electronics

MDPI

Article

Efficient System Identification of a Two-Wheeled Robot (TWR) Using Feed-Forward Neural Networks

Muhammad Aseer Khan [1,*], Dur-e-Zehra Baig [2], Husan Ali [1], Bilal Ashraf [1], Shahbaz Khan [1], Abdul Wadood [1,*] and Tariq Kamal [3,*]

[1] Department of Electrical Engineering, Air University, Aerospace & Aviation Campus, Kamra 43570, Pakistan
[2] Faculty of Electrical Engineering, Ghulam Ishaq Khan Institute of Engineering Sciences and Technology, Topi 23640, Pakistan
[3] School of Technology and Innovations, Electrical Engineering, University of Vaasa, 65200 Vaasa, Finland
* Correspondence: aseer.khan@au.edu.pk (M.A.K.); abdulwadood@aack.au.edu.pk (A.W.); tariq.kamal@uwasa.fi (T.K.)

Abstract: System identification of a Two-Wheeled Robot (TWR) through nonlinear dynamics is carried out in this paper using a data-driven approach. An Artificial Neural Network (ANN) is used as a kinematic estimator for predicting the TWR's degree of movement in the directions of x and y and the angle of rotation Ψ along the z-axis by giving a set of input vectors in terms of linear velocity 'V' (i.e., generated through the angular velocity 'ω' of a DC motor). The DC motor rotates the TWR's wheels that have a wheel radius of 'r'. Training datasets are achieved via simulating nonlinear kinematics of the TWR in a MATLAB Simulink environment by varying the linear scale sets of 'V' and '(r ± Δr)'. Perturbation of the TWR's wheel radius at Δr = 10% is introduced to cater to the robustness of the TWR wheel kinematics. A trained ANN accurately modeled the kinematics of the TWR. The performance indicators are regression analysis and mean square value, whose achieved values met the targeted values of 1 and 0.01, respectively.

Keywords: multiple-input multiple-output (MIMO); system identification; neural network implementation; neural networks; nonlinear systems; two-wheeled robot (TWR); multi-layer perceptron

Citation: Khan, M.A.; Baig, D.-e.-Z.; Ali, H.; Ashraf, B.; Khan, S.; Wadood, A.; Kamal, T. Efficient System Identification of a Two-Wheeled Robot (TWR) Using Feed-Forward Neural Networks. *Electronics* **2022**, *11*, 3584. https://doi.org/10.3390/electronics11213584

Academic Editors: George A. Papakostas and Josip Musić

Received: 7 September 2022
Accepted: 31 October 2022
Published: 2 November 2022

1. Introduction

This research aims to accurately model the dynamics of a Two-Wheeled Robot (TWR) using a data-driven approach through its nonlinear kinematics. An Artificial Neural Network (ANN) was used as a dynamic model for predicting the TWR's degree of movement and angular rotation. ANN learning datasets were obtained from the fundamental kinematics of the TWR, which is nonlinear. Results show that the ANN accurately modeled the dynamics of the TWR.

The literature revealed that robotics played a vital role in the advancement of many research domains, such as Modeling and Controls [1–3], Signal Processing, and Measurement and Instrumentation [4]. With the advancements in control engineering, Autonomous Wheel Robots (AWR) are mostly used in hazardous areas where human intervention is not permissible [5]. Moreover, the feedback control system design that is developed for AWR is simpler compared to legged robots [6,7]. The dynamics of an autonomous TWR are similar to an inverted pendulum. Therefore, a TWR is nonlinear and unstable in its kinematics motion [8] and thus difficult to capture through experimental observations. Moreover, designing the control system requires an accurate estimation of AWR/TWR [9]. In this study, the ANN was adopted as an estimator for TWR dynamics, which was identified through a data-driven approach that was achieved from a nonlinear model of the TWR.

System identification using experimental datasets and datasets run through the nonlinear mathematical model make space for intelligent control systems. While designing a nonlinear control system does not require complex mathematics, system identification

using measured input/output datasets requires experimental design, experimental data analyses, and selection of the appropriate model structure based on standard criteria such as AIC, parameter estimation of a known structure model, and model validation [10]. There are several methodologies used for system identification using a data-driven approach, such as a time series model (e.g., ARX [11], the grey-box model [12], and the black-box model, i.e., ANN [13]). To reduce complexity when designing Nonlinear Control Systems, we adopted a system identification using the black-box model and trained the ANN as the estimated model of a TWR dynamic using datasets acquired from the simulation studies of the nonlinear dynamics of a TWR. System identification of a two-wheeled, self-balancing robot was performed using the experimental datasets in two different phases [14]. This study compares linear models such as ARX, ARMAX, Bl, and OE with nonlinear models such as Wiener and Hammerstein. In all analyses, it is shown that executing saturation as an output nonlinearity block in a Wiener-type model structure is capable of modeling highly nonlinear systems with acceptable accuracy. Many autonomous TWR systems used for desired-trajectory tracking are developed using reinforcement learning (RL) with conventional control techniques that balance and self-control the TWR [15–20]. These techniques do not require supervised learning; most require the real-time experience of a robot that learned the environment and planned the trajectory. However, unsupervised learning, i.e., RL, faces stability issues because of robotics kinematics [21]. Thus, an expert system based on a sensory, data-driven approach may require information on TWR kinematics. Thus, prior to designing the control law for a TWR system, an ANN must be supervisory-learned and replicate the TWR kinematics. System identification of a TWR using an ANN at a single operating condition was carried out in [13]. In this scenario, the ANN modeled the TWR kinematics when the input linear velocity of the DC motor was constant. Recently, we identified the TWR kinematics using a nonlinear time series model by generating the datasets at various operating conditions [22]. The identified dynamics validated the fundamental nonlinear kinematics of the TWR. However, this technique could not be utilized for designing the ANN-based control system.

In this paper, the dynamics of a nonlinear TWR are accurately modeled by training a multilayer ANN using the datasets that were obtained in [22]. The obtained results indicate that the ANN accurately modeled the behavior of TWR kinematics. The paper outline is described as follows: Section 1 describes the introduction to the presented work and also briefly reveals the literature review relevant to the TWR kinematics for modeling and controls. Section 2 discusses the adopted methodology that was used for the system identification of a TWR. Section 3 briefly explains the fundamental nonlinear model of a TWR, which is used for obtaining the input/output datasets. Section 4 describes the ANN estimation using the datasets that were obtained from the nonlinear model of the TWR. Section 5 presents the simulation results, and Section 6 concludes the obtained results.

2. Methodology for System Identification of TWR

The nonlinear differential equations that represent the input/output dynamic of a TWR were implemented in Simulink. The training datasets at various operating conditions were achieved via the nonlinear kinematics of TWR. Each dataset consisted of more than 600 samples. The operating conditions were varied in terms of linear velocity 'V' of a DC motor and the perturbation of the radius of a wheel 'Δr'. This perturbation, the variation from the actual radius of a wheel 'r' of a robot, was about ±10r. The obtained simulation dataset was exported to MATLAB workspace for analysis and ANN training purposes. ANN in a feedforward network had hidden layer with 100 neurons between inputs and outputs layers. A total of 70% of the acquired datasets was used for training, while 30% was used for validation and testing purposes. Comparative analysis between the acquired datasets and the estimated datasets was carried out using regression analyses and computing Mean Square Error (MSE) at various operating conditions. Regression analysis achieved a value of 1, and MSE achieved a value of 0.01. Both performances measured achieved the desired benchmark.

3. Dynamics of a Two-Wheeled Robot (TWR)

The dynamics of the TWR are considered when it is moving on a planar surface, and its diagram is shown in Figure 1 [13].

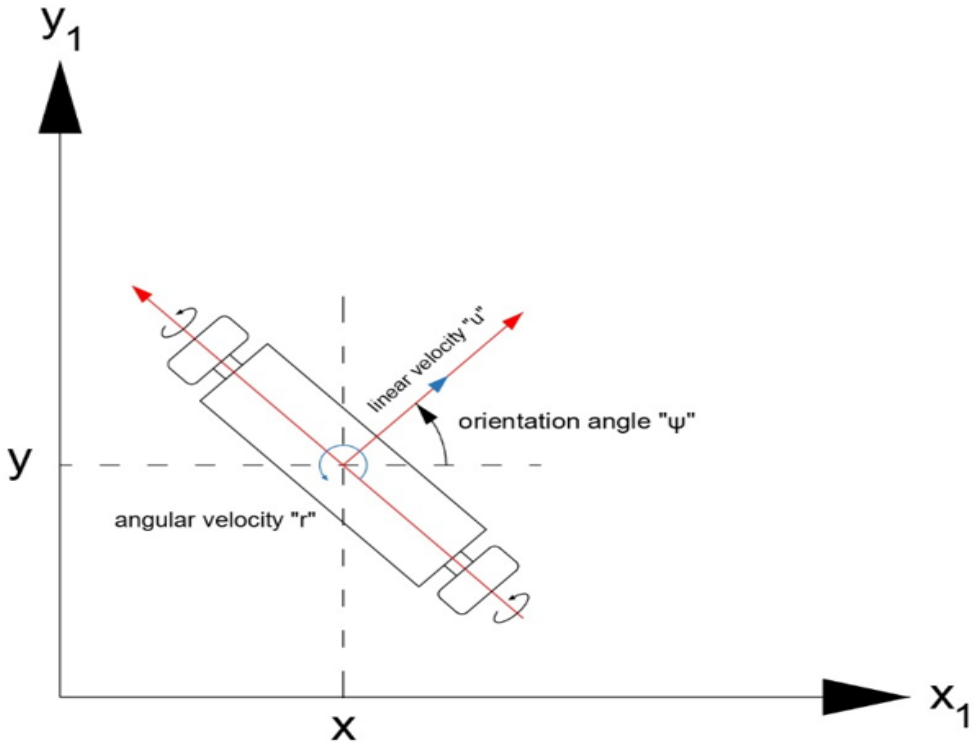

Figure 1. Dynamics of TWR on a planar surface and velocity diagram.

The TWR's motion changes with respect to time, so its position coordinates (both x and y) and orientation angle Ψ are represented by Equation (1) given below.

$$\mathrm{O} = \begin{bmatrix} x \\ y \\ \Psi \end{bmatrix}, \tag{1}$$

The wheels of the TWR are driven by two high-torque DC motors; both motors have a difference in rotational speed, which enables the TWR to move with a linear velocity 'u' and an angular velocity 'r'. Velocity 'u' is along the y-axis, while velocity 'v' is along the x-axis. The differential equations, which describe the position coordinates and orientation angle, are represented by Equations (2)–(4).

$$\dot{x} = \mathrm{u}\cos\Psi - \mathrm{v}\ \sin\ \Psi, \tag{2}$$

$$\dot{y} = \mathrm{u}\sin\Psi + \mathrm{v}\ \cos\ \Psi \tag{3}$$

$$\dot{\Psi} = r \tag{4}$$

4. System Identification of TWR Using Artificial Neural Networks

Firstly, the work on neurons was presented by a neurophysiologist named Warren McCulloch and a mathematician named Walter Pitts in 1943 [23]. They published their work in a paper explaining how the neuron functions. They validated their concepts by modeling a simple neural network using electrical circuits. Their presented model of the neuron is named the McCulloch–Pitts (MCP) neuron, which played an important role in the development of ANN. Some additional features were added to this model, as it has some shortcomings. Later, in 1958, the concept of a perceptron was introduced by Frank

Rosenblatt [24]. It is actually an MCP neuron, and the inputs are initially passed through some preprocessors called 'association units'.

An ANN is a complex network used to represent artificial neurons. It receives the data as the input signal and processes it through a complex neuron structure to obtain the desired output. The available input/output datasets are used to train the ANN. There are three types of learning categories: supervised learning, unsupervised learning, and reinforcement learning. In supervised learning, input and output data and labels are fed to algorithms. They then predict the desired results based on the training patterns between input and output. In unsupervised learning, there are no labels, and output is identified according to the patterns identified by the neural network within the data. It means there is no human involvement in unsupervised learning. In reinforcement learning, there is a feedback element that is provided by the human to the neural network as to whether the prediction is right or wrong.

The basic building blocks i.e. artificial neuron or perceptron consist of input, weight, bias, activation function, and output. The neuron is fed with multiple inputs and gives a single output. A neural network has layers of neurons, namely input layers, hidden layers, and output layers, respectively. The input data are received by the input layer, computations are performed by the hidden layers, and at last, the output layer predicts the output [25]. Figure 2 represents the working scenario of ANNs for the system of a TWR. On the left side, there are three inputs, which are fed to the ANN as the input layer and then passed on to the hidden layer with 100 neurons. The ANN is trained for the respective outputs so that it can correctly predict the true outputs, when they will be validated, and tests new inputs.

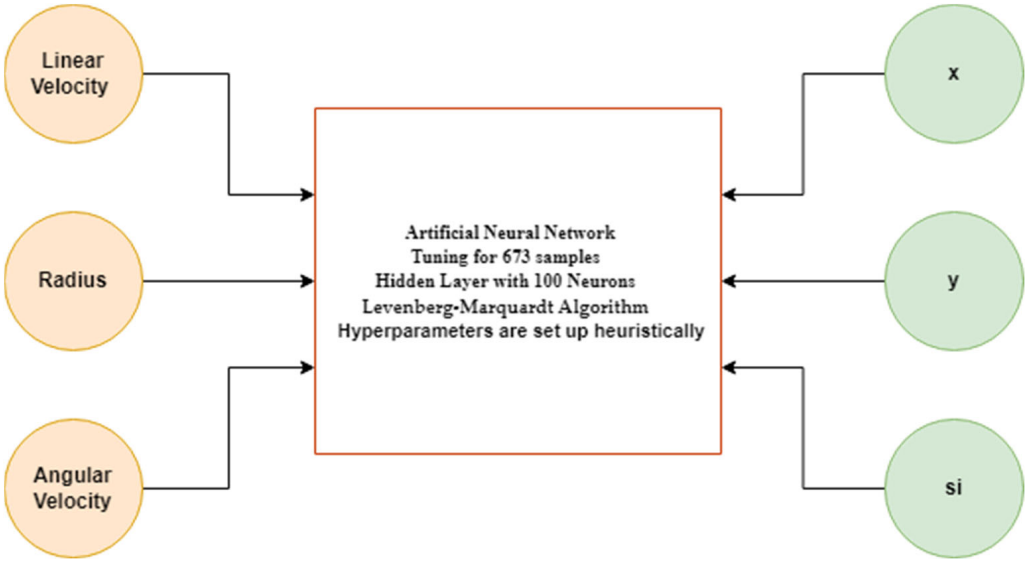

Figure 2. Schematic diagram of ANN of TWR.

In the input data, weights are assigned to each input to convert those inputs into hidden layers. Weights are multiplied with the input data, and their sum is provided to neurons present in the hidden layers. The neuron is then fed with a bias value. All the values are summed up and then transited through an activation function. The activation function decides whether a specific neuron becomes activated or not. If a neuron is activated, its information is transferred to other layers until it reaches the output layer. Overall, this whole work is summed up [26] in Figure 3.

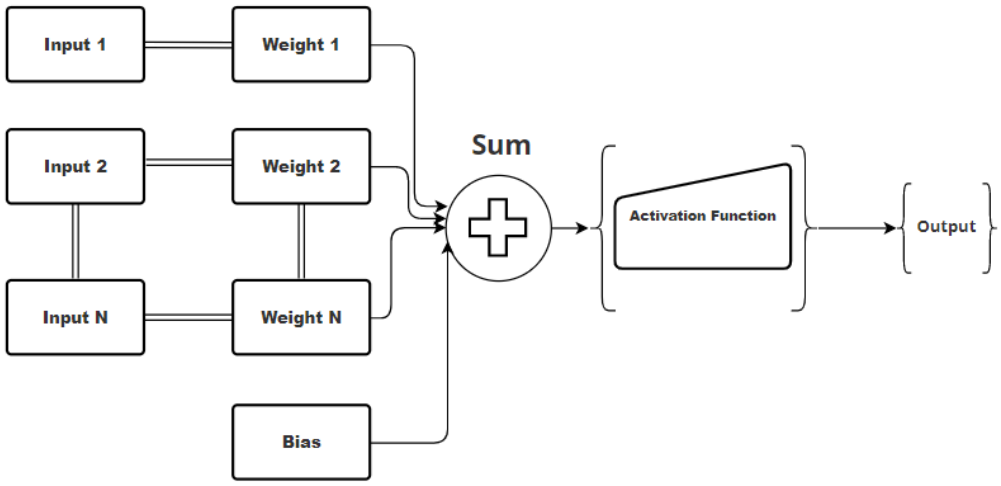

Figure 3. Working principle of artificial neuron.

5. Simulation Results

The nonlinear kinematics, Equations (2)–(4), of a TWR, were modeled in Simulink, and a dataset of 673 samples was acquired. This whole dataset is made up of samples taken under 12 different conditions with varying values of the linear velocity of the DC motor and radius. The sampling period was 0.25 s. Each dataset has different values of radius 'r' and linear velocity, thus having different output values. This radius 'r' converts the rotation of the high-torque DC motor into linear velocity 'v' along 'x' and 'y' directions according to a well-known formula given by Equation (5).

$$V = r\omega \tag{5}$$

where 'ω' shows the angular velocity of the DC motor.

Both wheels of the TWR were operated by high-torque DC motors, and the rotation of both motors covered the linear displacement along the x- and y-axis. Angular rotation was measured along the z-axis, and displacement along the z-axis was zero.

In ANN training, inputs and outputs were fed separately after acquiring the data in the workspace. The neurons in the input and output layers are three, as for this system, there are three inputs and three outputs. The number of hidden neurons was heuristically selected as 100 to reach the desired goal of minimum MSE and maximum regression value. The network chosen was a feedforward network. The training algorithm chosen for this system was 'Levenberg–Marquardt' because it takes less time but more memory. In total, 470 samples were taken for training purposes, while 101 samples were taken for validation and testing. The performance of the training state is shown in Figure 4. It shows the response of the gradient curve, the Mu parameter (i.e., the control parameter for the algorithm), and how it directly affects the error convergence value. The gradient value is 0.19114 at an epoch of 1000, the value of the control parameter is 0.001 at an epoch of 1000, and validation fails are 0 at an epoch of 1000. The training was terminated at the 1000th epoch due to the maximum number of epochs being reached, i.e., a hyperparameter that was set in tuning, as shown in Figure 4.

The results of the MSE for training, testing, and validation are shown below in Table 1.

Table 1. Results of tuning ANN of TWR.

Results	Samples	Mean Square Error (MSE)	Regression (R)
Training	470	1.66340×10^{-3}	9.99999×10^{-1}
Validation	101	2.80631×10^{-3}	9.99999×10^{-1}
Testing	101	2.20462×10^{-3}	9.99999×10^{-1}

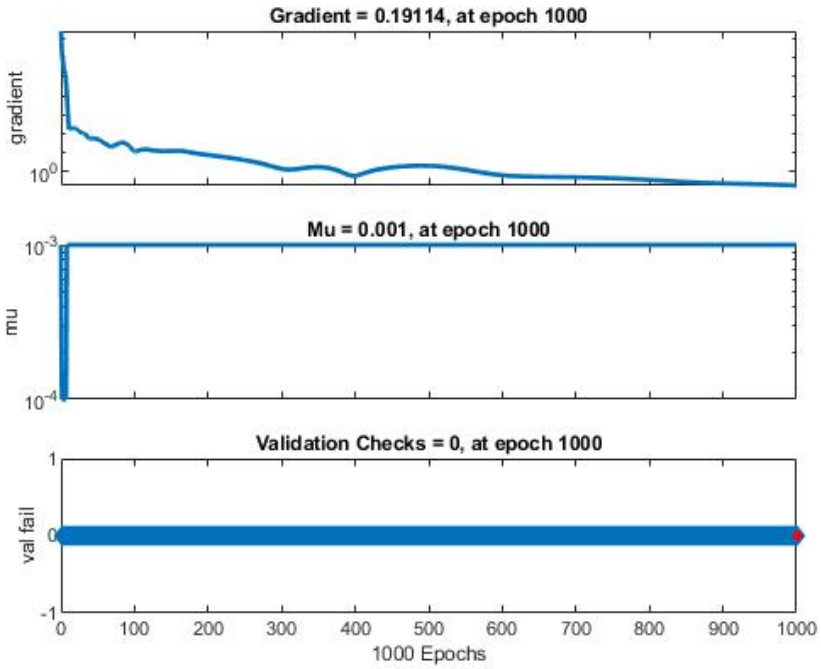

Figure 4. Training state performance.

According to the results given in Table 1, in all three phases, the minimum value of mean square error goes to 10^{-3}. The regression values extracted for this trained ANN show that the model is close to the ideal one, as its value is 0.99999. A plot of MSE is shown in Figure 5, which shows the best validation performance of an MSE of 0.028063 that was achieved at epoch 1000. This figure depicts how the MSE value varies.

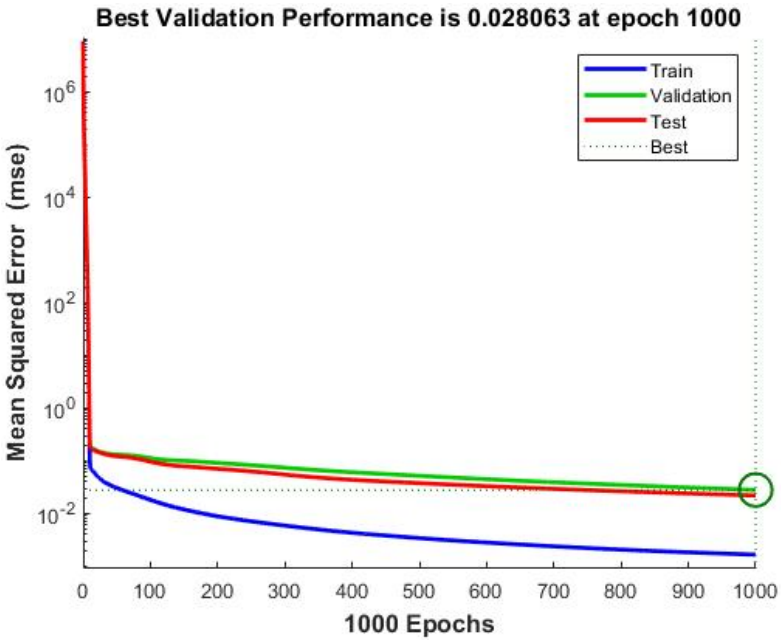

Figure 5. Best Validation Performance.

Figure 6, given below, was plotted from the the Neural Network toolbox in MATLAB, which shows how many errors in training, testing, and validation are close to the zero level, as shown in orange. Blue is for the training phase, green is for validation, and red is for the testing phase. These depict the minimum values of error levels. In Figure 6, the maximum error goes up to 0.0397, which is close to the zero-error level, and it cannot be achieved for a real-time database.

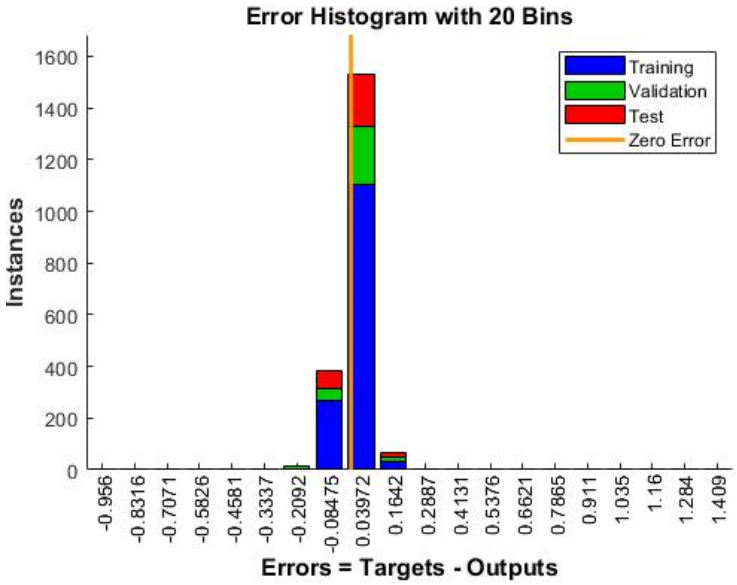

Figure 6. Error histogram for training, validation, and testing.

If we talk about the regression values, ideally, the value is '1', which is shown in the plots of regression given in Figure 7. The circles show the samples, and if the circles are on the center-fit line, it means that targeted values and outputs match. The closer a sample is to the fit line, the higher the regression value will be for that particular phase. All the responses of regression give the value of '1', and the case is similar for the overall regression analysis response for the combined phases.

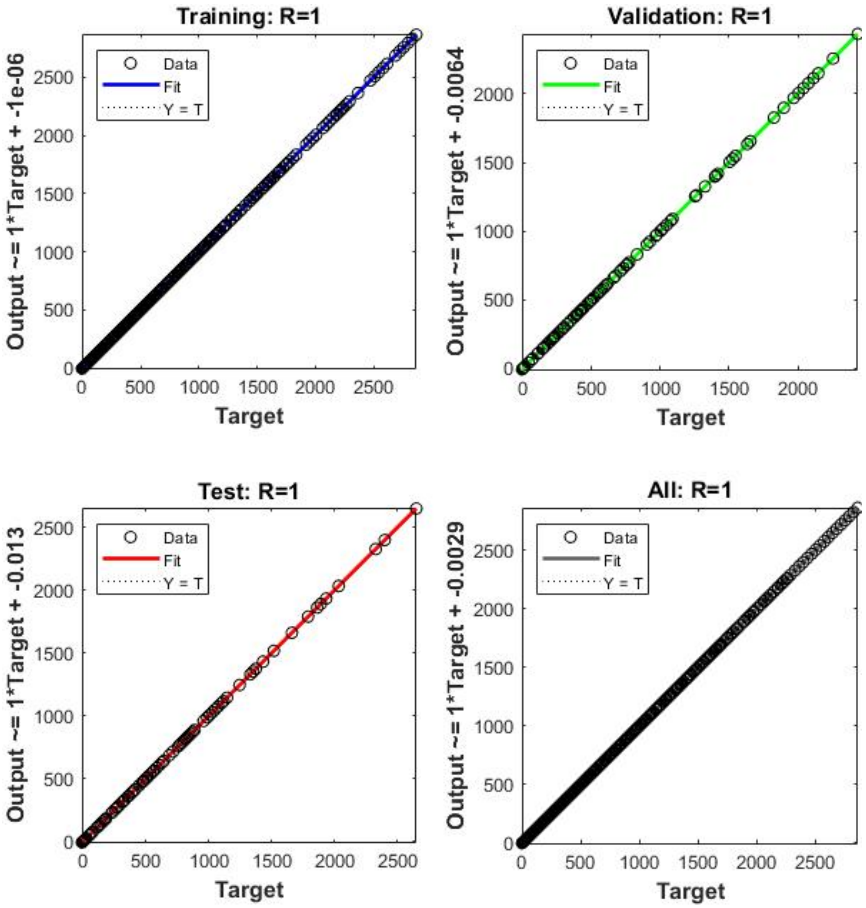

Figure 7. Regression analysis of all three phases and overall performance.

6. Conclusions

System identification of a TWR is presented in this paper using a feedforward ANN. From the Simulink environment, input/output datasets at various operating conditions are obtained from the simulated kinematics of the TWR. In a previous study, training an ANN in a feedforward network was achieved using a single operating condition when the values of V and r of the TWR were fixed. In this paper, 12 different datasets were acquired based on varying the values of all the inputs. Supervised learning was performed for the ANN as the labels of inputs and outputs were provided. The system was trained, and 30% of the dataset was kept for validation and testing purposes so our system could be verified in terms of MSE and regression parameters. The MSE and regression analysis discussed in the previous section show that the ANN fully captured the dynamics of all three outputs of the TWR up to a maximum level. The drawbacks of the previous methods are fully covered, and an almost ideal value of regression, i.e., 0.9999, was achieved in all three categories. Furthermore, a less complex and more efficient method was used to model the dynamics of the TWR, which is more appropriate than previous research work [13].

The adopted methodology for system identification exhibits an outstanding estimation of the TWR nonlinear kinematics. Comparative analyses between the validation datasets and the estimated datasets achieve the desired benchmark in terms of regression analysis, which is 1, and an MSE of 0.01. Furthermore, the accuracy in terms of MSE and regression value is far better than the one presented in [13]. Results prove that the ANN proposed in this paper gave the most optimal performance for training, the validation dataset, and testing datasets. The limitations of the paper are that a slightly higher number of neurons are used in the tuning phase, which can be reduced in the future to obtain enhanced performance in terms of regression value and MSE.

Author Contributions: Conceptualization, M.A.K. and D.-e.-Z.B.; methodology, M.A.K. and D.-e.-Z.B.; software, M.A.K. and B.A.; validation M.A.K., H.A., D.-e.-Z.B. and S.K.; formal analysis, M.A.K., D.-e.-Z.B., H.A., B.A. and S.K.; investigation, M.A.K. and H.A.; resources, H.A. and A.W.; data curation, M.A.K. and D.-e.-Z.B.; writing—original draft preparation, M.A.K.; writing—review and editing, M.A.K., D.-e.-Z.B., H.A., B.A., S.K., A.W. and T.K.; visualization, M.A.K., D.-e.-Z.B., H.A., B.A. and S.K.; supervision, M.A.K., D.-e.-Z.B. and H.A.; project administration, M.A.K., H.A. and D.-e.-Z.B., funding acquisition, A.W. and T.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: If any researcher wants to obtain the dataset generated during this research, he/she can acquire it by sending an email to the first author, and it will be made available on request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Padois, V. Control and Design of Robots with Tasks and Constraints in Mind. Robotics [cs.RO]. Ph.D. Thesis, Université Pierre et Marie Curie (Paris 6), Paris, France, 2016.
2. John, R.A.; Tiwari, N.; Patdillah, M.I.B.; Kulkarni, M.R.; Tiwari, N.; Basu, J.; Bose, S.K.; Ankit; Yu, C.J.; Nirmal, A.; et al. Self healable neuromorphic memtransistor elements for decentralized sensory signal processing in robotics. *Nat. Commun.* **2020**, *11*, 4030. [CrossRef] [PubMed]
3. Edwards, J. Signal Processing Supports Robotic Innovation. IEEEXplore. 2022. Available online: https://signalprocessingsociety.org/publications-resources/ieee-signal-processing-magazine/2022/03 (accessed on 22 October 2022).
4. Sun, Y. Design and Research on Distributed Control System of Humanoid Robot Based on Automation Technology. In Proceedings of the 2021 IEEE Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC), Dalian, China, 14–16 April 2021; pp. 1063–1066. [CrossRef]
5. Peltier, M.D. Trajectory Control of a Two-Wheeled Robot. Master's Thesis, University of Rhode Island, Kingston, RI, USA, 2012.
6. Goher, K.M. A two-wheeled machine with a handling mechanism in two different directions. *Robot. Biomim.* **2016**, *3*, 17. [CrossRef]
7. Chan, R.P.M.; Stol, K.A.; Halkyard, C.R. Review of modelling and control of two-wheeled robots. *Annu. Rev. Control* **2013**, *37*, 89–103, 1367–5788. [CrossRef]

8. Liu, Q.; Cong, Q. Kinematic and dynamic control model of wheeled mobile robot under internet of things and neural network. *J. Supercomput.* **2022**, *78*, 8678–8707. [CrossRef] [PubMed]
9. Morin, P.; Samson, C. Motion Control of Wheeled Mobile Robots. In *Springer Handbook of Robotics*; Siciliano, B., Khatib, O., Eds.; Springer: Berlin/Heidelberg, Germany, 2008. [CrossRef]
10. Baig, D.-E.-Z.; Su, H.; Cheng, T.M.; Savkin, A.V.; Su, S.W.; Celler, B.G. Modeling of human Heart Rate response during walking, cycling and rowing. In Proceedings of the 2010 Annual International Conference of the IEEE Engineering in Medicine and Biology Conference, IEEE Engineering in Medicine and Biology Society, Buenos Aires, Argentina, 31 August–4 September 2010; pp. 2553–2556. [CrossRef]
11. Baig, D.-E.-Z.; Savkin, A.; Celler, B.G. Estimation of oxygen consumption during cycling and rowing. In Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, IEEE Engineering in Medicine and Biology Society, San Diego, CA, USA, 28 August–1 September 2012; pp. 711–714. [CrossRef]
12. Cheng, T.M.; Savkin, A.V.; Celler, B.G.; Su, S.W.; Wang, L. Nonlinear modeling and control of human heart rate response during exercise with various work load intensities. *IEEE Trans. Biomed. Eng.* **2008**, *55*, 2499–2508. [CrossRef] [PubMed]
13. Uddin, N. System Identification of Two-Wheeled Robot Dynamics Using Neural Networks. *J. Phys. Conf. Ser.* **2020**, *1577*, 012034. [CrossRef]
14. Shahraki, M.; Aliyari, S.M.; Mousavinia, A. Two wheel self-balanced mobile robot identification based on experimental data. In Proceedings of the 2017 Iranian Conference on Electrical Engineering (ICEE), Tehran, Iran, 2–4 May 2017; pp. 883–888. [CrossRef]
15. Chang, C.L.; Liou, K.H. Reinforcement Learning-Based Two-Wheel Robot Control. In *Recent Advances in Intelligent Information Hiding and Multimedia Signal Processing. IIH-MSP 2018. Smart Innovation, Systems and Technologies*; Pan, J.S., Ito, A., Tsai, P.W., Jain, L., Eds.; Springer: Cham, Switzerland, 2019; Volume 110. [CrossRef]
16. Ruan, X.; Cai, J.; Chen, J. Learning to Control Two-Wheeled Self-Balancing Robot Using Reinforcement Learning Rules and Fuzzy Neural Networks. In Proceedings of the 2008 Fourth International Conference on Natural Computation, Jinan, China, 18–20 October 2008; pp. 395–398. [CrossRef]
17. Yan, J. Hierarchical Reinforcement Learning Based Self-balancing Algorithm for Two-wheeled Robots. *Open Electr. Electron. Eng. J.* **2016**, *10*, 69–79. [CrossRef]
18. Zheng, Q.; Wang, D.; Chen, Z.; Sun, Y.; Liang, B. Continuous reinforcement learning based ramp jump control for single-track two-wheeled robots. *Trans. Inst. Meas. Control* **2021**, *44*, 892–904. [CrossRef]
19. Bozek, P.; Karavaev, Y.L.; Ardentov, A.A.; Yefremov, K.S. Neural network control of a wheeled mobile robot based on optimal trajectories. *Int. J. Adv. Robot. Syst.* **2020**, *17*, 1729881420916077. [CrossRef]
20. Maity, A.; Bhargava, S. Design and Implementation of a Self-Balancing Two-Wheeled Robot Driven by a Feed-Forward Backpropagation Neural Network. *Int. Res. J. Eng. Technol.* **2020**, *7*, 3876–3881.
21. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*; MIT Press: Cambridge, MA, USA, 1998.
22. Khan, M.A.; Baig, D.-E.-Z.; Ashraf, B.; Ali, H.; Rashid, J.; Kim, J. Dynamic Modeling of a Nonlinear Two-Wheeled Robot Using Data-Driven Approach. *Processes* **2022**, *10*, 524. [CrossRef]
23. Abraham, T.H. (Physio)logical circuits: The intellectual origins of the McCulloch-Pitts neural networks. *J. Hist. Behav. Sci.* **2002**, *38*, 3–25. [CrossRef] [PubMed]
24. Rosenblatt, F. *Principles of Neurodynamics. Perceptrons and the Theory of Brain Mechanisms*; Cornell Aeronautical Lab Inc: Buffalo, NY, USA, 1961.
25. Andrej, K.; Janez, B.; Andrej, K. Chapter 1. Introduction to the artificial neural networks. In *Artificial Neural Networks*; Suzuki, K., Ed.; IntechOpen: Rijeka, Croatia, 2011.
26. Decaro, C.; Montanari, G.; Molinariz, R.; Gilberti, A.; Bagnoli, D.; Bianconi, M.; Bellanca, G. Machine Learning Approach for Prediction of Hematic Parameters in Hemodialysis Patients. *IEEE J. Transl. Eng. Health Med.* **2019**, *7*, 4100308. [CrossRef] [PubMed]