Human-Robot Collision Detection based on Neural Networks

Article · March 2018

DOI: 10.18178/ijmerr.7.2.150-157

Human-Robot Collision Detection Based on Neural Networks

Abdel-Nasser Sharkawy[1, 2*] and Nikos Aspragathos[2]
[1]Mechanical Engineering Department, Faculty of Engineering, South Valley University, Qena 83523, Egypt
[2] Department of Mechanical Engineering and Aeronautics, University of Patras, Rio 26504, Greece
Email: eng.abdelnassersharkawy@gmail.com, asprag@mech.upatras.gr

***Abstract*—In this paper, an approach based on multilayer neural network is proposed for human-robot collisions detection. The neural network is trained by Levenberg-Marquardt algorithm to the dynamics of the robot with and without external contacts to detect unwanted collisions of the human operator with the robot using only the proprietary position and joint torque sensors of the manipulator. The proposed method is evaluated experimentally using the 7-DOF KUKA LWR manipulator and the results illustrate that the developed system is efficient and very fast in detecting the collisions.**

***Index Terms*—Collision Detection, Neural Networks, Levenberg-Marquardt, Proprietary Sensors.**

I. Introduction

When the robots and humans share the same workspace, safety is very important factor because the proximity of the operator to the robot can lead to potential injuries so a system for safety based on collision avoidance or detecting the collision should be available. Collision can be avoided by having the knowledge of the environment using vision or proximity sensors. Mohammed et.al [1] introduced a solution based on vision using virtual 3D models of robots, real images of human operators from depth cameras and vision sensing units. Using the proximity sensors, Lam et.al [2] presented an invisible sensitive skin built inside the robot arm using 5 contactless capacitive sensors and specially designed antennas. Although these methods can be used to avoid the collision but modifications in the robot body are required for the sensors installation and the cost is increased.

For improving the safety system in HRI apart from the level of collision avoidance, a second level of collision detection and reaction is required if the first level protection fails. Some researchers sought to develop methods to detect the collisions. Fault detection methods that were based on the principle: small changes or faults in a structure can cause significant deviations in its dynamic behavior were proposed in robotics research. De Luca and Mattone's idea was to handle a collision at a generic point along the robot as a fault of its actuating system [3]. Cho et.al [4] continued their research and proposed the disturbance observer method to detect the collisions based on the generalized momentum and joint torque sensors.

Another Approach to detect the collisions is based on impedance control. Morinaga and Kosuge [5] proposed a nonlinear adaptive impedance control law without using external sensors and based on the difference between the reference and actual input torque to the manipulator.

Approaches based on fuzzy logic and neural network systems were proposed. Dimeas et.al [6, 7] implemented two methods one based on intelligent fuzzy identification and another based on time series. The fuzzy system detected the collisions very fast and accurately by having the lowest threshold value. In addition, the time series system estimated the collision torque by only using the measured joint position signal but its threshold was higher than the fuzzy system and similar to that of the model-based approach. Shujun Lu et.al [8] presented a collision detection approach using two six-axis force/torque sensors; one on the base and the other on the wrist and they developed two systems one based on a neural network training and another model based. Although the results illustrated the validity of the developed collision detection scheme but using two sensors make the cost quite high.

In this paper, a Neural Network (NN) based approach is proposed considering the properties of the NN. The NN derives its computing power through its massively parallel distributed structure and its ability to learn and therefore to generalize. These two information processing capabilities make it possible for neural networks to find good approximate solutions to complex (large-scale) problems that are intractable. NN offer also useful properties and capabilities such Nonlinearity, Input–Output Mapping and Adaptivity [9]. NN can approximate any function or in another meaning they have a kind of universality [10] e.g. approximation of smooth batch data containing the input, output and possibly gradient information of a function [11], approximating the derivatives of a function [12].

In the proposed method, a NN is trained by Levenberg-Marquardt (LM) algorithm and does not require a priori knowledge of the dynamic model of the robot and it is simple and easily applied. This method is applied without using any external sensors just using the position and joint torque sensors which are proprietary to the KUKA LWR manipulator that is used for the experiments. Our

Manuscript received November 14, 2017; revised February 6, 2018.

doi: 10.18178/ijmerr.7.2.150-157

method can be used in any robot having joint torque sensors without knowledge of its model since the estimation of external torque given by the robot controller is used only for training the NN and it can be replaced by an external sensor then the trained NN is used to estimate the external torque and detect the collision. NN Toolbox in Matlab is applied to the data for fast training and convergence.

II. Collision Detection Method

The dynamics of an n-link robot assumed for the flexible joint robot can be defined as [13]:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau + DK^{-1}\dot{\tau} + \tau_{ext} \quad (1)$$

$$B\ddot{\theta} + \tau + DK^{-1}\dot{\tau} = \tau_m \quad (2)$$

$$\tau = K(\theta - q) \quad (3)$$

where the vectors $q, \dot{q}, \ddot{q} \in R^n$ contain the joint positions of the manipulator and their corresponding time derivatives, $\theta \in R^n$ is the measured motor position, $M(q) \in R^{n\times n}$ is the inertia matrix that depends on the variable q and contains unknown constant terms, $C(q,\dot{q}) \in R^{n\times n}$ is a matrix containing the Coriolis and centrifugal terms that depends on the variables q and $\dot{q}$ and contains unknown constant terms, $G(q) \in R^n$ is the gravity vector that depends on the variable q and contains unknown constant terms. $K = diag(K_i) \in R^{n\times n}$ and $B = diag(B_i) \in R^{n\times n}$ are the diagonal, positive definite joint stiffness, and motor inertia matrices, respectively and they are unknown constants, and $D = diag(D_i) \in R^{n\times n}$ is the diagonal positive semi-definite joint damping matrix and it is unknown constant. The vector $\tau \in R^n$ represents the measured joint torque, $\tau_m \in R^n$ is the motor input torques vector and $\tau_{ext} \in R^n$ is the external torque vector from the collision acting on the robot and it can be calculated from (1) by

$$\tau_{ext} = M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) - \tau - DK^{-1}\dot{\tau} \quad (4)$$

Because it is difficult to determine the unknown dynamic coefficients in the dynamic model of the manipulator as shown from (1) to (4), a neural network based nonlinear estimation model is used in this paper to approximate the τ_{ext} function given by (4). In the proposed method, a NN is trained using the data with and without collision measured by the joint position and torque sensors. In this method the estimation of external torque given by the robot controller is used only for training the NN but this could be measured by any other external sensor so our method can be used in any robot without knowledge its model having joint torque sensors.

After many trials and experiments using different sets of inputs to train the network using LM algorithm, it is found that the main inputs for the neural network that give us the minimum mean squared error (mse) are the current position error $q_e(k)$ between the desired and actual joint position, the previous position error $q_e(k-1)$, the actual joint velocity $\dot{q}$ and the measured joint torque τ. We found these inputs are greatly influenced by the presence of a collision. However, large alterations in these variables are not always indicators of a collision, as these changes can also be observed under normal operation, due to high inertial forces during abrupt changes of the velocity. Since this phenomenon is related to the joint velocity so this is the cause of using the signal of the actual joint velocity $\dot{q}$ as an input for the training in order to distinguish the collision spikes and this discussed also in [6, 7]. In addition, it is found that using the actual joint velocity the mean squared error is reduced considerably during training of the NN. Although the signals from the torque sensors include a small noise, no digital filters are used to avoid delays. To be sure that this assumption is correct, a sinusoidal motion q with variable frequency is commanded on a single joint of KUKA LWR robot (Joint E1) and the collisions are performed by the human hand touching the manipulator as shown in Fig.1 the $q_e, \dot{q}$ and τ inputs to the NN without and with collision occurrence are illustrated.

From Fig. 1, it is clear that there are sudden changes and spikes in the variables in case of collisions compared with the corresponding variables without collision. Without collision, small alterations and spikes in the position error and measured joint torques are resulted from the friction and the high inertial forces appearing on the links (blue dashed curves). In case of collision, the spikes in the position error and the measured joint torque diagrams (small black spot) that face the small spikes in the actual joint velocity diagram are representing the collision and the other spikes from the inertial forces. The spike at the end of the position error diagram (small blank spot) in the two curves means that the robot starts to brake. To show more clearly what the small spikes in the actual joint velocity diagram represents, for example, the interval [0, 12] seconds from Fig. 1 is extended in Fig. 2 where the small spikes at points **c** represent collisions whereas the spikes at points **i** come from the inertial forces.

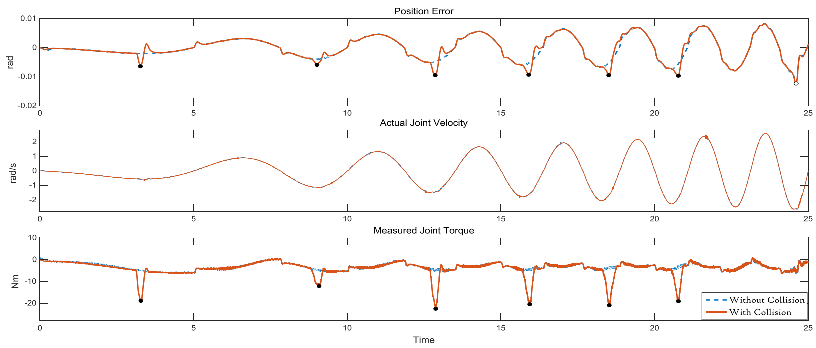


Figure 1. The main Inputs of the neural network.

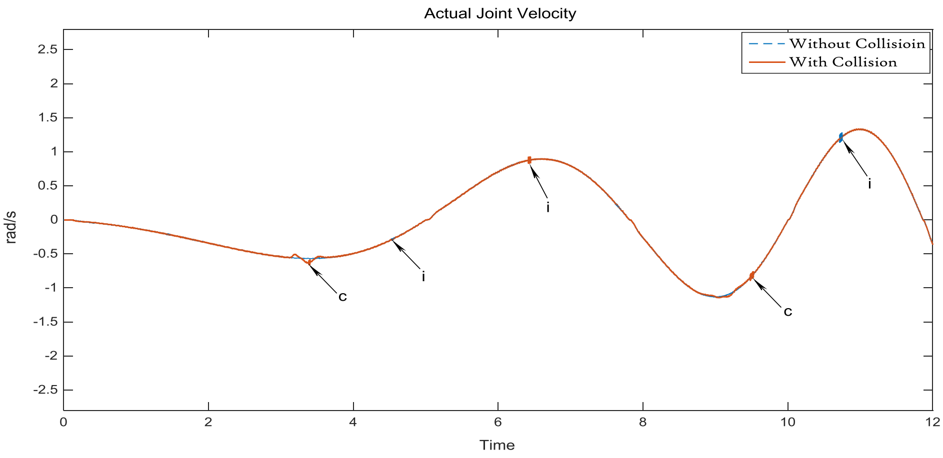


Figure 2. The small spikes in the actual joint velocity diagram that differentiate between the collisions and inertial forces.

III. Neural Network Design

Multilayer neural network is a powerful tool for the non-linear systems identification. It can adapt itself by changing the network parameters in a surrounding environment and can easily handle imprecise, fuzzy, noisy, and probabilistic information [14]. NN has vital role in identification of the dynamic systems and fault detection since it not only can be used to detect the occurrence of the fault but it also provides a postfault model of the robotic manipulator. This postfault model can be effectively used to isolate and identify the fault and, if possible, for accommodation of the failure [15].

Levenberg-Marquardt learning is used here to train the network and do the work in fast and stable way. Levenberg-Marquardt algorithm is a type of the Second-order optimization techniques that have a strong theoretical basis and provide significantly fast convergence and it is considered an approximation to Newton's Method [16, 17]. Compared with other learning methods, LM learning is used because it has the trade-off between the fast learning speed of the classical Newton's method and the guaranteed convergence of the gradient descent [16, 18]. LM algorithm always suits to larger data sets and converges in less iterations and in shorter time than the other training methods.

Three layers are used to compose the network as it is shown in Fig. 3; the first one is the input layer that contains the four inputs for the NN which are the joint position error $q_e(k)$, the previous position error $q_e(k-1)$, the actual joint velocity $\dot{q}$ and the measured joint torque τ, the second is the non-linear hidden layer and the third is the output layer that calculates the estimated external torque τ'_{ext} that is compared with the estimated external torque τ_{ext} derived by the Kuka robot controller (KRC). It should be noted here that the estimated external torque τ_{ext} is used only for training of the network and the external collision force can be measured by any external sensor and transformed to the joint torque by the Jacobian.

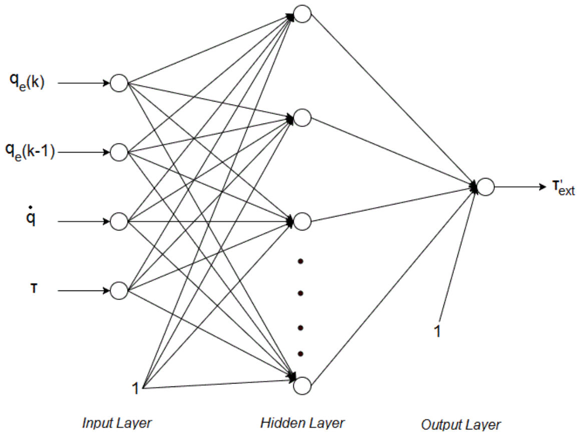


Figure 3. Multilayer Neural Network

The training error $e(t)$ should be small as possible. From the block diagram that illustrates the process of training the neural network as shown in Fig. 4, the error is given by

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

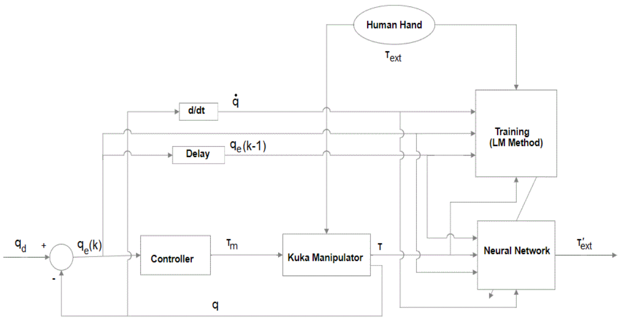


Figure 4. Block Diagram of the Neural Network system trained by LM Algorithm.

IV. Experiments

The proposed contact detection method is implemented on KUKA LWR IV manipulator (Light Weight Robot) as shown in Fig. 5. KUKA LWR robot is characterized by an extremely light anthropomorphic structure with 7 revolute joints and driven by compact brushless motors via harmonic drives. The presence of such transmission elements introduces a dynamically time-varying elastic displacement at each joint, between the angular position of the motor and that of the driven link. All joints are equipped with position sensors on the motor and link sides, and with a joint torque sensor. The KR C5.6 lr robot controller unit, together with the so-called Fast Research Interface (FRI) [19], is able to provide (at a 1 msec sampling rate) the link position q, velocity $\dot{q}$ and joint torque τ measurements and the estimation of the external torques τ_{ext}.

Figure 5. Experimental setup with the KUKA LWR manipulator.

The robot performs a single joint motion around the vertical axis. The selected excitation signal $q_d(t)$ is a sinusoidal profile of the joint position with variable frequency since it enables the sufficient dynamic excitation of the structure and the acquisition of rich signals. The motion of joint E1 is given as

$$q_d(t) = -A + A\cos(2\pi f t) \tag{6}$$

where $A = \frac{\pi}{4}$ and f is the frequency which is linearly increasing from 0.05 Hz to 0.326 Hz. This frequency produces angular velocity ω up to 2.05 rad/s.

The training data are divided into two sets: in the first set the robot joint performs the motion without any external force applied to the robot body and in the second set the same motion is performed with the user performing collisions suddenly and stochastically with his hand. During the experiments the robot is commanded to move with position control mode and no reaction strategy is implemented. The performed collisions are applied momentarily and the safety of the human operator during this experiment is taken into consideration.

A. NN Training

Using the data with and without contact together, the neural network system is created and trained. The total number of input-output pairs collected from the experiments and used are 56358. From these data 90% are used for training, 5% for validation and 5% for testing. After more different trials and initializations, it is found that the best number of hidden neurons are 90 and the number of iterations are 932 that give us the minimum mse and the adequate threshold. The training process is very fast and stable and applied using Matlab on an *Intel(R) Core(TM) i3-6100 CPU @ 3.70GHz* processor. The experiments of NN structure determination and training are discussed in APPENDIX A.

The trained NN is evaluated with the same data-set that are used for the training. The difference between the external torque τ_{ext} given by KRC and the external torque τ'_{ext} estimated by the NN system is illustrated in Fig. 6. It is clear that the approximation error of the collision torque is high, compared to the error of contact-free motion (blue dashed curve) where the average of the absolute error values is very small (0.0955Nm) and the maximum value of the absolute values of the error is 1.6815 Nm that is used as the threshold above which, a collision is assumed, when $|\tau'_{ext}| > 1.6815$. In the literature, thresholds were defined in different ways. In [20], the threshold was defined as 10% of the maximum nominal torque of the robot, Dimeas et.al [6] used it as the maximum train error from contact-free motion training and in [8] it was defined as a value below the contact force that parameterizes the unified pain tolerance limit of a human, which was determined in [21]. In this paper threshold is identified as in [6] and after calculating it, it is found that it is also near to 10% of the maximum nominal torque of the robot.

To examine more carefully the approximation error, the estimated external torque resulted from KRC and the estimated external torque from the NN are compared as it is shown in Fig. 7.

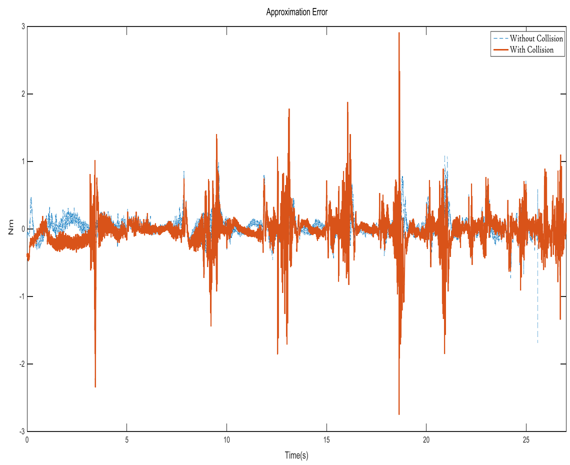


Figure 6. Neural Network approximation error

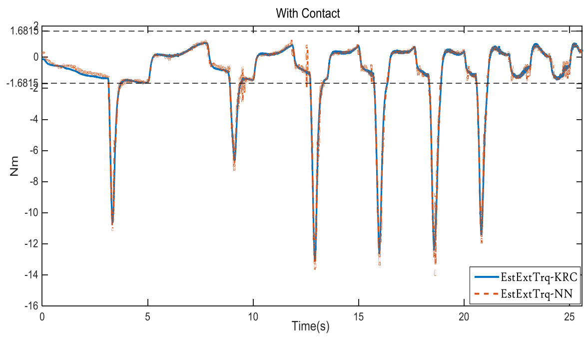


Figure 7. The two estimated external collision torques from KRC and NN

Using the proposed method, the collision can be identified very fast. For instance using the interval [2.5, 4.5] seconds from Fig. 7 where the first collision is occurred, the detection time is calculated easily as shown in Fig. 8. The detection time is calculated as the elapsed time from the start of a collision (point **a** in the curve, where the slope starts monotonically increasing) to the moment when the estimated external collision torque by the trained NN exceeds the threshold (point **b** in the curve) and from the figure the detection time is 11.5 ms after collision occurrence.

B. *Verification and Testing*

Since the training data were obtained only with variable velocity, the proposed method is tested in the experimental setup by commanding the robot to perform a single joint motion around the vertical axis with constant velocity profiles to verify that the method is able to identify collisions and considering its ability to generalize under a variety of conditions. The trained NN is evaluated with two different speeds, equal to 0.5rad/s and 1.0rad/s respectively, as it is shown in Fig. 9 where with 6 collisions occurred on each of them with different values and directions.

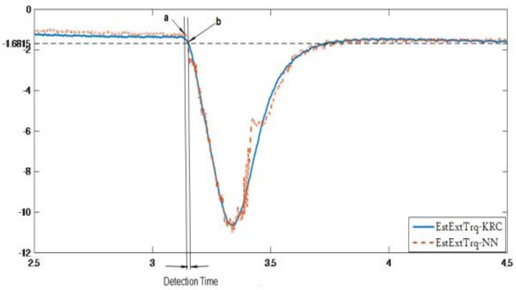
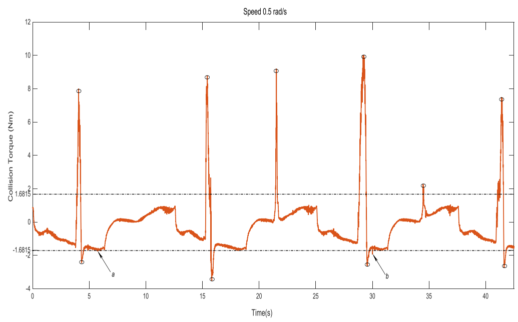
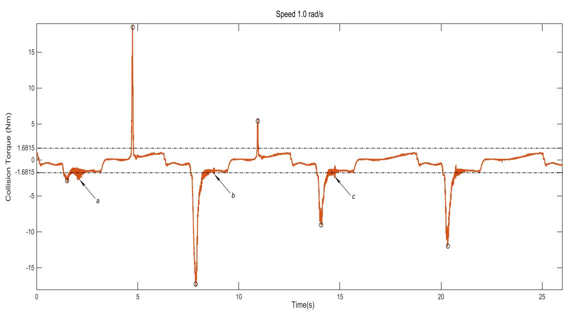


Figure 8. Calculation the collision detection time for the first collision.



(a) At speed 0.5 rad/s



(b) At speed 1.0 rad/s

Figure 9. The estimated external collision torque from the proposed method.

At speed 0.5 rad/sec (Fig. 9a), the trained NN is able to identify all collisions (small rings in the curve) but 2 collision alerts falsely detected since does not correspond to an actual collision (point **a** and **b** in the curve). The collision detection time is low e.g. at the second collision is 31.92 ms. At speed 1.0 rad/sec (Fig. 9b), all collisions are detected (small rings in the curve) but 3 collision alerts falsely detected (point **a**, **b** and **c** in the curve). The collision detection time at the second collision is 13.3 ms.

To confirm the validity and efficiency of the proposed method under a wide range of operating conditions and acquire a performance measure, another 25 trials of collisions (NC) are evaluated with various magnitudes, directions and different velocities of motion. Table I provides the performance in terms of the number of the correct detected collisions (CC), the number of false negatives (FN) which is the number of collisions not detected by the method and the number of the false positives (FP) which are the collision alerts provided by the method when there is not an actual collision.

TABLE I. PERFORMANCE OF THE COLLISION DETECTION METHOD FOR DIFFERENT COLLISION SCENARIOS AND VELOCITIES

Method	NC	CC	FN	FP
NN Trained by LM	25	21	4	2
Percentage		84 %	16 %	8 %

Conventions: NC: number of collisions, CC: number of correct detected collisions, FN: number of false negatives, FP: number of the false positives.

Table I shows that the proposed method (NN) succeeds to detect the collisions with high percentage (84 %). Also the number of the false positive collisions is low (8 %) which means that our method is less sensitive to the external disturbances and unmodelled parameters so it is robust.

V. DISCUSSION OF THE RESULTS

The proposed method is easily applied and understood and the training is very fast. The fuzzy based part of [6] is compared with our method. The estimated external torque τ_{ext} given by the robot controller is used here only for training the NN, whereas Dimeas et.al [6] used the external torque measured by an ATI F/T Nano 25 force sensor for verification and the training. The data used in our proposed method for training the NN are 56,358 input-output pairs, which are lower than the data used by Dimeas et.al which were 70,000. The average error of contact-free motion in our method is 0.0955 Nm which is a little bit lower than resulted by Dimeas et.al which is 0.1 Nm so using the previous position error $q_e(k-1)$ in the proposed method improves the convergence and the reliability more than done by Dimeas et.al.

In [8], the cost of method proposed by Lu et.al is quite high because two force sensors were used, whereas our method does not require any external sensors and is used with any robot without model knowledge but having joint torque sensors.

Our method presents also low detection time for the collisions. It should be noted that because of the different data and operating conditions used in our paper and the other two papers, it is difficult to compare quantitatively the time required for the detection of the collisions.

VI. Conclusion and Future Work

In this paper, a method was proposed for human-robot collision detection based on the multilayer neural network trained by Levenberg-Marquardt algorithm. The training was stable and very fast. The inputs to the NN were derived from the position and joint torque sensors and the method was able to detect the collision of the robot with the human hand very quickly. The NN system was designed by trial and error to evaluate the external collision torque and was trained using the collected data with and without collision after carrying out the experiments based on the motion of a single joint (Joint E1) of the 7-DOF KUKA LWR robot. From that way, a rich signal was obtained and the training error became very small.

Because of the good results here, in the future the extension of the proposed approach is considered to implement the collision detection system for multiple joints of the manipulator, where apart from detecting the occurrence of collision the collided link will be identified too.

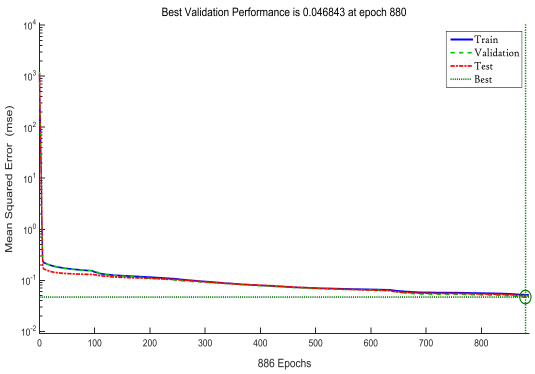
Appendix A

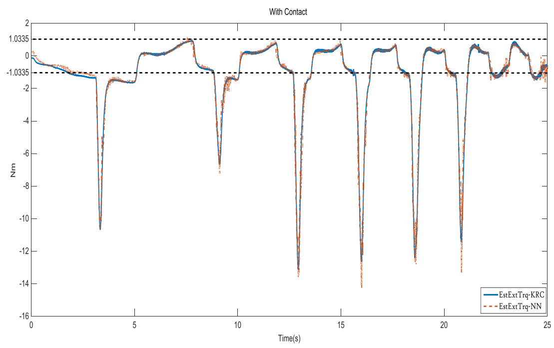
Many experiments are done using different sets of inputs to train the NN by LM algorithm using the data with and without contact together. For every set of inputs, different numbers of hidden neurons, starting from 30 to 120 hidden neurons, are tried to train the NN and with every number of hidden neurons, many different initializations of the NN are used until the minimum and the best mean squared error and the adequate threshold are obtained. Three cases only from the all experiments are presented here to show the process of NN structure determination and training.

Case 1 (Fig. 10): the current position error $q_e(k)$, the actual joint velocity $\dot{q}(k)$ and the measured joint torque τ are used as the three inputs to the NN. After many trials using different numbers of hidden neurons and initializations, it is found that the lowest mse (0.046843) is obtained using 120 hidden neurons as shown in Fig. 10a. The average of the absolute error values of contact-free motion is 0.0856 Nm. The threshold is 1.0335 Nm and using this value, the proposed method fails to detect the collisions well and the performance is poor because most of the collisions start at a point below the threshold value and there are FP, which are the collision alerts when there is not an actual collision, as in the interval [22, 25] seconds shown in Fig. 10b.

Case 2 (Fig. 11): the current position error $q_e(k)$, the current actual joint velocity $\dot{q}(k)$, the previous actual joint velocity $\dot{q}(k-1)$ and the measured joint torque τ are used as the inputs to the NN. After many trials using different numbers of hidden neurons and initializations, it is found that the lowest mse (0.069297) is obtained using 70 hidden neurons which is higher than case 1 as shown in Fig. 11a. The average of the absolute error values of contact-free motion is 0.1010 Nm which also is high compared with the other cases. The threshold is 1.3495 Nm and using this value, the proposed method is better than case 1 but it gives a lot of FP, more than 5 FP as shown in Fig. 11b so the performance of the method also is not good.

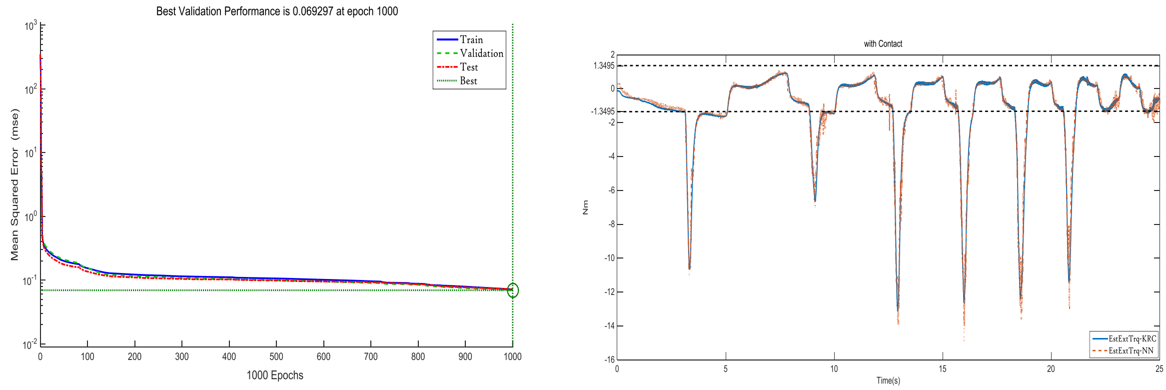
Case 3 (The best): This is the best case used in this paper where the current position error $q_e(k)$, the previous position error $q_e(k-1)$, the actual joint velocity $\dot{q}(k)$ and the measured joint torque τ are used as the inputs to the NN. After many trials using different numbers of hidden neurons and initializations, it is found that the lowest mse (0.040644) is obtained using 90 hidden neurons which is the lowest value compared with the all cases as shown in Fig. 12. The average of the absolute error values of contact-free motion is 0.0955 Nm. The threshold is 1.6815 Nm and using this value, the proposed method is the best case which can detect the collisions accurately and succeeds with high percentage as discussed in the paper and shown in Fig. 7.





(a) The lowest mse value. (b) The two estimated external collision torques from KRC and NN.

Figure 10. Case 1: Using the current position error $q_e(k)$, the actual joint velocity $\dot{q}(k)$ and the measured joint torque τ as the inputs to the NN.



(a) The lowest mse value. (b) The two estimated external collision torques from KRC and NN.

Figure 11. Case 2: Using the current position error $q_e(k)$, the current actual joint velocity $\dot{q}(k)$, the previous actual joint velocity $\dot{q}(k-1)$ and the measured joint torque τ as the inputs to the NN.

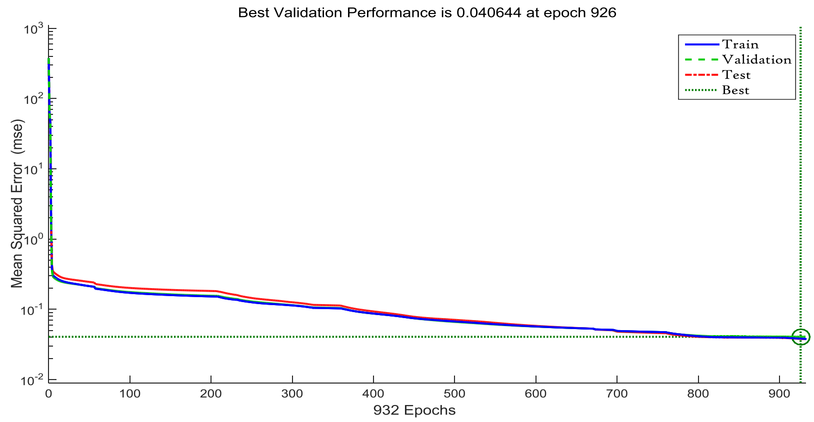


Figure 12. Case 3: the lowest Mse value using the current position error $q_e(k)$, the previous position error $q_e(k-1)$, the actual joint velocity $\dot{q}(k)$ and the measured joint torque τ as the inputs to the NN

ACKNOWLEDGMENT

Abdel-Nasser Sharkawy is funded by the "Egyptian Cultural Affairs & Missions Sector" and "Hellenic Ministry of Foreign Affairs Scholarship" for Ph.D. study in Greece. This paper is partly sponsored by the programme of introducing Talents of Discipline to University ("111 Program") under Grant No. B16034.

REFERENCES

[1] A. Mohammed, B. Schmidt, and L. Wang, "Active collision avoidance for human – robot collaboration driven by vision sensors," *Int. J. Comput. Integr. Manuf.*, vol. 30, no. 9, pp. 970–980, 2017.

[2] T. L. Lam, H. W. Yip, H. Qian, and Y. Xu, "Collision avoidance of industrial robot arms using an invisible sensitive skin," in *Proc. 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 4542–4543.

[3] A. De Luca and R. Mattone, "Sensorless robot collision detection and hybrid force / motion control," in *Proc. the 2005 IEEE International Conference on Robotics and Automation*, April 2005, pp. 999–1004.

[4] C. Cho, J. Kim, S. Lee, and J. Song, "Collision detection and reaction on 7 DOF service robot arm using residual observer," *J. Mech. Sci. Technol.*, vol. 26, no. 4, pp. 1197–1203, 2012.

[5] S. Morinaga and K. Kosuge, "Collision detection system for manipulator based on adaptive impedance control law," in *Proc. the 2003 IEEE International Conference on Robotics &Automation*, 2003, pp. 1080–1085.

[6] F. Dimeas, L. D. Avenda, E. Nasiopoulou, and N. Aspragathos, "Robot collision detection based on fuzzy identification and time series modelling," in *Proc. the RAAD 2013, 22nd International Workshop on Robotics in Alpe-Adria-Danube Region*, 2013.

[7] F. Dimeas, L. D. Avendano-valencia, and N. Aspragathos, "Human - Robot collision detection and identification based on fuzzy and time series modelling," *Robotica*, pp. 1–13, May 2014.

[8] S. Lu, J. H. Chung, and S. A. Velinsky, "Human-robot collision detection and identification based on wrist and base force / torque sensors," in *Proc. 2005 IEEE International Conference on Robotics and Automation*, April 2005, pp. 796–801.

[9] S. Haykin, *Neural Networks and Learning Machines*, Third Edit. Pearson, 2009, pp.1-6.

[10] M. A. Nielsen, *"Neural networks and deep learning."* Determination Press, 2015, ch.4.

[11] S. Ferrari and R. F. Stengel, "Smooth function approximation using neural networks," *IEEE Trans. NEURAL NETWORKS*, vol. 16, no. 1, pp. 24–38, 2005.

[12] K. Hornik, M. Stinchcombe, and H. White, "Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks," *Neurul Networks*, vol. 3, pp. 551–560, 1990.

[13] M. W. Spong, "Modeling and control of elastic joint robots," *IEEE J. Robot. Autom.*, vol. 109, no. 4, pp. 310–319, 1987.

[14] K. L. Du and M. N. S. Swamy, *Neural Networks in a Softcomputing Framework*. London: Springer, 2006.

[15] A. T. Vemuri and M. M. Polycarpou, "Neural-network-based robust fault diagnosis in robotic systems," *IEEE Trans. NEURAL*

Networks, vol. 8, no. 6, pp. 1410–1420, 1997.
[16] K. Du and M. N. S. Swamy, *Neural Networks and Statistical Learning*. Springer, 2014, ch.5, pp.127-133.
[17] D. W. Marquardt, “An algorithm for least-squares estimation of nonlinear parameters,” *J. Soc. Ind. Appl. Math.*, vol. 11, no. 2, pp. 431–441, 1963.
[18] M. T. Hagan and M. B. Menhaj, “Training feedforward networks with the marquardt algorithm,” *IEEE Trans. NEURAL NETWORKS*, vol. 5, no. 6, pp. 2–6, 1994.
[19] *KUKA.FastResearchInterface 1.0, KUKA System Technology (KST)*, version 2, D-86165 Augsburg, Germany, 2011.
[20] S. Haddadin, A. Albu-sch, A. De Luca, and G. Hirzinger, “Collision detection and reaction : A contribution to safe physical human-robot interaction,” in *Proc. 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008, pp. 3356–3363.
[21] Y. Yamada, Y. Hirasawa, S. Huang, Y. Umetani, and K. Suita, “Human – robot contact in the safeguarding space,” *230 IEEE/ASME Trans. MECHATRONICS*, vol. 2, no. 4, pp. 230–236, *1997*.

Abdel-Nasser Sharkawy was born in Qena, Egypt in September 29, 1991, received his B.Sc. degree and M.Sc. degree in Mechanical Engineering (Mechatronics Division) from Faculty of Engineering, South Valley University, Qena, Egypt in 2013 and 2016. He started working as an assistant lecturer at the same university from 2013. Currently he is pursuing his Ph.D. degree in Mechanical Engineering (Robotics Group) at University of Patras, Patras, Greece in the field of Human-Robot Cooperation. His PhD thesis is funded by the “Egyptian Cultural Affairs & Missions Sector” and “Hellenic Ministry of Foreign Affairs Scholarship”. His research areas of interest include mechatronic systems, human - robot interaction, robot control and rehabilitation.

Nikos Aspragathos (Professor) leads the Robotics Group in Mechanical & Aeronautics Engineering Department, University of Patras, Greece. His main research interests are robotics, intelligent motion planning and control for static and mobile robots and for dexterous manipulation of rigid and non-rigid objects, knowledge-based design, industrial automation, and computer graphics. He is reviewer for about 40 Journals and more than 30 conferences, member of the editorial board of the Mechatronics Journal, ROBOTICA and ISRN Robotics. He published more than 70 papers in Journals and more than a 130 in conference proceedings. He was and is currently involved in research projects funded by Greek and European Union sources.