

UNIVERSITÀ DEGLI STUDI DI
NAPOLI FEDERICO II

Laurea magistrale in ingegneria dell'automazione e robotica

FIELD AND SERVICE ROBOTICS

HOMEWORK 4

A.Y. 2023/24

Professor: Fabio Ruggiero

Salvatore Del Peschio

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

Buoyancy is a hydrostatic effect that occurs when a rigid body is submerged in a fluid, it significantly influences the dynamics and control of underwater vehicles. The buoyancy force is an upward force acting at the center of buoyancy. The force is calculated as

$$\mathbf{f}_b^b = -R_b^T \begin{bmatrix} 0 \\ 0 \\ \rho\Delta g \end{bmatrix}$$

with ρ being the density of the fluid and Δ the volume of the displaced fluid. In underwater robotics, the buoyancy force is critical because it helps counteract the weight of the underwater vehicle. The effect of gravitational and buoyancy forces determines whether the vehicle floats, sinks, or hovers at a certain depth. In aerial robotics, buoyancy is typically neglected because the air has a much lower density compared to water. As a result, the buoyancy force has negligible impact on the dynamics of aerial vehicles. In addition, the primary forces in aerial robotics are gravity and the thrust generated by the propellers, these forces are much more significant than any other force that might act on the aerial vehicle.

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

- a) The added mass effect considers an additional load to the structure.
- b) The added mass effect is considered in underwater robotics since the density of the underwater robot is comparable to the density of the water.
- c) The damping effect helps in the stability analysis.
- d) The Ocean current is usually considered as constant, and it is better to refer it with respect to the body frame.
 - a) **False:** The effect does not refer to an additional load to the structure. Instead, it accounts for the inertia of the fluid surrounding the body that is accelerated along with it.
 - b) **True:** In underwater robotics, the effect is significant because the density of water is comparable to the density of the system. This effect must be considered because the fluid surrounding the robot moves with it, contributing to the overall inertia.
 - c) **True:** The damping effect plays a crucial role in the stability analysis of underwater vehicles. This effect includes dissipative drag and lift forces due to the fluid's viscosity, which are modeled using quadratic damping terms grouped into the matrix \mathbf{D}_{RB} . We can analyze the time derivative of the Lyapunov candidate function, that according to the book (*Underwater Robots*, G. Antonelli, p.72) is:

$$\dot{V} = -\mathbf{s}_v^T [\mathbf{K}_D + \mathbf{D}_{RB}] \mathbf{s}_v$$

where:

- \mathbf{K}_D is a positive definite control gain matrix.
- \mathbf{D}_{RB} is the positive definite damping matrix.

Therefore, the matrix D_{RB} ensures that the system's energy is dissipated over time. Energy dissipation is essential for stability as it ensures that unwanted oscillations and movements of the vehicle are reduced. The Lyapunov candidate function $V > 0$, its time derivative is always negative or zero, this satisfies the Lyapunov criterion for asymptotic stability.

- d) **False:** While the ocean current is usually considered constant, it is better to be referred respect to the world frame, not the body frame. With this approach is considered constant and irrotational in the fixed world frame, which simplifies the dynamic modeling and control of the underwater vehicle.

$$v_c = \begin{bmatrix} v_{c,x} \\ v_{c,y} \\ v_{c,z} \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \dot{v}_c = 0_6$$

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:
 - a) 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*);
 - b) 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.

Whole-body control schemes offer a comprehensive approach to motion planning and control. These schemes aim to achieve desired accelerations ($\ddot{q}f^*$) and ground reaction forces (fgr^*) by solving quadratic optimization problems, effectively decoupling motion planning from control tasks.

$$\begin{aligned} \min_{\zeta} \quad & f(\zeta) \\ \text{s.t.} \quad & A\zeta = b, \\ & D\zeta \leq c. \end{aligned}$$

The quadratic optimization problem involves minimizing an objective function while satisfying the constraints that ensure dynamic consistency, non-sliding contact, torque limits, and leg swing tasks. In the control algorithm we can choose between six gaits for our robot, each one involves a different coordination of leg movements, which is regulated by a foot scheduler providing a sequence of feet in contact with the ground and those in swing phase:

- **Trot Gait:** Use diagonal leg pair movement for faster locomotion but sacrifices stability due to a linear support polygon.
- **Bound Gait:** Involves coordinated movement of front and rear leg pairs, offering dynamic motion with intermittent ground contact.
- **Pacing Gait:** Lateral legs move in combination, providing dynamic motion but less so than gallop and bound gaits.
- **Gallop Gait:** The fastest gait with single-foot contact, is also the less stable due to the single contact with the ground.
- **Trot-Run Gait:** Similar to trot gait but with reduced leg swing, offering stability with brief moments of all legs in swing phase.
- **Crawl Gait:** Maintains three feet in constant ground contact, ensuring stability with a triangular support polygon but sacrificing speed.

All simulations were initially conducted without altering any parameters ($v_d = 0.5\text{m/s}$, $m = 5.5\text{kg}$ and $\mu = 1$). Among the various gaits tested, crawl, trot, and trot-run emerged as the top performers. The crawl gait demonstrated the most precise position tracking, effectively following the reference trajectory. Trot-run, on the other hand, excelled in velocity tracking, maintaining consistent speed with minimal deviations.

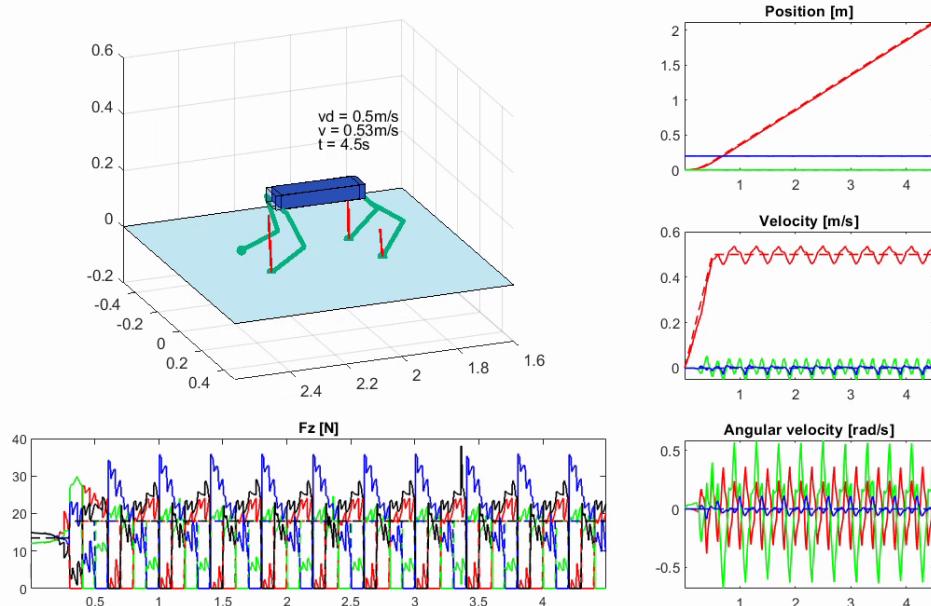
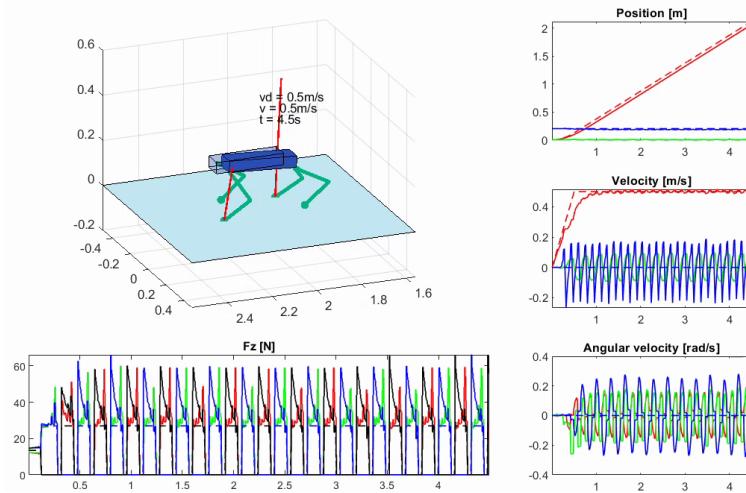
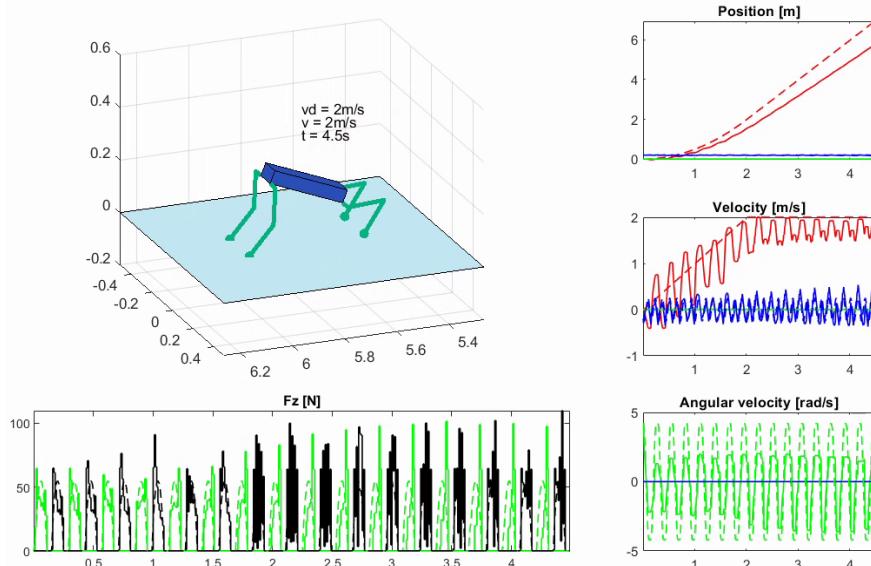


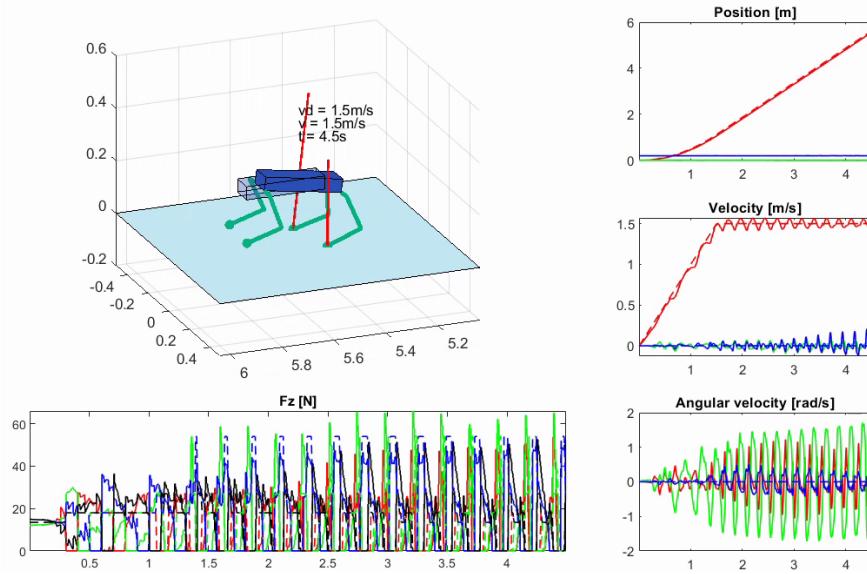
Figure 1: Gait: Crawl, $v_d = 0.5\text{m/s}$, $m = 5.5\text{kg}$, $\mu = 1$

Figure 2: Gait: Trot-run, $v_d = 0.5\text{m/s}$, $m = 5.5\text{kg}$, $\mu = 1$

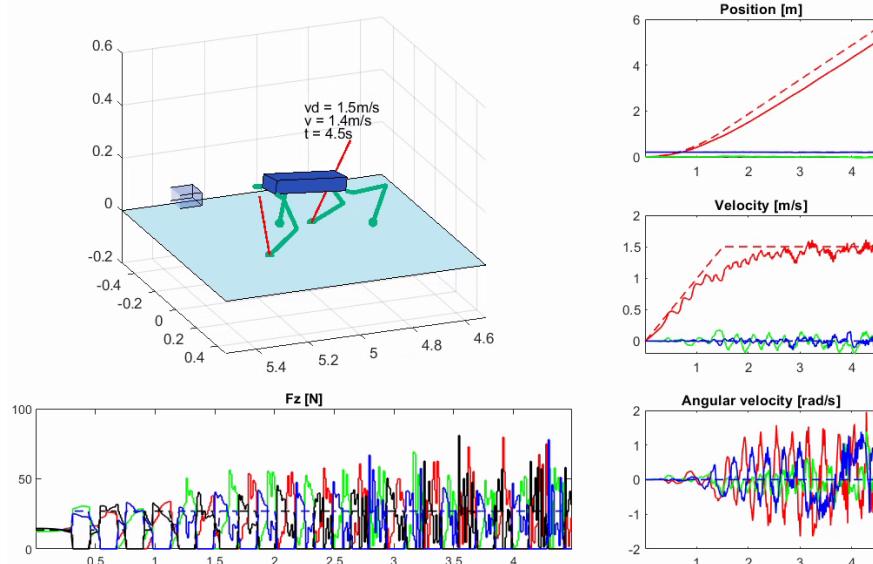
Bound and gallop gaits, being more dynamic in nature, exhibited significant oscillations during low-speed tracking. Both crawl and trot gaits demonstrated the lowest vertical ground reaction forces F_z , indicating a stable interaction with the ground.

Figure 3: Gait: Bound, $v_d = 0.5\text{m/s}$, $m = 5.5\text{kg}$, $\mu = 1$

To explore the effects of increased speed, additional simulations were conducted by incrementing the velocity. Under these conditions, crawl gait remained the most stable, though it began to deviate from its typical tripod stance, occasionally resembling gallop-like movements where only one or two legs were in contact with the ground.

Figure 4: Gait: Crawl, $v_d = 1.5\text{m/s}$, $m = 5.5\text{kg}$, $\mu = 1$

Trot and bound gaits maintained good velocity tracking; however, they displayed instances where leg-ground collisions could potentially occur.

Figure 5: Gait: Trot, $v_d = 1.5\text{m/s}$, $m = 5.5\text{kg}$, $\mu = 1$

Pacing gait showed decent performance, balancing speed and stability, while gallop gait became unstable, with uncontrolled rotations around the z-axis and poor tracking performance overall.

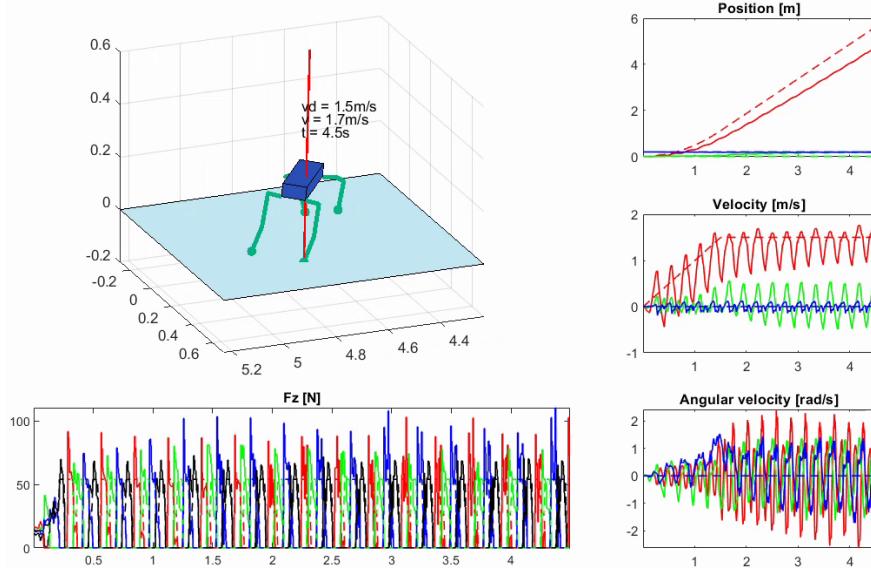


Figure 6: Gait: Gallop, $v_d = 1.5\text{m/s}$, $m = 5.5\text{kg}$, $\mu = 1$

Subsequent simulations were performed by doubling the robot's weight. In these scenarios, crawl, trot, and trot-run gaits effectively supported the increased weight, demonstrating good tracking performance. The crawl gait, in particular, remained the most stable, handling the added mass without significant issues.

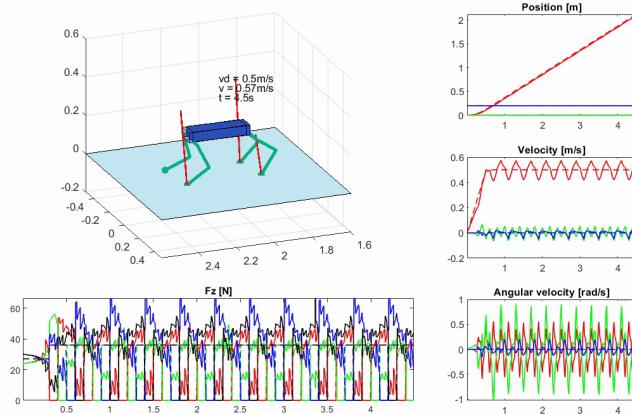


Figure 7: Gait: Crawl, $v_d = 0.5\text{m/s}$, $m = 11\text{kg}$, $\mu = 1$

Pacing gait, however, struggled under the increased weight, resulting in ground collisions.

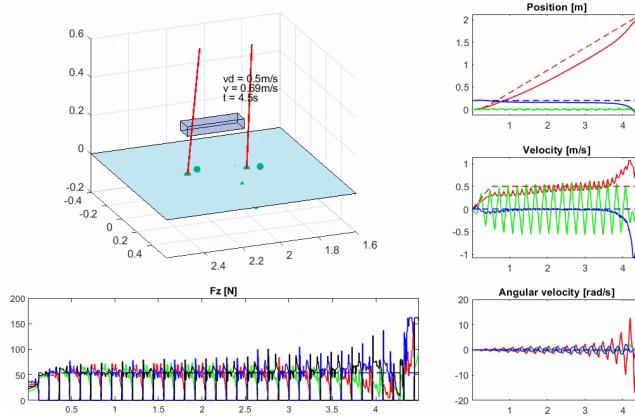


Figure 8: Gait: Pacing, $v_d = 0.5\text{m/s}$, $m = 11\text{kg}$, $\mu = 1$

Bound and gallop gaits also became increasingly unstable, exhibiting unpredictable behavior due to the higher dynamic forces involved.

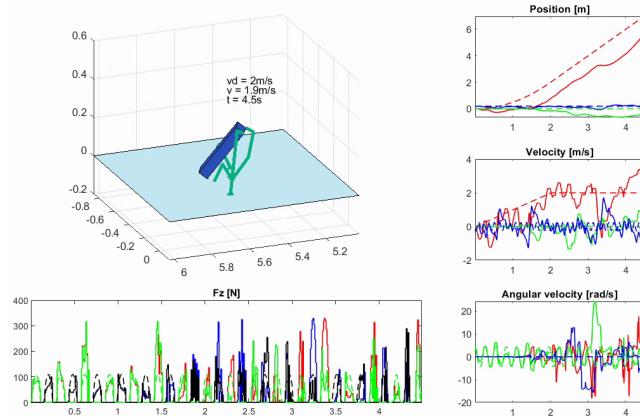


Figure 9: Gait: Bound, $v_d = 0.5\text{m/s}$, $m = 11\text{kg}$, $\mu = 1$

Conversely, simulations with halved weight were conducted to assess the effects of reduced mass. In these cases, 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.

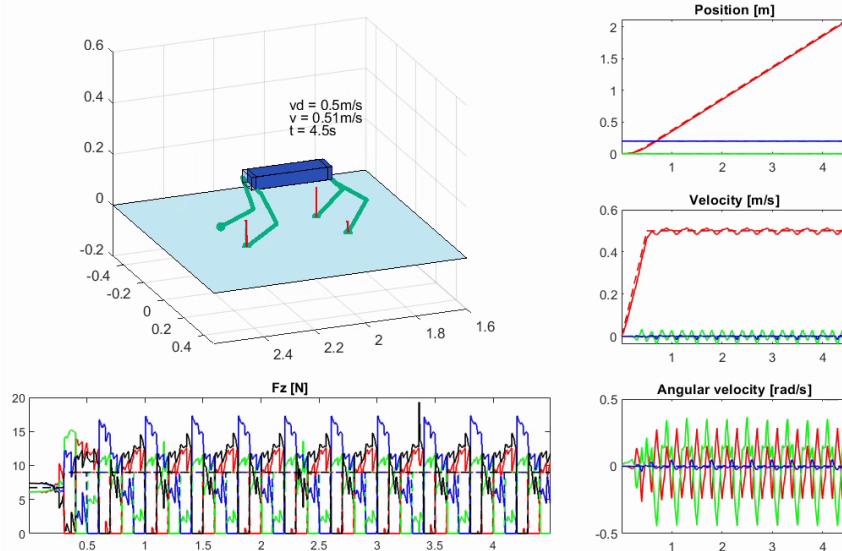


Figure 10: Gait: Crawl, $v_d = 0.5\text{m/s}$, $m = 2.75\text{kg}$, $\mu = 1$

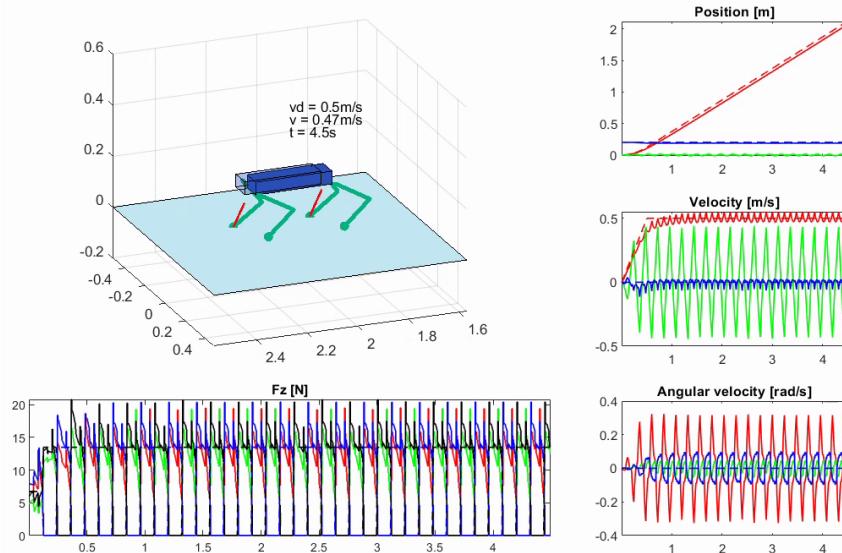
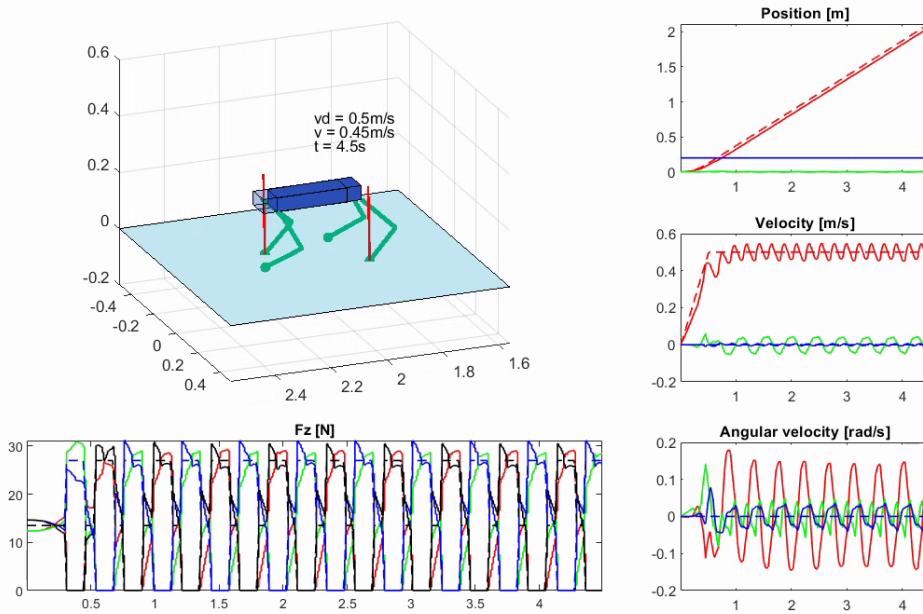
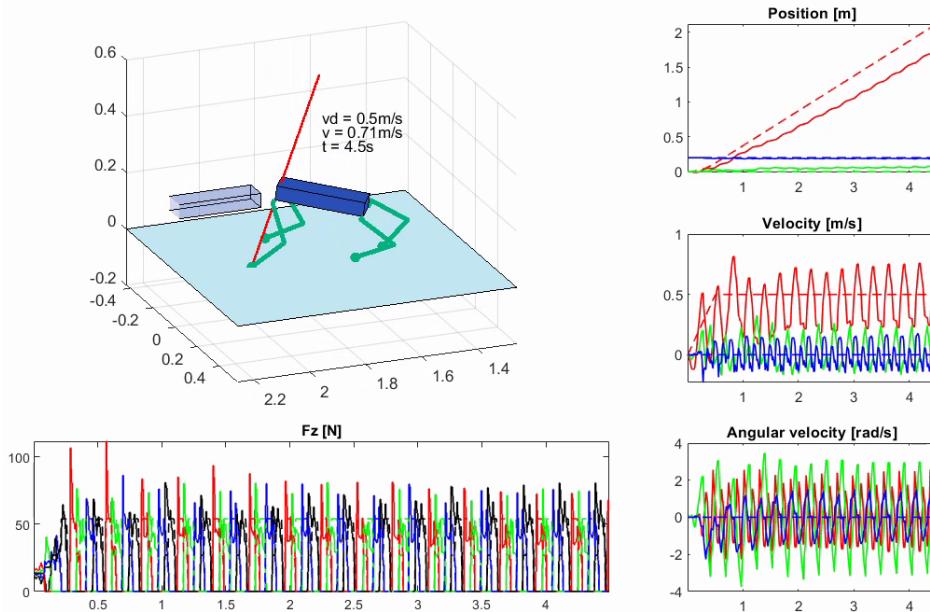


Figure 11: Gait: Pacing, $v_d = 0.5\text{m/s}$, $m = 2.75\text{kg}$, $\mu = 1$

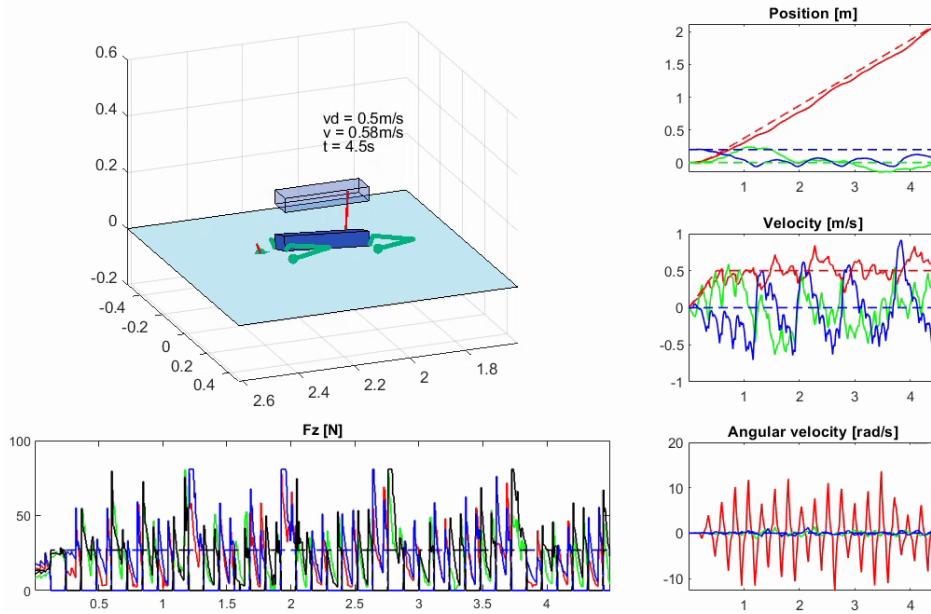
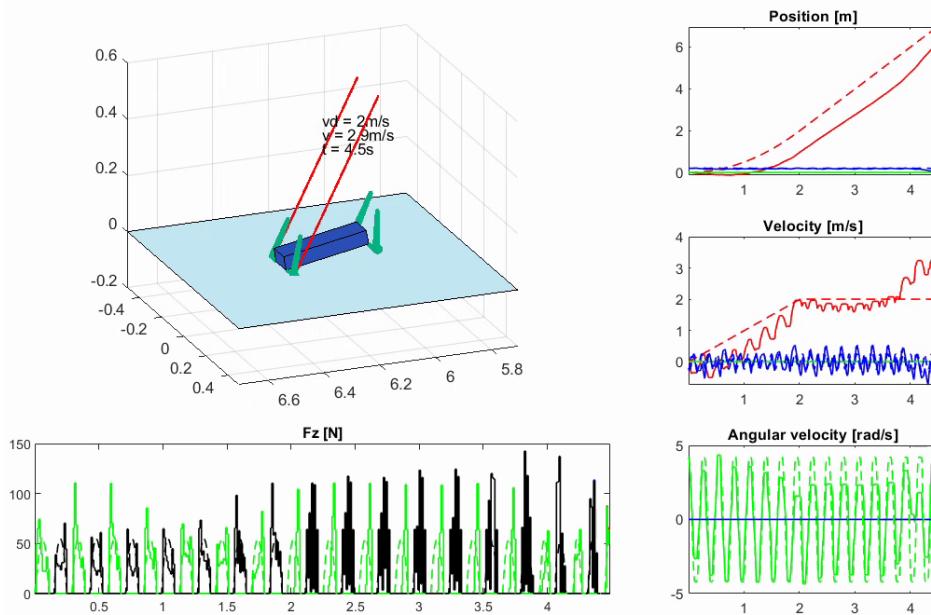
Other gaits, such as bound and gallop, showed no significant differences from standard simulations. Finally, simulations were conducted on a slippery terrain to simulate low-friction conditions ($\mu = 0.5$), akin to moving on ice. Under these conditions, stable gaits like crawl, trot, and trot-run maintained their performance, showing resilience against the reduced friction.

Figure 12: Gait: Trot, $v_d = 0.5\text{m/s}$, $m = 5.5\text{kg}$, $\mu = 0.5$

However, dynamic gaits like gallop, pacing, and bound struggled to maintain stability. The gallop gait, in particular, exhibited poor performance with frequent slips and uncontrolled movements.

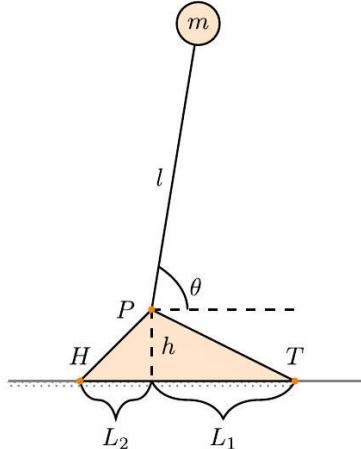
Figure 13: Gait: Gallop, $v_d = 0.5\text{m/s}$, $m = 5.5\text{kg}$, $\mu = 0.5$

Pacing and bound gaits became completely unstable, unable to cope with the slippery surface, leading to frequent falls and erratic behavior.

Figure 14: Gait: Pacing, $v_d = 0.5\text{m/s}$, $m = 5.5\text{kg}$, $\mu = 0.5$ Figure 15: Gait: Bound, $v_d = 0.5\text{m/s}$, $m = 5.5\text{kg}$, $\mu = 0.5$

Overall, these additional simulations, each altering one parameter at a time, provide a comprehensive understanding of how different gaits perform under varying conditions. The results highlight the importance of gait selection and parameter tuning in achieving optimal stability and control for quadruped robots in diverse environments.

4. Consider a legged robot as in the picture below. The foot and the leg are assumed to be mass-less. 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 $\theta = \frac{\pi}{2} + \epsilon$?
 - 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).
 - Supposing to have an actuator at the ankle capable of perfectly cancelling the torque around P due to the gravity (i.e., $\ddot{\theta} = 0, \dot{\theta} = 0$), what value of θ can you achieve without falling?



- Without an actuator at point P , the system is not stable at $\theta = \frac{\pi}{2} + \epsilon$. This system is similar to the inverted pendulum, where the equilibrium point $\theta = \frac{\pi}{2}$ is unstable. A small angular displacement ϵ from this point will cause the system to move away from the equilibrium point.
- The zero-moment point (ZMP) is defined as the point on the ground where the sum of the horizontal momenta of the ground reaction forces is zero. Considering the robot model and the geometric configuration, given that in our legged robot the support polygon is a segment, the ZMP x coordinate can be computed as follows:

$$p_z^x = p_c^{x,y} - \frac{\dot{p}_c^z}{\ddot{p}_c^z - g_0^z} (\ddot{p}_c^x - g_0^x) - \frac{\dot{L}^y}{m(\ddot{p}_c^z - g_0^z)}$$

$$p_z^x = l \cos(\theta) - \frac{h + l \sin(\theta)}{-l(\sin(\theta)\dot{\theta}^2 - \cos(\theta)\ddot{\theta}) + g} \left(-l(\cos(\theta)\dot{\theta}^2 + \sin(\theta)\ddot{\theta}) \right) - \frac{ml^2\ddot{\theta}}{m(-l(\sin(\theta)\dot{\theta}^2 - \cos(\theta)\ddot{\theta}) + g)}$$

where p_z^x is the horizontal distance from the projection of P on the ground.

- c) Suppose we have an actuator at the ankle capable of perfectly cancelling the gravitational torque around P (i.e., $\ddot{\theta} = 0, \dot{\theta} = 0$). We can determine the value of θ that allows the robot to remain stable without falling. This occurs when the projection of the center of mass (CoM) $p_c^x = l \cos(\theta)$ lies within the support segment defined by points H and T . It can be expressed as:

$$\begin{aligned} -L_2 &\leq l \cos(\theta) \leq L_1 \\ \theta &\in \left[\arccos\left(\frac{L_1}{l}\right), \arccos\left(-\frac{L_2}{l}\right) \right] \end{aligned}$$

The robot can remain stable without falling for values of θ that satisfy this condition.