Filomat 32:5 (2018), 1547–1558
https://doi.org/10.2298/FIL1805547D

Published by Faculty of Sciences and Mathematics,
University of Niš, Serbia
Available at: http://www.pmf.ni.ac.rs/filomat

Multiple robots motion control to transport an object

Yanyan Dai[a], Dianwei Qian[b], SukGyu Lee*[a]

[a] *Yeungnam University*
[b] *North China Electric Power University*

Abstract. This paper presents transporting algorithm for multiple robots to transport a concave or convex object. The object transporting includes three processes: calculating proper points process; approaching proper points process; and transporting an object process. Using fuzzy sliding mode control algorithm, we design a kinematic velocity controller. We also propose a dynamic torque controller by adaptive sliding mode control algorithm. Finally, simulations and experiment show good performance of proposed methods.

1. Introduction

Object transportation has received considerable attention, over the past decade [1]. In [2], an object is pushed from a single rotational contact point. In [3], a robot pushes a disk shaped object. It allows compliance of the manipulated object against the obstacle. Kube et al. [4] presents swarm robots push a large box, and the system could be emergency. A great number of robots are applied to move an object [5]. Suarod et al. [6] described a heuristic method to calculate formation. Since the system should first calculate a series of candidate points, it increases computation time within the system. Considering pros and cons of above object transportation methods, this paper combines object pushing and object caging to transport concave and convex objects. In this paper, multiple robots transport a concave or convex object.

There are some method for robot motion control, such as EKF [7], optimal feedback control [8], robust control [9], intelligent control [10], decentralized control [11], and Sliding mode control. EKF requires a long time to update the desired states. Intelligent control requires high computation power. In robust control, a priori information about the bounds of the uncertain or time-varying parameters is necessary. About the drawback of decentralized control, it is difficult to coordinate overall system. Using fuzzy sliding mode control algorithm, the kinematic velocity controller is designed. Since it is difficult to produce the perfect velocity for robot dynamics as the kinematic controller, the torque controller is proposed, based on adaptive integral sliding mode control algorithm.

2010 *Mathematics Subject Classification*. 68T40.
Keywords. Mobile robots; Kinematic velocity controller; Dynamic torque controller; Fuzzy adaptive sliding mode control.
Received: 11 September 2017; Accepted: 11 November 2017
Communicated by Hari M. Srivastava

Email addresses: guangyanyan1129@hotmail.com (Yanyan Dai), dianwei.qian@ncepu.edu.cn (Dianwei Qian), sglee@ynu.ac.kr (SukGyu Lee*)

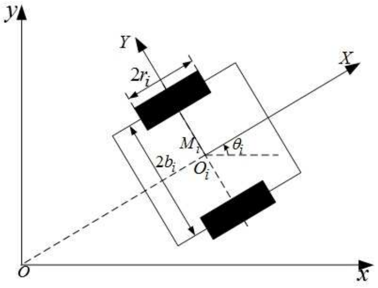

Figure 1: Model of nonholonomic mobile robot.

2. Robot model

In Fig. 1, robot R_i has a passive wheel and two actuated wheels to achieve the motion and orientation. r_i is the radius of wheels. $2b_i$ is the distance between two wheels. The mass center of the robot is M_i, and O_i is located in the middle point between two wheels. o, x, y is a global reference frame, and O_i, X, Y is the local reference frame. The motion of robot R_i is as $q_i = (x_i, y_i, \theta_i)^T$, where x_i, y_i, and θ_i are x_i coordinate, y_i coordinate, and the bearing angle, respectively. The constraint of the robot is as:

$$\dot{y}_i \cos\theta_i - \dot{x}_i \sin\theta_i = 0 \tag{1}$$

$V_i = (v_i, \omega_i)^T$ is defined as linear and angular velocity of robot . The surface friction is ignored. The model of the nonholonomic robot is defined as:

$$\dot{q}_i = S(q_i)V_i(t) \tag{2}$$

$$\bar{M}(q_i)\dot{V}_i + \bar{C}(q_i, \dot{q}_i)V_i + \bar{\tau}_{id} = \bar{B}\tau_i \tag{3}$$

where $s(q_i) = \begin{bmatrix} \cos\theta_i & 0 \\ \sin\theta_i & 0 \\ 0 & 1 \end{bmatrix}$, $\bar{M} = \begin{bmatrix} M_i & 0 \\ 0 & I_i \end{bmatrix}$, $\bar{C} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$, and $\bar{B} = \frac{1}{r}\begin{bmatrix} 1 & 0 \\ b & -b \end{bmatrix}$.

$\tau_i = (\tau_{i1}, \tau_{i2})^T$ is the torque on left and right wheels. m_i is the mass of the robot. I_i is the moment of inertia of the robot.

3. Object transporting

3.1. Calculating proper points process

In this section, robots calculate their proper points, based on the shape of an object. All robots are assumed to be the same size and model. R_{robot} is the radius of the robot. The robot can measure the distance and the direction toward the object. Then, the object corner vectors is calculated as $X_j = (x_j, y_j)$, where $j(j = 1, 2, ..., n)$ is the number of the corners of the object counter-clockwise, $X_1 = X_{n+1}$. The center vector $C_j(x_j^c, y_j^c)$ for the object edge between $X_j X_{j+1}$ can be given by Eqs. (4) and (5).The height from center point C_j of the object is defined as d_j.

$$x_j^c = \frac{1}{2}(x_j + x_{j+1}) \tag{4}$$

$$y_j^c = \frac{1}{2}(y_j + y_{j+1}) \tag{5}$$

The angle α_j is calculated as

$$\alpha_j = a\tan 2((y_{j+1} - y_j), (x_{j+1} - x_j)) \tag{6}$$

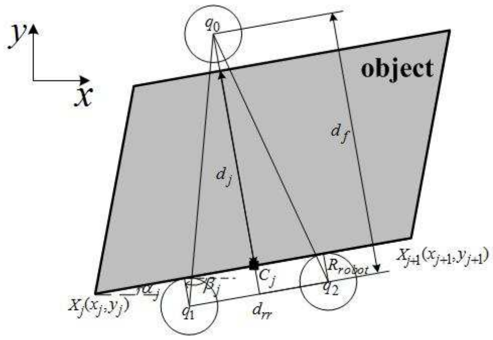

Figure 2: Three robots form a triangle transportation formation, where two follower robots contact the same edge of the object.

In Fig. 2, concave point is X_{j+2}, and the concave angle λ_{j+2} can be calculated as

$$\lambda_{j+2} = \alpha_{j+2} - (\alpha_{j+2} + \alpha_{j+1} - \pi)/2 \tag{7}$$

The distance connecting the corner point X_{j+2} and the center point of the robot can be calculated as

$$d^{XR}_{j+2} = R_{robot}/sin((\alpha_{j+2} + \alpha_{j+1} - \pi)/2) \tag{8}$$

If one edge is much longer than the other edges, two follower robots contact with the same long edge of the object. Three robots form an isosceles triangle formation as in Fig. 2. q_0 describes the proper point of leader robot. q_1 and q_2 denote proper points for follower robots 1 and 2.

Two follower robots are symmetrically located by the center point of the edge. The Euclidean distance between two follower robots' centers is defined as $k + 2R_{robot}$, where k is a positive constant. $\chi(m)(m = 1, 2)$ is the Euclidean distance between the contact point of one follower robot and edge center point. $\chi(m)$ is

$$\chi(m) = \begin{cases} -\frac{1}{2}d_{rr} & \text{m = 1} \\ \frac{1}{2}d_{rr} & \text{m = 2} \end{cases} \tag{9}$$

Based on (4)-(6), the contact point vector $P_m(x^p_m, y^p_m)$ ($m = 1, 2$) for the follower robot is:

$$x^p_m = x^c_j + \chi(m)\cos\alpha_j \tag{10}$$

$$y^p_m = y^c_j + \chi(m)\sin\alpha_j \tag{11}$$

The angle β_j is as Eq. (12). If the direction of β_j is obtained from the direction of α_j in counter-clock wise, $\beta_j = \alpha_j + \pi/2$; otherwise, $\beta_j = \alpha_j - \pi/2$.

$$\beta_j = \alpha_j \pm \pi/2 \tag{12}$$

The proper points q_m $(m = 1, 2)$ for two follower robots can be calculated as

$$x_m = x^p_m + R_{robot}\cos\beta_j \tag{13}$$

$$y_m = y^p_m + R_{robot}\sin\beta_j \tag{14}$$

Three robots form an isosceles triangle transportation formation. The height of the transportation formation d_f is calculated as Eq. (15), where δ is a compensation constant.

$$d_f = d_j + 2R_{robot} + \delta \tag{15}$$

The proper point q_0 for the leader robot is:

$$x_0 = \frac{1}{2}(x_1 + x_2) + d_f \cos(\alpha_j + \pi/2) \quad (16)$$

$$y_0 = \frac{1}{2}(y_1 + y_2) + d_f \sin(\alpha_j + \pi/2) \quad (17)$$

To extend the object transportation direction range, two follower robots contact with different edges of the object. If the concave object is a triangle shape object, two follower robots contact two centers of two adjacent edges of the object; otherwise, two follower robots contact two centers (C_j and C_{j+2}) of non-adjacent edges to form an isosceles triangle formation as in Fig. 3. q_0 describes the proper point of leader robot. q_1 and q_2 are proper points for follower robot 1 and 2, respectively. Two follower robots are positioned symmetrically the center of two different edges. In Fig. 3, two follower robots contact with C_j and C_{j+2}. The angle β_j is as Eq. (12). The proper point q_1 for follower robot R_1 can be calculated as

$$x_1 = x_j^c + R_{robot} \cos \beta_j \quad (18)$$

$$y_1 = y_j^c + R_{robot} \sin \beta_j \quad (19)$$

The proper point q_2 for follower robot R_2 is:

$$x_2 = x_{j+2}^c + R_{robot} \cos \beta_{j+2} \quad (20)$$

$$y_2 = y_{j+2}^c + R_{robot} \sin \beta_{j+2} \quad (21)$$

The angle φ_{12} from x-axis to the connecting line between follower robot 1 and 2 is given by

$$\varphi_{12} = a \tan 2((y_2 - y_1), (x_2 - x_1)) \quad (22)$$

The leader robot and two follower robots form a triangle transportation formation. The height d_f of the transportation formation is calculated as (23), where δ is a compensation constant.

$$d_f = d_{j+1} + 2R_{robot} + \delta \quad (23)$$

The proper point q_0 for leader robot is:

$$x_0 = \frac{1}{2}(x_1 + x_2) + d_f \cos(\varphi_{12} + \pi/2) \quad (24)$$

$$y_0 = \frac{1}{2}(y_1 + y_2) + d_f \sin(\varphi_{12} + \pi/2) \quad (25)$$

3.2. Approaching proper points process

To approach robots' proper points, safe paths are designed. The shortest safe paths are calculated, depending on the theta* algorithm [12]. As an obstacle, the object boundary is extended by robot radius.

3.3. Transporting an object process

If there is no obstacle, three robots form the transportation formation and move to the goal. If there is an obstacle, robots should avoid the obstacle first, and then move to their goals. During obstacle avoidance, the object and robots are recognized as a rigid body, which uses theta* algorithm to avoid the obstacle. According to the width of the rigid body, the boundary of the obstacle is extended.

4. Robot Motion Control

In this section, based on robots kinematic model (2), using sliding mode control algorithm, we first design a kinematic controller to bound the error posture $E_i = (x_{iE}, y_{iE}, \theta_{iE})^T$, and $\lim_{t\to\infty} \|(x_{iE}, y_{iE}, \theta_{iE})^T\| = 0$. Secondly, based on robots dynamic model (3), we use adaptive integral slid-ing mode control algorithm to design a torque controller. Finally, we present the structure of the presented control scheme.

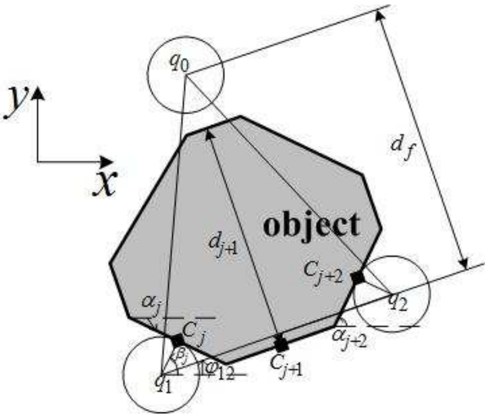

Figure 3: Three robots form a triangle transportation formation, where two follower robots contact different edges of the object.

4.1. Kinematic controller

Based on waypoint state $q_{iw} = (x_{iw}, y_{iw}, \theta_{iw})^T$ and current state $q_{ic} = (x_{ic}, y_{ic}, \theta_{ic})^T$, the state errors de-fined as $e_i = (x_{ie}, y_{ie}, \theta_{ie})^T = q_{iw} - q_{ic}$. Add a Jacobian, we transfer the state error e_i as E_i.

$$E_i = \begin{bmatrix} x_{iE} \\ y_{iE} \\ \theta_{iE} \end{bmatrix} = \begin{bmatrix} \cos\theta_{ic} & \sin\theta_{ic} & 0 \\ -\sin\theta_{ic} & \cos\theta_{ic} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{ie} \\ y_{ie} \\ \theta_{ie} \end{bmatrix} \tag{26}$$

Using Eq. (26) and $\dot{x}_{iw}\sin\theta_{iw} - \dot{y}_{iw}\cos\theta_{iw} = 0$, $\dot{E}_i$ is:

$$\dot{E}_i = \begin{bmatrix} \dot{x}_{iE} \\ \dot{y}_{iE} \\ \dot{\theta}_{iE} \end{bmatrix} = \begin{bmatrix} \cos\theta_{iE} \\ \sin\theta_{iE} \\ 0 \end{bmatrix} v_{iw} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \omega_{iw} + \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} v_{ic} + \begin{bmatrix} y_{iE} \\ -x_{iE} \\ -1 \end{bmatrix} \omega_{ic} \tag{27}$$

where v_{iw} and ω_{iw} are the reference linear and angular velocities.

According to the back-stepping algorithm, switching function is designed. When $x_{iE} = 0$ and assuming $\theta_{iE} = -\arctan(v_{iw}y_{iE})$, the Lyapunov function is

$$V_y = \frac{1}{2}y_{iE}^2 \tag{28}$$

The derivative of the Lyapunov function is

$$\dot{V}_y = y_{iE}\dot{y}_{iE} = y_{iE}(v_{iw}sin\theta_{iE} - x_{iE}\omega_i) = -x_{iE}y_{iE}\omega_i - y_{iE}v_{iw}sin(\arctan(v_{iw}y_{iE})) \tag{29}$$

Lemma: For any $x \in R$ and $|x| < \infty$, there is $\phi(x) = xsin(\arctan x) \geq 0$, iff $x = 0, ' =' $ is tenable.

Proof: when $x = 0$, then $\phi(0) = 0$; when $x \in (0, \infty)$, there is $\arctan x \in (0, \pi/2)$, then $\sin(\arctan x) > 0$, namely $\phi(x) > 0$; when $x \in (-\infty, 0)$, there is $\arctan x \in (-\pi/2, 0)$, then $\sin(\arctan x) < 0$, namely $\phi(x) > 0$.

Based on Lemma, $y_{iE}v_{iw}\sin(\arctan(v_{iw}y_{iE})) \geq 0$. When $y_{iE}v_{iw} = 0$, $y_{iE}v_{iw}\sin(\arctan(v_{iw}y_{iE})) = 0$. When $x_{iE} = 0$ and $\theta_{iE} = -\arctan(v_{iw}y_{iE})$, Eq. (29) is:

$$\dot{V}_y = -x_{iE}y_{iE}\omega_i - y_{iE}v_{iw}sin(\arctan(v_{iw}y_{iE})) \leq 0 \tag{30}$$

The switching function s_i for robot R_i is as:

$$s_i = \begin{bmatrix} s_{i1} \\ s_{i2} \end{bmatrix} = \begin{bmatrix} x_{iE} \\ \theta_{iE} + \arctan(v_{iw} y_{iE}) \end{bmatrix} \tag{31}$$

The controller is designed to converge s_{i1} and s_{i2} to zero. Then x_{iE} converges to zero, and θ_{iE} converges to $-\arctan(v_{iw} y_{iE})$. The system state y_{iE} converges to zero.

To make $s_i \dot{s}_i < 0$, $\dot{s}_i$ is defined as

$$\dot{s}_i = \begin{bmatrix} \dot{s}_{i1} \\ \dot{s}_{i2} \end{bmatrix} = \begin{bmatrix} -\varepsilon_{i1}\mathrm{sgn}(s_{i1}) - k_{i1}s_{i1} \\ -\varepsilon_{i2}\mathrm{sgn}(s_{i2}) - k_{i2}s_{i2} \end{bmatrix} \tag{32}$$

where ε_{i1}, ε_{i2}, k_{i1} and k_{i2} are positive constants. Setting $\alpha_i = \arctan(v_{iw} y_{iE})$, $\dot{s}_i$ for robot R_i is

$$\begin{aligned} \dot{s}_i &= \begin{bmatrix} -\varepsilon_{i1}\mathrm{sgn}(s_{i1}) - k_{i1}s_{i1} \\ -\varepsilon_{i2}\mathrm{sgn}(s_{i2}) - k_{i2}s_{i2} \end{bmatrix} = \begin{bmatrix} \dot{x}_{iE} \\ \dot{\theta}_{iE} + \frac{\partial \alpha_i}{\partial v_{iw}} \dot{v}_{iw} + \frac{\partial \alpha_i}{\partial y_{iE}} \dot{y}_{iE} \end{bmatrix} \\ &= \begin{bmatrix} v_{iw} \cos\theta_{iE} - v_i + y_{iE}\omega_i \\ \omega_{iw} - \omega_i + \frac{\partial \alpha_i}{\partial v_{iw}} \dot{v}_{iw} + \frac{\partial \alpha_i}{\partial y_{iE}} (v_{iw} \sin\theta_{iE} - x_{iE}\omega_i) \end{bmatrix} \end{aligned} \tag{33}$$

The kinematic controller for robot R_i is:

$$\begin{aligned} \begin{bmatrix} v_i \\ \omega_i \end{bmatrix} &= \begin{bmatrix} v_{iw} \cos\theta_{iE} + y_{iE}\omega_i - \dot{s}_{i1} \\ \dfrac{\omega_{iw} + \frac{\partial \alpha_i}{\partial v_{iw}} \dot{v}_{iw} + \frac{\partial \alpha_i}{\partial y_{iE}} (v_{iw} \sin\theta_{iE}) - \dot{s}_{i2}}{(1 + \frac{\partial \alpha_i}{y_{iE}} x_{iE})} \end{bmatrix} \\ &= \begin{bmatrix} v_{iw} \cos\theta_{iE} + y_{iE}\omega_i + \varepsilon_{i1}\mathrm{sgn}(s_{i1}) + k_{i1}s_{i1} \\ \dfrac{\omega_{iw} + \frac{\partial \alpha_i}{\partial v_{iw}} \dot{v}_{iw} + \frac{\partial \alpha_i}{\partial y_{iE}} (v_{iw} \sin\theta_{iE}) + \varepsilon_{i2}\mathrm{sgn}(s_{i2}) + k_{i2}s_{i2}}{(1 + \frac{\partial \alpha_i}{y_{iE}} x_{iE})} \end{bmatrix} \end{aligned} \tag{34}$$

where $\frac{\partial \alpha_i}{\partial v_{iw}} = \frac{y_{iE}}{1+(v_{iw} y_{iE})^2}$, and $\frac{\partial \alpha_i}{\partial v_{iw}} = \frac{v_{iw}}{1+(v_{iw} y_{iE})^2}$.

Proof: consider the Lyapunov function $V_1 = \frac{1}{2}s_{i1}^2 + \frac{1}{2}s_{i2}^2$. Differential V_1 about time t as

$$\dot{V}_1 = -\varepsilon_{i1} |s_{i1}| - k_{i1}s_{i1}^2 - \varepsilon_{i2} |s_{i2}| - k_{i2}s_{i2}^2 \leq 0 \tag{35}$$

When $x_{iE} = 0$ and $\theta_{iE} = -\arctan(v_{iw} y_{iE})$, $\dot{V}_1 = 0$. When $x_{iE} \neq 0$ and $\theta_{iE} \neq -\arctan(v_{iw} y_{iE})$, $\dot{V}_1 < 0$. The system is asymptotically stable in the sense of Lyapunov.

To weaken the charting, two fuzzy sliding mode controllers are designed. In controller 1, $|s_{i1}|$ and ε_{i1} are the input and output variables, respectively. h_{i1I} and h_{i1O} are input and output scaling. Define $|\bar{s}_{i1}| = h_{i1I} |s_{i1}|$ and $\bar{\varepsilon}_{i1} = h_{i1O}^{-1} \varepsilon_{i1}$.

Five fuzzy sets describe the fuzzy partitions, such as very big (VB), big (B), medium (M), small (S), and very small (VS). The triangular functions are applied for input variable and output variable, respectively. Fig. 4 shows the input and output membership functions. The fuzzy rules of the fuzzy sliding mode 1 are as: IF $\bar{s}_{i1}$ is VS, THEN $\bar{\varepsilon}_{i1}$ is VS.

The output is

$$\bar{\varepsilon}_{i1} = \frac{\sum_{g=1}^{5} \bar{\varepsilon}_{i1g} \mu(\bar{\varepsilon}_{i1g})}{\sum_{g=1}^{5} \mu(\bar{\varepsilon}_{i1g})} \tag{36}$$

Based on $\bar{\varepsilon}_{i1}$, ε_{i1} is used in Eq. (34) to obtain v_i. In controller 2, $|s_{i2}|$ and ε_{i2} are the input and output variables. ε_{i2} is used to obtain ω_i.

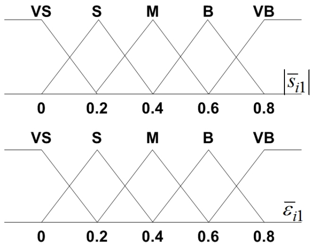

Figure 4: Triangular function.

4.2. Dynamic controller

Define the kinematic controller in Eq. (34) as $V_{id} = [v_{id}, \omega_{id}]^T$, and the velocity tracking error is as

$$V_{id} = V_{id} - V_{ic} = \begin{bmatrix} v_{id} - v_{ic} \\ \omega_{id} - \omega_{ic} \end{bmatrix} \tag{37}$$

The integral sliding surface is defined as

$$s_i = \begin{bmatrix} s_{i1} \\ s_{i2} \end{bmatrix} = V_{ie} + \eta \int_0^t V_{ie}(\tau) d\tau \tag{38}$$

where η is a positive integral constant.
Ignoring the unknown disturbance $\bar{\tau}_{id}$, based on (3), the following equation can be obtained:

$$\dot{V}_i = \bar{A}_i \tau_i = \bar{M}^{-1}(q_i)\bar{B}\tau_i = \begin{bmatrix} I_i & I_i \\ m_i b_i & -m_i b_i \end{bmatrix} \tag{39}$$

Adding (39), the derivative of (38) can be

$$\dot{s}_i(t) = \dot{V}_{ie}(t) + \eta V_{ie}(t) = \dot{V}_{id} - \dot{V}_{ic} + \eta V_{ie}(t) = \dot{V}_{id} - \bar{A}_i \tau_i + \eta V_{ie}(t) \tag{40}$$

The dynamic torque controller is designed as

$$\tau_i = \tau_{ieq} + \tau_{isw} = \bar{A}_i^{-1}[\dot{V}_{id} + \eta V_{ie} + K_i sgn(s_i)] \tag{41}$$

where τ_{ieq} is equivalent control. τ_{isw} is switching control, and is able to constraint unknown disturbance. $sgn(s_i) = \begin{bmatrix} sgn(s_{i1}) & sgn(s_{i2}) \end{bmatrix}^T$. K_i is a positive definite gain. Define $K_i = \hat{c}_i s_i sgn(s_i)$. $\hat{c}_i$ is the estimate of c_i, c_i is a positive constant, and $\tilde{c}_i = \hat{c}_i - c_i$.

Proof: consider the Lyapunov function $V_2 = \frac{1}{2} s_i^T s_i + \frac{\tilde{c}_i^2}{2\gamma_i}$. Differential V_2 about time t as

$$\begin{aligned} & \dot{V}_2 = s_i^T \dot{s}_i + \tfrac{\tilde{c}_i \dot{\tilde{c}}_i}{\gamma_i} \\ & = s_i^T(\dot{V}_{id} - \bar{A}_i \tau_i + \eta V_{ie}(t)) + \tfrac{\tilde{c}_i \dot{\tilde{c}}_i}{\gamma_i} \\ & = s_i^T(\dot{V}_{id} - \bar{A}_i \bar{A}_i^{-1}[\dot{V}_{id} + \eta V_{ie} + K_i sgn(s_i)] + \eta V_{ie}(t)) + \tfrac{\tilde{c}_i \dot{\tilde{c}}_i}{\gamma_i} \\ & = -s_i^T K_i sgn(s_i) + \tfrac{\tilde{c}_i \dot{\tilde{c}}_i}{\gamma_i} \\ & = -s_i^T \hat{c}_i s_i [sgn(s_i)]^2 + \tfrac{\tilde{c}_i \dot{\tilde{c}}_i}{\gamma_i} \\ & \leq -s_i^T \hat{c}_i s_i [sgn(s_i)]^2 - \tilde{c}_i(\|s_i\| - \tfrac{\dot{\tilde{c}}_i}{\gamma_i}) \end{aligned} \tag{42}$$

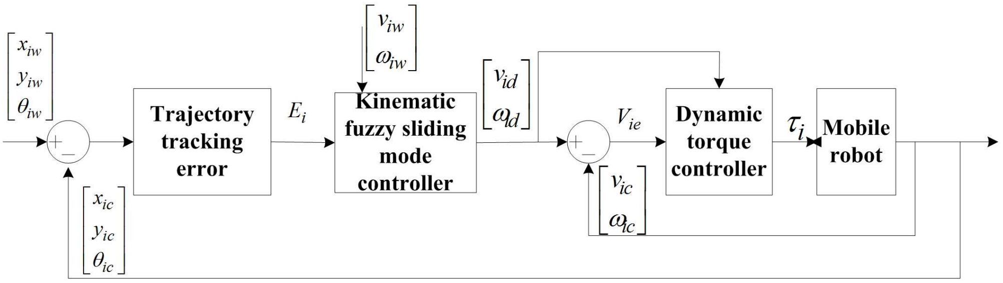
$[x_{iw}, y_{iw}, \theta_{iw}]$
Trajectory tracking error
E_i
$[v_{iw}, \omega_{iw}]$
Kinematic fuzzy sliding mode controller
$[v_{id}, \omega_{id}]$
V_{ie}
Dynamic torque controller
τ_i
Mobile robot
$[v_{ic}, \omega_{ic}]$
$[x_{ic}, y_{ic}, \theta_{ic}]$

Figure 5: Structure of the control scheme.

The parameter is selected as

$$c_i = \gamma_i \|s_i\| \tag{43}$$

Therefore, (42) can be rewritten as

$$\begin{aligned}&\dot{V}_2 \leq -s_i^T \hat{c}_i s_i [sgn(s_i)]^2 - \tilde{c}_i(\|s_i\| - \tfrac{\dot{\hat{c}}_i}{\gamma_i}) \\ &= -s_i^T \hat{c}_i s_i [sgn(s_i)]^2 \leq 0\end{aligned} \tag{44}$$

when $V_{ie} = 0$, $\dot{V}_2 = 0$. When $V_{ie} \neq 0$, $\dot{V}_2 < 0$. The system is asymptotically stable.

The structure of the presented control scheme is illustrated in Fig. 5. The waypoint state of the robot and the reference linear and angular velocities of the robot are system input. The system output is the configuration of the robot. Using the sliding mode control algorithm, the system calculates the kinematic velocity controller. Based on adaptive integral sliding mode control algorithm, the system calculates the dynamic torque controller.

5. Simulations

The physical parameters of the robots are as: $b_i = 0.1315m$, $r_i = 0.095m$, $m_i = 7.52kg$, $I_i = 2.325kgm^2$. Controller parameters are chosen as: $k_{i1} = k_{i2} = 0.0001$, $\varepsilon_{i1} = \varepsilon_{i2} = 0.001$, $\gamma_i = 0.01$. In the simulations, the time step is defined as: $dt = 0.1s$. The velocity constraints of the robot are set as: $v_{max} = 0.08(m/s)$, and $\omega_{max} = 0.62(rad/s)$.

Fig. 6 shows that three robots transport a convex object. The initial states of three robots are as follows: $(0.2m, 1.7m, 0)^T$, $(0.2m, 0.2m, 0)^T$, and $(0.2m, 1m, 0)^T$. Based on the calculating proper points process for a convex object, proper points for three robots are located at $(1.8m, 1.7m)$, $(1.2m, 0.2m)$, and $(1m, 1.1m)$. The object is regarded as an obstacle, three robots approach to their proper points, based on theta star algorithm. After approaching proper points, three robots maintain the transportation formation to transport the convex object. Leader robot's trajectory is calculated by Eqs. (45)-(47).

$$x_1(t+1) = x_1(t) + 0.04 \tag{45}$$

$$y_1(t+1) = y_1(t) + 0.04 \tag{46}$$

$$\theta_1(t+1) = a\tan 2((y_1(t+1) - y_1(t)), (x_1(t+1) - x_1(t))) \tag{47}$$

Figs. 7-9 show trajectory tracking errors of three robots, using the proposed control scheme.

Figs. 10-12 show velocity tracking errors of three robots, using the proposed control scheme.

This experiment shows that three robots can transport a concave object, avoiding two obstacles, as in Fig. 13. The initial states of three robots are as: $(0.2m, 1.7m, 0)^T$, $(0.2m, 0.2m, 0)^T$, and $(0.2m, 1m, 0)^T$. Based

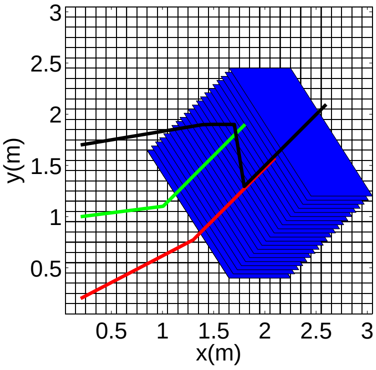

Figure 6: Three robots transport a convex object.

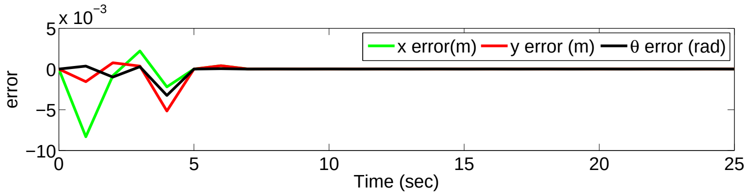

Figure 7: Leader robot's trajectory tracking errors.

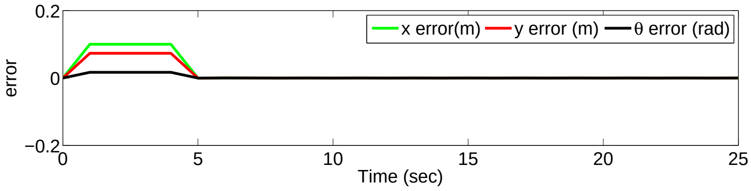

Figure 8: Follower robot 1's trajectory tracking errors.

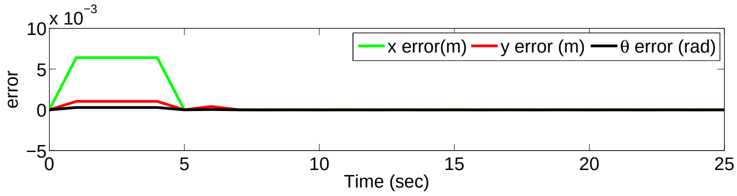

Figure 9: Follower robot 2's trajectory tracking errors.

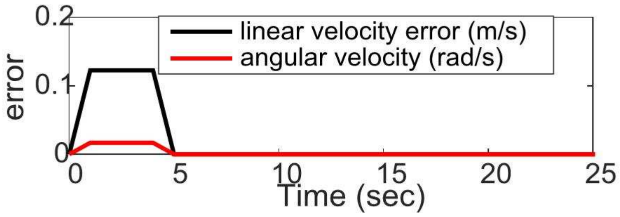

Figure 10: Leader robot's velocity tracking errors.

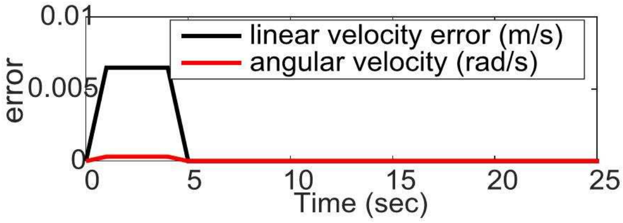

Figure 11: Follower robot 1's velocity tracking errors.

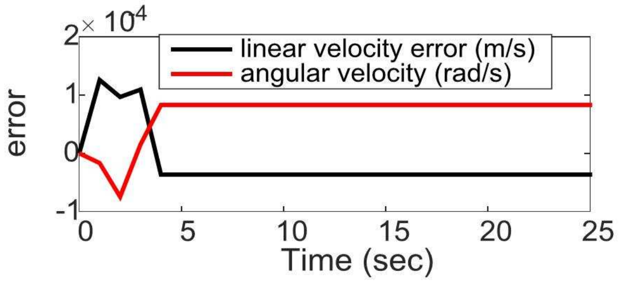

Figure 12: Follower robot 2's velocity tracking errors.

on the calculating proper points process for a concave object, proper points for three robots are located at $(1.9m, 1.8m)$, $(2.5m, 1m)$, and $(1.4m, 1m)$. The obstacles are on $(0.7m, 2.6m)$ and $(0.7m, 3.5m)$. The leader robot's trajectory is calculated by Eqs. (48)-(50).

$$x_1(t+1) = x_1(t) \tag{48}$$

$$y_1(t+1) = y_1(t) + 0.055 \tag{49}$$

$$\theta_1(t+1) = a\tan 2((y_1(t+1) - y_1(t)), (x_1(t+1) - x_1(t))) \tag{50}$$

6. Experiment

Three mobile robots are used to transport an object. The robot uses the odometry platform to localize itself. Parameters in the experiment are the same as in the simulation. Fig. 14 shows snapshots of three robots transport a convex object. At t=0, three robot form a line formation. At t=55s, three robots form a transportation formation. At t=99s, the object is transported to the goal point. Fig. 15 presents trajectories of three robots. Comparing the result in Fig. 6, three robots' trajectories are approximately same. The experiment video link is https://www.dropbox.com/s/4vo6crd0w4upq5x/video

7. Conclusion

In order to solve concave or convex object transportation problem, three transportation processes are designed. For robot motion control, depending on fuzzy sliding mode control algorithm, we designed a kinematic velocity controller; according to adaptive integral sliding control algorithm, we proposed a dynamic torque controller. The simulations and experiment show good performance of all methods.

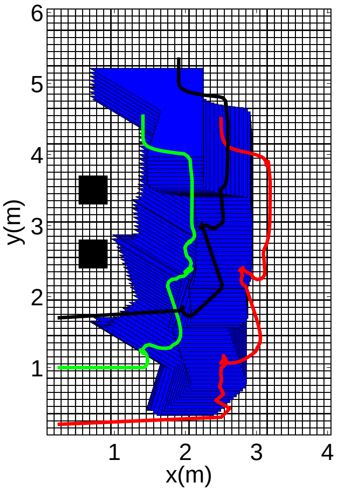

Figure 13: Three robots transport a concave object avoiding two obstacles.

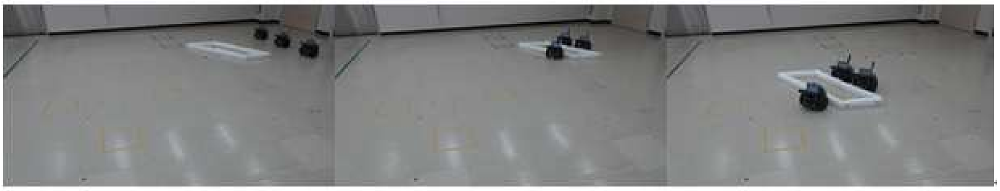

Figure 14: Snapshots of three robots (a) t=0; (b) t=55s; (c) t=99s (From left to right).

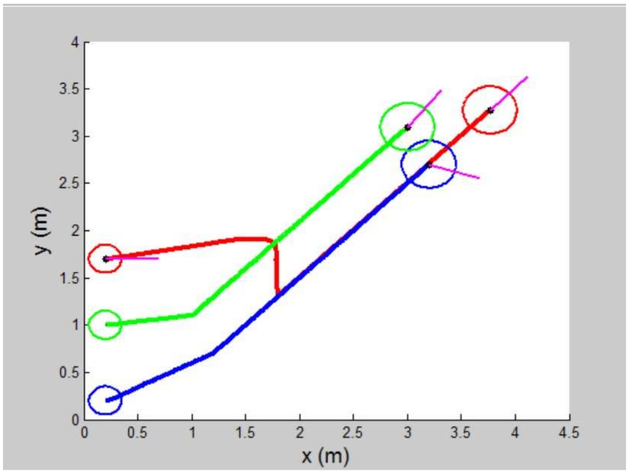

Figure 15: Three robots trajectories.

References

[1] Z. D. Wang and V. Kumar, Object Closure and Manipulation by Multiple Cooperating Mobile Robots, IEEE Int. Conf. on Robotics and Auto. (2002) 394–399.

[2] M. Salganicoff, G. Metta, A. Oddera, and G. Sandini, A Vision-Based Learning Method for Pushing Manipulation, In AAAI fall symposium series: Machine Learning in Vi-sion: What Why and How? (1993).

[3] D. Nieuwenhuisen, A. F. van der Stappen, and M. H. Overmars, Path Planning for Pushing a Disk using Com-pliance, In IEEE/RSJ IROS (2005) 4061–4067.

[4] C. R. Kube, and H. Zhang, The Use of Perceptual Cues in Multi-Robot Box-Pushing, In Proc. of the IEEE Int. Conf. on Robotics and Auto. (ICRA) (1996) 2085–2090.

[5] P. Sharma, A. Saxena, and A. Dutta, Optimal Arrest and Guidance of a moving Prismatic Object using Multiagents, Robotica (2007) 1–13.

[6] N. Suarod, N. Boonpinon, and A. Sudsang, A Heuristic Method for Computing Caging Formation of Polygonal Object,IEEE Int. Conf. on Robotics and Biomimetics (2007) 823–828.

[7] L. Teslic, L. Skrjanc, G. Klancar, Using a LRF Sensor in the Kalman-Filtering-Based Localization of a Mobile Ro-bot, ISA Transactions 49 (2010) 145–153.

[8] M. S. Miah, W. Gueaieb, Mobile Robot Trajectory Tracking Using Noisy RSS Measurements: An RFID Ap-proach, ISA Transactions 53 (2014) 433–443.

[9] J. Ghommam, H. Mehrjerdi, and M. Saad, Robust For-mation Control without Velocity Measurement of the Leader Robot, Control Engineering Practice 21 (2013) 1143–1156.

[10] Y. H. Chang, C. L. Chen, W. S. Chan, and H. W. Lin, Type-2 Fuzzy Formation Control for Collision-Free Multi-Robot Systems, Int. J. of Fuzzy Systems 15 (2013) 435–451.

[11] A. L. Yang, W. Naeem, G. W. Irwin, and K. Li, Stability Analysis and Implementation of a Decentralized Forma-tion Control Strategy for Unmanned Vehicles, IEEE Trans. on Control Systems Technology 22 (2014) 706–720.

[12] A. Nash, K. Daniel, S. Koenig, and A. Felner, Theta*: Any-Angle Path Planning on Grids, In Proceedings of the AAAI Conference on Artificial Intelligence (AAAI) (2007) 1177–1183.