Presentation of Apollo3D

LIANG Zhiwei, FENG Huanhuan, ZHANG Zhihua, CUI Xuanyu, DENG Yangyang, ZHOU Zixuan
Nanjing University of Posts and Telecommunications, Nanjing, 210046

1 Team 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 amount 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.

2 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 $state^t{}_i(x^t{}_i, y^t{}_i, \theta^t{}_i)$ to $state^t{}_i\left(x^t{}_{i-1} - x^w{}_i, y^t{}_{i-1} - y^w{}_i, \theta^t{}_{i-1} - \theta^w{}_i\right)$, shown in Fig.1.

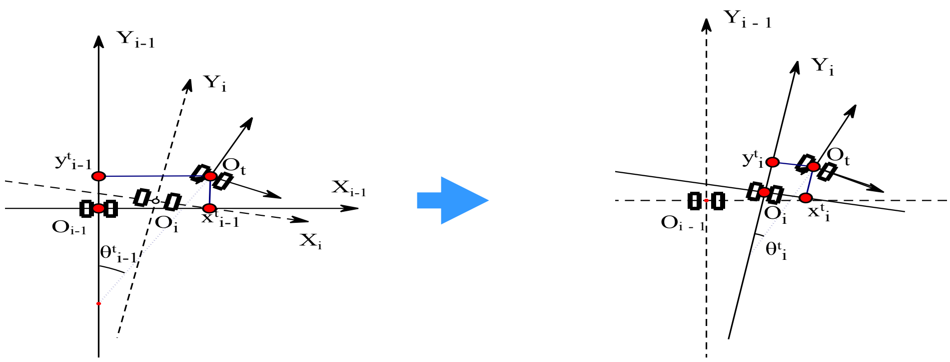

Fig.1. The change of goal state during the walking of robots

Let $x^w{}_i / x^t{}_{i-1} = y^w{}_i / y^t{}_{i-1} = \theta^w{}_i / \theta^t{}_{i-1} = k_i$, $k_i > 0$. As a result, the $x^t{}_i, y^t{}_i, \theta^t{}_i$ 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:

$$\begin{cases} x^w{}_i = v_i \bullet x^t{}_{i-1} \Big/ \sqrt{x^t{}_{i-1}^{2} + y^t{}_{i-1}^{2} + \theta^t{}_{i-1}^{2}} \\ y^w{}_i = v_i \bullet y^t{}_{i-1} \Big/ \sqrt{x^t{}_{i-1}^{2} + y^t{}_{i-1}^{2} + \theta^t{}_{i-1}^{2}} \\ \theta^w{}_i = v_i \bullet \theta^t{}_{i-1} \Big/ \sqrt{x^t{}_{i-1}^{2} + y^t{}_{i-1}^{2} + \theta^t{}_{i-1}^{2}} \end{cases} \tag{1}$$

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:

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

According to the (1) and (2), 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 (2), 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.

3 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.2. 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.

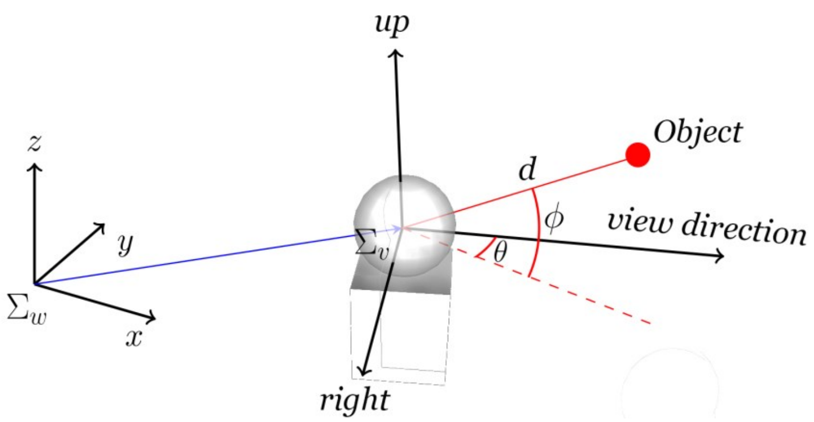

Figure 2: Diagram of the robot vision system

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:

$$\overline{Bel}(x_t) = p(x_t \mid x_{t-1}, u_{t-1}) Bel(x_{t-1}) \tag{3}$$

$$p(x_t \mid z_t, u_{t-1}) = \mu p(z_t \mid x_t) \overline{Bel}(x_t) \tag{4}$$

Equation (3) 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 (4) 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:

$$S_t = \{< x_t^{(i)}, w_t^{(i)} >\}_{i=1,\cdots} \tag{5}$$

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.

4 Hierarchical Decision Making

The team size of RoboCup 3D growth from 6 in 2010 to 9 in 2011, and finally 11 2013, 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.

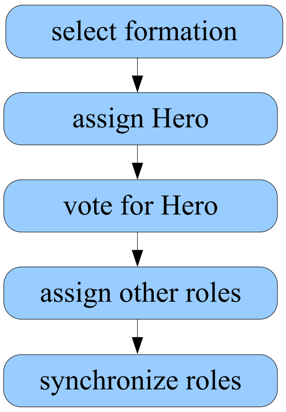

Figure 3.Flow chart of assigning roles

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.

5 Conclusion

In this presentation, 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.