ICCT 2013

"The 3rd International Conference on Convergence Technology 2013"

Vol. 2 No. 1

Date: July 3~6, 2013
Place: Chiang Mai, Thailand
Co-organized by:
- Korea Convergence Society
- NTIS Division and GSDC Center at Korea Institute of Science and Technology Information (KISTI)
- Korea Institute of Information, Electronics and Communication Technology
- IT Convergence Society for SMB
- The Society of e-Learning
- The Korea Society of Digital Policy & Management
- The Korean Association for Comparative Government
- Xian Jiaotong–Liverpool University, China

Sponsored by:
- KISTI NTIS Division & GSDC Center, Selim Technical Service Group Co., Ltd.
- UCUBE Co., Ltd., KYUNGBONG Co., Ltd.
- LG CNS, SK Telecom Co., Ltd., MA Information Technology Co., Ltd.
- DUPLEX Co., Ltd., Informade Co., Ltd., SANE Co., Ltd., ALLFORLIND Co., Ltd.
- Dong-HiTech Co., Ltd., KTNET, LOTTE Data Communication Company, SUNGWON Information Technology Co., Ltd.
- Korea IT Consulting Co., Ltd., UNIEVER Co., Ltd., Daewoo Information System Co., Ltd.
- Innovdn.inc, GS Intervision, Bluerever, APPLTEK, WiNUSTech Co., Ltd.
- SEJOONG Information Co., Ltd., I&T Inc., HIMS, DAELIM Co., Ltd.
- BOKANG SYSTEM Co., Ltd.

KCS KOREA CONVERGENCE SOCIETY
http://www.kcons.or.kr
Abstract A practical and useful trajectory planning considering physical limitations through convolution in joint space for mobile robots is presented. Smooth commands in joint velocity considering velocity and systems limit along the path are computed. The effectiveness of the algorithm is shown through numerical simulations and practical application to a robot simulator.

Keywords: Trajectory, Bezier curve, Mobile robots, Physical Limits

1. Introduction

A mobile robot is a nonholonomic system, and various trajectory planning approaches have been addressed by researchers. The trajectory denotes the path that robot should traverse as a function of time. For the mobile robot, the trajectory in task space is position, \((x(t), y(t))\) in the Cartesian coordinate, and orientation, \(\theta(t)\), for the center of the robot so that its configuration \(q(t) = [x(t), y(t), \theta(t)]^T\) [1].

In this case, the continuous path designed in task space translates to the requirement for twice differentiable trajectories in joint space that means continuous velocity commands for each of two wheels. A convolution-based trajectory generation method satisfying system specifications was suggested in [2]. This paper utilizes a Bezier curve-based two-differentiable path planning with satisfying heading angles and follows the designed path while considering velocity limits of translational velocity and rotational velocity. We suggest an algorithm providing smooth velocity profiles in joint space which are actuator command to the robot while considering the velocity limits. Finally, we consider the path error in terms of sampling time when making trajectory which follows the predefined Bezier curved path.

2. Smooth trajectory planning considering velocity limits

The trajectory planning takes a smooth path \(\rho(u)\) [1], which is a Bezier curve-based path that considers the rotating angle using a constant parameter \(u\), as an input and tries to find velocity profiles in joint space that follows this path. Therefore, we first build a smooth path. With considering configurations at its starting and destination, a trajectory is commonly generated using a third degree Bezier curve consisting of a starting point \(P_s\), final destination \(P_f\), and control points \(C_1\) and \(C_2\) [1, 3].

When physical limits are \(v_{\text{max}} = 0.5\text{m/s}\), \(a_{\text{max}} = 0.2\text{m/s}^2\), and \(j_{\text{max}} = 0.2\text{m/s}^3\), the velocity trajectory satisfies physical limits moving from \((0, 0, 45^\circ)\) to \((3, 1, 45^\circ)\) through \((1, 1, 0^\circ)\). The path can be designed using Bezier curve. The center velocity is obtained using convolution to travel the distance of the path, which considers velocity limit. And the path \(\rho(u(t))\) of considering heading angles, which is shown as dotted-line in Figure 3, is obtained center velocity. Then joint velocities could not satisfy physical limits where the angular velocities are high.

The joint velocity commands can be maintained within the maximum velocity limit by correcting the center velocity as shown in equation (1). By doing convolution again with modified velocity limit \(v'_{\text{max}}\), joint velocity profiles can be generated that satisfy actual actuator’s physical limits.

\[
v_{\text{comp}} = \frac{\text{max}(v_y - y)}{2}, v'_{\text{max}} = v_{\text{max}} - v_{\text{comp}} \quad (1)
\]

In this way, the velocity commands for two wheels satisfying actuator’s physical limits along the curved path are shown in Figure 2. The distance traveled driven by the generated joint velocity commands are constant due to the convolution characteristics. Only the traveling time is slightly increased because of the limited velocity.

We applied the proposed algorithm to the robot simulator, Marilou Robotics Studio in anyKode [4]. Figure 3 shows simulation
result in task space obtained through the drive by the velocity commands for two wheels, where the robot dimension is $r = 10\text{cm}$, $D = 40\text{cm}$. The results showed that the robot successfully followed the planned path of $\rho(u(t))$ denoted as dotted-line and the traveling time slightly increased after considering the actual actuator’s velocity limits. The proposed trajectory generation method can be used to generate velocity commands for actual driving and controlling.

![Figure 1. Joint velocity trajectories with center velocity limit.](image1)

![Figure 2. Joint velocity trajectories with actuator’s velocity limit.](image2)

![Figure 3. Simulation result driven by joint velocity commands.](image3)

3. Path error according to sampling time
When applying the proposed method, we should consider sampling time to produce velocity commands. Figure 4 shows the generated trajectory according to sampling times. The results show that the errors increase as sampling times increase. However, if a system cannot drive commands within sampling time, the resulted trajectory also cannot follow the predefined path.

![Figure 4. Trajectory according to sampling time.](image4)

4. Conclusion
A practical velocity trajectory generation to travel smoothly along a curved path within the actuator’s physical limits was proposed. And we considered the effect of sampling time to cope with control loop. In the future, this method can be applied to obstacle avoidance algorithms that satisfy velocity limits at any configuration.

Acknowledgments. This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (No. 2012-006057).

5. References