INTECH
open science | open minds

OPEN ACCESS
ARTICLE

International Journal of Advanced Robotic Systems

Fuzzy Vector Field Orientation Feedback Control-Based Slip Compensation for Trajectory Tracking Control of a Four Track Wheel Skid-Steered Mobile Robot

Regular Paper

Xuan Vinh Ha[1], Cheolkeun Ha[1,*] and Jewon Lee[2]

1 School of Mechanical Engineering, University of Ulsan, Republic of Korea
2 Prigent Ltd, Republic of Korea
* Corresponding author E-mail: hacheo121c@yahoo.co.kr

Received 18 Jan 2013; Accepted 11 Mar 2013

DOI: 10.5772/56355

Abstract Skid-steered mobile robots have been widely used in exploring unknown environments and in military applications. In this paper, the tuning fuzzy Vector Field Orientation (FVFO) feedback control method is proposed for a four track wheel skid-steered mobile robot (4-TW SSMR) using flexible fuzzy logic control (FLC). The extended Kalman filter is utilized to estimate the positions, velocities and orientation angles, which are used for feedback control signals in the FVFO method, based on the AHRS kinematic motion model and velocity constraints. In addition, in light of the wheel slip and the braking ability of the robot, we propose a new method for estimating online wheel slip parameters based on a discrete Kalman filter to compensate for the velocity constraints. As demonstrated by our experimental results, the advantages of the combination of the proposed FVFO and wheel slip estimation methods overcome the limitations of the others in the trajectory tracking control problem for a 4-TW SSMR.

Keywords Skid-Steered Mobile Robot (SSMR), Path Tracking Control, Vector Field Orientation (VFO) Method, Fuzzy Logic Control (FLC), Extended Kalman Filter (EKF), Slip Estimation

1. Introduction

In recent years, many mobile robotic system applications have been developed. Several types of system have been designed and manufactured based on the locomotion systems of robots. With a 4-TW SSMR, there is no steering mechanism and the direction of motion is changed by turning the left and the right side tracked wheels at different velocities. Because of complex wheel-ground interactions and kinematic constraints, the path tracking control of the mobile robot makes it difficult to obtain an accurate experimental result.

In the literature, there are several approaches to solving the path tracking problems of a mobile robot. For example, a combination of kinematic and torque control frameworks using a backstepping technique to join robot kinematics into dynamics allows us to apply approaches from a model-dependent computed torque method (CTM) to a robust sliding mode control (SMC) method [1]. In [2], the adaptive SMC is designed for an uncertain dynamic model with parametric uncertainties associated with the camera system. Another type of controller for an autonomous mobile robot is the stable tracking control method which is proposed based on an error model of the kinematic model [3]. Although there are many interesting features to all these control methods, they are difficult to tune, in contrast to the flexibility of the fuzzy logic control (FLC). The fuzzy logic controller for the path tracking of a wheeled mobile robot is presented in [4]. A fuzzy adaptive tracking control method for wheeled mobile robots is proposed, where unknown slippage occurs and violates the nonholonomic constraint in the form of state-dependent kinematic and dynamic disturbances [5]. In [6], a pseudo-static friction model is used to capture the interaction between the wheels and the ground, and an adaptive control algorithm is designed to simultaneously estimate the wheel/ground contact friction information and control the mobile robot's trajectory as desired.

In previous work, the novel Vector Field Orientation (VFO) feedback control method was developed for a Differentially Driven Vehicle (DDV), a Threecycle Mobile Robot and a 4-WD SSMR in [7-9]. Also in [10], the extended VFO control method for a mobile vehicle was proposed to compensate for skid-slip phenomena.

In a mobile robot's field of operation, wheel slip limits traction and braking ability, especially for a four track-wheel differentially skid-steered mobile robot (4-TWD SSMR). Accurate estimation of slip is essential to obtaining the precise position of a mobile robot operating in unstructured terrain. Several approaches to slip estimation have been developed. For example, using a sliding mode observer (SMO) as a means to estimate slip parameters based on the kinematic model of a skid-steering vehicle is proposed in [11]. In another study, a novel method combining an optical flow algorithm with a sliding mode observer to estimate the slip parameters of a SSMR is introduced [12]. However, in order to validate the proposed method, it needs to be tested on more complex trajectories and longer travelling distances. Reference [13] proposes an experimental slip model for exact kinematic modelling and the parameters of this model are determined based on experimental analysis.

The contribution of this paper is the combination of the proposed FVFO and novel wheel slip compensation methods for the trajectory tracking of a mobile robot. First, the tuning fuzzy Vector Field Orientation (FVFO) feedback control method is proposed for a 4-TW SSMR. While the stable tracking control in [3] and conventional VFO feedback control methods in [7-10] do not perform well over a wide range of operating conditions because of the fixed gains used, the FVFO used the FLC to tune parameters of the VFO method. That is the flexibility of the proposed method, thanks to which we are able to obtain more precise trajectories. Second, in the field of mobile robotics, wheel slip limits the traction and braking ability of the robot; the methods in [11-13] retain these limitations when the robot is travelling on different ground surfaces with different and longer trajectory shapes. In this paper, we propose a novel method for online wheel slip estimation based on a discrete Kalman filter to compensate for the velocity constraints. The experimental results show that the combination of the proposed FVFO and novel wheel slip compensation methods overcomes the limitations of the others in the trajectory tracking control problem. This system can be enhanced by using trajectory tracking control for a 4-TW SSMR. The experiment is performed on the NT - Hazard Escape - 1®, as shown in Figure 1.

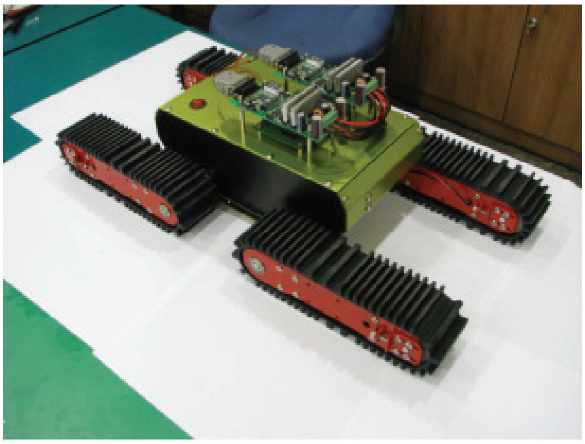

Figure 1. The NT – Hazard escape – 1 Mobile Robot

This paper is organized as follows: Section 2 introduces the experimental NT – Hazard escape – 1 mobile robot setup. Section 3 proposes a tuning fuzzy Vector Field Orientation (FVFO) feedback control method for a mobile robot. The implemented extended Kalman filter and slip estimation are designed in Section 4. The experimental results, including analysis and evaluation, are discussed in Section 5. Finally, Section 6 presents the conclusion.

2. The experimental NT-Hazard escape – 1 mobile robot setup

The NT – Hazard escape – 1 mobile robot, with four NT – track wheels, considered in this study, has been applied in a disaster area, has been used to run through and observe rough terrain including a construction site and has run up stairways. A robot with small actuators was created for the first time in Korea, providing the robots with various functions. Each of the eight motors operates and functions independently. The robot can easily lift 2 tracks even when fully loaded. The robot can move whilst maintaining its central balance like a tank. In addition, the RS232C is supported and enabled by an embedded board and computer which operate through wireless telecommunication. The dimensions of the mobile robot are shown in Figure 2.

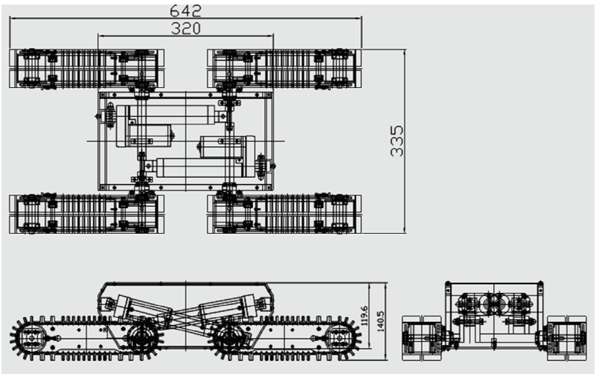

Figure 2. Dimensions of the NT – Hazard escape - 1

	Range	Accuracy	Resolution
Attitude	±180^0	0.1^0	0.002^0
Angular Rate	±3000/S		0.146^0/S
Acceleration	±2G		0.0025G
Heading	±180^0	0.1^0	0.002^0

Table 1. The AHRS – 03 – 300 specifications

For research purposes, an experimental mobile robot was designed. The system configuration is represented by the element component configuration shown in Figure 3, while a real photograph of the experimental system is displayed in Figure 4. Using these figures, the AHRS (AHRS – 03 – 300) was mounted on the top centre of the robot on a base platform. The AHRS specifications are shown in Table 1. Two incremental encoders (E40H-8-1024-3-N-5) were installed on the robot's front left and front right track wheels. A notebook model ASUS U36S was used to collect data from the sensors through an AVR 128-pro board and also performed all the algorithms in real-time. Here, the software control algorithm for the system was coded in the programming language C# (using the Microsoft Visual Studio 2008 version).

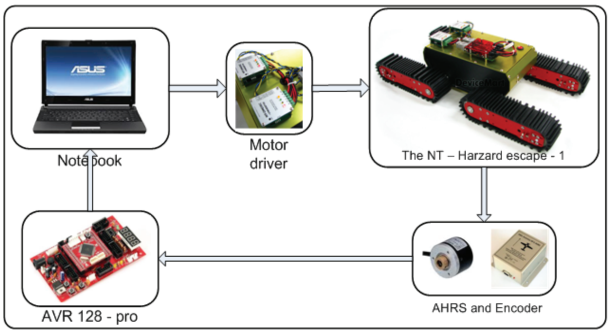

Figure 3. The system configuration

During the experiment, the measurement sampling frequency for the AHRS and incremental encoders was set to 20Hz. The robot was programmed to follow desired trajectories on concrete terrain. Based on the measurements from these two sensors, the robot position values are derived using a Kalman Filtering technique and then the trajectory tracking controllers are applied to drive the robot along desired trajectories.

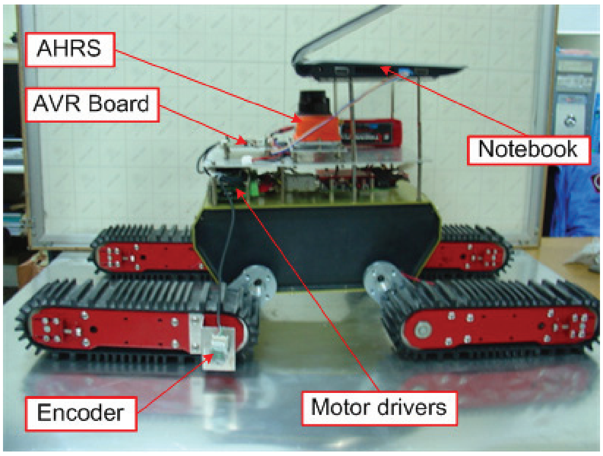

Figure 4. The real photograph of the experimental system

3. The tuning Fuzzy VFO feedback control method for the path tracking of the mobile robot

A path tracking control method is used to control the trajectory of the experimental robot's motion as desired, based on closed–loop control systems and the sensor fusion technique for feedback control signals.

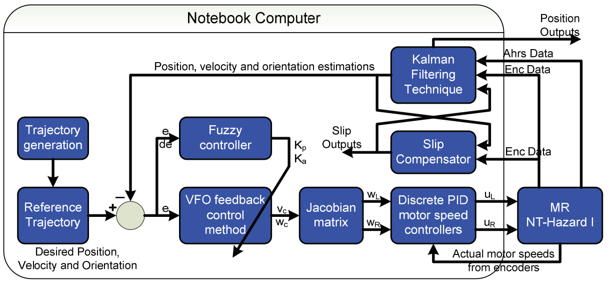

Figure 5. The diagram of the implemented control algorithm

Figure 5 shows a diagram of the implemented control algorithm. As shown in this figure, the control systems include three closed-loop controls: two low-level closed-loop controls for motor speed on the left and right sides, and a high-level closed-loop control for the robot's position. For low-level closed-loop controls, discrete PID controllers are applied, and the tuning fuzzy VFO (FVFO) feedback control method is designed for high-level closed-loop control. Then, the extended Kalman filter (EKFx) is utilized to estimate the positions, velocities and orientation angles, which are used for feedback control signals in the high-level closed-loop control, based on the AHRS kinematic model and velocity constraints. Additionally, a discrete Kalman filter for the slip estimator (KFλ) is proposed to compensate for the velocity constraints for EKFx in order to improve the accuracy of estimation values. The EKFx and KFλ are implemented simultaneously using measurements from the AHRS sensor and encoders. They are introduced in the next sections.

3.1 The VFO feedback control method

The stable tracking control (STC) method is introduced in [3-4], based on an error model of the kinematic model. The linear velocity output *v_c* and the angular velocity output *ω_c* of the control law are:

$$\begin{aligned} v_c &= v_d \cos e_{b\Psi} + K_x e_{bx} \\ \omega_c &= \omega_d + v_d(K_y e_{by} + K_\Psi \sin e_{b\Psi}) \end{aligned} \tag{1}$$

where v_d, ω_d are the desired linear and angular velocities of the mobile robot ; *K_x* , *K_y* and K_Ψ are all positive constants ; and e_{bx}, e_{by} and $e_{b\Psi}$ are three instantaneous error variables associated with the body frame. In this work, the VFO control feedback method is applied to high-level closed-loop control of the robot's position [7-10]. The kinematics modelling of the 4-TW SSMR can be expressed as

$$\begin{bmatrix} \dot{X}_N \\ \dot{Y}_N \\ \dot{\Psi} \end{bmatrix} = \begin{bmatrix} \cos\Psi & 0 \\ \sin\Psi & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_B \\ \omega \end{bmatrix} = \begin{bmatrix} \cos\Psi \\ \sin\Psi \\ 0 \end{bmatrix} v_B + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \omega \tag{2}$$

or

$$\dot{q} = g_1 u_1 + g_2 u_2 \tag{3}$$

where $q = [X_N, Y_N, \Psi]^T$, $g_1 \triangleq \begin{bmatrix} \cos\Psi & \sin\Psi & 0 \end{bmatrix}^T$, $g_2 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$, $u_1 = v_B$ and $u_2 = \omega$. The VFO approach comes from a simple geometrical interpretation of a possible time evolution of Equation (2) or (3).

In the VFO strategy the trajectory tracking control problem is divided into two subtasks: convergence of position and orientation to the desired values. The vehicle is driven by *the pushing control v_B* with a careful pushing strategy, while *the orienting control w* is responsible for matching the vehicle's heading vector with the position convergence vector. Such tasks are accomplished with the help of the choice of a proper convergence vector field, which defines the instantaneous convergence direction and orientation of the vehicle. The control position errors $e_p = [e_x, e_y]^T$ are defined as:

$$e_p = \begin{bmatrix} e_x \\ e_y \end{bmatrix} = \begin{bmatrix} X_d - X_N \\ Y_d - Y_N \end{bmatrix} = q_d^* - q^* \tag{4}$$

where $q_d^* = [X_d, Y_d]^T$ and $q^* \triangleq [X_N, Y_N]^T$ are the desired and actual positions of the mobile robot. The convergence vector field is defined as $h \triangleq [h_1, h_2, h_3]^T = [h_p^T, h_o]^T \in \mathbb{R}^3$ where $h_p \in \mathbb{R}^2$ defines the convergence direction and orientation of the q^* sub-state, and h_o defines the convergence orientation of the Ψ variable. For VFO tracking control, the positional component of the convergence vector field is defined as a linear combination of the control position errors and feedforward terms as follows:

$$h_p = \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} = K_p e_p + \dot{q}_d^*, \quad K_p > 0 \tag{5}$$

where K_p is a design parameter, $\dot{q}_d^*$ is the desired velocity vector.

Using kinematics modelling (2), the simultaneous attenuation of both the position and orientation errors during a transient stage requires highly oscillatory vehicle movement, which usually does not occur in practice [10]. To avoid transient oscillations the third component of h can be designed to introduce the so-called auxiliary orienting variable $\psi_a = \arg(h_p)$ rather than the desired orientation angle Ψ_d. The auxiliary orienting variable ψ_a allows the robot to approach a position trajectory $q_d^* = [X_d, Y_d]^T$ smoothly, and is described by Equation (6) as:

$$\psi_a = A\tan 2c(h_2 \,\mathrm{sgn}(v_d), h_1 \,\mathrm{sgn}(v_d)) \tag{6}$$

where $A\tan 2c(.,.) : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is a continuous version of the four-quadrant function $A\tan 2(.,.) : \mathbb{R} \times \mathbb{R} \to [-\pi, \pi)$ (refer to the appendix in [7]). The auxiliary orienting variable is now introduced as:

$$e_a = \psi_a - \Psi \tag{7}$$

Since the third component, which is the convergence orientation, is:

$$h_o = h_3 = K_a e_a + \dot{\psi}_a, \quad K_a > 0 \tag{8}$$

where K_a is a design parameter, and

$$\dot{\psi}_a = \frac{\dot{h}_2 h_1 - h_2 \dot{h}_1}{h_2^2 + h_1^2} \tag{9}$$

Using the principles of the VFO method, the VFO feedback control law is:

$$u_1 = g_1^{*T} h_p \Rightarrow v_c = u_1 = h_1 \cos\Psi + h_2 \sin\Psi \tag{10}$$

$$u_2 = h_o \Rightarrow \omega_c = u_2 = K_a e_a + \dot{\psi}_a \tag{11}$$

where *v_c* and *ω_c* are the linear and angular velocity outputs from the VFO controller. The left and right angular speed inputs $[\omega_L, \omega_R]$ are computed using *v_c* and *ω_c* by using the Jacobian matrix as follows:

$$\begin{bmatrix} \omega_L \\ \omega_R \end{bmatrix} = \Im \begin{bmatrix} v_c \\ \omega_c \end{bmatrix} = \begin{bmatrix} \frac{1}{r} & \frac{-L}{2r} \\ \frac{1}{r} & \frac{L}{2r} \end{bmatrix} \begin{bmatrix} v_c \\ \omega_c \end{bmatrix} \tag{12}$$

where *r* is the radius of the tracked wheel and *L* is the distance between the left and right sides of the robot.

3.2 The tuning fuzzy VFO feedback control method

From Equations (5) and (8), it can be seen that the position and orienting components of the convergence vector field are similar to a PD control, but the derivative parameter is fixed at one. The K_p and K_a are considered to be proportional parameters. However, the conventional VFO feedback control does not offer reasonable performance over a wide range of operating conditions because of the fixed gains used. That is why fuzzy logic has to be chosen to tune the proportional parameters K_p and K_a of the VFO feedback control automatically. This is called the fuzzy VFO feedback control method (FVFO). Based on the definition of the control position errors $e_p = [e_x, e_y]^T$, Equation (5) can be rewritten as

$$h_p = \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} = \begin{bmatrix} K_{px} & 0 \\ 0 & K_{py} \end{bmatrix} \begin{bmatrix} e_x \\ e_y \end{bmatrix} + \dot{q}_d^*, \quad K_{px}, K_{py} > 0 \quad (13)$$

where the K_p in Equation (5) is separated into two components K_{px}, K_{py} corresponding to e_x and e_y, respectively. Then the three parameters K_{px}, K_{py} and K_a are tuned by using three fuzzy tuners. The detailed fuzzy VFO scheme is shown in Figure 6.

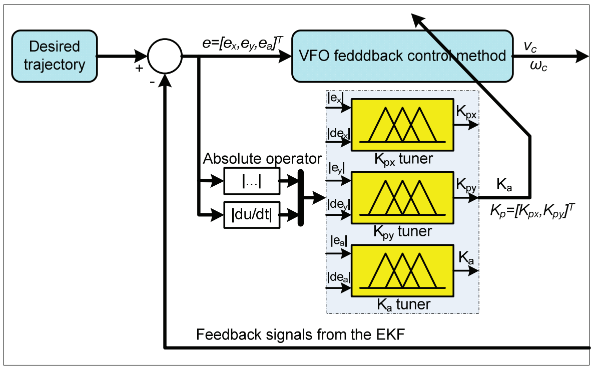

Figure 6. The tuning fuzzy VFO scheme of the high-level closed-loop control

From Figure 6 it can be seen that there are three fuzzy tuners for the three output parameters K_{px}, K_{py} and K_a. Two input signals are needed for each fuzzy tuner [14]; the absolute error $|e|$ and derivative error $|de|$, such as $|e_x|$ and $|de_x|$ for the K_{px} tuner, $|e_y|$ and $|de_y|$ for the K_{py} tuner, and $|e_a|$ and $|de_a|$ for the K_a tuner. The ranges of these inputs are from 0 to 5, which are obtained from the absolute values of the system error and its derivative through the scale factors chosen empirically from experiments. Triangle and trapeze membership functions are then utilized to create the fuzzy input partitions. Here, five membership functions (VS, S, M, B, VB) representing the five input states (very small, small, medium, big, very big), respectively, are used for the controller. Details of the fuzzy inputs' membership functions are shown in Figures 7 and 8 (a = b = 5).

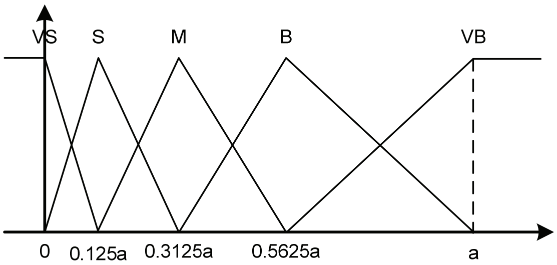

Figure 7. Membership functions of inputs $|e_x|$, $|e_y|$ and $|e_a|$

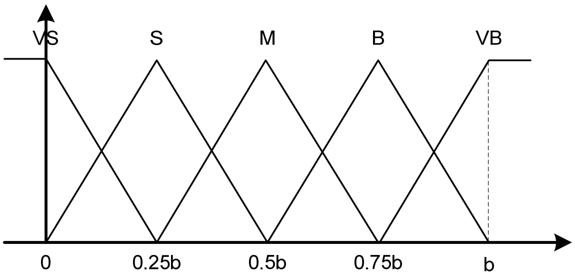

Figure 8. Membership functions of inputs $|de_x|$, $|de_y|$ and $|de_a|$

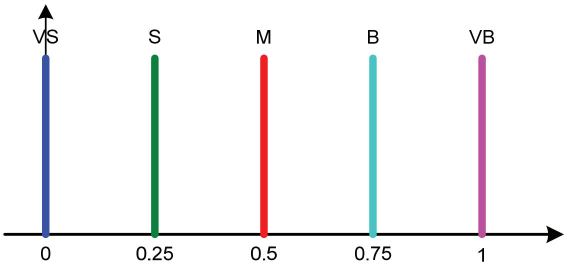

Figure 9. Membership functions of the outputs kp_x, kp_y and kp_a

There are three outputs from the three fuzzy tuners: k_{px}, k_{py} and k_a with the outputs having ranges from 0 to 1. Singleton membership functions are then used for the fuzzy output partitions. Figure 9 shows five membership functions (VS, S, M, B, VB) corresponding with the five output states (very small, small, medium, big, very big), respectively. Because the same properties of the three output parameters K_{px}, K_{py} and K_a are used as proportional parameters, fuzzy rules are composed generally as follows, using the above fuzzy sets of input and output variables:

Rule *i*: If $|e_x|$ ($|e_y|$ or $|e_a|$) is A_i and $|de_x|$ ($|de_y|$ or $|de_a|$) is B_i then k_{px} (k_{py} or k_a) is C_i, $i = 1, 2, \ldots, n$

where *n* is the number of fuzzy rules; A_i, B_i and C_i are the i^{th} fuzzy sets of the input and output variables used in the fuzzy rules. A_i, B_i and C_i are also the linguistic variable values of the input and output signals in the fuzzy tuners.

kpx,kpy,ka		|de|				
		VS	**S**	**M**	**B**	**VB**
|e|	**VS**	VS	VS	VS	VS	VS
	S	M	M	S	S	S
	M	B	B	M	M	M
	B	VB	VB	M	M	M
	VB	VB	VB	VB	VB	VB

Table 2. Rule table of the fuzzy tuners

The rules of the fuzzy tuners are designed as shown in Table 2. The MAX-PROD formula is chosen as the main strategy for the implication process:

$$\mu_{\text{out}}^{\text{i}} = \max\left(\mu(\text{e}) \cdot \mu(\text{de})\right) \tag{14}$$

where $\mu(\text{e})$, $\mu(\text{de})$ are membership values with respect to input variables; $\mu_{\text{out}}^{\text{i}}$ is the membership value with respect to the output variable at the *ith* rule. The centroid defuzzification method is used to convert the aggregated fuzzy set to a crisp output value. In this case, because the membership functions for the fuzzy output partitions are in Singleton form, the outputs of fuzzy tuners are calculated as:

$$\text{y}_{\text{out}} = \frac{\sum_{\text{i}=1}^{25} \mu_{\text{out}}^{\text{i}} \cdot \text{y}_{\text{out}}^{\text{i}}}{\sum_{\text{i}=1}^{25} \mu_{\text{out}}^{\text{i}}} \tag{15}$$

where $\text{y}_{\text{out}}^{\text{i}}$ is the output value of the *ith* rules which can be determined in Figure 9; the output of the fuzzy tuner y_{out} is *kpx*, *kpy* or *ka*. Then, these output values of the fuzzy tuners are substituted into Equation (16) to compute the proportional parameters K_{px}, K_{py} and K_{a} in Equations (8) and (13).

$$\begin{aligned} \text{K}_{\text{px}} &= \text{K}_{\text{px min}} + \text{k}_{\text{px}}(\text{K}_{\text{px max}} - \text{K}_{\text{px min}}) \\ \text{K}_{\text{py}} &= \text{K}_{\text{py min}} + \text{k}_{\text{py}}(\text{K}_{\text{py max}} - \text{K}_{\text{py min}}) \\ \text{K}_{\text{a}} &= \text{K}_{\text{a min}} + \text{k}_{\text{a}}(\text{K}_{\text{a max}} - \text{K}_{\text{a min}}) \end{aligned} \tag{16}$$

where $[\text{K}_{\text{px max}}, \text{K}_{\text{px min}}]$, $[\text{K}_{\text{py max}}, \text{K}_{\text{py min}}]$ and $[\text{K}_{\text{a max}}, \text{K}_{\text{a min}}]$ are the ranges of K_{px}, K_{py} and K_{a}, respectively. In this paper, these ranges are set from 2 to 4.5and they are determined from experiments.

4. Extended Kalman Filter (EKFx) and Slip estimation (KFλ) design

4.1 AHRS kinematic model and velocity constraints

4.1.1 AHRS kinematics model

We define a navigational reference frame N(X,Y,Z) and robot body frame B(x,y,z) as shown in Figure 10. Let $P_N(t)=[X_N(t), Y_N(t), Z_N(t)]^T \in \mathbb{R}^3$, $V_N(t)=[V_x(t), V_y(t), V_z(t)]^T \in \mathbb{R}^3$ and $\Theta = [\phi, \theta, \psi]^T \in \mathbb{R}^3$ denote the position, the velocity and the attitude angle vectors of AHRS in the N frame, respectively. We also define AHRS acceleration, angular rate measurement, acceleration walking bias and angular rate walking bias in the B frame as $a_B = [a_{Bx}, a_{By}, a_{Bz}]^T \in \mathbb{R}^3$, $\omega_B = [\omega_{Bx}, \omega_{By}, \omega_{Bz}]^T \in \mathbb{R}^3$, $b_{aB} = [b_{aBx}, b_{aBy}, b_{aBz}]^T \in \mathbb{R}^3$ and $b_{gB} = [b_{gBx}, b_{gBy}, b_{gBz}]^T \in \mathbb{R}^3$, respectively.

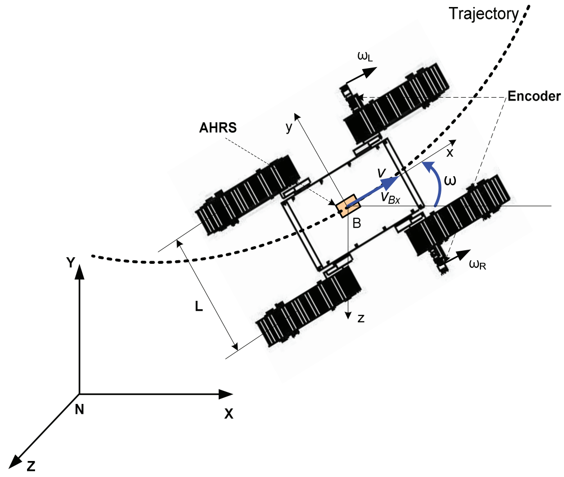

Figure 10. An AHRS kinematics model of a 4TW SSMR

After subtracting the constant offset and local gravity vector, the acceleration and angular rate models can be described by [15]:

$$\text{a}_{\text{B}} = \ddot{\text{x}}_{\text{B}} + \text{b}_{\text{aB}} + \text{w}_{\text{accel}} \tag{17}$$

$$\omega_{\text{B}} = \text{r}_{\text{B}} + \text{b}_{\text{gB}} + \text{w}_{\text{gyros}} \tag{18}$$

with

$$\dot{\text{b}}_{\text{aB}} = -\frac{1}{\tau_{\text{a}}}\text{b}_{\text{aB}} + \sqrt{\frac{2\text{f}_{\text{s}}\sigma_{\text{baB}}^2}{\tau_{\text{a}}}}\text{w}_{\text{accbias}} \tag{19}$$

$$\dot{\text{b}}_{\text{gB}} = -\frac{1}{\tau_{\text{g}}}\text{b}_{\text{gB}} + \sqrt{\frac{2\text{f}_{\text{s}}\sigma_{\text{bgB}}^2}{\tau_{\text{g}}}}\text{w}_{\text{gyrobias}} \tag{20}$$

where the true acceleration vector, the true angular rate vector of the vehicle in the B frame, the acceleration white noise and the angular rate white noise are $\ddot{\text{x}}_{\text{B}} = [\ddot{\text{x}}_{\text{Bx}}, \ddot{\text{x}}_{\text{By}}, \ddot{\text{x}}_{\text{Bz}}]^{\text{T}}$, $\text{r}_{\text{B}} = [\text{r}_{\text{Bx}}, \text{r}_{\text{By}}, \text{r}_{\text{Bz}}]^{\text{T}}$, w_{accel} and w_{gyros}, respectively. And τ_{a}, σ_{baB}^2, $\text{w}_{\text{accbias}}$, τ_{g}, σ_{bgB}^2 and $\text{w}_{\text{gyrobias}}$ are time constants, noise variances and zero-mean white noises with $\text{E}[\text{w}^2] = 1$ of acceleration walking bias and angular rate walking bias and f_{s} is the sampling frequency.

We define the state vector $\chi(\text{t}) = \left[\text{P}_{\text{N}}(\text{t}), \text{V}_{\text{N}}(\text{t}), \Theta(\text{t}), \text{b}_{\text{aB}}(\text{t}), \text{b}_{\text{gB}}(\text{t})\right] \in \mathbb{R}^{15}$ of the model process. The kinematic motion equation for AHRS can be simplified as:

$$\dot{\chi} = \begin{bmatrix} V_I \\ C_B^N\left(a_B - b_{aB} + w_{accel}\right) \\ q_\Theta\left(\omega_B - b_{gB} + w_{gyros}\right) \\ -\dfrac{1}{\tau_a} b_{aB} \\ -\dfrac{1}{\tau_g} b_{gB} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \sqrt{\dfrac{2f_s\sigma_{baB}^2}{\tau_a}} w_{accbias} \\ \sqrt{\dfrac{2f_s\sigma_{bgB}^2}{\tau_g}} w_{gyrobias} \end{bmatrix} \quad (21)$$

where C_B^N and q_Θ are the transformation matrix from the B frame to the N frame and transformation matrix of the Euler angles, as given by the following matrices:

$$C_B^N = \begin{bmatrix} c_\theta c_\psi & -s_\psi c_\phi + c_\psi s_\phi s_\theta & s_\phi s_\psi + c_\psi s_\theta c_\phi \\ c_\theta s_\psi & c_\phi c_\psi + s_\theta s_\phi s_\psi & -s_\phi c_\psi + s_\theta c_\phi s_\psi \\ -s_\theta & c_\theta s_\phi & c_\phi c_\theta \end{bmatrix} \quad (22)$$

$$q_\Theta = \begin{bmatrix} 1 & s_\phi t_\theta & c_\phi t_\theta \\ 0 & c_\theta & -s_\theta \\ 0 & s_\phi / c_\theta & c_\phi / c_\theta \end{bmatrix} \quad (23)$$

where $c_\theta = \cos\theta, s_\theta = \sin\theta, t_\theta = \tan\theta$ and the same notation convention is used for angles ϕ and ψ .

4.1.2 Velocity constraints

In the 4-TW SSMR, the motors that power the wheels on each side are geared internally to ensure that the velocity of the two adjacent wheels on each side are synchronized and thus have the same velocity at ground contact. We define v_L, v_R as the left and right linear velocities of the robot, thus, we have:

$$\begin{aligned} v_L &= v_{1x} = v_{2x} \\ v_R &= v_{3x} = v_{4x} \end{aligned} \quad (24)$$

where v_{1x} , v_{2x} , v_{3x} and v_{4x} are the centre linear velocities for the front-left, rear-left, front-right and rear-right wheels, respectively, as shown in Figure 11. The longitudinal wheel slips of the left and right wheels λ_L, λ_R are defined as ratios of the wheel velocities and centre velocities [16], as follows

$$\begin{aligned} \lambda_L &= \frac{r\omega_L - v_L}{r\omega_L} \\ \lambda_R &= \frac{r\omega_R - v_R}{r\omega_R} \end{aligned} \quad (25)$$

where ω_L, ω_R are the wheel angular speeds for the left and right sides of the mobile robot. From Equation (25), it can be seen that the wheel slip is $\lambda \in [0,1]$ if the wheel is under traction and $\lambda \in (-\infty, 0]$ when the wheel is braking. Using the definition of the slip in (25), we have

$$\begin{aligned} v_L &= (1-\lambda_L) r\omega_L \\ v_R &= (1-\lambda_R) r\omega_R \end{aligned} \quad (26)$$

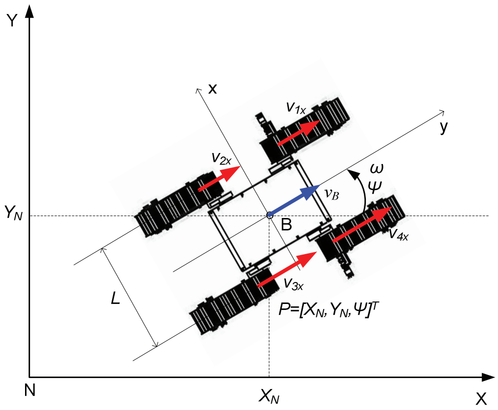

Figure 11. The 4-TW SSMR kinematics

Because of the symmetric mechanical structure of the robot, it can be assumed that the centre of mass (COM) of the robot is located at the centre of the geometry (COG) of the body frame. The AHRS coordinate is located at the COG of the body frame in the B frame, as shown in Figure 10. Using two wheel encoders (one for each side of the vehicle), we obtain the AHRS velocity vector $v_B = [v_{Bx}, v_{By}, v_{Bz}]^T$ in the B frame. From (26), using two wheel encoder measurements and estimated slip parameters, we obtain the longitudinal velocity v_{Bx} as:

$$v_{Bx} = \frac{v_L + v_R}{2} = \frac{r}{2}\left[(1-\lambda_L)\omega_L + (1-\lambda_R)\omega_R\right] \quad (27)$$

Based on [16], since the four tracked wheels of our robot are always in contact with the ground and since the AHRS is fixed on the robot platform, the velocity constraints v_{By} , v_{Bz} in the *y*-axis and *z*-axis directions for the AHRS device can be simplified to equal zero. The noises in the longitudinal velocity v_{Bx} can be expressed as the sum of the noise of the left and right wheels' angular speeds ω_L and ω_R as measured by the encoders. It is assumed that the noises for ω_L and ω_R have a normal distribution with a zero mean Gaussian and corresponding variances, and there is no cross-correlation between the noise of ω_R and of ω_L [17]. The noise variance of the longitudinal velocity v_{Bx} can be expressed as:

$$\sigma_{vBx}^2 = \frac{r^2}{4}\left((1-\lambda_R)^2 \operatorname{var}(\omega_R) + (1-\lambda_L)^2 \operatorname{var}(\omega_L)\right) \quad (28)$$

The variances σ_{vBy}^2 , σ_{vBz}^2 of the lateral velocity v_{By} and the ground surface topography v_{Bz} , respectively, are also expressed by zero. In order to obtain the longitudinal velocity v_{Bx} and its noise variance, we have to estimate slips λ_L and λ_R . A new method to estimate these two slip values is introduced in Section 4.3

4.2 Extended Kalman Filter (EKFx) design

Now, we define the state variable's vector $X(k) = [P_N(k), V_N(k), \Theta(k), b_{aB}(k), b_{gB}(k)]^T \in \mathbb{R}^{15}$. We rewrite the AHRS kinematics equation shown in (21) in discrete-time form, then we obtain the process model as:

$$\begin{aligned} X(k) &= f(X(k-1),u(k),w(k),\Delta T) \\ &= X(k-1)+\Delta T * g(X(k-1),u(k),w(k)) \end{aligned} \tag{29}$$

where the AHRS input signals at the k^{th} sampling time are $u(k)=[a_B(k),\omega_B(k)]^T \in \mathbb{R}^6$, and the data sampling period is ΔT. Also, the nonlinear function g of the process model is:

$$\begin{aligned} g(X(k-1),u(k),w(k)) &= \begin{bmatrix} g_P \\ g_V \\ g_\Theta \\ g_{aB} \\ g_{gB} \end{bmatrix} \\ &= \begin{bmatrix} V_I(k-1) \\ C_B^N(k-1)\left(a_B(k)-b_{aB}(k-1)+w_{accel}(k)\right) \\ q_\Theta(k-1)\left(\omega_B(k)-b_{gB}(k-1)+w_{gyros}(k)\right) \\ -\dfrac{1}{\tau_a} b_{aB}(k-1) \\ -\dfrac{1}{\tau_g} b_{gB}(k-1) \end{bmatrix} \\ &+ \begin{bmatrix} 0 \\ 0 \\ 0 \\ \sqrt{\dfrac{2f_s\sigma_{baB}^2}{\tau_a}} w_{accbias}(k) \\ \sqrt{\dfrac{2f_s\sigma_{bgB}^2}{\tau_g}} w_{gyrobias}(k) \end{bmatrix} \end{aligned} \tag{30}$$

and

$w(k)=[w_{accle}(k),w_{gyros}(k),w_{accbias}(k),w_{gyrobias}(k)]^T \in \mathbb{R}^{12}$ is the process noise vector.

With velocity constraints (27), the AHRS velocities in the B frame are considered as the measurement vector $y(k)=[v_{Bx}(k),v_{By}(k),v_{Bz}(k)]^T \in \mathbb{R}^3$. Including the wheel encoder measurement noise and ground topography, the measurement model is rewritten in discrete-time form as:

$$y(k)=h(X(k)+n(k))=\left(C_B^N\right)^T V_N + n(k) \tag{31}$$

where *n(k)* represents measurement noises. We assume that the measurement noises *n(k)* are independent, zero-mean and Gaussian white noise, i.e., *n(k)~N(0, R)*. The covariance matrix of the measurement noise is:

$$R(k)=\mathrm{diag}(\sigma_{vBx}^2,\sigma_{vBy}^2,\sigma_{vBz}^2) \in \mathbb{R}^{3\times 3} \tag{32}$$

where σ_{vBx}^2, σ_{vBy}^2, σ_{vBz}^2 are calculated in Equation (28). The *EKFx* was implemented by using the systems (29) and (31) in order to obtain the estimated positions, velocities and attitudes of the robot.

4.3 Slip estimation (KFλ) method

In order to determine wheel slip parameters to compensate for the velocity constraints and their variance values in Section 4.1.2, a discrete Kalman filter, denoted by *KFλ*, is designed. By giving v_{Bx} in the body frame B and yaw rate $\dot\psi$, the wheel centre velocities along the x-axis for the left and right side's wheels are calculated by

$$\begin{aligned} v_L &= v_{Bx} - \frac{L}{2}\dot\psi \\ v_R &= v_{Bx} + \frac{L}{2}\dot\psi \end{aligned} \tag{33}$$

By substituting (33) into (25), we have:

$$\begin{aligned} \lambda_L &= \frac{r\omega_L - v_{Bx} + \frac{L}{2}\dot\psi}{r\omega_L} \\ \lambda_R &= \frac{r\omega_R - v_{Bx} - \frac{L}{2}\dot\psi}{r\omega_R} \end{aligned} \tag{34}$$

$$\Rightarrow \begin{aligned} \omega_L &= \frac{1}{1-\lambda_L}\left[\frac{v_{Bx}}{r} - \frac{L\dot\psi}{2r}\right] \\ \omega_R &= \frac{1}{1-\lambda_R}\left[\frac{v_{Bx}}{r} + \frac{L\dot\psi}{2r}\right] \end{aligned} \tag{35}$$

We define the unknown parameters:

$$\gamma_L = \frac{1}{1-\lambda_L};\gamma_R = \frac{1}{1-\lambda_R},\ \ \lambda \in \left(-\infty,1\right] \Rightarrow \gamma \in \left[0,+\infty\right) \tag{36}$$

Substituting (36) into (35), the relationship between the angular speed from the encoders and unknown parameters is described as:

$$\begin{bmatrix} \omega_L \\ \omega_R \end{bmatrix} = \begin{bmatrix} \dfrac{v_{Bx}}{r} - \dfrac{L\dot\psi}{2r} & 0 \\ 0 & \dfrac{v_{Bx}}{r} + \dfrac{L\dot\psi}{2r} \end{bmatrix} \begin{bmatrix} \gamma_L \\ \gamma_R \end{bmatrix} \tag{37}$$

The slip estimation based on the discrete Kalman filer (KFλ) is now established. We assume that the slip process is a random process and then the unknown parameter process can be modelled in discrete-time form

$$\gamma(k) = \Phi_k \gamma(k-1) + w_\gamma(k-1) \tag{38}$$

where $\gamma(k)=[\gamma_L(k),\gamma_R(k)]^T$ is the process state vector at time k step, $w_\gamma(k)=[w_{\gamma 1}(k),w_{\gamma 2}(k)]^T$ is a process noise vector, and Φ_k is the state transition matrix which is defined as:

$$\Phi_k = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{39}$$

From Equation (37), a measurement model can be written which relates to the encoder measurement noises as:

$$z(k) = H_\gamma(k)\gamma(k) + n_\gamma(k) \tag{40}$$

where $z(k) = [\omega_L(k), \omega_R(k)]^T$ is the angular speed vector from two encoders at time step *k*, $n_\gamma(k) = [n_{\gamma 1}(k), n_{\gamma 2}(k)]^T$ is the measurement noise vector and $H_\gamma(k)$ is the measurement matrix, which is calculated as:

$$H_\gamma(k) = \begin{bmatrix} \left(\dfrac{\hat{v}_{Bx}(k)}{r} - \dfrac{L\hat{\dot{\psi}}(k)}{2r}\right) & 0 \\ 0 & \left(\dfrac{\hat{v}_{Bx}(k)}{r} + \dfrac{L\hat{\dot{\psi}}(k)}{2r}\right) \end{bmatrix} \tag{41}$$

where $\hat{v}_{Bx}(k), \hat{\dot{\psi}}(k)$ are the estimation values of the linear and angular velocities of mobile robot in the body frame, which are computed as:

$$\hat{v}_{Bx}(k) = \hat{V}_x(k)\cos(\hat{\psi}(k)) + \hat{V}_y(k)\sin(\hat{\psi}(k)) \tag{42}$$

$$\hat{\dot{\psi}}(k) = \frac{\hat{\psi}(k) - \hat{\psi}(k-1)}{\Delta T} \tag{43}$$

where $\hat{V}_x(k)$, $\hat{V}_y(k)$ and $\hat{\psi}(k)$ are given by the EKFx in Section 4.2. Systems (38) and (40) are used to perform the *KFλ* in order to estimate the wheel slip parameters of the left and right sides. From Equation (36), the estimated slip parameters at the *k* step are calculated using the estimated unknown parameters $\hat{\gamma}_L(k)$ and $\hat{\gamma}_R(k)$ as follows:

$$\begin{aligned} \hat{\lambda}_L(k) &= 1 - \frac{1}{\hat{\gamma}_L(k)} \\ \hat{\lambda}_R(k) &= 1 - \frac{1}{\hat{\gamma}_R(k)} \end{aligned} \tag{44}$$

Based on previous descriptions, the estimated slip values $\hat{\lambda}_L$ and $\hat{\lambda}_R$ are obtained indirectly using the unknown parameters $\hat{\gamma}_L$ and $\hat{\gamma}_R$. With this method, the current values $\hat{v}_{Bx}(k)$,$\hat{\dot{\psi}}(k)$ (in Equations (42) and (43)) are calculated by using the current state $\hat{X}(k)$ and $\hat{X}(k-1)$ of the EKFx in Section 4.2. Similarly, the current estimated slip values in Equation (44) are used to compute the longitudinal velocity $v_{Bx}(k)$ in (27) and its noise variance value in (28) in order to perform the EKFx. After these analyses, there are two Kalman filters running concurrently; this is called a *dual estimation algorithm* [18-19]. As illustrated in Figure 12, the top EKFx generates state estimates and requires $\hat{\lambda}(k-1)$ for the measurement update, the bottom KFλ generates slip state estimates and requires $\hat{X}(k-1)$ for the measurement update. Table 3 describes the dual estimation process using mathematical equations [20-21].

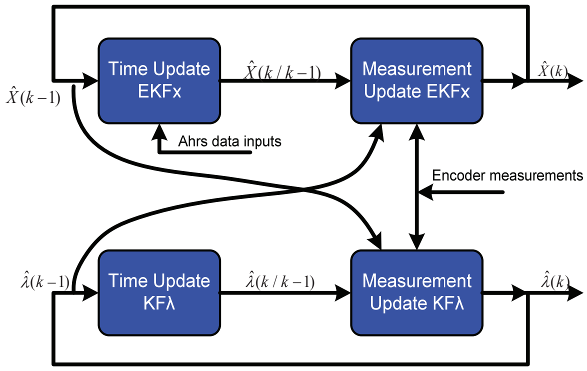

Figure 12. The dual estimation algorithm

○ *Initialize with:*

$$\hat{\gamma}(0) = E[\gamma(0)], P_\gamma(0) = E[(\gamma(0) - \hat{\gamma}(0))(\gamma(0) - \hat{\gamma}(0))^T]$$
$$\hat{X}(0) = E[X(0)], P(0) = E[(X(0) - \hat{X}(0))(X(0) - \hat{X}(0))^T]$$

○ *For* $k \in \{1, \ldots, \infty\}$, *the prediction step for KFλ is*

$$\hat{\gamma}(k/k-1) = \Phi_{k-1}\hat{\gamma}(k-1)$$
$$P_\gamma(k/k-1) = \Phi_{k-1}P_\gamma(k-1)\Phi_{k-1}^T + Q_\gamma$$

and the update step for KFλ is

$$H_\gamma(k) = \begin{bmatrix} \left(\dfrac{\hat{v}_{Bx}(k-1)}{r} - \dfrac{L\hat{\dot{\psi}}(k-1)}{2r}\right) & 0 \\ 0 & \left(\dfrac{\hat{v}_{Bx}(k-1)}{r} + \dfrac{L\hat{\dot{\psi}}(k-1)}{2r}\right) \end{bmatrix}$$

$$K_\gamma(k) = P_\gamma(k/k-1)H_\gamma^T(k)\left[H_\gamma(k)P_\gamma(k/k-1)H_\gamma^T(k) + R_\gamma(k)\right]^{-1}$$

$$\hat{\gamma}(k) = \hat{\gamma}(k/k-1) + K_\gamma(k)[z(k) - H_\gamma(k)\hat{\gamma}(k/k-1)]$$

$$P_\gamma(k) = \left[I - K_\gamma(k)H_\gamma(k)\right]P_\gamma(k/k-1)$$

$$\hat{\lambda}_L(k) = 1 - \frac{1}{\hat{\gamma}_L(k)}, \quad \hat{\lambda}_R(k) = 1 - \frac{1}{\hat{\gamma}_R(k)}$$

where Q_γ and $R_\gamma(k)$ are the process and measurement covariance matrices of $w_\gamma(k)$ and $n_\gamma(k)$, respectively.

○ *The prediction step for EKFx is*

$$\hat{X}(k/k-1) = \hat{X}(k-1) + \Delta T g(\hat{X}(k-1), u(k), 0)$$

$$P(k/k-1) = F_X(k)P(k-1)F_X^T(k) + W(k)QW^T(k)$$

and the update step for EKFx is

$$v_{Bx}(k)=\frac{r}{2}\left[(1-\hat{\lambda}_L)\omega_L(k)+(1-\hat{\lambda}_R)\omega_R(k)\right]$$

$$\sigma^2_{vBx}=\frac{r^2}{4}\left((1-\hat{\lambda}_R(k))^2\,\mathrm{var}\left(\omega_R\right)+(1-\hat{\lambda}_L(k))^2\,\mathrm{var}\left(\omega_L\right)\right)$$

$$y(k)=[v_{Bx}(k),v_{By}(k),v_{Bz}(k)]^T\in\mathbb{R}^3$$

$$R(k)=\mathrm{diag}(\sigma^2_{vBx},\sigma^2_{vBy},\sigma^2_{vBz})\in\mathbb{R}^{3\times3}$$

$$S(k)=H_X(k)P(k/k-1)H_X^T(k)+R(k)$$

$$K(k)=P(k/k-1)H_X^T(k)S^{-1}(k)$$

$$\hat{X}(k)=\hat{X}(k/k-1)+K(k)[y(k)-h(\hat{X}(k/k-1),0)]$$

$$P(k)=P(k/k-1)-K(k)H_X(k)P(k/k-1)$$

where

$$F_X(k)=\left.\frac{\partial f(X,u(k),0)}{\partial X}\right|_{\hat{X}(k-1)},\ W\left(k\right)=\left.\frac{\partial f}{\partial w}\right|_{\left(\hat{X}(k-1),u(k)\right)},$$

$$H_X(k)=\left.\frac{\partial h(X,0)}{\partial X}\right|_{\hat{X}(k/k-1)},$$

Q is the process covariance and the detailed calculations of $F_X(k)$, $W(k)$ and $H_X(k)$ are described in the Appendix.

- *Updating estimation values* $\hat{v}_{Bx}(k),\hat{\dot{\psi}}(k)$ *for KFλ as*

$$\hat{v}_{Bx}(k)=\hat{V}_x(k)\cos(\hat{\psi}(k))+\hat{V}_y(k)\sin(\hat{\psi}(k))$$

$$\hat{\dot{\psi}}(k)=\frac{\hat{\psi}(k)-\hat{\psi}(k-1)}{\Delta T}$$

Table 3. The dual estimation equations

5. Experimental results

In this section, experiments were performed to prove the effectiveness of the combination of the proposed FVFO and slip compensation methods for the robot's position in a real-time application. The system was performed on concrete terrain in an outdoor environment. The experiments involve two scenarios:

- *Scenario 1*: The mobile robot is controlled along a desired trajectory with two curves in two cases: without and with slip compensation.
- *Scenario 2*: The mobile robot is controlled along a desired circular trajectory with a 2-m radius in two cases: without and with slip compensation.

5.1 The use of the FVFO method for mobile robot without slip compensation

In this section, the proposed FVFO method is performed in two scenarios without slip compensation. Additionally, the stable tracking control method (STC) and the conventional VFO feedback control were applied in order to compare them with the FVFO results.

In the first scenario, the output trajectories of the three methods are shown in Figure 13 are and compared to the desired trajectory. The response of STC is quite different from the desired response, while those of VFO and FVFO are close to the desired response. The linear and angular velocities of the three methods, shown in Figures 14 and 15, are also close to the desired velocities despite the mechanical vibrations and sensor noises.

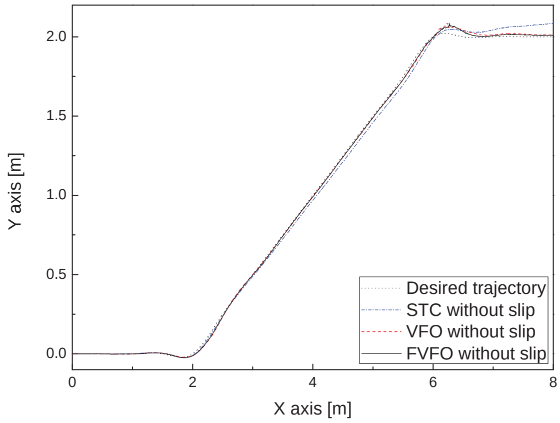

Figure 13. The desired, STC, VFO and FVFO trajectories, respectively, without slip compensation in scenario 1

In Figure 16, the position error of FVFO, which is the distance from the estimated position of the robot at the defined instant to the nearest point of the desired trajectory, at any instance in time is the closest to zero compared to those of the STC and conventional VFO. The results suggest that the proposed FVFO has improved the performance of the solving of the path tracking problem. Consequently, the Root Mean Square Error (RMSE) of the robot's position of FVFO is the smallest (0.0494[m]), while those of the STC and VFO are 0.0994[m] and 0.0845[m], respectively.

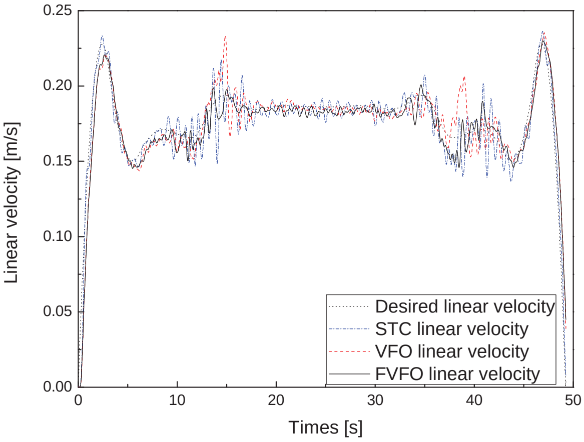

Figure 14. The desired, STC, VFO and FVFO linear velocities, respectively, without slip compensation in scenario 1

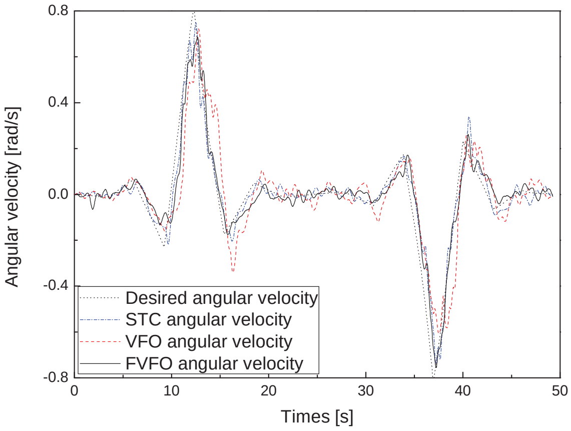

Figure 15. The desired, STC, VFO, FVFO angular velocities, respectively, without slip compensation in scenario 1

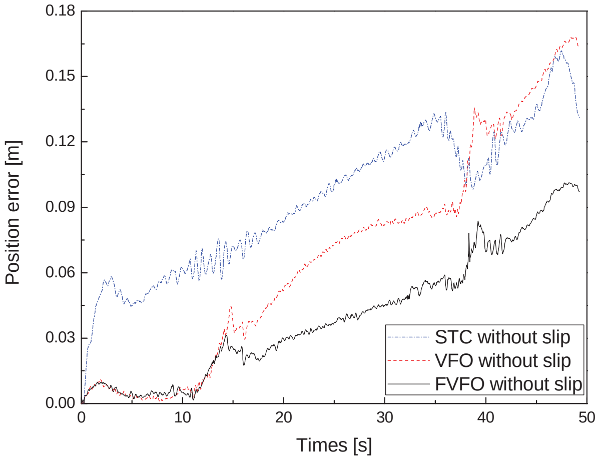

Figure 16. The position errors of STC, VFO and FVFO, respectively, without slip compensation in scenario 1

The trajectory responses of the three methods in the second scenario are shown in Figure 17. The linear angular velocities and position errors of the three methods are shown in Figures 18, 19 and 20. It is clear that the FVFO result is the best (with the smallest RMSE (0.0688 [m])).

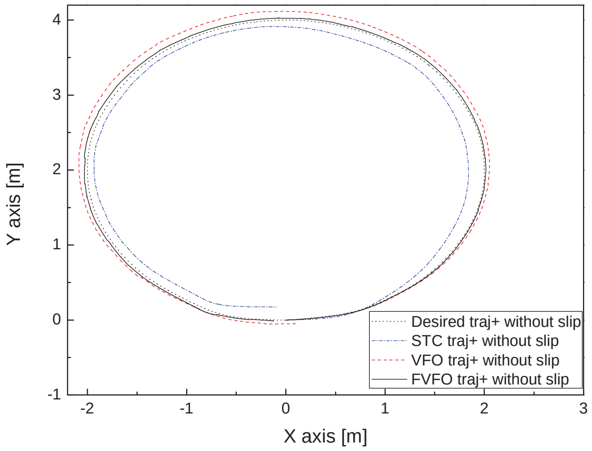

Figure 17. The desired, STC, VFO and FVFO trajectories, respectively, without slip compensation in scenario 2

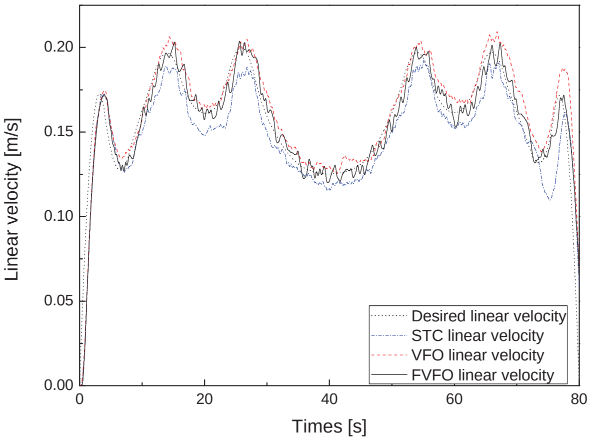

Figure 18. The desired, STC, VFO and FVFO linear velocities, respectively, without slip compensation in scenario 2

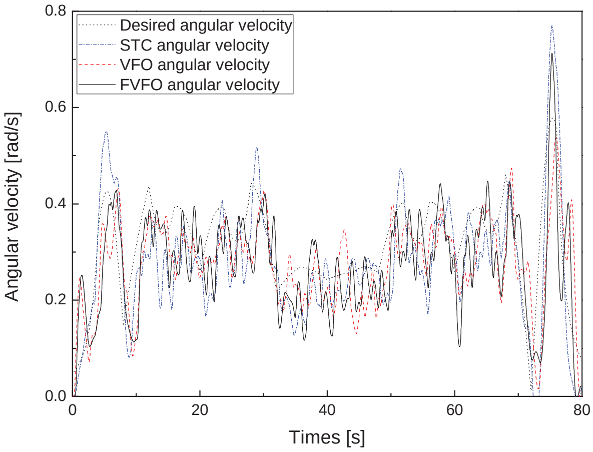

Figure 19. The desired, STC, VFO, FVFO angular velocities, respectively, without slip compensation in scenario 1

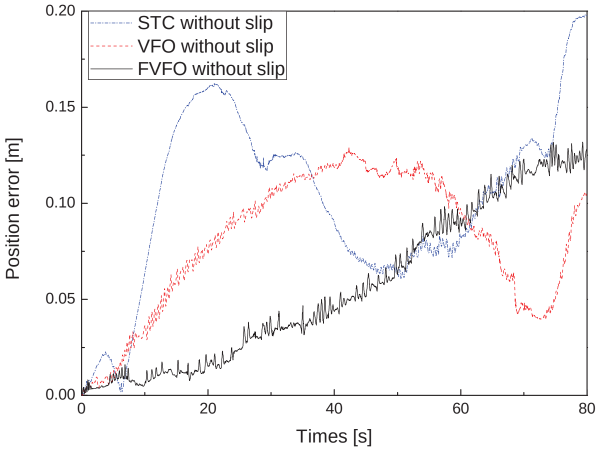

Figure 20. The position errors of STC, VFO and FVFO, respectively, without slip compensation in scenario 2

5.2 The use of the FVFO method for a mobile robot with slip compensation

In Section 5.1, although the FVFO method reduced more position errors than the STC and VFO method, the system error is still significant. In the field of mobile robotics, wheel slippage limits the traction and braking ability of the robot, and causes large position errors in path tracking control. In this section, the system is performed again with slip compensation.

The STC, VFO and FVFO trajectories with slip compensation in both scenarios are shown in Figures 21 and 24. The FVFO responses with slip compensation are closest to the desired responses. Based on the slip estimation, the estimated values from EKFx are more precise than the previous values without slip compensation. By considering position errors, it is clear that the performance results with slip compensation are more accurate than previous results without slip, as shown in Figures 22 and 25. Consequently, the smallest RMSEs of the proposed method are 0.0124[m] and 0.0121[m] corresponding to the first and second scenarios.

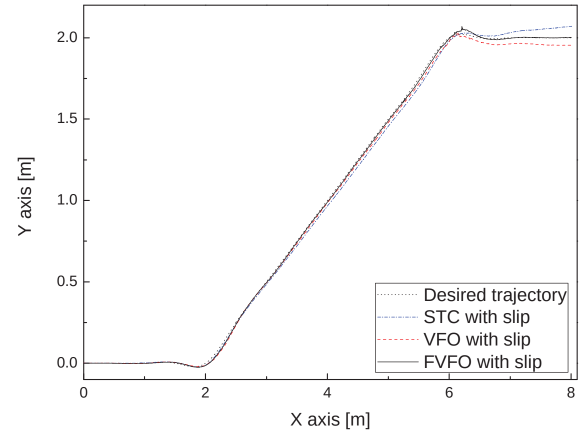

Figure 21. The desired, STC, VFO and FVFO trajectories, respectively, with slip compensation in scenario 1

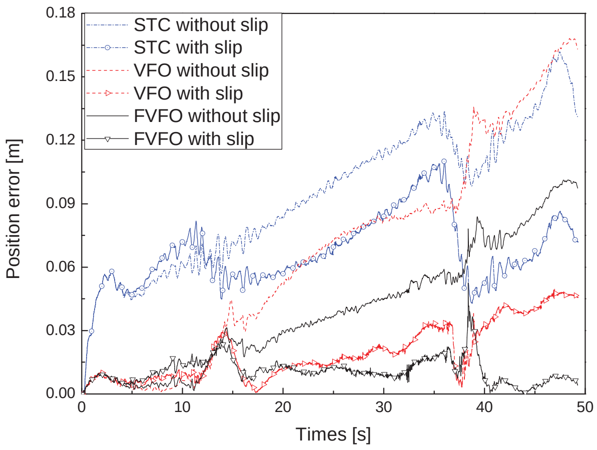

Figure 22. The position errors of STC, VFO and FVFO, respectively, without and with slip compensation in scenario 1

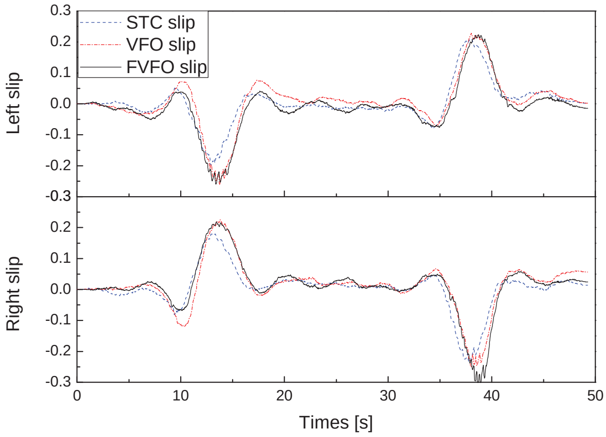

Figure 23. The estimated slip parameters of the left and right wheels of the STC, VFO and FVFO methods, respectively, in scenario 1

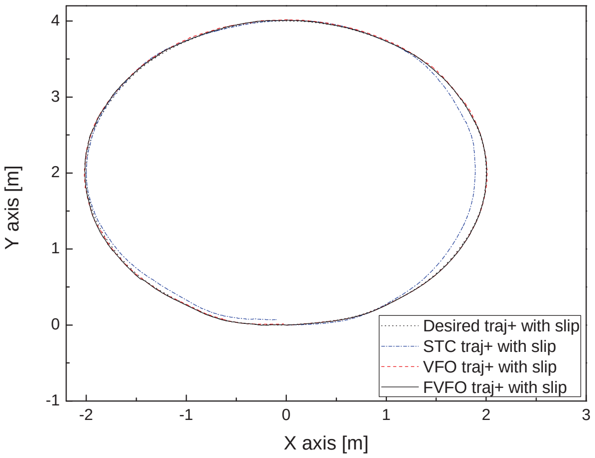

Figure 24. The desired, STC, VFO and FVFO trajectories, respectively, with slip compensation in scenario 2

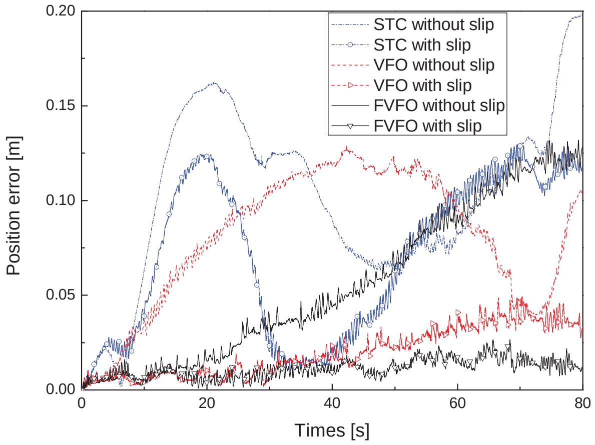

Figure 25. The position errors of STC, VFO and FVFO, respectively, without and with slip compensation in scenario 2

It is clear from the above results that slip compensation is very effective in the field of mobile robotics. For the first scenario, the robot follows the desired trajectory, which has two curves. For the first curve, the angular speed of the right wheels is greater than that of the left wheels. Then, the traction phenomenon occurs on the right side while the braking phenomenon appears on the left. Consequently, the right wheel slip value is positive and the left wheel slip value is negative. Similarly, for the right rotation in the second curve, the right wheel slip value is negative and the left wheel slip value is positive. Furthermore, when the robot travels along the straight segment, less force is needed to drive the robot and the left and right wheel slips are close to zero. Figure 23 illustrates the estimated slip parameters of the left and right wheels for the three methods.

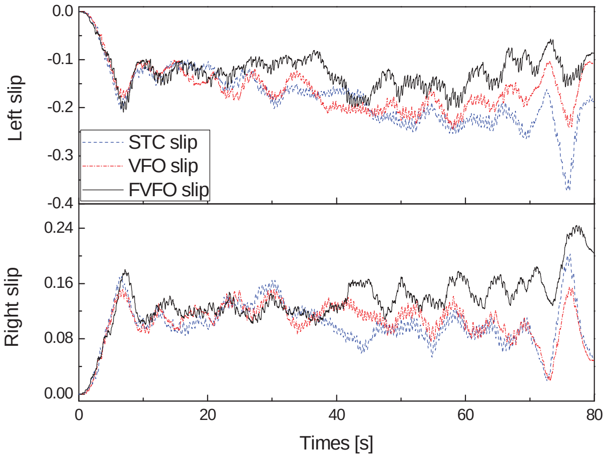

Figure 26. The estimated slip parameters of the left and right wheels of the STC, VFO and FVFO methods, respectively, in scenario 2

For the second scenario, the robot follows the desired circular trajectory counterclockwise. The angular speed of the right wheels is always larger than that of the left wheels. Consequently, the right wheel slip value is always positive and the left wheel slip value is always negative, as shown in Figure 26. All types of the estimated slip parameters in both scenarios are consistent. Tables 4 and 5 show RMSEs of all the experiments in this work.

RMSE [m]	STC	VFO	FVFO
Without slip	0.0994	0.0845	0.0494
With slip	0.0688	0.0239	0.0124

Table 4. The Root Mean Square Errors (RMSEs) of the robot position in scenario 1

RMSE [m]	STC	VFO	FVFO
Without slip	0.1116	0.0882	0.0688
With slip	0.0818	0.0236	0.0121

Table 5. The Root Mean Square Errors (RMSEs) of the robot position in scenario 2

6. Conclusions

This paper presents experiments into a method of controlling the 4-TW SSMR along a desired trajectory based on estimated feedback control signals from an extended Kalman Filter, which uses measurements from an AHRS and two incremental encoders. In this paper, we proposed a combination of the proposed FVFO and novel wheel slip compensation methods for the trajectory tracking of a mobile robot. The extended Kalman filter is utilized to estimate the positions, velocities and orientation angles, which are used for feedback control signals in the FVFO method, based on an AHRS kinematic motion model and velocity constraints. Experimental results show the advantages of the combination of these two methods, which are effective at overcoming the limitations of the others in the trajectory tracking control problem for a 4-TW SSMR, e.g. the commercialized Hazard Escape I® mobile robot.

7. Acknowledgments

This work was supported by the 2011 Research Fund of University of Ulsan.

8. Appendix

$$F_X(k) = \left.\frac{\partial f(X,u(k),0)}{\partial X}\right|_{\hat{X}(k-1)}$$

$$= \begin{bmatrix} I_3 & I_3\Delta T & 0_3 & 0_3 & 0_3 \\ 0_3 & I_3 & G_V(k)\Delta T & -C_B^N\Delta T & 0_3 \\ 0_3 & 0_3 & I_3 + G_\Theta(k)\Delta T & 0_3 & -q_\Theta\Delta T \\ 0_3 & 0_3 & 0_3 & I_3(1-\Delta T/\tau_a) & 0_3 \\ 0_3 & 0_3 & 0_3 & 0_3 & I_3(1-\Delta T/\tau_g) \end{bmatrix}$$

$$G_V = \frac{\partial g_V}{\partial \Theta} = \begin{bmatrix} \frac{\partial C_B^N}{\partial \phi} a_B & \frac{\partial C_B^N}{\partial \theta} a_B & \frac{\partial C_B^N}{\partial \psi} a_B \end{bmatrix}$$

$$G_\Theta = \frac{\partial g_\Theta}{\partial \Theta} = \begin{bmatrix} \dot{\theta}\tan\theta & \frac{\dot{\psi}}{c_\theta} & 0 \\ -\dot{\psi}c_\theta & 0 & 0 \\ \frac{\dot{\theta}}{c_\theta} & \dot{\psi}\tan\theta & 0 \end{bmatrix}$$

$$W(k) = \left.\frac{\partial f}{\partial w}\right|_{\left(\hat{X}(k-1),u(k)\right)} = \begin{bmatrix} O_3 & O_3 & O_3 & O_3 \\ C_B^N(k) & O_3 & O_3 & O_3 \\ O_3 & q_\Theta(k) & O_3 & O_3 \\ O_3 & O_3 & I_3 & O_3 \\ O_3 & O_3 & O_3 & I_3 \end{bmatrix}$$

$$H_X(k) = \left.\frac{\partial h(X,0)}{\partial X}\right|_{\hat{X}(k/k-1)} = \begin{bmatrix} 0_3 & H_V(k) & H_\Theta(k) & 0_3 & 0_3 \end{bmatrix}$$

$$H_V(k) = \frac{\partial h(v_B)}{\partial V_N} = \left(C_B^N(k)\right)^T$$

$$H_\Theta = \frac{\partial h(v_B)}{\partial \Theta} = \frac{\partial}{\partial \Theta}\left(\left(C_B^N\right)^T V_N\right)$$

9. References

[1] Fierro, R. and Lewis, F.L. (1997) Control of a Nonholonomic Mobile Robot: Backstepping Kinematics into Dynamics. Journal of Robotic Systems: 14, 149-163.

[2] Yang, F. and Wang, C. (2012) Adaptive tracking control for uncertain dynamic nonholonomic mobile robots based on visual servoing. Journal of Control Theory and Applications: 10, 56-63.

[3] Kanayama, Y., Kimura, Y., Miyazaki, F. and Noguchi, T. (1990) A stable tracking control method for an autonomous mobile robot. Proceedings of the IEEE International Conference on Robotics and Automation: 1, 384-389.

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

[5] Chwa, D. (2012) Fuzzy Adaptive Tracking Control of Wheeled Mobile Robots With State-Dependent Kinematic and Dynamic Disturbances. IEEE Transactions on Fuzzy Systems: 20, 587 - 593.

[6] Yi, J., Song, D., Zhang, J. and Goodwin, Z. (2007) Adaptive Trajectory Tracking Control of Skid-Steered Mobile Robots. IEEE International Conference on Robotics and Automation: 2605 - 2610.

[7] Michalek, M. and Kozlowski, K. (2010) Vector-Field-Orientation Feedback Control Method for a Differentially Driven Vehicle. IEEE Transactions on Control Systems Technology: 18, 45-65.

[8] Michalek, M. and Kozlowski, K. (2005) Trajectory Tracking for a Threecycle Mobile Robot: The Vector Field Orientation Approach. IEEE European Control Conference on Decision and Control: 1119 - 1124.

[9] Arslan, S. and Temeltas, H. (2011) Robust Motion Control of a Four Wheel Drive Skid-Steered Mobile Robot. International Conference on Electrical and Electronics Engineering: II-415 - II-419.

[10] Michalek, M.M., Dutkiewicz, P., Kielczewski, M. and Pazderski, D. (2010) Vector-Field-Orientation Tracking Control for a Mobile Vehicle Disturbed by the Skid-Slip Phenomena. Journal of Intelligent and Robotic Systems: 59, 341-365.

[11] Song, Z., Zweiri, Y.H., Seneviratne, L.D. and Althoefer, K. (2006) Non-linear Observer for slip estimation of skid-steering vehicle. International Conference on Robotics and Automation: 1499-1504.

[12] Song, X., Seneviratne, L.D., Althoefer, K. and Song, Z. (2008) A robust slip estimation method for skid-steered mobile robots. International Conference on Control, Automation, Robotics and Vision: 279 - 284.

[13] Moosavian, S.A.A. and Kalantari, A. (2008) Experimental slip estimation for exact kinematics modeling and control of a Tracked Mobile Robot. International Conference on Intelligent Robots and Systems: 22-26.

[14] Ahn, K.K., Truong, D.Q., Thanh, T.Q. and Lee, B.R. (2008). Online self-tuning fuzzy proportional-integral-derivative control for hydraulic load simulator. Proceedings of the Institution of Mechanical Engineers. Part I: Journal of Systems and Control Engineering: 222, 81-95.

[15] Flenniken IV, W.S., Wall, J.H. and Bevly, D.M. (2005) Characterization of Various IMU Error Sources and the Effect on Navigation Performance. Proceedings of the International Technical Meeting of the Satellite Division of The Institute of Navigation: 967-978.

[16]Yi, J., Wang, H., Zhang, J., Song, D., Jayasuriya, S. and Liu, J. (2009) Kinematic Modeling and Analysis of Skid-Steered Mobile Robots With Applications to Low-Cost Inertial-Measurement-Unit-Based Motion estimation. IEEE Transactions on Robotics: 25, 1087-1097.

[17] Teslic, T., Klancar, G. and Skrjanc, I. (2008) Kalman-filtering-based localization of mobile robot with a LRF in a simulated 2D environment. IEEE Mediterranean Electrotechnical Conference: 316-322.

[18] Wan, E.A., Merwe, R.V.D. and Nelson, A.T. (1999) Dual Estimation and the Unscented Transformation. Neural Information Processing Systems: 666-672.

[19] Haykin, S. (2001) Kalman Filtering and Neural Networks. John Wiley & Sons, Inc.

[20] Welch, G. and Bishop, G. (1995) An Introduction to the Kalman Filter. University of North Carolina at Chapel Hill, Department of Computer Science

[21] Brown, R.G. and Hwang, P.Y.C. (1996) Introduction to Random Signals and Applied Kalman Filtering, Third Edition. 496p.