ISSN: 2832-790X
DOI: 10.33552/OJRAT.2024.03.000554
Online Journal of Robotics & Automation Technology
Iris Publishers

Research Article

An Investigative Study of the Navigation of a Self-Balancing Robot in A Dynamic Environment

Mishek Musa* and Uche Wejinya

Department of Mechanical Engineering, University of Arkansas, Fayetteville, AR, 72701, USA

***Corresponding author:** Mishek Musa, Department of Mechanical Engineering, University of Arkansas, Fayetteville, AR, 72701, USA.
Email: mjmusa@uark.edu. **Phone:** +1-479-575-4800

Received Date: June 12, 2024
Published Date: June 26, 2024

Abstract

Stability control of two-wheeled balancing robotic systems has been extensively studied in literature; however, few studies extend the work to a high-level navigational control, particularly in a dynamic environment. In this work, a simple cascaded PID controller is designed and implemented to control the low-level linear velocity, angular velocity, and pitch angle of the robot. Furthermore, the artificial potential field approach is employed in this work to provide high-level navigational control of the robot in both static and dynamic environments via simulation. The results of the studies provide validation of the approach and show that the robot is able to successfully track both static and randomly moving targets while avoiding collisions with both static and dynamic obstacles. This is all implemented while maintaining stability of the robot and closely tracking reference velocity commands.

Keywords: Self-Balancing Robot; Artificial Potential Fields; Robot Simulation

Introduction

The field of robotics has made great strides in recent years, becoming an integral part of everyday life. Robots are able to repeatedly and accurately perform tasks that are arduous, time-consuming, or too dangerous for humans to undertake. This is particularly true for mobile robots which similarly to humans are capable of locomotion within their environment typically through the use of legs, wheels, or tracks [1]. Most popularly, wheeled mobile robots have been applied to a variety of applications including material handling, healthcare, search and rescue, and many others. Wheeled mobile robots can be designed in a variety of configurations primarily dependent on the number of wheels they have [1]. In this work, the focus will be on a wheeled mobile robot that has only two wheels and so is statically unstable. Two-wheeled balancing robots (TWBR) have several advantages over other classes of mobile robots that implement more wheels to guarantee static stability.

First, having only two wheels allows for a high degree of maneuverability as the robot has a turning radius of zero which enables it to turn in place. This is beneficial for operation in confined spaces such as narrow corridors or tunnels. Second, since the robot needs to be actively stabilized, it is able to maintain stability even in the presence of external disturbances, whereas robots with more wheels may easily tip over and become unstable in the presence of external disturbances. This feature also allows two-wheeled robots to be taller or have a higher center of mass (gravity) than other mobile robots. While the design of controllers for this class of mobile robot is more complicated, the benefits they provide can easily

outweigh the cost. The basic structure of a TWBR often comprises of two wheels mounted co-axially on either side of the main robot body, where the center of mass of the robot is above the wheel axles [2]. The inherent instability of this configuration leads to a control problem that resembles the classic inverted pendulum controls problem. Several solutions have been proposed to stabilize the robot. One popular approach that has been investigated due to its ease of implementation is the Proportional- Integral-Derivative (PID) controller [3- 6]. PID controllers only require the tuning of three gains and do not require a model of the plant to work, however this simplicity comes at the expense of low robustness. More complicated controllers have been presented to increase the robustness and adaptability of the system including fuzzy logic control [7, 8], linear quadratic regulator (LQR) control [9, 10], sliding mode control [11, 12] and model predictive control (MPC) [13].

While many researchers have investigated and continue to investigate control strategies for TWBR, few studies provide further evaluation of the system under high level navigational control. This is especially true for dynamic environments where both the targets and obstacles are moving. Several planning and navigation strategies have been proposed for mobile robots in general including A∗ [14], Probabilistic Road Maps (PRM) [15], Artificial Potential Fields (APF) [16], Genetic Algorithms (GA) [17], etc. In this work, the focus will be on the navigational strategy for a TWBR in a dynamic environment using the artificial potential field (APF) approach. The APF approach was first proposed by Khatib in 1986 [16] and makes use of a computationally efficient formulation for robot navigation. In general, the APF approach works by introducing an at- tractive potential field for bringing the robot towards a desired location/ target, while a repulsive potential field is used for obstacle avoidance. In a dynamic environment, where both the goal and obstacles are moving, the changing environment can affect the performance of APF. To overcome this, Ge et al. proposed the potential fields in terms of both the relative positions and velocities between the robot and the goal and obstacles [18].

Montiel et al. presented a parallel evolutionary APF ford dynamic environments enabling the possibility to control the robot in complex real-world scenarios if a reachable configuration set exists [19]. Sun et al. introduced a modified APF approach that employs a dynamic window approach to avoid local minima and uses a danger index to avoid moving obstacles [20]. These methods have shown promising results for general mobile robot planning, however, few if any have been studied using the full dynamics of a TWBR. In this work, the APF approach is investigated for the high-level navigation of a two-wheeled balancing robot in a dynamic environment. Not only is it important for the robot to maintain its stability throughout the duration of its motion, but it must also be able to successfully navigate through the environment and reach the target in a collision-free manner. The work is organized as follows: The section, System Modeling describes the kinematics and dynamics of the TWBR. Section Control System Design outlines the low-level controller for maintaining balance and linear and angular velocity control along with the high level APF controller. Section, Simulation Studies presents the results of the simulated TWBR under both low- and high-level control in static and dynamic environments. Lastly, the work is concluded in the Conclusions and Discussions section.

System Modeling

Kinematics

A rendering of the proposed TWBR is shown in Figure 1 that describes the variables used in the kinematic and dynamic formulations. From the kinematic perspective, the TWBR can be considered as a differential drive robot. This class of mobile robots can control their orientation simply by varying the relative velocities of the wheels and does not require additional steering motion [21]. In this work, it is assumed that no lateral slip of occurs between the wheels and the ground surface, and that the robot is undergoing pure rolling, hence no slipping of the wheel along the longitudinal axis and no orthogonal skidding. Using these assumptions, the following non-holonomic kinematic constraints can be applied to the robot.

$$-\dot{x}_c \sin\psi + \dot{y}_c \cos\psi = 0 \tag{1}$$

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

$$\dot{x}_c \cos+ \dot{y}_c\psi - \left(\frac{d}{2}\right)\dot{\psi} - r\dot{\gamma}_L = 0 \tag{3}$$

where $\dot{x}_c$ and $\dot{y}_c$ are the linear velocities of the robot in the inertial frame, ψ is the yaw rotation angle of the robot, and $\dot{\gamma}_L$ and $\dot{\gamma}_R$ are the angular velocities of the left and right wheels respectively. The other parameters for the TWBR are shown in Table 1. The forward velocity $\dot{x}$ and the angular velocity ω of the robot can be written in terms of the angular velocities of the robot wheels as follows:

Table 1: Parameters used in the TWBR system modelling.

Symbol	Unit	Description
d	m	Distance between the wheels
r	m	Radius of the wheels
l	m	Length of the pendulum
m_b	kg	Mass of the robot body
m_w	kg	Mass of a wheel
J	kg-m^2	Mass moment of inertia of each wheel w.r.t. the wheel axis
k	kg-m^2	Mass moment of inertia of each wheel w.r.t the vertical axis
I_x, I_y, I_z	kg-m^2	Mass moment of inertia of the robot body
T_R	N-m	Torque applied to the right wheel
T_L	N-m	torque applied to the left wheel
$\dot{x}$	m-s^{-1}	Forward velocity of the robot
$\dot{\psi}$	rad-s^{-1}	Yaw Rate of the robot
θ	rad	Pitch angle of the robot

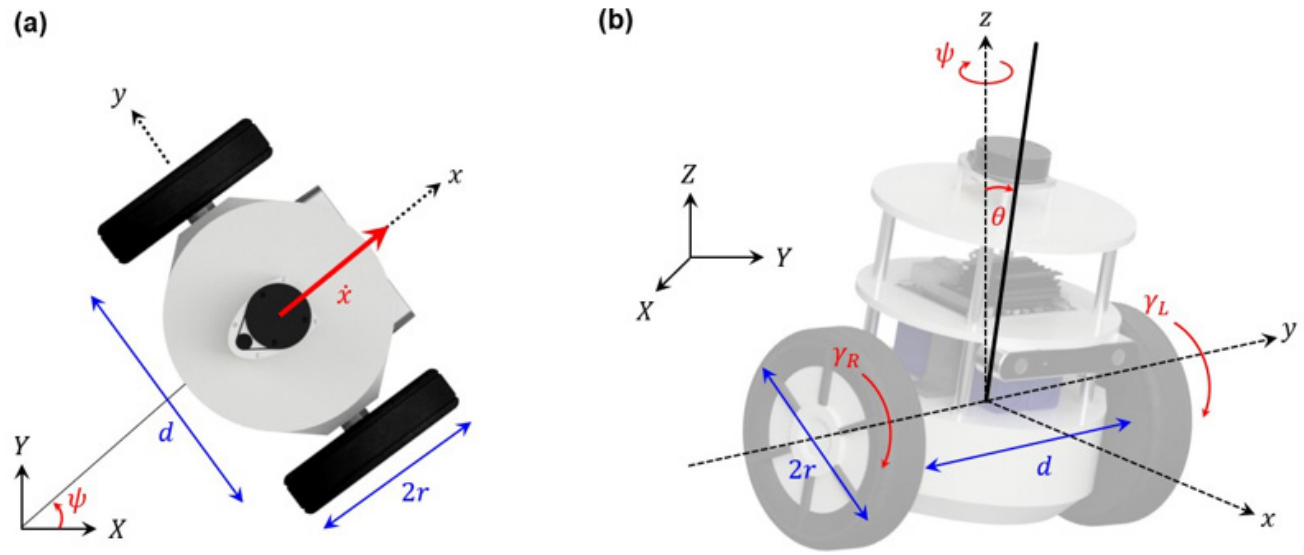

Figure 1: (a) Top view of the TWBR used to formulate the robot kinematics, (b) Parameters used for the robot dynamics formulation.

$$\dot{x} = \frac{r\left(\dot{\gamma}_r + \dot{\gamma}_L\right)}{2} \quad (4)$$

$$\omega = \frac{r\left(\dot{\gamma}_r - \dot{\gamma}_L\right)}{2} \quad (5)$$

Rewriting in matrix form, the relationship between the robot velocities and the wheel angular velocities is given by

$$\begin{bmatrix} \dot{x} \\ \omega \end{bmatrix} = \begin{bmatrix} \frac{r}{2} & \frac{r}{2} \\ \frac{r}{d} & -\frac{r}{d} \end{bmatrix} \begin{bmatrix} \dot{\gamma}_R \\ \dot{\gamma}_L \end{bmatrix} \quad (6)$$

The kinematics of the robot can also be represented by the ve-

locities of the robot in the inertial frame in terms of the robot velocities in the global frame as follows:

$$\begin{bmatrix} \dot{x}_c \\ \dot{y}_c \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \omega \end{bmatrix} \quad (7)$$

Dynamics

The dynamics of the TWBR are formulated using the Lagrangian approach described in Kim et al. [22]. For the dynamic formulation, the generalized coordinates can be defined as

$$q = \begin{bmatrix} x_c & y_c & \psi & \theta & \gamma_L & \gamma_R \end{bmatrix}^T \quad (8)$$

The kinetic energy of the system can be written as the sum of the translational and rotational energies as $T = T_{kinetic} + T_{rot}$ where $T_{kinetic}$ and T_{rot} are given as follows:

$$T_{kinetic} = \frac{1}{2} m_w v_R^2 + \frac{1}{2} m_w v_L^2 + \frac{1}{2} m_b v_b^2 \quad (9)$$

$$T_{rot} = \frac{1}{2}\left(\left(I_x - I_z \right) \sin^2\theta + I_z + 2K \right) \psi^2 + I_y \dot{\theta}^2 + \frac{1}{2} J \dot{\gamma}_L^{\,2} + \frac{1}{2} J \dot{\gamma}_R \quad (10)$$

where v_R, v_L, and v_b are the linear velocities of the right wheel, left wheel, and robot body respectively. Since it is assumed that the motion of the TWBR is planar, the potential energy of the system is related to the pitch of the robot body. This is given as

$$V = m_b g l \cos\theta \quad (11)$$

The Lagrangian dynamic equation of motion is then given as

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}} \right) - \frac{\partial L}{\partial \dot{q}} \quad (12)$$

where L = T - V. Solving the Lagrangian subject to the kinematic constraints described previously, the dynamics equations can be written in terms of the generalized coordinates x, θ, and ψ as

$$\begin{aligned} &\left(m_B + 2m_W + \frac{2J}{r^2} \right) \ddot{x} - m_B l \left(\dot{\psi}^2 + \dot{\theta}^2 \right) s\theta \\ &+ \left(m_B l c\theta \right) \ddot{\theta} + \left(2/r \right) c_\alpha \left(\dot{x}/r - \dot{\theta} \right) = \left(T_L + T_R \right)/r \end{aligned} \quad (13)$$

$$\begin{aligned} &\left(I_2 + m_B l^2 \right) \ddot{\theta} + \left(m_B l c\theta \right) \ddot{x} + \left(I_3 - I_1 - m_B l^2 \right) \dot{\psi}^2 s\theta c\theta \\ &- m_B l_g s\theta - 2c_\alpha \left(x/r - \dot{\theta} \right) = -\left(T_L + T_R \right) \end{aligned} \quad (14)$$

$$\begin{aligned} &\left(I_3 + 2K + m_w \frac{d^2}{2} + J \frac{d^2}{2r^2} - \left(I_3 - I_1 - m_B l^2 \right) s^2\theta \right) \ddot{\psi} \\ &+ \left(m_B l \dot{x} - 2\left(I_3 - I_1 - m_B l^2 \right) \dot{\theta} c\theta \right) \dot{\psi} s\theta \\ &+ c_\alpha \dot{\psi} d^2 / \left(2r^2 \right) = \left(T_R + T_L \right) d/(2r) \end{aligned} \quad (15)$$

where sθ and cθ are shortened representations of the trigonometric functions sin θ and cos θ respectfully.

Motor Model

The wheels can be driven using a standard DC motor. The circuit model for a DC motor is shown in Figure2. The output torque, τ, is proportional to the armature current, i, as follows

$$\tau = K_t i \quad (16)$$

where K_t is a positive, non-zero torque constant. Additionally, the induced back electromotive force (EMF), *e*, is proportional to the angular velocity of the shaft given by

$$e = K_e \dot{\gamma} \quad (17)$$

where K_e is a positive, non-zero EMF constant. It can be assumed that $K_t = K_e$ and so will be represented by the single constant K. Neglecting the effects of friction and inertia, the governing equation of the DC motor can then be written as

$$L\frac{di}{dt} + Ri = V - K\dot{\gamma} \quad (18)$$

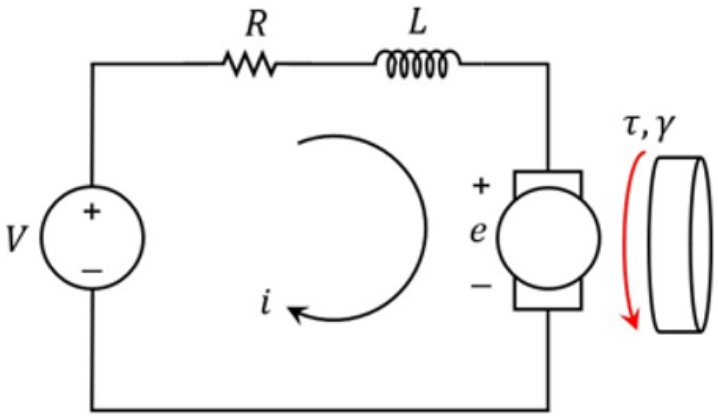

Figure 2: Schematic diagram of a brushed DC motor.

Control System Design

Cascaded PID Control

As previously stated, the inverted pendulum-like system is naturally unstable, and so requires active control of the robot states. From the dynamics equations, the three controllable states of the robot are the forward velocity $\dot{x}$, the pitch θ, and the yaw rate $\dot{\psi}$. Furthermore, from the dynamic's equations, the linear velocity and pitch processes are coupled, whereby one process drives the other. In order to control the forward velocity of the robot, the robot must be able to initially pitch forward to move forward, backward to move backwards or maintain a value close to zero to balance.

For these coupled processes, a cascaded control system was designed to maintain a desired linear velocity, while also driving the pitch of the robot to the balanced vertical state. Both controllers implemented are PID controllers. An additional PI controller was designed to control the angular velocity process of the robot. Actions taken by this controller could be done without the introduction of significant disturbance to the system. A schematic of the designed control system is shown in Figure 3 where $\dot{x}_{\text{ref}}$ and $\dot{\psi}_{re}$ f are the reference linear and angular velocities respectively.

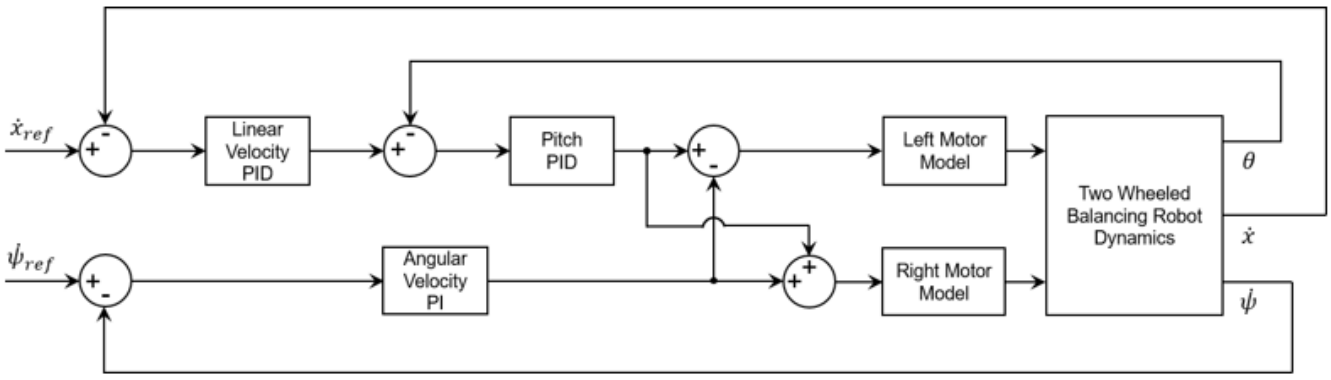

Figure 3: Schematic diagram of the cascaded control system for the TWBR. PID controllers are used for the linear velocity and pitch processes, while a PI controller is implemented for the angular velocity process.

Artificial Potential Fields

For motion planning of the mobile robot in a dynamic environment, this involves controlling the robot motion to track a randomly moving target while avoiding obstacles. In this work it is assumed that the positions and velocities of the robot and target are known. Additionally, it is assumed that the position of any obstacle can be accurately measured on-line. In order to drive the robot toward the target while avoiding obstacles, attractive and repulsive potential fields are utilized. In a dynamic environment, an attractive potential defined only in terms of the relative distance between the robot and target is not sufficient. To overcome this, the potential filed can be modified to include the relative velocity between the robot and target as proposed in Ge et al. [18] The attractive potential is then given as

$$U_{att} = \frac{1}{2}K_p\left(p_{tar} - p_{rob}\right)2 + \frac{1}{2}K_v\left(v_{tar} - v_{rob}\right)2 \quad (19)$$

where p_{rob} and p_{tar} are the positions of the robot and target respectively, v_{rob} and v_{tar} are the velocities of the robot and target respectively, and K_p and K_v are positive scalar parameters that affect the magnitude of the potential field. The force needed to drive the robot to the target can then be found by the negative gradient of the attractive potential in terms of both position and velocity.

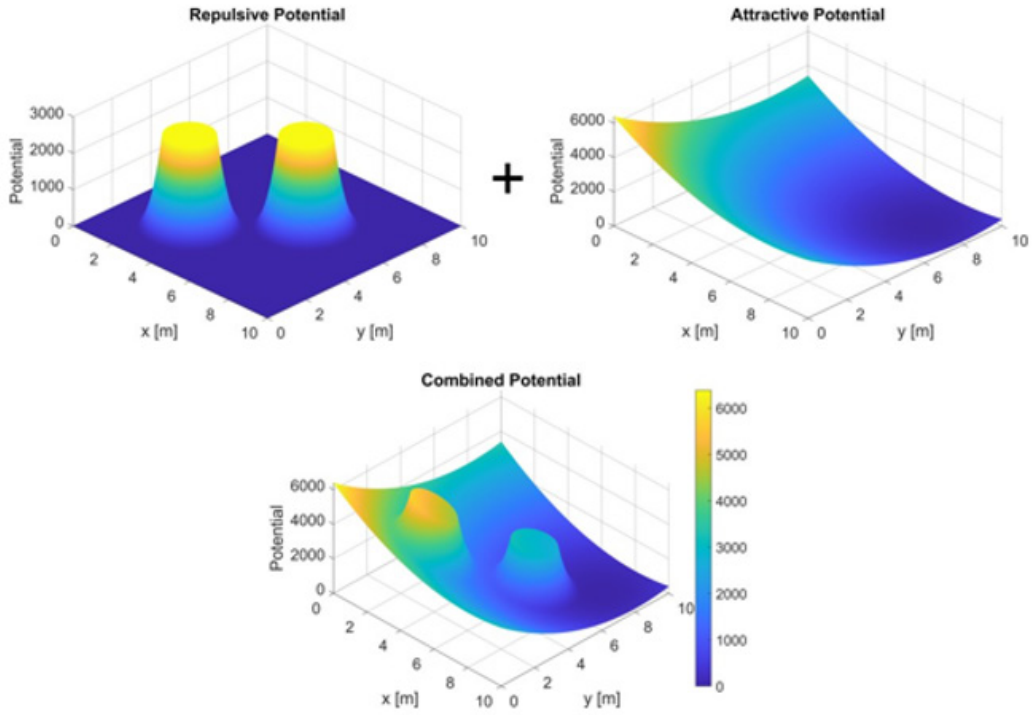

Figure 4: Repulsive, attractive, and combined potential fields for a simple environment. The combined potential field is found by taking the sum of the repulsive and attractive potential fields.

$$F_{att} = -\nabla_p U_{att} - \nabla_v U_{att} = -\frac{\partial U_{att}}{\partial p} - \frac{\partial U_{att}}{\partial v} \quad (20)$$

$$F_{att} = K_p\left(p_{tar} - p_{rob}\right) + K_v\left(v_{tar} - v_{rob}\right) \quad (21)$$

For avoiding obstacles, a repulsive field is generated that is dependent on the relative distance between the robot and the obstacles. The repulsive potential is given as follows

$$U_{rep} = \begin{cases} \frac{1}{2}K_o\left[\left(\frac{1}{d_i} - \frac{1}{\rho_o}\right)^2 - \left(d_i - \rho_o\right)^2\right] & if\ di \le \rho_o \\ 0 & if\ di > \rho_o \end{cases} \quad (22)$$

where d_i is the distance between the robot and the i^{th} obstacle, ρ_o is the range that the robot is capable of detecting obstacles, and K_o is a positive scalar parameter that affects the magnitude of the potential field. From the equation it can be seen that the repulsive potential field goes to zero outside of the obstacle detection range of the robot and so the robot only acts under the influence of the attractive potential. Similar to the attractive force, the repulsive force can be found by the negative gradient of the repulsive potential.

$$F_{rep} = -\nabla_p U_{att} = \begin{cases} K_o \left[\frac{1}{d^2_i} \left(\frac{1}{d_i} - \frac{1}{\rho_o} \right) - \left(d_i - \rho_o \right) \right] & if\ di \le \rho_o \\ 0 & if\ di > \rho_o \end{cases} \quad (23)$$

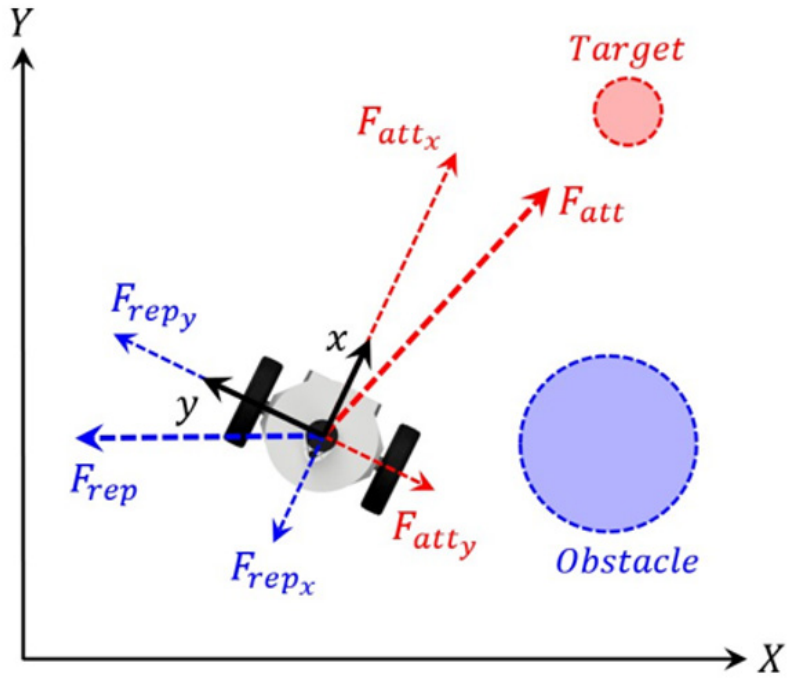

Figure 5: Attractive and repulsive forces applied to the TWBR given a target and one obstacle in the environment.

An example of the repulsive and attractive potential fields is shown in Figure 4 where the bright yellow color indicates regions of high potential, and the dark blue areas represent regions of low potential. Also depicted is the combined potential field which is found by simply taking the sum of the attractive and repulsive fields as follows

$$U = U_{att} + U_{rep} \quad (24)$$

The applied forces on the TWBR are shown in Figure 5. With the attractive and repulsive forces, the total force, F, acting on the robot, similarly to the combined potential field is found by the sum of the attractive and repulsive forces as follows:

$$F = F_{att} + F_{rep} \quad (25)$$

Using the x and y components of F, the desired heading angle of the robot can be computed as

$$\alpha = \tan^{-1} \frac{F_y}{F_x} \quad (26)$$

A proportional controller is then used to determine the desired yaw rate of the robot given by

$$\dot{\psi} = K_s \left(\alpha - \psi \right) \quad (27)$$

where Ks is a positive gain. The linear velocity of the robot is determined based on its proximity to obstacles. As the robot enters the detecting range of the obstacles, its velocity is decreased whereas when the robot is not near obstacles it moves faster. The linear velocity of the robot is given by

$$v = \begin{cases} v_{max} & if\ di \le \rho_o \\ \frac{d_i}{\rho_{o\ max}} & if\ di < \rho_o \end{cases} \quad (28)$$

where v_{max} is the maximum velocity of the robot, di is the distance between the robot and obstacles, and ρ_0 is the obstacle detecting range of the robot.

Simulation Studies and Results

Low-Level Control Performance

To investigate the performance of the low-level controller for the TWBR, a simulation study was performed. The kinematics, dynamics, and design cascaded PID controller for the TWBR were implemented in Matlab and Simulink (ver. Matlab R2021b). The gains of the PID controllers were manually tuned to achieve the desired response of no steady state error with minimal overshot. In the simulation, the TWBR is given a reference linear velocity step input of $\dot{x}$ = 0.1 m/s and angular velocity $\dot{\psi}$ = 0 rad/s. From Figure 6a it can be seen that the TWBR reaches the reference linear velocity and stabilizes after about 1 s. Initially the robot moves slightly backwards to allow the robot to pitch forward; this starts the forward motion of the robot. The initial forward pitch of the robot is shown in Figure 6b, then the robot stabilizes to its vertical, balanced orientation. The motor torque output of the controller is shown in Figure 6c. Given the suitable performance of the low-level controller, high-level navigational tasks will be further investigated.

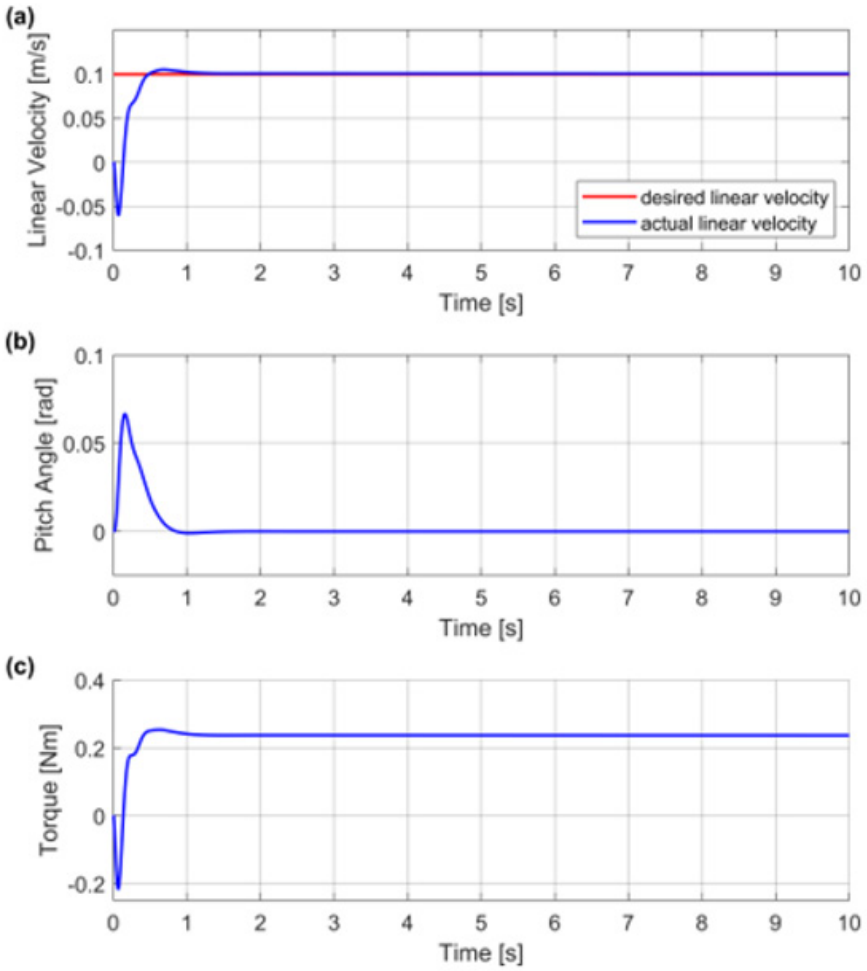

Figure 6: Response of the simulated TWBR given = 0.1 m/s and angular velocity = 0 rad/s: (a) linear velocity, (b) pitch angle, and (c) motor torque.

Static Target Tracking

The artificial potential field method described previously was integrated into the simulation to provide high level navigational control of the TWBR. Using this approach first in a static environment, given the desired target location and specifying the location of the obstacles, $\dot{x}_{ref}$ and ssss are computed and fed into to the low-level cascade controller. The environment consists of a static target and three static obstacles. The robot initially begins at the position [0, 0] with initial ψ = 0 rad, while the target is located at [10, 10]. The static environment is shown in Figure 7 where the background is given as the contour of the potential fields. The dark blue color indicates low potential while the yellow color indicates locations with higher potential. The path taken by the robot is given by the solid red line, and the goal is represented by the green star. It can be seen that the robot is able to successfully navigate through the environment collision free. Additionally, the robot is able to maintain its stability throughout the duration of the run. Figure 8a and Figure 8b show that the robot closely tracks the reference linear and angular velocities. From Fig. 8c the robot again initially pitches forward to begin its forward motion but quickly stabilizes itself within ~ 1.5 s. After the initial large disturbance, the robot maintains its pitch angle within the range of -0.0102 rad and 0.006 rad throughout its motion.

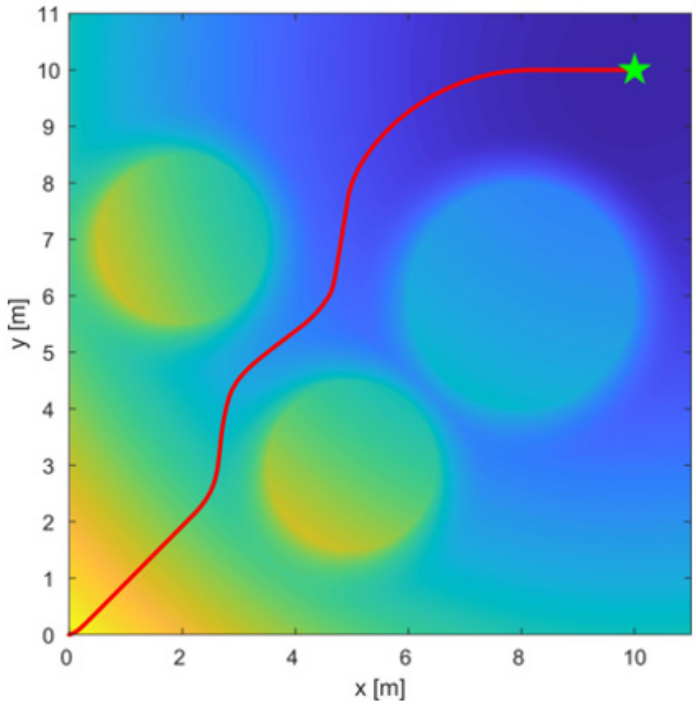

Figure 7: Static environment for the TWBR. The target is shown as the green star, and the path taken by the robot is given by the solid red line. The background is overlaid with the potential fields where the yellow color indicates regions of high potential, and the dark blue color indicates regions of low potential.

Dynamic Target Tracking with Static Obstacles

The artificial potential field method was then applied to a dynamic target tracking scenario. In this scenario, the robot exists in an environment with static obstacles, however, it attempts to track a randomly moving target. The target moves at a constant linear velocity of 0.25 m/s and its angular velocity is randomized in the range of ±1 rad/s. The robot initially begins at the position [0, 0], while the target begins at [10, 15]. The environment also includes static obstacles interfering with the path of the robot. The robot tracking the dynamic target is shown in Figure 9 where the red circle represents the robot, the green star represents the randomly moving target, and the background is set as the contour of the potential fields imposed by the target and obstacles. The path taken by robot is shown by the red line at different instances in time of the simulation. It can be seen that the robot is able to track the randomly moving target and successfully converge to its location while avoiding the obstacles.

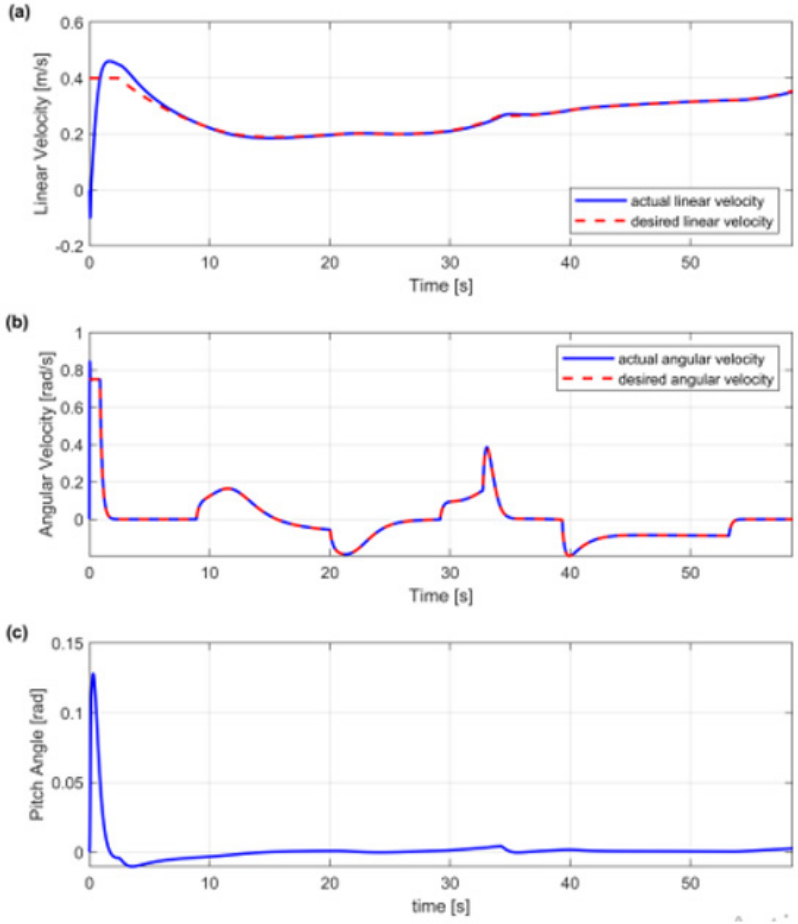

Figure 8: Response of the simulated TWBR in a static tracking environment (a) linear velocity, (b) angular velocity, and (c) pitch angle.

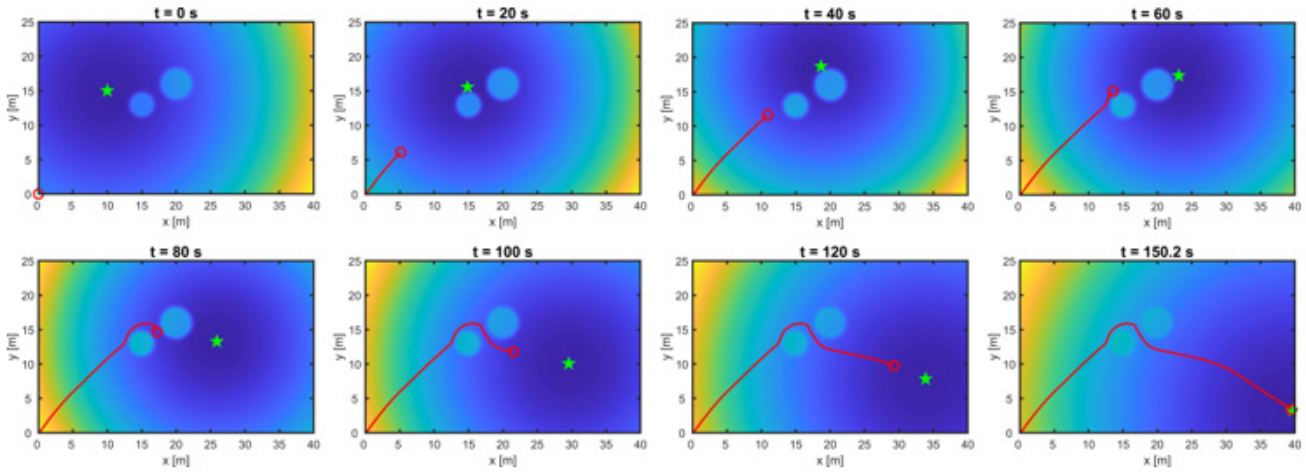

Figure 9: Trajectory of the TWBR at different time instants in a dynamic environment with a moving target and static obstacles. The target is shown as the green star, and the path taken by the robot is given by the solid red line. The background is overlaid with the potential fields where the yellow color indicates regions of high potential, and the dark blue color indicates regions of low potential.

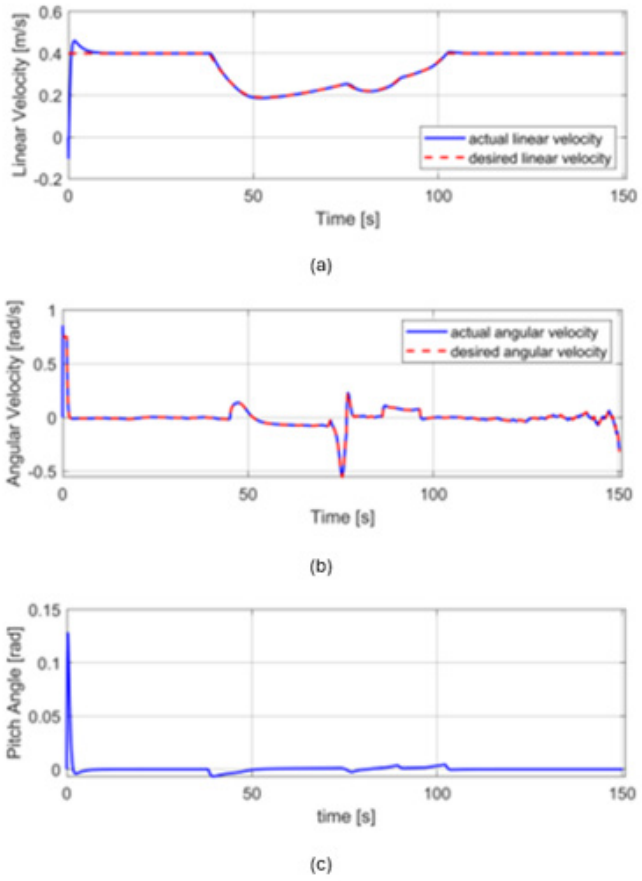

Figure 10: Response of the simulated TWBR in a dynamic tracking environment (a) linear velocity, (b) angular velocity, and (c) pitch angle.

Additionally, the robot is able to maintain its stability throughout the duration of the run. Figure 10a and Figure 10b show that the robot is able to follow the reference linear and angular velocities with minimal error. From Figure 10c the robot again initially pitches forward to begin its forward motion but quickly stabilizes itself within ~ 1.75 s. After the initial large disturbance, the robot maintains its pitch angle within the range of -0.0068 rad and 0.0043 rad throughout its motion.

Dynamic Target Tracking with Dynamic Obstacles

In this second case of dynamic target tracking, rather than avoiding static obstacles, the robot exists in an environment with dynamic obstacles. In this scenario, the target again moves at a constant linear velocity of 0.25 m/s and its angular velocity is randomized in the range of ±1 rad/s. However, rather than the obstacles existing at static positions they are given velocity profiles as well. The first target has a radius of 1.5 m and moves with a constant linear velocity of 0.28 m/s in the $\frac{3\pi}{4}$ direction. The second target has a radius of 2 m and moves with a constant linear velocity of 0.1 m/s in the $\frac{3\pi}{2}$ direction. The robot initially begins at the position [0, 0], while the target again initially begins at [10, 15]. It should be noted that the motions of the target and obstacles are independent of each other and so the obstacle's potential fields have no influence over the target. The robot tracking the dynamic target is shown in Figure 11 where similarly to the previous case the red circle represents the robot, the green star represents the randomly moving target, and the background is set as the contour of the potential fields imposed by the target and obstacles. It can be seen that as time changes both the target and obstacle potential fields are hanging, and the robot is able to adjust its path accordingly. The robot is successfully able to navigate to the randomly moving target while avoiding the obstacles. Additionally, even with the added complexity of moving obstacles, the robot is able to maintain its stability throughout the duration of the run. Figure 12a and Figure 12b show that the robot is able to follow the reference linear and angular velocities with minimal error. From Figure 12c the robot again initially pitches forward to begin its forward motion but quickly stabilizes itself within ~ 1.75 s. After the initial large disturbance, the robot maintains its pitch angle within the range of -0.0088 rad and 0.0071 rad throughout its motion.

Conclusions and Discussions

In this work, both the low-level stability control and high-level navigational control of a TWBR are investigated. The differential drive kinematics and dynamics of the TWBR were implemented in a Matlab and Simulink to enable control system design. The low-level stability controller consists of a cascaded control system that makes use of PID controllers for linear velocity and pitch angle control and a PI controller for angular velocity control. The demonstrated controller showed acceptable performance for the TWBR subject to a linear velocity step input, with minimal overshot and no steady-state error. In addition, the robot quickly stabilizes itself and maintains balance. While the result is good there are some issues to be addressed with the control system. The tuning of the PID gains were done manually which is a time intensive process requiring many iterations to converge to ideal performance of the system. This can be mitigated with an intelligent self-tuning PID system. Secondly, the PID controllers are sensitive to parametric uncertainty which leads to poor performance if any of the system parameters are changed after the gains are tuned. Future work will investigate more complex control strategies that are robust to uncertainty in estimated parameters.

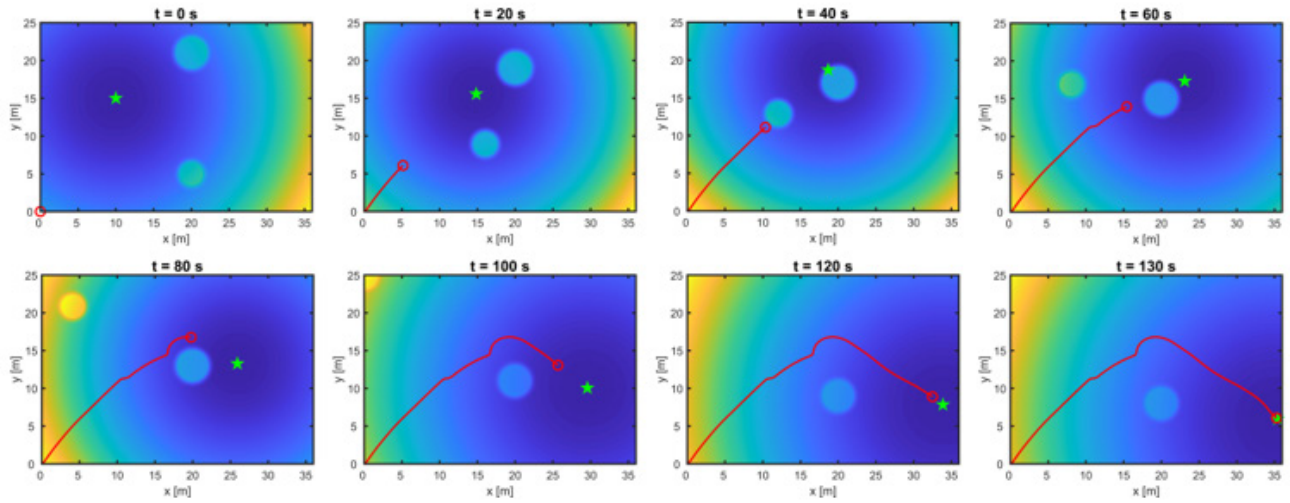

Figure 11: Trajectory of the TWBR at different time instants in a dynamic environment with a moving target and moving obstacles. The target is shown as the green star, and the path taken by the robot is given by the solid red line. The background is overlaid with the potential fields where the yellow color indicates regions of high potential, and the dark blue color indicates regions of low potential.

The high-level control of the TWBR was then investigated using the artificial potential field approach. The first simulation was done in a static environment with a static target and static obstacles, followed by two cases of the robot navigating in a dynamic environment with a randomly moving target and static and dynamic obstacles. In each case the robot was successfully able to navigate to the target while avoiding obstacles. More importantly, the robot maintained its stability throughout each of the runs. While the simulations demonstrated good performance, the technique could be improved by applying adaptive changing fields. This would be particularly beneficial in obstacle dense environments. This work seeks to fill the gap in the knowledge surrounding self-balancing robots with respect to high level navigational control in dynamic environments. It is important for the robot to maintain its stability throughout the duration of its motion while successfully navigating through the environment and converging to the target in a collision-free manner. This was demonstrated using simulations with a cascaded controller for low-level stability and artificial potential fields for high-level navigation. Our future work will investigate more complex control strategies for both the low-level and high-level control with the intent of extending the work beyond simulation and deploying the algorithms on a real prototype robot.

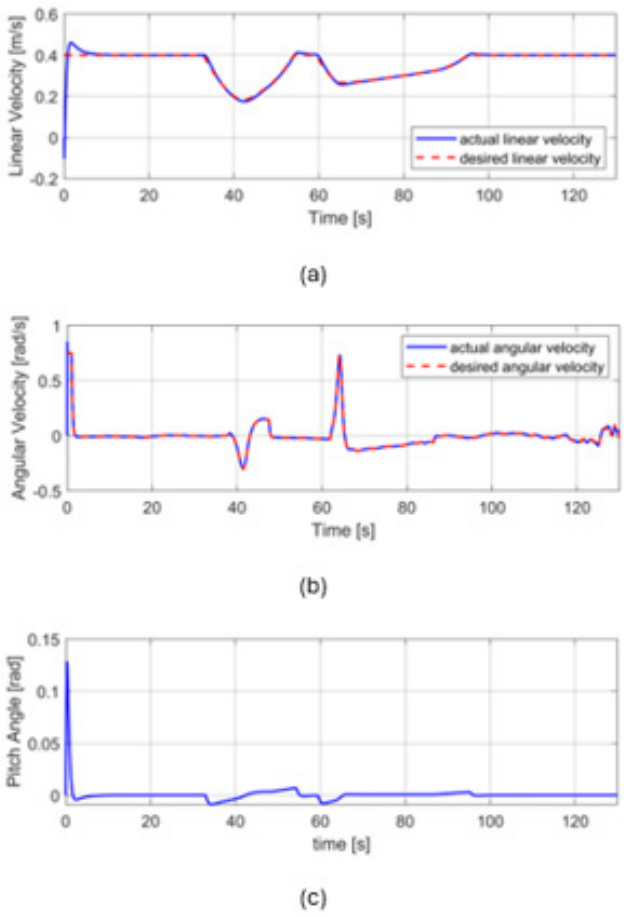

Figure 12: Response of the simulated TWBR in a dynamic tracking environment (a) linear velocity, (b) angular velocity, and (c) pitch angle.

Acknowledgements

None.

Conflict of interest

No conflict of interest.

References

1. FRubio,FValero,CLlopis-Albert(2019)Areviewofmobilerobots: Concepts, methods, theoretical framework and applications, In ternational Journal of Advanced Robotic Systems 16 (2): 1729881419839596.
2. Chan RPM, Stol KA, Halkyard CR (2013) Re-view of modelling and control of two-wheeled robots, Annual Reviews in Control 37(1): 89-103.
3. W An, Y Li (2013) Simulation and control of a two-wheeled self-balancing robot, in: 2013 IEEE International Conference on Robotics and Biomimetics (ROBIO), pp. 456-461.
4. O Jamil, M Jamil, Y Ayaz, K Ahmad (2014) Modeling, control of a two-wheeled self-balancing robot, in: 2014 International Conference on Robotics and Emerging Allied Technologies in Engineering (iCREATE), IEEE, pp. 191-199.
5. E Philip, S Golluri (2020) Implementation of an autonomous self-balancing robot using cascaded pid strategy, in: 2020 6th Inter- national Conference on Control, Automation and Robotics (IC-CAR), IEEE, pp. 74-79.
6. ANK Nasir, MA Ahmad, R Ghazali, NS Pakheri (2011) Performance comparison between fuzzy logic controller (flc) and pid controller for a highly nonlinear two-wheels balancing robot, In: 2011 First International Conference on Informatics and Computational Intelligence, pp. 176-181.
7. J Wu, W Zhang (2011) Design of fuzzy logic controller for two-wheeled self-balancing robot, in: Proceedings of 2011 6th international forum on strategic technology, IEEE 2: 1266-1270.
8. D Anisimov, TS Dang, VN Dinh, Dmitry Anisimov (2018) Development of a microcontroller-based adaptive fuzzy controller for a two-wheeled self-balancing robot. Microsystem Technologies 24 (9): 3677-3687.
9. F Sun, Z. Yu, H Yang (2014) A design for two-wheeled self-balancing robot based on kalman filter and LQR, in: 2014 International Conference on Mechatronics and Control (ICMC), pp. 612-616.
10. J Fang (2014) The LQR controller design of two-wheeled self-balancing robot based on the particle swarm optimization algorithm. Mathematical Problems in Engineering 2014.

11. S Yuan, G Lei, X Bin (2016) Dynamic modeling and sliding mode controller design of a two-wheeled self-balancing robot, in: 2016 IEEE international conference on mechatronics and automation, IEEE, pp. 2437-2442.

12. N Esmaeili, A Alfi, H Khosravi (2017) Balancing and trajectory tracking of two-wheeled mobile robot using backstepping sliding mode control: design and experiments, Journal of Intelligent & Robotic Systems 87 (3): 601-613.

13. MM Azimi, HR Koofigar (2013) Model predictive control for a two wheeled self-balancing robot, in: 2013 First RSI/ISM Inter- national Conference on Robotics and Mechatronics (ICRoM), pp. 152-157.

14. J Yao, C Lin, X Xie, A J Wang, CC Hung (2010) Path planning for virtual human motion using improved a* star algorithm, in: 2010 Seventh international conference on information technology: new generations, IEEE, pp. 1154-1158.

15. NM Amato, Y Wu (1996) A randomized roadmap method for path and manipulation planning, in: Proceedings of IEEE international conference on robotics and automation, IEEE 1: 113-120.

16. O Khatib (1986) Real-time obstacle avoidance for manipulators and mobile robots, in: Autonomous robot vehicles, Springer, pp. 396-404.

17. J Tu, SX Yang (2003) Genetic algorithm-based path planning for a mobile robot, in: 2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422), IEEE 1: 1221-1226.

18. SS Ge, YJ Cui (2002) Dynamic motion planning for mobile robots using potential field method, Autonomous robots 13 (3): 207-222.

19. O Montiel, R Sepu´lveda, U Orozco Rosas (2015) Optimal path planning generation for mobile robots using parallel evolutionary artificial potential field, Journal of Intelligent & Robotic Systems 79 (2): 237-257.

20. J Sun, G Liu, G Tian, J Zhang (2019) Smart obstacle avoidance using a danger index for a dynamic environment, Applied Sciences 9 (8): 1589.

21. SK Malu, J Majumdar (2014) Kinematics, localization and control of differential drive mobile robot, Global Journal of Research in Engineering.

22. S Kim, S Kwon (2015) Dynamic modeling of a two-wheeled inverted pendulum balancing mobile robot, International Journal of Control, Automation and Systems 13 (4): 926-933.