American Journal of Engineering Research (AJER)
e-ISSN: 2320-0847 p-ISSN : 2320-0936
Volume-5, Issue-11, pp-247-256
www.ajer.org
Research Paper Open Access

Comparative analysis for NN inverse model controller and back-stepping controller on mobile robots

Margarita Gjonaj[1], Betim Çiço[2], Arnisa Myrtellari[3]
[1](Automation Department, Polytechnic University of Tirana, Albania)
[2](Faculty of Contemporary Sciences and Technologies/South East European University, Macedonia)
[3](Automation Department, Polytechnic University of Tirana, Albania)

ABSTRACT: *This work addresses the design and implementation of a neural network based controller for the trajectory tracking of a differential drive mobile robot. A neural network based tracking control algorithm is proposed and simulation and experimental results are presented. The algorithm is a control structure that makes possible the integration of a back-stepping controller and a neural network (NN) computed-torque controller for a nonholonomic mobile robot. Integration of a neural network controller and the kinematic based controller gives the advantage of dealing with unmodeled and unstructured uncertainties and disturbances to the system. Comprehensive system modeling including robot kinematics, dynamics and actuator modeling has been done. The dynamic modeling is done Lagrangian methodologies for nonholonomic systems. Simulation of the robot model and different controllers has been done using Matlab and Matlab Simulink.*

Keywords: *Back-stepping Controller, Mobile Robot, Neural Network*

I. INTRODUCTION

Basically, robots can be divided into two categories, fixed and mobile robots. Fixed robots are mounted on a fixed surface and materials are brought to the workspace near the robot. A fixed robot is normally used in mass production, as in car factories, for welding or stamping. Mobile robots have the capability to move around in their environment and are not fixed to one physical location; therefore, the mobile robot can be defined as a mechanical device that performs automated tasks, whether according to direct human supervision, a pre-defined program, or a set of general guidelines, using artificial intelligence (AI) techniques [2]. Mobility is the robot's capability to move from one place to another in unstructured environments to a desired target. Mobile robots can be categorized into wheeled, tracked or legged robots, and they are more useful than fixed robots. Mobile robots are increasingly used in industry, in service robotics, for factories (e.g. in delivering components between assembly stations) and in difficult to access or dangerous areas such as space, military environments, nuclear-waste cleaning and for personal use in the forms of domestic vacuum cleaners and lawn mowers [2, 3, 4].

Over the last decade, the design and engineering of mobile robot systems acting autonomously in complex, dynamic and uncertain environments has remained a challenge. Such systems have to be able to perform multiple tasks, and therefore must integrate a variety of knowledge-intensive information processes on different levels of abstractions guaranteeing real-time execution, robustness, adaptability and scalability [6].

The trajectory tracking control for mobile robots is a fundamental problem, which has been investigated exhaustively by the scientific community. Several papers deal about the design of control laws for mobile robot with its dynamic model, for instance in trajectory tracking (De la Cruz et al., 2006; Dong et al., 2005; Albagul et al., 2004; Yang et al., 1999; Zhang et al., 1998). One of first investigation results for this problem was deal in Kanayama et al. (1990), where author uses the Lyapunov theory to design the tracking controller. Nevertheless, this and others controllers do not take in count the restrictions in the control signals because it is a hard task to implement[5, 10].

The work includes the development and construction of neural network based trajectory tracking controllers. The controller is a structure that makes possible the integration of a kinematic-based controller and a neural network computed torque controller for a nonholonomic differential drive mobile robot. The combined kinematic and torque control method is developed using backstepping and the system stability is proved using Lyapunov theory. The controller will be applied to the trajectory tracking problem and the results of its implementation on a mobile robot platform will be presented. [6,8,9,10,11,12,13]

II. FORWARD KINEMATIC MODEL OF MOBILE ROBOT

The goal of the robot kinematic modeling is to find the robot speed in the inertial frame as a function of the wheels speeds and the geometric parameters of the robot (configuration coordinates). In other words we want to establish the robot speed $\dot{q} = [\dot{x}\dot{y}\dot{\theta}]^T$ as a function of the wheel speeds $\dot{\varphi}_R$ and $\dot{\varphi}_L$ and the robot geometric parameters or we want to find the relationship between control parameters ($\dot{\varphi}_R$ and $\dot{\varphi}_L$) and the behavior of the system in the state space. The robot kinematics generally has two main analyses, one Forward kinematics and one Inverse kinematics:[9]

Forward kinematics:

$$\dot{q} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = f(\dot{\varphi}_R, \dot{\varphi}_L, gemometric\ parameters) \tag{1}$$

Inverse kinematics:

$$\begin{bmatrix} \dot{\varphi}_R \\ \dot{\varphi}_L \end{bmatrix} = f(\dot{x}, \dot{y}, \dot{\theta}) \tag{2}$$

Assume a differential drive mobile robot setup which has two wheels with the radius of R_a placed with a distance L from the robot center as shown in Fig. 1:The following notations will be used in this work:A: The intersection of the axis of symmetry with the driving wheels axis. C: The center of mass of the platform. a: The distance between the center of mass and driving wheels axis in x-direction. L: The distance between each driving wheel and the robot axis of symmetry in y-direction. R_a: The radius of each driving wheel. $\dot{\varphi}_R$: The rotational velocity of the right wheel. $\dot{\varphi}_L$: The rotational velocity of the left wheel. v: The translational velocity of the platform in the local frame. ω: The rotational velocity of the platform in the local and global frames.

The forward kinematic problem can be described as the problem of finding the following function:

$$\dot{q} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = f(\dot{\varphi}_R, \dot{\varphi}_L, L, R_a, \theta) \tag{3}$$

The speed of each wheel in the robot frame is $R_a\dot{\varphi}$, therefore the translational speed in the robot frame is the average velocity:

$$v = R_a \frac{\dot{\varphi}_R + \dot{\varphi}_L}{2} \tag{4}$$

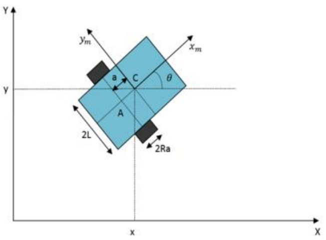


Figure 1: The differential drive mobile robot model

And the rotational velocity is:

$$\omega = \frac{R_a}{2L}(\dot{\varphi}_R - \dot{\varphi}_L) \tag{5}$$

The robot position in the inertial and robot frame can be defined as follows:

$$q_I = [x_I \quad y_I \quad \theta_I]^T \tag{6}$$

$$q_R = [x_R \quad y_R \quad \theta_R]^T \tag{7}$$

The mapping between these two frames is through the standard orthogonal rotation transformation:

$$\dot{q}_R = R(\theta)\dot{q}_I \tag{8}$$

$$R(\theta) = \begin{bmatrix} cos\theta & sin\theta & 0 \\ -sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{9}$$

Therefore the robot velocity in the global or inertial frame is:

$$\dot{q}_I = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \frac{R_a}{2} \begin{bmatrix} \dot{\varphi}_R + \dot{\varphi}_L \\ 0 \\ \frac{\dot{\varphi}_R - \dot{\varphi}_L}{L} \end{bmatrix} = \begin{bmatrix} R_a \frac{\dot{\varphi}_R + \dot{\varphi}_L}{2} \cos(\theta) \\ R_a \frac{\dot{\varphi}_R + \dot{\varphi}_L}{2} \sin(\theta) \\ \frac{R_a}{2L}(\dot{\varphi}_R - \dot{\varphi}_L) \end{bmatrix} \tag{10}$$

The above equation is the general forward kinematic equation for a differential drive mobile robot.

III. DYNAMIC MODELING OF THE MOBILE ROBOT

In order to produce motion, forces must be applied to the mobile robot model. These forces are modelled by studying of the motion of the dynamic model of the differential wheeled mobile robot shown in Fig.1. It deals with mass, forces and speed associated with this motion. The dynamics model can be described by the following dynamic equations based on Euler Lagrange formulation [23, 24, 25, 26]

$$M(q)\ddot{q} + V(q,\dot{q}) + F(\dot{q}) + G(q) + \tau_d = B(q)\tau - A^T(q)\lambda \tag{11}$$

Where: $M(q)$ is the symmetric positive definite inertia matrix, $V(q,\dot{q})$ is the centripetal and coriolis matrix, $F(\dot{q})$ is the surface friction matrix, $G(q)$ is the gravitational vector, τ_d Denoted bounded unknown disturbances including unstructured unmodeled dynamics, $B(q)$ is the input transformation matrix, τ is the input vector, $A^T(q)$ is the matrix associated with the constraints, λ is the vector of the constraint forces.

By taking a look at the forward kinematic equation, we write S (q) matrix as the modified forward kinematic matrix which has two velocity terms related to the distance between the robot centroid and wheel axis. Therefore, we can write the following equation for the system

$$\dot{q} = \begin{bmatrix} \dot{x}_c \\ \dot{y}_c \\ \dot{\theta} \end{bmatrix} = S(q)v_a(t) = \begin{bmatrix} cos\theta & -asin\theta \\ sin\theta & acos\theta \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{12}$$

It can easily be proved that the S (q) matrix has the following relation with A(q) matrix:

$$S^T(q)A^T(q) = 0 \tag{13}$$

The above equation is useful when we want to eliminate the constraint term from the main dynamic equation as you will see in the next step. Differentiating equation (12), we have:

$$M(q)\left[\dot{S}(q)v_a(t) + S(q)\dot{v_a}(t)\right] + V_m(q,\dot{q})S(q)v_a(t) + F(\dot{q}) + G(q) + \tau_d = B(q)\tau - A^T(q)\lambda \tag{14}$$

$$\ddot{q} = \dot{S}(q)v_a(t) + S(q)\dot{v_a}(t) \tag{15}$$

The next step to eliminate the constraint matrix $A^T(q)\lambda$ is to multiply equation (14) by $S^T(q)$ as follows ,as it can be seen from the above equation, we have $S^T(q)\ A^T(q)$which is zero according to equation (13). Therefore the constraint term is eliminated and the new dynamic equation is:

$$\bar{M}(q)\dot{v_a}(t) + \bar{V}_m(q,\dot{q})v_a(t) + \bar{F}(\dot{q}) + \bar{G}(q) + \bar{\tau}_d = \bar{B}(q)\tau \tag{16}$$

IV. KINEMATIC MODEL BASED BACKSTEPPING CONTROLLER

The kinematic based back-stepping controller for a nonholonomic mobile robot is first proposed in 1992 by Kanayama [10] and is been used by so many other researchers in this field. A stable tracking control rule for a nonholonomic mobile robot which neglects the vehicle dynamics and is based on the steering system is described [10]. In this control system two postures for the robot are going to be used: the reference posture $q_r = [x_r \quad y_r \quad \theta_r]^T$ and a current posture $q_c = [x_c \quad y_c \quad \theta_c]^T$. The reference posture is the goal posture and the current posture is the real posture at the moment. The block diagram of this control structure is shown in Fig.2:

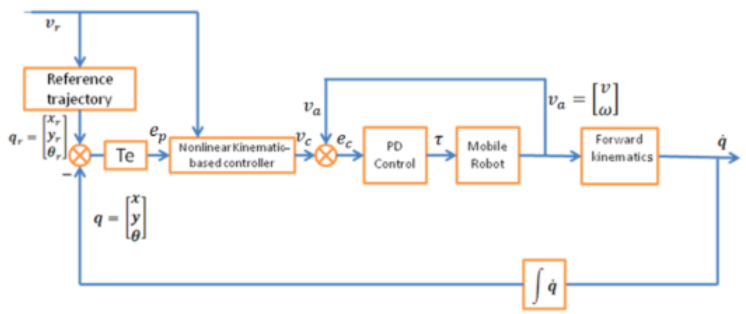


Figure.2: The kinematic based back-stepping controller

We define an error posture or the tracking error e_p in the basis of the frame linked to the mobile platform or the local frame as follows:

$$e_p = \begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix} = T_e(q_r - q) \tag{17}$$

$$\begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix} = \begin{bmatrix} cos\theta & sin\theta & 0 \\ -sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r - x \\ y_r - y \\ \theta_r - \theta \end{bmatrix} \tag{18}$$

The control problem in this case will be to calculate a control rule for the vehicle, which calculated the target velocities $v_c = f(e_p, v_r, K)$ that makes the system asymptotically stable. The proposed kinematic based control rule is as follows:

$$v_c = \begin{bmatrix} v_r cos e_\theta + K_x e_x \\ \omega_r + K_y v_r e_y + K_\theta v_r sin e_\theta \end{bmatrix} \tag{19}$$

$$v_c = f(e_p, v_r, K)$$

$$K = (K_x, K_y, K_\theta)$$

Where K_x, K_y and K_θ are positive constants. The PD controller in the block diagram of Fig. 2 is a linear controller which is responsible to convert the velocity output of the controller to torque inputs for the robot. The stability of the above control rule will be proved using the Lyapunov stability method in the next section.

LYAPUNOV STABILITY ANALYSIS

The Lyapunov stability analysis of the control rule in equation (19) is described as follows: [6,8,16]

Lemma 1:

According to equation (18) we have:

$$\begin{bmatrix} \dot{e}_x \\ \dot{e}_y \\ \dot{e}_\theta \end{bmatrix} = \dot{e}_p = f(t, e_p) = \begin{bmatrix} \omega(e_p, q_r)e_y - v(e_p, q_r) + v_r cos e_\theta \\ -\omega(e_p, q_r)e_x + v_r sin e_\theta \\ \omega_r - \omega(e_p, q_r) \end{bmatrix} \tag{20}$$

Using the robot kinematic equation (10) and the tracking error equation (18) we obtain the lemma.Clearly $V \geq 0$ and $V = 0$ if $e_p = 0$, therefore the above V function is a positive definite function.Furthermore, by using lemma 2 we have the derivative of the proposed Lyapunov function V is a negative definite function which demonstrates that the point $e_p = 0$ is uniformly asymptotically stable under the conditions that v_r and ω_r are continuous and v_r, ω_r, K_x, K_y and K_θ are bounded. The above Lyapunov stability analysis is used in reference [8] to prove the stability of the proposed controller.We demonstrated that the system is stable for any combination of K_x, K_y andK_θ. However, since we need a non-oscillatory, but not too slow response of the robot, we have to find an optimal parameters set for this controller. At this stage, the proper gains for each reference trajectory have to found by tuning.

V. THE BACKSTEPPING CONTROLLER DESIGN USING THE NONLINEAR FEEDBACK METHOD

The previous section was about selecting a velocity control v(t) defined in equation (19) for the steering system or the kinematic model of the robot. In this section we desire to convert such a prescribed control v(t) into a torque control τ(t) for the actual physical platform[15,17]. As it is mentioned in the modeling chapter, the complete equations of motion of the nonholonomic mobile platform are given by equations (12) and (16):

$$\dot{q} = \begin{bmatrix} \dot{x}_c \\ \dot{y}_c \\ \dot{\theta} \end{bmatrix} = S(q)v(t) = \begin{bmatrix} cos\theta & -asin\theta \\ sin\theta & acos\theta \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{21}$$

$$\bar{M}(q)\dot{v}(t) + \bar{V}_m(q, \dot{q})v(t) + \bar{F}(\dot{q}) + \bar{G}(q) + \bar{\tau}_d = \bar{B}(q)\tau \tag{22}$$

The complete dynamics consists of kinematic steering system (21) plus some extra dynamics (22). Standard approaches to nonholonomic control such as the kinematic based controller deal only with (21) ignoring the actual vehicle dynamics [20,21]. The application of this method to the case of the nonholonomic mobile robot is that by applying an appropriate nonlinear feedback we can linearize the nonlinear dynamics and then apply the kinematic based controller to the linear system. Let u be an auxiliary input, then by applying the nonlinear feedback:

$$\tau = \bar{B}^{-1}(q)[\bar{M}(q)u + \bar{V}_m(q, \dot{q})v(t) + \bar{F}(\dot{q})] \tag{23}$$

We can convert the dynamic equation (22) to:

$$\bar{M}(q)\dot{v}(t) + \bar{V}_m(q, \dot{q})v(t) + \bar{F}(\dot{q}) + \bar{G}(q) + \bar{\tau}_d = \bar{B}(q)\left[\bar{B}^{-1}(q)[\bar{M}(q)u + \bar{V}_m(q, \dot{q})v(t) + \bar{F}(\dot{q})]\right] \tag{24}$$

$$\dot{v}(t) = u \tag{25}$$

Therefore the complete equations of motion of the system will be:

$$\dot{q} = S(q)v(t) \tag{26}$$

$$\dot{v}(t) = u \tag{27}$$

In performing the nonlinear feedback (23) it is assumed that all the dynamical quantities such as $\bar{M}(q)$, $\bar{V}_m(q, \dot{q})$ and $\bar{F}(\dot{q})$ are exactly known and $\bar{\tau}_d = 0$..This is the case of the system (26,27) where the equilibrium point cannot be made asymptotically stable by any smooth time-invariant state feedback controller. A back-stepping controller is used to stabilize system (26,27). Consider a system given by

$\dot{x} = f(x) + g(x)\xi$ (28)

$\dot{\xi} = u$ (29)

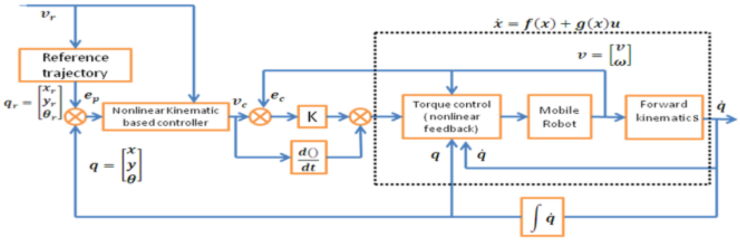
With f(0)=0 and f and g are smooth functions. Designing a feedback control to stabilize the system at (x=0, = 0) is called a back-stepping controller. Comparing equations (28,29) and (26,27) one can find out that the linearized nonholonomic mobile robot equations are in the form of (28,29) and a back-stepping control algorithm will make it stable. The proposed control input to stabilize system (26,27) is as follows:

$u = \dot{v}_c + K_4(v_c - v)$ (30)

$K_4$is a positive definite, diagonal matrix given by:

$K_4 = k_4 I$ (31)

The stability of the above control algorithm (30) was done using the Lyapunov stability analysis.
The general back-stepping control structure is shown in the block diagram of Fig.3 :
Figure 3: the backstepping controller with the nonlinear feedback structure



VI. NEURAL NETWORK STRUCTURE

Artificial NN are modeled on biological process for information processing, including specifically the nervous system and its basic unit, the neuron. Signals are propagated in the form of potential differences between the inside and outside of cells. The mathematical model of a neuron is depicted in Fig. 4:

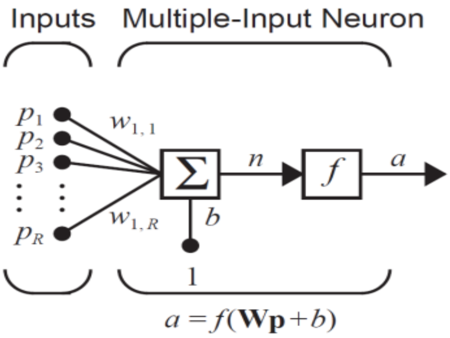


Figure 4: Mathematical model of a neuron

The input weightsw_j, the firing threshold b (also called the bias), the summation of the weighted inputs and the nonlinear activation function f are shown in the above figure[10,11,12,13,14]. If the cell inputs are n signals at the time instant k, $x_1(k), x_2(k), x_3(k), \ldots x_n(k)$ and the output is the scalar y (k), the mathematical equation of the neuron can be written as follows:

$y(k) = f(\sum_{j=1}^{n} w_j x_j(k) + b)$ (32)

The basic universal approximation theory says that any smooth function f(x) can be approximated arbitrarily closely on a compact set using a two layer NN with appropriate weights. Specifically, let f(x) be a general smooth function. Then given a compact set S and a positive numberϵ_N, there exists a two layer NN such that:

$f(x) = W^T \sigma(V^T x) + \epsilon$ (33)

The value ϵ is called the NN function approximation error and it decreases as the number of hidden layer neuronsL increase. On the other hand, on the compact set S, as S becomes larger, the required L generally increases correspondingly. The neural network acting as a function approximate is shown in Fig. 5:

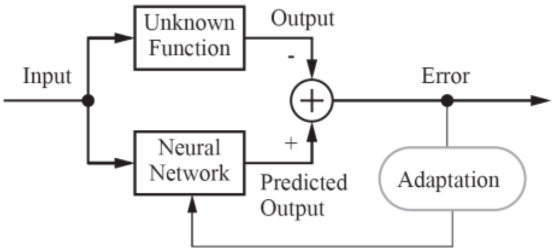


Figure 5: the function approximation structure for neural networks

Even though the above theorem says that there exists an NN that approximates any function f(x), it should be noted that it does not show how to determine the required weights. It is in fact not an easy task to determine the weights so that an NN does indeed approximate a given function f(x) closely enough [24].

VII. THE NN INVERSE MODEL TRAJECTORY TRACKING CONTROLLER

A control structure that integrates the back-stepping controller and a neural network computed torque controller for nonholonomic mobile robots is proposed. A combined kinematic/torque control law is developed using back-stepping and the stability is proved using Lyapunov approach[8,16,18]. The NN controller proposed in this work can deal with unmodeled bounded disturbances and unstructured unmodeled dynamics in the vehicle. The trajectory tracking controller structure using neural networks inverse model is shown in Fig. 6 and The traditional back-stepping controller structure is shown in Fig.7 again for the comparison purpose:

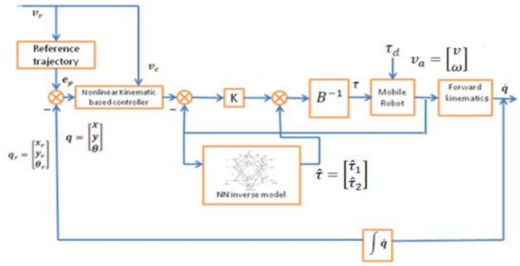


Figure 6: the neural network inverse structure

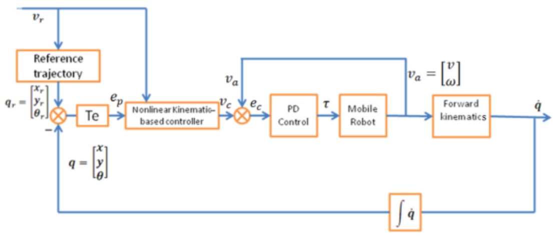


Figure 7: the back-stepping controller model controller structure

Comparing Fig. 6 and Fig. 7, one can find out that in the proposed neural network inverse model control structure, no knowledge about the dynamics of the robot is assumed and the function of the neural network is to learn the vehicle dynamics online and reconstruct the dynamic model of the system [16,20,22]. Given the desired velocity $v_c(t)$ which is the output of the kinematic controller equation (19), the trajectory tracking error will be:

$$e_c = v_c - v \qquad (34)$$

Differentiating the above equation and using the mobile robot dynamic equation (16):

$$\dot{e}_c = \dot{v}_c - \dot{v} \qquad (35)$$

$$\bar{M}(q)\dot{v}(t) + \bar{V}_m(q,\dot{q})v(t) + \bar{F}(\dot{q}) + \bar{G}(q) + \bar{\tau}_d = \bar{B}(q)\tau \qquad (36)$$

The function f is the function containing the mobile robot nonlinearities:

$$f(v, v_c, \dot{v}_c) = \bar{M}(q)\dot{v}_c + \bar{V}_m(q,\dot{q})v_c(t) + \bar{F}(\dot{q}) \qquad (37)$$

As it can be seen from the above equation, f is a function of $v, v_c, \dot{v}_c$ which are all measurable. This function contains all the mobile robot parameters such as masse, moments of inertia and friction coefficient [21, 22, 24]. In the experiment, this function is partially known, therefore a suitable control action for the robot can be written as follows:

$$\bar{\tau} = \hat{f} + K_4 e_c \qquad (38)$$

Where K_4 is the same diagonal positive definite gains used in the back-stepping controller (30) and$\hat{f}$ is an estimate of the robot nonlinear function f and is provided by the neural network inverse model. Substituting the above control rule in the equation (37) we have:

$$\bar{M}(q)\dot{e}_c = -\bar{V}_m(q,\dot{q})e_c - \hat{f} - K_4 e_c + \bar{\tau}_d + f(v, v_c, \dot{v}_c) \qquad (39)$$

We can define the function approximation error as follows:

$$\tilde{f} = f - \hat{f} \qquad (40)$$

Substituting equation (4.76) in (4.75) and simplifying it we have:

$$\bar{M}(q)\dot{e}_c = -(\bar{V}_m(q,\dot{q}) + K_4)e_c + \tilde{f} + \bar{\tau}_d \tag{41}$$

At this stage, the control design problem is how to design the gain matrix K_4 and the estimate $\tilde{f}$ so that both the error signal e_c and the control signal are bounded. The important issue here is that we should design the controller in a way that both the error signal $e_c(t)$ and the control signal stay bounded.

VIII. SIMULATION AND RESULTS

The simulation results of the neural network inverse model controller can be divided into the following two categories: 1.The normal trajectory tracking results without any external disturbance on the robot
2.The trajectory tracking in presence of an external disturbance on the robot motor torques, velocities and position. The simulation results are allocated for comparison study between the normal trajectory tracking between back-stepping controller and neural network inverse model controller. The neural network which is used in the NN controller block in the online inverse model neural network uses the weights obtained from comprehensive offline training. A trajectory tracking comparison with a linear reference trajectory with any external disturbance is shown in Fig. 8:

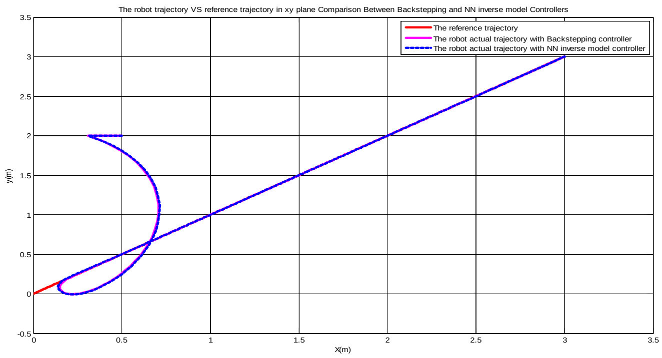


Figure 8: The robot trajectory in x-y plane Back-stepping and NN inverse model controller

The velocity errors of the system which shows the effect of the velocitycontrol part of the system are shown in Fig. 9 and The time response of the two different weights of the neural network is shown in Fig. 10:

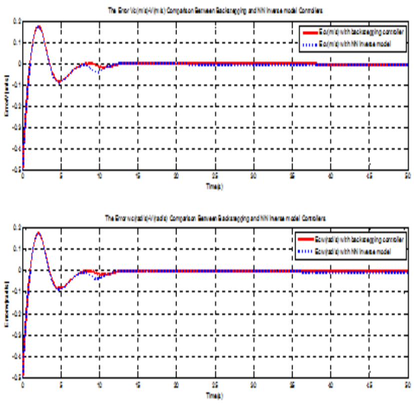


Figure 9: The output velocities Error response comparison between Back-stepping and NN inverse model controller

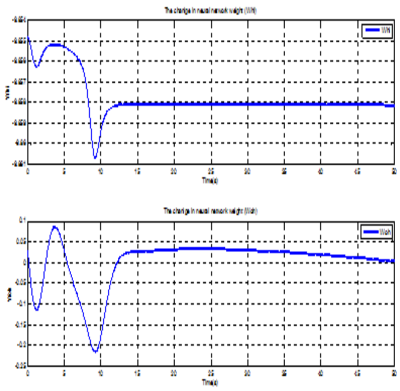


Figure 10: The Neural Network weights response with time

Looking at the above time response, one can find out that the neural network is learning and the gains will become steady and constant after the velocity errors of the system become zero. The other trajectory and robot initial condition that will give a better view on the controller performance is shown in Fig. 11 and The time response of the two sample neural network weights is shown in Fig. 12

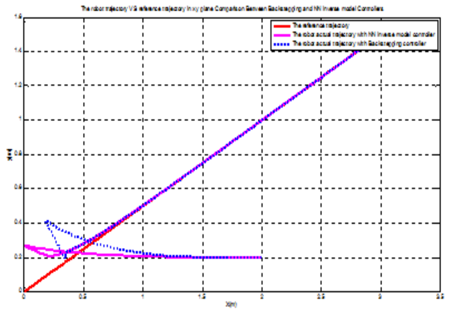


Figure 11: The robot trajectory in x-y plane Back-stepping and NN inverse model controller

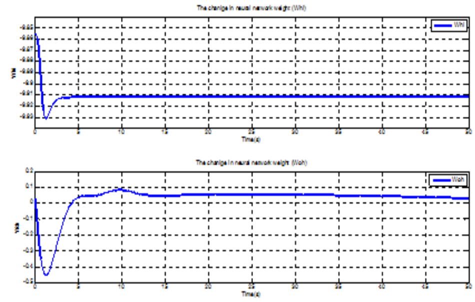


Figure 12: The Neural Network weights response with time

The next step of the performance analysis of this proposed controller is to introduce disturbance to the system and compare the performance with the back-stepping controller. The tracking and disturbance rejection performance of the NN inverse model controller in comparison with the back-stepping controller is shown in Fig. 13:

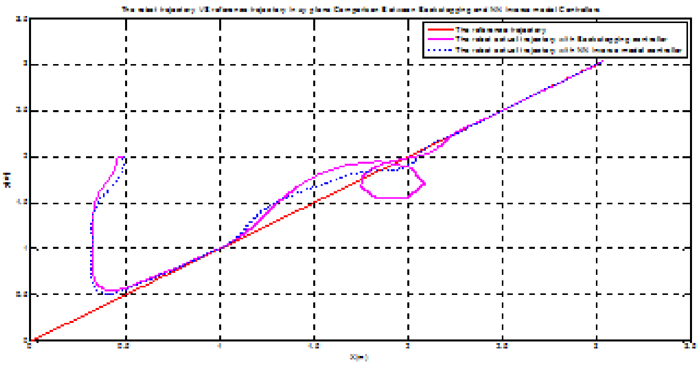


Figure 13: the robot trajectory in X-Y plane with disturbance

The disturbance rejection ability of this controller comes from the fact that the disturbances are occurring in the velocity control loop and the neural network is acting on the velocity errors of the system. The system states errors are shown in Fig. 14 and the system velocity error time responses are shown in Fig. 15:

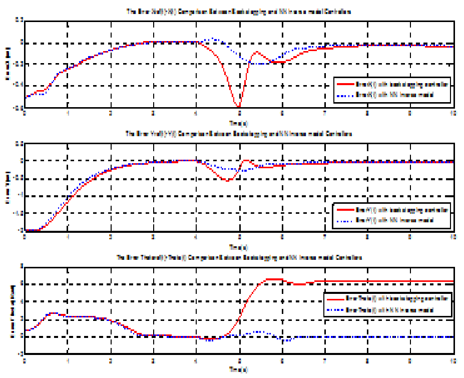


Figure 14: the system states errors with disturbance

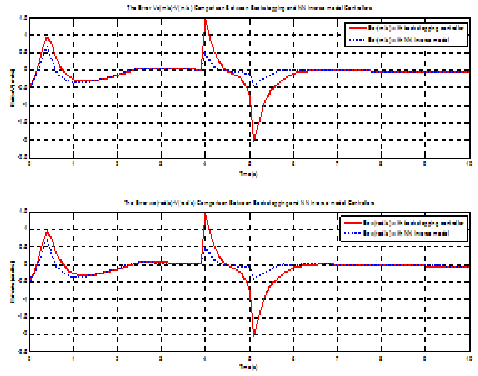


Figure 15: the velocity errors of the system with disturbance

The neural network weights time responses which are shown in Fig. 16 can clear out the neural networks action on the system:

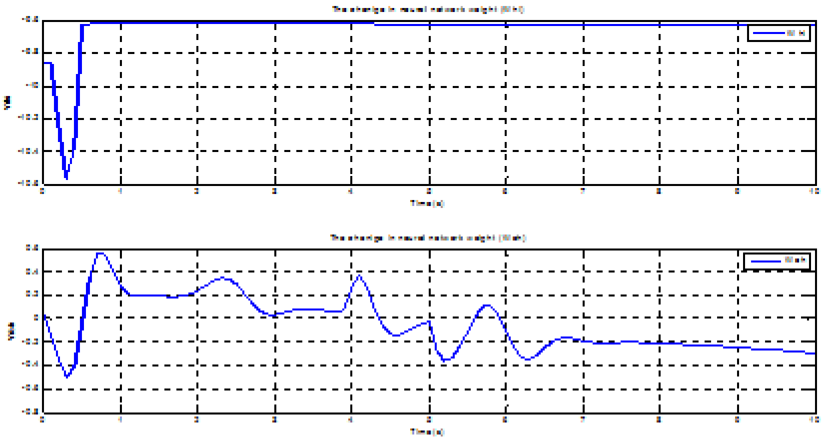


Figure 16: The Neural Network weights response with time with disturbance

IX. CONCLUSION

Designing a novel NN-based adaptive back-stepping controller using the neural network direct model approximation of the robot which highly improves the traditional back-stepping controller tracking performance.The above comparison analysis between the neural network inverse model controller shows that this proposed controller has the following advantages over the back-stepping controller:

1. This controller can deal with unmodeled bounded disturbances and unstructured unmodeled dynamics in the vehicle which are a part of any environment that robot wants to perform the trajectory tracking.
2. No knowledge of the system dynamics and parameters is needed in order to compensate for the nonlinear terms in the system. Especially when a readymade robot platform with unknown parameters and inner layers is used for the tracking task.

The other trajectory tracking controller that can be designed to improve the performance of the above controllers is the one that can adapt itself and the back-stepping controller gains to each trajectory and robot location. The gains of the back-stepping controller in all of the above simulation performances are tuned according to each trajectory and robot initial location. In order to eliminate the back-stepping controller gain tuning step which is a very time consuming and inefficient approach, an adaptive gain tuning controller using the neural network direct model in designed and implemented in the next section of this section.

REFERENCES

[1] G. Artus, P. Morin, and C. Samson, "Tracking of an omnidirectional target with a nonholonomic mobile robot," in *Proc. IEEE Conf. Adv. Robot.(ICAR), 2003, pp. 1468–1473.*

[2] R. Siegwart and I. R. Nourbakhah, Introduction to Autonomous Mobile Robots. Cambridge, MA: MIT Press, 2004.

[3] A. Filipescu, V. Minzu, B. Dumitrascu and E. Minca, Trajectory Tracking and Discrete-Time Sliding-Mode Control of Wheeled Mobile Robot. Proceedings of the IEEE International Conference on Information and Automation, Shenzhen, China, 2011, pp. 27-32.

[4] B. Krose, Environment Representations for Cognitive Robot Companions. ERCM News, No. 53, 2003, pp. 29-30.

[5] R. Barzamini and A. Afshar, "Dynamic adaptive tracking control for wheeled mobile robots," AmirKabir University of Technology, Tehran, Iran, 2006.

[6] R. W. Brockett, "Asymptotic stability and feedback stabilization," in Differential Geometric Control Theory, R. W. Brockett, R. S. Millman, and H. J. Sussmann, Eds. Boston, MA: Birkauser, 1983.

[7] J. M. Yang and J. H. Kim, "Sliding mode control for trajectory tracking of nonholonomic wheeled mobile robots ," in IEEE Transactions on Robotics and Automation , vol. 15, pp. 578-587, June 1999.

[8] Y. Kanayama et al., "A stable tracking control method for an autonomous mobile robot," in Proc. IEEE Conf. Robot. Autom. 1990, pp. 384–389.

[9] G. Campion, G. Bastin, and B. d'Andrea-Novel, "Structural properties and classification of kinematic and dynamic models of wheeled mobile robots," in IEEE Trans. Robot. Automat., vol. 12, pp. 47–62, Jan. 1996.

[10] G. Campion and G. Bastin, "On adaptive linearizing control of omni-directional mobile robots," MTNS, pp. 531-538, Amsterdam, Holland, 1989.

[11] G. Klancar and I. Skrjanc, "Tracking-error model-based predictive control for mobile robots in real time," in Robotics and autonomous systems , vol. 25, pp. 460-469, Sep. 2007.

[12] J. Ye, "Tracking control for nonholonomic mobile robots: Integrating the analog neural network into the backstepping technique," in Neurocomputing, vol. 71, pp. 3373-3378, Dec. 2007.

[13] J. Velagic, N. Osmic and B. Lacevic, " Neural Network controller for mobile robot motion control," in International journal of intelligent systems and technologies, vol. 21, pp. 470-479, Oct. 2008.

[14] R. Fierro and F. L. Lewis, "Control of a Nonholonomic Mobile Robot Using Neural Networks," in IEEE Transactions on Neural Networks, vol. 9, pp. 589-600, July 1998.

[15] R. Fierro and F. L. Lewis, "Control of a nonholonomic mobile robot: backstepping kinematics into dynamics," in Proc. Of 34th IEEE Conf. on Decision and Control, Piscataway, NJ, IEEE press, USA, 1995, pp. 3805-3810.

[16] A. Bloch and S. Drakunov, "Stabilization of a nonholonomic system via sliding modes," in Proc. Of IEEE international conference on Decision Control, Madrid, Spain, 1994, pp. 2961-2963.

[17] K. N. Faress, M. T. El hargy and A. A. El kosy, " Trajectory tracking control for a wheeled mobile robot using fuzzy logic controller," in Proceedings of the 9th WSEAS International Conference on Systems, Athens, Greece,2005, pp. 1261-1269.

[18] P. Morin and C. Samson, "Control of Nonholonomic Mobile Robots Based on the Transverse Function Approach," in IEEE Trans. on Robotics, vol. 25, pp. 1058-1073, Oct. 2009.

[19] D. Buccieri, D. Perritaz, P. Mullhaupt, Z. P. Jiang and D. Bonvin, "Velocity-Scheduling Control for a Unicycle Mobile Robot:

[20] L. Ge, R. Li, D. Yu and L. Zhao, " A Nonlinear Adaptive Variable Structure Trajectory Tracking Algorithm for Mobile Robots," in Intelligent robotics and applications, vol. 15, pp. 892-900, Nov. 2008.

[21] G. Oriolo, A. D. Luca and M. Vendittelli, "WMR Control via Dynamic Feedback Linearization: Design, Implementation, and Experimental Validation," in IEEE Trans. on Control Systems Thechnology , vol. 10, pp. 835-852, Nov. 2002.

[22] F. Dong and W. L. Xu, "Adaptive Tracking Control of Uncertain Nonholonomic Dynamic System," in IEEE Transactions on Automatic Control, vol. 46, pp. 450-454, March 2001.

[23] O. Nells, Nonlinear System Identification. London: Springer-Verlag, 2001.

[24] J. Ye, Adaptive Control of Nonlinear PID-Based Analogue Neural Network for a Nonholonomic Mobile Robot. Neurocomputing, Vo. 71, 2008, pp. 1561-1565.

[25] T. Das, I.N. Kar and S. Chaudhury, Simple Neuron-Based Adaptive Controller for a Nonholonomic Mobile Robot Including Actuator Dynamics. Neurocomputing, Vol. 69, 2006, pp. 2140-2151.

[26] A. Filipescu, V. Minzu, B. Dumitrascu and E. Minca, Discrete-Time Sliding Mode Control of Wheeled Mobile Robots. Proceedings of the 8th Asian Control Conference ASCC, Kaohsiung, Taiwan, May 15-18, 2011, pp. 771-776.