Following A Dynamic Object Through A Transient Response Adjustable MPC

Sanghyun Hong†, Justin Miller∗, and Jianbo Lu∗

Abstract—Robots can help humans fulfill various tasks in efficient and safe manners. For example, robots can navigate while loaded with packages to assist humans in delivery tasks, and they can also collaborate with Autonomous Vehicles (AV) to reach areas that AVs cannot drive. As robots are expected to interact with humans, vehicles, and other robots, following these dynamic objects is considered a crucial functionality. This paper will propose a mobile robot motion controller for following a dynamic object based on Model Predictive Control (MPC) that can reflect the dynamics of object for an appropriate response speed and ensure safety of mobile robot against predicted risks. The MPC is formulated in a way that is simple enough to compute control commands in real time without an optimization solver but can accomplish the above tasks successfully. The proposed motion controller is demonstrated through extensive simulations and real-time experiments of a test mobile robot.

I. INTRODUCTION

Human-robot interaction has gained increasing attention, allowing humans to fulfill tasks quickly and safely through assistance of robots. In order for mobile robots to interact with humans, a crucial functionality is following a person in a way that ensures efficient human-robot interaction. For example, if a mobile robot follows a delivery person with packages on it, following distance should be within a reasonable range for delivery efficiency and the mobile robot should be able to regulate its maneuvers to ensure safety of itself and its packages. In addition to human-robot interaction, following functionality is also important when a mobile robot needs to collaborate with another mobile robot, e.g., mobile robot platooning for mass package delivery.

To follow an object, mobile robots need an object tracking algorithm that recognizes the location of the object and a motion control algorithm that creates maneuvers that will follow the object in an appropriate manner. The motion control algorithm should be able to follow an object with a reasonable distance while ensuring safety of maneuver and avoiding collisions with obstacles. There have been various motion control algorithms proposed to follow people. Lee et al. [1] used Dynamic Window Approach (DWA) within the Robot Operating System (ROS) navigation stack. A drawback to that approach is that the prediction of the trajectory is limited because of a short prediction time horizon. Moreover, safe maneuvering of the mobile robot and correct following of the planned path were not taken into consideration in its baseline algorithm. They had to implement additional control schemes to compensate for limitations of DWA. Doisy et al. [2] utilized a path following algorithm that applied a discontinuous feedback control law proposed by Morin et al. [3], which is known as Sliding Mode Control (SMC). Shanee et al. [4] implemented Proportional Control (P-Control) along with the SMC for path following algorithm in order to follow a person with a pre-defined angle. However, their approach considered neither the response speed to the motion of persons nor safe maneuvering of mobile robots. Chen et al. [5] developed a vision-based person following in which a simple proportional control was used without accounting for safe maneuvering of mobile robots and collisions with obstacles. Olmedo et al. [6] developed a people following mobile robot with laser scanners, but it is based on PID control which does not take safety of the mobile robot and optimality of control into consideration. Although Leigh et al. [7] extended the methodology utilizing laser scanners so that mobile robots can track legs of person, they employed the same PID controller for following persons.

In following a highly dynamic target, e.g., a human, it is important to find optimal control commands that balance the trade-off between navigation cost (e.g., time and actuation) and safety (e.g., collision avoidance and safe cornering) before any potential risk is imminent. Therefore, in this paper, we will apply Model Predictive Control (MPC) to develop a motion control algorithm for following a dynamic object. There are several approaches used for motion control of mobile robots, e.g., Dynamic Window Approach (DWA) [8] and Trajectory Rollout (TR) [9], that are based on the predictive optimal control framework. However, they mainly focused on reaching a goal without taking into account safe maneuvering of mobile robots, response speed, or tight path following.

This paper proposes a motion controller for following a dynamic object based on MPC that is adaptable to reflect the dynamics of the object while ensuring the safety of mobile robots. The MPC is formulated in a way that is simple enough to compute control commands in real time without an optimization solver while also achieving the above objectives successfully. The proposed motion controller is demonstrated through extensive simulation and experimental tests.

II. PROBLEM DEFINITION

The problem to be addressed in this paper is related to how a mobile robot is following a dynamic object. The solution to this problem has application in various domains, e.g., package delivery and personal assistance. Consider the
case shown in Fig. 1, where a blue mobile robot is tasked to follow the orange mobile robot by tracking its relative location. Once the blue robot detects the orange robot, it sets a goal with an offset \( d_{off} \) from the orange robot, as indicated by the yellow circle in Fig. 1. Then, the blue mobile robot follows the goals that are published whenever a time period or minimum displacement is met as the orange robot moves.

The blue robot computes control commands, \( u \), in the form of linear and angular velocities that will depend on the distance, \( d(k) \), from the blue robot to the goal at time \( k \). In Fig. 1, the length of arrows in Fig. 1 represents the magnitude of the computed linear velocity. Additionally, there are constraints on the linear and angular accelerations, i.e. the upper bound on the magnitude of acceleration, to prevent excessively aggressive motion that can cause a dangerous situation. In order to follow a highly dynamic object, e.g., a human, that changes maneuvers continuously and drastically, the wheeled mobile robot should be able to account for the response speed for the highly dynamic motion of the object. In addition, for the sake of safety, the mobile robot should have the capability to avoid collisions with obstacles and loss of control.

III. FOLLOWING MOTION CONTROL WITH MODEL PREDICTIVE CONTROL

When following a dynamic object, if the object changes maneuvers, a mobile robot needs to respond to the changes of motion by accelerating linear and angular velocities so that it does not lose track of the dynamic object. In addition, if a dynamic object stays at a steady state, a mobile robot also needs to find appropriate navigation velocities to remain within an appropriate distance from the object. Meanwhile, if any potential risks are anticipated, the mobile robot should prioritize safety and compute control commands to prevent dangerous situations at the cost of causing the mobile robot to fall behind. This implies that there exists trade-offs between following distance and safety, thus leading to the need for an optimal control method. In addition, taking action preemptively against potential risks is the best way to ensure safety, which requires looking ahead. This leads us to prediction of motion in the motion control scheme.

Therefore, we will use Model Predictive Control (MPC), also known as receding horizon optimal control, to develop a motion control algorithm that follows a dynamic object with safety ensured. With the MPC framework, a constrained optimal control problem is solved at time step \( k \) to obtain a sequence of optimal control commands, i.e. \( u(k), \ldots, u(k+N) \), where \( N \) represents the prediction time horizon. Once the optimal control commands are computed, only the first command \( u(k) \) will be applied to mobile robot to generate a desired maneuver. Then, the above procedures are repeated at every time step afterwards.

In this paper, we formulate the constrained optimal control problem for the MPC-based motion controller by considering safe motion of mobile robot, tight path following, collision avoidance, and realistic accelerations, as follows.

\[
\begin{align*}
\min_U & \quad h(\{s(j)\}_{j=k_c}^{k_c+N}, U) \\
\text{subj. to} & \quad s(k + 1) = f(s(k), u(k)) \\
& \quad \psi^{lb} \leq \dot{\psi}(k_c + j) \leq \dot{\psi}^{ub} \\
& \quad a_x^{lb} \leq a_x(k_c + j) \leq a_x^{ub} \\
& \quad a_y^{lb} \leq a_y(k_c + j) \leq a_y^{ub} \\
& \quad \text{for } j = 0, 1, \cdots, N,
\end{align*}
\]

where \( k_c \) represents the time step at which the above problem is solved and \( N \) represents the time horizon for prediction. In (1), the constrained optimal control problem minimizes a cost function \( h(\cdot) \) in (1a) which is a function of a predicted sequence of system states \( \{s(j)\}_{j=k_c}^{k_c+N} \) and control inputs \( U = \{u(j)\}_{j=k_c}^{k_c+N} = [u(k_c), \ldots, u(k_c+N)] \).

The system, i.e., a mobile robot, state \( s(k) \) evolves based on the system dynamic equation \( f(\cdot) \) in (1b). Accelerations in linear and rotational motion, i.e., \( a_x(k) \) and \( a_y(k) \), are constrained by lower and upper bounds in (1c)-(1e), where the superscripts \( lb \) and \( ub \) represent lower bound and upper bound, respectively. As shown in Fig. 2, the subscripts \( x \) and \( y \) represent the longitudinal and lateral directions in the local frame of mobile robot, and \( \psi \) represents the rotational motion about \( z \) axis pointing toward readers, i.e. yaw motion. The global frame \( OXYZ \) in Fig. 2 is an inertial frame that is fixed with a constant orientation.

A. Mobile Robot System Dynamics

In this subsection, we define the state \( s \) for the mobile robot in (1b) that consists of the coordinate \( X \) and \( Y \) in the global frame, yaw/heading angle \( \psi \), longitudinal velocity \( v_x \), lateral velocity \( v_y \), and yaw rate \( \dot{\psi} \), i.e., \( s = \)
While following an object, it may lose contact with the ground and slide. Furthermore, if the mobile robot follows an object with packages loaded, it may cause the packages to fall when taking a sharp turn that generates a large lateral acceleration. Therefore, the lateral acceleration is included in the cost function for safe maneuvering of the mobile robot. With a nonholonomic mobile robot, we can obtain \( a_y \) from the fourth element in (2b) by substituting zero to \( \dot{v}_y \) as \( a_y = v_x \psi \).

To head for a goal, the distance to the goal \( d_g \) is included in the cost, as presented in (3c). Note that the distance to the goal at the end of the predicted trajectory is used, as shown in Fig. 3b, and the distance is not a straight line distance to account for the distance required to navigate without collision with obstacles.

The fourth term in (3d) contains the distance from the mobile robot to the planned path \( d_p \), as shown in Fig. 3c, and therefore, this cost term is utilized for tight path following. The path is planned by a path planning algorithm, e.g., A* algorithm [11], and the details are not discussed herein since it is beyond the scope of this paper. If a path is planned periodically to reflect a dynamic environment, the planned path can be used as guidance for navigation of mobile robot. Therefore, following a planned path tightly can be helpful for the mobile robot to go around obstacles.

### C. Accelerations to Track Constant Reference Velocities

In general, the decision variables in (1) would be the sequence of accelerations in linear and rotational motion, \( U = \{a_x(j)\}_{j=k_c}^{k_c+N}, \{a_y(j)\}_{j=k_c}^{k_c+N}, \{\psi(j)\}_{j=k_c}^{k_c+N}\} \), which is a vector in the dimension of \( 3(N+1) \). But, in this paper, we utilize the steady state reference tracking control [12] to define the accelerations such that the mobile robot can converge to the constant reference velocities, \( v_x^{ref}, v_y^{ref}, \) and \( \psi^{ref} \), over the prediction time horizon \( \tau_0 = N\tau \), as follows.

\[
\begin{align*}
\begin{bmatrix}
 a_x(k) \\
 a_y(k) \\
 \dot{\psi}(k)
\end{bmatrix} &=
\begin{bmatrix}
 K_x(v_x^{ref} - v_x(k)) \\
 K_y(v_y^{ref} - v_y(k)) \\
 K_\psi(\psi^{ref} - \psi(k))
\end{bmatrix},
\end{align*}
\]

where the feedback gains are

\[
K_i = \frac{1-p_i}{\tau}, \quad i \in \{x, y, \psi\}.
\]

With the above approach, the accelerations are constrained to functions of the constant reference velocities. In the definition of gain in (5), \( p_i \) represents the closed loop pole that determines the rate of convergence, and its absolute value should be \( |p_i| < 1 \) for stability. As the value of \( p_i \) becomes smaller, the velocity of the mobile robot converges to the reference one faster.

By substituting (4) and (5) into (1), the decision vector becomes \( U = [v_x^{ref}, v_y^{ref}, \psi^{ref}] \), which is in the 3-dimensional space, since \( v_x(k), v_y(k), \) and \( \psi(k) \) in (4) are obtained as the state variables. Therefore, the constrained optimal control problem in (1) can be solved with a significantly reduced computational cost.

If \( p_i \) is zero in (5), the mobile robot achieves the constant reference velocity immediately after one sampling time \( \tau \),
and the reference velocity is maintained for the entire prediction time horizon \( \tau_h \) since the accelerations become zero after one sampling time. Although the immediate convergence to the reference velocities with \( p_i = 0 \) is not realistic, it has been assumed previously in the existing motion control algorithms, e.g., Dynamic Window Approach (DWA) [8] and Trajectory Rollout (TR) [9]. However, this paper will use nonzero \( p_i \) in simulations and experiments.

With the accelerations constrained to converge to constant reference velocities, it is found out that the prediction time horizon provides a way to determine how quickly a mobile robot responds to motion of a dynamic object.

**Theorem 3.1:** Suppose an MPC defined in the form of (1) whose cost function includes the distance to goal at the end of predicted trajectory, \( d_g(k_c + N) \). If the linear accelerations, \( a_x \) and \( a_y \), are constrained to track constant reference velocities, \( v_x^{ref} \) and \( v_y^{ref} \), the time horizon for prediction, \( \tau_h = \tau N \), acts as the time constant to determine transient response.

**Proof:** The above statement can be proved by direct proof. Without loss of generality, consider a mobile robot heading for a static goal on a straight line in the X direction of the global frame, and it can generate the acceleration \( a_x(k) \) to converge the constant reference velocity \( v_x^{ref} \) immediately. In addition, cost terms other than the distance to a goal and acceleration constraints are excluded. Then, the MPC is formulated as follows.

\[
\begin{align*}
\min_{v_x^{ref}} & \quad d_g(k_c + N) \\
\text{subj. to} & \quad \begin{bmatrix} X(k + 1) \\ v_x(k + 1) \end{bmatrix} = \begin{bmatrix} X(k) + \tau v_x(k) \\ v_x(k) + \tau a_x(k) \end{bmatrix}, \\
& \quad a_x(k) = \frac{1}{\tau}(v_x^{ref} - v_x(k)).
\end{align*}
\]

If the above problem is solved at every time step with sampling time \( \tau \), we always obtain the optimal constant reference velocity \( v_x^{ref} \) such that \( v_x^{ref} \tau_h = d_g(k) \), where \( \tau_h = \tau N \) and \( d_g(k) \) is the distance to goal at the time \( k \) when the above problem is solved.

In the continuous time \( t \) domain, we can rewrite as follows.

\[
v_x^{ref} \tau_h = d_g(t),
\]

and since \( v_x^{ref} = \dot{X}(t) \),

\[
\dot{X}(t) = \frac{d_g(t)}{\tau_h} = \frac{X_g - X(t)}{\tau_h},
\]

where \( X(t) \) and \( X_g \) represent the coordinates of mobile robot and goal in the global frame, respectively.

Applying the Laplace Transform to (7) with the assumption of zero initial condition yields,

\[
sX(s) = \frac{1}{\tau_h} \frac{X_g}{s} - \frac{1}{\tau_h} X(s)
\]

\[
\Rightarrow \quad (s + \frac{1}{\tau_h})X(s) = \frac{1}{\tau_h} \frac{X_g}{s} \\
\Rightarrow \quad X(s) = \frac{X_g/\tau_h}{s(s + 1/\tau_h)}
\]

where the Laplace Transform of \( X(t) \) is represented by \( X(s) = \mathcal{L}\{X(t)\} \) and \( s \) represents the complex variable.

Then, the Inverse Laplace Transform of (8) yields

\[
X(t) = \frac{X_g(1 - e^{-t/\tau_h})}{\tau_h}.
\]

By the definition of time constant, the time at which \( X(t) \) reaches around \( (1 - 1/e) \times 63.2 \% \) of \( X_g \) is the prediction time horizon \( \tau_h \), as shown in (9).

**D. Solving the MPC**

In order to solve the MPC in (1), the method used in DWA and TR are applied in this paper. Therefore, \( n_i \) values are sampled from each decision variable, and a search space is constructed with all possible combinations of the sampled values. Then, by searching for the grid point in the search space that generates the smallest cost in (3), the optimal combination of decision variable values can be found.

In this paper, due to (4), the decision variables are \( U = [v_x^{ref}, v_y^{ref}, \psi^{ref}] \). Therefore, for example, we sample \( n_{ux} \) values spacing the range of \([v_x(k) - \tau_h a_x^{ub}, v_x(k) + \tau_h a_x^{ub}]\).
for the decision variable $v_x^{ref}$. In other words, the $i^{th}$ sample for $v_x^{ref}$ is

$$v_x = \tau_h a_x^b + \frac{\tau_h (a_x^b + a_y^b)}{n_{v_x}} i.$$  

Likewise, $n_{v_y}$ and $n_{\psi}$ values are sampled for the decision variables $v_y^{ref}$ and $\psi^{ref}$, respectively. If a nonholonomic mobile robot is used, we sample values from $v_x^{ref}$ and $\psi^{ref}$ only, and the optimal decision variables $v_x^*$ and $\psi^*$ are found by evaluating costs for the sampled values.

With the above approach, the MPC in (1) can be solved without an optimization solver. Most optimization solvers assume the cost function is differentiable, and therefore if we do not use an optimization solver, a non-differentiable cost function can be formulated. In addition, we can guarantee a global minima within the search space. If we define an MPC with many decision variables and many sample values are required for the decision variables, this approach is not feasible in real time and an optimization solver may be necessary. But, we have the 3 dimensional decision vector, and, in practice, with mobile robots, we do not have to search for $v_x^{ref}$ values at very fine resolutions, e.g., 0.01 m/s. Therefore, we can solve the MPC in (1) in real time with a reasonably small number of samples for the decision variables.

**IV. Simulation Results**

In this section, the proposed motion controller will be evaluated in simulation using a Robot Operating System (ROS) environment. The dynamic object following software stack is implemented as an extension of the ROS navigation stack [13] with the motion controller algorithm being replaced by the one proposed in this paper. As a result of using this framework, the Path Planner [14] in Fig. 4 plans a path from a mobile robot to a goal on 2-dimensional costmaps created by the costmap_2d ROS package [15] with a fast interpolated navigation function.

In the simulation, we use an ideal nonholonomic mobile robot platform on a 2-dimensional surface to verify performance of the proposed motion controller without extrinsic influences, e.g., roll motion and time lagging of actuator. Therefore, the 2-dimensional mobile robot platform is implemented as a ROS node, as shown in Fig. 4, based on the dynamic vehicle model in (2) with $v_y = 0$ and $v_y = 0$. The mobile robot in simulation takes in the optimal commands $v_x^*$ and $\psi^*$ computed by the proposed motion controller and yields the pose $(X, Y, \psi)$ in the global frame along with its velocity states $v_x$ and $\psi$. The states of mobile robot are updated every 0.05 seconds. The velocity states $v_x$ and $\psi$ are provided to the motion controller as feedback signals. A dynamic object is simulated through the Goal Publisher in Fig. 4 that publishes goals moving with specified velocities every 0.1 seconds. Note that a localization algorithm does not run in the proposed following software stack since a prior map is not necessary for following a dynamic object in this paper. The parameters for the motion controller are listed in Table I, and the motion control algorithm computes the control commands at an update frequency 5 Hz.

**TABLE I: Motion Controller Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_x$</td>
<td>0.2</td>
<td>$\omega_y$</td>
<td>0.0001</td>
</tr>
<tr>
<td>$p_y$</td>
<td>0.2</td>
<td>$\omega_y$</td>
<td>1.0</td>
</tr>
<tr>
<td>$\tau$</td>
<td>0.1 [s]</td>
<td>$\omega_y$</td>
<td>0.001</td>
</tr>
<tr>
<td>$\psi^{ref}$</td>
<td>4.5 [rad/s]</td>
<td>$\omega_y$</td>
<td>0.01</td>
</tr>
<tr>
<td>$\psi^{ref}$</td>
<td>-4.5 [rad/s]</td>
<td>$n_{v_y}$</td>
<td>60</td>
</tr>
<tr>
<td>$a_x^{ref}$</td>
<td>0.5 [m/s$^2$]</td>
<td>$n_{v_x}$</td>
<td>30</td>
</tr>
</tbody>
</table>

**A. Prediction Time Horizon as Time Constant**

As discussed previously, with the proposed motion control formulation, the prediction time horizon $\tau_h$ acts as the time constant, and therefore we can tune $\tau_h$ to adjust the rate of convergence to a goal, as shown in Fig. 5. In this simulation,

![Fig. 5: Different rate of convergence to a goal due to different prediction time horizon](image)

the goal was static and located at 3 m in front of the mobile robot for the four different $\tau_h$ values, 1.5, 2.0, 2.5, and 3.0 seconds.

As shown in Fig. 5, with the smaller $\tau_h$, the mobile robot accelerates $v_x$ harder to reach the goal faster. The time steps when the mobile robot navigates 63.2% of the distance to goal are around 0.9, 1.2, 1.7, and 2.0 seconds, respectively. The values of $\tau_h$ do not match the measured time steps exactly, but this simulation demonstrates that $\tau_h$ can be used to adjust the rate of convergence to a goal. In the simulation presented in Fig. 6, the mobile robot follows a moving object with two different prediction time
horizons $\tau_h$. The positions of the moving object are published as goals in the ROS environment. The speed of the object is 0.0 m/s at first, and it reaches 1.0 m/s immediately after around 2.0 seconds has passed. This abrupt speed change simulates following a highly dynamic object, e.g., a human.

In Fig. 6, the mobile robot with $\tau_h = 2.5$ falls behind the one with $\tau_h = 1.5$ at the end of simulation. This is because the velocity $v_x$ of the mobile robot increases faster to catch up with the dynamic goal more quickly with $\tau_h = 1.5$ than $\tau_h = 2.5$, as shown in Fig. 7.

B. Tight Path Following

The proposed motion controller takes tight path following into consideration through the cost term in (3d). The tightness of the path following can be adjusted by tuning the weight $\omega_p$. Fig. 8 presents the trajectories of the mobile robot and goal of a simulation in which the mobile robot moves straight first and starts taking left turn to follow the dynamic goal with $v_x = 2.0$ m/s and $\dot{\psi} = 0.7$ rad/s. In order to verify the performance of tight path following, the safe maneuver cost term in (3b) is set to zero. In Fig. 8, the orange line represents the last path in the simulation with $\tau_h = 2.5$

C. Safe Maneuver

The lateral acceleration $a_y$ is included as a cost term in (3b) to generate safe maneuvers for the mobile robot while taking turns. In the simulation with $\tau_h = 2.5$ in Fig. 9, the mobile robot undergoes the same scenario as in Fig. 8. The tight path following cost term is used, and therefore the mobile robot tries to create a sharper turn than the dynamic object.
V. EXPERIMENTAL RESULTS

In the simulation, the proposed motion controller is implemented in a ROS framework and the mobile robot is modeled as a separate ROS node. Therefore, experimental tests can be conducted by simply replacing the Mobile Robot in Fig. 4 with a real mobile robot platform.

A. Mobile Robot Platform

The test robot is built on the Clearpath Jackal differential-drive wheeled platform shown in Fig. 11. The base platform receives linear and angular velocity commands and provides odometry data from wheel encoders. The platform is outfitted with additional Lidar sensors. An RPLidar A3 2D Lidar is mounted on the front bumper and is used for obstacle detection. A Velodyne VLP-32C 3D Lidar is mounted on top and is used for target tracking.

B. Human Tracking and Goal Broadcasting

In the experimental tests, a human tracking algorithm replaces the Goal Publisher in Fig. 4 to broadcast goals for the proposed motion controller. Human tracking is performed using the Leg Tracker ROS Package that is based on the work in [7]. A 2D laserscan is generated from the Velodyne and passed to the leg tracker ROS node. The leg tracker maintains unique labels for each human around the robot. The robot can be assigned a specific label ID to follow as its target. A direction vector is computed between the 2D pose of the target and the 2D pose of the robot within the global frame, i.e., the odometry frame in the ROS navigation stack. A goal is assigned as the location along that vector that is offset 1 m behind the target. Goals are repeatedly generated and provided to the motion controller at a rate of 7.5 Hz.

C. Following Performance

Tests with the real mobile robot platform also utilize the same control update frequency 5 Hz and parameters listed in Table I as in the Simulation Results except for $\tau = 0.2$ second, and they demonstrate the finding that the prediction time horizon acts as the time constant. In Fig. 12, the percentage of travel distance is presented when the mobile robot is commanded to navigate to a static goal in front of it. The time steps when the mobile robot navigates 63.2% of the distance are around 1.7, 2.5, and 3.2 seconds for $\tau_h = 2.5$, $\tau_h = 3.5$, and $\tau_h = 4.5$, respectively. This implies that the prediction time horizon has a linear relationship with the time constant.

Fig. 11: Test robot consisting of a Clearpath Jackal base with an RPLidar A3 on the front bumper and a Velodyne VLP-32C 3D Lidar on top.

Fig. 12: Different rate of convergence to a goal due to different prediction time horizon.

To validate the collision avoidance and following performance, the mobile robot was assigned to follow a person who is going over an obstacle, performing a zigzag maneuver, and turning around. In this test, the motion controller computed the following control commands with the prediction time horizon $\tau_h = 2.5$. As shown in the red box in Fig. 13a, the person walked in a straight-line path stepping over an obstacle while the mobile robot deviates from the person’s trajectory to go around that obstacle. The gray dots in Fig. 13a and Fig. 13b represent occupied grid points in the costmap used in motion control. After avoiding collision with the obstacle, the mobile robot follows the person walking in a zigzag motion. In Fig. 13b, the mobile robot follows the person turning around in a tight space.
A motion controller to follow a dynamic object based on MPC. It is proved that the prediction time horizon acts as a time constant to determine the transient response in the proposed MPC formulation. Therefore, the proposed motion control algorithm can adapt to the dynamics of object by tuning the prediction time horizon. It also takes safe maneuvering and tight path following into consideration. The motion controller is formulated in a way that is simple enough to compute control commands in real time without an optimization solver but can still achieve the objectives successfully. The proposed motion controller are demonstrated through extensive simulation and experiments of a test mobile robot.

VI. CONCLUSION

This paper proposes a motion controller to follow a dynamic object based on MPC. It is proved that the prediction time horizon acts as a time constant to determine the transient response in the proposed MPC formulation. Therefore, the proposed motion control algorithm can adapt to the dynamics of object by tuning the prediction time horizon. It also takes safe maneuvering and tight path following into consideration. The motion controller is formulated in a way that is simple enough to compute control commands in real time without an optimization solver but can still achieve the objectives successfully. The proposed motion controller are demonstrated through extensive simulation and experiments of a test mobile robot.

REFERENCES


Fig. 13: Obstacle Avoidance and Following A Person

In addition, since the mobile robot takes safe maneuvers into consideration, it slows down if a sharp turn is required to follow the person, as shown in Fig. 14. The longitudinal velocity decreases when the magnitude of yaw rate increases significantly in the red boxes.

Fig. 14: Oblique velocity and yaw rate generated when a sharp turn is required to follow a person