2008 IEEE International Conference on
Robotics and Automation
Pasadena, CA, USA, May 19-23, 2008

Vision-based Docking Using an Autonomous Surface Vehicle

Matthew Dunbabin
Autonomous Systems Laboratory
CSIRO ICT Centre
P.O. Box 883, Kenmore, QLD 4069, Australia
matthew.dunbabin@csiro.au

Brenton Lang and Brett Wood
School of Engineering
Griffith University
170 Kessels Rd, Nathan, QLD 4111, Australia
{brenton.lang,brett.wood2}@student.griffith.edu.au

***Abstract*— This paper describes the development of a novel vision-based Autonomous Surface Vehicle with the purpose of performing coordinated docking manoeuvres with a target, such as an Autonomous Underwater Vehicle, at the water's surface. The system architecture integrates two small processor units; the first performs vehicle control and implements a virtual force based docking strategy, with the second performing vision-based target segmentation and tracking. Furthermore, the architecture utilises wireless sensor network technology allowing the vehicle to be observed by, and even integrated within an ad-hoc sensor network. Simulated and experimental results are presented demonstrating the autonomous vision-based docking strategy on a proof-of-concept vehicle.**

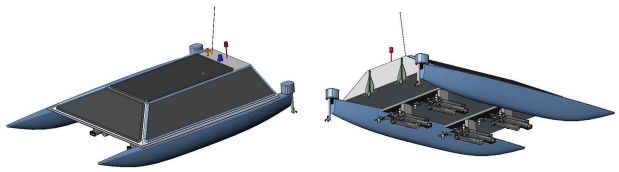
I. Introduction

Traditionally, logistical support for the deployment, operation, communication and localisation of Autonomous Underwater Vehicles (AUVs) has been provided using manned vessels at considerable expense. However, there is now increasing interest in the use of Autonomous Surface Vehicles (ASVs) to provide an automated solution for localisation and communication between surface and subsurface robotic vehicles. For example, moving baseline navigation for the Odyssey AUV was demonstrated using multiple SCOUT ASVs developed by MIT [1], [2]. A similar study was conducted by Pascoal et al. [3] using a catamaran style vessel. Other applications, such as [4], considered the formation control of ASVs for marine security applications.

A recent review of ASV technology by Caccia [5], describes how ASV prototype systems have been developed for both research and military applications. The primary areas of research identified in the literature have focused on vehicle design and construction [1], [6]–[9], navigation and control [10]–[12] and path-planning and obstacle avoidance [6].

However, we have identified an opportunity to develop systems that extend the capabilities of an ASV to deploy, recover and transport AUVs. This would significantly improve the operational logistics associated with offshore underwater and surface robotic systems and is the focus of this paper.

Figure 1 shows an illustration of the CSIRO multi-AUV-carrying ASV concept. The purpose of this robotic system is to carry a number, typically 4, Starbug AUVs [13] to strategic locations for underwater monitoring tasks. Once at the location, the ASV automatically deploys an AUV, then moves to the next location to repeat the process. Once an AUV has completed its mission, it surfaces and rendezvous with the ASV where the two vehicle's coordinate their actions to dock and the AUV is lifted out of the water. The AUV's can then upload data as well as recharge themselves from the ASV's onboard solar panels whilst being transported to the next survey location.



(a) Topside view (b) Underside view

Fig. 1. The CSIRO multi-AUV carrying ASV concept. The top surface of is covered in solar panels for propulsion as well as recharging the AUVs, which are carried on the underside of the ASV.

There are a number of research challenges relating to both the AUV and ASV to realize this coordinated control. Considered here is the first step in enabling an ASV to automatically detect a target, such as an AUV, floating on the surface and robustly manoeuvre itself over the target. As an initial approach to target identification a vision-based solution was adopted.

A recent study by Martins et al [14] also considers a vision system for the docking control of an ASV and they provide some experimental results of the system in docking with a torpedo style AUV. Our solution differs from [14] in that we desire the entire system to operate from a series of distributed low-power processors, and therefore, the vision and vehicle control systems proposed use different approaches for image segmentation and trajectory control.

This paper describes the first step in realising the multi-AUV carrying ASV through the development and experimental demonstration of a coordinated vision-based docking and control strategy for automated retrieval of a surfaced AUV.

A. Paper Outline

The remainder of this paper is structured as follows; Section II gives an overview of the Starship ASV used in this investigation. Section III describes the vision-based target identification system, with Section IV describing the Potential Field vehicle control strategy. Simulation results of the docking strategy are shown in Section V, with the experimental performance in pool and lake trials evaluated in Section VI. Finally, Section VII concludes the paper and proposes further work.

978-1-4244-1647-9/08/$25.00 ©2008 IEEE.

Fig. 2. The Starship ASV prototype.

II. Starship ASV

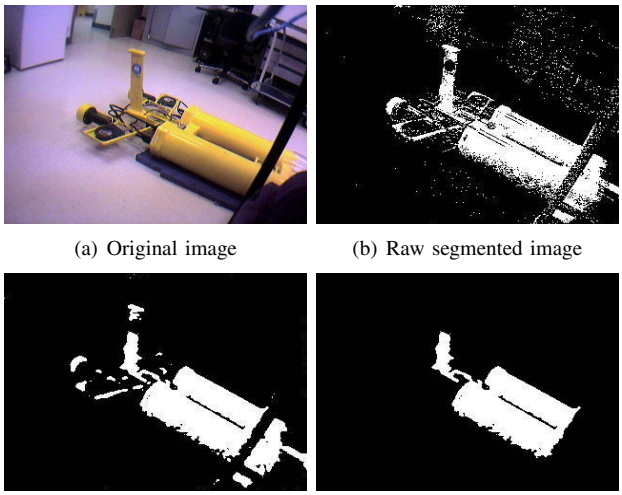
In order to prove the vision and control systems, a small low-cost ASV was developed. The prototype design, known as Starship, is of a twin hull (catamaran style) construction that allows the target, the Starbug AUV [13], to fit between the two hulls for deployment, retrieval and transportation. Figure 2 shows the Starship ASV as used in this study.

The ASV is 2.0m long and 1.3m wide with the hulls constructed from PVC pipe. The vehicle's buoyancy has capacity to lift a 30kg object out of the water. Forward and rotational propulsion is provided by two fixed AUV thrusters, manufactured by Seabotix, located at the rear of the vehicle, one on each hull. The camera system is installed on top of a tripod and slanted downward to allow improved field-of-view for target detection in the water. The vehicle's power is provided by a series of batteries stored in the tubes on the top of each hull.

The system consists of three primary modules; the ASV controller, the vision system, and the base station. The ASV controller uses the Fleck^{TM} [15], [16] wireless sensor network board as the core processor. This controller performs the vehicle control, obstacle avoidance, mission execution, and interfaces with external wireless sensor networks and the base station. Fleck^{TM} GPS and Inertial Measurement Unit (IMU) expansion boards provide the navigation sensing. Furthermore, it has the ability to control the Seabotix thrusters through an I^{2}C bus.

The vision system consists of an OmniVision OV7640 color CMOS camera and a Blackfin 600MHz DSP for local image processing. The vision system communicates the processed target tracking results to the Fleck ASV controller via an RS232 serial link. The system also offers a Ethernet connection for debugging and viewing of raw and segmented images.

To allow remote wireless access to the Starship ASV, a PDA was connected serially to a Fleck^{TM} and used as a base station. The base station allowed rapid development and task assignment for the ASV with the ability to wirelessly set control gains and various docking parameters. Through the addition of a GPS, the base station also has the ability to act as a surrogate docking target for testing purposes. Further details of the hardware and system architecture are presented in [17].



(a) Original image (b) Raw segmented image

(c) Morphologically processed image (d) Final image

Fig. 3. Stages of image segmentation and morphological functions.

III. Vision System

The target identification and tracking system was implemented on the Blackfin processor running a Linux operating system. The image capture and processing software was written using the DDX middleware [18]. The ASV vision system has four primary functions: (1) Target segmentation, (2) lens distortion correction, (3) image to global coordinate transformation, and (4) robust target tracking.

A. Target Segmentation

As the desired targets are primarily yellow in color, such as the AUV or a life-jacket, a color-based image segmentation technique was employed. The captured 640x480 pixel RGGB Bayer image is subsampled to 320x240 pixels and converted to the HSV color space for segmentation. HSV [19] was chosen due to better color constancy compared to other representations such as YUV. As yellow appears within an angular range in HS space, each pixel was evaluated and accepted if it was within a pre-specified color range. Figure 3(a) shows an example image showing the AUV target with Figure 3(b) showing the raw segmented image.

A dilation then erosion morphological operation is performed on the raw segmented image to remove random noise as shown in Figure 3(c). Finally, a segmented "blob" clustering strategy as described in Section III-D, is used to select the final target. Figure 3(d) shows the resulting segmented target for the example image. This entire image segmentation process ran at approximately 2Hz on the Blackfin processor.

B. Distortion correction

To correct the effects of lens distortion in the image, the radial distortion model [20] was used and implemented in software using the OpenCV library. To improve processing

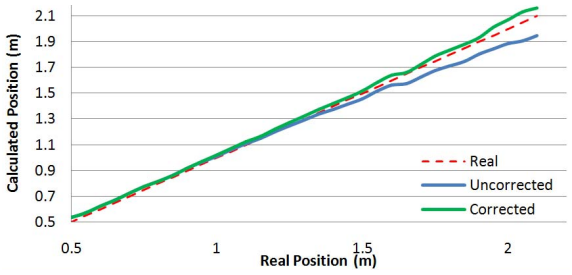


Fig. 4. Effect of target position estimate with distortion correction using the ASV vision system.

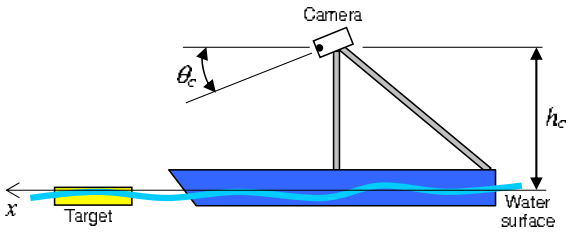


Fig. 5. Geometry of the vision system on the ASV.

performance, only the segmented pixels from the previous section were corrected and the target area recalculated and the centroid of the target estimated.

The effect of distortion on the estimated position of the target with respect to the vehicle is illustrated in Figure 4. This figure shows that without distortion correction, the estimated target centroid position can be up to 5% in error from the actual measured position. Here the estimated target position is calculated using the method described in Section III-C.

C. Coordinate Transform

Figure 5 shows the basic geometry for the vision system on the ASV where the camera is located at a height h_c above the surface plane and is tilted at an angle θ_c down from the horizontal. The origin of the vehicle (and camera) centred coordinate system is directly below the camera on the water's surface with the x-axis forward, and the y-axis pointing out of the page.

The coordinates of the target centroid with respect to the vehicle coordinate system (x_t, y_t) are given by

$$x_t = \frac{h_c}{\tan\left(\theta_c - \tan^{-1}\left(\frac{v_t}{f_v}\right)\right)} \quad (1)$$

$$y_t = \frac{-u_t \; d_{t_x}}{f_u} \quad (2)$$

$$d_{t_x} = \sqrt{h_c^2 + x_t^2} \quad (3)$$

where u_t and v_t are the centroid coordinates of the segmented image as determined in Section III-B (origin at the center of the image), and f_u and f_v are the camera focal lengths in the image u- and v-directions respectively.

To account for the motion of the ASV during experiments, the target position was transformed to the global coordinate

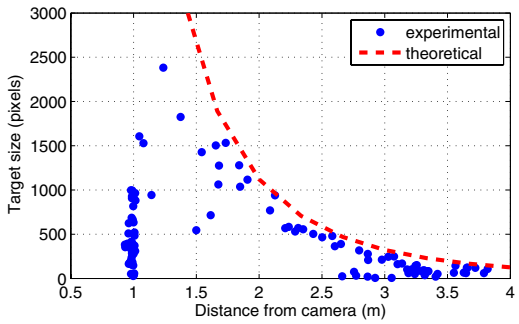


Fig. 6. Measured and expected target size with distance from ASV vision system for 3 docking missions.

system by a Euler angle homogeneous transform using the measured roll, pitch and yaw from the Fleck^{TM} IMU.

D. Target Tracking

Target selection is based on consideration of the n_b largest segmented "blobs" within the image ($3 \leq n_b \leq 8$). The area of each blob is scaled based on its distance away from the vehicle to compensate for camera perspective effects. A clustering algorithm was developed which maintains a register of these scaled blobs in close proximity to each other. The algorithm estimates the area of each cluster and its vehicle centred centroid coordinates. A validity measure is given to each cluster based on its temporal stability within the image and its size compared to the expected target size. The cluster with the highest validity measure is selected as the target.

Figure 6 shows a comparison between the observed target image size and a theoretical prediction of the image target size with distance from the camera during a series of vision-based docking manoeuvres. As seen, the selected target size corresponds well for distances greater than 1.5m. At distances below 1.5m, the target begins to move beyond the camera's field-of-view and an underestimate of the area results. However, in general, the vision and clustering tracking algorithms are capable of reliably determining the size and location of the target.

IV. ASV Docking Control

In order to dock with a target of interest, the ASV must be capable of moving from its current pose, given by $\mathbf{P} = (x_{ss}, y_{ss}, \psi_{ss})$, to a desired global position and orientation angle, $\mathbf{P}^* = (x^*, y^*, \psi^*)$. Here the Cartesian position of the vehicle can be measured using either GPS or by resolving the relative vision-based position estimate into the global coordinate system. The heading is measured from the Fleck^{TM} IMU.

A. ASV Docking Motion Control

Due to the limited processing power of the ASV Controller, Potential or Virtual Force Fields rather than a dedicated path planner was chosen for vehicle motion control. Virtual force fields have been used extensively for robot motion control [21]. Their advantage in this application is

they are computationally simple to implement, and allow the introduction of additional virtual forces to protect the target from collision or avoid obstacles.

In this study there are two virtual forces considered, a linear attraction force (F_a) and an inverse square repelling force (F_r) given by

$$F_a = K_a.r \tag{4}$$

$$F_r = \frac{K_r}{r^2} \tag{5}$$

where r is the Euclidean distance from the ASV center-of-gravity to the action point of the virtual force. The constants K_r and K_a represent the strength of each virtual force.

The docking procedure is divided into two stages; The first stage is used to move the vehicle to within a pre-specified radius of the target, with the second stage providing the mechanism for the ASV to align with the target (ψ^*) and move over the target to complete the docking procedure.

1) Stage 1 Docking: In this docking phase, the ASV and target are separated by a relatively large distance, typically beyond detection by the ASV vision system. In this case, GPS positions and compass heading are typically used to define the target and ASV positions.

Through the wireless exchange of position and heading information between the ASV and target, this stage moves the ASV to a position directly behind the target to assist with the final alignment conducted in *Stage 2*. Figure 7(a) illustrates the *Stage 1* scenario where the actual target location and orientation is given by $(x_{sb}, y_{sb}, \psi_{sb})$, however, the desired position is set at a distance d_1 directly behind the target at point $P1$ where $\psi^* = \psi_{sb}$.

The global Cartesian coordinates of $P1$ are given by:

$$x_1 = x^* + d_1 \sin(\psi^* + \pi) \tag{6}$$

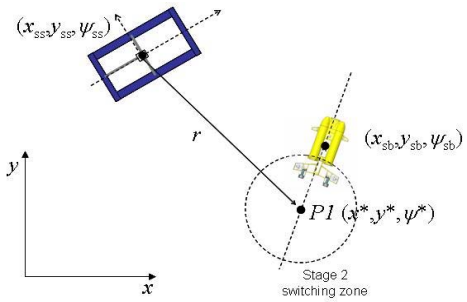
$$y_1 = y^* + d_1 \cos(\psi^* + \pi) \tag{7}$$

In this docking stage, there are two virtual forces acting. The first is a linear attraction force (F_{a_1}) located at $P1$, with the other force, a repelling force (F_{r_1}), located at the actual target center-of-gravity. The repelling force is used to avoid any collision between the ASV and the target. To ensure the virtual forces exerted on the vehicle do not exceed the ASV's propulsion capability, they are each clipped to a maximum thruster force, F_{max}.

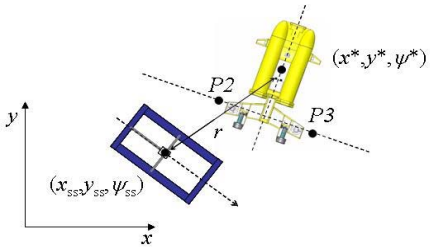
These two virtual forces are then resolved into their respective Cartesian components by considering their line-of-action between their point source location and the ASV, $(F_{a_{1x}}, F_{a_{1y}})$ and $(F_{r_{1x}}, F_{r_{1y}})$ respectively.

The resulting force vector acting on the ASV is the linear superposition of the individual virtual forces resolved in the global Cartesian coordinate system such that $F_x = F_{a_{1x}} + F_{r_{1x}}$ and $F_y = F_{a_{1y}} + F_{r_{1y}}$.

When the ASV distance r is within a pre-specified radius (d_{stage2}) of $P1$, the docking controller switches to *Stage 2*.



(a) Stage 1



(b) Stage 2

Fig. 7. The two stages of ASV docking with a target using virtual force fields.

2) Stage 2 Docking: The second phase of docking is designed to allow the ASV to appropriately align with the target before moving over it to complete the dock. In this case, the vision system, rather than GPS, is typically used to achieve the finer control required to dock with the target.

Figure 7(b) illustrates the *Stage 2* docking scenario. In this case, the desired position is set to the actual target position with a strong linear attracting virtual force set to this location. Additionally, there are two repelling virtual forces set perpendicular to the alignment axes at a distance d_s behind and perpendicular the target as denoted by points $P2$ and $P3$. These global points are defined by:

$$d_s = \max(d_{min}, \quad \alpha\ r) \tag{8}$$

$$x_{2,3} = x^* + d_s \sin(\psi^* \pm \pi/2) \tag{9}$$

$$y_{2,3} = y^* + d_s \cos(\psi^* \pm \pi/2) \tag{10}$$

where d_{min} and α are predetermined constants.

The resulting virtual force vector acting on the ASV is again a superposition of the three *Stage 2* virtual forces resolved into the global Cartesian coordinate system.

B. Thruster Control

The ASV has two fixed parallel thrusters which are independently controlled allowing the vehicle to turn on the spot. In this study it was desired to directly control the thruster forces. The motor forces have, through experiment, been determined to be linearly proportional to their on-board motor controller input.

The control inputs determined by the ASV controller are a vehicle centric forward force and rotational moment (F_v, M_ψ).

Algorithm 1 ASV docking and control procedure.

1) Determine the distance between the vehicle pose ($\mathbf{P}$) and the desired pose ($\mathbf{P}^*$) such that $r = ||\mathbf{P} - \mathbf{P}^*||$.
2) If ($r < 0.2m$) end, else goto 3.
3) If ($r < d_{stage\ 2}$), goto 6, else goto 4.
4) In stage 1: Calculate the linear attracting virtual force at a point behind the target and the repelling force at the COG of the target for obstacle avoidance.
5) Goto 7.
6) In Stage 2: Calculate the virtual repelling forces at points P_1 and P_2 and an attraction force at the ASV COG.
7) Determine the resultant virtual force vector acting on the ASV.
8) Calculate the ASV centric control forces F_v, F_ψ
9) Calculate the individual thruster forces F_L and F_R.
10) Goto 1

These vehicle centric forces are then translated to individual left and right motor forces (F_L, F_R).

The vehicle's thruster controller was designed such that it would be biased towards yaw control. This allows the ASV to appropriately turn towards the target to reduce the total distance travelled. The ASV heading angle control setpoint (ψ_F) is set to the resulting virtual force vector acting on the vehicle such that

$$\psi_F = \tan^{-1}\left(\frac{F_x}{F_y}\right) \tag{11}$$

The demanded vehicle rotational moment (M_ψ) is then calculated using a proportional controller and an equivalent total thruster force to provide this moment (F_ψ) determined such that

$$F_\psi = \frac{2M_\psi}{w} = \frac{2K_\psi(\psi_F - \psi)}{w} \tag{12}$$

where w is the distance between the thrusters, K_ψ is a constant of proportionality. This equivalent rotation force is then clipped such that ($-F_{max} \leq F_\psi \leq F_{max}$).

The remaining available thrust is then converted to forward motion such that

$$F_v = F_{max} - |F_\psi| \tag{13}$$

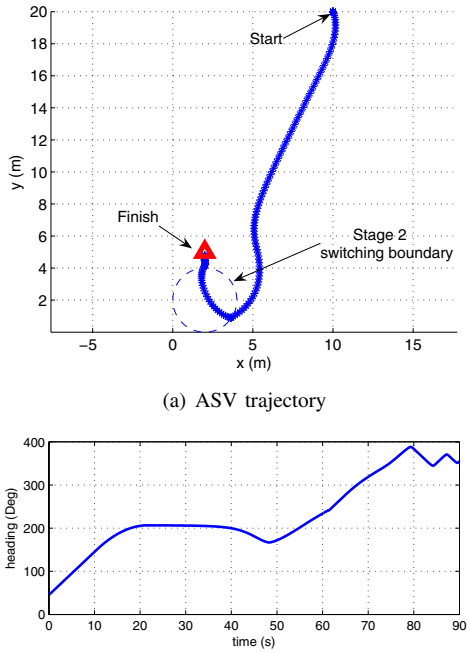
The individual thruster forces are therefore determined by

$$F_L = \left(F_v + F_\psi\right)/2 \tag{14}$$

$$F_R = \left(F_v - F_\psi\right)/2 \tag{15}$$

C. Docking Controller Implementation

Algorithmically, the docking and control strategies presented in previous sections are implemented as shown in Algorithm 1. As seen, this approach is relatively computationally inexpensive which is necessary for implementation on the Fleck^{TM} embedded microcontroller.



(a) ASV trajectory

(b) ASV heading angle

Fig. 8. An example simulation of the docking strategy. The ASV starts with pose $(10, 20, \pi/4)$, with a final desired pose of $(2, 5, 0)$.

V. Simulation Results

A simulation of the system was developed to evaluate the performance of the vehicle controller and docking strategy presented in Section IV. To approximate the dynamics, a simple quasistatic model of the ASV is assumed taking into account maximum forward and rotational speed constraints. The procedure described in Algorithm 1 utilising Equations (4)-(15) was used to determine the control inputs.

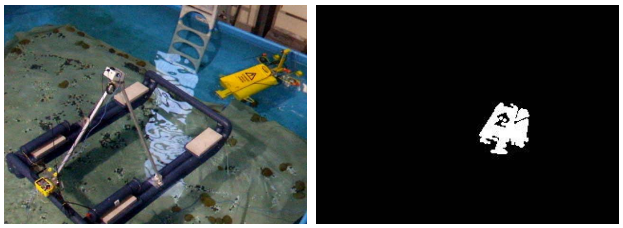
A sensitivity study using the simulation was conducted to determine appropriate control gains and virtual force settings and observe the expected dynamic behaviour of the system. Figure 8 shows the results of a simulation where the initial pose of the vehicle is $(10, 20, \pi/4)$ and the final desired pose is $(2, 5, 0)$, that is point North. As seen in Figure 8(a), the vehicle travels towards a point below the target location and when it reaches the *Stage 2* boundary, moves back up towards the target. Figure 8(b) shows the simulated heading angle of the ASV in this case. It can be seen that at $t = 50$s, *Stage 2* of the docking is initiated and the heading angle then approaches 360 Degrees (North) at the end of the simulation.

VI. Experimental Results

A series of experiments were conducted with the Starship ASV using the vision and docking strategies presented above. In these trials, the Starbug AUV [13] was used as a passive target, however, it could wirelessly communicate its GPS location and heading angle to the ASV for alignment.

A. Pool trials

A series of pool experiments were conducted to validate the vision-based target identification and vehicle control



(a) ASV and AUV in pool (b) Segmented target image

Fig. 9. Typical pool experiment showing the ASV during docking with the AUV. Also shown is the segmented image from the vision system.

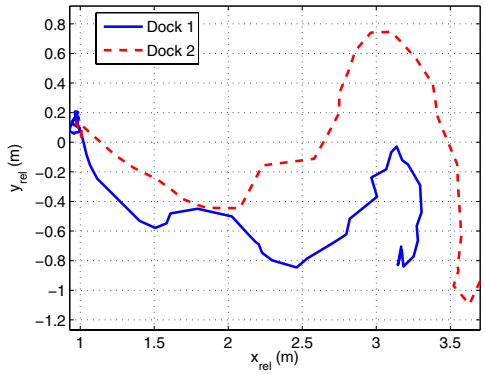
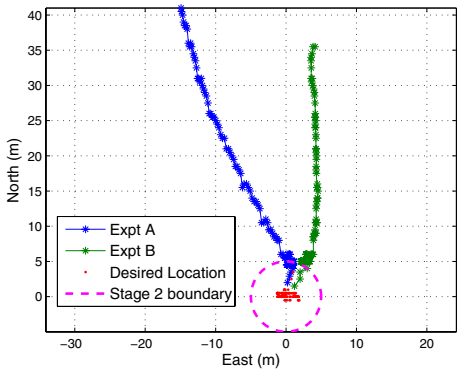
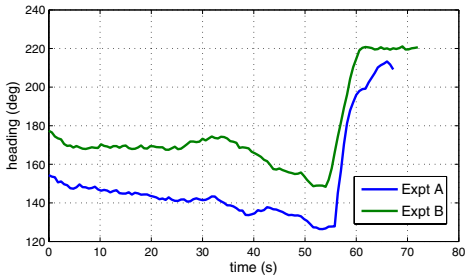


Fig. 10. Measured target position with respect to the ASV coordinate frame during two representative docking experiments in the pool.



(a) ASV trajectory



(b) ASV heading angle

Fig. 11. Measured GPS position and heading angle of the ASV during two lake experiments where the ASV approach direction was close to the desired alignment angle.

systems, as well as the docking strategy. Here the target (AUV) was placed at one end of the pool and the ASV placed randomly elsewhere in the pool. The ASV was then remotely instructed to dock with the target.

Figure 9(a) shows a representative image of the ASV midway through a docking manoeuvre with the AUV where it has aligned with, and is about to moving over the target. Figure 9(b) shows the segmented target image from the vision system for the instance shown in Figure 9(a).

The relative position of the target with respect to vehicle's own coordinate frame is shown in Figure 10 for two representative docking experiments with different initial start conditions. Here it can be seen that the control system manoeuvres the vehicle to get it close to the alignment axis ($y = 0$) before moving forward towards the target.

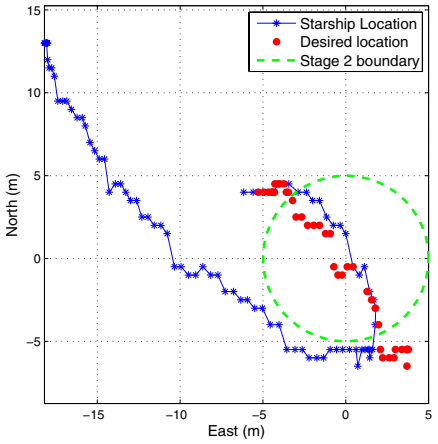
B. Lake trials

Outdoor experiments were performed on Lake Wivenhoe in Brisbane, Australia. Similar to the pool experiments, the target's GPS position and heading angle were remotely communicated to the ASV and the vehicle then commanded to perform a docking manoeuvre. Figure 11(a) shows the recorded GPS track of the ASV during two docking experiments with the same target position but different start locations. The desired alignment angle was 220 Degrees from North. In these experiments the ASV's approach direction to the target was close to the desired alignment angle. Figure 11(b) shows the recorded ASV heading angle during the two docking manoeuvres where it can be seen that after approximately 55s, the second stage of docking commences with the ASV's heading angle then approaching the desired alignment angle.

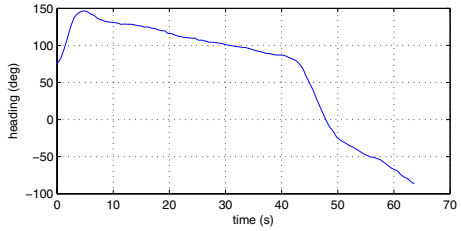
Figure 12(a) shows the recorded GPS track of the ASV during another experiment where the ASV approach angle was out of phase with the desired alignment angle. In this particular experiment, the desired alignment angle was -90 deg, however, the target drifted during the docking manoeuvre in a North-West direction as shown by the desired location position. The location of the stage 2 switching boundary shown is the point where the ASV first reaches the 5m boundary radius. Figure 12(b) shows the recorded heading angle of the ASV during the experiment. Figure 12(c) shows the measured target position relative to the vehicle coordinate frame during the final stages of docking with the target. This experiment illustrates the robustness of the vision system and control strategy in performing accurate docking with a slowly drifting target.

VII. Conclusions

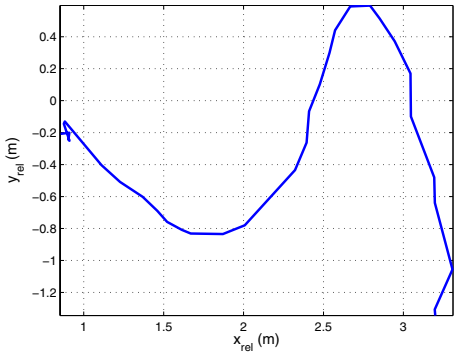
A novel approach to coordinated docking between an Autonomous Surface Vehicle (ASV) and an Autonomous Underwater Vehicle (AUV) has been presented. A real-time vision-based target identification system was developed and implemented on a low-power processor. The processed target coordinates are sent serially to the ASV controller, another low-power processor, which performs all vehicle control, sensor integration, and mission execution. The docking con-



(a) ASV GPS trajectory.



(b) ASV heading angle.



(c) Relative target position with respect to vehicle coordinate frame.

Fig. 12. Lake docking experiment where desired alignment angle is -90 Degrees and the target is allowed to drift: (a) Measured GPS position, (b) heading angle, and (c) the image relative target position with respect to the ASV.

trol strategy, based on virtual forces, allows obstacles to be introduced and avoided. The approach allows docking with passive (immobile) or active targets, that may or may not wirelessly communicate with the ASV. Simulation and experimental trials in a pool and lake have demonstrated that the system is capable of autonomous docking with a surfaced AUV.

ACKNOWLEDGMENT

The authors would like to thank the contributions of other members of the CSIRO ICT Centre Autonomous Systems Laboratory: Cedric Pradaliar, Les Overs, Polly Alexander and John Whitham. Also, thanks goes to Alistair Grinham from the University of Queensland for his assistance during vehicle field trials.

REFERENCES

[1] J. Curcio, J. Leonard, and A. Patrikalakis, "SCOUT - a low cost autonomous surface platform for research in cooperative autonomy," in *Proc. MTS/IEEE OCEANS 2005*, September 2005, pp. 725–729.

[2] J. Curcio, J. Leonard, J. Vaganay, and A. Patrikalakis, "Experiments in moving baseline navigation using autonomous surface craft," in *Proc. MTS/IEEE OCEANS 2005*, September 2005, pp. 730–735.

[3] A. Pascoal, P. Oliveira, C. Silvestre, L. Sebastiao, M. Rufino, V. Barosso, J. Gomes, G. Ayela, P. Coince, M. Cardew, A. Ryan, H. Braithwaite, N. Cardew, J. Trepet, N.Seube, J. Champeau, P. Dhaussy, V. Sauce, R. Moitie, R. Santos, F. Cardiogos, M. Brussieux, and D. Dando, "Robotic ocean vehicles for marine science applications: the european asimov project," in *Proc. MTS/IEEE OCEANS 2000*, September 2000, pp. 409–415.

[4] B. Bishop, "Design and control of platoons of cooperating autonomous surface vessels," in *Proc. 7th Annual Maritime Transportation System Research and Technology Coordination Conference*, November 2004.

[5] M. Caccia, "Autonomous surface craft: prototypes and basic research issues," in *Proc. 14th Mediterranean Conference on Control and Automation*, June 2006, pp. 1–6.

[6] J. Larson, M. Bruch, and J. Ebken, "Autonomous navigation and obstacle avoidance for unmanned surface vehicles," in *Proc. SPIE Unmanned Systems Technology VIII*, April 2006.

[7] J. Manley, A. Marsh, W. Cornforth, and C. Wiseman, "Evolution of the autonomous surface craft AutoCat," in *Proc. MTS/IEEE OCEANS 2000*, September 2000, pp. 403–408.

[8] A. Leonessa, J. Mandello, Y. Morel, and M. Vidal, "Design of a small, multi-purpose, autonomous surface vessel," in *Proc. OCEANS 2003*, 2003, pp. 544–550.

[9] C. Reed, B. Bishop, and J. Waters, "Hardware selection and modelling for a small autonomous surface vessel," in *Proc.38th Southeastern Syposium on System Theory*, March 2006, pp. 196–200.

[10] T. Vaneck, "Fuzzy guidance controller for an autonomous boat," *Control Systems Magazine*, vol. 17, pp. 43–51, 1997.

[11] J. Alves, P. Oliveira, R. Oliveira, A. Pascoal, M. Rufino, L. Sebastiao, and C. Silvestre, "Vehicle and mission control of the DELFIM autonomous surface craft," in *Proc. 14th Mediterranean Conference on Control and Automation*, June 2006, pp. 1–6.

[12] M. Reyhanoglu and A. Bommer, "Tracking control of an underactuated autonomous surface vessel using switched feedback," in *Proc. IECON 2006*, November 2006, pp. 3833–3838.

[13] M. Dunbabin, J. Roberts, K. Usher, G. Winstanley, and P. Corke, "A hybrid AUV design for shallow water reef navigation," in *Proceedings of the 2005 International Conference on Robotics and Automation*, Barcelona, Apr. 2005, pp. 2117–2122.

[14] A. Martins, J. Almeida, H. Ferreira, H. Silva, N. Dias, A. Dias, C. Almeida, and E. Silva, "Autonomous surface vehicle docking manouvre with visual information," in *Proc. International Conference on Robotics and Automation*, April 2007, pp. 4994–4999.

[15] P. Corke, P. Valencia, P. Sikka, T. Wark, and L. Overs, "Long-duration solar-powered wireless sensor networks," in *Proc. Emnets*, Cork, June 2007.

[16] P. Sikka, P. Corke, and L. Overs, "Wireless sensor devices for animal tracking and control," in *Proc. First IEEE Workshop on Embedded Networked Sensors*, Tampa, Florida, Nov. 2004, pp. 446–454.

[17] M. Dunbabin, B. Lang, and B. Wood, "Towards coordinated vision-based docking using an autonomous surface vehicle," in *Proc. Australian Conf. Robotics and Automation*, Brisbane, December 2007.

[18] E. Duff, "DDXVIDEO: A lightweight video framework for autonomous robotic platforms," in *Proc. Australian Conf. Robotics and Automation*, Sydney, December 2005.

[19] A. Smith, "Color gamut transform pairs," in *IGGRAPH '78: Proceedings of the 5th annual conference on Computer graphics and interactive techniques*, 1978, pp. 12–19.

[20] R. Cucchiara, C. Grana, A. Prati, and R. Vezzani, "A hough transform-based method for radial lens distortion correction," in *Proc. 12th International Conference on Image Analysis and Processing*, 2003.

[21] J. Borenstein and Y. Koren, "Real-time obstacle avoidance for fast mobile robots," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 19, pp. 1179–1187, 1989.