FSR- Homework 4

Stefano Riccardi P38000210

May 2024

1 Point 1

When a rigid body is submerged in a fluid under the effect of gravity, the buoyancy effect must be considered. This is a hydrostatic effect caused by the buoyancy force, which is a force that can be seen as acting on a point, the center of buoyancy, opposing the gravity force and, thus, trying to lift the body immersed in a fluid such as water. It is important to note that the center of buoyancy is different from the center of mass. Due to this difference, a torque is generated, which can affect the stability of the submerged body.

This behavior can be explained by Archimedes' principle, which states that a body submerged in a fluid experiences an upward buoyant force equal to the weight of the displaced fluid. This buoyant force can be expressed according to Archimedes' formula:

$$F = \rho \cdot \Delta \cdot g$$

where

- ρ is the density of the fluid,
- Δ is the submerged volume of the body (which depends on whether the body is partially or fully submerged),
- g is the acceleration due to gravity.

It's possible to express the buoyancy force in the body frame using the transpose of the rotation matrix between body frame and world frame:

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

Note that this force depends on the density of the fluid in which the body is immersed. Since the density of air is much lower compared to that of water and of the moving mechanical system, the buoyancy effect is neglected in aerial robotics.

2 Point 2

2.1 The added mass effect considers an additional load to the structure

FALSE - The added mass effect cannot be considered simply as an additional load on the structure. Instead, it is a more complex phenomenon that must be accounted for in the dynamic model of a structure moving through a fluid. In the case of a completely submerged underwater robot, the added mass effect is incorporated into the dynamic model by modifying the inertia matrix usign the matrix:

$$M_A \in R^{6\times 6}$$

which is a matrix not necessarily positive definite, where the elements are generally non-zero. This matrix is positive definite only in the case of an ideal fluid, the UUV has low-speed velocity, there are no currents, and there are no waves in the sea.

Specifically, the added mass introduces changes that result in a new inertia matrix, which is no longer positive definite. This modified matrix reflects the influence of the surrounding fluid on the dynamics of the structure, capturing how the fluid's inertia affects the robot's motion. Thus, the added mass effect is more than just an additional load.

2.2 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 - The effect of added mass is significant in underwater robotics because the density of the water is comparable to that of the robot. When an underwater robot moves, it must displace the surrounding water and this displacement creates additional forces due to the inertia of the displaced water. Since the density of the water is relatively high and close to that of the robot, these forces are considerable and must be taken into account to accurately model the robot's dynamics.

In contrast, this effect is not considered in UAVs because the air density is much lower than that of the UAV. The low air density means that the inertial forces generated by air displacement are negligible. Therefore, the effect of added mass has no significant impact on UAV dynamics and can be safely ignored

2.3 The damping effect helps in the stability analysis

TRUE - The damping effect helps in the stability analysis. Indeed, damping reduces oscillations and helps stabilise the system by dissipating energy, which is crucial for maintaining control and stability. This effect in the context of aerial and underwater robotics is caused by the viscosity of the fluid in which

the robot is immersed, which causes the presence of dissipative drag and lift forces on the body. Due to the high viscosity of water, damping is particularly relevant in underwater robotics.

2.4 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 is usually considered constant and irrotational but is better referred to with respect to the world frame, not the body frame. In the body frame, the ocean current would appear to change continuously as the body moves, while in the world frame, it can be treated as a constant vector.

$$\nu_c = \begin{bmatrix} \nu_{c,x} \\ \nu_{c,y} \\ \nu_{c,z} \\ 0 \\ 0 \\ 0 \end{bmatrix} \in R^6, \dot{\nu}_c = 0_6$$

This effect can be incorporated into the dynamic model of a rigid body moving in a fluid by considering the relative velocity in the body-fixed frame $\mathbf{v}_r$ during the derivation of the Coriolis, centripetal, and damping terms.

$$v_r = \begin{bmatrix} \dot{p}_b^b \\ \omega_b^b \end{bmatrix} - R_b^T v_c$$

3 Point 3

3.1 Quadratic function

To implement the quadratic function to be minimised in order to impose the desired ground reaction forces and the desired CoM reference on the robot with the following form:

$$\begin{aligned} \min \quad & \frac{1}{2} x^T H x + g^T x \\ \text{s.t} \quad & A_{\text{ineq}} x \leq b_{\text{ineq}}, \\ & A_{\text{eq}} x = b_{\text{eq}}. \end{aligned}$$

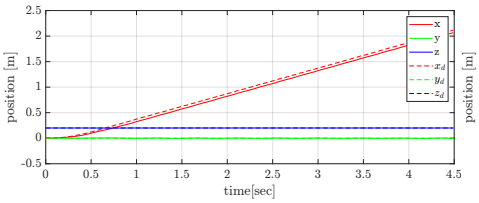
the cost function, equality constraints and inequality constraints parameters were passed to the QP solver qpSWIFT as follow:

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

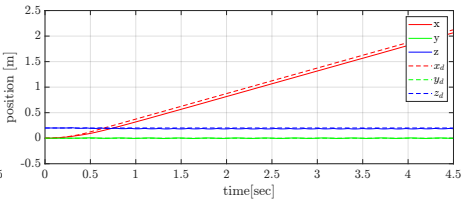
3.2 Simulation

In order to explore the different behaviors of the robot by varying the parameters, several simulations were carried out, and the results are commented below.

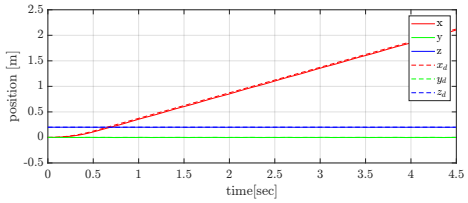
The first simulation was conducted by keeping all the robot parameters and references unchanged, exploring all the gaits. Results show that for all gaits, the position of the CoM and the ground reaction force follow the desired reference. Comparing the behaviors for the various gaits, it is noted that the best results in terms of position tracking occur with the crawl, trot, and trot-run gaits, where the position error remains almost constant (Figure 3.1).



(a) Trot gait



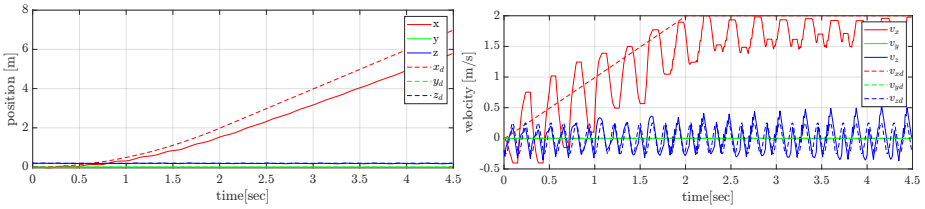
(b) Trot-run gait



(c) Crawl gait

Figure 3.1: Position tracking for different gaits

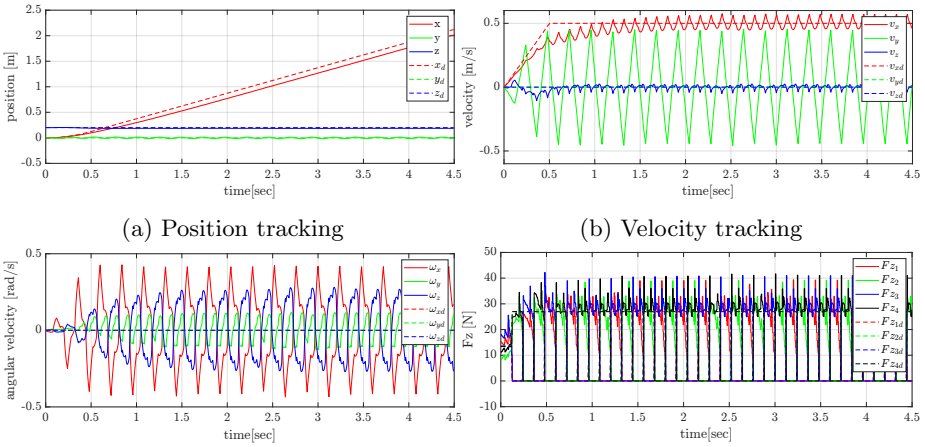
The worst result is observed with the bound gait, where the position error constantly increases, and the velocity does not follow the desired velocity, resulting in a significant error (Figure 3.2).



(a) Position tracking (b) Velocity tracking

Figure 3.2: Bound gait

Regarding the pacing gait, it achieves good position tracking but exhibits a lot of oscillation in velocity and angular velocity around the reference values, and less smooth behavior concerning Fz (Figure 3.3).

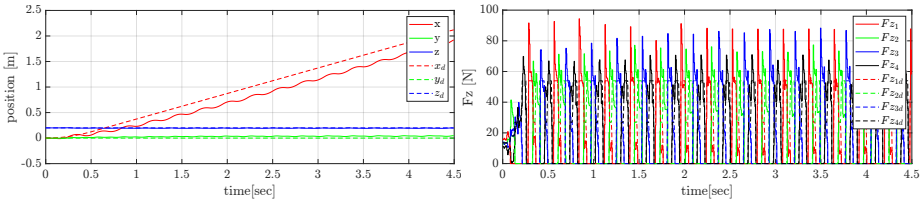


(a) Position tracking (b) Velocity tracking

(c) Angular Velocity tracking (d) Fz tracking

Figure 3.3: Pacing gait

The gallop gait presents a more or less constant error in position tracking during the simulation, while the ground reaction forces' overshoot increases significantly (Figure 3.4).



(a) Position tracking (b) Fz tracking

Figure 3.4: Gallop gait

Another simulation was conducted by varying only the reference velocity from 0.5 m/s to 1.6 m/s. It turns out that for some configurations the robot's behavior changes. Comparing the behavior of the trot gait at 1.6 m/s with the previous one, we can notice that the position tracking is not efficient, the position error tends to increase(Figure 3.5).

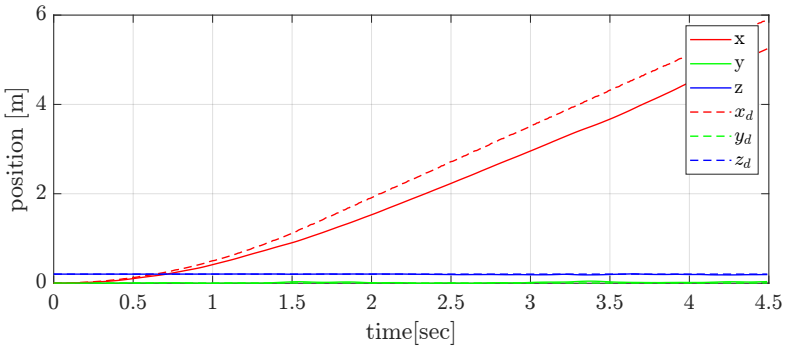
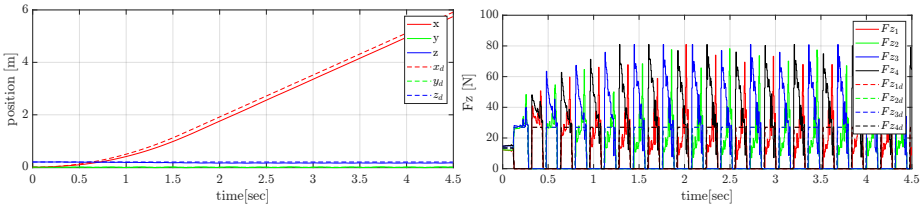


Figure 3.5: Position Tracking Trot gait

In contrast, for the trot-run gait, the position tracking behavior remains almost unchanged, although the overshoot of the ground reaction forces increases significantly (Figure 3.6).

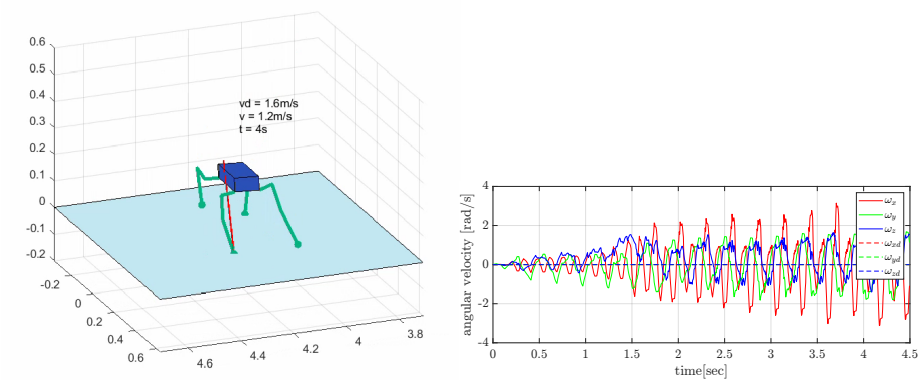


(a) Position tracking (b) Fz tracking

Figure 3.6: Trot-run gait

A particularly significant variation in the robot's behavior can be observed in the gallop gait, where the robot rotates due to an increase in the error in

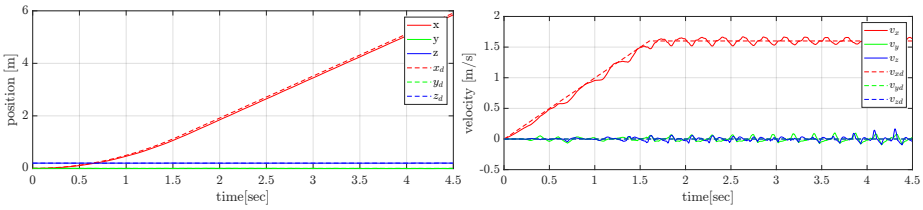
angular velocity tracking (Figure 3.7).



(a) Robot Frame (b) Angular Velocity tracking

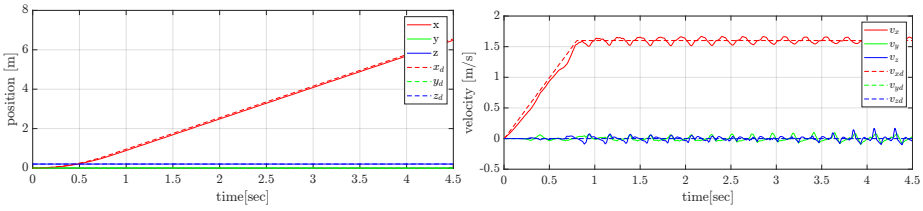
Figure 3.7: Gallop gait

Once again, the best behavior is observed with the crawl gait(Figure 3.8). For this reason, another simulation was implemented by increasing the desired acceleration, but no significant differences were observed in the quadruped robot's behavior(Figure 3.9).



(a) Position tracking (b) Velocity tracking

Figure 3.8: Crawl gait

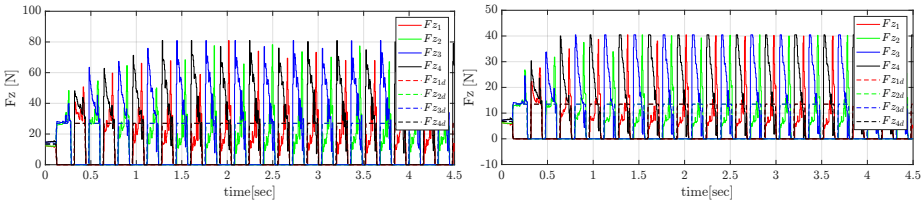


(a) Position tracking (b) Velocity tracking

Figure 3.9: Crawl gait with desired acceleration

Regarding the pacing gait and the bound gait, the robot's behavior remains almost unchanged compared to that at 0.5 m/s.

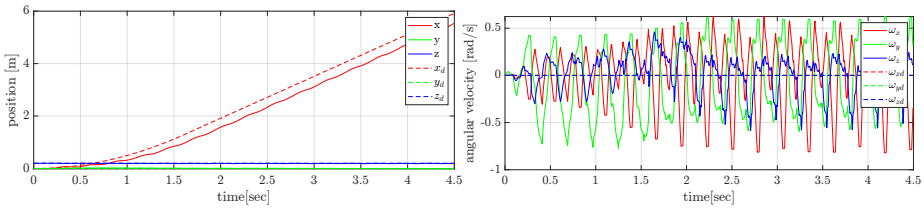
To improve the performance concerning the tracking of the CoM position and the ground reaction force, the mass of the robot was reduced to $m = \frac{5.5}{2}$ kg while keeping the desired velocity $v_d = 1.6$ m/s. Generally, the behaviors in the various gaits did not undergo significant variations in position tracking. However, as expected, the ground reaction force F_z decreases, as shown in the comparison between F_z in the run-trot gait with $m = 5.5$ kg and $m = 2.75$ kg, where a 50% decrease in the ground reaction force is visible (Figure 3.10).



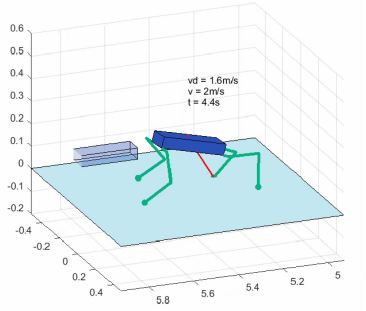
(a) m=5.5 (b) m=2.75

Figure 3.10: Fz comparison in run-trot gait with different mass

A particularly interesting result was achieved with the gallop gait, where the error in angular velocities decreased significantly, and the robot exhibited better performance compared to the previous configuration(Figure 3.11).



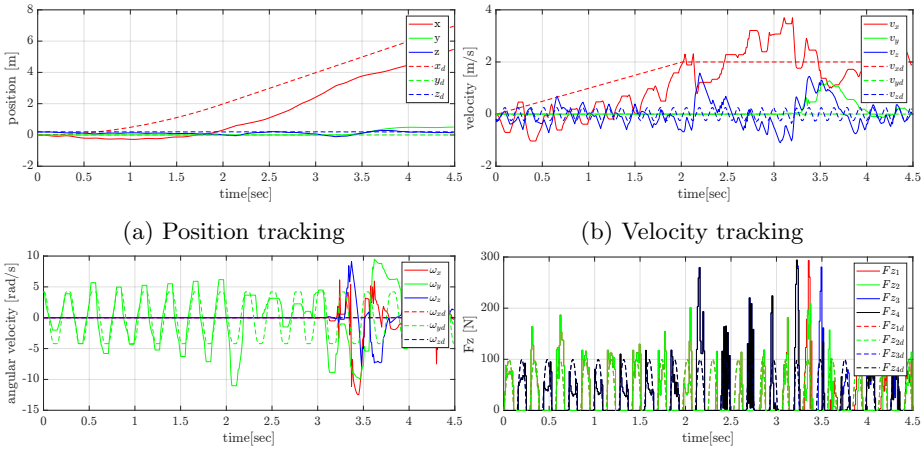
(a) Position Tracking (b) Angular velocity tracking



(c) Robot Frame

Figure 3.11: Gallop gait m=2.75

Whereas, attempting to increase the mass to $m = 10$ kg results in an increase in the ground reaction force and, in some cases, the robot goes into instability. In the bound gait, where it is observed that the robot completely fails the task, there is an increase in the ground reaction forces and angular velocities, as well as poor position tracking (Figure 3.12).



(a) Position tracking

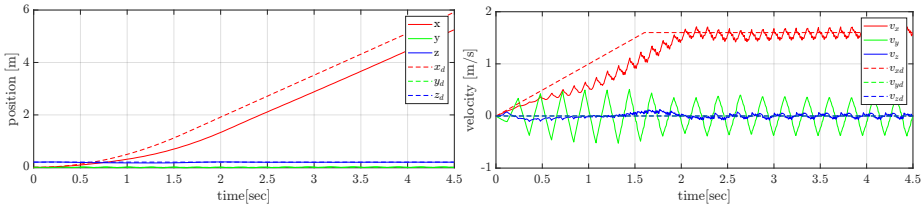
(b) Velocity tracking

(c) Angular Velocity tracking

(d) Fz tracking

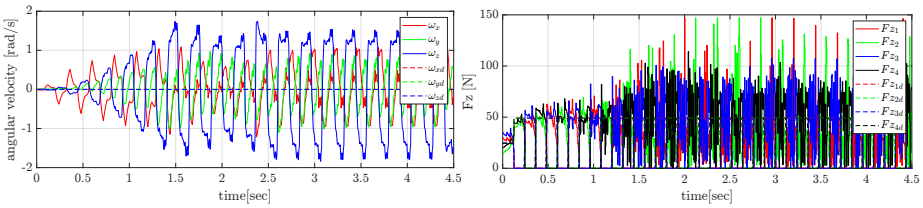
Figure 3.12: Bound gait m=10kg

In other cases, such as the pacing gait, the robot maintains a larger position tracking error compared to simulations with a lower mass, but the robot still manages to maintain good behavior and proper tracking of linear velocities, despite an increase in ground reaction forces and angular velocities(Figure 3.13).



(a) Position tracking

(b) Velocity tracking

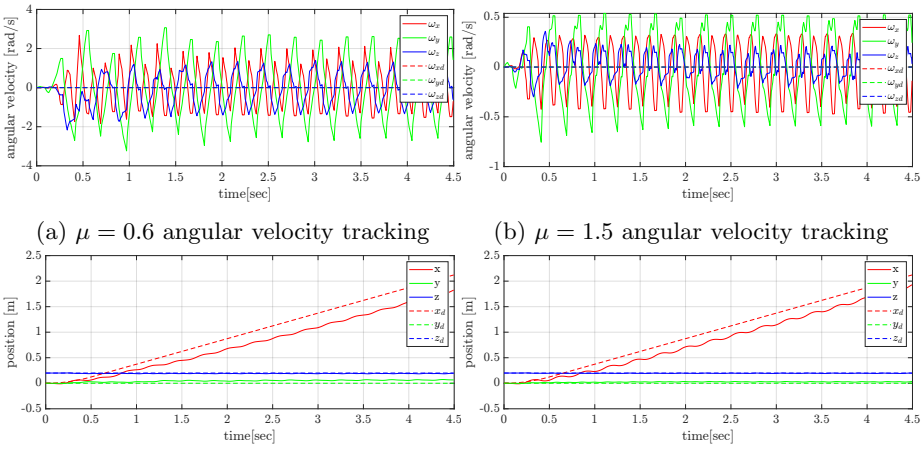


(a) Angular Velocity tracking (b) Fz tracking

Figure 3.14: Pancing gait m=10kg

Also in this case, the best result is also the crawl gait. By changing the friction coefficient μ, two simulations were conducted, $\mu = 0.6$ and $\mu = 1.5$. In both cases, the desired speed is 0.5 m/s and the mass is 5.5 kg. It results that in some gaits there are no significant variations, such as in the crawl gait which continues to have the best performance, and in the trot and trot run where the performances do not change significantly. Interesting results were obtained for the pacing gait and for the gallop gait.

For the gallop gait, it is found that increasing the coefficient μ decreases the error related to the angular velocity(Figure 3.15, b), resulting in better position tracking (Figure 3.15).

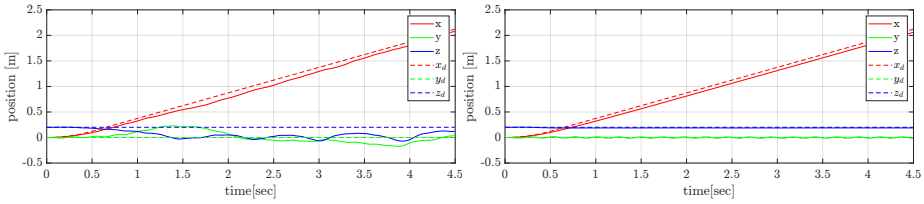


(a) $\mu = 0.6$ angular velocity tracking (b) $\mu = 1.5$ angular velocity tracking

(c) $\mu = 0.6$ position tracking (d) $\mu = 1.5$ position tracking

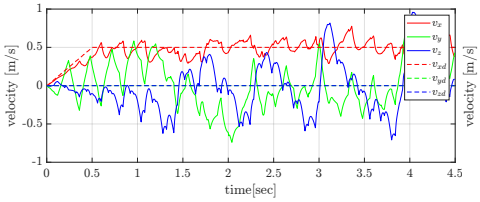
Figure 3.15: Gallop gait, $\mu = 0.6$ sx $\mu = 1.5$ dx

For the pacing gait, it is observed that by decreasing the coefficient to $\mu = 0.6$, the robot fails the task by falling, while by increasing μ to $\mu = 1.5$, the robot completes the task with good position and velocity tracking. Below is a comparison of the results, on the right $\mu = 0.6$, on the left $\mu = 1.5$ (Figure).

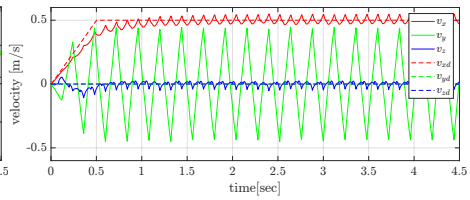


(a) $\mu = 0.6$ position tracking

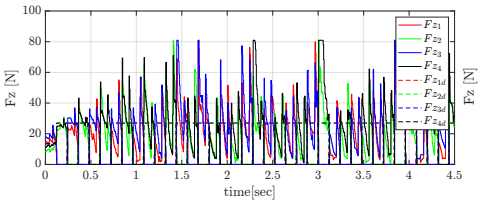
(b) $\mu = 1.5$ position tracking



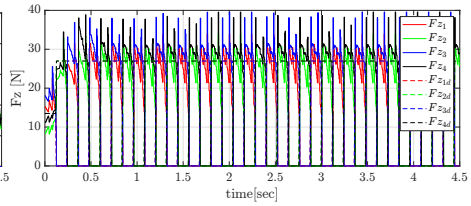
(c) $\mu = 0.6$ velocity tracking



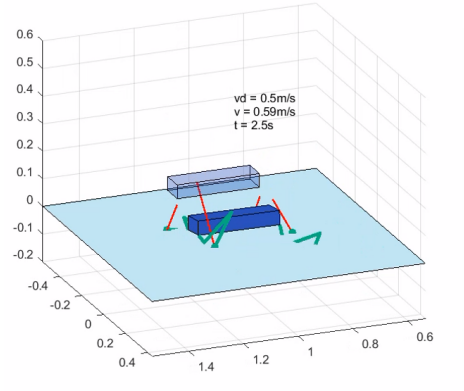
(d) $\mu = 1.5$ velocity tracking



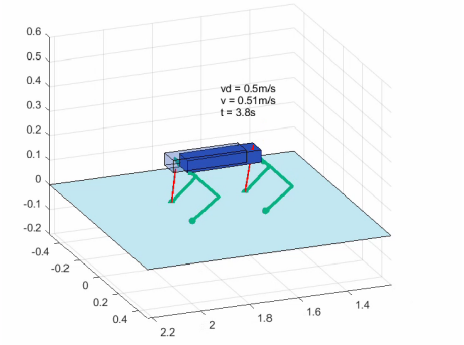
(e) $\mu = 0.6$ Fz tracking



(f) $\mu = 1.5$ Fz tracking



(g) $\mu = 0.6$ Robot configuration



(h) $\mu = 1.5$ Robot configuration

Figure 3.16: Pacing gait, $\mu = 0.6$ sx $\mu = 1.5$ dx

[N.B Results and videos for all simulations are in 'FSR-HW4/Ex3-Results'.]

4 Point 4

Part a: Stability without an actuator at point P

In this configuration, without an actuator at point P, the leg behaves similarly to an inverted pendulum positioned vertically ($\theta = \frac{\pi}{2}$). At $\theta = \frac{\pi}{2} + \epsilon$, the robot's leg is almost vertical but slightly tipped over. Since there is no actuator at P, there is no torque or force that can correct the tipping motion. Gravity will cause the mass m to fall further in the direction of ϵ. Thus, the system is not stable at $\theta = \frac{\pi}{2} + \epsilon$.

Part b: Zero-moment point (ZMP) computation

The Zero-Moment Point (ZMP) is the point where the total moment due to gravity and inertial forces is zero. It can be matemathicaly expressed as:

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

With:

$$S = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \in SO(2)$$

It's possible to considering only the x component of this equation, since the robot leg can only move in x and z direction and rotate around the y-axis. For this reason the equation can be rewritten as follow:

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

At this point it is possible to do some geometrical and physical considerations:

- Position in the x direction (p_x):

  $$p_x = l \cdot \cos\theta \quad \Rightarrow \quad \ddot{p}_x = l \cdot \left(-\ddot{\theta} \cdot \sin\theta - \dot{\theta}^2 \cdot \cos\theta\right);$$

  This equation describes the horizontal position of a point c as a function of the leg angle θ. The second derivative gives the horizontal acceleration, considering both the angular acceleration and the angular velocity squared.

- Position in the z direction (p_z):

  $$p_z = l \cdot \sin\theta + h \quad \Rightarrow \quad \ddot{p}_z = l \cdot \left(\ddot{\theta} \cdot \cos\theta - \dot{\theta}^2 \cdot \sin\theta\right);$$

  This equation describes the vertical position of the point c. The second derivative provides the vertical acceleration, again considering both the angular acceleration and the angular velocity squared.

- Gravitational acceleration components:

  $g_x = 0 \quad \text{and} \quad g_z = -g, \quad$ because we have a flat horizontal surface;

  These components indicate that there is no horizontal gravitational force (since the surface is flat), and the vertical gravitational force is $-g$.

- Angular momentum (L):

  $L = \mathbf{l} \times m \cdot \mathbf{v} = m \cdot l^2 \cdot \omega \quad (\mathbf{l} \text{ and } \mathbf{v} \text{ are orthogonal and } v = \omega \cdot l) = m \cdot l^2 \cdot \dot{\theta};$

  The angular momentum is calculated as the cross product of the position vector $\mathbf{l}$ and the linear momentum $m \cdot \mathbf{v}$. Since $\mathbf{l}$ and $\mathbf{v}$ are orthogonal, the angular momentum simplifies to $m \cdot l^2 \cdot \dot{\theta}$.

- Derivative of angular momentum:

  $$\dot{L} = m \cdot l^2 \cdot \ddot{\theta};$$

  The time derivative of the angular momentum gives the rate of change of angular momentum, which depends on the moment of inertia $m \cdot l^2$ and the angular acceleration $\ddot{\theta}$.

Substituting these quantities inside the ZMP it's possible to compute the ZMP as follow:

$$ZMP = l\cos\theta - \frac{l\sin\theta + h}{l(\ddot{\theta}\cos\theta - \dot{\theta}^2\sin\theta) + g} \cdot l(-\ddot{\theta}\sin\theta - \dot{\theta}^2\cos\theta) - \frac{l^2 \cdot \ddot{\theta}}{l(\ddot{\theta}\cos\theta - \dot{\theta}^2\sin\theta) + g} \tag{1}$$

Part c: Value of θ with an actuator

In this situation, with the actuator counteracting the gravitational torque around point P, the robot's balance depends on the projection of the mass m onto the ground remaining within the support polygon. The support polygon in this case is defined by the line $L = L1 + L2$. Therefore, the robot will remain stable as long as the following condition is met:

$$-L_2 \leq l \cdot \cos\theta \leq L_1 \rightarrow -\frac{L_2}{l} \leq \cos\theta \leq \frac{L_1}{l} \tag{2}$$

Results that the robot is stable for the following values of θ

$$arccos\left(\frac{L_1}{l}\right) \leq \theta \leq arccos\left(-\frac{L_2}{l}\right) \tag{3}$$