# Miracle3D: Team Description Paper for RoboCup 3D Soccer Simulation Leagu

Siying Chen , Fan Wang

Heifei Normal University , China

**Abstract.** This paper describes Miracle3D team and the developments by the we did.It also describes the changes of the architecture of the Miracle3D team. In addition, it includes researches about Omnidirectional walking and Path planning in Miracle3D.

## 1 Introduction

Miracle3D is a simulation soccer 3D team was established in 2012, and attended several competitions. Miracle 3D simulation robot soccer team participated in the 2012 state competition for the first time. Miracle3D get Anhui Robocup3D championship in 2013, the same year, get Robocup3D national first prize in China and Robocup3D IranOpen2014 the final eight. Miracle3D get Anhui Robocup3D bronze in 2014, the same year, get Robocup3D national a fourth in China.

There are some problems with our team code. These problems are what we are studying. We haven't found the good solution for some problems, such as the precise positioning of robot and ball(particle filter for localization), Long-Distance Kicking, the fastest walks. We constantly improve our code, but we have many problems can't resolve in short time. So we decided to refer to the base code UT Austin Villa released. In order to speed up development to the team, and focus on multiagent system research. now, we use the base code UT Austin Villa released, see https://github.com/LARG/utaustinvilla3d , and add strategies in the code.

The remainder of the paper is organized as follows. In Section 2 a overview of UT Austin Villa RoboCup 3D Simulation Base Code Release is given. Section 3 introduce our team architecture. Section 4 introduce team's role assignment. The future works in Section 5.

## 2 Overview Team

UT Austin Villa RoboCup 3D Simulation Base Code Release is highly modular and provides us with the flexibility to modify and develop easily.
The following features are included in the release: [1]
* Omnidirectional walk engine based on a double inverted pendulum model [2]
* skill description language for specifying parameterized skills/behaviors
* Getup (recovering after having fallen over) behaviors for all agent types

* A couple basic skills for kicking one of which uses inverse kinematics [3]
* Sample demo dribble and kick behaviors for scoring a goal
* World model and particle filter for localization
* Kalman filter for tracking objects
* All necessary parsing code for sending/receiving messages from/to the server
* Code for drawing objects in the RoboViz [4] monitor
* Communication system previously provided for drop-in player challenges 4
* An example behavior/task for optimizing a kick

## 3 Team Architecture

The low layer comprises a communication module and a receiving execution module. It acts as the last level of the layer structure, and its role is Server communication, its function includes two aspects: sending and receiving, as a receiver, the communication module needs to obtain information from the Server, send a message to the world after the analytical model; as the transmitter, Agent decision feedback to Server through the communication module, information parsed once again passed to the world model, the world model is updated according to the information from the communication module.

The skills layer is also the basic action layer, the player's basic movements and skills defined within this layer, such as walking, shooting, positioning, intercept etc. The skills layer is the basis of the whole decision- making layer, and it is the bridge between the low layer and decision layer. But whether it is the analysis of the message, or visual positioning will have a deviation, the impact on the decision-making, the need to use the relevant algorithm to reduce the impact of errors. The decision layer is equivalent to the human brain, which is responsible for coordinating the team strategy, according to the market situation and make different stations, pass, dribble etc.

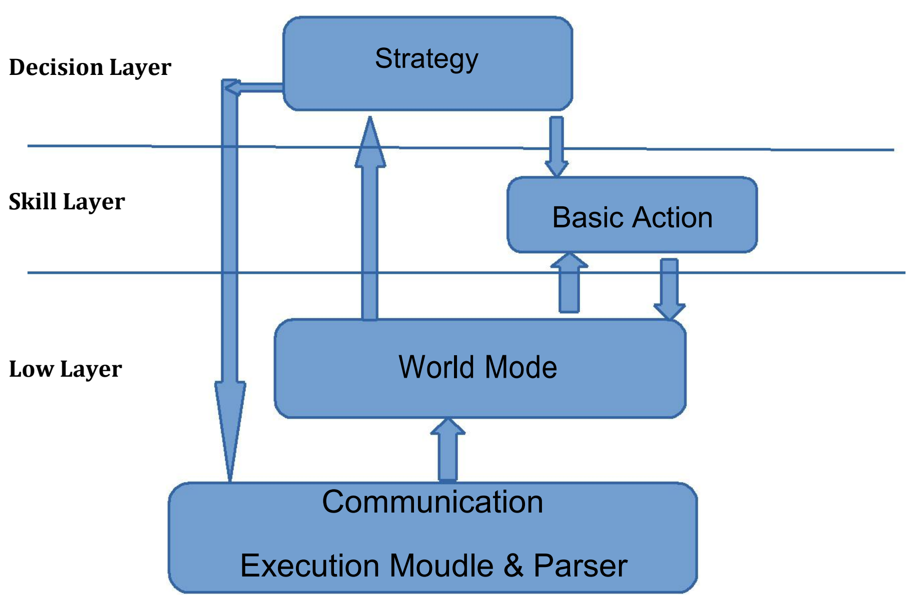


**Fig. 1.** Team Architecture of Miracle3D

## 4 Omnidirectional walking and Path planning

### 4.1 Omnidirectional walking

We use a multi-layer structure to control the robot's omnidirectional walking.The walking controller is divided into four layers according to the function:Path planning, Gait generation, Trajectory generation, Joint control.The path planner receives the target state and the robot's current state to plan the path.The gait generator draws the path according to the current state.The trajectory generator generates the joint angle trajectory.The joint angle controller calculates the angular velocity of the joint.

### 4.1 Path planning

First, determine the path without considering obstacles in path planning, then check if the re are obstacles in the path. If there is one, divide the path into two according to the bound aries of the obstacle and search again. Repeat this process until there is no obstacle in the chosen path.

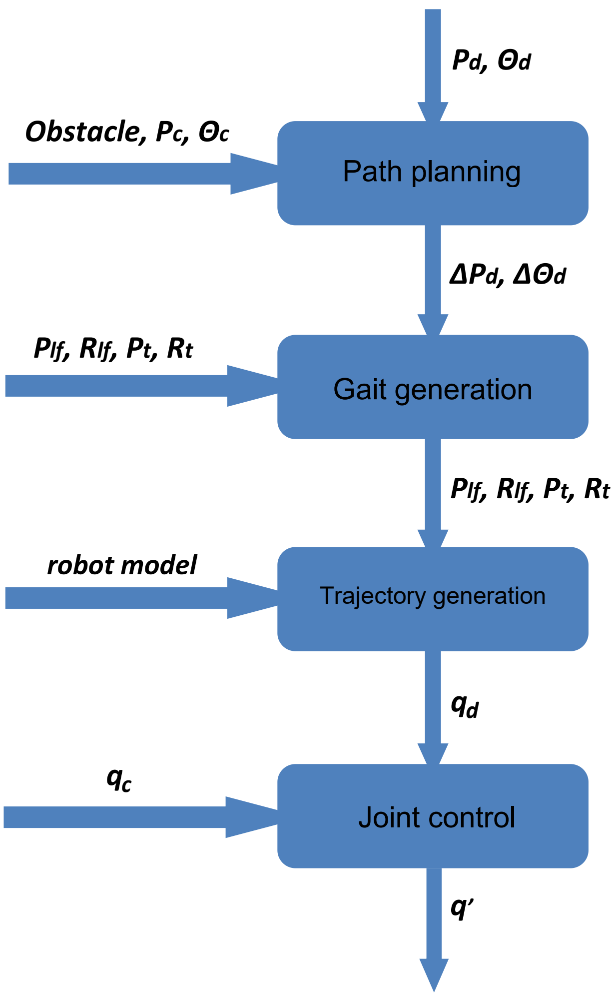


**Fig. 2.** Multip-layer structure of walking controller

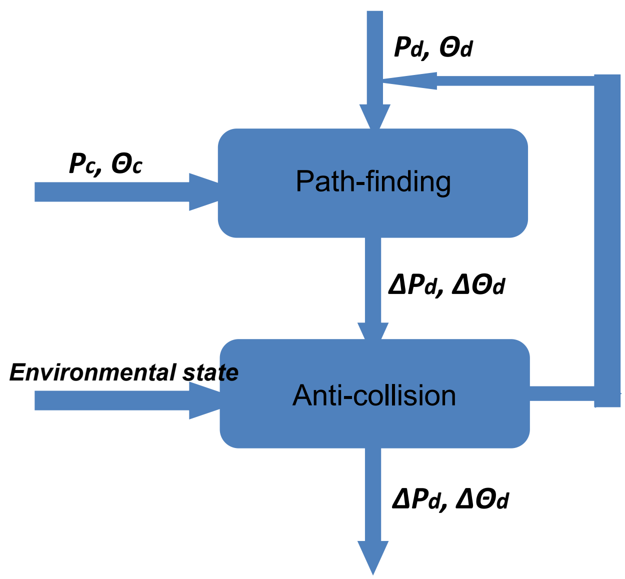


**Fig. 3.** Walking path planning

## 5 Future Works

Now many teams have their own ways to optimize the kick action. they can kick the ball far away, the action is completed in a short time. We are studying the action of robot kicking: selecting a kick point on the ball, planning the trajectory. We aim to implement dynamic role assignment, we also implement the kick trajectory based on bessel curves [5] [6] and optimize the kick action based on EMA-ES [6][7].

## Acknowledgements

Now, our team is based on base code UT Austin Villa released. Thank UT Austin Villa's teammates, they released stable base code. We acknowledge their effort, publications and their theories.

## References

1. Patrick MacAlpine , Peter Stone. UT Austin Villa RoboCup 3D Simulation Base Code Release.In: Proceedings of the RoboCup International Symposium 2016 (RoboCup 2016),Leipzig, Germany, July 2016.
2. MacAlpine, P., Barrett, S., Urieli, D., Vu, V., Stone, P.: 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-12) (July 2012)
3. MacAlpine, P., Urieli, D., Barrett, S., Kalyanakrishnan, S., Barrera, F., Lopez-Mobilia, A., Ştiurcă, N., Vu, V., Stone, P.: UT Austin Villa 2011: A champion agent in the RoboCup 3D soccer simulation competition. In: Proc. of 11th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2012) (June 2012)
4. Stoecker, J., Visser, U.: Roboviz: Programmable visualization for simulated soccer. In: Rfer, T., Mayer, N., Savage, J., Saranl, U. (eds.) RoboCup 2011: Robot Soc-cer World Cup XV, Lecture Notes in Computer Science, vol. 7416, pp. 282–293. Springer Berlin Heidelberg (2012)
5. http://www.tsplines.com/resources/class_notes/Bezier_curves.pdf
6. Juan Liu.: The Motion Planning and Cooperation Mechanism of Biped Robots in RoboCup3D Simulation Environment, 2014.
7. Mike Depinet, Patrick MacAlpine, and Peter Stone.: Keyframe Sampling, Optimization, and Behavior Integration: Towards Long-Distance Kicking in the RoboCup 3D Simulation League. In: Proceedings of the RoboCup International Symposium 2014 (RoboCup 2014), Joao Pessoa, Brazil, July 2014.