the poor man's control error is monotonically decreasing, i.e., $\|\boldsymbol{\varepsilon}_{t+1}(\widehat{\mathbf{x}}_1)\| \leq \|\boldsymbol{\varepsilon}_t(\widehat{\mathbf{x}}_1)\|$ since $0 \leq \beta_1 < 1$. Similarly, the monotone convergence domain of the FO is defined by

$$\|\boldsymbol{\varepsilon}_{t+1}(\widehat{\mathbf{x}}_F) - \boldsymbol{\varepsilon}_{t+1}(\mathbf{x})\| \leq \beta_F \|\boldsymbol{\varepsilon}_t(\widehat{\mathbf{x}}_F) - \boldsymbol{\varepsilon}_t(\mathbf{x})\|, \tag{34}$$

so that the respective control error is monotonically decreasing for all $\boldsymbol{\varepsilon}_t(\widehat{\mathbf{x}}_F)$ in the domain

$$\mathcal{D}_F = \|\boldsymbol{\varepsilon}_t(\widehat{\mathbf{x}}_F) - \boldsymbol{\varepsilon}_t(\mathbf{x})\| < \delta_F. \tag{35}$$

Although it is not straightforward to rigorously determine the "size" of the convergence domains $\mathcal{D}_1$ and $\mathcal{D}_F$ separately, it is less difficult to compare them relatively with each other.

Given the initial guess $\widehat{\mathbf{x}}_0$, if the iterative estimation procedure converges toward the solution $\mathbf{x}$, then

$$\begin{aligned}\|\widehat{\mathbf{x}}_{k+1} - \mathbf{x}\| \leq \beta_e \|\widehat{\mathbf{x}}_k - \mathbf{x}\| \leq \beta_e^2 \|\widehat{\mathbf{x}}_{k-1} - \mathbf{x}\| \leq \cdots \\ \leq \beta_e^k \|\widehat{\mathbf{x}}_1 - \mathbf{x}\| \leq \beta_e^{k+1} \|\widehat{\mathbf{x}}_0 - \mathbf{x}\|\end{aligned} \tag{36}$$

and

$$\widehat{\mathbf{x}}_k \to \widehat{\mathbf{x}}_F \quad \text{as} \quad k \to \infty. \tag{37}$$

Provided the unified servoing framework, applying (37) and (36) in the control error for an image at $t+1$, and then using (32), yield

$$\|\boldsymbol{\varepsilon}_{t+1}(\widehat{\mathbf{x}}_F) - \boldsymbol{\varepsilon}_{t+1}(\mathbf{x})\| \leq \|\boldsymbol{\varepsilon}_{t+1}(\widehat{\mathbf{x}}_1) - \boldsymbol{\varepsilon}_{t+1}(\mathbf{x})\| \tag{38}$$

$$\leq \beta_1 \|\boldsymbol{\varepsilon}_t(\widehat{\mathbf{x}}_1) - \boldsymbol{\varepsilon}_t(\mathbf{x})\|. \tag{39}$$

This result shows that if the PM converges, then the FO also converges. As a remark, such a result is in fact very conservative as only a subset of the parameters is updated in the PM.

References

[1] F. Chaumette and S. Hutchinson, "Visual servo control part I: Basic approaches," *IEEE Robot. Autom. Mag.*, vol. 13, no. 4, pp. 82–90, Dec. 2006.

[2] L. Thaler and M. A. Goodale, "Beyond distance and direction: The brain represents target locations non-metrically," *J. Vis.*, vol. 10, no. 3, pp. 1–27, 2010.

[3] P. A. Beardsley, I. D. Reid, A. Zisserman, and D. W. Murray, "Active visual navigation using non-metric structure," in *Proc. IEEE Int. Conf. Comput. Vis.*, 1995, pp. 58–64.

[4] V. Kallem, M. Dewan, J. Swensen, G. Hager, and N. Cowan, "Kernel-based visual servoing," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, San Diego, CA, USA, 2007, pp. 1975–1980.

[5] M. Vargas and E. Malis, "Visual servoing based on an analytical homography decomposition," in *Proc. Joint IEEE Conf. Decision Control Eur. Control Conf.*, 2005, pp. 5379–5384.

[6] S. Benhimane and E. Malis, "Homography-based 2D visual servoing," in *Proc. IEEE Int. Conf. Robot. Autom.*, Orlando, FL, USA, 2006, pp. 2397–2402.

[7] G. Silveira and E. Malis, "Direct Visual Servoing: Vision-based estimation and control using only nonmetric information," *IEEE Trans. Robot.*, vol. 28, no. 4, pp. 974–980, Aug. 2012.

[8] M. Irani and P. Anandan, "All about direct methods," in *Proc. Workshop Vis. Algorithms, Theory Pract.*, 1999.

[9] G. Silveira and E. Malis, "Visual servoing from robust direct color image registration," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, St. Louis, MO, USA, 2009, pp. 5450–5455.

[10] G. Silveira, "Photogeometric direct visual tracking for central omnidirectional cameras," *J. Math. Imag. Vis.*, vol. 48, no. 1, pp. 72–82, 2014.

[11] R. Horst and P. M. Pardalos, Eds., *Handbook of Global Optimization*. Norwell, MA, USA: Kluwer, 1995.

[12] G. Silveira, E. Malis, and P. Rives, "The efficient E-3D visual servoing," *Int. J. Optomechatron.*, vol. 2, no. 3, pp. 166–184, 2008.

[13] C. Collewet, E. Marchand, and F. Chaumette, "Visual servoing set free from image processing," in *Proc. IEEE Int. Conf. Robot. Autom.*, Pasadena, CA, USA, 2008, pp. 1050–4729.

[14] S. Han, A. Censi, A. D. Straw, and R. M. Murray, "A bio-plausible design for visual pose stabilization," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2010, pp. 5679–5686.

[15] A. Dame and E. Marchant, "Mutual information-based visual servoing," *IEEE Trans. Robot.*, vol. 27, no. 5, pp. 958–969, Oct. 2011.

[16] G. Silveira and E. Malis, "Direct visual servoing with respect to rigid objects," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, San Diego, CA, USA, 2007, pp. 1963–1968.

[17] S. K. Nayar, S. A. Nene, and H. Murase, "Subspace methods for robot vision," *IEEE Trans. Robot.*, vol. 12, no. 5, pp. 750–758, Oct. 1996.

[18] K. Deguchi, "A direct interpretation of dynamic images with camera and object motions for vision guided robot control," *Int. J. Comput. Vis.*, vol. 37, no. 1, pp. 7–20, 2000.

[19] G. Silveira and E. Malis, "Unified direct visual tracking of rigid and deformable surfaces under generic illumination changes in grayscale and color images," *Int. J. Comput. Vis.*, vol. 89, no. 1, pp. 84–105, 2010.

[20] F. W. Warner, *Foundations of Differential Manifolds and Lie Groups*. New York, NY, USA: Springer-Verlag, 1987.

[21] D. G. Luenberger, *Linear and Nonlinear Programming*. Reading, MA, USA: Addison-Wesley, 1984.

Visual Servoing Trajectory Tracking of Nonholonomic Mobile Robots Without Direct Position Measurement

Kai Wang, *Member, IEEE*, Yunhui Liu, *Fellow, IEEE*, and Luyang Li, *Student Member, IEEE*

***Abstract*—Localization is one of the most difficult and costly problems in mobile robotics. To avoid this problem, this paper presents a new controller for the trajectory tracking of nonholonomic mobile robots using visual feedback without direct position measurement. This controller works on the basis of a novel adaptive algorithm for estimating the global position of the mobile robot online using natural visual features measured by a vision system and its orientation and velocity measured by odometry and Attitude and Heading Reference System (IMU&Compass) sensors. The nonholonomic motion constraint of mobile robots is fully taken into account, compared with most of the existing visual servo controllers for mobile robots. The Lyapunov theory is used to prove that the proposed adaptive visual servo controller gives rise to asymptotic tracking of a desired trajectory and convergence of the position estimation to the actual position. A graphical processing unit is adopted to implement the proposed adaptive controller in parallel to achieve real-time detection and tracking of visual features. Experiments on a mobile robot are conducted to validate the effectiveness and robust performance of the proposed controller.**

***Index Terms*—Adaptive control, nonholonomic mobile robots, trajectory tracking, visual servoing.**

Manuscript received November 13, 2013; revised March 20, 2013; accepted April 11, 2014. Date of publication May 7, 2014; date of current version August 4, 2014. This paper was recommended for publication by Associate Editor P. R. Giordano and Editor G. Oriolo upon evaluation of the reviewers' comments. This work was supported in part by Hong Kong RGC under Grant 415110 and Grant 414912. This paper, with detailed proofs and experiment data, is the extended version of a previous paper presented at the 2013 IEEE International Conference on Robotics and Automation [19].

K. Wang and L. Li are with the Department of Mechanical and Automation Engineering, Chinese University of Hong Kong, Shatin, Hong Kong (e-mail: kwang@mae.cuhk.edu.hk; lyli@mae.cuhk.edu.hk).

Y. Liu is with the Department of Mechanical and Automation Engineering, Chinese University of Hong Kong, Shatin, Hong Kong, and also with the State Key Laboratory of Robotics Technology and Systems, Harbin Institute of Technology, Harbin 150001, China (e-mail: yhliu@mae.cuhk.edu.hk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TRO.2014.2317891

I. Introduction

The differential drive or steering drive mobile robots are subject to nonholonomic constraints, and their trajectory tracking control is challenging because there exists no smooth time-invariant feedback controller [1], [2]. Various model-based control methods, including discontinuous controllers [3], time-varying controllers [4], [5], and hybrid controllers [6], have been developed. All the existing model-based trajectory tracking controllers work under the assumption that the position and orientation of the robot can be accurately measured. However, despite tremendous efforts, localization of mobile robots still remains as one of the most difficult problems in robotics and suffers from limit localization range and/or limit accuracy [7]–[9]. The objective of our work is to eliminate the requirement for the global position measurement in motion control of mobile robots and to design a controller that does not require direct position measurement for the trajectory tracking of nonholonomic mobile robots.

Visual servoing presents an effective framework to control mobile robots without localizing their positions [10]–[14]. Visual servoing directly feeds back the information about image features to the controller or employs them to estimate the position of the robot. Existing works on visual servoing of mobile robots can be classified into regulation [10], path following [11], [12], and tracking moving objects [13]. Mariottini *et al*. brought forward an image-based regulation controller to align the robot with the goal, by zeroing the epipoles [10]. Coulaud *et al*. [11] proposed an image-based path-following controller without explicitly utilizing the position of the robot. Cherubini *et al*. [12] developed a position-based one assuming that the position of the robot could be measured. Tsai *et al*. designed a robust visual tracking controller to track a dynamic moving target, based on a dual-Jacobian visual interaction model [13]. Chen *et al*. used a prerecorded image sequence of three target points to define a desired trajectory, and designed a homography-based kinematic controller to track the image-based trajectory [14]. Path following consists in following a geometric path marked on the ground, not a time-varying trajectory. Visual servoing regulation and tracking of moving objects are different from the problem addressed in this paper. Image-based trajectory tracking has the limitation that the image sequences should be stored beforehand and, hence, is not suitable for universal applications. To the best of our knowledge, no position-based visual servoing controller has been developed for the trajectory tracking of mobile robots with nonholonomic constraints.

In this paper, we present a position-based visual servo controller for the trajectory tracking of nonholonomic mobile robots. This controller does not require direct position measurement of the mobile robot, but employs as feedback an estimated position, which is generated by an adaptive estimator from visual information, orientation, and linear velocity of the robot. Compared with the position, the orientation and linear velocity of a mobile robot can be measured more easily and accurately by Attitude and Heading Reference System (AHRS, IMU&compass) sensors and encoders, etc. The controller design is based on the works of Kanayama *et al*. [4] and Jiang and Nijmeijer [5]. The difference lies in the design of a new nominal reference using the estimated position, not the real position used in [5]. The algorithm for estimating the position is similar to the model-based adaptive algorithm, but it estimates the position by minimizing online a nominal image error newly introduced on the basis of perspective projection geometry [15]. This nominal image error can be linearly parameterized by the estimation errors. To improve the robustness of the controller, the speeded up robust features (SURF) [16] are used. It should be noted that the position estimator is embedded into the control loop in our work, unlike in many other works [17], [18], where the position estimation is carried out in an independent loop. In this paper, it is proved by the Lyapunov theory that the proposed controller with the adaptive position estimator leads to asymptotic trajectory tracking and the convergence of the estimated position to the real one when the desired linear velocity does not converge to zero. To ensure the real-time performance, graphical processing unit (GPU) is employed to accelerate the online position estimation by parallelly processing multiple feature points. The experiments conducted on a mobile platform have verified the satisfactory position estimation and trajectory tracking performances (effectiveness and robustness) of the proposed controller.

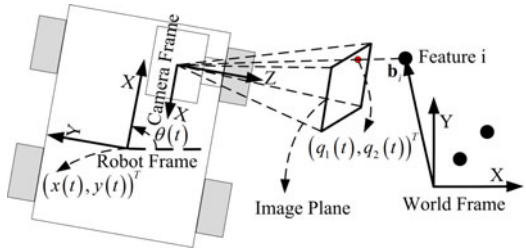

Fig. 1. Coordinate frames of the mobile robot.

The contribution can be summarized as follows: First, we propose the first position-based visual servo controller for the trajectory tracking of nonholonomic mobile robots to the best of our knowledge. Second, we develop an adaptive algorithm for estimating the position of a robot using natural image features based on a new nominal image error that can be linearly parameterized by the unknown positions of the robot and features. Third, the asymptotic stability of the controller is proved by the Lyapunov theory and the performance is verified by experiments.

This paper is organized as follows. Section II reviews the kinematics of the nonholonomic robots and the projection geometry of the perspective camera. The controller design is presented in Section III. Section IV shows the experimental results. Finally, Section V concludes this study.

II. Preliminaries

A. Nonholonomic Mobile Robots

A wheeled nonholonomic robot is assumed to take a planar motion, and its pose is represented by a 3×1 vector $\mathbf{p}(t) = (x(t), y(t), \theta(t))^T$ with respect to a world frame, as shown in Fig. 1. Let $(x(t), y(t))^T$ and $\theta(t)$ denote the position and orientation of the robot, respectively. The kinematics of the mobile robot are given by the following equation:

$$\dot{\mathbf{p}}(t) = \begin{pmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{\theta}(t) \end{pmatrix} = \begin{pmatrix} \cos\theta(t) & 0 \\ \sin\theta(t) & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \nu(t) \\ \omega(t) \end{pmatrix} \tag{1}$$

where $\nu(t)$ and $\omega(t)$ represent linear and angular velocities of the robot, respectively, which are used as the control inputs. The major issue in the mobile robot control is designing $\nu(t)$ and $\omega(t)$ to achieve the control objective. It is well known that (1) imposes nonholonomic constraints on robot motion, which make the controller design difficult.

B. Problem Statement

The robot moves in a natural environment where no global positioning system or other global localization system is available for measuring its position. The robot is equipped with odometry and AHRS sensors to measure its velocity and orientation.

Assumption 1: Linear and angular velocities of a wheeled mobile robot can be measured (then, controlled by low-level controllers)

accurately in the case of nonslipping rotation between the wheels and the ground, if high-resolution encoders are installed at the wheels.

A desired trajectory of the robot is denoted by the desired pose and desired velocity, respectively, as follows:

$$\begin{cases} \mathbf{p}_d(t) = (x_d(t), y_d(t), \theta_d(t))^T \\ \dot{\mathbf{p}}_d(t) = \left(\dot{x}_d(t), \dot{y}_d(t), \dot{\theta}_d(t)\right)^T. \end{cases} \tag{2}$$

The trajectory defined in (2) is differentiable and bounded. The desired speed and angular velocity of the robot can be calculated by

$$\begin{cases} \nu_d(t) = \sqrt{\dot{x}_d^2(t) + \dot{y}_d^2(t)} \\ \omega_d(t) = \dot{\theta}_d(t). \end{cases} \tag{3}$$

Problem 1: Given a desired trajectory, which is denoted by (2), of a mobile robot subject to the nonholonomic constraint (1), design proper $(\nu(t), \omega(t))^T$ by using feedback from the camera, odometry, and AHRS sensors so that the robot can track the desired trajectory.

C. Perspective Projection Geometry

The robot is equipped with a perspective camera, and the visual information is fused with the velocity and orientation of the robot, to estimate the global position. For simplicity and clear presentation, we derive the projection geometry and controller/estimator using a single feature point. The extension to multiple features is straightforward.

Denote the 3-D position of a fixed feature point with respect to the world frame by $\mathbf{b} = (b_x, b_y, b_z)^T$, which is an unknown constant 3×1 vector in a static environment. The feature point is projected onto the image plane of the camera as a point with image coordinates $\mathbf{q}(t) = (q_1(t), q_2(t))^T$, which change with the motion of the robot. Based on the perspective projection model, we have

$$\begin{pmatrix} \mathbf{q}(t) \\ \mathbf{1} \end{pmatrix} = \frac{1}{z(t)} \mathbf{M} \underbrace{\begin{pmatrix} \mathbf{R}^T(\theta(t)) & 0_{3\times 1} \\ 0_{1\times 3} & \mathbf{1} \end{pmatrix}}_{\mathbf{T}(\theta(t))} \begin{pmatrix} b_x - x(t) \\ b_y - y(t) \\ b_z - 0 \\ 1 \end{pmatrix} \tag{4}$$

where $\mathbf{M}$ is the 3×4 constant perspective projection matrix determined by the intrinsic and extrinsic parameters of the camera. Let

$$\mathbf{M} = \left(\mathbf{m}_1^T, \mathbf{m}_2^T, \mathbf{m}_3^T\right)^T. \tag{5}$$

Moreover, $\mathbf{R}(\theta(t))$ is the 3×3 rotation matrix of the robot with respect to the world frame, and $\mathbf{T}(\theta(t))$ is the 4×4 homogeneous transformation matrix. $z(t)$ is the depth of the feature point with respect to the camera frame, which is given by

$$z(t) = \mathbf{m}_3^T \mathbf{T}(\theta(t)) \begin{pmatrix} b_x - x(t) \\ b_y - y(t) \\ b_z - 0 \\ 1 \end{pmatrix}. \tag{6}$$

Property 1: The numerator in (4) can be represented in the following linear form:

$$\mathbf{M}\mathbf{T}(\theta(t)) \begin{pmatrix} b_x - x(t) \\ b_y - y(t) \\ b_z - 0 \\ 1 \end{pmatrix} = \mathbf{\Phi}(\theta(t)) \begin{pmatrix} x(t) \\ y(t) \\ \mathbf{b} \end{pmatrix} + \boldsymbol{\delta} \tag{7}$$

where the 3×5 matrix $\mathbf{\Phi}(\theta(t))$ and the constant vector $\boldsymbol{\delta}$ (see details at the bottom of the next page) do not depend on the unknown vector $\mathbf{b}$ nor on the position of the robot.

TABLE I
SYMBOLS USED IN SECTION III

Symbol	*Definition*
$\Delta\mathbf{p}(t)$	the pose error
$\mathbf{e}(t) = (e_x(t), e_y(t), e_\theta(t))^T$	the transformed pose error
$\Delta\hat{\mathbf{p}}(t)$	estimation of the pose error
$\hat{\mathbf{e}}(t) = (\hat{e}_x(t), \hat{e}_y(t), e_\theta(t))^T$	estimation of the transformed pose error
$(\hat{x}(t), \hat{y}(t))^T$	estimation of the robot position
$(\tilde{x}(t), \tilde{y}(t))^T$	$(x(t) - \hat{x}(t), y(t) - \hat{y}(t))^T$
$(\tilde{e}_x(t), \tilde{e}_y(t))^T$	$(e_x(t) - \hat{e}_x(t), e_y(t) - \hat{e}_y(t))^T$
$\tilde{\mathbf{e}}(t)$	$\mathbf{e}(t) - \hat{\mathbf{e}}(t)$
$\hat{\mathbf{b}}(t) = (\hat{b}_x(t), \hat{b}_y(t), \hat{b}_z(t))^T$	estimation of the feature position
$\tilde{\mathbf{b}}(t)$	$\mathbf{b} - \hat{\mathbf{b}}(t)$
$\mathbf{s}(t)$	the nominal image error
$\hat{z}(t)$	estimated depth of the feature
$\boldsymbol{\beta}(t)$	$(b_x - x(t), b_y - y(t), b_z - 0)^T$
$\hat{\boldsymbol{\beta}}(t)$	$(\hat{b}_x(t) - \hat{x}(t), \hat{b}_y(t) - \hat{y}(t), \hat{b}_z(t) - 0)^T$

Property 2: The depth of the feature point in (4) can be represented as the following linear form:

$$z(t) = \boldsymbol{\eta}^T(\theta(t)) \begin{pmatrix} x(t) \\ y(t) \\ \mathbf{b} \end{pmatrix} + \gamma \tag{8}$$

where the 5×1 vector $\boldsymbol{\eta}(\theta(t))$ and the constant scalar γ (see details at the bottom of the next page) do not depend on the unknown vector $\mathbf{b}$ nor on the position of the robot.

III. POSITION-BASED VISUAL SERVOING TRAJECTORY TRACKING

This section presents a new position-based visual servo controller for the trajectory tracking of a nonholonomic mobile robot. This controller does not use any measured position as feedback by adopting an estimator to estimate the position online from the visual feedback and velocity/orientation of the robot. This new controller is developed on the basis of the work of Kanayama *et al.* [4] and the controller proposed by Jiang and Nijmeijer [5]. We first review their work and then present the detailed design of our controller.

The symbols used in this section are listed in Table I. To make the following derivation concise, we introduce a 2×2 matrix as follows:

$$\mathbf{A}(\theta(t)) = \begin{pmatrix} \cos\theta(t) & \sin\theta(t) \\ -\sin\theta(t) & \cos\theta(t) \end{pmatrix}. \tag{9}$$

Property 3: $\mathbf{A}^T(\theta(t))\mathbf{A}\left(\theta(t) - \frac{\pi}{2}\right)$ is a skew-symmetric matrix, that is

$$\mathbf{A}^T(\theta(t))\mathbf{A}\left(\theta(t) - \frac{\pi}{2}\right) = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}. \tag{10}$$

Property 4: $\mathbf{A}^T(\theta(t))\mathbf{A}(\theta(t))$ is an identity matrix, that is

$$\mathbf{A}^T(\theta(t))\mathbf{A}(\theta(t)) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}. \tag{11}$$

A. Tracking Control of Nonholonomic Mobile Robots

It is assumed that the pose of the mobile robot can be obtained. The pose error of the robot is given by

$$\mathbf{\Delta p}\left(t\right)=\mathbf{p}_{d}\left(t\right)-\mathbf{p}\left(t\right)=\begin{pmatrix}\Delta x\left(t\right)\\ \Delta y\left(t\right)\\ \Delta\theta\left(t\right)\end{pmatrix}=\begin{pmatrix}x_{d}\left(t\right)-x\left(t\right)\\ y_{d}\left(t\right)-y\left(t\right)\\ \theta_{d}\left(t\right)-\theta\left(t\right)\end{pmatrix}. \tag{12}$$

In [4], Kanayama *et al*. introduced the following useful transformation of the errors:

$$\mathbf{e}\left(t\right)=\begin{pmatrix}e_{x}\left(t\right)\\ e_{y}\left(t\right)\\ e_{\theta}\left(t\right)\end{pmatrix}=\begin{pmatrix}\cos\theta\left(t\right) & \sin\theta\left(t\right) & 0\\ -\sin\theta\left(t\right) & \cos\theta\left(t\right) & 0\\ 0 & 0 & 1\end{pmatrix}\mathbf{\Delta p}\left(t\right). \tag{13}$$

Note that the coefficient matrix in (13) is of full rank; therefore, the convergence of vector $\mathbf{e}\left(t\right)$ to zero guarantees the convergence of the pose error $\mathbf{\Delta p}\left(t\right)$ to zero. By differentiating (11) and following the derivation process in [4], the following error dynamics can be obtained:

$$\dot{\mathbf{e}}\left(t\right)=\begin{pmatrix}\nu_{d}\left(t\right)\cos e_{\theta}\left(t\right)\\ \nu_{d}\left(t\right)\sin e_{\theta}\left(t\right)\\ \omega_{d}\left(t\right)\end{pmatrix}+\begin{pmatrix}-1 & e_{y}\left(t\right)\\ 0 & -e_{x}\left(t\right)\\ 0 & -1\end{pmatrix}\begin{pmatrix}\nu\left(t\right)\\ \omega\left(t\right)\end{pmatrix}. \tag{14}$$

For the error dynamics system (14), Jiang and Nijmeijer proposed the following controller in [5]:

$$\begin{pmatrix}\nu\left(t\right)\\ \omega\left(t\right)\end{pmatrix}=\begin{pmatrix}\nu_{d}\left(t\right)\cos e_{\theta}\left(t\right)+k_{x}e_{x}\left(t\right)\\ \omega_{d}\left(t\right)+k_{\theta}e_{\theta}\left(t\right)+\nu_{d}\left(t\right)e_{y}\left(t\right)\sin e_{\theta}\left(t\right)/e_{\theta}\left(t\right)\end{pmatrix}, \tag{15}$$

where k_x and k_θ are positive gains, and $\lim_{e_{\theta}\left(t\right)\to 0}\frac{\sin e_{\theta}\left(t\right)}{e_{\theta}\left(t\right)}=1$.

It has been proved [5] that the controller (15) leads to the asymptotic convergence of the position error to zero.

B. Tracking Control Using the Estimation of the Position

When there is no global localization system, the position of the robot is unavailable, and hence, the controller (15) cannot be implemented directly.

Generally speaking, the controller proposed in this paper has a similar form to (15), but employs an estimation $\left(\hat{x}\left(t\right),\hat{y}\left(t\right)\right)^{T}$ of the robot position. The corresponding estimation of the pose error of the robot is given by

$$\mathbf{\Delta\hat{p}}\left(t\right)=\begin{pmatrix}\Delta\hat{x}\left(t\right)\\ \Delta\hat{y}\left(t\right)\\ \Delta\theta\left(t\right)\end{pmatrix}=\begin{pmatrix}x_{d}\left(t\right)-\hat{x}\left(t\right)\\ y_{d}\left(t\right)-\hat{y}\left(t\right)\\ \theta_{d}\left(t\right)-\theta\left(t\right)\end{pmatrix}. \tag{16}$$

It is important to note that the orientation angle can be measured without any accumulation error by AHRS sensors. Therefore, the third component of $\mathbf{\Delta\hat{p}}\left(t\right)$ represents the true orientation error, i.e., $\Delta\hat{\theta}\left(t\right)=\Delta\theta\left(t\right)$.

The estimated and true pose errors of the robot are related by

$$\mathbf{\Delta\hat{p}}\left(t\right)=\mathbf{\Delta p}\left(t\right)+\begin{pmatrix}\tilde{x}\left(t\right)\\ \tilde{y}\left(t\right)\\ 0\end{pmatrix}. \tag{17}$$

The estimation of the transformed error is given by

$$\hat{\mathbf{e}}\left(t\right)=\begin{pmatrix}\hat{e}_{x}\left(t\right)\\ \hat{e}_{y}\left(t\right)\\ e_{\theta}\left(t\right)\end{pmatrix}=\begin{pmatrix}\cos\theta\left(t\right) & \sin\theta\left(t\right) & 0\\ -\sin\theta\left(t\right) & \cos\theta\left(t\right) & 0\\ 0 & 0 & 1\end{pmatrix}\mathbf{\Delta\hat{p}}\left(t\right). \tag{18}$$

Replacing $\mathbf{e}\left(t\right)$ in the controller (15) by $\hat{\mathbf{e}}\left(t\right)$ leads to the following control law:

$$\begin{pmatrix}\nu(t)\\ \omega(t)\end{pmatrix}=\begin{pmatrix}\nu_{d}\left(t\right)\cos e_{\theta}\left(t\right)+k_{x}\hat{e}_{x}\left(t\right)\\ \omega_{d}\left(t\right)+k_{\theta}e_{\theta}\left(t\right)+\nu_{d}\left(t\right)\hat{e}_{y}\left(t\right)\sin e_{\theta}\left(t\right)/e_{\theta}\left(t\right)\end{pmatrix}. \tag{19}$$

Substituting the controller (19) into the error dynamics (14) of the robot leads to the following closed-loop equation:

$$\begin{aligned}\dot{\mathbf{e}}\left(t\right)=&\begin{pmatrix}-k_{x}\hat{e}_{x}\left(t\right)+\hat{e}_{y}\left(t\right)\omega\left(t\right)\\ \nu_{d}\left(t\right)\sin e_{\theta}\left(t\right)-\hat{e}_{x}\left(t\right)\omega\left(t\right)\\ -k_{\theta}e_{\theta}\left(t\right)-\nu_{d}\left(t\right)\hat{e}_{y}\left(t\right)\sin e_{\theta}\left(t\right)/e_{\theta}\left(t\right)\end{pmatrix}\\ &+\begin{pmatrix}\tilde{e}_{y}\left(t\right)\\ -\tilde{e}_{x}\left(t\right)\\ 0\end{pmatrix}\omega\left(t\right).\end{aligned} \tag{20}$$

From (13), (17), and (18), we obtain

$$\begin{pmatrix}\tilde{e}_{x}\left(t\right)\\ \tilde{e}_{y}\left(t\right)\end{pmatrix}=\mathbf{A}\left(\theta\left(t\right)\right)\begin{pmatrix}\tilde{x}\left(t\right)\\ \tilde{y}\left(t\right)\end{pmatrix}. \tag{21}$$

$$\mathbf{M}=\mathbf{\Omega}\cdot\left(\mathbf{R}_{c}^{T},-\mathbf{R}_{c}^{T}\boldsymbol{\lambda}\right)=\begin{pmatrix}f_{1} & 0 & q_{10}\\ 0 & f_{2} & q_{20}\\ 0 & 0 & 1\end{pmatrix}\begin{pmatrix}-1 & 0 & 0 & \lambda_{1}\\ 0 & 0 & -1 & \lambda_{3}\\ 0 & -1 & 0 & \lambda_{2}\end{pmatrix}=\begin{pmatrix}-f_{1} & -q_{10} & 0 & f_{1}\lambda_{1}+q_{10}\lambda_{2}\\ 0 & -q_{20} & -f_{2} & f_{2}\lambda_{3}+q_{20}\lambda_{2}\\ 0 & -1 & 0 & \lambda_{2}\end{pmatrix}$$

where $\mathbf{\Omega}$ is the 3×3 intrinsic matrix of the camera, $\mathbf{R}_c$ and $\boldsymbol{\lambda}=\left(\lambda_1,\lambda_2,\lambda_3\right)^{T}$ are the 3×3 rotation matrix and 3-D position of the camera with respect to the robot frame, respectively, f_1 and f_2 are the focal lengths in terms of pixels, and $\left(q_{10},q_{20}\right)^{T}$ is the principal point.

$$\boldsymbol{\eta}\left(\theta\left(t\right)\right)=\left(-\sin\theta\left(t\right),\cos\theta\left(t\right),\sin\theta\left(t\right),-\cos\theta\left(t\right),0\right)^{T},\qquad\boldsymbol{\delta}=\left[\left(f_{1}\lambda_{1}+q_{10}\lambda_{2}\right),\left(f_{2}\lambda_{3}+q_{20}\lambda_{2}\right),\lambda_{2}\right]^{T}\qquad\gamma=\lambda_{2}$$

$$\mathbf{\Phi}\left(\theta\left(t\right)\right)=\begin{pmatrix}f_{1}\cos\theta\left(t\right)-q_{10}\sin\theta\left(t\right) & f_{1}\sin\theta\left(t\right)+q_{10}\cos\theta\left(t\right) & -f_{1}\cos\theta\left(t\right)+q_{10}\sin\theta\left(t\right) & -f_{1}\sin\theta\left(t\right)-q_{10}\cos\theta\left(t\right) & 0\\ -q_{20}\sin\theta\left(t\right) & q_{20}\cos\theta\left(t\right) & q_{20}\sin\theta\left(t\right) & -q_{20}\cos\theta\left(t\right) & -f_{2}\\ -\sin\theta\left(t\right) & \cos\theta\left(t\right) & \sin\theta\left(t\right) & -\cos\theta\left(t\right) & 0\end{pmatrix}$$

C. Online Position Estimation

This section presents the algorithm to estimate the robot position. From (21), we can directly obtain

$$\begin{pmatrix} \tilde{e}_y(t) \\ -\tilde{e}_x(t) \end{pmatrix} \omega(t) = -\mathbf{A}\left(\theta(t) - \frac{\pi}{2}\right)\omega(t)\begin{pmatrix} \tilde{x}(t) \\ \tilde{y}(t) \end{pmatrix}. \tag{22}$$

With (20)–(22) and Property 3, we can obtain (23), shown below, by taking the inner product of both sides of $\mathbf{e}^T(t) = \hat{\mathbf{e}}^T(t) + \tilde{\mathbf{e}}^T(t)$:

$$\mathbf{e}^T(t)\dot{\mathbf{e}}(t) = \hat{\mathbf{e}}^T(t)\dot{\mathbf{e}}(t) + \mathbf{h}^T(t)\mathbf{A}(\theta(t))\begin{pmatrix} \tilde{x}(t) \\ \tilde{y}(t) \end{pmatrix} \tag{23}$$

where $\mathbf{h}(t) = \begin{pmatrix} -k_x\hat{e}_x(t) + \hat{e}_y(t)\omega(t) \\ \nu_d(t)\sin e_\theta(t) - \hat{e}_x(t)\omega(t) \end{pmatrix}$. This equation plays an important role in the design of the position estimator.

Visual feature points (SURF) in the environment are extracted and tracked to assist the online estimation of the robot's position. Positions of the robot and features are estimated together in our method because they can promote the performance of the estimation of each other, in a SLAM-like style. Using $\hat{\mathbf{b}}(t)$ and (6), the estimated depth $\hat{z}(t)$ of the feature is

$$\hat{z}(t) = \mathbf{m}_3^T\mathbf{T}(\theta(t))\begin{pmatrix} \hat{b}_x(t) - \hat{x}(t) \\ \hat{b}_y(t) - \hat{y}(t) \\ \hat{b}_z(t) - 0\ cr1 \end{pmatrix}. \tag{24}$$

Introduce the following nominal image error as [19]

$$\mathbf{s}(t) = \hat{z}(t)\begin{pmatrix} q_1(t) \\ q_2(t) \\ 1 \end{pmatrix} - \mathbf{MT}(\theta(t))\begin{pmatrix} \hat{b}_x(t) - \hat{x}(t) \\ \hat{b}_y(t) - \hat{y}(t) \\ \hat{b}_z(t) - 0 \\ 1 \end{pmatrix}. \tag{25}$$

From Properties 1 and 2, the vector $\mathbf{s}(t)$ can be represented as a linear form of the estimated positions

$$\mathbf{s}(t) = \mathbf{W}(\theta(t), \mathbf{q}(t))\left(\hat{x}(t), \hat{y}(t), \hat{\mathbf{b}}(t)\right)^T + \boldsymbol{\phi}(\mathbf{q}(t)) \tag{26}$$

where the 3×5 matrix $\mathbf{W}(\theta(t), bfq(t))$ and the 3×1 vector $\boldsymbol{\phi}(\mathbf{q}(t))$ are shown at the bottom of the page.

From (4), for actual positions of the feature and the robot, the right-hand side of (30) is equal to zero, namely

$$\mathbf{W}(\theta(t), \mathbf{q}(t))(x(t), y(t), \mathbf{b}(t))^T + \boldsymbol{\phi}(\mathbf{q}(t)) = 0. \tag{27}$$

From (26) and (27), clearly

$$\mathbf{s}(t) = -\mathbf{W}(\theta(t), \mathbf{q}(t))\left(\tilde{x}(t), \tilde{y}(t), \tilde{\mathbf{b}}(t)\right)^T. \tag{28}$$

Based on (23) and (26), we propose the following algorithm to estimate the positions of the robot and the feature point online:

$$\begin{pmatrix} \dot{\hat{x}}(t) \\ \dot{\hat{y}}(t) \\ \dot{\hat{\mathbf{b}}}(t) \end{pmatrix} = \begin{pmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \mathbf{0}_{3\times 1} \end{pmatrix} - \boldsymbol{\Gamma}\left\{\mathbf{U}(t) + \mathbf{W}^T(\theta(t), \mathbf{q}(t))\mathbf{K}\mathbf{s}(t)\right\} \tag{29}$$

where

$$\mathbf{U}(t) = \begin{pmatrix} \mathbf{A}^T\left(\theta(t) - \frac{\pi}{2}\right)\omega(t)\begin{pmatrix} \hat{e}_x(t) \\ \hat{e}_y(t) \end{pmatrix} - \mathbf{A}^T(\theta(t))\mathbf{h}(t) \\ \mathbf{0}_{3\times 1} \end{pmatrix}_{5\times 1}$$

and $\boldsymbol{\Gamma}_{5\times 5}$ and $\mathbf{K}_{3\times 3}$ are positive-definite and diagonal gain matrices, respectively. $(\dot{x}(t), \dot{y}(t))^T$ is the global velocity of the robot measured by the on-board sensors.

D. Stability Analysis

Theorem 1: The controller (19) with the online estimator (29) results in

$$\lim_{t\to\infty}\hat{e}_x(t) = 0, \quad \lim_{t\to\infty}e_\theta(t) = 0, \quad \text{and} \quad \lim_{t\to\infty}\mathbf{s}(t) = \mathbf{0} \tag{30}$$

if $\lim_{t\to\infty}\nu_d(t) \neq 0$

$$\lim_{t\to\infty}\hat{e}_y(t) = 0. \tag{31}$$

Proof: Consider the following quadratic function:

$$V(t) = \frac{1}{2}\mathbf{e}^T(t)\mathbf{e}(t) + \frac{1}{2}\left(\tilde{x}(t), \tilde{y}(t), \tilde{\mathbf{b}}^T(t)\right)\boldsymbol{\Gamma}^{-1}\begin{pmatrix} \tilde{x}(t) \\ \tilde{y}(t) \\ \tilde{\mathbf{b}}(t) \end{pmatrix}. \tag{32}$$

By multiplying both sides of the closed-loop equation (20) by $\hat{\mathbf{e}}^T(t)$, and considering (23), we have

$$\begin{aligned} \mathbf{e}^T(t)\dot{\mathbf{e}}(t) &= -k_x\hat{e}_x^2(t) - k_\theta e_\theta^2(t) \\ &\quad - \begin{pmatrix} \hat{e}_x(t) \\ \hat{e}_y(t) \end{pmatrix}^T\mathbf{A}\left(\theta(t) - \frac{\pi}{2}\right)\omega(t)\begin{pmatrix} \tilde{x}(t) \\ \tilde{y}(t) \end{pmatrix} \\ &\quad + \mathbf{h}^T(t)\mathbf{A}(\theta(t))\begin{pmatrix} \tilde{x}(t) \\ \tilde{y}(t) \end{pmatrix}. \end{aligned} \tag{33}$$

Differentiating the second term of $V(t)$ in (32) and noting that $\dot{\mathbf{b}}(t) = \mathbf{0}_{3\times 1}$ for the static feature point, we can obtain the following equation with (28) and (29):

$$\begin{aligned} &\left(\tilde{x}(t), \tilde{y}(t), \tilde{\mathbf{b}}^T(t)\right)\boldsymbol{\Gamma}^{-1}\begin{pmatrix} \dot{\tilde{x}}(t) \\ \dot{\tilde{y}}(t) \\ \dot{\tilde{\mathbf{b}}}(t) \end{pmatrix} = -s^T(t)\mathbf{K}\mathbf{s}(t) \\ &+ (\tilde{x}(t), \tilde{y}(t))\left\{\mathbf{A}^T\left(\theta(t) - \tfrac{\pi}{2}\right)\omega(t)\begin{pmatrix} \hat{e}_x(t) \\ \hat{e}_y(t) \end{pmatrix} - \mathbf{A}^T(\theta(t))\mathbf{h}(t)\right\}. \end{aligned} \tag{34}$$

$$\mathbf{W}(\theta(t), \mathbf{q}(t)) = (q(t), 1)^T\eta^T(\theta(t)) - \boldsymbol{\Phi}(\theta(t))$$

$$= \begin{pmatrix} -f_1\cos\theta(t) - \Delta q_1(t)\sin\theta(t) & -f_1\sin\theta(t) + \Delta q_1(t)\cos\theta(t) & f_1\cos\theta(t) + \Delta q_1(t)\sin\theta(t) & f_1\sin\theta(t) - \Delta q_1(t)\cos\theta(t) & 0 \\ -\Delta q_2(t)\sin\theta(t) & \Delta q_2(t)\cos\theta(t) & \Delta q_2(t)\sin\theta(t) & -\Delta q_2(t)\cos\theta(t) & f_2 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\boldsymbol{\phi}(\mathbf{q}(t)) = \gamma(\mathbf{q}(t), 1)^T - \boldsymbol{\delta} = [(\lambda_2 q_1(t) - f_1\lambda_1 - q_{10}\lambda_2), (\lambda_2 q_2(t) - f_2\lambda_3 - q_{20}\lambda_2), 0]^T$$

where $\Delta q_1(t) = q_1(t) - q_{10}$, and $\Delta q_2(t) = q_2(t) - q_{20}$.

By summarizing (33) and (34), we obtain

$$\dot{V}(t) = -k_x \hat{e}_x^2(t) - k_\theta e_\theta^2(t) - s^T(t)\mathbf{K}\mathbf{s}(t). \quad (35)$$

Therefore, $V(t)$ is bounded, so are $\mathbf{e}(t)$ and the estimation error $(\tilde{x}(t), \tilde{y}(t), \tilde{\mathbf{b}}^T(t))^T$. With the bounded estimation errors and (28), $\mathbf{s}(t)$ is bounded as well. With bounded $\mathbf{e}(t)$ and $(\tilde{x}(t), \tilde{y}(t), \tilde{\mathbf{b}}^T(t))^T$, $\hat{e}_x(t)$ and $\hat{e}_y(t)$ are bounded based on their definitions. From the online position estimator (29), $(\dot{\hat{x}}(t), \dot{\hat{y}}(t), \dot{\hat{\mathbf{b}}}(t))^T$ and $(\dot{\tilde{x}}(t), \dot{\tilde{y}}(t), \dot{\tilde{\mathbf{b}}}(t))^T$ are bounded based on bounded velocity and orientation of the robot and the above boundedness analysis.

Differentiating (35) leads to

$$\ddot{V}(t) = -2k_x \hat{e}_x(t)\dot{\hat{e}}_x(t) - 2k_\theta e_\theta(t)\dot{e}_\theta(t) - 2\mathbf{s}^T(t)\mathbf{K}\dot{\mathbf{s}}(t). \quad (36)$$

From (18) and (28), $\dot{\mathbf{s}}(t), \dot{\hat{e}}_x(t)$, and $\dot{\hat{e}}_y(t)$ are bounded since $(\dot{\tilde{x}}(t), \dot{\tilde{y}}(t), \dot{\tilde{\mathbf{b}}}(t))^T, (\dot{\hat{x}}(t), \dot{\hat{y}}(t), \dot{\hat{\mathbf{b}}}(t))^T$, and the velocity and orientation of the robot are bounded. From (20), $\dot{e}_\theta(t)$ is bounded since $e_\theta(t)$ and $\hat{e}_y(t)$ are bounded. Consequently, $\ddot{V}(t)$ is bounded, which means that $\dot{V}(t)$ is uniformly continuous. From Barbalat's lemma [20], we have $\lim_{t\to\infty}\dot{V}(t) = 0$; then, we can conclude the convergence results in (30). Differentiating (20), $\ddot{e}_\theta(t)$ is bounded since $e_\theta(t)$, $\nu_d(t), \hat{e}_y(t), \dot{e}_\theta(t), \dot{\nu}_d(t)$, and $\dot{\hat{e}}_y(t)$ are bounded; therefore, $\dot{e}_\theta(t)$ is uniformly continuous. Moreover, from (30), $e_\theta(t)$ has a finite limit as $t \to \infty$. Then, based on Barbalat's lemma [20], we can conclude $\lim_{t\to\infty}\dot{e}_\theta(t) = 0$. With $\lim_{t\to\infty}\dot{e}_\theta(t) = 0$ and $\lim_{t\to\infty}\dot{e}_\theta(t) = 0$, from (20), we have

$$\lim_{t\to\infty}\nu_d(t)\hat{e}_y(t) = 0. \quad (37)$$

When $\lim_{t\to\infty}\nu_d(t) \neq 0$, we can conclude (31). ■

It is important to note that the convergence of $(\hat{e}_x(t), \hat{e}_y(t))^T$ does not directly mean the convergence of the position error $(\Delta x(t), \Delta y(t))^T$; therefore, further proof is needed.

Proposition 1: If $\lim_{t\to\infty}\nu_d(t) \neq 0$, with (30) and (31), then

$$\lim_{t\to\infty}\left(\tilde{x}(t), \tilde{y}(t), \tilde{\mathbf{b}}(t)\right)^T = \mathbf{0}, \quad \lim_{t\to\infty}\Delta\mathbf{p}(t) = \mathbf{0}. \quad (38)$$

Proof: In Theorem 1, we have proved that

$$\lim_{t\to\infty}\hat{\mathbf{e}}(t) = \mathbf{0} \Rightarrow \lim_{t\to\infty}\Delta\hat{\mathbf{p}}(t) = \mathbf{0} \Rightarrow \lim_{t\to\infty}\left[\Delta\mathbf{p}(t) + \left(\tilde{x}(t)\ \tilde{y}(t), 0\right)^T\right] = \mathbf{0}. \quad (39)$$

From the proof of Theorem 1, $V(t)$ in (32) is lower bounded and decreasing ($\dot{V}(t) \leq 0$); therefore, when $t \to \infty, V(t)$ converges to a limit [20]. With $\dot{V}(t) \leq 0$ and (21), $(\tilde{x}(t), \tilde{y}(t), \tilde{\mathbf{b}}^T(t))^T$ and hence $\tilde{\mathbf{e}}(t)$ are bounded. From (30), (31), and bounded $\tilde{\mathbf{e}}(t)$, we can conclude that $\lim_{t\to\infty}\hat{\mathbf{e}}(t) = \mathbf{0}$ and $\lim_{t\to\infty}\hat{\mathbf{e}}(t) = \mathbf{0}$. With the former analysis, we have

$$\begin{aligned} V(t) &= \frac{1}{2}\lim_{t\to\infty}\left[(\hat{\mathbf{e}}(t) + \tilde{\mathbf{e}}(t))^T(\hat{\mathbf{e}}(t) + \tilde{\mathbf{e}}(t))\right. \\ &\quad \left. + \left(\tilde{x}(t), \tilde{y}(t), \tilde{\mathbf{b}}^T(t)\right)\mathbf{\Gamma}^{-1}\begin{pmatrix}\tilde{x}(t)\\ \tilde{y}(t)\\ \tilde{\mathbf{b}}(t)\end{pmatrix}\right] \\ &= \frac{1}{2}\lim_{t\to\infty}\left[\tilde{\mathbf{e}}^T(t)\tilde{\mathbf{e}}(t) + \left(\tilde{x}(t), \tilde{y}(t), \tilde{\mathbf{b}}^T(t)\right)\mathbf{\Gamma}^{-1}\begin{pmatrix}\tilde{x}(t)\\ \tilde{y}(t)\\ \tilde{\mathbf{b}}(t)\end{pmatrix}\right]. \end{aligned} \quad (40)$$

Based on (21), $\tilde{\mathbf{e}}^T(t)\tilde{\mathbf{e}}(t)$ can be rewritten as

$$\tilde{\mathbf{e}}^T(t)\tilde{\mathbf{e}}(t) = (\tilde{x}(t), \tilde{y}(t))\mathbf{A}^T(\theta(t))\mathbf{A}(\theta(t))\begin{pmatrix}\tilde{x}(t)\\ \tilde{y}(t)\end{pmatrix}. \quad (41)$$

With Property 4, (40), and (41), we can conclude that $\lim_{t\to\infty}(\tilde{x}(t), \tilde{y}(t), \tilde{\mathbf{b}}^T(t))^T$ exists.

The convergence of $s(t)$ implies that the estimated positions of the robot and the feature point satisfy the projection (4). From the well-known results in computer vision, when both the estimated and true positions satisfy the projection (4), the following equation holds:

$$\lim_{t\to\infty}\left[\hat{\boldsymbol{\beta}}(t) - \alpha(t)\boldsymbol{\beta}(t)\right] = \mathbf{0} \quad (42)$$

where $\alpha(t)$ is a scalar, meaning the projection ambiguity (namely $\boldsymbol{\beta}(t)$ in (4) can be arbitrarily scaled due to the uncertain $z(t)$).

Equation (42) can be rewritten as

$$\begin{aligned} \lim_{t\to\infty}\left[\hat{\boldsymbol{\beta}}(t) - \alpha(t)\boldsymbol{\beta}(t)\right] &= \lim_{t\to\infty}\left[\left(\hat{\boldsymbol{\beta}}(t) - \boldsymbol{\beta}(t)\right)\right. \\ &\quad \left. + \left(1 - \alpha(t)\right)\boldsymbol{\beta}(t)\right] = \mathbf{0}. \end{aligned} \quad (43)$$

From the existence of $\lim_{t\to\infty}(\tilde{x}(t), \tilde{y}(t), \tilde{\mathbf{b}}^T(t))^T$ and (43), we can conclude that $\lim_{t\to\infty}(1 - \alpha(t))\beta(t)$ exists.

To simplify the analysis further, we assume that $b_z \neq 0$. From (42) and the existence of $\lim_{t\to\infty}(1 - \alpha(t))\boldsymbol{\beta}(t)$, we have

$$\lim_{t\to\infty}(\alpha(t) - \hat{b}_z(t)/b_z) = 0 \quad (44)$$

$$\begin{aligned} \lim_{t\to\infty}(1 - \alpha(t))\boldsymbol{\beta}(t) &= \lim_{t\to\infty}[(1 - \hat{b}_z(t)\ b_z)\boldsymbol{\beta}(t) \\ &\quad - (\alpha(t) - \hat{b}_z(t)/b_z)\boldsymbol{\beta}(t)]. \end{aligned} \quad (45)$$

With (44) and bounded $\beta(t)$, we can conclude that $\lim_{t\to\infty}(\alpha(t) - \hat{b}_z(t)/b_z)\beta(t) = 0$; then, from (45), we can conclude the existence of $\lim_{t\to\infty}(1 - \hat{b}_z(t)/b_z)\beta(t)$. Let

$$\mathbf{f}(t) = (1 - \hat{b}_z(t)/b_z)\boldsymbol{\beta}(t). \quad (46)$$

Differentiating (46), we have

$$\dot{\mathbf{f}}(t) = \dot{\hat{b}}_z(t)\boldsymbol{\beta}(t)/b_z + (1 - \hat{b}_z(t)/b_z)(\dot{x}(t), \dot{y}(t), 0)^T. \quad (47)$$

With bounded $(\dot{x}(t), \dot{y}(t))^T$ and $(\ddot{x}(t), \ddot{y}(t))^T$, $(\dot{x}(t), \dot{y}(t), 0)^T$ and $\beta(t)$ are uniformly continuous. With bounded $\mathbf{s}(t)$ and $\dot{\mathbf{s}}(t)$, and considering (29), $\hat{b}_z(t)$ and $\dot{\hat{b}}_z(t)$ are uniformly continuous. As $t \to \infty, \mathbf{f}(t)$ has a finite limit, and $\dot{\mathbf{f}}(t)$ is uniformly continuous. From Barbalat's lemma [20], we have

$$\lim_{t\to\infty}\dot{\mathbf{f}}(t) = 0. \quad (48)$$

From (29), we have $\lim_{t\to\infty}\dot{\hat{b}}_z(t) = 0$. Then, from (47), (48), and bounded $\boldsymbol{\beta}(t)$, we have

$$\lim_{t\to\infty}(1 - \hat{b}_z(t)/b_z)(\dot{x}(t), \dot{y}(t), 0)^T = 0. \quad (49)$$

With $\lim_{t\to\infty}\nu_d(t) \neq 0$ and (19), we have $\lim_{t\to\infty}(\dot{x}(t), \dot{y}(t))^T \neq \mathbf{0}$. Then, from (49) and $\lim_{t\to\infty}(\dot{x}(t), \dot{y}(t))^T \neq \mathbf{0}$, we have $\lim_{t\to\infty}\hat{b}_z(t) = b_z$. With (43), (44), $\lim_{t\to\infty}\hat{b}_z(t) = b_z$, and bounded $\boldsymbol{\beta}(t)$, we have

$$\lim_{t\to\infty}(\hat{\boldsymbol{\beta}}(t) - \boldsymbol{\beta}(t)) = 0. \quad (50)$$

Namely, the estimation of the displacement between the robot and the feature point converges to its real value. If we locate the origin of the world frame at the position of the first feature point with the converged $\mathbf{s}_1(t)$, then based on (50)

$$\lim_{t\to\infty}\left[\begin{pmatrix}\hat{x}(t)\\ \hat{y}(t)\end{pmatrix}-\begin{pmatrix}x(t)\\ y(t)\end{pmatrix}\right]=\mathbf{0}. \tag{51}$$

For the other converged feature points, based on (50) and (51), it is clear that

$$\lim_{t\to\infty}\left(\hat{\mathbf{b}}(t)-\mathbf{b}(t)\right)=0. \tag{52}$$

By summarizing (39), (51), and (52), the convergence result in (38) can be obtained.

The former proof is based on the assumption that $b_z\neq 0$. Practically, we can eliminate the $b_z=0$ case by promising that $\hat{b}_z(t)$ in (29) satisfies $\hat{b}_z(t)\neq 0$ after the convergence of $\mathbf{s}(t)$. ■

E. Discussion

Without the global position measurements, we use the estimated positions as the feedback of the trajectory tracking controller and prove that the tracking error and the estimation error converge to zero, as long as the desired linear velocity is not zero. To implement our algorithm, the orientation and linear/angular velocities of the robot should be measured to a good accuracy, and some visual features should be extracted and continuously tracked.

Remark 1: The proposed estimator guarantees the convergence of the displacement between the feature point and the robot. To describe the positions of the features, we need a frame. Such a frame could be located at one feature point or anywhere else. Another frame (the global frame) is needed to describe the position of the robot. An ideal setting is to make the two frames overlap each other. For this purpose, we need to know the position of one feature point with respect to the global frame. In the unknown environment, such information is not available. One practical solution is that we set the initial position of the robot as the origin of the world frame. Before $\mathbf{s}_i(t)$ for any feature is convergent to a small value, we update the robot position using integration of the encoders and AHRS (IMU&compass) sensors. After $\mathbf{s}_i(t)$ of one feature is smaller than a threshold, the position of the feature can be estimated from the robot's position, and then, the visual feature will be used to estimate the position of the robot. This will cause an offset error, which is a common problem in map registration of mobile robots in unknown environments no matter what sensors and algorithms are used.

Remark 2: The real global velocity of the robot is utilized in the estimator (29). In the applications, the global velocity of the robot can only be measured to some accuracy; therefore, the robustness of our controller against the measurement noises will be investigated in our future work. However, without the theoretical robustness analysis, the experimental results in the following section have shown the robust performance of our controller against the long-term accumulation errors of the noisy measurements.

Remark 3: Multiple feature points with the converged nominal estimation error provide multiple robot position candidates, whose average is taken as the estimated robot position. In the tracking control process, generally, one feature point cannot always stay in the field of view (FOV) of the camera. During the motion of the robot, some feature points will move away from the FOV and new feature points will move in the FOV. The new feature points will replace those moving away. This could cause concerns about the continuity of the system, but the experiments demonstrate the satisfactory performance of our algorithm. The effect of the discontinuity on the stability is a topic to be addressed in the future.

Fig. 2. Wheeled robot in its experimental environment and the snapshots taken by the on-board camera during the experiments.

Remark 4: We do not register any information of the environment in the proposed algorithm. Therefore, regardless of whether the same features are detected, the robot treats them as new features. Certainly, including some loop-closure checks may help reduce error accumulation, but the controller complexity will be increased. Our algorithm is simple, i.e., not managing any map or information in the process, compared with the SLAM-based methods. The following experiments demonstrate that such a simple algorithm works well.

IV. Experiments

To verify the performance of the proposed controller, we have carried out three experiments on a differential-drive wheeled mobile robot (see Fig. 2). The robot is equipped with *Innalabs* miniAHRS (IMU/compass) to measure its orientation/angular velocity at 120 Hz and two encoders to measure the linear velocity at 2 kHz. A low-cost calibrated camera is mounted on the robot to capture the right-side view, as shown in Fig. 2.

The right-side facing camera provides images with a resolution of 640 × 480 at 30 frames/s, and some snapshots taken during the experiments are shown in Fig. 2. No artificial feature points were used in the experiments. The SURF points were extracted from the natural surrounding environment of the robot, and their relative motions with respect to the robot on the image plane were tracked for the position estimation. To realize the real-time efficiency, extraction and tracking of 100–200 SURF for every image frame were performed in parallel by the GPU *Geforce* GTX 580 3 GB. The processor of the on-board computer is 3.4-GHz *Intel Core i7-2600*, with 4-GB RAM. The frequency of the control system is 30 Hz. The *OptiTrack* camera system installed on the ceiling is employed to measure the trajectory of the robot, which is used as the ground truth to evaluate the errors of the trajectory tracking and the position estimation.

In the first experiment, the robot traces the following circle trajectory:

$$\begin{cases}(x_d(t),y_d(t),\theta_d(t))^T=\left(R\cdot\sin(\theta_d(t)),d-R\cdot\cos(\theta_d(t)),\right.\\ \quad\left.\omega_d(t)\cdot t\right)^T\\ (\omega_d(t),\nu_d(t))^T=((2\pi/25)\,\text{rad/s},(\pi/25)\,\text{m/s})^T\end{cases}$$
$$R=0.5\text{ m},\quad d=0.7\text{ m},\quad 0\text{ s}\le t\le 250\text{ s}. \tag{53}$$

The running time is 250 s, which corresponds to moving along the circle ten times. Fig. 3(a) shows the desired trajectory, the trajectory measured by OptiTrack, and the trajectory estimated by our position estimator. The corresponding position/orientation errors and linear/angular velocity errors are shown in Figs. 4 and 5, respectively.

As shown in Fig. 4, the errors between the estimated position and that measured by OptiTrack are within a few centimeters. Using our position estimator and controller, the experimental results confirmed the convergence of the circle trajectory tracking error and the position estimation errors to zero.

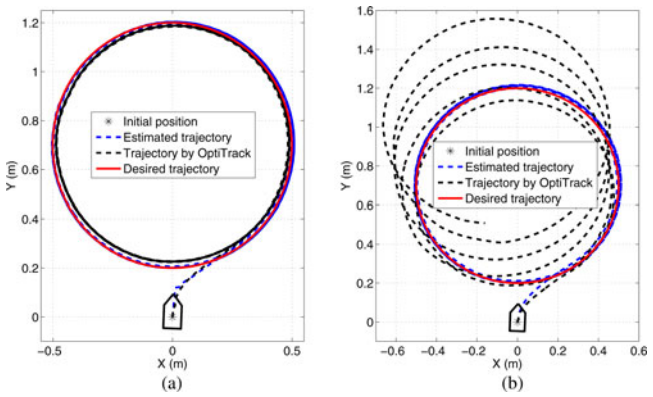

Fig. 3. (a) Results of the circle trajectory tracking experiment by our estimator. (b) Results of the circle trajectory tracking experiment by integration.

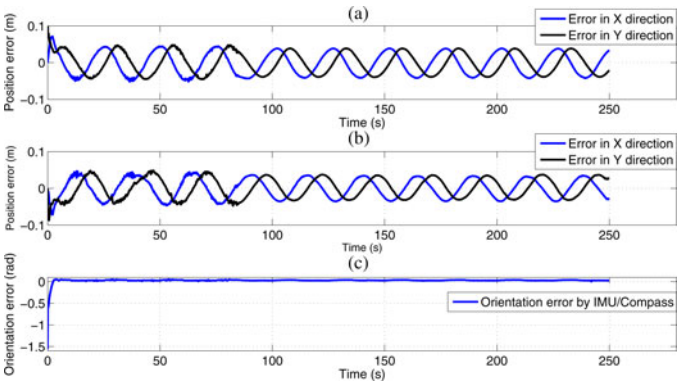

Fig. 4. (a) Position errors between the desired and estimated trajectories. (b) Position errors between the trajectory measured by OptiTrack and the estimated one by the robot. (c) Orientation error between the desired one and that measured by IMU/Compass.

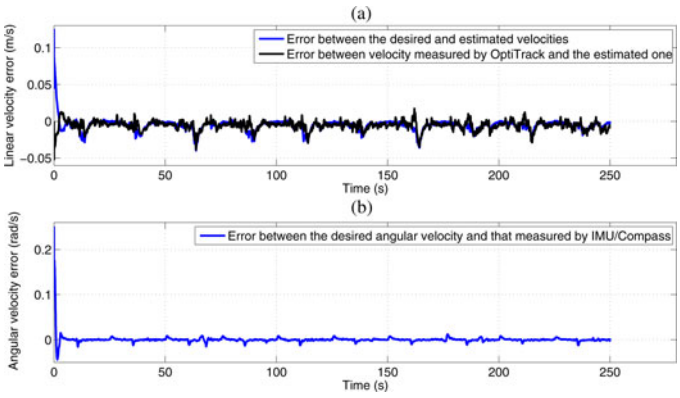

Fig. 5. (a) Linear velocity errors. (b) Angular velocity errors.

In order to show the good performance of the position estimator (29), we have conducted another experiment where the position estimator (29) is substituted by the integration of the encoder and AHRS sensors. The corresponding experimental results are shown in Figs. 3(b), 6, and 7.

As shown in Fig. 3(b), by replacing our position estimator (29) with the integration of the encoder and AHRS sensors, the ground truth trajectory measured by the OptiTrack system deviates a lot from the desired trajectory. The corresponding experimental results are shown in Figs. 6 and 7. The robot just moves along the circle

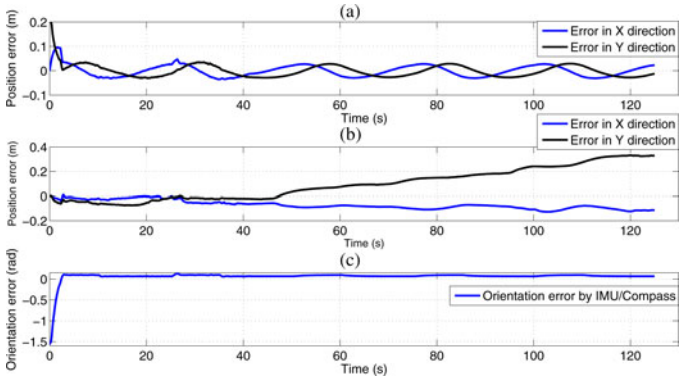

Fig. 6. (a) Position errors between the desired and estimated trajectories. (b) Position errors between the estimated trajectory by the robot and that measured by OptiTrack. (c) Orientation error between the desired one and that measured by IMU/Compass.

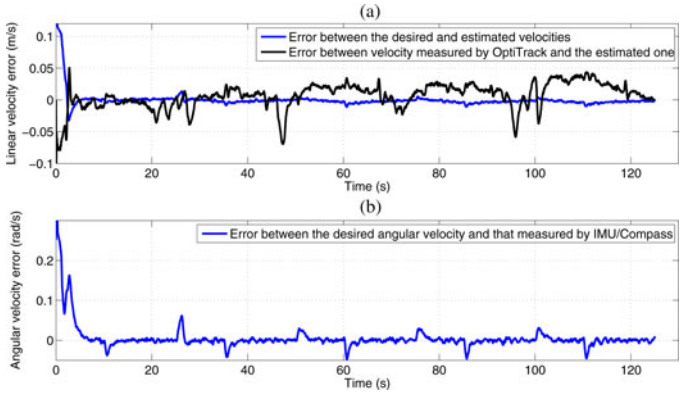

Fig. 7. (a) Linear velocity errors. (b) Angular velocity errors.

five times in the experiment due to the divergence of the robot's trajectory from the desired one and the limited size of the experimental area. Comparing Fig. 3(a) and (b), it is clear to see that in the two cases, both the estimated trajectories converge to the desired trajectories and it is promised by the controller (19) and the estimator (29), but the actual trajectory (by OptiTrack) converges to the desired trajectory only by using our position estimator that has the robust performance against the long-term accumulation errors of the noisy measurements.

In the second experiment, the robot traces the following line trajectory:

$$\begin{cases} \left(x_d\left(t\right), y_d\left(t\right), \theta_d\left(t\right)\right)^T = \left(\nu_d\left(t\right)\cdot t/\sqrt{2}, d + \nu_d\left(t\right)\cdot t/\sqrt{2}\right. \\ \quad \left.\pi/4\text{rad}\right)^T \\ \left(\omega_d\left(t\right), \nu_d\left(t\right)\right)^T = \left(0\ \text{rad/s}, \sqrt{2}/15\ \text{m/s}\right)^T \end{cases}$$
$$d = 0.2\ \text{m}, \quad 0\ \text{s} \le t \le 15\ \text{s}. \tag{54}$$

Fig. 8(a) shows the desired trajectory, the trajectory measured by OptiTrack, and the trajectory estimated by our position estimator. The corresponding position/orientation errors and linear/angular velocity errors are shown in Figs. 9 and 10, respectively.

As shown in Figs. 9 and 10, by our position estimator and controller, the experimental results verified the convergence of the line trajectory tracking error and the position estimation errors to zero.

Similar to the previous circle tracking experiments, we have performed another line tracking experiment by substituting the position estimator (29) with the integration of the encoder and AHRS sensors.

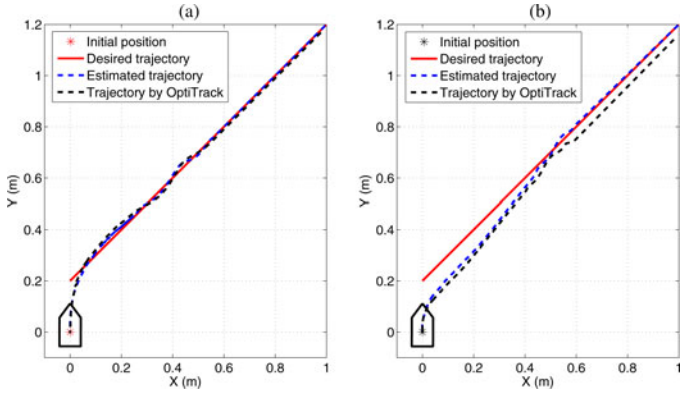

Fig. 8. (a) Results of the line trajectory tracking experiment by our estimator. (b) Results of the line trajectory tracking experiment by integration.

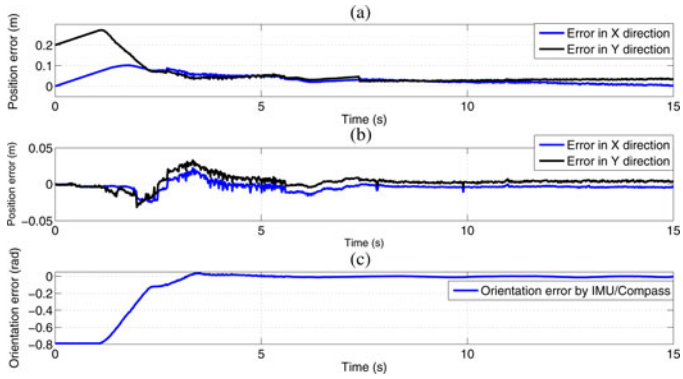

Fig. 9. (a) Position errors between the desired and estimated trajectories. (b) Position errors between the estimated trajectory by the robot and that measured by OptiTrack. (c) Orientation error between the desired one and that measured by IMU/Compass.

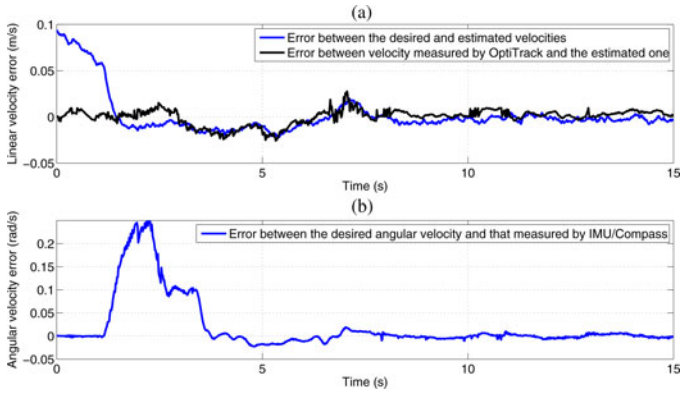

Fig. 10. (a) Linear velocity errors. (b) Angular velocity errors.

The corresponding experimental results are shown in Figs. 8(b), 11, and 12. The comparison of Figs. 8(a) and (b), similar to the previous circle tracking, shows that both the estimated trajectories converge to the desired trajectories in the two cases based on the controller (19) and the estimator (29), but the actual trajectory (by OptiTrack) converges to the desired trajectory only by using our position estimator that is robust to the long-term accumulation errors of the noisy measurements.

The experimental results of trajectory tracking show that by using our controller and online position estimator, the desired trajectories (circle and line) can be asymptotically tracked, and the estimations of

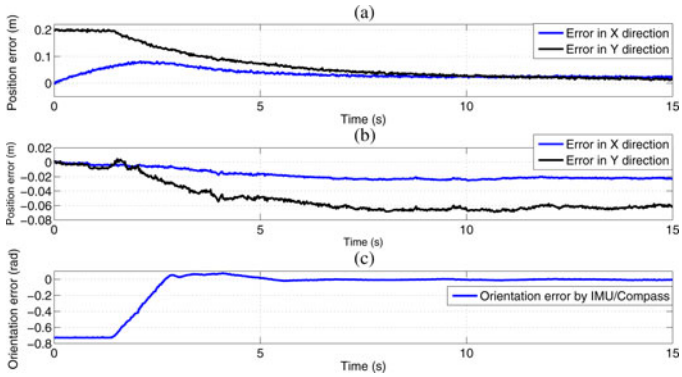

Fig. 11. (a) Position errors between the desired and estimated trajectories. (b) Position errors between the estimated trajectory by the robot and that measured by OptiTrack. (c) Orientation error between the desired one and that measured by IMU/Compass.

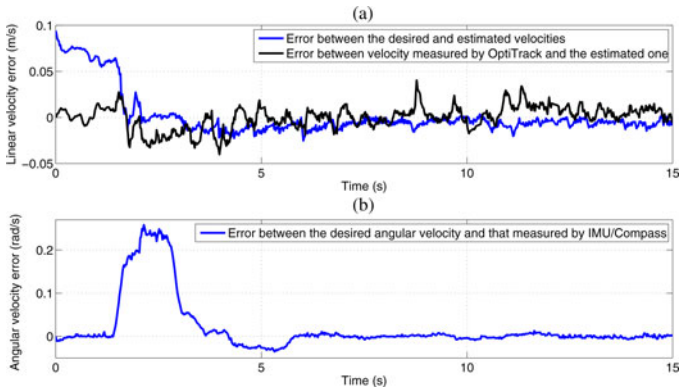

Fig. 12. (a) Linear velocity errors. (b) Angular velocity errors.

the robot positions are promised to converge to the actual positions (by OptiTrack). Meanwhile, our algorithm shows robust performance against the long-term accumulation errors of the noisy measurements. The experimental results of the trajectory tracking prove the validity of our algorithm.

V. Conclusion

This paper has proposed a new controller for visual servoing trajectory tracking of nonholonomic mobile robots without direct position measurement. To avoid measurement of the global position of the robot that is usually necessary in the existing trajectory tracking controllers, a novel adaptive estimator has been designed to estimate the global position online, using natural visual features measured by a vision system, and its orientation and velocity measured by the odometry and AHRS sensors. It has been proved by the Lyapunov theory that the proposed controller, together with the embedded position estimator, gives rise to the asymptotic tracking of a desired trajectory and convergence of the position estimation to the actual position. To ensure the real-time performance, GPU has been adopted to accelerate the online computation by processing multiple SURF points in parallel. The experimental results have verified the validity of the proposed algorithm and demonstrated its robust performance against the long-term accumulation errors of the noisy measurements. The effect of discontinuity, which is caused by features entering and leaving FOV of the camera, on the stability of our algorithm is an open theoretical topic to be addressed in the future.

ACKNOWLEDGMENT

The authors would like to acknowledge the anonymous reviewers for their constructive comments and for pointing out an error in the stability proof.

REFERENCES

[1] R. A. Brooks, "A robust layered control system for a mobile robot," *IEEE J. Robot. Autom.*, vol. RA-2, no. 1, pp. 14–23, Mar. 1986.

[2] R. W. Brockett, "Asymptotic stability and feedback stabilization," in *Differential Geometric Control Theory*. Boston, MA, USA: Birkhäuser, 1983, pp. 181–191.

[3] J. M. Yang and J. H. Kim, "Sliding mode control for trajectory tracking of nonholonomic wheeled mobile robots," *IEEE Trans. Robot. Autom.*, vol. 15, no. 3, pp. 578–587, Jun. 1999.

[4] Y. Kanayama, Y. Kimura, F. Miyazaki, and T. Noguchi, "A stable tracking control method for an autonomous mobile robot," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1990, vol. 1, pp. 384–389.

[5] Z. P. Jiang and H. Nijmeijer, "Tracking control of mobile robots: A case study in backstepping," *Automatica*, vol. 33, pp. 1393–1399, 1997.

[6] J. P. Hespanha and A. S. Morse, "Stabilization of nonholonomic integrators via logic-based switching," *Automatica*, vol. 35, pp. 385–393, 1999.

[7] G. Reina, A. Vargas, K. Nagatani, and K. Yoshida, "Adaptive Kalman filtering for GPS-based mobile robot localization," in *Proc. IEEE Int. Workshop Safety, Security Rescue Robot.*, 2007, pp. 1–6.

[8] M. Betke and L. Gurvits, "Mobile robot localization using landmarks," *IEEE Trans. Robot. Autom.*, vol. 13, no. 2, pp. 251–263, Apr. 1997.

[9] P. Bonnifait and G. Garcia, "Design and experimental validation of an odometric and goniometric localization system for outdoor robot vehicles," *IEEE Trans. Robot. Autom.*, vol. 14, no. 4, pp. 541–548, Aug. 1998.

[10] G. L. Mariottini, G. Oriolo, and D. Prattichizzo, "Image-based visual servoing for nonholonomic mobile robots using epipolar geometry," *IEEE Trans. Robot.*, vol. 23, no. 1, pp. 87–100, Feb. 2007.

[11] J. B. Coulaud, G. Campion, G. Bastin, and M. De Wan, "Stability analysis of a vision-based control design for an autonomous mobile robot," *IEEE Trans. Robot.*, vol. 22, no. 5, pp. 1062–1069, Oct. 2006.

[12] A. Cherubini, F. Chaumette, and G. Oriolo, "Visual servoing for path reaching with nonholonomic robots," *Robotica*, vol. 29, no. 7, pp. 1037–1048, 2011.

[13] C. Y. Tsai, K. T. Song, X. Dutoit, H. Van Brussel, and M. Nuttin, "Robust visual tracking control system of a mobile robot based on a dual-Jacobian visual interaction model," *Robot. Auton. Syst.*, vol. 57, no. 6/7, pp. 652–664, 2009.

[14] J. Chen, W. E. Dixon, D. M. Dawson, and M. McIntyre, "Homography-based visual servo tracking control of a wheeled mobile robot," *IEEE Trans. Robot.*, vol. 22, no. 2, pp. 406–415, Apr. 2006.

[15] Y. H. Liu, H. Wang, C. Wang, and K. K. Lam, "Uncalibrated visual servoing of robots using a depth-independent interaction matrix," *IEEE Trans. Robot.*, vol. 22, no. 4, pp. 804–817, Aug. 2006.

[16] H. Bay, T. Tuytelaars, and L. V. Gool, "SURF: Speeded up robust features," in *Proc. Eur. Conf. Comput. Vis.*, 2006, pp. 404–417.

[17] Y. Fang, W. E. Dixon, D. M. Dawson, and P. Chawda, "Homography-based visual servo regulation of mobile robots," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 35, no. 5, pp. 1041–1050, Oct. 2005.

[18] D. Tick, A. C. Satici, J. Shen, and N. Gans, "Tracking control of mobile robots localized via chained fusion of discrete and continuous epipolar geometry, IMU and odometry," *IEEE Trans. Cybern.*, vol. 43, no. 4, pp. 1237–1250, Aug. 2013.

[19] K. Wang, Y. H. Liu, and L. Li, "Vision-based tracking control of nonholonomic mobile robots without position measurement," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2013, pp. 5245–5250.

[20] J. J. Slotine and W. Li, *Applied Nonlinear Control*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1991, pp. 123–126.