MRL-SPL Team

Technical Report

Task Assignment Module

Author(s):
Novin Shahroudi, Ali Piry

February 2, 2018

Contents

1 Introduction **2**

2 Implementation **2**
2.1 Formation System 2
2.2 Tasks 4
2.3 Time cost and assignment 5
2.4 Game static rules 6
2.5 Synchronization 6
2.6 Comm-less Scenario 7

3 Discussions **8**

4 Conclusion **9**

5 Future Works **10**

1 Introduction

Task assignment module is being developed and employed since 2015 competitions. Since then we have improved the module by adding new features. This module is mainly inspired by the "Positioning to Win" [1] at the Simulation3D league. Our main motivation was to see how an approach could be adapted from a simulation setting to SPL. Furthermore having a general framework for multi-agent coordination is an essential part of good soccer playing team of robots. Our main contributions were the adaptation to the SPL, time cost measurement instead of distance, and our own behavior structure. The building blocks of the implementation are being discussed in section 2. We will discuss challenges and benefits of the work in section 3. Conclusion and future works is presented respectively in section 4 and 5.

2 Implementation

Task Assignment consists of some building blocks which based on relevance we cover them here. Except for the formation which depends on an external tool (*PlanEditor for manipulation of the formations*), all other parts implemented as a part of the task assignment module. Fig. 1 shows the interaction of the task assignment with other components of the behavior control system.

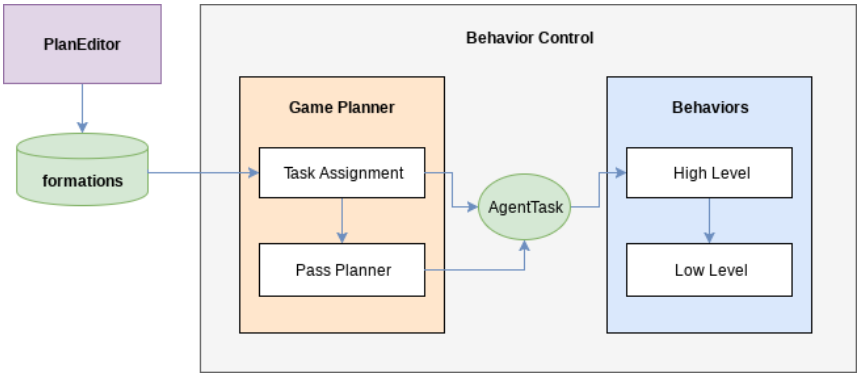

Figure 1: Task Assignment interaction with different components of the system

2.1 Formation System

We use a similar approach proposed by the original paper to define formation of the agents, however with some adaptations to SPL and realization to our behavior

control architecture. Each formation includes information about the position of the agents based on the number of players and states of the game. Formations are stored and being loaded as configuration files in the beginning of each game. Also, some parameters such as the number of supporters for a given leader in a post can be defined as a general policy of each game within the formation files.

We use a different terminology compared to the original paper. Here we have two closely related terms, role, and post. The post is the long-term task of an agent, the role is a short-term task. Each post is mapped to a position on the field. Some of the posts could be a Goal Keeper, a Defender, a Halfback, a Forward, etc. that is defined on the formation file as a static point. Similarly, a role corresponds to a position relative to the ball such as a Leader or Supporter. Role positions are not predefined in the formation, rather calculate in real-time. An agent may have a post and a role at the same time or just a post. However, an agent is supposed to do one task at a time, hence a role has priority over the post. For instance, a defender may become a leader. As long as it is a leader it stops to act as a defender. Although calculations for role and post performed separately, at the end, there is a one-to-one map between tasks of the agents and positions on the field.

Figure 2: PlanEditor, a tool for defining formations and configuration of the high level strategies in the game.

Using the *PlanEditor* we may define new formations or modify existing ones. The formation files are being loaded at the beginning on the memory and appropriate one is being loaded/switched in runtime based on a number of players or state of the game. This provides a flexibility to have different strategies in terms of formation positions and other strategic parameters in different states of the game as a general policy of that game.

Although post positions are defined within the formation, they are also overwritten in many situations to adapt to the dynamics of the game. First, all post positions

Task	Definition	Position
Leader	Closest agent to the ball	Task Assignment
Leader2	Second closest agent to the ball	Task Assignment
Supporter	Second closest agent to the ball	Low level behavior
Defender	A post on the defense line	Low level behavior
Palang	For the kickOffUs scenario	Task Assignment
Goal Keeper	Always assigned to player 1	Low level behavior

Table 1: List of currently employed tasks and their description

are changed within a range based on ball position in the field. This is depicted in Fig. 3, blue circles (posts) and vertical/horizontal arrows (the range of each post movement based on ball position). Second, most of the tasks correspond to a low-level behavior in which the position is being planned and executed. The only time that the default static post positions are not overwritten by the low-level behavior is in the ready state.

Among the tasks mentioned in Table 1 Defender position is derived in low level. although it is almost in a neighborhood of the static point defined in the formation. However, the newly calculated position is not reflected in the task assignment which is neglected due to the proximity of the derived position. It is also the same for the goal keeper, however, goal keeper is not included in the task assignment calculations. Also, for the supporter its position is being derived in the low level, however unlike the defender, since it is assigned to the module with time cost comparison it would break the integrity.

2.2 Tasks

There are different tasks in a soccer team that each agent may take to achieve the common goals. As mentioned in the formation section, each agent can take a position based on its post or role position. Fig. 3 & 4 depict result of the task assignment in which vectors v1 to v5 represent assignment of the agent. GoalKeeper (labeled GK in Fig. 4 assigned to GK post, all other players take stationary posts labeled as DF, P1, P2, P3 in case of ball absence. In case the ball is in the field and detected by the team, stationary posts will move as a function of ball position. More importantly, a leader (LD) and a supporter (SUP) role (based on strategy configurations of the game) will be assigned considering time costs calculated by the task assignment algorithm. This will be discussed more thoroughly in the next section. There are some circumstances that leader may fall behind the opponent. To that end, another agent that is closer to the ball will become the second leader together with the first

leader until the one which is closer reaches to the front of the opponent, not behind.

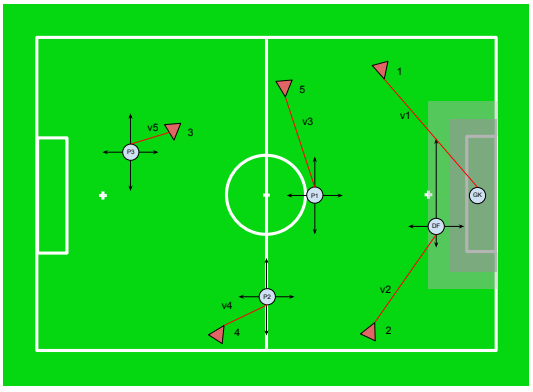

Figure 3: Task assignment in absence of the ball

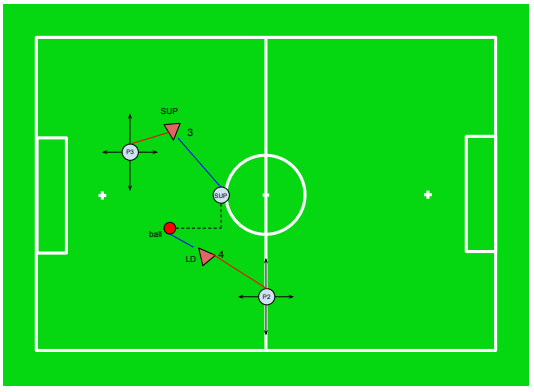

Figure 4: Task assignment in presence of the ball

2.3 Time cost and assignment

As mentioned earlier, a formation consists of a set of positions called role/post that describes each robot task in a soccer game. Each role/post is a target position. Mapping n robots (R) to n targets (T) is a linear assignment problem aiming to find minimum sum of costs where cost function C is $C : R \times T \to \mathbb{R}$ and $f : R \to T$ is a bijection of robots to targets. Eq. 1 formulates the latter. We utilize an extension of a heuristic method explained in [**?**] to solve this assignment problem.

$$\sum_{r \in R} C(r, f(r)) \tag{1}$$

In our implementation, assignments are time optimal. Fig. 5 depicts the time cost calculation criteria such as distance and rotation to the target.

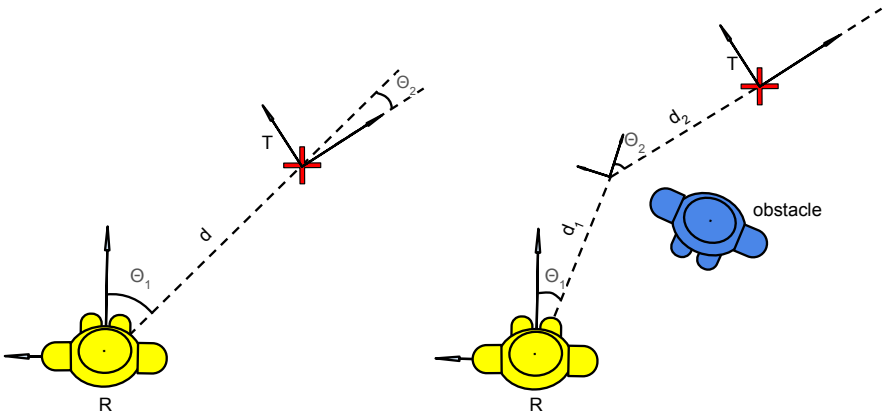

Figure 5: Calculation of time cost for robot R to target T in which d_1 and d_2 are the modified distance with respect to the obstacle, θ_1 and θ_2 are rotations required.

2.4 Game static rules

Since SPL has illegal defender rule, we decided to figure it out in task assignment. When kick off is the opponent team and game starts, team players must not get into the circle for 10 seconds. This time is tracked after whistle detected. If it was opponent kick off, leader approaches the ball and yet steps in place until either the ball is moved or 10-second is over. We also try to get the benefit of the kick off as an opportunity to score a goal. When kick off is with us, the leader player passes to the "palang" robot who goes to his position in opponent half and "palang" gets the ball and kicks to goal. "Palang" is a post position in the formation that is placed close to the slide lines in the ready and when the game goes to playing it is placed in the opponent's half so that it increases its chances to receive a pass. The passing mechanism is part of the pass planner module and is out of the scope of current report, however it worth to mention that the pass planner works in a passive fashion so that every agent finds best passing options to other agents based on different circumstances and stages of the game so that no explicit communication is required between these two modules.

2.5 Synchronization

Synchronization can play an important role with respect to the quality of the coordination. Due to different disturbances in the system such as in communication or low-level input data the outcome of the task assignment can be inconsistent. Hence,

as proposed by the original paper we also use voting to make the system robust using a majority's consensus on different decisions. In a sense, it aggregates decisions such that in execution it is almost a uniform global decision. However, it is incapable of some problems raised by the communication medium. For example, the voting is ineffective when teammate robots divide into groups (Fig. 6). They can communicate within the group but not with robots from the other group(s). To this end, we have two group of robots each coordinate within each other. However, in the worst case, there will be only two unsynchronized groups in the short term, but as time passes this can lead to a total unsynchronized team. Unfortunately, we do not have any empirical data to prove that such scenarios may occur, however, based on RoboCup experience it seems that such cases exist in such wifi-crowded hostile environments.

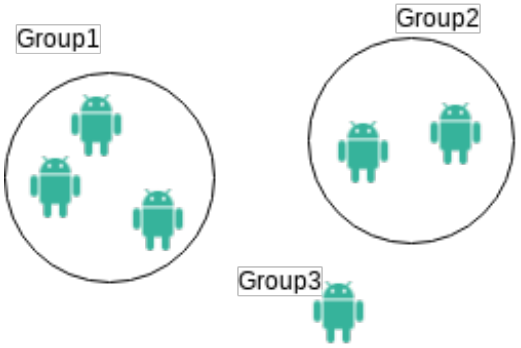

Figure 6: Unfortunate communication scenario

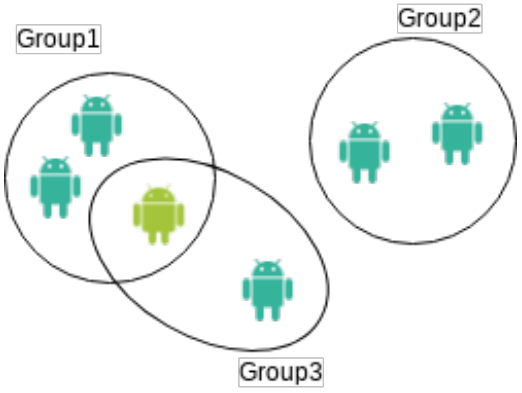

Figure 7: Unfortunate communication scenario with presence of captain

In our implementation, anything that requires synchronization can be voted and decisions can be made based on consensus. An agent is picked as a captain (Fig. 7) by consensus that its vote has higher weight compared to the others. This agent is chosen based on its network connectivity and other factors that make it a more stable one in the game. Voting introduces additional overhead in the transmitted network packet. Fig. 8 shows the task assignment pipeline with presence of voting mechanism.[1]

2.6 Comm-less Scenario

Inconsistencies in a robust communication channel over the Wifi provided by the access points impose challenges for the multi-agent coordination. These can be in

[1]We did not use voting synchronization in the RoboCup 2017.

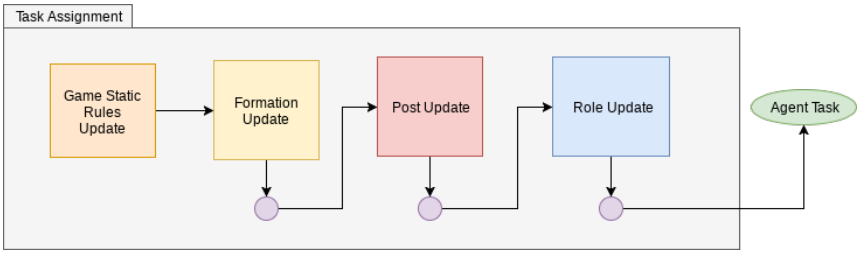

Figure 8: Task Assignment Pipeline, circles show the synchronization for the process above

form of delayed packets, or totally jamming of the network. Since the formation is changed based on the number of players inconsistencies influence the coordination quality drastically. Although voting may help, in some sever situations it's ineffective (such as total network jamming). Our attempts at determining such scenarios on-fly and switching the formation were not successful. One of the challenges was the fact that agents may have a different view on the number of players in the game. This may be due to network latencies and other underlying communication issues. To this end, currently, the comm-less scenario is a configuration for the deployment. In the firsts scenario, agents are assigned statically to their posts defined on the formation. At the same time, the role (LD/SUP) can be assigned dynamically. In this scenario agents only play based on three main information: 1. the ball position, 2. their own Voronoi cell, and 3. assumption of the number of players based on configuration. The other scenario is a total static game where agents only interact with the ball when the ball is in their region. This can be a solution to the inconsistencies in the connected number of players where we experience an instable communication, however, it comes with the cost of playing in a complete static manner.

3 Discussions

Current approach uses the exact position of agents and ball to calculate time-costs for task assignment in a distributed fashion. Inconsistencies in the reported robot position for each agent introduces major inconsistencies in the performance of task assignment which its result would be oscillations in decisions. Furthermore, some assumptions about the movement of the robots may not be accurate since in reality because of some obstacles or physical difficulties robots may fail to follow a determined path. Voting compensates such deficiencies to some level however it is probably beneficial to use an encoding of the problem such that coordinates and exact position are not employed directly.

In the task assignment level, the opponents' position and their role in the game are not considered which in many circumstances leads to low quality of coordination against an opponent. One of the best examples is when the leader falls behind the opponent, yet it is the closest robot compared to other teammates so it remains the leader, however, this assignment is ineffective especially if the opponent is faster than us. We have introduced some hacks such as a second leader to overcome these circumstances but they are hard coded and cannot be considered as an ultimate solution.

Having different formations for different game states and number of players can be beneficial in terms of defining direct policies, however, it makes them hard to maintain and less flexible. Also, the separate post and role assignment bring some level of flexibility for the comm-less scenario, however it breaks the integrity of the system. Furthermore, the former, and latter create a level of complexity for synchronization as it requires different synchronization on each of them which may lead to potential integrity issues as well.

In the current implementation position of the agents are derived in the low-level behavior skills where all the motion planning occurs as well as the execution of the motion commands. So that the positions in the task assignment level are being treated as abstract and approximate position. Although this separation works well it can be a source of inconsistencies as well as integrity since it does not reflect the newly calculated position back into the assignment calculations.

4 Conclusion

Although we did not test and use the voting implementation during RoboCup 2017 we have had almost a stable game play. That was mainly due to stable input data and good communication link. The complex scenario is often come useful where the communication link is not reliable. We have had a match with static post and dynamic role assignment. Our experimental results show that although underlying assumptions about the SPL environment is not totally met, the adaptations could reach us to a fair result. Now having all levels of behavior control system developed by ourselves gives us a good insight into more effective contributions to each module. Considering developments in other relevant fields seem to have promising methods for multi-agent coordination in a self-organizing manner. Neuro-evolutionary approaches could be of particular attention for the future to explore the multi-agent coordination fronts with.

5 Future Works

The formation formulation can be improved to make it more usable for dynamic scenarios. At the same time, other multi-agent coordination algorithms can be studied and evaluated to cover shortcomings of the currently employed approach. A good approach would be the one that does not depend on the position of agents directly.

- generalize formation system
- fixing issues concerning the integrity of the task assignment
- considering on approaches such as distributed auction algorithm or deep learning
- integration of the task assignment with passing mechanism
- quantitative assessment of the performance especially the synchronization rate

References

[1] Patrick MacAlpine, Francisco Barrera, and Peter Stone. Positioning to win: A dynamic role assignment and formation positioning system. In Xiaoping Chen, Peter Stone, Luis Enrique Sucar, and Tijn van der Zant, editors, *RoboCup 2012: Robot Soccer World Cup XVI*, pages 190–201, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.