IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 1, 2020, accepted June 14, 2020, date of publication June 18, 2020, date of current version June 30, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3003376

Reconfiguration During Locomotion by Pavement Sweeping Robot With Feedback Control From Vision System

LIM YI[1], ANH VU LE[1,2], ABDULLAH AAMIR HAYAT[1], CHARAN SATYA CHANDRA SAIRAM BORUSU[1], (Member, IEEE), RAJESH ELARA MOHAN[1], NGUYEN HUU KHANH NHAN[2], AND PRATHAP KANDASAMY[1]

[1]ROAR Laboratory, Engineering Product Development Pillar, Singapore University of Technology and Design, Singapore 487372
[2]Optoelectronics Research Group, Faculty of Electrical and Electronics Engineering, Ton Duc Thang University, Ho Chi Minh City 700000, Vietnam

Corresponding author: Nguyen Huu Khanh Nhan (nguyenhuukhanhnhan@tdtu.edu.vn)

This work was supported in part by the National Robotics Programme under its Robotics Enabling Capabilities and Technologies under Project 192 25 00051, in part by the National Robotics Programme under its Robot Domain Specific under Project 192 22 00058, and in part by the Agency for Science, Technology and Research.

ABSTRACT Routine cleaning the pavement is an essential requirement to maintain a sustainable environment for social life. The different width and type of pavements raise the challenges for autonomous vehicles with fixed shape to operate effectively. In this paper, we introduce the vision based reconfiguration of self-reconfigurable pavement sweeping robot called Panthera, which can adjust its frame width to ease the cleaning tasks to become friendly with different pavement geometry. The expansion and compression operations of the Panthera width are implemented by rotating one high torque motor connecting with the lead screw rod to change the opening angle of linkage hinges. The Panthera cleaning and locomotion operations are synchronized with changing the robot width according to the output of detected pavement width. To this end, the segmented pavement leveraged on the masked based deep convolutional neural network (DCNN) is used as input for the proposed closed-loop feedback control method, enabling the robot to adjust the requirement of changing the width during locomotion accurately. The proposed PID scheme takes into account the robot kinematic design with the flexibility of width changing modes. The experiments were carried out in real environments demonstrated the autonomous reconfiguration robot width with various locomotion scenarios on pavements of varying width.

INDEX TERMS Reconfigurable robots, mechanism design, feedback control, sensor fusion, pavement sweeping robot.

I. INTRODUCTION

With the increase of transport initiatives, establishments of new communities, pavement infrastructure will also increase. The pavement is the building block for cycling networks that connects cities and towns and is built in the urban environment, in the parks and underground. For example, the Land Transport Authority (LTA) of Singapore had recently completed 200km worth of sheltered walkways. Due to outdoor conditions, pavements are prone to getting dirty with leave fall from trees, dust accumulation, and litter from pedestrians. To ensure that these pavements are usable and safe, regular cleaning has to be conducted. However, regular cleaning of such networked pavements requires a tremendous workforce, and various researches have been going into automated cleaning equipment and robots to alleviate the human resources required. The robot is vital for automation of various tasks and has revolutionized many fields in the 21^{st} century. One such type of robot that is becoming increasingly important is cleaning robots, programmed to work autonomously or semi-autonomously in both indoor [1]–[3] and outdoor environments [4]–[6]. Such cleaning equipment and robots can perform repeated and routine cleaning in a pre-determined geographical setting. However, such robots

The associate editor coordinating the review of this manuscript and approving it for publication was Yingxiang Liu.

TABLE 1. Existing pavement cleaning vehicles.

Specifications	GM363 [15]	SS [16]	CN 101 [17]
Source of Power	Diesel	24V	Diesel
Noise	78dB	80dB	80dB
Holonomic	Non-holonomic	Non-holonomic	Non-holonomic
Length	3580mm	1750mm	3792mm
Width	1140mm	1310mm	1050mm
Height	1980mm	940mm	1950mm
Payload	600kg	50kg	600kg

are fixed in dimensions and are unable to adapt their size to clean pavements with different sizes. Examples of robots with fixed dimensions can be found in Table 1. Most of them are commercial pavement sweeping robots that are able to reduce the need for manpower to clean pavements. However, due to their fixed dimension, they suffer performance limitations, which impede their cleaning potential. This limitation gave rise to multiple pain points for cleaners as the robots used are unable to clean when they are unable to navigate pavements of varying widths or when a fixed obstacle blocks the pavement, and there is not enough space for it the robot to pass through. Currently, robot autonomy in unstructured and non-predefined geographical settings is intensively researched to push forward the next state of the art robot adaptation and autonomy algorithm for robots.

Self-reconfigurable robots are robots that are able to adapt their morphology according to the requirements demanded by the geographical setting. Reconfigurable robots can reconfigure to overcome constraints that a fixed dimension robot is unable to perform and have huge potential for various industries. For example, in the cleaning industry, there are even commercial products such as hTetro [7] to tackle use cases where normally fixed dimension robots are unable to perform. hTetro is a reconfigurable vacuum cleaning robot that is able to clean more area [3] and perform better than a non-reconfigurable autonomous vacuum cleaner. From the autonomous reconfigurable robot perspective, evolvability, multi-ability, and survivability are essential to reconfigurability. There are three types of reconfigurability [8]. They are nested reconfigurability, intra reconfigurability, and inter reconfigurability. There are many existing reconfigurable robots such as hTetro [7], hTetrakis [2], hTrihex [9], mantis [10], sTetro [11] and Scorpio [12]. The quadruped wheeled robot with variable wheel footprint kinematics of the Tarantula robot is presented in [13]. The pavement sweeping self-reconfigurable robot, named Panthera, was introduced in [14] with its design and reconfiguration ability. Here, the self-configuration of Panthera during operation is controlled by manually sending the control commands based on the observation while there was no reconfiguration during locomotion.

In autonomous navigation of robotic applications, implementation of a control system which adapts flexibly to various working conditions such as the roughness, the gradient of terrains is primarily needed. Different types of controller techniques are used for autonomous vehicles. Smoother control and driveability of robots can be done via a PID controller. A Genetic algorithm (GA) [18] is used with a PID parameter to control the speed of the motors, fuzzy logic with PID controller [19], and auto-tuning the PID [20] are done for the improvised dynamic behavior of the DC motors. A reliable and robust PID controller is implemented to control the position, speed of the steering unit, and lead screw DC motor. Since the flexible and dynamic gaits available of reconfigurable robot, the feedback control techniques with the parameter input from the sensor such as object features from vision is needed to archive the smooth locomotion.

Specifically, considering the pavement sweeping activities, when maneuvering along public pavements, the fixed shape of existing cleaning equipment and robots results in the high probability of obstructing the pedestrians on pavements and limiting the working environment. For instance, when a robot encounters a small pavement or an obstruction in the pavement, they are unable to clean the pavement, which inhibits its performance. Therefore, a robot should be able to adapt flexibly in such conditions to improve robot and surrounding environments synergy. These are the motivations of design a reconfigurable pavement sweeping robot. Reconfigurable ability, however, raise other challenges during the operation of the robot. Since the pavement width size varies, to archive fully autonomous operation, reconfigurable robots need to understand to address the questions such as when it should reconfigure its size and how fast it should reconfigure its size. However, there are limited works that describe using perspective sensors to understand dynamically changing pavement widths and how to use the information to perform a controlled reconfiguration during locomotion of a pavement sweeping robot. To this end, a novel perspective algorithm needs to be integrated with the reconfiguration during locomotion kinematics in order to derive the pavement width information in front of the robot.

Multiple approaches have been proposed to find the pavement width using advanced sensors and computer vision algorithms. The sensor fusion approach is commonly used is the RGB camera or RGB-D camera [21] for pavement width measurement. Algorithms such as Canny edge detection [22] to detect the edges of the pavement and subsequently converted the image coordinates of the edges to real-world coordinates [23] are well developed and works well for most cases. However, it is unable to detect the edges of the pavement if obstacles block the pavement. The detection for objects such as the pedestrian, group of pedestrians [24] on the pavement needed to be addressed to make the autonomous function. Another issue is that different lighting conditions will affect the canny edge detection. Although it can be reduced by converting RGB camera to HSL or HSV for the canny edge, stereo cameras are used as sensors to enhance the robustness in varying light conditions.

Another approach is to use semantic segmentation [25], which is a combination of object detection, classification, and localization. Over the past ten years, there have been increasing researches in deep learning convolutional network,

FIGURE 1. Panthera: Reconfigurable pavement sweeping robot.

especially in the field of autonomous cars. Deep learning [26] can employ neural networks that can identify unique features of different objects automatically. Hence, the deep learning approach significantly outweighs traditional detection algorithms such as feature-based Histogram of Orientation, SIFT techniques [27]. The difficulty in using deep learning is the high computational power [28] required as large networks generally need very high computational power. In recent years, new technologies have pushed forward computational power and led to the development of parallel computing [29], which allows computation to be done in a shorter time. Some examples are the development of Compute Unified Device Architecture (CUDA) [30] and NVIDIA CUDA Deep Neural Network (CuDNN) [31], which allows computers to process deep learning computations with Graphic Processing Units at a much faster rate. Besides, with the advances in the Internet of Things (IoT) and cloud computing [32] where the processing of high computational power algorithms required by robots can be executed online on readily available technologies such as Google Colab, Amazon Web Services (AWS) or Microsoft Azure [33], segmented segmentation has become much more reliable and feasible for autonomous robots.

In this paper, a detailed explanation of a pavement sweeping robot named Panthera with self-reconfigurable its frame size, as seen in Figure 1 will be given. Panthera has the ability to perform sweeping while it reconfigures its width during moving, and has multiple locomotion abilities such as a zero angle turning radius and has the mechanisms to adapt to its environment dynamically. Moreover, we propose the Panthera's perception to allow Panthera to perform its sweeping while adapting to changing pavement widths in dynamically changing geographical settings of outdoor pavements. Using the proposed perception ability, Panthera will derive steering unit speed, lead screw speed as inputs for its kinematics and control to perform reconfiguration smoothly. With Panthera's perception, Panthera will be able to use the input for kinematics and control to function safely and develop a

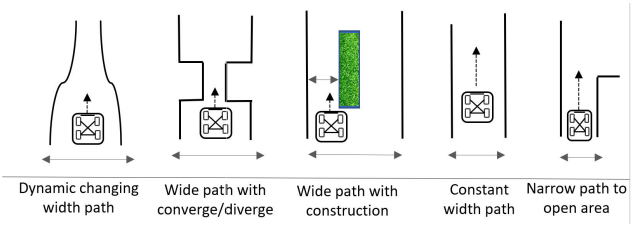


FIGURE 2. Pavement work space situations for the cleaning vehicle.

greater affinity to its dynamically changing pavement surroundings as it adapts accordingly. Using RGB-D sensors and new methods of semantic segmentation such as SegNet [34], DeepLab [35], pixel coordinates of pavements, and in turn, real-world coordinates and estimates of pavement widths at different sections of the image can be determined. From the different estimated pavement widths, the control scheme for reconfiguration planning of the robot can be developed. Reconfiguration planning output includes the rate of reconfiguration in the width and speed of respective robot motors. Robot Operating System (ROS) [36] master will make decisions based on RGB-D sensors input and adaptively control the robot locomotion and reconfiguration by controlling lead screw motor speed, wheel motors speed and heading angles of the steering units. This ensures that the robot is able to adaptively change its width in dynamic environments where pavement width changes and pedestrians density changes with a high level of control and safety. Adaptively changing its width during locomotion while performing sweeping operations enables Panthera to improve the sweeping process as it enables the robot to perform sweeping even while navigating through pavements of varying width which is not possible for fixed dimension robots. When the pavement width is tight, it reduces its width to navigate through small spaces, when the pavement is large, it increases its width to maximize area under the robot so that more can be swept under Panthera, thus saving energy and optimizing the cleaning operation.

The rest of this paper is organized as follows. Section II gives the detail on the robot design. The varying pavement width reconfiguration is presented in Section III. Section IV describes the kinematic robot locomotion during reconfiguration while moving operations. The experimental result is detailed in Section V. Finally Section VI concludes the paper.

II. ROBOT ARCHITECTURE

In this section, the robot architecture of Panthera [14] is dealt with in detail for brevity. The terminologies and their definition are shown in APPENDIX.

A. DESIGN PRINCIPLES

The work in [37] highlights the required features and architectural needs for autonomous mobile robots in real-world scenarios. Figure 2 shows the different common real-world scenarios that a cleaning robot will endeavor: a) Dynamic changing pavement with varying width, b) Wide path with converging/diverge such as changing pavement widths,

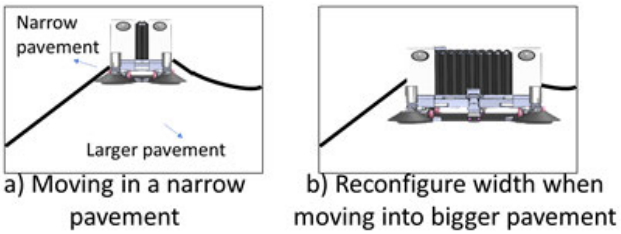


FIGURE 3. Proof of design for Panthera with reconfigurable ability.

c) Wide path with construction, d) Narrow pavement with constant pavement width, e) Narrow path to open area or wider pavement with varying width. Panthera is a robot that is designed to tackle such scenarios and has the following capabilities:

- Panthera has four differential drive steering units for locomotion and is capable of moving sideways.
- Panthera is designed to have a zero turning radius and can make very sharp turns.
- Panthera has a visual perception capability to understand dynamically changing pavements and pedestrian density.
- Panthera is reconfigurable in the width and range from 70cm to 200cm.
- Panthera has a payload of up to 200kgs.

The proof of concept of the self-reconfiguration of the platform for the narrow and wider pavement is depicted in Figure 3. For brevity the design layout for the robot is discussed in brief in the next section.

B. DESIGN LAYOUT

The design of the Panthera [14] can be classified into 5 different components: Electronic Design, Structural mechanical design, Cover design and Locomotion mechanical design.

1) ELECTRICAL DESIGN

The traction batteries are twelve 2V batteries in series, which are used to power the eight-wheel DG-158A 24V motors, two motors to rotate the scrubbing wheels, and a lead screw motor. Proximity and auxiliary sensors are attached to the body for reconfiguration and navigation. Batteries are connected to the RoboClaw microcontroller, which is controlled by an Arduino Mega 2560. Incremental encoder and absolute encoder data are sent to the Arduino for feedback control, and the Arduino will output the commands for RoboClaw to control the wheel motors, lead screw motor, and the brush motors. RGB-D sensors data will be processed by an industrial computer that is running on ROS. The industrial computer will send the commands to the Arduino which will, in turn, send the command to the RoboClaw to perform locomotion and reconfiguration.

2) STRUCTURAL DESIGN

Structural mechanical design is the body of the Panthera shown in Figure 5. It consists of an Aluminium frame that

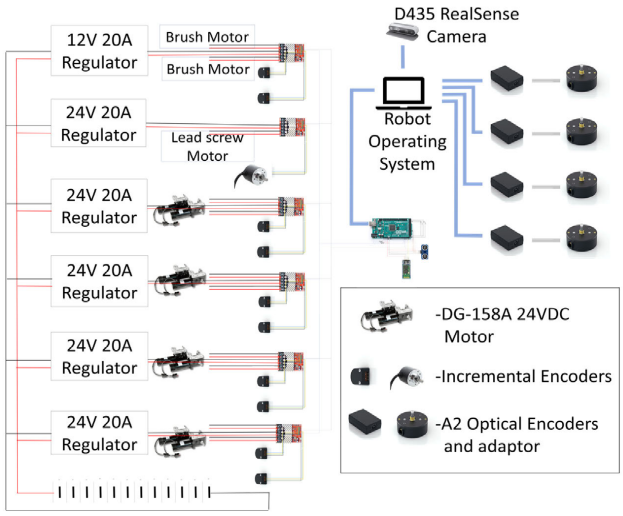


FIGURE 4. Electrical layout of Panthera.

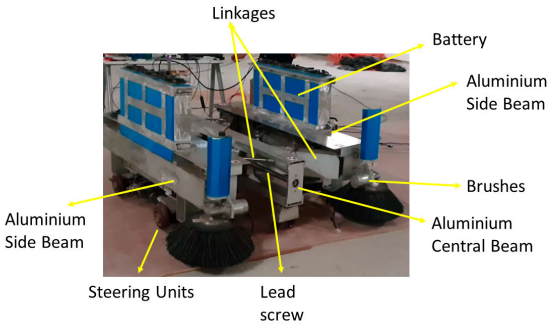
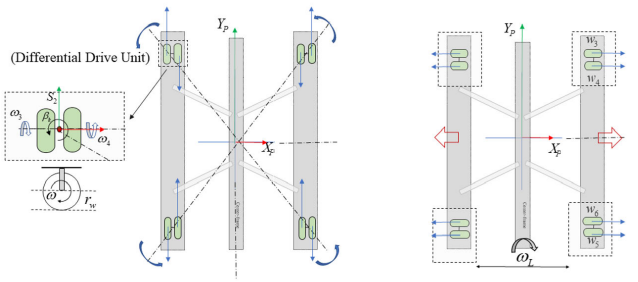


FIGURE 5. Structural and Mechanical Layout.

supports the wheels, brushes, cover, battery, and a scissors mechanism responsible for the reconfiguration [14]. In the scissor's mechanism, Panthera has a central beam that connects to two side beams via linkages. The central beam is a hollow rectangular cross-section of 8 mm thickness and supports the lead screw (primary diameter = 32 mm, pitch $p_{leadscrew}$ = 10 mm) with the half of its length machined as with left-handed thread and another half with a right-handed screw thread [14]. Four linkages are used to connect the central beam to the respective side beams, and one of the four linkages are connected to the lead screw. The linear motion of the lead screw carriage is transmitted to the side beams via the linkages, and as a result, the side frame will be able to expand and contract. The one end of the screw thread is connected to the driving motor via universal coupling and other end fixed with flange bearing. The lead screw actuator is powered by a 110 rpm. 24 Volt motor. When fully contracted, dimensions of Panthera are 1.75 by 0.8 by 1.6 meters and, when fully expanded, is 1.75 by 1.7 by 1.6 meters.

3) COVER DESIGN

The cover of the Panthera is a custom-built 1.5mm artificial lather fabric-based bellow design [14] that is able to contract



a) 4 differential drive unit on Panthera side beam

b) Differential drive units during reconfiguration

FIGURE 6. Differential driving units and Panthera frame.

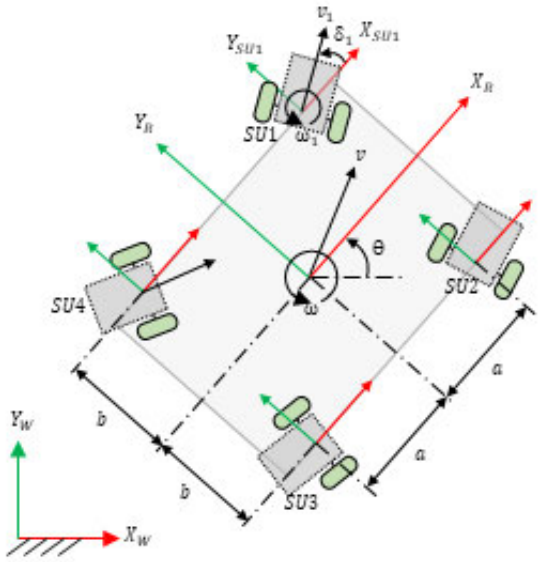


FIGURE 7. Panthera with 4 differential drive steering units.

and expand with the platform. It is further reinforced with hollow stainless-steel pipes and is attached to side frames. The side frames are connected to the side beams. The cover also protects the mechanisms and electronics in Panthera. It prevents dust and other foreign materials such as leaf fall from disrupting Panthera functionality during operation.

4) LOCOMOTION MECHANICAL DESIGN

Panthera is similar to a four-wheel-independent steering and four-wheel-independent driving (4WIS4WID) mobile robot. Compared to common 4WIS4WID robots, Panthera has differential drive steering units to replace the four wheels in 4WIS4WID, as illustrated in Figure 7. The advantage of having differential drive steering units over four standard wheels in 4WIS4WID is the reduction in required rotational torque about the steering axis Z_{SU_i} (directed out of the paper) and hence quicker steering response. The locomotion design of the Panthera consists of the wheel's locomotion and the lead screw scissors mechanism. The length of the lead screw between the coupling and the bearing as 59 inches, 0.9 inches,

FIGURE 8. Steering unit direction.

the theoretical, critical speed is found to be 1320 rpm [14]. The lead screw mechanism is constrained by the linkages and the fixity of the bearings. Three points on bearings support the lead screw with two at the end of the shaft and one in the middle of the shaft. The lead screw is supported between motor coupling and the bearing, leading to a fixed-fixed boundary condition. The maximum operating speed of the screw can be considered as 1056 rpm. The locomotion of Panthera consists of 4 differential steering units, as seen in Figure 6. Two differential units are connected to each side beam. Each of the differential units consists of two standard wheels. Each wheel is driven by a DC brushed geared motor of 24 Volts and 130 rpm. The differential drive steering unit is able to perform a 360-degree rotation, achieving an omnidirectional driving unit. They can be controlled independently by providing the desired steering angle required seen in Figure 8. The differential drive steering unit is connected to the side beam with two hinges to distribute the robot's weight evenly. From Figure 6, during reconfiguration to expand its width, the differential drive units will be angled away from the center of the Panthera while the lead screw mechanism actuator will rotate. The wheels' speed moving away from the center beam must be synchronized with the lead screw actuator and controlled by an Arduino.

C. SYSTEM ARCHITECTURE

The ROS-based block diagram of the proposed system using a client-server model is shown in Figure 9. The proposed system is built in the workstation machine with the eight cores CPU, 16G Ram, and separate GPU card located inside the body frame of Panthera [14]. The motors for eight locomotion wheels, two motors for scrubbing wheels, and the control system, sensors devices of the Panthera, are powered by the 24 volts batteries. The LIDAR sensors enabling autonomous navigation are fixed on top of the robot platform. The perception unit using an RGB-D RealSense D435 camera with Field of View (FOV) 135 degrees is mounted facing the front view to extract color and depth streams of surrounding scenes. The objects outside the FOV are not captured by the realsense camera. This information is transmitted to the server to conduct the high-level processes of object understanding that enables Panthera to adjust the shape according to the width and distance of the objects in the environments.

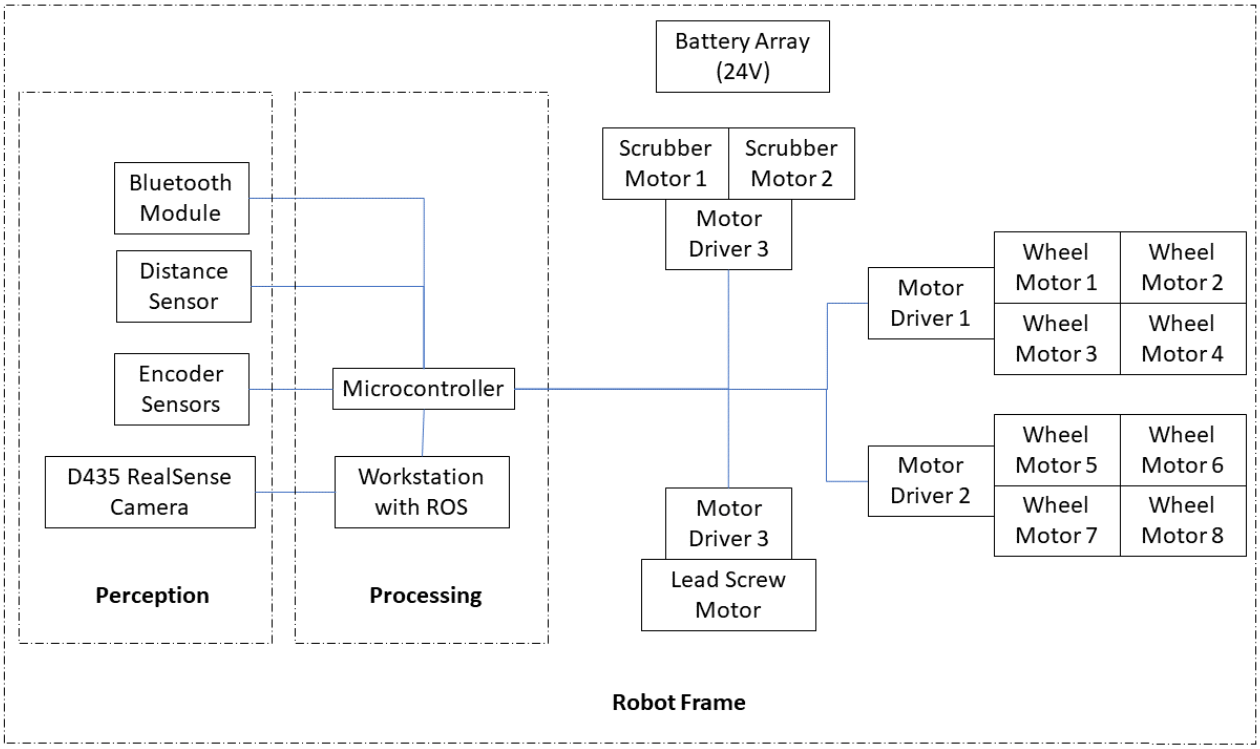


FIGURE 9. Panthera system architecture.

III. SELF-RECONFIGURATION WITH VARYING PAVEMENT WIDTH

In this section the framework for the reconfiguration and the methodology are discussed.

A. THE FRAMEWORK OF DYNAMIC PAVEMENT AND PANTHERA SYNERGY

This paper uses deep learning convolutional neural network semantic segmentation for the perception of the dynamically changing pavement width environment to enhance synergy between Panthera and its surroundings. Panthera stream images from the D435 RealSense camera to determine the pavement widths, w_i and depth from Panthera, d_i, in different sections, $i = 1, 2$, of the image. From the pavement width and based on its current locomotion velocity, it will calculate the steering angle required for the differential drive steering units. The rate of reconfiguration, which involves the lead screw motor speed and differential driving unit speed and steering angle, will be derived. Therefore, using information about the pavement width, Panthera will be able to reconfigure its dimension accordingly to operate in the pavement with varying width. It is assumed that the robot is parallel to the pavement walkway. To ensure safety and performance during reconfiguration, the lead screw motor rotary encoder is used to derive the current reconfiguration state of the robot and determine the width of the Panthera body frame where a proportional integral and differential control is implemented. This allows Panthera to adaptively change its reconfiguration

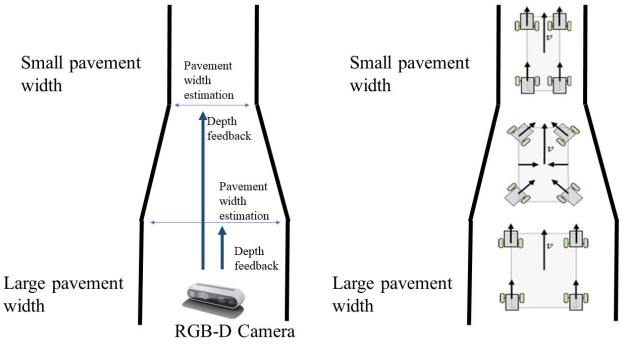


FIGURE 10. Panthera adapting to dynamic varying pavement width.

speed when demanded from the perspective algorithm. The only limitation for the change in reconfiguration speed is the maximum speed of the lead screw motor, which will provide an operating maximum of about 16mm/s.

B. METHODOLOGY OF DYNAMIC PAVEMENT AND PANTHERA SYNERGY

Over the past 20 years, various researches have been studying road width estimation, especially in autonomous driving vehicles. However, little or no research has been studying about a robot using a perception of dynamically changing pavement to perform reconfiguration in width during locomotion. This involves the integration between perspective algorithms and kinematic models of a reconfigurable robot. In the past,

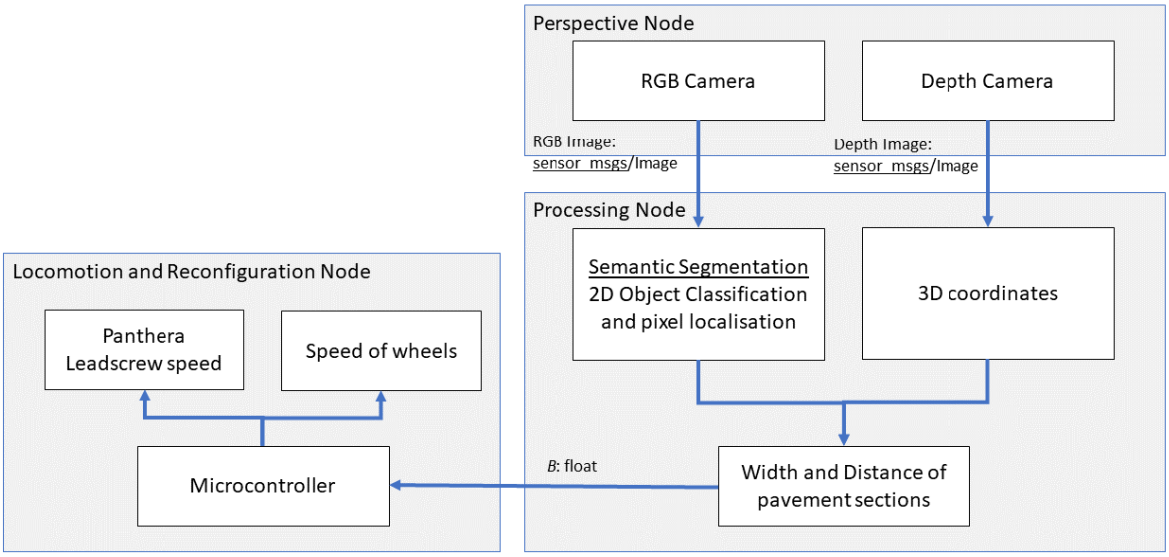


FIGURE 11. The block diagram of vision-based reconfigurable mechanism.

conventional feature-based methods such as Canny edge detection were useful in perspective algorithms such as detecting road edges for pavement width estimation. However, there are many issues about tuning the right parameters for different kinds of roads. For example, when a road does not have proper road signs or marking, canny edge detection does not work well. With the advances in technology, newer methods such as semantic segmentation using deep convolutional neural network enables object classification and pixel localization. Current state-of-the-art semantic segmentation methods are used in many autonomous cars today. Deep learning neural networks are able to train different types of pavements, even when pavements do not have enough markings, which is a challenging task for conventional feature detection algorithms. With deep learning models, classes are trained without hard coding to specify specific features in a class. The difficulty in using deep learning is the amount of time gathering quality data sets for training as well as the computational power required to train the deep learning model to be computed in real-time.

With the introduction of Deeplab and Segnet, object classification and pixel localization can be performed at a low cost and in real-time [38]. Other semantic segmentation includes Fast-SCNN [39] and FasterSeg [40] which are capable of performing high accuracy and low-cost computational power for semantic segmentation. Together with Panthera high specifications computer, Panthera uses the Robot Operating System (ROS) [36] framework, which provides the infrastructure for real-time communication. The ROS master node will be established on the on-board computer in Panthera and establishes channels for ROS clients. ROS nodes can subscribe and publish topics to other ROS nodes parallelly. Panthera ROS-based block diagram using the on-board computer as the master is as shown in Figure 11. The perspective node will publish sensor_msgs/Image type RGB and Depth images to the processing node. The processing node will subscribe sensor_msgs/Image RGB images from the perspective node and perform the semantic segmentation real-time to obtain the pavement and non-pavement classification and the localized pixels. To obtain more stable values of pavement width, the processing node will obtain the average of the minimum, x_{li}, and maximum x_{ri}, x pixel coordinate, and the corresponding y pixel coordinate from the sections in the image received as seen in Figure 12. After calculating the x and y pixel coordinates, the processing node will derive the corresponding depth values. As the RGB and Depth image comes from the same sensor RealSense D435 camera, the processing node subscribes sensor_msgs/Image RGB and Depth images parallelly, and as they share the same pixel mapping, the corresponding depth values can be obtained from the localized x and y pixel coordinates. It is assumed that the field of view of the D435 camera is parallel to the pavement that Panthera is moving in. The raw depth image has holes whereby the pixel values are noisy and not reliable. To reduce noise and inaccuracy, filters are added to improve the performance of the sensor [41]. An adaptive directional filter [42] is applied to the depth image to fill up holes, suppressing the depth images. After the depth image is filtered, the depth value of the pavement section is calculated using the following formula.

$$d_i = \frac{\sum_{\Omega_i} d_k}{\Omega_i} \tag{1}$$

where d_i is the depth for each section in the image, Ω_i is the number of pixels in section i, and d_k is the depth value per pixel. The formula takes the average depth value in the section where the pixel is classified as pavement. The depth for the section equals the summation of depth values

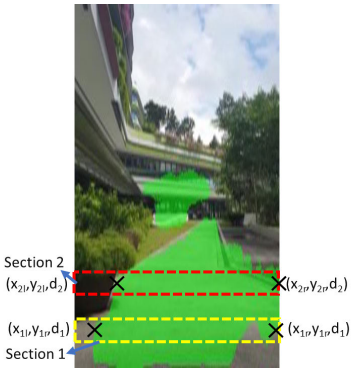


FIGURE 12. Width detection by sections semantic segmentation coordinates.

of pixel classified as pavement within the section over the total number of pixels classified as pavements. To get the estimated width of the pavement at each section, we need to convert it from pixel coordinates to real-world coordinates. Using the D435 camera intrinsic and extrinsic matrix, we are able to convert the obtained pixel coordinates into real-world coordinates. By applying the camera calibration method, which uses the camera to take many images of a chessboard, the camera intrinsic and extrinsic matrix can be obtained. Camera calibration helps us to find the focal length, f_x and f_y, and the optical centres c_x and c_y of the camera giving us the camera matrix, P. Camera calibration also helps us to find the rotational matrix, $\mathbf{R}$, and translation vector, t, and we can derive the extrinsic matrix, E, where E = [R T]. The x and y pixel coordinates, p, can be converted to real-world coordinates, W, using the following formula.

$$p = P * [RT] * W = P * E * W. \tag{2}$$

The width of the pavement in a section, w_i, can be calculated from the magnitude difference of the left and right side of the pavement pixel coordinates in Figure 12. Using the magnitude formula, $w_i = \sqrt{(x_{ri} - x_{li})^2 + (y_{ri} - y_{li})^2 + (d_{ri} - d_{li})^2}$. As the y coordinates and depth values are the same for the same section, the width of the pavement in a section is the difference between the x_{li} and x_{ri}. Upon calculating the pavement widths for the different sections in the image, w_1 and w_2, the difference in pavement widths, $\Delta w = w_1 - w_2$ and the difference in depth, $\Delta d = d_2 - d_1$ can be calculated. From the geometry seen in Figure 13, the β can be derived from Figure 14 is the image sectioned for pavement width estimation.

$$\beta = \tan^{-1} \frac{\Delta w}{2\Delta d} \tag{3}$$

This β will be passed on to the locomotion and reconfiguration node for Panthera to reconfigure during locomotion at the desired rate of reconfiguration. With β from the vision system, Panthera will derive the speed of steering units and the speed of lead screw motor for feedback control and enable Panthera to reconfigure smoothly concerning the

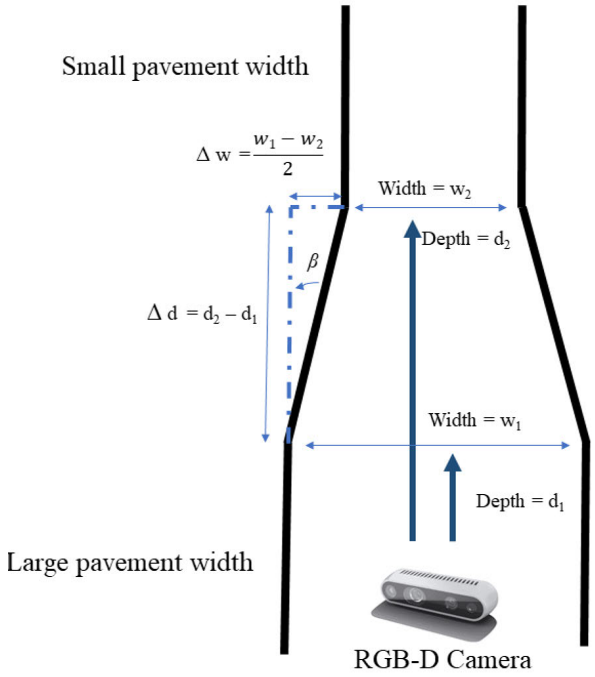


FIGURE 13. Section coordinates.

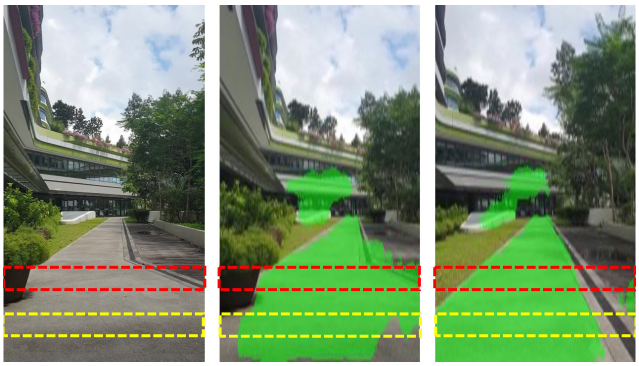


FIGURE 14. Image sectioned for pavement width estimation.

surrounding. The proposed algorithm is designed for even surface pavements and uniform gradient pavements. For pavements with varying terrain gradient, the proposed algorithm is less accurate as it assumes a uniform gradient between sections. However, this inaccuracy can be reduced by making the sections closer to each other. The kinematic locomotion model will be described in the next section.

IV. KINEMATIC MODEL FOR PANTHERA LOCOMOTION

A. WHEEL KINEMATICS

The position of the four steering units in the robot coordinate frame in Figure 7 is predefined as $(x^R_{SU1}, y^R_{SU1}) = (a, b)$; $(x^R_{SU2}, y^R_{SU2}) = (a, -b)$; $(x^R_{SU3}, y^R_{SU3}) = (-a, -b)$; $(x^R_{SU4}, y^R_{SU4}) = (-a, b)$; where a is 335mm and b ranges from 190mm (fully contracted) to 650mm (fully expanded). v and v_i are linear velocity of the robot and steering unit SU_i

respectively, where $v = \sqrt{{v_x}^2 + {v_y}^2}$, $v_i = \sqrt{{v_{xi}}^2 + {v_{yi}}^2}$, v_x and v_y are the robot velocity component along X_R and Y_R axis respectively, and v_{xi} and v_{yi} are the ith steering unit velocity component along X_{SUi} and Y_{SUi} axis respectively.

The conditions of no wheel slipping and sliding orthogonal to wheel planes are assumed. The velocity of the steering units and the robot are related as shown in the following equations:

$$v_{xi} = v_i \cos\delta_i = v_x - y^R_{SUi}\omega \tag{4}$$

$$v_{yi} = v_i \sin\delta_i = v_y - x^R_{SUi}\omega \tag{5}$$

where δ_i is the angle between v_i and X_{SUi}, x^R_{SUi} and y^R_{SUi} are the coordinates of ith steering units in the robot coordinate frame, ω is the angular velocity of the robot.

By substituting the coordinates of the four steering units into Eq. 4 and Eq. 5, the velocities of the steering units and the robot are related as

$$A\begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} = B\begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix}$$

$$A = \begin{bmatrix} 1 & 0 & -b \\ 0 & 1 & a \\ 1 & 0 & b \\ 0 & 1 & a \\ 1 & 0 & b \\ 0 & 1 & -a \\ 1 & 0 & -b \\ 0 & 1 & -a \end{bmatrix}$$

$$B = \begin{bmatrix} c\delta_1 & 0 & 0 & 0 \\ s\delta_1 & 0 & 0 & 0 \\ 0 & c\delta_2 & 0 & 0 \\ 0 & s\delta_2 & 0 & 0 \\ 0 & 0 & c\delta_3 & 0 \\ 0 & 0 & s\delta_3 & 0 \\ 0 & 0 & 0 & c\delta_4 \\ 0 & 0 & 0 & s\delta_4 \end{bmatrix} \tag{6}$$

The pseudo-inverse matrix of A is obtained as shown:

$$A^+ = \begin{bmatrix} 1/4 & 0 & 1/4 & 0 & 1/4 & 0 & 1/4 & 0 \\ 0 & 1/4 & 0 & 1/4 & 0 & 1/4 & 0 & 1/4 \\ -b/k & a/k & b/k & a/k & b/k & -a/k & -b/k & -a/k \end{bmatrix}$$

where $k = 4a^2 + 4b^2$. Multiplying A^+ to Eq. 6, the velocities of the robot is calculated as shown:

$$\begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} = \begin{bmatrix} c\delta_1/4 & c\delta_2/4 & c\delta_3/4 & c\delta_4/4 \\ s\delta_1/4 & s\delta_2/4 & s\delta_3/4 & s\delta_4/4 \\ P_1 & P_2 & P_3 & P_4 \end{bmatrix}\begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} \tag{7}$$

where $c\delta_i$ and $s\delta_i$ represents cosine and sine of δ_i respectively and $P_i = (-y^R_{SUi}c\delta_i + x^R_{SUi}s\delta_i)/k$

Including the steering angle of the steering units, the state vector of the robot representing the robot position and orientation in the global coordinate frame is given to be $q = \begin{bmatrix} x & y & \theta & \delta_1 & \delta_2 & \delta_3 & \delta_4 \end{bmatrix}^T$

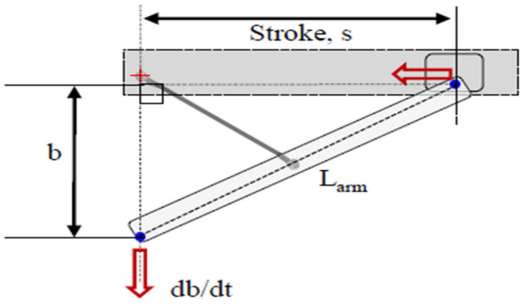


FIGURE 15. Panthera mechanical constrains.

The kinematic model is then represented as follows.

$$\dot{q} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\delta_1} \\ \dot{\delta_2} \\ \dot{\delta_3} \\ \dot{\delta_4} \end{bmatrix} = \begin{bmatrix} \frac{c1}{4} & \frac{c2}{4} & \frac{c3}{4} & \frac{c4}{4} & 0 & 0 & 0 & 0 \\ \frac{s1}{4} & \frac{s2}{4} & \frac{s3}{4} & \frac{s4}{4} & 0 & 0 & 0 & 0 \\ P_1 & P_2 & P_3 & P_4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix} \tag{8}$$

where $\dot{q}.$ is defined as the velocity vector in the world coordinate, ci and si are represented as $cos(\delta_i + \theta)$ and $sin(\delta_i + \theta)$, respectively.

Each steering unit is modeled as a differential drive robot, as illustrated in Figure 8. As the linear velocity of the right and left wheels vary, the steering unit will rotate and dynamically change its steering angle depending on the needs of locomotion and reconfiguration.

Together with the lead screw motor, Panthera is capable of performing reconfiguration in width during locomotion. Reconfiguration requires the lead screw motor speed differential driving unit speed and steering angle to be synchronized. The locomotion and reconfiguration node will receive the pavement width estimates for the different sections from the processing node and will calculate the required lead screw motor speed, differential drive unit steering angle based on the Panthera locomotion. The vision system will provide the steering angle and will derive the speed of differential drive steering units required for the system to be synchronized. This allows Panthera to reconfigure during locomotion at the rate required by the dynamic pavements and Panthera's current speed.

B. LEAD SCREW KINEMATICS

For the lead screw motor speed, differential drive unit speed and steering angle to be synchronized, it must obey the following mechanical constrain in Figure 15 due to Panthera's frame where $s = \sqrt{L^2_{arm} - b^2}$, where $\dot{s}$ is the velocity of the carriage and b is the distance between the side shaft and the central beam. There are three gaits for movement during reconfiguration, as seen in Figure 16. In the case of A,

the differential steering units on the right beam move towards the central beam while the left beam stays static concerning the robot's inertial frame. In the case of B, the differential steering units on the left beam move towards the central beam while the right beam stays static with respect to the robot's inertial frame. In the case of C, both side beams move towards the central beam, and the central beam stays static in the robot's inertial frame.

1) CASE A AND B

In Case-A and Case-B, one of the sides of the robot is moving is the straight line while the other side is closing into the centre of the robot. In Case-A, (SU_1, SU_4) will have zero steering angle $\delta_1 = \delta_4 = 0$ while the 2 right steering units (SU_2, SU_3) have $-\pi/2 < \delta_2 = \delta_3 < 0$. In Case-B, the right wheel move straight where $0 < \delta_1 = \delta_4 < \pi/2$ while $\delta_2 = \delta_3 = 0$. The speeds of the steering units moving in a straight line, V_j is

$$\begin{aligned} v_{yj} &= v_j sin(|\delta_i|) \\ &= \frac{v_{rj} + v_{lj}}{2} sin(|\delta_j|) \\ &= \frac{r\varphi_{rj} + r\varphi_{lj}}{2} sin(|\delta_j|) \\ &= \dot{b}_{MR} \end{aligned} \tag{9}$$

$$\begin{aligned} v_{xj} &= v_j cos(|\delta_i|) \\ &= \frac{v_{rj} + v_{lj}}{2} cos(|\delta_j|) \\ &= \frac{r\varphi_{rj} + \varphi_{lj}}{2} cos(\delta_j|) \\ &= \dot{b}_{MR} \end{aligned} \tag{10}$$

and moving into the robot V_k are as follows

$$v_{xk} = v_j cos(|\delta_i|) = [(v_{rj} + v_{lj})/2] cos(|\delta_j|) \tag{11}$$

$$v_{yk} = 0 \tag{12}$$

The distances of the steering units have to travel in the y-direction to compress twice in Case-A and Case-B as compared to in Case-C. The speed of the reconfiguration motor is as follows.

$$\begin{aligned} \dot{s}_{Case-A} &= \dot{s}_{Case-B} \\ &= \frac{\dot{s}_{Case-C}}{2} \\ &= \frac{-0.5\dot{b}_{MR} b_{MR}}{\sqrt{L_{arm}^2 - \dot{b}_{MR}^2}} \end{aligned} \tag{13}$$

2) CASE C

In Case-C, the central frame stationary gait, the robot is moving is a straight line while compressing and the robot to maintain its centroid along a straight, $0 < \delta_1 = \delta_4 < \pi/2$ while $-\pi/2 < \delta_2 = \delta_3 < 0$ and $\delta_1 = \delta_4 = -\delta_2 = -\delta_3$. δ_1, δ_2, δ_3 and δ_4 and the x and y component of the velocity are as follows

$$V_{yi} = V_i sin(|\delta_i|) \tag{14}$$

$$V_{xi} = V_i sin(\pi - |\delta_i|) \quad for\ i = 1, 2, 3, 4 \tag{15}$$

As V_y of the moving steering units must match the speed at which the width b changes $\dot{b}$, the speed at which the width b changes for moving while reconfiguration is as follows

$$\begin{aligned} v_{yi} &= v_i sin(|\delta_i|) \\ &= \frac{v_{ri} + v_{li}}{2} sin(|\delta_i|) \\ &= \frac{r\varphi_r i + r\varphi_{li}}{2} sin(|\delta_i|) \\ &= \dot{b}_{MR} \end{aligned} \tag{16}$$

The speed of the rotation of the reconfiguration while compressing to the center is

$$\dot{s}_{Case-C} = \frac{-\dot{b}_{MR} b_{MR}}{\sqrt{L_{arm}^2 - \dot{b}_{MR}^2}} \tag{17}$$

Using the same formula for Case A, B, and C, Panthera is also able to perform static reconfiguration where $v_{xi} = 0$. Similar to compression, Panthera is also capable of performing expansion. In the next section, we will look at the Panthera perspective algorithm to enable Panthera to reconfigure based on dynamically changing pavements. The vision system will provide the steering angle and will derive the speed of lead screw motor speed required for the system to be synchronized.

V. DYNAMIC PAVEMENT RECONFIGURATION EXPERIMENTAL RESULTS

To evaluate the ability to self-reconfigure due to changing pavement widths, an experimental environment where the pavement widths are dynamically changing. Panthera is set to operate positioned at the center of the pavements and parallel to the pavement, such as in Figure 17. It is assumed that there are no obstacles in the pavement and Panthera will perform Case C of the reconfiguration during locomotion to adapt to both sides of the dynamically changing pavement. Three experiments were conducted. In the first experiment, Panthera will start in a dynamic pavement width environment and move straight in the forward direction. The experiment ends when it reaches pavement of constant width. In the second experiment, Panthera will start in a pavement of constant width. It will move straight in the forward direction and move out of the pavement of constant width into a pavement of dynamically changing width. In the third and last experiment, Panthera will move straight along a pavement of constant width. From the three experiments, the values of β, lead screw motor speed, steering unit speed, and the width of the robot will be plotted against time for reconfiguration during locomotion, such as in Figure 18.

As the robot moves along the pavement, semantic segmentation will be applied to the RGB stream from the RealSense D435 camera sensor. In the experiment, the semantic segmentation VGG-16 model is applied to the RGB images. After using semantic segmentation to classify and localize the pixels, the average image pixel coordinates of the edges of the pavement width will be extracted from two sections

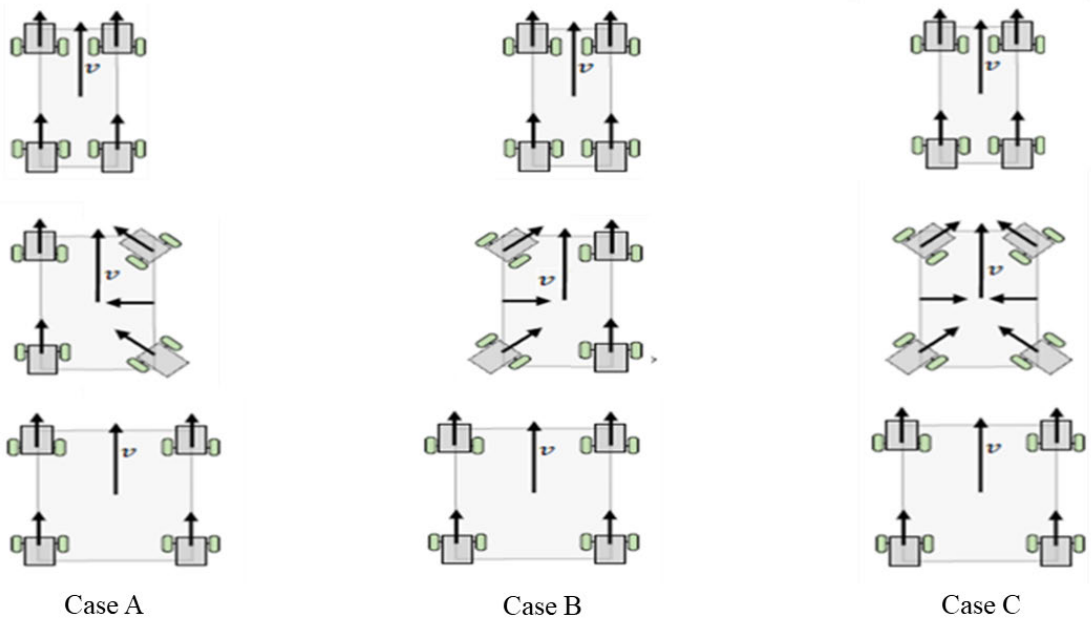


FIGURE 16. Panthera gaits for reconfiguration during locomotion.

FIGURE 17. Panthera platform. a) minimum width, b) maximum width.

of the images. Together with the depth information from the depth sensor, the pixel coordinates are converted to real-world coordinates using the homogeneous equation in 2.

A. PERFORMANCE OF VISION SYSTEM

A short-range distance of pavement widths was detected for width estimation. From Figure 19, the width estimates of sections 1 and 2 are 2.16m and 2.25m, respectively, while the ground truth of pavement width is 2.09m. For long-range distance where the depth values increase, the pavement estimates are less accurate, where it will estimate up to a value of 2.7m. This is because the RealSense D435 camera only has an operating range of about 10m, and the values are more affected by noise. After getting the world coordinates of the pavement with edges in the two sections, and by using equations in 3, the β values can be derived.

The filtered β derived will be used to pass on to the robot's kinematic model for reconfiguration to the center during moving. To perform reconfiguration during motion smoothly, the β and the kinematic model must be synchronized so that the robot can continue moving at the desired speed and, at the same time, reconfigure in shape. The values of β will determine the steering angles of the steering units (SU_1, SU_2, SU_3, SU_4) of the robot where $\delta_1 = \beta$, $\delta_2 = \beta$, $\delta_3 = -\beta$ and $\delta_4 = -\beta$. With an initial robot state of fully close and speed of 0.1m/s, the required speed of the steering units and the lead screw reconfiguration motors are derived. The robot's width is also presented to show how the robot reconfigures in width as it moves. The filtered β derived will be used to pass on to the kinematic model of the robot for reconfiguration to the center during moving.

B. FIRST EXPERIMENT

In the first experiment, it is performed in a dynamically changing pavement width where the pavement changes in width and then moves into the pavement with constant pavement width, as seen in Figure 20. At the start of the experiment, the robot is far from the pavement with a constant width. However, as the robot approaches the pavement with constant width, the beta slowly drops in value until zero when it finally reaches the pavement of constant width.

Based on the filtered β values, the lead screw speed of the robot in Figure 21 and the speed for each steering unit are as shown in Figure 22. Panthera will use these values as input to the respective motors and will be controlled via a proportional integral differential (PID) control. Using the PID control of Panthera of the motors speed and steering angle of the steering units, the control logic minimizes the error quotient and auto-tune the PID values for the better drivability of the Panthera, providing the desired lead screw speed, steering unit speed, and steering units heading angle

FIGURE 18. Reconfiguration in width during locomotion.

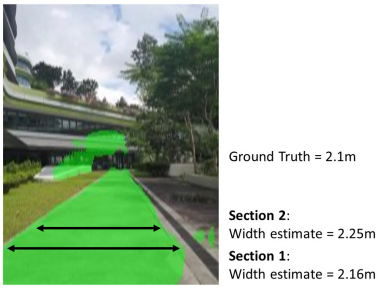


FIGURE 19. Pavement width estimate and ground truth.

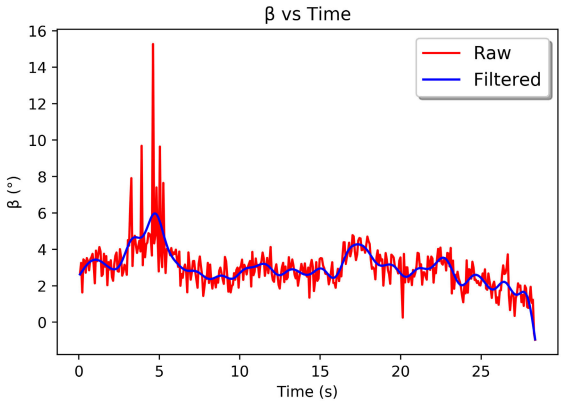


FIGURE 20. β against time.

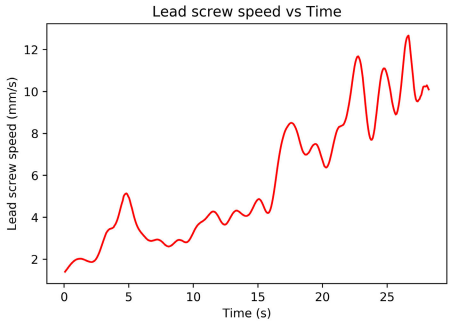


FIGURE 21. Lead screw motor speed against time.

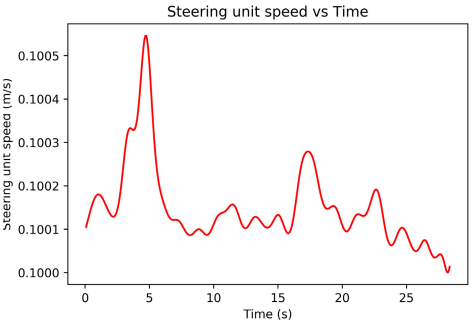


FIGURE 22. Steering unit speed against time.

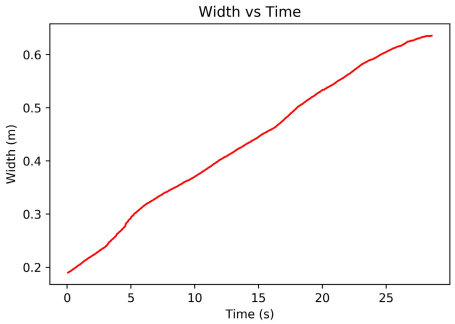


FIGURE 23. Width against time.

required to perform the reconfiguration during locomotion. In the process of reconfiguration, while moving, the width of the robot based on feedback control from vision is shown in Figure 23.

As the initial robot state is fully closed, the initial width is 0.19m. The width in Figure 23 is derived from the filtered beta angle and the steering unit speed using trigonometry. The result of using the proposed algorithm gives the smooth adaptation of Panthera width as it moves through dynamically changing pavement widths.

C. SECOND EXPERIMENT

In the second experiment, it is performed just before the robot leaves a pavement of constant pavement width into an open area with dynamically changing pavement widths, as seen in Figure 24. At the start of the experiment, the robot is still in the pavement of constant pavement width, and the value of β is around zero. However, as it was leaving the pavement of constant width, beta slowly increases from zero. After it exits

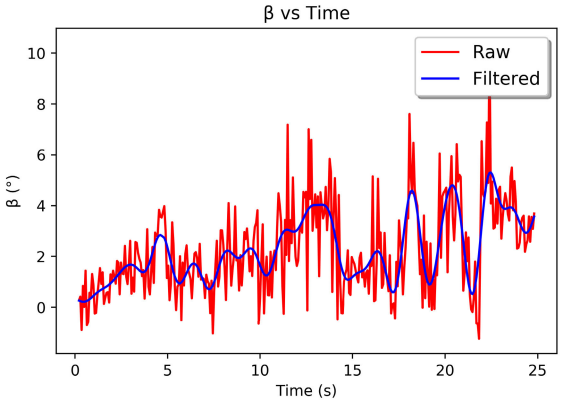


FIGURE 24. Width against time.

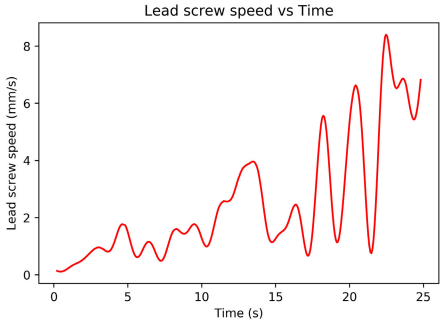


FIGURE 25. Lead screw speed against time.

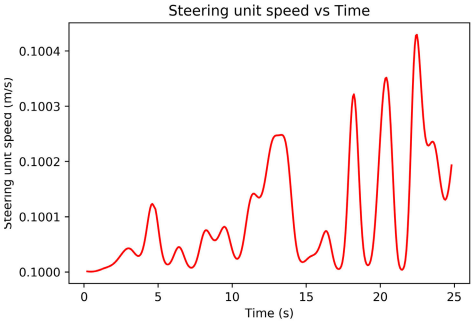


FIGURE 26. Steering unit speed against time.

from the pavement fully and goes into a more open pavement, the robot continues to detect changes in pavement width, and the value of β continues to increase.

Based on the filtered β values, the lead screw speed of the robot in Figure 25 and the speed for each steering unit Figure 26 are as shown in this figure. Panthera will use these values as input to the respective motors and will be controlled via a proportional integral differential (PID) control. Using the PID control of Panthera of the motors speed and steering angle of the steering units, the control logic minimize the error quotient and auto-tune the PID values for the better drivability of the Panthera, providing the desired lead screw speed, steering unit speed, and steering units heading angle required to perform the reconfiguration during locomotion.

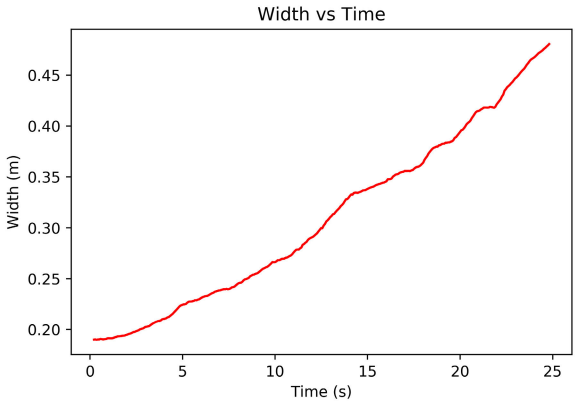


FIGURE 27. Width against time.

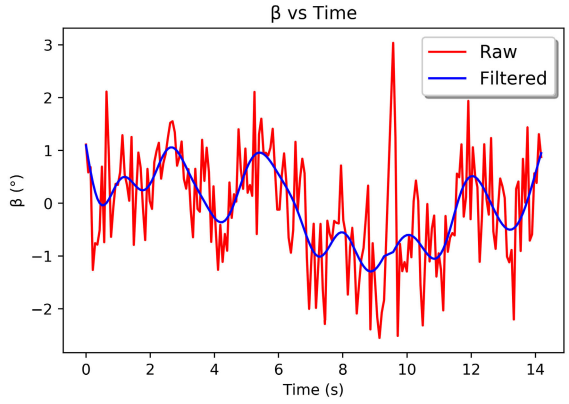


FIGURE 28. β against time.

In the process of reconfiguration, while moving, the width of the robot based on feedback control from vision is shown in the Figure 27.

As the initial robot state is fully closed, the initial width is 0.19m. The width in Figure 27 is derived from the filtered beta angle and the steering unit speed using trigonometry. The result of using the proposed algorithm gives the smooth adaptation of Panthera width as it moves through dynamically changing pavement widths.

D. THIRD EXPERIMENT

In the last experiment seen in Figure 28, it is performed on a pavement with a constant pavement width. At the start of the experiment, the robot is still in the pavement of constant pavement width, and the value of β is around zero. However, as it moves along the pavement, it continues detecting for pavement width changes. The value of β fluctuates around zero throughout the experiment. This is due to noise in data values from the RGB-D camera and also the noise in the semantic segmentation.

Based on the filtered β values, the lead screw speed of the robot Figure 29 and the speed for each steering unit Figure 30 are as shown in this figure. As these are values derived from the filtered β, the lead screw motor speed and the steering unit

FIGURE 29. Lead screw speed against time.

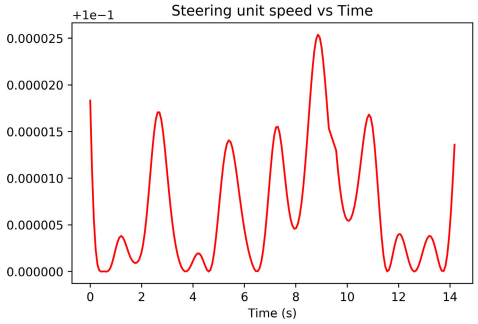


FIGURE 30. Steering unit speed against time.

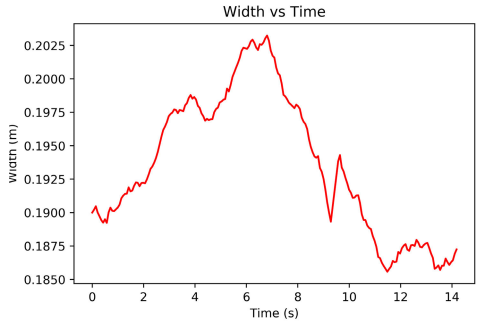


FIGURE 31. Width against time.

speed are oscillating with a small magnitude around 0 and 0.1, respectively.

As the initial robot state is fully closed, the initial width is 0.19m. The robot's width is derived from the filtered β angle and the steering unit speed using trigonometry. The result of using the proposed algorithm seen in Figure 31 shows that the width of the robot based on feedback control from vision varies only by a small magnitude around its initial robot state and is generally constant.

These three experiments show that the proposed algorithm can enable pavement sweeping robots to understand its surroundings and to provide information about the surroundings to allowing reconfigurable robots to reconfigure in width smoothly.

TABLE 2. Terminologies and definition.

Symbol	Definition
$p_{leadscrew}$	Pitch of lead screw
Z_{SU_i}	Steering axis of the robot: Directed out of paper
X_{SU_i}	Steering axis of the robot: Directed upwards of paper
Y_{SU_i}	Steering axis of the robot: Directed left of paper
w_i	Pavement width in section i
w	Pavement width
d_i	Depth value of section i
d_k	Depth value per pixel
Ω_i	Number of pixels in section i
f_x	Focal lenght of camera in pixel x axis
f_y	Focal lenght of camera in pixel y axis
c_x	Optical center in pixel x axis
c_y	Optical center in pixel y axis
P	Camera matrix
R	Rotational Vector
T	Translation Vector
E	Camera extrinsic matrix
W	World coordinate
p	Pixel coordinate
c_x	Optical center in pixel x axis
c_y	Optical center in pixel y axis
x_{li}	Average of minimum x coordinate in section i
x_{ri}	Average of maximum x coordinate in section i
y_{li}	Average of minimum x coordinate in section i
y_{ri}	Average of maximum x coordinate in section i
d_{li}	Depth of minimum x coordinate in section i
d_{ri}	Depth of maximum x coordinate in section i
Δw	Width in section 1 minus width in section 2
Δd	Depth in section 1 minus depth in section 2
d_{li}	Depth of minimum x coordinate in section i
d_{ri}	Depth of the maximum x coordinate in section i
β	Parameter used to derive required steering angle δ
v	Velocity of robot
v_x	X component of velocity robot
v_y	Y component of velocity robot
v_i	Velocity of steering unit i
v_{xi}	x component velocity of steering unit i
v_{yi}	y component velocity of steering unit i
$x^R_{SU_i}$	X coordinates of ith steering units in robot frame
$y^R_{SU_i}$	Y coordinates of ith steering units in robot frame
δ_i	Steering angle of SU_i: Angle between v$_i$andX_SUi
r	Radius of wheel = 108mm
b	Distance between center and side beam
$\dot{b}$	Rate of distance change between center and side beam
$\dot{s}$	The velocity of the carriage
$\dot{s}_{Case-A}$	velocity of carriage for Case-A
$\dot{s}_{Case-B}$	velocity of carriage for Case-B
$\dot{s}_{Case-C}$	velocity of carriage for Case-C
b_{MR}	Distance between the center and the side beam
$\dot{b}_{MR}$	Rate of change of distance between center and side beam
L_{arm}	Length of linkage bar
ω	Angular velocity of the robot
V_j	Speeds of the steering units
v_{rj}	Right wheel speeds of the steering units
v_{lj}	Left wheel speeds of the steering units
$\dot{\varphi}_{rj}$	Right wheel angular velocity of steering units
$\dot{\varphi}_{lj}$	Left wheel angular velocity of steering units
V_k	Speed of the steering units during moving reconfiguration
v_{xk}	X component speeds of the steering units
v_{yk}	Y component speeds of the steering units
v_{rk}	Right wheel speeds of the steering units
v_{lk}	Left wheel component speeds of the steering units
$\dot{\varphi}_{rk}$	Right wheel angular velocity of steering units
$\dot{\varphi}_{lk}$	Left wheel angular velocity of steering units

VI. CONCLUSIONS AND FUTURE WORK

The interaction framework between a novel reconfigurable pavement cleaning robot and dynamic changing pavement width is studied in this research. The ability of the platform to reconfigure the width allows it to operate smoothly with the change in pavement width. The model detects the pavement width using the mask based DCNN network in combination

with the depth image, allowing the robot to gain information about the dynamic pavement environment. This helps the robot to change the width accordingly to save energy and optimize the cleaning task. The future work will focus on reconfiguration of Panthera width based on dynamically changing pavement width and objects, pedestrians [14], and obstacles along the pavement and the adaptive control mechanism for dynamic environment path planning algorithms for the autonomous navigation to maximizing the pavement coverage and minimizing energy consumption. This future work will involve the reconfiguration during locomotion Case A and Case B where the robot will adapt to the dynamically changing pavement on one side to avoid an obstacle on the other side of the pavement. Another future work will also include the efficiency of the current vision approach on different pavement types and gradients. In the future with the more stable working platform including cleaning payload, the numerical evaluations about saves energy and optimize the cleaning task through sweeping will be conducted. The dynamic changing pavement width vision feedback and the robot's corresponding kinematic model are reported in the literature, which can also be part of the future work on interaction based on intentions for movement.

APPENDIX

See Table 2.

REFERENCES

[1] A. Le, P.-C. Ku, T. T. Tun, N. H. K. Nhan, Y. Shi, and R. Mohan, "Realization energy optimization of complete path planning in differential drive based self-reconfigurable floor cleaning robot," *Energies*, vol. 12, no. 6, p. 1136, Mar. 2019.

[2] A. V. Le, N. H. K. Nhan, and R. E. Mohan, "Evolutionary algorithm-based complete coverage path planning for tetriamond tiling robots," *Sensors*, vol. 20, no. 2, p. 445, Jan. 2020.

[3] A. Le, V. Prabakaran, V. Sivanantham, and R. Mohan, "Modified a-star algorithm for efficient coverage path planning in tetris inspired self-reconfigurable robot with integrated laser sensor," *Sensors*, vol. 18, no. 8, p. 2585, Aug. 2018.

[4] B. Ramalingam, A. Lakshmanan, M. Ilyas, A. Le, and M. Elara, "Cascaded machine-learning technique for debris classification in floor-cleaning robot application," *Appl. Sci.*, vol. 8, no. 12, p. 2649, Dec. 2018.

[5] A. V. Le, A. A. Hayat, M. R. Elara, N. H. K. Nhan, and K. Prathap, "Reconfigurable pavement sweeping robot and pedestrian cohabitant framework by vision techniques," *IEEE Access*, vol. 7, pp. 159402–159414, 2019.

[6] M. A. V. J. Muthugala, A. V. Le, E. S. Cruz, M. R. Elara, P. Veerajagadheswar, and M. Kumar, "A self-organizing fuzzy logic classifier for benchmarking robot-aided blasting of ship hulls," *Sensors*, vol. 20, no. 11, p. 3215, Jun. 2020.

[7] A. K. Lakshmanan, R. E. Mohan, B. Ramalingam, A. V. Le, P. Veerajagadeshwar, K. Tiwari, and M. Ilyas, "Complete coverage path planning using reinforcement learning for tetromino based cleaning and maintenance robot," *Autom. Construct.*, vol. 112, Apr. 2020, Art. no. 103078.

[8] N. Tan, N. Rojas, R. E. Mohan, V. Kee, and R. Sosa, "Nested reconfigurable robots: Theory, design, and realization," *Int. J. Adv. Robot. Syst.*, vol. 12, no. 7, p. 110, Jul. 2015.

[9] A. V. Le, R. Parween, R. E. Mohan, N. H. K. Nhan, and R. E. Abdulkader, "Optimization complete area coverage by reconfigurable hTrihex tiling robot," *Sensors*, vol. 20, no. 11, p. 3170, Jun. 2020.

[10] M. Kouzehgar, Y. K. Tamilselvam, M. V. Heredia, and M. R. Elara, "Self-reconfigurable façade-cleaning robot equipped with deep-learning-based crack detection based on convolutional neural networks," *Autom. Construct.*, vol. 108, Dec. 2019, Art. no. 102959. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0926580519301256

[11] M. Ilyas, S. Yuyao, R. E. Mohan, M. Devarassu, and M. Kalimuthu, "Design of sTetro: A modular, reconfigurable, and autonomous staircase cleaning robot," *J. Sensors*, vol. 2018, pp. 1–16, Jul. 2018.

[12] N. Tan, R. E. Mohan, and K. Elangovan, "Scorpio: A biomimetic reconfigurable rolling–crawling robot," *Int. J. Adv. Robot. Syst.*, vol. 13, no. 5, pp. 1–16, 2016, doi: 10.1177/1729881416658180.

[13] A. A. Hayat, K. Elangovan, M. R. Elara, and M. S. Teja, "Tarantula: Design, modeling, and kinematic identification of a quadruped wheeled robot," *Appl. Sci.*, vol. 9, no. 1, p. 94, Dec. 2018.

[14] A. A. Hayat, R. Parween, M. R. Elara, K. Parsuraman, and P. S. Kandasamy, "Panthera: Design of a reconfigurable pavement sweeping robot," in *Proc. Int. Conf. Robot. Automat. (ICRA)*, May 2019, pp. 7346–7352.

[15] *Green Machines Compact Air Sweeper*. Accessed: Oct. 28, 2019. [Online]. Available: https://tcs-slovakia.sk/wp-content/uploads/2017/11/tennant-636-brochure-en.pdf

[16] *Supersteam*. Accessed: Oct. 28, 2019. [Online]. Available: http://www.supersteam.com.sg/products/cleaning-machines/dream-robotic-suction-sweeper#undefined1

[17] *Johnston Sweepers*. Accessed: Oct. 28, 2019. [Online]. Available: https://www.johnstonsweepers.com/wp-content/uploads/2017/12/cn101-brochure-78115.pdf

[18] S. K. Suman and V. K. Giri, "Speed control of DC motor using optimization techniques based PID controller," in *Proc. IEEE Int. Conf. Eng. Technol. (ICETECH)*, Mar. 2016, pp. 581–587.

[19] K. Sharma and D. K. Palwalia, "A modified PID control with adaptive fuzzy controller applied to DC motor," in *Proc. Int. Conf. Inf., Commun., Instrum. Control (ICICIC)*, Aug. 2017, pp. 1–6.

[20] W.-J. Tang, Z.-T. Liu, and Q. Wang, "DC motor speed control based on system identification and PID auto tuning," in *Proc. 36th Chin. Control Conf. (CCC)*, Jul. 2017, pp. 6420–6423.

[21] J. Duan and A. Valentyna, "Road edge detection based on LiDAR laser," in *Proc. Int. Conf. Control, Automat. Inf. Sci. (ICCAIS)*, Oct. 2015, pp. 137–142.

[22] L. Ding and A. Goshtasby, "On the Canny edge detector," *Pattern Recognit.*, vol. 34, no. 3, pp. 721–725, Mar. 2001. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0031320300000236

[23] Y. M. Wang, Y. Li, and J. B. Zheng, "A camera calibration technique based on OpenCV," in *Proc. 3rd Int. Conf. Inf. Sci. Interact. Sci.*, Jun. 2010, pp. 403–406.

[24] A. V. Le and J. Choi, "Robust tracking occluded human in group by perception sensors network system," *J. Intell. Robot. Syst.*, vol. 90, nos. 3–4, pp. 349–361, Jun. 2018.

[25] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.

[26] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.

[27] T. Kobayashi, "BFO meets HOG: Feature extraction based on histograms of oriented p.d.f. gradients for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 747–754.

[28] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, "Deep learning for computer vision: A brief review," *Comput. Intell. Neurosci.*, vol. 2018, pp. 1–13, Feb. 2018.

[29] A. Gottlieb and G. Almasi, *Highly Parallel Computing*. Redwood City, CA, USA: Benjamin Cummings, 1989.

[30] S. Choi and K. Lee, "A CUDA-based implementation of convolutional neural network," in *Proc. 4th Int. Conf. Comput. Appl. Inf. Process. Technol. (CAIPT)*, Aug. 2017, pp. 1–4.

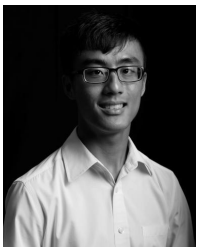
[31] S. Chetlur, C. Woolley, P. Vandermersch, J. Cohen, J. Tran, B. Catanzaro, and E. Shelhamer, "CuDNN: Efficient primitives for deep learning," 2014, *arXiv:1410.0759*. [Online]. Available: http://arxiv.org/abs/1410.0759

[32] Z. Du, L. He, Y. Chen, Y. Xiao, P. Gao, and T. Wang, "Robot cloud: Bridging the power of robotics and cloud computing," *Future Gener. Comput. Syst.*, vol. 74, pp. 337–348, Sep. 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X16000042

[33] B. S. Đorđević, S. P. Jovanović, and V. V. Timčenko, "Cloud computing in Amazon and microsoft Azure platforms: Performance and service comparison," in *Proc. 22nd Telecommun. Forum Telfor (TELFOR)*, Nov. 2014, pp. 931–934.

[34] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.

[35] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, Atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.
[36] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: An open-source robot operating system," in *Proc. ICRA Workshop Open Source Softw.*, Kobe, Japan, 2009, vol. 3, no. 3, p. 5.
[37] M. R. Elara, N. Rojas, and A. Chua, "Design principles for robot inclusive spaces: A case study with roomba," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2014, pp. 5593–5599.
[38] V. Badrinarayanan, A. Handa, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling," 2015, *arXiv:1505.07293*. [Online]. Available: http://arxiv.org/abs/1505.07293
[39] R. P. K. Poudel, S. Liwicki, and R. Cipolla, "Fast-SCNN: Fast semantic segmentation network," 2019, *arXiv:1902.04502*. [Online]. Available: https://arxiv.org/abs/1902.04502
[40] W. Chen, X. Gong, X. Liu, Q. Zhang, Y. Li, and Z. Wang, "FasterSeg: Searching for faster real-time semantic segmentation," 2019, *arXiv:1912.10917*. [Online]. Available: https://arxiv.org/abs/1912.10917
[41] Q. Lim, Y. He, and U.-X. Tan, "Real-time forward collision warning system using nested Kalman filter for monocular camera," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Dec. 2018, pp. 868–873.
[42] D. C. Hernández and K.-H. Jo, "Stairway tracking based on automatic target selection using directional filters," in *Proc. 17th Korea-Japan Joint Workshop Frontiers Comput. Vis. (FCV)*, 2011, pp. 1–6.



LIM YI received the B.Eng. degree from the Engineering Product Development (EPD) Pillar, Singapore University of Technology and Design (SUTD), in 2017. He is currently pursuing the master's degree in engineering (research) with SUTD. He is also working with the Robotics and Automation Research (ROAR) Laboratory, SUTD. He worked at Accenture as a Business and Technology Integrator Analyst and a Software Engineer liaising with many different organisations to work on new and emerging technologies. His work experiences include system architecture, data science, and databases. His current research interests include robotics, robotics control, robotics perception, robotics vision, and robotics navigation.

ANH VU LE received the B.S. degree in electronics and telecommunications from the Hanoi University of Technology, Vietnam, in 2007, and the Ph.D. degree in electronics and electrical from Dongguk University, South Korea, in 2015. He is currently with the Optoelectronics Research Group, Faculty of Electrical and Electronics Engineering, Ton Duc Thang University, Ho Chi Minh City, Vietnam. He is also working as a Postdoctoral Research Fellow with the ROAR Laboratory, Singapore University of Technology and Design. His current research interests include robotics vision, robot navigation, human detection, action recognition, feature matching, and 3-D video processing.

ABDULLAH AAMIR HAYAT received the B.Tech. degree in mechanical engineering from the Zakir Hussain College of Engineering and Technology (ZHCET), AMU, Aligarh, in 2009, the M.Tech. degree, in 2011, and the Ph.D. degree from IIT Delhi, India. He also worked as a Junior Research Fellow with the Programme for Autonomous Robotics Laboratory, IIT Delhi. He has been a Postdoctoral Research Fellow with the Singapore University of Technology and Design, Singapore. His research interests include kinematic identification, calibration, multibody dynamics, and reconfigurable robotics.

CHARAN SATYA CHANDRA SAIRAM BORUSU (Member, IEEE) received the B.Tech. degree in electronics and communication engineering from Amrita Vishwa Vidyapeetham, in 2019. He is currently working as a Visiting Research Fellow with the ROAR Laboratory, Singapore University of Technology and Design, Singapore. He worked at Robert Bosch, Coimbatore, as an Associate Engineer. He also worked as a Research Assistant with the Humanitarian Technology (HuT) Laboratory, a robotics research laboratory at the Amrita School of Engineering, Amritapuri. His research interests include robotics and VLSI.

RAJESH ELARA MOHAN received the B.E. degree from Bharathiar University, India, in 2003, and the M.Sc. and Ph.D. degrees from Nanyang Technological University, in 2005 and 2012, respectively. He is currently an Assistant Professor with the Engineering Product Development Pillar, Singapore University of Technology and Design. He is also a Visiting Faculty Member with the International Design Institute, Zhejiang University, China. He has published over 80 articles in leading journals, books, and conferences. His research interest includes robotics with an emphasis on self-reconfigurable platforms as well as research problems related to robot ergonomics and autonomous systems. He was a recipient of the SG Mark Design Award, in 2016 and 2017; the ASEE Best of Design in Engineering Award, in 2012; and the Tan Kah Kee Young Inventors' Award, in 2010.

NGUYEN HUU KHANH NHAN defended his Ph.D. thesis at the Institute of Research and Experiments for Electrical and Electronic Equipment, Moscow, Russia. He is working as a Lecturer with the Faculty of Electrical and Electronics Engineering, Ton Duc Thang University, Ho Chi Minh City, Vietnam. His research interests include VLSI, MEMS and LED driver chips, robotics vision, robot navigation, and 3-D video processing.

PRATHAP KANDASAMY received the B.Sc. degree in computer engineering from the Singapore University of Technology and Design, in 2010. He is currently working as a Research Officer with the Engineering Product Development Pillar, Singapore University of Technology and Design. His research interest includes mechanical designs.