Comparison between Position and Posture Recovery in Path Following

Sunglok Choi, JaeYeong Lee, and Wonpil Yu
Robot Research Department, ETRI, Daejeon, Republic of Korea
{sunglok, jylee, ywp}@etri.re.kr

Abstract—Path following is one of the most basic function of a mobile robot. It is a series of trials to attain its state to the target state. Its objective state determines the type of path followers: position, pose, and posture recovery. This paper compares performance of two path followers via experiments. One of them makes the robot proceed to position on the path, which is position recovery. The other tries to follow position incorporated with orientation and velocity, which is posture recovery. Comparison presents that posture recovery makes the robot more shaking when a location sensor has noise.

Keywords—Path Following, Path Tracking, Position Recovery, Pose Recovery, Posture Recovery, Pure Pursuit

1. Introduction

Path following and tracking are one of the oldest topics in robotics and control theory because they are essential to perform more complex tasks such as guidance, serving, exploration, and so on. Many researchers does not distinguish path following and tracking, but they have slightly different definition. The path following is the process to control the robot to pursue the given path. The path, \( P \), is usually represented as a series of position on the given space. The problem is usually formalized as follows:

\[
\begin{bmatrix}
  v_c \\
  w_c 
\end{bmatrix} = \text{Follow}\left( \begin{bmatrix}
  x_r \\
  y_r \\
  \theta_r 
\end{bmatrix}, \begin{bmatrix}
  v_t \\
  w_t 
\end{bmatrix}, P, \ldots \right),
\]

where \( x_r, y_r, \) and \( \theta_r \) is current position and orientation of the robot, and \( v_t, w_t \) is its linear and angular velocity. Two outputs, \( v_c \) and \( w_c \), are the desired linear and angular velocity to track the path. The robot usually operates on limited velocity and acceleration, whose bounds are noted as \( v_{\text{max}}, w_{\text{max}}, a_{\text{max}}, \) and \( \alpha_{\text{max}} \), respectively. In contrast, the path tracking is the process to control the robot to attain the desired state such as position and velocity, so it is called as reference tracking. Its problem definition is

\[
\begin{bmatrix}
  v_c \\
  w_c 
\end{bmatrix} = \text{Track}\left( \begin{bmatrix}
  x_r \\
  y_r \\
  \theta_r 
\end{bmatrix}, \begin{bmatrix}
  v_t \\
  w_t \\
  \theta_t 
\end{bmatrix}, \begin{bmatrix}
  v_d(t) \\
  w_d(t) \\
  \theta_d(t) 
\end{bmatrix}, \ldots \right),
\]

where \( x_d(t), y_d(t), \theta_d(t) \), and \( w_d(t) \) are the desired position, orientation, and velocity at time \( t \). The robot needs to calculate the desired position and velocity profile with respect to the given path. The path following is regarded as less restrictive, whose theoretical comparison with tracking was dealt in [1].

The path following can be classified with its objective state on the given path (Figure 1). The position recovery moves the robot on the path. It does not consider orientation and velocity of the robot, but it just tries to minimize distance from the path. In addition to position, the pose recovery tries to align the robot toward direction of the path. The posture recovery considers more about the path such as suitable velocity and acceleration. The path following is also categorized with its employed physics: kinematics and dynamics.

This paper compares two different kinematic path followers. The first method, pure pursuit [3], aims position recovery. The other adopts concept of landing when the robot moves toward the path. It will be noted as landing path tracker [4], which is posture recovery. They are briefly explained in Section 2. Their performance is compared in experiments using two different trajectories: linear and circular. The experiments and its discussion are described in Section 3.

2. Two Path Following Methods

2.1 Pure Pursuit (Position Recovery)

The pure pursuit [3] consists of two steps: target selection and velocity control. The first step chooses the target position to pursue on the given path. The target position is usually selected as follows:

\[
\begin{bmatrix}
  x_t \\
  y_t 
\end{bmatrix} = P\left( \arg\min_i \left( e(\|y_i\|, \bar{P}(i)) \right) + \delta_{\text{ahead}} \right),
\]

where \( e \) calculates Euclidean distance between two points, and \( \delta_{\text{ahead}} \) is the lookahead parameter. The lookahead parameter advances the target point to the goal point. The bigger lookahead makes the robot pursuit the path more smoothly.

The second step calculates necessary velocity to follow the target point \(^1\). The radial and angular error are calculated as

\[
\begin{bmatrix}
  v_t \\
  w_t \\
  \theta_t 
\end{bmatrix} = \text{Velocity}\left( \begin{bmatrix}
  x_t \\
  y_t \\
  \theta_t 
\end{bmatrix}, \begin{bmatrix}
  x_d(t) \\
  y_d(t) \\
  \theta_d(t) 
\end{bmatrix} \right).
\]

\(^1\)The pure pursuit in this paper is described in more easier form than the original [3].
In the degenerate case ($|\Delta \theta| > \pi/2$), the velocity is calculated as follows:

$$v_c = v_{\text{max}} \quad \text{and} \quad w_c = \frac{2v_c \sin \Delta \theta}{D}.$$  

In the degenerate case ($|\Delta \theta| > \pi/2$), the velocity is calculated as follows:

$$v_c = 0 \quad \text{and} \quad w_c = w_{\text{max}} \sin \Delta \theta,$$

where sign returns sign of the given value as like $-1$, $0$, and $+1$. The velocity makes the robot rotate until $|\Delta \theta| \leq \pi/2$. If the calculated angular velocity is bigger than its maximum ($w_c > w_{\text{max}}$), the velocity needs to be adjusted again as follows:

$$w_c = w_{\text{max}} \sin \Delta \theta \quad \text{and} \quad v_c = \frac{Dw_c}{2 \sin \Delta \theta}.$$

### 2.2 Landing Path Tracker (Posture Recovery)

The landing path tracker [4] originally solves the tracking problem, not the following problem. However, adopting the first step of the pure pursuit, it can be applied to path following. The target position is assigned as like Equation 3. The target orientation and velocity is selected as follows:

$$\theta_t = \Delta \theta, \quad v_t = v_{\text{max}} \quad \text{and} \quad w_t = 0.$$  

The landing path tracker calculates necessary velocity using a landing curve 2. It formulates error between the current position and target position as follows:

$$e_x = + (x_t - x_r) \cos \theta_t + (y_t - y_r) \sin \theta_t,$$

$$e_y = -(x_t - x_r) \sin \theta_t + (y_t - y_r) \cos \theta_t,$$

$$e_\theta = \theta_t - \theta_r,$$

where $e_x$, $e_y$, and $e_\theta$ are tangential, lateral, and angular error, respectively. They are described in Figure 2(b). The landing curve is selected as

$$L(x_L) = cx_r^2 \sin (e_y).$$  

The curve satisfies three initial values as follows:

$$L(x_L) |_{x_L=0} = 0, \quad \frac{dL(x_L)}{dx_L} |_{x_L=0} = 0, \quad \frac{d^2L(x_L)}{dx_L^2} |_{x_L=0} = 0,$$

which means that the robot will have the target position, orientation, and velocity at $x_L = 0$. From the landing curve (Equation 10), the tangential angle and its differential are derived as

$$\theta_L = \theta_t + \tan^{-1} \left( \frac{3c_x e_L^2 \text{sign}(e_y)}{\frac{6c_x e_L}{1 + \tan^2(\theta_t - \theta_r)} \dot{e}_y \text{sign}(e_y)} \right),$$

where $e_L = (|e_y|/c_x)^{1/3} \text{sign}(e_y)$ from Figure 2(b), and $\dot{e}_y$ is $-w_t e_x + v_t \sin \theta_t$ from Equation 9. The necessary angular velocity is derived using bang-bang control as follows:

$$\dot{e}_c = \dot{\theta}_L - w_r + \left[ 2a_{\text{max}} |\theta_L - \theta_t| \right]^{1/2} \text{sign}(\theta_L - \theta_t),$$

which makes the robot have the tangential angle $\theta_L$ and its corresponding angular velocity $\dot{e}_c$ at $x_L = 0$. Similarly, the linear velocity is also calculated as follows:

$$v_c = \dot{e}_x + \left[ 2a_{\text{max}} |e_x| \right]^{1/2} \text{sign}(e_x),$$

where $\dot{e}_x$ is $v_t - v_x \cos \theta_t + w_t e_x$ from Equation 9. It tries bang-bang control to minimize tangential error $e_x$.

### 3. Experiments

Experiments were performed on two kinds of trajectories as like Figure 3(a) and 4(a). The straight line path was generated from $[0, 0]^T$ to $[10, 0]^T$, and the circular path was generated from $[0, 0]^T$ to $[-0.5, 3.4]^T$, whose diameter is about 3.5 meters. The given robot can accelerate $a_{\text{max}} = 0.5 \text{ m/s}^2$ and $a_{\text{max}} = 50 \text{ deg/s}^2$. Its maximum velocity is $v_{\text{max}} = 0.5 \text{ m/s}$ and $v_{\text{max}} = 50 \text{ deg/s}$. The robot performs following algorithm every 0.02 seconds, that is, 50Hz. The position data was measured with unbiased Gaussian noise, whose standard deviation is 0.05 meters. The orientation also has unbiased Gaussian noise, whose standard deviation is 5 degrees. The robot was located at $[-0.5, -0.5, 0]^T$. The lookahead parameter $\delta_{\text{ahead}}$ was adjusted as 100 index, which was approximately 1 meter. The landing parameter $c_r$ was 0.02.

Figure 3 and 4 presents trajectory and velocity of the robot when it pursued the given paths. In the line path, the landing tracker (about 1m) approached the path earlier than the pure pursuit (about 2m). However, the landing tracker was shaking around $\pm 0.5 \text{ m}$, which is also observed in angular velocity (Figure 3(c)). In the circular path, the result was similar. The landing tracker reached the path earlier, but it had bigger lateral error.

### 4. Discussion

This paper describes two kinematic path followers: the pure pursuit and landing path tracker. The landing path tracker is expected to have better accuracy than the pure pursuit because it adopts posture recovery. However, the experimental results presented the landing path tracker was worse than the pure pursuit. Figure 5 shows the lateral error.
Fig. 3. Experimental Result in the Straight Line Trajectory

Fig. 4. Experimental Result in the Circular Trajectory

Fig. 5. The Lateral Error and Its Landing ($c_x = 0.02, \dot{c}_y = 1.0$)

$e_y$ and its corresponding tangential angle $\theta_L$ (Blue) and angular velocity $\dot{\theta}_L$ (Red). In the figure, the angular velocity around 0 is varied significantly, which causes shaking angular velocity $\dot{\omega}_c$ in case of small orientation error. The landing path tracker needs to be investigated more to solve this problem.

References