Apollo3D Team Description Paper

Qian Zhao, Keji He, Tongxin Cui, Zhiying,Ke, Yulei Lu,
Hao He, Chenxi Xu, Zhiwei Liang

College of Automation, Nanjing University of Posts and Telecommunications
qianzhaoapollo@163.com

Abstract

Apollo3D is a team in RoboCup soccer simulation 3D league. We mainly aim at building a systematical architecture of intelligent and skillful robots. In the newest 11 vs 11 version, due to the introduction of sensor noise and the expansion of the soccer field, a more accurate positioning and efficient upper strategy are need in order to avoiding robot being in a disorder. In the past year, our team Apollo3D successful devised a new localization system and a new set of cooperating tactics of the agents. In this paper, we introduce the mechanism of localization, dynamic footstep planning, omnidirectional walking skill and decision making system.

1 Introduction

Apollo Simulation 3D Team was established in 2006, and successfully attended several competitions. We have won the first place in Robocup 2013 and the second place in 2013 Iran Robocup recently. The simulated Nao is much like the real one that attracts a large mount of students to devote to this field. Thanks to the devotion and cooperation of these students, several achievements had been achieved in the past years.

With the developing and improving of the RoboCup3D platform, the number of players has increased to 11, the field has expanded to 600 square meters, and we all must use heterogeneous players. These changes urge us to reconsider the action of each robot, the localization and communication problem. On this basis, in order to enhance the overall performance, we redesign the decision making system of our robots. Section 2 will introduce the self-localization of particle filter and how to use Kalman filter to track the ball and other agents. Section 3 will introduce a footstep planner for biped robots using the method of sequence approximation. Section 4 will introduce the walking skill of Apollo3D. Section 5 will discuss the hierarchical role assignment and multi-agent cooperation system.

2 Localization

2.1 Particle Filter Self-localization

Humanoid robot self-localization means estimating the positions and orientations of the local coordinates $\sum_v$ relative to the world frame $\sum_w$ (Fig.1. This problem involves at least 6 configuration parameters (x, y, z, R, P, Y), and it is hard to build their correlations with the motion model using limited odometers and sensors. Meanwhile, most of the time that the robot actually needs to localize itself are when it walks upright on a flat surface and the hip joints are restricted in a horizontal plane. Thus the z, R, P (height, roll and pitch) of $\sum_v$ are bounded in a small range. So the robot only needs to predict the 2D position(x, y) and the heading direction θ.

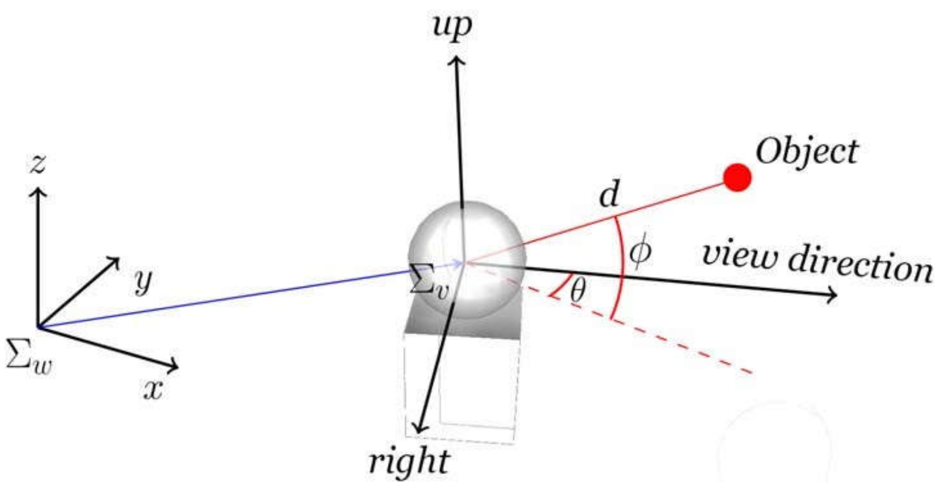

Figure 1: Diagram of the robot vision system

Particle filters estimate the posterior distribution of the state x_t of the dynamical system conditioned on the sensor measurement z_t and control information u_{t-1}, $Bel(x_t) \propto p(x_t \mid z_t,\, u_{t-1})$. This posterior can be computed recursively using Bayes rules and partially observable controllable Markov chains:

$$\overline{Bel}(x_t) = p(x_t \mid x_{t-1}, u_{t-1}) Bel(x_{t-1}) \tag{1}$$

$$p(x_t \mid z_t, u_{t-1}) = \mu\, p(z_t \mid x_t) \overline{Bel}(x_t) \tag{2}$$

Equation (1) is called motion update phase. where the robot needs to predict the new state of position and orientation x_t basing on its motion u_{t-1} according to its odometers and the last state $Bel(x_{t-1})$. Equation (2) is the observation update phase. In this phase, the robots update to the current state on condition of the measurement of the sensors z_t.

The key idea of the particle filter is to represent the posterior $p(x_t \mid z_t, u_{t-1})$ by a set of weighted state samples:

$$S_t = \{< x_t^{(i)}, w_t^{(i)} >\}_{i=1,\cdots n} \quad (3)$$

where each $x_t^{(i)}$ stands for an instance of estimated state with $w_t^{(i)}$ being its weight. Theoretically, as $N \to \infty$ the distribution of these samples match the density of the posterior. In practice, we use 1000 particles to approximate the posterior. Algorithm 1 shows the details.

Algorithm 1 Partile_filter(S_{t-1}, u_{t-1}, z_t):

$S_t := \emptyset$, $N = 1000$, $w_{total} = 0$
for $i := 1$ to N **do**
  draw index $j(i)$ with probability $\propto w_{t-1}^{(j(i))}$ in S_t
  $x_t^{(i)} :=$ **motion_model**$(u_{t-1}, x_{t-1}^{(j(i))})$
  $w_t^{(i)} := p(z_t | x_t^{(i)})$
  $w_{total} := w_{total} + w_t^{(i)}$
  $S_t := S_t \cup \{\langle x_t^{(i)}, w_t^{(i)} \rangle\}$
end for
for $i := 1$ to N **do**
  $w_t^{(i)} := w_t^{(i)} / w_{total}$
end for
return S_t

Finally, the algorithm returns S_t, we simply calculate the average of $x_t^{(i)}$ to stimate the state at time t.

2.2 Kalman Filter Tracking

In RoboCup3D environment, the position of the ball and agents keep changing all the time. If each individual agent can accurately predict other agents' movement, it will better seize the initiative. Especially at the risk of opponents shooting, our goalie's quick reponse to stop the ball largely depend on its prediction of the velocity of the ball. The Kalman filter not only can increase the accuracy of tracking other objects, but also can help predict their other states like velocity.

3 Dynamic Footstep Planner

The walking parameters of robots are defined as y (the forward direction), x (the lateral direction) and θ (the turning degree) at every time. In competition, the robots want to reach the goal as possible as fast. In order to handle the problem, the robots need to adjust the three above waking parameters in dynamic environment. A sequence theorem is employed to control the three parameters in our team codes. In a local coordinate system centered at the robot, the goal can be defined as $state^t_0\left(x^t_0, y^t_0, \theta^t_0\right)$. Once each walking step $walk^w_i\left(x^w_i, y^w_i, \theta^w_i\right)$ of robots is performed, the goal state is adjusted from $state^t_i(x^t_i, y^t_i, \theta^t_i)_i$ to $state^t_i\left(x^t_{i-1} - x^w_x, y^t_{i-1} - y^w_x, \theta^t_{i-1} - \theta^w_i\right)$, shown in Fig.2.

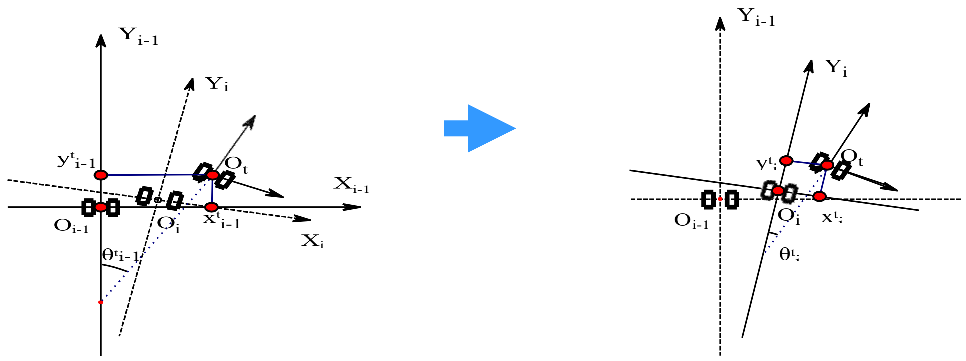

Fig.2. The change of goal state during the walking of robots

Let $x^w/x^t_{i-1} = y^w/y^t_{i-1} = \theta^w_i/\theta^t_{i-1} = k_i, k_i > 0$. As a result, the x^t_i, y^t_i, θ^t_i can arrive at zero at the same time. In other words, the robots arrive at the goal. Let $k_i = v_i \Big/ \sqrt{x^t_{i-1}{}^2 + y^t_{i-1}{}^2 + \theta^t_{i-1}{}^2}$, the following equation can be attained:

$$\begin{cases} x^w_i = v_i \bullet x^t_{i-1} \Big/ \sqrt{x^t_{i-1}{}^2 + y^t_{i-1}{}^2 + \theta^t_{i-1}{}^2} \\ y^w_i = v_i \bullet y^t_{i-1} \Big/ \sqrt{x^t_{i-1}{}^2 + y^t_{i-1}{}^2 + \theta^t_{i-1}{}^2} \\ \theta^w_i = v_i \bullet \theta^t_{i-1} \Big/ \sqrt{x^t_{i-1}{}^2 + y^t_{i-1}{}^2 + \theta^t_{i-1}{}^2} \end{cases} \tag{4}$$

where v_i is an important parameter which can control the walking speed of robots.

Due to the dynamic constraints, we need to assure $x^w_i \le x^w_{\max}$ $y^w_i \le y^w_{\max}$ $\theta^w_i \le \theta^w_{\max}$, where $x^w_{\max}$ is the maximum speed of robot lateral movement, $y^w_{\max}$ is the maximum forward speed, and $\theta^w_{\max}$ is the maximum turning speed at every step. In order to improve the walking speed as far as possible, let:

$$\begin{cases} k_{x_i} = x^t_{i-1} \Big/ \sqrt{x^t_{i-1}{}^2 + y^t_{i-1}{}^2 + \theta^t_{i-1}{}^2} \\ k_{y_i} = y^t_{i-1} \Big/ \sqrt{x^t_{i-1}{}^2 + y^t_{i-1}{}^2 + \theta^t_{i-1}{}^2} \\ k_{\theta_i} = \theta^t_{i-1} \Big/ \sqrt{x^t_{i-1}{}^2 + y^t_{i-1}{}^2 + \theta^t_{i-1}{}^2} \end{cases} \tag{5}$$

According to the (4) and (5), we can attain $\begin{cases} x^w_i = k_{x_i} \bullet v_i \\ y^w_i = k_{y_i} \bullet v_i \\ \theta^w_i = k_{\theta_i} \bullet v_i \end{cases}$. According to $x^w_i \le x^w_{\max}$ $y^w_i \le y^w_{\max}$ $\theta^w_i \le \theta^w_{\max}$ and (5), the following conditions can be computed:

$$\begin{cases} v_i \le x^w{}_{\max} / k_{x_i} \\ v_i \le y^w{}_{\max} / k_{y_i} \\ v_i \le \theta^w{}_{\max} / k_{\theta_i} \end{cases}.$$

As a result, if $v_i = \min(x^w{}_{\max} / k_{x_i}, \min(y^w{}_{\max} / k_{y_i}, \theta^w{}_{\max} / k_{\theta_i}))$, the robots can walk at the most speed in the dynamic constraints.

4 Omnidirectional Walking Skill

This section mainly describes the omnidirectional walking motion design of our team Apollo3D. In this section, we employed a model which is based on double linear inverted pendulum to predict and control the robot walking motion. And then we used machine learning algorithm to optimize walking parameters. Ultimately, we realized the rapidly and stably omnidirectional walking of biped robots in complex and dynamic environment.

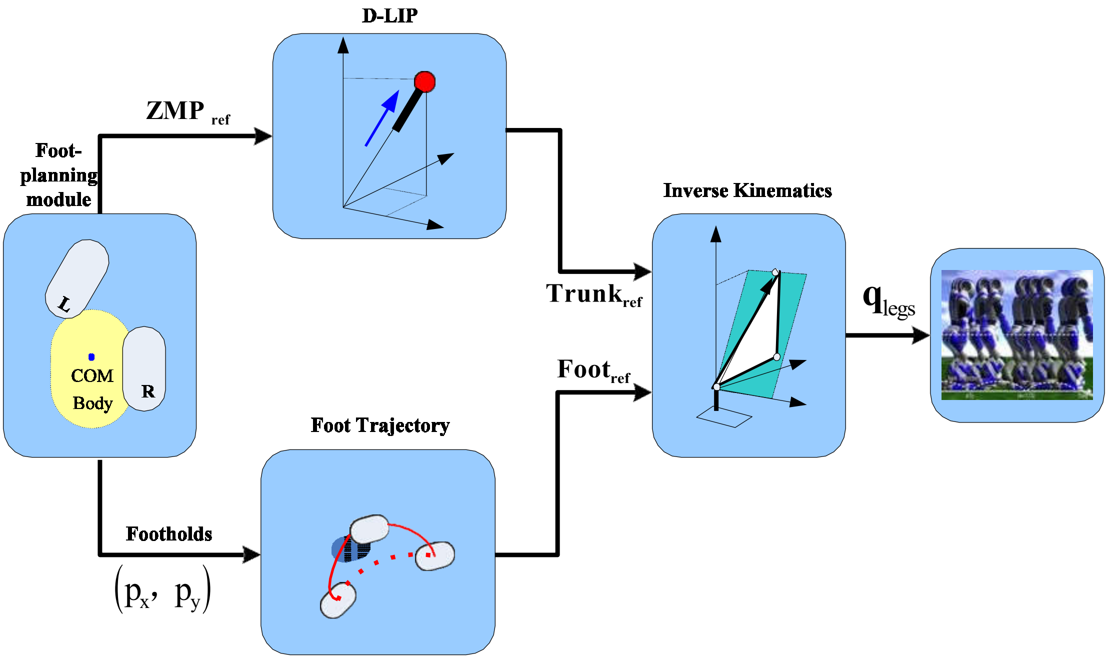

Figure3. Omnidirectional walking diagram

In the process of competition, the changing external environment requires the robot to alter its orientation at any time, turn agilely and forward fast. The walking method employed in this paper is presented in the Fig.3. First we can get the feasible footholds and compute the ZMP values based on the foot-planning module. Subsequently, the trunk trajectory of robot can be attained based on a linear inverted pendulum model (LIPM) with a predictive control method. As a result, we can plan the space trajectory of every two footholds in 3D space according to the cubic spline interpolation method. Meanwhile, each joint angle can be calculated according to inverse kinematics method. The pose of the robot's trunk can easily computed by the gyro sensor of NAO. Last but not least, we use machine learning algorithm to optimize the walking parameters.

5 Hierarchical Decision Making

The team size of RoboCup 3D growth from 6 in 2010 to 9 in 2011, and finally 11 last year, which raised the concern of better multi-agent corporation. Increased the number of players though, the robot who is on the ball is unique for any single moment. So far, distant passing skills between robots are still impractical for most teams, how the dribbler control the ball becomes the key to win a game.

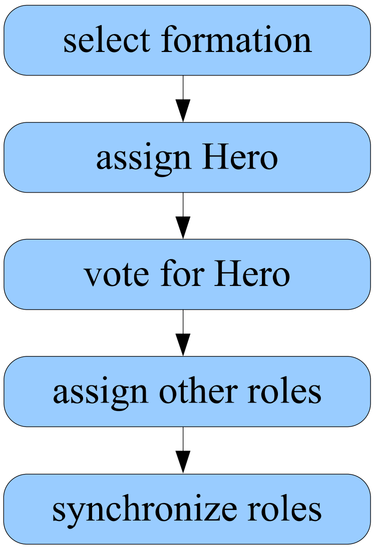

Figure 4.Flow chart of assigning roles

The dribbler, called the Hero role in our model, bears the heaviest burden in a competition. In the tactic of Apollo3D, agents first select a formation according to the position of the ball, then choose a Hero. Since the vision of agents is restricted, and it has errors that every agent's perception of self and other players' locations, multiple Heroes may appear at the same time, causing chaotic collisions around the ball, which brings negative effects on controlling the ball. To solve this issue, we employed a voting method to select the best Hero, using the communication system to synchronize each agent's selection. Since it has time delays in the communication system, and agents cannot 100% sure about its selection, we gives each vote a weight describe by a probability value between $(0,1)$.

When the Hero is dribbling, we make sure that every other player stay on a ascendant position to assist attacking. Each role (or position) is assigned according to the robot's location in the current formation. Meanwhile, we also have to synchronize other roles among agents, to prevent potential risk of collision and keep the order of attacking.

Here we use the following criteria for choosing the Hero:

- Whether the player is fall down.
- Whether the ball is visible to the player.
- Player's distance to ball.
- Whether the player is in front of the ball or behind it (players in front of the ball often need extra time for turning).
- Whether the player is goalie (competition rule stipulated the goalie has to be NO.1).
- Whether this player is Hero in last cycle.

6 Conclusion

In this paper, we discussed algorithms in biped robot localization, walking and multi-agent corporation. We proceed a large amount of experiments, and the results validated the reliability and superiority of these algorithms. Research on humanoid robot has gained popularity in Robotics, many researchers and engineers focus their research on this field. Our further work will focus on studying the strategy of the multi-agent cooperation and confrontation.

References

[1] Christopher G. Atkeson, Andrew W. Moore, and Stefan Schaal. Locally weighted learning. Artif. Intell. Rev., 11(1-5):11-73, 1997.

[2] Dieter Fox. Kld-sampling: Adaptive particle filters. In NIPS, pages 713-720, 2001.

[3] A.; Ro 04fer T.; Graf, C.; Ha 04rtl and T. Laue. A robust closed-loop gait for the standard platform league humanoid. Proceedings of the 4th Workshop on Humanoid Soccer Robots A workshop of the 2009 IEEE-RAS Intl. Conf. On Humanoid Robots (Humanoids 2009), 2009.

[4] E. Hashemi, M.G. Jadid, M. Lashgarian, M. Yaghobi, and R.N.M. Shafiei. Particle filter based localization of the nao biped robots. In System Theory (SSST), 2012 44th Southeastern Symposium on, pages 168 -173, march 2012.

[5] Qiang Huang, Kazuhito Yokoi, Shuuji Kajita, Kenji Kaneko, Hirohiko Arai, Noriho Koyachi, and Kazuo Tanie. Planning walking patterns for a biped robot. IEEE Transactions on Robotics, 17(3): 280-289, 2001.

[6] P. Maybeck. The Kalman filter: An introduction to concepts. Au-tonomous Robots, 1990.

[7] Andrew Y. Ng, H. Jin Kim, Michael I. Jordan, and Shankar Sastry. Autonomous helicopter flight via reinforcement learning. In NIPS, 2003.

[8] Simon Thompson, Satoshi Kagami, and Koichi Nishiwaki. Localisation for autonomous humanoid navigation. In Humanoids, pages 13-19, 2006.

[9] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. Probabilistic robotics. MIT Press, Cambridge, Mass., 2005.

[10]Xinxin Wang, Xuesong Yan, Yongshan Zhang, and Siyu Pan. Kalman filter in the robocup 3d positioning. In Computer Science and Electronics Engineering (ICCSEE), 2012 International Conference on, volume 3, pages 47-52, march 2012.

[11]P. MacAlpine, D. Urieli, S. Barrett, S. Kalyanakrishnan, F. Barrera, A. Lopez-Mobilia, N. S 05tiurc 02a, V. Vu, and P. Stone. UT Austin Villa 2011 3D Simulation Team report. Technical Report AI11-10, The Univ. of Texas at Austin, Dept. of Computer Science, AI Laboratory, December 2011.