Universitat Politècnica de Catalunya
Facultat de Matemàtiques i Estadística

Master in Advanced Mathematics and Mathematical Engineering

Master's thesis

A Remotely-driven Hoverboard With Platform Leaning Control

Esteve Tarragó

Supervised by Advisor: Enric Celaya, Tutor: Mercè Ollè

September, 2019

I would first like to thank my thesis advisor Prof. Enric Celaya. Enric was always open whenever I ran into a trouble spot or had a question about my research or writing. He consistently allowed this thesis to be my own work, but steered me in the right the direction whenever he thought I needed it.

I would also like to thank Prof. Lluís Ros who was also advising me through the first steps of the design and always gave constructive feedback.

Also this project wouldn't have been possible without my lab co-workers. I would like to highlight three of them: Iñigo Moreno, Sergi Hernandez and Patrick Grosch. They taught me all kinds of stuff and I am very grateful for their time.

This research opportunity was possible thanks to IRI (Institut de Robtica i Informtica Industrial). They facilitated me a workstation, material and a fantastic workshop where to build my robot.

Abstract

This thesis is centered around the design, construction and control of a remotely controlled hoverboard. The main challenge of this project is the study of the leaning control of the hoverboard platform and the restrictions it induces.

Nowadays, segways are a popular way of human transportation and there also exist segway robots. Both, human-driven and autonomous segway robots have their speed determined by their inclination. (e.i. If you wish to go forward in a segway then you should put your weight in the same direction). So in these systems the inclination is not a degree of freedom but a necessary condition from moving. Unblocking this degree of freedom may help two-wheel robots perform new tasks as measurements, taking images or samples from other inclinations, avoiding obstacles, etc.

We have studied different mechanisms to control the leaning of the platform while allowing the movement of the robot. Then, we studied the dynamics of our system in order to determine its dimensions and create an optimal design accordingly. We have also run simulations with different policies to see how our system would evolve.

Finally, we built and programmed our robot so it can be remotely operated from anywhere.

Keywords

Dynamic, System, Control, Design

Contents

1 Introduction **4**

2 Initial design considerations **6**

2.1 Flywheel design . . . 6

3 Optimization Setup **10**

3.1 Restrictions . . . 10

3.2 Requirements . . . 11

3.3 Cost function . . . 11

4 Mechanical analysis **12**

4.1 Inclination control . . . 12

4.2 Wheels torque . . . 12

4.3 Flywheel torque . . . 13

4.4 Maximum speed, acceleration and terrain inclination . . . 14

4.4.1 Speed and acceleration with no terrain inclination ($\alpha = 0$) . . . 15

4.4.2 Terrain inclination ($\alpha > 0$) . . . 18

5 Optimization Results **19**

6 Rectilinear movement dynamics with $r_{flywheel}$ fixed **24**

6.1 Simulations . . . 25

6.1.1 Controlling the platform inclination . . . 26

6.1.2 Letting the platform turn . . . 29

7 Flywheel brake study **32**

7.1 System of differential equations . . . 32

7.2 Results . . . 33

8 Rectilinear movement with $r_{flywheel}$ free **36**

9 Components **38**

9.1 Mechanical components . . . 38

9.1.1 Central Body . . . 38

9.1.2 Lateral Body . . . 38

9.1.3 Wheels . . . 39

9.1.4 Flywheel . . . 39

9.1.5 Bearings . . . 40

9.1.6 Reinforcements . . . 40
9.2 Electronic components . . . 40
9.2.1 Raspberry Pi . . . 40
9.2.2 Batteries . . . 41
9.2.3 Printed Circuit Board . . . 41
9.2.4 DC Motor . . . 42
9.2.5 H Bridge . . . 43
9.2.6 Rotatory Encoder . . . 43
9.2.7 Accelerometer . . . 44

10 Control **45**
10.1 Position control . . . 45
10.2 Speed control . . . 46
10.3 Inclination control . . . 47

11 Software Design **48**
11.1 User Interface . . . 48
11.2 Planner . . . 49
11.3 Controller . . . 49
11.3.1 LED . . . 49
11.3.2 Motor . . . 49
11.3.3 Accelerometer . . . 49
11.3.4 Stabilizer . . . 49

12 Conclusion **50**

Figure 1: Picture of Tibi and Dabo, *two segway robots*

1. Introduction

In the IRI lab we have two segway robots, Tibi and Dabo shown in figure 1. Both of them move the same way. They must incline their body to command the wheels' rotation. We will get deeper in why these phenomena happens in section 4.

This movement restriction may cause some problems when trying to avoid obstacles, as for example going through low roof path. Another limitation is the acceleration that the robot can achieve which is directly related with the inclination that it's limited to 90 degrees in the best case. Also, the amount of uphill is limited with an additional problem: the safety system stops the robot if it detects something near it (in this case the ground).

So we decided to build a prototype of segway robot that could solve this problem by controlling its inclination independently. The chosen robot is inspired in a *segway hoverboard*, similar to the one appearing in Figure 2. The two wheels are controlled with classic control algorithms and the inclination of the central body is controlled with a flywheel mechanism that we will discuss in section 2.1.

Figure 2: Picture of a commercial *segway hoverboard*

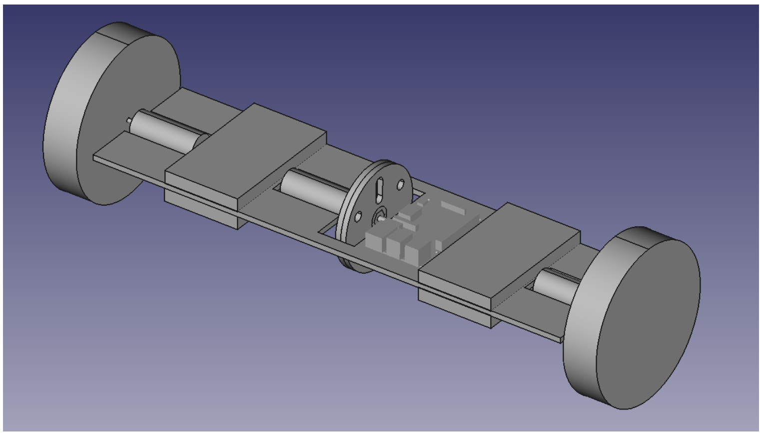


Figure 3: Isometric render view

2. Initial design considerations

The first thing we decided was the number of actuators. Most of segway robots include two motors for the motion control, but we added a third on in order to control the inclination. So we ended up with three motors in total because we want to control three degrees of freedom (inclination and speed of both wheels).

In order to control the inclination of the platform we needed to produce an external torque to the platform. We considered three methods: accelerating a flywheel, holding a pendulum in a non-vertical position and air friction with a fan. We discarded the last one due to the high speeds we needed to obtain a reasonable torque on the platform.

Both methods have strengths in different situations, so we decided to build a mixed piece that could combine both. The flywheel mode allows to deliver the maximum torque from the beginning but fails to deliver a continuous torque due to the motor achieving max speed. In the other hand the pendulum allows the system to apply a constant amount of torque over time.

We took two more restriction in our design. The first one symmetry along the motors/inclination axis in order to have an equilibrium in all possible inclinations without the need of external forces. We also took in consideration that some experiments may start produce arbitrary rotations of the platform, so none of the configurations should touch the ground to avoid crashes. Figure 6 illustrates this last restriction.

The design of the robot was done with the 3D design software Free-cad and most of the parts were 3D printed. 3D printing has also its own restrictions, for example not being able to print parts bigger than 25 cm. All part files are uploaded to the GitHub repository `https://github.com/tarragoesteve/TFM` under the hardware folder. So anyone can build this robot.

You can see the main views of an initial design on Figure 3, 4, 5 and 6.

2.1 Flywheel design

To control the inclination of the body two strategies are taken into account. Creating torque by a pendulum or accelerating the flywheel. In order to experiment with both of them we designed a part to allow both configurations by placing weights in different spots, see figure 7.

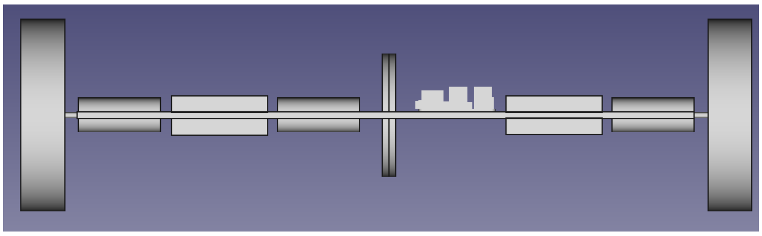


Figure 4: Front render view

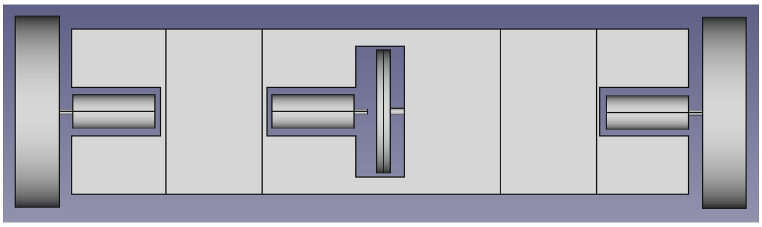


Figure 5: Top render view.

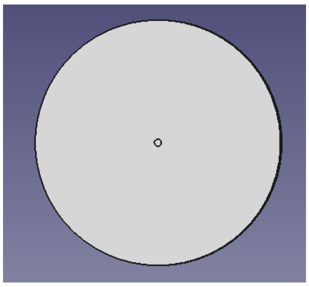


Figure 6: Side render view.

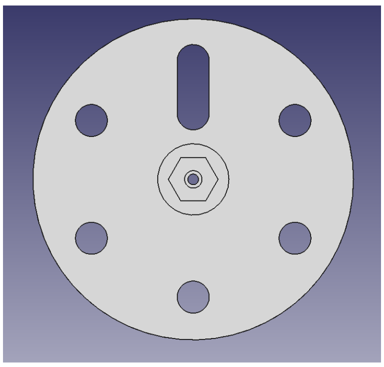


Figure 7: Flywheel side render view.

In order to create a configuration with maximum pendulum torque we have done the following computation. We denote the pendulum torque by τ, consider the masses are cylinders of mass $m_{cylinder}$ with radius $r_{cylinder}$ and width w and the radius of the flywheel is $r_{flywheel}$. We want to place all N masses at the same distance of the center of the flywheel (r_{max}) in a regular polygon except for one mass that will be at r_{min}. See figure 7.

Each mass weights:

$$m_{cylinder} = \rho \cdot w \cdot \pi \cdot r_{cylinder}^2$$

We neglect the mass of the flywheel structure versus the mass of the cylinders. All the gravitational torque is created by the cylinder masses and all of them are compensated with the opposite weight except for the two masses with different radius.

One of the weights can be placed along a track. The distance to the center will vary from $r_{min} = r_{cylinder} + r_{motor-axis} \approx r_{cylinder}$ to $r_{max} = r_{flywheel} - r_{cylinder}$.

The maximum torque takes place when these two masses are aligned horizontal with respect to the ground and the movable weight is at distance r_{min} from the center.

$$\tau_{max}(r_{cylinder}) = m_{cylinder} \cdot g \cdot r_{max} - m_{cylinder} \cdot g \cdot r_{min} = m_{cylinder} \cdot g \cdot (r_{flywheel} - 2 \cdot r_{cylinder})$$

In order find the radius r_c that maximizes τ we first compute the derivative:

$$\frac{\partial \tau_{max}(r_{cylinder})}{\partial r_c} = g \cdot (\frac{\partial m}{\partial r_c} \cdot (r_f - 2 \cdot r_c) - m_{cylinder} \cdot 2)$$

$$\frac{\partial m_{cylinder}}{\partial r_{cylinder}} = 2 \cdot \rho \cdot w \cdot \pi \cdot r_{cylinder}$$

And make it zero to find the maximum:

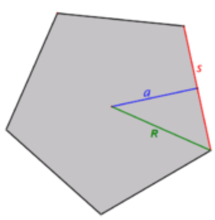
$$\frac{\partial \tau_{max}(r_{cylinder})}{\partial r_{cylinder}} = 0$$

Substituting and simplifying we get:

$$\frac{\partial m}{\partial r_{cylinder}} \cdot (r_{flywheel} - 2 \cdot r_{cylinder}) = m_{cylinder} \cdot 2 \Rightarrow 2 \cdot \rho \cdot w \cdot \pi \cdot r_{cylinder} \cdot (r_{flywheel} - 2 \cdot r_{cylinder}) = \rho \cdot w \cdot \pi \cdot r_c^2 \cdot 2$$

$$\Rightarrow r_c \cdot (r_{flywheel} - 2 \cdot r_{cylinder}) = r_{cylinder}^2 \Rightarrow (r_{flywheel} - 2 \cdot r_{cylinder}) = r_{cylinder} \Rightarrow \boxed{r_f = 3 \cdot r_{cylinder}}$$

The circumradius R from the center of a regular polygon to one of the vertices is related to the side length s by:



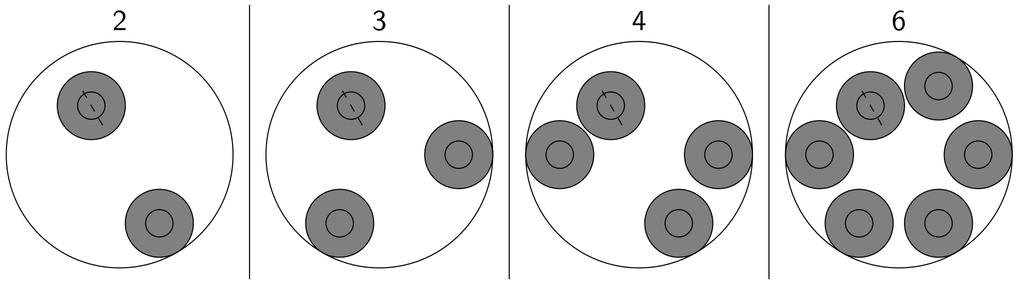
$$R = \frac{s}{2 \cdot \sin \frac{\pi}{n}}$$

In our case:

$$R = r_{flywheel} - r_{cylinder};$$

$$s = 2 \cdot r_{cylinder}$$

Substituting in the circumradius equation we get n = 6, so we will use up to 6 masses in our flywheel. We will have a variable number of masses N that we will be able to add to the flywheel as shown in the following table:



3. Optimization Setup

In this section we will set up the requirements and the cost function we want to optimize.

3.1 Restrictions

The restrictions are a list of inequalities that our system has to fulfill. The first restriction is due to the initial design requirements. Finally, the other three are somehow arbitrary but will help us to reduce the size of the robot.

1. We will place our flywheel in a hole on our robot. We don't want to touch the ground in any configuration so:

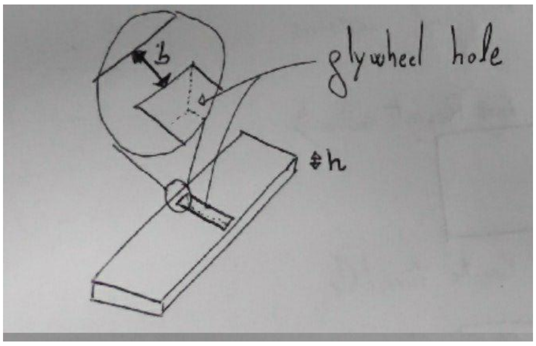


Figure 8: Flywheel hole diagram

$$r_{wheel} > \sqrt{(r_{flywheel} + b)^2 + (\frac{h}{2})^2}$$

2. Being able to insert the robot in to a wheel of diameter 0.5 m so:

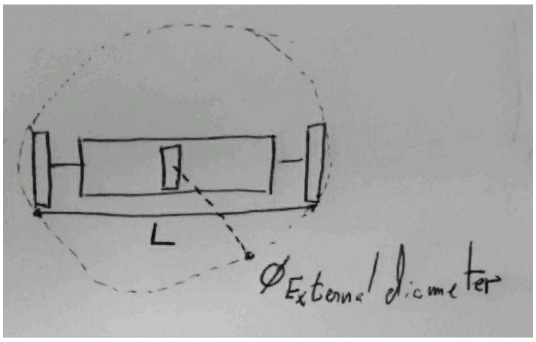


Figure 9: External diameter diagram

$$0.25m > \sqrt{r_{wheel}^2 + L^2/4}$$

3. We can place all electronic the devices:

$$L > 0.3m + w$$

4. Maximum weight of the robot: 5 kg

3.2 Requirements

We would like our robot to reach some mechanical specifications. These are related to mechanical equations that we will develop in the next section. They refer to the max speed, acceleration and terrain inclination the robot may achieve while controlling its platform inclination. We have divided our specifications in two blocks according to the two mechanisms.

Flywheel mode

1. $\dot{y}_{max}$ (equation 10) $> 0.1 m/s$.
2. $\ddot{y}_{max}$ (equation 9) $> 1 m/s^2$.
3. $sin(\alpha_{max})$ (equation 13) > 0.16.

Pendulum mode

1. $\dot{y}_{max}$ (equation 12) $> 1 m/s$.
2. $\ddot{y}_{max}$ (equation 11) $> 0.1 m/s^2$.
3. $sin(\alpha_{max})$ (equation 14) > 0.02.

3.3 Cost function

In addition to fulfilling the previous inequalities, we will adjust our design parameters [w (width of the cylinders), N(number of cylinders), r wheel and r flywheel] to minimize a cost function.

We will maximize the maximum sinus in the pendulum mode (equation 14) because it gives the robot the capacity to deliver force in a permanent state.

And we will also maximize the square of the max speed the robot can achieve in flywheel mode (equation 10) because it is proportional to the energy the robot can deliver using the flywheel at a certain moment.

Both equations try to maximize different modes so the robot we have a compromise between the two of them.

$$cost(r_{flywheel}, r_{wheel}, w, N) = -sin(\alpha_{max})_{pendulum} - \dot{y}^2_{max-flywheel} \tag{1}$$

In the next section we will find out what are the values of these equations with the construction parameters.

4. Mechanical analysis

In this section we will analyze and understand the key dynamics of our robot, so we can choose the design parameters based on performance indicators. All the analysis is made supposing the inclination of the platform is maintained fixed. As parameters, we have the width of the flywheel masses w, the number of masses N, the radius of the flywheel $r_{flywheel}$, and the radius of the wheels r_{wheel}.

4.1 Inclination control

In order to keep the inclination of the platform at a certain angle ϕ we must be able to compensate all the torque being applied to the platform.

$$\ddot{\phi} \cdot I_{platform} = \tau_{platform}$$

Assuming that the platform is well-balanced (the center of masses is located at the rotation axis by our design restriction) and neglecting the torque generated by the friction with air, the sum of all the torques in the motor axis applied to the platform is equal to the sum of the torque applied by the motors:

$$\tau_{platform} = \sum \tau_{motors}$$

The torque that the motors deliver to the wheels and to the flywheel create a reaction in the platform in the opposite direction.

$$\tau_{platform} = -\tau_{motor-right-wheel} - \tau_{motor-left-wheel} - \tau_{motor-flywheel}$$

If we want keep the inclination ϕ, we must be able to cancel $\tau_{platform}$. Observe that the angular acceleration $\ddot{\phi}$ of the platform is linearly dependent with the torque it receives.

$$\begin{gathered} 0 = -\tau_{motor-right-wheel} - \tau_{motor-left-wheel} - \tau_{motor-flywheel} \Rightarrow \\ \tau_{motor-right-wheel} + \tau_{motor-left-wheel} = -\tau_{motor-flywheel} \end{gathered} \tag{2}$$

In other words, we must overpass the torque of the wheels with the torque of the flywheel if we want to control the inclination.

4.2 Wheels torque

The wheel torque we can induce is limited by the motor specifications. Note that the maximum torque of the motor is a function of velocity and in particular at max speed the torque is zero. See figure 10 to see the plot of this function.

$$\tau_{motor-wheel}(\omega_{wheel})$$

We assume that the wheels just roll and do no slip. The robot is pushed by the wheels that make a force $F_{friction}$ against the ground in the contact point. See figure 11.

We can express the torque at the center of the wheel as:

$$\tau_{motor-wheel} + F_{friction} \cdot r_{wheel} = I_{wheel} \cdot \dot{\omega}_{wheel}$$

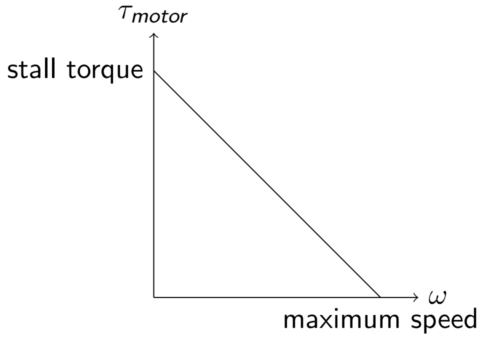


Figure 10: Motor torque.

$$\tau_{motor-wheel} = I_{wheel} \cdot \dot{w}_{wheel} - F_{friction} \cdot r_{wheel} \tag{3}$$

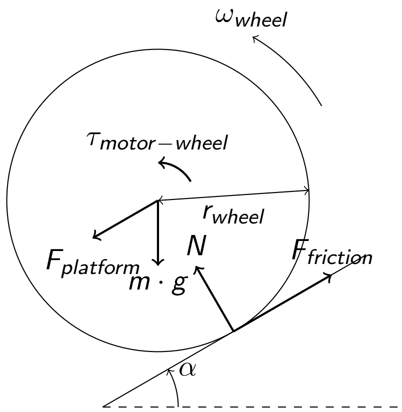


Figure 11: Wheel force diagram.

4.3 Flywheel torque

The flywheel torque we can induce is also limited by the motor specifications.

Assuming a general configuration of the flywheel where the moving mass is at distance r from the axis and angle θ, see figure 12, we formulate its torque the following way:

$$\tau_{motor-flywheel} + m_{cylinder} \cdot g \cdot (r - r_{max}) \cdot \sin\theta = \ddot{\theta} \cdot I_{flywheel}(r)$$

$$\tau_{motor-flywheel} = \ddot{\theta} \cdot I_{flywheel}(r) + m_{cylinder} \cdot g \cdot (r_{max} - r) \cdot \sin\theta \tag{4}$$

Note that in equation 4 the two terms correspond to the two two mechanisms: acceleration of the flywheel and position of the pendulum.

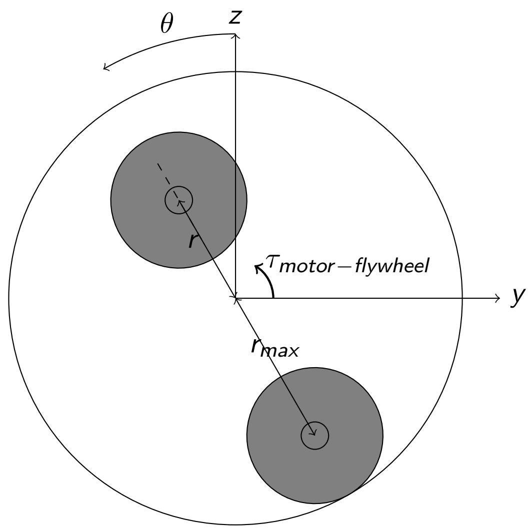


Figure 12: Flywheel diagram for $N = 2$

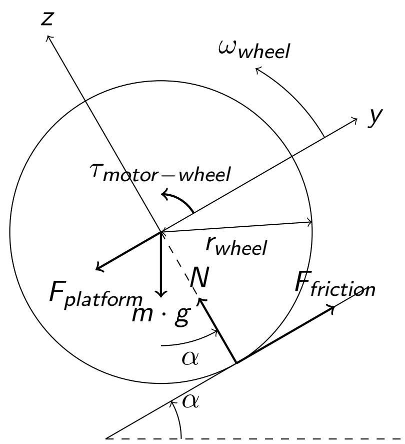


Figure 13: Wheel forward force diagram.

4.4 Maximum speed, acceleration and terrain inclination

In this subsection we would like to study the maximum speed and acceleration the robot may obtain in straight direction and the maximum terrain inclination it can stand on α.

We will assume both wheels turn at the same speed, have the same $F_{friction}$ and the same τ_{wheel}:

$$\omega_{wheel-left} = \omega_{wheel-right} = \omega_{wheel}$$

Applying Newton's first law in the y-axis of figure 13:

$$\ddot{y} \cdot m_{total} = 2 \cdot F_{friction} - m_{total} \cdot g \cdot sin(\alpha)$$

Substituting $F_{friction}$ taking in to account equation 3:

$$\ddot{y} \cdot m_{total} = 2 \cdot \frac{I_{wheel} \cdot \dot{w}_{wheel} - \tau_{motor-wheel}}{r_{wheel}} - m_{total} \cdot g \cdot sin(\alpha)$$

Using equation 2:

$$\Rightarrow \ddot{y} \cdot m_{total} = \frac{2 \cdot I_{wheel} \cdot \dot{w}_{wheel}}{r_{wheel}} + \frac{\tau_{motor-flywheel}}{r_{wheel}} - m_{total} \cdot g \cdot sin(\alpha) \tag{5}$$

We will now study different cases to better understand this equation.

4.4.1 Speed and acceleration with no terrain inclination ($\alpha = 0$)

The objective here is to obtain the maximum speed and acceleration we can get starting from rest in a plain surface.

The equation we get by substituting $\alpha = 0$ in equation 5:

$$\ddot{y} \cdot m_{total} = \frac{\tau_{motor-flywheel}}{r_{wheel}} + \frac{2 \cdot I_{wheel} \cdot \dot{w}_{wheel}}{r_{wheel}}$$

Substituting equation 4:

$$\ddot{y} \cdot m_{total} = \frac{\ddot{\theta} \cdot I_{flywheel}(r) + m_{cylinder} \cdot g \cdot (r_{max} - r) \cdot \sin\theta}{r_{wheel}} + \frac{2 \cdot I_{wheel} \cdot \dot{\omega}_{wheel}}{r_{wheel}} \tag{6}$$

We will now split the study in two cases:

1. **Flywheel case:** r is fixed to $r = r_{max}$

   Then:

   $$\ddot{y} \cdot m_{total} = -\frac{\ddot{\theta} \cdot I_{flywheel}(r_{max})}{r_{wheel}} + \frac{2 \cdot I_{wheel} \cdot \dot{\omega}_{wheel}}{r_{wheel}}$$

   Taking in to account the following relation:

   $$-\omega_{wheel} \cdot r_{wheel} = \dot{y} \Rightarrow -\dot{\omega}_{wheel} \cdot r_{wheel} = \ddot{y} \tag{7}$$

   Substituting in equation 6:

   $$-\dot{\omega}_{wheel} \cdot r_{wheel} \cdot m_{total} = \frac{\ddot{\theta} \cdot I_{flywheel}}{r_{wheel}} + \frac{2 \cdot I_{wheel} \cdot \dot{\omega}_{wheel}}{r_{wheel}}$$

   $$-\dot{\omega}_{wheel} \cdot (r_{wheel} \cdot m_{total} + \frac{2 \cdot I_{wheel}}{r_{wheel}}) = \frac{\ddot{\theta} \cdot I_{flywheel}}{r_{wheel}}$$

   We now define R as a non dimensional constant being the quotient between $\dot{w}_{wheel}$ and $-\ddot{\theta}$.

   $$R = \frac{\dot{\omega}_{wheel}}{-\ddot{\theta}} = \frac{I_{flywheel}}{r_{wheel}^2 \cdot m_{total} + 2 \cdot I_{wheel}} \tag{8}$$

   The moments of inertia are:

   $$I_{wheel} \approx \frac{1}{2} \cdot m_{wheel} \cdot r_{wheel}^2$$

   $$I_{flywheel} \approx N \cdot m_{cylinder} \cdot r_{max}^2$$

Substituting those in equation 8 we get:

$$R \approx \frac{N \cdot m_{cylinder} \cdot r_{max}^2}{r_{wheel}^2 \cdot m_{total} + 2 \cdot I_{wheel}} < 1$$

We can see that R will always be smaller than 1 because $N \cdot m_{cylinder} < m_{total}$ and $r_{max} < r_{wheel}$.

This means that the maximum acceleration of the wheels will be limited by the acceleration of the flywheel. The same is true for the speed.

In order to get the forward acceleration we can use equation 7

$$\ddot{y} = -\dot{\omega}_{wheel} \cdot r_{wheel} = R \cdot \ddot{\theta} \cdot r_{wheel}$$

And using equation 4 we get that the maximum is:

$$\tau_{motor}(\omega) = \ddot{\theta} \cdot I_{flywheel}(r) \Rightarrow \ddot{\theta} = \frac{\tau_{motor}(\dot{\theta})}{I_{flywheel}(r)}$$

$$\ddot{y}_{max} = R \cdot \frac{\tau_{motor}(\dot{\theta})}{I_{flywheel}(r)} \cdot r_{wheel}$$

$$\Rightarrow \ddot{y}_{max} = \frac{I_{flywheel}}{r_{wheel}^2 \cdot m_{total} + 2 \cdot I_{wheel}} \cdot \frac{\tau_{motor}(\dot{\theta})}{I_{flywheel}(r)} \cdot r_{wheel}$$

$$\boxed{\ddot{y}_{max} = \frac{\tau_{motor}(\dot{\theta}) \cdot r_{wheel}}{r_{wheel}^2 \cdot m_{total} + 2 \cdot I_{wheel}}} \tag{9}$$

In order to compute the maximum speed we assume that the initial conditions are $\dot{\theta} = 0$ and $\omega_{wheel} = 0$

$$\omega_{wheel-max} = \int_{t=0}^{t=t_{max}} \dot{\omega}_{wheel} \cdot dt$$

Now we will proceed to do a change of variables in the integral.

$$\frac{\partial \dot{\theta}}{\partial t} = \ddot{\theta} \Rightarrow dt = \frac{d\dot{\theta}}{\ddot{\theta}}$$

$$\omega_{wheel-max} = \int_{\dot{\theta}=0}^{\dot{\theta}=\dot{\theta}_{max}} \frac{\dot{\omega}_{wheel}}{\ddot{\theta}} \cdot d\dot{\theta} = \int_{\dot{\theta}=0}^{\dot{\theta}=\dot{\theta}_{max}} -R \cdot d\dot{\theta} = -R \cdot \dot{\theta}_{max}$$

$$\boxed{\dot{y}_{max} = r_{wheel} \cdot R \cdot \dot{\theta}_{max}} \tag{10}$$

And $\dot{\theta}_{max}$ is a limitation imposed by the motor specifications. Note that this is the maximum speed we can get using the flywheel system starting from rest.

2. **Pendulum:** $\dot{\theta} = 0$, and r is fixed to $r = r_{min}$

Using equation 6 and $\ddot{\theta} = 0$

$$\ddot{y} \cdot m_{total} = \frac{m_{cylinder} \cdot g \cdot (r_{max} - r_{min}) \cdot \sin\theta}{r_{wheel}} + \frac{2 \cdot I_{wheel} \cdot \dot{w}_{wheel}}{r_{wheel}}$$

Multiplying by r_{wheel} both sides of the equation we get:

$$\ddot{y} \cdot m_{total} \cdot r_{wheel} = m_{cylinder} \cdot g \cdot (r_{max} - r_{min}) \cdot \sin\theta + 2 \cdot I_{wheel} \cdot \dot{w}_{wheel}$$

And using $\dot{w}_{wheel} = -\frac{\ddot{y}}{r_{wheel}}$

$$\ddot{y} \cdot m_{total} \cdot r_{wheel} + 2 \cdot I_{wheel} \cdot \frac{\ddot{y}}{r_{wheel}} = m_{cylinder} \cdot g \cdot (r_{max} - r) \cdot \sin\theta$$

With some manipulation:

$$\ddot{y} = \frac{m_{cylinder} \cdot g \cdot (r_{max} - r_{min}) \cdot \sin\theta}{m_{total} \cdot r_{wheel} + \frac{2 \cdot I_{wheel}}{r_{wheel}}}$$

Which is maximum when $\sin\theta = 1$

$$\boxed{\ddot{y}_{max} = \frac{m_{cylinder} \cdot g \cdot (r_{max} - r_{min}) \cdot}{m_{total} \cdot r_{wheel} + \frac{2 \cdot I_{wheel}}{r_{wheel}}}} \tag{11}$$

Since the maximum acceleration obtained is constant, the speed could increase without limit. To get a meaningful estimation of the maximum speed, we will take into account the air friction that becomes more and more important as the speed increases.

$$F_{drag} = \frac{1}{2} \cdot \rho \cdot C_D \cdot A \cdot \dot{y}^2$$

Adding this term to equation 5 we get:

$$\ddot{y} \cdot m_{total} = \frac{2 \cdot I_{wheel} \cdot \dot{w}_{wheel}}{r_{wheel}} + \frac{\tau_{motor-flywheel}}{r_{wheel}} - m_{total} \cdot g \cdot sin(\alpha) - F_{drag}$$

And making $\ddot{y} = 0$ and $\alpha = 0$.

$$F_{drag} = \frac{\tau_{motor-flywheel}}{r_{wheel}}$$

$$\frac{1}{2} \cdot \rho \cdot C_D \cdot A \cdot \dot{y}^2 = m_{cylinder} \cdot g \cdot (r_{max} - r_{min}) \cdot sin(\theta)/r_{wheel}$$

The maximum $\dot{y}$ is then obtained when $\theta = \frac{\pi}{2}$:

$$\dot{y}_{max} = \sqrt{\frac{2 \cdot m_{cylinder} \cdot g \cdot (r_{max} - r_{min})}{\rho \cdot C_D \cdot A \cdot r_{wheel}}}$$

We will pick $C_D = 1$, $\rho = 1.2kg/m^3$ and $A = 0.01m^2$ for our computations.

Also, note that the speed is also limited by maximum speed a motor can get $\dot{\theta}_{max}$.

$$\boxed{\dot{y}_{max} = min(\dot{\theta}_{max} \cdot r_{wheel}, \sqrt{\frac{2 \cdot m_{cylinder} \cdot g \cdot (r_{max} - r_{min})}{\rho \cdot C_D \cdot A \cdot r_{wheel}}})} \tag{12}$$

4.4.2 Terrain inclination ($\alpha > 0$)

The goal of this subsection is to study which is the maximum terrain inclination α_{max} on which the robot can stay, standing still without rolling down.

Substituting $\ddot{y} = 0$ and $\dot{\omega}_{wheel} = 0$ in equation 5

$$0 = \frac{\tau_{motor-flywheel}}{r_{wheel}} - m_{total} \cdot g \cdot sin(\alpha)$$

$$m_{total} \cdot g \cdot sin(\alpha) = \frac{\tau_{motor-flywheel}}{r_{wheel}}$$

$$sin(\alpha) = min(1, \frac{\tau_{motor-flywheel}}{m_{total} \cdot g \cdot r_{wheel}})$$

Substituting equation 4

$$sin(\alpha) = min(1, \frac{\ddot{\theta} \cdot I_{flywheel}(r) + m_{cylinder} \cdot g \cdot (r_{max} - r) \cdot \sin\theta}{m_{total} \cdot g \cdot r_{wheel}})$$

We are going to distinguish the same two cases as in the previous section:

1. Flywheel case: r is fixed to $r = r_{max}$ The maximum inclination at a certain moment:

$$\boxed{sin(\alpha_{max}) = min(1, \frac{\ddot{\theta} \cdot I_{flywheel}}{m_{total} \cdot g \cdot r_{wheel}}) = min(1, \frac{\tau_{motor-flywheel}(\dot{\theta})}{m_{total} \cdot g \cdot r_{wheel}})} \quad (13)$$

   This angle doesn't give us a lot of information because it may not be fulfilled in a permanent state.

2. Pendulum: $\dot{\theta} = 0$, and r is fixed to $r = r_{min}$

$$sin(\alpha) = \frac{min(\tau_{motor-flywheel}, m_{cylinder} \cdot g \cdot (r_{max} - r_{min}) \cdot \sin\theta)}{m_{total} \cdot g \cdot r_{wheel}}$$

   Which is maximum when $\sin\theta = 1$

$$\boxed{sin(\alpha_{max}) = \frac{min(\tau_{motor-flywheel}, m_{cylinder} \cdot g \cdot (r_{max} - r_{min}))}{m_{total} \cdot g \cdot r_{wheel}}} \quad (14)$$

   Note that the $sin(\alpha_{max})$ will not be more than 1.

5. Optimization Results

Our procedure has been making a grid with the four parameters of the robot design: w (width of the cylinders), N(number of cylinders), r_{wheel} and $r_{flywheel}$. We have fixed $r_{flywheel}$ to a number of discrete values between 0.05 and 0.125 and, for each one, we evaluate the cost for each point of the grid, remove those which do not fulfill the restrictions, and keep the configuration providing the minimum cost.

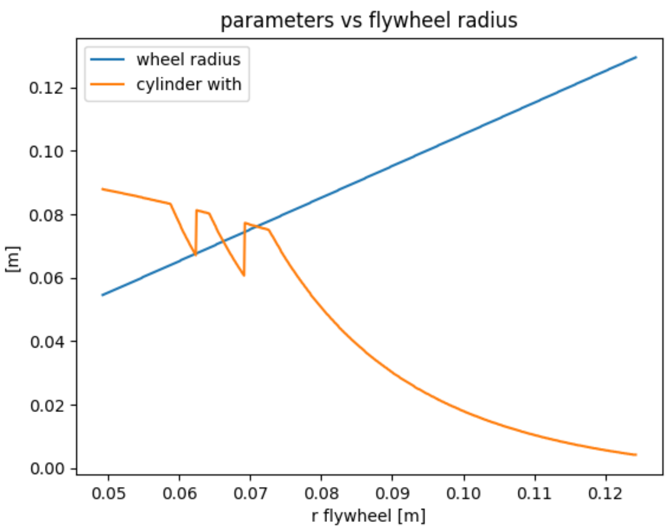


Figure 14: Plot of the wheel radius and the width of the cylinder that was optimal for each flywheel radius.

We can observe that the relation between the flywheel radius and the wheel radius is linear. This is because we are always on the edge of the restriction number one.

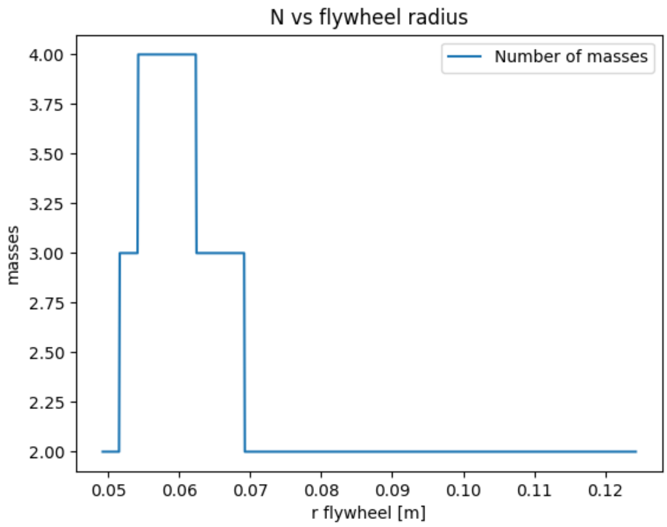


Figure 15: Plot of the N that minimize the cost function.

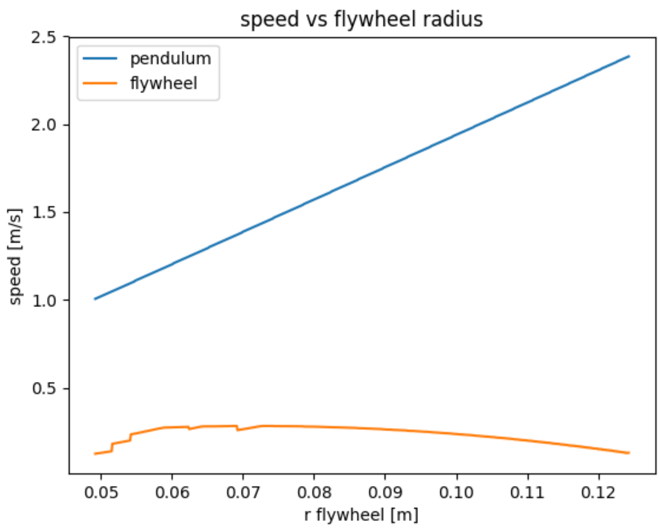


Figure 16: Plot of the equations 10 and 12 at the parameters that minimize the cost function and fullfil the requirements and restrictions

In this figure we can observe that the speed of in the pendulum mode increases linearly with the flywheel radius. This is because it is linearly related with the wheel radius.

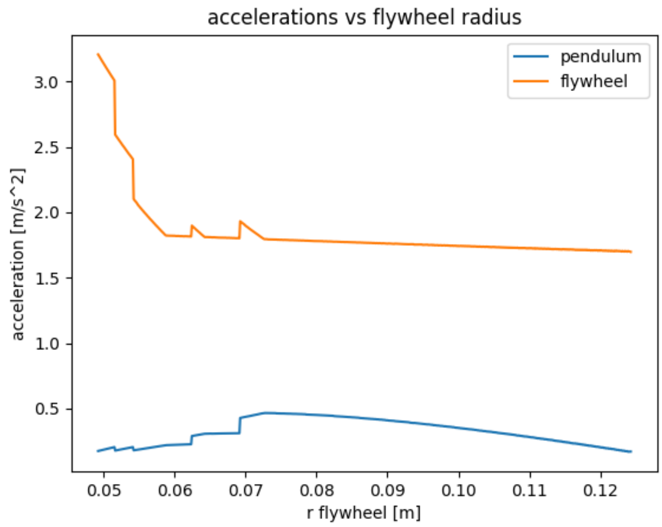


Figure 17: Plot of the equations 9 and 11 at the parameters that minimize the cost and fulfill the requirements and restrictions

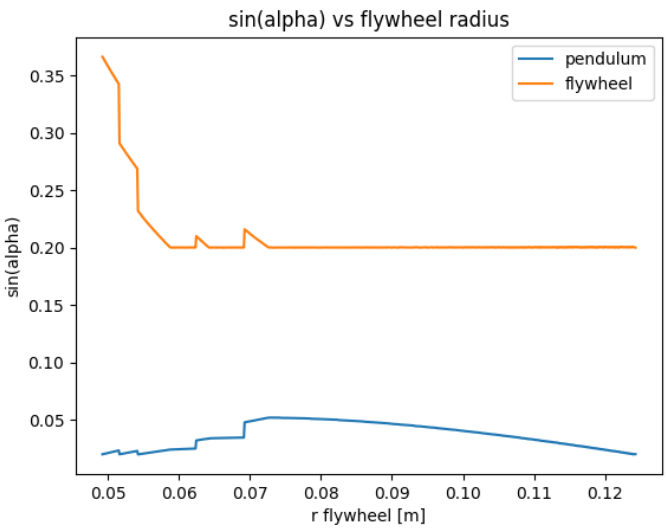


Figure 18: Plot of the equations 13 and 14 at the parameters that minimize the cost and fulfill the requirements and restrictions

We can see in figure 18 that when the flywheel radius exceeds 0.07 m the requirement of the flywheel sinus becomes active.

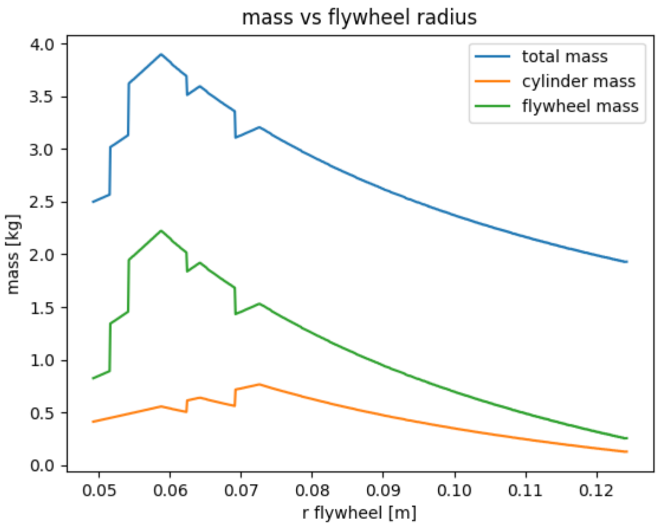


Figure 19: Plot of the mass for each configuration.

We can see in figure 19 that the total mass never exceeds the 5 kg limit.

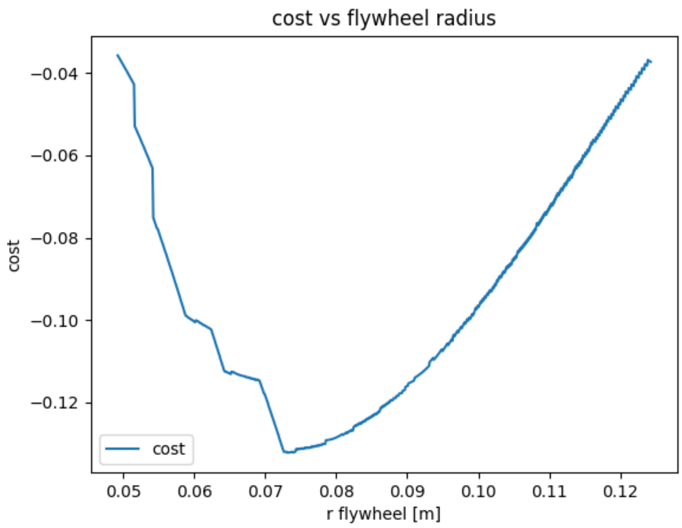


Figure 20: Plot of the equation 1 for each configuration.

Seeing the parameters that minimize cost and the material that we had available our selected parameters were:

$r_{flywheel}$	r_{wheel}	w	N
8cm	10cm	5cm	2

With these parameters we get the following specifications:

Total mass	2,91 kg
Pendulum	
Maximum sinus	0,042
Maximum speed horizontal	1,84 m/s
Maximum acceleration horizontal	0,38 m/s^2
Flywheel	
Maximum sinus	0,171
Maximum speed horizontal	0,24 m/s
Maximum acceleration horizontal	1.52 m/s^2

6. Rectilinear movement dynamics with $r_{flywheel}$ fixed

In order to study the dynamics of the robot we will use Lagrange mechanics. To reduce the number of variables we will study the case of rectilinear movement by imposing that both wheels turn at the same speed. We will also set the radius of the free weight to $r_{flywheel}$. Note that in this analysis the platform is not assumed to keep a fixed inclination as before, but it is allowed to rotate.

The generalized coordinates (q) will be:

1. $\phi_{ground-wheel}$: rotation of the wheel respect the ground.
2. $\phi_{wheel-platform}$: rotation of the platform respect the wheel.
3. $\phi_{platform-flywheel}$: rotation of the flywheel respect the platform.

We will use two auxiliary variables:

1. $\phi_{ground-platform} = \phi_{ground-wheel} + \phi_{wheel-platform}$: rotation of the platform respect the ground.
2. $\phi_{ground-flywheel} = \phi_{ground-platform} + \phi_{platform-flywheel}$: rotation of the flywheel respect the ground.

The total potential energy:

$$V = m_{cylinder} \cdot (r_{flywheel} - r_{flywheel-max}) \cdot cos(\phi_{ground-flywheel}) \cdot g \tag{15}$$

The total kinetic energy:

$$T = \frac{1}{2} \cdot [\dot{\phi}^2_{ground-wheel} \cdot I_{wheel} + \dot{\phi}^2_{ground-platform} \cdot I_{platform} + \dot{\phi}^2_{ground-flywheel} \cdot I_{flywheel} + \dot{\phi}^2_{ground-wheel} \cdot r^2_{wheel} \cdot m_{total}] \tag{16}$$

The Lagrangian is defined as:

$$L = T - V \tag{17}$$

Lagrange's equation is:

$$\frac{d}{dt}(\frac{\partial L}{\partial \dot{q}_j}) = \frac{\partial L}{\partial q_j} + F_j \tag{18}$$

So in our case:

$$\frac{\partial L}{\partial \dot{\phi}_{ground-wheel}} = \dot{\phi}_{ground-wheel} \cdot I_{wheel} + \dot{\phi}_{ground-platform} \cdot I_{platform} + \dot{\phi}_{ground-flywheel} \cdot I_{flywheel} + \dot{\phi}_{ground-wheel} \cdot r^2_{wheel} \cdot m_{total} \tag{19}$$

$$\frac{\partial L}{\partial \dot{\phi}_{wheel-platform}} = \dot{\phi}_{ground-platform} \cdot I_{platform} + \dot{\phi}_{ground-flywheel} \cdot I_{flywheel} \tag{20}$$

$$\frac{\partial L}{\partial \dot{\phi}_{platform-flywheel}} = \dot{\phi}_{ground-flywheel} \cdot I_{flywheel} \tag{21}$$

We define M as the following matrix:

$$\begin{pmatrix} I_{wheel} + I_{platform} + I_{flywheel} + r^2_{wheel} \cdot m_{total} & I_{platform} + I_{flywheel} & I_{flywheel} \\ I_{platform} + I_{flywheel} & I_{platform} + I_{flywheel} & I_{flywheel} \\ I_{flywheel} & I_{flywheel} & I_{flywheel} \end{pmatrix} \tag{22}$$

In matrix form and with our generalized coordinates:

$$\frac{\partial L}{\partial \dot{q}} = M \cdot \dot{q} \tag{23}$$

$$\frac{d}{dt}(\frac{\partial L}{\partial \dot{q}}) = M \cdot \ddot{q} \tag{24}$$

Let a be a constant:

$$a = m_{cylinder} \cdot (r_{flywheel} - r_{flywheel-max}) \cdot g \tag{25}$$

$$\frac{\partial L}{\partial q} = a \cdot sin(\phi_{ground-flywheel}) \cdot \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \tag{26}$$

So using Lagrange's equation we get:

$$\boxed{M \cdot \ddot{q} = a \cdot sin(\phi_{ground-flywheel}) \cdot \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} + F} \tag{27}$$

6.1 Simulations

In this subsection we study different policies for the generalized forces (F) applied to the robot in rectilinear movement case.

The robot has 2 actuators, the motors between the wheel and the platform and the motor between the flywheel and the platform. All motors in the systems have a limited torque related with the speed as you may see in Figure 10.

$$\boxed{F = \begin{pmatrix} 0 \\ \tau_{wheel-platform} \\ \tau_{platform-flywheel} \end{pmatrix}} \tag{28}$$

In order to compare the different policies we compare their efficiency in performing the same task: The robot must travel 30 radians (3 meters) and stop at the end. All the parameters from the simulation are set equal to the values found in section 5. Keep in mind that the wheel radius is 10 cm.

Here you can see a table that summarize the results:

Experiment	Total time(s)	Time to max speed(s)	Max speed(rad/s)	Time to brake(s)
Controlling inclination				
Flywheel	12,45	0,7	2,45	0,15
Pendulum	5,59	2,69	10,36	2,90
Waitress	8,24	4,12	5,49	4,12
Free inclination				
Double flywheel	3,9	1	8,51	0.27
Compose mechanism	3,82	2,3	10,2	2,04

6.1.1 Controlling the platform inclination

In the first two experiments we want to keep the platform in a horizontal position. That's the reason why we force the $\tau_{wheel-platform}$ equal to $\tau_{platform-flywheel}$ in the experiments flywheel and pendulum.

1. Flywheel

The movable weight is fixed at r_{max}. In this experiment we consign the platform-flywheel motor to output the maximum possible torque. Then the platform-wheels motors deliver the same torque. The platform-flywheel motor reaches in less than one second the maximum speed. This then limits the wheel-platform speed because no more torque can be applied one that point is reached.

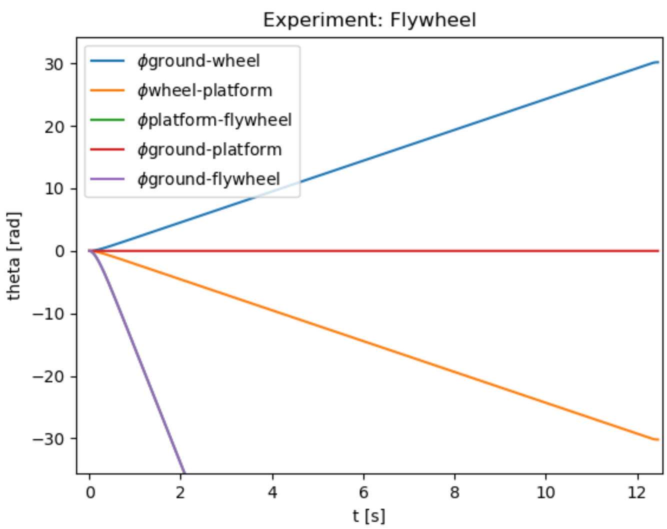


Figure 21: Plot of the angles for the flywheel experiment.

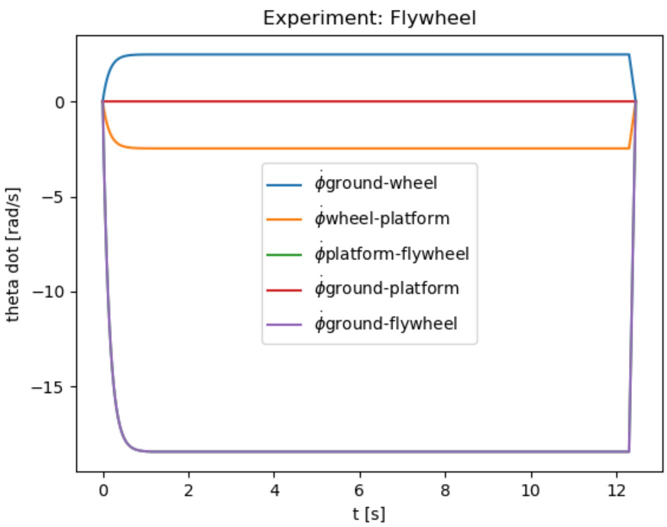


Figure 22: Plot of the angular velocities for the flywheel experiment.

2. Pendulum

The movable weight is fixed at r_{min}. In this experiment we consign the platform-flywheel motor to get to 90 with a PID controller. Then the platform-wheels motors deliver the same torque. The robot does not accelerate as fast as in the flywheel experiment but reaches a higher speed that allows it to travel the distance in less time. Note that the robot could have kept accelerating but had to brake before reaching the maximum speed.

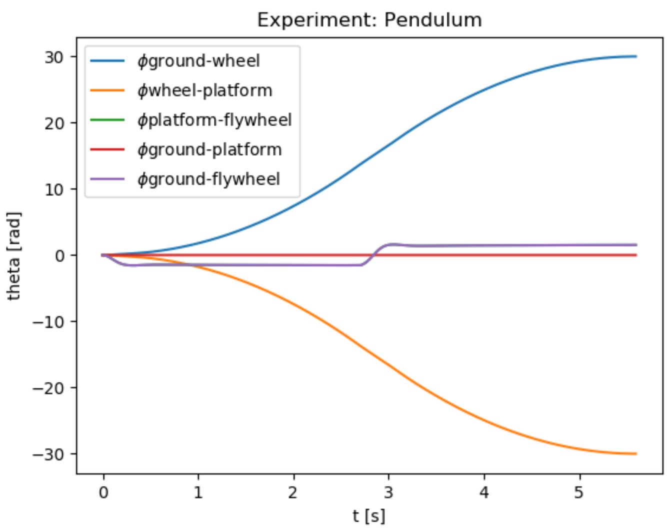


Figure 23: Plot of the angles for the pendulum experiment.

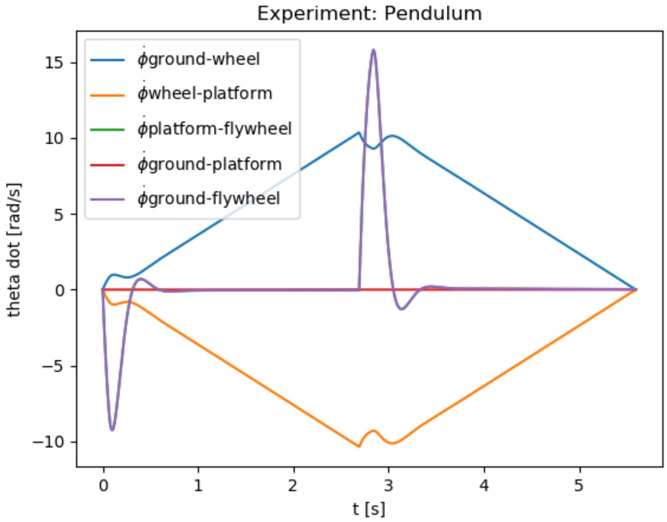


Figure 24: Plot of the angular velocities for the pendulum experiment.

3. Waitress

   The movable weight is fixed at r_{min}. This experiment is different from the previous two and its aim is to illustrate how the robot could follow and inclination consign. The experiment is designed the following way. A set point torque is given to the wheel-platform and the flywheel-motor PID has an inclination consign too. The inclination consign is such that the sum of accelerations received from and object in the platform is perpendicular to the platform.

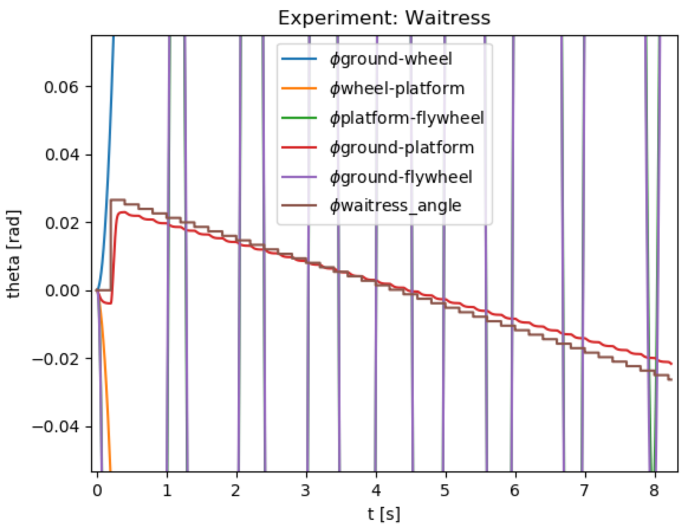


Figure 25: Plot of the angles for the waitress experiment.

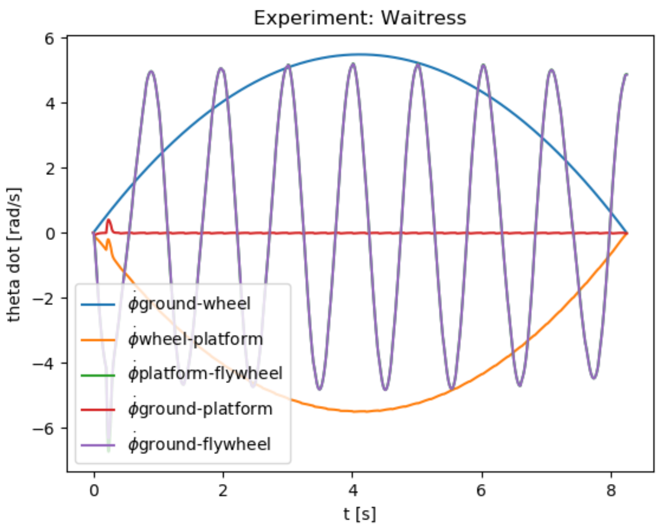


Figure 26: Plot of the angular velocities for the waitress experiment.

6.1.2 Letting the platform turn

In this experiment we take advantage of the fact that the platform may not be needed to stay horizontal in some displacement. The robot uses the platform as an additional flywheel to help the robot accelerate and brake.

1. Double Flywheel

   The movable weight is fixed at r_{max}. In this experiment we add the flywheel turn over the platform turn to create more torque to accelerate our wheels. We always do the maximum possible torque with all the motors.

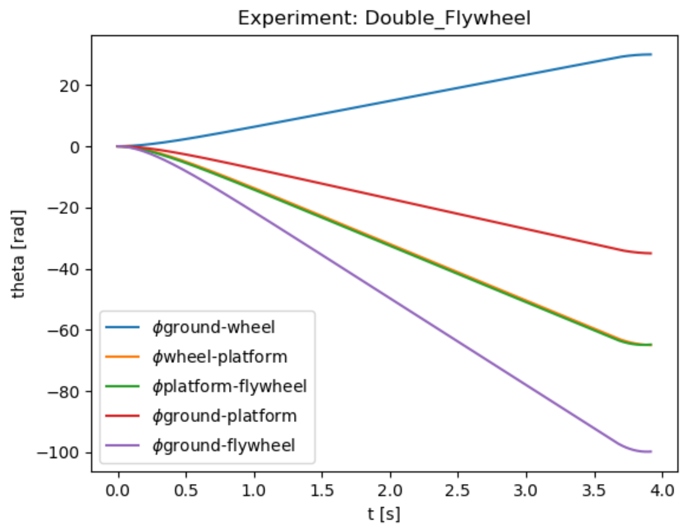


Figure 27: Plot of the angles for the double flywheel experiment.

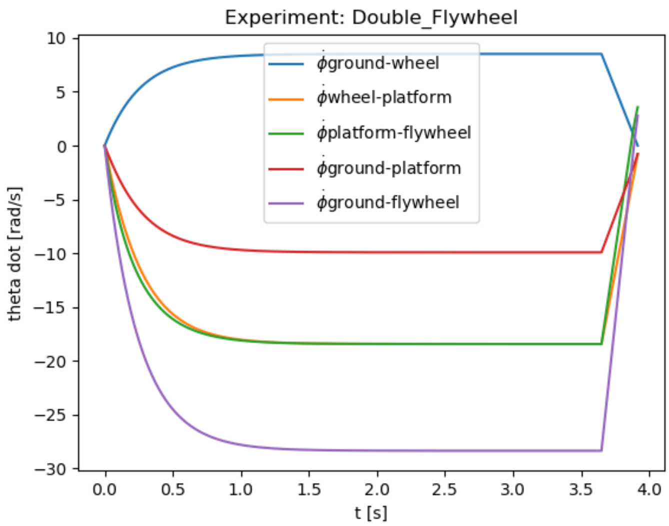


Figure 28: Plot of the angular velocities for the double flywheel experiment.

2. Compose

   The movable weight is fixed at r_{min}. In this experiment we add the pendulum effect to the flywheel effect produced by the flywheel. We do so with a PID controller for the angle ground-flywheel and making the maximum possible torque with the platform-wheel motors.

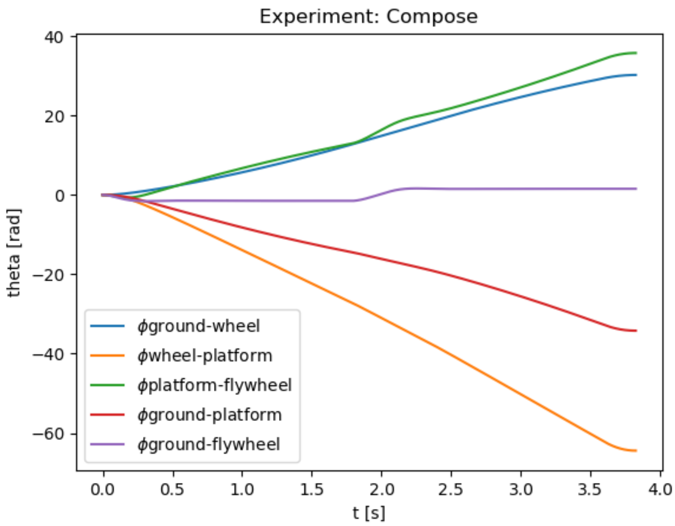


Figure 29: Plot of the angles for the compose experiment.

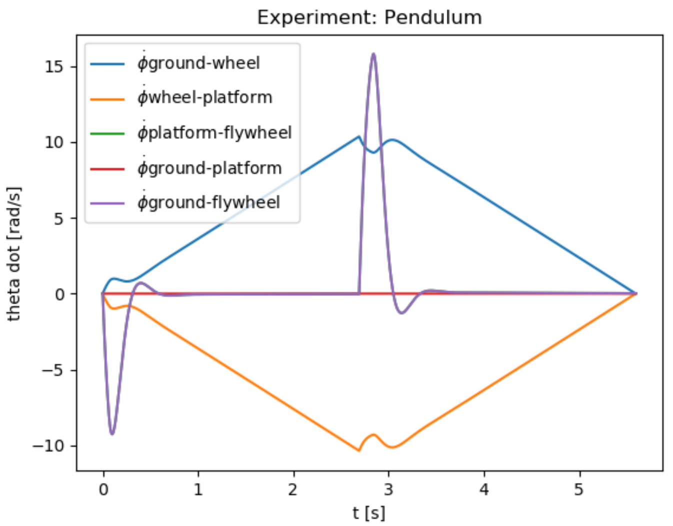


Figure 30: Plot of the angular velocities for the pendulum experiment.

7. Flywheel brake study

The problem with the flywheel method is that, when it reaches the maximum speed, the motor cannot apply torque. This section aims to study a possible way to break the flywheel without applying torque to the platform. The idea is combining both methods: we will leave the moving weight free.

The initial hypothesis was: If we let weight move, when it is going upward will have a larger radius than when is going downward and will produce an average external torque against the movement of the flywheel. From an energetic point of view we are transforming the rotation energy of the flywheel in to translation of the free cylinder and then releasing it trough collisions.

7.1 System of differential equations

To simplify the experiment we will assume that the platform inclination is constant and that the motor is not producing any torque to the platform.

As described in figure 12 we will use two variables to describe the flywheel position: r and θ.

Using equation 4:

$$\tau_{motor-flywheel} = \ddot{\theta} * I_{flywheel}(r) + m_{cylinder} * g * (r_{max} - r) * \sin(\theta)$$

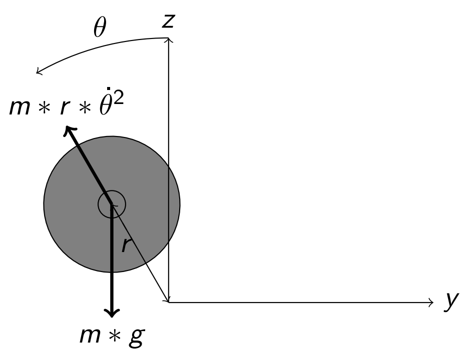


Figure 31: Cylinder force diagram.

As seen in figure 31, we can deduce Newton's equation for the distance from the cylinder to the center r. Note that we are adding the centrifugal force term due to the non-inertial frame.

$$\ddot{r} * m = -m * g * cos(\theta) + m * r * \dot{\theta}^2$$

$$\ddot{r} = -g * cos(\theta) + r * \dot{\theta}^2$$

The variables we will be using for our ODE system are: r,$\dot{r}$,θ,$\dot{\theta}$.

Note that we impose $\tau_{flywheel} = 0$ so the motor is not applying any torque.

$$\begin{cases} \dot{r} = \dot{r} \\ \ddot{r} = -g * cos(\theta) + r * \dot{\theta}^2 \\ \dot{\theta} = \dot{\theta} \\ \ddot{\theta} = \frac{m_{cylinder} * g * (r - r_{max}) * \sin\theta}{I_{flywheel}(r)} \end{cases}$$

Our initial conditions will be the free cylinder mass lying on the bottom of the flywheel and the flywheel turning at a speed θ_0:

$$\begin{cases} r = r_{max} \\ \dot{r} = 0 \\ \theta = \pi \\ \dot{\theta} = \theta_0 \end{cases}$$

We will use a Poincaré map to simulate the bounce with the end of the guides at $r = r_{min}$ and $r = r_{max}$. At each bounce we will reduce its kinetic energy by a percentage *bounce_percentage*.

7.2 Results

The parameters of the simulation where:

$$\begin{cases} r_{flywheel} = 8cm \\ r_{wheel} = 9cm \\ w = 7cm \\ \dot{\theta}_0 = 4.2\pi rad/s \\ bounce\_percentage = 0.0(totally\_inelastic) \end{cases}$$

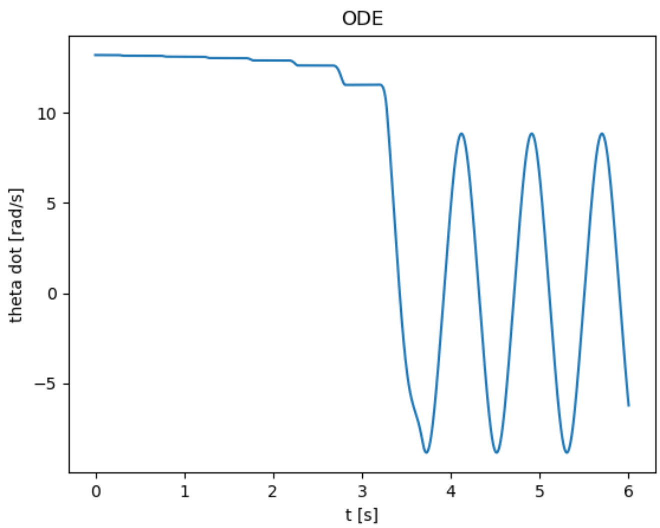


Figure 32: How the variable $\dot{\theta}$ evolve over time

As we can see in figure 32 the flywheel is braking until it becomes a pendulum and starts oscillating.

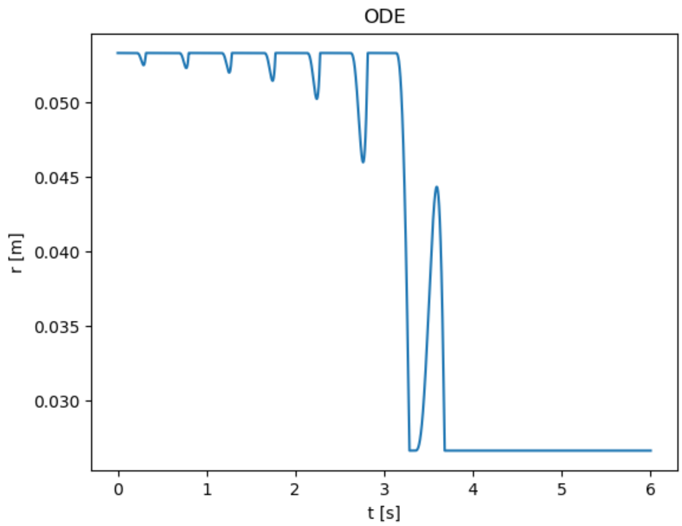


Figure 33: How the variable r evolves over time

In the first laps the cylinder is almost always at the maximum value of r, but as the speed decreases each lap the value of r decreases until it hits the r_{min}. In other words, at the beginning is operating as a flywheel but then thanks to the collisions it becomes a pendulum.

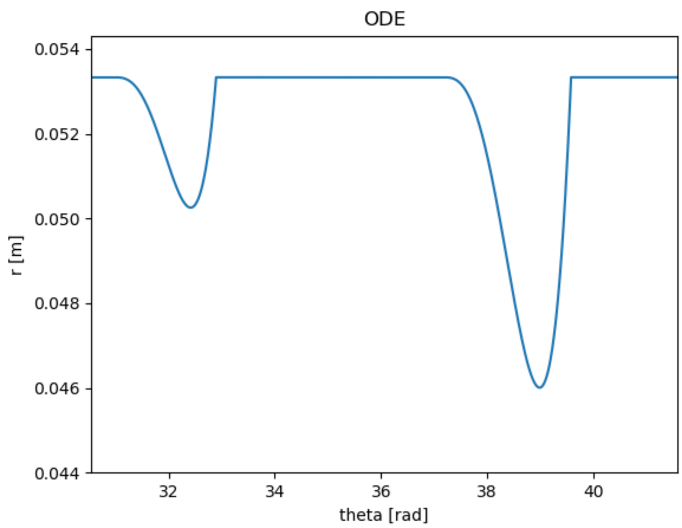


Figure 34: How the variable r evolve over θ zoomed

In image 34 we can appreciate that the r decreases slower than what it increases.

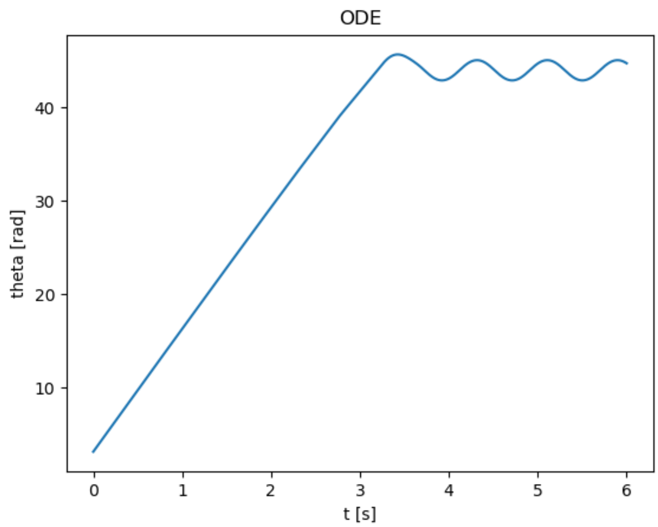


Figure 35: How the variable θ evolves over time.

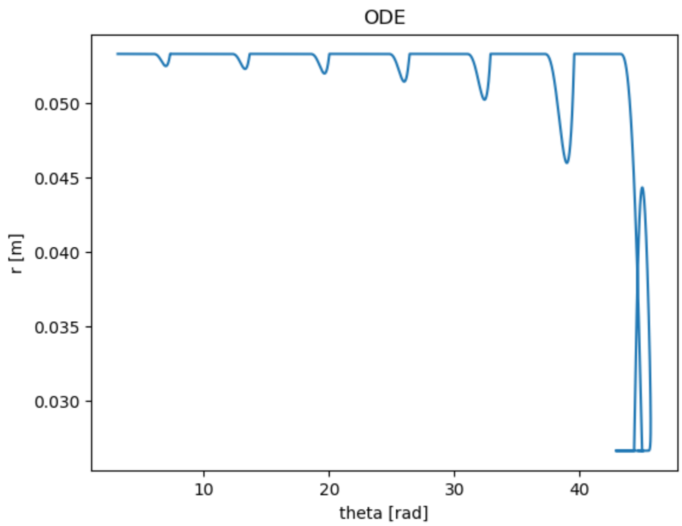


Figure 36: How the variable r evolves over θ.

8. Rectilinear movement with $r_{flywheel}$ free

In this section we have developed the Lagrange Mechanics for the system with one mass of the flywheel being free (i.e. Movable along a rail).

The generalized coordinates (q) will be:

1. $\phi_{ground-wheel}$: rotation of the wheel respect the ground.
2. $\phi_{wheel-platform}$: rotation of the platform respect the wheel.
3. $\phi_{platform-flywheel}$: rotation of the flywheel respect the platform.
4. r: distance from the center of the flywheel to the free cylinder.

We will use two auxiliary variables:

1. $\phi_{ground-platform} = \phi_{ground-wheel} + \phi_{wheel-platform}$: rotation of the platform respect the ground.
2. $\phi_{ground-flywheel} = \phi_{ground-platform} + \phi_{platform-flywheel}$: rotation of the flywheel respect the ground.

The total potential energy:

$$V = m_{cylinder} \cdot (r - r_{flywheel-max}) \cdot cos(\phi_{ground-flywheel}) \cdot g \tag{29}$$

The total kinetic energy:

$$T = \frac{1}{2} \cdot [\dot{\phi}^2_{ground-wheel} \cdot I_{wheel} + \dot{\phi}^2_{ground-platform} \cdot I_{platform} + \dot{\phi}^2_{ground-flywheel} \cdot I_{flywheel}(r) + \dot{\phi}^2_{ground-wheel} \cdot r^2_{wheel} \cdot m_{total} + \dot{r}^2 \cdot m_{cylinder}] \tag{30}$$

The Lagrangian is defined as:

$$L = T - V \tag{31}$$

Lagrange's equation is:

$$\frac{d}{dt}(\frac{\partial L}{\partial \dot{q}_j}) = \frac{\partial L}{\partial q_j} + F_j \tag{32}$$

So in our case:

$$\frac{\partial L}{\partial \dot{\phi}_{ground-wheel}} = \dot{\phi}_{ground-wheel} \cdot I_{wheel} + \dot{\phi}_{ground-platform} \cdot I_{platform} + \dot{\phi}_{ground-flywheel} \cdot I_{flywheel}(r) + \dot{\phi}_{ground-wheel} \cdot r^2_{wheel} \cdot m_{total} \tag{33}$$

$$\frac{\partial L}{\partial \dot{\phi}_{wheel-platform}} = \dot{\phi}_{ground-platform} \cdot I_{platform} + \dot{\phi}_{ground-flywheel} \cdot I_{flywheel}(r) \tag{34}$$

$$\frac{\partial L}{\partial \dot{\phi}_{platform-flywheel}} = \dot{\phi}_{ground-flywheel} \cdot I_{flywheel}(r) \tag{35}$$

$$\frac{\partial L}{\partial \dot{r}} = \dot{r} \cdot m_{cylinder} \tag{36}$$

We define M as the following matrix:

$$\begin{pmatrix} I_{wheel} + I_{platform} + I_{flywheel}(r) + r_{wheel}^2 \cdot m_{total} & I_{platform} + I_{flywheel}(r) & I_{flywheel}(r) & 0 \\ I_{platform} + I_{flywheel}(r) & I_{platform} + I_{flywheel}(r) & I_{flywheel}(r) & 0 \\ I_{flywheel}(r) & I_{flywheel}(r) & I_{flywheel}(r) & 0 \\ 0 & 0 & 0 & m_{cylinder} \end{pmatrix} \tag{37}$$

In matrix form and with our generalized coordinates:

$$\frac{\partial L}{\partial \dot{q}} = M \cdot \dot{q} \tag{38}$$

$$\frac{d}{dt}(\frac{\partial L}{\partial \dot{q}}) = M \cdot \ddot{q} + \dot{M} \cdot \dot{q} \tag{39}$$

Let's recall the definition of $I_{flywheel}(r)$

$$I_{flywheel}(r) = m_{cylinder} \cdot r^2 + C \tag{40}$$

Compute the derivative

$$\dot{I}_{flywheel}(r) = 2 \cdot m_{cylinder} \cdot r \cdot \dot{r} \tag{41}$$

We compute $\dot{M}$ as the following matrix:

$$\dot{M} = \dot{I}_{flywheel}(r) \cdot \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{42}$$

Let a be::

$$\frac{\partial L}{\partial q_{1..3}} = a = m_{cylinder} \cdot (r - r_{flywheel-max}) \cdot g \cdot sin(\phi_{ground-flywheel}) \tag{43}$$

Let b be:

$$\frac{\partial L}{\partial r} = b = \cdot m_{cylinder} \cdot r \cdot \dot{\phi}_{ground-flywheel}^2 - m_{cylinder} \cdot g \cdot cos(\phi_{ground-flywheel}) \tag{44}$$

$$\frac{\partial L}{\partial q} = \begin{pmatrix} a \\ a \\ a \\ b \end{pmatrix} \tag{45}$$

So using Lagrange's equation we get:

$$\boxed{M \cdot \ddot{q} + \dot{M} \cdot \dot{q} = \begin{pmatrix} a \\ a \\ a \\ b \end{pmatrix} + F} \tag{46}$$

9. Components

In this section we will explain all the components used during the building of the robot and how to get them.

9.1 Mechanical components

These components do not have any electronics and the important thing about them is their mechanical function.

9.1.1 Central Body

Used to host the flywheel, a motor and a bearing. It also has lateral tabs so it can easily be joined with the lateral body part. It's 3D printed and in the same orientation you see in figure 37.

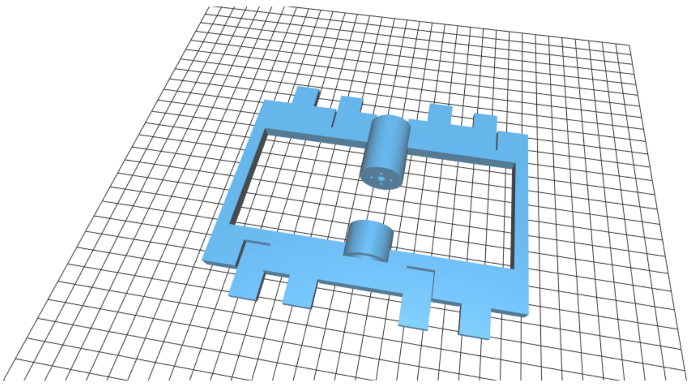


Figure 37: 3D Model of the central body.

9.1.2 Lateral Body

Used to host the wheel motor and the rest of the electronic components. It also has lateral tabs so it can easily be joined with the central body part. It's 3D printed and in the same orientation you see in figure 38.

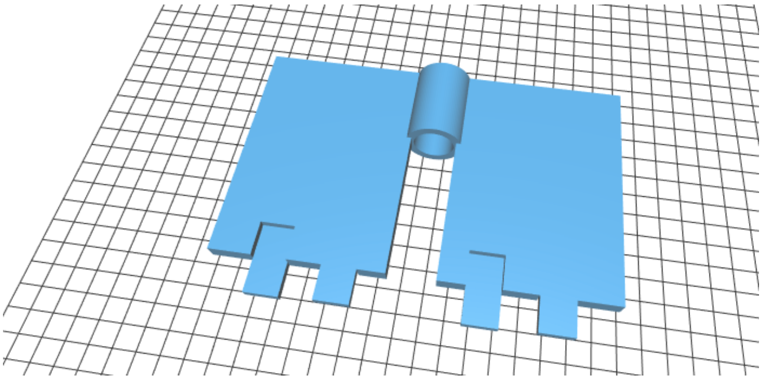


Figure 38: 3D Model of lateral body.

9.1.3 Wheels

The wheels are 3D printed. The orientation we used to print them is putting the motor axis vertical. Each wheel is produced by two symmetric parts that are glued together around the motor.

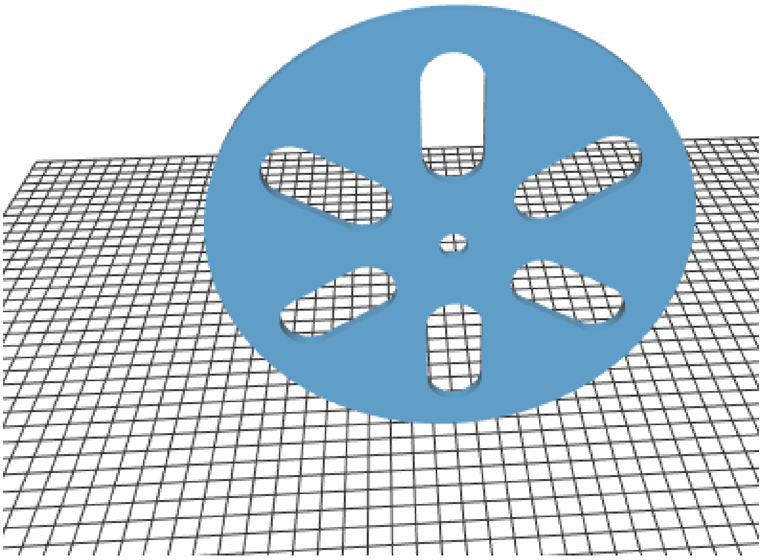


Figure 39: 3D Model of a wheel.

9.1.4 Flywheel

The flywheel is composed by 3 parts:

1. 3D printed wheels very similar to the one in figure 39
2. Steel bolts, washers and nuts.
3. Metallic axis.

9.1.5 Bearings

We are using angular balls bearing as the one seen in the figure 40 to surround the flywheel axis. This bearing is tight fit in to the 3D printed central body. The main purpose of the bearing is to support the flywheel metal axis and help the motor with the flywheel weight stress.

Figure 40: Photo of a bearing.

9.1.6 Reinforcements

After we built the first prototype, we saw that the robot was suffering from bending. So we decided to reinforce the structure with additional 3D printed tabs and two aluminum 'U' profiles to solve the problem.

9.2 Electronic components

9.2.1 Raspberry Pi

The Raspberry Pi is a small single-board computer. We are using Raspberry Pi 3 Model B. It has some features that we will use:

- GPIO pins: General Purpose Input/ Output pins are used to communicate with the motors and other peripherals.
- Ethernet connection: We used to be able to set up the Wi-Fi's connection.
- Wi-Fi's connection: We used to be able to control the robot without cables.
- 5V and 3V lines: It's useful to get the 3V to power the encoders.
- DC's hardware power: So we can power the Raspberry the same way as the other electronic components.
- I2C: I2C is a serial protocol for two-wire interface to connect low-speed devices like micro-controllers. We use it to communicate the Raspberry Pi

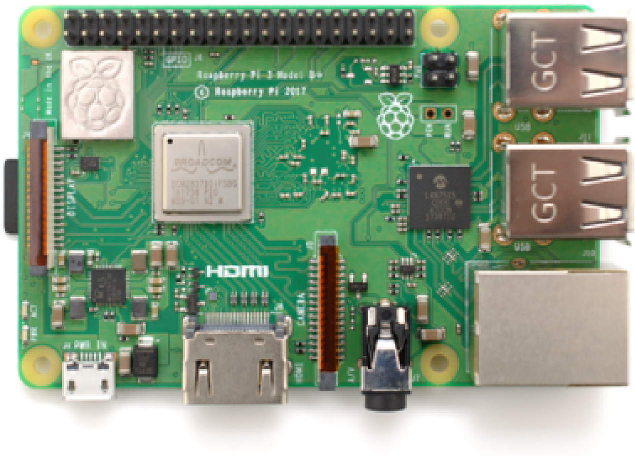


Figure 41: Raspberry Pi picture

Weight	42 g
Price per unit	35 euros
Number of units	1

9.2.2 Batteries

We are using four identical batteries to power our system. They are commercial external power supplies for smartphones. They can be recharged via a micro-USB. We use USB cables to transmit the power to an outlet that connects two batteries in series that power the DC Bridges. These bridges are powering the motors and the Raspberry.

These batteries do not always work as expected because they have integrated security measures that disconnect them when they detect that not power is being used.

In the other hand they incorporate many features as battery level indicator, incorporated recharge mechanisms and on/off buttons.

Weight	200 g
Price per unit	10 euros
Number of units	4

9.2.3 Printed Circuit Board

In this project we tested the basic electronic with a breadboard, once we knew the electronic components where working we decided to transfer our circuit to a Printed Circuit Board (PCB). In particular, we used a Double Sided PCB that allows routing traces in both sides.

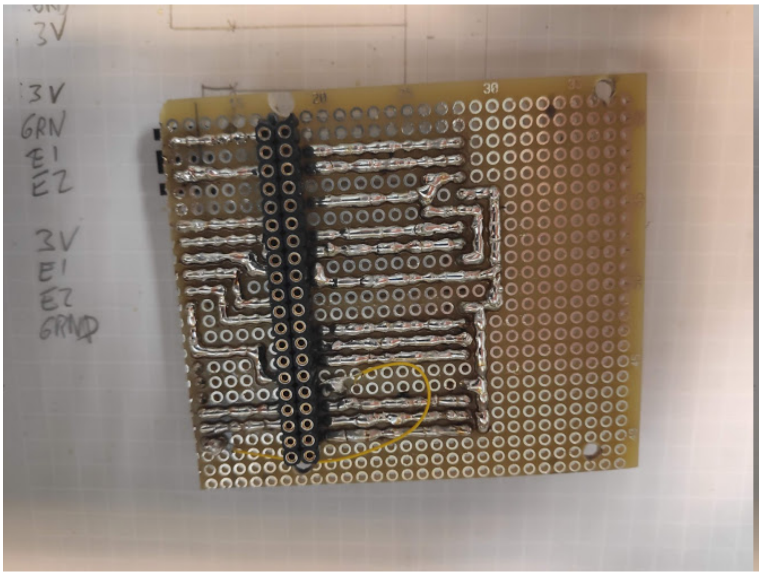


Figure 42: Printed circuit board

9.2.4 DC Motor

A DC Motor is a rotary electric machine that transforms electrical energy (in the form of direct current) into mechanical energy through electromagnetic interactions.

Here are the specifications of our three motors:

Operating voltage	between 3 V and 9 V
Free-run speed at 6 V	176 RPM
Free-run current at 6 V	80 mA
Stall current at 6V	900 mA
Stall torque at 6V	5 kgcm
Gear ratio	1:35
Reductor size	21 mm
Weight	85 g
Price per unit	10 euros
Number of units	3

Figure 43: DC Motor

9.2.5 H Bridge

An H bridge is an electronic circuit that switches the polarity of a voltage applied to a load (in our case DC motors). This allows the motor to go forwards and reverse.

Also, it has the possibility to modulate the output power through a PWM signal.

Our particular H Bridge (L298N) is capable of supporting two motors at the same time. This means that it has 6 input signals, 2 PWM (one for each motor) and 4 enable signals (2 for each motor).

The input power for the H bridge is 10V but it has losses and it delivers around 8,5 V to the motors. Also, it has a 5V power output that we used to feed the Raspberry Pi.

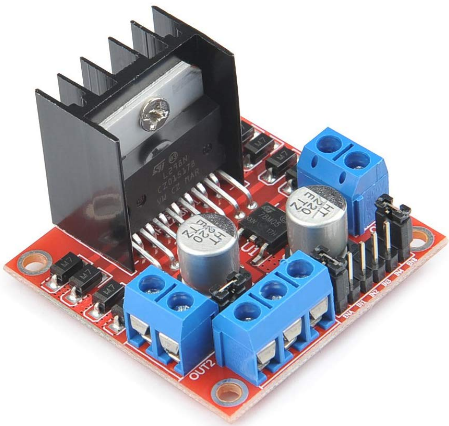


Figure 44: H Bridge

9.2.6 Rotatory Encoder

A rotary encoder, also called a shaft encoder, is an electro-mechanical device that converts the angular position or motion of a shaft or axle to analog or digital output signals.

Our encoder has two signals, channel A and channel B offset by 90 degrees (in quadrature). The direction of rotation can be determined by which channel is leading. If channel A is leading, the direction is taken to be clockwise, and if channel B is leading, the direction is counterclockwise.

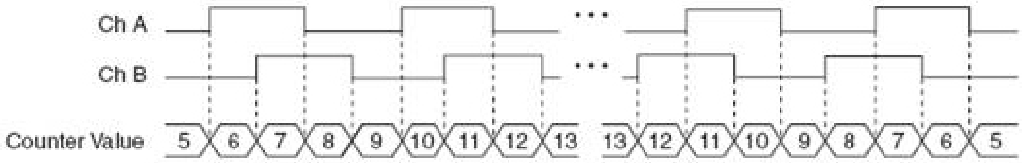


Figure 45: Encoder Signals. Image credit: National Instruments Corporation.

We programmed an interruption that fires every time a channel changes the state. It increments (or decrements) a counter, so we can then compute the position and the speed of each motor.

9.2.7 Accelerometer

An accelerometer is a device that measures proper acceleration. The one we are using is MPU-6050. We use the acceleration measured by the accelerometer to know in which inclination is the platform. The accelerometer is connected to the Raspberry Pi via I2C.

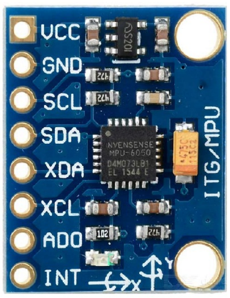


Figure 46: Pictures of a MPU-6050

10. Control

So far we have analyzed our system as a continuous system. However, we are using the Raspberry Pi to control the robot, so we will be using digital control. It's a branch of control theory that uses digital computers to act as system controllers.

In order to sample the position and the speed we use the rotatory encoders. They count the number of flags so there is a constant to convert them into radians.

All our digital control was made in periodical loops. To get the speed we divided the incremental position by the period of the loop. All the controllers are PID controllers, each of time with different inputs, constants and outputs.

A PID controller continuously calculates an error value $e(t)$ as the difference between a desired set point and a measured process variable and applies a correction based on proportional, integral, and derivative terms (denoted K_p, K_i, and K_d respectively).

To adjust the controller constants in all cases we did the following way:

1. Set all gains to zero.

2. Increase the K_p gain until the response to a disturbance is steady oscillation.

3. Increase the K_d gain until the oscillations go away (i.e. it's critically damped).

4. Repeat steps 2 and 3 until increasing the D gain does not stop the oscillations.

5. Set K_p and K_d to the last stable values.

6. Increase the K_i gain until it brings you to the set point with no oscillations desired.

10.1 Position control

The input variable is the angle we would like the motor to achieve. We sense that angle with our rotatory encoder. The output is a PWM signal and the direction the motor has to move.

The first steep was to control the position of a peg attached to the motor. This allowed us to move the rotor from outside and also check and adjust our encoder. In figure 47 you can see a picture of the set up we had.

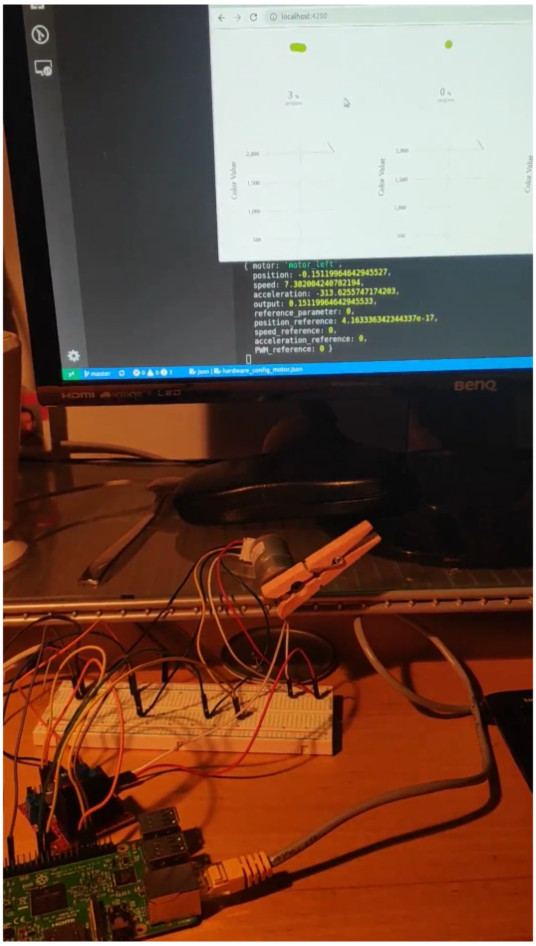


Figure 47: Picture of the set up to control the peg.

Once knowing the constants that worked well for our peg, we proceed the same method with the flywheel position but having a starting point from the peg experiment.

10.2 Speed control

The input variable is the desired speed. We get the real speed by dividing the incremental angle by the period (50 ms). This leaded to noisy measurements, so we decided to apply an average filter (3 samples) to the result.

We use the motor speed control to be able to control the direction in which the robot is moving. We transform a two-axis joystick signal (x,y) into speed set points for the PID of the wheels with the following formula:

$$\begin{aligned} Speed_{right} &= x + y \\ Speed_{left} &= x - y \end{aligned} \tag{47}$$

10.3 Inclination control

In order to control the inclination of the platform we also used a PID controller. We sense the inclination of the platform with the accelerometer, and then we compute the error as the difference between desired inclination and sensed inclination. Them, the we feed the

- In the flywheel case we simply output this error to the flywheel PWM motor signal.
- In the pendulum case this output is mapped to domain from -90 degrees to 90 degrees. Then this signal feeds another PID controller to control the position of the pendulum. We know the position of the pendulum by adding the inclination we sense to the rotation of the flywheel motor.

11. Software Design

We designed the software as modular and reusable as we could, having in mind possible future expansions of the project. The code is divided in three main blocks: user interface (UI), planner and controller.

The UI sends the user commands to the planner, and then the planner translates these commands to position or speed set points and sends it to the controller. Then the controller runs the control algorithms and reports back the sensors measurements.

The reality is that the planner workload now is low but in the future it may need to compute harder problems, as image processing, artificial intelligence or multi-robot coordination.

These three blocks are now running in three different computers but can run in any configuration if needed.

You can find all the code in the GitHub repository `https://github.com/tarragoesteve/TFM` under the software folder.

11.1 User Interface

The user interface is a 3 column layout, the left column is for the left motor, the central column for the platform inclination and the flywheel motor and the right one for the right motor. As you can see in figure 48, each column has three elements, a round progress bar to indicate the set point, a radio selection to select which parameter you wish to command and a plat of the sensor readings.

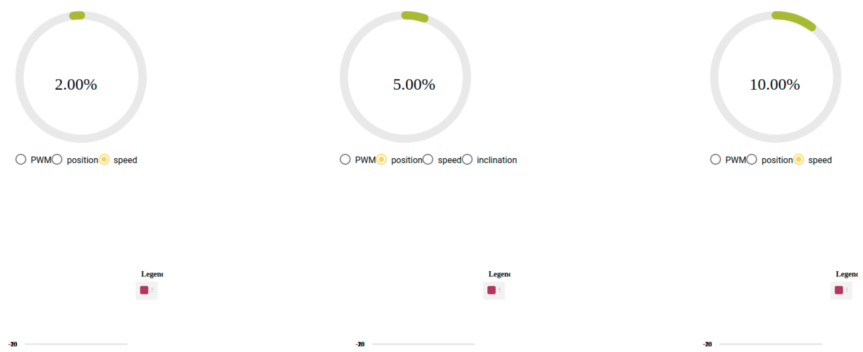


Figure 48: Screen shoot of the UI.

The user interface can be controlled by the keyboard or by a joy-stick connected to the computer. In our case we used a PS4 remote controller connected via bluetooth to the computer.

The whole interface was programmed with Angular, a TypeScript framework designed to code user interfaces. It is organized in modules and packages that one can install, as the round progress bar. An Angular Server is running in the client machine that can connect to the interface via website.

11.2 Planner

The planner main job is to receive and send messages through sockets to the UI and the controller. Sockets is a method to communicate different processes across the internet with very low latency. The planner is working on a server with a public IP so the other computers know where they have to exchange the information even if the UI or the Controller changes its IP.

The planner is now working mainly as a communication module but could be to plan robot moves based on the user input or any algorithm.

11.3 Controller

The controller is running on the Raspberry Pi. It has a main file (controller.js) that reads a configuration file (config.json) and creates instances of the components described on the configuration file. Once all components are created, it starts different threads for each of the components.

The different components we have: accelerometer, led, motor and stabilizer. Also, we have two auxiliary classes that are not components: PID and average filter. All the components that rely on hardware are using C++ code to optimize the code. This code is wrapped in TypeScript classes to make an easier integration of sockets and JSON files.

11.3.1 LED

This was the first components that we made to test the software. It has a GPIO pin as a parameter that it's where we connect the LED. Then the main loop changes

11.3.2 Motor

11.3.3 Accelerometer

11.3.4 Stabilizer

12. Conclusion

Thinking, designing and building a prototype is a big and rewarding challenge. There are many things that do not work as expected and you need to iterate again and find a solution to the problem. Some examples are too much deformation in the platform, too noisy speed measurements or 3D printed parts that we couldn't unpack.

On the other hand we came out with a model using the fundamental mechanics equations, but we are missing some frictions and other forces that do not appear in our model. This made our robot behave a little bit different from expected from simulations. With the flywheel mode the robot can hardly start moving and with the pendulum mode the platforms need to turn a little to start and break.

For a possible following iteration I would recommend much more powerful motors and a more robust structure. The robot now is not behaving perfectly but can move around more or less as expected, as you can see in this video (https://youtu.be/yWOIyIL4cP0).