

MULTI-MODEL ANALYSIS OF PREDICTIVE ADAPTIVE CRUISE CONTROL

MATEUS MUSSI BRUGNOLLI*, BRUNO AUGUSTO ANGELICO*, ARMANDO ANTÔNIO MARIA LAGANA†

**Depto. de Engenharia de Telecomunicações e Controle
Escola Politécnica da Universidade de São Paulo
São Paulo, São Paulo, Brasil*

*†Depto. de Engenharia de Sistemas Eletrônicos
Escola Politécnica da Universidade de São Paulo
São Paulo, São Paulo, Brasil*

Emails: mateus_mmb@usp.br, angelico@usp.br, lagana@lsi.usp.br

Abstract— Adaptive Cruise Control (ACC) systems for vehicles came as an improvement to cruise systems and there are several approaches for its development. In this paper, it is used a structure of two control loops for simulate the ACC system. The vehicle model was estimated using the system identification theory. An outer loop control manages radar data to adapt into a suitable cruise speed, and an inner loop control aims for the vehicle to reach cruise speed given a desired performance. For the inner loop, it is used two different approaches of model predictive control: a finite horizon prediction control, known as MPC, and an infinite horizon prediction control, known as IHMPC.

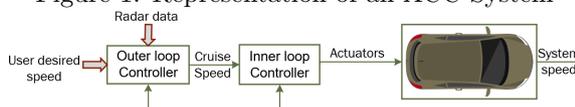
Keywords— ACC systems, System Identification, MPC, IHMPC, Quadratic Programming.

1 Introduction

The Advanced Driver Assistance Systems (ADAS) are technologies developed to help a driver to handle a vehicle more easily and to improve the driving safety. One of these technologies is the Cruise Control (CC), a system that controls the longitudinal speed of the vehicle to achieve a desired cruise speed. In 1995, the Japan improved this technology, creating the Adaptive Cruise Control (ACC) systems for their vehicles (Shakouri et al., 2015).

Usually in ACC researches, the control design is divided into two separate loops, an inner and an outer, as can be seen in Figure 1. For inner loop control, there are two input signals: cruise speed and system (vehicle) speed. This loop outputs are the actuators signals for the system. There are three input signals for outer loop control: user defined speed, vehicle speed and all important radar data, for example distance from the closest vehicle and its relative speed.

Figure 1: Representation of an ACC System



A control theory that has been very popular in ACC researches is the model predictive control theory (MPC). This technique is quite particular considering that its origins came from industrial control processes.

In Shakouri et al. (2015), a nonlinear model was used as a simulated system in order to compare three controllers: a proportional-integral (PI) controller with Gain Scheduling, a Balanced-based adaptive controller (Shakouri et al., 2012) and a nonlinear model predictive controller (NMPC).

This paper aims to analyze the performance of model predictive controllers for a future application in a Polo Sedan, to start the research of ADAS within the vehicle. Such vehicle has a customized Electronic Control Unit (ECU), which offers direct access to the engine using CAN messages. In this work, two different approaches of predictive controllers that use linear models are considered. Firstly, a conventional finite horizon MPC and later an infinite horizon MPC, known as IHMPC. To analyze the robustness of both controllers, this paper suggests simulations with system changing model.

2 System Identification

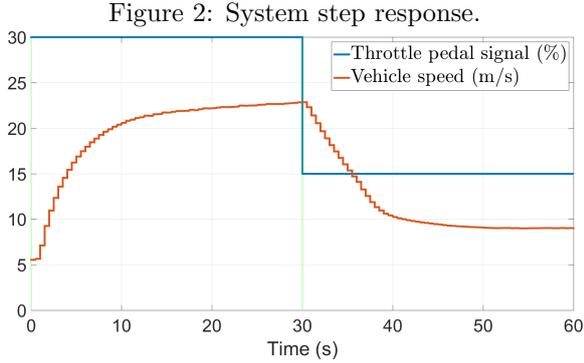
The equations for modeling a translational vehicle are steadfast to obtain (Shakouri et al., 2010). However, the biggest issue for phenomenological modeling is gathering the system parameters, in particular, for the Polo vehicle.

Given this situation, an alternative method for obtaining the vehicle model is using system identification theory (Dias et al., 2015). The selected input signal for the identification was Pseudo-random Binary Sequence, known as PRBS. The signal parameters for generating a PRBS signal are number of samples, frequency bandwidth, minimum and maximum signal limits.

Usually, the frequency bandwidth is determined considering the slowest time constant of the system. This time constant, named as τ , can be identified, for instance, applying a step input and analyzing the system output (Ljung, 1999).

The throttle pedal range varies from 0 (0% or not pushed) to 1 (100% or fully pushed). For an experiment of 60 seconds, in the first 30 seconds a

CAN message of 30% of throttle pedal was sent to the ECU. In the last 30 seconds, this CAN message was changed to 15% of throttle as input. The system step response is presented in Figure 2.



The step response pointed out that the system has a time delay of one time sampling of $\theta = 0.5$ seconds. Then, the system time constant was estimated as approximately $\tau = 5$ seconds. Using a time sampling of $T_s = 0.5$ seconds ($f_s = 2$ Hz) implies that the identification process would collect 10 samples of system constant τ , which is enough for system identification (Ljung, 1999).

A sufficient frequency bandwidth can be calculated with $f_s/20$ Hz to avoid short bit times, which are the slowest interval that PRBS remains constant during all of its sequence. Choosing $f_s = 2$ Hz gives a frequency bandwidth up to 0.1 Hz. The system is nonlinear and for this research the system identification algorithm produces linear models. Additionally, the vehicle speed response changes with each set gear. Given these circumstances, it was necessary to make some study proposals.

Firstly, all system identification and control will be established with the vehicle set in the third gear, since its speed range is wide and sufficient for ACC. The only system input is the throttle pedal, since the electronic brake system is still not available in the vehicle. The system output is the vehicle speed.

During the identification experiments, there are intervals that the PRBS maintains constant, either its maximum or minimum designed value. However, the front wheels of the vehicle are in constantly rolling over the dynamometer and, if the accelerator pedal is not enough, the motor engine stalls. Experimentally, this minimum value was identified as 10%. Moreover, if the accelerator pedal is overly pushed, the engine will exceed 6000 rotations per minute, which is detrimental for the engine. Again experimentally, this maximum value was identified as 35%.

Nevertheless, using a PRBS with its range from 10% to 35% would produce an inefficient linear model, because there are nonlinearities in the system that would strongly affect the identification. This research proposes to split accelerator

range into intervals and each one would produce a linear model. Each model are identified with different input signals, from 10% to 20%, another signal from 20% to 30% and lastly a signal from 30% to 35%.

The identification experiments were done with a dynamometer to assist automotive researches, shown in Figure 3. It has capability of setting a torque load and can be used to simulate different roads, for example higher slope or different terrains. Dynamometer load was used for increasing the number of identification experiments and, consequently, the number of linear models acquired.

Figure 3: Photo of dynamometer



2.1 PRBS specification

Next, in Table 1 it has been specified every PRBS signal used in this system identification. The time sampling chosen was $T_s = 0.5$ seconds, or $f_s = 2$ Hz. Every experiment lasted 4 minutes, resulting in 480 samples. The first 3 minutes, or 360 samples, were used for system identification and the last minute of experiment, 120 samples, were used for system validation. It was used three dynamometer loads: 0% (named as load A), 10% (load B) and 15% (load C).

In order to identify the system with each load, thus reducing correlations within each experiment, each input range created 12 minutes of data, or 1440 samples. Within these 12 minutes, the first 4 minutes were used for Load A, the following 4 minutes were used for Load B and the last 4 minutes for Load C.

Table 1: PRBS data specification

	Bandwidth	Minimum	Maximum
Signal 1	$[0 \ 0.1]$ Hz	10%	20%
Signal 2	$[0 \ 0.1]$ Hz	20%	30%
Signal 3	$[0 \ 0.1]$ Hz	30%	35%

2.2 ARX Models

An autoregressive-moving-average (ARX) model with one input and one output has the following structure (Ljung, 1999):

$$A^{na}(z^{-1})y(t) = B^{nb}(z^{-1})u(t)$$

Given an input u and an output y , an ARX model calculates all coefficients of the polynomials $A^{na}(z^{-1})$ and $B^{nb}(z^{-1})$ in order to reduce the error $e(t)$. The system time delay is defined as nk and the orders of $A^{na}(z^{-1})$ and $B^{nb}(z^{-1})$, respectively, na and nb . Given $T_s = 0.5$ seconds, for all ARX models the time delay order was defined as $nk = 1$ because the time delay for step response was $\theta = 0.5$ seconds. The orders were chosen utilizing the minimum description length (MDL) criteria. The chosen order for $A(z^{-1})$ was $na = 2$ and for $B(z^{-1})$ was $nb = 3$.

From now on, the ARX models will be referenced as shown in Table 2.

Table 2: Models designation

	Load A	Load B	Load C
Signal 1	Model 1A	Model 1B	Model 1C
Signal 2	Model 2A	Model 2B	Model 2C
Signal 3	Model 3A	Model 3B	Model 3C

In Table 3, it is shown all nine ARX models coefficients, changing the minimum and maximum input values and the dynamometer load. Every system has one sample delay ($nk = 1$) with $T_s = 0.5$ seconds.

Table 3: Coefficients of the ARX models

	a_1	a_2	b_1	b_2	b_3
Model 1A	-1.31	0.40	1.78	3.87	-0.78
Model 1B	-0.98	0.15	5.60	1.94	-0.07
Model 1C	-1.20	0.36	2.78	3.03	-0.14
Model 2A	-1.42	0.46	4.70	1.75	-1.97
Model 2B	-1.30	0.36	6.23	0.84	-1.00
Model 2C	-1.33	0.40	4.98	2.53	-1.31
Model 3A	-1.52	0.56	5.06	-1.28	-0.14
Model 3B	-1.33	0.38	7.50	-0.66	-1.23
Model 3C	-1.27	0.33	7.58	-0.10	-1.15

2.3 Models validation

For the model validation, the experiment data selected for validation were used to compare ARX models response with the corresponding experimental vehicle speed. The Fit indexes were calculated for 1, 5, 10, 50 and infinite (simulation) steps ahead and their results are presented in Table 4. Even in the worst situation, the lowest Fit index was 73.87%, that is not a poor value (Ljung, 1999).

3 Control Design

3.1 Outer loop controller

The algorithm frequently used for switching between CC and ACC modes is presented in Table 5 (Shakouri et al., 2015). Given the vehicle speed as v , user defined speed as v_{user} , a security speed as v_{ref} , the measured distance d with the closest vehicle, a security distance d_{ref} with the closest vehicle and a relative speed (v_r) between the closest vehicle and the system vehicle.

Table 4: Fit (%) for system validation, with several configurations of steps ahead

	1	5	10	50	∞
Model 1A	95.56	86.49	83.03	84.56	82.28
Model 1B	96.14	89.73	87.95	91.72	87.37
Model 1C	94.01	84.47	83.29	83.88	83.27
Model 2A	96.15	87.80	83.35	85.87	79.62
Model 2B	96.60	90.12	86.83	84.86	84.15
Model 2C	94.40	81.55	75.41	77.51	74.09
Model 3A	95.10	84.95	78.45	76.48	73.87
Model 3B	94.90	88.97	85.83	84.39	83.93
Model 3C	92.54	82.83	77.28	84.97	74.47

Table 5: Switching logic between CC and ACC modes

	$v < v_{ref}$	$(v \geq v_{ref})$ & $(v_r < 0)$	$(v \geq v_{ref})$ & $(v_r \geq 0)$
$d > d_{ref}$	CC	CC	CC
$d \leq d_{ref}$	ACC	ACC	CC

The controller for the outer loop must compute d_{ref} and v_{ref} to decide which mode must be activated. In Shakouri et al. (2015), they suggest computing d_{ref} as in Equation 1. The parameter ℓ is the length of the system vehicle, d_s it is an additional distance to avoid crashes and T_h is known as constant-time headway, which estimates the time of action and reaction for drivers. Usually, this value varies between 0.8 and 2 seconds, without any adverse conditions (Shakouri et al., 2015).

$$d_{ref} = \ell + d_s + T_h v \quad (1)$$

The suggested control law for computing v_{ref} of Shakouri et al. (2015) is shown in Equation 2. If the distance d to the closest vehicle is the same as the security distance d_{ref} , the system is safe to travel with the same speed of the closest vehicle v_l , commonly described as leading vehicle. Otherwise, it is necessary to change the security speed v_{ref} using a proportional gain K_p .

$$v_{ref} = v_l - K_p(d_{ref} - d) \quad (2)$$

If the controller is set in CC mode, the cruise speed for the inner loop control will be v_{user} , because it is safe to achieve that speed, given restrictions by d_{ref} and v_{ref} . If the controller is set in ACC mode, the cruise speed must be v_{ref} to keep a safe distance from the leading vehicle.

3.2 MPC Design

The first control design for the inner loop is using a finite horizon MPC. The formulation for the MPC design was chosen using a state space model with an incremental input $\Delta u(k) = u(k) - u(k-1)$ as follows:

$$\begin{cases} x(k+1) = Ax(k) + B\Delta u(k) \\ y(k) = Cx(k) \end{cases}$$

Each ARX model was transformed into this representation, given a realization in state space and incorporating the time delay as additional states. The input can be set in incremental form using two subsequent instants (Camacho and Alba, 2013). The objective function for finite horizon MPC can be described as (Maciejowski, 2002):

$$J_k^{\text{MPC}} = \sum_{j=1}^p (y(k+j|k) - y^{sp})^T Q \times (y(k+j|k) - y^{sp}) + \sum_{j=0}^{m-1} \Delta u(k+j|k)^T R \Delta u(k+j|k) \quad (3)$$

The element y^{sp} is the desired output value and the control parameters are: m is the control horizon; p is the prediction horizon; Q and R are weight matrices related to output error and control input, respectively.

Foremost, the output prediction vector is assembled, considering that after the sample $k+m$ there are no more control inputs ($\Delta u(k+m|k) = \Delta u(k+m+1|k) = \dots = 0$). The Equation (4) shows the prediction to the horizon p , compressing all m control signals into Δu_k .

$$\bar{y}(k) = \Phi x(k) + \Gamma \Delta u_k \quad (4)$$

with:

$$\Phi = \begin{bmatrix} CA \\ CA^2 \\ \vdots \\ CA^m \\ CA^{m+1} \\ \vdots \\ CA^p \end{bmatrix}$$

and

$$\Gamma = \begin{bmatrix} CB & 0 & \dots & 0 \\ CAB & CB & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ CA^{m-1}B & CA^{m-2}B & \dots & CB \\ CA^mB & CA^{m-1}B & \dots & CAB \\ \vdots & \vdots & \vdots & \vdots \\ CA^{p-1}B & CA^{p-2}B & \dots & CA^{p-m}B \end{bmatrix}.$$

Considering that the output set-point y^{sp} for any output prediction, then the set-point vector will be $\bar{y}^{sp} = \left[\underbrace{y^{sp} \dots y^{sp}}_p \right]^T$. To expand the sum

in (3), it is also necessary to consider weight matrices Q and R in their respective horizons, obtaining $\bar{Q} = \text{diag} \left[\underbrace{Q \dots Q}_p \right]$ and $\bar{R} = \text{diag} \left[\underbrace{R \dots R}_m \right]$. Expanding all elements of (3), the objective function J_k^{MPC} can be reduced to a quadratic form:

$$J_k^{\text{MPC}} = \Delta u_k^T H \Delta u_k + 2c_f^T \Delta u_k + c$$

where

$$\begin{cases} H = \Gamma^T \bar{Q} \Gamma + \bar{R}; \\ c_f^T = (\Phi x(k) - \bar{y}^{sp})^T \bar{Q} \Gamma; \\ c = (\Phi x(k) - \bar{y}^{sp})^T \bar{Q} (\Phi x(k) - \bar{y}^{sp}). \end{cases}$$

The control law for conventional MPC will be the solution of the following Quadratic Programming (QP), subject to control input constraints:

$$\begin{aligned} \min_{\Delta u_k} \Delta u_k^T H \Delta u_k + 2c_f^T \Delta u_k \\ \text{s.t.} \\ -\Delta u_{\max} \leq \Delta u(k+j|k) \leq \Delta u_{\max}, \\ u_{\min} \leq u(k+j|k) \leq u_{\max}, j = 0, 1, \dots, m-1 \end{aligned} \quad (5)$$

3.3 IHMPC Design

The second control design for the inner loop is using an infinite horizon MPC (IHMPC). The model representation chosen for the IHMPC was Output Prediction Oriented Model (OPOM) (Odloak, 2004; Martins and Odloak, 2016). As in MPC design, each ARX model was transformed into a state space representation, with incorporated time delays and in incremental form, but using the OPOM formulation. The objective function for IHMPC can be described as (Odloak, 2004):

$$J_k^{\text{IHMPC}} = \underbrace{\sum_{j=0}^{\infty} (y(k+j|k) - y^{sp} - \delta_y)^T Q (y(k+j|k) - y^{sp} - \delta_y)}_{J_1} + \sum_{j=0}^{m-1} \Delta u(k+j|k)^T R \Delta u(k+j|k) + \delta_y^T S_y \delta_y \quad (6)$$

For this control technique, it is essential to use slack variables δ_y for each output, because the control law will converge to an expression with equality constraints. Without slack variables, it is possible to have unfeasible solutions. The IHMPC control law must compute the control input Δu and slack variable δ_y . S_y is the weight matrix related to the slack variables and θ_{max} is the maximum time delay of the system. At first, an expansion of the infinite sum of J_1 of Equation (6) is expressed in Equation (7):

$$\begin{aligned}
J_1 = & \underbrace{\sum_{j=0}^{m+\theta_{max}} (y(k+j|k) - y^{sp} - \delta_y)^T Q (y(k+j|k) - y^{sp} - \delta_y)}_{J_{1a}} \\
& + \underbrace{\sum_{j=m+\theta_{max}+1}^{\infty} (y(k+j|k) - y^{sp} - \delta_y)^T Q (y(k+j|k) - y^{sp} - \delta_y)}_{J_{1b}} \quad (7)
\end{aligned}$$

Similar to the conventional MPC, an output prediction vector is calculated with $m + \theta_{max}$ as the prediction horizon:

$$\bar{y}(k) = \bar{A}x(k) + \bar{B}\Delta u_k \quad (8)$$

with θ_{max} being the maximum time-delay of the system and:

$$\bar{A} = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^m \\ \vdots \\ CA^{m+\theta_{max}} \end{bmatrix}$$

and

$$\bar{B} = \begin{bmatrix} 0 & 0 & \dots & 0 \\ CB & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ CA^{m-1}B & CA^{m-2}B & \dots & CB \\ \vdots & \vdots & \vdots & \vdots \\ CA^{m+\theta_{max}-1}B & CA^{m+\theta_{max}-2}B & \dots & CA^{\theta_{max}}B \end{bmatrix}$$

Consequently, creating an output set-point vector $\bar{y}^{sp} = \underbrace{[y^{sp} \dots y^{sp}]^T}_{[m+\theta_{max}+1]}$, a supporting vector $\bar{I}_{ny} = \underbrace{[I_{ny} \dots I_{ny}]^T}_{[m+\theta_{max}+1]}$ and adjusting the weighting matrix Q as $\bar{Q}_y = \text{diag} \left[\underbrace{Q \dots Q}_{[m+\theta_{max}+1]} \right]$, the first expansion element of J_1 can be calculated as:

$$\begin{aligned}
J_{1a} = & (\bar{A}x(k) + \bar{B}\Delta u_k - \bar{y}^{sp} - \bar{I}_{ny}\delta_y)^T \bar{Q}_y \times \\
& (\bar{A}x(k) + \bar{B}\Delta u_k - \bar{y}^{sp} - \bar{I}_{ny}\delta_y)
\end{aligned}$$

For the element J_{1b} , using the OPOM definition, the output prediction for any time instant j after $m + \theta_{max}$ can be described as:

$$\begin{aligned}
y(k+m+\theta_{max}+j|k) = & x^s(k+m+\theta_{max}|k) + \\
& \Psi x^d(k+m+\theta_{max}+j|k)
\end{aligned}$$

The previous equation can be described as:

$$y(k+m+\theta_{max}+j|k) = \bar{x}^s + \Psi \bar{x}^d(j) \quad (9)$$

Replacing expansion (9) in J_{1b} :

$$\begin{aligned}
J_{1b} = & \sum_{j=1}^{\infty} (\bar{x}^s + \Psi \bar{x}^d(j) - y^{sp} - \delta_y)^T Q_y \times \\
& (\bar{x}^s + \Psi \bar{x}^d(j) - y^{sp} - \delta_y)
\end{aligned}$$

Being F stable, the condition for J_{1b} to be bounded is:

$$\bar{x}^s - y^{sp} - \delta_y = 0$$

If the above condition is satisfied, J_{1b} remains as follows:

$$\begin{aligned}
J_{1b} &= \sum_{j=1}^{\infty} (\Psi \bar{x}^d(j))^T Q_y (\Psi \bar{x}^d(j)) \\
&= \bar{x}^d(0)^T \underbrace{\left(\sum_{j=1}^{\infty} F^j T \Psi^T Q_y \Psi F^j \right)}_{\bar{Q}_d} \bar{x}^d(0) \quad (10)
\end{aligned}$$

The matrix \bar{Q}_d can be calculated as solution of the following discrete Lyapunov equation:

$$\bar{Q}_d = F^T \Psi^T Q_y \Psi F + F^T \bar{Q}_d F$$

Next, the predicted states \bar{x}^s and $\bar{x}^d(0)$ are obtained:

$$\begin{cases} \bar{x}^s = N_s A^{\theta_{max}} (A^m x(k) + W \Delta u_k) \\ \bar{x}^d(0) = N_d A^{\theta_{max}} (A^m x(k) + W \Delta u_k) \end{cases}$$

where n_d is the number of poles of the system

and

$$\begin{cases} N_s = \begin{bmatrix} I_{ny} & 0_{ny \times n_d} & 0_{ny \times \theta_{max}} \end{bmatrix}; \\ N_d = \begin{bmatrix} 0_{n_d \times ny} & I_{n_d} & 0_{n_d \times \theta_{max}} \end{bmatrix}; \\ W = \begin{bmatrix} A^{m-1}B & A^{m-2}B & \dots & B \end{bmatrix}. \end{cases}$$

As well as in MPC formulation, $J_k^{\text{IHMP}}C$ can be reduced into a quadratic form:

$$\begin{aligned}
J_k^{\text{IHMP}}C = & \begin{bmatrix} \Delta u_k & \delta_y \end{bmatrix} \begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix} \begin{bmatrix} \Delta u_k \\ \delta_y \end{bmatrix} + \\
& 2 \begin{bmatrix} c_{f1} & c_{f2} \end{bmatrix} \begin{bmatrix} \Delta u_k \\ \delta_y \end{bmatrix} + c \quad (11)
\end{aligned}$$

where

$$\left\{ \begin{array}{l} H_{11} = (\bar{B})^T \bar{Q}_y (\bar{B}) + \bar{R} + \\ \quad (N_d A^{\theta_{max}} W)^T \bar{Q}_d (N_d A^{\theta_{max}} W); \\ H_{12} = -(\bar{B})^T \bar{Q}_y (\bar{I}_{ny}); \\ H_{21} = H_{12}^T; \\ H_{22} = (\bar{I}_{ny})^T \bar{Q}_y (\bar{I}_{ny}) + S_y; \\ c_{f1} = (\bar{A}x(k) - \bar{y}^{sp})^T \bar{Q}_y (\bar{B}) + \\ \quad (N_d A^{\theta_{max}+m} x(k))^T \bar{Q}_d (N_d A^{\theta_{max}} W); \\ c_{f2} = -(\bar{A}x(k) - \bar{y}^{sp})^T \bar{Q}_y (\bar{I}_{ny}); \\ c = (\bar{A}x(k) - \bar{y}^{sp})^T \bar{Q}_y (\bar{A}x(k) - \bar{y}^{sp}) + \\ \quad (N_d A^{\theta_{max}+m} x(k))^T \bar{Q}_d (N_d A^{\theta_{max}+m} x(k)). \end{array} \right.$$

The control law for IHMPC will be the solution of the following QP, subject to control input constraints and the restriction of the slack variables:

$$\begin{aligned} \min_{\Delta u_k, \delta_y} & \left(\begin{bmatrix} \Delta u_k & \delta_y \end{bmatrix} \begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix} \begin{bmatrix} \Delta u_k \\ \delta_y \end{bmatrix} + \right. \\ & \left. 2 \begin{bmatrix} c_{f1} & c_{f2} \end{bmatrix} \begin{bmatrix} \Delta u_k \\ \delta_y \end{bmatrix} \right) \\ & s.t. \\ & -\Delta u_{max} \leq \Delta u(k+j|k) \leq \Delta u_{max}, \\ & u_{min} \leq u(k+j|k) \leq u_{max}, j = 0, 1, \dots, m-1 \\ & N_s A^{\theta_{max}} (A^m x(k) + W \Delta u_k) - y^{sp} - \delta_y = 0 \end{aligned} \quad (12)$$

4 Simulations

Firstly, the tuning of the outer loop controller was set with the parameters of d_{ref} (Equation (1)) and v_{ref} (Equation (2)). The security distance d_{ref} and the security speed v_{ref} were computed with:

$$\ell = 4m; d_s = 10m; T_h = 2s; K_p = 0.022$$

During the simulations, the distance d with the leading vehicle was estimated integrating the relative speed v_r . In real applications, this distance should be given by a radar. With a low value of K_p , the ACC mode computes a cruise speed close to the leading vehicle speed v_l , with subtle adjustments related to the distance error.

The system actuator usually has three constraints: maximum value, minimum value and maximum slew rate. As discussed in Section 2, the maximum value of the throttle pedal is $u_{max} = 1$ and the minimum value is $u_{min} = 0$. Since there is no explicit limitation regarding throttle slew rate, it has been decided to use this constraint as a tuning parameter for a smoother controller response. The maximum slew rate chosen for MPC and IHMPC was $\Delta u_{max} = 0.1$, for positive and negative input variations. All simulations had a time sampling of $T_s = 0.5$ seconds.

4.1 ACC with MPC

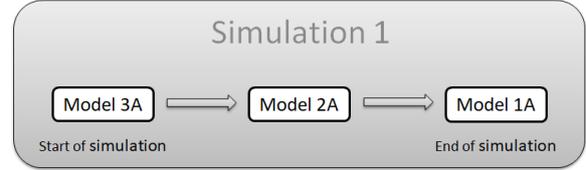
The tuning of MPC was performed with simulations in CC mode, obtaining the following controller parameters:

$$p = 10; m = 5; Q = 50; R = 1.$$

For simulation 1, the user defined speed was set constant with $v_{user} = 15$ m/s. The controlled vehicle begins the simulation with null speed and a leading vehicle drives at $v_l = 15$ m/s. During all simulation, the MPC computes its control law with Model 3A. At instants $t = [15, 40, 65]$, the leading vehicle suddenly breaks 10% of its speed, simulating a heavier traffic.

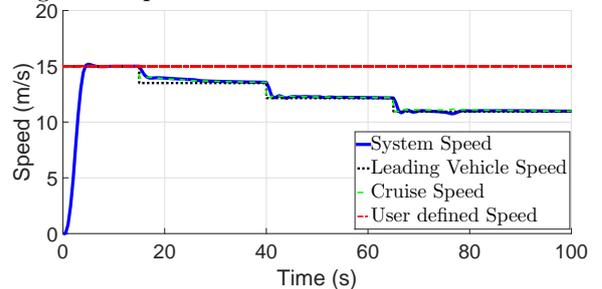
To analyze the robustness of the controller, during the simulation the system model smoothly changes. The system model response is calculated using a linear combination of three models, but at the start, the system model is equal to Model 3A (represented in Figure 4). For each time sampling, the importance of Model 3A slowly reduces and the weight of Model 2A increases. After the system model equals to Model 2A, its subsequent weight slowly reduces and the weight of Model 1A starts to increase, until the end of simulation.

Figure 4: System model diagram of Simulation 1.



The system speed response for Simulation 1 is shown in Figure 5. Also in this Figure, it is shown the leading vehicle speed v_l , the cruise speed computed by the outer loop controller and the user defined speed v_{user} . The controller was able to follow the constantly changing set-point (cruise speed), even with the system model changing with every time sampling.

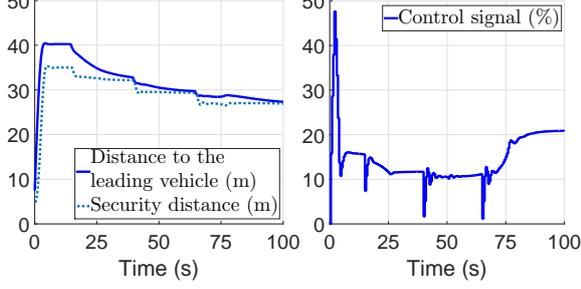
Figure 5: Speeds for the Simulation 1 with MPC.



In Figure 6, the simulated distance to the leading vehicle is shown, assuring a safe distance within all the simulation. In the same Figure, the control input adapts its intensity for the changes of the set-point and the system model response.

For simulation 2, the user defined speed was set constant with $v_{user} = 15$ m/s. The controlled

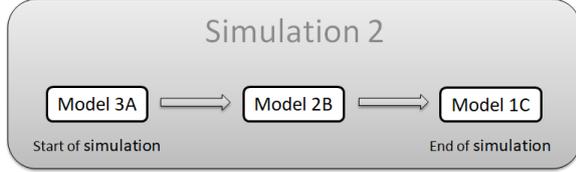
Figure 6: Distances and Control Signal for the Simulation 1 with the MPC.



vehicle also begins the simulation with null speed and a leading vehicle drives at $v_l = 15$ m/s. During all simulation, the MPC computes its control law only with Model 3A.

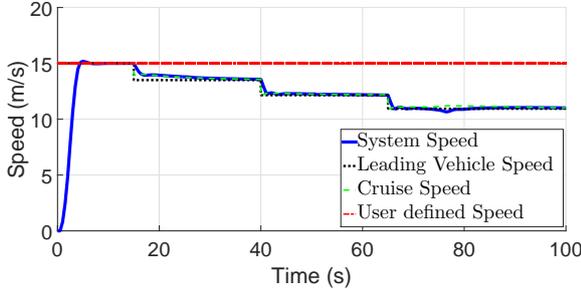
The leading vehicle also breaks at the same instants, but in this case for a different reason. In simulation 2, the system model changes from models of low load to models of higher loads, representing a harder terrain, for example with higher slope hill, snow or a dirt road (represented in Figure 7).

Figure 7: System model diagram of Simulation 2.



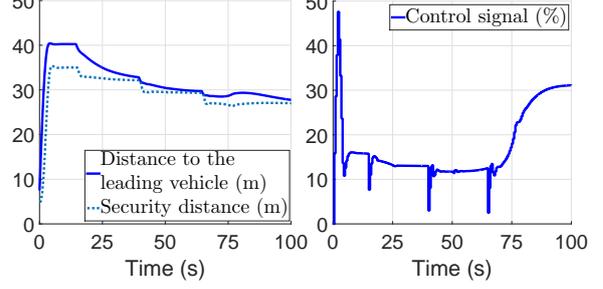
The system speed response and all other important speeds for Simulation 2 are shown in Figure 8. The controller also was able to follow the constantly changing set-point (cruise speed) and system model.

Figure 8: Speeds for the Simulation 2 with MPC.



In Figure 9, the distance to the leading vehicle remains safe during all the simulation. The control signal adapts with the changes of the set-point and the system model response, even with the different static gains for each model.

Figure 9: Distances and Control Signal for the Simulation 2 with the MPC.



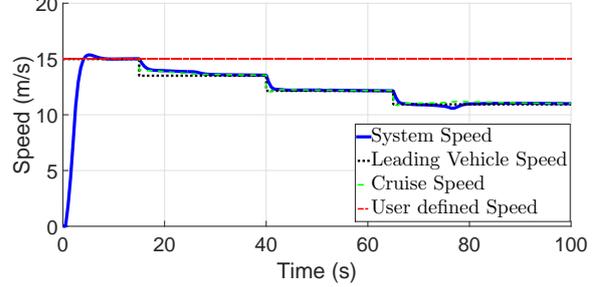
4.2 ACC with IHMPC

The tuning of IHMPC was also performed with simulations in CC mode, selecting:

$$m = 10; Q = 0.1; R = 100; S = 1000.$$

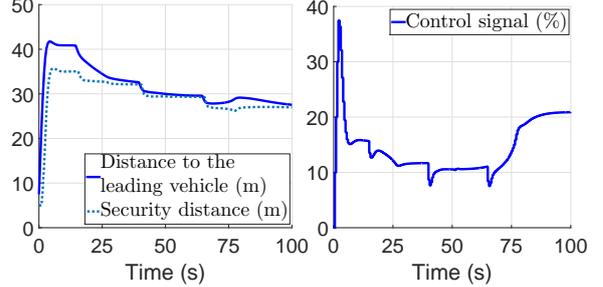
The same scenario of Simulation 1 (Figure 4) was repeated with IHMPC, using the same Model 3A for the model prediction. The system speed and the other speeds are shown in Figure 10. In comparison with MPC response, IHMPC had a slightly smoother response.

Figure 10: Speeds for the Simulation 1 with IHMPC.



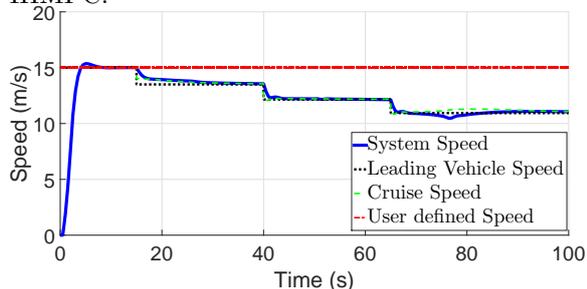
The simulated distance and the control input for Simulation 1 with IHMPC are shown in Figure 11. The distance d maintained at safe values during all the simulation.

Figure 11: Distances and Control Signal for the Simulation 1 with the IHMPC.



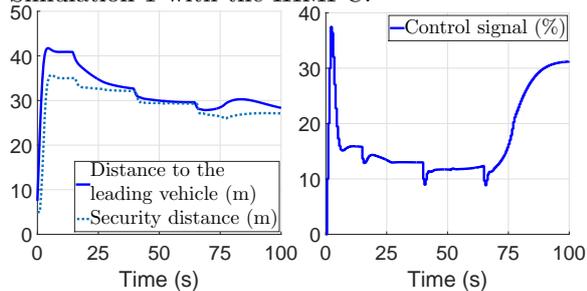
The scenario of Simulation 2 (Figure 7) was repeated with IHMPC. The system speed and the other speeds are shown in Figure 12.

Figure 12: Speeds for the Simulation 2 with IHMPC.



The simulated distance and the control input for Simulation 2 with IHMPC are shown in the Figure 13. The controller kept the distance at safe values during all the simulation.

Figure 13: Distances and Control Signal for the Simulation 1 with the IHMPC.



4.3 Discussions

Foremost, it is important to note that in both MPC approaches the control law results in a quadratic programming problem. In the matter of the ACC system performance, both controllers were successful to track the cruise speed while maintaining a safe distance to the leading vehicle. The controllers were able to perform with decent performance even in cases that the model for the prediction differs from the model carried out as the system.

There are subtle differences between the control signals of both approaches. Generally, the control signal of the IHMPC has fewer oscillations. Such a feature is due to the definition of the states $\bar{x}^d(j)$, in which the IHMPC control law tries to reduce the dynamic effects in the prediction after $m + \theta_{max}$ samples. If the IHMPC manages to reduce these dynamic states, the system will converge to the set-point smoothly.

5 Conclusions

The high Fit indexes of the ARX models indicates that the models appear to be validated. Regarding the performance of controllers, both MPC and IHMPC presented satisfactory results. IHMPC had slightly smoother responses compared to MPC. Although MPC and IHMPC have distinct objective functions, both controllers have a corresponding QP, which assists for future embedded applications. Even with the ACC controller having only one tuning parameter (K_p), its performance was satisfactory.

Acknowledgments

Thanks to the Conselho Nacional de Desenvolvimento Científico e Tecnológico for the scholarship, Process 132237/2017-2.

References

- Camacho, E. and Alba, C. (2013). *Model Predictive Control*, Advanced Textbooks in Control and Signal Processing, Springer London.
- Dias, J. E. A., Pereira, G. A. S. and Palhares, R. M. (2015). Longitudinal model identification and velocity control of an autonomous car, *IEEE Transactions on Intelligent Transportation Systems* **16**(2): 776–786.
- Ljung, L. (1999). *System Identification (2nd Ed.): Theory for the User*, Prentice Hall PTR, Upper Saddle River, NJ, USA.
- Maciejowski, J. (2002). *Predictive Control: With Constraints*, Pearson Education, Prentice Hall.
- Martins, M. A. and Odloak, D. (2016). A robustly stabilizing model predictive control strategy of stable and unstable processes, *Automatica* **67**(Supplement C): 132 – 143.
- Odloak, D. (2004). Extended robust model predictive control, *AIChE Journal* **50**(8): 1824–1836.
- Shakouri, P., Czczot, J. and Ordys, A. (2010). Longitudinal vehicle dynamics using simulink/matlab, *Conference...*, UKACC, IET, pp. 1–6.
- Shakouri, P., Czczot, J. and Ordys, A. (2012). Adaptive cruise control system using balance-based adaptive control technique, *Conference...*, IEEE, pp. 510–515.
- Shakouri, P., Czczot, J. and Ordys, A. (2015). Simulation validation of three nonlinear model-based controllers in the adaptive cruise control system, *Journal of Intelligent and Robotic Systems* **80**(2): 207–229.