

PAPER • OPEN ACCESS

System Identification of Two-Wheeled Robot Dynamics Using Neural Networks

To cite this article: N Uddin 2020 *J. Phys.: Conf. Ser.* **1577** 012034

View the [article online](#) for updates and enhancements.

You may also like

- [Machining of nickel-based super alloy inconel 718 using alumina nanofluid in powder mixed electric discharge machining](#)
Rakesh Kumar Patel and Mohan Kumar Pradhan
- [Impact of SiC Particle Incorporated Dielectric Medium on Machining Performance of AA7050/SiC/Al₂O₃ Hybrid Composites](#)
S. Syath Abuthakeer, Y. Aboobucker Parvez and J. Nashreen
- [Investigation of effect of thickness in micro drilling of Ti-6Al-4V alloy using tungsten copper electrode](#)
Sachin N Sawant and Sachin K Patil



ECS
The
Electrochemical
Society
Advancing solid state &
electrochemical science & technology

DISCOVER
how sustainability
intersects with
electrochemistry & solid
state science research

System Identification of Two-Wheeled Robot Dynamics Using Neural Networks

N Uddin¹

¹Department of Informatics, Universitas Pembangunan
Jaya Tangerang Selatan, Indonesia

E-mail: nur.uddin@upj.ac.id¹

Abstract. A system identification of two-wheeled robot (TWR) moving on planar space is presented by applying using neural networks. The system identification is to model the TWR dynamics which is a nonlinear system. The model is applied for estimating the TWR posture during the movements. Neural networks applied in the system identification is multi later perceptron. The neural networks consists of three layers with eight neurons at the first layer, five neurons at the second layer, and three neurons at the third layers. The neural networks is trained to model the TWR dynamics based on a set of input and output data. The system identification is demonstrated through computer simulations. The results show that the system identification using neural networks is able to model the TWR dynamics. The neural networks with learning rate 0.005 is able to estimate the TWR posture with convergence time 0.5 seconds.

1. Introduction

A two-wheeled robot (TWR) is a ground mobile robot that only uses two wheels to support the robot body. Each wheels is independently driven by a high-torque electric motor. The robot has an advantage of higher maneuverability due to the two-wheeled support. However, the two wheels support gives a stability problem to the robot. The robot is unstable in longitudinal mode where the pitching is statically unstable. An active stabilization system is introduced to stabilize the robot. The active stabilization system is actively stabilizing the TWR pitching movement using a state feedback control. Several works on developing the active stabilization of TWR have been presented by applying different control methods [1–4].

The active stabilization system makes the TWR stable and be readily applied in many applications. The high maneuverability makes the TWR to be a potential vehicle in developing a high-maneuver autonomous ground vehicle. An autonomous vehicle is a vehicle that has ability to move autonomously from one location to another location. The vehicle needs be equipped by a control system that acts as a vehicle driver. Such kind of the control system is known as a trajectory tracking control system.

Several works on developing autonomous vehicle based on the TWR have been reported. The first autonomous TWR was reported in [5]. The TWR was equipped by a controller that has three functions: 1) balancing the vehicle body, 2) control the vehicle speed, and 3) steering the vehicle to track a straight line trajectory. The controller is a multi input multi output (MIMO) controller that was designed using the optimal control method. The experimental results showed that the TWR was able to move autonomously on a desired straight line path. Another work on developing autonomous two-wheeled vehicle was presented in [6]. Trajectory tracking control



system developed in the work was designed using partial state states feedback linearization control method. The trajectory tracking control system is the main part of autonomous vehicles. Different control methods have been applied in designing the trajectory tracking system, for examples: optimal control [7], adaptive control [8–10], and predictive control [11].

The presented works on autonomous TWR were mostly applying model-based control methods in designing the trajectory tracking control systems. A mathematics model representing the TWR dynamics moving on a planar space is required the trajectory tracking control system design using model-based control methods. The mathematics model can be obtained through a system modeling. The system modeling results in a mathematics model by deriving the dynamics equations of a system based on the physical laws. Several works on system modeling of the TWR dynamics moving on planar space have been carried out through a derivation of the robot kinematics [7–12].

The system modeling is not the only method to obtain a mathematics model of a system dynamics. The mathematics model can also be obtained through a system identification [13–17]. The system identification determines a mathematics model of system dynamics by finding a function that approximates a relationship between input and output of the system dynamics. A set data of input and output is required in the system identification. The system identification can be done in time domain and frequency domain [17–19].

Artificial neural networks are nonlinear parallel signal processing inspired by the human brain [20]. The artificial neural networks are commonly termed by neural networks in the scientific publication as well as in this paper. The neural networks have capability to approximate any piece-wise continuous functions and including linear function as well as nonlinear function. The capability makes the neural networks to be applicable in system identification of linear and nonlinear systems. Several works on applying neural networks in system identification have been presented [21–25].

This paper presents a system identification using neural networks of two-wheeled robot dynamics moving on planar space. Presentation of the paper is organized as follows. Section I gives introduction, literature review of related works, and motivation of the work. Section II presents a derivation of the robot dynamics. Section III gives an overview of neural networks. Section IV presents a system identification using neural networks. Section V demonstrates the TWR system identification using neural networks through simulation. Finally, conclusion is presented in Section VI.

2. Two-Wheeled Robot Dynamics on Planar Space

A two-wheeled robot uses only two wheels to support the robot body. Both wheels are driven by two-independent direct-current (DC) motors. Figure 1 shows a TWR on a planar space. Position and orientation of the robot on the planar space is known as the robot posture and defined as follows:

$$\xi = \begin{bmatrix} x \\ y \\ \psi \end{bmatrix}, \quad (1)$$

where ξ is the robot posture, x and y are the robot position with respect to a reference coordinate system, and ψ is the orientation angle of the robot with respect to the reference coordinate system. Rotation of the DC motors drives the wheels such that the robot moves with linear velocity u and angular velocity r . The angular velocity is due to the difference rotational speed of the both motors. The moving robot makes the robot posture to be time varying. The robot posture dynamics are described by the following equation:

$$\begin{aligned} \dot{x} &= u \cos \psi \\ \dot{y} &= u \sin \psi \\ \dot{\psi} &= r. \end{aligned} \quad (2)$$

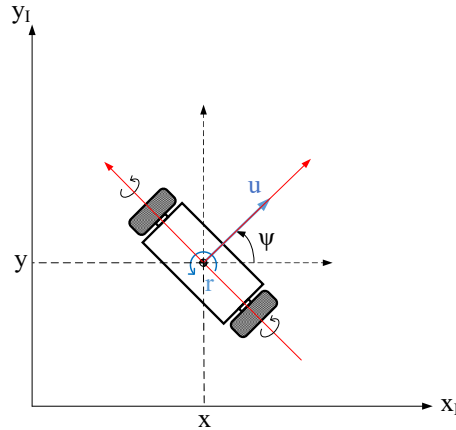


Figure 1. A two-wheeled robot on planar space and the velocities diagram.

The (2) is a mathematics model of TWR dynamics moving on planar space. The model is expressed by a nonlinear time differential equation. The model can be transformed into a difference equation based on the following relation:

$$\dot{x} = \frac{dx}{dt} = \frac{x(k+1) - x(k)}{\Delta t} \quad (3)$$

$$\dot{y} = \frac{dy}{dt} = \frac{y(k+1) - y(k)}{\Delta t} \quad (4)$$

$$\dot{\psi} = \frac{d\psi}{dt} = \frac{\psi(k+1) - \psi(k)}{\Delta t} \quad (5)$$

where Δt is the sampling time, k is the present sample, and $k+1$ is the next sample. The transformation will result in a nonlinear difference equation of the TWR dynamics model as follows:

$$\begin{aligned} x(k+1) &= x(k) + u(k)\Delta t \cos \psi(k) \\ y(k+1) &= y(k) + u(k)\Delta t \sin \psi(k) \\ \psi(k+1) &= \psi(k) + r(k)\Delta t. \end{aligned} \quad (6)$$

The difference equation (6) is very useful to simulate the TWR dynamics in computer.

3. Neural Networks

Neural networks have several types and one of them is a multi-layers perceptron (MLP). Figure 2 shows an architecture of MLP. The MLP consists of three layers. Output of the first layer is given by the following equation:

$$y_1 = \varphi_1(w_1 z + b_1), \quad (7)$$

where y_1 is the output vector of the first layer, φ_1 is the activation function of neurons at the first layer, w_1 is the weight matrix of the first layer, z is the input vector of the first layer, and b_1 is the bias vector of the first layer. Output of the first layer becomes the input for the second layer. The second layer output is computed using the following equation:

$$y_2 = \varphi_2(w_2 y_1 + b_2), \quad (8)$$

where y_2 is the output vector of the second layer, φ_2 is the activation function of neurons at the second layer, w_2 is the weight matrix of the second layer, and b_2 is the bias vector of the second

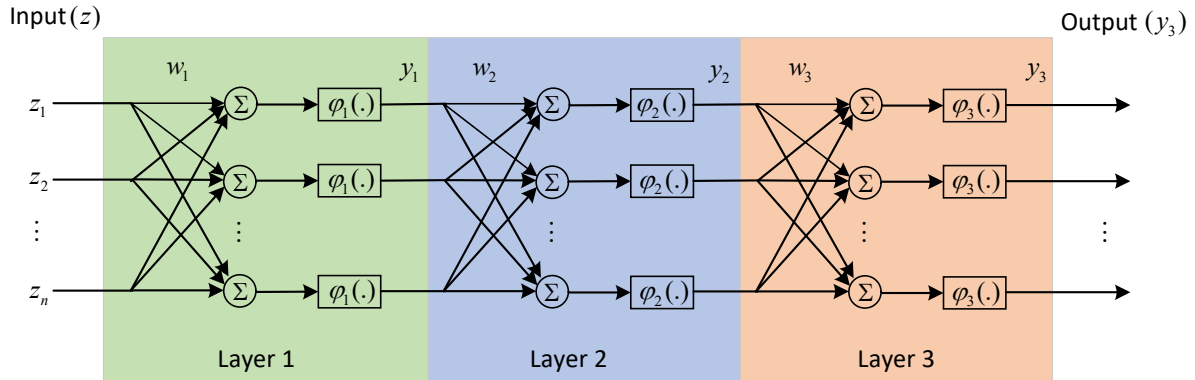


Figure 2. An architecture of three layers neural networks.

layer. The second layer output becomes the third layer input. Computation in the third layer results in output that is given by the following equation:

$$y_3 = \varphi_3(w_3 y_2 + b_3), \quad (9)$$

where y_3 is the third layer output vector, φ_3 is the activation function of neurons at the third layer, w_3 is the weight matrix of the third layer, and b_3 is the bias vector of the third layer. The third layer output is the MLP output as shown in Figure 2.

A linear function and a hyperbolic tangent are commonly applied as the activation function of multi layer perceptrons. The linear function is given by the following equation:

$$\varphi(v) = v, \quad (10)$$

while the hyperbolic tangent is given by:

$$\varphi(v) = \frac{e^v - e^{-v}}{e^v + e^{-v}}. \quad (11)$$

The linear function is commonly applied as the activation function of the last layer, while the hyperbolic tangent is commonly used as the activation function of the other layers.

Suppose the neural networks output is different to the desired output such that the neural networks produce error. The error is defined by the following equation:

$$e = d - y_3 \quad (12)$$

where e is the error of neural networks, d is the desired output, and y_3 is the actual output of neural networks. Neural networks have capability to improve their performance through a learning process. The learning process to update the neural networks parameters such that the error is minimized. The error minimization is done based on an optimization of an error cost function. A quadratic function is commonly used as the error cost function as given as follows:

$$E = \frac{1}{2} e^T e, \quad (13)$$

where E is the error cost function, e is the neural networks error given in (12), and the superscript $(.)^T$ is the transpose operator.

During the learning process, parameters of the neural networks, i.e: weight and bias are updated based on the following equation:

$$w_{new} = w_{cur} + \Delta w \quad (14)$$

where w_{new} is the updated weight, w_{cur} is the current weight, and Δw is the correction weight. Similarly, the bias is updated as follows:

$$b_{new} = b_{cur} + \Delta b \quad (15)$$

where b_{new} is the updated bias, b_{cur} is the current bias, Δb is the correction bias. The correction weight and the correction bias are proportional to the derivative of the error cost function with respect to the weight and bias, respectively. Both correction weight and bias are given as follows:

$$\Delta w = \eta \frac{\partial E}{\partial w} \quad (16)$$

$$\Delta b = \eta \frac{\partial E}{\partial b} \quad (17)$$

where E is the error cost function, w is the weight of a layer, b is the bias of a layer, and η is the learning rate. The learning rate is a constant representing the learning process speed of neural networks.

4. System Identification Using Neural Networks

The TWR dynamics on the planar space is a nonlinear system. It was described by a nonlinear difference equation in (6). The equation be generally expressed by:

$$\xi(k+1) = F_d[\xi(k), \mu(k)] \quad (18)$$

where ξ is the TWR posture, F_d is a nonlinear function, and μ is an input vector. The input vector is defined by:

$$\mu(k) = \begin{bmatrix} u(k) \\ r(k) \end{bmatrix}. \quad (19)$$

where u is the linear velocity and r is the angular velocity.

Neural networks is applied in system identification of the TWR dynamics. Since the neural networks have capability to approximate any functions, the TWR dynamics is modeled as follows:

$$\hat{\xi}(k+1) = F_n[z(k)] + B\mu(k) \quad (20)$$

where $\hat{\xi}$ is the estimated posture, F_n is a nonlinear function, $z(k)$ is the input of the function F_n , and B is a three by two constant matrix. The matrix B is known by selecting an arbitrary value. The nonlinear function F_n is an unknown function and it will be approximated by the neural networks. The posture robot is assumed to be available through measurements and the measurement noises are neglected. The current posture data, $\xi(k)$, is used as the input of neural networks to estimate the next posture, $\xi(k+1)$. Block diagram of the system identification is shown in Figure 3.

The different between the actual posture and the estimated posture is known as the estimation error. The estimation error at time k is defined as follows:

$$e_m(k) = \xi(k) - \hat{\xi}(k) \quad (21)$$

where $e_m(k)$ is the estimation error at time k .

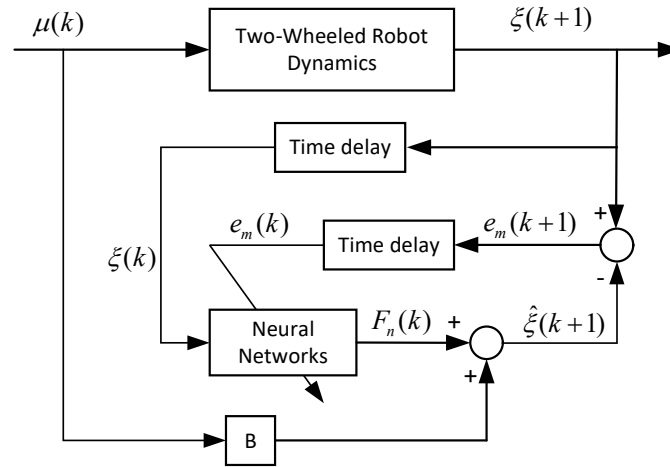


Figure 3. Block diagram of modeling TWR dynamics using neural networks.

The estimation error is equal to the neural networks error based on the two following facts. First, the (20) shows that the model has the same input as the actual system (18), i.e., $\mu(k)$. Second, the matrix B in the model is a constant matrix. The neural networks is trained to obtain F_n such that the estimation error is minimized. The training is done using the learning algorithm given in (14)-(17), where the cost function is now given by:

$$E(k) = \frac{1}{2} e_m^T(k) e_m(k). \quad (22)$$

5. Simulation

The system identification of TWR using neural networks is demonstrated through computer simulations. A simulation program is built based on the simulation diagram in the Figure 3. The nonlinear function F_n in the model is approximated by three layers neural networks. Each layers consists of different number of neurons, where eight neurons at the first layer, five neurons at the second layer, and three neurons in the third layer. Weight and bias of the neurons are initialized by small random numbers. The neural networks has three inputs as follows:

$$z(k) = \xi(k) = \begin{bmatrix} x(k) \\ y(k) \\ \psi(k) \end{bmatrix}. \quad (23)$$

The inputs are the actual posture of TWR that is assumed to be available through measurement and the noise measurements are neglected. The matrix B in (20) has size three by two and can be chosen arbitrary. In this case, the matrix B is defined as follows:

$$B = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}. \quad (24)$$

The system identification is simulated to estimate posture of a TWR moving on planar space. The TWR moves at a constant linear velocity 3 m/s and a constant angular velocity 1 rad/s. Two simulations are presented by varying the learning rate. The first simulation uses the neural networks with $\eta = 0.001$, while the second simulation uses neural networks with

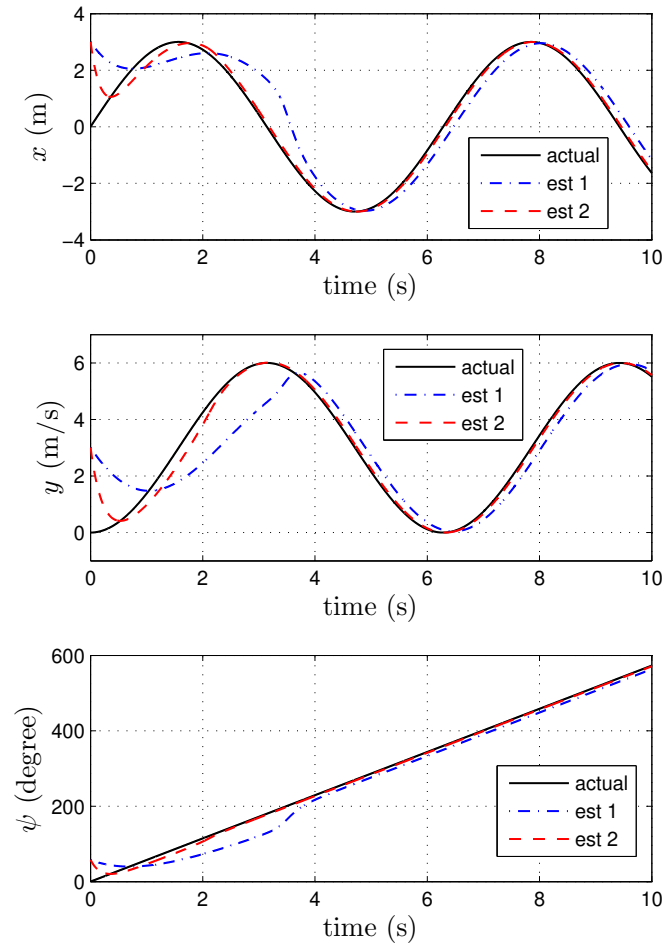


Figure 4. Simulation results of system identification using neural networks for estimating a two-wheeled robot posture.

learning rate $\eta = 0.005$. Both simulation results are shown in Figure 4. The estimated TWR posture using the neural networks with learning rate 0.001 is shown by the legend "est 1" in the figure. The legend "est 2" in the figure shows the estimated TWR posture using neural networks with learning rate 0.005. The simulation results show that estimated posture using the neural networks with learning rate 0.005 was able to converge to the actual posture in 0.5 seconds while using the neural networks with learning rate 0.001 requires 3.7 seconds. The neural networks with learning rate 0.005 results in less estimation error than the neural networks with learning rate 0.001.

6. Conclusion

A system identification using neural networks has been applied to estimate the posture of moving two-wheeled robot (TWR) on planar space. The neural networks were multi layers perceptron. The applied neural networks in the system identification consisted of three layers with eight neuron at the first layer, five neurons at the second layer, and three neuron at the third layer.

Two simulations of the TWR system identification were carried out by varying learning rate of the neural networks, 0.001 and 0.005. The simulation results shown that the neural networks with both learning rate were able to estimate the TWR posture. The neural networks with higher learning rate resulted in better estimation than the neural networks with lower learning rate. The neural networks with learning rate 0.005 were able to estimate the TWR posture with converging time 0.5 seconds which is quite fast in the online system identification. The fast converging makes the estimation results be applicable in a control system.

7. Further Work

The system identification results will be applied in developing an adaptive trajectory tracking control system of the two-wheeled robot.

8. References

- [1] Nomura T, Kitsuka Y, Suemitsu H and Matsuo T 2009 Adaptive backstepping control for a two-wheeled autonomous robot *ICCAS-SICE, 2009* (IEEE) pp 4687–4692
- [2] Chan R P M, Stol K A and Halkyard C R 2013 *Annual Reviews in Control* **37** 89–103
- [3] Uddin N 2017 Lyapunov-based control system design of two-wheeled robot *Computer, Control, Informatics and its Applications (IC3INA), 2017 International Conference on* (IEEE) pp 121–125
- [4] Uddin N, Nugroho T A and Pramudito W A 2017 Stabilizing two-wheeled robot using linear quadratic regulator and states estimation *Proc. of the 2nd International conferences on Information Technology, Information Systems and Electrical Engineering (ICITISEE)* (Yogyakarta, Indonesia)
- [5] Ha Y S *et al.* 1996 *Robotics and autonomous systems* **17** 65–80
- [6] Pathak K, Franch J and Agrawal S K 2005 *IEEE Transactions on robotics* **21** 505–513
- [7] Uddin N 2018 *Jurnal Infotel* **10** 90–97 ISSN 2460-0997
- [8] Cui R, Guo J and Mao Z 2015 *Nonlinear Dynamics* **79** 501–511
- [9] Yue M, Wang S and Sun J Z 2016 *Neurocomputing* **191** 44–54
- [10] Uddin N 2019 *Journal of Physics: Conference Series* **1373** 012052 URL
- [11] Yue M, An C and Sun J Z 2018 *International Journal of Control, Automation and Systems* **16** 265–274
- [12] Uddin N 2018 A robot trajectory tracking control system design using pole domination approach 2018 *IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)* (Vancouver) pp 506–512
- [13] Keviczky R H L 1999 *Nonlinear system identification: input-output modeling approach* (Norwell, MA: Kluwer)
- [14] Keesman K J 2011 *System identification: an introduction* (Springer Science & Business Media)
- [15] Pintelon R and Schoukens J 2012 *System identification: a frequency domain approach* (John Wiley & Sons)
- [16] Johansson R 1993 *System modeling and identification* vol 1 (Prentice Hall Englewood Cliffs, NJ)
- [17] Morelli E A and Klein V 2016 *Aircraft system identification: Theory and practice* (Sunflyte Enterprises Williamsburg, VA)
- [18] Jategaonkar R V 2015 *Flight vehicle system identification: A time-domain methodology* (American Institute of Aeronautics and Astronautics, Inc.)
- [19] Tischler M B and Cauffman M G 1992 *Journal of the American Helicopter Society* **37** 3–17
- [20] Haykin S S 2009 *Neural networks and learning machines* 3rd ed (Upper Saddle River, NJ: Pearson Education)
- [21] Harris J, Arthurs F, Henrickson J V and Valasek J 2016 Aircraft system identification using artificial neural networks with flight test data 2016 *International Conference on Unmanned Aircraft Systems (ICUAS)* (IEEE) pp 679–688
- [22] Chen S, Billings S and Grant P 1990 *International journal of control* **51** 1191–1214
- [23] Chen S and Billings S 1992 *International journal of control* **56** 319–346
- [24] Vargas J A, Grzeidak E and Alfaro S C 2016 Identification of unknown nonlinear systems based on multilayer neural networks and lyapunov theory 2016 *IEEE Symposium Series on Computational Intelligence (SSCI)* (IEEE) pp 1–7
- [25] Vargas J A, Pedrycz W and Hemerly E M 2019 *Neurocomputing* **329** 86–96

Acknowledgments

The author acknowledges financial support from the Universitas Pembangunan Jaya through an internal research grant in the scheme of Penelitian Unggulan Perguruan Tinggi 2019 (PUPT-2019).