This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2018.2869622, IEEE Transactions on Industrial Informatics

Vision based Guidance and Switching based Sliding Mode Controller for a Mobile Robot in the Cyber Physical Framework

Padmini Singh, Pooja Agrawal, Hamad Karki, Amit Shukla, *Member, IEEE,* Nishchal K. Verma, *Senior Member, IEEE,* and Laxmidhar Behera, *Senior Member, IEEE*

***Abstract*—This work proposes a vision based guidance strategy for safe navigation of a non-holonomic mobile robot in unknown indoor environments. The proposed switching based sliding mode control (SMC) law makes the robot follow the desired trajectory as given by the guidance law. The guidance strategy uses centroid of the depth map of an obstacle as obtained from the RGB-D sensor to generate the desired angular velocity. The fuzzy rule-based guidance is developed to generate desired linear velocity command. The analysis of guidance strategy is done for an infinite length obstacle. The proposed SMC is shown to be asymptotically stable using Krasovskii Method. The finite time convergence of robot navigation has been shown using Poincare Map method. The stability of the proposed SMC under burst losses has also been established. Experiments on the Pioneer P3-DX robot in different obstacle scenarios show that the robot safely navigates in presence of communication channel burst losses.**

***Index Terms*—Obstacle avoidance, Depth map, Centroid, Mobile robot, Sliding mode controller, Guidance strategy.**

I. Introduction

CYBER physical system (CPS) is a platform, which combines several features of computation, communication and control. Nowadays, researchers are solving most of the problems from cyber perspective [1] as recent application areas are based on the integration of all these three features of CPS.

The autonomous vehicle has gained popularity due to various real-time applications such as monitoring any task, obstacle avoidance, surveillance, border patrol, etc. Hence, a unified model is needed which overcomes the problem of interfacing within different domains. The effectiveness of a unified model depends on how much the interfacing of two domains is independent. One way of designing such model is the coupling of cyber parameters with the physical parameters. This is easier from the computational point of view red, but designing a controller for such complex system is difficult as all the parameters cannot be considered during design. For CPS based application, a single coupled model cannot give a direct or linear relationship between input and output of the system. Hence, the design of such CPS system is cumbersome.

Manuscript received August 27, 2017; revised December 31, 2017; May 19, 2018; July 9, 2018; accepted August 9, 2018.

Padmini Singh, Pooja Agrawal, Nishchal K. Verma and Laxmidhar Behera are with the Department of Electrical Engineering, Indian Institute of Technology, Kanpur, U.P., 208016, India (e-mail: padminis@iitk.ac.in; pooja.iitd@gmail.com; nishchal@iitk.ac.in; lbehera@iitk.ac.in)

Hamad Karki and Amit Shukla are with the Khalifa University Center for Autonomous Robotic Systems (KUCARS) of the Khalifa University of Science and Technology, Abu Dhabi. (e-mail: hamad.karki@ku.ac.ae; amit.shukla@ku.ac.ae)

From the CPS perspective, integrating autonomous mobile robots with communication domain requires the knowledge of hardware like a mobile robot, sensors etc. as well as software used for control and navigation. Lots of research works have been done for various mobile robot applications [2] [3] [4], which are based on potential field method, roadmap method, behaviour-based algorithm and control method for obstacle avoidance. Although some works have been done using artificial potential field approach [5] [6], they have the problems of multiple local minima and chattering. These problems can be solved using harmonic functions using non-gradient vector field approach [7] and a modified newton method [8]. In Roadmap approach, a guided path is generated over obstacle-free space [9] whereas in behaviour based approach, researchers fuse different behaviours for avoiding obstacles. Coordinating several behaviours efficiently is itself a challenge. In potential field and road-map method, one should have the prior knowledge of entire work space which requires lots of computation and thereby is not good for CPS environment. On the other hand behaviour based approach generally shows local convergence, which also requires lots of computation.

There are lots of vision based techniques have been proposed by the researchers [10] [11] [12] [13], as vision sensors provide rich information about the environment. These approaches are mainly based on optical flow and stereo vision [14]. Although both the techniques provide a depth map of the scene, they suffer from some limitations. Stereo vision is computationally expensive and requires more than one vision sensor, whereas optical flow based approaches are very sensitive to noise, and hence depth cannot be accurately measured.

To overcome the limitations of the above approaches, in this work, RGB-Depth sensor is used for obstacle avoidance. The advantage of using this sensor is that it provides accurate depth of the scene at a very fast rate [11]. This can help in designing a computationally efficient controller for achieving the guided path. Moreover, this sensor is capable of estimating the depth of the obstacle even in the night.

There may be disturbances and communication channel losses during the entire guidance operation. To achieve the guided path in finite time in such scenario an efficient and robust controller [15], [16] is required. In [17] and [18], switching signal and backstepping feedback linearization based approaches for following the guided path are presented,

1551-3203 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2018.2869622, IEEE Transactions on Industrial Informatics

respectively. An extended observer based approach for path tracking and obstacle avoidance is proposed in [19]. These approaches do not consider communication channel constraints. [20] proposes an optimal algorithm for the wireless network. The proposed approach handels the CPS issues with the help of RT-WMP protocol and robust sliding mode control technique.

Fig. 1 shows a CPS architecture for obstacle avoidance comprising physical systems, guidance and controller. Physical

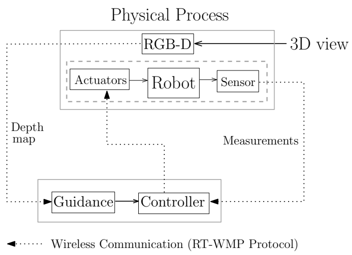


Fig. 1: CPS architecture for obstacle avoidance

systems include the robot, An RGB-D sensor, actuators and sensors. RGB-D sensor generates a depth map of the scene. The depth map is passed to the guidance block to generate desired commands to avoid the obstacles. These commands are given to the controller for the robot to follow the guided path.

The key contributions of the paper are summarized as follows:

1) Vision based guidance strategies as proposed in this paper has been validated for its effectiveness by finding out avoidance properties analytically in case of an infinite and finite length obstacle.
2) The switching based SMC consisting of two novels sliding surfaces as proposed in this paper ensures tracking errors in robot states to converge to zero simultaneously in the reaching phase only.
3) Further the finite time convergence using Poincare map method has also been established.
4) The stability of the proposed SMC in the presence of burst losses has been established.
5) Experiments on Pioneer P3-DX robot are conducted in different obstacle scenarios to validate the proposed approach in presence of communication channel losses.

The paper is organized as follows. Section II describes the problem and mathematical models. Section III discusses guidance strategies followed by their analysis of the guidance strategy in Section IV. The controller design is given in Section V. The stability analysis of designed controller is shown in Section VI. Simulations and experimental results are provided in Section VII and VIII, respectively. The conclusions are drawn in Section IX.

II. Problem Formulation

Fig. 2 shows a CPS model mainly consist of a mobile robot, RGB-D sensor and sliding mode controller communicating with each other through a wireless network. The objective of the work is to design a guidance law for robot safe navigation through indoor environment. And, a robust controller to achieve the desired command in finite time in the presence of random and burst losses. An RGB-D sensor is fixed with the robot which provides the depth map of the scene comes in camera field-of-view. Centroid of the depth map is used to generate the desired angular velocity command (ω_d). The depth of the pixel located at the image center is used to generate the desires linear velocity (v_d). The error between the commanded and robot actual states are passed to the controller for the robot to follow the desired commands. Here, switching based sliding mode controller is designed to achieve the desired commands.

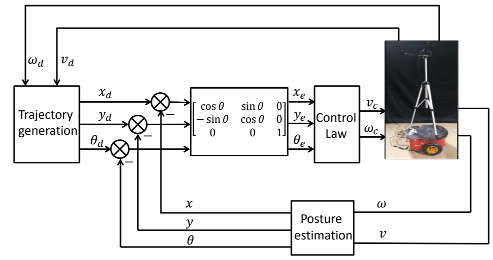


Fig. 2: System Overview

A. Robot Model

In this work, the kinematic model of the mobile robot is used as shown in Fig. 3. The mathematic model of the mobile robot is given by

$$\dot{x} = v\cos\theta, \quad \dot{y} = v\sin\theta, \quad \dot{\theta} = \omega \tag{1}$$

where, (x, y) and θ represent the position and orientation of the robot, respectively. Robot angular and linear velocities are

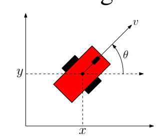


Fig. 3: Robot motion

represented by ω and v, respectively.

B. Communication Model

In this work, robot operating system (ROS) is used for experiments. Communication from a sensor to the controller, and controller to the robot is established by wireless channel, where communication medium uses RT-WMP protocol. The protocol is having some advanced features over IEEE802.11 and 802.11 RTS/CTS protocols, it maintains link quality between nodes by supporting message priorities, frame retransmission, frame duplication, offering better bandwidth and throughput to the nodes, delivering bounded and known worst case delayed messages, supports multihop communications to increase network coverage and has a built-in efficient error recovery mechanism that can recover from certain types of errors without jeoparding the behaviour [21] [22]. In the proposed work data computaion is taking place in two time scale as vision sensor data computation is too fast compared to actuator and controller data computation but the protocol handels this multi time scale data efficiently even in the presence of burst loss.

III. Guidance Strategies

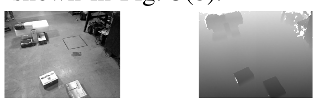
This section discusses the ground plan estimation followed by obstacle segmentation and proposed guidance laws.

1551-3203 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2018.2869622, IEEE Transactions on Industrial Informatics

A. Obstacle Segmentation

Figs. 4(a) and 4(b) show a grey-level image of a scene and corresponding depth map using a Kinect sensor. For segmentation, firstly, a depth map of the ground is subtracted from the scene depth map to remove the effect of ground plan. Fig. 5(a) shows the depth map of the ground using a Kinect sensor. Next, in the resultant depth map, the pixels with depth value more than a predefined threshold is defined as an obstacle and converted into white color. Remaining are considered as background and converted to black color. For avoidance, center portion of the segmented image's lower half is considered as shown in Fig. 5(b).



(a) (b)

Fig. 4: Images captured from Kinect: (a) a scene sample image (b) corresponding depth map

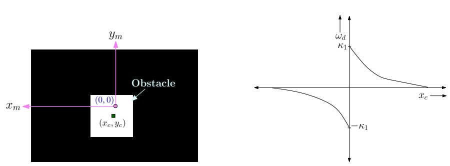
(a) (b)

Fig. 5: (a) Ground depth map (b) segmented image

B. Angular velocity guidance

The segmented image is used to compute the desired angular velocities, where pixels in white color are replaced by respective depth value. Centroid of the segmented image with obstacle depth is considered to design guidance law. Based on the heading direction, the robot has two possibilities as described in the next subsection.

1) Facing an obstacle: Robot faces an obstacle when intensity value of the obstacle depth image center pixel is not black as shown in Fig. 6a. Here, $(0,0)$ and (x_c, y_c) represent



(a) (b)

Fig. 6: Facing obstacle: (a) scenario (b) guidance command

the image center and centroid position, respectively. The value of y_c varies with the height distribution of the obstacle in the image plane, whereas x_c varies with the horizontal width distribution of the obstacle in the image plane. Hence, the ground vehicle can avoid the obstacle using x_c information in the image plane. Image frame x and y-axes are represented by x_m and y_m, respectively.

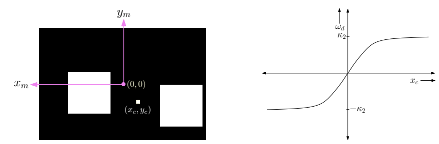
The desired value of the angular velocity (ω_d) for this scenario is obtained as

$$\omega_d = \kappa_1(\mathrm{sgn}(x_c) - \tanh(x_c)) \tag{2}$$

where, κ_1 is a proportionality constant. Fig. 6b shows the angular velocity variation with x_c. The figure shows that high value of guidance command will be applied when the image centroid x_c is close to the center.

2) Facing an open space: For this scenario, image center will have black pixel as shown in Fig. 7a. The angular velocity of the mobile robot is governed by

$$\omega_d = \kappa_2 \tanh(x_c) \tag{3}$$



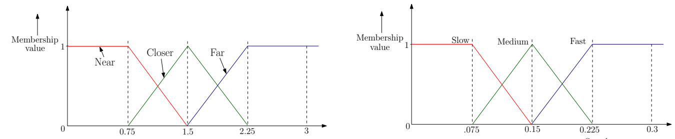
(a) (b)

Fig. 7: Facing open space: (a) scenario (b) guidance command

Fig. 7b shows the variation of guidance command with x_c. Low turn rate is applied for centroid closer to the image center, which is demanded as the vehicle has to pass between the obstacles. For the robot to pass through the passage. first, it is checked whether the separation between the obstacles is enough or not. If the width of the passage is safe for robot then desired command is computed using (3). Otherwise, guidance command is computed using (2).

C. Linear velocity guidance

To avoid sudden jerks, it is necessary to vary linear velocity to the distance of the obstacle. It is required that robot linear velocity should be slowed down when the angular velocity is high and vice-versa. Otherwise, wear and tear to the robot may happen. For achieving such a smooth linear velocity, a fuzzy guidance law based on the distance of the obstacle from the vehicle is presented. Although lots of work has been done using fuzzy logic guidance [23], [24], [25], in most of the cases data is coming from an ultrasonic sensor for detecting the obstacle position and depending upon the position of the obstacle rule base has been designed for angular and linear velocity guidance both. Obstacle avoidance laws using only fuzzy logic base requires numbers of rule base which increases the computational complexity. In the proposed work velocity guidance data is coming from vision sensor and input of the guidance is 'Range of image center depth'. Based on input range, three rule base to fire corresponding fuzzy output is created. Linear velocity is obtained by averaging the fired fuzzy output. From Fig. 8a, the range of the input is 0 to 3 m,



(a) (b)

Fig. 8: Fuzzy system: (a) input (b) output

where the range is equally divided into three parts. Depending upon the robot distance from the obstacle, any of the rule base can be fired. Table 1 shows three membership functions for input.

1551-3203 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2018.2869622, IEEE Transactions on Industrial Informatics

Fuzzy Input	Range (m)
Near	[0, 1.5]
Closer	[0.75, 2.25]
Far	[1.5, 3.0]

TABLE I

Fuzzy output	Range (m/s)
Slow	[0, 0.15]
Medium	[0.075, 0.225]
Fast	[0.15, 0.3]

TABLE II

Rule Base for fuzzy guidance

- **Rule1:** If Depth is near speed is slow.
- **Rule2:** If Depth is closer speed is medium.
- **Rule3:** If Depth is far speed is fast.

The range of output (linear velocity) is between 0 to 0.3 m/s. It is also divided into 3 equal parts. According to rules, fuzzy guidance selects the corresponding fuzzy output (membership value). Table 2 gives three membership functions for outputs.

Defuzzification is the process of finding a crisp output from corresponding fired rule-base membership value. In this paper, COG (Center of gravity) method for finding crisp output is used which is governed by the following formula.

$$v_d = \frac{\int \mu(v_f) v_f dv_f}{\int \mu(v_f) dv_f} \tag{4}$$

where, μ is membership value from input and v_f is the corresponding fuzzy value of the output. If obstacle is at 2 m distance from robot then rule 2 and 3 fires, and hence the corresponding crisp output value using (4) is 0.212 (m/s).

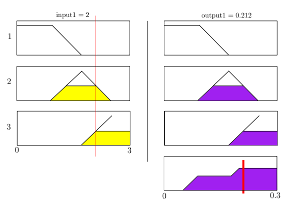


Fig. 9: Output of the fuzzy system.

IV. Analysis of Guidance Strategy for Infinite length Obstacle

Consider a robot with sensor facing an infinite length obstacle at initial heading θ as shown in Fig. 10. Point C represents the center of the depth image. Here, D shows a point located at the left of the image at an angle α_L. Depth of the point located at D for Kinect [26] can be written as

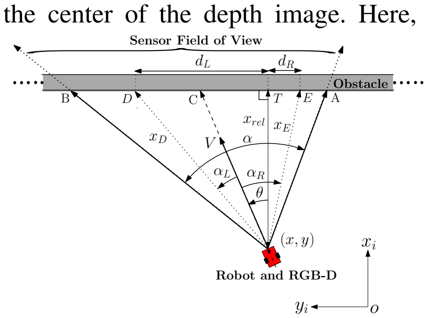


Fig. 10: A robot facing an infinite length obstacle.

$$x_D = \frac{x_o}{1 + \frac{x_o}{fb} d_L} \tag{5}$$

where, x_o is the distance of the reference pattern. The terms f, d_L, and b are infrared camera focal length, observed disparity in image plane and base length, respectively. From geometry, expression of d_L can be obtained as

$$d_L = x_{rel} \tan(\theta + \alpha_L) \tag{6}$$

where, x_{rel} represents the separation between the robot and the obstacle. Variation in x_{rel} and y_{rel} can be obtained as [27]

$$\dot{x}_{rel} = -\int v \cos\theta dt \tag{7}$$

$$\dot{y}_{rel} = -\int v \sin\theta dt \tag{8}$$

After substituting d_L from (6), (5) results in

$$x_D = \frac{x_o}{1 + \frac{x_o}{fb} x_{rel} \tan(\theta + \alpha_L)} \tag{9}$$

Similar to the point D, the depth of the point E located on the right side of the image at an angle α_R can be written as

$$x_E = \frac{x_o}{1 + \frac{x_o}{fb} x_{rel} \tan(\theta - \alpha_R)} \tag{10}$$

Centroid of the depth image can be computed as

$$x_c = \frac{\sum_{x_i} \sum_{y_i} x_i y_i De(x_i, y_i)}{\sum_{x_i} De(x_i, y_i)} \tag{11}$$

where, $De(x_i, y_i)$ represents the pixel depth located at the point (x_i, y_i). For analysis, single row is considered for centroid computation, so (11) can be written as

$$x_c = \frac{\sum_{x_i} x_i De(x_i, y_i)}{\sum_{x_i} De(x_i, y_i)} \tag{12}$$

As the image is split from the center, (12) can be written as

$$x_c = \frac{\int_0^{\delta_R} \int_0^{\delta_L} x^L x_D d\alpha_L d\alpha_R + \int_0^{\delta_R} \int_0^{\delta_L} x^R x_E d\alpha_L d\alpha_R}{\int_0^{\delta_R} \int_0^{\delta_L} x_D d\alpha_L d\alpha_R + \int_0^{\delta_R} \int_0^{\delta_L} x_E d\alpha_L d\alpha_R} \tag{13}$$

where, x^L and x^R represent the pixel position in the left and right halves of the image, respectively. The expressions for the x^L and x^R can be written as

$$x^L = f \tan \alpha_L \tag{14}$$

$$x^R = -f \tan \alpha_R \tag{15}$$

After substituting x_D, x_E, x^L and x^R from (9), (10), (14) and (15), respectively, (13) results in

$$x_c = \frac{f^2 b(\delta_L \ln|\cos\delta_R| - \delta_R \ln|\cos\delta_L|) + f x_{rel} x_0 (\ln|\cos\delta_R| \ln|\cos(\theta + \delta_L)| - \ln|\cos\delta_L| \ln|\cos(\theta - \delta_R)|)}{2fb\delta_L\delta_R + x_{rel} x_0 (\delta_L \ln|(\theta - \delta_R)| - \delta_R \ln|(\theta + \delta_L))} \tag{16}$$

As obstacle is infinite length, the values of δ_R and δ_L can be replaced by $\alpha/2$, where α represents the camera field of view. After substitution of x_D, x_E, x^L and x^R from (9), (10), (14) and (15) in (13), results in

$$x_c = -\frac{x_{rel} x_0 f \ln\left|\cos\frac{\alpha}{2}\right| \ln\left|\frac{\cos(\theta - \alpha/2)}{\cos(\theta + \alpha/2)}\right|}{0.5 f b \alpha^2 + x_0 x_{rel} \ln\left|\frac{\cos(\theta - \alpha/2)}{\cos(\theta + \alpha/2)}\right|} \tag{17}$$

1551-3203 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2018.2869622, IEEE Transactions on Industrial Informatics

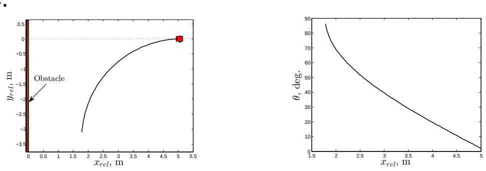
After substituting x_c from (17) in (2) to obtain desired value of the angular velocity for robot facing an obstacle is

$$\begin{aligned}\omega_d \;=\; & \kappa_1 \left(\mathrm{sgn}(x_c)\right.\\ & \left. + \tanh\left(\frac{x_{rel} x_0 f \ln\left|\cos\frac{\alpha}{2}\right| \ln\left|\frac{\cos(\theta-\alpha/2)}{\cos(\theta+\alpha/2)}\right|}{0.5 f b \alpha^2 + x_0 x_{rel} \ln\left|\frac{\cos(\theta-\alpha/2)}{\cos(\theta+\alpha/2)}\right|}\right)\right)\end{aligned} \tag{18}$$

Similarly, using (17) and (3), the desired value of the angular velocity for robot facing an open space is

$$\omega_d = -\kappa_2 \tanh\left(\frac{x_{rel} x_0 f \ln\left|\cos\frac{\alpha}{2}\right| \ln\left|\frac{\cos(\theta-\alpha/2)}{\cos(\theta+\alpha/2)}\right|}{0.5 f b \alpha^2 + x_0 x_{rel} \ln\left|\frac{\cos(\theta-\alpha/2)}{\cos(\theta+\alpha/2)}\right|}\right) \tag{19}$$

Consider a scenario, where the robot facing an infinite length obstacle from $(5, 0)$ m at initial heading $\theta_0 = 2°$. Velocity of the robot is assumed to be constant 0.3 m/s. The robot analytical trajectory using (2), (3), (18) and (19) are shown in Fig. 11a. The corresponding heading variations are shown in Fig. 11b. Results show that the proposed centroid strategy is able to navigate the robot safely by turning away from the obstacle.



(a) (b)

Fig. 11: Analytical results: (a) robot trajectory (b) corresponding heading profile

In case of finite length obstacle scenarios, the limits δ_L and δ_R are based on the obstacle portion covered by the left and right halves of the image, respectively. The limits can be given as [27]

$$\delta_L = \begin{cases} \frac{\alpha}{2} & \text{if } \delta \geq \frac{\alpha}{2}, \\ \delta & \text{if } 0 < \delta < \frac{\alpha}{2}, \\ 0 & \text{if } \delta \leq 0. \end{cases} \tag{20}$$

$$\delta_R = \begin{cases} \frac{\alpha}{2} & \text{if } \delta > 0, \\ \left(\frac{\alpha}{2} + \delta\right) & \text{if } -\frac{\alpha}{2} < \delta \leq 0, \\ 0 & \text{if } -\frac{\alpha}{2} \geq \delta. \end{cases} \tag{21}$$

where,

$$\delta = \theta_d - \theta, \tag{22}$$

$$\theta_d = \tan^{-1}\left(x_{rel}/y_{rel}\right) \tag{23}$$

V. SLIDING MODE CONTROLLER DESIGN

For achieving the guidance law in finite time, a novel switching based sliding mode controller is designed.

The reference trajectory is generated by the desired velocity model as follows,

$$\dot{x}_d = v_d \cos\theta_d, \quad \dot{y}_d \;= v_d \sin\theta_d, \quad \dot{\theta}_d = \omega_d \tag{24}$$

where, ω_d and v_d are computed as per (2), (3) and (4), respectively. Then, the error model of a non-holonomic mobile robot in body frame is,

$$\begin{bmatrix} x_e \\ y_e \\ \theta_e \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x - x_d \\ y - y_d \\ \theta - \theta_d \end{bmatrix} \tag{25}$$

where, x_e, y_e and θ_e are the difference between position and heading of the reference robot with the actual robot. Hence, the error dynamics can be derived as [28]

$$\begin{aligned} \dot{x}_e &= \omega y_e + v - v_d \cos\theta_e \\ \dot{y}_e &= -\omega x_e + v_d \sin\theta_e \\ \dot{\theta}_e &= \omega - \omega_d \end{aligned} \tag{26}$$

A. Control objective

The aim is to propose a control law that guarantees the tracking error achieves zero states in finite time

$$\lim_{t \to t_f} x_e = 0, \quad \lim_{t \to t_f} y_e = 0, \quad \lim_{t \to t_f} \theta_e = 0 \tag{27}$$

B. Design Steps

1) Sliding surface design: Designing of proper sliding surface is a major design problem.

Here, novel linear sliding surfaces are designed as a combination of error states i.e.:

$$s_1 = x_e - y_e, \quad s_2 \;= -x_e + y_e + \theta_e \tag{28}$$

Let us assume that the system is in sliding mode, i.e. $s_1 = 0$ and $s_2 = 0$ then

$$x_e = y_e \quad \text{and} \quad \theta_e = x_e - y_e = 0 \tag{29}$$

Remark: It is to be noted that simultaneously convergence of $s_1 = 0$ and $s_2 = 0$ does not imply that $x_e = y_e = 0$. To ensure the convergence of x_e and y_e it is required to find the convergence of a new variable $z_1 = x_e + y_e$, along with s_1 and s_2. This convergence is shown in Theorem 1.

2) Control law: In the second step, the aim is to design a finite time sliding mode controller. For this, the proposed control structure is:

$$v = v_{slide} + v_{reach}, \quad \omega \;= \omega_{slide} + \omega_{reach} \tag{30}$$

The derivative of the surfaces are

$$\dot{s}_1 = \dot{x}_e - \dot{y}_e, \quad \dot{s}_2 \;= -\dot{x}_e + \dot{y}_e + \dot{\theta}_e \tag{31}$$

Substituting all the values from eq. 26 in eq. 31, results in

$$\begin{aligned} \dot{s}_1 &= \omega y_e - v_d \cos\theta_e + v + \omega x_e - v_d \sin\theta_e \\ \dot{s}_2 &= -\omega y_e + v_d \cos\theta_e - \omega x_e - v + v_d \sin\theta_e + \omega - \omega_d \end{aligned} \tag{32}$$

Using the principle of invariance, v_{slide} and ω_{slide} are designed as

$$\begin{aligned} v_{slide} &= v_d \cos\theta_e + v_d \sin\theta_e - \omega y_e - \omega x_e \\ \omega_{slide} &= \omega_d \end{aligned} \tag{33}$$

The second components, v_{reach} and ω_{reach}, are reaching phase control inputs, which drives the error states on sliding surface in finite time. For diminishing the chattering, $\tanh$ function

1551-3203 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

is used. The proposed reaching law for linear and angular velocities are:

$$v_{reach} = -k_1 \tanh(s_1), \quad \omega_{reach} = -k_2 \tanh(s_2) \tag{34}$$

Combining (33) and (34) results in the following expressions of control law

$$\begin{aligned} v &= -\omega(x_e + y_e) + v_d \cos\theta_e + v_d \sin\theta_e - k_1 \tanh(s_1) \\ \omega &= \omega_d - k_2 \tanh(s_2) \end{aligned} \tag{35}$$

After simplifying (32), the reachability condition turned out to be

$$\dot{s}_1 = -k_1 \tanh(s_1), \quad \dot{s}_2 = k_1 \tanh(s_1) - k_2 \tanh(s_2) \tag{36}$$

Since $\dot{s}_2$ is a function of both s_1 and s_2, switching based convergence is happening. This is analyzed in the next section.

VI. Stability analysis

Lemma 1: Using Krasovskii method [29], for a continuous system $\dot{x} = f(x)$, $f(0) = 0$, $x \in R^n$, suppose there exist a Lyapunov function $V(x) = f^T(x)Pf(x)$, where P is a symmetric positive definite matrix, such that the following condition holds:

$$\dot{V}(x) = f(x)^T [PJ^T(x) + PJ(x)] f(x)$$

where, $[PJ^T(x) + PJ(x)] = -Q$, which is negative definite, then the system is asymptotically stable.

Theorem 1: Under conditions $4k_1k_2A_sD_s - k_1^2A_s > 0$ and $\sigma_1 <| k_2E_s + 2v_d |< \sigma_2$, error dynamics (26), sliding surfaces (28), and the control law given in (35), the asymptotic convergence of reachability law is guaranteed for any variable z_1, which is a function of x_e and y_e.

Proof: Let us introduce a variable z_1. The variables s_1, s_2 and z_1 in terms of x_e, y_e and θ_e are:

$$s_1 = x_e - y_e, \quad s_2 = -x_e + y_e + \theta_e, \quad z_1 = x_e + y_e \tag{37}$$

Using (37), x_e, y_e and θ_e in terms of s_1, s_2 and z_1 can be written as

$$x_e = \frac{z_1 + s_1}{2}, \quad y_e = \frac{z_1 - s_1}{2}, \quad \theta_e = s_1 + s_2 \tag{38}$$

The dynamics of z_1 can be given as:

$$\dot{z}_1 = \dot{x}_e + \dot{y}_e \tag{39}$$

Now, substituting the values of $\dot{x}_e$ and $\dot{y}_e$ from (26) in (39)

$$\dot{z}_1 = \omega y_e + v - v_d \cos\theta_e - \omega x_e + v_d \sin\theta_e \tag{40}$$

On putting the value of control law (35) in (40)

$$\dot{z}_1 = -2\omega_d x_e + 2k_2 \tanh(s_2) x_e + 2v_d \sin\theta_e - k_1 \tanh(s_1) \tag{41}$$

On further substitution, the values of x_e and θ_e from (38) in (41)

$$\begin{aligned} \dot{z}_1 &= -\omega_d(z_1 + s_1) + k_2(z_1 + s_1)\tanh(s_2) \\ &\quad + 2v_d \sin(s_1 + s_2) - k_1 \tanh(s_1) \end{aligned} \tag{42}$$

The objective is to find a Lyapunov function to ensure the convergence of s_1, s_2 and z_1. Using Krasovskii method, a Lyapunov function is considered as:

$$V(s) = f^T(s) P f(s) \tag{43}$$

where, $f(s) = [f_1(s) \quad f_2(s) \quad f_3(s)]^T$. Taking

$$f_1(s) = \dot{s}_1, \quad f_2(s) = \dot{s}_2, \quad f_3(s) = \dot{z}_1,$$

where, $\dot{s}_1 = -k_1 \tanh(s_1)$, $\dot{s}_2 = k_1 \tanh(s_1) - k_2 \tanh(s_2)$ and $\dot{z}_1 = -\omega_d(z_1 + s_1) + k_2(z_1 + s_1)\tanh(s_2) + 2v_d \sin(s_1 + s_2) - k_1 \tanh(s_1)$.

The derivative of (43) under the assumption $P = I$ (identity matrix) is,

$$\dot{V}(s) = f^T(s)[J^T(s) + J(s)] f(s)$$

where

$$J(s) = \begin{bmatrix} \frac{\partial f_1}{\partial s_1} & \frac{\partial f_1}{\partial s_2} & \frac{\partial f_1}{\partial z_1} \\ \frac{\partial f_2}{\partial s_1} & \frac{\partial f_2}{\partial s_2} & \frac{\partial f_2}{\partial z_1} \\ \frac{\partial f_3}{\partial s_1} & \frac{\partial f_3}{\partial s_2} & \frac{\partial f_3}{\partial z_1} \end{bmatrix}$$

$$= \begin{bmatrix} -k_1A_s & 0 & 0 \\ k_1A_s & -k_2D_s & 0 \\ \begin{smallmatrix} -\omega_d + k_2B_s \\ +2v_dC_s - k_1A_s \end{smallmatrix} & k_2E_sD_s + 2v_dC_s & -\omega_d + k_2B_s \end{bmatrix}$$

where, $A_s = \text{sech}^2(s_1)$, $B_s = \tanh(s_2)$, $C_s = \cos(s_1 + s_2)$, $D_s = \text{sech}^2(s_2)$, $E_s = (z_1 + s_1)$. Evaluating Q as:

$$Q = -[J^T(s)P + PJ(s)]$$

$$= \begin{bmatrix} 2k_1A_s & 0 - k_1A_s & \begin{smallmatrix} \omega_d - k_2B_s \\ -2v_dC_s + k_1A_s \end{smallmatrix} \\ -k_1A_s & 2k_2D_s & -k_2E_sD_s - 2v_dC_s \\ \begin{smallmatrix} \omega_d - k_2B_s \\ -2v_dC_s + k_1A_s \end{smallmatrix} & -k_2E_sD_s - 2v_dC_s & \omega_d - k_2B_s \end{bmatrix}$$

where, $A_s \epsilon [0 \;\; 1]$, $B_s \epsilon [-1 \;\; 1]$, $C_s \epsilon [-1 \;\; 1]$, $D_s \epsilon [0 \;\; 1]$, $\omega_d \epsilon [\underline{\omega}_d \;\; \bar{\omega}_d]$, and $\underline{\omega}_d$, $\bar{\omega}_d$ are respectively minimum and maximum value of angular velocity. For Q to be positive definite, the following conditions should be hold

$$\begin{gathered} 2k_1A_s > 0 \quad \text{and} \\ 4k_1k_2A_sD_s - k_1^2A_s > 0 \end{gathered} \tag{44}$$

$$\begin{aligned} (\omega_d - k_2B_s)[4k_1k_2A_sD_s - k_1^2A_s^2] - 2k_1A_s(k_2E_sD_s + 2v_dC_s)^2 \\ -2k_2D_s(\omega_d - k_2B_s - 2v_dC_s + k_1A_s)^2 \\ +2k_1A_s(k_2E_sD_s + 2v_dC_s)(\omega_d - k_2B_s - 2v_dC_s + k_1A_s) > 0 \end{aligned} \tag{45}$$

which holds for $\sigma_1 <| k_2E_s + 2v_d |< \sigma_2$. The detailed analysis is given in Appendix.

A. Finite time convergence of the state error

The finite time convergenge of the state error is obtained using Poincare map method. Using this method, the periodic orbit time of section $S = \{(s_1, s_2) \mid s_2 = 0\}$ is calculated with assumptions of '$\tanh_(s_1) \simeq \alpha$' and '$\tanh_(s_2) \simeq \beta$' as shown in Fig.12, where $\alpha \epsilon [-1 \;\; 1]$ and $\beta \epsilon [-1 \;\; 1]$.

1551-3203 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2018.2869622, IEEE Transactions on Industrial Informatics

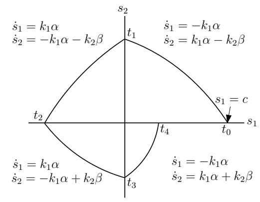


Fig. 12: Convergence of s_1, s_2 in finite time

Theorem 2: Under assumption $\| k_1\alpha \| > \| k_2\beta \|$ where values of α, β vary according to respective quadrant then control law (35) ensures the finite time convergence of state error.

Proof: Let at $t = t_0$, it is assumed that trajectory is at s_1 and having constant value c and s_2 is 0. In first quadrant $\dot{s_1} = -k_1\alpha$ and $\dot{s_2} = k_1\alpha - k_2\beta$. Hence, by solving first order differential equation in first quadrant time to reach surface s_2 is $t_1 = \frac{c}{k_1\alpha}$. In second quadrant $\dot{s_1} = k_1\alpha$ and $\dot{s_2} = -k_1\alpha - k_2\beta$. Then time to reach surface s_1 in negative direction is $t_2 = \frac{c[k_1\alpha - k_2\beta]}{(k_1\alpha)[k_1\alpha + k_2\beta]}$. In third quadrant $\dot{s_1} = k_1\alpha$ and $\dot{s_2} = -k_1\alpha + k_2\beta$. In this case, time to reach surface s_2 in negative direction is $t_3 = \frac{c[k_1\alpha - k_2\beta]}{(k_1\alpha)[k_1\alpha + k_2\beta]}$. Likewise, in fourth quadrant $\dot{s_1} = -k_1\alpha$ and $\dot{s_2} = k_1\alpha + k_2\beta$ and time to reach the surface s_1 in positive direction is $t_4 = \frac{c[k_1\alpha - k_2\beta]^2}{(k_1\alpha)[k_1\alpha + k_2\beta]^2}$. Now, the time period of the poincare section $S = \{(s_1, s_2) \mid s_2 = 0\}$ is $t_1 + t_2 + t_3 + t_4$, which is finite time. Hence, in light of lemma1 and Theorem2 the proposed SMC is finite time.

B. Stability under Burst Losses

Data communication is very important in CPS framework. Due to communication channel limitations, some signals may be lost during transmission. A bound on time interval for which system remain stable is analysed as follows:
The control signals v and ω at t^{th} time interval are:

$$\begin{aligned} v(t) &= -\omega(t)[x_e(t) + y_e(t)] + v_d(t)[\cos\theta_e(t) + \sin\theta_e(t)] \\ &\quad -k_1\tanh(s_1(t)) \end{aligned} \tag{46}$$

$$\omega(t) = \omega_d(t) - k_2\tanh(s_2(t)) \tag{47}$$

Consider Δt as the time duration in which control signal has not reached the actuator. This may occur due to packet losses or transmission delay. The actuators in that case will use the old control values. At time $t + \Delta t$ reachability condition will be:

$$\begin{aligned} \dot{s}_1(t+\Delta t) &= \omega(t)[x_e(t+\Delta t) + y_e(t+\Delta t) \\ &\quad -v_d(t+\Delta t)[\cos\theta_e(t+\Delta t) + \sin\theta_e(t+\Delta t)] + v(t) \end{aligned} \tag{48}$$

$$\begin{aligned} \dot{s}_2(t+\Delta t) &= -\omega(t)[x_e(t+\Delta t) + y_e(t+\Delta t) - v(t) + \omega(t) \\ &\quad +v_d(t+\Delta t)[\cos\theta_e(t+\Delta t) + \sin\theta_e(t+\Delta t)] - \omega_d(t+\Delta t) \end{aligned} \tag{49}$$

Substituting the value of $v(t)$ and $\omega(t)$ from (46) and (47) in (48) and (49) will result in

$$\begin{aligned} \dot{s}_1(t+\Delta t) &= \omega(t)[y_e(t+\Delta t) + x_e(t+\Delta t)] \\ &\quad -v_d(t+\Delta t)[\cos\theta_e(t+\Delta t) + \sin\theta_e(t+\Delta t)] - k_1\tanh(s_1(t)) \\ &\quad +v_d(t)[\cos\theta_e(t) + \sin\theta_e(t)] - \omega(t)[y_e(t) + x_e(t)] \end{aligned} \tag{50}$$

$$\dot{s}_1(t+\Delta t) = e_1(\Delta t) - k_1\tanh(s_1(t)) \tag{51}$$

where, $e_1(\Delta t) = \omega(t)[y_e(t+\Delta t) + x_e(t+\Delta t)] - v_d(t+\Delta t)[\cos\theta_e(t+\Delta t) + \sin\theta_e(t+\Delta t)] - \omega(t)[y_e(t) + x_e(t)] + v_d(t)[\cos\theta_e(t) + \sin\theta_e(t)]$ and

$$\begin{aligned} \dot{s}_2(t+\Delta t) &= -\omega(t)[y_e(t+\Delta t) + x_e(t+\Delta t)] \\ &\quad +v_d(t+\Delta t)[\cos\theta_e(t+\Delta t) + \sin\theta_e(t+\Delta t)] \\ &\quad +\omega(t)[y_e(t) + x_e(t)] - v_d(t)[\cos\theta_e(t) + \sin\theta_e(t)] \\ &\quad +k_1\tanh(s_1(t)) - \omega_d(t+\Delta t) + \omega_d(t)) - k_2\tanh(s_2(t)) \end{aligned} \tag{52}$$

$$\begin{aligned} \dot{s}_2(t+\Delta t) &= -e_1(\Delta t) + k_1\tanh(s_1(t)) + e_2(\Delta t) \\ &\quad -k_2\tanh(s_2(t)) \end{aligned} \tag{53}$$

where, $e_2(\Delta t) = -\omega_d(t+\Delta t) + \omega_d(t)$. According to reachability law of (36), the stable reachability conditions at time $'t + \Delta t'$ will be,

$$\dot{s}_1(t+\Delta t) \leq -k_1\tanh(s_1(t+\Delta t)) \tag{54}$$

$$\dot{s}_2(t+\Delta t) \leq k_1\tanh(s_1(t+\Delta t)) - k_2\tanh(s_2(t+\Delta t)) \tag{55}$$

From (51), (53), (54) and (55), the system stability should abide the following conditions:

$$e_1(\Delta t) \leq k_1[\tanh(s_1(t)) - \tanh(s_1(t+\Delta t))] \tag{56}$$

$$\begin{aligned} e_2(\Delta t) - e_1(\Delta t) &\leq k_1[\tanh(s_1(t+\Delta t)) - \tanh(s_1(t)] \\ &\quad +k_2[\tanh(s_2(t) - \tanh(s_2(t+\Delta t))] \end{aligned} \tag{57}$$

Hence, in time period of $'\Delta t'$ as long as the condition (56) and (57) are satisfied in presence of burst loss, system performance is stable.

VII. Simulation Results

In this section, simulations are carried out to show the effectiveness of the proposed controller only. For simulations, vision based guidance is not used to compute desired velocities as reference trajectories are predefined. Initial positions of the mobile robot are taken as $q(0) = [3, -3, -1]^T$ and initial velocities are $[v(0), \omega(0)] = [0, 0]^T$. The control parameters are set as $k_1 = 5$ and $k_2 = 2$. Following four cases are simulated, *case 1:* For setpoint stabilization $v_d = 0$, $\omega_d = 0$; *case 2:* For tracking a line $v_d = 2$, $\omega_d = 0$; *case 3:* For tracking a circle $v_d = 2$, $\omega_d = 1$; *case 4:* Tracking random path. Simulations results of cases 1, 2, 3 and 4 are shown in Figs. 13, 14, 15 and 16, respectively. Fig.16 results validate that proposed controller achieve the desired path in finite time compared to SMC in [30].

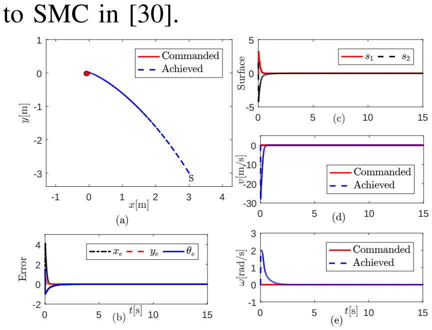


Fig. 13: Robot following a set point (Case 1): (a) trajectory (b) error profiles (c) surface profiles (d) linear velocity profiles (e) angular velocity profiles

1551-3203 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2018.2869622, IEEE Transactions on Industrial Informatics

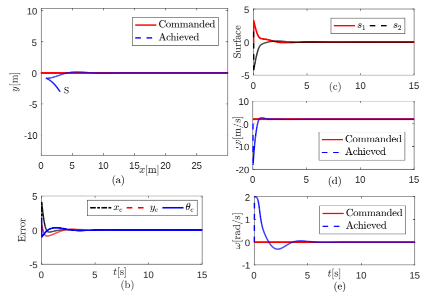


Fig. 14: Robot following a line (Case 2): (a) trajectory (b) error profiles (c) surface profiles (d) linear velocity profiles (e) angular velocity profiles

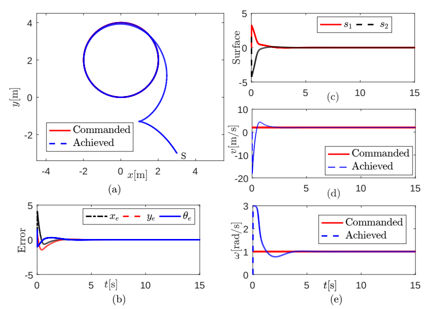


Fig. 15: Robot following a circle (Case 3): (a) trajectory (b) error profiles (c) surface profiles (d) linear velocity profiles (e) angular velocity profiles

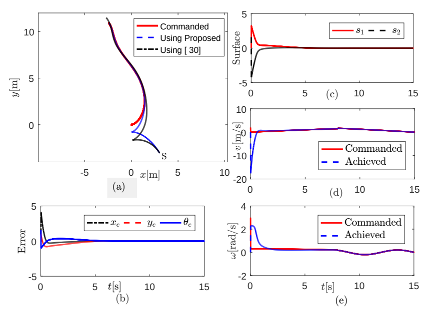


Fig. 16: Robot following random trajectory (Case 4): (a) trajectory (b) error profiles (c) surface profiles (d) linear velocity profiles (e) angular velocity profiles

VIII. Experimental setup and results

Experimental validation of the proposed approach for robot safe navigation is carried out in unknown indoor environments, where vision based guidance is used for computing the desired v_d and w_d as described in Section III. Fig. 17(a) shows the experimental setup, where Pioneer P3-DX is used to perform real-time experiments in Linux environment with Robot Operating System (ROS). Kinect V2 is equipped on top of the robot to generate the depth map. Note that, in this paper, it is assumed that coordinate frames of the robot and the camera are same. The sensor cannot give depth map for very close and far obstacles, in this work, a sensor is inclined at 30°. Frequency of the sensor is 30 Hz. Fig. 17(b) shows sample scenarios considered for experiments. Experiments are conducted in the absence and presence of channel losses. Two sample videos link are provided in [31] [32].

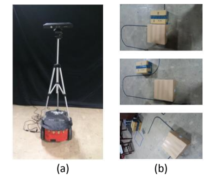


Fig. 17: (a) Experimental set-up (b) Sample scenarios

A. Without loss

In this section, an experiment is conducted in two obstacle scenario in the absence of channel losses. Fig. 18(a) shows the robot commanded and achieved navigation trajectories represented by solid and dashed lines, respectively. Corresponding commanded and actual linear and angular velocities are shown in Figs. 18(d) and 18(e), respectively. Results show that the robot is able to track the desired command computed by the guidance laws. Resultant position and angle error profiles are shown in Fig. 18(b). From the figure, it can be observed that all the errors converge to zero simultaneously and in finite time. Both the surface variations with time is shown in Fig. 18(c).

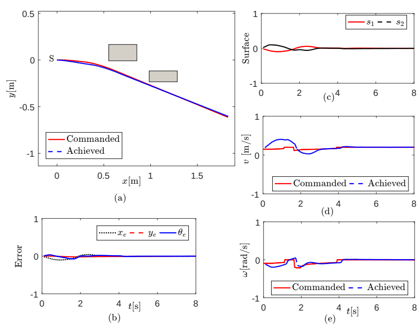


Fig. 18: Results in absence of losses: (a) trajectories (b) error profiles (c) surface profiles (d) linear velocity profiles (e) angular velocity profiles

B. With loss

Here, experimental results for two cases are discussed in presence of channel losses. In the first case, single obstacle scenario is considered with low values of losses. In the second case, burst losses are occurring in the channel as seen from the velocity profile of the robot. From trajectory tracking plots of Figs. 19 and 20, it is verified that sliding mode controller is robust under channel disturbances and velocity profile plot shows that after disturbance robot remains on the desired value.

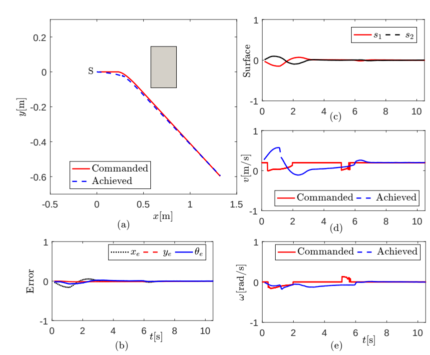


Fig. 19: Results in presence of losses for Case 1: (a) trajectories (b) error profiles (c) surface profiles (d) linear velocity profiles (e) angular velocity profiles

1551-3203 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2018.2869622, IEEE Transactions on Industrial Informatics

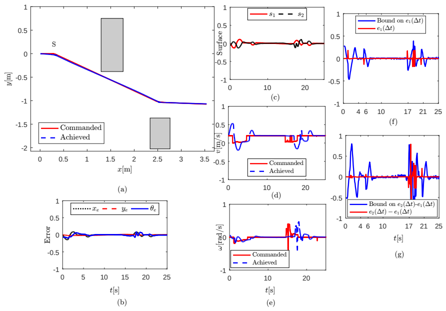


Fig. 20: Results in presence of losses for Case 2: (a) trajectories (b) error profiles (c) surface profiles (d) linear velocity profiles (e) angular velocity profiles (f) bound on $e_1(\Delta t)$ (g) bound on $e_2(\Delta t) - e_1(\Delta t)$

Figs. 20(f) and 20(g) show the boundedness of $e_1(\Delta t)$ and '$e_2(\Delta t) - e_1(\Delta t)$'. The bounds have been evaluated using right hand side of (56) and (57), respectively. The burst loss occurs in the duration of 4-6 seconds and 17-21 seconds. The system thus remains stable during these intervals as the bounds are not violated.

C. Comparative study

Here, proposed SMC is compared with an existing SMC [30]. Fig. 21 shows the experimental trajectories and the corresponding error profiles. The results show that the robot avoids the obstacle in finite time using the proposed approach. There are some recent advance works have been done on tracking of the non-holonomic mobile robot [33] [34] but the controller methodology proposed in this paper is better in the sense that it shows a finite-time convergence of the error as compared to asymptotic-convergence of the system error in all these works.

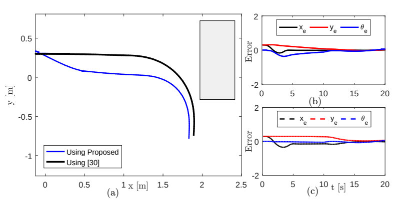


Fig. 21: Comparative result: (a) trajectories (b) error profile using proposed SMC (c) error profile of the existing SMC [30]

IX. Conclusion

This paper presents a vision based obstacle avoidance guidance law for a non-holonomic robot in unknown indoor environments. A vision based guidance strategy for avoiding any obstacle has been proposed and is validated analytically for an infinite length obstacle. The guidance law provides desired linear and angular velocities for the safe navigation of the mobile robot. A novel switching based sliding mode controller has been proposed to follow the guided path in finite time. The proof of the finite time convergence has been provided. The stability analysis of the proposed SMC in the presence of the burst losses has been established. Experiments are conducted using Pioneer P3-DX robot in a cyber physical environment while creating different obstacle scenarios within the laboratory. It has been shown that the proposed scheme is robust in the presence of burst losses. Apart from burst losses, a typical cyber physical framework would include delay, constraints over channel bandwidth and non-uniform sampling to reduce the communication burden. These issues will be covered as the future scope of the current work. This work can be extended for the outdoor applications as well with the proper choice of sensors, communication protocols and related processing.

X. Acknowledgement

This work has been supported by GRC, ADNOC, Abu Dhabi through the project EE/PI/2015449.

Appendix

Proof of Theorem 1: Infimum of first term of (45)

$$\inf[(\omega_d - k_2 B_s)(4k_1k_2A_sD_s - k_1^2A_s^2)]$$

$$\inf[4\omega_d k_1k_2A_sD_s - 4k_1k_2^2B_sA_sD_s - k_1^2A_s^2\omega_d + k_1^2k_2A_s^2B_s] \tag{58}$$

Taking infimum of first part of (58)

$$\inf(4\omega_d k_1 k_2 A_s D_s) = \begin{cases} -4 \mid \omega_d \mid k_1k_2 & \text{if } \omega_d < 0, \\ 0 & \text{if } \omega_d > 0. \end{cases} \tag{59}$$

Taking infimum of second part of (58)

$$-\sup(4k_1k_2^2B_sA_sD_s) = \begin{cases} -4k_1k_2^2 & \text{if } B_s > 0, \\ 0 & \text{if } B_s < 0. \end{cases} \tag{60}$$

Taking infimum of third part of (58)

$$-\sup(k_1^2A_s^2\omega_d) = \begin{cases} -k_1\bar{\omega}_d & \text{if } \bar{\omega}_d > 0, \\ 0 & \text{if } \bar{\omega}_d < 0. \end{cases} \tag{61}$$

Taking infimum of fourth part of (58)

$$\inf(k_1^2k_2A_s^2B_s) = \begin{cases} 0 & \text{if } B_s > 0, \\ -k_1^2k_2 & \text{if } B_s < 0. \end{cases} \tag{62}$$

The conditions obtained in (59), (60), (61) and (62) are summarised as follows:

$$-4 \mid \underline{\omega}_d \mid k_1k_2 - k_1^2k_2 \text{ if } \underline{\omega}_d < 0 \text{ and } B_s < 0,$$

-4$\mid \underline{\omega}_d \mid k_1k_2 + 4k_1k_2^2$if $\underline{\omega}_d < 0$ and $B_s > 0$,
$k_1 \mid \bar{\omega}_d \mid -k_1^2k_2$ if $\bar{\omega}_d > 0$ and $B_s < 0$,
$k_1 \mid \bar{\omega}_d \mid +4k_1k_2^2$ if $\bar{\omega}_d > 0$ and $B_s > 0$
Hence the infimum value of (58) is:

$$-4 \mid \underline{\omega_d} \mid k_1k_2 - k_1^2k_2 \tag{63}$$

Infimum of second term of (45) is

$$-\sup(2k_1A_s(k_2E_sD_s + 2v_dC_s)^2) = -2k_1\sup(k_2E_sD_s + 2v_dC_s)^2 \tag{64}$$

To obtain the value required in (64) it is neccessary to select the supremum value from the following conditions:

$$\inf \mid k_2E_sD_s + 2v_dC_s \mid = \begin{cases} \mid k_2E_s - 2v_d \mid & \text{if } E_s < 0, \\ \mid -2v_d \mid & \text{if } E_s > 0. \end{cases} \tag{65}$$

1551-3203 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2018.2869622, IEEE Transactions on Industrial Informatics

$$\sup \mid k_2 E_s D_s + 2v_d C_s \mid = \begin{cases} \mid 2v_d \mid & \text{if } E_s < 0, \\ \mid k_2 E_s + 2v_d \mid & \text{if } E_s > 0. \end{cases} \tag{66}$$

From conditions obtained in (65) and (66), the supremum value of (64) is:

$$\sup(2k_1 A_s (k_2 E_s D_s + 2v_d C_s)^2) = 2k_1 \mid k_2 E_s + 2v_d \mid^2 \tag{67}$$

Again infimum of third term of (45) is

$$\begin{aligned} &\sup(2k_2 D_s(\omega_d - k_2 B_s - 2v_d C_s + k_1 A_s)^2) \\ &= 2k_2 \sup(D_s(\omega_d - k_2 B_s - 2v_d C_s + k_1 A_s)^2) \end{aligned} \tag{68}$$

To obtain the value required in (68) it is neccessary to select the supremum value from the following conditions:

$$\inf \mid \omega_d - k_2 B_s - 2v_d C_s + k_1 A_s \mid = \mid \underline{\omega_d} - k_2 - 2v_d \mid \tag{69}$$

$$\sup \mid \omega_d - k_2 B_s - 2v_d C_s + k_1 A_s \mid = \mid \bar{\omega_d} + k_2 + 2v_d + k_1 \mid \tag{70}$$

Since the value obtained in (69) is less than the value obtained in (70), therefore the supremum value of (68) is

$$\begin{aligned} &\sup(2k_2 D_s(\omega_d - k_2 B_s - 2v_d C_s + k_1 A_s)^2) \\ &= 2k_2 \mid \bar{\omega}_d + k_2 + 2v_d + k_1 \mid^2 \end{aligned} \tag{71}$$

Infimum of last term of (45) can be obtained by the

$$\begin{aligned} &\inf(2k_1 A_s(\omega_d - k_2 B_s - 2v_d C_s + k_1 A_s) \\ &(k_2 E_s D_s + 2v_d C_s)) \\ &= -2k_1 A_s \sup \mid k_2 E_s D_s + 2v_d C_s \\ &\mid \sup \mid \bar{\omega_d} - k_2 B_s - 2v_d C_s + k_1 A_s \mid \end{aligned} \tag{72}$$

From (67) and (70) infimum value of (72) is:

$$-2k_1 \mid k_2 E_s + 2v_d \mid\mid \bar{\omega}_d + k_2 + 2v_d + k_1 \mid \tag{73}$$

Therefore from the infimum value obtained in (63), (67), (71) and (73), the condition for stability of (45) can also be written as:

$$\begin{aligned} &-4 \mid \bar{\omega}_d \mid k_1 k_2 - k_1^2 k_2 - 2k_1 (k_2 E_s + 2v_d)^2 \\ &-2k_2(\bar{\omega}_d + k_2 + 2v_d + k_1)^2 - 2k_1 \mid k_2 E_s + 2v_d \mid\mid \bar{\omega}_d \\ &+k_2 + 2v_d + k_1 \mid > 0 \end{aligned} \tag{74}$$

Dividing (74) by $-2k_1$

$$\begin{aligned} &(k_2 E_s + 2v_d)^2 + \frac{k_2(\bar{\omega}_d + k_2 + 2v_d + k_1)^2}{k_1} + \\ &\mid k_2 E_s + 2v_d \mid\mid \bar{\omega}_d + k_2 + 2v_d + k_1 \mid + 2 \mid \bar{\omega}_d \mid k_2 - \frac{k_1 k_2}{2} < 0 \end{aligned} \tag{75}$$

Let us consider variables X, σ_1 and σ_2 where, $X = \mid k_2 E_s + v_d \mid$ so that the condition obtained in (75) represents a quadratic inequality and the roots of the inequatily are σ_1 and σ_2. Since σ_1 and σ_2 are some functions of k_1 and k_2, therefore under condition X lies between σ_1 and σ_2 the system remains stable, i.e. if $\sigma_1 < \mid k_2 E_s + 2v_d \mid < \sigma_2$ (75) always holds true.

REFERENCES

[1] M. Kim, M.-O. Stehr, and C. Talcott, "A distributed logic for networked cyber-physical systems," *Science of Computer Programming*, vol. 78, no. 12, pp. 2453–2467, 2013.

[2] B.-F. Wu and C.-L. Jen, "Particle-filter-based radio localization for mobile robots in the environments with low-density wlan aps," *IEEE Trans. on Industrial Electronics*, vol. 61, no. 12, pp. 6860–6870, 2014.

[3] M. Zhang and H. H. Liu, "Game-theoretical persistent tracking of a moving target using a unicycle-type mobile vehicle," *IEEE Trans. on Industrial Electronics*, vol. 61, no. 11, pp. 6222–6233, 2014.

[4] E. DiGiampaolo and F. Martinelli, "Mobile robot localization using the phase of passive uhf rfid signals," *IEEE Trans. on Industrial Electronics*, vol. 61, no. 1, pp. 365–376, 2014.

[5] Y. Ma, G. Zheng, W. Perruquetti, and Z. Qiu, "Motion planning for non-holonomic mobile robots using the i-pid controller and potential field," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*. IEEE, 2014, pp. 3618–3623.

[6] B. Li, H. Du, and W. Li, "A potential field approach-based trajectory control for autonomous electric vehicles with in-wheel motors," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 8, pp. 2044–2055, 2017.

[7] D. Panagou, H. G. Tanner, and K. J. Kyriakopoulos, "Control of non-holonomic systems using reference vector fields," in *IEEE Conference on Decision and Control and European Control Conference*. IEEE, 2011, pp. 2831–2836.

[8] J. Ren, K. A. McIsaac, and R. V. Patel, "Modified newton's method applied to potential field-based navigation for nonholonomic robots in dynamic environments," *Robotica*, vol. 26, no. 01, pp. 117–127, 2008.

[9] S. M. LaValle and J. J. Kuffner Jr, "Randomized kinodynamic planning," *The International Journal of Robotics Research*, vol. 20, no. 5, pp. 378–400, 2001.

[10] M. Gupta, S. Kumar, L. Behera, and V. K. Subramanian, "A novel vision-based tracking algorithm for a human-following mobile robot," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 7, pp. 1415–1427, 2017.

[11] J. Biswas and M. Veloso, "Depth camera based indoor mobile robot localization and navigation," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. IEEE, 2012, pp. 1697–1702.

[12] N. A. Zainuddin, Y. Mustafah, Y. Shawgi, and N. Rashid, "Autonomous navigation of mobile robot using kinect sensor," in *Intl. Conf. on Computer and Communication Engineering*. IEEE, 2014, pp. 28–31.

[13] H. Emadi, T. Gao, and S. Bhattacharya, "Visibility-based target-tracking game: Bounds and tracking strategies," *IEEE Robotics and Automation Letters*, vol. 2, no. 4, pp. 1917–1924, 2017.

[14] K. McGuire, G. de Croon, C. De Wagter, K. Tuyls, and H. Kappen, "Efficient optical flow and stereo vision for velocity estimation and obstacle avoidance on an autonomous pocket drone," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 1070–1076, 2017.

[15] R. R. Nair, L. Behera, V. Kumar, and M. Jamshidi, "Multisatellite formation control for remote sensing applications using artificial potential field and adaptive fuzzy sliding mode control," *IEEE Systems Journal*, vol. 9, no. 2, pp. 508–518, 2015.

[16] P. M. Tiwari, S. Janardhanan, and M. un Nabi, "Rigid spacecraft attitude control using adaptive non-singular fast terminal sliding mode," *Journal of Control, Automation and Electrical Systems*, vol. 26, no. 2, pp. 115–124, 2015.

[17] J. Jin, Y. Kim, S. Wee, D. Lee, and N. Gans, "A stable switched-system approach to collision-free wheeled mobile robot navigation," *Journal of Intelligent & Robotic Systems*, vol. 86, no. 3-4, pp. 599–616, 2017.

[18] D. Chwa, "Robust distance-based tracking control of wheeled mobile robots using vision sensors in the presence of kinematic disturbances," *IEEE Trans. on Industrial Electronics*, vol. 63, no. 10, pp. 6172–6183, 2016.

[19] H. Yang, X. Fan, P. Shi, and C. Hua, "Nonlinear control for tracking and obstacle avoidance of a wheeled mobile robot with nonholonomic constraint," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 2, pp. 741–746, 2016.

[20] S. Manfredi and E. Di Tucci, "Decentralized control algorithm for fast monitoring and efficient energy consumption in energy harvesting wireless sensor networks," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 4, pp. 1513–1520, 2017.

[21] D. Tardioli and J. L. Villarroel, "Real time communications over 802.11: Rt-wmp," in *2007 IEEE Intl. Conf. on Mobile Adhoc and Sensor Systems*. IEEE, 2007, pp. 1–11.

1551-3203 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

[22] D. Tardioli, A. R. Mosteo, L. Riazuelo, J. L. Villarroel, and L. Montano, "Enforcing network connectivity in robot team missions," *The Intl. Journal of Robotics Research*, 2010.

[23] C.-S. Chiu, T.-S. Chiang, and Y.-T. Ye, "Fuzzy obstacle avoidance control of a two-wheeled mobile robot," in *Automatic Control Conference (CACS), 2015 International*. IEEE, 2015, pp. 1–6.

[24] B. Aissa, C. Fatima, and A. Yassine, "Data fusion strategy for the navigation of a mobile robot in an unknown environment using fuzzy logic control," in *Electrical Engineering-Boumerdes (ICEE-B), 2017 5th International Conference on*. IEEE, 2017, pp. 1–6.

[25] A. Nasrinahar and J. H. Chuah, "Effective route planning of a mobile robot for static and dynamic obstacles with fuzzy logic," in *Control System, Computing and Engineering (ICCSCE), 2016 6th IEEE International Conference on*. IEEE, 2016, pp. 34–38.

[26] K. Khoshelham, "Accuracy analysis of kinect depth data," *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 3812, pp. 133–138, Sep. 2011.

[27] P. Agrawal, A. Ratnoo, and D. Ghose, "Inverse optical flow based guidance for uav navigation through urban canyons," *Aerospace Science and Technology*, vol. 68, pp. 163–178, 2017.

[28] T. Fukao, H. Nakagawa, and N. Adachi, "Adaptive tracking control of a nonholonomic mobile robot," *IEEE transactions on Robotics and Automation*, vol. 16, no. 5, pp. 609–615, 2000.

[29] M. Gopal, *Digital Cont & State Var Met*. Tata McGraw-Hill Education, 2012.

[30] J. H. Lee, C. Lin, H. Lim, and J. M. Lee, "Sliding mode control for trajectory tracking of mobile robot in the rfid sensor space," *International Journal of Control, Automation and Systems*, vol. 7, no. 3, pp. 429–435, 2009.

[31] "Robot avoiding obstacle." [Online]. Available: https://youtu.be/rEgNO1yN2OE

[32] "Robot passing between the obstacle." [Online]. Available: https://youtu.be/msK7beCbkcc

[33] J. Mu, X.-G. Yan, S. K. Spurgeon, and Z. Mao, "Generalized regular form based smc for nonlinear systems with application to a wmr," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 8, pp. 6714–6723, 2017.

[34] W. Sun, S. Tang, H. Gao, and J. Zhao, "Two time-scale tracking control of nonholonomic wheeled mobile robots," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 6, pp. 2059–2069, 2016.

Padmini Singh received the Masters degree in electrical engineering from Indian Institute of Technology Delhi, Delhi, India, in 2010. She is currently working toward the Ph.D. degree in control and automation in the Department of Electrical Engineering, Indian Institute of Technology Kanpur, Kanpur India. Her primary research interests include cyber physical systems, sliding mode control and event trigger control.

Pooja Agrawal Pooja Agrawal received her Ph.D. degree in Aerospace Engineering from Indian Institute of Science, Bangalore, India, in 2017. She is presently working as a Post-Doctoral Fellow in the Department of Electrical Engineering, Indian Institute of Technology Kanpur, Kanpur, India. Her research interests include Guidance and Control, Motion Planning and Visual Servoing of Autonomous Vehicles.

Hamad Karki has done his Ph.D. from I Tokyo University of Technology, Japan with Robotics Group. At Tokyo University of Technology, his Ph.D. work was primarily focused on field robots which are basically tracked vehicles that have various off-road running features and applicable in rescue missions. His research experience expands across various domains of engineering with some of the key areas of interests are Robotics, Mechatronics and Control Systems.

Amit Shukla Amit Shukla is working at the Khalifa University Center for Autonomous Robotic Systems (KUCARS) of the Khalifa University of Science and Technology, Abu Dhabi. His research focus is on developing robotics and automation solutions with application of cutting edge Artificial Intelligence (AI) for various industries like Oil and Gas, Power, Manufacturing, Communication and other utility services. He did his B.Tech and M.tech from Indian Institute of Technology Kanpur (IITK), India, with specialization in the AI, robotics and control systems. He has also worked as executive engineer at aviation business of General Electric (GE) Bangalore, India. Later he obtained his PhD in control and optimization of Hybrid Electric Vehicle from Imperial College London, United Kingdom, with control systems group in year 2012. His research experience expands across various domains of engineering with some of the key areas of interests are Artificial Intelligence, Robotics, Machine Learning, Control Systems, Neural Networks, Autonomous Vehicles , Power Electronics and Energy Systems.

Nishchal Kumar Verma received the Ph.D. degree in electrical engineering from Indian Institute of Technology Delhi, New Delhi, India, in 2007. He is currently an Associate Professor in the Department of Electrical Engineering, Indian In- stitute of Technology Kanpur, Kanpur, India. His current research interests include the theory and applications of computational intelligence, big data analytics, Internet of Things, intelligent data mining algorithms, and computer vision

Laxmidhar Behera received the Ph.D. degree in electrical engineering from Indian Institute of Technology Delhi, New Delhi, India, in 1996. From 2000 to 2001, he was a Postdoctoral Fellow in the German National Research Center for Information Technology, GMD, Sank Augustin, Germany. He is currently a Professor in the Department of Electrical Engineering, Indian Institute of Technology Kanpur, Kanpur, India. His research interests include intelligent control, robotics, information retrieval, neural networks, cyber physical systems, and cognitive modeling.

1551-3203 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.