Multi-level Deceleration Scheme for Accurate Goal Arrival

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

Abstract—When we stop our car in front of the stop line, we step on the brake many times. It is referred as multi-level deceleration. This paper applies this idea into a mobile robot, which needs to arrive the goal accurately. Simulation results demonstrate effectiveness of multi-level deceleration.

Keywords—Goal Test, Deceleration, Velocity Profile

1. Introduction

A mobile robot needs to reach the target position accurately to perform its tasks. A intelligent robot usually composed of two layers, computing system and controlling system. The computing system executes high-level navigation tasks such as path planning and following, while the controlling system performs low-level wheel velocity control. The high-level navigation system usually does not control wheel velocity directly, but it just sends reference velocity to the low-level system. For example, it sends zero reference wheel velocity directly, but it just sends reference velocity to the control board when it needs to stop. The communication between two layers is usually performed through RS-232/485, CAN, Ethernet, USB, and others. When a robot send stop command near the goal, unpredictable factors cause inaccurate final position (e.g. communication delay, nonlinear deceleration profile, and wheel slippage).

A similar problem has been in our life when we stop a car in front of a stop line (Figure 1(a)). We do not step on the brake just once in this situation, but we press it many times. It causes multi-level deceleration as like Figure 1(b), and makes our car stop in front of the line accurately. Why do we push the brake many times, not just once? The most important reason is to reach near the line with low velocity. The slow approach permits us enough time to observe current position and decide the final stop. Moreover, the low velocity leads shorter braking distance, which reduces effect of unpredictable physics.

This paper suggests similar scheme with above multiple deceleration to achieve accurate goal arrival. Two-level deceleration is formulated in Section 2. The proposed scheme is operated in the high-level computing system, which does not control actuators directly. It is the most significant difference with point stabilization which is investigated in control theory and usually operated in the control system. Experimental results in Section 3 presents effectiveness of the proposed scheme, two-level deceleration.

This work was supported partly by the R&D program of the Korea Ministry of Knowledge and Economy (MKE) and the Korea Evaluation Institute of Industrial Technology (KEIT). (2008-S-031-01, Hybrid u-Robot Service System Technology Development for Ubiquitous City)

2. Two-level Deceleration Scheme

2.1 Problem Formulation

A robot pursues the given trajectory, whose goal position is noted as \((x_g, y_g)\). The robot periodically measures its current pose and velocity: current position \((x_r, y_r, \theta)\), linear velocity \(v_r\), and angular velocity \(\omega\). A user can send the desired reference velocity to the robot using the predefined command, \(\text{ControlVelocity}(v, \omega)\). The command makes the robot increase or decrease its velocity toward the given reference. The available linear and angular velocity is noted as \(v_m\) and \(\omega_m\), respectively. If the user executes \(\text{ControlVelocity}(0, 0)\), the robot decelerates and finally stops. However, its braking distance is predicted inaccurately due to many unpredictable factors such as communication delay and wheel slippage. Such inaccurate estimation causes early or late timing to send the command. The goal arrival problem is to make the robot reach the goal accurately using the command, \(\text{ControlVelocity}\). Its accuracy is measured by Euclidean distance as follows:

\[
d(x, y) = \sqrt{(x-x_g)^2 + (y-y_g)^2}.
\]

2.2 One-level Deceleration

The constant acceleration assumption makes many calculation simple. Under this assumption, the necessary distance to change linear velocity \(v_o\) to \(v\) is derived as follows:

\[
s(v_o, v) = \frac{v^2 - v_o^2}{2a},
\]

where \(a\) is the constant acceleration. Therefore, the braking distance simply becomes \(s(v_c, 0) = -v_c^2/2a\ (a < 0)\).

A robot usually performs the goal test periodically, which examines whether the robot reach the goal or not. The conventional goal test is the simple comparison as follows:

\[
\text{IsGoal}(t) = \begin{cases} 1 & d(x_t, y_t) \leq s(v_c, 0) \\ 0 & \text{otherwise} \end{cases}
\]
The robot sends the stop command and finishes its trajectory following when the goal test returns true \((T)\).

### 2.3 Two-level Deceleration

In two-level situation, the addition deceleration makes velocity slow enough to have small braking distance (Figure 2). The smaller braking distance means shorter time to stop, which causes smaller wheel slippage and less effect of non-linear deceleration profile. To perform two-level deceleration, the goal test does not return only true or false, but also the maximum operating velocity as follows:

\[
\text{IsGoal}(\cdot) = \begin{cases} 
(T, 0) & d(x_c, v_c) \leq s(v_c, 0) \\
(F, v_b) & d(x_c, v_c) \leq s(v_c, 0) + \delta_b \\
(F, v_m) & \text{otherwise}
\end{cases}
\]

where \(v_b\) is the brake velocity and \(\delta_b\) is the brake margin (Figure 2). The second threshold, \(s(v_c, 0) + \delta_b\), comes from \(s(v_c, v_b) + \delta_b + s(v_b, 0)\) since \(s(v_c, v_b) + s(v_b, 0) = s(v_c, 0)\).

Two parameters are necessary to define the second level. The brake velocity \(v_b\) is the operating linear velocity in the second level, which is sustained as long as the brake margin \(\delta_b\). The brake margin compensates underestimated distance \(s(v_c, v_b)\) when the robot changes its velocity from \(v_c\) to \(v_b\). As the brake velocity becomes smaller, the robot will converge the goal more slowly. The brake margin is opposite. As the brake margin becomes shorter, the robot will converge velocity slow enough to have small braking distance (Figure 2). The smaller braking distance means shorter time to stop.

\[
\delta_b = \frac{-v_b^2}{2a} \quad \text{where} \quad a < 0.
\]

Their three extreme cases become one-level deceleration: \(v_b = 0\) or \(v_b = v_m\) or \(\delta_b \leq -v_b^2/2a\).

### 3. Experimental Results

#### 3.1 Simulation Configuration

Simulation was performed to verify effectiveness of two-level acceleration. The robot was 2.0 meters away from the goal, whose initial velocity was 0.6 m/s and 0.0 rad/s. It can accelerate or decelerate as much as 0.5 m/s². Its period of goal test was 0.05 sec (20 Hz). The communication delay was modeled as additive Gaussian noise \(N(0.003, 0.001)\).

The robot moved 10% more than the estimation on the average due to wheel slippage, which is also modeled as additive Gaussian noise. The observation noise was applied when the robot measures position and velocity, which is additive Gaussian \(N(0,0,0.01)\). The brake velocity and margin were adjusted as 0.2 m/s and 0.3 m, respectively. Simulation were performed 100 times for statistically representative results.

#### 3.2 Result and Discussion

As a result, **two-level deceleration was 2 times more accurate than one-level, but it causes 1 second late arrival.** Figure 3 is a representative example. The more accurate arrival was accomplished when the brake velocity became smaller, but it occurred much later arrival. In case of \(v_b = 0.1\) m/s, accuracy increased 10 time more than one-level, but arrival was 4.5 second late. The brake margin did not affect accuracy significantly if it satisfied Equation 5 sufficiently. The user should be cautious because the later arrival results from the longer brake margin.

![Simulation Results](image)

### 4. Conclusion

This paper applied common driving skill, multi-level deceleration, into the mobile robot. The proposed method, two-level deceleration, can achieve accurate goal arrival in contrast conventional one-level brake. Simulation also confirmed its effectiveness. Two-level deceleration needs just two parameters, which are easily adjusted by the user due to its simple physical meaning.