**CHAPTER II**

**REVIEW OF RELATED LITERATURE**

**2.1 COVID-19 Epidemic**

The authors Maleki, Valipour, Ashrafi, Mokari, Jamali, Lucas (2008) defined the actions that are implemented by the soccer server. These commands are drive, kick, pantilt, say and catch. Drive function moves the agent (robot) by applying the force vector to center of it. Kick function kicks the ball if it is in a kickable distance. Pantilt function changes the view direction of an agent. Say function sends a message to all the players. Catch function (for goalkeepers only) holds and freezes the ball if the ball is in the catchable area.

High-level skills are those in which the world model information and the basic skills are being applied. These skills consist of the actions with ball like pass, shoot, dribble, etc and offball actions like mark, find object and information broadcasting.

Maleki et al (2008) defined that to perform those actions, we need some information about the ball treats, when a force with a particular angle is applied, considering the environment parameters. A number of complicated formulas could help predict the ball movements.

Shooting is the most important skill in soccer and all other skills are based on it. A player in this environment can only kick the ball in front of himself, so he needs to get behind the ball in a correct position to shoot the ball in his desired diretion.

In passing, the agent kicks the ball so that the other teammate can receive it. Maleki et al (2008) classifies passing into secure pass, normal pass and risky pass. Secure pass is when a player is completely sure that his pass will arrive to the player he wishes. Normal pass has a probability of success than a failure. Risky pass is the case where the probability of ball arrival exists, but the possibility of failure is more than its success.

The agent uses the skill dribble when he owns the ball in an almost free space and can’t find the other agents with better positions or can’t pass them the ball. The agent can also use clear ball when he owns the ball and can perform no other action or the player is in dangerous situation.

Actions without the ball make the agents to be arranged in positions so that they would have the most chance to create opportunities for team or to get the opponents opportunities. According to Maleki et al (2008) mark skill approaches two purposes, which is not to let the ball reach the opponent and not to let the opponents shoot to their desired position. Say skill is being used for alerting the agents and also to update their world model.

Guerrero, Ruiz-del-Solar, Diaz (2008) proposed a ball control action which is what the robot does after catching the ball and it consist in a relative displacement and rotation of the robot holding the ball and a kick of the ball.

**2.2 Laboratory Management**

Yong, Bao, Xin (2011) presented that one of the commonly used fast algorithms for path finding is a Rapidly-exploring random tree (RRT) algorithm. The RRT algorithm is designed for efficiently searching non-convex, high dimensional search spaces. RRTs incrementally reduce the expected distance of a randomly chosen point to the tree. The RRT algorithm is extremely simple and cheap to calculate but it is not optional. A path will be computed quickly but it is not guaranteed to be the cheapest, and will result different path for every search.

Gonzales, Taha (2007) used a fuzzy controller to determine the path of the mobile robot to the goal while avoiding any obstacles that it will face. The controller’s output will be the movement of the robot by controlling for both its left and right wheels. The two wheels are controlled by two independent fuzzy logic systems. Although these two systems are independent of each other, they have the same inputs which are the difference between the angle orientation between the robot and destination point, and the distance with respect to a common axis which is the center of the robot and desired destination. Another feature of the mobile robot is obstacle avoidance while the robot is on its way to the goal. The system initially finds all obstacle positions. If an obstacle is situated along the path the robot to the goal, then the robot would either turn left or right depending on which is the shortest distance to the goal. Obstacle detection is done by first monitoring all obstacles if they are within the range of the mobile robot’s path.

Janglova (2004) presented a motion planning algorithm that will move a robot safely in environment on the base of the data “visible”. Mapping of the workspace from the measured data and find the free space for robot motion and then determines the next robot azimuth for the safe step to the goal. Janglova (2004) used neural network technique to solve this problem. The measured data is used in the learning workspace for mapping the from robot workspace by the first neural network finding the free space segment. This segment is used as an input to the second neural network both with the goal location, which is used to determine the location of the proposed next navigation step for the moving robot. The algorithm is of an iterative type. In each iteration, the last orientation of the moving robot is stored and the neural network selects the direction of the next navigation step. To determine the direction, the status in the partial configuration space is required; the map from the range finder is proposed to give this status. Moreover, a control unit is used to provide information required by neural networks to control the operating sequence and to check the reachability of the goal configuration from the start configuration.

**2.3 DOLE Advisories**

The agent ability of performing an action according to the environment situation is tested by Decision Makers to confirm that action can be done by agent in that condition. These decision makers enumerated by Maleki et al (2008) are the decision making for shooting to a position, decision making for shoot to goal, decision making for pass, decision making for dribble and decision making to marking. Maleki et al (2008) used fuzzy rule base for all decision making implementation. The major issues in the decision making phase are the static assignment of roles and dynamic team strategy so Maleki et al (2008) adopted a formation/role system.

Guerrero et al (2008) implemented that teammate robots share their own estimated positions, the observations of the ball and of the other robots and each robots all the mobile objects using an EKF based approach. They proposed that any situation of the game may be evaluated in terms of how advantageous it its. In the moment where a robot holds the ball, it has infinite possible Ball Control Actions (BCA) that should be evaluated in order to decide for the best.

Yong et al. (2011) suggested that action prediction based on probabilistic neural network can be used to reinforce the learning approach to robot soccer. Probabilistic Neural Network (PNN) is a kind of neural network model used for classification. In this model the Bayes decision-making analysis of Parzen window estimate is performed by the neural network structure. PNN is developed from the Bayes decision strategy of multiple variables pattern classification. PNN can perform Bayes decision strategy and nonparametric estimators of the probability density function. It is used to classify patterns based on stored samples.

Authors Ball and Wyeth (2003) presented a study of classifying opponent’s behaviour in robot soccer. They divided their proposed system into three stage. In the first stage they used Naïve Bayesian Classifier to classify and recognise the opponent’s current behaviour. They indicated that the Bayesian Classifier is comparable to an expert designed fuzzy classifier in terms of performance. On the second stage they modelled the attributes the way which the opponents act or perform their behaviour. The modelled attributes will then be used predict the future. Last stage is the integration of the predictions into current multi-agent planning system to treat the opponent robots as agents with own preferences.

Abiyev, Bektas, Akkaya, Aytac (2013) implemented behaviour tree (BT)-based decision making where each tree is assigned a goal that will be achieved. The robot behaviour is a control law that satisfies a set of constraints to achieve a particular goal. Each behaviour defined by the set of actions. A BT enables modularity, making states nested within each other and thus forming a tree like structure and restricting transitions to only these nested states. The root node branched down to the tree until the leaf nodes are achieved. The leaf nodes are the based actions that define the behaviours. A BT is made up of three types of nodes, actions, decorator, composite. Composite and decorator nodes are used to control the flow within the tree and action nodes. Actions are used to change states such as calculating a new path or kicking the ball.

Composite nodes include set of nodes such as selector, sequence nodes, and their parallel and random versions. Selector and sequence are workhorse internal nodes. A sequence represents series of behaviours that we need to accomplish. A sequence will try to execute its children from left to right. If all of its children succeed, sequence will also succeed. If one of its children fails, sequence will stop and return failure.

Selector node will try to execute its first child. If its first child returns success, it will also return success, it will also return success if the child fails, it will try executing its next child until one of returns success, or the node runs out of children at which point the node will return failure.

Maravillas, Dadios (2007) divided the global goal into two main tasks, offense and defence-states. Each of these tasks is further divided into subtask that are individually executed by the agents. Each agent is given functions to accomplish these subtasks and does implicit communication and coordination with other agents in the multi-agent system. The fact that no two or more agents assume the same subtask shows some type of communications between agents. An exploration of the environment combines the actions of all agents at the highest level of hierarchy. The hierarchical multi-agent cooperation strategy allows the agents to master the skills of coordinated/joint execution of the main goal. Each agent concentrates on executing it, individual duty and subtasks level. Centralized control is achieved by means of global information: the agents individual actions are always in conjunction with the team’s main goal. Thus, confusion resulting from conflicting goals will be avoided. Individual positions of robots and balls are made available by the vision system of the robots. Given these data, information that is of primary importance for the performance of the robots is extracted at the highest level of hierarchy. As soon as the data is available it is broadcast to all robots. As such, execution of low level actions, such as blocking and intercepting the ball, are straightforward and faster.

Li, Chen, Sun (2001) validated the effect of Adapted Neuro-Fuzzy Inference System (ANFIS) using shoot. The decision maker decides whether or not to perform this action according to the possibility of a goal. The success rate of shoots is related to the states of other agents, such as position of the shooter, the positions of the opponent defenders, the defending ability of the opponent defenders, the position of the opponent goalie, and its defending ability.

References

Ball, D., Wyeth, G. (2003). Classifying an Opponent’s Behaviour in Robot Soccer. *Proceedings of the 2013 Australian Conference on Robotics and Automation, ACRA*. Australia

Alpaydin, E. (2010). Introduction to Machine Learning Second Edition. *The MIT Press.* Cambridge, Massachusetts.

Russel, S., Norvig, P. (1995). Artificial Intelligence: A modern Approach. *Pretence Hall.* Englewood Cliff, New Jersey.

Lobato, J. M., Adams, R. (2015). Probabilistic Backpropagation for Scalable Learning of Bayesian Neural Networks. *Proceedings of the 32nd International Conference on Machine Learning.* Lille, France

Maleki, K. N., Valipour, M. H., Ashrafi, R. Y., Mokari, S., Jamali, M. R., Lucas, C. (2008). A Simple Method for Decision Making in RoboCup Soccer Simulation 3D Environment. *Revista Avances en Sistemas e Informatica 5(3)* (pp 109-116).

Guerrero, P., Ruiz-del-Solar, J., Diaz, G. (2008). Probabilistic Decision Making in Robot Soccer. In U. Visser & F. Ribeiro & T. Ohashi & F. Dellaert (Eds.), *RoboCup 2007: Robot Soccer World Cup XI* (pp 29-39). Berlin: Springer.

Yong, D., Bao, X. C., Xin, H. Xu. (2011). A Multi-Agent Reinforcement Learning Approach to Robot Soccer. *Artificial Intelligence Review* (pp 193-211). Netherlands: Springer.

Abiyev, R., Bektas, S., Akkaya, N., Aytac, E. (2013). Behaviour Trees Based Decision Making for Soccer Robots. *Recent Advances in Mathematical Methods, Intelligent Systems and Materials* (pp. 54-59). Cyprus.

Gonzales, J., Taha, Z. (2007). Mobile Robot Navigation Using Open Computer Vision with Fuzzy Controller. *Journal of Advanced Computational Intelligence and Intelligent Informatics* (pp. 336-341).

Janglova, D. (2004). Neural Networks in Mobile Robot Motion. *International Journal of Advanced Robotic Systems Volume 1 Number 1* (pp 15-22).

Maravillas, E. A., Dadios, E.P. (2007). Fira Middle-League Robot Soccer Game Strategy. *Control and Intelligent Systems Vol. 35* (pp 377-384).

Shi, L., Chen, J., Ye, Z., Sun, Z., Learning Competition in Robot Soccer Game based on an Adapted Neuro-Fuzzy Inference System. *Proceedings of the 2011 IEEE International Symposium on Intelligent Control* (pp/ 195-199).