Mechanics and Mechanical Engineering
Vol. 15, No. 4 (2011) 21–29

Neural Control of a Mobile Robot Amigobot

Andrzej Burghardt
Zenon Hendzel
Rzeszów University of Technology
Department of Applied Mechanics and Robotics
andrzejb@prz.edu.pl, zenhen@prz.edu.pl

Received (16 December 2011)
Revised (17 January 2012)
Accepted (23 February 2012)

In the paper, the author provides a proposal of a quick prototyping environment for algorithms controlling homogenous, autonomous mobile robots. The solution proposed enables monitoring of the operation of a group of mobile robots consisting of the recording of motion parameters and signals from distance detectors as well as of controlling the drive units. Furthermore, the solution is flexible, enabling work with various numbers of robots as well as modification of their settings, and conforms to the real time system requirements assumed. In the course of the development process, AmigoBot mobile robots manufactured by ActiveMedia Robotics were used. The control and measuring environment proposed has been verified based on the sample behavioural task of "running in the middle of a free space". A neural algorithm of a mobile robot's AmigoBot movement control was proposed in the project. A control synthesis was carried out with the use of a mathematic model of the robot; a system's stability was proved in a Lapunow sense. In a control process artificial neural networks were used, linear because of weight. A received solution was verified with the use of a real mobile robot in an author control–measuring environment.

Keywords: Autonomous mobile robots, rapid prototyping, robot control

1. Introduction

The mobile robots are nonholonomic systems, nonlinear with complex dynamics, therefore appropriate control systems should be applied in order to realise properly generated movement's trajectory. Classic control methods are optimal control methods and various algebraic methods that require knowledge of a mathematic model of on object and parameters of this model. Their precise determining is often a very hard task, especially when the object's characteristics change during the work. For instance, resistance to motion undergoes a change depending on a base on which the robot moves. That is why models applied in practice describe parameters of the objects only approximately.

In a control theory, for some time, design methods of robust and adaptive control systems are developed, which does not require an exact knowledge of the object's model and take into account occurring of uncertainties in an open way. In the adaptive control method a control algorithm is designed in a way to make it able to modify independently its characteristics with changing work conditions.

Along with the mentioned methods applied to design control systems in uncertain conditions, what become more and more popular are formalisms defined as an artificial intelligence – neural networks and systems with a fuzzy logic. The artificial neural networks, because of an ability to approximate any nonlinear mapping and learning and adaptation, became an attractive tool applied in an analysis of nonlinear systems.

A neural algorithm of a mobile robot's AmigoBot movement control was proposed in the project. A proposed solution was verified with a use of an author environment of quick prototyping and a mobile AmigoBot robot.

2. Modelling

Mobile wheel robot, which physical parameters were used in a modelling process, is a laboratory AmigoBot robot (Fig. 1a). A model schematically showed in Fig. 1b was assumed to modelling dynamic characteristics. This model consists of: frame 3, two driving wheels 1 and 2 and free rolling castoring wheel 4.

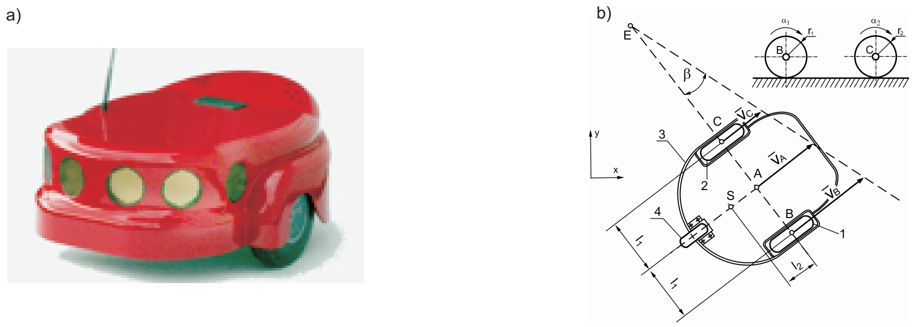

Figure 1 a) Mobile robot Amigobot, b) model of mobile robot

S point is the frame's centre of inertia, the frame's angle of rotation is defined as β, while angles of wheel's rotation with $r_1 = r_2 = r$ radiuses are defined as α_1 and α_2. A description of the mobile wheel robot's movement was realised on the assumptions that there is a lack of the driving wheels' skid and that the robot moves on a horizontal road.

The description of the mobile wheel robot's dynamics is written down generally in a vector-matrix form [2, 3, 4, 5].

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + F(\dot{q}) = Q \tag{1}$$

where:

M – matrix of generalized masses,

$$M = \begin{bmatrix} a_1 + a_2 + a_3 & a_1 - a_2 \\ a_1 - a_2 & a_1 + a_2 + a_3 \end{bmatrix} \tag{2}$$

C – matrix of coefficients of centrifugal forces and Coriolis forces,

$$C = \begin{bmatrix} 0 & 2a_4 \left(\dot{\alpha}_2 - \dot{\alpha}_1\right) \\ -2a_4 \left(\dot{\alpha}_2 - \dot{\alpha}_1\right) & 0 \end{bmatrix} \tag{3}$$

F – vector of friction wheels rolling forces,

$$F = \begin{bmatrix} a_5 sgn\dot{\alpha}_1 \\ a_6 sgn\dot{\alpha}_2 \end{bmatrix} \tag{4}$$

Q – vector of generalized forces,

$$Q = \begin{bmatrix} M_1 \\ M_2 \end{bmatrix} \tag{5}$$

Values of parameters $a_{(.)}$ were obtained in and identification process [2, 3, 4, 5].

3. Linear in the Parameter Neural Nets

Issues of modelling and control of nonlinear objects is complex. As a result of the lack of a systematic approach to analysis and synthesis of dynamic nonlinear systems so far, artificial neural networks, because of the ability of approximation of any nonlinear mapping and adaptation, became an attractive tool applied in nonlinear systems theory.

As everybody knows, neural networks have good static mapping properties. Applying of the neural networks to control in a real time may require recreating a full dynamics of controlled objects, which may result in a great dimensioning of dynamic networks. However, applying of linear neural networks, because of their weights, such as radial networks, B–splain networks, networks with functional expansions, prevent a problem of solutions explosion. Taking into consideration nonlinearity of a controlled object, a linear neural network with the first layer of weight generated randomly was applied to compensation its nonlinearity in this project.

Let's consider a neural network presented in Fig. 2.

The networks' input–output mapping from Fig. 2 has a form

$$y_i = \sum_{j=1}^{N} \left\{ w_{ij} S \left[\sum_{k=1}^{n} v_{jk} x_k + v_{vj} \right] + w_{wi} \right\} i = 1, 2..., r \tag{6}$$

Assuming input vector's element $x_0 \equiv 1$ and threshold values' vector $[v_{v1},\ v_{v2},\ ...,\ v_{vN}]^T$ as the first column of V^Tmatrix, it was written down:

$$y = W^T S \left(V^T x\right) \tag{7}$$

where $S = [S_1(.), S_2(.), ..., S_N(.)]^T$ is a vector of function describing neurons, which first element equals 1 and vector $[w_{w1}, w_{w2}, ...w_{wr}]^T$ makes the first column of W^T

matrix. From mathematic point of view, a two–layer network may approximate a continuous function of several variables. Any continuous function $f : D_f \subset R^n \to R^r$, where $D_f$is a compact subset R^n, may be approximated with any closeness by a two–layer neural network with appropriately selected weights.

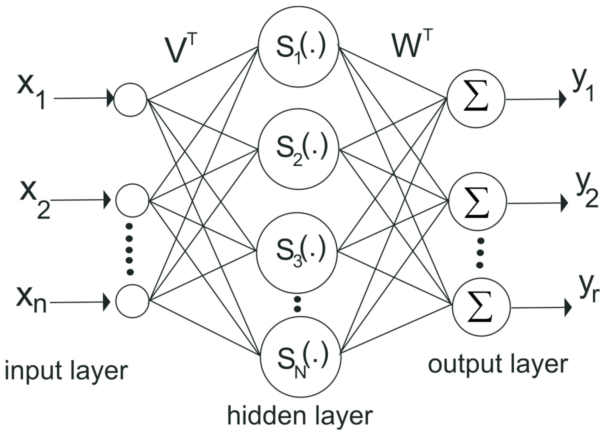

Figure 2 Structure of two–layer neural network

That means for a particular compactD_f set and positive value of an approximation error ε there is two–layer neural layer (Fig. 2) that a $f(x)$ function can be wrote down as [5]:

$$f(x) = W^T S\left(V^T x\right) + \varepsilon \tag{8}$$

for $||\varepsilon|| < \varepsilon_N$. If weights of the first layer of the V^Tnetwork are determined randomly [5], then W^Tweights of the second layer of the network define its characteristics and then it is a one-layer network. If we define $\phi(x) = S\left(V^T x\right)$, then we can write a redundancy (2) as:

$$y = W^T \phi(x) \tag{9}$$

where: $x \in R^n$, $y \in R^r$, $\phi(.) : R^n \to R^N$, and N is a number of neurons in a hidden layer. This network is a linear one because of W^T weights and has approximating properties of nonlinear functions. Sigmoidal bipolar functions, as a vector of basic networks functions, were assumed to approximation of nonlinearity. Then estimating of nonlinear$\mathbf{f}\left(\mathbf{x}\right)$ function is given by an equation:

$$\hat{\mathbf{f}}\left(\mathbf{x}\right) = \hat{\mathbf{W}}^T \mathbf{S}\left(\mathbf{V}^T \mathbf{x}\right), \tag{10}$$

where $\mathbf{V}$ is a constant weights matrix of an input layer generated randomly.

Functions of neurons activation are described by a redundancy:

$$\mathbf{S}\left(V^T \mathbf{x}\right) = \frac{2}{1 + \exp\left(-\beta \mathbf{V}^T \mathbf{x}\right)} - 1 \tag{11}$$

where β coefficient is responsible for slope of the function.

4. Neural control

Let's write dynamic equations of motion (1) of a mobile wheel robot in a form:

$$M\left(q,a\right)\ddot{q}+C\left(q,\dot{q},a\right)\dot{q}+F\left(\dot{q},a\right)=u \tag{12}$$

Let's define errors for a given trajectory of motion:

$$e=q_d(t)-q(t) \tag{13}$$

$$s=\dot{e}+\Lambda e \tag{14}$$

where Λ is a design matrix, diagonal with positive elements.

Differentiating (14) and taking into account (12), we will receive a description of a bounded system control in a generalized error function s in a form

$$M(q,a)\dot{s}=-C\left(q,\dot{q},a\right)s+f(x) \tag{15}$$

where a nonlinear $f(x)$ function is given by a redundancy

$$f(x)=M\left(q,a\right)\left(\ddot{q}_d+\Lambda\dot{e}\right)+C\left(q,\dot{q},a\right)\left(\dot{q}_d+\Lambda e\right)+F\left(\dot{q},a\right) \tag{16}$$

General form of a control signal with regards to a compensation of model nonlinearity is as follows [5]

$$u=\hat{f}+K_D s \tag{17}$$

where $\hat{f}$ is a function approximating f function, and

$$K_D s=K_D\dot{e}+K_D\Lambda e \tag{18}$$

Is an equation of PD regulator located in an external loop of a control system.

In a synthesis of an adaptive neural algorithm, a compensation of an object's nonlinearity was realized with a use of a neural network presented in a chapter 3.

Let's write$f(x)$ function as:

$$f(x)=W^T S(x)+\varepsilon \tag{19}$$

where ε is a bounded approximation error satisfying a redundancy $\|\varepsilon\|<\varepsilon_N$, $\varepsilon_N=const.$

Let's assume that ideal weight of the network is bounded, then an estimating of $f(x)$ function will be:

$$\hat{f}(x)=\hat{W}^T S(x) \tag{20}$$

where$\hat{W}$means an estimating of neural network weights determined in an adaptation algorithm.

Control law will have a form:

$$u=\hat{W}^T S(x)+K_D s \tag{21}$$

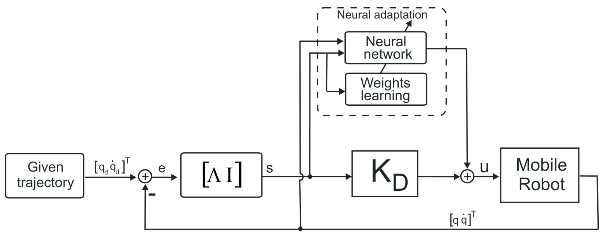

Figure 3 Neural control system structure

Determining estimated networks weights error:

$$\tilde{W} = W - \hat{W} \tag{22}$$

we will determine approximated function error:

$$\tilde{f}(x) = \tilde{W}^T S(x) + \varepsilon \tag{23}$$

Implementing the redundancy (23) to the equation (15), we will receive a description of a bounded control system:

$$M\left(q\right)\dot{s} + C\left(q, \dot{q}\right)s + K_D s = \tilde{W}^T S\left(x\right) + \varepsilon \tag{24}$$

The synthesis of neural control system of a mobile robot was carried out on the basis of Lapunow stability theory [5].

Let it be given a function:

$$L = \frac{1}{2}\left[s^T M(q)s + tr\left(\tilde{W}^T F^{-1}\tilde{W}\right)\right] \tag{25}$$

where:

$\tilde{W}$ – parameters estimation error,

F – diagonal matrix with positive elements.

Differential coefficient of L function will come to:

$$\dot{L} = s^T M(q)\dot{s} + \frac{1}{2}s^T \dot{M}(q)s + tr\left(\tilde{W}^T F^{-1}\dot{\tilde{W}}\right) \tag{26}$$

Taking a description of a bounded control system (24) to a redundancy (26), we will receive:

$$\begin{aligned} \dot{L} = -s^T K_D s + 0.5s^T\left(\dot{M}(q) - 2C(q, \dot{q})\right) s \\ + tr\left\{\tilde{W}^T\left(F^{-1}\dot{\tilde{W}} + S\left(x\right)s^T\right)\right\} + s^T\varepsilon \end{aligned} \tag{27}$$

Assuming an algorithm of network weights' learning in a form:

$$\dot{\hat{W}}(t) = FS(x)s^T \tag{28}$$

and knowing that a matrix:

$$\dot{M}(q) - 2C(q, \dot{q}) \tag{29}$$

is a skew-symmetric matrix, we will receive:

$$\dot{L} = -s^T K_D s + s^T \varepsilon \le -K_{D\,\min} \|s\|^2 + \varepsilon_N \|s\| \tag{30}$$

where $K_{D\,\min}$ is the least characteristic value of K_D matrix.

Because ε_N is a constant value, then $\dot{V} \le 0$ provided that a following condition is met:

$$\psi = \{s : \|s\| > \varepsilon_N / K_{D\,\min} \equiv b_s\} \tag{31}$$

From the redundancy $(4, 3, 13)$ it comes out that generalized following error s is uniformly finitely bounded to ψ set, with a practical limit assigned by b_S. Moreover, increasing in K_D coefficient of PD regulator causes decreasing in s following error, and in consequence also e and $\dot{e}$ errors, which are bounded on the basis of (31). Such a perspective of the neural control enables a correct work of a closed–loop control system as a result of activity of an internal loop with PD regulator up to a moment when a network starts to learn. That means a preliminary learning process of a network's weights is not required and a network's weights are estimated in a real time.

5. Verification of solutions

Verification of a proposed control algorithm was carried out in a control–measuring environment designed by the authors. In order to register signals from a real robot, taking advantage of the libraries provided by a producer, tolbox Matlab/Simulink software was built [1]. This solution enables to monitor work of a group of mobile Amigobot robots, which consists of registration of motion parameters, readings of a distance sensors, battery voltage measurement, and measurement of electric currents engines driving wheels and control of drives.

Look of a configuration window of sub–system made in Matlab/Simulink program shows Fig. 4.

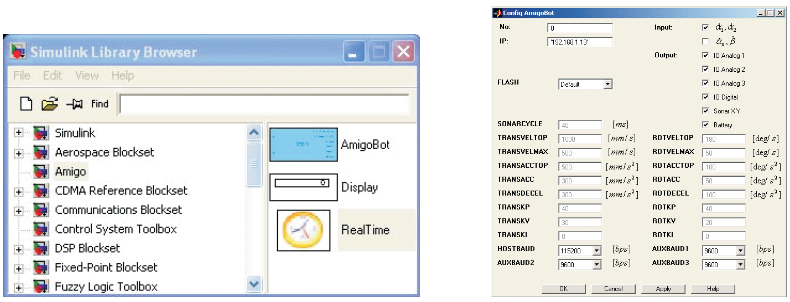

Figure 4 The configuration window of the Amigo sub–system

Registered random motion parameters and signals are shown in Fig. 5.

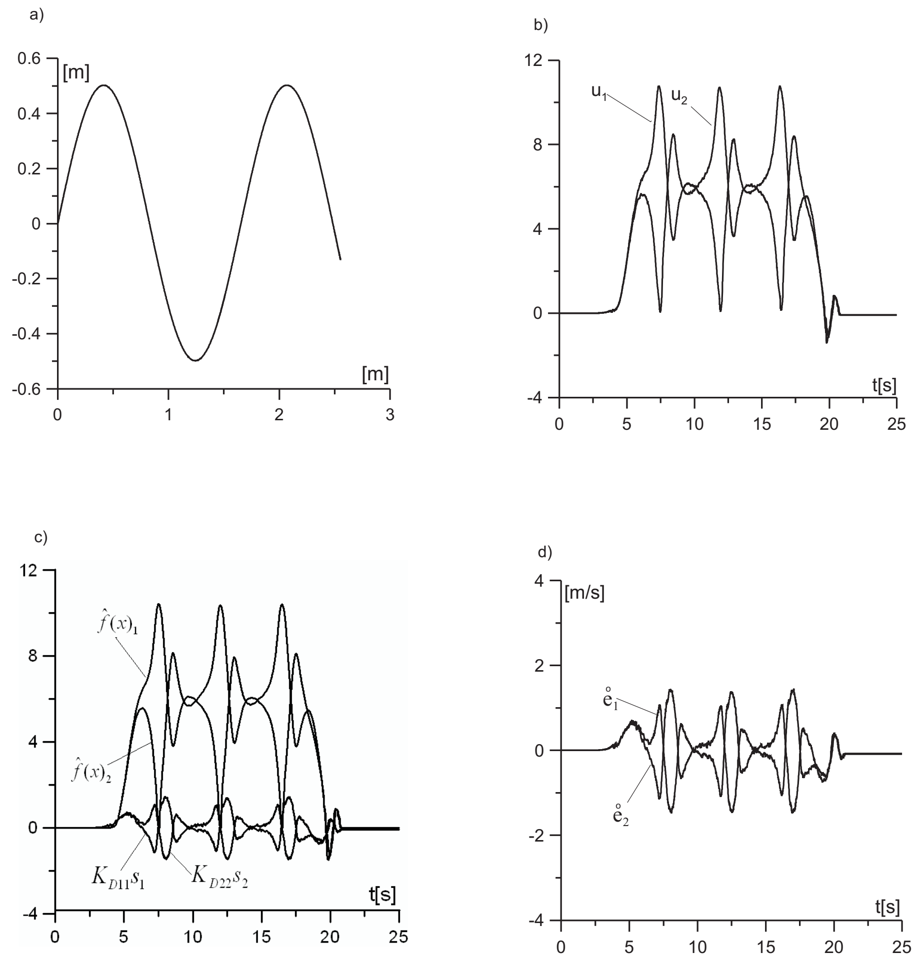

Figure 5 Random signals: a) motion track point A of a mobile robot b) control signals, c) adaptive control and PD control, d) errors of driving wheels realization speed

In Fig. 5b there is a changing of systems controlling the work of a laboratory AmigoBot robot presented, which A point moves on a track presented in Fig. 5a. The speed realization errors during realization of sinusoidal trajectory (Fig. 5a) is presented in Fig. 5d. Fig. 5c shows control signals tracks, which consist of nonlinearity compensating part and a control signal generated by PD regulator.

6. Conclusions

Proposed mobile robot movement control algorithm with a neural adaptation of model parameters enables a realization of complex motion trajectories. Verification results received in an author's control-measuring environment allows assuming that the solution is correct. The speed and displacement errors are concurrent to zero and estimated parameters are bounded.

Presented control algorithm may be implemented as a low level control system in hierarchical mobile robots control systems, providing realization of trajectory generated by superior control systems.

This paper is supported by Polish Government under Contract N N501 068838

References

[1] **Burghardt, A.**: Proposal for a rapid prototyping environment for algorithms intended for autonomous mobile robot control, *Mechanics and Mechanical Engineering*, Technical University of Łódź, vol. 12, No 1, p. 5–16, **2008**.

[2] **Burghardt A.**: Modelowanie dynamiki mobilnego robota kołowego równaniami Appella, *Acta Mechanica et Automatica*, vol.4, no.1 Białystok, **2010**.

[3] **Burghardt, A. and Giergiel, J.**: Modelling of Mobile Wheeled Robot with Dynamic Drive Compliance, *Modelling and Optimization of Physical Systems*, 9, Gliwice **2010**.

[4] **Burghardt A., Giergiel, J.**: Kinematics of a Robot Formation in Large–SizeTransportation, *Polish Journal of Environmental Studies*, Vol. 20, No. 5A, p.41–45, Hard Publishing Company, **2011**.

[5] **Giergiel, M., Hendzel, Z. and Żylski, W.**: Modelowanie i sterowanie mobilnych robotów kołowych, *PWN*, Warszawa, **2002**.