Robocup SLL Strategy Group 1

Adrian Swande*, Oskar Frej†, Gustav Samuelson‡, Ivan Blazanovic§, Lukas Bonkowski¶
School of Innovation, Design and Engineering, M.Sc.Eng Robotics
Mälardalens University, Västerås, Sweden
Email: *ase22003@student.mdu.se, †ofj22001@student.mdu.se, ‡gsn22003@student.mdu.se,
§lbi25001@student.mdu.se, ¶ibc24003@student.mdu.se

***Abstract*—An abstract should summarize the work in brief.**

***Index Terms*—Alphabetical, Be, In, Order, Should**

I. Introduction

The RoboCup [**?**] is a tournament where different teams compete against each other with soccer playing robots. The RoboCup Federation arranges several types of leagues where every league uses different types of robots in different shapes and sizes. Overall, this tournament aims to advance in the scientific field of mobile robots. This project will focus on the Small Size League (SSL), division B in particular. In the SSL division B teams compete in 6 vs 6 matches of two halves where each half is five minutes long with a five minute pause in between. The robots are constrained to certain physical dimensions according to the rules (the robots need to fit inside a cylinder of 0.18 meters width and 0.15 meters height) and the robots are built by the members of each team. The playing field is 10.4 times 7.4 meters with a playing area of 9 times 6 meters and the game is played with an orange golf ball. The rules of this league are similar to regular soccer but with several modifications. For example the rules include yellow and red cards, freekicks and penalties but also rules like maximum shooting speed and maximum dribbling length. The aim of this project is to develop a system that works well in simulation. That will be done by creating an AI system that can coordinate all six robots, handle the ball, score goals and defend against the opponents. In the long term the models we develop could be further developed and used in other works related to both RoboCup and other areas.

A. Research Questions

II. Background

Autonomous Mobile Robots: According to Kate Brush [**?**], an (Autonomous) Mobile Robot is a robot that is capable of moving around and navigating through its surroundings with the help of for example software, sensors and cameras. The robots are mainly fitted with legs, wheels or tracks that are used to transport itself around, but they are also used in aerial and nautical environments. They are mainly driven by an automated AI system that is in charge of decision making. Mobile robots have surged in popularity over the recent years (partly) due to their ability to operate in areas that humans can not/should not be in.

Reinforcement Learning: Reinforcement learning: According to Jacob Murel and Eda Kavlakoglu [**?**], reinforcement learning is a machine learning algorithm that is used to develop independent decision making in autonomous agents. Agents train by repeating similar tasks over a period of time or repetitions, where they learn independently through trial and error. A popular adaptation/version of the learning algorithm is Q-learning. According to GeeksForGeeks [**?**], Q-learning is a model-free RL algorithm that is used for training independent agents to make the best decision possible in each possible situation. It learns through a trial and error system, where it interacts with the environment to find the best method. A state-action-reward system is utilized, where the result of an action taken in a state is rewarded or penalized depending on the outcome. After a training iteration it writes its Q-values to a Q-table, where the values represent the best known expected reward for taking a given action in a given state. It updates the table using the Temporal Difference rule: $Q(S,A) \leftarrow Q(S,A) + \alpha\,(R + \gamma Q(S',A') - Q(S,A))$. For each state, the agent can either choose to explore or to exploit. Using the Epsilon-Greedy Policy (ϵ-greedy policy), the agent decides whether to take the best current known action (exploit), where the agent picks the best action with the highest Q-value based on the probability of $1-\epsilon$. Else it will try to find a new best possible action (explore), where the probability to explore is based simply on the ϵ-value. This is what allows the model to independently over time find the best possible outcomes for each state.

Q-learning is a reinforcement learning algorithm used for leaning the value of actions in states according to some reward definition. Given a finite amount of actions and states, the algoritm can learn by experience the optimal action to take at each state to ensure the maximum total reward according to some time horizon.

In 2013, Mnih [...] proposed a variant of Q-learning called Deep Q Network, in which a neural network is used to aproximate the optimal action-value function

$$Q^*(s,a) = \max_{\pi} \mathbb{E}\left[r_i + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots \,\middle|\, s_t = s,\, a_t = a,\, \pi\right] \tag{1}$$

Q-Learning:

III. EXPERIMENTATION

A. Strategy Hierarchy

The following Hierarchy shows how we could organize Team behavior across 5 layers, from high-level strategy down to low-level execution. Each layer builds on the one above it, enabling modular, scalable control.

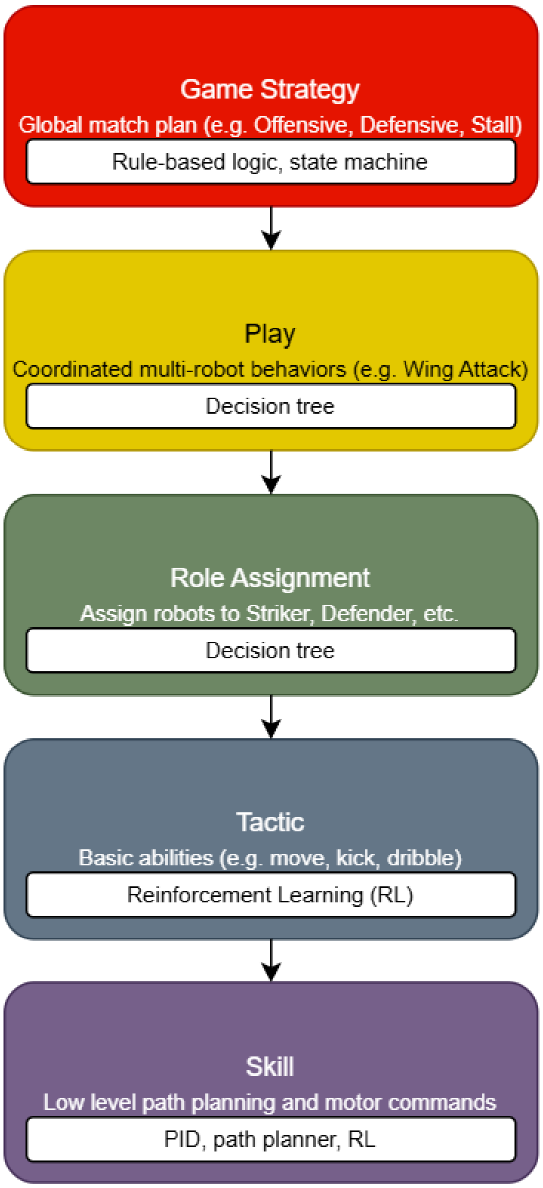

Figure 1. Hierarchical team behavior structure

1) Game Strategy: The top-layer defines the overall team behavior based on the given game state. If we take Rule-based logic for example, we could look at time left to play and score. Then if we are winning and the time is lower than a specified threshold, we could set the Strategy to Stall. This will then provide high-level context for all other decisions made below.

2) Play: The play layer selects coordinated maneuvers such as setting up a wing attack or forming a defensive wall. Selecting plays could be done by a decision tree based on factors like ball position, team formation, and opponent layout. Each play then sets constraints or goals for roles and tactics.

3) Role Assignment: This layer will dynamically assign robots to specific roles (e.g. striker, defender, goalie) based on their position, proximity to the ball, or other factors. Optimization algorithms such as Hungarian matching have been used with great success.

4) Tactic: The tactic layer defines what action a robot should take in its current role. This could be whether the robot should pass, dribble, shoot, or intercept. This layer's decisions are highly context-sensitive and reinforcement learning is a good choice.

5) Skill: The skill layer handles the low-level physical execution of actions. This could be moving to a position, kicking (how hard) or dribbling. Commonly used control methods are PID and path planning, but reinforcement learning can also be used to improve fine motor control, adaptability, or performance in unpredictable situations.

B. Training

To prepare our agents for RoboCup SSL, we are currently exploring VMAS (Vectorized Multi-Agent Simulator) as our training environment. VMAS is a lightweight and very customizable simulator to prototype and train multi-agent behaviors. We can set up custom scenarios that simulate different game situations. We can then use it to train both the Tactic and Skill layer behaviors using RL algorithms. In each scenario, we define the agent's observations, actions, and reward signals to shape the behavior we want them to learn. VMAS also supports fast, parallelized simulation with GPU acceleration, which allows us to efficiently run many environments at once. At this stage, we are still experimenting with different training approaches. The goal is to eventually transfer the trained policies to a more realistic simulator like grSim, where we can evaluate their performance in a full game setting.

C. Agent Architecture: Single vs Multi-Agent

At this stage, we are still exploring whether to model our AI system as a single-agent or a multi-agent architecture.

A **single-agent** approach would involve one central controller that receives the entire field state and outputs coordinated actions for all robots. This method simplifies coordination and is often easier to implement and train.

A **multi-agent** approach would assign each robot its own agent, possibly with limited field knowledge. This approach is more realistic and can model decentralized behavior, but introduces complexity in coordination and learning stability.

Our initial focus will likely lean toward the single-agent model to reduce complexity during development. However, we may transition to or experiment with a multi-agent setup depending on performance and scalability needs.

IV. METHOD

In this section, the method used to find an answer to the research questions should be presented.

If this report presents results from a literature search, this means providing sufficient information for allowing someone else to repeat the literature search and compare the results. I.e., a search using the phrases a, b, and c, was made in database x, y and z on the date Month Date, Year (e.g., July 31st, 2021). The search resulted in x hits. Then, information on how you chose which works to include in this report should be provided. The references should be used for answering your research questions.

If the work reports on an experiment, this part should provide information about the experimental setup, how the experiment was conducted, how data was collected and analyzed etc. Motivate methodological choices through references. Also an experiment should be presented with sufficient detail such that it can be repeated by someone else.

V. Results

This section should present answers to all research questions.

It is normal to have only one results section, but you can create more sections if finding it more appropriate. You can also divide results into subsections. Perhaps you want to refer to some other section, for example (see Section IV). You can also place figures, you should always reference these in the text, see Figure **??** for an example of a figure including subfigures. Remember that all figures should have a figure label explaining their content.

VI. Discussion

TEST!!!!! Here you can discuss your results, limitations and new questions that have arose while doing the work. Depending on the size of this report, you can present discussion and conclusion in one common section or in two separate ones.

VII. Conclusion

Acknowledgment

The authors would like to thank ... for his/her/their help and support during the process of writing this paper.