

Research Article

Design and Steering Control of a Center-Articulated Mobile Robot Module

Mehdi Delrobaei and Kenneth A. McIsaac

Department of Electrical and Computer Engineering, The University of Western Ontario, London, ON, Canada N6A 5B9

Correspondence should be addressed to Mehdi Delrobaei, mehdidelrobaei@gmail.com

Received 3 July 2011; Accepted 3 November 2011

Academic Editor: Yangmin Li

Copyright © 2011 M. Delrobaei and K. A. McIsaac. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper discusses the design and steering control for an autonomous modular mobile robot. The module is designed with a center-articulated steering joint to minimize the number of actuators used in the chain. We propose a feedback control law which allows steering between configurations in the plane and show its application as a parking control to dock modules together. The control law is designed by Lyapunov techniques and relies on the equations of the robot in polar coordinates. A set of experiments have been carried out to show the performance of the proposed approach. The design is intended to endow individual wheeled modules with the capability to merge and make a single snake-like robot to take advantage of the benefits of modular robotics.

1. Motivation

The capability of moving through a wide variety of remote areas has made *mobile robots* an interesting topic of research. But since there are different ways to move, the locomotion method selection is a challenging aspect of mobile robots design.

Inspired by their biological counterparts, mobile robots can walk, slide, and swim. In addition, conventional mobile robots travel using powered wheels. Generally, wheels have high efficiency; they are simple, and well suited to flat lands. But, the performance of wheeled mobile robots becomes seriously critical in exploring untraditional environments. For instance, wheeled mobile robots are inappropriate to move over rough terrain, sands, or water.

Chain-based robots tend to become increasingly an alternative to wheeled robots in robotic applications. While both legged and wheeled robots are unable to effectively enter narrow spaces or climb over obstacles, snake robots with many degrees of freedom can cross a narrow gap, climb over a rock, move over rough terrain or marshland, and even swim. However, snake-like robot locomotion is not efficient or appropriate where traditional wheeled systems can be used.

This contrast motivated us to study the transition between wheeled and modular robotics and present autonomous mobile robot modules capable of self-assembling to form up a chain-like robot. Figure 1 shows the main idea of the work. Each robot module is equipped with a docking connector (connection mechanism) on the front plate and a universal joint in the middle. Note that the modules are designed with an articulated central joint, rather than a traditional axle for steering. This design means that no additional actuators are necessary to permit the creation of a snake-like serial chain when the modules are docked.

2. Introduction

A robotic system can be defined as a collection of members which are employed to do particular tasks. For many applications, it is possible to use a certain structure to complete the tasks. However, in untraditional environments and unexpected situations, it is almost impossible for a fixed-architecture robot to meet all the task requirements.

The work presented in this paper enables mobile robots to overcome more sophisticated tasks and enables modular robots to change the number of their modules to complete a specific task.

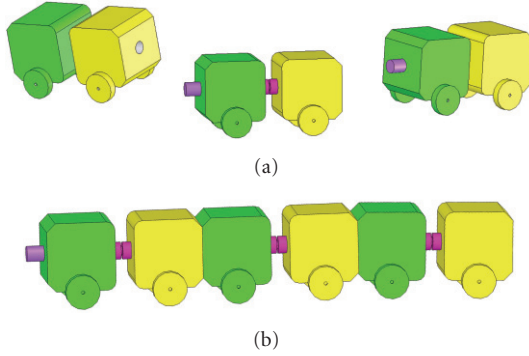


FIGURE 1: The main idea of the work: (a) each module is equipped with a docking connector and a universal joint; (b) the modules can dock and couple together using their connection mechanisms.

We investigate *autonomous docking between separate modules* which covers

- (i) design and construction of a suitable connection mechanism;
- (ii) investigation of a parking control algorithm to drive the robot modules to a defined position and orientation;
- (iii) implementation of the system using a localization system.

We have already presented the design details of the connection mechanism in [1]. The proposed mechanism is suitable for our application since it is lightweight, compact, and powerful enough to secure a reliable connection. It overcomes significant alignment errors, and it is considerably power efficient. So, here we focus on implementation and experiments using proper control algorithm and localization system.

This research contains the study of articulated-steering robots kinematics, using the common model for center-articulated mobile robots [2] with some modifications. After defining the model, the next step is to develop a stable control law to steer the robot modules from any initial position and orientation to the goal configuration.

The *feedback control of center-articulated mobile robots* has rarely been addressed in the literature [3]. In articulated steering, the heading of the robot changes by folding the hinged chassis units. Apostolopoulos [4] presented a practical analytical framework for synthesis and optimization of wheeled robotic locomotion configurations.

Choi and Sreenivasan [5] investigated the kinematics of a multimodule vehicle using a numerical approach. The number of actuators in this design can vary from nine in a fully active system to a minimum of three.

Load-haul-dump (LHD) vehicles which transport ore in underground mines are articulated-steering vehicles, and their steering kinematics resembles center-articulated mobile robots kinematics. Corke and Ridley [2] developed a general kinematics model of the vehicle that shows how heading angle evolves with time as a function of steering angle and velocity.

A path-tracking criterion for LHD trucks is proposed in [6]. Marshall et al. [7] have also investigated localization and steering of an LHD vehicle in a mining network.

In another work, Ridley and Corke [8] derived a linear, state-space, mathematical model of the vehicle, purely from geometric consideration of the vehicle and its desired path.

Visual navigation is increasingly becoming a more attractive method for robot navigation [9]. The field of visual navigation is of particular importance mainly because of the rich perceptual input provided by vision. Montesano et al. [10] have presented a method to relatively localize pairs of robots fusing bearing measurements and the motion of the vehicles.

Dorigo et al. and Mondada et al. [11, 12] presented the Swarm-bot platform. The basic components of the system, called s-bots, is equipped with eight RGB LEDs distributed around the module and a video graphics array (VGA) omnidirectional camera. The camera can detect s-bots that have activated their LEDs in different colors.

A docking control strategy for security robots recharging has been suggested by Luo et al. [13] based on detecting an artificial landmark. In this configuration, there is a camera mounted on top of the robot and the video signal is captured by the image frame grabber installed inside the main controller.

The works presented in [14, 15] have also reported experiments where a robot with on-board vision docks with another robot.

Many other researchers have also studied other aspects of modelling and reconfiguration of modular mobile robots [16–22].

This paper is organized as follows. Section 2 surveys the related work. Section 3 discusses the parking problem. Section 4 presents the experimental results. Finally, Section 5 concludes the paper, and Section 6 points out some future work.

3. Steering Control

This section addresses the closed-loop steering of the active-joint center-articulated mobile robot. As illustrated in Figure 1, each robot module has a universal joint in the middle, so once they are connected, each module adds 2-DOF to the chain. Therefore, we focus on the steering kinematic of such robots, which in this paper are called “center-articulated” (In this work, we focus on planar motion only. We designed the robot to have out-of-plane capability, but this is left for future work).

To avoid confusion between this type of mobile robots and tractor-trailer vehicles [23], we emphasize on the word “active-joint”. The modules are subject to move and dock to one another. Here we call this docking maneuver “parking control.”

We first propose a kinematic model of an active-joint center-articulated mobile robot, and then a proper law is derived to stabilize the configuration of the vehicle to a small neighborhood of the goal. The control law is designed by Lyapunov techniques and relies on the equations of the robot in polar coordinates.

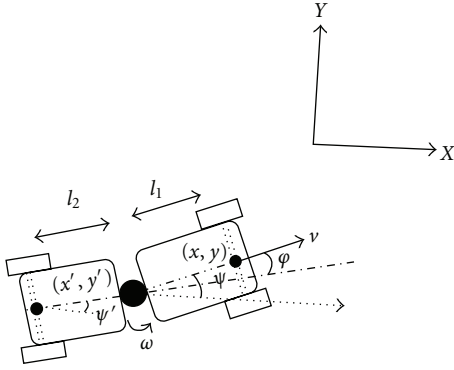


FIGURE 2: Diagram of a center-articulated mobile robot with respect to the target frame in Cartesian coordinates.

As discussed in [1], the designed connection mechanism allows significant misalignment. Therefore, steering the robot module to a small neighborhood of the goal is enough to achieve successful docking.

3.1. Kinematic Model. A center-articulated mobile robot consists of two bodies joined by an active joint. The vehicle is steered only by changing the body angle, since both axles are fixed.

Consider an active-joint center-articulated mobile robot positioned at a non-zero distance with respect to a target frame (Figure 2). The robot's motion is governed by the combined action of the linear velocity v and the angular velocity ω .

The kinematic equations of the robot which involve the robot's Cartesian position and the heading angle of the front body (x, y, ψ) can be written as

$$\dot{x} = v \cos \psi, \quad (1)$$

$$\dot{y} = v \sin \psi, \quad (2)$$

$$\dot{\psi} = \frac{\sin \varphi}{l_2 + l_1 \cos \varphi} v + \frac{l_2}{l_2 + l_1 \cos \varphi} \omega, \quad (3)$$

$$\dot{\varphi} = \omega, \quad (4)$$

where l_1 and l_2 are the lengths of the front and the rear parts of the robot, and φ is the body angle. Equations (1), (2), and (4) are similar to that of a simple differential-drive mobile robot. Equation (3) can be derived as follows.

The relationship between the front and the rear halves of the robot is given by

$$\begin{aligned} x' + l_2 \cos \psi' + l_1 \cos \psi &= x, \\ y' + l_2 \sin \psi' + l_1 \sin \psi &= y, \end{aligned} \quad (5)$$

where (x', y', ψ') denote the position and orientation of the rear part of the robot with respect to the target frame (Figure 2).

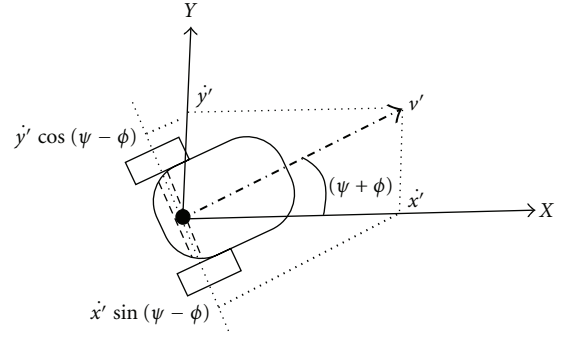


FIGURE 3: Derivation of nonholonomic constraints.

Taking time derivative of (5) gives

$$\begin{aligned} \dot{x}' - l_2 \dot{\psi}' \sin \psi' - l_1 \dot{\psi} \sin \psi &= \dot{x}, \\ \dot{y}' + l_2 \dot{\psi}' \cos \psi' + l_1 \dot{\psi} \cos \psi &= \dot{y}. \end{aligned} \quad (6)$$

We also know that $\psi' = \psi - \phi$. Therefore, considering (4), we can write

$$\dot{\psi}' = \dot{\psi} - \omega. \quad (7)$$

Substituting (1), (2), and (7) in (6) gives

$$\begin{aligned} \dot{x}' - l_2 (\dot{\psi} - \omega) \sin(\psi - \phi) - l_1 \dot{\psi} \sin \psi &= v \cos \psi, \\ \dot{y}' + l_2 (\dot{\psi} - \omega) \cos(\psi - \phi) + l_1 \dot{\psi} \cos \psi &= v \sin \psi. \end{aligned} \quad (8)$$

It is also assumed that there can be no motion parallel to the robot's axles. This constraint on rolling without slipping for the rear part implies that

$$\dot{x}' \sin(\psi - \phi) - \dot{y}' \cos(\psi - \phi) = 0. \quad (9)$$

This equation can be simply derived by projection of \dot{x}' and \dot{y}' onto wheels' axle (Figure 3).

Finally, solving (8) and (9) for $\dot{\psi}$ verifies (3).

The kinematic equations can also be written in polar coordinates. From Figure 4 we can write

$$e = \sqrt{x^2 + y^2}, \quad (10)$$

$$x = -e \cos \theta_1, \quad (11)$$

$$y = -e \sin \theta_1, \quad (12)$$

where e is the error distance, θ_1 is the error vector orientation with respect to the target frame, and θ_2 is the angle between the distance vector e and the linear speed vector.

The time derivative of (10) can be written as

$$\dot{e} = \frac{x\dot{x} + y\dot{y}}{e}. \quad (13)$$

Combining (11) and (12) with (13) yields

$$\dot{e} = -(\dot{x} \cos \theta_1 + \dot{y} \sin \theta_1). \quad (14)$$

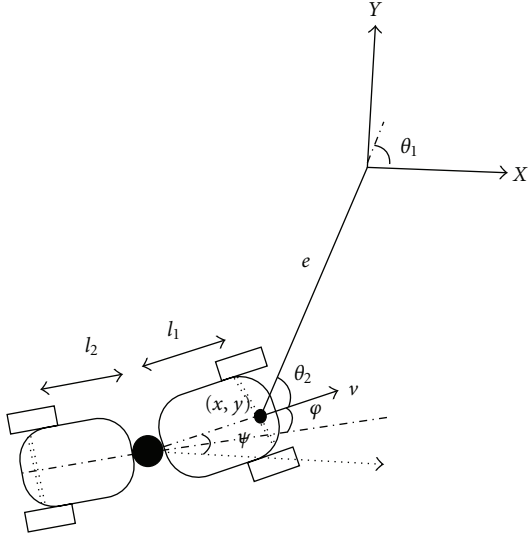


FIGURE 4: Diagram of a center-articulated mobile robot with respect to the target frame in polar coordinates.

Substituting (1) and (2) into (14) gives

$$\dot{e} = -v \cos \psi \cos \theta_1 - v \sin \psi \sin \theta_1. \quad (15)$$

So,

$$\dot{e} = -v \cos(\theta_1 - \psi). \quad (16)$$

As $\theta_2 = \theta_1 - \psi$, therefore

$$\dot{e} = -v \cos \theta_2. \quad (17)$$

Taking the time derivative of (11) and (12) and substituting (1) and (2) in the results yields

$$v \sin(\theta_1 - \psi) = e \dot{\theta}_1. \quad (18)$$

As $\theta_2 = \theta_1 - \psi$,

$$\dot{\theta}_1 = \frac{v \sin(\theta_2)}{e}. \quad (19)$$

Considering that $\dot{\theta}_2 = \dot{\theta}_1 - \dot{\psi}$, from (19) and (3) we obtain

$$\dot{\theta}_2 = \left(\frac{\sin \theta_2}{e} - \frac{\sin \varphi}{l_2 + l_1 \cos \varphi} \right) v - \frac{l_2}{l_2 + l_1 \cos \varphi} \omega. \quad (20)$$

Therefore, the kinematic equations of a center-articulated mobile robot in polar coordinates can be summarized as

$$\begin{bmatrix} \dot{e} \\ \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} -\cos \theta_2 & 0 \\ \frac{\sin \theta_2}{e} & 0 \\ \frac{\sin \theta_2}{e} - \frac{\sin \varphi}{l_2 + l_1 \cos \varphi} & -\frac{l_2}{l_2 + l_1 \cos \varphi} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}. \quad (21)$$

It is interesting to note that using polar coordinates allows for a set of state variables which closely resemble the same ones regularly used within our car-driving experience [24]. In the next section, it will be shown that (21) are suitable to design an appropriate control law for parking maneuvers.

3.2. Controller Design. The Lyapunov stability theory is a common tool to design control systems (see, for example, Bullo and Lewis [25] for a general introduction). Here we consider a simple quadratic equation as a candidate Lyapunov function.

Let the robot be initially positioned at a nonzero distance from the target frame. The objective of the parking control system is to move the robot so that it is accurately aligned with the target frame.

In other words, it is intended to find a stable control law $[v(e, \theta_1, \theta_2, \varphi), \omega(e, \theta_1, \theta_2, \varphi)]$ which drives the robot from any initial position $(e(0), \theta_1(0), \theta_2(0))$ to a small neighborhood of the target, $(0, 0, 0)$.

Consider the positive definite form

$$V = \frac{1}{2} \lambda_1 e^2 + \frac{1}{2} \lambda_2 \theta_1^2 + \frac{1}{2} \lambda_3 \theta_2^2 + \frac{1}{2} \lambda_4 \varphi^2, \quad (22)$$

$\lambda_1, \lambda_2, \lambda_3, \lambda_4 > 0.$

The time derivative of V can be expressed as

$$\dot{V} = \lambda_1 e \dot{e} + \lambda_2 \theta_1 \dot{\theta}_1 + \lambda_3 \theta_2 \dot{\theta}_2 + \lambda_4 \varphi \dot{\varphi}. \quad (23)$$

Substituting (21) in (23) gives

$$\begin{aligned} \dot{V} = & \left[(\lambda_2 \theta_1 + \lambda_3 \theta_2) \frac{\sin \theta_2}{e} - \lambda_1 e \cos \theta_2 - \frac{\lambda_3 \theta_2 \sin \varphi}{l_2 + l_1 \cos \varphi} \right] v \\ & + \left(\lambda_4 \varphi - \frac{l_2 \lambda_3 \theta_2}{l_2 + l_1 \cos \varphi} \right) \omega. \end{aligned} \quad (24)$$

It can be seen that letting

$$v = - \left[(\lambda_2 \theta_1 + \lambda_3 \theta_2) \frac{\sin \theta_2}{e} - \lambda_1 e \cos \theta_2 - \frac{\lambda_3 \theta_2 \sin \varphi}{l_2 + l_1 \cos \varphi} \right], \quad (25)$$

$$\omega = - \left(\lambda_4 \varphi - \frac{l_2 \lambda_3 \theta_2}{l_2 + l_1 \cos \varphi} \right) \quad (26)$$

makes $\dot{V} \leq 0$ which implies stability of the system states. Convergence (asymptotic stability) depends on the choice of λ_s , as discussed next.

3.3. Stability Analysis. The proposed candidate Lyapunov function V is lower bounded. Furthermore, \dot{V} is negative semidefinite and uniformly continuous in time (\ddot{V} is finite). Therefore, according to Barbalat's lemma [26], $\dot{V} \rightarrow 0$ as $t \rightarrow \infty$.

The time derivative of V can be expressed as

$$\begin{aligned} \dot{V} &= \Lambda_1 + \Lambda_2 \\ &= - \left[(\lambda_2 \theta_1 + \lambda_3 \theta_2) \frac{\sin \theta_2}{e} - \lambda_1 e \cos \theta_2 - \frac{\lambda_3 \theta_2 \sin \varphi}{l_2 + l_1 \cos \varphi} \right]^2 \\ &\quad - \left(\lambda_4 \varphi - \frac{l_2 \lambda_3 \theta_2}{l_2 + l_1 \cos \varphi} \right)^2. \end{aligned} \quad (27)$$

It is noted that as $\dot{V} \rightarrow 0$ and both Λ_1 and Λ_2 are squared, therefore $\Lambda_1 \rightarrow 0$ and $\Lambda_2 \rightarrow 0$. If λ_4 is selected to be very small, Λ_2 takes on the form

$$\Lambda_2 \approx - \left(\frac{l_2 \lambda_3 \theta_2}{l_2 + l_1 \cos \varphi} \right)^2. \quad (28)$$

So, $\Lambda_2 \rightarrow 0$ implies that $\theta_2 \rightarrow 0$.

As $\theta_2 \rightarrow 0$, Λ_1 also takes on a simpler form of

$$\Lambda_1 \approx - \left(\lambda_2 \theta_1 \frac{\theta_2}{e} - \lambda_1 e \right)^2. \quad (29)$$

Consequently, $\Lambda_1 \rightarrow 0$ gives

$$\lambda_2 \theta_1 \theta_2 \approx \lambda_1 e^2. \quad (30)$$

As $\theta_2 \rightarrow 0$, we get $e \rightarrow 0$.

Finally, in the limit where both e and θ_2 go to zero, θ_2/e is bounded and (25) gives

$$\lim_{e, \theta_2 \rightarrow 0} v = - \frac{\lambda_2 \theta_1 \theta_2}{e}. \quad (31)$$

Therefore, from (19) we obtain

$$\dot{\theta}_1 = -\lambda_2 \left(\frac{\theta_2}{e} \right)^2 \theta_1. \quad (32)$$

As $\lambda_2 > 0$ and $(\theta_2/e)^2$ is positive, from (32) it is found that θ_1 is stable and eventually approaches zero, though it may do so slowly.

Therefore, $\dot{V} \rightarrow 0$ results in $(e, \theta_1, \theta_2) \rightarrow (0, 0, 0)$.

Since this system is driftless, Brockett's condition [25] predicts that a smooth control will not stabilize the system. However, in this case, it is not necessary to stabilize the entire state of the system because φ is only the internal body angle, which can always be corrected by lifting one end thus eliminating the nonslip constraint on one axle. As a result, we work only to steer the triple (e, θ_1, θ_2) to a near neighbourhood of $(0, 0, 0)$, indicating that the robot is in position to dock.

In practice, there is a trade-off in selecting parameter λ_4 . Setting $\lambda_4 = 0$ stabilizes (e, θ_1, θ_2) while rendering φ uncontrollable. In such cases, φ can take on physically unrealizable values, for example, causing the robot to fold in on itself. By contrast, choosing λ_4 large can result in very slow approaches to the origin.

It should also be mentioned that the proposed model for the center-articulated mobile robots has a singularity at

$e = 0$, since according to (21), $\dot{\theta}_1$ and $\dot{\theta}_2$ are not defined at $e = 0$. The condition $e = 0$ cannot occur in finite time since the approach to zero is asymptotic.

One may also observe another singularity. If $l_2 + l_1 \cos \varphi = 0$ then θ_2 is not defined. If the robot is designed such that $l_2 > l_1$, this singularity never happens. If $l_2 = l_1$, $\cos \varphi = \pm \pi$ results in this singularity, but this case cannot occur since it means that the robot is fully folded back on itself.

Finally, we note that there is a special case where the controller is not able to stabilize the configuration of the robot. This special case occurs when both φ and θ_2 are initially zero. As can be observed from (25) and (26), in this situation $\omega = 0$ and $v = \lambda_1 e$. In fact, there is no control on θ_1 . The controller should recognize this special case and take a proper action. For instance, the controller can change the initial body angle to a nonzero value.

4. Experiments

In previous section, we introduced a control law to steer a center-articulated mobile robot in order to achieve successful parking (docking). In our experiments, we use a beacon-based localization system [27] to determine the pose of the robot relative to the target position, however that is only an implementation detail. In principle any localization scheme could be used. We include details of our beacon-based localization approach in this paper only for completeness of the description of the experimental setup.

The simulation results and the effect of measurement noise are also presented in our previous works [27, 28].

In this section, we first discuss the design details of the robot and the experimental setup. We then present experimental results on the robot system to verify the effectiveness of the proposed approach.

4.1. Robot Design. In order to provide a platform to perform our experiments, we designed and constructed an articulated-steering mobile robot. The robot module consists of a dual-actuated universal joint with servomotors as the joint actuators (Figure 5).

The robot is driven by a single-motor gearbox and the actuators are controlled by a control board. The robot is also equipped with an omnidirectional camera which measures the view angles of the beacons by means of a real-time color detection algorithm implemented on a PC. The PC calculates the control signals (the motor speed and the servo angle) which are transmitted to the control board through serial communication.

In this design, two servomotors are located on the yokes which turn the axles of the middle piece. Rotating the horizontal axle moves the joint up and down (pitch), and rotating the vertical axel moves the joint from side to side (yaw).

4.2. Visual System. We setup a vision-based localization system using an omnidirectional camera that provides a description of the environment around the robot. The system

Step 1: The image is rotated and flipped horizontally to be aligned with the actual position of the robot.
 Step 2: The image is cropped to only include the beacons area.
 Step 3: The raw image is labeled which allows to refer to the image currently being processed at a later time.
 Step 4: RGB filtering is performed to detect the first color.
 Step 5: Closing process is performed to connect nearby detected objects.
 Step 6: Blob-size filtering is performed to remove the image noise.
 Step 7: The center-of-gravity of the detected beacon is marked as the position of the beacon in the image plane.
 Step 8: Steps 4–7 are repeated for the second and third color, using the labeled raw image.

ALGORITHM 1: The real-time image-processing algorithm.

Step 1: α , β , and γ angles are computed, based on the position of the beacons received from the Image Processing application. (The angles are confined to $[-\pi, \pi]$)
 Step 2: Feedback parameters (e, θ_1, θ_2) are computed,
 Step 3: Control signals (v, w) are computed, based on equations (25),(26).
 Step 4: If the control signals exceed their limits, they are set to their maximum allowable values.
 Step 5: The computed control signals are scaled, and sent to the control-board.
 Step 6: Once a new image frame arrives, Steps 1–5 are repeated.

ALGORITHM 2: The control algorithm to steer the robot.

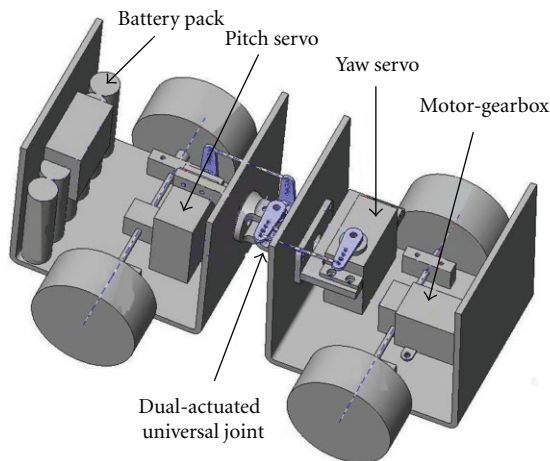


FIGURE 5: The design of the articulated-steering mobile robot.

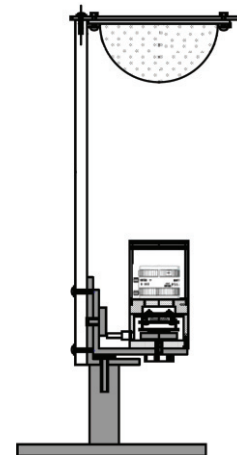


FIGURE 6: The design of the vision-based localization system, composed of a camera pointed upwards to a spherical mirror.

is composed of a camera pointed upwards looking at a spherical mirror, mounted on the top of the robot (Figure 6). We do not assume to know any camera/mirror calibration parameters (mirror size or focal length).

Three red, green, and blue color objects are considered as beacons, located on the top of the target rover to determine the pose of the target. The colored-beacons are detected by

the camera, and the images are transferred to an off-board computer for real-time processing.

4.3. Image Processing. Once the images are transferred to the PC, a machine-vision software processes incoming image data and detects the position of the beacons in the image plane.

Step 1: The position of the via points (for point V_{pi} , $(X_{V_{pi}}, Y_{V_{pi}})$) is pre-defined in the algorithm (Figure 7(b)). Detection flag (D-Flag) and Point-Flag (P-Flag) are two status flags to determine the status of the robot. P-Flag indicates the next via point that should be reached. D-Flag indicates if the via points are followed.

Step 2:

If $\alpha < 0$ and $\beta < 0$ and D-Flag = 0, then P-Flag = 1 and D-Flag = 1.

If $\alpha > 0$ and $\beta < 0$ and D-Flag = 0, then P-Flag = 2 and D-Flag = 1.

If $\alpha < 0$ and $\beta > 0$ and D-Flag = 0, then P-Flag = 5 and D-Flag = 1.

Step 3: If P-Flag = i , $X_r = X_r - X_{V_{pi}}$, and $Y_r = Y_r - Y_{V_{pi}}$.

So, the calculation of feedback parameters is changed.

Step 4: The control algorithm (Algorithm 2) is then followed to reach the specified via point.

Step 5:

If P-Flag = 4, then steer to the actual target frame.

If P-Flag = 3 and $e \ll$, then P-Flag = 4.

If P-Flag = 6 and $e \ll$, then P-Flag = 4.

If P-Flag = 2 and $e \ll$, then P-Flag = 3.

If P-Flag = 5 and $e \ll$, then P-Flag = 6.

If P-Flag = 3 and $e \ll$, then P-Flag = 4.

If P-Flag = 1 and $e \ll$, then P-Flag = 2.

Step 6: Steps 2–5 are repeated once a new image frame arrives.

ALGORITHM 3: The algorithm to follow the via points.

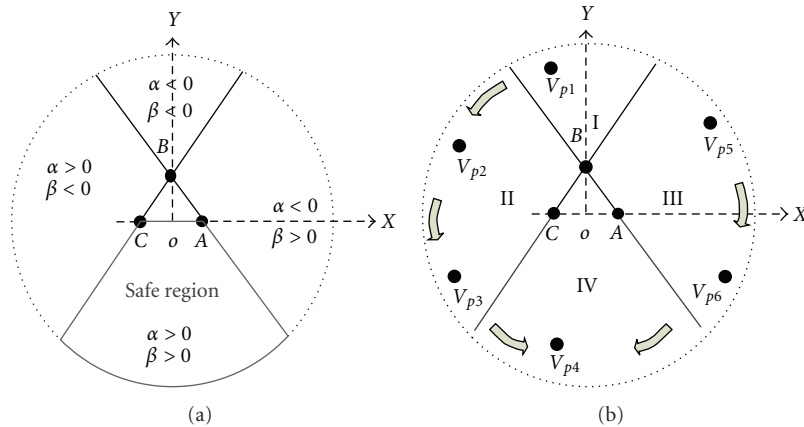


FIGURE 7: (a) The robot's workspace can be divided into four parts, considering the sign of α and β . The region where both α and β are positive is called safe region where the robot reaches the goal with no need to pass through the beacons. (b) A set of assumed via points in the workspace; the via points are reached in an order such that the robot is finally located in the safe region.



FIGURE 8: A general view of the built robot, including the docking connector and the visual system.

In this process, we first perform color filtering (RGB filtering), followed by a closing operation. In RGB filtering, all pixels that are not of the selected color are diminished. Closing has the effect of filling small and thin holes in objects by connecting nearby objects in the image plane [29].

Then, blob-size filtering is performed to remove objects below a certain size. As a result, each beacon is located as a single blob in the image. The center of gravity of each blob determines the position of each beacon in the image plane.

The image processor outputs the beacons' positions on the image plane, from which the bearing from each beacon can be computed. The steps of image processing algorithm are summarized in Algorithm 1.

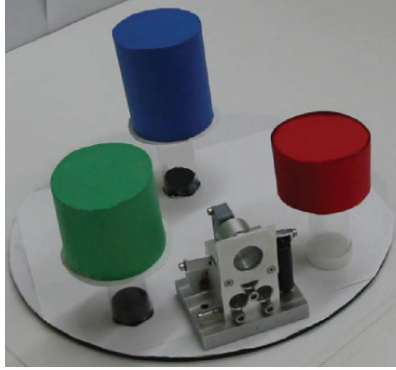


FIGURE 9: Three cylindrical-shaped color objects are used as the beacons, and the female connector is fixed on the origin of the target frame.

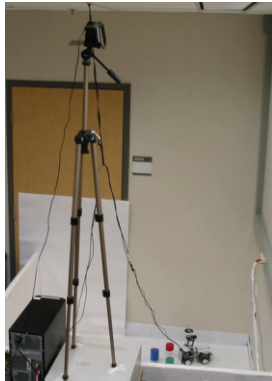
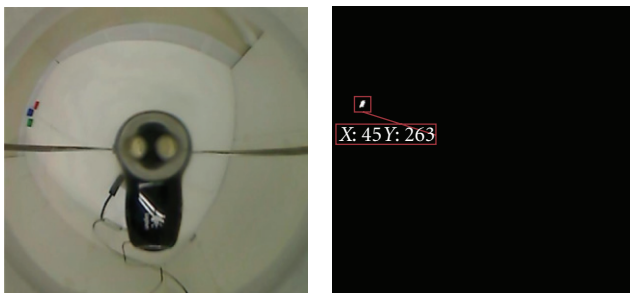
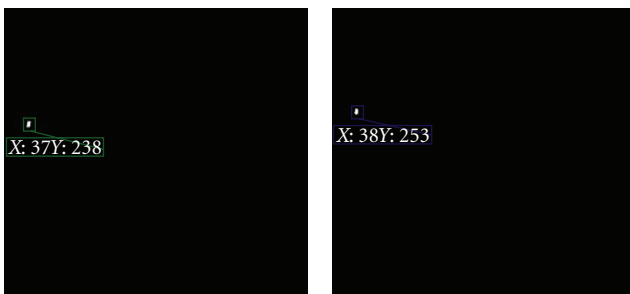


FIGURE 10: The experimental setup.



(a)

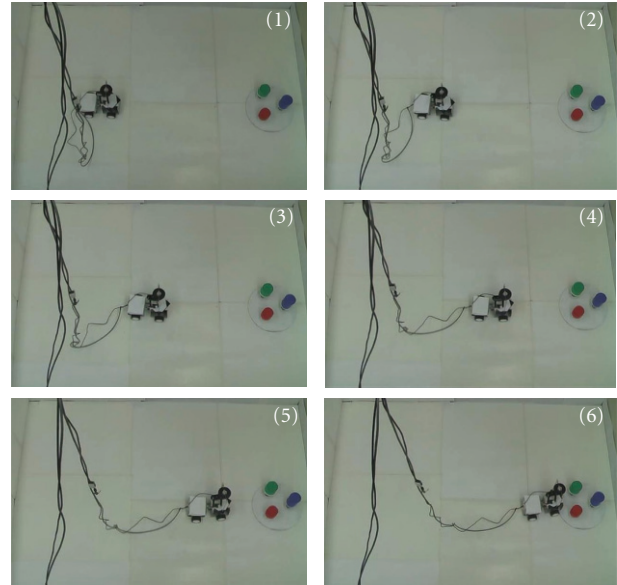
(b)



(c)

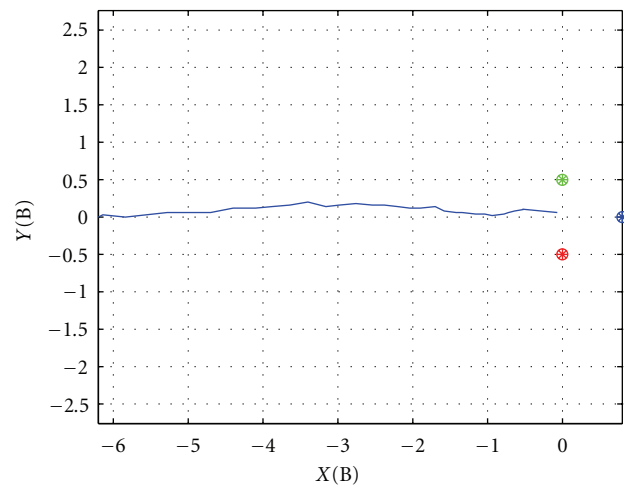
(d)

FIGURE 11: The process of detecting the beacons: (a) an image frame received from the camera, (b) red beacon detection, (c) green beacon detection, (d) blue beacon detection.



(a)

The robot's travelled path



(b)

FIGURE 12: The robot is initially positioned at $(6,0,0,0)$ (a) the snapshots of the experiment, (b) the traveled path of the robot.

4.4. Control Algorithm. In this section, we briefly describe the control algorithm to steer the robot to the neighborhood of the target. The control algorithm is implemented on the PC which receives the measurements of the beacons' positions and sends the control signals to the control board.

The algorithm starts with relative bearing measurement, using the data provided from the camera. In case that the measured angles are beyond the interval $[-\pi, \pi]$, $\pm 2\pi$ is added to the computed angle. It is noted that some parameters such as the beacons' position in target frame, robot's length, and the controller's gains are predefined in the algorithm.

Once the angles are determined, the feedback parameters (e, θ_1, θ_2) are calculated based on equations described in [27].

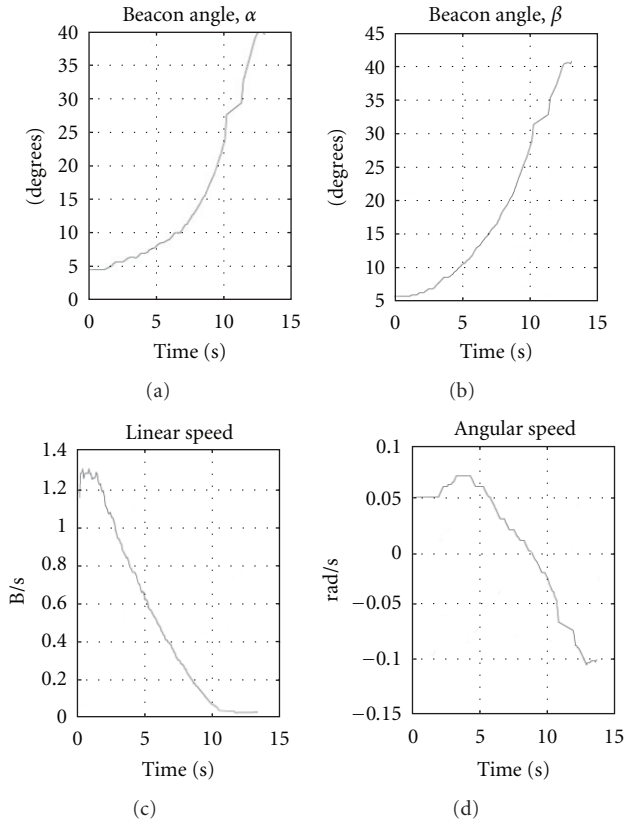


FIGURE 13: The changes in α , β , and the control signals for the first experiment.

Now that the feedback parameters are available, in the next step, the control signals (v , ω) are computed according to the equations given in Section 3.2.

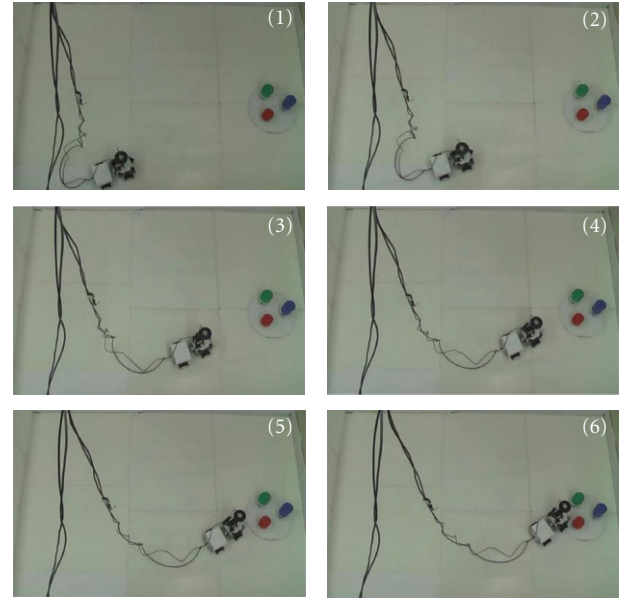
The calculated control signals are not directly applied to the robot. As there are limitations on robot's speed and body angle, the control signals are restricted within defined upper bounds. Finally, the control signals are scaled to adjust the effect of actuator dead band.

The whole process is performed once a new image frame arrives from the camera. The control algorithm is summarized in Algorithm 2.

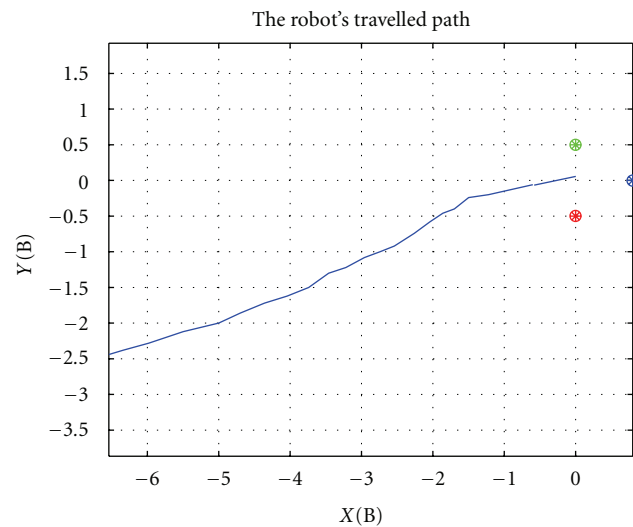
The simulation results reveal that, in some cases, the generated path passes through the beacon's region. Since this implies that two robots will be physically occupying the same space, the control algorithm must recognize this situation and resolve it. The solution is as follows, based on locating a set of via points in the robot's workspace.

As Figure 7(a) indicates, the robot's workspace can be divided into four parts, considering the sign of α and β angles. The region where both α and β are positive is considered as the safe region. This region is called safe since if the robot's initial position is located in this region, the robot reaches the goal with no need to pass through the beacons.

Therefore, if the control algorithm detects that the robot is not in the safe region, it steers the robot to first pass through a set of predefined via points. Figure 7(b) shows a set of assumed via points in the workspace. According to the



(a)



(b)

FIGURE 14: The robot is initially positioned at $(7,5\pi/8,0,0)$, (a) the snapshots of the experiment, (b) the traveled path of the robot.

figure, the via points are reached in an order such that the robot is finally located in the safe region.

The via points are reached using the same control algorithm summarized in Algorithm 2. For each point, the target frame is redefined for that specific via point. In this approach, the feedback still comes from the beacons located on the target frame, but using a simple translation, the robot is steered to a coordinate system, originated at the specified via point. Once the error distance to that via point is small enough, the next via point is followed till the robot is located in the safe region.

This process is done by setting/resetting status flags. The process of following via points is summarized in Algorithm 3.

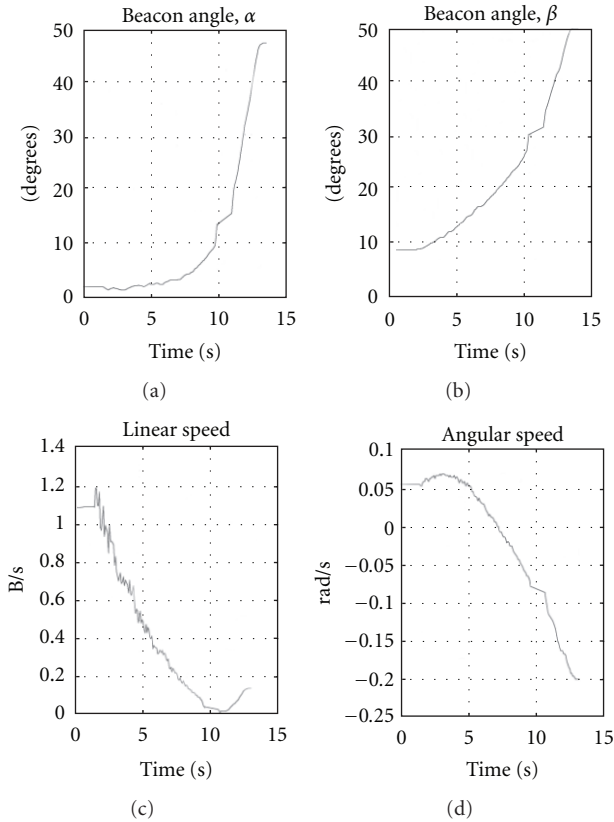


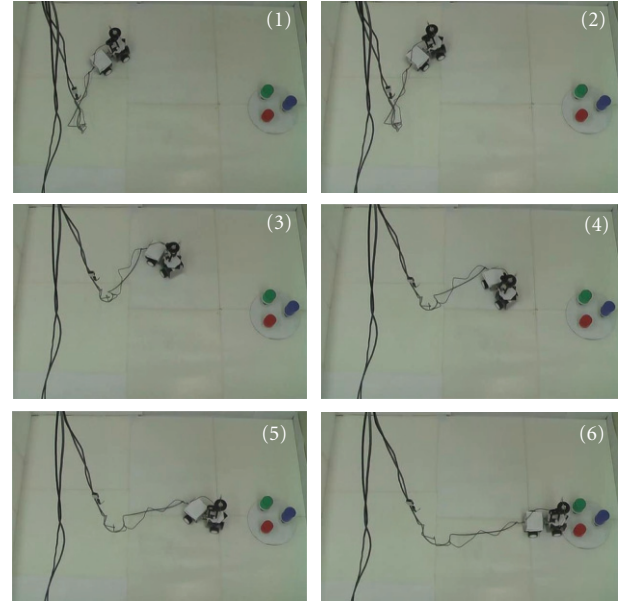
FIGURE 15: The changes in α , β , and the control signals for the second experiment.

4.5. Robot Construction. Based on the mentioned design approach, a prototype of the robot has been implemented including the docking connector and the visual system. Figure 8 presents a general view of the built robot. The prototype weighs less than 1.0 kg, and the total length of the robot is 24 cm.

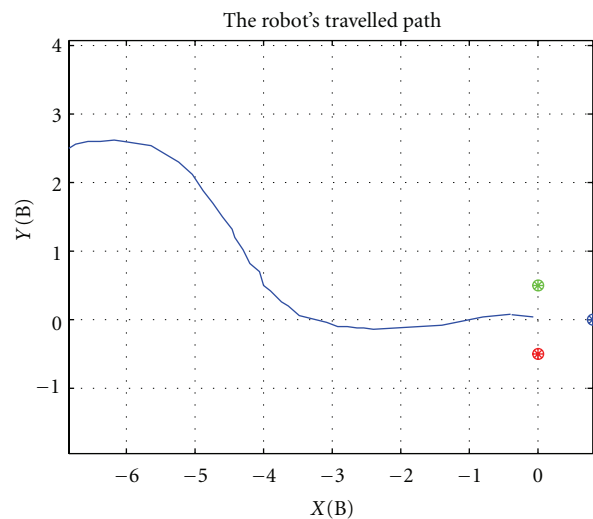
The universal joint's actuators are two Ultra Torque HSR-5995TG servomotors from HiTec company which gives maximum 3.8 Nm standing torque. The robot is driven by a twin-motor gearbox, consisting of two small DC motors, both connected to a single gearbox (ratio 344 : 1). The robot's actuators are controlled by the main control board equipped with an HCS12 Motorola microcontroller and a MicroCore-11 motor drive. The control board communicates with the PC through RS232 serial communication.

4.6. Experimental Setup. To test the proposed approach, an experimental setup has been designed in which the performance of the built articulated-steering robot is examined. The experimental setup is illustrated in Figures 9 and 10. The experiments have been made in a small field made of white styrofoam blocks.

Figure 11(a) illustrates an image frame received from the camera (the image is cropped). Figures 11(b)–11(d) show the results of image processing algorithm. The size of the cropped image is 420×400 pixels.



(a)



(b)

FIGURE 16: The robot is initially positioned at $(7.5, \pi/8, \pi/4, 0)$, (a) the snapshots of the experiment, (b) the traveled path of the robot.

The software platform used to implement the control algorithm is Microsoft Visual Studio 9.0. The codes are written in C++ and executed on a 3.0 GHz Intel dual core processor with 1 GB of RAM. The software platform calculates the control signals (the motor speed and the servo angle) which are transmitted to the control board through RS232 serial communication.

A power supply supplies 12 V DC at 1 A to the control board, including an HCS12 Motorola microcontroller which is programmed to generate appropriate PWM signals to the actuators.

We perform experiments while a fixed digital camera records the behavior of the robot. Image processing is

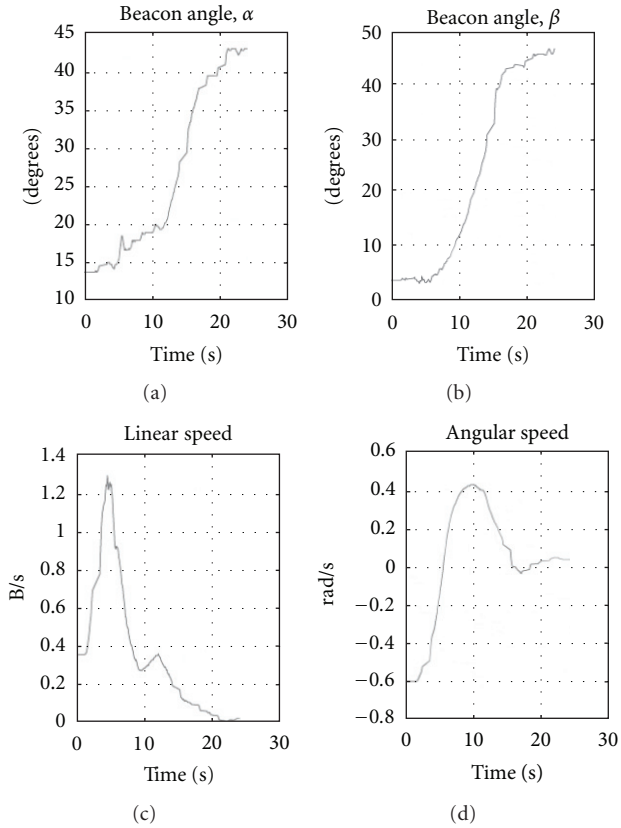


FIGURE 17: The changes in α , β , and the control signals for the third experiment.

then performed off-line using MATLAB Image Processing Toolbox to yield the robot's true position.

4.7. Experimental Results. A set of experiments were carried out to show the performance of the proposed algorithms. The experimental results are shown graphically in the following figures. In all experiments, the distance between the center of the beacons has been considered as the unit of distance, "B". So, the beacons' distances are $a = b = 1$ B, and as the beacons are located on an equilateral triangle, $\delta = \pi/3$ (see the Localization section in [27]). The lengths of the robot are also $l_1 = l_2 = 0.5$ B.

The controller gains are chosen to be $\lambda_1 = 0.8$, $\lambda_2 = 1.05$, $\lambda_3 = 0.95$, and $\lambda_4 = 0.02$. These gains are obtained by trial and error. It is noted that only choosing positive gains is enough to achieve a stable control law (see Section 3.2), but fine adjustment to improve the performance is done considering the actuators' limits.

Figure 12(a) shows the snapshots of an experiment where the robot is positioned at $(e, \theta_1, \theta_2, \varphi) = (6, 0, 0, 0)$. It is noted that the distance unit is in B. Figure 12(b) illustrates the traveled path of the robot for this experiment.

Figure 13 shows the changes in α and β angles (the inputs to the controller), sent from the image processing algorithm, as well as the control signals v and ω (the outputs of the controller), applied to the robot.

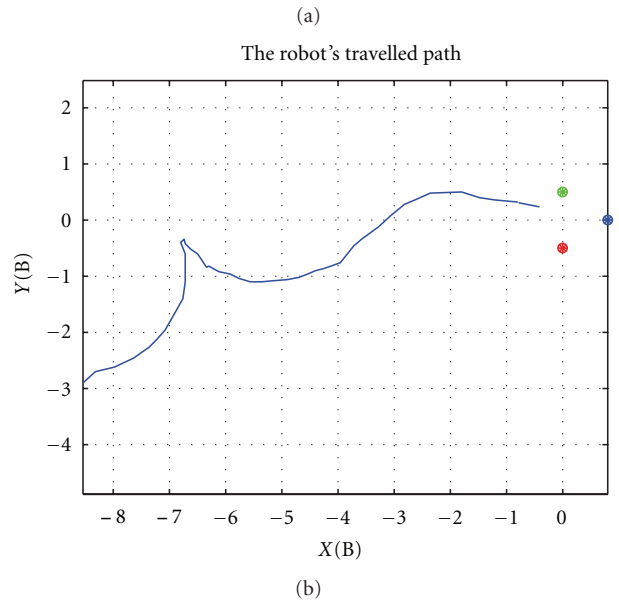
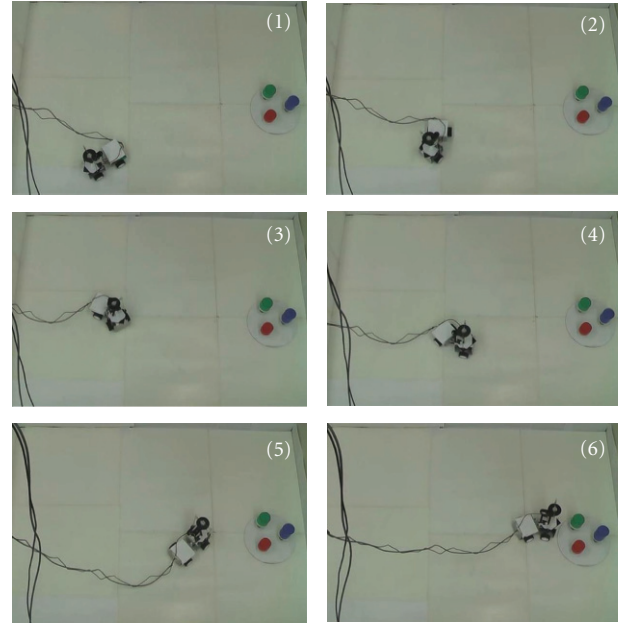


FIGURE 18: The robot is initially positioned at $(8.5\pi/8, -\pi, 0)$, (a) the snapshots of the experiment, (b) the traveled path of the robot.

Figures 14(a), 14(b), 15, 16, 17, 18 and 19 show the results of some other experiments, while the robot is located in different positions and orientations relative to the target frame (imagine that the red and the green beacons are located on the X-axis of the target frame). For each experiment, the snapshots of the experiment and the actual traveled path are presented. The figures also show the inputs and outputs of the controller. In these figures, the robot is located at $(7.5\pi/8, 0, 0)$, $(7.5, \pi/8, \pi/4, 0)$, and $(8.5\pi/8, -\pi, 0)$, respectively.

As mentioned in Section 4.4, in case the generated path passes through the beacon's region, the control system steers the robot through a set of predefined via points. The via

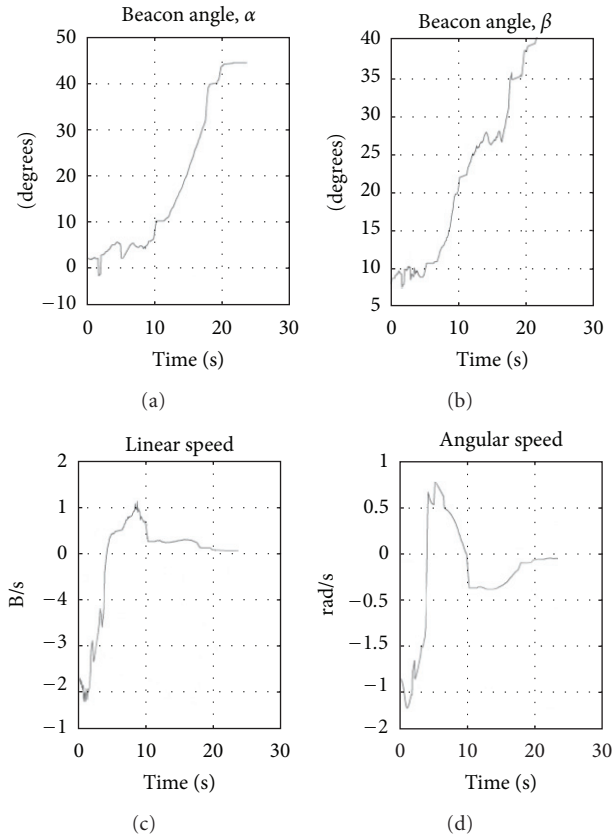


FIGURE 19: The changes in α , β , and the control signals for the fourth experiment.

points are reached in an order such that the robot is finally located in the safe region (Figure 7(b)).

Finally, Figure 20 shows the snapshots of an experiment where the initial position of the robot is not in the safe region. As can be seen, the robot has been steered appropriately to park the robot at the target.

Because our docking joint permits wide initial positioning errors (± 40 degrees of yaw and 11 mm of offset), once the robot falls below a threshold speed, it suffices to go forward in a straight line in order to accomplish docking. In these experiments, the robot docked successfully on each of trials.

We also performed a set of experiments with actual docking joints [1] to visualize the connection maneuver. As Figure 21 shows, the connector's pieces are fixed in front of the robot, as well as on the origin of the target frame. This experiment was repeated ten times, and for each trial, the robot successfully completed the connection maneuver.

5. Conclusion

This research work introduced the idea of a new robotic system including a team of wheeled mobile robots which are able to autonomously assemble and form a chain-like robot. The goal is to improve the performance of mobile robots by autonomously changing their locomotion method.

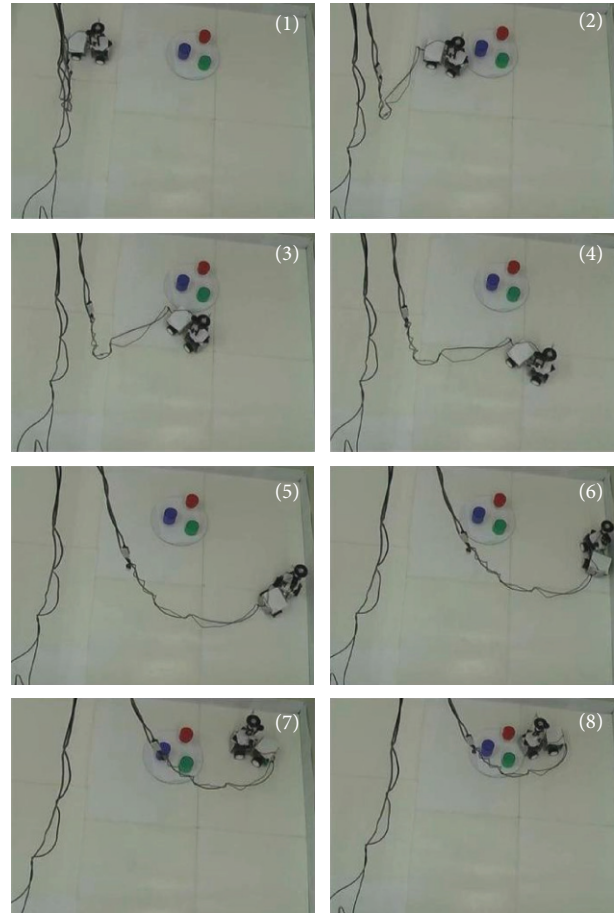


FIGURE 20: The snapshots of the second experiment where the initial position of the robot is not in the safe region.

We proposed the design of a center-articulated module to form the building block of a robot team that can form itself into a serial chain where required by terrain constraints.

Next, we proposed a kinematic model of an active-joint center-articulated mobile robot in polar coordinates, and then a proper control law was derived to stabilize the configuration of the vehicle to a small neighborhood of the goal, using Lyapunov techniques. The results reveal that choosing a suitable state model allows to use a simple Lyapunov function to achieve parking control, even if the feedback is noisy.

We finally designed and constructed a center-articulated mobile robot equipped with a beacon-based localization system to verify our approach. The experimental results show the effectiveness of the proposed approach.

6. Future Work

An important extension to this research work would be investigating self-reconfiguration, using a set of autonomous wheeled mobile robots and the proposed approach. The next step could also be maneuvering the chain to overcome tasks such as passing through a narrow space and climbing up

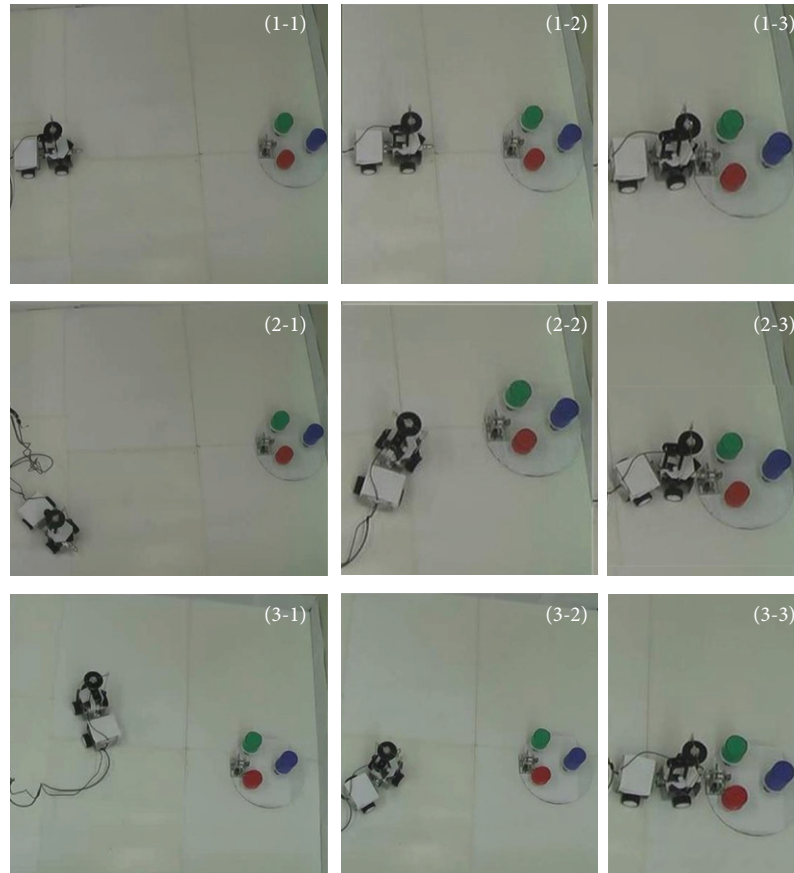


FIGURE 21: A set of experiments, performed to visualize the connection maneuver.

steps. A fully functional modular team of these units will also require the intelligence to examine surrounding terrain to determine when docking is necessary.

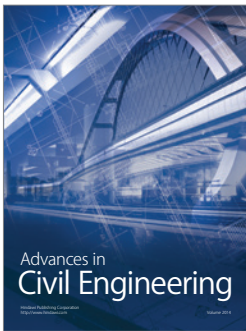
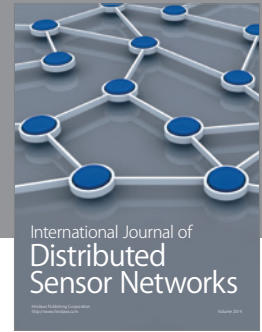
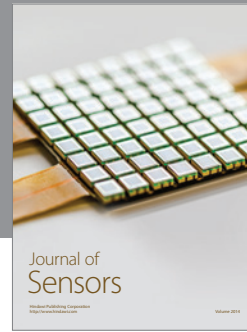
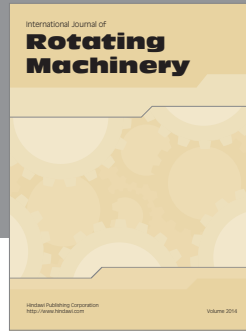
There are a large number of open research topics in repairable systems. The problem of designing a robot team with repair capabilities is complex and not easily solved [30]. The proposed configuration can be extended to answer some of the questions in this area.

Investigating dynamic docking problem can also be considered as a future work. In dynamic docking, the robot modules find and connect to one another while they are all moving. The dynamic docking offers faster docking time since there is no need for the modules to stop and perform parking maneuver. As the target is in motion, vector tracking should be investigated rather than regulation problem in parking control.

References

- [1] M. Delrobaei and K. A. McIsaac, "Connection mechanism for autonomous self-assembly in mobile robots," *IEEE Transactions on Robotics*, vol. 25, no. 6, pp. 1413–1419, 2009.
- [2] P. I. Corke and P. Ridley, "Steering kinematics for a center-articulated mobile robot," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 2, pp. 215–218, 2001.
- [3] J. P. Laumond, *Robot Motion Planning and Control*, Springer, 1998.
- [4] D. Apostolopoulos, "Analytical configuration of wheeled robotic locomotion," Tech. Rep. CMU-RI-TR-01-08, Robotics Institute, Carnegie Mellon University, 2001.
- [5] B. J. Choi and S. V. Sreenivasan, "Gross motion characteristics of articulated mobile robots with pure rolling capability on smooth uneven surfaces," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 2, pp. 340–343, 1999.
- [6] C. Altafini, "Path-tracking criterion for an LHD articulated vehicle," *International Journal of Robotics Research*, vol. 18, no. 5, pp. 435–441, 1999.
- [7] J. Marshall, T. Barfoot, and J. Larsson, "Autonomous underground tramming for center-articulated vehicles," *Journal of Field Robotics*, vol. 25, no. 6-7, pp. 400–421, 2008.
- [8] P. Ridley and P. Corke, "Load haul dump vehicle kinematics and control," *Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME*, vol. 125, no. 1, pp. 54–59, 2003.
- [9] G. N. DeSouza and A. C. Kak, "Vision for mobile robot navigation: a survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 2, pp. 237–267, 2002.
- [10] L. Montesano, J. Gaspar, J. Santos-Victor, and L. Montano, "Fusing vision-based bearing measurements and motion to localize pairs of robots," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 2333–2338, August 2005.
- [11] M. Dorigo, V. Trianni, E. Şahin et al., "Evolving self-organizing behaviors for a swarm-bot," *Autonomous Robots*, vol. 17, no. 2-3, pp. 223–245, 2004.

- [12] F. Mondada, G. C. Pettinaro, A. Guignard et al., "Swarm-bot: a new distributed robotic concept," *Autonomous Robots*, vol. 17, no. 2-3, pp. 193–221, 2004.
- [13] R. C. Luo, C. T. Liao, K. L. Su, and K. C. Lin, "Automatic docking and recharging system for autonomous security robot," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2953–2958, August 2005.
- [14] C. Bererton and P. K. Khosla, "Towards a team of robots with repair capabilities: a visual docking system," in *Proceedings of the 7th ISER Conference*, vol. 271, pp. 333–342, Berlin, Germany, 2001.
- [15] S. Murata, K. Kakomura, and H. Kurokawa, "Docking experiments of a modular robot by visual feedback," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '06)*, pp. 625–630, October 2006.
- [16] J. Liu, Y. Wang, S. Ma, and Y. Li, "Enumeration of the non-isomorphic configurations for a reconfigurable modular robot with square-cubic-cell modules," *International Journal of Advanced Robotic Systems*, vol. 7, no. 4, pp. 58–68, 2010.
- [17] J. Wang and Y. Li, "Kinematic analysis of a mobile robot with two-body frames," in *Proceedings of the IEEE International Conference on Information and Automation (ICIA '08)*, pp. 1073–1078, June 2008.
- [18] S. Hirose, *Biologically Inspired Robots (Snake-Like Locomotors and Manipulators)*, Oxford University Press, 1993.
- [19] S. Hirose, T. Shirasu, and E. F. Fukushima, "Proposal for cooperative robot "Gunryu" composed of autonomous segments," *Robotics and Autonomous Systems*, vol. 17, no. 1-2, pp. 107–118, 1996.
- [20] S. Hirose, R. Damoto, and A. Kawakami, "Study of Super-Mechano-Colony (concept and basic experimental setup)," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1664–1669, November 2000.
- [21] S. Hirose and M. Mori, "Biologically inspired snake-like robots," in *Proceedings of the IEEE International Conference on Robotics and Biomimetics (IEEE ROBIO '04)*, pp. 1–7, August 2004.
- [22] R. Damoto, A. Kawakami, and S. Hirose, "Study of super-mechano colony: concept and basic experimental set-up," *Advanced Robotics*, vol. 15, no. 4, pp. 391–408, 2001.
- [23] A. Astolfi, P. Bolzern, and A. Locatelli, "Path-tracking of a tractor-trailer vehicle along rectilinear and circular paths: a Lyapunov-based approach," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 1, pp. 154–160, 2004.
- [24] M. Aicardi, G. Casalino, A. Bicchi, and A. Balestrino, "Closed loop steering of unicycle-like vehicles via Lyapunov techniques," *IEEE Robotics and Automation Magazine*, vol. 2, no. 1, pp. 27–35, 1995.
- [25] F. Bullo and A. D. Lewis, *Geometric Control of Mechanical Systems*, Springer, 2004.
- [26] J. Slotine and W. Li, *Applied Nonlinear Control*, Prentice-Hall, 1991.
- [27] M. Delrobaei and K. A. McIsaac, "Parking control of an active-joint center-articulated mobile robot based on feedback from beacons," in *Proceedings of the 23rd Canadian Conference on Electrical and Computer Engineering (CCECE '10)*, Calgary, Canada, May 2010.
- [28] M. Delrobaei and K. A. McIsaac, "Parking control of a center-articulated mobile robot in presence of measurement noise," in *Proceedings of the IEEE Conference Robotics Automation and Mechatronics (RAM '10)*, pp. 453–457, Singapore, June 2010.
- [29] K. R. Castleman, *Digital Image Processing*, Prentice-Hall, 1996.
- [30] C. Bererton and P. K. Khosla, "Towards a team of robots with reconfiguration and repair capabilities," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '01)*, pp. 2923–2928, May 2001.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

