MARS Project

Marie Sklodowoska-Curie Individual Fellow

Grant agreement number n° 842147

Report: Lego MARS model

Realization of a toy model with Lego Mindstorm EV3 showcasing the basic physical principle of cantilever mass sensing.

Authors:

João Mouro, Bruno Tiribilli

Institute for Complex Systems, National Research Council (ISC-CNR),
Via Madonna del Piano 10 Sesto Fiorentino Firenze
joao.mouro@fi.isc.cnr.it , bruno.tiribilli@isc.cnr.it

This project has received funding from the European Union’s Horizon 2020 research and innovation program under the Marie Slodowska-Curie grant agreement No 842147

LEGO MARS Model

Introduction

Playing with LEGO we assembled a simple working model of a mass sensor where the resonant frequency dependence on the added mass can be clearly appreciated. The model has been used to explain, in a simple and accessible way, the physical phenomena at the base of cantilever sensing that is at the base of our research project MARS.

Description of the model

The model has been assembled using LEGO Mindstorm EV3 components, from a standard "Educational" set (#45544). Main components are:

- EV3 Intelligent Brick (# 95646)
- Large Motor (# 95658)
- Colour Sensor (# 95650)
- Pressure sensor (# 95648)

All components are positioned on a 48 x 48 LEGO platform, obtaining a stable and easily transportable setup (Figure 1).

In the MARS model the micro-cantilever is substituted with a metallic bar (350 mm x 20 mm x 0,5 mm) fixed at one end to a heavy metallic holder next to the platform. The "cantilever" can oscillate in a horizontal plane.

The detection of the cantilever's movements is performed by a simple optical method. A "Color Sensor" (code # 9694) is placed in front of an area of the bar made reflective with a piece of aluminium tape, close to the free end. (see Figure 1). The reflected light intensity strongly depends on the distance between the sensor and the bar.

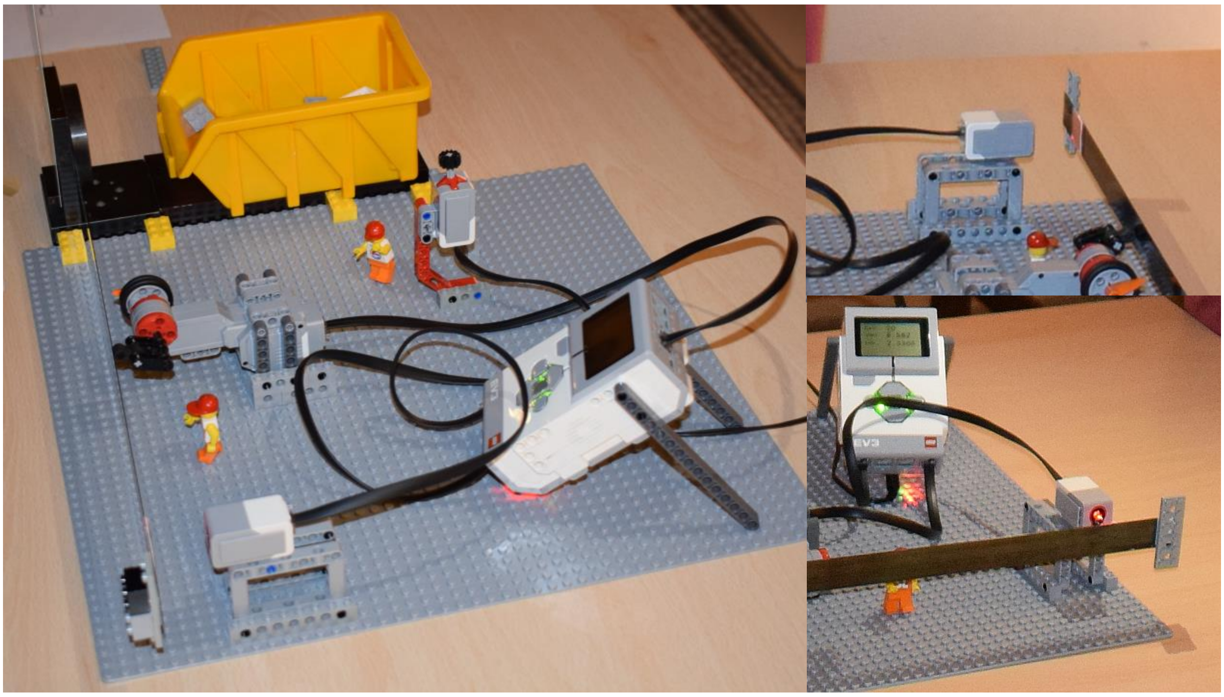

Fig 1 - The Lego Mars model. The reflected light sensor is positioned in front of the reflecting area of the cantilever. In the left image some bricks are fixed at free end as added masses.

A larger signal is obtained when the bar is near the light sensor, and it decreases as the bas moves far. At the end of the cantilever a LEGO base (6 x2) is glued, it can support the added masses constituted by some bricks. In the experiment we used a number of 2 x 2 bricks (code # 3003), each weigh 1,192 grams.

A rubber joint is attached to a motor (Fig 1) and positioned in such a way to move the cantilever away from its position at rest. A half turn clock-ways rotation of the motor release the cantilever that start to oscillate. This motor movement is activated by the operator when s/he press a button (pressure sensor) at the beginning of the measurement.

After a fixed number of oscillations (controlled by the EV3) the display shows the measured frequency then the motor rotates 180 degrees counter-clockwise, the cantilever is gently stopped, back in the starting position, ready for the next measurement.

Coding

The EV3 intelligent brick receives signal from the two sensor and control the motor, it manages the experiment thanks to a simple code. (Free software, LEGO® MINDSTORM EV3). The full program is shown in Fig 3.

At the beginning, as soon as the pressure sensor is pushed, the motor release the cantilever and it starts oscillating. Then the code enters a loop in which it looks at the signal from the reflected light sensor waiting until the value goes below a threshold "*DownLev*", then it waits until the value overcome an "*UpLev*" and the cycle is repeated a number of times, 20 in our case.

The two values (50 and 35 in figure 2) are selected near the mean value of the reflected light signal and fixed at the beginning of the code. This simple algorithm allows detecting the repetitions in the periodic signal from the reflected light sensor.

The total time interval spent to complete 20 cycles is used to calculate the oscillation frequency. In the final part, the results (number of cicles, time, frequency) are shown as output on the EV3 display, then the program moves the motor to dampen the cantilever and stop.

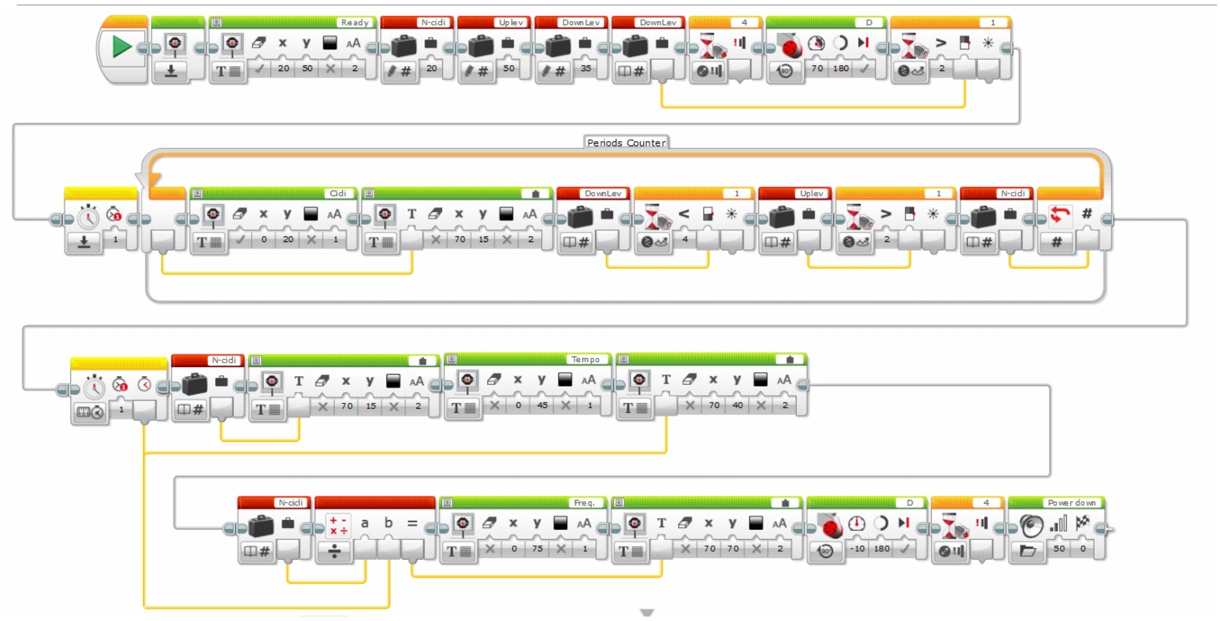

Fig 2 - Code of the Lego MARS model. The loop in the central section acquires the signal from the colour sensor operating in "reflected light" mode, performs comparison with the two threshold levels "DownLev" and "UpLev" to counts the cantilever oscillations . After 20 cicle are completed the program exits the loop.

Experiment

A measurement consist of adding a number of bricks on the support at the free end of the cantilever, run the program, start the oscillations and wait until 20 periods are detected. At the end the EV3 display will show the measured frequency value.
This result is directly related to the added masses. Presenting this physical effect in a simple and fun way is exactly the purpose of this LEGO model.
We performed several measurements adding up to 12 bricks. In Tab. 1 the results of theese simple experiments are reported: the number of added bricks, their weight and the corresponding measured frequency.

Table 1. Mass increment and frequency

Bricks	m_A (g)	f_A (Hz)
0	0	2,3313
1	1,192	2,2036
2	2,384	2,0868
3	3,576	1,9920
4	4,768	1,9008
5	5,960	1,8240
6	7,152	1,7615
7	8,344	1,6997
8	9,536	1,6437
9	10,728	1,5929
10	11,920	1,5475
11	13,112	1,5055
12	14,304	1,4647

It worth to remember that the frequency will change with added mass according to:

$$f_A = \frac{1}{2\pi}\sqrt{\frac{k}{m_A + m_{levetta}}} \quad (1),$$

where k is the elastic constant, m_A is the added mass . $m_{levetta}$ is the effective mass of the cantilever that is 0.24 the total mass (38.8 g).
From eq. 1 the value of the added mass can be calculated.

$$m_A = \frac{k}{4\pi^2 f_A^2} - 0.24 m_{levetta} \text{........(2)}$$

In Figure 3 a plot of experimental results presented in table 1 is shown together with a fitting curve. The coefficient A and B, extracted from the fitting, are related to the elastic constant k and the effective mass of the cantilever. The fitting equation and the coefficien are also shown in figure 3.

This curve graphically illustrated the physical relationship expressed in equation 2.
During an exhibition this curve has been used to show to the audience how a measured frequency value corresponds to the added mass, with excellent agreement and sensitivity.

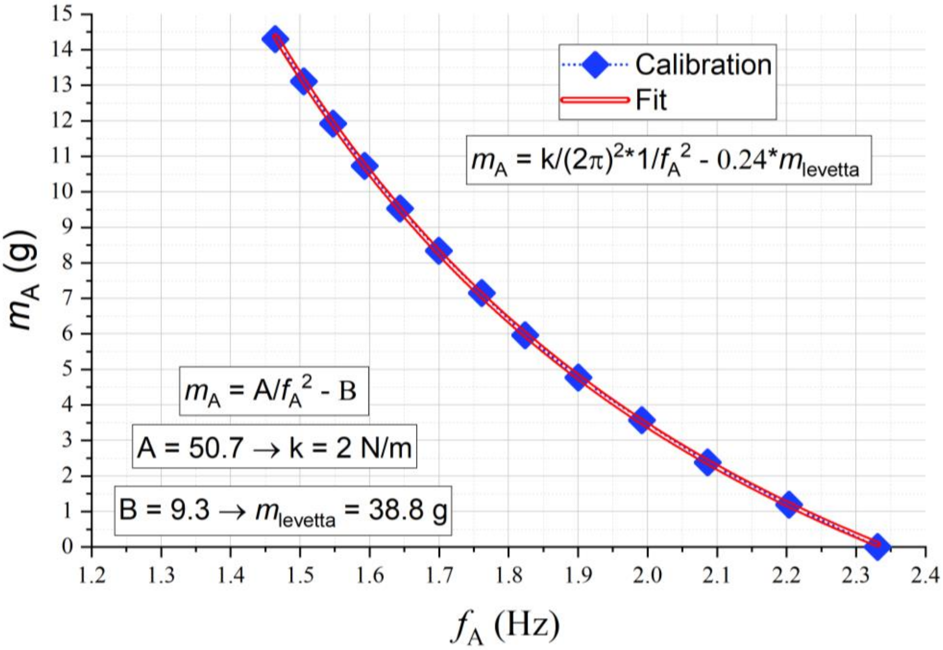

Fig. 3 Plot of experimental results (blue diamonds) presented in table 1 together with a fitting curve (in red). The elastic constant k and the effective mass of the cantilever are derived from fitting coefficients A and B.

Events and Dissemination

During the model construction attention has been paid to make evident the important aspects of the cantilever behaviour. The movements of the steel bar representing the microcantilever can be easily followed by naked eye. Sounds from EV3 are used to outline start and stop of the measurement. The number of cycles is updated continuously. The measured frequency is shown on the display in large font, readable by the audience.
Even a short presentation of a few minutes can include several measurements with different weights.
All these features facilitate presentation to students visiting the laboratory but also to adults and children.

Partecipation to "Bright Night - Notte dei Ricercatori" 24 settembre 2021

On the occasion of the Researchers' Night 2021 the CNR organized a dissemination event called "Bright Night - Notte dei Ricercatori". The event took place on the evening September 24, 2021, in Sesto Fiorentino. Due to Covid19 rules the number of attending people was limited to 80.

We contributed to this exhibition with a stand entitled: "Una Bilancia microscopica" (A Microscopic Balance) presenting experiments with our MARS model made out of Lego to explain the operating principle of the micromechanical device which is the key component of our project (Fig. 4).

A poster was used to introduce the main topics of the project and its scope: what is a cantilever; frequency shift due to mass or viscosity variations. Appropriate reference to EU funding was shown in the poster.

During the presentation it was pointed out that decreasing the size of the cantilever the sensitivity increases therefore microscopic size sensors such as the one in the MARS project can be used in many practical applications to detect extremely small masses.
The explanation of the experiment takes about 10 - 15 minutes in total and was repeated several times during the evening.

Some examples of measurements are available at:

1 - video 2021 video test 12 bricks   https://youtu.be/Jbs0nub-zlQ

2 - video 2021 video test 8 bricks   https://youtu.be/4I01fWyYUAo

Figure 4 Presentation of the Mars project at the “Bright Night - Notte dei Ricercatori, 2021”

The BRIGHT NIGHT 2021 event on 24 September 2021 was_video recorded by local TV broadcasters RTV38 and SestoTV.

Collaboration with the University of Florence, School of Engineering

The implementation of the feedback loop in the Lego MARS model has been the subject of an activity in collaboration with the University of Florence.

Three students, Lorenzo Bartemucci, Tommaso Borghesi and Nico Tiezzi, of the “Laboratory of Automatic Control” course dedicated their activity to develop a controlled feedback loop for the Lego MARS model.

Starting from the model described above they replicate the software using MATLAB Simulink, then they modified the action of the motor in order to obtain a continuous excitation of the cantilever.

Then a software algorithm has been developed to detect the phase by monitoring the signal from the reflected light sensor. Finally, the feedback algorithm uses this information to control the speed of the motor and maintain the resulting oscillation frequency.

Test have been made to verify the proper action of the feedback and the expected dependence of the frequency from the added mass.

The students described this activity as part of their curricula in the laboratory report entitled "Cantilever's Experience". (see Annex)

A video of this Lego MARSs model in action is available at this link:
https://youtu.be/SPzXKwg4Crs

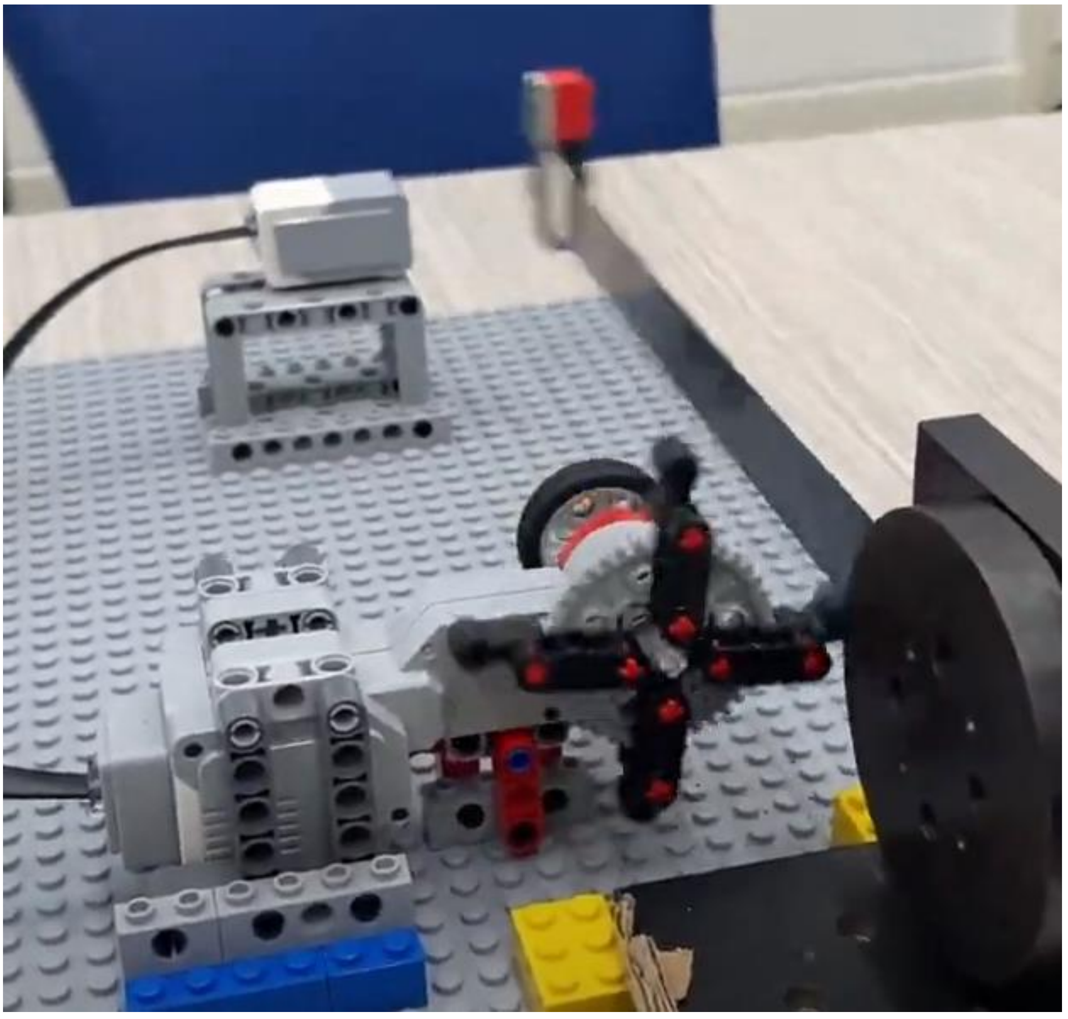

Figure 5 Lego Mars model built at the "Laboratory of Automatic Control" University of Florence with the continuous excitation feedback.

Conclusion

A Lego MARS model was built and shown to adults and children and to a group of students from the University of Florence.
We aroused a lot of interest and got very positive comments from all the attendees.
In the near future we will continue to play with Lego MARS, together with students of various levels, to illustrate in a fun way the operating principle of cantilever sensors.

ANNEX

Cantilever's Experience

Bartemucci Lorenzo, Borghesi Tommaso and Nico Tiezzi

June 17, 2022

Contents

1 Introduction

A cantilever is a type of beam constrained at one end with the other extending freely outwards. In most macro applications, the cantilever is rigid for small movements such as the wings of a jet or a balcony.

In micro-applications, some cantilevers are rigid allowing controlled movement, while others are more flexible allowing variable movements. Flexible micro-cantilevers are used in applications where an external force or intrinsic stress causes the rod to flex or bend (e.g. atomic force microscopes, diagnostic transducers, chemical array sensors). More rigid cantilevers are used as needles, probes or transport mechanisms for probes or transducers.

Several factors influence whether and how a cantilever moves or how it responds to external stimuli. These include its dimensions (length, width, thickness) and the properties of the material from which it is made. The geometrical shape as well as the material used to construct the rod determines the stiffness of the cantilever (how it responds when a force is applied). Microcantilevers are commonly used in microelectromechanical systems (MEMS). Such systems include the following applications:

- Atomic Force Microscopes
- Chemical sensor arrays
- Read/write storage devices
- Olfactory systems
- Environmental Monitoring
- RF switches

Many of these MEMS applications operate the cantilever in static mode or dynamic mode, in other words:

- Static mode is when the cantilever is in a static (stationary) state. Any displacement of the rod due to a load or intrinsic stress generated on or within the cantilever is measured.
- Dynamic mode is when the cantilever is externally actuated by a suitable actuator causing it to oscillate at its natural resonance frequency. Any change in load or mass results in a change in this frequency.

2 Frequency measurement as the mass varies

In the dynamic mode, the amount of material is measured by monitoring a change in the microcantilever's natural resonance frequency. When a dynamic microcantilever is initially excited by an external actuator (such as a piezoelectric, magnetic or electrostatic actuator), it begins to oscillate. The oscillation frequency is usually equal to or close to the natural frequency of the cantilever (or resonance frequency). Any change in the physical characteristics of the cantilever (such as its material, geometry or mass) changes its natural frequency. The resonance frequency is the frequency of a system oscillating at maximum amplitude. Together with little damping, this frequency is usually equal to the system's natural frequency. When a system reaches this frequency, this state is resonance.

When the mass of the system changes, the resonance frequency also changes. For example, when a child bounces on the end of a trampoline, it will oscillate at a frequency determined by the characteristics of the trampoline and the mass of the child. However, if the father and child bounce on the end of the trampoline, the frequency changes due to an added difference in mass.

In our case, we used such a device to observe how the frequency varies as a function of mass. The circuit is based on LEGO MINDSTORMS technology combined with the Simulink simulation programme. It consists of a Lego EV3 brick, a color sensor set to the intensity of reflected light, a servo-motor with integrated encoder, a pushbutton and a metal bar to the end of which Lego bricks are attached. These are gradually added to vary the oscillation frequency of the rod. Also attached to the bar is a layer of polished material on the opposite side so that the light emitted by the sensor is best reflected. The setup of our circuit is shown in the figure 1 below.

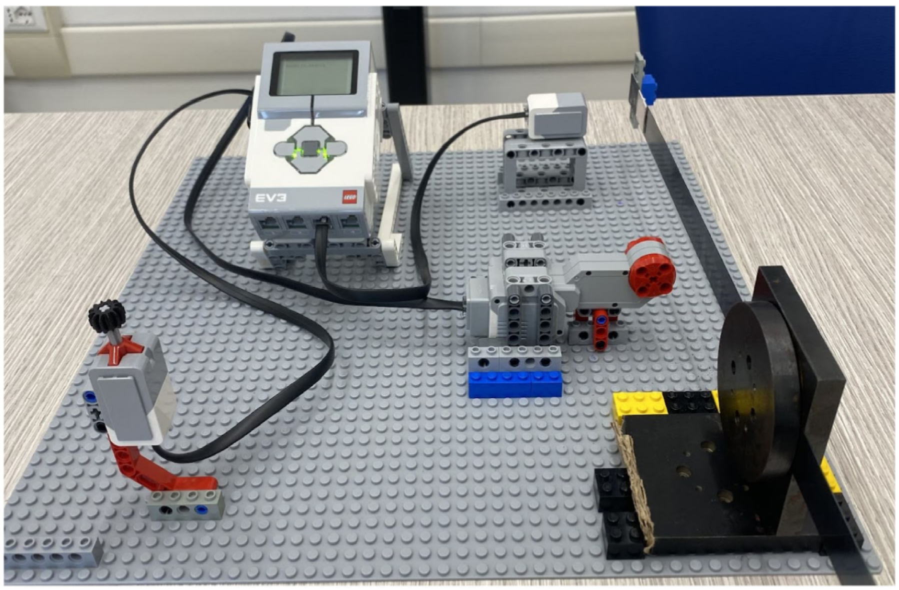

Figure 1: Initial setup

On Simulink, we build, using the block architecture, the circuit as shown in figure 2.

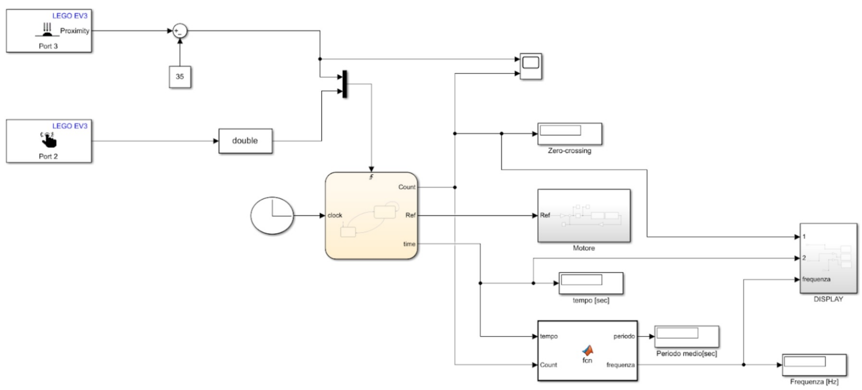

Figure 2: Simulink scheme for frequency calculation.

The operation consists of manually loading the bar (the first time) via a rotary system consisting of a rubber grommet appropriately connected to the servomotor, which sets up a damped harmonic motion as soon as the motor makes a rotation (in our case 180°). The oscillation must then be stopped by repositioning the rubber in its initial position. The stopping criterion involves counting twenty periods of the signal generated by the color sensor which, on Simulink, is represented by the block *Infrared Sensor* in *Proximity mode*. At the end of this count, the elapsed time is calculated and the command is sent to the motor to return to the initial position. All these operations are performed within the Chart (figure 3).

The initial *Stand* state is one in which the motor is stationary and assumes zero angular initial position (**Ref = 0**). As soon as the button defined by the **Button == 1** event is pressed, the reference becomes 180 and will remain so until the period count, as the **Count** variable has been initialized, is less than twenty. This is cyclically updated each time the signal completes a period. The **Period** event is triggered by the output signal from the *Infrared Sensor* every time it goes up. This type of event only works if the signal passes from zero, but because of the way the sensor block was made, it oscillates between 0 and 100, so it was necessary to move downwards by an arbitrary amount, however sufficient for the periods to be read. Once the counting is finished, while we're returning to the initial state, we calculate the elapsed time, computed in the time variable, as the difference between the simulation time when exiting the *Start* state (**clock** variable) and the one at the entry of the *Start* state (**t = clock**). In the output of the Chart we therefore have three quantities: **Count**,

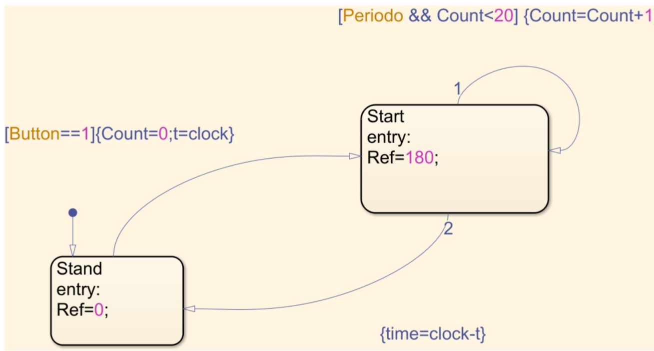

Figure 3: Counting logic inside the Chart block.

Ref and **time**. While **Ref** is used to control the motor in the *Motore* subsystem, the other two variables are used in the *Matlab Function* to calculate the elapsed time in the counting operation and the frequency.

We determined the average period of the signal because, due to quantisation, not all periods are exactly equal. Eventually, this gives the frequency. The reference variable for motor rotation takes on two values: 0 or 180. The negative feedback control system for the motor is shown in figure 4 to control the angular position of the servomotor.

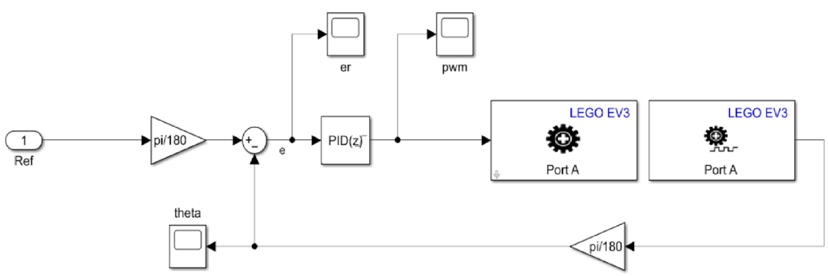

Figure 4: Control motor subsystem *Motor Model*.

The LEGO MINDSTORMS EV3 Motor and Encoder blocks allow us to control and measure quantities such as the speed and rotation of an EV3 motor. The inclusion of the pi/180 gains are justified by the fact that we control the angular variable in radians, and therefore since both the encoder output and the reference are in degrees, the appropriate conversions had to be made. We used a discrete-time proportional integral derivative PID controller, with 'tuned' parameters (tune command), to get the best response in a relatively short time and with little overshoot (table 1). As sampling time, we set Tc=0.005 sec. These values were obtained by using the encoder and servo-motor model shown in Figures 5 and 6 respectively.

P	I	D	N
155.69	360.85	10.5	74.95

Table 1: PID tuning values.

The motor model generates **theta** in radians while the encoder, essentially characterized by a *quantizer* block, generates it in degrees, so the output of the *Motor Model* must be converted appropriately.

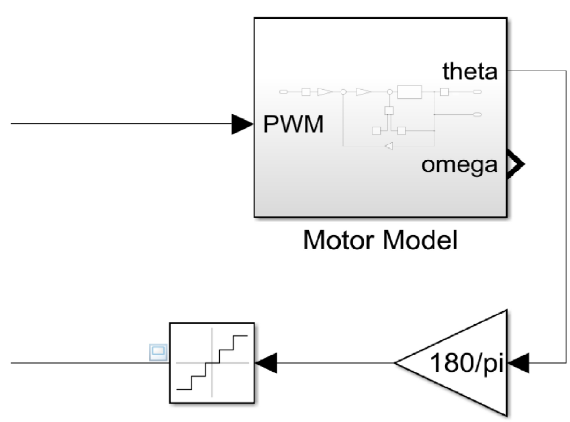

Figure 5: Encoder Model

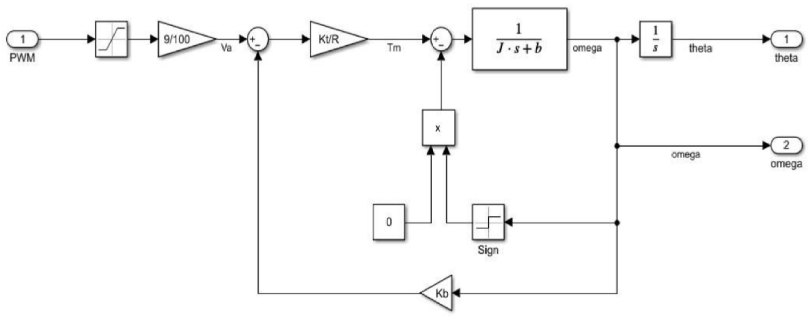

Figure 6: DC Motor Model

The motor model is derived from the armature circuit equations of a DC motor and the dynamics equations below:

$$v_a = Ri + K_b\omega + L\frac{di}{dt}$$

$$T_m = K_t i = T_d + b + J\frac{d\omega}{dt}$$

Combining the expressions results in a single one translated into the block diagram shown in Figure 6. The parameters required for the model are given in table 2. Zero friction torque Td was assumed.

R	$K_b[\frac{Nm}{A}]$	$K_t[\frac{Vs}{rad}]$	$b[\frac{Nms}{rad}]$	$J[\frac{Nms^2}{rad}]$
5.2	0.5	0.5	0.0055	0.0045

Table 2: Representative parameters of the DC motor.

Moving on to the experimental part, we collected frequency values, which could be viewed on the display of the Lego Mindstorms, as a function of the mass applied to the cantilever represented by the number of bricks applied to the end of the bar. For each mass value, the frequency measurement was repeated three times to verify that it was repeatable. Table 3 below shows the test data.

N° di masse	Frequenza n°1 [Hz]	Frequenza n°2 [Hz]	Frequenza n°3 [Hz]
1	4.362	4.329	4.314
3	3.861	3.917	3.879
6	3.367	3.407	3.389

Table 3: Frequency measurements at varying mass.

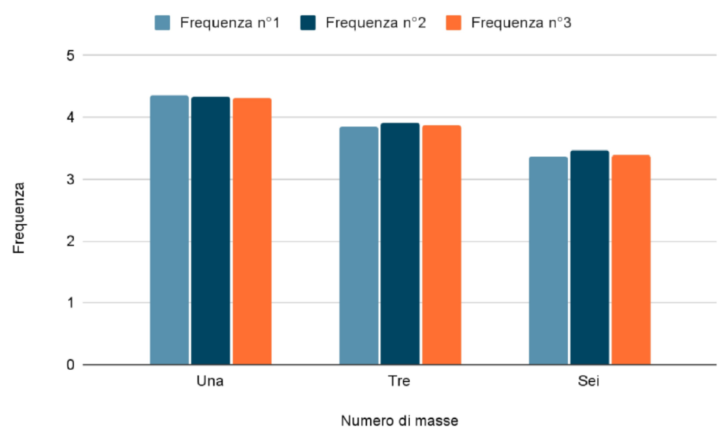

Figure 7: Frequency measurements at varying mass.

It's easy to observe, as can be seen in figure 7, that as the mass increases, the period of oscillation increases (the bar needs more inertia to move, resulting in a longer oscillation time), so the frequency decreases.

The LEGO Mindstorms system allows through Simulink a real-time supervision of the circuit's physical properties that we care about. Figure 8 shows the signal (the red one) read from the proximity sensor, which is the rod's oscillation. A count signal (the blue one) is also there, so that we can see when the Cantilever's oscillation period occurs and which gives us a step graphic ranging from zero to twenty.

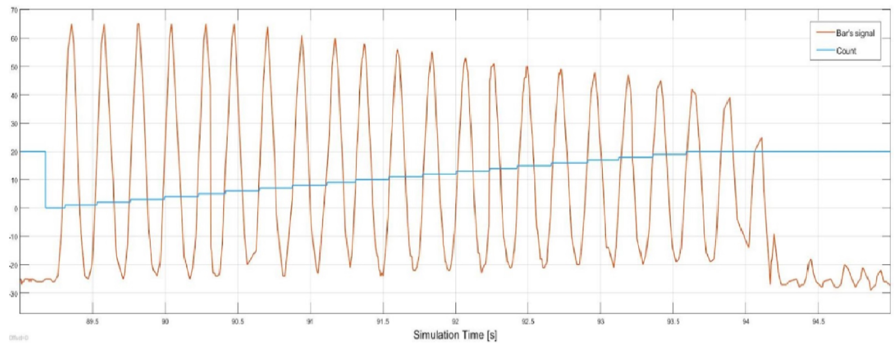

Figure 8: System's excitation

3 Maintaining the resonance frequency

Since the cantilever is a device characterized by an oscillating rod with one end fixed, it will be subject to a certain damping due to dissipative forces. As time passes, these cause the amplitude of the oscillations to decrease until the cantilever has reached its equilibrium position again. As Figure 8 above shows, the cantilever tends to oscillate with a maximum amplitude for approximately six periods, after which it can be seen that the amplitude of the oscillation reduces with each period.

In this second part of our study, we set ourselves the goal of realizing a solution that would allow the system to maintain its resonance frequency, which means, by means of the motor, providing a periodic frequency stress to the rod, equal to the system's own oscillation.

The previous test is somewhat preparatory to this second part of the experiment. In fact, the frequency values found by varying the mass, are in first approximation the natural pulsations of second-order systems. We know that resonance pulsation and natural pulsation are quantities that are very close to each other in numerical terms, so we can use the values obtained in the first part as a reference.

The first important difference is the change in the excitation of our system, no longer one-shot but continuous excitation. Many alternatives were proposed (gearwheel, cam, spring, ...) and the choice fell on the one with which we could have the best realization of the resonance. In our case, we adopted the solution of a wheel with four rubbers, each one offset by ninety degrees (see figure 9).

This choice turned out to be the most performing from a structural point of

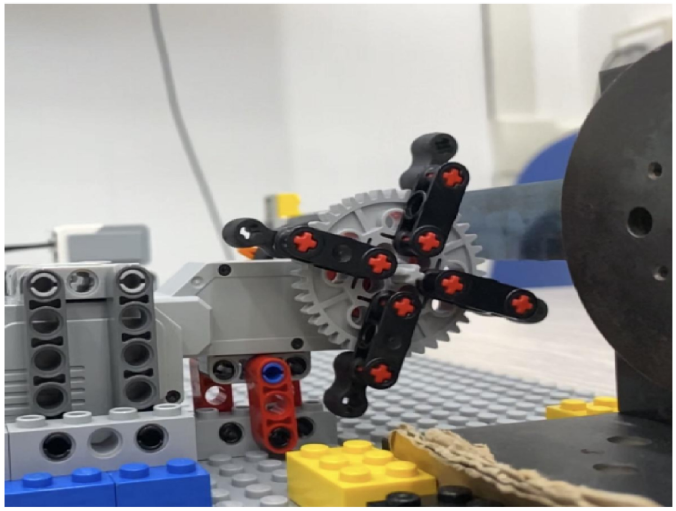

Figure 9: Bar's signal and count graph.

view, guaranteeing greater compactness of the excitation and reducing stress on the motor system plus encoder. Another fundamental aspect was to reason on which quantity to control in order to obtain the correct coupling between the rotating motor and the oscillating bar.

Preliminarily, we thought of a PWM speed control, but this proved unsuitable because each PWM value covers a large number of frequency values, resulting in a not very fine control. However, this first attempt proved useful in giving us some reference data for subsequent simulations. Table 4 below shows the PWM values for the Cantilever resonance as the number of applied bricks varies.

N° di masse	Frequenza di risonanza[Hz]	PWM
0	4.6	-55
1	4.3	-50
2	4	-47
3	3.7	-45

Table 4: PWM values with respective resonance frequency.

We can see, as already seen in the first part, that increasing the number of masses decreases the resonance frequency of the bar and consequently the speed at which the motor must rotate to maintain oscillation. The technique that allowed us to realize the coupling between the rod and the motor was to exploit the phase measured by the servomotor's encoder to perform a feedback control, using the phase of the periodic signal generated by the rod as a reference. It was therefore necessary to be able to extract the phase of the oscillation in some way. In Simulink, the scheme that realizes the resonance is shown below in figure 10. Going from left to right, we illustrate and explain the salient points of the circuit.

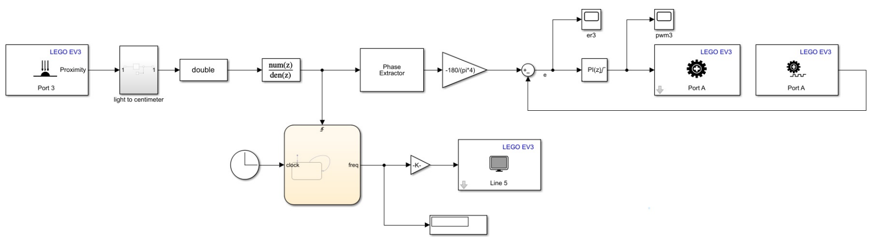

Figure 10: Simulink scheme for resonance.

At the output of the *Proximity* block, the signal passes into a subsystem in which the bar displacement read by the proximity sensor is converted into an actual displacement expressed in centimeters (figure 11).

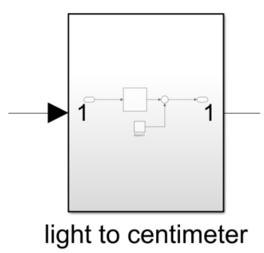

Figure 11: Subsystem for converting to centimeters.

We used a *Look-Up Table* (figure 12), a simulink block which is able to receive the signal coming from the *Proximity* block as input, and which gives as output its displacement in centimeters. Figure 13 shows the trends of the output signal from the sensor and the appropriately converted signal.

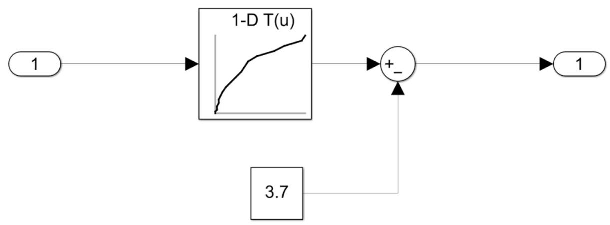

Figure 12: *Look-Up Table* and offset.

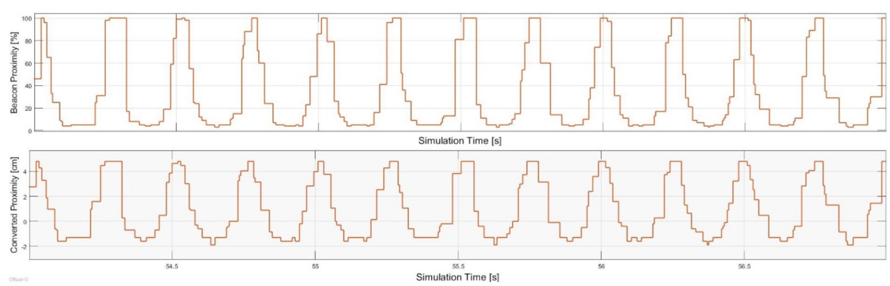

Figure 13: *Proximity* sensor reading (top) and real displacement with offset (bottom).

It was necessary to make direct measurements which we used within the *Look-Up Table* to derive the output as a function of the input. We proceeded by moving the bar from the furthest point, for which the sensor fails to take a reading (value 0), to the nearest point, for which the sensor reaches saturation (value 100). Table 5 shows the data reading:

N° of position	Sensor measurement	*Look-up table* output [cm]
1	0	8.5
2	1	8
3	1.5	7.4
4	1.5	7
5	3.5	6.5
6	4.5	6
7	5	5
8	8	4.5
9	12.5	4.1
10	14.5	3.7
11	18	3.5
12	22.5	3
13	28.5	2.7
14	36	2.4
15	46.5	2.1
16	65.5	1.8
17	76.5	1.5
18	97	1.3
19	100	1.1

Table 5: Direct measurements on the relationship between sensor and actual displacement.

The cantilever at rest is at a certain distance from the proximity sensor, so this will be the reference zero for the bar. This offset is 3.7 centimeters and this explains the sum block in figure 12 above. We therefore obtain a signal oscillating between 4.7 and -2.6 centimeters. As can be seen, perfect symmetry is not achieved because the relationship between the light and the actual displacement is not linear, thus causing asymmetry.

Now consider figure 14. The output of the conversion subsystem, as we know, has a periodic pattern, since it represents the oscillation of the rod. In order to measure the phase of this signal, we used the *Phase Extractor* block, which receives a periodic signal as input and gives its phase as output. In particular, this takes a reading of the signal at each step, starting from zero (every 180 degrees), obtaining a stepped signal. In order to make the incoming signal more compatible with the *Phase Extractor* block, we used a filter to smooth out the quantised signal. The gain placed at the output of the extractor is used to switch the phase - and therefore frequency - of the rod, to the one of the servo-motor.

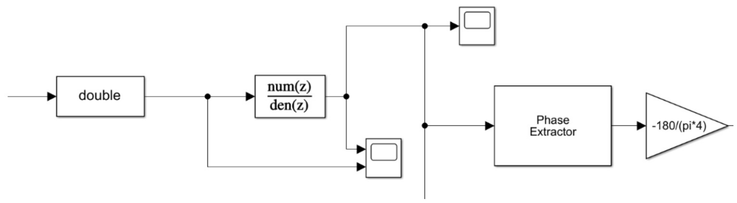

Figure 14: Signal filtering and phase extraction.

The grommets, located on the wheel at ninety degrees to each other, strike the rod four times for each complete rotation of the motor. If the frequency of the rod is to be used as a reference, a multiplication factor of a quarter must be used to compare it with the frequency of the motor.

Advancing from the gain block, the reference signal is sent to the motor system, formed by the LEGO MINDSTORMS EV3 Motor and Encoder blocks. As in the first part, a negative feedback is carried out, and used to compare the phase of the signal with the one measured by the encoder. Control in this case is carried through a PI discrete integral proportional controller with a sampling time Tc of 0.005 seconds. The tracking of the motor phase (blue) compared to that of the rod signal (red) is shown in figure 15.

Table 6 below, shows the values which best stabilize the system's response.

P	I
0.43	0.34

Table 6: PID tuning values.

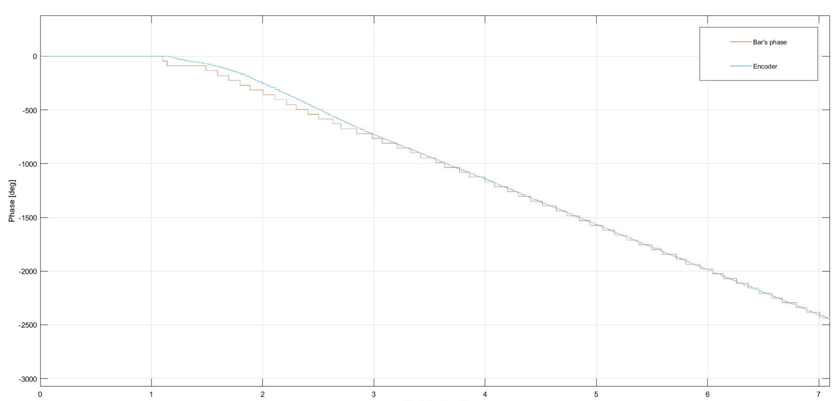

Figure 15: Phase tracking.

It can be seen that the phase of the rod oscillation is highly quantised compared to that of the motor measured by the encoder. The encoder has a sensitivity of one degree, while the *Phase Extractor* has a sensitivity of 180 degrees. Consequently, as shown in figure 16, the steady-state error is not zero but fluctuates around zero, since there is a constant difference repeating over time for positive and negative values. If we look at the previous tracking graph (figure 15), we can see that the phase of the motor centrally cuts the step phase of the oscillation, so we will obtain instances in which it is less than the oscillation and others in which it is greater.

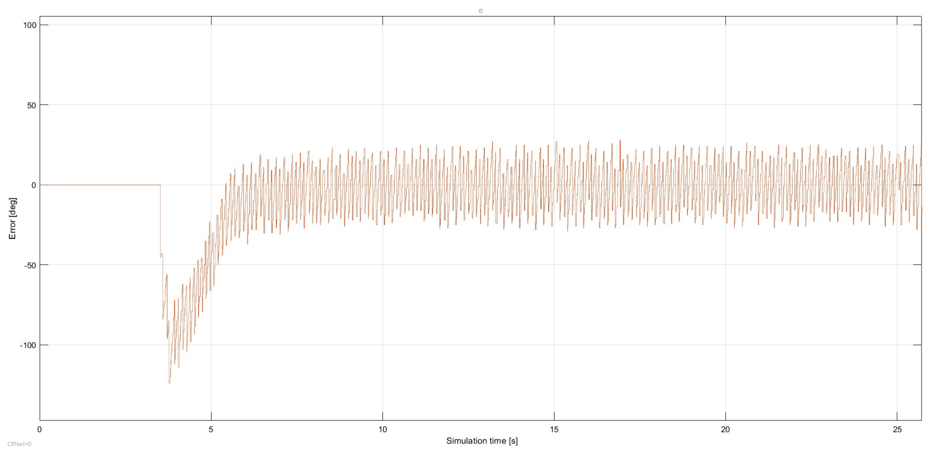

Figure 16: Tracking error.

Using the same operating principle as in the first part of the test, we eventually created a Chart that would allow us to calculate the frequency and represent it on the display of the LEGO brick (figure 17). As matter of fact, the Chart receives the (converted) periodic signal and the clock variable as input. At the output we have the frequency (**freq**) which we supply as input to the LEGO display block so it can be seen on the hardware. Figure 18 shows the logic for calculating the frequency. As with the first part, we have defined an event (**period**) which triggers the *Start* state every time the signal crosses zero on the rise. We calculate the period as the difference between the variable clock and **t** (initially equal to **clock**) and use it within the block to calculate the frequency value at each oscillation of the rod.

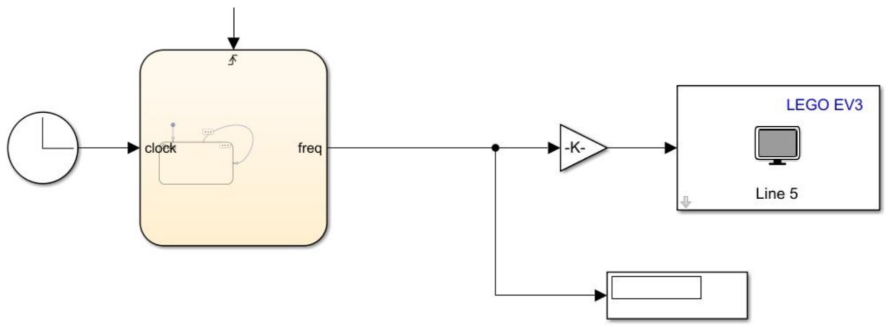

Figure 17: Calculation and display of the frequency.

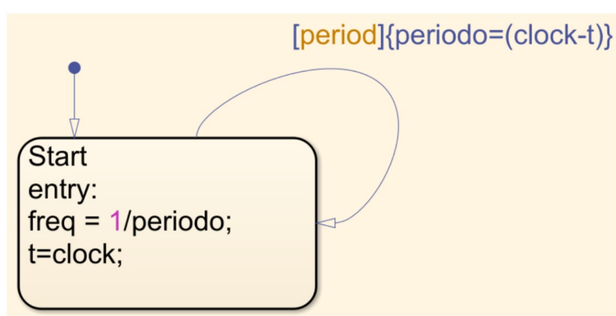

Figure 18: Calculation logic within the Chart.

Finally, we used the experimental test to check whether the PWM values when varying the mass, and thus the system's relative resonance frequency, were consistent with those previously measured in open loop (table 4). Figure 19 shows the trend of the motor speed in the four cases examined (from none to three masses) and we can observe how the values oscillate around those expected.

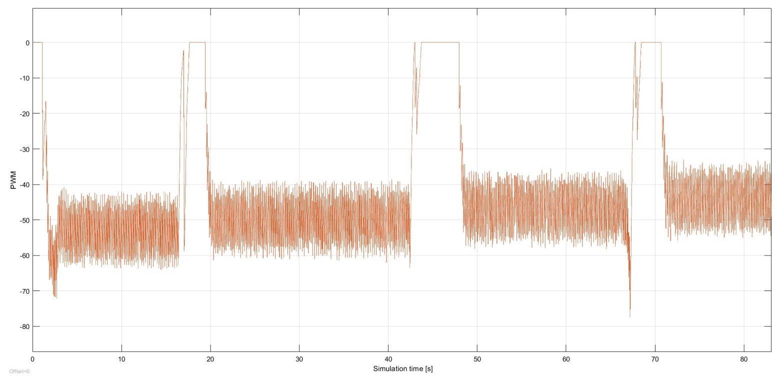

Figure 19: PWM trend as the mass changes.

This scheme just described, as we have already mentioned, was the one that allowed us to control the maintaining of the cantilever's resonance frequency as the mass varies in the best way possible. Moreover, this structure also performs very well when facing disturbances in the oscillation of the rod. If an attempt is made to brake the movement of the rod or even block it, the system reacts and brings it back into resonance, as can be seen in figure 20. There is an increase in the motor speed for a sufficient time to return to equilibrium. This happens because the offset applied to the cantilever signal is such that it detects even the smallest noise due to both instrumentation and possible small oscillations.

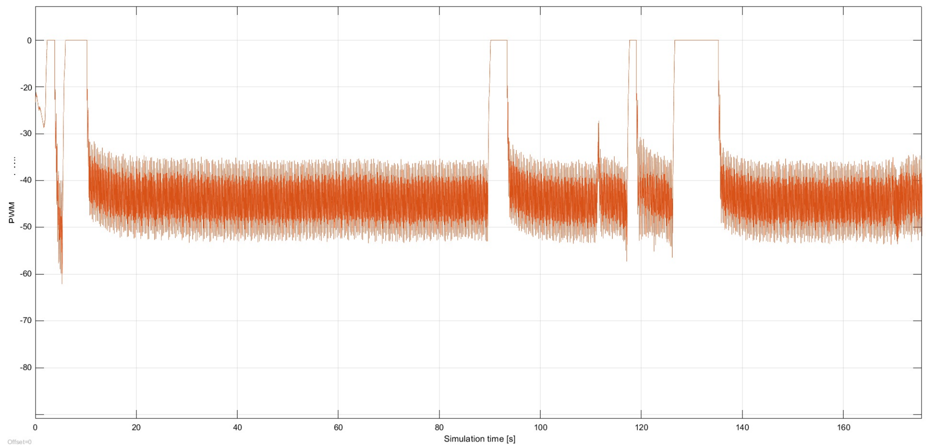

Figure 20: Maintaining resonance with disturbances.

4 Cantilever Model

As the last part of this study, we focused on modeling the system in the resonance regime. Figure 21 shows the equivalent diagram.

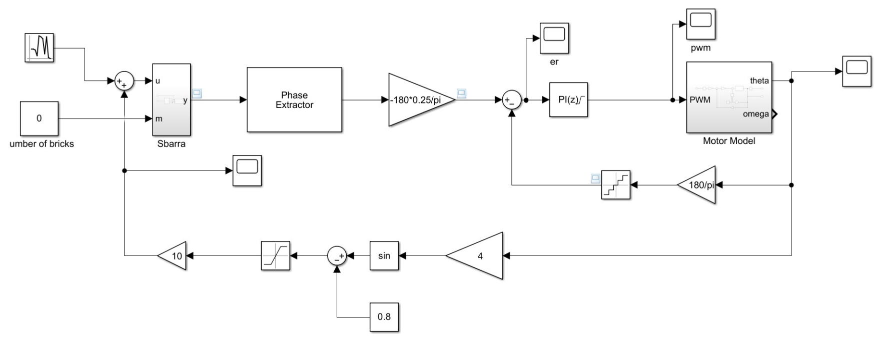

Figure 21: Resonant bar model.

The circuit's section from *Phase Extractor* towards right has been fully elucidated in the previous paragraphs, the only thing that changes in this case is the use of motor and encoder models. So we concentrate on what remains in figure 21. Inside *Sbarra* subsystem we find the rod's modeling as shown in figure 22. The swinging bar is a second-order system and we know its natural frequency because it's close to the one of the resonance. In Laplace's domain a system like this is stated by the following transfer function:

$$G(s) = \frac{1}{1 + 2\delta\frac{s}{\omega_n} + \frac{s^2}{\omega^2}} = \frac{\omega^2}{s^2 + 2\delta\omega_n s + \omega^2}$$

where ω_n is the natural frequency and δ damping ratio.

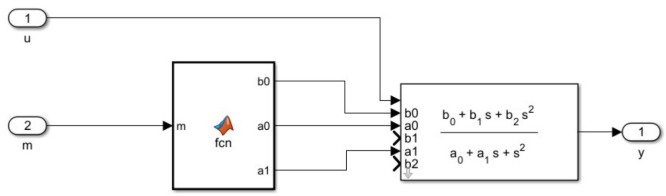

Figure 22: Bar swing model.

Thanks to the *Varying Transfer Function* block we could simulate the system in the four cases that we had applied previously. We used a Matlab Function to make this possible and by taking advantage of a switch-case logic we defined case by case the coefficients **a0**, **a1** and **b0** that mark out the transfer function (figure 23).

```
switch(m)
    case 0
        f=4.6;

        b0= (2*pi*4.6)^2;
        a1=2;
        a0=(2*pi*4.6)^2;

```

Figure 23: Calculation of the coefficients in the case of no mass applied to the rod.

We used the variable **m** to switch from a case to another and define how many bricks are attached on the Cantilever's tip. This variable is provided from the user through the *Constant* block which is located outside the examined subsystem. We had at our disposal the frequency values by the experimental tests, so it has been possible to calculate the natural frequencies and the coefficients, case by case. For all the case studies we supposed a damping ratio of $\delta = \frac{1}{\omega_n}$ or rather $a1 = 2$.

Another crucial aspect was the modeling of the system's excitation (figure 24), in other words, a little wheel, turned by the LEGO's motor, which has four rubbers assembled on it that hit the rod's hub every quarter of turn.

The idea was to simulate the contact between the bar and the tip of the rubber pad. The latter is not point-like but drags itself for a certain distance on the shaft, in which the exchange of forces takes place. It is therefore assumed that the magnitude of this force follows the trend of the engine phase, appropriately

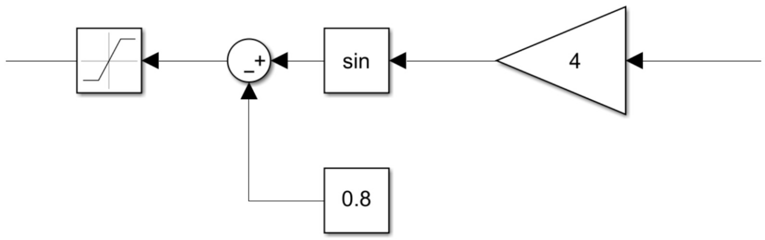

Figure 24: Excitation model.

brought back to the phase of the rod and extracted around the maximum signal. This justifies the value of the gain and the amount that is deported from the sine calculation. A saturation block has been inserted to eliminate the negative values coming out of the sin block, that is, to exclude forces opposite to those of excitation which in our case had no physical sense.

A gain was placed at the output of the saturation block so that the system would represent more faithfully the behavior of the real circuit. After various tests, an amplitude gain of 10 was chosen as an optimal value.

Finally, to make the system more realistic, but in particular to make it work, a noise was added to the excitation signal, using the *Random Number* block. This then is one of the two entrances to the *Sbarra* subsystem. Without the noise we would have no initial input and the circuit would behave like a quiescent rod.

Figure 25 below, shows the results simulated with Simulink in the case of three masses applied to the cantilever. The red signal is the one after the saturation block and it represents the modeled force applied to the hub's bar. Whereas the blue one is the the trend of the engine phase brought back to the phase of the rod.

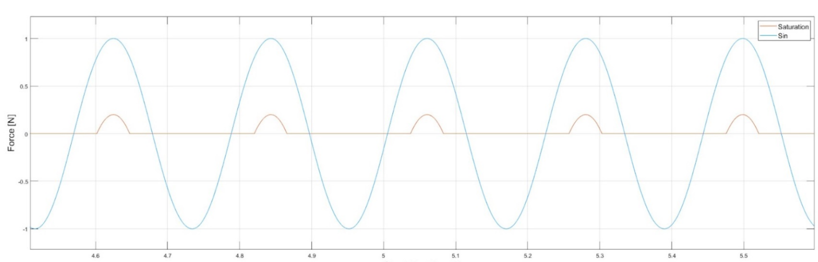

Figure 25: Extraction and excitation trend.

At the output of the *Sbarra* block we have obtained a signal that effectively

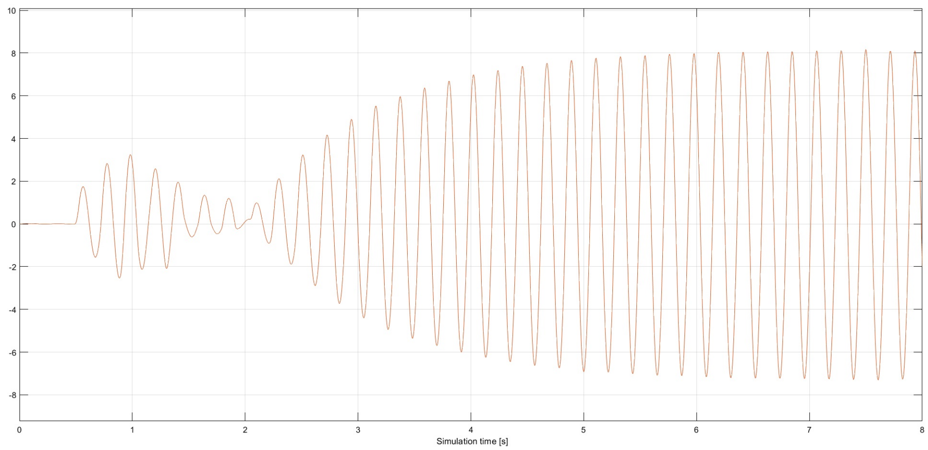

Figure 26: Modeled bar swing.

models the resonant frequency of the real system, as we see in figure 26. It is noted that after an initial transient where the bar is brought into motion, it then stabilizes and oscillates at its natural frequency.

The following signal, as described in the previous paragraphs, is sent to the input of the Phase extractor block which extracts the phase and is sent to the input of the negative feedback block. The values of the PI controller are the same as shown in table 6, with Tc = 0.005 seconds. Phase tracking, as seen in figure 27, is very similar to the real circuit (figure 15).

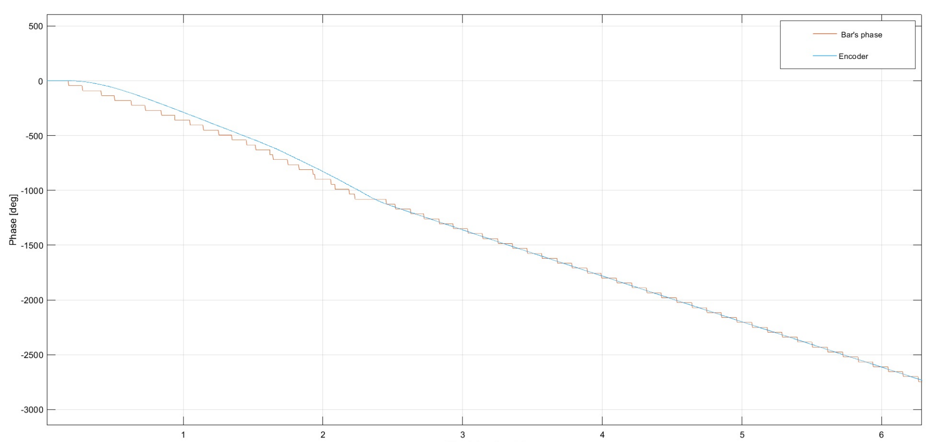

Figure 27: Model's phase tracking.

At the end we analyzed the tracking error and PWM's behavior that are inputs in *Motor Model*. We obtained values close to the ones of the real model. Indeed the error signal, after a transient beginning, started to swing around zero (figure 28). As far as the PWM is concerned, as shown in figure 29, we noticed that the value -45 as in the same as when we got in the previous experiment part. As we changed the mass we found similar results with the practical part.

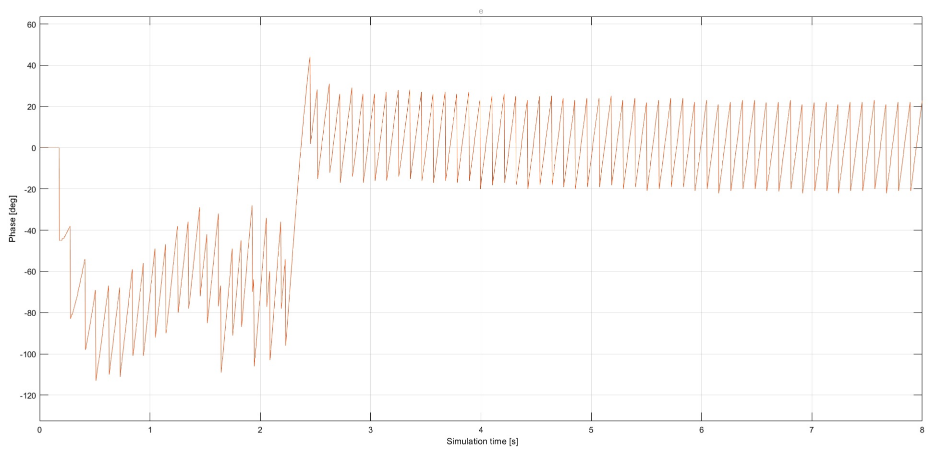

Figure 28: Model's error tracking.

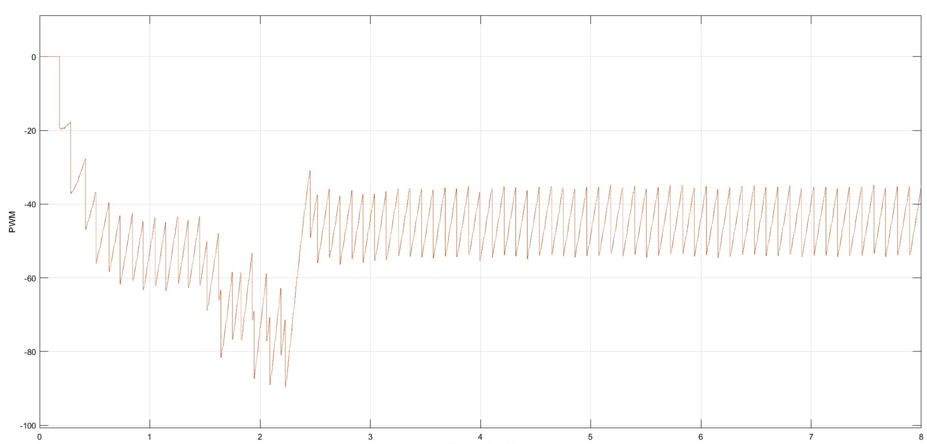

Figure 29: Model's PWM value.

5 Conclusions

In this experiment we have studied the behavior of the Cantilever: a moving rod constrained by one tip. In the first place we computed the bar's frequency oscillation by changing the number of LEGO bricks applied on it. We then observed how the Cantilever reacted with the mass variation which is a useful data for practical applications on a microscopic level such as the measurement of a body's weight or the density of a fluid.

We continued to study the subject matter wondering how to preserve the system in a resonance state or rather how to maintain a constant and maximum oscillation at the rod. We changed the circuit setup slightly and we found out that it's possible thanks to a negative feedback on the motor's phase. As desired setpoint for the control, we used the bar's phase or rather the swinging signal's phase caused by the turning of the motor. The operating method in this part was also a good reference for what we did in the experimental part, indeed we could compare the results to each other.

The turning point about making a suitable control was the using of the Simulink *Phase Extractor* block, which provided us to measure the signal of the swinging rod. However as future development we could enhance or even substitute the *Phase Extractor* block. Indeed, as shown in the previous graphs, this causes sawtoothed profiles and so the reading of mentioned graphs couldn't be clear. For which an initial idea could be the linearization of the output's block that could improve for instance model's error tracking.

6 Appendix

This section contains the code used in the Matlab Function blocks.

A - Computing syntax for frequency (chapter 2):

```
function [periodo,frequenza] = fcn(tempo,Count)

periodo=tempo/Count;
frequenza = 1/periodo;

end
```

B - Switching logic for resonance's model (chapter 4):

```
function [b0,a0,a1]= fcn(m)

b0=0;
a0=0;
a1=0;

switch(m)

    case 0

        f=4.6;
        b0=(2*pi*4.6)^2;
        a1=2;
        a0=(2*pi*4.6)^2;

    case 1

        f=4.3;
        b0=(2*pi*4.3)^2;
        a1=2;
        a0=(2*pi*4.3)^2;

    case 2

        f=4;
        b0=(2*pi*4)^2;
```

```
        a1=2;
        a0=(2*pi*4)^2;

    case 3

        f=3.7;
        b0=(2*pi*3.7)^2;
        a1=2;
        a0=(2*pi*3.7)^2;

end

if m>3
    b0=0;
    a1=0;
    a0=0;

if m<0
    b0=0;
    a1=0;
    a0=0;

end
end
end
```