Sensor Integration for Autonomous Docking of a Mobile Robot with Omnidirectional Platform

Farid Alijani[1]

Abstract – *Docking of mobile robots requires precise raw measurements for pose estimation relative to the docking platform to accomplish the task successfully. Besides, the pose estimation of the robot with the sensors in an indoor environment should be accurate enough for localization and navigation toward the docking platform. However, the sensors are entitled to errors due to the measurement uncertainty. In this paper, sensor integration is exploited to decrease the measurement ambiguity and increase the accuracy of the docking.*

Keywords*: Autonomous robotic systems, Mobile robots, Guidance navigation and control*

I. Introduction

Docking is a critical task for industries employing mobile robots with platforms to be lifted or moved temporarily. It is also crucial for applications in which the mobile robot is required to recharge its batteries autonomously. Moreover, autonomous operations are highly preferable in robotics since human interactions and errors are eliminated. Such behavior demands precise measurements of the certain tasks. For this purpose, the robot should be equipped with sensors to first localize itself in an unknown environment then precede the final goal configurations. However, the sensor measurements are not flawless and can be inconsistent. Therefore, sensor integration could result in higher accuracy for docking.

I.1. Motivation

The sensor integration is crucial to increase the versatility and the application domains of the mobile robots since it manipulates environment to achieve different tasks. Sensor integration also eliminates the human control of actions to increase the accuracy of the autonomous accomplishments.

Time is a crucial parameter in robotics and the robot is required to accomplish the docking task in the shortest possible time. However, fulfilling merely time-oriented requirements influences the pose-estimation accuracy or the environment compatibility which may cause irreversible damages on the workplace. Therefore, a compromise between different requirements seems essential. Control system is an applicable tool to accomplish the desired docking tasks and satisfy all the constraints concurrently.

By means of the motion sensors with which the mobile robot is equipped, approximate pose estimation is obtained over time relative to a starting point. However, the large position errors make this approach unreliable and demand other sensors to roughly estimate the current position of the robot within an indoor environment. It was initially thought that information from the built-in odemetry sensor could help with the localization on the docking environment. However, practical experience with such sensor reveals inaccurate estimation due to accumulative errors in large docking trajectories.

I.2. Objectives

The end-effectors and the mechanical collinear joints mounted on the mobile robot and the docking platform demand a precision of 2-3 mm for the position and 1-3 degrees for the orientation to accomplish the docking task. Sensor integration is employed to evaluate the accuracy and robustness of the pose estimation for sensors with respect to the docking platform.

However, multiple sensors are not the only aspect of the high accuracy in the docking of the robot. The control method which has the feedback from the sensor measurements is a bridge between the sensor data and the desired docking behavior. The autonomous docking of the mobile robots demands accurate sensor measurements cooperating with the control design. In this paper, the feasibility of the laser scanners and the vision sensors are investigated to obtain a docking policy for the docking trajectory and low average docking time.

I.3. Software and hardware integration

The mobile robot is Rob@work 3, with three degrees of freedom for 2D transformation and 1D rotation for planar motion. An omnidirectional platform enables the Rob@work 3 to employ arbitrary velocity and rotational commands to move freely in constrained places while controlling the kinematic chain of the platform along all

Manuscript received January 2016, revised February 2016

directions. The platform contains physical components, including sensors, actuators and an on-board computer. A good performance demands a desirable control for driving and steering of actuators as well as other components and setups on the docking platform such as end-effectors and collinear joints.

The platform has two radial safety laser scanners S300 manufactured by SICK AG mounted diagonally to perceive the environment and monitor dangerous indoor areas. The benefit of such design is to implement protective fields for all-round protection. Therefore, if the Rob@work 3 moves to hazardous states in an indoor environment, the control system which is established based on sensor measurement will avoid collision.

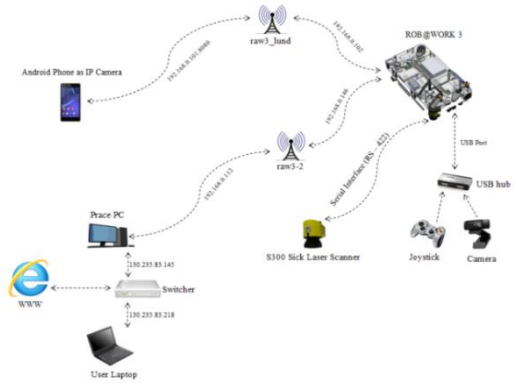


Fig. 1. Software and hardware overview for docking of Rob@work 3

The Android and the USB cameras mounted on the front side of Rob@work 3 are employed to increase the accuracy of the docking. In the meantime, visualization time to detect the target is compared with different vision sensors. The Android phone is used as the IP camera to embed live video as part of the main program for the pose estimation. The stream from the IP camera displays the docking area with the marker attached to the docking platform implying a target with a fixed coordinate system.

II. Methodology

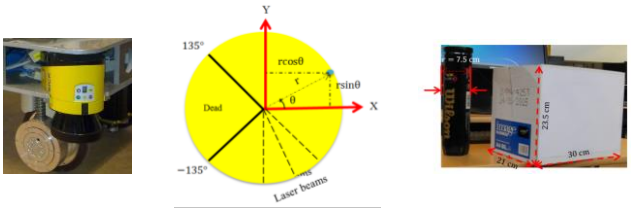
In this paper, two different methods are evaluated to accomplish docking of the mobile robot. They are based on the different sensor data measurements from laser scanner and vision sensors respectively to increase the accuracy of docking the Rob@work 3.

II.1. Autonomous Laser Scanner-based Docking

Knowing the position of the robot in a known or an unknown environment is a critical aspect since it will be used for further goal investigations and accurate pose estimation makes the goal achievement more robust. Two-dimensional laser scanners are widely used to obtain the pose estimation of the mobile robot in the environment. Laser scanners has a capability of the more accurate measurement to detect and avoid obstacles in walking, line following and pose estimation in the indoor environment. Docking of the Rob@work 3 with laser scanner demands a detectable marker attached to the docking platform, acting as a reference point to identify the docking platform. The laser sensor perceives the environment to detect all possible objects within the angular range of $[-135°, 135°]$ inside the docking area.

In this approach, two simple structures such as cylindrical bottle ($r = 7.5\ cm$) and a box ($30 \times 23.5 \times 21\ cm^3$) are employed to identify the docking platform as the reference marker for analyzing the raw data obtained from the laser scanners.

The SICK S300 is a two-dimensional radial laser scanner operating at 12V and 1A that measures 535 distance points and corresponding 534 laser beams within the range of $[-135°, 135°]$ where $\theta = 0°$ points to front side of the sensor and the middle laser beam. It is called radial laser scanner since the area to be monitored is scanned radially and cannot see through the objects during this process. The area behind the object is called dead zone since nothing is monitored. Besides of an angular range, the sensor has also a position range which is considered as distance to the obstacles, measured in meter.



a) Front view b) Geometry c) Different markers

Fig. 2. Sick S300 laser scanner

According to Fig. 2 b), the radial distance to surrounding objects can be obtained in the two-dimensional workspace as follows:

$$(x, y) = (r\cos\theta, r\sin\theta) \quad (1)$$

in which r is the radial distance to the objects and θ is the counter clockwise angle with respect to the horizontal axis while laser scanner perceives the environment.

The constant angle increment (θ_{inc}) of the laser scanner under investigation is computed based on the minimum (θ_{min}) and maximum (θ_{max}) angular range of $[-135°, 135°]$ with the certain number of laser beams ($lb = 534$) as follows:

$$\theta_{inc} = \left(\frac{\theta_{max} - \theta_{min}}{lb}\right). \quad (2)$$

The laser beam angle (θ_b) is identified as a vector with

an index i corresponding to number of laser beams (0, 1, 2 … 534) for the sensor as follows:

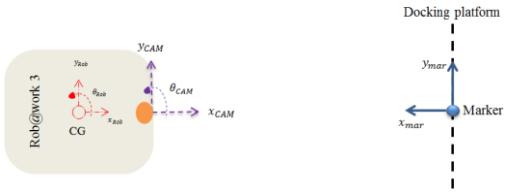
$$\theta_b(i) = \theta_{min} + (i \times \theta_{inc}). \quad (3)$$

The raw data of the two-dimensional laser scanner sensors on the Rob@work 3 are evaluated in different experiments to check their capability to accomplish the docking of the mobile robot precisely. The raw measurement data are the radial distances to the surrounding objects on the docking area.

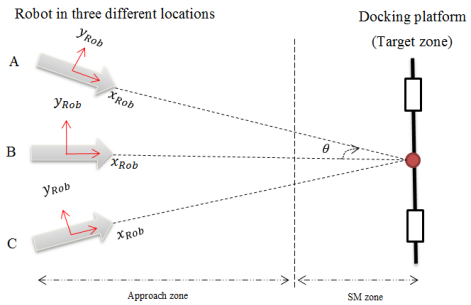
II.2. Computer Vision

Similar to eyes, cameras are helpful and effective in robotics since vision allows noncontact measurement in various domains, including object recognition, localization and manipulation [1]. The camera is practically employed as a vision sensor mounted on the front side of the Rob@work 3 to observe the environment and retrieve the visual information of the docking platform. The position of the camera is totally independent of which control configuration is employed as long as it visualizes the local environment, however, it is necessary to calibrate camera before the visualization process started and determine the geometric parameters of the camera. In the computer vison, the geometric camera parameters, including intrinsic and extrinsic matrices, are calculated for the calibration process [2].

The visual data acquisition and image processing task are executed in the mobile robot computer which also takes the control of omnidirectional actuators.



a) Coordinate systems



b) Docking area with different zones

Fig. 3. Graphical representation of Rob@work 3, Camera and marker

An analytical geometry also known as coordinate geometry is used to determine the camera correlation with the captured marker in a fixed coordinate system in which pose estimation of the robot with its onboard camera are determined accordingly. The geometric modeling is categorized to three different coordinates in real-time, collaborating in localization and docking, namely the mobile robot, camera and marker.

According to Fig. 3 a), y_{CAM}, y_{Rob} and y_{mar} are chosen such that they all have same direction to decreases the problem complexity in terms of the coordinate adaptation. It simplifies the control signal computation after the pose estimation. The advantage of such coordinate system is that orientation would enable the robot to approach the docking platform from different angles rather than only a primitive perpendicular case in which the robot is meant to approach the goal directly and the orientation is usually adjusted by the operator. Besides, vision feedback control is necessary to control orientation after extracting measurement of the sensor. Therefore, it facilitates the approach from more than one initial configuration toward the docking platform.

In this paper, the docking area is divided into three different zones in order to design a more precise controller for the aside position (y_{Rob}) and the orientation (θ_{Rob}): approach zone, safety margin (SM) zone and target zone. It is assumed that the Rob@work 3 always starts the docking process somewhere in the approach zone which is far from the docking platform along the x_{mar}-axis (approximately 18 cm). In this zone, the Rob@work 3 is allowed to have higher velocities. The SM, on the contrary, is defined approximately 18 cm from the docking platform to adjust the orientation along the θ_{mar}-axis, the aside position along the y_{mar}-axis, and the smoothness to accomplish higher accuracy before docking is completed.

In this approach, various configurations were investigated to compute the approprite steering and driving velocity commands for the actuators. According to Fig. 3 b), adjusting a movement along the y_{Rob}-axis and orientation along the θ_{Rob}-axis is more crucial than movement along the x_{Rob}-axis, since robot can move toward the docking platform even if a constant driving velocity ($Vel_x^{App.}$) is applied. Therefore, the control signal along the x_{Rob}-axis is the constant velocity when the robot moves toward the docking platform even though values could differ from zone to zone.

In this paper, the pose estimation of the camera with respect to the fixed marker on the docking platform is employed for the navigation and the localization to accomplish the docking without collision and misalignment. The vision sensor and marker are two main components in which the target is identified by the image features of the marker. Both components have coordinate frames and their origins and rotations are broadcasted with respect to the Rob@work 3.

The employed marker is a square-board ARTag with

definite shape and ID used to identify the docking platform. The chosen marker has a length of 80 mm with 8 mm of paddle as a thickness with white border. It is attached to the center of the docking platform and exactly between of end-effectors. Therefore, the camera position is estimated with respect to the fixed marker.

a) Original image b) Threshold image

Fig. 4. Detected ARTag in docking platform

According to Fig. 4 a), the target is assumed to be at the origin of the marker to simply initialize the reference values ($y_{mar} \cong 0$ m and $\theta_{mar} \cong 0°$). The ARTag identifies the docking platform since it has high capability of detecting the target even in environments consists of several disturbances on the background as depicted in Fig. 4 b). The ARTags also depict higher transparency to detect objects from further distances in the environments with which challenging illuminations exist in the workspace.

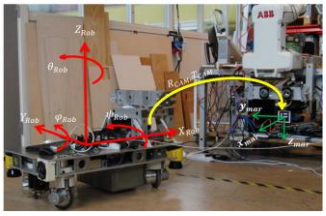


Fig. 5. Rob@work 3 and docking platform coordinate system

As it is shown in Fig. 5, the camera position is estimated in the fixed marker coordinate system. In this context, marker rotation ($\vec{r}_{mar} = [r_x, r_y, r_z]^T$) is converted to the rotation matrix ($R_{3\times3}$) to determine the orientation of the camera with respect to the fixed marker.

$$\theta = |\vec{r}_{mar}| = \sqrt{r_x^2 + r_y^2 + r_z^2} \tag{4}$$

The $R_{3\times3}$ is derived based on the norm of the marker rotation vector (θ) and rotation vector ($\vec{r}_{mar}$) as follows:

$$R_{3\times3} = \cos\theta I_{3\times3} + (1-\cos\theta)\, r_{mar} r_{mar}{}^T + \sin\theta \begin{bmatrix} 0 & -r_z & r_y \\ r_z & 0 & -r_x \\ -r_y & r_x & 0 \end{bmatrix}. \tag{5}$$

If $R_{3\times3}$ is simplified as follows:

$$R_{3\times3} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}, \tag{6}$$

the camera orientation (θ_{CAM}) is derived as :

$$\theta_{CAM} = tan^{-1}(\frac{r_{32}}{r_{33}}). \tag{7}$$

Since the Rob@work 3 has a planar motion, only applicable angle is yaw (θ_{CAM}). As such, the rotations along the x_{Rob}-axis and y_{Rob}-axis are eliminated. The positive orientation measurement resembles to the counter clockwise rotation, whereas clockwise rotation is equivalent to the negative one. Furthermore, if the mobile robot is placed perpendicular to the platform, the yaw is equal to zero ($\theta_{CAM} \cong 0°$). The camera position is estimated as follows:

$$X_{mar} = R_{CAM} X_{CAM} + T_{CAM} \tag{8}$$

in which $X_{mar} = [x_{mar}\ y_{mar}\ z_{mar}]^T$ and $X_{CAM} = [x_{CAM}\ y_{CAM}\ z_{CAM}]^T$.Translation ($T_{CAM}$) and rotation of the camera (R_{CAM}) are calculated with respect to the fixed coordinate system set on the marker attached to the docking platform as graphically sketched in Fig. 5. The camera is practically mounted on the front and middle side of the Rob@work 3. Therefore, T_{CAM} is essentially considered at the origin.

$$R_{CAM} = \begin{pmatrix} \cos\theta_{CAM} & -\sin\theta_{CAM} & 0 \\ -\sin\theta_{CAM} & \cos\theta_{CAM} & 0 \\ r_{31} & r_{32} & 1 \end{pmatrix};\ T_{CAM} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \tag{9}$$

Therefore, the 2D position and the 1D orientation of the camera are calculated in the fixed marker coordinate system since the mobile robot moves in a plane. The camera pose estimation can be simplified as follows [3]:

$$\begin{cases} x_{mar} = x_{CAM}\cos\theta_{CAM} - y_{CAM}\sin\theta_{CAM} \\ y_{mar} = -x_{CAM}\sin\theta_{CAM} + y_{CAM}\cos\theta_{CAM} \end{cases} \tag{10}$$

At this stage, the camera is already calibrated and its parameters are already derived. The detection process of fiducial and pose estimation approximately runs at 30 frames per second (fps) and can be applied in every frame.

II.3. Autonomous Vision-based Docking

A mobile robot equipped with vision sensors can potentially obtain an accurate manipulation and achieve the goal because of a noncontact evaluation of the environment [4]. The designed control system theoretically computes precise control signals if the input measurements have enough accuracy in the system.

In this approach, the visual information of the docking platform is used as a feedback to increase the overall accuracy of the control system. Two images are virtually involved, first is the target with an estimated position of

the docking platform, whereas second image acquires the current position of the camera and robot in real-time. This approach constantly evaluates the current position of the vision sensor with respect to the target to compute appropriate control signals.

The visual system provides position and orientation references for the control system. Therefore, position-based visual servoing (PBVS) basically serves as the feedback of the control system to extract the features of the image and compute the difference between the current positions of the mounted camera at the Rob@work 3 with the fiducial markers as the target. The PBVS approximately estimates the position and orientation of the camera coordinates with respect to the fixed marker, implying the target and sends appropriate the control signals to the mobile robot [5].

The target plays a crucial role to design the vision-based control system since the current position of the robot is evaluated online to provide appropriate control signals to eventually eliminate the position and orientation errors. Therefore, accurate measurements for the reference point leads to more appropriate the control signals.

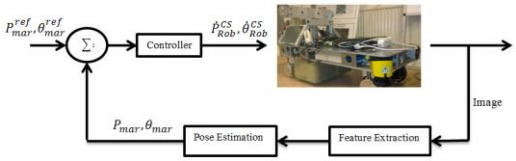


Fig. 6. Employed PBVS for docking the Rob@work 3

As it is shown in Fig. 6, vision-based closed loop control design is employed to compute the control signals for the Rob@work3. The reference positions ($P_{mar}^{Ref}(x_{mar}^{Ref}, y_{mar}^{Ref})$) and the orientation ($\theta_{mar}^{Ref}$) are determined when the Rob@work 3 is manually docked and set as the reference signals. The reference is constantly compared with estimated positions $P_{mar}(x_{mar}, y_{mar})$ and orientation (θ_{mar}) after the feature extraction of the markers to calculate the error signal as the input to the controller to obtain the robot commands.

The plant of the control system is the Rob@work 3 which contains the mobile platform as the process and four omnidirectional wheels as the actuators. The sampling period of the embedded control system on the Rob@work 3 is approximately 20 milliseconds to send the appropriate steering and driving velocity commands to the actuators and move the platform [6].

The control system is designed such that the Rob@work 3 always moves toward the docking platform to adjust X_{mar}. Furthermore, the movement along the x_{Rob}-axis is prohibited as soon as the Rob@work 3 enters the SM zone ($Vel_x^{SM} = 0$). This strategy is considered to ensure the camera position and orientation are precisely adjusted compared to the fixed marker coordinate system. Therefore, the thresholds values ($y_{thresh} = 2.5\,\text{mm}$ and $\theta_{thresh} = 2°$) are used to terminate the controllers and switch to forward movement along the x_{Rob}-axis again to complete the docking task.

The vision-based feedback control is designed such that the target is accomplished in the shortest average docking time. The time is initially recorded on the embedded controller of the Rob@work 3 via the ROS client library package during the data acquisition process. It is converted to the real time to give a better sense of docking time since the ROS framework has its own time configuration to compute the control signal. The conversion is done with an initial time (t_i^{ROS}), a final time (t_f^{ROS}), and recorded samples of the ROS time (t_s^{ROS}).

$$\Delta t = t_f^{ROS} - t_i^{ROS} \tag{11}$$

Each step is mathematically derived as the difference between two consecutive samples (Δt). The vector of real times (t_{real}), on the contrary, is calculated as follows and utilized for further computations and control design:

$$t_{real} = 0 : \frac{\Delta t}{t_s^{ROS}} : \Delta t \tag{12}$$

The error is the difference between the target (P_{ref}) and the current position of the camera (P_{cur}) in the fixed marker coordinate system calculated below:

$$e(t) = P_{ref} - P_{cur} \tag{13}$$

and control signal is defined as [7]:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \tag{14}$$

in which K_p , K_i , and K_d are proportional, integral and derivative gains, respectively. The control signal is sent to the embedded controller in the Rob@work 3 as a twist to move the platform toward the docking platform.

III. Experimental Results

This section presents the obtained experimental results from the autonomous docking of mobile robot with laser scanner and vision sensor, evaluated on the Rob@work 3. Data acquisition of vision sensor and laser scanner is synchronized even though the controllers are totally decoupled for these sensors, meaning that a change in data obtained by one sensor does not result in retaliation effect for another sensor.

III.1. Autonomous laser scanner-based docking

In this section, the results of several experiments for docking the Rob@work 3 using the laser scanners are presented. The experiments are conducted to evaluate the precision of the laser scanner to dock the Rob@work 3. In each experiment, approximate radial distance to surrounding objects with respect to the beam angle (θ_b) and top view distance are plotted (see Fig. 7). The Angle is calculated from the index (i) and the angle increment (θ_{inc}) of the laser beams as derived in (2) and (3).

The first experiment evaluates the laser scanner capabilities to detect objects with nominal distance range provided in the S300 Sick datasheet. In this experiment, no marker was attached to the docking platform and the mobile robot is placed in the docking area (see Fig. 7 a) and Fig. 7 b)). The objects with distance less than or equal to 10 cm from the Rob@work 3 are considered as obstacles and terminate the controller. Table 1indicates the theoretical range of the laser scanner sensor which has been approved by the conducted experiment.

TABLE I
LASER SCANNER CAPABILITY TO DETECT OBJECTS

Component	Manufacturer Datasheet	Experiment
Nominal Distance [m]	0-30	0-30

In the second experiment, the Rob@work 3 was manually docked and cylindrical bottle and a box were placed on the docking platform.

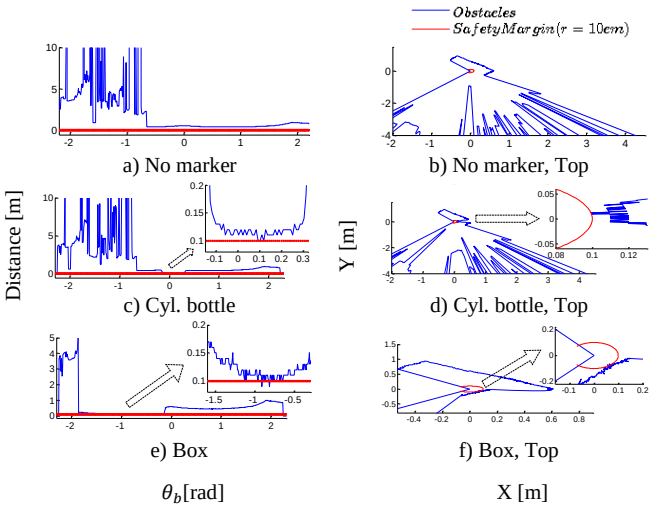


Fig. 7. Laser scanner analysis for marker detection

According to Fig. 7 c) and Fig. 7 d), several intersections with the safety margin were recorded which practically implies the cylindrical bottle and box were identified in various beam angles and positions as the target. The result of laser scanner to detect the box, shown in Fig. 7 e) and Fig. 7 f), is more inaccurate than cylindrical bottle since more samples with the distance less than or equal to 10 cm were detected. Since the length of the box is larger than the radius of the cylinder, more laser beams were occupied to detect the marker. It is detected with 1-2 cm accuracy by the front laser scanner.

To prove the laser sensor incapability of docking the Rob@work 3 with high precision, another experiment was conducted to detect the cylindrical bottle as the target in the docking platform considering the fact that neither the marker nor the mobile robot were replaced or moved during the experiment (see Fig. 8). In fact, the same experiment with cylindrical bottle presented in Fig. 7 c) and Fig. 7 d) was repeated to ensure the sensor can detect the target at the same samples.

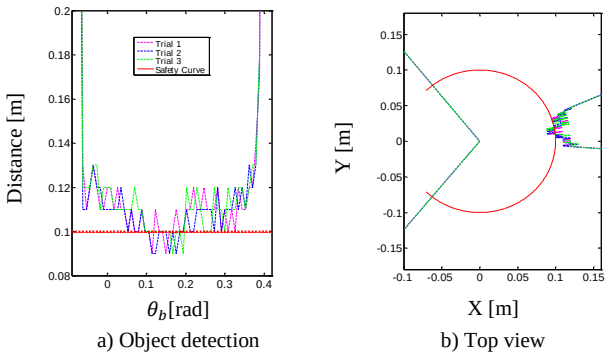


Fig. 8. Cylindrical bottle detection with laser scanner

According to Fig. 8, several samples detected as the target in each trial meaning that the laser sensor identifies the target in different positions in each trial. This implies less reliable measurement for the feedback of the control system if the laser scanner is employed to accomplish docking. Therefore, another sensor might be crucial to increase the docking accuracy of the Rob@work 3.

III.2. Autonomous vision-based docking

Vision sensor is employed to increase the accuracy of the localization and docking of the Rob@work 3 while it is placed in different initial configurations. In this approach, two different vision sensors evaluate the docking precision of the Rob@work 3. The evaluation is based on the sampling time for the recorded ROS time and real-time, computed from equations (11) and (12).

Since the video stream is via an IP address on the shared network between the Rob@work 3, the Android phone and the local PC, the data transformation delay is essentially larger than the USB camera and yields jitters resulting in shaky movement of the Rob@work 3 toward the docking platform. The task takes approximately 70-80 ms to detect the ARTag and compute the control signals if the USB camera is employed as the vision sensor, whereas it is approximately double (130-140 ms)

if the IP camera is utilized (see Fig. 9).

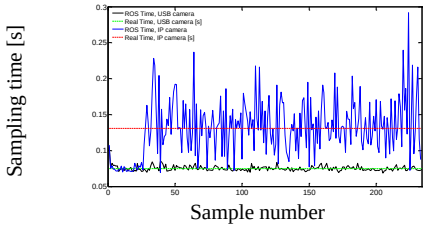


Fig. 9. Sampling time comparison for IP and USB camera

According to Fig. 9, the sampling time for the USB camera exhibits fewer oscillations than the IP camera. The network between the Rob@work 3 and the Android phone is a typical 2.4 GHz wireless band with the maximum speed of 54 Mbps which may not provide enough bandwidth for transmitting the MPEG-4 video stream with a resolution of 640 × 480 pixels and 30 fps. This indicates the reason for the shaky behavior of the mobile robot when the IP camera is employed as the vision sensor in the docking of the Rob@work 3.

As it is shown in Fig. 3 b), the constant velocity along the x_{Rob}-axis can be applied in the approach and SM zones instead of the feedback-vision control. The desired constant velocity to accomplish the docking is evaluated with respect to less average docking time and optimal trajectory. The velocities along the y_{Rob}-axis and the θ_{Rob}-axis, however, are the control signals computed with proportional controllers and set as the input for the embedded controller in the Rob@work 3. The evaluation time is the real-time computed from (11) and (12) and it is considered in the control design even though the experiments were conducted with the ROS time.

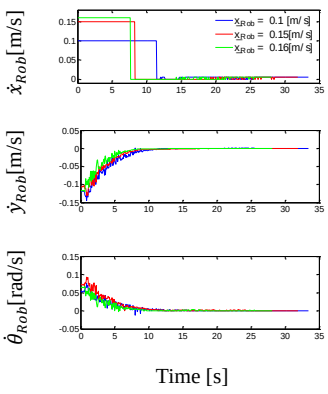


Fig. 10. Constant velocities along x_{Rob}-axis

The corresponding trajectories for the above experiment are depicted in Fig. 11, which contains the whole docking area, the SM zone, and the target which is identified as a circle with the origin of the reference value of the marker ($x_o = -0.2025$ m, $y_o = -0.019$ m) and the radius of the threshold along the x_{mar}-axis ($x_{thresh} = 1$ mm).

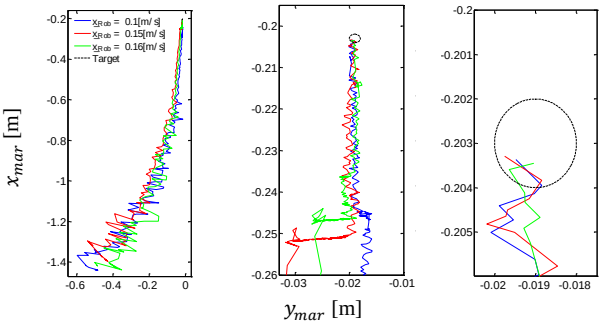


Fig. 11. Docking trajectories for constant velocities along x_{Rob}-axis

According to Fig. 10 and Fig. 11, forward velocity is chosen such that the approach time and the offset along the y_{mar}-axis is a compromise between the desired trajectory and less average docking time. Therefore, $\dot{x}_{Rob} = 0.15\,\frac{\text{m}}{\text{s}}$ is selected as the forward velocity for the further control design. This constant velocity, however, is only applied when the mobile robot is inside the approach zone. In the final zone in which movement along the y_{mar}-axis and the θ_{mar}-axis is adjusted, very small constant velocity of $\dot{x}_{Rob} = 6\,\frac{\text{mm}}{\text{s}}$ is applied to finalize the docking.

In control design, camera pose estimation with respect to the fixed marker computes the measurement errors and control signals along the y_{mar}-axis and the θ_{mar}-axis to accomplish the docking. Four PID controllers with different gains were implemented to evaluate the desired docking behavior of the Rob@work 3 with respect to time and trajectory.

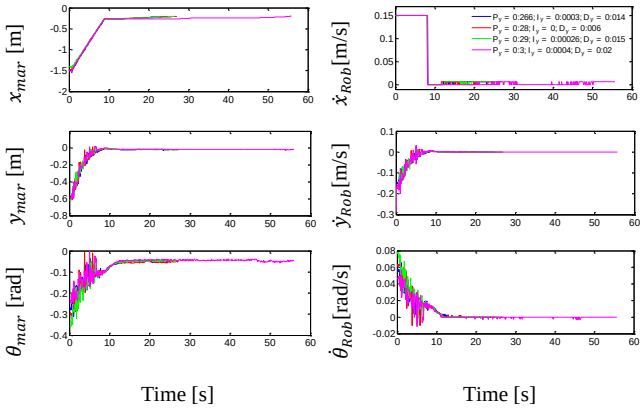
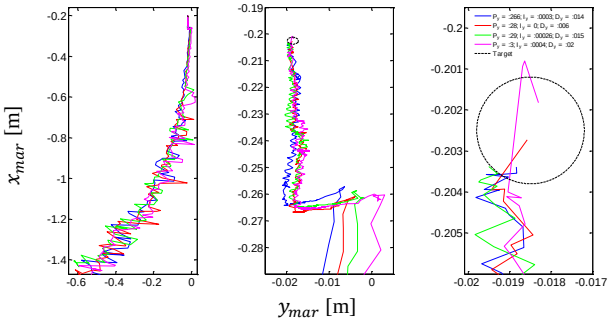


Fig. 12. Camera pose estimation and control signals

Theoretically, the slope of the position-time plot indicates the velocity of the mobile robot. The constant velocity along the x_{Rob}-axis is the derivative of the marker position along the x_{mar}-axis, shown in Fig. 12. The corresponding docking trajectories of the Rob@work 3 with different controllers are presented in

Fig. 13 which are signals after applying transformation and rotation matrices to the raw measurements of the image processing.



a) Approach + SM + Target b) SM + Target c) Target

Fig. 13. Docking trajectories for various vision-feedback controllers

The SM zone declares the obtained offset along the y_{mar}-axis for different controllers. The target zone is identified as a circle with an origin of the reference values, recorded while the Rob@work 3 was manually docked $(x_{mar}^{Ref}, y_{mar}^{Ref}, \theta_{mar}^{Ref})$ and radius of the threshold along the x_{mar}-axis ($x_{thresh} = 1$ mm). The desired average docking time and trajectory of the Rob@work 3 can be evaluated with the conducted experiments and summarized in TABLE II.

TABLE II
GAIN TUNING FOR VISION-BASED CONTROL IN DOCKING OF ROB@WORK 3

Experiment	DOCKING TIME (S)	Offset (y_{mar}-axis) [m]	Successful docking
1(Blue)	24.88	0.01411	Yes
2(Green)	26.61	0.01605	Yes
3(Red)	27.01	0.01473	Yes
4(Pink)	55.79	0.0201	No

The first experiment, indicated with blue color in Fig. 12 and Fig. 13, has the most desired response among the conducted experiments and it is considered for the testing stage in different configurations. To test the designed vision-feedback control, the mobile robot was placed in different initial configurations inside the docking area to evaluate the efficiency and robustness.

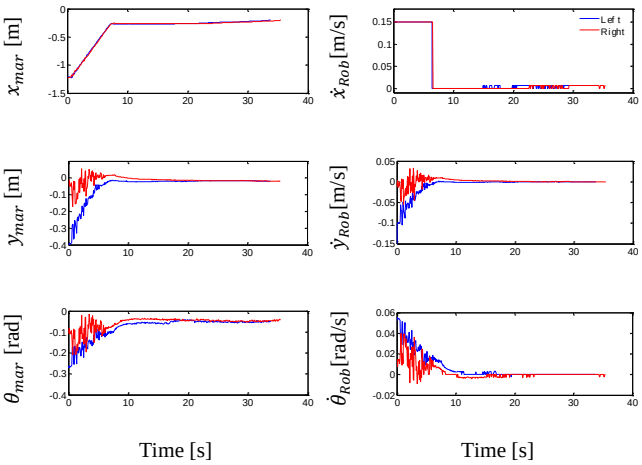
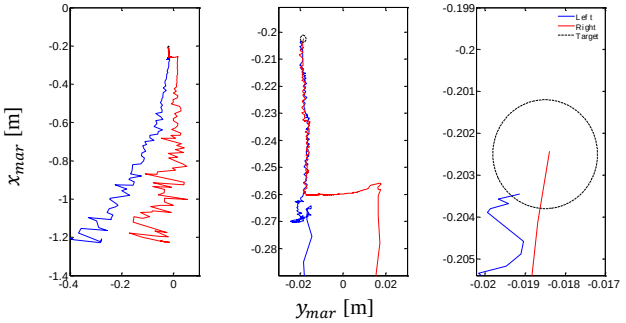


Fig. 14. Camera pose estimation and control signals for testing phase

The docking trajectory of the testing phase is shown in Fig. 15.



a) Approach + SM + Target b) SM + Target c) Target

Fig. 15. Docking trajectories for testing phase

IV. Discussion

The autonomous docking of a mobile robot demands accurate sensor measurement. In this paper, the sensor integration of the laser scanner and vision sensors were employed to obtain the desired behavior with respect to the short docking time and trajectory.

The S300 Sick laser scanner has 1-3 cm resolution to perceive environment and detect obstacles. The conducted experiments on this paper for different simple-shaped markers, however, have shown docking demands a high precision from the sensor measurements and laser scanner measurements have several uncertainties to detect the target appropriately in the data acquisition process.

In the vision-based approach, the raw measurements for pose estimation of the Rob@work 3 are approximated by applying the transformation matrix, then employed to design the vision-feedback control system to increase the precision. The target is represented by the ARTag marker, which is capable of publishing the orientation

(θ_{mar}^{Ref}) besides the position (x_{mar}^{Ref} and y_{mar}^{Ref}). The control signal computations demand accurate measurements of the ARTag for the pose estimation of the camera with respect to the docking platform ($x_{mar}, y_{mar}, \theta_{mar}$).

The visualization time for USB camera is shown to be less than the IP camera since the network to embed the video stream is shared between the Rob@work 3, Android phone and local PC. Therefore, data transformation delay is conspicuous due to the low bandwidth. Consequently, the mobile robot movement is less robust when the IP camera is employed. The average sampling time is computed to eliminate jitters and inconsistent measurements of the ROS time in control design. Although ROS is quite fast for the online task scheduling, it does not always provide the most deterministic timing for certain tasks. Therefore, ROS time does not seem practical for tasks with high frequency PID or motion control.

The experiments with the vision feedback has shown the essence of a control design for a sideway (y_{mar}) movement and orientation (θ_{mar}) over the forward movement toward the docking platform (x_{mar}). Different constant velocities along the x_{Rob}-axis on different zones, have direct impact on the obtained trajectory and the average docking time. Although slower movements along the x_{Rob}-axis in the approach zone decreases the offset along the y_{mar}-axis, the docking time would be considerably higher.

V. Conclusion

The mechanical setups of the tool changers mounted on the docking platform and the Rob@work 3 demand smooth, safe, and flexible motion control with a very precise alignment along the collinear joint axis to accomplish the docking. For this purpose, multiple sensors were employed to evaluate the accuracy of docking with respect to time and obtained trajectory. Docking platform is identified by a detectable marker for each sensor.

The laser scanners depicted several inconsistencies while detecting the marker throughout the experiments. Among the experiments, for instance, neither cylindrical bottle nor the box detected with high accuracy by the sensor when the robot was docked manually within 10cm of safety margin line. Therefore, since high position accuracy is the main concern and the built-in laser scanner sensors are mainly for the safety purposes, a vision sensor is added to the Rob@work 3 to increase the accuracy and robustness with the designed vision-feedback control.

Initially, camera position is estimated in the camera coordinate system. In the process, marker coordinate system is fixed to the docking platform. Therefore, the transformation is applied to the camera frame to estimate the camera position in the marker frame which will be published with the ROS transformation package.

The comparison between the results of the autonomous laser scanner-based docking and the autonomous vision-based docking reveals that the vision-feedback control system is more accurate and robust for docking of the Rob@work 3 considering desired docking with respect to time and trajectory.

References

[1] P. Corke, *Robotics, Vision and Control Fundamental Algorithms in MATLAB®* (Springer, 2013).

[2] D. A. Forsyth, J. Ponce, *Computer Vision: A Modern Approach* (Pearson Prentice Hall, 2012).

[3] S. M. LaValle, *Planning Algorithms* (Cambridge University Press, 2006).

[4] V. Lepetit, P. Fua, Monocular Mode-Based 3D Tracking of Rigid Objects: A survey, *Foundations and Trends® in Computer Graphics and Vision, vol. 1 n. 1*, 2005, pp. 1 – 89.

[5] W. J. Wilson, C. C. W. Hulls, G. S. Bell, Relative End-Effector Control Using Cartesian Position Based Visual Servoing, *IEEE Transactions on Robotics and Automation, vol. 12 n. 5*, 1996, pp. 684 – 696.

[6] S. Nilsson, *Real-time Trajectory Generation and Control of a Semi-Omnidirectional Mobile Robot*, MSc dissertation, Dept. Automatic Control, Lund Univ., Lund, Sweden, 2010.

[7] K. Ogata, *Modern Control Engineering* (Pearson Prentice Hall, 2010)

Authors' information

[1]MSc student, Department of Automatic Control, Lund University.

Farid Alijani was born on January 4th. 1990 in Iran. He started his MSc in mechanical engineering at Lappeenranta University of Technology, Finland in 2013 and continued at the department of Automatic Control at LTH, Sweden within the Nordic program for MSc thesis.

He has a research interest in intelligent robotics systems, computer vision and machine learning. His main focus is mainly on the autonomous applications for robotic manipulators.

