İSTANBUL UNIVERSITY CERRAHPAŞA

ORIGINAL ARTICLE
Electrica 2023; 23(3): 619-633

Data-Driven System Identification of a Modified Differential Drive Mobile Robot Through On-Plane Motion Tests

Murat Bakirci

Unmanned/Intelligent Systems Lab, Tarsus University, Faculty of Aeronautics and Astronautics, Mersin, Turkey

Cite this article as: M. Bakirci, "Data-driven system identification of a modified differential drive mobile robot through on-plane motion tests," *Electrica*, 23(3), 619-633, 2023.

ABSTRACT

A set of system identification experiments are conducted for a modified differential drive robot. A linear model was developed by performing identification experiments to verify the model and determine unknown parameters by utilizing the motion profiles of the robot. A discrete model based on travelled distance increment was used. Nonlinear model estimates have also been generated using automated identification functions from the MATLAB's system identification toolbox. The linear model was tested through obtained data and the results were compared with the nonlinear model. It was observed that the assumption of the linearly time-invariant model allows for the state-space formulas to be implemented so that these reproductions are developed in a relatively simple structure that does not unnecessarily complicate the control system.

Index Terms—Autonomous system, data-driven control, differential drive, mobile robot, system identification

I. INTRODUCTION

Due to extensive mobility capabilities within their environments, autonomous mobile robots have attracted a wide range of research interests in many engineering fields [1-3]. Diverse applications of these robots such as industry [4], research [5], military [6], education [7], and surveillance [8] along with many other potential applications [9], make them a popular choice. In particular, the mobilization of wheeled mobile robots (WMR) has exclusively escalated with the technological advancements in hardware manufacturing. This has motivated the robotics and control community to seek novel applications of WMRs through research fields such as modeling [10], control [11], identification [2], and simulation [12]. Taking into consideration all types of WMRs, differential drive mobile robots (DDMR) are prominent platforms in a wide variety of practical applications because of their straightforward architecture and dynamics. A typical DDMR has main wheels driven by DC motors and an unactuated rear wheel that stabilizes the 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 [13]. On the other hand, the differential drive characteristic of DDMRs makes them a common example of nonholonomic by reason of rolling constraints [1, 5]. There are a number of types of modeling that exist for these robotic platforms such as linear time-invariant models which can be attained by Taylor series linearization [14].

Indeed, mathematical models of most physical systems are composed of differential equations. Due to this fact, sometimes mathematical modeling has become very rigorous and even impossible to generate a model. Nonetheless, a model can be developed through an alternative approach which is the system identification method that utilizes the system measurement data [15, 16]. In particular, data-driven system identification provides system description by performing input and output data utilization [17]. There exist numerous publications in the literature on the application of system identification to robotics such as robot simulators [18], three-dimensional parallel robots [19], and robot training [20]. For DDMRs, physics-based and data-driven models are quite popular choices [17]. The behavior of a physical system can be predicted through physics-based models before its construction. Data-driven models, on the other hand, evaluate

Corresponding author:
Murat Bakırcı

E-mail:
muratbakirci@tarsus.edu.tr

Received: September 21, 2022

Accepted: May 30, 2023

Publication Date: August 1, 2023

DOI: 10.5152/electrica.2023.22164

a system based on various identification tests. Therefore, this model approach is able to outline the behavior of a system satisfactorily by handling an entire set of observations which makes the data-driven model approach widely applicable. Through system identification of a nonlinear model, behavioral aspects of a DDMR can be obtained. However, nonlinear model identification can be rigorous because complex sets of nonlinear equations of the systems are among the most difficult equations to solve. Since every nonlinear system can be represented by a linear model, they are preferably used in many system identification problems [14, 15]. However, it should be noted that this is only possible through obtaining a good linear representation of the system. In other words, this is only valid where the linear model adequately represents the system. Identification of linear models has many advantages over nonlinear models such as simplicity of model structure, identification through linear regression, and generally no convergence issue. In any case, even when such problems are encountered, they are relatively simple to handle with linear models. This work focuses on the data-driven system identification of a modified two-wheeled differential drive robot. The developed mobile robot has a high power-to-weight ratio, which makes it quite fast compared to its counterparts. A linear model has been developed through performing system identification experiments for verification of the model as well as identification of the unknown parameters by utilization of actual motion profiles of the robot. The identification procedure is carried out in the discrete-time domain by relating known input vectors and output of the unknown system. Butyl rubber-based wheels with high grip features have been developed to ensure the 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, it has been shown that a consistent model can be obtained even without taking such effects into account. A nonlinear model estimate has also been generated using the automated identification functions from the MATLAB toolbox.

II. LITERATURE REVIEW

Efforts to provide more accurate and consistent autonomy to mobile robots are being carried forward with many identification, control, and other related studies. In [21], the identification method of a DDMR was developed by including actuator dynamics. The authors developed a discrete model based on the increase in traveled distance, and the identification procedure was performed with the resulting linear and time-invariant model. The system identification and control methodology developed for a DDMR produced for the line following is presented in [22]. In this way, the performance of the robot has been tried to be improved by applying a controller based on system identification. Another study describes the odometric parameters of DDMRs with a calibration technique based on continuous-time kinematic equations [14]. The applied method tries to avoid the time discretization error by applying least squares estimation. This is accomplished by utilizing the measurements obtained by the camera integrated into the robot, as well as the position information obtained from the wheel encoders.

In [23], the control of a line-following DDMR is studied using a real, that is, non-ideal, actuator approach. Taking into account the velocity saturation of the actuator, how this affects the performance of the robot is discussed experimentally. The solution scheme for controlling a locomotion task of a DDMR using sliding mode is discussed in [9]. For this, the stability of the proposed dynamic model has been examined and the limits of the sliding mode have been investigated. Another study performs the identification procedure by developing a linear dynamic model based on the distance traveled by a DDMR [24]. Thus, the time-varying parameters of the robotic platform are estimated by simple identification methods. Motion control of a four-wheel drive mobile robot based on the vector field orientation technique is presented in [25]. The authors develop a realistic simulation environment, combining the traditional field orientation method with a multivariate orientation control in this environment. As a result, it is shown that the proposed method has a more accurate route-tracking capability. In [26], motion tests and navigation of a four-wheeled mobile robot are handled using a fuzzy logic controller. In this context, the dynamic model of the robot was developed and the ground plane identification was carried out. A chaotic path-planning strategy for a mobile robot used for field scanning is expressed in [27]. In another study, some innovative modeling and control applications for DDMRs are discussed in detail [12]. In this context, solutions are offered in order to overcome some of the main difficulties applicable to mobile robots. In [28], localization and identification for the multi-mobile robot system are discussed. By modifying the Probability Hypothesis Density (PHD) filter, which is a multi-target tracking technique, identification is performed by means of robot odometric data.

A unified dynamic modeling approach for DDMRs based on Lagrangian and Newton–Euler mechanics is presented in [29]. Along with the proposed method, the main difficulties encountered in modeling mobile robotic systems are described. Another study discusses the development of robust kinematics and closed-loop motion control of a DDMR capable of reaching specific target locations in indoor environments [30]. The authors try to verify the consistency of the designed kinematics and control method with the Lyapunoc criterion. In [31], an inverse kinematics model has been experimentally developed through direct control of a DDMR. The angular velocities of the robot wheels were adjusted to provide linear movement of the robot, taking into account all the mechanical and systematic errors of the robot, and the system performance was investigated. The method by which a mobile robot effectively uses environmental resources and makes them useful so that the robot can guide itself is presented in [32]. Another work experimentally develops a kinematic model framework based on external sensors for a skid-steering DDMR [10]. This is accomplished based on the limitation of the robot's circular centers of rotation. In [33], a specific controller adaptive to discrete-time form has been developed to control the trajectory of a DDMR. In the proposed method, which does not require a comprehensive model of the robot, identification of the dynamic model is performed using a neural network. A different modeling and trajectory control approach for an effective design of a mobile robot is discussed in [34]. Another study has examined the issue of wheel slippage in detail in DDMRs, and an accurate dynamic model is developed based on the resulting traction forces between the wheel and the ground [35]. With this developed model and control strategy, it is tried to prevent these sliding effects that cause various problems.

In [36], system identification of a DDMR is performed through an experimental environment created using a camera and specific ground markers. With the developed model, parameters such as system delay and friction are tried to be determined experimentally. Additionally, the linear model of the robot is obtained by the regression method and utilized for parameter estimation. In [37], a

dynamic model for DDMRs is developed based on the speed commands, thus taking into account the actuator effects. In particular, a solution is proposed for commercial mobile robots that only accept speed command as input. It is shown that the obtained dynamic model has many computational advantages. An effort to improve the motor torques of remote-controlled two-wheeled mobile robots and thus the consistency of the trajectory they are expected to follow is presented in [38]. The proposed method uses p-norm and can provide full power to the robot actuators if systematic errors are eliminated. Another study proposes a method that applies linear parameterization to the robot model, taking into account the actuator-induced effects of a DDMR [39]. In this way, the robotic system parameters become identifiable through the recursive least squares method. An overarching model for proper navigation of mobile robots used for industrial purposes is provided in [40]. The presented navigation method consists of two layers, making it easy to adapt this architecture to various conditions and environments. In [41], the performance of the kinematics controller is tried to be increased by considering the dynamic model of a mobile robot for situations that require fast movement or load carrying. Thus, the robot parameters can be identified properly and the distance error is greatly reduced.

The effort to enable linear control by applying partial linearization to a DDMR dynamic model that includes actuator effects is discussed in [42]. An optimization-based method is used to modify the robot control parameters. Another work presents an actuator control solution that largely eliminates error using the discretized kinematic model of DDMRs [43]. Moreover, with the proposed model, systematic faults are identified through the extended Kalman filter. In [44], the model predictive control strategy of a DDMR is discussed by developing a linearized dynamic model. The system gains obtained through the controller are adjusted by utilization of the measured moments. Considering a four-wheel DDMR as a multi-input single-output system, the stages of system identification are discussed in [45]. The proposed identification method is investigated through the autoregressive exogenous (ARX) and autoregressive moving average exogenous parametric models and it is observed that the ARX model is more suitable. Detailed trajectory tracking of an omnidirectional robotic platform is presented in [46]. Unlike traditional methods, the data obtained through the production phase is utilized, and thus proper modeling of the robotic system is carried out through system identification. In [47], a kinematic controller that generates the reference values of the commanded velocity inputs is adapted to a newly designed DDMR. Based on the nonlinear kinematic model, the proposed kinematic controller is effectively implemented. An analytical model that estimates the motion of a DDMR by taking into account torque and friction effects is discussed in [48]. An overview of the estimation of system parameters of mobile robots is given in [49]. In this context, system modeling, trajectory control and parameter identification are performed and their validation is tested.

III. TEST PROCEDURE

A. Test Robotic Platform

The test robot platform used in this study is based on the Stingray robot kit which has several properties that make it a highly useful indoor robotic system. As shown in Fig. 1, it has two main wheels that can rotate independently through the differential drive feature. The system consists of two large plates connected with supporting panels which provide extra volume for mounting external hardware. This has facilitated adding a variety of additional electronics in the Stingray used for the study to allow experimentation. The resultant robot is referred to as Modified Stingray Robot Platform (MSRP) as shown on the far right of Fig. 1. The MSRP motors, embedded sensors, and other electronic equipment are controlled by a multi-processor microcontroller. Motor encoders are used to measure the rotation of the motor shafts connected to the wheels, that is, to measure the wheel displacement in a certain time interval and thus to determine the speed of the robot. Motor encoders are used to measure the rotation of the motor shafts connected to the wheels and thus to determine the speed of the robot. Therefore, closed-loop control of the robot can be achieved, that is, parameters such as the speed of the robot and the distance traveled per unit time are controlled. The robot also has a Wi-Fi module that provides telemetry and remote communication capabilities. Through program control, data can be transmitted and received to any program that reads from and writes to the appropriate IP address and ports provided the computer is in the research lab local area network.

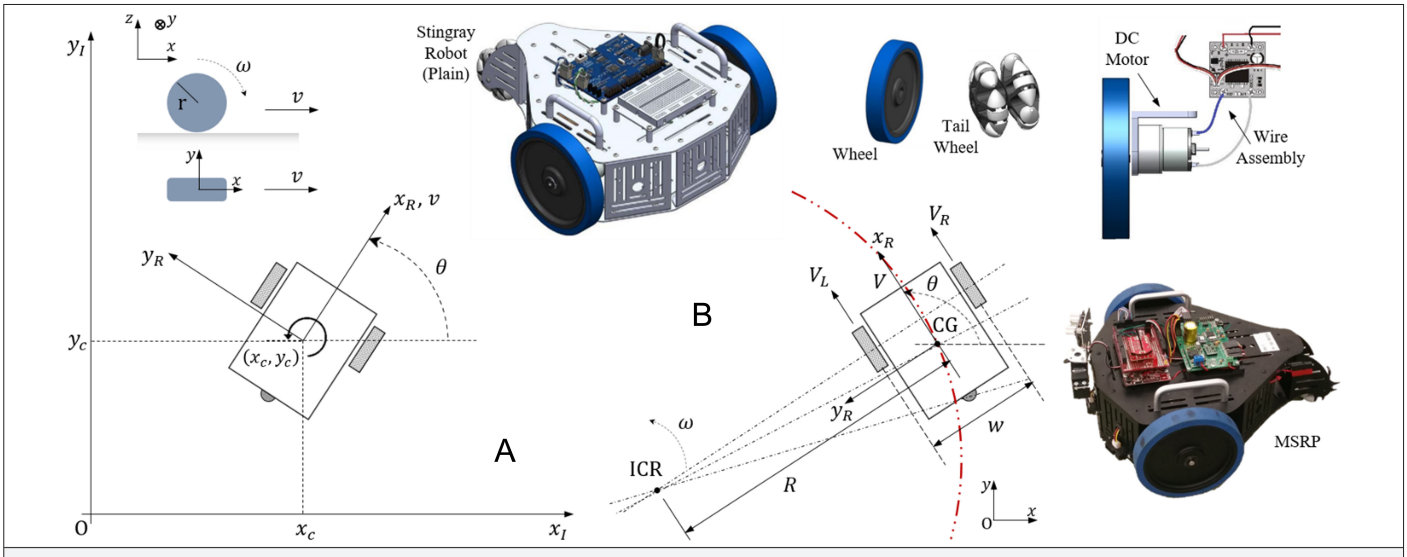

Fig. 1. Illustration of the robotic platform and coordinate systems.

Through the motor control board, the MSRP motors can be controlled using two different approaches. In "effort mode," a fixed percentage of total battery power is exposed to each motor. The effort is translated into a Pulse Width Modulation (PWM) signal that drives the motors directly. In "velocity mode," a closed-loop feedback controller is used to achieve the desired rotational speed for each motor. System feedback is obtained by rotary encoders. A Proportional Integral (PI) controller with feed-forward is used to generate PWM commands to track the desired speed. The reason for having a velocity control mode is that due to motor differences, applying the same voltage to both motors will not necessarily cause both motors to rotate at precisely the same speed. By using the encoders, the duty cycle can be adjusted through the motor control board to achieve the desired velocity.

B. Kinematic Model of the Robot

Although sensor readings provide some clues, generally there is no direct way to measure a robot's pose. Sensor data can properly be interpreted through kinematics to generate an accurate estimation. Commanding an equal "effort" or "velocity" input to both wheels will move MSRP forward. Moreover, setting different inputs to each wheel will pivot MSRP about the slower wheel. It is important to note that the amount of traveled distance of each wheel does not necessarily give the correct position of the robot. For instance, the distance traveled by each wheel can be the same for two different trajectories, yet, the final pose can be completely different. Unlike joint rotations of robot manipulators, the traveled distance is not enough to derive the pose of the robot. It is clear that kinematics depends on the specific configuration of the robot and each configuration has its own kinematic properties. In the case of MSRP, the locomotion actions are represented by wheel speeds and wheel steering values.

MSRP can be modeled as a flat rigid chassis mounted on wheels. Using the flat-earth assumption, the chassis is considered to be perfectly parallel to the ground; thus, it locomotes on a horizontal plane. Defining the inertial coordinate system by $\{x_I, y_I\}$ and the robot coordinate system by $\{x_R, y_R\}$ as shown in Fig. 1(a), full pose of the MSRP in the inertial frame can be denoted by

$$\xi_I = \begin{bmatrix} x & y & \theta \end{bmatrix}^T \tag{1}$$

To map the motion of MSRP from one coordinate system to another, a relationship between the inertial coordinate system and robot coordinate system is given by

$$\dot{\xi}_R = R(\theta)\dot{\xi}_I \tag{2}$$

where R is the orthogonal rotation matrix denoted by

$$R(\theta) = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3}$$

Linear velocity, v, is directly proportional to the angular velocity, ω, of the wheels as

$$\omega \cdot r = V \tag{4}$$

where r is the radius of the wheels. Considering the robot moving along a circle of radius R as shown in Fig. 1(b), the following equations hold based on (4)

$$\omega(R + w/2) = V_R \tag{5a}$$

$$\omega(R - w/2) = V_L \tag{5b}$$

where w is the distance between the wheels, V_L and V_R are the velocities of the left and right wheels, respectively. Solving (5a) and (5b) for R and ω

$$R = \frac{w}{2}\left(\frac{V_R + V_L}{V_R - V_L}\right) \tag{6}$$

$$\omega = \frac{V_R - V_L}{w} \tag{7}$$

According to (6) and (7), the robot moves through a straight line when $V_L = V_R$; rotates about its midpoint when $V_L = -V_R$; rotates about the left wheel when $V_L = 0$; and rotates about the right wheel when $V_R = 0$. To find the velocity of the geometric center of the chassis, (6) and (7) are plugged into $\omega = V/R$ and get

$$V = \frac{V_R + V_L}{2} \tag{8}$$

By defining the differential velocity V_D, another representation of V_L and V_R can be made as follows:

$$V_R = V + V_D \tag{9a}$$

$$V_L = V - V_D \tag{9b}$$

Note that the robot moves through a straight line when $V_D = 0$. Substituting (9a) and (9b) into (6) yields an expression which is a more convenient selection of speeds in order to trace desired circular paths

$$R = -\frac{w}{2}\frac{V}{V_D} \tag{10}$$

Through basic trigonometry, the motion equations can also be expressed as

$$\dot{x} = V\cos\theta \tag{11a}$$

$$\dot{y} = V\sin\theta \tag{11b}$$

$$\dot{\theta} = \omega \tag{11c}$$

Integration of both sides yields

$$x(t) = \int_{t_0}^{t} V(t)\cos\left[\theta(t)\right]dt \tag{12a}$$

$$y(t) = \int_{t_0}^{t} V(t)\sin\left[\theta(t)\right]dt \tag{12b}$$

$$\theta(t) = \int_{t_0}^{t} \omega(t)dt \tag{12c}$$

Substituting (7) and (8) into (12) yields the ultimate motion equations as follows

$$x(t)=\int_{t_0}^{t}\frac{V_R+V_L}{2}\cos\left[\theta(t)\right]dt \tag{13a}$$

$$y(t)=\int_{t_0}^{t}\frac{V_R+V_L}{2}\sin\left[\theta(t)\right]dt \tag{13b}$$

$$\theta(t)=\int_{t_0}^{t}\frac{V_R-V_L}{w}dt \tag{13c}$$

The structural design of MSRP allows it to perform four basic movements on a two-dimensional plane. These are forward motion, right turn, left turn, and rotation around its own axis. Since right and left turns are the symmetry of each other, it will be sufficient to perform tests with one of them. Due to the high grip feature of the wheels, the robot's rotation around its own axis was not included in the tests. Since it requires both wheels to move at the same speed but in opposite directions, it prevents the robot from performing this motion appropriately. Laboratory tests consist of two different locomotion cases which are a full left turn, $(V_L>0, V_R=0)$, and a forward motion, $(V_L=V_R>0)$. Considering the first case by assuming the velocity of the left wheel V_L is constant, then (13c) can be modified as follows:

$$\theta(t)=\int_{0}^{t}\frac{-V_L}{w}dt \tag{14}$$

$$\theta(t)=\frac{-V_L}{w}t \tag{15}$$

Substituting (15) into (13a) and (13b)

$$x(t)=\int_{t_0}^{t}\frac{V_R+V_L}{2}\cos\left[\frac{-V_L}{w}t\right]dt \tag{16}$$

$$y(t)=\int_{t_0}^{t}\frac{V_R+V_L}{2}\sin\left[\frac{-V_L}{w}t\right]dt \tag{17}$$

Calculation of these integrals yields

$$x(t)=\left(\frac{w}{V_L}\right)\sin\left(\frac{V_L}{w}t\right) \tag{18}$$

$$y(t)=\left(-\frac{w}{V_L}\right)\cos\left(\frac{V_L}{w}t\right) \tag{19}$$

A similar procedure can also be applied for forward motion case as follows:

$$\theta(t)=\int_{0}^{t}\left(\frac{V_R-V_L}{w}\right)dt \tag{20}$$

$$\theta(t)=\frac{V_R-V_L}{w}t \tag{21}$$

Substituting (21) into (13a) and (13b) yields

$$x(t)=\frac{V_R+V_L}{V_R-V_L}\frac{w}{2}\sin\left(\frac{V_R-V_L}{w}t\right) \tag{22}$$

$$y(t)=-\frac{V_R+V_L}{V_R-V_L}\frac{w}{2}\cos\left(\frac{V_R-V_L}{w}t\right)+\frac{V_R+V_L}{V_R-V_L}\frac{w}{2} \tag{23}$$

Rather than coming up with a continuous function, the microcontroller processes a series of samples at a specified sampling rate. Therefore, it would be more convenient to use the following discrete form, the full derivation of which is given in the Appendix.

$$\begin{bmatrix}x\\y\\\theta\end{bmatrix}_{t+1}=\begin{bmatrix}x\\y\\\theta\end{bmatrix}_{t}+\begin{bmatrix}dt\frac{V_L+V_R}{2}\cos\left(\theta_t+\frac{dt\omega_t}{2}\right)\\dt\frac{V_L+V_R}{2}\sin\left(\theta_t+\frac{dt\omega_t}{2}\right)\\dt\omega_t\end{bmatrix} \tag{24}$$

Thus, by sampling the wheel speeds (V_L, V_R) at a fixed time interval (dt), the inertial pose of the robot over time can be derived using the above equation.

C. Test Methodology

For most robotics applications, system identification, which is the identification of the theoretical input/output relation x generated from real input/output data, is complicated because of nonlinearity, which requires a rigorous mathematical procedure. However, system identification problems become significantly tractable if the system is absolutely linear. A linear system assumption generally, but not always, simplifies the problem in terms of model development. Thus, modeling can be done straightforwardly through the identification of the system parameters.

The system input of the MSRP is the commands given to the motors that drive the wheels, and the corresponding system output is the measured speeds of the wheels. In the absence of any initial input, u, the value of a linearized single-input single-output system is given as

$$x(t)=x(0)\exp(-t/\tau) \tag{25}$$

where the initial value is indicated by $x(0)$, and τ is the time constant of the system. The value of a system with an initial value of 0, a gain of G, and an input of u is as follows:

$$x(t)=Gu\left[1-\exp(-t/\tau)\right] \tag{26}$$

Thus, by measuring the system response and gain corresponding to the input value, the whole system can be modeled assuming that the linearized form of the system given by (24) is (25) and (26). Meanwhile, a similar approach can also be taken through the state-space (SS) method. The model generation procedure is summarized in Table I.

TABLE I. STATE-SPACE (SS) MODEL GENERATION ALGORITHM

Input	**{Obtained data file (.txt)}**
Step 1	Assess each column to a variable
Step 2	Package specific data set, $[I,O,f]$
Step 3	Create s-s model, m_{ss}
Step 4	Create model output, y_{ss}
Output	Model prediction

First, a MATLAB program infers the response of the motors to the system inputs, from which the system parameters are computed for each state. Meanwhile, the model output is analyzed through the code and compared with the experimental data. On the other hand, the SS model of the system is created through another program by using the input/output relation of the system, and thus, the model output is compared with the experimental (actual) values. For each iteration of the program's loop, an array is declared into which the data is stored. To save this on a file, a Wi-Fi module is used to upload the data to a computer through File Transfer Protocol (FTP) which is summarized in Table II.

D. Identification Procedure

For the DDMR platform under consideration, the system input is the commands sent to the wheels, and the corresponding system output is the wheel speeds. This system can be represented by the ARX model as follows:

$$v(t+1)=\sum_{\Gamma=1}^{-} m_k v(t-\Gamma+1)+\sum_{\Phi=1}^{\dagger} n_\Phi u(t-\Phi+1)+\Lambda(t+1) \tag{27}$$

where u and v represent model input and output respectively, m and n are model parameters, t stands for sampling rate, $\Lambda(t+1)$ is the system noise. In its implicit form, (27) can also be written as

$$v(t+1)=\zeta^T(t)\eta+\Lambda(t+1) \tag{28}$$

where ζ^T and η are defined as regression and parameter vector, respectively, and can be expressed as follows:

$$\zeta=\left[v(t) \quad \cdots \quad v(t-\alpha+1) \quad u(t) \quad \cdots \quad u(t-\beta+1)\right]^T \tag{29}$$

$$\eta=\left[m_1 \quad \cdots \quad m_\alpha \quad n_1 \quad \cdots \quad n_\beta\right]^T \tag{30}$$

TABLE II. MSRP MOVEMENT AND FILE TRANSFER ALGORITHM

Input	$\{I_L^{e,v}, I_R^{e,v}\}$
Step 1	Execute required movement
Step 2	Write necessary data to matrix
Step 3	Send data file to computer
Output	V_L, V_R

MSRP, Modified Stingray Robot Platform.

The identification procedure emerges as the determination of η based on measurements to be made on the system. Therefore, it is necessary to determine η using $v(t+1)$ and $\zeta(t)$ measurements obtained for all t values. For this, the above expression is evaluated as an optimization problem and its value is tried to be minimized as follows.

$$J(k,\bullet)=k^{-1}\sum_{t=0}^{k-1}\left[v(t+1)-\zeta^T(t)\bullet\right]^2 \tag{31}$$

Here, $v(t+1)-\zeta^T(t)\bullet$ denotes the prediction error of the system. Depending on the minimization of the value of this expression, the estimation error of the system will be reduced to the same extent. Equation (31) can also be written more formally as:

$$\check{\eta}(k)=\text{argmin}_\Theta J(k,\eta) \tag{32}$$

The block diagram showing the system identification procedure is shown in Fig. 2. The following (more appropriate) form can be used to make the solution of (31) simpler.

$$\check{\bullet}(k)=\underset{\bullet}{\text{argmin}}\left\{\left[V(k)-\Omega(k)\bullet\right]^T\left[V(k)-\Omega(k)\bullet\right]\right\} \tag{33}$$

where

$$V(k)=\left[v(1) \quad \cdots \quad v(k)\right]^T \tag{34}$$

$$\Omega(k)=\left[\zeta^T(0) \quad \cdots \quad \zeta^T(k-1)\right]^T \tag{35}$$

Then, (31) is differentiated with respect to η and is set equal to zero as it is desired to be minimized as follows:

$$\left[\frac{\partial J(k,\bullet)}{\partial\bullet}\right]_{\bullet=\check{\bullet}(k)}=-2\Omega^T(k)V(k)+2\Omega^T(k)\Omega(k)\check{\bullet}(k)=0 \tag{36}$$

Therefore,

$$\check{\bullet}(k)=\left[\Omega^T(k)V(k)\right]\left[\Omega^T(k)\Omega(k)\right]^{-1} \tag{37}$$

Equation (37) is the general solution to this optimization problem and can be expressed in its original form as follows:

$$\check{\bullet}(k)=\sum_{t=0}^{k-1}\zeta(t)v(t+1)\left[\sum_{t=0}^{k-1}\zeta(t)\zeta^T(t)\right]^{-1} \tag{38}$$

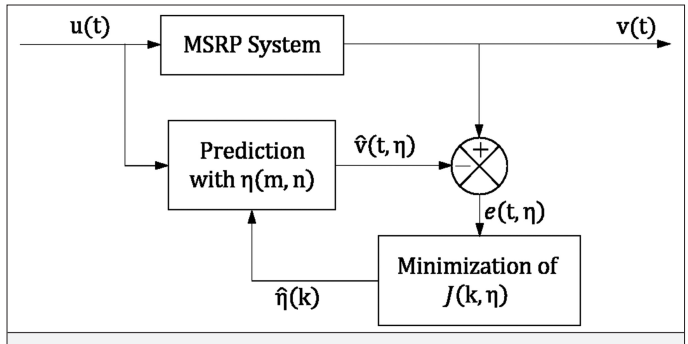

Fig. 2. Block diagram of the system identification procedure.

IV. RESULTS

Initially, a program is written to send step commands to the MSRP in order to move each wheel at a desired velocity and/or percentage effort. For each test, required data were collected at 20 Hz frequency. To implement movement on the robot and file transfer between the robot and the computer, a nested if-else loop was created to execute movements based on how much time has passed. For instance, for one test, the MSRP had to remain still for 2 seconds, move at 300 mm/s for 2 seconds, and stop again. A counter is initialized at zero and incremented by one, 20 times a unit interval when the code looped back through. Therefore, inputs to both wheels, wheel velocities, and odometer counts for each wheel had to be written to a matrix at 20 Hz. The counter is used to cycle through the array elements in the data file to obtain the required information. After all motion tests have been performed, the data file (.txt) is sent back to the computer through the FTP server client.

Figures 3, 4, and 5 show the results of the tests performed. The plots numbered from I to VIII indicate each individual test results. In order to compare the outputs in the same group, normalization (*) was performed and shown in Fig. 5. The horizontal dashed lines indicate normalized input levels. The plots reveal that the effort mode provides more accurate velocity outputs as the input increases. The velocity mode also yields accurate velocity outputs at high input values. When returning to 0 ($\tau \rightarrow 0$), both functions resulted in an undershoot, except when the effort input function received an input of 60% effort. For tests I–VIII, the system parameters need to be estimated.

The results of the tests performed are also compiled in Table III. (E-L) and (V-L) in the table indicate that the left wheel is moved in effort and speed modes, respectively. Since the right wheel is not commanded, the robot will make a rotation around this wheel. As shown in the table, the gain using velocity input was approximately 1 each time since the output velocity was about the same as the input function. The gain using effort input averaged to approximately 14. This gain will vary based on the battery voltage at the time of testing since the input function is the percentage of available battery voltage applied. The time constants from 0 to the step input were higher than from the step input to 0.

A total of nine more tests (IX to XVII) for forward locomotion have been conducted and the results are shown in Figs. 6, 7, 8, and 9. Similarly, the plots numbered from IX to XVI indicate individual test results. Figure 8 shows the normalized outputs for which V_L and V_R outputs of tests IX–XII, and V_L and V_R outputs of tests XIII–XVI,

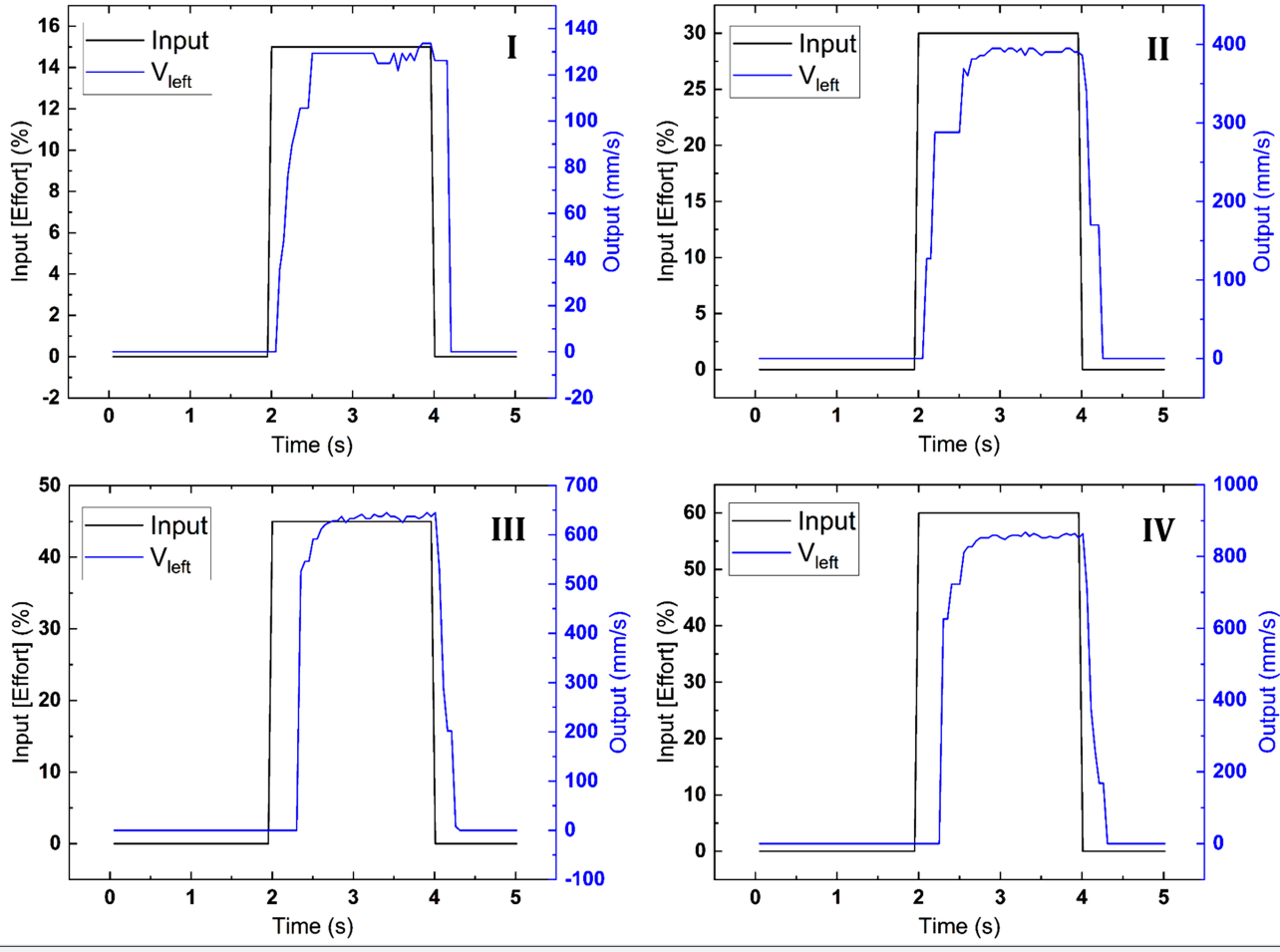

Fig. 3. MSRP responses to various effort inputs. MSRP, Modified Stingray Robot Platform.

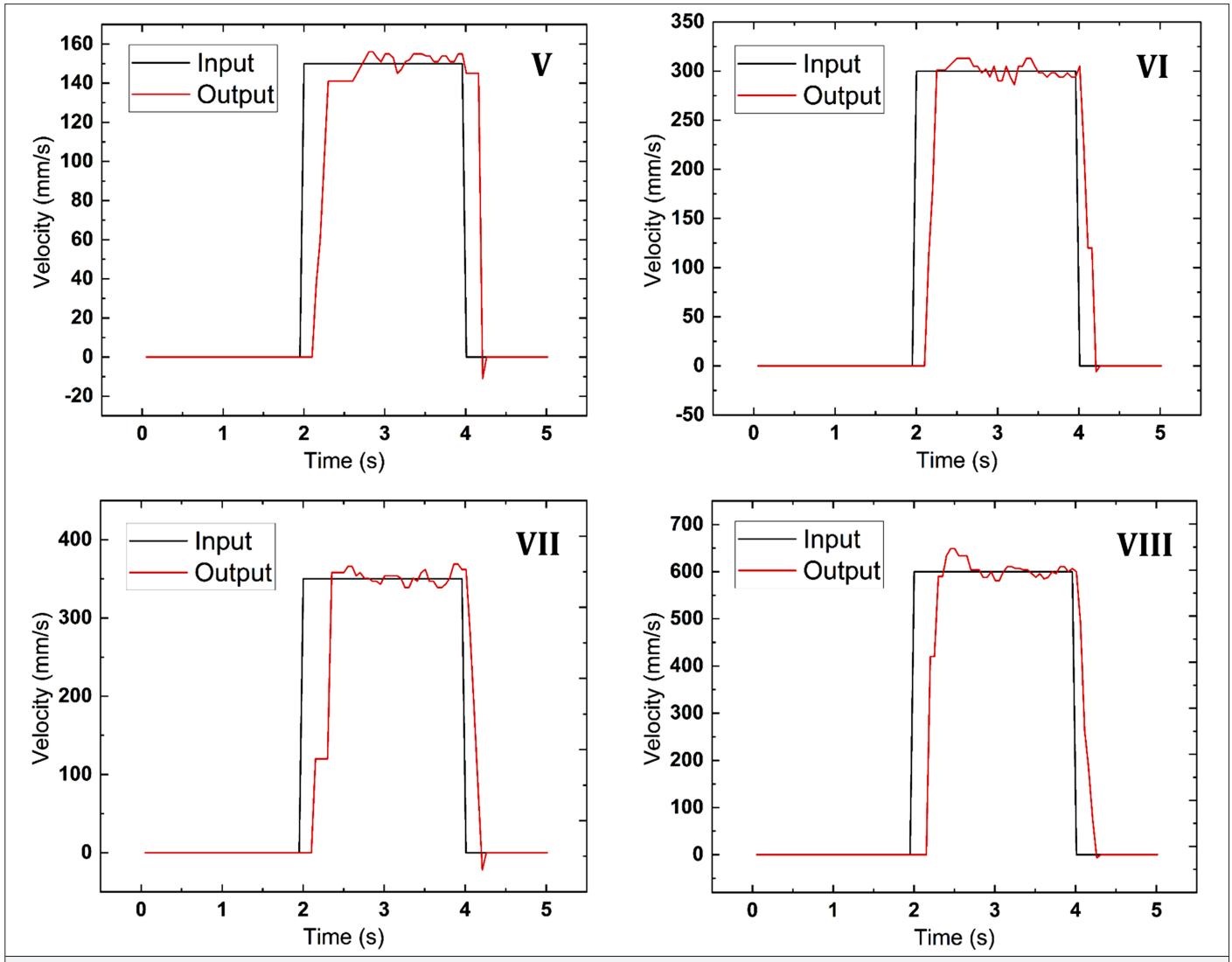

Fig. 4. MSRP responses to various velocity inputs. MSRP, Modified Stingray Robot Platform.

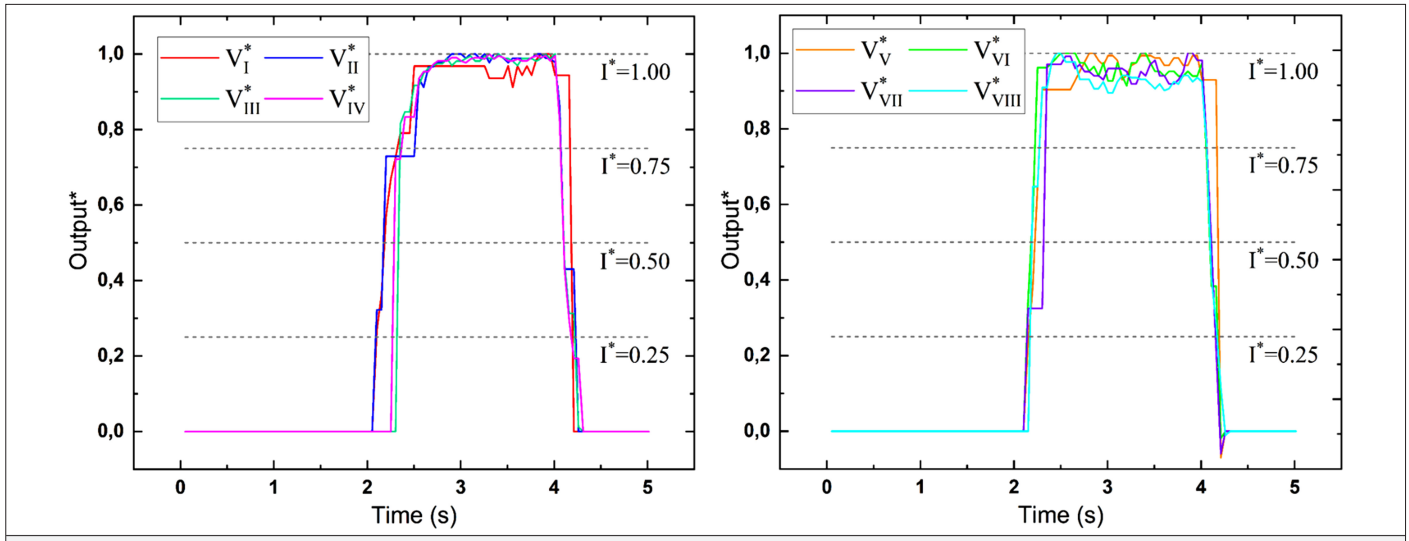

Fig. 5. Normalized MSRP responses to effort (left) and velocity (right) inputs. MSRP, Modified Stingray Robot Platform.

TABLE III. CIRCULAR LOCOMOTION TEST RESULTS

Test	Mode	Input	Gain	$\tau \leftarrow 0$	$\tau \rightarrow 0$
I	E-L	15%	9.132	0.603	0.247
III	E-L	30%	13.353	0.616	0.231
III	E-L	45%	14.544	0.415	0.452
IV	E-L	60%	14.177	0.305	0.339
V	V-L	0.15 m/s	1.054	0.528	0.228
VI	V-L	0.30 m/s	1.022	0.257	0.313
VII	V-L	0.35 m/s	1.009	0.524	0.455
VIII	V-L	0.60 m/s	1.017	0.443	0.222

$\tau \leftarrow 0$, step input from 0 to desired level; $\tau \rightarrow 0$, vice versa.

respectively. The horizontal dotted lines indicate the normalized input levels.

The main observation that can be made from these figures is that the effort input function resulted in less overshoot and less oscillations than the velocity input function. Both functions resulted in undershoot when going from the step input to 0. For the tests, the gain for each available steady-state level and the time constant for each step transition are compiled in Table IV. Note that, Test XVII is shown in two rows; thus, part 1 in the second row is actually part 4 of the same test. The result of the test XVII is shown separately since the input is completely different from other cases.

MATLAB's system identification functions were used to generate discrete, time-invariant, SS models using a data set that was obtained through both effort and velocity functions. Data set XI was chosen to generate the SS model for the effort mode function. Data set XV was chosen to generate the SS model for the velocity mode function. To generate each SS model, the data file was accessed using MATLAB and data columns were again assigned to variables which represent required data arrays. Data from which to create the SS model was compiled and then a SS model was created using appropriate functions. The SS model using the XI date set is denoted by (39). Similarly, the SS model using the XV data set is expressed with (40). The term denoted by e represents the total error.

$$x(t+1)=\begin{bmatrix}0.7592 & 0.1390\\ -0.4384 & -0.0198\end{bmatrix}x(t)+\begin{bmatrix}0.00121\\ 0.01456\end{bmatrix}u(t)+\begin{bmatrix}0.0002655\\ 0.0000078\end{bmatrix}e(t) \tag{39a}$$

$$y(t)=\begin{bmatrix}1457 & -120\end{bmatrix}x(t)+0u(t) \tag{39b}$$

$$x(t+1)=\begin{bmatrix}0.7592 & 0.1390\\ -0.4384 & -0.0198\end{bmatrix}x(t)+\begin{bmatrix}0.00121\\ 0.01456\end{bmatrix}u(t)+\begin{bmatrix}0.0002091\\ -0.0002883\end{bmatrix}e(t) \tag{40a}$$

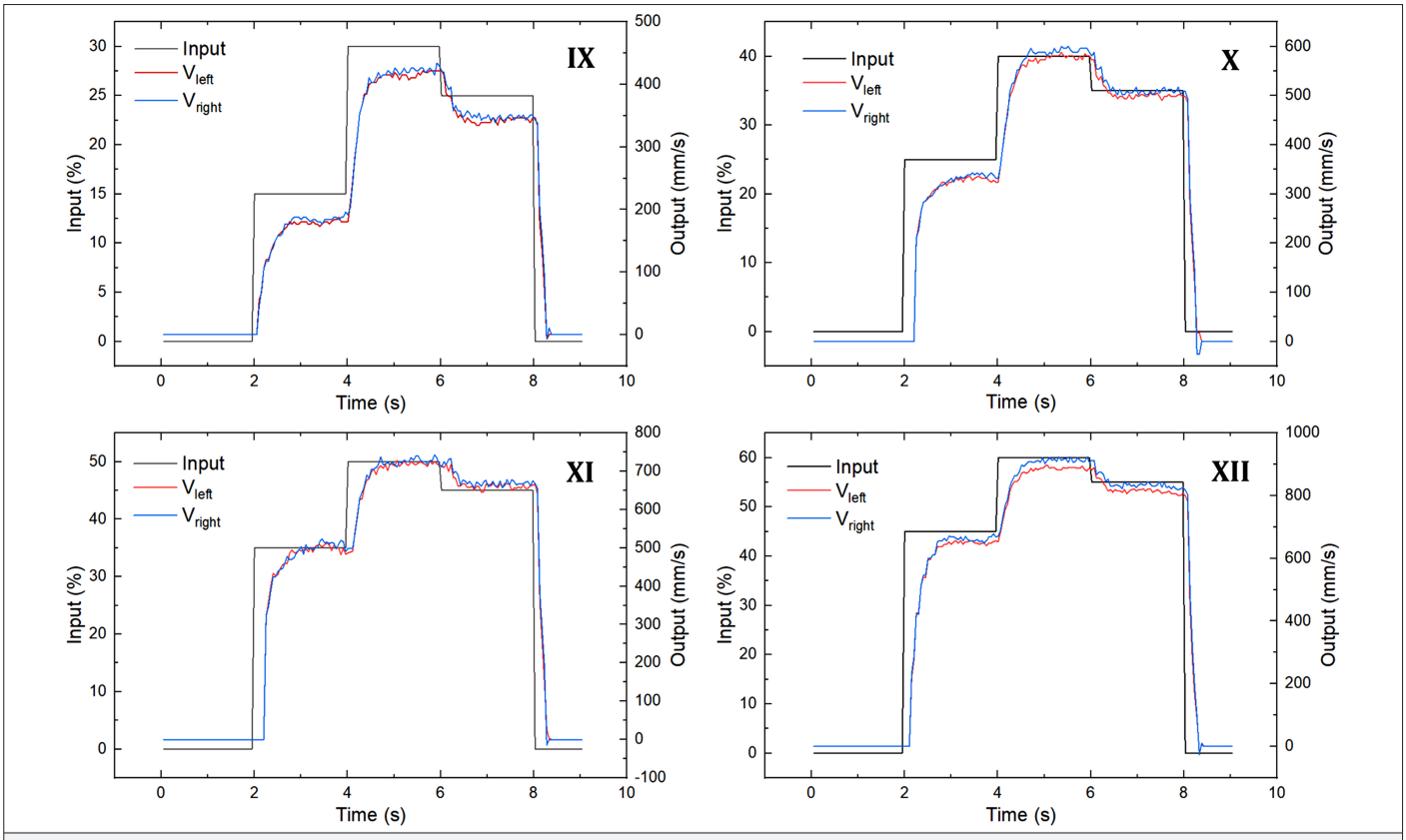

Fig. 6. MSRP responses to a variety of effort inputs. MSRP, Modified Stingray Robot Platform.

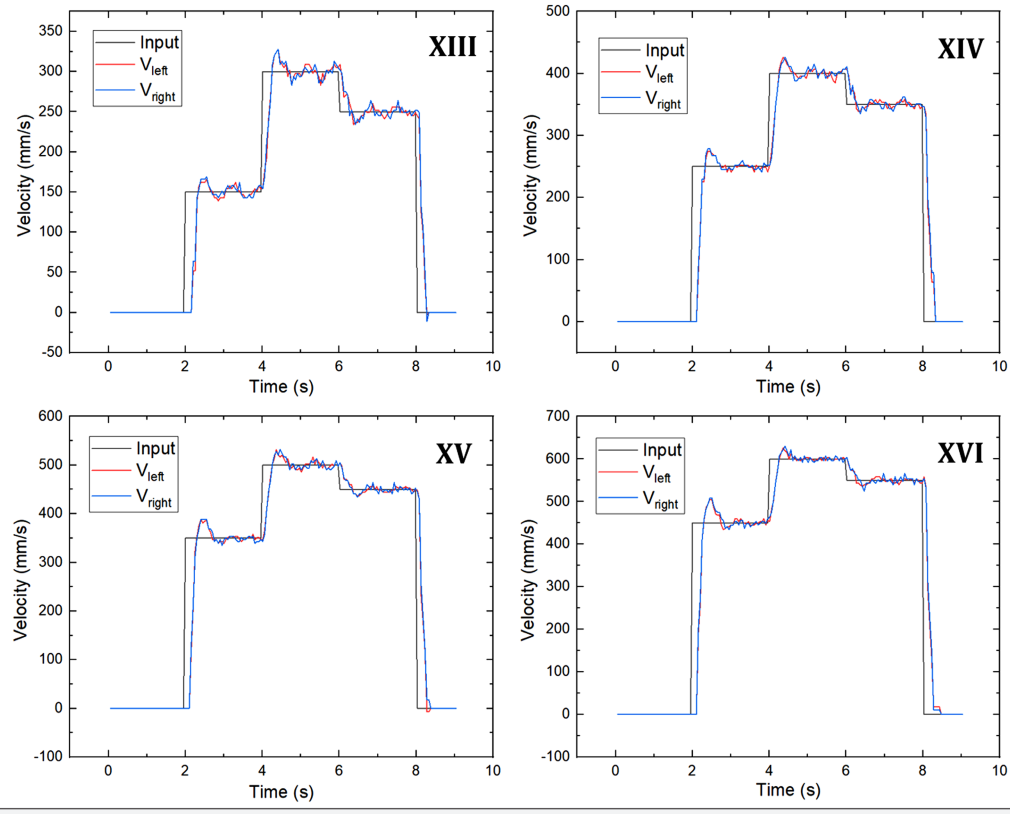

Fig. 7. MSRP responses to a variety of velocity inputs. MSRP, Modified Stingray Robot Platform.

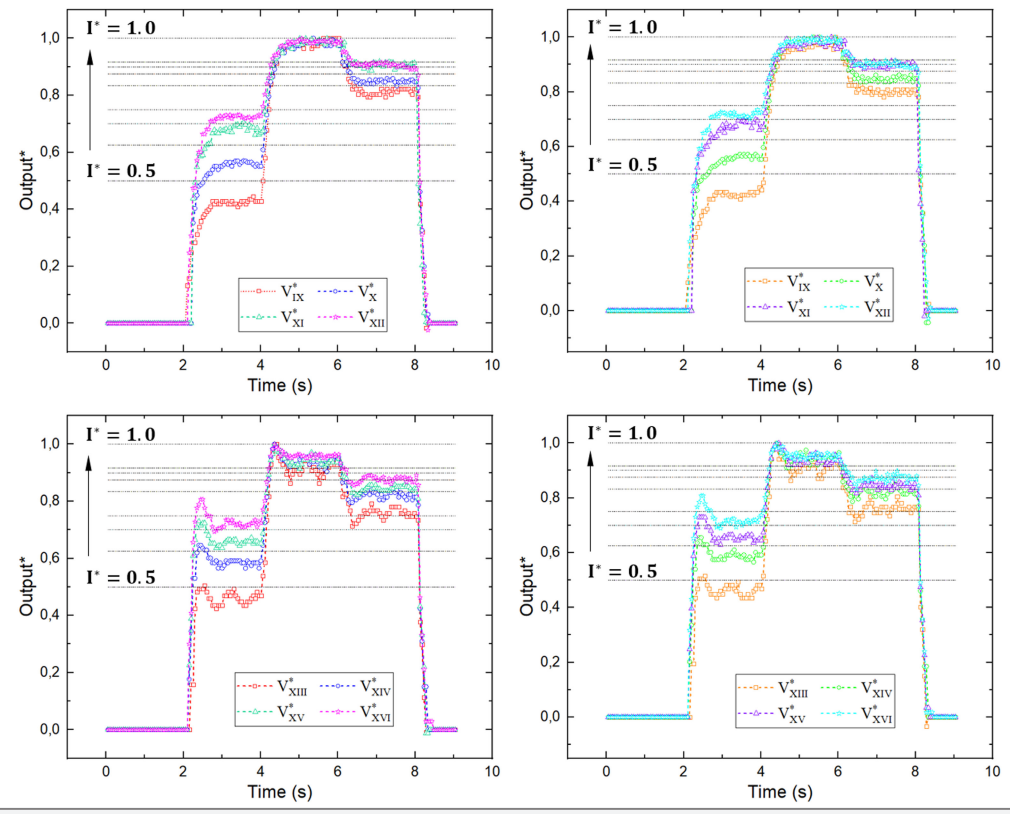

Fig. 8. Normalized MSRP responses to effort (top row) and velocity (bottom row) inputs. MSRP, Modified Stingray Robot Platform.

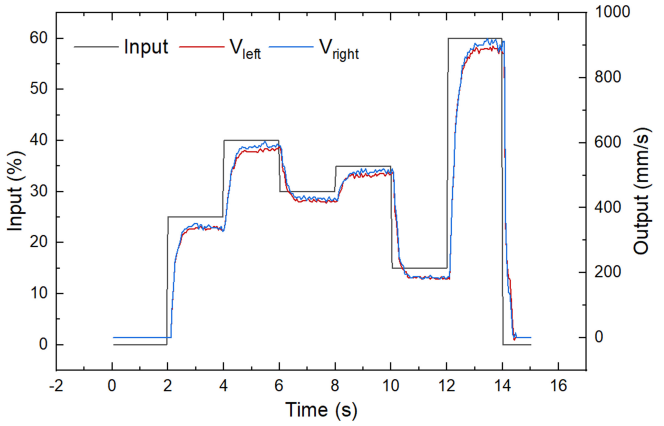

Fig. 9. MSRP response to Test XVII. MSRP, Modified Stingray Robot Platform.

$$y(t)=\begin{bmatrix}976.1 & -125.7\end{bmatrix}x(t)+0u(t) \quad (40b)$$

Each of the SS models was exercised using input from test XVII and plots for each model were created showing the input for test XVII, the actual output from the robot, and the predicted output. The results are shown in Fig. 10.

The SS model created by using test XI which utilized the effort-input function is shown in Fig. 10 (left). When the model was exercised using the input from test XVII, the model output has shown a good agreement with the actual output of each wheel. This similarity was then visualized by comparing the model output with the actual output of each wheel as shown in Fig. 10 (right). The SS model created by using test XV which utilized the velocity-input function is shown in Fig. 11. When the model was exercised using the input from test XVII, the model output was completely different than the actual output of each wheel as expected. This is because test XVII used the effort-input function. Thus, the SS model created using the results

TABLE IV. RESULTS OF THE FORWARD LOCOMOTION TEST

Test	Mode	*u*	G	τ	*u*	G	τ	*u*	G	τ
IX	E-B	15%	12.131	0.218	30%	14.175	0.237	25%	14.075	0.554
X	E-B	25%	14.225	0.241	40%	14.506	0.584	35%	14.281	0.534
XI	E-B	35%	14.283	0.525	50%	14.402	0.697	45%	14.446	0.571
XII	E-B	45%	14.454	0.633	60%	14.673	0.635	55%	14.558	0.786
XIII	V-B	0.15 m/s	1.039	0.507	0.30 m/s	1.057	0.457	0.25 m/s	1.033	0.566
XIV	V-B	0.25 m/s	1.026	0.515	0.40 m/s	1.038	0.447	0.35 m/s	1.069	0.374
XV	V-B	0.35 m/s	1.077	0.504	0.50 m/s	1.011	0.524	0.45 m/s	1.053	0.277
XVI	V-B	0.45 m/s	1.054	0.445	0.60 m/s	1.076	0.446	0.55 m/s	1.053	0.161
XVII	E-B	25%	14.078	0.624	40%	14.757	0.675	30%	13.839	0.453
XVII	E-B	35%	14.313	0.417	15%	13.338	0.362	60%	14.831	1.057

$u(k)$, input; G, gain; τ, time constant for step transition.

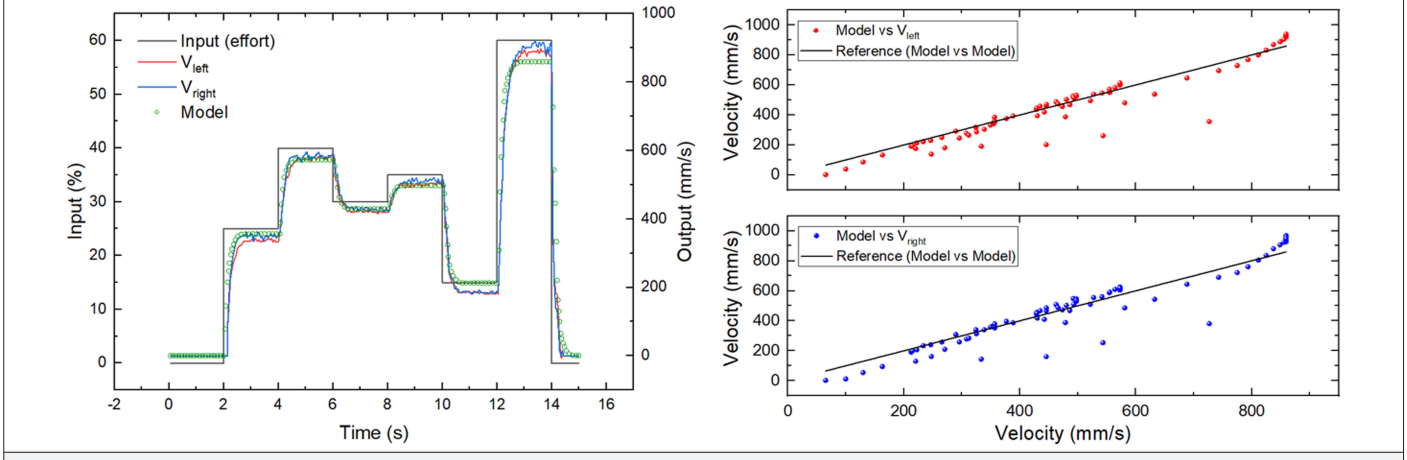

Fig. 10. State-space model output (test XI) for effort-input function.

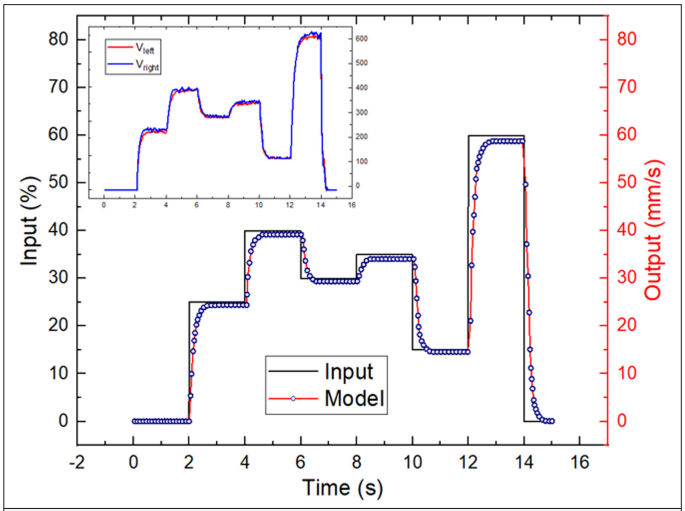

Fig. 11. State-space model output (test XV) for effort-input function.

of the effort-input function will map test XVII input to the expected output whereas using the velocity-input function, it will not.

A nonlinear model was also created and exercised with inputs from test XVII. To create the nonlinear model, the data file was accessed and data columns were again assigned to variables which represent required data arrays. The test XVII input, the actual output for each wheel, and the nonlinear model prediction were then graphed to analyze the results as shown in Fig. 12. The nonlinear model prediction shows extremely similar results as the actual output for each wheel. This similarity was also visualized by comparing the model output with the actual output of each wheel as shown on the right of Fig. 12. The main difference is that there is a larger amount of undershoot for the actual wheels when returning to 0 than the nonlinear model prediction. The nonlinear model produced an output more similar to the actual output than the SS model.

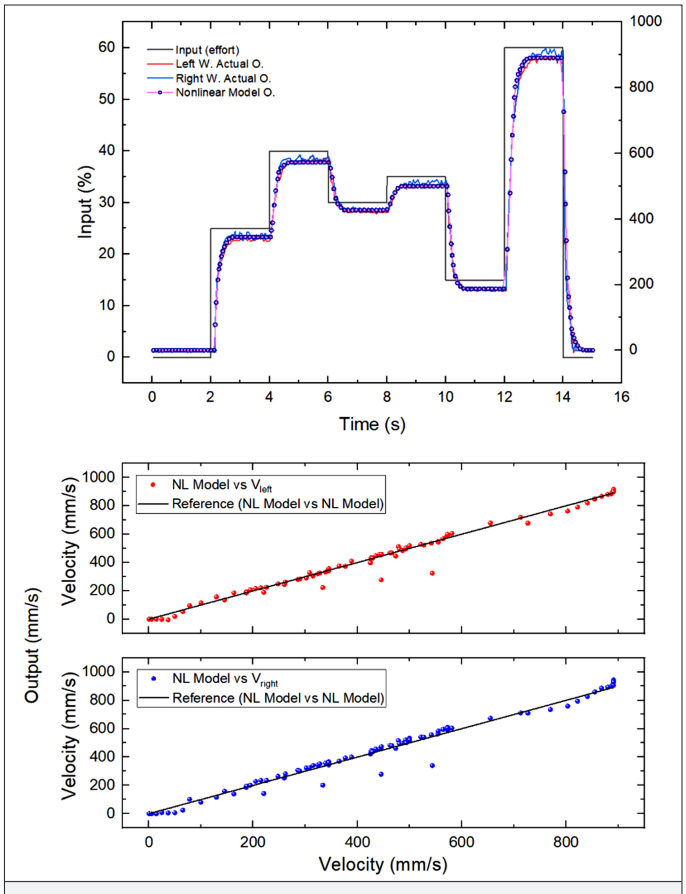

Fig. 12. Nonlinear model output for the effort-input function.

V. CONCLUSIONS

An accurate linearized model for a DDMR was developed to perform desired movements as well as return a file containing an array of experimental data to perform the identification procedure. Various tests have been conducted to identify key system parameters such as gain and time constant. Through system identification, a discrete time-invariant, SS model was generated using the obtained data sets. The predetermined input values were step inputs that yielded output, from which the gain and time constants were used to determine the SS models for the robot. A nonlinear model estimate was also generated and exercised with the system inputs. It was observed that while velocity mode seems to be more accurate and predictable, it had a more jagged steady state. In each test, the effort mode required longer times to change the velocity of the robot. It was also noticed that there were minor disparities between the left and right wheels in most cases, and it was troublesome to gauge exactly which one to go with. Additionally, the Matlab model buffers the noise from the signals to make them closer to how an ideal time-invariant system would behave. The SS models include buffering characteristics through the formulas that allow for the feedback and overshoot signals to become smoothed and ideal in a steady state characteristic. The assumption of the linearly time-invariant model allows for the SS formulas to be implemented so that these reproductions are developed in a relatively simple structure that does not unnecessarily complicate the control system.

Peer-review: Externally peer-reviewed.

Declaration of Interests: The author has no conflicts of interest to declare.

Funding: The author declared that this study has received no financial support.

REFERENCES

1. F. Spiess, L. Reinhart, N. Strobel, D. Kaiser, and S. Kounev, "People detection with depth silhouettes and convolutional neural networks on a mobile robot," *J. Image Grap*, vol. 9, no. 4, pp. 1–5, 2021.
2. W. Rahmanier and A. Wicaksono, "Design and implementation of a mobile robot for carbon monoxide monitoring," *J. Robo Cont.*, vol. 2, no. 1, pp. 1–6, 2021.
3. H. Huang, A. V. Savkin, M. Ding, and C. Huang, "Mobile robots in wireless sensor networks: A survey on tasks," *Comput. Netw.*, vol. 148, pp. 1–19, 2019. [CrossRef]
4. J. Szrek, J. Jakubiak, and R. Zimroz, "A mobile robot-based system for automatic inspection of belt conveyors in mining industry," *Energies*, vol. 15, no. 1, pp. 1–16, 2022.
5. K. Zhu and T. Zhang, "Deep reinforcement learning based mobile robot navigation: A review," *Tsinghua Sci. Technol.*, vol. 26, no. 5, pp. 674–691, 2021. [CrossRef]

6. T. Guo, J. Guo, B. Huang, and H. Peng, "Power consumption of tracked and wheeled small mobile robots on deformable terrains-model and experimental validation," *Mech. Mach. Theor.*, vol. 133, pp. 347–364, 2019. [CrossRef]
7. S. Wang, L. Jiang, J. Meng, Y. Xie, and H. Ding, "Training for smart manufacturing using a mobile robot-based production line," *Front. Mech. Eng.*, vol. 16, no. 2, pp. 249–270, 2021. [CrossRef]
8. A. M. Al-Obaidi, A. Al-Qassar, A. R. Nasser, A. Alkhayyat, A. J. Humaidi, and I. K. Ibraheem, "Embedded design and implementation of mobile robot for surveillance applications," *Indon J. Sci. Tech.*, vol. 6, no. 2, pp. 427–440, 2021.
9. J. Mu, X. G. Yan, S. K. Spurgeon, and Z. Mao, "Nonlinear sliding mode control of a two-wheeled mobile robot system," *Int. J. Model. Ident Cont.*, vol. 27, no. 2, pp. 75–83, 2009.
10. T. Wang, Y. Wu, J. Liang, C. Han, J. Chen, and Q. Zhao, "Analysis and experimental kinematics of a skid-steering wheeled robot based on a laser scanner sensor," *Sensors (Basel)*, vol. 15, no. 5, pp. 9681–9702, 2015. [CrossRef]
11. J. Zhai and Z. Song, "Adaptive sliding mode trajectory tracking control for wheeled mobile robots," *Int. J. Control*, vol. 92, no. 10, pp. 2255–2262, 2019. [CrossRef]
12. F. Bayat, S. Najafinia, and M. Aliyari, "Mobile robots path planning: Electrostatic potential field approach," *Expert Syst. Appl.*, vol. 100, pp. 68–78, 2018. [CrossRef]
13. G. Zhou, J. Luo, S. Xu, S. Zhang, S. Meng, and K. Xiang, "An EKF-based multiple data fusion for mobile robot indoor localization," *Assembly Autom.*, vol. 41, no. 3, pp. 274–282, 2021. [CrossRef]
14. G. Antonelli and S. Chiaverini, "Linear estimation of the physical odometric parameters for differential-drive mobile robots," *Auton. Robot.*, vol. 23, no. 1, pp. 59–68, 2007. [CrossRef]
15. M. Bakirci and B. Toptas, "Kinematics and autoregressive model analysis of a differential drive mobile robot," presented at the 2022 International Congress on Human-Computer Interaction, Optimization and Robotic Applications. Ankara, Turkey, Jun. 9–11, 2022. [CrossRef]
16. K. Y. Chen, Y. H. Lai, and R. F. Fung, "A comparison of fitness functions for identifying an LCD glass-handling robot system," *Mechatronics*, vol. 46, pp. 126–142, 2017. [CrossRef]
17. E. Olcay, C. Dengler, and B. Lohmann, "'Data-driven system identification of an innovation community model," *Int. Fed. Auto Cont.*, vol. 51, no. 11, pp. 1269–1274, 2018.
18. C. Urrea and J. Pascal, "Design, simulation, comparison and evaluation of parameter identification methods for an industrial robot," *Comp. Elec. Eng.*, vol. 67, pp. 791–806, 2018. [CrossRef]
19. P. C. Chen, "Using immune network in nonlinear system identification for a 3D parallel robot," *Information Technology J.*, vol. 8, no. 6, pp. 895–902, 2009. [CrossRef]
20. G. Zhao, P. Zhang, G. Ma, and W. Xiao, "System identification of the nonlinear residual errors of an industrial robot using massive measurements," *Robo. Comp.-Integ. Manu.*, vol. 59, pp. 104–114, 2019.
21. P. N. Guerra, P. J. Alsina, A. A. D. Medeiros, and A. P. Araujo Jr., "Linear modelling and identification of a mobile robot with differential drive," in 7th Conf. on Mobile Robo. and Compet. Portugal, 2004, pp. 1–8.
22. N. Mestre, M. P. Silva, C. Cardeira, and M. A. Botto, "Identification and control of an autonomous mobile robot," 7th Conf. on Mobile Robo. and Compet. Portugal, 2007, pp. 1–5.
23. G. Indiveri, A. Nüchter, and K. Lingemann, "High speed differential drive mobile robot path following control with bounded wheel speed commands," in Int. Conf. on robo. Italy, 2007, pp. 2202–2207. [CrossRef]
24. E. P. Mendes and A. A. D. Medeiros, "Identification of quasi-linear dynamic model with dead zone for mobile robot with differential drive," in Latin Ameri. Robo. Symp. And Intell. Robo. Meet. Brasil, 2010, pp. 132–137. [CrossRef]
25. S. Arslan and H. Temeltas, "Robust motion control of a four wheel drive skid-steered mobile robot," in 7th Int. Conf. on Electr. And Electro. Eng. Turkey, 2011, pp. 1–5.
26. P. Benavidez and M. Jamshidi, "Mobile robot navigation and target tracking system," 6th Int. Conf. on Syst. of Syst. Eng., USA, 2011, pp. 299–304. [CrossRef]
27. C. K. Volos, I. M. Kyprianidis, and I. N. Stouboulos, "A chaotic path planning generator for autonomous mobile robots," *Robo. and Auton. Syst.*, vol. 60, pp. 1–6, 2012.
28. P. Stegagno, M. Cognetti, L. Rosa, P. Peliti, and G. Oriolo, "Relative localization and identification in a heterogeneous multi-robot system," in IEEE Int. Conf. on Robo. And Automa. Germany, 2013, pp. 1857–1864. [CrossRef]
29. R. Dhaouadi and A. A. Hatab, "Dynamic modelling of differential-drive mobile robots using Lagrange and Newton-Euler methodologies: A unified framework," *Adv. Robo Autom.*, vol. 2, no. 2, pp. 1–7, 2013.
30. S. K. Malu, and J. Majumdar, "'Kinematics, localization and control of differential drive mobile robot," *Glob. J. of Res. in Eng.: H Robo. & Nano-Tech.*, vol. 14, no. 1, pp. 1–9, 2014.
31. E. Maulana, M. A. Muslim, and A. Zainuri, "Inverse kinematics of a two-wheeled differential drive an autonomous mobile robot," in Elec. Pow. Electron. Commun. Cont. And Inform. Semin. Indonesia, pp. 1–6, 2014. [CrossRef]
32. A. Corrales, M. Malfaz, and M. A. Salichs, "Signage system for the navigation of autonomous robots in indoor environments," *IEEE Trans. Indust. Inform.*, vol. 10, no. 1, pp. 680–688, 2014.
33. F. G. Rossomando, and C. M. Soria, "Identification and control of nonlinear dynamics of a mobile robot in discrete time using an adaptive technique based on neural PID," *Neural Comput. Applic.*, vol. 26, no. 5, pp. 1179–1191, 2015. [CrossRef]
34. A. Pandey, S. Jha, and D. Chakravarty, "Modeling and control of an autonomous three wheeled mobile robot with front steer," in 1st IEEE Int. Conf. on Robo. Computing, 2017, pp. 136–142. [CrossRef]
35. S. Konduri, E. O. C. Torres, and P. R. Pagilla, "Dynamics and control of a differential drive robot with wheel slip: Application to coordination of multiple robots," *J. Dyn. Syst., Meas., Control*, vol. 139, no. 1, pp. 14505–14510, 2017. [CrossRef]
36. I. F. Okuyama, M. R. O. A. Maximo, A. L. O. Cavalcanti, and R. J. M. Afonso, "Nonlinear grey-box identification of a differential drive mobile robot," in XIII Simposio Brasileiro de Automaçao Inteligente. Brasil, 2017, pp. 1–6.
37. F. N. Martins, M. Sarcinelli-Filho, and R. Carelli, "A velocity-based dynamic model and its properties for differential drive mobile robots," *J. Intell. Robot. Syst.*, vol. 85, no. 2, pp. 277–292, 2017. [CrossRef]
38. A. Said, Y. Davizon, R. Soto, C. Felix-Herran, C. Hernandez-Santos, and P. Espino-Roman, "An infinite-norm algorithm for joystick kinematic control of two-wheeled vehicles," *Electronics*, vol. 7, no. 164, pp. 1–15, 2018.
39. T. G. Alves, W. F. Lages, and R. V. B. Henriques, "Parametric identification and controller design for a differential drive mobile robot," *IFAC PapersOnLine*, vol. 51, no. 15, pp. 437–442, 2018. [CrossRef]
40. A. G. C. Gonzales, M. V. S. Alves, G. S. Viana, L. K. Carvalho, and J. C. Basilio, "Supervisory control-based navigation architecture: A new framework for autonomous robots in industry 4.0 environments," *IEEE Trans. on Imdust. Inform*, vol. 14, no. 4, pp. 1732–1743, 2018. [CrossRef]
41. D. Xie, S. Wang, and Y. Wang, "Trajectory tracking control of differential drive mobile robot based on improved kinematics controller algorithm," in Chinese Autom. Congress, China, 2018, pp. 2675–2680. [CrossRef]
42. T. G. Alves, W. F. Lages, and R. V. B. Henriques, "Non-linear pose stabilization controller for a differential-drive mobile robot: Optimization-based controller tuning," *IFAC PapersOnLine*, vol. 51, no. 22, pp. 536–541, 2018. [CrossRef]
43. P. Yazdjerdi and N. Meskin, "Design and real-time implementation of actuator fault-tolerant control for differential-drive mobile robots based on multiple-model approach," *J. Syst. Cont. Eng.*, vol. 232, no. 6, pp. 652–661, 2018. [CrossRef]
44. S. Bouzoualegh, E. Guechi, and R. Kelaiaia, "Model predictive control of a differential-drive mobile robot," *Acta Univ. Sapientiae Electr. Mech. Eng.*, vol.10, no.1, pp.20–41, 2018. [CrossRef]
45. B. Raafiu and P. A. Darwito, "Identification of four wheel mobile robot based on parametric modeling," in Int. Seminar on Intell. Tech. And Its Appli., Indonesia, 2018, pp. 1–5.
46. A. N. Amundhan, P. Sakthivel, A. P. Sudheer, and T. K. S. Kumar, "Design of controllers for omnidirectional robot based on the system

identification technique for trajectory tracking," *IOP Conf. Series: J. Phys. Conf. S.*, vol. 1240, pp. 1–9, 2019.
47. T. Q. Khai, Y. J. Ryoo, W. R. Gill, and D. Y. Im, "Design of kinematic controller based on parameter tuning by fuzzy inference system for trajectory tracking of differential-drive mobile robot," *Int. J. Fuzzy Syst.*, vol. 22, no. 6, pp. 1972–1978, 2020. [CrossRef]
48. Y. Luo, A. Boularias, and M. Aanjaneya, "Model identification and control of a low-cost wheeled mobile robot using differentiable physics," *arXiv Preprint ArXiv:2009.11465*, 2009.
49. F. Ardiani, M. Benoussaad, and A. Janot, "Comparison of least-squares and instrumental variables for parameters estimation on differential drive mobile robots," *IFAC PapersOnLine*, vol. 54, no. 7, pp. 310–315, 2021. [CrossRef]

Dr. Murat Bakirci received his M.S. and Ph.D. degrees from Old Dominion University, Norfolk, Virginia, USA. He was a research assistant at the Transportation Research Institute, Virginia Modeling Analysis and Simulation Center, and Collaborative Autonomous Systems Laboratory. Currently, he holds a lecturer position at the Faculty of Aeronautics and Astronautics at Tarsus University. His research interests include unmanned systems, intelligent transportation systems, robotics, avionics, MEMS sensor applications, and micro/nanofluidics.

APPENDIX: DERIVATION OF DISCRETE ODOMETRIC MODEL OF THE ROBOT

The following relations can be deduced from Fig. A1, where d , d_L and d_R are the distance traveled by the center, left wheel and right wheel of the robot, respectively.

$$\mathrm{d} = \mathrm{R}\Delta\theta \tag{A1}$$

$$\mathrm{d_L} = \left(\mathrm{R} - \frac{\mathrm{w}}{2}\right)\Delta\theta \tag{A2}$$

$$\mathrm{d_R} = \left(\mathrm{R} + \frac{\mathrm{w}}{2}\right)\Delta\theta \tag{A3}$$

When (A2) is multiplied by -1 and added by (A3), the following expression is obtained.

$$\Delta\theta = \frac{\mathrm{d_R} - \mathrm{d_L}}{\mathrm{w}} \tag{A4}$$

Substituting (A4) in (A2) and solving for *R* yields the following expression.

$$\mathrm{R} = \frac{\mathrm{w}}{\mathrm{d_R} - \mathrm{d_L}} \frac{\mathrm{d_R} + \mathrm{d_L}}{2} \tag{A5}$$

Then, substituting (A4) and (A5) in (A1) gives the distance traveled as expressed below.

$$\mathrm{d} = \frac{\mathrm{d_R} + \mathrm{d_L}}{2} \tag{A6}$$

The following relation can be deduced from Fig. A2(i), where *a* is the angle between the horizontal and the distance traveled between two consecutive positions.

$$\mathrm{a} = \frac{\Delta\theta}{2} \tag{A7}$$

The new orientation of the robot, *b*, is obtained from Fig. A2(ii) as follows.

$$\mathrm{b} = \Delta\theta \tag{A8}$$

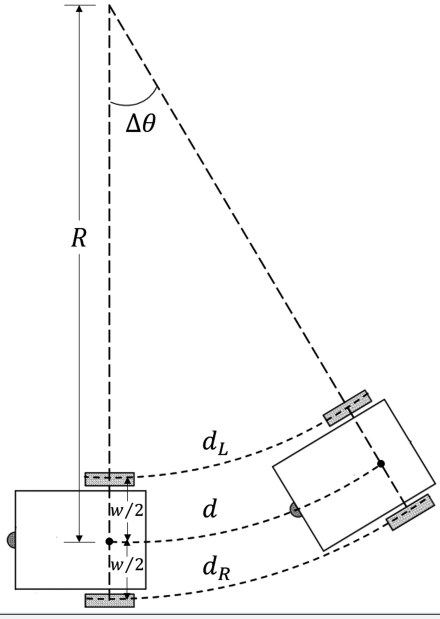

Fig. A1. Nonlinear model output for the effort-input function.

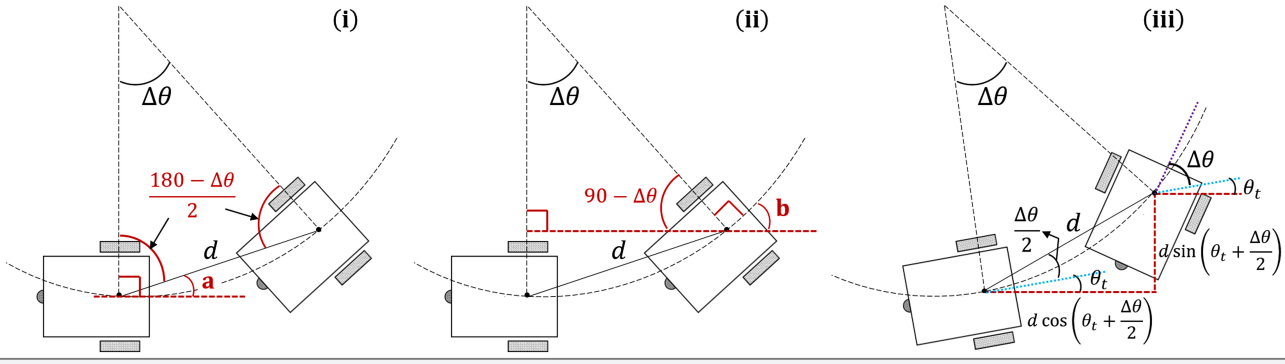

Fig. A2. Nonlinear model output for the effort-input function.

Thus, all necessary parameters have been derived to obtain the discrete odometric model of the robot. Finally, starting from Fig. A2(iii), the following model can be obtained for x, y, and θ.

$$x_{t+1} = x_t + d\cos\left(\theta_t + \frac{\Delta\theta_t}{2}\right) \tag{A9}$$

$$y_{t+1} = y_t + d\sin\left(\theta_t + \frac{\Delta\theta_t}{2}\right) \tag{A10}$$

$$\theta_{t+1} = \theta_t + \Delta\theta_t \tag{A11}$$

Noting that $\Delta\theta = dt\omega$ and $d = dt\left(\left(V_L + V_R\right)/2\right)$, the following final form (24) can be obtained.

$$\begin{bmatrix} x \\ y \\ \theta \end{bmatrix}_{t+1} = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}_t + \begin{bmatrix} dt\dfrac{V_L + V_R}{2}\cos\left(\theta_t + \dfrac{dt\omega_t}{2}\right) \\ dt\dfrac{V_L + V_R}{2}\sin\left(\theta_t + \dfrac{dt\omega_t}{2}\right) \\ dt\omega_t \end{bmatrix} \tag{A12}$$