Smooth Trajectory Planning Along Bezier Curve for Mobile Robots with Velocity Constraints

Gil Jin Yang and Byoung Wook Choi*

Department of Electrical and Information Engineering
Seoul National University of Science and Technology, Seoul, South Korea
yang6495@gmail.com, bwchoi@seoultech.ac.kr

Abstract

This paper presents a smooth path planning method considering physical limits for two-wheeled mobile robots (TMRs). A Bezier curve is utilized to make an S-curve path. A convolution operator is used to generate the center velocity trajectory to travel the distance of the planned path while considering the physical limits. The trajectory gained through convolution does not consider the direction angle of the TMR, so a transformational method for a center velocity trajectory following the planned path as a function of time of parameter for the Bezier curve is presented. Finally, the joint space velocity is computed to drive the TMR from the center velocity. The effectiveness of the proposed method was performed through numerical simulations. This algorithm can be used for path planning to optimize travel time and energy consumption.

Keywords: *TMR, Bezier curve, Smooth path planning, Convolution, Physical Limits*

1 Introduction

Two-wheeled mobile robots (TMRs) are recently becoming widely used as cleaning robots and intelligent service robots; thus, extensive research is underway on trajectory planning to minimize energy and optimize traveling time as well as to resolve issues regarding smooth traveling toward the desired destinations in workspaces [1-4].

A navigation system for a TMR largely consists of a path planner, a trajectory generator and a tracking controller. Path planning is about generating smooth paths while maintaining the desired position in workspaces. The trajectory generator aims to generate a velocity profile for the planned paths as a function of time. The tracking controller and driving controller are control systems that allow a TMR to travel along the predefined trajectory at a desired time while staying within its physical limits.

If the physical limits of a TMR during path planning and trajectory generation are considered, potential damage to a TMR can be reduced; trajectory tracking accuracy and tracking velocity can be improved [3]. To this end, velocity trajectory planning methods using a convolution operator have been suggested that consider the physical limits of a TMR in workspaces [2, 5]. However, the suggested method did not consider the direction angle, which is part of a TMR's kinematics. The method only considered the translational velocity and path as opposed to the center of a TMR in Cartesian coordinates. A smooth path planning method that considers the initial and final direction angles is the basic goal in path planning for a TMR.

Application to an actual TMR is also difficult since only the translational velocity limits at the center point in Cartesian coordinates is considered, and not the physical limitations of

* Corresponding author

actuators for driving the two wheels, which are dependent on the variations in angular velocity.

When generating a trajectory for a TMR, modifying the directions of a TMR after it stops during operation has been used because a discontinuity point may cause a slip or path deviation. In order to overcome this issue, research on path planning with continuous curvature for a TMR with kinematic limits has been conducted. Path planning methods have been studied for a TMR arriving in a desired position based on a starting position and direction angle using a Bezier curve [6].

In this study, a path based on a Bezier curve was generated in order to build a smooth path while considering the direction angle. A convolution operator was used to generate the central velocity to travel the planned path. In this process, the velocity trajectory can be generated while considering the maximum velocity and acceleration according to the physical limits of a TMR. The velocity trajectory gained through convolution is a trajectory which a robot travels such that the given distance does not consider the direction angle of the TMR. In order to consider the direction angle of the TMR, a transformation method for the trajectory is presented that consists of segmented paths along the designed Bezier curve with the central velocity generated through convolution. The trajectory obtained through the transformation process can be used for the TMR to smoothly follow the planned path while staying within the physical limits. Finally, a trajectory generation method in joint space that can be used as an actuator command for the TMR driving is proposed. The joint space trajectory limits the driving velocity profile along the two wheels in order to consider the actuator's physical limitations that depend on the direction angle of the central velocity.

In order to determine the effectiveness of the proposed method, numerical simulations were performed. The application of the planned trajectory to a simulator showed that the robot carried out desired tasks well while staying within its physical limits. This trajectory can be used for path planning to optimize time and energy consumption.

2. Bezier Curve based Path Planning

As shown in Figure 1, a TMR is represented in the coordinate system using the robot's central position and direction angle. The position consists of a world frame coordinate system and robot frame coordinate system. A TMR's position P_c is defined on the coordinate system as follows:

$$P_c = [x_c, y_c, \theta_c]^T, \tag{1}$$

where x_c, y_c, θ_c denote a robot's position and direction angle respectively. TMR's kinematic model can be obtained as follows:

$$P_c = \begin{bmatrix} \frac{r}{2}cos\theta_c & \frac{r}{2}cos\theta_c \\ \frac{r}{2}sin\theta_c & \frac{r}{2}sin\theta_c \\ \frac{r}{D} & -\frac{r}{D} \end{bmatrix} \begin{bmatrix} \omega_r \\ \omega_l \end{bmatrix}, \tag{2}$$

where r denotes the radius of a robot's wheels, D denotes the distance between its two wheels, ω_r denotes the right wheel's angular velocity, and ω_l denotes the left wheel's angular velocity.

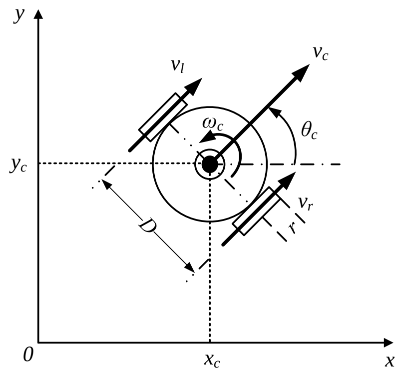

Figure 1. Kinematics of TMR

When planning a path for TMR's, the position and direction angle at its starting point and destination should be considered and a curved trajectory is commonly generated using Bezier curves [6]. As shown in Figure 2, a trajectory is generated using a Bezier curve consisting of an initial point $P_i(A_0, B_0)$, end point $P_f(A_3, B_3)$, and control points $C_1(A_1, B_1)$ and $C_2(A_2, B_2)$. An equation for the Bezier curve is calculated using C_1 and C_2. The equation of Bezier curve is given below in equation (3).

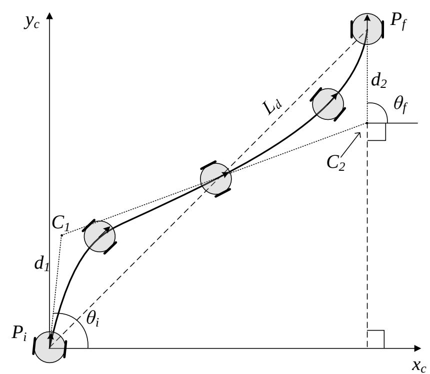

Figure 2. Bezier Curve-based path planning

$$\begin{aligned} x(u) &= \sum_{i=0}^{3} A_i J_{n,i}(u) \\ &= A_0(1-u)^3 + 3A_1 u(1-u)^2 + 3A_2 u^2(1-u) + A_3 u^3, \end{aligned} \tag{3-a}$$

$$\begin{aligned} y(u) &= \sum_{i=0}^{3} B_i J_{n,i}(u) \\ &= B_0(1-u)^3 + 3B_1 u(1-u)^2 + 3B_2 u^2(1-u) + B_3 u^3, \end{aligned} \tag{3-b}$$

In equation (3), u is an arbitrary value where $0 \le u \le 1$ and can be used to generate a smooth curve from a starting point to a target point: a more precise Bezier curve with a smaller increase. The path given by equation (3) does not consider velocity and is only parameterized by u.

3. Convolution based Trajectory Planning Following Bezier Curve

There has been research that the path generation method may use a convolution operator to create a central velocity trajectory of a TMR for smooth path generation while satisfying physical limits [2, 3].

In order to use convolution, a square-wave function $y_0(t)$ is defined as follows:

$$y_0(t) = \begin{cases} v_0, \; 0 \le t \le t_0 \\ 0, \; otherwise \end{cases}, \tag{4}$$

where the nth-applying convolution function $h_n(t)$ is defined as a square-wave function with the unit area in $0 \le t \le t_n$ as follows:

$$h_n(t) = \begin{cases} 1/t_n, \; 0 \le t \le t_n \\ 0, \; otherwise \end{cases}. \tag{5}$$

If function $y_n(t)$ is a resulting function to which the nth convolution is applied, the result of convolution $y_0(t)$ and $h_1(t)$ can be represented as $y_1(t)$ and $y_2(t)$ denotes the result of $y_1(t)$ and $h_2(t)$ convolution. The velocity function $v_c(t)$ generates the velocity command of the differentiable S-curve that considers the maximum velocity v_{max} for the robot to travel the distance S, as shown in Figure 3.

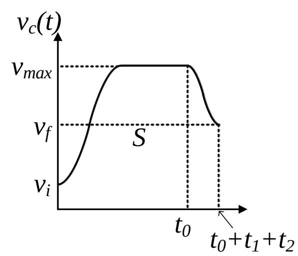

Figure 3. Convolution-based velocity command trajectory

Let a Bezier-curve-based path as shown in Figure 2 that considers the direction angle using a constant value u be $\rho(u)$. The distance traveled is calculated using formula (6) to generate the central velocity trajectory for the robot to travel along the distance S, as shown in Figure 3. The curved distance B_d along the path $\rho(u)$ from P_i to P_f as in Figure 2, is calculated as follows:

$$B_d = \sum_{u=0}^{1} \Delta\rho(u) = \sum_{u=0}^{1} \sqrt{(x(u+\Delta u) - x(u))^2 + (y(u+\Delta u) - y(u))^2} \tag{6}$$

The calculated distance B_d is the actual distance traveled along the path designed with Bezier curve which has a smooth curve. To generate the center velocity trajectory of a TMR using convolution, the distance S is thus used as an input value. Therefore, if the center velocity trajectory $v_c(t)$ is generated to have the traveling distance as $S = B_d$, then the trajectory using the advantages of convolution while considering velocity limits can make a smooth path. Here, v_i, v_f, v_{max} and the sampling time can be arbitrarily set according to the specifications of the TMR [2-3].

The generated central velocity trajectory of $v_c(t)$, as shown in Figure 3, travels along the distance S . However, the central velocity trajectory of TMR does not consider the direction of the robot. In other words, for any position $(x(u_i), y(u_i))$, the robot travels with velocity $v_c(t_i)$, as shown in Figure 2. In equation (2), the subsequent position can be moved to an entirely different position depending on the angle θ_i. In order to consider the positions in task space that depend on velocities in paths with direction angles, the parameter $u(t)$ of Bezier curve for the distance during the sampling time should be determined and calculated using equation (7). The trajectory $\rho(u(t))$ with the direction angle can be obtained by inputting the determined $u(t)$ into equation (3). In $\rho(u(t))$, if the sampling time is shorter, the path can more accurately follow $\rho(u)$ as generated by constant parameter value u.

$$u(t) = \sum_{t=0}^{t_0+t_1+t_2} v_c(t) \Big/ B_d \tag{7}$$

Here, $u(t)$ is defined as $0 \le u(t) \le 1$ and represents the parameter of the Bezier curve that depends on the central velocity. The trajectory generated by using $u(t)$ satisfies the maximum velocity allowed by the physical limits of a TMR while following the curved path with respect to the direction angles.

4. Simulation Results

Figure 4 illustrates the central velocity trajectory that satisfies the physical limits from the starting point $P_i(0, 0, 90°)$ to the target point $P_f(4, 4, 90°)$ and a Bezier curve trajectory tracking it. In this figure, the distance between positions of the trajectory is the distance driven by the central velocity function during sampling time. The results show that the synthesized Bezier curved trajectory was generated depending on the central velocity function's trajectory. Direction angles $\Delta\theta$ at each section and the angular velocity ω_c at the center were calculated as follows and are illustrated in Figure 4.

$$\Delta\theta = tan^{-1} \frac{y(\Delta u(t))}{x(\Delta u(t))} , \tag{8}$$

$$\omega_c = \frac{\Delta\theta}{\Delta t} \tag{9}$$

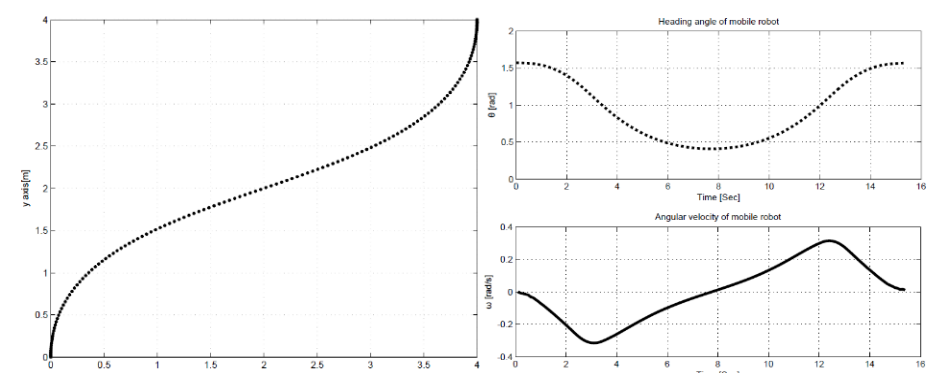

Figure 4. Smooth trajectory, direction angle and angular velocity considering maximum velocity limits

The trajectory for the TMR was generated as shown in Figure 4. The generated trajectory satisfies the physical limits described above and allows the TMR to travel along the curved path using its central velocity. The actual command for actuating the TMR is the angular velocity for both wheels. It can generate wheel velocity commands in joint space using equations (10) and (11):

$$\begin{aligned} \omega_r &= \frac{1}{r}(v_c + D/2 \cdot \omega_c) \\ \omega_l &= \frac{1}{r}(v_c - D/2 \cdot \omega_c) \end{aligned}, \tag{10}$$

$$\begin{aligned} v_r &= r\omega_r \\ v_l &= r\omega_l \end{aligned}. \tag{11}$$

The velocity command trajectory for two wheels obtained by formula (11) becomes the actual velocity command for the TMR, and the robot's angular velocity ω_c is represented by the difference between the two wheels' translational velocities as shown in equation (12):

$$\omega_c = \frac{v_r - v_l}{D}. \tag{12}$$

Figure 5 shows that when physical limits are v_{max}=0.5m/s, a_{max}=0.2m/s^2 and j_{max}=0.2m/s^3, then the central velocity trajectory satisfies the physical limits moving from the starting position (0, 0, 90°) to the target position (4, 4, 90°). When the velocity commands for the two wheels are generated, angular velocity shown in Figure 4 is used. The joint velocity commands for two wheels is used to drive two wheels to follow the Bezier curve based trajectory

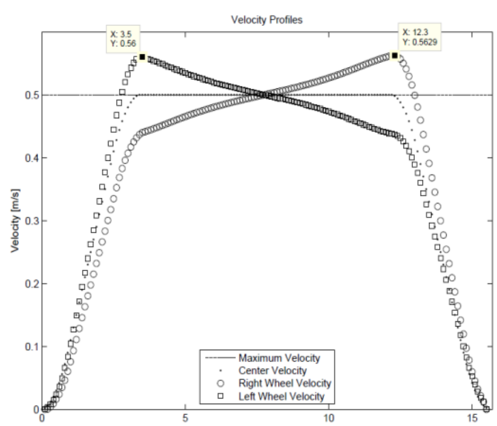

Figure 5. Velocity commands of two-wheels following smooth trajectory

Figure 6 shows the TMR's trajectory in task space obtained through the drive by the velocity commands for two wheels as shown in Figure 5. The results show that the robot successfully followed the Bezier curve along the planned path. Figure 7 shows the simulation results driven by actuator commands on the anyKode, Marilou Robotics Studio [10].

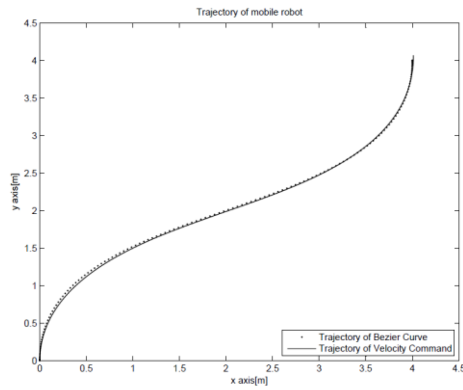

Figure 6. S-Curve and Trajectory of TMR

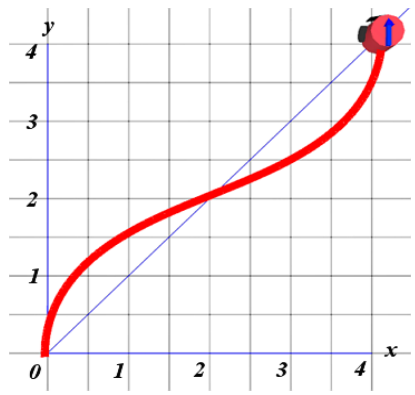

Figure 7. Trace of TMR driven by Actuator Velocity Commands

Figure 8 shows the tracking error between the Bezier curve and the trajectory generated according to sampling time of 1ms, 50ms, 10ms in equation (7). The error increases as angular velocity and sampling time increases. The effect of sampling time should be considered to control mobile robot.

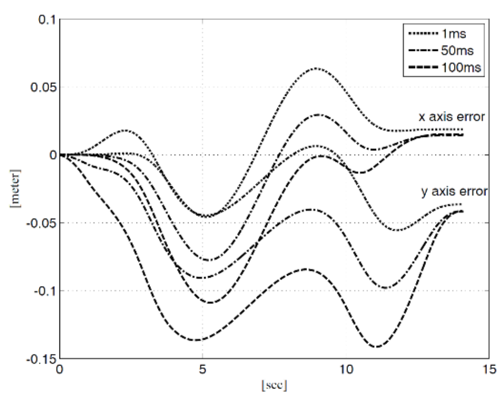

Figure 8. Error Gap While Travelling Along an S-Curve

The previous curve is S-curve so the tracking error could be compensated in whole path. We applied the proposed algorithm to another path which has C-curve. Another path is shown as a C-curve in Figure 9. The figure shows that when physical limits are v_{max}=0.5m/s, a_{max}=0.2m/s^2 and j_{max}=0.2m/s^3, the central velocity trajectory satisfies the physical limits moving from the starting position (0, 0, 0°) to the target position (4, 4, 90°).

Figure 10 shows the velocity commands of the two wheels following the path shown in Figure 9. In C-curve path, heading angle of TMR is changing monotonically. In this case, Figure 11 shows trajectory of TMR driven by the joint velocity commands described in Figure 10, which follows C-curve. Compared to the S-curve shown in Figure 6, the error is greater in the C-curve.

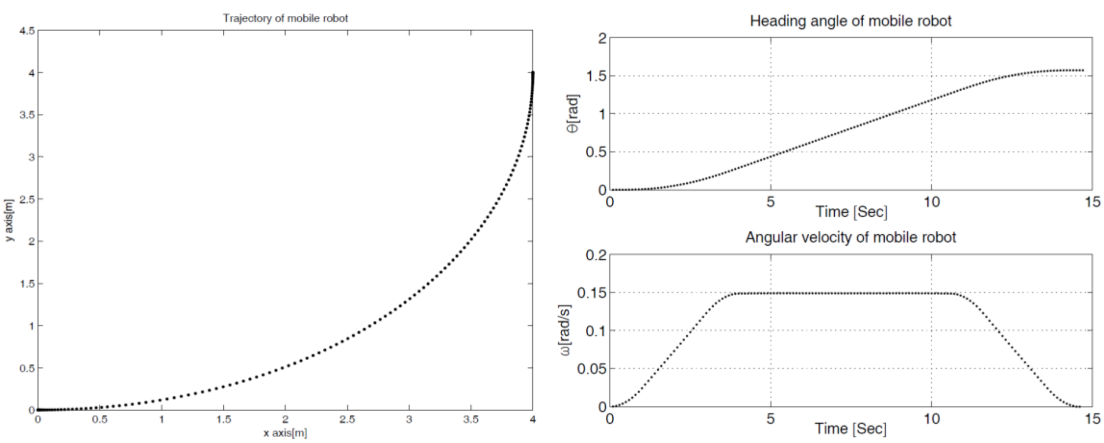

Figure 9. Smooth trajectory, direction angle and angular velocity considering maximum velocity limits

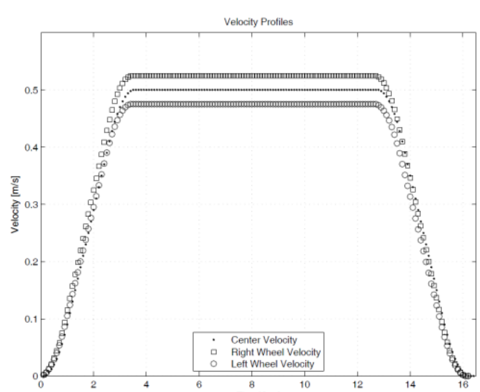

Figure 10. Velocity commands of two-wheels following smooth trajectory

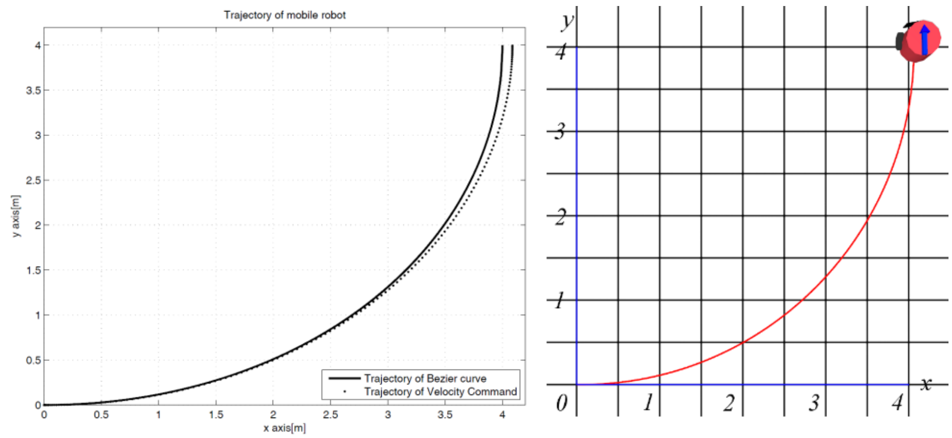

Figure 11. Trajectories of TMR and trace of TMR driven by actuator velocity commands

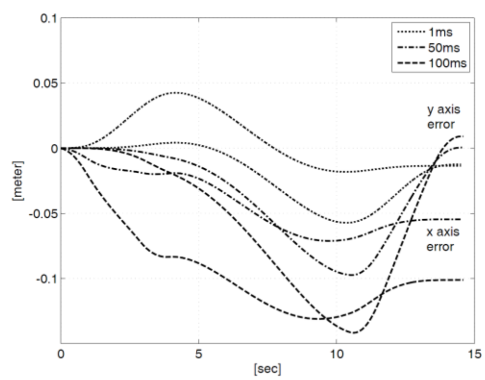

Figure 12. Error Gap While Travelling Along the C-Curve

Figure 12 shows the error of the path between the ideal parametric Bezier curve by using equation (3) and the generated path considering physical limits with various sampling time by using equation (7). The error was computed by finding the difference between the ideal smooth trajectory (as shown by the solid-line graph in Figure 11) and the trajectory of the Bezier curve (as shown by the dotted-line graph in Figure 11). The tracking error resulted from sampling time is also shown Cartesian trajectory simulated using the joint velocity commands. Compared to Figure 8, the errors in x coordinates become larger as tracking C-curve path. The shape of the error gap is different in depending on the target path shape. Therefore, the path's shape should be considered. If the sampling time is shorter, error is decreased.

5. Conclusions

A velocity command trajectory generation method was proposed that enables a TMR to travel smoothly along a curved path while staying within the actuator's physical limits for smooth run and control.

The proposed velocity trajectory generation method generates a trajectory to satisfy the maximum velocity as opposed to a central velocity of TMR using the characteristics of convolution, and the central velocity trajectory follows the Bezier curve based path to travel smoothly. In the future, this trajectory generation method can be applied to obstacle avoidance algorithms that satisfy velocity limits at the any points. Research on continuous path generation at the any points to satisfy physical limits is currently underway.

Tracking errors according to the sampling time during convolution and transformation process was examined. For smooth control, the effect of sampling time should be considered. If the velocity trajectory performs at a real-time operating system, then tracking error can be reduced. The performance evaluations of real-time mechanisms can predict the tracking error according to the system, performance evaluations of real-time mechanisms [7]. It can also analyze tracking error and apply the results to the controller so that tracking error can be reduced [8-9].

The path planning method proposed in this paper can be utilized for a path planning with optimized travelling time and an energy-efficient path planning that considers the limited battery power of a running robot.

Acknowledgements

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).

References

[1] J. J. Craig, “Introduction to Robotics”, Prentice-Hall, **(2005)**.
[2] G. Lee, B. J. Yi, D. I. Kim and Y. J. Choi, “New Robotic Motion Generation using Digital Convolution with Physical System Limitation”, IEEE Conference on Decision and Control and European Control Conference, **(2011)**, pp. 698-703.
[3] G. Lee, D. I. Kim and Y. J. Choi, “Faster and Smoother Trajectory Generation considering Physical System Limits under Discontinuously Assigned Target Angles”, IEEE International Conference on Mechatronics and Automation, **(2012)**, pp. 1196-1201.
[4] M. S. Jang, E. H. Lee and S. B. Choi, “A Study on Human Robot Interaction Technology Using a Circular Coordinate System for the Remote Control of the Mobile Robot”, International Journal of Control and Automation, vol. 5, **(2012)**, pp. 117-130.
[5] M. Lepetic, G. Klancar, I. Skrjanc, D. Matko and B. Potocnik, “Time optimal path planning considering acceleration limits”, Robotics and Automation Systems, vol. 45, **(2003)**, pp. 199-210.
[6] K. G. Jolly, R. S. Kumara and R. Vijayakumara, “A Bezier curve based path planning in a multi-agent robot soccer system without violating the acceleration limits”, Robotics and Automation Systems, vol. 57, **(2009)**, pp. 23-33.
[7] J. H. Koh and B. W. Choi, “Real-time Performance of Real-time Mechanisms for RTAI and Xenomai in Various Running Conditions”, International Journal of Control and Automation, vol. 6, **(2013)**, pp. 235-246.
[8] S. Sheel and O. Gupta, “High Performance Fuzzy Adaptive PID Speed Control of a Converter Driven DC Motor”, International Journal of Control and Automation, vol. 5, **(2012)**, pp. 71-88.
[9] A. Abdelkrim, C. Ghorbel and M. Benrejeb, “LMI-based tracking control for Takagi-Sugeno fuzzy model”, International Journal of Control and Automation, vol. 3, **(2010)**, pp. 21-36.
[10] AnyKode, Marilou Robotics Studio, http://www.anykode.com.