IDENTIFICATION OF QUASI-LINEAR DYNAMIC MODEL WITH DEAD ZONE FOR MOBILE ROBOT WITH DIFFERENTIAL DRIVE

Ellon P. Mendes*, Adelardo A. D. Medeiros*

*Department of Computer Engineering and Automation
Universidade Federal do Rio Grande do Norte
Natal, RN, Brazil

Emails: [ellonpaiva,adelardo]@dca.ufrn.br

***Abstract*—This note deals with the identification of a mobile robot with differential drive. The robot system is modeled by two MISO Hammerstein systems with input dead zones. The robot dynamic model is based on the traveled distance increment instead of the robot coordinates, making the model linear and allowing the application of classical methods of identification. Both parameters of linear and nonlinear blocks are estimated simultaneously through application of recursive least squares. Our main contribution is to use simple identification techniques to recursively estimate the linear and non-linear parameters of the robot system, that may vary over the time. The feasibility of the method is demonstrated by examples using both simulated and real robot.**

I. INTRODUCTION

The system identification deals with a problem of establishing a formulae governing a system phenomenon. While linear identification has been explored to the level that most linear identification problems can be solved with standard tools, nonlinear system identifications still require a superior effort to be achieved. Between the systems studied in the literature, the *Hammerstein* system has attracted significant interest due to its simplicity: it merely cascades a static nonlinear block followed by a linear dynamic block [1]. The Hammerstein system represent well various systems with input non-linearity, like the dead zones present in mobile robots.

The dead zone is considered as the operating range of input that produces no response in the dynamics of the system output [2]. In a mobile robot drove by electrical actuators, the dead zone appears usually due the friction present in the movement axis [3]. Besides the non-linearity present in the actuators, mobile robots usually have other nonlinear characteristics, like the nonholonomic restrictions present in the mobile robot with differential drive considered in this work, which increase the complexity of the model to be identified. A linear model for a mobile robot with differential drive was presented in [4], achieved by changing the system output signals, and allowing the application of linear identification techniques, but does not take into account the input non-linearity of the system.

This work presents a identification scheme based in the linear model described in *Guerra et al* [4] combined with the key term separation principle used in *Vörös et al* [5]. The robot system is represented by two MISO (Multiple-Input Single-Output) Hammerstein sub-systems, each one identified separately. The scheme allow the identification of both linear and nonlinear parameters simultaneously through direct application of known recursive least squares method, supplemented with the estimation of the internal variables.

The rest of the paper is organized as follows. Section II describes the robot and its modeling. Section III shows the strategy to estimate the traveled distance of the robot between two samples. Section IV presents the model of the dead zone considered in this work. Section V shows the equations and internal variables used by the recursive identification algorithm while Section VI describes the start up procedures used to initialize method. Section VII shows the results of the identification applied to a simulated robot. Finally, Section VIII presents the conclusions and the future perspectives.

II. ROBOT MODEL

Fig. 1 presents a diagram of the kind of robot we consider in this work. The robot has two wheels, driven by two independent electric motors. The wheels are placed at each side of the robot, in such a position that their rotation axis are coincident. The robot configuration is represented by the position of the center of the axis between the two wheels in the Cartesian space (x and y) and by its orientation θ (angle between the vector of the robot orientation and the abscissas axis).

A. Cinematic Model

The cinematic model describes the relations between the derivatives of robot position and orientation and the robot linear and angular speeds, v and w, without taking into account the causes of its movement. The cinematic model

978-0-7695-4231-7/10 $26.00 © 2010 IEEE
DOI 10.1109/LARS.2010.36

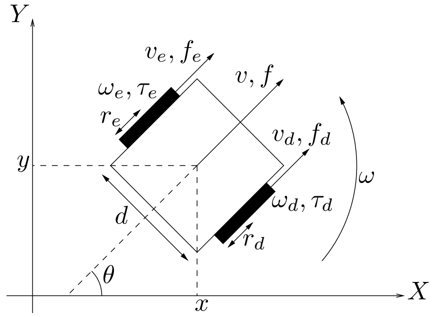

Fig. 1. Mobile Robot

for the considered robot in this work is presented (1).

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} v \\ w \end{bmatrix} \tag{1}$$

The equation (1) models the non-holonomic restrictions of the robot.

B. Dynamic Model

The dynamic model is derived from the physics laws that govern the several robot subsystems, including the actuator dynamics (electric and mechanical characteristics of the motors), friction and robot dynamics (movement equations). The derivation of this model for a small mobile robot was presented by [6].

For most robots, the modelling process generates a second-order model expressed by:

$$\mathbf{Ku} = \mathbf{M\dot{v}} + \mathbf{Bv} \tag{2}$$

where $\mathbf{v} = \begin{bmatrix} v & w \end{bmatrix}^T$ represents the robot linear and angular speeds, $\mathbf{u} = \begin{bmatrix} u_r & u_l \end{bmatrix}^T$ contains the input signals (usually armature tensions) applied to the right and left motors, $\mathbf{K}$ is the matrix which transforms the electrical signals $\mathbf{u}$ into forces to be generated by the robot wheels, $\mathbf{M}$ is the generalized inertia matrix and $\mathbf{B}$ is the generalized damping matrix, which includes terms of viscous friction and electric resistance.

C. Equivalent Linear Model

The complete robot model, obtained from the union of equations (1) and (2), can be represented by the following state equation:

$$\begin{bmatrix} \dot{v} \\ \dot{\omega} \\ \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} -\mathbf{M^{-1}B} & \vdots\ \mathbf{0} \\ \cdots\cdots\cdots & \cdots\cdots \\ \begin{matrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 0 & 1 \end{matrix} & \vdots\ \mathbf{0} \end{bmatrix} \begin{bmatrix} v \\ \omega \\ x \\ y \\ \theta \end{bmatrix} + \begin{bmatrix} \mathbf{M^{-1}K} \\ \cdots\cdots \\ \mathbf{0} \end{bmatrix} \begin{bmatrix} u_r \\ u_l \end{bmatrix} \tag{3}$$

$$\mathbf{y} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \\ x \\ y \\ \theta \end{bmatrix}$$

where the system output $\mathbf{y} = [\ x \quad y \quad \theta\ ]^T$ corresponds to the robot configuration.

To allow the application of linear discretization techniques, we will rewrite the system equations into a linear representation of the robot dynamic behavior. To attain this objective, we need to change the set of state variables: the robot's configuration, given by its position x and y and its orientation θ, will be described in terms of the robot linear displacement l and the robot orientation θ.

$$\begin{bmatrix} \dot{v} \\ \dot{\omega} \\ \dot{l} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} -\mathbf{M^{-1}B} & \vdots\ \mathbf{0} \\ \cdots\cdots\cdots & \cdots \\ \begin{matrix} 1 & 0 \\ 0 & 1 \end{matrix} & \vdots\ \mathbf{0} \end{bmatrix} \begin{bmatrix} v \\ \omega \\ l \\ \theta \end{bmatrix} + \begin{bmatrix} \mathbf{M^{-1}K} \\ \cdots\cdots \\ \mathbf{0} \end{bmatrix} \begin{bmatrix} u_r \\ u_l \end{bmatrix} \tag{4}$$

$$\mathbf{z} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \\ l \\ \theta \end{bmatrix}$$

The new state equation (4) has the same dynamic parameters than the original state equation (3) but a new system output $\mathbf{z} = [\ l \quad \theta\ ]^T$. We observe that the linear equivalent model was obtained without any simplification assumption. In this way, the linear model is an exact equivalent representation of the original nonlinear representation in (3). However, the "l" variable is not measurable: this aspect will be explored later.

D. Model Discretization

To allow the application of the classical estimation techniques, it is usual deriving a discrete transfer function equivalent to the model in (4). The first step is the transformation of the state space form into a continuous transfer matrix:

$$\begin{bmatrix} L(s) \\ \theta(s) \end{bmatrix} = \begin{bmatrix} G_{11}(s) & G_{12}(s) \\ G_{21}(s) & G_{22}(s) \end{bmatrix} \cdot \begin{bmatrix} U_r(s) \\ U_l(s) \end{bmatrix} \tag{5}$$

where each term $G_{ij}(s)$ has the following structure:

$$G_{ij}(s) = \frac{N_{ij}(s)}{D(s)} = \frac{\beta_{ij}(s-n_i)}{s(s-p_1)(s-p_2)}$$

Analysing the robot model, we see that $G_{ij}(s)$ has one linear and one angular pole. Then, if the robot has negligible asymmetries, the zero of each $G_{ij}(s)$ will be canceled with one of the poles. Thus, we can make the follow simplification:

$$G_{ij}(s) = \frac{\beta_{ij}(s-n_i)}{s(s-p_1)(s-p_2)} \simeq \begin{cases} \frac{\beta_{ij}}{s(s-p_l)} & \text{if } i=1 \\ \frac{\beta_{ij}}{s(s-p_\theta)} & \text{if } i=2 \end{cases}$$

In the second step, we calculate the four discrete transfer functions $G_{ij}(z)$:

$$\begin{bmatrix} L(z) \\ \theta(z) \end{bmatrix} = \begin{bmatrix} G_{11}(z) & G_{12}(z) \\ G_{21}(z) & G_{22}(z) \end{bmatrix} \cdot \begin{bmatrix} U_r(z) \\ U_l(z) \end{bmatrix} \tag{6}$$

where each term $G_{ij}(z)$ has the following structure:

$$G_{ij}(z) = \frac{N_{ij}(z)}{D_i(z)} = \frac{\delta_{ij}z - \epsilon_{ij}}{(z-1)(z-\alpha_i)}$$

To obtain a parametrization appropriated to the estimation, the transfer functions are converted to equivalent difference equations:

$$\begin{aligned} \Delta l_k =& \alpha_1 \Delta l_{k-1} && (7) \\ &+ \delta_{11} u_{r,k-1} - \epsilon_{11} u_{r,k-2} \\ &+ \delta_{12} u_{l,k-1} - \epsilon_{12} u_{l,k-2} \\ \Delta\theta_k =& \alpha_2 \Delta\theta_{k-1} && (8) \\ &+ \delta_{21} u_{r,k-1} - \epsilon_{21} u_{r,k-2} \\ &+ \delta_{22} u_{l,k-1} + \epsilon_{22} u_{l,k-2} \end{aligned}$$

where $\Delta l_k = l_k - l_{k-1}$ is the linear distance traveled in a sampling period, $\Delta\theta_k = \theta_k - \theta_{k-1}$ is the angular increment in the robot's orientation in the same period, and $u_{r,k}$,$u_{l,k}$ are the plant input signals.

III. TRAVELED DISTANCE COMPUTATION

The variable Δl is not measurable. In spite of this, we can use heuristics to calculate a plausible value for this variable, $\widetilde{\Delta l}$. The methodology developed to calculate $\widetilde{\Delta l}$ consists of the following steps:

1) calculation of the direction of the movement, i.e. the sign of $\widetilde{\Delta l}$
2) calculation of the path length $|\widetilde{\Delta l}|$

A. Computation of the Movement Direction

To obtain the sign of the traveled distance, i.e. knowing if the robot moved forward or backward, we calculate the value of ${}^0x_1$, the x coordinate of the current configuration point (x_1, y_1) calculated with respect to the reference frame attached to the previous robot position (x_0, y_0), as indicated in the Fig. 2. If ${}^0x_1$ is positive, the robot will have moved forward and $\widetilde{\Delta l} > 0$. Otherwise, the robot will have moved backward and $\widetilde{\Delta l} < 0$.

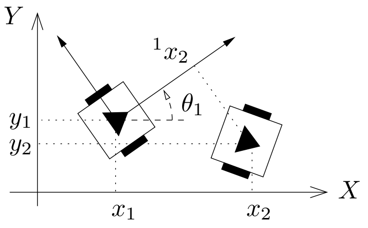

Fig. 2. Diagram for calculating the sign of $\widetilde{\Delta l}$

A simple coordinate transformation allows the calculation of ${}^0x_1$:

$$\begin{aligned} {}^0x_1 = x_1\cos\theta_0 - x_0\cos\theta_0 + \\ + y_1\sin\theta_0 - y_0\sin\theta_0 \end{aligned} \tag{9}$$

B. Path Length Computation

It is possible to obtain with reasonable precision the robot configuration (x, y, θ) in two consecutive sampling instants, but we cannot be sure about the path traveled by the robot between the first and the second configuration. However, a estimated value of $|\widetilde{\Delta l}|$ can be calculated using a bezier curve of third degree [7] between consecutive samples, where the initial orientation of the curve is equal to the robot orientation in the previous sample and the final orientation is equal to the actual robot orientation. Each control point of the curve is located in a straight line formed by the position and angle of the robot in each sample, with a distance of the robot position equal of a third of the euclidean distance between the samples, as shown in Fig. 3. The estimated value $|\widetilde{\Delta l}|$ can be achieved by numerical integration of the bezier curve. Note that the control points should agree with the movement direction of the robot (forward or backward).

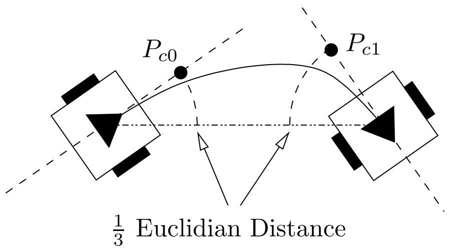

Fig. 3. Diagram for $|\widetilde{\Delta l}|$ computation

IV. DEAD ZONE MODEL

The system non-linearity is modeled by two symmetric dead zones, one for each system input. The dead zone is described by Fig. 4 and can be written as:

$$m(t) = \begin{cases} u(t) - D & \text{if } u(t) > D \\ u(t) + D & \text{if } u(t) < -D \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

where $D > 0$ is the dead zone constant and dead zone slopes have unitary gain (its always possible to fix one parameter in the Hammerstein model). The equation (10) can be rewrote as:

$$\begin{aligned} m(t) =& (u(t) - D)h(D - u(t)) \\ &+ (u(t) + D)h(u(t) + D) \end{aligned} \tag{11}$$

where $h(p)$ is a switching function defined as:

$$h(p) = \begin{cases} 0 & \text{if } p \geq 0 \\ 1 & \text{if } p < 1 \end{cases} \tag{12}$$

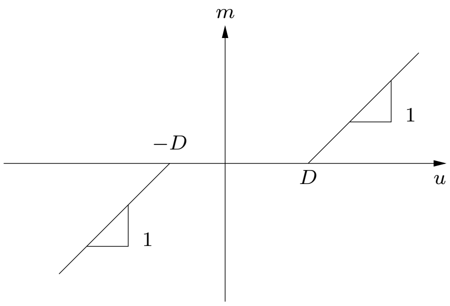

Fig. 4. Dead Zone Type Non-linearity

V. Model Identification

The MIMO robot system is represented by three blocks as shown in Fig. 5. The two input blocks are the dead zones while the last block represents the robot dynamic linear model. This MIMO system can be separated into the two MISO systems of the Fig. 6.

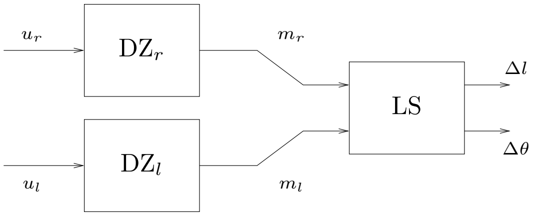

Fig. 5. MIMO Robot System

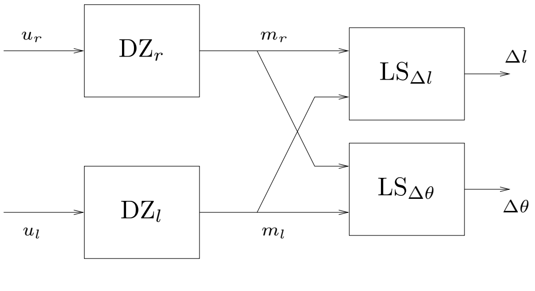

Fig. 6. Robot System as two MISO Systems.

Based on the Fig. 6, we can use (7) and (8) to obtain:

$$\begin{aligned} \Delta l_k =& \alpha_1 \Delta l_{k-1} \\ &+ \delta_{11} m_{r,k-1} - \epsilon_{11} m_{r,k-2} \\ &+ \delta_{12} m_{l,k-1} - \epsilon_{12} m_{l,k-2} \end{aligned} \tag{13}$$

$$\begin{aligned} \Delta \theta_k =& \alpha_2 \Delta \theta_{k-1} \\ &+ \delta_{21} m_{r,k-1} - \epsilon_{21} m_{r,k-2} \\ &+ \delta_{22} m_{l,k-1} + \epsilon_{22} m_{l,k-2} \end{aligned} \tag{14}$$

and making a partial-substitution of (11) in (13) and (14) we have:

$$\begin{aligned} \Delta l_k =& \alpha_1 \Delta l_{k-1} \\ &+ \delta_{11} u_{r,k-1} \left[h(D_r - u_{r,k-1}) + h(u_{r,k-1} + D_r) \right] \\ &+ D_r' \delta_{11} f_{r,k-1} - \epsilon_{11} m_{r,k-2} \\ &+ \delta_{12} u_{l,k-1} \left[h(D_l - u_{l,k-1}) + h(u_{l,k-1} + D_l) \right] \\ &+ D_l' \delta_{12} f_{l,k-1} - \epsilon_{12} m_{l,k-2} \end{aligned} \tag{15}$$

$$\begin{aligned} \Delta \theta_k =& \alpha_2 \Delta \theta_{k-1} \\ &+ \delta_{21} u_{r,k-1} \left[h(D_r - u_{r,k-1}) + h(u_{r,k-1} + D_r) \right] \\ &+ D_r'' \delta_{21} f_{r,k-1} - \epsilon_{21} m_{r,k-2} \\ &+ \delta_{22} u_{l,k-1} \left[h(D_l - u_{l,k-1}) + h(u_{l,k-1} + D_l) \right] \\ &+ D_l'' \delta_{22} f_{l,k-1} - \epsilon_{22} m_{l,k-2} \end{aligned} \tag{16}$$

where D_r',D_l',D_r'',D_l'' are the dead zone constants of the right and left inputs of each identification, m_r,m_l are the unmeasurable signals calculated by (11) using respectively D_r,D_l as dead zones limits and f_r, f_l are internal variables defined as:

$$f_{r,k} = \left[-h(D_r - u_{r,k}) + h(u_{r,k} + D_r) \right] \tag{17}$$

$$f_{l,k} = \left[-h(D_l - u_{l,k}) + h(u_{l,k} + D_l) \right] \tag{18}$$

Note that the two parallel identifications will lead to distinct values of the dead zone constants. The estimated value for D_r and D_l can be calculated by:

$$D_r = \frac{D_r' + D_r''}{2} \quad D_l = \frac{D_l' + D_l''}{2}$$

Another remark is that during the identification process, the maximum value of D_r, D_l should be limited by $D_{max} < u_{max}$ (the maximum input signal value).

The model shown in (15) and (16) has a a minimal number of parameters and all of them appear in the expressions linearly, except D_r and D_l, which appear both linearly and non-linearly. The model parameter estimation can be achieved by applying parameter identification methods in each of the equations, but since m_r,m_l and the internal variables are unmeasurable, an iterative algorithm with the estimations of m_r,m_l, f_1, f_2, f_3, f_4 should be used.

VI. Algorithm Initialization

In order to use the scheme described in Section V with a recursive method, it is necessary to have a good initial estimate of the linear and nonlinear model parameters. This can be achieved by collecting data from the robot (input signals and robot poses) through a certain initial time (called setup time) and then using it to calculate the initial estimate of the parameters.

First we apply a linear identification method (like least mean squares) to the collected data as if the model to be identified was linear, using the equations:

$$\begin{aligned}\Delta l_k =&\alpha_1 \Delta l_{k-1} \qquad (19)\\ &+ \delta_{11} u_{r,k-1} - \epsilon_{11} u_{r,k-2}\\ &+ \delta_{12} u_{l,k-1} - \epsilon_{12} u_{l,k-2}\\ \Delta\theta_k =&\alpha_2 \Delta\theta_{k-1} \qquad (20)\\ &+ \delta_{21} u_{r,k-1} - \epsilon_{21} u_{r,k-2}\\ &+ \delta_{22} u_{l,k-1} + \epsilon_{22} u_{l,k-2}\end{aligned}$$

and then we can use a nonlinear optimization algorithm (like flexible polyhedron method [8]) to find the linear and nonlinear parameters that reduce the following function

$$J = \sqrt{(\Delta l_k - \widehat{\Delta l_k})^2 + (\Delta\theta_k - \widehat{\Delta\theta_k})^2} \qquad (21)$$

where $\widehat{\Delta l_k}$ and $\widehat{\Delta\theta_k}$ are respectively the estimates of Δl_k and $\Delta\theta_k$, calculated through the equations (13),(14).

The parameters returned from the optimization process are the initial estimates for the recursive identification algorithm. The same set of initial data can be used to initialize other variables that the recursive method may use. Using a recursive method allow adapting the model to best fit the robot system, since it may vary over the time due several factors, like battery discharge or mechanical wear.

VII. RESULTS

The proposed method was implemented and tested with data provided from both simulated and real robot. The transfer functions of simulated robot are:

$$\begin{aligned}G_{11}(z) =&\frac{7.1181\times10^{-5}(z+0.9932)(z-0.7724)}{(z-1)(z-0.9821)(z-0.7706)}\\ G_{12}(z) =&\frac{7.0344\times10^{-5}(z+0.9948)(z-0.7688)}{(z-1)(z-0.9821)(z-0.7706)}\\ G_{21}(z) =&\frac{0.0341(z-0.9822)(z+0.9167)}{(z-1)(z-0.9821)(z-0.7706)}\\ G_{22}(z) =&\frac{-0.0333(z-0.9819)(z+0.9169)}{(z-1)(z-0.9821)(z-0.7706)}\end{aligned}$$

which can be simplified as:

$$\begin{aligned}G_{11}(z) =&\frac{7.1181\times10^{-5}z+7.0697\times10^{-5}}{(z-1)(z-0.9821)}\\ G_{12}(z) =&\frac{7.0344\times10^{-5}z+6.9978\times10^{-5})}{(z-1)(z-0.9821)}\\ G_{21}(z) =&\frac{0.0341z+0.0313}{(z-1)(z-0.7706)}\\ G_{22}(z) =&\frac{-0.0333z-0.0305}{(z-1)(z-0.7706)}\end{aligned}$$

and with dead zones $D_r = 0.25$,$D_l = 0.20$.

The identified linear parameters obtained through the proposed method are:

$$\begin{aligned}G_{11}(z) =&\frac{8.6728\times10^{-5}z+5.4438\times10^{-5}}{(z-1)(z-0.9818)}\\ G_{12}(z) =&\frac{6.6979\times10^{-5}z+7.5772\times10^{-5})}{(z-1)(z-0.9818)}\\ G_{21}(z) =&\frac{0.0343z+0.0312}{(z-1)(z-0.7699)}\\ G_{22}(z) =&\frac{-0.0335z-0.0308}{(z-1)(z-0.7699)}\end{aligned}$$

and the identified dead zones $D_r = 0.2493$,$D_l = 0.2013$.

The real robot data were obtained from a small mobile robot with two independent DC motors used in robot soccer competitions [9]. The input signals u_r and u_l are the armature voltages of the right and left DC motors, and the output $\mathbf{y} = [\ x \quad y \quad \theta\ ]^T$ is measured with a global vision system[10].

To correct identify the robot dead zones, it's important exciting the robot with some input signal levels where the robot do not move, e.g. input signals inside the dead zone limits. Figure 7 shows a example of a signal used through the identification process.

The outputs Δl and $\Delta\theta$ obtained through the identified model and measured by the vision system are compared in Figures 8 and 9.

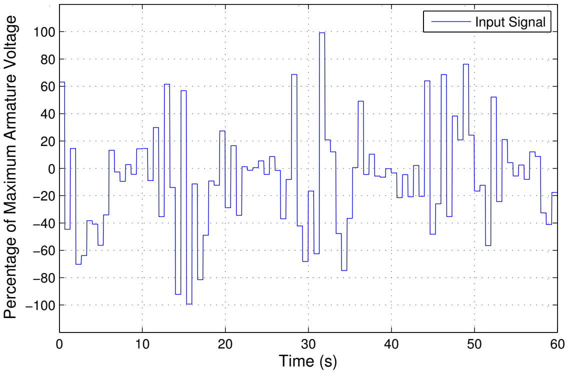

Fig. 7. Example of input signal applied to the robot

VIII. CONCLUSIONS

A new approach to parameter identification of mobile robot with differential drive and input non-linearity has been presented. The method has the advantage of estimating the parameters of both linear and nonlinear blocks of the robot system simultaneously. Furthermore, the recursive algorithm allows the adaptation of the model to changes that may occurs to the robot, like battery discharge and mechanical wear, that usually changes the system behaviour. The result shows the feasibility of the proposed approach and thus it can be used with control algorithms for mobile robots with same characteristics.

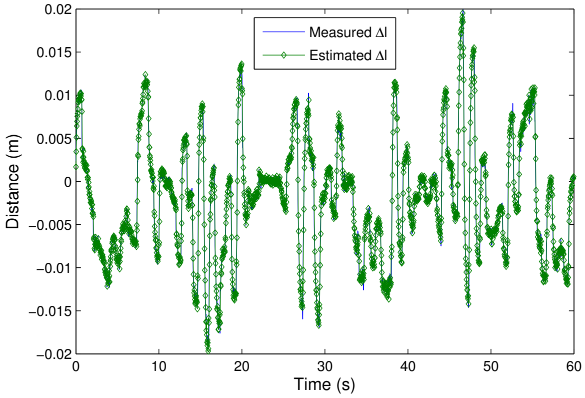

Fig. 8. Comparison between Δl measured and calculated by the proposed model

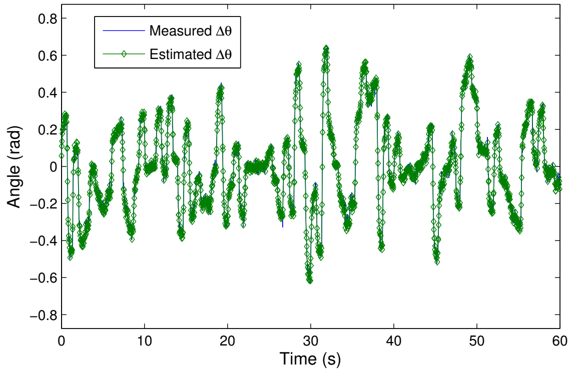

Fig. 9. Comparison between $\Delta\theta$ measured and calculated by the proposed model

In future works, it is expected to apply the described method along with a model based control scheme where the estimated robot parameters will be used to perform adaptive control.

References

[1] P. S. Zygmunt Hasiewicz and G. Mzyk, "Nonlinear system identification under varios prior knowledge," in *Proceedings of the 17th World Congress*. The International Federation of Automatic Control, July 2008, pp. 7849–7858.

[2] G. Tao and P. V. Kokotovic, "Adaptive control of plants with unknow dead-zones," in *IEEE Transactions on Automatic Control*, vol. 39, Jan. 1994, pp. 59–68.

[3] A. S. Sanca, "Controle com compensação de atrito para estruturas de base móveis de robôs manipuladores," Dissertação de mestrado, UFBA, Salvador, BA, 2006.

[4] P. N. Guerra, P. J. Alsina, A. A. D. Medeiros, and A. P. Araújo Jr., "Linear modelling and identification of a mobile robot with differential drive," in *ICINCO – International Conference on Informatics in Control, Automation and Robotics*, Setúbal, Portugal, 2004.

[5] J. Vörös, "Parameter identification of hammerstein systems with asymmetric dead-zones," *Journal of ELECTRICAL ENGINEERING*, vol. 55, no. 1-2, pp. 46–49, 2004.

[6] F. C. Vieira, P. J. Alsina, and A. A. D. Medeiros, "Micro-robot soccer team - mechanical and hardware implementation," in *Congresso Brasileiro de Engenharia Mecânica*, Nov. 2001, pp. 534–540.

[7] *Curves and Surfaces for CAGD: A Practical Guide*, 4th ed. Academic Press, October 1996.

[8] J. A. Nelder and R. Mead, "A simplex method for function minimization," *Computer Journal*, vol. 7, p. 308–313, 1965.

[9] A. C. T. Cerqueira, F. C. A. Lins, J. P. P. Pereira, A. A. D. Medeiros, and P. J. Alsina, "O time POTI de futebol de robôs da UFRN," in *ENRI - Encontro Nacional de Robótica Inteligente*, São Luiz, MA, setembro 2005, http://www.dca.ufrn.br/˜adelardo/.

[10] K. R. T. Aires, P. J. Alsina, and A. A. D. Medeiros, "A global vision system for mobile mini-robots," in *SBAI - Simpósio Brasileiro de Automação Inteligente*, Canela, RS, Brasil, 2001.