

Safety-Augmented Operation of Mobile Robots Using Variable Structure Control

Azad Ghaffari and Seyed Amir Hosseini Dastja

Abstract—The design process and complexity of existing safety controls are heavily determined by the geometrical properties of the environment, which affects the proof of convergence, design scalability, performance robustness, and numerical efficiency of the control. Hence, this paper proposes a variable structure control to isolate the environment's geometrical complexity from the control structure. A super-twisting algorithm is used to achieve accurate trajectory tracking and robust safety control. The safety control is designed solely based on distance measurement. First, a nominal safety model for obstacle avoidance is derived, where realistic system constraints are considered. The nominal model is well-suited for safety control design for obstacle avoidance, geofencing, and border patrol with analytically proven stability results. The safety control utilizes distance measurement to maintain a safe distance by compensating the robot's angular velocity. A supervisory logic is constructed to guarantee the overall stability and safety of the system. Operational safety and precision tracking are proven under parametric uncertainty and environmental uncertainty. The proposed design is modular with minimal tuning parameters, which reduces the computational burden and improves the control scalability. The effectiveness of the proposed method is verified against various case studies.

I. INTRODUCTION

Obstacle avoidance is an integral part of safety control for nonholonomic mobile robots, self-driving cars, unmanned aerial vehicles, and surface vehicles [1]–[6]. The literature includes various techniques such as potential field [7]–[12], collision cone [13]–[17], path planning [18]–[21], model predictive control [22]–[25], and sliding-based method [26], [27]. Among early works, one can refer to the papers by Khatib [28], Krogh [29], Aggarwal, Leitmann, Skowronski [30], [31], and Borenstein and Koren [32]. Recent developments include the usage of barrier certificates and control barrier functions to design safety controllers [33]–[38].

Control design for autonomous vehicles needs to satisfy at least two objectives, such as trajectory tracking and obstacle avoidance. Thus, methods based on artificial potential field, kinodynamic motion planning, model predictive control, and barrier functions have been used in safety control design. However, it is known that the number, distribution, and shape of obstacles and mobile robots affect the analytical and computational complexity of the control design. For example, geometrical properties and spatial distribution of the obstacles directly affect the control structure and dramatically complicate proof of global convergence for such algorithms.

Popular methods such as artificial potential field and barrier functions require careful modeling of each object in the operational environment as a field or barrier function. In the case of artificial potential fields, the designer must foresee trap situations. On the other hand, barrier certificates must satisfy Lyapunov-like conditions, which render the application of the method very limited. Moreover, methods based on kinodynamic motion planning do not lead to exact solutions and are computationally intensive.

Sliding-based methods have been proven effective in the design of safety-augmented control of autonomous robots [2], [12], [26], [27]. Therefore, this paper provides control-oriented models and standardized design instructions based on a super-twisting algorithm to obtain a variable structure control with guaranteed safety features. The proposed design isolates the properties of the environment from the control design. The proposed control has a predefined structure with minimal control parameters, which conveniently handles multiple safety control problems, including obstacle avoidance, geofencing, and border patrol. The super-twisting algorithm is chosen as the primary control module to achieve precision trajectory tracking and safe operation. The super-twisting algorithm is robust, guarantees finite-time convergence, has a minimal set of parameters with explicit stability bounds, and features smooth transient and perfect tracking behavior.

This work focuses on nonholonomic mobile robots, which are driven by a differential drive. It is assumed that the thrust and torque are limited, and the linear velocity is positive. Thus, the robot always moves forward. The angular velocity, on the other hand, can take positive or negative values. Feedback linearization and the super-twisting algorithm are used to achieve accurate, robust trajectory tracking. The reference trajectory is known a priori and may cross obstacles or other robots. Also, in geofencing applications where a virtual safety net is introduced, the reference trajectory may leave the geofence for some periods. Moreover, only distance measurements, from obstacles, robots, or geofence boundaries are available to each mobile robot.

A framework based on distance measurement is presented for control design with augmented safety. Two system models have been obtained, one for trajectory tracking and one for safety control. The model used to design the trajectory tracking control is obtained using feedback linearization, with maximized control actuation along the transformed axes. Then, the super-twisting algorithm is used to guarantee precision trajectory tracking. Convergence is proven analytically. The safety control is designed using the dynamic model of distance variation. Actual system constraints, including actuator

Azad Ghaffari is with the Department of Mechanical Engineering, Wayne State University, MI 48202, USA, aghaffari@wayne.edu. Seyed Amir Hosseini Dastja is with EPSI and Danieli Automation, Isfahan, Iran, s.amir.hossaini@gmail.com.

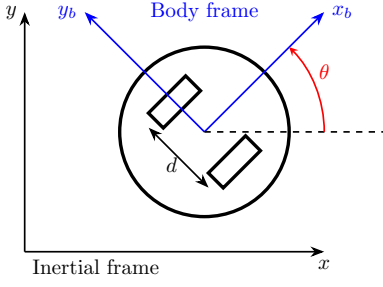


Fig. 1. Nonholonomic mobile robot and its reference frames

saturation, bounded velocity, and environmental characteristics, are used to obtain a nominal model suitable for safety control. The super-twisting algorithm compensates the effect of system uncertainties and environmental disturbances and maintains a safe distance between the robot and stationary or moving obstacles. When the safety control is active, the linear reference speed is modified such that the robot shadows the lead point of the reference trajectory. For small obstacles, however, the linear speed can be set to a fixed value.

A supervisory algorithm is designed to schedule the switching logic between trajectory tracking and safety control. The switching logic uses distance measurements, current robot position, and reference trajectory data to determine the time instants to switch back and forth between the two control modules. The supervisory algorithm handles obstacle avoidance and geofencing applications. The obstacle avoidance covers stationary and moving obstacles. The safety control can be used for border patrol applications.

The proposed modular safety control dramatically improves control scalability by utilizing the super-twisting algorithm. Regardless of the number and planar distribution of the obstacles or complexity of the reference trajectory, the proposed algorithm maintains the safe trajectory tracking for the nonholonomic mobile robot. The algorithm's analytical complexity and required processing power are linearly scaled when the number of obstacles increases in the experiment.

The rest of this paper is presented in the following order. Section II presents preliminary and system modeling. System linearization and allowable control bounds are obtained in Section III. Trajectory tracking is explained in Section IV. The safety control is presented in Section V. The supervisory algorithm is discussed in Section VI. Section VII presents numerical simulations to verify the effectiveness of the proposed method. Section VIII concludes the paper.

II. PRELIMINARY AND SYSTEM MODELING

The nonholonomic mobile robot is driven by a differential drive comprised of two identical electric wheels. Fig. 1 shows the schematic of the robot and inertial and body reference frames. The body reference frame is attached to the robot at the center of mass. The heading angle is θ , which is measured with respect to the x -axis. The kinematic equations of the robot are given as

$$\dot{x} = v \cos \theta \quad (1)$$

$$\dot{y} = v \sin \theta \quad (2)$$

$$\dot{\theta} = \omega, \quad (3)$$

where $p = [x \ y]^T$ is the robot's position in the inertial reference frame, θ is the heading angle, v is linear velocity, and ω is angular velocity. The force and torque generated by the electric wheels control the linear and angular velocity of the robot. Assuming that the left and right wheel's movement does not affect each other, one can decouple the dynamic equation of the left and right wheels. Also, the effect of friction and wheel slip on the dynamic model is negligible. Since the mobile robot is symmetric with identical left and right wheels and motors, it is reasonable to assume that both wheels have the same dynamic equation as shown in the following

$$\dot{v}_R = -av_R + bu_R + \Delta_R \quad (4)$$

$$\dot{v}_L = -av_L + bu_L + \Delta_L, \quad (5)$$

where a and b are positive constants, and v_R, v_L, u_R , and u_L are the right wheel speed, left wheel speed, right motor voltage, and left motor voltage, respectively. Also, Δ_R and Δ_L are perturbation terms representing model uncertainty and external disturbance for the right and left wheels. The perturbation terms are bounded and Lipschitz continuous. The Robot's linear and angular velocity is related to the wheels' velocity through the following equations

$$v = (v_R + v_L)/2 \quad (6)$$

$$\omega = (v_R - v_L)/d, \quad (7)$$

where d is the distance between the center of the wheels. Thus, the dynamic equations governing linear and angular velocity are obtained as

$$\dot{v} = -av + bu_v + \Delta_v \quad (8)$$

$$\dot{\omega} = -a\omega + bu_\omega + \Delta_\omega, \quad (9)$$

where

$$u_v = (u_R + u_L)/2 \quad (10)$$

$$u_\omega = (u_R - u_L)/d \quad (11)$$

$$\Delta_v = (\Delta_R + \Delta_L)/2 \quad (12)$$

$$\Delta_\omega = (\Delta_R - \Delta_L)/d, \quad (13)$$

where Δ_v and Δ_ω are bounded and Lipschitz continuous. Thus, the dynamic model of the mobile robot can be written as the following

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{v} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} v \cos \theta \\ v \sin \theta \\ \omega \\ -av + bu_v + \Delta_v \\ -a\omega + bu_\omega + \Delta_\omega \end{bmatrix}. \quad (14)$$

The voltage of the electric wheels are bounded as $|u_i| \leq U$ for $i = R, L$, where $0 < U < a^2d/b$. Using (10) and (11), one gets

$$|u_v| \leq U \quad (15)$$

$$|u_\omega| \leq 2U/d. \quad (16)$$

Consider the nominal model, where $\Delta_v = \Delta_\omega = 0$ in (14). Hence, if the initial condition satisfies $|v(0)| \leq bU/a$ and

$|\omega(0)| \leq 2bU/(ad)$, then one can show that

$$|v(t)| \leq bU/a \quad (17)$$

$$|\omega(t)| \leq 2bU/(ad), \quad (18)$$

for all $t > 0$.

The model of mobile robot for safety control and trajectory tracking can be transformed into combination of first- and second-order linear differential equations with perturbations. Thus, one can use the super-twisting algorithm (STA) to design the trajectory tracking and safety control. The design steps of the STA is explained. Consider the following system

$$\begin{aligned} \dot{z}_1 &= z_2 \\ \dot{z}_2 &= w + \Delta(t), \end{aligned} \quad (19)$$

where $\Delta(t)$ represents perturbations, including external disturbances and model uncertainties, which may be functions of time t . Assume $\Delta(t)$ is globally bounded and Lipschitz continuous, i.e.,

$$|\Delta(t)| \leq M_\Delta \quad (20)$$

$$\left| \dot{\Delta}(t) \right| \leq L_\Delta, \quad (21)$$

where M_Δ and L_Δ are some known positive constants. The control objective is to track a reference value, z_r , which is enough smooth. Denote the error variables as $e_1 = z_1 - z_r$ and $e_2 = z_2 - \dot{z}_r$. Hence, the error dynamics are obtained as

$$\begin{aligned} \dot{e}_1 &= e_2 \\ \dot{e}_2 &= w - \ddot{z}_r + \Delta(t). \end{aligned} \quad (22)$$

Consider a sliding surface as $\sigma = e_2 + \lambda e_1$, where $\lambda > 0$. The super-twisting algorithm is then designed as

$$w = w^{\text{eq}} - k_1 \sqrt{|\sigma|} \text{sign}(\sigma) - k_2 \int_0^t \text{sign}(\sigma) d\tau, \quad (23)$$

where $k_2 > L_\Delta$, $k_1 > 2\sqrt{k_2}$ [39], [40], and the equivalent control, w^{eq} , is obtained from $\dot{\sigma} = 0$ as

$$w^{\text{eq}} = \ddot{z}_r - \lambda e_2. \quad (24)$$

Moreover, if $|w| \leq M_w$, one can use an integrator anti-windup to improve the transient behavior of the STA.

III. LINEARIZATION AND ALLOWABLE CONTROL BOUNDS

The first control objective is to track a reference trajectory accurately. The super-twisting algorithm guarantees robust tracking performance with desirable transient and steady-state performance. First, the system dynamics are linearized using feedback linearization. Denote the following change of variables

$$[\eta_1 \ \xi_1]^T = [x \ y]^T + R_\theta [L \ 0]^T \quad (25)$$

$$[\eta_2 \ \xi_2]^T = R_\theta [v \ L\omega]^T, \quad (26)$$

where $L > 0$ is a positive constant, and R_θ is the rotation matrix given as

$$R_\theta = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}. \quad (27)$$

Also, two new control inputs u_η and u_ξ are defined as the following

$$\begin{bmatrix} u_\eta \\ u_\xi \end{bmatrix} = R_\theta \begin{bmatrix} -L\omega^2 - av + bu_v \\ v\omega - aL\omega + bLu_\omega \end{bmatrix}. \quad (28)$$

Applying (25)–(26), with (28) as the new control inputs, the original dynamic equation (14) is transformed into

$$\begin{bmatrix} \dot{\eta}_1 \\ \dot{\eta}_2 \\ \dot{\xi}_1 \\ \dot{\xi}_2 \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \eta_2 \\ u_\eta + \Delta_\eta \\ \xi_2 \\ u_\xi + \Delta_\xi \\ (\xi_2 \cos \theta - \eta_2 \sin \theta) / L \end{bmatrix}, \quad (29)$$

where

$$\begin{bmatrix} \Delta_\eta \\ \Delta_\xi \end{bmatrix} = R_\theta \begin{bmatrix} \Delta_v \\ L\Delta_\omega \end{bmatrix}. \quad (30)$$

Note that the perturbation terms are bounded and Lipschitz continuous. The transformed system is comprised of two double-integrators, and the angle appears as a zero dynamic. To convert (29) to the original dynamic equations (14), one can use the following inverse transformation

$$[x \ y]^T = [\eta_1 \ \xi_1]^T - R_\theta [L \ 0]^T \quad (31)$$

$$[v \ L\omega]^T = R_\theta^{-1} [\eta_2 \ \xi_2]^T \quad (32)$$

$$\begin{bmatrix} u_v \\ Lu_\omega \end{bmatrix} = \frac{1}{b} \begin{bmatrix} av + L\omega^2 \\ aL\omega - v\omega \end{bmatrix} + \frac{1}{b} R_\theta^{-1} \begin{bmatrix} u_\eta \\ u_\xi \end{bmatrix}. \quad (33)$$

Moreover, one can find the allowable range of the transformed control inputs for the nominal model, where $\Delta_\eta = \Delta_\xi = 0$. Denote $[w_1 \ w_2]^T = R_\theta^{-1} [u_\eta \ u_\xi]^T$. Thus, one can use (8) and (9) to transform (33) to the following

$$\begin{bmatrix} \dot{v} \\ L\dot{\omega} \end{bmatrix} = \begin{bmatrix} L\omega^2 \\ -v\omega \end{bmatrix} + \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}. \quad (34)$$

Consider $|w_1| \leq M_1$ and $|w_2| \leq M_2$, where M_1 and M_2 are positive constants. It is desirable to find M_1 and M_2 such that the acceleration bounds are satisfied. The bounds of the right-hand side of (34) are obtained as

$$-M_1 \leq w_1 + L\omega^2 \leq M_1 + \frac{4Lb^2}{a^2d^2} U^2 \quad (35)$$

$$-M_2 - \frac{2b^2}{a^2d} U^2 \leq w_2 - v\omega \leq M_2 + \frac{2b^2}{a^2d} U^2. \quad (36)$$

Also, the allowable bounds of the left-hand side of (34) are obtained as $|\dot{v}| \leq 2bU$ and $|L\dot{\omega}| \leq 4LbU/d$. Thus, the following must be satisfied

$$M_1 \leq 2bU - \frac{4Lb^2}{a^2d^2} U^2 \quad (37)$$

$$M_2 \leq \frac{4Lb}{d} U - \frac{2b^2}{a^2d} U^2. \quad (38)$$

To guarantee existence of positive M_1 and M_2 , the following must hold

$$\frac{bU}{2a^2} < L < \frac{a^2d^2}{2bU}. \quad (39)$$

Note that since $U < a^2d/b$, there is always a value for L that satisfies (39). Moreover, note that $\|R_\theta^{-1}\|_\infty \leq 1$. Thus, the bounds on u_v and u_ω are satisfied if the following holds

$$\left\| [u_\eta \ u_\xi]^T \right\|_\infty \leq U'(L), \quad (40)$$

where

$$U'(L) = \min \left\{ 2bU - \frac{4Lb^2}{a^2d^2}U^2, \frac{4Lb}{d}U - \frac{2b^2}{a^2d}U^2 \right\}. \quad (41)$$

Note that $L = d/2$ satisfies (39) and gives the maximum bounds of the transformed control inputs as

$$U'(d/2) = 2bU \left(1 - \frac{bU}{a^2d} \right). \quad (42)$$

The following lemma summarizes the discussion on the control bounds.

Lemma 1: Consider the system (14), where the following hold $|u_v| \leq U$, $|u_\omega| \leq 2U/d$, $|v(0)| \leq bU/a$, $\omega(0) \leq 2bU/(ad)$, where $0 < U < a^2d/b$. Then, $|v(t)| \leq bU/a$ and $|\omega(t)| \leq 2bU/(ad)$ for all $t > 0$ for the nominal system. Moreover, if (39) and (40) hold for the transformed inputs, the acceleration bounds of the original system (14) are satisfied. Also, $L = d/2$ maximizes the bounds of u_η and u_ξ , where the maximum bound is given by (42).

IV. TRAJECTORY TRACKING

The following explains the steps to design a trajectory control using the super-twisting algorithm. The reference trajectory is constructed such that the nonholonomic condition is satisfied, i.e.,

$$\dot{x}_r = v_r \cos \theta_r \quad (43)$$

$$\dot{y}_r = v_r \sin \theta_r \quad (44)$$

$$\dot{\theta}_r = \omega_r, \quad (45)$$

where the proper selection of v_r and ω_r , within the allowed bounds, creates a variety of reference trajectories. First, the reference values for the transformed dynamics are calculated using (43)–(45) and (25)–(26)

$$[\eta_{r1} \ \xi_{r1}]^T = [x_r \ y_r]^T + R_{\theta_r} [L \ 0]^T \quad (46)$$

$$[\eta_{r2} \ \xi_{r2}]^T = R_{\theta_r} [v_r \ L\omega_r]^T. \quad (47)$$

Denote the error variables as $\tilde{\eta}_i = \eta_i - \eta_{ri}$ and $\tilde{\xi}_i = \xi_i - \xi_{ri}$ for $i = 1, 2$, and $\tilde{\theta} = \theta - \theta_r$. Thus, the error dynamics are obtained as

$$\begin{bmatrix} \dot{\tilde{\eta}}_1 \\ \dot{\tilde{\eta}}_2 \\ \dot{\tilde{\xi}}_1 \\ \dot{\tilde{\xi}}_2 \\ \dot{\tilde{\theta}} \end{bmatrix} = \begin{bmatrix} \tilde{\eta}_2 \\ u_\eta - \dot{\eta}_{r2} + \Delta_\eta \\ \tilde{\xi}_2 \\ u_\xi - \dot{\xi}_{r2} + \Delta_\xi \\ \Psi \end{bmatrix}, \quad (48)$$

where

$$\Psi = \frac{(\xi_{r2} + \tilde{\xi}_2) \cos(\theta_r + \tilde{\theta}) - (\eta_{r2} + \tilde{\eta}_2) \sin(\theta_r + \tilde{\theta})}{L} - \omega_r, \quad (49)$$

where the bound of the control inputs is given by (40). In the subsequent analysis $L = d/2$.

Using the presented method in Section II, one can design the super-twisting algorithm for the error dynamics (48). The sliding surfaces are designed as

$$s_q = \tilde{q}_2 + c_q \tilde{q}_1, \quad c_q > 0, \quad q = \eta, \xi. \quad (50)$$

The super-twisting algorithm guarantees that s_q goes to zero in finite time, which means the following hold

$$[\tilde{\eta}_1 \ \tilde{\eta}_2]^T = [0 \ 0]^T \quad (51)$$

$$[\tilde{\xi}_1 \ \tilde{\xi}_2]^T = [0 \ 0]^T. \quad (52)$$

Therefore, at the equilibrium point, the zero dynamic equation transforms to

$$\dot{\tilde{\theta}} = \frac{\xi_{r2} \cos(\theta_r + \tilde{\theta}) - \eta_{r2} \sin(\theta_r + \tilde{\theta})}{L} - \omega_r. \quad (53)$$

Using (47), one can expand and simplify (53) to arrive at the following

$$\dot{\tilde{\theta}} = \frac{L\omega_r \cos \tilde{\theta} - v_r \sin \tilde{\theta}}{L} - \omega_r. \quad (54)$$

Denote $\phi = \arctan(v_r/(L\omega_r))$. Thus, (54) is further simplified as

$$\dot{\tilde{\theta}} = \frac{\sqrt{L^2\omega_r^2 + v_r^2}}{L} (\cos(\tilde{\theta} + \phi) - \cos \phi) \quad (55)$$

which has two equilibrium points at $\tilde{\theta}_{eq} = 0, -2\phi$. Applying the Jacobian linearization, one gets

$$\dot{\tilde{\theta}} = \begin{cases} -v_r \tilde{\theta}/L, & \text{if } \tilde{\theta}_{eq} = 0 \\ v_r \tilde{\theta}/L, & \text{if } \tilde{\theta}_{eq} = -2\phi \end{cases}. \quad (56)$$

Also, note that v_r is a positive real value. Hence, the equilibrium $\tilde{\theta}_{eq} = -2\phi$ is unstable. On the other hand, the equilibrium $\tilde{\theta}_{eq} = 0$ is asymptotically stable which means that the actual heading angle converges to the reference heading angle. Moreover, it immediately follows that $R_\theta = R_{\theta_r}$ since $\theta = \theta_r$. Hence, one can use (51)–(52), (25)–(26), and (46)–(47) to prove that $[x \ y]^T = [x_r \ y_r]^T$ and $[v \ \omega]^T = [v_r \ \omega_r]^T$ at the equilibrium. Thus, the proposed controller features perfect trajectory tracking.

The following proposition summarizes the control design and stability results.

Proposition 1: Consider that Lemma 1 holds for the mobile robot system given by (14). The reference trajectory is given by (43)–(45), where $v_r > 0$. Consider the nonlinear transformation given by (46) and (47), where $L = d/2$. The sliding surfaces are given by (50), where $c_\eta = c_\xi > 0$. Then, the STA asymptotically stabilizes the origin of (48) and the robot accurately tracks the reference trajectory.

In the next section, the problem of safety control is formulated using only distance measurement. A nominal safety model is obtained, and then the super-twisting algorithm is used to design the safety control.

V. SAFETY CONTROL

Recall the dynamic equation of the mobile robot given by (14). As the robot tracks a reference trajectory in an uncontrolled environment, stationary and moving objects may appear on the robot's path. Also, in geofencing applications, it is desirable to maintain the robot inside a virtual net even if the reference trajectory leaves the geofence. In border patrol applications, the robot is required to maintain a certain distance from the boundary of an arbitrary set. In all of these applications, the robot must maintain a safe distance with an

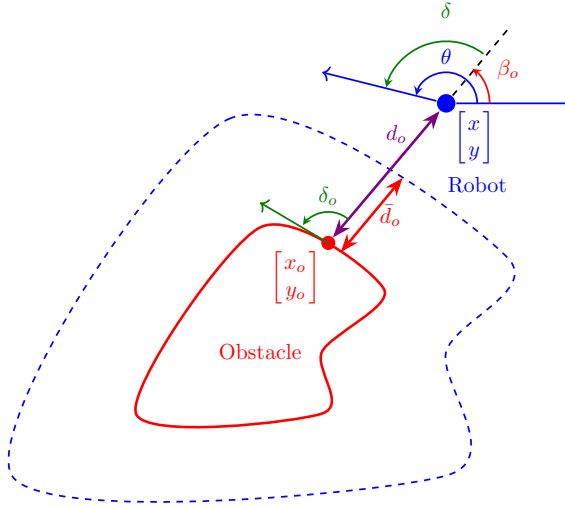


Fig. 2. An obstacle avoidance scenario is considered, where d_o is the closest distance from the robot to the obstacle. It is desirable to keep the robot's distance at \bar{d}_o from the obstacle.

object or reroute its path to maintain a safe distance with the set boundary. Thus, in the subsequent analysis, a nominal model is obtained to facilitate safety control based on distance measurements. Then, the super-twisting algorithm is used to design the safety control.

Consider a mobile robot given by (14), where the robot's position is $p = [x \ y]^T$. Also, as shown in Fig. 2, the closest point of the obstacle to the mobile robot is shown by $p_o = [x_o \ y_o]^T$. Since the location of p_o changes when the robot moves, one can calculate the translational and rotational velocity of p_o using the following relationships

$$v_o = \sqrt{\dot{x}_o^2 + \dot{y}_o^2} \quad (57)$$

$$\omega_o = \frac{\ddot{y}_o \dot{x}_o - \ddot{x}_o \dot{y}_o}{\dot{x}_o^2 + \dot{y}_o^2}. \quad (58)$$

As shown later, the knowledge of v_o and ω_o is not required for the control design. However, the upper bound of v_o and ω_o is required to prove the stability of the safety control. Also, since v_o and ω_o are constructed using (57) and (58), then the following holds for p_o

$$\dot{x}_o = v_o \cos \theta_o \quad (59)$$

$$\dot{y}_o = v_o \sin \theta_o \quad (60)$$

$$\dot{\theta}_o = \omega_o. \quad (61)$$

Denote the length of $p_o p$ as d_o and its angle with the horizontal axis as β_o calculated as

$$d_o = \sqrt{(x - x_o)^2 + (y - y_o)^2} \quad (62)$$

$$\beta_o = \arctan\left(\frac{y - y_o}{x - x_o}\right). \quad (63)$$

Denote $\delta = \theta - \beta_o$ and $\delta_o = \theta_o - \beta_o$. Note that $\theta - \theta_o = \delta - \delta_o$.

Next, take the derivative of d_o and β_o with respect to time.

$$\begin{aligned} \dot{d}_o &= (v \cos \theta - v_o \cos \theta_o) \frac{x - x_o}{d_o} + \\ &\quad + (v \sin \theta - v_o \sin \theta_o) \frac{y - y_o}{d_o} \\ &= (v \cos \theta - v_o \cos \theta_o) \cos \beta_o + \\ &\quad + (v \sin \theta - v_o \sin \theta_o) \sin \beta_o \\ &= v \cos \delta - v_o \cos \delta_o \end{aligned} \quad (64)$$

$$\begin{aligned} \dot{\beta}_o &= (v \sin \theta - v_o \sin \theta_o) \frac{\cos \beta_o}{d_o} - \\ &\quad - (v \cos \theta - v_o \cos \theta_o) \frac{\sin \beta_o}{d_o} \\ &= \frac{v}{d_o} \sin \delta - \frac{v_o}{d_o} \sin \delta_o. \end{aligned} \quad (65)$$

Taking the time derivative of \dot{d}_o gives

$$\ddot{d}_o = \dot{v} \cos \delta - v \dot{\delta} \sin \delta - \dot{v}_o \cos \delta_o + v_o \dot{\delta}_o \sin \delta_o. \quad (66)$$

Note that $\dot{\delta} = \omega - \dot{\beta}_o$ and $\dot{\delta}_o = \omega_o - \dot{\beta}_o$, where $\dot{\beta}_o$ is given by (65). Also, \dot{v} is given by (14). Thus, one can write (66) as the following

$$\begin{aligned} \ddot{d}_o &= (-av + bu_v + \Delta_v) \cos \delta - v\omega \sin \delta + v_o \omega_o \sin \delta_o + \\ &\quad + \dot{\beta}_o (v \sin \delta - v_o \sin \delta_o) - \dot{v}_o \cos \delta_o \\ &= (-av + bu_v) \cos \delta - v\omega \sin \delta + \frac{v^2}{d_o} \sin^2 \delta + \Delta_o, \end{aligned} \quad (67)$$

where

$$\begin{aligned} \Delta_o &= \Delta_v \cos \delta + v_o \omega_o \sin \delta_o + \frac{v_o^2}{d_o} \sin^2 \delta_o - \\ &\quad - \frac{2vv_o}{d_o} \sin \delta \sin \delta_o - \dot{v}_o \cos \delta_o. \end{aligned} \quad (68)$$

Note that the robot linear and angular velocity are bounded. Also, assuming that the safe boundary around the obstacle is smooth enough such that (57) and (58) is bounded and smooth enough, then one can find positive values L_{o1} and L_{o2} such that the following is satisfied

$$|\Delta_o| \leq L_{o1}, \quad |\dot{\Delta}_o| \leq L_{o2}. \quad (69)$$

The safety control is active when the robot's distance to the obstacle is within a determined safety bound, i.e.,

$$\bar{d}_o - \epsilon_o \leq d_o \leq \bar{d}_o + \epsilon_o, \quad (70)$$

where $0 < \epsilon_o \ll \bar{d}_o$. Also, the angle δ is either $\pi/2$ for counterclockwise turns or $-\pi/2$ for clockwise turns. In the subsequent analysis, the counterclockwise scenario is addressed, i.e.,

$$\pi/2 - \epsilon_\delta \leq \delta \leq \pi/2 + \epsilon_\delta, \quad (71)$$

where $0 < \epsilon_\delta \ll \pi/2$. Relationships between angles and distances are shown in Fig. 2. Using (70) and (71), one can accurately approximate (67) as the following

$$\ddot{d}_o = -v\omega + v^2/\bar{d}_o + \Delta'_o, \quad (72)$$

where $\Delta'_o = \Delta_o + \Delta_m$, where Δ_m accounts for the effect of model approximation, where $|\Delta'_o| < L'_{o1}$, $|\dot{\Delta}'_o| < L'_{o2}$, where L'_{o1} and L'_{o2} are positive constants. Denote $\zeta_1 = \bar{d}_o - \epsilon_o$

and $\zeta_2 = \dot{\zeta}_1$, where \bar{d}_o is the safe distance from the obstacle. Hence, the nominal safety model is given as

$$\begin{bmatrix} \dot{\zeta}_1 \\ \dot{\zeta}_2 \\ \dot{\omega} \\ \dot{v} \\ \dot{\delta} \end{bmatrix} = \begin{bmatrix} \zeta_2 \\ -v\omega + v^2/\bar{d}_o + \Delta'_o \\ -a\omega + b\omega_\omega \\ -av + bu_v \\ \omega - (v \sin \delta - v_o \sin \delta_o) / (\bar{d}_o + \zeta_1) \end{bmatrix}. \quad (73)$$

During the safety control phase, the translational speed is set to the reference value, v_r , and the angular velocity is compensated to maintain the safe distance at \bar{d}_o . To design the safety control, first, the backstepping technique is used to drive ζ_1 and ζ_2 to zero. Denote $\tilde{\omega} = \omega - \hat{\omega}_r$, $\tilde{v} = v - v_r$, and $\tilde{\delta} = \delta - \pi/2$, where

$$\hat{\omega}_r = v/\bar{d}_o - \hat{u}_\zeta/v. \quad (74)$$

Also, define the control inputs as

$$u_\omega = (-c_\omega \tilde{\omega} + a\omega + \dot{\hat{\omega}}_r)/b \quad (75)$$

$$u_v = (-c_v \tilde{v} + av)/b, \quad (76)$$

where $c_v > 0$ and $c_\omega > 0$. Thus, the safety model is updated as

$$\begin{bmatrix} \dot{\zeta}_1 \\ \dot{\zeta}_2 \\ \dot{\tilde{\omega}} \\ \dot{\tilde{v}} \\ \dot{\tilde{\delta}} \end{bmatrix} = \begin{bmatrix} \zeta_2 \\ \hat{u}_\zeta + \Delta''_o \\ -c_\omega \tilde{\omega} \\ -c_v \tilde{v} \\ \Phi \end{bmatrix}, \quad (77)$$

where

$$\Phi = \hat{\omega}_r + \tilde{\omega} - \frac{(v_r + \tilde{v}) \cos \tilde{\delta} - v_o \sin \delta_o}{\bar{d}_o + \zeta_1}, \quad (78)$$

where $\Delta''_o = \Delta'_o - v\tilde{\omega}$, where $|\Delta''_o| < L''_{o1}$ and $|\dot{\Delta}''_o| < L''_{o2}$, where L''_{o1} and L''_{o2} are positive constants. Note that $\hat{\omega}_r$ is an auxiliary input, which must satisfy (18). Thus, using (74), one arrives at

$$|\hat{\omega}_r v| \leq |v^2/\bar{d}_o| + |\hat{u}_\zeta|, \quad (79)$$

where

$$|\hat{\omega}_r v| - |v^2/\bar{d}_o| \leq 2b^2 U^2/(a^2 d) - b^2 U^2/(a^2 \bar{d}_o). \quad (80)$$

Therefore, the bounds on \hat{u}_ζ are obtained as

$$|\hat{u}_\zeta| \leq (bU/a)^2 (2/d - 1/\bar{d}_o), \quad (81)$$

where $\bar{d}_o > d/2$, which verifies the fact that the safe distance, measured from the robot's center of mass, must be larger than the wheels' distance from the robot's center of mass. Otherwise, the robot will collide with the obstacle.

The super-twisting algorithm is used to design \hat{u}_ζ . First, the sliding surface is designed as

$$s_\zeta = \zeta_2 + c_\zeta \zeta_1, \quad c_\zeta > 0. \quad (82)$$

Next, the super-twisting algorithm proves that s_ζ goes to zero in finite time which in turn proves that ζ_1 and ζ_2 converge to zero asymptotically. Hence, the robot is maintained at the safe distance, \bar{d}_o , from point p_o .

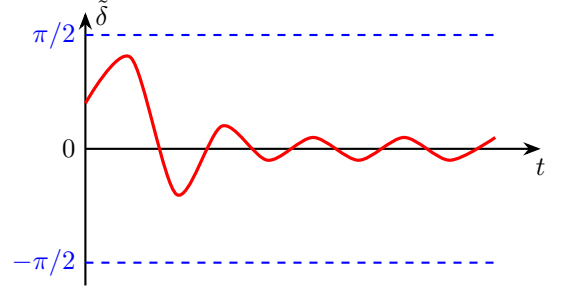


Fig. 3. Variation of $\tilde{\delta}$ over time. The safety algorithm maintains $\tilde{\delta} \approx 0$.

Furthermore, the error system (77) shows that $\tilde{\omega}$ and \tilde{v} converge to zero. Assume that the obstacle is a single stationary point, i.e., $v_o = 0$. Thus, the zero dynamic can be written as

$$\dot{\tilde{\delta}} = \hat{\omega}_r - \frac{v_r}{\bar{d}_o} \cos \tilde{\delta}. \quad (83)$$

The behavior of (83) depends on the sign of $\hat{\omega}_r$. If the robot is within the avoidance distance and $d_o > \bar{d}_o$, then $-\pi/2 < \tilde{\delta} < 0$. Thus, the super-twisting algorithm creates enough positive angular velocity to drive the distance toward \bar{d}_o . Therefore, $\tilde{\delta} \rightarrow 0$. On the other hand, if $d_o < \bar{d}_o$, then $0 < \tilde{\delta} < \pi/2$. Thus, the super-twisting algorithm creates enough negative angular velocity to drive the distance to \bar{d}_o , and thus $\tilde{\delta} \rightarrow 0$. The super-twisting algorithm continuously modifies the angular velocity to maintain the robot at the safe distance. Thus, as shown in Fig. 3, the two behaviors always are present, which means the zero dynamic is stable in an average sense, and thus $\tilde{\delta} \approx 0$. A similar argument can be used for the case when the obstacle is moving, or the problem of border patrol is addressed. Similar results also can be obtained for the case of clockwise safety control.

The following proposition summarizes the results of counterclockwise safety control.

Proposition 2: Consider the mobile robot modeled by (14), where Δ_v and Δ_ω are bounded and Lipschitz continuous. Assume the robot is in the vicinity of the avoidance distance, i.e., $|d_o - \bar{d}_o| \leq \epsilon_o$, where $0 < \epsilon_o \ll d_o$. Moreover, the closest point of the obstacle to the robot changes as (59)–(61), where v_o and ω_o are bounded and enough smooth. The control law is given as (74)–(76), where \hat{u}_ζ is designed using the super-twisting algorithm. Then, the origin of the error dynamics (77) is stable and the robot maintains safe distance \bar{d}_o with the obstacle.

VI. SUPERVISORY ALGORITHM

The proposed control algorithm comprises trajectory tracking and safety control, where the stability of each module is separately proven. This section presents the supervisory algorithm that provides the switching logic between the two modules. The proposed supervisory algorithm can handle obstacle avoidance of stationary and moving objects, geofencing, and border patrol applications.

The mobile robot dynamic equation is given by (14). Denote $X = [x \ y \ \theta \ v \ \omega]^T$, where $X \in \mathcal{X}$, where $\mathcal{X} \subseteq \mathbb{R}^5$. The reference trajectory is defined as $X_r = [x_r \ y_r \ \theta_r \ v_r \ \omega_r]^T$. Note that the reference trajectory satisfies the nonholonomic

condition. A set of initial conditions $\mathcal{X}_0 \subset \mathcal{X}$ and a set of unsafe states $\mathcal{X}_u \subset \mathcal{X}$ are given. The safety is achieved if all the state trajectories initiated inside \mathcal{X}_0 avoid the unsafe set for all $t > 0$. The avoidance zone of an obstacle is a strip of a predefined width around the obstacle. In keep-in geofencing applications, the geofence's avoidance zone is a strip of predefined width around the inner border. The width of the avoidance zone may differ between obstacles. The safe set of the mobile robot is bound by the geofence borders and excludes the border's avoidance zone and all obstacles and their avoidance zones.

It is desirable to design the supervisory algorithm such that the robot accurately tracks the reference trajectory inside the safe set and successfully avoids the unsafe set. Consider the following assumptions.

Assumption 1: The system starts from a safe state.

Assumption 2: The stationary obstacles are far apart with non-intersecting avoidance zones. Also, the robot can navigate safely through spaces between neighboring obstacles.

Assumption 3: At any given instant, no more than two robots are on a collision path.

Assumption 4: The obstacle dimensions are comparable to the robot dimensions.

Consider counterclockwise safety control is implemented in the following discussion. Similar results can be produced for clockwise safety control. The position error in the robot body frame is obtained as

$$[e_x \ e_y]^T = R_\theta [x_r - x \ y_r - y]^T, \quad (84)$$

where R_θ is the rotation matrix given by (27). Note that e_y is the projection of position error vector along y_b -axis. See Fig. 4.

Since the system initially is at a safe state, $\{X, X_r\} \subset \mathcal{X}_0$, the trajectory tracking algorithm is active when the experiment starts. If the reference trajectory and robot are in the vicinity of an obstacle or another robot, $\{X, X_r\} \subset \mathcal{X}_u$, the safety control is activated. The safety control may be deactivated when the reference trajectory leaves the avoidance zone. However, the robot's position relative to the reference trajectory affects the decision to reactivate the trajectory tracking.

As Fig. 4 shows, the reference trajectory leaves the avoidance zone at point p_{r1} while the robot is at point p_1 . One can observe that the straight path to the reference trajectory intersects with the avoidance zone, i.e., $e_y > 0$. Thus, if the safety control is deactivated, the robot may go back inside the avoidance zone. Hence, excessive switching between safety control and trajectory tracking may worsen system performance and harm the actuators. On the other hand, when the robot arrives at point p_2 , where the corresponding reference point is at p_{r2} , one can observe that the straight line between the robot and reference point no longer crosses the avoidance zone. Thus, $e_y \leq 0$ indicates that if the safety control is deactivated, the robot leaves tangent to the avoidance boundary. Thus, the safety control can be turned off when the reference trajectory is outside the avoidance region and $e_y \leq 0$. Hence, excessive switching is avoided. If the clockwise safety control is implemented, $e_y \geq 0$ deactivates the safety control.

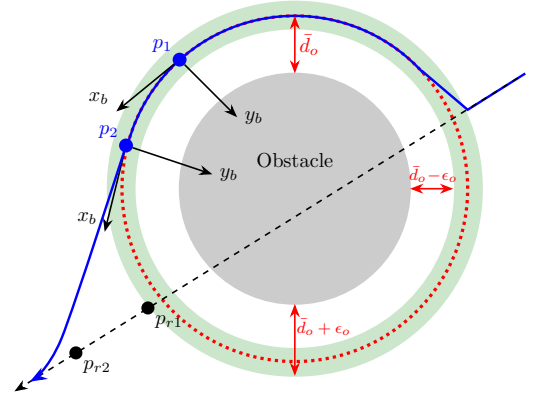


Fig. 4. Obstacle avoidance scenario. The reference trajectory is shown in dashed black. The robot moves from north-east to south-west on the solid blue curve. If the trajectory control activated at p_1 , excessive switching happens between trajectory tracking and safety control. The safe distance is d_o and $0 < \epsilon_o \ll \bar{d}_o$. The green strip is the avoidance zone.

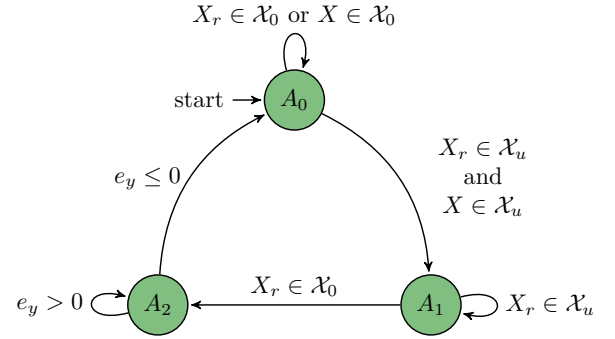


Fig. 5. State machine for obstacle avoidance. The trajectory control is active in state A_0 , and the safety control is active in states A_1 and A_2 . If conditions $e_y > 0$ and $e_y \leq 0$ are replaced with $e_y < 0$ and $e_y \geq 0$, respectively, the logic can be used for keep-in geofencing or border patrol.

The supervisory algorithm for obstacle avoidance is shown in Fig. 5. Similar results can be produced for clockwise safety control. If the obstacle is oversize, the obstacle avoidance problem can be treated as a geofencing problem. In keep-in geofencing, it is desirable to maintain the robot inside the safe zone at a safe avoidance distance with the border regardless of the changes in the reference trajectory. The safe zone comprises a strip of width \bar{d}_o around the inner border. If the conditions $e_y > 0$ and $e_y \leq 0$ in Fig. 5 are replaced with $e_y > 0$ and $e_y \geq 0$, respectively, one obtains the supervisory algorithm with counterclockwise safety control for keep-in geofencing and border patrol applications.

VII. CONTROL IMPLEMENTATION AND CASE STUDIES

Four case studies are designed to verify the effectiveness of the proposed safety control, including a) avoiding stationary obstacles, b) safe passage of colliding robots, c) keep-in geofencing, and d) border patrol.

The trajectory tracking control is the same for the four case studies as given in the following

$$u_q = u_{eq} - k_{q1} \sqrt{|s_q|} \text{sign}(s_q) - k_{q2} \int_0^t \text{sign}(s_q) d\tau, \quad (85)$$

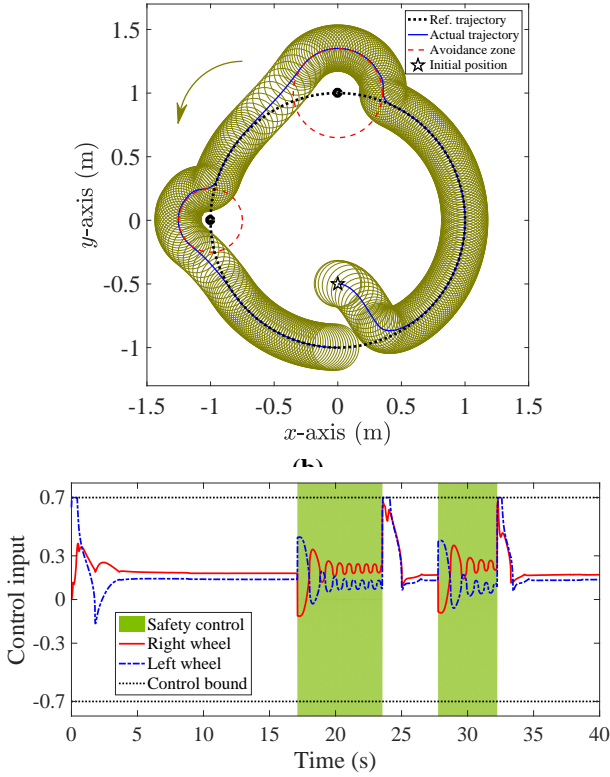


Fig. 6. (a) The robot accurately tracks the reference trajectory outside the avoidance zone and successfully avoids the stationary obstacles. Solid blue is the center of mass of the robot. (b) Evolution of control commands versus time. The safety control is active in the green area.

where $s_q = \tilde{q}_2 + c_q \tilde{q}_1$ for $q = \eta, \xi$. The equivalent control is given as

$$u_q^{eq} = \ddot{q}_r - c_q \tilde{q}_2, \quad q = \eta, \xi. \quad (86)$$

The control parameters are set as $k_{q1} = 2$, $k_{q2} = 0.5$, $c_q = 10$ for $q = \eta, \xi$. Note that $L = d/2$. Thus, one can use (40) and (42) to obtain the control bounds as $|u_q| \leq U_q$, where

$$U_q = 2bU \left(1 - \frac{bU}{a^2 d} \right), \quad q = \eta, \xi, \quad (87)$$

where $a = b = 3.85$, $d = 0.235$ m, $U = 0.7$ V. The saturated control command is then calculated as $u_q^{\text{sat}} = U_q \text{sat}(u_q/U_q)$ for $q = \eta, \xi$, where the saturation function is defined as

$$\text{sat}(\psi) = \begin{cases} \psi & \text{if } |\psi| < 1 \\ \text{sign}(\psi) & \text{if } |\psi| \geq 1 \end{cases}. \quad (88)$$

Since the control (85) includes an integrator and the control bounds are known, one can add an integrator anti-windup to improve the closed-loop performance of the system. Next, one can use (33) to calculate u_v and u_ω from u_η^{sat} and u_ξ^{sat} , where $|u_v| \leq U$, $u_\omega \leq U_\omega$, where $U_\omega = 2U/d$. Hence, the control command of the left and right wheels are obtained as

$$\begin{bmatrix} u_R \\ u_L \end{bmatrix} = \begin{bmatrix} 1 & d/2 \\ 1 & -d/2 \end{bmatrix} \begin{bmatrix} U \text{sat}(u_v/U) \\ U_\omega \text{sat}(u_\omega/U_\omega) \end{bmatrix}. \quad (89)$$

Because of the actuator limitation, the saturated command signals are applied to the wheels, i.e., $U \text{sat}(u_R/U)$ and $U \text{sat}(u_L/U)$.

A. Avoiding Stationary Obstacles

Assume that the robot tracks a reference trajectory in an environment where stationary obstacles are present. The reference trajectory may cross the obstacles. The safety control is given as (75)–(76), where $\hat{\omega}_r$ is given by (74). However, as explained earlier in Fig. 3, $\tilde{\delta}$ does not settle at a fixed value, which means $\hat{\omega}_r$ features continuous change. Thus, to improve performance robustness, a modified version of u_ω is implemented as the following

$$u_\omega = (-c_\omega \tilde{\omega} + a\omega) / b, \quad (90)$$

where $\tilde{\omega} = \omega - \hat{\omega}_r$, where $\hat{\omega}_r$ is given by (74). The super-twisting algorithm is used to design \hat{u}_ζ as

$$\hat{u}_\zeta = -c_\zeta \zeta_2 - k_{\zeta 1} \sqrt{|s_\zeta|} \text{sign}(s_\zeta) - k_{\zeta 2} \int_0^t \text{sign}(s_\zeta) d\tau, \quad (91)$$

where $s_\zeta = \zeta_2 + c_\zeta \zeta_1$. The control parameters are designed as $k_{\zeta 1} = 0.8$, $k_{\zeta 2} = 0.04$, $c_\zeta = 1$, $c_u = c_\omega = 5$. The bounds of \hat{u}_ζ are given by (81). Thus, an integrator anti-windup can be added to the safety algorithm to further improve the control performance. Also, to incorporate the control saturation, one can modify (74) as

$$\hat{\omega}_r = v / \bar{d}_o - \text{sat}(\hat{u}_\zeta / U_\zeta) / v, \quad (92)$$

where $U_\zeta = (bU/a)^2 (2/d - 1/\bar{d}_o)$, where \bar{d}_o is the safe distance.

Two stationary obstacles are considered at positions $p_1 = [0 \ 1]^T$ and $p_2 = [-1 \ 0]^T$, where the safe distance is 0.35 m for p_1 and 0.25 m for p_2 . The robot is located at $p_0 = [0 \ -0.5]^T$. As shown in Fig. 6(a), the robot accurately tracks the reference trajectory and successfully avoids the two obstacles. The green circles represent the robot's actual dimension, and the dashed red circles are the avoidance zones. Note that the robot is modeled as a point mass. The solid blue line represents the center of mass of the robot. The reference velocities are given as $v_r = 0.5\pi$ m/s and $\omega_r = 0.5\pi$ rad/s. The evolution of the control signals is shown in Fig. 6(b), where the green area shows the duration where the safety control is active. During the safety control, $v_r = 0.5\pi$ m/s, and the angular velocity is manipulated to maintain the safe distance with the obstacle.

B. Safe Passage of Colliding Robots

Consider a situation where two robots are moving on the same trajectory in the opposite direction. Thus a collision is inevitable without safety control. Each robot is equipped with the same obstacle avoidance algorithm designed for avoiding stationary obstacles. As Fig. 7(a) shows, the robots safely pass each other. The robots are identical and $\bar{d}_o = 0.5$ m for both robots. As shown in Fig. 7(b), the safe distance is kept at \bar{d}_o , although the reference trajectories result in a collision. The initial positions of the robots are $[1.4 \ 0]^T$ m and $[0 \ -1.4]^T$ m.

C. Keep-In Geofencing

The proposed algorithm can also handle the problem of geofencing and border patrol. The trajectory tracking and safety

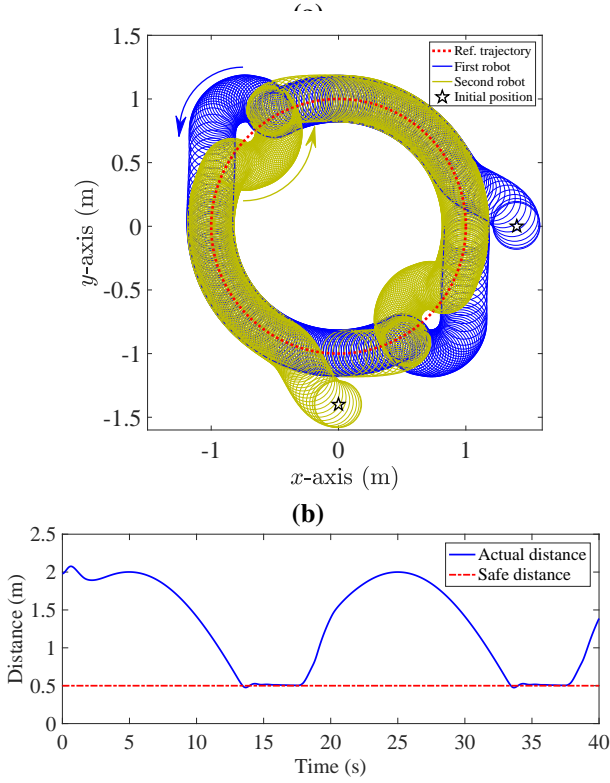


Fig. 7. (a) Two robots move on the same trajectory in opposite directions. The blue robot moves counterclockwise and the green robot moves clockwise. The proposed algorithm guarantees safe operation. (b) Distance between the two robots. The robots do not violate the safety distance.

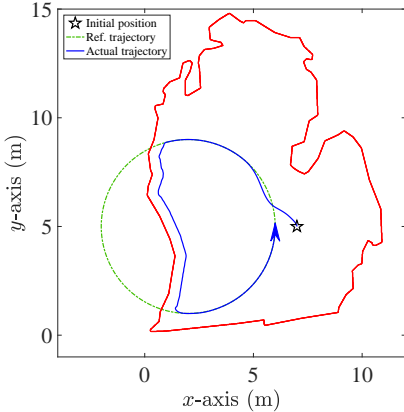


Fig. 8. The proposed algorithm accurately tracks the reference trajectory while maintaining the robot at a safe distance inside the geo-fence shown in solid red.

control have the same structure as the obstacle avoidance, except $c_\zeta = 10$, $c_u = c_\omega = 3$. Moreover, performance may be improved by modifying the linear reference speed during the safety phase such that the robot shadows the lead point of the reference trajectory.

For example, for keep-in geofencing application, if the length of the reference trajectory outside the border, is longer than the safe border cropped by the reference trajectory, one can use the following

$$\hat{v}_r = v_r \cos(\theta - \theta_r), \quad (93)$$

to modify the linear reference velocity. Providing a general modification algorithm for the linear reference velocity is

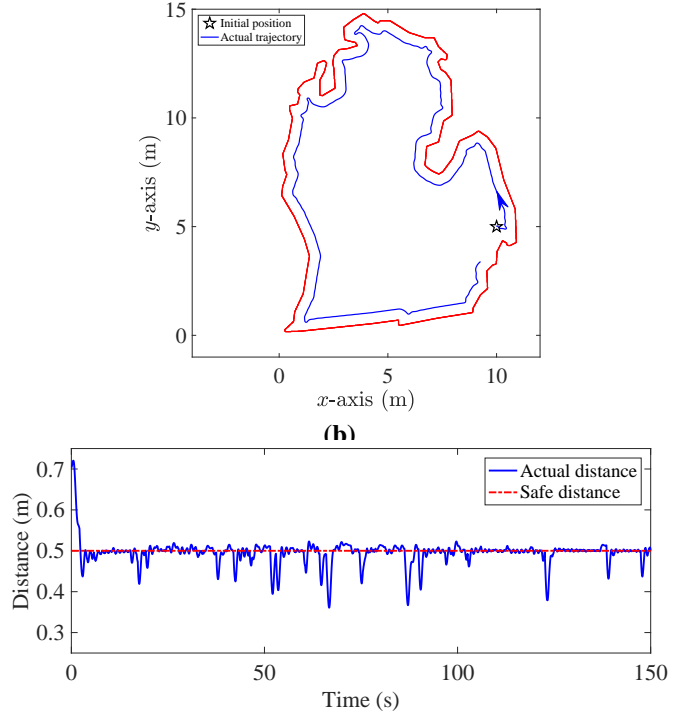


Fig. 9. (a) The robot performs border patrol with a constant linear velocity while maintaining a safe distance with the border shown in red. (b) Robot's distance from the border. The average robot's distance from the border is 0.49 m. In 93% of the time, the robot's distance remains above 0.45 m.

outside the scope of this work.

The geofencing scenario is shown in Fig. 8. Although the reference trajectory leaves the border, the robot maintains a safe distance at $d_o = 0.5$ m with the border. The linear speed is updated to keep up with the lead point of the reference trajectory.

D. Border Patrol

As shown in Fig. 9(a), the safety algorithm used for geofencing can handle border patrol applications, where the linear speed is fixed at 0.5π m/s. As shown in Fig. 9(b), the robot's distance is maintained very close to the safe distance, $\bar{d}_o = 0.5$ m. In fact, in 93% of the time, the safe distance remains above 0.45 m. Tighter control bounds can be achieved by recalibrating the safety control gains, which may cause control over-actuation. In the case studies, the safety algorithm uses counterclockwise evasive maneuvers. Similar results can be produced for clockwise safety control.

VIII. CONCLUSIONS

The proposed variable structure control successfully isolates environmental properties from the performance criteria. It is shown that the derived nominal safety model adequately handles various safety-critical scenarios such as collision avoidance and geofencing with minimal information from the environment. The control structure comprises trajectory tracking, safety control, and a supervisory logic. Moreover, the super-twisting algorithm not only guarantees stability and safety but also facilitates control implementation. The proposed method

dramatically simplifies the design steps and reduces the computation burden of the control compared to popular methods such as potential fields or path planning. Regardless of the number of obstacles or mobile robots, the algorithm's core components remain the same. Control bounds are obtained for the transformed systems, which facilitate adding features like integrator anti-windup to improve control performance. The results of four different case studies, conducted under realistic system constraints, verified the effectiveness of the proposed method. Future work extends the proposed method to unmanned aerial vehicles.

REFERENCES

- [1] T. Peng, L. Su, R. Zhang, Z. Guan, H. Zhao, Z. Qiu, C. Zong, and H. Xu, "A new safe lane-change trajectory model and collision avoidance control method for automatic driving vehicles," *Expert Systems with Applications*, vol. 141, p. 112953, 2020.
- [2] M. Mancini, N. Bloise, E. Capello, and E. Punta, "Sliding mode control techniques and artificial potential field for dynamic collision avoidance in rendezvous maneuvers," *IEEE Control Systems Letters*, vol. 4, no. 2, pp. 313–318, 2019.
- [3] H. Taghavifar, B. Xu, L. Taghavifar, and Y. Qin, "Optimal path-planning of nonholonomic terrain robots for dynamic obstacle avoidance using single-time velocity estimator and reinforcement learning approach," *IEEE Access*, vol. 7, pp. 159 347–159 356, 2019.
- [4] X. Liang, X. Qu, N. Wang, Y. Li, and R. Zhang, "Swarm control with collision avoidance for multiple underactuated surface vehicles," *Ocean Engineering*, vol. 191, p. 106516, 2019.
- [5] J. Ghommam, L. Iftikhar, and M. Saad, "Adaptive finite time path-following control of underactuated surface vehicle with collision avoidance," *Journal of Dynamic Systems, Measurement, and Control*, vol. 141, no. 12, 2019.
- [6] A. Behjat, S. Paul, and S. Chowdhury, "Learning reciprocal actions for cooperative collision avoidance in quadrotor unmanned aerial vehicles," *Robotics and Autonomous Systems*, vol. 121, p. 103270, 2019.
- [7] Z. Pan, D. Li, K. Yang, and H. Deng, "Multi-robot obstacle avoidance based on the improved artificial potential field and PID adaptive tracking control algorithm," *Robotica*, vol. 37, no. 11, pp. 1883–1903, 2019.
- [8] M. Karkoub, G. Atınc, D. Stipanovic, P. Voulgaris, and A. Hwang, "Trajectory tracking control of unicycle robots with collision avoidance and connectivity maintenance," *Journal of Intelligent & Robotic Systems*, vol. 96, no. 3–4, pp. 331–343, 2019.
- [9] W. H. Huang, B. R. Fajen, J. R. Fink, and W. H. Warren, "Visual navigation and obstacle avoidance using a steering potential function," *Robotics and Autonomous Systems*, vol. 54, no. 4, pp. 288–299, 2006.
- [10] K. Pathak and S. K. Agrawal, "An integrated path-planning and control approach for nonholonomic unicycles using switched local potentials," *IEEE Transactions on Robotics*, vol. 21, no. 6, pp. 1201–1208, 2005.
- [11] I. Mas and C. Kitts, "Obstacle avoidance policies for cluster space control of nonholonomic multirobot systems," *IEEE/ASME Transactions on Mechatronics*, vol. 17, no. 6, pp. 1068–1079, 2011.
- [12] E. J. Rodríguez-Seda, C. Tang, M. W. Spong, and D. M. Stipanović, "Trajectory tracking with collision avoidance for nonholonomic vehicles with acceleration constraints and limited sensing," *The International Journal of Robotics Research*, vol. 33, no. 12, pp. 1569–1592, 2014.
- [13] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *The International Journal of Robotics Research*, vol. 17, no. 7, pp. 760–772, 1998.
- [14] A. Chakravarthy and D. Ghose, "Obstacle avoidance in a dynamic environment: A collision cone approach," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 28, no. 5, pp. 562–574, 1998.
- [15] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics & Automation Magazine*, vol. 4, no. 1, pp. 23–33, 1997.
- [16] Z. Qu, J. Wang, and C. E. Plaisted, "A new analytical solution to mobile robot trajectory generation in the presence of moving obstacles," *IEEE Transactions on Robotics*, vol. 20, no. 6, pp. 978–993, 2004.
- [17] J. Alonso-Mora, P. Beardsley, and R. Siegwart, "Cooperative collision avoidance for nonholonomic robots," *IEEE Transactions on Robotics*, vol. 34, no. 2, pp. 404–420, 2018.
- [18] R. Fareh, M. Baziyad, M. H. Rahman, T. Rabie, and M. Bettayeb, "Investigating reduced path planning strategy for differential wheeled mobile robot," *Robotica*, vol. 38, no. 2, pp. 235–255, 2020.
- [19] K. Chu, M. Lee, and M. Sunwoo, "Local path planning for off-road autonomous driving with avoidance of static obstacles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 4, pp. 1599–1616, 2012.
- [20] F. Lamiraud, D. Bonnafous, and O. Lefebvre, "Reactive path deformation for nonholonomic mobile robots," *IEEE Transactions on Robotics*, vol. 20, no. 6, pp. 967–977, 2004.
- [21] A. W. Divelbiss and J. T. Wen, "A path space approach to nonholonomic motion planning in the presence of obstacles," *IEEE Transactions on Robotics and Automation*, vol. 13, no. 3, pp. 443–451, 1997.
- [22] P. Falcone, F. Borrelli, J. Asgari, H. E. Tseng, and D. Hrovat, "Predictive active steering control for autonomous vehicle systems," *IEEE Transactions on Control Systems Technology*, vol. 15, no. 3, pp. 566–580, 2007.
- [23] S. Li, Z. Li, Z. Yu, B. Zhang, and N. Zhang, "Dynamic trajectory planning and tracking for autonomous vehicle with obstacle avoidance based on model predictive control," *IEEE Access*, vol. 7, pp. 132 074–132 086, 2019.
- [24] B. Pinkovich, E. Rivlin, and H. Rotstein, "Predictive driving in an unstructured scenario using the bundle adjustment algorithm," *IEEE Transactions on Control Systems Technology*, 2020.
- [25] X. Zhang, A. Liniger, and F. Borrelli, "Optimization-based collision avoidance," *IEEE Transactions on Control Systems Technology*, 2020.
- [26] A. S. Matveev, H. Teimoori, and A. V. Savkin, "A method for guidance and control of an autonomous vehicle in problems of border patrolling and obstacle avoidance," *Automatica*, vol. 47, no. 3, pp. 515–524, 2011.
- [27] A. S. Matveev, M. C. Hoy, and A. V. Savkin, "The problem of boundary following by a unicycle-like robot with rigidly mounted sensors," *Robotics and Autonomous Systems*, vol. 61, no. 3, pp. 312–327, 2013.
- [28] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Autonomous robot vehicles*. Springer, 1986, pp. 396–404.
- [29] B. H. Krogh, "Guaranteed steering control," in *1985 American Control Conference*. IEEE, 1985, pp. 950–955.
- [30] G. Leitmann and J. Skowronski, "Avoidance control," *Journal of Optimization Theory and Applications*, vol. 23, no. 4, pp. 581–591, 1977.
- [31] R. Aggarwal and G. Leitmann, "Avoidance control," *Journal of Dynamic Systems, Measurement, and Control*, pp. 152–154, 1972.
- [32] J. Borenstein and Y. Koren, "Real-time obstacle avoidance for fast mobile robots," *IEEE Transactions on systems, Man, and Cybernetics*, vol. 19, no. 5, pp. 1179–1187, 1989.
- [33] H. Kong, F. He, X. Song, W. N. N. Hung, and M. Gu, "Exponential-condition-based barrier certificate generation for safety verification of hybrid systems," in *International Conference on Computer Aided Verification*, 2013.
- [34] M. Z. Romdlony and B. Jayawardhana, "Stabilization with guaranteed safety using Control Lyapunov-Barrier Function," *Automatica*, vol. 66, pp. 39–47, 2016.
- [35] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Transactions on Automatic Control*, vol. 62, pp. 3861–3876, 2017.
- [36] P. Glotfelter, I. Buckley, and M. Egerstedt, "Hybrid nonsmooth barrier functions with applications to provably safe and composable collision avoidance for robotic systems," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1303–1310, 2019.
- [37] A. Ghaffari, I. Abel, D. Ricketts, S. Lerner, and M. Krstic, "Safety verification using barrier certificates with application to double integrator with input saturation and zero-order hold," in *American Control Conference*, 2018.
- [38] A. Ghaffari, "Operational safety control for unmanned aerial vehicles using modular barrier functions," in *American Control Conference*, 2020.
- [39] L. Derafa, A. Benallegue, and L. Fridman, "Super twisting control algorithm for the attitude tracking of a four rotors UAV," *Journal of the Franklin Institute*, vol. 349, pp. 685–699, 2012.
- [40] J. A. Moreno, "A linear framework for the robust stability analysis of a generalized super-twisting algorithm," in *International Conference on Electrical Engineering, Computing Science and Automatic Control*, 2009, pp. 1–6.