

Longitudinal Model Identification and Velocity Control of an Autonomous Car

Jullierme Emiliano Alves Dias, Guilherme Augusto Silva Pereira, *Senior Member, IEEE*, and Reinaldo Martinez Palhares

Abstract—This paper presents the model identification and the velocity control of an autonomous car. The control system was designed so that the car is controlled at low speeds, where the main applications for the vehicle's autonomous operations include parking and urban adaptive cruise control. A longitudinal model of the car was used in the control loop to compensate the nonlinear behavior of its dynamics. Since the determination of the vehicle's model is a difficult step in the design of model-based controllers, the main contribution of this paper is the use of an empirically determined model to this end. In this paper, the structure of the model was conceived from the car's physics equations, but its parameters were estimated using data-based identification techniques. An important contribution of this paper is the fact that, although the model is strictly linear, we can change its parameters as a function of the operation point of the vehicle to represent the engine's and the transmission's nonlinear behaviors. Moreover, in this paper, we propose a way to include changes in the longitudinal dynamics caused by the automatic gear shifting. The validation of the proposed controller was conducted by computer simulations and real-world experiments.

Index Terms—Intelligent vehicle, mathematical model, system identification, velocity control.

I. INTRODUCTION

AUTONOMOUS vehicles are likely to play a major role in future transportation systems, since they provide many potential benefits, such as increased safety and higher utilization of roads. Motivated by this, since 2007, the Group for Research and Development of Autonomous Vehicles (PDVA) at the Federal University of Minas Gerais (UFMG/Brazil) has been working on the design and development of the Autonomous Car Developed at UFMG (CADU, a Portuguese acronym), which is a commercial automobile (Chevrolet Astra Sedan 2003) equipped with automatic transmission, ABS, and drive-by-wire throttle (see Fig. 1). Commands such as throttle,



Fig. 1. Autonomous Car Developed at UFMG (CADU).

brake, steering, and gear shifting have been automated and controlled by a computer through a real-time network. Several developments have been made in this project, which allowed the car to drive itself in scenarios with obstacles using cameras and other sensors [1]. In this paper, we present the longitudinal velocity controller of this vehicle.

Commercially, the longitudinal velocity controllers are included in the cruise control (CC) systems. In general, the CC is functional only in medium and high speeds (above 30 or 40 km/h); for this reason, it cannot be directly used in autonomous cars that operate at urban traffic speeds. The main difference between the standard CC systems and the controllers needed by autonomous vehicles is the fact that the former only controls the throttle to maintain a constant desired speed. It is the driver's responsibility to brake the car if the current speed is no longer safe for travel. Longitudinal controllers for autonomous cars need to control both throttle and brake to ensure that a velocity profile (specified by a higher level controller) is followed by the car. Only recently some commercial cars have been equipped with systems, such as the adaptive CC (ACC) systems, which also brake the vehicle and allow it to operate in traffic conditions, in which the vehicle's speed can range from 0 to more than 100 km/h.

In this paper we present a practical way to implement a model-based velocity controller that operates in longitudinal velocities that range from 0 to 40 km/h. The system controls the vehicle using throttle and brake and, thus, can be used in autonomous operations. Provided that some specific tests can be executed with the vehicle to construct a mathematical model, this controller is a candidate to control any autonomous car. Since it operates at low speeds, it can be also useful, for example to compose autonomous parking or ACC [2] systems of

Manuscript received February 5, 2014; revised June 4, 2014; accepted July 14, 2014. Date of publication August 28, 2014; date of current version March 27, 2015. This work was supported in part by the Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG/Brazil), Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq/Brazil) and Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES/Brazil). The work of J. E. A. Dias, G. A. S. Pereira, and R. M. Palhares was supported by a CNPq Scholarship. The Associate Editor for this paper was M. Da Lio.

The authors are with the School of Engineering, Universidade Federal de Minas Gerais, Belo Horizonte, MG 31270-901, Brazil (e-mail: julliermead@ufmg.br; gpereira@ufmg.br; palhares@cpdee.ufmg.br).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TITS.2014.2341491

any automobile. It is important to mention that controlling the velocity of the vehicle at low speeds presents some challenges. Despite the use of the brake, at low speeds the dynamics of the powertrain, which presents nonlinear behavior, dominates the dynamics of the body, making it difficult to use simplified models, such as those proposed in [3], to describe the vehicle's behavior. Because of this, several controllers for autonomous cars, such as some of the DARPA Urban Challenge participants, rely on engine maps that relate torque and velocity to linearize the system [4], [5].

In this paper, to compose the proposed controller, a model of the car's longitudinal dynamics was obtained using gray-box identification techniques [6], having as input the throttle level and the vehicle longitudinal velocity. The proposed model was inspired by the work in [7], which shows a study in which several parametric models for an automobile were identified and compared. Similar to [7], the model used in this paper is linear in essence, but here, due to an approach that allows for changing the parameters of the model, it can correctly represent the vehicle, which is a nonlinear system. This is in consonance with the results in [7], in which Hunt *et al.* showed that the interpolation of several linear models is a good way to represent the car. Differently from [7], in this paper, compensation for gear change was included in the model by online estimating the transmission rate and using this information to change the gain of the model. This, in addition to the use of braking as a second input for the model, makes it possible to use the mathematical model to perform speed control for all possible operation modes of the vehicle.

Similar to what was proposed in [3], our controller is based on the **inverse model of the vehicle**, which is used to compensate the system dynamics, including its nonlinear behavior. However, in [3], and in other works, such as [8], which proposed a different model-based controller, the main difficulty encountered in making the controller to work in practice is to have a precise model for the vehicle. Good models are, in general, highly dependent on complete knowledge of the components of the car and of their dynamical nonlinear behaviors, including the engine map. This knowledge may be available for the vehicle's manufacturer, but it is usually hard to obtain by an end user. Therefore, the main contribution of this paper is a methodology that allows the empirical determination of the vehicle's mathematical model. Within this methodology, our key contribution is to change the parameters of the model as a function of the vehicle's velocity and gear ratio. This allows a very simple model structure to represent the car in a wide range of velocities, including speeds smaller than 40 km/h, where the nonlinear behavior of the powertrain dominates the system dynamics. Although we present a controller that uses the determined model to control the vehicle's velocity, the modeling methodology, which does not depend on any previous knowledge about the vehicle, can be used along with any model-based velocity controller, such as those found in [9]–[11].

To the best of the authors' knowledge, the methodology presented in this paper fills a gap in the literature for low-speed longitudinal control of autonomous vehicles built without manufacturer support. Notice that most autonomous cars that

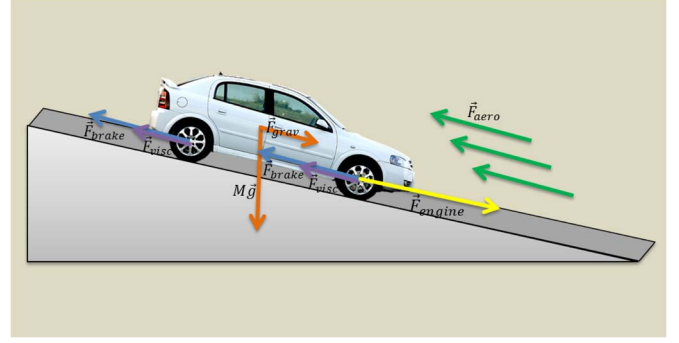


Fig. 2. Diagram of longitudinal forces acting on a car.

participate in the DARPA Urban Challenge, for instance, were developed with the aid of the vehicles' manufacturer [5], [12]. A few teams, such as [13] and [14], have developed their own controllers, but unfortunately, these teams have not published details about the controller design, which is done in this paper.

The rest of this paper is organized as follows. Sections II and III show the methodologies used to obtain a longitudinal model of the vehicle and to design the velocity controller, respectively. Simulations and experimental results are presented in Section IV. Finally, conclusions are presented in Section V.

II. LONGITUDINAL MODELING

A. Model Structure

As seen in [15], the longitudinal dynamics of a commercial car may be described by Newton's second law. Thus, the inertial force applied to the car is given by the sum of the following forces: the viscous resistance due to the components of the transmission system (\vec{F}_{visc}), the aerodynamic drag (\vec{F}_{aero}), the braking force (\vec{F}_{brake}), the climbing resistance or down-grade force (\vec{F}_{grav}), and the engine drive force transmitted to the wheels by the transmission system (\vec{F}_{engine}). The following general equation is then posed:

$$\frac{d}{dt}(M\vec{v}) = \vec{F}_{\text{visc}} + \vec{F}_{\text{aero}} + \vec{F}_{\text{brake}} + \vec{F}_{\text{grav}} + \vec{F}_{\text{engine}} \quad (1)$$

where M represents the total mass of the vehicle, and \vec{v} is its longitudinal velocity. Fig. 2 shows a free-body diagram representing (1).

For any situation, forces \vec{F}_{visc} , \vec{F}_{aero} , and \vec{F}_{brake} are always opposed to the movement and, therefore, have always a negative sign. On the other hand, \vec{F}_{grav} and \vec{F}_{engine} may insert or remove kinetic energy to the car, depending on the slope of the terrain and the operation point of the engine, respectively.

To start the development of a parametric model that is suitable for parameter identification, we assume the simplest situation where the car moves in a flat and leveled surface, without braking, and at low speeds. In this condition, forces \vec{F}_{grav} , \vec{F}_{brake} , and \vec{F}_{aero} in (1) are very small and can be neglected.

One of the remaining forces, i.e., \vec{F}_{engine} , which represents the main force that determines the movement, is caused by the

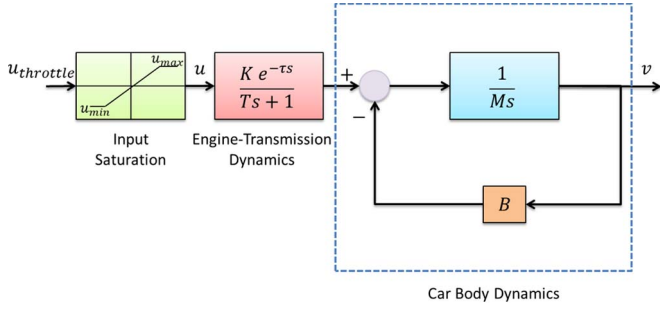


Fig. 3. Block diagram of the dynamic system. The inputs u_e and u represent the throttle level, respectively, before and after saturation required by the ECU. The resulting velocity is designated by v .

engine and is transferred to the car by the transmission system. These components are very difficult to be modeled, since they present a nonlinear behavior. Moreover, in commercial automobiles, some devices, such as air conditioning and hydraulic power steering pump, are connected to the transmission system, thus changing the system behavior. Therefore, the design of a mathematical model to describe \vec{F}_{engine} is a highly complex and challenging task.

Given these difficulties, in this paper, the simplification proposed in [15] is initially adopted, which consists in representing the engine transmission system by a first-order linear system with time constant T , static gain K , and time delay τ . Additionally, the model's input, which corresponds to the throttle signal sent to the electronic control unit (ECU) of the vehicle's powertrain, is limited between the value corresponding to the engine's idle speed and a maximum value chosen for security reasons. After executing some experiments, we realized that this simple model is not a good representation of the system and needs to be adapted as a function of the engine's operation point, as we will show later in this paper.

By writing (1) in the frequency domain, we can represent the longitudinal dynamics of the car by the block diagram shown in Fig. 3. Force \vec{F}_{visc} in this diagram was considered to be related to viscous friction among internal parts of the vehicle, being proportional to its longitudinal speed with constant B .

By computing the resulting transfer function of the block diagram in Fig. 3, where the input is the throttle level u , and the output is the velocity v , we have

$$G(s) = \frac{V(s)}{U(s)} = \frac{\frac{K}{MT} e^{-\tau s}}{(s + \frac{1}{T})(s + \frac{B}{M})} \quad (2)$$

which represents the dynamics of the vehicle in continuous time. To use this model in computational applications, it is necessary to obtain a discrete version of (2). Using the Z transform and a zero-order hold [16], we obtain the following discrete model:

$$G(z) = \frac{V(z)}{U(z)} = \frac{\alpha + \beta z^{-1}}{1 + \gamma z^{-1} + \sigma z^{-2}} z^{-\frac{\tau}{T_a}} \quad (3)$$

where T_a is the sampling period, which, in this work, was chosen such that rate τ/T_a would be an integer, and parameters

α , β , γ , and σ are given by

$$\begin{aligned} \alpha &= K \left[\frac{\gamma(BT - M) + (e^{-\frac{BT_a}{M}} - 1)TB + M(e^{-\frac{T_a}{T}} - 1)}{B(BT - M)} \right] \\ \beta &= K \left[\frac{\sigma(BT - M) - e^{-\frac{BT_a}{M}}TB - e^{-\frac{T_a}{T}}M}{B(BT - M)} \right] \\ \gamma &= - \left(e^{-\frac{T_a}{T}} + e^{-\frac{BT_a}{M}} \right) \\ \sigma &= e^{-[\frac{T_a}{T} + \frac{BT_a}{M}]} \end{aligned}$$

Model (3) presents some unknown parameters (K , T , B , M , e , and τ) that are difficult to obtain without making individual tests with the components of the vehicle. Moreover, remember that a linear model has been used to explain a nonlinear system, which makes some of these parameters vary as a function of the operating point. In this paper, we propose the use of data-based parameter identification to overcome some of these difficulties. For this, (3) must be rewritten such that it becomes suitable to have its parameters estimated by well-known identification techniques.

In this paper, we choose to represent model (3) as an Auto Regressive with eXogeneous inputs (ARX) model. A typical equation for such a model is given by

$$A(q)y(k) = B(q)u(k) + w(k) \quad (4)$$

where $y(k)$, $u(k)$, and $w(k)$, are, respectively, the model's output, the model's input, and additive white noise for a given sample k . $A(q)$ and $B(q)$ are polynomials of the type

$$\begin{aligned} A(q) &= 1 - a_1 q^{-1} - \dots - a_{n_y} q^{-n_y} \\ B(q) &= b_1 q^{-1} + \dots + b_{n_u} q^{-n_u} \end{aligned}$$

where q^{-1} is the delay operator, and the coefficients a_1, a_2, \dots, a_{n_y} and b_1, b_2, \dots, b_{n_u} are the parameters related to delayed input and output samples, respectively.

It is possible to transform (3) directly into an ARX model by simply substituting z^{-1} by q^{-1} . This substitution is necessary because, strictly speaking, q^{-1} is an operator, which represents a delay of one sample, whereas z^{-1} is the inverse of the complex variable z . Therefore, the ARX model corresponding to (3) can be written as

$$\begin{aligned} y(k) &= a_1 y(k-1) + a_2 y(k-2) + b_1 u \left(k - \frac{\tau}{T_a} \right) \\ &\quad + b_2 u \left(k - \frac{\tau}{T_a} - 1 \right) + e(k) \end{aligned} \quad (5)$$

where $a_1 = -\gamma$, $a_2 = -\sigma$, $b_1 = \alpha$, and $b_2 = \beta$.

The estimate of the parameters of (5) can be done using a least squares method over pairs of input and output data. Using such a method, we find the parameters that minimize the error of the model for one-step-ahead prediction. To use a least squares method, we rewrite the model as

$$y(k) = \psi^T(k-1)\hat{\theta} + \xi(k) \quad (6)$$

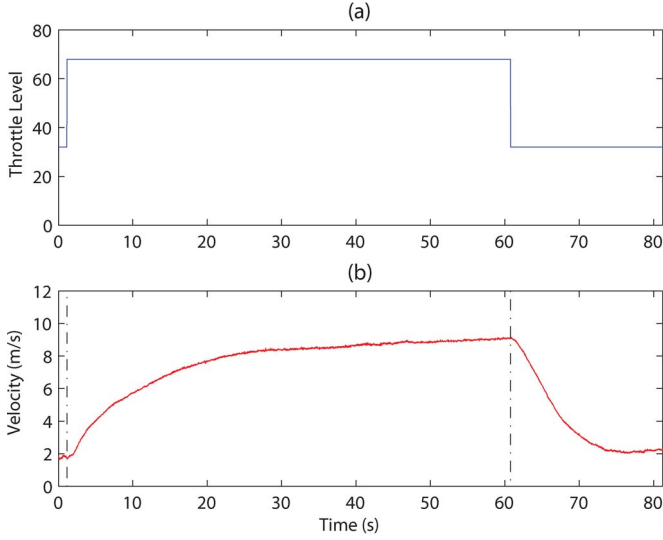


Fig. 4. Step response. (a) Input: throttle level. (b) Output: velocity in meters per second.

where $\psi^T(k-1)$ is a vector that contains input and output data collected until instant $k-1$, $\hat{\theta}$ is the vector of estimated parameters at instant k , and $\xi(k)$ represents the residuals between the estimated output $\hat{y}(k)$ and the measured output $y(k)$. When applied to a mass of data, (6) must be written in matrix form as

$$Y = \Psi \hat{\theta} + \Xi \quad (7)$$

where Y is the measured output vector, Ψ is the regressors matrix, and Ξ is the residual vector, for all time instants k . The parameter vector $\hat{\theta}$ may be obtained via the minimum norm solution

$$\hat{\theta} = [\Psi^T \Psi]^{-1} \Psi^T Y. \quad (8)$$

Special care must be taken in relation to the data used for parameter estimation. This is discussed in the following section.

B. Parameter Estimation

The quality of the estimated parameters in an identification process depends on the degree of excitation of the input signal during data acquisition. Given this, one typical idea is to use wide spectrum random or pseudorandom signals. One of these signals, which is easily generated in a computer, is the pseudorandom binary signal (PRBS) [6], [17]. This is a binary signal defined by a sequence of n bits that can assume two levels, namely, $-V$ and $+V$. The time duration of each bit, i.e., T_b , together with the number of bits n , determines the period of the signal. A typical PRBS is shown in Fig. 5(a).

In this paper, a PRBS representing the throttle of the vehicle was used to excite the car. To define the parameters of this PRBS, we first performed a step response test, which is shown in Fig. 4. In this test, we have applied a positive throttle level, and after 60 s, we have removed it. No brake is applied. Notice that there is an asymmetry on the velocity profile caused by the engine braking. This happens because, before $t = 60$ s,

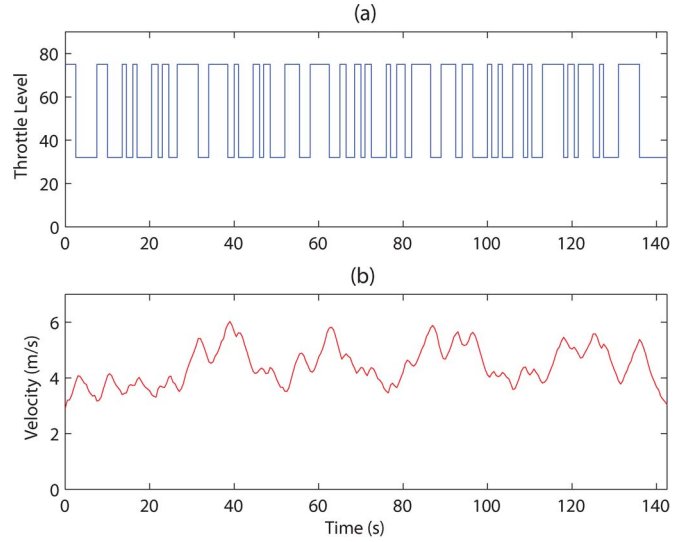


Fig. 5. Test with PRBS input. (a) Input: throttle level $-V = 32$ and $+V = 75$; e. (b) Output: velocity in meters per second.

the only force in the direction of the movement was due to the engine. After $t = 60$ s, both the engine and the vehicle's body create resistance forces contrary to the movement, causing the vehicle to decelerate faster than it accelerates. Thus, the smallest constant of time is due to the vehicle's deceleration. We have estimated a constant of time of about 5.5 s in this region. Constant T_b of the PRBS is generally chosen to stay between $1/10$ and $1/3$ of the smallest constant of time, i.e., T_{\min} [18]. Based on this, we chose T_b to be 1.2 s, which corresponds to the median of the interval between $1/10$ and $1/3$ of the constant of time. The number of bits of the PRBS was chosen to be $n = 12$, to guarantee a large period, generally larger than the total time of the test. For $-V$, which is the lowest level of the PRBS, we chose the minimum level accepted by the vehicle's ECU. In our experimental platform, this level happens to be represented by the number 32, in a scale that grows up to 200. For $+V$, we chose to use six distinct values, resulting in six different PRBSs. These values were 58, 66, 75, 83, 92, and 100.

During the tests, we collected the longitudinal velocity of the vehicle using a digital encoder connected to one of the rear wheels. The first tests were performed in first gear using a sampling frequency of 10 Hz. After the tests, the data were decimated so that the final sampling time is the sampling time of the discrete model (3), i.e., T_a . This sampling time was chosen to be $T_a = 0.5$ s, which is approximately one tenth of the smallest constant of time (5.5 s). As an example, Fig. 5 shows the results of a typical test.

The step response of the system is also useful in estimating the time delay. Observing the data in Fig. 4, we have estimated a time delay, i.e., τ , of 0.5 s. Since T_a is equal to τ , we can guarantee a single time delay between input (throttle level) and output (longitudinal velocity). Therefore, we can rewrite (5) as

$$y(k) = a_1 y(k-1) + a_2 y(k-2) + b_1 u(k-1) + b_2 u(k-2) + e(k). \quad (9)$$

TABLE I
ESTIMATED PARAMETERS (\hat{a}_1 , \hat{a}_2 , \hat{b}_1 , AND \hat{b}_2) FOR DIFFERENT THROTTLE LEVELS (+V)

	+V = 58	+V = 66	+V = 75	+V = 83	+V = 92	+V = 100
\hat{a}_1	0.82	1.14	1.31	1.37	1.41	1.45
\hat{a}_2	0.10	-0.21	-0.37	-0.43	-0.46	-0.50
\hat{b}_1	2.21×10^{-3}	2.37×10^{-3}	2.59×10^{-3}	3.31×10^{-3}	3.86×10^{-3}	3.98×10^{-3}
\hat{b}_2	4.27×10^{-3}	3.25×10^{-3}	2.83×10^{-3}	2.43×10^{-3}	1.59×10^{-3}	1.11×10^{-3}

To estimate a_1 , a_2 , b_1 , and b_2 using (8), we write these parameters and the collected data as

$$\hat{\theta} = \begin{bmatrix} \hat{a}_1 \\ \hat{a}_2 \\ \hat{b}_1 \\ \hat{b}_2 \end{bmatrix}$$

$$\Psi = \begin{bmatrix} y(2) & y(1) & u(2) & u(1) \\ y(3) & y(2) & u(3) & u(2) \\ \vdots & \vdots & \vdots & \vdots \\ y(N-2) & y(N-3) & u(N-2) & u(N-3) \\ y(N-1) & y(N-2) & u(N-1) & u(N-2) \end{bmatrix}$$

$$Y = \begin{bmatrix} y(3) \\ y(4) \\ \vdots \\ y(N-1) \\ y(N) \end{bmatrix}$$

where N is the number of data samples.

Table I shows the estimated parameters for the six different PRBSs used. As previously mentioned, each PRBS has a different maximum level, i.e., +V. Residual analyses were performed and, mostly, validate the estimated parameters. A small polarization (bias) of the parameters was observed in a few cases, what can be explained by the linear approximation of the engine transmission system.

Observing Table I, it can be noticed that the estimated parameters change as a function of the throttle level. Since we want a model that is valid for any throttle level, we chose to interpolate the parameters using quadratic functions. Fig. 6 shows plots of the resultant interpolations. Using the plots in Fig. 6, it is possible to define an ARX model to represent the car (in first gear and without brake) for each operation point (throttle level). The validation of this model is presented in the following section.

C. Model Validation

To validate the first-gear no-brake model identified in the previous section, a second data set was collected. This data set, different from the data in the previous section, simulates a human operator driving. Using this data set, we have simulated the model in two situations: 1) free simulation, where the actual input signal is used in the model to generate estimates of the longitudinal speed; and 2) prediction of n -steps ahead, where, in addition to the actual input, the measured values of velocity are used to estimate the velocity of the car n -samples in the future. The main difference between these two types of simulation is that in the free simulation, only velocity values estimated by the model are used to estimate the next value, whereas in the n -step-ahead prediction, one actual measured

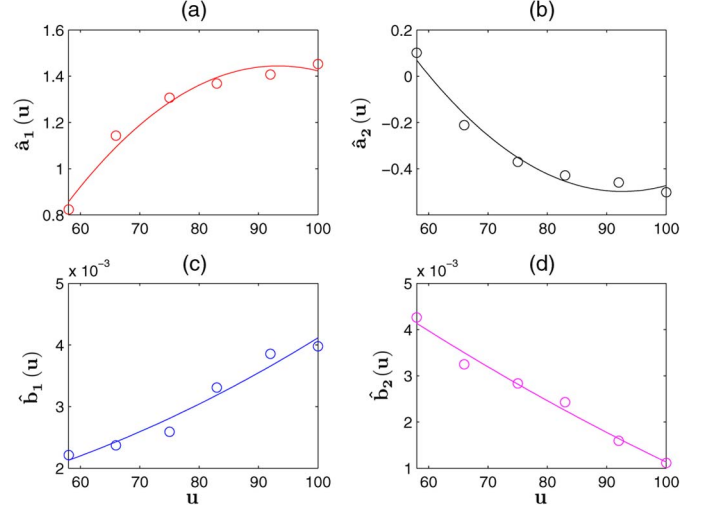


Fig. 6. Quadratic interpolation functions of the estimated parameters (a) $\hat{a}_1(u)$, (b) $\hat{a}_2(u)$, (c) $\hat{b}_1(u)$, and (d) $\hat{b}_2(u)$. For a given throttle level, four parameters are determined by quadratic interpolation.

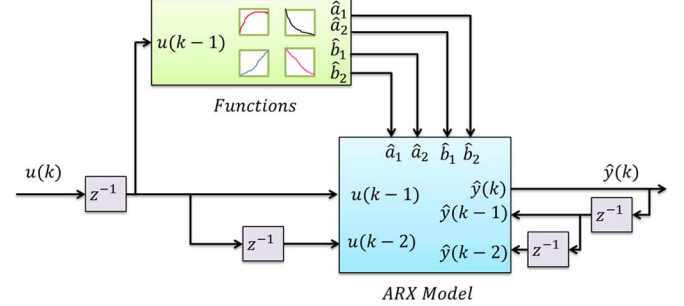


Fig. 7. Block diagram of the model's free simulation. Measured data are used as inputs to the model and also to define parameters \hat{a}_1 , \hat{a}_2 , \hat{b}_1 , and \hat{b}_2 . The estimated output is represented by \hat{y} .

velocity value is used every n steps in this estimation process. Notice that n -step-ahead prediction is a special case of free simulation: The model makes a free prediction for n samples, when it is reinitialized with measured data. The diagram in Fig. 7 illustrates the free simulation process. Throttle level (u) is used both as input for the ARX model and to compute the parameters, which are dependent on the operation point.

Fig. 8 shows a typical model validation using free simulation and five-step-ahead prediction. It can be observed that the model output presents a behavior that is very similar to the actual longitudinal velocity. For this specific figure, the root mean square error (RMSE) between the model output and the measured velocity for the free simulation is 0.884 m/s, and that for the five-step-ahead prediction is 0.360 m/s. It is worth mentioning that a manual adjustment of 35% in the input regressors \hat{b}_1 and \hat{b}_2 , related to the static gain of the system, was necessary in this step to obtain such errors. This

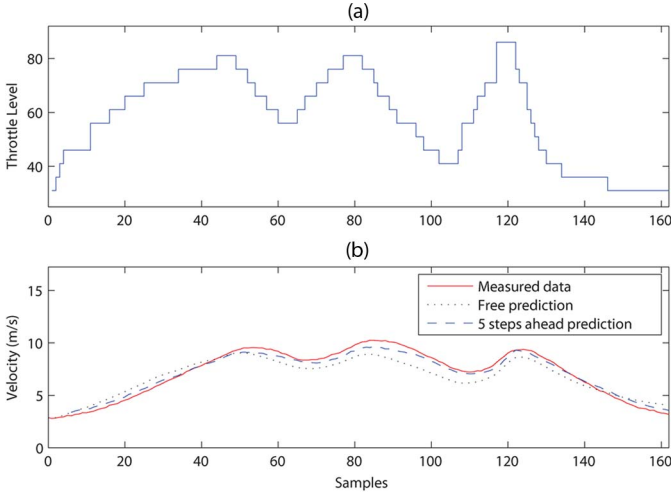


Fig. 8. Model validation. (a) Input: throttle level manually adjusted. (b) Output: measured velocity (m/s), free prediction and five-step-ahead prediction using the model.

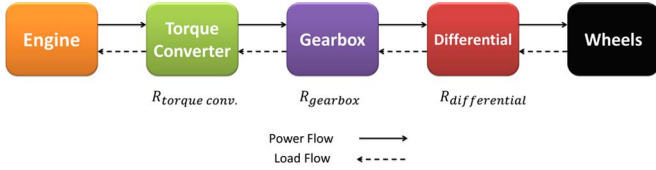


Fig. 9. Diagram of the gear ratio between the engine and wheels. Adapted from [3].

adjustment is usually necessary since PRBSs are not sufficient to estimate steady-state parameters, such as the static gain. In our model, this gain is basically responsible for converting throttle input into velocity. In the following section, we will show how another modification on this gain can be done so gear changes are included in the model.

D. Gear Change

The force generated by the engine is propagated to the wheels by means of the transmission system. This is composed of several parts, including the torque converter (present only on automatic gear cars), the transmission box, and the differential, as shown in Fig. 9. In this paper, it is assumed that the transmission relation of the torque converter, i.e., $R_{torqueconv}$, is mostly unitary, and that of the differential, i.e., $R_{differential}$, is constant, been related to the gear box, the only transmission relation that changes during the movement of the car.

This way, the gear box of the vehicle changes the velocity relation between the engine and the wheels using predefined rates. Except for the time of action of the torque converter, the gear shifting may be simply represented by a changing in the gain of force applied to the wheels. In this paper, we propose that this switching is done by the use of a variable, i.e., RTn , that multiplies the static gain of the first-gear model found in the previous sections. This variable is defined as the rate between the transmission relations of gear n (1, 2, 3, or 4) and the transmission relation of the first gear, i.e.,

$$RTn = \frac{Rn}{R1} \quad (10)$$

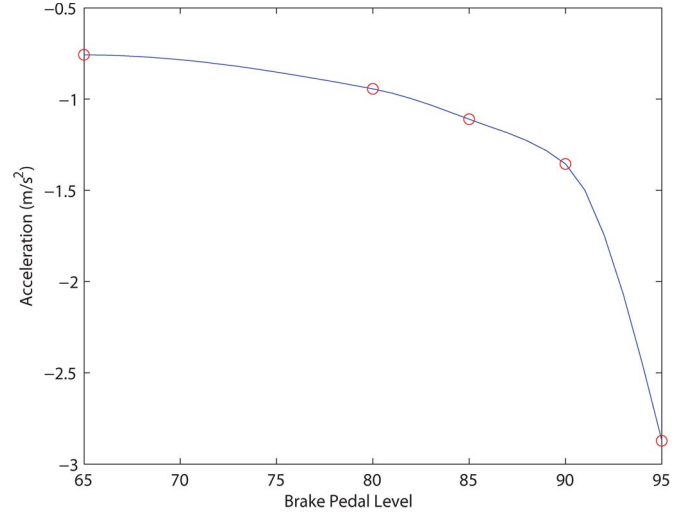


Fig. 10. Interpolation of longitudinal deceleration average levels versus the brake pedal level applied.

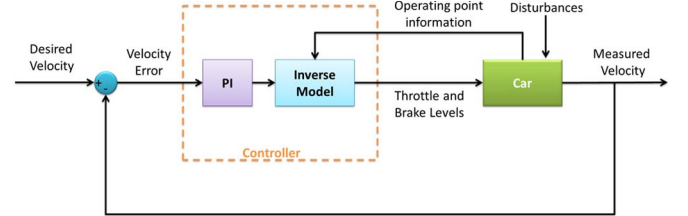


Fig. 11. Control strategy.

where $R1$ is the transmission relation between the wheels and the engine corresponding to the first gear, and Rn is the same variable corresponding to the other gears. These variables may be computed as

$$Rn = \frac{\omega_{wheels}}{\omega_{eng}} \quad (11)$$

where ω_{wheels} and ω_{eng} are the angular velocities of wheels and engine, respectively. Thus, the compensation of gear changes may be done using instantaneous measurements of ω_{wheels} and ω_{eng} . In our car, ω_{wheels} is measured using an encoder installed on one of the rear wheels, and ω_{eng} is measured by using the original inductive sensor of the vehicle. The validation of the model using this strategy to compensate for gear changes is omitted in this paper due to space constraints; however, in Section III, we show that it was successfully used to control the car. An extensive validation of the model can be seen in [19]. The following section shows our approach to introducing the brake input into the vehicle's model.

E. Braking System

So far, the only system input considered was the throttle level. Here, the brake level is also considered as an input for the vehicle. The identification of the car using brake as input is challenging once the application of PRBSs to the brake pedal would certainly cause the car to stop. Moreover, it is well known that the behavior of the braking system depends on

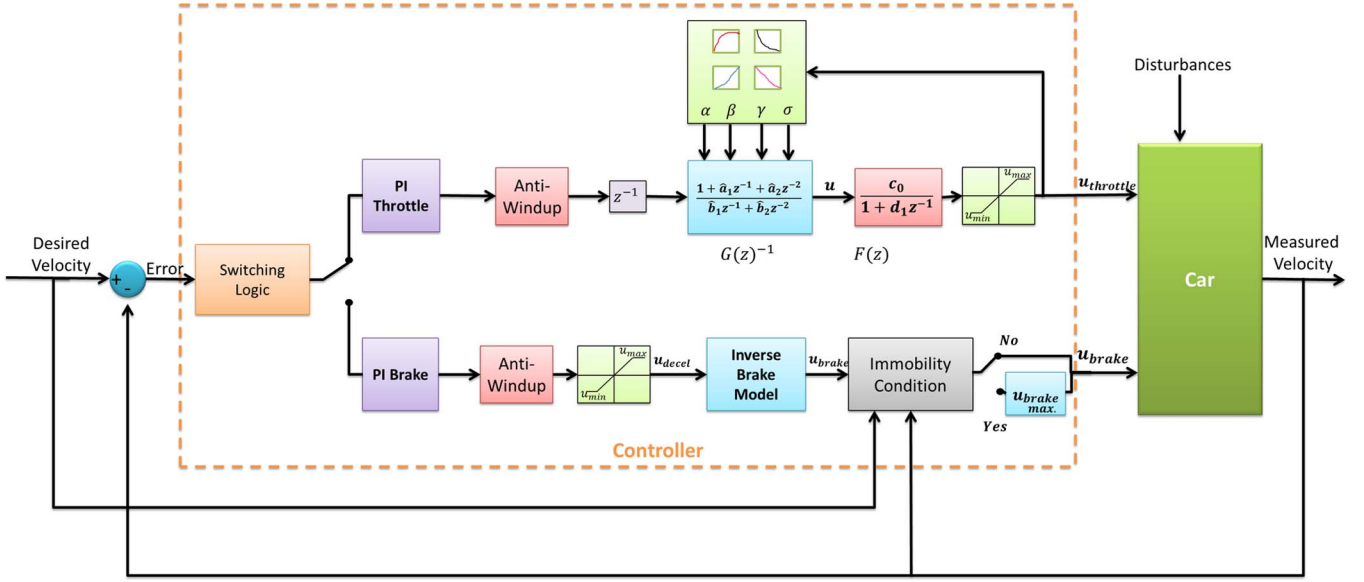


Fig. 12. Block diagram of the controller with throttle and brake control actions.

physical conditions such as temperature and friction, which are time-varying and nonlinear conditions, respectively.

The proposed solution is then to obtain a simple model that relates the brake level with the corresponding deceleration of the car. Step response tests were performed with our autonomous car moving on a flat surface and performing straight-line trajectories for five brake levels (65, 80, 85, 90, and 95) set as a percentage of the course of a linear actuator installed on the brake pedal. Once the deceleration was measured for these five levels, an interpolation, using a set of cubic polynomials, was performed to consider all other possible values. This interpolation is shown in Fig. 10. Despite the simplicity of this static model, it has been proved sufficient to aid in the control of the car, as we will show next.

III. VELOCITY CONTROL

In this paper, we intended to have a velocity controller that would allow the vehicle to move in urban scenarios, which are mostly comprised of low speeds and stationary situations. Moreover, as a requirement, we intended to constrain the vehicle's accelerations to values that are comfortable to its passengers and that are socially acceptable to pedestrians outside the vehicle [20]. In this paper, we aimed to limit the vehicle's acceleration to 2.5 m/s^2 , which is a comfort limit reported in the literature [21]. Moreover, we were seeking for a simple controller that could be easily implemented in the hardware of the vehicle, running in real time.

The model obtained in the previous section assumes that the vehicle has two inputs, which are throttle and braking levels. These variables must be manipulated by the controller. In this paper, similar to [3], we propose the use of the car's inverse model to compensate for the vehicle's nonlinear dynamics. The inverse model is used to generate throttle and braking levels from velocity values, thus making the system approximately linear. This technique is quite similar to the one adopted in computed torque control strategies used to control robotic

TABLE II
TECHNICAL SPECIFICATIONS OF THE VEHICLE USED IN THIS WORK

Mass	1467 [Kg]
Dimensions	$4199 \times 1709 \times 1431$ [mm]
Engine displacement	2400 [cm ³]
Number of cylinders	4
Number of valves	16
Fuel	Gasoline
Maximum power	104 @ 5000 [kW - rpm]
Maximum torque	200 @ 3000 [Nm - rpm]
Transmission type	automatic gearbox with torque converter
Gear ratios	3.67, 2.1, 1.39 and 1.00
Differential ratio	2.86
Wheels radius	317 [mm]

manipulators [22]. Once the nonlinearities are compensated, linear controllers may be used to control the system. The linear controller chosen in this work was a proportional–integral (PI) controller, justified by its versatile structure, robustness, and ease of tuning. The block diagram of the suggested control strategy is shown in Fig. 11.

Based on Fig. 11, the detailed block diagram that considers throttle and braking actions is shown in Fig. 12. Notice in this diagram that there is a switching logic responsible for choosing between throttle and braking loops. This is in consonance with the fact that it is not possible to use brake and throttle pedals at the same time. The switching criterion used in this work considers the required acceleration to achieve the desired velocity. To avoid successive switching associated to this logic, as suggested in [23], a hysteresis region was implemented. This way, when the required acceleration is positive, the throttle loop is chosen. When it is smaller than -0.25 m/s^2 , the brake loop is active. Between 0 and -0.25 m/s^2 , we have a hysteresis region.

In the throttle loop, the error between desired and measured velocities is computed and processed by a PI controller with proportional and integral gains empirically set to be $K_p = 1.0$ and $K_i = 0.1$, respectively. These gains were tuned to have a fast overdamped response that respects the acceleration limit of 2.5 m/s^2 , which is the limit of comfort for a human passenger. The resultant signal is analyzed by an antiwindup strategy [24].

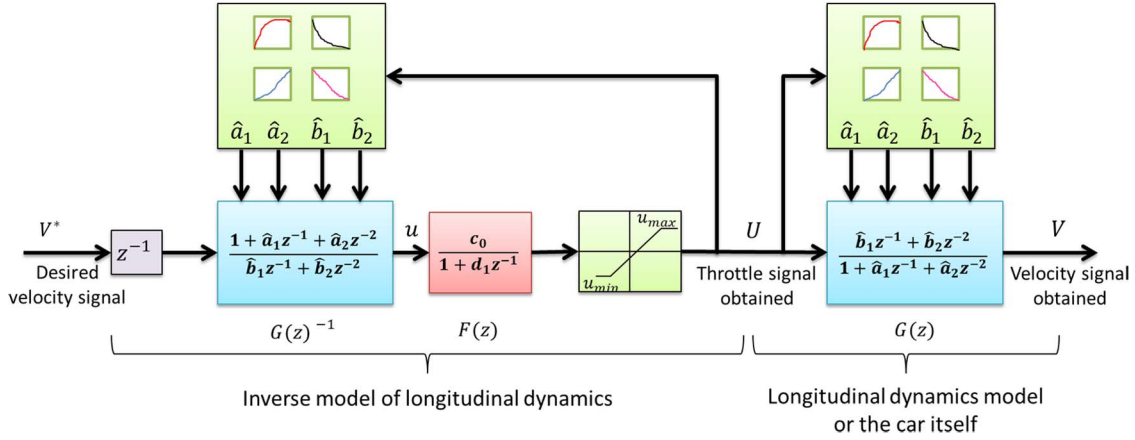


Fig. 13. Validation of the inverse model. The inverse model of the longitudinal dynamics is connected to the forward longitudinal model.

This strategy prevents the integral term of the controller from accumulating errors when the controller output is saturated or the braking loop is selected, i.e., the controller is unable to affect the controlled variable in a condition known as *winding up*. Without the antiwindup strategy, the integral term becomes too large in relation to the proportional term, thus making the system slow and oscillatory.

After the antiwindup strategy, the inverse throttle model ($G(z)^{-1}$) generates the throttle levels u_{acc} , which are filtered and applied to the car. Filter $F(z)$ is necessary because the vehicle's ECU does not permit high-frequency noise as inputs. Notice that this filter is not necessary during system identification. In that case, the signals were generated by a computer, whereas now, the control signal is corrupted by the noisy measured velocity. The parameters of the filter, i.e., $d_1 = -0.8$ and $c_0 = 0.2$, were experimentally chosen, observing the response of the vehicle to the signals. After being filtered, the throttle output must also be saturated to conform with the range allowed by the ECU.

The braking loop in Fig. 12 follows the same idea of the throttle one. A PI controller, with $K_p = 0.30$ and $K_i = 0.02$, is used to compensate for the uncertainties of the inverse model and eventual disturbances. Different from the throttle loop, however, is the presence of a block that guarantees null velocity when this is required. When it is necessary, this block stops the vehicle following a smooth velocity trajectory in a way that the passengers are not exposed to high accelerations. The following section presents the experimental results that validate the proposed controller.

IV. EXPERIMENTAL RESULTS

This section presents the results of experiments designed to evaluate the proposed control law and, consequently, the role of the identified model in the control loop. All experiments were performed using the autonomous car CADU (see Fig. 1), whose technical specifications are shown in Table II. It is important to mention that this information was not used during the development of the vehicle's model, which was estimated using experimental data only, as shown in Section II. The control level of the car's architecture is composed by sensors and actuators that communicate with a computer using an RS-485 bus and a

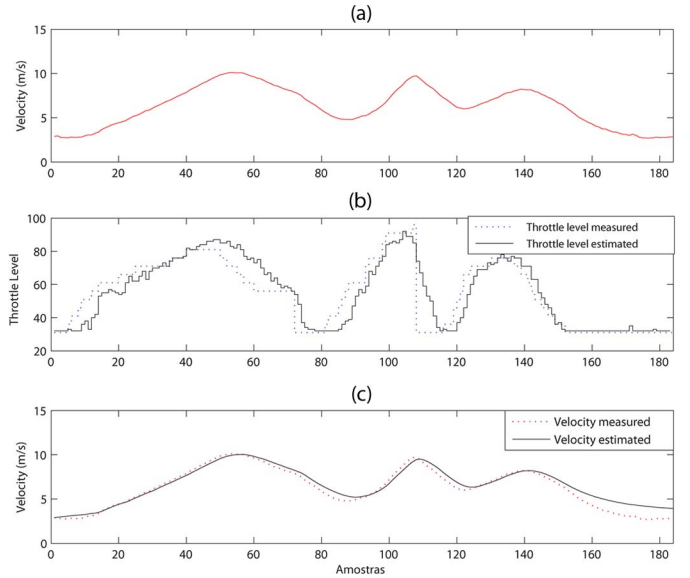


Fig. 14. Inverse model validation. (a) Experimental velocity data of an input–output set. (b) Comparison between experimental and throttle levels estimated by the inverse model. (c) Output of the forward model taking as input the throttle level estimated by the inverse model. The velocity in figure (a) is also shown in (c) for the sake of comparison.

ModBus protocol [25]. The computer is composed by a Mini-ITX standard board with a 1.6-GHz dual-core Atom processor and 2 GB of RAM. This computer runs a Linux operating system with the Real-Time Application Interface [26] patch, which guarantees determinism in time [27]. The system was set to operate at 10 Hz.

Our first experiment intended to validate the inverse model. For this purpose, the block diagram in Fig. 13 was used. The blocks responsible for computing the throttle level from the required velocity using the inverse model are shown on the left-hand side of this diagram, whereas on the right-hand side are the blocks that represent the forward model, which are replaced by the vehicle itself during actual experiments. Using this block diagram, a simulation was performed using a set of input–output data experimentally obtained from the real car. In Fig. 13, the signal obtained from the velocity sensor is used as the desired velocity, i.e., V^* [see Fig. 14(a)]. The estimated throttle input (U) is then compared with the actual throttle signal in Fig. 14(b). Fig. 14(c) compares the response of the

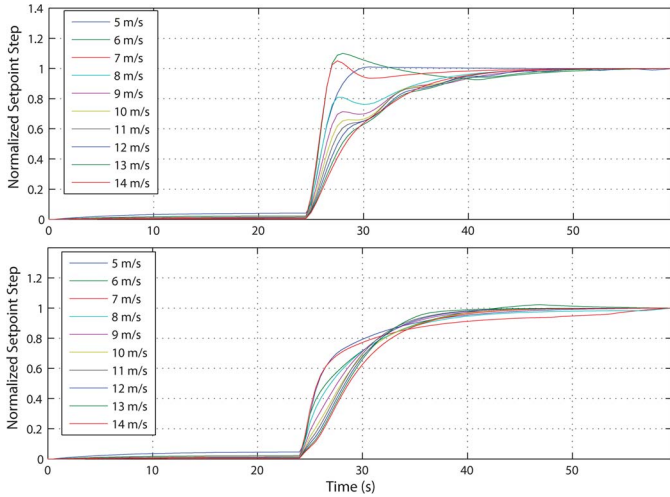


Fig. 15. Simulation of the controller (top figure) in the absence of the inverse model and (bottom figure) in the presence of the inverse model for different normalized setpoints.

forward model to the estimated throttle level with the actual velocity measured with the sensors. It is worth mentioning that those results were obtained using free simulation.

Notice in Fig. 14(b) that the estimated throttle signal is similar to the measurement signal, with some delay. This was expected, since a delay of at least two samples is observed in Fig. 13. In Fig. 14(c), which validates both the forward and inverse model, we conclude by the high level of similarity between the signals that the model can be used to compensate for the system nonlinearity.

To show the importance of this compensation in the control loop, we have simulated the controller without and with the inverse model. Without the inverse model, the loop becomes a simple PI controller. This PI controller was tuned in a specific point of operation to make the system critical damped. Fig. 15, top image presents the normalized step response of this controller for several operation points. Similar simulations with the controller with the inverse model are shown in Fig. 15, bottom image. Notice that, due to parameter variation of the model (which is a consequence of the nonlinearity of the system), the response of the controller without the inverse model varies as a function of the required velocity. On the other hand, the step response of the controller with the inverse model remains with almost the same behavior for all setpoints, as expected.

Fig. 16 presents the results of real-world experiments where the vehicle was kept in first gear and followed a linear path on a flat surface. It is possible to see that the controller followed the reference. In this experiment, brake actuation was used only when it was necessary to decelerate the car, as shown in the last plot in Fig. 16. By differentiating the velocity profile in Fig. 16, it was possible to see that the vehicle's acceleration ranges from -2.45 to 1.67 m/s^2 , which respects the limit of comfort sensation of 2.5 m/s^2 .

In Fig. 17, we show another experiment where the reference was kept constant at 20 km/h , but different from the previous experiment, the car was moving on a circuit with slopes and that included two laps in a roundabout. The two bottom graphs in Fig. 17 show changes in the car inclination and orientation

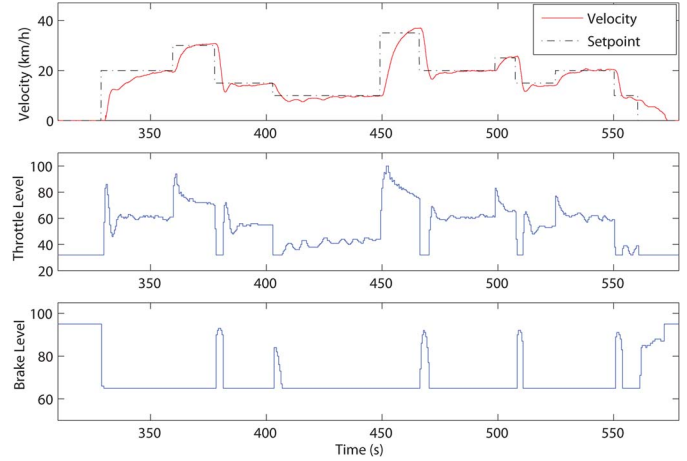


Fig. 16. Validation of the controller including brake input.

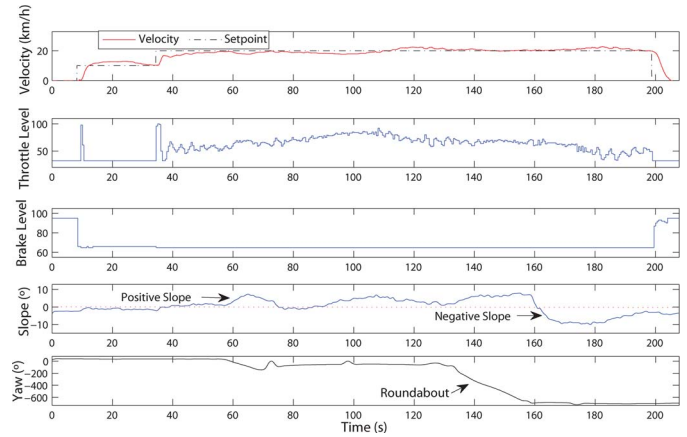


Fig. 17. Experimental validation of the controller in cases of runway slopes and changes in steering (yaw angle).

collected using an inertial measurement unit inside the vehicle. It can be observed that in regions of positive slopes, the velocity initially decreases, but the system is able to react, increasing the throttle level to reject the disturbance. The opposite happened when a negative slope is observed. This experiment also showed that the controller was able to reject another external disturbance, which is due to the steering of the front wheels on the roundabout. The RMSE between the setpoint and the measured velocity during this experiment was 1.35 km/h , what corresponds to 6.54% of the setpoint. This error seems to be reasonable, if the disturbances imposed on the vehicle are considered.

Similar tests were performed when the car was allowed to change gears. A typical result for this situation is shown in Fig. 18. In this figure, apart from the output velocity and the vehicle's inputs, we also show the engine speed that, along with the speed of the car, is used to estimate the gear ratio [see (11)]. This estimate is shown in the last plot in the figure. Observe that, due to the torque converter, the estimate assumes continuous values and not only a set of discrete values (one for each gear) as one would expect. Remember that the ratio estimate is used to adapt the static gain of the car's mathematical model. The results in Fig. 18 shows that, even in keeping the gains tuned for first gear, the controller is able to control

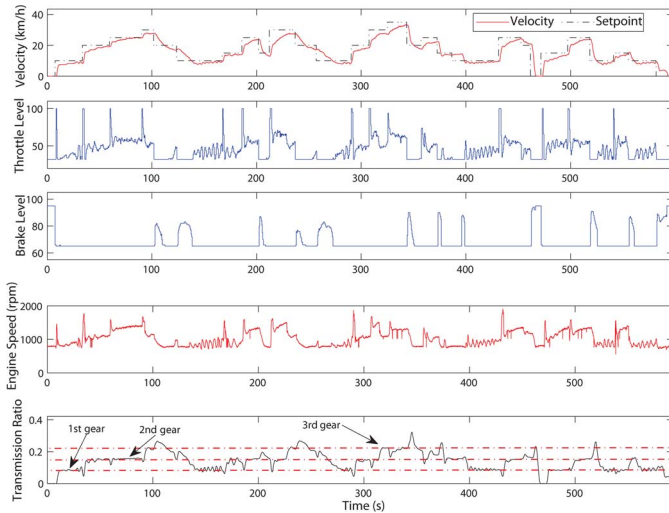


Fig. 18. Experimental validation of the controller in the drive move (gear change is allowed).

the car. However, it is possible to notice some degradation on its performance in relation to first-gear results. While the vehicle's acceleration ranges from -2.15 to 1.86 m/s², the average rising time grown from 4.6 s in first gear to about 9.5 s when gear change is allowed. This probably happens because, even with the adaptation of the gain, the identified model cannot well represent the changes in the vehicle dynamics during gear shifting. A more specific nonlinear model that includes the torque converter dynamics would be necessary in these situations.

V. CONCLUSION

This paper has proposed and implemented a methodology to determine the longitudinal model of an autonomous car and a strategy to use this model to improve the vehicle's velocity control. The controller uses the inverse model of the vehicle to compensate for the nonlinear dynamics caused by the engine and the transmission system of the car. The modeling methodology and the control system were tested in an actual autonomous car.

The main characteristics of the modeling approach are as follows. 1) It is data based and does not depend on any prior knowledge about the vehicle. 2) It can be applied to any automated automobile for which it is possible to actuate throttle and braking and measure the speed. Engine rotations are necessary if gear changes are to be considered. Accelerations and other variables do not need to be measured. 3) The resultant model is a simple difference equation that can be easily computed in the vehicle's embedded hardware.

The proposed controller relies on a simple scheme to select between two control loops, i.e., one that manipulates throttle and another that actuates the braking system. Each loop is basically composed by a PI controller in cascade with the inverse dynamic model of the vehicle, responsible for linearizing the system. This simple control architecture was implemented on a real-time system and was able to control the vehicle in uneven and nonflat surfaces with a good performance. One important observation is that, different from commercial CC systems, the

implemented controller was used to drive the car at low speeds that range from 0 to 40 km/h. Within this range, the controller was also able to keep the acceleration of the vehicle close to 2.5 m/s², which is the maximum level at which human passengers feel comfortable. This suggests that the controller can be used as the inner loop of autonomous parking and ACC systems, applications that require low and varying speeds with passenger comfort.

ACKNOWLEDGMENT

The authors would like to thank T. Arruda, M. T. Pires, G. Castro, J. H. Costa, B. Freitas, and F. Pujatti for their help during the development of this work. They would also like to thank the anonymous reviewers for their important comments and suggestions.

REFERENCES

- [1] D. A. Lima and G. A. S. Pereira, "Navigation of an autonomous car using vector fields and the dynamic window approach," *J. Control, Autom. Elect. Syst.*, vol. 24, no. 1/2, pp. 106–116, Apr. 2013.
- [2] S. Thanok and M. Parnichkul, "Adaptive cruise control of a passenger car using hybrid of sliding mode control and fuzzy logic control," in *Proc. 8th Int. Conf. Autom. Eng.*, Apr. 2012, pp. 34–39.
- [3] R. Rajamani, *Vehicle Dynamics and Control*, 2nd ed. Minneapolis, MN, USA: Springer-Verlag, 2012.
- [4] A. Bacha *et al.*, "Odin: Team VictorTango's entry in the DARPA Urban Challenge," *J. Field Robot.*, vol. 25, no. 8, pp. 467–492, Aug. 2008.
- [5] F. W. Rauskolb *et al.*, "Caroline: An autonomously driving vehicle for urban environments," *J. Field Robot.*, vol. 25, no. 9, pp. 674–724, Sep. 2008.
- [6] L. Ljung, *System Identification—Theory for the User*. Upper Saddle River, NJ, USA: Prentice-Hall, 1987.
- [7] K. Hunt, J. Kalkkuhl, H. Fritz, and T. Johansen, "Constructive empirical modelling of longitudinal vehicle dynamics using local model networks," *Control Eng. Pract.*, vol. 4, no. 2, pp. 167–178, Feb. 1996.
- [8] L. Nouveliere and S. Mammar, "Experimental vehicle longitudinal control using second order sliding modes," in *Proc. Amer. Control Conf.*, Jun. 2003, vol. 6, pp. 4705–4710.
- [9] H. Liang, K. T. Chong, T. S. No, and S.-Y. Yi, "Vehicle longitudinal brake control using variable parameter sliding control," *Control Eng. Pract.*, vol. 11, no. 4, pp. 403–411, Apr. 2003.
- [10] X.-Y. Lu and J. Hedrick, "Longitudinal control design and experiment for heavy-duty trucks," in *Proc. Amer. Control Conf.*, Jun. 2003, vol. 1, pp. 36–41.
- [11] A. Girard, S. Spry, and J. Hedrick, "Intelligent cruise control applications: Real-time embedded hybrid control software," *IEEE Robot. Autom. Mag.*, vol. 12, no. 1, pp. 22–28, Mar. 2005.
- [12] M. Montemerlo *et al.*, "Junior: The Stanford entry in the Urban Challenge," *J. Field Robot.*, vol. 25, no. 9, pp. 569–597, Sep. 2008.
- [13] B. J. Patz, Y. Papelis, R. Pillat, G. Stein, and D. Harper, "A practical approach to robotic design for the DARPA Urban Challenge," *J. Field Robot.*, vol. 25, no. 8, pp. 528–566, Aug. 2008.
- [14] J. Bohren *et al.*, "Little Ben: The Ben Franklin racing team's entry in the 2007 DARPA Urban Challenge," *J. Field Robot.*, vol. 25, no. 9, pp. 598–614, Sep. 2008.
- [15] K. Osman, M. Rahmat, and M. Ahmad, "Modelling and controller design for a cruise control system," in *Proc. 5th Int. CSPA*, Mar. 2009, pp. 254–258.
- [16] C. L. Phillips and H. T. Nagle, *Digital Control System Analysis and Design*, 3rd ed. Upper Saddle River, NJ, USA: Prentice-Hall, 1995.
- [17] T. Söderström and P. Stoica, *System Identification*: Prentice-Hall, 1989.
- [18] L. A. Aguirre, *Introdução à Identificação de Sistemas: Técnicas Lineares e Não-Lineares Aplicadas a Sistemas Reais*. Belo Horizonte, MG, Brazil: Editora UFMG, 2007, 3a edição.
- [19] J. E. A. Dias, "Modelagem longitudinal e controle de velocidade de um carro autônomo," M.S. thesis, Universidade Federal de Minas Gerais, Belo Horizonte, MG, Brazil, 2013.
- [20] H. I. Christensen and E. Pacchierotti, "Embodied social interaction for robots," in *Proc. Conv. Soc. Study Artif. Intell. Simul. Behav.*, Hertfordshire, U.K., 2005, pp. 40–45.

- [21] K. Yi and J. Chung, "Nonlinear brake control for vehicle CW/CA systems," *IEEE/ASME Trans. Mechatron.*, vol. 6, no. 1, pp. 17–25, Mar. 2001.
- [22] J. J. Craig, *Introduction to Robotics: Mechanics and Control*, 2nd ed. Boston, MA, USA: Addison-Wesley, 1989.
- [23] J. Hedrick, J. Gerdes, D. Maciucă, and D. Swaroop, "Brake system modeling, control and integrated brake/throttle switching: Phase I," California PATH Research Report, UC Berkeley, Berkeley, CA, USA, 1997.
- [24] K. J. Aström and T. Hägglund, *Advanced PID Control*. Research Triangle Park, NC, USA: ISA, 2005.
- [25] Modbus, Accessed in 02/02/2014 Modbus Over Serial Line Specification. [Online]. Available: <http://www.modbus.org/>, Accessed in 02/02/2014
- [26] RTAI, Accessed in 02/02/2014 RTAI—Real Time Application Interface. [Online]. Available: <https://www.rtai.org/>, Accessed in 02/02/2014
- [27] A. Barbalace *et al.*, "Performance comparison of VxWorks, Linux, RTAI, Xenomai in a hard real-time application," *IEEE Trans. Nucl. Sci.*, vol. 55, no. 1, pp. 435–439, Feb. 2008.



Jullierme Emiliano Alves Dias received the B.S. degree in electrical engineering from Federal University of Viçosa (UFV), Viçosa, Brazil, in 2010 and the M.S. degree in electrical engineering from Federal University of Minas Gerais (UFMG), Belo Horizonte, Brazil, in 2013.

He is a Researcher with the Department of Research and Innovation of Neocontrol Soluções em Automação S/A, supported by the Program for Fixing Human Resources of CNPq/Brazil. His research interests include embedded systems, electronic in-

strumentation, mobile robotics, automation, modeling, and control of dynamical systems.



Guilherme Augusto Silva Pereira (SM'13) received the B.S. and M.S. degrees in electrical engineering and the Ph.D. degree in computer science from Federal University of Minas Gerais (UFMG), Belo Horizonte, Brazil, in 1998, 2000, and 2003, respectively.

From November 2000 to May 2003 he was a Visiting Scientist with the General Robotics, Automation, Sensing, and Perception Laboratory, University of Pennsylvania, Philadelphia, PA, USA. Since July 2004 he has been an Associate Professor with the Department of Electrical Engineering, Federal University of Minas Gerais, where he is the Director of the Computer Systems and Robotics Laboratory, which is one of the laboratories that comprise the Group for Research and Development of Autonomous Vehicles, UFMG. His research interests include cooperative robotics, robot navigation, autonomous vehicle development, computer vision, and distributed sensing.

Dr. Pereira is a member of the Sociedade Brasileira de Automática. He received the Gold Medal Award from the Engineering School of UFMG for garnering first place among the electrical engineering students in 1998.



Reinaldo Martinez Palhares received the B.E. degree from Federal University of Goiás, Goiânia, Brazil, in 1992 and the M.S. and Ph.D. degrees from University of Campinas, Campinas, Brazil, in 1995 and 1998, respectively, all in electrical engineering.

From 1998 to 2002 he was with PUC/Minas, Brazil. Since 2002 he has been an Associate Professor with Federal University of Minas Gerais, Belo Horizonte, Brazil. In a broad sense, his main research interests include robust linear/nonlinear control/filtering theory, time delays, fault detection and isolation, fuzzy theory, soft computing, and multicriteria optimization theory and applications.

Dr. Palhares is a member of the Editorial Board of *Journal of Applied Mathematics* and of the Review Board of *Applied Intelligence*.