Comparison of Model Predictive Control Techniques for Active Suspension

German Montanez, Diego Patino, Diego Mendez
Departamento de Ingeniería Electrónica
Pontificia Universidad Javeriana
Bogotá - Colombia
{gmontanez,patino-d,diego-mendez}@javeriana.edu.co

***Abstract*—This paper presents the develop and analysis of four control techniques implemented in an embedded system for an active suspension. The three techniques are based on model predictive control (MPC): The MPC off-line interpolation by piecewise affine systems (PWA), MPC neural network interpolation (NN), generalized model predictive control on-line (GMPC) and state space feedback (SSF). Finally, it is possible to reduce the necessary time to compute the control law with interpolating methods.**

***Index Terms*—Predictive control, state feedback, neural networks, PWA ,GMPC, embedded control, active suspension.**

I. Introduction

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

In order to implement the MPC in an embedded system 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 [3] [4]. For this reason, interpolation techniques are used to compute the control signal faster in comparison to solving the optimization problem on-line [5].

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 [4] [5] [6].

To compare the results of the interpolation, the generalized model predictive control on-line GMPC was implemented as the third control technique in this paper. The last method, state space feedback (SSF), was also implemented 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.

II. 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 1 shows the active suspension model. 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.

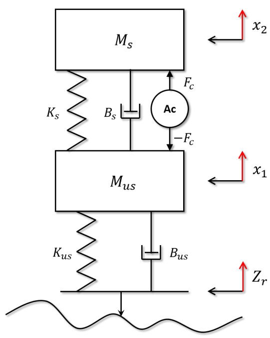

Fig. 1. Active Suspension Model [7].

Regarding the notation in Figure 1, the dynamic model for the active suspension is:

$$\begin{aligned} \dot{X} &= AX + BU \\ Y &= CX \end{aligned} \tag{1}$$

where,

$$X = \begin{bmatrix} Z_1 \\ Z_2 \\ Z_3 \\ Z_4 \end{bmatrix} U = \begin{bmatrix} F_c \\ \dot{Z}_r \end{bmatrix} \tag{2}$$

$$\begin{aligned} A &= \begin{bmatrix} 0 & 1 & 0 & -1 \\ -\frac{K_s}{M_s} & -\frac{B_s}{M_s} & 0 & \frac{B_s}{M_s} \\ 0 & 0 & 0 & 1 \\ \frac{K_s}{M_{us}} & \frac{B_s}{M_{us}} & -\frac{K_{us}}{M_{us}} & -\frac{B_s+B_{us}}{M_{us}} \end{bmatrix} \\ B &= \begin{bmatrix} 0 & 0 \\ \frac{1}{M_s} & 0 \\ 0 & -1 \\ -\frac{1}{M_{us}} & \frac{B_{us}}{M_{us}} \end{bmatrix} C = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \end{aligned} \tag{3}$$

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.

III. Control Techniques

The general design objective of the 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, which is called the delay horizon H_w [6] [8].

Figure 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 states 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.

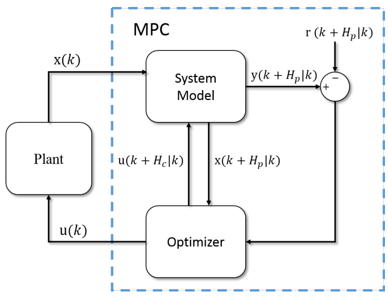

Fig. 2. MPC general idea.

Equation 4 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 ($Q(i)$ penalizes the output variables and $R(i)$ penalizes the magnitude of the incremental variable $\Delta\hat{u}$) and the $r(k + i|k)$ represents the reference trajectory.

$$\min_{\Delta\hat{u}(k)} \Phi(k) = \min_{\Delta\hat{u}(k)} \sum_{i=H_w}^{H_p} \|\hat{y}(k+i|k) - r(k+i|k)\|^2_{Q(i)} + \sum_{i=0}^{H_c-1} \|\Delta\hat{u}(k+i|k)\|^2_{R(i)} \tag{4}$$

Although several control signal samples are computed, only the first control sample is applied. The prediction assumes that the system has a 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* [6].

A. Interpolation by piecewise affine systems (PWA)

The physical constraints of the system define the operational regions and the multiparametric program finds polytopes that involve all of these regions. For each polytope found, an approximate linear system is associated that simulates the original system performance. Then, the optimization problem is solved for each approximated system and the control law is calculated.

Using the Multi-Parametric toolbox in Matlab, the corresponding matrices for each operation region were calculated. Then, by solving the cost function (see equation 5, 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.

$$\min_{\Lambda^r,\Upsilon^r,\Gamma^r} J = \min_{\Lambda^r,\Upsilon^r,\Gamma^r} x(k)^T \Lambda^r x(k) + \Upsilon^r x(k) + \Gamma^r \tag{5}$$

Once the active region is found, the control law is executed using the equation 6. The matrices Ψ^r and G^r represent the set of control matrices for each region and $U(k)$ is the value of the control signal.

$$U(k) = \Psi^r x(k) + G^r \tag{6}$$

B. 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 us 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.

The simplest type of neural network is the feed-forward topology, for this reason, this topology was chosen to generate the control function.

C. Generalized model predictive control (GMPC)

The GMPC is an MPC optimization problem with no constrains [8]. The cost function for the optimization problem is the same function for the MPC (Equation 4) but the matrix Q is the identity.

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 $\partial J/\partial \Delta U = 0$ [8]. Applying this in equation 4, it is obtained:

$$\Delta U = (\Phi^T\Phi + R)^{-1}\Phi^T(-Fx(k)) \tag{7}$$

Finally, the optimal response is written as:

$$u(k) = u(k-1) - (\Phi^T\Phi + R)^{-1}\Phi^T(Fx(k)) \tag{8}$$

D. Space state feedback (SSF)

This is the most fundamental form of control for linear systems because it uses the principal action of control: each state is multiplied by a gain to feedback the system.

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 $u(k) = Kx(k)$ [9].

IV. Implementation

A. Parameter identification

The system sampling time was the most important parameter to consider. This sampling time is required to develop the embedded control program and it is also necessary to take, properly acquire, the measurements about the system's 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 for 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 for both cases is: 1ms.

The other parameters that could interfere with the processing time are the horizons. The horizons can increase the size of the matrix ϕ and in order to reduce the processing time is convenient to reduce the number of calculations. For this reason, by experimentation, the smallest values of the horizons were identified, without affecting the control performance. These values are: $H_p = 4$ and $H_c = 3$.

B. The Embedded Program

The program that finds the control signal must:

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

This process has to be executed in less time than the sampling period, otherwise, the control sample could cause a different behavior in the system.

C. The Micro-controler

The micro-controller must be fast enough to acquire the asynchronous measurements and compute the control signal, and considering the maximum number of instructions for the control program and the interruption rate, the operating frequency must be greater than 75MHz.

Besides these characteristics, the micro-controller must have:

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

Most of the micro-controllers with such an operating frequency have the necessary resources to handle this application.

According with the features previously mentioned, the Atmel SAM3N4C 32-bit microcontroller, that operates at a maximum frequency of 100 MHz, was chosen for the control application. The development kit SAM3N-EK by Atmel [10] allows the evaluation of the SAM3N series devices and create embedded applications.

V. Results

A. Simulation Results

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

Figure 3 shows the simulation results of each control technique along with the expected result found by the Matlab MPC Toolbox. As it can be seen in the system step response, all the control methods reduce the acceleration in the superior mass.

In the case of the PWA and NN methods, 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 between the step response results, the SSF and the MPC methods occur because the control method theory states that, while the MPC takes the system states and the output signals directly and calculate the control signal, the SSF technique only takes the information about the states.

The response of the GMPC simulation is the best approximation to the Matlab solution, in comparison with the other control methods, for both cases. The behavior of the control allows to keep the position of the superior mass (M_s).

B. Experimental Results

The implementation results of each control techniques are shown in Figure 4. All control methods reduce the acceleration of the system and present a similar behavior compared with the simulation results.

The GMPC method presents a step response with an opposite effect 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 appears, the control performance is the best approximation of all the compared techniques.

The PWA, NN and SSF methods have a similar performance. All of these techniques stabilize the system faster than the GMPC but cannot keep the position of the superior mass. Also, these implemented controls present an overshoot greater than that the GMPC method.

By analysing 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

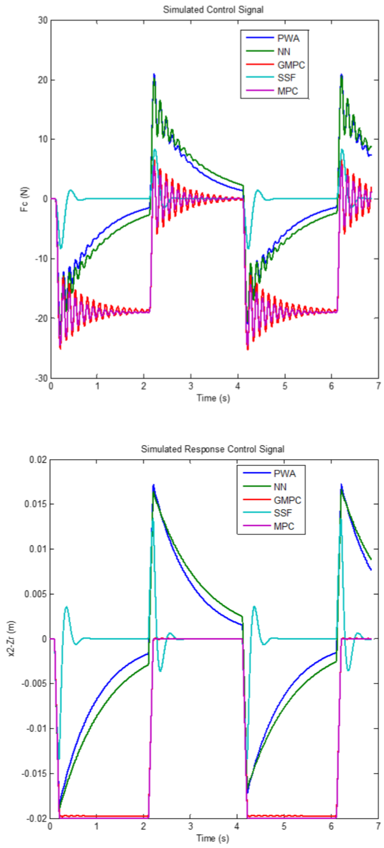

Fig. 3. Simulation results of all the techniques compared with MPC toolbox result.

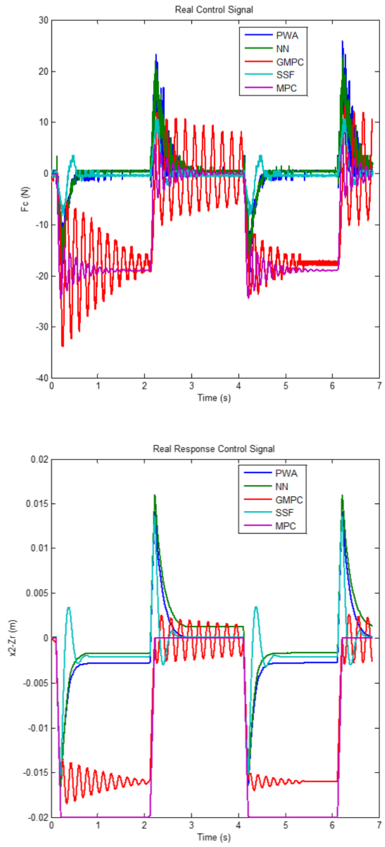

Fig. 4. Experimental results of all the techniques compared with MPC toolbox result.

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 I summarizes the computational time 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 I
CONTROL CHARACTERISTICS COMPARISON.

Control Technique	**Computation Time**	t_s
PWA	14.4-5.04μs	0.42s
NN	92.4-55.3 μs	0.431s
GMPC	35.7-12.45 μs	1.23s
SSF	5.3-1.8 μs	0.48s

VI. CONCLUSIONS

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

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. For this reason, solving the optimization problem completely in an embedded system becomes an unnecessary task in this case.

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.

REFERENCES

[1] S. Qin and T. A. Badgwell, “A survey of industrial model predictive control technology,” *Control Engineering Practice*, vol. 11, no. 7, pp. 733–764, Jul. 2003.
[2] J. Freudenberg and B. Krogh, “Industry needs for embedded control education,” *American Control Conference*, pp. 4543–4550, 2005.
[3] A. A. Kheriji, F. Bouani, M. Ksouri, and M. B. Ahmed, “A Microcontroller Implementation of Model Predictive Control,” pp. 68–74, 2011.
[4] Y. Wang and S. Boyd, “Fast model predictive control using online optimization,” *Control Systems Technology*, vol. 18, no. 2, pp. 267–278, 2010.
[5] C. N. Jones and M. Morari, “Polytopic Approximation of Explicit Model Predictive Controllers,” *IEEE Transactions on Automatic Control*, vol. 55, no. 11, pp. 2542–2553, Nov. 2010.
[6] E. F. Camacho and C. Bordons, *Model Predictive Control*, 2nd ed., Mar. 2007.
[7] A. M. C. Quanser, “Active Suspension Plant Active Suspension System.”
[8] W. Liuping, *Model Predictive Control System Desing and Implementation Using MATLAB*, 2009.
[9] J. S. Bay, *Fundamentals of Linear State Space Systems*, 1st ed., 1999.
[10] Atmel, “SAM3N-EK Development Board.”