Control of a Differentially Driven Mobile Robot Using Radial Basis Function Based Neural Networks

GÖKHAN BAYAR, E. İLHAN KONUKSEVEN, A. BUĞRA KOKU
Mechanical Engineering Department
Middle East Technical University
Orta Doğu Teknik Üniversitesi,
Makina Mühendisliği Bölümü, 06531 Balgat
Ankara / TURKEY
bayar@metu.edu.tr http://www.me.metu.edu.tr/people/bayar

Abstract: - This paper proposes the use of radial basis function neural networks approach to the solution of a mobile robot orientation adjustment using reinforcement learning. In order to control the orientation of the mobile robot, a neural network control system has been constructed and implemented. Neural controller has been charged to enhance the control system by adding some degrees of award. Making use of the potential of neural networks to learn the relationships, the desired reference orientation and the error position of the mobile robot are used in training. The radial basis function based neural networks have been trained via reinforcement learning. The performance of the proposed controller and learning system has been evaluated by using a mobile robot that consists of a two driving wheels mounted on the same axis, and a free wheel on the front for balance.

Key-Words: - Mobile robot, control, neural networks, radial basis function, learning, trajectory

1 Introduction

Desired orientation adjustment and trajectory tracking problems for the mobile robots have been studied by the researchers over the last decade [1-3]. It is seen from the related research that extensive efforts have not been spent on the robustness of the control of the mobile robot orientation control [1-5]. Although most of the studies have been conducted on solving the problem of motion under non-holomonic constraints using the kinematic model, there is limited number of studies related to the problem of kinematic controller and the dynamics of the mobile robot [6, 7].

One of the problems in mobile robotics is the estimation of the robot position in its working environment. In order to solve this problem, some models have been developed. The idea behind these models is first to estimate a confidence interval of the robot position and then to compare it with the estimation obtained from robot's dead reckoning system. Neural network strategies have been applied in these models where network can adapt itself in real-time to changing conditions and learn [22]. Neural network control strategies have also been applied to the applications of autonomous navigation with obstacle avoidance, path planning and multiple targets tracking of mobile robots [23, 24, 25].

Another intensive use of neural networks takes place in applications consisting of intelligent and adaptive control [8, 9]. In order to control the nonlinear dynamics of the mobile robots, use of neural networks has been performed. These applications make use of the prominent advantage of neural networks; ability to learn and good performance for the approximation of nonlinear functions [10, 11, 12, 13].

Traditionally the learning capability of a multilayer neural network has been applied to the navigation problem in mobile robot applications. In these approaches the neural networks are trained in a preliminary off-line learning phase with navigation pattern behaviors [7, 15, 16].

Often, the control systems, which are used on the mobile robot applications, utilize multilayer neural networks. However, the robustness of the system to be controlled cannot be increased and high degrees of nonlinearities cannot be avoided by using the multilayer neural networks [14].

Ever so often, mobile robots are constructed with simple control schemes such as PID control. However, the capability for manual regulations of the parameters of the controller is to be deficient in order to compensate for disturbances acting upon the mobile robot such as deriving the mathematical model of the system with the ground interaction (i.e., including friction forces or observing differences of the inertial forces between one the modeled and the one obtained from the real case). After obtaining the proper controller parameters of

the system manually, the controller adapted to the system is expected to work well generally for small fluctuations. In the case of encountering a large fluctuation/disturbance, however, the controller parameters have to be continuously adjusted otherwise, the robot's performance will significantly be degraded. Since some parameters such as desired speed, terrain characteristics, motor voltage and current flow, tire inflation are going to change during operation, the parameters of the controller should be regulated continuously. Using fixed or manually changed controller parameters is at the bottom line not practical when robot dynamics is subject to continuous change. To minimize human intervention and to improve robot performance, the control system should be able to modify its parameters automatically under varying conditions.

In this paper, in the first section, some well known mobile robot platforms are going to be introduced briefly. Section is completed with a comparison chart indicating the types of neural networks structures and learning strategies adapted to these robots. For this work, radial basis function based neural networks approach has been chosen since multilayer neural networks can exhibit highly nonlinear behaviors. In order to increase the tracking robustness, a reinforcement learning strategy has been adopted. Radial basis function neural networks and the proposed learning strategy, reinforcement learning, have been adapted to the control of the specified mobile robot model. By using this adaptation, the control system is going to be introduced some degree of robustness.

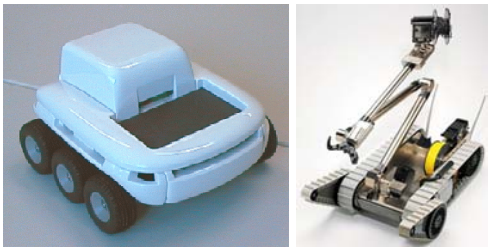
2 Overview of Mobile Robots

Mobile robots are used in a variety of environments for achieving different tasks. There is a differentiation between operational environments that is based on whether the robot operates indoors or outdoors. Indoors generally provide well structured environments with smooth surfaces to detect and to move on. On the other hand outdoor environments might appear in various conditions such as smooth, rough or mixed terrain. Some common indoor robotic tasks are given as; cleaning, patient assistance, surveillance, etc. The major difference between indoor and outdoor applications is that, outdoor, the outdoor environment is almost never controlled, motion planning and execution is more complex and the robot can frequently encounter unexpected situations.

(a) (b)

Fig. 1. Sojourner [31] and Pioneer [32]

The popular one among the mobile robots is Sojourner shown by Figure 1-a. It was constructed for the mission of exploring planet Mars. Pioneer mobile robot base was constructed for the exploration of Sarcophagus at Chernobyl (Figure 1-b). Mobile robot platform Koala was designed for easy use and easy transportation shown by Figure 2-a. Compact size, modularity and easy control are noticeable features of the robot Koala. Packbot is a tough, light weight mobile robot (Figure 2-b).



(a) (b)

Fig. 2. Koala and Packbot [33]

Packbot was constructed for searching explosive ordnance disposal, search and surveillance. The robot can also be aimed for the vital tasks like bomb squads, swat teams and so many military purposes. MR-5 shown in Figure 3 is a mobile robot platform developed for searching hazardous environments. It can operate in any weather and terrain condition.

Fig. 3. MR-5 [34]

A powerful mobile robot, Nomad, was constructed for Antarctic applications (Figure 4). It can operate autonomously without any external intervention.

Fig. 4. Nomad [35]

In the section overview of mobile robots, some of the famous mobile robot platforms have been given. These robots have been effectively used for special tasks for a while. All these robots have to follow a given reference trajectory. Therefore a control strategy for having a successful trajectory tracking must be used.

As seen from Table 1, specified mobile robot platforms have been controlled via neural networks control approach. In that table, types of neural networks which have been used for a specific application of the robot are given. Furthermore methods of neural networks approach are used due to their learning capability. In Table 1, the learning rules, which have been used to give the learning capability to the mobile robot, are also given. The abbreviations used in Table 1 are as follows: SL, ML, RBF and RC denote single layer (neural network) NN, multi layer NN, radial basis function NN and recurrent NN, respectively. RL, SL, USL, and QL denote reinforcement learning, supervised learning, unsupervised learning, and Q learning, respectively. Consequently it can be observed from Table 1 that radial basis function based neural networks have not been fully adapted to the specified mobile robots. Furthermore, reinforcement learning procedure and radial basis function neural networks have not been combined and employed for a purpose that is about tracking of a given desired orientation profile.

Table 1. Neural network control of the some of the commonly used mobile robot platforms

Mobile Robot	Neural Network Control				Learning Rule			
	SL	ML	RBF	RC	RL	SL	USL	QL
Sojourner	-	+	-	+	+	+	+	+
Pioneer	+	+	-	+	+	+	+	+
Koala	+	+	-	+	+	+	+	-
Packbot	+	+	+	-	-	+	-	-
MR-5	-	+	-	-	-	+	+	-
Nomad	+	+	-	+	+	+	+	+

3 Neural Network Structures and Learning Algorithms

A neural network is composed of numerous neurons that are connected in different ways. Figure 5 shows the atomic structure of a network, namely a neuron.

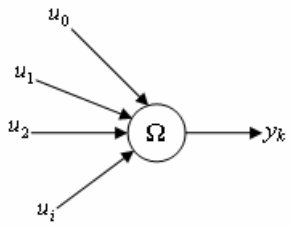


Fig. 5. Simplified structure of a network

There are some types of neural networks structures that can be used for determining and controlling the system. The mostly used ones are single-layer neural networks, multilayer networks, radial basis function based neural networks and recurrent neural networks. Feedforward neural networks can be built by single-layer networks which is the simplest form. Fundamentally this type of networks is composed of a single neuron (Figure 6). In order to regulate the weights of the networks for the system changes, the learning algorithm can be developed by the single-layer networks strategy [26, 27]. The learning algorithm for adapting the weights of the networks can be summarized as:

$$\omega(k+1) = \omega(k) + \eta e(k) u(k) \qquad (1)$$

where u is the input signal, e is the error, and η is the design parameters. Note that this algorithm supposes a global minimum due to errors that depends on the weights.

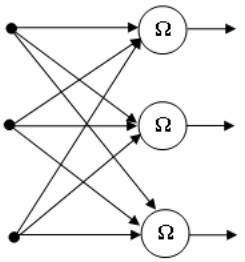


Fig. 6. Single-layer neural network

Single-layer and multi-layer neural network structures deviate from each others by presence of one hidden layer(s) as seen from Figure 7. In each layer all neurons have their inputs. At the same time these inputs are output signal of the behind layer. For the multilayer neural networks structure, networks may be partially-connected or fully-

connected by connections of every neuron in the adjacent forward layer. The special characteristics for a multilayer perception may be given as follows: a chance of a high degree of connectivity of the network, possibility of having one or more hidden layers and smoothness of the activation function of each neuron. Multilayer neural networks strategy has also ability for learning. Therefore, it may be a powerful networks type for some applications.

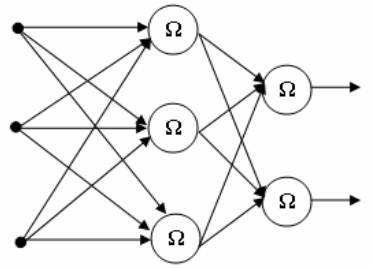


Fig. 7. Multilayer neural network

In order to encode the on time information coming from the controlled system by using static neural networks, delay inputs and outputs may be introduced to the networks. In such a way, there are some restrictions due to encoding a limited number of previous evaluated outputs and affecting inputs. Using such a strategy requires amount of time, memory and computation power. Hence recurrent neural networks emerged. As seen from Figure 8 recurrent neural network is apart from the static neural network in that it has a feedback. There is a chance using recurrent neural networks that a learning algorithm can be implemented to such a network.

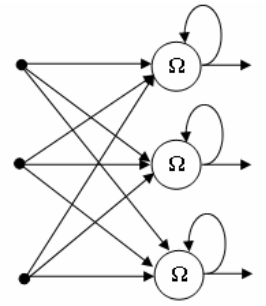


Fig. 8. Recurrent neural network

Radial basis function (RBF) was developed for solving the multivariate interpolation problem [27]. The RBF and neural networks application were first combined and reported in [28]. The basic principle of RBF neural networks is that it consists of multi-dimensional variables. They are related to the distance between the input vectors, u and a center, M_c, where the distance, d, can be calculated as:

$$d = \sqrt{(u-c)^T (u-c)} \tag{2}$$

The RBF neural networks has a feed-forward structure that consists of a single hidden layer locally tuned units. The units are fully interconnected to an output layer of linear units shown in Figure 9. The RBF neural networks have three different layers. The structure of the input layer is constructed from input neurons. Through the instrumentality of the RBF, these neurons are used for nonlinear transformation of the inputs. And the last layer should be the output layer.

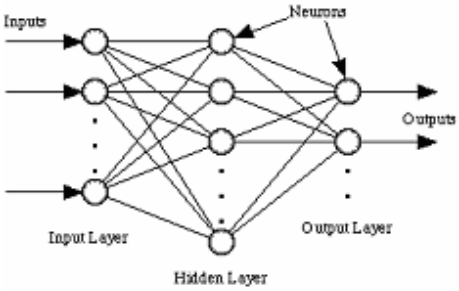


Fig. 9. Radial basis function neural network

Different types of functions may be used in RBFs, and commonly used ones can be listed as follows;

$$f(d) = d \ \text{(linear function)} \tag{3}$$

$$f(d) = d^3 \ \text{(cubic function)} \tag{4}$$

$$f(d) = e^{-\frac{d^2}{2\sigma^2}} \ \text{(Gaussian function)} \tag{5}$$

$$f(d) = \left(\frac{d^2}{\sigma^2}\right)^{\frac{1}{2}} \ \text{(quadratic function)} \tag{6}$$

A special but commonly used RBF network assumes a Gaussian basis function for the hidden units given in Equation (5). A general mathematical representation for the RBF neural networks given in Figure 9 can be given by:

$$y = F(u) = \sum_{i=1}^{n} \omega_i f\left(\left\|u - c_i\right\|^2_{r_i}\right) \tag{7}$$

where ω is the weights, *f(.)* is the RBF, u is the input, c is the center, r is the radius and n is the number of the neurons. The distance can be defined as:

$$\left\|u - c_i\right\|^2_{r_i} = \frac{(u-c_i)^T (u-c_i)}{r_i} \tag{8}$$

Another way to but the same equation can be given as follows:

$$\left\|u-c_i\right\|^2_{r_i} = (u-c_i)^T \forall_i^{-1} (u-c_i) \qquad (9)$$

where $\forall_i^{-1} = R_i^T R_i$ is constructed by defining of R as:

$$R_i = \begin{bmatrix} \frac{1}{\sqrt{r_i}} & 0 & 0 & . & . & 0 \\ 0 & \frac{1}{\sqrt{r_i}} & 0 & . & . & 0 \\ 0 & 0 & \frac{1}{\sqrt{r_i}} & 0 & . & 0 \\ . & . & . & . & . & . \\ 0 & . & . & . & 0 & \frac{1}{\sqrt{r_i}} \end{bmatrix}$$

The matrix $\forall_i^{-1}$ can be interpreted as a covariance one. If matrix R_i is given as following, then an elliptic radial base may be obtained.

$$R_i = \begin{bmatrix} \frac{1}{\sqrt{r_{1i}}} & 0 & 0 & . & . & 0 \\ 0 & \frac{1}{\sqrt{r_{2i}}} & 0 & . & . & 0 \\ 0 & 0 & \frac{1}{\sqrt{r_{3i}}} & 0 & . & 0 \\ . & . & . & . & . & . \\ 0 & . & . & . & 0 & \frac{1}{\sqrt{r_{ii}}} \end{bmatrix}$$

The Equation (3) can be given into a form which is one of the most popular RBF based on Gaussian functions:

$$f\left(\left\|u-c_i\right\|^2_{r_i}\right) = \exp(u-c_i)^T \forall_i^{-1} (u-c_i) \qquad (10)$$

The RBF neural networks of which the mathematical representation is given in Equation (10) has the power of approximating any continuous nonlinear function to a linear one [29]. Thus this property is a reason for preferring this strategy into robot control applications.

One of the most significant properties of a neural network is that it gives an opportunity to learn. Neural network can learn via performing an interaction with its environment and its information sources. In order to enhance the performance of a system that is controlled by a neural network, a learning rule is activated in that network. By means of adapting the weights of the networks according to a range of predefined performance, learning can be achieved. In other words, learning process is carried on the parameters of a weight space for obtaining a best solution and the process tries to optimize the predefined objective function. There are a few number of learning algorithm rules which are commonly used by the researchers. Some of them are supervised, unsupervised, competitive and reinforcement learning. In the supervised learning strategy, every input that comes from the working environment is forced to collaborate with a desired reference input. In the unsupervised learning procedure, it is the objective that the performance function defined based on the output of the network should be optimized. In the competitive learning strategy, the simple architecture is assumed. That means a single-layer is activated. Each of the single-layer units are encountered the same input and they produce outputs.

In order to construct a general learning procedure, the error can be defined for N training examples;

$$\varepsilon = \frac{1}{2}\sum_{k=1}^{N} e^2(k) \qquad (11)$$

with

$$e(k) = d(k) - \sum_{i=1}^{m} \omega_i f\left(\left\|u(k)-c_i\right\|^2_{r_i}\right) \qquad (12)$$

The quadratic error ε can be minimized with respect to ω_i, c_i and r_i. The algorithm process can be performed by analyzing these parameters (weight regulation, center regulation or radius regulation). In the process of network, weight is continuously changed, weight adaptation should be performed. In case of changing weights, using the gradient descent, weights can be regulated as given in Equation (13).

$$\omega_i(k+1) = \omega_i - \hbar_w \frac{\partial \varepsilon(k)}{\partial \omega_i(k)} \qquad (13)$$

While chancing weights of the networks, centers should be updated. The adaptation of the centers can be performed as:

$$c_i(k+1) = c_i(k) - \hbar_c \frac{\partial \varepsilon(k)}{\partial c_i(k)} \qquad (14)$$

The regulation of the radius can also be performed as given in Equation (15).

$$\forall_i^{-1}(k+1) = \forall_i^{-1}(k) - \hbar_r \frac{\partial \varepsilon(k)}{\partial \forall_i^{-1}(k)} \qquad (15)$$

In Equations (13-15), i terms are changing from 1 to n. $\hbar_w$, $\hbar_c$ and $\hbar_r$ are the design parameters and they can be related with the following range.

$$\hbar_r < \hbar_c < \hbar_w$$

Reinforcement learning is a process of trial and error designed to maximize the expected value of a criterion function known as a reinforcement signal [30]. The fundamental idea of the reinforcement learning is that if an action is followed by nearby the desired action, then the desire of the action is awarded. Otherwise, that of the action is weakened. In other words, action may be reinforced or unreinforced according to the planned events. Reinforcement learning requires that the weights of the neural networks should be updated for every response and produced an evaluated signal. By this way, for a desired reference, the probability of award is tried to maximize.

4 Modeling of Mobile Robot's Motion

The mobile robot considered here is shown in Figure 10. It has two driving wheels mounted on the same axis, and a front passive wheel for balance. The two driving wheels are controlled independently by DC motors. The dynamic property of the mobile robot and the kinematic relationships are given by the following equations [16, 20].

$$I\ddot{\phi} = f_r 2b - f_l 2b \qquad (16)$$

$$m\dot{v} = f_r + f_l \qquad (17)$$

$$I_t\ddot{\theta}_i + c_v\dot{\theta}_i = pu_i - rf_i \quad (i = r, l) \qquad (18)$$

$$r\dot{\theta}_r = v + 2b\dot{\phi} \qquad (19)$$

$$r\dot{\theta}_l = v - 2b\dot{\phi} \qquad (20)$$

where m is the mass of the mobile robot. f_l and f_r are the driving forces for the left and right wheels. $2b$ is the distance between left and right wheels. ϕ is the orientation of robot according to the absolute coordinate system OXY. I and I_t are the moment of inertia of the mobile robot and wheel, respectively. r is radius of the wheel. c_v and p are viscous friction and stiffness coefficients, respectively. u is driving input. θ is rotational angle of the wheel. v is velocity of the mobile robot.

Using the equation sets given above, one can define the state variables for the mobile robot as;

$$\bar{x} = \begin{bmatrix} v & \phi & \dot{\phi} \end{bmatrix}^T \qquad (21)$$

The manipulated input and output variables are constructed as;

$$\bar{u} = \begin{bmatrix} u_r & u_l \end{bmatrix}^T \quad \bar{y} = \begin{bmatrix} v & \phi \end{bmatrix}^T \qquad (22)$$

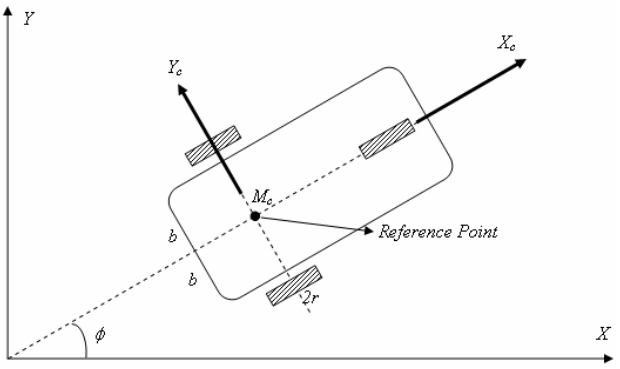


Fig.10. Model of the mobile robot.

State space representation of the system is given as;

$$\begin{pmatrix} \dot{v} \\ \dot{\phi} \\ \ddot{\phi} \end{pmatrix} = \begin{pmatrix} a_1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & a_2 \end{pmatrix} \begin{pmatrix} v \\ \phi \\ \dot{\phi} \end{pmatrix} + \begin{pmatrix} b_1 & b_1 \\ 0 & 0 \\ b & -b_2 \end{pmatrix} \begin{pmatrix} u_r \\ u_l \end{pmatrix} \qquad (23)$$

where

$$a_1 = -\frac{2c_v}{mr^2 + 2I_t} \qquad a_2 = -\frac{4c_v b^2}{Ir^2 + 2I_t(2b)^2}$$

$$b_1 = \frac{pr}{mr^2 + 2I_t} \qquad b_2 = -\frac{2prb}{Ir^2 + 4I_t b^2}$$

Equation (23) can be represented as;

$$\begin{aligned} \dot{x} &= Ax + Bu \\ Y &= Cx \end{aligned} \qquad (24)$$

The governing equation for the mobile robot's motion can be given as;

$$\dot{v} = a_1 v + b_1(u_r + u_l) \qquad (25)$$

$$\ddot{\phi} = a_2\dot{\phi} + b_2(u_r - u_l) \qquad (26)$$

$$\ddot{v} = a_1\dot{v} + b_1\left(\dot{u}_r + \dot{u}_l\right) \Rightarrow \dot{u}_r = \frac{\ddot{v} - a_1\dot{v}}{b_1} - \dot{u}_l \qquad (27)$$

5 Design of the Control System that Contains RBF-NN and Learning Mechanism

The proposed radial basis function neural networks control system is given in Figure 11.

The control system consists of the model, radial basis function neural networks controller, reference desired orientation generator, learning mechanism structure, controller parameters regulator and a reference model for generating the reinforcement learning signal.

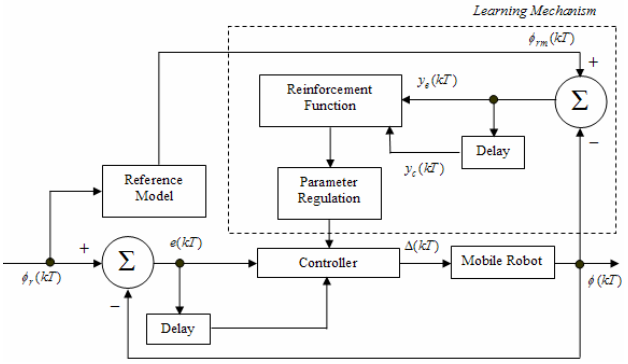


Fig. 11. Mobile robot motion control system [19].

5.1 Construction of Radial Basis Function Based Neural Networks

The proposed radial basis function based neural networks is shown in Figure 12. u_i is the inputs ($i=1,2,\ldots,n$) and $y=F(u)$ is the output as given in Equation (7). Here whole radial basis function neural networks in process are denoted by $F(u)$. The input to the receptive field unit is u and its output is shown by $\Gamma(u)$ and given in Equation (28). The receptive field unit has award, ξ_i

Let the receptive field unit be

$$\Gamma_i(u) = \exp\left(-\frac{\left|u - c_i\right|^2}{G_i^{\,2}}\right) \qquad (28)$$

where G_i is Gauss spread function.

The parameters of the neural networks should be regulated. These regulations are performed in order to obtain better response. The regulation conducted on the structure of the neural networks is to be related with modifying of the awards of the receptive field units continuously. It is assumed that the awards of the receptive field units have been regulated by the approach;

$$\xi_i(kT+1) = \xi_i(kT) + \lambda(kT+1)\Gamma_i(kT) \qquad (29)$$

where $\Gamma_i(kT)$ is the output of the i^{th} receptive field unit, $\lambda(kT)$ is the reinforcement signal.

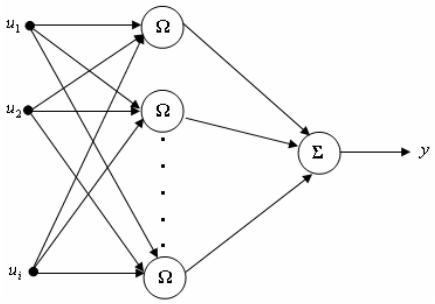


Fig. 12. Proposed radial basis function based neural networks model.

5.2 Learning Process

In order to obtain good response from the system that is controlled, reinforcement learning strategy is one of the preferred adaptive control approaches. In this study, a proper reinforcement learning procedure for the radial basis function neural networks has been developed. For this procedure, the basis of the mechanism is regulating the neural networks by reinforcement signal. That signal is generated via reinforcement function, λ. The procedure guarantees that the mechanism tries to learn. It can achieve this task by evaluating the generated plant input and the measured plant output (plant denotes the mathematical model of the mobile robot). The reinforcement learning structure checks whether the action fails or not.

After the action is performed, the response is evaluated by the target. In case of approaching the target, the award of the learning mechanism is weighted. If result is the direct contrary, the award of the signal is to be weakened. While the control system evaluates the relationship between the generated plant input and the measured plant output, it efforts to learn how the system behaves, how the system is detained into the range of desired field and how the action reaches the target.

The reinforcement function, which generates the reinforcement signal, fulfills its mission using the data collected from the system. In order to create a reinforcement function generally a reference model is used [19,21,22]. The reference model is constructed for organizing the performance of the system. The model tries to generate signal which

corresponds the output of the plant. Reference model can be selected by the designer as a first, second or higher order transfer function. In this work, reference model has been selected as a first order transfer function for obtaining a smooth response and it is given in Equation (30).

$$G(s) = \frac{b}{s+a} \tag{30}$$

Reference model can be given in discrete form as;

$$y_{rm}(kT) = \frac{1}{aT+2}\left(\begin{matrix} (2-aT)y_{rm}(kT-T)+... \\ \xi T\left(R(kT)+R(kT-T)\right) \end{matrix} \right) \tag{31}$$

y_{rm} and R denote input and output of the reference model in discrete form, respectively. After giving reference input, the difference between the response of the system and the response of the reference model produces the error. The error form is given as;

$$y_{error}(kT) = y_{rm}(kT) - y(kT) \tag{32}$$

In order to characterize the performance of the system, the first derivative has been used for obtaining the error between the reference model output and plant output.

$$y_p(kT) = \frac{y_{error}(kT) - y_{error}(kT-T)}{T} \tag{33}$$

Let choose the reinforcement function (related with Equation (29)) as [19,20,21];

$$\hat{\lambda}\left(y_{error}, y_p\right) = \hbar\left(-\hbar_{error} y_{error} - \hbar_p y_c\right) \tag{34}$$

where $\hbar$, $\hbar_{error}$, $\hbar_p$ are the design parameters;

6 Simulation Results

In order to check the reliability of the radial basis function neural networks controller and learning mechanism developed for the mobile robot orientation adjustment control, a set of demonstrations has been constructed. Desired orientation for the mobile robot has been tried to track by using the developed controller. The overall system has been designed and implemented using Matlab environment.

The physical parameters of the mobile robot are I=10kg.m^2, M=200kg, b=0.15m, I_t=0.005kg.m^2, c_v=0.05kg/s, p=5, r=0.1m. Moreover, the reference heading velocity, v, is 1m/s [17, 18].

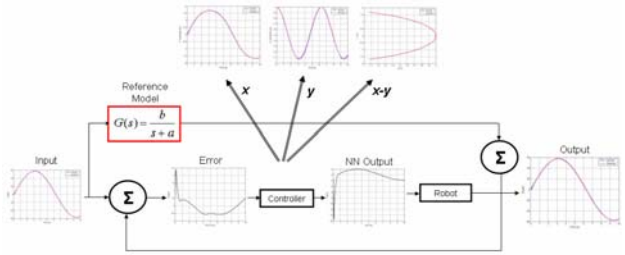


Fig.13. Schematic view of the control procedure

Schematic view of the control procedure of the mobile robot orientation is given in Figure 13. In the simulation studies, the results of position, orientation, and angular velocities have been simulated. Desired and the actual orientation obtained using the neural networks controller with the parameters of the reference function (a= b=1/50) is given in Figure 14. In Figure 15, the parameters are changed to a=b=1/15. In order to obtain the best fit parameters of the reference model, the absolute error, which is obtained between desired and actual orientation, has been scanned. By this way the most appropriate model parameters have been determined. To emphasize the importance of the model parameters on the overall control, two different models are presented in the simulation studies. Using two different set of parameters, the model has been exposed to the desired orientation profile. As seen from Figure 15, with the second parameter set, control system, which is continuously modified by neural network, tries to track the desired profile.

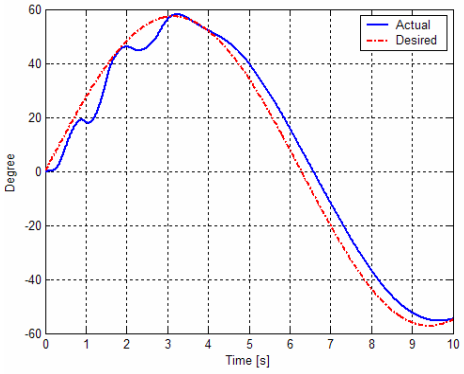


Fig. 14. Orientation of the mobile robot with the parameters of the reference function a=b=1/50.

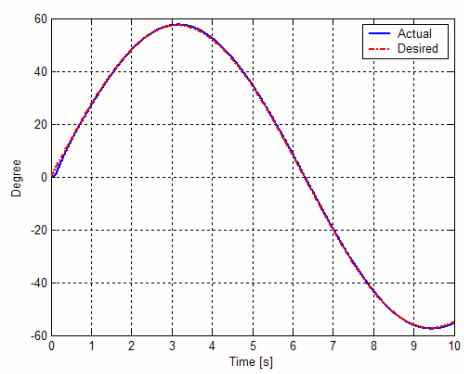


Fig. 15. Orientation of the mobile robot with the parameters of the reference function *a=b=1/15*.

In Figure 16, output of the proposed radial basis function based neural networks controller is given. Neural controller tries to keep the system at the desired orientation; meanwhile, the learning mechanism endeavors to adapt the controller to the coming portion of the desired trajectory.

As seen from Figure 17, orientation error carried out from the actual and desired orientation has been given. Control system enhances itself via improved signal generated by the learning system. It can be emphasized that the improvement in the control system can be recognized.

A neural control system has been constructed with employing the reinforcement learning mechanism. When reinforcement signal is very small, it is made to zero. Hence, reinforcement signal can not make a response to small fluctuations during regulating the neural controller. By this way the learning mechanism is only charged when the regulation is required.

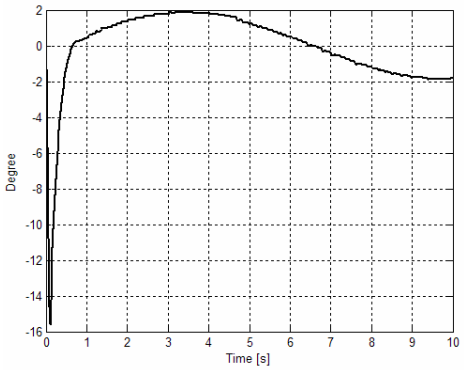


Fig. 16. Radial basis function neural networks controller output.

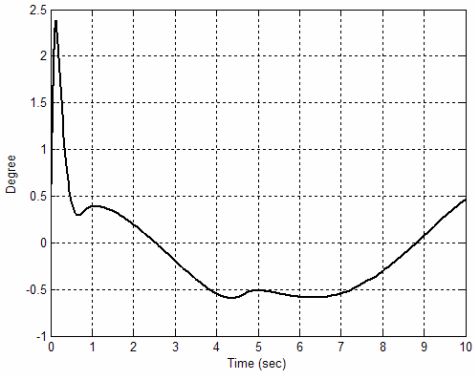


Fig. 17. Orientation error between actual and desired one.

The position of the mobile robot in the global frame *{XOY}* can be defined by the position of the mass center of the mobile robot system, denoted by M_c, which is the center of mobile robot, and the angle between robot local frame *{XcMcYc}* and global frame.

Kinematic equations of the mobile robot are;

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{pmatrix} = \begin{pmatrix} \cos\phi & 0 \\ \sin\phi & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v \\ \omega \end{pmatrix} \tag{35}$$

$$\begin{pmatrix} v \\ \omega \end{pmatrix} = \begin{pmatrix} r & 0 \\ \frac{r}{2b} & -\frac{r}{2b} \end{pmatrix} \begin{pmatrix} v_r \\ v_l \end{pmatrix} \tag{36}$$

where x and y are coordinates of the center of the mobile robot, v and w are linear and angular velocities of the robot, v_r and v_l are velocities of right and left wheels.

Position of the mobile robot in the global frame is given in Figure 18. Desired response obtained by the reference orientation trajectory, and the actual response obtained by the system controlled, have been get in quick succession. Time histories of *X* and *Y* coordinates are exhibited in Figure 19 and 20.

In Figure 21, time histories of *X* and *Y* coordinate errors are given.

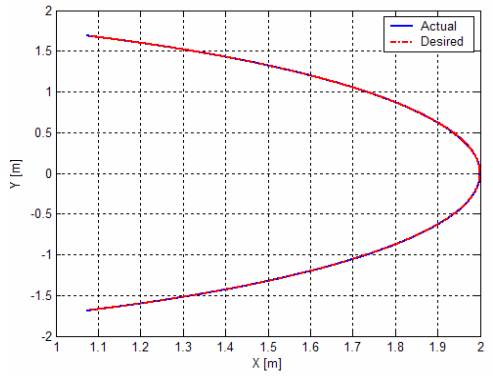


Fig. 18. Position of the mobile robot in the global frame.

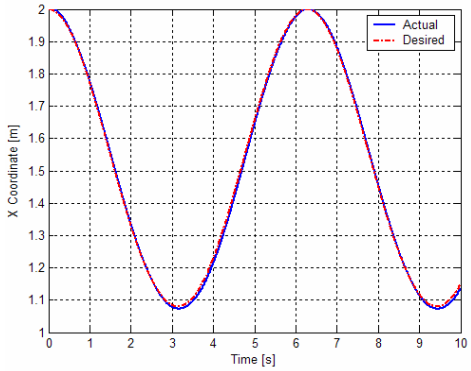


Fig. 19. Time history of X-Coordinate.

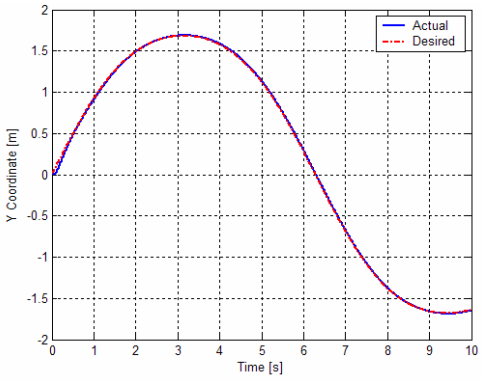


Fig. 20. Time history of *Y* Coordinate.

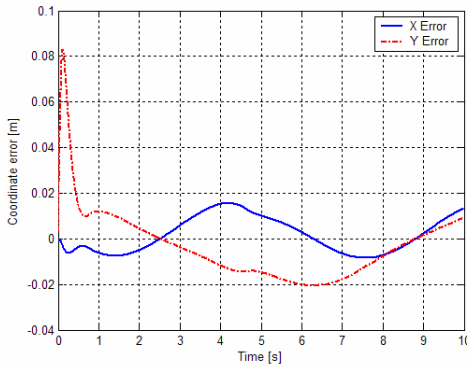


Fig. 21. *X* and *Y* coordinate errors.

7 Conclusion

In this paper, design of radial basis function neural networks controller and proper reinforcement learning mechanism for tracking of a desired orientation profile of the mobile robot have been presented. Neural controller has been charged to enhance the control system by adding some degrees of award. It has been achieved that neural networks system has learned the relationship between the desired directional orientation and the error position of the mobile robot. The radial basis function based neural networks have been trained via reinforcement learning. The performance of the proposed controller and the learning system have been investigated by using mobile robot that consists of a two driving wheels mounted on the same axis, and a front passive wheel for balance. For the modeling and simulation sections, both dynamic and kinematic models of the mobile robot have been utilized. The system has been exposed to a desired orientation trajectory profile. The proposed neural controller has enhanced itself with the cooperation of learning mechanism to track the given trajectory. Simulation results demonstrate the effectiveness of the proposed neural control system and the learning mechanism. It is planned that the procedure of which the details are given in this paper is going to be utilized on a real mobile robot. The simulation and the real-case results are going to be compared and presented for the researches who work on this subject.

References:

[1] M. L. Corradini, G. Ippoliti, S. Longhi and S. Michelini, Neural Networks Inverse Model Approach for the Tracking Problem of Mobile Robot, *Proc. of RAAD 2000, Maribor, Slovenia*, 2000, pp. 17-22.

[2] P. Jiang and H. Nijmeijer, Tracking Control of Mobile Robots: A Case Study in Backstepping, *Automatica*, Vol.33, 1997, pp. 1393-1399.

[3] C. Canudas de Wit, H. Khennouf, C. Samson and 0. J. Sordalen, Nonlinear Control Design for Mobile Robots, *Recent Trends in Mobile Robots*, World Scientific, Vol.11, pp. 121-156.

[4] J. M. Yang and J.H. Kim, Sliding Mode Motion Control of Nonholonomic Mobile Robots, *IEEE Control Systems*, Vol.19(2), 1999, pp. 15-23.

[5] R. Fierro and F. L. Lewis, Robust Practical Point Stabilization of a Nonholonomic Robot Using Neural Networks, *Journal of Intelligent and Robotic Systems*, Vol.20, 1997, pp. 295-317.

[6] C. Samson, Velocity and Torque Feedback Control of a Nonholonomic Cart, *Lecture Notes in Control and Information Science*, C. Canudas de Wit, Ed. Berlin, Germany: Springer-Verlag, 1991, pp. 125–151.

[7] R. Fierro, F. L. Lewis, Control of a Nonholomonic Mobile Robot Using Neural Networks, *IEEE Transactions on Neural Network*, Vol.9, No.4, 1998, pp. 589-600.

[8] A. G. Barto, Neuronlike Adaptive Elements that can Solve Difficult Learning Control Problems, *IEEE Transactions on Systems, Man and Cybernetics*, Vol.13, 1983, pp. 834-846.

[9] Z. Hendzel, Adaptive Critic Neural Networks for Motion Control of Wheeled Mobile Robot, *Nonlinear Dynamics*, Vol.50, No.4, 2007, pp. 849-855.

[10] K. S. Narenda, K. Pathasarathy, Identification and Control of Dynamic Systems Using Neural Network, *IEEE Transactions on Neural Networks*, Vol.1, No.1, 1990, pp. 4-27.

[11] D. Nguyen, B. Widrow, Neural Networks for Self-Learning Control Systems, *IEEE Control Systems Magazine*, Vol.10, No.1, 1990, pp. 18-23.

[12] K. Hornik, M. Stinchombe, H. White, Universal Approximation of an Unknown Mapping and Its Derivatives Using Multilayer Feedforward Networks, *Neural Networks*, Vol.3, 1990.

[13] J. Velagic, N. Osmic, B. Lacevic, Neural Network Controller for Mobile Robot Motion Control, *Proceedings of World Academy of Science*, Eng. And Tech. Vol.30, 2008, pp. 810-815.

[14] A. D'Amico, G. Ippoliti, S. Longhi, A Radial Basis Function Networks Approach for the Tracking Problem of Mobile Robots, *IEEE/ASME Int. Conference on Advanced Intelligent Mechatronics Proceedings*, 2001, pp. 498-503.

[15] K. Berns, R. Dillman, and R. Hofstetter, An application of a backpropagation network for the control of a tracking behavior, *in Proc. IEEE Int. Conf. Robot. Automat.*, Vol.3, Sacramento, CA, 1991, pp. 2426–2431.

[16] S. Nagata, M. Sekiguchi, and K. Asakawa, Mobile robot control by a structured hierarchical neural network, *IEEE Contr. Syst.*, 1990, pp. 69-76.

[17] K. Watanabe, J. Tang, M. Nakamura, S. Koga and T. Fukuda, Mobile Robot Control Using Fuzzy-Gaussian Neural Networks, *Proc. of IEEE/RSJ International Conf. on Robots and System*, 1993, pp. 919-925.

[18] J. Peng, Y. Wang, W. Sun, Trajectory Tracking Control for Mobile Robot Using Recurrent Fuzzy Cerebellar Model Articulation Controller, *Neural Information Processing*, Vol.11, No.1, 2007, pp. 15-23.

[19] K. M. Passino, Biomimicry for Optimization, Control and Automation, *Springer-Verlag* London, 2005.

[20] D. M. Skapura, Building Neural Networks, *ACM Press*, New York, USA, 1996.

[21] M. H. Hassoun, Fundamentals of Artificial Neural Networks, *MIT Press*, Londan, England, 1995.

[22] A Sanchez, M. A. Sanz Bobi, Fast Position Estimation for Autonomous Robot Navigation, *WSEAS Int. Conf. on Signal Processing, Robotics and Automation (ISPRA 02)*, Spain, 2002.

[23] V. Tiponut, A. Gacsadi, C. Caleanu, I. Gavrilut, Neural Network Guided Robot Collectivity-An Experimental Setup, *WSEAS Int. Conf. on Neural Networks*, Croatia, 2006.

[24] S. Pennacchio, F. Abissi, S. Petralia, G. Stendardo, New Intelligent Controller for Mobile Robot Navigation in Unknown Environments, *WSEAS Int. Conf. on Fuzzy Systems*, Portugal, 2005.

[25] N. Rahman, A. R. Jafri, M. U. Keerio, Fuzzy Behaviour Based Navigation of a Mobile Robot for Tracking Multiple Targets in an Unstructured Environment, *WSEAS Int. Conf. on Robotics, Control and Manufacturing Tech.*, China, 2006.

[26] F.Rosenblatt, The Perceptron: a Probalilistic Model for Information Storage and Organization in the Brain, *Psychological Review*, vol. 65, pp. 386-408, 1958.

[27] A. S. Poznyak, E. N. Sanchez, W. Yu, Differential Neural Networks for Robust *Nonlinear Control, World Scientific Pub.*, 2001.

[28] D. S. Broomhead, D. Lowe, Multivariable Functional Approximation and Adaptive Networks, *Complex Systems*, vol. 2, pp. 321-335, 1988.

[29] J. Park, I. W. Sandberg, Universal Approximation Using Radial Basis Function Networks, *Neural Computation*, vol. 3, pp. 246-257, 1991.

[30] M. H. Hassoun, Fundamentals of Artificial Neural Networks, Cambridge, *MA: MIT Press*, 1995.

[31] J. Matijevic, Characterization of the Martian Surface Deposits by the Mars Pathfinder Rover,

Sojourner, *Science*, no. 278, pp. 1765-1768, 1997.

[32] R. V. Martin, J. Martinez, et al., An Active Perception Control Architecture for Autonomous Robots, *Proc. Of the 7th WSEAS Int. Conf. on Automation and Inf.*, pp. 144-149, Croatia, 2006.

[33] www.irobot.com/filelibrary/pdfs/gi/accessories/iRobot_Accessories.pdf, Last Accessed Date, November, 2008.

[34] V. Kordic, A. Lazinica, M. Merdan, The Design of a Pair of Identical Mobile Robots to Investigate Co-Operative Behaviours, *Cutting Edge Robotics*, ARS/pIV, Germany, 2005.

[35] http://www.frc.ri.cmu.edu/projects/meteorobot/Nomad/Nomad.html, Last Accessed Date, November, 2008.