Field and Service Robotics Homework

Homework4

Submitted by

Nimrod Millenium Ndulue

P38000202

Professor Fabio Ruggiero

Exercise 1

Buoyancy, a fundamental principle in fluid mechanics, plays a pivotal role in the operation and efficiency of underwater robots. Understanding the buoyancy effect, its hydrostatic nature, and its distinction from dynamic forces is essential for designing and controlling underwater robots effectively. Buoyancy, is the upward force exerted by a fluid on an immersed object, is a consequence of the hydrostatic pressure difference between the top and bottom of the object. This pressure difference arises due to the weight of the fluid displaced by the object. Unlike dynamic effects,e.g added mass ,which are generated fluid flow and change over time, buoyancy is a hydrostatic effect, dependent solely on the fluid's density and the volume of the displaced fluid, as shown in the subsequent equation:

$$b = \rho\Delta \; ||g|| \tag{1}$$

where:

- $g = \begin{bmatrix} 0 & 0 & g \end{bmatrix}^T \in R^3$
- $g \in R$, the gravity acceleration
- $\rho \in R$,the density of the water
- $\Delta \in R$,the volume of the body

One crucial aspect of buoyancy is its application of force at the center of buoyancy r_b^b, which is typically different from the center of mass r_c^b of the immersed object. While gravity acts on the center of mass, buoyancy exerts its force at the center of buoyancy, where such force is expressed by the relation:

$$f_b^b = -R_b^T \begin{bmatrix} 0 \\ 0 \\ b \end{bmatrix} = -R_b^T \begin{bmatrix} 0 \\ 0 \\ \rho\Delta \end{bmatrix} \tag{2}$$

Notice that in such a relation the minus sign is present because as said before the buoyancy is an upward force, meaning that it pushes the body up, while gravity pushes it down. It is also important to notice that the displacement of forces creates moments that influence the stability and orientation of underwater robots. Understanding and managing these moments are vital for controlling the vehicle's motion and stability underwater. A possible solution to such a problem is obtained by guaranteeing that both centers are positioned on the same axis, so the torques generated will have a smaller amplitude. Furthermore, the wrench due to the buoyancy and gravity(another static effect in underwater robots) in the body-fixed frame is expressed:

$$g_{rb}^b = -\begin{bmatrix} f_g^b + f_b^b \\ S(r_c^b)f_g^b + S(r_c^b)f_b^b \end{bmatrix} \in R^6 \tag{3}$$

Finally, we can notice that the buoyancy effect has been neglected in the other types of robots studied during the course because the air density is much lower than the density of the moving mechanical system. In underwater robots such effect has to be considered since the density of water is comparable to the density of the robot, making the fluid's reaction force significant. It is important to notice that since the buoyancy force gives a non-null contribution even when the UUV is still such force is referred to as a persistent dynamic term since it acts on the body at each instant of time and does not depend on the movement of the robot.

Exercise 2

a)**False.**

The statement is incorrect because it refers to the added mass as a ”load,” implying it represents a weight or mass. In reality, the added mass is a reaction force exerted by the fluid surrounding the robot during its movements underwater. This reaction force is equal in magnitude and opposite in direction to the force exerted by the robot.

b)**True.**

In other types of robots studied during the course, this effect was neglected because the air density is much lower than the density of the moving mechanical system. However, in this case, it must be considered because the previous assumption no longer holds. Hence,the density of water is comparable to the density of the robot, making the fluid’s reaction force significant.

c)**True**

The damping effect, represented by the matrix D_{RB}, significantly enhances the stability of the underwater robot. As D_{RB} is a positive definite matrix, implying it’s eigenvalues are negative, it makes the system’s convergence to the desired state faster. Furthermore, in the context of Lyapunov-like stability analysis, the damping effect plays a crucial role. In equation (4) a Lyapunov function V is used to analyze the system’s stability. The presence of the damping term $s_v^T D_{RB} s_v$ in the derivative of the Lyapunov function $\dot{V}$(see equation(5)) contributes negatively, ensuring that $\dot{V}$ is negative semi-definite. This negative semi-definiteness is vital for applying Barbalat’s Lemma, which requires the Lyapunov function V to be bounded and $\dot{V}$ to tend to zero as time approaches infinity. The damping term thus facilitates proving that the system states converge to their desired values, ensuring asymptotic stability.

$$V = \frac{1}{2} s_v^T M_v s_v + \frac{1}{2} \tilde{\theta}_K^T K_\theta \tilde{\theta}_K + \frac{1}{2} k_p \eta_1^T \eta_1 + k_o \tilde{Z}^T \tilde{Z}, \tag{4}$$

$$\dot{V} = -s_v^T (K_D + D_{RB}) s_v - k_p \lambda_p \eta_1^T \eta_1 - k_o \lambda_o \epsilon^T \epsilon. \tag{5}$$

Note: The equations of V and $\dot{V}$,respectivly equations (4) and (5) where taken from the book *G.Antonelli,”Underwater robots”,Springer Tracts in Advanced Robotics,2014*.

d)**False.**

The statement is false because,while it is indeed accurate to view the ocean current as constant due to its classification as a disturbance(notion advantageous for control system design),it’s equally important to note that such disturbance can’t be considered constant when referenced with respect to the body frame. To consider the ocean current constant, it must be referenced in relation to the earth-fixed frame.

Exercise 3

In general, the dynamic model of legged robots represents them as single rigid bodies subject to forces at their contact points with the ground. The trajectory tracking of the desired center of mass (CoM) trajectory is achieved through the modulation of ground reaction forces, while maintaining balance. Consequently, legged robots can move solely through their contact with the environment (ground), and by exploiting this contact, it becomes possible to control the system. Specifically, legged robots are controlled using a whole-body control scheme, wherein the desired joint accelerations $\ddot{q}_j$ and the desired ground reaction forces f_{gr} are obtained as outputs from a "quadratic problem," which is a critical component of the control scheme. It is crucial to recognize that solving the quadratic optimization problem for a legged robot is a fundamental step in determining the control inputs that achieve the desired performance objectives while adhering to the robot's physical constraints, that ensure dynamic consistency, non-sliding contact, torque limits, and leg swing tasks. Additionally, it is important to highlight that the cost function is designed to track the reference trajectory of the robot's center of mass. Subsequently are reported the equations,on which such optimization is based:

$$\begin{aligned} \text{minimize} \quad & f(\zeta) \\ \text{subject to} \quad & A\zeta = b, \\ & D\zeta \leq c. \end{aligned}$$

The chosen vector of control variables is:

$$\zeta = \begin{bmatrix} \ddot{r}^T & \ddot{q}_j^T & f_{gr}^T \end{bmatrix}^T \in R^{n_b+n_j+3n_{st}}$$

where :

$$r_c = \begin{bmatrix} p_c^T & \eta_c^T \end{bmatrix}^T$$

Subsequently is reported the implementation of the quadratic function utilizing the QP solver:

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

The function call `qpSWIFT(sparse(H), g, sparse(Aeq), beq, sparse(Aineq), bineq)` solves the QP problem. The constraints of the problem are defined by the matrices Aeq and $Aineq$, and the vectors beq and $bineq$, which represent equality and inequality constraints, respectively. These matrices are passed in the sparse function to enhance computational efficiency. The output $zval$ contains the optimal solution to the QP problem, providing the desired control inputs that minimize the objective function while satisfying all constraints. In the context of quadruped robots control, we understand that the robot has the capability to move with various gaits, which change depending on the movement of its feet during locomotion. In our case study, trough the "Foot Scheduler," we can select different gaits:

- **Crawl gait**: This is the most stable gait, with three feet always in contact with the ground, forming a triangular support polygon. However, this increased stability comes at the cost of reduced speed.
- **Pacing gait**: In this gait, the lateral legs move in combination. It is more dynamic than the crawl gait but less dynamic than the bound and gallop gaits.
- **Trot gait**: This gait involves the movement of diagonal leg pairs and allows for faster locomotion. However, it sacrifices stability for speed, as it always forms a linear support polygon.
- **Trot-run gait**: Similar to the trot gait, but the legs in the swinging phase cover less distance. Unlike pure trotting, this gait does not have a stance phase with all four legs on the ground, with a brief moment where all legs are in the swinging phase.
- **Bound gait**: In this gait, the front and rear legs work in pairs, resulting in moments when no feet are in contact with the ground. This makes it one of the most dynamic gaits, along with the gallop.
- **Gallop gait**: This is the fastest gait, typically observed in animals like horses. It provides the highest velocity for a quadruped robot but has a small margin of stability, as there are moments with only one foot in contact with the ground.

Simulations:

Below are the simulations of the quadruped robot, where various physical parameters (mass and friction coefficient) and control parameters (desired speed v_d and prediction horizon) are altered across different gaits. This will enable us to determine the optimal values based on their performance.To begin with, in order to understand how the performance of different gaits changes with varying parameters, the initial simulations were conducted using the default parameters and observing the performance of the different gaits with these settings.

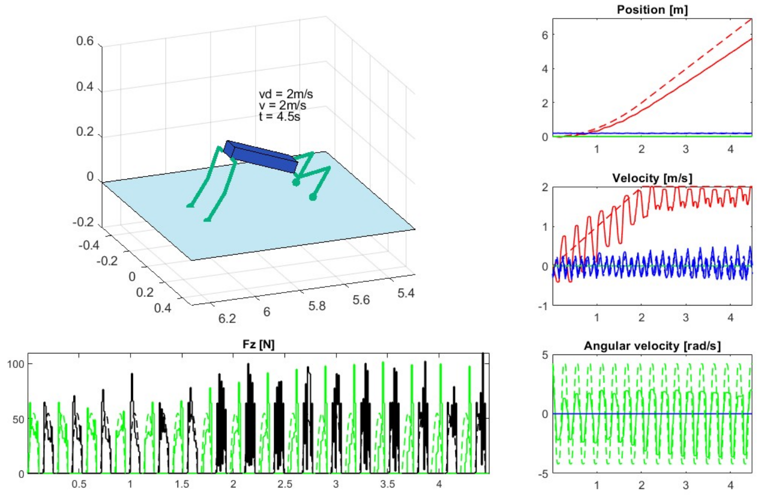

Figure 1: bound gait: $m = 5.5$kg,$\mu = 1$,$v_d = 0.5$

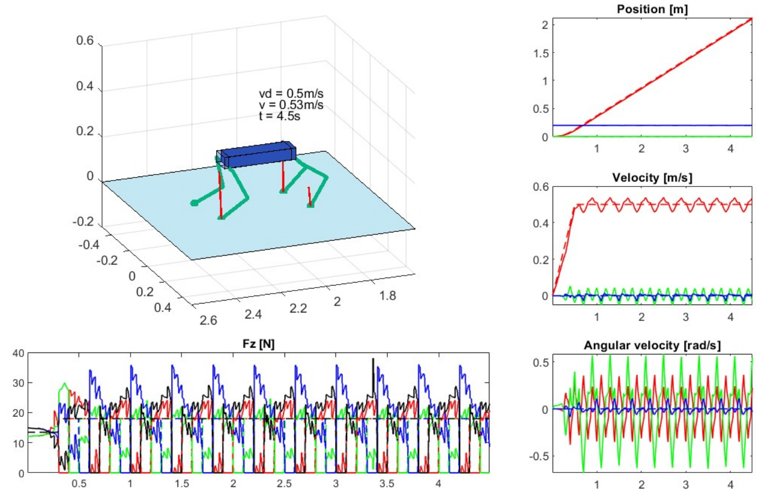

Figure 2: crawl gait: $m = 5.5$kg,$\mu = 1$,$v_d = 0.5$

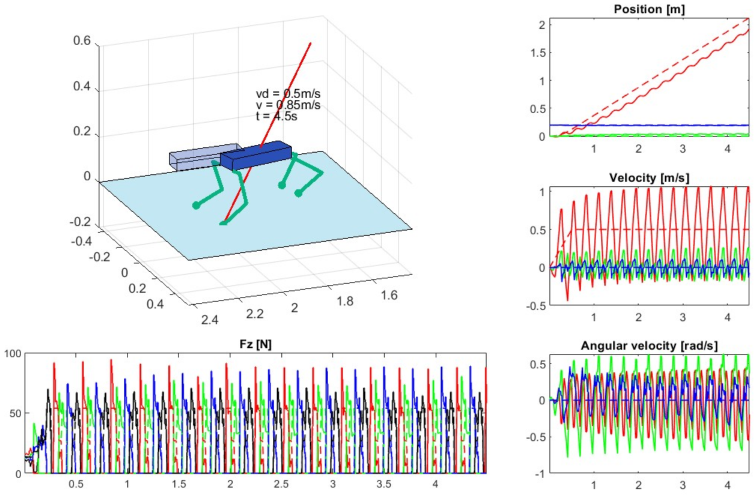

Figure 3: gallop gait: $m = 5.5$kg,$\mu = 1$,$v_d = 0.5$

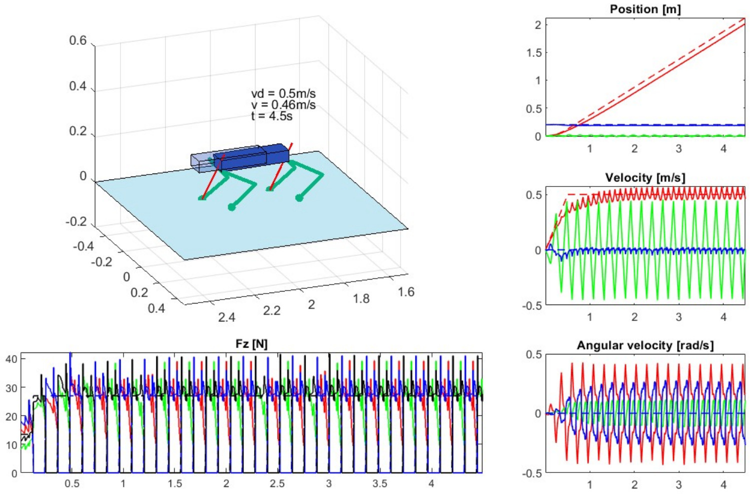

Figure 4: pacing gait: $m = 5.5$kg,$\mu = 1$,$v_d = 0.5$

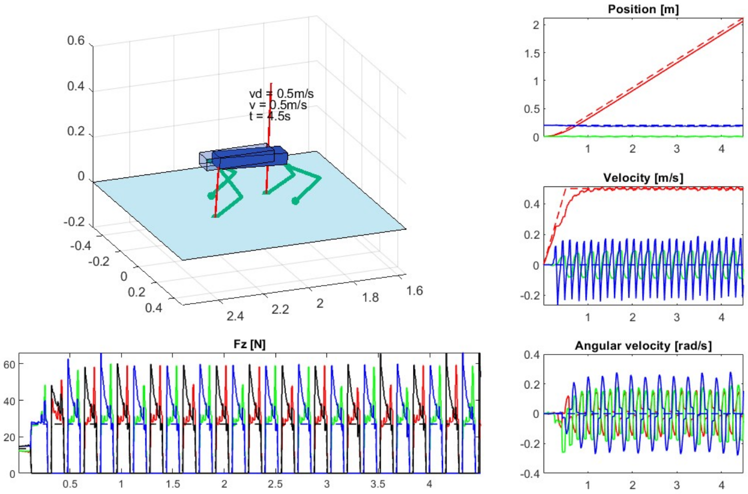

Figure 6: trot run gait: $m = 5.5$kg,$\mu = 1$,$v_d = 0.5$

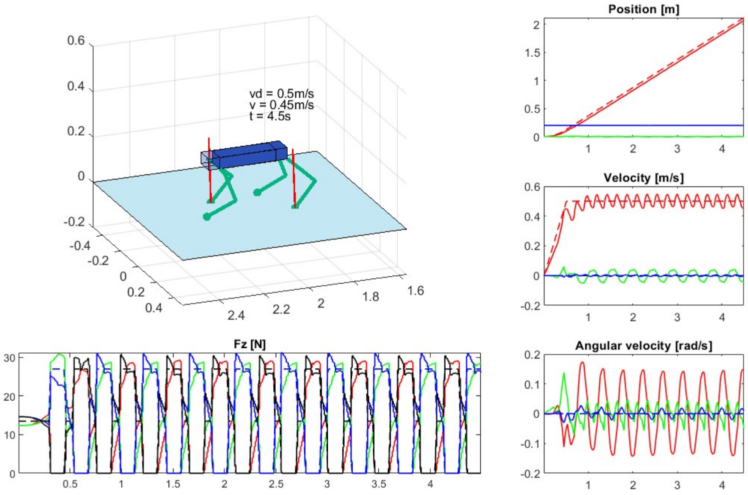

Figure 5: trot gait: $m = 5.5$kg,$\mu = 1$,$v_d = 0.5$

- **Crawl Gait**: Best for precise position tracking, minimal vertical ground reaction forces (40 N), and stable ground interaction.
- **Trot Gait**: Similar position tracking performance to crawl, with slightly lower peak F_z (30 N).
- **Trot-Run Gait**: Superior in velocity tracking, maintaining consistent speed with minimal deviations.
- **Gallop and Bound Gaits**: Highly dynamic, with significant speed oscillations and the highest peak forces (45 N for gallop, 100 N for bound).

In conclusion, for low-speed applications, the crawl gait is the most effective, followed by the trot gait. These gaits ensure precise position tracking and stable ground interaction, whereas the more dynamic gaits like gallop and bound are less suitable due to their significant oscillations and higher peak forces.

Furthermore are reported the simulation varying v_d, in order to understand how do the perfomances of the gait change.

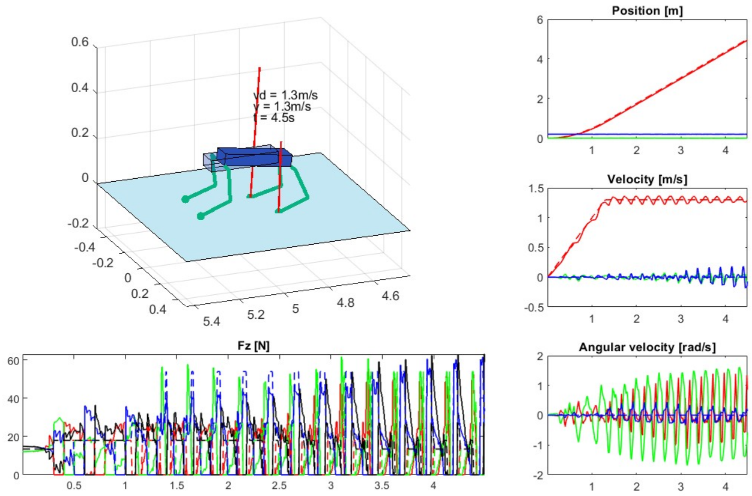

Figure 7: crawl gait: $m = 5.5$kg,$\mu = 1$,$v_d = 1.3$

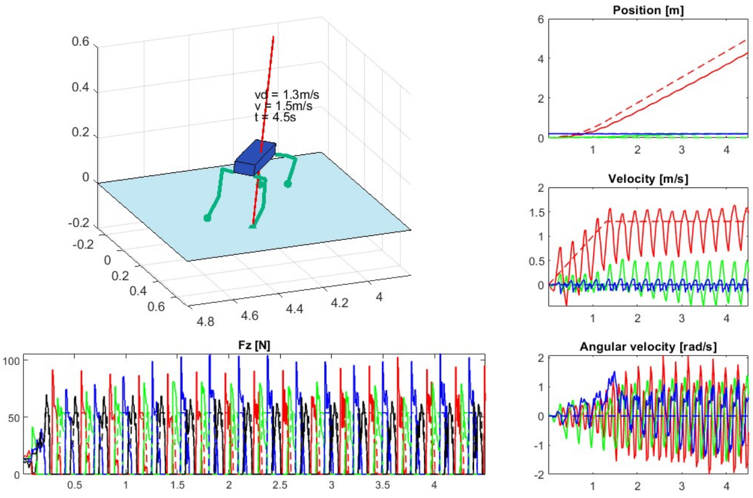

Figure 8: gallop gait: $m = 5.5$kg,$\mu = 1$,$v_d = 1.3$

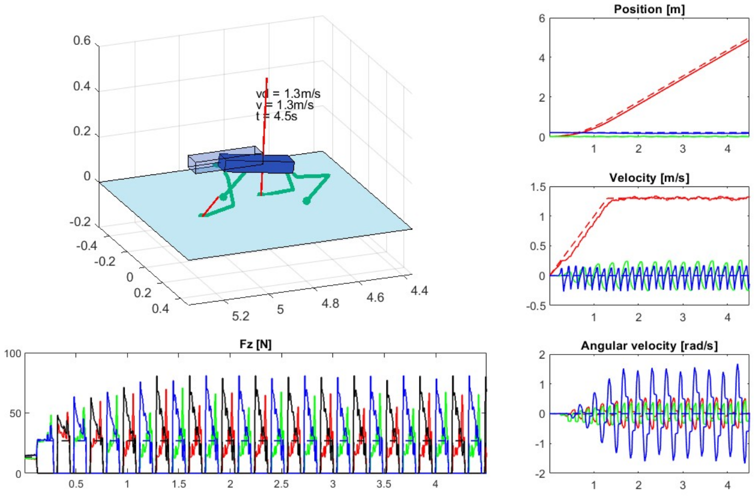

Figure 9: trot run gait: $m = 5.5$kg,$\mu = 1$,$v_d = 1.3$

- **Crawl Gait**: Best for precise position tracking at lower speeds, but at higher speeds (1.3 m/s), the gait sequence deviates, resulting in moments resembling gallop-like movements. The crawl gait is not suitable for higher velocities.
- **Trot Gait**: Maintains good velocity tracking at higher speeds but shows instances of potential leg-ground collisions. Position tracking performance diminish with respect to the case with $v_d = 0.5$.Note that the peak is also higher(F_z around 60N).
- **Trot-Run Gait**: Performs better than crawl at higher velocities (1.3 m/s), with peak F_z greater than 70 N. It also maintains consistent speed with minimal deviations but displays instances of potential leg-ground collisions.
- **Gallop Gait**: More suitable for higher velocities than crawl, with peak F_z greater than 100N. However, it becomes unstable with uncontrolled rotations around the z-axis and poor tracking performance overall.
- **Pacing Gait**: Shows good performance in velocity tracking and stability.Note that the performance still diminishes with respect to the performance with the default values.
- **Bound Gait**: bad velocity tracking, and like trot, displays instances of potential leg-ground collisions.

In conclusion, for low-speed applications, the crawl gait remains the most effective for precise position tracking and stable ground interaction. However, at higher velocities, trot-run and gallop gaits become more effective, despite their increased peak forces and potential stability issues. Crawl gait is not suitable for higher velocities due to deviations from its typical stance and instability. Moreover,decreasing the coefficient of friction (μ) increases the likelihood of the robot becoming unbalanced. Below are report the plots of the performance of different gaits under these conditions.

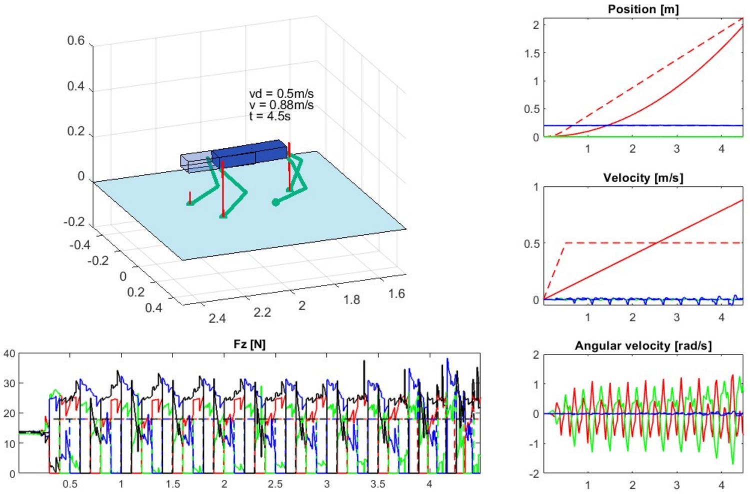

Figure 10: crawl gait: $m = 5.5$kg,$\mu = 0.02$,$v_d = 0.5$

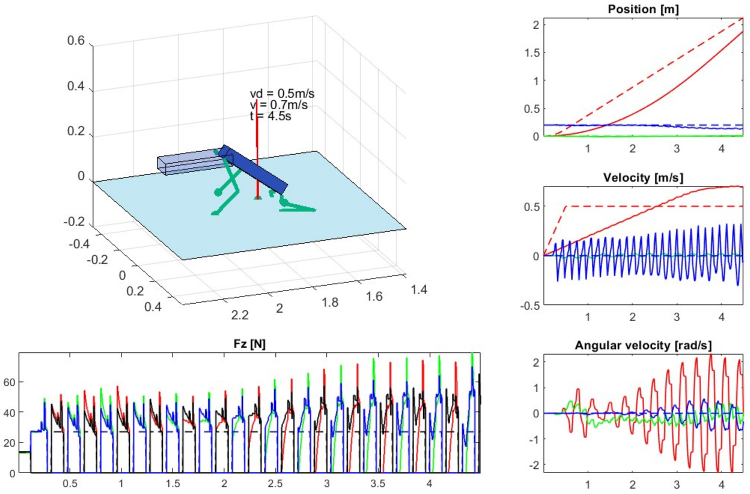

Figure 11: trot run gait: $m = 5.5$kg,$\mu = 0.02$,$v_d = 0.5$

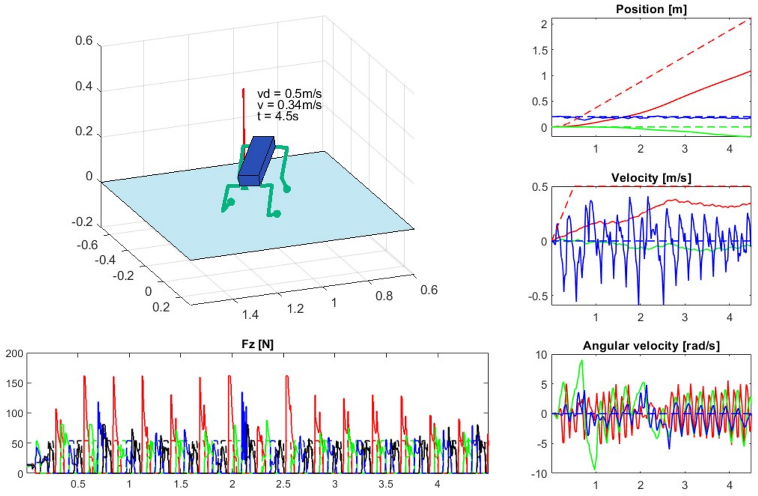

Figure 12: gallop gait: $m = 5.5$kg,$\mu = 0.02$,$v_d = 0.5$

- **Crawl Gait**: Best for stability and position tracking when friction is decreased. The ground reaction forces (F_z) peak at 45 N, demonstrating stable interaction with the ground.
- **Trot Gait**: Comparable to the crawl gait in terms of position tracking and ground reaction forces, with F_z also peaking at 45 N. Maintains good performance despite the reduced friction.
- **Bound Gait**: The robot moves with really low velocity in the heading direction, and present bad tracking position and velocity tracking. The quadruped moves slowly because the reduced friction causes it to slip, thereby preventing it from exploiting the ground in order to exert the necessary reactive force to facilitate rapid movement.
- **Pacing Gait**: Not stable under decreased friction, leading to the robot falling.
- **Trot run Gait**:The quadruped moves slowly because the reduced friction causes it to slip, thereby preventing it from exploiting the ground in order to exert the necessary reactive force to facilitate rapid movement.
- **Gallop Gait**: Similarly unstable, causing the robot to fall and encounter singularity during the simulation.

Subsequent simulations were conducted to assess the effects of augmenting and diminishing the robot's weight(mass) on gait performance.

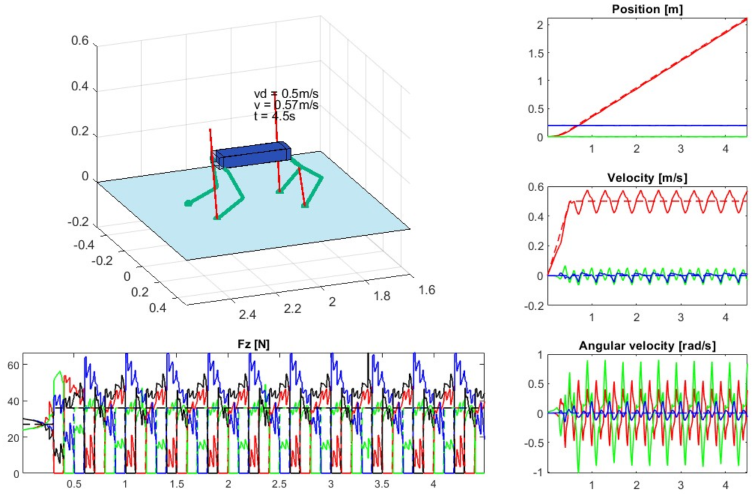

Figure 13: crawl gait: $m = 11$kg,$\mu = 1$,$v_d = 0.5$

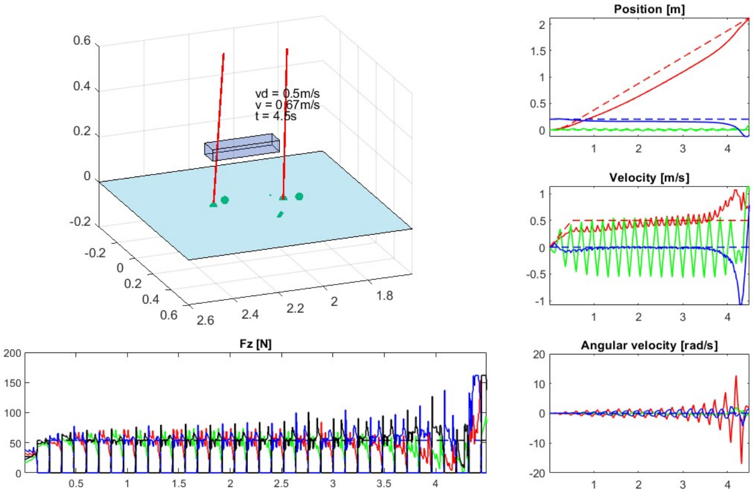

Figure 14: pacing gait: $m = 11$kg,$\mu = 1$,$v_d = 0.5$

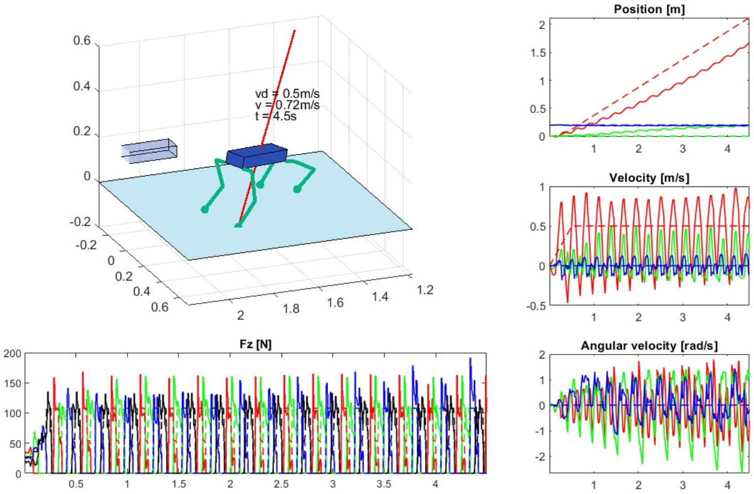

Figure 15: gallop gait: $m = 11\text{kg}$,$\mu = 1$,$v_d = 0.5$

When the robot's weight was doubled, crawl, trot, and trot-run gaits effectively supported the increased weight, demonstrating good tracking performance. The crawl gait remained the most stable, handling the added mass without significant issues. However, pacing gait struggled, resulting in ground collisions. Bound and gallop gaits became increasingly unstable, exhibiting unpredictable behavior due to the higher dynamic forces involved.

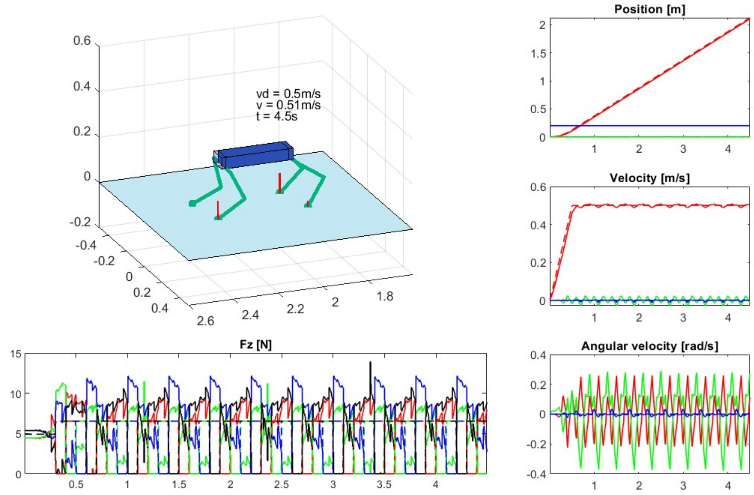

Figure 16: crawl gait: $m = 2\text{kg}$,$\mu = 1$,$v_d = 0.5$

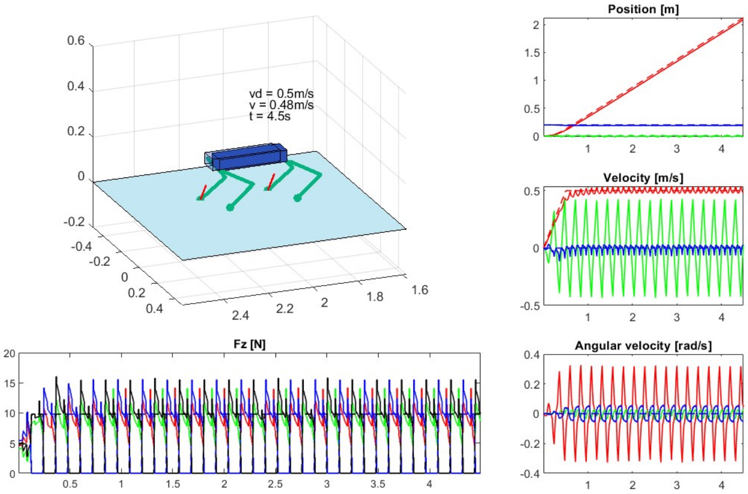

Figure 17: pacing gait: $m = 2$kg,$\mu = 1$,$v_d = 0.5$

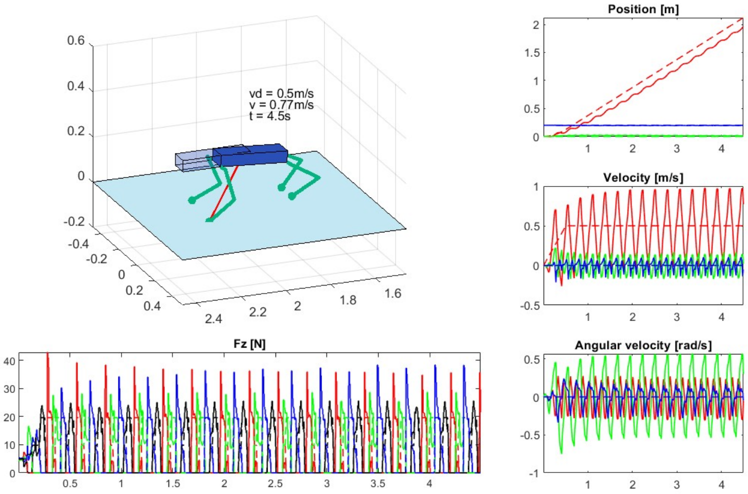

Figure 18: gallop gait: $m = 2$kg,$\mu = 1$,$v_d = 0.5$

Conversely, when the robot's weight was diminished, the crawl gait exhibited near-perfect behavior, maintaining excellent stability and control. Trot, trot-run, and pacing gaits also showed improved performance, benefiting from the reduced weight, which made control easier. The reduced mass helped mitigate oscillations and enhanced the robot's overall responsiveness. Other gaits, such as bound and gallop, showed no significant differences from standard simulations.

Exercise 4

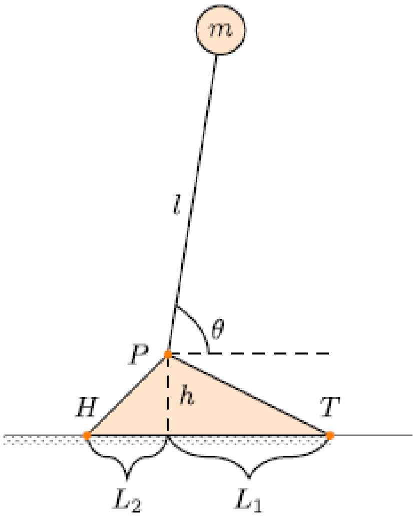

Figure 19:

a)At the point $\theta = \frac{\pi}{2}$ it possible to notice that if the system lack an actuator at the point P it won't be able to counteract the torque created by the gravitational force, in such scenario the system is equivalent to an inverted pendulum. Trough the stability analysis of the latter, we know that there is an unstable equilibrium point at $\theta = \frac{\pi}{2}$. Furthermore, applying a small displacement defined by $+\epsilon$ at the equilibrium point, thus $\theta = \frac{\pi}{2} + \epsilon$ will lead the system to get far from the equilibrium point since it is **unstable**.

b)Referring to Figure (19), we can determine the ZMP, which in this scenario is represented solely by an x-coordinate, given the support polygon is a segment.To ensure that the discussion on the calculation of the "Zero Moment Point" is as comprehensible as possible, the equation describing it is presented:

$$p_z^x = p_c^x - \frac{p_c^z}{(\ddot{p}_c^z - g_0)}(\ddot{p}_c^x - g_0) - \frac{1}{m(\ddot{p}_c^z - g_0)}(\ddot{p}zc - gz0)\dot{L}_y \tag{6}$$

We can notice that to obtain p_z^x it is first necessary to compute p_{xc}, p_{zc}, $\ddot{p}xc$, $\ddot{p}zc$ and $\dot{L}_y$. Referring to Figure (19) we are able to obtain p_{xc}, p_{zc}, and computing twice the derivative we obtain $\ddot{p}xc$, and $\ddot{p}zc$.We also have to take into account that in this scenario, the system is characterized by a single mass, m. Thus, the Center of Mass (CoM) of the robotic system in the z-x plane is determined directly by the coordinates p_c^x and p_c^z, which denote the position of the mass within the plane. Below are reported the results of the computation:

$$p_c^x = l\cos(\theta) \quad \ddot{p}_c^x = -l\sin(\theta)\ddot{\theta} - l\cos(\theta)\dot{\theta}^2 \tag{7}$$

$$p_c^z = h + l\sin(\theta) \quad \ddot{p}_c^z = -l\sin(\theta)\dot{\theta}^2 + l\cos(\theta)\ddot{\theta} \tag{8}$$

$$g_0 = -g \tag{9}$$

$$\dot{L}_y = ml^2\ddot{\theta} \tag{10}$$

Substituing the results obtained in the equation(6), we obtain the **Zero Moment Point**:

$$p_z^x = l\cos(\theta) - \frac{h + lsin(\theta)}{l\ddot{\theta}cos(\theta) - l\dot{\theta}^2sin(\theta) + g}(-l\ddot{\theta}sin(\theta) - l\dot{\theta}^2cos(\theta)) - \frac{l^2\ddot{\theta}}{l\ddot{\theta}cos(\theta) - l\dot{\theta}^2sin(\theta) + g} \tag{11}$$

c)Considering an ankle actuator that can perfectly nullify the torque around point P caused by gravity, resulting in $\dot{\theta} = 0$ and $\ddot{\theta} = 0$. In this scenario, the ground is a one-dimensional space. Using the equation $p_{xc} = l\cos(\theta)$, we can represent the projection of the center of mass (CoM) on the ground. It is crucial to note that, in this setup, the support polygon is simply the segment HT. By examining the frame configuration shown in Figure (19), we can geometrically determine the range of θ values that the robot can sustain without toppling over. The relevant equations are as follows:

$$-L_2 \leq l\cos(\theta) \leq L_1 \tag{12}$$

$$\begin{cases} \theta \geq \arccos\left(\frac{L_1}{l}\right) \\ \theta \leq \arccos\left(\frac{-L_2}{l}\right) \end{cases} \tag{13}$$

If the conditions specified in equation (13) are not met, the projection of the CoM will fall outside the support segment HT, resulting in the robot losing its balance and falling.