Torella Gaetano - Homework 4

1. Describe the buoyancy effect and why it is considered in underwater robotics while it is neglected in aerial robotics.

   In the analysis of underwater robots, one of the most important factors to take into account is the buoyancy effect; in fact, when a rigid body is submerged in a fluid under the effect of gravity, an additional inertia of the fluid surrounding the body has to be considered. This effect is described by the Archimedes' principle which states that a body immersed in a fluid, such as water, experiences an upward force (buoyancy) equal to the weight of the fluid displaced by the body; this force determines how the body behaves in the water, i.e. whether it floats, sinks, or remains in balance. The additional inertia provided by this effect is accelerated by the movement of the body and is necessary to provide a force to achieve this acceleration in order to compensate the reaction force that the fluid exerts on the robot. The added mass effect is function of the body's geometry and it provide both Coriolis and centripetal contributions. The buoyancy force is a hydro static effect and represent a force that acts on a point, the center of buoyancy, opposing the gravity force and lifting the body immersed in a fluid. The buoyancy force in the body frame can be formally expressed as follows:

$$f_b^b = -R_b^T \begin{bmatrix} 0 \\ 0 \\ \rho * \Delta * g \end{bmatrix} \tag{1}$$

   Where g is the gravitational acceleration, Δ is the volume of the body, ρ is the density of the fluid and R_b^T transpose of the rotation matrix between body frame and world frame. As said before, the force contribution provided by the buoyancy effect depends on the density of the fluid in which the body is submerged; since the density of air is much lower respect to the density of water and of the moving mechanical system, the buoyancy effect is neglected in aerial robotics.

2. Briefly justify whether the following expressions are true or false.

   - The added mass effect considers an additional load to the structure.
     FALSE: the added mass effect does not refer only to an additional load or mass being physically added to the structure; in fact, this effect must be taken into account in the dynamic model of a structure moving through a fluid. In the case of a completely submerged underwater robot, the added mass effect provides an additional inertia, modifying the inertia matrix which is no longer positive definite. Thus, the added mass effect is more than just an additional load because it fundamentally alters the dynamic properties of the system.
   - The added mass effect is considered in underwater robotics since the density of the underwater robot is comparable to the density of the water.
     TRUE: it's important to consider the added mass in underwater robotics because, for the Archimedes' principle, the water provides a boost to the robot since the density of the water is comparable to that of the robot. During the underwater motion, the robot displace the surrounding water which provides an additional force due to the inertia of this; this force must be taken into account during the modelling of the robot dynamic. The effect of this added mass is not considered in applications like aerial mobile robots because, always for Archimedes' principle, the density of air is much lower than that of the UAV; the lower density of the air makes its displacements generated by inertial forces negligible, therefore, the effect of added mass has no significant impact on UAV dynamics and can be safely ignored.
   - The damping effect helps in the stability analysis.
     TRUE: In fact, the damping effect plays a crucial role in stability analysis, acting as a stabilising force within dynamic systems. It serves to decrease oscillations and helps stabilise the system by dissipating energy, a vital aspect in maintaining control and stability, especially in complex and dynamic environments. In the field of aerial and underwater robotics, this effect stems from the viscosity of the fluid surrounding the robot, resulting in dissipation of drag and lift forces on the body. Particularly in the field of underwater robots, given the high viscosity of water, damping becomes particularly important, where precise control and stability are essential to operate effectively in harsh aquatic environments.

- The Ocean current is usually considered as constant, and it is better to refer it with respect to the body frame.
  FALSE: The ocean current represents one of the main disturbances for underwater robots and is usually considered constant and irrotational but, in the analysis of underwater robot control, it is best expressed with respect to the world frame and not the body frame. In fact, if expressed in the body frame, the ocean current changes continuously as the underwater robot's pose changes, while with a more proper notation in the world frame it can be seen as a constant vector whose components are shown in (eq: 2). The effect of the ocean current is introduced into the dynamic model of the robot by accounting for the relative velocity in the body-fixed frame (eq: 3).

$$\nu_c = \begin{bmatrix} \nu_{c,x} \\ \nu_{c,y} \\ \nu_{c,z} \\ 0 \\ 0 \\ 0 \end{bmatrix} \in R^6, \dot{\nu}_c = 0_6 \qquad (2) \qquad\qquad v_r = \begin{bmatrix} \dot{p}_b^b \\ \omega_b^b \end{bmatrix} - R_b^T v_c \qquad (3)$$

3. Consider the Matlab files within the quadruped_simulation.zip file. Within this folder, the main file to run is MAIN.m. The code generates an animation and plots showing the robot's position, velocity, and z-component of the ground reaction forces. In this main file, there is a flag to allow video recording (flag_movie) that you can attach as an external reference or in the zip file you will submit. You must:

- implement the quadratic function using the QP solver qpSWIFT, located within the folder (refer to the instructions starting from line 68 in the file MAIN.m); To implement the quadratic function using the QP solver to impose the desired ground reaction forces and desired reference for CoM to the robot the parameters regarding the cost function and equality and inequality constraints were passed through the qpSWIFT function as follow:

```
[zval,basic_info,adv_info] = qpSWIFT(sparse(H),g,sparse(Aeq),beq,sparse(Aineq),
    bineq);
```

  Where:
  - zval is the variable containing the results of the quadratic problem
  - H is the sparse matrix of dimension (nxn) that represents the cost matrix
  - g is a dense column vector of size n
  - Aeq and Aineq are sparse matrices of dimension (pxn) and (mxn) respectively with p number of equality constraints and m number of inequality constraints
  - beq and bineq are dense column vectors of size p and m respectively

  The qpSWIFT function solve the quadratic problem in the following form:

$$\begin{aligned} min \ \tfrac{1}{2}x\prime \ H \ x + g\prime \ x \qquad s.t. \\ A_{eq} \ x = b_{eq} \\ A_{ineq} \ x \leq b_{ineq} \end{aligned} \qquad (4)$$

- modify parameters in the main file, such as the gait and desired velocity, or adjust some physical parameters in get_params.m, such as the friction coefficient and mass of the robot. Execute the simulation and present the plots you find most interesting: you should analyze them to see how they change with different gaits and parameters and comment on them.
  The first analysis was done for all the gaits maintaining the default parameters; from the following plots can be notice that varying the different gaits the control is able to follow the velocity and position references of the CoM quite well. The best gaits in this analysis, in terms of tracking the position and velocity references, are the trot (fig: 1), the running trot (fig: 5) and the crawl (fig: 6); in particular, comparing the first two gaits, can be noticed that trot has a better behavior than running trot, while the crawl gait proves to be more precise and stable than the others even if it has some oscillations in terms of angular velocity, this is because there are always three feet in position during the movement. As far as pacing gait (fig: 3) is concerned, considerable oscillations can be seen both in terms of linear and angular velocity, but nevertheless the system is quite regular with a slight error in the tracking of the position. The gallop gait (fig: 4) has a constant error in the position tracking but a considerable increase in reaction forces compared to the other gaits due

to the particular characteristics of this gait. The worst one is the bound gait (fig: 2) where it's possible to see an increasing error in the tracking of the position along the x-axis and a velocity profile with a noticeable non-constant error.

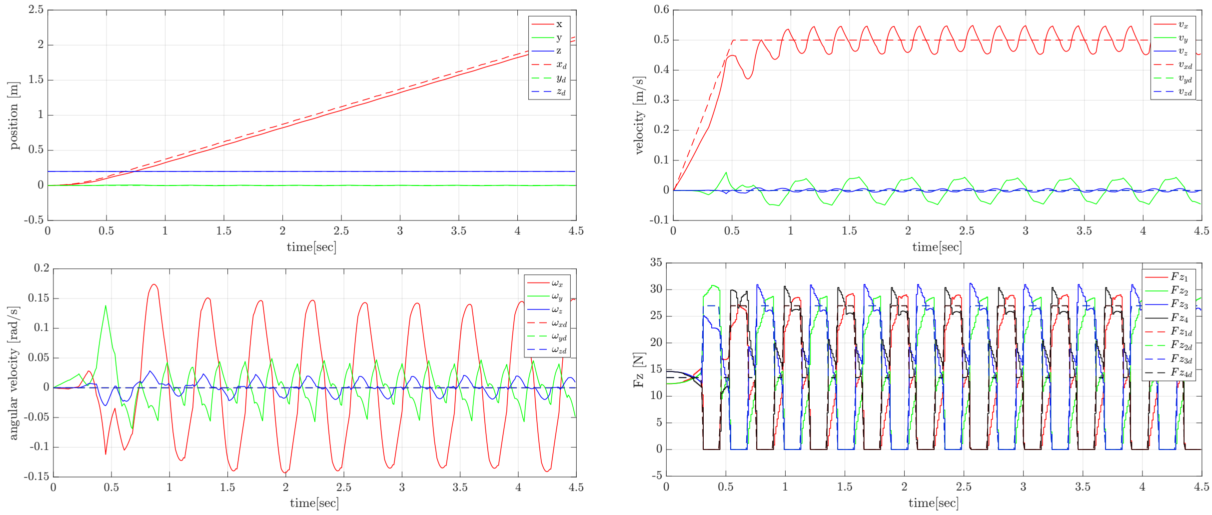


Figure 1: Trot gait

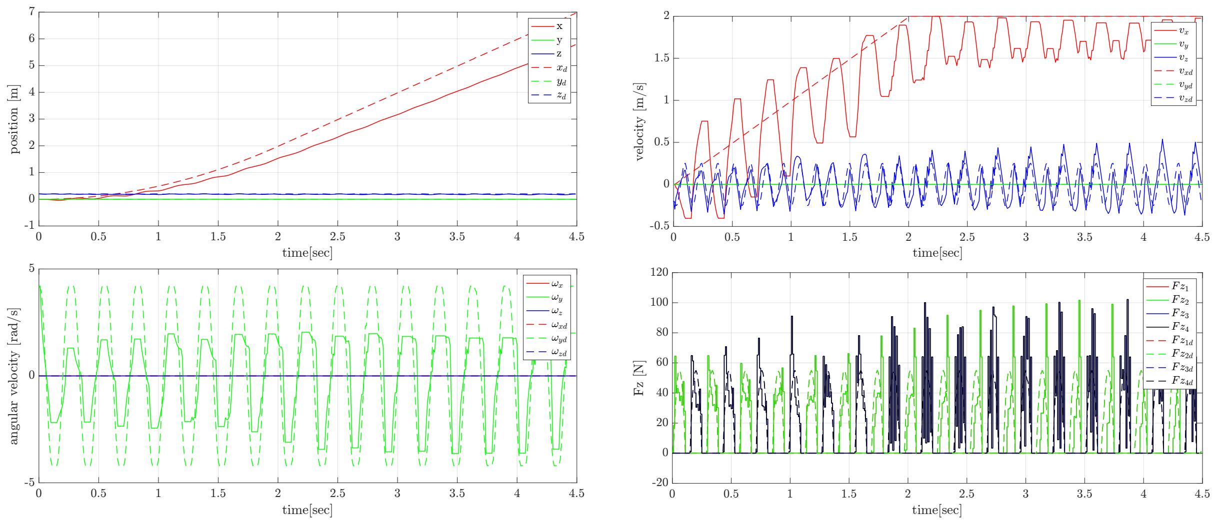


Figure 2: Bound gait

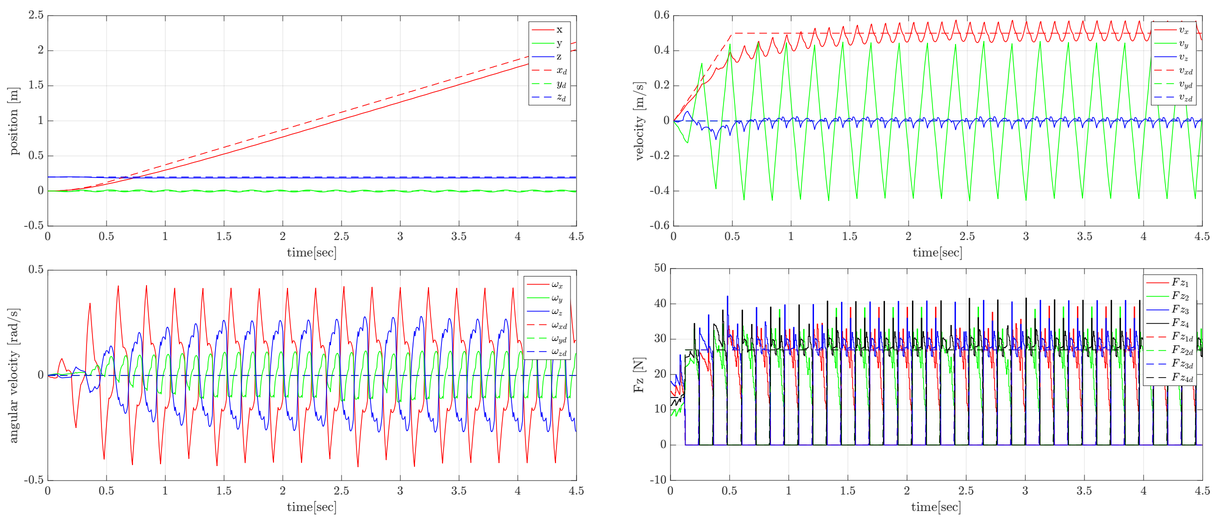


Figure 3: Pacing gait

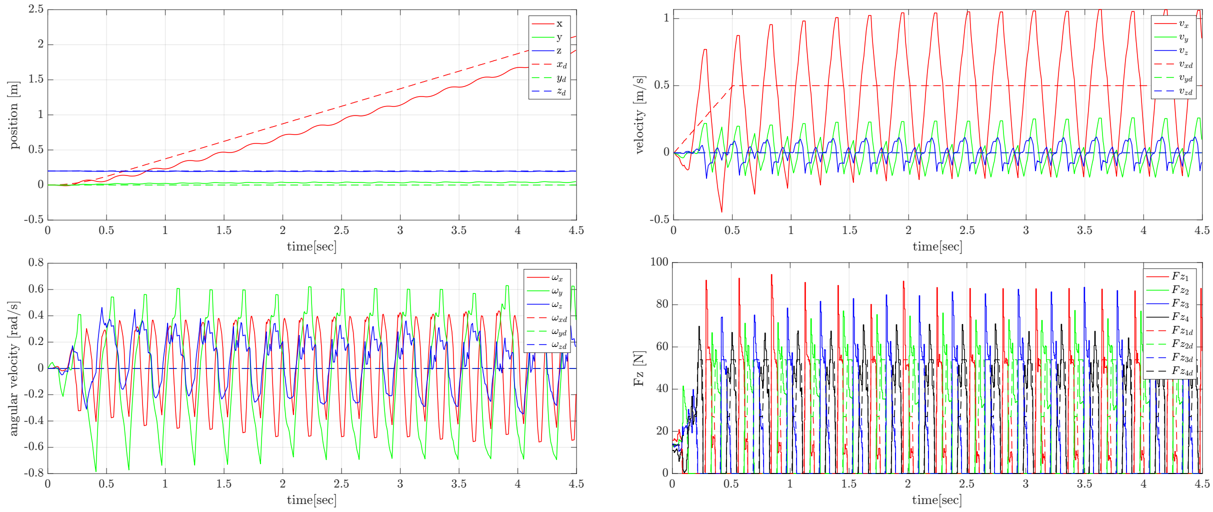


Figure 4: Gallop gait

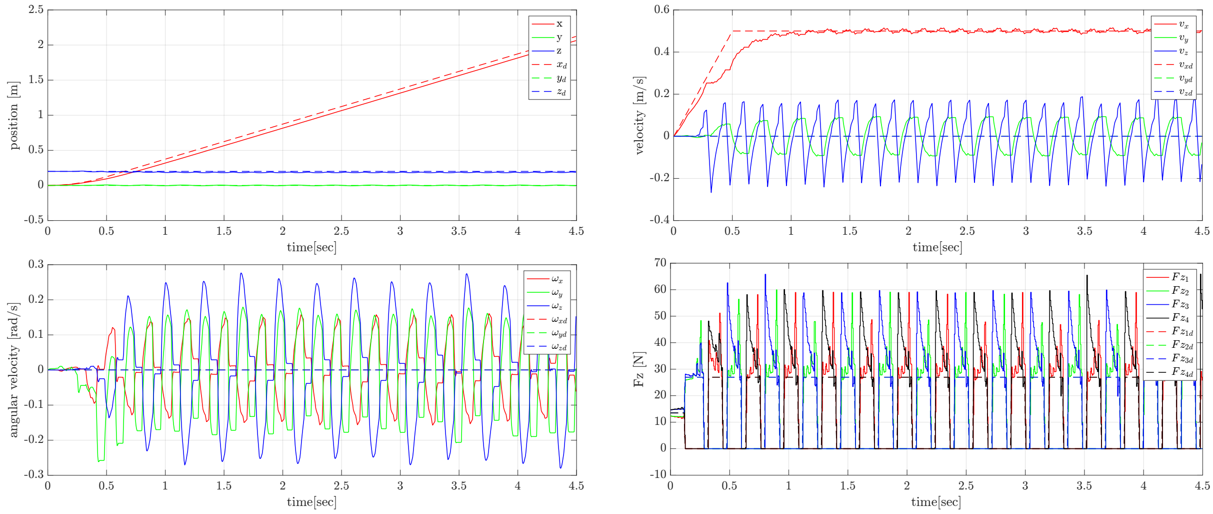


Figure 5: Trot run gait

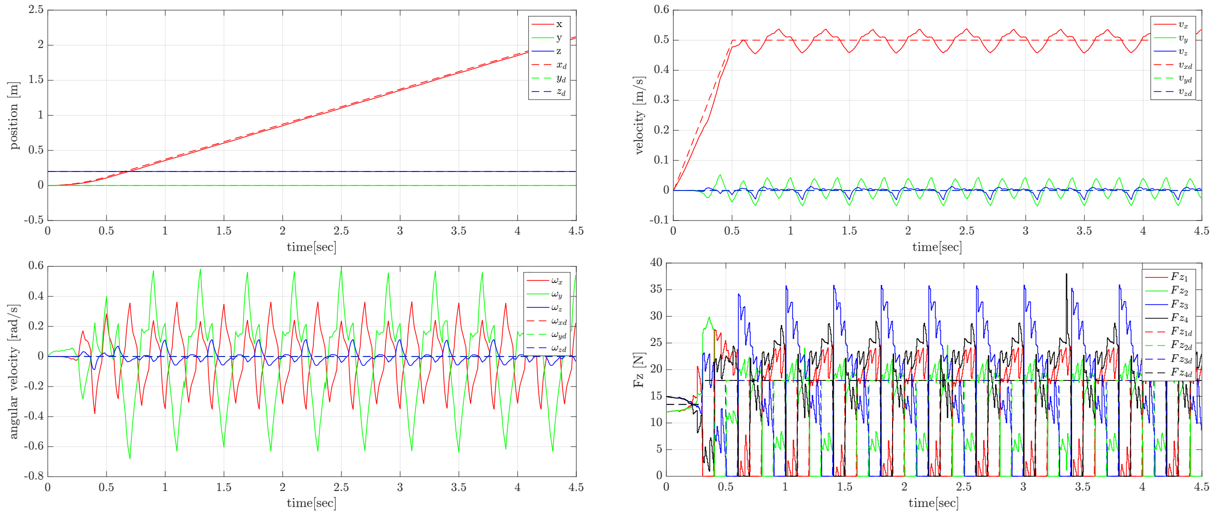


Figure 6: Crawl gait

Another analysis was done increasing the value of the desired velocity to $v_d = 2.5m/s$. It's possible to notice that the system is less stable with this velocity even for the trot gait, in fact the robot cannot maintain its balance correctly as shown by the angular velocities plot and the video frame (fig: 7). This loss of balance can also be seen in other gaits, in particular in galloping as shown by (fig 8) and (fig: 9) where the errors of position and velocity increase during the simulation and the robot rotates. The trot running gait improves (fig: 10), but the crawl one is still the best (fig: 11).

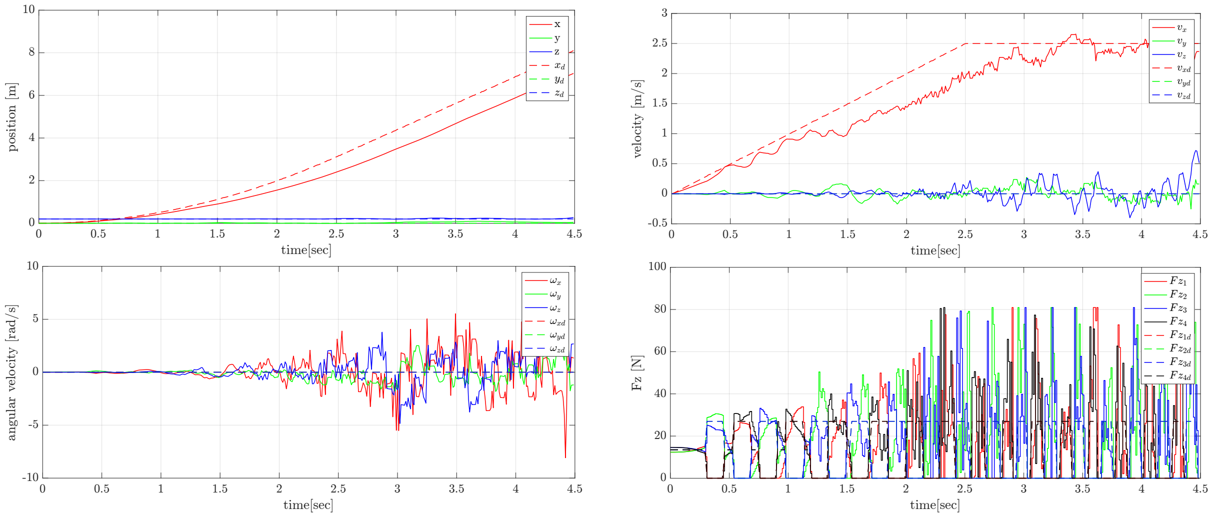
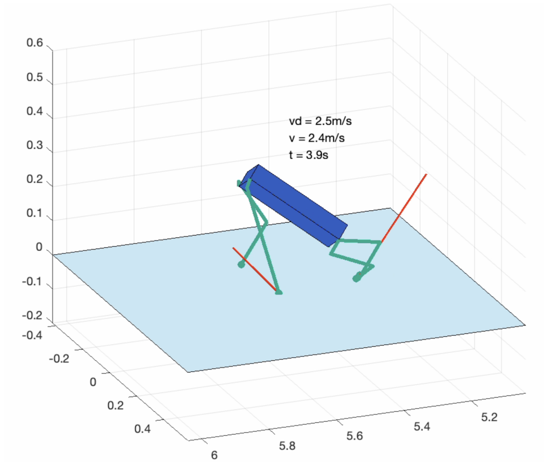


Figure 7: Trot gait $v_d = 2.5m/s$



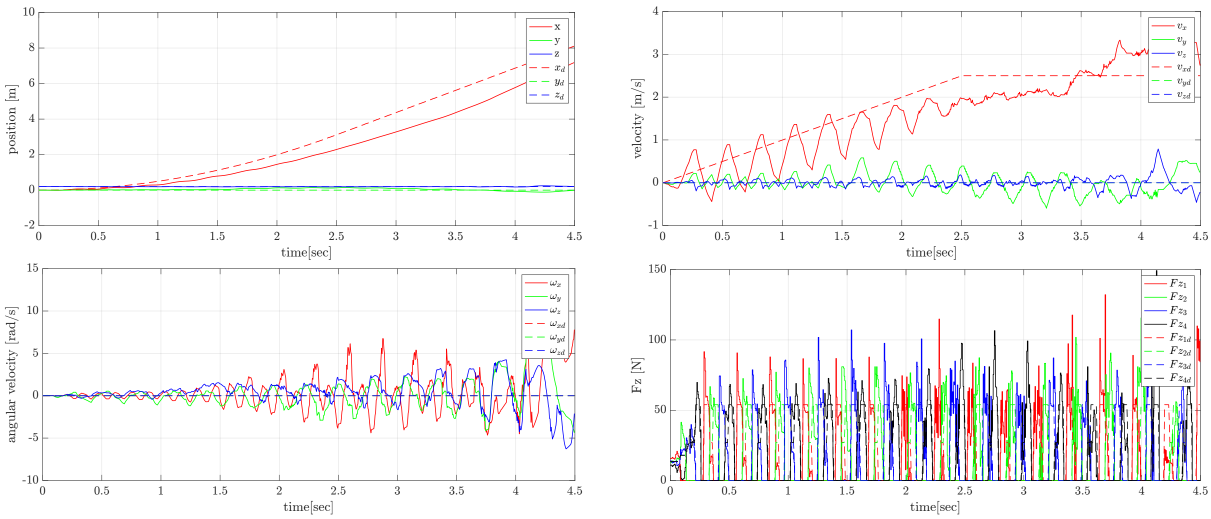


Figure 8: Gallop gait $v_d = 2.5m/s$

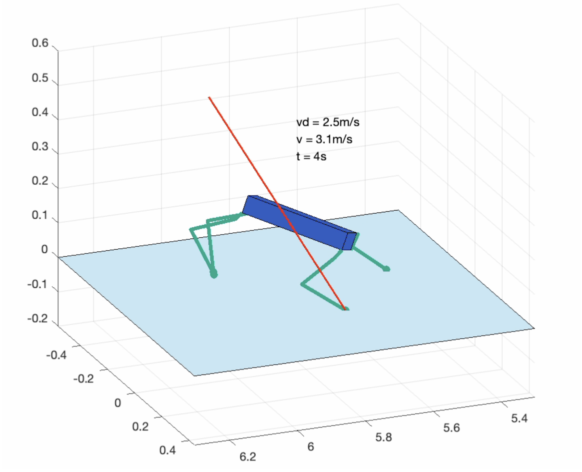


Figure 9: Video frame gallop gait $v_d = 2.5m/s$

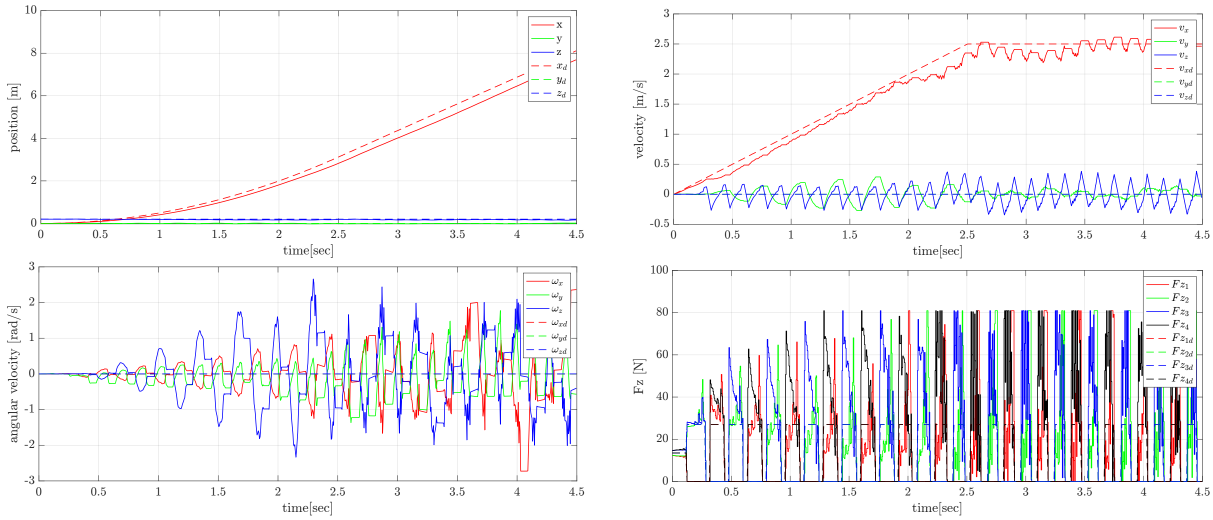


Figure 10: Trot run gait $v_d = 2.5m/s$

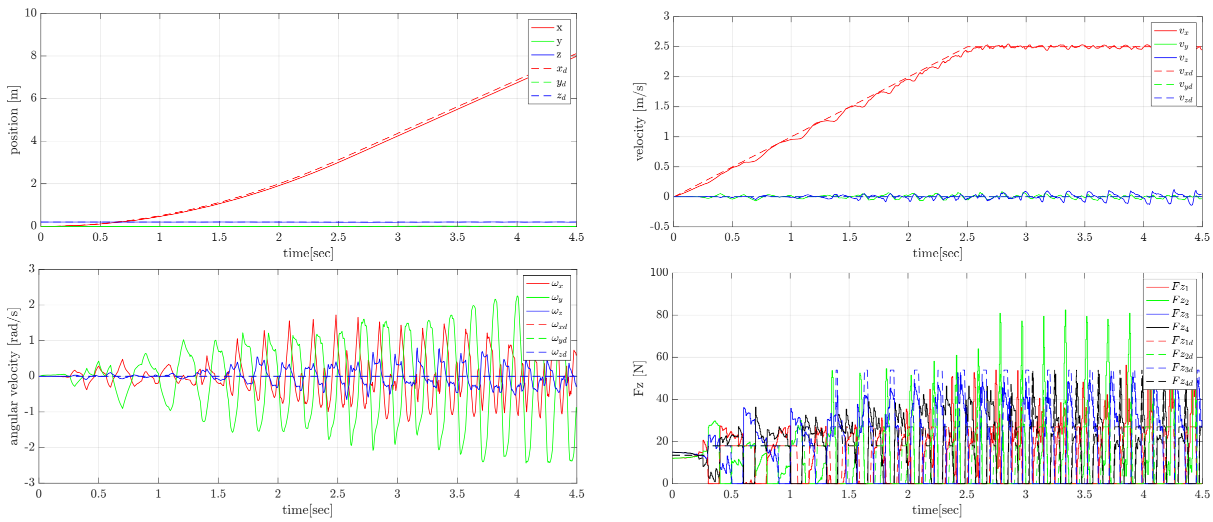


Figure 11: Crawl gait $v_d = 2.5m/s$

The following analysis was done decreasing the value of the mass of the robot to $m = 2.5kg$. It is easy to see how for each gait the reaction forces F_z decrease as the mass of the robot is decreased, and a decrease in the oscillations in angular velocity for all gaits except the trot one (fig: 12) where they increase. With regard to the pacing gait (fig: 14), an improvement in the tracking of the position and speed reference is noted, the gallop gait appears mostly unchanged, while the best trend is confirmed to be the crawl gait with a better tracking of the reference speed (fig: 16).

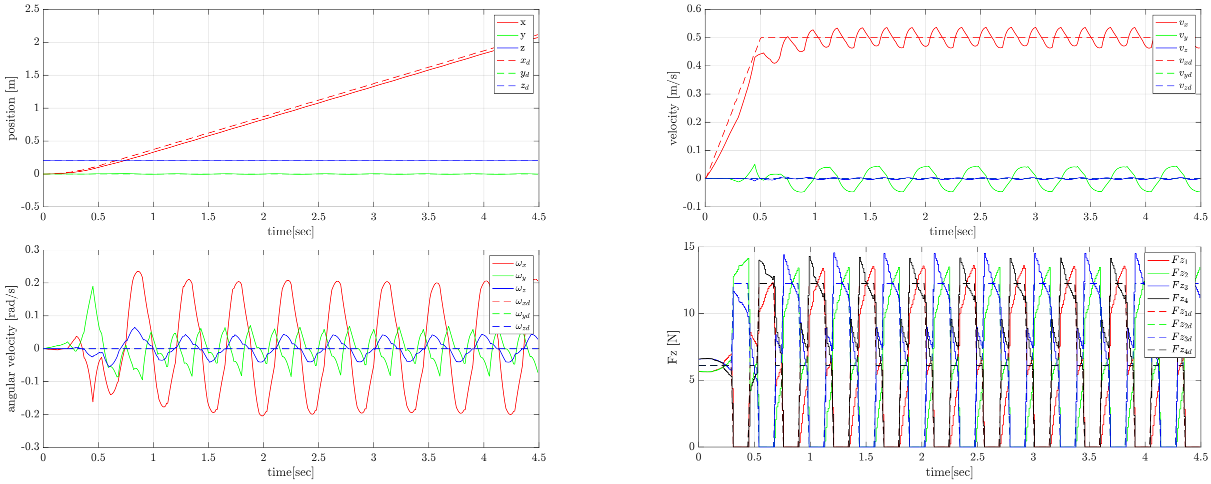


Figure 12: Trot gait $m = 2.5kg$

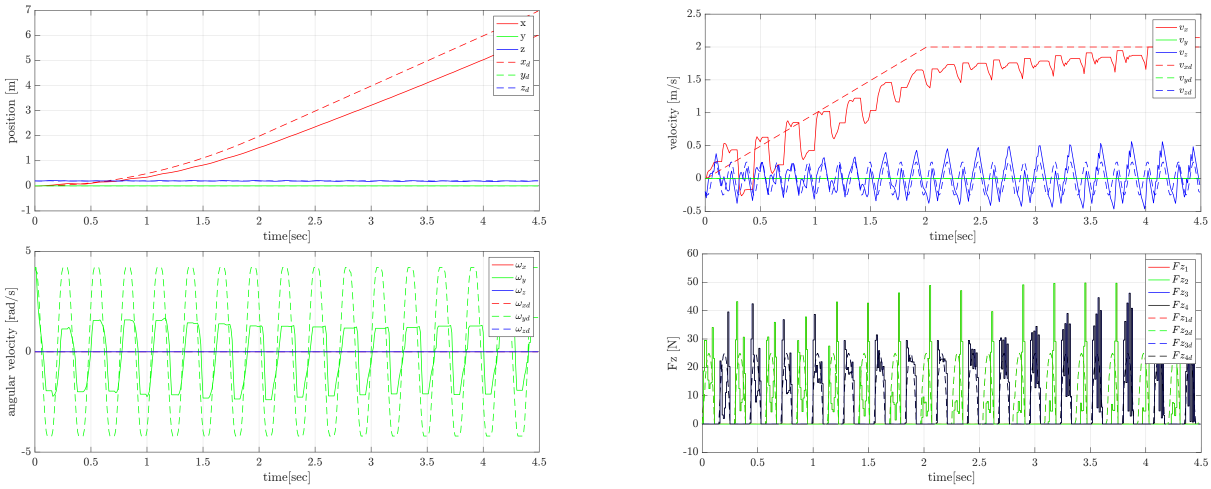


Figure 13: Bound gait $m = 2.5kg$

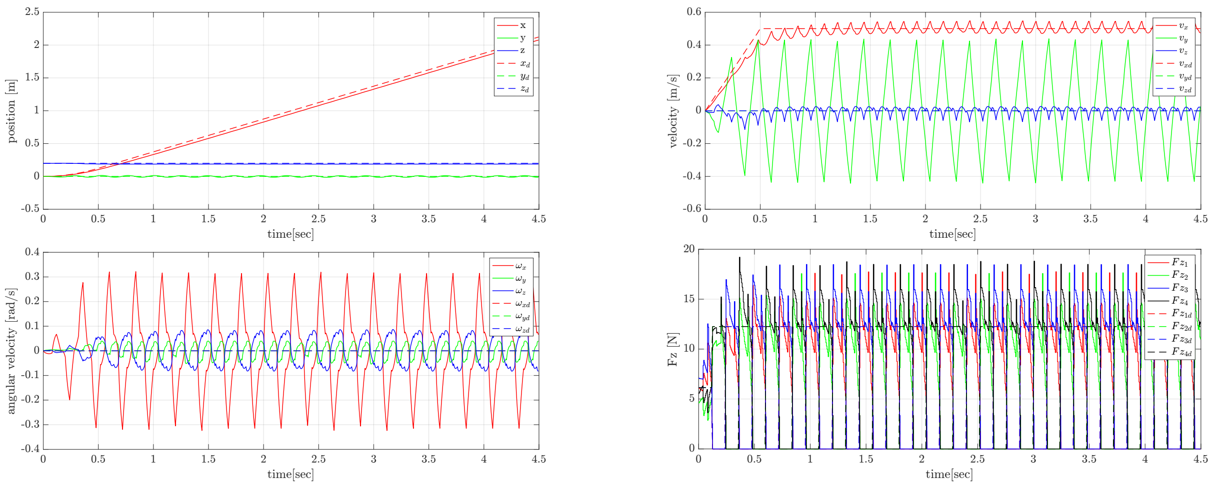


Figure 14: Pacing gait $m = 2.5kg$

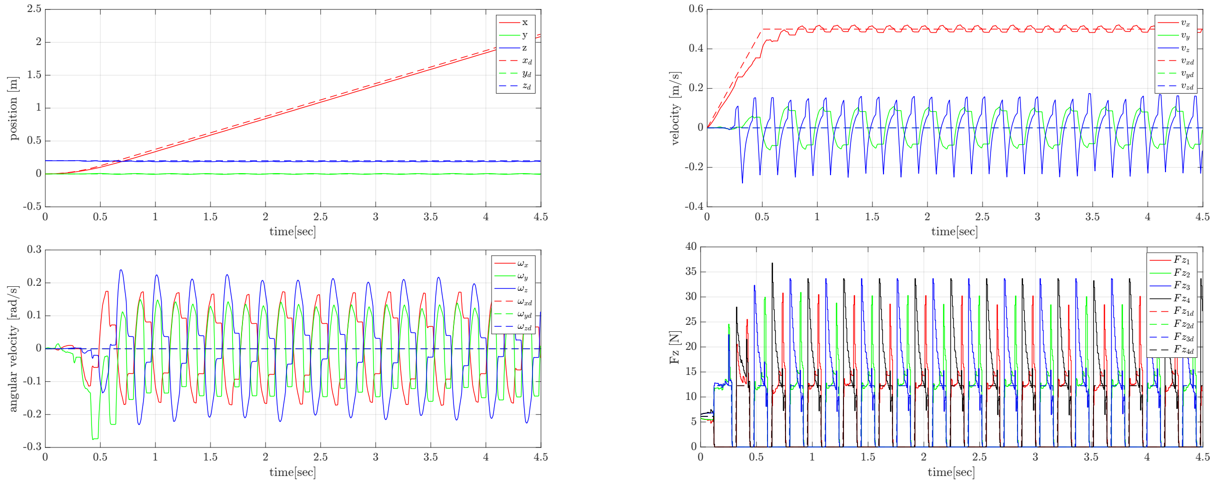


Figure 15: Trot run gait $m = 2.5kg$

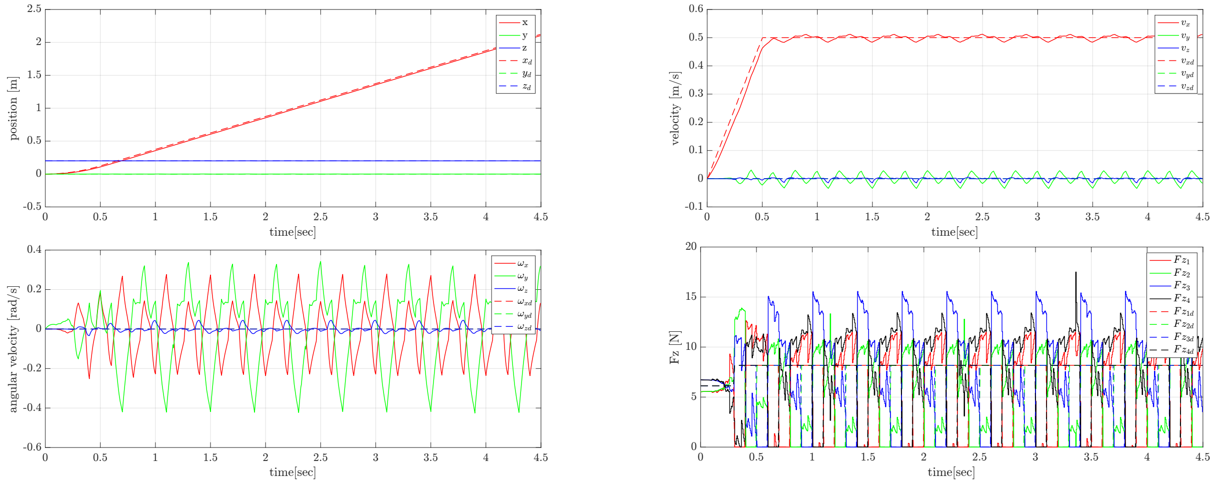


Figure 16: Crawl gait $m = 2.5kg$

Another analysis was carried out by increasing the value of the mass to $m = 10.0kg$ and, in contrast to the previous case in which the mass of the robot was decreased, for each gait the reaction forces F_z increase as the mass of the robot is increased. The increase in the mass of the quadruped caused a serious problem for the bound gait (fig: 17), which even leads the robot to compenetrate with the ground as shown in the figure (fig: 18), causing a completely incorrect simulation.

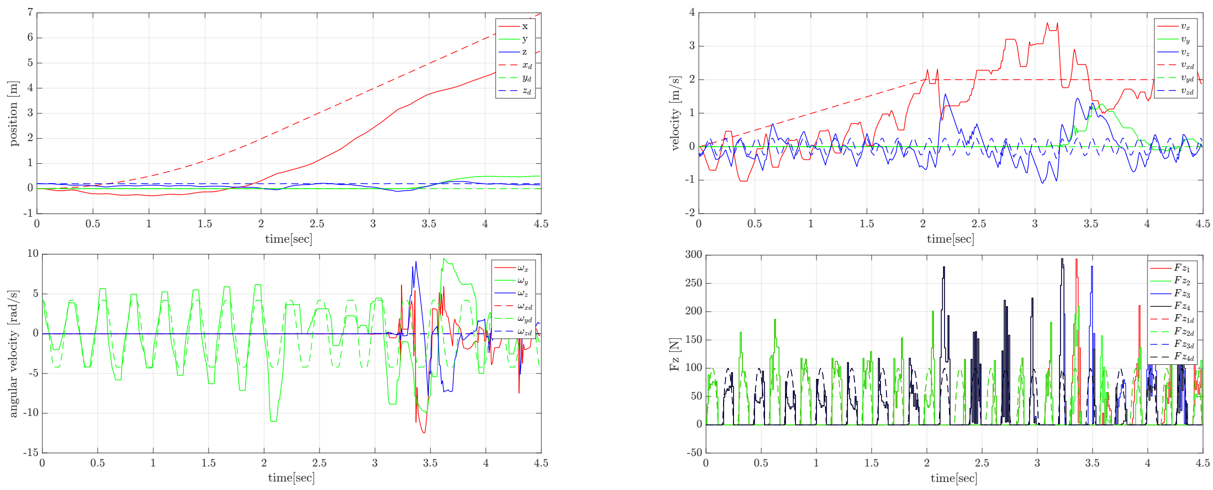


Figure 17: Bound gait $m = 10.0kg$

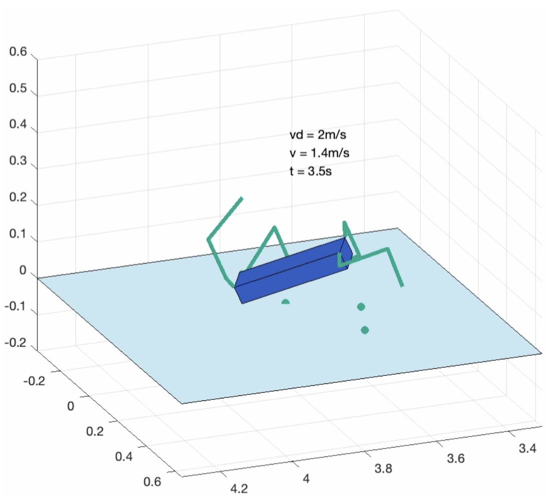


Figure 18: Video frame bound gait $m = 10.0kg$

From the following images, it can be seen that the increase in the robot's mass brings further disadvantages as in the case of pacing (fig: 19) and gallop (fig: 20) in which there is an increase in positional error and even a doubling of angular velocity as in the case of trot run (fig: 21) and pacing gaits (fig: 19).

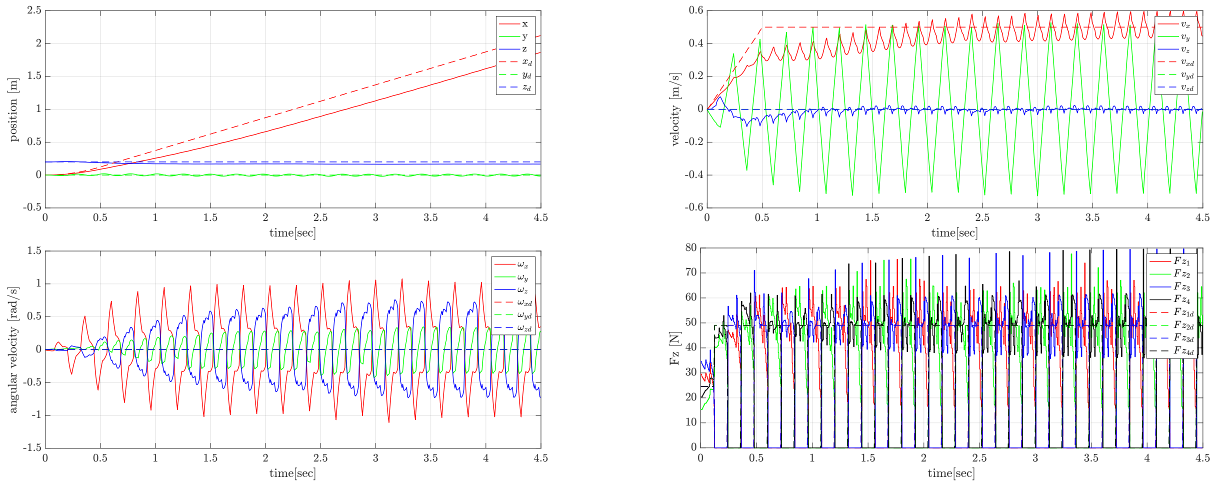


Figure 19: Pacing gait $m = 10.0kg$

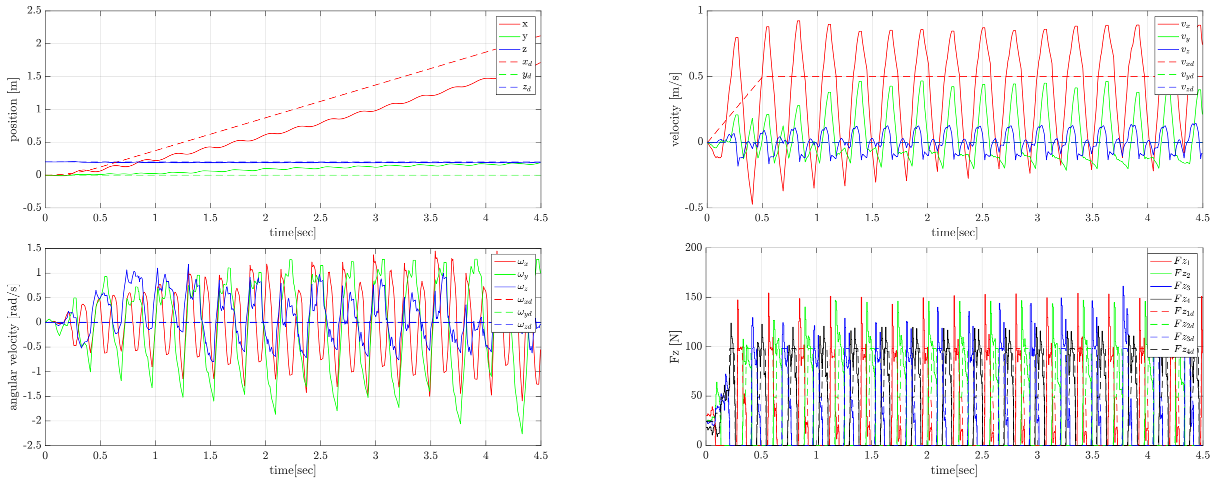


Figure 20: Gallop gait $m = 10.0kg$

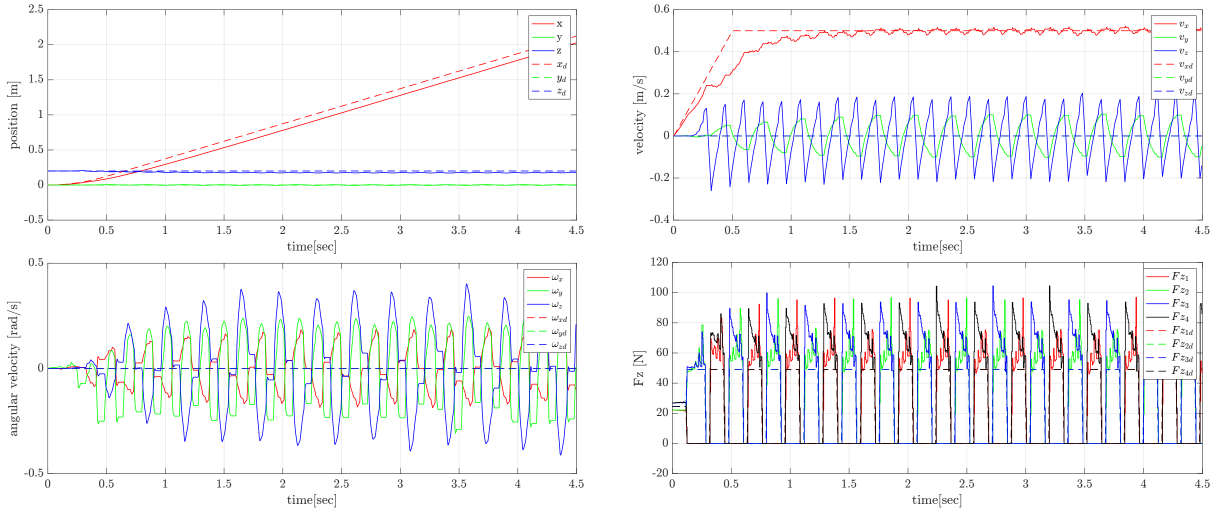


Figure 21: Trot run gait $m = 10.0kg$

The last analysis was done by varying the value of the friction coefficient to $\mu = 0.5$ where no major changes appeared for the trot gait, while the bound gait showed a notable error in linear velocity and position (fig: 22) especially in the last interval of the simulation.

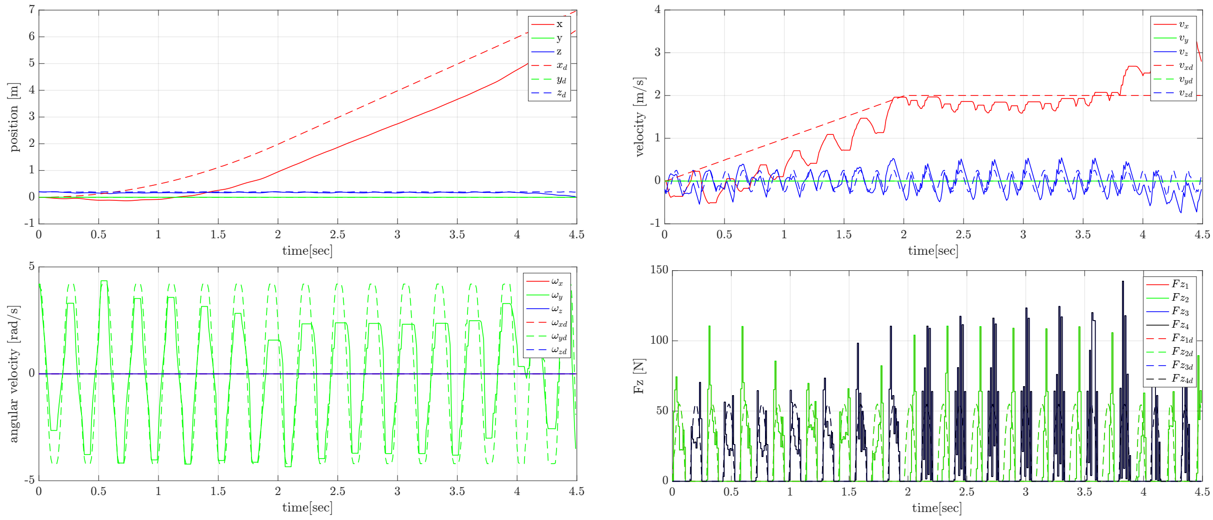


Figure 22: Bound gait $\mu = 0.5$

The pacing gait (fig: 23) showed considerable error in both angular and linear velocity and position along the z-axis as shown by the frame of the simulation video in which the robot goes under the surface of the plane (fig: 25a). With regard to the gallop gait (fig: 24), it is possible to notice an increasing position error and a considerable increase in angular velocity causing a loss of balance of the quadruped; in fact, the robot tends to rotate during the trajectory around the z-axis as shown in the figure (fig: 25b) Instead, the change in the coefficient of friction μ does not affect the running trot (fig: 26) and crawl gait (fig: 27).

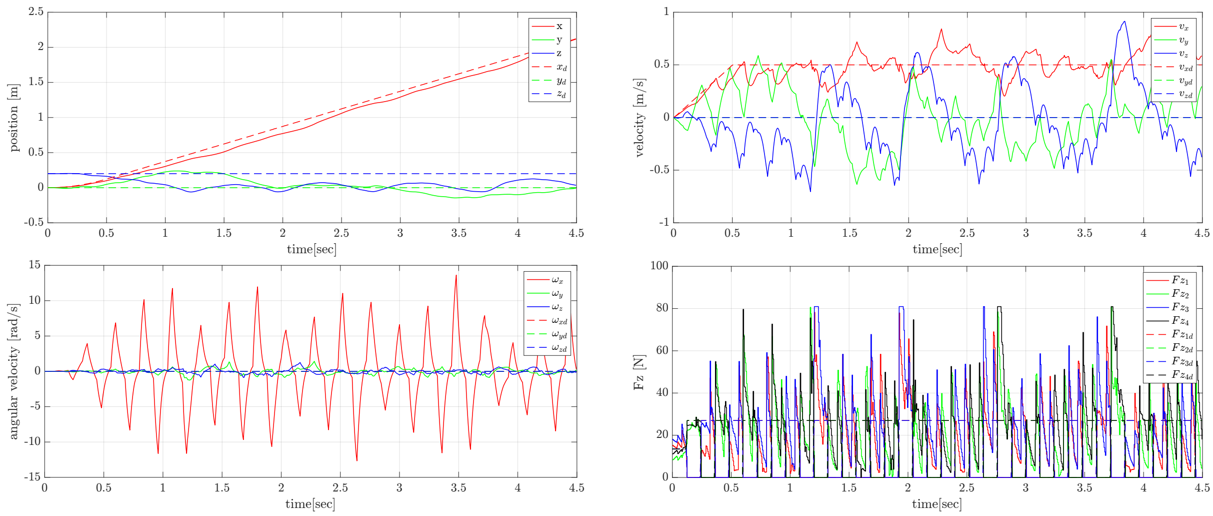


Figure 23: Pacing gait $\mu = 0.5$

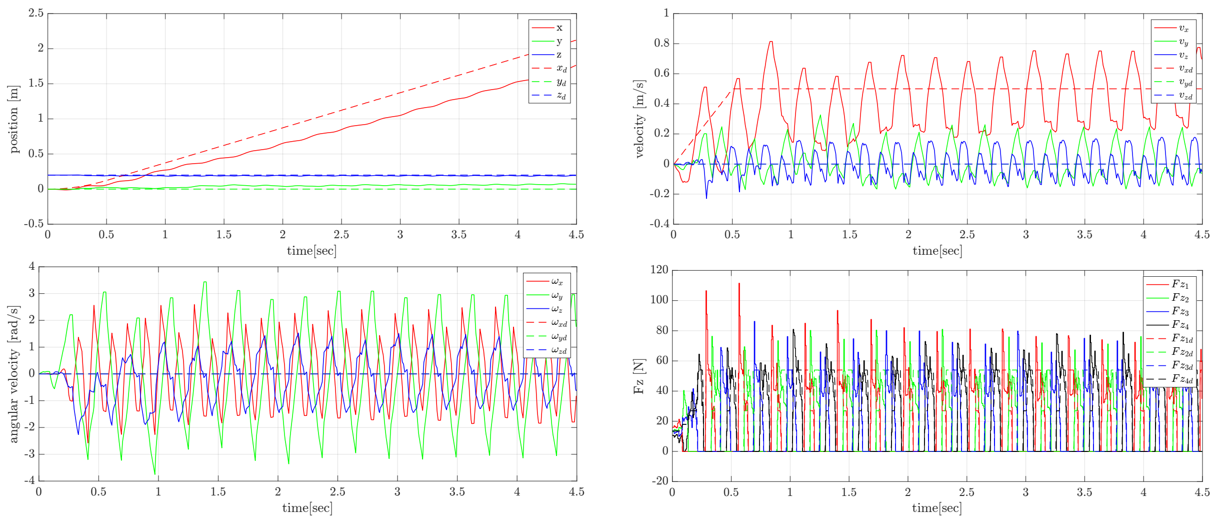
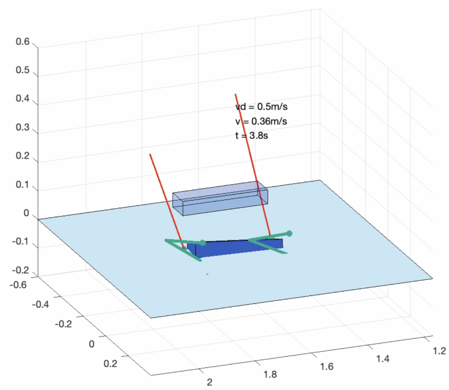
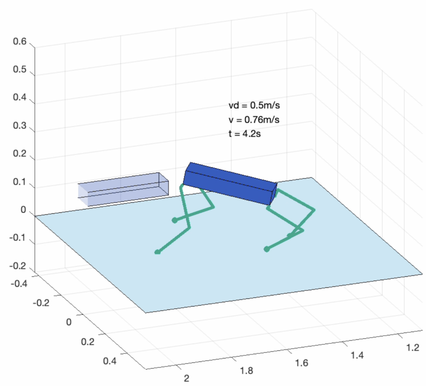


Figure 24: Gallop gait $\mu = 0.5$



(a) Frame pacing $\mu = 0.5$



(b) Frame gallop $\mu = 0.5$

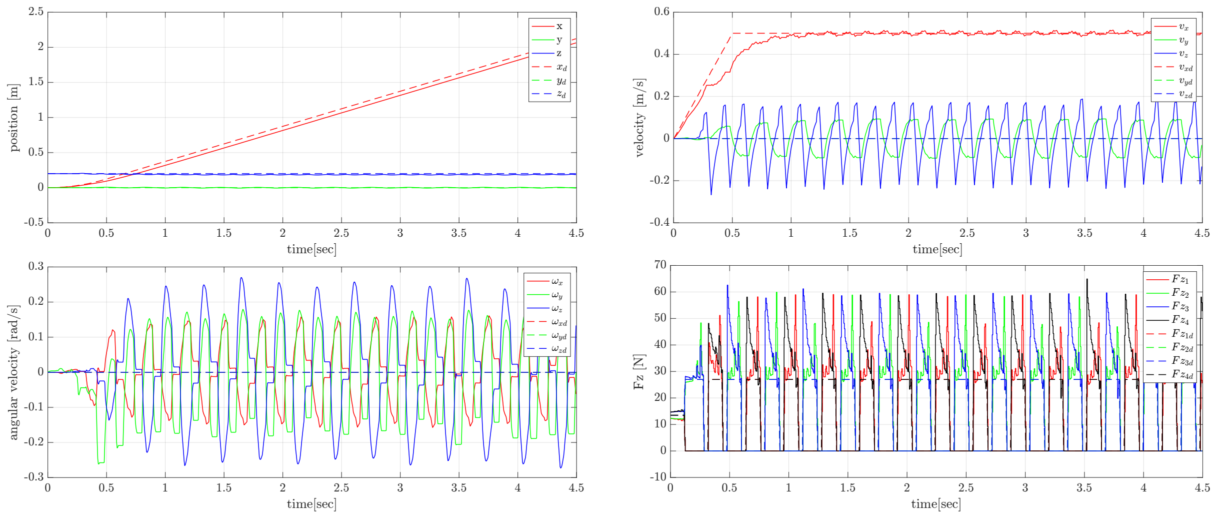


Figure 26: Trot run gait $\mu = 0.5$

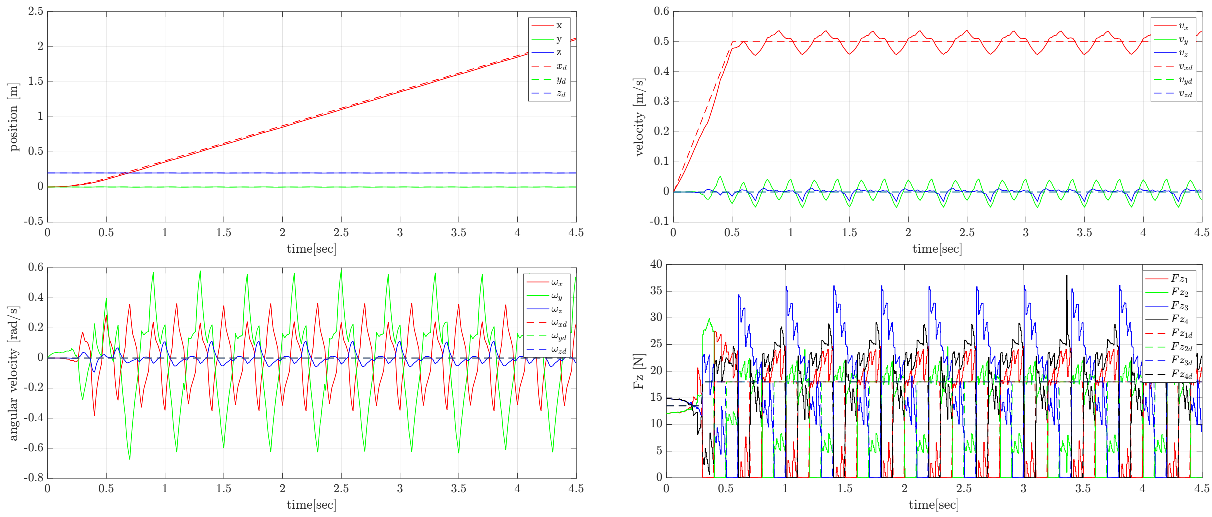
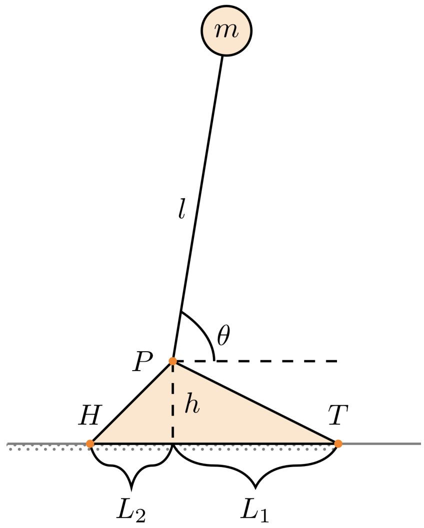


Figure 27: Crawl gait $\mu = 0.5$

The analysis of the simulations carried out shows that: increasing the speed reference can cause instability in the robot's movement in almost any gait, so it is preferable to adopt lower speeds in any gait; decreasing the mass tends to make the robot's gait more stable and to follow the speed reference more accurately and quickly; decreasing the coefficient of friction, on the other hand, leads the robot to a less regular behaviour causing instability due to the decrease in linear acceleration, so it is preferable to adopt specific materials for the robot's feet according to the composition of the surface on which it must move. In addition to these simulations, other analyses were conducted, whose results are not presented here but from which it can be seen that increasing the coefficient of friction can lead to an increase in tracking error for some gaits such as gallop, while others to be more accurate, particularly with a reduction in oscillations in angular velocities. All these simulations show that the best gaits in terms of stability and accuracy are the trot and crawl gait.
All the results of the undertaken simulations are available on GitHub: gaetanotorella/FSR_HW4

4. Consider a legged robot as in the picture below. The foot and the leg are assumed to be massless. The point T represents the toe, the point H represents the heel, and the point P is the ankle. The value of the angle θ is positive counterclockwise and it is zero when aligned to the flat floor. Answer to the following questions by providing a brief explanation for your them.



- Without an actuator at the point P, is the system stable at $\theta = \frac{\pi}{2} + \epsilon$?
  In this situation, where there's no actuator present at point P, the system is not stable and the leg stability becomes precarious as the it approaches the angle $\theta = \frac{\pi}{2} + \epsilon$. This instability stems from the fact that, at this specific configuration, the leg exhibits characteristics similar to those of an inverted pendulum when it's perpendicular to the ground ($\theta = \frac{\pi}{2}$). If one considers an inverted pendulum in perfectly vertical equilibrium, it's clear that it's in an inherently unstable state, since the slightest push in any direction will cause it to swing. Similarly, when the leg reaches $\theta = \frac{\pi}{2} + \epsilon$, any small external perturbation, such as a slight push or weight shift, can alter its delicate balance. Consequently, the system's tendency to deviate from this precarious equilibrium makes it highly susceptible to even the smallest perturbations, resulting in significant deviations from the desired configuration. Therefore, without an actuator at point P to provide corrective forces, the system lacks the necessary mechanisms to maintain stability under such conditions.
- Without an actuator at the point P (i.e., $\ddot{\theta} \neq 0$, $\dot{\theta} \neq 0$), compute the zero-moment point on the ground as a function of θ and the geometric and constant parameters (if necessary).

  The zero-momentum point (ZMP) is the point on the floor in which the ground reaction momentum (GRM) components around the x and y axes are null; it can be computed in general as follow:

$$ZMP = p_c^{x,y} - \frac{p_c^z}{\ddot{p}_c^z - g_0^z} \cdot (\ddot{p}_c^{x,y} - g_0^{x,y}) + \frac{1}{m \cdot (\ddot{p}_c^z - g_0^z)} \cdot S \cdot \dot{L}^{x,y} \qquad S = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \in SO(2) \quad (5)$$

  Where the first term represents the position of the CoM, the second is the acceleration of the CoM and finally the third term represents the angular momentum. But since the robot leg is represented only in two dimension and assuming that the plane in which the robot is illustrated is the xz plane, the robot leg can only move in x and z direction and rotate around the y-axis; for this reason, in the above expression the y coordinate is negligible and the equation can be rewritten as follow:

$$ZMP = p_c^x - \frac{p_c^z}{\ddot{p}_c^z - g_0^z} \cdot (\ddot{p}_c^x - g_0^x) - \frac{1}{m \cdot (\ddot{p}_c^z - g_0^z)} \cdot \dot{L}^y \qquad (6)$$

  Referring to the figure above, it can be imposed that:
  - $p_c^x = l \cdot \cos\theta$
  - $p_c^z = l \cdot \sin\theta + h$
  - $g_0^x = 0$ and $g_0^z = -g$
  - $L = \overrightarrow{l} \times m \cdot \overrightarrow{v} = m \cdot l^2 \cdot \dot{\theta}$

Once the expressions of p_c^x p_c^z and L are obtained, it's possible to compute their time derivative as follow:

- $\ddot{p}_c^x = l \cdot (-\ddot{\theta} \cdot \sin\theta - \dot{\theta}^2 \cdot \cos\theta)$
- $\ddot{p}_c^z = l \cdot (\ddot{\theta} \cdot \cos\theta - \dot{\theta}^2 \cdot \sin\theta)$
- $\dot{L} = m \cdot l^2 \cdot \ddot{\theta}$

In this way it's possible to compute the ZMP as follow:

$$ZMP = l\cos\theta - \frac{l\sin\theta + h}{l(\ddot{\theta}\cos\theta - \dot{\theta}^2\sin\theta) + g} \cdot l(-\ddot{\theta}\sin\theta - \dot{\theta}^2\cos\theta) - \frac{l^2 \cdot \ddot{\theta}}{l(\ddot{\theta}\cos\theta - \dot{\theta}^2\sin\theta) + g} \tag{7}$$

- Supposing to have an actuator at the ankle capable of perfectly cancelling the torque around P due to the gravity (i.e., $\ddot{\theta} \neq 0$, $\dot{\theta} \neq 0$), what value of θ can you achieve without falling?

In the case under consideration, the presence of an actuator at the P point of the ankle allows the cancellation of the momentum around the y-axis caused by gravity. In this situation, in order to balance the robot, the mass projected onto the ground must fall within the support polygon, which in this case is represented by a line whose expression is given by $L = L_1 + L_2$. Therefore, the following condition must be achieved in order to guarantee the stability of the robot:

$$-L_2 \leq l \cdot \cos\theta \leq L_1 \rightarrow -\frac{L_2}{l} \leq \cos\theta \leq \frac{L_1}{l} \tag{8}$$

Extracting θ from the above expression, shows that the robot is stable for the following values

$$arccos\left(\frac{L_1}{l}\right) \leq \theta \leq arccos\left(-\frac{L_2}{l}\right) \tag{9}$$