IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 5, 2020, accepted January 17, 2020, date of publication January 30, 2020, date of current version February 11, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2970468

A sEMG-Based Shared Control System With No-Target Obstacle Avoidance for Omnidirectional Mobile Robots

HAIYI KONG[1], CHENGUANG YANG[2], (Senior Member, IEEE), GUANG LI[3], (Member, IEEE), AND SHI-LU DAI[1], (Member, IEEE)

[1]School of Automation Science and Engineering, South China University of Technology, Guangzhou 510641, China
[2]Bristol Robotics Laboratory, University of the West of England, Bristol BS16 1QY, U.K.
[3]School of Engineering and Materials Science, Queen Mary University of London, London E1 4NS, U.K.

Corresponding author: Chenguang Yang (cyang@ieee.org)

This work was supported in part by the Foshan Science and Technology Innovation Team Special Project under Grant 2018IT100322, and in part by the National Nature Science Foundation (NSFC) under Grant 61473120.

ABSTRACT We propose a novel shared control strategy for mobile robots in a human-robot interaction manner based on surface eletromyography (sEMG) signals. For security reasons, an obstacle avoidance scheme is introduced to the shared control system as collision avoidance guidance. The motion of the mobile robot is a resultant of compliant motion control and obstacle avoidance. In the mode of compliant motion, the sEMG signals obtained from the operator's forearms are transformed into human commands to control the moving direction and linear velocity of the mobile robot, respectively. When the mobile robot is blocked by obstacles, the motion mode is converted into obstacle avoidance. Aimed at the obstacle avoidance problem without a specific target, we develop a no-target Bug (NT-Bug) algorithm to guide the mobile robot to avoid obstacles and return to the command line. Besides, the command moving direction given by the operator is taken into consideration in the obstacle avoidance process to plan a smoother and safer path for the mobile robot. A model predictive controller is exploited to minimize the tracking errors. Experiments have been implemented to demonstrate the effectiveness of the proposed shared control strategy and the NT-Bug algorithm.

INDEX TERMS Shared control system, surface electromyography, no-target Bug algorithm, model predictive control.

I. INTRODUCTION

Nowadays, the concept of human-robot shared control has been involved in many research fields, such as medical rehabilitation, autonomous vehicles and swarm intelligence control. The shared control strategy combines the autonomous control of robots with the intelligent decision-making of operators, which greatly improves the operation ability of the system [1]. In [2], the operator controls the movement of a walking-assistant robot by exerting force on it. In [3], real-time activity parameters of the driver were introduced into the lane keeping assist system to realise the shared lateral control of the intelligent vehicle. In [4], skilled operators were integrated into the control loop of unmanned aerial vehicles (UAVs) for autonomy sharing between humans and UAVs.

In human-robot interaction, human biological and psychophysical signals often act as an expression of human intent. In order to capture these human signals, a great number of sensors have been developed. In general, they can be divided into noncontact sensors and contact sensors according to the way of signal catching [5]. Noncontact sensors are nonintrusive and collect psychophysical signals mainly based on visual or audio manners. Contact sensors are nonintrusive or intrusive. They capture physiological signals through the contact with human bodies [6]. Compared with psychophysical signals, physiological signals have better inference power to infer an operator's state [7].

Electromyography (EMG) is one of the most widely-used physiological signals. Normally, they are collected by surface

The associate editor coordinating the review of this manuscript and approving it for publication was Liang Ding.

EMG (sEMG) sensors, such as data gloves and armbands, which are noninvasive and can provide the operator with a better operating experience than intramuscular EMG sensors [8]. As the sEMG signals are highly associated with the activation level of muscles being measured, they can be used for the estimation of muscle stiffness and gestures recognition [9], [10]. With the maturity of the gesture recognition technology, gesture control has become a popular way of enabling the operator to interact naturally with robots. Moreover, muscle stiffness can act as a human command for the motion control of robots in a shared control system [11].

It is necessary to incorporate the function of autonomous obstacle avoidance for the safety of both operators and robots in a robot system [12]–[14]. The robot should be able to move in compliance with the operator's command while avoid colliding with any obstacles. Many path planning methods have been successfully applied to systems as collision-free guidance, e.g., A-star [15] and Rapidly-Exploring Random Tree (RRT) [16]. However, their performance depends on the environment map. The amount of calculation increases dramatically as the map expands. Bug is a dynamic path planning method, which is suitable for obstacle avoidance with limited information [17]. It guides the robot to walk along the boundary of obstacles to avoid collisions and takes the original moving direction as a determinant of leaving obstacles. Artificial potential field (APF) method has been extensively utilised for local path planning owing to its simple principle and low-time consumption [18]. However, it requires a specific target to calculate the attractive force. When the human's commands given to robots are moving direction and linear velocity rather than an exact coordinate of the target in the global coordinate system, most existing path planning methods can not work. Inspired by the Bug algorithm, we develop a no-target Bug (NT-Bug) algorithm, which can guide the mobile robot to avoid both static and dynamic obstacles in the case of no specific target. As the Bug algorithm does not rely on the prior information of the environment, in terms of complicated obstacles, the path generated is inevitably zigzag, which is difficult to track in practice. Therefore, we introduce the moving direction given by the operator into the obstacle avoidance process to plan a smoother and safer path for the mobile robot.

Trajectory tracking is a crucial part of the mobile robot system. The objective of tracking control is to make the robot movement consistent with the reference trajectory in the required posture. In order to optimize the tracking performance, a lot of efforts have been made to improve the controller. In [19], an integral sliding controller was proposed to deal with the problem of steady state error. In [20], targeted at the uncertain nonholonomic mobile robot, a simple adaptive controller incorporating actuator dynamics was developed. In [21], nonlinear model predictive control (MPC) was employed for the steering rate control of the omnidirectional robot. MPC, also known as moving horizon optimal control, is suitable for both linear systems and nonlinear systems. It uses an explicit model to predict the future plant

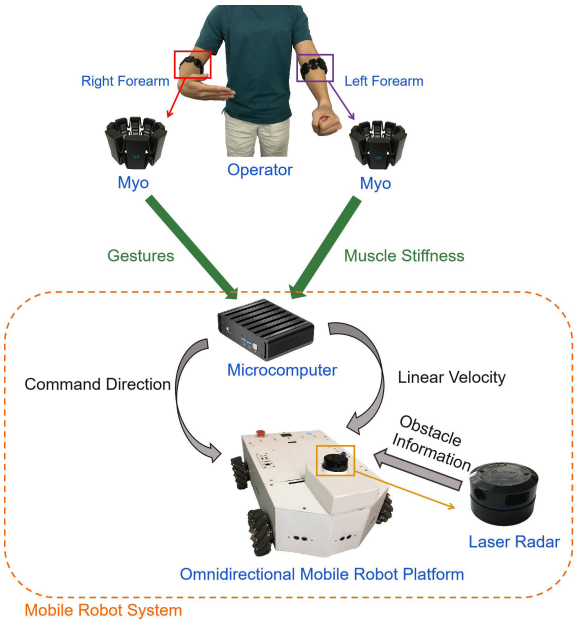


FIGURE 1. The shared control system architecture.

behaviors and selects the optimal control sequence based on an objective function [22]. Because of the outstanding capability of dealing with control problems with input and state constraints, MPC has been widely applied to robot systems for trajectory prediction and tracking control [23].

In this paper, we propose a shared control strategy in which the operator can control the motion of the mobile robot using the sEMG signals. For security consideration, the mobile robot is designed to avoid obstacles autonomously. Therefore, the motion of the mobile robot is a resultant of compliant motion control and autonomous obstacle avoidance. Firstly, the sEMG signals are collected from the operator for gesture recognition and muscle stiffness extraction, respectively. Then, the gesture recognition result is converted into the moving direction command for the mobile robot, while the muscle stiffness is used to control the linear velocity. Aimed at the real-time obstacle avoidance problem without a specific target, a novel NT-Bug algorithm is proposed, which can guide the mobile robot to avoid both static and dynamic obstacles and return to the `command line`. Command moving direction given by the operator is introduced to the obstacle avoidance process to plan a smoother and safer path for the mobile robot. In order to minimize the tracking errors, MPC is exploited to the system for tracking control of the reference trajectory.

II. PRELIMINARIES

A. DESCRIPTION OF THE SHARED CONTROL SYSTEM

The shared control system consists of an operator, two Myo armbands and an omnidirectional mobile robot. The whole system architecture is shown in Fig. 1.

There are 8 EMG electrodes embedded in a Myo armband. We use Myo armbands to capture the sEMG signals from the

Gesture	Action	Command Direction
First	Move Forward	0
Wave Left	Move Left	π/2
Fingers Spread	Move Backward	π
Wave Right	Move Right	3π/2
Double Tap	Stop	0

FIGURE 2. Relationship between gestures and command directions.

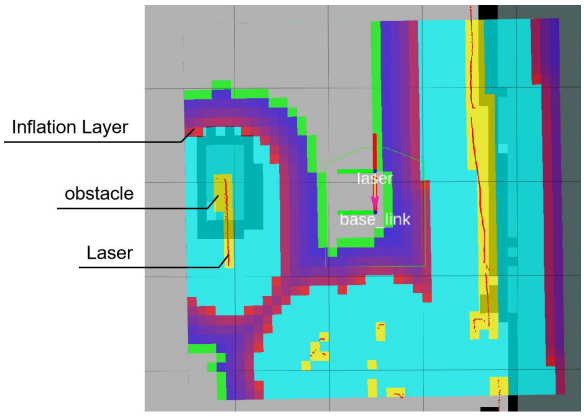


FIGURE 3. Local cost map.

operator at a sampling frequency of 200 Hz. The captured sEMG signals are sent to the client via bluetooth.

The omnidirectional mobile robot has four mecanum wheels which are symmetrically distributed on both sides of it. The capability of moving in arbitrary direction with any orientation allows it to move as the operator's command [24]- [26]. A laser radar is installed in the front of the mobile robot platform to detect obstacles in the environment.

In the control process, the operator is required to wear a Myo on both forearms. The sEMG signals from the left forearm are used to extract muscle stiffness, while those from the right forearm are used for gesture recognition. Then, the muscle stiffness is converted into a linear velocity command for the mobile robot, while the result of gesture recognition is transformed into the moving direction of command motion. The Myo device initially supports recognition of five pre-set gestures: `Fist`, `Fingers Spread`, `Wave Left`, `Wave Right`, `Double Tap`, which are used to command the mobile robot to move forward, backward, left, right and stop, respectively. We use `command direction` to represent the angle between the direction vector of command motion and the X-axis of the global coordinate system and `command line` to represent the straight line along the `command direction`. The relationship between gestures and `command direction` is shown in Fig. 2.

The system consists of two modes of motion: compliant motion and obstacle avoidance. In the mode of compliant motion, the mobile robot moves in the `command direction` given by the operator. Once the distance between the closest obstacle and the mobile robot is smaller than the minimum safety distance, the motion mode would be switched to autonomous obstacle avoidance. Then, the mobile robot moves along the outline of the obstacle to avoid collisions under the control of the NT-Bug algorithm. When the mobile robot safely bypasses the obstacle and returns to the `command line`, the motion mode would be turned to compliant motion and the mobile robot keeps moving in the `command direction`. The linear velocity of the mobile robot is completely controlled by the operator and varies in accordance with muscle stiffness.

B. OBSTACLE DETECTION

In actual application, we need to be aware of the distance between the mobile robot and obstacles in real time to take actions to avoid collisions. Based on the `Costmap_2d`, we establish a dynamic local cost map centered on the mobile robot using the data from the laser radar to obtain the obstacle information, which is shown in Fig. 3.

`Costmap_2d` is an open-source algorithm package in the robot operating system (ROS), which can automatically produce a 2D cost map based on the data from sensors such as cameras and radars [27]. In the cost map, the environment is divided into a series of grid cells and each of them has a cost value which represents the potential of collisions. Considering the size of the mobile robot, the original cost map is usually processed by inflation algorithm, which enlarges the influence area of obstacles.

Since the grid cells of obstacles have the largest cost value, we can seek out the grid cells that belong to obstacles according to the cost value and obtain their coordinates relative to the mobile robot. Through calculating the Euclidean distance between the robot center and the grid cells of obstacles, we can obtain the minimum Euclidean distance as the closest obstacle distance.

C. MUSCLE STIFFNESS EXTRACTION BASED ON sEMG SIGNALS

The sEMG signals from the left forearm of the operator are use to estimate the muscle stiffness. As a Myo has 8 EMG sensors, it can collect sEMG signals from 8 channels at a time. Adding the amplitude of the sEMG signals from 8 channels:

$$A_{emg}(k) = \sum_{i=1}^{N} \mid E_i(k) \mid, \tag{1}$$

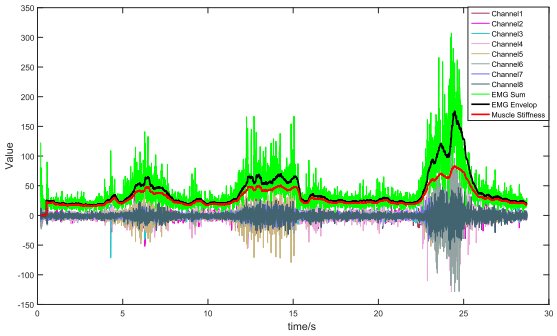


FIGURE 4. The extraction of muscle stiffness ($w = 50$, $\alpha = -0.01$).

where $N = 8$ represents the number of channels, k represents the current sampling instant, E_i is the sEMG signal from the ith channel. Then, a moving average filter is applied to acquire the sEMG envelop at the sampling instant k:

$$\epsilon_{emg}(k) = \frac{1}{w}\sum_{i=0}^{w} A_{emg}(k-i), \tag{2}$$

where w is defined as the window size. According to [28], the muscle stiffness can be extracted by:

$$\varsigma_{emg}(k) = \frac{e^{\alpha\epsilon_{emg}(k)} - 1}{e^{\alpha} - 1}, \tag{3}$$

where e^x is the exponential function, α is a nonlinear shape factor. The extraction of muscle stiffness is shown in Fig. 4.

D. LINEAR VELOCITY

The linear velocity of the mobile robot is controlled by the operator. In general, we regard the relationship between force and speed as linear, which is related to our living experience. For example, the harder you tramp the accelerator of a car, the faster the car will be. In view of the operation habit of humans, we define the linear velocity of the mobile robot is proportional to muscle stiffness, that is, as muscle stiffness increases, the linear velocity increases accordingly. Firstly, normalize the muscle stiffness:

$$\varrho_{emg}(k) = \frac{\varsigma_{emg}(k) - \varsigma_{min}}{\varsigma_{max} - \varsigma_{min}}, \tag{4}$$

where $\varrho_{emg}(k)$ can be seen as a control gain, ς_{min} and ς_{max} are the minimum and maximum magnitude of muscle stiffness, respectively.

Then, according to the mapping relationship between muscle stiffness and the linear velocity of the mobile robot, we have:

$$v_{linear}(k) = (v_{max} - v_{min})\varrho_{emg}(k) + v_{min}, \tag{5}$$

where v_{min} and v_{max} are the minimum and maximum linear velocity of the mobile robot, respectively.

III. NO TARGET BUG ALGORITHM

For a mobile robot system, it is necessary to realise autonomous obstacle avoidance to guarantee the safety of the mobile robot and the operator. Ordinarily, the obstacle avoidance path is generated by a path planner based on obstacle information and a global or local target which is given directly by the operator or chosen from the static path generated in advance. However, in our shared control system, this specific target does not exist. The motion control of the mobile robot is realized by controlling its moving direction and linear velocity in real time. In other words, the operator will tell the mobile robot "where to go" and "how fast to run", but will not tell it "where to stop". The mobile robot will keep moving in the same direction until it receives a new direction command or is blocked by obstacles. Due to the particularity of the condition, most existing path planning methods can not work. Aimed at the problem of real-time obstacle avoidance without a target, we propose a novel NT-Bug path planning method based on the traditional Bug algorithm to guide the mobile robot to avoid obstacles. As the NT-Bug algorithm only relies on the obstacle information obtained by the sensor rather than the prior information of the environment, it is suitable for static or dynamic obstacle avoidance.

When the mobile robot receives a new moving direction command from the operator, the `command line` is updated according to the position of the mobile robot and the `command direction` at the current sampling instant. The command v_x and v_y velocities are the velocity component of linear velocity along the X-axis and Y-axis of the global coordinate system. The function of the NT-Bug algorithm is to guide the mobile robot to avoid obstacles and return to the `command line`. At every sampling time, the NT-Bug algorithm calculates the moving direction for the mobile robot.

To be clear, the following notions are introduced: The center of the mobile robot is denoted as O^m; the closest obstacle cell is denoted as B; $\ell(O^m, B)$ represents the closest obstacle distance; ℓ_{safe} is the minimum safety distance related to the size of the mobile robot; referring to the traditional potential field method, we consider the obstacle as the virtual repulsive potential field and define θ_{obs} as the direction of the force exerted by B on the mobile robot; θ represents the moving direction of the mobile robot; θ_{mode} represents the `command direction`; θ_{esc} is a once-and-for-all decided upon direction of passing around an obstacle, which is either $\pi/2$ or $-\pi/2$; d represents the distance from O^m to the `command line`; d_σ is a distance threshold; *pattern* is used to represent the different stage of the NT-Bug algorithm. Specially, *patter* $= 0$ represents the motion mode of compliant control.

A. NO-TARGET BUG1

The pseudo-code of the NT-Bug1 algorithm is presented in Algorithm 1. Once the closest obstacle distance $\ell(O^m, B)$ is smaller than the minimum safe distance ℓ_{safe}, the motion mode would be converted to obstacle avoidance. Inspired by the Bug algorithm, we choose the way of moving around the outline of the obstacles to prevent collisions in the NT-Bug1. Therefore, the moving direction of the mobile robot can be

Algorithm 1 No-Target Bug1

Input: $\theta_{esc}, \theta_{obs}, \ell_{obs}$, d
Output: θ
  if $\ell_{obs} < \ell_{safe}$ and $pattern = 0$ **then**
    $pattern \leftarrow 1$;
    $\theta \leftarrow \theta_{obs} + \theta_{esc}$;
  else if $pattern = 1$ **then**
    if $\mathrm{d} < \mathrm{d}_\sigma$ **then**
      $pattern \leftarrow 0$;
      break;
    else
      $\theta \leftarrow \theta_{obs} + \theta_{esc}$;
    end if
  end if

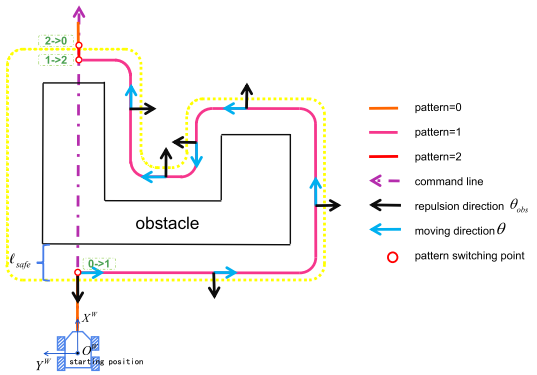


FIGURE 5. The procedure of passing a concave obstacle in NT-Bug1 algorithm ($\theta_{esc} = \pi/2$).

obtained by:

$$\theta = \theta_{obs} + \theta_{esc}. \tag{6}$$

We set $pattern = 1$ to represent the stage of moving parallel to the outline of the obstacle. The mobile robot would keep moving in this way until the condition $\mathrm{d} < \mathrm{d}_\sigma$ is satisfied, which means the mobile robot has already bypassed the obstacle and reached the `command line`. d_σ is small enough to meet the accuracy requirement of the system. Then, the motion mode is turned to compliant control, and the mobile robot keeps moving along the `command line`. The procedure of passing a concave obstacle in NT-Bug1 algorithm is illustrated in Fig. 5

B. NO-TARGET BUG2

The NT-Bug1 algorithm can successfully guide the mobile robot to avoid obstacles and return to the `command line`. However, as it is not sensitive to the shape of obstacles, the produced path could become tortuous referring to complicated obstacles, which is difficult to track in practice. In view of the application scenarios are often indoor workplaces where the obstacles are mostly irregularly convex or concave, we introduce the `command direction` into the obstacle avoidance process to plan a smoother and safer path.

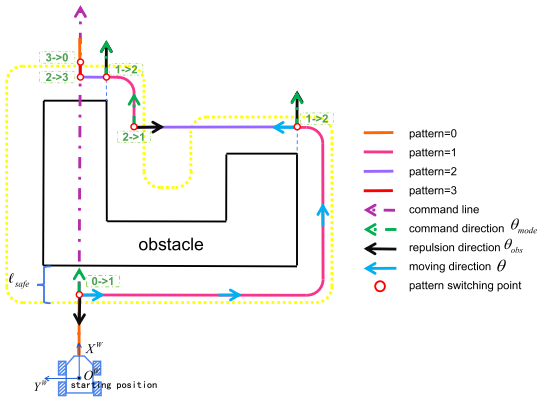


FIGURE 6. The procedure of passing a concave obstacle in NT-Bug2 algorithm ($\theta_{esc} = \pi/2$).

Algorithm 2 No-Target Bug2

Input: $\theta_{mode}, \theta_{esc}, \theta_{obs}, \ell_{obs}$, d
Output: θ
  if $\ell_{obs} < \ell_{safe}$ and $pattern = 0$ **then**
    $pattern \leftarrow 1$;
  else if $pattern = 1$ **then**
    if $\theta_{obs} = \theta_{mode}$ **then**
      $pattern \leftarrow 2$;
    else if $\mathrm{d} < \mathrm{d}_\sigma$ **then**
      $pattern \leftarrow 3$;
    end if
  else if $pattern = 2$ **then**
    if $|\theta_{obs} - (\theta_{mode} + \theta_{esc})| > \frac{\pi}{2}$ **then**
      $pattern \leftarrow 1$;
    else if $\mathrm{d} < \mathrm{d}_\sigma$ **then**
      $pattern \leftarrow 3$;
    end if
  else if $pattern = 3$ **then**
    if $\ell_{obs} > \ell_{safe}$ **then**
      $pattern \leftarrow 0$;
      break;
    else if $|\theta_{obs} - \theta_{mode}| > \frac{\pi}{2}$ **then**
      $pattern \leftarrow 1$;
    end if
  end if
  if $pattern = 1$ **then**
    $\theta \leftarrow \theta_{obs} + \theta_{esc}$;
  else if $pattern = 2$ **then**
    $\theta \leftarrow \theta_{mode} + \theta_{esc}$;
  else if $pattern = 3$ **then**
    $\theta \leftarrow \theta_{mode}$;
  end if

The pseudo-code of the NT-Bug2 algorithm is presented in Algorithm 2 and the procedure of passing a concave obstacle in NT-Bug2 algorithm is illustrated in Fig. 6. As what mentioned before, in the stage of $pattern = 1$, the mobile robot moves parallel to the outline of obstacles to prevent collisions.

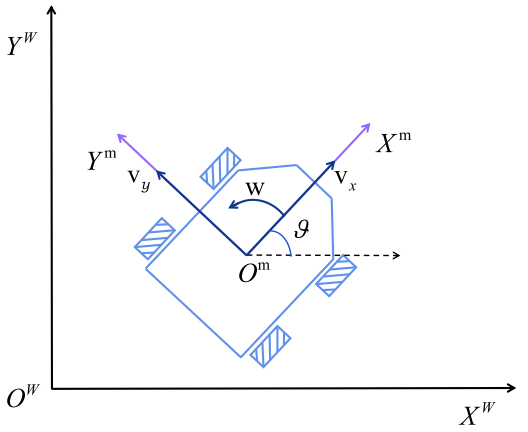


FIGURE 7. Coordinate systems.

pattern 1 → 2: The condition $\theta_{obs} = \theta_{mode}$ is satisfied, which means the obstacles could not affect the movement of the mobile robot along the `command direction`. The mobile robot would keep moving in the current direction to approach the `command line`.

pattern 1 → 3/2 → 3: The mobile robot reaches the `command line` and the closest obstacle will not impede its movement, whereas it is still within the distance ℓ_{safe}. It will move in the `command direction` to be away from the obstacle.

pattern 2 → 1/3 → 1: The mobile robot is blocked by another obstacle, it would move around the outline of the obstacle to avoid it.

pattern 3 → 0: The mobile robot has successfully bypassed the obstacle and moves along the `command line`. The control mode is converted to compliant control.

In the NT-Bug2, the `command direction` is regarded as a judgment to estimate the threat level of obstacles to the robot movement. As the condition $\theta_{obs} = \theta_{mode}$ is satisfied, the repulsion force of the obstacle is consistent with the `command direction`, which means the obstacle would not block the robot movement along the `command line`. Therefore, the mobile robot can stop walking around obstacles and move towards the command line directly. In this way, the length of useless path can be effectively shorten, and the path is smoother and easier to track for the mobile robot.

IV. MOTION CONTROL OF THE MOBILE ROBOT

A. DYNAMIC ERROR MODEL

For the sake of simplify, regarding the mobile robot as a mass point. Define the global coordinate system as $\mathrm{X}^W\mathrm{O}^W\mathrm{Y}^W$ and the robot coordinate system as $\mathrm{X}^m\mathrm{O}^m\mathrm{Y}^m$ (Fig. 7). Introduce the transformation matrix between the global coordinate system and the robot coordinate system:

$$\mathfrak{R}(\vartheta) = \begin{bmatrix} \cos\vartheta & \sin\vartheta & 0 \\ -\sin\vartheta & \cos\vartheta & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{7}$$

where ϑ is a rotation angle.

The kinematic model of the mobile robot can be expressed as

$$\dot{\mathbf{X}} = \begin{bmatrix} \dot{\mathrm{x}} \\ \dot{\mathrm{y}} \\ \dot{\vartheta} \end{bmatrix} = \mathfrak{R}^{-1}(\vartheta) \begin{bmatrix} \mathrm{v}_x \\ \mathrm{v}_y \\ \mathrm{w} \end{bmatrix}, \tag{8}$$

where $\mathbf{X} = [\mathrm{x}, \mathrm{y}, \vartheta]^T$ represents the current posture of the mobile robot in $\mathrm{X}^W\mathrm{O}^W\mathrm{Y}^W$, w is the angular velocity, and v_x and v_y are the velocity components along the X^m-axis and Y^m-axis, respectively. In the same way, we can obtain the kinematic model of the reference posture $\mathbf{X}_r = [\mathrm{x}_r, \mathrm{y}_r, \vartheta_r]^T$. Then, the state error between the current posture and the reference posture of the mobile robot can be defined as

$$\mathbf{X}_e = \begin{bmatrix} \mathrm{x}_e \\ \mathrm{y}_e \\ \vartheta_e \end{bmatrix} = \mathfrak{R}(\vartheta) \begin{bmatrix} \mathrm{x}_r - \mathrm{x} \\ \mathrm{y}_r - \mathrm{y} \\ \vartheta_r - \vartheta \end{bmatrix}. \tag{9}$$

By taking the derivative of (9), we can obtain the dynamic error model:

$$\dot{\mathbf{X}}_e = \begin{bmatrix} \dot{\mathrm{x}}_e \\ \dot{\mathrm{y}}_e \\ \dot{\vartheta}_e \end{bmatrix} = \begin{bmatrix} -\mathrm{v}_x + \mathrm{w}\mathrm{y}_e + \mathrm{v}_{xr}\cos\vartheta_e - \mathrm{v}_{yr}\sin\vartheta_e \\ -\mathrm{v}_y - \mathrm{w}\mathrm{x}_e + \mathrm{v}_{xr}\sin\vartheta_e + \mathrm{v}_{yr}\cos\vartheta_e \\ \mathrm{w}_r - \mathrm{w} \end{bmatrix}, \tag{10}$$

where $[\mathrm{v}_{xr}, \mathrm{v}_{yr}, \mathrm{w}_r]^T$ is the reference velocity. Define $\mathbf{u} = \begin{bmatrix} \mathrm{v}_{xr}\cos\vartheta_e - \mathrm{v}_x \\ \mathrm{v}_{yr}\cos\vartheta_e - \mathrm{v}_y \\ \mathrm{w}_r - \mathrm{w} \end{bmatrix}$ and linearize (11) at the equilibrium point ($\mathbf{X}_e = 0, \mathbf{u} = 0$). Then, we can acquire the linearized dynamic error model:

$$\dot{\mathbf{X}}_e = \begin{bmatrix} \dot{\mathrm{x}}_e \\ \dot{\mathrm{y}}_e \\ \dot{\vartheta}_e \end{bmatrix} = \mathbf{u} + \begin{bmatrix} 0 & \mathrm{w}_r & -\mathrm{v}_{yr} \\ -\mathrm{w}_r & 0 & \mathrm{v}_{xr} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathrm{x}_e \\ \mathrm{y}_e \\ \vartheta_e \end{bmatrix}. \tag{11}$$

B. MODEL PREDICTIVE CONTROLLER

The task of trajectory tracking is to steer the mobile robot to track the reference trajectory, that is, to find an optimal control input $\mathbf{u}$, so that:

$$\lim_{t\to\infty} |\mathbf{X}_r - \mathbf{X}| < \psi, \tag{12}$$

where ψ is a small neighborhood containing the origin.

In this system, the reference trajectory consists of two parts: command trajectory and obstacle avoidance trajectory. With regard to the former, the moving direction of the mobile robot is given by the operator, while it is calculated by the NT-Bug algorithm in the latter. According to the dead reckoning algorithm, we can deduce the reference trajectory based on the moving direction and the linear velocity. At every sample time, the reference posture can be obtained by:

$$\begin{aligned} \mathrm{x}_r(k+1) &= \mathrm{x}_r(k) + \Delta\mathrm{x} = \mathrm{x}_r(k) + \mathcal{T}\mathrm{v}_{xr} \\ &= \mathrm{x}_r(k) + \mathcal{T}v_{linear}\cos\theta, \end{aligned} \tag{13}$$

$$\begin{aligned} \mathrm{y}_r(k+1) &= \mathrm{y}_r(k) + \Delta\mathrm{y} = \mathrm{y}_r(k) + \mathcal{T}\mathrm{v}_{yr} \\ &= \mathrm{y}_r(k) + \mathcal{T}v_{linear}\sin\theta, \end{aligned} \tag{14}$$

$$\vartheta_r(k+1) = \vartheta_r(k) + \Delta\vartheta = \vartheta_r(k) + \mathcal{T}\mathrm{w}, \tag{15}$$

where $\mathbf{X}_r(0) = [\mathrm{x}_r(0), \mathrm{y}_r(0), \vartheta_r(0)]^T = \mathbf{X}(0)$, $\Delta\mathrm{x}$ and $\Delta\mathrm{y}$ represent the displacement increment along X^W- and Y^W-axis, $\Delta\vartheta$ represents the angle increment, and $\mathcal{T}$ is the sampling period.

Remark 1: In our design, the operator can use gestures to control the mobile robot to move forward, backward, left and right. Due to the limited number of recognizable gestures, the rotary movement of the mobile robot is not involved. Therefore, the angle increment in (15) is zero.

MPC is used for tracking control in the system. By transferring the control problem into a minimization problem of the cost function and calculating it online, a control sequence can be acquired. The first control action of the control sequence is the optimal control input applied to the system.

Rewrite the dynamic error model (11) in a discrete time form:

$$\mathbf{X}_e(k+1) = \mathcal{A}(k)\mathbf{X}_e(k) + \mathcal{B}(k)\mathbf{u}(k), \tag{16}$$

where

$$\mathcal{A} = \begin{bmatrix} 1 & \mathcal{T}\mathrm{w}_r & -\mathcal{T}\mathrm{v}_{yr} \\ -\mathcal{T}\mathrm{w}_r & 1 & \mathcal{T}\mathrm{v}_{xr} \\ 0 & 0 & 1 \end{bmatrix},$$
$$\mathcal{B} = \begin{bmatrix} \mathcal{T} & 0 & 0 \\ 0 & \mathcal{T} & 0 \\ 0 & 0 & \mathcal{T} \end{bmatrix}.$$

According to (16), we can define the cost function as

$$\mathbb{F}(\mathbf{X}_e, \mathbf{u}) = \sum_{i=1}^{\Phi} \mathbf{X}_e^T(k+i|k)\Lambda_Q\mathbf{X}_e(k+i|k) + \sum_{i=0}^{\Psi-1} \Delta\mathbf{u}^T(k+i|k)\Lambda_R\Delta\mathbf{u}(k+i|k), \tag{17}$$

where Φ and Ψ represent the prediction horizon and control horizon, respectively, and they satisfy $0 \leq \Psi \leq \Phi$ and $1 \leq \Phi$. Λ_Q and Λ_R are weighting matrices, which are positive and symmetric. $\mathbf{X}_e(k+i|k)$ is the prediction state predicted at the sample time k. $\mathbf{u}(k+i-1|k) + \Delta\mathbf{u}(k+i|k) = \mathbf{u}(k+i|k)$. The optimal control input is obtained by solving the minimization of the cost function with constrains:

$$\hbar = \min_{\mathbf{u}} \ \mathbb{F}(\mathbf{X}_e, \mathbf{u}), \tag{18}$$
$$s.t. \ \mathbf{X}_e(k) \in [\mathbf{X}^{\min}, \mathbf{X}^{\max}]$$
$$\mathbf{u} \in [\mathbf{u}^{\min}, \mathbf{u}^{\max}]$$
$$\Delta\mathbf{u} \in [\Delta\mathbf{u}^{\min}, \Delta\mathbf{u}^{\max}] \tag{19}$$

where $\mathbf{X}^{\min}$ and $\mathbf{X}^{\max}$ are the state constrains, $\mathbf{u}^{\min}$ and $\mathbf{u}^{\max}$ are the input control constrains, $\Delta\mathbf{u}^{\min}$ and $\Delta\mathbf{u}^{\max}$ are the input increment constrains.

Define the vectors:

$$\mathfrak{X}_k = [\mathbf{X}_e(k+1|k), \mathbf{X}_e(k+2|k) \ldots, \mathbf{X}_e(k+\Phi|k)]^T, \tag{20}$$

$$\mathfrak{U}_k = [\mathbf{u}(k|k), \mathbf{u}(k+1|k) \ldots, \mathbf{u}(k+\Psi-1|k)]^T, \tag{21}$$

$$\Delta\mathfrak{U}_k = [\Delta\mathbf{u}(k|k), \Delta\mathbf{u}(k+1|k) \ldots, \Delta\mathbf{u}(k+\Psi-1|k)]^T. \tag{22}$$

Then, the predicted output of (16) can be denoted as

$$\mathfrak{X}_k = \Theta\Delta\mathfrak{U}_k + \varpi + \xi, \tag{23}$$

where

$$\Theta = \begin{bmatrix} \mathcal{B}(k|k-1) & \ldots & 0 \\ \vdots & \ddots & \vdots \\ \vdots & \ddots & \vdots \\ \mathcal{B}(k+\Phi-1|k-1) & \ldots & \mathcal{B}(k+\Phi-1|k-1) \end{bmatrix},$$
$$\varpi = \begin{bmatrix} \mathcal{A}(k|k-1)\mathbf{X}_e(k|k-1) \\ \vdots \\ \mathcal{A}(k+\Phi-1|k-1)\mathbf{X}_e(k+\Phi-1|k-1) \end{bmatrix},$$
$$\xi = \begin{bmatrix} \mathcal{B}(k|k-1)\mathbf{u}(k-1) \\ \vdots \\ \mathcal{B}(k+\Phi-1|k-1)\mathbf{u}(k-1) \end{bmatrix}.$$

Based on (23), the optimized objective function (18) can be converted as

$$\hbar = min\ (\Theta\Delta\mathfrak{U}_k + \varpi + \xi)^T\tilde{\Lambda}_Q(\Theta\Delta\mathfrak{U}_k + \varpi + \xi) + \Delta\mathfrak{U}_k^T\tilde{\Lambda}_R\Delta\mathfrak{U}_k, \tag{24}$$
$$s.t.\ \Theta\Delta\mathfrak{U}_k + \varpi + \xi \in [\mathfrak{X}^{\min}, \mathfrak{X}^{\max}],$$
$$\Delta\mathfrak{U}_k \in [\Delta\mathfrak{U}^{\min}, \Delta\mathfrak{U}^{\max}],$$
$$\mathfrak{U}_{k-1} \in [\mathfrak{U}^{\min}, \mathfrak{U}^{\max}],$$
$$\mathfrak{U}_{k-1} + \tilde{I}\Delta\mathfrak{U}_k \in [\mathfrak{U}^{\min}, \mathfrak{U}^{\max}], \tag{25}$$

where $\tilde{\Lambda}_Q \in \mathcal{R}^{3\Phi\times3\Phi}$ and $\tilde{\Lambda}_R \in \mathcal{R}^{3\Psi\times3\Psi}$ are appropriate dimensional matrices, $\mathfrak{X}^{\min}$ and $\mathfrak{X}^{\max}$ are the state constrains, $\mathfrak{U}^{\min}$ and $\mathfrak{U}^{\max}$ are the input control constrains, $\Delta\mathfrak{U}^{\min}$ and $\Delta\mathfrak{U}^{\max}$ are the input increment constrains, and

$$\tilde{I} = \begin{bmatrix} I & 0 & \cdots & 0 \\ I & I & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ I & I & \cdots & I \end{bmatrix} \in \mathcal{R}^{3\Psi\times3\Psi}.$$

To solve the optimization function (24), we transform it into a quadratic programming (QP) problem:

$$min\ \frac{1}{2}\Delta\mathfrak{U}_k^T\varphi\Delta\mathfrak{U}_k + \rho^T\Delta\mathfrak{U}_k, \tag{26}$$
$$s.t.\ \Delta\mathfrak{U}^{\min} \leqslant \Delta\mathfrak{U}_k \leqslant \Delta\mathfrak{U}^{\max},$$
$$\begin{bmatrix} -\tilde{I} & \tilde{I} & -\Theta & \Theta \end{bmatrix}\Delta\mathfrak{U}_k \leqslant \begin{bmatrix} -\mathfrak{U}^{\min} + \mathfrak{U}_{k-1} \\ \mathfrak{U}^{\max} - \mathfrak{U}_{k-1} \\ -\mathfrak{X}^{\min} + \varpi + \xi \\ \mathfrak{X}^{\max} - \varpi - \xi \end{bmatrix}, \tag{27}$$

where

$$\varphi = \begin{bmatrix} (\Theta^T\tilde{\Lambda}_Q\Theta + \tilde{\Lambda}_R) & \Theta^T\tilde{\Lambda}_Q \\ \tilde{\Lambda}_Q\Theta & \tilde{\Lambda}_Q \end{bmatrix},$$
$$\rho = \begin{bmatrix} 2\Theta^T\tilde{\Lambda}_Q(\varpi + \xi) \\ 2\tilde{\Lambda}_Q(\varpi + \xi) \end{bmatrix}.$$

FIGURE 8. Experiment environment.

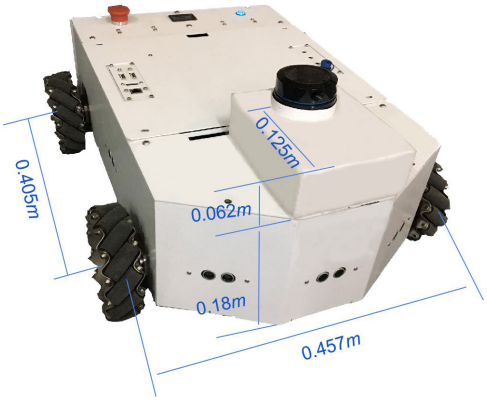
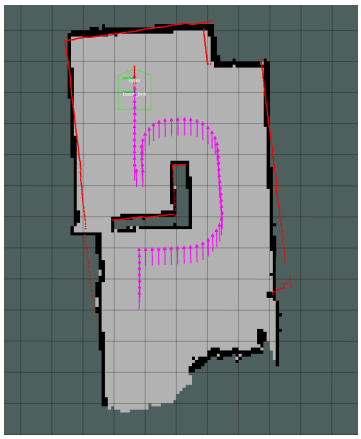


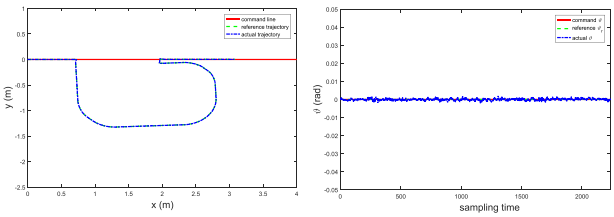
FIGURE 9. The four-wheel omnidirectional mobile robot.

V. EXPERIMENTS

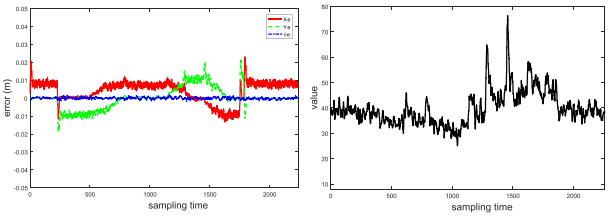
In this section, we apply the proposed NT-Bug algorithm to the shared control system to assist the operator in controlling the motion of the mobile robot. Three experiments are carried out to validate the effectiveness of the proposed shared control strategy and the novel NT-Bug algorithm. The experimental site is about 3m × 5m, as shown in Fig. 8. The four-wheel omnidirectional mobile robot is driven by four motors of 90 power and is controlled by a computer with a CPU of core i7-5500U. A laser radar with 8-meter scanning distance and 0.18 degrees angle accuracy is mounted at the front of the mobile robot platform (Fig. 9). The global coordinate system $\mathrm{X}^W\mathrm{O}^W\mathrm{Y}^W$ is set up at the starting position of the mobile robot, while its origin O^W is at the center of the mobile robot and X^W-axis is consistent with the forward direction of the mobile robot. Additionally, in order to show the experimental results more intuitively, we built a two-dimensional grid map for each experiment in advance. The actual position of the mobile robot would be recorded directly on the map in real time. The MPC parameters in the experiments are set as $\Phi = 3$, $\Psi = 2$, $\mathfrak{U}^{\max} = -\mathfrak{U}^{\min} = [1.5, 1.5, 1.5, 1.5, 1.5, 1.5]^T$, $\mathfrak{U}^{\max} = -\mathfrak{U}^{\min} = \Delta\mathfrak{U}^{\max} = -\Delta\mathfrak{U}^{\min} = [10, 10, 10, 10, 10, 10]^T$, $\mathfrak{X}^{\max} = -\mathfrak{X}^{\min} = [10, 10, 10, 10, 10, 10, 10, 10, 10]^T$, $\tilde{\Lambda}_Q = 2000I$, and $\tilde{\Lambda}_R = 40I$, where $I \in \mathcal{R}^{3\Phi\times 3\Phi}$ is an identity matrix. The minimum and maximum magnitude of the muscle stiffness are set as $\varsigma_{min} = 8$ and $\varsigma_{max} = 80$. The minimum and maximum linear velocity of the mobile robot are $v_{min} = 0.05m/s$ and $v_{max} = 0.15m/s$, respectively.



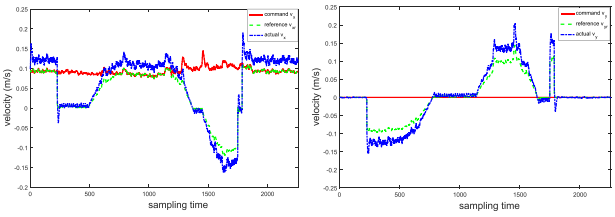
(a) Real-time position of the mobile robot recorded on the grid map.



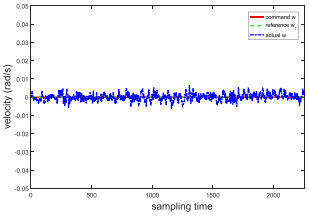
(b) Trajectory. (c) Orientation.



(d) Tracking error. (e) Muscle stiffness.

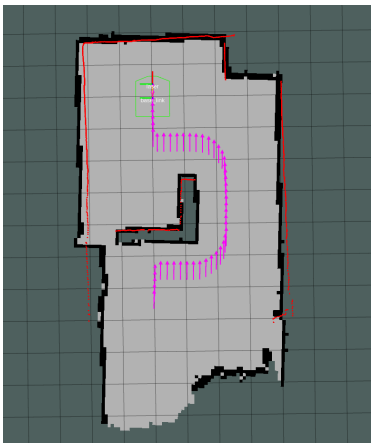


(f) v_x-Velocity. (g) v_y-Velocity.

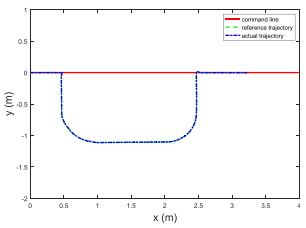


(h) w-Velocity.

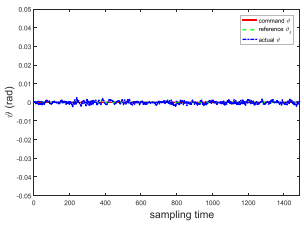
FIGURE 10. L-shaped obstacle avoidance in NT-Bug1 algorithm.



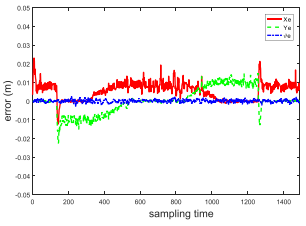
(a) Real-time position of the mobile robot recorded on the grid map.



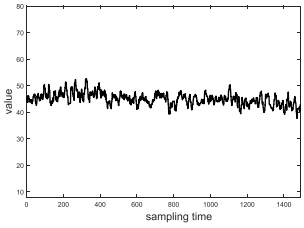
(b) Trajectory.



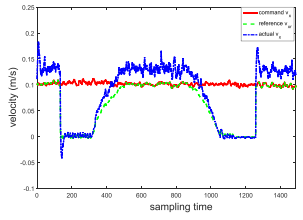
(c) Orientation.



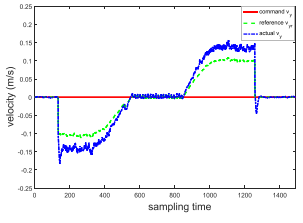
(d) Tracking error.



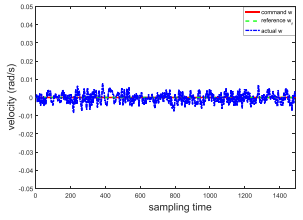
(e) Muscle stiffness.



(f) v_x-Velocity.

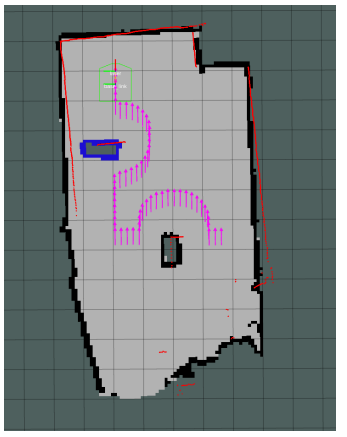


(g) v_y-Velocity.

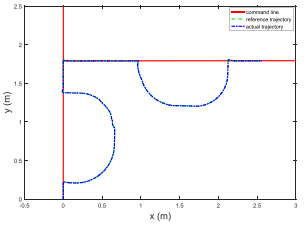


(h) w-Velocity.

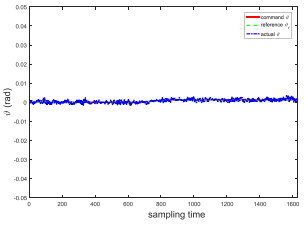
FIGURE 11. L-shaped obstacle avoidance in NT-Bug2 algorithm.



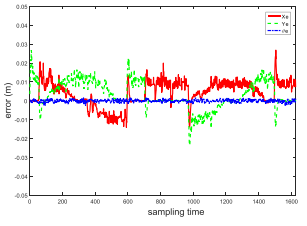
(a) Real-time position of the mobile robot recorded on the grid map.



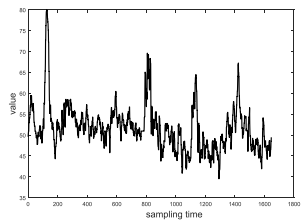
(b) Trajectory.



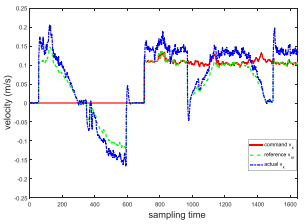
(c) Orientation.



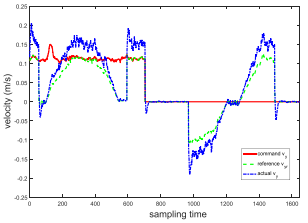
(d) Tracking error.



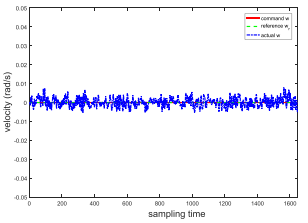
(e) Muscle stiffness.



(f) v_x-Velocity.



(g) v_y-Velocity.



(h) w-Velocity.

FIGURE 12. Static and dynamic obstacle avoidance in NT-Bug2 algorithm.

In the first experiment, there is a L-shaped obstacle in the workspace. The operator controls the mobile robot to move forward. The NT-Bug1 algorithm is applied to the shared control system for collision prevention. Its parameters are set as $\ell_{safe} = 0.55$, $\theta_{esc} = \pi/2$ and $\mathrm{d}_\sigma = 0.002$. The NT-Bug2 algorithm is introduced in the second experiment to demonstrate its advantage referring to the concave obstacle. For comparison, its experimental settings are the same as that of the first experiment. The last experiment is to validate the effectiveness of the NT-Bug2 algorithm for static and dynamic avoidance. One obstacle is put in the workspace in advance, the other is placed during the process of the experiment. Under the control of the operator, the mobile robot moves left at the first stage and moves forward at the second stage. The parameters of the NT-Bug2 are set as $\ell_{safe} = 0.5$, $\theta_{esc} = \pi/2$ and $\mathrm{d}_\sigma = 0.002$.

The results of the first and second experiments are shown in Figs. 10 and 11, respectively. In both experiments, the actual trajectories almost coincide with the reference trajectories, and the tracking errors are small (within $0.01m$), which suggests the trajectory tracking controller works effectively. Since the mobile robot is commanded to move forward, the command v_x is determined by the linear velocity while the command v_y retains zero. That is why the curves of command v_x are consistent with the curves of muscle stiffness, while the curves of command v_y remain zero. However, in the first experiment, the path around the contour of the obstacle is longer, and there are more corners on the trajectory, increasing the risk of collision. In contrast, the trajectory in the second experiment is smoother, and the switching frequency between v_x and v_y is lower than that in the first experiment. The comparison of the experiment results shows that the NT-Bug2 algorithm is more efficient for the shared control system. It also can plan a smoother and safer path for the mobile robot.

The last experiment results are shown in Fig. 12. The blue block on the grid map represents a dynamic obstacle, while the black block represents a static obstacle. The mobile robot is commanded to move left in the first stage, where the command v_x remains zero while the command v_y is in accordance with the muscle stiffness. At about sampling time 700, the mobile robot receives a new motion command of moving forward, that is why the curve of command v_x starts to vary according to the muscle stiffness while the curve of command v_y remains zero. The actual trajectory overlaps with the reference trajectory and the trajectory error maintains small when the system reaches the stable state. As Fig. 12(a) shown, the mobile robot can successfully bypass both static and dynamic obstacle in the NT-Bug2 algorithm. Therefore, the proposed NT-Bug2 algorithm is suitable for both dynamic and static avoidance.

VI. CONCLUSION

In this paper, we propose a novel shared control strategy for the mobile robot in a human-robot interaction manner based on the sEMG signals. To prevent the mobile robot from colliding with any obstacles, a collision avoidance scheme is introduced into the system. Aimed at the obstacle avoidance problem without a specific target, we develop a novel NT-Bug algorithm to guide the mobile robot to avoid obstacles and return to the `command line`. By adding the `command direction` into the obstacle avoidance process as a condition for the stage switching of the algorithm, the NT-Bug algorithm can plan a smoother and safer path for the mobile robot. MPC is introduced to control the mobile robot to track the reference trajectory. The experiment results have validated the effectiveness of the proposed shared control strategy and the NT-Bug algorithm.

REFERENCES

[1] L. Zhang, R. Liu, and M. Yue, "Shared control of robot based on fuzzy discrete event system," in *Proc. 11th World Congr. Intell. Control Autom.*, Jun. 2014, pp. 537–542.

[2] S.-Y. Jiang, C.-Y. Lin, K.-T. Huang, and K.-T. Song, "Shared control design of a walking-assistant robot," *IEEE Trans. Control Syst. Technol.*, vol. 25, no. 6, pp. 2143–2150, Jan. 2017.

[3] A.-T. Nguyen, C. Sentouh, and J.-C. Popieul, "Driver-automation cooperative approach for shared steering control under multiple system constraints: Design and experiments," *IEEE Trans. Ind. Electron.*, vol. 64, no. 5, pp. 3819–3830, May 2017.

[4] A. Franchi, C. Secchi, M. Ryll, H. Bulthoff, and P. Giordano, "Shared control: Balancing autonomy and human assistance with a group of quadrotor UAVs," *IEEE Robot. Autom. Mag.*, vol. 19, no. 3, pp. 57–68, Sep. 2012.

[5] Y. Lin, "A Natural contact sensor paradigm for nonintrusive and real-time sensing of biosignals in human-machine interactions," *IEEE Sensors J.*, vol. 11, no. 3, pp. 522–529, Mar. 2011.

[6] D. A. Abbink, M. Mulder, and E. R. Boer, "Cooperative dynamic vehicle control allocation using time-variant differential games," *Cognition, Technol. Work*, vol. 14, no. 1, pp. 19–28, Mar. 2012.

[7] J. N. Bailenson, E. D. Pontikakis, I. B. Mauss, J. J. Gross, M. E. Jabon, C. A. Hutcherson, C. Nass, and O. John, "Real-time classification of evoked emotions using facial feature tracking and physiological responses," *Int. J. Hum.-Comput. Stud.*, vol. 66, no. 5, pp. 303–317, May 2008.

[8] T. Farrell and R. Weir, "A comparison of the effects of electrode implantation and targeting on pattern classification accuracy for prosthesis control," *IEEE Trans. Biomed. Eng.*, vol. 55, no. 9, pp. 2198–2211, Sep. 2008.

[9] M. Simao, N. Mendes, O. Gibaru, and P. Neto, "A review on electromyography decoding and pattern recognition for human-machine interaction," *IEEE Access*, vol. 7, pp. 39564–39582, 2019.

[10] D. Ao, R. Song, and J. Gao, "Movement performance of human–robot cooperation control based on EMG-driven hill-type and proportional models for an ankle power-assist exoskeleton robot," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 25, no. 8, pp. 1125–1134, Aug. 2017.

[11] P. K. Artemiadis and K. J. Kyriakopoulos, "An EMG-based robot control scheme robust to time-varying EMG signal features," *IEEE Trans. Inf. Technol. Biomed.*, vol. 14, no. 3, pp. 582–588, May 2010.

[12] W. Xu, J. Huang, Y. Wang, C. Tao, and L. Cheng, "Reinforcement learning-based shared control for walking-aid robot and its experimental verification," *Adv. Robot.*, vol. 29, no. 22, pp. 1463–1481, Nov. 2015.

[13] T. Carlson and Y. Demiris, "Collaborative Control for a Robotic Wheelchair: Evaluation of Performance, Attention, and Workload," *IEEE Trans. Syst., Man, Cybern. B. Cybern.*, vol. 42, no. 3, pp. 876–888, Jun. 2012.

[14] D. Vanhooydonck, E. Demeester, A. Hüntemann, J. Philips, G. Vanacker, H. Van Brussel, and M. Nuttin, "Adaptable navigational assistance for intelligent wheelchairs by means of an implicit personalized user model," *Robot. Auton. Syst.*, vol. 58, no. 8, pp. 963–977, Aug. 2010.

[15] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. Syst. Sci. Cybern.*, vol. SSC-4, no. 2, pp. 100–107, Jul. 1968.

[16] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning," Dept. Comput. Sci., Iowa State Univ., Ames, IA, USA, Tech. Rep. TR:98-11, Oct. 1998.

[17] V. Lumelsky and A. Stepanov, "Dynamic path planning for a mobile automaton with limited information on the environment," *IEEE Trans. Autom. Control*, vol. AC-31, no. 11, pp. 1058–1063, Nov. 1986.
[18] P. Vadakkepat, K. Chen Tan, and W. Ming-Liang, "Evolutionary artificial potential fields and their application in real time robot path planning," in *Proc. Congr. Evol. Comput.*, Jul. 2000, pp. 256–263.
[19] X. Yang, P. Wei, Y. Zhang, X. Liu, and L. Yang, "Disturbance observer based on biologically inspired integral sliding mode control for trajectory tracking of mobile robots," *IEEE Access*, vol. 7, pp. 48382–48391, 2019.
[20] B. S. Park, S. J. Yoo, J. B. Park, and Y. H. Choi, "A simple adaptive control approach for trajectory tracking of electrically driven nonholonomic mobile robots," *IEEE Trans. Control Syst. Technol.*, vol. 18, no. 5, pp. 1199–1206, Sep. 2010.
[21] T. A. V. Teatro, J. M. Eklund, and R. Milman, "Nonlinear model predictive control for omnidirectional robot motion planning and tracking with avoidance of moving obstacles," *Can. J. Electr. Comput. Eng.*, vol. 37, no. 3, pp. 151–156, 2014.
[22] B. Ding, "Comments on 'constrained infinite-horizon model predictive control for fuzzy-discrete-time systems," *IEEE Trans. Fuzzy Syst.*, vol. 19, no. 3, pp. 598–600, Jun. 2011.
[23] S. Yu, X. Li, H. Chen, and F. Allgöwer, "Nonlinear model predictive control for path following problems," *Int. J. Robust Nonlinear Control*, vol. 25, no. 8, pp. 1168–1182, May 2015.
[24] H. Kim and B. K. Kim, "Online minimum-energy trajectory planning and control on a straight-line path for three-wheeled omnidirectional mobile robots," *IEEE Trans. Ind. Electron.*, vol. 61, no. 9, pp. 4771–4779, Sep. 2014.
[25] T. Terakawa, M. Komori, K. Matsuda, and S. Mikami, "A novel omnidirectional mobile robot with wheels connected by passive sliding joints," *IEEE/ASME Trans. Mechatronics*, vol. 23, no. 4, pp. 1716–1727, Aug. 2018.
[26] M. Wada and S. Mori, "Holonomic and omnidirectional vehicle with conventional tires," in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr. 1996, pp. 3671–3676.
[27] D. V. Lu, D. Hershberger, and W. D. Smart, "Layered costmaps for context-sensitive navigation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2014, pp. 709–715.
[28] D. G. Lloyd and T. S. Buchanan, "A model of load sharing between muscles and soft tissues at the human knee during static tasks," *J. Biomech. Eng.*, vol. 118, no. 3, pp. 367–376, Aug. 1996.

HAIYI KONG received the B.Eng. degree in automation from the College of Electrical Engineering, Guangxi University, Nanning, China, in 2017. She is currently pursuing the master's degree with the School of Automation Science and Engineering, South China University of Technology, Guangzhou, China.

Her research interests include mobile robotics, model predictive control, path planning, and human–machine interaction.

CHENGUANG YANG (Senior Member, IEEE) received the Ph.D. degree in control engineering from the National University of Singapore, Singapore, in 2010.

He has performed a Postdoctoral research in human robotics with Imperial College London, London, U.K., from 2009 to 2010. He is currently a professor of robotics. His research interests include human–robot interaction and intelligent system design. He has been awarded the EU Marie Curie International Incoming Fellowship, the U.K. EPSRC UKRI Innovation Fellowship, and the Best Paper Award of the IEEE TRANSACTIONS ON ROBOTICS as well as over ten conference best paper awards.

GUANG LI (Member, IEEE) received the Ph.D. degree in electrical and electronics engineering with a specialization in control systems from The University of Manchester, Manchester, U.K., in 2007.

He is currently a Senior Lecturer in dynamics modeling and control with the Queen Mary University of London, London, U.K. His research interests include constrained optimal control, model predictive control, adaptive robust control, and control applications, including renewable energies and energy storage.

SHI-LU DAI (Member, IEEE) received the B.Eng. degree in thermal engineering, and the M.Eng. and Ph.D. degrees in control science and engineering from Northeastern University, Shenyang, China, in 2002, 2006, and 2010, respectively.

He was a Visiting Student with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore, from November 2007 to November 2009, and a Visiting Scholar with the Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN, USA, from October 2015 to October 2016. Since 2010, he has been with the School of Automation Science and Engineering, South China University of Technology, Guangzhou, China, where he is currently a Professor. His current research interests include adaptive and learning control, and distributed cooperative systems.