Politecnico di Milano
Automation and Control Engineering

Automation and Control Laboratory

Control of Linear Vibrations

Alessio Russo, Gianluca Savaia, Alberto Ficicchia

Academic Year 2015/2016

Prof. Facchinetti A.
Dr. Costa A.
Dr. Giussani M.
Dr. Izardi M.

Contents

Chapter 1

Project description

The system is composed by:

1. A DC direct drive brushed motor
2. Three carts
3. Three springs
4. Several weights
5. Three encoders for the position
6. A PoliArd board

The mechanism consists of up to three mass carriages interconnected by bi-directional springs. The mass carriage suspension is an anti-friction ball bearing type with approximately ± 3 cm of available travel. The linear drive is comprised of a gear rack suspended on an anti-friction carriage and pinion coupled to the motor shaft. Optical encoders measure the mass carriage positions.

Three springs can be attached between the carts or between the first cart and the base plate and the mass of all the carts can be adjusted using the weights provided (500 ± 5 g each).

The encoder type is CP-850-4000-ECP: it is an optical incremental digital rotary shaft encoder. The number 4000 indicates the line counter (i.e. the number of equally spaced radial lines per 360 mechanical degrees on the incremental encoder code disk). Moreover has a resolution of 14 bits.
A low power light source is used to generate two 90 degrees out of phase sinusoidal signals on the detectors as the moving plate rotates with respect to the stationary plate. The moving plate rotates by means of an iron string wound up in it and then attached to the cart. The position is measured by calculating the angular displacement of the disk.The motor encoder is used to transmit both voltage and current to the Arduino board.

The 24 Volt DC motor has a case diameter of 63 mm and 194 Watts of output power. It has a nominal speed of 3700 rpm and a nominal Torque of 270 mN m. The DC motor has nominally a resistance of $2 \cdot 0.7\ \Omega$ and an inductance of $2 \cdot 1.05$ mH.

The poliArd board mounts an Arduino Due board. The board has a 24 V power supply and it's capable of measuring up to 5 A current. It has two motor drivers and four encoder interfaces. The board is programmable and the control strategy can be implemented using Matlab 2015a. To power on the board the user has to flip the first lever from the left. The other two levers can be programmed freely.

1.1 Issues

1.1.1 Motor saturation

From simple test it's easy to see that measured current saturates. What can be observed is that even though the current reaches a maximum value, the cart continues to move so this means that the sensor is not able to measure above a given current. In particular, it can be seen from Figures (specificare numero!!!) that the sensor saturates to 4.42 A and to -10 A.
In any case the Arduino Due cannot reliably measure more than 5 A current as the data-sheet specifies, so from this point onward we will limit all test currents to 5 A in both directions. Even in this way a difference remains between positive and negative measured current.

Table 1.1: My caption

$\mathbb{E}[i(t)], v < 0$	$\text{var}(i(t)),\ v < 0$	$\mathbb{E}[i(t)], v > 0$	$\text{var}(i(t)),\ v > 0$
-0.020	0.0015	-0.022	0.0014
-0.554	0.0035	0.683	0.0013
-1.758	0.0119	1.568	0.0015
-2.703	0.0274	2.312	0.0016
-3.964	0.0359	3.182	0.0016
-5.454	0.0523	4.316	0.0022

1.1.2 Current noise

After applying the protector to the system we measured the current noise at different current levels.

We applied a ladder type signal to measure the noise when the current was constant at different current levels. From the values obtained it can be seen that the variance of the negative current increases as the current increases in modulus instead if the current is positive the variance is small and almost constant. The fault of this behaviour for negative current is probably of the current measurement system.

From the variance values of the current for $v(t) < 0$ it can be seen that the variance grows linearly with the modulus of the current, whilst for $v(t) > 0$ is almost constant. Therefore for $v(t) < 0$ we decided to estimate the variance by simple linear interpolation:

$$\text{var}(i(t)) = a + b\mathbb{E}[i(t)] \tag{1.1}$$

by means of least squares interpolation:

$$\theta = (X^T X)^{-1} X^T Y, \quad Y = \begin{bmatrix} \text{var}(i_1) \\ \vdots \\ \text{var}(i_n) \end{bmatrix}, X = \begin{bmatrix} 1 & \mathbb{E}[i_1] \\ \vdots & \\ 1 & \mathbb{E}[i_n] \end{bmatrix}, \theta = \begin{bmatrix} a \\ b \end{bmatrix} \tag{1.2}$$

where i_j represents the j-eth segment of current data taken from the ladder experiment. The results can be seen in figure(number!!!).

All these experiment were done in order to estimate the variance of the current sensor and to have enough information to build the variance matrixes for the Kalman filter.

1.1.3 Encoder signal

Another problem is the one of the signal given by the encoders. This signal is a voltage impulse given to the Arduino. Unfortunately, the measure showed by Poliscope is not in centimetres so a conversion was in order. The idea was to give a reference signal in an open loop configuration. Once the transient is passed both Poliscope value and the real displacement of the cart were measured. The Tables (!!!numero) show the values of the various measurements. The final column shows the ratio between the Poliscope value and the displacement. This ratio has a mean value of about 560 and this is the value that has been used to convert all the measurements of the experiments.

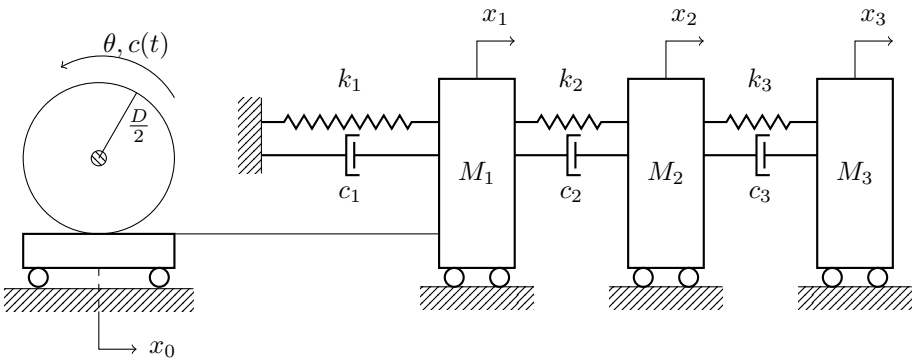
1.2 System protection

To avoid undesired behaviour of the system, a Simulink protection of the system has been designed. The system monitors current, voltage and displacement and if they overshoot a given value an alert is triggered that switches off the system to prevent any damage. Moreover, Arduino Due doesn't behave very well when the reference signal starts at the same time of its power on. For this reason, the system has been designed in order to start the signal only after 3 seconds the user pushed the lever number 2. This logic has been implemented using a finite state machine through the library Stateflow in Matlab. The third lever instead it's used to change from a reference signal to the potentiometer of the board. The system is composed by four macroblocks: - Input Manager - Protector - System - Controller Input manager Decides which input to use between potentiometers and signals. If the third switch of the Arduino board is enabled then the input switches from the signals to the manual potentiometers. One of the potentiometer is used to insert a disturbance in the control variable and the other one is used to give a variable and constant reference signal. Protector This part of the system is in turn composed by five protective blocks. One for the Voltage, one for the reference, three for the displacement. The voltage protector simply

saturates the voltage to avoid damage to the Arduino board. The position protectors give an alert message if the displacement of any cart is more than ±3cm. The reference protection saturates the reference to ±3 and a parallel circuit disables the input if the alert is active. Finally, there's a sixth block that contains two finite states machines: - Init Manager - Alert Manager The first one is simply used to delay the start of the signals by 3 seconds when the board is turned on. This is done because, as said above, if the signals and the board switch on at the same time then the board behaves badly. The second one processes all the signals and decides when to enable or disable the alert status. In particular, if the alert signal is on, the motor will be disabled and it won't be enabled again until the voltage goes under ±0.3 V. System Inside the system the reference signal is subtracted to the value measured by the position encoder and this error will be used to stabilize the system. Prior to the tests, either encoder 1 or 2 can be selected as feedback. The last block is the motor whose logic status is controlled by the motor disabler signal and whose continuous value is changed by the reference. Controller This block changes depending on which controller we want to use. When the motor disable signal is enabled there is a loop which rapidly pulls the voltage to 0 V in order to ready the system quickly. This exploit is used in order to avoid that the voltage inside the motor saturates for some reason and thus avoiding a total block of the system and the need of a manual intervention.

Chapter 2

Linear system modelling



2.1 Motor, Pinion and Rack modelling

A DC Brushless motor can be modelled with a simple low-pass filter transfer function. Because of that, only a resistance R and an inductance L are needed to model it. Thus, let D be the diameter of the disk attached to the motor, θ the the angular position of the disk, J the inertia of the motor, and $c_l(t)$ the load torque. We can also assume non-linearities based on the angle and its rate $\dot{\theta}$: we will denote such non-linearities with $f_m(\theta, \dot{\theta})$.

The output torque of the motor, then, is given by:

$$c(s) = K_e i(s) \tag{2.1}$$

and:

$$i(s) = \frac{1}{Ls + R}(v(s) - K_e s\theta(s)) \tag{2.2}$$

where $v(s)$ is the Laplace transform of $v(t)$, the input voltage to the motor. Notice that the term $K_e s\theta(s)$ describes the back-emf effect.
The differential equation describing the motion of the disk is:

$$J\ddot{\theta} = c(t) - c_l(t) - f_m(\theta, \dot{\theta})$$

In our case, since it's not possible to detach the pinion and the rack from the motor, J includes the inertia of that system, and $c_l(t)$ is the torque load attached to the rack.

The rack position is given by x, which is equal to 0 when the rack is at the center. Since the rack is attached to the disk, and by neglecting non-lineart effects such as back-lash, we can say that $x = \frac{D}{2}\theta$. Moreover, the total force transmitted from the motor is:

$$F(t) = \frac{2}{D}(c(t) - J\ddot{\theta}) = \gamma i(t) - \frac{4}{D^2}J\ddot{x}$$

Where $\gamma = \frac{2K_e}{D}$.

2.2 Carts, Springs and Dampers Modelling

Each cart has the same shape and mass M_c. Let M_i denote the total mass of the i-eth cart including the load, $i \in 1, 2, 3$.
Let x_i be the position of each cart, in cm. The small friction coefficient of the sliding guide can be approximated with a viscous friction $C_s(M)$ which depends on the mass of the cart plus the load.

Finally, each spring is modeled as linear spring. Since we have 3 springs, we have labeled their stiffness as K_l, K_m, K_h where l, m, h stand for low, medium and high. The damping contribute given by each spring is labeled as C_l, C_m, C_h. Therefore let C_i denote the total damping contribution for the i-eth cart.

Therefore, because of Newton's First Law, each cart has equation:

$$M_i\ddot{x}_i + C_i\dot{x}_i + K_i x_i = F(t)$$

Thus, for each cart 2 states are needed to describe its behaviour.

2.3 1 DOF State Space Model

Including the motor 3 states are needed to describe the behaviour of the system. Let x_1 be the state of the motor, x_2 the position of the cart and x_3 its velocity. The back-emf term in the motor is given by: $-K_e s\theta = -K_e \frac{2}{D}\dot{x} = -\gamma x_3$. Therefore the equations to consider are:

$$\begin{cases} \dot{x}_1 = -\frac{R}{L}x_1 + \frac{1}{L}v(t) - \frac{\gamma}{L}x_3 \\ M\dot{x}_3 = -Cx_3 - Kx_2 + F(t) \end{cases} \tag{2.3}$$

Since:

$$F(t) = \gamma i(t) - \frac{4}{D^2}J\ddot{x}$$

By letting $\hat{M} = (M + \frac{4}{D^2}J)$, we have:

$$\begin{cases} \dot{x} = \begin{bmatrix} -\frac{R}{L} & 0 & -\frac{\gamma}{L} \\ 0 & 0 & 1 \\ \frac{\gamma}{\hat{M}} & -\frac{K}{\hat{M}} & -\frac{C}{\hat{M}} \end{bmatrix} + \begin{bmatrix} \frac{1}{L} \\ 0 \\ 0 \end{bmatrix} v(t) \end{cases} \tag{2.4}$$

where $v(t)$ is the external control input.

2.4 2 DOF State Space Model

To derive the equations of motion for the carts we can use the Lagrangian approach.
Let T, V, D be the kinetic, potential and dissipated energy. Then:

$$\begin{aligned} T &= \frac{1}{2}\Big(M_1 + \frac{4}{D^2}J\Big)\dot{x_1}^2 + \frac{1}{2}M_2\dot{x_2}^2 \\ V &= \frac{1}{2}k_1x_1^2 + \frac{1}{2}k_2(x_2 - x_1)^2 \\ D &= \frac{1}{2}c_1\dot{x_1}^2 + \frac{1}{2}c_2(\dot{x_2} - \dot{x_1})^2 \end{aligned}$$

Let Q be the external forces acting on the systems:

$$Q = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \gamma i(t)$$

The equations of motion are given by:

$$\frac{d}{dt}\Big(\frac{\partial T}{\partial x_i}\Big) - \frac{\partial T}{\partial \dot{x}_i} + \frac{\partial V}{\partial x_i} + \frac{\partial D}{\partial \dot{x}_i} = Q_i$$

Thus we can write:

$$\begin{bmatrix} \hat{M}_1 & 0 \\ 0 & M_2 \end{bmatrix} \ddot{x} + \begin{bmatrix} c_1 + c_2 & -c_2 \\ -c_2 & c_2 \end{bmatrix} \dot{x} + \begin{bmatrix} k_1 + k_2 & -k_2 \\ -k_2 & k_2 \end{bmatrix} x = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \gamma i(t)$$

Then:

$$\ddot{x} = M^{-1}C\dot{x} + M^{-1}Kx + M^{-1}B\gamma i(t)$$

where:

$$M = \begin{bmatrix} \hat{M}_1 & 0 \\ 0 & M_2 \end{bmatrix}, C = \begin{bmatrix} c_1 + c_2 & -c_2 \\ -c_2 & c_2 \end{bmatrix}, K = \begin{bmatrix} k_1 + k_2 & -k_2 \\ -k_2 & k_2 \end{bmatrix}, B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

Finally, let x_1 be the state of the motor, x_2 and x_3 the position of the first and second cart, and x_4, x_5 their velocities, then the stace space model, is:

$$\begin{cases} \dot{x} = \left[\begin{array}{c|cc|cc} -\frac{R}{L} & 0 & 0 & -\frac{\gamma}{L} & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ \hline M^{-1}B\gamma & \multicolumn{2}{c|}{-M^{-1}K} & \multicolumn{2}{c}{-M^{-1}C} \end{array}\right] x + \begin{bmatrix} \frac{1}{L} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} v(t) \end{cases} \tag{2.5}$$

where $x \in \mathbb{R}^5$.

2.5 3 DOF State Space Model

As for 2 degree of freedom we can make use again of the Lagrangian Approach:

$$T = \frac{1}{2}(M_1 + \frac{4}{D^2}J)\dot{x}_1^2 + \frac{1}{2}M_2\dot{x}_2^2 + \frac{1}{2}M_3\dot{x}_3^2$$
$$V = \frac{1}{2}k_1x_1^2 + \frac{1}{2}k_2(x_2 - x_1)^2 + \frac{1}{2}k_3(x_3 - x_2)^2$$
$$D = \frac{1}{2}c_1\dot{x_1}^2 + \frac{1}{2}c_2(\dot{x_2} - \dot{x_1})^2 + \frac{1}{2}c_3(\dot{x}_3 - \dot{x}_2)^2$$

Let Q be the external forces acting on the systems:

$$Q = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \gamma i(t)$$

The equations of motion are given by:

$$\frac{d}{dt}\left(\frac{\partial T}{\partial x_i}\right) - \frac{\partial T}{\partial \dot{x}_i} + \frac{\partial V}{\partial x_i} + \frac{\partial D}{\partial \dot{x}_i} = Q_i$$

Thus we can write:

$$\begin{bmatrix} \hat{M}_1 & 0 & 0 \\ 0 & M_2 & 0 \\ 0 & 0 & M_3 \end{bmatrix} \ddot{x} + \begin{bmatrix} c_1 + c_2 & -c_2 & 0 \\ -c_2 & c_2 + c_3 & -c_3 \\ 0 & -c_3 & c_3 \end{bmatrix} \dot{x} + \begin{bmatrix} k_1 + k_2 & -k_2 & 0 \\ -k_2 & k_2 + k_3 & -k_3 \\ 0 & -k_3 & k_3 \end{bmatrix} x = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \gamma i(t)$$

Then:

$$\ddot{x} = M^{-1}C\dot{x} + M^{-1}Kx + M^{-1}B\gamma i(t)$$

where:

$$M = \begin{bmatrix} \hat{M}_1 & 0 & 0 \\ 0 & M_2 & 0 \\ 0 & 0 & M_3 \end{bmatrix}, C = \begin{bmatrix} c_1 + c_2 & -c_2 & 0 \\ -c_2 & c_2 + c_3 & -c_3 \\ 0 & -c_3 & c_3 \end{bmatrix}, K = \begin{bmatrix} k_1 + k_2 & -k_2 & 0 \\ -k_2 & k_2 + k_3 & -k_3 \\ 0 & -k_3 & k_3 \end{bmatrix}, B = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

Finally, let x_1 be the state of the motor, x_2, x_3, x_4 the position of the first, second cart and third cart, and x_5, x_6, x_7 their velocities, then the stace space model, is:

$$\begin{cases} \dot{x} = \left[\begin{array}{c|ccc|ccc} -\frac{R}{L} & 0 & 0 & 0 & -\frac{\gamma}{L} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ \hline M^{-1}B\gamma & \multicolumn{3}{c|}{-M^{-1}K} & \multicolumn{3}{c}{-M^{-1}C} \end{array}\right] x + \begin{bmatrix} \frac{1}{L} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} v(t) \end{cases} \tag{2.6}$$

where $x \in \mathbb{R}^7$.

Chapter 3

Identification introduction

The system considered can be easily modelled and identified without the need to use black-box identification to identify the system.
For completeness both *white-box* and *grey-box* identification are used.

First of all the problem of whether to consider a *closed* or *open* loop system is considered. In fact *back-emf* can be seen as a gain acting on the velocity of the cart, thus it's a gain on the closed loop.

Then, using both *white-box* and *grey-box* identification we identified the main parameters of the system:

1. Resistance and inductance for the motor.
2. Mass, stiffness and damping for the cart and the springs.

Last, identification of non-linearities are considered.

3.1 Validation cost function

An important aspect of the identification process is validation of results and this can be done in many ways.

We will mainly compare two signals, thus effectiveness in capturing the shape of a signal is essential for the type of validation function that we will use.

For this purpose we can make use of a distance function $d(x, y) : \mathbb{R}^n \times \mathbb{R}^n] \to [0, 1]$ induced by a generic norm $n(x) : \mathbb{R}^n \to [0, \infty)$. In this case we can construct d in the following way:

$$d(x, y) = \frac{1}{1 + n(x - y)}$$

The problem, then, is to find a norm capable of capturing the essential information of a signal.

Usually the L_2 norm is used, since it's related to the signal energy, and from a statistical point of view it corresponds to the variance of the difference of two signals. It's called *MSE-Mean Square Error*: an estimator of the overall deviations between prediction and measurements. Mathematically:

$$\text{MSE} = \mathbb{E}[(x - y)^2]$$

Where n is the dimension of x, y.

Why does it corresponds to the L_2 norm? First of all, notice that $\mathbb{E}[vw]$, where v, w are random variables, corresponds to a non-scaled projection of v on w. Any projection can be written in terms of a generic scalar product $< \cdot, \cdot >$, because of the Projection Theorem, thus:

$$\mathbb{E}[(v - w)^2] =< v - w, v - w >$$

The last term corresponds to the square of a norm $||v - w||^2$, which can be proven to be the L_2 norm:

$$\mathbb{E}[(v - w)^2] = \frac{1}{T} \int_0^T (v - w)^2 dt = \frac{1}{T} ||v - w||_2^2$$

But does the L_2 norm capture information about the shape of a signal? For signals with finite energy the L_2 norm in time corresponds to a L_2 norm in frequency, due to the Parseval's Theorem. Therefore minimizing the L_2 norm

implies the minimization of shape differences between the two signals.

Therefore, *MSE* as defined before, is the square of a norm. Thus:

$$n(z) = \sqrt{\mathbb{E}[z^2]}$$

is a norm. And the validation cost function is:

$$d(x,y) = \frac{1}{1+n(x-y)} = \frac{1}{1+\sqrt{\mathbb{E}[(x-y)^2]}}$$

.

3.2 Open vs Closed loop identification

In this experiment we had the necessity to choose whether to consider back-emf in the identification process or to completely ignore it.
As a matter of fact, ignoring it would mean to neglect a feedback component. But how much can it affect identification of other parameters?

Consider for example the following 2-nd order system, such as the system considered in the experiment:

$$G(s) = \frac{1}{Ms^2 + cs + k}$$

First consider a feedback loop with a constant gain ρ on the feedback. Thus the closed loop transfer function is:

$$T(s) = \frac{G(s)}{1+\rho G(s)} = \frac{1}{Ms^2 + cs + (k+\rho)}$$

The effect of ρ is to change the length of the poles, i.e. their absolute value, since for polynomial with real coefficients the zero-degree coefficient is the product of all roots.
Just compare with $s^2 + 2\xi\omega_0 s + \omega_0^2$, it's easy to see that $\omega_0^2 = \frac{k+\rho}{M}$.

In our case back-emf acts on the velocity of the cart, so if we have a feedback loop on the position, on the feedback we have γs, and the closed loop transfer function is:

$$T(s) = \frac{1}{Ms^2 + cs + k + \gamma s}$$

So what is the effect of γs? Again, if we compare with $s^2 + 2\xi\omega_0 s + \omega_0^2$ we have:

$$c + \gamma = 2\xi\omega_0$$

Where ξ has a strict relationship with the angle formed between the real negative axis and a pole, θ :

$$\theta = \arctan\Big(\frac{\sqrt{1-\xi^2}}{\xi}\Big)$$

So the effect of γs is to rotate the poles, but to which extent is this effect negligible?

From data we are mainly dealing with values of $\xi \in (0, 0.5)$, so we can approximate the value of θ:

$$\theta \approx \arctan\Big(\frac{1-\frac{\xi^2}{2}}{\xi}\Big) = \arctan\Big(\frac{1}{\xi} - \frac{\xi}{2}\Big) \approx \arctan\Big(\frac{1}{\xi}\Big)$$

Notice that in the last step we made use of the fact that $\frac{1}{\xi} \gg \frac{\xi}{2}$. Check figure 3.1 to compare the approximation.
Then, how much does θ change for a small variation of ξ?

$$\frac{d\theta}{d\xi} = -\frac{1}{1+\xi^2} = -1 + \frac{\xi^2}{1+\xi^2}$$

For $\xi < 0.5$ the change is almost linear, as seen from figure 3.1. Moreover $\frac{d\theta}{d\xi} \approx -1$ for $0 < \xi < 0.5$, so the slope of the curve is almost -1.

In our case $\xi = \frac{c+\gamma}{2\omega_0} = \frac{c}{2\omega_0} + \frac{\gamma}{2\omega_0}$, so the contribution of the backemf is $\frac{\gamma}{2\omega_0}$.

From the motor datasheet $\gamma \ll 1$ and from experiments ω_0 is always greater than $10\frac{rad}{sec}$, therefore the contribution is small, less than 1 and since the contribution to θ is linear with proportion ~ -1 also the change in θ is less than 1 degree, therefore backemf can be ignored and open-loop identification can be applied.

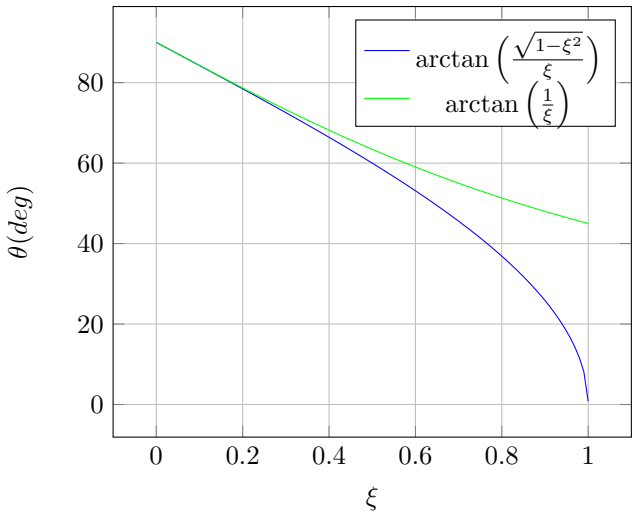


Figure 3.1: Approximated value of θ vs the real one

Chapter 4

White box identification

4.1 Detached system: cart and springs identification

To accurately identify the mass of the cart and the stiffness/-damping of the spring ,the motor was detached from the cart, in order to reduce influence of friction due to the pinion and rack.

So we obtain a system like the one considered in figure 4.1.

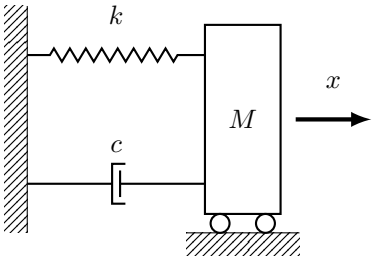


Figure 4.1: Cart detached from the motor diagram.

The differential equation governing this system is given by:

$$M\ddot{x} + c_i\dot{x} + k_i x = f(t)$$

where M [kg] is the total mass of the system , c_i [$\mathrm{N\,s\,m^{-1}}$] comprehends the damping of the i-eth spring and the viscous damping of the sliding guide. Finally k_i [$\mathrm{N\,m^{-1}}$] is the stiffness of the i-eth spring, and $f(t)$ represents external forces acting on the system (such as non-linear friction components).

4.1.1 Experiment description

For each spring we conducted 2 experiments, one without any load and one with a load of 0.986 [kg], each repeated 3 times. To accurately identify the mass of the cart and the stiffness/damping of the spring ,the motor was detached from the cart, in order to reduce influence of friction due to the pinion and rack.

For each experiment the cart was released from an initial condition $x(0) = x_0 \neq 0$ and 0 velocity, such that

the force that the spring was exerting on the cart was sufficient enough to make negligible the very small component of the static friction acting on the cart.
Notice that the initial condition differs for each spring since the stiffness is very different for each spring.

If we neglect the external forces acting on the cart, which are negligible since they are small non-linear components, then the system considered is:

$$\begin{cases} M\ddot{x} + c_i\dot{x} + k_i x = 0 \\ x(0) \in [1,3]\text{cm} \\ \dot{x}(0) = 0 \end{cases} \tag{4.1}$$

Then data regarding the position of the cart is collected, and from that data the pulsation, damping ratio, mass and stiffness are retrieved.

4.1.2 Experiment analysis

Using 4.1 the response in time can be obtained by using the Laplace transform. Let $X(s)$ be the Laplace transform of $x(t)$, then:

$$mX(s)(s^2 - x(0)s) + cX(s)(s - x(0)) + kX(s) = 0$$

and:

$$X(s) = x(0)\frac{(ms + c)}{ms^2 + cs + k}$$

If we solve in $X(s)$ and then apply the inverse Laplace transform, we obtain the response in time:

$$x(t) = e^{-\xi\omega_0 t}(A\cos(\omega t) + B\sin(\omega t))$$

where $\xi = \frac{c}{2\sqrt{Mk}}, \omega_0 = \sqrt{\frac{k}{M}}, \omega = \omega_0\sqrt{1-\xi^2}$, and A, B depend on $x(0), \xi$.

Since the pulsation is the same for both sinusoidal components we have:

$$x(t) = Ce^{-\xi\omega_0 t}\sin(\omega t + \phi)$$

Where $C = \sqrt{A^2 + B^2}, \phi = \arctan(A/B)$.

Knowing those equations we are able to extract data from the response in the following way:

- To measure ω we can just extract the period T: the difference in time between the first and second peak is taken, and that difference is the period. Then ω is just $\frac{2\pi}{T}$. We consider only the first and second peak because at the beginning non-linearities such as static and coloumb friction are negligible.
- To measure ξ also the first and second peak are considered. Let t_0, t_1 be the times at which there is the first and second peak. Notice that $t_0 = 0, t_1 = T$, and $x(T) = Ae^{-\xi\omega_0 T}$.
  Then, consider:
  $$\log\left(\frac{x(0)}{x(T)}\right) = \log(e^{\xi\omega_0 T}) = \xi\omega_0 T = \frac{\xi}{\sqrt{1-\xi^2}}2\pi$$
  Then
  $$\xi = \frac{a}{\sqrt{a^2+1}}, \quad a = \frac{1}{2\pi}\log\left(\frac{x(0)}{x(T)}\right)$$
  Once M, k are known we can calculate the damping from $c = 2\xi\sqrt{Mk}$. Observe that for $a \sim 0 \Rightarrow \xi \sim a$.
  Since damping
- To identify each spring and the mass of the cart we made use of the fact that the we have two type of experiments for each spring: one without any load, and one with a load of 0.986 kg. We obtain a system of linear equations:
  $$\begin{cases} \frac{k_i}{m_c + m_l} = \omega_l^2 \\ \frac{k_i}{m_c} = \omega_{nl}^2 \end{cases}$$
  Where m_c is the mass of the cart, m_l the mass of the load, ω_l the pulsation of the system with the load, ω_{nl} the pulsation of the system without the load. It's a system with two unknowns (k_i, m_c) and two equations, so we can solve it. We can rewrite it in matrix form:
  $$\begin{bmatrix} 1 & -\omega_l^2 \\ 1 & -\omega_{nl}^2 \end{bmatrix}\begin{bmatrix} k_i \\ m_c \end{bmatrix} = \begin{bmatrix} w_l^2 m_l \\ 0 \end{bmatrix}$$
  and solve for (k_i, m_c).

4.1.3 Experiment results

Since there are 3 springs let's denote the set of springs as $K = \{k_l, k_m, k_h\}$ where l stands for low, m for medium and h for high. In a similar manner we define the various pulsation: for example ω_{m-nl} is the pulsation for the system with spring k_m and no load.

Pulsation In the table below are shown the various mean of the pulsation and their relative standard deviation:

(ω_{avg} [rad s^{-1}],ω_{std} [rad s^{-1}])	k_h	k_m	k_l
with load	(21.2989, 0)	(14.2800, 0.0671)	(10.6495, 0)
with no load	(34.9066, 0)	(23.7101, 0.1792)	(17.6991, 0.1005)

Table 4.1: Pulsation of the cart detached from the motor. Various configuration are shown (with a load of 0.986 [kg] and no load) for the various springs.

It's interesting to note that even if we considered to average all the periods by considering the various peaks of the signal, and not only the first two peaks, we would have obtained the same results. This is an hint of the fact that the principal non-linearity, i.e. coloumb friction, is negligible.

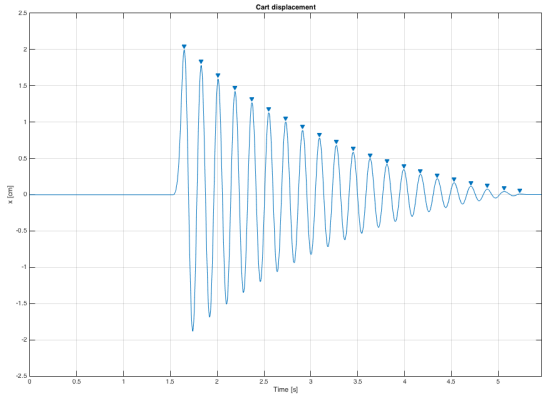


Figure 4.2: Displacement of the cart with spring k_h and load 0.986 [kg].

Cart mass and springs stiffness By using mean pulsation the resultant average mass of the cart m_c is 0.5685 [kg] with standard deviation 0.0141 [kg]. Results also for the springs are shown in table **??**.

(k_h [N m^{-1}], m_c [kg])	(k_m [N m^{-1}], m_c [kg])	(k_l [N m^{-1}], m_c [kg])
(712.5990, 0.5848)	(315.5074, 0.5612)	(175.2819, 0.5595)

Table 4.2: Identified springs and cart mass

Damping and damping ratio The mean values for the damping ratio, including their standard deviation, are shown in table **??** for the various springs, with and without a load.

(ξ_{avg},ξ_{std})	k_h	k_m	k_l
with load	(0.0128, 0.0007)	(0.0238, 0.0018)	(0.0346, 0.0036)
with no load	(0.0179, 0.0025)	(0.0301, 0.0013)	(0.0379, 0.0040)

Table 4.3: Damping ratio. Various configuration are shown (with a load of 0.986 [kg] and no load) for the various springs.

From the values shown in table **??** it seems that the damping C is function of the mass, in fact we don't obtain the same dampingif we consider the damping ratio with no load or with load. For example consider k_h: with a

load we obtain $C = 0.0128 \cdot 2 \cdot \sqrt{k_h M} = 0.8520$ [N s m^{-1}], without load: $C = 0.0179 \cdot 2 \cdot \sqrt{k_h m_c} = 0.7206$ [N s m^{-1}]. This is most likely an effect due to friction, and the various damping values are shown in table **??**.

C [N s m^{-1}]	k_h	k_m	k_l
with load	0.8520	1.0542	1.1423
with no load	0.7206	0.8063	0.7567

Table 4.4: Damping values. Various configuration are shown (with a load of 0.986 [kg] and no load) for the various springs.

We can therefore linearly characterize the damping value as function of the mass centered in m_c, for each spring:

$$C(m) = C_{nl} + \frac{C_l - C_{nl}}{m_l}(m - m_c) = C_{nl} + \alpha(m - m_c)$$

The different values of α, the difference quotient, are shown in table **??**

	k_h	k_m	k_l
$\frac{C_l - C_{nl}}{m_l}$ [N s m^{-1} kg^{-1}]	0.1334	0.2514	0.3911

Table 4.5: Damping difference quotient. Due to friction damping changes for different weights, we can therefore characterize the damping in a linear way with the formula: $C(m) = C_{nl} + \frac{C_l - C_{nl}}{m_l}(m - m_c) = C_{nl} + \alpha(m - m_c)$. Values of the difference quotient are shown for the different springs.

4.1.4 Validation

Validation was done in a similar fashion as the experiments conducted to identify the system parameters. The cart was released from a random initial condition x_0 and then released.

Measurements were compared with the output smulation of a model constructed from the identified parameters, and results were compared with the cost function defined in 3.1.

On average, the cost function, on a scale from 0 to 1, gave a fit of 0.9449 with standard deviation 0.0263. The minimum was 0.8778, and maximum 0.9716.

In figure 4.7 it is visible that due to unmodelld effects, such as friction and stiffness not being exactly linear, there is a loss of accuracy on the long run, but accuracy is very high in the beginning, where non-linearities are not relevant yet. Thus it is safe to assume that the parameters of the cart are the ones identified by this experiment.

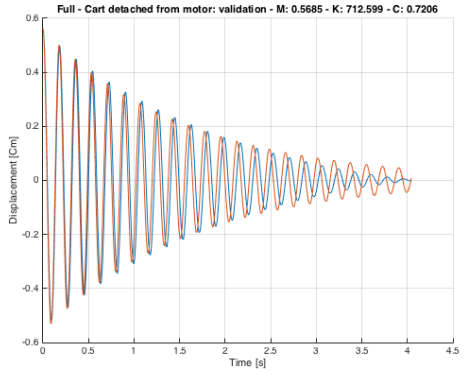


Figure 4.3: Validation test without load and K_h.

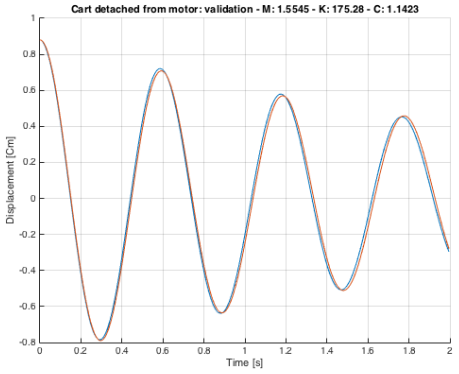


Figure 4.4: Validation test without load and K_l.

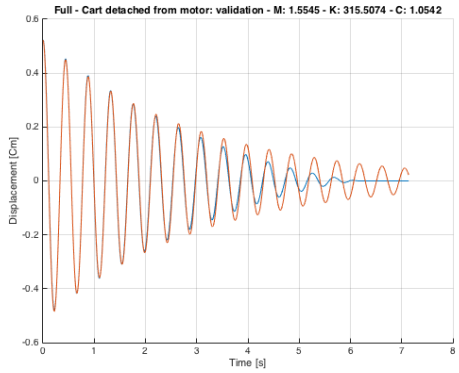


Figure 4.5: Validation test with load and K_m.

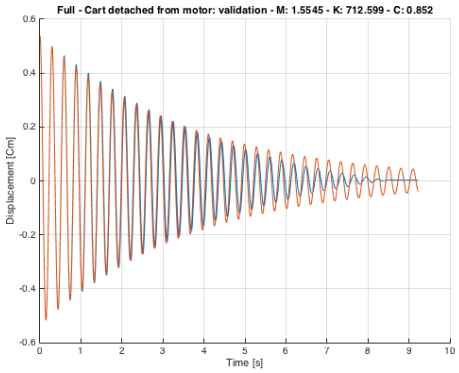


Figure 4.6: Validation test with load and K_h.

Figure 4.7: Validation tests of the cart detached from the motor. In orange the simulated output, in blue the measurement of the validation test. Effect of friction is clearly visible, such as in 4.5, where the cart stops after 6 seconds.

4.2 Motor identification

The motor can be modelled as a first order low pass filter, and the main parameters are:

- Resistance of the motor, R [Ω].
- Indutance of the motor, L [H].
- Torque constant, K_e [$\mathrm{N\,m\,A^{-1}}$].

From motor specifications the nominal values, which are identified with n as subscript, are:

$$R_n = 1.4[\Omega], L_n = 0.0021[H], K_{en} = 0.118[\mathrm{N\,m\,A^{-1}}]$$

The nominal cut-off frequency of the motor is $f_n = \frac{R_n}{L_n} = 106.10$ [Hz], which is slightly above the Nyquist frequency of the system 100 [Hz], therefore accurate identification by means of grey or black box identification may give a bias on the identification of the cut-off frequency.
Since resistance of the motor can be identified by steady state value of the current, only inductance may have a biased value.

4.2.1 Experiment description

For each spring K_i, two experiments were done, with the motor attached to the cart, one with a load of 0.986 kg, and one without any load (just the cart itself).

Since we can measure the current and because of our assumption that back-emf can be ignored (hence the system is

open-loop), we can directly identify the motor from the input voltage and the output current. Moreover, back-emf doesn't influence the steady state value of the current (since it acts on the velocity of the cart, which is 0 at steady state), therefore resistance of the motor can be accurately identified. Regarding the inductance, we can not accurately identify it, since because of our sampling time the motor can almost considered like a gain system. Therefore rising time and the use of estimation techniques will be used.

The input voltage is a square wave with period 10:

$$v(t) = \begin{cases} a, & t \in [t_0, t_0 + 5] \\ -a, & t \in [t_0 + 5, t_0 + 10] \end{cases}$$

where t_0 is the beginning time of the square wave, and $a = 3$ for K_h, K_m and $a = 2$ for K_l. In fact the current is proportional to the output torque of the motor, which ultimately acts on the cart which is attached with a spring to a wall, therefore less voltage is needed for springs with low stiffness to move the cart.

The system to be considered is:

$$L\dot{i}(t) + Ri(t) = v(t)$$

Where $i(t)$ is the current measured from the motor.

Finally, each experiment was run for a period of time of about ≈ 40 [s].

4.2.2 Experiment analysis

The system considered is a stable system, therefore at steady state $\dot{i}(t) = 0$ and $R = \frac{v(t)}{i(t)}$.

Because of noise sensor the mean value was taken as steady state value, after the transient due to back-emf. Since we know the input voltage we can then calculate R.

Regarding the inductance, since the motor can be modelled as a low pass filter of the first order, its response in time has the form $i(t) = v(1 - e^{-\tau t})$, where $\tau = \frac{R}{L}$. When $t = \frac{3}{\tau}$, which is about our sampling time since $\frac{3}{\tau_n} = \frac{3}{2\pi f_n} \approx 0.0045$ [rad s^{-1}],we have $i(\frac{3}{\tau}) \approx 0.95v$. Therefore we can check if after one time step the value is approximately 0.95 times the value of the input voltage.

Ultimately we can use the function *tfest* in matlab to identify a system of the first order given the input and output data.

4.2.3 Experiment results

From experiments steady state value of the current seems to change a little, based on the fact whether v is positive or negative. Thus let R_1 represent mean value of resistance when v is positive, and R_2 when v is negative.

From data mean values are: $R_1 = 1.3069$ [Ω], $R_2 = 1.2330$[Ω], with standard deviation $\sigma_1 = 0.0984\Omega, \sigma_2 = 0.0460\Omega$.

R then can be computed using a weighted average:

$$R = \frac{\sigma_2^2 R_1 + \sigma_1^2 R_2}{\sigma_1^2 + \sigma_2^2} = 1.2462\Omega$$

And standard deviation:

$$R_{std} = \sqrt{\frac{\sigma_1^2 + \sigma_2^2}{2}} = 0.0059\Omega$$

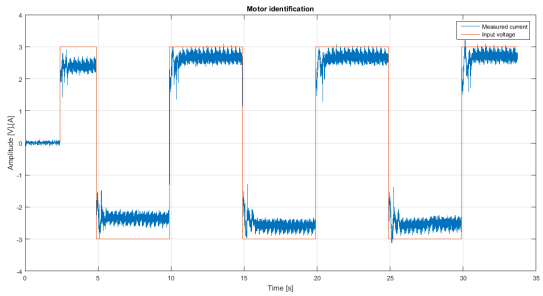
Using the *tfest* command the average and standard deviation values for R, L are:

$$R = 1.2689\Omega, R_{std} = 0.0562\Omega$$

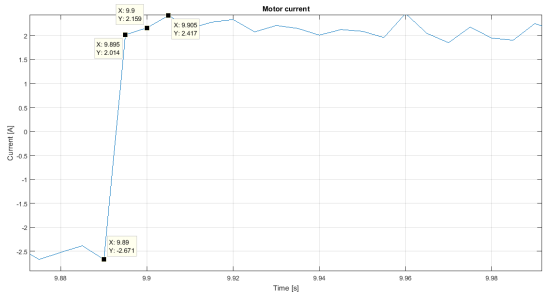
$$L = 0.0024\text{H}, L_{std} = 0.0002\text{H}$$

Using again a weighted average the mean value of R is $R = 1.2464\Omega$ with standard deviation 0.04Ω.

Regarding the inductance, the estimated value fits the nominal value. This can also be seen from figure 4.8 that steady state is approached in about 2 steps of the sampling time.



(a) Input voltage and output current of the motor



(b) Rising time of the current, it can be seen that steady state is approached very quickly.

Figure 4.8

4.2.4 Validation

Validation was done using a random input with normal distribution $N(0, \frac{9}{4})$ (Check figure 4.9). Notice that we couldn't have used that input as source for identification because we already know a model of the system, and so we are not interested in black box modeling.

On average, using the cost function defined in 3.1, with a first order system of the type:

$$G(s) = \frac{1}{Ls + R}$$

where $R = 1.2689\Omega, L = 0.0024H$, there is an average fit value $d(i_{real}, i_{sim}) = 0.8142$ with standard deviation 0.0364. In case black box identification is used, with a 3rd order system we would obtain an average fit value of

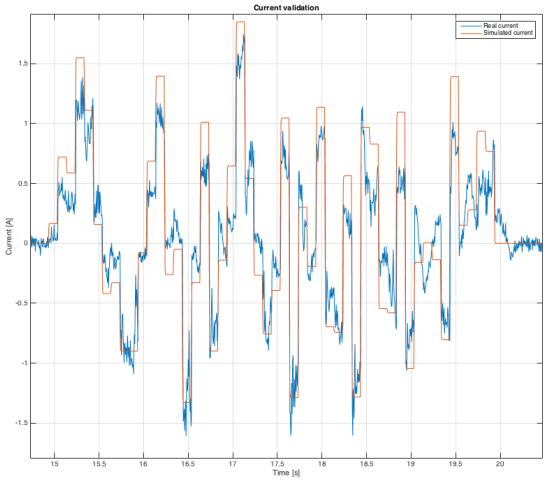


Figure 4.9: Validation of the motor with a random input signal $v(t) \sim N(0, \frac{9}{4})$.

0.8527 with standard deviation 0.0424.

4.3 Overall system identification

The system was modelled as an open loop system as specified in section 3.1. We have mainly a series connection of two systems: the motor and the cart, which are identified by the following transfer functions:

$$G_1(s) = \frac{1}{Ls + R} \quad G_2(s) = \frac{\gamma}{Ms^2 + Cs + K}$$

Since we are using the same tests used to identify the motor (section 4.2), and the same techniques used to identify the cart detached from the motor (section 4.1), those two sections are not presented.

4.3.1 Experiment results

Results are summarized in a similar fashion to section 4.1:

Pulsation In the table below are shown the various mean of the pulsation of the cart and their relative standard deviation:

(ω_{avg} [rad s^{-1}],ω_{std} [rad s^{-1}])	k_h	k_m	k_l
with load	(20.2100, 0.0022)	(13.0633, 0.3208)	(10.7500, 0.0308)
with no load	(30.4300, 0.2895)	(18.3767, 0.2011)	(15.6240, 0.1820)

Table 4.6: Pulsation of the cart attached to the motor. Various configuration are shown (with a load of 0.986 [kg] and no load) for the various springs.

System gain, mass and stiffness By using mean pulsation the resultant average mass of the system, including the cart, is 0.8906 [kg] with standard deviation 0.1146 [kg]. Results also for the stiffness are shown in table **??**. Therefore it results that the mass contribution of the motor is:

(k_h [N m^{-1}], m [kg])	(k_m [N m^{-1}], m [kg])	(k_l [N m^{-1}], m [kg])
(720.56, 0.778)	(340.14, 1.007)	(216.38, 0.8864)

Table 4.7: Identified stiffness and mass of the overall system

$$M_{motor} = 0.8906 - 0.5685 = 0.3221\text{Kg}$$

.

The identified gain of the motor, based on the fact that at steady state we have:

$$x(\infty) = \frac{\gamma}{k} i(\infty)$$

gives an average value of $\gamma = -2.0680$ N A^{-1} and standard deviation $\gamma_{std} = 0.2935$ N A^{-1}. Notice that $x(\infty), i(\infty)$ refers to the corresponding value at steady state.

Damping and damping ratio The mean values for the damping ratio, including their standard deviation, are shown in table **??** for the various springs, with and without a load.

(ξ_{avg},ξ_{std})	k_h	k_m	k_l
with load	(0.1356, 0.00022)	(0.1949, 0.0080)	(0.2230, 0.0042)
with no load	(0.1670, 0.0084)	(0.2569, 0.0124)	(0.2973, 0.0376)

Table 4.8: Damping ratio. Various configuration are shown (with a load of 0.986 [kg] and no load) for the various springs.

From the values shown in table **??** it seems that the damping C is function of the mass, like in **??**. The various damping values are shown in table **??**.

C [N s m^{-1}]	k_h	k_m	k_l
with load	10.2855	9.5558	8.9973
with no load	9.0517	8.4089	8.2736

Table 4.9: Damping values. Various configuration are shown (with a load of 0.986 [kg] and no load) for the various springs.

We can therefore linearly characterize the damping value as function of the mass centered in m_c, for each spring:

$$C(m) = C_{nl} + \frac{C_l - C_{nl}}{m_l}(m - m_c) = C_{nl} + \alpha(m - m_c)$$

The different values of α, the difference quotient, are shown in table **??**

	k_h	k_m	k_l
$\frac{C_l - C_{nl}}{m_l}$ [N s m^{-1} kg^{-1}]	1.2514	1.1631	0.7340

Table 4.10: Damping difference quotient. Due to friction damping changes for different weights, we can therefore characterize the damping in a linear way with the formula: $C(m) = C_{nl} + \frac{C_l - C_{nl}}{m_l}(m - m_c) = C_{nl} + \alpha(m - m_c)$. Values of the difference quotient are shown for the different springs.

4.3.2 Validation

For the validation procedure we used the same input voltage used in the validation of the motor, section 4.2.4. $v(t) \sim \alpha N(0, \frac{9}{4})$, where $\alpha \in (0.7, 1.5)$ based on which spring was attached to the first cart.

Validation 1 DOF

For each spring we made two tests, one without load and one with a load of about 1kg, for a total of 6 tests.
Validation using only one cart can be summarised in an average fit value of 85.42 %, with standard deviation of 1.68 %. Because of such high value, there is no need to consider more details in the models such as back-emf or nonlinearities.
In figure 4.14 some validation plots are shown.

Validation 2 DOF

Validation for 2 degree of freedom used the same random distribution of the validation input for 1 degree of freedom. We tested the combination K_h, K_m and K_l, K_h. For each combination we made 3 tests, one without any load on the two carts, one with half kilogram on each cart, and one with 1 kilogram only on the second cart.
Since now we have 2 outputs, the position of the first and second cart, we need to average the results for the first and second cart.

For the first cart we have an average fit of 87.29 % with standard deviation 4.5%. For the second cart the average fit is 85.79%, with standard deviation 4.8 %.
This results in a mean of 86.54 %, and standard deviation 4.68 %. Some validation plots are shown in figure 4.19 for the second cart.

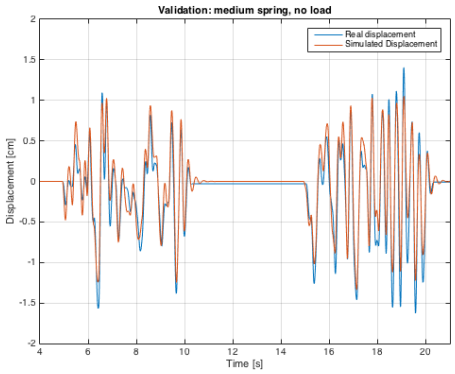


Figure 4.10: Validation test without load and K_m.

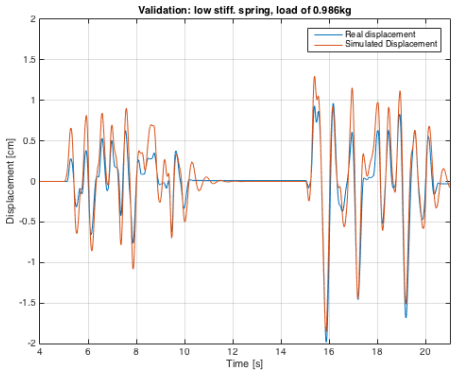


Figure 4.11: Validation test with load and K_l.

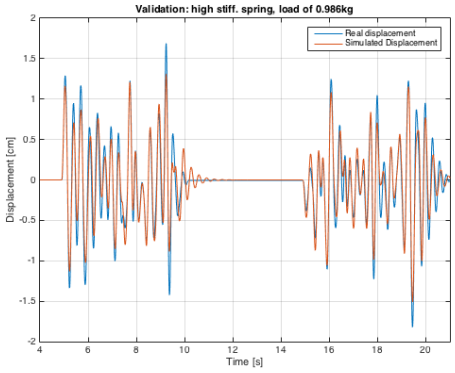


Figure 4.12: Validation test with load and K_h.

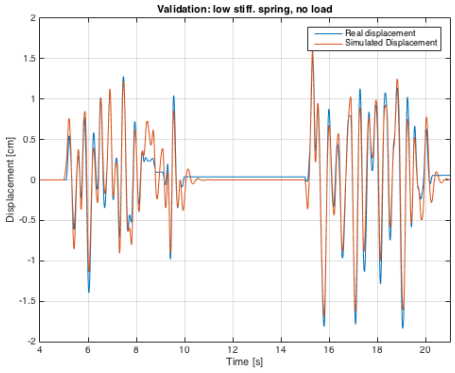


Figure 4.13: Validation test with no load and K_l.

Figure 4.14: Validation tests of the cart attached to the motor. In orange the simulated output, in blue the measurement of the validation test.

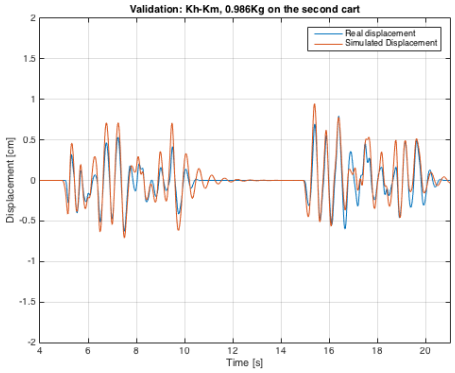


Figure 4.15: Validation test without load on the first cart, and a load of 0.986kg on the second one. The configuration K_h, K_m was used.

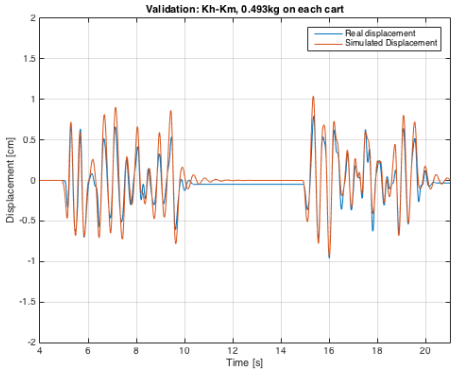


Figure 4.16: Validation test a load of about 0.493 kg on both the carts. The configuration K_h, K_m was used.

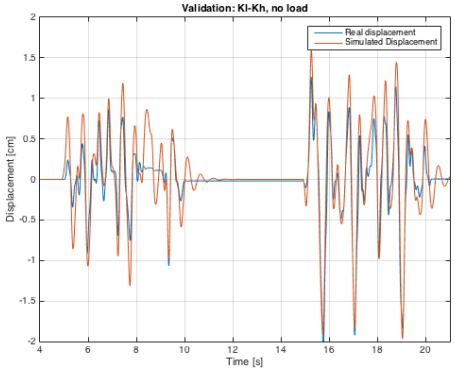


Figure 4.17: Validation test with no load on both the carts. The configuration K_l, K_h was used.

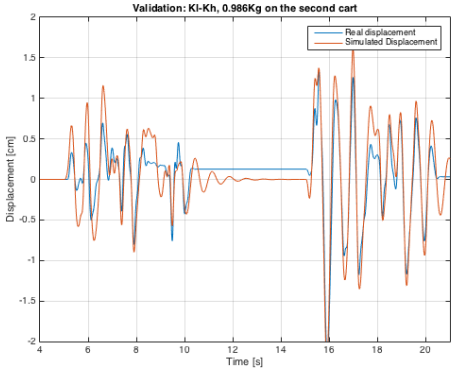


Figure 4.18: Validation test without load on the first cart, and a load of 0.986kg on the second one. The configuration K_l, K_h was used.

Figure 4.19: Validation tests of 2 degree of freedom. Only the output for the second cart is shown. In orange the simulated output, in blue the measurement of the validation test.

Chapter 5

Gray box identification

This approach consists in defining the simplest model that can explain the real system and fit your model to experimental data collected in open loop. Therefore, we neglect every nonlinearity of the system and consider only two basic transfer functions: the motor

$$I(s) = \frac{1}{Ls + R} V(s) \tag{5.1}$$

and the cart

$$X(s) = \frac{1}{Ms^2 + Cs + K} F(s) \tag{5.2}$$

connected by a constant γ which converts the measured current into a linear force exerced onto the cart for each time instant

$$\gamma = \frac{2K_e}{D} = \frac{f(t)}{i(t)} \tag{5.3}$$

The overall scheme can be drawn as follows:

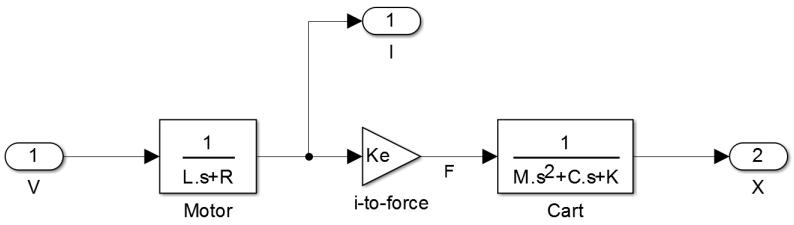


Figure 5.1: Scheme of the model employed in the identification process. The only input is the voltage applied to the motor and the accessible outputs are the measured current and the position of the cart.

In this way we are lumping all the uncertainties into 6 parameters: L, R, γ, M, C, K. This means that C (for example) does not represent the damping coefficient of the spring only, but it includes all the linearised friction which is happening in the plant; the same holds for the others. Our sole interest is to derive a model which represents as close as possible the system from the perspective of the controller rather than identifying as precisely as possible the single parameters.

The experiment setup is very straightforward: we feed a train of pulses into the system and measure both the current and displacement of the cart; this is repeated for each spring. The pulses must be long enough to allow the cart to reach the equilibrium. Once offline, the current is filtered in order to remove the high frequency noise added by the sensor and the experiment is cut so to find the pulse window which is less noisy; this piece of data is used in the following to perform the identification.

The identification is performed by an optimization method. We define the measured current as $i(t)$ while the predicted one as $\hat{i}(t) = \frac{1}{Ls+R}$; the objective of the optimization is

$$\min_{L,R} \|i - \hat{i}\|_2^2(t) \tag{5.4}$$

which is well known in the literature as *nonlinear least squares*. Since the objective function is trivially nonlinear, we cannot guarantee the uniqueness of the minima and we need to start the optimization several times from a different set of initial condition in order to validate the results.

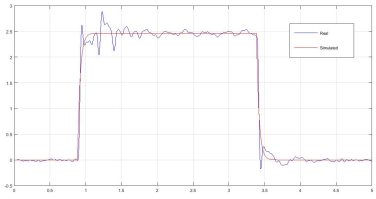


Figure 5.2: Fitting of the model of the motor.

This process yields the following results:

R	L
1.3 Ω	0.0220 H

The same applies for the identification of the cart, whilst this time we make use of the current as input and the displacement as output. Following the previous notation, the problem becomes

$$\min_{\gamma,M,C,K} \|x - \hat{x}\|_2^2(t) \tag{5.5}$$

where

$$\hat{x}(t) = \frac{\gamma}{Ms^2 + Cs + K} i(t) \tag{5.6}$$

We can actually use either $i(t)$ or $\hat{i}(t)$ in order to have a less noisy signal as input, but the output is not very different in each case. In this scenario, the choice of the starting point of the optimization is more sensitive and needs more start than the previous and the output parameters are more variable, although the poles of the model transfer function are always the same.

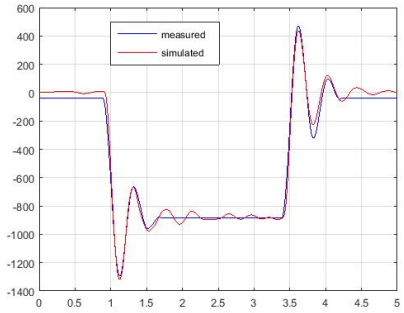


Figure 5.3: Fitting of the model of the cart.

The identified parameters in this stage are listed below.

γ	M	K_h	K_m	K_l	C_h	C_m	C_l
$-178.57\ \frac{N}{A}$	$1.3\ Kg$	$625\ \frac{N}{m}$	$281\ \frac{N}{m}$	$162\ \frac{N}{m}$	$9\ \frac{Ns}{m}$	$6\ \frac{Ns}{m}$	$8\ \frac{Ns}{m}$

Chapter 6

Non-linearities identification

6.1 Non-linear gain

As previously mentioned the motor has shown its non-linear behaviour in some occasions, especially for the control of 2-DOF.
During the initial identification stage it was observed that the gain of the motor, i.e.: $\frac{1}{R}$ was input-varying, therefore non-linear. This was not seem as a problem, since it is a type of non-linearity whose sector is $[\alpha, \beta]$ with $\alpha > 0, \beta < M < \infty, M \in \mathbb{R}$. Therefore with a high-gain feedback loop, simply with an integrator, it's possible to linearise this effect.

Obviously, since the poles of the motor is not changing it means that either $\frac{R}{L}$ is a constant, with R, L varying at the same velocity, which seems improbable, or that there is a non-linear gain that can be modelled as:

$$i(t) = \frac{1}{Ls + R} f(v)v = \frac{1}{Ls + R} g(v) \tag{6.1}$$

Where $M(s)$ is the standard linear form of the motor, as previously presented, and $g : \mathbb{R} \to \mathbb{R}$, $\frac{dg}{dv} \in [\alpha, \beta]$. Moreover, it seems to be a function with different behaviour for $v > 0, v < 0$.

Because of this, an identification process for both v negative and positive was conducted.

Let the ideal current given by the following expression, where $R = 1.3\Omega, L = 0.0024H$.

$$i_{id} = \frac{1}{Ls + R} v(t) \tag{6.2}$$

Then the real current is:

$$i(t) = i_{id} f(v) \tag{6.3}$$

Next we modelled $f(v)$ as a cubic function:

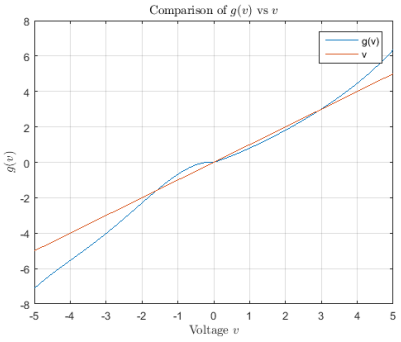
$$f(v) = \alpha + \beta v + \gamma v^2 + \delta v^3 \tag{6.4}$$

And used a least squares procedure to identify (α, β, γ). As a reference signal a step generator signal $r(t)$ was used with increasing voltage and period 5 seconds, with $r(t) \in [-5, 5]$. Next we used the procedure aforementioned only on the steady state portion of the measured current $i(t)$.

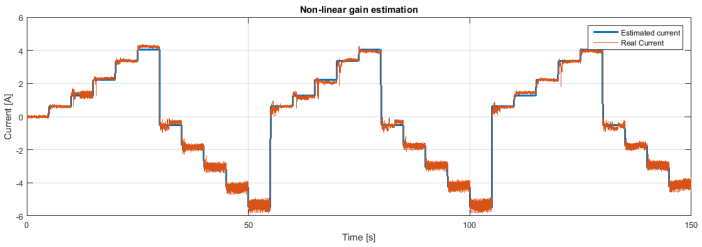
Finally we can model $f(v)$ in the following way:

$$f(v(t)) = \begin{cases} 0.6564 + 0.1712v - 0.0317v^2 + 0.0044v^3 & v > 0 \\ -0.21 - 1.1397v - 0.2756v^2 - 0.0226v^3 & v < 0 \end{cases} \tag{6.5}$$

And $g(v) = f(v)v$.



(a) Plot of $g(v), v$.



(b) Identified model vs real current.

Figure 6.1

6.2 Motor cogging identification

It was found that for low voltage the motor acts as if the voltage is 0, therefore the current and consequently the torque is 0.
This is a type of static friction, though it's not a mechanical one. In fact, if it was a mechanical one we would see the current increase but the shaft of the motor don't rotate until the torque is big enough to counteract the static friction effect. In this case is more an electrical type of static friction, called *cogging* effect.

Cogging torque of electrical motors is the torque due to the interaction between the permanent magnets of the rotor and the stator slots of a Permanent Magnet (PM) machine. This torque is position dependent and its periodicity per revolution depends on the number of magnetic poles and the number of teeth on the stator.Cogging torque results in torque as well as speed ripple; however, at high speed the motor moment of inertia filters out the effect of cogging torque.

In figure 6.2 such effect is shown. On the left plot it's visible that when the input voltage $v(t)$ for $|v(t)| < c$, the output current is 0. Since the torque is proportional to the current, also the torque is 0. On the right plot is visible the effect on the cart: when the cart starts to move, after having overcome the static friction, it stops because the torque goes to 0 because the current is 0. But this effect does not happen because of mechanical static friction, otherwise we would not see a flat current curve.

How can we model such effect? It's some kind of dead-zone effect, and we can model it such that the real voltage entering the motor is:

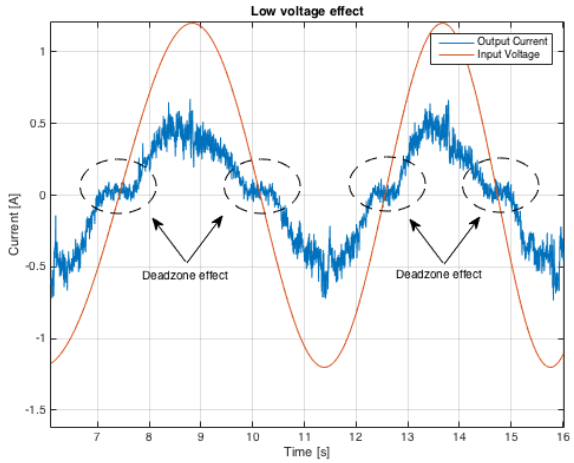
$$v_{real}(t) = \begin{cases} 0 & |v(t)| < c \\ v(t) & |v(t)| \geq c \end{cases} \tag{6.6}$$

Where $c \in \mathbb{R}$ needs to be identified. For this purpose, notice that inside the dead-zone the current is a white noise signal. Therefore we should find a constant α such that the signal $|i(t)| < \alpha$ is a white noise signal. We can make use of the fact that $v(t) \in C^\infty$ signal, find a set $\hat{t} = \{t \in \mathbb{R} : |v(t)| < c\}$ and check the whiteness of the current in that set $\hat{t}$. Since v is continuous, that set is measurable (using the Lebesgue measure). Also the current is continuous, therefore it's also continuous on that set. For this purpose the following identification procedure was used:

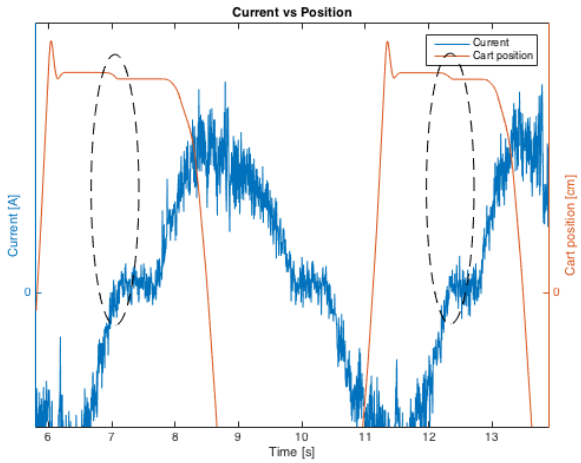
1. Use as voltage input a sinusoid with frequency pulsation $\omega = 1.25\ \mathrm{rad\,s^{-1}}$.
2. Choose $c \in [0, \max(v(t))]$, where $v(t)$ is the input voltage. Consider only the voltage that satisfies $|v(t)| < c$.
3. Run the Anderson Whiteness Test on the current for those values of t where $|v(t)| < c$, with margin of tolerance of 1%.
4. If the signal is not white, run again the test with $\alpha_{new} = \alpha - \Delta$, where $0 < \Delta < \alpha$. In our case Δ is fixed with value 10^{-2}. In case the signal is white, stop. Notice that the bisection method cannot be used.

Running this test gives an approximate value of $c \approx 0.35$ V. The simulated current, using the parameters identified for the motor, is shown in figure 6.3. Moreover, using the cost function mentioned at the beginning of this part, we obtain a fit value of 0.89 on the validation data (done using a different sinusoid).

Instead if we minimise a least squares problem $J = \frac{1}{2}\sum_{t=0}^{T}(i_{sim}(t) - i_{real}(t))^2$, using the *Genetic Algorithm* implemented in Matlab, gives a value of $c = 0.4621$.



(a) Current vs Voltage plot



(b) Current vs Position plot

Figure 6.2: On the left: plot of the input voltage and output current. The deadzone effect on the current sets the current to 0 for small v. On the right we can see the effect on the position of the cart.

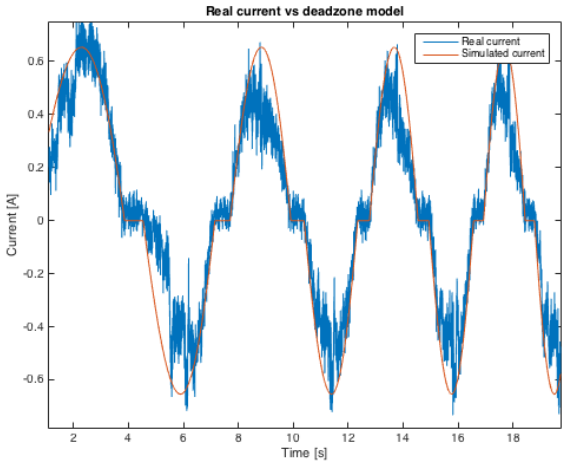


Figure 6.3: Identified model vs real current.

6.3 Cart static friction

Finally, of the major non-linearities, the static friction of the cart was evaluated. To do so we made use of the information regarding the non-linearities of the motor.
The most basic identification procedure is to linearly increase the voltage input to the motor until the cart moves, but since the motor is an intermediate system, we had to identify first the non-linearities of the motor.

Next we need to know the resolution of the encoder: the resolution is, as specified at the beginning of the report, of 14 bits. Therefore, since 560 is the factor we need to consider to convert the number of steps of the encoder to cm the final resolution is $\Delta = 0.0342$ cm. The input voltage used is:

$$v(t) = 0.03t \tag{6.7}$$

The static friction force is defined as:

$$f_s = \mu_s N \tag{6.8}$$

Where N is the vertical load and μ_s the static friction coefficient. The experiment was executed with only 1 cart and no loads, therefore the total load N is just the mass of the cart, which is 0.5685 kg.

The total force transmitted from the motor is

$$F(t) = \gamma i(t) \tag{6.9}$$

Therefore for t such that $F(t) = f_s$ we have:

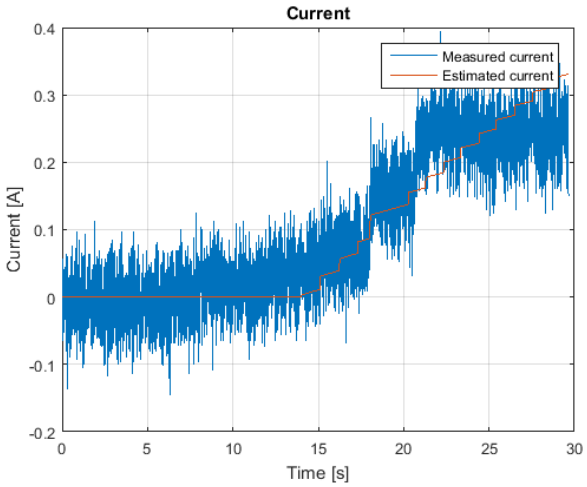
$$\mu_s = \frac{\gamma}{N} i \tag{6.10}$$

which is the static friction condition. Therefore we need to have an accurate knowledge of the current in order to estimate μ_s. At $t = 30$ the cart moves out of the encoder resolution and starts moving. For such t the current is about 0.3 A, therefore we can estimate that the static friction is:

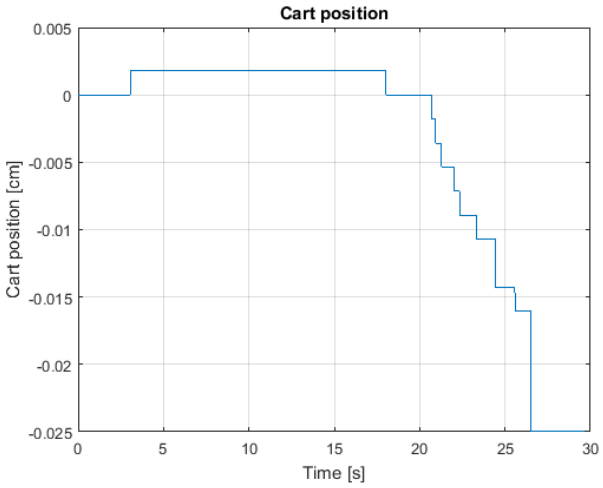
$$\mu_s \approx \frac{2.068}{0.5685} 0.3 \approx 1.1 \tag{6.11}$$

In figure is shown the plot of the current and its estimate. In figure the plot of the cart position.

Chapter 7



(a) Plot of $g(v), v$.



(b) Identified model vs real current.

Figure 6.4

State filtering and Model Observers

During the course of the project the development of a state observer was deemed necessary in order to overcome the following problems:

1. Faulty encoder sensors: the first sensor had some problems with its string, and the third encoder is broken.
2. Noisy current sensor, as described in the previous chapter.
3. State feedback: LQ, Pole Placement, $\cdots$.

Since the validated model, using white box techniques, has an overall validation fit of over 80%, we can make use of a linear observer, such as Luenberger's Observer or the Kalman Filter. In both cases we can model them as:

$$\dot{\hat{x}} = A\hat{x} + Bu + L(y - C\hat{x}) \tag{7.1}$$

Where L in case of the Luenberger Observer is chosen such that $A - LC$ has eigenvalues that are 10 times faster than the eigenvalues of the system. The Kalman Filter chosen L optimally (in a L_2 sense) by solving the Riccati's equation. Notice that u is the input voltage to the motor.

Both were tested, though a first implementation issue was the conversion of the observer to a discrete model, because of some problem with the Arduino Board in case we were using the continuous model. The model was formulated in continuous time and then discretised using a zero-order hold.

The following tests were done in order to validate the efficiency of the observer:

1. Estimate of the carts position (all 3 degree of freedom) using only the current.
2. Estimate of the carts position (second and third cart) using only the current and the data from the first encoder.
3. Estimate of the second and third carts position using only the data from the first encoder.

Regarding the first tests the first thing to notice is that the system is unobservable. In fact, by measuring only the current we cannot extract the system dynamics, thus results are identical to those extracted by the validation tests.

The second and third tests (figure **??**) were done in order to measure how much improves the estimate the fact that we measure the current in the second test. For sure measuring also the current *makes* the system more observable, in fact the minimum singular value of the observability matrix is about 0.16, whilst if we measure only the position of the first cart the minimum singular value is 0.03. But, this is not an indication that the current improves the estimate. In fact, as told before, the motor has dynamics much faster than those of the carts, therefore we can't say much using only the current.
Because of that the results of the two tests are almost identical, the only benefit of measuring also the current is that the Kalman filter is less ill-conditioned (because the observability matrix is less conditioned).

In the first case we obtain:

$$d(x_{2,kalman}, x_{2,real}) = 0.918, \frac{||x_{2,real} - x_{2,kalman}||_\infty}{||x_{2,real}||_\infty} = 0.181$$

In the second one:

$$d(x_{2,kalman}, x_{2,real}) = 0.912, \frac{||x_{2,real} - x_{2,kalman}||_\infty}{||x_{2,real}||_\infty} = 0.187$$

Where $d(x, y)$ is the distance function used in this project, and $\frac{||x_r - x_k||_\infty}{||x_r||_\infty}$ it's a measure of the relative error. As previously explained, results are almost identical.

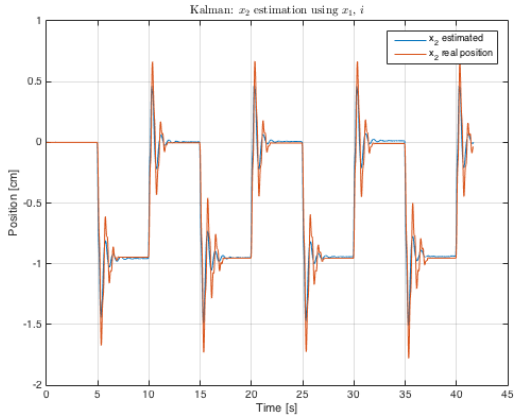


Figure 7.1: Estimate of the second cart position using i, x_1. On the first cart there are 0 masses, on the second one there are two, and on the third one there is only one. The spring setup is K_h, K_l, K_m.

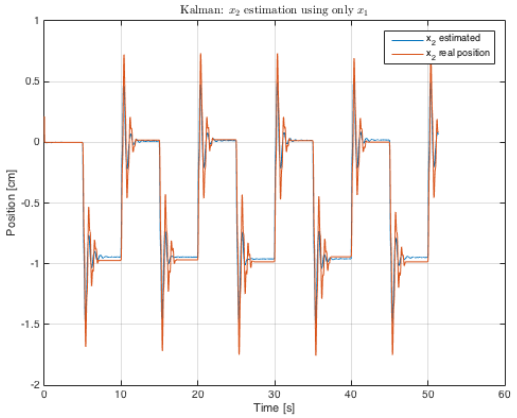


Figure 7.2: Estimate of the second cart position using x_1. On the first cart there are 0 masses, on the second one there are two, and on the third one there is only one. The spring setup is K_h, K_l, K_m.

7.1 Extended Kalman Filter

In this section we present the design of a filter which allows to estimate online the stiffness of a spring.

The setup is the one degree of freedom open loop model fed by a voltage, where the current and cart position can be sensed. The plant is described by the state-space model

$$\dot{x} = \begin{bmatrix} -\frac{R}{L} & 0 & 0 \\ 0 & 0 & 1 \\ \frac{\gamma}{\tilde{M}} & -\frac{K}{\tilde{M}} & -\frac{C}{\tilde{M}} \end{bmatrix} x + \begin{bmatrix} \frac{1}{L} \\ 0 \\ 0 \end{bmatrix} v(t) \tag{7.2}$$

and we suppose the value for K is unknown and needs to be estimated online. We include K as a state variable which is constant over time. The new system is nonlinear (K multiplies x_1)

$$\begin{bmatrix} \dot{i} \\ \dot{x}_1 \\ \ddot{x}_1 \\ \dot{K} \end{bmatrix} = \begin{bmatrix} -\frac{R}{L} i(t) \\ \dot{x}_1 \\ \frac{\gamma}{M} i - \frac{1}{M} K x_1 - \frac{C}{M} \dot{x}_1 \\ 0 \end{bmatrix} + \begin{bmatrix} \frac{1}{L} \\ 0 \\ 0 \\ 0 \end{bmatrix} v(t) \tag{7.3}$$

and we want to observe the state so to obtain the value of spring stiffness K, making use of the extended Kalman filter (EKF). The filter has been designed in discrete time, hence the equations above has to be discretized wrt the sample time T.

$$\dot{x} \approx \frac{x(k+1) - x(k)}{T} = f(x) + Bv(k) \implies x(k+1) = x(k) + Tf(x) + TBv(k) \tag{7.4}$$

Matlab does not provide an implementation for EKF and we needed to write it from scratch. We then designed a function which, given the current measurements of input $v(t)$ and output $i(t), x_1(t)$, outputs the state prediction $\hat{x}$. The *Matlab* implementation is listed in the appendix.

The verification of the observer was performed on *Simulink* making use of the scheme in figure which allow us to compare the true state with the observed one.

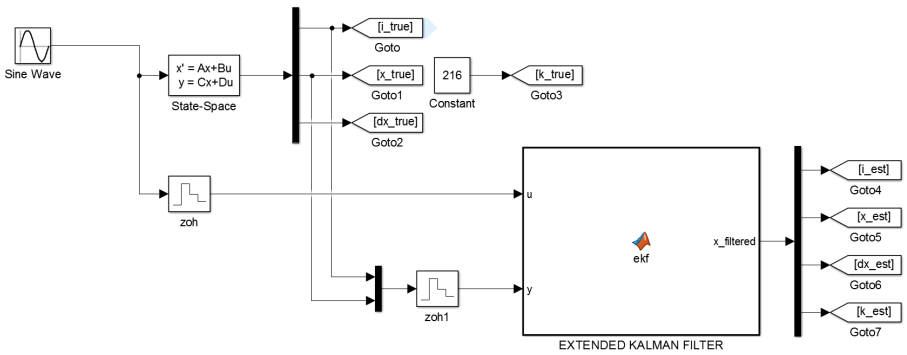
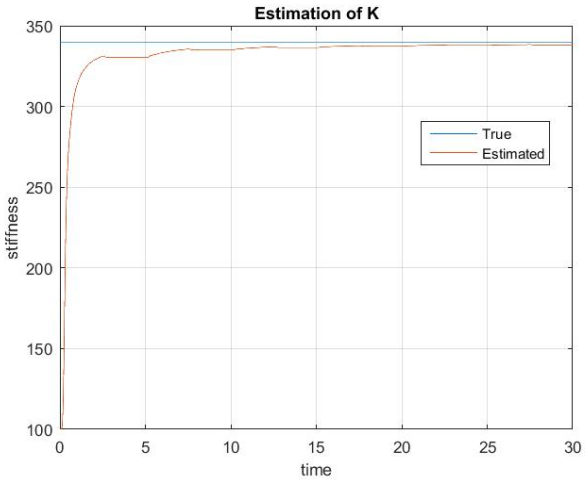


Figure 7.3: Simulink scheme used for verification.

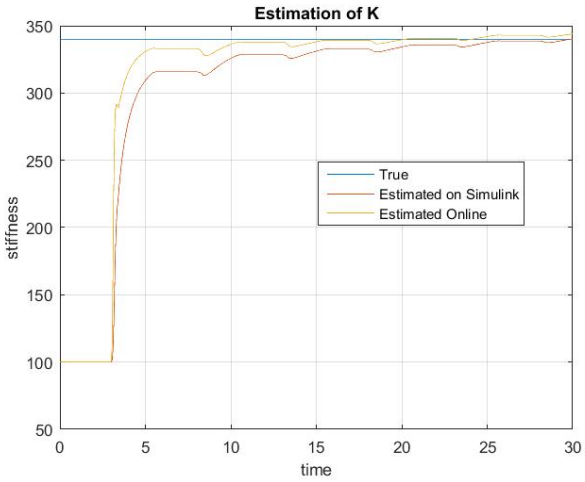
The results are satisfactory as the unknown parameter is estimated with good accuracy.

The system has then been ported onto Arduino and we ran the same experiment we performed offline. We expect the filter to converge to the same value as before but the transient might be different due to nonlinearities not included in simulink. The results are very satisfactory also in the online testing and resemble nicely the simulation.

The plot shown in figure 7.4 has been obtained by measuring the output of the EKF on arduino and, as comparison, the output of the filter on *Simulink* fed by the measured signal on the board in order to compare the outputs using the same data. Although we used the same solver, the results are a little different, probably due to numeric error on the Arduino board.



(a) Estimation by feeding a train of pulses in *Simulink*.



(b) Online estimation by feeding a train of pulses. Data measurements are fed to *Simulink* in order to be able to compare the outcome.

Figure 7.4: Estimation of the spring stiffness in one degree of freedom.

Chapter 8

Control of 1 Degree of Freedom

8.1 Loopshaping

The first attempt to control the positioning of the cart consists in the simplest technique of directly shaping the bode diagram of the open loop transfer function.

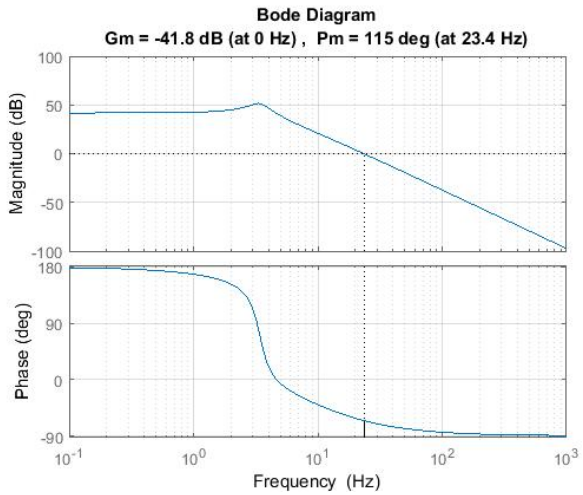


Figure 8.1: Bode diagram of the loop transfer function of the plant (motor and cart).

We consider the simplest model in order to perform the tuning of the controller. The motor and cart are represented by the linear transfer functions

$$F(s) = \frac{K_e}{Ls + R} \qquad X(s) = \frac{1}{Ms^2 + Cs + K} \tag{8.1}$$

connected in series, where $F(s)$ represents the linear force exerted by the motor and $X(s)$ the position of the cart. The parameters have been identified as explained in detail in the previous section. It is important to remind that the identification has been performed without scaling the encoder values into centimeters; this means that the conversion to and fro centimeters must be performed outside the control loop.

The openloop plant described by the equations above has three poles: one high frequency pole coming from the motor dynamic, and two complex conjugate poles placed nearby the imaginary axis. The latters are responsible from the oscillatory behaviour of the cart and need to be canceled out from the openloop. Furthermore, the dynamic does not contain any integral action, so we need to introduce it in the controller if we want to achieve zero static error at regime (namely, a perfect positioning of the cart).

The considered model for the controller is then:

$$R(s) = K_R \frac{(s + z)(s + z^*)}{s(s + 100)} \tag{8.2}$$

where z, z^* are the openloop complex conjugate poles of the plant, which are trivially dependant on the configuration of the plant (namely, spring and mass). Since we do not know the exact position of this pair of poles (making it impossible to perfectly cancel them out), it is better to position the pair of zeros on the left side of the expected position; in this way we are attracting the poles away from the imaginary axis. The pole in $s = -100$ is only needed to obtain a causal regulator and its positioning in high frequency will not perturb the control dynamic in any way. Eventually, the controller gain has to be tuned in order to obtain a desired cutoff frequency which will affect the response time of the controlled plant as well as its stability.

We then build a simulink model to fine tune the parameters taking into account the saturation of the control input, which affects the maximum bandwidth achievable. The challenge is to push the performances (bandwidth) as close as possible to the saturation of the actuator: this way we are getting the best performance allowed in the linear region of operability. The target output we expect is a positioning time below one second, resulting in a bandwidth around 1 Hz.

Once satisfied with the simulation, it is time to upload the controller onto Arduino for the real test. We cannot expect that the output of the simulink model will perfectly match the real test, but we do expect to see the same settling time, gain and no saturation for the actuator. As expected the transient does not match the simulink model (which is probably due to the not considered nonlinearities of the plant), but this difference can hardly be detected by the naked eye.

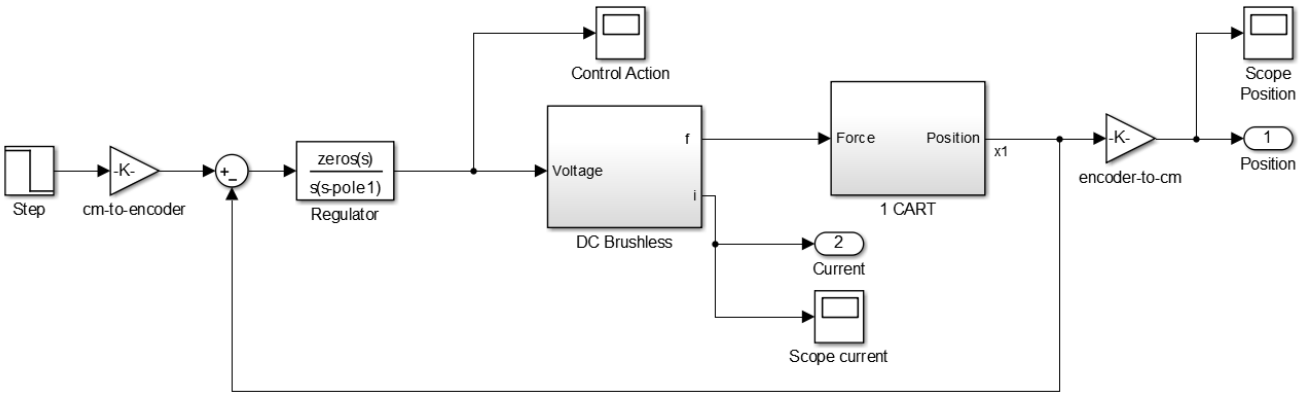


Figure 8.2: Simulink model used for verification of the control design.

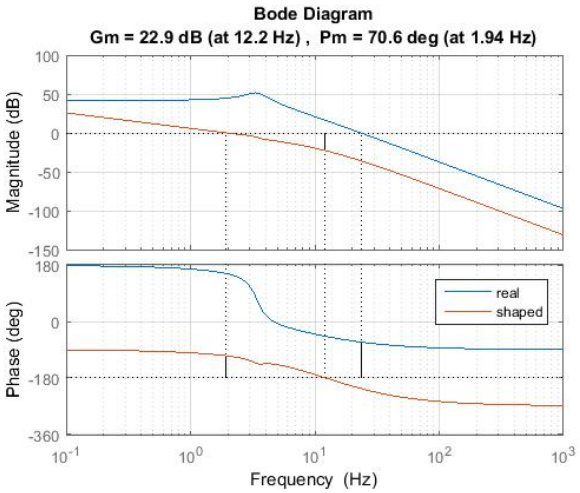
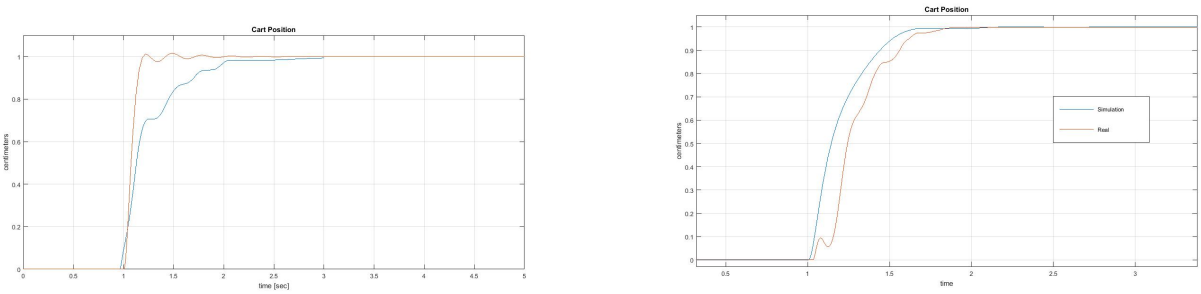


Figure 8.3: Bode diagram of the loop transfer function after the shaping.



(a) Strongest spring K_h.

(b) Weakest spring K_l.

Figure 8.4: Comparisons of the step response between the simulated system and the real one.

8.2 RHP-Zeros Control

During the development of the project it was apparent that using an integrator was necessary to eliminate steady state-error and reduce the effect of non-linearities due to the high-gain linearization effect induced by the integrator. Though, this led to some drawbacks, such as low cut-off frequency. This was induced by the fact that we have 2 complex poles near the imaginary axis, which are the cart natural frequencies. One of the best way to design a controller was to introduce two complex zero in order to cancel the effect of those complex poles. Another way was to use positive zeros in our controller.

It is well known from the root locus technique that zeros *attract* poles for increasing gain. Then it was observed, that, if we place positive zeros sufficiently near to the real axis, and not too far from the complex poles aforementioned, those poles with a sufficient gain would fall in a zone where the output signal response of the system would have low overshoot.

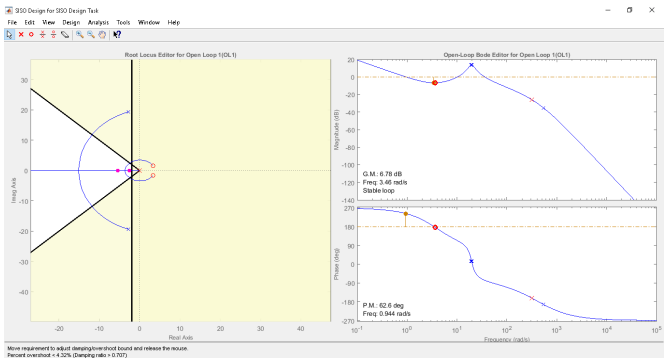


Figure 8.5: Bode diagram of the loop transfer function of the plant (motor and cart).

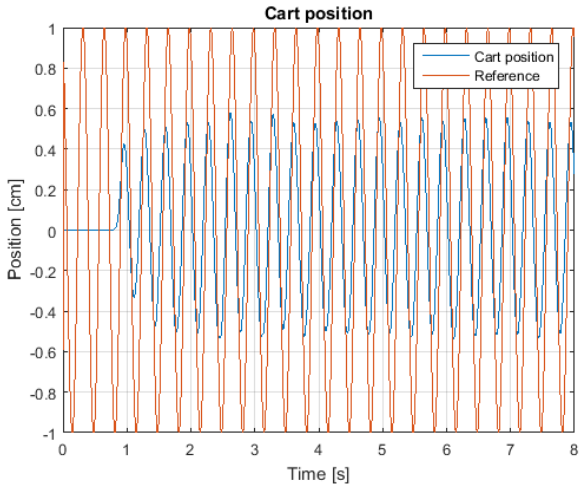
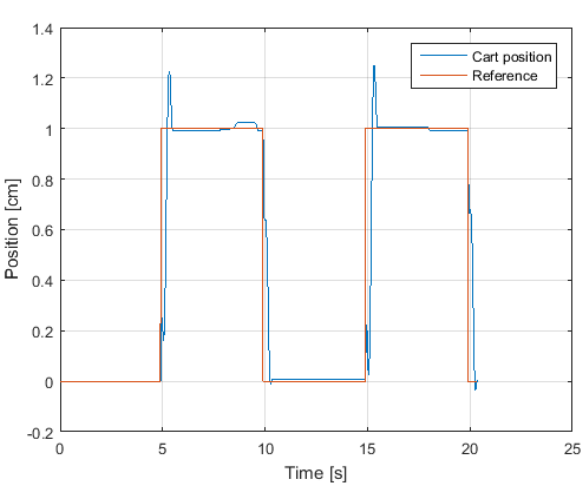
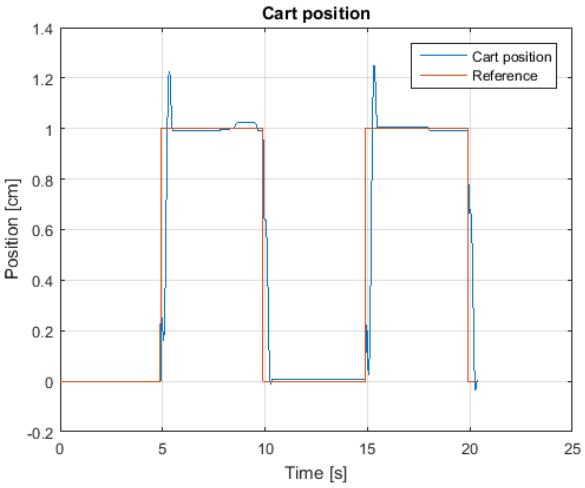


Figure 8.6: Bode diagram of the loop transfer function of the plant (motor and cart).



(a) LQG K_l



(b) LQG K_h

Figure 8.7: On the left: plot of the cart position with LQG control and K_l. On the right plot of the cart position with K_h.

8.3 H_∞ Control Design

In this section we show how to design a H_∞ controller for this system. The augmented plant consists in three shaping functions which are in charge of designing the closed loop sensitivity, complementary sensitivity and effort sensitivity; look at Appendix for further details on H_∞.

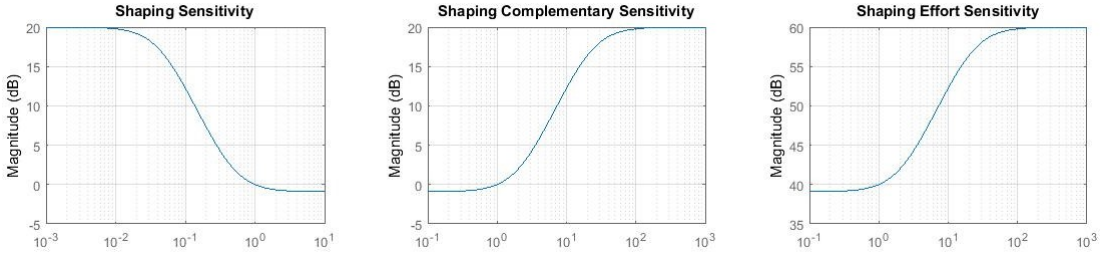
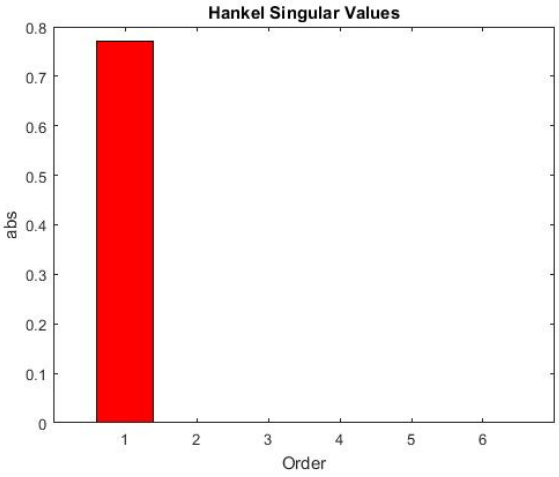


Figure 8.8: Shaping functions.

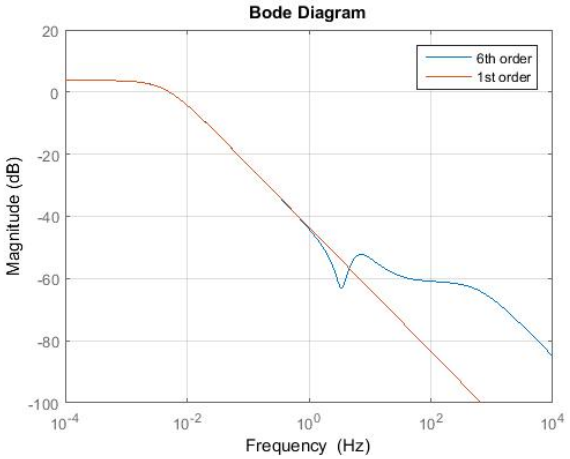
Each function has three degrees of freedom: low frequency gain, cross-over frequency and high frequency gain. The most important one is the cross-over frequency which defines the closed loop bandwidth of the system; in our case, we are aiming at a response time of about one second which corresponds to a cross-over frequency of 1 Hz ($\omega = 2\pi$ rad s^{-1}). The gains are tuned so to obtained the desired static error at regime, since we are not including any integral action; the higher the gain the smaller the static error will be. The crucial function in our application is the sensitivity wrt the control input because our actuator constrains us to work in the range $\pm 5V$; hence, we need to give more weight to penalize the control effort and obtain a feasible action; this last process is performed by an iteration process of trial and error: given a shape, we test the correspondant controller on simulink and check how the control input behaves; if the input saturate then we need to weight more the effort.

The controller output of the H_∞ design has as many poles as the augmented plant; in our case, the plant has 3 poles (two from the cart and one from the motor) and so the shaping functions (one pole each), thus the final controller is a 6th order transfer function. Since we previously saw that the loopshaping technique achieves good performances with a 3rd order, it is interesting to investigate the option of reducing the complexity of the controller. With this aim, we compute the singular values of the Hankel matrix in order to see how much information is contained by every degree. It turns out that only the first order of complexity brings enough information and can approximate the controller in the bandwidth of interest. In practice, approximating to the third order is a robust solution which will not perturb for sure the theoretical results.

The downside of this controller is that it does not have any integral action, but this can be easily fixed by moving the slowest pole (of the controller) to zero. This way we are introducing an integral action without affecting the other properties of the designed controller. In Matlab this can be done by converting the state-space model of the controller into a *zpk* model and by literally editing the slowest pole into zero.

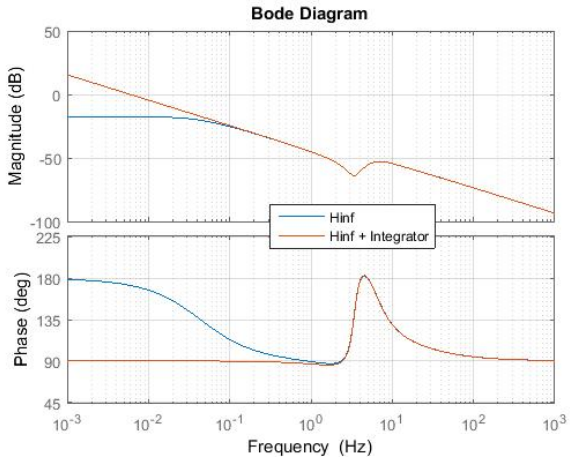


(a) Singular values of the Hankel matrix.

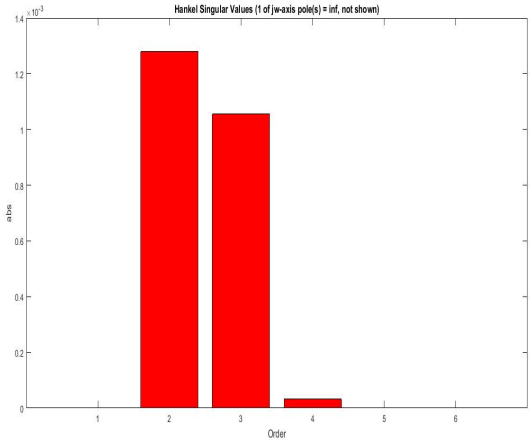


(b) Approximation with only the first order.

Figure 8.9



(a) Introduction of the integral action into the controller.

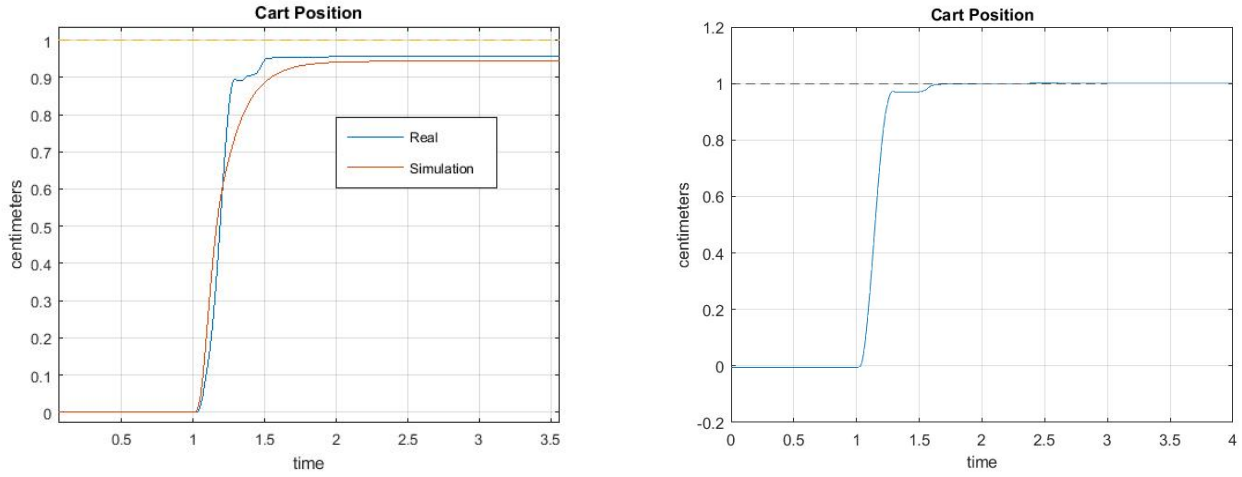


(b) Singular values of the Hankel matrix for the controller with integral action.

Figure 8.10

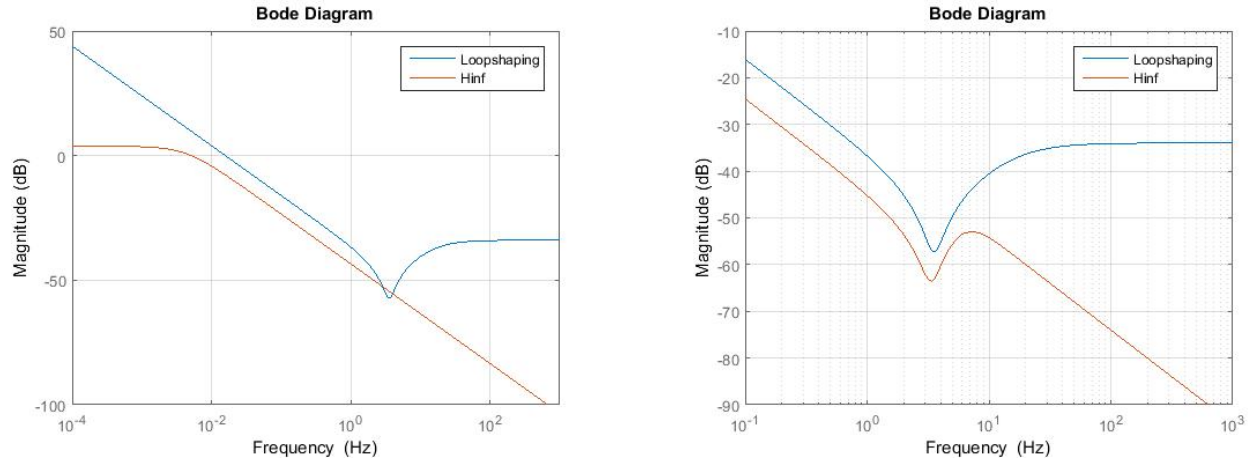
Again, we can apply a system reduction to the controller by analizing the singular values of the Hankel matrix. This time the minimum order to get a good approximation is 3, and the bode diagram perfectly overlaps the original one.

We now focus on the step responses of the designed controller. It is interesting to see the difference between the design with and without integral action. As it is expected the integrator guarantees zero steady state error and gives better overall performance. The final controller obtained is not much different from the one designed by loopshaping as it is easy to see by plotting the bode diagrams; in particular the H_∞ solution gives more attenuation for higher frequencies.



(a) H_∞ without integrator (b) H_∞ with integrator.

Figure 8.11: Comparison between step response of H_∞ controller.



(a) H_∞ without integrator (b) H_∞ with integrator.

Figure 8.12: Comparison between Loopshaping and H_∞.

8.4 LQG Control Design

Of the feedback schemes, such as *Pole-Placement*, we decided to choose the *LQR* control design. Unfortunately, because of noise in the current sensor, a Kalman Filter was needed. In fact a simple low pass filter on the current feedback was not adequate enough, since it would make the system too slow.

The Kalman Filter made use of all the available inputs, and only for the case of $2,3$ degree of freedom the non-linearities of the motor were considered in order to improve the performances.

Remember that LQR solves the optimal problem:

$$minJ = \int_0^\infty x^T Q x + u^T R u dt \tag{8.3}$$

which gives an optimal solution in the form of feedback control:

$$u = -Kx \tag{8.4}$$

First we decided to try an LQG design with an integrator, by adding an extra state:

$$\dot{e} = -x + r \tag{8.5}$$

But this gave unacceptable performances. To have good performances, by simulations we needed to have the state matrix weight Q with very large eigenvalues, greater than 10^2. Because of that the feedback gain K was very

large, and the control input u saturated the motor, and made the system unstable. The integrator problem can be explained by root locus techniques, as explained in the Positive Zeros Control section.

To solve this problem we removed the integrator and added a feedfoward gain on the reference:

$$u = -K_x x + K_r r \tag{8.6}$$

Where K_r is:

$$K_r = \frac{1}{C(-(A - BK))^{-1}B} \tag{8.7}$$

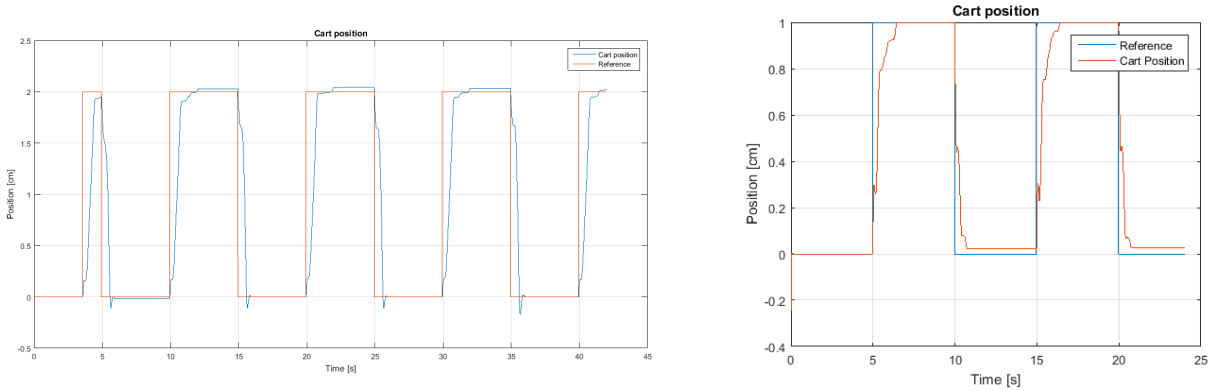
With C selecting the appropriate output, in this case x_1, thus $C = \begin{bmatrix} 0 & 100 & 0 \end{bmatrix}$. This K_r makes sure the system has unitary gain.
The weight matrix was selected in order to let $x_1 \to 0$ quickly with acceptable performances. After some iterations the matrix Q was chosen as:

$$Q = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{8.8}$$

And $R = 1$. Moreover, the LTR procedure was not considered since there is no noise in the control input channel.

Results are shown below for K_l and K_h. Notice the static error, induced by the fact that the non-linearities of the motor were not considered (such as the dead-zone, which changes the gain of the system).

Chapter 9



(a) LQG K_l (b) LQG K_h

Figure 8.13: On the left: plot of the cart position with LQG control and K_l. On the right plot of the cart position with K_h.

Control of 2 Degree of Freedom

9.1 LQG Control Design

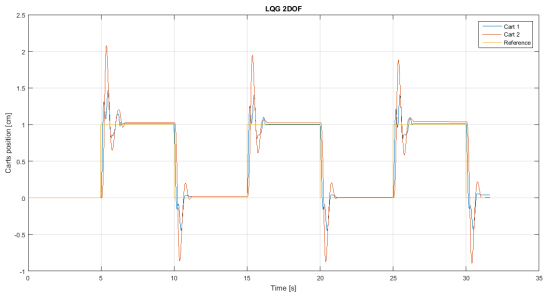


Figure 9.1: Openloop frequency response from the motor input to the position of the second cart in the case of low and medium springs employed.

9.2 H_∞ Control Design

This control strategy gave good results for one degree of freedom, hence we choose it again to be implemented with two carts.

One important aspect is worth pointing out. The critical aspect in this scenario is which output feedback to the control input. If we feedback the first cart position, the control strategy won't be much different from the 1 degree of freedom, since the second cart is just equivalent to a heavier mass placed onto the first cart. This statement is confirmed by the fact that the controller designed in the previous chapter works fine in this scenario, meaning it is robust enough to uncertainties on the model.

Controlling the second output is instead more challenging since we have two springs in between the control action and the output to be controlled. In this case the controller must take into account both carts dynamics, and this is what we will discuss in this chapter.

The plant, defined as the series of actuator followed by two carts is a 5th order system, where two complex conjugate poles come from the first cart, two from the second cart and one high frequency pole from the motor.

The controller design is exactly the same as the one presented in the previous chapter, hence we won't replicate it here but will only show the results.

As it possible to see, the current has some oscillations which perturb the whole system and gives very poor performance. The first attempt was to reduce te bandwidth of the controller but the output, although deprived from oscillations, resulted too slow (over two seconds in order to get to steady state) and therefore not acceptable.

We decided to make use of a Kalman filter in order to insert an inner loop controlling the current. The result showed an important boost in performance. The rising time is now satisfactory and the current does not show any oscillation.

Chapter 10

Control of 3 Degree of Freedom

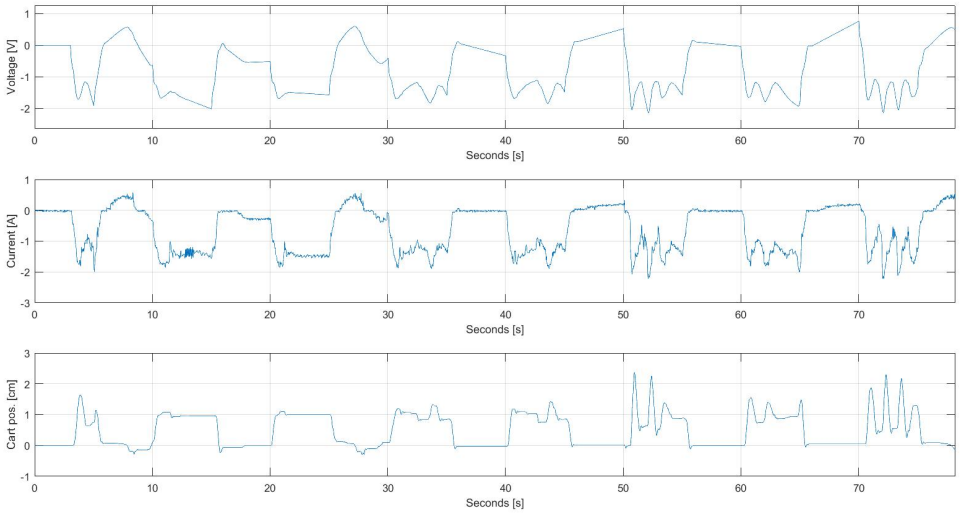


Figure 9.2: Response to a train of pulses when the current is not controlled. It is hard for the system to maintain the value at steady state.

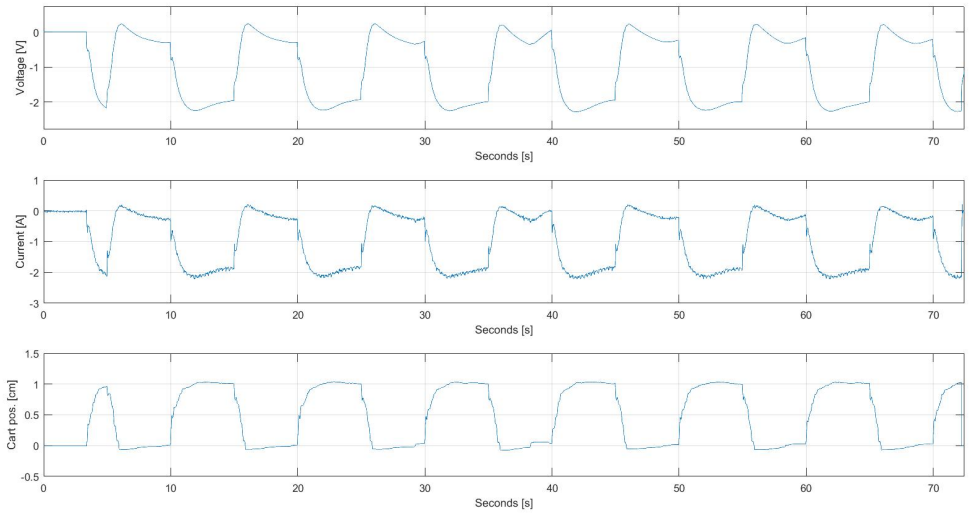


Figure 9.3: Response to a train of pulses when the current is controlled. Now the response is much smoother.

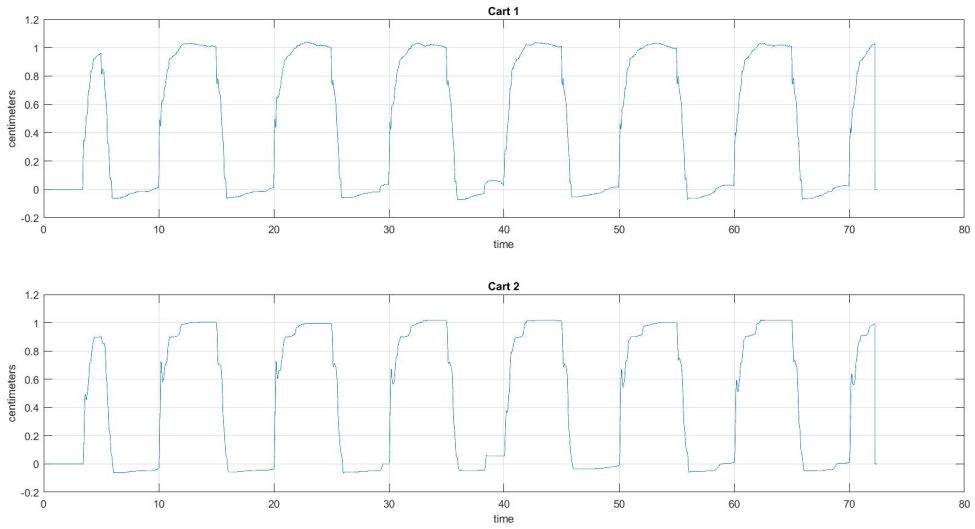


Figure 9.4: Carts displacement over time due to a train of pulses. Both carts do not overshoot.

10.1 LQG Control Design

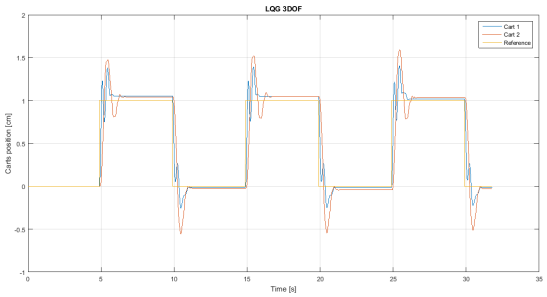


Figure 10.1: Openloop frequency response from the motor input to the position of the second cart in the case of low and medium springs employed.

10.2 H_∞ Control Design

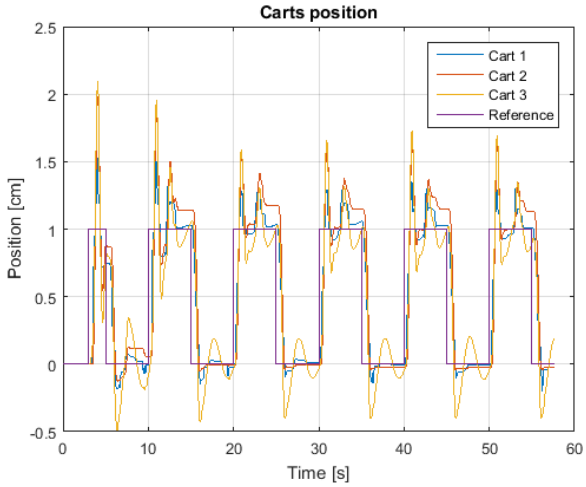


Figure 10.2: Openloop frequency response from the motor input to the position of the second cart in the case of low and medium springs employed.

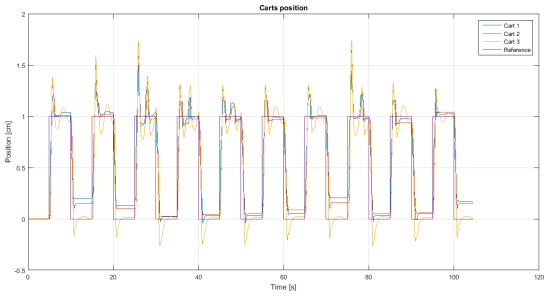


Figure 10.3: Openloop frequency response from the motor input to the position of the second cart in the case of low and medium springs employed.

Conclusions

The team

The team is composed by 3 people, all holding a B.Sci. in Engineering:

1. *Alessio Russo*: B.Sci. degree in Computer Engineering, enrolled at the M.Sci. degree Automation and Control Engineering at Politecnico di Milano. He's also an ASP student, and his thesis will focus on Robust and Adaptive Control of Quadrotors.
2. *Gianluca Savaia*: BSc in Computer Engineering and currently attending the MSc Automation and Control Engineering. Master project focuses on autonomous drive and he is in charge of designing a steering control.
3. *Alberto Ficicchia*: BSc degree in Electronics Engineering, enrolled at the M.Sci. degree in Automation and Control Engineering at Politecnico di Milano. His thesis will focus on Stochastic Model Predictive Control, comparing analytical and scenario approaches.

Appendix

10.3 H_∞ Control Design

This control strategy consists in an optimization problem where the designer can specify a performance output to minimize; this minimization problem may be constructed in many possible ways by augmenting the plant with as many shaping function as the designer is willing to include.

In this scenario we decided to use the most common set-up given in the scheme depicted in figure 10.4. This scheme is valid for every scenario (number of degrees of freedom) described in this report, where $G(s)$ is the transfer function of the focus plant.

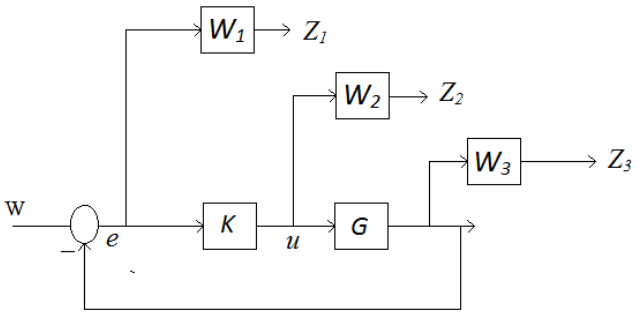


Figure 10.4: Augmented plant. $G(s)$ represents the transfer function of the actuator and cart. $K(s)$ is the output controller of the H_∞ minimization problem wrt the shaping functions $W_1(s), W_2(s), W_3(s)$.

The minimization problem is stated as:

$$\min_{W_1,W_2,W_3} \left\| \begin{matrix} z_1 \\ z_2 \\ z_3 \end{matrix} \right\|_\infty = \min_{W_1,W_2,W_3} \left\| \begin{matrix} S(s)W_1(s) \\ K(s)W_2(s) \\ T(s)W_3 \end{matrix} \right\|_\infty \tag{10.1}$$

where, by letting $L(s) = K(s)G(s)$ be the openloop transfer function, the transfer functions S, K, T are defined as

$$S(s) = \frac{1}{1+L(s)} \tag{10.2}$$

$$K(s) = \frac{K(s)}{1+L(s)} \tag{10.3}$$

$$T(s) = \frac{L(s)}{1+L(s)} \tag{10.4}$$

10.4 Effect of positive zeros

Now it's briefly described what are the drawbacks of RHP-zeros in the loop. From a frequency analysis we can get an insight of what happens: RHP-zeros add -180 degree to phase, therefore it's best to have those zeros at high frequency (thus the modulus of the zeros) to have their effect quickly dissipate without any effect. To analyze in time we need to take the partial fraction expansion. Let $G(s)$ be our system considered and $C(s)$ our controller. Then:

$$C(s) = \hat{k}\frac{(s-a)(s-\overline{a})}{s(s+p)}, a \in \mathbb{C}, p \in \mathbb{R}$$

where $\overline{a}$ is the complex conjugate of a, and $\hat{k} = \frac{kp}{|a|^2}$, with $\mathbb{R}(a) > 0$. The closed loop transfer function is:

$$T(s) = \frac{CG(s)}{1 + CG(s)} = \frac{\hat{k}(s-a)(s-\overline{a})\gamma}{s(s+p)(Ls+R)(Ms^2+Cs+K) + \hat{k}(s-a)(s-\overline{a})\gamma}$$

With output signal $y(t) = \mathcal{L}[T(s)R(s)]^{-1}$. For a certain k it's possible to stabilize the system, that can be seen using root locus technique. Notice that the motor since has a pole in $s \approx 600\ \mathrm{rad\,s^{-1}}$ its term sL can be ignored. In closed loop, it is possible to demonstrate that we have 2 complex conjugate poles in the left half plane, and two negative poles on the real axis that if the gain increases more will become complex conjugate poles. Thus we have 4 poles, and the transfer function can be rewritten as:

$$T(s) = \frac{\hat{k}(s-a)(s-\overline{a})\gamma}{(s+p_1)(s+p_2)(s^2+2\alpha s+\beta)}$$

Where $p_2 > p_1 > 0$. The partial fraction expansion becomes:

$$T(s) = \frac{A}{s+p_1} + \frac{B}{s+p_2} + \frac{Cs+D}{s^2+\alpha s+\beta}$$

and in time it's:

$$y(t) = Ae^{-p_1 t)} + Be^{-p_2 t} + Ce^{-\alpha t}\Big(\cos(\theta t) + \frac{\frac{D}{C}-\alpha}{\theta}\sin(\theta t)\Big)$$

Where $\theta^2 = \beta - \alpha^2$. Making use of the fact that for example $A = \lim_{s\to -p_1}(s+p_1)T(s)$ we obtain:

$$A = \frac{\hat{k}(-p_1-a)(-p_1-\overline{a})\gamma}{(-p_1+p_2)(p_1^2-2\alpha p_1+\beta)}$$
$$B = \frac{\hat{k}(-p_2-a)(-p_2-\overline{a})\gamma}{(-p_2+p_1)(p_2^2-2\alpha p_2+\beta)}$$

And

$$\lim_{t\to 0^+}\dot{y}(t) = \lim_{s\to\infty} sT(s) = A + B + C = 0 \Rightarrow C = -A - B$$

$$T(0) = \frac{\hat{k}a^2\gamma}{p_1 p_2 \beta} = \frac{A}{p_1} + \frac{B}{p_2} + \frac{D}{\beta}$$

It can be proven that $\operatorname{sign} C = -\operatorname{sign}\mathbb{R}(a)$, whilst A, B, D mantain their sign. In fact C is:

$$C = \frac{-\hat{k}\gamma}{p_2-p_1}\Big(\frac{(-p_1-a)(-p_1-\overline{a})}{p_1^2-2\alpha p_1+\beta} - \frac{(-p_2-a)(-p_2-\overline{a})}{p_2^2-2\alpha p_2+\beta}\Big)$$

Notice that the sign of C, with the hyphotesis $p_2 > p_1$, depends only on the numerator of the fractions inside the brackets (not the denominators, which are always positive):

$$C = \frac{-\hat{k}\gamma}{p_2-p_1}\Big(\frac{p_1^2-2p_1\mathbb{R}(a)+a^2}{p_1^2-2\alpha p_1+\beta} - \frac{p_2^2-2p_2\mathbb{R}(a)+a^2}{p_2^2-2\alpha p_2+\beta}\Big)$$

For $\mathbb{R}(a) < 0 \Rightarrow C > 0$ and viceversa. This effects leads to $\dot{y}(t) < 0$ for some t, causing the *undershoot* effect.

More in general, by looking at the expression of $T(s)$, it's possible to notice that the effect of $\mathbb{R}(a)$ is visible starting from the $r+1$ time derivative of $y(t)$, where r is the relative degree, thus $\operatorname{sign} y^{r+1}(0) = \operatorname{sign}\hat{k}\gamma\mathbb{R}(a)$. But derivative up to the r-th order are positive, thus for this reason the undershoot effect is delayed and does not appear right at the beginning.

Finally an even number of positive zeros display the undershoot effect after a small time. On the other hand it can be proven [1]]that an odd number of positive zeros show the undershoot effect right at the start. Obviously the effect of those zeros can be reduced by moving them far away from the origin (i.e. in high frequency), where the gain of the system is very low, but, then, the root locus would change.

10.5 Extended Kalman Filtering

Because of some problems with the Arduino Board with the continuous Kalman Filter we decided to use the discretized version of the Extended Kalman Filter. The theoretical details can be found at **??**. To discretize the system the Forward Euler Method was used, therefore $\dot{x} \approx \frac{1}{t_s}(x_{n+1} - x_n)$, where t_s is the sampling time. Then, if the system in continuous time is described by the following equation:

$$\dot{x} = Ax + Bu \tag{10.5}$$

In discrete time is:

$$x_{n+1} = (I + At_s)x_n + Bt_s u \tag{10.6}$$

Below is shown the matlab implementation of the algorithm:

```
function x_filtered  = ekf(u, y)
    %% Discrete Kalman filter for continous system
    % Inputs:
    % u: input to the system
    % y: output of the system
    % Output:
    % x_filtered: Kalman state estimate


    % P and xhat need to be updated each iteration
    % The other variables are parameters of the real vector.
    persistent P xhat Q R Res L M Gamma C B H

    if isempty(P)%First iteration, initialize values
        P = eye(4);
        xhat = [0; 0; 0; 100];
        Q = 1e-10*eye(4);
        R = diag([0.0127; 1e-10]);

        M = 1.8;
        C = 8.9973;
        Res = 1.3;
        L = 0.0024;
        Gamma = -2.068;

        B = 1/200.*[1/L; 0; 0; 0]; %Discretized Input Matrix B
        H = [1 0 0 0; 0 100 0 0]; % Output current and position
    end

    % Linearize the continuos system around xhat
    A = [ -Res/L,        0,              0,      0; ...
          0,             0,              1,      0; ...
        Gamma/M,         -xhat(4)/M,     -C/M,   -xhat(2)/M; ...
        0,               0,              0,      0];

    A = eye(4) + 1/200.*A; %Discretize A

    % Real nonlinear system computed in xhat
    f = [ -Res/L*xhat(1); ...
          xhat(3); ...
          Gamma/M*xhat(1) - xhat(4)*xhat(2)/M - C/M*xhat(3); ...
          0 ];

    % Discretize f
    f = 1/200.*f;
    f = f + xhat;

    % Update Prediction
    x_pred = f + B*u;   % x_k+1 = f(x_k) + Bu_k
    y_pred = H*x_pred;  % y_k+1 = H*x_k+1

    % Update P
    P = A*P*A' + Q;

    % Update Kalman Gain
    K = P*H'/(H*P*H'+R);

    %Update Prediction
    xhat = x_pred + K*(y - y_pred);

    % Update P
```

```
P = (eye(size(K*H)) - K*H)*P;

% Output the state-estimate
x_filtered = xhat;
```

Bibliography

[1] Dennis S. Bernstein Jesse B. Hoagg. "On the Zeros, Initial Undershoot, and Relative Degree of Lumped-Mass Structures". In: *Proceedings of the American Control Conference* (2006).