NEURAL TRACKING TRAJECTORY OF THE MOBILE ROBOT KHEPERA II IN INTERNAL MODEL CONTROL STRUCTURE

ŠUSTER PETER, JADLOVSKÁ ANNA

Dept. of Cybernetics and Artificial Intelligence, Faculty of Electrical Engineering and Informatics, TU Košice, Letná 9, 042 00 Košice, Slovakia, Tel.: +421 55 6252575, Fax: +421 55 6253574, URL: web.tuke.sk/kkui
e-mail:peter.suster@tuke.sk, anna.jadlovska@tuke.sk

Abstract: This paper introduces a solution to the reference trajectory tracking problem done by a differential wheeled mobile robot Khepera II. The paper includes a mathematical model of the mobile robot, which we use for the acquisition of a set training data for creating forward and inverse neural model. The forward neural model of the mobile robot we implemented into the control structure of IMC together with the inverse neural model. The purpose of the control structure was the reference trajectory tracking, which we verified using the Neural Network Toolbox of Matlab/Simulink.

Keywords: mobile robot, MLP neural network, forward neural model, inverse neural model, tracking trajectory, Internal Model Control structure

1 INTRODUCTION

Mobile robots are suitable for different applications in an environment where the high degree of autonomy is required. The requirement of autonomy of the mobile robot has become an important part the scientific research in the last decades. Since this is a mobile robot, the requirement of autonomy is applied in particular way to move the robot among the obstacles in the area i.e. tracking defined reference trajectory. One way how to ensure a high degree of autonomy of reference trajectory tracking of the mobile robot is to use model predictive control approach [Kühne et all., 2005]. Another way to track trajectory of the mobile robot is to use a neuro-fuzzy controller [Masár, 2007]. In that paper, neuro-fuzzy controller is designed, which allows the mobile robot to track the given trajectory.

Modification approach [Masár, 2007] is mentioned in our paper with difference that instead of the neuro-fuzzy controller we proposed nonparametric neural controller as inverse neural model. Simulation model of the mobile robot is based on a real mobile robot Khepera II of K-Team Corp. [K-Team], which was used to verify the proposed algorithms for tracking defined reference trajectory. Mobile robot Khepera II has been chosen because we have mobile robots Khepera III at disposal in our department, on which we would like to apply our obtained results.

The paper is organized as follows. The part two includes a mathematical model of the mobile robot, which consists of kinematic and dynamic parts. We proposed a control structure to ensure that the mobile robot tracks one trajectory from the set of reference trajectories. The mathematical model together with control structure, we used to obtain training data for modeling forward and inverse neural model in the Matlab/Simulink. Part three contains a design of forward neural model starting with an acquisition of the training data going through training of model to validating of forward neural model of a mobile robot. The focus of this paper is in the part four and in the part five. In the part four there is a design of inverse neural model of the mobile robot, which will be used as a nonparametric neural controller for tracking defined reference trajectory in the control structure IMC. Part five deals with simulation verification control structure IMC for tracking reference trajectory of the mobile robot. Verification of IMC control structure for tracking of the reference trajectory of the mobile robot is situated in the part five. Output of IMC control structure, into which we have implemented the forward and inverse neural model, was made a trajectory of simulation model of the mobile robot.

2 MATHEMATICAL MODEL OF MOBILE ROBOT KHEPERA II

Mobile robot Khepera II has a differential chassis, based on two independently driven wheels with common axis of rotation. The chassis is completed by the third wheel, which rotates in all the directions and ensures the stability of the mobile robot.

2.1 KINEMATIC MODEL OF THE MOBILE ROBOT

The created model is based on the several assumptions, namely that the robot moves on a perfect flat surface without sliding, and also neglects the rolling resistance of the wheels. Position of the mobile robot is given by the coordinates x, y and angle θ, which represents the rotation mobile robot in relation to the chosen coordinate system. Mobile robot is controlled by an angular velocity of the wheels ω_L, ω_R. Between the angular velocities ω_L, ω_R and peripheral speeds v_L, v_R there are the following relations

$$v_L = r\omega_L \quad , \quad v_R = r\omega_R \tag{1}$$

where r is radius of the wheel.
The movement in the plane x, y applicable relations

$$v_L = \left(R + \frac{b}{2}\right)\omega \quad , \quad v_R = \left(R - \frac{b}{2}\right)\omega \tag{2}$$

where R is radius of the turning robot and b is the distance between the wheels. After modifying the equations (2) we obtain the equation for a peripheral and an angular velocity of the mobile robot

$$v = \frac{v_L + v_R}{2} \quad , \quad \omega = \frac{v_L - v_R}{b} \tag{3}$$

The position and the rotation of the robot in the space can be based on the above to express the following equations, which form a kinematic model of the mobile robot (Fig.1)

$$\begin{aligned} \dot{x}(t) &= \frac{v_R + v_L}{2}\cos\theta \\ \dot{y}(t) &= \frac{v_R + v_L}{2}\sin\theta \\ \dot{\theta}(t) &= \omega = \frac{v_R - v_L}{b} \end{aligned} \tag{4}$$

where the inputs into the system are speed wheels v_L and v_R, and the outputs are x, y, θ. The kinematic model (Fig.1) allows us to determine the position and the rotation of the robot under the condition that we know the initial state of the robot and we have updated information about the speed of the individual wheel [Šembera et all., 2007].

2.2 DYNAMIC MODEL OF THE MOBILE ROBOT

The kinematic model does not include friction forces acting on the wheel and the total mass of the mobile robot, so we have extended the mathematical model of the dynamic part (Fig.2), which has the following shape

$$ma_t = F_L + F_R \tag{5}$$

$$J\varepsilon = \frac{(F_L - F_R)b}{2} \tag{6}$$

where tangent acceleration a_t is given by mass of the robot m and tangent forces F_L and F_R, which acting on the wheels due to change in the rotation speed. The angular acceleration ε is determined by the same forces, moment of inertia of the robot J and distance between the wheels b [Gajdušek et all., 2006].

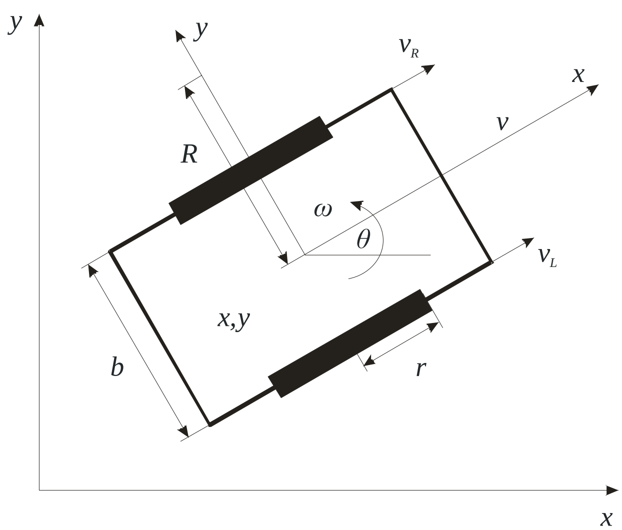

Fig. 1 - Kinematic model of the mobile robot

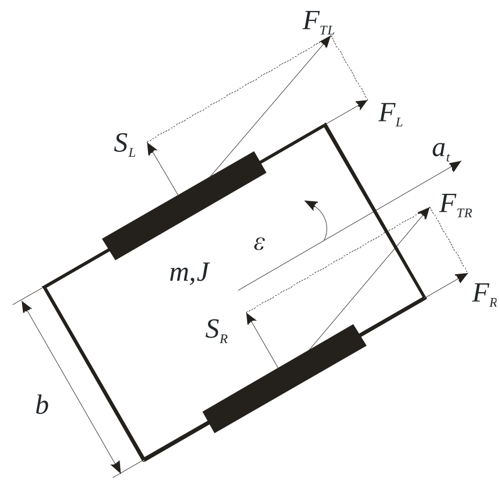

Fig. 2 - Dynamic model of the mobile robot

If we choose the state variable peripheral speed of the mobile robot $x_1(t) = v(t)$, so after modifying the equation (5) we get the following equation

$$\dot{x}_1(t) = \frac{F_L}{m} + \frac{F_R}{m} \tag{7}$$

If in the equation (7) we consider that tangent forces F_L and F_R are input variables $u_1(t)$ and $u_2(t)$ i.e. $u_1(t) = F_L$ and $u_2(t) = F_R$ then the equation (7) will have the shape

$$\dot{x}_1(t) = \frac{1}{m}u_1 + \frac{1}{m}u_2 \tag{8}$$

When we apply the same procedure to the equation (6) with difference that we have selected for the state variable $x_2(t)$ as an angular velocity of the mobile robot $x_2(t) = \omega(t)$ with unchanged inputs u_1 and u_2 after modifying we get second state equation of the dynamic model of the mobile robot

$$\dot{x}_2(t) = \frac{b}{2J}u_1 - \frac{b}{2J}u_2 \tag{9}$$

An angular speeds ω_L and ω_R $(\omega_L = \dot{\theta}_L, \omega_R = \dot{\theta}_R)$ of the mobile robot are driven by the voltages U_L and U_R. Differential equations expressing this fact have the shape [Dominguez, 2007]:

$$J\ddot{\theta}_L(t) + F_T\dot{\theta}_L(t) + F_L r = U_L \tag{10}$$

$$J\ddot{\theta}_R(t) + F_T\dot{\theta}_R(t) + F_R r = U_R \tag{11}$$

where F_T is the friction force acting on the wheel.
If we consider that next state variable $x_3(t)$ is an angular speed of the left wheel i.e. $x_3(t) = \dot{\theta}_L(t)$ then the equation (10) will have the shape

$$\dot{x}_3(t) = -\frac{F_T}{J}x_3(t) - \frac{r}{J}F_L + \frac{1}{J}U_L \tag{12}$$

For the input $u_3(t)$ we have chosen voltage, which we drove the left wheel i.e. $u_3(t) = U_L$ then after modifying the equation (12) we get the following state equation

$$\dot{x}_3(t) = -\frac{F_T}{J}x_3(t) - \frac{r}{J}u_1 + \frac{1}{J}u_3 \tag{13}$$

The same procedure as we applied for the equation (10) with exception that we have selected for the state variable $x_4(t)$ an angular speed of the right wheel i.e. $x_4(t) = \dot{\theta}_R(t)$. For input $u_4(t)$ we have chosen by voltage, which we used to drive of the right wheel i.e. $u_4(t) = U_R$ then after modifying we will get the final state equation

$$\dot{x}_4(t) = -\frac{F_T}{J}x_4(t) - \frac{r}{J}u_2 + \frac{1}{J}u_4 \tag{14}$$

From the equations (8), (9), (13), (14) we have obtained a dynamic model of the mobile robot in the state space

$$\begin{aligned}
\dot{x}_1(t) &= \frac{1}{m}u_1 + \frac{1}{m}u_2 \\
\dot{x}_2(t) &= \frac{b}{2J}u_1 - \frac{1}{2J}u_2 \\
\dot{x}_3(t) &= -\frac{F_T}{J}x_3(t) - \frac{r}{J}u_1 + \frac{1}{J}u_3 \\
\dot{x}_4(t) &= -\frac{F_T}{J}x_4(t) - \frac{r}{J}u_2 + \frac{1}{J}u_4
\end{aligned} \tag{15}$$

where the state variables system and their derivates have the following physical meaning:
$x(t) = [x_1(t), x_2(t), x_3(t), x_4(t)] = [v(t), \omega(t), \omega_L(t), \omega_R(t)]$
$\dot{x}(t) = [\dot{x}_1(t), \dot{x}_2(t), \dot{x}_3(t), \dot{x}_4(t)] = [a_t(t), \varepsilon(t), \varepsilon_L(t), \varepsilon_R(t)]$
the inputs into the system are: $u = [u_1(t), u_2(t), u_3(t), u_4(t)] = [F_L, F_R, U_L, U_R]$
and the outputs from the system are: $y(t) = [y_1(t), y_2(t)] = [x_3(t), x_4(t)]$.

2.3 IMPLEMENTATION OF THE MODEL ROBOT INTO THE SIMULATION LANGUAGE MATLAB/SIMULINK

As was mentioned in section 2.1 and 2.2 the mathematical model of the mobile robot consists of a kinematic (4) and dynamic parts (15). Kinematic model of the mobile robot serves for determining the coordinates the current position x, y and the angle θ, which represents the rotation of the mobile robot in the relation to the chosen coordinate system. The inputs into the kinematic model are an angular velocities ω_L, ω_R, which are generated by the dynamic model (Fig.2). From the state equations of the dynamic model of the mobile robot (15) we can obtain state description in the shape

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) \end{aligned} \tag{16}$$

where matrixes A, B, C have following structure

$$A = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{F_T}{J} & 0 \\ 0 & 0 & 0 & -\frac{F_T}{J} \end{pmatrix}, \; B = \begin{pmatrix} \frac{1}{m} & \frac{1}{m} & 0 & 0 \\ \frac{b}{2J} & -\frac{b}{2J} & 0 & 0 \\ -\frac{r}{J} & 0 & \frac{1}{J} & 0 \\ 0 & -\frac{r}{J} & 0 & \frac{1}{J} \end{pmatrix}, \; C = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{16a}$$

To ensure satisfactory properties of the dynamic model of the mobile robot, we have proposed PI controller parameters using the standard shape method (Fig. 3 dynamic model + PI controller). For the design parameters of PI controller we obtained the transfer function from the state description for the inputs u_3 and u_4 which has the following shape

$$F(s) = \frac{b_1 s + b_0}{a_2 s^2 + a_1 s + a_0} \tag{17}$$

We have proposed a simulation scheme of the model of the mobile robot (Fig.3) in the Matlab/Simulink, based on the equations of the kinematic (4) and dynamic models (15) together with the proposed PI controllers, for the defined parameters of the mobile robot:
$J = 0.1\,\mathrm{kgm}^2$ – moment of inertia, $r = 0.005\,\mathrm{m}$ – radius of the wheel, $F_T = 0.0003\,\mathrm{N}$ – friction force, $m = 0.08\,\mathrm{kg}$ – mass of robot, $b = 0.05\,\mathrm{m}$ – distance between wheels.

The input into the model robot are an angular velocities for the left and the right wheels (fider_L, fider_R) and outputs are coordinates the current position of the robot (x, y) and current rotation of robot (angle)

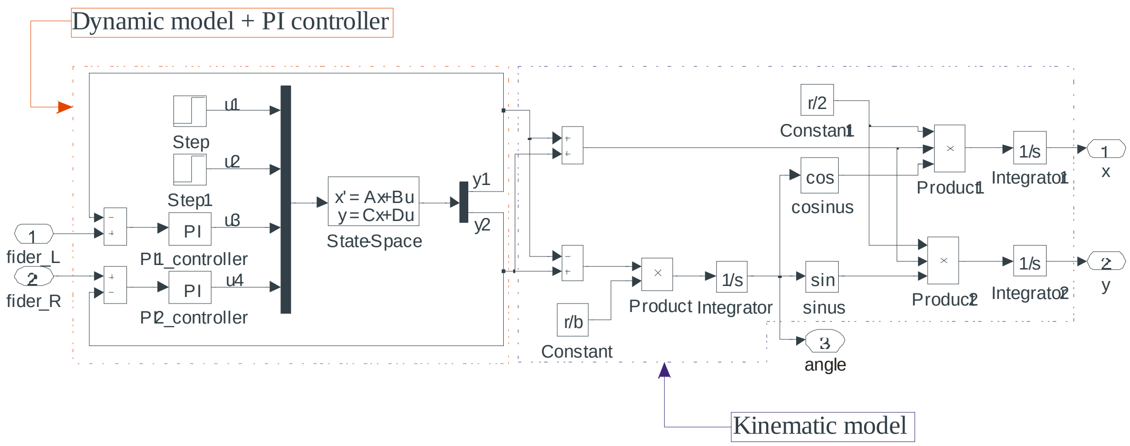

Fig. 3 – Simulation scheme of the model robota

We proposed a control structure to ensure that the mobile robot tracks one trajectory of the set of reference trajectories (Fig.4) [Fic, 2009]. The inputs into control structure of the model robot are coordinates of the current position of the model robot *x*, *y* (real x, real y) and the coordinates of the reference trajectory x_{ref}, y_{ref} (reference x, reference y). We have calculated Euclidean distance between current and desired position of the model robot by means of these coordinates. The input into control structure is also actual rotation of robot, which compares with calculated rotation of the robot. The outputs from control structure are angular velocities for the left and the right wheel (fider_L, fider_R).

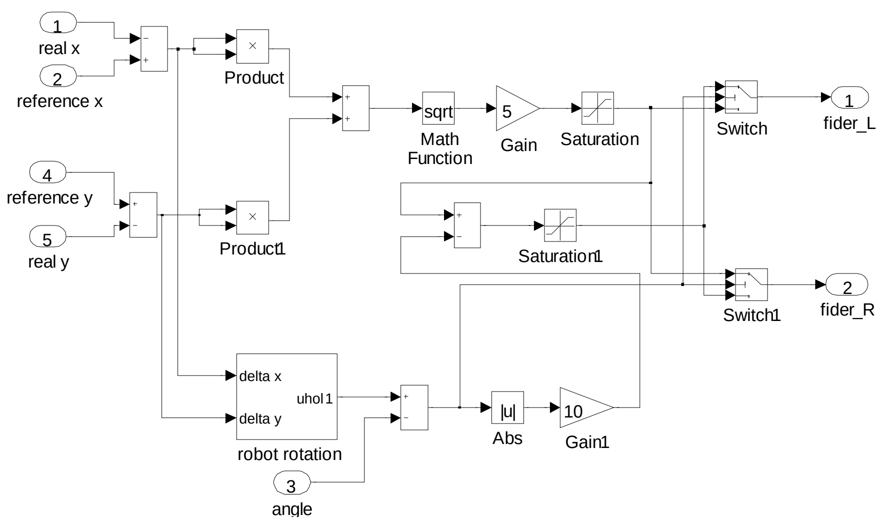

Fig. 4 – Simulation scheme of the control structure

We used subsystems control structure and model of the robot in the simulation scheme for acquisition of training data needed for design of forward (Fig.6) and inverse neural models (Fig.12), which we have implemented into control structure IMC with the aim of tracking of the defined reference trajectory. Simulations were carried out by the sample period $T_{vz} = 0{,}01s$.

3 FORWARD NEURAL MODEL OF THE MODEL ROBOT

Neural model, which approximates dynamic of the system is called forward model. Neural network is placed in parallel with the identification system and the error between output of the neural network $\hat{y}(k+1)$ and output of the dynamic system $y(k+1)$, the so-called prediction error, is used as training signal for neural network (Fig.5) of MLP type.

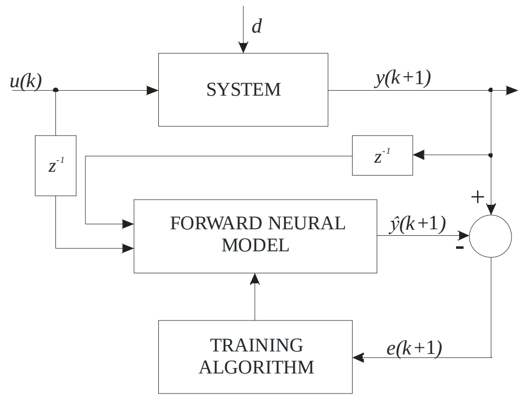

Fig. 5 – Identification structure based on output prediction error

If the output of the neural model is $\hat{y}(k+1)$ then we can express the equation by approximation

$$\hat{y}(k+1) = \hat{f}\left[y(k),\ldots,(k-n+1),u(k),\ldots,u(k-m+1)\right] \tag{18}$$

where $\hat{f}$ is non-linear input-output representation dynamics of the mobile robot by the neural model and $y(k)$ resp. $u(k)$ is n – output resp. m – input of the previous values [Jadlovská et all., 2002].

3.1 OBTAINING OF THE TRAINING DATA

We proposed the following simulation scheme for to obtain a set of training data in the Simulink:

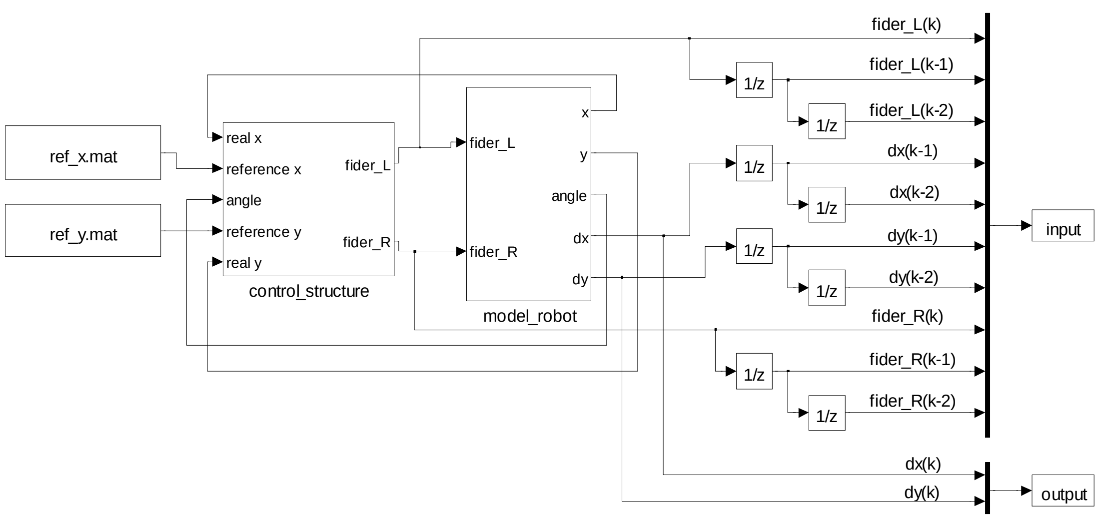

Fig. 6 – Simulation scheme for acquisition of training data for forward neural model

During the simulation, for the input of the system was defined reference trajectory, which was represented by the x and y coordinates. Then control structure generated inputs for the model of robot that are shown in Fig. 7

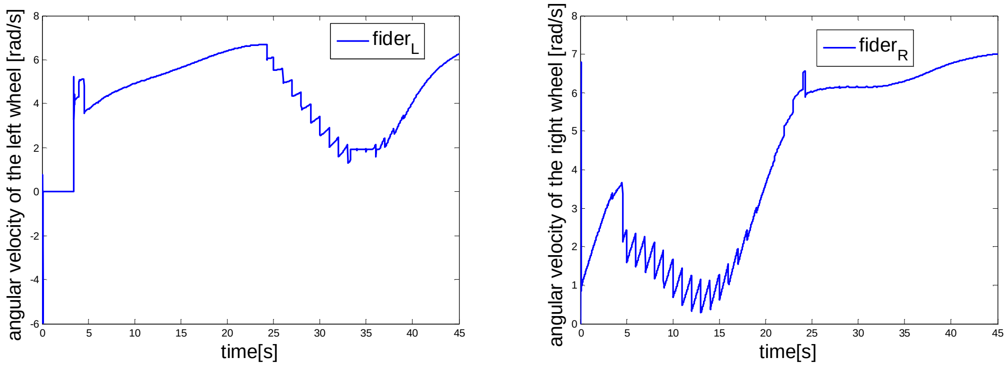

Fig. 7 – The inputs to the system to obtain training data for the forward neural model

Output signals from the system are shown in the following Fig. 8

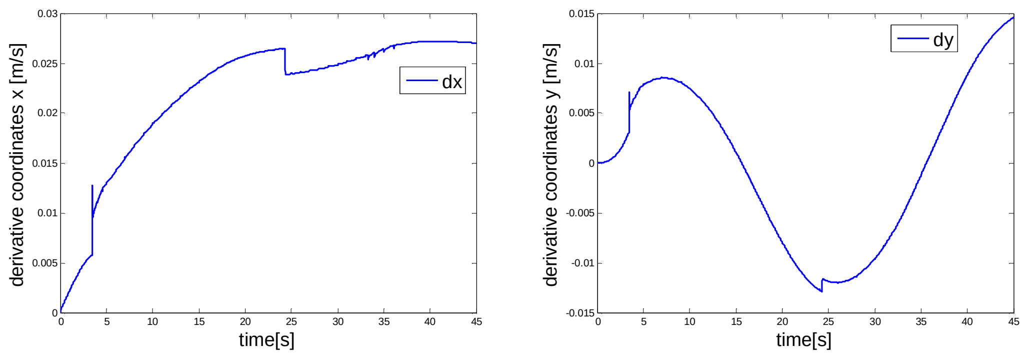

Fig. 8 – Outputs from the system to obtain training data for the forward neural model

3.2 TRAINING OF THE FORWARD NEURAL MODEL OF THE MOBILE ROBOT

We have chosen Gauss-Newton optimization method, which minimized the criterion (19) for training of the forward neural model of the mobile robot:

$$J = \frac{1}{2N} \sum_{i=1}^{n} \left(y(i) - \hat{y}(i) \right)^2 \qquad (19)$$

where N is number of the samples in the set of the training data (Fig.5). For training of the forward neural model, we used a forward neural network of Multi Layer Perceptron (MLP) type with ten neurons in the input layer, with ten neurons in the hidden layer and with two neurons in the output layer. The training of forward neural model was carried out by the Levenberg-Marquardt algorithm using Neural Network Toolbox.

3.3 VALIDATION OF THE FORWARD NEURAL MODEL OF THE MOBILE ROBOT

The validation of the model is the next step after the training of the neural model. For validation of the training model, we used the following simulation scheme:

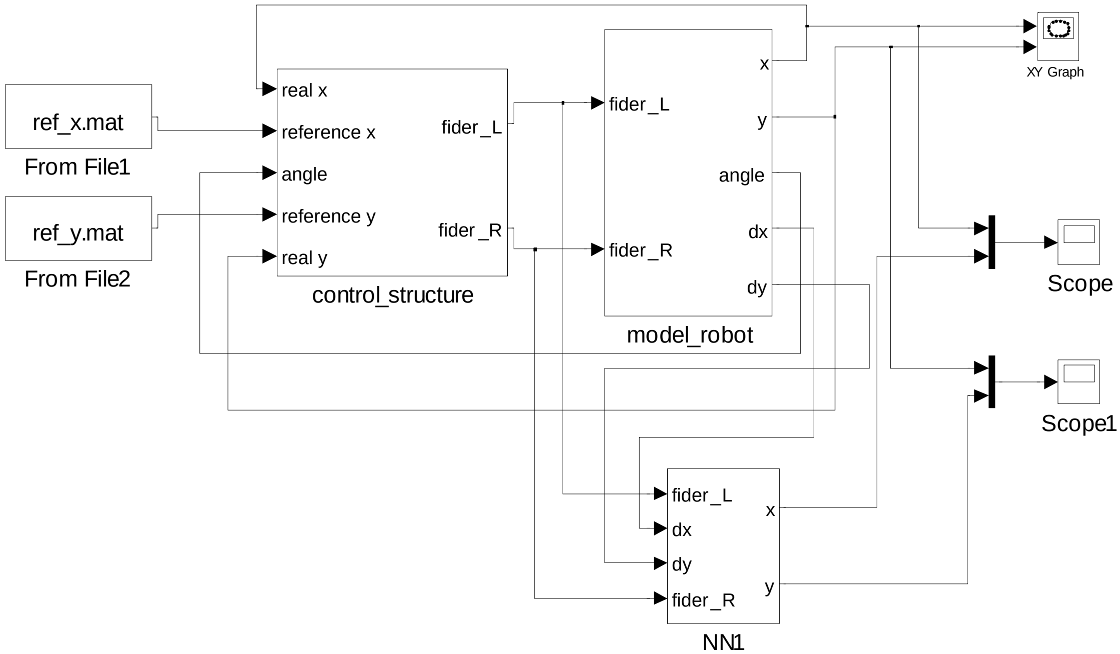

Fig. 9 – Validation of the forward neural model of mobile robot

We supplied a new reference trajectory for the input of the simulation system (Fig.9), which caused that the input of model was supplied different angular velocities than during phase of training. The following pictures are graphs comparing the outputs of the model robot and of the neural model of robot (NN1).

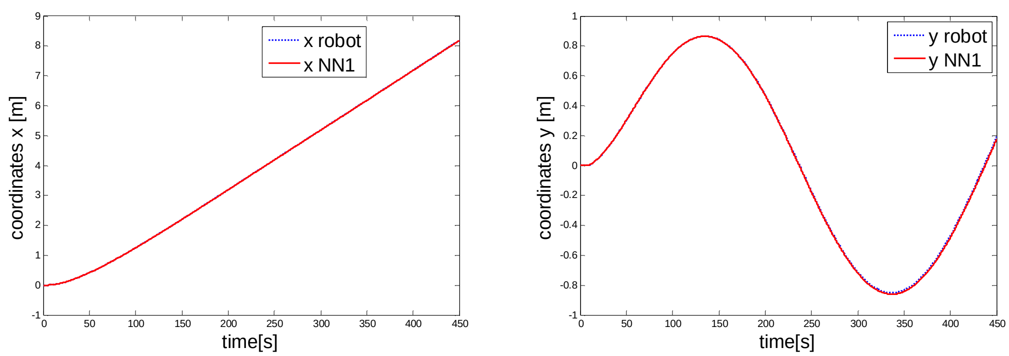

Fig. 10 – Comparison outputs of the system and of the forward neural model

From Fig.10 we can see that our proposed forward neural model approximates dynamic of system of mobile robot even if we changed reference trajectory. Forward neural model (NN1) therefore can be used in the control structure with the internal model (IMC) (part 5).

4 INVERSE NEURAL MODEL OF THE MOBILE ROBOT

Inverse neural model of the system is an important part of the theory of control. If the forward neural model is described by the equation (18), then the inverse model can be expressed in the form:

$$u(k) = f^{-1}[r(k+1), y(k), \ldots, y(k-n+1), u(k), \ldots, u(k-m+1)] \tag{20}$$

where $y(k+1)$ is an unknown value, therefore it is substituted by the reference value of the control variable $r(k+1)$. To obtain inverse neural model, we have chosen General training architecture (Fig.11). Signal $u(k)$ is applied to the input of the structure based on input predictive error with the aim of to obtain a corresponding system output $y(k)$, while the neural network is trained by the error $e_u(k)$, which is obtained as the difference of the neural model output $\hat{u}(k)$ and input signal $u(k)$ into the system [Jadlovská et all., 2002].

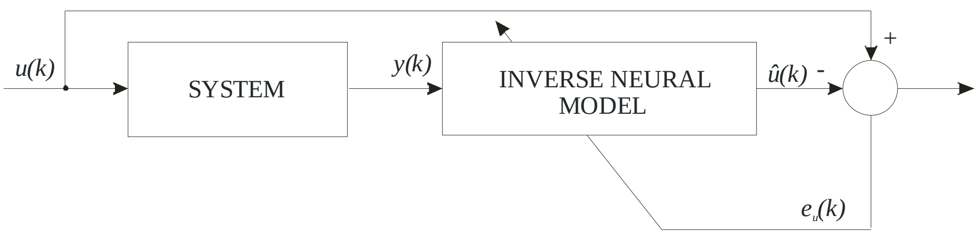

Fig. 11 - General training structure

We have obtained a set of training data for designing an inverse neural model from the following simulation scheme:

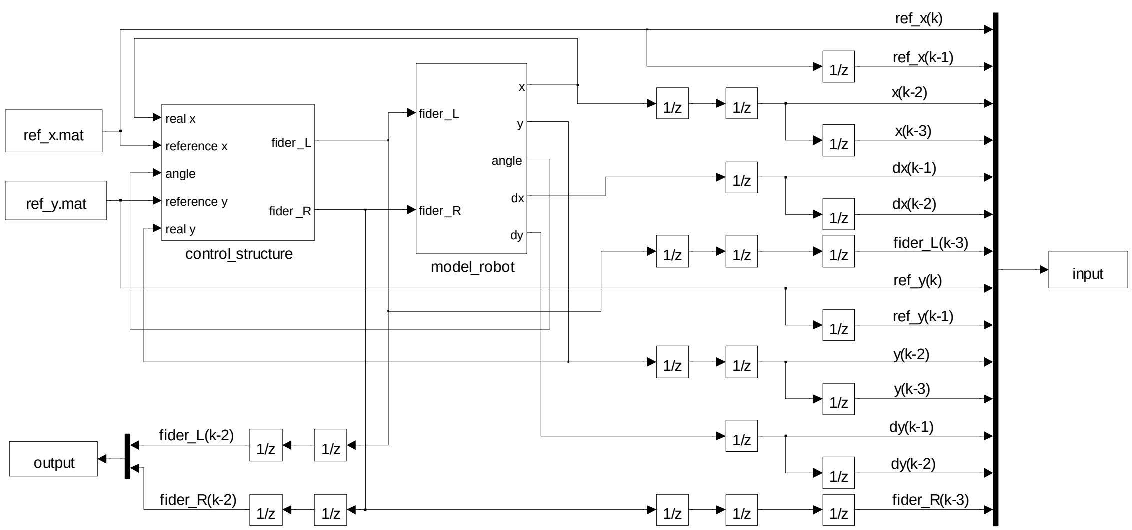

Fig. 12 – Simulation scheme for acquisition of training data for inverse neural model

We supplied to the input of the simulation scheme defined reference trajectory, which was represented by the x and y coordinates. The inputs to the model robot generated by the control structure are shown in the Fig.13. Output from the simulation scheme is a set of training data, which we have used to train inverse neural model in the General training structure. For training the inverse neural model, we have used the forward neural network of Multi Layer Perceptron (MLP) type with fourteen neurons in the input layer, with five neurons in the hidden layer and with two neurons in the output layer. Training of inverse neural model has been carried out by the Levenberg-Marquardt algorithm.

The obtained inverse neural model was applied in the control structure IMC as nonparametric neural controller for tracking defined reference trajectory of mobile robot Khepera II. We have proposed the IMC filter into control structure for better tracking trajectory. The goal of the tracking is to control the movement of mobile robot from the point A to the point B along the chosen reference trajectory (Fig.14).

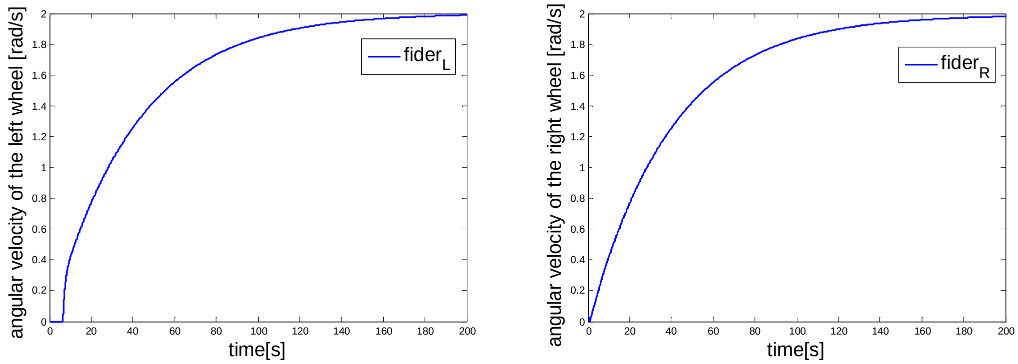

Fig. 13 - The inputs to the system to obtain training data for inverse neural model

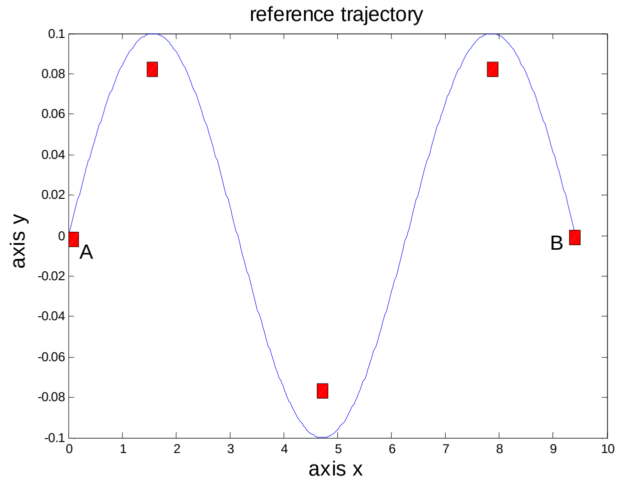

Fig. 14 – The reference trajectory of the mobile robot

5 APPLICATION OF CONTROL STRUCTURE IMC AT THE TRACKING DEFINED REFERENCE TRAJECTORY OF MOBILE ROBOT

We have implemented the obtained forward (part 3), inverse (part 4) neural model and IMC filter into the Internal Control Structure (IMC) (Fig.15) with the aim of verify tracking the defined reference trajectory (Fig.16) [Kajan, 2009].

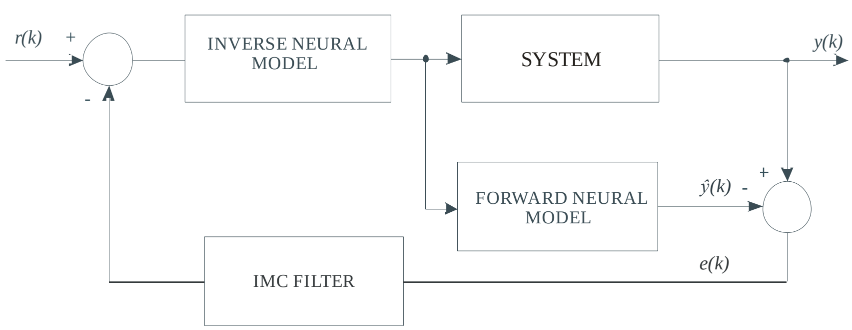

Fig. 15 – The internal model control (IMC) structure

The output from simulation scheme (Fig.16 – XY Graph) is current trajectory of simulation model of the mobile robot (Fig.17) controlled by the nonparametric neural controller (NN2).

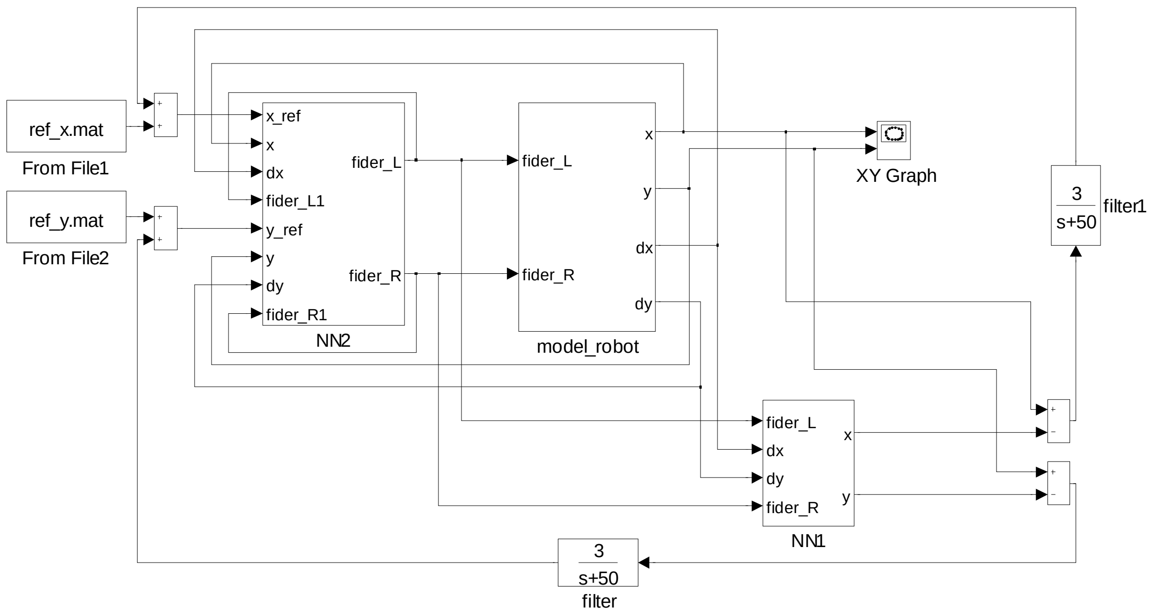

Fig. 16 – Validation of the inverse neural model of the mobile robot in the control structure IMC

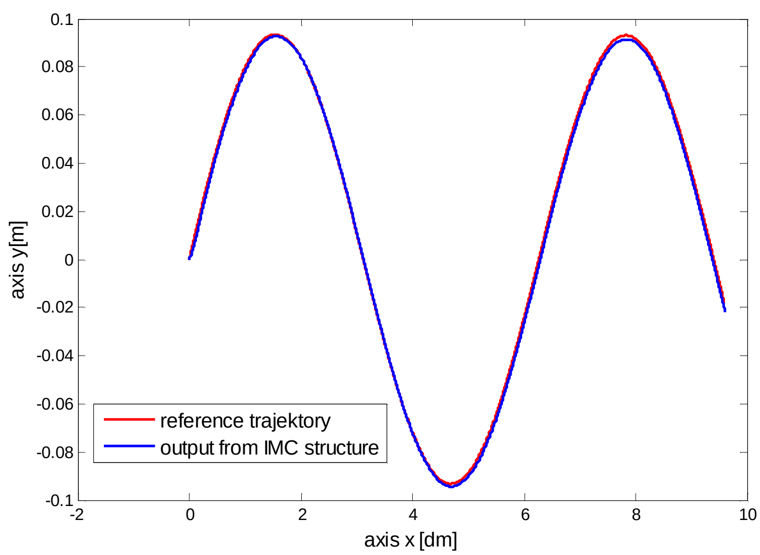

Fig. 17 – Comparison defined reference trajectory and output from IMC structure

From Fig.17, we can see that simulation model of mobile robot Khepera II follows the defined reference trajectory. We verified the functionality the obtained neural models for other sinus trajectories with other amplitudes. If we want to change trajectory, it is necessary to train a new inverse and forward neural model.

CONCLUSION

The aim of this paper was to use intelligent approach for tracking of the defined reference trajectory based on forward neural networks. Mathematical model of the mobile robot, which consists of kinematic and dynamic part, was controlled by the proposed control structure when the acquisition of training data, necessary for proposal forward and inverse neural models. The obtained neural models were implemented into the control structure IMC, by means of which we was able to track of the defined reference trajectory. If we want to change trajectory we must to train the new neural models. The training of neural models is done offline that we can to obtain set of the inverse neural models and then we can use the individual model for tracking of the various types of trajectories. We want to use the obtained knowledge in the field tracking reference trajectory of the mobile robot for real mobile robot Khepera III, which are at disposal in our laboratory at the Department of Cybernetics and Artificial Intelligence.

Acknowledgements

This contribution is the results of the Vega project implementation: Multiagent Network Control System with Automatic Reconfiguration (No.1/0617/08), supported by the Scientific Grant Agency of Slovak Republic and of the project implementation: Centre of Information and Communication Technologies for Knowledge Systems (project number: 26220120020), supported by the Research and Development Operational Program funded by the ERDF.

References

DOMINGUEZ, D. 2007. VRML and Simulink Interface for the Development of 3-D Simulator for Mobile Robots. *Proceedings of world academy of science, Engineering and Technology,* 2007, volume. 25 November 2007. ISSN 1307 - 6884

FIC, J. 2009. *Riadenie mobilného robota Khepera II s využitím metód umelej inteligencie.* Diplomová práca. Košice: Technická univerzita v Košiciach, Fakulta elektrotechniky a informatiky, 2009. 67.

GAJDUŠEK, M.; ŠOLC, F. 2006. Generovaní časově optimální trajektorie pro mobilního robota s diferenciálním řízením. *AT&P Journal*, 2006, No. 2, pp 86-89.

JADLOVSKÁ, A.; SARNOVSKÝ, J. 2002. Aplikácia inverzného neurónového modelu nelineárneho procesu v štruktúre priameho inverzného riadenie. *AT&P Journal*, 2002, No.10, pp.75-77, No. 11, pp.84-86, No.12, pp. 46, 2002, ISSN 1335-2237

KAJAN, S. 2009. Comparison of some neural control structures for nonlinear system. *Journal of Cybernetics and Informatics*, 2009, volume 8 2009. ISSN 1336-4774

KIM, B; TSIOTRAS, P. 2002. Controllers for Unicycle-Type Wheeled Robots: Theoretical Results and Experimental Validation. IEEE Transactions on robotics and automation, 2002, No. 3, pp. 294-307.

KÜHNE, F.; LAGES, W. F.; GOMES da SILVA, J. M. 2005. Mobile robot trajectory tracking using model predictive control. In.: *VII SBAI/II IEEE LARS*, São Luís, september 2005.

K-TEAM. Dostupné na: <http://www.k-team.com >.

MASÁR, I. 2007. Inteligentný regulátor na sledovanie trajektórie mobilným robotom. *Automatizace*, 2007, No 2, pp. 80-85.

ŠEMBERA, J.; ŠOLC, F. 2007. Modelovaní a řízení mobilního robotu s diferenciálním podvozkem. *AT&P Journal PLUS*, 2007, No. 1, pp 203-207.