



Motion Tracking of Four-Wheeled Mobile Robots in Outdoor Environments Using Bayes' Filters

Deok-Kee Choi¹

Received: 28 November 2021 / Revised: 10 November 2022 / Accepted: 8 February 2023 / Published online: 22 February 2023
© The Author(s), under exclusive licence to Korean Society for Precision Engineering 2023

Abstract

The demand for outdoor wheeled mobile robots (WMRs) is rapidly growing to assist humans in outdoor environments such as transportation, exploration, rescue, security, agriculture, military, etc. To effectively control outdoor WMRs, we need motion models applicable in such unknown environments. Conventional modeling mainly concerns physics laws or equations of motion, such as kinematics, dynamics, terrains, and wheel-ground interactions. Modeling WMRs on unstructured ground is more complicated than in a well-developed indoor environment. To alleviate such difficulties, we looked at a data-driven approach instead. We built a four-wheeled mobile robot with wheel encoders installed, with which the forward and inverse differential kinematic solutions were derived. Then, we performed more than a thousand test runs in outdoor environments, having the robot run on normal, icy, and sandy roads, including test runs under constraints partially blocked by brick and grid-type holes. We employed Bayes' filter because the robot's tri-variate states (two linear velocities and the rotation) are not directly measurable through wheel encoders. With such uncertainty, Bayes' filtering technique of the Kalman filter and a newly developed unscented Kalman filter were applied to infer how each wheel's speed affects the robot's velocity. We established a probabilistic motion model, where the differential kinematic solutions are combined with uncertainty from outdoor environments. Consequently, we could closely track the robot's motion. This modeling technique can be used to develop better outdoor WMRs.

Keywords Bayes' filter · Unscented Kalman filter · Kalman filter · Four-wheeled mobile robot · Data-driven model

1 Introduction

Mobile robots are becoming increasingly popular across different industrial areas. Recently, a mobile robot has been stated as a machine manipulated by artificial intelligence that uses sensors and other technology to identify its surroundings and move around its environment. Mobile robots have physical robotic elements, such as wheels, tracks, legs, etc. Among the different locomotion devices of mobile robots, wheeled mobile robots (WMRs) have drawn more attention because of structural simplicity than other devices [1]. The demand for outdoor WMRs is rapidly growing to assist humans. For example, autonomous wheeled mobile robots

or vehicles are being planned for many applications such as transportation, exploration, rescue, security, military, etc.

Even so, a robot equipped with various sensors is likely to operate differently due to inexplicable factors in outdoor environments. For example, the wheel encoder might lose counting steps upon wheel slippage; the radar may work unsatisfactorily under dense fog or heavy rainy days. An ultrasonic sensor may not detect bounced signals due to bad weather. A camera only works properly in a well-lit environment. A global positioning system (GPS) can quickly lose signals among high-rise buildings or dense forests. Such harsh outdoor environments have demanded an advanced intelligent system to control mobile robots. Conventional WMRs have been developed in indoor environments such as warehouses, factories, etc. However, to control WMRs in outdoor environments, we need somewhat different motion models than indoor environments because of various uncertainty. Thus, the realization of the motion of WMRs is solely vital, and we need to understand how a robot moves in various outdoor

✉ Deok-Kee Choi
dkchoi@dankook.ac.kr

¹ Department of Mechanical Engineering, Dankook University, 152 Jukjeon-ro, Suji-gu, Yongin-si, Gyeonggi-do 16890, Republic of Korea

environments. Even so, tracking the motion of WMRs in an outdoor environment is not a trivial problem. A typical robot tracking would be computed from a distance traveled given a velocity or acceleration. However, no mobile robot runs on a perfect road. There are different kinds of obstacles that raise and lower the speed, even with slippage. Unlike well-established indoor environments, we have not fully understood how WMRs respond to various outdoor environments.

The conventional analysis mainly concerns several categories of physics laws or differential equations on the movement of robots. The categories of the analysis are composed of kinematics (rolling, yawing, pitching), dynamics (steering, skidding, braking), terrains (uphill, tilted roads), and wheel-ground interactions (ice, rocky, sand). Therefore, most researchers have focused on building detailed physics models concerning each category. This requires a vast amount of research to come up with each of the models. Even for simple robots, the dynamic models are usually very complex, and they would not consider friction between wheels and various ground, wheel slippage, etc. The wheel-ground interaction depends on many parameters, which require prior knowledge of the ground parameters (friction coefficient, wheel slippage, sand friction, etc.). Furthermore, lateral slippage usually may not be observed; thus, a more detailed study is required [2]. Therefore, modeling a WMR moving outdoors is far more complex than in indoor environments because of the interaction between the wheels and the ground.

In this study, we built a four-wheeled mobile robot with encoders installed on each wheel. Tracking a four-wheeled robot is quite involved because each wheel rotates independently and is subjected to different road, traction, and load conditions. Ideally, we need four models for each wheel, which is impractical. Instead of separately establishing physics laws for road conditions, we employed the data-driven approach. In so doing, we may analyze dynamic systems without full knowledge of physics models. All models are inherently not accurate because they come with uncertainty. Thus, models can only partially explain the robot's underlying phenomena and its environment. Therefore, the probabilistic approach can play an essential role in recognizing uncertainty. Probabilistic algorithms present results in the form of probability distributions. The Bayesian paradigm has been known to fit well with such problems. The filtering technique has played a vital role successfully in many disciplines for temporal data. We employed two variants of Bayes' filter, the Kalman filter and the unscented Kalman filter, to estimate the robot velocity given wheel speed measurement in an outdoor environment.

In summary,

1. We derived forward and inverse differential kinematic solutions for four-wheeled mobile robots with the fixed-standard wheel system.
2. We carried out more than a thousand test runs in outdoor environments, having the robot run on normal, icy, and sandy roads under a constraint, partially blocked by brick and stuck with grid-type holes. As a result, we obtained wheel speed measurements on different road conditions.
3. We applied the Kalman filter and the unscented Kalman filter, which stemmed from Bayes' filtering technique, to infer how each wheel's speed affects the robot's velocity in different road conditions.
4. We could closely track tri-variate states depicting the robot's motion: two linear velocities and rotation. We demonstrated how to relate the differential kinematic solutions to a probabilistic model with uncertainty in outdoor environments.
5. We adopted five outdoor environments for estimating the robot's motion for demonstration purposes; however, we think the present method can be applied to any unknown outdoor environment without significant modifications.

This paper is organized as follows: A brief review of the current research is mentioned in Related work; the wheel equation for a four-wheeled mobile robot with the fixed-standard wheel system is derived in Kinematics, a more general version of the wheel equation is also developed. Bayesian formalism applicable to this study is explained in detail in Bayes' filter along with the Kalman filter and the unscented Kalman filter. An experimental setup for the robot is described in the Four-wheeled mobile robot. Upon more than a thousand test runs, the estimated robot velocity is listed in Results and Discussion. The brief summary is expressed with further study in Conclusion.

2 Related Work

In general, studies on wheeled mobile robots (WMRs) have been well established [3, 4]. With mainly kinematics [5] and dynamics [6] combined, the models can be utilized for control [7] and design of WMRs equipped with a general or standard type of wheels. Obviously, unlike kinematics models, the dynamics model would be pretty complex to define the motion of WMRs in general because of not a few physics laws and empirical rules to be parameterized [8]. Those parameters should be valued through experimental data, which demands vast experimentation. Furthermore, dynamics models are likely to be more sensitive to external

factors; for example, even a slight offset of the center of mass of a robot by an unexpected external force might fail to meet the robot's design criterion.

The most popular WMRs may be robots with two-differential drive systems [9]. The differential drive is a two-wheeled system with independent motors for each wheel. Thus, two-wheeled mobile robots can make many different movements, even spinning themselves on site simply by differing wheel speeds. Conventional indoor mobile robots typically operate on plain ground where wheel slip is negligible. On the contrary, mobile robots more often experience low traction in outdoor environments: unknown environments [10], skidding and slipping [11], wheel slippage [12], and yawing [13, 14]. It is quite challenging to catch the effects of outdoor environments on the robot's motion [15, 16] that mapping of the outdoor environment becomes quite important [17]. For example, it may not be possible to exposit the exact movement when robots operate on ice or a slippery surface. There is no sure way for physics laws or equations to explain such an ambiguous situation. In addition, the control of WMRs in unpredictable environments calls for advanced techniques such as Fuzzy control [18, 19]. Therefore, Outdoor robots bring about additional challenges not considered in indoor environments. Therefore, WMRs require dynamic and kinematic models, not to mention terrain. For example, acknowledging the unexpected environment in the autonomous drive is more critical because a robot should run without any human operators' intervention [20]. Even we might need some degree of terrain classification using machine learning techniques [21].

In this study, we set up a four-wheeled mobile robot with a differential drive system. A study on four-wheeled mobile robots with different wheel systems for trajectory tracking [22], calibration of systematic errors [23], and design and control [24] was published. We derived forward and inverse differential kinematic solutions for four-wheeled mobile robots with the fixed-standard wheel system. We need more information on mobile robots' movement in outdoor environments. In order to solve robot motion problems relating to outdoor environments, where the conventional approaches have much difficulty, we chose a probabilistic paradigm and experiment data. As for the experiment, we carried out about a thousand test runs in outdoor environments, having the robot run on normal, icy, and sandy roads. In addition, we performed tests on the robot under a constraint partially blocked by brick and being stuck with grid-type holes. As a result, we obtained wheel speed measurements on different road conditions. In general, the motion of a robot can be best described in a set of equations or differential equations. Nevertheless, when we are concerned with a temporal process such as robot motion, employing the state-space process may be more convenient, where the process can be calculated recursively over time. Analytically, perfectly modeling a system is impossible except for the most trivial problems.

The probabilistic approach of the Bayesian framework with state-space models best fits the filtering technique, which can formulate an entire prediction-update-estimation process recursively [25, 26]. The Bayes filter framework comprises an initial distribution (prior), a motion model distribution, and a measurement model distribution (likelihood). The reduced posterior is equivalent to the updated state variables in the corresponding state-space model. In particular, probabilistic approaches are typically more robust in the face of sensor flaws, noise, environment dynamics, etc. There are a few variants of Bayes' filtering techniques we can apply to infer how each wheel's speed affects the robot velocity on different road conditions: the Kalman filter [27], the unscented Kalman filter [28], and the extended Kalman filter [29], and so on. In this study, we employed the Kalman filter and the unscented Kalman filter for comparison. We demonstrated how to relate the differential kinematic solutions to a probabilistic model with uncertainty for better design considering outdoor environments. Hence, with the help of Bayes' filtering technique, we could estimate the motion of a four-wheeled mobile robot at each time step given measured wheel speeds.

3 Kinematics

This section will pursue a solution to wheel equations for a four-wheeled mobile robot in planar environments, subject to two kinematic constraints and three state variables (two linear velocities and rotation) that characterize the robot's pose in motion.

3.1 Wheel Equations

A schematic of a mobile robot with fixed standard wheels of four is shown in Fig. 1, in which we employ four frames, the inertial, robot body, steering, and wheel frames, to define the position and the velocity of the robot [30]. The position of a wheel in the inertial frame can be expressed as:

$${}^I\mathbf{P} = {}^I\mathbf{P}_R + {}^R\mathbf{P}_S + {}^S\mathbf{P}_W \quad (1)$$

where the coordinate frames I for Inertial, R for Robot's body, S for steering, and W for wheels. Thus, ${}^I\mathbf{P}_R$ denotes the position of the robot, ${}^R\mathbf{P}_S$ is the steering offset and ${}^S\mathbf{P}_W$ for the wheel offset. By taking the time derivative of Eq. (1), the velocity vectors are

$${}^I\mathbf{V}_W = {}^I\mathbf{V}_R + {}^I\boldsymbol{\Omega}_R \times {}^R\mathbf{P}_S + {}^I\boldsymbol{\Omega}_R \times {}^S\mathbf{P}_W + {}^R\boldsymbol{\Omega}_S \times {}^S\mathbf{P}_W \quad (2)$$

where ${}^I\mathbf{V}_R$ is the robot velocity and ${}^I\boldsymbol{\Omega}_R$ is the robot angular velocity. It is noted that the position vector ${}^R\mathbf{P}_S$ is fixed for the standard wheel system at frame $\{R\}$ that ${}^R\mathbf{V}_S = {}^R\dot{\mathbf{P}}_S = 0$ upon considering standard wheels. Frame $\{S\}$ and $\{W\}$ are

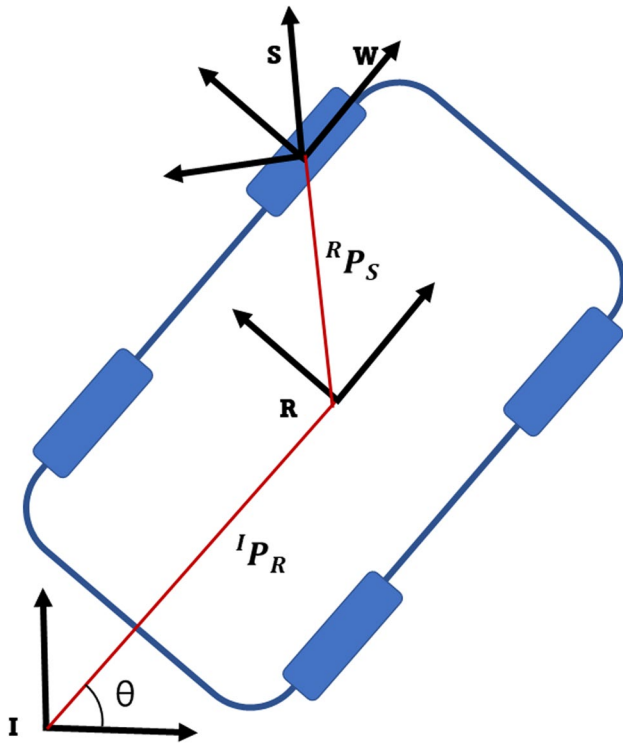


Fig. 1 Four coordinate frames for the fixed standard four-wheeled robot pose. I for the inertial frame, R for the robot body frame, S for the steering frame, and W for the wheel frame. ${}^I\mathbf{P}_R$ is the position vector from I to R , and ${}^R\mathbf{P}_S$ is for from R to S , respectively. The robot body frame makes an angle of θ concerning the inertial frame

concentric that ${}^S\mathbf{V}_W = {}^S\dot{\mathbf{P}}_W = 0$. Moreover, there is no wheel offset that ${}^S\mathbf{P}_W = 0$. Thus, the velocity equation of Eq. (2) can be written

$${}^I\mathbf{V}_W = {}^I\mathbf{V}_R + {}^I\boldsymbol{\Omega}_R \times {}^R\mathbf{P}_S \tag{3}$$

In order to apply the kinematic constraints residing in the wheel plane, the velocity equation of Eq. (3) needs to be projected on the wheel plane to yield

$${}^W({}^I\mathbf{V}_W) = {}^W({}^I\mathbf{V}_R) + {}^W({}^I\boldsymbol{\Omega}_R \times {}^R\mathbf{P}_S) \tag{4}$$

The projected components of LHS and RHS of Eq. (4) in wheel frame $\{W\}$ are:

LHS of Eq. (4):

$${}^W({}^I\mathbf{V}_W) = {}^W R^I \mathbf{V}_W \tag{5}$$

RHS of Eq. (4):

$$\begin{aligned} {}^W({}^I\mathbf{V}_R) + {}^W({}^I\boldsymbol{\Omega}_R \times {}^R\mathbf{P}_S) &= {}^W R^I \mathbf{V}_R + {}^W({}^I\boldsymbol{\Omega}_R \times {}^R\mathbf{P}_S) \\ &= {}^W R^S R^R R^I \mathbf{V}_R + {}^W \boldsymbol{\Omega}_R \times {}^W R^R \mathbf{P}_S \\ &= {}^W R^S R^R R^I \mathbf{V}_R + {}^W \boldsymbol{\Omega}_R \times {}^W R^R \mathbf{P}_W \end{aligned} \tag{6}$$

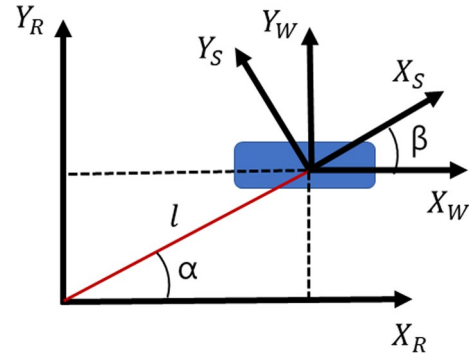


Fig. 2 A fixed standard wheel and the coordinate systems. R for the robot body frame, S for the steering frame, and W for the wheel frame. α is the steering angle, and β denotes the wheel angle

where ${}^W R_I$, ${}^W R_S$, ${}^S R_R$, and ${}^R R_I$ denote to a rotation in the frame referring to the subscript with respect to the frame indicating the superscript, respectively. Two frames of $\{S\}$ and $\{W\}$ are concentric that ${}^R\mathbf{P}_S = {}^R\mathbf{P}_W$ as shown in Fig. 2. Substituting Eqs. (5) and (6) into Eq. (4) to get

$$\underbrace{{}^W R^I \mathbf{V}_W}_{\text{Wheel Velocity}} = \underbrace{{}^W R^S R^R R^I \mathbf{V}_R}_{\text{Robot Velocity}} + \underbrace{{}^W \boldsymbol{\Omega}_R \times {}^W R^R \mathbf{P}_W}_{\text{Robot Rotation}} \tag{7}$$

We projected the robot velocity ${}^I\mathbf{V}_R$ of Eq. (7) into the wheel frame $\{W\}$ as shown in Fig. 2 to yield

$${}^W R^S R^R R^I \mathbf{V}_R = {}^W R^S R^R R^I \begin{bmatrix} \dot{x} \\ \dot{y} \\ 0 \end{bmatrix} = R(\alpha + \beta)R(\theta) \begin{bmatrix} \dot{x} \\ \dot{y} \\ 0 \end{bmatrix} \tag{8}$$

where \dot{x} and \dot{y} are the linear velocity of a robot in their respective directions, and the rotations are

$$R(\theta) = \begin{bmatrix} \sin \theta & \cos \theta & 0 \\ -\cos \theta & \sin \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R(\alpha + \beta) = \begin{bmatrix} \sin(\alpha + \beta) & \cos(\alpha + \beta) & 0 \\ -\cos(\alpha + \beta) & \sin(\alpha + \beta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Each wheel-ground contact is a single point, and normal forces acting on the wheel-ground contact points are constant depending on the mass of a robot and gravity. Throughout the kinematic wheel equation derivation steps, we assume that wheels do not slip while rolling. The longitudinal constraint is due to the rolling of wheels on the ground, and the lateral constraint is due to sliding. It is noted that the wheel velocity ${}^W R^I \mathbf{V}_W$ in LHS of Eq. (7) is given as the wheel constraints to each wheel:

$${}^w R^I \mathbf{V}_W = \begin{bmatrix} r\dot{\phi} \\ 0 \\ 0 \end{bmatrix} \begin{array}{l} \text{rolling constraint} \\ \text{no-sliding constraint} \\ \text{planar motion assumption} \end{array} \quad (9)$$

where r is the radius of a wheel, and $\dot{\phi}$ denotes the wheel speed. The rotation in Eq. (7) can be calculated by

$${}^w \mathbf{\Omega}_R \times {}^w R^R \mathbf{P}_W = \begin{bmatrix} 0 & -\dot{\theta} & 0 \\ \dot{\theta} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} l \cos \beta \\ -l \sin \beta \\ 0 \end{bmatrix} = \begin{bmatrix} l \sin \beta \\ l \cos \beta \\ 0 \end{bmatrix} \dot{\theta} \quad (10)$$

where, as shown in Fig. 2, l is the position vector ${}^R \mathbf{P}_S$, β is the wheel angle, and $\dot{\theta}$ denotes the rotation of a robot body (yawing). Equations (8), (9), and (10) are substituted into Eq. (7) to yield two constraints applying to a wheel:

- Rolling constraint:

$$[\cos(\alpha + \beta), \sin(\alpha + \beta), l \sin \beta] R(\theta)^I \dot{\xi} - r\dot{\phi} = 0 \quad (11)$$

- No-sliding constraint (lateral movement):

$$[-\sin(\alpha + \beta), \cos(\alpha + \beta), l \cos \beta] R(\theta)^I \dot{\xi} = 0 \quad (12)$$

where the pose ${}^I \xi = [x, y, \theta]^T$ and velocity ${}^I \dot{\xi} = [\dot{x}, \dot{y}, \dot{\theta}]^T$. We have derived the wheel equation as Eqs. (11) and (12) subject to the two kinematic constraints.

3.2 Differential Kinematics

We have already derived the wheel equation in Sect. 3.1, where a set of wheel speed (constraints) determines robot velocity. Each wheel operates independently in the four-wheeled mobile robot; thus, we must apply the wheel equation to each wheel separately, called the differential kinematic equation. The wheel equation of Eqs. (11) and (12) for a robot with n wheels of the radius r_i , for $i = 1, \dots, n$ is written as follows.

$$\begin{aligned} J_1(\beta_i)R(\theta)^I \dot{\xi} - r\dot{\phi}_i &= 0 \\ C_1(\beta_i)R(\theta)^I \dot{\xi} &= 0 \end{aligned} \quad (13)$$

where

$$\begin{aligned} J_1(\beta_i) &= [\cos(\alpha_i + \beta_i), \sin(\alpha_i + \beta_i), l \sin \beta_i] \\ C_1(\beta_i) &= [-\sin(\alpha_i + \beta_i), \cos(\alpha_i + \beta_i), l \cos \beta_i] \end{aligned}$$

where α_i is the steering offset angle and β_i is the wheel offset angle of i -th wheel. Stacking Eq. (13) to yield a compact matrix equation:

$$\begin{bmatrix} J_1(\beta) \\ C_1(\beta) \end{bmatrix} {}^R \dot{\xi} = \begin{bmatrix} J_2 \\ 0 \end{bmatrix} \dot{\Phi} \quad (14)$$

where ${}^R \dot{\xi} = R(\theta)^I \dot{\xi}$, $\dot{\Phi} = [\dot{\phi}_1, \dots, \dot{\phi}_n]^T$, and

$$J_1(\beta) = \begin{bmatrix} J_1(\beta_1) \\ \vdots \\ J_1(\beta_n) \end{bmatrix} \quad J_2 = \text{diag}(r_1, \dots, r_n) \quad C_1(\beta) = \begin{bmatrix} C_1(\beta_1) \\ \vdots \\ C_1(\beta_n) \end{bmatrix}$$

For a robot with four wheels, we have $J_1(\beta) \in \mathbb{R}^{4 \times 3}$ and $C_1(\beta) \in \mathbb{R}^{4 \times 3}$ because each wheel has one rolling constraint and one no-sliding constraint defined in Eqs. (11) and (12). Equation (14) can be written in a compact form:

$$\mathbf{A} {}^R \dot{\xi} = \mathbf{B} \dot{\Phi} \quad (15)$$

where $\mathbf{A} \in \mathbb{R}^{8 \times 3}$ and $\mathbf{B} \in \mathbb{R}^{8 \times 4}$, and

$$\mathbf{A} = \begin{bmatrix} J_1(\beta) \\ C_1(\beta) \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} J_2 \\ 0 \end{bmatrix}$$

or

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & -l \sin \alpha \\ 1 & 0 & l \sin \alpha \\ 1 & 0 & l \sin \alpha \\ 1 & 0 & -l \sin \alpha \\ 0 & 1 & l \cos \alpha \\ 0 & 1 & l \cos \alpha \\ 0 & 1 & -l \cos \alpha \\ 0 & 1 & -l \cos \alpha \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} r & 0 & 0 & 0 \\ 0 & r & 0 & 0 \\ 0 & 0 & r & 0 \\ 0 & 0 & 0 & r \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Because \mathbf{A} is a rectangular matrix that we cannot obtain an inverse matrix to solve the forward differential equation in Eq. (15) for ${}^R \dot{\xi}$ by a direct manner. Thus, the solution can be given via a pseudo-inverse matrix:

$${}^R \dot{\xi} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{B} \dot{\Phi} \quad (16)$$

or in an explicit form:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} r/4 & r/4 & r/4 & r/4 \\ 0 & 0 & 0 & 0 \\ -r \sin \alpha/4l & r \sin \alpha/4l & r \sin \alpha/4l & -r \sin \alpha/4l \end{bmatrix} \begin{bmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \\ \dot{\phi}_3 \\ \dot{\phi}_4 \end{bmatrix}$$

This concludes the derivation of the forward kinematic solution to the wheel equation upon two kinematic constraints. Hence, with Eq. (16), we may estimate the robot velocity given each wheel speed.

For later use, we need another equation called inverse kinematic solution, where we can relate the wheel speed with the robot velocity. This can be done by solving Eq. (15) for the wheel speed $\dot{\Phi}$:

$$\begin{aligned} \dot{\Phi} &= (\mathbf{B}^T \mathbf{B})^{-1} \mathbf{B}^T \mathbf{A} {}^R \dot{\xi} \\ &= \mathbf{H} {}^R \dot{\xi} \end{aligned} \quad (17)$$

where

$$\mathbf{H} = (\mathbf{B}^T \mathbf{B})^{-1} \mathbf{B}^T \mathbf{A}$$

or

$$\mathbf{H} = \begin{bmatrix} \frac{1}{r} & 0 & -\frac{l \sin(\alpha)}{r} \\ \frac{1}{r} & 0 & \frac{l \sin(\alpha)}{r} \\ \frac{1}{r} & 0 & \frac{l \sin(\alpha)}{r} \\ \frac{1}{r} & 0 & -\frac{l \sin(\alpha)}{r} \end{bmatrix}$$

We have established the forward differential kinematic equation of Eq. (16) and inverse differential kinematic equation of Eq. (17) for the four-wheeled mobile robot.

3.3 Maneuverability

The mobility of a robot is about movement in the environment. The fundamental constraint is the limitation that each wheel should meet the no-sliding constraint, where no lateral movement or slip is allowed. Complying the constraint $C_1(\beta)^R \dot{\xi} = 0$ in Eq. (14), the velocity ${}^R \dot{\xi}$ is the null space of the projection matrix $C_1(\beta)$. The null space is the space N such that vector $n \in N$, that is $C_1(\beta)n = 0$. The robot kinematics is closely related to the set of independent constraints of wheels, which are also directly connected to the rank of the projection matrix. The rank of $C_1(\beta)$ is the number of independent constraints. Thus, three degrees of freedom exist in each wheel. However, since two kinematic constraints apply to each wheel, as we discussed before, the maneuverability may change accordingly. The rank of $C_1(\beta)$ of the robot can be calculated as two and is higher than that of a two-wheeled mobile robot, which is one. Interestingly, the four-wheeled mobile robot has less maneuverability because of degenerated configuration. The degree of mobility can be defined as $\delta_m = 3 - \text{rank}(C_1(\beta))$. Thus, $\delta_m = 1$ for the robot. This can be explained that the robot may control only forward or backward velocity by varying each wheel speed. However, it may not always be possible for the robot to keep the constraints, in particular, in the outdoor environment. The robot may slip and slide perpendicular to the wheel plane. Therefore, such a violation, for example, on an icy road or a terrain with fine grain sand, may result in significant errors on the odometer and cause position tracking into less accurate. That is a reason that we pursued in this study the velocity approach in Eq. (15) rather than the odometric approach.

4 Bayes Filters

4.1 Motivation

To track the robot's motion, we need a model closely following the robot's behavior over time. Although we have derived the differential kinematic equations in Sect. 3.1, this

equation may contain some unknown uncertainty. Alternatively, we may need to understand in depth how the robot will behave in ever-changing wheel-ground interactions in outdoor environments. A variety of uncertainty may arise due to various factors: such as erratic hardware, software flaws, motors, sensors, or even slippage of wheels. Filtering techniques have been used to alleviate such uncertainties for many applications. The Bayesian framework can explain the technique best, forming the Kalman filtering technique and several other probabilistic algorithms. In the probabilistic fashion, the estimated velocity of a robot can be represented by a probability density function relating to the state space. The idea with the Bayes filter is to employ Bayes' rule, and the corresponding notion of conditional probability to form probability distributions representing a belief in the robot's state, given the robot's previous state and predicted state. The filter outputs the probability of the state variables at each time step. The prediction is the prior belief of the robot's state after a motion before updating recent sensor measurements. In the measurement update step, we compute the posterior.

In this study, rather than disregarding uncertainty and forming an estimated state, we employed the probability to yield mathematically optimal estimates using the unscented Kalman filter and Kalman filter algorithm. The goal is to account for the robot's uncertainty as it interacts with outdoor environments with insufficient information.

4.2 Bayes' Filter and Robot Motion

We have sought a model that can describe the robot's motion in outdoor environments over time. As seen in the forward differential equation of Eq. (16), the velocity $\dot{\xi}$ of a robot is dictated by the wheel speed measurements Φ . Hence, an ideal motion model can be written in a joint distribution of the velocity and wheel speed measurement:

$$p(\dot{\xi}_1, \dots, \dot{\xi}_k, \Phi_1, \dots, \Phi_k) \quad (18)$$

If we could get such a model of Eq. (18), the motion of a robot can be tracked and explained over the entire time steps. However, dictating the dependencies of future measurements on all previous measurements, the model would be impractical because its complexity would grow unbounded as the number of measurements accrues. For example, when a new measurement Φ_{k+1} comes in, we need to recalculate the states over the entire time steps, which is computationally intractable in practice. Therefore, we may alleviate the requirement and calculate the probability for the most recent step, confining the full joint probability model to a tractable model using some assumptions. The Markov assumption implies that the current state variable $\dot{\xi}_k$ contains all the

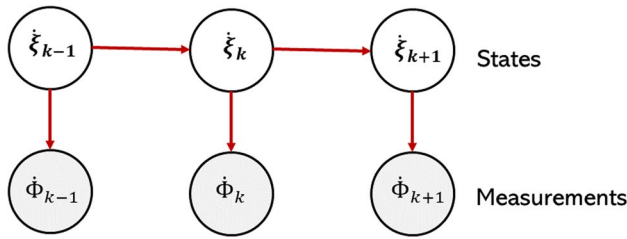


Fig. 3 Graph of a state space model for a robot’s motion and measurement. The Markov process implies that the current state variable ξ contains all the information needed to compute the next state. We may update the state variable only through the measurements Φ and the most recent estimated state

information needed to compute the next state. This paradigm can be thought of as the state-space model in Fig. 3. In so doing, we may come up with a motion model with fewer requirements to meet.

Bayes’ filter technique can be used to estimate the state variables ξ . It is noted that the forward kinematic solution we have derived in Sect. 3.1 is a motion model without considering noise, whereas Bayes’ filter model could embrace uncertainty both in the system equations and the measurement.

$$\begin{aligned}
 bel(\xi_k) &= p(\xi_k | \Phi_{1:k}) \\
 &= \frac{p(\Phi_{1:k} | \xi_k) p(\xi_{1:k})}{\int p(\Phi_k | \xi_k, \Phi_{1:k-1}) p(\xi_k | \Phi_{1:k-1}) d\xi_k} \quad (20) \\
 &= \eta p(\Phi_k | \xi_k, \Phi_{1:k-1}) p(\xi_k | \Phi_{1:k-1})
 \end{aligned}$$

where η is the normalizing constant, which is a marginalization of the denominator over ξ_k :

$$\eta = \left[\int p(\Phi_k | \xi_k, \Phi_{1:k-1}) p(\xi_k | \Phi_{1:k-1}) d\xi_k \right]^{-1}$$

We have derived the belief for estimating the unknown state based on the measurement as shown in Eq. (20). However, in this study, we employed the probabilistic models of the motion and measurement separately; thus, Eq. (20) still seems impractical to be used for the calculation. Therefore, with further mathematical manipulation, we expanded Eq. (20) into a more explicit form with models of motion and measurement. With the help of Bayes’ rule and Markov process assumption, it resulted in a more tractable equation shown in Eq. (21).

Algorithm 1 Algorithm Bayes Filter

- 1: **for** all ξ_k **do**
- 2: $\overline{bel}(\xi_k) \leftarrow \int_{\xi_{k-1}} \underbrace{p(\xi_k | \xi_{k-1})}_{\text{Motion model}} \overline{bel}(\xi_{k-1}) d\xi_{k-1}$ ▷ Prediction step
- 3: $bel(\xi_k) \leftarrow \underbrace{\eta p(\Phi_k | \xi_k)}_{\text{Measurement model}} \overline{bel}(\xi_k)$ ▷ Update step
- 4: **end for**
- 5: **return** $bel(\xi_k)$

Bayes’ filter algorithm [31] can be briefly explained in Algorithm 1. We need a model that can mimic the behavior of a four-wheeled mobile robot, in particular, under unknown outdoor environments. However, it is impossible to earn the fully jointed probability for the model shown in Eq. (18) analytically; a conditional probability can be employed instead. In so doing, we may define the belief for the current state as

$$bel(\xi_k) = p(\xi_k | \Phi_{1:k}) \quad (19)$$

It is shown that the belief of Eq. (19) is based on the measurement over time as presented by a probability density function distributed over the state space. By applying Bayes’ rule to Eq. (19) to yield

$$\begin{aligned}
 bel(\xi_k) &= \eta p(\Phi_k | \xi_k, \Phi_{1:k-1}) p(\xi_k | \Phi_{1:k-1}) \quad (B) \\
 &= \eta p(\Phi_k | \xi_k) p(\xi_k | \Phi_{1:k-1}) \quad (M) \\
 &= \eta p(\Phi_k | \xi_k) \int p(\xi_k | \xi_{k-1}, \Phi_{1:k-1}) p(\xi_{k-1} | \Phi_{1:k-1}) d\xi_{k-1} \quad (T) \\
 &= \eta p(\Phi_k | \xi_k) \int p(\xi_k | \xi_{k-1}) p(\xi_{k-1} | \Phi_{1:k-1}) d\xi_{k-1} \quad (M) \\
 &= \eta p(\Phi_k | \xi_k) \int p(\xi_k | \xi_{k-1}) bel(\xi_{k-1}) d\xi_{k-1} \quad (21)
 \end{aligned}$$

where (B) for Bayes’ rule applied, (M) for Markov process assumed, (T) for total probability, where

$$bel(\xi_{k-1}) = p(\xi_{k-1} | \Phi_{1:k-1})$$

As can be seen in Eq. (21), we derived a recursive Bayes filter. The recursive refers to the fact that the belief in the present time $bel(\xi_k)$ is determined by the belief in the previous time step $bel(\xi_{k-1})$ recursively. In addition, the belief $bel(\xi_k)$ is expressed in terms of the motion model $p(\xi_k|\xi_{k-1})$ and the measurement model $p(\Phi_k|\xi_k)$. For the computation's sake, the recursive process can be divided into two steps shown in Algorithm 1: one is for the prediction, and the other for the update. The prediction step is defined as

$$\overline{bel}(\xi_k) = \int p(\xi_k|\xi_{k-1})bel(\xi_{k-1})d\xi_{k-1}$$

and the update step is written as

$$bel(\xi_k) = \eta p(\Phi_k|\xi_k)\overline{bel}(\xi_k)$$

In conclusion, as shown in Algorithm 1, with the help of the Bayes filter, through recursive fashion on the prediction and update, we can successfully track the motion of the four-wheeled mobile robot based on the wheel-speed measurement. There are a few filters that are Bayes' filter's practical realization, avoiding direct analytic high-dimensional integration; the Kalman filter is a classical optimal filter for linear-Gaussian models. The unscented Kalman filter is a sigma-point transformation-based extension of the Kalman filter. In this study, we employed the Kalman filter and the unscented Kalman filter to realize Bayes' filter in comparison.

4.3 Kalman Filter

The classical recursive solution to the state space estimation is called the optimal filter. The filter computes the posterior distribution of the state or the belief $bel(\xi_k)$ given the measurements as written in Eq. (21). Typical linear Gaussian state-space models in probabilistic notation are expressed as

$$\begin{aligned} p(\Phi_k|\xi_k) &= \mathcal{N}(\Phi_k|\mathbf{H}\xi_k, \mathbf{R}) && \text{Measurement model} \\ p(\xi_k|\xi_{k-1}) &= \mathcal{N}(\xi_k|\mathbf{F}\xi_{k-1}, \mathbf{Q}) && \text{Motion model} \\ p(\xi_0) &= \mathcal{N}(\mathbf{m}_0, \mathbf{P}_0) && \text{Prior distribution} \end{aligned} \quad (22)$$

where the measurement matrix \mathbf{H} , the state transition matrix \mathbf{F} , the measurement noise matrix $\mathbf{R} = \mathcal{N}(0, \sigma_\phi^2)$, the system noise matrix $\mathbf{Q} = \mathcal{N}(0, \sigma_\xi^2)$, and σ_ϕ and σ_ξ are their respective covariances. The prior distribution $p(\xi_0)$ models the information already known with the mean \mathbf{m}_0 and covariance \mathbf{P}_0 . Substitution of Eq. (22) into Eq. (21), the optimal filter solution via Kalman filter is obtained in a form of Gaussian distribution. Hence, the solution in Eq. (19) can be obtained with the mean \mathbf{m}_k and covariance \mathbf{P}_k as follows:

$$p(\xi_k|\Phi_{1:k}) = \mathcal{N}(\mathbf{m}_k, \mathbf{P}_k) \quad (23)$$

We assumed that the robot's velocity might not vary significantly at each time step due to the low wheel speeds and rather a short sampling time ($\Delta t = 0.1$ s). Thus, the transition matrix of Eq. (22) can be set as $\mathbf{F} = \mathbf{I}$, which is the identity matrix. The measurement matrix \mathbf{H} is given in Eq. (17). All of the state variables are set to zero initially, the initial covariance is 5.0, the system noise \mathbf{Q} is a discrete white noise with a variance of 0.5, and a variance of the measurement noise \mathbf{R} is 0.5. Those values are hyperparameters, which can be adjusted by the optimal computational and accuracy consideration through several test runs.

4.4 Unscented Kalman Filter

Unscented Kalman Filter (UKF) is one of newly developed Bayes filters [32]. The UKF utilizes a series of transformation and sigma points instead of direct use of state variables as in the Kalman filter. The entire process of UKF can be divided into three algorithms: Sigma point algorithm, unscented transform, an unscented Kalman filter algorithm.

Algorithm 2 Algorithm Van der Merwe Sigma Point Algorithm

```

procedure VAN DER MERWE SIGMA POINT ALGORITHM( $\mu, \Sigma$ )
  for all  $2n + 1$  sigma points do
     $\lambda = \alpha^2(n + \kappa) - n$ 
     $\chi_0 \leftarrow \mu$ 
     $\chi_i \leftarrow \mu + \left[ \sqrt{(n + \lambda)\Sigma} \right]_i$ 
     $\chi_i \leftarrow \mu + \left[ \sqrt{(n + \lambda)\Sigma} \right]_{i-n}$ 
     $w_0^m = \frac{\lambda}{n + \lambda}$ 
     $w_0^c = \frac{\lambda}{n + \lambda} + 1 - \alpha^2 + \beta$ 
     $w_i^m = w_i^c = \frac{1}{2(n + \lambda)}$ 
  end for
  return  $\chi, w^m, w^c$ 
end procedure

```

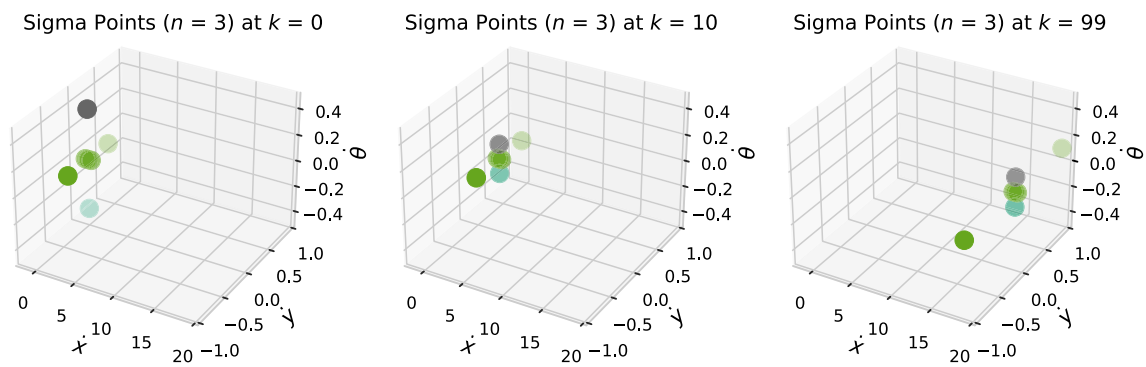



Fig. 4 Sigma points χ at $k = 0, k = 10,$ and $k = 99$ for the unscented Kalman filter on a normal road. The depth of the point color is proportional to the value of θ

4.4.1 Van der Merwe’s Scaled Sigma Point Algorithm

There are a few algorithms for sigma points generation, among which Merwe’s Scaled Sigma Point Algorithm has been most popular for superior performance to others. The Merwe’s algorithm uses three parameters to control how the sigma points are distributed and weighted: $\alpha = 0.1, \beta = 2,$ and $\kappa = 0$ are used in this study. A detailed algorithm is shown in Algorithm 2. The algorithm takes the mean and covariance of the state variables and outputs the corresponding sigma points $\chi,$ the weights $w^m,$ and $w^c.$ The sampled sigma points for a normal road are displayed in Fig. 4. In the figure, the points at three different time steps are shown in the temporal order. The figure shows a pattern of how the sigma points are generated as time goes.

The unscented transform (UT) is the essence of the unscented Kalman filter algorithm as written in Algorithm 3. Firstly, the sigma points χ are passed through a nonlinear function resulting in a transformed set of points $\mathcal{Y}.$ The UT takes the sigma points sampled from some arbitrary probability distribution, passes them through a function f (motion model in this study), and outputs a Gaussian, of which weighted mean μ and weighted variance $\Sigma,$ for each transformed point.

4.4.2 Unscented Transform

Algorithm 3 Algorithm Unscented Transform

```

procedure UT( $\mathcal{Y}, w^m, w^c, Noise$ )
  for  $i = 0$  to  $2n$  do  $\triangleright n = \dim(\xi)$ 
     $\mu \leftarrow \mu + w_i^m \mathcal{Y}_i$ 
     $\Sigma \leftarrow \Sigma + w_i^c (\mathcal{Y}_i - \mu)(\mathcal{Y}_i - \mu)^T + Noise$ 
  end for
  return  $\mu, \Sigma$ 
end procedure

```

Algorithm 4 Algorithm Unscented Kalman Filter

```

procedure UNSCENTED KALMAN FILTER( $\dot{\xi}^-, \dot{\Phi}$ )
  for all  $\xi$  do
     $\mathcal{Y} \leftarrow f(\chi)$  ▷ Prediction step
     $\dot{\xi}^-, \mathbf{P}^- \leftarrow UT(\mathcal{Y}, w^m, w^c, \mathbf{Q})$  ▷ Unscented transform (Algorithm 3)
     $\mathcal{Z} = h(\mathcal{Y})$  ▷ Update step
     $\mu_{\dot{\Phi}}, \mathbf{P}_{\dot{\Phi}} \leftarrow UT(\mathcal{Z}, w^m, w^c, \mathbf{R})$  ▷ Unscented transform (Algorithm 3)
     $\mathbf{y} = \dot{\Phi} - \mu_{\dot{\Phi}}$  ▷ Measurement residual
     $\mathbf{P}_{\xi\dot{\Phi}} = \sum_{i=0}^{2n} w_i^c (\mathcal{Y}^i - \dot{\xi}^-)(\mathcal{Z}^i - \mu_{\dot{\Phi}})^T$  ▷ Cross covariance
     $\mathbf{K} \leftarrow \mathbf{P}_{\xi\dot{\Phi}} \mathbf{P}_{\dot{\Phi}}^{-1}$  ▷ Kalman gain
     $\dot{\xi} \leftarrow \dot{\xi}^- + \mathbf{K}\mathbf{y}$  ▷ States estimates
     $\mathbf{P} \leftarrow \mathbf{P}^- - \mathbf{K}\mathbf{P}_{\dot{\Phi}}\mathbf{K}^T$ 
  end for
  return  $\dot{\xi}, \mathbf{P}$ 
end procedure

```

4.4.3 Prediction Step

The UKF is divided into two steps: the prediction and the update step. A transformed set of sigma points is

$$\mathcal{Y} = f(\chi) \quad (24)$$

The prediction step of the UKF computes the prediction (prior) as listed in Algorithm 4 using the sigma points χ generated from Algorithm 2. The predicted mean $\dot{\xi}^-$ and covariance \mathbf{P}^- using the UT on the transformed sigma points are computed as follows:

$$\dot{\xi}^-, \mathbf{P}^- = UT(\mathcal{Y}, w^m, w^c, \mathbf{Q})$$

where

$$\dot{\xi}^- = \sum_{i=0}^{2n} w_i^m \mathcal{Y}^i, \quad \mathbf{P}^- = \sum_{i=0}^{2n} w_i^c (\mathcal{Y}^i - \dot{\xi}^-)(\mathcal{Y}^i - \dot{\xi}^-)^T$$

This result is obtained while the measurement is yet seen; therefore, the correction can be made through the update step.

4.4.4 Update Step

The UKF performs the update in measurement space as shown in Algorithm 4. Thus, we need to map the sigma points into the measurement space using a measurement function h . A set of transformed points in the measurement space is:

$$\mathcal{Z} = h(\mathcal{Y})$$

The mean and covariance of those points of \mathcal{Z} through the UT are given by

$$\mu_{\dot{\Phi}}, \mathbf{P}_{\dot{\Phi}} = UT(\mathcal{Z}, w^m, w^c, \mathbf{R}) \quad (25)$$

where

$$\mu_{\dot{\Phi}} = \sum_{i=0}^{2n} w_i^m \mathcal{Z}^i, \quad \mathbf{P}_{\dot{\Phi}} = \sum_{i=0}^{2n} w_i^c (\mathcal{Z}^i - \mu_{\dot{\Phi}})(\mathcal{Z}^i - \mu_{\dot{\Phi}})^T + \mathbf{R}$$

The residual of the measurement \mathbf{y} is

$$\mathbf{y} = \dot{\Phi} - \mu_{\dot{\Phi}} \quad (26)$$

To compute the Kalman gain we first compute the cross covariance of the state and the measurements of Eqs. (25) and (26).

$$\mathbf{P}_{\xi\dot{\Phi}} = \sum_{i=0}^{2n} w_i^c (\mathcal{Y}^i - \dot{\xi}^-)(\mathcal{Z}^i - \mu_{\dot{\Phi}})^T \quad (27)$$

And then the Kalman gain is defined with Eqs. (25) and (27) as

$$\mathbf{K} = \mathbf{P}_{\xi\dot{\Phi}} \mathbf{P}_{\dot{\Phi}}^{-1}$$

Finally, we obtain the estimated $\dot{\xi}$ and \mathbf{P} using the residual and the Kalman gain as shown in Algorithm 4 as follows.

$$\dot{\xi} = \dot{\xi}^- + \mathbf{K}\mathbf{y}, \quad \mathbf{P} = \mathbf{P}^- - \mathbf{K}\mathbf{P}_{\dot{\Phi}}\mathbf{K}^T \quad (28)$$

In so doing, we may conclude that the tri-variate states (two linear velocities and rotation) of the robot has been realized as a numerically tractable form in Eq. (28).

5 Four-Wheeled Mobile Robot

In this study, we set up a four-wheeled mobile robot, as shown in Fig. 5, operated by a DC motor at each wheel, which comes with a 120:1 gearbox and an integrated quadrature encoder that provides a resolution of 16 pulse single per round giving a maximum output of 1920 within one round. The dimension of the robot is the length of 24 cm, the width of 17.5 cm, a diameter of a wheel of 6.6 cm. A Raspberry Pi single-board computer delivers commands into the robot and receives measurements from the encoders. The reference wheel speed is set to 6.54 rad/s (200 encoder ticks per second) for all the wheels throughout the test runs. In other words, regardless of road conditions, we demand that each wheel spins at the reference wheel speed. Thus, as long as an ideal condition to the robot’s motion holds without any hindrance, the robot would move forward at 21.6 cm/s in the heading direction. The information of electric specification for the robot is listed in Table 1.

6 Results and Discussion

We had more than a thousand test runs along with five different road conditions: normal, sandy, icy, partially blocked, and grid holes. The detail of the tests is listed in Table 2. The reference wheel speed was set to a constant of 6.54 rad/s (about 62.5 rpm) throughout the tests.

Table 1 Overall specification of the robot system and the encoders

Item	Content
Gear ratio	120:1
No-load speed @ 6V	160 rpm
No-load speed @ 3V	60 rpm
No-load current @ 6V	0.17A
No-load speed @ 3V	0.17A
Max Stall current	2.8A
Max Stall torque	0.8 kgf cm
Rated torque	0.2 kgf cm
Motor operating voltage	3 to 7.5V
Operating ambient temperature	– 10 to 60
Encoder	SJ01 (16 pulses per revolution)

The overall schematic of different roads is displayed in Fig. 6. For example, we may anticipate that wheels can quickly get stuck with fine grain sand. In Fig. 7, variation of each wheel speed on a normal road for 10 s is shown. We initiated the robot on a normal road, demanding each wheel spin at the reference wheel speed so that the robot may go straight in the heading direction. Interestingly, the rear right wheel spins faster than the others. The median and variance of the velocities is shown in Fig. 8. However, it is not clear that what causes the rear right wheel to run faster. Since we built the robot with identical DC motors and wheels, we expected all wheels to run at the same speed upon the same road condition. It could be an encoder error

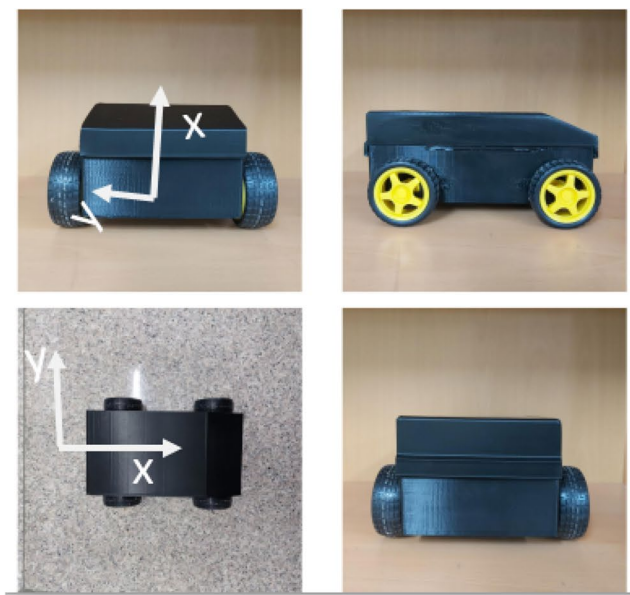


Fig. 5 Different views of the four-wheeled mobile robot with a fixed standard wheel system. Four optical encoders are equipped into each wheel to send wheel speed data through the Raspberry Pi computer board

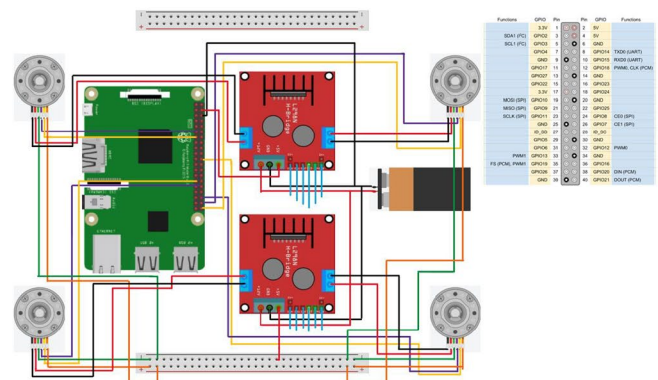


Table 2 Specification of the five different road conditions in which the robot runs more than a thousand tests

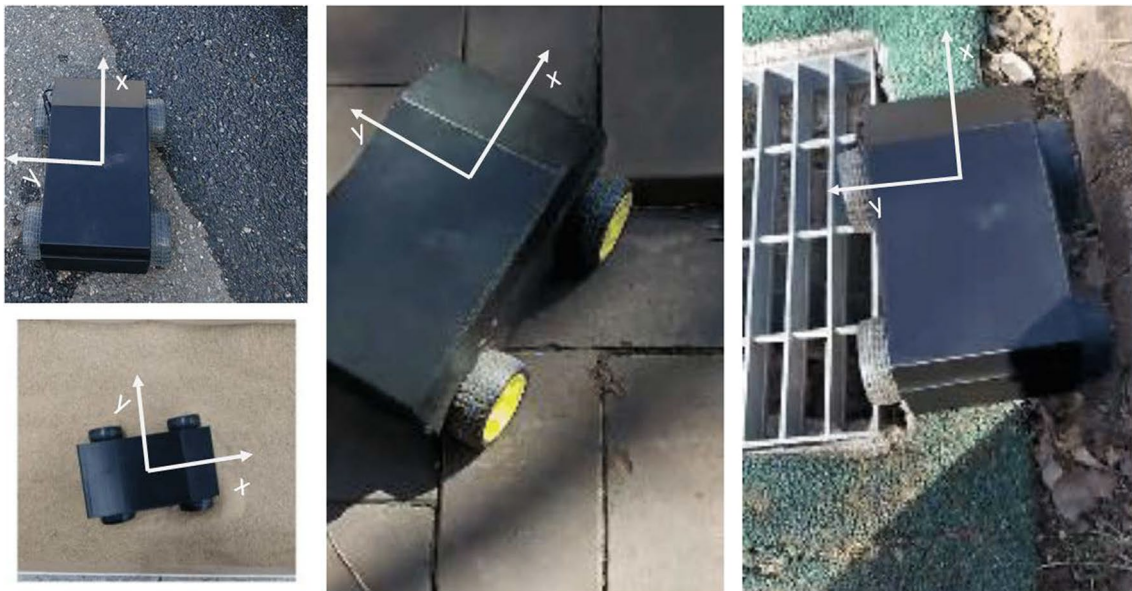
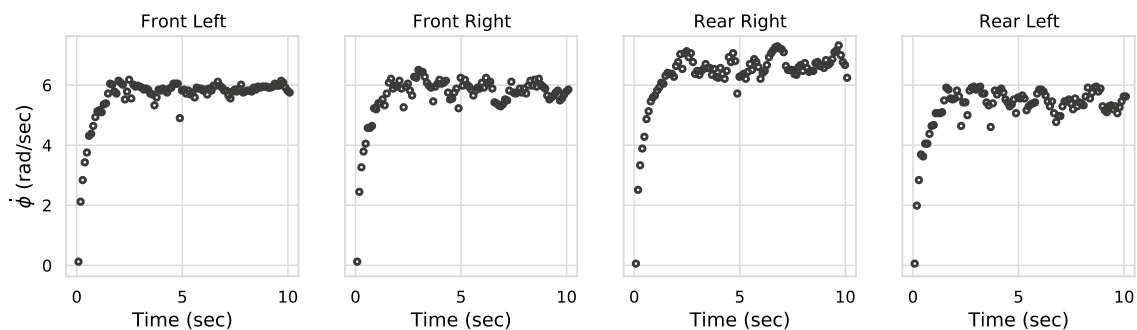
Roads	Wheel speed (rad/s)	Number of test	Remark
Normal	6.54	200 >	Dry
Ice	6.54	200 >	Thin ice
Sand	6.54	200 >	Fine grain sand
Partially blocked	6.54	200 >	Physically blocked
Grid holes	6.54	200 >	Grid type holes

The wheel speed of each wheel is set to 6.54 rad/s (about 62.5 rpm) for the entire tests

or road conditions, or unknown flaws in the robot itself. Unmistakably, this unbalanced wheel speed can bring about some uncertainty and difficulty to the analysis. Therefore,

we need to know whether that unbalance in wheel speeds should be regarded as statistically significant or not. That is, the presence of this unbalance may not be justified without rigorous statistical examination. Thus, we performed an ANOVA (Analysis of Variance) test on it. With the ANOVA $F = 28.5$ and the p-value is significant $p < 0.05$, we may conclude significant speed differences among wheels with a 95% confidence level. Hence, we learned that not all wheels run at an identical speed upon demanding them so. Such an unexpected issue is not uncommon in the actual world, making modeling harder.

A comparison of different filtering techniques for the estimated state velocities of the robot on a normal road is shown in Fig. 9. The velocity of heading direction \dot{x} is displayed in the figure (left). The expected velocity (red-dashed line) is 21.6 cm/s (6.25 rad/s with 6.6 cm of the diameter of the wheel). However, any of the estimated velocity has not

**Fig. 6** The robot runs on difference roads. Icy road (top left), sandy road (bottom left), partially blocked by a brick (center), and grid type holes (right)**Fig. 7** Measured wheel speeds of the robot on a normal road

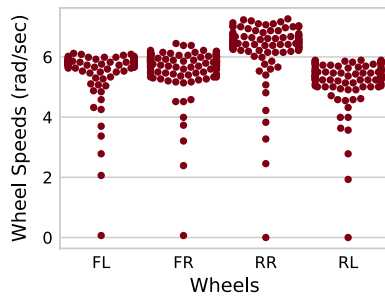


Fig. 8 A swarm plot of the measured wheel speed of each wheel of the robot on a normal road. Note FL (forward left), FR (forward right), RR (rear right), and RL (rear left). It is observed that RR seems to spin faster than the others

reached the designated value. We were not sure whether it was because of the unbalanced wheel speed we observed earlier or varying ground-wheel conditions, or both. Among two filtering algorithms, the unscented Kalman filter shows an excellent fit to the estimated velocity from experiment data (no filter). Furthermore, the no-lateral movement constraint seems to be kept all the time (center pane in the figure). Thus, the robot does not make a lateral move (no sliding). Even so, the robot’s rotation (right) shows some interesting behaviors. Even if no rotation is theoretically allowed due to the fixed wheel, some positive rotation rate is seen, which is the counter-clockwise rotation. We may conclude that such rotation is caused by the unbalanced wheel speed in which the rear right wheel runs faster and the road condition.

In Fig. 10, it is shown that the probability density function of the estimated velocity in the robot heading direction \dot{x} evolves at each time step, which is calculated via the Kalman filter. As time goes, the distribution gets narrow and higher probability density value; we may conclude that the model closely estimates the robot’s motion on a normal road.

In Fig. 11, the measured wheel speeds on an icy road are shown. Each wheel’s speed appears relatively more scattered. Besides, considering the faster speed in the rear-right wheel discussed earlier, the front-left wheel spins a bit fast. That can be because of the slippage of the wheels on an icy road. Once a wheel gets into a slippage, the robot may not get back to a designated pose without external intervention, such as a human operator. We observed that no single test was similar to the others during several test runs on an icy road. Therefore, analytical modeling based on a wheel-ground interaction for an icy road might not be practical at all. It is because each time we place the robot on an icy road, an ice condition may change due to inexplicable factors such as the non-uniform melting of ice due to wheel-ice contacts.

The performance comparison of modeling on an icy road is shown in Fig. 12. The velocity \dot{x} (left) in the heading direction is significantly displaced from the reference and scattered. Even so, the unscented Kalman filter appears to perform exceptionally well against the uncertainty of the measurements. The lateral velocity \dot{y} (center) seems to remain around zero. As for the rotation rate $\dot{\theta}$ (right), significant impacts must have been exerted to the robot that the rotation rate peaks of about 0.6 rad/s during 4–7 s. Then, the robot would likely move sideways quite violently due to this heavy rotational rate change. According to the forward differential equation in Eq. (16), the robot is not supposed to be rotated at any moment (see the dashed red line in the figure). However, this rule seems to be much violated due to an icy road. That helpful information might be hard to obtain when only sticking to analytical models without allowing any uncertainty in motion and measurement models.

In Bayes’ framework, the estimated results are produced in probability, particularly for Gaussian distribution. We drew a plot with the mean and covariance of the distribution. In so doing, we may get more information on how the filtering algorithm works on various conditions. In Fig. 13,

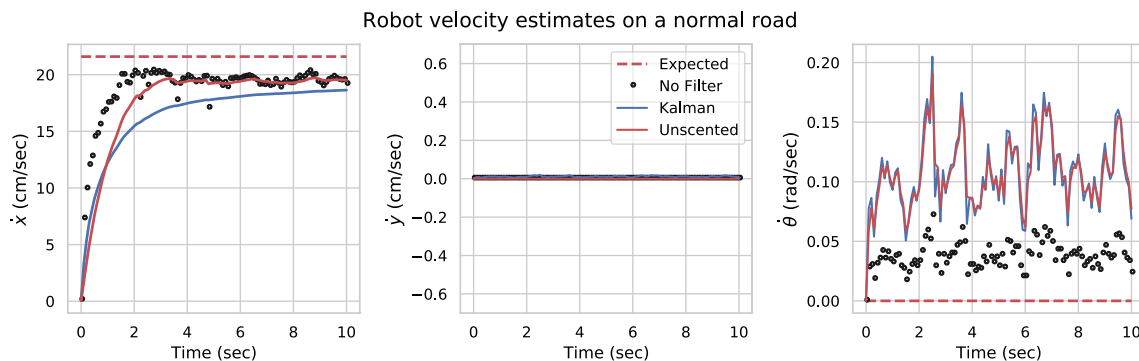


Fig. 9 A comparison of the estimated state velocities (\dot{x} , \dot{y} , and $\dot{\theta}$) of the robot on a normal road through different algorithms (no filters, Kalman filter, and unscented Kalman filter). The expected velocity is 21.6 cm/s (6.25 rad/s with 6.6 cm of the diameter of the wheel)

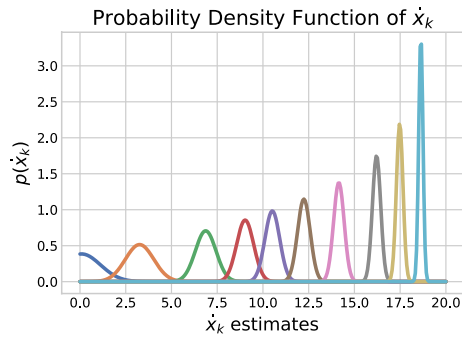


Fig. 10 On a normal road, the estimated velocity ξ in the form of Gaussian distribution is evolving at each time step. Each distribution is calculated via the Kalman filter

the ellipsoids of the covariance obtained from the Kalman filter and unscented Kalman filter are shown, respectively. The estimated velocity ξ is composed of three elements of velocities \dot{x} , \dot{y} , and $\dot{\theta}$. The ellipsoid represents the covariance space; thus, the larger the space, the less belief we can rely on from the result. Also, we need to pay some attention to the shape of an ellipsoid. For example, the ellipsoid being similar to a sphere can be considered that three elements are

not correlated much. The dots on the surface of an ellipse are sampled from the tri-variate normal distribution. Three panels at the top of the figure show the temporal trend of covariance via the Kalman filter. As time goes, although the ellipsoid gets smaller and longer, the red dots appear to be away from the surface of the ellipsoids. This explains why the Kalman filter solution deteriorates over time. Likewise, three panels at the bottom of the figure show the trend of the covariance from the unscented Kalman filter. As can be seen, most of the green dots appear to dwell closely on the surface of the ellipsoids. As is seen in the comparison, we may conclude that the unscented Kalman filter works better than the Kalman filter in general.

Figure 14 shows the wheel speeds on a sandy road. The road is covered by a 2-cm thick fine grain of sand as shown in Fig. 6. We might expect that the robot gets in easily stuck intermittently into slippage states. Thus, it may be hard for the robot to maintain the designated wheel speed all the time. Furthermore, front-left and rear-left wheels spin at the reverse rotation rate of about negative 2 rad/s during the first half of the test run. Then, the wheel comes to a stop and then gets back in the reverse direction. In this test, we can be sure

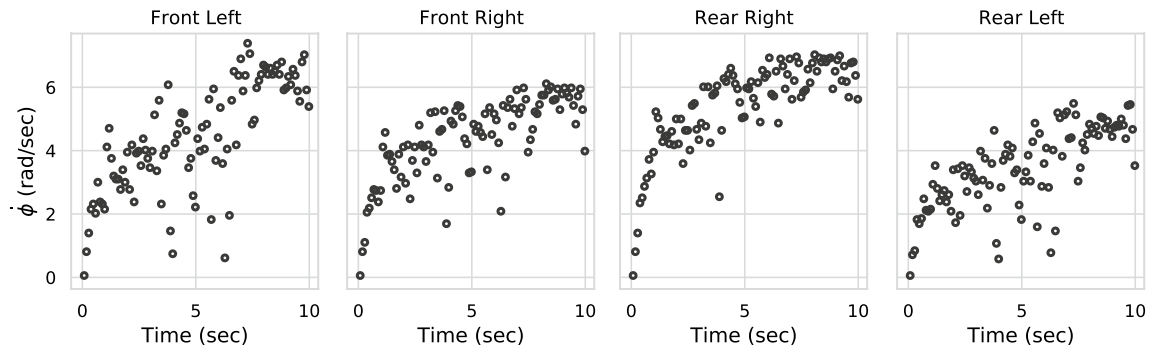


Fig. 11 Measured wheel speeds of the robot on an icy road

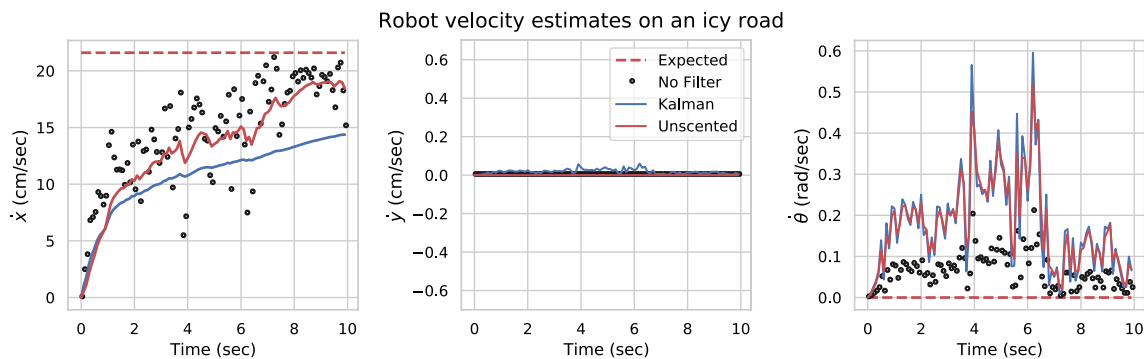


Fig. 12 A comparison of the estimated state velocities (\dot{x} , \dot{y} , and $\dot{\theta}$) of the robot on an icy road through different algorithms (no filters, Kalman filter, and unscented Kalman filter). The expected velocity is 21.6 cm/s (6.25 rad/s with 6.6 cm of the diameter of the wheel)

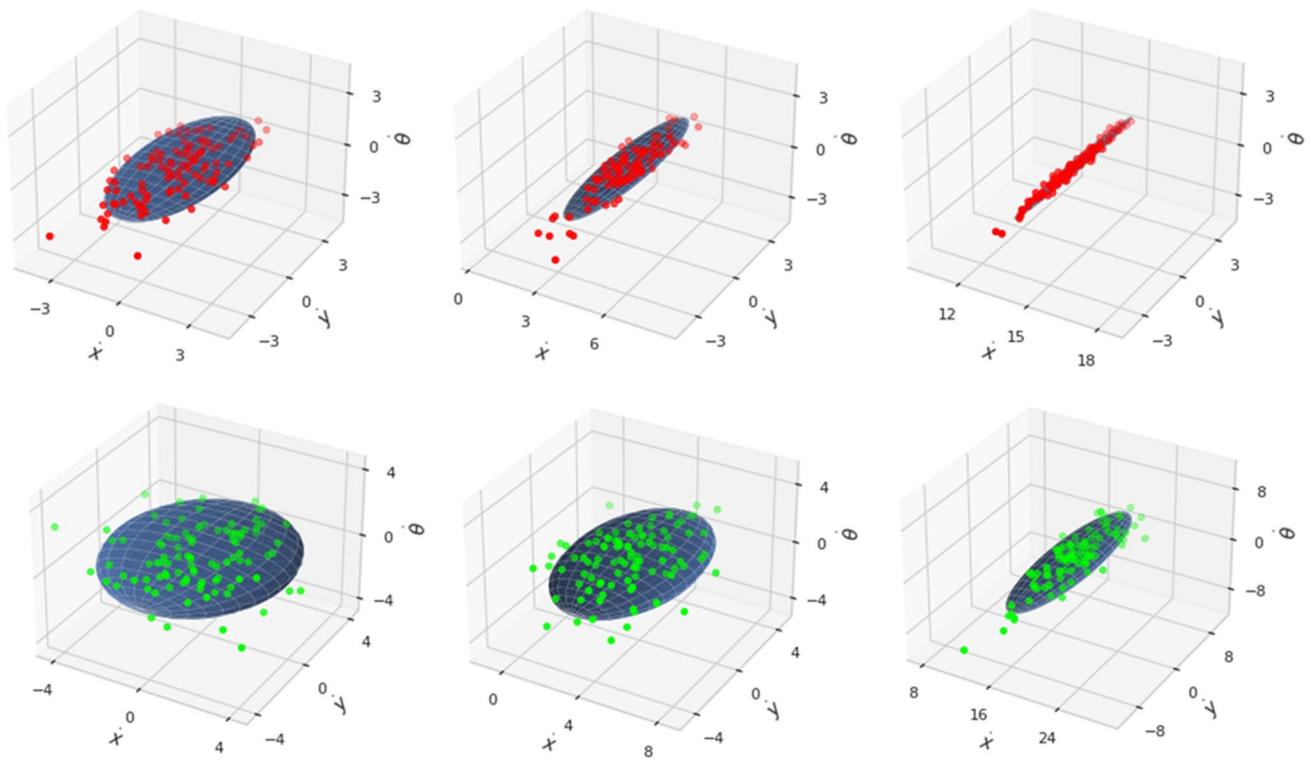


Fig. 13 Comparison of the estimated covariances on an icy road calculated through the Kalman filter (red) and the unscented Kalman filter (green) at $T = 0$, $T = 0.5$, and $T = 9.9$ s. In order to get a better appearance for comparison, each ellipsoid is drawn not to scale. The

red (Kalman filter) and green (unscented Kalman filter) dots are sampled from their respective tri-variate Gaussian distributions. (Color figure online)

that fine grain of sand is a great hindrance to the wheels' spin, in particular, during the first half of the test run.

A comparison of the estimated state velocities of the robot on a sandy road is shown in Fig. 15. The velocity in the heading direction \dot{x} goes up and down. It has reached about 16 cm/s at most, which is below the expected velocity. As for the lateral velocity \dot{y} , it lands around zero; thus, the no-sliding constraint has been met. The rotation rate shows a significant peak between 2 and 3 s. The unbalanced wheel speed may cause this peak in the rotation. Furthermore, the

rotation rate has changed significantly in a relatively short time. It may be because of the big motion that the robot makes in order to get out of stuck in the sand. As for the overall performance, the unscented Kalman filter gets the highest score.

In Fig. 16, each wheel speed when the robot is partially blocked by brick is shown. Initially, the wheel speeds go down immediately. Around 5 s, the wheel speeds of the front-left and rear-right wheels bounce back to near the designated wheel speed. It can be explained that the robot being

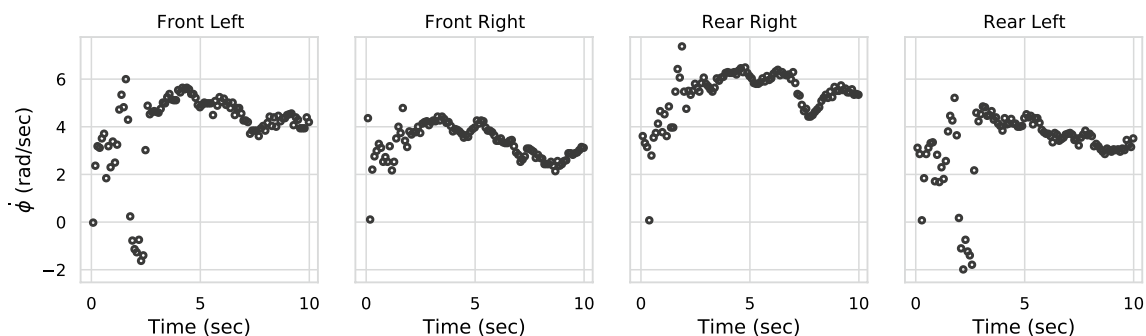


Fig. 14 Measured wheel speeds of the robot on a sandy road

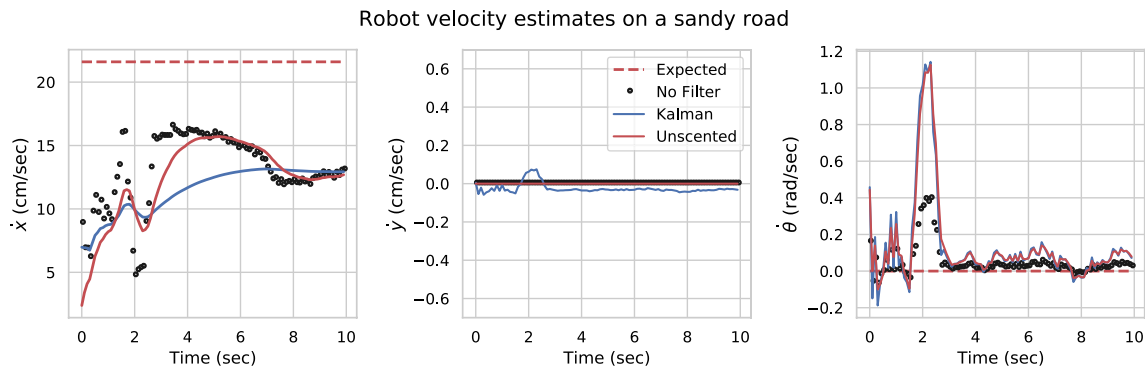


Fig. 15 A comparison of the estimated state velocities (\dot{x} , \dot{y} , and $\dot{\theta}$) of the robot on a sandy road through different algorithms (no filters, Kalman filter, and unscented Kalman filter). The expected velocity is 21.6 cm/s (6.25 rad/s with 6.6 cm of the diameter of the wheel)

blocked can cause a lowering of its wheel speeds. Since the robot head and the brick are partially contacted over a small area, the robot tends to rotate clockwise about the brick. Even though the brick keeps the robot from going forward, the rotation will occur to be out of the blockage; thus, the wheel speeds rise again.

The estimated velocities of the robot partially blocked are displayed in Fig. 17. The velocity of heading direction \dot{x} seems to be directly affected by a subtle change of each wheel speed due to the blockage. It appears that the brick keeps the robot from moving forward, and \dot{x} is suppressed under a speed of 15 cm/s. The lateral velocity remains around zero without violating the constraint. The rotation rate shows quite an interesting result that in the first half, the counter-clockwise rotation occurs, followed by the clockwise rotation. That direction change informs us of helpful information about how the robot makes movement subject to a brick on a partial contact. As seen in the figure, the unscented Kalman filter performs well throughout the test run.

Figure 18 shows the wheel speeds on grid-type holes. The robot initially starts on a normal road and then gets into the grid holes. We might anticipate that the robot does not freely move as intended because of the complicated ground-wheel

conditions of grid-type holes. This may be true, as seen in the figure. However, it is pretty difficult for us to get any explicable information from the result in this case. Besides, as seen in Fig. 6, two wheels get stuck in the grid holes, whereas the other wheels are on a dirt road. There might not exist an easy solution to incorporate such road conditions into physics laws or empirical equations to be used as a wheel-ground interaction model.

In Fig. 19, the velocity of heading direction \dot{x} appears to be affected by the grid holes. In this case, the two robot wheels get stuck into the grid holes while the others are on the normal road. The lateral velocity remains around zero. The rotation rate $\dot{\theta}$ shows that the positive and negative rotation rates. However, it is not easy to explain the cause of the rotations because of the complexity of the motion in this case.

We have made several comparisons between Bayes' algorithms (the Kalman filter and the unscented Kalman filter). Besides comparing the results, it might be interesting to pay attention to the numerical aspect of those algorithms. Since Bayes' rule is composed of the multiplication of likelihoods and priors to yield posteriors, we need to check the behavior of likelihoods given each measurement. The maximum likelihood estimation (MLE) often calculates the optimal parameters for

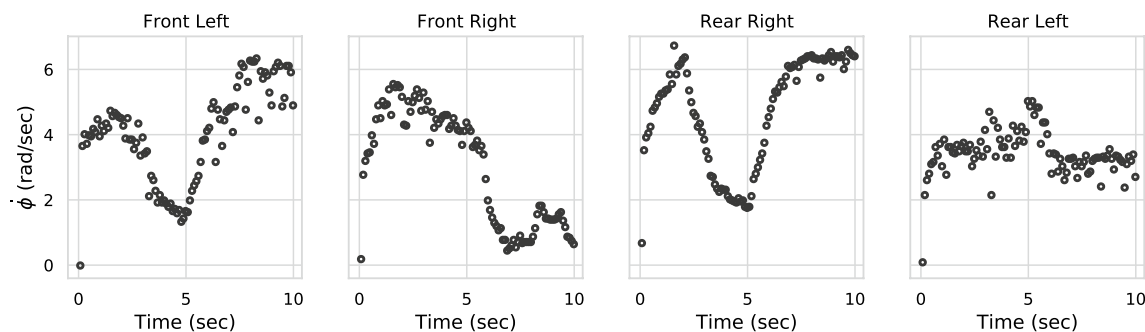


Fig. 16 Measured wheel speeds of the robot on partially blocked

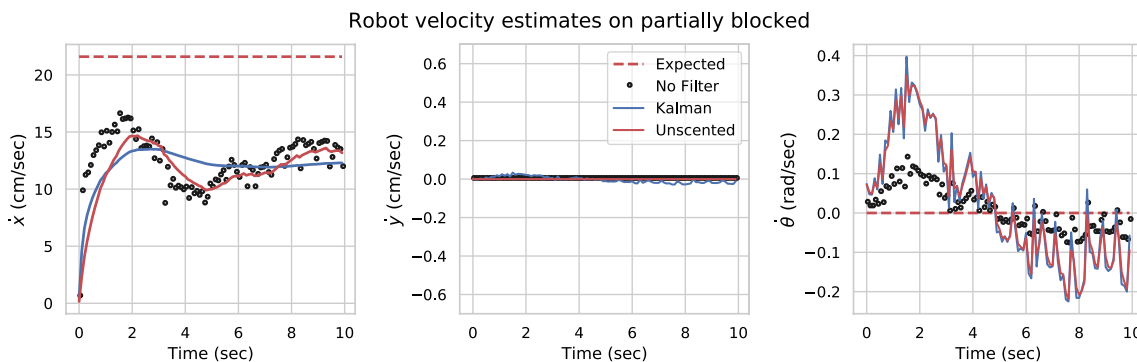


Fig. 17 The estimated velocities of the robot on partially blocked

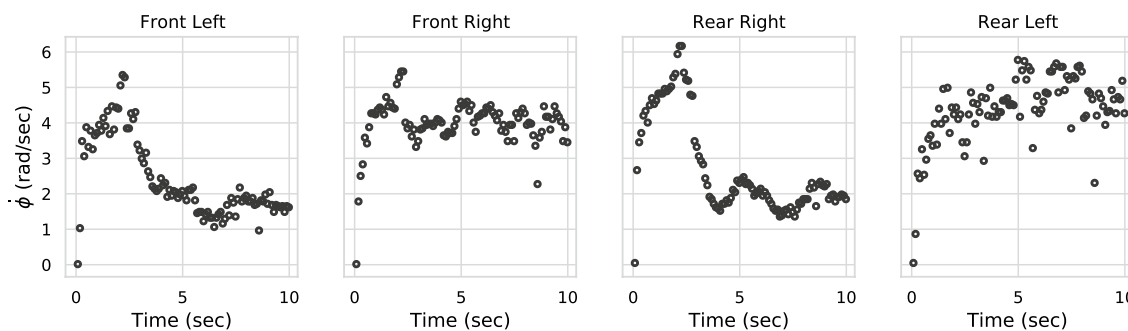


Fig. 18 Measured wheel speeds of the robot on grid holes

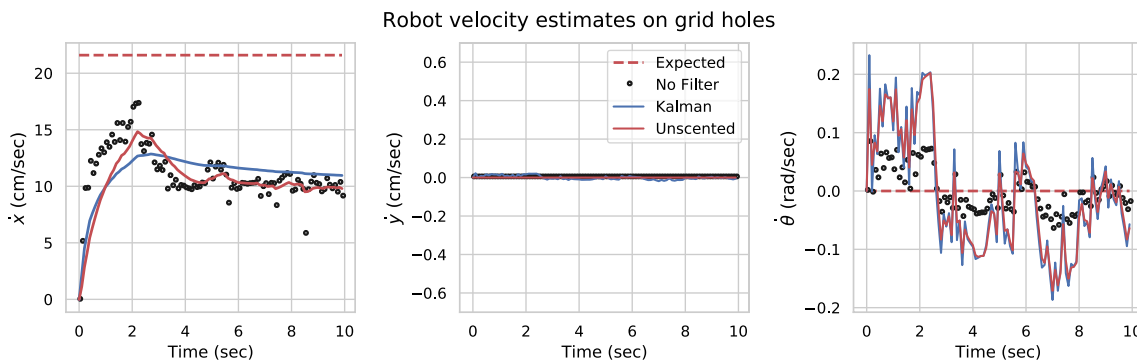


Fig. 19 A comparison of the estimated state velocities (\dot{x} , \dot{y} , and $\dot{\theta}$) of the robot on grid type holes through different algorithms (no filters, Kalman filter, and unscented Kalman filter). The expected velocity is 21.6 cm/s (6.25 rad/s with 6.6 cm of the diameter of the wheel)

the likelihood given measurements. The likelihood can be written:

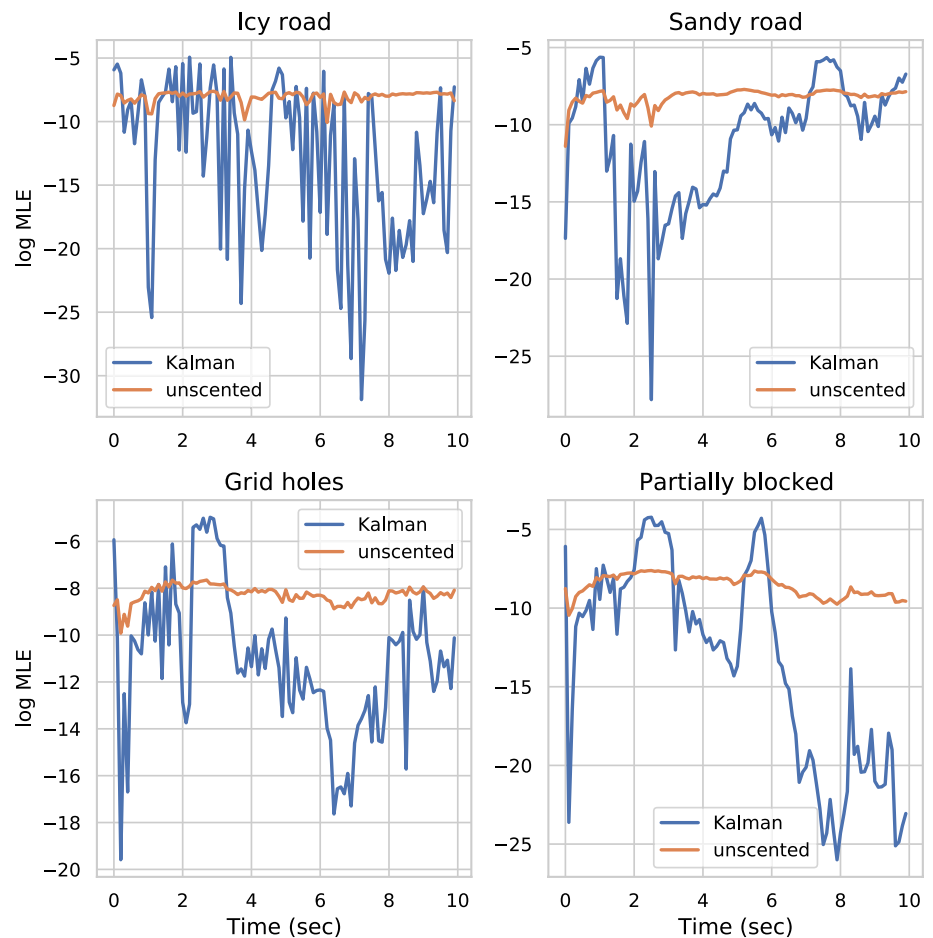
$$\begin{aligned}
 \mathcal{L}(\Theta) &= \log p(\Phi|\Theta) \\
 &= \log \prod_{i=1}^n p(\phi_i|\Theta) \\
 &= \sum_{i=1}^n \log p(\phi_i|\Theta)
 \end{aligned}
 \tag{29}$$

where Θ is the parameter given the measurement $\Phi = \{\phi_1, \dots, \phi_n\}$. Applying maximum likelihood estimation to Eq. (29) to yield

$$\hat{\Theta} = \operatorname{argmax}_{\Theta} \mathcal{L}(\Theta)$$

where $\hat{\Theta}$ is the optimal parameter. In Fig. 20, the trend of MLE values of two algorithms on different road conditions is shown. For all cases, the unscented Kalman filter has

Fig. 20 Comparison of log MLE (maximum likelihood estimation) convergence of Kalman filter and unscented Kalman filter for four different cases: Icy road, sandy road, grid holes, and partially blocked



clearly shown superior performance to those of the Kalman filter. The Kalman filter seems to fluctuate severely per varying measurements, whereas the unscented Kalman filter keeps a relatively steady state for all the cases. This may explain that the unscented Kalman filter is not likely to be affected by measurement uncertainties, guaranteeing more reliable results.

In summary, we carried out more than a thousand test runs over different road conditions. Using Bayes' filtering techniques, we could obtain the estimated velocities of the robot given only measured wheel speeds. In most cases, the kinematic constraints are met well, particularly in the lateral velocity being zero.

7 Conclusion

As the popularity and demand for outdoor wheeled mobile robots are rapidly growing, the lack of fundamental study on kinematics, dynamics, wheel-ground interaction, and terrains can hinder the further development of wheeled mobile

robots. Particularly autonomous wheeled mobile robots should run without human intervention in unknown outdoor environments. This study derived forward and inverse differential kinematic solutions for four-wheeled mobile robots with the fixed-standard wheel system. Furthermore, with a thousand tests performed in outdoor environments, we could build the motion model of the robot. With the help of Bayes' filtering framework, we learned that the unscented Kalman filter outperformed other algorithms, such as the Kalman filter, for the task. The notion of ensemble modeling of kinematics and data-driven models demonstrated in this study can be extended to any wheeled mobile robot as long as the fundamental kinematic solution and measurement data are given. This ensemble approach would incorporate analytical and data-driven methods into a recursive state-space equation to model the robot's motion against outdoor environments. Undoubtedly, further study is needed; for example, we could have employed machine learning to find significant patterns inside seemingly unstructured data of wheel speeds in outdoor environments. We hope to see more research to utilize machine learning in this field soon.

Supplementary Information The online version contains supplementary material available at <https://doi.org/10.1007/s12541-023-00777-9>.

References

- Ortigoza, R. S., Marcelino-Aranda, M., Ortigoza, G. S., Guzman, V. M. H., Molina-Vilchis, M. A., Saldana-Gonzalez, G., Herrera-Lozada, J. C., & Olguin-Carbajal, M. (2012). Wheeled mobile robots: A review. *IEEE Latin America Transactions*, 10(6), 2209–2217.
- Park, H.-G., Ahn, K.-K., Park, M.-K., & Lee, S.-H. (2018). Study on robust lateral controller for differential GPS-based autonomous vehicles. *International Journal of Precision Engineering and Manufacturing*, 19(3), 367–376.
- Alexander, J. C., & Maddocks, J. H. (1989). On the kinematics of wheeled mobile robots. *The International Journal of Robotics Research*, 8(5), 15–27.
- Campion, G., Bastin, G., & Dandrea-Novet, B. (1996). Structural properties and classification of kinematic and dynamic models of wheeled mobile robots. *IEEE Transactions on Robotics and Automation*, 12(1), 47–62.
- Muir, P. F., & Neuman, C. P. (1987). Kinematic modeling of wheeled mobile robots. *Journal of Robotic Systems*, 4(2), 281–340.
- Zhao, Y., & BeMent, S. L. (1992). Kinematics, dynamics and control of wheeled mobile robots. In *Proceedings 1992 IEEE international conference on robotics and automation* (pp. 91–92). IEEE Computer Society.
- Morin, P., & Samson, C. (2008). Motion control of wheeled mobile robots. *Springer Handbook of Robotics*, 1, 799–826.
- Hoang, N.-B., & Kang, H.-J. (2015). Observer-based dynamic parameter identification for wheeled mobile robots. *International Journal of Precision Engineering and Manufacturing*, 16(6), 1085–1093.
- Jung, C., Moon, C., Jung, D., Choi, J. S., & Chung, W. (2014). Design of test track for accurate calibration of two wheel differential mobile robots. *International Journal of Precision Engineering and Manufacturing*, 15(1), 53–61.
- Lee, T.-L., & Wu, C. J. (2003). Fuzzy motion planning of mobile robots in unknown environments. *Journal of Intelligent and Robotic Systems*, 37(2), 177–191.
- Yoo, S. J. (2010). Adaptive tracking control for a class of wheeled mobile robots with unknown skidding and slipping. *IET Control Theory & Applications*, 4(10), 2109–2119.
- Balakrishna, R., & Ghosal, A. (1995). Modeling of slip for wheeled mobile robots. *IEEE Transactions on Robotics and Automation*, 11(1), 126–132.
- Kruse, O., Mukhamejanova, A., & Mercorelli, P. (2022). Super-twisting sliding mode control for differential steering systems in vehicular yaw tracking motion. *Electronics*, 11, 1330.
- Kruse, O., Mukhamejanova, A., & Mercorelli, P. (2022). Differential steering system for vehicular yaw tracking motion with help of sliding mode control. In *2022 23rd international Carpathian control conference (ICCC)* (pp. 58–63).
- Fox, D., Burgard, W., & Thrun, S. (1999). Markov localization for mobile robots in dynamic environments. *Journal of Artificial Intelligence Research*, 11, 391–427.
- Ward, C. C., & Iagnemma, K. (2008). A dynamic-model-based wheel slip detector for mobile robots on outdoor terrain. *IEEE Transactions on Robotics*, 24(4), 821–831.
- Básaca-Preciado, L. C., Sergiyenko, O. Y., Rodríguez-Quinonez, J. C., García, X., Tyrsa, V. V., Rivas-Lopez, M., Hernandez-Balbuena, D., Mercorelli, P., Podrygalo, M., Gurko, A., & Tabakova, I. (2014). Optical 3D laser measurement system for navigation of autonomous mobile robot. *Optics and Lasers in Engineering*, 54, 159–169.
- Mercorelli, P. (2018). Using fuzzy PD controllers for soft motions in a car-like robot. *Advances in Science, Technology and Engineering Systems Journal*, 3(6), 380–390.
- Mercorelli, P. (2018). Fuzzy based control of a nonholonomic car-like robot for drive assistant systems. In *2018 19th international Carpathian control conference (ICCC)* (pp. 434–439).
- Kim, Y.-G., Kwak, J.-H., Hong, D.-H., Kim, I.-H., Shin, D.-H., & An, J. (2012). Autonomous terrain adaptation and user-friendly tele-operation of wheel-track hybrid mobile robot. *International Journal of Precision Engineering and Manufacturing*, 13(10), 1781–1788.
- Lee, H., & Chung, W. (2018). Terrain classification for mobile robots on the basis of support vector data description. *International Journal of Precision Engineering and Manufacturing*, 19(9), 1305–1315.
- Caracciolo, L., De Luca, A., & Iannitti, S. (1999). Trajectory tracking control of a four-wheel differentially driven mobile robot. In *Proceedings 1999 IEEE international conference on robotics and automation (Cat. No. 99CH36288C)* (Vol. 4, pp. 2632–2638). IEEE.
- Jung, D., Seong, J., Moon, C., Jin, J., & Chung, W. (2016). Accurate calibration of systematic errors for car-like mobile robots using experimental orientation errors. *International Journal of Precision Engineering and Manufacturing*, 17(9), 1113–1119.
- Lauria, M., Nadeau, I., Lepage, P., Morin, Y., Giguère, P., Gagnon, F., Létourneau, D., & Michaud, F. (2006). Design and control of a four steered wheeled mobile robot. In *IECON 2006-32nd annual conference on IEEE industrial electronics* (pp. 4020–4025). IEEE.
- Roumeliotis, S. I., & Bekey, G. A. (2000). Bayesian estimation and Kalman filtering: A unified framework for mobile robot localization. In *Proceedings 2000 ICRA. Millennium conference. IEEE international conference on robotics and automation. Symposia proceedings (Cat. No. 00CH37065)* (Vol. 3, pp. 2985–2992). IEEE.
- Moon, C., & Chung, W. (2016). Practical probabilistic trajectory planning scheme based on the rapidly-exploring random trees for two-wheeled mobile robots. *International Journal of Precision Engineering and Manufacturing*, 17(5), 591–596.
- Lefferts, E. J., Landis Markley, F., & Shuster, M. D. (1982). Kalman filtering for spacecraft attitude estimation. *Journal of Guidance, Control, and Dynamics*, 5(5), 417–429.
- Wan, E. A., Van Der Merwe, R., & Haykin, S. (2001). The unscented Kalman filter. *Kalman Filtering and Neural Networks*, 5(2007), 221–280.
- Choi, K.-S., & Lee, S.-G. (2010). Enhanced slam for a mobile robot using extended Kalman filter and neural networks. *International Journal of Precision Engineering and Manufacturing*, 11(2), 255–264.
- Craig, J. J. (2009). *Introduction to robotics: Mechanics and control, 3/E*. Pearson Education India.
- Thrun, S., Burgard, W., & Fox, D. (2006). Probabilistic robotics. *Kybernetes*, 35, 1299–1300.
- Wan, E. A., & Van Der Merwe, R. (2000). The unscented Kalman filter for nonlinear estimation. In *Proceedings of the IEEE 2000 adaptive systems for signal processing, communications, and control symposium (Cat. No. 00EX373)* (pp. 153–158). IEEE.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.



Deok-Keel Choi Professor at the department of mechanical engineering, Dankook university in Korea. His research interest is of application of machine learning unto mechanical engineering.