Available online at www.sciencedirect.com
ScienceDirect
Robotics and Autonomous Systems 56 (2008) 461–479
Robotics and Autonomous Systems
www.elsevier.com/locate/robot

Omni-directional mobile robot controller based on trajectory linearization

Yong Liu[a], J. Jim Zhu[a,*], Robert L. Williams II[b], Jianhua Wu[c]

[a] School of Electrical Engineering and Computer Science, Ohio University, Athens, OH 45701, United States
[b] Department of Mechanical Engineering, Ohio University, Athens, OH 45701, United States
[c] Ohio Design Center, SEWS Inc. Marysville, OH 43040, United States

Received 20 June 2006; received in revised form 27 August 2007; accepted 28 August 2007
Available online 24 October 2007

Abstract

In this paper, a nonlinear controller design for an omni-directional mobile robot is presented. The robot controller consists of an outer-loop (kinematics) controller and an inner-loop (dynamics) controller, which are both designed using the Trajectory Linearization Control (TLC) method based on a nonlinear robot dynamic model. The TLC controller design combines a nonlinear dynamic inversion and a linear time-varying regulator in a novel way, thereby achieving robust stability and performance along the trajectory without interpolating controller gains. A sensor fusion method, which combines the onboard sensor and the vision system data, is employed to provide accurate and reliable robot position and orientation measurements, thereby reducing the wheel slippage induced tracking error. A time-varying command filter is employed to reshape an abrupt command trajectory for control saturation avoidance. The real-time hardware-in-the-loop (HIL) test results show that with a set of fixed controller design parameters, the TLC robot controller is able to follow a large class of 3-degrees-of-freedom (3DOF) trajectory commands accurately.

Keywords: Mobile robot; Nonlinear control; Trajectory linearization; Omni-directional; Sensor fusion; Robocup

1. Introduction

An omni-directional mobile robot is a type of holonomic robots. It has the ability to move simultaneously and independently in translation and rotation. The inherent agility of the omni-directional mobile robot makes it widely studied for dynamic environmental applications [1,2,15]. The annual international Robocup competition in which teams of autonomous robots compete in soccer-like games, is an example where the omni-directional mobile robot is used.

The Ohio University (OU) Robocup Team's entry Robocat is for the Robocup small-size league competition. The current OU Robocup robots are Phase V omni-directional mobile robots, as shown in Fig. 1. The Phase V Robocat has three omni-directional wheels, arranged 120 deg apart. Each wheel is driven by a DC motor installed with an optical shaft encoder. An overhead camera above the field of play senses the position and the orientation of the robots using a real-time image processing algorithm and the data are transmitted to the robot through an unreliable wireless communication channel.

A precise trajectory tracking control is a key component for applications of omni-directional robots. The trajectory tracking control of an omni-directional mobile robot can be divided into two tasks, path planning and trajectory following [3,5]. Path planning calls for computing a feasible and optimal geometric path. Optimal trajectory path planning algorithms for the omni-directional mobile robots are discussed in [3,4,14,16–18]. In [14], the dynamic path planning for omni-directional robot is studied considering the robot dynamic constraints. In this paper, the main focus is on accurate trajectory following control, given a feasible trajectory command within the robot physical limitations. While optimal dynamic path planning is not within the scope of the paper, an ad hoc yet effective time-varying bandwidth command shaping filter is employed for actuator saturation and integrator windup avoidance, in the presence of an abrupt command trajectory violating the robot's dynamic constraints.

* Corresponding address: School of Electrical Engineering and Computer Science, 353 Stocker Center, The Ohio University, Athens, OH 45701, United States. Tel.: +1 740 597 1506; fax: +1 740 593 0007.
E-mail addresses: Yong.Liu.1@ohio.edu, yl542401@ohio.edu (Y. Liu), zhuj@ohio.edu (J.J. Zhu), williar4@ohio.edu (R.L. Williams II), jw322400@yahoo.com (J. Wu).

0921-8890/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.robot.2007.08.007

Nomenclature

Body frame

r	Angular rate of body rotation (rad/s)
u, v	Velocity component in the body frame (m/s)
f_1, f_2, f_3	Traction force of each wheel (N)
E_1, E_2, E_3	Applied voltage on each motor (V)
$\omega_{m1}, \omega_{m2}, \omega_{m3}$	Motor shaft speed (rad/s)

World frame

(x, y)	Robot location (m)
Ψ	Robot orientation angle (rad)

Mechanical constants

m	Robot mass (kg)
I_z	Robot moment inertia (kg m^2)
R	Wheel radius (m)
L	Radius of robot body (m)
n	Gear ratio
δ	Wheel orientation angle (30°) (deg)

The omni-directional robot controller design has been studied based on different robot dynamics models. In an early version of Robocat controller [6,14], which is similar to [16], only kinematics are considered in the controller design. Each motor is controlled by an individual PID controller to follow the speed command from inverse kinematics. Without considering the coupled nonlinear dynamics explicitly in the controller design, the trial-and-error process of tuning the PID controller gains is tedious [14]. In [3,17,19], kinematics and dynamics models of omni-directional mobile robots have been developed, which include the motor dynamics but ignored the nonlinear coupling between the rotational and translational velocities. Thus the robot dynamics model is simplified as a linear system. In [3,4,17], optimal path planning and control strategies have been developed for position control without considering orientation control, and the designed controller was tested in simulations and experiment. In [19], two independent PID controllers are designed for controlling position and orientation separately based on the simplified linear model. In [6–8,14], a nonlinear dynamic model including the nonlinear coupling terms has been developed. In [7], a resolved-acceleration control with PI and PD feedback has been developed to control the robot speed and orientation angle. It is essentially a feedback linearization control. That controller design is tested on the robot hardware. In [8], based on the same model in [7], PID, self tuning PID, and fuzzy control of omni-directional mobile robots have been studied. In [12], a variable-structure-like nonlinear controller has been developed for general wheel robot with kinematics disturbance, in which a globally uniformly ultimately bounded stability (GUUB) is achieved. In [27], feedback linearization control for wheeled mobile robot kinematics has been developed and tested on a two-wheel nonholonomic robot. In [28], a fuzzy tracking controller has been developed for a four-wheel differentially steered robot without explicit model of the robot dynamics.

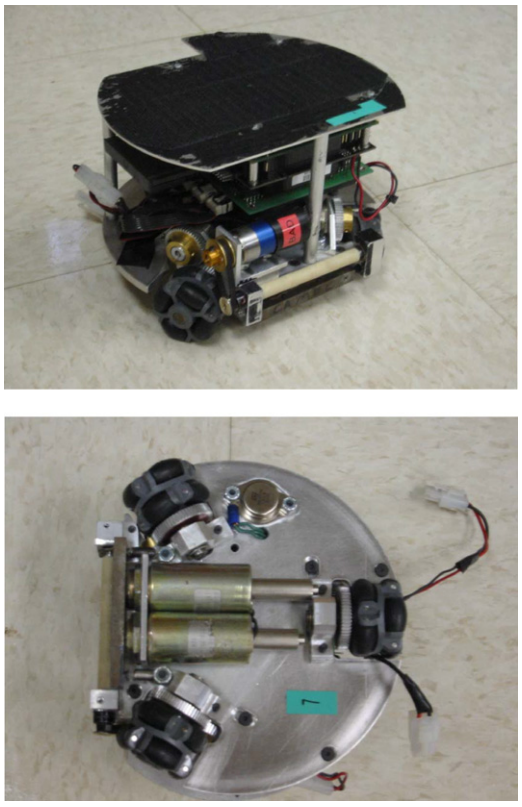


Fig. 1. Phase V Robocat robot.

In this paper first, a detailed nonlinear dynamics model of the omni-directional robot is presented, in which both the motor dynamics and robot nonlinear motion dynamics are considered. Instead of combining the robot kinematics and dynamics together as in [6–8,14], the robot model is represented by the separated kinematics equation and the body frame dynamics equation. Such representation facilitates the robot motion analysis and controller design.

Second, based on the nonlinear robot model, a 3-degrees-of-freedom (3DOF) robot tracking controller design, using a nonlinear control method, is presented. To simplify the design, a two-loop controller architecture is employed based on the time-scale separation principle and singular perturbation theory. The outer-loop controller is a kinematics controller. It adjusts the robot position and orientation to follow the commanded trajectory. The inner-loop is a dynamics controller which follows the body rate command given by the outer-loop controller. Both outer-loop and inner-loop controllers employ a nonlinear control method based on linearization along a nominal trajectory. It is known as trajectory linearization control (TLC) [10]. Preliminary results of the proposed robot TLC controller have been summarized in [9]. It is worth noting that the presented controller can be used as a velocity and orientation controller, which is similar to the controller structure in [7]. TLC combines the nonlinear dynamic inversion and linear time-varying eigenstructure assignment in a novel way, and has been successfully applied to missile and reusable

launch vehicle flight control systems [10,11]. The nonlinear tracking and decoupling control by trajectory linearization can be viewed as an ideal gain-scheduling controller designed at every point on the trajectory. TLC can achieve exponential stability along the nominal trajectory, therefore it provides robust stability and performance along the trajectory without interpolation of controller gains. The developed robot TLC controller serves as both position controller and trajectory following control with the same set controller parameters. Compared with the nonlinear controllers in [12,27], the proposed TLC mobile robot controller deals with both kinematic disturbances (outer-loop controller) and dynamic disturbance (inner-loop controller). Compared with [12], the TLC controller can achieve robust performance under less strict assumptions, while eliminating the chattering control signals in [12]. It should be noted that the structure of TLC is different from another nonlinear control method—feedback linearization control (FLC) [30–32]. In an FLC design, a nonlinear dynamic system is transformed to a linear system via a nonlinear coordinate transformation and a nonlinear state feedback that cancels the nonlinearity in the transformed coordinates. Then a linear time-invariant (LTI) controller is designed for the transformed linear system to satisfy the disturbance and robustness requirements for the overall system. The FLC relies on the nonlinearity cancellation in the transformed coordinate via nonlinear state feedback. In an actual control system, the cancellation of nonlinear terms will not be exact due to modeling errors, uncertainties, measurement noise and lag, and the existence of parasitic dynamics. The LTI feedback controller designed under the nominal conditions may not effectively handle the nonlinear time-varying residual dynamics.

Third, a sensor fusion scheme for robot position and orientation measurements is presented. From real-time experiments, it is observed that the accurate position and orientation measurements are essential for the controller performance. On-board sensors, such as motor shaft encoders, can be used to estimate robot location and orientation by integrating the measured robot velocity. Such estimation is fast, but also has inevitable cumulative errors introduced by the wheel slippage and the sensor noise. Calibration methods for mobile robot odometry have been developed to reduce the position estimation error [25,26]. While these methods have enhanced the accuracy of odometry position estimation, the estimation drift cannot be eliminated without an external reference. On the other hand, global position reference sensors, such as a vision system using a roof camera senses the robot location and orientation directly without drifting. However, it is relatively slow and sometimes unreliable due to the image processing and communication errors. Thus, a sensor fusion technique is presented, which combines both the global vision system and on-board sensor estimation to provide an accurate and reliable location measurement. It is based on a nonlinear Kalman filter using trajectory linearization.

In Section 2, the omni-directional mobile robot dynamics model is presented. Based on this model, in Section 3, a dual-loop robot TLC controller is developed. In Section 4, the sensor fusion method is described. In Section 5, controller parameter tuning and the time-varying bandwidth command shaping filter are discussed. In Section 6, real-time hardware-in-the-loop (HIL) test results are presented.

2. The omni-directional mobile robot model

In this section, the robot equations of motion are derived based on some typical simplifying assumptions. It is assumed that the wheels have no slippage in the direction of traction force. Only viscous friction forces on the motor shaft and gear are considered. The wheel contact friction forces that are not in the direction of traction force are neglected. The motor electrical time constant is also neglected. The developed dynamics model is similar to those in [6–8,14]. The unmodeled slippage has been studied in [13]. In the controller design, the slippage, as well as other ignored dynamics, are considered as perturbations to the simplified dynamics model. The close-loop controller based on the simplified model is capable of compensating for disturbances and perturbations. Moreover, the slippage of the wheels also introduces measurement when the motor shaft encoder readings are used to estimate the robot position and orientation. In this paper, such measurement errors are compensated by the sensor fusion scheme described in Section 4.

There are two coordinate frames used in the modeling: the body frame {B} and the world frame {W}. The body frame is fixed on the moving robot with the origin in the robot geometric center, which is assumed to be the center of gravity, as shown in Fig. 2(a). The world frame is fixed on the field of play, as shown in Fig. 2(b). Symbols used in the robot dynamic model are listed in the nomenclature.

From the force analysis shown in Fig. 2(c), in the body frame we have

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} rv \\ -ru \\ 0 \end{bmatrix} + H \cdot B \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix}, \tag{1}$$

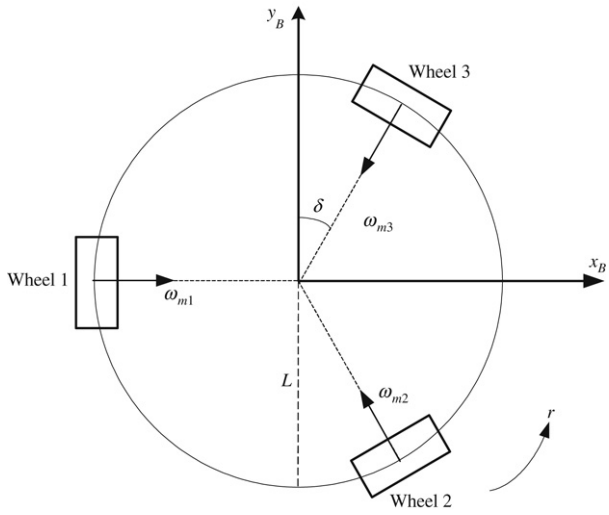
where

$$H = \begin{bmatrix} \frac{1}{m} & 0 & 0 \\ 0 & \frac{1}{m} & 0 \\ 0 & 0 & \frac{1}{I_Z} \end{bmatrix},$$

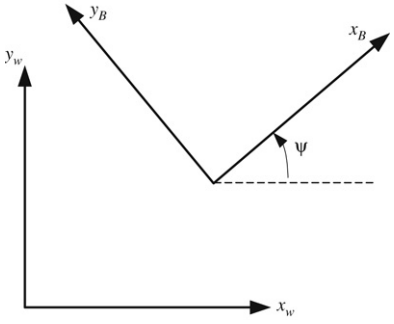
$$B = \begin{bmatrix} 0 & \cos\left(\frac{\pi}{6}\right) & -\cos\left(\frac{\pi}{6}\right) \\ -1 & \sin\left(\frac{\pi}{6}\right) & \sin\left(\frac{\pi}{6}\right) \\ L & L & L \end{bmatrix}.$$

From the mobile robot geometry, we have

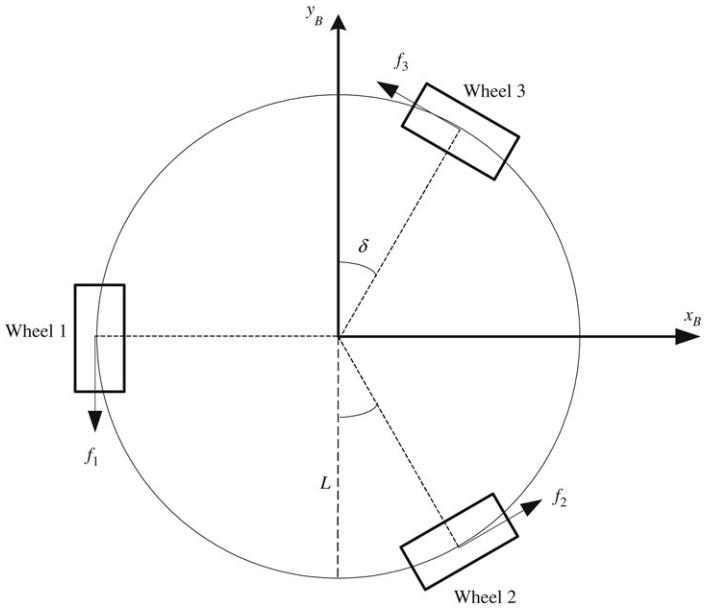
$$\begin{bmatrix} u \\ v \\ r \end{bmatrix} = (B^{\mathrm{T}})^{-1} \frac{R}{n} \begin{bmatrix} \omega_{m1} \\ \omega_{m2} \\ \omega_{m3} \end{bmatrix}. \tag{2}$$



(a) Body frame {B}.



(b) World frame {W}.



(c) Force analysis.

Fig. 2. Coordinate frames and force analysis.

The dynamics of each DC motor are described using the following equations

$$L_a \frac{\mathrm{d}i_a}{\mathrm{d}t} + R_a i_a + k_3 \omega_m = E \quad \text{and}$$

$$J_0 \dot{\omega}_m + b_0 \omega_m + \frac{Rf}{n} = k_2 i_a,$$

where E is the applied armature voltage, i_a is the armature current, ω_m is the motor shaft speed, L_a is the armature inductance, R_a is the armature resistance, k_3 is the back emf constant, k_2 is the motor torque constant, J_0 is the combined inertia of the motor, gear train and wheel referred to the motor shaft, b_0 is the viscous-friction coefficient of the motor, gear and wheel combination, R is the wheel radius, f is the wheel traction force, and n is the motor to wheel gear ratio. Since the electrical time constant of the motor is very small compared to the mechanical time constant, we can neglect the motor electric circuit dynamics, which leads to $L_a \frac{\mathrm{d}i_a}{\mathrm{d}t} = 0$ and $i_a = \frac{1}{R_a}(E - k_3\omega_m)$. With this assumption, and using vector notation, the dynamics of the three identical motors can be written as

$$J_0 \begin{bmatrix} \dot{\omega}_{m1} \\ \dot{\omega}_{m2} \\ \dot{\omega}_{m2} \end{bmatrix} + b_0 \begin{bmatrix} \omega_{m1} \\ \omega_{m2} \\ \omega_{m3} \end{bmatrix} + \frac{R}{n} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} = \frac{k_2}{R_a} \begin{bmatrix} E_1 \\ E_2 \\ E_3 \end{bmatrix} - \frac{k_2 k_3}{R_a} \begin{bmatrix} \omega_{m1} \\ \omega_{m2} \\ \omega_{m3} \end{bmatrix}. \tag{3}$$

By combining (1)–(3), we get the dynamics model of the mobile robot in the body frame {B} with the applied motor

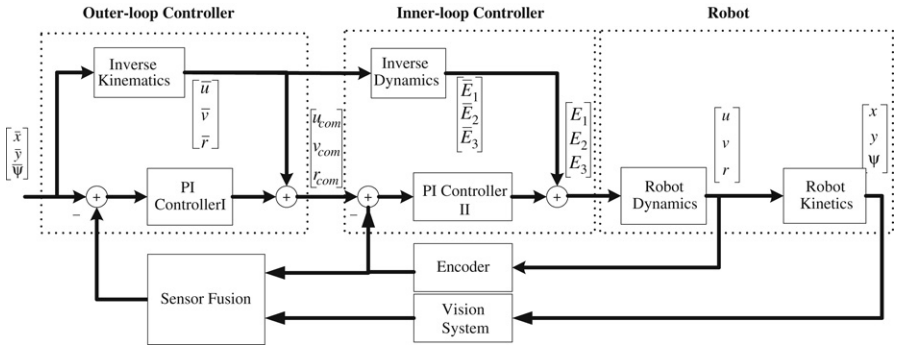


Fig. 3. Robot TLC controller structure.

voltages E_1, E_2, E_3 as the control input

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{r} \end{bmatrix} = G^{-1} \begin{bmatrix} rv \\ -ru \\ 0 \end{bmatrix} - G^{-1} HBB^{\mathrm{T}} \left(\frac{k_2 \cdot k_3}{R_a} + b_0 \right) \times \frac{n^2}{R^2} \begin{bmatrix} u \\ v \\ r \end{bmatrix} + G^{-1} HB \frac{k_2 n}{R \cdot R_a} \begin{bmatrix} E_1 \\ E_2 \\ E_3 \end{bmatrix}, \tag{4}$$

where $G = (I + HBB^{\mathrm{T}} \frac{n^2 J_0}{R^2})$.

The kinematics of the robot is given by a coordinate transformation from the body frame to the world frame

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\Psi} \end{bmatrix} = \begin{bmatrix} \cos \Psi(t) & -\sin \Psi(t) & 0 \\ \sin \Psi(t) & \cos \Psi(t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ r \end{bmatrix}. \tag{5}$$

The equations of motion (4) and (5) describe the simplified robot behavior. In this model the friction constant b_0 can be determined experimentally. Eqs. (4) and (5) show that omni-directional mobile robot has coupled MIMO nonlinear dynamics. The nonlinear term $G^{-1}\begin{bmatrix} rv & -ru & 0 \end{bmatrix}^{\mathrm{T}}$ in (4) is introduced by robot rotation. The robot nonlinear dynamics can be reduced to a linear system if either the robot does not rotate while in translation, or the robot rotates at a fixed position without translation. In both the cases, linear controllers can be applied to the dynamics equation (4), as in [3,17,19]. While the kinematics equation (5) can be easily inverted to yield an open-loop kinematics controller, the position tracking error dynamics are nonetheless coupled, nonlinear and time-varying (when $r \neq 0$), rendering the feedback position tracking error stabilization control a challenge for LTI controller design techniques.

When linear time-invariant controllers are used for 3DOF command trajectories, especially for commands with high translational and rotational velocities, the nonlinear robot kinematics and dynamics can no longer be ignored. For example, the maximum translational and rotational velocities of the Phase V Robocat robot are estimated at 1.18 (m/s) and 16.86 (rad/s), respectively [14]. This yields a perturbation in acceleration to the linearized dynamics model [14] as high as $\sup_t \|G^{-1}[\, r(t)v(t) \quad -r(t)u(t) \quad 0\,]^{\mathrm{T}}\|_\infty = 2.5$ (m/s^2), which is not accounted for by the linear controller design. This perturbation will significantly reduce the domain of stability, or even destabilize the system due to other modeling errors and disturbances. In order to maintain stability of the robot controller, different feedback gains have to be scheduled for different trajectories, or the command trajectories have to be restricted to separate translation trajectories and rotation trajectories. The early version of Robocat robot employed a linear controller similar to those in [3,17,19]. It was observed that the robot lost stability when it was given a command with both translational and rotational motions. To achieve better performance, a nonlinear controller design, which can directly deal with the intrinsic nonlinearity and the coupling of robot motions while tracking any given feasible trajectories, is desired.

3. Trajectory linearization controller design

In the previous Robocat controller design, three independent motor speed controllers are employed. As for most omni-directional robots, the open-loop command of each motor controller is computed by dynamic inversion of the robot kinematics. However, due to the inevitable errors in the open-loop controller, in most experiments the robot cannot follow the desired trajectories with satisfactory performance.

In this section, a controller design based on Trajectory Linearization Control (TLC) is presented. A two-loop controller architecture is employed, as shown in Fig. 3. The outer-loop controller adjusts the robot position to follow the commanded trajectory. The inner-loop is a body rate controller which follows the body rate command from the outer-loop controller. In both outer-loop and inner-loop controllers, TLC is employed. A TLC controller consists of two parts. The first part is an open-loop controller which computes the nominal control and nominal trajectory using a pseudo-dynamic inverse of the plant model. The second part is a feedback controller which stabilizes the system tracking error dynamics along the nominal trajectory. The dual-loop design is based on the singular perturbation theory, commonly known as the time-scale separation principle, which assumes that the inner-loop is exponentially stable and the inner-loop's bandwidth is much higher than the outer-loop dynamics, so that the outer-loop controller can be designed by ignoring the inner-loop dynamics. This assumption is satisfied by assigning appropriate closed-loop PD-eigenvalues [34] to both the control loops. Controller parameter tuning is discussed in Section 5.1.

3.1. Outer-loop controller

First, from (5), the nominal body rate for a desired trajectory $\begin{bmatrix}\overline{x}(t) & \overline{y}(t) & \overline{\Psi}(t)\end{bmatrix}^{\mathrm{T}}$ is

$$\begin{bmatrix}\overline{u}\\ \overline{v}\\ \overline{r}\end{bmatrix} = \begin{bmatrix}\cos\overline{\Psi}(t) & \sin\overline{\Psi}(t) & 0\\ -\sin\overline{\Psi}(t) & \cos\overline{\Psi}(t) & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}\dot{\overline{x}}(t)\\ \dot{\overline{y}}(t)\\ \dot{\overline{\Psi}}(t)\end{bmatrix}, \tag{6}$$

where $\begin{bmatrix}\dot{\overline{x}}(t) & \dot{\overline{y}}(t) & \dot{\overline{\Psi}}(t)\end{bmatrix}^{\mathrm{T}}$ and $\begin{bmatrix}\overline{x}(t) & \overline{y}(t) & \overline{\Psi}(t)\end{bmatrix}^{\mathrm{T}}$ are calculated using a pseudo-differentiator from the command $\begin{bmatrix}x_{\mathrm{com}}(t) & y_{\mathrm{com}}(t) & \Psi_{\mathrm{com}}(t)\end{bmatrix}^{\mathrm{T}}$. The general form of a second-order pseudo-differentiator is illustrated by the following generic state space model for $x(t)$.

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix}\overline{x}(t)\\ \dot{\overline{x}}(t)\end{bmatrix} = \begin{bmatrix}0 & 1\\ -a_{d1}(t) & -a_{d2}(t)\end{bmatrix}\begin{bmatrix}\overline{x}(t)\\ \dot{\overline{x}}(t)\end{bmatrix} + \begin{bmatrix}0\\ a_{d1}(t)\end{bmatrix}x_{\mathrm{com}}(t), \tag{7}$$

where $\overline{x}(t)$ is a lowpass-filtered $x_{\mathrm{com}}(t)$, and $\dot{\overline{x}}(t)$ is the approximated derivative of $x_{\mathrm{com}}(t)$. For a fixed bandwidth pseudo-differentiator, $a_{d1}(t) = \omega_{n,\mathrm{diff}}^2, a_{d2}(t) = 2\zeta\omega_{n,\mathrm{diff}}$, where ζ is the damping ratio; $\omega_{n,\mathrm{diff}}$ is the natural frequency, which is proportional to the bandwidth of the lowpass filter that attenuates high-frequency gain, thereby making the pseudo-differentiator causal and realizable. This pseudo-differentiator can also be used as an automatic command shaping filter with time-varying bandwidth. It will be further discussed in Section 5.2.

Defining

$$\begin{bmatrix}e_x & e_y & e_\Psi\end{bmatrix}^{\mathrm{T}} = \begin{bmatrix}x(t) & y(t) & \Psi(t)\end{bmatrix}^{\mathrm{T}} - \begin{bmatrix}\overline{x}(t) & \overline{y}(t) & \overline{\Psi}(t)\end{bmatrix}^{\mathrm{T}}$$
$$\begin{bmatrix}\tilde{u} & \tilde{v} & \tilde{r}\end{bmatrix}^{\mathrm{T}} = \begin{bmatrix}u & v & r\end{bmatrix}^{\mathrm{T}} - \begin{bmatrix}\overline{u}(t) & \overline{v}(t) & \overline{r}(t)\end{bmatrix}^{\mathrm{T}}$$

and linearizing (5) along the nominal trajectories $[\overline{x}(t)\ \overline{y}(t)\ \overline{\Psi}(t)]^{\mathrm{T}}$ and the nominal input $[\overline{u}(t)\ \overline{v}(t)\ \overline{r}(t)]^{\mathrm{T}}$ yields the error dynamics

$$\begin{bmatrix}\dot{e}_x\\ \dot{e}_y\\ \dot{e}_\Psi\end{bmatrix} = A_1(t)\begin{bmatrix}e_x\\ e_y\\ e_\Psi\end{bmatrix} + B_1(t)\begin{bmatrix}\tilde{u}\\ \tilde{v}\\ \tilde{r}\end{bmatrix} \tag{8}$$

where

$$A_1(t) = \begin{bmatrix}0 & 0 & -\overline{u}(t)\sin\overline{\Psi}(t) - \overline{v}(t)\cos\overline{\Psi}(t)\\ 0 & 0 & \overline{u}(t)\cos\overline{\Psi}(t) - \overline{v}(t)\sin\overline{\Psi}(t)\\ 0 & 0 & 0\end{bmatrix}$$
$$B_1(t) = \begin{bmatrix}\cos\overline{\Psi}(t) & -\sin\overline{\Psi}(t) & 0\\ \sin\overline{\Psi}(t) & \cos\overline{\Psi}(t) & 0\\ 0 & 0 & 1\end{bmatrix}.$$

Now, a proportional-integral (PI) feedback control law is designed to stabilize the tracking error.

$$\begin{bmatrix}\tilde{u}\\ \tilde{v}\\ \tilde{r}\end{bmatrix} = -K_{\mathrm{P1}}\begin{bmatrix}e_x\\ e_y\\ e_\Psi\end{bmatrix} - K_{\mathrm{I1}}\begin{bmatrix}\int e_x(t)\mathrm{d}t\\ \int e_y(t)\mathrm{d}t\\ \int e_\Psi(t)\mathrm{d}t\end{bmatrix}. \tag{9}$$

Define the augmented outer-loop tracking error vector by

$$\gamma = \begin{bmatrix}\gamma_1 & \gamma_2 & \gamma_3 & \gamma_4 & \gamma_5 & \gamma_6\end{bmatrix}^{\mathrm{T}} = \begin{bmatrix}\int e_x(t)\mathrm{d}t & \int e_y(t)\mathrm{d}t & \int e_\Psi(t)\mathrm{d}t & e_x & e_y & e_\Psi\end{bmatrix}^{\mathrm{T}}.$$

Then the closed-loop tracking error state equation can be written as

$$\dot{\gamma} = A_{1c}\gamma = \begin{bmatrix}\boldsymbol{O}_3 & \boldsymbol{I}_3\\ -B_1K_{I1} & A_1 - B_1K_{P1}\end{bmatrix}\gamma,$$

where $\boldsymbol{O}_3$ denotes the 3×3 zero matrix, and $\boldsymbol{I}_3$ denotes the 3×3 identity matrix. Now select K_{I1} and K_{P1} to achieve the desired closed-loop tracking error dynamics

$$A_{1c} = \begin{bmatrix}\boldsymbol{O}_3 & \boldsymbol{I}_3\\ \mathrm{diag}\begin{bmatrix}-a_{111} & -a_{121} & -a_{131}\end{bmatrix} & \mathrm{diag}\begin{bmatrix}-a_{112} & -a_{122} & -a_{132}\end{bmatrix}\end{bmatrix},$$

where, for time-invariant closed-loop dynamics, $a_{1j1} > 0$, $a_{1j2} > 0$, $j = 1, 2, 3$, are the coefficients of the desired closed-loop characteristic polynomial of each channel given by $\lambda^2 + a_{1j2}\lambda + a_{1j1}$. It can be verified that

$$K_{I1} = -B_1^{-1}\mathrm{diag}\begin{bmatrix}-a_{111} & -a_{121} & -a_{131}\end{bmatrix}$$
$$K_{P1} = B_1^{-1}\left(A_1 - \mathrm{diag}\begin{bmatrix}-a_{112} & -a_{122} & -a_{132}\end{bmatrix}\right). \tag{10}$$

The body rate command to the inner-loop is given by

$$\begin{bmatrix}u_{\mathrm{com}}\\ v_{\mathrm{com}}\\ r_{\mathrm{com}}\end{bmatrix} = \begin{bmatrix}\overline{u}\\ \overline{v}\\ \overline{r}\end{bmatrix} + \begin{bmatrix}\tilde{u}\\ \tilde{v}\\ \tilde{r}\end{bmatrix}. \tag{11}$$

3.2. Inner-loop controller

From (4), the nominal motor control input voltages $[\overline{E}_1(t)\ \overline{E}_2(t)\ \overline{E}_3(t)]^{\mathrm{T}}$ for the nominal body rate $[\overline{u}(t)\ \overline{v}(t)\ \overline{r}(t)]^{\mathrm{T}}$ are given by

$$\begin{bmatrix}\overline{E}_1\\ \overline{E}_2\\ \overline{E}_3\end{bmatrix} = (HB)^{-1}\frac{R\cdot R_a}{k_2n}G\begin{bmatrix}\dot{\overline{u}}\\ \dot{\overline{v}}\\ \dot{\overline{r}}\end{bmatrix} - (HB)^{-1}\frac{RR_a}{k_2n}\begin{bmatrix}\overline{r}\overline{v}\\ -\overline{r}\overline{u}\\ 0\end{bmatrix} + B^{\mathrm{T}}\frac{R_an}{k_2R}\left(\frac{k_2k_3}{R_a} + b_0\right)\begin{bmatrix}\overline{u}\\ \overline{v}\\ \overline{r}\end{bmatrix}, \tag{12}$$

where $[\dot{\overline{u}}(t)\ \dot{\overline{v}}(t)\ \dot{\overline{r}}(t)]^{\mathrm{T}}$ are calculated from $[\overline{u}(t)\ \overline{v}(t)\ \overline{r}(t)]^{\mathrm{T}}$ using the pseudo-differentiator given in (7).

Defining

$$\begin{bmatrix}e_u & e_v & e_r\end{bmatrix}^{\mathrm{T}} = \begin{bmatrix}u & v & r\end{bmatrix}^{\mathrm{T}} - \begin{bmatrix}\overline{u}(t) & \overline{v}(t) & \overline{r}(t)\end{bmatrix}^{\mathrm{T}}$$
$$\begin{bmatrix}\tilde{E}_1 & \tilde{E}_2 & \tilde{E}_3\end{bmatrix}^{\mathrm{T}} = \begin{bmatrix}E_1 & E_2 & E_3\end{bmatrix}^{\mathrm{T}} - \begin{bmatrix}\overline{E}_1 & \overline{E}_2 & \overline{E}_3\end{bmatrix}^{\mathrm{T}}$$

and linearizing (4) along the nominal trajectories $[\overline{u}(t)\ \ \overline{v}(t)\ \ \overline{r}(t)]^{\mathrm{T}}$ and the nominal motor control $[\overline{E}_1\ \ \overline{E}_2\ \ \overline{E}_3]^{\mathrm{T}}$ yields the linearized inner-loop tracking error dynamics

$$\begin{bmatrix}\dot{e}_u\\ \dot{e}_v\\ \dot{e}_r\end{bmatrix} = A_2(t)\begin{bmatrix}e_u\\ e_v\\ e_r\end{bmatrix} + B_2\begin{bmatrix}\tilde{E}_1\\ \tilde{E}_2\\ \tilde{E}_3\end{bmatrix}, \tag{13}$$

where

$$A_2(t) = G^{-1}\begin{bmatrix}0 & \overline{r}(t) & \overline{v}(t)\\ -\overline{r}(t) & 0 & -\overline{u}(t)\\ 0 & 0 & 0\end{bmatrix} - G^{-1}HBB^{\mathrm{T}}\left(\frac{k_2\cdot k_3}{R_a}+b_0\right)\frac{n^2}{R^2}$$

$$B_2 = G^{-1}HB\cdot\frac{k_2 n}{R\cdot R_a}.$$

Design the PI feedback control law by

$$\begin{bmatrix}\tilde{E}_1\\ \tilde{E}_2\\ \tilde{E}_3\end{bmatrix} = -K_{\mathrm{P2}}\begin{bmatrix}e_u\\ e_v\\ e_r\end{bmatrix} - K_{\mathrm{I2}}\begin{bmatrix}\int e_u(t)\mathrm{d}t\\ \int e_v(t)\mathrm{d}t\\ \int e_r(t)\mathrm{d}t\end{bmatrix} \tag{14}$$

and define the augmented inner-loop tracking error vector by

$$\eta = \begin{bmatrix}\eta_1 & \eta_2 & \eta_3 & \eta_4 & \eta_5 & \eta_6\end{bmatrix}^{\mathrm{T}} = \left[\int e_u(t)\mathrm{d}t \quad \int e_v(t)\mathrm{d}t \quad \int e_r(t)\mathrm{d}t \quad e_u \quad e_v \quad e_r\right]^{\mathrm{T}}.$$

Then the closed-loop tracking error state equation can be written as

$$\dot{\eta} = A_{2c}\eta = \begin{bmatrix}\boldsymbol{O}_3 & \boldsymbol{I}_3\\ -B_2K_{I2} & A_2 - B_2K_{P2}\end{bmatrix}\eta.$$

Now select K_{I2} and K_{P2} to achieve the desired closed-loop tracking error dynamics

$$A_{2c} = \begin{bmatrix}\boldsymbol{O}_3 & \boldsymbol{I}_3\\ \mathrm{diag}\begin{bmatrix}-a_{211} & -a_{221} & -a_{231}\end{bmatrix} & \mathrm{diag}\begin{bmatrix}-a_{212} & -a_{222} & -a_{232}\end{bmatrix}\end{bmatrix}$$

where $a_{2j1} > 0$, $a_{2j2} > 0$, $j = 1, 2, 3$, are the coefficients of the desired (time-invariant) closed-loop characteristic polynomial of each channel given by $\lambda^2 + a_{2j2}\lambda + a_{2j1}$. It can be verified that

$$K_{\mathrm{I2}} = -B_2^{-1}\mathrm{diag}\begin{bmatrix}-a_{211} & -a_{221} & -a_{231}\end{bmatrix} \tag{15}$$
$$K_{\mathrm{P2}} = B_2^{-1}\left(A_2 - \mathrm{diag}\begin{bmatrix}-a_{212} & -a_{222} & -a_{232}\end{bmatrix}\right).$$

Finally, the applied voltage to the motors is given by

$$\begin{bmatrix}E_1\\ E_2\\ E_3\end{bmatrix} = \begin{bmatrix}\overline{E}_1\\ \overline{E}_2\\ \overline{E}_3\end{bmatrix} + \begin{bmatrix}\tilde{E}_1\\ \tilde{E}_2\\ \tilde{E}_3\end{bmatrix}. \tag{16}$$

4. Position and orientation measurements using sensor fusion

In this section, the sensor fusion method for our omni-directional mobile robots is briefly described. Detailed design and test results are published in [33]. The sensor fusion method combines on-board encoder sensor and the global vision system measurements, thereby providing reliable and accurate position and orientation measurements. In [29], a similar sensor fusion method is developed for mobile robot using encoder, GPS and gyroscope. Different from [29], the sensor fusion method developed in this section employs a nonlinear Kalman filter technique which uses the nominal trajectory generated by the robot outer-loop controller pseudo-inverse to linearize the nonlinear robot kinematics. A gating technique is also used to remove the corrupted vision measurements.

To facilitate applying the Kalman filter algorithm, first the robot kinematics equation (5) is discretized using the forward Euler method with time-interval T

$$\begin{bmatrix}x[k]\\ y[k]\\ \Psi[k]\end{bmatrix} = \begin{bmatrix}x[k-1]\\ y[k-1]\\ \Psi[k-1]\end{bmatrix} + \begin{bmatrix}\cos\Psi[k-1]\cdot T & -\sin\Psi[k-1]\cdot T & 0\\ \sin\Psi[k-1]\cdot T & \cos\Psi[k-1]\cdot T & 0\\ 0 & 0 & T\end{bmatrix} \times \begin{bmatrix}u[k-1]\\ v[k-1]\\ r[k-1]\end{bmatrix}. \tag{17}$$

The Robocat body rate can be calculated from on-board sensor (motor encoder). The robot position and orientation can be determined from the vision system. The body rate measurement $\begin{bmatrix}\hat{u}[k] & \hat{v}[k] & \hat{r}[k]\end{bmatrix}^{\mathrm{T}}$ and vision system measurement $\begin{bmatrix}z_1[k] & z_2[k] & z_3[k]\end{bmatrix}^{\mathrm{T}}$ at time-step k are defined as

$$\begin{bmatrix}\hat{u}[k]\\ \hat{v}[k]\\ \hat{r}[k]\end{bmatrix} = \begin{bmatrix}u[k]\\ v[k]\\ r[k]\end{bmatrix} + \begin{bmatrix}w_1[k]\\ w_2[k]\\ w_3[k]\end{bmatrix},$$
$$\begin{bmatrix}z_1[k]\\ z_2[k]\\ z_3[k]\end{bmatrix} = \begin{bmatrix}x[k]\\ y[k]\\ \Psi[k]\end{bmatrix} + \begin{bmatrix}d_1[k]\\ d_2[k]\\ d_3[k]\end{bmatrix}, \tag{18}$$

where $\begin{bmatrix}w_1[k] & w_2[k] & w_3[k]\end{bmatrix}^{\mathrm{T}}$ is the body rate measurement noise, and $\begin{bmatrix}d_1[k] & d_2[k] & d_3[k]\end{bmatrix}^{\mathrm{T}}$ is the vision system noise. Both $\begin{bmatrix}w_1[k] & w_2[k] & w_3[k]\end{bmatrix}^{\mathrm{T}}$ and $\begin{bmatrix}d_1[k] & d_2[k] & d_3[k]\end{bmatrix}^{\mathrm{T}}$ are assumed to be zero-mean white noise with normal distribution, such that

$$p\left(\begin{bmatrix}w_1[k] & w_2[k] & w_3[k]\end{bmatrix}^{\mathrm{T}}\right) \sim N(0, Q[k])$$
$$p\left(\begin{bmatrix}d_1[k] & d_2[k] & d_3[k]\end{bmatrix}^{\mathrm{T}}\right) \sim N(0, R[k]),$$

where $Q[k] \in \mathbb{R}^{3\times3}$ is the body rate measurement covariance, and $R[k] \in \mathbb{R}^{3\times3}$ is the vision system observation noise covariance.

Table 5.1
Controller parameters

Outer-loop		
Pseudo-differentiator damping ratio and natural frequency (Maximum)	ζ	[0.7 0.7 0.7]
	$\omega_{n,\text{diff}}$ (rad/s)	[8 8 8]
Closed-loop	$\begin{bmatrix} a_{111} & a_{121} & a_{131} \end{bmatrix}$	[16 16 16]
Characteristic polynomial	$\begin{bmatrix} a_{112} & a_{122} & a_{132} \end{bmatrix}$	[7.2 7.2 7.2]
Closed-loop damping ratio and natural frequency (Maximum)	ζ	[0.9 0.9 0.9]
	ω_n (rad/s)	[4 4 4]
Inner-loop		
Pseudo-differentiator damping ratio and natural frequency (Maximum)	ζ	[0.7 0.7 0.7]
	$\omega_{n,\text{diff}}$ (rad/s)	[40 40 40]
Closed-loop	$\begin{bmatrix} a_{211} & a_{221} & a_{231} \end{bmatrix}$	[400 400 400]
Characteristic polynomial	$\begin{bmatrix} a_{212} & a_{222} & a_{232} \end{bmatrix}$	[36 36 36]
Closed-loop damping ratio and natural frequency (Maximum)	ζ	[0.9 0.9 0.9]
	ω_n (rad/s)	[20 20 20]

In practice, the vision system may lose frames or misidentify the robot on the field of play. Gating is a technique for eliminating most unlikely outlier measurements [24]. There are several commonly used gating algorithms. In this paper, a simple rectangular gate is used. The nonlinear Kalman filter algorithm for robot location and orientation estimates with gating is summarized below.

Step 1: At time-step k, estimate robot position and orientation from the on-board sensor body rate measurement

$$\begin{bmatrix} x[k]^- \\ y[k]^- \\ \Psi[k]^- \end{bmatrix} = \begin{bmatrix} \hat{x}[k-1] \\ \hat{y}[k-1] \\ \hat{\Psi}[k-1] \end{bmatrix} + \begin{bmatrix} T\cos\hat{\Psi}[k-1] & -T\sin\hat{\Psi}[k-1] & 0 \\ T\sin\hat{\Psi}[k-1] & T\cos\hat{\Psi}[k-1] & 0 \\ 0 & 0 & T \end{bmatrix} \times \begin{bmatrix} \widehat{u}[k-1] \\ \widehat{v}[k-1] \\ \widehat{r}[k-1] \end{bmatrix}, \tag{19}$$

where $\begin{bmatrix} x[k]^- & y[k]^- & \Psi[k]^- \end{bmatrix}^{\mathrm{T}}$ is a priori location estimation. Then calculate the prediction covariance as

$$P[k]^- = A[k]\cdot P[k-1]\cdot A[k]^{\mathrm{T}} + W[k]\cdot Q[k-1]\cdot W[k]^{\mathrm{T}}, \tag{20}$$

where

$$A[k] = \begin{bmatrix} 1 & 0 & -T\sin\overline{\Psi}[k-1]\,\overline{u}[k-1] - T\cos\overline{\Psi}[k-1]\,\overline{v}[k-1] \\ 0 & 1 & T\cos\overline{\Psi}[k-1]\,\overline{u}[k-1] - T\sin\overline{\Psi}[k-1]\,\overline{v}[k-1] \\ 0 & 0 & 1 \end{bmatrix}$$

$$W[k] = \begin{bmatrix} T\cos\overline{\Psi}[k-1] & -T\sin\overline{\Psi}[k-1] & 0 \\ T\sin\overline{\Psi}[k-1] & T\cos\overline{\Psi}[k-1] & 0 \\ 0 & 0 & T \end{bmatrix}.$$

Step 2: Read the vision system measurement $[z_1[k]\ z_2[k]\ z_3[k]]^{\mathrm{T}}$. If the vision system data is not available, go to Step 4. If the vision system data is available, calculate the innovation residue using (21):

$$\begin{bmatrix} e_{z_1}[k] \\ e_{z_2}[k] \\ e_{z_3}[k] \end{bmatrix} = \begin{bmatrix} z_1[k] \\ z_2[k] \\ z_3[k] \end{bmatrix} - \begin{bmatrix} x[k]^- \\ y[k]^- \\ \Psi[k]^- \end{bmatrix}. \tag{21}$$

The rectangular Gating is defined as

$$|e_{z_i}[k]| \le 3\sqrt{\sigma_{R_i}^2 + \sigma_{P_i}^2}, \quad i = 1, 2, 3, \tag{22}$$

where $\sigma_{R_i}^2$ is the diagonal element of the vision system noise covariance, and $\sigma_{P_i}^2$ is the ith diagonal element of the prediction covariance $P^-[k]$. If all innovation residues satisfy the above gating condition, the vision system is considered valid, and proceed to Step 3; otherwise, goto Step 4.

Step 3: Correction with valid vision data

$$K[k] = P[k]^-\left(P[k]^- + R[k]\right)^{-1} \tag{23}$$
$$P[k] = (I - K[k])\,P[k]^-.$$

The posterior estimation is

$$\begin{bmatrix} \hat{x}[k] \\ \hat{y}[k] \\ \hat{\Psi}[k] \end{bmatrix} = \begin{bmatrix} x[k]^- \\ y[k]^- \\ \Psi[k]^- \end{bmatrix} + K[k]\begin{bmatrix} e_{z_1}[k] \\ e_{z_2}[k] \\ e_{z_3}[k] \end{bmatrix}. \tag{24}$$

Goto Step 1.

Step 4: Calculate the prediction covariance without correction.

$$\begin{bmatrix} \hat{x}[k] \\ \hat{y}[k] \\ \hat{Psi}[k] \end{bmatrix} = \begin{bmatrix} x[k]^- \\ y[k]^- \\ \Psi[k]^- \end{bmatrix} \tag{25}$$

$$P[k] = P[k]^-. \tag{26}$$

Goto Step 1. ◇

The estimated measurement $\left[\hat{x}[k], \hat{y}[k], \hat{z}[k]\right]$ is used for outer-loop feedback at the time-step k. In (20), $A[k]$ and $W[k]$ are generated by linearizing measurement error along the nominal trajectory. The nonlinear Kalman filter employed in this paper is motivated by the nonlinear observer design

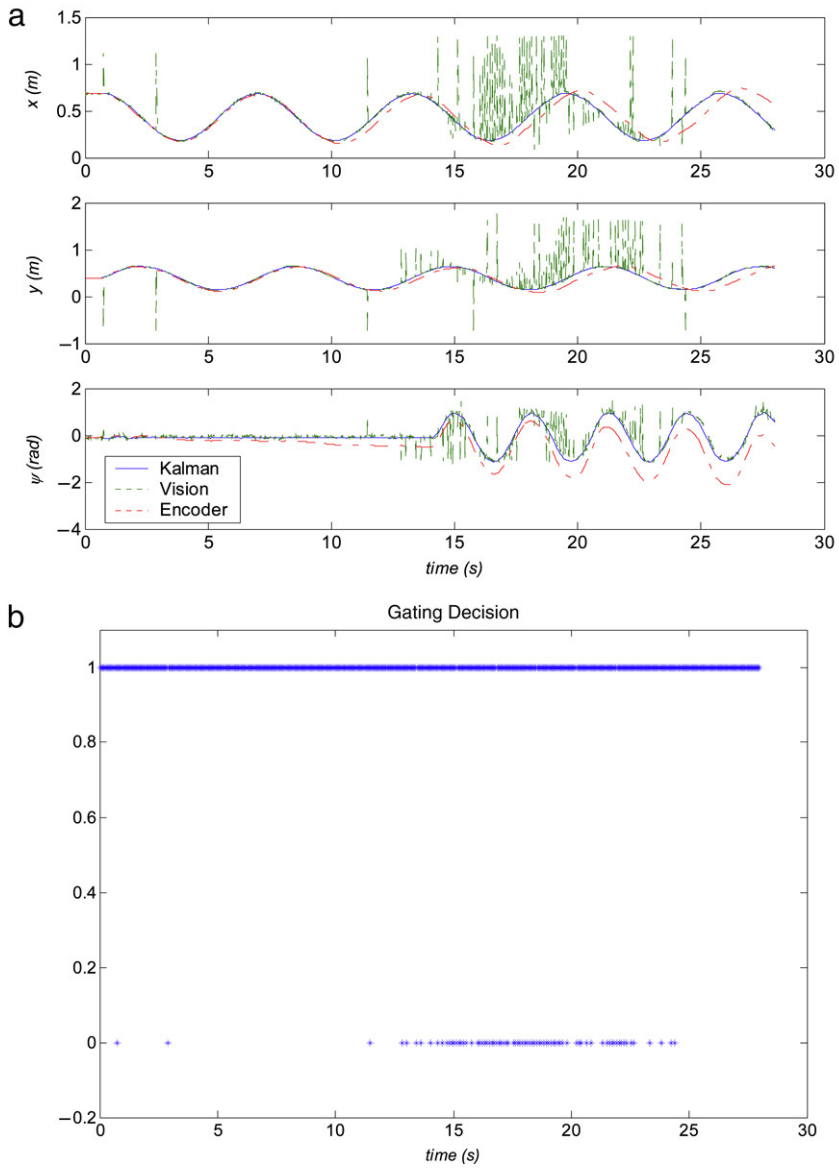


Fig. 4. Sensor fusion result. (a) Position and orientation estimation comparison. (b) Gating decision.

based on trajectory linearization [20]. It is similar to linearized Kalman filter [21] in that both techniques use an open-loop nominal trajectory for linearization. But they differ in that the closed-loop TLC controller ensures that the actual robot trajectory stays close to the nominal trajectory. It is different from extended Kalman filter (EKF) algorithm [22,23], where the estimation of the filter is used as the nominal trajectory to linearize the error dynamics.

5. Controller parameter tuning and adaptive command shaping

5.1. Parameter tuning

The unique structure of TLC provides robust stability and performance along trajectory without interpolation of controller gains. Thus the controller parameter tuning is relatively easier than the linear controller tuning [14]. One set of fixed controller parameters can be used for all command trajectories. The controller parameters in the TLC controller are the pseudo-differentiator bandwidths and the closed-loop feedback gains of both inner-loop and outer-loop controllers. The pseudo-differentiator bandwidths of both inner-loop and outer-loop for a feasible command are set high to reduce the tracking error due to phase delay. The closed-loop bandwidth tuning follows these guidelines: (1) The outer-loop controller bandwidth should satisfy the robot tracking requirement. (2) The inner-loop controller bandwidth should be at least three times higher than the outer-loop to satisfy the singular perturbation assumption. (3) The inner-loop controller bandwidth should be at least three times smaller than the motor and sensor bandwidths. (4) The sampling rate must be at least twice the lowest bandwidth of the components in the closed-loop system. (5) The closed-loop bandwidth is kept as low as possible when satisfying other requirements, in order to reduce the control power consumption and noise.

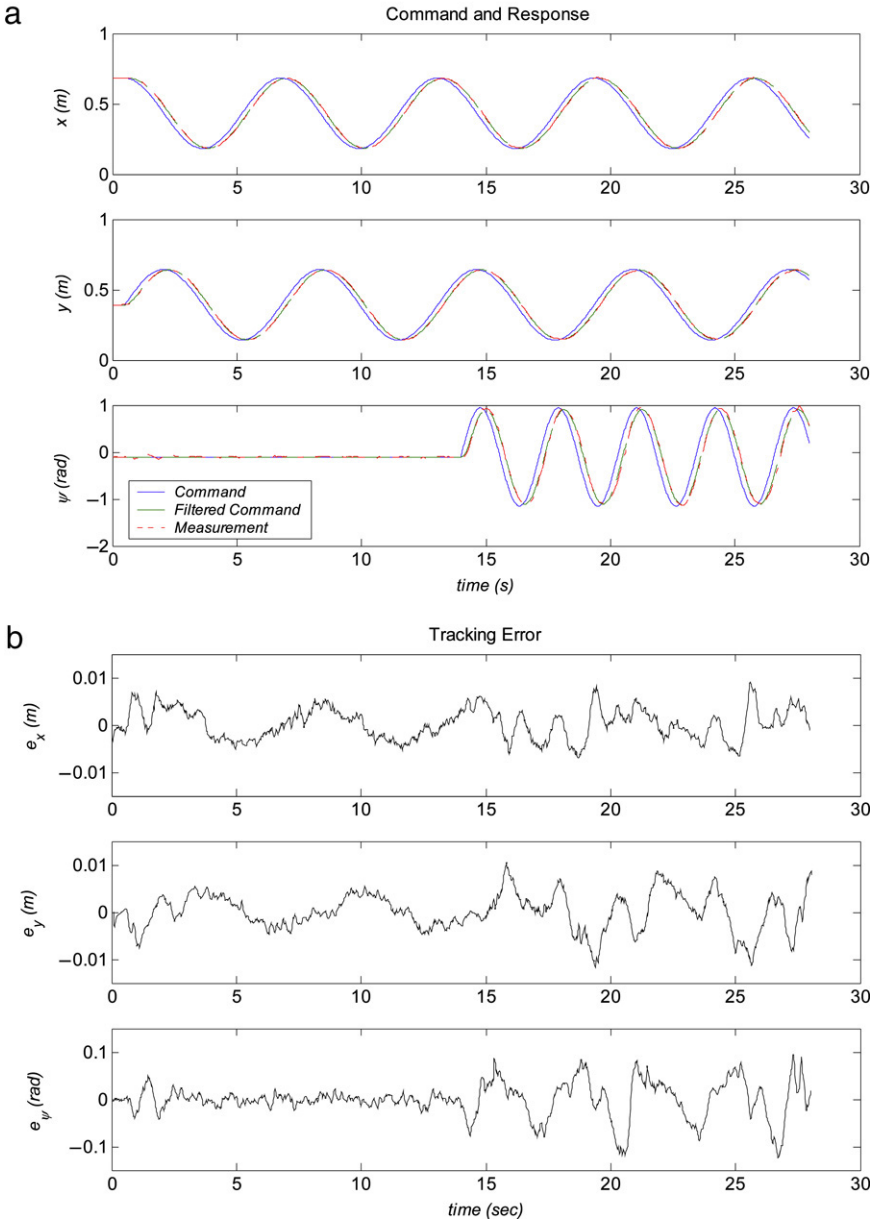


Fig. 5. Circle trajectory with orientation. (a) Tracking response. (b) Tracking errors. (c) Motor voltages. (d) Robot trajectory.

The Robocat requires that the tracking error for a fixed position command is on the order of 10^{-3} m (1 mm), and for dynamic trajectories is on the order of 10^{-2} m (1 cm). The orientation angle tracking error should be less than 6 deg (0.1 rad). The motor mechanical time constant is 11 ms. Thus the motor bandwidth is about 90 rad/s without considering mechanical latch. The motor speed is measured by applying a pseudo-differentiator on the motor encoder reading. The bandwidth of the pseudo-differentiator is initially set to 40 rad/s, which is considered as the sensor bandwidth. Thus the inner-loop bandwidth should be set to less than 13.3 rad/s.

In the parameter tuning, the damping ratios for both outer-loop and inner-loop closed-loop dynamics are set as 0.9. At this damping ratio, the bandwidth is approximately the same as ω_n. The ratio between inner-loop and outer-loop closed-loop bandwidth is set as 5:1, in order to satisfy the singular perturbation requirement. Thus the only parameters to tune are the outer-loop closed-loop characteristic bandwidths. The tuning process started with a small bandwidth that can achieve controller stability. Then the bandwidth is gradually increased to meet the tracking performance requirement. It is observed that higher bandwidth can reduce tracking error. However, a set of very high closed-loop bandwidths, combined with the backlash and dead zones in the robot mechanical components and limited measurement accuracy, induced controller chattering. The tuned controller parameters are listed in Table 5.1.

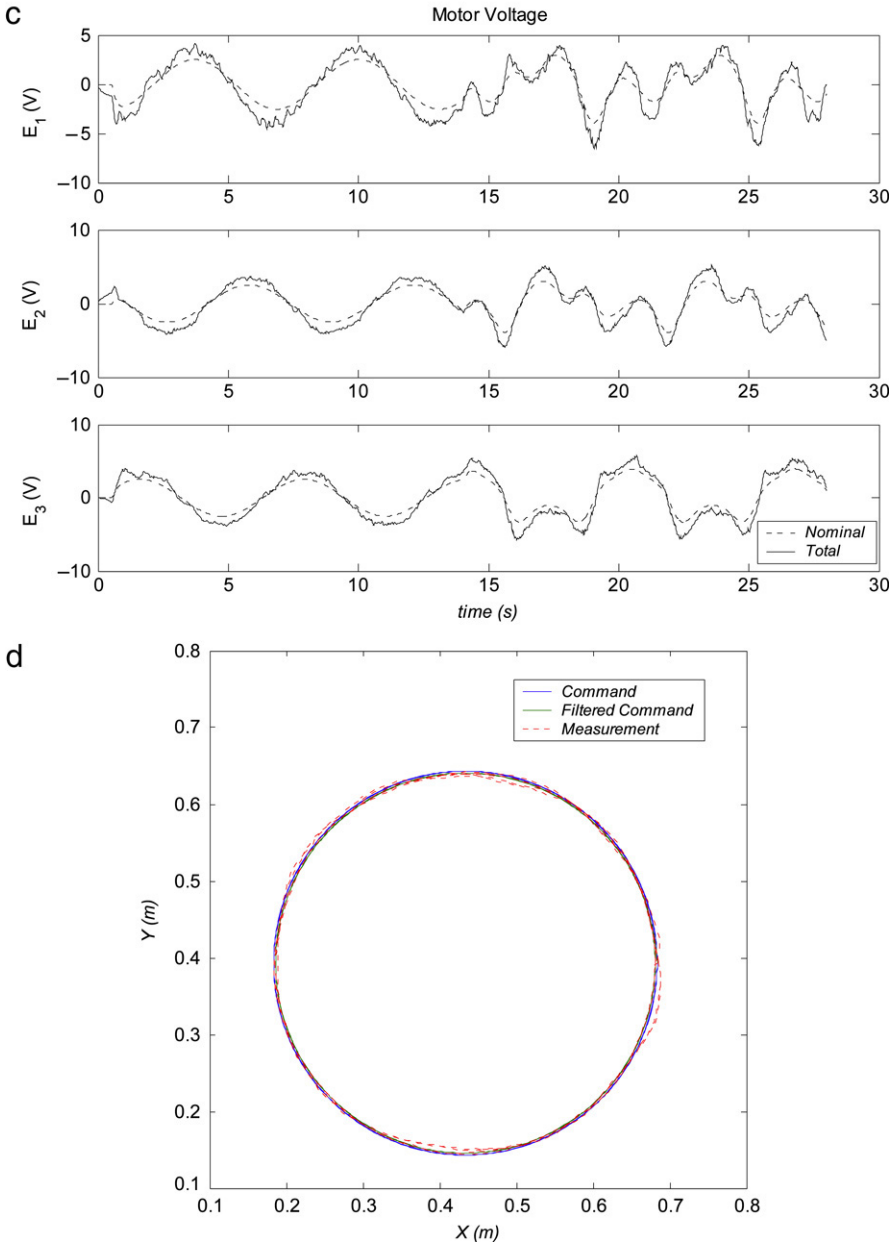


Fig. 5. (*continued*)

5.2. Time-varying bandwidth command shaping filter

In the controller design, the command trajectory is assumed to be a trackable trajectory within the robot physical limitations. However, in a dynamic environment, such as in Robocup competition, an infeasible trajectory command may be generated, which usually causes motor control input saturation and integral control windup. This problem is typically dealt by an optimal path planning algorithm, which is usually difficult to design and computation intensive in implementation. We propose an ad hoc, yet effective method to deal with this problem using the outer-loop pseudo-differentiator with a time-varying bandwidth as a command shaping filter to render the command into a feasible one. A time-varying bandwidth pseudo-differentiator can be represented by the following generic state space model for x command.

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix}\overline{x}(t)\\ \dot{\overline{x}}(t)\end{bmatrix} = \begin{bmatrix}0 & 1\\ -\omega_{n,\mathrm{diff}}^2(t) & -\left(2\zeta\omega_{n,\mathrm{diff}}(t) - \dfrac{\dot{\omega}_{n,\mathrm{diff}}(t)}{\omega_{n,\mathrm{diff}}(t)}\right)\end{bmatrix} \times \begin{bmatrix}\overline{x}(t)\\ \dot{\overline{x}}(t)\end{bmatrix} + \begin{bmatrix}0\\ \omega_{n,\mathrm{diff}}^2(t)\end{bmatrix} x_{\mathrm{com}}(t). \tag{27}$$

Note that (27) differs from the familiar quadratic synthesis formula (7) with constant bandwidth by a term $-\dot{\omega}_{n,\mathrm{diff}}(t)/\omega_{n,\mathrm{diff}}(t)$. A simpler implementation is employed in the robot controller whereby a constant rate is used. In the discrete time implementation, ω_{dec} denotes a constant

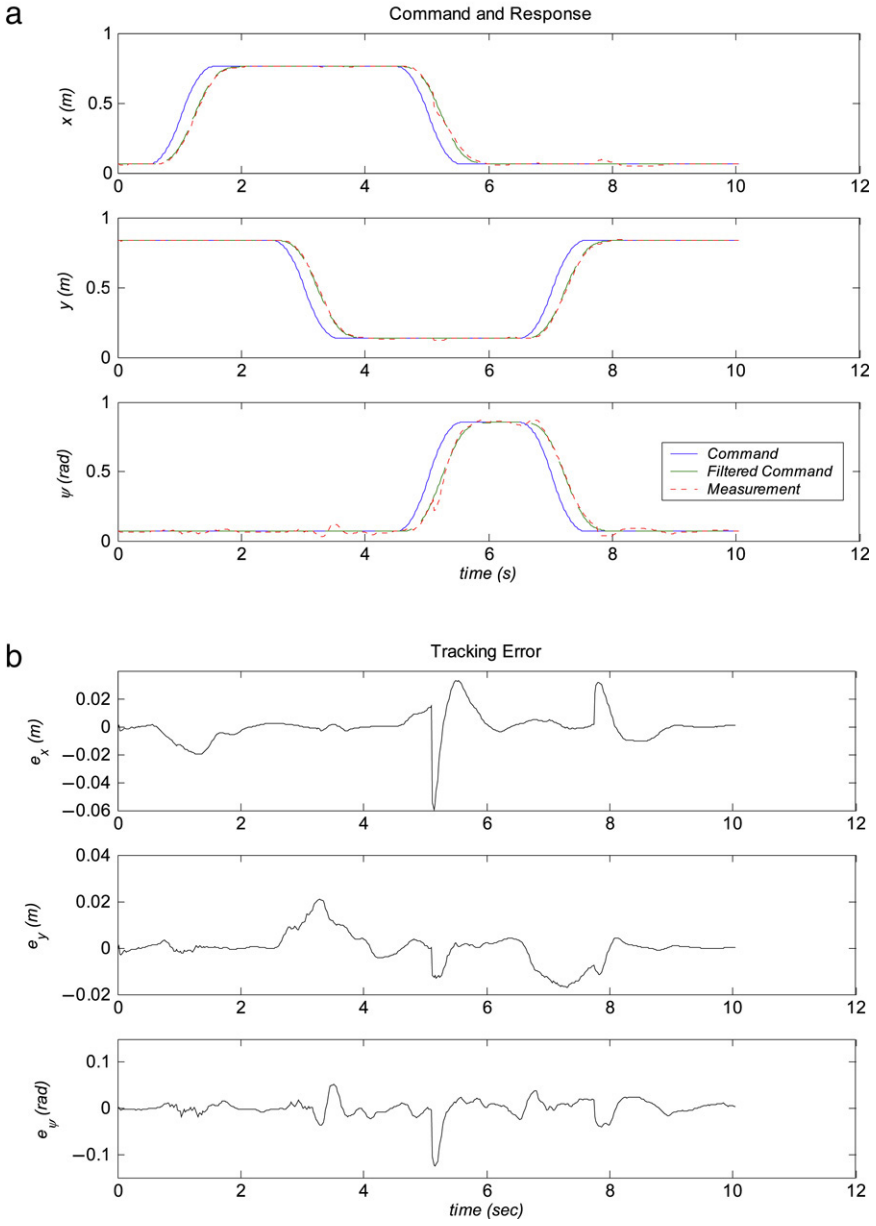


Fig. 6. Square command tracking performance. (a) Tracking response. (b) Tracking errors. (c) Motor voltages. (d) Robot trajectory.

decrement of the bandwidth per control cycle, whereas ω_{inc} denotes a constant increment per control cycle. When applying such a time-varying command shaping filter, the nominal motor inputs $\left[\overline{E}_1\ \overline{E}_2\ \overline{E}_3\right]$ are used to determine how to adjust the bandwidth. If any one of the three nominal voltages exceeds the maximum motor voltage value (12 V), the bandwidth is decreased. If all nominal voltages are under 10 V, the bandwidth is increased. Note that the filter bandwidth only changes between $\omega_{n,\text{diff(min)}}$ and $\omega_{n,\text{diff(max)}}$ where the $\omega_{n,\text{diff(max)}}$ is the same as the nominal $\omega_{n,\text{diff}}$ in Table 5.1. The initial bandwidths of the pseudo-differentiators are set as $\omega_{n,\text{diff(max)}}$. The time-varying bandwidth pseudo-differentiator parameters are listed in the Table 5.2.

The time-varying command shaping filter can also be used to simplify the path planning. A dynamic path planning algorithm for an omni-directional mobile robots needs to consider both velocity and acceleration constrains that are mainly determined by the maximum motor voltages. To generate an optimal 3DOF dynamic path is very complex. In [17,14], only a point-to-point optimal path is discussed without considering the rotation. The command shaping filter is capable of rendering an infeasible path into a desired feasible one in which motors work close to their maximum voltages wherever possible. In Section 6, real-time Test 3 illustrates the effectiveness of the proposed command shaping filter method, where a large 3DOF step command is used as a worst case command without a path planning.

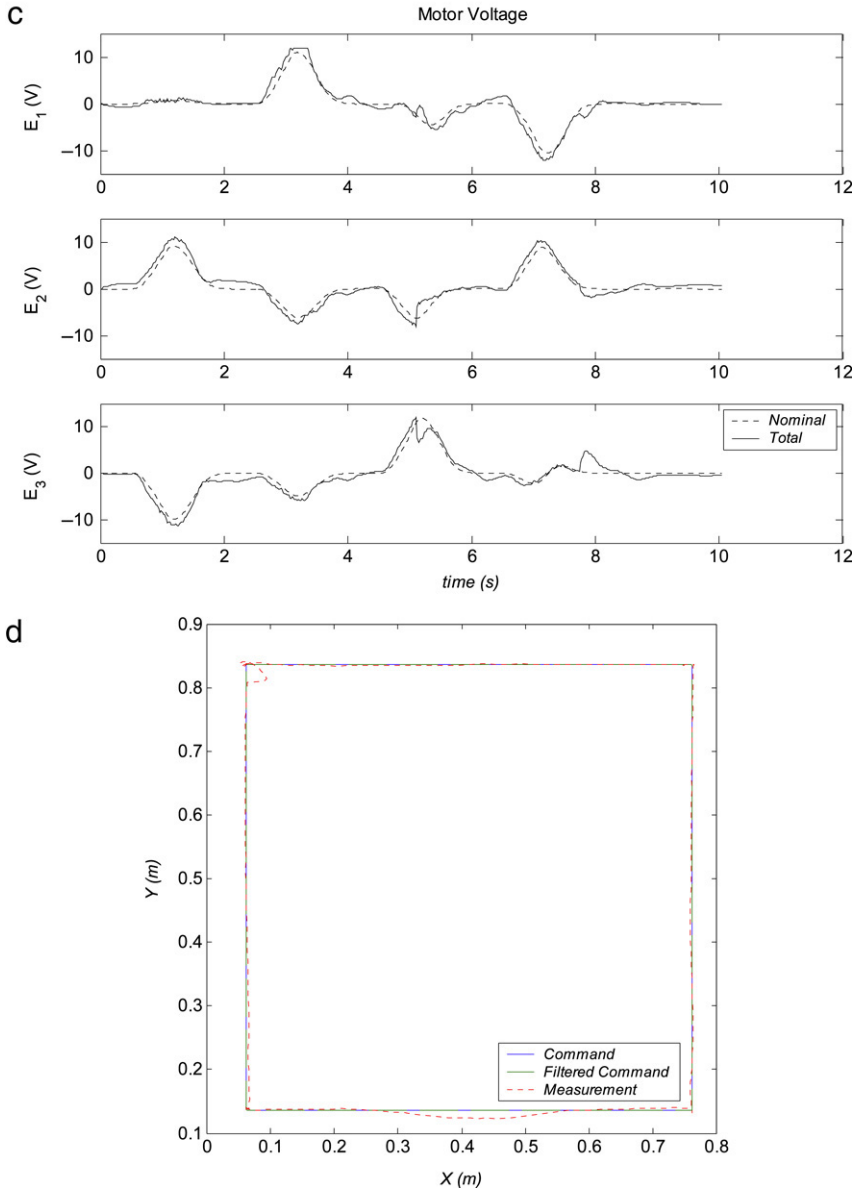


Fig. 6. (*continued*)

Table 5.2
Outer-loop time-varying pseudo-differentiator parameters

$\omega_{n,\text{diff(min)}}$ (rad/s)	[0.8 0.8 0.8]
$\omega_{n,\text{diff(max)}}$ (rad/s)	[8 8 8]
ω_{dec} (rad/s)	0.6
ω_{inc} (rad/s)	0.2

6. Real-time test results

The omni-directional mobile robot TLC controller was first verified in Simulink simulation, and then tested in a real-time hardware-in-the-loop (HIL) simulation. In the HIL test, Quanser's Wincon, plus Mathworks' Simulink and Real-time Workshop are used to develop a prototype of the real-time TLC controller for the robot of Fig. 1. A Cognachrome 2000 vision system with a YC-100 CCD camera is used to sense the robot position and orientation.

Based on HIL test results, the TLC controller has been implemented on the Robocat robots and has been used in Robocup 2003, Robocup 2004 and the first year American Open. The TLC controller is programmed using C++ in Linux on the on-board PC104 computer. The sampling rate of the controller is set to 100 HZ. Similar to HIL simulation, the body rate is measured using motor shaft encoders, while the robot location is provided by a roof camera vision system. In these real-time competitions, the TLC robot controller provides accurate and robust trajectory following performance.

In this section, results of three real-time HIL tests are presented.

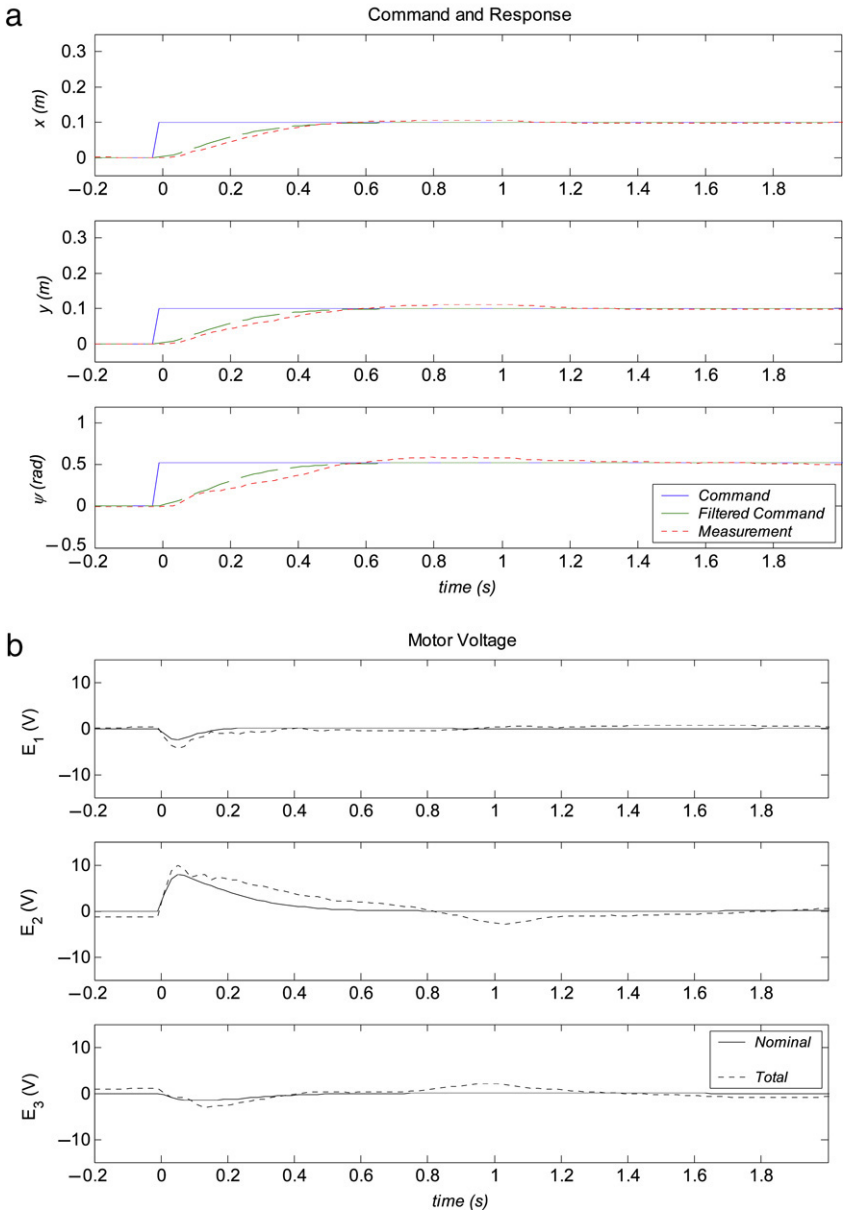


Fig. 7. Step response to [0.1 m 0.1 m 30 deg]. (a) Tracking response. (b) Motor voltage.

Test 1: In this test, the command is to accelerate from the initial position and draw a circle of 0.25 m radius at an angular rate of 2π rad/s. In the first 14 s, the robot orientation angle is fixed. After 14 s, the robot orientation command is a sinusoidal curve between -45 deg and 45 deg with the frequency 4π rad/s. The overall command is a 3DOF trajectory with varying velocity, acceleration, orientation angle and varying load on each wheel. The sensor fusion performance is illustrated in Fig. 4. In Fig. 4(a), the position and orientation measurements by three different methods are compared. The solid line is the sensor fusion estimation, the dotted line is the vision system data, and the dashed line is the encoders estimation. It can be seen that the encoder estimation diverged slowly, while the vision measurement has many corrupted data. The sensor fusion uses the vision system to continuously calibrate the encoder estimation and discard the invalid vision data. Fig. 4(b) shows the gating decision, where 1 means acceptance of the vision data, 0 means rejection of the vision data.

The tracking response, tracking error, motor voltages and robot trajectory are illustrated in Fig. 5. Fig. 5(a) shows the trajectory command, the filtered nominal command and the robot response. In this figure, the robot position and orientation are obtained from sensor fusion. It can be seen that the robot can precisely follow the filtered command trajectory. Fig. 5(b) shows that the peak tracking error (difference between the robot response and the filtered command) is under 1 cm bound and the orientation angle tracking error is less than 6 deg; both satisfied the controller performance requirements. It should be noted that the position tracking errors are achieved on the same order with and without a robot rotation. This shows that the controller has effectively decoupled the rotational

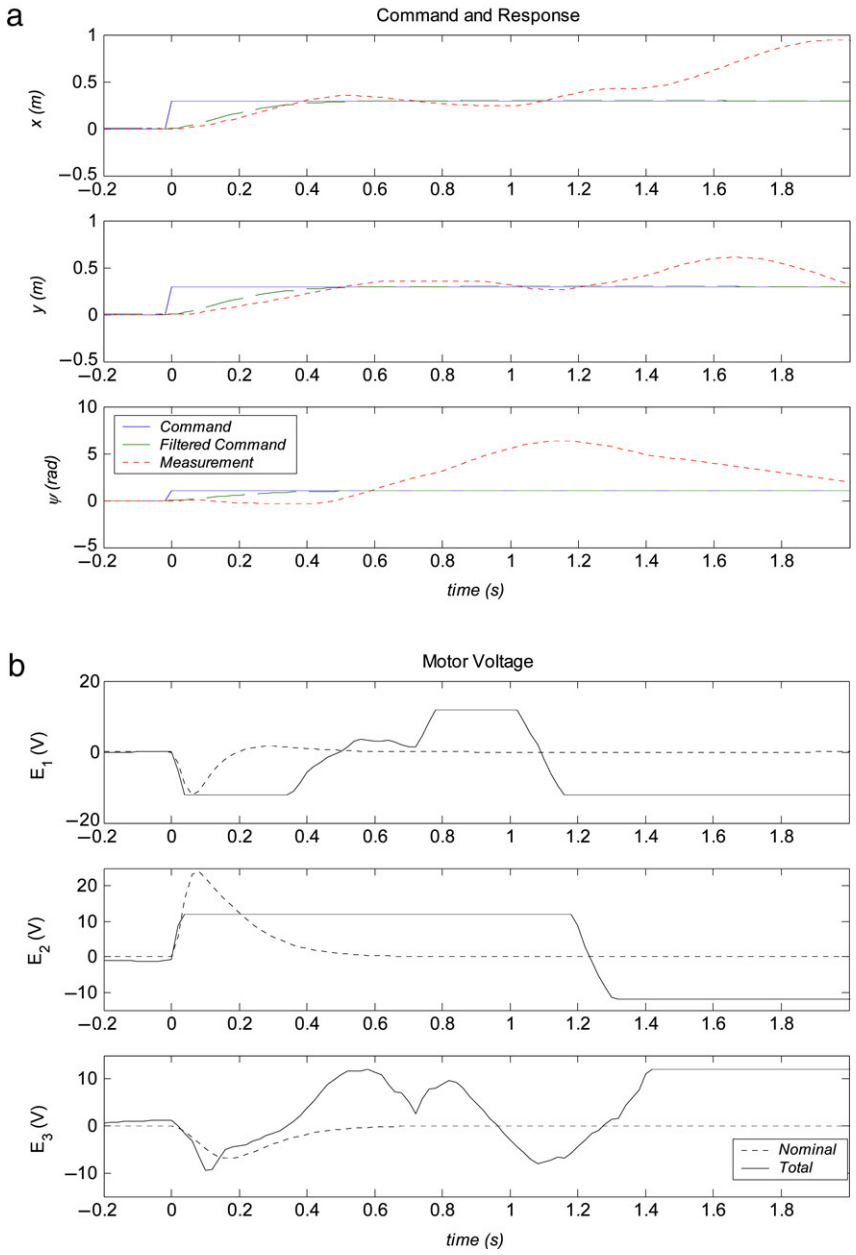


Fig. 8. Step response to [0.3 m 0.3 m 60 deg] with fixed pseudo-differentiator bandwidth. (a) Tracking response. (b) Motor voltage.

motion from the translational motion. Fig. 5(c) shows the nominal voltage and total voltage applied to each motor. The difference between them is the feedback control input to compensate for the disturbance and modeling error. In this test, the nominal voltage does not exceed the predefined limit, thus the time-varying command filter bandwidth remains at its maximum value. Fig. 5(d) shows the robot response trajectory. The difference between the robot actual trajectory and the commanded trajectory is small.

Test 2: In this test, the robot is commanded to follow a square trajectory. On two sides of the square, the robot orientation is fixed. On the other two sides, the robot is commanded to rotate 45 deg for each side. The commanded trajectory has acceleration and deceleration of 2.5 m/s^2. This value is close to the largest possible acceleration based on the robot experiment. The tracking response, tracking error, motor voltages and robot trajectory are illustrated in Fig. 6. The peak tracking error in this test is larger than the previous test since both encoder and vision system have larger sensor noise at higher acceleration and velocity, which reduces the measurement accuracy.

Test 3: This test is designed to illustrate the function of time-varying bandwidth command shaping filter. A large step command in x, y and Ψ is used to represent those abrupt command trajectories that violate the robot dynamic constraints. The motor voltages in the test saturate at ± 12 V. Three different test scenarios are presented. Without loss of generality, the robot initial positions are all shifted to [0 m 0 m 0 deg] in the plot. The first scenario is a relatively small step command [0.1 m 0.1 m 30 deg]. From Fig. 7(a) and (b), it can be seen that steady position tracking error is of

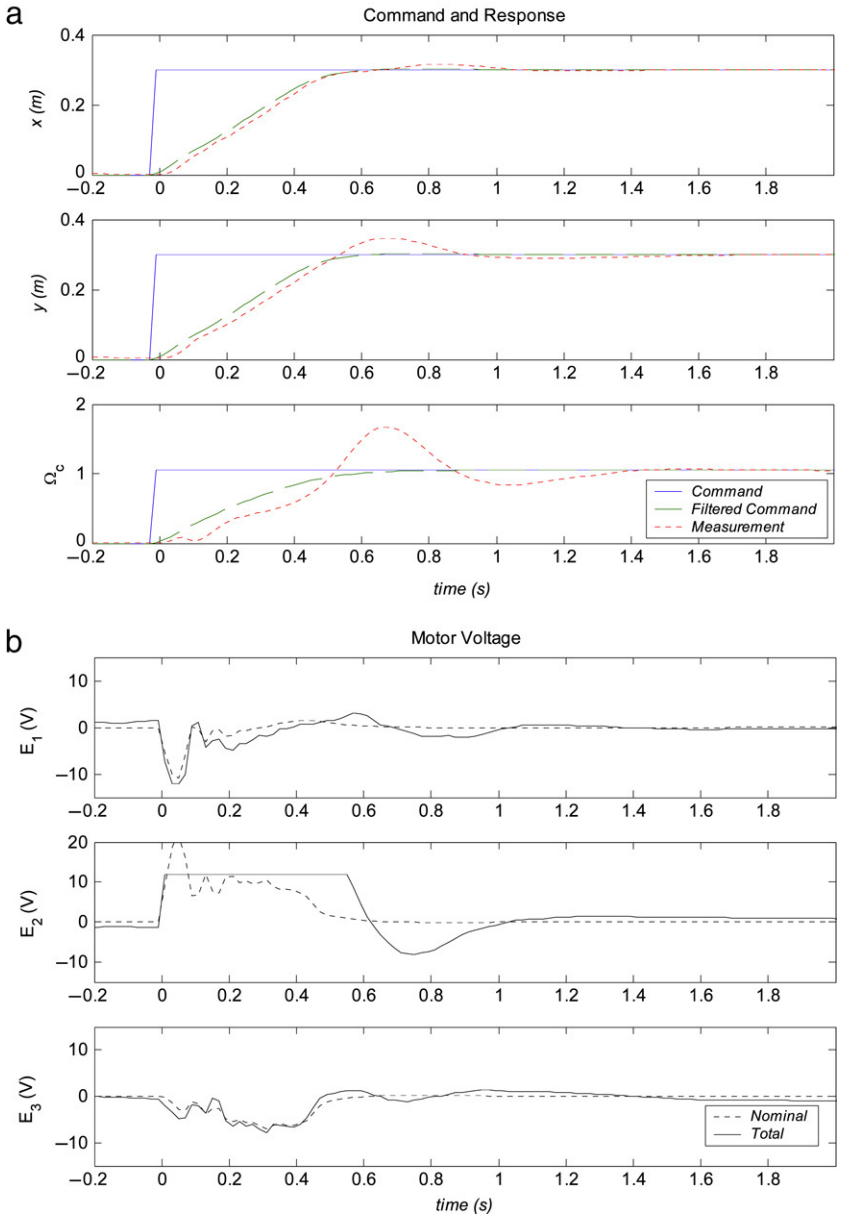


Fig. 9. Step response to [0.3 m 0.3 m 60 deg] time-variant pseudo-differentiator bandwidth. (a) Tracking response. (b) Motor voltage. (c) Time-varying bandwidth.

the order of 10^{-3} m and the peak orientation tracking error is less than 2 deg. Fig. 7(c) shows that both nominal motor voltage and total voltage are under the motor voltage limit, thus the command filter bandwidth is kept at its maximum value. The second and third scenarios are a larger step command [0.3 m 0.3 m 60 deg]. Fig. 8 shows the test results with the command filter bandwidth fixed at its maximum value. In Fig. 8(b), it can be seen that the voltages for motor 1 and 2 are saturated first, which causes the output to deviate from the command trajectory significantly, as shown in Fig. 8(a). Fig. 9 shows the test results with time-varying bandwidth. With time-varying bandwidth command filter, the controller is able to track the large step command, as shown in Fig. 9(a). Fig. 9(b) and (c) shows that once the nominal voltage reached the maximum value, the command filter bandwidth is reduced, which prevents loss of stability. Fig. 9(b) shows that although the motor voltage is saturated for a short time, the controller maintains stability. At the steady state, the command filter bandwidth is increased automatically, which guarantees the steady tracking performance. It can be seen the steady state tracking error in Fig. 9 is on same order as that of scenario 1 in Fig. 7. It is noted that in these tests, all position and orientation measurement are estimated by the sensor fusion method.

In these real-time tests, the controller parameters have been kept the same as listed in Tables 5.1 and 5.2. It can be concluded that the designed robot controller, with a set of fixed parameters, is capable of following a large class

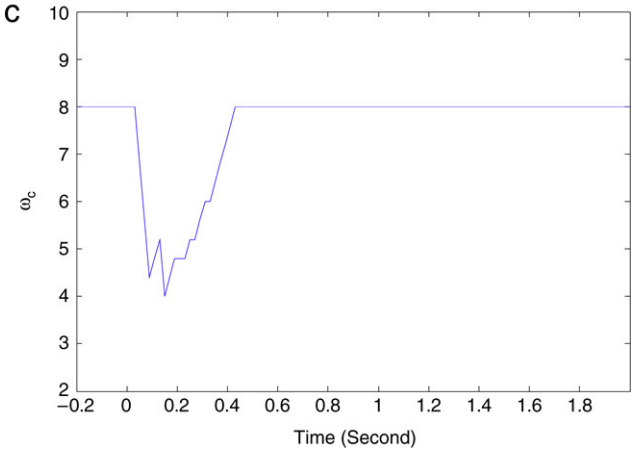


Fig. 9. (*continued*)

of challenging 3DOF command trajectories with satisfactory performance. Additional test results, which are omitted here due to page limitation, have also demonstrated that the controller performance is robust to modeling errors, such as asymmetric robot mass distribution and different field of play friction characteristics.

7. Conclusion

In this paper, nonlinear equations of motion for an omni-directional mobile robot have been derived including rigid body kinematics, dynamics and motor dynamics. Based on this model, a novel nonlinear controller using Trajectory Linearization Control (TLC) has been designed. The robot controller employs a dual-loop structure. The outer-loop kinematics controller adjusts the robot position and orientation to follow commanded trajectory. The inner-loop dynamics controller follows the body rate command from the outer-loop controller. A sensor fusion scheme for position and orientation measurements using both a global vision system and an on-board odometry sensor has been presented which improve the controller tracking performance. Furthermore, a time-varying command shaping filter is used to improve the controller robustness to an abrupt command trajectory and to simplify path planning. The controller parameter tuning guideline has been presented. Real-time test performances have shown that with a set of fixed controller parameters, the robot TLC controller can follow various 3DOF trajectories accurately.

It is noted that using the time-varying PD-eigenstructure assignment, the closed-loop tracking error stabilizing controller can be synthesized with time-varying gain (bandwidth), which can be used for real-time trade off between tracking performance and robustness, control energy consumption, and to adapt to unknown environmental conditions, such as the tractive friction characteristics of the field of play. In addition, the time-varying bandwidth command shaping filter, albeit effective, is somewhat ad hoc and empirical as a substitute for optimal path planning. A rigorous optimal bandwidth varying law is highly desired. A comparison between the trajectory linearization control based Kalman filter design method and other linearized and extended Kalman filter algorithms should be performed to identify the advantages and weakness of the proposed method. These are some directions for future studies.

It is remarked that the multi-loop TLC structure presented herein is potent of solving multiple problems from an implementation point of view, and has been successfully applied to many challenging control applications, such as flight control for reusable launch vehicles (RLV) [11,35,36], air-breathing hypersonic vehicles [37], bank-to-turn and skid-to-turn missiles [38,39] and vertical takeoff and landing (VTOL) unmanned aerial vehicles [40], as well as closed-loop control of air flow over a delta wing [41]. It is also remarked that, although in general the Certainty Equivalence (CE) principle does not hold for nonlinear systems, the effectiveness of the TLC-Kalman-Filter based sensor fusion algorithm shown in this paper serves as a successful demonstration of the fact that even under inclement theoretical operating conditions such as the coordinated 3DOF maneuvers of an omni-directional robot, CE may still be applicable, which facilitates separate design of the state feedback controller and state estimation.

Acknowledgments

The authors would like to thank Dr. David Chelberg, Dr. Maarten Uijt de Haag, Dr. Douglas Lawrence and Dr. Greg Kremer for the helpful discussions during this development. Thanks are also due to the Ohio University Robocat team members Xiaofei Wu, Qiang Zhou, Ted Smith, Matthew Gillen, Mark Tomko, Mark Goldman, Ben Snyder and Rui Huang. The authors gratefully acknowledge the many valuable and constructive critiques from the reviewers.

References

[1] F.G. Pin, S.M. Killough, A new family of omnidirectional and holonomic wheeled platforms for mobile robots, IEEE Transactions on Robotics and Automation 10 (2) (1994) 480–489.

[2] M.-J. Jung, H.-S. Kim, S. Kim, J.-H. Kim, Omni-directional mobile base OK-II, in: Proceedings of the IEEE International Conference on Robotics and Automation, 2000, pp. 3449–3454.

[3] T.K. Nagy, R. D'Andrea, P. Ganguly, Near-optimal dynamic trajectory generation and control of an omnidirectional vehicle, Robotics and Autonomous Systems 47 (1) (2004) 47–64.

[4] O. Purwin, R. D'Andrea, Trajectory generation and control for four wheeled omnidirectional vehicles, Robotics and Autonomous Systems 54 (1) (2006) 13–22.

[5] J.-K. Choi, B.K. Kim, Near minimum-time direct voltage control algorithms for wheeled mobile robots with current and voltage constraints, Robotica 19 (1) (2001) 29–39.

[6] L. Wilson, J.Y. Lew, Design and modeling of a redundant omni-directional RoboCup goalie, in: RoboCup 2001 International Symposium, Seattle, WA, August, 2001.

[7] K. Watanabe, Y. Shiraishi, S.G. Tzafestas, J. Tang, T. Fukuda, Feedback control of an omnidirectional autonomous platform for mobile service robots, Journal of Intelligent and Robotic Systems 22 (3) (1998) 315–330.

[8] K. Watanabe, Control of an omnidirectional mobile robot, in: Proceedings of 1998 Second International Conference on Knowledge-Based Intelligent Electronic Systems, 1998, pp. 51–60.

[9] Y. Liu, X. Wu, J. Zhu, J. Lew, Omni-directional mobile robot controller design by trajectory linearization, in: Proceedings of the American Control Conference, 2003, pp. 3423–3428.

[10] M.C. Mickle, R. Huang, J. Zhu, Unstable nonminimum phase nonlinear tracking by trajectory linearization control, in: Proceedings of IEEE Conference on Control Applications, 2004, pp. 812–818.

[11] J. Zhu, B.D. Banker, C.E. Hall, X-33 ascent flight controller design by trajectory linearization — A singular perturbational approach, in: Proceedings of AIAA Guidance, Navigation, and Control Conference, 2000, AIAA-2000-4159.

[12] W.E. Dixon, D.M. Dawson, E. Zergeroglu, Tracking and regulation control of a mobile robot system with kinematic disturbances: A variable structure-like approach, Journal of Dynamic Systems, Measurement, and Control 122 (4) (2000) 616–623.

[13] R.L. Williams II, B.E. Carter, P. Gallina, G. Rosati, Dynamic model with slip for wheeled omni-directional robots, IEEE Transactions on Robotics and Automation 18 (3) (2002) 285–293.

[14] J. Wu, Dynamic path planning of an omni-directional robot in a dynamic environment, Ph.D. Dissertation, Ohio University, Athens, OH, 2005.

[15] M. West, H. Asada, Design of a holonomic omnidirectional vehicle, in: Proceedings of the 1992 IEEE International Conference on Robotics and Automation, 1992, pp. 97–103.

[16] I.E. Paromtchik, U. Rembold, A practical approach to motion generation and control for an omnidirectional mobile robot, in: Proceedings of 1994 IEEE International Conference on Robotics and Automation, 1994, pp. 191–200.

[17] T.K. Nagy, P. Ganguly, R. D'Andrea, Real-time trajectory generation for omni-directional vehicle, in: Proceedings of the American Control Conference, 2002, pp. 286–291.

[18] D.-H. Kim, J.-H. Kim, A real-time limit-cycle navigation method for fast mobile robots and its application to robot soccer, Robotics and Autonomous Systems 42 (1) (2003) 17–30.

[19] H.A. Samani, A. Abdollahi, H. Ostadi, S.Z. Rad, Design and development of a comprehensive omni directional soccer robot, International Journal of Advanced Robotic Systems 1 (3) (2004) 191–200.

[20] R. Huang, M.C. Mickle, J.J. Zhu, Nonlinear time-varying observer design using trajectory linearization, in: Proceedings of the American Control Conference, 2003, pp. 4772–4778.

[21] R.G. Brown, P.Y.C. Hwang, Introduction to Random Signals and Applied Kalman Filtering: With MATLAB Exercises and Solutions, 3rd ed., Wiley, 1997.

[22] T. Kailath, A.H. Saye, B. Hassibi, Linear Estimation, Prentice Hall, Upper Saddle River, NJ, 2000.

[23] J.L. Crassidis, J.L. Junkins, Optimal Estimation of Dynamic Systems, CRC Press, Boca Raton, 2004.

[24] D.L. Hall, S.A.H. McMullen, Mathematical Techniques in MultiSensor Data Fusion, 2nd ed., Artech House, 2004.

[25] L. Ojeda, D. Cruz, G. Reina, J. Borenstein, Current-based slippage detection and odometry correction for mobile robots and planetary rovers, IEEE Transactions on Robotics 22 (2) (2006) 366–378.

[26] G. Antonelli, S. Chiaverini, G. Fusco, A calibration method for odometry of mobile robots based on the least-squares technique: Theory and experimental validation, IEEE Transactions on Robotics 21 (5) (2005) 994–1004.

[27] G. Oriolo, A.D. Luca, M. Vendittelli, M.: WMR control via dynamic feedback linearization: Design, implementation and experimental validation, IEEE Transactions on Control Systems Technology 10 (6) (2002) 835–852.

[28] E. Maalouf, M. Saad, H. Saliah, A higher level path tracking controller for a four-wheel differentially steered mobile robot, Robotics and Autonomous Systems 54 (1) (2006) 23–33.

[29] P. Goel, S.I. Roumeliotis, G.S. Sukhatme, Robust localization using relative and absolute position estimates, in: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 1999, pp. 1134–1140.

[30] A. Isidori, Nonlinear Control Systems, 3rd ed., Springer, 1995.

[31] A. Isidori, C.I. Byrness, Output regulation of nonlinear systems, IEEE Transactions on Automatic Control 35 (2) (1990) 131–140.

[32] C.I. Byrnes, A. Isidori, Output regulation for nonlinear systems: An overview, International Journal of Robust and Nonlinear Control 10 (5) (2000) 323–337.

[33] Y. Liu, R.L. Williams, J.J. Zhu, Integrated control and navigation for omni-directional mobile robot based on trajectory linearization, in: Proceedings of the American Control Conference, 2007, pp. 2153–2158.

[34] Y. Liu, J.J. Zhu, Singular perturbation analysis for trajectory linearization control, in: American Control Conference, 2007, pp. 3047–3052.

[35] J. Zhu, A. Scott Hodel, Kerry Funston, Charles E. Hall, X-33 entry flight controller design by trajectory linearization — a singular perturbational approach, in: AAS-01-012, American Astronautical Society Guidance and Control Conference, Breckenridge, CO, Jan. 2001, in: Guidance and Control 2001, Advances in the Astronautical Sciences, vol. 107, American Astronautical Society [AAS GNC 2001], pp. 151–170.

[36] T. Bevacqua, E.A. Best, A. Huizenga, D. Cooper, J.J. Zhu, Improved trajectory linearization flight controller for reusable launch vehicles, in: Proc. 42nd AIAA Aerospace Science Meeting, Reno, NV, Jan. 2004, AIAA 2004-0875.

[37] Tony A. Adami, J. Jim Zhu, Michael A. Bolender, David B. Doman, Michael Oppenheimer, Flight control of hypersonic scramjet vehicles using a differential algebraic approach, AIAA-2006-6559, in: Proceedings, AIAA Guidance, Navigation and Control Conference, Keystone, CO, August 2006.

[38] M.C. Mickle, J.J. Zhu, Bank-to-turn roll-yaw-pitch autopilot design using dynamic nonlinear inversion and PD-eigenvalue assignment, in: Proc. American Control Conference, Chicago, IL, 2000, pp. 1359–1364.

[39] M.C. Mickle, J. Zhu, Skid to turn control of the APKWS missile using trajectory linearization technique, in: Proceedings, 2001 American Control Conference, Crystal City, VA, June 2001, pp. 3346–3351.

[40] Rui Huang, Yong Liu, Joseph Althaus, Nathan Daniher, Christopher Engel, J. Jim Zhu, Guidance, navigation and control system design for a tri-propeller VTOL UAV, AIAA-2007-6459, in: Proceedings, AIAA Guidance, Navigation and Control Conference, Hilton Head, SC, August 2007.

[41] Y. Liu, M. Wu, J.J. Zhu, D.A. Lawrence, E.J. Gutmark, J.H. Myatt, C.A. May, Reactive flow control of delta wing vortex, in: Proc. AIAA Guidance, Navigation, and Control Conference and Exhibit, August 2006, Keystone, CO, AIAA-2006-6189.

Dr. Yong Liu is currently working at VIASYS Health Care Inc., as a control systems engineer. He received his Ph.D. degree from the School of Electrical Engineering and Computer Science at Ohio University in June, 2007. He received his B.E. and M.E. from the School of Electrical Engineering and Automation at Tianjin University, Tianjin, China in 1997 and 2000 respectively. His research fields are robotics, nonlinear adaptive control, flight control, closed-loop active flow control and embedded control system. He has published 14 conference papers and 6 journal papers in his research fields. He worked at Intelligent Automation Inc. in 2005 on NASA and Airforce SBIR research projects.

J. Jim Zhu received the B.S. Degree in Industrial Automation from the Beijing Polytechnic University in 1976. From the University of Alabama in Huntsville he received the M.S. Degree in Electrical Engineering in 1984, the M.A. Degree in Mathematics in 1986, and the Ph.D. Degree in Electrical Engineering with an honor of Highest Academic Achievement in 1989. Zhu joined Louisiana State University (LSU) in July, 1990, where he held the positions of Assistant Professor and Associate Professor through August 2000. Since then he has been a Professor in the School of Electrical Engineering and Computer Science at Ohio University.

Dr. Zhu's main research area and contribution is in time-varying linear systems theory and nonlinear control system design. His current research interests and activities include linear and nonlinear dynamical and control systems theory, with applications in aerospace vehicles, robotics and smart materials. To date he has published more than 100 papers in mathematical and engineering journals and conference proceedings. Since 1990 he has served as principal investigator (PI) or Co-PI on sponsored research projects totaling more than $4 million, including nearly $3M projects on aerospace vehicle guidance and flight control systems and closed-loop aerodynamic flow control. Dr. Zhu received the NSF Research Initiation Award in 1991. He was an AFOSR Summer Faculty Associate in 1995, a NASA Summer Research Fellow in 1999, and a National Research Council Summer Research Fellow in 2002. Dr. Zhu is a Senior Member of IEEE and a Senior Member of AIAA. He was a technical associate editor of the Control Systems Magazine from 1996 to 1997, an Associate Editor of the IEEE Control System Society (CSS) Conference Editorial Board (CEB) from 1994 to 1997, and the CEB Chair/Editor from 1998 to 2001. He was the 34th Conference on Decision and Control (CDC) Program Committee Vice-Chairman, the 43rd CDC Publication Chair and the General Chair for the 28th IEEE Southeastern Symposium on System Theory (SSST). He was an Elected Member of the IEEE CSS Board of Governors from 2001 to 2003. He is currently an Associate Editor for International Journal and Control, Automation and Systems.

Dr. Robert L. Williams II is a professor of mechanical engineering at Ohio University, focusing on robotics and haptics research and education. Previously he worked for 5 years at NASA Langley Research Center as space roboticist. He earned the Ph.D. in mechanical engineering from Virginia Tech. Dr. Williams has published 26 journal articles and over 50 conference papers in dynamics, control, robotics, and haptics. He is a reviewer for many journals including ASME and IEEE journals. He has been PI on externally-funded projects totaling over $2M. During his 10 years at Ohio University, he has worked two summers each at NASA Kennedy Space Center and Wright-Patterson AFB. During his 2002–2003 sabbatical he worked for the NIST Intelligent Systems Division in Gaithersburg MD. Dr. Williams' research interests include parallel robots, cable-suspended robots, mobile robots, and haptics for education and training. Dr. Williams is the ASME advisor for the Ohio University student chapter, he has taught summer robotics programs to middle school students, and he is the MathCounts coach for Athens (OH) Middle School. He serves as lead guitarist in a praise band for a Methodist church.

Dr. Jianhua Wu received B.Sc degree and M.Sc degree in Mechanical Engineering in 1983 and 1986 from Shanghai Jiao-tong University, Shanghai, China, M.Eng degree in Mechanical Engineering in 1993 from Waseda University, Tokyo, Japan and Ph.D. degree in Integrated Engineering in 2005 from Ohio University, Athens, Ohio. He is currently working at Ohio Development Center, SEWS, Inc. He has active interest in dynamics, control system design and implementation and mobile robot.