2005 American Control Conference
June 8-10, 2005. Portland, OR, USA

Nonlinear Tracking Control for Nonholonomic Mobile Robots with Input Constraints: An Experimental Study

Wei Ren, Ji-Sang Sun, Randal W. Beard, Timothy W. McLain

Abstract— **This paper presents experimental results of nonlinear trajectory tracking controllers for nonholonomic mobile robots with fixed-wing UAV-like input constraints. Programmed to emulate an unmanned air vehicle flying at constant altitude, a nonholonomic mobile robot is assigned to follow a desired trajectory to transition through a sequence of targets in the presence of static and dynamic threats. Two velocity controllers with input constraints are proposed for tracking control. Hardware results using these two velocity controllers are compared to simulation results of dynamic controllers based on nonsmooth backstepping to demonstrate the effectiveness of our approach.**

I. Introduction

Control problems for wheeled mobile robots have been a topic of interest in recent years. One of the challenges inherent in these problems comes from the nonholonomic constraints, to which many wheeled robots are subjected. Another challenge results from actuator saturation constraints.

Previous approaches to control of mobile robots can be roughly categorized as either stabilization or tracking. Based on Brockett's necessary condition for feedback stabilization [1], nonholonomic systems cannot be stabilized via smooth time-invariant state feedback. The stabilization of mobile robots is mainly tackled via discontinuous feedback [2], [3] and continuous time-varying feedback [4], [5], [6]. The tracking control problem can be achieved via sliding mode approaches [7], backstepping techniques [8], [9], [10], or optimal control strategies [11].

Unmanned air vehicles (UAVs) equipped with low-level altitude-hold, velocity-hold, and heading-hold autopilots can be modeled by kinematic equations similar to those of mobile robots. The inherent properties of fixed-wing UAVs impose input constraints of positive minimum velocity due to the stall conditions of the aircraft, bounded maximum velocity, and saturated heading rate. However, most existing approaches to mobile robot tracking control are not directly applicable to the UAV problem since negative velocity commands are allowed in these approaches.

With mobile robots programmed to emulate UAVs flying at constant altitude, the main purpose of this paper is to demonstrate experimental results of the tracking controllers accounting for velocity and heading rate constraints similar to those of UAVs both as a proof of concept and as a step toward hardware tests on small fixed-wing UAVs. Although its ability to fully emulate a fixed-wing UAV is limited, a mobile robot platform provides an effective tool to pre-test algorithms designed for UAVs. From our past experience, the methodology of first testing some UAV algorithms on the mobile robot platform and then implementing them on UAV hardware not only helps us learn practical implementation issues but avoids damaging UAV crashes. The problem considered here is related to the simulation studies in [12], where a constrained control Lyapunov function approach is used to address the UAV trajectory tracking control problem. In this paper, we conduct experimental tests where a nonholonomic mobile robot is assigned to follow a desired trajectory to transition through several targets in the presence of static and dynamic threats. We present experimental results of two velocity controllers, where one is a saturation controller and the other is a discontinuous controller. These hardware results are also compared to simulation results of two dynamic controllers that are based on nonsmooth backstepping. One novel feature of the velocity controllers in this paper is that they are computationally simple and can be easily implemented with low-cost microcontrollers onboard UAVs. In addition, the controllers are shown to be robust to measurement noise. Finally, only piecewise continuity is required for the reference velocities while other approaches for mobile robot tracking control require uniform continuity (c.f. [8], [9], [10]). This feature makes the controllers feasible for our system, where a real-time trajectory generator outputs "bang-bang" control references to achieve time-extremal properties (c.f. [13]).

II. Experimental Setup

The experimental tests were conducted in the Multi-AGent Intelligent Coordination and Control (MAGICC) Laboratory at Brigham Young University.

A. Mobile Robot Testbed

The MAGICC Lab mobile robot testbed consists of a 5 m by 5 m field. Fig. 1 shows the schematic hardware/software structure of the testbed.

In our experiments, all high-level control algorithms including the trajectory tracker are performed on a host computer running Matlab/Simulink under the Linux operating system. An overhead camera capturing image frames at 30 frames per second is mounted on the ceiling directly above the testbed to measure the position and heading of each robot. The control algorithms utilize the MAGICC Mobile Robot Toolbox (MMRT) [14] that runs under Simulink.

W. Ren is with the Space Systems Laboratory, University of Maryland. weiren@ssl.umd.edu

R. W. Beard is with the ECE Department, Brigham Young University. J.-S. Sun and T. W. McLain are with the ME Department, Brigham Young University.

0-7803-9098-9/05/$25.00 ©2005 AACC

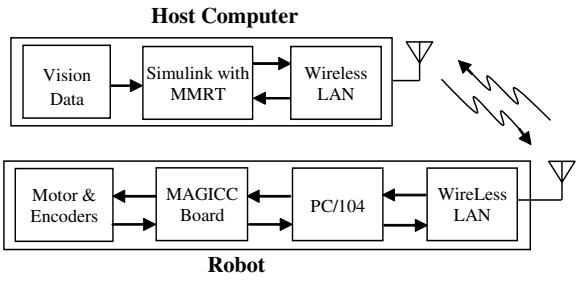

Fig. 1. Hardware/software structure of the mobile robot testbed.

MMRT provides a convenient interface for rapid implementation and testing of mobile robot control applications.

Fig. 2 shows the canister robot used in our experiments. Control commands from the host computer are sent to a PC/104 computer onboard the mobile robot over a wireless LAN. The PC/104 stack shown in Fig. 3 then sends motor commands to and receives encoder count updates from the MAGICC board. The MAGICC board, as shown in Fig. 4, is a re-configurable high performance hardware controller designed for use in light weight, compact robotics and control applications at BYU [15]. The MAGICC board combines a 29 MHZ Rabbit processor with 6 high current motor drivers, 4 hardware quadrature turn encoders, and 10 analog inputs. The MAGICC board produces PWM output to the motors and calculates the robot linear and angular velocities, which can then be sent to the host computer to estimate robot state information.

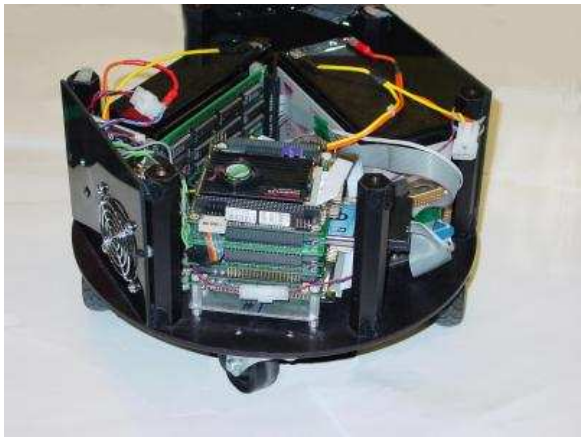

Fig. 2. Canister robot.

B. Software Architecture

Fig. 5 shows the software architecture implemented in our experiments. The architecture consists of five components: target manager (TM), waypoint path planner (WPP), waypoint manager and real-time trajectory generator (WM & TG), trajectory tracker (TT), and low-level robot control (LLC).

The top three components in Fig. 5 have been addressed in [16], [13], [17]. Next, we describe low-level robot

Fig. 3. PC/104 stack.

Fig. 4. The MAGICC board.

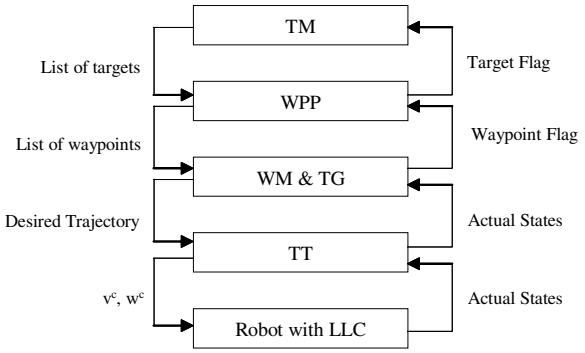

Fig. 5. Software architecture.

control. The trajectory tracker will be discussed in the next section.

Low-level control algorithms are implemented in the MAGICC board with the objective of maintaining commanded robot linear and angular velocities during the experiments. Fig. 6 shows a PID control loop for the commanded linear and angular velocities. Note that the trajectory tracker outputs the commanded linear and angular velocities v^c and ω^c. They are then converted to the commanded left and right wheel voltages denoted by V_l^c and V_r^c respectively via the conversion factor K_1. The actual left and right wheel voltages denoted V_l^a and V_r^a respectively are then converted back to the actual linear and angular velocities v^a and ω^a respectively via the conversion factor K_2.

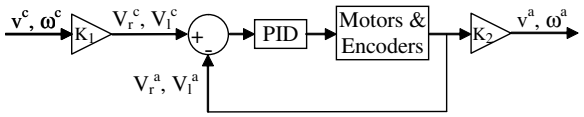

Fig. 6. PID control loop for v^c and ω^c.

III. Tracker Design

The kinematic equations of a nonholonomic mobile robot are given by

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

where (x, y) is the Cartesian position of the robot center, θ is the orientation, v is the linear velocity, and ω is the angular velocity. The simplified dynamic equations of motion are given by

$$m\dot{v} = F, \quad J\dot{\omega} = \tau, \tag{2}$$

where m is the mass, J is the mass moment of inertia, F is the force, and τ is the torque applied to the robot. Here friction effects have been neglected.

In order to simulate fixed-wing unmanned air vehicles flying at constant altitude, the following input constraints are imposed on the robot:

$$0 < v_{min} \le v \le v_{max}, \quad -\omega_{max} \le \omega \le \omega_{max}, \tag{3}$$

where $\omega_{max} > 0$.

In this section, the desired reference trajectory $(x_r, y_r, \theta_r, v_r, \omega_r)$ generated by the trajectory generator satisfies

$$\dot{x}_r = v_r \cos(\theta_r), \quad \dot{y}_r = v_r \sin(\theta_r), \quad \dot{\theta}_r = \omega_r,$$

where v_r and ω_r are piecewise continuous and satisfy $\inf_{t\ge 0} v_r(t) > v_{min}$, $\sup_{t\ge 0} v_r(t) < v_{max}$, and $\sup_{t\ge 0} |\omega_r(t)| < \omega_{max}$.

Without loss of generality, the constraints for v_r and ω_r can be written as

$$\begin{aligned} v_{min} + \epsilon_{v1} \le & v_r \le v_{max} - \epsilon_{v2} \\ -\omega_{max} + \epsilon_{\omega 1} \le & \omega_r \le \omega_{max} - \epsilon_{\omega 2}, \end{aligned} \tag{4}$$

where ϵ_{v1}, ϵ_{v2}, $\epsilon_{\omega 1}$, and $\epsilon_{\omega 2}$ are positive control parameters.

With regard to the kinematic model (1), the control objective is to find feasible inputs v and ω such that $|x_r - x| + |y_r - y| + |\theta_r - \theta| \to 0$ as $t \to \infty$.

Transforming the tracking errors expressed in the inertial frame to the robot frame, the error coordinates [18] become

$$\begin{bmatrix} x_e \\ y_e \\ \theta_e \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r - x \\ y_r - y \\ \theta_r - \theta \end{bmatrix}. \tag{5}$$

Accordingly, the tracking error model can be represented as

$$\begin{aligned} \dot{x}_e &= \omega y_e - v + v_r \cos(\theta_e) \\ \dot{y}_e &= -\omega x_e + v_r \sin(\theta_e) \\ \dot{\theta}_e &= \omega_r - \omega. \end{aligned} \tag{6}$$

Following [10], Eq. (6) can be simplified as

$$\dot{x}_0 = u_0 \tag{7a}$$

$$\dot{x}_1 = (\omega_r - u_0) x_2 + v_r \sin(x_0) \tag{7b}$$

$$\dot{x}_2 = -(\omega_r - u_0) x_1 + u_1, \tag{7c}$$

where

$$(x_0, x_1, x_2) \triangleq (\theta_e, y_e, -x_e) \tag{8}$$

and $u_0 \triangleq \omega_r - \omega$ and $u_1 \triangleq v - v_r \cos(x_0)$.

The input constraints under the transformation become

$$\underline{\omega} \le u_0 \le \bar{\omega}, \quad \underline{v} \le u_1 \le \bar{v}, \tag{9}$$

where $\underline{\omega} \triangleq \omega_r - \omega_{max}$, $\bar{\omega} \triangleq \omega_r + \omega_{max}$, $\underline{v} \triangleq v_{min} - v_r \cos(x_0)$, and $\bar{v} \triangleq v_{max} - v_r \cos(x_0)$ are time-varying due to state dependence and time-varying properties of v_r and ω_r.

Note from Eq. (7) that x_1 is not directly controllable when both x_0 and x_2 go to zero. To avoid this situation we introduce another change of variables.

Let

$$\bar{x}_0 = \lambda x_0 + \frac{x_1}{\sqrt{x_1^2 + x_2^2 + 1}}, \tag{10}$$

where $\lambda > 0$. Accordingly, $x_0 = \frac{1}{\lambda}(\bar{x}_0 - \frac{x_1}{\sqrt{x_1^2+x_2^2+1}})$. Obviously, $(\bar{x}_0, x_1, x_2) = (0,0,0)$ is equivalent to $(x_0, x_1, x_2) = (0,0,0)$, which is in turn equivalent to $(x_e, y_e, \theta_e) = (0,0,0)$ and $(x_r, y_r, \theta_r) = (x, y, \theta)$ since Eqs. (5) and (8) are invertible transformations. Therefore, the original tracking control objective is converted to a stabilization objective, that is, it is sufficient to find feasible control inputs u_0 and u_1 to stabilize $\bar{x}_0$, x_1, and x_2. With the same input constraints (9), Eq. (7a) can be rewritten as

$$\begin{aligned} \dot{\bar{x}}_0 = & \frac{1 + x_2^2}{\sqrt{x_1^2 + x_2^2 + 1}^3} v_r \sin(x_0) + \frac{x_2}{\sqrt{x_1^2 + x_2^2 + 1}} \omega_r \\ & + (\lambda - \frac{x_2}{\sqrt{x_1^2 + x_2^2 + 1}}) u_0 - \frac{x_1 x_2}{\sqrt{x_1^2 + x_2^2 + 1}^3} u_1. \end{aligned} \tag{11}$$

In the sequel, we first design velocity controllers based on the kinematic model (1), where a saturation controller and a discontinuous controller will be given. Then we apply the nonsmooth backstepping approach proposed in [19] to design force and torque controllers based on the dynamic model (2) for comparison purposes.

Let $\chi = [\bar{x}_0, x_1, x_2]^T$. In [12] we have shown that for $k > \frac{1}{2}$ and $\lambda > \kappa$ in Eq. (10), where κ is a positive constant expressed precisely in [12], $V_0(\chi) = \sqrt{\bar{x}_0^2 + 1} + k\sqrt{x_1^2 + x_2^2 + 1} - (1 + k)$ is a constrained CLF for system (7) with input constraints (9) such that $\inf \dot{V}_0(\chi) \le -W(\chi)$, where $W(\chi)$ is a continuous positive-definite function.

Define a signum-like function as $\mathrm{sgn}(a, b, c) = b$ if $a < 0$, $\mathrm{sgn}(a, b, c) = 0$ if $a = 0$, and $\mathrm{sgn}(a, b, c) = c$ if $a > 0$. By mimicking the proof in [12] that V_0 is a constrained CLF for system (7), it is straightforward to verify that V_0 is a constrained Lyapunov function for system (7) with control inputs $u_0 = \mathrm{sgn}(\bar{x}_0, \bar{\omega}, \underline{\omega})$ and $u_1 = \mathrm{sgn}(x_2, \bar{v}, \underline{v})$. Noting that $v = u_1 + v_r \cos(x_0)$ and $\omega = \omega_r - u_0$, a discontinuous controller for the kinematic model (1) is given by

$$v^c = \begin{cases} v_{min}, & x_2 > 0 \\ v_r, & x_2 = 0 \\ v_{max}, & x_2 < 0 \end{cases}, \quad \omega^c = \begin{cases} \omega_{max}, & \bar{x}_0 > 0 \\ \omega_r, & \bar{x}_0 = 0 \\ -\omega_{max}, & \bar{x}_0 < 0 \end{cases}.$$

Define the saturation function as $\mathrm{sat}(a, b, c) = b$ if $a < b$, $\mathrm{sat}(a, b, c) = a$ if $b \le a \le c$, and $\mathrm{sat}(a, b, c) = c$ if $a > c$, where it is assumed that $b < c$. Similarly, V_0 is also a constrained Lyapunov function for system (7) with control inputs $u_0 = \mathrm{sat}(-\eta_\omega \bar{x}_0, \underline{\omega}, \bar{\omega})$ and $u_1 = \mathrm{sat}(-\eta_v x_2, \underline{v}, \bar{v})$, where η_ω and η_v are required to be greater than some positive constants which are expressed precisely

in [12]. Therefore, a saturation controller for the kinematic model (1) is given by

$$v^c = \begin{cases} v_{min}, & -\eta_v x_2 < \underline{v} \\ v_r \cos(x_0) - \eta_v x_2, & \underline{v} \le -\eta_v x_2 \le \bar{v} \\ v_{max}, & -\eta_v x_2 > \bar{v} \end{cases},$$

$$\omega^c = \begin{cases} \omega_{max}, & -\eta_\omega \bar{x}_0 < \underline{\omega} \\ \omega_r + \eta_\omega \bar{x}_0, & \underline{\omega} \le -\eta_\omega \bar{x}_0 \le \bar{\omega} \\ -\omega_{max}, & -\eta_\omega \bar{x}_0 > \bar{\omega} \end{cases}.$$

Given kinematic control laws, a standard way to extend the kinematic control laws to dynamic strategies is to apply backstepping techniques. It is obvious that both v^c and ω^c are not differentiable for the discontinuous controller and the saturation controller. Note that the continuity of the saturation controller depends on the continuity of v_r and ω_r. In this paper, v_r and ω_r are only assumed to be piecewise continuous. In fact, the reference heading rate output from the real-time trajectory generator are "bang-bang" signals to achieve time-extremal properties. As a result, traditional backstepping techniques are not applicable to find dynamic control laws for the dynamic model (2). Therefore we resort to the nonsmooth backstepping approach proposed in [19] to tackle this problem.

Note that Eqs. (11), (7b), (7c), and (2) can be rewritten as

$$\dot{\chi} = f(t,\chi) + g(\chi)\xi, \quad \dot{\xi} = \nu, \tag{12}$$

where $\xi = [v, \omega]^T$, $\nu = [F/m, \tau/J]^T$,

$$f(t,\chi) = \begin{bmatrix} \lambda\omega_r + \frac{1+x_2^2}{\sqrt{x_1^2+x_2^2+1}^3} v_r \sin(x_0) + \frac{x_1 x_2}{\sqrt{x_1^2+x_2^2+1}^3} v_r \cos(x_0) \\ v_r \sin(x_0) \\ -v_r \cos(x_0) \end{bmatrix}$$

and

$$g(\chi) = \begin{bmatrix} -\frac{x_1 x_2}{\sqrt{x_1^2+x_2^2+1}^3} & -(\lambda - \frac{x_2}{\sqrt{x_1^2+x_2^2+1}}) \\ 0 & x_2 \\ 1 & -x_1 \end{bmatrix}.$$

Let $\phi(t,\chi) = [v^c, \omega^c]^T$ represent the saturation or discontinuous control law described above for the kinematic model (1). Let $\tilde{\phi}(t,\chi)$ denote the generalized time derivative of ϕ and let μ represent the minimum norm element of $\tilde{\phi}(t,\chi)$ (see [20], [19]). Define

$$\nu = [\nu_F, \nu_\tau]^T = \mu - K(\xi - \phi(t,\chi)) - \left(\frac{\partial V_0}{\partial \chi} g(\chi)\right)^T, \tag{13}$$

where K is a 2×2 symmetric positive definite matrix and

$$\left(\frac{\partial V_0}{\partial \chi} g(\chi)\right)^T = \begin{bmatrix} -\frac{\bar{x}_0}{\sqrt{\bar{x}_0^2+1}} \frac{x_1 x_2}{\sqrt{x_1^2+x_2^2+1}^3} + k\frac{x_2}{\sqrt{x_1^2+x_2^2+1}} \\ -(\lambda - \frac{x_2}{\sqrt{x_1^2+x_2^2+1}}) \frac{\bar{x}_0}{\sqrt{\bar{x}_0^2+1}} \end{bmatrix}.$$

Let the dynamic control law be given by

$$F = m\nu_F, \quad \tau = J\nu_\tau. \tag{14}$$

Theorem 1: The dynamic control law (14) guarantees that $|x - x_r| + |y - y_r| + |\theta - \theta_r| + |v - v_r| + |\omega - \omega_r| \to 0$ asymptotically as $t \to \infty$.

Proof: Consider a Lyapunov function candidate $V = V_0(\chi) + \frac{1}{2}(\xi - \phi(t,\chi))^T(\xi - \phi(t,\chi))$. Note that $\dot{V}_0 \le -W(\chi)$ if $\xi = \phi(t,\chi)$ from the argument that V_0 is a Lyapunov function for the kinematic model (1). Following Theorem 5 in [19], it can be verified that the control law (14) guarantees that $\|\chi\| + \|\xi - \phi(t,\chi)\| \to 0$ asymptotically as $t \to \infty$. Noting that $\phi(t,0) = [v_r, \omega_r]^T$, it is straightforward to see that $|\bar{x}_0| + |x_1| + |x_2| + |v - v_r| + |\omega - \omega_r| \to 0$ asymptotically as $t \to \infty$, which in turn implies that $|x - x_r| + |y - y_r| + |\theta - \theta_r| + |v - v_r| + |\omega - \omega_r| \to 0$ asymptotically as $t \to \infty$. ■

Note that unlike the case of Theorem 5 in [19], ξ does not approach zero since here we consider a tracking problem where $\phi(t,0) = [v_r, \omega_r]^T$ while Theorem 5 in [19] considers a stabilization problem where $\phi(0) = 0$.

IV. Experimental Results

In this section, we present hardware results of the tracker using both the saturation velocity controller and the discontinuous velocity controller derived in Section III. These hardware results are also compared to simulation results using the dynamic controller (14). With a nonholonomic mobile robot programmed to emulate in hardware an unmanned air vehicle flying at constant altitude, the robot is assigned to follow a desired trajectory so as to transition through several known targets in the presence of static and dynamic threats.

Table I shows the specifications of the robot and parameters used to obtain the experimental results.

TABLE I
PARAMETERS USED IN THE EXPERIMENTS.

Parameter	Value	Parameter	Value
m	10.1 (kg)	$v_{\min}$	0.075 (m/s)
J	0.13 (kg m^2)	$v_{\max}$	0.24 (m/s)
λ	1	$\omega_{\max}$	2 (rad/s)
η_v	3	v_r	$\in [0.15, 0.19]$ (m/s)
η_ω	10	ω_r	$\in [-1.25, 1.25]$ (rad/s)

Figs. 7, 8, and 9 show the hardware results of the tracker using the two velocity controllers under relatively large control authority $|\omega^c| \le 2$ (rad/s), that is, $\epsilon_{\omega 1} = \epsilon_{\omega 2} = 0.75$ (rad/s). In Fig. 7, we show the trajectories of the robot transitioning through two targets in the presence of static threats and popup threats, where stars indicate the starting points of the trajectories. Fig. 8 compares the tracking errors of the two velocity controllers. Due to vision noise, there exist a steady-state tracking error of about 0.05 meters and glitches in the heading tracking errors for both controllers. Note that the controllers are robust to glitches in the robot orientation measurement. Fig. 9 compares the reference and commanded velocities for both controllers. We can see that the velocities of the discontinuous controller

switch frequently in time. Although similar performance is achieved using both controllers, we notice that the motion of the robot using the saturation controller is smooth while the motion of the robot using the discontinuous controller has significant jerks in the velocity results.

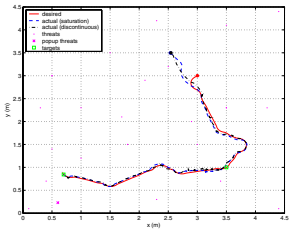

Fig. 7. Desired and actual robot trajectories using velocity controllers when there are two targets and $\epsilon_{\omega 1} = \epsilon_{\omega 2} = 0.75$ (rad/s).

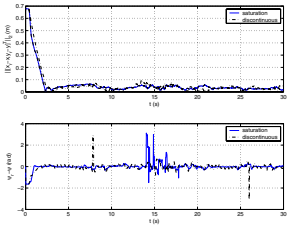

Fig. 8. Tracking errors using velocity controllers when there are two targets and $\epsilon_{\omega 1} = \epsilon_{\omega 2} = 0.75$ (rad/s).

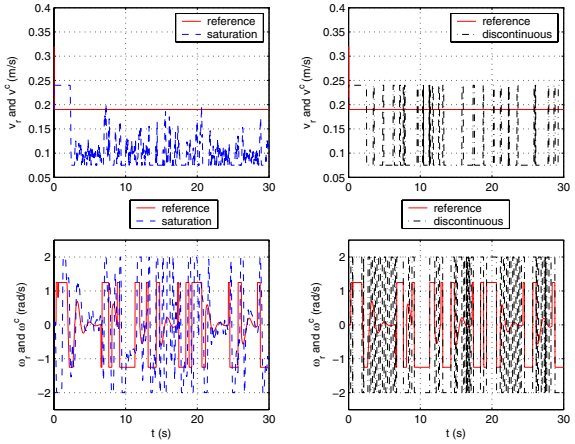

Fig. 9. Reference and commanded velocities using velocity controllers when there are two targets and $\epsilon_{\omega 1} = \epsilon_{\omega 2} = 0.75$ (rad/s).

As a comparison to the above hardware results using velocity controllers, we also show simulation results using the dynamic controller based on nonsmooth backstepping in Figs. 10, 11, 12, and 13. The robot used in our testbed has physical constraints for force and torque of $|F| \leq 30$ N and $|\tau| \leq 230$ Nm. We choose $K = \text{diag}\{2, 2\}$ in Eq. (13). Note that nonsmooth backstepping is applied to both the saturation velocity controller and the discontinuous velocity controller. The dynamic controllers based on nonsmooth backstepping for both velocity controllers have similar tracking performances as shown in Figs. 10 and 11. However, compared to the case using nonsmooth backstepping for the saturation velocity controller, switching phenomena for actual linear and angular velocities and control forces and torques are more severe in the case of using nonsmooth backstepping for the discontinuous controller as shown in Fig. 12 and 13. By comparing Figs. 8 and 11, we see that the hardware results using velocity controllers are almost comparable to the simulation results using dynamic controllers.

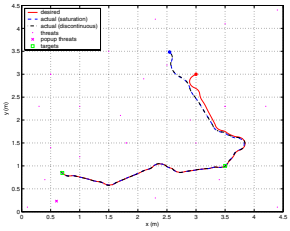

Fig. 10. Desired and actual robot trajectories using nonsmoooth backstepping when there are two targets and $\epsilon_{\omega 1} = \epsilon_{\omega 2} = 0.75$ (rad/s).

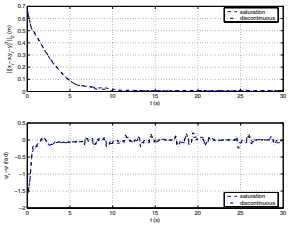

Fig. 11. Tracking errors using nonsmooth backstepping when there are two targets and $\epsilon_{\omega 1} = \epsilon_{\omega 2} = 0.75$ (rad/s).

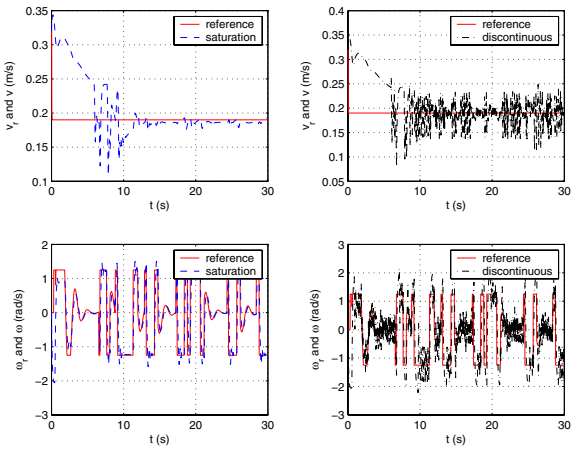

Fig. 12. Reference and actual velocities using nonsmooth backstepping when there are two targets and $\epsilon_{\omega 1} = \epsilon_{\omega 2} = 0.75$ (rad/s).

As a final test, we reduce the control authority to $|\omega^c| \leq 1.45$ (rad/s), that is, $\epsilon_{\omega 1} = \epsilon_{\omega 2} = 0.2$ (rad/s), and introduce one more target in the test field. Figs. 14, 15, and 16 show the hardware results of the tracker using the saturation velocity controller in this situation. As shown in Fig. 16, ω^c is constrained within $[-1.45, 1.45]$ (rad/s) compared to Fig. 9 where ω^c is constrained within $[-2, 2]$ (rad/s). Note that similar performances are still achieved with much smaller control authority for ω^c and the robot transitions through three target consecutively as desired.

V. Conclusion and Future Directions

This paper has presented experimental results of mobile robot tracking control using two velocity controllers accounting for UAV-like input constraints. As a step toward hardware implementation on our small fixed-wing UAVs, the controllers are computationally simple and are shown to be effective in achieving good tracking performance for mobile robots programmed to emulate UAVs flying at constant altitude. We are currently working on implementing the controllers on small fixed-wing UAVs. One implementation challenge results from the relatively low update rate of the UAV position and heading measurement. While the main control loop runs at approximately 130 HZ and the real-time trajectory generator runs at 20 HZ, the GPS can only

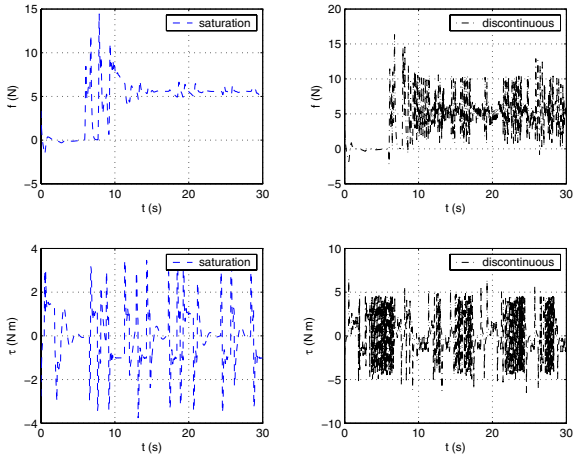

Fig. 13. Control forces and torques using nonsmooth backstepping controller when there are two targets and $\epsilon_{\omega 1} = \epsilon_{\omega 2} = 0.75$ (rad/s).

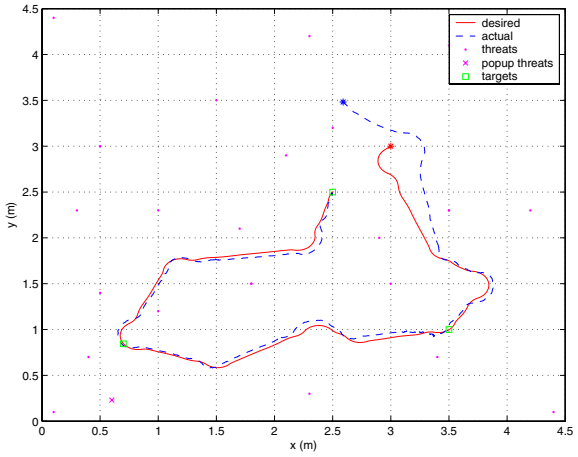

Fig. 14. Desired and actual robot trajectories using the saturation controller when there are three targets and $\epsilon_{\omega 1} = \epsilon_{\omega 2} = 0.2$ (rad/s).

provide UAV position and heading information at 1 Hz. A Kalman filter will be designed to provide missing data and mitigate the low UAV position and heading update rate. Another challenge results from the fact that the autopilot response to heading step is not truly first order as assumed. In addition, the rising time of the autopilot response to heading step is generally unknown. As a result, an adaptive control technique needs to be developed to estimate the rising time. Explicitly accounting for tracking error due to wind is also a challenge for implementing the tracking controllers on UAV hardware.

Acknowledgments

The authors gratefully acknowledge Derek Kingston, Jeff Anderson and Matt Blake for their assistance in obtaining the experimental results.

REFERENCES

[1] R. W. Brockett, "Asymptotic stability and feedback stabilization," in *Differential Geometric Control Theory* (R. S. Millman and H. J. Sussmann, eds.), pp. 181–191, Birkhaüser, 1983.

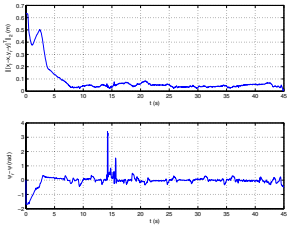

Fig. 15. Tracking errors using the saturation controller when there are three targets and $\epsilon_{\omega 1} = \epsilon_{\omega 2} = 0.2$ (rad/s).

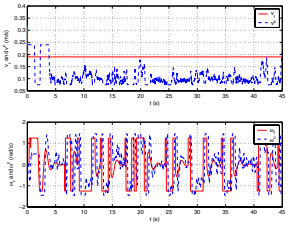

Fig. 16. Reference and commanded velocities using the saturation controller when there are three targets and $\epsilon_{\omega 1} = \epsilon_{\omega 2} = 0.2$ (rad/s).

[2] A. M. Bloch and N. H. McClamroch, "Control of mechanical systems with classical nonholonomic constraints," in *Proc. of the CDC*, (Tampa, FL), pp. 201–205, 1989.
[3] A. Astolfi, "Discontinuous control of nonholonomic systmes," *Systems and Control Letters*, vol. 24, pp. 37–45, 1996.
[4] J. P. Pomet, "Explicit design of time-varying stabilizing control laws for a class of controllable systems without drift," *Systems and Control Letters*, vol. 18, pp. 147–158, 1992.
[5] R. M. Murray and S. S. Sastry, "Nonholonomic motion planning: Steering using sinusoids," *IEEE Trans. on Automatic Control*, vol. 38, no. 5, pp. 700–716, 1993.
[6] Y.-P. Tian and S. Li, "Exponential stabilization of nonholonomic dynamic systems by smooth time-varying control," *Automatica*, vol. 38, no. 7, pp. 1139–1146, 2002.
[7] J.-M. Yang and J.-H. Kim, "Sliding mode motion control of nonholonomic mobile robots," *Control Systems Magazine*, vol. 19, no. 2, pp. 15–23, 1999.
[8] Z.-P. Jiang and H. Nijmeijer, "Tracking control of mobile robots: A case study in backstepping," *Automatica*, vol. 33, pp. 1393–1399, 1997.
[9] Z.-P. Jiang, E. Lefeber, and H. Nijmeijer, "Saturated stabilization and track control of a nonholonomic mobile robot," *Systems and Control Letters*, vol. 42, pp. 327–332, 2001.
[10] T.-C. Lee, K.-T. Song, C.-H. Lee, and C.-C. Teng, "Tracking control of unicycle-modeled mobile robots using a saturation feedback controller," *IEEE Trans. on Control Systems Technology*, vol. 9, no. 2, pp. 305–318, 2001.
[11] P. Souères, A. Balluchi, and A. Bicchi, "Optimal feedback control for line tracking with a bounded-curvature vehicle," *Int. J. of Control*, vol. 74, pp. 1009–1019, July 2001.
[12] W. Ren and R. W. Beard, "Trajectory tracking for unmanned air vehicles with velocity and heading rate constraints," *IEEE Trans. on Control Systems Technology*, vol. 12, no. 5, pp. 706–716, 2004.
[13] E. P. Anderson and R. W. Beard, "An algorithmic implementation of constrained extremal control for UAVs," in *Proc. of the AIAA GNC*, (Monterey, CA), August 2002. Paper No. AIAA-2002-4470.
[14] J. Kelsey, "MAGICC multiple robot toolbox (MMRT): A Simulink-based control and coordination toolbox for multiple robotic agents," Master's thesis, Brigham Young University, Provo, UT 84602, 2002.
[15] http://www.ee.byu.edu/magicc/.
[16] R. W. Beard, T. W. McLain, M. Goodrich, and E. P. Anderson, "Coordinated target assignment and intercept for unmanned air vehicles," *IEEE Trans. on Robotics and Automation*, vol. 18, no. 6, pp. 911–922, 2002.
[17] T. W. McLain, R. W. Beard, and J. M. Kelsey, "Experimental demonstration of multiple robot cooperative target intercept," in *AIAA GNC*, (Monterey, CA), August 2002. AIAA Paper No. AIAA-2002-4678.
[18] Y. J. Kanayama, Y. Kimura, F. Miyazaki, and T. Noguchi, "A stable tracking control method for an autonomous mobile robot," in *Proc. of the CRA*, pp. 384–389, 1990.
[19] H. G. Tanner and K. J. Kyriakopoulos, "Backstepping for nonsmooth systems," *Automatica*, vol. 39, pp. 1259–1265, 2003.
[20] D. Shevitz and B. Paden, "Lyapunov stability theory of nonsmooth systems," *IEEE Trans. on Automatic Control*, vol. 39, no. 9, pp. 1910–1914, 1994.