Journal of Information Systems Engineering and Management
2024, 9(3)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

NN-MPC based Multi-Agent Systems based on FITNET Neural Network Architecture

Piyush Chaubey[1*], Anilkumar Markana[2], Dhaval R Vyas[3], and Deepak Kumar Goyal[4]

[1]PhD student, Electrical Engineering Department, Pandit Deendayal Energy University, Gandhinagar Gujarat, India and Assistant Teaching Associate, Electrical Engineering Department, Government Engineering College, Bharatpur, Rajasthan, India. piyush.cphd21@sot.pdpu.ac.in*
[2]PhD Supervisor & Assistant Professor, Electrical Engineering Department, Pandit Deendayal Energy University, Gandhinagar Gujarat, India. anil.markana@spt.pdpu.ac.in
[3]PhD student, Electrical Engineering Department, Pandit Deendayal Energy University, Gandhinagar Gujarat, India. dhaval.vphd18@sot.pdpu.ac.in
[4]Associate Professor & Head, Electrical Engineering Department, Government Engineering College, Bharatpur Rajasthan, India, dkg_elect@rediffmail.com

ARTICLE INFO

Received: 18 Aug 2024

Accepted: 26 Nov 2024

ABSTRACT

This paper presents a comparative study of FITNET and CASCADE Neural Network-based Model Predictive Control (NN-MPC) for Multi-Agent Systems (MAS). The NN-MPC framework is applied to the leader-follower consensus problem, where agents coordinate to achieve collective behavior. In this approach, the MPC is utilized to predict the future values the control objective which is optimized by minimizing a cost function with various neural networks architectures. A comparative analysis is conducted based on varying training strategies, and the effectiveness of both the configurations are evaluated through simulation studies on a quadrotor fleet. The results demonstrate that FITNET NN based MPC significantly enhances the performance and consensus behavior of multi-agent systems.

Keywords: Model Predictive Control, FITNET, CASCADE Neural Network Architectures, Multiagent Systems, Leader- Follower, Consensus.

1. Introduction

Multi-agent systems (MAS) are gaining prominence across various domains, including robotics, distributed control, telecommunications, collaborative decision support, and economics [1]. MAS presents unique challenges due to the need for coordination and cooperation among multiple agents, often in dynamic and uncertain environments [2]. Model Predictive Control has emerged as a powerful framework for controlling complex systems, offering advantages such as constraint handling, disturbance rejection, and trajectory optimization [3].

The integration of learning techniques with MPC has opened new avenues for enhancing the performance and adaptability of multi-agent systems. Integrating learning techniques with MPC offers the potential to significantly enhance the performance of multi-agent systems, enabling adaptation to dynamic environments, learning from past experiences, and refining decision-making processes [4]. MPC has been used to solve a various challenge of the control problems, including autonomous driving [5], power system stability control [6], batch process control [7], [8] and chemical process control [9].

Neural network-based learning enhances MPC by adapting the system dynamics, cost functions, or constraint sets based on data [10]. The shallow neural network involves learning intricate functions, presenting inherent limitations when contrasted with deep architectures [11]. There are various architectures for shallow neural networks with distinguished characteristics. Fit Net neural network structure is the basic one of the feed-forward neural networks, in which activity is propagated unidirectional layer-by-layer from the input up to the output stage, with no feedback connections within or between layers [12].

Another neural network structure can be cascade-forward network, in which each layer receives input from all previous layers [13]. In cascade network structure, unlike standard feed-forward networks where only the first layer

Journal of Information Systems Engineering and Management
2024, 9(3)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

directly receives the input, in cascade networks, every layer receives the input directly, facilitating the learning of more intricate and hierarchical representations of the input data [14].

The paper contributes as the following:

– This work contributes to the comparative analysis of different FITNET NN based MPC for multi-agents with CASCADE NET based MPC to find optimal solution to the consensus problem of leader-follower system.

– The results are validated using mean square error (MSE) for both FITNET and CASCADE structure. The outcome with the best training function is presented with trade-off between fast response and least error performance.

2. Problem Formulation

This study addresses the consensus problem in cooperative Multi-Agent Systems (MAS), where the goal is for all agents to reach a common state through local interactions over a communication network [15–17]. The communication among agents is modeled using graph theory.

Let $G = (T, D, A)$ be a directed graph representing the communication topology, where $T = \{t_1, t_2, . . . , t_n\}$ is the set of nodes (agents), and $D \subseteq T \times T$ denotes the set of directed edges. The weighted adjacency matrix is $A = [a_{ij}] \in R^{n\times n}$, and the Laplacian matrix is defined as $L = [l_{ij}]$, where

$$l_{ii} = \sum_{j\in N_i} a_{ij}, \ , \ l_{ij} = -a_{ij} \text{ for } i \neq j.$$

To account for switching topologies, we model $G_{rr}(\theta) = (T(\theta), D(\theta), A(\theta))$, where the mode $\theta(k) \in S = \{1, 2, . . . , q\}$ is governed by a homogeneous Markov chain [20].

Each agent is modeled with a first-order integrator system:

$$\dot{x}_i(t) = u_i(t) \tag{1}$$

where $x_i(t)$ is the state and $u_i(t)$ is the control input.

The consensus control law is defined as:

$$u(k) = (L(\theta(k) * K_k)\, x(k) \tag{2}$$

where K_k is the control gain matrix, and $x(k)$ is the collective state vector of all agents.

To achieve this control objective, we employ a FITNET Neural Network-based Model Predictive Control (NN-MPC) strategy, illustrated in Fig. 1. The MPC layer predicts system behavior and computes a cost function, while the NN layer learns to optimize the cost by adjusting K_k using inputs $x(k)$ and $\theta(k)$.

In a more general setting, the agent dynamics can include non-linearity and external disturbances:

$$\dot{x}_i(t) = A\, x_i(t) + Bu_i(t) + f\,(t, x_i(t)) + B_d\, d_i(t) \tag{3}$$

where $f(t, x_i(t))$ models nonlinear internal dynamics, and $d_i(t)$ is the disturbance input.

The proposed approach is simulated using MATLAB 2022, and the performance is analyzed for various NN training algorithms to achieve consensus in a fleet of quadrotors [18].

Fig. 1 shows the block diagram of FITNET based NN-MPC for multi-agent system. In which it is clearly shown that the value of control gain K_k is found using FITNET based optimization and MPC prediction.

Journal of Information Systems Engineering and Management
2024, 9(3)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

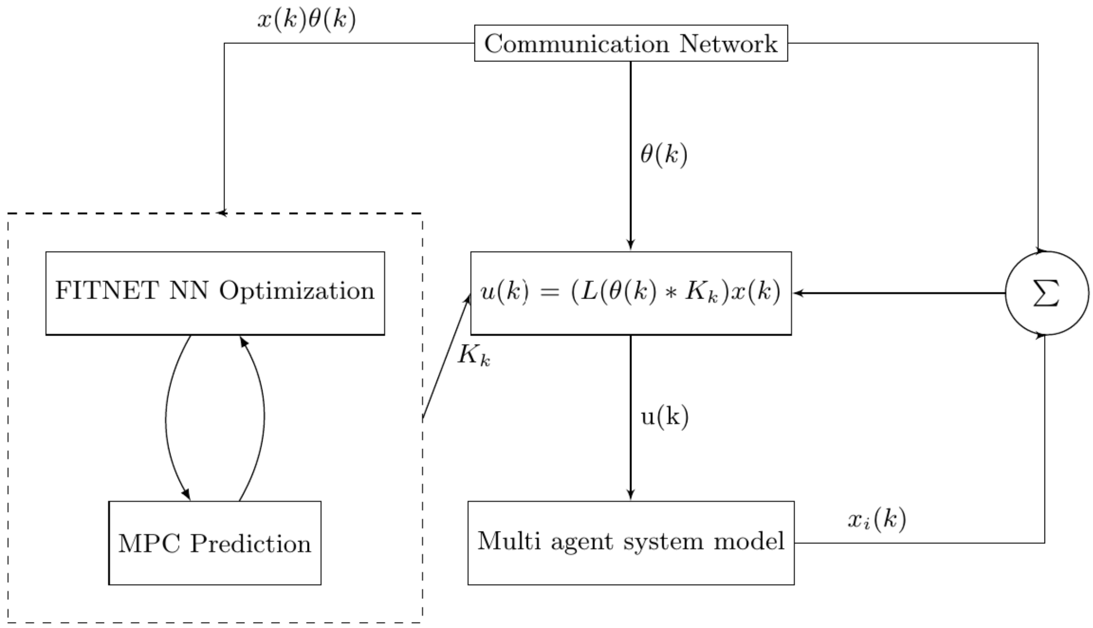


Fig. 1. Block diagram of FITNET NN-based MPC for Multi-Agent Systems [15, 21]

3. FITNET Neural Network Optimization based MPC for Multi-Agent System

This section outlines the integration of Model Predictive Control (MPC) with FITNET and CASCADE Neural Network (NN)-based optimization for achieving consensus in a Multi-Agent System (MAS).

Consider a MAS where the objective is for follower agents to track the state of a leader. The consensus objective can be expressed in terms of the state error between the i^{th} follower and the leader agent:

$$E(k) = x_i(k) - x_0(k) \tag{4}$$

where $x_i(k)$ and $x_0(k)$ are the states of the i^{th} follower and the leader at time step k, respectively.

To guide the system towards consensus, we define a quadratic cost function associated with the error state:

$$V(E(t), \theta(t)) = E^T(t) P_p E(t), \tag{5}$$

where P_p is a positive-definite matrix and $\theta(t)$ denotes the communication topology mode at time t. The predicted cost function over a future horizon h is given by:

$$J(E(k)) = \sum_{t=k+1}^{k+h+1} \mathrm{E}_{x k|k} \left[V(E(t), \theta(t))\right] \tag{6}$$

To minimize this cost, the optimal control gain matrix is computed by:

$$K_k = \arg \min_{K_k, \ldots, K_{k+h}} J(E(k)). \tag{7}$$

This optimization is handled by an NN-based learning mechanism that updates the gain K_k by training the network to reduce the predicted cost $J(E(k))$ over iterations. This combined NN-MPC framework enables the multi-agent system to achieve consensus effectively under dynamic network topologies.

4. FITNET AND CASCADE NET Neural Architectures

4.1 Fit-Net Architecture

Fig. 2. shows the architecture of Fit Net neural network. Fit Net is a simple feed-forward neural network with one hidden layer [19]. It's commonly used for function approximation and regression tasks.

$$J = y = f(W_2 \cdot f(W_1 x + b_1) + b_2) \quad (8)$$

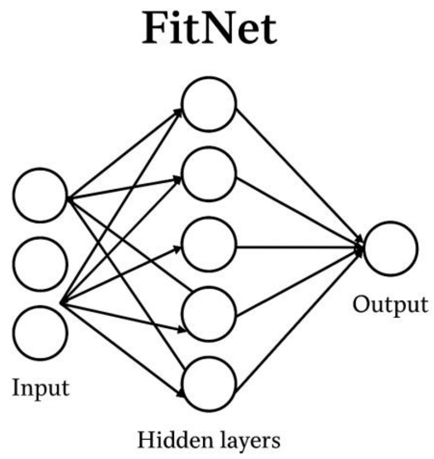


Fig. 2. Architecture of Fit Net Neural Network

4.2 Cascade-Forward Net Architecture

Fig. 3 shows the architecture of recurrent neural network. A Cascade-Forward Net allows connections from the input layer to all subsequent layers, including to the output directly, enhancing learning flexibility [20].

$$J = y = f(W_3 \cdot f(W_2 \cdot f(W_1 x + b_1) + W_4 x + b_2) + b_3) \quad (9)$$

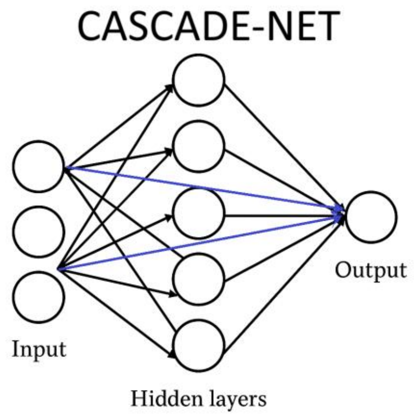


Fig. 3. Architecture of Cascade Neural Network

Journal of Information Systems Engineering and Management
2024, 9(3)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

where, h_t is the hidden state of the NN, x_t is the input, y_t is the output at time step t. W_h, W_x & W_y are weight matrices that are learned during training, b and c are bias vectors that are learned during training, f and g are sigmoid activation functions applied element-wise to the weighted inputs.

There is a feedback connection from the hidden node h to itself, which allows the network to maintain a memory of previous inputs. The weights W_h, W_y and W_z are adjusted to obtain the optimized output for a given set of input data.

The input x_t to Neural Network can be given as optimal control gain matrix K_k,

$$x(t) = [1\ K^{11} \ldots K^{1n} \ldots K^{m1} \ldots K^{pq}]' \quad (10)$$

Following algorithm shows and explains how the optimization is achieved with various training functions for NN based model predictive control. Firstly, the system x (0) and the mode θ (0) are initialized and the gain K_k is set to zero. NN training starts with the initial conditions to compute the cost function J given by the (8 - 9) and then the control gain matrix U is calculated (2).

To compute the future predicted cost J mentioned in (6), the cost function V given by (5) is calculated over the prediction horizon and all the values of V are summed to give the prediction cost. The values of J and K_k are updated and stored for next iteration and the process is repeated until the cost satisfy the criteria γ set for the optimum value of the cost function.

5. Simulation study

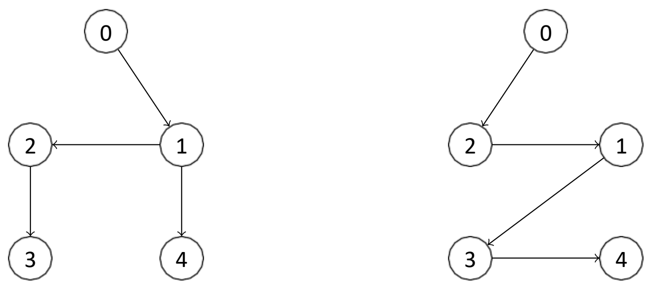
The dynamics of quadrotor fleet described in [18] is used for the simulation purpose and the data used for the simulation is referred from [15], [21]. In which, d is the disturbance for the ith agent.

Let $f(t, x_i(t)) = 0.01 \sin(x_i(t))$, $A = \begin{bmatrix} 0 & 1 \\ 0 & -0.5 \end{bmatrix}$, $B = \begin{bmatrix} 0.8 \\ 1.2 \end{bmatrix}$, $B_d = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$, The initial values are same as considered in [21].

The Laplacian matrices are defined as $L(1) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ -1 & 0 & 0 & 1 \end{bmatrix}$ and $L(2) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ -1 & 0 & 0 & 1 \end{bmatrix}$

The probability transition matrix π used is $\pi = \begin{bmatrix} 0.95 & 0.05 \\ 0.02 & 0.98 \end{bmatrix}$

Fig. 4 shows the directed graph used corresponding to $L(1)$ and $L(2)$. Discretization of the continuous time system is done at sampling period $T_s = 0.01$.



a) L(1) directed graph **b) L(2) directed graph**

Fig. 4. Directed graph topology

The simulation is done hereby with N = 10 neurons, and MPC prediction horizon is chosen to be 100 for the best results in both the feedforward neural network optimization and recurrent neural network optimization which is considered as sufficient to obtain the optimal point. The convergence point can be achieved for $\lambda = 0.01$. The

Journal of Information Systems Engineering and Management
2024, 9(3)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

measurement parameter considered for the validation is mean-squared error of the *r*-th state of the *i*-th agent (MSE_{ri}) given by

$$MSE = \frac{1}{k_f} \sum_{k=0}^{k_f} |e_{ri}(k)|^2 \tag{11}$$

where k_f is the value at end of the time of convergence and $e_{ri}(k)$ is the *r*-th value of the error $e_i(k)$.

6. Result analysis and discussion

Fig. 5 shows the result of NN based MPC with the Fit Naet architecture. It reflects that all the agents achieve consensus at 3 sec for 'position state' whereas the same is achieved after 4 sec for the 'velocity state'.

Fig. 6 shows the result of NN based MPC with the cascade network architecture. It reflects that all the agents achieve consensus at 4 sec for 'position state' and at 5 sec for the 'velocity state'.

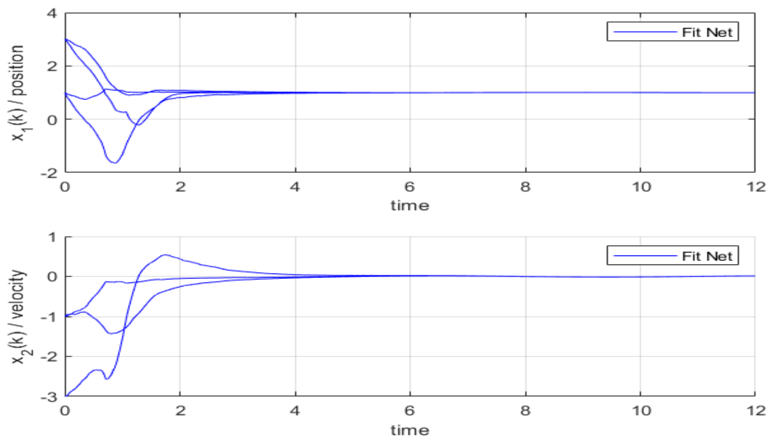


Fig. 5. Position and velocity response of the followers with Fit Net architecture

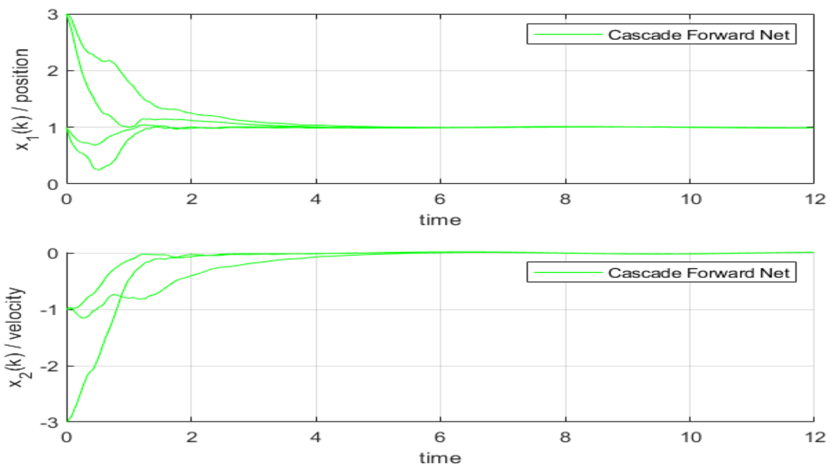


Fig. 6. Position and velocity response of the followers with Cascade Forward Net architecture

Table 1 shows the comparison of mean squared errors for the 'position state' and 'velocity state' of various neural network architectures for achieving consensus of follower agents. It can be observed that the least error is found with three cases as Fit Net and CFN for 'position state' and 'velocity state'. MSE is least for both 'position state' and 'velocity state' error for the followers '2' if Fit Net architecture is used neural network structure. In other cases, like

'Position state' of follower '4' and 'velocity state' of follower '1' and '4' the MSE is minimum with CFN architecture. However, the negotiation on the MSE for the 'position state' and 'velocity state' can be accepted to an extent for the nature and speed of response among different training network architectures.

Table 1. Mean squared error comparison of FITNET and CASCADE Neural Network based MPC for Mult-agent system

(i_{th} agent,r_{th} state)	Fit Net	Cascade Forward Net (CFN)	Least MSE
(1,1)	0.0866	0.1028	Fit Net
(1,2)	0.0967	0.0740	CFN
(2,1)	0.0010	0.0022	Fit Net
(2,2)	0.0204	0.0237	Fit Net
(3,2)	0.3387	0.1978	CFN
(4,1)	0.0826	0.0377	CFN
(4,2)	0.3825	0.3324	CFN

Based on the observations and comparison made in this study, it is evident that the RNN based MPC demonstrates successful achievement of consensus in the minimum time thereby showing least MSE for the 'position state' of follower '1'and 'velocity state' of follower '3'. However, no single training function provides least MSE for all agents for the 'position state' and 'velocity state'.

7. Conclusion

In conclusion, the findings of this study reveals that the NN-based model predictive controller (MPC) based on FITNET gives better results in terms of agents achieving consensus in minimum time elapsed as compared to CASCADE neural network architectures. Also, there is substantial amount of percentage decrease in the MSE for the 'position state' of follower '1 & 2'and 'velocity state' of follower '2' that validates the effectiveness of the recurrent neural network-based training network. However, there is no architecture found that provides least MSE for all agents for the 'position state' and 'velocity state'. Future research efforts should be focused to find such a learning-based MPC that can provide fast response as well as least MSE for most of the agents to achieve the consensus of leader- follower system.

References

[1] Khalil, Khaled M., Mohamed Abdel-Aziz, Taymour T. Nazmy, and Abdel-Badeeh M. Salem, "Machine learning algorithms for multi-agent systems." In Proceedings of the *International Conference on Intelligent Information Processing, Security and Advanced Communication*, pp. 1-5. 2015.

[2] Canese, Lorenzo, Gian Carlo Cardarilli, Luca Di Nunzio, Rocco Fazzolari, Daniele Giardino, Marco Re, and Sergio Spano`, "Multi-agent reinforcement learning: A review of challenges and applications." *Applied Sciences* 11, no. 11 (2021): 4948.

[3] Mayne, David Q., "Model predictive control: Recent developments and future promise." *Automatica* 50, no. 12 (2014): 2967-2986.

[4] Norouzi, Armin, Hamed Heidarifar, Hoseinali Borhan, Mahdi Shahbakhti, and Charles Robert Koch, "Integrating machine learning and model predictive control for automotive applications: A review and future directions." *Engineering Applications of Artificial Intelligence* 120 (2023): 105878.

[5] Jin, X., J. Yang, and J. Wang. "MPC-based trajectory tracking control for autonomous vehicles: a survey." *IEEE Trans. Intell. Vehic* 4, no. 1 (2019): 39-56.

[6] M. Piyush, P. Vivek and M. Anilkumar, "Model Predictive Control of Series Compensation for Transient Stability Enhancement of SMIB System", *Journal of Energy and Management*, vol 1, no.2, (2017).

[7] M, Anilkumar, N. Padhiyar, K. Moudgalya. "Multi-objective prioritized control of a semi-batch process with multiple feed and multiple products using economic MPC", *Indian Control Conference (ICC)*, (2018).

2024, 9(3)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

[8] M, Anilkumar, N. Padhiyar, K. Moudgalya, "Multi-criterion control of a bioprocess in fed-batch reactor using EKF based economic model predictive control", *Chemical Engineering Research and Design*, (2018).

[9] Maciejowski, J. M., "Predictive Control with Constraints" *Pearson Education Limited*, 2002.

[10] Ashoori, Ahmad, Behzad Moshiri, Ali Khaki-Sedigh, and Mohammad Reza Bakhtiari, "Optimal control of a nonlinear fed-batch fermentation process using model predictive approach." *Journal of Process Control* 19, no. 7 (2009): 1162-1173.

[11] Bengio, Yoshua, "On the challenge of learning complex functions." *Progress in Brain Research* 165 (2007): 521- 534.

[12] Myung, Hwan-Chun, and Z. Zenn Bien, "Design of the fuzzy multi-objective controller based on the eligibility method." *International journal of intelligent systems* 18, no. 5 (2003): 509-528.

[13] Dubreuil-Vall, Laura, Giulio Ruffini, and Joan A. Camprodon, "Deep learning convolutional neural networks discriminate adult ADHD from healthy individuals on the basis of event-related spectral EEG." *Frontiers in neuroscience* 14 (2020): 251.

[14] Zhu, Yulin, Jiang Wang, Huiyan Li, Chen Liu, and Warren M. Grill, "Adaptive parameter modulation of deep brain stimulation based on improved supervisory algorithm." *Frontiers in neuroscience* 15 (2021): 750806.

[15] Bruno R.O. Floriano, Alessandro N. Vargas, Joao Y. Ishihara, Henrique C. Ferreira, "Neural-network-based model predictive control for consensus of nonlinear systems". *Engineering Applications of Artificial Intelligence*, Volume 116, 105327, ISSN 0952-1976, (2022).

[16] Ren, W., Beard, R.W., Atkins, E.M, "Information consensus in multivehicle cooperative control". *IEEE Control Syst. Mag*. 27 (2), 71–82, (2010).

[17] T. Zhang and H. YU, "Average Consensus in Networks of Multi-Agent with Multiple Time-Varying Delays. International Journal of Communications", *Network and System Sciences*, Vol. 3 No. 2, pp. 196-203, (2010) doi: 10.4236/ijcns.2010.32028.

[18] Gao, J., Li, J., Pan, H., Wu, Z., Yin, X., Wang, H, "Consensus via event-triggered strategy of nonlinear multi-agent systems with Markovian switching topologies". *ISA Trans*. 104, 122–129, (2020).

[19] Le Borgne, Yann-A., and Gianluca Bontempi, "Machine learning for credit card fraud detection-practical hand- book." *ACM SIGKDD explorations newsletter* 6, no. 1 (2004): 1-6.

[20] Aitken, Kyle, and Stefan Mihalas, "Neural population dynamics of computing with synaptic modulations." *Elife* 12 (2023): e83035.

[21] Chaubey, Piyush, Anilkumar Markana and Dhaval R. Vyas, "RNN-Based Model Predictive Control of Multi-agent System Using Switching Topologies." *In International Conference on Data Science and Applications*, pp. 157-168. Singapore: Springer Nature Singapore, 2023.