

PONTIFICIA UNIVERSIDAD JAVERIANA



Pontificia Universidad
JAVERIANA
Colombia

UNDERGRADUATE THESIS

Implementation of a Model Based Predictive Control Algorithm in a Digital System

Author:
German D. Montañez P.

Supervisor:
Ph.D. Diego A. Patiño G.

May 2014

ARTÍCULO 23 DE LA RESOLUCIÓN N^o 13 DE JUNIO DE 1946

“La universidad no se hace responsable de los conceptos emitidos por sus alumnos en sus proyectos de grado. Sólo velará porque no se duplique nada contrario al dogma y la moral católica y porque los trabajos no contengan ataques o polémicas puramente personales. Antes bien, que se vea en ellos el anhelo de buscar la verdad y la justicia”.

Contents

Contents	ii
1 Introduction	1
2 Theoretical Framework	2
2.1 Active Suspension Model	2
2.1.1 Model Identification	3
2.2 Control Techniques	4
2.2.1 Interpolation by Piecewise Affine Systems (PWA)	5
2.2.2 Interpolation by Neural Network (NN)	6
2.2.3 Generalized Model Predictive Control (GMPC)	6
2.2.4 Space State Feedback (SSF)	8
3 Implementation	10
3.1 Parameter Identification	10
3.1.1 Specific Parameters for the Methods	10
3.2 Implementation Complexity	11
3.3 The Embedded Program	12
3.4 The Micro-controller	12
3.5 Signal Conditioning	14
4 Results	16
4.1 Simulation Results	16
4.2 Experimental Results	20
5 Conclusions	23
Bibliography	24

1 | Introduction

The model predictive control (MPC) is the second control law more used in the industry after PID controller. The MPC uses a dynamical model to predict the future behavior of the system to be controlled, also the MPC can handle multivariable case and input or output constraints [1][2][3].

In order to implement the MPC in embedded systems is necessary to take into account the processing times of the algorithm. If efforts are focused on solving the optimization problem, the implementation of MPC will have a long execution time [4][5]. For this reason, interpolation techniques are used to compute the control signal faster in comparison to solving the optimization problem on-line [6][7][8].

The main objective of this work is to present the implementation of nonlinear state space feedback by means of a model predictive control interpolation in an digital system. In order to reach the main objective, the following specific objectives were set:

- Select a predictive control algorithm with restrictions for the active suspension (provided by the university) and evaluate their performance offline by simulations.
- Find an algorithm that performs the interpolation of the results of the predictive control based on system states.
- Design and implement the hardware that executes the algorithm that applies the interpolated control law¹.
- Evaluate the performance of the controller by contrasting the results in the simulation and the implemented hardware.

The interpolation by piecewise affine systems (PWA) and neural network interpolation (NN) are two common techniques used for approximating the MPC optimization problem. These two methods offer simplicity to find an approximation for the control signal and require less operations than solving an optimization problem[5][6][8][9].

To compare the results of the interpolation, it was decided to implement the generalized model predictive control on-line (GMPC) as the third control technique in this project. The last method, state space feedback (SSF), was chosen because is one of the most used control methods.

For each control technique, it was developed an embedded algorithm in the same microcontroller with specific features identified during the control design phase. Once the controllers were implemented, it was possible to analyse and compare the performance differences between them (e.g. state stable error, maximum overshoot and setting time). Finally, an MPC was simulated in order to compare the execution time and the precision of each controller.

¹During the implementation phase, it decided to use a development kit and a stage to signal conditioning

2 | Theoretical Framework

2.1 Active Suspension Model

The active suspension emulates the car wheel behavior. The control objective in this system is to reduce the vibration to improve ride comfort and road handling.

Figure 2.1 shows the model of the active suspension. The parameters K_{us} and K_s represent the elasticity spring constant, B_{us} and B_s are constants produced by viscous friction in the dampers, and the masses M_{us} and M_s are the plates in the Active Suspension. The actuator Ac is a motor that generate the force F_c to control the system. The variables x_2 , x_1 and Z_r indicate the reference to the superior plate, middle plate and the floor disturbances, respectively.

The system has two stages, the first stage (K_{us}, B_{us}, M_{us}) emulates the wheel elasticity and the second stage (M_s, K_s, B_s) simulates a normal suspension and adds an actuator (Ac) for the control.

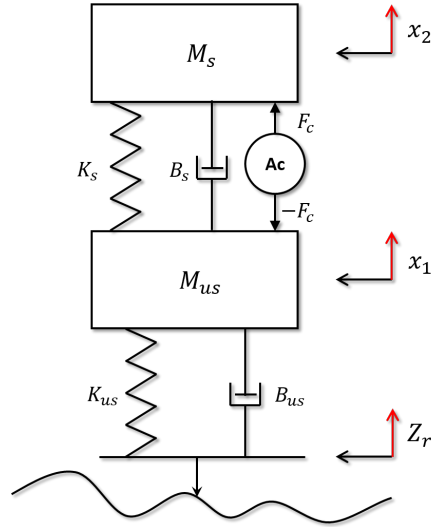


FIGURE 2.1: Active Suspension Model [10].

Regarding the notation in the Figure 2.1 the dynamic model of the active suspension is given by the set of differential equations in (2.1) as follows:

$$\begin{aligned}
 & -M_s \ddot{x}_2 - K_s(x_2 - x_1) \\
 & -B_s(\dot{x}_2 - \dot{x}_1) + F_c = 0 \\
 & -M_{us} \ddot{x}_1 + K_s(x_2 - x_1) + B_s(\dot{x}_2 - \dot{x}_1) \\
 & -F_c - K_{us}(x_1 - Z_r) - B_{us}(\dot{x}_1 - \dot{Z}_r) = 0
 \end{aligned} \tag{2.1}$$

By choosing $Z_1 = x_2 - x_1, Z_2 = \dot{x}_2, Z_3 = x_1 - Z_r, Z_4 = \dot{x}_1$ as state variables the continuous model for the active suspension is:

$$\begin{aligned}\dot{X} &= AX + BU \\ Y &= CX\end{aligned}\tag{2.2}$$

Where,

$$X = \begin{bmatrix} Z_1 \\ Z_2 \\ Z_3 \\ Z_4 \end{bmatrix} \quad U = \begin{bmatrix} F_c \\ \dot{Z}_r \end{bmatrix}\tag{2.3}$$

$$\begin{aligned}A &= \begin{bmatrix} 0 & 1 & 0 & -1 \\ -\frac{K_s}{M_s} & -\frac{B_s}{M_s} & 0 & \frac{B_s}{M_s} \\ 0 & 0 & 0 & 1 \\ \frac{K_s}{M_{us}} & \frac{B_s}{M_{us}} & -\frac{K_{us}}{M_{us}} & -\frac{B_s+B_{us}}{M_{us}} \end{bmatrix} \\ B &= \begin{bmatrix} 0 & 0 \\ \frac{1}{M_s} & 0 \\ 0 & -1 \\ -\frac{1}{M_{us}} & \frac{B_{us}}{M_{us}} \end{bmatrix} \quad C = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}\end{aligned}\tag{2.4}$$

According to this model, only the column in the matrix B that multiplies entrance F_c can control the system. For the control design, it is important to consider a constraint in the actuator force ($-35N < F_c < 35N$) [10].

In this analysis of the model, two outputs were chosen. The position between the floor level and superior mass ($M_s - Z_r$) and the speed of M_s ($Z_2 = \dot{x}_2$). By controlling these states, the acceleration can be controlled indirectly.

2.1.1 Model Identification

The mathematical model of the system is crucial to the MPC. For this reason, the system was characterized by optimization in order to get more precise parameters.

The simulated behavior of the model and the experimental system response for a step signal were compared in order to find the optimal parameters that reduce the error between the model and the real system. By choosing the experimental results as reference, the optimization problem was defined.

The cost function (J) that represent the identification problem is shown in equation (2.5), in this function, the deviation of the initial response is penalized to assure that the optimal signal begins in zero.

$$J = \sum_{t=0}^{t_f} (Y - Y_{ref})^2 + Y(0)^2\tag{2.5}$$

$Y(0)$ represents the deviation of the initial response, Y is the trajectory to be found and Y_{ref} is the experimental reference.

By using an specialized software the optimization problem was solved. The table 2.1 shows the comparison between the manufacturer specification and the optimal parameters identified.

TABLE 2.1: Parameters Values Comparison

Parameter	Specified	Optimal
M_s	2.45 kg	2.45 kg
K_s	900 Nm	952.7 Nm
B_s	$7.5 \text{ N}\frac{\text{m}}{\text{s}}$	$16.8 \text{ N}\frac{\text{m}}{\text{s}}$
M_{us}	1 kg	1 kg
K_{us}	2500 Nm	2447 Nm
B_{us}	$5 \text{ N}\frac{\text{m}}{\text{s}}$	$5 \text{ N}\frac{\text{m}}{\text{s}}$

2.2 Control Techniques

Excepting for the state space feedback control method, the PWA, the NN and the GMPC techniques are based on the model predictive control (MPC).

The general design objective of MPC is to compute a trajectory of a future manipulated variable \hat{u} to optimize the future behaviour of the plant output \hat{y} . The optimization is performed within a limited number of samples called horizons. There are two types of horizons, the predictive horizon and the control horizon. The predictive horizon H_p represents the number of samples in the future to predict the system performance and the control horizon H_c represents the number of control signal samples to be found. In some cases, a delay time is presented it is called the delay horizon H_w [9][11][12][13].

Figure (2.2) shows the general idea of the MPC. The model predictive control is integrated by the System Model and the Optimizer. First, the optimizer interacts with the system model (by sending the control signal $u(k + H_c|k)$), in order to know the system future states ($x(k + H_p|k)$) and the error between the output ($y(k + H_p|k)$) and the reference ($r(k + H_p|k)$); once the optimal output trajectory is computed by the optimizer, the sample $u(k)$ is sent to the plant. Finally, the initial conditions of the system model are actualized ($x(k)$) regarding to the plant response.

The notation $x(k + H_p|k)$ indicates that the signal depends on the conditions at time k , in general.

Equation (2.6) shows the cost function of the optimization problem to be solved in order to find the optimal output in the system. Sub-index $R(i)$ and $Q(i)$ are tuning matrices to penalize variables ($Q(i)$ penalize the output variables and $R(i)$ penalize the magnitude of the incremental variable $\Delta\hat{u}$ and the $r(k + i|k)$ represent the reference trajectory.

$$\begin{aligned} \Phi(k) = & \sum_{i=H_w}^{H_p} \|\hat{y}(k + i|k) - r(k + i|k)\|_{Q(i)}^2 \\ & + \sum_{i=0}^{H_c-1} \|\Delta\hat{u}(k + i|k)\|_{R(i)}^2 \end{aligned} \quad (2.6)$$

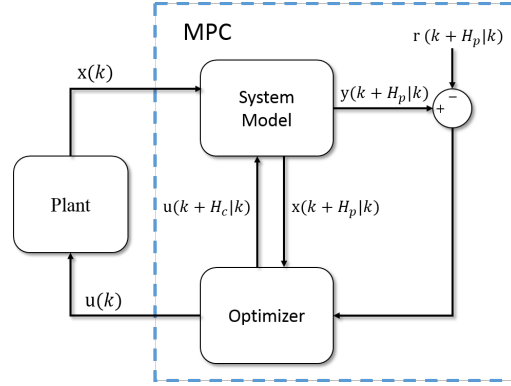


FIGURE 2.2: MPC General Idea.

Although several control signal samples are computed only the first control sample is applied. The prediction assumes that the system has normal behavior all the time but the external disturbances can not be predicted, for this reason, the optimization is implemented in each sampling time and only the first control sample affects the next state space measure. In the literature, this concept is called *receding horizon control* [9][11][13].

For the active suspension, we consider as a restriction only the force produced by the actuator.

Using the MPC toolbox in Matlab to simulate the control performance for the system mathematical model [14], we solve the control problem to get a result reference for the interpolated controls (PWA and NN).

2.2.1 Interpolation by Piecewise Affine Systems (PWA)

By multiparametric programming, the system is divided into operation regions and approximated to different linear piecewise systems for each operation region [6][15].

The physical constraints of the system define the operation regions and the multiparametric program finds polytopes that involve all of these regions. For each polytope found, it is associated with an approximate linear system that simulates the original system performance. Then, the optimization problem is solved for each approximated system and the control law is calculated based on the solution of the MPC problem [6][15].

Using the Multi-Parametric toolbox in Matlab the corresponding matrices for each operation region were calculated. Then, by solving the cost function (equation 2.7) (where the super-index r denotes the active region and the matrices Λ^r , Υ^r and Γ^r correspond to the set of affine system models calculated), the operation region is identified according to the system states.

$$\min_{\Lambda^r, \Upsilon^r, \Gamma^r} J = \min_{\Lambda^r, \Upsilon^r, \Gamma^r} x(k)^T \Lambda^r x(k) + \Upsilon^r x(k) + \Gamma^r \quad (2.7)$$

Once the active region is founded, the control law is executed with the equation (2.8). The matrices Ψ^r and G^r represent the set of control matrices for each region. Then:

$$U(k) = \Psi^r x(k) + G^r \quad (2.8)$$

This process is executed for each sample time and only the first sample of control signal is extracted and applied to the system.

2.2.2 Interpolation by Neural Network (NN)

For LTI systems, the solution of the control signal for the instant $k+1$ for any initial condition and input signal in the instant k is unique. This characteristic of LTI systems allows to approximate the control law in a function that relates the states and the input control signal. By using the Neural Network fitting toolbox in Matlab this function was computed [16].

The simplest type of neural network is the feed-forward topology (Figure 2.3), for this reason, this topology was chosen to generate the control function[16]. In order to determine the number of neurons, several nets were trained with only one hidden layer and $\tanh(x)$ as activation function.

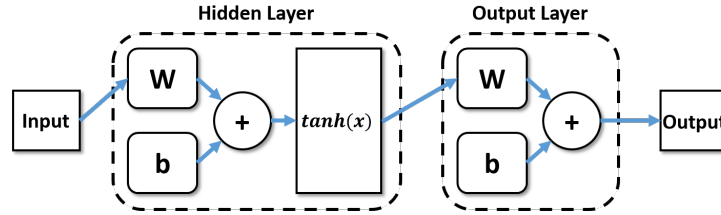


FIGURE 2.3: Feed-Forward Topology.

In order to find this function, it is necessary to choose the input and output training data [16]. For this particular problem, the system states and the floor disturbance were used as the input training data, and the control signal found by simulating the MPC problem was used as the output training data.

The data for training the neural net were extracted of the states and the control signal. It is important to emphasize that, in order to replicate the results obtained for this particular net, it is necessary to use a sinusoidal signal with sweep frequency as input perturbation in the mathematical model to generate the control signal and the state responses.

According to the results of the trained networks, it was found that with less of 27 neurons the net is not able to respond fast enough to track the training data.

2.2.3 Generalized Model Predictive Control (GMPC)

The GMPC is an MPC optimization problem without constraints[11]. The cost function for the optimization problem is the same function for the MPC (Equation (2.6)) but the matrix Q is the identity.

$$\min_{\Delta \hat{u}(k)} J(k) = \min_{\Delta \hat{u}(k)} \sum_{i=0}^{H_p} \|\hat{y}(k+i|k) - r(k+i|k)\|^2 + \sum_{i=0}^{H_c} \|R\Delta \hat{u}(k+i|k)\|^2 \quad (2.9)$$

In order to find the solution of the optimization, it is required recursive form to express the outputs and the states. By choosing as new estates $x(k) = [\Delta x_m(k)^T y(k)]^T$ the system model was rewritten¹.

$$\begin{aligned}
 \underbrace{\begin{bmatrix} \Delta x_m(k+1) \\ y(k+1) \end{bmatrix}}_{x(k+1)} &= \underbrace{\begin{bmatrix} A_m & z(n)^T \\ C_m A_m & 1 \end{bmatrix}}_A \underbrace{\begin{bmatrix} \Delta x_m(k) \\ y(k) \end{bmatrix}}_{x(k)} + \underbrace{\begin{bmatrix} B_m \\ C_m B_m \end{bmatrix}}_B \Delta u(k) \\
 y(k) &= \underbrace{\begin{bmatrix} z(n) & 1 \end{bmatrix}}_C \underbrace{\begin{bmatrix} \Delta x_m(k) \\ y(k) \end{bmatrix}}_{x(k)} \quad z(n) = \underbrace{\begin{bmatrix} 0 & \dots & 0 \end{bmatrix}}_{n\text{-states}}
 \end{aligned} \tag{2.10}$$

Using the augmented space state model, the system response to future states and inputs was written in matrix form as follows:

$$\begin{aligned}
 Y &= Fx(k_i) + \Phi \Delta U, \\
 Y &= \begin{bmatrix} y(k_i + 1|k_i) \\ y(k_i + 1|k_i) \\ \vdots \\ y(k_i + H_p|k_i) \end{bmatrix} \\
 \Delta U &= \begin{bmatrix} \Delta u(k_i) \\ \Delta u(k_i + 1) \\ \vdots \\ \Delta u(k_i + H_c - 1) \end{bmatrix} \quad F = \begin{bmatrix} CA \\ CA^2 \\ \vdots \\ CA^{H_p} \end{bmatrix} \\
 \Phi &= \begin{bmatrix} CB & 0 & \dots & 0 \\ CAB & CB & \dots & 0 \\ \vdots & \vdots & \dots & \vdots \\ CA^{H_p-1}B & CA^{H_p-2}B & \dots & CA^{H_p-H_c}B \end{bmatrix}
 \end{aligned} \tag{2.11}$$

Replaced the equation (2.11) in the cost function (2.9):

$$\begin{aligned}
 J &= (Y - R_s)^T (Y - R_s) + \Delta U^T R \Delta U \\
 J &= (Fx(k) - R_s)^T (Fx(k) - R_s) \\
 &\quad - 2\Delta U^T \Phi^T (Fx(k) - R_s) \\
 &\quad + \Delta U^T (\Phi^T \Phi + R) \Delta U
 \end{aligned} \tag{2.12}$$

For the Active Suspension, the reference trajectory R_s is always 0 because the control design criterion is to keep the system in the least energy state. A necessary condition to find the minimum of J is $\partial J / \partial \Delta U = 0$ [11]. Applying this, it is obtained:

¹The sub-index m denotes the variables to belong to the original model.

$$\frac{\partial J}{\partial \Delta U} = -2\Phi^T(R_y - Fx(k_i)) + 2(\Phi^T\Phi + R)\Delta U \quad (2.13)$$

$$\frac{\partial J}{\partial \Delta U} = -2\Phi^T(R_y - Fx(k_i)) + 2(\Phi^T\Phi + R)\Delta U$$

$$\Delta U = (\Phi^T\Phi + R)^{-1}\Phi^T(-Fx(k_i)) \quad (2.14)$$

Finally the optimal response is written as:

$$u(k) = u(k-1) - (\Phi^T\Phi + R)^{-1}\Phi^T(Fx(k_i)) \quad (2.15)$$

The second order derivative of J is:

$$\frac{\partial^2 J}{\partial \Delta U^2} = (\Phi^T\Phi + R) = H \quad (2.16)$$

This term is called the hessian in the optimization literature. The optimal solution has the same form that Gauss-Newton (2.17) method, this is a common method to solve optimization problems.

$$u(k) = u(k-1) - H^{-1}(k)\frac{\partial J(k)}{\partial \Delta u} \quad (2.17)$$

2.2.4 Space State Feedback (SSF)

This is the most fundamental form of control for space systems because use the principal action of control: each state is multiplied by a gain to feedback the system (Figure 2.4).

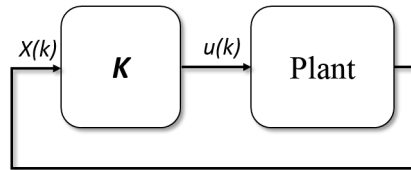


FIGURE 2.4: State Space Feedback.

In order to find the vector of gains (K) a Linear Quadratic Regulator (LQR) was implemented. This method consists on making the transition from the initial state $x(k_0)$ to the final state $x(k) = 0$ using the control function $u(k) = Kx(k)$.

The cost function (2.18) penalizes the deviation of the states and inputs from zero. Q and R are weighting matrices defined in order to penalize the states or the outputs according to design control criteria[17].

$$\min_{u(k)} \sum_{k=0}^{\infty} (x(k)^T Q x(k) + u(k)^T R u(k)) \quad (2.18)$$

Based on the output signals, the values of the matrices Q and R were chosen. In the matrix Q the states Z_1 , Z_2 and Z_3 were penalized and, to choose matrix R it was taking into account that it is desirable to avoid abrupt changes in the control signal.

The optimization problem was solved by a specialized software in order to find the value of K .

3 | Implementation

3.1 Parameter Identification

In order to implement the three control methods based on the MPC, it is necessary to analyze all the parameters that can interfere with the processing time.

The system sample time was the most important parameter to considering. This time is required to develop the embedded control program and it is also necessary to take, properly, the measurements about the system behavior. The sampling time was chosen by experimentation in order to find the largest possible time without affecting the system controllability.

Moreover, the next control sample must be calculated each sampling, consequently the control signal must be computed in less time than the sampling period. By simulation, with large horizons, an acceptable sampling time is: 1ms.

TABLE 3.1: Parameters Values

Parameter	Value
M_s	2.45 kg
K_s	952.7 Nm
B_s	16.8 N $\frac{m}{s}$
M_{us}	1 kg
K_{us}	2447 Nm
B_{us}	5 N $\frac{m}{s}$
T_s	1 ms
H_p	4
H_c	3

The other parameters that could interfere with the processing time are the horizons. In equation (2.11) it is shown that the value of the horizons can increase the size of the matrix ϕ and in order to reduce the processing time is convenient the least number of calculations. For this reason, by experimentation, the smallest values of the horizons were identified, without affecting the control performance.

Table 3.1 shows the value of the parameters previously mentioned and the value of the system constants found by optimization.

3.1.1 Specific Parameters for the Methods

1. PWA: For the MPC interpolated by PWA, the matrices of penalization Q and R found, were:

$$Q = I_{4 \times 4} \quad R = 10 \quad (3.1)$$

The matrix Q was chosen as the identity, because it was desirable that the states had the same importance in the cost function, and the value of R penalizes abrupt changes in the control signal.

Equation (3.2) shows the limits of the states to generate the polytopes of piecewise affine system.

$$\begin{aligned} -0.06m &< Z_1 < 0.06m \\ -10m/s &< Z_2 < 10m/s \\ -0.06m &< Z_3 < 0.06m \\ -10m/s &< Z_4 < 10m/s \\ -35N &< Fc < 35N \end{aligned} \quad (3.2)$$

2. NN: The neural net was trained with 100000 iterations, 30 neurons, $\tanh(x)$ as the activation function, five inputs and one output with 153701 training data. The function trained has the following form:

$$K(Z_1, Z_2, Z_3, Z_4, Z_r) = F_c \quad (3.3)$$

The training error of the net was evaluated with all inputs data training and the result was: 1.4%, and the validating error was evaluated with 10000 data and the result was: 2.5%

3. GMPC: Regarding equation (2.14) in section 2.2.3 the matrix R was chosen in order to penalize the increment in the control signal.

$$R = 0.1 * I_{4 \times 4} \quad (3.4)$$

4. SSF: By experimentation, the values of the matrices Q and R was chosen as:

$$Q = \begin{bmatrix} 450 & 0 & 0 & 0 \\ 0 & 30 & 0 & 0 \\ 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 0.01 \end{bmatrix} \quad R = 0.01 \quad (3.5)$$

Finally the K value is:

$$K = [23.33 \quad 43.10 \quad -100.56 \quad 4.24] \quad (3.6)$$

3.2 Implementation Complexity

The four control techniques were implemented in the same micro-controller. For each technique, it was created a function with the least quantity of instructions as possible. The following considerations led to achieve such characteristics:

1. PWA: By analysing the behavior of control law, the number of regions calculated was reduced discarding some regions that was not used to compute the control signal.

2. NN: It is necessary to use math functions and big matrix operations. In general, the micro-controllers are not able to execute complex mathematical functions. Function $\tanh(x)$ was approximated in a linear region by the Taylor series, and to handle the asymptotical region a saturator was implemented.
3. GMPC: By analyzing the solution of the optimal control problem (equation (2.14)) can be concluded that the ϕ and F matrices are constants. This characteristic allows to reduce instructions in the micro-controller because the matrices can be calculated before programming the GMPC function.
4. SSF: In this case, although it is not possible to reduce the instructions to execute the control method.

3.3 The Embedded Program

The program that find the control signal must:

- Take the system measurements in asynchronous mode and calculate the states variables.
- Execute the control method.
- Send the control signal sample to the actuator.

This process (Figure 3.1) has to be executed in less time that the sample period, otherwise, the control sample could cause a different behavior in the system.

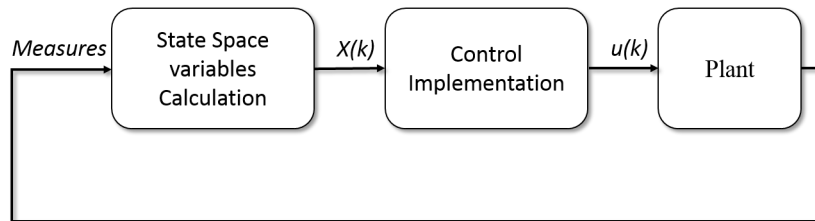


FIGURE 3.1: Implementation Closed Loop.

Figure 3.2 shows the program block diagram. As inputs, the program takes the measures given by the encoders in the system; the "Interrupt Handle" block takes the count about encoders signals. The "Timer" block determines the time to be generated the state variables and the control signal. The "Conversion of State Space Variables" block takes the encoder count and generates the states variables for the control. Finally, the "Control Program" block generates the control signal sample, this signal is sending to the plant.

3.4 The Micro-controller

In order to take the measures, the Active Suspension has three encoder that must be read in quadrature. For this it was used an interruption handle must be implemented. The critical case

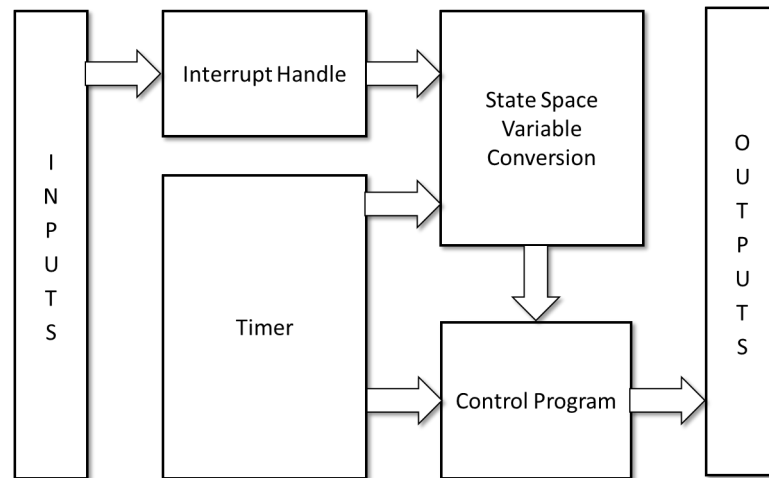


FIGURE 3.2: Program Block Diagram.

happens when the three encoders are move at the same time, in this case an interruption is generated every $50\mu s$ for each encoder.

Considering the maximum number of instructions for the control program based on the simulations, and the interruption rate, the operating frequency must be greater than 75MHz.

Besides these characteristics, the micro-controller must have:

- A timer handle module.
- Four interrupt priority levels.
- Floating point handle.
- At least 10 GPIO pins.

Most of the micro-controller with such operating frequency has the necessary resources to handle the application.

According to the features previously mentioned, the Atmel SAM3N4C¹ 32-bit microcontroller, that operates at a maximum frequency of 100 MHz, was chosen for the control application[18]. The development kit SAM3N-EK by Atmel[19] allows the evaluation of the SAM3N series devices and create embedded applications.

The microcontroller can be programed in C and counts with:

- A 60 to 130 MHz programmable PLL (input from 3.5 to 20 MHz), capable to provide the clock MCK to the processor (Only 100MHz available) and to the peripherals.
- 256 Kbytes of embedded Flash.
- 24 Kbytes of embedded SRAM with dual bank.

¹The microcontroller was chosen from the inventory of the University laboratory. The SAM3N-EK development kit, particularly, has the necessary features to the control application.

- Floating point handle (by software).
- 2x32 bit PIO and 1x15 bit PIO connection interfaces.
- Power regulation.
- PMW, TIMER and UART peripherals.
- 7 interrupt priority levels.
- Multiply and Divide dedicated instruction

Furthermore, the programming tools provides APIs to help the developer.

3.5 Signal Conditioning

To send the control signal to the actuator it is necessary to convert the digital signal in an analog signal, in order to reduce the number of pins used, the control signal was send by the micro-controller in PWM (Pulse pulse width modulation) form. Before the control signal was been applied in the actuator, the PWM signal is converted in analog signal by a pass-low filter and it is adjusted according to the specification about the concordance between the voltage applied and the force in the actuator.

The micro-controller count with a PWM peripheral whit maximum frequency operation of 500kHz, by taking this value as PWM main frequency the filter was designed.

The filter output signal should stabilize as faster as possible to not interfere with the effect of control signal, in order to achieve this, the pass-band frequency was set at 10kHz.

Lastly, the filter designed is show in the figure 3.3 and the main characteristics of the filter designed are shown in table 3.2.

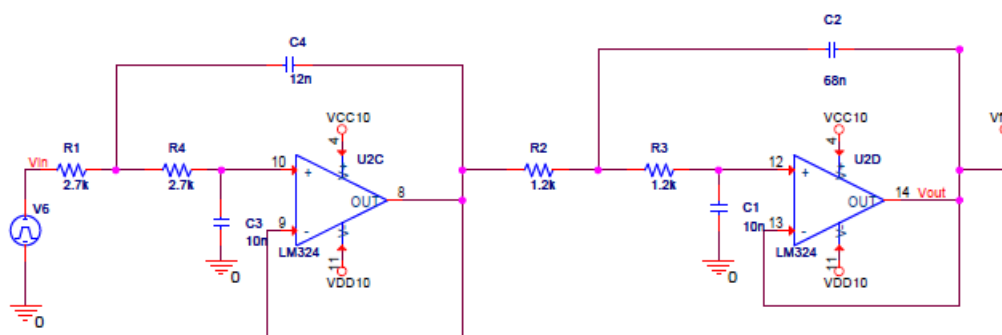


FIGURE 3.3: Pass Low Filter.

After the PWM filter stage, a level signal conditioning was implemented in order to set the 50% of duty cycle as 0V and amplify the signal. First, a differential amplifier was implemented to set the 0V, then, an amplifier stage converts the signal in proportional values of voltage signal to actuator force. The figure 3.4 shows the configuration of the level signal conditioner.

The integrated circuit LM324 was chosen to implement the signal conditioning. The manufacturer suggest this integrated circuit for filters and amplifiers applications [20].

TABLE 3.2: Filter Characteristics

Type	Butterworth filter
Topology	Sallen-key
Passband Frequency	11 kHz
Gain	0dB
Settling time	10 μ s

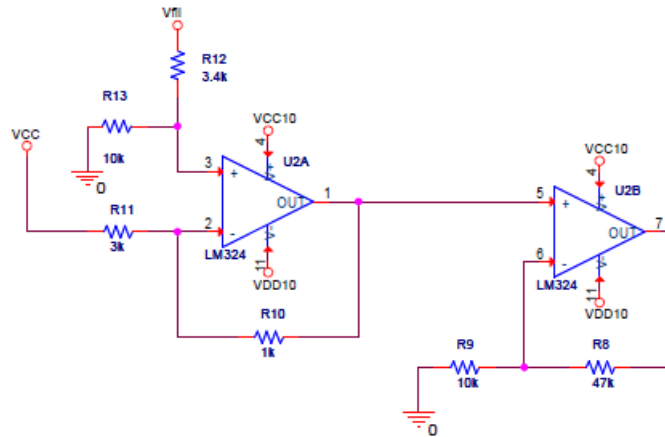


FIGURE 3.4: Level Signal Conditioner.

Regarding to the encoder signals, the micro-controller can to receive the signals directly².

²For more information about the encoders and Quanser modules see [21]

4 | Results

The performance of each controller was tested with two methods: A step response (Figure 4.1 -A and B-) and a sweep frequency response¹(Figure 4.1 -C and D-), both signals can be implemented in the real plant in order to facilitate the comparison between the real and simulated results.

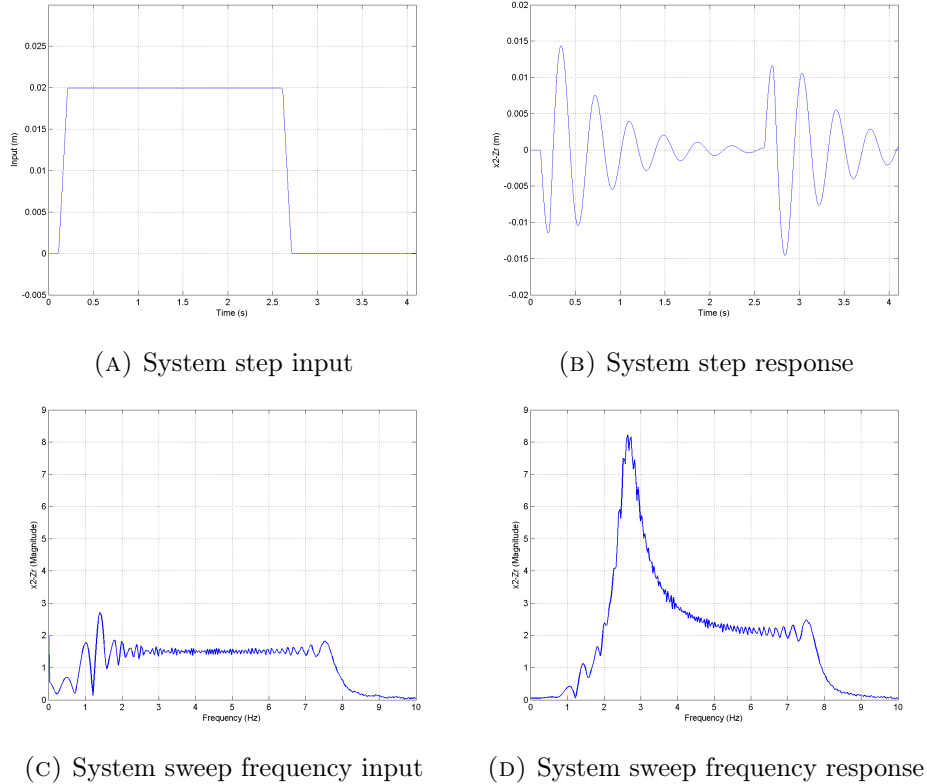


FIGURE 4.1: Plant Response

The control objective is to reduce the acceleration in the superior mass (M_s), and to keep the position compensate the effect of the changes in the floor level.

4.1 Simulation Results

Before choosing and programming the micro-controller the control programs functionality was verified by simulation.

Figures 4.2 to 4.5 show the simulation results of each control technique along with the expected result found by the Matlab MPC Toolbox. In the figures, the blue line represents the simulation results of the implemented code and the black line represents the result generated by the Matlab MPC Toolbox. Also, the control signals are shown for each simulated response.

¹The frequency operation rank is 1Hz to 8Hz

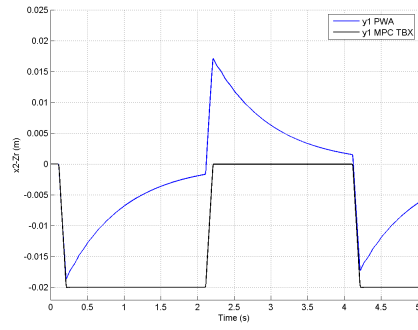
In the case of the PWA and NN methods (Figures 4.2A and 4.3A, respectively), the difference between the simulation results and the response given by Matlab MPC Toolbox (for the step signal), is due to the fact that the system has physical restrictions, in other words, it is possible to achieve more similar simulation results between the two methods and the MPC Toolbox, but, these techniques could not be implemented in practice because this would damage the system.

The differences of the results, for the step response, between the SSF and the MPC methods (Figure 4.4A) occur because the control method theory states that, while the MPC takes the system states and the output signal directly to calculate the control signal, the SSF technique only takes the information about the states.

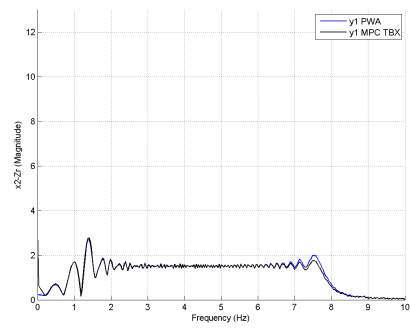
The response to the GMPC simulation is the best approximation to the Matlab solution, in comparison with the other control methods, for both cases (Figure 4.5). In respect of the step response (Figure 4.5A), the behavior of the control allows to keep the position of the superior mass (M_s), and as it can be observed for the sweep frequency response (Figure 4.5B) the GMPC has the same behavior of the MPC control signal reference.

Regarding the frequency response (Figures 4.2B to 4.5B), all the controllers remove the resonance effect and the differences between the control method used and the Matlab solution are not significant.

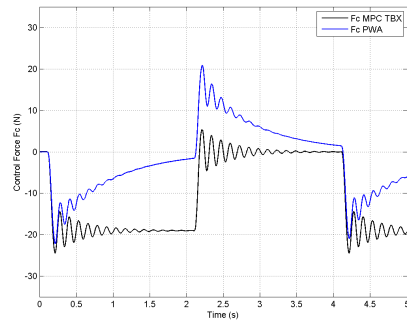
As it can be seen in the results for the system step response, all the control methods reduce the acceleration in the superior mass.



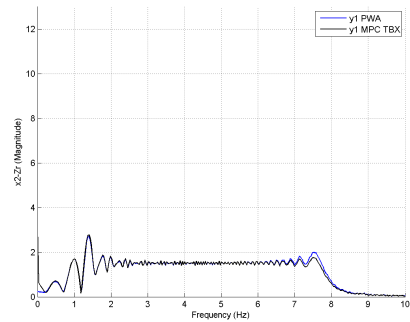
(A) System step response



(B) System sweep response

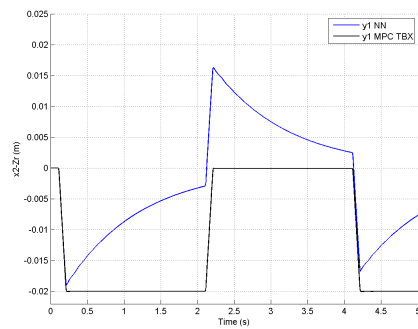


(C) Control Signal Step

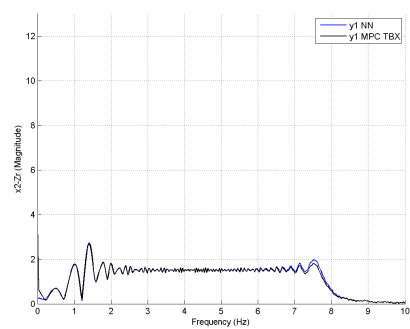


(D) Control Signal Sweep

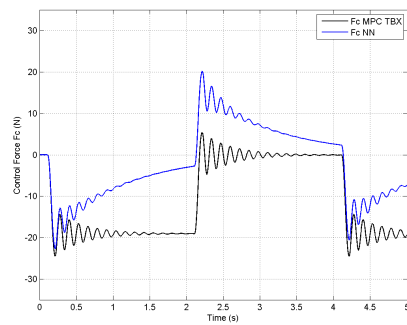
FIGURE 4.2: PWA Simulation



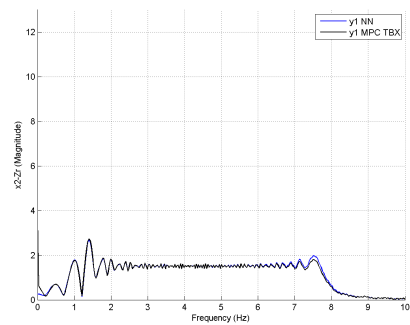
(A) System Step response



(B) System Sweep response

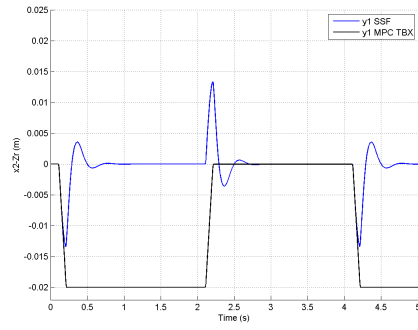


(C) Control Signal Step

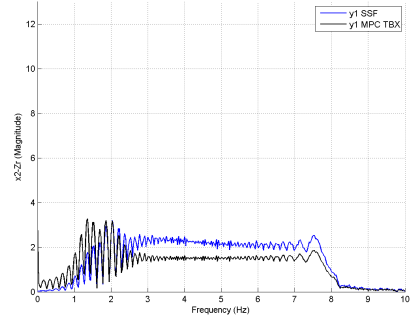


(D) Control Signal Sweep

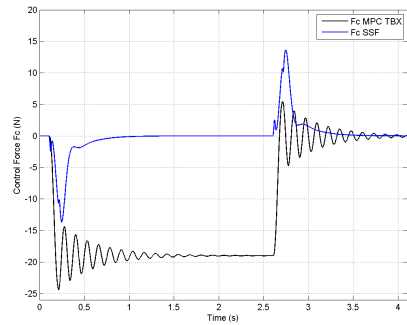
FIGURE 4.3: NN Simulation



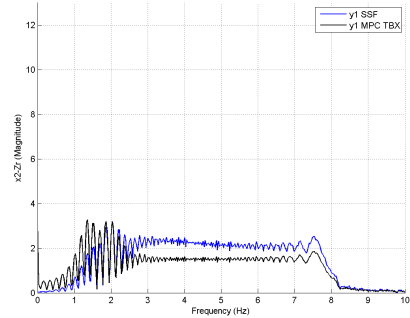
(A) System Step response



(B) System Sweep response

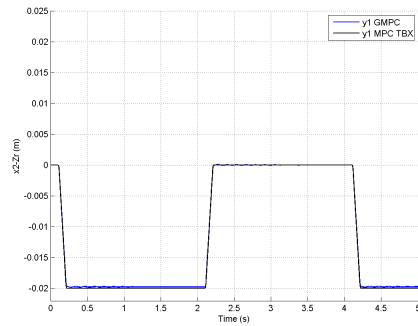


(C) Control Signal Step

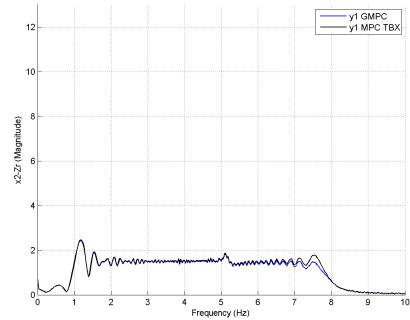


(D) Control Signal Sweep

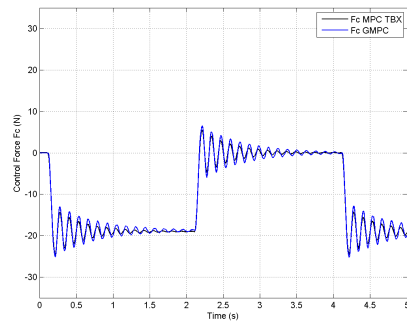
FIGURE 4.4: SSF Simulation



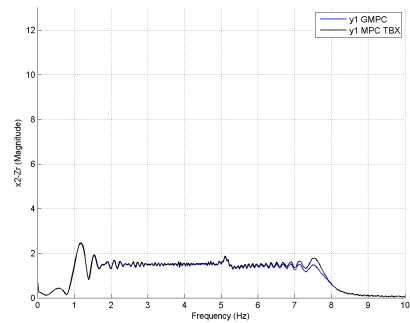
(A) System Step response



(B) System Sweep response



(C) Control Signal Step



(D) Control Signal Sweep

FIGURE 4.5: GMPC Simulation

4.2 Experimental Results

The real results of each control techniques are shown from figures 4.6 to 4.9. The blue line represents the the system physical response obtained by using the embedded program in the micro-controller, and the black line represents the model response obtained by simulation. Also, the control signals computed by the micro-controller are shown for each system response.

All control methods reduce the acceleration of the system and present similar behavior compared with the simulated step response.

The PWA, NN and SSF methods (Figures 4.7, 4.8 and 4.9, respectively) have a similar performance for the step signal response. All of these techniques stabilize the system faster than the GMPC but can not keep the position of the superior mass (M_s). This phenomena is due to the modifications mentioned in the analysis of the simulation results (section 4.1). Also, these implemented controls present an overshoot greater than the GMPC method.

Regarding the frequency response (Figures 4.6B to 4.9B), the differences between the control method used and the simulation results are not significant except for the GMPC sweep frequency response (figure 4.6B). Although the GMPC control changes the resonance frequency of the system, it does not reduce this effect significantly. This control technique requires to apply the actuator force immediately but the actuator motor applies the force gradually, in other words, a delay time is presented between the voltage sample and the actuator effect produced by the motor dynamics of the actuator.

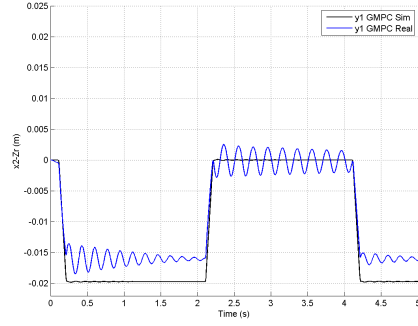
The GMPC method (figure 4.6A) presents a step response with an effect contrary to that of the input, this performance allows to keep the position of the superior mass and remove the changes in the floor level. Although a small oscillation is presented, the control performance is the best approximation of all the compared techniques.

By analysis of the energy for the control signals, the GMPC requires more energy than other techniques because; in order to keep the position of the superior mass it is necessary to keep a force in the actuator. On the other hand the PWA and NN techniques respond to abrupt changes in the floor level and returning gradually the actuator to its inactive state. Regarding the SSF control signal, the control method allows finding the signal with less energy to achieve the control objective.

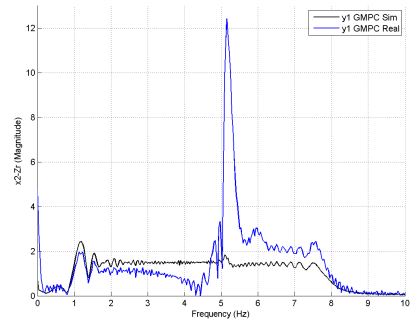
Table 4.1 summarizes the computation time, the maximum overshoot and the setting time (t_s) for all the techniques for the step signal response . For each control technique, the execution time fluctuates due to the encoder interruption.

TABLE 4.1: Control Characteristics Comparison.

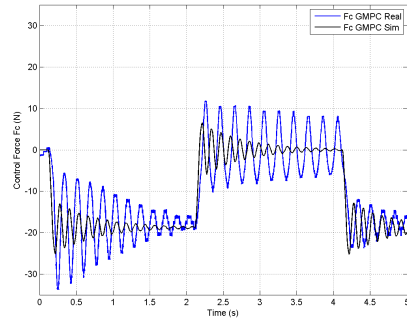
Control Technique	Computation Time	M_p	t_s
PWA	14.4-5.04 μ s	117%	0.42s
NN	92.4-55.3 μ s	116%	0.431s
GMPC	35.7-12.45 μ s	5%	1.23s
SSF	5.3-1.8 μ s	119%	0.48s



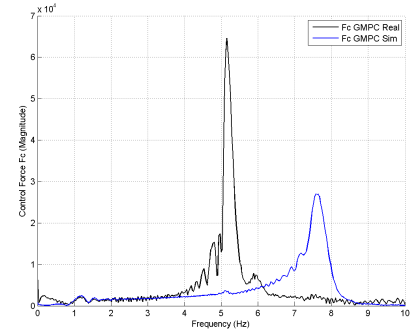
(A) System Step response



(B) System Sweep response

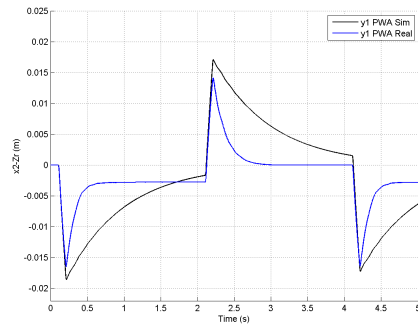


(C) Control Signal Step

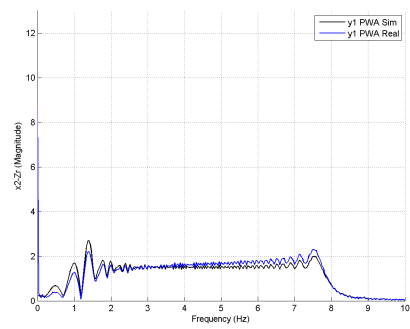


(D) Control Signal Sweep

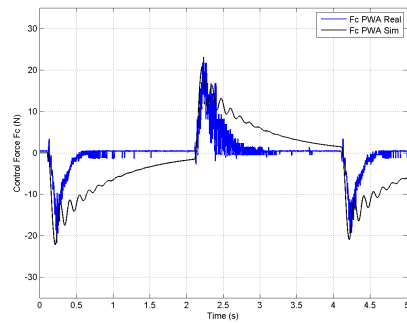
FIGURE 4.6: GMPC real result



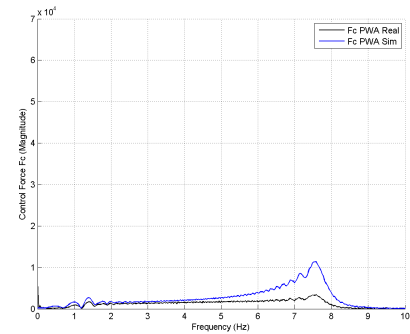
(A) System Step response



(B) System Sweep response

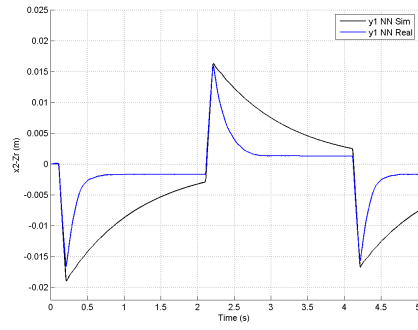


(C) Control Signal Step

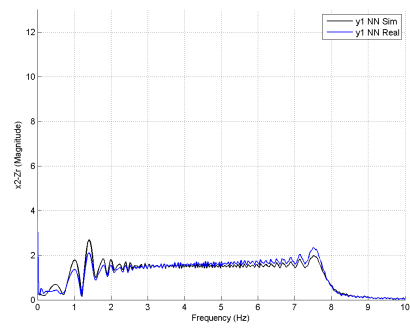


(D) Control Signal Sweep

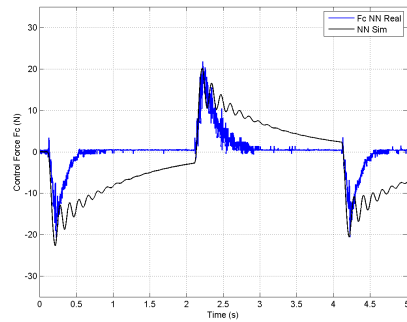
FIGURE 4.7: PWA real result



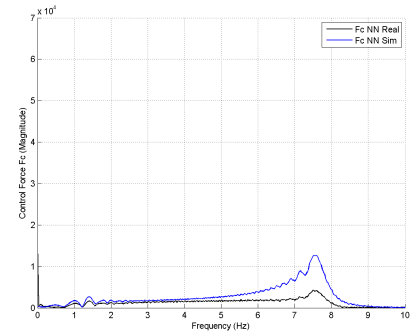
(A) System Step response



(B) System Sweep response

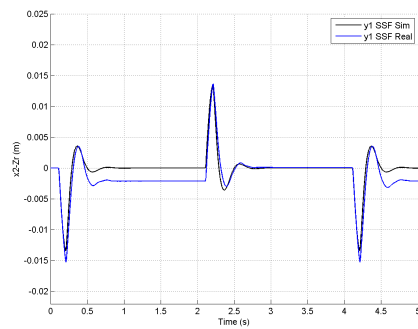


(C) Control Signal Step

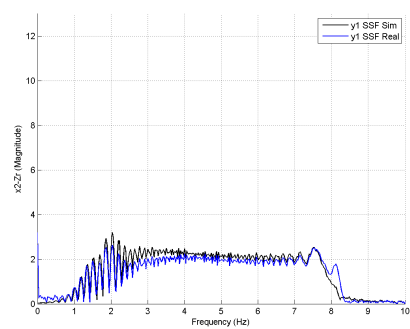


(D) Control Signal Sweep

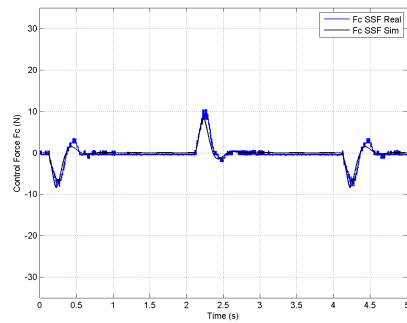
FIGURE 4.8: NN real result



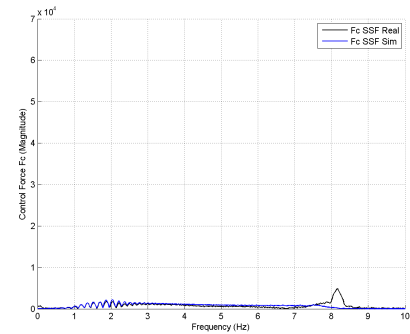
(A) System Step response



(B) System Sweep response



(C) Control Signal Step



(D) Control Signal Sweep

FIGURE 4.9: SSF real result

5 | Conclusions

- All the control methods presented can reduce the acceleration in the superior mass in accordance with the principal control objective.

The GMPC method presented a similar performance than the simulated MPC, but the effect is lower, this happens because the actuator dynamical behavior was not considered in the model.

- Although the parameters of the system are provided by the manufacturer, performing the system identification is recommendable because the continued use and deterioration of the system can change the parameters in the plant. The implementation of robust control techniques is another possibility to design control, because they also take into account the possible changes in the parameters.
- For the simulations it is possible to find better control behaviors than the ones presented, but this results do not consider characteristics that are not included in the model, for example, the actuator force is not applied immediately in the system. Despite the fact that this delay in the actuator force is not considered, the principal control objective was achieved and the simulation provides enough information about the control performance.
- Based on the process of the controls design and the implementation of each one of them, it can be concluded that the influence of the processing time not only depends on the control method, it also depends on the considerations in the control performance and the control objective.

Solving the optimization problem completely in an embedded system becomes an unnecessary task in this case, because it is possible to achieve approximated results with interpolating methods and it is possible to solve the optimization problem by reducing the number of necessary operations to compute the control law.

- In order to improve the control methods, it is feasible to implement two independent modules, the first module will be for the data acquisition and the second module for the control processing.

If the modules are working with parallel processors, they can reduce the execution time and it will be possible to implement a more complex model which includes the actuator behavior. Also, by auto-tuning control methods, the differences between the expected results and experimental results are going to improve.

Bibliography

- [1] S.Joe Qin and Thomas a. Badgwell. A survey of industrial model predictive control technology. *Control Engineering Practice*, 11(7):733–764, July 2003. ISSN 09670661. doi: 10.1016/S0967-0661(02)00186-7. URL <http://linkinghub.elsevier.com/retrieve/pii/S0967066102001867>.
- [2] JS Freudenberg and BH Krogh. Industry needs for embedded control education. *American Control Conference, . . .*, pages 4543–4550, 2005. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1470712.
- [3] Manuel Berenguel, Jose Manuel Guzman, and Sebastian Dormido. Interactive Teaching of Constrained Generalized Predictive Control. (April), 2005.
- [4] Amira Abbes Kheriji, Faouzi Bouani, Mekki Ksouri, and Mohamed Ben Ahmed. A Micro-controller Implementation of Model Predictive Control. pages 68–74, 2011.
- [5] Yang Wang and Stephen Boyd. Fast model predictive control using online optimization. *Control Systems Technology, IEEE . . .*, 18(2):267–278, 2010. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5153127.
- [6] Colin N. Jones and Manfred Morari. Polytopic Approximation of Explicit Model Predictive Controllers. *IEEE Transactions on Automatic Control*, 55(11):2542–2553, November 2010. ISSN 0018-9286. doi: 10.1109/TAC.2010.2047437. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5617688>.
- [7] DQ Mayne, JB Rawlings, CV Rao, and POM Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36, 2000. URL <http://www.sciencedirect.com/science/article/pii/S0005109899002149>.
- [8] Manfred Morari and Jay H. Lee. Model predictive control: past, present and future. *Computers & Chemical Engineering*, 23(4-5):667–682, May 1999. ISSN 00981354. doi: 10.1016/S0098-1354(98)00301-9. URL <http://linkinghub.elsevier.com/retrieve/pii/S0098135498003019>.
- [9] Eduardo F. Camacho and Carlos Bordons. *Model Predictive Control*. Second edi edition, March 2007. ISBN 9781119963981. doi: 10.1002/9781119941446.ch3.
- [10] Advanced Mechatronics Collection:ASP Quanser. Active Suspension Plant Active Suspension System. URL http://www.quanser.com/products/active_suspension.
- [11] Wang Liuping. *Model Predictive Control System Desing and Implementation Using MATLAB*. 2009. ISBN 9781848823303.
- [12] J. A. Rossiter. *Model-Based Predictive Control A Practical Approach*. 2003.
- [13] J. M. Maciejowski. *Predictive Control whit Constraints*. 2000.
- [14] Alberto Bemporad, Manfred Morari, and N Lawrence Ricker. Model Predictive Control Toolbox.

- [15] M Kvasnica, P Grieder, M Baotić, and M Morari. Multi-parametric toolbox (MPT). *Hybrid Syst. Comput. ...*, 2004. URL http://link.springer.com/chapter/10.1007/978-3-540-24743-2_30.
- [16] Mark Hudson Beale, Martin T Hagan, and Howard B Demuth. Neural Network Toolbox. 2014.
- [17] John S. Bay. *Fundamentals of Linear State Space Systems*. First edi edition, 1999.
- [18] Systick Counter, Nested Vector, Interrupt Controller, and Kbytes Rom. *AT91SAM ARM-based*. Atmel.
- [19] *SAM3N-EK Development Board*. Atmel. URL <http://www.atmel.com/Images/doc11080.pdf>.
- [20] *Quadruple Operational Amplifiers LM324*. Texas Instruments. URL <http://www.ti.com/lit/ds/symlink/lm324.pdf>.
- [21] Angelica Hernandez and Nicolas Vargas. DISEÑO E IMPLEMENTACIÓN DE UNA TARJETA DE ADQUISICIÓN DE DATOS INALÁMBRICA PARA EL CONTROL DEL HELICÓPTERO 2-DOF DE QUANSER®, 2013.