



UNIVERSITÀ<sup>DEGLI STUDI DI</sup>  
NAPOLI FEDERICO II

Scuola Politecnica e delle Scienze di Base  
Corso di Laurea Magistrale in Ingegneria dell'Automazione e Robotica

Field and Service Robotics

## ***HOMEWORK ag4in***

Academic Year 2024/2025

Relatore

**Ch.mo prof. Fabio Ruggiero**

Candidato

**Emmanuel Patellaro**

**P38000239**

Repository

[https://github.com/EmmanuelPat6/Field\\_and\\_Service\\_Robotics\\_Homework\\_4.git](https://github.com/EmmanuelPat6/Field_and_Service_Robotics_Homework_4.git)

## Exercise 1

Let us describe the **Buoyancy Effect**. It is an **Hydrostatic Effect**<sup>1</sup> (with the Gravity Effect), since it is valid even when the body is not moving.

In fact, when a rigid body is submerged in a fluid under the effect of gravity, two more forces must be considered: the *Gravitational Force* and the *Buoyancy*.

For this reason, basically, we consider two centers: the *Center of Gravity* and the *Center of Buoyancy* that might not coincide with the *Center of Mass* of the Robot given by its geometry.

It is possible to compute the buoyancy<sup>2</sup> as  $b = \rho\Delta\|\bar{g}\|$ , with the buoyancy force  $f_b^b$  acting at the center of buoyancy  $r_b^b \in \mathbb{R}^3$  and the wrench  $g_{rb}^b$  due to the gravity and buoyancy in the Body-Fixed Frame equal to

$$f_b^b = -R_b^T \begin{bmatrix} 0 \\ 0 \\ b \end{bmatrix} = -R_b^T \begin{bmatrix} 0 \\ 0 \\ \rho\Delta g \end{bmatrix} \quad g_{rb}^b = - \begin{bmatrix} f_g^b + f_b^b \\ S(r_c^b)f_g^b + S(r_b^b)f_b^b \end{bmatrix} \in \mathbb{R}^6$$

The *Dynamic Model of an Underwater Vehicle* is very similar to the one of an *Aerial Vehicle* that is very similar to the one of a *Rigid Body floating inside the 3D Space*.

The only thing to do is to add something that we have neglected with the Aerial Robotics. In fact, in the case under consideration, *the buoyancy effect is considered in Underwater Robotics, while it is neglected in Aerial Robotics*. This because of the *comparison* between the *density of the body* and the *density of the water* in the first case, and the *air* in the second one.

In fact **the density of the body is not comparable to the density of the air itself, which is much smaller with respect the density of the body**. Furthermore, **the density of the body, moving in the water, is comparable with the density of the water itself**, and so this effect cannot be disregarded. Water is approximately 830 times denser than air and, considering the buoyancy formula seen before, it is possible to observe that buoyancy is linearly dependent on  $\rho$ . So, smaller is  $\rho$ , smaller is this effect, and this is the reason why for a drone, the effect is neglected.

---

<sup>1</sup>Unlike the other effects like, for example, the Added Mass and Inertia, the Damping Effect or the Current Effect that are Hydrodynamic Effects

<sup>2</sup>With  $\Delta \in \mathbb{R}$ , volume of the body,  $\rho \in \mathbb{R}$ , density of the water and  $\bar{g} = [0 \ 0 \ g]^T \in \mathbb{R}^3$

---

## Exercise 2

Let us briefly justify whether the following expressions are true or false.

- a. "*The added mass effect considers an additional load to the structure*" is **FALSE**. This because the Added Mass is **NOT** a quantity of fluid to add to the system, such that the structure has a bigger mass. In fact, when an object moves inside a fluid, the particles of the fluid itself, surrounding the object, are accelerated. Because of this, a *Virtual Force* that acts on these particles is generated and it consists in a reaction force to the motion of the structure, equal in magnitude and opposite in the direction of motion. This reaction force is called **ADDED MASS CONTRIBUTION**<sup>3</sup>. So, basically, *it is something related to the Inertia of the system and it is as the object, during the motion, weighed more than its real weight*. Furthermore, it is not a friction force because this is not opposed to the velocity but, as mentioned, it is something at the *acceleration level*.
- 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*." is **TRUE**. In fact, this happens for the same reason related to the density of the water described in Exercise 1. As already seen, this effect cannot be neglected in water, whereas for aerial robots it can be neglected in air, due to the fact that the density of the air cannot be comparable to that of the drone.
- c. "*The damping effect helps in the stability analysis*" is **TRUE**. In fact, because of the viscosity of the fluid, dissipative drag and lift forces on the body are generated. The simplification of considering only quadratic damping terms has done, grouping all of them into a positive definite damping matrix  $D_{RB} \in \mathbb{R}^{6 \times 6}$  usually constant. So, since it is a positive definite matrix, *the damping effect accelerates the system convergence to the desired state*, as can be easily seen thanks to the *Lyapunov's Theory*. In fact, considering<sup>4</sup>

$$\dot{V} = -s_v^T(K_D + D_{RB})s_v$$

the added term  $-s_v^T D_{RB} s_v$ , it contributes to the *negativity* of  $\dot{V}$  more quickly. This behavior is fundamental for proving *asymptotic stability* and achieving *robustness*.

---

<sup>3</sup>The added mass matrix  $M_A \in \mathbb{R}^{6 \times 6}$  associated of this effect is not necessarily positive definite and there is also an Added Coriolis and Centripedal Contribution  $C_A \in \mathbb{R}^{6 \times 6}$

<sup>4</sup>Remember that the equations of motion of a UUV is  $M_V \begin{bmatrix} \ddot{p}_b^b \\ \ddot{\omega}_b^b \end{bmatrix} + (C_V + D_{RB}) \begin{bmatrix} \dot{p}_b^b \\ \dot{\omega}_b^b \end{bmatrix} + g_{rb}^b = \tau_v$

---

d. "*The Ocean current is usually considered as constant, and it is better to refer it with respect to the body frame*" is **FALSE**. This is because the Ocean current is considered *constant* and *irrotational*, but it is expressed in the *World Frame*. In practice, the Ocean current is modeled as a velocity contribution that it is possible to measure in the World Frame, with three components and zero acceleration such that

$$v_c = [v_{c,x} \ v_{c,y} \ v_{c,z} \ 0 \ 0 \ 0]^T \in \mathbb{R}^6, \quad \dot{v}_c = \mathbf{0}_6$$

So, at the end, the related effect can be added to the dynamic model of the underwater robot simply considering the relative velocity in the body-fixed frame<sup>5</sup>

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

Thanks to this, the model is much simple to handle.

## Exercise 3

Let us consider the Matlab files within the *quadruped\_simulation.zip* file.

The aim of the **Whole-Body Controller** is to control the entire body in such a way to satisfy the entire task, decoupling the motion planning from the control. Let us focus now on the **Quadratic Problem**<sup>6</sup>, which is also an addition to robustness for the control, necessary to satisfy several constraints very important for Legged Robots, such as *dynamic consistency*, *non-sliding contact*, *torque limits* and *swing legs task*.

The control inputs are the *accelerations of the joints* and the *ground reaction forces (GRF)*, which desired values are obtained through this QP<sup>7</sup>

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

---

<sup>5</sup>This is done thanks to the transpose of the rotation matrix  $R_b$

<sup>6</sup>Other important blocks of this controller are: Motion Planner, Wrench Reference, Actuation Torques, Foot Scheduler and Momentum-Based Observer

<sup>7</sup>It is implemented in MATLAB thanks to *qpSwift*

---

with<sup>8</sup>  $\zeta = \begin{bmatrix} \ddot{r}_c^T & \ddot{q}_j^T & f_{gr}^T \end{bmatrix}^T \in \mathbb{R}^{n_b+n_j+3n_{st}}$  and

$$f(\zeta) = \|\bar{J}_{st,c}^T \Sigma \zeta + \bar{J}_{st,c}^T \hat{f}_{st} - W_{CoM,des}\| + \|\zeta\|$$

with  $\Sigma \in \mathbb{R}^{n_b+n_j+3n_{st}}$  in order to select the last  $3n_{st}$  elements of  $\zeta$ , the GRF, in order to track the desired wrench. It is possible to have 6 different types of gait:

- **Trot:** a diagonal gait where the robot moves diagonal leg pairs in a cross configuration. It is stable and efficient at moderate speeds
- **Bound:** a gait where the front legs move together, followed by the rear legs moving together. It resembles a leaping motion and is used for faster, dynamic movement
- **Pacing:** the legs on the same side (e.g., front left and rear left) move together. It provides lateral stability and it is sometimes used on flat terrain
- **Gallop:** a high-speed gait involving an asymmetrical leg sequence, each foot is in contact with the ground one at a time, alternately. It is less stable, but very fast.
- **Trot Run:** a faster variation of the trot, maintaining the diagonal leg movement pattern but with a more dynamic and extended stride
- **Crawl:** a slow, stable gait where only one leg moves at a time, ensuring maximum ground contact and balance, suitable for rough terrain or precise tasks

**LINKS TO ALL VIDEOS RELATED TO EACH SIMULATION (AND ADDITIONAL ONES) WILL BE INCLUDED AT THE END OF THIS REPORT IN THE 'VIDEO REFERENCES' SECTION.**

Let us show the final general plots for each gait with the **Initial Parameters** given in the MATLAB Script.

---

<sup>8</sup> $r_c = [p_c^T \quad \eta_c^T]^T$ . Adding this new State-Variables the acceleration of the CoM will be minimized

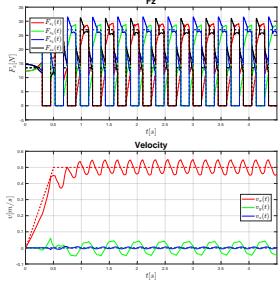


Figure 1: Trot Gait  $v_d = 0.5 \text{ m/s}$ ,  $\mu = 1$  and  $m = 5.5 \text{ Kg}$

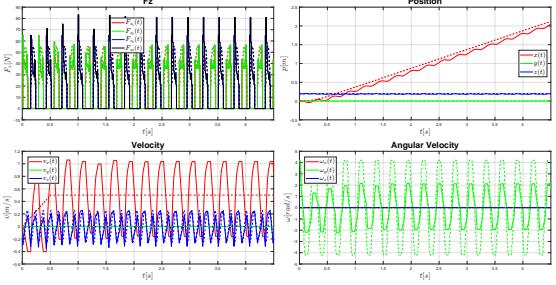


Figure 2: Bound Gait  $v_d = 0.5 \text{ m/s}$ ,  $\mu = 1$  and  $m = 5.5 \text{ Kg}$

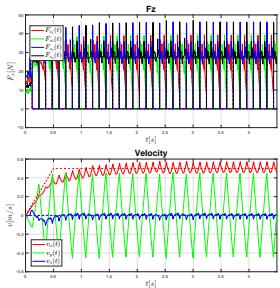


Figure 3: Pacing Gait  $v_d = 0.5 \text{ m/s}$ ,  $\mu = 1$  and  $m = 5.5 \text{ Kg}$

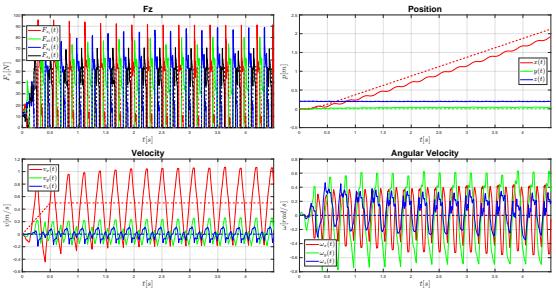


Figure 4: Gallop Gait  $v_d = 0.5 \text{ m/s}$ ,  $\mu = 1$  and  $m = 5.5 \text{ Kg}$

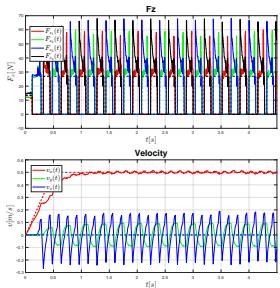


Figure 5: Trot Run Gait  $v_d = 0.5 \text{ m/s}$ ,  $\mu = 1$  and  $m = 5.5 \text{ Kg}$

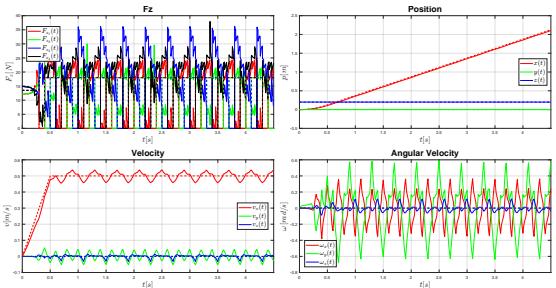


Figure 6: Crawl Gait  $v_d = 0.5 \text{ m/s}$ ,  $\mu = 1$  and  $m = 5.5 \text{ Kg}$

As can be seen from the plots and the videos, for each gait, the robot can exhibit better or worse behaviors and results.

With the initial parameters mentioned earlier, *excellent performance* can be observed for both the **Trot Gait** and the **Crawl Gait**, as they are the easiest to execute and offer *better stability*.

The **Trot Run Gait** also yields relatively *good results*. The decrease in accuracy regarding the tracking of the desired position compared to the Trot Gait is due to the fact that, in the run version, the robot never places all four feet on the ground at once. Obviously, as can be easily inferred from the name, there is a

*better velocity tracking*, due to the same reason mentioned above, and so a better response in terms of *dynamic performance*. However, this also leads to *higher  $F_z$*  values (approximately double).

*Fairly good results* are instead obtained with the **Pacing Gait**, although it shows *higher velocity values along the  $y$ -axis* compared to previous cases. This is due to the fact that two legs on the same side move alternately, causing a slight *lateral "wobble"* of the quadruped robot.

*Very poor results* are instead observed with the other two gaits: the **Bound Gait** and the **Gallop Gait**. The reason lies in the *more complex dynamics* and *lower stability* of the robot caused by the more demanding motions required by these types of gaits.

Especially in the first case (**Bound Gait**), the system shows an *almost good position tracking*, with an *oscillatory behavior* due to the motion characteristic of this type of gait, but exhibits a *poor velocity tracking*. The results show significant *oscillations around the desired velocity values* and *peak for the angular velocity* values much lower than required.

In the case of the **Gallop Gait**, the situation is quite similar, but with a *slightly worse position tracking performance*, also due to the rotation around the  $z$ -axis, with noticeable "*stall*" phases between steps. *Velocity oscillations* are even higher than in the *Bound Gait*. In both cases, the vertical ground reaction forces  $F_z$  are significantly *higher* than those observed in the previous gaits, reaching maximum values of up to 100 N.

Now let us change some parameters individually in order to verify how they affect the results obtained and the behavior of the quadruped robot, starting from the **velocity**  $v_d$

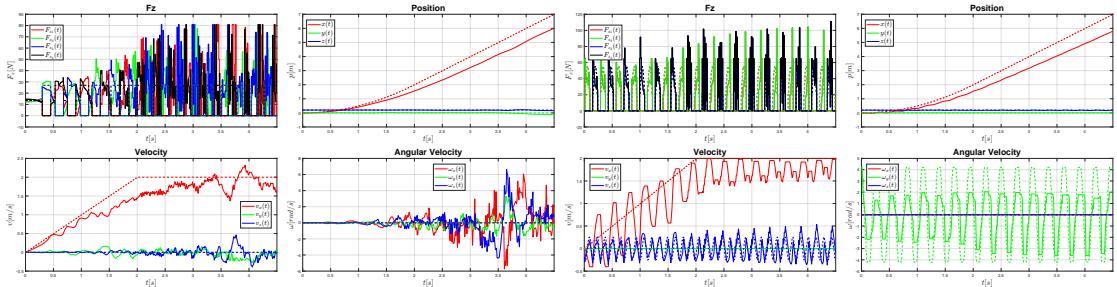


Figure 7: Trot Gait  $v_d = 2$  m/s,  
 $\mu = 1$  and  $m = 5.5$  Kg

Figure 8: Bound Gait  $v_d = 2$  m/s,  
 $\mu = 1$  and  $m = 5.5$  Kg

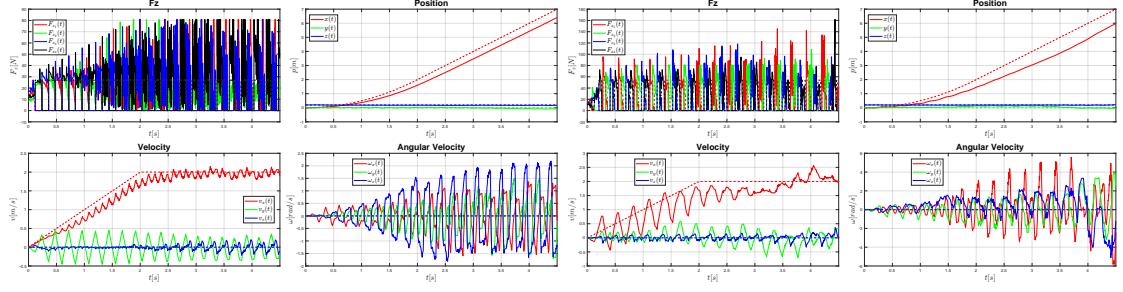


Figure 9: Pacing Gait  $v_d = 2$  m/s,  $\mu = 1$  and  $m = 5.5$  Kg

Figure 10: Gallop Gait  $v_d = 2$  m/s,  $\mu = 1$  and  $m = 5.5$  Kg

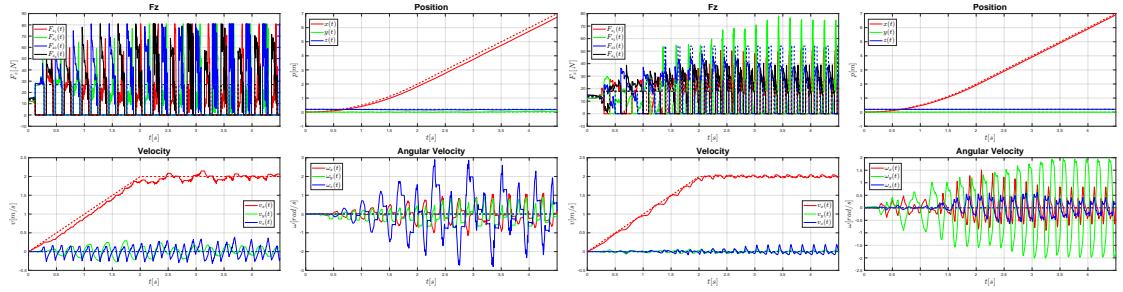


Figure 11: Trot Run Gait  $v_d = 2$  m/s,  $\mu = 1$  and  $m = 5.5$  Kg

Figure 12: Crawl Gait  $v_d = 2$  m/s,  $\mu = 1$  and  $m = 5.5$  Kg

As can be seen from the plots, **as velocities increase, performance tends to degrade**, both in terms of *tracking* the desired values and in terms of *stability*. As expected, *higher linear and angular velocities* are accompanied by increased *GRF* values.

Considering each gait, in most cases, the robot loses step and stability once a certain speed limit is reached.

A fairly good performance is observed with the **Trot Gait** and the **Bound Gait**, where the overall movement is satisfactory, but *position tracking deteriorates*, particularly toward the end of the simulation.

The **Pacing Gait** and **Trot Run Gait** both perform reasonably well; however, the robot tends to *tip* either *sideways* or *backward*.

A completely *incorrect behavior* is obtained with the **Gallop Gait**, where the quadruped robot ends up *rotating on itself* and completely *losing balance*.

On the other hand, the **Crawl Gait** exhibits *excellent behavior*, almost comparable to the case with lower speeds. This is obviously due to the *high stability* provided by this type of movement.

Instead, for *lower speed values*, the behaviors improve due to the reduced demands in terms of dynamic performance.

Now let us change the **mass**  $m$  of the quadruped robot.

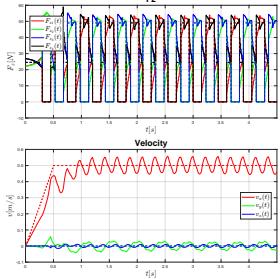


Figure 13: Trot Gait  $v_d = 0.5$  m/s,  
 $\mu = 1$  and  $m = 10$  Kg

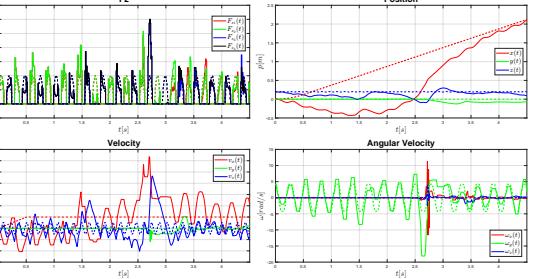
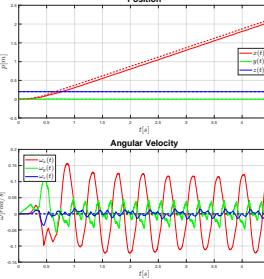


Figure 14: Bound Gait  $v_d = 0.5$  m/s,  
 $\mu = 1$  and  $m = 10$  Kg

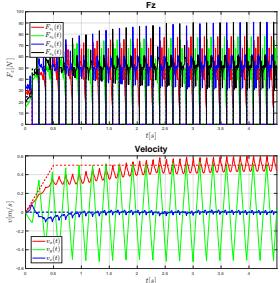


Figure 15: Pacing Gait  $v_d = 0.5$  m/s,  
 $\mu = 1$  and  $m = 10$  Kg

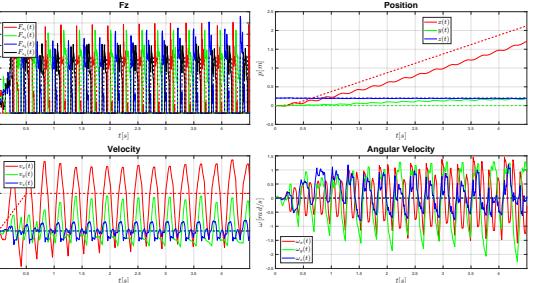
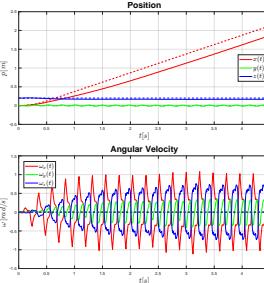


Figure 16: Gallop Gait  $v_d = 0.5$  m/s,  
 $\mu = 1$  and  $m = 10$  Kg

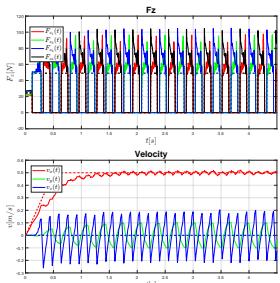


Figure 17: Trot Run Gait  $v_d = 0.5$  m/s,  
 $\mu = 1$  and  $m = 10$  Kg

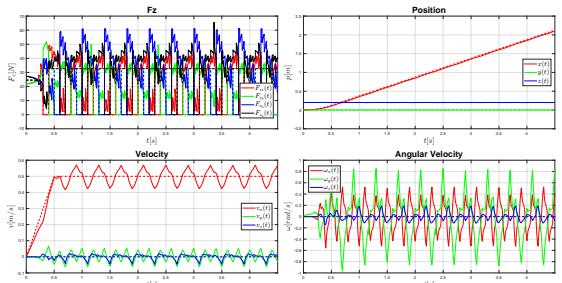


Figure 18: Crawl Gait  $v_d = 0.5$  m/s,  
 $\mu = 1$  and  $m = 10$  Kg

As can be seen from the plots provided, and as one might easily expect, depending on the gait considered, *the performance and results worsen when the mass increases, and improve when the mass decreases* (as the robot becomes easier to move). Mass values of 10 Kg and 2 Kg were considered, compared to the initial example of 5.5 Kg. For this reason, only the results obtained with the increased mass will be discussed.

First of all, as is easy to imagine, there is an *increase in the GRF proportional to the increase in the mass m* of the robot under examination.

As for the **Trot** and the **Trot Run Gaits**, they are very similar to the initial cases, and therefore they provide *good tracking* of the desired values (with obviously a better velocity tracking in the second case).

For gaits requiring "*jumps*" or *more complex dynamic movements*, such as the **Bound Gait**, the robot tends to *bend forward* during these motions due to the *increased weight*, eventually reaching a configuration that, in reality, would likely *prevent further movement and result in an irreversible fall*. This can be especially noticed by considering the *behavior of the variable z(t)*, which goes well below the desired value.

For the other gaits, *all previously observed effects are amplified to varying degrees*, such as the robot *rotating on itself* in the **Gallop Gait** or the *sideways imbalance* in the **Pacing Gait**.

An *optimal behavior* is, once again, obtained with the **Crawl Gait**, which, as always, proves to be *the best* in terms of both *stability* and *achieved results*.

To conclude, let us change the **friction coefficient**  $\mu$ . Only a lower value with respect the first one ( $\mu = 1$ ) is considered.

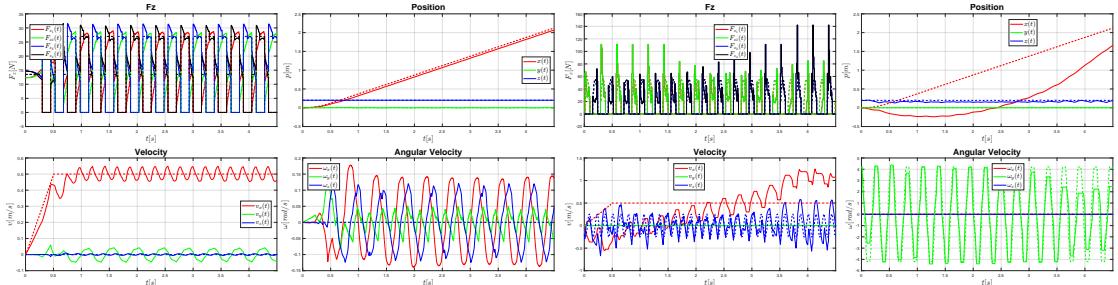


Figure 19: Trot Gait  $v_d = 0.5$  m/s,  
 $\mu = 0.3$  and  $m = 5.5$  Kg

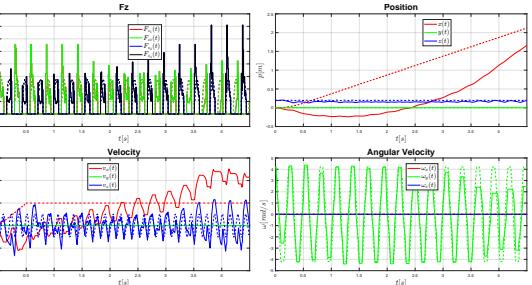


Figure 20: Bound Gait  $v_d = 0.5$  m/s,  
 $\mu = 0.3$  and  $m = 5.5$  Kg

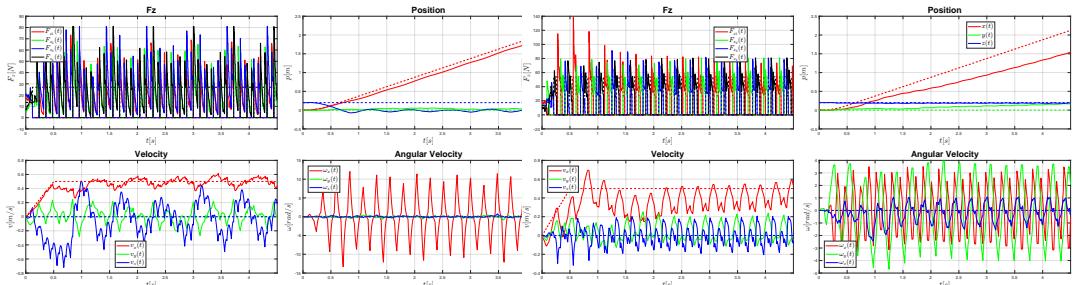


Figure 21: Pacing Gait  $v_d = 0.5$  m/s,  
 $\mu = 0.3$  and  $m = 5.5$  Kg

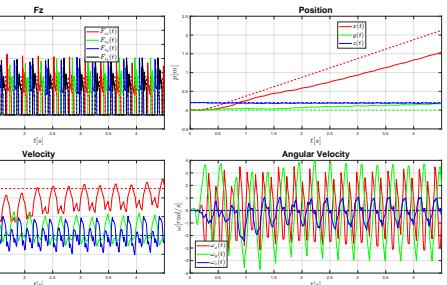


Figure 22: Gallop Gait  $v_d = 0.5$  m/s,  
 $\mu = 0.3$  and  $m = 5.5$  Kg

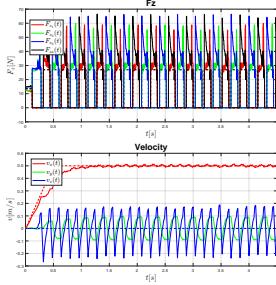


Figure 23: Trot Run Gait  $v_d = 0.5 \text{ m/s}$ ,  $\mu = 0.3$  and  $m = 5.5 \text{ Kg}$

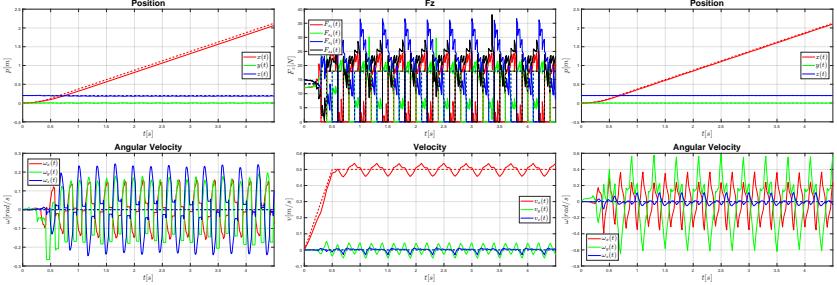


Figure 24: Crawl Gait  $v_d = 0.5 \text{ m/s}$ ,  $\mu = 0.3$  and  $m = 5.5 \text{ Kg}$

With a friction coefficient  $\mu = 0.3$ , the **Trot**, **Trot Run**, and **Crawl Gaits** continue to show *good behavior* and *excellent results*, both in terms of *velocity* and *position tracking*. For even lower values (simulations were also performed with  $\mu = 0.03$ ), some *performance degradation* can naturally occur, especially for the first two gaits mentioned.

As for the **Bound Gait**, there is initially a *strange behavior*, most likely due to the "*slipping*" caused by the *low friction coefficient*, where the robot tends to move backward after the first jump before starting to move forward in its intended direction. A pronounced *overshoot* can also be observed in the velocity (after an initial *undershoot* caused by the previously described effect), clearly indicating that once the robot begins moving, it takes more time to *regulate* its speed and *stabilize* at the desired value.

For the **Pacing Gait**, low friction coefficients make its implementation *infeasible*, as the robot tends to "fall" from the beginning due to the *alternating outward movement* of the four legs.

Very *poor results* are also obtained with the **Gallop Gait**, where the robot "*slides*" while rotating on itself, unsuccessfully attempting to regain *balance*.

## Exercise 4

Let us consider the **Rimless Wheel**. It is a simple mechanical model consisting of a central hub with evenly spaced spokes but no outer rim and it is often used in physics and robotics to study **passive-dynamic walking**. Considering some assumptions like the *Single Support Phase*<sup>9</sup>, the fact that the collisions with the ground are always *inelastic* and *impulsive* and the fact that the stance foot acts as a pin joint and does *not slip*, it is possible to consider the very simple *Stance*

<sup>9</sup>There is always only one leg in contact with the ground and it is periodically changed

## Dynamics as the Pendulum Dynamics

$$\ddot{\theta} = \frac{g}{l} \sin(\theta)$$

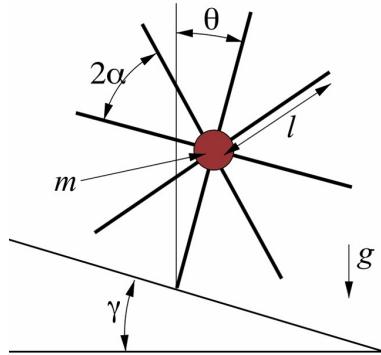


Figure 25: Rimless Wheel

Let us assume that the system is started in a configuration directly after a *transfer support*  $\theta(0^+) = \gamma - \alpha$  ( $\gamma + \alpha$  for negative value of  $\dot{\theta}_0$ ) and that  $0 \leq \gamma < \frac{\pi}{2}$ ,  $0 < \alpha < \frac{\pi}{2}$  and  $l > 0$ . Then the **forward walking** occurs when<sup>10</sup>  $\dot{\theta}(0^+) > \omega_1$ . There is the change of contact of the legs when  $\theta(t) = \gamma \pm \alpha$  and when the *potential energy* is converted into *kinetic energy* with velocity  $\dot{\theta}(t^-) = \sqrt{\dot{\theta}^2(0^+) + 4\frac{g}{l} \sin \alpha \sin \gamma}$ .

To begin with, we modify the **Initial Angular Velocity**  $\dot{\theta}_0$ , using both positive and negative values. As always, within the project, there are numerous other simulations in addition to those included in the following report. In this particular case, considering  $l = 1\text{m}$ ,  $\alpha = \frac{\pi}{8}\text{ rad}$  and  $\gamma = 0.08$ , it is possible to obtain  $\omega_1 = 0.9754\text{ rad/s}$ ,  $\gamma - \alpha = -0.3127\text{ rad}$  and  $\gamma + \alpha = 0.4727\text{ rad}$ .

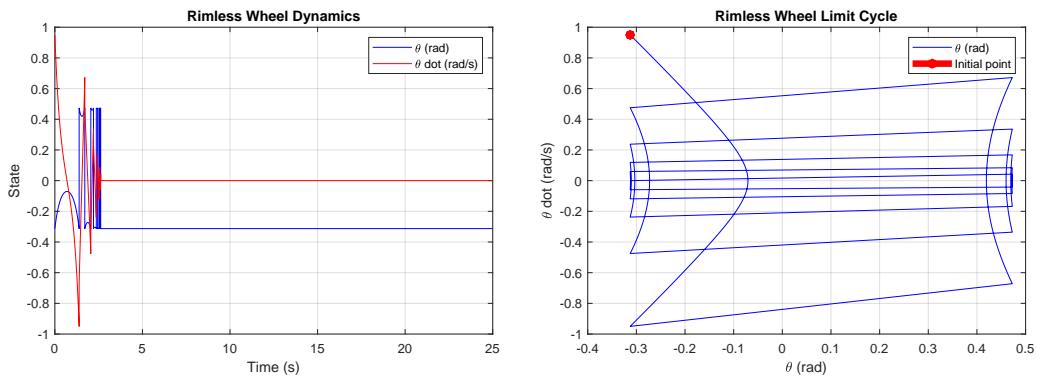


Figure 26:  $\dot{\theta}(0^+) = 0.95\text{ rad/s}$

<sup>10</sup> $\omega_1 = \sqrt{2\frac{g}{l}(1 - \cos(\gamma - \alpha))}$  is a threshold at which the system has enough Kinetic Energy to vault the mass over the stance leg and take a step

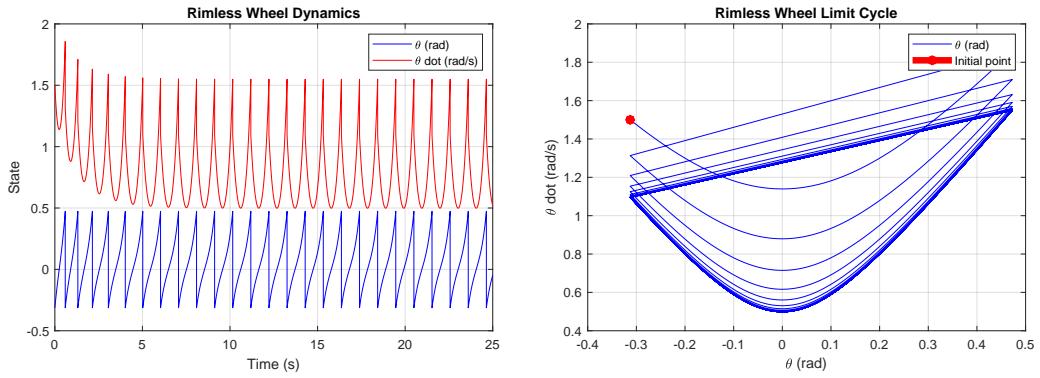


Figure 27:  $\dot{\theta}(0^+) = 1.5 \text{ rad/s}$

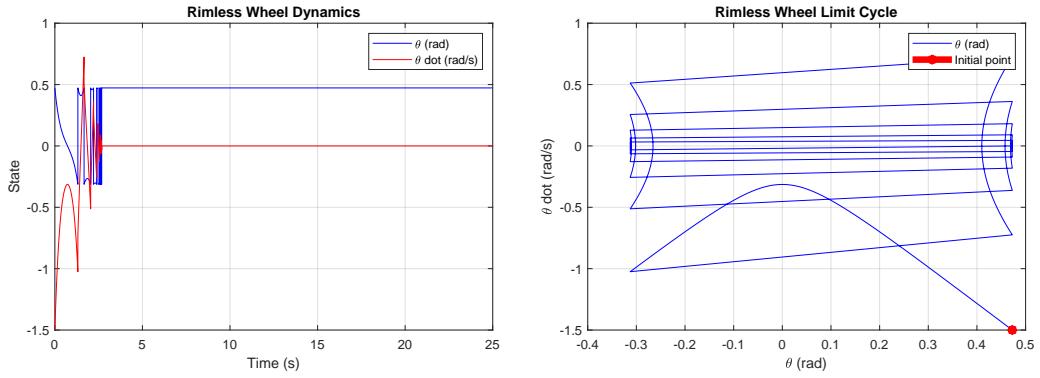


Figure 28:  $\dot{\theta}(0^+) = -1.5 \text{ rad/s}$

As can also be seen from the previous figures, once the wheel starts rotating, after a certain period it can either **run out of energy** and **stand still** or it quickly falls into a **stable periodic solution**, which is very similar to the *passive gait* for a human-like robot. In fact, when  $\dot{\theta}(0^+) > \omega_1$  as in Figure 27, the resulting phase portrait is a **Stable Limit Cycle** and the *Rimless Wheel Dynamics* is a *periodic solution* where the system *oscillates* between the two previously found  $\theta$  values.

In the other cases, as just mentioned, the solution **stabilizes** at one of the two values, with the *velocity* of course being *zero*. Here the phase portrait is the one, for example, in Figure 26, which shows a **convergence to the equilibrium point**, in this case,  $(-0.3127, 0)$  (the other one is  $(0.4727, 0)$ ).

It is possible to identify their respective **Basins of Attraction**<sup>11</sup>.

<sup>11</sup>Set of initial conditions that lead a system to evolve toward a particular stable state or attractor (like a limit cycle)

The Basin of Attraction for the **Limit Cycle** is

$$\left\{ (\theta_0, \dot{\theta}_0) \mid (\theta_0 = \gamma - \alpha \vee \theta_0 = \gamma + \alpha), \dot{\theta}_0 > \omega_1 \right\},$$

while the Basin of Attraction of the two **Equilibrium Points** is

$$\left\{ (\theta_0, \dot{\theta}_0) \mid (\theta_0 = \gamma - \alpha \vee \theta_0 = \gamma + \alpha), \dot{\theta}_0 \leq \omega_1 \right\}.$$

Now, to conclude, let us **modify the other parameters**, while keeping the initial angular velocity fixed at 0.95 rad/s.

Let us start by changing the **Leg Length  $l$** .

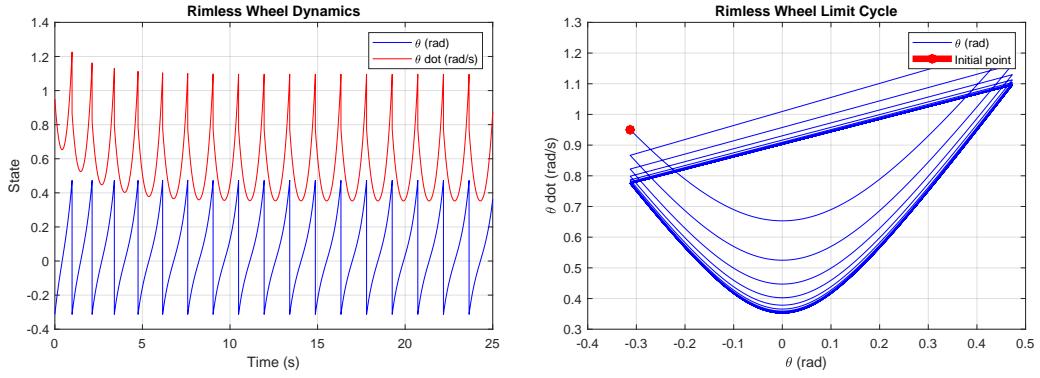


Figure 29:  $l = 2 \text{ m}$

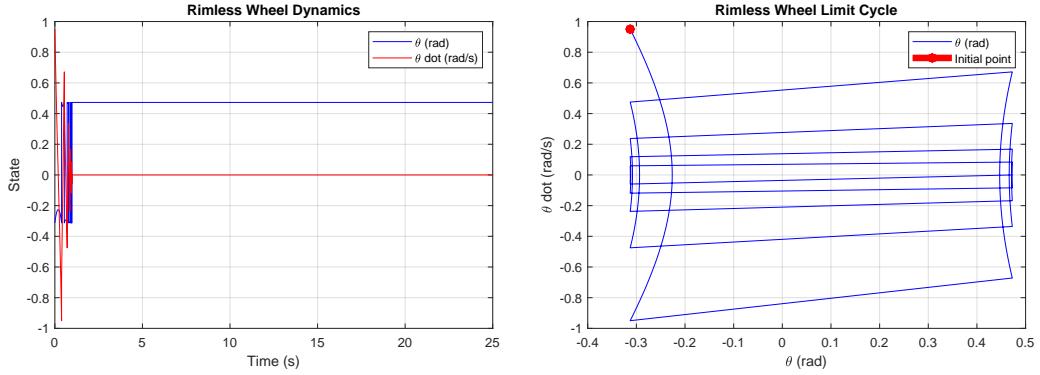


Figure 30:  $l = 0.5 \text{ m}$

By increasing the value of  $l$ , it can be seen from the formula shown in the footnote 10 that  $\omega_1$  decreases (in this case, for  $l = 2 \text{ m}$ ,  $\omega_1 = 0.6897 \text{ rad/s} < \dot{\theta}_0 = 0.95 \text{ rad/s}$ ). Therefore, *the greater the value of  $l$ , the lower the initial angular velocity required to obtain a periodic solution and a stable limit cycle*. The shape of the *Limit Cycle* obviously remains the same (since the values of  $\gamma$  and  $\alpha$  are

unchanged), with only the velocity values different.

For the same reason, *for smaller values of  $l$ ,  $\omega_1$  will be greater and, therefore, a higher initial angular velocity will be needed to obtain the previously mentioned stable periodic solution* (for  $l = 0.5$  m  $\omega_1 = 1.3794$  rad/s  $> \dot{\theta}_0 = 0.95$  rad/s). The difference compared to the initial case is a *faster convergence to the Equilibrium Point*.

Now, let us change the **Half Inter-Leg Angle**  $\alpha$ . By changing this parameter and inserting appropriate values, the *number of legs* of the wheel will also change.

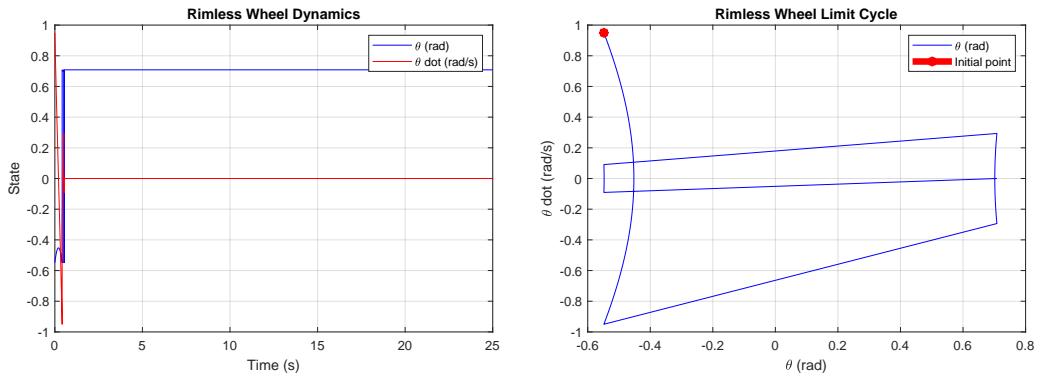


Figure 31:  $\alpha = \frac{\pi}{5}$  rad  $\rightarrow$  5 Legs

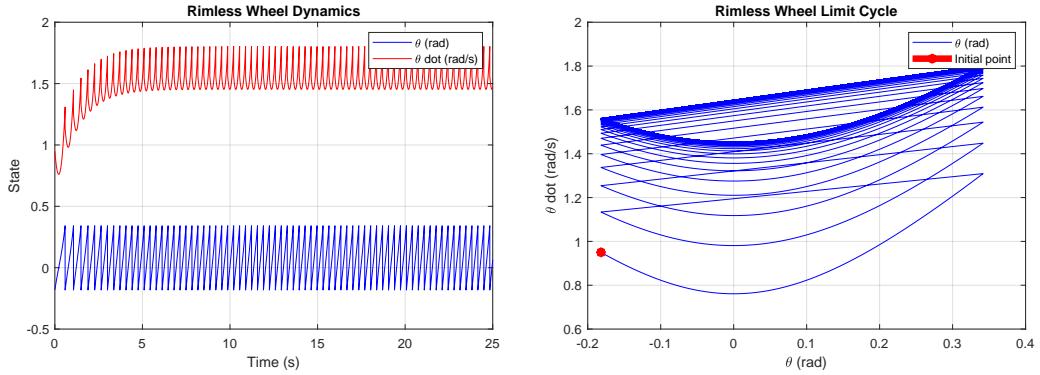


Figure 32:  $\alpha = \frac{\pi}{12}$  rad  $\rightarrow$  12 Legs

As can be seen from Figure 31, *increasing  $\alpha$  reduces the number of 'steps' taken by the robot due to the greater distance between each leg*. Moreover, as expected, much *higher values for  $\omega_1$  are obtained* (in this case, 1.6960 rad/s). On the contrary, *decreasing  $\alpha$  results in a higher number of legs and a shorter distance between them*. The evolution of  $\theta(t)$  obviously becomes much *faster*, and  $\omega_1$  *drops significantly*.

The more  $\alpha$  decreases, the *denser* the resulting *Limit Cycle* becomes but, basically,

the *shape* remains the same but with lower value of  $\gamma + \alpha$  and  $\gamma - \alpha$  (various examples are available in the GitHub directory linked on the first page).

To conclude, let us change the value of the **Slope Angle**  $\gamma$ .

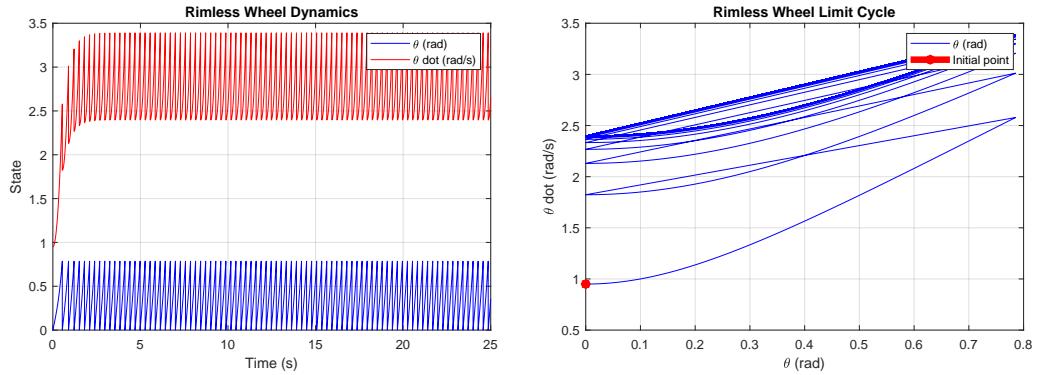


Figure 33:  $\gamma = \alpha = \frac{\pi}{8}$

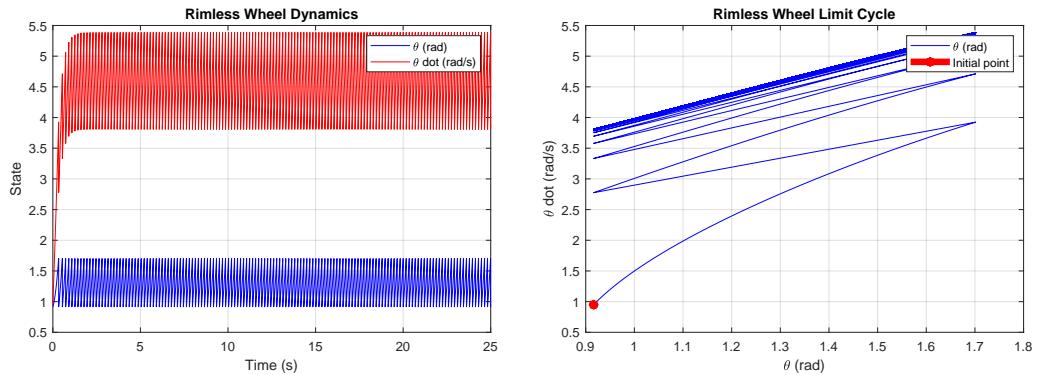


Figure 34:  $\gamma = \frac{5\pi}{12}$

As  $\gamma$  varies, many things change. For relatively *small values*, comparable to the initial value 0.08, the behavior remains almost the same (and for this reason it is not shown).

*The smaller this value is, the more the system will converge to one of the two Equilibrium Points*, even for high values of  $\dot{\theta}_0$ . For values of  $\gamma$  equal to  $\alpha$ , as can be easily verified, the value of  $\omega_1$  becomes 0, and a *Limit Cycle* very similar to the previous ones is obtained, which can be entered even with relatively *low initial angular velocity*  $\dot{\theta}_0$ .

The situation is different for  $\gamma > \alpha$ . As shown in the last figure,  $\omega_1$  increases because in this case *the mass is always ahead of the stance foot and the standing fixed point disappears*. Here, the *Limit Cycle* takes on a much *flatter shape* compared to the previous simulations, with higher values of both  $\gamma + \alpha$  and  $\gamma - \alpha$ .

---

## Exercise 3 Video References

**Initial Parameters:** <https://youtu.be/zsvwuidYBVA>

**$v_d = 0.1 \text{ m/s}$ :** <https://youtu.be/wfgPg33TG4I>

**$v_d = 1 \text{ m/s}$ :** <https://youtu.be/LdQIC4KqdFk>

**$v_d = 2 \text{ m/s}$ :** <https://youtu.be/lbGirsvQ-jg>

**$m = 2 \text{ Kg}$ :** <https://youtu.be/K7Yc1mpLM2Q>

**$m = 10 \text{ Kg}$ :** <https://youtu.be/Q2z1ytsvv74>

**$m = 20 \text{ Kg}$ :** [https://youtu.be/q\\_sDeBgh89Q](https://youtu.be/q_sDeBgh89Q)

**$\mu = 0.3$ :** <https://youtu.be/cx6c10DDQ4g>

**$\mu = 0.03$ :** <https://youtu.be/YjZ7hFpPIvI>