Autonomous Docking of Hybrid-Wheeled Modular Robots With an Integrated Active Genderless Docking Mechanism

Shubhdildeep Singh Sohal
Mem. ASME
Robotics and Mechatronics Laboratory,
Department of Mechanical Engineering,
Virginia Tech,
Blacksburg, VA 24060
e-mail: shubh94@vt.edu

Bijo Sebastian
Mem. ASME
Robotics and Mechatronics Laboratory,
Department of Mechanical Engineering,
Virginia Tech,
Blacksburg, VA 24060
e-mail: bijo7@vt.edu

Pinhas Ben-Tzvi[1]
Mem. ASME
Robotics and Mechatronics Laboratory,
Department of Mechanical Engineering, Electrical
& Computer Engineering,
Virginia Tech,
Blacksburg, VA 24060
e-mail: bentzvi@vt.edu

This paper presents a self-reconfigurable modular robot with an integrated two degrees-of-freedom (DOF) active docking mechanism. Active docking in modular robotic systems has received a lot of interest recently as it allows small versatile robotic systems to coalesce and achieve the structural benefits of large systems. This feature enables reconfigurable modular robotic systems to bridge the gap between small agile systems and larger robotic systems. The proposed self-reconfigurable mobile robot design exhibits dual mobility using a tracked drive for longitudinal locomotion and a wheeled drive for lateral locomotion. The 2-DOF docking interface allows for efficient docking while tolerating misalignments. To aid autonomous docking, visual marker-based tracking is used to detect and re-position the source robot relative to the target robot. The tracked features are then used in Image-Based Visual Servoing to bring the robots close enough for the docking procedure. The hybrid-tracking algorithm allows eliminating external pixelated noise in the image plane resulting in higher tracking accuracy along with faster frame update on a low-cost onboard computational device. This paper presents the overall mechanical design and the integration details of the modular robotic module with the docking mechanism. An overview of the autonomous tracking and docking algorithm is presented along with a proof-of-concept real-world demonstration of autonomous docking and self-reconfigurability. Experimental results to validate the robustness of the proposed tracking method, as well as the reliability of the autonomous docking procedure, are also presented. [DOI: 10.1115/1.4051519]

Keywords: mechanism design, mobile robots, robot design

Introduction

With the increasing demand for versatile robotic platforms, the need for adaptable robotic structures is on the rise. Robotic systems are required to perform a variety of tasks in diverse and uncertain environments. In this regard, modular self-reconfigurable mobile robots have emerged as a practical solution over the past decade.

Modular systems [1–7] allow for the advantages of long-chained [8–10] or humanoid [11,12] configurations while minimizing the complexity of each system [5,8]. This allows the overall system to bring about the versatility, stability, and increased spatial workspace of larger systems while retaining the flexibility of small robotic systems. Mobile robots have a higher locomotion speed [9,10,13], modularity [14,15], and load-carrying potential [16–21] in comparison to arrays [22] or lattice-based [23–25], homogenous [26], and heterogeneous [23,27] modular robots, which propels the research interest in mobile robotics. The use of mobile robots allows for the control of sensor-equipped [24–30] individual modules while tolerating misalignments on rugged terrain [5]. However, the lack of multidirectional mobility [16,17] limits the misalignment tolerance during docking operations. Therefore, a self-reconfigurable mobile robot with multidirectional mobility (STORM) [31] was developed to address these specific issues. The STORM system consists of two independent symmetrically invertible robotic modules. The hybrid-wheeled module has multidirectional mobility, and the

Manipulation module is capable of lifting and manipulating payloads. Detailed comparison of different modular robotic systems, their existing limitations, and addressing these limitations through proposed STORM modules is discussed in the past [11,12,31].

Active docking is the key feature that enables modular reconfigurable robotic systems to form a wide variety of configurations. Existing modular robotic systems use bi-gendered [32], gendered [2,3,33–35], or genderless [36] couplers (as shown in Table 1) to actively attach with other robotic modules, allowing them to be more versatile in comparison to rigid-structured robots. A recent review [40] of the various coupling mechanisms provides a comprehensive analysis of the various coupling techniques, their associated sensing methodologies, and existing limitations. Based on the inferences drawn from this review, the genderless high-strength efficient fail-safe high misalignment tolerant (GHEFT) mechanism [36] was designed. The two degrees-of-freedom (2-DOF) GHEFT mechanism can handle misalignments in all six-axis (*X–Y–Z–Roll–Pitch–Yaw*) with an ability to couple from either of the robots in case of multi-robot interactions where one of the coupling mechanism is facing a malfunction. The key differences like the ability to handle greater loads, non-back drivable coupling, and so on make this mechanism an ideal choice for the integration with the self-reconfigurable modular robot, which is also the major focus of this work.

While the previous research focused solely on the individual testing and validation of STORM [31] and GHEFT [36] modules separately, this paper describes the design and integration of a new STORM robotic module with a built-in docking mechanism. The proposed design also addresses the limitations and extends the capabilities of the previous STORM [31] module. This research will also serve as a motivation to analyze autonomously generated multi-robot assemblies, operating in unknown environments, as a part of future research.

Table 1 Classifying robots based on the docking mechanisms

		Robot	Sensor	Coupling	Coupling type	Fail safe
Robot type	Miniature	M-TRAN [6]	Infrared (IR), camera	Hooks	Gendered	No
		SAMBOT [33]	IR	Hooks	Gendered	No
		MBLOCKS [32]	IR, Hall effect	Magnetic linkage	Gendered	No
		Telecubes [37]	IR, Magnetic	Magnetic linkage	Bi-gendered	No
		Catoms [38]	IR	Magnetic linkage	Genderless	Yes
	Macro	Transmote [39]	IR, Angle/Tilt	Lock and key	Gendered	No
		Tanbot [13]	Camera, IR, Gyro	Pin and hole-latch	Gendered	No
		JL II [5]	IR, Global positioning system (GPS), Ultrasonic	Hooks-gripper	Gendered	No
		Polybot G2 [34]	IR, Hall effect	Pin and hole-latch	Bi-gendered	No
		CONRO [2]	IR	Pin and hole-latch	Gendered	No
		Swarm-bots [19]	Camera, IR	Hooks-gripper	Gendered	No

In addition to the modular robotic platform and the active docking mechanisms, reliable autonomous docking requires effective detection and tracking of the target module. The majority of the existing work in autonomous docking relies on close-range proximity sensors [3,30,33], restricting the search-area coverage of each independent module and the available workspace (as shown in Table 1). While vision-based approaches have been used in Refs. [41–43], they are limited by the target-pattern scalability resulting in the detection of false positives. To overcome the limitations, a knowledge-based approach was developed and presented in this paper. While the previous publication presented the above approach in a premature phase, this work demonstrates its applicability in real-world conditions. The hybrid target tracking (HTT) algorithm is made to run onboard the newly integrated STORM modules using the integrated sensor hardware. The output of the HTT algorithm is then given to the Image-Based Visual Servoing (IBVS) [44,45] controller also running onboard the robot to demonstrate autonomous docking capability in real-world conditions.

The major contributions of this work can be summarized as follows:

(1) A novel mobile robot, having hybrid locomotion capability (track-based and wheel-based) and an integrated 2-DOF active docking mechanism (GHEFT).
(2) The autonomous docking of the proposed robot is validated experimentally, using the HTT algorithm and IBVS control.

The overall layout of the paper is presented as follows; the motivation for a self-reconfigurable robotic module with an integrated docking is presented in Sec. 2, followed by a detailed discussion on the proposed robot design and the related hardware. An overview of the Hybrid Target Tracking algorithm used to aid autonomous docking is presented in Sec. 4, followed by experimental validation using IBVS in Sec. 5. Section 6 concludes the paper summarizing the contributions of this work and inferences drawn from the experiments. Directions for future work are also presented in this section.

For additional details on existing modular robotic systems and docking mechanisms, as well as the advantages brought out by STORM and GHEFT systems, interested readers are referred to prior research [31,36,46–49].

Application Overview and Related Challenges

The motivation for the docking integration attributes to the research in reconfigurable robots providing better flexibility, mobility, and manipulation (shown in Fig. 1). Most of the positioning-based inter-robot docking takes place in a close-proximity range using infrared (IR) sensing [6] and ultrasonic [33,37–39] sensors. However, sensory failures can accumulate in real-world situations due to the presence of obstacles. A vision-based technique using blinking patterns in Ref. [6] can be problematic under the presence of a similar light source in its surroundings. The lack of vision to recognize the target becomes a shortcoming for this type of autonomy. The development of the robots using smart cameras [7,13] and the simultaneous localization and mapping (SLAM) based approach [4] can overcome such limitations. However, the inability to carry payload limits their application versatility due to their small size. A combination of camera and tracked-based mobility [13] tries to highlight the in-field application requirements. However, the robot is limited by its locomotion capabilities, and chain-like formation results in higher misalignments resulting in less effective reconfiguration. Moreover, the use of color segmentation can be erroneous due to changing scalability, and the presence of the target pattern in an intermixed environment. A comparison presented in Ref. [48] lists the challenges of using a target-tracking algorithm on a low-cost onboard computer. Additionally, autonomous docking requires that the modular robotic systems be able to navigate autonomously toward each other, such that the two docking mechanisms are within a threshold distance of each other [1,43]. These shortcomings have led to the development of a reconfigurable robot called STORM [31], a docking mechanism called GHEFT [36], and the HTT tracking algorithm [49]. STORM has two independent modules, manipulation, and locomotion (as
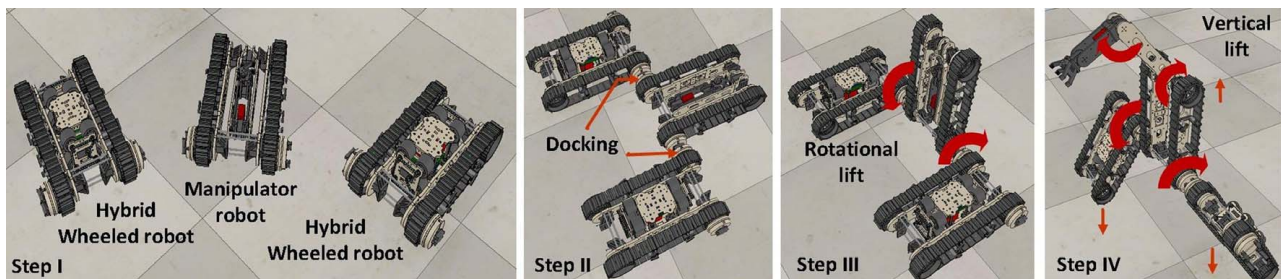

Fig. 1 Step-wise robot docking simulation for humanoid configuration performed in V-REP using STORM hybrid-wheeled and manipulator modules
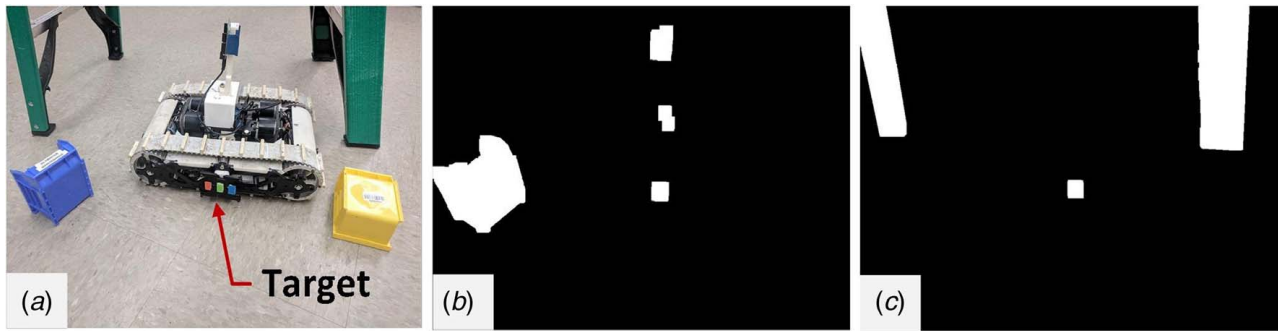
Fig. 2 Image thresholding for colors in the image similar to (b) 3rd marker point and (c) 2nd marker point, for an image shown in (a). Failed target detection due to multiple occurrences of the same color.

shown in Fig. 1). The use of bidirectional mobility helps to minimize the positioning error in locomotion while the use of a docking mechanism helps with eliminating the errors resulting from sensor detections. The robot is interfaced with a 2-DOF, high-misalignment tolerant mechanism [36] capable of handling high stress under multi-robot assembly.

The selection of the GHEFT mechanism based on literature review [40] is done for the following three main reasons:

(1) *Genderless and Fail-Safe*—The Genderless feature enables the mechanism to actuate from either inside or outside along with bidirectional coupling. Such a feature is a crucial factor in the case of malfunctioning, as compared to its relevant counterparts, namely gendered and bi-gendered mechanisms.

(2) *High Strength*—The mechanism can tolerate 34% higher payloads and 6% higher moments. The distribution of stress through a large surface area of the clamp makes it reliable under higher loads compared to shafts, pins, or hook-type mechanisms. Moreover, the non-backdrivable nature of the dual spiral grooved cam makes it efficient under such instances.

(3) *Tolerating Higher Misalignments*—Since sensor and control errors can get accumulated due to measurement uncertainty, a docking module should be capable of tolerating such misalignments. It can tolerate higher misalignments along with 6-DOF motion compared to the other mechanism as highlighted in this review.

One of the major challenges is that the localization accuracy achievable in field robotic applications is not sufficient for autonomous docking. As mentioned in previous work [49], visual feedback is more reliable as compared to the various proximity-based sensing techniques, specifically for field robotic applications. Vision-based detection and tracking allow the robotic modules to come together without relying on absolute positioning methods. However, vision-based techniques have their own set of challenges. Detection and tracking can be difficult when the target is intermixed

Table 2 Sensing methodology used in robots for docking

Technique	Limitations
Feature-based detection [4,13]	Lack of ability to correctly identify targets in uncertain environment and low frame update
Proximity-based sensing [2,5,6,32–39]	Small operating range, identification of obstacles in the docking path
Omnidirectional camera [28,30,43]	Lack of flexibility for upside-down robot configuration
Markers [4,50]	Large size of the markers in order to achieve and reliable accuracy
Deep-learning object detection [51]	High computational cost, and low frame rate (1–2 fps) on low power computing devices

with its surroundings (as shown in Fig. 2). Using markers for positioning [50] can be unreliable under low-lighting conditions. The deep-learning algorithm-based object detection [51] is not feasible on a low-cost architecture, as it delivers low-fps of approximately 1–2 fps on the single-board computer (SBC) like Raspberry-Pi, Odroid, and so on. Such a frame rate is not reliable when high-speed tracking is required for real-time applications. As such, color-based target tracking [13] is a commonly used approach due to its low computational expense and reliable tracking under variable lighting conditions, often along with an active visual marker. To summarize, Table 2 lists the limitations of the feature-based techniques and sensing methodologies used by self-reconfigurable robots existing in literature based on their ability to do onboard processing and traverse in an uncertain environment.

To address the above challenges, the HTT algorithm [49] capable of tracking a multi-colored marker is used for the autonomous docking process. The techniques assume that one of the robotic modules (target module) is stationary and the other (source module) moves towards the target module to achieve autonomous docking. In addition to providing reliable detection and tracking, the technique has a low computational cost allowing it to be run onboard the limited computational capabilities of modular robotic systems. The method combines knowledge-based detection with dual tracking techniques to prevent the loss of target features. Note that while this method was presented in previous work [49], its applicability in real-life autonomous docking was not demonstrated. This paper introduces the various components needed for autonomous robot docking namely STORM with integrated GHEFT and HTT algorithm and presents the experimental validation of autonomous docking in real-world conditions.

Mechanical Design and Hardware Integration

The electromechanical design of the proposed robotic module and the integrated 2-DOF active docking mechanism is presented in the following sub-sections. The four sub-sections discuss (a) main robot design, (b) integrated docking mechanism, (c) hardware and onboard sensing, and (d) actuation units of the robot. The robot design section will talk about the main robot design, its hybrid mobility, and the changes made to the existing robotic design to incorporate the active docking mechanism. Furthermore, the hardware section will also touch on some aspects of the computing power and related sensors, which are later used to do onboard vision-based tracking of the target robot relative to the source robot.

Modular Robot. Even though initial prototypes of STORM modules [31] and GHEFT mechanism [36] have been built separately, a modified prototype was built to integrate the GHEFT mechanism and test autonomous docking capability. The robot itself consists of hybrid mobility, where a track-based and wheel-based mechanism can be actuated separately, to make locomotion adaptable to an unknown environment. Furthermore, this adds
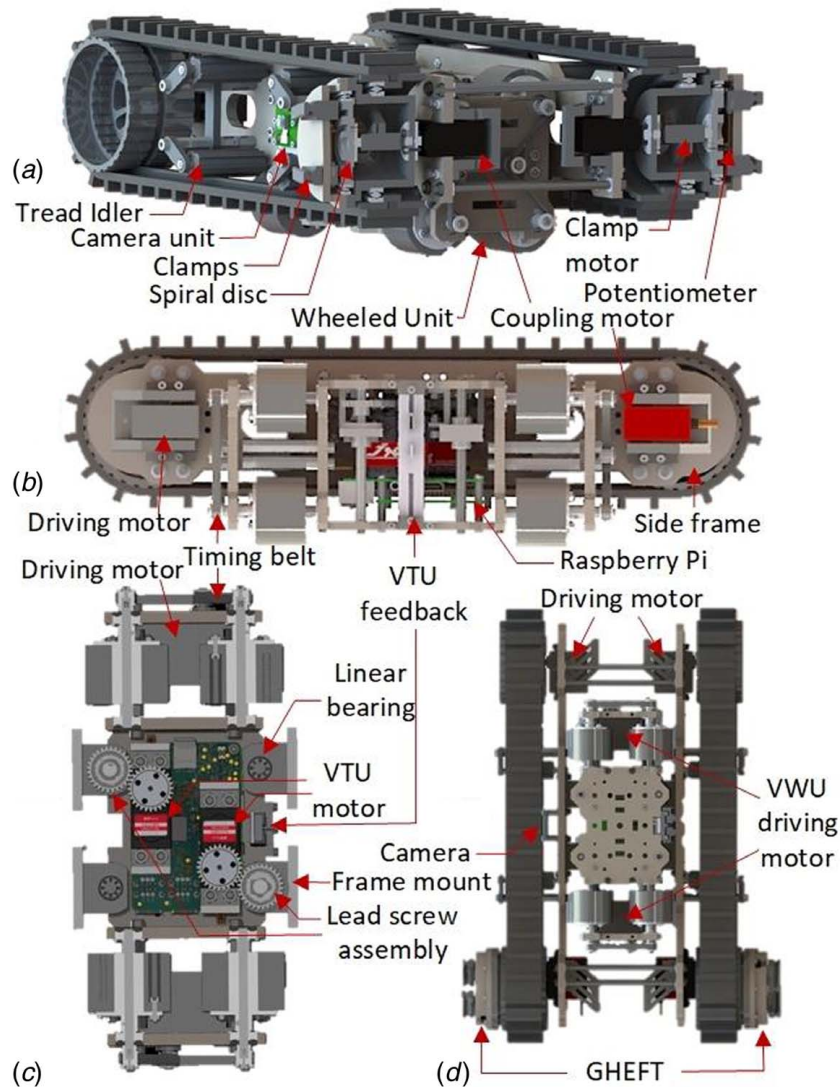
Fig. 3 Computer-aided design (CAD) illustration of the STORM robot: (a) isometric cut-section view of the front side showing the inner side of the GHEFT mechanism, (b) the side cut-section representing Vertical Translation unit (VTU) and the wheeled assembly of the robot, (c) the architecture of the VTU exposing the dual actuation of the lead-screw assembly, and (d) top view of the robot showing the overall actuating positions of the robot

convenience to the longitudinal and transversal mobility of the robot in densely packed environments. Each of the mobility modes, either track-based or wheel-based possesses independent 2-DOF. There is a constrained motion about the prismatic joint (lead-screw assembly) connecting the track-based and wheel-based mechanisms which help to actuate either track-based or wheel-based mechanisms at once. As shown in Fig. 3, each side of the track-based assembly is controlled using two independent motors to make it easily drivable on uneven terrains. Similarly, the wheel-based is a belt-driven mechanism that is controlled using two independent motors. This robotic module is referred to as a source and target robot interchangeably throughout the rest of the paper. The robot possesses an overall wheelbase (L), track width (W), wheel radius (r), and a total height (H) of 310 mm, 250 mm, 40 mm, and 92 mm, respectively. The weight of the robot is 1.75 kg with all the parts manufactured using three-dimensional (3D) printing material. As compared to the previously developed STORM robotic module, the vertical translation unit (VTU) assembly in the new version presented in this paper was modified to house two motor-driven lead-screw mechanisms compared to one as

shown in Fig. 3(c). The motion of this VTU assembly is controlled using the feedback from the potentiometer, which measures the positioning of the wheel-based assembly relative to the track-based assembly. This makes it easier to actuate one of the wheel-based or track-based assemblies when performing the autonomous docking to generate multi-robot assembly. A camera has been attached to the side frame of the robot to detect the target robot using the tracking algorithm discussed in the later section of this paper. The feedback of each actuator is recorded using a positioning sensor to autonomously control the source robot relative to the target robot. These updated features will be discussed in detail in the following sections.

Docking Mechanism. The modified STORM module was integrated with a GHEFT active docking mechanism (Fig. 4). The docking mechanism has 2-DOF; the case rotation and the non-backdrivable spiral cam mechanism for the motion of clamps. The overall mechanism provides a maximum clamping force of 38.8 N and rotational torque of 2 Nm (using the calculations in
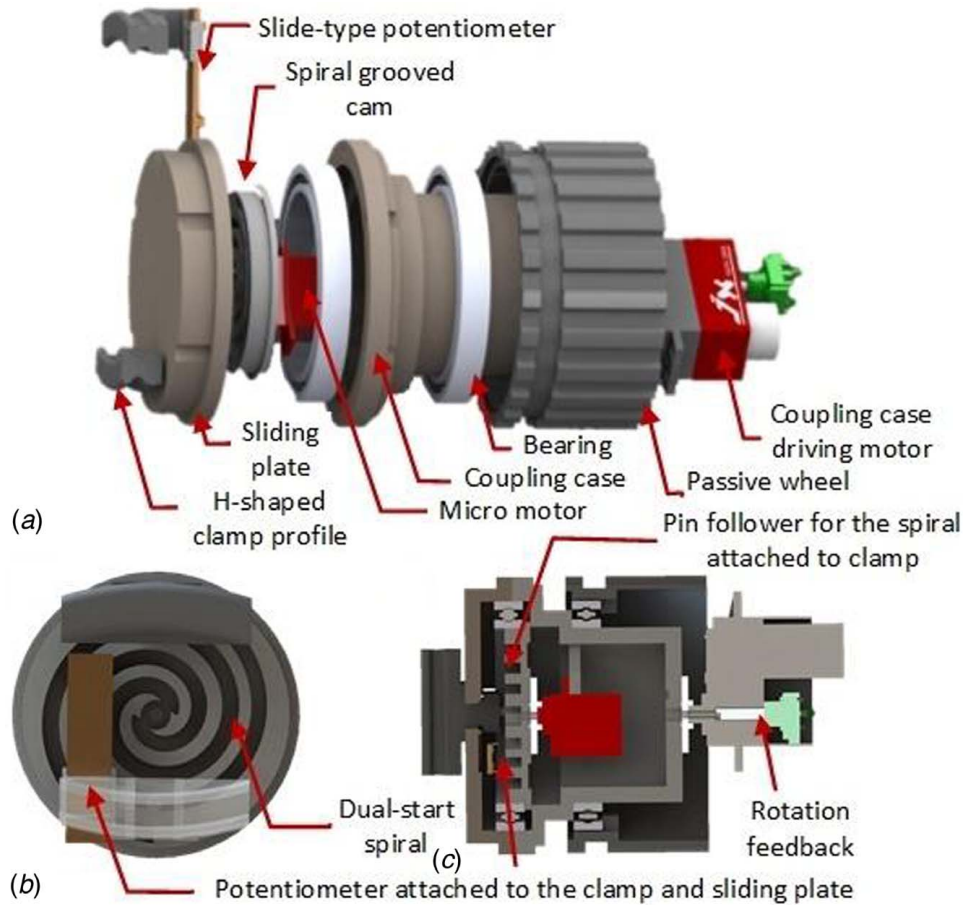
Fig. 4 (*a*) **Exploded view of the docking mechanism,** (*b*) **spiral grooved cam profile, and** (*c*) **compact view of the joint assembly**

Ref. [36]). Similar to STORM, the GHEFT mechanism was also modified from the previously presented version (Fig. 4). The new GHEFT mechanism was reduced down to 65% of its original size (Fig. 5(*a*)) to make it proportionate to the size of the robot. The number of starts of the spiral grooved cam for the clamps [36] was increased from 1 to 2 (as shown in Fig. 4(*b*)) to provide equal translation for both clamps and eliminate the offset in case of a single spiral. The spirals are constrained with a geometric maximum (D_{max}) and minimum diameter (D_{min}) of 50 mm and 6 mm, respectively. The slip-ring used for the electrical contacts

in the original mechanism has been eliminated to restrict the motion of the case to one complete rotation in either direction. This limit was implemented using the encoder feedback attached to the motor of the case. This change was made considering the different multi-robot assemblies, which can be achieved. Furthermore, considering the small size of the robot, the worm-gear assembly to drive the case has been replaced with a direct-drive input using a high-torque motor. A slide-type potentiometer with a maximum travel length of 20 mm is attached to the inner side of the sliding plate with one of the clamps to provide motion feedback when
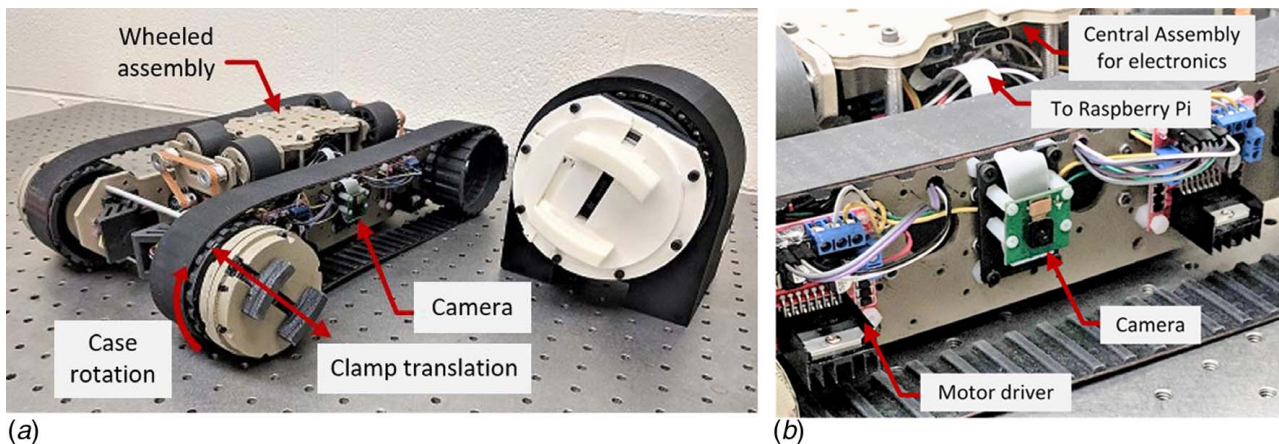

Fig. 5 (*a*) **STORM robot interfaced with 2-DOF prototype compared with the original GHEFT mechanism (size reduction by 65%);** (*b*) **onboard camera of source robot**
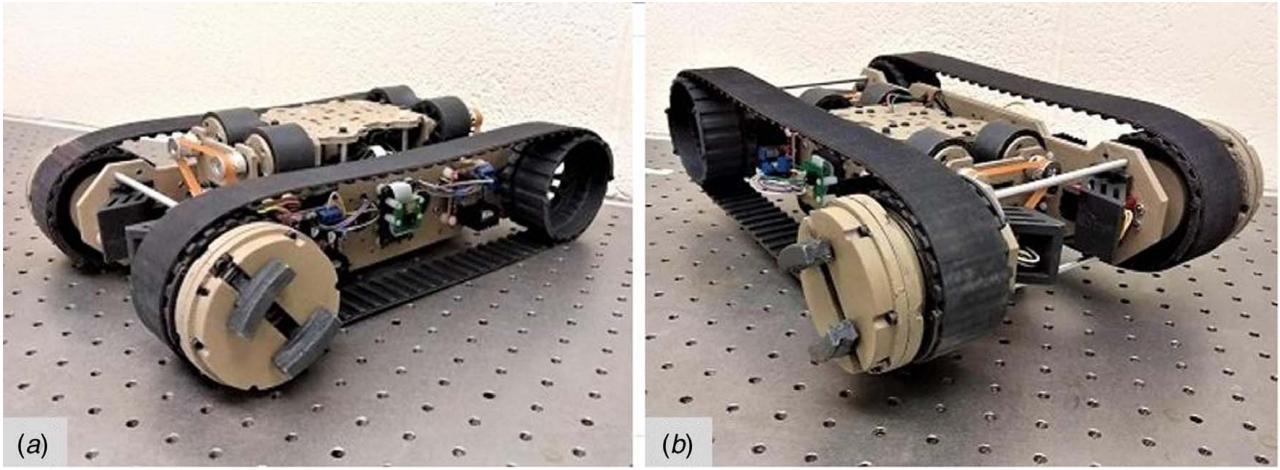
Fig. 6 STORM robot: (a) track-actuated, (b) wheeled-actuated (or Upside down) configuration

coupling with the target robot. The mechanism design of the GHEFT mechanism interfaced with the passive wheel of the robot is shown in Figs. 6(a) and 6(b), having the ability to be operated independently about a ball bearing contact. The rotary motion of the case is recorded using an encoder attached to the motor having a 20 pulse per revolution (PPR) resolution. The high misalignment tolerance capability of the mechanism clamps helps to overcome the low-resolution value of the encoder.

Onboard Sensing and Processing. The HTT algorithm and IBVS control were implemented onboard the robot to enable autonomous docking. The robot is equipped with multiple sensors and a processing unit to do onboard processing for the positional feedback provided by these sensors. The onboard image processing is done using a Raspberry-Pi 3 Model B SBC. The SBC is connected to a Teensy 3.2 microcontroller to control the direct current (DC) motors over serial communication. An Arducam 5-MP camera (shown in Fig. 5(b)) is attached to the side frame of the robot to give visual feedback of the marker attached to the target robot. The displacement of the H-shaped clamps (x_s^c, x_t^c) of the GHEFT coupling mechanism is measured using a slide-type potentiometer (Fig. 4(b)) with a travel length of 20 mm. Here, the use of s and t subscripts represents the source and the target robot, respectively. Another slide-type potentiometer with a travel length of 60 mm is used to measure the vertical displacement (x_s^v) of the vertical

translation unit (VTU). A rotary encoder is attached to the shaft of the coupling motor (θ_s, θ_t) to measure the angular position of the coupling case of both the robots. The misalignment tolerance capability of the GHEFT mechanism accounts for the low resolution of the coupling encoder. An MPU-9250 inertial measurement unit (IMU) sensor is used to keep track of the *Roll–Pitch–Yaw* angles (α, β, γ) of both robots. The built-in Wi-Fi of the SBC is used to create a transmission control protocol and internet protocol (TCP/IP) network to share the sensor data between the target and the source robot (α_s, α_t, β_s, β_t, γ_s, γ_t, x_s^c, x_t^c, θ_s, θ_t). This sensor data sharing, except the use of a camera, helps to perform the Pre-IBVS positioning and orientation of the source robot relative to the target robot, which is discussed in the later sections of this paper.

Actuation. The motion of the tracked unit and the coupling rotation of the docking mechanism are achieved using a high-torque DC motor whereas the lateral (wheeled) motion of the VTU is controlled using a high-speed DC motor. Each motor is equipped with an encoder to get rotational feedback and is controlled using the L298N motor driver. The maximum and the minimum velocity of the module are 0.345 m/s and 0.155 m/s, respectively. The wheeled assembly is driven using two pairs of a four-wheel timing belt mechanism to actuate each side independently. Each pair possesses an upper and lower velocity limit of 0.265 m/s and
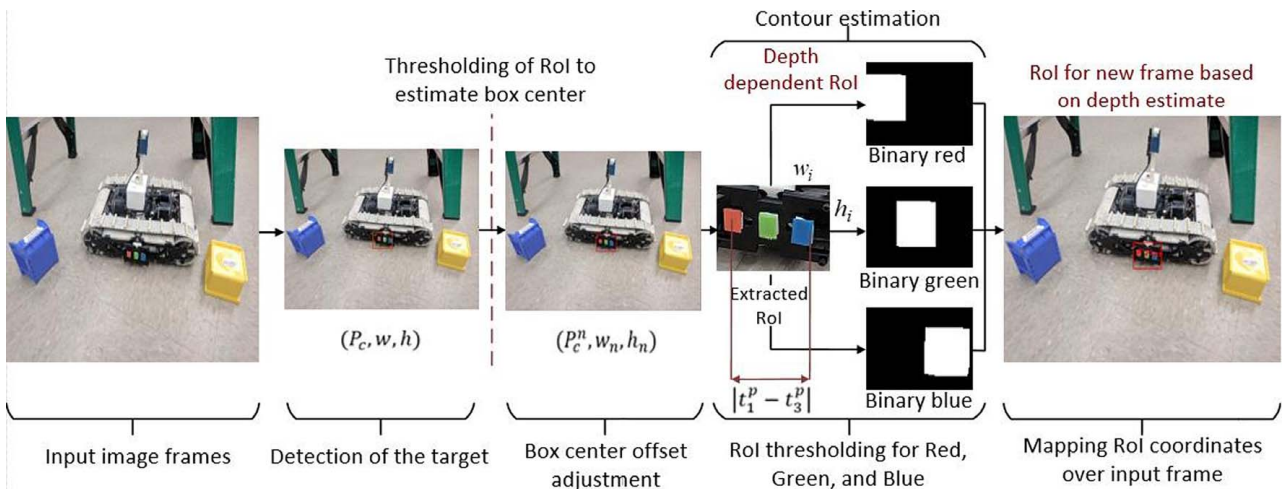

Fig. 7 Hybrid Target Tracking (HTT) algorithm with a step-by-step image formulation, representing offset adjustment and depth-dependent RoI. The input takes the visual feedback from the camera resulting in a feature estimation on each consecutive image using the HTT algorithm.

0.22 m/s. The motorized lead screw ($d_{thread} = 6$ mm, *Pitch* = 2 mm) and passive shafts help in guiding the mechanical motion of the VTU through a brass nut and a linear bearing ($d_{bore} = 6$ mm) on either side of the mechanism. The velocity of the lead-screw mechanism is 0.1 m/s and is constant throughout the actuation. The clamping mechanism is controlled using a high-torque DC motor with feedback from the clamp potentiometer. The implementation of the velocity control under IBVS has been done using a dual H-bridge motor driver (L298N) having a maximum current rating of 2 A per channel.

Hybrid Visual Marker Tracking

To track the target robot relative to the source robot, Hybrid Target Tracking (HTT) algorithm is used (shown in Fig. 7). The algorithm presented in Ref. [49] has been modified to make it adaptable to the changing environmental conditions for real-world validation. The template-based approach of the detection-phase of the algorithm has been replaced with a learning-based approach. The HTT algorithm overall has two stages, namely, detection using YOLO-tiny (You Only Look Once) [51] and tracking using color segmentation (CS) with optical flow (OF) [52] running in parallel on the detected Region of Interest (*RoI*). The use of only CS tracking can fail under occlusion while the use of only OF-based tracking can incur drift over a long time. However, a combination of both the tracking methods running together in parallel eliminates the drawbacks of each method. This allows achieving robust tracking of features at a significantly higher frame rate [49]. A visual markers red, green, and blue (LED_1, LED_2, LED_3) are used as a target, which also acts as a training input for the detection model. The goal is to leverage the centroid information of each feature of the colored target relative to the original image frame, $I(m, n)$.

The detected *RoI* (with the center, width, and height as P_c, w, and h, respectively) is selected based on the parameters of the YOLO algorithm such that, $RoI \subset I$. The selected *RoI* is then filtered (erosion (ε_B) followed by dilation (δ_B)) to find the center of the target using Eq. (1) to handle any offsets during initial detection. This can be written as

$$\mathcal{E}_B(\mathbf{T}) = \mathbf{T} \ominus \mathbf{B} \triangleq \{x : \mathbf{B}_x \subset \mathbf{T}\}$$
$$\delta_B(\mathbf{T}) = \mathbf{T} \oplus \mathbf{B} \triangleq \{x : \mathbf{B}_x \cap \mathbf{T} \neq \varnothing\} \tag{1}$$

where **B** is a structuring element and **T** is the threshold image of the *RoI* formed after conversion to Hue, Saturation, and Value (HSV) colorspace. If $O_f (= (u_{of}, v_{of}) \in I)$ defines the pixel offset from the center as per Eq. (2), then the new center of *RoI*, P_{new}, can be written as

$$P_{new} = P_c + (u_{of}, v_{of})$$
$$\frac{w_n}{w_t} = \frac{h_n}{h_t} = \frac{d_2^p}{d_2} = k \tag{2}$$

where w_n, h_n, w_t, h_t, d_2^p, d_2 are the new width and height of the bounding box, target width and height, blob diameter, and the center light-emitting diode (LED) actual diameter, respectively, and k is a proportional constant. The dimensions of the blob are used as a reference to create the initial bounding box with an applied offset. The new *RoI* ($RoI_{new} \subset I$) formed after applying the offset is then selected for CS-based tracking of the algorithm. Once the center of the extracted RoI_{new} has been located, the following equation can be used to obtain the updated velocity feedback for OF-based tracking

$$I(u_2, v_2, t) = I(u_2 + \Delta u_2, v_2 + \Delta v_2, t + \Delta t)$$
$$\mathbf{I}_{u2}\mathbf{u}_2 + \mathbf{I}_{v2}\mathbf{v}_2 + \mathbf{I}_t = 0 \tag{3}$$

where u_2, v_2 are the coordinates of P_{new} at time t, $\mathbf{I}_{u2}$, $\mathbf{I}_{v2}$ are partial gradients of the image with respect to u_2, v_2, and $\mathbf{u}_2$, $\mathbf{v}_2$ represent the flow velocity of the point. This terminology can be applied to

Table 3 Target detection and tracking

Input: Image frames using onboard camera
Output: Position of the bounding box and continuous tracking
1. Retrieve initial image, *I*, to locate target marker
2. Extract the bounding box using the trained detection model, (P^c, w, h)
3. Adjust bounding box offset to get (P_n^c, w_n, h_n)
4. Store (P_n^c, w_n, h_n) to *curr_point*
5. Initialization, video capture on
6. Initialization, *OpticalFlow* and *ColorSegmentation* to get t_p^i $\forall$ $i = 1,2,3$
7. while (True)
8. If, *prev_frame* is empty
    copy *curr_frame* to *prev_frame*
else
    Get $|t_p^1 - t_p^3|$ for d_{est}
    OpticalFlow, swap *curr_point* and *prev_point*
swap, *curr_frame* and *prev_frame*
end

each of the projected points (t_1^p, t_2^p, t_3^p) on the Image plane, I, as $P_1^{roi}(u_1^r, v_1^r)$, $P_2^{roi}(u_2^r, v_2^r)$, $P_3^{roi}(u_3^r, v_3^r)$. The final bounding box for the tracking is created by comparing the width of the blob at the center and the distance between the end blobs ($d_{est} = |t_1^p - t_3^p|$). The change in the shape of the bounding box with the change in distance is managed by multiplying a proportional constant k (calculated from the initial depth estimate) with the width and the height of the box. The final set of the tracking points can be written as follows:

$$P_i^I = P_c^n - \left(\frac{w}{2}, \frac{h}{2}\right) + P_i^{roi}, \ \forall \ i = 1, 2, 3 \tag{4}$$

where w, h, P_i^{roI}, and P_i^I represent the instantaneous width and height of the *RoI*, the coordinate relative to the *RoI* and image plane, respectively. The pseudo-code for the tracking algorithm is shown in Table 3.

Performance Analysis of the Algorithm. In addition to the image processing results presented in Ref. [49], some additional experimentation was carried out to validate the tracking consistency of the algorithm. The tracking consistency of the HTT algorithm has been validated by placing a colored marker rotating in a circular manner while placed at varying distances (0.3–1.2 m) from the camera. The camera was mounted on a vertical platform such that the optical center coincides with the center of rotation of the marker. The detected pixel values (approximately 3000 data points per depth-mark) of the marker during motion (in pixels) were recorded and plotted along with the distance values (in cm), to get the pixel-level accuracy. These tracking data were also compared against the expected trajectory of the target, as shown in Figs. 8(*a*) and 8(*b*).

The mean error plot for the tracked point and the estimated depth from the camera based on the calibration parameters are shown in Fig. 9(*a*). The error in the tracking of OF center and CS center is compared to the deviation in the flow estimate of the bounding box with the detected marker center. The high variation (±4 pixels for CS at 110 cm) in error is primarily due to the limited resolution (320×320) of the camera, which leads to a low pixel area at larger distances. The results for the performance-related experimentation are shown in Table 4, to compare the actual depth with the estimated depth from the algorithm. The use of OF with CS provides the following advantages:

Tracking. In cases where the color tracking fails, OF consistently tracks the *RoI* window to avoid failure. This is not applicable if *RoI* is formed using only the color tracking approach (Fig. 9(*b*)).

Performance. The loss in *RoI* due to failure in color tracking causes a repeated search of the colored target in the overall image
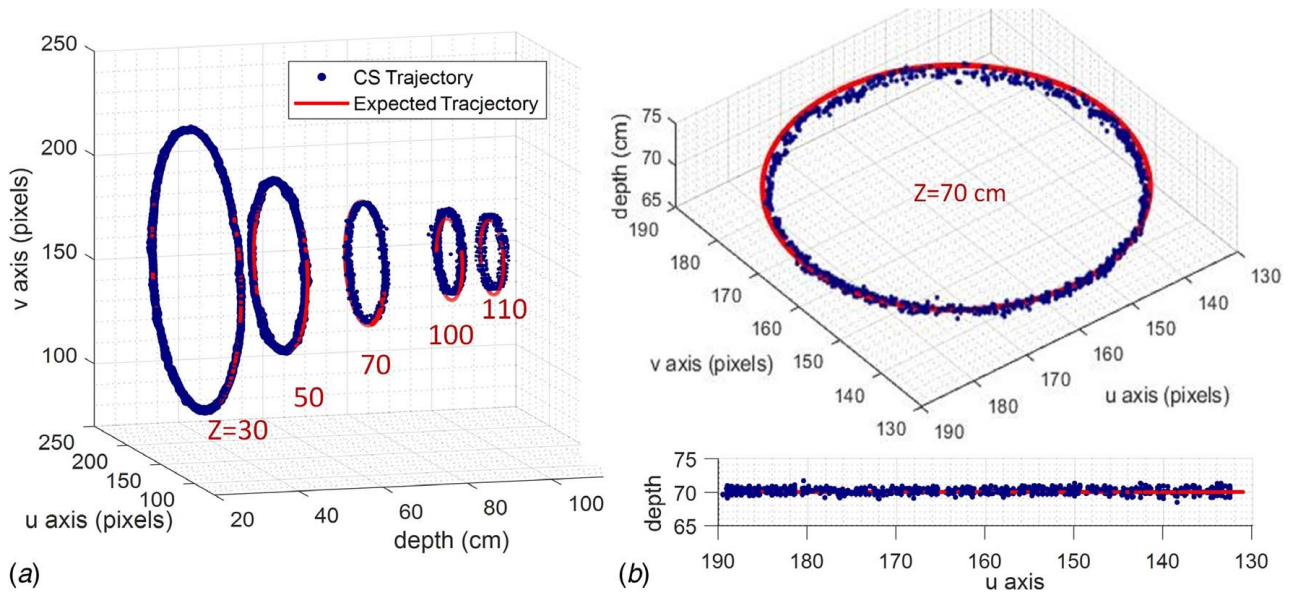
Fig. 8 Performance results of the HTT algorithm: (*a*) final trajectory formulation based on the bar setup placed at different depth values and (*b*) multiple views when the bar was placed at 70 cm
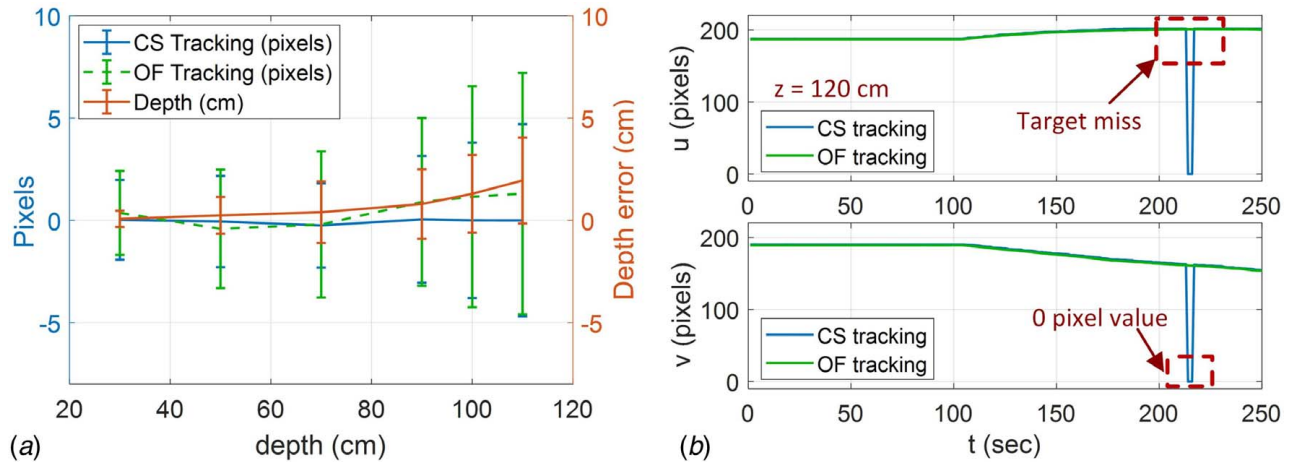

Fig. 9 Performance results of the HTT algorithm: (*a*) mean error plot for the tracking data and (*b*) importance of optical flow (OF) for consistent tracking where a color segmentation (CS) technique fails to detect any particular target

plane if the OF is not used for *RoI*. This reduces performance and increases failures.

Implementation and Experimental Results

The experimental validation for autonomous docking using the proposed robot prototype combined with the HTT algorithm is presented in this section. In the beginning, the experimental setup is discussed to discuss the position of the source and the target robot in the environment and the type of external sensors used to measure the trajectory of the robot. This section is followed by a discussion about adjusting the position and orientation of the source robot relative to the target robot, using the onboard sensor data before using the HTT and IBVS technique. Afterward, the tracking of the target robot relative to the source robot is discussed using the HTT algorithm, which further helps to perform the IBVS-based locomotion and is concluded with a discussion using the experimental results.

Experimental Setup. As mentioned before, a colored visual marker is attached to the side frame of the target robot. The camera attached to the side frame (shown in Fig. 10) of the source robot is used to track the targets to proceed with autonomous locomotion. As mentioned in the Introduction, the autonomous docking stage happens after initial coarse alignment using localization information of the source and target robots. This stage brings the target robot within the camera field of view of the source robot.
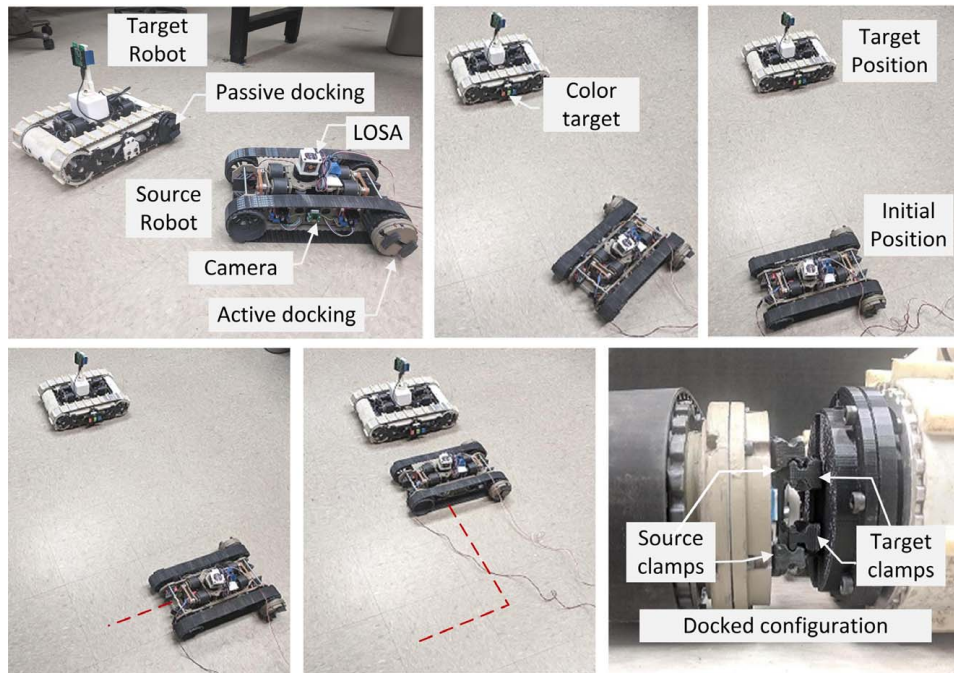
Table 4 Performance results analysis for HTT algorithm

Actual depth (cm)	Estimated depth (cm)	OF Estimation (pixel)	CS Estimation (pixel)
30	30 ± 0.5	± 2.1	± 2
50	50.2 ± 0.7	± 2.5	± 2
70	70.5 ± 1	± 3.4	± 1.8
90	91 ± 1	± 4	± 3.5
100	101.3 ± 1.2	± 5.5	± 3.5
110	112 ± 1.5	± 6	± 4

Fig. 10 Experimental setup for the IBVS demonstration followed by *Yaw–X–Y–Z* axis alignment and the clamping of the docking mechanism with the target robot initially placed at a distance of 60 cm from the source robot
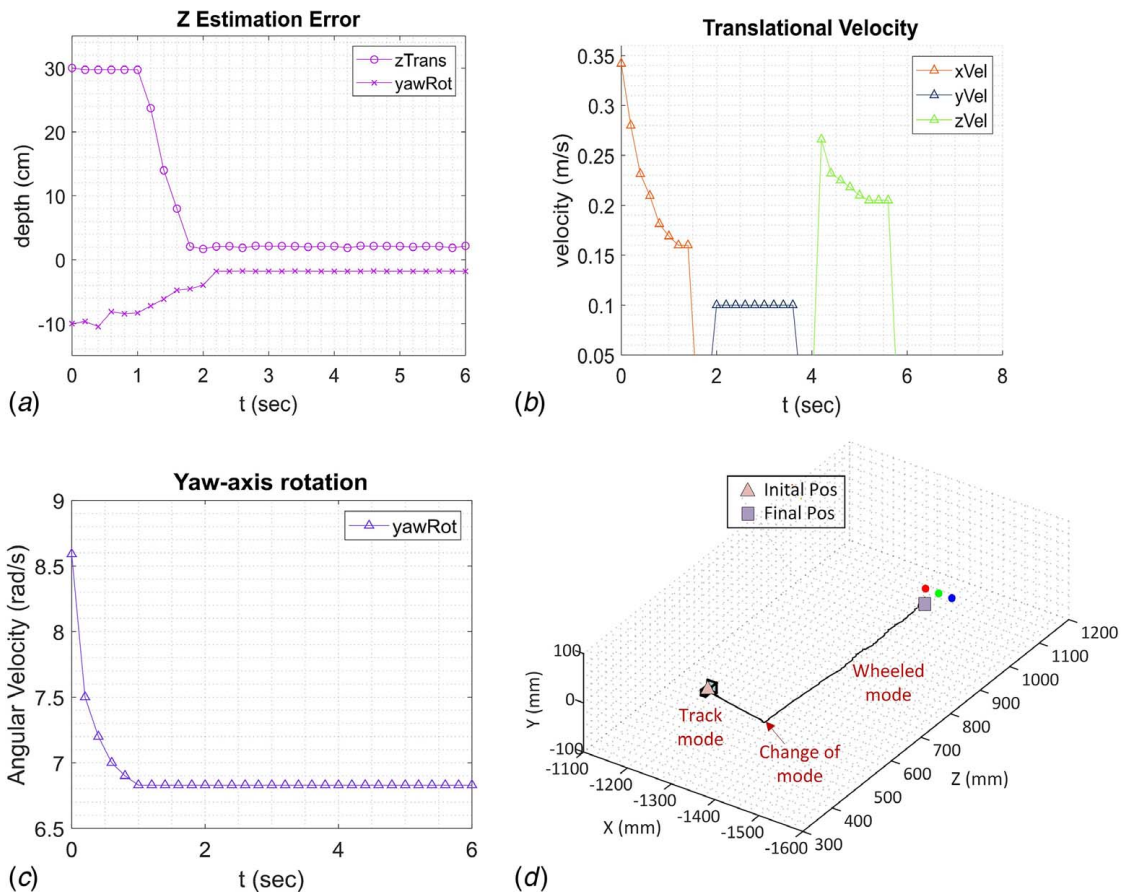

Fig. 11 (*a*) Error estimation plot based on the motion along the *Z*-axis and *Yaw* axes, (*b*) translational and (*c*) angular velocity curves for the overall docking procedure *X–Y–Z* axes, and (*d*) real-world motion trajectory tracking of the robot using LOSA, where a LOSA marker is attached to the top of the hybrid-wheeled locomotion robot
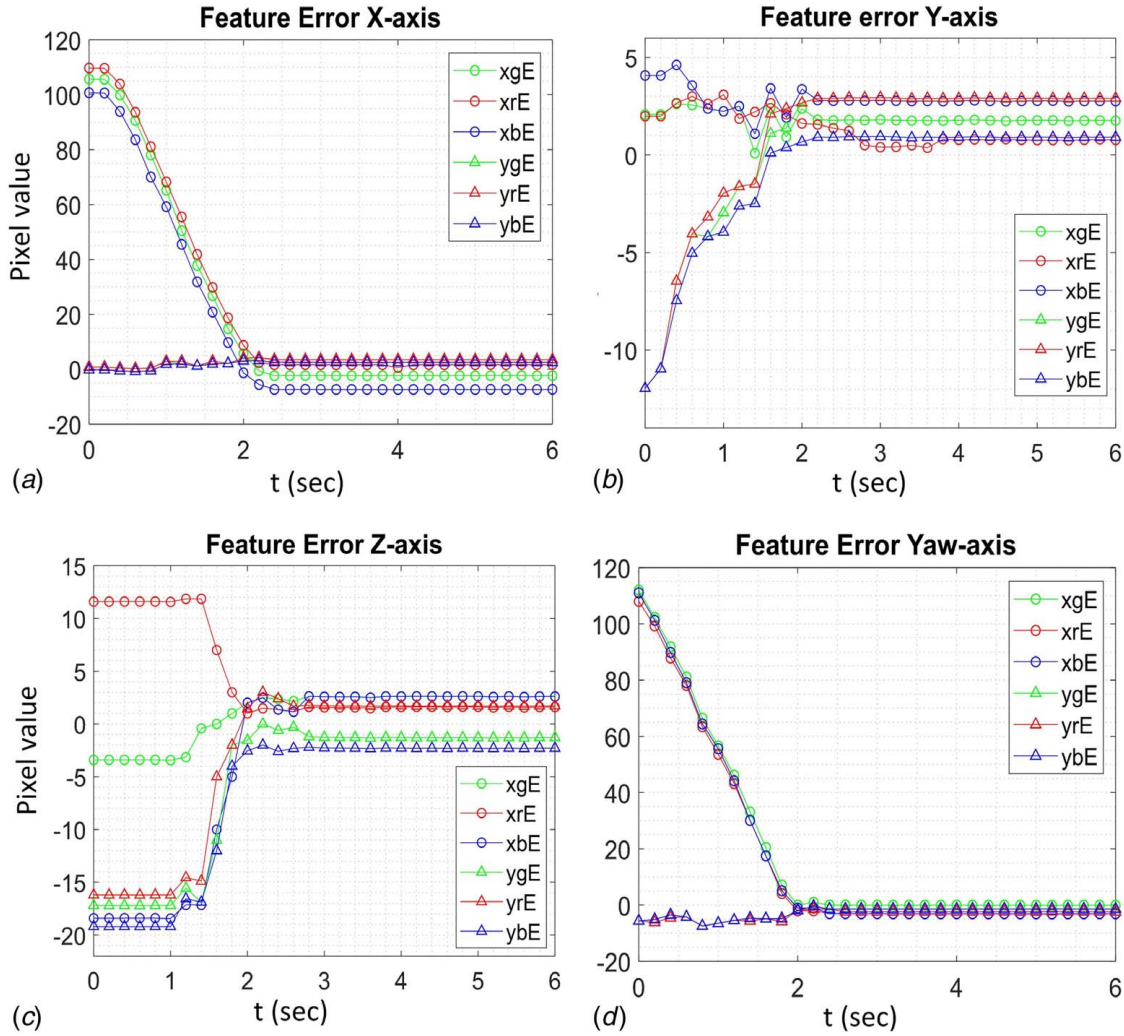
Fig. 12 Feature error plots for the motion of the source robot along (*a*) *X*, (*b*) *Y*, (*c*) *Z*, and (*d*) *Yaw* directions relative to the target robot

For this experiment, the source robot is placed at a distance of 250 mm along the *X*-axis and 600 mm along the *Z*-axis from the target and executes motion in the following order: $X–Y_i–Z_i–Y–Z$. Here, the *i* subscript represents the initial motion to position the robot at a certain distance from the target before minimizing the error along the *Y*- and *Z*-axes. Since error along the *Y*-axis cannot be minimized directly with the source robot placed 60 cm from the target robot, an initial estimate for the *Y*-axis (30 cm on x_s^v) is defined to enable the wheeled assembly by actuating the VTU. This actuation is defined by Y_i. The *Y*-axis error is minimized only when the robot is at an estimated depth of 30 cm using wheeled actuation under Z_i. This value (30 cm) was calculated based on the calibration runs to place the target features at the desired location in the image plane using the source robot. The accepted error range for motion along each axis is defined under the IBVS section.

The IBVS *Yaw*-axis motion of the robot is recorded separately to validate the orientation alignment in the case of errors followed by the implementation of the Pre-IBVS orientation using sensor data. The robot is placed with a *Yaw* angle of 40 deg and at a distance of 60 cm along with the *Z*-axis relative to the target robot, shown in Fig. 10. The IBVS control is initialized after orientation error based on the sensor data received from the target robot has been minimized. The 3D locomotion trajectory is tracked using an external sensor, light optical sensor array (LOSA), shown in Fig. 11(*d*). LOSA is an active IR marker-based tracking system developed by Kumar and Ben-Tzvi [48] capable of achieving millimeter-level

absolute positioning accuracy at a very high update rate of 250 Hz. The estimating accuracy varies from 1.18 mm to 50.56 mm for tracking a target placed at 2.1–5.4 m, respectively.

Pre-IBVS Orientation and Positioning. The Pre-IBVS orientation and positioning involve the alignment of the source robot relative to the target robot using the onboard sensor data. This helps to make the visual marker of the target robot within the visible field of view of the camera attached to the side frame of the source robot. The sensor data (IMU, clamp potentiometer, and coupling case encoder) from the target robot module $(\alpha_t, \beta_t, \gamma_t, x_t^c, \theta_t)$ are transferred to the source robot module using the TCP/IP communication, where one robot (target) acts as a sender while the other acts as a receiver (source). Additionally, it is assumed (as a part of prototype validation) that at the beginning of autonomous docking, the height difference (Δh) between two robots is no more than $x_s^v - [(x_s^c/2) + (x_t^c/2)]$. This limit helps to readjust any height-based differences between the source and the target robot using the potentiometer feedback attached to the hybrid-wheeled assembly. The clamp translation has to follow the geometric constraint, such that $D_{\min}/2 < x_s^c, x_t^c < D_{\max}/2$ with an offset of 5 mm due to the shape of the clamp. The accepted error range of ±5 deg for γ_s, ±1 pulse per revolution (PPR) for θ_s, and ±5 mm for h_s is used due to the low-speed actuation of the VTU. At first, the orientation of the source robot is adjusted about the *Yaw* axis relative to the target robot using the onboard IMU sensor data from both robots. The
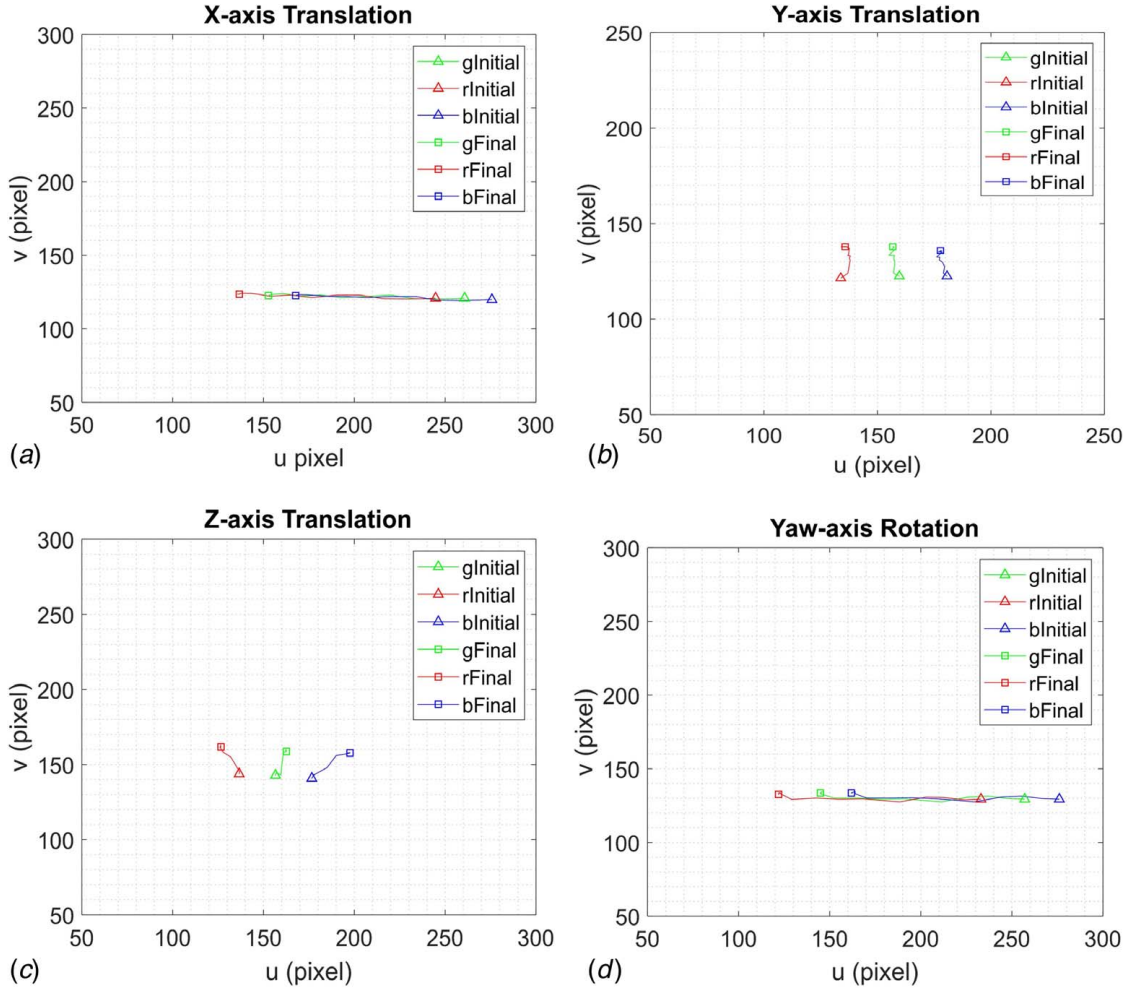
Fig. 13 The motion of the target features in the image plane as viewed from the camera on the source robot. The source robot is autonomously controlled to move from (*a*) *X*-axis, (*b*) *Y*-axis, (*c*) *Z*-axis, and (*d*) *Yaw* axis.

mechanism alignment about the *Yaw* axis is followed by error minimization for the coupling case angle and clamp potentiometer. This helps to prepare the coupling mechanism of the source robot available to couple with the coupling mechanism of the target robot. Apart from these common differences, there could be a relative height difference between the two robots due to the terrain, which can be minimized by the actuation of the VTU using the potentiometer sensor feedback attached to this assembly. The final error (Δe) minimization can be summarized as follows:

$$\Delta e = \text{minimize}(\Delta e^{imu}, \ \Delta h^c, \ \Delta \theta) \tag{5}$$

where Δe^{imu}, Δh^c, and $\Delta \theta$ are the error difference between the IMU sensor data, clamp potentiometer, and coupling encoder data, respectively. It should be noted that instead of the STORM modules being symmetric about all three axes (X, Y, and Z), the proposed autonomous docking can work even if either of the robots is flipped over (as shown in Fig. 6). Such a scenario can be realized when the robot is operating on uneven terrain. This type of positioning can be detected based on the IMU readings and accounting for it by reversing the actuation along that axis.

Alignment Using IBVS. Once the Pre-IBVS orientation and alignment are complete, the source robot proceeds with the Image-Based Visual Servoing technique using the HTT algorithm to complete the autonomous docking with the target robot. Based on a review of the widely used Image-Based Visual Servoing techniques [44,45], the classical IBVS technique was used in this work to

perform autonomous docking using the target-tracking feedback provided by the HTT algorithm (shown in Fig. 11). The robot in its tracked and wheeled mode behaves like a non-holonomic skid steer robot with velocities v_x, v_y, v_z, and ω_y and an ability to adjust its height ($0 < y < 60$ mm). The aim is to move the current feature projection $t^p = t^p_1 - t^p_2 - t^p_3$ of the target $t_1 - t_2 - t_3$ to the desired location $t^d = t^d_1 - t^d_2 - t^d_3$ in the image plane. The camera attached to the side frame of the robot projects the target features onto the Image Plane, I. Here t^p_1, t^p_2, t^p_3, t^d_1, t^d_2, and t^d_3 correspond to the world and pixel coordinates (u^p_1, v^p_1), (u^p_2, v^p_2), (u^p_3, v^p_3), (u^d_1, v^d_1), (u^d_2, v^d_2), and (u^d_3, v^d_3), respectively. The aim is to minimize the error (Δe_f), between the feature projection and the desired feature location in the image plane such that

$$\Delta e_f = \text{minimize}(t^p - t^d) \tag{6}$$

The velocity of the robot $[v_c, \omega_c]^T$, defined as $[v_x, v_y, v_z, \omega_x, \omega_y, \omega_z]^T$, corresponds to the feature error Δe_f where ω_x and ω_z are 0. The actuation for each locomotion mode of the robot is done step by step in the following order $X–Y_i–Z_i–Y–Z$ as shown in Fig. 11(*d*).

Experimental Results. The motion along each axis is actuated step by step (in the following order, $X–Y–Z$ and *Yaw* (shown in Fig. 12). The experimental results representing the motion of the features in the image plane are shown in Fig. 13. The top row highlights the feature error for motion along each of the
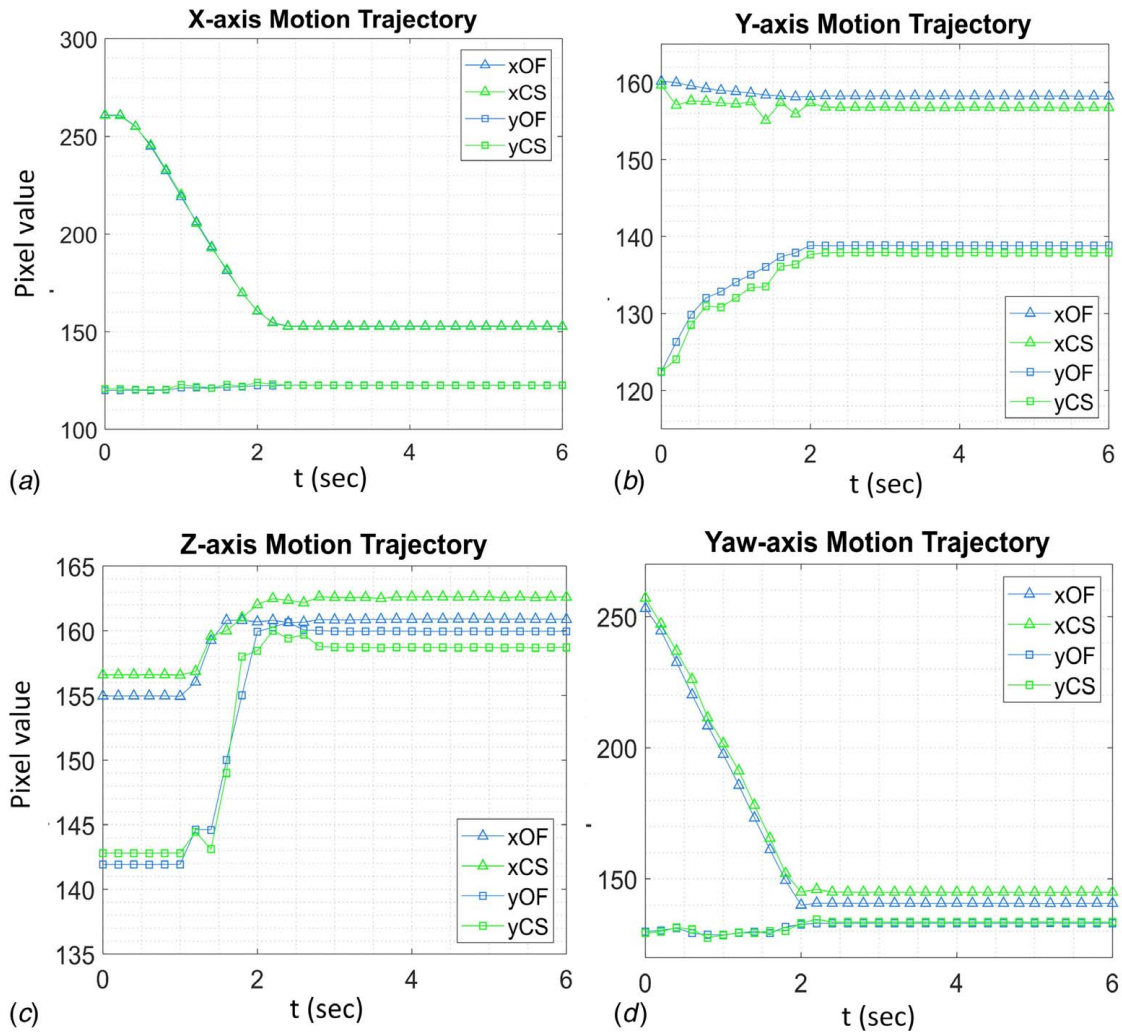
Fig. 14 Tracking trajectory comparison of the OF and the CS center for a motion along the (*a*) *X*-axis, (*b*) *Y*-axis, (*c*) *Z*-axis, and (*d*) *Yaw*-axis with an initial depth position of 60 cm

above-mentioned axes. The trajectory of the features in the image plane is presented in the next row. It can be inferred from the graphs that the error along each axis is minimized in under 2.5 s for a given setup, with an acceptable error range of ±5 pixels. The motion along the Z-axis is shown in Fig. 11(*c*) as estimated by the robot. Figure 14 compares the trajectory error between the CS and OF tracking of the center feature.

Furthermore, the whole docking procedure is performed in under 7 s, much faster than the case if only tracked actuation were to be used. The proof-of-concept setup used to validate the proposed autonomous docking procedure makes use of a low-cost electronic architecture to perform onboard detection and tracking, which also results in limited computational capabilities. The use of an external power source also limits the motion of the robot. The impact of the depth with the autonomous docking or the tracking of the target has already been shown in Fig. 9(*a*), where the mean error minimizes with the reducing depth value, which further helps to provide better accuracy with the tracking. The results[2] show that autonomous docking could be achieved using the STORM multi-directional mobility robot and GHEFT docking mechanism, under the action of the proposed HTT algorithm and IBVS tracking. Future work will focus on addressing the limitations of the existing system to improve the autonomous docking performance as mentioned in the following section.

Conclusion and Future Work

This paper presented a self-reconfigurable mobile robot with an integrated 2-DOF active docking mechanism. The robot possesses hybrid mobility where both track-based and wheel-based mechanisms can be actuated for locomotion to improve adaptability to terrains. Furthermore, this robot has been used to perform autonomous docking of multiple robots where the active mode of the 2-DOF docking mechanism has been demonstrated to couple or de-couple with the target robots. This docking process is achieved and experimentally validated using a hybrid-tracking approach, which helps to visually track and aid the tracking of the target robot relative to the source robot. Additionally, the onboard sensors play a crucial role in the relative positioning and synchronized working of both the track-based and the wheel-based mobility of the robot. This also helped to analyze the key characteristics of the hybrid mobility mechanism compared to conventional robots. The experiments consolidate the versatility of the mobile mechanism and tracking algorithm for docking in mobile robots.

As a part of future work, we envision extending the docking applications in several ways with the development of a manipulator robotic module with docking capabilities. This will enable the modular robotic system to achieve configurations as shown in Fig. 1. Using such configurations, we hope to extend the manipulation capabilities of the modular robot by incorporating grasping with added modularity. Experimental validation of the docking application will be further extended to analyze the performance

---

[2]https://www.youtube.com/watch?v=lmZlqtlIcX0

under unstructured terrain with misalignment along all six axes (X–Y–Z–Roll–Pitch–Yaw).

Appendix

The relation of the pixel coordinates and the corresponding 3D point coordinate (X, Y, and Z) can be obtained using its 2D point projection (x, y) as

$$x = X/Z = (u - c_u)/f\alpha$$
$$y = Y/Z = (v - c_v)/f \tag{A1}$$

where (u, v) represents the pixel coordinates of a point on the image plane. The set (c_u, c_v, f, α) defines the intrinsic parameters of the camera, where (c_u, c_v) represents the coordinates of the principal point, f is the focal length, and α is the ratio of the pixel dimensions. The time derivative of the above equation gives

$$\dot{x} = (\dot{X} - x\dot{Z})/Z$$
$$\dot{y} = (\dot{Y} - y\dot{Z})/Z \tag{A2}$$

Since the camera placement of the robot is an eye-in-hand configuration so the velocity of the 3D point (P) can be related to the spatial velocity (v, ω) of the camera as

$$\dot{P} = -v \; - \; \omega \times P = \begin{cases} \dot{X} = -v_x - \; \omega_y Z \\ \dot{Y} = -v_y \\ \dot{Z} = -v_z + \; \omega_y X \end{cases} \tag{A3}$$

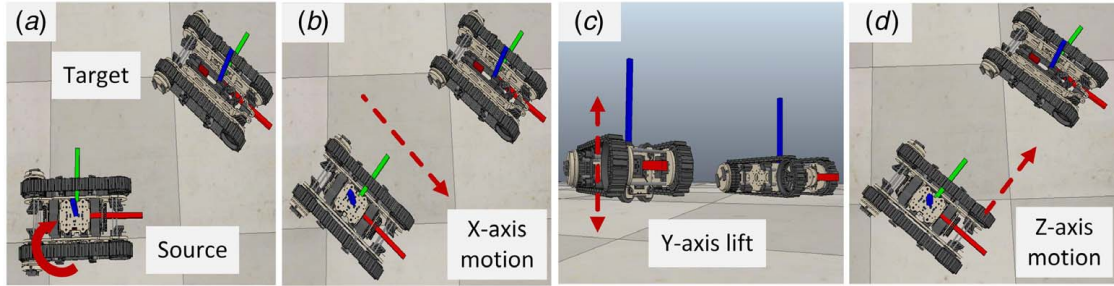**Fig. 15 IBVS simulation using locomotion (source) and manipulator robot (target) shown in V-REP simulated environment, where local X, Y, and Z-axes are defined by red, blue and green bar: (*a*) orientation alignment using onboard positioning sensor data, (*b*) translational motion along X-axis, (*c*) upward lift for wheeled assembly actuation, and (*d*) translational motion along Z-axis**
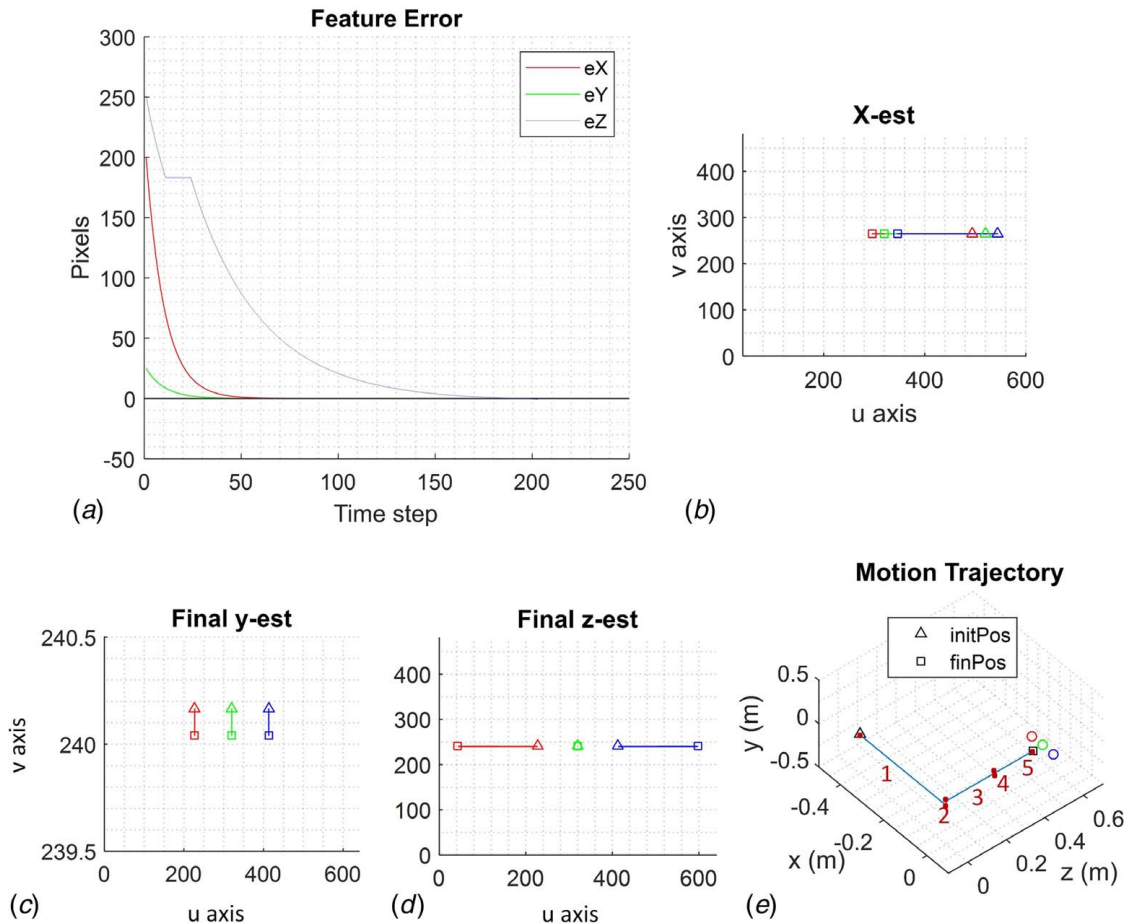

Fig. 16 IBVS simulation for the autonomous docking control: (*a*) feature error plot, (*b*) X-axis, (*c*) Y-axis, (*d*) Z-axis, and (*e*) trajectory based on real-world coordinates

Considering the motion configuration of the robot (shown in Fig. 15), it has two independent DOFs in each mode and an additional DOF along the Y-axis. The image interaction matrix (L) for a single feature using the above-mentioned equations can be written as

$$L_i = \begin{bmatrix} -\dfrac{f}{\rho_u Z} & 0 & \dfrac{u_i}{Z} & 0 & -\dfrac{f^2 + \rho_u^2 u_i^2}{\rho_u f} & 0 \\ 0 & -\dfrac{f}{\rho_v Z} & \dfrac{v_i}{Z} & 0 & -\dfrac{\rho_v u_i v_i}{f} & 0 \end{bmatrix}, \quad \forall\ i = 1, 2, 3$$

(A4)

where f is the calibrated focal length, Z is the estimated depth value, (ρ_u, ρ_v) defines the pixel size, and (u, v) defines the pixel coordinate of the corresponding feature. Solving these parameters for all the three features results in an Image interaction matrix of size 6×6. The velocity of the robot $[v_c, \omega_c]^T (= [v_x, v_y, v_z, \omega_x, \omega_y, \omega_z]^T)$ corresponds to the feature error, Δe_f, where ω_x and ω_z is 0. Based on these parameters and Eq. (A5), the velocity control for the required locomotion can be written as

$$[v_c, \omega_c]^T = \lambda \left(\begin{bmatrix} L_1 & L_2 & L_3 \end{bmatrix}^T \right)^{-1} \Delta e_f$$

(A5)

where λ, L_1, L_2, and L_3 represent the scalar gain, image interaction matrix of features 1, 2, and 3, respectively. These velocity parameters (Eq. (A5)) are used as the velocity inputs of the source robot and are shown in Fig. 16 using feature error plots and real-world motion trajectory of the robot.

References

[1] Yim, M., Shen, W. M., Salemi, B., Rus, D., Moll, M., Lipson, H., Klavins, E., and Chirikjian, G. S., 2007, "Modular Self-reconfigurable Robot Systems: Challenges and Opportunities for the Future," IEEE Rob. Autom. Mag., **14**(1), pp. 2–11.

[2] Castano, A., Behar, A., and Will, P. M., 2002, "The CONRO Modules for Reconfigurable Robots," IEEE/ASME Trans. Mechatron., **7**(4), pp. 403–409.

[3] Brown, H. B., Vande Weghe, J. M., Bererton, C. A., and Khosla, P. K., 2002, "Millibot Train for Enhanced Mobility," IEEE/ASME Trans. Mechatron., **7**(4), pp. 452–461.

[4] Daudelin, J., Jing, G., Tosun, T., Yim, M., Kress-Gazit, H., and Campbell, M., 2018, "An Integrated System for Perception-Driven Autonomy With Modular Robots," Sci. Rob., **3**(23).

[5] Wang, W., Yu, W., and Zhang, H., 2010, "JL-2: A Mobile Multi-robot System With Docking and Manipulating Capabilities," Int. J. Adv. Rob. Syst., **7**(1).

[6] Pacheco, M., Fogh, R., Lund, H. H., and Christensen, D. J., 2015, "Fable II: Design of a Modular Robot for Creative Learning," IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, May 26–30, pp. 6134–6139.

[7] Shirmohammadi, B., Taylor, C. J., Yim, M., Sastra, J., and Park, M., 2007, "Using Smart Cameras to Localize Self-assembling Modular Robots," Proceedings of the 1st ACM/IEEE International Conference on Distributed Smart Cameras, Vienna, Austria, Sept. 25–28, pp. 76–80.

[8] Murata, S., Yoshida, E., Kamimura, A., Kurokawa, H., Tomita, K., and Kokaji, S., 2002, "M-TRAN: Self-reconfigurable Modular Robotic System," IEEE/ASME Trans. Mechatron., **7**(4), pp. 431–441.

[9] Delrobaei, M., and McIsaac, K. A., 2009, "Connection Mechanism for Autonomous Self-assembly in Mobile Robots," IEEE Trans. Rob., **25**(6), pp. 1413–1419.

[10] Gilpin, K., and Rus, D., 2010, "Modular Robot Systems: From Self-assembly to Self-disassembly," IEEE Rob. Autom. Mag., **17**(3), pp. 38–55.

[11] Moubarak, P., Alvarez, E., and Ben-Tzvi, P., 2013, "Reconfiguring a Modular Robot Into a Humanoid Formation: A Multi-body Dynamic Perspective on Motion Scheduling for Modules and Their Assemblies," IEEE International Conference on Automation Science and Engineering. (CASE), Madison, WI, pp. 687–692.

[12] Moubarak, P., and Ben-Tzvi, P., 2012, "Modular and Reconfigurable Mobile Robotics," Rob. Autom. Syst., **60**(12), pp. 1648–1663.

[13] Zhong, M., Li, M., and Sun, L., 2017, "Tanbot: A Mobile Self-reconfigurable Robot Enhanced With Embedded Positioning Module," Intelligent Robotics and Applications: First International Conference, ICIRA, Cleveland, OH, Aug. 6–9.

[14] Kim, Y., and Minor, M., 2010, "Distributed Kinematic Motion Control of Multi-robot Coordination Subject to Physical Constraints," Int. J. Rob. Res., **29**(1), pp. 92–109.

[15] Kimura, H., and Hirose, S., 2002, "Development of Genbu: Active Wheel Passive Joint Articulated Mobile Robot," 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems, Lausanne, pp. 823–828.

[16] Ben-Tzvi, P., Goldenberg, A. A., and Zu, J. W., 2008, "Design and Analysis of a Hybrid Mobile Robot Mechanism With Compounded Locomotion and Manipulation Capability," ASME J. Mech. Des., **130**(7), p. 072302.

[17] Ben-Tzvi, P., Goldenberg, A. A., and Zu, J. W., 2010, "Articulated Hybrid Mobile Robot Mechanism With Compounded Mobility and Manipulation and Onboard Wireless Sensor/Actuator Control Interfaces," Mechatronics, **20**(6), pp. 627–639.

[18] Bhatt, R., Tang, C. P., Abou-Samah, M., and Krovi, V., 2005, "A Screw-Theoretic Analysis Framework for Payload Manipulation by Mobile Manipulator Collectives," 2005 ASME International Mechanical Engineering Congress and Exposition, IMECE 2005, Orlando, FL, pp. 1597–1606.

[19] Wilson, S., Gameros, R., Sheely, M., Lin, M., Dover, K., Gevorkyan, R., Haberland, M., Bertozzi, A., and Berman, S., 2016, "Pheeno, A Versatile Swarm Robotic Research and Education Platform," IEEE Robot. Autom. Lett., **1**(2), pp. 884–891.

[20] Kume, Y., Hirata, Y., Wang, Z. D., and Kosuge, K., 2002, "Decentralized Control of Multiple Mobile Manipulators Handling a Single Object in Coordination," 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems, Lausanne, pp. 2758–2763.

[21] Abou-Samah, M., Tang, C. P., Bhatt, R. M., and Krovi, V., 2006, "A Kinematically Compatible Framework for Cooperative Payload Transport by Nonholonomic Mobile Manipulators," Auton. Robots, **21**(3), pp. 227–242.

[22] Kim, Y., and Minor, M. A., 2010, "Coordinated Kinematic Control of Compliantly Coupled Multi-robot Systems in an Array Format," IEEE Trans. Rob., **26**(1), pp. 173–180.

[23] Lyder, A., Garcia, R., and Stoy, K., 2008, "Mechanical Design of Odin, an Extendable Heterogeneous Deformable Modular Robot," IEEE/RSJ International Conference on Intelligent Robots and Systems, Nice, France, Sept. 22–26.

[24] Kutzer, M. D., Moses, M. S., Brown, C. Y., Scheidt, D. H., Chirikjian, G. S., and Armand, M., 2010, "Design of a New Independently-Mobile Reconfigurable Modular Robot," 2010 IEEE International Conference on Robotics and Automation, ICRA 2010, Anchorage, AK, May 3–7, pp. 2758–2764.

[25] Davey, J., Kwok, N., and Yim, M., 2012, "Emulating Self-reconfigurable Robots—Design of the SMORES System," 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura-Algarve, Portugal, pp. 4464–4469.

[26] Baca, J., Hossain, S. G. M., Dasgupta, P., Nelson, C., and Dutta, A., 2014, "ModRED: Hardware Design and Reconfiguration Planning for a High Dexterity Modular Self-reconfigurable Robot for Extra-Terrestrial Exploration," Rob. Auton. Syst., **62**(7), pp. 1002–1015.

[27] Baca, J., Ferre, M., and Aracil, R., 2012, "A Heterogeneous Modular Robotic Design for Fast Response to a Diversity of Tasks," Rob. Auton. Syst., **60**(4), pp. 522–531.

[28] Spletzer, J., Das, A. K., Fierro, R., Taylor, C. J., Kumar, V., and Ostrowski, J. P., 2001, "Cooperative Localization and Control for Multi-robot Manipulation," IEEE International Conference on Intelligent Robots and Systems, Vol. 2, pp. 631–636.

[29] Heo, D. H., Oh, A. R., and Park, T. H., 2011, "A Localization System of Mobile Robots Using Artificial Landmarks," Proceedings of 7th IEEE International Conference on Automation Science and Engineering, CASE 2011, Trieste, Oct. 29–Nov. 3, pp. 139–144.

[30] Das, A. K., Fierro, R., Kumar, V., Ostrowski, J. P., Spletzer, J., and Taylor, C. J., 2002, "A Vision-Based Formation Control Framework," IEEE Trans. Rob. Autom., **18**(5), pp. 813–825.

[31] Kumar, P., Saab, W., and Ben-Tzvi, P., "Design of a Multi-directional Hybrid-Locomotion Modular Robot With Feedforward Stability Control," Proceedings of the 2017 ASME IEDTC/CIE, Vol. 5B: 41st Mechanisms and Robotics Conference, p. V05BT08A010.

[32] Suh, J. W., Homans, S. B., and Yim, M., 2002, "Telecubes: Mechanical Design of a Module for Self-Reconfigurable Robotics," Proceedings of the IEEE International Conference on Robotics and Automation, Washington, DC, Nov. 3–7, pp. 4095–4101.

[33] Saab, W., and Ben-Tzvi, P., 2016, "A Genderless Coupling Mechanism With 6-DOF Misalignment Capability for Modular Self-Reconfigurable Robots," ASME J. Mech. Rob., **8**(6), p. 061014.

[34] Romanishin, J. W., Gilpin, K., and Rus, D., 2013, "M-Blocks: Momentum-Driven, Magnetic Modular Robots," Proceedings of the IEEE International Conference on Intelligent Robots and Systems, San Diego, CA, Oct. 29–Nov. 2, pp. 4288–4295.

[35] Kirby, B. T., Aksak, B., Campbell, J. D., Hoburg, J. F., Mowry, T. C., Pillai, P., and Goldstein, S. C., 2007, "A Modular Robotic System Using Magnetic Force Effectors," Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, San Diego, CA, Dec. 11–14, pp. 2787–2793.

[36] Qiao, G., Song, G., Zhang, J., Sun, H., Wang, W., and Song, A., 2012, "Design of Transmote: A Modular Self-Reconfigurable Robot with Versatile Transformation Capabilities," Proceedings of the IEEE International Conference on Robotics and Biomimetics, Guangzhou, China, pp. 1331–1336.

[37] Wei, H., Chen, Y., Tan, J., and Wang, T., 2011, "Sambot: A Self-assembly Modular Robot System," IEEE/ASME Trans. Mechatron., **16**(4), pp. 745–757.

[38] Yim, M., Zhang, Y., Roufas, K., Duff, D., and Eldershaw, C., 2002, "Connecting and Disconnecting for Chain Self-configuration With Polybot," IEEE/ASME Trans. Mechatron., **7**(4), pp. 442–451.

[39] Moubarak, P., and Ben-Tzvi, P., 2014, "A Tristate Rigid Reversible and Non-Back-Drivable Active Docking Mechanism for Modular Robotics," IEEE/ASME Trans. Mechatron., **19**(3), pp. 840–851.

[40] Saab, W., Racioppo, P., and Ben-Tzvi, P., 2019, "A Review of Coupling Mechanism Design for Modular Reconfigurable Robots," Robotica, **37**(2), pp. 378–403.

[41] Groß, R., Bonani, M., Mondada, F., and Dorigo, M., 2006, "Autonomous Self-assembly in a Swarm-Bot," International Symposium on Autonomous Mini Robots for Research and Edutainment, pp. 314–322.

[42] Bererton, C., and Khosla, P., 2000, "Toward a Team of Robots with Repair Capabilities: A Visual Docking System," Proceedings of the 7th International Symposium on Experimental Robotics, Jan. 25, Springer, Berlin/Heidelberg, pp. 333–342.

[43] Park, M., Chitta, S., Teichman, A., and Yim, M., 2008, "Automatic Configuration Recognition Methods in Modular Roots," Int. J. Rob. Res., **27**(3–4), pp. 403–421.

[44] Corke, P. I., 2017, *Robotics, Vision and Control—Fundamental Algorithms in MATLAB*, 2nd ed., Vol. 118, Springer, New York.

[45] Hutchinson, S., Hager, G., and Corke, P., 1996, "A Tutorial on Visual Servo Control," IEEE Trans. Rob. Autom., **12**(5), pp. 651–670.

[46] Sebastian, B., and Ben-Tzvi, P., 2019, "Physics-Based Path Planning for Autonomous Tracked Vehicles in Challenging Terrain," J. Intell. Rob. Syst., **95**(2), pp. 511–526.

[47] Sebastian, B., and Ben-Tzvi, P., 2019, "Support Vector Machine-Based Real-Time Terrain Estimation for Tracked Robots," Mechatronics, **62**, p. 102260.

[48] Kumar, A., and Ben-Tzvi, P., 2016, "Spatial Object Tracking System Based on Linear Optical Sensor Arrays," IEEE Sensors J., **16**(22), pp. 7933–7940.

[49] Sohal, S. S., Saab, W., and Ben-Tzvi, P., "Improved Alignment Estimation for Autonomous Docking in Mobile Robots," Proceedings of the 2018 ASME IDETC/CIE, Vol. 5A: 42nd Mechanisms & Robotics Conference, Quebec City, Quebec, Canada, Aug. 26–29, p. V05AT07A072.

[50] Garrido-Jurado, S., Muñoz-Salinas, R., Madrid-Cuevas, F. J., and Marín-Jiménez, M. J., 2014, "Automatic Generation and Detection of Highly Reliable Fiducial Markers Under Occlusion," Pattern Recognit., **47**(6), pp. 2280–2292.

[51] Redmon, J., Divvala, S., Girshick, R., and Farhadi, A., 2016, "You Only Look Once: Unified, Real-Time Object Detection," IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, June 27–30, pp. 779–788.

[52] Lucas, B. D., and Kanade, T., 1981, "An Iterative Image Registration Technique With an Application to Stereo Vision," Proceedings of 7th International Joint Conference on Artificial Intelligence, Vancouver, BC, Canada, Aug. 24, pp. 674–679.