

Miracle3D: Team Description Paper for RoboCup 3D Soccer Simulation League

Lei Liu, Zeyu Song, Hongdao Xie, Pengxiang Li, Fan Wang

Hefei Normal University, China

Abstract. This paper describes the Miracle3D team and the developments we did. It also describes the changes to the architecture of the Miracle3D team's structure. In addition, it also includes the research about Reinforcement learning methods optimize parameters.

1 Introduction

Miracle3D is a simulation soccer 3D team established in 2012 and has attended several competitions. Miracle 3D simulation robot soccer team participated in the state competition for the first time in 2012. In 2013, Miracle3D won the champion of Anhui Robot World Cup. In the same year, it won the Robocup3D national first prize in China and reached the last eight in Robocup3D IranOpen2014. In 2014, Miracle3D won the bronze medal in Anhui Robocup3D, and in the same year, it won the fourth place in China.

There were some problems with our team code. These are the problems we are studying. We haven't yet found the good solution to the problems of precise positioning of the robot and the ball (particle filter for localization), Long-Distance Kicking, and the fastest walking. We continue to improve our code, but we have a lot of problems that can't be solved in a short time. So, we decided to refer to the base code published by UT Austin Villa. In order to speed up the development of the team and focus on the research of multiagent system. Now, we use the base code released by UT Austin Villa, see <https://github.com/LARG/utaustinvilla3d>, and add strategies to the code.

The rest of this paper is organized as follows. In the second section, the basic code of UT Austin Villa RoboCup 3D Simulation is outlined. The third part introduces our team structure. Section 4 covers gait parameter optimization. The future is in Section 5.

2 Overview Team

UT Austin Villa RoboCup 3D Simulation Base Code Release is highly modular, providing us with the flexibility to modify and develop.

This version includes the following features:

- * Omnidirectional walking engine based on double inverted pendulum model
- * A skill description language for specifying parameterized skills/behaviors
- * Getup (recovering after having fallen over) behaviors for all robot types

- * A couple basic skills for kicking one of which uses inverse kinematics
- * 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 by drop-in player challenge 4
- * An example behavior/task for optimizing a kick

3 Team Architecture

The low layer includes a communication module and a receiving and executing module. As the last layer of the layer structure, it is used for server communication. Its function includes two aspects: sending and receiving. As the receiver, the communication module needs to obtain information from the Server and send message to the outside world through the analytical model. As the transmitter, the robot feeds the decision back to the Server through the communication module, and then passes the parsed information to the world model again, and updates the world model according to the information from the communication.

The skills layer is also the basic action layer, where player define basic movements and skills, such as walking, shooting, positioning, intercept, etc. The skills layer is the foundation of the entire decision- making layer and the bridge between the low layer and decision layer. However, both the analysis of message and the visual positioning will have deviations, which will affect the decision-making. Relevant algorithms need to be used to reduce the impact of errors. Decision-makers are the brains of the people responsible for coordinating team strategies based on market conditions, making different stations, passing, dribbling and so on.

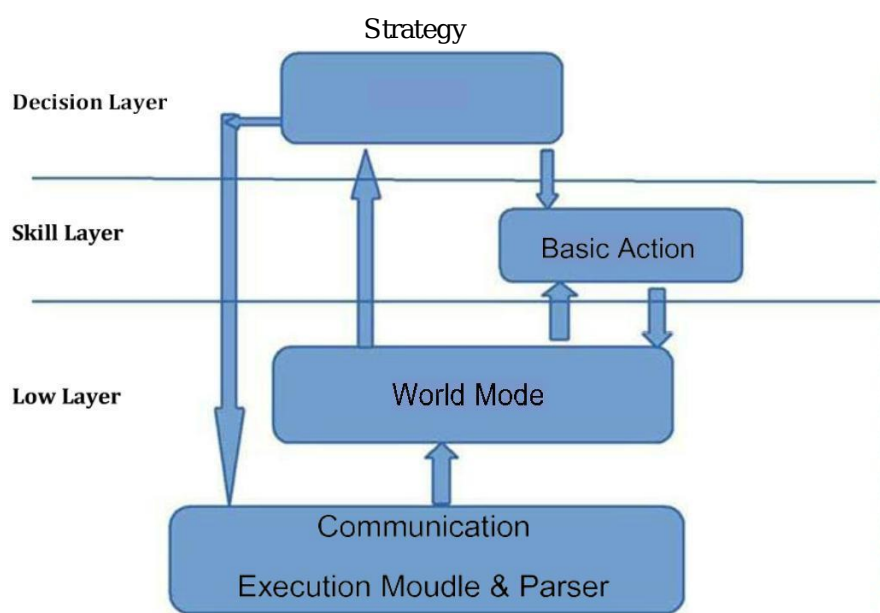


Fig. 1. Team Architecture of Miracle3D

4 Gait Parameter Optimization

The robot gait optimization problem is a multi-parameter function optimization problem, and naturally we thought of using genetic algorithm to optimize it.

4.1 Preference

We refer to some materials and select the following 16 parameters as optimization parameters in combination with the actual situation.

utwalk_max_step_size_angle

utwalk_max_step_size_x

utwalk_fraction_on_ground

utwalk_fraction_still

utwalk_fwd_offset

utwalk_fwd_offset_factor

utwalk_phase_length

utwalk_swing_ankle_offset

utwalk_max_step_size_y

utwalk_walk_height

utwalk_step_height

utwalk_fraction_in_air

utwalk_fraction_moving

utwalk_max_acceptable_com_error

utwalk_max_normal_com_error

utwalk_default_com_pos_x

4.2 Adaptive Genetic Algorithm

In the classical genetic algorithm, the mutation probability P_m and the crossover probability P_c are fixed values. By dynamically calculating p_m and p_c , it is ensured that individuals with high fitness have lower probability of crossover and mutation, while individuals with low fitness have higher probability of crossover and mutation. Generate a new individual. Use the following formula:

$$P_c = \begin{cases} P_{c1} - \frac{(P_{c1}-P_{c2})(f'-f_{avg})}{f_{max}-f_{avg}}, & f' \geq f_{avg} \\ P_{c1}, & f' < f_{avg} \end{cases} \quad (1)$$

$$P_m = \begin{cases} P_{m1} - \frac{(P_{m1}-P_{m2})(f_{max}-f)}{f_{max}-f_{avg}}, & f \geq f_{avg} \\ P_{m1}, & f' < f_{avg} \end{cases} \quad (2)$$

f_{max} ——The largest fitness value in the group

f_{avg} ——The average fitness value of the population in each generation

f' ——The larger fitness value of the two individuals to be crossed

f ——The fitness value of the individual to mutate

$P_{c1}=0.9$, $P_{c2}=0.6$, $P_{m1}=0.1$, $P_{m2}=0.00$

4.3 Simulated Annealing Genetic Algorithm

In order to solve the problem that the classical genetic algorithm falls into local optimal solution prematurely, simulated annealing algorithm is introduced. If it is larger before the mutation, accept the mutation with a certain probability.

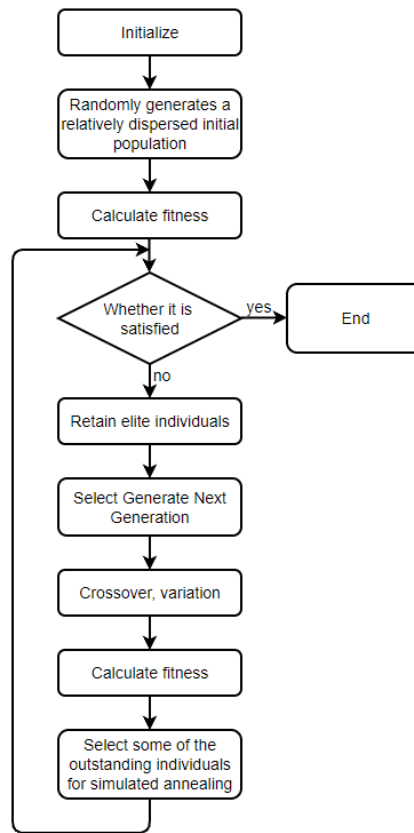


Fig. 2. Algorithm logic diagram

4.4 initialization

Select appropriate initial seed parameters, directly assign values to some individuals (15%), and apply the parameters perturbed by normal distribution to another part of individuals (85%). While ensuring feasible individuals, increase the diversity of initial individuals to prevent the algorithm premature convergence.

4.5 Selection, Crossover and Mutation Operators

For the selection of individuals, we need to ensure the positive correlation between the selection probability and the fitness, so we choose the traditional roulette selection operator. Based on the adaptive genetic algorithm, on the premise of whether to cross to generate new individuals with p_c probability, let each gene locus has a 30%~50% crossover probability. In practice, we use 45% as a compromise. The mutation operator is the simplest, and each locus has a probability mutation of p_m .

4.6 individual evaluation

We need to optimize the moving speed of the robot, and naturally select how many distances the robot has completed from A to B in a certain period of time as the fitness F :

- 1) Read the selected individual parameters and write them into the agent parameter file;
 - 2) Use the agent to measure the fitness five times, remove the lowest value, and take the average as the final fitness;
 - 3) Repeat the operation to traverse all individuals in the population among them;
- the fitness is measured 5 times, and the lowest value is removed in order to reduce the impact of the robot's wrestling and obtain more aggressive individuals.

4.7 result

We optimized about 80 individuals and 80 generations for the type 0 robot, and the final results are as follows:

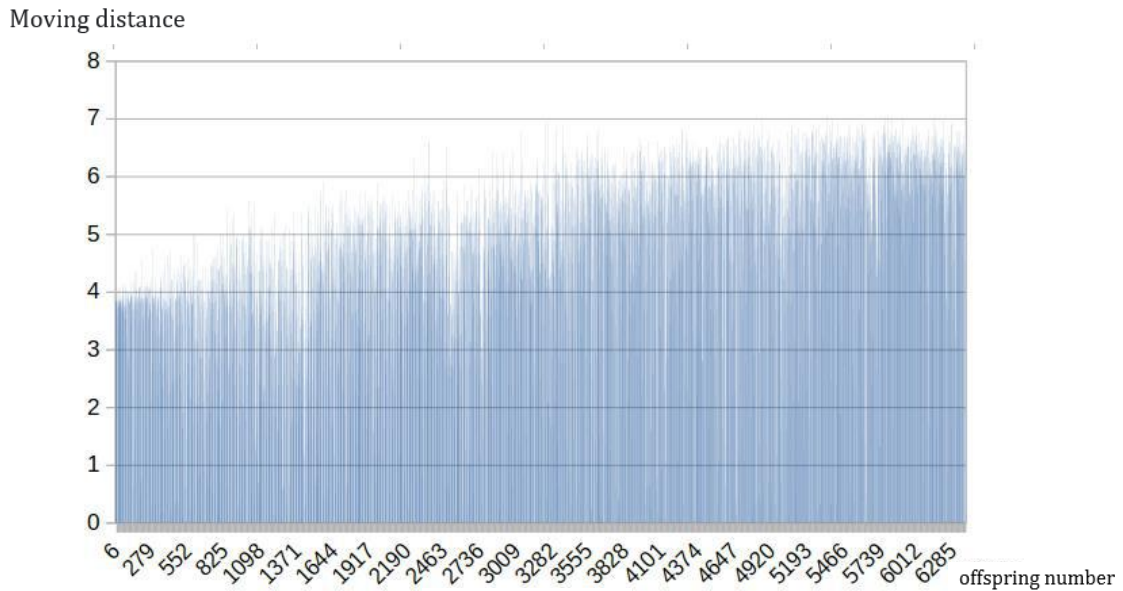


Fig. 3. Parameter optimization diagram

The moving distance of the robot is significantly improved, which proves that the optimization is effective.

5 Future Works

In order to solve the formation and Gait Parameter Optimization of football teams, it is necessary to study the Application of Gait Parameters in its different environment, and to establish a systematic robocup3d robot's gait optimization mechanism. Based on reinforcement learning, the related algorithms are used to perform the agent's kicking and walking parameters training optimizations.

6 Acknowledgements

Now, our team is based on basic code published by UT Austin Villa. Thanks to the teammates of UT Austin Villa's teammates, they released stable base code. We thank them for their effort, publications and 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., Știurcă, N., Vu, V., Stone, P.: UT Austin Villa 2011: A champion robot in the RoboCup 3D soccer simulation competition. In: Proc. of 11th Int. Conf. on Autonomous robots and Multiagent Systems (AAMAS 2012) (June 2012)

4 O'Rourke J . Computational Geometry in C [M] . Beijing: Mechanics 5. Industry Press , 2005: 161 – 165 .

5 Bowyer A . Computing Dirichlet tessellations [J] . The Computer Journal, 1981, 24(2) : 162 – 166 .

6 Amenta N, Bern M, Kamvysselis M . A new Voronoi – based surface reconstruction algorithm [C] // Proceeding of SIGGRAPH'98 . Danvers: Assison – Wssley Publishing Company, 1992: 415 – 421 .

7 Edelsbrunner H, Shah N R . Incremental topological flipping works for regular triangulations [J] . Algorithmica, 1996, 15(3) : 223 – 241 .

8 Rongyi He, Chunguang Li. RoboCup3D simulation robot gait optimization research [J]. Computer and Modernization, 2018 (3).