ORIGINAL PAPER

MODELLING AND SIMULATION OF DISTRIBUTED SYSTEMS USING INTELLIGENT MULTI-AGENTS

OTILIA ELENA DRAGOMIR[1,2]

Manuscript received: 19.03.2022; Accepted paper: 22.06.2022;

Abstract *Among the challenges in the field of the paper we have identified the necessity of control methods and tools of mobile robots (MRS) equipped with robot manipulator (RM) serving assembly/ disassembly mechatronic lines (A/DML).The framework of the work reported here is represented by SMART&ASTI A/DML served by two MRs with RM, working collaboratively and the goal is development of a multi-agent system able control interactively trajectories of MRs, working collaboratively to serve the A/DML, by avoiding the collisions between them. The advantage offered by the proposed solution consists in graphical representation of the trajectory of MRs working collaboratively, as well as the current status of them, while the users are able to interact intuitively with the MRs trough the proposed GUI The added value of the paper consists in the implementation of multi-agents systems in a complex A/DML served by two MRs equipped with RM, working collaboratively, the increased autonomy in communication between the entities of entire system, and the adaptive control of MRs trajectories.*

Keywords: *multi-agents system; mobile robots; assembly; disassembly; mechatronic line.*

1. INTRODUCTION

The concept of Industry 4.0, as well as industrial intelligence, have been used in different contexts since are relative new concepts and embraces several techniques and methods. The necessary tools to implement industrial intelligence towards modelling and simulation, predictive analysis or optimisation rely firstly on the possibility to model and simulate the process, or system and to test solutions and scenarios without stopping and interfere with their physical operation. Secondly, having data available it is possible to provide optimal solutions to support decision, such as, optimal industrial assembly/ disassembly (A/D) operations scheduling and planning. Finally, using predictive analysis, mostly based on data analytics and machine learning algorithms give the possibility to predict the behaviour patterns of an A/D process, to find anomalies, classify events, predict events.

Ones of the most important actors in the Industry 4.0 context are the robots. The robotics market is growing exponentially and will continue to do so for the next decade, by having robots capable of performing evermore difficult tasks. The widespread of robot technology is helped by artificial intelligence (AI). Nowadays the applications using robots controlled by AI knowledge to perform A/D tasks are limited, being mostly focused on increasing robots' autonomy and reasoning.

---

[1] Valahia University of Targoviste, Department of Automation, Computer Science and Electrical Engineering, 130004 Targoviste, Romania. E-mail: otilia.dragomir@valahia.ro.
[2] Academy of Romanian Scientists, 050044 Bucharest, Romania.

https://doi.org/10.46939/J.Sci.Arts-22.2-a19

Thus, in the context of the Industry 4.0 served by robots, this paper addresses modelling and simulation of mobile robots (MRs) serving an A/D mechatronic line (A/DML) using intelligent agents. These artificial intelligence tools allow us to take into consideration different working scenarios of MRs, analyses operational events and to take into consideration the specificity of each A/D workstation. The approach based on the integration of intelligent agents in the control of mobile robots allows us to reduce the down times of the A/D workstations because the trajectories of mobile robots will be optimized in relation to the existing configuration of the A/DML, the structural characteristics of the environment where they can move and the potential obstacles.

2. SMART&ASTI A/DML A/DML SERVED BY MOBILE ROBOTS FRAMEWORK

2.1. CONTEXT

The assembly/ disassembly industry is undergoing vast changes with the rapid development of production automation, process control, information technologies and networking. Adaptable assembly/ disassembly systems have been developed to achieve more flexibility to enable adding product variants and scaling productions [1]. This complex structures, assembly/ disassembly mechatronic lines (A/DML), represent the framework of our research reported here.

The architecture of SMART&ASTI A/DML available in the laboratory of the coordinator is based on ASTI technology, served by wheeled mobile robots (MRs) equipped with robotic manipulators (RM), working collaboratively. It is composed of 6 work stations with components storing areas, 1 workstation for quality check and to store the products that pass the test, and 1 workstation for the disassembly of the faulty parts (Fig. 1).

Figure 1. Assembly/Disassembly stations from SMART&ASTI A/DML served by MRs

The processing stations have warehouse parts associated, in each of them being located one of the components that are assembled to the final product. Each station is equipped with position sensors for a precise localization to the corresponding warehouse. The piece parts filling on the conveyor belt is achieved via actuators type of pneumatic piston driven by a pneumatic system. The architecture of the automation system is a distributed one and consists of SIEMENS Simatic S7-300 with a series CP 314C-2 DP processor and a communication module SIEMENS CP 343-2. The automation system is connected to the PROFIBUS DP which interfaces with MA auxiliary modules I/O type SIEMENS ET200S IM 151-1 stations distributed on each of the flexible system for assembly/disassembly. Each of the 6 SIEMENS ET200S-IM 151-1 modules has digital and analogue I/O signals taking signals from transducers and giving commands to actuators. A SIEMENS Simatic HMI TP 177 operator panel is connected on the PROFIBUS DP terminal, through which the system status can be checked and an execution process of assembly or disassembly can be implemented.

Secondly, our research is based on wheeled mobile robots (MRs) equipped with robotic manipulators (RM). In our context, SMART&ASTI A/DML is served by two MRs working collaboratively and two configurations of available MRs in the laboratory can be considered to serve the A/DML: a). one Pioneer 3-DX equipped with RM or Pioneer 6-DOF Arm Cyton 1500 used for manipulation, and the other PatrolBot used for transport b). one Pioneer 3-DX equipped with RM and the other Pioneer 6-DOF Arm or Cyton 1500, each of them used for transport/manipulation.

2.2. PROGRAMMING METHODS OF MOBILE ROBOTS

Conventional programming methods or simulation programs of MRs equipped with RM serving A/DML require specialized knowledge so fast and flexible changes are difficult to realize [2]. Intelligence and collaboration between reconfigurable, adaptable and smart MRs serving A/DML to evolve and quickly respond to the increasing product variety.

Different approaches can be found in literature which aim at abstracting the programming procedure so that the MRs program code are automatically generated [3]. Intuitive programming methods are developed especially in the field of human-robot collaboration. MRs enable humans and robots to work together within the same workspace and are already used in industry. These can be guided by hand, whereby points or trajectories can be stored (kinesthic teaching). Thus, MRs offer a suitable hardware design for flexible and simple programming. On the other hand, the software is often simplified using program blocks to abstract the textual programming [4, 5].

A further simplification of programming can be achieved by translating human actions directly into a robot program Programming-by- Demonstration (PbD) procedures are based on the imitation of human actions by the robot. By deriving motion paradigms, the demonstrated actions can also be transferred to other or similar tasks [6]. Another approach of programming MRs is the use of intelligent agents. In order to generate waypoints for determining the path of the robot, these software tools are used to recognize obstacles and interpret them, respectively [7, 8].

This paper relies on the result of our previous research papers in modelling [9, 10], simulation through informatics technologies based on: machine learning techniques, neural networks, synchronized hybrid Petri Nets [11], and real- time control of MRs serving A/DML using Lab View or C++ [12].

The added value of the paper consists in the implementation of multiagents systems in a complex A/DML served by two MRs equipped with RM, working collaboratively, the increased autonomy in communication between the entities of entire system, and the adaptive control of MRs trajectories.

In this framework and considering the challenges described above, this paper proposal consists in development of a MAS able to control the trajectory of MRs serving A/DML collaboratively, avoiding the collisions between them. Thus, in the second part of the paper the work is focused on the modelling of the proposed MAS. The MAS architecture is simulated and tested using different acting scenarios.

3. MULTI-AGENT SYSTEMS FRAMEWORK

3.1. INTELLIGENT AGENTS AND MULTI-AGENT SYSTEMS

Multiagent systems (MAS) are artificial intelligence (AI) technologies possessing significant (but limited) processing, sensing, communication as well as memory and energy storage capabilities. Theirs integration in mobile robotic systems modelling [13] help to define the flow of information: how the environment is perceived, how it is transformed and finally, how decisions are being made. In MAS different intelligent agents coexist and have the capabilities depicted in Fig. 2.

Figure 2. Characteristics of MAS

One of the most important characteristics of agents is that they are self-contained meaning that they are uniquely identifiable individuals. They are also autonomous so they can act on their own within an environment. Although agents make their own decision, they are influenced by information from interacting with other agents. This way agents show adaptive behaviour meaning they can modify their behaviour based on interaction with other agents. Most agents have a goal which allows them to compare outcomes of behaviour and to act on these outcomes. Besides interacting with each other, agents also interact with their environment. The environment in which agents act can be provided by geographic information.

Intelligent agents are hardware or software entities situated in the same environment, able to perform autonomous actions in order to meet theirs design objectives [14]. Evolving side by side, they need to share the same physical environmental information (e.g., measurements, intentions) as well as resources and services (e.g., storage space, energy supplies, processing power, Internet or routing services) in order to realize their full potential and prolong their mission [15]. However, the interactions of these agents must not compromise the objectives of each individual agent.

For instance, a robot can have a task to go from one point to another in some time, but in addition to this task, it can be asked that the robot avoid some area or obstacles. In addition to these it has own structural constraints: limited velocity or acceleration. Thus, physical constraints, it might not be enough to minimize a cost, but one might also want to minimize a cost under control or state constraints.

Considering these tools, we will advance the state-of-the-art and our previous experience in the following directions:

The MAS is an evolving technique used to control robot with no risk and cost effective. The user can observe the robot's movement and behaviour when performing a task, on the interface. To make sure the robot can perform the task using the optimum path, analysis, discussions and trial-and-error testing need to be conducted.

The development of MAS to control the trajectories of mobile robots will be optimized in relation to the existing configuration of the A/DML, the structural characteristics of the environment where they can move and the potential obstacles. These approach offers potentially great and cost-effective benefits. The time for which a robot is out of production may be reduced by as much as 85% by using offline programming because the used simulation program will offer the ability to record and analyse the manipulations during a procedure.

Using MAS for MRs path planning and control via GUI as well as for the display the virtual content will enable the users to improve their skills, as well as test them as to whether or not their manipulative skills and paths are at an acceptance level of performance for a given task.

The proposed MAS-based robot programming system consists of a series of objects, which can be viewed from any position and angle by the user through the GUI design for this purpose. Objects have certain actions assigned to them, which simulate the actual environment. Through this virtual environment, users will be able to select the appropriate sensing modality and programming that interact directly with the actual case; will change the position and orientation of the monitor window in the virtual world in real-time using input devices.

On the other hand, users will have the possibility to simulate and plan an operation through the robot simulation program. They will run trials as many times as they want with no risk and extra cost. The possibility of damage to the robot is reduced to a minimum level because the robot commands are applied after testing in a virtual environment. It also removes the user from a potential hazardous environment.

Offline programming within the virtual environment also will reduce the required skills of a programmer, programming error and programming time. In additions, experiences users can use the system to teach and demonstrate their techniques through the proposed GUI. Programs are generated and tested through virtual teaching. Therefore, A/DML is not disturbed

3.2. MAS PROPOSED ARCHITECTURE

The application of the proposed concept will be achieved following an approach based on MAS hybrid layered architecture The hybrid formalism of MAS is the dominant approach towards modelling and analyses of the robotic systems. Our approach it is based on three reactive or deliberative layer architecture (Fig. 3).

The mobile robots integrating intelligent agents (MRIA) have the ability to perceive information from the environment through a perception subsystem (layer 0 from Fig. 3). The

MRs used in our research collect data information related to the environment using ultrasound sonars, laser scanner to find objects, infrared proximity sensors to identify the walls; magnetic compass for orientation; sensors on the gripper to pick up the pieces. It computes a force based on weights proportional with the half of the distance till the station. Thus, the MRIA will have ensured the sensing, perceptive and knowledge acquisition abilities.

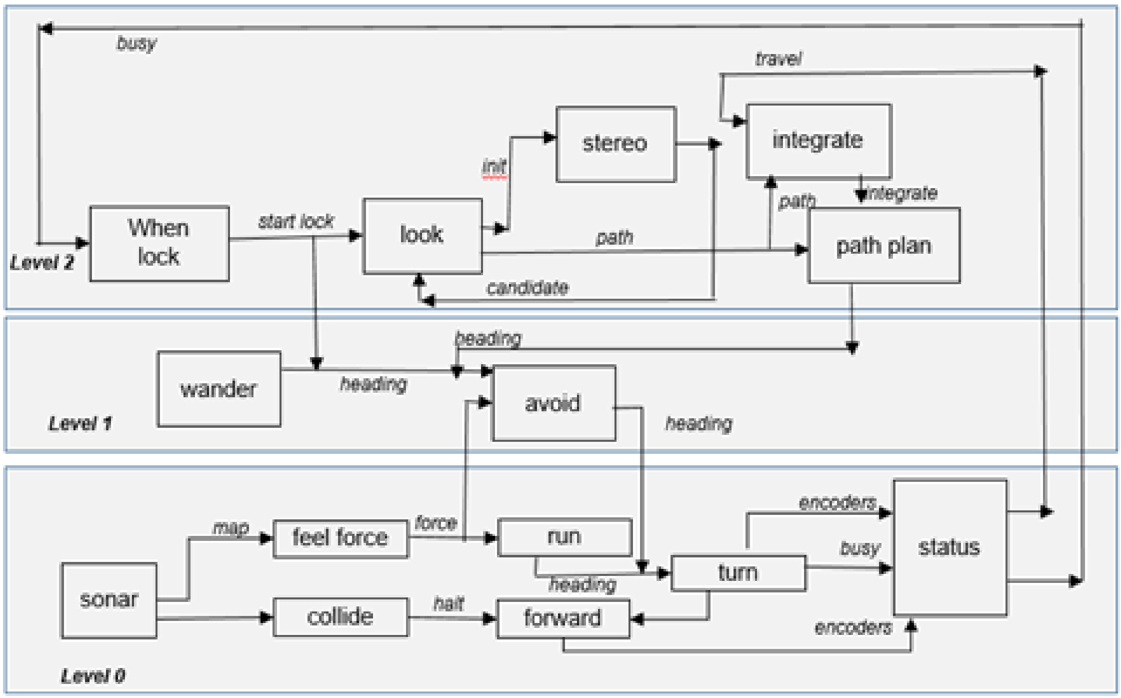


Figure 3. MAS architecture

In accordance with the perceived map of the environment and having as specific goal the request to perform a certain action, the MRIA decide which action to do: run (feel force) or stay. It is not the operating system who decides which action will be performed like in the distributed systems. Layer 0 contains reactive agents which respond to the environmental changes and never focus on a goal, meaning that possess only decision making and acting abilities. The modelling subsystem (Fig. 4) the basic characteristics of intelligent agents integrated in MR along with their attributes are defined.

Each of these abilities provide the opportunity to the agent to transform the inputs into beliefs or desires (goals) and the outputs into intentions. The structure of this subsystem a hierarchical one, composed of three layers: beliefs- desires- intentions (BDI). Because the agents generate a specific response, relative to the environment change or interact each other, this subsystem is a reactive one.

The action subsystem relies on means-ends analysis method for updating the agents' attributes in response to their interactions with other agents and the environment.

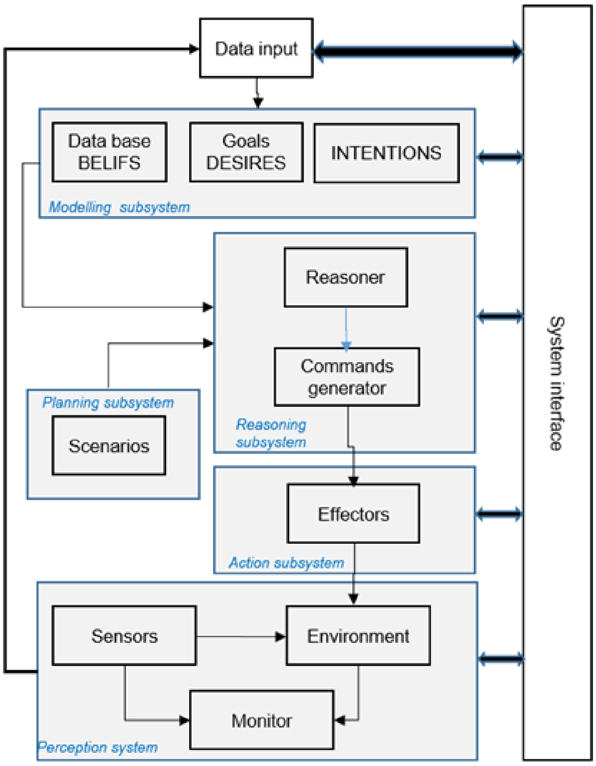


Figure 4. MAS information flow

Precisely the MRIA receive a representation of the current state of the environment, a goal to be achieved and a representation of the action to be performed. In consequence a sequence of actions (plan) is generated. The actions are integrated in a plan, have preconditions and post conditions and are defined as proactive behavioural rules. The plan is composed of deliberative functions, generated to specify which agents from MAS interact, when and how interact. They ensure the learning and inference abilities for the agents.

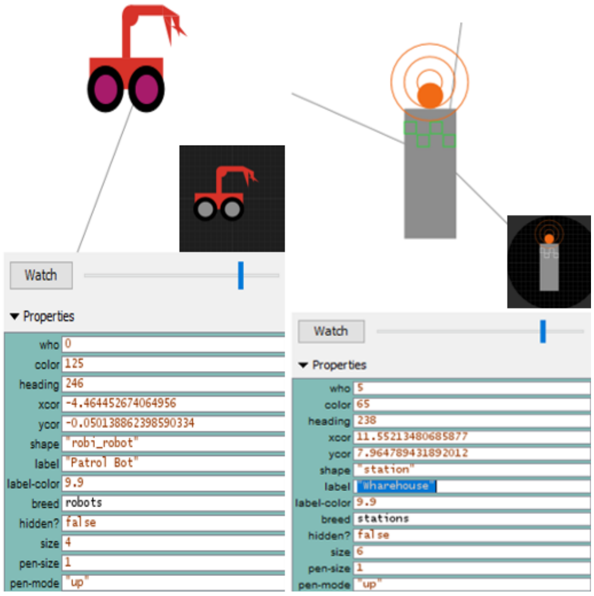
4. MULTI-AGENT SYSTEM SIMULATION AND RESULTS

4.1. MULTI-AGENT SYSTEM SIMULATION

Netlogo software [16] was used to achieve the goal of our paper, to control the trajectory of MRs serving A/DML collaboratively, avoiding the collisions between them. It is a free and open source programmable modelling environment best suited for modelling and simulating complex systems, even for prototyping complex models [17]. Netlogo uses turtles, patches, links and the observer to design the intelligent agents evolving in the environment.

The proposed MAS uses turtles for design of MRs. Each of them have own behaviour, constraints and requirements which influences the environment. These characteristics are represented as sliders in the interface and by changing theirs values: the speed, the direction or the visibility area of the MRs, for each tick, can be adjusted.

For modelling the A/DML stations are used agents of turtle’s type (Fig. 5), even in our environment have fixed place in the A/DML architecture. The MRs are the agents that move around in the environment.

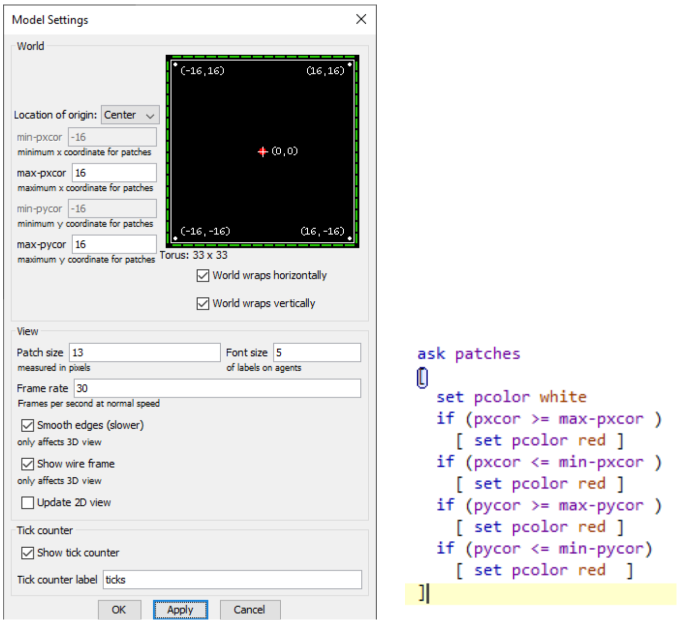


who 0
color 125
heading 246
xcor -4.464452674064956
ycor -0.050138862398590334
shape "robi_robot"
label "Patrol Bot"
label-color 9.9
breed robots
hidden? false
size 4
pen-size 1
pen-mode "up"

who 5
color 65
heading 238
xcor 11.55213480685877
ycor 7.964789431892012
shape "station"
label "Wharehouse"
label-color 9.9
breed stations
hidden? false
size 6
pen-size 1
pen-mode "up"

Figure 5. MAS – turtle agents' configuration

The environment of the model consists of a grid of cells called patches, identified with x and y coordinates. As mentioned in our previous papers, the total distance between stations is 4,980 meters. The environment used by Netlogo is rectangular and does use the exact borders of the laboratory where the A/DML is installed. This means that the length and width of the rectangular environment is determined by maximum length of 10,500 meters and 7,500 meters' width of the laboratory where the A/DML is installed. As a result, the total environment of the model has a surface of 78,75 m^2. The total number of patches is determined by the minimum and maximum values of the orthogonal axes of coordinates. NetLogo starts up with 1089 patches total, meaning that, for our model, the size of one patch represents 0, 0723 m^2 (Fig. 6).



Location of origin: Center
min-pxcor -16
minimum x coordinate for patches
max-pxcor 16
maximum x coordinate for patches
min-pycor -16
minimum y coordinate for patches
max-pycor 16
maximum y coordinate for patches
Torus: 33 x 33
☑ World wraps horizontally
☑ World wraps vertically
Patch size 13
measured in pixels
Font size 5
of labels on agents
Frame rate 30
Frames per second at normal speed
☑ Smooth edges (slower)
only affects 3D view
☑ Show wire frame
only affects 3D view
☐ Update 2D view
☑ Show tick counter
Tick counter label ticks

```
ask patches
[
  set pcolor white
  if (pxcor >= max-pxcor )
    [ set pcolor red ]
  if (pxcor <= min-pxcor )
    [ set pcolor red ]
  if (pycor >= max-pycor )
    [ set pcolor red ]
  if (pycor <= min-pycor)
    [ set pcolor red  ]
]
```

Figure 6. MAS – patches agents' configuration

Links agents do not have coordinates but they are visual lines connecting two or more turtles. These are used for visual representation of the MRs movement in the environment. (Fig. 7).

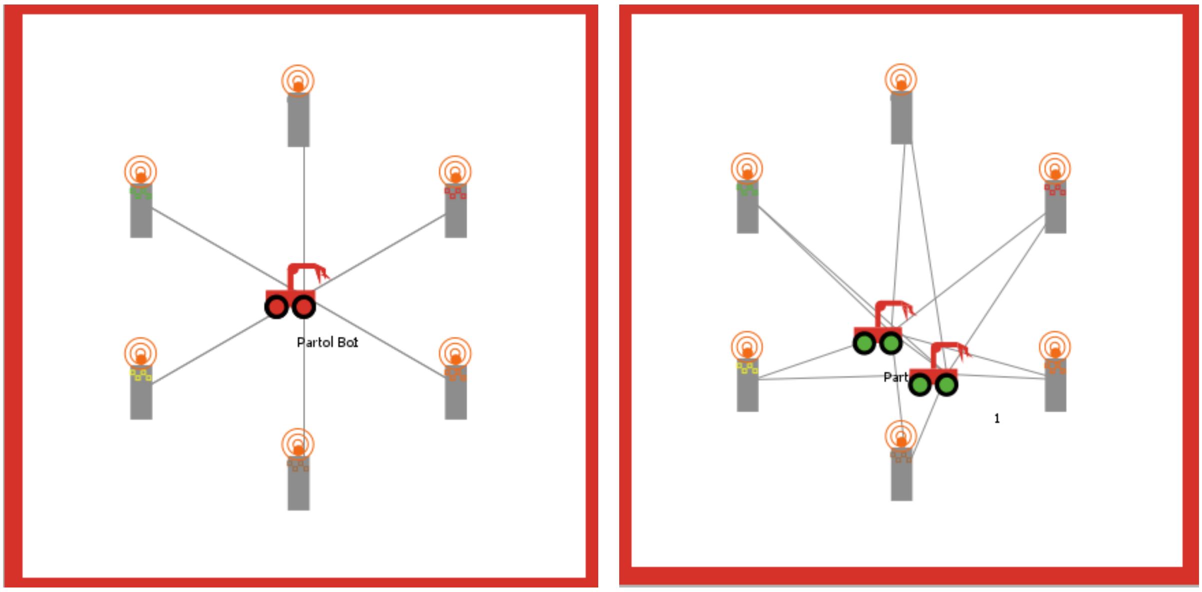


Figure 7. Links agents connecting MRs with A/DML stations.

In the simulation mode, on the GUI the user can place extra obstacles of different shapes, for the MRs, in addition with the predefined ones represented by the A/DML placement in the environment and physical limitation of the laboratory (Fig. 7). Also, it is possible the change of: the frame of reference so the visualisation of the environment to be cantered on the MRs, the speed of the robots, or the direction of their movement.

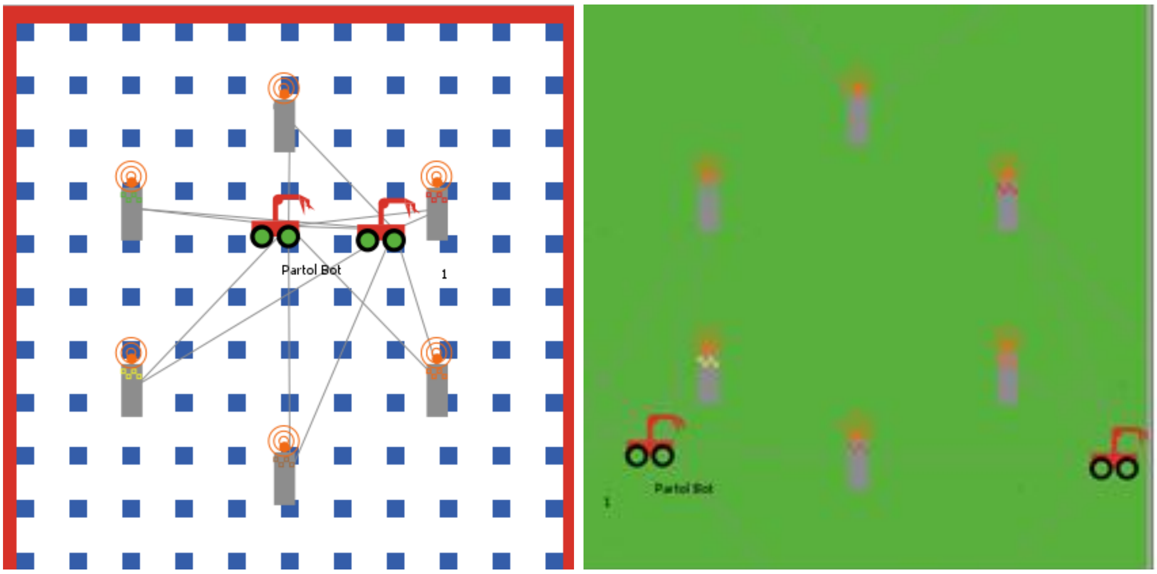
4.2. TESTS AND RESULTS

Scenario analysis is a method which maps the diversity of different scenarios. Its aim is to explore the range of possible choices, and to test how well such choices may succeed in various possible futures [18].

In the first scenario the behaviour of the robot was on “look ahead” mode, meaning that the MRs looked directly ahead to see if there is an obstacle at distance defined by radius-length slider and turn a random amount. If the behaviour of the robot is on “boid” mode, it uses a basic cone of vision sense as defined by the radius-angle and radius- length sliders to determine if there are any obstacles ahead, and turn a random number amount as defined by rate-of-random-turn slider, with tendency to turn to the right (Fig. 8).

During the simulation tests, we observed that, by setting different values, for the MRs vision cone radius-length or radius-angle, in some cases the MRs don’t see the obstacles (blue patches in Fig. 8a) and as a result will run right over the top of them (all the patches became green in Fig. 8b).

Moreover, due to the initial random placement of the robot in the environment, even it has a tendency to head in a right turning direction as the rate of random turn to the right (as specified by the slider) is twice that of the rate of random turn to the left. Sometimes it gets stuck, seemingly trapped in the same place.



a) b)

Figure 8. a) Scenario 1; b) Scenario 1 results.

In the second scenario (Fig. 9) the simulations test with the model were run to find a solution for the location of MRs and of the A/DML stations. In this respect, multiple random distributions were created and then simulations are run with the model to produce multiple solutions to the location problem. For solving the location allocation problem, two criteria are formulated: to control trajectories of MRs, working collaboratively to serve the A/DML, by avoiding collisions between them and of them with other obstacles.

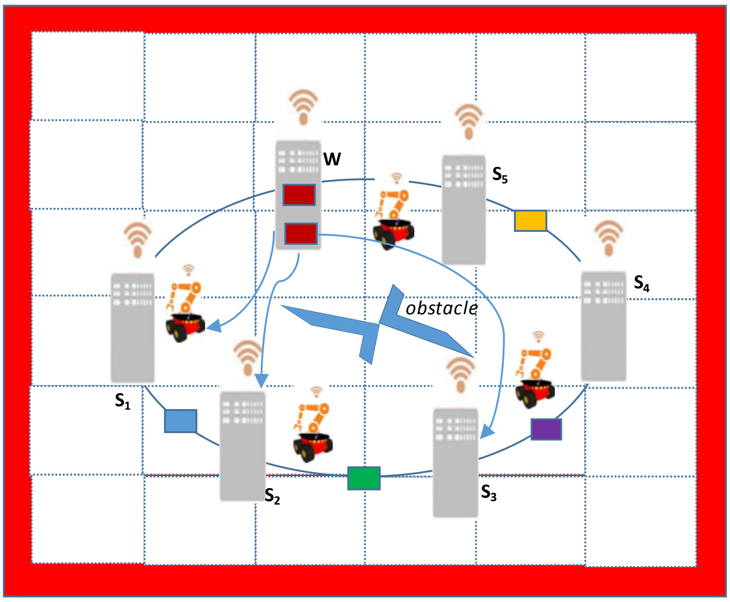


Figure 9. Scenario 2.

Thus, by increasing the radius-length (while keeping the other variables the same) discernibly changes the behaviour of the robots. Instead of serving most of the A/D stations, when the radius length is large then MRs serve collaboratively a much smaller number of A/D stations, usually the ones placed on the top of the environment.

By, setting the robots' speed to 0.1, the rate-of-random-turn to 40, radius-angle to 300, radius-length to 1, followed by moving the speed slider from "normal speed" to "faster" will result in a fast serving of the A/D stations while reliably avoiding the obstacles.

By, increasing the radius-length to 5 (while keeping the other variables the same) discernibly changes the behaviour of the robots. Instead of covering most of the environment, they cover a rectangular path of width 4 to 5 around the outside of the environment but indented by about 3 to 4 patches in from the outer boundary. It seems to refrain from going inside of the rectangular path, and sometimes gets stuck spinning around one of the obstacles.

5. CONCLUSIONS

This paper proposed a MAS able to control the trajectory of MRs serving A/DML collaboratively, by avoiding the collisions between them and with other obstacles.

Firstly, were discussed the concepts of multiagents systems and it was underlined the increasing interest for these intelligent tools. The researchers preferred these tools in the last decade, because of overall system performance, reliability, extensibility, robustness, maintainability, responsiveness, flexibility, and reuse.

In the second part of the paper the work is focused on the modelling and simulation of the proposed MAS architecture. The MAS hybrid layered architecture was tested using different scenarios.

The work is in progress in this area of research, in several directions: new scenarios have to be identified in order to improve de knowledge data base of MAS; the model could be extended to add gradual acceleration and deceleration for the robots; new scenarios should be identified in order to minimize the travel durations of MRs to the A/DML stations, according to the stages of an assembly/disassembly cycle.

REFERENCES

[1] Shen, W., Wang, L., Hao, Q., *IEEE Transaction on System, Man, and Cybernetics-Part C: Application and Reviews*, **36**(4), 563, 2006.

[2] Blesing, C., Luensch, D., Stenzel, J., Korth, B., Concept of a multi-agent based decentralized production system for the automotive industry, *Advances in Practical Applications of Cyber-Physical Multi-Agent Systems*: *The PAAMS Collection*, 19, 2017.

[3] Yuan, M.L., Ong, S.K., Nee, A.Y.C., *International Journal of Production Research*; **46**(7), 1745, 2008.

[4] Ong, S.K., Wang, Z.B., *CIRP Annals–Manufacturing Technology*, **60**(1), 1, 2011.

[5] Henderson, S., Feiner, S., Evaluating the benefits of augmented reality for task localization in maintenance of an armored personnel carrier turret, *Proceedings of the IEEE and ACM International Symposium on Mixed and Augmented Reality*, 135, 2009.

[6] Zhu, J., Ong, S.K., Nee, A.Y.C., Online authoring for augmented reality remote maintenance, *Proceedings of the International Conference on Computer Graphics and Imaging*, 87, 2011.

[7] Zhang, J., Ong, S.K., Nee, A.Y.C., *Computer-Aided Design*, **42**(12), 1167, 2010.

[8] Shen, Y., Ong, S.K., Nee, A.Y.C., *Design Studies*; **31**(2), 118, 2010.

[9] Dragomir, F., Minca, E., Dragomir, O.E., Filipescu, A., *Sensors Journal*, **19**(15), 3266, 2019.

[10] Duca, O.G., Gurgu, I.V., Minca, E., Filipescu, A., Dragomir, F., Dragomir, O.E., *Proceedings of 23rd International Conference on System Theory, Control and Computing (ICSTCC)*, 620, 2019.

[11] Dragomir, F., Caramida, M.S., Dragomir, O.E., Minca, E., *Journal of Science and Arts*, **19**(2), 529, 2019.

[12] Minca, E., Filipescu, A., Coanda, H.G., Dragomir, F., Dragomir, O.E., Filipescu, A., *Proceedings of the International Conference on System Theory, Control and Computing (ICSTCC)*, 335, 2018.

[13] Shaaban, H.M., Ali, M., *Global Journal of Flexible Systems Management*, **20**, 267, 2019.

[14] Shoham, Y., Leyton-Brown, K., *Multi-agent systems: algorithmic, game-theoretic, and logical foundations*, England, 2008.

[15] Keogh, K., Sonenberg, L., *Applied Science*, **10**(15), 5335, 2020

[16] NetLogo 2022, http://ccl.northwestern.edu/netlogo.html. Last accessed: March 10th, 2022.

[17] Page, J., Keane, T.M., Naughton, T.J., *Journal of Parallel and Distributed Computing*, **70**(7), 758, 2010.

[18] Benth, F.E., Ibrahim, N.A., *Journal of Energy Markets*, **10**(3), 1, 2017.