PROCEEDINGS OF
INTERNATIONAL CONFERENCE ON ADVANCED TECHNOLOGIES
https://proceedings.icatsconf.org/
11th International Conference on Advanced Technologies (ICAT'23), Istanbul-Turkiye, August 17-19, 2023.

Using a Soccer team as a test bed for multi-agent systems simulation

Areen Naji [1], Rashid Jayousi [2], Amjad Rattrout [3]

[1] *Information Technology Engineering, Arab American University, Palestine*
a.naji1@student.aaup.edu, ORCID: 0009-0008-5189-977X

[2] *Computer science, Al Quds University, Palestine*
rjayousi@staff.alquds.edu, ORCID: 0000-0002-3685-5516

[3] *Computer science, Arab American University, Palestine*
amjad.rattrout@aaup.edu, ORCID: 0000-0003-1842-4613

Abstract

The STMAS system is intended to imitate a soccer team and its behavior; we believe it can be used effectively as a test bed for multi-agent systems. It is constructed utilizing distributed agents that interact, communicate, and negotiate with each other to achieve the team objectives. It is based on the Jade simulation platform. The system is tested and compared to a pure soccer team using multiple MAS techniques. The results demonstrated that applying MAS techniques of negotiation and task distribution improves team performance, and STMAS is offered as an efficient test bed for new and distinct MAS techniques with varied scenario experiments. In addition, a mathematical model is created to compare the simulation results. Overall, STMAS provides a versatile and efficient MAS simulation and evaluation test bed. It is an excellent platform for comparing and evaluating various MAS approaches. **Keywords** ***MAS, STMAS, TPA, OTPA, CAS, MARL.***

I. INTRODUCTION

Overall, this paper describes a multi-agent system used to simulate the dynamic environment of a football match. The system is designed to be adaptive to different scenarios and is tested using the Jade Simulation Platform. In the literature review, various existing works on multi-agent systems are discussed. The system architecture is illustrated and the system design is explained in detail. Different scenarios are simulated and the results are discussed. Finally, the effectiveness of STMAS is investigated in the conclusion.

A. *Related works*

This section will attempt to cover some related works that have built test beds for multi-agent systems in various ways, as well as works related to using the soccer game in multi-agent systems.

It has recently undergone an evolutionary development in the use of MAS to find solutions for complex situations. For instance, Corchado and his partners in [18] developed a MAS university practical application that uses a project monitoring intelligent agent system for student supervisor positioning and student-teacher meeting scheduling. They proposed to use their system to evaluate the autonomous agents and test their integration and their learning methods. They evaluated their system as easier in comparison with other works; however, this work does not deal with complex or dynamic situations.

On the other hand, there is a big challenge in designing MAS, which is the organization that controls the agents' interactions. In this context, [1] sheds light on the multi-agent systems organization and the constraints that autonomous and heterogeneous agents have to follow when dealing with open systems. They used the soccer team as an illustration tool for the model because soccer games have protocols that need to be respected at each stage. They described the MOIS+ tool for developing organized multi-agent systems, in which there is a middleware for monitoring the agents' commitments with the organizational constraints at the system level, and the agents use Jason features in their interactions; this is the agent-level organization. Although they described the organization concept in detail, the soccer team example was presented as a case study without digging deeply into simulation and experiential design.

Clemente and his colleagues [2] analyzed the team's collective behavior corresponding to their ball possession status; they studied two teams' behaviors and positions. They collected data with high-resolution cameras from many corners of the soccer field, which was divided into nine sectors, and studied the frequency of the players' occupation during the entire game at each sector to understand some collective tendencies. They created histograms for the most field areas occupied, taking into account the moments with and without ball possession. This article taught us that teammate distribution can sometimes be imbalanced, and this has an impact on team efficiency.

The authors in [3] described their implementation of a soccer team of micro-robots and described a multi-agent systems simulation developing with MATLAB/Simulink; they also discussed how the interactions between multiple agents change their mental states. Their practical implementations were conceived using object-oriented paradigms. Each team consists of three microrobots. Their project was a first step toward a general application of agents in the world of automatic control and robotics using common tools. But they found that object-oriented paradigms are not extended enough and agents have problems being applied in practical implementations.

[4] described Soccer Server could be used to compare the performance of a neural network architecture versus a decision tree algorithm in learning soccer play-plan selection. The authors pointed out the main concepts in multi-agent systems as cooperation protocols, distributed control, and effective communication. They chose soccer as an example domain because it provides a dynamic, real-time environment in which it is still relatively easy for such tasks to be classified, monitored, and assessed. The Soccer Server is a simulator of the game of soccer in which players are controlled by individual client programs. Communication between individual clients is possible only by sending commands to the Soccer Server.

In the works [5, 6], the authors used an earlier version of the Soccer Server to investigate learning in multi-agent environments. And in [7, 8, 9, 10], they used reinforcement learning to develop the skills of a soccer-playing robot. Their robot learns how to shoot a ball effectively by constructing internal state spaces that represent the environment.

Many projects have worked with multi-agent systems test beds, but in this work, we propose a more comprehensive test bed using the soccer game case because it is a complex dynamic system that has cooperative and competitive situations. In contrast to related works in soccer game simulators, our work analyzes team members' behavior as it relates to multi-agent systems organizations, and their decision-making is guided by a mathematical model.

II. STMAS ARCHITECTURE

The main components of the STMAS system, as well as their relationships, are depicted in Fig. 1.

The team players will be the agents with the name TPA, and the opponent team players will also be agents with the name OTP. Each TPA should keep track of all other TPAs while also keeping track of OTPAs and determining who is controlling the ball. He acts depending on his role; at the same time, he interacts with other TPAs to adaptively decide his next action to achieve the team's goals.

A. STMAS components

Every multi-agent system has to contain four kinds of components to be officially a multi-agent system. These components are environment, agents, interaction, and organization. So the first important step before designing a multi-agent system is to determine these four components.

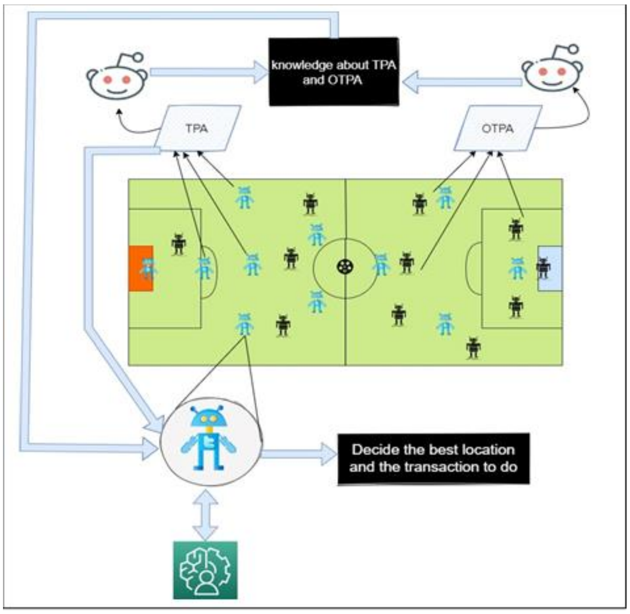

Fig 1: STMAS architecture

So, as it is illustrated by STMAS's structure main components:

- Environment: The football stadium and arena are the environments in which the STMAS is stationed.
- Agents: the players in the team (TPA) and the players of the opponent team (OPA) and the advisors (AA). All of them are the agents of the STMAS.
- Interaction: Every APA should see and recognize other APAs and TPA and AA, so all agents recognize others, AA should use communication tools such as blackboard to write their acknowledgment about other agents to be considered in TPA decision-making.
- Organization: the football arena is an organization with a huge number of rules and protocols that will be discussed later.

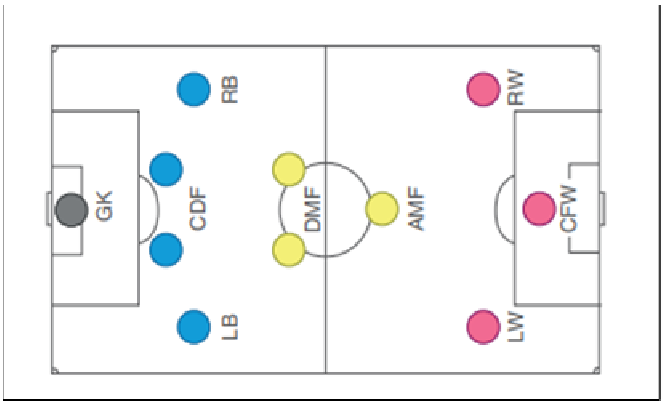

Fig 2: TPA's positions

B. STMAS agent's goals

The second important step before designing a multi-agent system is to determine the objective that the system is trying to achieve.

The system as a whole has two main goals: shooting a goal or, at the very least, protecting the team goal from opponent shooting. This is about the team level.

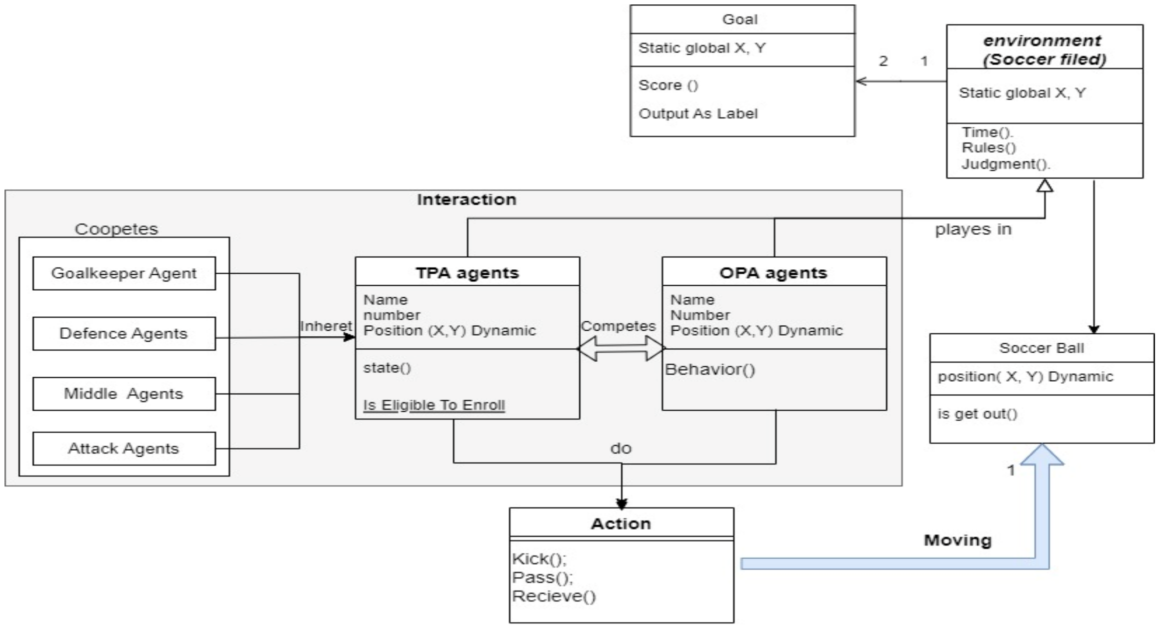

Fig 3: STMAS design

On the other hand, at the member level, the agents are organized into three main groups, and each group has its role and subgoals.

- The defense agents (TPAs) specialize in defending the team goal from the opposing team's shots while also attempting to steal the ball from the opposing team and pass it to the attack agents. As illustrated in Fig. 2, the defensible group occupies various positions [12], which are as follows:
  - I. GK: the goalkeeper, the most defensive position in soccer, his main job is to stop opposition goals; he also organizes the defense and builds play from the back; is the only player who can use their hands in their 18-yard box (except for throw-ins!).
  - II. CDF: Central defender: shuts down opposition attackers, may employ zonal or man-marking strategies, brings the ball out from the back.
  - III. RB: Right full-back: lines up on either side of the defense, marks opposing wingers, assists the wide midfielder ahead of them, may overlap and send crosses into the opposing box, and frequently takes throw-ins.
  - IV. LB: Left full-back: Overlaps and crosses into the opposing box; still mark opposing wingers when necessary.
- The attack agents aim to shoot the ball into the opponent's goal, or at least stay close to the enemy's goal area so that they can take advantage of any opportunity to achieve the goal. and they are positioned as follows:
  - I. LW: Left winger: The widest attacking player, who takes on opposition defenders, provide crosses into the box, and meets crosses from the opposite wing.
  - II. RW: right winger; similar to the left winger, but stands on the right side of the attacking zone.
  - III. CFW: Center forward: closest player to the opponent's goal; responsible for scoring goals; holds the ball up until teammates can join the attack; harasses opposition defenders.
- The midfielder, represented by the yellow positions in Fig. 2, is divided into two main roles:
  - I. DMF: Defensive midfielder: Sits in front of the defense, wins the ball back with tackles and interceptions, covers teammates when they go forward, and harasses opposition attackers.
  - II. AMF: Attacking midfielder: dictates play from behind the strikers and creates goalscoring chances for the attackers, Technique, and creativity are essential, and being able to shoot from a distance is advantageous.

We can consider the overall picture of the system's architecture complete once the STMAS main components are identified, the system goals are determined, and the agents are classified based on their objectives and roles. In the next section, the system design will be illustrated.

III. STMAS DESIGN

In this section, STMAS components will be designed as object-oriented objects having special attributes, communication protocols, and organization controlling their properties, and the system state design will be explained on two levels: team states and agent states. The design of artificial intelligence here is inspired by the book [13], and the multi-agent reinforcement learning (MARL) application in agent decision-making will also be illustrated.

As illustrated in Fig. 3, the STMAS environment is the soccer field; it is a rectangular space with the two static dimensions "width" (presented by the variable x) and "height" (presented by the variable y). The soccer ball must be inside the boundaries of the soccer field during the match, and when the ball is kicked out of the field, the match will be stopped and then started again, but the ball will be given to the opposing team of the exiting player's team (the team of the player who kicked the ball out of the field).

The soccer field contains two goals, one for the system team (team goal, TG), and the other for the opponent team (opponent team, OG). When the ball gets inside the team goal, the opponent's team points will be increased, and vice versa. These goals are presented by cuboids, which have 3 dimensions: (x: width, y: length, z: height). As a result, there are two instances of the object goal TG (Xtg, Ytg, and Ztg) and OG (Xog, Yog, and Zog).

So, the STMAS components are the soccer field containing two goals (TG, OG), one soccer ball, and two teams of players (TPA, OPA). TPA are the team players' agents, which are interacting cooperatively and are classified in the previous section into four groups depending on their rules and constraints; each agent has a special position depending on his characteristics. OPA is the opponent team players; they are also considered agents because they have competitive interactions with TPA agents and have to be monitored and their behavior has to be recognized and analyzed to be considered in the decision-making of the TPA, which are adaptive agents because state changes happen continuously in the match. TPA must be autonomous; they make decisions to help achieve the main goals, and each agent follows his classification rules based on all dynamic state changes.

Now we are going to illustrate the system state design according to team state and member state levels.

A. STMAS states design

1) The initial state

When the match began, each player in TPA should have been in his initial position (X0, Y0), as shown in the TPA Positions previously. When the ball is gotten out of the field, the match will be started again, and so on. The TPA is returned to its initial state.

2) Attacking state

When one of the TPAs takes the soccer ball, he will be the controller of the ball; the TPA in the controlling state is called the CTPA, and the TPAs' attacking state is directly activated. The CTPA will send a message to other TPAs to support him. The best distribution for supporting the CTPA will be done to move the ball effectively toward the OG. Every TPA from the central TPA agents and the attackers will calculate his score to estimate his qualifications to support the CTPA. The TPA with the highest score will be the support TPA.

Every TPA scores himself to decide if he qualifies to be the STPA, which is suggesting that he take the ball and be the CTPA in the next step. This score will depend on many factors which are illustrated in fig 4:

- The distance between the player and the CTPA means that the player has to be near enough to the CTPA as soon as possible to be able to receive the ball, so the player with the smallest distance from the CTPA will take the ball for a shorter period. As a result, the player attempts to minimize the distance between the TPA (x_P, y_P) and CTPA (x_c, y_c) which is calculated with equation (1):

$$dc = \sqrt{(x_P - x_c)^2 + (y_P - y_c)^2} \qquad (1)$$

- Simultaneously, the distance between the player TPA (x_P, y_P) and the OG (x_g, y_g) must be kept to a minimum; this is why the attacking TPA runs directly toward the OG to be close to it.

$$\mathrm{d}g = \sqrt{\left(x_p - x_g\right)^2 + \left(y_p - y_g\right)^2} \qquad (2)$$

- The obstacles are OPAs between the player and the CTPA; to be eligible to receive the ball from the CTPA, there should be no OPAs between them. That is why the attacking TPA runs away from those obstacles. And the obstacles OPA between the player and the OG, to move the ball toward the OG, should be as few as possible.

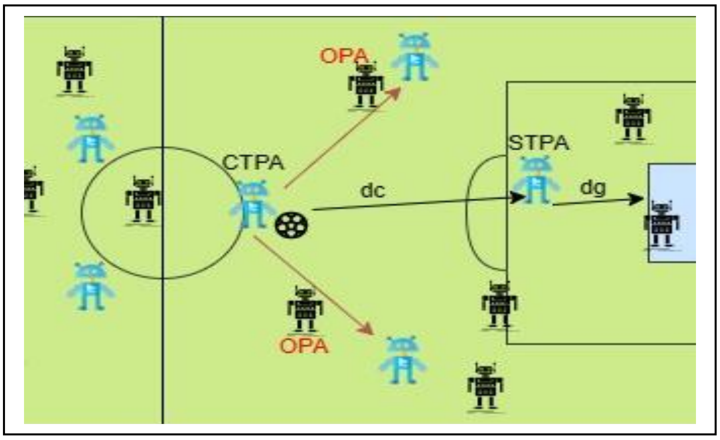

Fig 4: STPA decision making

So, the TPA that has the minimum value of the factors illustrated above will be the optimal player for the supporting task, so he must have no obstacles between him and the CTPA, and at the same time he has to match the equation: such that dc is the distance between the TPA and the CTPA, dg is the distance between the TPA and the OG, and N.O is the number of obstacles between the TPA and OG.

The STPA determination process is a multi-agent negotiation process aimed at finding the best player agent in the best location to take control of the ball in the next step. Each TPA proposes to be the STPA, and the TPA with the highest score will be the STPA, which will send a message to the CTPA to kick the ball to him. If the CTPA has a lower score than the STPA, he will send an OK message to the STPA, and the STPA will be ready to accept it.

If the opposing team gains possession of the ball at any time, the team will immediately enter the defense state, as shown below.

Finally, when the CTPA reaches near the OG, he has to shoot the goal with his maximum force to get inside the goal, avoiding the opponent's goalkeeper and other OPAs. When it is done, the TPA points will be incremented and the team will return to their initial state while the match time is not over.

3) Defending state

When one of the opponent players (OTP) takes the ball (he will be the controller player), the STP team will enter the defensive state.

They have two main goals: one is to take the ball back under team control again, and the other important goal is to protect the TG from the OTP shoots.

At this stage, the defense agents' group and the middle agent's group number are both seven, and all of them will be DAP such that they will cooperate to defend their team such that they negotiate iteratively to find the best distribution of the tasks and location as follows:

- The nearest DAP to the ball will be in the DAP0 state, which will chase the ball and try to take it. So, the distance between the player and the ball could be computed with equation (1), it must be the minimum to take the bid in this case.
- The most dangerous OTPs are those closest to the TG; they are labeled OTP1, OTP2, and OTP3, and are arranged in order of proximity to the TG. Because of this, three DPAs have to cover them. such that the nearest DAP to OTP1 will have the state DAP1, which chases OTP1. So, the distance between DAP and OTP1 has to be the minimum to accept this bid. And the same procedure will be followed to define which DAP will be DAP2 and DAP3.
- The rest of the DAPs will take positions near TG and not be covered by other DAPs.

These defense states of all DAPs will update continuously; this is the practical meaning of the concept of "real-time negotiation." Because the agents define the tasks and negotiate to find the optimal distributions of these tasks cooperatively, the agent who has the agreed-upon qualifications for the task will take the bid and execute the task. At the same time, other agents on his team compete with him. So, if there is another agent who is more appropriate to take the task, it will be taken. I mean while the agents cooperate to do the tasks, at the same time they compete to help the team achieve its goals most rapidly.

B. The effect of MARL in the STMAS mathematical model:

After studying the overview [19] and analyzing STMAS agents' behaviors, it is necessary to apply MARL science to this system, as the following scenarios illustrate.

STMAS is a cooperative tasks system at the level of the TPA's because their MARL goal is to learn how to increase the global reward function of their team; at the same time, it is a competitive tasks system at the level of the team (TPA) and the opposite team (OTPA), because the TPA agent's MARL goal is to learn how to minimize the OTPA score while increasing the score of their team; thus, STMAS could be considered a mixed system.

At the TPA level, because their relationship is cooperative and In cooperative tasks, there are many methods for training multiple agents to work together, like coordination-free methods, coordination-based methods, and indirect coordination methods. TPAs cannot coordinate their actions without communication between them or at least tracking their environment to manipulate it, so the second and third approaches will be better than the first one to use in this case.

Every agent of TPA has to keep tracking the environment, which is affected by the team state, and take the attacking state actions that increase the team's chances of scoring a new goal.

So, according to the Markov decision process, the TPA agent's actions will affect the environment state because the sum of all TPA actions will transfer the team from state x to state x+1, and each action must maximize the team reward function.

STMAS is a tuple (X, U1,...,Un, f, r1,...,rn), where n is the number of agents, which is 11 because the team number of players is 11, X is the finite set of environment states, Ui, i = 1,...,n are the finite sets of actions available to the agents, yielding the joint action set U = U1 Un, the state transition probability function, and ri: X U X R, i = 1,...,n.

The state transitions are the result of all TPA agents working together, uk = [uT1, k,..., uT n, k] T, uk U, ui, k Ui (where T represents vector transpose). The policies hi and XUi[0, 1] combine to form policy h. Because the agents' rewards ri, k+1 are dependent on the joint action, their returns are dependent on the joint policy:

$$R_i^{\boldsymbol{h}}(x) = \mathrm{E}\left\{ \sum_{k=0}^{\infty} \gamma^k r_{i,k+1} \,\middle|\, x_0 = x, \boldsymbol{h} \right\}$$

In STMAS, agents' decision-making process is adaptable to the state of the system at three levels, which are the environment state, the TPA (team) state, and the player state.

For instance, if the environment state explains that TPA is attacking and OTPA is defending, the TPA state, as is obvious, is an attacking state, the player state is CTPA for the controller player, and other TPAs try to take the action having the best result as the Markovian process dictates. This action is to target the position having the best score, which consists of three factors: distance between TPA and the CTPA, the distance between TPA and the opponent goal (OG), and the number of obstacles between TPA and So, the rewarding function will be scored by the following equation:

$$\gamma_{score}(x_P, y_P) = \sqrt{(x_P - x_c)^2 + (y_P - y_c)^2} + \sqrt{\left(x_P - x_g\right)^2 + \left(y_P - y_g\right)^2} + \text{no.ops}(x_P, y_P) \qquad (3)$$

The optimal position for STPA is the position (x_P, y_P) that has the minimum score. In this case, the maximum reward is the minimum score value. Because of this, the following reward function will affect the reward score:

$$\gamma(x_P, y_p) = 1/(\gamma_{score}(x_P, y_p)) \qquad (4)$$

IV. SIMULATION

To comprehend the behavior of the proposed system, validate its functionality, and provide evidence for correct decision-making, it should be simulated and tested using highly efficient software able to test different scenarios or process changes.

In this field, several platforms have been developed specifically for multi-agent system simulation. we used the Jade platform, which is a software framework that makes agent application development easy, in compliance with the FIPA specifications for interoperable intelligent multi-agent systems. JADE is an open-source platform, and the complete system can be downloaded from the JADE home page. We used it because it has many useful features, is based on object-oriented programming, and includes ready-to-use libraries for agent interaction and communication.

In the STMAS Java Project Main Class, an instance of the soccer field class is created. As it is shown in fig. 5 it is a two-dimensional graphical interface, and its constructor creates the soccer ball instance, goals, TPA agents, and OTPA agents; all of these components are described in the design section with their attributes.

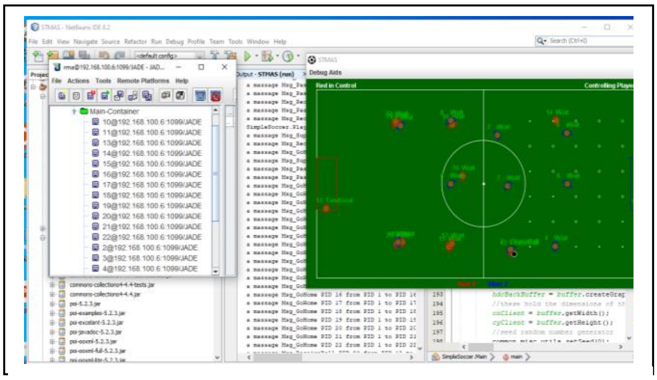

Fig 5: STMAS simulation screen

Each TPA has a moving entity class instance and could do several kinds of actions, like kick, chase, or receive the soccer ball. TPA also interacts with other TPAs by tracking the team's and soccer ball's states, communicating with other TPAs as shown above, and making decisions based on equation 4.

Many simulation scenarios are conducted to test the system's performance and the agents' behaviors, but before all, we conduct an experiment to prove that the agents' decision-making is grounded in the mathematical model.

We simulated the soccer match several times and focused on the attacking state as a test case—the point when the CTPA decides which is the best TPA to be the STPA. This situation is illustrated in Fig. 4.

When the CTPA chose the TPA for the STPA state, the positions of every TPA and OTPA member were printed in an excel file. Therefore, by applying equations 3 and 4 to all agents, the best TPA qualified to be an STPA according to the mathematical model was identified. The mathematical model's chosen STPA IDs and the simulation's chosen IDs were identical, as shown in Fig. 6.

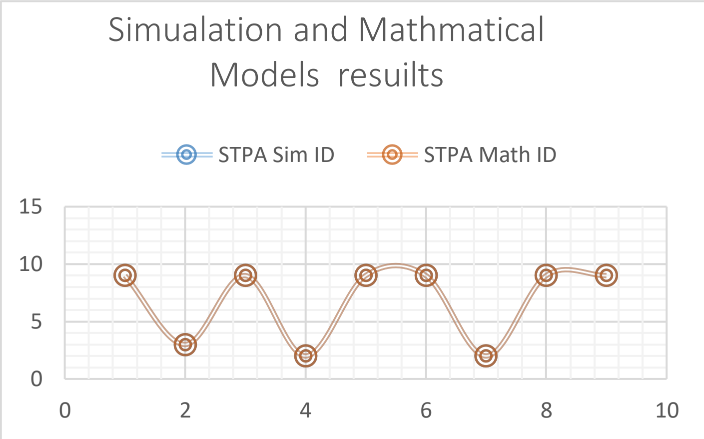

Fig 6: Simulation and mathematical model outputs

The STMAS's usefulness as a multi-agent system was demonstrated in the second experiment. In this experiment, we simulated a match between the STMAS team (TPA) and an opponent team of simple soccer team (OTPA), in which the attributes of MAS were deactivated. As a result, we had to do a simulation comparison between STMAS and pure soccer system.

The match was simulated ten times, with each match lasting five minutes, and the scores for the STMAS and Simple soccer teams were recorded as a test bench for system performance, as shown in fig. 7.

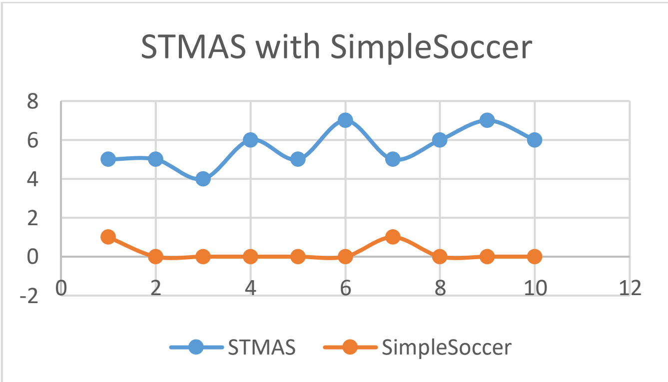

Fig 7: Comparision between STMAS and simple soccer

By changing the roles of the team members and simulating the match again with the new parameters, many scenarios could be simulated to understand the effect of the agent's behaviors, roles, and communication on team performance.

In this experiment, we simulated the match ten times between the TPA who present a fully dynamic STMAS team initiated with 5 TPA who are given the role defender and 5 with the role attacker, the goalkeeper, and the opponent team, which was also STMAS but restricted the dynamic property by giving 7 players the role defender with obligate them from attacking area, and the rest 3 members are given the attack role. The output scores for the teams are plotted in fig. 8

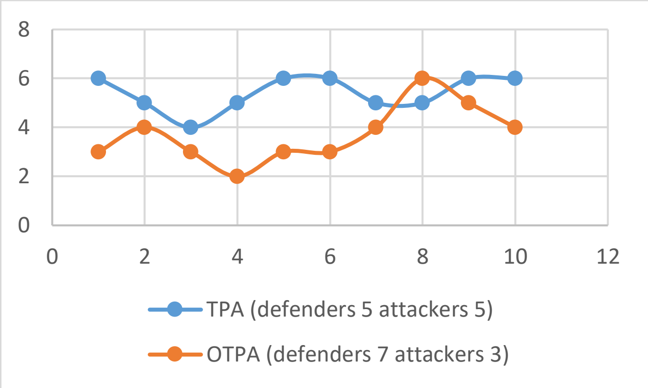

Fig 8: TPA and OTPA with different roles

V. DISCUSSION

The first experiment results shown in fig. 6 demonstrated that the simulation model is a reflection of the designed mathematical model, and the decision-making for agents is based on the well-known Markov decision process model; all of this demonstrates the system's reliability.

Looking at the second experiment results illustrated in fig. 7, it is clear that the STMAS scores are generally higher than the simple soccer scores. We can confirm this by comparing the means of the scores for all matches in the experiment; the mean of the STMAS scores was 5.6, while the simple soccer scores mean was 0.2, demonstrating the significant difference between them. and it provides an answer to the question of what the advantage of a multi-agent system solution is. It absolutely makes a difference in the soccer team case because no one can imagine a team playing football without interacting with its team members and cooperating with them to achieve the team goals, which explains why the team members must be fully cooperative. At the same time fully competitive with the opponent team.

Figure 8 depicts the final experiment results, which show that the TPA jailed score had a mean of 5.4, and the OTPA score had a mean of 3.7, indicating that the effectiveness of the STMAS increased when the attacking and defending roles were distributed up on the TPAs, and the agents dynamically changed their position according to the design mathematical model.

VI. CONCLUSION

This work illustrates the design of a soccer team multi-agent system STMAS in which the agents are the team members' TPAs and the opponent's OTPA, the agents continue to attack all other agents in the system and interact with the TPAs in a fully cooperative approach, while treating with the OTPAs in a fully competitive approach, to help the TPA team achieve the two main goals of shooting more goals in the opponent goal and saving their own goal from the opposite goals.

The system simulation of various scenarios demonstrated the STMAS's dependability and effectiveness as a test bed for multi-agent systems.

REFERENCES

[1] Hübner, Jomi & Sichman, Jaime. (2007). Developing organized multi-agent systems using the MOISE+ model: Programming issues at the system and agent levels. Int. J. Accounting, Auditing, and Performance Evaluation. 1-1. 10.1504/IJAOSE.2007.016266.

[2] Clemente, Filipe & Couceiro, Micael & Martins, Fernando. (2013). Soccer Teams Behaviors: Analysis of the Team's Distribution in Function to Ball Possession. Research Journal of Applied Sciences, Engineering, and Technology. 6. 130-136. 10.19026/rjaset.6.4047.

[3] J.Ll. de la Rosa, A. Oller, J. Vehí, J. Puyol, Soccer team based on agent-oriented programming, Robotics, and Autonomous Systems,1997.

[4] NODA Itsuki, MATSUBARA Hitoshi, HIRAKI Kazuo, and Ian FRANK, Soccer Server: a tool for research on multi-agent systems, Electrotechnical Laboratory, Tsukuba, Ibaraki 305, Japan, April 4, 1997.

[5] P. Stone and M. Veloso. Learning to pass or shoot: collaborate or do it yourself. unpublished manuscript,1995.

[6] Peter Stone and Manuela Veloso. Towards collaborative and adversarial learning: A case study in robotic soccer. Proceedings SS-96-01, pp. 88{92, 1996 AAAI Symposium, March 1996.

[7] Eiji Uchibe, Minoru Asada, Shoichi Noda, and Koh Hosoda. The coordination of multiple behaviors for a mobile robot acquired by vision-based reinforcement learning (in Japanese). Report SIG-FAI-9403-4, JSAI, Mar. 1995.

[8] Minoru Asada, Shoichi Noda, and Koh Hosoda. Non-physical intervention in robot learning based on life method. In Proc. of Machine Learning Conference Workshop on Learning from Examples vs. Programming by Demonstration, 1995.

[9] M. Asada, S. Noda, and K. Hosoda. Action-based sensor space categorisation for robot learning. In Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems 1996 (IROS96),pages 1502{1509, 1996.

[10] E. Uchibe, M. Asada, and K. Hosoda. Behavior coordination for a mobile robot using modular reinforcement learning. In Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems 1996 (IROS96), pages 1329{1336, 1996.

[11] https://www.bundesliga.com/en/bundesliga/news/soccer-positions-explained-names-numbers-what-they-do-2579-786

[12] Efecto relativo de la edad en fútbol profesional: influencia del nivel competitivo y la posición Relative Age Effect in Professional Football: Influence of Competitive Level and Playing Position - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/Football-pitch-showing-the-distribution-of-the-specific-positions-of-the-players_fig2_336210275 [accessed 15 Feb, 2023]

[13] Buckland, M. (2005). Programming game AI by example. Jones & Bartlett Learning.

[14] A multi-agent reinforcement learning approach to robot soccer | Semantic Scholar

[15] Onağ, Zeynep & Tepeci, Mustafa. (2014). Team Effectiveness in Sports Teams: The Effects of Team Cohesion, Intra Team Communication and Team Norms on Team Member Satisfaction and Intent to Remain. Procedia - Social and Behavioral Sciences. 150. 420-428. 10.1016/j.sbspro.2014.09.042.

[16] Adaptive Leadership in the Military Decision Making Process (army.mil)

[17] Sancho, P., Moreno-Ger, P., Fuentes-Fernández, R., & Fernández-Manjón, B. (2009). Adaptive Role Playing Games: AnImmersive Approach for Problem Based Learning. Educational Technology & Society,12,(4), 110–124

[18] Corchado, J. M., Lees, B., & Rees, N. (1997, February). A multi-agent system "test bed" for evaluating autonomous agents. In Proceedings of the first international conference on Autonomous agents (pp. 386-393).

[19] Buşoniu, L., Babuška, R., & De Schutter, B. (2010). Multi-agent reinforcement learning: An overview. Innovations in multi-agent systems and applications-1, 183-22