Trajectory Tracking Controller Based on Linear Model Predictive Control for Omni-Wheeled Mobile Robots with Velocity Command Limits

Reona Suzuki[*a)] Non-member, Masahide Ito[*b)] Senior Member

This paper presents a trajectory tracking controller based on linear model predictive control for omni-wheeled mobile robots. The controller predicts the system input/output based on the extended robot model while takes the velocity command limits into account in the same way of our previous controller. The performance of the proposed controller was evaluated through an experiment.

Keywords: linear model predictive control, trajectory tracking, omni-wheeled robots, velocity command limits, constraint reduction

1. Introduction

RoboCup Soccer Small Size League (SSL) is a robotic soccer competition in which omni-wheeled mobile robots play soccer autonomously through global vision and wireless communication. The rules changed last year include expanding the field and increasing the number of robots, which implies that more advanced cooperative plays are important to dominate the game. Such plays are based on the motion control performance of individual robot. In Ref. (1), focusing on the translational motion and reducing its quadratic constraint into linear ones designed a trajectory tracking controller based on linear model predictive control (LMPC). It is found, however, that motion with relatively major change of robot's orientation can deteriorate the control performance. To improve this drawback, the paper extends the LMPC-based controller of Ref. (1) to the one taking the robot orientation into account.

2. Overview of SSL System

The SSL system is shown in Fig. 1. Cameras overlooking the field provide top view images, and then the image-processing server called SSL-Vision extracts information of robot's position and orientation, ball position and so on from the images. Based on such information, a team computer estimates the game situation, decides an appropriate strategy, and transmits the velocity commands to each individual robot via radio communication. In the SSL, omni-wheeled mobile robots are used as soccer players. Each robot moves according to the velocity commands by a PID controller on the board. The above-mentioned process is updated every 1/60 seconds.

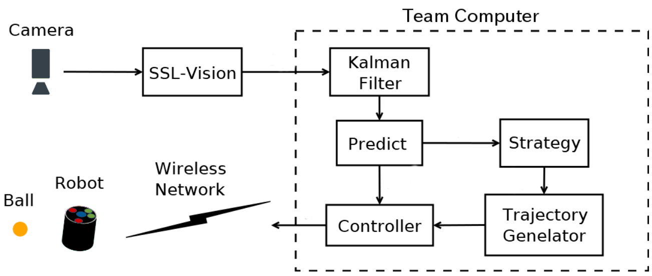

Fig. 1. Typical process of SSL system

The control flow in the team computer can abstract as follows. A strategy chosen by a team computer provides the target position for each robot. A set-point trajectory is generated from the current position to the target position by using a polynomial. The velocity commands are given such that the robot tracks the trajectory by appropriate feed back. Note that we need to design a controller in consideration of hardware limitation so as not to deteriorate the control performance.

3. LMPC-based controller in consideration of velocity command limits

Especially in MPC, while predicting future input/output by the discrete time model of the robot, Ref. (1) proposed an LMPC-based controller as a trajectory tracking algorithm in the team computer. It takes limits of transmitting velocity commands into account, but ignores kinematics of robot's orientation on the model for simplicity. This paper extends the model with respect to robot's orientation and also derives an LMPC-based controller using the extended model. As a model including kinematics of robot's orientation, we consider

$$\begin{cases} \boldsymbol{x}(k+1) = \boldsymbol{x}(k) + T_s \begin{bmatrix} \alpha_x & 0 & 0 \\ 0 & \alpha_y & 0 \\ 0 & 0 & \alpha_\phi \end{bmatrix} \boldsymbol{u}(k - H_w) \\ \boldsymbol{y}(k) = \boldsymbol{x}(k) \end{cases}, \tag{1}$$

where $\boldsymbol{u} = [{}^r v_x, {}^r v_y, {}^r \omega]^\top$ is input, $\boldsymbol{x} = [{}^r p_x, {}^r p_y, {}^r \theta]^\top$ is state, $\boldsymbol{y}$ is output, T_s is sampling time, H_w is time delay, and $\alpha_x, \alpha_y, \alpha_\phi$ are scale parameters, respectively. Letting us express set-point trajectories for output and input as $\boldsymbol{y}^\star = [y_1{}^\star, y_2{}^\star, y_3{}^\star]^\top$ and $\boldsymbol{u}^\star = [u_1{}^\star, u_2{}^\star, u_3{}^\star]^\top$, we adopt the following performance index:

a) Correspondence to: is151048@cis.aichi-pu.ac.jp
b) Correspondence to: masa-ito@ist.aichi-pu.ac.jp
* School of Information Science and Technology, Aichi Prefectural University
1522-3 Ibaragabasama, Nagakute, Aichi 480-1198, Japan.

$$V(k) = \sum_{i=H_w}^{H_p} \|\hat{\boldsymbol{y}}(k+i|k) - \boldsymbol{y}^\star(k+i|k)\|^2_{Q(i)} + \sum_{i=0}^{H_u} \|\hat{\boldsymbol{u}}(k+i|k) - \boldsymbol{u}^\star(k+i|k)\|^2_{\mathcal{R}(i)}\,, \qquad (2)$$

where, H_p is the prediction horizon, H_u is the control horizon, $\hat{\boldsymbol{y}}$ is the predicted output, $\hat{\boldsymbol{u}}$ is the predicted input, $\mathcal{R}(i), Q(i)$ are the weight matrices, respectively on the other hand, we have to take velocity command limits and into consideration. Due to hardware limitation, the constraint on the velocity commands are

$$\|\boldsymbol{v}\| \le \bar{v} \qquad (3)$$

and

$$|{}^r\omega| \le \bar{\omega}\,, \qquad (4)$$

where $\boldsymbol{v} = [u_1, u_2]^\top$, $\bar{v}, \bar{\omega}$ represent limits of the velocities. However, $\|\boldsymbol{v}\| \le \bar{v}$ is a quadratic constraint. We focus on LMPC because the computational cost must be low; The LMPC approach cannot can handle a (linearly constrained) quadratic program ((LC)QP), but handle a quadratically constrained quadratic program (QCQP). In Ref. (1), by using some decomposing angle

$$\phi(k) = \mathrm{atan2}\left(u_2^\star(k), u_1^\star(k)\right) \qquad (5)$$

based on values of set-point trajectories $u_1^\star, u_2^\star$, the quadratic constraint is reduced into the following linear ones.

$$|u_1| \le \bar{v}_x \quad \text{and} \quad |u_2| \le \bar{v}_y\,, \qquad (6)$$

where

$$\bar{v}_x(k) = |\bar{v}\cos\phi(k)| \quad \text{and} \quad \bar{v}_y(k) = |\bar{v}\sin\phi(k)|. \qquad (7)$$

Therefore, our proposed controller computes $\hat{\boldsymbol{u}}(k)$ online so as to minimize Eg. (2) subject to Egs. (4), (5) and (6).

4. Experiment

This section evaluates the proposed controller in comparison with the conventional ones in Ref. (1). Let $P_0(-4\,\mathrm{m}, -2\,\mathrm{m})$ be the initial position, $P_1(4\,\mathrm{m}, -2\,\mathrm{m})$ be the initial target position, and $P_2(1\,\mathrm{m}, 2\,\mathrm{m})$ be the final target position. The following experiment was conducted:

(1) Place the robot at Point P_0.

(2) Start moving the robot from P_0 to P_1; face the front of the robot toward P_1.

(3) Change the target position from P_1 to P_2 when the robot reaches within two meters of P_1.

(4) Drive the robot to P_2; face the robot toward P_2.

(5) End the experiment when the robot arrives at P_2.

The experimental result is shown in Fig. 2.

From the result, it can be seen that the proposed controller tracks the set-point trajectory with less overshoot than the conventional one.

5. Concluding Remarks

This paper has extended the controller in Ref. (1). In particular, we adopted the kinematic model describing not only translational motion but also rotational one, and then derived an LMPC controller based on the extended model. Future tasks include verifying the situation when a team strategy among multiple robots is performed.

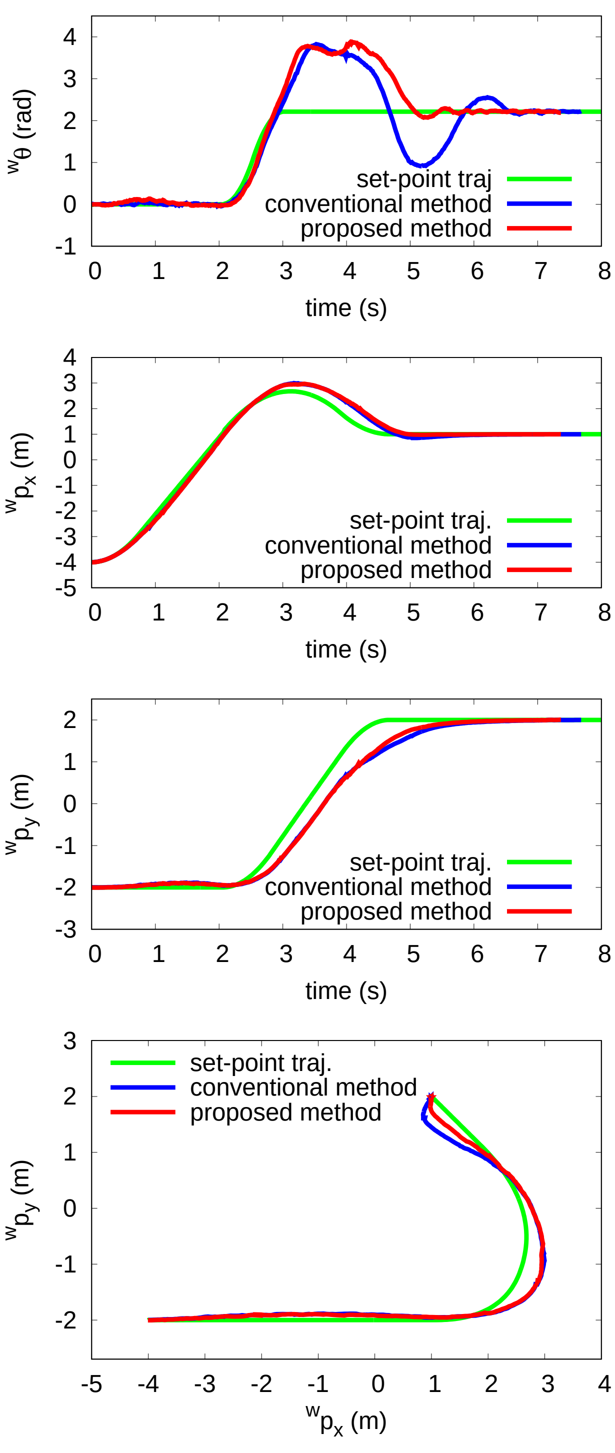

Fig. 2. Experimental result

References

(1) M.Ito, H. Kusakabe, Y.Adachi, R.Suzuki, J.Du, Y.Ando, Y.Izawa, S.Isokawa, T.Kato and T. Nanase:"RoboDragons 2018 Extended Team Description," RoboCup Soccer Small Size League, 2018