TÉCNICO LISBOA

STATE-SPACE CONTROL

Linear State Feedback Control

Francisco Gonçalves, 78479
Luís Rei, 78486
João Girão, 78761
João Belfo, 78913

June 6, 2017

Abstract

In control theory, the linear–quadratic–Gaussian (LQG) control problem is one of the most fundamental optimal control problems, including those related to perturbed non-linear systems. In this report an implementation of such a controller is described to solve the equilibration problem of a Furuta pendulum in its unstable upwards position, and applied within a region of the state-space that is close to the upwards, zero velocity, state. The Furuta pendulum is a rotational pendulum that is actuated at its basis by a direct current motor with a gear.

This work illustrates a situation of fast prototyping, in which a simple cyber-physical system (a system with interacting physical and computational parts) is created.

The algorithms and controllers developed and used are described, as well as simulation and experimental results.

Keywords: Optimal control, Furuta Pendulum, Linear control

Contents

List of Figures

List of Tables

1 Introduction

The aim of this project was the development of an LQG controller to balance a Furuta pendulum at its upwards, unstable, equilibrium position. The LQG controller can be used to deal with uncertain linear systems disturbed by additive white Gaussian noise, without having the complete state information by undergoing control subject to quadratic costs. When applied to linear time-varying systems it enables the design of linear feedback controllers for non-linear uncertain systems. Since this type of controller is simply the combination of a Kalman filter, i.e. a linear–quadratic estimator (LQE), with a linear–quadratic regulator (LQR), the separation principle guarantees that these can be designed and computed independently.

In order to solve this problem a linearized model of the system is given. Controllability and observability studies are then conducted on said system and the LQR and LQE are solved. Several weights for the observer and controller are tested in the laboratory in order to obtain the best design possible (described as little variation as possible of the angles of the Furuta pendulum). We present below our results and the combination that allowed us to maximize the performance of the controller. We analyze the inconsistencies between the approximated simulated system and the real-world application.

2 Plant Model

The plant model is nonlinear - it is found in appendix A.

An equilibrium point consists of the pendulum pointing upwards, with all the variables still. This equilibrium corresponds to all the variables being zero. If the variables denote increments with respect to this equilibrium, a linearized model that represents the plant with reasonably good accuracy is given by

$$\frac{\partial x}{\partial t} = Ax + Bu \tag{1}$$

$$y(t) = Cx(t) + Du(t) \tag{2}$$

You can find matrices A, B, C and D in the MATLAB data structure 'fpmatrices.mat' in an annexed folder. The state is the vector of dimension 5 given by (see figure 1).

Table 1: State variables dictionary

State Variable	Variable
x_1	α
x_2	$\dot{\alpha}$
x_3	β
x_4	$\dot{\beta}$
x_5	i

in which α and β are the angle of the horizontal bar and the angle of the pendulum with respect to the vertical, respectively, as can be seen in figure 1. The variable i is the motor current.

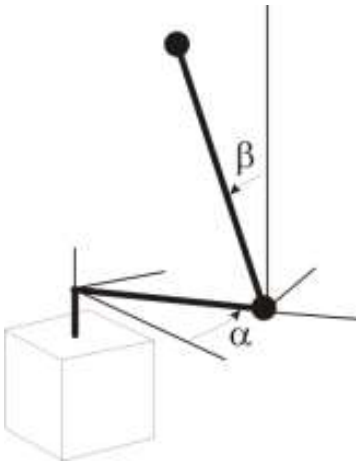


Figure 1: Inverted pendulum

3 Home preparation

In this section we will focus on answering the questions to be done at home.

3.1 Question 1.

Through the linearized control system available to us we extract the eigenvalues of the system's dynamic matrix (matrix "A"):

$$\lambda = \{0,\ -737.3184,\ -19.3673,\ -5.7612,\ 6.99740\}.$$

There is one eigenvalue whose value has a positive real part, allowing us to conclude that the open loop system is unstable.

3.2 Question 2.

A system is controllable if the rank of the controllability matrix (3) is equal to the dimension of the system's state. In this particular system, the dimension of the state is 5.

$$Ctr = \begin{bmatrix} 0 & 0 & 1.9178\times10^{+04} & -1.44885\times10^{+07} & 1.0689\times10^{+10} \\ 0 & 1.9178\times10^{+04} & -1.4488\times10^{+07} & 1.0689\times10^{+10} & -7.8814\times10^{+12} \\ 0 & 0 & 1.9146\times10^{+04} & 1.4464\times10^{+07} & 1.0672\times10^{+10} \\ 0 & 1.9146\times10^{+04} & 1.4464\times10^{+07} & 1.0672\times10^{+10} & -7.8690\times10^{+12} \\ 333{,}3367 & -2.5181\times10^{+05} & 1.8577\times10^{+08} & -1.3698\times10^{+11} & 1.0100\times10^{+14} \end{bmatrix} \tag{3}$$

Using MATLAB, we see that the rank of this matrix is 5, and so, equal to the dimension of the state - the open loop system is controllable.

3.3 Question 3.

A system is observable if the rank of the observability matrix is equal to the dimension of the state. Using MATLAB, we were able to compute the observability matrix. The output matrix of the system (4) considers the output being x_1 and x_3.

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \tag{4}$$

To consider the output just being x_3 we just have to eliminate the first line of matrix "C". After that the new matrix will be defined as

$$C = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \end{bmatrix} \tag{5}$$

Now we can compute the second observability matrix. In (6) we consider the output being only x_3, meaning, we consider the matrix (5) in the calculations. In matrix (7) we consider the output being x_1 and x_3, meaning, we consider (4) in the calculations.

$$Obs1 = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & -0{,}0174 & 63{,}8319 & -0{,}0071 & 57{,}4388 \\ 0 & -1.3327\times10^{-04} & -0.8152 & 63{,}8320 & -4.3392\times10^{+04} \\ 0 & 1.0068\times10^{+07} & -2.7295\times10^{+05} & 29.6025 & 3.2016\times10^{+07} \end{bmatrix} \tag{6}$$

$$Obs2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & -0{,}01744 & 20{,}7861 & -0{,}0023 & 57{,}5344 \\ 0 & -0{,}01744 & 63{,}8319 & -0{,}0071 & 57{,}4388 \\ 0 & -1.3349\times10^{+04} & -0.5096 & 20{,}7862 & -4.3464\times10^{+04} \\ 0 & -1.3327\times10^{+04} & -0.8152 & 63{,}8320 & -4.3392\times10^{+04} \\ 0 & 1.0085\times10^{+07} & -2.7616\times10^{+05} & 30.2658 & 3.2067\times10^{+07} \\ 0 & 1.0068\times10^{+07} & -2.7295\times10^{+05} & 29.6025 & 3.2016\times10^{+07} \end{bmatrix} \tag{7}$$

Using Matlab to compute the rank of (6) and (7), we see that 'Obs1' has a rank of 4 and 'Obs2' a rank of 5. This means that if in the system's output we consider just the x_3 variable to be measured, the open loop system is not observable. And if we consider x_1 and x_3 to be measured in the output, the open loop system is observable.

3.4 Question 4.

The plot of the Bode diagram of the open loop system is presented in figure 2.

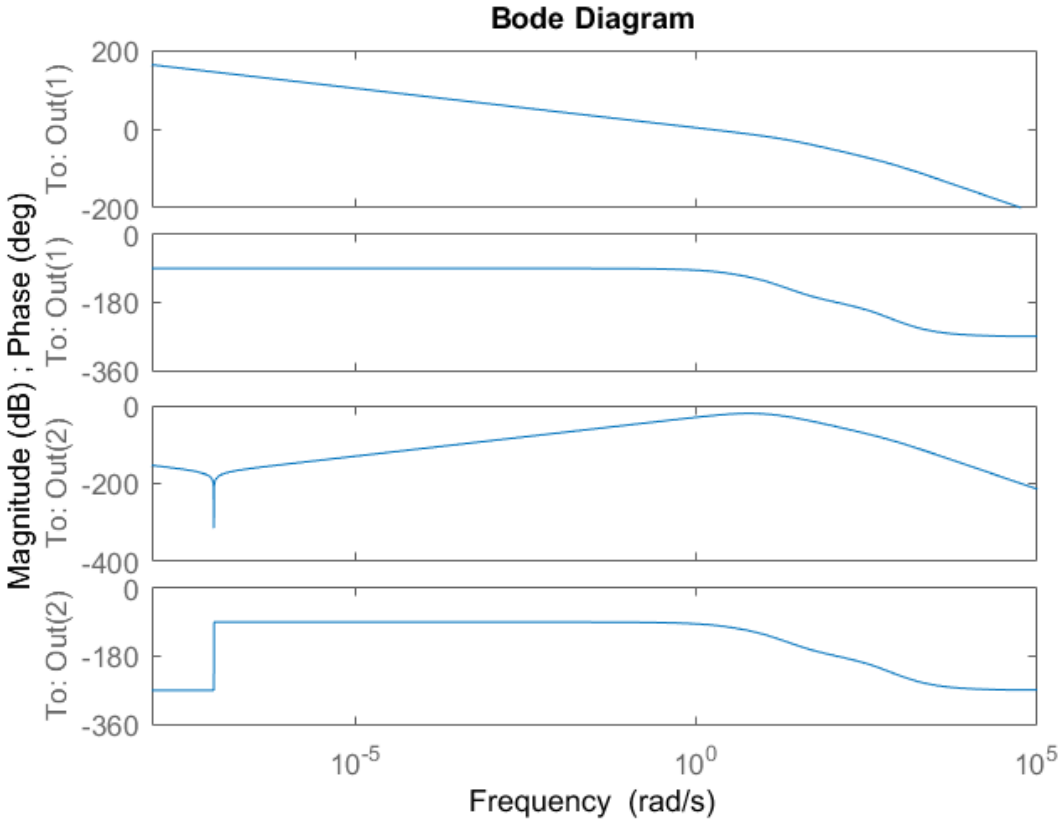
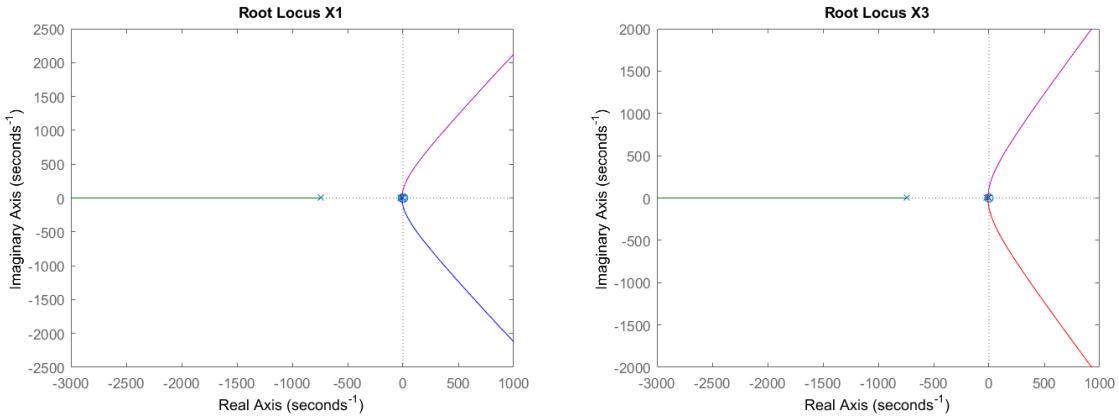


Figure 2: Bode diagram of the open loop system

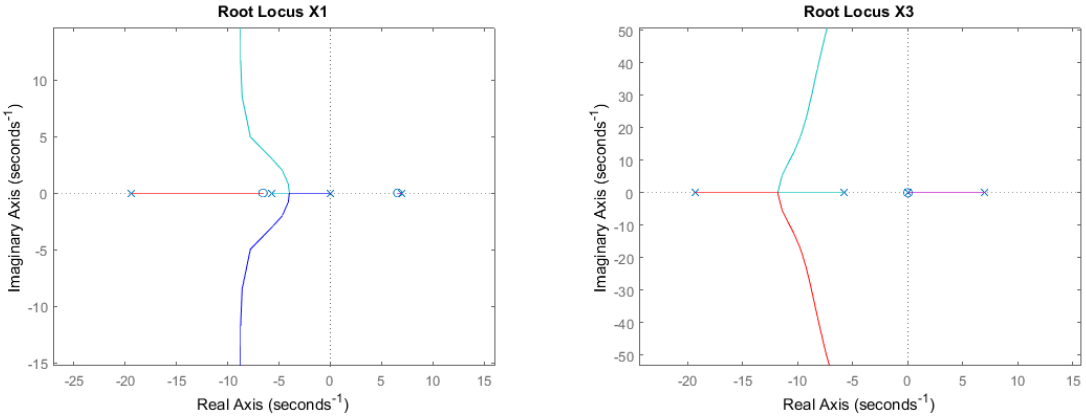
To analyze the bode diagram we resorted to the analysis by root-locus (figure 3).



(a) Root-locus of x_1 (b) Root-locus of x_3

Figure 3: Root-locus

Examining the poles closer to zero we get figure 4.



(a) Root-locus of x_1 (b) Root-locus of x_3

Figure 4: Root-locus

Let's take a closer look on figure 4 (b).

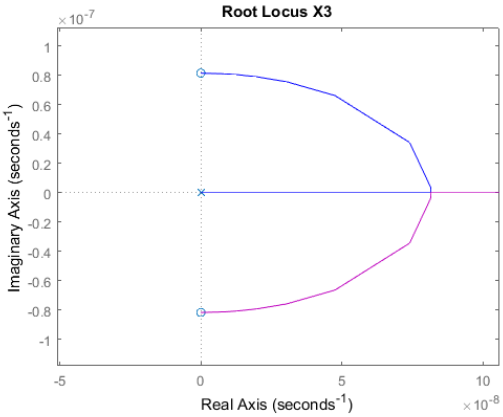


Figure 5: Root-locus

This model is a liberalization of an non-linear system around a non-stable equilibrium point. We know this because in question 1 we saw that there is an Eigenvalue with positive real part.

Analysing the bode diagram, we can see that the if we input a signal with various frequencies, the X_1 output will always decrease. X_3 will have different values but, as the frequency increases, it will decrease as well. These different values can be explained using the root locus.

3.5 Question 5.

In this subsection our aim is to find the vector of gains of the controller. In this phase of the design process it is assumed that all the components of the state are available. The state feedback controller is defined by

$$u(t) = -Kx(t) \tag{8}$$

When coupled with the plant model (1), the control law (8) yields the closed-loop model

$$\frac{\partial x}{\partial t} = (A - BK)x \tag{9}$$

If the pair (A, B) is controllable (a condition checked in question 2 above), then, it is always possible to find the vector of controller gains such that the closed dynamics $A - BK$ has its eigenvalues at the specified eigenvalues, wherever they might be. Therefore, to design the controller, one approach might be to specify the closed-loop eigenvalues and then compute. However, in this work, we follow a different (although linked) path, in which the controller is optimized in a systematic way. Thus, we compute such as to minimize the quadratic cost

$$J = \int_0^\infty (x^T Q_r x + R_r u^2) dt \tag{10}$$

The solution to the problem of minimizing defined by (10) amounts to finding the positive definite matrix that verifies the Algebraic Riccati equation (ARE)

$$A^T P + PA^T - PBR_r^{-1} B^T P + Q_r = 0, \tag{11}$$

and then computing from

$$K = R_r^{-1} BP \tag{12}$$

These computations can be readily made in MATLAB using the *lqr* function to compute the state feedback vector of gains.

Trying different values for Q (a positive semidefinite matrix) shaped as

$$Q = rand(5,5)' \times rand(5,5),$$

and R (a positive scalar) with values

$$R = 10 : 10 : 100,$$

resulted in

$$Kr = \left[-0{,}1440 \;\; -1{,}4588 \;\; 14{,}3684 \;\; 2{,}0638 \;\; 0{,}0502. \right]. \tag{13}$$

To compute the closed loop poles that result from a given design, figure 6, we can compute the eigenvalues of $A - BK$ through the *eig* function in MATLAB.

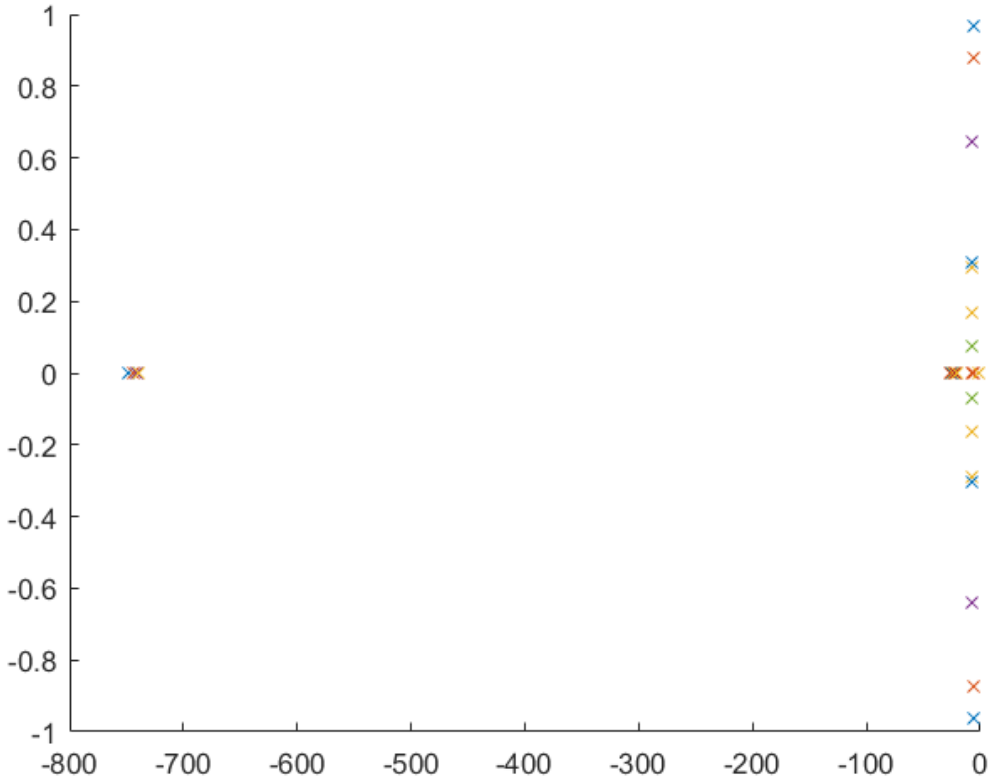


Figure 6: Closed loop poles

3.6 Question 6.

Using the parameters expressed in figure 7 we test the state feedback controller by running the 'statefdbk' SIMULINK block diagram.

```
Qr = diag([10,0,1,0,0]); %Weight Matrix for x in the integral
Rr = 1; %Weight for the input variable
K = lqr(A, B, Qr, Rr); %Calculate feedback gain

% Simulate controller
x0 = [0 0 -0.1745 0 0]';
D = [0 0 0 0 0]';
C = eye(5,5);
T = 2; % Time duration of the simulation
sim('statefdbk',T);
```

Figure 7: System's simulation conditions

It is remarked that, since the reference is zero, the excitation must come from the initial conditions that must be non-zero. The results are represented in figure 8.

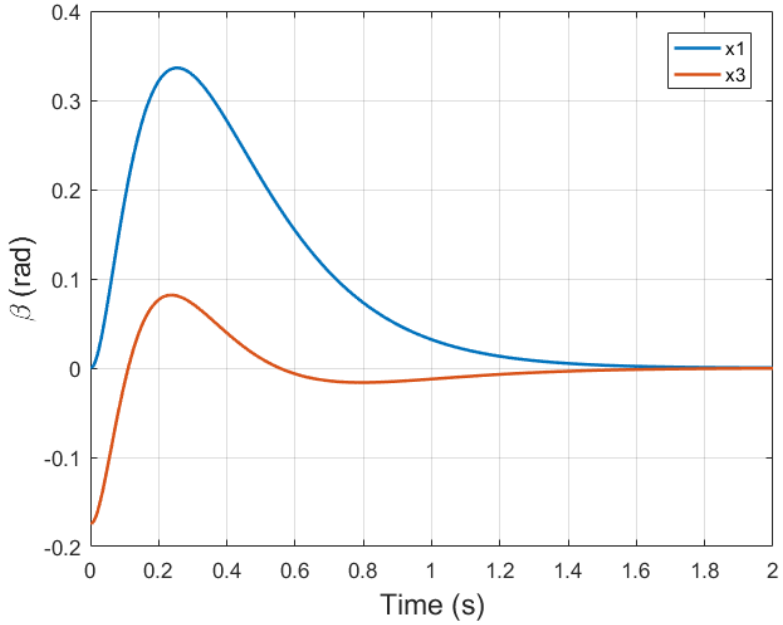


Figure 8: Testing the controller gains

3.7 Question 7.

In this subsection our aim is to find the vector of gains of the observer (state estimator). The observer is a piece of software that builds an estimate $\hat{x}$ of the state x. It consists of a replica of the plant model excited by the same input u, plus the difference between what one expects the plant output y to be (given by $C\hat{x}$) and the actually observed output, y. In mathematical terms, the observer is defined by the differential equation

$$\dot{\hat{x}} = A\hat{x} + Bu + L(y - C\hat{x}) \tag{14}$$

The observer adds poles to the closed-loop. These poles are the eigenvalues of $A - LC$.

The vector of observer gains is designed such that the estimation error converges to zero. A way to do this is to employ the Kalman filter, for which the gain is computed using the MATLAB function *lqe*.

Using parameters from figure 9 we get (15).

```
G = eye(size(A)); %Gain of the process noise
Qe = eye(size(A))*10; %Variance of process errors
Re = eye(2); %Variance of measurement errors
L = lqe(A, G, C, Qe, Re); %Calculate estimator gains
```

Figure 9: Testing the controller gains

$$L = \begin{bmatrix} 3{,}2666 & 1{,}1371 \\ 0{,}9820 & 11{,}8725 \\ 1{,}1371 & 14{,}2789 \\ 8{,}0784 & 97{,}5895 \\ -0{,}2990 & -3{,}6109 \end{bmatrix} \tag{15}$$

3.8 Question 8.

When using the observer, the feedback is not from the state (assumed to be unavailable for measure) but from its estimate, according to

$$u(t) = -K\hat{x}(t) \tag{16}$$

Combining the observer equation (14), and the control equation (16), it is possible to write the following state model for the controller

$$\dot{\hat{x}} = \hat{x}(A - BK - LC) + Ly, \tag{17}$$

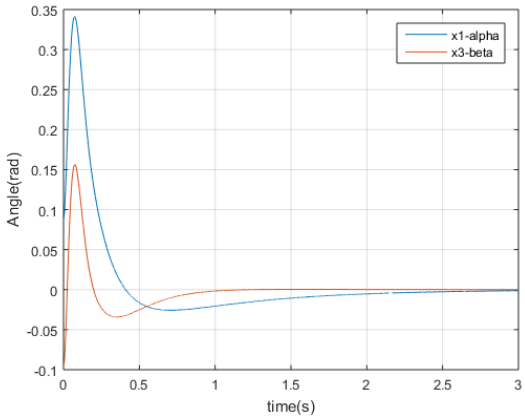
$$u(t) = -K\hat{x}(t). \tag{18}$$

According to (17) and (18), the controller is represented by a state model with input y (the plant output), and output u (the plant input). Therefore, the new dynamics matrix is $A - BK - LC$ (the controller "A" matrix), the input matrix is L (the controller "B" matrix), and the output equation is $-K$ (the "C" matrix of the controller). We may therefore use the state-space block of SIMULINK named 'observer_SIMPLES' in annex to define the controller in a compact way.

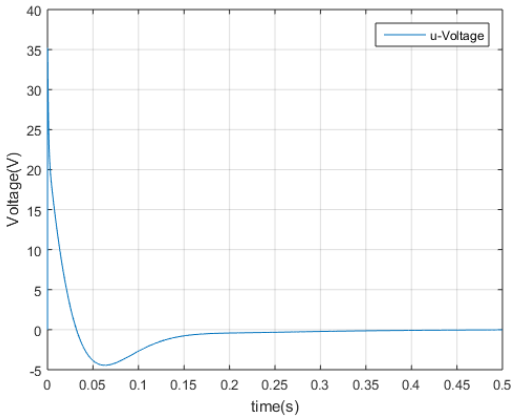
Using the parameters in figure 10 we arrive at figure 11.

```
T = 3;
D = zeros(1,2);

% Initial Conditions
alpha = 0.09;
beta = -0.0945;
x0 = [alpha 0 beta 0 0]';

G = eye(size(A)); %Gain of the process noise
Qe = eye(size(A))*10; %Variance of process errors
Re = eye(2); %Variance of measurement errors

L = lqe(A, G, C, Qe, Re);

Rr = 1;
Qr = diag([10,0,1,0,0]);
K = lqr(A, B, Qr, Rr);
sim('observer_SIMPLES');
```

Figure 10: Testing the controller gains



(a) System variables



(b) Input

Figure 11: Results from interconnecting the plant and controller

4 Laboratory sessions

This section marks the beginning of the laboratory work. After designing a state feedback controller for a linear system described by a state model, including a state observer, and simulating it using SIMULINK we ought to test the design in the real system, replacing the block that simulates the plant (inverted pendulum) by blocks connected to A/D and D/A converters that interconnect SIMULINK computations with the physical system in real time.

This work is all about the creation of a simple cyber-physical system, illustrating a situation of fast prototyping. In this example, the interconnection between the "cyber" and the physical" parts of the plant to achieve this objective (equilibrate the pendulum) is striking as the pendulum would not stand up if the "cyber" part (the controller) is not working properly.

4.1 Simulation of the controller

The goal of our work is to change the feedback gains - K - and the estimator gains - L - of the system to improve our system's performance. In this section we proceed to simulate the controller and the set-up we developed at "home" (points 1 to 8), trying different weights for the cost functions and comparing the results obtained.

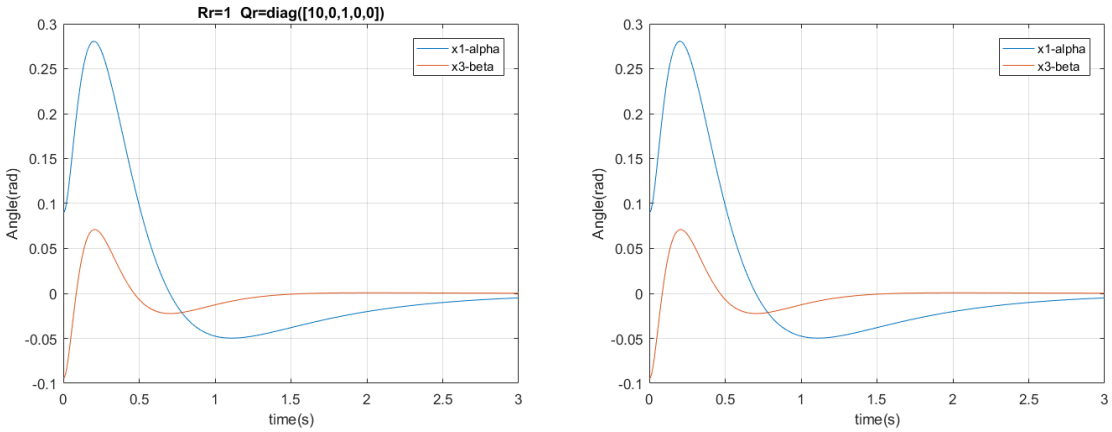
In order to find the "optimal" solution for this problem, a criteria must be set to define what this "optimal" result actually is. We tried to interpret our optimal solution as a 'segway' like behavior. This means that we want the head (β) to be as still and stable as possible while maintaining a more flexible (less restricted but still small movement oriented) base (α). In this section we aim to get the lowest stabilizing time constant on the angles. At 4.2 we make a frequency and amplitude analysis and aim to get the lowest of those parameters.

It is also important to note how we treat Q matrix and R scalar from (10) as "cost" matrices (we can make the analogy to a grocery market where the matrices are the products' prices). The lower their values, the less they cost and the more we will get from them. This is important when analyzing the implications that an increase in these values has on our system (e.g. the more we value α (x_1), the less we will have of it, meaning this state variable will be minimized).

The R scalar acts as a 'control' knob as it works a tuning parameter to balance speed and controllability of the system.

To achieve our goals we follow the next logical steps:

1. Firstly, we checked that changing Q_r and R_r by the same magnitude does not affect the system (the *lqr* remains the same!).



(a) Experiment 1 (b) Both values multiplied by 50

Figure 12: Studying Q-R relationship

The tensions on the conditions of figure 12 are the same as well (figure 13).

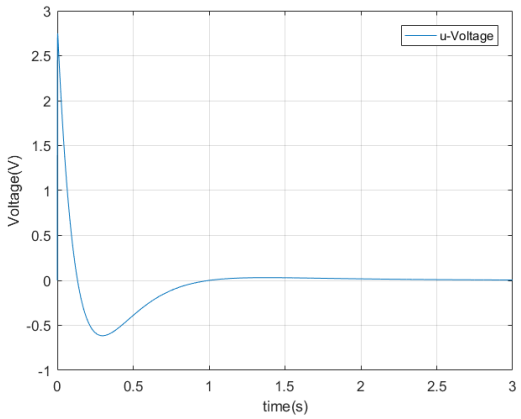
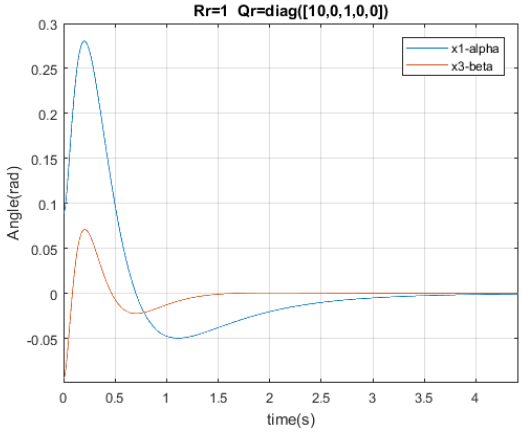
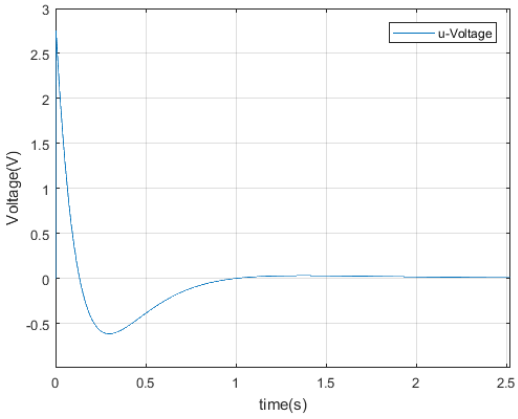


Figure 13: Input tension u

2. Secondly, we try to find the best Q and R matrix values combinations.

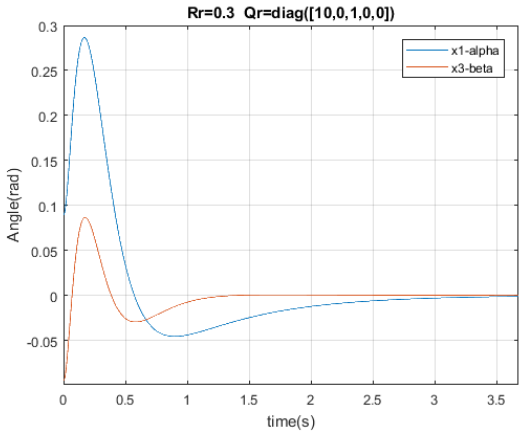


(a) State variables x_1 and x_3

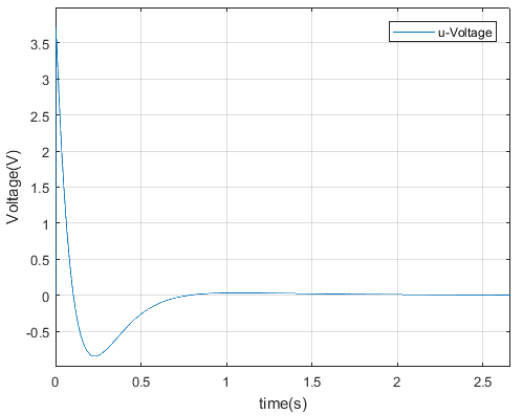


(b) Tension u

Figure 14: R_r fixed at 1

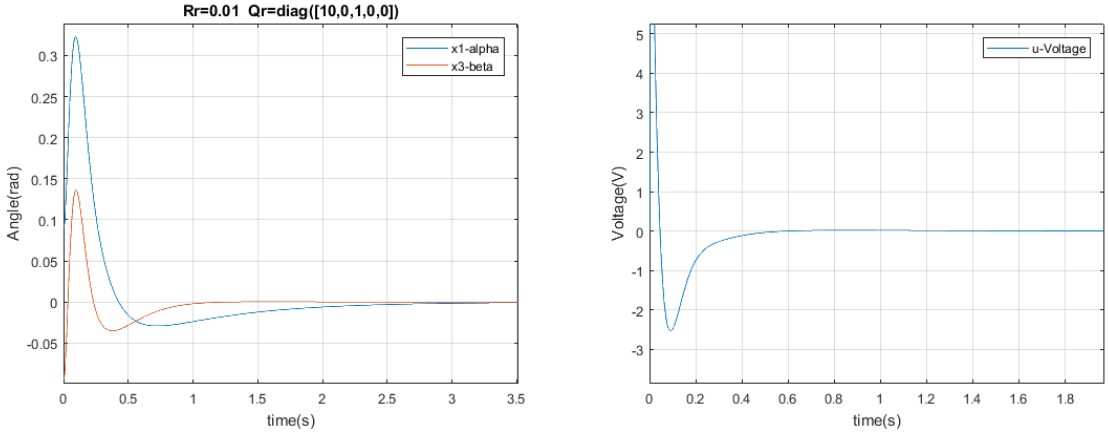


(a) State variables x_1 and x_3



(b) Tension u

Figure 15: R_r fixed at 0.3

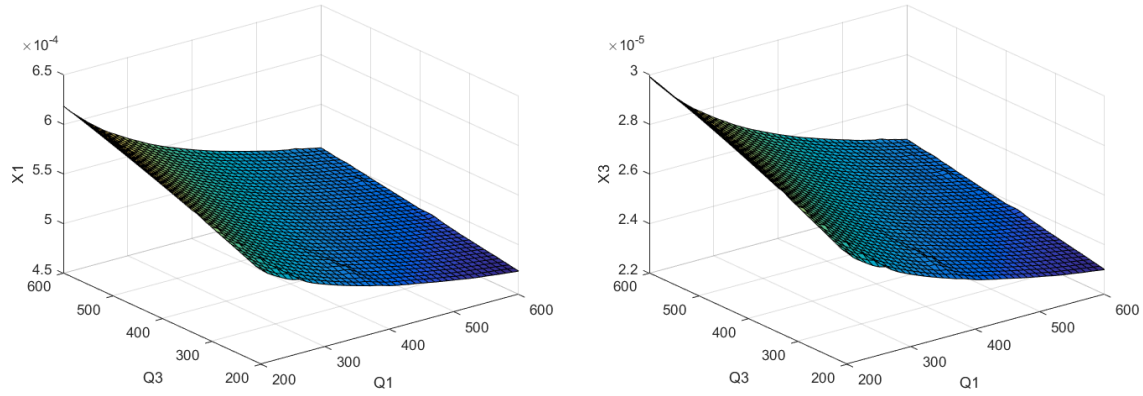


(a) State variables x_1 and x_3 (b) Tension u

Figure 16: R_r fixed at 0.01

As we can see the lowering of the R_r parameter makes the system faster, but more uncontrolled as the tension u tends to move away from the limit established at 5 V.

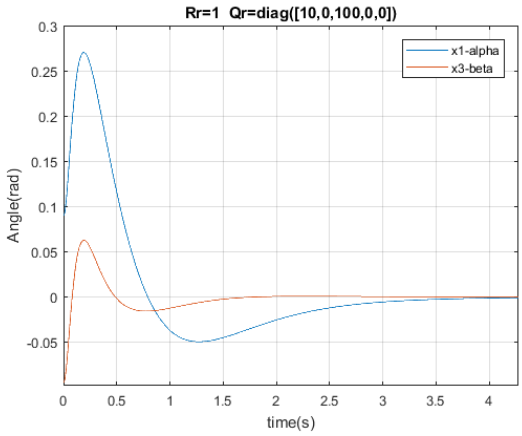
3. Lastly, we change the Q_r matrix and we visualize how the system changes as we provide different "costs" to each of our system variables. We try to control β, followed by α, and attempt to combine the best of both worlds - which we cannot do at a first instance.



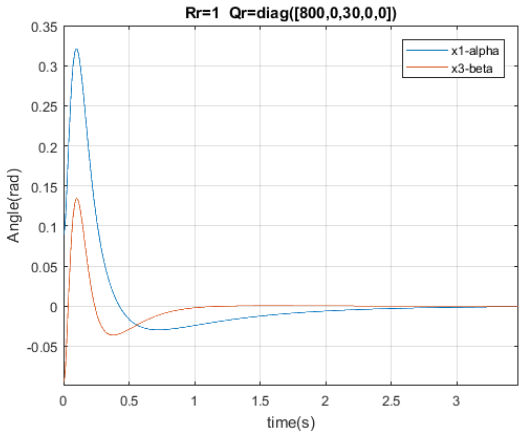
(a) X_1 variation (b) X_3 variation

Figure 17: Examining variation of state variables with Qr matrix

After making this analyze we perform a wide range of tests.



(a) β is more expensive



(b) α is more expensive

Figure 18: Higher price on α and β

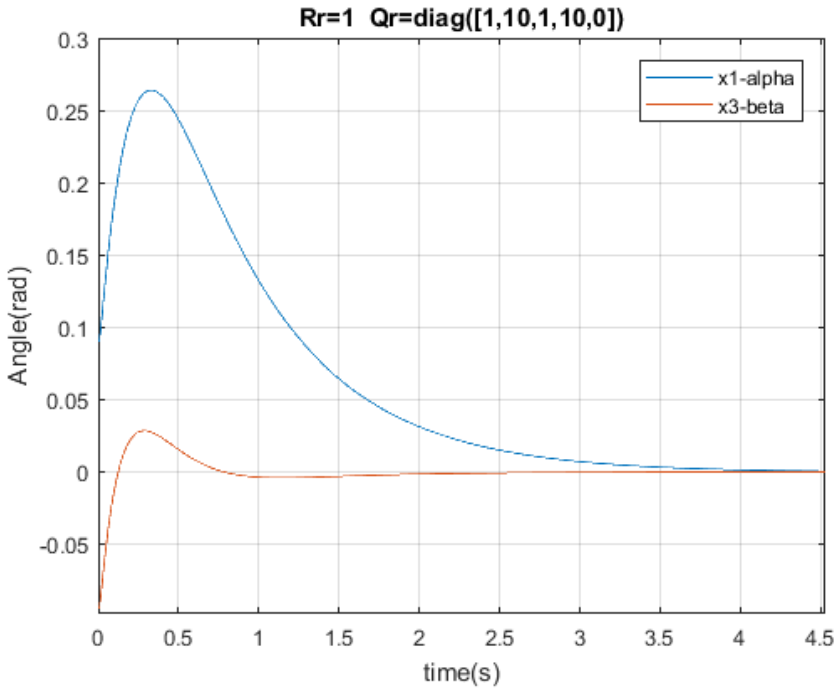


Figure 19: Higher price on x_2 and x_4

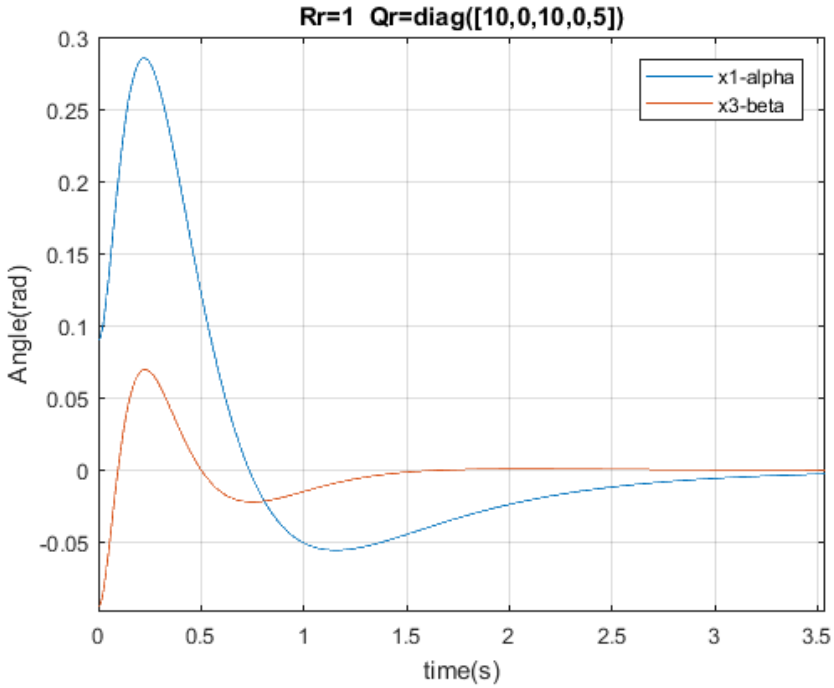


Figure 20: Higher price on x_5

The script 'analysis' is an auxiliary script that allows us to obtain simulations and extensively analyze the implications of changing the Q matrix (figure 17). The parameters were classified as better the more they made the observed angles tend to the reference zero.

4.2 Real system testing

The Rotary Servo Base Unit is equipped with an optical encoder and a potentiometer to measure the output shaft position, and a tachometer to measure the speed of the motor. According to appendix B the potentiometer and tachometer measurement range of the unit is specified at ± 5 V - value that we established as a saturation point in the simulations. Other restrictions that we specify are the lower and upper angle bound (30° and 120° respectively) within which we consider the motor is able to have a linearized behavior.

Figures' 21 to 24 axis are seconds (horizontal) and radians (vertical). α is presented in blue, and β in red.

Giving a low cost to β we see an α that oscillates immensely - only when the pendulum is falling with a big momentum does the motor try to catch it.

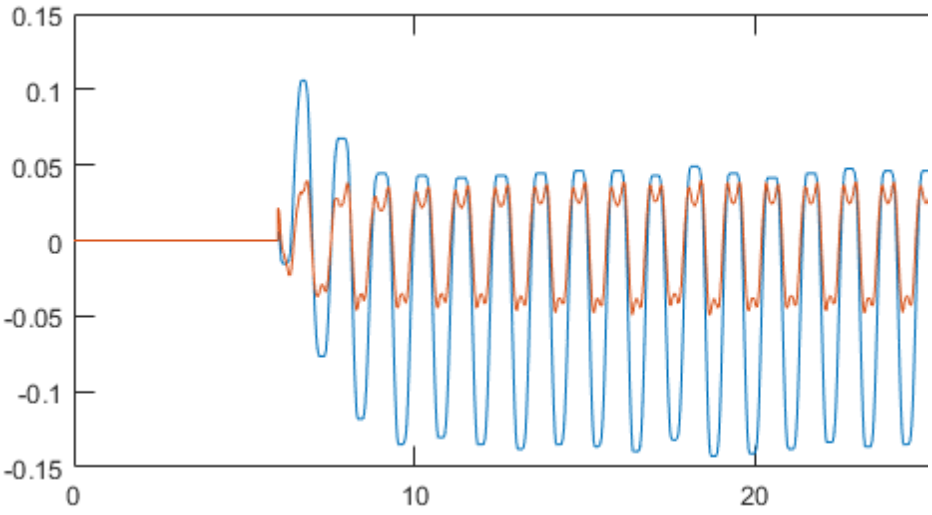


Figure 21: Giving high importance to α

As α assumes larger and larger values, a smaller perturbation in β can kill the motor, and so, we try to stabilize β first.

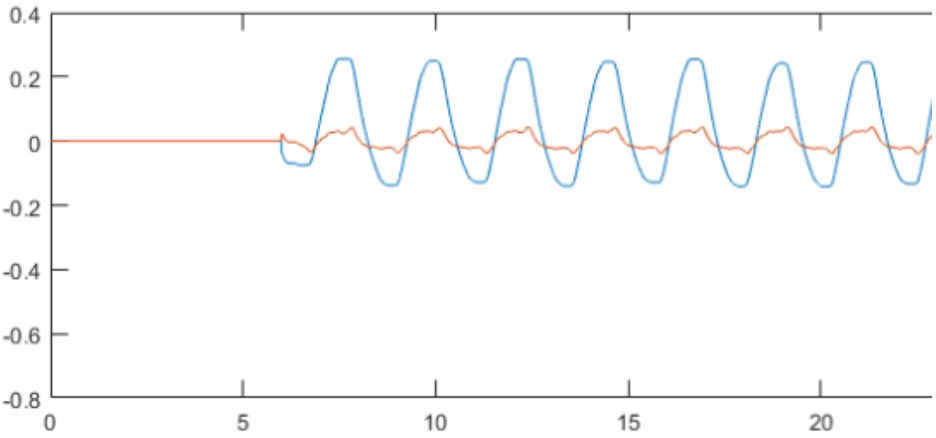


Figure 22: Giving high importance to β

The more importance we give to α, by strucking the pendulum and messing with β, the lesser is the controller's capacity to compensate and stabilize the system.

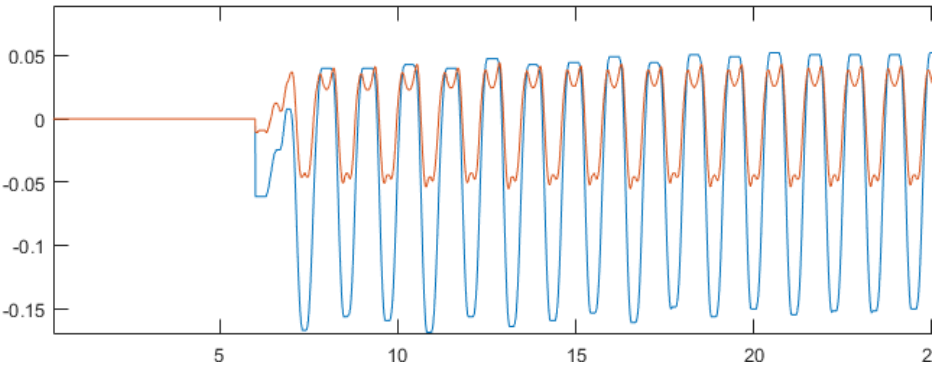


Figure 23: Giving equal importance to α and β

To better analize our results we make a frequency analysis on them. Our goal is to achieve the lowest frequency possible, and amplitude.

One of the problems of the motor is associated with its saturation point at around ± 5 V.

Another problem related to the DC motor used is the fact that inputs whose norm is below a certain value aren't high enough to generate a response, in our case this value stood between 0.3 and 0.5 V (although we know it to be closer to 0.5 V). Dead zone (also know as Deadband or Neutral zone) is the technical name of this second problem we face which makes the DC motor imprecise (something not very desirable!). The dead zone is associated with the input tension U. We could explore other alternatives to surpass the problem related with friction and the tendency to go to a limit cycle, but that tendency would only be attenuated and would never be fully resolved. This problem makes the approximation of α to zero in the simulations performed to never occur in the real system.

This phenomenon can be simulated with SIMULINK (file in annex to this report). The ideal situation would be to ask Quanser the parameters that characterize the non-linearity to build a set of blocks to better approximate our model to the real thing. That proved to be an enormous step and we ended up using the simpler controller.

The motor used has brushes that provide friction and induce the tendency to a limit cycle.

Limit cycles are only seen in nonlinear systems and they are a closed trajectory in the state space. They correspond to a never changing amplitude and frequency.

An alternative to the DC motor would be a Stepper. This brushless DC electric motor divides a full rotation into a number of equal steps. The motor's position can then be commanded to move and hold at one of these steps without any feedback sensor (an open-loop controller), as long as the motor is carefully sized to the application in respect to torque and speed. But would this absence of brushes really solve our friction problem? The answer is no, it wouldn't. In limit, the stepper has less precision than our DC motor. it's advantage and the reason it is used is his iterative step process (we can provide poles to follow until we achieve the desired behavior).

We changed the errors to influence our results. R_e takes importance from the measurements, the biggest source of uncertainty. We found the best results when we increased the variance associated with process errors, and reduced significantly the variance of measurements errors. These results do not constitute the ones with the lowest frequency but they have the best magnitude-frequency of all the results obtained (-0.05 $<$ | α | $<$ 0.05 rads, -0.05 $<$ | β | $<$ 0.05 rads with 1.1737 Hz)

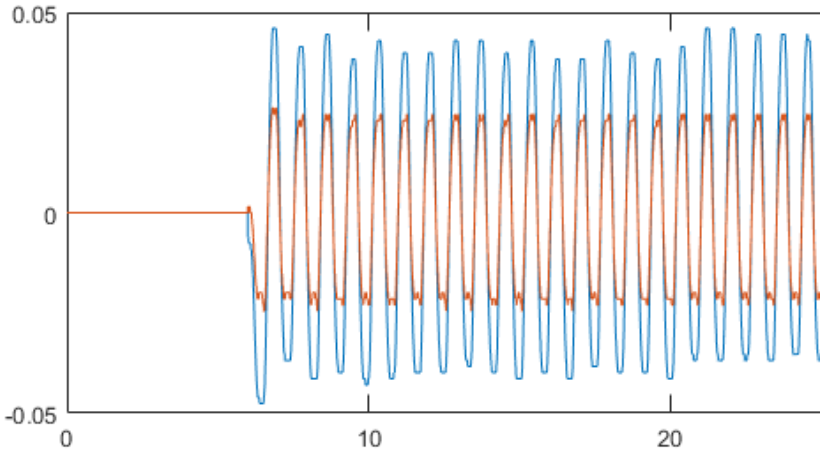
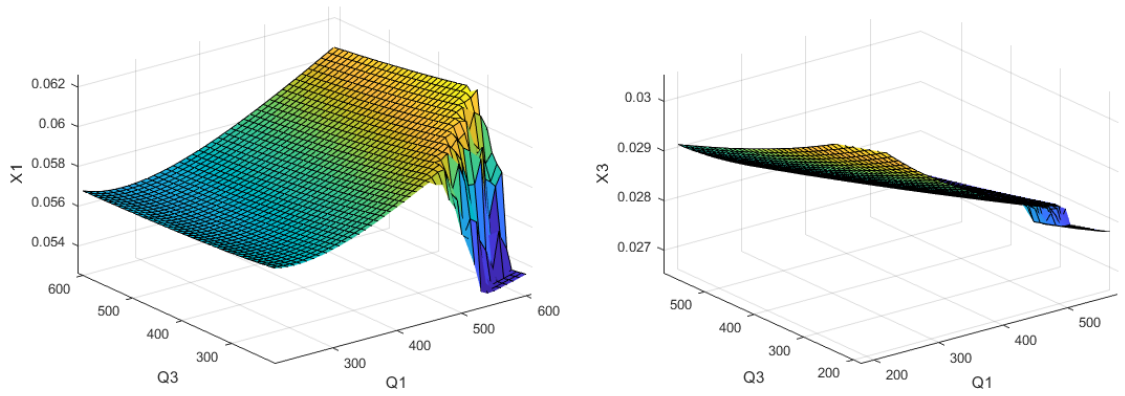


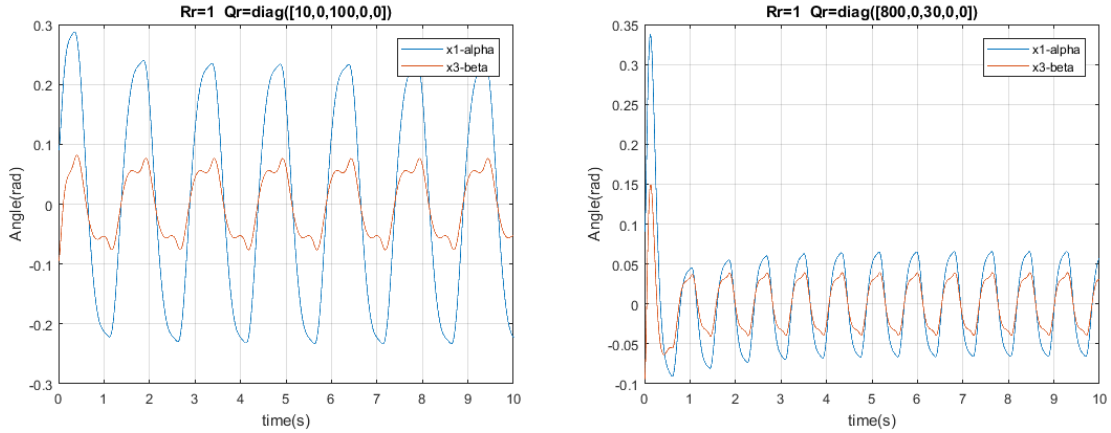
Figure 24: The best result we got

In appendix C we find the simulation blocks that allowed us to simulate the sinusoidal behavior of the state variables' evolution through time. With that block we simulate again the conditions tested in 4.1. The results are shown in figures 25 to 28.



(a) X_1 variation (b) X_3 variation

Figure 25: Variation of state variables with Qr matrix (system with dead zone)



(a) β is more expensive (b) α is more expensive

Figure 26: Higher price on α and β

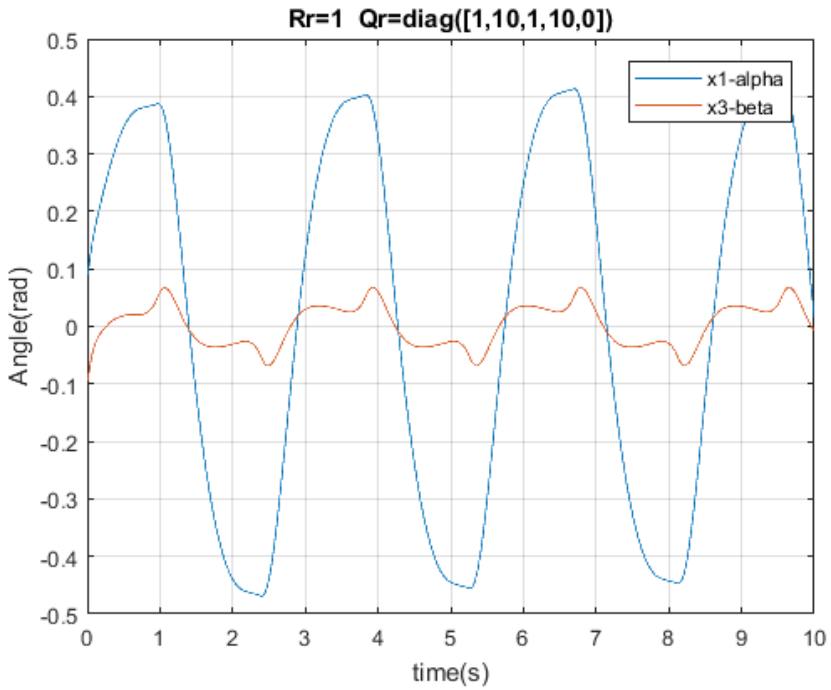


Figure 27: Higher price on x_2 and x_4

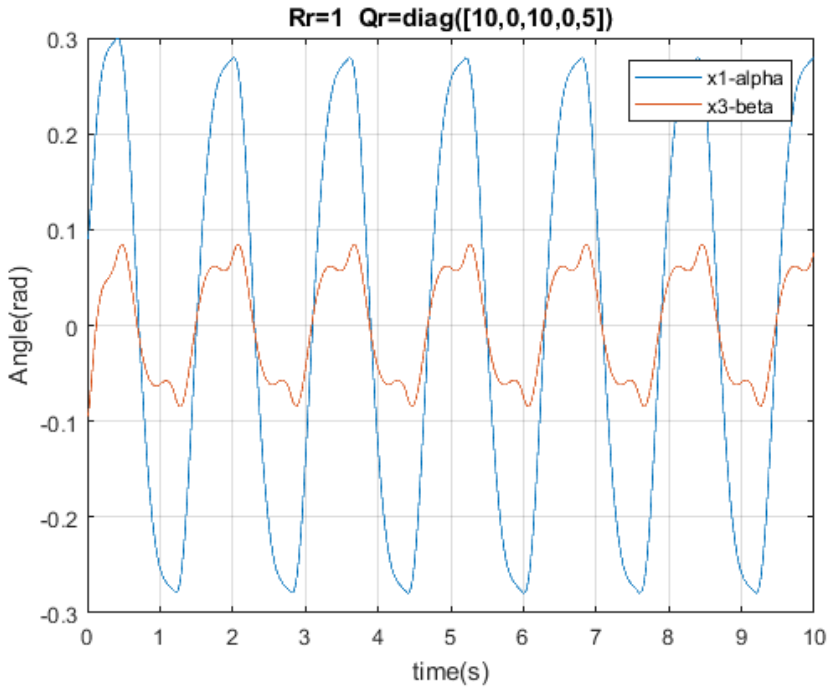


Figure 28: Higher price on x_5

4.3 Improving the design

Lastly, we improve the controller to follow the reference - a solution that would allow us to have the best of both worlds, a good α and β behavior. There are two types of friction: static and dynamic. In our case the most concerning one was the static, since the dynamic is associated only with a loss in velocity.

The static error originates from the gear inside the servo unit and from the motor's dead zone. To adjust and attenuate this error we added some logic to the SIMULINK blocks of the 'PCI_6221_SIMPLES' - that becomes 'PCI_6221_ATENUADO' (appendix D). The logic introduced in the new system model is a subtraction of an integral factor of α to the state space controller outputs.

The components are not without physical faults as the DC motor has millimetric flaws on its gears that make his behavior not 100% desirable. This defects make the α move slightly even after our motor has stopped the arm in the vertical axis. To try and solve this issue we could apply a descriptive function to the diagram blocks we use to run our tests. This would not completely resolve the problem, but attenuate its effect severely, making the best approximation possible.

References

[1] Franklin, Powell and Emami-Naeinil, *Feedback Control of Dynamic Systems*, Pearson/Addison-Wesley (several editions). Chapter 7.

[2] Samuel Balula (2016), *Nonlinear control of an inverted pendulum*, M. Sc. Thesis, IST, Universidade de Lisboa. Available in Fénix, at the course webpage.

[3] Course slides. Available in Fénix, at the course web page.

[4] Ivan Virgala, Peter Frankovský and Mária Kenderová (2013), *Friction Effect Analysis of a DC Motor*. Available at the Science and Education Publishing webpage.

Appendix

A Non-linear model of the pendulum

$$\dot{x} = \begin{bmatrix} \dot{x_1} \\ \dot{x_2} \\ \dot{x_3} \\ \dot{x_4} \\ \dot{x_5} \end{bmatrix} = \begin{bmatrix} x_2 \\ [-J_2[K_{a1}x_2 - K_f x_5 + x_4(L_{cm2}L_{e1}m_2x_4 + 2J_2x_2cos(x_3))sin(x_3)] + L_{cm2}L_{e1}m_2cos(x_3)[-K_{a2}x_4 \\ +(gL_{cm2}m2 + J_2x_2^2cosx_3)sinx_3]/ \\ [L_{cm2}^2L_{e1}^2m_2^2cos^2x_3 + J_2(J_0 + J_2sin^2x_3)] \\ x_4 \\ [K_{a2}x_4 - gL_{cm2}m_2sin(x_3)(J_0 + J_2sin^2x_3) + cosx_3[(J_0J_2x_2^2 + L_{cm2}^2L_{e1}^2m_2^2x_4^2)sin(x_3) \\ -J_2^2x_2^2sin_{x_3}^3 + L_{cm2}L_{e1}m_2(K_{a1}x_2 - K_f x_5 + J_2x_2x_4sin(2x_3))]]/ \\ [L_{cm2}^2L_{e1}^2m_2^2cos_{x_3}^2 - J_2(J_0 + J_2sin^2x_3)] \\ (K_t x_2 - Rx_5 + u)/Lb \end{bmatrix},$$

where

α Angle of the joint between the base and the horizontal arm

β Angle of the joint between the horizontal arm and pendulum

J_0 Moment of inertia at the base joint of the horizontal arm and pendulum

J_2 Moment of inertia at the joint of the pendulum

K_f Torque produced by the motor per current unit

K_t Counter-electromotive force term, coupling the angular speed and current of the motor

K_{a1} Friction coeficient between base and the horizontal arm

K_{a2} Friction coeficient between the horizontal arm and the pendulum

L_{cm1} Distance from axis of rotation to centre of mass of the horizontal arm

L_{cm2} Distance from axis of rotation to centre of mass of the pendulum

L_{e1} Lenght of the horizontal arm

L_{e2} Lenght of the pendulum

m_1 Mass of the horizontal arm

m_2 Mass of the pendulum

B Motor specifications

SYSTEM SPECIFICATIONS ROTARY SERVO BASE UNIT

CURRICULUM TOPICS PROVIDED

Modeling Topics
- First-principles derivation
- Experimental derivation
- Transfer function representation
- Frequency response representation
- Model validation

Control Topics
- PID
- Lead Compensator

FEATURES

- Ten add-on modules are easily interchangeable
- High quality DC motor and gearbox
- High resolution optical encoders to sense position
- Continuous turn potentiometer to sense position
- Tachometer to sense motor speed
- Robust machined aluminum casing with stainless steel gears
- Variable loads and gear ratios
- Optional slip ring for continuous measurement from instrumented modules
- Easy-connect cables and connectors
- Fully compatible with MATLAB®/Simulink® and LabVIEW™
- Fully documented system models and parameters provided for MATLAB®, Simulink, LabVIEW™ and Maple™
- Open architecture design, allowing users to design their own controller

DEVICE SPECIFICATIONS

SPECIFICATION	VALUE	UNITS
Plant Dimensions (L x W x H)	15 x 15 x 18	cm
Plant Weight	1.2	kg
Nominal Voltage	6	V
Motor Maximum Continuous Current (recommended)	1	A
Motor Maximum Speed (recommended)	6000	RPM
Potentiometer Bias Power	±12	V
Potentiometer Measurement Range	±5	V
Tachometer Bias Power	±12	V
Tachometer Measurement Range	±5	V
Tachometer Sensitivity	0.0015	V/RPM
Encoder Resolution	4096	counts/rev.
Gear Ratio (high gear configuration)	70	n/a

Figure 29: Quanser motor specifications

C Observer with dead zone SIMULINK block diagram

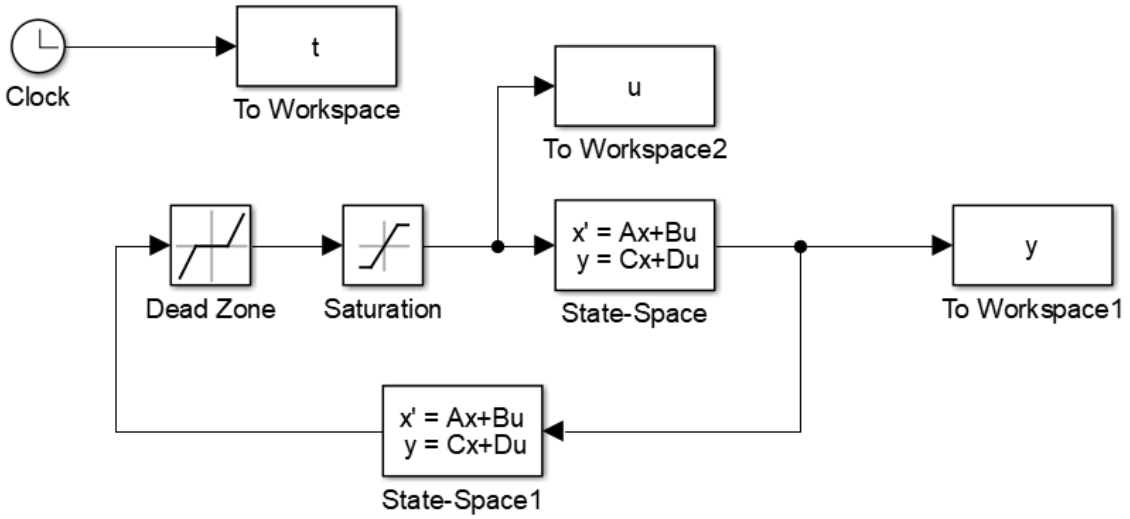


Figure 30: System with dead zone that behaves more like the real world system

D PCI_6221_ATENUADO SIMULINK block diagram

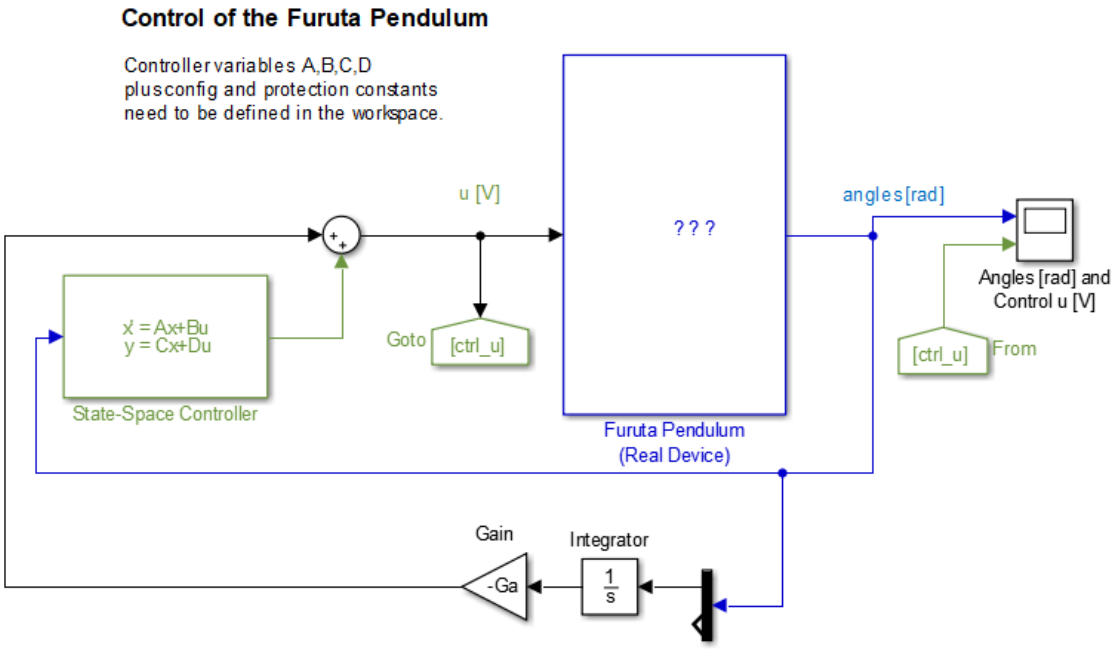


Figure 31: System that attenuates static error