

Multi Agent learning in adversarial settings

Proposal for B.Tech Project (CS498) by Rohit Vaish (Y6402) & Sandip Kumar Gupta (Y6425)

Under the supervision of Dr. Amitabh Mukerjee & Dr. K.S.Venkatesh

1 Problem Statement

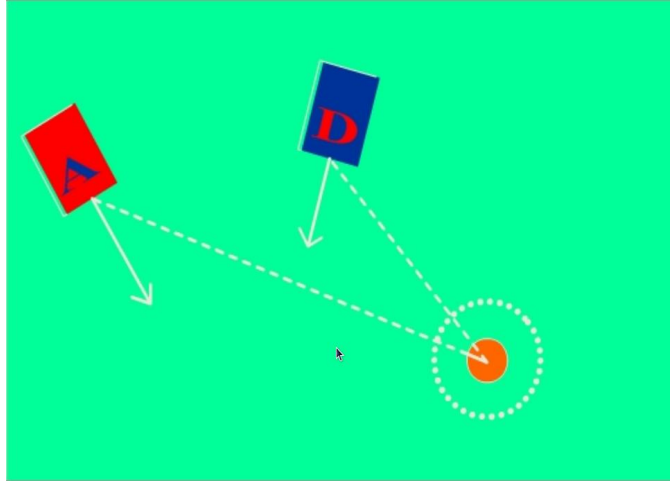


Figure 1: Arena setup for the problem statement. The attacker and defender robots are shown with the “A” and “D” tags respectively. White lines depicting their distance from the ball and direction of heading are also shown.

1.1 The Setting

2 Motivation for the problem statement

The problem statement straightaway forms an important issue encountered in Robosoccer, whereby the defender shall have to prevent the attacker (this time with the ball) from advancing towards the goal (or another player). But an even more important reason for conducting such study stems from the fact that this problem beautifully captures the idea of *stochastic learning in a multiagent adversarial setting*. Here, the action selection of each agent essentially depends on the *joint* action of all the agents. The transition of each agent from a given state to the next state depends on the current states and actions of other agents, and the rewards/penalties (as a result of choosing one such action) are then appropriately distributed. Hence, each agent tries to maximise its total reward by choosing the “most appropriate” action in the given setting, and continues to learn in this process.

3 Simultaneous Multi-robot learning

We model the robot learning strategies based on the following two heuristics:

- The attacker should try to move as *close* to the ball and in the *least* amount of time.
- The defender should try to keep the attacker away from the ball for *as long as possible*.

Before assigning the rewards and penalties for individual robots, we define following two functions:

- **Distance Potential Function**, $f(r) \rightarrow$ The reward function for the attacker that increases with decreasing distance of the attacker from the ball.
- **Timing Potential Function**, $g(t) \rightarrow$ The penalty function for the attacker that increases with increasing time taken by the attacker to reach the ball.

The exact form of these functions shall be discussed later.

3.1 Rewards/Penalties for Attacker

- The attacker is assigned a reward R_0 for successfully accomplishing the task of reaching the ball within the given time frame. Similarly, an equal penalty is inflicted in case the task is not carried out.
- A reward $f(r)$ is assigned to the attacker, depending on its distance from the ball. This motivates the attacker to move as close to the ball as possible.
- A penalty $g(t)$ is inflicted on the attacker, depending on the time it takes to accomplish the task. The more the time, the more is the penalty. This urges the attacker to finish the job in least time.

So, the attacker reward function can be written as:

$$R_A = \pm R_0 + f(r) - g(t) \quad (1)$$

3.2 Rewards/Penalties for Defender

- The defender is assigned a reward R_0 for successfully accomplishing the task of preventing the attacker from reaching the ball within the given time frame. Similarly, an equal penalty is inflicted in case the task is not carried out.
- A penalty $f(r)$ is inflicted on the defender, depending on attacker's distance from the ball. This motivates the defender to keep the attacker as far away from the ball as possible.
- A reward $g(t)$ is assigned to the defender, depending on the time for which it successfully keeps the attacker away from the ball. The more the time, the more is the reward.

So, the defender reward function can be written as:

$$R_D = \pm R_0 - f(r) + g(t) \quad (2)$$

Such a modelling of reward functions is primarily based on *Reinforcement Learning*, whereby the agents take actions in a environment so as to maximise their long term reward. It should , however, be noted that the above strategy selection has been chosen to keep the given problem commensurate with a *zero sum game*, one in which gain of one is an equivalent loss of the other.

4 Further extensions

We plan to make the given problem statement more involved through the following modifications:

- The no. of attackers and defenders can be increased to two each, in which case the action of an agent will depend on not one, but three other agents. Moreover, the attackers & defenders can collaborate with each other to carry out their tasks. This can be an excellent way of studying *cooperative* multi-robot learning in an adversarial environment.
- Some obstacles such as walls etc. can be planted on the arena, to restrict the motion of agents and invoke the need of path planning.

5 References

1. [Michael Bowling and Manuela Veloso, 2002], *Simultaneous Adversarial Multirobot learning*.