UNIVERSITÀ DI TRENTO
Department of Industrial Engineering

Advanced optimization-based robot control

Homework 1: Reactive Control

Students:
Claudia Gava [257872]
Matthias Cordioli [258797]

Academic Year 2025/2026

1 Weights and Gains in TSID

The first part of the homework involves analyzing the data obtained from the robot simulation, modifying some parameters and observing how the tracking changes as a result.

1.1 Default Case

The first phase involves setting the frequency and amplitude to define the lateral sinusoidal trajectory 1: $f = [0, 0.5, 0]$, $amp = [0, 0.05, 0]$.

Analysis

The theoretical behavior for the defined inputs would involve sinusoidal motion in y and a constant position for the other two axes. It is observed that the robot fails to follow the reference trajectory along any axis. Along the x-axis, the error grows and then stabilizes, but its order of magnitude is quite low. Along z, however, the position oscillates, slowly shifting its mean value towards the reference. In y, the actual position has the same sinusoidal trend as the reference and seems to be in phase, but the amplitude is lower. The velocity and acceleration graphs confirm this analysis of the positions. The velocities have errors similar to the previous ones, although the orders of magnitude are lower, and in x, towards the end, the actual velocity gets very close to the reference. There is, however, a tracking error regarding the initial velocity in y. The accelerations, on the other hand, do not even come close to the desired ones. Probably with the default parameters, the TSID controller must find a compromise among all active tasks, and the CoM task does not have a high enough priority to completely dominate the posture task, which tends to keep the robot in its initial configuration. The result is a "damped" movement that represents the best compromise found by the solver.

1.2 Comparison between Increasing the Weight and Increasing the Gain

The second phase involves working on the configuration file, modifying the weight associated with the CoM task and the Kp gain of the controller in two different cases that are compared:

- Case 1: $w_{com} = 10$, $kp_{com} = 10$
- Case 2: $w_{com} = 1$, $kp_{com} = 100$

Analysis

In both cases, the actual CoM trajectory approaches the sinusoidal reference: in case 1 2 the settling is slower but overall better than in case 2 3, which is slightly out of phase. In x and z, there are errors with respect to the fixed reference, but the order of magnitude and the trends in both cases are similar. The velocities show the same problems as the positions for each axis, but in this case, the errors in x and y for case 1 are about half of those in case 2. For the accelerations in y, the two cases behave differently. In case 1, the reference, desired, and actual accelerations are practically superimposed after 1s, whereas in case 2, the reference and actual trends overlap, but the desired one is out of phase and has a different amplitude. This certainly makes case 1 preferable to the second. In both cases, the references for the x and z axes are zero, and the actual value oscillates around them (in z, the mean value is slightly positive compared to the ref.). Compared to the desired value, however, the actual acceleration deviates significantly in both cases. However, in case 1, the desired signal has an amplitude an order of magnitude lower than in case 2. This means that in the latter case, the controller is commanding very strong and sudden accelerations to instantly cancel out any minimal error. Although effective, this approach results in a harsher control (low stability and vibrations) and high "effort," which would require high torques potentially damaging to the motors of a real robot. In conclusion, for a real robot, the configuration of case 1 is preferable where, by increasing the weight of the CoM task (10 times higher than default), the optimization solver gave absolute priority to the CoM task, sacrificing other lower-weight tasks (like posture) to minimize the tracking error. Increasing the proportional gain (kp) of the controller, on the other hand, makes the latter much more "aggressive" in correcting the error. The second case is certainly fine in simulation and might be suitable for highly reactive systems, where latency is critical and the physical system is designed to withstand abrupt commands.

1.3 Increase in Frequency

The analysis continues with the configuration deemed best from the previous point (case 1). The frequency of the trajectory in y is therefore increased 4: $f = [0, 1, 0]$.

Analysis
Compared to the previous case, it is observed that by increasing the frequency, the actual and reference CoM positions no longer coincide; in fact, the error increases over time. The velocity trends, in turn, do not match, while the accelerations in y partially overlap with the desired ones, but only in the lower part of the wave: the upper oscillation is instead saturated. Furthermore, the peaks reached by the desired acceleration in this case increase their maximum to about $2.7m/s^2$, compared to the previous case of $0.5m/s^2$ (the minimum value remains almost unchanged). In conclusion, although the controller is correct (as demonstrated in point 2), the robot cannot follow the faster trajectory, likely because the system's own physics (motor power, actuator limits, bandwidth, response delays) impose insurmountable limits on the accelerations it can achieve.

2 Importance of Tasks for Biped Walking

In this second part, the TSID controller is used to make the Romeo robot perform a 6-step walk, following pre-calculated reference trajectories. The analysis focuses on the importance of the various control tasks (relative assigned weights), first proceeding with a $0.15m$ step and then lengthening it to $0.30m$.

2.1 Part 1 - Short Steps

In the first part, the configuration with the short step ($0.15m$) is used, and what is observed with the default data is that the robot does not perform any movement, but rather falls to the ground after nonsensical rotations of the torso and arms. The position, velocity, and acceleration graphs confirm that the robot does not follow the reference tracking, and in fact, the actual and reference trends tend to diverge over time. The different weights are then modified (at least 1 order of magnitude), in particular $w_{posture}$, until a configuration is obtained that allows all six steps to be performed.

Analysis
Changing the weight of the different tasks is analogous to varying their priority, assigning more importance to the task with the highest weight. In this case, starting from the default values, one weight was changed at a time to better observe the variations. Since the given references concern CoM tracking anyway, we first raised w_{com}, which influences factors such as weight forces (masses), inertia, and balance. This increases the coincidence between the actual and reference trajectories for position, velocity, and acceleration. In any case, it was not enough, and the robot did not move and fell to the ground. Then, as suggested, $w_{posture}$ was modified, first by increasing it, but not seeing great improvements, and then by decreasing it. The introduction of a very small $w_{posture}$ weight acts as a regularizer for the behavior of the entire robot body, which appears more coordinated in its movements. It was seen that it would also be sufficient to modify only $w_{posture}$ by putting an even smaller value (e.g., 10^{-3}) leaving w_{com} at default: the results would still be good, although the tracking errors in z would be slightly higher; the pressure trends would be closer to the lower limit, but this aspect could perhaps be mitigated by acting directly on the center of pressure (w_{cop}) 5.

Finally, a sufficiently good configuration was found that allowed all 6 steps to be performed, keeping the robot in a stable position at the end of the walk: $w_{com} = 1e2$ (default=1) and $w_{posture} = 1e - 2$ (default=0) 6.

With respect to this good configuration, the effects obtained by modifying the weights of the remaining tasks were analyzed. Both raising and lowering the weight of the pressure task (w_{cop}) causes the robot to fold in on itself and fall to the ground. The angular momentum task (w_{am}) is already set with a fairly low value, and lowering it further does not lead to great differences, while raising it does not penalize the success of the walk but makes the hip movements more pronounced and unnatural, with larger peaks in acceleration, although the pressures remain more contained. The foot task (w_{foot}) controls the flight phase, and if raised, it does not significantly change the result, while if lowered, it causes the robot to have an imprecise foothold already on the second step, then definitively lose balance and fall. Similarly, the weight associated with the foot's ground contact ($w_{contact}$), if raised, does not change the result (it only makes the motion apparently more "fluid" and faster in the visual simulation), but if lowered, it makes the robot fall. The tasks related to dynamics ($w_{forceRef}$, $w_{torque,bounds}$, $w_{joint,bounds}$) do not seem to modify the final result much, and only the forces one, if raised, makes the robot fall backward: perhaps the effects would be more appreciable if there were a specific plot on the efforts to which the motors or joints are subjected.

2.2 Part 2 - Long Steps

This section analyzes the ability of the previously used controller to adapt to a more demanding task, i.e., a longer step ($0.30m$), explaining how the weights of the various tasks will need to be further modified.

Analysis

At this point, the simulation is re-run by setting the file with the longer steps in the configuration and leaving the task weights modified that had allowed the six steps to be completed in the previous point ($w_{com} = 1e2$, $w_{posture} = 1e - 2$). With this setup, the performance seems to start well, but already on the second step, it becomes unbalanced and falls backward 7.

In an a priori analysis, it can certainly be said that the trajectory with 0.30 m steps is more dynamic: it requires greater accelerations, wider leg movements, and a greater excursion of the CoM. This will increase the conflict between the various tasks. For example, moving the leg quickly to take a long step (w_{foot}) will inevitably damage the CoM trajectory (w_{com}). This is why the balance found for the previous point is no longer sufficient in this case.

What was seen right away is that the most critical factor is related to the robot's balance, which very easily tends to become unbalanced backward. To have good balance, good foot placement and good management during the flight phase are necessary, which is when the body can most easily become unbalanced. Focusing on this aspect and putting the CoM task in the background ($w_{com} < w_{foot}$), $w_{com} = 1$ was set back to default, and $w_{foot} = 1e2$ (default=1) was raised, leaving $w_{posture} = 1e - 2$. By doing so, a small momentary error on the CoM is overlooked and can be corrected in the next step. If, on the contrary, the ratio between the two factors is similar, the robot becomes "too cautious," refusing to perform the wide leg movement required so as not to disturb the CoM, and thus failing the final task. With this new configuration, the robot seems to proceed well, but it takes 5 out of 6 steps and falls on the last one. Evidently, a wrong step compromises the next one, and the error increases during the walk 8. This problem was solved by also raising $w_{am} = 1e - 4$ (default=1e-6) 9.

The graphs of the trends in this case are good, although not excellent, given that there are non-zero errors and that the desired maximums and minimums are not always reached. The actual tracking often does not perfectly follow the reference, and this is already visible in the CoM positions and obviously then amplified in the accelerations. However, the overall behavior is good, and this loss of precision was necessary to achieve the final goal, which was to have the robot perform all 6 steps.

3 Plots

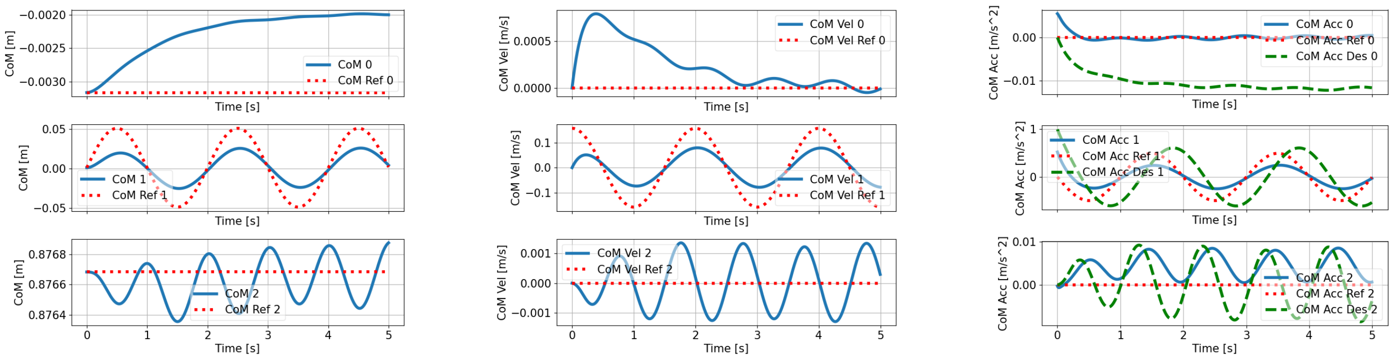

Figure 1: Pos, Vel, Acc of COM with default configuration

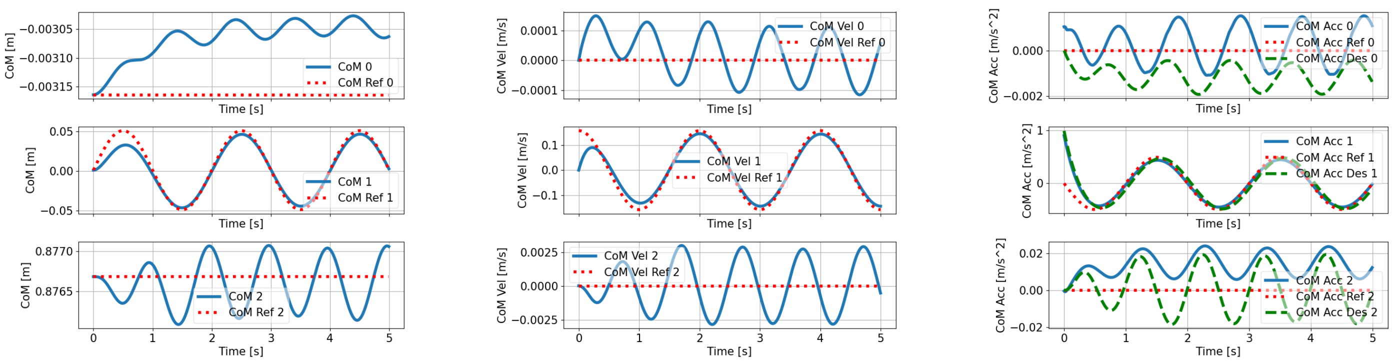

Figure 2: Pos, Vel, Acc of COM with configuration of case 1

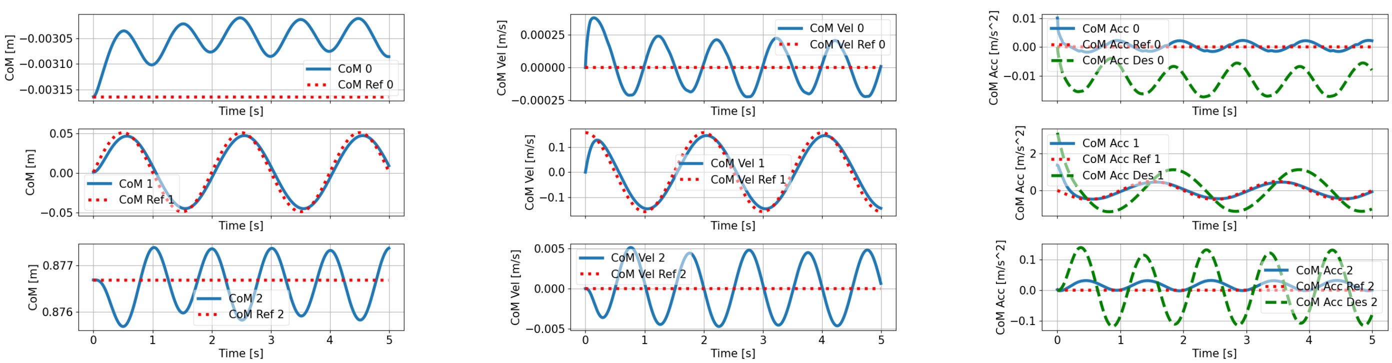

Figure 3: Pos, Vel, Acc of COM with configuration of case 2

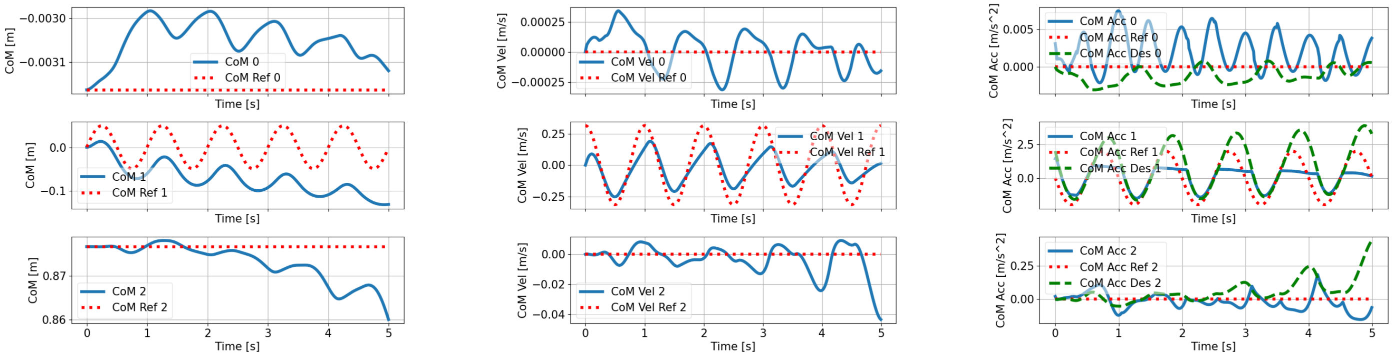

Figure 4: Pos, Vel, Acc of COM with higher frequency configuration

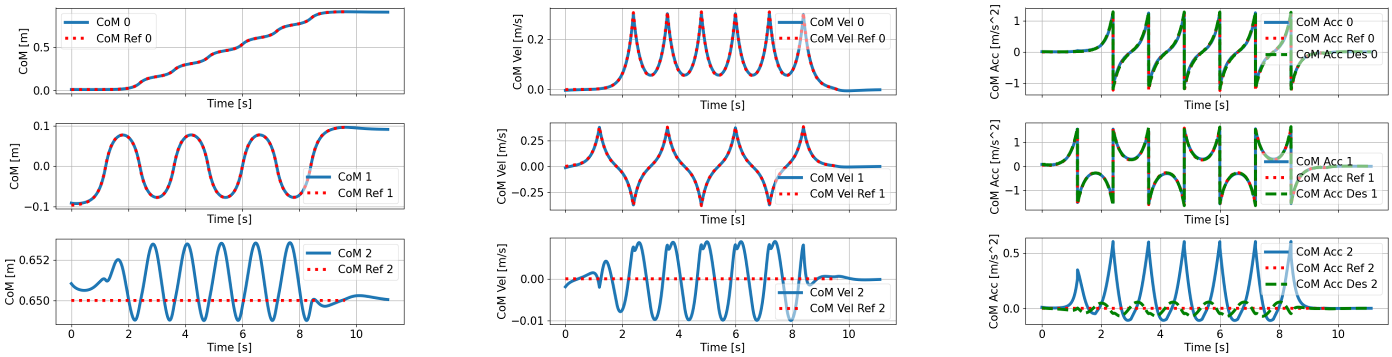

Figure 5: Pos, Vel, Acc of COM with only the modification of *w_posture*

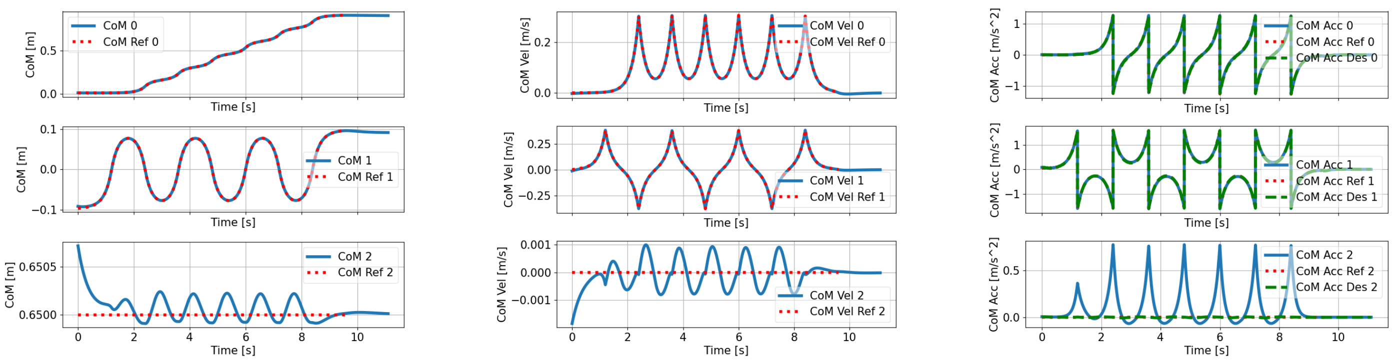

Figure 6: Pos, Vel, Acc of COM with the modification of *w_posture* and *w_com*

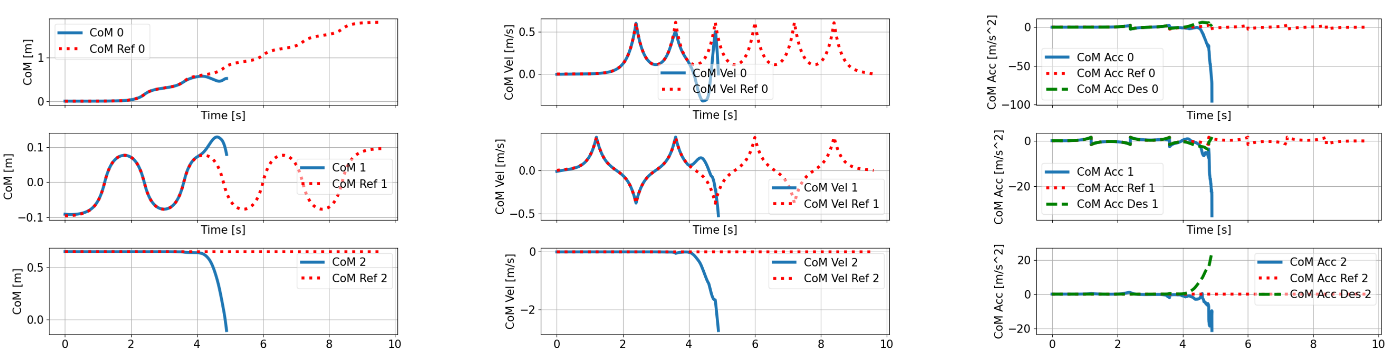

Figure 7: Pos, Vel, Acc of COM with the setup of the previous step

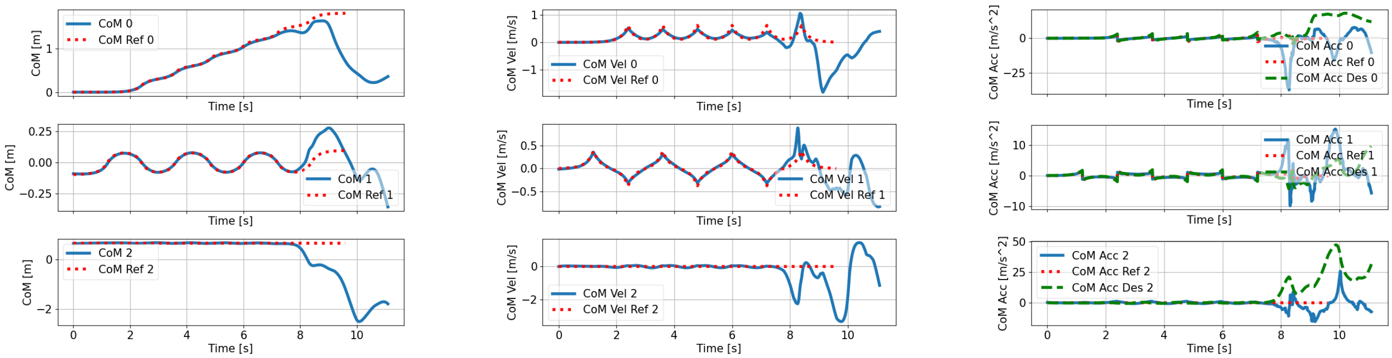

Figure 8: Pos, Vel, Acc of COM with the modification of $w\_posture$ and w_foot

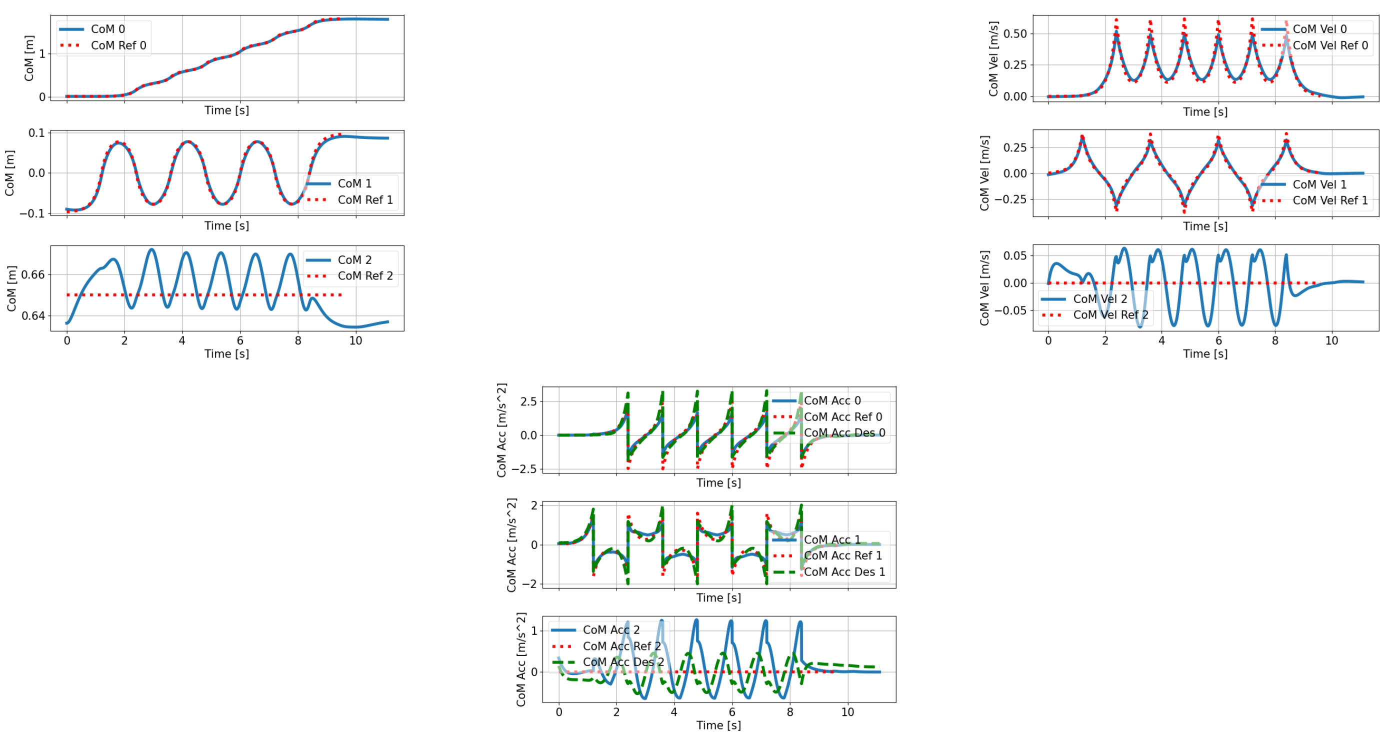

Figure 9: Pos, Vel, Acc of COM with the modification of $w\_posture$, w_foot and w_am