logistics

Article

NARX Neural Network for Safe Human–Robot Collaboration Using Only Joint Position Sensor

Abdel-Nasser Sharkawy [1,2,*] and Mustafa M. Ali [1]

[1] Mechatronics Engineering, Mechanical Engineering Department, Faculty of Engineering, South Valley University, Qena 83523, Egypt
[2] Mechanical Engineering Department, College of Engineering, Fahad Bin Sultan University, Tabuk 47721, Saudi Arabia
* Correspondence: abdelnassersharkawy@eng.svu.edu.eg or eng.abdelnassersharkawy@gmail.com

Abstract: *Background*: Safety is the very necessary issue that must be considered during human-robot collaboration in the same workspace or area. *Methods*: In this manuscript, a nonlinear autoregressive model with an exog-enous inputs neural network (NARXNN) is developed for the detection of collisions between a manipulator and human. The design of the NARXNN depends on the dynamics of the manipulator's joints and considers only the signals of the position sensors that are intrinsic to the manipulator's joints. Therefore, this network could be applied and used with any conventional robot. The data used for training the designed NARXNN are generated by two experiments considering the sinusoidal joint motion of the manipulator. The first experiment is executed using a free-of-contact motion, whereas in the second experiment, random collisions by human hands are performed with the robot. The training process of the NARXNN is carried out using the Levenberg–Marquardt algorithm in MATLAB. The evaluation and the effectiveness (%) of the developed method are investigated taking into account different data and conditions from the used data for training. The experiments are executed using the KUKA LWR IV manipulator. *Results*: The results prove that the trained method is efficient in estimating the external joint torque and in correctly detecting the collisions. *Conclusions*: Eventually, a comparison is presented between the proposed NARXNN and the other NN architectures presented in our previous work.

Keywords: human–manipulator collaboration; collision detection; NARX neural network; joint position sensor; training and testing; evaluation and effectiveness

check for updates

Citation: Sharkawy, A.-N.; Ali, M.M. NARX Neural Network for Safe Human–Robot Collaboration Using Only Joint Position Sensor. *Logistics* **2022**, *6*, 75. https://doi.org/10.3390/logistics6040075

Academic Editors: Ieva Meidutė-Kavaliauskienė, Şemsettin Çiğdem and Renata Činčikaitė

Received: 27 June 2022
Accepted: 12 October 2022
Published: 18 October 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

During the collaboration between a human and manipulator, safety is the most relevant issue that needs to be considered in the procedural design. This is because when the human is close to the manipulator, injuries are expected to happen. Therefore, a safety system must exist for the robotic manipulator. The main challenge is to implement a cheap system, and at the same time, it should be easily adapted to the robot and have high effectiveness. Safety is also taken into accounts in the control system design of human–robot collaboration (HRC).

Researchers have put forth many efforts to develop a system for HRC safety, whether as a collision avoidance method or as a collision detection method. The classifications of these methods are presented in Figure 1.

Collision avoidance methods depend on the depth sensors and vision system, as presented in Refs. [1–6]. A proximity sensor-based approach was also proposed by Lam et al. in [7]. In this approach, five sensors of contactless and capacitive types and implemented antennas were used. These types of methods need modifications in the robot's body, and the cost of the sensors is high.

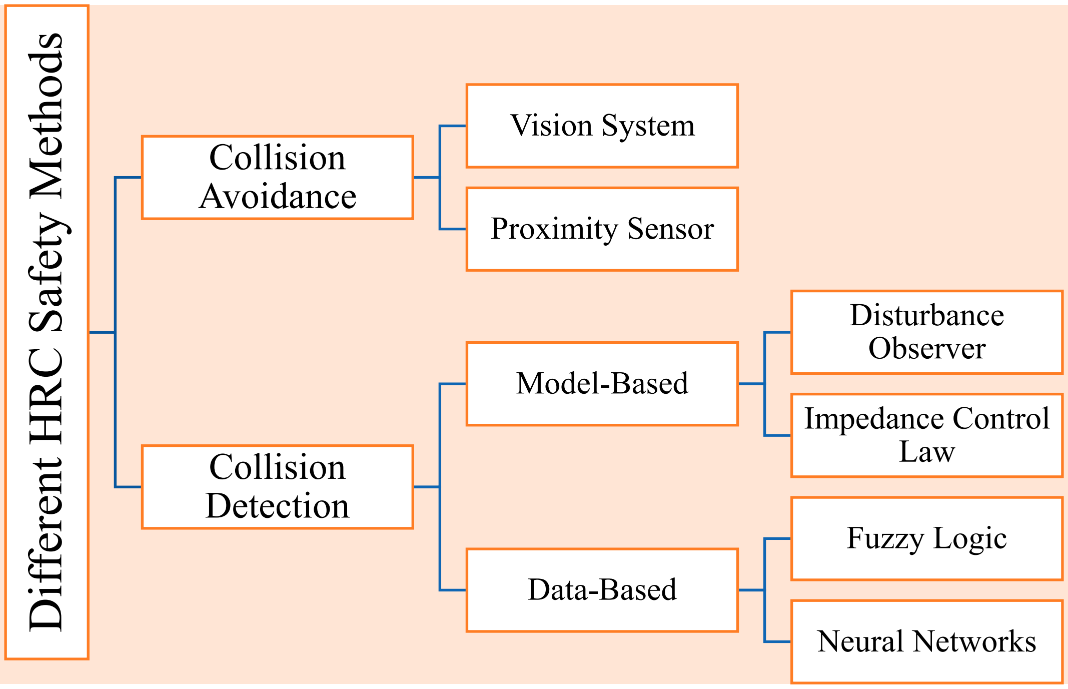


Figure 1. The classification of the safety methods of HRC.

Collision detection methods are classified into two types: model-based and data-based. Model-based methods depend mainly on the dynamic parameters of the robot's model, whereas the data-based methods are based on data. The proposed approach in the current paper is a data-based method; therefore, most of our concentration focuses on these types of methods. Model-based methods are classified into two approaches: disturbance observer and impedance control law-based methods. In [8,9], the disturbance observer was presented. The approach depends on the signals of the torque sensors of the manipulator's joints. Thus, this method could be applied and uses only collaborative robotic manipulators that contain joint torque sensors. Morinaga and Kosuge in [10] developed a collision-detection method depending on the impedance controller. Their approach used the torque signal (the error between the actual and the desired input torques of the robotic manipulator). The problems with the methods based on the robot's model are that these methods are inaccurate and uncertain because there is uncertainty in the dynamic parameters determination [11–14]. Therefore, this negatively influences the effectiveness of the method to detect the collisions, and the fault alarms are increased [11]. In addition, the calculations of the inverse dynamics are very complex [15].

Data-based methods were considered for collision detection between a human and a manipulator. These methods depend on fuzzy logic and neural networks. In [16], Dimeas et al. used the fuzzy-logic-based method for collision detection. Their system was designed depending on the signals of both the joint position and the torque sensors. Their method was applied to a 2-DOF planar robot, and it was efficient in detecting the collisions. However, it can only be applied and used with the collaborative robots that contain a joint torque sensor. Furthermore, they used one fuzzy system for each joint, which means that they needed seven fuzzy systems if they applied their method to a 7-DOF robot. Therefore, this is very complex and needs more time and effort. In [14], S. Lu and his group used the NN-based approach for the detection of collisions. Their developed system was designed based on the history of the joint angles of the manipulator and the two external sensors (the base, and the wrist force/the torque sensors). Their method was investigated using 1-DOF and 2-DOF robotic manipulators. The effectiveness in the calculations of their method was presented by executing only one collision by the manipulator, but it was not investigated by using many random collisions. Kim et al., in [17], proposed a versatile modularized NN to detect collisions with a collaborative manipulator. The momentum observer torques were required for their method. In [18], four NN architectures were proposed: the auto regressive NN, the recurrent NN (RNN), the convolutional long short-term memory NN (CLSTMNN) and the mixed CLSTMNN. These architectures were designed based on position, velocity, and motor current.

In our previous work presented in [19–24], four NN architectures were designed and trained for the collision detection. These architectures were (1) multilayer feedforward NN having one hidden layer (MLFFNN-1), (2) multilayer feedforward NN having two hidden layers (MLFFNN-2), (3) the cascaded forward NN (CFNN), and (4) the recurrent NN (RNN). MLFFNN-1 was developed using both the position and torque sensors that are intrinsic to the manipulator' joints. Thus, the application of this structure is only with the collaborative robotic manipulators that possess joint torque sensor. The experimental investigation of this structure was applied to the 1-DOF, 2-DOF planar, and 3-DOF robotic manipulators. The other three structures (MLFFNN-2, CFNN, and RNN) were developed using the position sensor only of the manipulator's joint. Thus, the applications of these structures are with any conventional robot. MLFFNN-2 was investigated experimentally using 1-DOF and 2-DOF planar robotic manipulators, whereas CFNN and RNN were investigated experimentally using the 1-DOF manipulator. The effectiveness and the performance measure of these architecture were presented in different conditions and used different data from the training ones.

From this discussion, a data-based method is a desirable solution for the detection of robot collisions with humans. This type of method should avoid the explicit knowledge of the dynamic parameters of the robot model and the complex calculations of the inverse dynamics. Furthermore, it should be applied to any conventional robot, and its effectiveness (%) must be presented with the use of many different conditions and cases.

The work of the current paper is considered as an extension to our previous work presented in [19–24]. The main contribution and novelty of the current paper is discussed as follows. A NARXNN is developed to detect human–manipulator collisions. The NARXNN is designed depending only on the signals of the position sensors that are intrinsic to the manipulator's joints. This qualifies the proposed method to be used with any conventional robot. The training of the designed NARXNN is executed using data with and without collisions and using Levenberg–Marquardt learning. The knowledge of the dynamic parameters of the robot model such as the inertia forces, the Coriolis and centrifugal forces, and the gravitational forces, is not needed in the proposed method, and these parameters are not available in most of the current robots. The test and the verification of the trained NARXNN are carried out using the same data that are used for the training. Furthermore, its effectiveness is investigated using different data and conditions from the training ones, considering many random collisions with the manipulator. The experiments are performed using KUKA LWR IV manipulator considering only one joint's motion. However, the method can be applied with all joints of the robotic manipulator. The obtained results reveal that the trained NARXNN is an efficient method in estimating and detecting the collisions. A comparison is presented between the developed NARXNN and other previous NN architectures discussed in [19,20]. In addition, a comparison with other previously published methods is executed.

The rest of this manuscript is outlined as follows. Section 2 discusses the dynamics of the manipulator joints, why the NN is used, and the followed methodology/steps in the current work to detect the collision. Section 3 shows the design of the proposed NARXNN, and Section 4 illustrates the training and the testing of the designed NARXNN using the same data. Section 5 discusses the trained NARXNN's evaluation considering many random collisions with the manipulator and using different data and conditions from the training ones. In Section 6, a comparison between the NARXNN and other NN architectures used for detecting the collisions is presented. Finally, Section 7 provides the main points that are discussed in this manuscript with some future work.

2. Dynamics of Manipulator Joints

The external torque of the manipulator's joint τ_{ext} is determined from the dynamics equation of the manipulator as follows [25,26]:

$$\tau_{ext} = M(\theta)\,\ddot{\theta} + C\left(\theta, \dot{\theta}\right)\,\dot{\theta} + G(\theta) - \tau \tag{1}$$

The symbols used in this equation are defined in Table 1.

Table 1. The meaning of the symbols used in the dynamic Equation (1).

Symbol	Meaning
$\boldsymbol{\theta} \in \mathbb{R}^{n}$	A vector representing the positions of the manipulator joints
$\dot{\boldsymbol{\theta}} \in \mathbb{R}^{n}$	A vector representing the velocities of the manipulator joints
$\ddot{\boldsymbol{\theta}} \in \mathbb{R}^{n}$	A vector representing the accelerations of the manipulator joints
$\boldsymbol{M}(\boldsymbol{\theta}) \in \mathbb{R}^{n \times n}$	The inertia matrix of the manipulator
$\boldsymbol{C}\left(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}\right) \in \mathbb{R}^{n \times n}$	The Coriolis and centrifugal matrix of the manipulator
$\boldsymbol{G}(\boldsymbol{\theta}) \in \mathbb{R}^{n}$	The gravity vector of the manipulator
$\boldsymbol{\tau} \in \mathbb{R}^{n}$	The actuator torque of the manipulator joints
$\boldsymbol{n}$	The number of the links or the joints of the manipulator

As discussed in the introduction part, the parameters of this equation are not known in most of the current robots. This leads to difficulty in determining the external joint torque [15]. Furthermore, the techniques depending on the dynamic robot model are uncertain and inaccurate [11–14]. Therefore, this negatively influences the effectiveness for the detection of the collisions that occur between the human and manipulator.

According to this, a method depending on the NN's structure is developed to save human–manipulator collaboration. The NN is efficient in approximating any function, whether linear or nonlinear [27–29]. In addition, it can be generalized under different conditions. The NN-based approach solves the inverse dynamics problems of the manipulator efficiently [30–32]. From the different NN architectures, NARXNN is proposed. NARXNN is the dynamic and recurrent NN, and it possesses the connections of feedback enclosing several network layers [33,34]. The structure of this NN is the nonlinear generalization of the well-known ARX models. This NN predicts the time series very well [35,36] and is widely used with the nonlinear dynamic systems [37,38].

In the current work, this NARXNN is trained using the Levenberg–Marquardt (LM) algorithm. The LM algorithm [39,40] is a second-order optimization method that has the advantage of fast convergence. In addition, it approximates the method of Newton. The LM algorithm can achieve a fast learning speed of the method of classical Newton and at the same time the convergence of the gradient descent [39,41]. Compared with other training algorithms, LM converges with less repetitions/iterations and in a shorter time. However, this algorithm needs larger data. For solving this issue, enough data are used in the current work.

The followed steps and methodology with the proposed NARXNN for the detection of the robot collisions with humans are presented in Figure 2. All these steps are discussed in detail in the next sections.

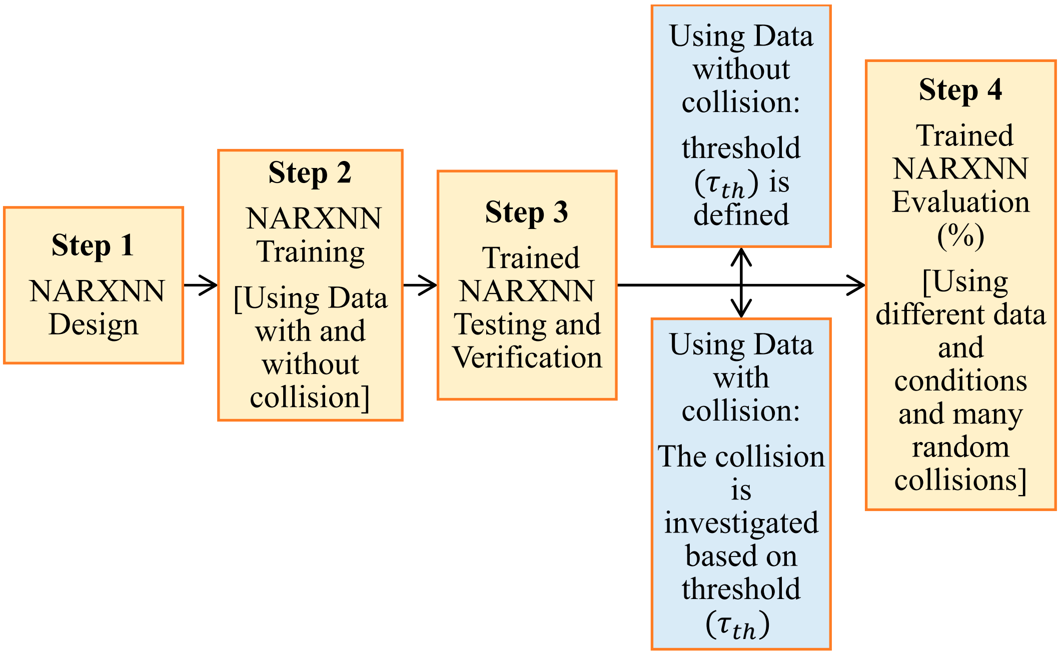

Step 1
NARXNN Design
Step 2
NARXNN Training
[Using Data with and without collision]
Step 3
Trained NARXNN Testing and Verification
Using Data without collision:
threshold (τ_{th}) is defined
Using Data with collision:
The collision is investigated based on threshold (τ_{th})
Step 4
Trained NARXNN Evaluation (%)
[Using different data and conditions and many random collisions]

Figure 2. The followed steps and methodology with the proposed NARXNN.

3. NARXNN Design

In this section, the design of the NARXNN for the detection of the robot collisions with humans is presented in detail.

The same sinusoidal motion presented in the previous work [19,20] is used here in the current work. This motion is executed using frequencies that are variable and are commanded to the E1′s joint of the KUKA LWR IV manipulator, as shown in Figure 3. Two experiments are considered. In the first experiment, the motion of the robot joint is free of contact, whereas in the second experiment, the human hand performs some random collisions with the robot during its motion.

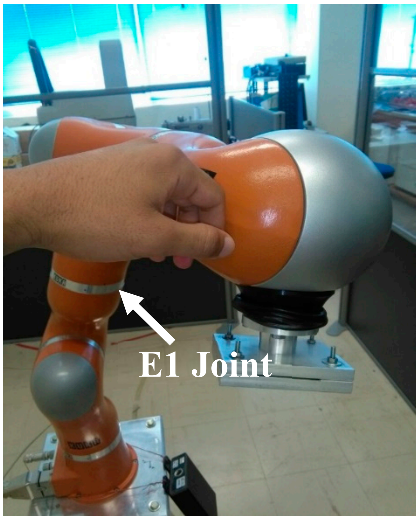
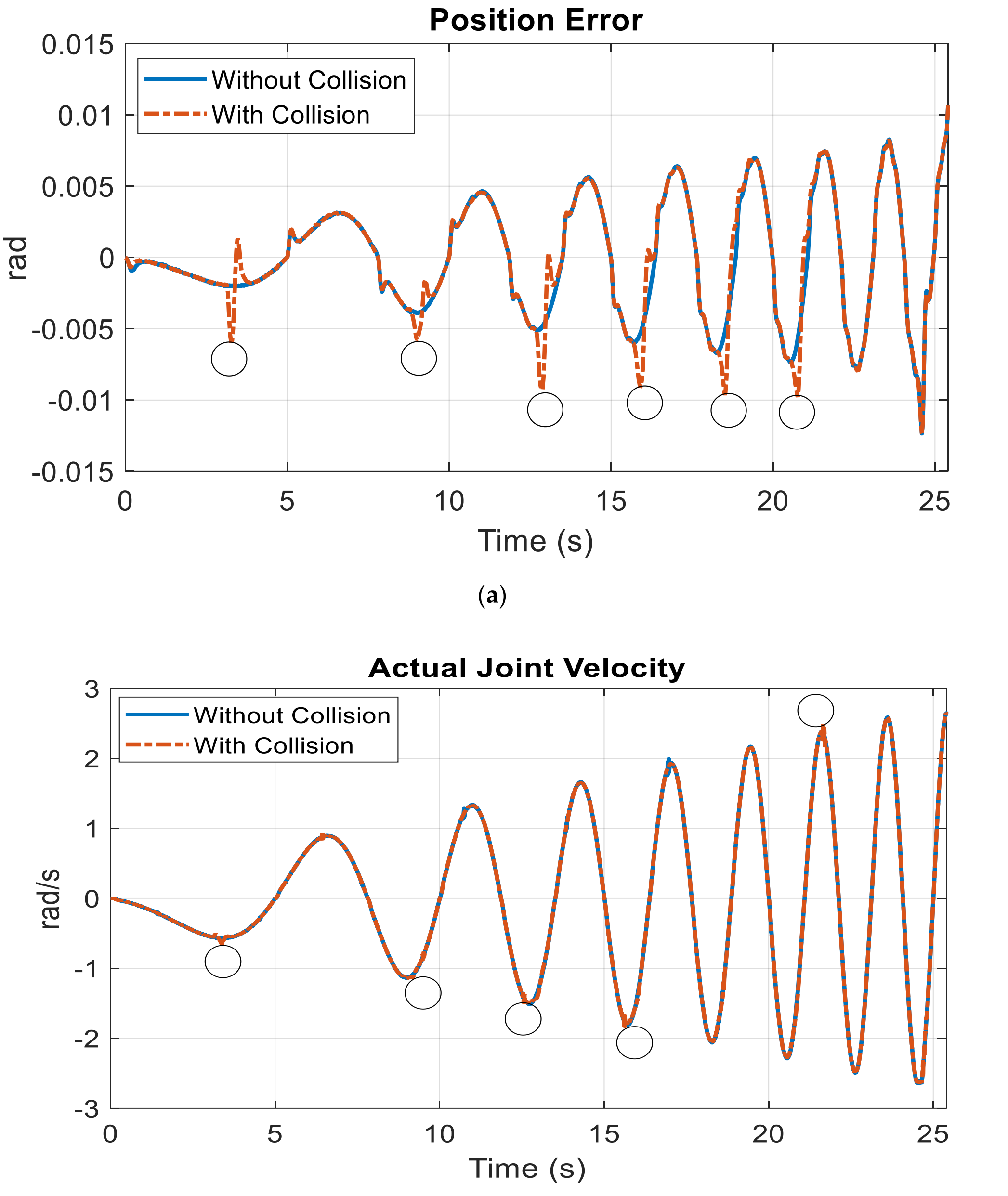


Figure 3. The setup for experiments with the KUKA LWR IV robot. This figure is taken from our previous work presented in [19,20].

The NARXNN is designed depending only on the positions' sensors that are intrinsic to the manipulator's joint. Therefore, the application and the use of this method are for any conventional robot. The main criteria to select the inputs of the NARXNN to achieve the high performance of the NN are (1) the smallest mean squared error (MSE), and therefore (2) the smallest training error. There are three inputs, and they are defined as follows:

(1) The current position error of the joint $\widetilde{\theta}(k)$, presenting the difference between the desired and the actual value of the joint's position;
(2) The previous position error of the joint $\widetilde{\theta}(k-1)$;
(3) The actual velocity of the joint $\dot{\theta}_{ac}(k)$.

The actual velocity of the joint is obtained from the KUKA robot controller (KRC). However, this velocity could be determined via the numerical differentiations of the joint's position. The inputs for the case of a collision and for the in case of free-of-contact motion are shown in Figure 4.



(**a**)

(**b**)

Figure 4. The selected inputs of the designed NARXNN. (**a**) The current error of the joint's position. (**b**) The actual velocity of the joint.

In Figure 4, the red color curves represent the inputs in the case there is a collision, whereas the blue color curves represent the inputs in the case there is no collision. The spikes marked by the small blank circle represent the collisions. More details about these inputs are found in our previous work in Ref. [19].

The NARXNN architecture includes three types of layers as in the following order: the layer of input, the layer of hidden neurons that is nonlinear, where its activation function is the hyperbolic tangent (tanh), and the layer of output that is also nonlinear, where its activation function is the hyperbolic tangent (tanh). The output layer estimates the external joint torque τ'_{ext}. The input delay vector is [0 1], and the output delay vector is [1 2]. This proposed architecture is presented in Figure 5. The equations representing the feedforward part of the NARXNN are given as follows.

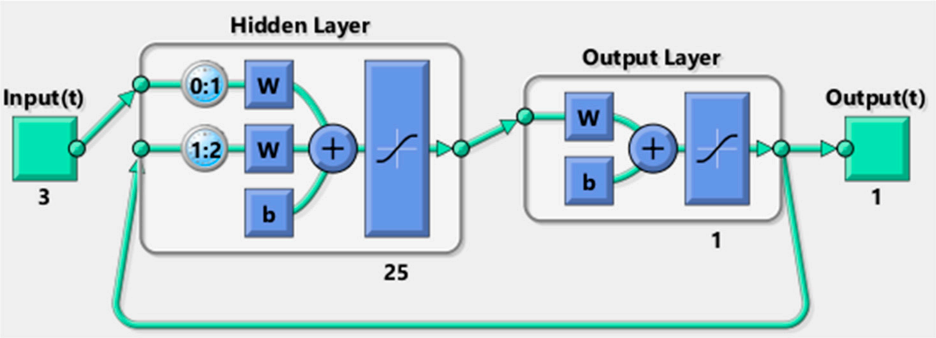


Figure 5. The developed NARXNN structure. The inputs are the current error of the position, the previous error of the position, and the actual velocity of the joint. The output is the estimated external joint's torque.

The output of the hidden neuron j that is in the hidden layer is given by

$$y_j = \varphi_j(h_j) = \varphi_j\left(\left(\sum_{i=0}^{3} w_{ji}x_i\right) + (c_{j1}\tau'_{ext}(k-1)\right) \tag{2}$$

where x_i represents the inputs to the NN. $x_0 = 1$, $x_1 = \widetilde{\theta}(k)$, $x_2 = \widetilde{\theta}(k-1)$, and $x_3 = \dot{\theta}_{ac}(k)$. $\tau'_{ext}(k-1)$ is the previous value of the estimated external joint of NARXNN. w_{ji} is the weight between the input i and the hidden neuron j, and c_{j1} is the weight between the input $\tau'_{ext}(k-1)$ and the hidden neuron j.

The activation function of the hidden layer is given by

$$\varphi_j(h_j) = \tanh(h_j) \tag{3}$$

The estimated external torque by the NARXNN, τ'_{ext}, is calculated by

$$\tau'_{ext} = \psi_k(O) = \psi_k\left(\sum_{j=0}^{n} b_{1j}y_j\right) = \tanh\left(\sum_{j=0}^{n} b_{1j}y_j\right) \tag{4}$$

where b_{1j} is the weight between the hidden neuron j and the estimated output of the NARXNN.

The external joint's torque τ_{ext} is given/provided by KRC and is used only for training the designed NARXNN architecture. In any other type of robot where τ_{ext} is not given/existing, the external collision's force can be measured by any type of external force sensor. The error of training $e(t)$ must be very small. This error is calculated from the following equation:

$$e(t) = \tau_{ext} - \tau'_{ext} \tag{5}$$

The training process of the designed NARXNN architecture is discussed in detail in the next section.

4. NARXNN Training Then Testing

In this section, the training process of the designed NARXNN is presented. In addition, the test of this trained NN structure is investigated. The following steps and methodology in these processes are summarized as follows:

(1) Collect the data with and without collisions from the experiments with the KUKA LWR robot.
(2) Initialize the parameters of the NARXNN and select the suitable number of hidden neurons.
(3) Train the designed NARXNN, using the collected data with and without collisions.
(4) After the training is completed, check the performance of the NARXNN by investigating the resulting MSE.
(5) If the resulting MSE is a high value and is not satisfactory, go again to step 2.
(6) If the resulting MSE is very small and close to zero (satisfactory), perform the following:
 - ➢ 6.1 Test the trained NARXNN by using the data without collisions that is used for training and check the training/approximation error.
 - ➢ 6.2 If this training/approximation error is a low value and is satisfactory, calculate the collision threshold and then go to step 7.
 - ➢ 6.3 If this training/approximation error is a high value and is not satisfactory, go again to step 2.

(7) Test the trained NARXNN using the data with collision that is used for the training process and check the collisions using the determined collision threshold.
(8) Check the effectiveness (%) of the trained NARXNN by performing many random different collisions with the robot based on the determined collision threshold.

All these steps are discussed and presented in detail in this section and the next sections.

The collected data from the executed experiments of the KUKA LWR manipulator in the first case where the motion has no collisions and in the second case where the collisions exist are used for training the designed NARXNN. The training occurs using the LM algorithm in MATLAB and using Intel(R) Core(TM) i5-8250U CPU @ 1.60 GHz processor. The number of pairs of input–output that are collected and used for the training process are 56358. These data are divided randomly into three parts: the first part comprising 80% is for training, the second part comprising 10% is used to make the validation, and the last part also comprising 10% is used to make the testing process. After investigating many different initializations of weights and many different numbers of the hidden neurons for finding the best parameters that achieve the high performance of the NARXNN, we obtained the following:

(1) The hidden neuron number was 25.
(2) The iteration/repetition number was 1000.
(3) The lowest MSE was 0.34353. The equation used for calculating this MSE is given as follows:

$$\text{MSE} = \sum\nolimits_{i=1}^{n} \frac{\left(\tau_{ext}(i) - \tau'_{ext}(i)\right)^2}{n} \tag{6}$$

(4) The training time was 34 min and 28 s. This time had no importance because the important aim is to obtain a very well trained NARXNN that can detect and identify the robot's collisions with humans efficiently. In addition, the training occurs offline.

The resulting MSE from the training process of the designed NARXNN is presented in Figure 6.

As shown in Figure 6, the resulting MSE is low and settles at the value of 0.34353. This proves that the designed NARXNN is trained well and is qualified for estimating the external joint's torque.

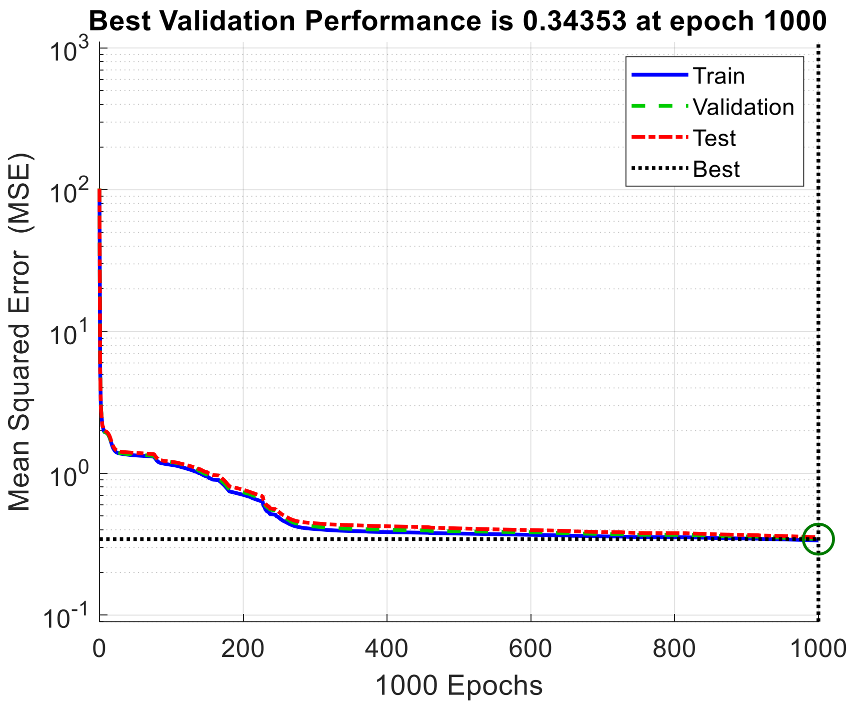


Figure 6. The resulting MSE through the training process of the designed NARXNN architecture.

After the process of training is finished in a complete way, this trained NARXNN is tested and verified using the same data that were used for the training. In the first step, using the data without collision, the approximating error that is the difference between the estimated external joint torque τ'_{ext} and the provided one τ_{ext} is calculated and presented in Figure 7.

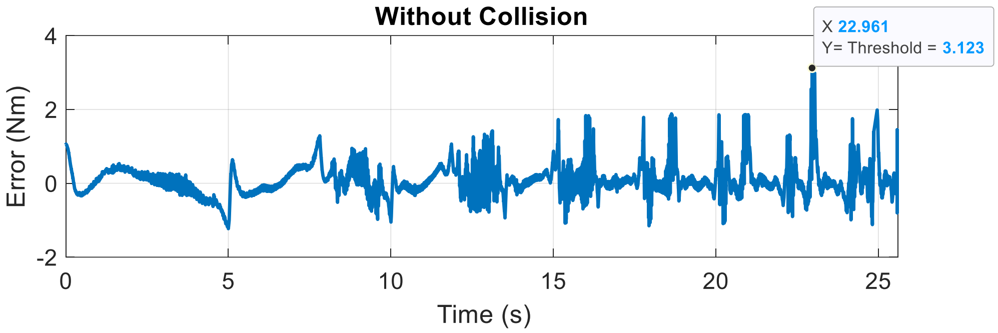


Figure 7. The approximating error that is calculated by subtracting the estimated external joint's torque by NARXNN from the provided one by KRC, in case there is no collision.

As shown in Figure 7, the approximating error in the case that there is no contact or collision is low. The average/mean of the absolute value of this error is 0.2759 Nm, and the standard deviation is 0.3342. The collision threshold is represented as the maximum of the absolute value of this error, as defined in our previous work in [19–24]. Therefore, the collision threshold τ_{th} is equal to 3.123 Nm. The collision is detected once the external joint torque exceeds the collision threshold $(|\tau'_{ext}| > \tau_{th})$. The comparison between the defined collision's threshold and ISO standards is presented as follows. In ISO/TS (15066) [42], the worst case for the reasons of safety is equal to 65 N, which represents the maximum value of the permissible force affecting the human face, creating pain or a minor injury, in the case of contact in a quasi-static condition. In the case of configurations of the KUKA LWR IV robotic manipulator, which represents the worst case for the human operator, the maximum external joint torque τ_{max} according to this force is equal to 30.42 Nm. By comparing the collision threshold by the maximum external torque τ_{max}, the collision threshold is very low. This proves that the collaboration between the human and robot is in the region of safety, and the detection of the collision happens before causing pain or slight injurie. More discussions and illustrations about the collision threshold and how it is defined by other researchers are presented in our previous work [23,24,43].

In the second step, using the data with collision, the approximating error that is the difference between the estimated external joint's torque τ'_{ext} and the provided τ_{ext} is also calculated. This approximation error and the convergence between both torques are presented in Figures 8 and 9.

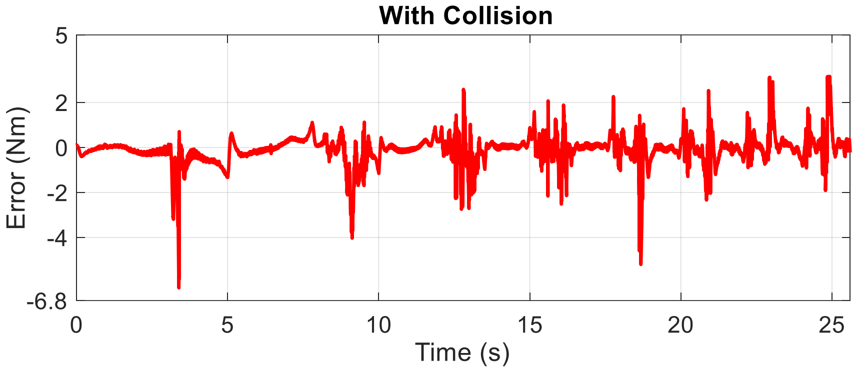


Figure 8. The approximating error calculated by subtracting the estimated external joint torque by NARXNN from the provided torque of KRC, in the case of a collision.

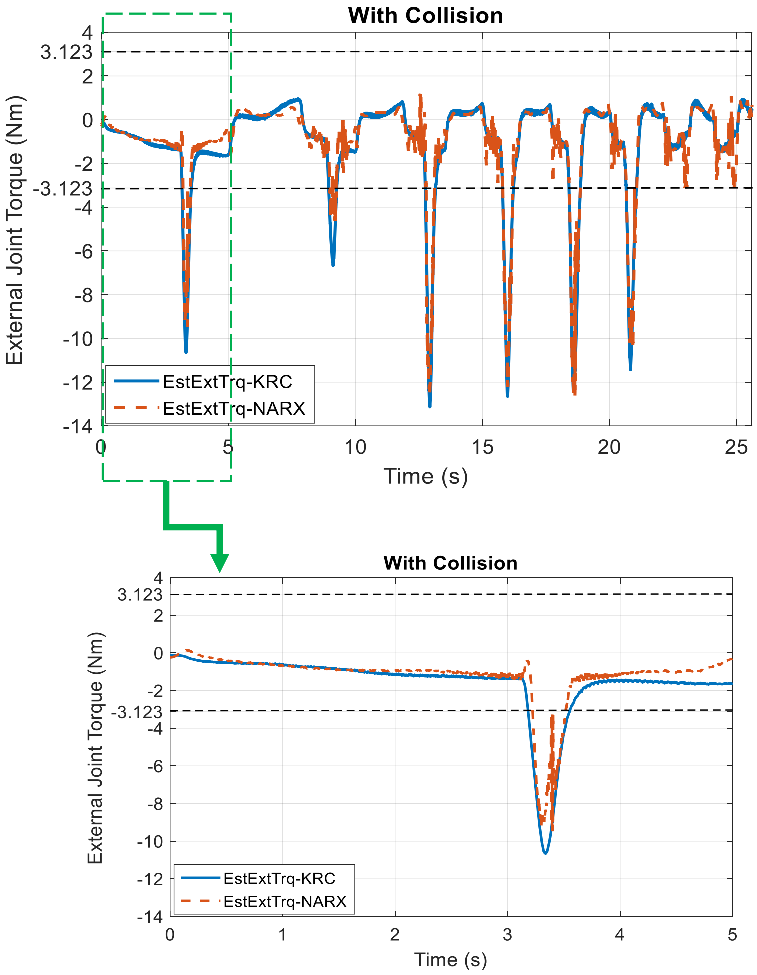


Figure 9. The estimated external joint torque of the NARXNN is compared by the provided torque by KRC, in the case of no collision.

As shown in Figures 8 and 9, the approximation between the estimated external torque and the provided one, in the case of collisions, is good. This illustrates that the developed NARXNN is well trained. The average absolute value of this approximating error is 0.3965 Nm, and the corresponding standard deviation (std.) is 0.5839. The approximation error in the case of collision is higher than the corresponding one in the case of no collision (Figure 7), which is logical. As shown in Figure 9, when a collision occurs, the estimated external joint torque exceeds the proposed threshold. This proves that the intended NARXNN is an efficient method for detecting the collision correctly.

The effectiveness (%) and the evaluation of the presented NARXNN using different data from the training data are presented in the next section.

5. NARXNN Evaluation and Effectiveness

In this section, the effectiveness (%) of the NARXNN is investigated using different data from the training data. During the time of carrying out this study, the real robot did not exist to evaluate the NARXNN online (during the real-time work with the robot). Therefore, the NARXNN is evaluated and investigated offline by the same data presented in our previous papers [19,20]. These data are real and are obtained from the experimental work with the KUKA robot considering a sinusoidal motion with different velocities. During this motion, 25 random robot collisions were achieved with human hands at different directions and magnitudes. These achieved collisions are true negatives, TN. The evaluation of the NARXNN is illustrated in Table 2 considering the following parameters:

(1) Correctly detected collisions are represented as true positives (TP).
(2) Actual collisions not detected by the NARXNN are represented as false negatives (FN).
(3) Alerts of collisions obtained by the trained NARXNN when no actual collision occurs are represented as false positives (FP).

Table 2. The effectiveness of the trained NARXNN considering the 25 executed random collisions.

The Trained NARXNN Effectiveness		
The Parameter	**The Number**	**The Percentage (%)**
TN	25	100
TP	22	88
FN	3	12
FP	1	4
The overall effectiveness $\left(\frac{\mathrm{TP}-\mathrm{FP}}{\mathrm{TN}} \times 100\right)$	84%	

From Table 2, the proposed trained NARXNN is an efficient method for the detection of the robot collision with the human. Its effectiveness is high (84%). Furthermore, the FP collision number is very low (4%), and this reveals that the NARXNN is robust. Its sensitivity to disturbances as well as unmodeled parameters is low.

The quantitative and qualitative comparison between the proposed trained NARXNN and the other previous NN architectures presented in our previous work [19,20] and other methods is discussed in the next section.

6. Quantitative and Qualitative Comparisons

In this section, a comparison between the proposed method and other previous published methods is presented.

Firstly, the comparison is presented between the proposed trained NARXNN and the previous NN architectures presented in our previous work [19,20]. These architectures are as follows:

- MLFFNN-1;
- MLFFNN-2;
- CFNN;
- RNN.

The comparison considers different parameters, as illustrated in Table 3. $\widetilde{\theta}(k)$ is the current error of position, $\widetilde{\theta}(k-1)$ is the first previous error of position, $\widetilde{\theta}(k-2)$ is the second previous error of position, $\dot{\theta}_{ac}(k)$ is the current actual velocity of joint, $\dot{\theta}_{ac}(k-1)$ is the previous actual velocity of the joint, and τ_{msr} is the measured joint torque.

Table 3. The comparison between the developed NN structures (MLFFNN-1, MLFFNN-2, RNN, CFNN, NARXNN) for human–manipulator collision detection.

Parameter	**NN's Structure**				
	MLFFNN-1	**MLFFNN-2**	**CFNN**	**RNN**	**NARXNN**
Layers	3	4	3	3	3
Main Inputs	$\widetilde{\theta}(k)$, $\widetilde{\theta}(k-1)$, $\dot{\theta}_{ac}(k)$, τ_{msr}	$\widetilde{\theta}(k)$, $\widetilde{\theta}(k-1)$, $\widetilde{\theta}(k-2)$, $\dot{\theta}_{ac}(k)$, $\dot{\theta}_{ac}(k-1)$	$\widetilde{\theta}(k)$, $\widetilde{\theta}(k-1)$, $\dot{\theta}_{ac}(k)$,	$\widetilde{\theta}(k)$, $\widetilde{\theta}(k-1)$, $\dot{\theta}_{ac}(k)$,	$\widetilde{\theta}(k)$, $\widetilde{\theta}(k-1)$, $\dot{\theta}_{ac}(k)$,
Hidden neurons	90	35 in the first hidden layer, and 35 in the second hidden layer	35	20	25
Epochs/ Repetitions	932	1000	952	906	1000
Smallest MSE	0.040644	0.21682	0.392	0.43078	0.34353
Training time	29 min and 47 s	1 h, 53 min, and 18 s	4 min and 24 s	4 h, 41 min, and 53 s	34 min and 28 s
Average or mean of absolute of approximation error—case of free of contact motion	0.0955 Nm	0.2362 Nm	0.2992 Nm	0.3061 Nm	0.2759 Nm
Average or mean of absolute of approximation error—case of collision	0.1398 Nm	0.2779 Nm	0.4365 Nm	0.4456 Nm	0.3965 Nm
Collision threshold	1.6815 Nm	2.7423 Nm	3.4520 Nm	3.7500 Nm	3.123 Nm
FP collisions	8%	4%	0%	0%	4%
FN collisions	16%	16%	16%	20%	12%
Overall effectiveness	76%	80%	84%	80%	84%
Application	This structure is used in robots with torque sensors.	The structures are used with any conventional robot.			

From Table 3, the MSE, approximating the error mentioned in Section 4, and the collision threshold of the NARXNN structure are higher in comparison with the corresponding ones by MLFFNN-1 and MLFFNN-2 architectures. However, the effectiveness of the NARXNN for the detection of collisions is higher in comparison with other architectures. The NARXNN, MLFFNN-2, CFNN, and RNN structures are used with any conventional robot, whereas the MLFFNN-1 is used only with collaborative robotic manipulators, where the torque sensors exist, including the estimation of the torque signals of the joints.

Second, the effectiveness of the proposed method is compared with the corresponding one of other previously related approaches such as the fuzzy system [16], the time series-based approach [16], the classifier based on NN [44], the classifier based on support vector machine (SVM) [45], and Lu et al.'s approach based on NN [14]. This comparison includes the effectiveness of the method (%) in the collision detection and the number of FP and FN collisions and is presented in Figure 10. As shown in Figure 10, the proposed NARXNN and the classifier based on SVM have high effectiveness (%) compared with other methods. The problem with classifications is the neglect in the external force amplitude (collision). In addition, the classifier cannot be used as a practical system for this force estimation. The false positive collisions are very low with the proposed NARXXNN, and the fuzzy system compared with others. The effectiveness and the performance measure (%) of Lu et al.'s approach [14] were missing/not included.

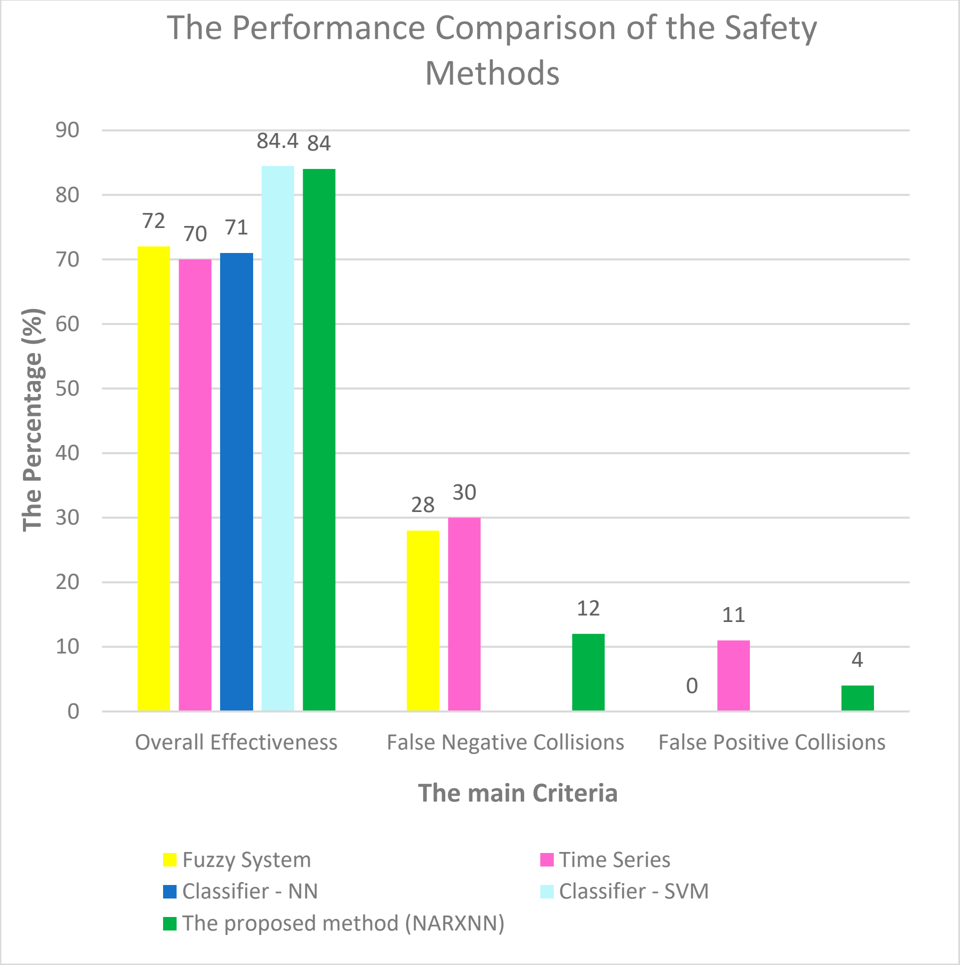


Figure 10. The comparison between the performance of the proposed method (NARXNN) and other previously related published methods.

7. Conclusions and Future Works

In this paper, a NARXNN was designed and trained to detect the collisions occurring between a human and the manipulator. For designing the proposed NARXNN, only the signals of the position sensors that are intrinsic to the manipulator's joints were considered. This qualifies the NN structure to be used with any conventional robot or manipulator. The training of the NARXNN was executed using data generated from the experimental work with the KUKA LWR robotic manipulator considering two cases: the first case involved joint motion free of collision, and the second case was joint motion of random robot collisions with the human. The test and the verification of the NARXNN structure was investigated by the same training data. In addition, the effectiveness, and the evaluation (%) of the NARXNN were investigated by the use of different data and conditions from the training conditions. The results from this methodology prove that the approximation error between the estimated collision and the actual one is low. Furthermore, the trained NARXNN is efficient for estimating and detecting the collision with high percentage (84%). The number of FP collisions was very low (4%).

A comparison was presented between the current NARXNN and the previous NN architectures such as (1) MLFFNN-1, (2) MLFFNN-2, (3) CFNN, and (4) RNN. From this comparison, we concluded that the approximating error of NARXNN was higher compared with MLFNN-1 and MLFFNN-2. However, the effectiveness of the proposed NARXNN was higher. MLFFNN-2, CFNN, RNN, and NARXNN can be applied to any robot, whereas MLFFNN-1 can only be used with collaborative robotic manipulators where the signals of the joint torque sensors are available. The proposed NARXNN was compared with other previously published methods, and the comparison revealed that the proposed method has a high effectiveness (%) in detecting the collisions.

The results obtained in the current work motivate us to perform future work to further investigate the presented NN structures by executing many random collisions (e.g., two hundred collisions), and then, the trained NN effectiveness can be determined. In addition, new methods that depend on deep and machine learning can be investigated for human–manipulator collision detection.

Author Contributions: Conceptualization, A.-N.S.; Formal Analysis, A.-N.S.; Investigation, A.-N.S.; Methodology, A.-N.S. and M.M.A.; Software, A.-N.S.; Validation, A.-N.S.; Visualization, A.-N.S. and M.M.A.; Writing—original draft, A.-N.S.; Writing—review and editing, A.-N.S. and M.M.A. Most of the work in this paper was conducted by A.-N.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

Acknowledgments: The authors thank their colleagues, who provided insight and expertise that greatly assisted in the research.

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

References

1. Flacco, F.; Kroger, T.; De Luca, A.; Khatib, O. A Depth Space Approach to Human-Robot Collision Avoidance. In Proceedings of the 2012 IEEE International Conference on Robotics and Automation, Saint Paul, MN, USA, 14–18 May 2012; pp. 338–345.
2. Schmidt, B.; Wang, L. Contact-less and Programming-less Human-Robot Collaboration. In Proceedings of the Forty Sixth CIRP Conference on Manufacturing Systems 2013, Setubal, Portugal, 29–30 May 2013; Volume 7, pp. 545–550.
3. Anton, F.D.; Anton, S.; Borangiu, T. Human-Robot Natural Interaction with Collision Avoidance in Manufacturing Operations. In *Service Orientation in Holonic and Multi Agent Manufacturing and Robotics*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 375–388, ISBN 9783642358524.
4. Kitaoka, M.; Yamashita, A.; Kaneko, T. Obstacle Avoidance and Path Planning Using Color Information for a Biped Robot Equipped with a Stereo Camera System. In Proceedings of the 4th Asia International Symposium on Mechatronics, Singapore, 15–18 December 2010; pp. 38–43.
5. Lenser, S.; Veloso, M. Visual Sonar: Fast Obstacle Avoidance Using Monocular Vision. In Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003), Las Vegas, NV, USA, 27–31 October 2003.
6. Ali, W.G. A semi-autonomous mobile robot for education and research. *J. King Saud Univ. Eng. Sci.* **2011**, *23*, 131–138. [CrossRef]
7. Lam, T.L.; Yip, H.W.; Qian, H.; Xu, Y. Collision Avoidance of Industrial Robot Arms using an Invisible Sensitive Skin. In Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, Algarve, Portugal, 7–12 October 2012; pp. 4542–4543.
8. Haddadin, S.; Albu-sch, A.; De Luca, A.; Hirzinger, G. Collision Detection and Reaction: A Contribution to Safe Physical Human-Robot Interaction. In Proceedings of the 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, Nice, France, 22–26 September 2008; pp. 3356–3363.
9. Cho, C.; Kim, J.; Lee, S.; Song, J. Collision detection and reaction on 7 DOF service robot arm using residual observer. *J. Mech. Sci. Technol.* **2012**, *26*, 1197–1203. [CrossRef]

10. Morinaga, S.; Kosuge, K. Collision Detection System for Manipulator Based on Adaptive Impedance Control Law. In Proceedings of the 2003 IEEE International Conference on Robotics & Automation, Taipei, Taiwan, 14–19 September 2003; pp. 1080–1085.
11. Jung, B.; Koo, J.C.; Choi, H.R.; Moon, H. Human-robot collision detection under modeling uncertainty using frequency boundary of manipulator dynamics. *J. Mech. Sci. Technol.* **2014**, *28*, 4389–4395. [CrossRef]
12. Min, F.; Wang, G.; Liu, N. Collision Detection and Identification on Robot Manipulators Based on Vibration Analysis. *Sensors* **2019**, *19*, 1080. [CrossRef] [PubMed]
13. Indri, M.; Trapani, S.; Lazzero, I. Development of a Virtual Collision Sensor for Industrial Robots. *Sensors* **2017**, *17*, 1148. [CrossRef]
14. Lu, S.; Chung, J.H.; Velinsky, S.A. Human-Robot Collision Detection and Identification Based on Wrist and Base Force/Torque Sensors. In Proceedings of the 2005 IEEE International Conference on Robotics and Automation, Barcelona, Spain, 18–22 April 2005; Volume i, pp. 796–801.
15. Cho, C.; Kim, J.; Kim, Y.; Song, J.-B.; Kyung, J.-H. Collision Detection Algorithm to Distinguish Between Intended Contact and Unexpected Collision. *Adv. Robot.* **2012**, *26*, 1825–1840. [CrossRef]
16. Dimeas, F.; Avendano-valencia, L.D.; Aspragathos, N. Human–Robot collision detection and identification based on fuzzy and time series modelling. *Robotica* **2014**, *33*, 1886–1898. [CrossRef]
17. Kim, D.; Lim, D.; Park, J. Transferable Collision Detection Learning for Collaborative Manipulator Using Versatile Modularized Neural Network. *IEEE Trans. Robot.* **2021**, *38*, 2426–2445. [CrossRef]
18. Czubenko, M.; Kowalczuk, Z. A simple neural network for collision detection of collaborative robots. *Sensors* **2021**, *21*, 4235. [CrossRef]
19. Sharkawy, A.-N.; Aspragathos, N. Human-Robot Collision Detection Based on Neural Networks. *Int. J. Mech. Eng. Robot. Res.* **2018**, *7*, 150–157. [CrossRef]
20. Sharkawy, A.-N.; Mostfa, A.A. Neural Networks' Design and Training for Safe Human-Robot Cooperation. *J. King Saud Univ. Eng. Sci.* **2021**, 1–15. [CrossRef]
21. Sharkawy, A.-N.; Koustoumpardis, P.N.; Aspragathos, N. Manipulator Collision Detection and Collided Link Identification based on Neural Networks. In *Advances in Service and Industrial Robotics. RAAD 2018. Mechanisms and Machine Science*; Nikos, A., Panagiotis, K., Vassilis, M., Eds.; Springer: Cham, Switzerland, 2018; pp. 3–12.
22. Sharkawy, A.-N.; Koustoumpardis, P.N.; Aspragathos, N. Neural Network Design for Manipulator Collision Detection Based only on the Joint Position Sensors. *Robotica* **2020**, *38*, 1737–1755. [CrossRef]
23. Sharkawy, A.-N.; Koustoumpardis, P.N.; Aspragathos, N. Human–robot collisions detection for safe human–robot interaction using one multi-input–output neural network. *Soft Comput.* **2020**, *24*, 6687–6719. [CrossRef]
24. Sharkawy, A.-N. Intelligent Control and Impedance Adjustment for Efficient Human-Robot Cooperation. Doctoral Dissertation, University of Patras, Patras, Greece, 2020.
25. Murray, R.M.; Li, Z.; Sastry, S.S. *A Mathematical Introduction to Robotic Manipulation*; CRC Press: Boca Raton, FL, USA, 1994; ISBN 9780849379819.
26. Sharkawy, A.N.; Koustoumpardis, P.N. Dynamics and computed-torque control of a 2-DOF manipulator: Mathematical analysis. *Int. J. Adv. Sci. Technol.* **2019**, *28*, 201–212.
27. Haykin, S. *Neural Networks and Learning Machines*, 3rd ed.; Pearson: London, UK, 2009; ISBN 9780131471399.
28. Nielsen, M.A. *Neural Networks and Deep Learning*; Determination Press: San Francisco, CA, USA, 2015; Available online: http://neuralnetworksanddeeplearning.com/ (accessed on 18 May 2022).
29. Sharkawy, A.-N. Principle of Neural Network and Its Main Types: Review. *J. Adv. Appl. Comput. Math.* **2020**, *7*, 8–19. [CrossRef]
30. Smith, A.C.; Hashtrudi-Zaad, K. Application of neural networks in inverse dynamics based contact force estimation. In Proceedings of the 2005 IEEE Conference on Control Applications, Toronto, ON, Canada, 28–31 August 2005; pp. 1021–1026.
31. Patiño, H.D.; Carelli, R.; Kuchen, B.R. Neural Networks for Advanced Control of Robot Manipulators. *IEEE Trans. Neural. Networks* **2002**, *13*, 343–354. [CrossRef]
32. Goldberg, K.Y.; Pearlmutter, B.A. *Using a Neural Network to Learn the Dynamics of the CMU Direct-Drive Arm II*; Carnegie Mellon University: Pittsburgh, PA, USA, 1988.
33. Leontaritis, I.J.; Billings, S.A. Input-output parametric models for non-linear systems Part I: Deterministic non-linear systems. *Int. J. Control* **1985**, *41*, 303–328. [CrossRef]
34. Boussaada, Z.; Curea, O.; Remaci, A.; Camblong, H.; Bellaaj, N.M. A nonlinear autoregressive exogenous (NARX) neural network model for the prediction of the daily direct solar radiation. *Energies* **2018**, *11*, 620. [CrossRef]
35. Mohanty, S.; Patra, P.K.; Sahoo, S.S. Prediction of global solar radiation using nonlinear auto regressive network with exogenous inputs (narx). In Proceedings of the 2015 39th National Systems Conference, NSC 2015, Greater Noida, India, 14–16 December 2015.
36. Pisoni, E.; Farina, M.; Carnevale, C.; Piroddi, L. Forecasting peak air pollution levels using NARX models. *Eng. Appl. Artif. Intell.* **2009**, *22*, 593–602. [CrossRef]
37. Zibafar, A.; Ghaffari, S.; Vossoughi, G. Achieving transparency in series elastic actuator of sharif lower limb exoskeleton using LLNF-NARX model. In Proceedings of the 4th RSI International Conference on Robotics and Mechatronics, ICRoM 2016, Tehran, Iran, 26–28 October 2016; pp. 398–403.

38. Bouaddi, S.; Ihlal, A.; Ait mensour, O. Modeling and Prediction of Reflectance Loss in CSP Plants using a non Linear Autoregressive Model with Exogenous Inputs (NARX). In Proceedings of the 2016 International Renewable and Sustainable Energy Conference (IRSEC), Marrakech, Morocco, 14–17 November 2016; pp. 1–4.
39. Du, K.; Swamy, M.N.S. *Neural Networks and Statistical Learning*; Springer: Berlin/Heidelberg, Germany, 2014; ISBN 9781447155706.
40. Marquardt, D.W. An Algorithm for Least-Squares Estimation of Nonlinear Parameters. *J. Soc. Ind. Appl. Math.* **1963**, *11*, 431–441. [CrossRef]
41. Hagan, M.T.; Menhaj, M.B. Training Feedforward Networks with the Marquardt Algorithm. *IEEE Trans. Neural. Networks* **1994**, *5*, 2–6. [CrossRef] [PubMed]
42. *ISO/TS 15066*; Robots and Robotic Devices–Collaborative Robots 2016. International Organization for Standardization: Geneva, Switzerland, 2016; pp. 1–40.
43. Sharkawy, A.-N.; Koustoumpardis, P.N. Human–Robot Interaction: A Review and Analysis on Variable Admittance Control, Safety, and Perspectives. *Machines* **2022**, *10*, 591. [CrossRef]
44. Briquet-Kerestedjian, N.; Wahrburg, A.; Grossard, M.; Makarov, M.; Rodriguez-Ayerbe, P. Using neural networks for classifying human-robot contact situations. In Proceedings of the 2019 18th European Control Conference, ECC 2019, Naples, Italy, 25–28 June 2019; pp. 3279–3285.
45. Franzel, F.; Eiband, T.; Lee, D. Detection of Collaboration and Collision Events during Contact Task Execution. In Proceedings of the IEEE-RAS International Conference on Humanoid Robots, Munich, Germany, 19–21 July 2021; pp. 376–383.