COBEM 2023 27TH INTERNATIONAL CONGRESS OF MECHANICAL ENGINEERING

ABCM Brazilian Society of Mechanical Sciences and Engineering
27th ABCM International Congress of Mechanical Engineering
December 04-08, 2023, Florianópolis-SC, Brazil

COB-2023-2100
MODE OBSERVER BASED ON MOVING HORIZON ESTIMATION APPROXIMATION THROUGH NEURAL NETWORKS

Lara Candido Alvim
Leonardo Dias Pereira
Helon Vicente Hultmann Ayala
Mechanical Engineering Department of Pontifical Catholic University of Rio de Janeiro (PUC-Rio)
Marques de Sao Vicente, 225, Zip code 22453-900, Rio de Janeiro, RJ, Brazil
llaracalvim@aluno.puc-rio.br; leonardo.pereira@aluno.puc-rio.br; helon@puc-rio.br

Elias Dias Rossi Lopes
Mechanical Engineering Department of Military Institute of Engineering (IME)
General Tibúrcio Sq, 80 - Urca, Zip code 22290-270, Rio de Janeiro, RJ, Brazil
eliasrossi@ime.eb.br

Abstract. *The advances in robotics allow a growing range of applications for robotic manipulators, such as the execution of tasks in which the human and robot are sharing the same environment or that require the detection ability of the End-effector for safety or performance issues, which encourages the application of control methods for contact detection. In this paper, for the contact detection estimation, we divide the system states into two groups the free mode when the robot is not in contact and the contact mode when it is. We consider contact as a control mode of the robotic manipulator. Mode detection is achieved based on identifying the active states of the nonlinear switching system. Artificial Neural Networks (ANNs) approach is applied to detect the estimated system states and then the system modes. This method estimates an approximative function that describes a dataset by minimizing the error between predicted and expected outputs. The dataset used to train the ANNs results of the Moving-Horizon State Estimation (MHSE) implementation for contact detection of a robotic manipulator. The main idea behind the MHSE approach is to estimate the system states using past measurements and a moving horizon window to solve at each instant of time a constrained nonlinear optimization problem. The implemented ANNs method can estimate the states and classify the mode effectively, presenting low RMSE values, high values of R^2 above 0.9, and a reduction in the processing time of the estimation algorithm when compared with the MHSE method.*

Keywords: *Switching System, Contact Detection, Moving-Horizon Estimation, Artificial Neural Networks (ANNs)*

1. INTRODUCTION

Safety is an important topic of study in Robotics, mainly due to the increase of tasks that require Human-Robot collaboration (HRC) or interaction (HRI), (Magrini et al., 2020; Zacharaki et al., 2020; Bi et al., 2020). The detection, identification, and prevention of collisions are becoming more relevant and necessary topics to be developed and consequently allow the integration between model-based methods and data-based methods to deal with this problem of avoiding harmful collisions and guaranteeing efficient performance in task execution, (Dong et al., 2020; Briquet et al., 2019; Sharkawy et al., 2019; Sharkawy et al., 2020).

Considering a switching behavior of a robotic dynamical model is possible to include the collision event as an internal variation of the model and use a Model-based approach, such as the Moving-Horizon State Estimation (MHSE) that allows the estimation of system dynamics through the solution of an optimization problem, to observe the behavior of the system in terms of collision, (Baglietto et al., 2012; Alvim et al., 2022). On the other hand, Artificial Neural Networks (ANN) which is a type of data-based approach, can also be designed to improve robotic system observability through training with data generated by model-based methods (Jiang et al., 2017). In this context some contributions can be highlighted, such as (Sharkawy et al., 2021, 2020, 2019), that propose applications of Neural Networks for detecting human-manipulator collisions or collisions avoidance, using datasets with and without contact obtained by sensors or dynamical model to training the NN and (Park et al., 2022) that use robot collision data for training the NN, obtained through the dynamical model. Concerning the application of neural networks as an approximative filter of the MHSE method, (Alessandri et al., 2011) and (Brunello et al., 2020) present results of approximate estimation and reduction of computational requirements.

In this work, we advance the state of the art of Alvim et al., 2022 proposing a Feedforward Neural Network training approach to obtain an approximate of the MHSE mode estimation for contact detection considering switching in the manipulator dynamics. The remainder of this work is organized as follows. Section 2 describes the robotic manipulator used to simulate the proposed methodology. Section 3 presents the methodology and the formulation of the approximative filter using NN. Section 4 presents the results. Lastly, section 5 gives the conclusions.

2. CASE STUDY

Considering both configurations for the 1-DOF robotic manipulator of this work presented in Figure 1 to represent our mechanical system, on the left-hand side we have the Free Model, comprised of two inertial elements, the motor J_m and the link J_l, both connected by an axis with a flexible joint modeled as a spring k, representing the system working under normal conditions with the link moving freely around the axis. And, on the right-hand side, we have the Contact Model comprised of only the first inertial element J_m, representing the system under faulty condition with the link unable to move due to some type of contact.

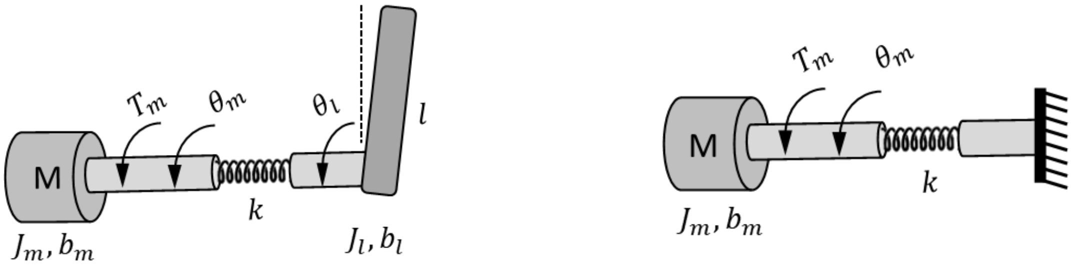

Figure 1. 1-DOF Robotic manipulator: Free Mode (Left), Contact Mode (Right)

The dynamical model of the 1-DOF robotic manipulator is defined by the ordinary differential equations in Eq. (1).

$$\boldsymbol{J_m\ddot{\theta}_m + b_m\dot{\theta}_m + k(\theta_m - \theta_l) = T_m} \tag{1a}$$

$$\boldsymbol{J_l\ddot{\theta}_l + k(\theta_m - \theta_l) + mgl\,sin\,\theta_l = 0} \tag{1b}$$

where, T_m is the motor torque, b_m is the motor viscous friction coefficient, b_l is the link viscous friction coefficient, θ_m is the angular position of the motor, θ_l is the angular position of the link, m is the link mass, g is gravity acceleration, and l is the link length. Suitable values for the system parameters are given to (Zhang et al., 2010) and (Fan and Arcak, 2003). Considering that the model behaves as a switched system, in addition to the gravity that behaves as a nonlinearity of the model, the switching between the free model and the contact model also behaves as a nonlinearity, therefore the dynamical model can be represented in the nonlinear space state representation, as present in Eq. (2).

$$\boldsymbol{\dot{x} = f(x, u)} \tag{2}$$

where $x = (x_1\ x_2\ x_3\ x_4) = \left(\theta_m\ \dot{\theta}_m\ \theta_l\ \dot{\theta}_l\right)^T$ is the vector of the continuous state variables, $\dot{x}$ is the derivative of the state variables, $u = (\ u_1) = (T_m)$ is the input, and f is a nonlinear function. Beyond the continuous states, the switched system has two discrete states represented by the system models, also called here mode, and represented by $\sigma_t \in \{1,2\}$, where $\sigma_t = 1$ is the (Free Mode) and $\sigma_t = 2$ is the (Contact Mode). Therefore, considering that the function varies in each mode the system equation is given in the form $\dot{x} = f_{\sigma_t}(x, u)$ and may be represented by Eq. (3).

$$\boldsymbol{\dot{x}} = \begin{cases} \boldsymbol{f_1(x_t, u_t)} = \begin{bmatrix} \boldsymbol{x_2} \\ \boldsymbol{\frac{1}{J_m}[u_1 - b_m x_2 - K(x_1 - x_3)]} \\ \boldsymbol{x_4} \\ \boldsymbol{\frac{1}{J_l}[k(x_1 - x_3) - mglsin(x_3)]} \end{bmatrix}, & \boldsymbol{if\ \sigma(t) = 1} \\ \boldsymbol{f_2(x_t, u_t)} = \begin{bmatrix} \boldsymbol{x_2} \\ \boldsymbol{\frac{1}{J_m}[u_1 - b_m x_2 - K(x_1 - x_3)]} \\ \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix}, & \boldsymbol{if\ \sigma(t) = 2} \end{cases} \tag{3}$$

3. METHODOLOGY

This section briefly describes the case study as well as the nonlinear dynamic model which is used in the Moving-Horizon Estimator.

3.1 Moving-Horizon State Estimation for switching systems

The dynamics of hybrid systems incorporate continuous and discrete states of a system, and to understand the behavior of these systems it is necessary to consider the variability of the subsystems or modes. For this, it is possible to isolate the discrete states of the system in switched events and consider that we have a switching system (Liberzon, 2003).

Mode observability is one of the capabilities of the MHSE method that can be used to evaluate switching systems, as it allows understanding the condition or internal state of a system based on external outputs and verifying the possibility of identifying and differentiating the outputs (Baglietto et al., 2012). MHSE consists of obtaining the current state of the system based on a finite window of past measurements N, also called observation window, by solving an optimization problem that is recalculated at each instant of time within the window until the current instant.

Considering a system in the type of Eq. (4), where the function f and h are nonlinear functions varying according to each mode $\sigma \in \mathcal{M}$ over time, $t \in Z_+ = \{0,1, \dots, \}$ is the time instant, $x_t \in \mathcal{X} \in \mathbb{R}^n$ is the state vector, $u_t \in \mathbb{R}^m$ is the control vector, $\varepsilon_t \in \mathbb{R}^n$ is the vector of additive disturbance affecting the system dynamics, $y_t \in \mathbb{R}^p$ is the observation vector and $\eta_t \in \mathbb{R}^p$ is the vector of measurements noises, the MHSE problem for switching systems is determine whether σ can be identified within a fixed estimation window, from the output $y_{t-N,t}$.

$$\boldsymbol{x_{t+1}} = \boldsymbol{f_\sigma(x_t\,, u_t) + \varepsilon_t} \tag{4a}$$

$$\boldsymbol{y_t} = \boldsymbol{h_\sigma(x_t) + \eta_t} \tag{4b}$$

The state estimates $[\hat{x}_{t-N|t}, \dots \hat{x}_{t|t}]$ varies within a window $[t-N, t]$ and the mode estimate $\hat{\sigma}_t$ remain constant. Both are calculated at time $t = N, N+1, \dots$, on basis of an information vector $I_t = [y_{t-N}, \dots, y_t, u_{t-N}, \dots, u_{t-1}]^T$ and of a prediction $\bar{x}_{t-N}$ in the beginning of the window $[t-N]$. The estimates at the current time can be obtained through the estimation procedure in Eq. (5),

$$\boldsymbol{\hat{\sigma}_t\,, \hat{x}_{t-N|t}} \in \mathbf{arg}\min_{s.t.\ \hat{x}_{t-N|t} \in \mathcal{X}\ ,\ \hat{\sigma}_t \in \mathcal{M}} \boldsymbol{J\left(\hat{x}_{t-N|t}, \bar{x}_{t-N}, \hat{\sigma}_t, I_t\right)} \tag{5}$$

by minimizing a least-squares cost function in the form of Eq. (6).

$$\boldsymbol{J\left(\hat{x}_{t-N|t}, \bar{x}_{t-N}, \hat{\sigma}_t, I_t\right) = \mu \parallel \hat{x}_{t-N|t} - \bar{x}_{t-N} \parallel^2 + \sum_{i=t-N}^{t} \parallel y_i - h_{\hat{\sigma}}\left(\hat{x}_{i|t}\,, u_i\right) \parallel^2} \tag{6}$$

where the coefficient μ is a positive scalar that serves as a penalty parameter that influences the fit of the estimated and predicted continuous state estimation at the beginning of the window. In this work, the function h does not depend on u_i.

The set of estimates is calculated according to Eq. (7) without noise, in the form of Eq. (4a) and the prediction is propagated to the next iteration as in Eq. (8)

$$\boldsymbol{\hat{x}_{k+1|t} = f_{\hat{\sigma}_t}\left(\hat{x}_{k|t}, u_k\right) \quad k = t-N, \dots, t-1} \tag{7}$$

$$\boldsymbol{\bar{x}_{t-N+1} = f_{\hat{\sigma}_{t-N}}\left(\hat{x}_{t-N|t}, u_{t-N}\right) \quad t = N, N+1, \dots} \tag{8}$$

The methodology can be summarized in three main steps: (1) Perform m optimization procedures with the cost function $J(\hat{x}_{t-N|t}, \bar{x}_{t-N},\ \hat{\sigma}_t\,, I_t)$, (2) Define the optimal estimates (i) of the discrete ($\hat{\sigma}$) and continuous states ($\hat{x}$) finding the smallest cost function $J(\hat{x}^i_{t-N|t}, \bar{x}_{t-N},\ \hat{\sigma}^i_t\,, I_t)$, and (3) Propagate the prediction $\bar{x}_{t-N+1}$. A more detailed description of this methodology and the implementation results are given in our previous work (Alvim et al., 2022).

3.2 Moving Horizon State Estimation for Contact Detection with Neural Networks

Artificial Neural Networks (ANNs) are part of the set of supervised learning methods that seek to approximate a function to a dataset by minimizing the error between predicted and estimated outputs during the training process of a network (Ray,2019). For this, it is necessary to provide a set of observations data for the development of estimates that approximate the unknown function, also known as the target function.

The Neural Networks that approximate the Moving-Horizon States and Mode Estimations, called NNMHSE in this work, can be defined as

$$\hat{\boldsymbol{x}}_{NNMHSE} = \boldsymbol{NN}(\hat{\boldsymbol{x}}_{MHSE}, \boldsymbol{I}_t) \tag{9}$$

$$\hat{\boldsymbol{\sigma}}_{NNMHSE} = \boldsymbol{NN}(\hat{\boldsymbol{\sigma}}_{MHSE}, \boldsymbol{I}_t) \tag{10}$$

where the input of such a network is the information vector I_t and the estimations vectors of $\hat{x}$ and $\hat{\sigma}$ along the receding horizon. Among the various neural network architectures, this work uses a Feedforward Neural Network, due to its simplicity and applicability. The solution of the NNMHE may be performed through software toolboxes, therefore the training algorithm used was the Levenberg-Marquardt (LMA), available on MATLAB®. Figure 2 schematizes the NNMHE problem.

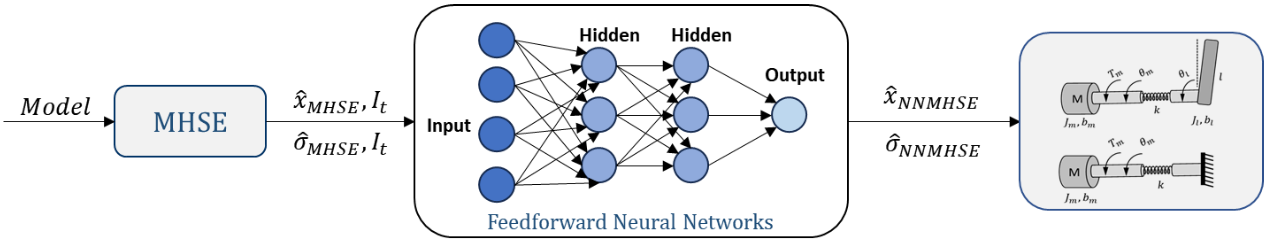

Figure 2. NNMHE Problem

Although it is a metric for linear regression, the coefficient of determination R² is often used as a validation metric for neural network models, due to the ease of visualization of the model fit, however, according to (Spiess and Neumeyer, 2010) for some models this metric may be inappropriate, therefore, it should be analyzed along with other non-linear regression metrics. For this reason, in this work, the evaluation metrics for validation of the fit between the real and the estimated data are the coefficient of determination (R^2) and the Root Mean Squared Error (RMSE) calculated as Eq. (11) and Eq. (12).

$$\boldsymbol{R^2} = \boldsymbol{1} - \frac{\sum_{t=1}^{n} (\boldsymbol{x}_i - \hat{\boldsymbol{x}}_i)^2}{\sum_{t=1}^{n} (\boldsymbol{x}_i - \bar{\boldsymbol{x}}_i)^2} \tag{11}$$

$$\boldsymbol{RMSE} = \sqrt{\frac{\boldsymbol{1}}{\boldsymbol{n}} \sum_{i=1}^{n} (\boldsymbol{x}_i - \hat{\boldsymbol{x}}_i)^2} \tag{12}$$

For the R^2 metric, the results range from 0 to 1. $R^2=0$ indicates that the model does not predict the output, and $R^2=1$ that the model predicts the output perfectly. Regarding the RMSE, the smaller the error value, the better the fit of the model to the data set.

4. RESULTS

This section describes the implementation results of the approximative filter with ANNs for the Moving-Horizon State Estimation Method. The data acquisition was done by the implementation of the MHSE method by (Alvim et al., 2022), and was obtained through the simulation of a dataset of 24,000 samples, collected with a sampling period of 100 ms and a Horizon length $N = 10$.

Two NN were created, a regression NN for state estimation and a classification NN for Mode Estimation. The NN input parameters were the set of inputs U of the estimation window, the set of measured outputs Y within the estimation window, the states $\hat{x}$, and modes $\hat{\sigma}$ estimated in the final of the window. Both NN training were performed offline with the MATLAB® Neural Networks toolbox.

The NN architecture is composed of 2 hidden layers and 10 neurons, and the tests were performed for different amounts of epochs. The purpose of varying the epochs is to spend more time training a network that can estimate the parameters well and can be tested later with noisy signals. Figures 3 and 4 present the input signal used for the MHSE simulation and the input Mode used to train the NN.

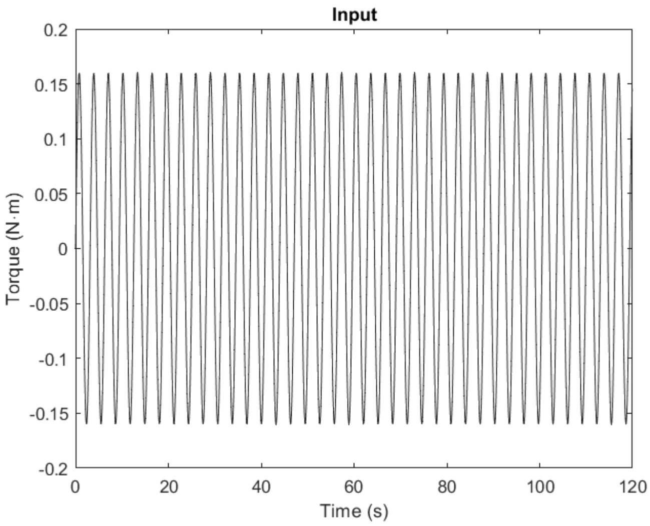

Figure 3. MHSE Input signal.

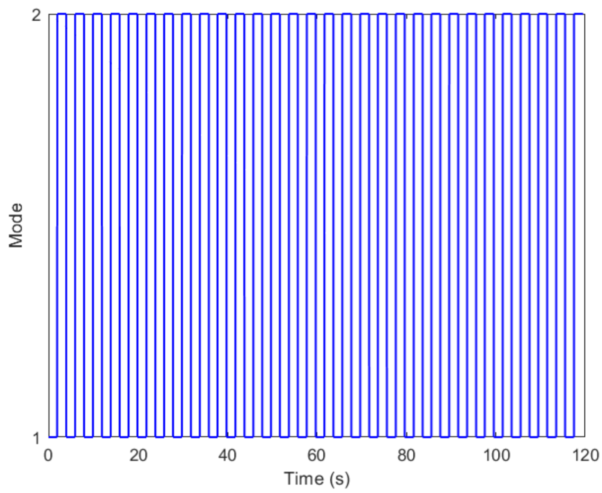

Figure 4. NNMHSE Input Mode

To verify the accuracy of the adjustment of the Network to the estimation method, the coefficient of determination (R^2) and the Root Mean Squared Error (RMSE) is calculated. Table 2 presents the NN training time for each epoch and the RMSE values for each of the states numbered (1-4), being: 1 and 3 – Angular position and 2 and 4 – angular velocity, for the motor and link, respectively.

Table 1. RMSE for the States Approximation

Epochs	**Time(s)**	**RMSE1**	**RMSE2**	**RMSE3**	**RMSE4**
10	23.1589	4.53E-04	2.22E-03	9.74E-04	1.57E-03
20	34.3695	1.13E-03	1.24E-03	6.85E-04	1.08E-03
30	49.9881	5.23E-04	4.99E-04	5.32E-04	5.72E-04
40	63.7439	5.48E-04	6.21E-04	5.65E-04	5.78E-04
50	82.0597	2.16E-03	1.88E-03	1.93E-03	2.30E-03
60	105.3591	2.15E-04	2.32E-04	2.01E-04	3.27E-04
70	120.5277	5.37E-04	5.65E-04	5.35E-04	6.22E-04
80	130.2284	1.30E-03	1.35E-03	1.03E-03	1.26E-03
90	147.3986	4.03E-05	3.57E-05	4.35E-05	4.43E-05
100	202.9350	3.16E-04	2.35E-04	2.90E-04	3.53E-04

The low RMSE values for state estimation indicate a good model fit of the network to the data set. However, as the training and validation data belong to the same simulation and are in the same operating range, it is necessary to validate the network with other data sets.

The R^2 graph was generated for each of the states of the systems, that is, the motor angular position of the (θ_m) and link angular position (θ_l), the motor angular velocity ($\dot{\theta}_m$) and link angular velocity ($\dot{\theta}_l$), and for each one of the epoch variations (10 - 100). The left-hand side of Figure 5 presents the regression for the 10-epochs network that showed the worst R^2 values. For the networks of 20 to 100 epochs, the R^2 values are equal to 1, and the regression graphs present slight variations in the estimated values. Therefore, the right-hand side of Figure 5 presents the adjustment between the predicted values and those estimated by the network only for one of the cases in which the network has 20 epochs. After analyzing the adjustment validation metrics, the NN with 10 neurons and 20 epochs is chosen to demonstrate state estimation.

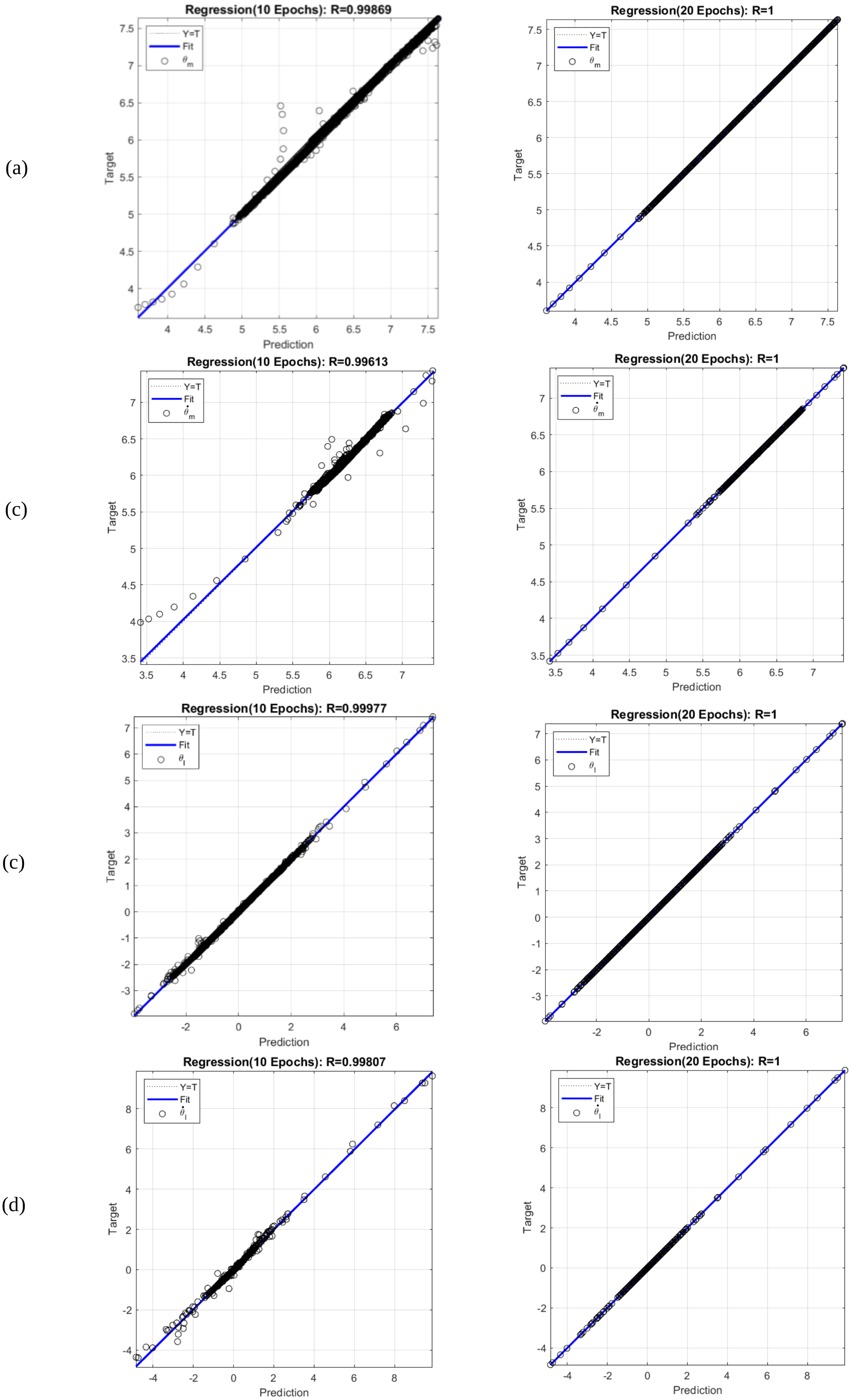

Figure 5. R² metric for the (a) Motor Position, (b) Motor Velocity, (c) Link Position, (d) Link Velocity.

Figures 6 and 7 present the comparison of the results between the MHSE method and the NNMHSE filter for the state's estimation. The graph shows only 12 seconds of the 120 seconds of simulation for better data visualization, considering that are many samples for training the network. It is possible to observe that the NNMHSE effectively estimates the states. The variations that occur in the signal every 2s represent the change in the manipulator's operation modes, always starting in the free operation mode and then passing to the contact operation mode.

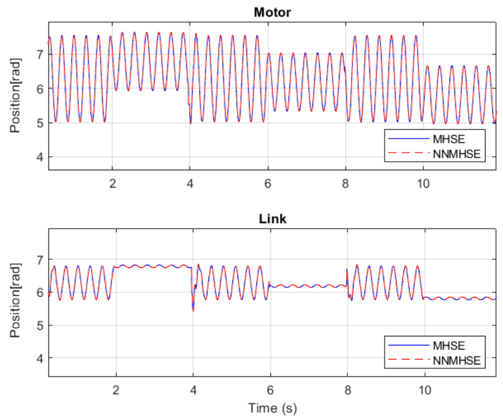

Figure 6. Comparison between the MHSE and the NNMHSE methods for the Position Estimation.

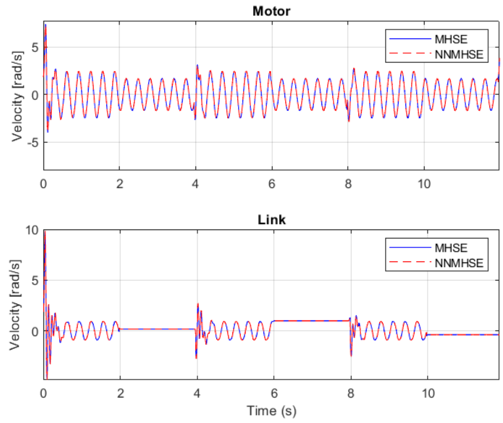

Figure 7. Comparison between the MHSE and the NNMHSE methods for the Velocity Estimation.

Table 2 presents the network training time for estimating the mode with 10 neurons and 2 hidden layers. To verify the network adjustment for mode estimation R^2 value and the RMSE are also calculated for different epoch values. It is possible to observe that all estimates have a high R^2 value, above 0.9.

Table 2 – R^2 and RMSE for the Mode Approximation

Epochs	**Time(s)**	**R^2**	**RMSE**
10	5.9454	0.9468	1.09E-01
20	10.8720	0.9798	6.84E-02
30	13.3603	0.9935	3.99E-02
40	17.0577	0.9995	1.07E-02
50	20.4367	0.9998	7.30E-03
60	24.4084	0.9986	1.89E-02
70	28.3328	0.9990	1.57E-02
80	31.4428	1.0000	2.80E-03
90	34.6771	1.0000	3.90E-03
100	38.5203	1.0000	1.30E-03

Figures 8 and 9 present the R^2 and RMSE value variation curve for each classification NN for mode estimation.

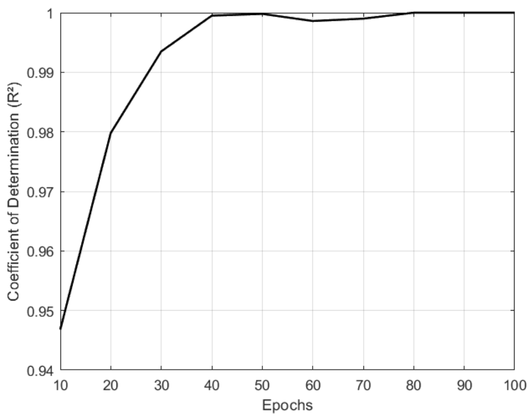

Figure 8. R^2 for the Mode Approximation

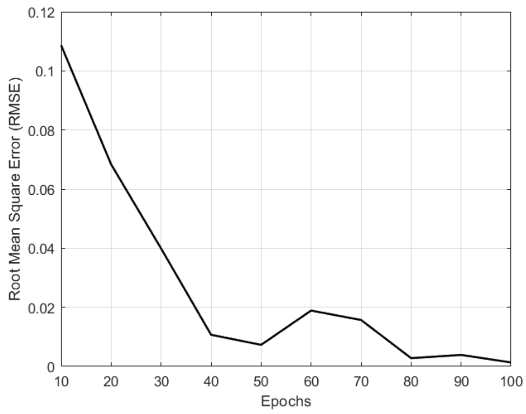

Figure 9. RMSE for the Mode Approximation

From 80 epochs upwards, the NN presents an almost constant adjustment for the value of R^2 with a value equal to 1. The results of the mode classification with NN compared with the MHSE estimation is presented in Figure 10, for the NN of 100 Epochs. Mode 1 is equivalent to the manipulator in free movement, and mode 2 represents the manipulator in contact. Because it is an approximation, it is possible to observe that the mode estimation presents classification failures, that is, instead of the network finding only two values (1 and 2) referring to the system modes the network finds approximate values, nevertheless, the network efficiently infers the modes.

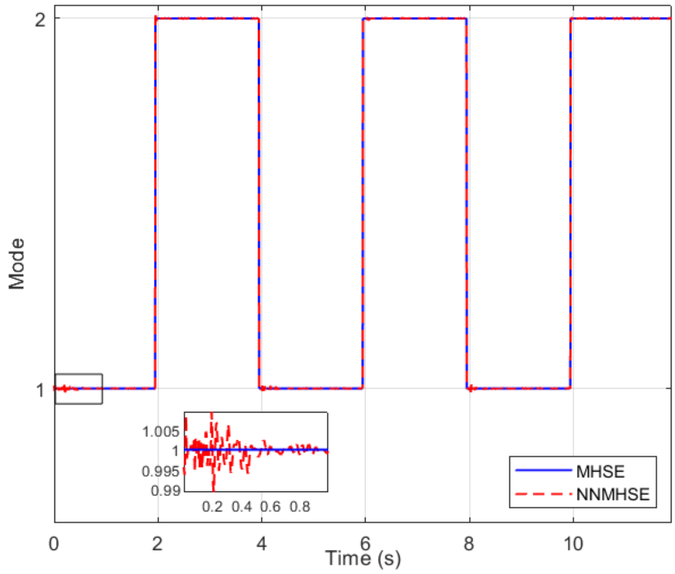

Figure 10. Comparison between the estimation with the MHSE and the NNMHSE

However, the main gain of applying the network is the reduction in the processing time of the estimation. Figure 11 compares the processing time per sample of the MHSE and the NNMHSE and shows that NNMHSE is almost four times faster on average than the MHSE.

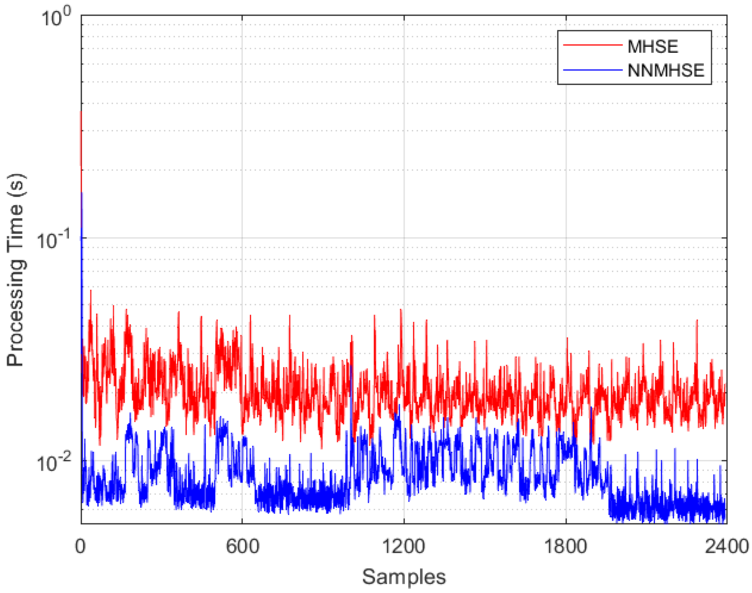

Figure 11. Comparison between the MHSE and NNMHSE processing time.

5. CONCLUSION

This paper proposes the application of a switching approximative filter for the MHSE method based on a feedforward NN trained by the Levenberg-Marquardt algorithm on MATLAB® for contact detection in a 1-DOF robotic manipulator. Two NN architectures were trained, one for state regression and another for operating mode classification. Both NN approximates the estimation model presenting good results for the states and mode estimation, with low estimation error values, an average fit above 90% for the R^2, and a reduction in the average processing time to four times faster than the

MHSE processing time. Despite the promising results presented, the NNs need further investigations by implementing different types of NN and datasets. So, in the future, we expect to evaluate the effectiveness of the proposed approach by testing estimation with noise, with fewer outputs, comparing the method with other filters and estimators, and for manipulators with more degrees of freedom.

6. ACKNOWLEDGMENT

The authors gratefully acknowledge that this study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001, and by the Mechanical Engineering Department of the Pontifical Catholic University of Rio de Janeiro.

7. REFERENCES

Alessandri, A., Baglietto, M., Battistelli, G., & Gaggero, M. (2011). Moving-horizon state estimation for nonlinear systems using neural networks. IEEE Transactions on Neural Networks, 22(5), 768-780.

Alvim, L. C., Pereira, L. D., Lopes, E. D. R., & Ayala, H. V. H. A moving horizon state-estimation approach for switching system of a Single-Link Flexible Joint Manipulator. In: Congresso Brasileiro de Automática, 24., 2022, Fortaleza.

Baglietto, M., Battistelli, G., Ayala, H. V. H., & Tesi, P. Mode-observability conditions for linear and nonlinear systems. In 2012 IEEE 51st IEEE Conference on Decision and Control (CDC) (pp.1941-1947). IEEE. 2012.

Bi, Z. M., Luo, C., Miao, Z., Zhang, B., Zhang, W. J., & Wang, L. (2021). Safety assurance mechanisms of collaborative robotic systems in manufacturing. Robotics and Computer-Integrated Manufacturing, 67, 102022.

Briquet-Kerestedjian, N., Wahrburg, A., Grossard, M., Makarov, M., & Rodriguez-Ayerbe, P. (2019, June). Using neural networks for classifying human-robot contact situations. In 2019 18th European Control Conference (ECC) (pp. 3279-3285). IEEE.

Brunello, R. K. V., Sampaio, R. C., Llanos, C. H., dos Santos Coelho, L., & Ayala, H. V. H. (2020). Efficient Hardware Implementation of Nonlinear Moving-horizon State Estimation with Artificial Neural Networks. IFAC-PapersOnLine, 53(2), 7813-7818.

Dong, Y., Ren, T., Hu, K., Wu, D., & Chen, K. (2020). Contact force detection and control for robotic polishing based on joint torque sensors. The International Journal of Advanced Manufacturing Technology, 107, 2745-2756.

Fan, X., & Arcak, M. Observer design for systems with multivariable monotone nonlinearities. Systems & Control Letters, 50(4), 319-330, 2003.

Jiang, Y., Yang, C., Na, J., Li, G., Li, Y., & Zhong, J. (2017). A brief review of neural networks based learning and control and their applications for robots. Complexity, 2017.

Liberzon, D. Switching in Systems and Control. Birkhauser, 2003.

Ray, S. "A Quick Review of Machine Learning Algorithms," 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), Faridabad, India, 2019, pp. 35-39, doi: 10.1109/COMITCon.2019.8862451.

Magrini, E., Ferraguti, F., Ronga, A. J., Pini, F., De Luca, A., & Leali, F. (2020). Human-robot coexistence and interaction in open industrial cells. Robotics and Computer-Integrated Manufacturing, 61, 101846.

Park, K. M., Park, Y., Yoon, S., & Park, F. C. (2021). Collision detection for robot manipulators using unsupervised anomaly detection algorithms. IEEE/ASME Transactions on Mechatronics, 27(5), 2841-2851.

Sharkawy, AN., Koustoumpardis, P.N., Aspragathos, N.A. (2019). Manipulator Collision Detection and Collided Link Identification Based on Neural Networks. In: Aspragathos, N., Koustoumpardis, P., Moulianitis, V. (eds) Advances in Service and Industrial Robotics. RAAD 2018. Mechanisms and Machine Science, vol 67. Springer, Cham. https://doi.org/10.1007/978-3-030-00232-9_1

Sharkawy, A., Koustoumpardis, P., & Aspragathos, N. (2020). Neural Network Design for Manipulator Collision Detection Based Only on the Joint Position Sensors. Robotica, 38(10), 1737-1755. doi:10.1017/S0263574719000985

Sharkawy, A. N., & Mostfa, A. A. (2021). Neural networks' design and training for safe human-robot cooperation. Journal of King Saud University-Engineering Sciences. 34(8):582–596.

Spiess, AN., Neumeyer, N. An evaluation of R2 as an inadequate measure for nonlinear models in pharmacological and biochemical research: a Monte Carlo approach. BMC Pharmacol 10, 6 (2010). https://doi.org/10.1186/1471-2210-10-6

Zacharaki, A., Kostavelis, I., Gasteratos, A., & Dokas, I. (2020). Safety bounds in human robot interaction: A survey. Safety science, 127, 104667.

Zhang, X., Polycarpou, M. M., Parisini, T. Fault diagnosis of a class of nonlinear uncertain systems with Lipschitz nonlinearities using adaptive estimation. Automatica, v. 46, n. 2, p. 290-299, 2010.

8. RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.