The problems of camera measurements in tracking-error fuzzy control of mobile robots

SAŠO BLAŽIČ[1], EL-HADI GUECHI[2], JIMMY LAUBER[2], MICHEL DAMBRINE[2], GREGOR KLANČAR[1]
[1]Faculty of Electrical Engineering, [2]LAMIH laboratory UMR CNRS 8530
[1]University of Ljubljana, [2]University of Valenciennes and Hainaut-Cambrésis
[1]Tržaška 25, Ljubljana, [2]Mont Houy, 59313 Valenciennes cedex
[1]SLOVENIA, [2]FRANCE
[1]{saso.blazic, gregor.klancar}@fe.uni-lj.si, [2]{el-hadi.guechi, jimmy.lauber, michel.dambrine}@univ-valenciennes.fr

Abstract: This paper deals with Takagi-Sugeno modelling and control of nonholonomic mobile robots in the case when the measurements are given by the camera. The measurements in such case are difficult to deal with: they are given only in discrete time samples, usually they are delayed, a high-level noise is present on the measurements. The nonlinear tracking error-model is solved analytically under the premise of ZOH present at the system input. The nonlinear discrete model is developed. Several discretization issues are discussed and the modelling errors are analysed. The sector nonlinearity approach is used for constructing the Takagi-Sugeno model. The control is designed in the LMI framework. Some performance issues are discussed on the simulation cases.

Key-Words: Takagi-Sugeno, PDC control, Mobile robot, Kinematic model, Discretization, Delay, Decay rate

1 Introduction

Mobile, autonomous robots are about to become an important element of the "factory of the future" [19]. Their flexibility and their ability to react in different situations [13] open up totally new applications, leaving no limit to the imagination. To drive the mobile robot from its initial point to the target point, the robot must follow previously planned path. Many restrictions are usually imposed on the path that is being designed. These may arise from physical limitations [11], (moving) obstacles etc.

Several controllers were proposed for mobile robots with nonholonomic constraints, where the two main approaches to controlling mobile robots are posture stabilization and trajectory tracking. The aim of posture stabilization is to stabilize the robot to a reference point, while the aim of trajectory tracking is to have the robot follow a reference trajectory. For mobile robots trajectory tracking is easier to achieve than posture stabilization. This comes from the assumption that the wheel makes perfect contact with the ground, resulting in nonholonomic constraints, which means that not all the velocities are possible at a certain moment. An extensive review of nonholonomic control problems can be found in [10]. According to Brockett's condition [4] nonholonomic systems cannot be asymptotically stabilized around equilibrium using smooth time-invariant feedback. Completely nonholonomic, driftless systems are controllable in a nonlinear sense; therefore, asymptotic stabilization can be obtained using time-varying, discontinuous or hybrid control laws. An exponentially stable, discontinuous feedback controller was proposed by [5] and the point stabilization of mobile robots via state-space exact-feedback linearization using proposed coordinates was studied in [12].

Trajectory tracking is more natural for mobile robots. Usually, the reference trajectory is obtained by using a reference robot; therefore, all the kinematic constraints are implicitly considered by the reference trajectory. The control inputs are mostly obtained by a combination of feedforward inputs, calculated from reference trajectory, and feedback control law, e.g. in [8]. Lyapunov stable time-varying state-tracking control laws were also used [14], where the system's equations are linearized with respect to the reference trajectory, and by defining the desired parameters of the characteristic polynomial the controller parameters are calculated. The stabilization to the reference trajectory requires a nonzero motion condition. Many variations and improvements of this simple and effective state-tracking controller followed in later research.

The approach proposed in our paper is based on discrete Takagi-Sugeno fuzzy model, introduced in [17], obtained from the discrete nonlinear model of the kinematic error. Then, a classical Parallel Distributed Compensation (PDC) law [18] is computed using LMI techniques [3]. The proposed architecture is valid for all the trajectories when their linear and angular velocities are bounded. Moreover, with this approach control law can be implemented easily in real time since it is possible to find stabilizing gains that can operate for several trajectories when their linear an angular velocities are bounded and the stability property is proven for any initial condition in a pre-specified compact set of the state space.

The main point in this paper is to discuss discretization issues of the T-S model and also to discuss the performance of the PDC control.

This paper is organized as follows. In Section 2 the tracking-error model of the mobile robot in continuous case is shown. In Section 3 several discrete versions of the error model are discussed. The T-S model is developed in Section 4. Section 5 deals with the PDC control of the mobile robot. Section 6 discusses the problem of delayed measurements, section 7 gives some simulation results, while the conclusions are stated in Section 8.

2 Continuous kinematic error-model of trajectory tracking

By taking into account the non slipping condition, the kinematic model of the mobile robot in the X-Y plane can be written as follows:

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

where the considered control inputs of the mobile robot v and w are the linear and the angular speed of the robot, respectively. The output variables are x and y (the robot gravity-centre position) and θ (the angle between the speed vector and the X-axis, i.e. the robot orientation).

Fig. 1 illustrates the definition of the posture error $e=\begin{bmatrix} e_x & e_y & e_\theta \end{bmatrix}^T$ expressed in frame of the real robot and determined – using the actual posture $q=\begin{bmatrix} x & y & \theta \end{bmatrix}^T$ of the real robot and the reference posture $q_r=\begin{bmatrix} x_r & y_r & \theta_r \end{bmatrix}^T$ of an virtual reference robot – by the equation:

$$\begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} (q_r - q) \tag{2}$$

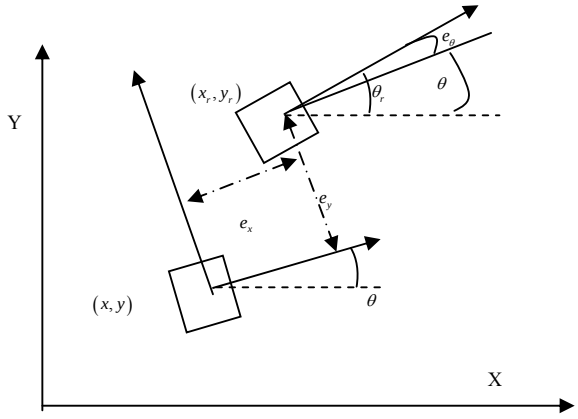

Fig. 1. Posture error

From (1) and (2) and assuming that the virtual robot has a kinematic model similar to (1), the posture error model can be written as follows [7]:

$$\begin{bmatrix} \dot{e}_x \\ \dot{e}_y \\ \dot{e}_\theta \end{bmatrix} = \begin{bmatrix} \cos e_\theta & 0 \\ \sin e_\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_r \\ w_r \end{bmatrix} + \begin{bmatrix} -1 & e_y \\ 0 & -e_x \\ 0 & -1 \end{bmatrix} u \tag{3}$$

where v_r is the linear reference velocity and w_r is the angular reference velocity. The control law is then defined as $u^T=\begin{bmatrix} v & w \end{bmatrix}$. Very often (e.g. [9]) the control is decomposed as:

$$u = \begin{bmatrix} v \\ w \end{bmatrix} = \begin{bmatrix} v_r\cos(e_\theta) + v_b \\ w_r + w_b \end{bmatrix} \tag{4}$$

Inserting the control (4) into (3), the resulting model is given by:

$$\begin{bmatrix} \dot{e}_x \\ \dot{e}_y \\ \dot{e}_\theta \end{bmatrix} = \begin{bmatrix} 0 & w_r & 0 \\ -w_r & 0 & v_r\frac{\sin e_\theta}{e_\theta} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix} + \begin{bmatrix} -1 & e_y \\ 0 & -e_x \\ 0 & -1 \end{bmatrix} u_B \tag{5}$$

where $u_b^T=\begin{bmatrix} v_b & w_b \end{bmatrix}$ is the feedback signal to be defined later.

3 Discretization of the kinematic model

The continuous model is not suitable for the implementation. In our case the information about

the posture of the robot is obtained each 33 ms and also the control is sent to the robots with the same frequency.

3.1 Euler discretization

The simplest method of discretization is the Euler integration formula where the derivative of the function is replaced with the difference quotient, and the following discrete model is obtained from (5):

$$\begin{bmatrix} e_x(k+1) \\ e_y(k+1) \\ e_\theta(k+1) \end{bmatrix} = \begin{bmatrix} 1 & T w_r & 0 \\ -Tw_r & 1 & Tv_r \frac{\sin e_\theta}{e_\theta} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix} + T \begin{bmatrix} -1 & e_y \\ 0 & -e_x \\ 0 & -1 \end{bmatrix} u_B \tag{6}$$

where all the signals on the right hand-side of the equation are functions of k.

3.2 The exact ZOH-discretization

The exact discrete nonlinear kinematic model will be derived in this section. It is obtained on the premise of constant input signals between sampling instants kT ($k \in \mathbb{Z}$) of the output, i.e. ZOH is being used at the system input:

$$\begin{aligned} v(t) &= v(k) = const., \quad kT \le t < (k+1)T \\ w(t) &= w(k) = const., \quad kT \le t < (k+1)T \end{aligned} \tag{7}$$

Let us now analyse the third equation in (1) when $kT \le t < (k+1)T$:

$$\begin{aligned} \int_{kT}^{t} \dot{\theta} dt &= w(k) \int_{kT}^{t} dt = w(k)(t-kT) \\ \theta(t) &= \theta(k) + w(k)(t-kT) \end{aligned} \tag{8}$$

Note that the functional dependence on k or t is not used very formally in the text, i.e. $\theta(\cdot)$ does not have the meaning.

The orientation of the robot changes piece-wise linearly and it is possible to analytically determine the position of the robot in the next sampling instant based on the posture and the control input in the current sampling instant. The equations for the position can be integrated:

$$\begin{aligned} \int_{kT}^{(k+1)T} \dot{x} dt &= v(k) \int_{kT}^{(k+1)T} \cos(\theta(t)) dt = \\ &= v(k) \int_{kT}^{(k+1)T} \cos\big(\theta(k) + w(k)(t-kT)\big) dt \\ \int_{kT}^{(k+1)T} \dot{y} dt &= v(k) \int_{kT}^{(k+1)T} \sin(\theta(t)) dt = \\ &= v(k) \int_{kT}^{(k+1)T} \sin\big(\theta(k) + w(k)(t-kT)\big) dt \end{aligned} \tag{9}$$

Analytical solution for $x(k+1)$ is:

$$\begin{aligned} x(k+1) &= x(k) + \tfrac{v(k)}{w(k)}\big(\sin(\theta(k) + w(k)T) - \sin(\theta(k))\big) \\ &= x(k) - \tfrac{v(k)}{w(k)}(1 - \cos(w(k)T))\sin(\theta(k)) + \\ &\quad + \tfrac{v(k)}{w(k)}\sin(w(k)T)\cos(\theta(k)) \end{aligned} \tag{10}$$

Similarly the model for $y(k+1)$ can be obtained:

$$\begin{aligned} y(k+1) &= y(k) + \tfrac{v(k)}{w(k)}(1 - \cos(w(k)T)) \times \\ &\quad \times \cos(\theta(k)) + \tfrac{v(k)}{w(k)}\sin(w(k)T)\sin(\theta(k)) \end{aligned} \tag{11}$$

The discrete model of the mobile robot is completed by the model for $\theta(k+1)$ obtained from (8):

$$\theta(k+1) = \theta(k) + Tw(k) \tag{12}$$

The reference model for the mobile robot is the same as the model (10), (11), (12), only the input signals are $v_r(k)$ and $w_r(k)$:

$$\begin{aligned} x_r(k+1) &= x_r(k) - \tfrac{v_r(k)}{w_r(k)}(1 - \cos(w_r(k)T))\sin(\theta_r(k)) \\ &\quad + \tfrac{v_r(k)}{w_r(k)}\sin(w_r(k)T)\cos(\theta_r(k)) \\ y_r(k+1) &= y_r(k) + \tfrac{v_r(k)}{w_r(k)}(1 - \cos(w_r(k)T))\cos(\theta_r(k)) \\ &\quad + \tfrac{v_r(k)}{w_r(k)}\sin(w_r(k)T)\sin(\theta_r(k)) \\ \theta_r(k+1) &= \theta_r(k) + Tw_r(k) \end{aligned} \tag{13}$$

Taking into account (2), (10), (11), (12), and (13) the following nonlinear discrete model is obtained:

$$\begin{aligned} e_x(k+1) &= e_x \cos\big(wT\big) + e_y \sin\big(wT\big) \\ &\quad - \tfrac{v_r}{w_r}\sin\big(wT - 2w_rT - e_\theta\big) \\ &\quad + \tfrac{v_r}{w_r}\sin\big(wT - w_rT - e_\theta\big) - \tfrac{v}{w}\sin\big(wT\big) \\ e_y(k+1) &= -e_x \sin\big(wT\big) + e_y \cos\big(wT\big) \\ &\quad - \tfrac{v_r}{w_r}\cos\big(wT - 2w_rT - e_\theta\big) \\ &\quad + wT\cos\big(wT - w_rT - e_\theta\big) \\ &\quad - \tfrac{v}{w}\big(1 - \cos\big(wT\big)\big) \\ e_\theta(k+1) &= e_\theta(k) - Tw(k) + Tw_r(k) \end{aligned} \tag{14}$$

where the explicit dependence on k is omitted in the right hand-side of the equations to make them clearer.

Introducing the control (4) into (14) a very complex model is obtained. The equations of the resulting model with $v_b = 0$ and $w_b = 0$ depend linearly on $e_x(k)$ and $e_y(k)$, so it is very simple to put it to the matrix form:

$$e(k+1) = \begin{bmatrix} \cos(w_r T) & \sin(w_r T) & \frac{v_r \sin(e_\theta)(\cos(w_r T)-1)}{w_r e_\theta} \\ -\sin(w_r T) & \cos(w_r T) & \frac{v_r \sin(e_\theta)\sin(w_r T)}{w_r e_\theta} \\ 0 & 0 & 1 \end{bmatrix} e(k) \quad (15)$$

where $e^T(k) = \begin{bmatrix} e_x(k) & e_y(k) & e_\theta(k) \end{bmatrix}$. It needs to be stressed here that (15) was not obtained by linearization of the nonlinear model. Actually, all the nonlinear dependencies are kept, therefore the model is exact. The solution for the input part is much more nonlinear and only approximate solution for the B matrix can be derived:

$$B_z = T \begin{bmatrix} -\frac{\sin(w_r T)}{w_r T} & b_{z12} \\ \frac{\cos(w_r T)-1}{w_r T} & b_{z22} \\ 0 & -1 \end{bmatrix}$$

$$\begin{aligned} b_{z12} &= -e_x \sin(w_r T) + e_y \cos(w_r T) + \\ &+ \frac{v_r \cos(e_\theta)(1-\cos(w_r T)) - v_r \cos(w_r T + e_\theta)}{w_r} \\ &+ \frac{v_r \cos(e_\theta)\sin(w_r T)}{w_r^2 T} \end{aligned} \quad (16)$$

$$\begin{aligned} b_{z22} &= -e_x \cos(w_r T) - e_y \sin(w_r T) + \\ &+ \frac{-v_r \cos(e_\theta)\sin(w_r T) - v_r \sin(w_r T + e_\theta) + v_r \sin(e_\theta)}{w_r} \\ &+ \frac{v_r \cos(e_\theta)(1-\cos(w_r T))}{w_r^2 T} \end{aligned}$$

The error model obtained from (15) and (16)

$$e(k+1) = A_z e(k) + B_z u_b(k) \quad (17)$$

where $u_b^T(k) = \begin{bmatrix} v_b(k) & w_b(k) \end{bmatrix}$ is very complex for the realization of the Takagi-Sugeno model obtained by sector nonlinearity approach [18]. This is due to the high number of nonlinearities in matrices A_z (4) and B_z (4). Eight nonlinearities means that the T-S model would have 2^8=256 fuzzy rules and the number of the LMIs would be in the range of 2^{16}.

3.3 The simplified ZOH-discretization

In order to simplify the nonlinear model from the previous subsection and reduce the number of nonlinearities the following simplifications were introduced into (15) and (16):

$$\begin{aligned} \cos(w_r T) &\approx 1 \\ \sin(w_r T) &\approx w_r T \end{aligned} \quad (18)$$

The matrix A_z of the resulting model is the same as in the Euler model (6), while the matrix B_z becomes:

$$B_z = T \begin{bmatrix} -1 & e_y - e_x w_r T + v_r T \sin e_\theta \\ 0 & -e_x - e_y w_r T - 2 v_r T \cos e_\theta \\ 0 & -1 \end{bmatrix} \quad (19)$$

The simplified model possesses similar complexity as the Euler one since the number of nonlinearities for the T-S model is 4 (2 in the matrix A_z and 2 in the matrix B_z).

3.4 Comparison of individual models

The models presented in subsections 3.1, 3.2 and 3.3 are compared in this subsection. The comparison means here that the elements of matrices A_z and B_z are compared by calculating their absolute difference.

It is very easy to see that the difference between A_z matrices is the highest when w_r and T are the highest while $e_\theta = 0$. For the matrix B_z the highest difference was obtained by using numerical methods. In this analysis the search is limited to the set: $|e_x| \le 0.1, |e_y| \le 0.1, |e_\theta| \le \frac{\pi}{2}, 0 \le v_r \le 1, |w_r| \le 2$.

The results are shown in Table 1. The error always means maximum absolute difference between the "Euler" model (section 3.1) or the "simplified" model (section 3.3) and the "exact" nonlinear model (section 3.2). Each row shows the values where the error is the biggest. This means that $B_{12\text{Euler}}$ and $B_{12\text{simplified}}$ are not analysed in the same point of the space $(e_x, e_y, e_\theta, v_r, w_r)$. Only the cases where the absolute error is the biggest are shown.

The analysis shows that the only two big (relative) differences occur in the case of the elements B_{12} and B_{22} if the Euler approximation is used (in both cases the approximations are more than the factor 2 too big) while in the case of the "simplified" model the error is low.

Table 1. Worst case elements of the matrices

Matrix Element	Exact value	Approx.	max. error
A_{11}, A_{22}	0.99782	1	0.00218
A_{12}, -A_{21}	0.06595	0.06600	$4.8\cdot10^{-5}$
A_{13}	-0.00109	0	0.00109
A_{23}	0.03298	0.03300	$2.4\cdot10^{-5}$
B_{11}	-0.99927	-1	$7.3\cdot10^{-4}$
B_{21}	-0.03298	0	0.03298
$B_{12\text{Euler}}$	-0.00199	-0.00330	0.00131
$B_{12\text{simplified}}$	0.00330	0.00330	$2.8\cdot10^{-7}$
$B_{22\text{Euler}}$	0.00144	0.00330	0.00186
$B_{22\text{simplified}}$	-0.00494	-0.00549	$5.4\cdot10^{-4}$

3.5 Hybrid ad-hoc model

The Euler model (especially its matrix B) is not very suitable for the control design. It is preferable to used the "simplified" error model. It is also possible to use some hybrid model while still keeping the number of nonlinear functions in the T-S model equal to 4. Obviously, three nonlinear functions (for A_{23}, B_{12}, and B_{22}) can be used directly from the exact model, the fourth one can be $\sin w_rT$ instead of w_rT which is used in the simplified model. Note that the approximation of the $\cos w_rT$ with 1 is worse that the approximation of the $\sin w_rT$ with w_rT, so it would be better to use the approximation for the cosine function, but this would add another nonlinearity into the model. It is also possible to approximate the element B_{21} without increasing the number of nonlinearities:

$$\frac{\cos(w_rT)-1}{w_rT}\approx-\frac{\sin w_rT}{2}-\frac{w_r^3T^3}{24}+\ldots \qquad (20)$$

In the simplified model (in Section 3.3) 0 is used as an approximation for B_{21}, i.e. the error of the approximation is $O(T)$ while in the case of (20) the error is of the size $O(T^3)$).

4 Takagi-Sugeno model of the robot

The discrete TS model is represented through the following polytopic form [17]:

$$e(k+1)=\sum_{i=1}^{r}h_i\left(z(k)\right)\left(A_ie(k)+B_iu(k)\right) \qquad (21)$$

In order to construct the model the approach with sector nonlinearity will be used [18]. This means that the nonlinearities have to be taken from the nonlinear model and used in the premise vector $z(k)$. The vector $z(k)$ in our case is:

$$z(k)=\begin{bmatrix}\sin w_rT\\ \frac{v_r\sin(e_\theta)\sin(w_rT)}{w_re_\theta}\\ b_{z12}\\ b_{z22}\end{bmatrix} \qquad (22)$$

The matrices A_z and B_z are:

$$A_z=\begin{bmatrix}1 & z_1 & 0\\ -z_1 & 1 & z_2\\ 0 & 0 & 1\end{bmatrix}\quad B_z=T\begin{bmatrix}-1 & z_3\\ -\frac{1}{2}z_1 & z_4\\ 0 & -1\end{bmatrix} \qquad (23)$$

Now we have to find minimum and maximum values of the 4 nonlinear functions:

$$z_{l\min}<z_l<z_{l\max}\quad l=1,2,3,4 \qquad (24)$$

where again we limit our search to the following set of the space:

$$|e_x|\le0.1,|e_y|\le0.1,|e_\theta|\le\tfrac{\pi}{2},0\le v_r\le1,|w_r|\le2 \qquad (25)$$

The number of rules is $r=2^4=16$. The matrices of the model are:

$$A_i=\begin{bmatrix}1 & \varepsilon_i^1 & 0\\ -\varepsilon_i^1 & 1 & \varepsilon_i^2\\ 0 & 0 & 1\end{bmatrix}\quad B_i=T\begin{bmatrix}-1 & \varepsilon_i^3\\ -\frac{1}{2}\varepsilon_i^1 & \varepsilon_i^4\\ 0 & -1\end{bmatrix} \qquad (26)$$

where

$$\varepsilon_i^1=\begin{cases}z_{1\min} & \text{for } 1\le i\le8\\ z_{1\max} & \text{else}\end{cases}$$

$$\varepsilon_i^2=\begin{cases}z_{2\min} & \text{for } 1\le i\le4 \text{ and } 9\le i\le12\\ z_{2\max} & \text{else}\end{cases}$$

$$\varepsilon_i^3=\begin{cases}z_{3\min} & \text{for } i\in\{1,\ 2,\ 5,\ 6,\ 9,\ 10,\ 13,\ 14\}\\ z_{3\max} & \text{else}\end{cases} \qquad (27)$$

$$\varepsilon_i^4=\begin{cases}z_{4\min} & \text{for } i=2k-1,k=1,2,\ldots8\\ z_{4\max} & \text{else}\end{cases}$$

5 PDC control of the robot

In order to stabilize the discrete TS fuzzy model (21), a PDC (Parallel Distributed Compensation) control law is used [18]:

$$u_B(k) = -\sum_{i=1}^{r} h_i\left(z(k)\right) F_i e(k) = -F_z e(k) \qquad (28)$$

Several results concerning the stability of the T-S model with the PDC controllers exist. The problem is often solved within the LMI framework.

Here the solution that tries to optimise the decay rate of the system will be used [18]:

$$\underset{\mathbf{X},\mathbf{Y},\mathbf{M}_1 \ldots \mathbf{M}_r}{\text{minimize}}\ \beta$$

subject to

$$\mathbf{X} > \mathbf{0}, \mathbf{Y} \geq \mathbf{0},$$

$$\begin{bmatrix} \beta\mathbf{X} - (r-1)\mathbf{Y} & \mathbf{X}\mathbf{A}_i^T - \mathbf{M}_i^T\mathbf{B}_i^T \\ \mathbf{A}_i\mathbf{X} - \mathbf{B}_i\mathbf{M}_i & \mathbf{X} \end{bmatrix} > 0, \qquad (29)$$

$$\begin{bmatrix} \beta\mathbf{X} + \mathbf{Y}, \\ \frac{1}{2}\left(\mathbf{A}_i\mathbf{X} + \mathbf{A}_j\mathbf{X} - \mathbf{B}_i\mathbf{M}_j - \mathbf{B}_j\mathbf{M}_i\right), \\ \qquad \frac{1}{2}\left(\mathbf{A}_i\mathbf{X} + \mathbf{A}_j\mathbf{X} - \mathbf{B}_i\mathbf{M}_j - \mathbf{B}_j\mathbf{M}_i\right)^T \\ \qquad\qquad \mathbf{X} \end{bmatrix} \geq 0$$

$i < j$ subject to $h_i \cap h_j = \varnothing$

The above generalized eigenvalue problem can be solved by using numerical algorithms. The results are optimal decay rate that has to satisfy $0 \leq \beta < 1$ and

$$\mathbf{F}_i = \mathbf{M}_i\mathbf{X}^{-1} \qquad (30)$$

The problem is that the LMI Toolbox in Matlab cannot solve the LMIs with the '$\geq$'. In this case the '$>$' relation can be used instead in the definition of the inequalities.

The other problem that is more serious is that the solution for β does not carry much information about the actual decay rate. Very often the "decay rate" β predicts instability although the system is stable. This problem can be partly overcome if a new variable is introduced:

$$\gamma = \beta - 1 \qquad (31)$$

A slightly modified generalized eigenvalue problem is then obtained:

$$\underset{\mathbf{X},\mathbf{Y},\mathbf{M}_1 \ldots \mathbf{M}_r}{\text{minimize}}\ \gamma$$

subject to

$$\mathbf{X} > \mathbf{0}, \mathbf{Y} \geq \mathbf{0},$$

$$\begin{bmatrix} \gamma\mathbf{X} + \mathbf{X} - (r-1)\mathbf{Y} & \mathbf{X}\mathbf{A}_i^T - \mathbf{M}_i^T\mathbf{B}_i^T \\ \mathbf{A}_i\mathbf{X} - \mathbf{B}_i\mathbf{M}_i & \mathbf{X} \end{bmatrix} > 0, \qquad (32)$$

$$\begin{bmatrix} \gamma\mathbf{X} + \mathbf{X} + \mathbf{Y}, \\ \frac{1}{2}\left(\mathbf{A}_i\mathbf{X} + \mathbf{A}_j\mathbf{X} - \mathbf{B}_i\mathbf{M}_j - \mathbf{B}_j\mathbf{M}_i\right), \\ \qquad \frac{1}{2}\left(\mathbf{A}_i\mathbf{X} + \mathbf{A}_j\mathbf{X} - \mathbf{B}_i\mathbf{M}_j - \mathbf{B}_j\mathbf{M}_i\right)^T \\ \qquad\qquad \mathbf{X} \end{bmatrix} \geq 0$$

$i < j$ subject to $h_i \cap h_j = \varnothing$

The new algorithm finds the solution for $\gamma < 0$ easier than the original algorithm for $\beta < 1$. The obtained decay rate still does not show the actual decay rate. This is of course due to a very conservative approach which is a general characteristic of this approach.

The "optimal" results are unusable because the control system is too quick. This problem can be solved by using the constraint on the input [18]:

The constraint $v^2(t) + w^2(t) < \mu^2$ is enforced at all times if the following LMIs are added to the set of LMIs (32):

$$\begin{bmatrix} \mathbf{X} & \mathbf{M}_i^T \\ \mathbf{M}_i & \mu^2\mathbf{I} \end{bmatrix} \geq 0, \qquad \mathbf{X} - \phi^2\mathbf{I} \geq 0 \qquad (33)$$

where $|e(0)| \leq \phi$.

The above addition does limit the control signal successfully, but is also very conservative since it works for any initial condition. If (33) is to be used, the parameters ϕ and μ need some tuning.

A lot of relaxations of certain conditions exist in the literature. The efficient solution of LMIs was not the goal of this paper.

6 The problem of delay

Besides having the output defined only in discrete moments, the measurements are also delayed. In order to have good performance control, this delay has to be compensated. This is usually achieved by running the robot model in parallel to the actual robot. The control scheme is shown in Fig. 2.

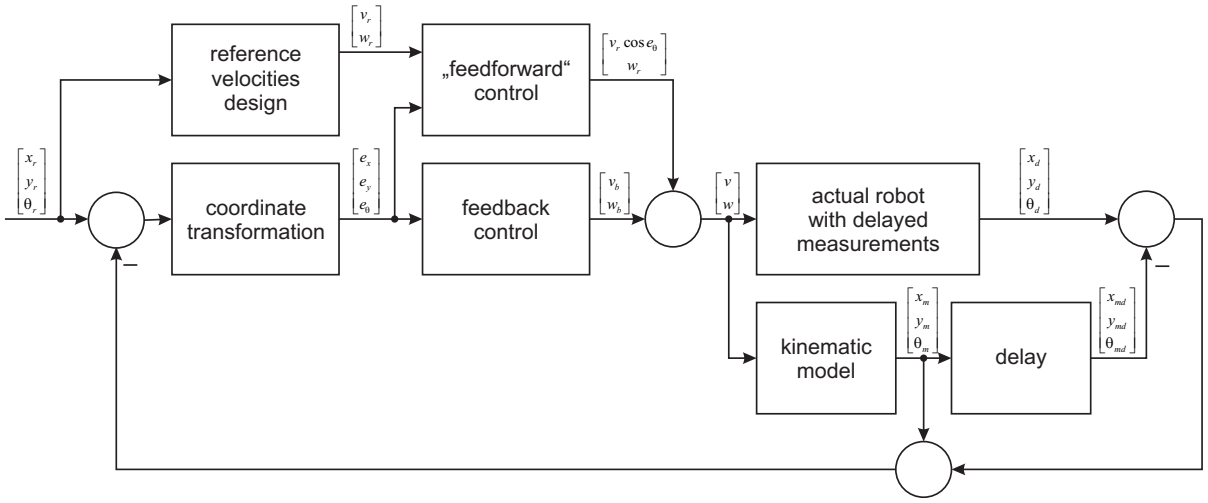

Fig. 2. The control scheme in the case of delayed measurements

Although it is possible to solve the problem by different observers, in our case a simple Smith predictor is used. We have to obtain the current posture of the robot (x, y, θ) without having it measured directly. Instead we can calculate it from the actual measurements (x_d, y_d, θ_d), the output of the model (x_m, y_m, θ_m), and the output of the model that is delayed for the same time as the measured signals are ($x_{md}, y_{md}, \theta_{md}$). The idea is to estimate the undelayed output with the premise that the difference between the undelayed and the delayed outputs of the model is the same as the difference between the undelayed and the delayed outputs of the actual robot. We therefore feed back the "undelayed" signals:

$$\begin{aligned} \hat{x} &= x_d + (x_m - x_{md}) \\ \hat{y} &= y_d + (y_m - y_{md}) \\ \hat{\theta} &= \theta_d + (\theta_m - \theta_{md}) \end{aligned} \tag{34}$$

7 Simulation experiments

There were two experiments conducted in this paper. The first shows the trajectory tracking with the discrete PDC controller proposed in this paper. The results are shown in Figs. 3, 4, and 5. Very good results are obtained in this case. It can be observed that the tracking errors converge to 0.

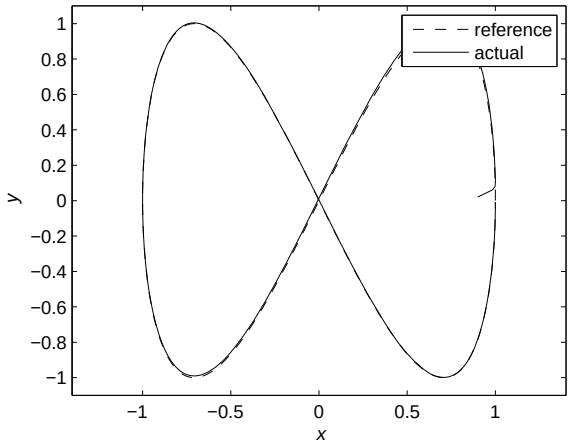

Fig. 3. The reference trajectory (dashed) and the actual one (solid)

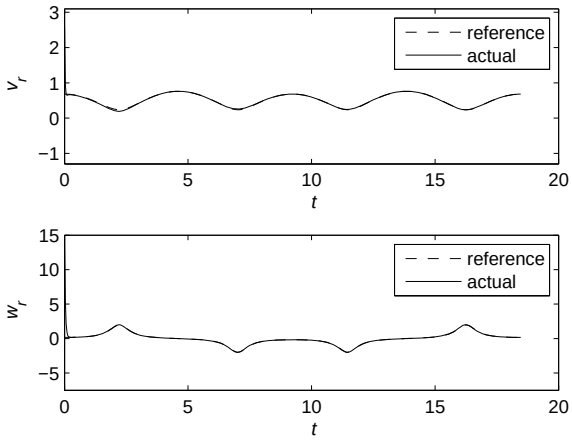

Fig. 4. The linear (upper figure) and the angular (lower figure) velocities

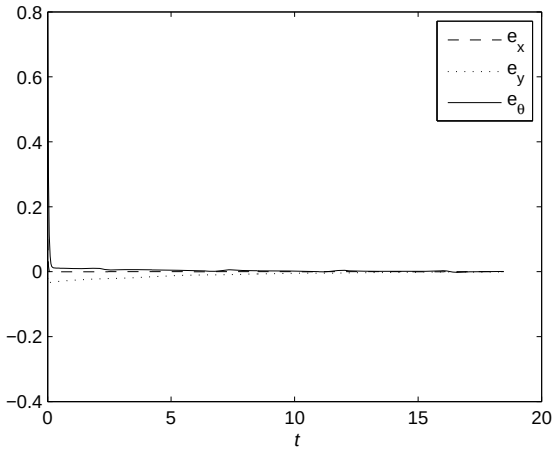

Fig. 5. Convergence of the errors

In the second experiment the delay was also present on the measured outputs. In the first case (Figs. 6, 7, 8) the delay is one sample time (33,3 ms). The measured signal is used directly as the output of the system. The delay causes that the system starts oscillating while no oscillations are found in the delay-free case (Figs. 3, 4, 5). In the second case (Figs. 9, 10, 11) the effect of delay is reduced by feeding back the signals according to (34). When testing the delay of 2 sample times (66,7 ms) instability occurred in the case when measured signals were treated as the outputs of the system. Very good results are still obtained in this case if using Smith predictor. It needs to be stressed that the model of the system was not initialised with the same values as the actual system. The initial error of the model was 0.1 m in the x-direction, 0.1 m in the y-direction, and 45^o in the rotation.

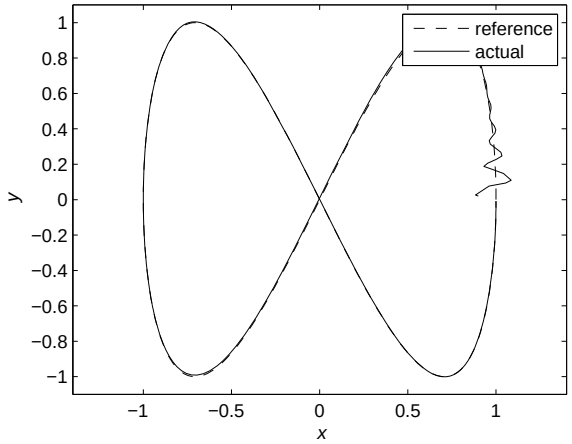

Fig 6. The experiment with the delay of 1 sample time – the delay not taken into account: the reference trajectory (dashed) and the robot path (solid)

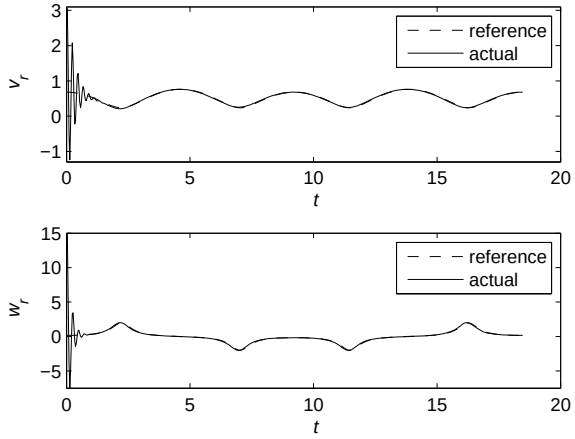

Fig 7. The experiment with the delay of 1 sample time – the delay not taken into account: the linear (upper) and the angular (lower figure) velocities

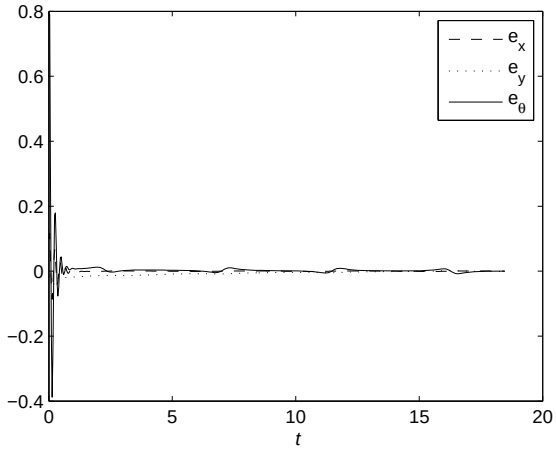

Fig 8. The experiment with the delay of 1 sample time – the delay not taken into account: the errors

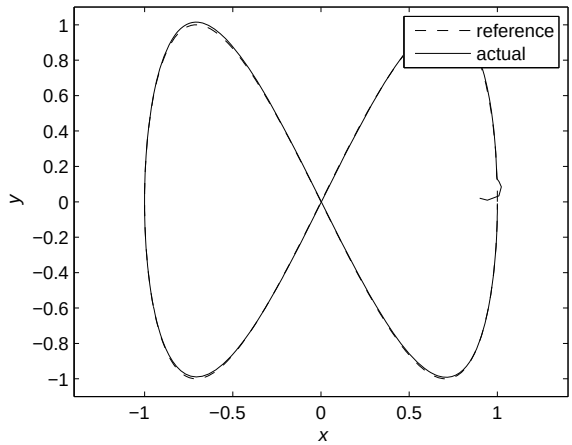

Fig 9. The experiment with the delay of 1 sample time – the delay compensated: the reference trajectory (dashed) and the robot path (solid)

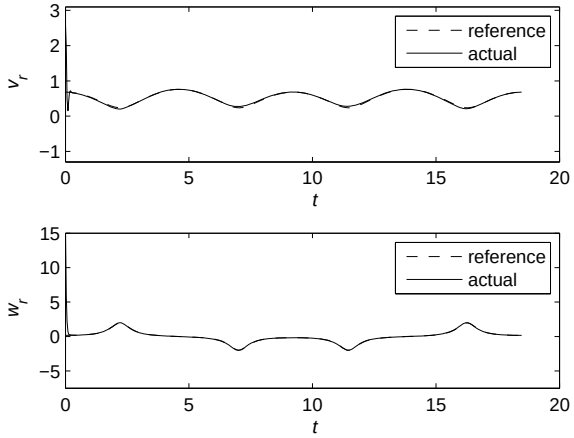

Fig 10. The experiment with the delay of 1 sample time – the delay compensated: the linear (upper) and the angular (lower) velocities

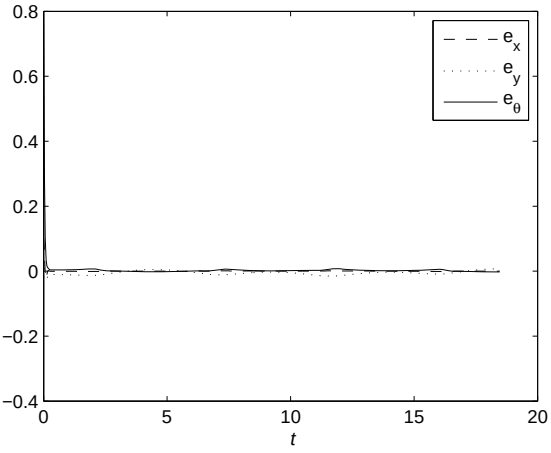

Fig 11. The experiment with the delay of 1 sample time – the delay compensated: the errors

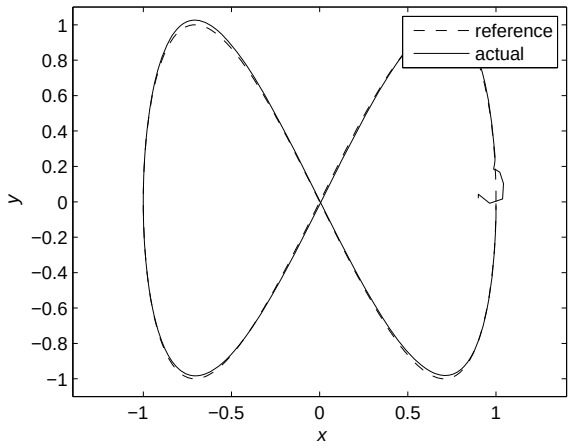

Fig 12. The experiment with the delay of 2 sample times – the delay compensated: the reference trajectory (dashed) and the robot path (solid)

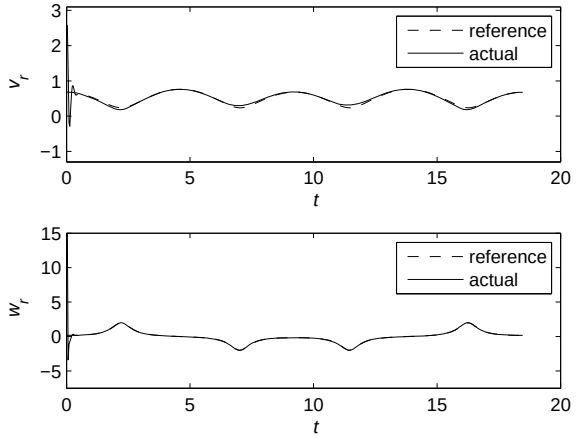

Fig 13. The experiment with the delay of 2 sample times – the delay compensated: the linear (upper figure) and the angular (lower figure) velocities

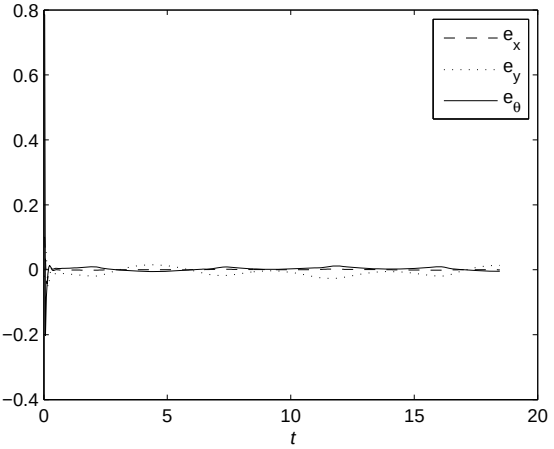

Fig 14. The experiment with the delay of 2 sample times – the delay compensated: convergence of the errors

8 Conclusion

This paper deals with Takagi-Sugeno modelling and control of nonholonomic mobile robots. The nonlinear tracking error-model is solved analytically under the premise of ZOH present at the system input. The nonlinear discrete model is developed. Several discretization issues are discussed and the modelling errors are analysed. It is shown that the model obtained by the Euler integration method possesses big errors in some cases and it should be not used. Models of similar complexity result in much better precision. The sector nonlinearity approach is used for constructing the Takagi-Sugeno model. The effect of delay is also studied. The control is designed in the LMI framework. Some

performance issues are discussed on the simulation cases.

References:

[1] M. Aicardi, Coordination and control of a team of mobile robots, WSEAS Transactions on Systems, Vol. 6, No. 6, 2007, pp. 1116-23.

[2] S. Blažič, I. Škrjanc, D. Matko, Simple fuzzy adaptive control for a class of nonlinear plants, WSEAS Transactions on Systems, Vol. 3, No. 2, 2004, pp. 784-788.

[3] S. Boyd et al., Linear matrix inequality in system and control theory, Studies in Applied Mathematics, Philadelphia, 1994.

[4] R. W. Brockett, Asymptotic stability and feedback stabilization, in: R.W. Brockett, R.S. Millman, H.J. Sussmann (Eds.), Differential Geometric Control Theory, Birkhauser, Boston, MA, 1983, pp. 181–191.

[5] C. Canudas de Wit, O.J. Sordalen, Exponential Stabilization of Mobile Robots with Nonholonomic Constraints, IEEE Transactions on Automatic Control, Vol. 37, No. 11, 1992, pp. 1791–1797.

[6] K. N. Faress, M. T. El Hagry, A. A. El Kosy, Trajectory tracking control for a wheeled mobile robot using fuzzy logic controller, WSEAS Transactions on Systems, Vol. 4, No. 7, 2005, pp. 1017-21.

[7] El H. Guechi, J. Lauber, M. Dambrine, S. Blažič, G. Klančar, Tracking-error model-based PDC control for mobile robots with acceleration limits, submitted to IEEE International Conference on Fuzzy Systems FUZZ-IEEE, 2009.

[8] D.-H. Kim, J.-H. Oh, Tracking control of a two-wheeled mobile robot using input–output linearization, Control Engineering Practice, Vol. 7, No. 3, 1999, pp. 369–373.

[9] G. Klančar, I. Škrjanc, Tracking-error model-based predictive control for mobile robots in real time, Robotics and Autonomous Systems, Vol. 55, No. 6, 2007, pp. 460-469.

[10] I. Kolmanovsky, N. H. McClamroch, Developments in Nonholonomic Control Problems, IEEE Control Systems, Vol. 15, No. 6, 1995, pp. 20–36.

[11] M. Lepetič, G. Klančar, I. Škrjanc, D. Matko, B. Potočnik, Time optimal planning considering acceleration limits, Robotics and Autonomous Systems, Vol. 45, 2003, pp. 199-210.

[12] K. Park, H. Chung, J.G. Lee, Point stabilization of mobile robots via statespace exact feedback linearization, Robotics and Computer Integrated Manufacturing, Vol. 16, 2000, pp. 353–363.

[13] L. Podsedkowski, J. Nowakowski, M. Idzikowski, I. Vizvary, A new solution for path planning in partially known or unknown environment for nonholonomic mobile robots, Robotics and Auto-nomous Systems, Vol. 34, 2001, pp. 145–152.

[14] C. Samson, Time-varying feedback stabilization of car like wheeled mobile robot, International Journal of Robotics Research, Vol. 12, No. 1, Vol. 1993, 55–64.

[15] I. Škrjanc, M. Lepetić, J. L. Figueroa, S. Blažič, Fuzzy model-based predictive control for a CSTR with multiple steady state: A simulation study and a comparison with other nonlinear MBPC control algorithms, WSEAS Transactions on Systems, Vol. 3, No. 2, 2004, pp. 789-94.

[16] I. Škrjanc, Monitoring of waste-water treatment plant based on fuzzy model, WSEAS Transactions on Systems, Vol. 6, No. 3, 2007, pp. 637-642.

[17] T. Takagi, M. Sugeno, Fuzzy identification of systems and its application to modeling and control, IEEE Trans. On System, Man and Cybernetics, Vol. 15, No. 1, 1985, pp. 116-132.

[18] K. Tanaka, H. O. Wang, Fuzzy Control Systens Design and Analysis: A Linear Matrix Inequality Approach, John Wiley & Sons, Inc., 2001.

[19] Y. Ting, W. I. Lei, H. C. Jar, A Path Planning Algorithm for Industrial Robots, Computers & Industrial Engineering, Vol. 42, 2002, pp. 299-308.