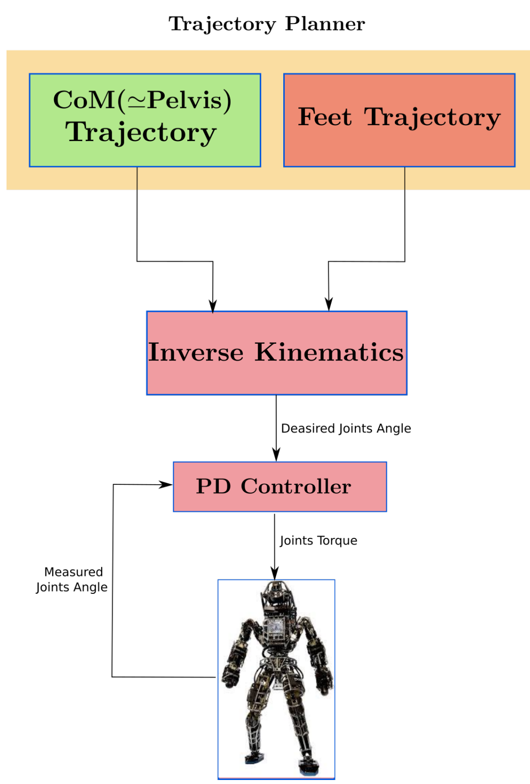
Locomotion planning based on Divergent Component of Motion (DCM)

Introduction

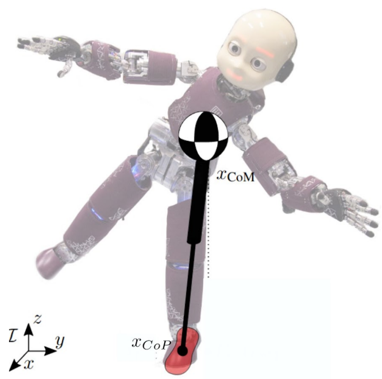
In this project you will plan the Center of Mass (CoM) trajectory for a biped by using the Divergent Component of Motion (DCM) concept for locomotion on flat terrain. The code structure will include three blocks: the DCM Planner, Foot Trajectory Planner, and Inverse Kinematics. The only block that you need to implement is the DCM Planner. In other words, you only need to open and edit the `DCMTrajectoryGenerator.py` class and follow the comments that have been written in this class. In the following figure, the different blocks of locomotion planning and control has been illustrated. The `DCMTrajectoryGenerator.py` is responsible for CoM motion generation that has been indicated by a green rectangle. Note that we consider a practical assumption that CoM is located on a fixed point on the pelvis.



DCM and CoM Motion Planning

In this section we will plan the DCM and CoM trajectory following the method presented in: Englsberger, Johannes, Christian Ott, and Alin Albu-Schäffer. "Three-dimensional bipedal walking control based on divergent component of motion.“ IEEE transactions on Robotics 31.2 (2015): 355-368.(254).

Here we elaborate the equations in detail. For the inverted pendulum equation we have:



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

This equation has been derived by finding the momentum around the Center of Pressure (COP, also ZMP) that is equal to zero with the dynamic balancing condition. Then we define the DCM dynamics as follows:

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

where ξ is the DCM, x_c is the CoM position, and $\omega = \sqrt{\frac{g}{z_c}}$ is the natural frequency of the DCM dynamics. By reordering (2), we can derive the CoM dynamics:

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

This shows that the CoM has stable first-order dynamics (i.e. it follows the DCM). By differentiating (2) and inserting (3) and (1), we can write the DCM dynamics:

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

The DCM has unstable first-order dynamics (it is “pushed” by the CoP), whereas the CoM follows the DCM with stable first-order dynamics.

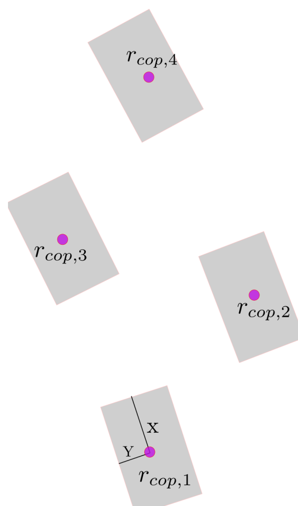
To find the desired DCM trajectory from given constant CoPs, the solution for the DCM Dynamics is:

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

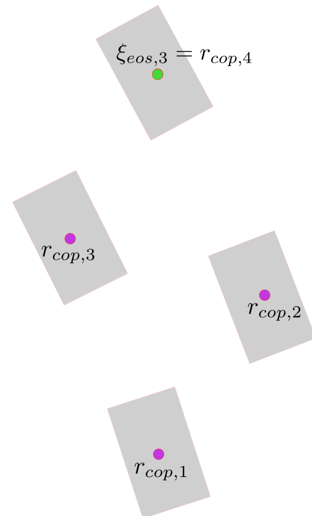
where $\xi_0 = \xi(0)$. The “internal” timestep t is reset at the beginning of each step, i.e., $t \in [0, T]$ (T is the duration of the step). In the following, we will present a step by step method for DCM and CoM motion planning based on (5).

1. First, we select the foot step position and step duration based on the desired velocity and considering the kinematic and dynamic constraint of the robot:

2. Place the desired CoP in a fixed location inside of the foot print. This condition guarantees dynamic balance during locomotion:



3. We place the last DCM position on the last CoP (Capturability constraint). For planning, we assume that the DCM will come to a stop over the final previewed foot position, i.e., $\xi_{eos,N-1} = r_{cop,N}$ (where *eos* is end-of-step):



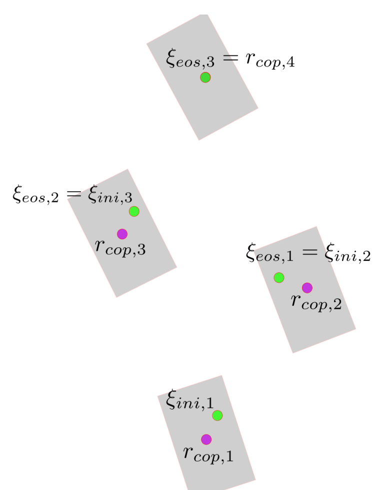
4. By having the constant desired CoP positions for each step and the last DCM Position (located on the CoP), we find the desired DCM locations at the end of each step via recursion using equation (5):

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

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

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

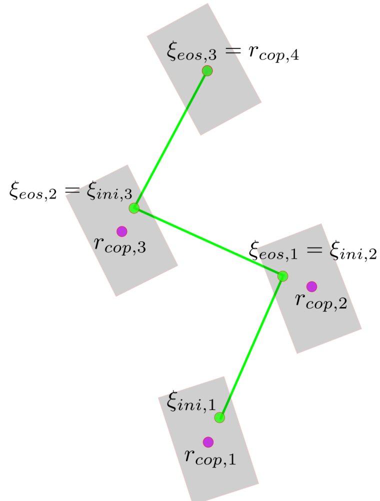
where $\xi_{ini,i}$ is the ith initial desired DCM.



5. Now based on (7) and (5), the reference trajectories for the DCM position (for single support (SS) phase) of the ith step can be computed as:

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

The "internal" step time t is reset at the beginning of each step, i.e., $t \in [0, T]$ (T is the duration of the step).



6. As planned so far, the DCM trajectory only considers single support phases (and instantaneous transitions between them), so there is a drawback of a discontinuous CoP reference and thus discontinuous desired external forces at the support transitions. This leads to discontinuities in the commanded joint torques, which can be infeasible for a physical robot due to its limited actuator dynamics. This motivates the derivation of DCM trajectories that lead to continuous CoP transitions. For a desired DCM position and velocity, the corresponding CoP is obtained from (4) as:

$$r_{CoP} = \xi - \frac{\dot{\xi}}{\omega} \tag{10}$$

This means that a reference trajectory with continuous DCM position and velocity results in a continuous CoP trajectory and external force trajectories. This motivates the use of a third-order polynomial interpolation to "smooth" the vertices of the preliminary DCM reference trajectory (corresponding to smooth transition during double support (DS) phase). Given a desired DS duration t_{DS}, the idea is to compute two points

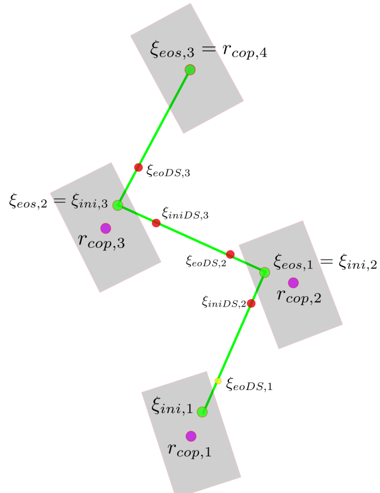
$$\xi_{iniDS,i} = r_{cop,i-1} + (\xi_{ini,i} - r_{cop,i-1})e^{-\omega \Delta t_{DS,ini}} \tag{11}$$

$$\xi_{eoDS,i} = r_{cop,i} + (\xi_{ini,i} - r_{cop,i})e^{\omega \Delta t_{DS,end}} \tag{12}$$

where *iniDS* stands for "initial double support" and *eoDS* stands for "end of double support" on the preliminary DCM trajectory where the DCM before and after support transition would be:

$$\Delta t_{DS,ini} = \alpha_{DS,ini} t_{DS} \tag{13}$$

$$\Delta t_{DS,end} = (1 - \alpha_{DS,ini}) t_{DS} \tag{14}$$



Note: The first step is a little different and we have:

$$\xi_{iniDS,1} = \xi_{ini,1} \tag{11b}$$

$$\xi_{eoDS,1} = r_{cop,1} + (\xi_{ini,1} - r_{cop,1})e^{\omega \Delta t_{DS,end}} \tag{12b}$$

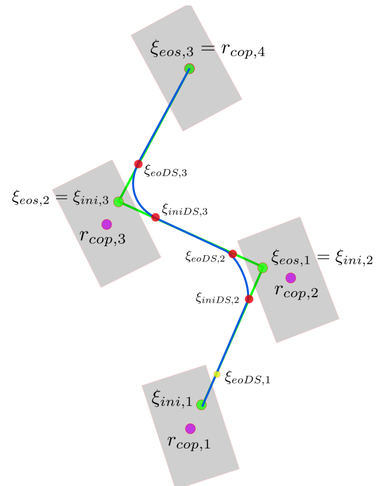
7. We use a cubic interpolation for finding the DCM trajectory in double support (DS) phase. Replace the DCM planned trajectory for the DS with the corresponding part that was for single support (the double support (DS)). In this study, we choose the parameter $\alpha = 0.5$. $\xi_{iniDS,i}$ and $\xi_{eoDS,i}$ (and the corresponding DCM velocities) are used as boundary conditions for the interpolation polynomial. For initial and final DCM position and velocity boundary conditions, a polynomial parameter matrix can be found as:

$$P = \begin{bmatrix} \frac{2}{T_{DS}^3} & \frac{1}{T_{DS}^2} & -\frac{2}{T_{DS}^3} & \frac{1}{T_{DS}^2} \\ -\frac{3}{T_{DS}^2} & -\frac{2}{T_{DS}} & \frac{3}{T_{DS}^2} & -\frac{1}{T_{DS}} \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \xi_{iniDS,i} \\ \dot{\xi}_{iniDS,i} \\ \xi_{eosDS,i} \\ \dot{\xi}_{eosDS,i} \end{bmatrix} \tag{15}$$

where T_{DS} denotes the total duration of the transition. With P, for any time $t \in [0, T_{DS}]$ the DCM position and velocity can be computed as

$$\begin{bmatrix} \xi(t) \\ \dot{\xi}(t) \end{bmatrix} = \begin{bmatrix} t^3 & t^2 & t & 1 \\ 3t^2 & 2t & 1 & 0 \end{bmatrix} P \tag{16}$$

We just replace the part of the DCM trajectory for DS phase with the corresponding part of the preliminary planned DCM trajectory that was only based on single support.



In the above figure, the green line is a preliminary DCM trajectory based only on single support phase, and the blue line is the final DCM trajectory that also includes double support phase. After having the full DCM trajectory, we can find the CoM trajectory by substituting the current DCM and CoM into equation (3) and then calculating a numerical integration for finding CoM position.

Report

Please upload your code, a video of your results, and create a report including details on the method, plots, and answers to the following questions:

Questions

1. Based on equation (5), which physical parameters will affect the rate of divergence of the DCM dynamics?
2. In the DCM motion planning, how do we guarantee dynamic balancing conditions?
3. If we do have dynamic balancing guarantees, why is the robot not able to walk without parameter tuning?
4. In order to achieve stable locomotion, which parameters did you tune and what are their values?
5. Find the cost of transport for this walking planner and compare with the previous matlab homework (3-link walker). Why are these different?
6. Change the step position and duration in the ”Planning Feet Trajectories” section to achieve faster locomotion. What is the fastest walking speed you can achieve and what are the corresponding parameters?