International Journal of Control, Automation, and Systems 22(7) (2024) 2273-2282
http://dx.doi.org/10.1007/s12555-023-0288-8
ISSN:1598-6446 eISSN:2005-4092
http://www.springer.com/12555

Design and Modeling of a Two-wheeled Differential Drive Robot

John Moritz*, Mishek Musa, and Uche Wejinya

Abstract: There are many existing publications on balancing two-wheeled, differential drive robot (TWDDR) covering dynamic modeling, kinematic modeling, path planning, control architecture design and/or simulations. However, there are few papers that cover all of these in a comprehensive manner that is approachable to beginner robotics researchers. This paper provides step-by-step details of the robotic design process including dynamic modeling, kinematic modeling, linearization, autonomous navigation, path planning, and stability control. A cascaded PID control architecture is presented that is capable of stabilizing the robot in less than 1 s with minimal steady-state error and performing large force and torque disturbance rejection. Additionally, a high-level path planning algorithm based on artificial potential fields is demonstrated.

Keywords: Artificial potential fields, mobile robotics, nonlinear dynamics, PID control.

1. INTRODUCTION

Two-wheel, differential-drive robot (TWDDR) is a system with properties analogous to the classic inverted pendulum system, which has one degree of freedom. The inverted pendulum consists of a mass balanced above a pivot point. The pivot point is attached to a linear actuator, or cart, that stabilizes the pendulum by pushing the pivot point underneath the mass. This underactuated system has been investigated by researchers for decades due to its instability and non-linearity, and it is often used as a test for new controllers in development.

The TWDDR differs from the inverted pendulum in that it has 3 degrees of freedom: in-plane translation, and rotation about its vertical axis (yaw). Compared to other types of mobile robots, the TWDDR has a high degree of maneuverability in narrow spaces such as hallways, sidewalks, and warehouse environments. This makes TWDDR highly suited for applications in last-mile delivery, pedestrian transportation, warehouse automation, etc. The inverted pendulum problem has already been implemented in many technologies outside of academia, e.g., SpaceX's Falcon 9 landing function [1], Segway HT [2], Boston Dynamics's Handle [3], etc.

Typically, the dynamic model is developed using the Euler-Lagrange approach [4], which is used for the inverted pendulum due to the convenience of describing the system's motion by its energies (scalar quantities) rather than forces (vector quantities). Many designs include a third castor wheel to keep it stabilized, eliminating a degree of freedom for the system [5]. This significantly simplifies the dynamic model as well as the control model for the robot. The existence of a castor wheel can impede the robot's maneuverability in very narrow or cluttered spaces.

A large number of the existing research on TWDDR have a scope limited to only one or two of the following: dynamic modeling, kinematic modeling, stability control, path planning, etc. Some of the works that focus on system modeling include the following. Kim *et al.* presented a step-by-step process for modeling the TWDDR using the Euler-Lagrange energy method and Kane's momentum method [6]. However, they do not present any control method for this model. Similarly, Hatab *et al.* presents the Euler-Lagrange method and the Newton-Euler method to model the dynamics for the TWDDR [7]. They also present DC motor actuator modeling, but they do not present any control algorithm. Siradjuddin *et al.* model their dynamics with the Euler-Lagrange method, however, they fail to include non-holonomic constraints, only presenting simulations for pitch control using a PID controller [4].

On the other hand, other works prioritize the control system design. Owojori *et al.* present a design for a TWDDR that uses a LQR controller to optimally control the robot pitch [8]. However, their design does not control yaw angle, and therefore has only one degree of freedom (1-DOF). Similarly, Mathew *et al.* proposed an optimal LQR controller for the upright balance of a self-balancing robot, however, no consideration was given to the complete system [9]. Lin *et al.* present a control model that focuses on implementation of hardware without detailing dynamic

Manuscript received May 24, 2023; revised November 17, 2023; accepted April 11, 2024. Recommended by Associate Editor Ning Sun under the direction of Senior Editor Dongjun Lee. The authors thank the Department of Mechanical Engineering, Twenty-First Century Professorship grant for supporting this work.

John Moritz, Mishek Musa, and Uche Wejinya are with the Department of Mechanical Engineering, University of Arkansas, 863 W. Dickson St., Fayetteville, AR, USA (e-mails: {jamoritz, mjmusa, uwejinya}@uark.edu).
* Corresponding author.

modeling [10]. Suprapto *et al.* presented a comparative study on PI, PD, and PID controllers for the robot, however little to no consideration was given to the dynamic modeling of the system [11].

In addition to the system modeling and control system design, High-level path planning algorithms are frequently integrated into robotic systems, but their application in the context of the TWDDR is rarely documented. Some of the common path planning algorithms include A-star [12], artificial potential fields (APF) [13], Dijkstra's algorithm [14], rapidly exploring random trees (RRT) [15], probabilistic road maps (PRM) [16], and others. One approach of particular interest is the method of artificial potential fields. The technique is well suited for local path planning and obstacle avoidance, is conceptually simple and has ease of implementation, and can produce computationally efficient responsiveness.

In this work, a dynamic model using the Euler-Lagrange approach while including non-holonomic constraints, the inverse kinematic equations, the control architecture for the three degrees of freedom (3-DOF), and a path planning algorithm based on artificial potential fields, are developed. Simulation results are presented to test the entire developed system with the overarching goal of demonstrating both the high-level and low-level performance of the TWDDR. Section 2 describes the coordinate frames used to model the system, Section 3 presents the kinematic model, while Section 4 describes the dynamic model, Section 5 describes the path planning algorithm, Section 6 goes over the PID control architecture, Section 7 describes the simulation experiments, and lastly Section 8 is the conclusion.

2. COORDINATE FRAMES

Prior to modeling, it is important to define the three coordinate frames of the system. There is the fixed global frame denoted by the subscript G, the chassis frame is denoted by C, and the robot body frame is denoted by B. The robot consists of three rigid bodies: left wheel, right wheel, and the body, denoted by the subscript L, R, and b, respectively. The rotation from frame C with respect to B by angle α is given by the rotational matrix in (1) as follows:

$$ {}^{B}_{C}R(\alpha) = \begin{bmatrix} \cos(\alpha) & 0 & -\sin(\alpha) \\ 0 & 1 & 0 \\ \sin(\alpha) & 0 & \cos(\alpha) \end{bmatrix}. \tag{1}$$

Fig. 1 shows the three different coordinate systems, as well as the positional and rotational vectors.

$$\vec{\omega}_c = \begin{bmatrix} 0 \\ 0 \\ \dot{\theta} \end{bmatrix} ; \ \vec{\omega}_b = \begin{bmatrix} 0 \\ \dot{\alpha} \\ 0 \end{bmatrix}. \tag{2}$$

The rotational vectors for the chassis frame and body frame are given in (2).

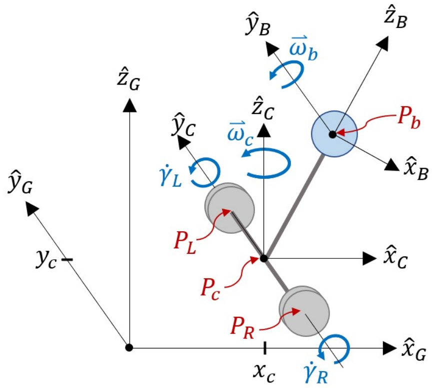

Fig. 1. TWDDR coordinate frames.

3. KINEMATIC MODEL

3.1. Kinematic constraints

The TWDDR is subject to both holonomic and non-holonomic constraints. Holonomic constraints are constraints on configuration that reduces the total number of controllable degrees of freedom. Non-holonomic constraints are velocity constraints that cannot be integrated to an equivalent configuration. There are three holonomic constraints relating the left and right wheels, and the body. Fig. 2 displays the TWBR in the Navigation Plane, helping visualize the multibody constraints.

The center of mass of each of the three rigid bodies are constrained by the following holonomic conditions, with respect to the global frame as shown in (3)-(5)

$$P_b = \begin{bmatrix} x_b \\ y_b \\ z_b \end{bmatrix} = \begin{bmatrix} x_c + L\sin\alpha\cos\theta \\ y_c + L\sin\alpha\sin\theta \\ L\cos\alpha \end{bmatrix}, \tag{3}$$

$$P_L = \begin{bmatrix} x_L \\ y_L \\ z_L \end{bmatrix} = \begin{bmatrix} x_c - \frac{d}{2}\sin\theta \\ y_c + \frac{d}{2}\cos\theta \\ 0 \end{bmatrix}, \tag{4}$$

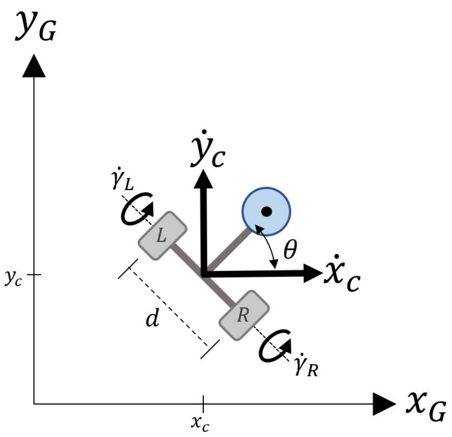

Fig. 2. TWDDR configuration in global navigation plane.

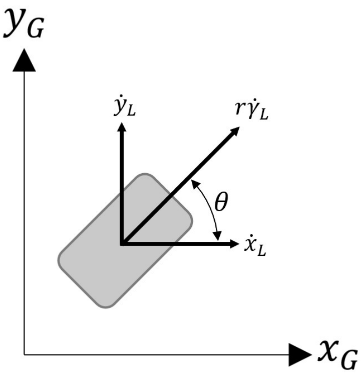

Fig. 3. Wheel velocity kinematics.

$$P_R = \begin{bmatrix} x_R \\ y_R \\ z_R \end{bmatrix} = \begin{bmatrix} x_c + \frac{d}{2}\sin\theta \\ y_c - \frac{d}{2}\cos\theta \\ 0 \end{bmatrix}. \tag{5}$$

The two non-holonomic constraints are "pure rolling" of the wheels, and "non-drifting", or zero velocity in the chassis frame's $\vec{y}$ direction. Fig. 3 helps visualize the wheel velocity vectors. The pure rolling constraints can be represented by (6) and (7)

$$\dot{x}_L \cos\theta + \dot{y}_L \sin\theta = r\dot{\gamma}_L, \tag{6}$$

$$\dot{x}_R \cos\theta + \dot{y}_R \sin\theta = r\dot{\gamma}_R, \tag{7}$$

which can be substituted with (4) and (5) to result in (8) and (9)

$$\dot{x}_c \cos\theta + \dot{y}_c \sin\theta - \frac{d}{2}\dot{\theta} - r\dot{\gamma}_L = 0, \tag{8}$$

$$\dot{x}_c \cos\theta + \dot{y}_c \sin\theta + \frac{d}{2}\dot{\theta} - r\dot{\gamma}_R = 0. \tag{9}$$

The mapping of the chassis velocity from the global frame to the chassis frame can be summarized by (10) [6,7,17-21]

$$\dot{x}_c \cos\theta - \dot{y}_c \sin\theta = \dot{x}. \tag{10}$$

The non-lateral drifting constraint is represented by zero-velocity in the direction perpendicular to the heading velocity, as shown in (11)

$$\dot{x}_c \sin\theta - \dot{y}_c \cos\theta = 0. \tag{11}$$

3.2. Navigational kinematics

In the study of mobile robotics, inverse kinematics is the most appropriate since the objective is to find the motor inputs necessary to achieve a desired motion and configuration. Firstly, the linear and rotational velocities are related to wheel angular velocities. Consider the case where the left wheel has a slower linear velocity than the right wheel. In this case, the robot will be rotating about a point called the instantaneous center of curvature (ICC). The robot's linear velocity, v_c, is described by (12)

$$v_c = \dot{\theta}R, \tag{12}$$

where R is the distance from the ICC to the midpoint between the two wheels and $\dot{\theta}$ is the angular velocity of the robot turning. Equation (12) can be altered to describe the wheel velocities both in terms of their individual rotational velocities about y_c, and in terms of their rotational yaw velocity about the ICC, as shown in (13) and (14).

$$\vec{v}_L = \dot{\gamma}_L r = \dot{\theta}(R+d), \tag{13}$$

$$\vec{v}_R = \dot{\gamma}_R r = \dot{\theta}(R-d). \tag{14}$$

Solving (12) for R and substituting it into (13) and (14) will result in the relationships shown in (15) and (16) [6].

$$\dot{\gamma}_L = \frac{\dot{x} + \dot{\theta}d}{r}, \tag{15}$$

$$\dot{\gamma}_R = \frac{\dot{x} - \dot{\theta}d}{r}. \tag{16}$$

Here, we have the inverse kinematic equations that describes the individual wheel velocities written in terms of linear velocity and angular yaw velocity.

4. DYNAMIC MODEL

The Euler-Lagrange method is used to determine the equations of motion using the Lagrangian function as shown in (17) [17].

$$L = T - V, \tag{17}$$

where L is the Lagrangian function, T is the kinetic energy, and V is the potential energy. We begin by describing the kinetic and potential energies of each of the three rigid bodies we have defined. The Kinetic energy of our system, as shown in (18) and (19), is categorized by two types, translational and rotational, which is given by the subscripts *trans* and *rot*, respectively.

$$T_{trans} = \frac{1}{2}m(\vec{v}_b^T \times \vec{v}_b) + \frac{1}{2}M(\vec{v}_L^T \times \vec{v}_L) + \frac{1}{2}M(\vec{v}_R^T \times \vec{v}_R), \tag{18}$$

$$T_{rot} = \frac{1}{2}\vec{\omega}_b^T \vec{I}_b \vec{\omega}_b + \frac{1}{2}\vec{\omega}_L^T \vec{I}_w \vec{\omega}_L + \frac{1}{2}\vec{\omega}_R^T \vec{I}_w \vec{\omega}_R. \tag{19}$$

Here, m is the mass of the body, M is the mass of each wheel, and the inertia matrices are shown in (20) and (21).

$$\vec{I}_b = diag\{I_{\hat{x}}, I_{\hat{y}}, I_{\hat{z}}\}, \tag{20}$$

$$\vec{I}_w = diag\{K, J, K\}. \tag{21}$$

By taking the time derivative of (3)-(5) we find the velocity vectors of our rigid bodies, with respect to the fixed global frame, as shown in (22)

$$\vec{v}_b = \begin{bmatrix} \dot{x}_b \\ \dot{y}_b \\ \dot{z}_b \end{bmatrix} = \begin{bmatrix} \dot{x}_c - \dot{\alpha} L \cos\alpha \cos\theta - \dot{\theta} L \sin\alpha \sin\theta \\ \dot{y}_c + \dot{\alpha} L \cos\alpha \sin\theta + \dot{\theta} L \sin\alpha \cos\theta \\ -\dot{\alpha} L \sin\alpha \end{bmatrix}, \tag{22}$$

$$\vec{v}_L = \begin{bmatrix} \dot{x}_L \\ \dot{y}_L \\ \dot{z}_L \end{bmatrix} = \begin{bmatrix} \dot{x}_c - \dot{\theta}\frac{d}{2}\cos\theta \\ \dot{y}_c - \dot{\theta}\frac{d}{2}\sin\theta \\ 0 \end{bmatrix}, \tag{23}$$

$$\vec{v}_R = \begin{bmatrix} \dot{x}_R \\ \dot{y}_R \\ \dot{z}_R \end{bmatrix} = \begin{bmatrix} \dot{x}_c + \dot{\theta}\frac{d}{2}\cos\theta \\ \dot{y}_c + \dot{\theta}\frac{d}{2}\sin\theta \\ 0 \end{bmatrix}. \tag{24}$$

The angular velocities of the three rigid bodies with respect to their moving frames C or B are as follows [6]:

$$\vec{\omega}_b^C = {}^B_C R(\alpha) \times \vec{\omega}_c + {}^B\vec{\omega}_b = \begin{bmatrix} -\dot{\theta}\sin\alpha \\ \dot{\alpha} \\ \dot{\theta}\cos\alpha \end{bmatrix}, \tag{25}$$

$$\vec{\omega}_L = \vec{\omega}_C + \begin{bmatrix} 0 \\ \dot{\gamma}_L \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ \dot{\gamma}_L \\ \dot{\theta} \end{bmatrix}, \tag{26}$$

$$\vec{\omega}_R = \vec{\omega}_C + \begin{bmatrix} 0 \\ \dot{\gamma}_R \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ \dot{\gamma}_R \\ \dot{\theta} \end{bmatrix}. \tag{27}$$

By substituting (20)-(27) into (18) and (19) we find that our energies are given by

$$\begin{aligned} T_{trans} = {} & \left(M + \frac{m}{2}\right)\dot{x}_c^2 + \left(M + \frac{m}{2}\right)\dot{y}_c^2 + M\dot{\theta}^2\frac{d^2}{4} \\ & + \frac{m}{2}\dot{\alpha}^2 L^2 + \frac{m}{2}\dot{\theta} L^2 \sin^2\alpha + m\dot{x}_c\dot{\alpha} L\cos\alpha\sin\theta \\ & - m\dot{x}_c\dot{\theta} L\sin\alpha\sin\theta + m\dot{y}_c\dot{\alpha} L\cos\alpha\sin\theta \\ & + m\dot{y}_c\dot{\theta} L\sin\alpha\cos\theta, \end{aligned} \tag{28}$$

$$\begin{aligned} T_{rot} = {} & 0.5 I_{\hat{x}}\dot{\theta}^2\sin^2\alpha + 0.5 I_{\hat{z}}\dot{\theta}^2\cos^2\alpha + 0.5 I_{\hat{z}}\dot{\alpha}^2 \\ & + \frac{J}{2}(\dot{\gamma}_L^2 + \dot{\gamma}_r^2) + \dot{\theta}^2 K, \end{aligned} \tag{29}$$

$$V = mgL\cos\alpha. \tag{30}$$

By substituting our kinetic and potential energies into (17), we can formulate our Lagrangian function as follows in (31)

$$\begin{aligned} L = {} & \left(M + \frac{m}{2}\right)\dot{x}_c^2 + \left(M + \frac{m}{2}\right)\dot{y}_c^2 + M\dot{\theta}^2\frac{d^2}{4} + \frac{m}{2}\dot{\alpha}^2 L^2 \\ & + \frac{m}{2}\dot{\theta} L^2\sin^2\alpha + m\dot{x}_c\dot{\alpha} L\cos\alpha\sin\theta \\ & - m\dot{x}_c\dot{\theta} L\sin\alpha\sin\theta + m\dot{y}_c\dot{\alpha} L\cos\alpha\sin\theta \\ & + m\dot{y}_c\dot{\theta} L\sin\alpha\cos\theta + 0.5 I_{\hat{x}}\dot{\theta}^2\sin^2\alpha \\ & + 0.5 I_{\hat{z}}\dot{\theta}^2\cos^2\alpha + 0.5 I_{\hat{z}}\dot{\alpha}^2 + \frac{J}{2}(\dot{\gamma}_L^2 + \dot{\gamma}_r^2) \\ & + \dot{\theta}^2 K - mgL\cos\alpha. \end{aligned} \tag{31}$$

With our newly formulated Lagrangian function, we see that it is a function of six generalized coordinates q_i, which is given by

$$q = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \end{bmatrix} = \begin{bmatrix} x_c \\ y_c \\ \alpha \\ \theta \\ \gamma_L \\ \gamma_R \end{bmatrix}. \tag{32}$$

The Euler-Lagrange method does not inherently account for non-holonomic constraints and non-conservative forces, such as friction. Therefore, we use generalized forces and the Lagrange multiplier method to include these, as shown in (33) [17].

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = Q_i + C_i. \tag{33}$$

Q_i represents the generalized forces such as inputs and non-conservative forces, and C_i represents the generalized constraint forces. For $i = 1, \ldots, 6$, we find Q_i to be describe by (34)-(39)

$$Q_1 = 0, \tag{34}$$

$$Q_2 = 0, \tag{35}$$

$$Q_3 = -(Q_5 = Q_6), \tag{36}$$

$$Q_4 = 0, \tag{37}$$

$$Q_5 = F_L - c(\dot{\gamma}_L - \dot{\alpha}), \tag{38}$$

$$Q_6 = F_R - c(\dot{\gamma}_R - \dot{\alpha}), \tag{39}$$

and the generalized constraint forces are given by (40)

$$C_i = \sum_{j=1}^{3} \lambda_j a_{j,i}, \tag{40}$$

where λ_j represents the Lagrange multipliers, and $a_{j,i}$ represents the constraint functions. The constraint functions are as follows in (41)-(43) [6,20]:

$$a_{1,i} = \frac{\partial}{\partial \dot{q}_i}(\dot{x}_c\cos\theta + \dot{y}_c\sin\theta - \frac{d}{2}\dot{\theta} - r\dot{\gamma}_L), \tag{41}$$

$$a_{2,i} = \frac{\partial}{\partial \dot{q}_i}(\dot{x}_c\cos\theta + \dot{y}_c\sin\theta + \frac{d}{2}\dot{\theta} - r\dot{\gamma}_R), \tag{42}$$

$$a_{3,i} = \frac{\partial}{\partial \dot{q}_i}(\dot{x}_c\sin\theta + \dot{y}_c\cos\theta). \tag{43}$$

Expanding the Euler-Lagrange equations of motion with respect to the generalized coordinates gives (44)-(49).

$$\begin{aligned} & \ddot{x}_c(2M + m) + \ddot{\alpha} mL\cos\alpha\sin\theta - \ddot{\theta} mL\sin\alpha\sin\theta \\ & \quad - \dot{\alpha}^2 mL\sin\alpha\sin\theta - \dot{\theta}^2 mL\sin\alpha\cos\theta \\ & \quad + \dot{\alpha}\dot{\theta} mL\cos\alpha\cos\theta - \dot{\alpha}\dot{\theta} mL\cos\alpha\sin\theta \\ & = \lambda_1\cos\theta + \lambda_2\cos\theta + \lambda_3\sin\theta, \end{aligned} \tag{44}$$

$$\begin{aligned}
&\ddot{y}_c(2M+m)+\ddot{\alpha}mL\cos\alpha\sin\theta+\ddot{\theta}mL\sin\alpha\sin\theta\\
&\quad-\dot{\alpha}^2mL\sin\alpha\sin\theta+\dot{\theta}^2mL\sin\alpha\cos\theta\\
&\quad+\dot{\alpha}\dot{\theta}mL\sin\alpha\cos\theta+\dot{\alpha}\dot{\theta}mL\cos\alpha\sin\theta\\
&=\lambda_1\sin\theta+\lambda_2\sin\theta+\lambda_3\cos\theta, \qquad (45)
\end{aligned}$$

$$\begin{aligned}
&\ddot{\alpha}(mL^2+I_{\hat{y}})+\ddot{x}_cmL\cos\alpha\sin\theta+\ddot{y}_cmL\cos\alpha\sin\theta\\
&\quad+\dot{x}_c\dot{\theta}mL\cos\alpha(\cos\theta+\sin\theta)-0.5\dot{\theta}mL^2\sin(2\alpha)\\
&\quad-0.5\dot{\theta}^2I_{\hat{x}}\sin(2\alpha)+\dot{\theta}^2I_{\hat{z}}\sin\alpha\cos\alpha-mgL\sin\alpha\\
&=-(F_L+F_R)+c(\dot{\gamma}_L+\dot{\gamma}_R)-2\dot{\theta}c, \qquad (46)
\end{aligned}$$

$$\begin{aligned}
&0.5\ddot{\theta}Md^2+2K+I_{\hat{x}}\sin^2\alpha+I_{\hat{z}}\cos^2\alpha\\
&\quad-\ddot{x}_cmL\sin\alpha\sin\alpha+\ddot{y}_cmL\sin\alpha\cos\alpha\\
&\quad+0.5\dot{\alpha}mL^2\sin(2\alpha)-\dot{x}_c\dot{\alpha}mL\cos\alpha(\sin\theta+\cos\theta)\\
&=-\lambda_1\frac{d}{2}+\lambda_2\frac{d}{2}, \qquad (47)
\end{aligned}$$

$$J\ddot{\gamma}_L=F_L-c(\dot{\gamma}_L-\dot{\theta})-\lambda_1 r, \qquad (48)$$

$$J\ddot{\gamma}_R=F_R-c(\dot{\gamma}_R-\dot{\theta})-\lambda_2 r. \qquad (49)$$

The first two Lagrange multiplier terms λ_1 and λ_2 can be eliminated by solving (48) and (49) for them, and then substituting them into (44), (45), and (47). We can also solve for λ_3 using (44). Lastly, by applying the relationships in (10), (15), and (16), we find that we have equations of motion with respect to 3 generalized coordinates, x, α, and θ, as shown in (50)-(52).

$$\begin{aligned}
\ddot{x}=&\Big(\frac{u}{r}+mL(\dot{\theta}^2+\dot{\alpha})\sin\alpha-\cos\alpha mL\ddot{\alpha}\\
&-\frac{2c}{r}\Big(\frac{\dot{x}}{r}-\dot{\alpha}\Big)\Big)\Big[2M+m+\frac{2J}{r^2}\Big]^{-1}, \qquad (50)
\end{aligned}$$

$$\begin{aligned}
\ddot{\alpha}=&(-u+2c(\frac{\dot{x}}{r}-\dot{\alpha})+mgL\sin\alpha\\
&-\ddot{x}mL\cos\alpha\dot{\theta}^2\cos\alpha\sin\alpha(I_{\hat{z}}-I_{\hat{x}}-mL^2))\\
&\times(I_{\hat{y}}+mL^2)^{-1}, \qquad (51)
\end{aligned}$$

$$\begin{aligned}
\ddot{\theta}=&\Big(\frac{\phi}{2r}-(mL\dot{x}+2\dot{\alpha}\cos\alpha(I_{\hat{x}}+mL^2-I_{\hat{z}}))\dot{\theta}\sin\alpha\\
&-0.5\frac{c\dot{\theta}d^2}{r^2}\Big)\Big(I_{\hat{z}}+3K+0.5Md^2+0.5\frac{Jd^2}{r^2}\\
&+(I_{\hat{x}}+mL^2-I_{\hat{z}})\sin^2\alpha\Big)^{-1}. \qquad (52)
\end{aligned}$$

Next it is important that the derived equations of motion are linearized to properly utilize classical design techniques such as PID control. Since the goal is to stabilize the robot around the operating point of $\alpha = 0$, the tilt of the robot can be assumed to be small or within just a few degrees. Therefore, the equations of motion can be linearized by using small-angle approximation ($\alpha \approx 0$), where we assume the following approximations in (53) and (54) [22].

$$\sin\alpha\approx\alpha, \qquad (53)$$

$$\cos\alpha\approx 1. \qquad (54)$$

By substituting this approximation into (50)-(52) and solving these equations for $\ddot{x}$, $\ddot{\alpha}$, and $\ddot{\theta}$ in terms of first order time derivatives or less, the following equations motion are obtained in (55)-(57).

$$\begin{aligned}
\ddot{x}=&\Big(\frac{u}{r}+mL(\dot{\theta}^2+\dot{\alpha})\alpha-mL\ddot{\alpha}-\frac{2c}{r}\Big(\frac{\dot{x}}{r}-\dot{\alpha}\Big)\Big)\\
&\times\Big[2M+m+\frac{2J}{r^2}\Big]^{-1}, \qquad (55)
\end{aligned}$$

$$\begin{aligned}
\ddot{\alpha}=&(-u+2c(\frac{\dot{x}}{r}-\dot{\alpha})+mgL\alpha\\
&-\ddot{x}mL\dot{\theta}^2\alpha(I_{\hat{z}}-I_{\hat{x}}-mL^2))(I_{\hat{y}}+mL^2)^{-1}, \qquad (56)
\end{aligned}$$

$$\begin{aligned}
\ddot{\theta}=&\Big(\frac{\phi}{2r}-(mL\dot{x}+2\dot{\alpha}(I_{\hat{x}}+mL^2-I_{\hat{z}}))\dot{\theta}\alpha\\
&-0.5\frac{c\dot{\theta}d^2}{r^2}\Big)\Big(I_{\hat{z}}+3K+0.5Md^2+0.5\frac{Jd^2}{r^2}\\
&+(I_{\hat{x}}+mL^2-I_{\hat{z}})\alpha^2\Big)^{-1}. \qquad (57)
\end{aligned}$$

Now, we have our linearized equations of motion in terms of three generalized coordinates.

5. PATH PLANNING

Artificial potential field (APF) path planning is a method of global navigation, where the configuration space is described as an artificial potential field that draws the robot toward the potential minimum. APF's consist of two components: attractive fields and repulsive fields. The attractive field can be thought of as a "well' where he global minimum is the navigation goal. The repulsive fields are attributed to obstacles within the configuration space.

The attractive field is stplit into two types: conical and paraboloidal. These potential fields are scalar fields whose gradient we interpret as the velocity vector field. For conical vector fields, the is constant for any given point in the field, which is interpreted as constant velocity. For paraboloidal vector fields, the magnitude converges to zero closer to the goal. However, the magnitude increases infinitely further from the goal. The desired velocity behavior of many mobile robot applications is to have an upper bound and converge to zero at the goal. This is known as a "soft-landing". In order to accomplish this, these two fields are combined in a piecewise function, while maintaining continuity at a threshold distance from the goal. Fig. 4 shows the mapping of conical and paraboloidal artificial potentials fields obtained using MATLAB.

The attractive potential field is defined using the following piece-wise function in (58) [13,23].

$$U_a(x,y)=\begin{cases}0.5k_a\|e(x,y)\|^2, & \|e(x,y)\|\le\rho,\\ k_b\|e(x,y)\|, & \|e(x,y)\|>\rho,\end{cases} \qquad (58)$$

where ρ is the threshold distance from the target, k_a and k_b are the gain values, and $e(x,y)$ is the distance error. The threshold distance is the error magnitude where the conical potential field transitions to paraboloidal potential field. It is chosen to allow enough distance for the controller to

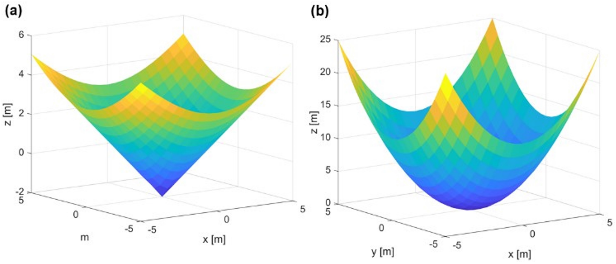

Fig. 4. Conical and paraboloidal artificial potential field.

perform a “soft-landing” without overshooting the target. The distance error given by (59)

$$e(x,y)=\begin{bmatrix}x_{goal}-x_c\\ y_{goal}-y_c\end{bmatrix}. \tag{59}$$

To maintain continuity at ρ, the condition in (60) must be satisfied [13,23]

$$\lim_{\|e(x,y)\|\to\rho^-}U_a(x,y)=\lim_{\|e(x,y)\|\to\rho^+}U_a(x,y), \tag{60}$$

resulting in the following conditions that must be maintained

$$k_b=\rho k_a. \tag{61}$$

Here, we take the negative gradient of the attractive potential field, to result in the vector field shown in (62).

$$\begin{aligned}F_a(x,y)&=-\nabla U_a(x,y)\\ &=\begin{cases}k_a e(x,y), & \|e(x,y)\|\le\rho,\\ k_b\frac{e(x,y)}{\|e(x,y)\|}, & \|e(x,y)\|>\rho.\end{cases}\end{aligned} \tag{62}$$

The attractive vector field is described as the chassis velocity vector with respect to the global coordinate frame, as shown in (63). It is much easier to directly control velocity rather than to directly control position, using a PID controller. This is because if there is a large distance error, the controller will use a very large input to drive the error to zero.

$$F_a(q)=\begin{bmatrix}v_x\\ v_y\end{bmatrix}. \tag{63}$$

With respect to the chassis frame, the chassis set-point velocity is given as shown in (64).

$${}^{c}v_x=\|F_a(x,y)\|_2. \tag{64}$$

Fig. 5 shows the combination of conical and paraboloidal artificial potential fields, with continuity maintained at $e(x,y)=\rho$. The set point yaw angle is given by (65).

$$\theta=\operatorname{atan2}(v_y,v_x). \tag{65}$$

However, the function $\operatorname{atan2}(y,x)$ has a discontinuity occurring at the interface of the 2 and 3 quadrants, or $\theta=\pm\pi$.

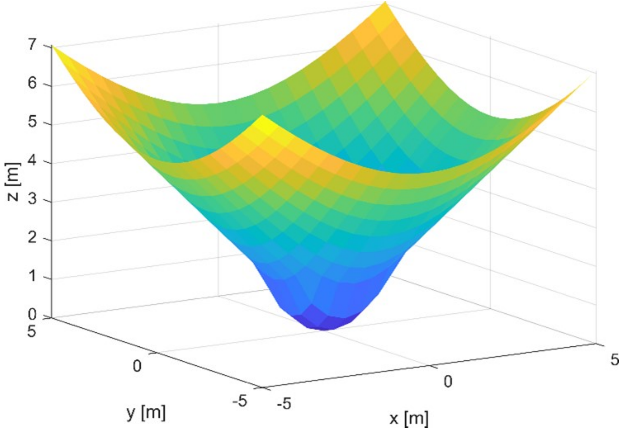

Fig. 5. Combined potential field.

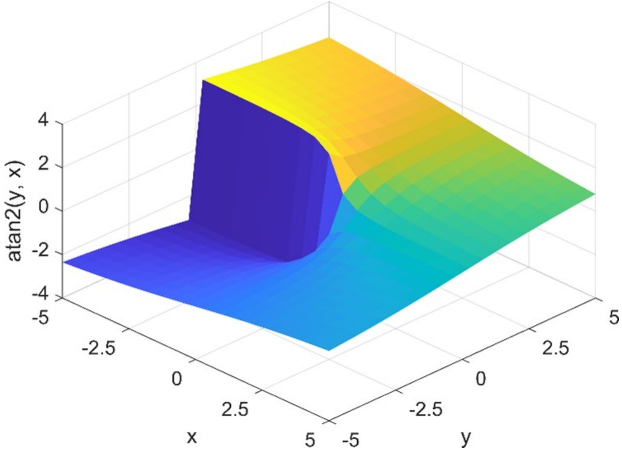

Fig. 6. ATAN2(Y,X) surface mapping.

If the robot is pursuing a moving target that requires it to turn more than $\pm\pi$ radians, we must implement a conditional statement that “winds up” the desired yaw angle to more than $\pm\pi$ radians.

Fig. 6 shows the mapping of ATAN2(Y,X) and its discontinuity at $(X,Y)=(-|x|,0)$. The windup condition relies on the direction of the angular velocity to determine whether we add or subtract π from our θ_{des}. For the condition, we use a step delay to find the direction of the desired angular velocity, as shown in (66).

$$\theta_{des}(\Delta t)=\theta_{des}(t_{n+1})-\theta_{des}(t_n). \tag{66}$$

t_n is the time value at step n. At the discontinuity, $\theta_{des}=N\pi$, where $N\in\mathbb{Z}$, the condition in (67) will be invoked.

$$\theta_{des}=\begin{cases}\theta_{des}+2\pi, & \theta_{des}(\Delta t)\ge 0,\\ \theta_{des}-2\pi, & \theta_{des}(\Delta t)<0.\end{cases} \tag{67}$$

This method of using APF to determine velocity set-points has drawbacks. For one thing, if the robot is pursuing a moving target with the goal of a “soft-landing” the robot

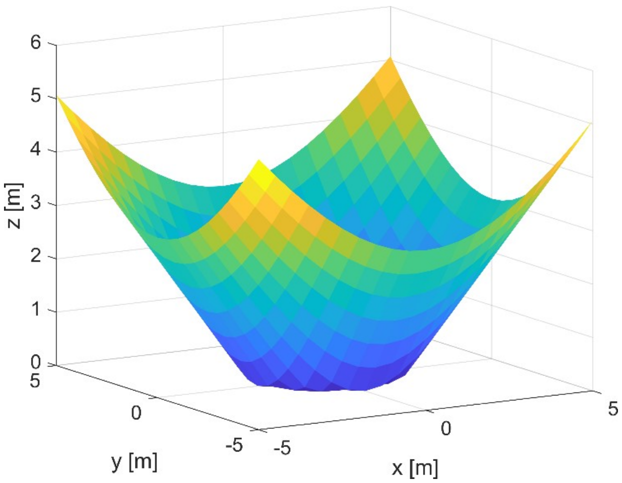

Fig. 7. Truncated cone APF.

will never reach the target and, instead, chase the target until the end of the simulation. To prevent this, instead of including the paraboloidal potential field, we consider the potential field to be a like a truncated cone, where the truncated section is equal to zero. This will maintain a constant velocity set point. If the target is slower than the velocity set point, the robot can reach the target. The truncated cone potential field is visualized by Fig. 7.

6. PID CONTROL

The most common controller in linear systems is the PID controller, which stands for proportional, integral, derivative. It utilized feedback error to determine a value for the control input, as shown in (68) [22].

$$u(t) = K_P e(t) + K_I \int_0^t e(\tau) d\tau + K_D \frac{de(t)}{dt}, \tag{68}$$

where $e(t)$ is the error, and K_P, K_I, K_D are the gains for the proportional, integral, and derivative terms, respectively, and $u(t)$ is the control signal. The error is the difference of the set point and the measure process variable. Fig. 8 shows the negative feedback control loop using a PID controller. Using this, we can create a block diagram representing the control of our 3 control variables. Fig. 9 shows the control architecture in block diagram form for the three control variables, α, $\dot{x}$, and $\dot{\theta}$, where the set points for $\dot{x}$ and $\dot{\theta}$ are determined by the artificial potential fields.

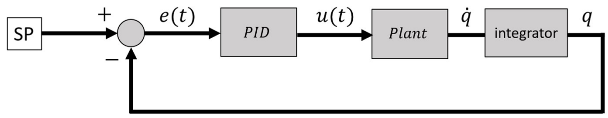

Fig. 8. PID feedback control.

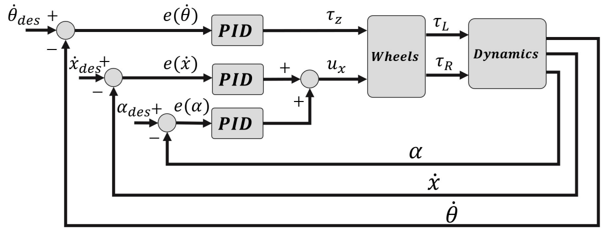

Fig. 9. System control loop.

7. SIMULATIONS

The robot will be subject to many types of simulations to evaluate its ability to perform tasks and reject disturbances. These tests are ordered from least to most difficult: force disturbance, torque disturbance, stationary target navigation, moving target navigation, and moving target navigation with disturbances. The fixed parameters of the TWDDR, such as mass properties, gains, and dimensions are based on a prototype CAD model, and they are given in Table 1.

7.1. Force disturbance

First, we begin the simplest task of maintaining stability while experiencing an impulse disturbance. Forward and

Table 1. Parameters.

Parameter	Symbol	Unit	Value
Mass of body	m	kg	2.575
Mass of wheel	M	kg	0.2121
Gravitational acceleration	g	m/s^2	9.81
Length	L	m	0.3895
Body mass moment of inertia	$(I_{\hat{x}}, I_{\hat{y}}, I_{\hat{z}})$	kg·m^2	(0.1278, 0.1248, 4.641$e-3$)
Wheel mass moment of inertia	(K, J, K)	kg·m^2	(5.229$e-3$, 2.651$e-4$, 5.229$e-3$)
Friction coefficient	c	N·s/rad	1
Wheel distance	d	m	0.15
Pitch controller gains	$(K_p, K_I, K_D)_\alpha$	-	(250, 2000, 31.7)
Velocity controller gains	$(K_p, K_I, K_D)_{\dot{x}}$	-	(10, 250, 25)
Yaw rate controller gains	$(K_p, K_I, K_D)_{\dot{\theta}}$	-	(5, 1, 1)
Distance threshold	ρ_x	m	0.5
Velocity setpoint	$(\dot{x}_{near}, \dot{x}_{far})$	m/s	(0, 12)
Angular distance threshold	ρ_θ	rad	$\pi/50$
Angular velocity setpoint	$(k_a, k_b)_{\dot{\theta}}$	rad/s	(50/3, $\pi/3$)

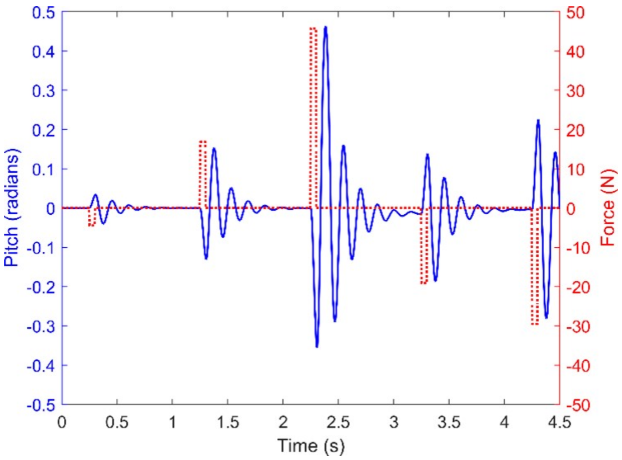

Fig. 10. Pitch response to impulse disturbances.

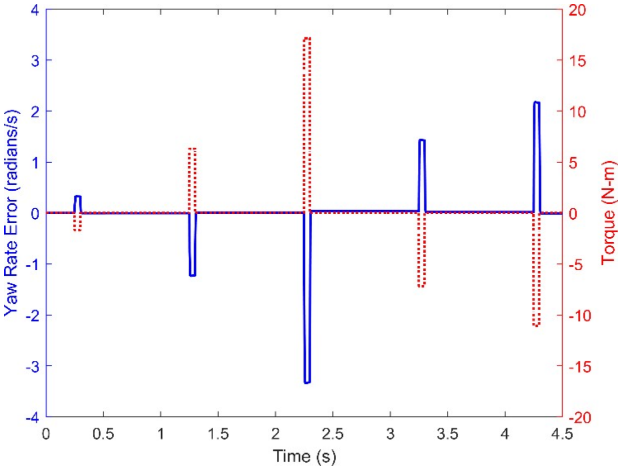

Fig. 11. Yaw response to torque input.

yaw velocity are both uncontrolled. The disturbance is set to be an impulse that has a period of 1 second, with its pulse width being 5% of the period. The magnitude of each impulse is set a random value on the interval $[-50,\,50]$ N.

Fig. 10 plots the pitch response and the input force disturbance of the robot. The robot maintains stability, though, with significant oscillation. The pitch angle never exceeds ± 0.47 rad, or ± 26.9 deg, and responds with a maximum settling time is 0.85 seconds.

7.2. Torque disturbance

In this simulation, a torque impulse disturbance is applied to the robot about the chassis' z-axis with random inputs $\tau \in [-20,\,20]$ Nm. Fig. 11 illustrates the robot's yaw rate error response to random torque disturbances during a 4.5 second simulation. The simulation results showcase the system's resilience and capability to counteract torque disturbances, with disturbance magnitudes reaching a value of 17.5 Nm. The yaw rate error exhibits a rapid and immediate response, driving towards $\theta = 0$ rad/s.

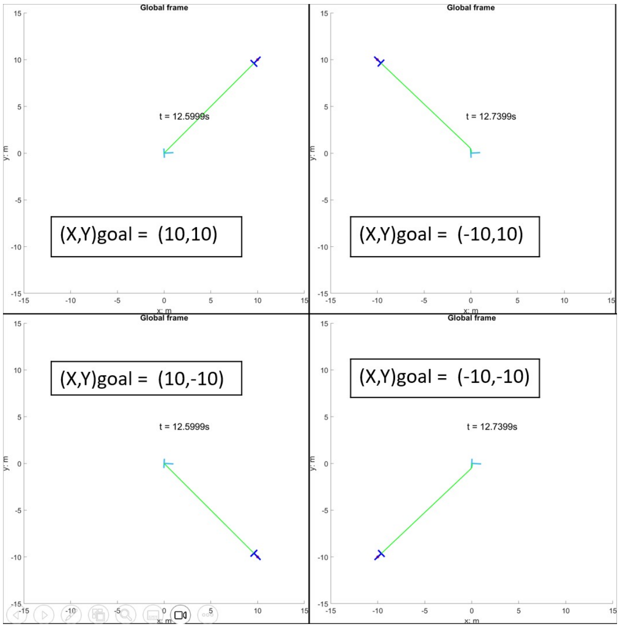

Fig. 12. Stationary target navigation: (a) Target located at $[10,\,10]$, (b) target located at $[-10,\,10]$, (c) target located at $[10,\,-10]$, (d) target located at $[-10,\,-10]$.

7.3. Stationary target

Now, the robot has the task of navigating to a stationary target. This will evaluate the robot's yaw velocity and linear velocity PID controllers while maintaining upright stability.

Fig. 12 shows four different stationary target navigation scenarios for the robot. The dark blue "T" shape represents the robot's final configuration, the light blue "T" shape represents the robot's initial configuration, the green line represents the path taken, and the red circle represents the target. As we can see here, the robot is able to navigate to its specified target.

7.4. Moving target

Here, the target moves clockwise and counterclockwise around $(x,y) = (0,0)$ at a velocity larger than the robot's maximum set point velocity. This assesses the robot's ability to chase a target, as well as operate through its angular discontinuity, and it tests the conditional statement that we implemented. It should be noted that, since the target is moving faster than the robot's maximum set point velocity, the robot will not ever reach the target. Therefore, the simulation is capped at $t = 41$ s.

Figs. 13 and 14 show the navigation toward a moving target. Fig. 13 shows the target moving counterclockwise, and Fig. 14 shows the target moving clockwise. The pink line represents the target's path. The robot successfully navigates toward the target while maintaining its stability. We can see that the windup conditional is successful moving at a positive or negative angular velocity through the discontinuity.

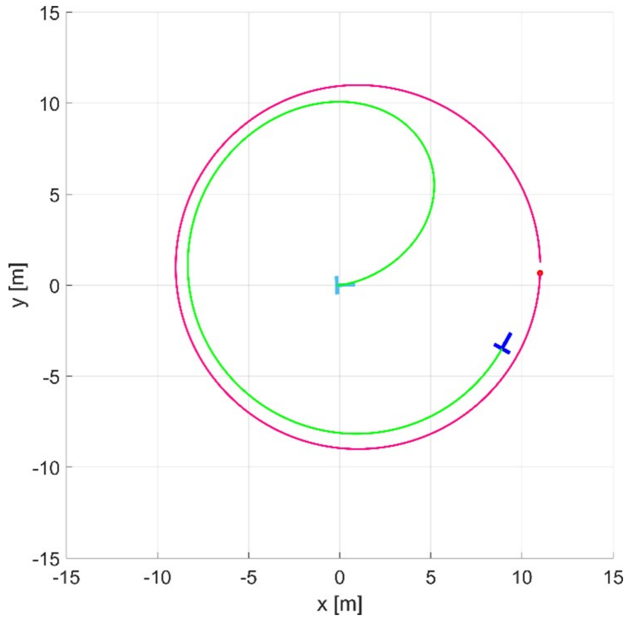

Fig. 13. Counterclockwise target pursuit.

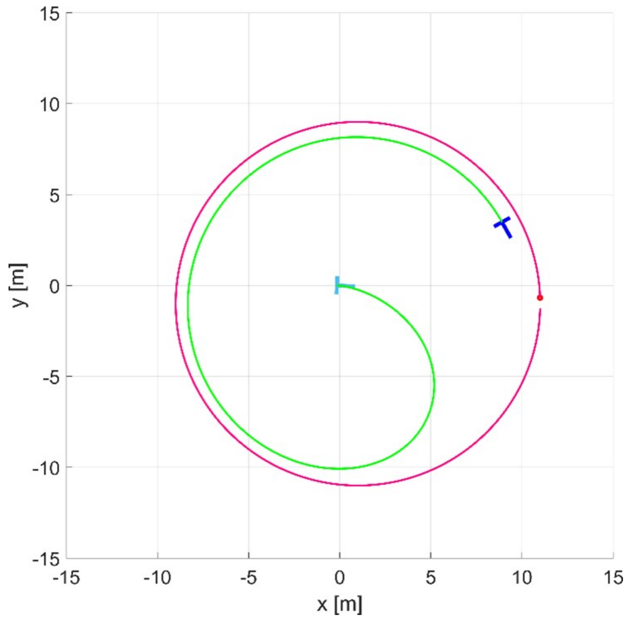

Fig. 14. Clockwise target pursuit.

8. CONCLUSION

We have successfully designed a TWDDR and displayed that it can autonomously navigate stationary and moving targets while balancing and rejecting disturbances. The dynamic model was found using the Euler-Lagrange approach and it included non-holonomic velocity constraints. The equations of motion were then linearized at $\alpha = 0$ to increase the performance of the controllers. Three PID controllers were implemented to control the pitch, yaw angular velocity, and forward x velocity. The yaw angular velocity and forward x velocity set points were are autonomously determined using the method of artificial potential fields. The robot model was subject to multiple simulations to test different aspects of the control architecture. Disturbance simulations yielded desirable results, with controllers able to stabilize α with a settling time of 0.85 s and θ with rapid, immediate response. Simulations testing the autonomous navigation verified the robot's ability to maneuver to both stationary and moving targets.

9. FUTURE WORK

Further work will be done to design a controller capable of stabilizing the yaw velocity, designing implementation of hardware such as DC motors, motor drives, IMU's, etc., include obstacle maneuvering, sensor filtering, computer vision for obstacle detection, accurate localization, etc.

CONFLICT OF INTEREST

The authors declare that there is no competing financial interest or personal relationship that could have appeared to influence the work reported in this paper.

REFERENCES

[1] J. Elke, J. Pei, R. J. Caverly, and D. G. Egziabher, "A low-cost and low-risk testbed for control design of launch vehicles and landing systems," *Proc. of IEEE Aerospace Conference*, Big Sky, MT, USA, pp. 1-12, 2021.

[2] H. Nguyen, J. Morrell, K. Mullens, A. Burmeister, S. Miles, N. Farrington, K. Thomas, and D. Gage, "Segway robotic mobility platform," *Proc. of SPIE - The International Society for Optical Engineering*, Philadelphia, PA, USA, October 27-28, 2004.

[3] "Handle," (in en), ROBOTS: Your Guide to the World of Robotics.

[4] I. Siradjuddin, M. Syafaat, T. S. Patma, S. Adhisuwignjo, T. Winarno, A. Komarudin, and D. Widiatmoko, "PID controller for a differential drive robot balancing system," *Journal of Physics: Conference Series*, 1402, 044021, December 2019.

[5] M. Zohaib, S. M. Pasha, H. Bushra, K. Hassan, and J. Iqbal, "Addressing collision avoidance and nonholonomic constraints of a wheeled robot: Modeling and simulation," *Proc. of International Conference on Robotics and Emerging Allied Technologies in Engineering (iCREATE)*, Islamabad, Pakistan, pp. 306-311, 2014.

[6] S. Kim and S. Kwon, "Dynamic modeling of a two-wheeled inverted pendulum balancing mobile robot." *International Journal of Control, Automation, and Systems*, vol. 13, pp. 926-933, 2015.

[7] R. Dhaouadi and A. A. Hatab, "Dynamic modelling of differential-drive mobile robots using Lagrange and Newton-Euler methodologies: A unified framework," *Advances in Robotics & Automation*, vol. 2, no. 2, 2013.

[8] A. O. Owojori, K. F. Akingbade, W. O. Apena, and E. O. Ogunti, “Stability control modelling under dynamic motion scenario of a differential drive robot,” *Journal of Engineering Studies and Research*, vol. 27, no. 3, pp. 64-73, 2021.

[9] A. Mathew, R. Ananthu, P. Binsy, A. Vahid, C. Thomas, and S. Sidharthan, “Design and control of a two-wheel self-balancing robot,” *Proc. of IP Conference Series: Materials Science and Engineering*, December 2021.

[10] W. Lin, H. Zhong, and F. H. Li, “Design and implementation of control system for two-wheeled self-balancing robot,” *Journal of Electronic Measurement and Instrument*, vol. 27, no. 8, pp. 750-759, 2013.

[11] B. Y. Suprapto, D. Amri, and S. Dwijayanti, “Comparison of control methods PD, PI, and PID on two wheeled self balancing robot,” *Proc. of International Conference on Electrical Engineering, Computer Science and Informatics*, Yogyakarta, Indonesia, August 2014.

[12] W. Y. Loong, L. Z. Long, and L. C. Hun, “A star path following mobile robot,” *Proc. of 4th International Conference on Mechatronics (ICOM)*, Kuala Lumpur, Malaysia, pp. 1-7, 2011.

[13] Q. Zhu, Y. Yan, and Z. Xing, “Robot path planning based on artificial potential field approach with simulated annealing,” *Proc. of 6th International Conference on Intelligent Systems Design and Applications*, Jian, China, pp. 622-627, 2006.

[14] H. Wang, Y. Yu, and Q. Yuan, “Application of Dijkstra algorithm in robot path-planning,” *Proc. of 2nd International Conference on Mechanic Automation and Control Engineering*, Hohhot, pp. 1067-1069, 2011.

[15] I. Noreen, A. Khan, and Z. Habib, “Optimal path planning using RRT* based approaches: A survey and future directions” *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 7, no. 11, 2016.

[16] G. Sanchez and J.-C. Latombe, “Using a PRM planner to compare centralized and decoupled planning for multi-robot systems,” *Proc. of IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, Washington, DC, USA, vol. 2, pp. 2112-2119, 2002.

[17] M. R. Flannery, “d’Alembert-Lagrange analytical dynamics for nonholonomic systems,” *Journal of Mathematical Physics*, vol. 52, no. 3, 032705, 2011.

[18] Y. Kim, S. H. Kim, and Y. K. Kwak, “Dynamic analysis of a nonholonomic two-wheeled inverted pendulum robot,” *Journal of Intellegent and Robotic Systems*, vol. 44, pp. 25-46, 2005.

[19] Y. Tian, N. Sidek, and N. Sarkar, “Modeling and control of a nonholonomic wheeled mobile robot with wheel slip dynamics,” *Proc. of IEEE Symposium on Computational Intelligence in Control and Automation*, Nashville, TN, USA, pp. 7-14, 2009.

[20] K. M. Lynch and F. C. Park, *Modern Robotics: Mechanics, Planning, and Control*, The Cambridge University Press, 2017.

[21] H. Cen and B. K. Singh, “Nonholonomic wheeled mobile robot trajectory tracking control based on improved sliding mode variable structur,” *Wireless Communications and Mobile Computing*, vol. 2021, 2974839, 2021.

[22] R. C. Dorf and R. H. Bishop, *Modern Control Systems*, Pearson, 2011.

[23] H. Choset, K. M. Lynch, S. Hutchinson, G. A. Kantor, and W. Burgard, *Principles of Robot Motion: Theory, Algorithms, and Implementations*, MIT press, 2005.

John Moritz received his B.S. and M.S. degrees in mechanical engineering from University of Arkansas, in 2020 and 2023, respectively. His research interests include dynamic modeling, non-linear control, and deep reinforcement learning or robotics applications.

Mishek Musa received his B.S. and M.S. degrees in mechanical engineering from the University of Arkansas, Fayetteville, AR, USA, in 2019 and 2021, repsectively. He is currently pursuing a Ph.D. degree in mechanical engineering from the University of Arkansas, Fayetteville, AR, USA. His research interests include robotics, mechatronic systems, and control system design.

Uche Wejinya received his B.S. and M.S. degrees in electrical and computer engineering from Michigan State University, East Lansing, MI, USA, in 2000 and 2002, respectively. He then received a Ph.D. degree in electrical engineering in August 2007 from Michigan State University. In February 2008, he joined the Department of Mechanical Engineering at the University of Arkansas - Fayetteville (UAF) where he held a tenure-track assistant professor position until June of 2014. Since July 2014, he has been a tenured associate professor in the Department of Mechanical Engineering at the University of Arkansas - Fayetteville (UAF). His research interests include mechatronics with emphasis on nanoelectronics; control systems design and application, robotics, biomechanics, batteries & energy storage devices, modeling and simulation of robotics systems.

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