Improving Position-Time Trajectory Accuracy in Vehicle Stop-and-Go Scenarios by Using a Mobile Robot as a Testbed

Murat Bakirci[1]*, Mecit Cetin[2]

*[1]*Unmanned/Intelligent Systems Lab, Faculty of Aeronautics and Astronautics, Tarsus University, Mersin 33400 Turkey, e-mail: muratbakirci@tarsus.edu.tr*
[2]Transportation Research Institute, Old Dominion University, Norfolk, VA 23528 USA

Abstract: This study sets an example of how mobile robotic vehicles can be used effectively in research on intelligent transportation systems. Especially the stop-and-go mobility seen in heavy traffic conditions was simulated with a mobile robot, and the study is focused on how to obtain distance-time trajectories more accurately under these conditions. System identification tests of the mobile robotic platform, whose kinematic model was developed, were also carried out, and all solutions regarding robot movement were obtained. For the congested traffic simulation, various stop-and-go points are designated on a predetermined straight route segment to mimic behavior of a vehicle in congested traffic. Robot trajectories were obtained under different scenarios by using both GPS data and a kinematic model through the utilization of motor encoders. More accurate and consistent trajectories were achieved by fusing these trajectories with the Extended Kalman Filter. The main contribution of this study is demonstrating how the number of stop-and-go positions can improve the accuracy in estimating the robot/vehicle trajectory. The paper shows how the cumulative error in predicting the trajectories in reduced as the number of stops increases. For example, the trajectory estimated for a scenario involving five stop-and-go points is 94% more accurate than that for the case with a single stop.

Keywords: mobile robot, vehicle trajectory, stop-and-go, intelligent transportation system, positioning accuracy, data fusion.

1. INTRODUCTION

Autonomous robotic systems find a wide range of applications in many areas due to their free movement capacity and recent technological advancements (Luckcuck et al., 2020; Gnambs and Appel, 2019; Chen et al., 2021; Tsitsimpelis et al., 2019). Especially the popularity of wheeled mobile robots (WMR) has increased considerably in parallel with the developments in hardware production (Gao et al., 2018; Ding et al., 2020; Zhai and Song, 2019; Leena and Saju, 2016). Most of the control and simulation studies in the field of robotics are also performed on these systems. The most widely used WMR types are differential drive mobile robots (DDMR) due to their well-defined kinematics and dynamic models. A typical DDMR consists of two main wheels driven by direct-current motors, and a third wheel that balances the entire robotic platform. To change the orientation of a DDMR, a differential velocity between its wheels is applied as an input command, thus, an inevitable wheel slippage takes place. Independent control of DDMR motors provides exclusive maneuvering capability which makes these robots highly efficient, particularly in indoor applications (Martins et al., 2017; Ke et al., 2017; Iqbal et al., 2020; Bakirci and Toptas, 2022).

The widespread use of GPS-based positioning is due to the fact that it offers a global solution and is easily accessible. However, although its use is widespread, it is insufficient especially in critical applications that require high accuracy such as autonomous driving (Yurtsever et al., 2020; Wang et al., 2019). Various methods have been developed to achieve more precise positioning and increase accuracy. One of the most well-known methods is to obtain a better positioning by increasing the signal quality with the use of multiple navigation satellite systems (Yang et al., 2018). Through the use of multiple GNSSs, an elimination of residual range errors on both L1 and L2 ray paths can be achieved which provides more precise correction for the signal delay and phase advance induced by ionosphere (Ovodenko et al., 2015). However, new transportation applications such as automatic guidance and control systems for driverless vehicles will require a higher level of positioning accuracy (Weiskircher et al., 2017). In addition, the fact that dense urban environments cause significant inconsistency in the reception of GPS signals should not be overlooked. Through the integration of GPS and inertial navigation systems (INS), an accurate positioning can be achieved by reducing systematic errors (Zhang et al., 2018). In addition, more consistent positioning can be achieved by estimating and eliminating signal noise by using advanced software in similar integrated systems. Combining camera systems and GPS signals is another method developed to increase positioning accuracy (Min et al., 2019). Moreover, there are also methods using distance measuring instruments (DMI) and LiDAR systems for the same purpose (Paijitprapaporn et al., 2021). Studies in which the positioning error can be reduced by using numerical and statistical methods are also quite remarkable (Marais et al., 2012). In addition, it has been shown that positioning can be significantly improved through image processing techniques (Khan et al., 2020).

Most of the methods mentioned in the previous paragraph are techniques developed within the scope of intelligent transportation systems (ITS), and new ones are being added every day. Adapting robotic systems to ITS applications is a

relatively new concept and researchers have developed scaled testbeds for testing CAVs or Connected and Automated Vehicles (Chalaki et al., 2022). Current studies are generally on individual or fleet control and route planning of mini-vehicles (Veres and Moussa, 2020; Lai et al., 2022). However, mobile robotic systems can play an important role in the development of many important ITS applications such as positioning, heading estimation, vehicle trajectory determination, instantaneous speed estimation, vehicle platooning, cooperative merging optimization, and optimization of operations of signalized intersections among others. As vehicles are becoming more and more like robotic devices with advanced computing and communications capabilities, the methods or findings to be developed with these robotic systems provide significant insights for building real-world CAV applications. Moreover, using actual vehicles in experimental studies is often costly and may pose significant safety risks (Feng et al., 2020). On the other hand, similar experimental studies carried out with mobile robotic systems eliminate these problems and allow more experiments to be completed more effectively. As a result, it is quite reasonable to argue that mobile robotic systems equipped with a wide variety of sensors are critical in developing future ITS and CAV applications.

In this study, the stop-and-go motion of a vehicle in heavy traffic conditions was simulated with a differential drive mobile robot and in this case, the positioning accuracy of a vehicle was tried to be increased. In order to accurately capture the motion of the robotic platform, a kinematic model has been developed and this model has been verified through system identification tests. Extended Kalman Filter (EKF) was used to combine the inaccurate position information obtained by GPS with the information from the motor encoders, and the amount of error in positioning was corrected up to 93%.

2. METHODOLOGY

2.1 Test Robotic Platform

The robotic platform used in the experiments is a mobile robot with differential driving feature and can be used both indoors and outdoors depending on the application type. It has an advanced microcontroller and is equipped with the electronic systems to control the motors that move the wheels. In addition, its operational capacity can be greatly expanded by supporting external electronics such as sensors so that it can be used in different applications. Three microprocessors are embedded in the microcontroller of the mobile robot, which is quite sufficient to control various sensors and motors. The movement of the main wheels is provided by DC motors, that is, the speeds of the motors are proportional to the applied power input. For use in research, the robotic platform has been equipped with several sensors and additional capabilities such as inertial measurement unit (IMU) and GPS. The plain and fully equipped mode of the robot is shown in Fig. 1. The rotational speeds of the motors are measured by motor encoders integrated into the motor shafts. In this way, the distance traveled and the speed of the robot can be controlled in a closed loop. In addition, a Wi-Fi module has been added to the robot to enable telemetry and remote communication.

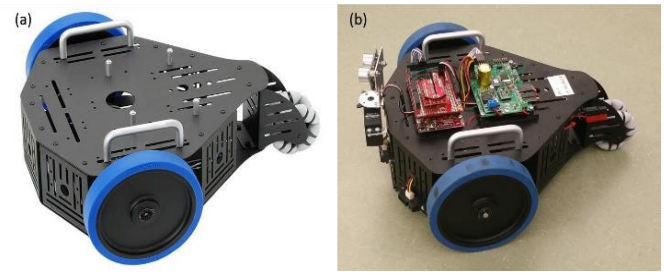

Fig. 1. The test robot platform in plain (a) and fully equipped mode (b).

Thus, data can be transmitted and received to the robot through program control as long as it is in the local area network. All data is available under program control and the user does not need to interface directly to the microcontroller. The main processor board runs the program. The input commands are downloaded to this board for execution. A C/C++ library has also been developed to provide access to all robot features, including motor control, sensor readings, and telemetry.

The motor control board has the ability of controlling the robot motors using two different approaches. In *effort mode*, the board exposes each motor to a fixed percentage of total battery power, as specified by the controlling software. In *velocity mode*, the board uses a closed-loop feedback controller to achieve the desired rotational speed for each motor. Having a velocity control mode is not only useful but also necessary. Due to the differences in the production process of the motors, they may not respond to the same input with the same output. That is, even if the same voltage is applied to both motors, the rotational speeds of the motors may not be exactly the same. The motor control board can adjust the duty cycle using encoders and thus the desired speed can be achieved. All of this is done at the controller level and user never have to interface to this controller directly. Due to this important advantage, the velocity mode was preferred in the experiments.

2.2 Robot Kinematics

Through kinematic analysis, the motion and orientation of the robot can be obtained. The equations of motion of the mobile robot are expressed as

$$x(t) = \int_{t_0}^{t} v(t)\cos\big(\theta(t)\big)\,dt \tag{1}$$

$$y(t) = \int_{t_0}^{t} v(t)\sin\big(\theta(t)\big)\,dt \tag{2}$$

$$\theta(t) = \int_{t_0}^{t} \omega(t)\,dt \tag{3}$$

where x and y are the coordinates of the robot, θ is the orientation, v is the speed of the geometric center, and ω is the angular velocity of the robot. ω and v can be defined as

$$\omega = \frac{v_r - v_l}{d} \tag{4}$$

$$v = \frac{v_r + v_l}{2} \tag{5}$$

v_r and v_l are the speed of the robot's right and left wheels, respectively, and d is the distance between the right and left wheels, i.e., the width of the robot chassis. If (4) and (5) are

used in equations of motion, the following ultimate equations of motion are obtained.

$$x(t) = \int_{t_0}^{t} \left(\frac{v_r+v_l}{2}\right)\cos\big(\theta(t)\big)\,dt \tag{6}$$

$$y(t) = \int_{t_0}^{t} \left(\frac{v_r+v_l}{2}\right)\sin\big(\theta(t)\big)\,dt \tag{7}$$

$$\theta(t) = \int_{t_0}^{t} \left(\frac{v_r-v_l}{d}\right)dt \tag{8}$$

Considering that $t_0 = 0$, integrating (8) and substituting into (6) and (7) yields

$$x(t) = \frac{v_r+v_l}{v_r-v_l}\frac{d}{2}\sin\left(\frac{v_r-v_l}{d}t\right) \tag{9}$$

$$y(t) = -\frac{v_r+v_l}{v_r-v_l}\frac{d}{2}\cos\left(\frac{v_r-v_l}{d}t\right) + \frac{v_r+v_l}{v_r-v_l}\frac{d}{2} \tag{10}$$

Instead of producing a continuous function, the microcontroller processes a series of samples at a given sampling rate. Therefore, it is necessary to use the discrete form. When the discrete integration is applied to the above expressions, the discrete form of the ultimate equation of motion is obtained as follows.

$$(x)_{t+1} = x_t + dt\,\frac{v_r+v_l}{2}\cos\left(\theta_t + \frac{dt\omega_t}{2}\right) \tag{11}$$

$$(y)_{t+1} = y_t + dt\,\frac{v_r+v_l}{2}\sin\left(\theta_t + \frac{dt\omega_t}{2}\right) \tag{12}$$

$$(\theta)_{t+1} = \theta_t + dt\omega_t \tag{13}$$

The robot kinematics can be treated linearly through the Jacobian matrix. Nonlinearity in the position (P) estimation can be obtained with the following Jacobian

$$J = \begin{bmatrix}\frac{\partial P}{\partial x} & \frac{\partial P}{\partial y} & \frac{\partial P}{\partial \theta}\end{bmatrix} = \begin{bmatrix}1 & 0 & -dt\sin\left(\theta + {d\theta}/{2}\right)\\ 0 & 1 & dt\cos\left(\theta + {d\theta}/{2}\right)\\ 0 & 0 & 1\end{bmatrix} \tag{14}$$

In a similar manner, the Jacobian given below can be used to consider the slippage that may occur in the robot wheels.

$$J_s = \begin{bmatrix}\frac{1}{2}\cos\left(\theta+\frac{d\theta}{2}\right) - \frac{dt}{2d}\sin\left(\theta+\frac{d\theta}{2}\right) & \frac{1}{2}\cos\left(\theta+\frac{d\theta}{2}\right) + \frac{dt}{2d}\sin\left(\theta+\frac{d\theta}{2}\right)\\ \frac{1}{2}\sin\left(\theta+\frac{d\theta}{2}\right) + \frac{dt}{2d}\cos\left(\theta+\frac{d\theta}{2}\right) & \frac{1}{2}\sin\left(\theta+\frac{d\theta}{2}\right) - \frac{dt}{2d}\sin\left(\theta+\frac{d\theta}{2}\right)\\ \frac{1}{d} & -\frac{1}{d}\end{bmatrix} \tag{15}$$

The aim of this study is to simulate and analyze the stop-and-go waves that occur in heavy traffic conditions. In such situations, the speed of vehicles in general is low (5 to 15 mph). In these conditions, there is usually no (wheel) slippage to consider, and therefore, there is generally no need for the use of Jacobian. Therefore, 6 m/s, which is close to the maximum speed of the robot, was chosen in the tests to run a realistic scenario. Moreover, for the test robot, Butyl rubber-based wheels with high grip feature have been specially developed to ensure ideal rolling of the wheels. Hence, the possible effects of sliding and slipping, which may arise from the high speed of the robot, have been reduced to negligible levels, thus a consistent model can be obtained even without taking such effects into account.

2.3 System Identification

Before carrying out driving simulations with the mobile robot, it is important to investigate how the motors that move the wheels work in practice. For this, a set of data-driven system identification experiments were performed in the laboratory environment. This process can be quite complicated due to the fact that the system is not linear and the data obtained contains high amount of noise. However, the assumption of a linear system greatly simplifies this problem in terms of model development. Thus, the robotic platform can be modeled directly by inferring some key identification parameters.

The system input is the command given to the motor that turns the wheel, while the output is the measured wheel speed. A first order system is expressed as follows when there is no system input.

$$x(t) = x(0)e^{(-t/\tau)} \tag{16}$$

where τ and $x(0)$ indicate time constant and the initial value of the system respectively. In addition, a system starting from state zero and having a specific step input u is expressed as follows.

$$x(t) = Gu\big[1 - e^{(-t/\tau)}\big] \tag{17}$$

where G is the system gain. Consequently, a first-order system can be modeled by measuring the response of a system for a given input, and measuring the system gain for steady-state input.

To move the wheels with the desired speed or effort rate, step commands are sent to the mobile robot and the required data is collected at a frequency of 10 Hz. For each test, the robot was allowed to stop for the first two seconds, move in line with the command for the next two seconds, and then stop again. The system inputs, the output of these inputs, the wheel speeds, and the encoder data are recorded in a matrix at 10 Hz. After all the tests are completed, the obtained information is turned into a data file and then sent back to the workstation through the ftp server.

The results of the system identification tests performed are given in Table 1. The tests were conducted by keeping one wheel stationary and examining the movement of the other. Using both *effort mode* and *velocity mode*, how both wheels respond to different inputs was investigated. "E-L" in the table means that the left wheel is moved using the effort mode, and "V-L" means that the left wheel is moved again using the velocity mode. For example, since only the left wheel will move in V-L mode, the robot will pivot around the right wheel, which remains stationary. The same tests were also performed for the right wheel and are indicated in the table with as (E-R) and (V-R) modes.

Table 1. Identification test results.

Test	Mode	u	G	$\tau \leftarrow 0$	$\tau \rightarrow 0$
1	E-L	10%	10.565	0.313	0.235
2	E-L	30%	13,437	0.428	0.316
3	E-L	50%	14,776	0.388	0.369
4	E-L	70%	14.565	0.411	0.305
5	E-R	10%	9.821	0.334	0.247
6	E-R	30%	13,556	0.346	0.352
7	E-R	50%	14,713	0.434	0.333

8	E-R	70%	14.409	0.420	0.328
9	V-L	1 m/s	1.035	0.237	0.219
10	V-R	1 m/s	1.075	0.254	0.223
11	V-L	2 m/s	1.096	0.374	0.336
12	V-R	2 m/s	1.020	0.355	0.337
13	V-L	3 m/s	1.080	0.361	0.348
14	V-R	3 m/s	1.024	0.395	0.389
15	V-L	4 m/s	1.025	0.365	0.373
16	V-R	4 m/s	1.021	0.346	0.344
17	V-L	5 m/s	1.010	0.379	0.337
18	V-R	5 m/s	1.082	0.401	0.375
19	V-L	6 m/s	1.042	0.423	0.387
20	V-R	6 m/s	1.056	0.431	0.384

It can be seen that the gain values in the table show notable differences in the response of the wheel motors to different inputs in the effort mode. In the velocity mode, however, this difference is quite small and the outputs produced in response to all inputs are approximately 1, as expected. In the 19th and 20th sets of experiments, the responses of the wheels to the 6m/s speed command were investigated, which is the speed to be used in the positioning experiment. The gain amounts of the wheels in each experiment are very close to each other. However, it should not be overlooked that there is little difference, which can cause significant errors in the positioning experiment. For example, in long-term locomotion tests, this small difference can grow over time as the error is added up. The last two columns in the table represent the time constant for the step transition. In other words, it refers to the time it takes to reach the steady-state velocity and vice versa. It is seen that these values are largely consistent and vary at most around 10% for the same input, which is a common situation.

2.4 Data Fusion

A standout amongst the most generally executed sensor fusion strategies utilized as a part of both autonomous systems and navigation, is the Extended Kalman Filter (EKF). The EKF is portrayed as an ideal estimator for nonlinear dynamic systems. Its recursive properties and capacity to make predictions of the systems make it to a mainstream scientific procedure. The predictions of the system are conceivable when the error covariances are zero mean, Gaussian distributed and accepted stationary with time.

The EKF's capacity to make predictions can be separated into two essential stages: the prediction and state update. The prediction step utilizes the dynamic model of the system and process covariance error to make a prediction of the system. The state update step, alongside the state covariance error, are utilized to peruse the system's state (particular attributes, for example, velocity, acceleration, and so forth). In case of comparing prediction and state update, fundamental changes can be decided for modification in the following prediction cycle. Assuming that a linearized system can be modeled utilizing state space equations with some measure of process noise error (e.g. a robot going along a flawed road). The measurements taken will likewise have some measure of error (sensor errors).

The Extended Kalman filter (EKF) is an extension of the Kalman filter for nonlinear dynamical systems. Considering the following nonlinear system with covariance $\mathcal{P}$

$$\dot{\Gamma}(t) = \mathcal{F}(\Gamma, u, \omega, t) \tag{18}$$

where $\mathcal{F} \in \mathbb{R}^{n\times n}$ is an arbitrary function, $\Gamma(t)$ is the state vector $u(t)$ is the system input, and ω is the unknown zero mean white Gaussian disturbance. A linear relationship assumption can be made between $\Gamma(t)$ and system measurements, μ_k, which contain white noise, ν_k, with covariance $\mathcal{Q}$ as follows

$$\mu_k = a\Gamma_k + \nu_k \tag{19}$$

Considering the nonlinear equations of motion and the measurements of the system, the minimum variance estimate of the state vector is computed with the EKF. Here, the primary functionality of the EKF is to linearize the nonlinear system using Taylor series expansion and compute the Kalman Filter (KF) estimate of the resulting linearized system. The two main steps of the EKF algorithm mentioned above, namely prediction and update, can be more broadly expressed as follows.

Applying the Taylor expansion to the arbitrary vector function $\mathcal{F}$ about nominal state Γ_0 yields

$$\dot{\Gamma}(t) = \mathcal{F}(\Gamma_0, u_0, \omega_0) + \frac{\partial \mathcal{F}}{\partial \Gamma}\Big|_{\Gamma_0} [\Gamma(t) - \Gamma_0] + \frac{\partial \mathcal{F}}{\partial u}\Big|_{u_0} [u(t) - u_0] + \frac{\partial \mathcal{F}}{\partial \omega}\Big|_{\omega_0} [\omega(t) - \omega_0] + \Theta \tag{20}$$

where Θ indicates higher order terms. Here, an assumption can be made that the mean state, $\bar{\Gamma}(t)$ is known. What remains is to propagate the state $\Gamma(t)$ at the nominal state such as

$$\Gamma_0 = \bar{\Gamma}(t)\,, u_0(t) = u(t)\,, \omega_0(t) = 0 \tag{21}$$

Applying these to (20) and neglecting Θ yields

$$\dot{\Gamma}(t) = \mathcal{F}(\bar{\Gamma}(t), u(t)) + \frac{\partial \mathcal{F}}{\partial \Gamma}\Big|_{\bar{\Gamma}} [\Gamma(t) - \bar{\Gamma}(t)] + \frac{\partial \mathcal{F}}{\partial \omega}\Big|_{\omega_0} \omega(t) \tag{22}$$

The expected value of both sides results

$$\dot{\bar{\Gamma}}(t) = \bar{\mathcal{F}}(\bar{\Gamma}(t), u(t)) \tag{23}$$

Consequently, first order Taylor series approximation of state equations and the state estimate are acquired. A differential expression for the estimation error covariance matrix can be found using (18) and (23) as follows

$$\mathcal{r}(t) = E\{[\Gamma(t) - \bar{\Gamma}(t)][\Gamma(t) - \bar{\Gamma}(t)]^T\} \tag{24}$$

$$\dot{\mathcal{r}}(t) = E[\Gamma(t)\mathcal{F}^T] - \bar{\Gamma}(t)\bar{\mathcal{F}}^T + E[\mathcal{F}\Gamma(t)^T] - \bar{\Gamma}(t)\bar{\mathcal{F}}^T \tag{25}$$

Since first and third terms on the right hand side of (25) cannot be computed, an approximation needs to be made. Utilizing (22) by substituting the expansion of the state and computing expectation operations yields

$$\dot{\mathcal{r}}(t) = \Phi(t)\mathcal{r}(t) + \mathcal{r}(t)\Phi(t)^T + \Omega\mathcal{p}\Omega^T \tag{26}$$

where $\Phi(t)$ and Ω are defined as

$$\Phi = \left.\frac{\partial \mathcal{F}(\Gamma,u,\omega,t)}{\partial \Gamma}\right|_{\Gamma(t)=\bar{\Gamma}(t)} \quad (27)$$

$$\Omega = \left.\frac{\partial \mathcal{F}(\Gamma,u,\omega,t)}{\partial \omega}\right|_{\omega(t)=0} \quad (28)$$

(23) and (26) form the prediction step of the EKF. Thus, the state estimate and state error covariance are proceeded to the next step through numerical integration of these equations, and get

$$\bar{\Gamma}(t+1) = \bar{\Gamma}_{k+1}^{-} \quad \text{and} \quad r(t+1) = r_{k+1}^{-} \quad (29)$$

Now, the 'update' step of the EKF can be formed. It is assumed that the measurement at $k+1$ has already handled above. While the system described in (22) evolves according to the predicted state estimate, it remains unchanged in the measurement phase. Thus, the state estimate is updated similar to the KF as follows

$$\bar{\Gamma}_{k+1}^{+} = \bar{\Gamma}_{k+1}^{-} + K_{k+1}(\mu_{k+1} - a\bar{\Gamma}_{k+1}^{-}) \quad (30)$$

where K_{k+1} is the Kalman gain and is computed as

$$K_{k+1} = r_{k+1}^{-} a^{T}(a r_{k+1}^{-} a^{T} + q)^{-1} \quad (31)$$

and the updated state error covariance, r_{k+1}^{+}, is also computed as

$$r_{k+1}^{+} = (I - K_{k+1}a) r_{k+1}^{-} (I - K_{k+1}a)^{T} + K_{k+1} q K_{k+1}^{T} \quad (32)$$

Using the initial state estimate, $\bar{\Gamma}_0$, and the state error covariance, r_0, the filtering process is initialized as follows

$$\bar{\Gamma}_0 = E(\bar{\Gamma}_0) \quad (33)$$

$$r_0 = E[(\Gamma_0 - \bar{\Gamma}_0)(\Gamma_0 - \bar{\Gamma}_0)^{T}] \quad (34)$$

The process of realizing the new speed estimation through the fusion of the speed data collected is shown with the flowchart in Fig. 2. The speed data obtained through GPS and the model are included in the algorithm after the initial estimation (33-34). Then, calculating the gain with (31), the speed estimate is updated utilizing these speed inputs. A speed projection is made by updating the covariance (32) in each estimation cycle. Thus, a new speed estimate (30) is updated in each cycle.

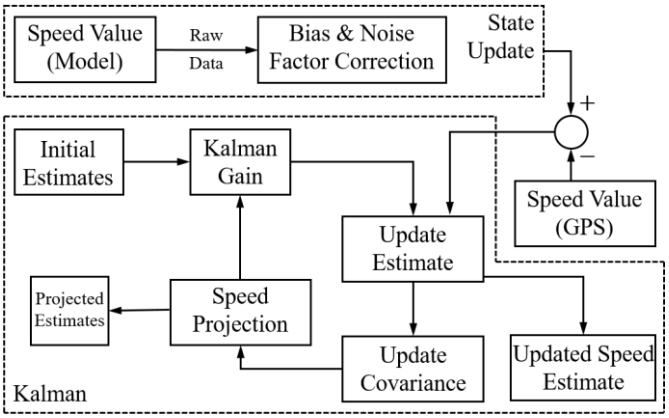

Fig. 2. Flowchart of data fusion strategy.

2.5 Inadequacy of GPS-based Positioning

GPS-based positioning is a versatile and valuable technology that has become an integral part of many transportation systems due to its low cost, continuous availability, and global coverage. It is used for a range of purposes, including navigation, route planning, vehicle tracking, and safety and security. While GPS-based positioning is a useful technology for many transportation applications, its accuracy may not always be sufficient to support operations that require high precision. In recent years, various techniques have been developed to improve GPS accuracy, such as differential GPS (DGPS), Real-Time Kinematic (RTK) GPS, and carrier phase tracking. These techniques can improve the accuracy of GPS-based positioning, but they may still not be sufficient for applications that require very high precision, such as certain scientific and military applications. It is worth noting that even with the use of these additional technologies, achieving high accuracy can still be challenging, as factors such as signal interference, atmospheric conditions, and environmental obstacles can affect positioning accuracy. Therefore, achieving high-precision positioning often requires a combination of multiple technologies and advanced techniques.

Using multiple Global Navigation Satellite Systems (GNSS) systems can also improve positioning accuracy in challenging environments where signal reception may be degraded, such as in urban canyons, forests, or near tall buildings. This is because different GNSS systems may have different satellite configurations and signal strengths, which can improve the overall performance of the positioning solution. The use of multiple GNSS systems provide more precise correction for errors induced by the ionosphere, including the signal delay and phase advance. These ionospheric errors can cause inaccuracies in the positioning solution, particularly for signals transmitted on the L1 and L2 frequencies. In addition to improving ionospheric corrections, using multiple GNSS systems can also improve the robustness and reliability of positioning solutions by providing redundancy in case of satellite failures or outages. This is particularly important for safety-critical applications such as aviation, where reliable and accurate positioning information is essential.

New transportation applications such as automatic guide and control systems for driverless vehicles will require a higher level of positioning accuracy than traditional GNSS-based systems can provide. Future autonomous and intelligent transportation systems will rely on highly accurate and reliable positioning information to navigate safely and efficiently, avoid collisions, and make real-time decisions based on their surroundings. This requires not only high-precision positioning technology, but also robust and reliable communication and control systems that can process and integrate data from multiple sources in real-time. To support these emerging transportation applications and services, there is a need for continued innovation and development in positioning technologies and techniques, including the use of advanced GNSS augmentation systems, sensor fusion approaches, and high-precision mapping and localization techniques.

The test robot uses the GYGPSV1 NEO-8M GPS module for navigation. This 72-channel module operates with an input supply voltage in the 3-5 volt range, and has a navigation sensitivity of 147 dBm. Moreover, it can receive up to 3 GNSS concurrently and supports all satellite augmentation systems.

However, this module, similar to other MEMS sensors, can produce noisy positioning data due to a variety of factors, including the quality of the received signals, the presence of environmental interference, and the inherent limitations of the receiver hardware and software as mentioned above. Although it is stated that it provides position accuracy 2 to 2.5 meters according to the manufacturer's catalog, with our preliminary tests, it was observed that this positioning error can be higher depending on the conditions of the environment. It should be noted that in this study, the traveled distance-time variation of the robot is investigated. In other words, cumulative distance curves with respect to time are obtained. GPS data with different amount of errors in each position determination will cause an additional inconsistency in the formation of this cumulative curve. Thus, the cumulative increment of distance will be fluctuating rather than linear. One possible solution to reducing or completely eliminating these errors in time-distance curves is the utilization of odometer data of the robot. By utilizing sensor fusion and incorporating data from the robot's odometer sensor, it may be possible to obtain more accurate time-distance curves.

2.6 Data Collection

A program is written that sends step commands to the robot to move the mobile robotic vehicle at the required speed on the determined route. It is possible to receive the motion test data at different data transmission rates, and 5 Hz is preferred for this study. In addition, inputs for each wheel, wheel speeds and odometer numbers for each wheel are written into a matrix at 10 Hz. The counter was used to capture the required information from the test result and to increment the rows of the data file as the test continued. When the test is finished, the collected information is turned into a data packet and returned to the control computer via ftp.

While OBD sourced robot trajectory is obtained with (11-13), GPS sourced robot trajectory is calculated using collected latitude and longitude information as follows.

$$\alpha = \left[\sin^2\left(\frac{\Delta\varphi}{2}\right)\right] + \left[\cos(\varphi_1)\cos(\varphi_2)\sin^2\left(\frac{\Delta\lambda}{2}\right)\right] \quad (35)$$

$$\beta = 2\text{atan2}\left(\sqrt{\alpha}, \sqrt{1-\alpha}\right) \quad (36)$$

$$d_{i\to i+1} = R \cdot \beta \quad (37)$$

where φ is latitude, λ is longitude, and R is the Earth's Radius.

3. EXPERIMENT and RESULTS

The route used for the experiments is a straight walking trail on the campus and it includes some environmental interferences, such as tall trees, which may reduce localization accuracy by blocking GPS signals. The total length between start and end points is approximately 300 meters and five stop-and-go points were designated. The experiments were designed to identify GPS-sourced localization error by processing dynamic travel data collected by the GPS and motor encoders of the robot. When the robot is in velocity mode, the propulsion board implements a control loop that seeks to track the desired velocity for each motor. Like all controllers, it is not perfect but provides a consistent reference for movement that uses the encoders to improve the accuracy of the robot movement. Typically, the larger error occurs during situations requiring large acceleration and deceleration, as the velocity may temporarily overshoot the desired value. The robot motors provide a top speed of about 6.9 m/s and values greater than that will not be achieved, yielding full speed. For safety and equipment longevity, the propulsion board limits the maximum speed to about 6.5 m/s. In the experiments, both motors were commanded to move at a speed of 6.0 m/s. The GPS-sourced robot trajectory is computed by utilizing the collected latitude and longitude information as in (35-37). Similarly, the odometer-sourced robot trajectory is also computed through (11-13).

The input given to the robot and the resulting speed outputs, obtained by the GPS (V_{GPS}) and kinematic model (V_{KM}), are shown in Fig. 3. It can be observed that the GPS-based velocity profile differs from the input values and can be inaccurate up to twice the velocity values. The primary reason for this is the high amount of error in the positioning information obtained by GPS. On the other hand, it is clear that the speed profile obtained through the kinematic model, that is based on the values read from the motor encoders, is relatively closer to the

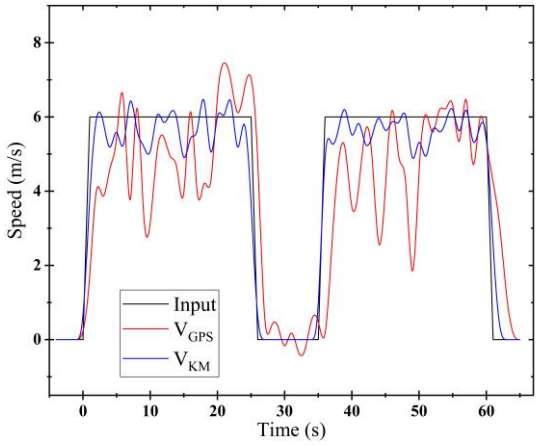

Fig. 3. Input command, and corresponding speed output profiles obtained via GPS and kinematic model.

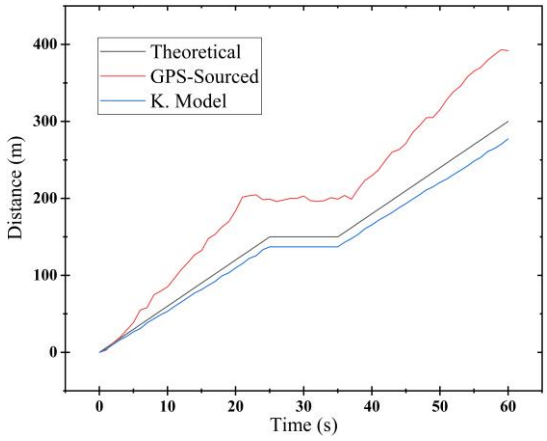

Fig. 4. Robot trajectories in a bounded time-space (t-s) region obtained via GPS and kinematic model, and comparison with the theoretical trajectory.

system input. It takes approximately 1.15 seconds for the robot to reach the commanded speed of 6 m/s from the stationary state, that is, the zero-speed state, which is more than the expected threshold value of 0.8 seconds based on the motor

characteristics. When the robot reaches the intermediate stop, its speed drops to zero in about the same time. However, the second time the robot is commanded to move, it takes much longer than expected to return to steady-state speed. The robot, which continues to move by making oscillations around the commanded speed value of 6 m/s, and when it receives the final stop command, it performs well and stops in a time below the threshold value of 0.8 seconds. It should be noted here that the main reasons why the actual speed profile of the robot oscillates around the system input is due to the slightly different motor gains and ground conditions.

Fig. 4 illustrates the robot's trajectory in a bounded time-space (t-s) region obtained through GPS and kinematic model. The obtained trajectories were also compared with the theoretical trajectory. The theoretical trajectory is generated for the case where the robot moves at exactly 6 m/s speed, which is the system input, and the gains of both motors are 1. Besides, it is assumed that the ground conditions are optimum and there are no effects such as drift and wheel slip. Similar to the speed profile, the robot trajectory obtained by GPS differs significantly from the theoretical one due to the high error in the GPS data. It should be noted here that since each subsequent data in this trajectory obtained by GPS includes the previous error, the total error increases rapidly. This is the main reason why this obtained trajectory differs largely from the theoretical one. On the other hand, the trajectory obtained through the kinematic model based on the motor encoders is relatively closer to the theoretical trajectory. This trajectory, which overlaps with the theoretical trajectory in the first several seconds of the movement, later differs from the theoretical trajectory due to various systematic errors. Although there is a small amount of error due to different motor gains and ground conditions, this error is approximately 14 m at its maximum. Therefore, although it may seem like an insignificant error on the graph, it needs to be fixed. This amount of error is not within the acceptable range, especially in applications such as autonomous driving.

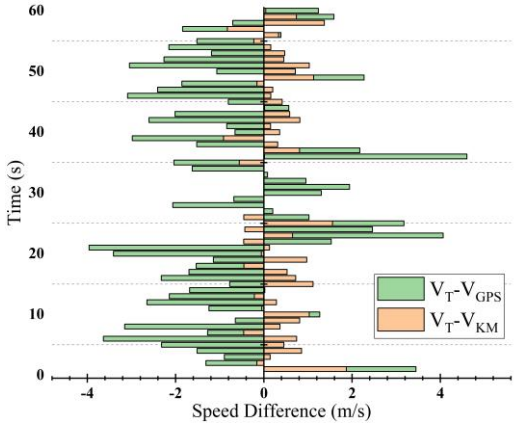

Fig. 5. Differences between the speed profiles obtained with the GPS and the kinematic model and the input profile.

The differences of the speed data obtained by the GPS and kinematic model during the locomotion of the robot from the input (theoretical velocity profile) are shown in Fig. 5. The high amount of error in the speed data obtained by the GPS can be clearly seen in this graph. The average error of the GPS-based speed profile is 2.27 m/s, and the minimum and maximum error amounts are 0.11 m/s and 4.17 m/s, respectively. The similarity ratio of the two profiles was analyzed with the root-mean-square error (RMSE) value and found to be 3.815. On the other hand, the average error of the speed values obtained through the kinematic model is 0.46 m/s and is approximately seven times less than that obtained with the GPS. The minimum and maximum error amounts are 0 m/s and 1.92 m/s, respectively. The RMSE value is 0.629, which is one-sixth the RMSE value of the speed profile obtained with GPS.

The trajectory obtained by fusing the GPS and kinematic model trajectories through the Kalman filter (KF) is given in Fig. 6. As Fig. 6 shows, this trajectory is quite compatible with the theoretical model and overlaps in most places. The correlation coefficient, r, between the trajectory obtained by KF and the theoretical one is 0.9977. It should be noted that shortly before the first and final stop points, the trajectory starts to diverge from the theoretical trajectory. To be more precise, there are notable deviations from the theoretical trajectory between $t = 18 - 25$ and $t = 45 - 60$ intervals. This is because although positioning error is largely eliminated through KF, the small amount of error that cannot be eliminated are compounded, causing increasingly larger error in each subsequent data point. The intermediate stop point, on the other hand, acts as a reference point and ensures that the experimentally obtained and theoretical speeds are equalized, so the error decreases to zero at this point. After the robot starts

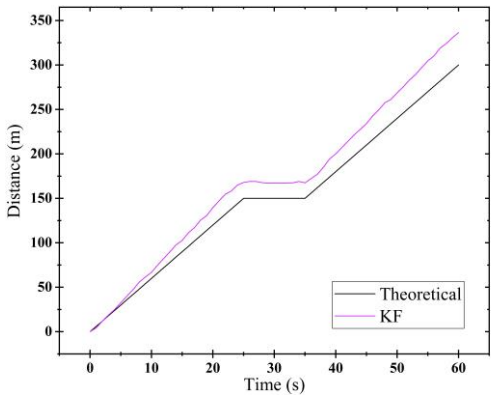

Fig. 6. New robot trajectory obtained by Kalman filtering (KF).

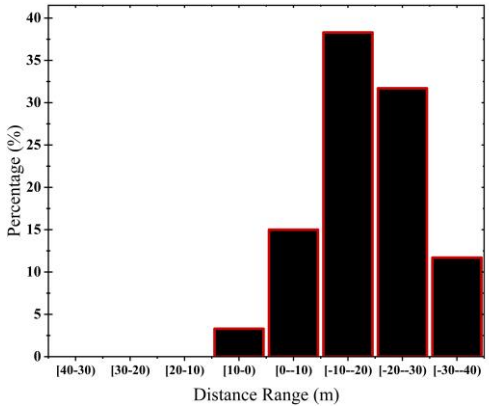

Fig. 7. Histogram representation of the distance differences between the trajectory obtained by KF and the theoretical trajectory.

moving again, each position is determined with a small error and these errors are added to the next data points.

In Fig. 7, the comparison of the two trajectories in Fig. 6 is shown as a histogram. According to the histogram, 19% of the distance differences between the trajectory obtained with KF and the theoretical one are 10 meters or less. Larger distance errors, on the other hand, result from errors being superimposed and transferred to the next data point, as mentioned above.

In order to better measure the reliability of the proposed method, 15 more tests were performed and the results of these

Table 2. Statistical comparison of the theoretical trajectory and the trajectory obtained by KF.

	r	MAE	RMSE	NRMSE
Trial 0	0.9977	-18.1720	21.3070	0.0124
Trial 1	0.9968	-17.8501	21.2878	0.0123
Trial 2	0.9971	-17.9472	21.2944	0.0123
Trial 3	0.9785	-24.0935	20.8994	0.0121
Trial 4	0.9852	-21.5685	21.0404	0.0122
Trial 5	0.9975	-17.9580	21.3603	0.0123
Trial 6	0.9864	-19.8307	21.0379	0.0122
Trial 7	0.9897	-18.0263	21.1365	0.0123
Trial 8	0.9908	-18.0464	22.1821	0.0123
Trial 9	0.9925	-18.0773	21.3598	0.0123
Trial 10	0.9913	-18.0556	21.1707	0.0123
Trial 11	0.9853	-17.9466	21.0057	0.0122
Trial 12	0.9697	-17.6631	20.5059	0.0120
Trial 13	0.9859	-17.8935	21.0552	0.0122
Trial 14	0.9960	-18.4097	20.8891	0.0123
Trial 15	0.9874	-17.4462	21.5285	0.0122

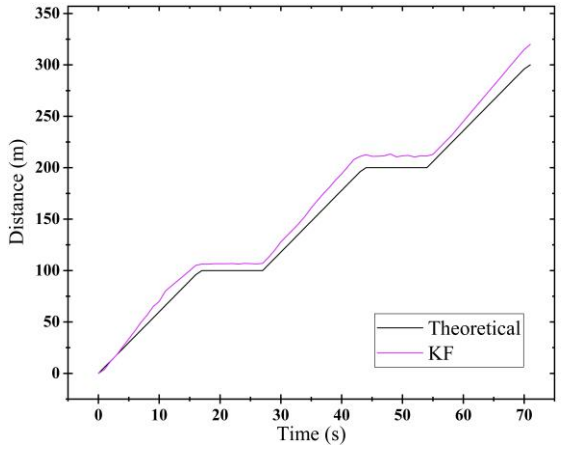

Fig. 8. Robot trajectory obtained by KF as a result of the motion test including 2 stop-and-go points.

tests are compiled in Table 2. Various statistical comparisons of the ground truth and trajectories obtained as a results of GPS and odometer outputs fused through KF are given in the table. Pearson's correlation coefficient, r, mean absolute error

(MAE), root mean square error (RMSE), and normalized root mean square error (NRMSE) was chosen as the benchmark index. Trial 0 indicates the results of the first trial, which is also graphically explained above. All other tests were also performed under the same conditions as Trial 0. As can be seen from the table, the test results are generally quite close to each other and this shows the consistency of the applied method. Although there is a significant difference only in Trial 3 and Trial 4, these differences correspond to an error of up to 12%.

In the second set of experiments, two stop-and-go points were designated, that is, the robot stopped twice along the entire route. As in the previous experiment, the robot was given the command to stop for 10 seconds at these stop-and-go points and then continues on its way with a speed of 6m/s. As a result of this experiment, the new trajectory obtained after fusing the trajectories obtained with GPS and kinematic model through KF is shown in Fig. 8. It is clearly seen that a better result is obtained when compared to the previous case, namely the experiment with a single stop-and-go point. When statistical comparison is made, a trajectory that is approximately 50% closer to the theoretical trajectory is obtained compared to the previous case. It should also be noted that, when the trajectory obtained is carefully examined, it is relatively closer to the theoretical trajectory at the beginning of the motion, while it deviates from the theoretical trajectory as the motion continues. When the robot starts to move again after the stopping point, it is seen that the same situation repeats. Therefore, motion tests with more stops should be performed to examine the dependence of positioning accuracy on the number of stop-and-go points.

The results of the motion tests performed by selecting three and five stop-and-go points are given in Fig. 9 and Fig. 10, respectively. As expected, the test trajectory with three stop-and-go points is relatively much closer to the actual trajectory than the previous trajectory. An additional 25% correction was achieved compared to the trajectory with two stop-and-go points. On the other hand, for the test with five stop-and-go points, the best trajectory was obtained by providing an increase of approximately 40% in trajectory accuracy compared to the test containing three stop-and-go points.

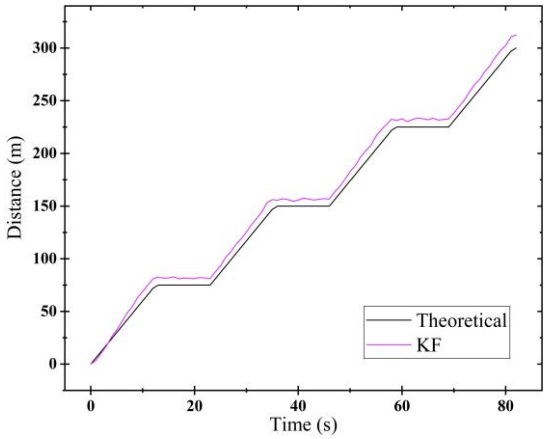

Fig. 9. Robot trajectory obtained by KF as a result of the motion test including 3 stop-and-go points.

When all test results are evaluated together, it is observed that the accuracy of the trajectory obtained with KF increases exponentially as the stop-and-go point increases. When the test results with 1, 2, 3, and 5 stop-and-go points are compared with the statistical indicators, the results in Fig. 11 are obtained. It is also seen numerically that as the number of stop-

and-go points increases, the trajectory accuracy increases exponentially.

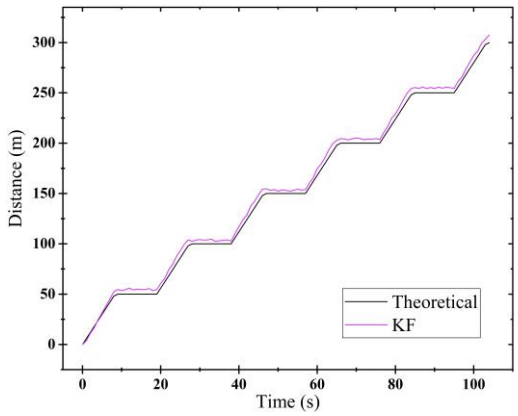

Fig. 10. Robot trajectory obtained by KF as a result of the motion test including 5 stop-and-go points.

Again, as can be seen from the figure, as the number of stop-and-go points increases, the correlation between the theoretical and KF-obtained trajectories increases. In addition, the error amounts obtained with MAE, RMSE, and NRMSE also decrease exponentially as the number of stop-and-go points increases. This can be attributed to the fact that each stop-and-go point acts as a reference point, eliminating the total error added to each other to a large extent before starting the ongoing motion, giving the result as if the robot were starting the motion from the very beginning. So much so that this fact is clearly visible from each trajectory. The trajectory of the robot, which started to move again after full stop, is very close to the theoretical trajectory at the beginning, while the deviations from the theoretical trajectory increase as

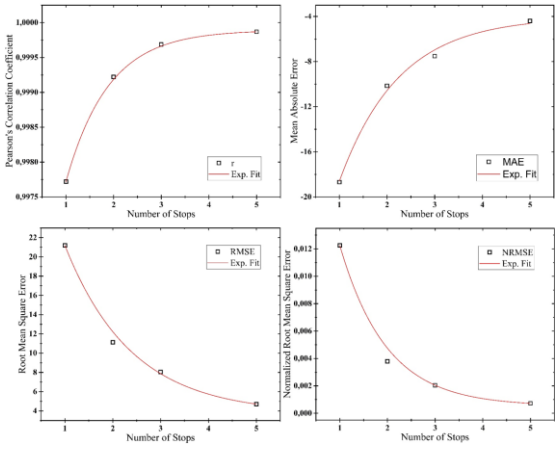

Fig. 11. Representation of the dependence of orbital accuracy on the number of stop-and-go points with statistical indicators.

the motion continues. From this it is deduced that as the number of stop-and-go increases, a more accurate trajectory can be obtained.

4. DISCUSSION

Simulating traffic congestion is important for studying the effectiveness of various intelligent transportation systems (ITS). Mobile robots could provide an effective mechanisms to test different sources of localization and sensor errors. Such approaches, as presented in this study, can provide researchers with valuable insights into the systematic errors and help better understand traffic flow data in different scenarios. Moreover, this allows for controlled experiments that can be repeated in a consistent manner, providing reliable data for analysis. The robot can be programmed to mimic the behavior of human drivers, such as acceleration and braking patterns, lane changes, and reaction times. Additionally, the robot can be used to test different ITS technologies, such as adaptive cruise control and lane departure warning systems, in a safe and controlled environment. In such studies, a challenge may arise when the motors that rotate the right and left wheels of the robot provide slightly different gains. This can cause the robot to exhibit uneven or unpredictable behavior, such as drifting to one side or turning at a slightly different rate on one side compared to the other. This problem was observed by system identification tests. To address this issue, one possible solution is to calibrate the motors to ensure they provide equal gains. This can be done by adjusting the power input to each motor until they produce the same output torque, which can be measured using motor encoders. Another solution is to incorporate a closed-loop control system that can detect and compensate for any differences in the motor gains. This is achieved by using sensors to monitor the speed and position of the wheels and adjusting the power input to each motor in real-time to ensure they rotate at the same speed.

It is worth mentioning a critical benefit of this work with a mobile robot. Traditional traffic control systems have limitations in their ability to monitor traffic density and respond to changing traffic conditions. These systems are often fixed in place and can only monitor a specific section of road or intersection, which can make it difficult to detect and respond to traffic waves or other disruptions. Moreover, for the same reason, these systems can inspect these waves only in limited time intervals. Another disadvantage is the high installation cost associated with fixed infrastructure. Fixed traffic signals and other fixed actuators require significant capital investment for installation, maintenance, and replacement. This can be a barrier to implementing effective traffic management systems in areas with limited resources or in situations where the cost of installing and maintaining fixed infrastructure is prohibitive. The test robot in this study can be considered as the simulation of a fully autonomous vehicle or a probe vehicle equipped with internal sensors. Thus, the proposed method is highly cost-effective, and eliminates the problem of collecting data at a fixed location and in a limited interval. As data can be collected while the vehicle/robot is in motion, real-time travel data can be obtained continuously, thus providing a deeper understanding of the dynamics of stop-and-go waves and traffic flow in general.

5. CONCLUSIONS

In this study, a simulation of stop-and-go mobility, which is frequently seen in traffic, was carried out with a mobile robot, and the robot's trajectory was obtained with both the kinematic model and GPS. Relatively accurate robot trajectories were obtained by fusing these distance-time trajectories. The dependence of the accuracy of the estimated trajectories on the number of stop-and-go points along the route was investigated. It has been observed that more accurate trajectories are obtained as the number of stop-and-go points increases. The

trajectory obtained with the test completed with 5 stop-and-go points is 94% more accurate than the trajectory obtained with the test completed with a single stop. This is because the cumulative error is reset every time the robot stops. Thus, the magnitude of cumulative errors added to each other at each collected data point decreases. In other words, with the convergence of the data obtained with GPS and kinematic model at stop-and-go points, the positioning error at these points is greatly reduced. Moreover, it has been shown that the use of robotic platforms for driving tests in ITS studies offers a range of benefits, including increased efficiency, repeatability, and control, as well as the ability to test a wide range of scenarios and technologies. Unlike conventional methods, the proposed method is highly cost-effective, and eliminates the problem of collecting data at a fixed location and in a limited time interval.

REFERENCES

Bakirci, M., Toptas, B. (2022). Kinematics and autoregressive model analysis of a differential drive mobile robot. *IEEE 4th International Congress on Human-Computer Interaction, Optimization and Robotic Applications*, Ankara, Turkey, 1-6.

Chalaki, B., Beaver, L.E., Mahbub, A.M.I, Bang, H., Malikopoulos A.A. (2022). A research and educational robotic testbed for real-time control of emerging mobility systems: from theory to scaled experiments. *IEEE Control Systems Magazine*, 42(6), 20-34.

Chen, G., Yang, X., Zhang, X., Hu, H. (2021). Water hydraulic soft actuators for underwater autonomous robotic systems. *Applied Ocean Research*, 109(102551), 1-12.

Ding, L., Huang, L., Li, S., Gao, H., Deng, H., Li, Y., Liu, G. (2020). Definition and application of variable resistance coefficient for wheeled mobile robots on deformable terrain. *IEEE Transactions on Robotics*, 36(3), 894-909.

Feng, S., Feng, Y., Yu, C., Zhang, Y., Liu, H. (2020). Testing scenario library generation for connected and automated vehicles, part I: methodology. *IEEE Transactions on Intelligent Transportation Systems*, 22(3), 1573-1582.

Gao, X., Li, J., Fan, L., Zhou, Q., Yin, K., Wang, J., Song, C., Huang, L., Wang, Z. (2018). Review of wheeled mobile robots' navigation problems and application prospects in agriculture. *IEEE Access*, 6, 49248-49268.

Gnambs, T., Appel, M. (2019). Are robots becoming unpopular? Changes in attitudes towards autonomous robotic systems in Europe. *Computers in Human Behavior*, 93, 53-61.

Iqbal, J., Xu, R., Halloran, H., Li, C. (2020). Development of a multi-purpose autonomous differential drive mobile robot for plant phenotyping and soil sensing. *Electronics*, 9(9), 1-19.

Ke, F., Li, Z., Yang, C. (2017). Robutst tube-based predictive control for visual servoing of constrained differential-drive mobile robots. *IEEE Transactions on Inductrial Electronics*, 65(4), 3437-3446.

Khan, M.N., Jamil, M., Gilani, S.O., Ahmad, I., Uzair, M., Omer, H. (2020). Photo detector-based indoor positioning systems variants: a new look. Computers & Electrical Engineering, 83, 1-13.

Lai, Y., Yang, F., Meng, G., Lu, W. (2022). Data-driven flexible scheduling and route optimization. *IEEE Access*, 8, 1-15.

Leena, N., Saju, K.K. (2016). Modelling and trajectory tracking of wheeled mobile robots. *Procedia Technology*, 24, 538-545.

Luckcuck, M., Farrell, M., Dennis, L.A., Dixon, C., Fisher, M. (2020). Formal specification and verification autonomous robotic systems: a survey. *ACM Computing Surveys*, 52(5), 1-41.

Marais, J., Ambellouis, S., Flancquart, A., Lefebvre, S., Meurie, C, Ruichek, Y. (2012). *Social and Behavioral Sciences*, 48, 796-805.

Martins, F.N., Sarcinelli-Filho, M., Carelli, R. (2017). A velocity-based dynamic model and its properties for differential drive mobile robots. *Journal of Intelligent & Robotic Systems*, 85, 277-292.

Min, H., Wu, X., Cheng, C., Zhao, X. (2019). Kinematic and dynamic vehicle model-assisted global positioning method for autonomous vehicles with low-cost GPS/camera/ın-vehicle sensors. *Sensors*, 19(24), 1-24.

Ovodenko, V.B., Trekin, V.V., Korenkoa, N.A., Klimenko M.V. (2015). Investigating range error compensation in UHF radar through IRI-2007 real-time updating: preliminary results. *Advances in Space Research*, 56(5), 900-906.

Paijitprapaporn, C., Thongtan, T., Satirapod, C. (2021). Accuracy assessment of integrated GNSS measurments with LIDAR mobile mapping data in urban environments. *Measurement: Sensors*, 18(100078), 1-5.

Tsitsimpelis, I., Taylor, C.J., Lannox, B., Joyce, M.J. (2019). A review of ground-based robotic systems for the characterization of nuclear environments. *Progress in Nuclear Energy*, 111, 109-124.

Veres, M., Moussa, M. (2019). Deep learning for intelligent transportation systems: a survey of emerging trends. *IEEE Transactions on Intelligent Transportation Systems*, 21(8), 3152-3168.

Wang, J., Liu, J., Kato, N. (2019). Networking and communications in autonomous driving: a survey. *IEEE Communications Surceys & Tutorials*, 21(2), 1243-1274.

Weiskircher T., Wang, Q., Ayalew B. (2017). Predictive guidance and control framework for (semi-)autonomous vehicles in public traffic. *IEEE Transactions on Control Systems Technology*, 25(6), 2034-2046.

Yang, Y., Xu, Y., Li, J., Yang, C. (2018). Progress and performance evaluation of BeiDou global navigation satellite system: data analysis based on BDS-3 demonstration system. *Science China Earth Science*, 61, 614-624.

Yurtsever, E., Lambert, J., Caarballo, A., Takeda, K. (2020). A survey of autonomous driving: common practises and emerging technologies. *IEEE Access*, 8, 58443-58469.

Zhai, J., Song, Z. (2019). Adaptive sliding mode trajectory tracking control for wheeled mobile robots. *International Journal of Control*, 92(10), 2255-2262.

Zhang, Y., Shen, C., Tang, J., Liu, J. (2018). Hybrid algorithm based on MDF-CKF and RF for GPS/INS system during GPS outages. *IEEE Access*, 6, 35343-35354.