Homework 4 FSR

Antonio Manzoni P38000234

1 Exercise 1

A body submerged in a fluid under the effect of gravity is subjected to an hydrostatic effect called buoyancy. This can be expressed in formula as follows:

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

Where ρ is the density of the fluid, Δ is the volume of the submerged part of the body, while $\overline{\mathbf{g}} = \begin{bmatrix}0 & 0 & g\end{bmatrix}^T$ is the gravity acceleration expressed in vector form. The gravity force acts at the center of mass r_c^b (all the quantities are referred with respect to the Body frame) and is equal to:

$$f_g^b = R_b^T \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} \tag{2}$$

While the bouyancy force acts at the center of bouyancy r_b^b and is equal to:

$$f_b^b = -R_b^T \begin{bmatrix} 0 \\ 0 \\ b \end{bmatrix} \tag{3}$$

The minus sign is due the fact that this force tends to counteract the gravity effect pushing the body upwards. Now it is possible also to retrieve the wrench due the gravity and buoyancy in the body-fixed frame:

$$g_{rb}^b = -\begin{bmatrix} f_g^b + f_b^b \\ S(r_c^b)f_g^b + S(r_b^b)f_b^b \end{bmatrix} \tag{4}$$

A consideration to do is that in aerial robotics the bouyancy effect can be negligible because the density of the air is much lighter than the one of the robot; instead for underwater applications it is relevant because the density of the fluid is comparable to the density of the robot.

2 Exercise 2

- **a.** This statement is **false** because the added mass is an effect provoked due the fact that the robot starts to accelerate in a fluid with high density and so the particles of this fluid that are surrounding the system start to accelerate, generating a reaction force which is equal in magnitude but opposite in direction to the force generated by the robot's acceleration. So, at the end, inside the model we introduce another force that is added to the one generated by the mass of the robot times its acceleration.

- **b.** This statement is **true** because, for instance, if we consider the case of UAVs this effect is negligibile because the density of the air is much lighter than the density of the robot. So, we can say that in underwater applications, where the density of the fluid is comparable to the density of the robot, this effect must be considered.

- **c.** This statement is **true**, in fact the hydrodynamic damping improves the stability of a system by reducing the amplitude of oscillations or vibrations through the dissipation of energy, this occurs through fluid friction or viscous effects when an object moves through a fluid. Moreover, hydrodynamic damping generates forces that oppose the motion of the oscillating object so, these forces help maintain the system closer to its equilibrium position, reducing the likelihood of large deviations that can lead to instability. From a control theory point of view it is possible to consider only quadratic damping terms and group all these terms inside a damping matrix

$$D_{RB} \in \mathbb{R}^{6\times 6}$$

that is positive definite and usually its terms are constant. The assumption that this matrix is positive definite helps for the sketch that the time derivative of a Lyapunov function is negative semi-definite.

- **d.** This statement is **false** because it is possible to consider the Ocean current as constant and irrotational only if we express it in the World frame since in the Body frame time-varying and non-linear terms arise. So we can express this effect as follows:

$$\nu_c = \begin{bmatrix} \nu_{c,x} \\ \nu_{c,y} \\ \nu_{c,z} \\ 0 \\ 0 \\ 0 \end{bmatrix} \in \mathbb{R}^6, \quad \dot{\nu}_c = \mathbf{0}_6 \tag{5}$$

Where we consider only the first three components of the Ocean current and not the ones that are related to the angular part. Then, since a lot of measurements are not available in the World frame, we report these informations in the Body frame as follows:

$$\nu_r = \begin{bmatrix} \dot{p}_b^b \\ \omega_b^b \end{bmatrix} - R_b^T \nu_c \tag{6}$$

3 Exercise 3

The first request of the exercise is to implement the quadratic function using the QP solver qpSWIFT inside the `Main.m` file, in such way:

$$[\texttt{zval}, \texttt{basic\_info}, \texttt{adv\_info}] = \texttt{qpSWIFT}(\texttt{sparse}(H), g, \texttt{sparse}(Aeq), \texttt{beq}, \texttt{sparse}(Aineq), \texttt{bineq});$$

In particular, the matrices that we give to qpSWIFT as inputs represent the equality constraints(for dynamical consistency and contact constraint) and the inequality constraints (for non-sliding contact, torque limits and for the swing leg task) that the controller must satisfy, while the output zval contains the desired accelerations and ground reaction forces. The next step is to make various simulations. For all the gaits I'll show first the simulations that have been carried out with the nominal values that were already set inside the code and then varying some control or physical parameters. I've chosen to modify the following ones:

1. **velocity along x**: increasing it to 2m/s;
2. **mass**: decreasing it to 2kg and increasing to 10kg;
3. **friction coefficient**: decreasing it to 0.5 and 0.05;
4. **kp for the swing phase**: decreasing it to 10;

I've not reported all these changes for all the gaits because some of them produce results that are not so different from the ones obtained with the nominal values. All the videos and images are uploaded on my Github page.

- **Gait 0 (Trot)**

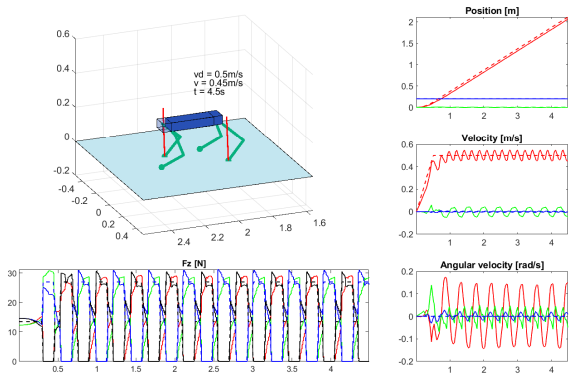


Figure 1: Gait 0 with nominal values: mass = 5.5kg, $velx_d$ = 0.5m/s, μ = 1, kp_{sw} = 300

This gait is characterized by the fact that the legs on the same diagonal move together. It is possible to notice that the robot has always either two or four legs in contact with the ground.

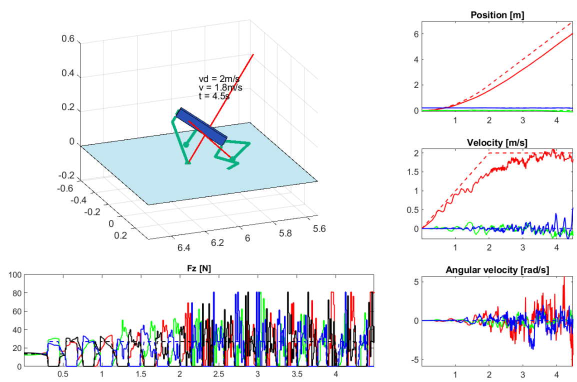


Figure 2: Gait 0 with nominal values and $velx_d$ = 2m/s

In this case the robot is not able to reach the desired velocity after 1 second as in the previous case, this produces huge errors on position and velocity that imply a bad motion.

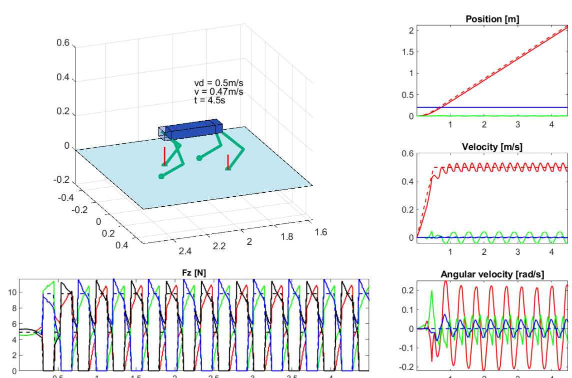


Figure 3: Gait 0 with nominal values and mass = 2kg

Decreasing the mass to 2kg improves a little bit the performances of the velocity and position and also decrease the Fz. Increasing the mass brings to have opposite results.

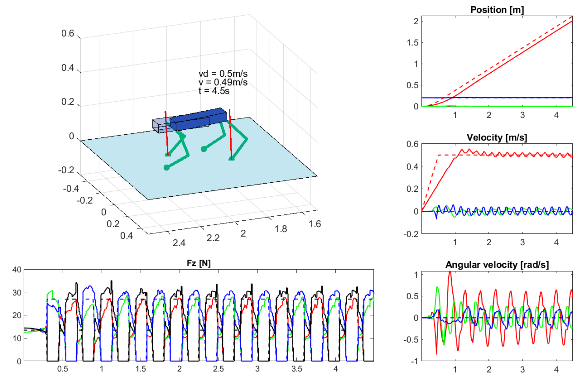


Figure 4: Gait 0 with nominal values and $\mu = 0.05$

Decreasing the friction coefficient to 0.05 tends to tilt the robot forward during the motion and, compared to the case with all nominal values, the robot presents greater difficulties to move during the first second.

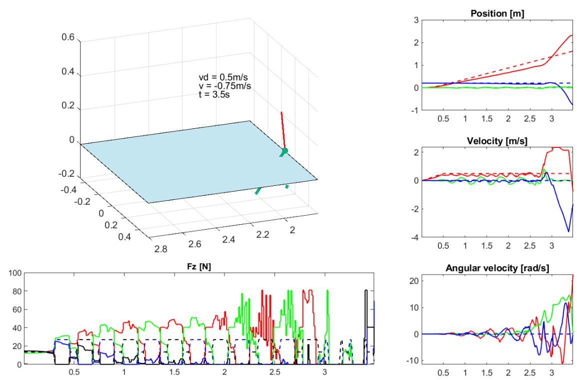


Figure 5: Gait 0 with nominal values and $kp_{sw} = 10$

With kp equal to 10 occurs a strange behavior in which the robot gets stucked inside the ground.

- **Gait 1 (Bound)**

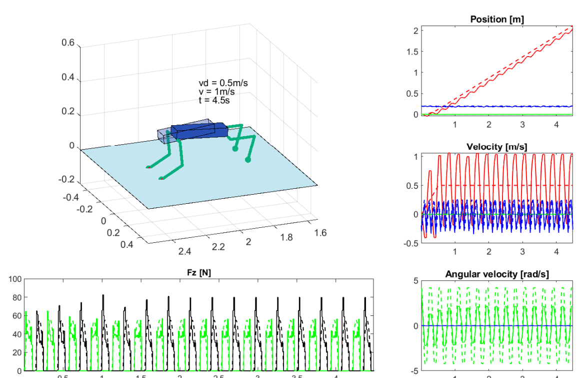


Figure 6: Gait 1 with nominal values: mass = 5.5kg, $velx_d = 0.5$m/s, $\mu = 1$, $kp_{sw} = 300$

This movement is similar to a jump in which the robot alternates the movement of the front legs with the one of the rear legs. In this case the robot has either zero or two legs touch the ground at the same time.

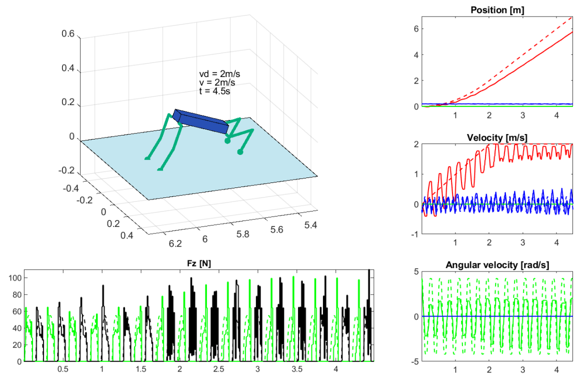


Figure 7: Gait 1 with nominal values and $velx_d$ = 2m/s

Bringing the velocity to 2m/s increases the position error and it is possible to see in the video "`gait1_v2`" that after 1.5 seconds the rear legs start to hit the terrain during the motion.

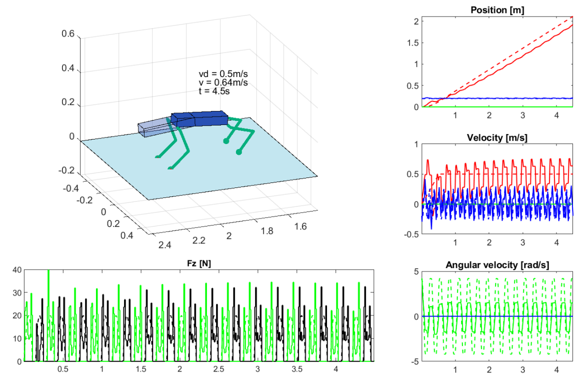


Figure 8: Gait 1 with nominal values and mass = 2kg

With a mass equal to 2kg the robot has shorter velocities peaks but the velocity profile presents some irregularities.

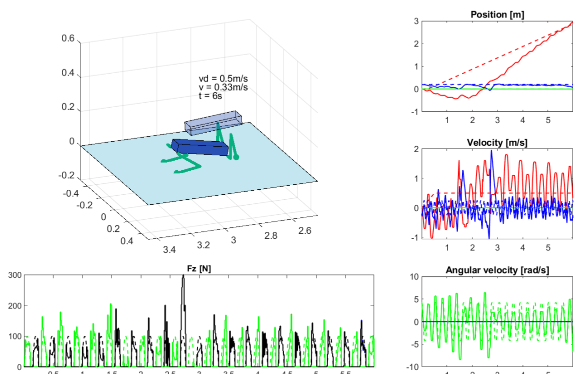


Figure 9: Gait 1 with nominal values and mass = 10kg

With a mass equal to 10 the robot falls down to the ground.

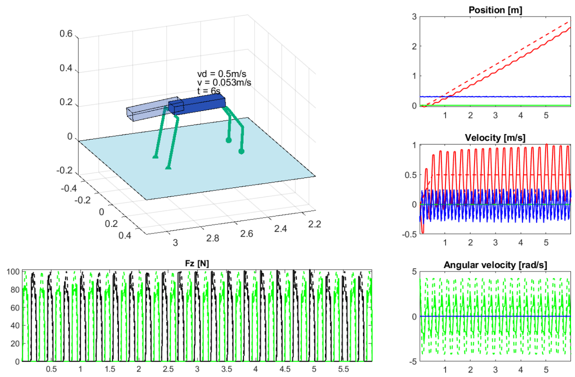


Figure 10: Gait 1 with nominal values, mass = 10kg and z0 = 0.3m

By lifting the center of mass by 0.1m the robot is able to perform the motion correctly.

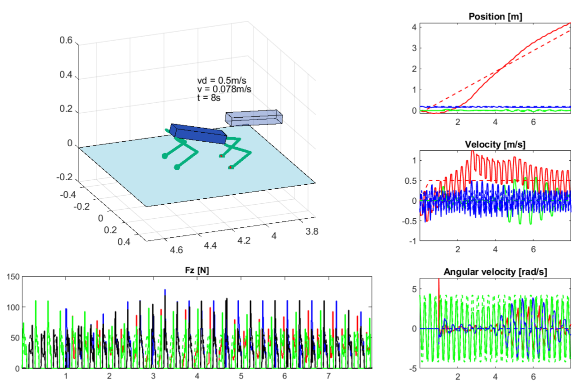


Figure 11: Gait 1 with nominal values and $\mu = 0.5$

Decreasing the friction coefficient to 0.5 causes some difficulties during the start where it is possible to see an huge position error, then there is a phase that goes from t= 2s to t= 6s in which the robot exceeds the desired value of velocity to compensate the position error and then another phase where the robot stabilizes itself.

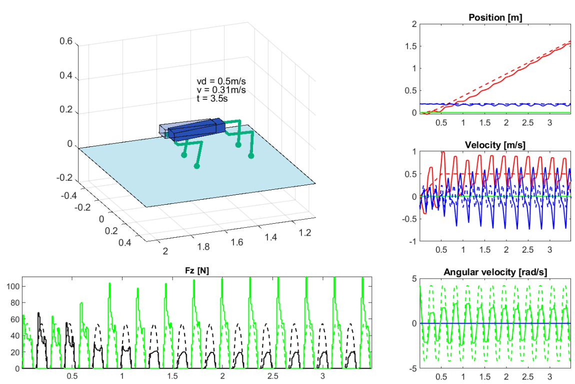


Figure 12: Gait 1 with nominal values and $kp_{sw} = 10$

Decreasing the factor kp to 10 for the swing phase produces a motion in which the robot tends to withdraw his front legs towards the body.

- **Gait 2 (Pacing)**

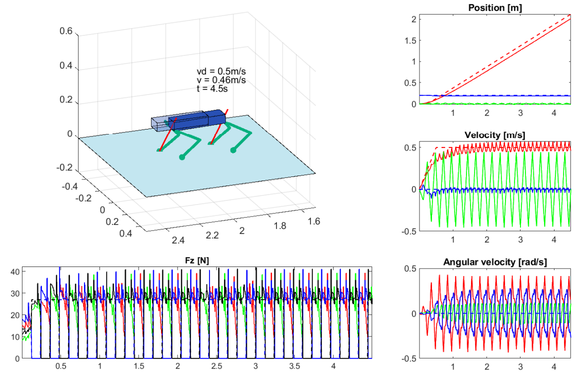


Figure 13: Gait 2 with nominal values: mass = 5.5kg, $velx_d$ = 0.5m/s, $\mu = 1$, $kp_{sw} = 300$

In this gait the legs on the same side move together. In this gait the robot has always two legs touching the ground.

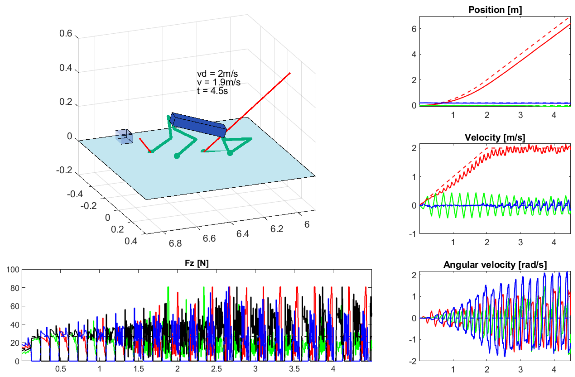


Figure 14: Gait 2 with nominal values and $velx_d$ = 2m/s

Increasing the the velocity to 2m/s brings the rear legs of the robot to hit the ground after 3s as it is possible to see in the video "`gait2_v2`".

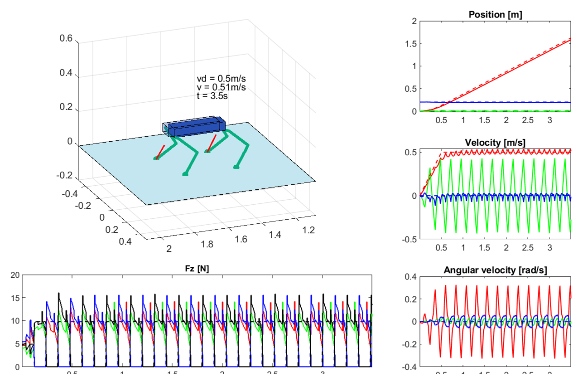


Figure 15: Gait 2 with nominal values and mass = 2kg

With a mass equal to 2kg the robot has a better behavior in general. Increasing the mass produces the opposite effects.

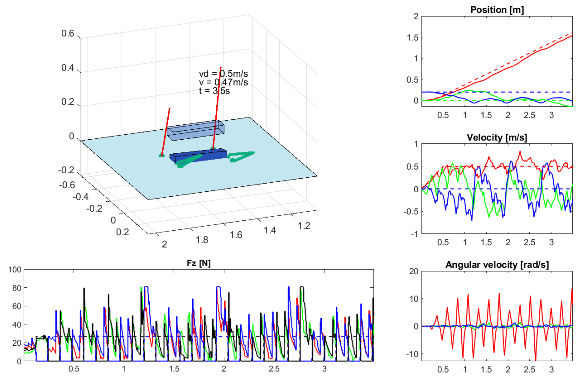


Figure 16: Gait 2 with nominal values and $\mu = 0.5$

Decreasing the friction coefficient to 0.5 brings the robot to dive inside the terrain.

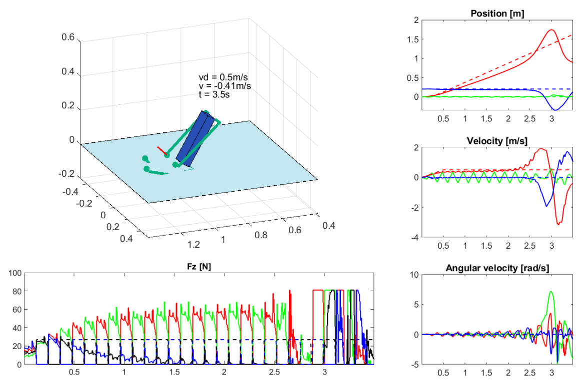


Figure 17: Gait 2 with nominal values and $kp_{sw} = 10$

With kp equal to 10 the robot falls down after 3s as it is possible to see in the video "`gait2_kp10`".

- **Gait 3 (Gallop)**

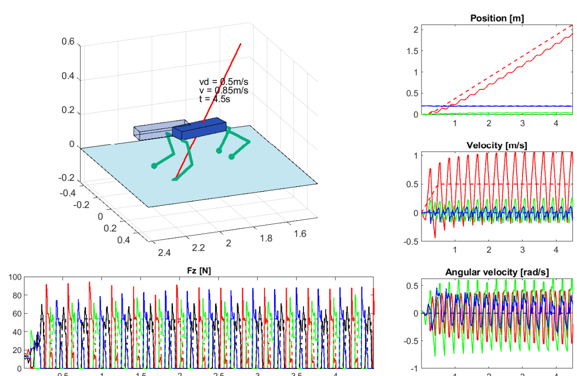


Figure 18: Gait 3 with nominal values: mass = 5.5kg, $velx_d = 0.5$m/s, $\mu = 1$, $kp_{sw} = 300$

This gait is similar to the previous one but in this case periodically one or two legs touch the ground. The first leg to move is the front left one, then the front right one, then the rear left one and finally the rear right one.

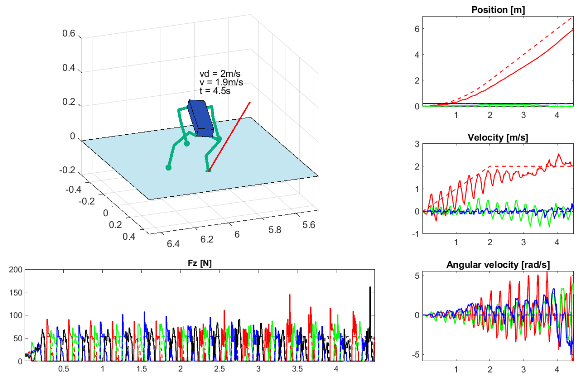


Figure 19: Gait 3 with nominal values and $velx_d = 2$m/s

Increasing the velocity to 2m/s takes the robot to walk while rotating on itself.

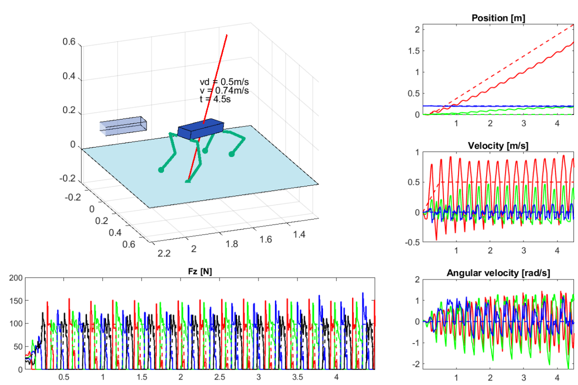


Figure 20: Gait 3 with nominal values and mass = 10kg

Increasing the mass worsens the general behavior of the gait, while decreasing the mass does not give substantial improvements.

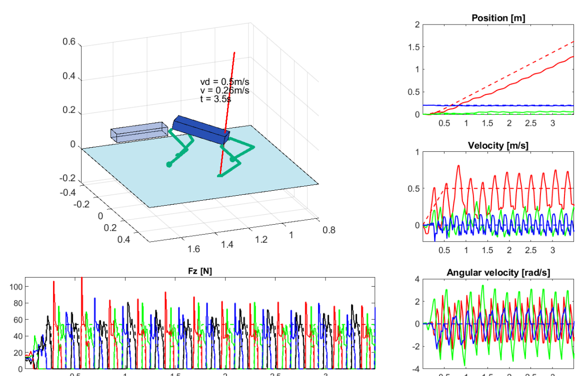


Figure 21: Gait 3 with nominal values and $\mu = 0.5$

Decreasing the friction coefficient to 0.5 produces shorter velocities peaks but the velocity profile presents some irregularities; also the position error and the angular velocity increase.

- **Gait 4 (Trot-run)**

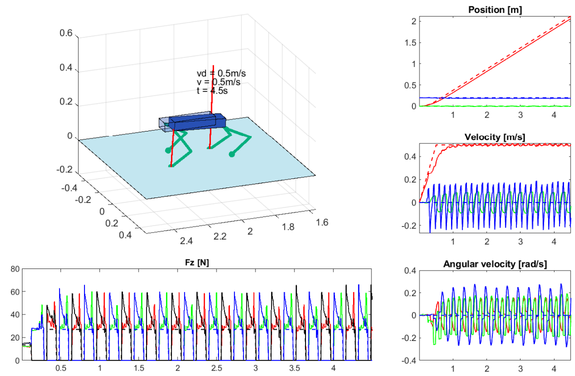


Figure 22: Gait 4 with nominal values: mass = 5.5kg, $velx_d$ = 0.5m/s, μ = 1, kp_{sw} = 300

This gait is similar to the first but in this case the robot has always only two legs in contact with the ground and a bouncy motion appears.

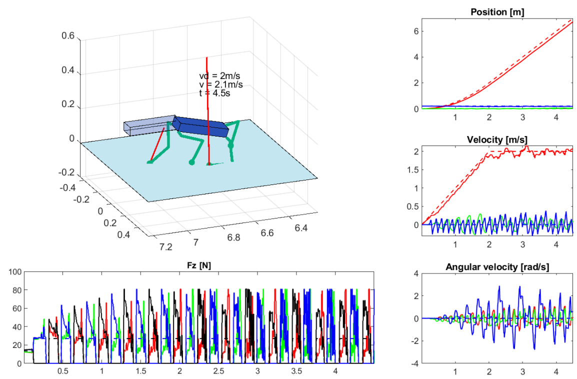


Figure 23: Gait 4 with nominal values and $velx_d$ = 2m/s

Increasing the the velocity to 2m/s brings the rear legs of the robot to hit the ground after 2s as it is possible to see in the video "`gait4_v2`".

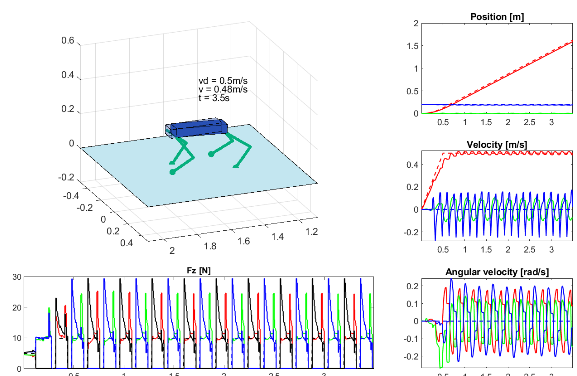


Figure 24: Gait 4 with nominal values and mass = 2kg

With a mass equal to 2kg the robot has a better behavior in general. Increasing the mass produces the opposite effects.

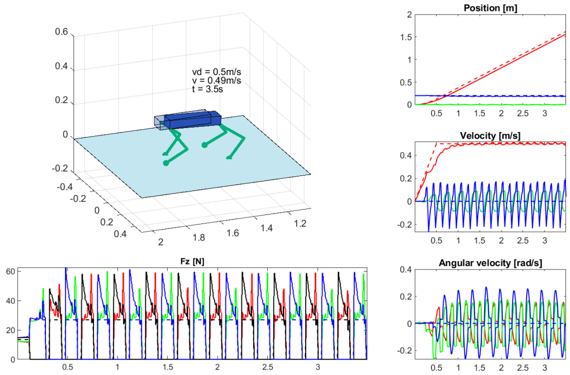


Figure 25: Gait 4 with nominal values and $\mu = 0.5$

Reducing the friction coefficient to 0.5 has the effect to bring the robot at the desired velocity after a greater amount of time compared to the case with nominal values.

- **Gait 5 (Crawl)**

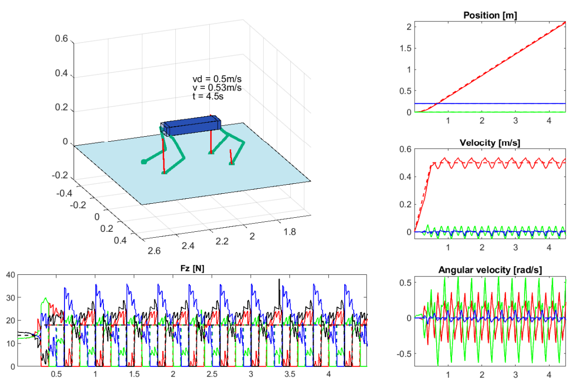


Figure 26: Gait 5 with nominal values: mass = 5.5kg, $velx_d$ = 0.5m/s, $\mu = 1$, $kp_{sw} = 300$

In this gate the robot moves one leg at time and in particular moves in order first the left front leg, then the right front leg, then the left rear and finally the right rear. It differs from the gait 3 because the motion does not produce bounces on the ground.

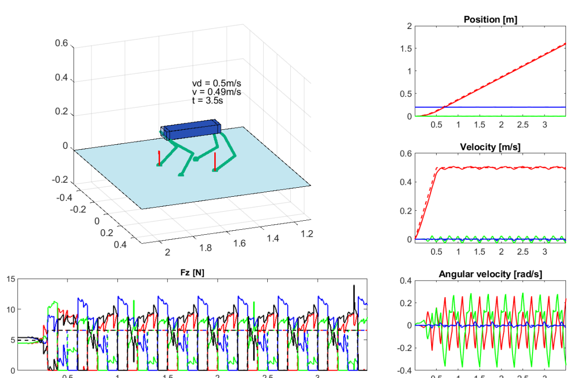


Figure 27: Gait 5 with nominal values and mass = 2kg

Decreasing the mass slightly reduce the velocity's peaks and the angular velocity.

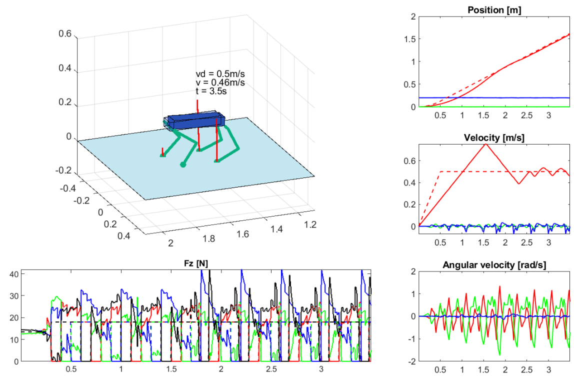
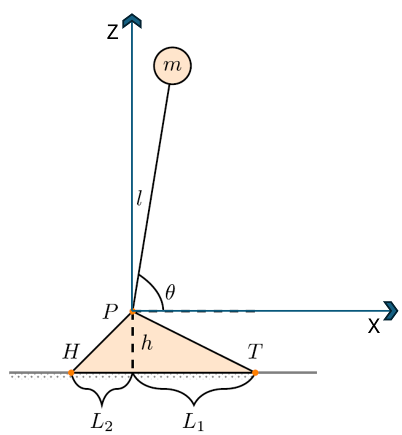


Figure 28: Gait 5 with nominal values and $\mu = 0.05$

Decreasing the friction coefficient has the effect to produce an evolution of the velocity which increase linearly until t= 1.5s and then decrease linearly until t=2.5s, then the behaviour settles down.

4 Exercise 4

The exercise requests to answer to some questions related to the system shown in figure:



- **a.** Without an actuator at point P the system is not stable at $\theta = (\pi/2) + \epsilon$ cause it is possible to associate the leg to an inverted pendulum where the mass is located above the center of rotation. Introducing a disturbance makes that the weight force tends to carry the mass away from the equilibrium point ($\theta = \pi/2$).
- **b.** As it is possible to see from the previous figure I've placed the origin of the reference frame in the point P, the x axis is parallel to the ground and points to the right and the z axis is perpendicular to the ground and pointed upwards. Now it is possible to compute the zero-momentum point as follows:

$$p_z^x = p_c^x - \frac{p_c^z}{\ddot{p}_c^z - g_0^z}(\ddot{p}_c^x - g_0^x) + \frac{1}{m(\ddot{p}_c^z - g_0^z)} S\dot{L}^y \tag{7}$$

Where:

1. $g_0 = [0 \quad 0 \quad -g]^T$ is the gravity vector;
2. S is the selection matrix for the angular momentum and it is equal to -1;
3. $\dot{L} = ml^2\ddot{\theta}$ is the time derivative of the angular momentum.
4. From the figure it is possible to retrieve p_c as follows:

$$p_c = (p_c^x, p_c^z) = (lcos(\theta), h + lsin(\theta)) \tag{8}$$

and from that it is possible to calculate the first and second derivatives:

$$\dot{p_c} = (\dot{p_c^x}, \dot{p_c^z}) = (-l\dot{\theta}sin(\theta), l\dot{\theta}cos(\theta)) \tag{9}$$

$$\ddot{p_c} = (\ddot{p_c^x}, \ddot{p_c^z}) = (-l\ddot{\theta}sin(\theta) - l\dot{\theta}^2cos(\theta), l\ddot{\theta}cos(\theta) - l\dot{\theta}^2sin(\theta)) \tag{10}$$

And finally:

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

- **c.** Now, supposing to have an actuator placed on P capable to cancel perfectly the torque ($\ddot{\theta} = 0, \dot{\theta} = 0$), the request is to find the value of θ that is possible to achieve without falling. Basically, the values of θ that are eligible to fulfill the request are the ones such that the vertical projection of the center of mass is contained inside the support polygon (that in this case is the segment $\overline{HT}$). So, we should satisfy this relation:

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

And from the last one it is possible to retrieve θ:

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