FIELD AND SERVICE ROBOTICS (FSR) – a.y. 2023/2024

Homework N.2

Vincenzo Palomba
P38000180

University of Naples Federico II
Department of Electrical Engineering and Information Technology

Abstract

The report summarizes the completion of the fourth homework assignment, consisting of four exercises on legged and underwater robotics. Topics covered include the effects of hydrodynamics and hydrostatics, the analysis of a quadruped controller simulation, and an in-depth focus on a single-leg example in the final part.

Exercise 1

Buoyancy is an essential concept in fluid mechanics, particularly for the design and control of underwater robots. It refers to the upward force exerted by a fluid on a submerged object, resulting from the hydrostatic pressure difference between the object's top and bottom surfaces. This pressure difference arises from the weight of the fluid displaced by the object. Unlike dynamic effects, such as added mass, which result from fluid flow and vary over time, buoyancy is a hydrostatic effect that depends only on the fluid's density and the volume of the displaced fluid, as illustrated in the following equation:

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

With:

- b represents the buoyant force
- ρ stands for the fluid's density
- Δ denotes the volume of the displaced fluid
- $\overline{g} = [0 \quad 0 \quad g]^T$ is the acceleration vector due to gravity

One crucial aspect of buoyancy is its application of force at the center of buoyancy $\mathbf{r_b^b}$, which is typically different from the center of mass $\mathbf{r_c^b}$ of the submerged object. While gravity acts on the center of mass, buoyancy exerts its force at the center of buoyancy. This forces can be expressed by the following relations:

$$\mathbf{f_g^b} = R_b^T \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} \quad \mathbf{f_b^b} = -R_b^T \begin{bmatrix} 0 \\ 0 \\ b \end{bmatrix} = -R_b^T \begin{bmatrix} 0 \\ 0 \\ \rho \Delta g \end{bmatrix} \tag{2}$$

The minus sign in the second relation indicates that buoyancy is an upward force, opposing gravity's downward force. This displacement creates moments affecting the stability and orientation of underwater robots. Proper control of these moments is crucial for maintaining the vehicle's motion and stability. One solution is

to align the centers of buoyancy and gravity along the same axis, reducing torque amplitudes. Additionally, the wrench resulting from buoyancy and gravity (another static effect in underwater robots) in the body-fixed frame can be expressed as follows:

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

Finally, it is important to observe that the buoyancy effect has been disregarded in other types of robots studied during the course because the air density is significantly lower than the density of the mechanical systems in motion. In fact while the air density is 830 times less then the water density, and assuming the robot is made of the same material in both cases,the buoyancy force for aerial robots is negligible (about 830 times less). So, if the robot is sufficiently heavier than the volume of the displaced fluid the effect can be neglected, meaning that $g_{rb}^b \approx \begin{bmatrix} f_g^b \\ S(r_c^b)f_g^b \end{bmatrix}$. It's important to note that the buoyancy force acts as a persistent static term, continuously affecting the underwater vehicle even when stationary, independent of the robot's movement.

Exercise 2

1. **False**

   It is incorrect referring to the added mass as simply an additional weight added to the robot. In fact, the **added mass** is a reaction force exerted by the fluid surrounding the robot during its underwater movements. This reaction force is equal in magnitude and opposite in direction to the force exerted by the robot. This effect is hydrodynamic and is divided into two components, represented by $\mathbf{M_A}$ and $\mathbf{C_A}$, which are not necessarily positive definite.

2. **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 underwater robots, this assumption is invalid. The density of water is comparable to the density of the robot, making the fluid's reaction force significant and thus necessitating its consideration.

3. **True**

   The **damping effect**, represented by the matrix $\mathbf{D_{RB}}$, enhances the stability of the underwater robot. Since D_{RB} is a positive definite matrix, meaning its eigenvalues are positive, it accelerates the system's convergence to the desired state.

   The viscosity of the fluid causes the presence of dissipative drag and lift forces on the body, and, to simplify things, it is helpful consider only quadratic damping terms. In equation (4), a Lyapunov function V is used to examine the system's stability. The inclusion 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 is essential for applying Barbalat's Lemma, which requires the Lyapunov function V to be bounded and $\dot{V}$ to approach zero as time goes to infinity. Consequently, the damping term helps demonstrate that the system states converge to their desired values, ensuring asymptotic stability.

$$V = \frac{1}{2}s_v^T M_v s_v + \frac{1}{2}\tilde{\theta}_v K_\theta \tilde{\theta}_v + \frac{1}{2}k_p \tilde{\eta}_1^T \tilde{\eta}_1 + k_o \tilde{z}^T \tilde{z}, \tag{4}$$

$$\dot{V} = -s_v^T (K_D + D_{RB}) s_v - k_p \lambda_p \tilde{\eta}_1^T \tilde{\eta}_1 - k_o \lambda_o \tilde{\epsilon}^T \tilde{\epsilon}. \tag{5}$$

4. **False**

   The statement is false because, although the ocean current is usually considered as constant, given its classification as a disturbance (a concept helpful for control system design), it is important to note that this disturbance cannot be regarded as constant relative to the body frame. For the **ocean current** to be considered constant, as well as irrotational, it must be referenced in relation to the earth-fixed frame. Current velocity in world frame is expressed by:

$$v_c = \begin{bmatrix} v_{cx} \\ v_{cy} \\ v_{cz} \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad \dot{v}_c = 0 \tag{6}$$

Exercise 3: Quadruped Simulation

To manage various types of legged robot it is usually employed a control-scheme named *whole-body control* (WBC). The objective of WBC is to determine the optimal joint accelerations $\ddot{\mathbf{q}}_\mathbf{j}^*$ and ground reaction forces $\mathbf{f}_\mathbf{gr}^*$, derived by solving a quadratic programming problem (QP), as one can see in the equations below:

$$\min_{\zeta} \quad f(\zeta) \tag{7}$$

$$\text{subject to} \quad A\zeta = b \tag{8}$$

$$D\zeta \leq d \tag{9}$$

Where the chosen vector of control variables is:

$$\zeta = [\ddot{r}_c^T \ \ddot{q}^T \ f_{gr}^T] \in \mathbb{R}^{n_b+n_j+3n_{st}} \quad \text{with} \quad \mathbf{r_c} = [p_c^T \ \eta_c^T]^T \tag{10}$$

The cost function aims at tracking the CoM's reference and the trajectory of the swing feet coming from the motion planner, reducing as much as possible the control effort. Moreover, it aims to achieve accurate tracking of the desired wrench, while adhering to several constraints. These constraints include Dynamic consistency, Non-sliding contact, Torque limits, Swing leg task.

In practice, this QP problem is implemented using the `qpSWIFT` function.

```
% interface with the QP solver qpSWIFT
zval = qpSWIFT(sparse(H),g,sparse(Aeq),beq,sparse(Aineq),bineq);
```

The problem's constraints are specified by the matrices `Aeq` and `Aineq`, along with the vectors `beq` and `bineq`, representing the equality and inequality constraints, respectively. The objective function is represented by the matrices `H` and `g`, while the output, `zval`, contains the optimal solution to the QP problem, providing the desired control inputs solving the constrained optimization problem.

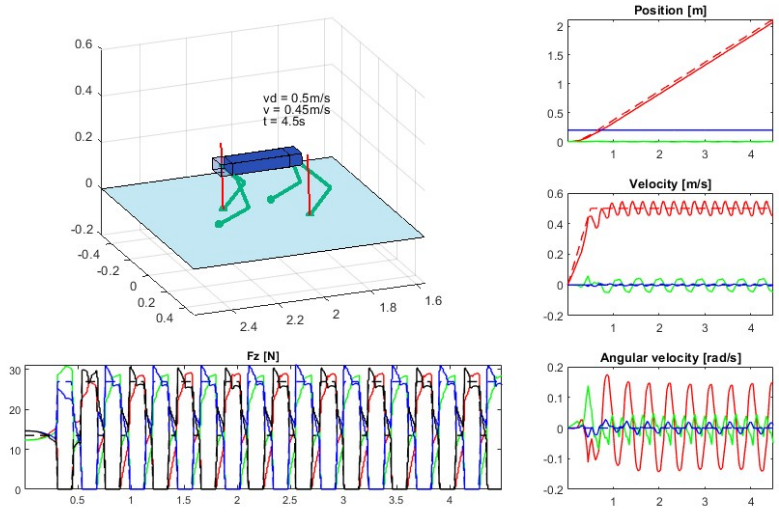
To create an appropriate reference for the swing leg movements and the center of mass (CoM), we also need input from the foot scheduler. This scheduler provides the sequence of feet that are in contact with the ground and those that are swinging. This functionality is crucial for imposing the desired gait on the legged robot. Via the "Foot Scheduler," we can select various gaits:

- **Trot:** The trot gait involves diagonal leg pair movements, enabling faster locomotion but sacrificing stability due to its line-shaped support polygon.
- **Bound:** The bound gait involves the front and rear legs working in pairs, resulting in dynamic movements. There are moments where no feet are in contact with the ground, making it one of the most dynamic gaits, along with the gallop.
- **Pacing:** Lateral legs work in combination, resulting in dynamic movements. While it is more dynamic than the crawl gait, it is less dynamic than the gallop and bound gait.
- **Gallop:** The gallop gait offers the highest velocity for a quadruped robot but has a small stability margin because there are moments when only one foot is in contact with the ground.
- **Trot-Run:** 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; instead, there is a brief moment where all legs are in the swinging phase.
- **Crawl:** Is the most stable gait because it always has three feet in contact with the ground, forming a triangular support polygon, unlike the trot gait where the support polygon is a line. However, the higher stability comes at the cost of reduced speed.

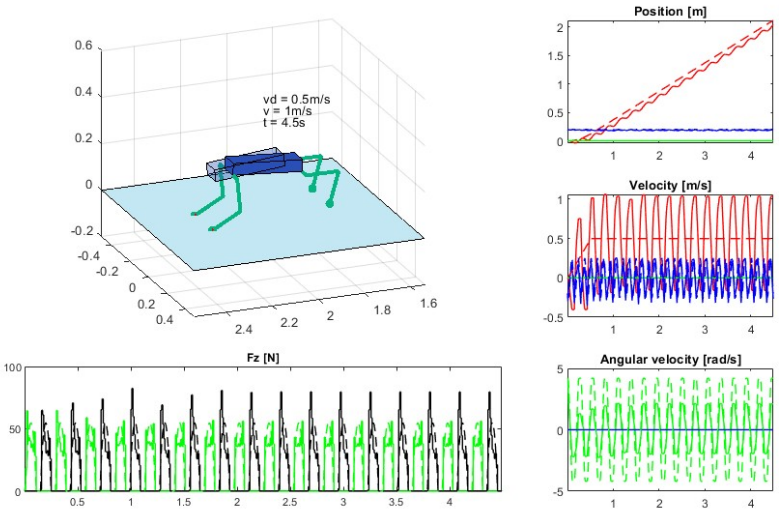
In the simulation, the plot of ground reaction forces reveals the type of gait being performed.

The periodic sequence describes the gait. The plot of $\mathbf{F_z}$ (ground reaction forces along the z-axis) illustrates how the information from the foot scheduler translates into $\ddot{\mathbf{q}}^*$ and $\mathbf{f}_\mathbf{gr}^*$. For each type of gait, the corresponding ground reaction forces F_z will reflect this pattern.

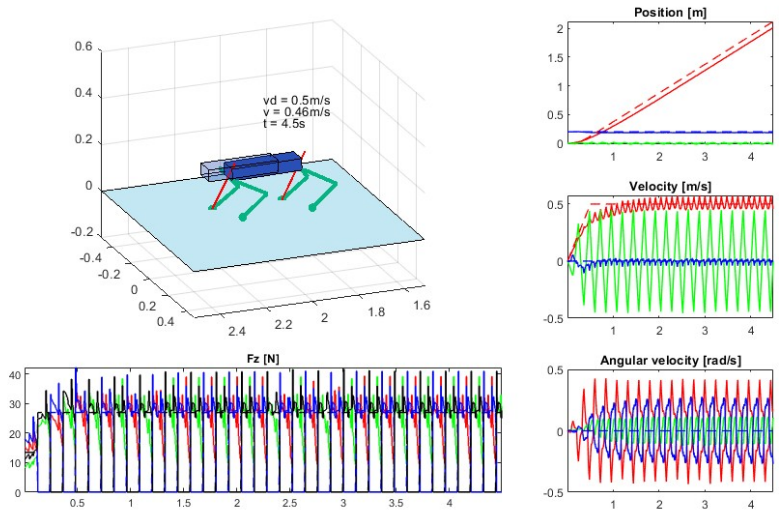
Here are the simulation results of the quadruped robot, where different physical parameters and control values are varied across different gaits. Analyzing these behaviors will help identify the optimal values. Firstly, using the default parameters, the following results were obtained for each gait:



(a) Trot Gait with $m = 5.5\ kg$, $\mu = 1$ and $v_d = 0.5\ m/s$

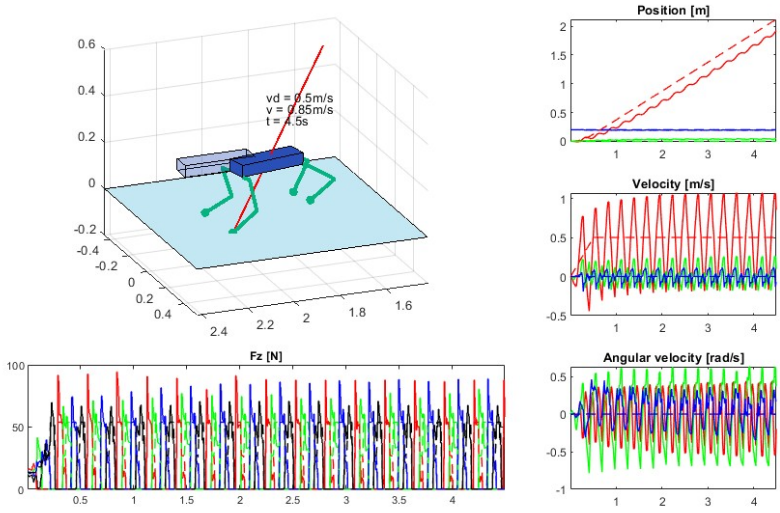


(b) Bound Gait with $m = 5.5\ kg$, $\mu = 1$ and $v_d = 0.5\ m/s$

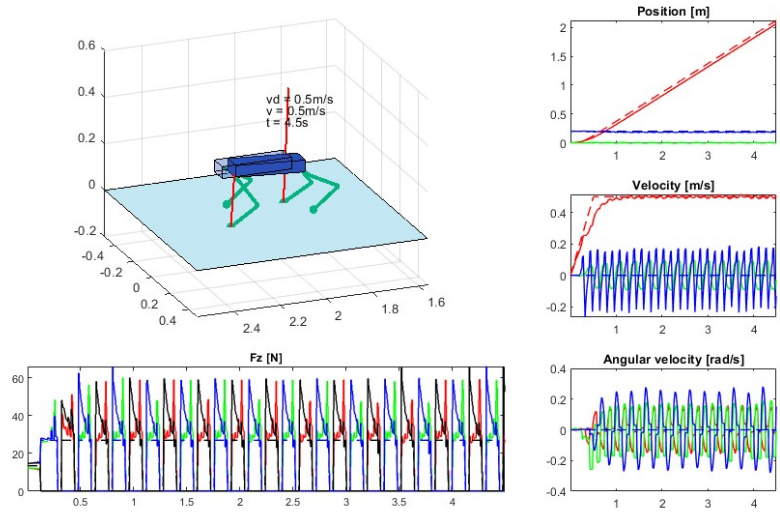


(c) Pacing Gait with $m = 5.5\ kg$, $\mu = 1$ and $v_d = 0.5\ m/s$

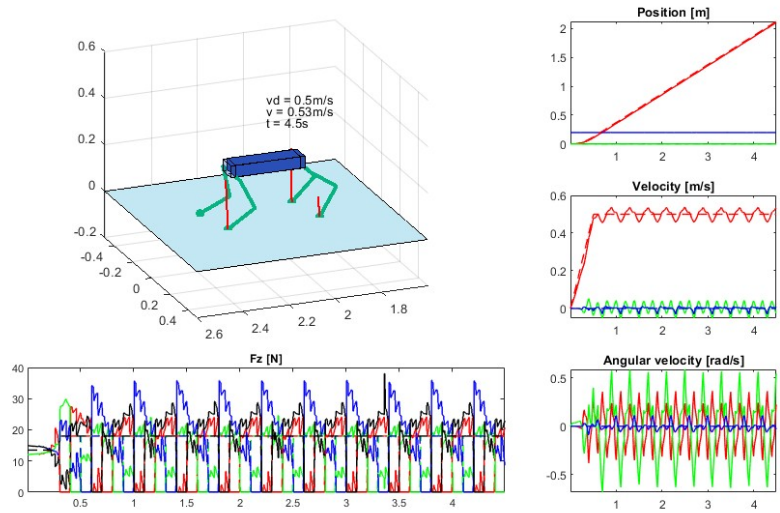
Figure 1: Trot, Bound and Pacing Gait with default parameters



(a) Gallop Gait with $m = 5.5\ kg$, $\mu = 1$ and $v_d = 0.5\ m/s$

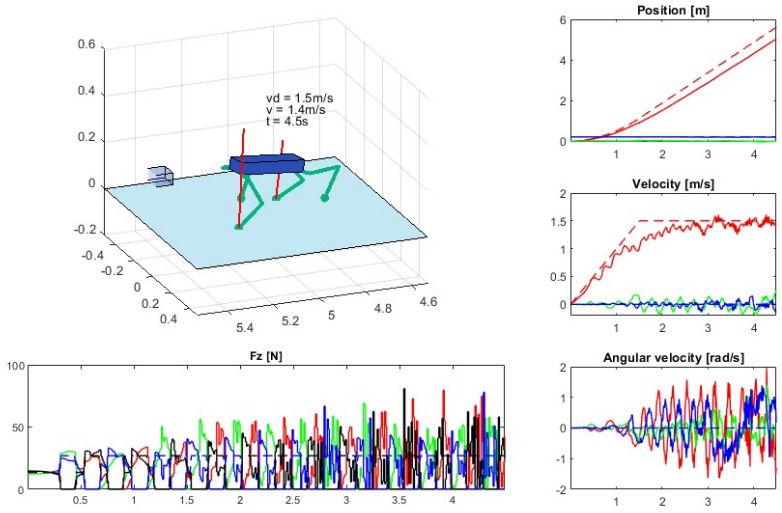


(b) Trot Run Gait with $m = 5.5\ kg$, $\mu = 1$ and $v_d = 0.5\ m/s$

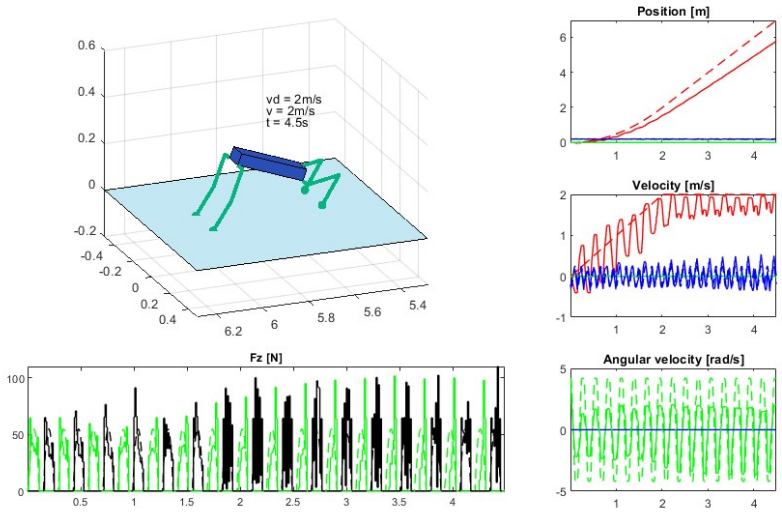


(c) Crawl Gait with $m = 5.5\ kg$, $\mu = 1$ and $v_d = 0.5\ m/s$

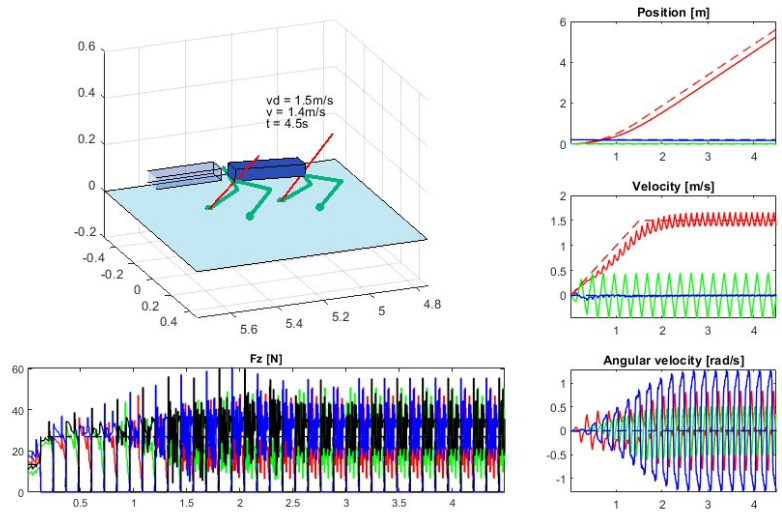
Figure 2: Gallop, Trot Run and Crawl Gait with default parameters



(a) Trot Gait with $m = 5.5\ kg$, $\mu = 1$ and $v_d = 1.5\ m/s$

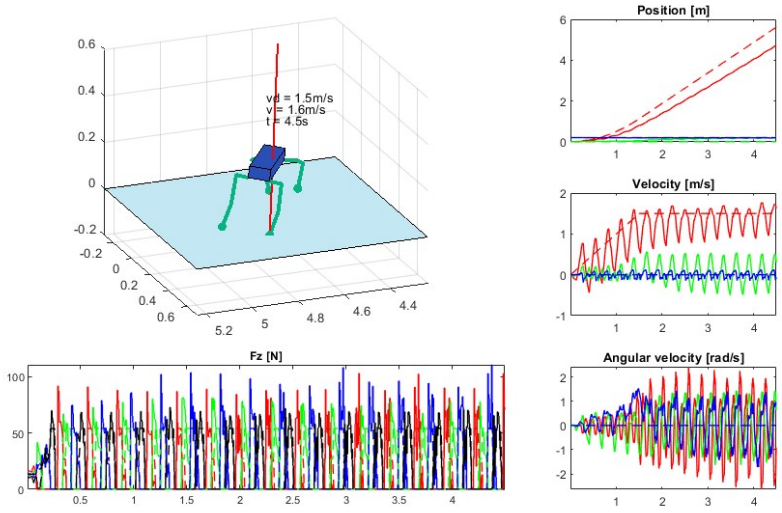


(b) Bound Gait with $m = 5.5\ kg$, $\mu = 1$ and $v_d = 1.5\ m/s$

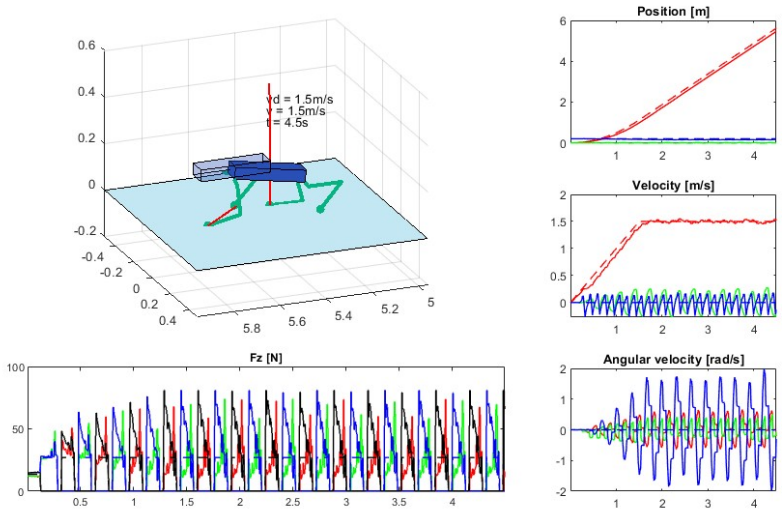


(c) Pacing Gait with $m = 5.5\ kg$, $\mu = 1$ and $v_d = 1.5\ m/s$

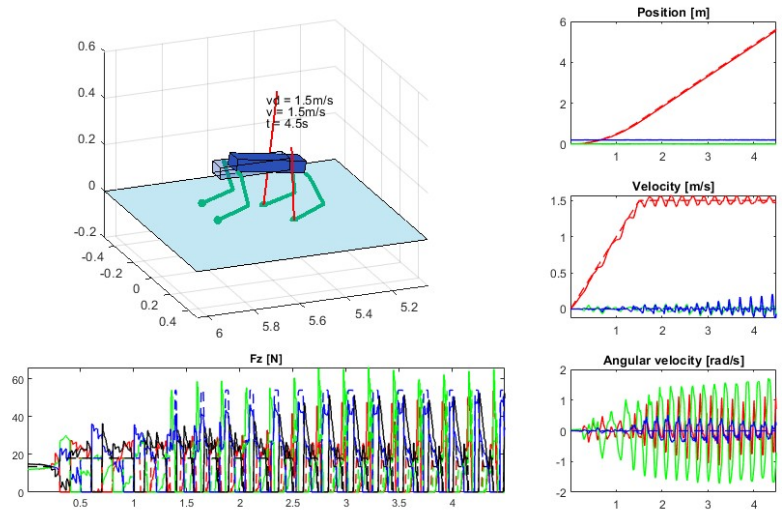
Figure 3: Trot, Bound and Pacing Gait with $v_d = 1.5\ m/s$



(a) Gallop Gait with $m = 5.5\ kg$, $\mu = 1$ and $v_d = 1.5\ m/s$



(b) Trot run Gait with $m = 5.5\ kg$, $\mu = 1$ and $v_d = 1.5\ m/s$



(c) Crawl Gait with $m = 5.5\ kg$, $\mu = 1$ and $v_d = 1.5\ m/s$

Figure 4: Trot run, Gallop and Crawl Gait with $v_d = 1.5\ m/s$

Taking as reference the figure 1 and 2, is possible to analyze the quadruped behaviour under standard condition and with the default parameters:

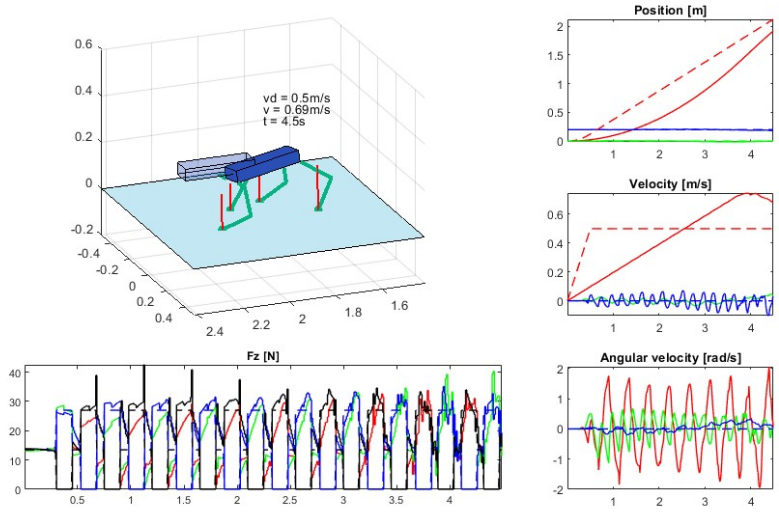
- **Crawl Gait**: precise position tracking, good velocity tracking despite the oscillatory behaviour, that is anyway characterized by low amplitude oscillation near the desired value. F_z (z-component of the ground reaction forces) shows low peak (40 N), and stable ground interaction.
- **Trot Gait**: Similar tracking behaviour of Crawl Gait, with high frequency velocity oscillations and less stability. F_z has bounded peaks (30 N).
- **Trot-Run Gait**: best in steady-state velocity tracking, with low amplitude oscillation behaviour and high accuracy in position tracking. Higher peak of F_z (65 N).
- **Pacing Gait**: decent position tracking and discrete velocity tracking, similar to the trot-run gait. Lower peak in F_z between the highly dynamic gaits (45 N).
- **Bound Gait**: worst velocity tracking between all the gaits, highly dynamic, with high amplitude speed oscillation. F_z peak of 80 N.
- **Gallop Gait**: bad position tracking with high amplitude velocity oscillations. Impulsive ground reaction forces with high peak of F_z (90 N).

To achieve low-speed motion, the crawl gait is the most reliable due to its inherent stability, which is why quadrupedal animals often use it for walking. The crawl gait, along with the trot, ensures good tracking and stability. In contrast, the gallop and bound gaits are less suitable because of their dynamic nature, characterized by higher oscillations and peak forces.

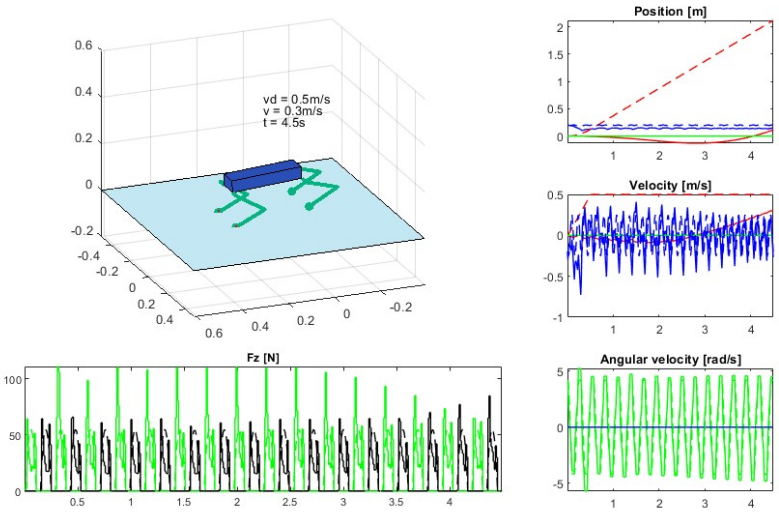
Increasing the desired velocity in the heading direction of the floating base v_d, as briefly showed in figure 3 and 4, the following observation can be made:

- **Crawl Gait**: Despite its precise position tracking at low speed, increasing the speed increase the error, resulting in an increasing oscillatory behaviour. Moreover the current sequence of feet in contact with the ground no longer conforms to the typical pattern of a crawl gait, making it not feasible for high speed velocity.
- **Trot Gait**: As well similar tracking behaviour of Crawl Gait, with more bounded peak of ground reaction forces (60 N).
- **Trot-Run Gait**: best in velocity tracking and high accuracy in position tracking. It maintains consistent speed at steady state with minimal deviations, with F_z peaks higher than 70 N and an increased angular velocity ω_z respect the default simulation.
- **Pacing Gait**: lower peak in F_z between the highly dynamic gaits with good balancing. Robust tracking of position and velocity references, despite worse result respect the default case.
- **Bound Gait**: bad velocity and position tracking, presenting different types of velocity armonics, that can lead to instability.
- **Gallop Gait**: More suitable for higher velocities than trot and crawl gait, it presents some drifting phenomena around its CoM z-axis that can lead to instability and poor tracking of the reference. It has higher impulse of F_z, that are around 100 N.

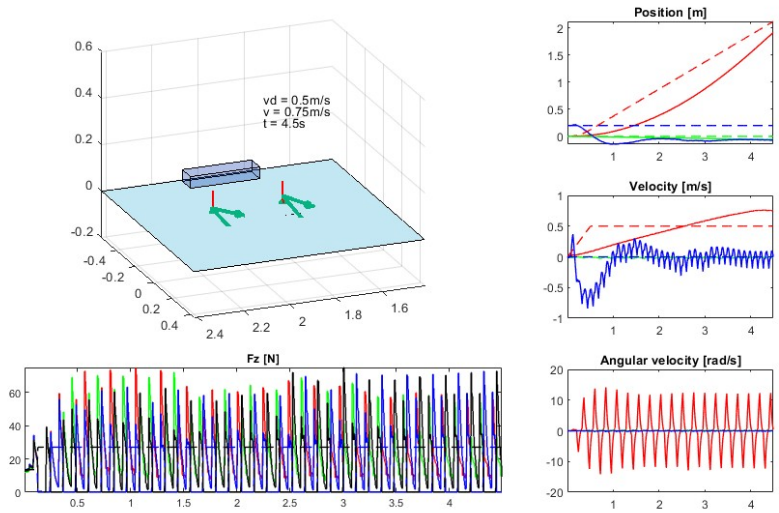
The most effective gaits for medium speed are pacing and trot run, due to their tracking accuracy and stability. Crawl also remain a good choice for position tracking, but it appears that the sequence of the feet in contact with ground don't satisfied the pattern anymore. For high speed gallop seems to be the go to choice, but it present a challenge due to its instability.



(a) Trot Gait with $m = 5.5\ kg$, $\mu = 0.02$ and $v_d = 0.5\ m/s$

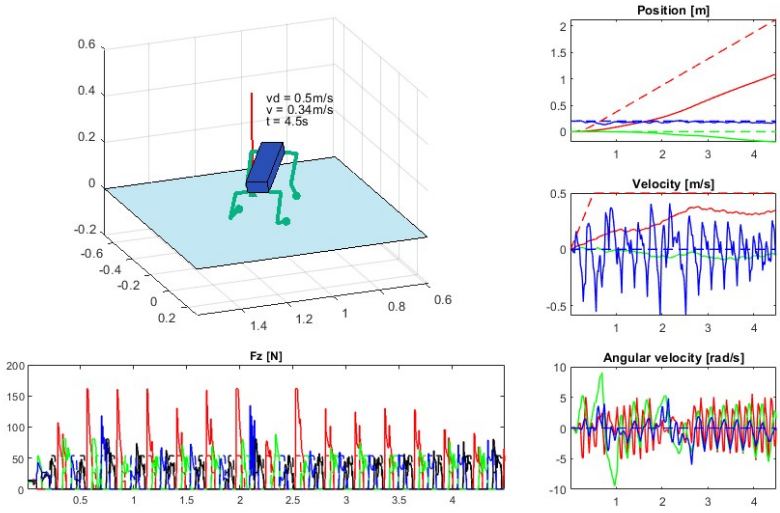


(b) Bound Gait with $m = 5.5\ kg$, $\mu = 0.02$ and $v_d = 0.5\ m/s$

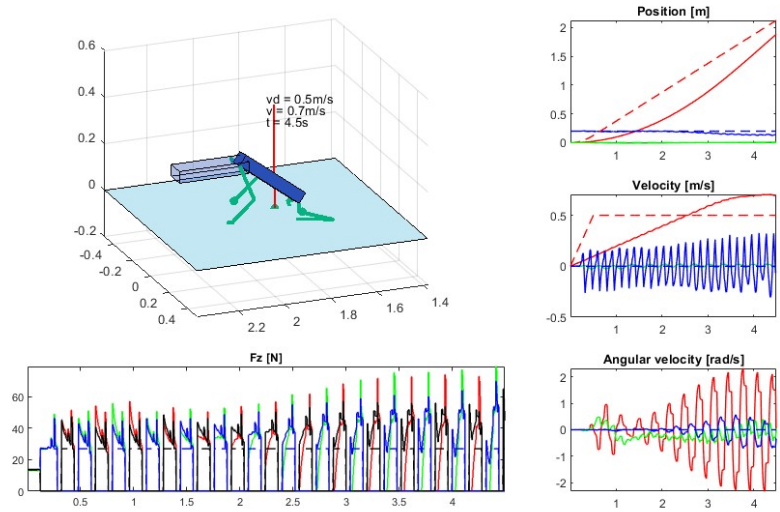


(c) Pacing Gait with $m = 5.5\ kg$, $\mu = 0.02$ and $v_d = 0.5\ m/s$

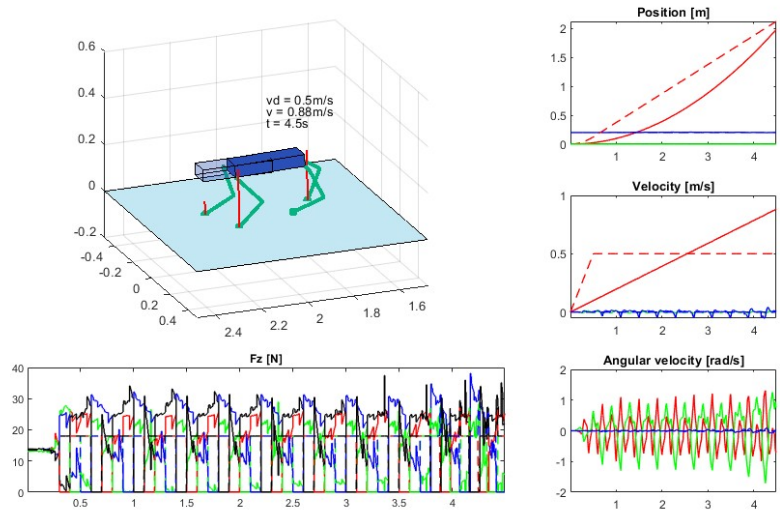
Figure 5: Trot, Bound and Pacing Gait with $\mu = 0.02$



(a) Gallop Gait with $m = 5.5\ kg$, $\mu = 0.02$ and $v_d = 0.5\ m/s$

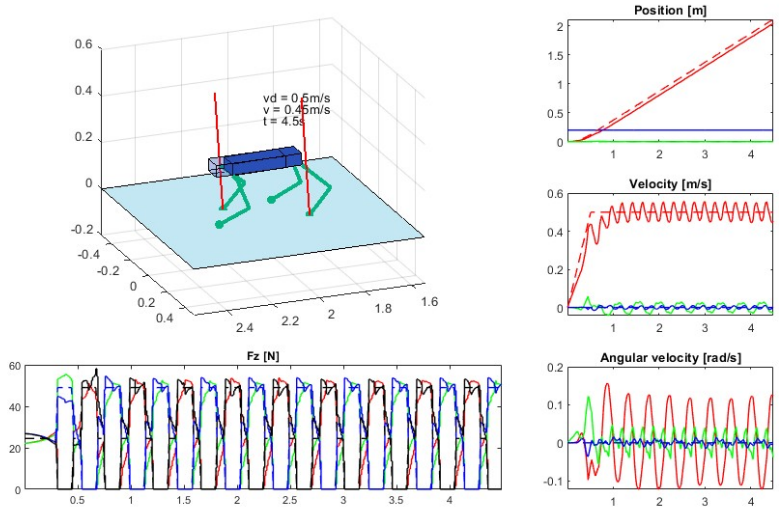


(c) Trot Run Gait with $m = 5.5\ kg$, $\mu = 0.02$ and $v_d = 0.5\ m/s$

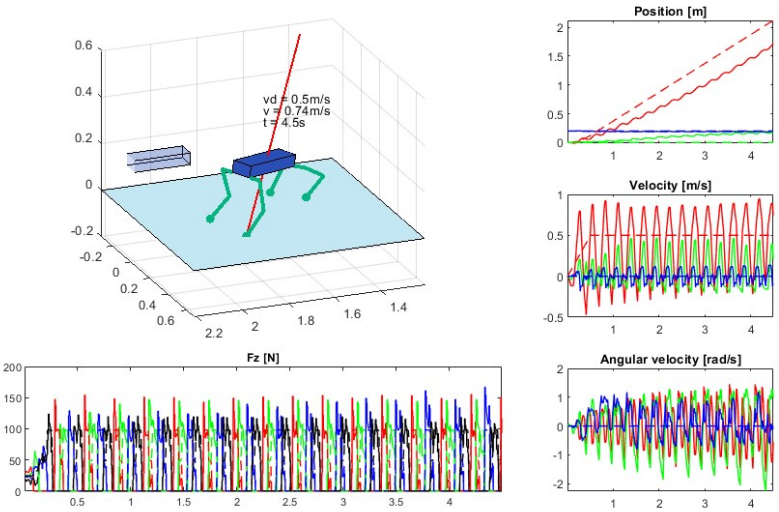


(b) Crawl Gait with $m = 5.5\ kg$, $\mu = 0.02$ and $v_d = 0.5\ m/s$

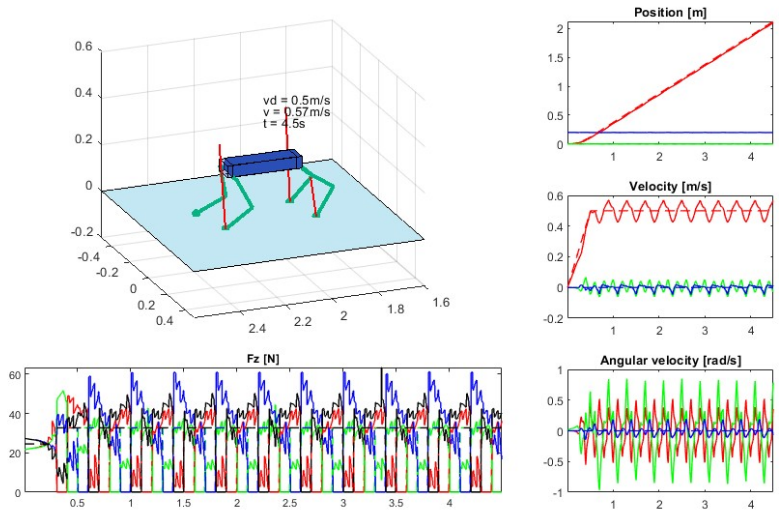
Figure 6: Gallop, Trot Run and Crawl Gait with $\mu = 0.02$



(a) Trot Gait with $m = 10\ kg$, $\mu = 1$ and $v_d = 0.5\ m/s$

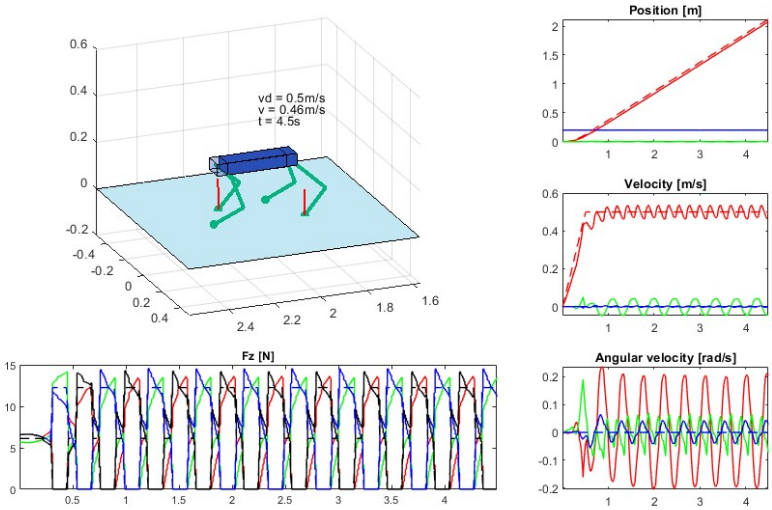


(b) Gallop Gait with $m = 10\ kg$, $\mu = 1$ and $v_d = 0.5\ m/s$

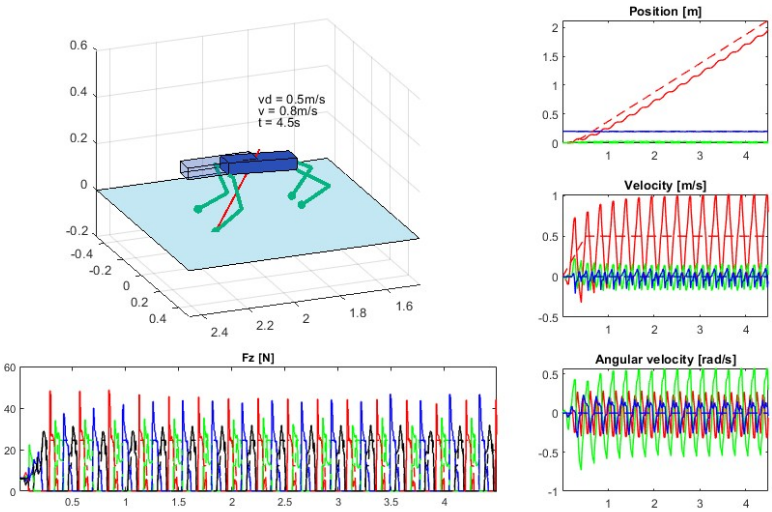


(c) Crawl Gait with $m = 10\ kg$, $\mu = 1$ and $v_d = 0.5\ m/s$

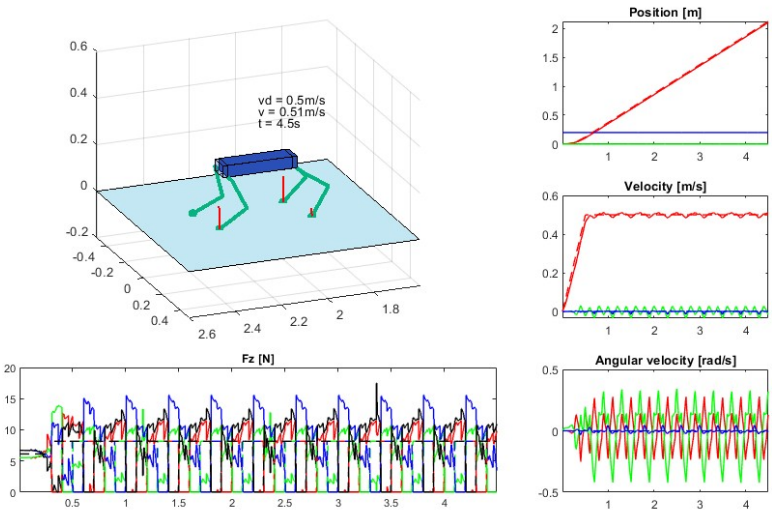
Figure 7: Trot, Gallop and Crawl Gait with $m = 10\ kg$



(a) Trot Gait with $m = 2.5\ kg$, $\mu = 1$ and $v_d = 0.5\ m/s$



(b) Gallop Gait with $m = 2.5\ kg$, $\mu = 1$ and $v_d = 0.5\ m/s$



(c) Crawl Gait with $m = 2.5\ kg$, $\mu = 1$ and $v_d = 0.5\ m/s$

Figure 8: Trot, Gallop and Crawl Gait with $m = 2.5\ kg$

Decreasing the friction coefficient μ, as briefly showed in figure 5 and 6, it can be verified that the overall stability decreases, while also the tracking time increases:

- **Crawl Gait**: performs well even under low friction conditions due to its stable pattern foot sequence and high tracking accuracy. In fact, the ground reaction forces F_z are comparable to the default case, ensuring consistent grip, with peaks around 40 N.
- **Trot Gait**: Shows position tracking similar to the Crawl gait under reduced friction conditions, with marginally lower body stability (resulting in higher angular velocity oscillations). Demonstrates better velocity reference tracking compared to the Crawl gait. Moreover F_z peaks around 40 N.
- **Trot-Run Gait**: Under low friction conditions, its balance diminishes gradually as it maintains less contact with the ground compared to the trot gait. This leads to an increase in the amplitude of body oscillations.
- **Pacing Gait**: result unbalanced under low friction, with high CoM's angular velocity oscillation.
- **Bound Gait**: it suffers from low friction. In fact the quadruped moves with low speed or remains stuck, due to the reduced adherence that causes the slipping of its legs with the surface.
- **Gallop Gait**: it presents some drifting phenomena around its CoM z-axis that can lead to instability, causing the robot to fall and encounter singularities. Poor tracking of the reference.

Reducing the coefficient of friction results in low grip conditions and increased delay in position tracking. The crawl and trot gaits exhibit the best performance, with comparable position tracking and ground reaction forces, while, in the bound gait, the robot remains stuck in the initial position. Moreover, The pacing and gallop gaits are unstable and lead to the robot falling down.

Finally were made some simulations augmenting and diminishing the quadruped mass, in order to analyze the effects on the different gaits.

When the robot's weight was increased (figure 7), the crawl, trot, and trot-run gaits effectively managed the added load, maintaining good tracking performance. The crawl gait remained the most stable, handling the additional mass without significant issues. However, the pacing gait struggled, leading to ground collisions. The bound and gallop gaits became increasingly unstable, displaying unpredictable behavior due to the higher dynamic forces involved.

When the robot's weight was reduced (figure 8), the crawl gait maintained excellent stability and control, showing near-perfect behavior. The trot, trot-run, and pacing gaits also improved, with reduced oscillations and enhanced responsiveness. However, the bound and gallop gaits showed no significant differences from standard simulations.

In order to improve higher dynamic gait performance, the $K_{p,sw}$ (the parameter in the optimal function controlling the swing legs) control parameter has been decreased. This improvement occurs because reducing $K_{p,sw}$ allows swing legs to track poorly their swinging movement, hence the optimization focuses less on them, allowing attention to other factors such as errors on the center of mass. (figure 9)

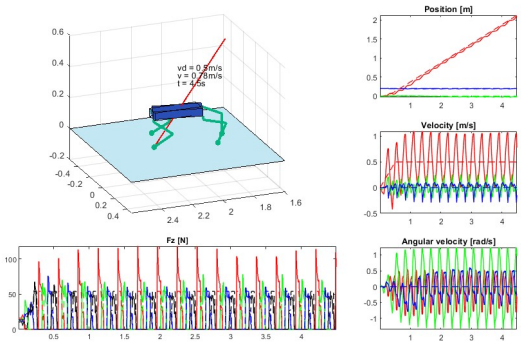


Figure 9: Gallop Gait with $m = 5.5\ kg$, $\mu = 1$, $v_d = 0.5\ m/s$, $k_{p,sw} = 20$

The code of the `quadruped_simulation`, as well as the output video of the various simulations are on github: https://github.com/vincip99/HW_Field_Service_Robotics

Exercise 4

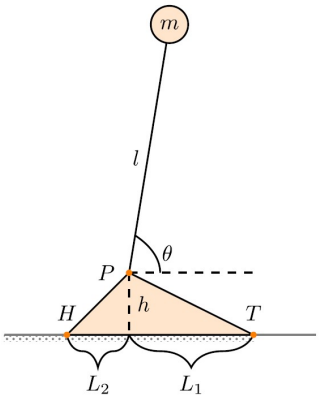


Figure 10: legged robot scheme

- Without an actuator at the point P, we cannot counter the torque created by the gravitational force. In fact, this system without actuation is similar to an inverted pendulum, where the equilibrium point at $\theta = \frac{\pi}{2}$ is unstable.

  Therefore, if there is a small angular displacement $+\epsilon$ from this point, the system will move indefinitely away from the equilibrium position. Consequently, the system in the configuration $\theta = \frac{\pi}{2} + \epsilon$ is not stable.

- The Zero Moment Point (ZMP) is the point where the horizontal moments of the ground reaction forces with respect to the Center of Pressure (CoP) are zero. By examining Figure (10), we can compute the ZMP. In our case, it is only defined by an x-coordinate since we consider the support polygon as a segment. Therefore, we use Equation (11) to compute p_c^x, p_c^z, $\ddot{p}_c^x$, and $\ddot{p}_c^z$.

$$p_z^x = p_c^x - \frac{p_c^z}{(\ddot{p}_c^z - g_0)}(\ddot{p}_c^x - g_0) - \frac{1}{m(\ddot{p}_c^z - g_0)}(\ddot{p}_{zc} - g_{z0})\dot{L}_y \tag{11}$$

  In this exercise, the only mass is m. Therefore, the Center of Mass (CoM) of the robotic system in the z-x plane is simply the position of m in the plane, given by the coordinates p_c^x and p_c^z. By computing the second time derivatives of these quantities, we can also compute $\ddot{p}_c^x$ and $\ddot{p}_c^z$ (as shown in Equations (12) and (13)).

$$p_c^x = l\cos(\theta) \quad \ddot{p}_c^x = -l\sin(\theta)\ddot{\theta} - l\cos(\theta)\dot{\theta}^2 \tag{12}$$

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

$$p_c^z = h + l\sin(\theta) \quad \ddot{p}_c^z = -l\sin(\theta)\dot{\theta}^2 + l\cos(\theta)\ddot{\theta} \tag{13}$$

$$g_0 = -g \tag{12}$$

$$p_z^x = l\cos(\theta) - \frac{h + lsin(\theta)}{l\ddot{\theta}cos(\theta) - l\dot{\theta}^2 sin(\theta) + g}(-l\ddot{\theta}sin(\theta) - l\dot{\theta}^2 cos(\theta)) - \frac{l^2\ddot{\theta}}{l\ddot{\theta}cos(\theta) - l\dot{\theta}^2 sin(\theta) + g} \tag{14}$$

- In this case, supposing to have an actuator at the ankle capable of perfectly cancelling the torque around P, due to the gravity $\dot{\theta} = 0$ and $\ddot{\theta} = 0$, and the ground is a 1D space. From the Equation $p_{xc} = l\cos(\theta)$, which represents the projection of the CoM on the ground.

  Next, recall that the support polygon in this case is only the segment HT. By examining how the frame is set up in Figure (7), we can geometrically determine the range of θ values that the robot can achieve without falling, as shown in Equations (15) and (16).

$$-L_2 \leq l\cos(\theta) \leq L_1 \tag{15}$$

$$\begin{cases} \theta \geq \arccos\left(\frac{L_1}{l}\right) \\ \theta \leq \arccos\left(\frac{-L_2}{l}\right) \end{cases} \tag{16}$$

In conclusion, if Equation (16) is not satisfied, the CoM projection falls outside the support segment HT, causing the robot to fall.