T H E A R C H I V E O F M E C H A N I C A L E N G I N E E R I N G

VOL. LVI 2009 Number 3

DOI: 10.24425/ame.2009.132098

Key words: wheeled mobile robot, approximate dynamic programming, reinforcement learning, robust tracking control, actor-critic structure

ZENON HENDZEL [*], *MARCIN SZUSTER* [**]

APPROXIMATE DYNAMIC PROGRAMMING IN ROBUST TRACKING CONTROL OF WHEELED MOBILE ROBOT

In this work, a novel approach to designing an on-line tracking controller for a nonholonomic wheeled mobile robot (WMR) is presented. The controller consists of nonlinear neural feedback compensator, PD control law and supervisory element, which assure stability of the system. Neural network for feedback compensation is learned through approximate dynamic programming (ADP). To obtain stability in the learning phase and robustness in face of disturbances, an additional control signal derived from Lyapunov stability theorem based on the variable structure systems theory is provided. Verification of the proposed control algorithm was realized on a wheeled mobile robot Pioneer–2DX, and confirmed the assumed behavior of the control system.

1. Introduction

Nonholonomic mobile robots are control objects with nonlinear dynamics. Although there exists a systematic approach to nonlinear control design, namely optimal control theory [6], in most cases it is not on-line applicable. This main drawback of classical optimal control is determined by its high computational effort and usually backstepping design of the control law which is unsuitable for feedback control. In the literature, reinforcement learning was proved to be a powerful tool for nonlinear control overcoming limitations of classical optimal control techniques [1, 2, 3, 9]. The main advantage of ADP control design is the fact that, although it is based on the optimal control theory, the control signal is computed forward, from the first step of the discrete process, in comparison to classical method of

[*] *University of Technology, 8 Powstańców Warszawy St., 35-959 Rzeszów, Poland; E-mail: zenhen@prz.edu.pl*

[**] *University of Technology, 8 Powstańców Warszawy St., 35-959 Rzeszów, Poland; E-mail: mszuster@prz.edu.pl*

the Bellman's dynamic programming, where optimal control signal can be calculated only backward in time (from the last step of the discrete process to the first step).

The paper is concerned with the application of ADP for tracking control of WMR. The objective of the presented approach is to construct a control system capable of converging to an optimal adaptive control policy for a WMR. This solution is based on the BSA reinforcement learning method of actor-critic architecture [1, 9] for continuous-time, continuous state [2]. The presented algorithm does not require pre-learning and works on-line without robot model knowledge. Stability of the control system is achieved by an additional supervisory control, derived from Lyapunov stability theory [7]. The results of research presented in the article continue the authors' earlier works related to synthesis of wheeled mobile robots control algorithms using neural networks, fuzzy logic and reinforcement learning methods for WMR nonlinearities compensation.

The paper is organized as follows. Section 1 is a short introduction into the control of WMR. Section 2 presents a dynamical model of a two-wheeled mobile robot. Section 3 formulates and describes the synthesis of a tracking control algorithm of actor-critic architecture for nonlinear dynamics of a WMR. Section 4 presents the stability analysis of a closed control system and, based on this analysis, a synthesis of a controller. Section 5 contains results of verification experiment realized on the wheeled mobile robot Pioneer–2DX. Section 6 concludes and summarizes the research project.

2. Model of 2-wheeled mobile robot dynamics

We analyzed the movement of a wheeled mobile robot in the *xy* plane [4]. The schematic diagram of a 2-wheeled mobile robot (with a third, free rolling castor wheel) is shown in Fig. 1.

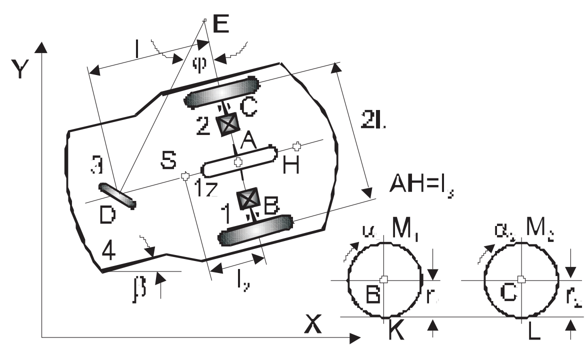

Fig. 1. Diagram of a 2-wheeled mobile robot (with a third, free rolling castor wheel)

In a mathematical model of a mobile robot we include dynamics equations of a WMR movement and properties of a driving system composed of two DC motors with permanent magnets [8]. We also take into account couplings between motors and driver wheels by reduction gears with a ratio R. Taking the generalized coordinates vector of a wheeled mobile robot as

$$q = \alpha \tag{1}$$

where $\alpha = [\alpha_1, \alpha_2]^T$, α_i – angle of self-turn driver wheels, i = 1, 2, and using Maggie's formalism [4], the dynamics of WMR can be written as

$$M_r\ddot{q} + C_r(\dot{q})\dot{q} + F_r(\dot{q}) = u \tag{2}$$

where M_r is the inertia matrix of WMR, $C_r(\dot{q})$ is the Coriolis/centripetal matrix, $F_r(\dot{q})$ are the rolling friction terms and u represents the wheel-driving moments. The electric motors dynamics is given by

$$J_s\ddot{\varphi}_s + B_s\dot{\varphi}_s + Ru = k_sU \tag{3}$$

$\varphi_s \in \Re^2$, where vector $\varphi_s = [\varphi_{s_1}, \varphi_{s_2}]^T$ contains angles of self-turn DC motor shafts, matrix J_s contains the motor inertias, B_s contains the rotor damping constants and back emf., and R is a diagonal matrix containing the gear ratios. The control input is the motor voltage $U \in \Re^2$, with k_s the diagonal matrix of motor torque constants.

Using relations $\alpha = R\varphi_s$, $\|R_i\| < 1$, the torque scales conversely as $u = R^{-1}\tau_s$. This effect is already included in (2). Taking into account DC motors dynamics (3) we can write the dynamical equations of a wheeled mobile robot motion in the form

$$\left(J_s + R^2M_r\right)\ddot{q} + R^2C_r(\dot{q})\dot{q} + R^2F_r(\dot{q}) + B_s(\dot{q}) = Rk_sU \tag{4}$$

We can write equation (4) in a short form

$$M\ddot{q} + C_m(\dot{q})\dot{q} + F(\dot{q}) = U \tag{5}$$

The objective of a mobile robot control system is to select the control signal U, that guarantees realization of a tracking movement for a prescribed motion trajectory $q_d = [\alpha_{1d}, \alpha_{2d}]^T$. Define the tracking error e(t) and filtered tracking error s(t) by [5],[7]

$$e = q - q_d \tag{6}$$

$$s = \dot{e} + \Lambda e \tag{7}$$

with Λ a positive definite design parameter matrix.

Differentiating (7) and substituting into (5) we obtain the mobile robot dynamics expressed in terms of the filtered error

$$M\dot{s} = -C_m(\dot{q})s - f(x) + U \tag{8}$$

where $f(x)$ is unknown non-linear function defined as

$$f(x) = M(\ddot{q}_d - \Lambda\dot{e}) + C_m(\dot{q})(\dot{q}_d - \Lambda e) + F(\dot{q}) \tag{9}$$

where, assuming $\dot{v} = \ddot{q}_d - \Lambda\dot{e}, v = \dot{q}_d - \Lambda e$, vector x contains all variables necessary to calculate f(x), $x = \left[q^T, \dot{q}^T, v^T, \dot{v}^T\right]^T$. A sort of approximation-based controller is derived by setting

$$U = \hat{f} - Ks + u_s \tag{10}$$

with $\hat{f}(x)$ being an estimate of f(x), $Ks = K\dot{e} + K\Lambda e$ is an outer PD loop, $K = K^T$ – positive matrix and $u_s(t)$ an auxiliary signal to provide stability and robustness in the face of disturbances and modeling errors. In the presented paper the estimation of $\hat{f}(x)$, which is an approximate of non-linear function f(x), is implemented by ASE.

3. Control algorithm with actor-critic architecture

Reinforcement learning designs constitute a class of *approximate* dynamic programming methods [3, 9] that use incremental optimization combined with a parametric structure to efficiently approximate the optimal cost and control. They optimize a short-term cost metric that ensures optimization of the cost over all future times. Neural networks are the parametric structures of choice, because they easily handle large-dimensional input and output spaces and can learn in an incremental fashion. The simplest RL architectures are based on ADP. They implement a critic network (ACE – Adaptive Critic Element) to approximate the cost-to-go in the Bellman equation [3, 6] and an action network (ASE – Associate Search Element) to approximate the optimal control law.

This paper presents the design of nonlinear compensator based on a modification of BSA algorithm [1, 2]. In Fig. 2, diagram of a wheeled mobile robot tracking control system is shown. Specifically, the control input U mainly consists of an ASE which generates the signal u_{RL}, a fixed gain controller Ks and a supervisor term u_s.

Based on the architecture of reinforcement learning control algorithm BSA and its modification for continuous-time, continuous-state [2], we em-

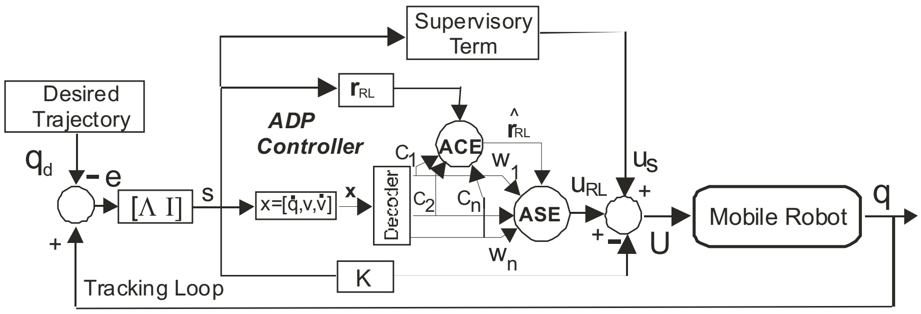

Fig. 2. Mobile robot tracking control system with actor-critic architecture

ploy two layer neural network (NN) as two parametric structures [5, 7]. The output of the two-layer NN is given by the equation

$$y = W^T \Gamma\left(V_{RL}^T x\right) \tag{11}$$

where $\Gamma(.)$ is a vector of neurons' activation functions. If the first-layer weights V_{RL} are predetermined by some a priori method, then only the second-layer weights W are considered to define the NN. Our goal is to find the control law

$$u_{RL} = \hat{f}(x) \tag{12}$$

for the dynamical system (8), that optimizes the expected reinforcement for a certain period in the future. We define the value function as [2]

$$V(s(t)) = \int_t^{\infty} \frac{1}{\tau_c} e^{\frac{\tau - t}{\tau_c}} r_{RL}(s(\tau), u(\tau)) d\tau \tag{13}$$

where constant τ_c is introduced for normalization [2], value of τ_c satisfies equation $\int_0^{\infty} \frac{1}{\tau_c} e^{-\frac{\tau - t}{\tau_c}} d\tau = 1$, and $r_{RL}(t)$ is the immediate reinforcement signal. The filtered tracking error s(t) is the performance measure, that can be viewed as the real-valued reinforcement of the WMR performance. When s(t) is small, system performance is good. The process of generating the bounded reinforcement signal $r_{RL}(t)$ is defined as [7]

$$r_{RL_k}(t) = \frac{\rho_k}{1 + e^{-b_k s_k(t)/\rho_k}} - \frac{\rho_k}{1 + e^{b_k s_k(t)/\rho_k}}, \quad k = 1, 2 \tag{14}$$

where the value of r_{RL_k} is bounded in the interval $[-\rho_k, \rho_k]$ with boundary value $\rho_k > 0$, whereas positive constant b_k makes smooth transitions between boundary values of reinforcement signal $r_{RL}(t)$ possible. ADP designs are based on an algorithm that cycles between a control law-improvement routine and a value function-determination operation. A local consistency condition for the value function is as follows [2]

$$\tau_c \dot{V}(s(t)) = V(s(t)) - r_{RL}(t) \tag{15}$$

Let P(t) be the prediction of the value function. If the prediction is correct, then the following equation should be satisfied

$$\tau_c \dot{P}(t) = P(t) - r_{RL}(t) \tag{16}$$

If the prediction is not accurate, prediction error is determined as

$$\hat{r}_{RL}(t) = r_{RL}(t) - P(t) + \tau_c \dot{P}(t) \tag{17}$$

The critic is a parametric structure P(t) with some weights C that is used to approximate prediction of the value function and generates a signal in the form

$$P(t) = C^T \phi \tag{18}$$

where

$$\phi = \Gamma\left(V_{RL}^T x\right) \tag{19}$$

for V_{RL} – fixed matrix of NN first layer weights randomly chosen in initialization process, $\Gamma(.)$ – vector of neurons' activation functions.

Critic weights adaptation is performed according to the relation

$$\dot{C}_i(t) = \kappa \hat{r}_{RL}(t) e_{c_i}(t) \tag{20}$$

and activity path e_{c_i} is described as follows

$$\dot{e}_{c_i}(t) = (1-\lambda)\phi_i(t) - (1-\lambda)e_{c_i}(t) \tag{21}$$

where $\phi = [\phi_1, ..., \phi_n]^T$, n is a number of neurons in NN hidden-layer, κ denotes the learning rate and λ determines the trace decay rate ($0 \leq \lambda \leq 1$).

The control signal u_{RL} is given by the ASE output which is of the form

$$u_{RL} = \delta \Gamma(W^T \phi) \tag{22}$$

where δ – scaling factor of actor output. By analogy, the law of actor weight adaptation takes the form

$$\dot{W}_i(t) = \xi \hat{r}_{RL}(t) e_{a_i}(t) \tag{23}$$

where ξ is real valued constant and the activity trace is define by the relation

$$\dot{e}_{a_i}(t) = (1 - \lambda)\phi_i(t) - (1 - \lambda)e_{a_i}(t) \tag{24}$$

In typical approach, ADP addresses the problem of an agent that must learn behavior through trial-and-error interactions in a dynamic environment [1, 2]. In tracking control of WMR such solution is not acceptable. The main features of the developed schemes are that no trials are required to train the controllers because of dropping the stochastic unit in equation (22). To obtain stability in the learning phase and robustness in face of disturbances, an additional control signal derived from Lyapunov stability theorem based on the variable structure systems theory is provided. In the following, the details of the stability analysis will be discussed.

4. Stability analysis of a closed loop control system

Now, our task is to ensure closed loop control system stability without changing the control signal $u_{RL}(x)$, designed previously. This means that the control signal U must be designed in the way providing the stability of the closed loop system, i.e. the filtered tracking error s(t) must by uniformly bounded, which is equivalent to

$$|s_i| \leq \varepsilon_i \forall t > 0 \tag{25}$$

where ε_i, i = 1, 2, are constants. For this purpose, we supplement the existing control signal with the supervisory term u_S [10], which is non-zero if the state vector reaches the boundaries of the set

$$\{s_i : |s_i| \leq \varepsilon_i, i = 1, 2\} \tag{26}$$

For the thus stated problem we introduce the control vector

$$U = u_{RL} - Ks + I^* u_S \tag{27}$$

where I_i^* is an indicator and is equal to 1, when $|s_i| \geq \varepsilon_i$ and $I_i^* = 0$, when $|s_i| < \varepsilon_i$. Inserting (27) into (7), we get a description of the closed loop system in the form

$$M\dot{s} = -C_m(\dot{q})s - f(x) - Ks + u_{RL} + I^* u_S \tag{28}$$

Let us consider a positive definite function

$$L = \frac{1}{2} s^T M s \tag{29}$$

Using (28) and considering the case $|s_i| \geq \varepsilon_i$, we have

$$\dot{L} = s^T M\dot{s} + \frac{1}{2}s^T\dot{M}s = -s^T Ks + \frac{1}{2}s^T\left(\dot{M} - 2C_m\right)s + s^T\left[f(x) + u_{RL}\right] + s^T u_S \quad (30)$$

Assuming that $\dot{M} - 2C_m$ is a skew-symmetric matrix, we obtain

$$\dot{L} \leq -s^T Ks + \sum_{i=1}^{2} |s_i| \left[|f_i(x)| + |u_{RLi}|\right] + \sum_{i=1}^{2} s_i u_{si} \quad (31)$$

If we choose supervisory control signal in the form

$$u_{Si} = -\left[F_i + |u_{RLi}| + \eta_i\right] \operatorname{sgn} s_i \quad (32)$$

where it was assumed that

$$|f_i(x)| \leq F_i \quad i = 1,2 \quad F_i > 0, \eta_i > 0 \quad (33)$$

finally we get the result

$$\dot{L} \leq -\sum_{i=1}^{2} \eta_i |s_i| - s^T Ks \quad (34)$$

Derivative of the Lyapunov function is negative definite, thus the closed system (28) is asymptotic stable with respect to filtered tracking error s(t).

5. Verification of the proposed control algorithm

The proposed actor-critic tracking control algorithm was verified on the experimental system shown in Fig. 3., which consists of a PC computer with dSpace Control Desk and Matlab/Simulink software, DS1102 digital signal processing (DSP) board and the two-wheeled mobile robot Pioneer-2DX. The DS1102 controller board is an ISA PC card based on Texas Instrument TM320C31 – 60 MHz third generation floating point DSP which builds the main processing unit providing fast instruction cycle for numerically intensive algorithms. It contains 128K memory fast enough to allow zero wait state operation and several I/O peripheral subsystems implemented. Pioneer-2DX is a two-wheeled, computer controlled mobile robot, for research and unknown environments exploration. It is equipped with Siemens 88C166 20MHz on-board programmable microcontroller, which handles the low-level tasks, such as sensor processing, communication and motors control. Mobile robot is equipped with 8 ultrasonic range finders for map-building processes and two DC 12V motors coupled with gears and position encoders (9850

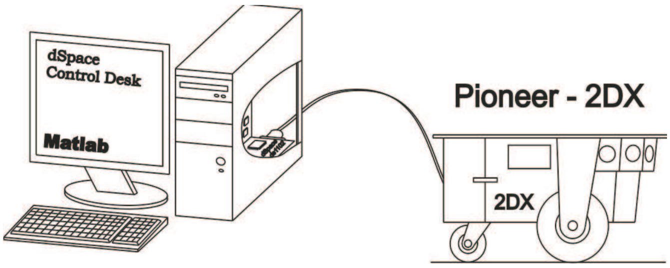

Fig. 3. Scheme of a work station

ticks per revolution). Maximal linear velocity of the Pioneer-2DX mobile robot is equal to $v_{A\max} = 1.6[m/s]$.

Verification of the proposed controller was realized for movement of a chosen point A of a mobile robot on a desired trajectory in a form of a loop [4]. The desired trajectory consists of five characteristic stages:

a) *starting, linear motion*:

$$v_A^* = \frac{v_A}{t_r}\left(t - t_p\right) \qquad t_p \le t \le t_r, \qquad \dot{\alpha}_1 = \dot{\alpha}_2 = \frac{v_A^*}{r} \qquad \dot{\beta} = 0$$

where: $\dot{\beta}$ – an angular velocity of WMR frame self-turn, t_p – starting time, t_r – motion duration, $r = r_1 = r_2$ – radius of the driving wheel

b) *movement with constant velocity, when $v_A = const$*:

$$\dot{\alpha}_1 = \dot{\alpha}_2 = \frac{v_A}{r} \qquad t_r \le t \le t_1 \qquad \dot{\beta} = 0$$

where: t_1 – time of a movement with constant velocity,

c) *curvilinear motion on a circular track with radius l_R, for*:

$$\mathrm{v_A} = \text{const},\ \mathrm{l_R} = 0.75[\mathrm{m}], \quad t_1 \le t \le t_2, \quad \dot{\alpha}_1 = \frac{v_A}{r} + \frac{l_1}{r}\dot{\beta}, \quad \dot{\alpha}_2 = \frac{v_A}{r} - \frac{l_1}{r}\dot{\beta}$$

where: l_1 – results from WMR geometry, t_2 – end of a curvilinear motion,

d) *end of the curvilinear motion taking into account a transitory period, then linear motion with constant velocity* ($\mathrm{v_A}$ *=const)*:

$$\dot{\alpha}_1 = \dot{\alpha}_{10} - \left(\dot{\alpha}_{10} - \frac{v_A}{r}\right)\left(1 - e^{-\gamma t}\right), \quad \dot{\alpha}_2 = \dot{\alpha}_{20} - \left(\frac{v_A}{r} - \dot{\alpha}_{20}\right)\left(1 - e^{-\gamma t}\right) \quad t_2 \le t \le t_3$$

where: t_3 – time of the transitory period, γ – constant of approximation of transition curves, $\dot{\alpha}_{10}, \dot{\alpha}_{20}$ – values of angular velocities of driving wheels at the beginning of the transitory period. The approximation of transition curves in the transitory period, make smooth realization of trajectory (velocity and acceleration) possible.

e) *breaking*:

$$v_A^* = v_A - \frac{v_A}{t_h}(t - t_3), \qquad t_3 \leq t \leq t_k, \qquad \dot{\alpha}_1 = \dot{\alpha}_2 = \frac{v_A^*}{r}, \qquad \dot{\beta} = 0$$

where: t_k – end of simulation time, t_h – breaking time.

In the presented experiment, we assumed linear acceleration time equal to retardation time, $t_r = t_h$, and maximal linear velocity of a chosen point A of a mobile robot equal to $v_A = 0.4$[m/s]. Fig. 4a shows the desired angles of a self-turn α_{1d} and α_{2d} for driving wheels, Fig. 4b desired angular velocities $\dot{\alpha}_{1d}$ and $\dot{\alpha}_{2d}$ for wheels 1 and 2, Fig. 4c desired angular accelerations $\ddot{\alpha}_{1d}$, $\ddot{\alpha}_{2d}$, and Fig. 4d the desired path of point A movement.

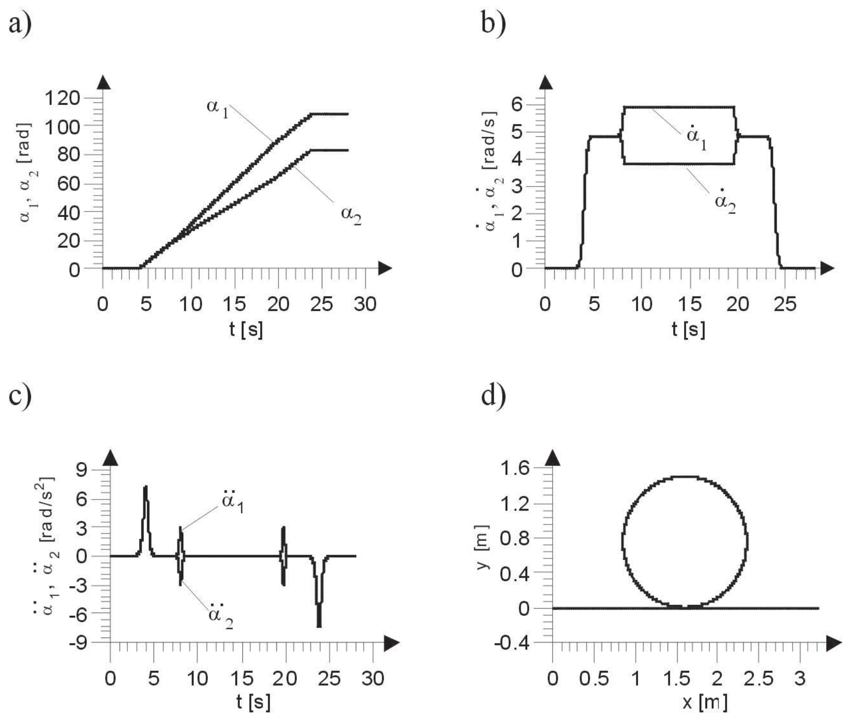

Fig. 4. a) Desired angles of a self-turn α_{1d} and α_{2d} [rad] for driving wheels, b) desired angular velocity $\dot{\alpha}_{1d}$ for wheel 1 and $\dot{\alpha}_{2d}$ for wheel 2 [rad/s], c) desired angular accelerations $\ddot{\alpha}_{1d}$ and $\ddot{\alpha}_{2d}$ [rad/s^2], d) desired path of the point A movement

Verification of the proposed algorithm was realized on the wheeled mobile robot Pioneer-2DX. The results obtained from experiment for the chosen movement parameters and control signals of the designed controller are shown in Fig. 5. and Fig. 6. Control signal U, shown in Fig. 5a, according to the assumed control law (27) consists of ACE-ASE control signal u_{RL} (Fig. 5b), supervised control signal u_S and PD control signal u_{PD} = Ks

(Fig. 5c). Reinforcement signals r_1 and r_2, used for ASE weights tuning, are shown in Fig. 5d.

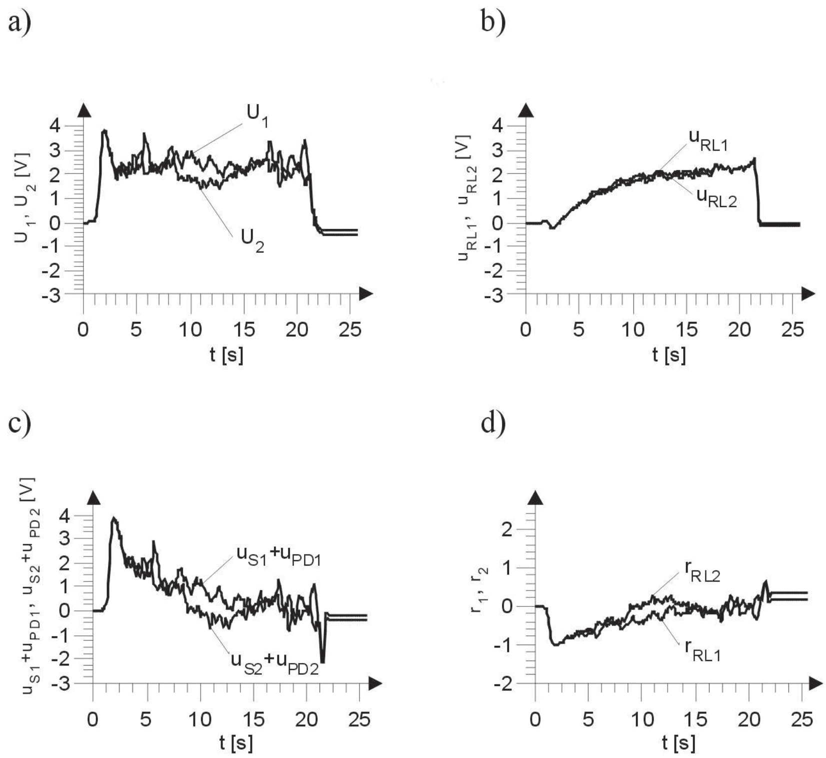

Fig. 5. a) Control signal: U_1 for 1. wheel and U_2 for 2. wheel, b) control signal u_{RL1} and u_{RL2}, generated by actor (ASE) neural network, c) control signal generated by supervised term and PD controller (u_{PD} = Ks), $u_{S1} + u_{PD1}$ and $u_{S2} + u_{PD2}$, d) reinforcement signal r_{RL_1} and r_{RL_2}

In Fig. 6a and Fig. 6b there are shown angle of self-turn tracking errors e_1 and e_2 respectively, as well as angular velocity tracking errors $\dot{e}_1$ and $\dot{e}_2$. The obtained values of errors are bounded. Fig. 6c shows desired and realized angular velocities, Fig. 6d desired and realized path of the point A movement.

In neural networks initialization process, values of actor-critic structure weights were chosen as equal to zero, and then adapted in a learning process. Weights W of actor (ASE) neural network, shown in Fig. 7a, are bounded and converge to fixed values after the learning process, similarly as critic (ACE) weights C, shown in Fig. 7b.

For numerical rating of the realized experiment several quality ratings were chosen (with an index 1 for first wheel, and 2 for second wheel):

– maximal value of a tracking error: $e_{\max 1}$ [rad], $e_{\max 2}$ [rad],

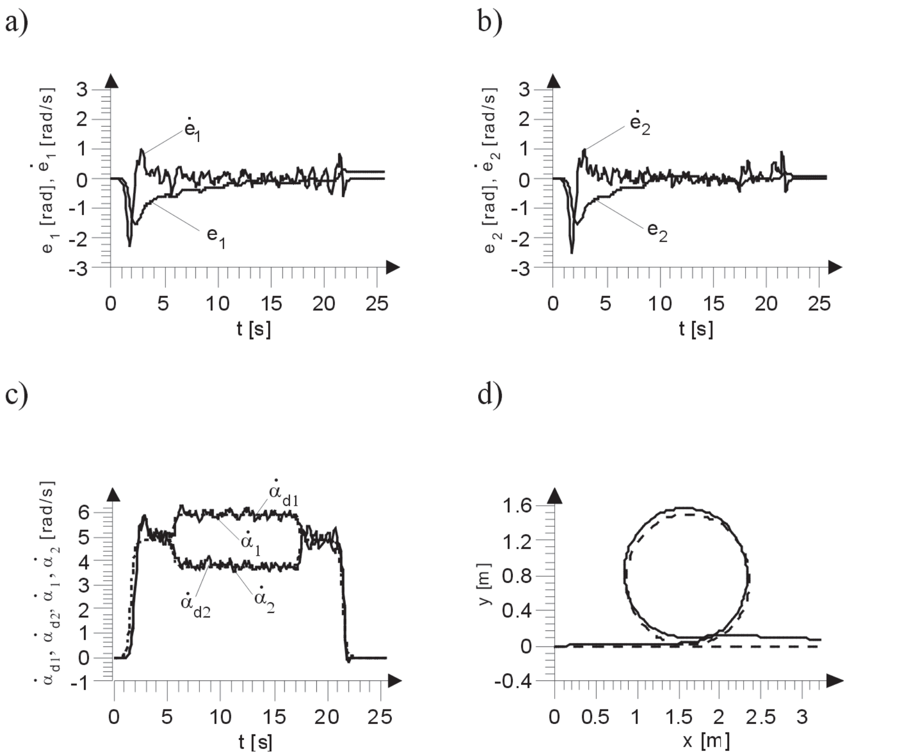

Fig. 6. Angle of self-turn tracking error $e_1 = \alpha_1 - \alpha_{1d}$ [rad] and angular velocity tracking error $\dot{e}_1 = \dot{\alpha}_1 - \dot{\alpha}_{1d}$ [rad/s] for 1. wheel, b) angle of self-turn tracking error e_2 [rad] and angular velocity tracking error $\dot{e}_2$ [rad/s] for 2. wheel, c) desired ($\dot{\alpha}_{1d}, \dot{\alpha}_{2d}$ – dashed line) and realized ($\dot{\alpha}_1, \dot{\alpha}_2$) angular velocities, d) desired (dashed line) and realized path of the point A movement

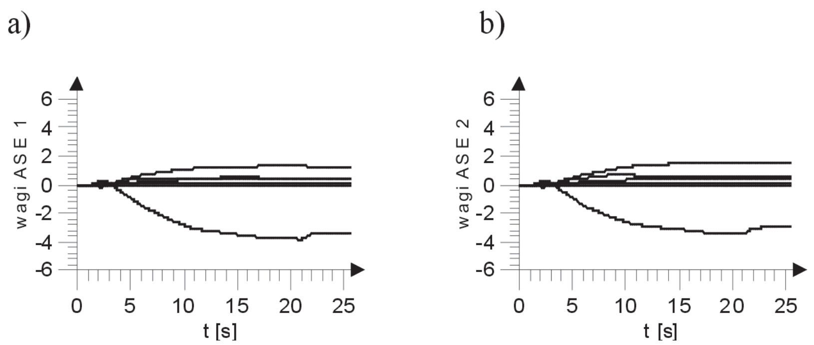

Fig. 7. Weights W of actor (ASE_1) neural network for wheel 1, b) weights C of critic (ACE_1) neural network for wheel 1

– maximal value of an angular velocity tracking error: $\dot{e}_{\max 1}$ [rad/s], $\dot{e}_{\max 2}$ [rad/s],
– root mean square error of tracking errors e_1 and e_2, defined as:

$\varepsilon_1 = \sqrt{\frac{1}{n}\sum_{k=1}^{n} e_{1k}^2}$ [rad], $\varepsilon_2 = \sqrt{\frac{1}{n}\sum_{k=1}^{n} e_{2k}^2}$ [rad], where k – an index of following iteration steps, n = 257 – number of iteration steps,
– root mean square error of an angular velocity errors:
$\dot{\varepsilon}_1 = \sqrt{\frac{1}{n}\sum_{k=1}^{n} \dot{e}_{1k}^2}$ [rad/s], $\dot{\varepsilon}_2 = \sqrt{\frac{1}{n}\sum_{k=1}^{n} \dot{e}_{2k}^2}$ [rad/s]. Values of quality ratings are shown it Tab. 1.

Table 1.

Values of quality ratings

Wheel:	$e_{\max}$ [rad]	$\dot{e}_{\max}$ [rad/s]	ε [rad]	$\dot{\varepsilon}$ [rad/s]
1.	1.525	2.318	0.428	0.389
2.	1.513	2.494	0.388	0.397

The results of verification for the proposed controller confirmed correctness of the assumed control algorithm.

6. Conclusions

In this study, we have conducted the synthesis of a tracking control algorithm of a two-wheeled mobile robot, based on reinforcement learning with an actor-critic architecture. Unlike traditional reinforcement learning approaches, we adopted an NN based ASE to approximate the nonlinear dynamics of a WMR. The algorithm works on-line without knowledge of the robot model and can prevent time consuming trial and error learning. Stability of the closed loop control system is achieved by the additional supervisory control, derived from the Lyapunov stability theory. Simulation results for the proposed controller demonstrate that the control objective can be achieved effectively and successfully.

This research was supported by MEiN Grant No. 4 T07A 030 29.

Manuscript received by Editorial Board, May 05, 2009;
final version, August 28, 2009.

REFERENCES

[1] Barto A. G., Sutton R. S., Anderson C. W.: Neuronlike adaptive elements that can solve difficult learning control problems, IEEE Transactions on Systems, Man and Cybernetics, 1985, Vol. 13, No. 5, pp. 834-846.
[2] Doya K.: Reinforcement learning in continuous time and space, Neural Computation, 2000, Vol. 12, No. 1, pp. 243-249.

[3] Ferrari S., Stengel R. F.: Model-Based Adaptive Critic Design, in Handbook of Learninkg and Approximate Dynamic Programming, Willey – IEEE Press, 2004, pp. 65-95.
[4] Giergiel J., Hendzel Z., Żylski W.: Modelling and Control of Wheeled Mobile Robots, Warsaw, WNT, (in Polish), 2002.
[5] Hendzel Z.: An adaptive critic neural network for motion control of a wheeled mobile robot, Nonlinear Dynamics, 2007, Vol. 50, No. 4, pp. 849-855.
[6] Levis F. L., Syrmons V. L.: Optimal Control, New York, Willey, 1995.
[7] Levis F. L., Liu K., Yesildirek A.: Neural net robot controller with guaranteed tracking performance, IEEE Transaction on Neural Networks, 1995, Vol. 6, No. 3, pp. 703-715.
[8] Spong M. W., Vidyasagar M.: Robot modeling and control, Warsaw, WNT, (in Polish), 1997.
[9] Sutton R. S., Barto A. G.: Reinforcement Learning: An Introduction, Cambridge, MIT Press, 1998.
[10] Wang L. X.: A Course in Fuzzy Systems and Control, New York, Prentice-Hall Inc., 1997.

Zastosowanie adaptacyjnego programowania dynamicznego w sterowaniu ruchem mobilnego robota kołowego

S t r e s z c z e n i e

W pracy przedstawiono nowe ujęcie problematyki sterowania nadążnego mobilnym robotem dwukołowym. Algorytm bazuje na metodzie uczenia ze wzmocnieniem o strukturze aktor-krytyk i nie wymaga uczenia wstępnego, działa on-line bez znajomości modelu robota. Element generujący sterowania (aktor – ASE) oraz element generujący sygnał wewnętrznego wzmocnienia (krytyk – ACE) są zrealizowane w postaci sztucznej sieci neuronowej (SN). Prezentowany algorytm sterowania zweryfikowano na rzeczywistym obiekcie, dwukołowym robocie mobilnym Pioneer-2DX. Badania potwierdziły poprawność przyjętego rozwiązania.