Robotica (2022), **40**, pp. 1475–1500
doi:10.1017/S0263574721001223

CAMBRIDGE
UNIVERSITY PRESS

RESEARCH ARTICLE

Design and development of a novel autonomous scaled multiwheeled vehicle

Aaron Hao Tan, Michael Peiris, Moustafa El-Gindy and Haoxiang Lang*

Department of Automotive, and Mechatronics Engineering, University of Ontario Institute of Technology, Oshawa, Ontario, Canada L1H 7K4
***Corresponding Author:** haoxiang.lang@ontariotechu.ca

Received: 15 January 2020; **Revised:** 30 July 2021; **Accepted:** 3 August 2021; **First published online:** 21 September 2021

Keywords: Multiwheeled vehicle; design; autonomous navigation; visual servoing; path planning

Abstract

This article proposes the design and development of a novel custom-built, autonomous scaled multiwheeled vehicle that features an eight-wheel drive and eight-wheel steer system. In addition to the mechanical and electrical design, high-level path planning and low-level vehicle control algorithms are developed and implemented including a two-stage autonomous parking algorithm is developed. A modified position-based visual servoing algorithm is proposed and developed to achieve precise pose correction. The results show significant gains in accuracy and efficiency comparing with an open-source path planner. It is the aim of this work to expand the research of autonomous platforms taking the form of commercial and off-road vehicles using actuated steering and other mechanisms attributed to passenger vehicles. The outcome of this work is a unique autonomous research platform that features independently driven wheels, steering, autonomous navigation, and parking.

1. Introduction

The world of automobiles has experienced several milestones in its development since its inception in 1886 by Karl Benz [1]. From inventions such as automatic transmissions, satellite navigation to sensor-based cruise control, automotive engineers have produced several commercialized and innovative solutions that made traveling easier, more affordable, and accessible. Today, the automotive industry is seeing its latest revolution centered around the automation of transportation systems. This revolution entails the retiring of old manual gasoline vehicles with driverless electric systems to create a more convenient and safer way of travel [2]. This is accomplished by vehicles becoming more intelligent through the addition of sensors. Benefits of this include better vehicle accessibility for those who cannot drive and car-sharing features that lessen traffic congestion. The potential to impact the streets for road users is unmatched; not to mention, its ability to generate numerous job and business opportunities around the world [2]. These advantages along with several unmentioned ones quickly made autonomous vehicle technologies an extremely sought-after research topic. As a result, substantial efforts are made by automakers, technology companies, and academic institutions to collaboratively accelerate progress in this field.

Presently, autonomous navigation features for vehicles with traditional configurations such as two axles and front-wheel steering are studied and documented extensively with real-world deployment [3,4]. However, multiwheeled vehicles have not received nearly as much attention due to their limited market. This type of vehicle finds its applications primarily in off-road and military settings because of its ability to traverse through rough terrains. As the push for more autonomous navigation capabilities continues, a shared space with mobile robots begins to emerge due to the comparable fundamentals. Likewise, mobile robots and scaled vehicle platforms are often smaller than their life-size counterpart which permits researchers to conduct experiments in indoor laboratories. This is tremendously convenient when

focusing on the different subsystem algorithms pertaining to mapping, localization, and path planning since a full-size vehicle model is not always necessary during development.

Nevertheless, a problem that exists is that commercial mobile robots sold today are generally equipped with a differential drive set up and often lack car-like features such as steering and suspension. This problem is even more apparent for mobile robots that feature multiaxles. As a result, a novel eight-wheel drive and eight-wheel steer (8WD8WS) scaled robotic vehicle is designed and developed in this article along with both high and low-level control algorithms to enable autonomous point-to-point navigation. The platform will be referred to as the scaled multiwheeled vehicle (SMWV). The motive is to create a physical prototype that features car-like suspension and steering for future autonomous vehicles and dynamics control research. The novelty of the work includes the designed autonomous research platform, capable of independent drive and steer of each of its eight wheels as well the associated kinematic model. In addition, two main algorithms are developed for the autonomous motion of the SMWV including a parking algorithm and a modified position-based visual servoing (M-PBVS) algorithm. Both algorithms are validated through experimentation in the following sections of this article.

The remaining portions of this article are organized into the following sections: related work on this topic is covered in the next section with a detailed description of the design and development of the SMWV thereafter. The different methodologies relating to motion control, path planning, and visual servoing are discussed in Section 5 and 6, respectively, with experimental results and discussion presented in Section 7. Following this, Section 8 describes the conclusions reached and recommended future works, followed by acknowledgments and references.

2. Related Work

Because of its ability to maneuver in rough terrains, multiwheeled vehicles often find their applications in off-road environments that range from agriculture [5,6] to space exploration [7]. Related work in this field is categorized into three major areas which are the novel design and development of the mobile platform [8–14], the low-level vehicle control, and the high-level navigation algorithms. The following covers publications within the past decade in each of these areas to provide readers with an understanding of the current state of the art.

Starting with the design and development of novel multiaxle vehicles, recent publications on six-wheeled robots are proposed in [8]. In these articles, the primary application focuses on space exploration robots. An interesting agriculture application is seen by Kumar et al. [9] where a novel platform operates in a garden for plant identification and classification using neural networks. The goal is to recognize and determine the amount of water and fertilizer necessary to facilitate optimal growth. The above-mentioned papers all focus on six-wheeled designs that feature rocker-bogie suspension and differential drive trains. In terms of other wheel configuration for robots, an interesting application seen by Prabhu et al. [10] uses a six-wheeled articulated robot that has its design parameters optimized for step climbing operations. Although the mentioned papers thus far feature multiwheeled drive trains; none share similarities with traditional vehicles in terms of steering such as the work conducted by Garcia et al. [11]. The authors develop a 12 degrees of freedom model for a four-wheel drive, four-wheel steer (4WD4WS) robot as well as validating the model on a prototype. Another group of researchers [12] developed a similar platform with added features for lane following, reverse and parallel parking using machine vision and fuzzy controllers. For soil sample collection and fertilizer dispensing, [13] proposes another 4WD4WS platform where a new extended Ackerman steering principle is introduced. Since robots with independently steered wheels are theoretically capable of multiple steering modes, a comprehensive analysis for a 4WD4WS platform is described in [14]. The steering modes studied include front-wheel, all-wheel, crab, and diamond steer. From the mentioned papers, recent publications on novel designs of all-wheel drive and all-wheel steer (AWDAWS) platforms are primarily focused on four-wheel variations while multiwheeled platforms achieve a maximum of eight wheels with the lack of car-like steering and suspension due to system complexity.

https://doi.org/10.1017/S0263574721001223 Published online by Cambridge University Press

In terms of low-level kinematics and dynamics control systems for multiwheeled vehicles, papers such as [15,16] set the basis for modeling of six and eight-wheeled platforms, respectively. A kinematics control law that considers wheel yaw, roll, and suspension pitch for a 4WS4WD vehicle is proposed in [11]. Aponte et al. [17] designed a dynamic model for a four-wheeled strawberry collecting robot that also analyzed tire soil interactions in addition to testing a physical prototype. Motion control with in-wheel motors is described for a six-wheel drive and six-wheel steer (6WD6WS) vehicle in [18] where vehicle dynamics performance are improved, implementing independent wheel torque and steering control; the results from this work are validated based on simulation. Vehicle stability and maneuverability are discussed in [19] where both an upper and lower controller works together to determine steering angles based on longitudinal forces, yaw moment, and tire force information. To ensure the vehicle accurately follows a given path, a bounded velocity motion controller with nonlinear control techniques is described in [20]. Beyond control system development, the controllability of a similar vehicle for high-speed navigation in rough terrains is studied in [21]. All the mentioned work in control systems up to this point along with the others that are available generally focuses on four to six-wheeled vehicles with limited literature for eight-wheeled vehicles.

In terms of high-level navigation algorithms of AWDAWS vehicles, recent publications have centered around either path planning or path following. An application in this area is seen by Li et al. [22] where a hybrid visual servo trajectory strategy is developed for wheeled mobile robots equipped with onboard vision systems and an adaptive controller is designed to periodically update the objective distance. As seen by Wu et al. [23] a nonholonomic four-wheeled robot employs asymptomatic tracking control using a neural controller. In this way, tracking errors can be reduced when subjected to actuator saturation and external disturbances simultaneously. Another study completed by Wu et al. [24] used a four-wheeled robot and proposed a control algorithm for training an artificial neural network for path planning. The system consists of two artificial neural networks to ensure optimal motion for steering from the current position of the mobile robot to a prescribed position taking its orientation into account. One of the neural networks serves to specify the position and the size of the obstacle, and the other forms a continuous trajectory to reach it. The neural network is trained on the basis of samples obtained by modeling the equations of motion in the form of Euler's Elastica. In other works, a two-wheel drive, two-wheel steer, car-like robot is analyzed alongside its kinematic model by Chen et al. [25]. In addition, the authors used the back-stepping method alongside a sliding mode controller to reduce error. Path planning using A* and the Dynamic Window Approach is implemented in research conducted by Lin et al. [26] for a 4WD4WS robot. This work is improved in [27] where pose estimation with RTK GPS and wheel encoders through an extended Kalman filter is applied. Most recently, a path planning technique that utilizes 7-order Bezier curves is developed to also provide velocity and acceleration profiles for a 4WD4WS vehicle in [28]. In this work, the vehicle is represented as a rigid body with previously determined characteristics such as mass and inertia. Conversely, recent path following algorithms includes a basic approach that considers kinematic geometry is presented in [29]. Hamerlain et al. [30] studied the control of a car-like robot using a custom-designed practical tracking controller using the second-order sliding mode control of the super twisting algorithm. Ghaffari et al. [31] utilize Mamdani fuzzy logic controllers to follow waypoints that are generated based on the curvature-derived point selection algorithm. Further development of this approach can be found in [32].

For 8WD8WS vehicles specifically, research focused on dynamics control and path planning has been published over the last few years by members of the Crash Simulation and Vehicle Dynamics Lab at the University of Ontario Institute of Technology. The work in control systems started most notably with torque distribution in [33] for an 8WD8WS vehicle. This work was later improved by [34] with a feedforward zero side slip controller that is implemented to generate the rear-axle steering angles. An optimal path planning algorithm based on the artificial potential field is proposed in [35] to drive the vehicle to a goal destination. Later, a robust heading angle controller using h-infinity is introduced to overcome system disturbances such as noise [36]. All mentioned works are tested in simulation with promising results; however, physical experiments are required for further validation.

From the mentioned literature, it is evident that an area lacking investigation is experimental research using 8WD8WS mobile robot systems. As a result, a novel SMWV that features independent suspension,

https://doi.org/10.1017/S0263574721001223 Published online by Cambridge University Press

Figure 1. *Physical model of the SMWV.*

car-like steering, and driven wheels is introduced in this article. The intent is to create an innovative platform that will serve as a research tool for future autonomous vehicle technologies. The design and development of this custom SMWV are covered in all its major areas such as mechanical, electrical, and software systems. In mechanical systems, the design of the chassis, suspension, and steering are discussed. To actuate the vehicle, sensor instrumentation and hardware architecture are described in detail along with the derivation of a kinematics model. Lastly, low-level driving and steering controllers that consider Ackerman's geometry is proposed in this work along with an incremental-based localization algorithm. All algorithms are combined with open-source path planners in robot operating system (ROS) to create a fully functioning SMWV platform with obstacle avoidance and navigation capabilities.

3. The Scaled Multiwheeled Vehicle

The mechanical system of the SMWV is classified into four different subsystems; namely, chassis, suspension, driving, and steering. Each of these subsystems is illustrated in this section along with detailed descriptions regarding the design decisions. A model is derived after all aspects of the mechanical and hardware architecture design are covered to describe the kinematics and steering geometry of the vehicle. Figure 1 shows the physical model of the SMWV that was designed and built over 2 years.

3.1. Chassis and Suspension System

The design of the chassis of the vehicle resembles the shape of the letter "T" as seen in Fig. 2. It is made of five sheets of aluminum with one for the front, back, left, right, and bottom. Each of these sheets is bent into shape and riveted together to form a rigid chassis. This design features two internal layers which are utilized for the steering and driving systems. As highlighted in Fig. 2, attached to each wheel is an upper and lower control arm which is inspired by a double-wishbone suspension system [37] which traditionally uses two "wish-bone" shaped arms to control the vertical motion of the wheel. The lower control arm is custom-made with water jet steel to ensure durability. Each suspension features gas shocks that are reinforced by coils to improve maneuverability in rough terrain. This kind of system has the advantage of creating negative camber as the wheel travels through its range vertically. Because of this, the vehicle can achieve greater handling abilities due to improved stability as tires can maintain contact with the surface.

3.2. Driving and Steering System

Next, the driving and steering systems are discussed in this section. As previously mentioned, the "T" shape chassis is designed with two internal layers housing the necessary driving and steering components

https://doi.org/10.1017/S0263574721001223 Published online by Cambridge University Press

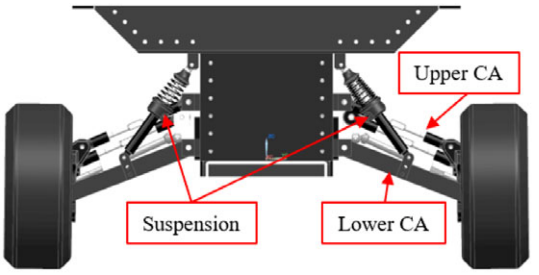

Figure 2. *Front view of the SMWV.*

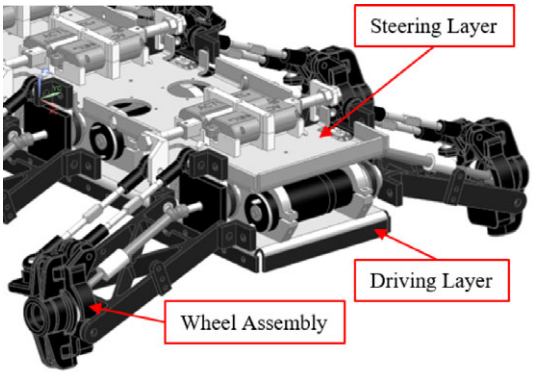

Figure 3. *Internal layers of the SMWV.*

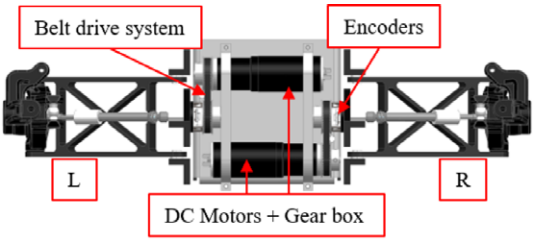

Figure 4. *Top view of the 1st axle in the driving layer.*

to actuate the vehicle. Each layer has a surface area of approximately 0.10 m^2. Figure 3 illustrates the driving layer and the steering layers.

Starting from the bottom with the driving layer, there are two DC motors per axle for a total of eight independently driven wheels. Each DC motor is attached to a 33:1 gearbox to reduce the total rotational speed while increasing the output torque. More specifically, the maximum rotational output speed is 217 rpm while the nominal output torque is 2.41 Nm. With these specifications, the vehicle can achieve a top speed of approximately 1.80 m/s. Due to the limitations imposed by the dimension of the chassis, it is not possible to align two motors along an axle for a direct drive system unless the motors were placed at an angle. This would create an inefficient drive train; therefore, a belt drive system with a 1:1 ratio per wheel is implemented instead. This setup enables the motors to be mounted in parallel with the axle axis as illustrated in Fig. 4. In this figure, the top and bottom DC motors are driving the left and right wheels, respectively. Placed in between the sidewall of the chassis and the output pulley is an encoder that is mounted on the output shaft.

The steering layer sits approximately 6.35 cm on top of the driving layer where linear actuators are attached to each wheel assembly through a tie rod. Each actuator has a total stroke of 50 mm with 25 mm being the neutral position. Steering of each wheel is accomplished by extending and retracting each

https://doi.org/10.1017/S0263574721001223 Published online by Cambridge University Press

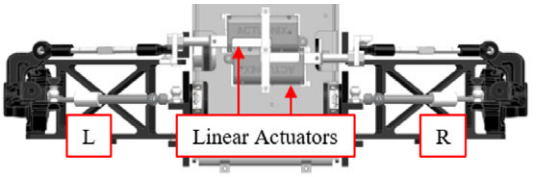

Figure 5. *Top view of the 1st axle in the steering layer.*

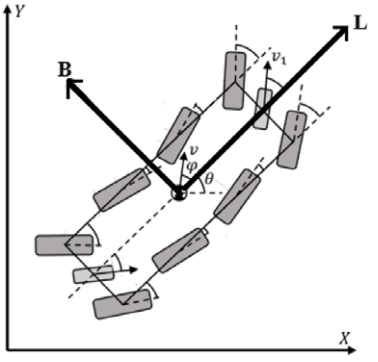

Figure 6. *Global (X,Y) and Local (L,B) reference frame of SMWV.*

actuator. In Fig. 5, an extended left actuator with a retracted right actuator would steer both wheels to the left. The tightest turn would happen at full extension and retraction, respectively. Built into each actuator are potentiometers that provide stroke position feedback. The maximum stroke of each linear actuator and achievable steering angles are 50 mm and 35 degrees, respectively. The relationship between this feedback and the achieved steering angle is derived experimentally as seen in Fig. 12, in the following sections of the paper.

3.3. Kinematics Model

3.3.1. Robot Position and Reference Frame

In this section, the kinematics model of the SMWV is derived. Siegwart & Noorkbash's reference book is utilized to provide a guideline to develop the kinematic model [38]. The complete kinematic model also utilized attributes from the two-wheel bicycle model of a vehicle. Figure 6 demonstrates the relationship between the global reference frame (signified by axes X and Y) and the local reference frame oriented in the direction of the wheelbase (B) and the longitudinal axis (L).

3.3.2. Robot Maneuverability

The SMWV is composed of eight standard steerable wheels which increase complexity when analyzing the maneuverability of the vehicle. The first component to analyze is the degree of mobility (δ_{m}) which is a measure of the number of degrees of freedom of the robot chassis that can be immediately manipulated through changes of wheel velocity [38]. As illustrated by Fig. 7, a simplified bicycle model along the longitudinal axis is illustrated between the physical wheels of the vehicle. More specifically, ($\boldsymbol{\delta_{Li}}$, $\boldsymbol{\delta_{Ri}}$) denotes the steering angles of the wheels of the vehicle while ($\boldsymbol{\delta_1}$, $\boldsymbol{\delta_4}$) denotes the steering angle average of the first and last axle.

As seen in Fig. 7 for the complete kinematic model of the vehicle, the eight wheels can be found to connect to an instantaneous centre of rotation which contributes one kinematic constraint. Therefore, the mobility can be calculated as a difference between the number of constraints and total possible degrees of freedom of the SMWV which is equal to 3, thus the degree of mobility (δ_m) is seen to be equal to 2.

https://doi.org/10.1017/S0263574721001223 Published online by Cambridge University Press

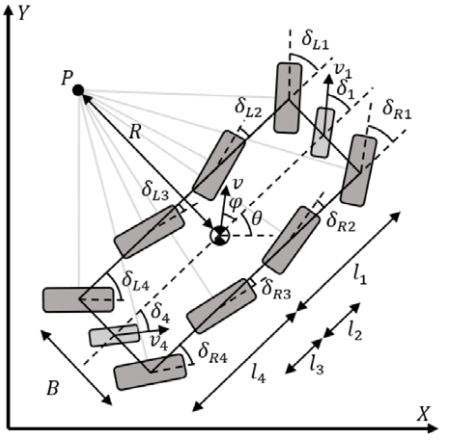

Figure 7. *Kinematic model of 8 × 8.*

Another important parameter is the degree of steerability (δ_s) which considers that steered wheels do not have an instantaneous effect on the pose of the robot chassis but affect the pose over time. The degree of steerability accounts for each independent, steerable wheel. Due to the SMWV having eight steerable wheels, in general, the degree of steerability for this vehicle is 8. However as seen in proceeding sections to streamline vehicle control, several steering configurations which coordinate and limit wheel steering angles have been developed. This, therefore, reduces the degree of steerability but allows for useful steering maneuvers.

Finally, the degree of maneuverability (δ_M) can be defined as the overall degrees of freedom that a robot can manipulate. As seen in Eq. (1), the degree of maneuverability (δ_M) is the sum of the degree of mobility (δ_m) and the degree of steerability (δ_s)

$$\delta_M = \delta_m + \delta_s = 2 + 8 = 10 \tag{1}$$

3.3.3. Robot Kinematic Model and Steering Angle Constraints

The linear velocities are calculated by the product of the total longitudinal velocity, v, and the vehicle orientation with respect to the x-axis, θ, along with the angle between the direction of the velocity with respect to the longitudinal axis of the vehicle, φ. The rate of change of the heading angle is denoted by $\dot{\theta}$, which is calculated by considering the length to the center of gravity (CG) of the vehicle from the front and rear axles. Equation (2) represents the nonlinear continuous-time relationships of the different velocities of the system where ($\dot{x}$, $\dot{y}$) are the linear velocities along the respective axis.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = v * \begin{bmatrix} \cos(\theta + \varphi) \\ \sin(\theta + \varphi) \\ \frac{\cos(\varphi)}{l_1 + l_4}(\tan\delta_1 - \tan\delta_4) \end{bmatrix} \tag{2}$$

To find the velocity at the CG, the velocity average of the first and last axle of the vehicle, (v_1, v_4), is calculated as shown below:

$$v = \frac{v_1 \cos(\delta_1) + v_4 \cos(\delta_4)}{2\cos(\varphi)} \tag{3}$$

Moving forward, the angle between v and the longitudinal axis of the vehicle is calculated with Eq. (4):

$$\varphi = \tan^{-1}\left(\frac{l_1 \tan(\delta_1) + l_4 \tan(\delta_4)}{l_1 + l_4}\right) \tag{4}$$

https://doi.org/10.1017/S0263574721001223 Published online by Cambridge University Press

Table I. *8WD8WS SMWV specification.*

Chassis dimension (mm)	**1124.6 × 606.4 × 596.3**
Total wheel base (mm)	604
Track width (mm)	460
Tire radius (mm)	88.9
Suspension rating	30 lb shocks + 10 lb coils
Total mass (kg)	40
Max speed	1.80 m/s
Max steering angle	35 degrees
Max climb angle	35 degrees

To simplify Eqs. (3) and (4), the path curvature of the vehicle, σ, along with some assumptions are considered. Starting with the curvature equation which is calculated as the inverse of the turning radius which is better defined as the quotient of angular and linear velocities. This relationship is illustrated in Eq. (5):

$$\sigma = R^{-1} = \frac{\dot{\theta} + \dot{\varphi}}{v} \tag{5}$$

Besides the curvature equation, the necessary assumptions made to simplify Eq. (2) include setting the CG location to the middle of the vehicle body and assuming the velocities, (v_1, v_4), are equal in magnitude but opposite in direction. With these assumptions applied, Eq. (2) and the derivative of Eq. (4) is substituted into Eq. (5) to form the kinematics model below:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = v * \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \\ \sigma \end{bmatrix}, \; where \; \sigma = \frac{2\tan(\delta)}{l} \tag{6}$$

As shown in Fig. 7, the scaled 8WD8WS vehicle is designed to follow Ackerman's steering geometry [39] to reduce tire degradation. Since the vehicle is in all-wheel steer mode, the instantaneous center of curvature is denoted by, P, which intersects the CG of the vehicle. From there, eight steering angles relations shown as Eqs. (7a)–(7d) representing the kinematic constraints are calculated as where l_i and t denotes the length of each axle to the CG and the track width of the vehicle, respectively.

$$\delta_{L1} = \tan^{-1}\left(\frac{l_1}{R - B/2}\right), \; \delta_{R1} = \tan^{-1}\left(\frac{l_1}{R + B/2}\right) \tag{7a}$$

$$\delta_{L2} = \tan^{-1}\left(\frac{l_2}{R - B/2}\right), \; \delta_{R2} = \tan^{-1}\left(\frac{l_2}{R + B/2}\right) \tag{7b}$$

$$\delta_{L3} = \tan^{-1}\left(\frac{l_3}{R - B/2}\right), \; \delta_{R3} = \tan^{-1}\left(\frac{l_3}{R + B/2}\right) \tag{7c}$$

$$\delta_{L4} = \tan^{-1}\left(\frac{l_4}{R - B/2}\right), \delta_{R4} = \tan^{-1}\left(\frac{l_4}{R + B/2}\right) \tag{7d}$$

3.4. Vehicle Specifications

With the mentioned subsystems, Table I summarizes all basic dimensions and performance specifications of the SMWV.

4. Electronics Hardware Architecture

The hardware architecture of the SMWV is described in this section with relationships between all electrical components and specifications listed. Starting with the full system hardware architecture in

https://doi.org/10.1017/S0263574721001223 Published online by Cambridge University Press

Table II. *Electronics component specifications.*

Component	Specification
Laptop	Intel Core i5-5300, 8GB RAM
Battery	14.8V LiPo 5000 mAh and 12V NiMh 2800 mAh
IMU	UM7-LT Orientation Sensor
Laser scanner	RPLIDAR A2M8 360 Laser Scanner
Receiver	Logitech Gamepad F710
Steering controller	Arduino Uno + DFROBOT Quad Motor Driver
Driving controller	RoboteQ SDC2130 Brushed Motor Controller + 32-bit microcomputer
Linear actuators	Actuonix P16 Micro Linear Actuator 50mm, 22:1 with a potentiometer
DC motors	Maxon DC Brushed motors with 33:1 gearbox
Encoders	AMT10 incremental encoder

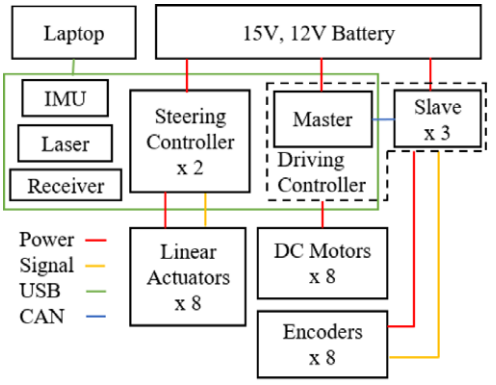

Figure 8. *Full system hardware architecture.*

Fig. 8, the central processing unit of the SMWV is an onboard laptop computer loaded with Ubuntu 14.04. This computer is interfaced with various controllers and sensors via a Universal Serial Bus (USB) hub. Beginning with the controllers, two types are embedded in the vehicle. The first type is denoted as the steering controller which controls up to four linear actuators per unit. Since one of the novelties of the vehicle is its 8WS setup; therefore, two steering control units are necessary to control one linear actuator per wheel. Besides receiving and providing 12V from the onboard power supply to the linear actuators, the steering controllers also receive feedback from a built-in potentiometer that enables closed-loop stroke/steering control. On the other hand, the second type of controller is denoted as the master driving controller where a single unit is implemented per axle to control up to two DC motors. For simplicity, the driving controllers are set up in a way where only a master is controlled by the laptop via USB and the remaining three are controlled via CAN as slave nodes. Attached to the end of every DC motor are encoders that provide feedback for closed-loop speed control. As shown in Fig. 8, each driving controller receives 15V from the onboard power supply and provides them to the DC motors where it is then stepped down to 5V for the encoders. In terms of sensor instrumentation, a 9 DOF inertial measurement unit (IMU) and 360-degree laser scanner are integrated into the SMWV along with a Bluetooth receiver for close-range teleoperation.

Table II shows the specifications of the different components embedded within the SMWV. All components are carefully selected based on size and power constraints:

5. Navigation Methodologies

In this section, the algorithms to achieve both low-level vehicle control and high-level path planning are described. These algorithms include Proportional – Integral – Derivative (PID) controllers for both

https://doi.org/10.1017/S0263574721001223 Published online by Cambridge University Press

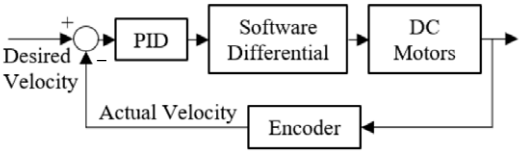

Figure 9. *PID differential driving controller block diagram.*

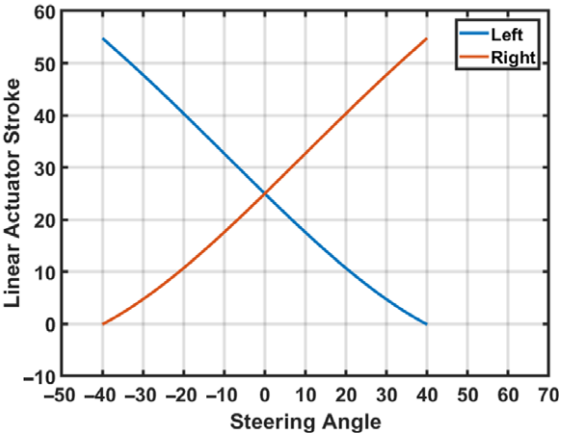

Figure 10. *Steering angle versus linear actuator stroke.*

driving and steering as well as an incremental localization method with wheel encoders and IMU. Once the low-level algorithms are established, they are consolidated with global and local path planners to achieve obstacle avoidance and navigation tasks.

5.1. PID Differential Driving Controller

Starting with the driving controller, an incremental encoder is attached to the output shaft of each DC motor for closed-loop control, as mentioned previously. A PID controller is implemented to ensure the error between the desired and actual vehicle velocity remains as minimal as possible during operation. With the encoder feedback, the controller calculates a velocity for the CG of the vehicle, v, which is used by the software differential to generate inner and outer wheel speeds in the presence of nonzero yaw commands. The goal is to improve steering maneuverability and decrease tire scrubbing. Equation (8) calculates the differential speed based on linear and angular velocity as well as the track width, B, of the vehicle. Fig. 9 shows the PID control block diagram for the motors.

$$Right_{velocity} = v - \left(\dot{\theta} * B/2\right) \tag{8a}$$

$$Left_{velocity} = v + \left(\dot{\theta} * B/2\right) \tag{8b}$$

5.2. PID Steering Controller

Once the desired steering angles are calculated based on Ackerman's geometry from Section III, an actuator stroke position control algorithm is implemented to ensure satisfactory output. Since the vehicle features independent linear actuators for steering, built-in potentiometers are used for stroke position estimation. The maximum stroke of each linear actuator and achievable steering angles are 50 mm and 35 degrees, respectively. Figure 10 illustrates the relationship between steering angle and actuator stroke based on experimental data.

Equation (9) shows 2 third-order polynomials that model this relatively linear relationship except at full actuator extension. A control law was subsequently derived based on the PID control scheme and

https://doi.org/10.1017/S0263574721001223 Published online by Cambridge University Press

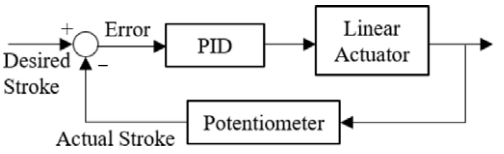

Figure 11. *PID steering controller block diagram.*

this model. Figure 11 shows the PID control block diagram for the linear actuators.

$$Stroke_{\text{left}} = \left(5 \times 10^{-5}\right) \delta_{\text{L}}^3 + 0.0014\delta_{\text{L}}^2 - 0.7589\delta_{\text{L}} + 24.974 \tag{9a}$$

$$Stroke_{\text{right}} = -\left(5 \times 10^{-5}\right) \delta_{\text{R}}^3 + 0.0014\delta_{\text{R}}^2 + 0.7586\delta_{\text{R}} + 24.974 \tag{9b}$$

5.3. Incremental Localization Algorithm

To localize the SMWV, a dead reckoning algorithm that utilizes both wheel encoders and an IMU is implemented. In the proposed strategy, the wheel encoders and IMU are responsible for linear and angular displacement, respectively. The wheel encoders show acceptable performance in short-range navigation; however, IMU drift issues are hard to ignore. From the experiment, it was determined that the yaw drift is extra prominent during long-distance navigation and when the vehicle is in an immobile state. For the former issue, a linear drift was deduced during physical trials; therefore, a compensator was integrated as a remedy. To alleviate the latter issue, an algorithm that stops updating the orientation of the vehicle when it is stationary and resumes when the vehicle becomes mobile is integrated. The results show promising capabilities for the application of this work.

5.4. Path Planning

After low-level vehicle control and localization are established, two open-source algorithms are implemented to achieve high-level global and local path planning [13]. Starting with global path planning, the classic Dijkstra's algorithm is deployed to solve the problem of generating a path between the initial and goal destination. This algorithm works by evaluating a grid cell where values are assigned to every node to represent the cost of arriving. With this information, an iterative approach is utilized to find the shortest path. To consider real-time sensor data and mobile robot dynamics, a local path planner known as the "Timed Elastic Band" is implemented. The primary goal of this algorithm is to transform a series of waypoints into a trajectory that considers time intervals. It achieves this by modifying the global planner's output based on a multiobjective optimization framework. The objective function is the weighted summation of components, f_k, which considers topics such as the nonholonomic constraint, fastest path, distance to waypoints, and obstacles. This is illustrated within Eq. (10) where B is defined as a sequence of robot pose and time intervals, $(Q,\ \tau)$:

$$f\left(B\right) = \sum_k \gamma_k f_k\left(B\right) \tag{10}$$

The previously mentioned component topics are represented in two ways. For the first way, objective functions that consider the nonholonomic constraint, as well as the fastest path, are shown in Eqs. (11) and (12). In the following equation, $d_{i,\,i+1}$ denotes the direction vector between two consecutive waypoints:

$$f_k\left(x_i,\ x_{i+1}\right) = \left\| \left[\begin{pmatrix} \cos\beta_i \\ \sin\beta_i \\ 0 \end{pmatrix} + \begin{pmatrix} \cos\beta_{i+1} \\ \sin\beta_{i+1} \\ 0 \end{pmatrix} \right] \times d_{i,\,i+1} \right\|^2 \tag{11}$$

$$f_k = \left(\sum_{i=1}^{n} \Delta T_i\right)^2 \tag{12}$$

For the second way, distance to each waypoint on the generated global path as well as nearby obstacles, are considered with the following two penalty functions. Equations (11), (12), (13), and (14) are

https://doi.org/10.1017/S0263574721001223 Published online by Cambridge University Press

Table III. *SMWV autonomous navigation pseudocode*

```
1. if (goal received)
2. run Dijkstra's Algorithm
3. launch localization algorithm
4. while (goal != arrived)
5.  run Timed Elastic Band
6.  run driving PID controller
7.  convert lin/ang velocity in to turning radius
8.  calculate appropriate steering angles
9.  run steering PID controller
10. end
```

combined with Eq. (9) to form the complete multiobjective optimization framework:

$$f_{\text{path}} = e\left(d_{\min,j},\ r_{p\max},\ \in,\ S,\ n\right) \tag{13}$$

$$f_{\text{obstacle}} = e\left(-d_{\min,j},\ -r_{\text{omin}},\ \in,\ S,\ n\right) \tag{14}$$

6. Autonomous Navigation Algorithm

The different methodologies introduced in the previous section are combined to enable autonomous navigation with the SMWV. The user is required to provide a goal pose for the mobile robot within the map frame. Once a goal is received, the global planner determines the shortest collision-free path and the localization algorithm begins to launch in the background. The local planner is responsible to transform the global plan (series of waypoints) into linear and angular velocities that are then sent to the lower controllers for actuation. Table III summarizes the methodologies into pseudocode to provide the reader with an understanding of the workings of the vehicle.

7. Modified Position-Based Visual Servoing

Alternative steering configurations are explored to develop a two-stage algorithm called modified position-based visual servoing (M-PBVS). In this work, the alternative configurations are referred to as diamond steer (DS) and synchronous steer (SS) and they are illustrated in Position B of Fig. 12. Beginning with the first stage of the M-PBVS algorithm which is to correct the orientation of the SMWV using DS. In this stage, the M-PBVS approach utilizes the onboard camera sensor to search for the visual landmark simultaneously as the SMWV pivot. Once the landmark is detected, a closed-loop orientation control using purely vision as feedback is deployed until the longitudinal axis of the SMWV is perpendicular to the x-axis of the visual landmark. By doing so, the SMWV does not depend on its odometry sensors rather just the accuracy of the pose estimation algorithm. The pivot action is illustrated for Position B in Fig. 12. Once the orientation is corrected, the M-PBVS algorithm enters the second stage which utilizes the SS configuration. Since the SMWV features a maximum steering angle, δ_{max}, of 30 degrees, two scenarios of control are possible; namely, one with a direct goal and the other with an alternate goal. The first scenario happens when the approach angle, δ_{approach}, is within the maximum steering angle such as Positions B and C so the SMWV can directly arrive at the desired. In the second scenario, the approach angle is greater than the maximum steering angle; as a result, an alternative goal is calculated to re-position the SMWV in a way that would achieve the direct scenario.

This is illustrated by positions A, D, and E. From there, the proposed M-PBVS controller proceeds to minimize the position error based on visual feedback. Since the heading angle of the SMWV does not change during SS, the visual landmark remains within the camera's field of view during its course. The next section details the mathematical model and controller design of the proposed M-PBVS algorithm.

https://doi.org/10.1017/S0263574721001223 Published online by Cambridge University Press

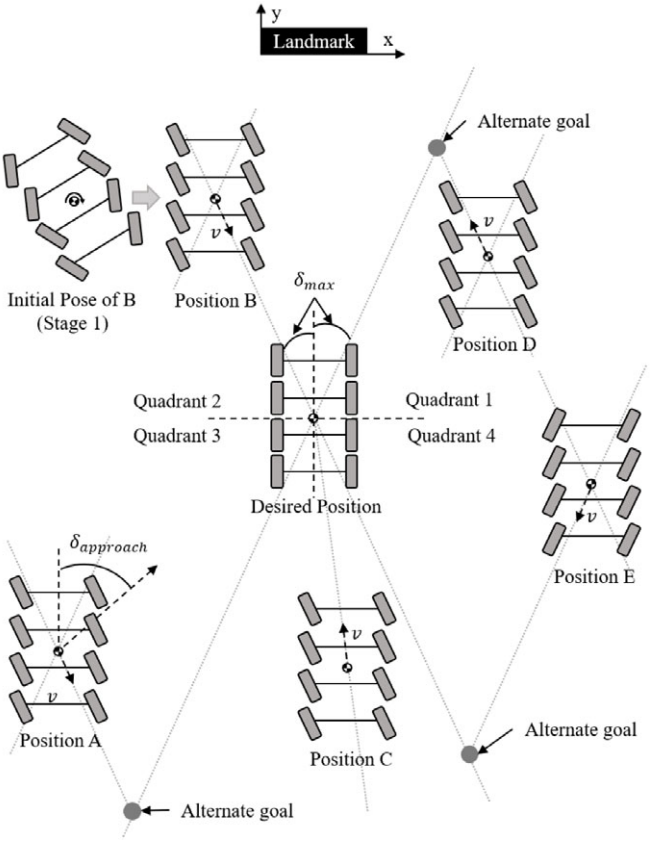

Figure 12. *Difference scenarios of M-PBVS.*

7.1. System Modeling

The forward kinematics of a two-joint manipulator is used to model the mobile robot system in this case. By establishing the relationship between different coordinate frames within the manipulator, a Jacobian matrix is derived to relate the velocity of the end effector with respect to the world, ${}^{0}_{eff}V$, to the individual joint velocities $\left[\dot{\theta}_1\ \dot{d}_2\right]^T$. This is shown in Eq. (14) where the joint velocities, $\left(\dot{\theta}_1\ \dot{d}_2\right)$ signify the robot's angular and linear velocities, $\dot{\theta}_1$ respectively. More specifically, represents the SMWV's angular velocity during DS and steering velocity during SS as discussed in later sections. This model restricts the movement of the robot to the x–y plane of the world frame with lateral slip neglected.

$$ {}^{0}_{eff}V = \begin{bmatrix} v_x \\ v_y \\ v_z \\ w_x \\ w_y \\ w_z \end{bmatrix} = \begin{bmatrix} d_2C_1 & S_1 \\ d_2S_1 & -C_1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{d}_2 \end{bmatrix} = J \begin{bmatrix} \dot{\theta}_1 \\ \dot{d}2_1 \end{bmatrix} \tag{14} $$

7.2. Stage One: Orientation Control

With the kinematics model complete, the next step is to derive the control law for stage one. To accomplish this, Lyapunov's control scheme is applied as shown in Eq. (15) where k represents the proportional gain. The error term for orientation, $e(t)_w$, is the difference between the current and desired image feature sets as extracted by a pose estimation algorithm. The proposed M-PBVS algorithm formulates the control law with respect to the desired camera frame; therefore, the orientation error

https://doi.org/10.1017/S0263574721001223 Published online by Cambridge University Press

term is equivalent to the orientation vector, ${}^{c}_{c}\phi$, since the desired orientation is zero. In this work, the pose estimation algorithm implemented is developed where the output is the position and orientation estimation of the visual landmark.

$$e(\dot{t})_w = -k\, e(t)_w = -k({}^{cd}_{c}\phi - 0)_w \tag{15}$$

Moving forward, the derivation of the orientation control law begins with representing the angular velocity of the camera with respect to the desired camera frame, ${}^{cd}_{c}w$, as the angular velocity of the camera with respect to itself, ${}^{c}_{c}w$ with the help of a rotation matrix, ${}^{cd}_{c}R$.

$${}^{c}_{c}w = \left({}^{cd}_{c}R\right)^{T} {}^{cd}_{c}w \tag{16}$$

Next, a transformation matrix, $T\,(\phi)$, is applied to convert the orientation expressed in Euler's angles, ${}^{cd}_{c}\phi$, into angular velocities of the camera relative to the desired camera frame, ${}^{cd}_{c}w$.

$${}^{cd}_{c}w = T(\phi)\,{}^{cd}_{c}\dot{\phi} \tag{17}$$

Accordingly, Eq. (17) is substituted into Eq. (16) to form the following.

$${}^{c}_{c}w = ({}^{cd}_{c}R)^{T} T(\phi)\,{}^{cd}_{c}\dot{\phi} \tag{18}$$

Since $e(t)_w$ is equivalent to ${}^{cd}_{c}\phi$; as a result, Eq. (18) is rewritten as below which considers the rate of the error, $e(\dot{t})_w$.

$${}^{c}_{c}w = ({}^{cd}_{c}R)^{T} T(\phi) * e(\dot{t})_w \tag{19}$$

By substituting Lyapunov's control scheme from Eq. (15) into Eq. (19), the following control law for the angular velocity of the camera frame is derived.

$${}^{c}_{c}w = -k\left({}^{cd}_{c}R\right)^{T} T\,(\phi) * e(t)_w \tag{20}$$

Since the SMWV is only pivoting without translation at this stage; therefore, the two joint manipulator model is reduced to just the revolute joint. As a result, the angular velocity of the SMWV's base is related to the angular velocity of the camera as shown in Eq. (21) where, ${}^{b}_{c}R$, represent an identity rotation matrix because the camera is rigidly attached to the base.

$${}^{b}_{b}w = {}^{b}_{c}R\,{}^{c}_{c}w \tag{21}$$

Next, the angular velocity of the base, ${}^{b}_{b}w$, is related to the angular velocity of the base with respect to the world, ${}^{0}_{b}w$ by Eq. (22).

$${}^{0}_{b}w = {}^{0}_{b}R\,{}^{b}_{b}w \tag{22}$$

Using Eq. (14), the angular velocity of the SMWV's base frame is described based on the angular joint velocity, $\dot{\theta}$, using a Jacobian matrix, J_w, as seen below.

$${}^{0}_{\text{base}}w = J_w\,\dot{\theta} \tag{23}$$

By combining Eqs. (20), (21), (22), and (23), the complete law that controls the angular velocity of the SMWV based on the orientation error is shown below. It is important to note that a pseudo-inverse for the Jacobian matrix, J_w^{+}, is applied to approximate the inverse kinematics.

$$\dot{\theta} = -k\, J_w^{+}\, {}^{0}_{b}R\,{}^{b}_{c}R\left({}^{cd}_{c}R\right)^{T} T\,(\phi)\;\; e(t)_w \tag{24}$$

Using differential kinematics, the angular velocity of the SMWV is further expressed in terms of wheel velocities as shown in Eq. (25).

$$\dot{\theta} = \begin{bmatrix} -\frac{r}{L} & \frac{r}{L} \end{bmatrix} \begin{bmatrix} \omega_{\text{left}} \\ \omega_{\text{right}} \end{bmatrix} \tag{25}$$

https://doi.org/10.1017/S0263574721001223 Published online by Cambridge University Press

7.3. Stage Two: Position Control

Stage two of the M-PBVS algorithm begins with first determining which of the two scenarios the SMWV is currently in (direct or alternate goal). To do this, the desired position coordinates, $(x_d,\ y_d)$, are compared with the current, $(x,\ y)$ as shown in Eq. (26). The approach angle is then measured against the maximum steering angle to determine which is larger in value.

$$\delta_{\text{approach}} = \frac{y_d - y}{x_d - x} \tag{26}$$

If the approach angle is less than the maximum steering angle then the SMWV is in the direct goal scenario where the steering angles are set as shown below.

$$\delta_{\text{steering}} = \tan^{-1}\delta_{\text{approach}} \tag{27}$$

If the approach angle is greater than the maximum steering angle, then the SMWV requires an alternate goal which is calculated by finding the intersection between lines extended from the current and desired positions at the maximum steering angle as calculated by Eqs. (26) and (27). These lines are represented by grey dash lines in Fig. 12.

$$\text{current:y} = \tan\left(\pm\delta_{\max}\right)\ x + b \tag{28}$$

$$\text{desired:}y_d = \tan\left(\pm\delta_{\max}\right)\ x_d + b_d \tag{29}$$

By doing so, the alternate goal is guaranteed to be within the achievable steering angle. Once the SMWV has arrived at the alternate goal, the new approach angle towards the desired goal should be equivalent to the maximum steering angle; thereby making it a direct goal. It is important to note that there are two alternate goal solutions at each initial position because of paths calculated based on positive and negative $\delta_{\max}$. To solve this issue, the alternate goal with the shortest distance from its current position is always selected. For example, alternate goal scenarios in Quadrant 1 and 2 would always result in the SMWV generating a forward velocity as shown by Position D in Fig. 12. On the other hand, the opposite is true for alternate goal scenarios in Quadrants 3 and 4 as illustrated by Positions A and E where the shortest distance requires the SMWV to reverse. Once the direct or alternate goal positions are either detected or calculated, Lypapunov's proportional control scheme is applied similarly to stage one. However, the only difference is that the error term in the second stage consists of two vectors which are the translational vector, ${}^{cd}_{c}s$, and the orientation vector, ${}^{cd}_{c}\phi$. Starting with the latter, the derivation of the orientation control for stage two is the same as stage one; however, steering velocity, $\dot{\varphi}$, is used for SS instead of the angular velocity from Eq. (24). Also, the revolute joint's angle is constrained by the maximum steering angle. The following illustrates the steering control law.

$$\dot{\varphi} = -k\, J^{+}_{w}\, {}^{0}_{b}R\, {}^{b}_{c}R \left({}^{c_d}_{c}R\right)^{T} T\left(\phi\right)\ e(t)_w \tag{30}$$

where $-\delta_{\max} < \theta_1 < \delta_{\max}$. On the other hand, Lyapunov's control scheme is applied to the translational vector similarly to Eq. (15) for orientation as shown below.

$$e(\dot{t})_t = -k({}^{cd}_{c}s - 0)_s \tag{31}$$

Next, the translational vector is represented in terms of translational camera velocity, ${}^{c}_{c}v$, as shown below.

$${}^{c}_{c}v = ({}^{cd}_{c}R)^{T}\ {}^{c\dot{d}}_{c}s \tag{32}$$

From here, the rate of change of the translational vector, ${}^{c\dot{d}}_{c}s$, is equivalent to the rate of the error as suggested by Eq. (31). As a result, the translational control law is written as follows.

$${}^{c}_{c}v = -k\left({}^{c_d}_{c}R\right)^{T} e(t)_s \tag{33}$$

The above control law is capable of controlling the translational velocity of the camera; however, it is not complete in the sense that it does not consider SMWV's base frame. Therefore, rotation matrices

https://doi.org/10.1017/S0263574721001223 Published online by Cambridge University Press

Table IV. *Mobile robot parking pseudocode*

```
1. run Dijkstra's algorithm  //global plan
2. while (!parking_area)
3.  run TEB  //local plan
4. Diamond steer //enter M-PBVS
5. if (tag_detected)
6.  Determine direct or alternate goal scenario
7.  Synchronous steer //position control
```

between the world and camera frame are substituted into Eq. (31) to formulate the translational control law below.

$$v = -k J_t^{+} {}^0_b R \, {}^b_c R \left({}^{c_d}_{c} R \right)^T e(t)_s \tag{34}$$

Lastly, the full control law for stage two of the proposed M-PBVS is completed by combining the steering control from Eq. (28) and the translational control as seen above to form the following where $s(d)$ is a skew matrix that describes the camera's position with respect to the base frame.

$$\begin{bmatrix} v \\ \dot{\varphi} \end{bmatrix} = -k J^{+} E L e(t) \tag{35}$$

$$E = \begin{bmatrix} {}^0_b R & 0_{3x3} \\ 0_{3x3} & {}^0_b R \end{bmatrix} \begin{bmatrix} {}^b_c R & s(d) \, {}^b_c R \\ 0_{3x3} & {}^b_c R \end{bmatrix} \tag{36}$$

$$L = \begin{bmatrix} \left({}^{c_d}_{c} R \right)^T & 0 \\ 0 & \left({}^{c_d}_{c} R \right)^T * T(\phi) \end{bmatrix} \tag{37}$$

7.4. Mobile Robot Parking Algorithm

As mentioned previously, a common limitation in recent literature regarding autonomous parking is the lack of obstacle avoidance and the use of outdated sensors. Additionally, specific coordinates of the charging station are often required. These two requirements cause fundamental problems since obstacle and parking station locations can frequently change depending on the environment. To combat these constraints, the proposed algorithm works in two stages. First, the path planners are utilized to generate a collision-free path towards the parking area. Once arrived, the SMWV utilizes the proposed M-PBVS algorithm to correct its pose and reach the desired location precisely. This method allows the SMWV to plan its path online while specific parking station coordinates are not necessary. The pseudocode of the proposed mobile robot parking approach is presented in Table IV.

8. Experimental Results

In this section, the proposed algorithm is implemented in the physical model for two different experiments. The first one is denoted as the "Slalom Test" which is intended to evaluate the vehicle's ability to maneuver between obstacles in both directions. The second test is denoted as the "Parking Test" which is intended to evaluate the navigation ability in tight spaces. For the parking test, the overall travel of the vehicle is less than 0.2 m, however, this is accurate as the purpose of this test is to correct errors in vehicle pose during parking. Both tests illustrate common scenarios that an autonomous platform would encounter. The experimental setup and results from both tests are shown below.

https://doi.org/10.1017/S0263574721001223 Published online by Cambridge University Press

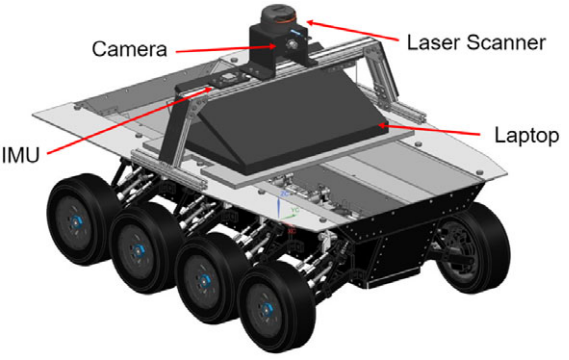

Figure 13. *Experimental setup.*

8.1. Experiment Set-Up

As previously mentioned, the vehicle is instrumented with a 360-degree laser scanner and IMU for obstacle detection and localization, respectively. Figure 13 shows the sensor placement relative to the base frame of the vehicle. The laptop is placed on top of the chassis at the center of the vehicle. The two experiments are conducted in an indoor environment with flat smooth surfaces and opaque, rigid obstacles. Due to the placement of the laser scanner at the top of the chassis, a minimum obstacle height of 0.6 m is necessary for obstacle detection during navigation.

8.2. Slalom Test

Starting with the Slalom Test, the vehicle begins at the origin with a goal at approximately $(10, -0.8)$ in the map frame. Two barrier obstacles are placed along the way with dimensions of approximately 1.0×0.6 m. Figure 14(a) successfully illustrates the robot's trajectory as generated by both Dijkstra and timed elastic band (TEB), around the obstacles towards its goal. The whole navigation took approximately 30 s with the maximum linear and angular velocities constrained to 0.3 m/s and 0.25 rad/s, respectively. In Fig. 14(b), the desired velocity remained at the maximum velocity throughout the course until the last 5 s where the vehicle slowed down to adjust its pose. Since the desired velocity represents the output of the TEB local planner, the actual velocity data exhibits significantly more noise as it is obtained from achieved wheel speeds. Regardless of the noise, it is apparent that the vehicle was able to follow the path planner commands as the overall trends of both data resemble each other. In the same graph, the inner and outer wheel velocities are also displayed. In this case, the left and right wheel velocities zig-zag one another as the angular velocities from Fig. 14(c) are taken into account. In this figure, the vehicle attempts to clear the first and second obstacle between 0–18 s and 18–28 s, respectively. By the convention assigned in the experimental setup section, a negative angular velocity implies a right steer command. With this in mind, it is easy to see that the differential speed from Fig. 14(b) matches accordingly as outer wheels always exhibit a higher velocity than inner due to turning radius difference. In Fig. 14(d), the steering angles of the front two axles are illustrated. It is important to note that the rear steering angles exhibit the same magnitude but opposite in direction (for spacing and clarity, only the front two-axle steering angles are displayed). From this figure, the maximum steering angle reached is 35 degrees. When looking closer at Fig. 14(d) around 25 s, the vehicle is attempting to steer right after clearing the second obstacle.

During this, the first axle right wheel exhibits the highest turning angle, followed by the left wheel of the same axle and then the right and left wheels of the second axle, accordingly. This relationship represents the Ackerman geometry that was discussed in the previous sections. Figure 15 shows consecutive images of the slalom experiment with the top row displaying the physical SMWV and the bottom row displaying the accompanying laser data visualization. This test shows the overall successful navigation ability of the vehicle in an obstacle-ridden environment using its steering capabilities and arriving at the destination within a minimal tolerance with regards to both position and orientation.

https://doi.org/10.1017/S0263574721001223 Published online by Cambridge University Press

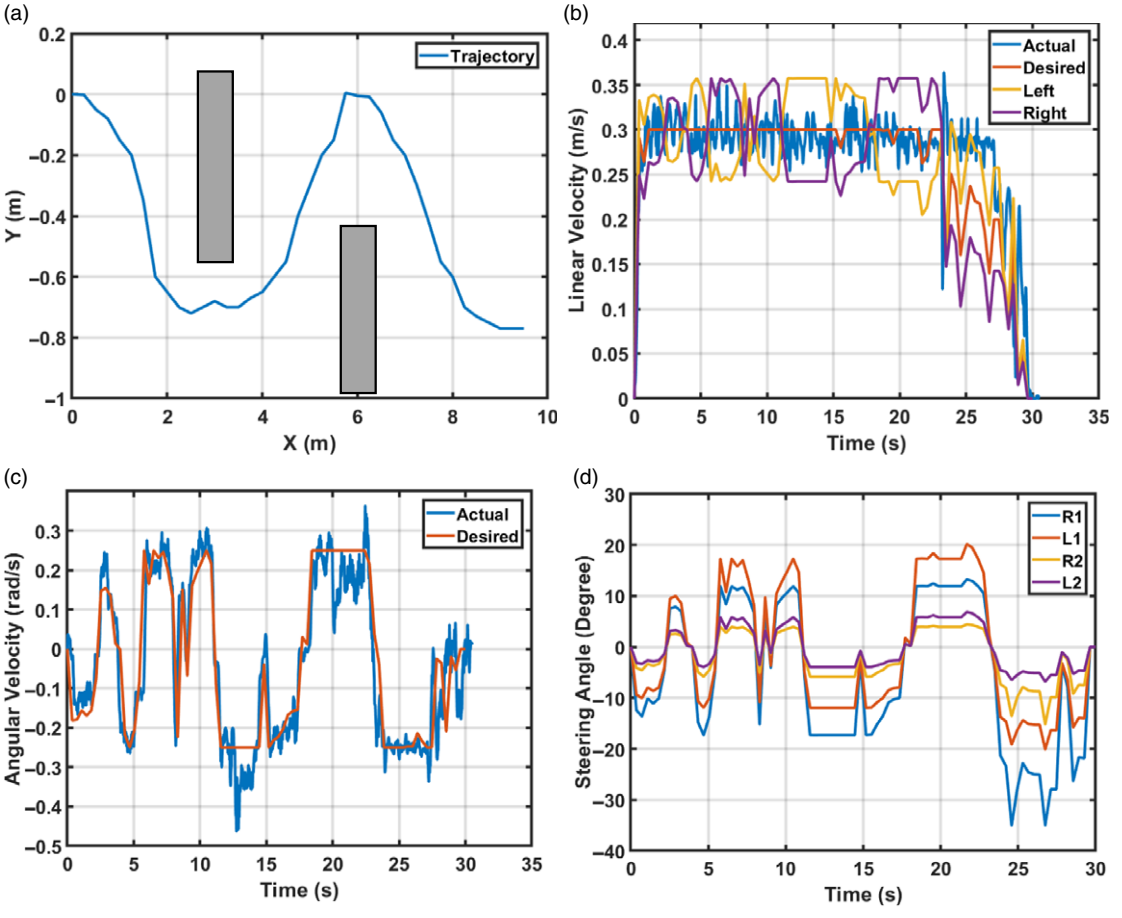

Figure 14. *(a) robot trajectory, (b) linear velocity, (c) front two-axle steering angles, (d) angular velocities.*

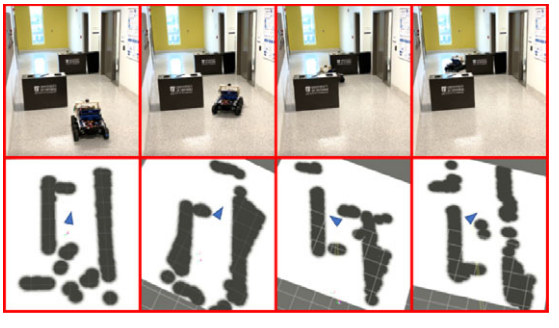

Figure 15. *Physical experiment (top) and RVIZ (bottom).*

8.3. Parking experimentation

To validate the proposed two-stage Parking algorithm, first, the navigation ability of the SMWV is tested. From there, the proposed M-PBVS algorithm is employed to alleviate any pose inaccuracies. Lastly, the M-PBVS algorithm is compared with traditional path planning to study the improvements made.

https://doi.org/10.1017/S0263574721001223 Published online by Cambridge University Press

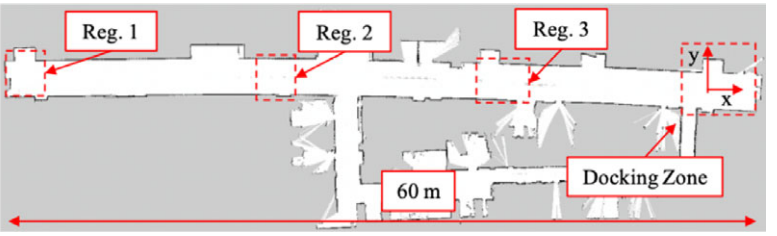

Figure 16. *Autonomous navigation experiment setup.*

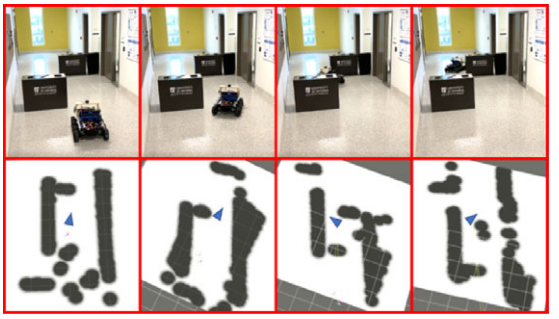

Figure 17. *Physical experiment (top) and RVIZ (bottom).*

8.3.1. Autonomous Navigation Test

This test intends to evaluate stage one of the proposed algorithms which is the SMWV's ability to navigate towards the parking area. To do so, three initial regions that are located throughout an indoor hallway as seen in Fig. 16 are chosen. The map of the test environment was acquired prior to the experimentation using standard ROS packages, namely the Open Slam implementation known as Gmapping [40]. During the experiment, the map was loaded and the robot's initial position was manually defined prior to autonomous travel. Within each region, three different positions and orientations are randomly selected as starting poses. From there, all poses are sent to the parking area via Dijkstra's and TEB algorithm to account for any obstacles along the way [41,42]. To ensure a proper reset of error, sensor power is cycled between tests. When looking closer at one of the trials, Fig. 16 illustrates the path planner's ability to utilize the onboard sensors to successfully navigate around obstacles towards its goal. Figure 17 shows consecutive images of this experiment with the top row displaying the physical SMWV and the bottom row displaying the accompanying laser data visualization. Since the accuracy of the final achieved pose is of great importance in parking applications, the test is repeated nine times. Taking a look at the overall trend, Table V tabulates the different starting poses as well as the final achieved poses with their respective average position and orientation error. From these results, it is evident that Region 1 exhibits the highest amount of drift, leading to the largest deviation of 2.06 m and 0.26 rad between achieved and desired. Region 3 achieves a more accurate result with a deviation of 0.98 m and 0.19 rad. The final positions are plotted in Fig. 18, which shows the tolerance window that is calculated by taking the average error of all final positions. The result is a circle with a 1.5-m radius. Based on the findings of this experiment, it was concluded that the SMWV can navigate autonomously; however, with an average position and orientation error of 1.54 m and 0.22 rad, respectively.

8.3.2. M-PBVS Test

With the findings of the previous experiment, the following evaluates the M-PBVS algorithm's performance in terms of close quarters pose correction. Starting with the experimental setup, all tests conducted in this section are performed in an indoor lab environment with smooth surfaces. The desired

https://doi.org/10.1017/S0263574721001223 Published online by Cambridge University Press

Table V. *Navigation test results*

Reg.	Starting $(x,\ y,\ \theta)$	Final $(x_f,\ y_f,\ \theta_f)$	Avg. Pos. Error	Avg. Ori. Error
1-1	(–60, 0, 0)	(1.85, –1.1, 0.26)	2.06 m	0.26 rad
1-2	(–58, 1, –0.17)	(–1.5, –1.4, –0.21)		
1-3	(55, 1.2, 0.09)	(–1.4, 1.4, 0.30)		
2-1	(40, –1, 0.26)	(1.63, 1.2, 0.24)	1.57 m	0.22 rad
2-2	(42, 0.5, 0)	(–1.3, 0.5, 0.19)		
2-3	(44, 1, –0.17)	(–0.9, –0.95, –0.21)		
3-1	(27, 1, –0.26)	(0.8, –0.75, 0.16)	0.98 m	0.19 rad
3-2	(26, –0.5, 0.3)	(–0.7, 0.8, 0.21)		
3-3	(23, –1.4, 0.09)	(0.6, 0.5, –0.19)		

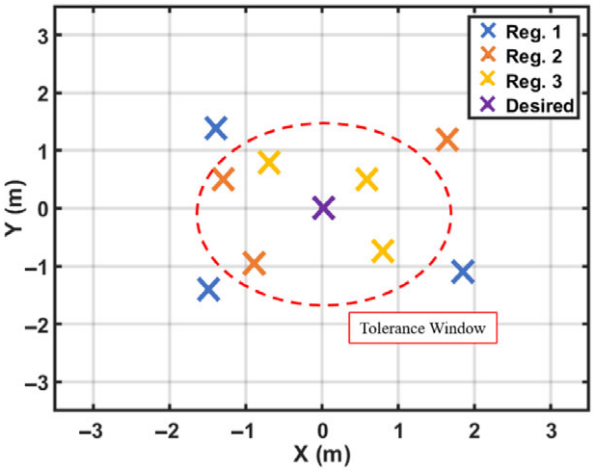

Figure 18. *Drift results.*

position is chosen in front of a visual landmark at (0, 0) with a heading angle of zero as shown in Fig. 19. Note here that the heading angle, ϑ, is measured relative to the longitudinal axis of the SMWV where a counter-clockwise rotation is deemed positive. From the desired pose, four quadrants are identified based on a cartesian coordinate system. When considering the results of the previous experiment as well as the setup presented here, two sets of experiments are conducted. The first test known as the M-PBVS Test begins with the SMWV at a position and orientation error that fits within the previously acquired result. From there, it is the M-PBVS's responsibility to determine whether it is in a direct or alternate goal scenario and then subsequently dock the SMWV as precisely as it can. The second test is called the Comparison Test which studies the performance differences between the proposed M-PBVS algorithm and the traditional path planner. Experimental data such as trajectory, linear/angular velocity as well as position and orientation errors are presented.

Starting with the M-PBVS test, position and orientation errors are introduced by placing the SMWV at position (0.15, 0.9) with a –0.2 rad heading angle in Quadrant 1. As mentioned before, the desired position is the origin of the cartesian plane with a 0 rad heading angle. In Fig. 20, the initial and final pose of the SMWV is illustrated on top of the trajectory that is generated by the proposed M-PBVS algorithm. Beginning with the first stage of the M-PBVS algorithm where an angular velocity is generated to reduce the heading error with the DS configuration. This velocity is evident during the first 4 s of the test where it reached a maximum of 0.15 rad/s, pivoting the SMWV clockwise about its center as seen in Fig. 21. Because of this, the orientation error is reduced to approximately zero as shown in Fig. 22 while both the linear velocity and position error remain unchanged. Once the orientation is corrected, the SMWV enters the second stage of M-PBVS which first determines whether it is in a direct goal or alternate goal

https://doi.org/10.1017/S0263574721001223 Published online by Cambridge University Press

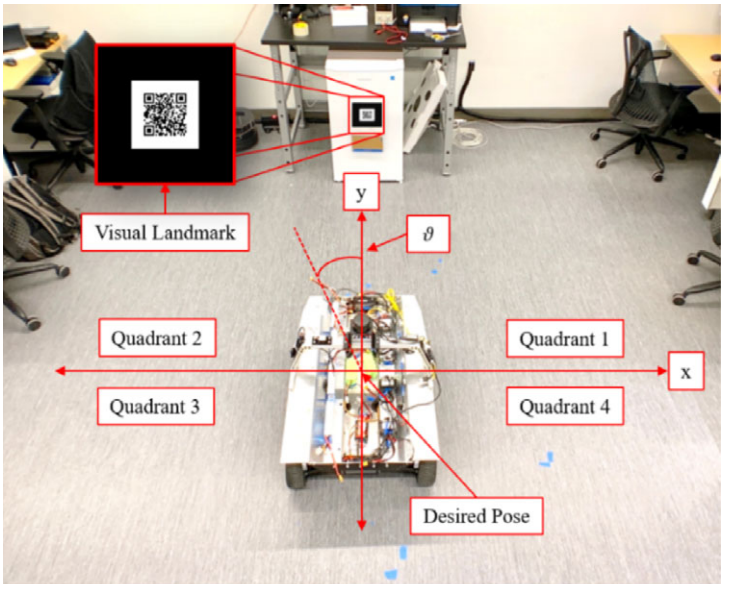

Figure 19. *Experimental setup.*

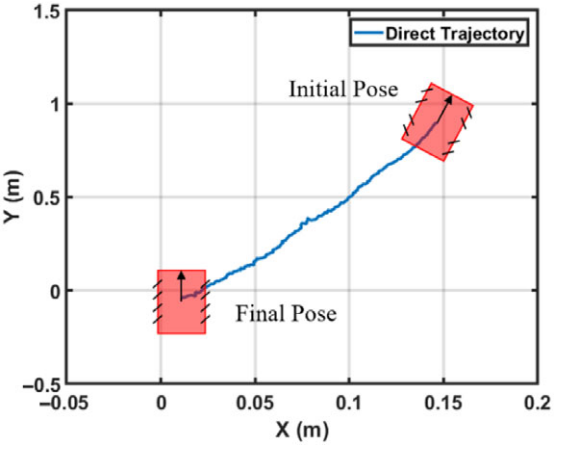

Figure 20. *Direct goal trajectory.*

scenario. Based on the calculation from Eq. (26), the initial position qualifies as a direct goal scenario as the initial $\delta_{approach}$ is approximately 10 degrees which is less than the δmax. Consequently, the generated steering angles are shown by Fig. 23 which remains between 8 and 10 degrees in a SS configuration until it arrives at the intended goal. In parallel with the steering control, the linear velocity reached a maximum of 0.2 m/s before slowly reducing to zero over the next 20 s. Also, the angular velocity is zero while the linear velocity is negative because the desired pose is set to behind the initial as shown in Fig. 22 during the second stage. Snapshots of the physical experiment along with the camera's field of view are shown in Fig. 24. The result from this test successfully illustrates the M-PBVS's ability to correct SMWV's initial pose to centimeter accuracy as the final achieved position is (0.01, –0.04) which yields an error percentage of 5.6%.

Next is the Comparison Test which is intended to gain a better insight into the performance difference between TEB and M-PBVS. This time, the initial pose is located at (0.64, –0.91) within Quadrant 2 with an initial orientation of –0.28 rad. This pose is chosen such that the M-PBVS algorithm would enter an alternate goal scenario to avoid showing similar results as the previous test. The trajectory of both

https://doi.org/10.1017/S0263574721001223 Published online by Cambridge University Press

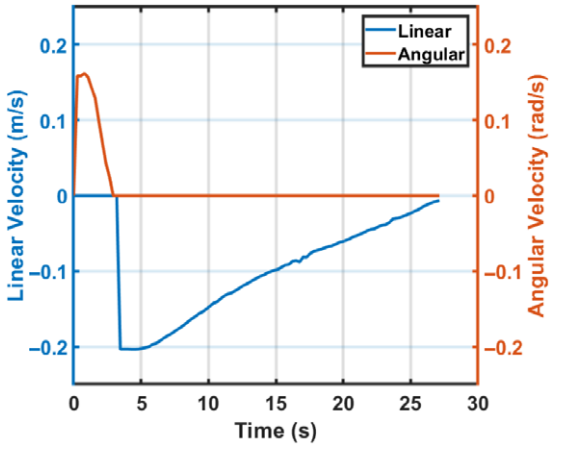

Figure 21. *Direct goal velocity.*

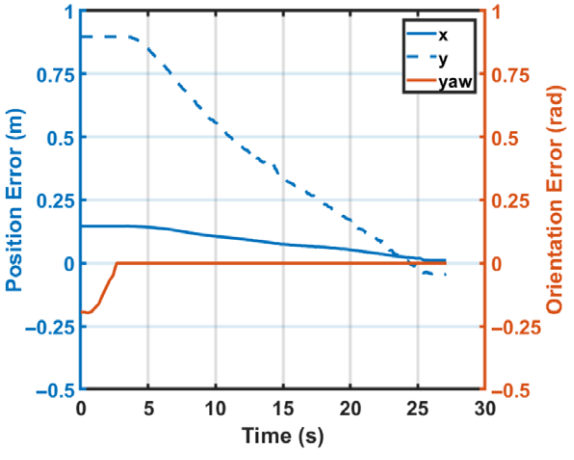

Figure 22. *Direct goal pose error.*

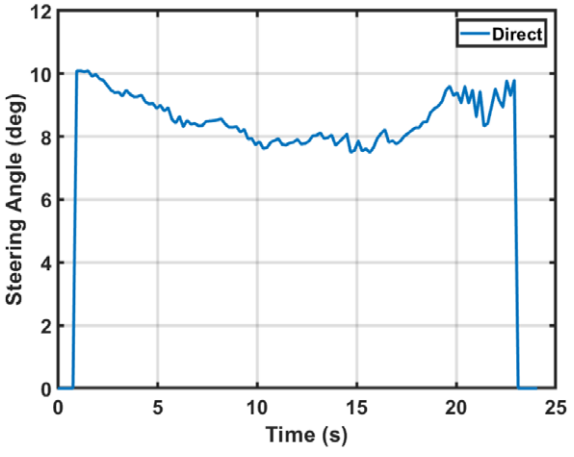

Figure 23. *Direct goal steering angle.*

https://doi.org/10.1017/S0263574721001223 Published online by Cambridge University Press

Table VI. *Comparison results.*

	Final pose $(x_f,\ y_f,\ \theta_f)$	**Pos. error**	**Ori. error**	**Total distance**
TEB	(0.23, 0.17, 0.17)	27.31%	60.71%	8.91 m
M-PBVS	(0.02, –0.05, –0.002)	4.31%	0.71%	1.85 m

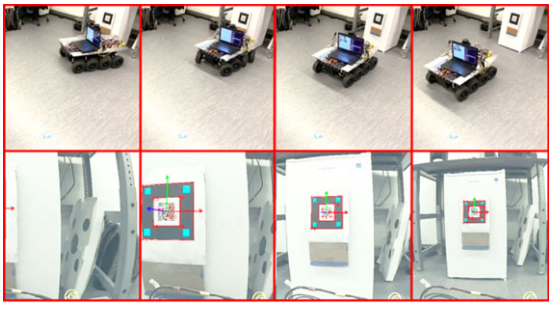

Figure 24. *Direct goal test: physical experiment (top) and camera view (bottom).*

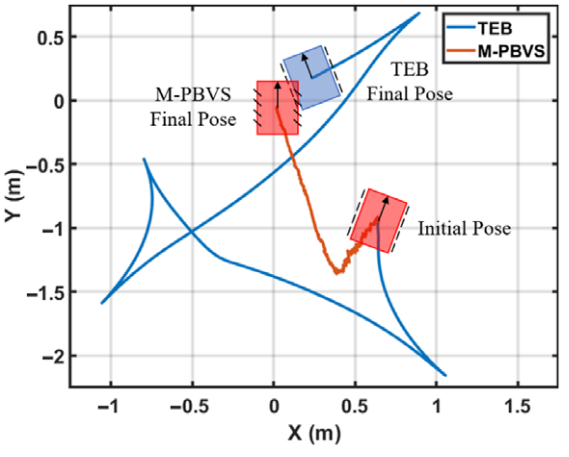

Figure 25. *Alternate goal trajectory comparison.*

algorithms is plotted in Fig. 25 where the TEB planner changes the direction of the SMWV a total of four times while the M-PBVS algorithm generates a path that only requires a change in direction once. The result of this is a much higher traveled distance of 8.91 m for the TEB when compared to the 1.85 m by the M-PBVS algorithm. Furthermore, the M-PBVS's efficiency is also prevalent when looking at Fig. 26 where the position error reduced to zero in 11 s compared to 19 s from the TEB planner. Also, the final achieved position of the TEB and M-PBVS algorithm is (0.23, 0.17) and (0.02, –0.05), respectively, which shows that the M-PBVS algorithm is much more accurate at position correction. In addition, the orientation error from Fig. 27 further validates the ability of the M-PBVS algorithm as it reduces the orientation error within 3 s with an error of 0.71%. The TEB planner utilizes multiple turns to correct its orientation yet its final result yields less accuracy than the M-PBVS algorithm with an error of 60.71%. From both comparison tests in this section, it is evident that the simplicity of the M-PBVS algorithm achieves higher accuracy and efficiency in both position and orientation correction when compared with the TEB planner. The following table (Table VI) tabulates the performance metrics between the two algorithms during this test.

https://doi.org/10.1017/S0263574721001223 Published online by Cambridge University Press

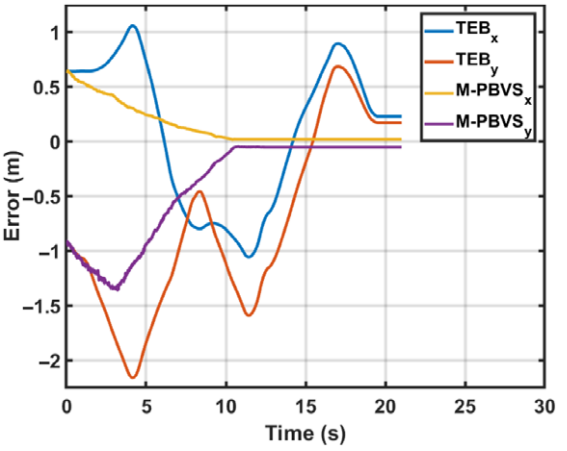

Figure 26. *Alternate goal position error comparison.*

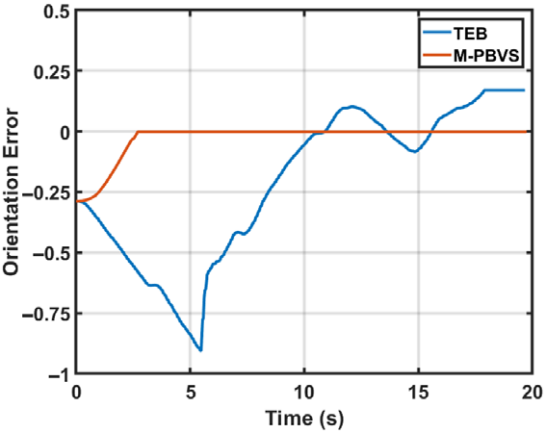

Figure 27. *Alternate goal orientation error comparison.*

9. Conclusion and Future Work

The motivation behind this work is to develop a novel 8WD8Ws SMWV platform that is capable of autonomous navigation features. As presented, all subsystems of the vehicle including chassis, suspension, steering, driving, and parking were discussed in detail. Furthermore, the electronics hardware architecture is also presented to permit independent wheel steering and driving. Beyond the hardware, the necessary software to enable low-level motion control and localization is proposed and combined with high-level path planners to achieve obstacle avoidance and navigation. For the low-level control, PID controllers for both speed and steering angles are implemented. Localization is achieved through an incremental method that combines both the wheel encoders and IMU for position and orientation, respectively. The global and local path planners employed are Dijkstra's algorithm and the TEB planner. The results are obtained from physical experiments that include two separate tests that are designed to study different performance aspects.

From the navigation experiment, the resultant trajectory for one of the experiments is presented to show the SMWV's ability in navigating around obstacles towards the parking area. Furthermore, the developed steering and speed control proved to be fully functioning and able to keep up with the high-level path planning. To further study the effects of sensor drift, the final achieved poses from nine initial poses were recorded. The results showed that the arrival position and orientation error are proportionally

https://doi.org/10.1017/S0263574721001223 Published online by Cambridge University Press

related to the distance between the desired and starting pose. In other words, SMWVs that came from the farthest initial pose finished with the largest deviation from the desired. By taking all the deviations from each test, an average error of 1.5 meters was calculated. This error was then represented as a tolerance window in the form of a circle with a 1.5-m radius.

Based on the drift results acquired from the previous test, the proposed M-PBVS algorithm must be able to bring the 1.5 m, and 0.22 rad tolerance down to within 10 cm and 0.05 rad, respectively, to ensure successful parking. For the first experiment, the M-PBVS algorithm achieved a final position and orientation error of 5.78% and 1.07%, respectively which translated to a position error of approximately 7.20 cm and an orientation error of 0.003 rad. In the second experiment, the proposed M-PBVS algorithm was compared with TEB to evaluate its performance upgrades in pose correction. The results showed that the M-PBVS algorithm was far more accurate and faster than TEB at close quarters pose correction. This was because the M-PBVS algorithm utilized a visual landmark to correct its pose effectively in a two-stage manner while the TEB algorithm is constrained by the SMWV's minimum turning radius. As a result of these limitations, the M-PBVS algorithm traveled nearly 7.06 m less than the TEB equipped SMWV. In addition, the M-PBVS algorithm repeatedly scored a higher accuracy as presented by both its position and orientation error percentage when compared with the TEB's. These results validated the benefits of the M-PBVS algorithm as it was able to capitalize on the mechanical design of the proposed SMWV platform.

For future analysis, an enhanced control scheme in addition to position-based analysis will be leveraged. This will allow for better performance with rugged terrain and other more complex environments. This will allow for the full capabilities of the 8WD8WS vehicle to be showcased as its real-world counterpart rarely finds itself being used on roads or other basic conditions. In addition, more robust sensor fusion using an Extended Kalman Filter will also be investigated. Overall, the SMWV platform was successfully developed and validated.

Acknowledgment. The authors express their gratitude to NSERC DG for partially funding this study as well as Matt Levins, Eric McCormick, and Abdul Al-Shanoon for their assistance during experiments.

Reference

1. M. Crouse, "Today in Engineering History: Car Pioneer Karl Benz Born," (in English), *Product Design & Development*, **2015** Nov 25 2016-01-13 (2015).
2. S. A. Bagloee, M. Tavana, M. Asadi, and T. Oliver, "Autonomous vehicles: challenges, opportunities, and future implications for transportation policies," *J. Mod. Transp.* **24**(4), 284–303 (2016).
3. P. M. Blok, K. van Boheemen, F. K. van Evert, J. Ijsselmuiden, and G.-H. Kim, "Robot navigation in orchards with localization based on Particle filter and Kalman filter," *Comput. Electron. Agric.* **157**, 261–269 (2019).
4. J. Peterson, W. Li, B. Cesar-Tondreau, J. Bird, K. Kochersberger, W. Czaja, and M. McLean, "Experiments in unmanned aerial vehicle/unmanned ground vehicle radiation search," *J. Field Robot.* **36**(4), 818–845 (2019).
5. O. Bawden, J. Kulk, R. Russell, C. McCool, A. English, F. Dayoub, C. Lehnert, and T. Perez, "Robot for weed species plant-specific management," *J. Field Robot.* **34**(6), 1179–1199 (2017).
6. R. F. Carpio, C. Potena, J. Maiolini, G. Ulivi, N. B. Rossello, E. Garone, and A. Gasparri, "A Navigation Architecture for Ackermann Vehicles in Precision Farming," *IEEE Robot. Autom. Letters* **5**(2), 1102–1109 (2020).
7. T. J. M. Sanguino, "50 years of rovers for planetary exploration: A retrospective review for future directions," *Robot. Autonom. Syst.* **94**, 172–185 (2017).
8. M. Islam, M. Chowdhury, S. Rezwan, M. Ishaque, J. Akanda, A. Tuhel, and B. Riddhe, *Novel design and performance analysis of a Mars exploration robot: Mars rover mongol pothik*. pp. 132–136 (2017).
9. S. Kumar, I. Gogul, M. Raj, S. K. Pragadesh, and J. Sebastin, "Smart Autonomous Gardening Rover with Plant Recognition Using Neural Networks," *Procedia Comput. Sci.* **93**, 975–981, 12/31 (2016).
10. S. G. Radhakrishna Prabhu, R. C. Seals, P. J. Kyberd, and J. C. Wetherall, "A survey on evolutionary-aided design in robotics," *Robotica* **36**(12), 1804–1821 (2018).
11. E. A. Martínez-García, E. Lerín-García, and R. Torres-Córdoba, "A multi-configuration kinematic model for active drive/steer four-wheel robot structures," *Robotica* **34**(10), 2309–2329 (2016).
12. T. H. S. Li, M. H. Lee, C. W. Lin, G. H. Liou, and W. C. Chen, "Design of Autonomous and Manual Driving System for 4WIS4WID Vehicle," *IEEE Access* **4**, 2256–2271 (2016).
13. Q. Qiu, Z. Fan, Z. Meng, Q. Zhang, Y. Cong, B. Li, N. Wang, and C. Zhao, "Extended Ackerman Steering Principle for the coordinated movement control of a four wheel drive agricultural mobile robot," *Comput. Electron. Agric.* **152**, 40–50 (2018).

14. Y. Ye, L. He, and Q. Zhang, "Steering Control Strategies for a Four-Wheel-Independent-Steering Bin Managing Robot," *IFAC-PapersOnLine* **49**(16), 39–44, 2016/01/01/ (2016).
15. C. C. G. Segura, J. C. M. Hernandez, M. S. Dutra, M. M. Mauledoux, and O. F. S. Avilés, "Ackerman Model for a Six-Wheeled Robot (ACM1PT)," *Appl. Mech. Mater.* **823**, 441–446 (2016).
16. M. Stania, "Analysis of the Kinematics of an Eight-Wheeled Mobile Platform," *Solid State Phenom.* **198**, 67–74, 03/11 (2013).
17. P. Menendez-Aponte, X. Kong, and Y. Xu, "An Approximated, Control Affine Model for a Strawberry Field Scouting Robot Considering Wheel–Terrain Interaction," *Robotica* **37**(9), 1545–1561 (2019).
18. C. Kim, A. M. Ashfaq, S. Kim, S. Back, Y. Kim, S. Hwang, J. Jang, and C. Han, "Motion Control of a 6WD/6WS wheeled platform with in-wheel motors to improve its maneuverability," *Int. J. Control Autom. Syst.* **13**(2), 434–442 (2015).
19. W. G. Kim, J. Y. Kang, and K. Yi, "Drive control system design for stability and maneuverability of a 6WD/6WS vehicle," *Int. J. Autom. Tech.* **12**(1), 67–74 (2011).
20. R. Oftadeh, M. M. Aref, R. Ghabcheloo, and J. Mattila, "Bounded-velocity motion control of four wheel steered mobile robots," ed: IEEE, 255–260 (2013).
21. A. P. Aliseichik and V. E. Pavlovsky, "The model and dynamic estimates for the controllability and comfortability of a multiwheel mobile robot motion," *Autom. Remote Control* **76**(4) 675–688 (2015).
22. F. Yan, B. Li, W. Shi, and D. Wang, "Hybrid Visual Servo Trajectory Tracking of Wheeled Mobile Robots," *IEEE Access* **6**, 24291–24298 (2018).
23. Y. Wu and Y. Wang, "Asymptotic tracking control of uncertain nonholonomic wheeled mobile robot with actuator saturation and external disturbances," *Neural Comput. Appl.* **32**(12) 8735–8745 (2020).
24. P. Bozek, Y. L. Karavaev, A. A. Ardentov, and K. S. Yefremov, "Neural network control of a wheeled mobile robot based on optimal trajectories," *Int. J. Adv. Robot. Syst.* **17**(2), 172988142091607 (2020).
25. H. Chen, H. a. Yang, X. Wang, and T. Zhang, "Formation control for car-like mobile robots using front-wheel driving and steering," *Int. J. Adv. Robot. Syst.* **15**(3), 172988141877822 (2018).
26. L. Chih-Jui, H. Su-Ming, W. Ying-Hao, Y. Cheng-Hao, H. Chien-Feng, and T.-H. S. Li, "Design and implementation of a 4WS4WD mobile robot and its control applications," ed: IEEE, 235–240 (2013).
27. H. Bo, "Precise navigation for a 4WS mobile robot," *J. Zhejiang Univ. Sci.* **7**(2) 185–193 (2006).
28. D. Penglei and J. Katupitiya, "Path planning and tracking of a 4WD4WS vehicle to be driven under force control," ed: IEEE, 1709–1715 (2014).
29. Y. Li, L. He, and L. Yang, *Path-following control for multi-axle car-like wheeled mobile robot with nonholonomic constraint*, 268–273 (2013).
30. F. Hamerlain, T. Floquet, and W. Perruquetti, "Experimental tests of a sliding mode controller for trajectory tracking of a car-like mobile robot," *Robotica* **32**(1), 63–76 (2014).
31. S. Ghaffari and M. R. Homaeinezhad, "Intelligent path following of articulated eight-wheeled mobile robot with nonholonomic constraints," in *2016 4th International Conference on Robotics and Mechatronics (ICROM)*, 173–178 (2016).
32. S. Ghaffari and M. R. Homaeinezhad, "Autonomous path following by fuzzy adaptive curvature-based point selection algorithm for four-wheel-steering car-like mobile robot," *Proc. Inst. Mech. Eng., Part C* **232**(15), 2655–2665 (2018).
33. H. Ragheb, M. El-Gindy, and H. Kishawy, "Torque Distribution Control for Multi-Wheeled Combat Vehicle," 2014. Online]. Available: https://doi.org/10.1115/DETC2014-34034.
34. M. El-Gindy and P. D'Urso, "Development of control strategies of a multi-wheeled combat vehicle," *Int. J. Autom. Control* **12**, 325, 01/01 (2018).
35. A. Mohamed, M. El-Gindy, J. Ren, and H. Lang, *Optimal Collision-Free Path Planning for an Autonomous Multi-Wheeled Combat Vehicle*, p. V003T01A002 (2017).
36. A. Mohamed, M. El-Gindy, and J. Ren, "Design and Performance Analysis of Robust H∞ Controller for a Scaled Autonomous Multi-Wheeled Combat Vehicle Heading Control," 2018. [Online]. Available: https://doi.org/10.1115/DETC2018-85032.
37. C. Moreno Ramírez, M. Tomás-Rodríguez, and S. A. Evangelou, "Dynamic analysis of double wishbone front suspension systems on sport motorcycles," *Nonlinear Dyn.* **91**(4), 2347–2368, 2018/03/01 (2018).
38. R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Introduction to autonomous mobile robots*, 2nd ed. ed. (Intelligent robotics and autonomous agents). Cambridge, MA: MIT Press, 2011.
39. E. I. Adamu, M. O. Afolayan, S. Umaru, and D. K. Garba, "The modelling and control of the drive system of an Ackermann Robot using GA optimization," *Niger. J. Technol.* **37**(4), 1008 (2018).
40. A. Koubaa, *Robot Operating Systems (ROS) - The Complete Reference*. Cham: Springer International Publishing AG, 2016.
41. B. Magyar, N. Tsiogkas, J. Deray, S. Pfeiffer, and D. Lane, "Timed-Elastic Bands for Manipulation Motion Planning," *IEEE Robot. Autom. Letters* **4**(4), 3513–3520 (2019).
42. R. Mao and X. Ma, "Research on Path Planning Method of Coal Mine Robot to Avoid Obstacle in Gas Distribution Area," *J. Robot.* **1–6**, 2016 (2016).

Cite this article: A.H. Tan, M. Peiris, M. El-Gindy and H. Lang (2022). "Design and development of a novel autonomous scaled multiwheeled vehicle", Robotica **40**, 1475–1500. https://doi.org/10.1017/S0263574721001223