Proc. of the 3rd International Conference on Electrical, Communication and Computer Engineering (ICECCE)
12-13 June 2021, Kuala Lumpur, Malaysia

2021 International Conference on Electrical, Communication, and Computer Engineering (ICECCE) | 978-1-6654-3897-1/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICECCE52056.2021.9514097

Using NN-DEVS Approach for Modelling and Simulation of Imperfect Systems: Application to the Reactive Navigation of Autonomous Robot

Kadda Mostefaoui
Ecole nationale Supérieure d'Informatique BP 68M, 16309 Oued-Smar,
Alger, Algérie
k_mostefaoui@esi.dz
https://orcid.org/0000-0002-6367-1245

Youcef Dahmani
Department of Computer Science University of Tiaret
Tiaret, Algeria
dahmani_y@yahoo.fr
https://orcid.org/0000-0001-6528-1825

Bendaoud Mebarek
Department of Computer Science University of Tiaret
Tiaret, Algeria
mebarekbendaoud@yahoo.fr
https://orcid.org/0000-0002-6838-3867

Mohamed Goucem
Department of Computer Science University of Tiaret
Tiaret, Algeria
goucemohamedinginf@gmail.com

***Abstract*—In this paper, we are interested in the modelling and simulation of imperfect systems in the DEVS context. We want to hybrid the DEVS formalism with artificial neural networks and to propose a new modelling and simulation approach which makes it possible to represent the behavior of imperfect systems.**
The problematic of our work is the integration into the DEVS formalism of tools from artificial intelligence allowing the representation, manipulation, and processing of imperfect data (imprecise, uncertain).
NN-DEVS is a new hybrid approach which allows to extend the classic DEVS formalism. This new approach is effective in uncertain systems where the behavior of the system is stochastic.
To validate the proposed NN-DEVS approach, we apply this approach to a complex reactive navigation system of a mobile robot.

Keywords—Modelling; Simulation; DEVS; Artificial Neural Networks; Reactive navigation.

I. Introduction

Simulation is a discipline whose aim is to develop models for an existing or theoretical system, once created this model, it is possible to run and analyze results. It is used in research and industry, whenever the construction of a prototype test is too expensive or that the usual mathematical study is too long to see too complex.

The concepts of simulation and modelling are intrinsically linked, modelling and simulation are used to study the behavior of dynamic systems; in the literature several methods have been proposed for the analysis of the behavior of these systems in a more realistic, simple and very flexible way.

Different research and implementations have been conducted on the theme of modelling and simulation of complex systems. Therefore, today there are a large range of methods, tools and software products in this context .

A modelling approach called DEVS formalism is used as a modelling and simulation tool for many systems, this formalism is widely used to model and simulate dynamic systems with discrete events.. In the mid-seventies, several formal work has developed to provide a rigorous common basis for the modelling and simulation of discrete events [1], this formalism was introduced by Professor BP Zeigler [2].

The approach proposed in this study is a hybrid modelling approach based on systems theory and the notion of state [2]. The notions of modularity, hierarchy, multi-modelling and the capacity of openness provided by the DEVS formalism make it possible to combine in a coherent way formalisms or paradigms based on the general theory of systems and centered on states, These characteristics makes the formalism a solid formal framework for modelling and simulating dynamical systems and thus makes a formalism suitable for many areas of application. [3][4][5][6][20][21].

In the literature there are many approaches to modelling the complexes system, generally the researchers use the method of the Artificial Intelligence like the artificial neural networks (ANN) [15][16], the fuzzy logic (FL) [17], in other researchs they used the combination of the precedent approach to realized the Neural-fuzzy.

The main objective of this study is to propose a new hybrid approach named NN-DEVS based on the DEVS formalism integrated with artificial neural networks, this approach is used to study the behavior of uncertain dynamic systems (stochastic).

In order to evaluate our NN-DEVS approach we applied this approach on a reactive navigation system of a mobile robot.

This article is organized as follows: we start with the presentation of the basic concepts of the DEVS formalism as well as the theory of artificial neural networks. Then we model by our NN-DEVS approach a reactive navigation system of a mobile robot. We then carry out simulations in order to validate our approach. Finally, we conclude our work and give some perspectives of this work.

II. BACKGROUND

In this section we present the DEVS formalism and artificial neural networks (ANN).

A. The DEVS formalism

The DEVS formalism is used to model and simulate discrete systems. DEVS gives a robust model, based on atomic models and on the coupling of higher level models. This allows for hierarchical modeling. DEVS is independent of the implementation of simulators. This formalism is used in different applications such as, in robotics, computer architectures, ecological modeling, manufacturing system, environmental systems and transport systems.

The DEVS Atomic Model

The atomic model autonomously describes the behavior of the system. A DEVS atomic model is based on continuous time, inputs, outputs, states and functions (output, transition, the time advance).Tthe internal transition function that moves the system from one state to another autonomously. It depends on the time spent in the state. When an internal event occurs, the model produces an output. the external transition function allows the reaction of the system to an external event dependent on the current state of the system, the input and the elapsed time in that state.

An atomic model is specified as follows [2]:

$$AM = (X, Y, S, \delta_{ext}, \delta_{int}, \lambda, ta)$$

X : are the input values of the model.

Y : are the output values of the model.

S : are the possible states of the model.

δ_{int} : $S \rightarrow S$: is the internal transition function.

δ_{ext}: $Q \times S \rightarrow S$: is the external transition function.

$Q=\{(s,e) \mid s \in S. 0 \leq e \leq ta(s)\}$: total state set

e: is the time elapsed

λ : the output function.

ta: the time advance function of state $s \in S$.

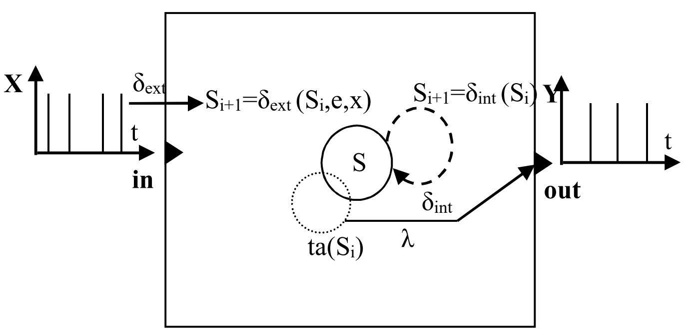

Fig. 1. Semantics of the DEVS atomic model

The DEVS Coupldед Model

The coupled model makes it possible to describe a complex system, this model is formed by the interconnection of several atomic models. In a higher level description this new model is used as a base model, this allows hierarchical modelling. Formally a classical DEVS coupled model is defined as follows [2]:

$$CM = \langle X, Y, D, \{Md \mid d \in D\}, EIC, EOC, IC, Select\rangle$$

X: are the input values of the coupled model.

Y: are the output values of the coupled model.

D: is the set of component names of the coupled model.

M_d: are the component names of the coupled model.

EIC: are the couplings of the external inputs of the coupled model.

EOC: are the couplings of the external outputs of the coupled model.

IC: are the internal couplings of the coupled model which connect component outputs to component inputs.

Select: $2^D \rightarrow D$: is the selection function, this function allows to give priority between the elements two by two, in order to avoid any conflict.

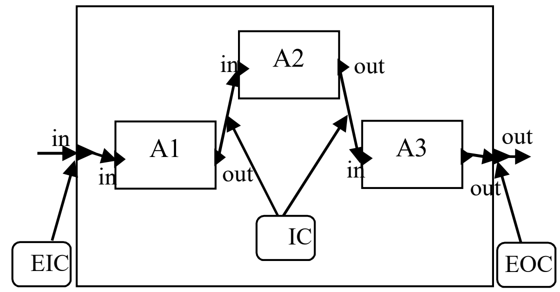

Fig. 2. Description of a DEVS coupled model

B. Artificial Neural Networks

For decades, Artificial Neural Networks (ANNs) have been used in several scientific fields such as medicine, biology, ecology, and economics, etc.). Artificial neural networks are excellent tools for pattern recognition, complex signal processing, classification and prediction [10]. Artificial Neural Networks (ANNs) are models inspired by biology in the form of a mathematical model that mimics the functioning of a biological neuron.

ANNs are alternative models to mathematical models, these models belong to nonparametric and nonlinear statistical models which respond to identification problems., prediction, etc. [18]. A neural network is made up of a set of artificial neurons interconnected by weights whose values influence the behavior of the entire structure. The rules by which the operation of adjusting connections is performed characterize the network learning algorithm.

1) Process of learning

In an artificial neural network, each neuron of a layer is linked with the other neurons of the other layers to solve a well-determined problem on the data supplied at the input of the network. Once the network is formed, inputs are injected and network learning is started to estimate these parameters. There are basically two types of learning, unsupervised learning and supervised learning. For unsupervised learning, examples are presented to the network which is left to organize itself. In the case of supervised learning, we seek to impose a given operation on the network. The network parameters are adjusted from the input / output pairs presented.

Different algorithms, such as back-propagation [7] [8] [9], allow training of the neural network. We try to obtain from the network a response preestablished as being correct. We have a knowledge base of expected input / output type. We then compare the output to the expected output. We introduce an error function that we will try to minimize by modifying the weights of the network.Once the weights defined by this algorithm from examples known to the user, we will try to extrapolate the network, by providing it with unknown inputs.

The basic structure of a back-propagation neural network [13] is shown in Figure. 3. It consists of mathematical modules called neurons (Figure 4)[13]. These neurons are distributed in layer, which means that the input layer is made up of all the inputs in the model. Each of these inputs is transmitted to all neurons in the hidden layer where it is weighted by a multiplicative weight (W_{ij}). A bias (0 or 1) is added to the sum of the weighted inputs to produce an intermediate result modulated by a transfer function, then passed to neurons in the output layer where it is weighted (W_{jk}) and repeated the same operations leading to model outputs (the output layer will contain as many neurons as there are variables to be modeled).

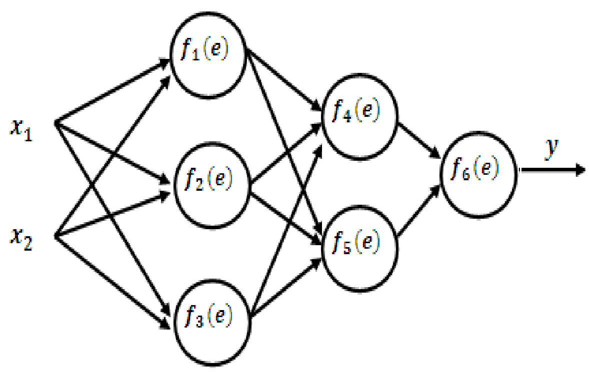

Fig. 3. General structure of a backpropagation neural network[13].

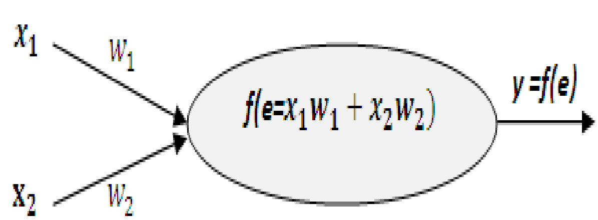

Fig. 4. Typical neuron structure in backpropagation neural network[13].

Among the different architectures of artificial neural networks we have adopted for this study Multi-Layer Perceptrons (MLP) [19]. MLPs are the most used in supervised learning approaches, that is, when an association between two types of data, respectively representing the input and the output of the network, must be learned. The Multi-layer Perceptron is the largest class of neural networks due to the simplicity of its learning algorithm and its aptitude for approximation and generalization [11]. In an MLP artificial neurons are organized in layers. Two extreme layers correspond to the layer which receives the input data, and the layer which provides the result of the processing carried out (output layer). The intermediate layers are called hidden layers, their number is variable. Connectivity between successive layers is total and each connection is weighted

III. Application and test of the NN-DEVS approach

The complexity of artificial systems is growing and the crucial importance of better understanding and mastering the complexity of systems, require the design of new modeling and problem solving methods.

The modelling and simulation of complex systems involves the processing and analysis of information or data, for which the values are often imprecise, uncertain, or incomplete

The NN-DEVS is a new hybrid approach which allows to extend the classical formalism of DEVS. This new approach is effective in uncertain systems where the behavior of the system is stochastic as well in many fields of application. The proposed NN-DEVS approach is a hybrid approach of artificial neural networks with the DEVS formalism which makes it possible to respond to our problem, namely the definition of a method for taking into account the uncertainties linked to transitions between states. The NN-DEVS allows to represnt the stochastic (uncertain) behavior of dynamic systems.

In order to test and apply the NN-DEVS approach, we have chosen as a case the modelling and simulation of a reactive navigation system of a mobile robot. The behavior of autonomous mobile robots is deeply random due to their interaction with complex and unpredictable environments.

The specification of uncertain system with the NN-DEVS approach is similar to the classical DEVS. The only difference exists when the system exhibits probabilistic or stochastic behavior unlike a deterministic system. In this case the future state of the system is not determined with the transition function of the classical DEVS formalism but with the use of a new stochastic function which determines the next most probable state in the future, and which is obtained using the artificial neural networks which detrmines the evolution of the model.

A. The model of the mobile robot system using DEVS

The mobile robot is a complex system which includes: the subsystem localization, the subsystem perception, the subsystem controller ans the subsystem actuator.

The subsystem Localization: this module estimates the current position of the robot which is obtained by information coming from the proprioceptive sensors.

The subsystem perception : This module is very important for the safety of the robot if the environment is cluttered with fixed or mobile obstacles. It provides characteristic measurements of the position that the robot can acquire in its environment by detecting objects that bypass..

The subsystem controller: a mobile robot is controlled by a control module which reads the data received by the sensors, interprets them, calculates the motor commands and sends them to the actuators

The subsystem actuator: In order to move within and interact with its environment, a robot is equipped with actuators. For example, a robot has one or more motors that can turn its wheels in order to perform movements. Usually the robot wheels are controlled by two drive controls

Figure 5 represents the model of a reactive navigation system of a mobile robot using the DEVS formalism.

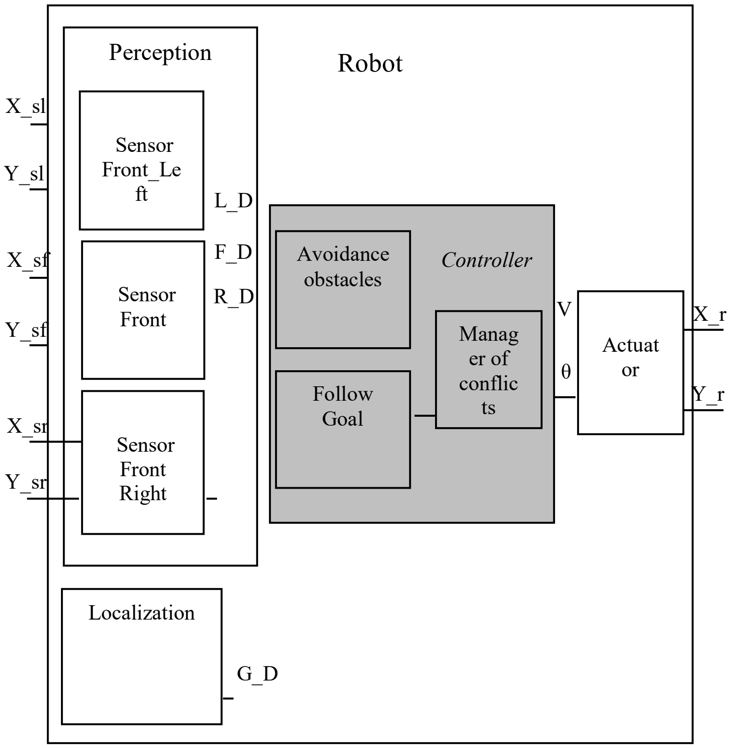

Fig. 5. Model of a reactive navigation system of a mobile robot using DEVS

B. *The NN-DEVS Controller*

Robots therefore need a high level of planning in order to generate the correct action corresponding to the context of the task and the environment to which they are attached.for this purpose, the Artificial Intelligence community is developing several approaches for the description of the action and the generation of plans for the robot

The mobile robot is a physical agent performing tasks in its environment, endowed with the capacities of perception, decision and action. The objective is to allow the robot to react automatically without collision with unforeseen objects, automatically (without human intervention) when the environment becomes more complex (ie partially known or dynamic), For this, the robot must follow the diagram corresponding to the paradigm (Perceive-Decide-Act).

The reactive navigation of an autonomous mobile robot in unfamiliar environments is considered a very difficult task in the field of mobile robotics. This involves controlling the evolution of the robot in environments characterized by their complexity (they can be vast, imprecise, dynamic or unknown, etc.). Which brings us to the definition of certain elementary behaviors such as: convergence towards a goal, avoidance of obstacles, pursuit of a trajectory or of a target, etc.

In this work we are interested in the use of DEVS formalism combined with artificial neural networks. The basic principle of the behavior-based navigation system is to subdivide the overall navigation task into a set of elementary action behaviors (behavior 1, behavior 2,…, behavior n); easy to design and manage. The classical DEVS formalism is a deterministic formalism and it is not suitable for systems whose state transitions are stochastic [22]. The DEVS formalism is a multi-modelling environment, therefore it allows the integration of many other formalisms or modelling methods.

Figure 6 represents the principle design of our controller in DEVS.

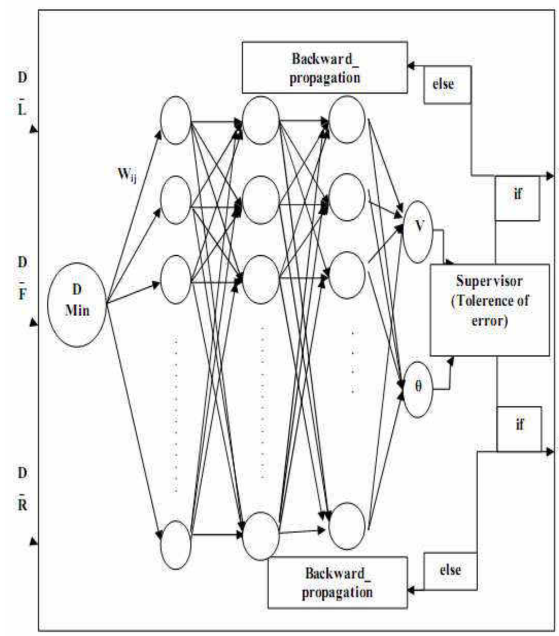

Fig. 6. A Neural Network controller in DEVS

1) *Specification of the controller in NN-DEVS*

The formal specification of the controller using the NN-DEVS approach is as follows:

$$\text{Controller} = (\ X, Y, S, \delta_{int}, \delta_{ext}, \lambda\ , ta\ )$$

Where:

X: D (L_D, F_D, R_D): are the values of the distances sent by by the subsystem perception.

Y: V, Δθ are the values of the speed of movement of the robot respectively the value of the angle of deviation of the robot.

S: “Initial State”, “Stop State” , “Turning right State”, “Turning left State”, “Moving Forward State”.

δ_{int}: The internal transition function.

δ_{ext}: ANN (Distance) neural network controller, receives the value of the input distance (D) from the perception subsystem.

λ: the values calculated by the controller are sent to the subsystem actuator.

ta : function which gives the lifetime of the state S.

2) *The neural network controller*

The neural network controller proposed is a network with five layers where: one layer for the input data with one neuron, and one layer for the output with two neurons (one for the speed of robot and the second for the angle) , and three hidden layers with 15,14,13 neurons respectively.

One used the algorithm of the BackPropagation of the gradient and as supervisor one used the results of a ANN controller. The function of activation of the neuron is the sigmoid function.

Two behaviors are used based on a reactive approach for mobile robot navigation:

Obstacle avoidance behavior and target tracking behavior. The subsystem controller receives as inputs the values of the distances L_D, F_D and R_D sent by the perception subsystem and G_R the angle existing between the target and the robot, subsequently and after a calculation this subsystem returns the value $\Delta\theta$ of the change in direction of the robot and the value ΔV the variation of the speed of the robot at the subsystem actuator.

IV. Simulations and Results

Several examples of mobile robot navigation in indoor environments will be presented to validate the proposed NN-DEVS approach. The environment used takes into account the constraints of modelling and movement of the robot used in several situations such as: free space and the environment with static obstacles.

To specify, design and develop our system, we used the POWERDEVS simulator for the modelling and simulation of the mobile robot's reactive navigation system.

POWERDEVS is a simulator different from other existing simulators. It is a flexible simulator and offers general usage, it provides a high level of DEVS formalism. POWERDEVS is a functional, abstract tool and provides specific components and interfaces, these characteristics qualify it as a domain-specific modeling and simulation tool [12].

Figure.7 represents the implementation on the POWERDEVS environment of the reactive navigation system of a mobile robot. Each subsystem is modeled by a corresponding atomic model.

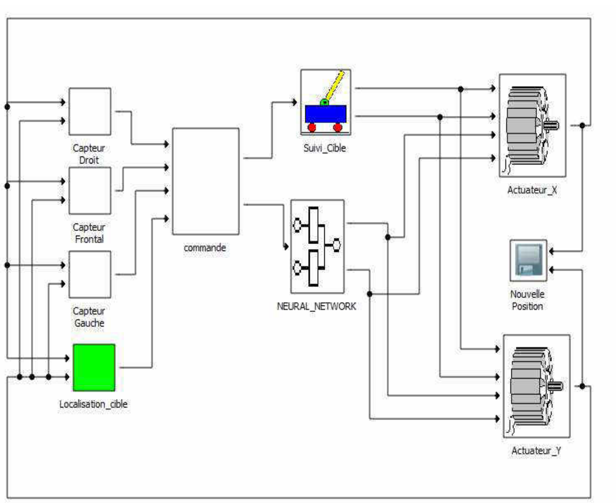

Fig. 7. Representation of the model under POWERDEVS.

In the first test shown in Figure. 8 When the robot's perception sensors do not detect any obstacle near it, the task then becomes a direct orientation towards the target to reach it. It is also called free navigation to a goal.

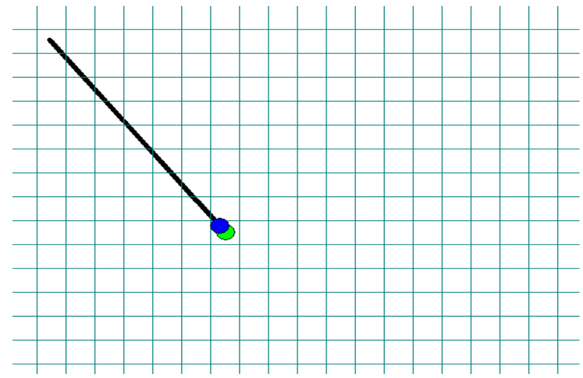

Fig. 8. The trajectory in the first case.

If the robot's environment contains one or more obstacles, the robot must be able to avoid collisions with these objects. The autonomous navigation system contains two basic behaviors: following the target behavior and another for obstacle avoidance. The robot performs the appropriate action to reach the final destination safely without risk of collision with objects by triggering one of two behaviors depending on the perceived situation.

Figures 9 and 10 show examples of navigation of the mobile robot in the presence of obstacles in the environment. Obstacle avoidance is a basic behavior present in almost all movements of mobile robots. Collisions can occur when moving the robot towards the target. The autonomous mobile robot must have an effective obstacle avoidance capability. As shown in the figures in all cases, the robot is able to navigate autonomously and can achieve its target efficiently by successfully avoiding obstacles regardless of its initial position.

The movement trajectory obtained and the actions generated show that the proposed control system gives better performance and efficiency.

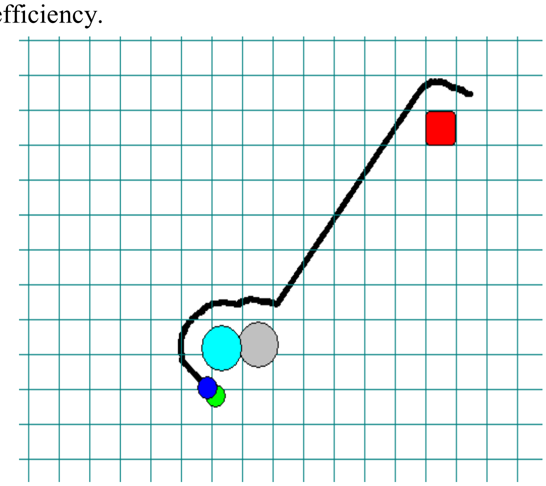

Fig. 9. The navigation of the mobile robot in the presence of obstacles.

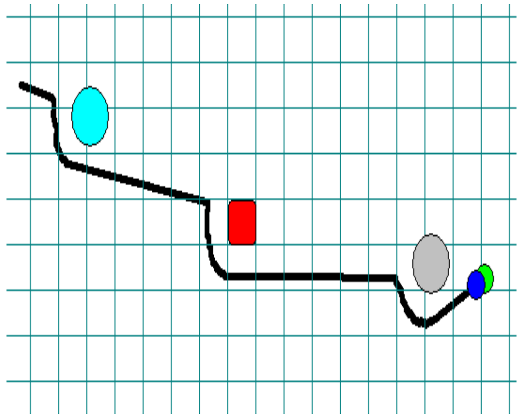

Fig. 10. The navigation of the mobile robot in the presence of obstacles.

V. Conclusion and Future Work

In this article we have proposed a new hybrid approach called NN-DEVS which combines the classical DEVS formalism with artificial neural networks to model systems with imperfect parameters.

The DEVS formalism is based on systems theory. More precisely, The notions of modularity, hierarchy, multi-modelling and the capacity of openness provided by the DEVS formalism make it possible to combine in a coherent way formalisms or paradigms based on the general theory of systems and centered on states, These characteristics makes it a formalism suitable for many areas of application.

This formalism was chosen both for its ability to integrate heterogeneous models and for its notions of coupling and hierarchical decomposition

The new NN-DEVS approach allows to represnt the stochastic (uncertain) behavior of dynamic systems.

In order to validate the NN-DEVS approach, various simulation examples are provided using the POWERDEVS tool to simulate the behavior of an autonomous mobile robot. This approach has shown its performance for the different behaviors of the mobile robot.

The proposed NN-DEVS approach proves its validation in the modelling and simulation of simple systems also in complex systems.

Several perspectives are possible following our work First, we consider the practical implementation of the methods studied on a real mobile robot,. On the other hand we can integrate other soft computing mechanisms, (fuzzy_neuro, genetic algorithms etc ...).

References

[1] P. A. Bisgambiglia, "Approximate modeling approach for discrete event systems: Application to the study of propagation of forest fires," Ph.D. dissertation, University of Corsica - Pasquale Paoli., France, 2008.

[2] B.P. Zeigler, *Theory of Modeling and Simulation*. New York: Wiley, 1976.

[3] F. Barros, "Dynamic structure discrete event system specification : a new formalism for dynamic structure modelling and simulation," in *Proceedings of Winter Simulation Conference* , 1995.

[4] A. Uhrmarcher. "Dynamic Structures in Modeling and Simulation : A Reflective Approach," *ACM Transactions on Modeling and Computer Simulation*, vol. 11, pp. 206–232, 2001.

[5] L. Ntaimo and B.P. Zeigler, "Expressing a forest cell model in parallel DEVS and timed cell-DEVS formalisms," in *Proceedings of the 2004 Summer Computer Simulation Conference*, 2002.

[6] A. Troccoli and W. Gabriel, "Implementing parallel cell-DEVS," in *IEEE, editor, Proceedings of the 36th Annual Simulation Symposium*, 2003.

[7] J. S. R. Jang, C. T. Sun, E. Mizutani, *Neuro Fuzzy and Soft Computing*. NJ: Prentice Hall, 1997.

[8] Z. Effendi, R. Ramli, J. A. Ghani, "Back Propagation Neural Networks for Grading Jatropha curcas Fruits Maturity," *American Journal of Applied Sciences*, vol. 7, issue 3, pp. 390-394, 2010.

[9] M. C. O'Neill, "Training back-propagation neural networks to define and detect DNA-binding sites," *Nucleic Acids Research*, vol. 19, issue 2, pp. 313-318, 1991.

[10] P. J. Drew, J. R. Monson, "Artificial neural networks," *Surgery*, vol. 127, issue 1, pp. 3-11, 2000.

[11] Y. Huang, "Advances in artificial neural networks-methodological development and application," *Algorithms*, vol. 2, issue 3, pp. 973–1007, 2009.

[12] PowerDEVS, http://sourceforge.net/projects/powerdevs/

[13] R. Kenaya, K. C. Cheok, "Back Propagation Neural Controller for a Two-Drive Robot Vehicle", in Proceedings of the World Congress on Engineering and Computer Science , San Francisco, USA, 2010.

[14] F. Bergero, E. Kofman, "PowerDEVS: A Tool for Hybrid System Modeling and Real Time Simulation," *Simulation*, vol. 87, issue1-2, pp. 113-132, 2011.

[15] B. R.Valluru and V. R. Hayagriva, *C++ Neural Networks and Fuzzy Logic*. USA: M & T Books, 1995.

[16] S. Haykin, *Neural Networks*. UK: Macmillan College Publishing, 1994.

[17] Y. Dahmani, A Benyettou, "Fuzzy Reinforcement Rectilinear Trajectory Learning," *Journal of Applied Sciences*, vol. 4, issue 3, pp. 388-392, 2004.

[18] A. Carling, *Introducing Neural Networks*. USA: John Wiley & Sons, 1992.

[19] M. B. Christopher, *Neural Networks for pattern recognition*. New York: Oxford University press, 1995.

[20] N. Seddari, S. Boukelkoul, A. Bouras, M. Belaoued, M. Redjimi, "A new transformation approach for complex systems modelling and simulation: Application to industrial control system," *International Journal of Simulation and Process Modelling*, vol. 16, issue 1, pp. 34-48, 2021.

[21] Y. Dahmani, H. N . B. Ali, A. Boubekeur, "XML-based devs modelling and simulation tracking," *International Journal of Simulation and Process Modelling*, vol. 15, issue 1–2, pp. 155–169, 2020.

[22] K. Mostefaoui, Y. Dahmani, "Modeling of the Reactive Navigation of Autonomous Robot using the Discrete Event System Specification DEVS," *International Journal of Computer Applications*, vol. 45, issue 9, pp. 19-24, 2012.