AIUT3D 2017 Soccer Simulation 3D League Team Description Paper

Navid Hoseini Izadi, Mohamad Roshanzamir, Mahdi Tajmir Riahi, Mahdi Abbasi Soureshjani, Seyed Yagoub Ashkoofaraz, Mohammad Amin Fahami, Maziar Palhang

Artificial Intelligent Laboratory,
Department of Electrical and Computer Engineering,
Isfahan University of Technology, Isfahan, Iran

palhang@cc.iut.ac.ir

Abstract. In this paper we describe AIUT3D soccer simulation team activities and achievements briefly. We will describe our agents' skills and their behavior in team working mechanism. Our code is based on UT-Austin-Villa base code. We have modified some aspects of the base code for example the original kick skill is replaced with a new one. Moreover, we have optimized our new kick skill using CMA-ES algorithm. We have also implemented a collision avoidance mechanism using potential field algorithm. Our team also utilizes a basic ball passing mechanism to enforce cooperation between teammates.

1 Introduction

RoboCup 3D soccer simulation is a client server platform. The server communicates with the clients via message passing. The clients perceive the simulated environment state using the received messages. These messages contain visual and vocal information about other players' positions, ball and goal positions, field borders etc. Based on these information, the clients must make appropriate decisions and send them to the server. This process is repeated every 20 ms.

As this field is a simulated environment, the research teams can work on it with low costs. Moreover, the researchers do not need to get involved with the low level hardware details of real robots. Like the 2D platform, this platform provides a multi-agent environment in which agents must cooperate with each other to achieve a common goal. Unlike the 2D platform, the basic skills like walking, kicking, diving, etc need to be designed and implemented by the team developers. This will cause the 3D platform to be more challenging compared to the 2D platform. However, the 3D platform is closer to the real world. Consequently, the achievements in this platform is more likely to be useable in the real world scenarios which is the ultimate goal of the RoboCup competitions. Currently, RoboCup 3D soccer robot of choice is Nao.

AIUT3D Soccer Simulation team is developed by a group of students of Isfahan University of Technology. The team has been working on UT-Austin-Villa base code [1] for a year. Instead of implementing the team code from scratch, we decided to use an existing base code so that we can catch up with other teams faster. Although our long term development plan consists of an original code design and implementation.

Currently, our team uses the basic skills of UT-Austin-Villa base code such as omnidirectional walk, getting up and different kick types. Not only we have optimized these skills, but also we have designed and optimized a new long range kick skill.

In the remainder of the paper, the basic skills are described in section 2. Section 3 discusses our results on our own kick skill. In section 4, our high level strategy is presented. The last section describes our future goals.

2 Agent basic skills

The agent basic skills are walking, getting up and kicking. These skills are the minimum requirements for a 3D soccer team.

2.1 walking and getting up skills

The first and the most important skill required in 3D platform is walking. Currently, an omnidirectional walk engine has been implemented by UT-Austin-Villa team [2]. The engine is released in the base code but to be competitive with top teams, it needs to be optimized. Our team uses the basic omnidirectional walk engine.

It is highly probable that the robots fall down during walking or kicking due to collision with other robots. So it is necessary to have a quick getting up mechanism. The robot may fall on its back or its front. In both cases the robot must have a mechanism to get up. Again our team uses UT-Austin-Villa getting up mechanism.

2.2 kicking skill

After walking, the kicking is the most important skill. This is because the primary method to create cooperation between teammates is ball passing. To perform ball passing, we need to carry out accurate and fast kicks. More importantly, to score a goal we need a strong kick.

UT-Austin-Villa base code has a key frame based skill execution mechanism. The kick skill has some parameters that we have optimized using CMA-ES algorithm [3]. While the base kick skill has a short range, it is useful for ball passing.

3 Designed kick skill

As it is mentioned before, one of the most important skills for a 3D soccer team is long range kick. UT Austin Villa base code has a short range kick about six meters. While this kick is useful to get familiar with skill execution of the base code, it is not strong enough to threaten the opponent goal. This drove us toward designing a kick with longer range. The designed kick had a range of 10 meters. After optimization using CMA-ES [3], the final range became about 13 meters. The designed kick consists of a set of key frames. Each key frame specifies the target value for robot joints. A key frame can contain any subset of robot joints. Moreover, each key frame has a time variable dictating the admissible time interval within which the target angles are to be reached.

Since the body type of Nao robots are different, the optimization has been performed to tune the joint values, time factors and robot offset w.r.t the ball for each body type, separately.

4 Team strategy

One of the important requirements in 2D and 3D platforms is team coordination and cooperation. The 2D teams have achieved good level of cooperation but this is not the case in 3D platform. The reason for this shortcoming is lack of robust basic skills in 3D platform.

AIUT3D team can utilize potential field obstacle avoidance algorithm to enforce cooperation and coordination implicitly. Using this method, agents are capable of the following behaviors:

1. Tracking the ball
2. Keeping a safe distance from teammates in order to avoid collision
3. Moving toward desired positions like opponent's goal in offense mode and our own goal in defense mode.

Since team work is crucial to the ultimate team success, we tried to incorporate some sort of basic ball passing to achieve a level of team cooperation. Some ad hoc rules may apply as well, and these are in the test stages.

4.1 Potential field algorithm

One of the simple yet robust obstacle avoidance algorithms is potential field [4]. This method relies on a set of repulsive points and an attractive point. The environment obstacles form the repulsive points. The robot must reach the target point which is the attractive point. After perceiving these points using robot sensors, the algorithm considers an artificial potential field around each of the points which is modeled by a potential function U(q). The force vector F(q) acting on point q can be computed as follows [5]:

$F(q) = -\nabla U(q)$

where $\nabla U(q)$ is the gradient vector of U at point q. The potential function U(q) acting on the robot is the sum of attractive and repulsive potential functions:

$U(q) = U_{att}(q) + U_{rep}(q)$

So the force vector F(q) will be the resultant vector of repulsive and attractive force vectors:

$F(q) = F_{att}(q) - F_{rep}(q) = -\nabla U_{att}(q) - \nabla U_{rep}(q)$

Now the robot must move along F(q) until it reaches the target point.

4.2 Potential field for team coordination

Our strategy module makes use of potential field algorithm to enforce coordination between teammates. To do so, the repulsive and attractive points need to be defined. Based on the game status which could be offense or defense and the ball owner, these points will be different. In offense mode opponent players are considered as repulsive points since the teammates must avoid them on their way to the opponent's goal. In defense mode, the opponent player who owns the ball is considered as attractive point.

The teammates are always considered as repulsive points since we do not want them to collide with each other. The opponent goal, the own goal will be attractive points depending on the game status. The only point which is always attractive is the ball.

In the original setup of potential field [4] the environment obstacles and target point are assumed to be fixed. However, as discussed above, the repulsive and attractive points are determined based on environment status and our team strategy. This means that the resultant force vector F(q) must be updated in every cycle.

Although the potential field method is useful for implicit team coordination, due to its reactive nature, it cannot address all the needs in the team strategy. This means sometimes we override the potential field decisions. For example, when a teammate is close to opponent's goal it is no longer necessary to obey F(q). Instead the teammate decides to kick toward the opponent's goal.

4.3 Basic ball passing

Whenever an agent decides to pass the ball, it needs to choose a teammate. Based on the teammates' positions in the field and their distance to the ball owner, the ball is passed to the teammate which seems to be in a better position compared to the ball owner itself. After passing the ball, the former ball owner decides to go forward or get back to its original post. In other words, each agent has a duty post to which it will return whenever no better action is possible e.g. when it is too far from the ball.

5 Future works

Although the potential field module is very useful for team coordination, its performance heavily depends on the right choices for attractive and repulsive points. In future we aim to determine the optimal attractive and repulsive points based on game status. Our current kick range cannot compete with top teams. So, an important part of our future work focuses on better kick optimization. The optimization of omnidirectional walk engine and the getup skill will also be considered as our future work.

References

[1] P. MacAlpine and P. Stone, "UT Austin Villa RoboCup 3D Simulation Base Code Release," in *RoboCup 2016: Robot Soccer World Cup XX*, ed. Berlin: Springer Verlag, 2016.

[2] P. MacAlpine, S. Barrett, D. Urieli, V. Vu, and P. Stone, "Design and Optimization of an Omnidirectional Humanoid Walk: A Winning Approach at the RoboCup 2011 3D Simulation Competition," in *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence (AAAI)*, ed. Toronto, Ontario, Canada, 2012.

[3] N. Hansen, "The CMA Evolution Strategy: A Comparing Review," in *Towards a New Evolutionary Computation: Advances in the Estimation of Distribution Algorithms*, J. A. Lozano, P. Larrañaga, I. Inza, and E. Bengoetxea, Eds., ed Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 75-102.

[4] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *Int. J. Rob. Res.*, vol. 5, pp. 90-98, 1986.

[5] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Introduction to autonomous mobile robots*: MIT press, 2011.