Vision-Based Docking for Robotic Towing of Mobile Platforms

J. P. Mafaldo[1†], P. C. G. Rodrigues[1†], M. R. da Silva Júnior[1] and A. Durand-Petiteville[1]

***Abstract*— This work has its origins in the design of autonomous stretchers and presents a vision-based docking system designed to autonomously connect a differential drive robot to a mobile platform. A visual marker is mounted on the mobile platform, allowing its detection through a forward-facing camera onboard the robot, and the robot is equipped with a gripper to connect to the two devices. The connection, similar to a docking problem, is performed by executing a sequence of maneuvers. To do so, a hybrid control strategy is employed, using position-based control for coarse maneuvering and image-based control for precise final alignment, overcoming the robot's non-holonomic constraints. Numerous experimental tests demonstrate the system's feasibility and highlight its potential for improving hospital logistics using low-cost, modular solutions.**

I. INTRODUCTION

Robotics has been identified as one of the main solutions to help improve healthcare systems by enabling high levels of patient care, clinical productivity, and safety for both patients and healthcare workers [1], [2]. In fact, many applications related to logistics [3], pharmacy, wheelchairs [4], rehabilitation, remote assessment, and surgical robots [5] have already been developed. However, little work has been done to automate tasks involving hospital stretchers. Among the problems studied, we can first mention motorization. For example, [6] presents a study on the health benefits of a novel robotic-assisted omnidirectional hospital bed transporter in comparison with a conventional manual transport stretcher. Another challenge approach consists in providing stretchers the ability to navigate autonomously. In [7], the authors equipped a stretcher with a laser range finder to allow it to navigate while avoiding obstacles and, more specifically, walking people. In [8], the stretcher is semi-autonomous: an agent drives the stretcher, while an obstacle avoidance system deals with hazards. More recently, a solution relying on radio frequency tags to navigate in hospitals was presented in [9]. Thus, while autonomous stretchers are not yet available, it is believed that these systems could improve efficiency and reduce musculoskeletal injuries in clinicians.

In this work, instead of considering autonomous stretchers, it is proposed to explore an approach where the stretchers are moved by a small fleet of autonomous robots. Such an approach has two main advantages over those based on autonomous stretchers: (i) there is no need to replace the stretchers already present in the hospital, and (ii) the number of intelligent agents is reduced. To develop an autonomous stretcher-towing robot, it is necessary to address two specific challenges: (i) autonomously connecting the robot to the stretcher and (ii) autonomously navigating with a towed stretcher. This work focuses on the first challenge and presents a proof of concept using a robot and a mobile platform emulating a stretcher.

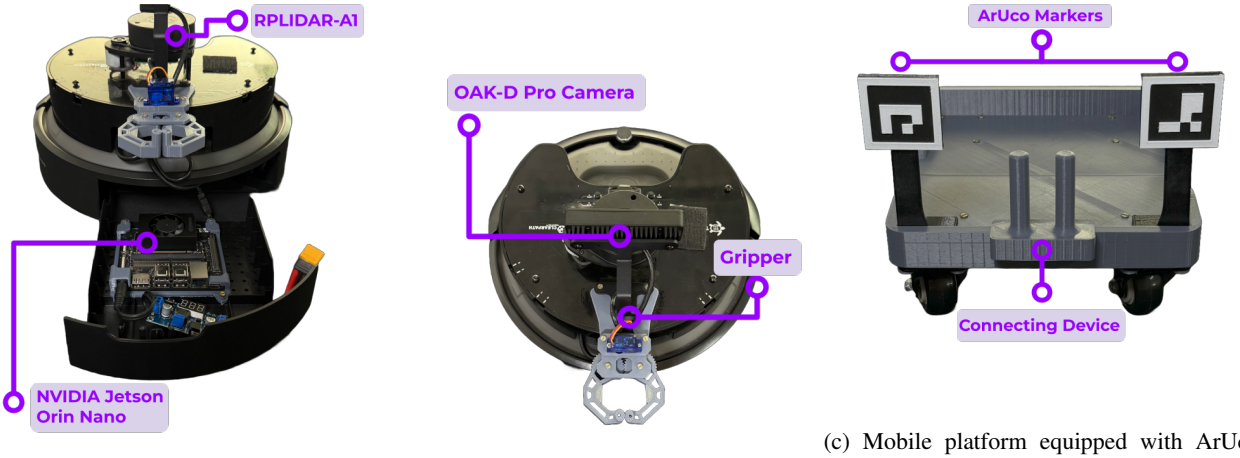
The connecting task involves aligning the robot with the mobile platform and performing a coupling using a robotic gripper. This process is framed as a docking problem due to its reliance on precise positioning and interaction [10] and can be solved relying on infrared emitter-receiver, laser rangefinder, or camera. When using an infrared emitter-receiver [11], the docking station has to be equipped with an active device. This solution is not cost-effective for numerous docks, or stretchers, in the present case. In the case of laser rangefinders [12], [13], the sensors have to be at the same height as the pattern of interest they have to detect. This solution is not versatile enough for dealing with different versions of the same object. Finally, cameras allow us to detect passive landmarks of interest at different heights. In [14], [15], [16], the Cartesian coordinates of the landmarks are estimated to then drive the robot to a desired position using a position-based controller. However, such an estimation is highly sensitive to camera calibration and measures errors leading to poor positioning.

In this paper, it is proposed to investigate the use of vision-based servoing to autonomously dock a robot to a mobile platform. Such an approach has several advantages. First, a camera has a large field of view, reducing the constraints of positioning relative to the platform. Next, relying on computer vision allows detecting different types of landmarks. Finally, the landmark of interest does not have to be active. However, vision-based controllers aim at controlling the pose of the camera without any guarantee on the state of the robotic system embedding the sensor. To overcome this limitation, it is proposed to develop a docking strategy composed of distinct maneuvers. Each maneuver is triggered based on criteria and is executed either by an image-based or by a position-based visual controller. In the remainder of the paper, we detail the image processing, criteria, and controller design required to visually dock a robot. Thus, the second section presents the material, while the third one details the control strategy. Finally, the fourth section presents the results obtained during the tests.

*This work was supported by the research and innovation cooperation project between Softex (funded by MCTI through Law 8.248/91 in the scope of the National Priority Program) and CIn-UFPE.

†Both authors contributed equally.

[1]All authors are affiliated with Centro de Informatica, Universidade Federal de Pernambuco, Av. Jornalista Anibal Fernandes, s/n - Cidade Universitaria, 50.740-560, Recife, PE, Brazil. [jpmp, pcgr, mrsj, adrien][at]cin.ufpe.br



(a) Internal view of the electronics bay.

(b) Top view of the mobile robot.

(c) Mobile platform equipped with ArUco markers and coupling posts.

Fig. 1: Mobile robot and mobile platform.

II. HARDWARE

To design our solution to autonomously dock a mobile robot with a mobile platform, we rely on the TurtleBot4 platform. It consists of a differential drive robot whose linear and angular speeds, respectively v and ω, can be controlled. It is equipped with its standard equipment: a 2-D laser rangefinder (RPLIDAR-A1 from SLAMTECH), an RGB-D camera (OAK-D Pro from Luxonis), and a single-board computer (Raspberry Pi4). Two pieces of equipment were added to the platform for this project. First, another single-board computer (Jetson Orin Nano from NVIDIA) with a battery was included to increase the computational capacity of the robot (see Fig. 1a). Second, a 3-D printed gripper was installed at the back of the robotic platform (see Fig. 1b). The gripper is operated by a servomotor, and thus it is possible to automatically open or close it.

To emulate a hospital stretcher, the mobile platform shown in Fig. 1c was designed and 3-D printed. It consists of a platform equipped with four caster wheels and coupling posts. The latter is made of two rigid tubes and allows the gripper to physically attach the robot to the platform. Moreover, the platform is equipped with two 3-D printed ArUco markers [17]. The markers are passive devices used by the robot to drive and position itself relative to the platform. The markers were ultimately 3D-printed to prevent physical deformations that could hinder detection and precise robot positioning.

III. AUTONOMOUS DOCKING

The strategy designed in this work aims at autonomously docking a differential drive robot to a mobile platform, *e.g.,* a hospital stretcher, using a camera as the main sensor while driving. As already mentioned, cameras can be used to efficiently detect landmarks of interest and provide relevant data to position the robot relative to the passive device. Regarding the control of the robot, vision-based servoing can be divided into two main categories [18]. On the one hand, image-based controllers rely on measures computed in the image space. Such an approach is known to be little sensitive to calibration and measurement errors and thus allows a precise relative positioning of the camera. However, classical image-based servoing requires the camera to be holonomic, which can be achieved by using a holonomic or redundant mobile robot. Holonomic robots are usually not indicated to tow platforms due to their price and small relative payload. This is why it was decided to use a differential drive robot. Classically, such robots are equipped with a camera on a pan-platform to add a degree of freedom and create a redundant system [19]. For the coupling problem, this solution cannot be used. Indeed, a rigid spatial relation between the camera and the gripper is necessary to guarantee that regulating an error in the image space is equivalent to positioning the gripper. The use of an additional degree of freedom would lead to an infinite number of gripper positions for a unique error regulation. It is then proposed to control the non-holonomic platform using a proportional controller regulating an error expressed in the image space. Such an approach leads to an accurate positioning but has a non-infinite set of initial configurations. It is then necessary to start the image-based control from a position that will lead to a correct positioning.

On the other hand, position-based controllers rely on perspective geometry to estimate the pose of the robot relative to a landmark. They are suitable to drive non-holonomic robots, such as differential drive ones, from and to any given pose [20]. However, pose estimation is highly sensitive to calibration and measurement errors, making it challenging to accurately position the platform. For this reason, it is proposed to rely on position-based controllers to perform maneuvers while the robot is far from the goal, *i.e.,* while accuracy is not mandatory. These maneuvers aim at positioning the robot at a pose from which the image-based controller can be started. Since no single approach is sufficient to achieve docking, the robot must perform a sequence of maneuvers carried out with various control modalities. In the remainder of this section, we first present

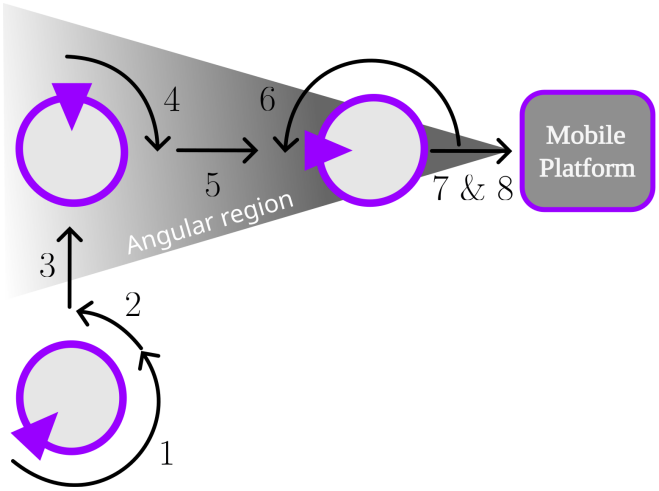


Fig. 2: Sequence of maneuvers to autonomously dock: Initial alignment (1) - Enter Angular Region (2-4) - Approach (5) - Connect Devices (6-8).

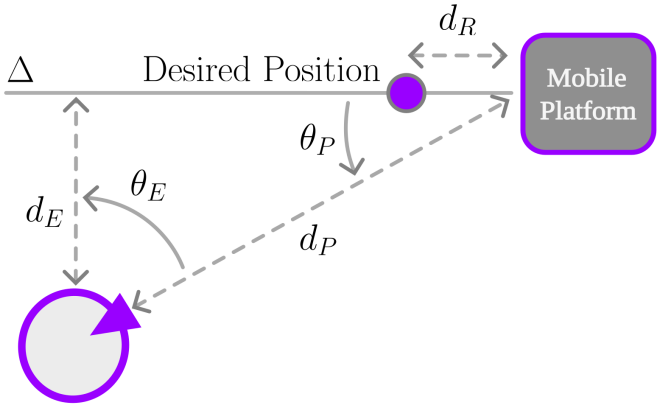


Fig. 3: Parameters for entering the angular region and connecting the devices.

the sequence of maneuvers and then detail how each of them is performed.

A. Docking Maneuver Breakdown

The autonomous docking strategy is composed of up to 4 maneuvers shown in Fig. 2.

- `Initial Alignment` (Step 1): The robot rotates to center the landmark in the field of view of the camera. From this pose, it determines if it lies within an angular region defined by the user. This region represents the initial positions from which the robot will accurately reach the desired position by being driven by the image-based controller. Thus, if it lies within the region, the `Enter the Angular Region` maneuver is skipped and the `Approach` one is started.
- `Enter Angular Region` (Steps 2–4): First, the robot rotates to be perpendicular to the angular region bisectrix (2). Next, it moves forward to lie on the bisectrix (3). Finally, it rotates to align with the mobile platform (4).
- `Approach` (Step 5): The robot approaches the mobile platform to reach a user-defined relative position.
- `Connect Devices` (Steps 6-8): First, the robot performs a 180° rotation to face the gripper to the mobile platform (6). Next, it moves backward to position the gripper around the connecting device (7). The gripper is closed to establish a physical connection between the robot and the mobile platform (8).

B. Image Processing and Pose Estimation

The images are processed using the `ArUco` module from the `OpenCV` library (see Fig. 4). It consists in detecting the two ArUco markers and in providing the (u_i, v_i) coordinates of the eight corners p_i, with $\forall i \in \mathbb{N},\ 1 \leq i \leq 8$. The center of mass of the two ArUco markers, whose coordinates are denoted (u_C, v_C), is computed by averaging the coordinates (u_i, v_i) of the points p_i. The coordinates (u_C, v_C) are used in the following to feed the image-based controllers.

The `solvePnP` module also allows us to estimate the relative orientation θ_P and distance d_P to the mobile platform (see Fig. 3). Indeed, knowing the geometry of the markers and using perspective geometry, it is possible to obtain a rough estimate of these parameters. They are used in the following to feed open-loop position-based controllers.

C. Initial Alignment

The robot can be initially in any pose. Thus, the landmark can be outside or on the borders of the field of view. The `Initial Alignment` process is made of two steps. The first one is performed in an open-loop fashion and consists of rotating the robot until the landmark is completely visible. The second part aims to center the landmark in the image. To do so, the robot is controlled in a closed-loop fashion, and the angular velocity ω is computed using Eq. (1), where K_a is a user-defined scalar gain.

$$\omega = K_a e_a \text{ with } e_a = v_c \tag{1}$$

Regulating e_a centers the landmark in the field of view of the camera.

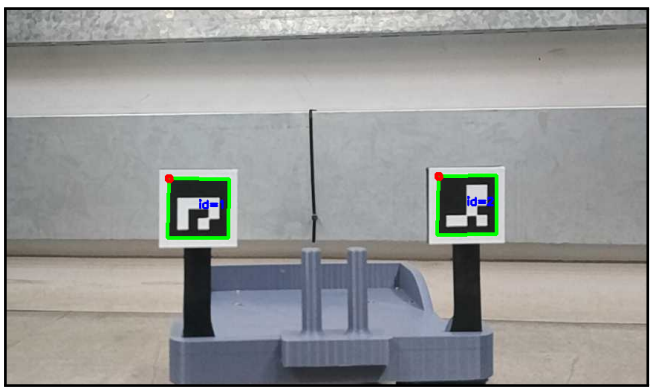


Fig. 4: Example of AruCo detection.

D. Enter Angular Region

As it was previously mentioned, it is not possible to drive a differential drive robot from any pose to a given one by uniquely relying on an image-based controller. However, it is possible to define an area from which the robot can successfully reach the desired pose. In this work, such an area is represented by an angular region whose origin is the mobile platform and its angle is experimentally determined (see Sec. V). This maneuver thus aims at driving the robot, initially outside the angular region, to the line Δ orthogonal to the mobile platform (see Fig. 3). To do so, the angle θ_E and distance d_E connecting the current robot pose to the line Δ are calculated using Eq. (2) and (3).

$$\theta_E = \pi - \theta_P \tag{2}$$

$$d_E = \sin(\theta_P) d_P \tag{3}$$

Thus, θ_E is used to first rotate the robot towards the line Δ, and d_E to move forward to reach the line. Note that the rotation and translation are performed in an open-loop fashion from an exteroceptive perspective, *i.e.,* the wheel encoders are the only sensors used. The pose reached after maneuvering most likely does not lie on the line Δ. Such an inaccuracy can be tolerated, as the aim of this maneuver was to enter the angular region from which the image-based controller can be started. Finally, using Eq. (1), the robot rotates to face the mobile platform.

E. Approach

When the robot lies within the angular region and is aligned to the mobile platform, it is then possible to execute the `Approach` process. It aims to drive the robot to a pose of reference, relying on images. The pose of reference lies in front of the mobile platform at an user-defined distance d_R (see Fig. 3). To design the image-based controller, an image is captured at the pose of reference and processed to compute the coordinates of the center of mass of the markers. They are denoted (u_C^*, v_C^*) and represent the desired values. For each coordinate, the error between the current and desired values controls one degree of freedom of the robot: u controls the linear velocity while v controls the angular velocity. Such a behavior is encoded in Eq. (4) and (5), where K_v and K_ω are two scalar gains. Regulating (4) and (5) drives the robot to a known pose relative to the mobile platform where the connecting maneuver can be started.

$$v = K_v e_v \text{ with } e_v = u_c - u_c^* \tag{4}$$

$$\omega = K_\omega e_\omega \text{ with } e_\omega = v_c - v_c^* \tag{5}$$

F. Connect Devices

The final maneuver aims to connect the two devices from a relative pose known with a certain degree of accuracy. For this reason, it is possible to rely on open-loop solutions. Initially, in front of the passive device, the robot first rotates by $180°$ to align the gripper with the coupling device. Next, it drives backward by a distance of d_R to position the open gripper around the coupling device. Finally, the gripper is closed to complete the connection to the mobile platform.

IV. RESULTS

In this section, we present the results obtained with the autonomous visual docking of the TurtleBot4 robot to the mobile platform. The strategy is implemented using the Python language coupled with ROS2 Humble on a Raspberry Pi 4 and a NVIDIA Jetson Orin Nano board. The control parameters are set to $K_a = 0.5$, $K_v = 0.25$, and $K_\omega = 0.5$. Finally, the robot is controlled at a frequency of 15 Hz.

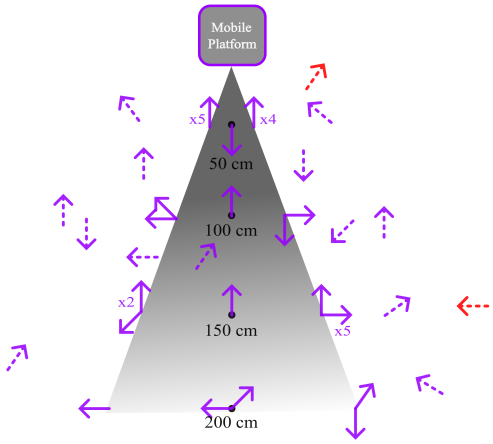


Fig. 5: Map of the initial poses (Purple: successful - Red: failed - Plain: angular region tests - Dashed: additional tests).

It was decided to configure the angular region with an angle of $40°$ ($\pm 20°$). This value was obtained empirically after a few tests. To guarantee that it was possible to reach the desired pose from any initial one inside the angular region, 30 tests were performed. The spatial distribution is shown in Fig. 5 (plain arrows). The robot successfully reached the desired position and managed to connect to the mobile platform from the 30 poses within the angular region. The `Enter Angular Region` maneuver was executed in one occasion due to an inaccurate pose estimation. Indeed, many initial poses were at the edge of the angular region and as it was mentioned earlier, the pose calculation algorithm only provides a rough estimate. Despite the unnecessary maneuvers, the robot managed to dock autonomously.

A breakdown of the test starting at the pose (100 cm, $10°$) is shown in Fig. 6. Initially, the robot cannot perceive the tags (step 0). It then performs a rotation up to center the landmarks in the image (step 1). After estimating its pose relative to the mobile platform, it identifies itself as being in the angular region. For this reason, it starts the `Approach` maneuver, *i.e.,* the image-based visual servoing, to drive towards the desired relative pose (step 2). The image-based visual servoing stops when the error in the image is small enough, meaning the robot has reached the goal (step 3). It next rotates by $180°$ to face the gripper towards the mobile device (step 4), moves backward and closes the gripper to connect to the mobile platform (step 5).

Another test is detailed in Fig. 7, focusing on the `Approach` maneuver and more specifically on the image-based controller. Indeed, this maneuver is the most important

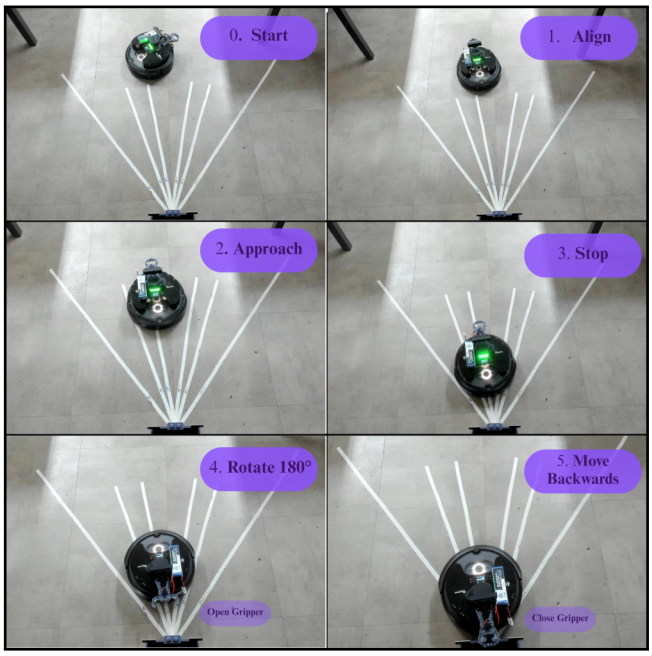


Fig. 6: Example of visual docking starting within the angular region.

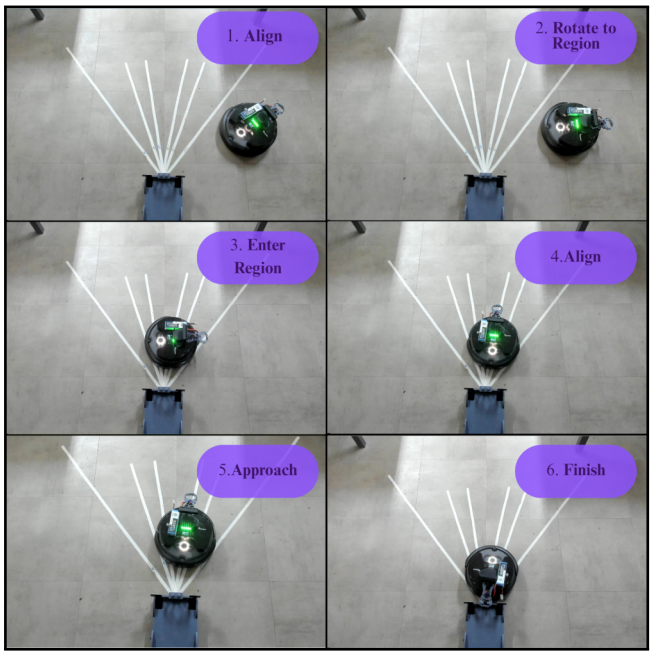


Fig. 8: Example of visual docking starting outside of the angular region.

one as it aims at accurately driving the robot to a position from which it will execute the `Connect Devices` in an open-loop fashion. Thus, in the first image (a), the original error in the image space between the current perception of the ArUco tags and the desired one is shown. Using this visual error to feed the image-based controllers encoded in Eq. (4) and (5), it is possible to drive the robot towards the goal, as shown in (b) and (c). Finally, the visual error is vanished (d), corresponding to the correct positioning of the robot relative to the Aruco tags, *i.e.,* to the mobile platform.

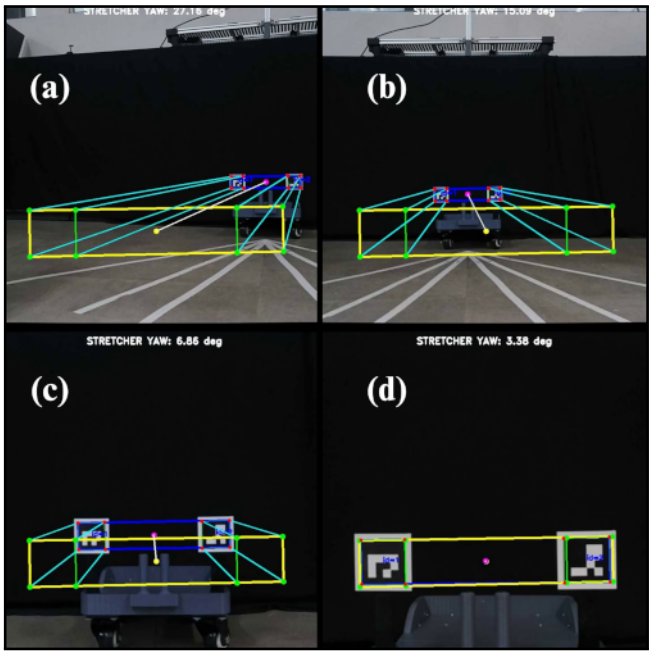


Fig. 7: Images captured during `Approach`: (a) initial image - (b)(c) intermediate images - (d) final image (Purple dots: current ArUco corners and center of mass - Yellow dot: desired center of mass).

15 additional tests were performed. The initial poses were randomly picked inside and outside the angular region. They are represented in Fig. 5 by the dashed arrows. Among the 15 tests, 13 were successful while 2 failed. First, we focus on a test starting at (50 cm, 55°) and requiring the `Enter Angular Region` maneuver. The different maneuvers are shown in Fig. 8. Initially, the robot rotates to look for the landmarks (step 1). Next, after estimating its relative pose to the mobile platform, it identifies being outside the angular region and starts the `Enter Angular Region` maneuver. First, it rotates to be orthogonal to the angular region bisectrix (step 2) and then drives forward to lie on the bisectrix (step 3). It ends the maneuver by rotating on itself to center the landmark in the field of view of the camera (step 4). The `Approach` maneuver is then executed to place the robot in front of the mobile device (step 5). Finally, the `Connect Devices` is executed: the robot performs a 180° rotation, moves forwards, and closes the gripper to connect the devices (step 6).

Finally, we analyze the two failing tests. The first one, (150 cm, 50°), starts far from the mobile platform. With such a distance and angle, the Aruco detector does not work properly. Thus, during the `Initial Alignment` maneuver, the markers were not detected, and the robot did not stop rotating. For the second test (50 cm, 55°), the camera perceives the landmarks with a strong perspective. The pose estimation is then less accurate and the `Enter Angular`

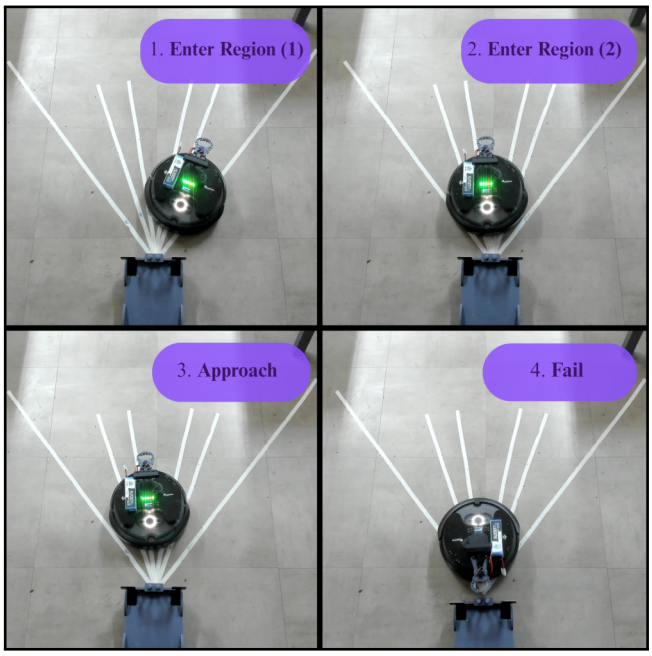


Fig. 9: Example of docking failure.

`Region` maneuver ends outside the angular region (step 1 in Fig. 9). It should be noticed that this region is narrow close to the mobile platform. Identifying itself outside the angular region, the robot starts a second `Enter Angular Region` maneuver ending within the region (step 2). However, the robot is too close to the platform, and the `Approach` maneuver actually moves backward (step 3). Unfortunately, the stopping criteria are inadequate for a backward movement, and the robot does not stop sufficiently close to the desired pose. Thus, the `Connect Devices` maneuver, performed in an open-loop fashion, fails at positioning the gripper around the connecting plots (step 4).

V. CONCLUSIONS

The work presented in this paper has its origins in hospital logistics problems and more particularly in the design of autonomous stretchers. Thus, we have addressed the problem of autonomously docking a robot to a mobile platform relying on visual servoing. The vision-based approach has the advantage of being able to detect diverse types of landmarks in large fields of view. From a control point of view, the designed strategy relies on a sequence of maneuvers performed by position-based or image-based visual servoing controllers. The tests performed have demonstrated the relevancy of a vision-based docking solution. The proof of concept being conclusive, it is now necessary to implement the proposed approach on a larger robot aiming at towing hospital stretchers. In addition to the change of scale, the final rotation angle could be adjusted based on the measures in the image made at the end of the `Approach` maneuver. Moreover, a contact sensor could be added to the gripper to stop the backward displacement to reduce the connecting failure. Such improvements could benefit the solution and make the development of autonomous stretchers possible.

REFERENCES

[1] T. Mettler, M. Sprenger, and R. Winter, "Service robots in hospitals: new perspectives on niche evolution and technology affordances," *European Journal of Information Systems*, vol. 26, no. 5, pp. 451–468, 2017.

[2] D. Silvera-Tawil, "Robotics in healthcare: A survey," *SN Computer Science*, vol. 5, no. 1, p. 189, 2024.

[3] A. G. Ozkil, Z. Fan, S. Dawids, H. Aanes, J. K. Kristensen, and K. H. Christensen, "Service robots for hospitals: A case study of transportation tasks in a hospital," in *2009 IEEE international conference on automation and logistics*. IEEE, 2009, pp. 289–294.

[4] J. Leaman and H. M. La, "A comprehensive review of smart wheelchairs: Past, present, and future," *IEEE Transactions on Human-Machine Systems*, vol. 47, no. 4, pp. 486–499, 2017.

[5] C. V. Riga, C. D. Bicknell, A. Rolls, N. J. Cheshire, and M. S. Hamady, "Robot-assisted fenestrated endovascular aneurysm repair (fevar) using the magellan system," *Journal of Vascular and Interventional Radiology*, vol. 24, no. 2, pp. 191–196, 2013.

[6] Z. Guo, R. B. Yee, K.-R. Mun, and H. Yu, "Experimental evaluation of a novel robotic hospital bed mover with omni-directional mobility," *Applied ergonomics*, vol. 65, pp. 389–397, 2017.

[7] C. Wang, A. V. Savkin, R. Clout, and H. T. Nguyen, "An intelligent robotic hospital bed for safe transportation of critical neurosurgery patients along crowded hospital corridors," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 23, no. 5, pp. 744–754, 2014.

[8] C. Wang, A. S. Matveev, A. V. Savkin, R. Clout, and H. T. Nguyen, "A semi-autonomous motorized mobile hospital bed for safe transportation of head injury patients in dynamic hospital environments without bed switching," *Robotica*, vol. 34, no. 8, pp. 1880–1897, 2016.

[9] V. Loganathan, S. Prabakar, R. Praveenraja, V. Raja, and S. Viisvesh, "Automated stretcher," in *2024 2nd International Conference on Artificial Intelligence and Machine Learning Applications Theme: Healthcare and Internet of Things (AIMLA)*. IEEE, 2024, pp. 1–6.

[10] M. C. Silverman, D. Nies, B. Jung, and G. S. Sukhatme, "Staying alive: A docking station for autonomous robot recharging," in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, vol. 1. IEEE, 2002, pp. 1050–1055.

[11] M. S. Rao and M. Shivakumar, "Ir based auto-recharging system for autonomous mobile robot," *Journal of Robotics and Control (JRC)*, vol. 2, no. 4, pp. 244–251, 2021.

[12] S. Vongbunyong, K. Thamrongaphichartkul, N. Worrasittichai, and A. Takutruea, "Automatic precision docking for autonomous mobile robot in hospital logistics-case-study: battery charging," in *IOP Conference Series: Materials Science and Engineering*, vol. 1137, no. 1. IOP Publishing, 2021, p. 012060.

[13] Y. Liu, "A laser intensity based autonomous docking approach for mobile robot recharging in unstructured environments," *IEEE Access*, vol. 10, pp. 71 165–71 176, 2022.

[14] R. C. Luo, C. T. Liao, and K. C. Lin, "Vision-based docking for automatic security robot power recharging," in *IEEE Workshop on Advanced Robotics and its Social Impacts, 2005.* IEEE, 2005, pp. 214–219.

[15] U. Kartoun, H. Stern, Y. Edan, C. Feied, J. Handler, M. Smith, and M. Gillam, "Vision-based autonomous robot self-docking and recharging," in *2006 World Automation Congress*, 2006, pp. 1–8.

[16] F. Guangrui and W. Geng, "Vision-based autonomous docking and re-charging system for mobile robot in warehouse environment," in *2017 2nd International Conference on Robotics and Automation Engineering (ICRAE)*. IEEE, 2017, pp. 79–83.

[17] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognition*, vol. 47, no. 6, pp. 2280–2292, 2014.

[18] F. Chaumette and S. Hutchinson, "Visual servo control. i. basic approaches," *IEEE Robotics & Automation Magazine*, vol. 13, no. 4, pp. 82–90, 2006.

[19] A. Durand-Petiteville and V. Cadenat, "Visual predictive control for differential drive robots with parallel implementation on gpu," *Computers and Electrical Engineering*, vol. 102, p. 108120, 2022.

[20] P. I. Corke, W. Jachimczyk, and R. Pillat, *Robotics, vision and control: fundamental algorithms in MATLAB*. Springer, 2011, vol. 73.