51st IEEE Conference on Decision and Control
December 10-13, 2012. Maui, Hawaii, USA

A Trajectory Tracking Control Scheme Design for Nonholonomic Wheeled Mobile Robots with Low-level Control Systems

Chang Boon Low
DSO National Laboratories, Singapore
Email: lchangbo@dso.org.sg

Abstract— **Motivated by formation control of multiple nonholonomic mobile robots, this paper presents a trajectory tracking control scheme design for nonholonomic mobile robots that are equipped with low-level linear and angular velocities control systems. The design includes a nonlinear kinematic trajectory tracking control law and a tracking control gains selection method that provide a means to implement the nonlinear tracking control law systematically based on the dynamic control performance of the robot's low-level control systems. In addition, the proposed scheme, by design, enables the mobile robot to execute reference trajectories that are represented by time-parameterized waypoints. This feature provides the scheme a generic interface with higher-level trajectory planners. The trajectory tracking control scheme is validated using an iRobot Packbot's parameteric model estimated from experimental data.**

I. INTRODUCTION

Formation control of multiple mobile robots has been an emerging issue where its solutions have numerous military and civilian applications. The capability of controlling multiple mobile robots cooperatively provides favourable force multiplying effects and robustness to failure. To achieve desirable cooperation among multiple mobile robots, it is critical to control and coordinate each robot in timely manner. This fundamental spatial-temporal control capability is known as trajectory tracking control.

Many trajectory tracking controllers have been proposed for nonholonomic mobile robots at both kinematic and dynamic levels. Some of the kinematic trajectory tracking controllers reported in the literature are [1], [2], [3], [4], [5], [6] to name a few. These proposed control designs assumed perfect linear and angular velocities controls. In [1], [3], locally exponentially stable kinematic tracking control laws were proposed, and in [4], [5], researchers proposed globally asymptotically stable kinematic control laws which guarantee tracking control stability for any initial tracking control error. Although the global controllers offer a large region of attraction, the initial tracking control error is desirable to be small to avoid control inputs saturations in practice. And in general, exponentially stable controllers offer better robustness performance against unmodelled uncertainties [7] compared to asymptotically stable controllers. From implementation perspective, it is undesirable to simply apply a kinematic controller on the robots without considering the dynamic effects of the robot in the design. Moreover, implementation details of these controllers are not discussed.

Trajectory tracking controllers that are designed at dynamic level can also be found in [3], [8], [9], [10], [11] where the assumed control inputs are torques that apply on the robots wheels. In [8], input-output linearization trajectory tracking controllers are proposed to execute a trajectory where the control output point is a linearly displaced point projected along the nonholonomic robot's forward body axis. And in [3], [9], [10] backstepping control design methods are applied to the control problem. In general, sufficiently accurate inertial parameters of the mobile robot's dynamic models that are required for implementing these controllers are difficult to estimate in many practical cases. In addition, many Commercially available Off-the-Shelf (COTS) mobile robots available in the market such as [12], [13], are equipped with low-level vehicle control systems such as linear velocity and angular velocity control systems where the accessible control inputs are the input commands to these control systems. In such applications, these control laws are difficult to apply on these mobile robots to achieve the desired trajectory tracking motion control.

In this paper, we propose a trajectory tracking control scheme for nonholonomic mobile robots to execute reference trajectories that are represented by time-parameterized waypoints. The mobile robot is equipped with low-level linear and angular velocities control systems where the accessible control inputs are the linear and angular velocities input commands of these control systems. A nonlinear kinematic tracking control law and a real-time trajectory reference generation algorithm are proposed to achieve stable trajectory tracking control capability along a set of discrete waypoint trajectory. Additionally, a tracking control gains selection method is proposed to provide a systematic and stable implementation for the tracking control scheme on a given mobile robot, respecting the dynamic constraints imposed by the robot's low-level control systems. These salient features not only allow the scheme to be effectively applied to COTS mobile robots that are commonly found in the industry, but also providing a generic interface between the control scheme with higher level trajectory planners.

II. PROBLEM FORMULATION

A. Problem Description

In this problem, we consider the trajectory tracking control problem of a nonholonomic mobile robot that is equipped with low-level linear and angular velocities control systems. Figure 1 depicts the mobile robot where its pose (position and heading) is defined in a global inertial (X, Y) axes, (X_b, Y_b) denotes the body axes attached on the robot's centre,

978-1-4673-2064-1/12/$31.00 ©2012 IEEE

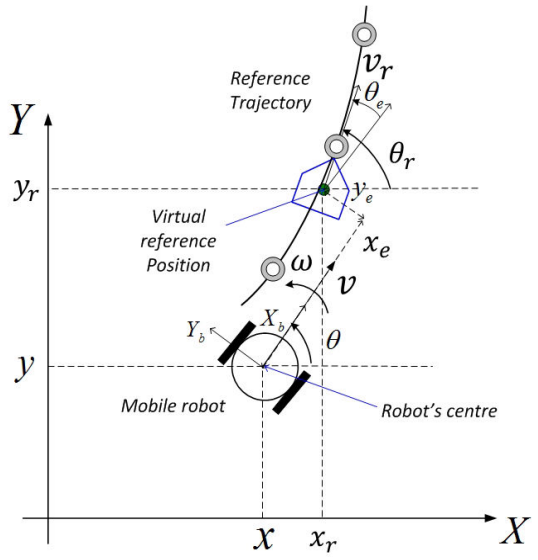

Fig. 1. Tracking control problem

$x,\ y \in \mathbb{R}$ denote the coordinates of the inertial position of the robot's centre, $\theta \in [-\pi, \pi)$ denotes the heading of the mobile robot, v denotes the linear velocity of the robot, where $\omega \triangleq \dot{\theta} \in \mathbb{R}$ denotes the angular velocity of the robot. The kinematic model of the nonholonomic robot is modelled as

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

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

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

The two low-level control systems of the mobile robot that regulate v and ω are modelled as $v(\mathbf{s}) = G_v(\mathbf{s})v_c(\mathbf{s})$ and $\omega(\mathbf{s}) = G_\omega(\mathbf{s})\omega_c(\mathbf{s})$ where $v_c,\ \omega_c \in \mathbb{R}$ denote the linear and angular velocities input commands. $G_v(\mathbf{s})$ and $G_\omega(\mathbf{s})$ denote $n \geq 1$ order stable transfer functions of the control systems. $\mathbf{s}$ denotes the Laplace variable.

The trajectory tracking control problem considered here refers to the problem of controlling the mobile robot such that its position and heading angle follow a time-varying virtual reference point or trajectory (see Figure 1). $x_r(t),\ y_r(t) \in \mathbb{R}$ and $\theta_r(t) \in [-\pi, \pi)$ denote the time-varying reference position and heading trajectories. These reference trajectories approximately satisfy the nonholonomic constraint equation, i.e., $-\dot{x}_r \sin\theta_r + \dot{y}_r \cos\theta_r \approx 0$, or equivalently

$$\dot{\mathbf{q}}_r = \begin{bmatrix} \dot{x}_r \\ \dot{y}_r \\ \dot{\theta}_r \end{bmatrix} \approx \begin{bmatrix} v_r\cos\theta_r \\ v_r\sin\theta_r \\ \omega_r \end{bmatrix}, \quad (4)$$

where $\mathbf{q}_r(t) = (x_r(t), y_r(t), \theta_r(t))$, and $v_r(t), \omega_r(t) \in \mathbb{R}$ are constructed to produce the desired reference motion. With regard to equation (4), it is assumed that $v_r(t)$ and $\omega_r(t)$ are uniformly bounded with $\inf_{\forall t \geq t_0} |v_r(t)| > 0$.

As defined in previous works [1], [3], the tracking control error variables are $x_e, y_e \in \mathbb{R}$ and $\theta_e \in [-\pi, \pi)$ where the pose tracking control error is defined as

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

x_e is the position tracking control error projected along X_b axis, y_e is the position tracking control error projected along Y_b axis, and θ_e is the heading control error. With these definitions, we state our trajectory tracking control objective as follows.

For a given small initial tracking control error $\mathbf{q}_e(t_0)$, find a trajectory tracking control law for (v_c, ω_c) such that the mobile robot stably follows the reference trajectory $\mathbf{q}_r(t)$, that is, the tracking control error $\mathbf{q}_e(t)$ is locally uniformly ultimately bounded [7] for $\forall t \geq t_0$ and converges to a small neighborhood about zero.

B. Waypoint Reference Trajectory

From implementation perspective, it is desirable to design a trajectory tracking control scheme that is able to execute trajectories that are expressed in different parameterizations with minimum or no modification to the scheme. To achieve this goal, we proposed a tracking control scheme that executes a reference trajectory that is defined by time-parameterized waypoints. In this manner, trajectories that are represented in different forms can be easily converted and expressed in the general waypoint representation.

Without loss of generality, a continuous reference trajectory $\mathbf{P}(t) = [x_{ref}(t)\ y_{ref}(t)\ \theta_{ref}(t)]$ can be represented by $\mathbf{P}(s) = [x_{ref}(s)\ y_{ref}(s)\ \theta_{ref}(s)\ v_{ref}(s)\ k_{ref}(s)]$, where $(x_{ref}(s), y_{ref}(s))$ denotes the geometric path of the trajectory, s denotes the curvilinear coordinate of the trajectory, $k_{ref}(s)$ denotes the curvature of the trajectory, and $v_{ref}(s)$ denotes the velocity profile of the trajectory. In this work, this continuous trajectory is approximated and represented by a discrete waypoint trajectory represented by a sequence of waypoints called *timing waypoints*, where each waypoint k for $k = 1, \ldots, m$ is parameterized by an array of five parameters $w_k = [x_k\ y_k\ \theta_k\ v_k\ k_k]$. A m-waypoint trajectory is represented by a sequence of arrays $\mathbf{w} = \{w_1, \ldots, w_m\}$ (see Figure 2). (x_k, y_k) denotes the position of waypoint k in (X, Y), θ_k is the angle between the line segment of waypoints $(k, k+1)$ and X axis, where v_k and k_k denote the velocity and curvature of the waypoint k. The curvature at waypoint k is defined as $k_k = \frac{\theta_{k+1} - \theta_k}{l_k}$ where $l_k = \|(x_k, y_k) - (x_{k+1}, y_{k+1})\|$. In this work, it is assumed that $\mathbf{w}$ approximates $\mathbf{P}(s)$ sufficiently, i.e., the Euclidean distance $\|\bullet\|$ between two sequential waypoints is sufficiently close at high-acceleration and nonlinear regions of $\mathbf{P}(s)$. This assumption enables the trajectory control scheme to exploit the curvature information better to achieve effective trajectory control. Here, we also assume that the velocities between waypoint k and $k+1$ are linearly interpolate with respect to time.

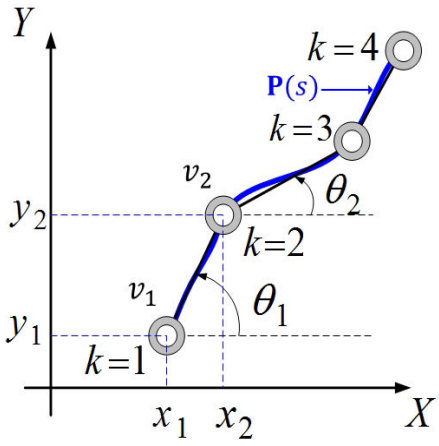

Fig. 2. Discrete Waypoint Trajectory

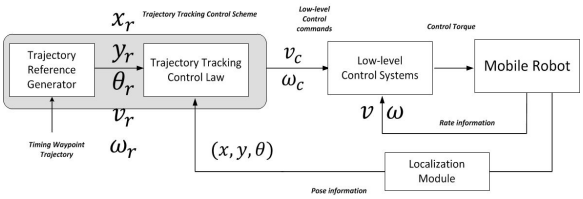

Fig. 3. Trajectory Tracking Control Architecture

III. Trajectory Tracking Control Scheme Design

A. Trajectory Tracking Control Architecture

This section briefly describes the block diagram of the trajectory tracking control scheme. The main components of the scheme are the *tracking control law* and the *trajectory reference generator* (see Figure 3). At every control sample Δt, the localization module determines the instantaneous absolute pose of the mobile robot and sends it to the tracking control law to compute the tracking control error and feedback control commands (v_c, ω_c). These computed commands are then sent to the two low-level control systems to manuever the mobile robot. Note that only robot's pose is required by the scheme and one example of such localization system that provides this information is the Global Positioning System (GPS).

B. Trajectory Reference Generator

In this work, the trajectory tracking control scheme is implemented on a digital computer, and it is desirable that the instantaneous reference generator module outputs the reference pose accurately and stably without drifting for stable tracking control.

Before we present the trajectory reference generation algorithm, we augment an additional variable t_k to array w_k for $k = 1, \ldots, m$, to denote the time that the moving reference point reaches position (x_k, y_k). Let $\tau_k = [x_k \ y_k \ \theta_k \ v_k \ k_k \ t_k]$ denotes the array of an augmented waypoint k where the augmented reference waypoint trajectory is denoted by $\tau = \{\tau_1, \ldots, \tau_m\}$. Assuming that $\mathbf{w}$ is given, t_k for $k = 1, \ldots, m$ can be computed via the following computation steps.

1: $k \leftarrow 1$
2: $t_1 \leftarrow t_0$
3: **repeat**
4: $\quad k \leftarrow k + 1$
5: $\quad t_k \leftarrow 2\|(x_k, y_k) - (x_{k-1}, y_{k-1})\|/(v_{k-1} + v_k) + t_{k-1}$
6: **until** $k = m$

Procedure WP_RGT(τ, t)

1: **while** $t \geq t_k$
2: $\quad k \leftarrow k + 1$
3: **endwhile**
4: $k \leftarrow k - 1$
5: $\Delta T \leftarrow t - t_k$
6: $\theta_r \leftarrow \theta_k + (\theta_{k+1} - \theta_k)\Delta T/(t_{k+1} - t_k)$
7: $v_r \leftarrow v_k + (v_{k+1} - v_k)\Delta T/(t_{k+1} - t_k)$
8: $\Delta d \leftarrow v_k \Delta T + \frac{1}{2}(v_r - v_k)\Delta T$
9: $x_r \leftarrow x_k + \Delta d \cos\theta_r$
10: $y_r \leftarrow x_k + \Delta d \sin\theta_r$
11: $k_{ave} \leftarrow \frac{k_k + k_{k+1}}{2}$
12: $\omega_r \leftarrow v_r k_{ave}$
13. **return** $\{x_r, y_r, \theta_r, v_r, \omega_r\}$

Note that variable k of procedure WP_RGT$(\bullet)$ is a static variable which is initialized as 1. Procedure WP_RGT$(\bullet)$ computes the reference pose and reference velocities at every control sample based on the augmented waypoint trajectory τ. This goal can be achieved by first identifying the segment that the reference point resides at time t, then followed by computing the reference position (x_r, y_r) from the pre-waypoint of the segment. The reference pose generation algorithm exploits the absolute reference position information of the waypoints to compute exact reference pose and reference velocities at every control sample without introducing undesirable errors and implementation challenges due to numerical integration of the nonlinear kinematic equations (4). The procedure WP_RGT$(\bullet)$ is implemented in MATLAB and evaluated on an Intel Core 2 Q9400 processor with a CPU speed of 2.66 GHz, and the runtime of the evaluation is shown in Table I. m denotes the number of waypoints of each trajectory. The results show that the procedure runtime

m	mean ($m$sec)	Std dev ($m$sec)
578	0.0534	0.245
662	0.056	0.260
1149	0.0539	0.250
1619	0.00547	0.249
1808	0.00579	0.240
2740	0.0588	0.249

TABLE I
RUNTIME EVALUATION

is low and independent on the number of waypoints and thus the procedure is sufficiently fast to be implemented in real-time.

C. Trajectory Tracking Control Law Design

This section presents the trajectory tracking control law design of the control scheme depicted in Figure 3. In this design, the tracking control law is specifically designed to facilitate proper control gains selection to handle the dynamic effects of the low-level control systems and at the same time without requires accurate parametric values of these control systems when implementing the scheme. One approach to achieve these goals is by using *inner-outer two loops* control design technique [14] which has been applied on fixed-wing UAVs trajectory tracking control problem [15].

In the design, we utilize the inner-outer loops control design technique to develop the trajectory tracking control law. Here, we assume that the control inputs are v and ω during the design phase, and then implement it by choosing suitable tracking control gains based on the dynamic control performance of the robot's low-level control systems so that the assumption holds well.

For longitudinal tracking control, we apply the longitudinal tracking control law proposed in [1], which is

$$v = k_x x_e + v_r \cos\theta_e, \tag{6}$$

where $k_x > 0$ is a positive constant.

For lateral tracking control, by differentiating equation (5) and by some algebraic manipulations, the dynamic of the tracking control error can be written as

$$\dot{\mathbf{q}}_e = \begin{bmatrix} \dot{x}_e \\ \dot{y}_e \\ \dot{\theta}_e \end{bmatrix} = \begin{bmatrix} \omega y_e - v + v_r \cos\theta_e \\ -\omega x_e + v_r \sin\theta_e \\ \omega_r - \omega \end{bmatrix}. \tag{7}$$

From equation (7), it can be observed that the lateral control error y_e is influenced by an intermediate state variable θ_e, and this observation suggests that we can apply *sequential-loop control design* that has been applied for aircraft control [16] to derive the tracking control law for the lateral tracking control loop. First, we design a stable closed-loop subsystem $\dot{\theta}_e = f_{\theta_e}(\theta_e, u_1)$, then followed by designing an outer control loop to control the lateral tracking control error y_e based on the auxiliary input u_1. This sequential loop design approach provides a clear physical interpretation on the control actions of the control variables in the lateral tracking control loop. One stable control law for subsystem $\dot{\theta}_e$ is $\omega = \omega_r - k_\theta(u_1 - \theta_e)$ where $k_\theta > 0$ is a positive constant. This control law leads subsystem $\dot{\theta}_e$ to the following stable linear closed-loop control subsystem, $\dot{\theta}_e = k_\theta(u_1 - \theta_e)$.

Assuming the θ_e tracks u_1 well, we propose $u_1 = \text{sat}_{\frac{\pi}{2}}(-v_r k_y y_e)$ where k_y is a positive constant that satisfies $k_y > 0$, and

$$\text{sat}_{\delta_1}(x) = \begin{cases} x & \forall |x| \le \delta_1 \\ \text{sgn}(x)\delta_1 & \forall |x| > \delta_1 \end{cases} \tag{8}$$

to stabilize the outer lateral tracking error control loop. The overall lateral feedback control law can be written as

$$\omega = \omega_r - k_\theta(\text{sat}_{\frac{\pi}{2}}(-v_r k_y y_e) - \theta_e). \tag{9}$$

Note that function $\text{sat}_{\frac{\pi}{2}}(-v_r k_y y_e)$ is chosen because it constraints the intermediate control input and output θ_e of subsystem $\dot{\theta}_e = f_{\theta_e}(\theta_e, u_1)$ to $[-\frac{\pi}{2}, \frac{\pi}{2}]$ for $\forall y_e$. This property is desirable from practical perspective because $|\theta_e(t_0)| < \frac{\pi}{2}$ allows $|\theta_e(t)| < \frac{\pi}{2}$ for $\forall t$ and hence the feedback control action would drive the control error $\mathbf{q}_e(t) \to \mathbf{0}$ instead to another isolated equilibrium point at $\mathbf{q}_e = (0, 0, \pi)$ which is undesirable in practice. We summarize the tracking control law as follows.

Proposition 1: Assuming that $v_r > 0$, then the equilibrium point $\mathbf{q}_e = \mathbf{0}$ of the closed-loop system (7) and (6), (9) is uniformly locally stable.

Proof: Consider the positive definite Lyapunov function $V_s = \frac{1}{2}x_e^2 + \frac{1}{2}y_e^2 + \frac{1}{k_\theta k_y}(1 - \cos\theta_e)$ for $|\theta_e| < \frac{\pi}{2}$. Its derivative along (7) with tracking control law (6), (9) is $\dot{V}_s = x_e(\omega y_e - k_x x_e) + y_e(-\omega x_e + v_r \sin\theta_e) + \frac{\sin\theta_e}{k_\theta k_y}(-k_\theta\theta_e - k_\theta \text{sat}_{\frac{\pi}{2}}(v_r k_y y_e))$. At a local neighbour about $\mathbf{q}_e = \mathbf{0}$ where $|v_r k_y y_e| < \frac{\pi}{2}$, $\text{sat}_{\frac{\pi}{2}}(v_r k_y y_e) = v_r k_y y_e$, $\dot{V}_s$ can be rewritten as $\dot{V}_s = -k_x x_e^2 - \frac{1}{k_y}\theta_e \sin\theta_e \le 0$. By applying Lyapunov stability theory [7], the equilibrium point $\mathbf{q}_e = \mathbf{0}$ of the closed-loop system is uniformly locally stable and this completes the proof.

Note that the linearized linear system of the closed-loop system at equilibrium point $\mathbf{q}_e = \mathbf{0}$ is $\dot{\mathbf{q}}_e = \mathbf{A}\mathbf{q}_e$, where

$$\mathbf{A} = \begin{bmatrix} -k_x & \omega_r & 0 \\ -\omega_r & 0 & v_r \\ 0 & -k_\theta v_r k_y & -k_\theta \end{bmatrix}. \tag{10}$$

The characteristic equation of matrix $\mathbf{A}$ is $s^3 + a_1 s^2 + a_2 s + a_3 = 0$, where $a_1 = k_\theta + k_x$, $a_2 = v_r^2 k_\theta k_y + k_x k_\theta + \omega_r^2$, and $a_3 = v_r^2 k_x k_\theta k_y + \omega_r^2 k_\theta$. Similar to [1], the coefficients a_i for $i = 1, 2, 3$ are positive, and $a_2 a_1 - a_3 > 0$. By Routh's stability Criterion [17], the real part of all eigenvalues of $\mathbf{A}$ are negative. Hence by applying the stability result of slow-varying systems [18], we can conclude the error convergence behavior of the closed-loop system via the following statement.

Proposition 2: Assuming that $v_r > 0$ and ω_r are continuous where $\dot{v}_r$ and $\dot{\omega}_r$ are sufficiently small, then the equilibrium point $\mathbf{q}_e = \mathbf{0}$ of the closed-loop system (7) and (6), (9) is uniformly locally exponentially stable.

Proof: The proof of this result follows the proving approach used in Proposition 2 of [1] and hence is omitted here.

In practice, it is desirable to estimate the convergence rate of the tracking control error, in particular the convergence rate of the position tracking error (x_e, y_e) and its relation with the tracking control gains. To estimate this error convergence rate, we assume that the response time of the outer lateral control loop is slower than the inner-loop subsystem $\dot{\theta}_e = k_\theta(u_1 - \theta_e)$. In such cases, we assume that $\theta_e = -v_r k_y y_e$, and hence the closed-loop equations about $(x_e, y_e) = (0, 0)$ can be approximated by

$$\dot{x}_e = \omega y_e - k_x x_e \tag{11}$$

$$\dot{y}_e = -\omega x_e - v_r^2 k_y y_e. \tag{12}$$

With this approximation, we can apply the result presented in

[15] to this estimation of convergence rate by the following result.

Proposition 3: [15] Assuming that v_r is a positive constant, and $\theta_e = -v_r k_y y_e$, then for small initial tracking error $\mathbf{q}_e(t_0)$, the position tracking control error $\chi = (x_e, y_e)$ of the closed-loop system (11)-(12) has an approximated convergence rate of $\|\chi(t)\| \leq \|\chi(t_0)\| e^{-\alpha(t-t_0)}$ for $\forall t \geq t_0$ where $\alpha = \min\{k_x, v_r^2 k_y\}$.

In this work, the control design is based on inner-outer loop control design method where the proposed kinematic tracking control law, by design, assumes perfect control inputs (v, ω). The proposed static tracking control law (6),(9) is implemented as

$$v_c = k_x x_e + v_r \cos\theta_e \quad (13)$$

$$\omega_c = \omega_r - k_\theta(\text{sat}_{\frac{\pi}{2}}(-k_y y_e) - \theta_e). \quad (14)$$

Note that one desirable feature about the proposed control law is that both tracking control laws are continuous, non-singular and well-defined at all operating states. This feature ensures that there is no abrupt change of control command signals that could impose undesirable demand on the robot's actuator slew rates. Another point to highlight here is that although Proposition 2 states that the closed-loop system (7) and (6), (9) is uniformly locally exponentially stable, during implementation, $v \neq v_c$ and $\omega \neq \omega_c$ due to imperfect control performance of the two low-level control systems. These low-level control errors lead to a bounded non-vanishing perturbation injected into the nominal tracking control closed-loop system (7) and (6), (9). And since, the nominal tracking control loop is uniformly locally exponentially stable, the perturbed system is locally uniformly ultimately bounded [7]. In practice, we would expect that the implemented tracking control system is uniformly locally ultimate bounded where the ultimate bound largely depends on the control performance of the low-level control system.

D. Tracking Control Gains Design

In Section III-C, we proposed a nonlinear kinematic tracking control law, assuming that $(v = v_c)$ and $(\omega = \omega_c)$. This assumption can be realized at implementation by selecting appropriate tracking control gains of the control law so that the control inputs $(v, \omega) \approx (v_c, \omega_c)$ which in turn leads to stable closed-loop stability via equations (13)-(14). The proposed nonlinear kinematic tracking control law, by design, possesses strong linear behaviour about the equilibrium point $\mathbf{q}_e = \mathbf{0}$. This property enables linear analysis on the behaviour of the closed-loop system about the equilibrium point, which leads to guidelines in selecting control gains for the nonlinear kinematic tracking control law to achieve $(v, \omega) \approx (v_c, \omega_c)$.

Assuming that v_r is constant, the closed-loop tracking control loop (7) and (6), (9) in the neighbourhood about $\mathbf{q}_e = \mathbf{0}$ can be approximated by

$$\dot{x}_e = -k_x x_e \quad (15)$$

$$\dot{y}_e = v_r \theta_e. \quad (16)$$

$$\dot{\theta}_e = k_\theta(u_1 - \theta_e), \quad (17)$$

where $u_1 = \text{sat}_{\frac{\pi}{2}}(-v_r k_y y_e)$. Figure 4 depicts the block diagram of equation (15). To achieve good inner-outer two loops control implementation, it is desirable that the response time of the dynamic control system $v = G_v(\mathbf{s})v_c$ is at least five times faster than the response time of its outer closed-loop (15) [14]. This dynamical constraint suggests that the tracking control gain k_x needs to satisfy $0 < B_{x_e} < \frac{B_v}{5}$, and this implies

$$0 < k_x < \frac{B_v}{5} \quad (18)$$

where $B_{x_e} = k_x$ is the bandwidth of the longitudinal tracking control loop (15) where B_v is the bandwidth of the low-level linear velocity control system $G_v(\mathbf{s})$ [15].

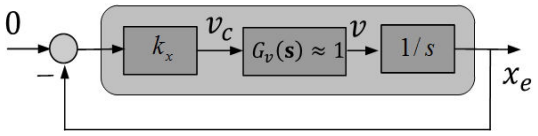

Fig. 4. Longitudinal tracking control loop

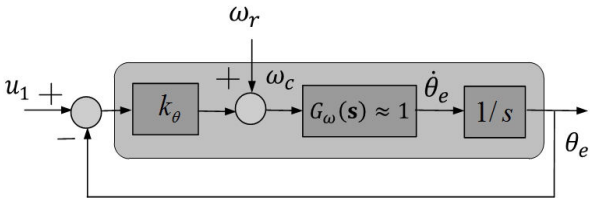

Fig. 5. Heading inner control loop $\dot{\theta}_e = k_\theta(u_1 - \theta_e)$

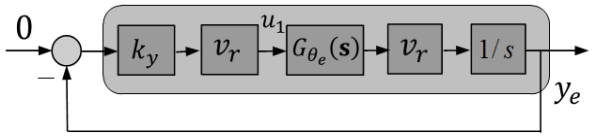

Fig. 6. Lateral tracking control loop

Similarly, Figure 6 depicts the approximated lateral tracking control loop (16)-(17) about equilibrium point $\mathbf{q}_e = \mathbf{0}$, assuming that (x_e, y_e) is small. Let $G_{\theta_e}(\mathbf{s})$ denotes the transfer function of (17) with a bandwidth of B_{θ_e}. To achieve $\omega \approx \omega_c$ in practice, we first select k_θ of the inner-loop heading control subsystem (17) such that the response time of the angular velocity control system $\omega = G_\omega(\mathbf{s})\omega_c$ is five times faster than the response time of the subsystem (17), i.e., $0 < B_{\theta_e} < \frac{B_\omega}{5}$,

$$\Rightarrow 0 < k_\theta < \frac{B_\omega}{5} \quad (19)$$

where B_ω is the bandwidth $G_\omega(\mathbf{s})$. Figure 5 depicts the signal block diagram of $G_{\theta_e}(\mathbf{s})$. Next, we define the bandwidth of the lateral outer loop (16)-(17) by selecting control gain k_y such that

$$0 < B_{y_e} < \frac{B_\omega}{5}. \quad (20)$$

This constraint is to ensure that the control signal computed by control law (9) is track-able by $\omega = G_\omega(\mathbf{s})\omega_c$. B_{y_e} denotes the bandwidth of the lateral outer loop depicted in Figure 6. This control gain k_y can be easily determined via *frequency loop shaping technique* [17] to satisfy inequality (20).

Note that the implemented control sample frequency $f_c = \frac{1}{\Delta t}$ must be sufficiently fast with respect to the control signals presence within the tracking control loops. In Section III-D, we have presented linear analysis on the tracking control loops to estimate the operating bandwidth of the control loops. With this information, we can apply similar control sample selection approach used in [15] to ensure that the chosen control sampling frequency is sufficiently fast for both longitudinal and lateral tracking control loops. The following guideline suggests that the control sampling frequency f_c to be

$$f_c \geq \frac{30}{2\pi} \max\{B_{\theta_e}, B_{x_e}, B_{y_e}\}. \quad (21)$$

Inequality (21) provides a guideline in choosing the control sample frequency for the tracking control scheme [17].

IV. Simulation

This section presents simulation validations of the proposed tracking control scheme based on an estimated parametric motion model. The model is estimated based on logged experimental data.

A. iRobot Packbot Mobile Robot's Motion Model

iRobot 510 Packbot is a ruggedized, all-terrain, all-weather tracked vehicle which has two main tracks (left and right tracks) to provide the traction force for mobility [12]. The robot's height, width, and length are 0.17m, 0.40m, and 0.8m respectively, and it has an approximate weight of 25 kg. The robot is equipped with DC motors to provide the required actuation for the longitudinal and the lateral motions. The robot has a low-level robot's on-board computer that controls the DC motors to actuate the motion of the robot. There are two low-level control modules located within the on-board computer, controlling the robot's linear and angular velocities. The computer provides (v_c, ω_c) control command interface that allows external software component to command the two low-level control modules. The robot is equipped with wheel encoders to measure the robot's linear and angular velocities for the robot's low-level control modules to regulate the velocities with respect to the command inputs. In addition, the robot is also equipped with a Fibre-Optic Gyroscope (FOG) sensor, to measure the angular velocity of the robot for logging purposes.

B. System Identification

A series of experiments were performed to collect experimental data for estimating the parametric motion model of the robot. In these runs, two sets of data were collected. One data set is used for the model estimation, and another set is used for model validation. The encoders and the gyroscope data were logged at a frequency of 20Hz.

The implemented parametric motion model is a combination of the nonholonomic kinematic equations (1)-(3) and two discrete dynamic models $v(i) = G_{v_d}(z)v_c(i)$ and $\omega(i) = G_{\omega_d}(z)\omega_c(i)$, representing the two low-level control systems of the mobile robot. i denotes the discrete time. The building blocks of the motion model is shown in Figure 7. The key objective of the models is to capture the dominant transient and steady-state behaviors of the low-level control systems. This objective can be achieved by fitting a tailor-made transfer function for $G_{v_d}(z)$ and $G_{\omega_d}(z)$ using the logged step transient responses data of the control systems. This motion model is implemented in MATLAB SIMULINK to validate the proposed trajectory tracking control scheme in the simulation.

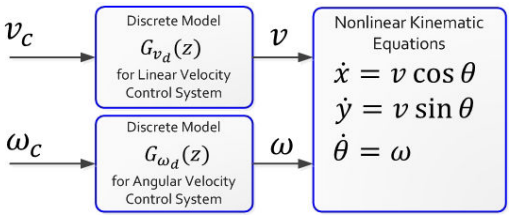

Fig. 7. Block diagram of the robot's motion model

The tailor-made parametric model chosen for $G_{v_d}(z)$ and $G_{\omega_d}(z)$ is the *Output Error (OE) model* [19] that has a form of $y(i) = G(z, \vartheta)u(i) + w(i)$ where $w(i)$ is a disturbance term, and $\vartheta = \{b_1, \ldots, b_{n_b}, f_1, \ldots, f_{n_f}\}$ are the parameters. A second-order output error model is chosen since the control variables of the low-level control systems are velocities and it is reasonable to assume that each low-level controller has an integrator component. The sample time of the transfer functions is chosen as $T_s = 0.05$ sec.

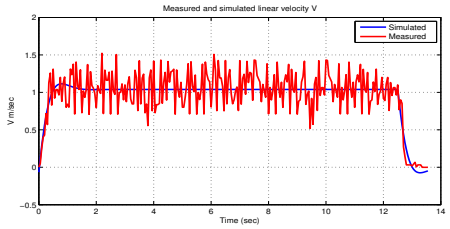

Fig. 8. Experimental and simulated step responses of linear velocity control system

Figure 8 depicts the simulated and measured step linear velocity responses based on the estimated motion model using the second set of experimental data. In the experimental runs, linear velocity responses data of the step command inputs $(v_c, \omega_c) = (1\text{m/sec}, 0\text{rad/sec})$ were collected for the parameters estimation and validation. Similarly, two sets of angular velocity responses data of the step command inputs $(v_c, \omega_c) = (0\text{m/sec}, 0.7\text{rad/sec})$ were collected for the estimation and validation. The parameters of the tailor-made model are computed using MATLAB "*ident*" Graphical User Interface (GUI) of *MATLAB system identification toolbox* [20]. The estimated discrete model of the linear velocity

control system $G_v(\mathbf{s})$ is

$$G_{v_d}(z) = \frac{0.1714z^{-1} - 0.13144z^{-2}}{1 - 1.709z^{-1} + 0.7449z^{-2}}. \quad (22)$$

The transfer function has a rise time $t_r = 0.38$ sec, bandwidth of $B_{v_d} = 5.3$ radsec^{-1}, and a static gain $G_{v_d}(1) = 1.113$.

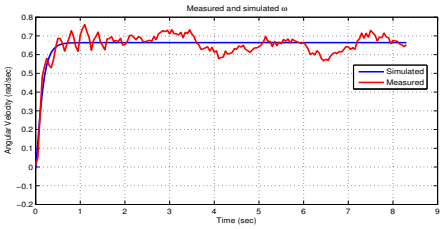

Fig. 9. Experimental and simulated step responses of angular velocity control system

Simiarly, Figure 9 depicts the simulated and measured step angular velocity responses based on the second validating data set. The estimated discrete model of the angular velocity control system is

$$G_{\omega_d}(z) = \frac{0.1101z^{-1} + 0.1101z^{-2}}{1 - 0.9719z^{-1} + 0.204z^{-2}}. \quad (23)$$

The rise time, bandwidth, and static gain of (23) are $t_r = 0.3$ sec, $B_{\omega_d} = 7.2$ rad/sec, and $G_{\omega_d}(1) = 0.948$ respectively.

C. Simulation Setup

The tracking control scheme is implemented with the robot's nonlinear motion model (1)-(3) and (22)-(23) to evaluate the control performance. The linear and angular velocities of the reference trajectory are $v_r(t) = 1$ m/sec and $\omega_r(t) = 0.2$ rad/sec for $\forall t$. The reference trajectory is discretized and represented in the timing waypoints as discussed in Section II-B. In the simulation runs, the tracking control error at initial time $t_0 = 0$ is $\mathbf{q}_e(0) = (3\text{m}, 3\text{m}, 0.1\text{rad})$, and the initial low-level control systems' outputs are $(v(t_0), \omega(t_0)) = \mathbf{0}$.

The estimated bandwidth of the linear velocity control system B_v is 5.5 rad/sec. One longitudal tracking control gain that satisfies inequality (18) is $k_x = 0.5$. For lateral tracking control loop, since the estimated bandwidth of the robot's angular velocity control system B_ω is 7.2 rad/sec, we first select $k_\theta = 1$ to meet inequality (19). As for k_y, we select a positive value such that inequality (20) is satisfied. One such positive constant is $k_y = 0.50$. The bandwidth of the longitudinal tracking loop is $B_{x_e} = 0.5$ rad/sec, where the bandwidth of the lateral tracking loop is $B_{y_e} = 0.7062$ rad/sec. One control sample frequency that satisfies (21) is $\Delta t = 0.1$ sec (10Hz).

D. Simulation Results

Figures 10 and 11 depict the tracking control performance of the tracking control law (6), (9) and the control inputs of the robot. Figure 10 shows that the implemented tracking

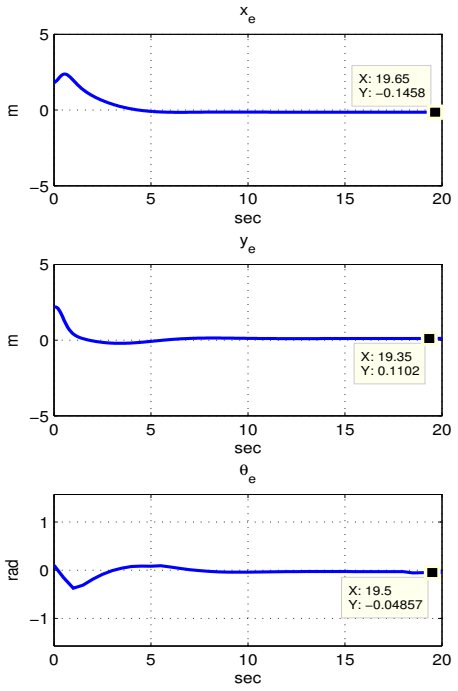

Fig. 10. Tracking control error $x_e(t), y_e(t), \theta_e(t)$

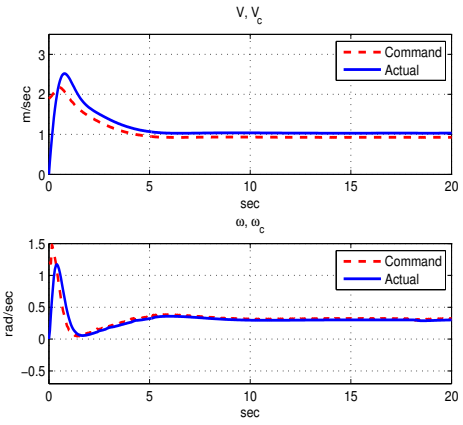

Fig. 11. Command and actual control inputs

control law converged the initial tracking control error $\mathbf{q}_e(0)$ to a small neighbourhood about zero in an exponentially fashion. The steady-state error of this simulation run is $\mathbf{q}_e(\infty) = (-0.14\text{m}, 0.11\text{m}, 0.04\text{rad})$. This non-zero steady-state error is due to the non-unity static gains of the low-level control systems $\{G_{v_d}(1), G_{\omega_d}(1)\}$ implemented in the simulation (see Figure 11). The result suggests that the tracking control law is robust towards imperfect control performance of the low-level control systems. The smooth non-oscillating responses validate the effectiveness of the proposed tracking control scheme.

V. CONCLUSIONS

This paper presents a trajectory tracking control scheme that is designed for mobile robots that are equipped with low-level linear and angular velocities control systems where the accessible control inputs are the command inputs to these control systems. In this work, the proposed trajectory tracking control law provides a locally uniformly ultimately bounded performance. The tracking control law is specifically designed to enable systematic and proper selection of tracking control gains according to the dynamic constraints of the robot's low-level control systems. A real-time trajectory reference generation algorithm is developed to provide accurate and stable reference pose (position and heading) generation at every control sample for the tracking control law to execute. The reference generation avoids errors and implementation uncertainties due to numerical integration of the nonlinear kinematic equations. This desirable feature enables the tracking control scheme to be applied reliably over long duration applications. Another positive feature about the scheme is that the reference trajectory is represented by a set of discrete waypoints. This feature provides a general and open interface between the tracking control scheme and higher-level trajectory planners. Experimental data of an iRobot Packbot 510 mobile robot is used to estimate the robot's parameteric motion models. These estimated motion model is implemented in MATLAB SIMULINK for validating the proposed tracking control scheme in simulations. The numerical data suggests that the tracking control scheme provides a locally uniformly ultimately bounded control performance. Future work will look into experimental implementation of the proposed tracking control scheme on a real physical nonholonomic mobile platform and application of the tracking control scheme to formation control problems.

ACKNOWLEDGMENT

The author would like to thank Mr New Ai Peng, Mr Hikaru Fujishima, Mr Peh Tzer Shiuann from DSO for supporting this research work.

REFERENCES

[1] Y. Kanayama, Y. Kimura, and F. Miyazaki, "A stable tracking control method for an autonomous mobile robot," *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 1, pp. 384-389, May, 1990.

[2] Y. F. Zheng, "Recent trends in mobile robots," *World Scientific*, 1993.

[3] Z. P. Jiang, and H. Nijmeijer, "Tracking control of mobile robots: a case study in backstepping," *Automatica*, vol. 33, no. 7, Feb, 1997.

[4] Z. P. Jiang, E. Lefeber, and H. Nijmeijer, "Saturated Stabilization and tracking of a nonholonomic mobile robot," *Systems and Control Letters*, vol. 42, pp. 327-332, 2001.

[5] W. Ren, J.-S. Sun, R. Beard, and T. McLain, "Experimental Validation of an autonomous control system on a mobile robot platform," *The Institution of Engineering and Technology*, vol. 1, no. 6, pp. 1621-1629, Nov, 2007.

[6] J. S. Mejía, K. Srivastava, and D. M. Stipanović, "Collision Avoidance and Trajectory Tracking Control based on Approximations of the Maximum Function," *Proceedings of IEEE American Control Conference*, pp. 3051-3056, June, 2010.

[7] H. K. Khalil, "Nonlinear Systems," Prentice Hall, 3rd Ed, 2002.

[8] B. d'Andréa-Novel, G. Campion, and G. Bastin, "Control of nonholonomic wheeled mobile robots by state feedback linearization," *International Journal of Robotics Research*, vol. 14, no. 6, pp. 543-559, 1995.

[9] R. Fierro, and F. L. Lewis, "Control of a nonholonomic mobile robot: Backstepping Kinematics into Dynamics," *Journal of Robotic Systems*, vol. 14, no. 3, pp. 149-163, 1997.

[10] Z. P. Jiang, and H. Nijmeijer, "A Recursive Technique for Tracking Control of Nonholonomic Systems in Chained Form," *IEEE Transactions on Automatic Control*, vol. 44, no. 2, pp. 265-279, Feb, 1999.

[11] W. E. Dixon, D. M. Dawson, E. Zergeroglu, and F. Zhang, "Robust tracking and regulation control for mobile robots," *International Journal of Robust and Nonlinear Control*, vol. 10, no. 1, pp. 199-216, 2000.

[12] IRobot home page, "http://www.irobot.com/gi/ground/510_PackBot", Dec, 2010.

[13] http://imara.inria.fr/imara/platforms/hardware/vehicles/cycab

[14] S. Skogestand, and I. Postlethwaite, "Multivariable Feedback control Analysis and Design," John Wiley and Sons, 2nd Ed, 2005.

[15] C. B. Low, "A Trajectory Tracking Control Design for Fixed-wing Unmanned Aerial Vehicles," *Proceedings of IEEE Conference on Control Applications*, pp. 2118-2123, Sep, 2010.

[16] B. L. Stevens, and F. L. Lewis, "Aircraft control and simulation," 2nd Ed, 2003.

[17] G. F. Franklin, J. D. Powell, A. E. Naeini, "Feedback Control of Dynamic Systems," Prentice Hall, 4th Ed, 2001.

[18] W. J. Rugh, "Linear System Theory," Prentice Hall, pp. 117-119, 1993.

[19] L. Ljung, "System Identification: Theory for the user," Prentice Hall, 2nd Ed, 1999.

[20] MATLAB System Identification Toolbox home page, "http://www.mathworks.com/products/sysid/", Dec, 2010.