Control Engineering Practice 74 (2018) 153–167

Contents lists available at ScienceDirect

Control Engineering Practice

journal homepage: www.elsevier.com/locate/conengprac

Fully automatic visual servoing control for work-class marine intervention ROVs

Satja Sivčev *, Matija Rossi, Joseph Coleman, Gerard Dooly, Edin Omerdić, Daniel Toal

MaREI—Marine and Renewable Energy, Ireland
University of Limerick, Ireland

ARTICLE INFO

Keywords:
Underwater manipulation
Manipulator control
Visual servoing
Robot arm
Underwater inspection and intervention
Marine robotics
ROV

ABSTRACT

ROVs with hydraulic manipulators are extensively used for subsea intervention. With camera feedback from the scene, manipulators are teleoperated and slaved to pilot held master arms. While standard for offshore oil and gas, for challenging applications in waves or currents a new approach is required. We present development of robot arm visual servo control approaches used in manufacturing and the transfer and adaption of these to underwater hydraulic manipulators. This is the first time a visual servoing algorithm for automated manipulation has been developed and verified, through subsea trials, on a commercial work-class ROV with industry standard hydraulic manipulators.

1. Introduction

This paper presents the research and development of semi-autonomous Remotely Operated Vehicle (ROV) manipulator control systems using vision based servo control which are suitable for deployment on the global fleet of work class ROVs. These systems are designed to replace the teleoperation role of pilots with auto-assist functions enabling ROVs to address challenging conditions encountered in emerging sectors such as Marine Renewable Energy (MRE) (offshore wind, floating wind, wave energy conversion and tidal energy conversion).

Work-class submarine ROVs equipped with robot manipulators have been the workhorse of subsea operations for many years in marine sectors such as marine civil engineering, marine science, military and chiefly in the offshore oil and gas industry. A wide range of subsea tasks undertaken by ROVs is done using underwater manipulators, including pipe inspection (Christ & Wernli, 2014), salvage of sunken objects (Chang, Chang, & Cheng, 2004), mine disposal (Djapic et al., 2013; Fletcher, 2000), surface cleaning (Davey, Forli, Raine, & Whillock, 1999), valve operating, drilling, rope cutting (Christ & Wernli, 2014), cable laying and repair, clearing debris and fishing nets, biological (Jones, 2009) and geological sampling (Noé, Beck, Foubert, & Grehan, 2006), archaeological work (Coleman, Ballard, & Gregory, 2003), etc. Work-class ROVs are generally equipped with one advanced seven function manipulator (six degrees-of-freedom plus the jaw/gripper) and one less advanced five function supporting grabber arm. The latter is used to anchor the ROV onto the hydro engineering structure on which the intervention is to take place while the former performs the actual intervention operation. Automatic capabilities of subsea robot manipulator systems are generally significantly lower compared to their industrial robot counterparts. The majority of automated industrial robotic arms used in factories are electrically driven and utilize servo control. Motion of these robots is usually pre-programmed at a high level using dedicated PC software suites. These control/programming environments include full kinematic engines (implementing forward and inverse kinematics) and enable programming servo controlled robots to automatically follow detailed motion control programmes including interaction with target(s). Additionally, advanced robot systems often integrate advanced sensors such as vision systems and visual servoing techniques in order to deal with non-static target objects of various shape, colour, etc., while addressing these target objects in automatic programme operation (Corke, 2011). Another important feature of industrial robotics is that the environment can be controlled and specifically designed and built to ease the robotic automation task, i.e. known fixtures, lighting, etc. Marine field robots by contrast work in real world subsea environments which are significantly more variable and challenging. The majority of commercial underwater manipulators are not servo controlled and none are supported with kinematic engine control approaches. They are predominantly hydraulically driven, and utilize traditional teleoperation approaches with an open loop control system, completely reliant on human operator skill and experience. The pilot who is located on the support vessel acquires visual feedback of the

* Corresponding author at: University of Limerick, Ireland.
E-mail address: satja.sivcev@ul.ie (S. Sivčev).
URL: http://www.mmrrc.ul.ie (S. Sivčev).

https://doi.org/10.1016/j.conengprac.2018.03.005
Received 2 August 2017; Received in revised form 25 January 2018; Accepted 2 March 2018
Available online 23 March 2018

scene through camera and/or forward looking sonar systems and often simultaneously performs multiple tasks: manipulates the robot arm(s), flies the vehicle, or performs underwater inspection (Yuh, 2000). The pilot must handle or interact with an enormous quantity of information dispersed across different screens, constantly looking from camera to gauge distances or check different angles, while also adjusting camera fields of view and lighting. As a result, the operator is often under high cognitive load which can prevent important information from being correctly perceived and result in failed or prolonged missions due to pilot fatigue. Sometimes ROV pilots face dangerous and stressful situations, e.g. British Petroleum ROV fleet working to shut off the well and stop the oil spill in the Deepwater Horizon disaster (Cavnar, 2010). ROV operations are generally not performed in the top 20 m–40 m splash zone but rather in the relatively quiescent conditions below on or near the seabed. By contrast, MRE energy farm plant is located in the splash zone in challenging environments, so the device may even be in motion. As motion disturbances affecting the underwater vehicle and the manipulator become significant, the task execution with a human pilot in the loop becomes difficult and eventually impossible. A human operator can react only after the change has already happened, and therefore even an experienced operator is likely to fail at performing IRM operations in such challenging conditions. With current state-of-the-art commercial ROV control systems, simple tasks from an industrial robotics perspective can become difficult even for a very skilled pilot/operator due to difficulties such as poor visibility, poor 3D perception based on 2D image presented on screen, and pilot fatigue. This makes subsea operations time consuming and therefore very expensive, i.e. the cost of mobilizing a support vessel with ROV systems can cost €18,000 per day for research vessels and well in excess of €50,000 per day for oil and gas touch down operations support.

Despite their significant utility to date in deep water operations, commercial intervention ROV technologies as used in other sectors are not sufficient for operating in shallow waters with high waves and currents. Development of new robotic capabilities is necessary to support large scale MRE operations for construction/roll out, Inspection Repair and Maintenance (IRM), monitoring and control of MRE installations. Such MRE installations are by design located in dynamic, high energy sites where the wind, current and wave energies offshore are maximized. Service robots are essential to allow the nascent MRE sector to develop and grow in an economically viable manner. The IRM operational conditions for MRE will under many circumstances be above operating limits of current ROV platform technology (O'Connor, Lewis, & Dalton, 2013; Omerdic, Toal, & Leahy, 2010). The motivation thus, is to research and develop ROV systems and control techniques for IRM operations in current and wave regimes of increasing strength and specifically deal with challenges in the performance and control of ROVs at high energy MRE sites (Omerdic, Toal, Dooly, Miller, & Coleman, 2012; Toal, Omerdic, & Dooly, 2011). Referring to Fig. 1 we wish to develop robot control capability to move away from the origin in the 3D plot moving along each of the axes.

Since the beginning of the 90's the topic of autonomous underwater manipulation has been attracting the attention of various researchers. The OTTER (Wang, Rock, & Lees, 1995) and AMADEUS (Lane et al., 1997) projects were among the first to tackle this research area. Antonelli (2014) provided a good theoretical background for underwater manipulators from the modelling and control point of view. More recent progress has been achieved within the TRIDENT FP7 project (Simetti, Casalino, Torelli, Sperindé, & Turetta, 2014) where an electric robot arm manufactured by Graal Tech mounted on an AUV has been used for autonomous detection and retrieval of an object from the sea floor (Ribas et al., 2015). However, for work-class ROV intervention work, such electric manipulators are not designed or available with sufficient power as specified by ocean engineering contractor requirements, e.g. the manipulator-operated torque tool, which uses the wrist rotate function of the manipulator to generate the required torque is used to operate ISO 13628 Class 1 (67 N m) and 2 (271 N m) (ISO 13628-8:2002, 2002) interfaces without the need of a hydraulically operated torque tool (Christ & Wernli, 2014). Actuation forces of subsea hydraulic and electric manipulators are presented in Table 2, and the weight of typical ROV operated tools used in the offshore oil and gas industry are summarized in Table 1. Analysis of these two tables leads to the conclusion that the majority of electrical manipulators would struggle even to lift, let alone intervene with most of the tools. One of the few research groups that have been working with a commercial underwater manipulator (Schilling Orion 7P) is DFKI-Lab Bremen where automated plugging of a deep-sea connector in a wet laboratory testbed has been conducted within the CManipulator project (Hildebrandt, Kerdels, Albiez, & Kirchner, 2009). As outlined, the majority of academic research experiments in the field of autonomous underwater manipulation have been carried out on electrical robotic arms which are either prototypes or recently commercialized. Additionally, all those advanced subsea autonomous manipulation solutions found in literature (Cieslak, Ridao, & Giergiel, 2015; Evans, Redmond, Plakas, Hamilton, & Lane, 2003; Marani & Yuh, 2014) are related to intervention AUVs, which are not industry standard but rather a concept in development and are also considerably power constrained. Not all subsea operations can be performed with electric arms which is why these prototype manipulators are not ready for adoption in offshore industry. There are sound reasons why all work-class ROVs use hydraulic manipulators (depth rating, very high carrying capacity and torque, straightforward field maintenance, etc.). Despite the significant advances achieved by the academic community over the years, the autonomous approach has not been adopted by the commercial ocean engineering sector which still employs traditional telemanipulation approaches with human pilot in the loop for work-class ROVs. Since commercial work-class ROVs are equipped as standard with hydraulic manipulators, which are considerably underdeveloped in the sense of autonomy in comparison with stationary industrial robot arms used in factories, our challenge and goal is to develop advanced control systems that can be employed on these robotic arms with little to no hardware modification. This paper presents investigations, development and adaptation of industrial robot arm (visual) servo control approaches used for typical industrial manufacturing applications and the transfer of these techniques to challenging underwater robotics tasks. For the first time, a solution that works with standard commercial systems already employed in the industry and the global fleet of work-class ROVs is presented. The novelty and contribution of this paper is as a first in the development and implementation of the approach of visual servo control in the subsea manipulator field for the existing marine industry standard commercial work-class ROVs. Our system is able to replicate what an ROV pilot does by the traditional

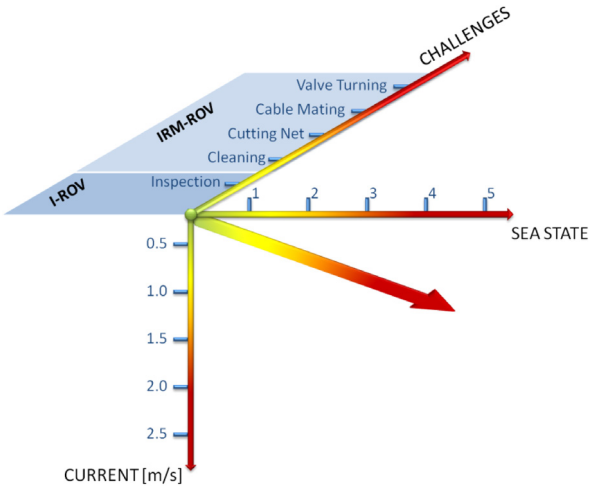


Fig. 1. Difficulty matrix classification with increasing current strength, sea state and challenges of robotic target applications.

Table 1
Actuation force comparison between hydraulic and electric ROV manipulators.

Manufacturer	Model	Actuation	Lift capacity max nom. (full ext.) [kg]	Wrist torque [N m]	Grip force [kgf]
ISE Ltd.	Magnum 7	Hydraulic	454 (295)	108	205
Schilling	Titan 4	Hydraulic	454 (122)	170	417
KNR Systems Inc.	HYDRA UW3	Hydraulic	300 (121)	350	300
Profound Technology	M1P	Hydraulic	275 (250)	175	652
Schilling	Orion 7P/7R	Hydraulic	250 (68)	205	454
Kraft	Predator	Hydraulic	227 (91)	135	135
Hydro-Lek	40400	Hydraulic	150 (210)	75	/
Forum Perry	TA40	Hydraulic	125 (250)	150	509
Cybernetix	Maestro	Hydraulic	100 (96)	190	150
Eca Hytec	Arm 7E	Electric	40 (40)	25	80
Eca Hytec	Arm 7E Mini	Electric	25 (25)	25	50
Eca Hytec	Arm 5E	Electric	25 (25)	25	60
Eca Hytec	Arm 5E Micro	Electric	10 (10)	10	50
Graal Tech	UMA	Electric	10 (/)	/	/
Ocean Innovation System	BE5-500	Electric	/ (16)	1.6	100
Ansaldo	MARIS 70800	Electric	8 (/)	/	20.4

means of teleoperation and it does so faster than the pilot and totally autonomously. Instead of replacing the commercial manipulator systems which are hydro-mechanically well designed, our approach makes them semi-autonomous by applying modern control practices and testing them in challenging subsea robotics tasks.

The remainder of the paper is organized as follows: Section 2 covers the analysis of underwater manipulation scenarios to be addressed and a brief description of recent developments in this sector. Section 3 presents the developed software and the latest algorithms integrated into it. Section 4 describes field experiments and illustrates the efficiency of the proposed approach. Finally, Section 5 concludes and describes future development plans.

2. Sub-sea manipulation task analysis

This section presents analysis of scenarios for implementation of servo control approaches including vision-based control strategies for underwater robot manipulation on ROVs equipped with manipulators. An example of such ROV is Holland I—an Irish Marine Institute owned work-class ROV (see Fig. 2) which is equipped with two seven function Schilling Orion 7P manipulators and primarily used for scientific missions. One of the typical tasks in marine science sector is collecting a sample from the seabed and putting it in a sample container which is fixed on the vehicle or held by the other manipulator. Fig. 3 illustrates the worksite scene of such operation where another scientific ROV, an Ocean Exploration Trust owned ROV called Hercules, is performing sediment core sampling. The scenarios being investigated are outlined below and start with relatively straight forward tasks advancing to more challenging applications.

Scenario 1: ROV is equipped with a five function manipulator and a seven function manipulator both of which are assumed to have angular position sensors such as resolvers in each joint. The task that is addressed in this scenario is placing a grasped sample in a sample container using a seven function manipulator. The sample container is either fixed on the vehicle or held by the five function manipulator. The solution for this scenario is covered in Sivčev, Coleman, Adley, Dooly, Omerdić, and Toal (2015) where an algorithm was developed using standard forward and inverse kinematics techniques common in industrial robotics. Based on the information provided by joint position sensors, along with the known relative pose between the sample container and the robot manipulator base as well as the relative pose between two manipulator bases both of which can be measured or obtained from the ROV geometry model, this algorithm provides the end effector trajectory, in either Cartesian or joint space, which enables the task execution.

Scenario 2: This scenario is identical to the first scenario except that the five function manipulator is assumed not to be as advanced, i.e. it does not have angular position sensors integrated into each joint as is not uncommon for subsea manipulators. One way to compensate for this shortfall is by adding a vision system in the control loop in the form of a camera system mounted either on the robot manipulator or the ROV. Thus, by the aid of visual servoing techniques, it is possible to develop algorithms which generate the desired end effector motion.

Fig. 2. ROV Holland I with two Schilling Orion 7P manipulator.

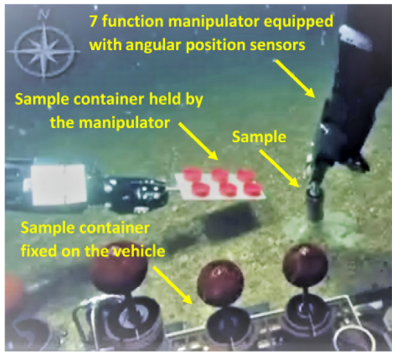


Fig. 3. Sediment core sampling performed by ROV Hercules.

Scenario 3: Unlike the preceding scenarios which deal only with the automation of the placing task stage, this scenario also takes the target object acquisition/grasping into consideration. Moreover, it deals with the interaction of the seven function arm with a target object independent of the ROV base platform. Both static and non-static targets are addressed. Further complication is introduced if the ROV cannot be parked on the seabed during the task execution but has to hover.

Table 2
Summary of typical ROV operated tools.

Tool type	Manufacturer	Weight in air [kg]	Weight in water [kg]	Tool type	Manufacturer	Weight in air [kg]	Weight in water [kg]	Tool type	Manufacturer	Weight in air [kg]	Weight in water [kg]
Torque Tool Class 1–2	FET	26	18	Hub Cleaning Tool	J2 Subsea	21.8	/	Soft Rope Cutter	FET	9.7	/
Torque Tool Class 1–2	J2 Subsea	30.8	25.9	Hub Cleaning Tool	Fugro	46.9	27.5	Soft Rope Cutter	Fugro	10	7
Torque Tool Class 1–4	Jupiter Subsea	48.9	37.1	Hub Cleaning Tool	IKM	58	35	Soft Rope Cutter	J2 Subsea	10	7
Torque Tool Class 1–4	Oceaneering	39	30	Hub Cleaning Tool	Oceaneering	60	40	Soft Rope Cutter	J2 Subsea	15	10
Torque Tool Class 1–4	Oceaneering	43	32	Cleaning Brush Tool	J2 Subsea	7	7	Soft Rope Cutter	Fugro	15	10.5
Torque Tool Class 1–4	Fugro	47	38	Cleaning Brush Tool	FET	10	7	Soft Rope Cutter	Fugro	34.5	23.5
Torque Tool Class 1–4	FET	45	35	Cleaning Brush Tool	ROVQUIP	11.3	7.9	Soft Rope Cutter	Webtool	34.5	23.5
Torque Tool Class 1–4	Oceaneering	64	43	Cleaning Brush Tool	Fugro	18	18	Wire Rope Cutter	Fugro	19	16
Torque Tool Class 5	FET	54	41	Cleaning Brush Tool	Oceaneering	25	21	Wire Rope Cutter	ROVQUIP	21	/
Torque Tool Class 5	Jupiter Subsea	85.5	68.4	Cleaning Brush Tool	IKM	43	31	Wire Rope Cutter	FET	22	/
Torque Tool Class 5	Fugro	90	69	Gasket Removal Tool	J2 Subsea	15	/	Wire Rope Cutter	J2 Subsea	22	19
Torque Tool Class 6	Fugro	107	111	Gasket Removal Tool	ROVQUIP	15.3	10.4	Wire Rope Cutter	FET	43	/
Torque Tool Class 7	Fugro	123.1	82.6	Gasket Removal Tool	Fugro	20.8	15.6	Wire Rope Cutter	J2 Subsea	45	/
Gear Adapter Class 4 to 5	Oceaneering	36	28	Gasket Removal Tool	Fugro	27.8	19.2	Wire Rope Cutter	Fugro	45	32
Gear Adapter Class 4 to 6	Oceaneering	63	46	Diamond Wire Saw	IKM	42	26.9	Wire Rope Cutter	Fugro	120	105
Gear Adapter Class 4 to 7	Oceaneering	71	54	Diamond Wire Saw	Oceaneering	71	50	Wire Rope Cutter	FET	125	/
NORM Inspection Tool	Oceaneering	13.3	/	Diamond Wire Saw	IKM	155	99	Subsea Drill	IKM	140	110
Linear Override Tool	Fugro	41	32	Diamond Wire Saw	IKM	239	159	PH Probe	Oceaneering	30	17
Linear Override Tool	Fugro	49	38	Diamond Wire Saw	Mirage Subsea	413	235	PH Probe	IKM	15	15
Linear Override Tool	i-Tech[7]	55.4	44.5	Diamond Wire Saw	Mirage Subsea	566	269				
Linear Override Tool	Oceaneering	81.6	77.1								

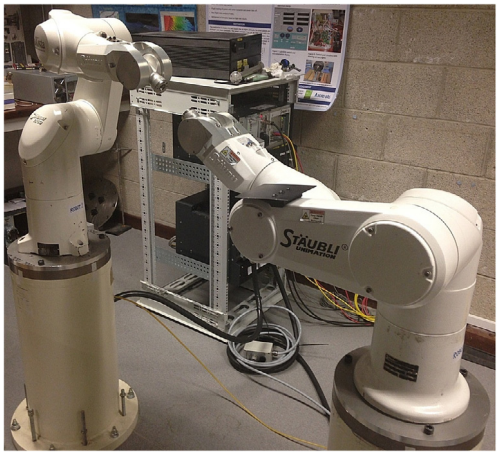


Fig. 4. Two Staubli TX60 industrial robot arms.

Fig. 5. Schilling Titan 2—hydraulic seven function subsea manipulator mounted on a lab test bench.

The robot arm hardware employed in the experimental work to date includes: two 6 axis servo controlled all electric Staubli TX60 industrial robot arms (see Fig. 4), a Schilling Titan 2 hydraulic seven function underwater manipulator (see Fig. 5), a Point Grey Bumblebee 2 stereo vision system (see Fig. 6), a Point Grey Blackfly camera (see Fig. 7) and the robot control software and machine vision software developed by the authors using Matlab Robotics Toolbox, LabVIEW and OpenCV. The developed software can be employed for real robot manipulator motion control as well as in simulations using mathematical models with virtual reality animation designed with Virtual Reality Modelling Language (VRML). The developed algorithms have been tested in simulations and on experimental setup both in dry laboratory controlled conditions as well as underwater in real world conditions.

3. Position Based Visual Servoing (PBVS) algorithm

This section describes the designed algorithms that can cover scenarios 2 and 3 described in the previous section. The algorithms designed are based on the robot arm kinematic model and are intended for testing in experimental setup as well as in real world application implementation. The developed software has been initially tested using two Staubli TX60 industrial robot arms and a Point Grey Bumblebee 2 camera mounted on the wrist of one of the robot arms. Having proved to be satisfactory, the software was modified for use on a Schilling Titan 2 subsea manipulator with a Point Grey Blackfly camera

Fig. 6. Point Grey Bumblebee 2—stereo vision system.

Fig. 7. Point Grey BlackFly camera with Kowa lens and a suitable subsea housing manufactured by Sexton.

mounted on the wrist. In this configuration it was tested both in dry laboratory experiments as well as subsea with the manipulator installed on a commercial ROV Holland I. The results of both experimental setups are presented in this paper. The algorithms encapsulated in the software are purely kinematical, in the sense that they provide kinematic parameters as output signals which is effectively the input reference for the existing manipulator's hydraulic servo control unit. This means that the manipulator is treated as an ideal positioning device as we believe that the off the shelf manipulators have sufficient capabilities to utilize this approach. The forward and inverse kinematics modelling solutions developed for Staubli TX60 robot arm and Schilling Titan 2 manipulator can be found in Sivčev et al. (2015). The developed software consists of a visual based motion control algorithm which can be classified as Position Based Visual Servoing (PBVS). Underwater camera imaging often suffers from various problems such as limited range visibility, low contrast, blurring, etc. (Schettini & Corchs, 2010). Development of a vision system based on the fusion of camera and sonar imaging, that is intended to deal with these issues, is planned. We considered implementing other algorithms such as Image Based Visual Servoing (IBVS) and hybrid methods (Deng, 2004). However, since forward looking sonars provide position information (distance and angle in a polar coordinate system), a PBVS algorithm is a more suitable choice for a camera imaging part. The scheme of the developed PBVS algorithm is shown in Fig. 8. This algorithm can be separated in two main components which are pose estimation and motion control.

3.1. Pose estimation

In order to simplify the object (target) detection and pose estimation, it was decided to use fiducial markers (Fig. 9) which are a well-established pose estimation tool. Using a calibrated camera (Zhang, 2000) and having the information of the exact geometry of the fiducial marker, it is possible to estimate its pose relative to the camera by means of a planar homography based algorithm (Agarwal, Jawahar, & Narayanan, 2005). The accuracy of the pose estimation depends on how well the camera is calibrated and how reliable the fiducial marker model is replicated. The machine vision part of the software performing the fiducial marker pose estimation has been developed in C++ in the form of DLL libraries which can be used within other software. This algorithm continuously takes images captured by the camera, removes distortion using the intrinsic parameters acquired from

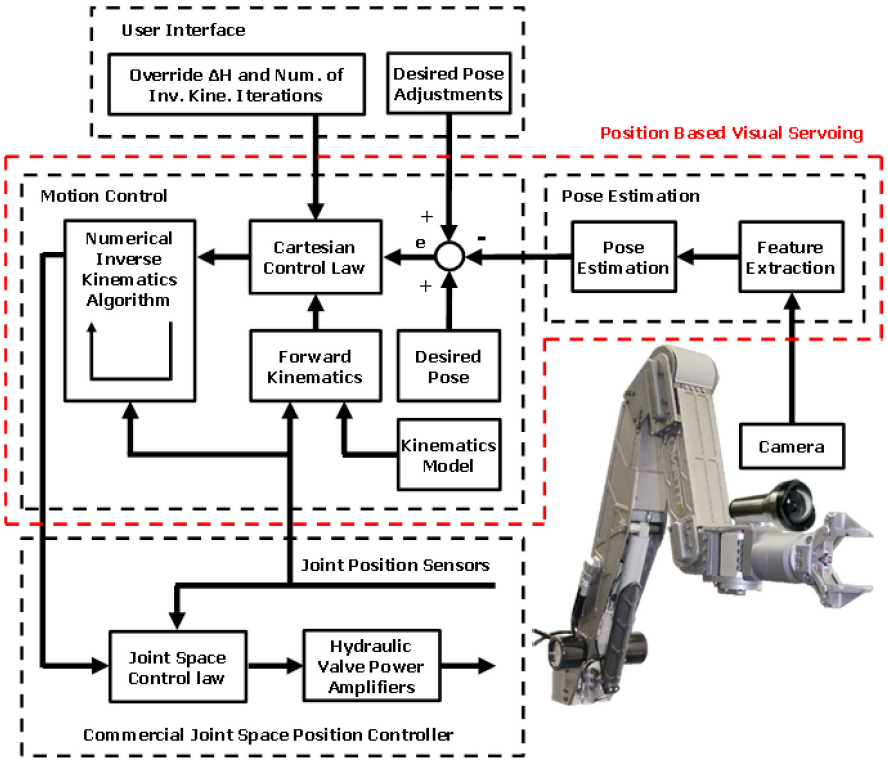


Fig. 8. Position Based Visual Servoing algorithm scheme.

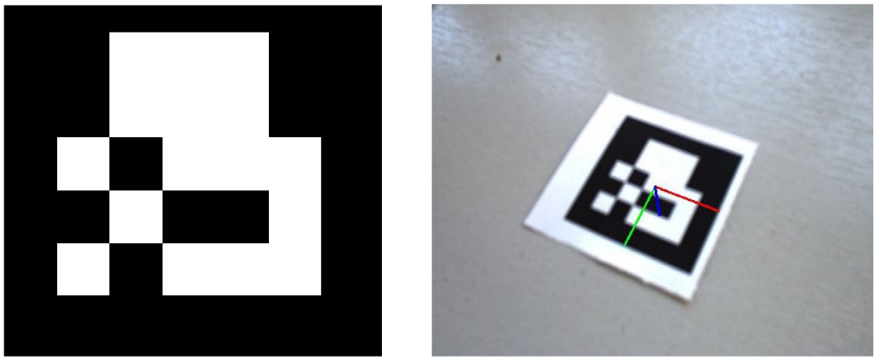


Fig. 9. Fiducial marker.

the camera calibration process, detects the fiducial marker (Garrido-Jurado, Muñoz-Salinas, Madrid-Cuevas, & Marín-Jiménez, 2014) on the undistorted images and based on the extrinsic parameters, also acquired from the camera calibration, provides the estimated pose of the fiducial marker in the camera reference frame (H_M^{CAM}) as an output (Fig. 10). This approach assumes that the target is equipped with an object of known geometry such as a fiducial marker. A more advanced real-time 3D reconstruction method that performs pose estimation and dense surface reconstruction without relying on specific features is currently under development (Rossi et al., 2015). A dense model obtained in this way can be used to identify a target (either manually or automatically from e.g. a CAD model) and compute its position relative to the camera without the requirement for a fiducial marker.

3.2. Motion control

By fixing a fiducial marker rigidly in the vicinity of the target, so that the relative pose between target and marker (H_T^M) is known, determining the pose of the target in the camera reference frame (H_T^{CAM}) becomes straightforward and is given by:

$$H_T^{CAM} = H_M^{CAM} \cdot H_T^M \tag{1}$$

Additionally, assuming that the relative pose between the tool central point (TCP) and the camera (H_{TCP}^{CAM}) is known, it is straightforward to calculate the relative pose between the target and the TCP (H_T^{TCP}) from the expression:

$$H_{TCP}^{CAM} \cdot H_T^{TCP} = H_T^{CAM} \tag{2}$$

as:

$$H_T^{TCP} = (H_{TCP}^{CAM})^{-1} \cdot H_T^{CAM} = (H_{TCP}^{CAM})^{-1} \cdot H_M^{CAM} \cdot H_T^M \tag{3}$$

Defining the H_T^{TCP} as an operational (Cartesian) space control variable and using it to control the motion of the manipulator might seem as an intuitive approach as it is obvious that its convergence to the identity matrix would lead to the TCP reaching the target.

However, since the estimation of the H_T^{TCP} is conducted indirectly, any errors introduced by the inexact information of the target to the

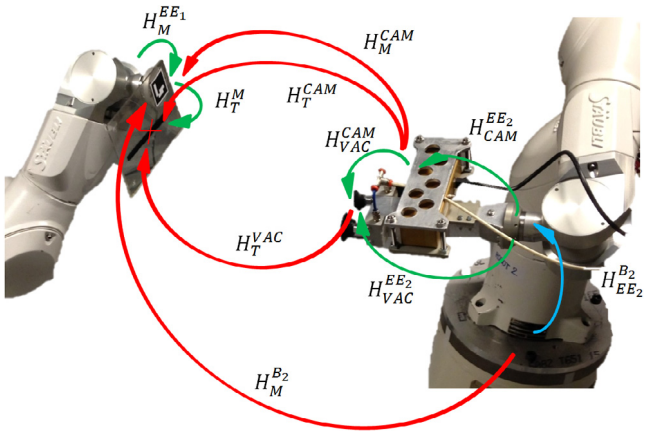


Fig. 10. Relevant homogeneous transformations for pose estimation.

marker (H_T^M), TCP to the camera (H_{TCP}^{CAM}) and camera to the end-effector ($H_{CAM}^{EE_2}$) homogeneous transformations, will introduce an error in the H_T^{TCP} which will reduce the effectiveness of the visual servoing algorithm. Having reliable relevant CAD models (target object with the fiducial marker, gripper, camera, mounts, etc.) might provide sufficiently accurate homogeneous transformations. If these are unavailable, referring to additional measurement techniques might be necessary. The problematic homogeneous transformations to determine accurately are $H_{CAM}^{EE_2}$ and H_{TCP}^{CAM} as both of them are referenced to the coordinate frame of the camera sensor (which is inside the camera housing) and this information is often (if not always) unavailable from the camera data sheets. One of the solutions is to refer to so called "Eye-hand" calibration methods such as Tsai's method (Tsai & Lenz, 1989) in order to estimate the $H_{CAM}^{EE_2}$. Having the information of this homogeneous transformation and reliable CAD models it is possible to estimate the relative pose between the TCP and camera as:

$$H_{TCP}^{CAM} = (H_{CAM}^{EE_2})^{-1} \cdot H_{TCP}^{EE_2} \tag{4}$$

On the other hand, a simple but effective solution that can compensate for the potential imperfections caused by inexact knowledge of the relevant homogeneous transformations ($H_{CAM}^{EE_2}$, H_{TCP}^{CAM} and H_T^M) is to resort to a sort of teach by showing method, by manually moving the robot arm towards the desired pose (TCP in grasp position), and recording the value of the homogeneous transformation which is the output of the computer vision pose estimation algorithm. This value can thus be assigned to the value of homogeneous transformation referred to as the desired pose of the marker in camera reference frame ($H_{des\,M}^{CAM}$). Therefore, the operational space control variable can be defined as the "difference" between the actual and the desired pose of marker relative to the camera:

$$H_{error} = H_M^{CAM} \cdot (H_{des\,M}^{CAM})^{-1} \tag{5}$$

Controlling this error variable so that it converges to the identity matrix would lead to the TCP reaching the target. Having chosen this approach, the next step for designing an appropriate controller is to determine the desired end-effector pose in the base frame ($H_{des\,EE_2}^{B_2}$) that corresponds to the $H_{des\,M}^{CAM}$. It might be compelling at this point to record this value as well during the described "teach by showing" method. However, this would work only if the target (marker) is stationary and its pose relative to the robot base is unchanging (fixed on the ROV). In a more general case, this is unknown and variable as the target (marker) can be anywhere in the workspace of the manipulator. On the other hand, the value of the homogeneous matrix $H_{des\,M}^{CAM}$ is constant independent of where the target in the workspace of the robot is as we assume that there is only one way to grasp an object.

In order to find the corresponding $H_{des\,EE_2}^{B_2}$ it is necessary to determine what relative motion ($H\Delta$) is the end-effector required to perform so that the actual pose of the marker relative to the camera becomes identical to the desired value. Another way of describing $H\Delta$ is the relative pose between the end-effector in the desired pose and the initial pose expressed in the reference frame of the end-effector in the initial pose. This relationship can be determined from the observed expression:

$$H_{CAM}^{EE_2} \cdot H_M^{CAM} = H\Delta \cdot H_{CAM}^{EE_2} \cdot H_{des\,M}^{CAM} \tag{6}$$

where $H_{CAM}^{EE_2}$ is constant and known from the CAD model, H_M^{CAM} is acquired from the pose estimation algorithm, and $H_{des\,M}^{CAM}$ is the desired value described earlier. Using simple matrix operations on this expression leads to:

$$H\Delta = H_{CAM}^{EE_2} \cdot H_M^{CAM} \cdot (H_{des\,M}^{CAM})^{-1} \cdot (H_{CAM}^{EE_2})^{-1} \tag{7}$$

where $H\Delta$ is in a homogeneous transformation matrix form. Finally, the desired end-effector pose in the base frame can be calculated by:

$$H_{des\,EE_2}^{B_2} = H_{EE_2}^{B_2} \cdot H\Delta \tag{8}$$

where $H_{EE_2}^{B_2}$ is the pose of the end-effector in the robot base frame calculated using forward kinematics with the values of the joint positions corresponding to the moment of capturing a camera image for the pose estimation algorithm.

This value represents the reference for the motion control in Cartesian space as it would be traditionally defined. At this point the motion control part of the visual servoing algorithm will be described.

It is important to emphasize that the motion control component of the PBVS algorithm deals only with kinematics, i.e. generating reference motion parameters which are to be forwarded to the existing low level joint space positioning motion controller. In order to find the joint variables corresponding to the desired pose of end-effector in the robot base frame ($H_{des\,EE_2}^{B_2}$), it is necessary to solve the inverse kinematics problem. This is done by means of the closed loop second-order inverse kinematics algorithm with pseudo inverse Jacobian proposed in Sivčev et al. (2015), where the value of the desired end-effector pose relative to the robot base frame ($H_{des\,EE_2}^{B_2}$) is assigned to $Xd(t)$. A numerical method for solving inverse kinematics is used as the Schilling Titan 2 manipulator does not have a spherical wrist and therefore a closed form analytical solution does not exist. Another advantage of the numerical approach is that with slight modifications it can be utilized for a manipulator with any physical configuration with any number of joints whereas the analytical solution is limited to six joints with a configuration which possess a spherical wrist.

Assuming that the camera is well calibrated and that the errors introduced with other modelling imperfections are negligible, an open loop control scheme often referred in literature as the "Look then move" method where the pose is estimated just once (Corke, 1996) can be applied. In that case, the inverse kinematics numerical algorithm would be designed so that it would run in as many iterative loops as required until the solution converges to the predefined error threshold. Apart from neglecting the possibility of errors introduced in the modelling which will lead to inaccuracy, the disadvantage of this method is that the time required for the execution of the sole inverse kinematics algorithm is variable and unknown in advance and could take too long. This method might prove to be sufficiently accurate for the stationary target, and only assuming that the modelling is very good, but it is likely to fail for the target in motion due to the variable relative pose between the camera and the target. Therefore, closing the loop is necessary, both for the reason of modelling errors which are inevitable and for the future purpose of addressing targets in motion. This realization led us towards adopting the dynamic "look-and move" visual servoing scheme (Sanderson & Weiss, 1980). Closing the loop in this manner, the

proposed PBVS algorithm consists of two loops, the outer loop which is in charge of pose estimation and reference setting and the inner loop which is in charge of inverse kinematics.

3.3. Discussion of the PBVS algorithm

This subsection presents discussion on certain parameters that affect the efficiency of the developed PBVS controller.

As highlighted in the previous section, the inverse kinematics solution is computed by means of a numerical integration method in discrete time (Sivčev et al., 2015). The value of the integration interval affects the efficiency of the algorithm. One way to improve the algorithm performance is to have an integration interval that is variable and adaptively modified. An alternative method that provides the same effect, is to pre-set the integration interval to a certain constant value and have a discrete inverse kinematics algorithm with a variable number of iterations which is adaptively modified by the Cartesian controller based on some strategy. Greater values for this parameter results in the output of the inverse kinematics becoming more accurate, as it gradually converges to the given desired value. However, processing each inverse kinematics loop takes certain amount of time so the number of loops determines the frequency of the outer loop. In other words, it determines how often the pose of the target is re-estimated. In the case where target is stationary, this is not too relevant, and this parameter does not have an upper boundary. It can be set to any value that leads to the minimal total time required for the task to be executed. However, as soon as the target in motion comes into play, this parameter gets an upper boundary. The reason for this is that increasing the number of iterations over a certain value will inevitably lead to the execution of the inverse kinematics algorithm taking too much time and the object will leave the field of view of the camera before the next image is taken for the pose estimation. This causes the opening of visual feedback because of the lack of visual measurements and eventually to servoing failure. Reducing the number of iterations proved to be a good way to keep the object in the field of view of the camera during the task execution. However, having insufficient number of iterations will, due to the higher mismatch of the inverse kinematics output and the given desired value, increase the total number of outer iterations which will increase the total time required for the task to be executed. Therefore, it is clear that in order to minimize the total time required for the execution of the task while keeping stability in mind, it is essential to find an appropriate value for this parameter or even an algorithm that adaptively changes the value based on some law and relevant inputs (distance to the target, target velocity, etc.).

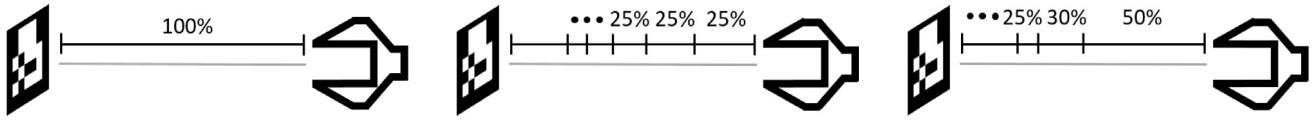
We have explained one method to reduce the risk of the target object leaving the camera field of view. As the PBVS acts directly on operational space variables, with appropriate path planning algorithm, the camera's trajectory can be directly controlled in the Cartesian space and this problem can be prevented. This can be done by additional operations on the ΔH homogeneous transformation, which represents the estimated relative motion that the end-effector needs to perform in so that the gripper reaches the target. By adaptive modification of this value, it is possible to control how the gripper approaches the target, rather than just making it reach the target. Therefore, instead of having $H\Delta$ as a fixed value homogeneous transformation matrix obtained based on the pose estimation result, it is possible to modify it to the desired needs. A simple method that can reduce the risk of the target leaving the field of view of the camera, is to form a parameter that represents the percentage of the "path" along the straight line in Cartesian space between the initial and the desired pose (Fig. 11). Using this parameter, the value of the desired relative pose $H\Delta$ and interpolation techniques it is possible to find the corresponding relative pose. Position coordinates can be computed by means of standard linear interpolation, and the orientation parameters by the spherical linear interpolation method (Dam, Koch, & Lillholm, 1998) for Quaternions. By adopting this method, the inverse kinematics algorithm will now have as a desired value an intermediate point in the Cartesian space between the initial and the final pose which will eventually lead to the manipulator reaching the target. This will be done in steps covering a certain amount of the path towards the target rather than at once. The value of this parameter determines how small steps are to be taken while approaching the target. This can be also used as a method to control the approach velocity. This will inevitably increase the total time required for the task to be executed, but it is useful as it plays an important role in preventing the target object leaving the field of view of the camera. Another benefit is that it introduces a safety factor. If the target is in motion, there is certainly a possibility of collision. If the target motion is such that it is approaching the gripper, having big intermediate steps in the visual servoing loop could lead to collision. On the other hand, if the steps are small enough, approaching of the target object can be "dealt" with and appropriate motion control utilized that will lead to the successful task execution, without collision. Although it cannot ensure that the target object stays in the field of view of the camera, the experiments with the stationary target have shown that this method can reduce the risk of collision. However, an increase in target object motion might entail the need to implement robust path planning methods for visual servoing, some of which can be found in Baumann, Léonard, Croft, and Little (2010), Chesi, Hashimoto, Prattichizzo, and Vicino (2004), Kazemi, Gupta, and Mehrandezh (2009) and Thuilot, Martinet, Cordesses, and Gallice (2002).

The two described parameters (number of iterations of the inverse kinematics algorithm and the parameter that represents the percentage of the relative desired pose) affect the efficiency and the speed of task execution. It is important to emphasize that these parameters are mutually dependent. Therefore, in order to maximize performance of the visual servoing algorithm, it is necessary to address their effect in combination rather than independently. Additionally, other relevant information can and should be taken into account when addressing the described parameters. One of them is the velocity of the target in motion which can be estimated, analysed and taken into consideration. Depending on the target velocity, the relevant parameters can be adaptively modified. If the target is slow, a bigger portion of ΔH in combination with a larger number of iterations should be suitable. On the other hand, if the target is fast, smaller steps should be taken in combination with less inverse kinematics iterations. Another important factor is the distance between the camera and the target. The closer the object is to the camera, the larger it is on the image plane and can therefore more easily leave the image plane, especially if it is moving relatively fast. Therefore, the closer the gripper is to the target, the number of iterations should be reduced and the portion of the motion increased, to reach the target promptly in this terminal phase.

4. Experiments

4.1. Description

The addressed scenario for the validation of the developed software including the proposed PBVS algorithm consists of Schilling Titan 2 manipulator equipped with a Point Grey BlackFly camera mounted on the wrist of the manipulator (primary manipulator for visual servoing), Schilling Orion 7P manipulator (supporting manipulator for cooperative manipulation tests) and various test panels equipped with fiducial markers and standard T-bar handles simulating target devices for different intervention tasks typical for ROV manipulators. A T-bar handle is a standardized mechanical interface between manipulator jaws and subsea tooling equipment and infrastructure (ISO 13628-8:2002, 2002). The Titan 2 miniature master arm—Master Controller Unit (MCU) was replaced with the developed software running on a dedicated topside PC which is communicating over an available serial communication channel with the Schilling Titan 2 Slave Controller Unit (SCU), which is located on the ROV near the manipulator base. This software is an application presented in Sivčev et al. (2015) which was further



(a) "One shot". (b) Fixed. (c) Variable.

Fig. 11. Concept of target approaching with variable percentage of path.

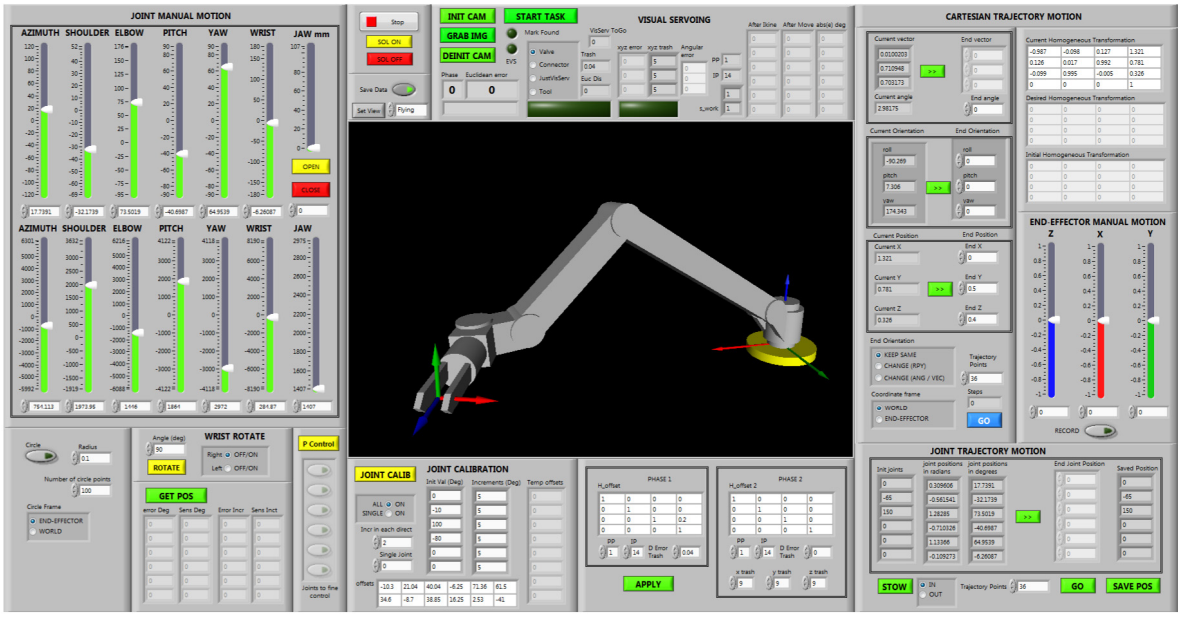


Fig. 12. Developed LabVIEW application front panel of virtual instrument.

developed in order to support testing of the algorithms described in this paper. Fig. 12 presents the user interface of the developed application. The development of the software is largely done in LabVIEW using the Robotics Toolbox (Corke, 2011) and partly in Visual Studio using several open source libraries and SDKs (OpenCV, ArUco, Triclops and FlyCapture). The automated manipulator task experiments which were carried out are:

1. Visual Servoing algorithm validation task—where the software estimates the pose of the fiducial marker and controls the manipulator so that it moves to and stays in the vicinity of the target for a specified amount of time. This task can be carried out with different speed settings and different initial positions of the fiducial marker as long as it is in the workspace of the manipulator. In case the marker is not in the field of view of the camera the manipulator enters a "search" phase performing pan/tilt motion with the end-effector to locate the target.
2. Grabbing the Tool—where the visual servoing algorithm is implemented to enable the manipulator to automatically locate the T-bar handle representing a subsea tool, approach it, grab it and pull it out of the tool holder.
3. Turning the Valve—where the manipulator automatically locates, grabs and rotates the T-bar handle representing a valve handle.
4. Plugging the Connector—where the manipulator automatically plugs the T-bar (which is already held in its jaws) in a hole, simulating the process of plugging a subsea connector.

Additionally, manual and semi-automatic functions integrated into the developed pilot control software which were also tested include:

1. Joint space motion—where the user can manually control robot's joints separately by moving a slider or can set a final position in joint space (six angles plus jaw opening) and generate a smooth trajectory from the initial to the desired joint position. This function also facilitates the auto stow/unstow of the manipulator.
2. Straight line Cartesian space motion—where the user can manually move the end-effector in three directions (x, y, and z) in both end-effector and world coordinate frame or can set the desired point in Cartesian space in world or end-effector coordinate frame and generate a trajectory from the initial to the desired pose forcing the end-effector to move in a straight line. Implementing this function, the final orientation can be kept the same as in the initial pose or it can be specified using either roll-pitch-yaw or angle-axis representation.
3. Circular motion in Cartesian space—where the user can specify a circle radius and chose a relevant coordinate frame (world or end-effector) and generate a trajectory which forces the end-effector of the manipulator to move in a circle, suitable for cleaning operations.

Regardless of what manner the robot motion is generated, in the outputs of the developed software are joint angles which are mapped to the inputs of the Schilling Titan 2 manipulator slave controller unit. The user interface of the developed software also provides real time animation of the manipulator motion.

Another function integrated into the developed software is joint calibration. The necessity to develop this was due to the insufficient absolute accuracy of the integrated Schilling Titan 2 manipulator joint position controller within the slave controller. For a given position command the joints move to the desired position but with a certain error which was determined by reading the joint position values from the

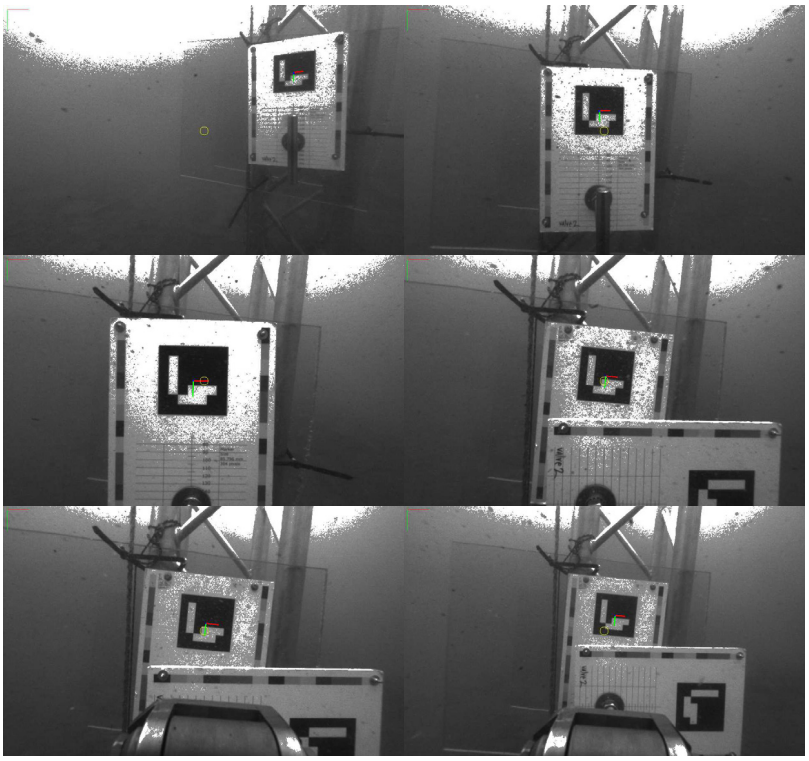


Fig. 13. Images taken by the Point Grey Camera during the execution of the "Turning the Valve" intervention task.

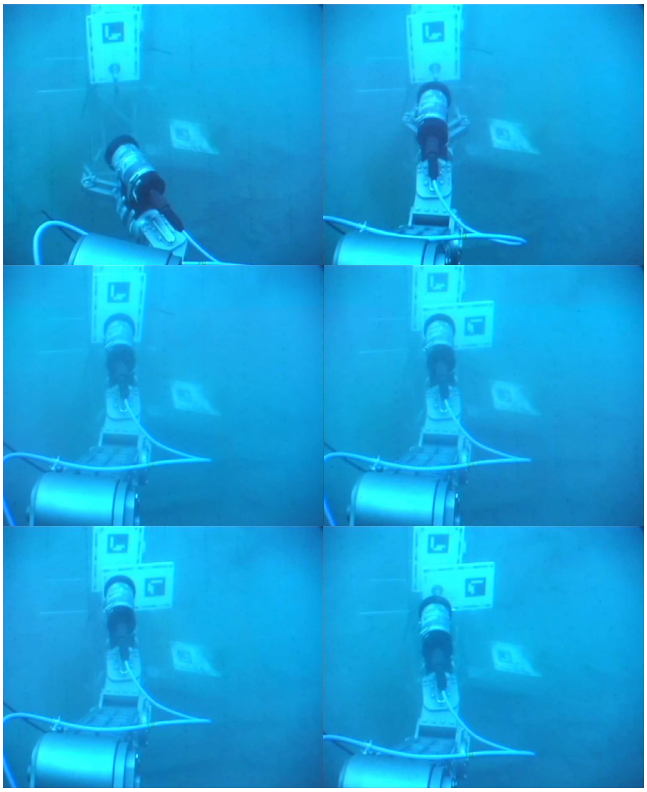


Fig. 14. Images taken by the camera mounted on the ROV during the execution of the "Turning the Valve" intervention task.

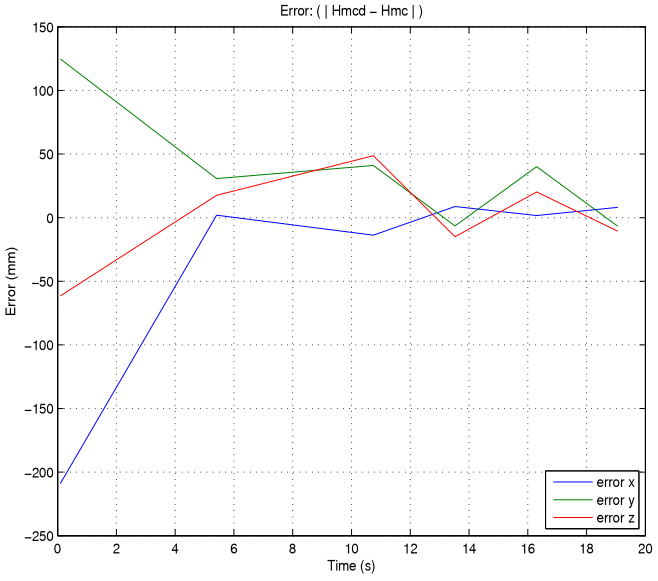


Fig. 15. Position error during the execution of the valve turning intervention task.

resolvers after the motion. This error is significant and for specific joints it reached up to 1.5°. It was determined that the error is different for each joint and that it depends from which direction the joint is moving to the desired position, i.e. hysteresis. Also, the error value is different for the different desired position through the range of motion. However, in the vicinity of the specified desired position the error value turned out to be more or less repeatable. Therefore, by performing certain motions relative to the specified pose of the robot, the joint calibration algorithm estimates the errors between the commanded joint positions and positions to which the manipulator actually moved and based on that error calculates the command offsets equipped to reduce this error.

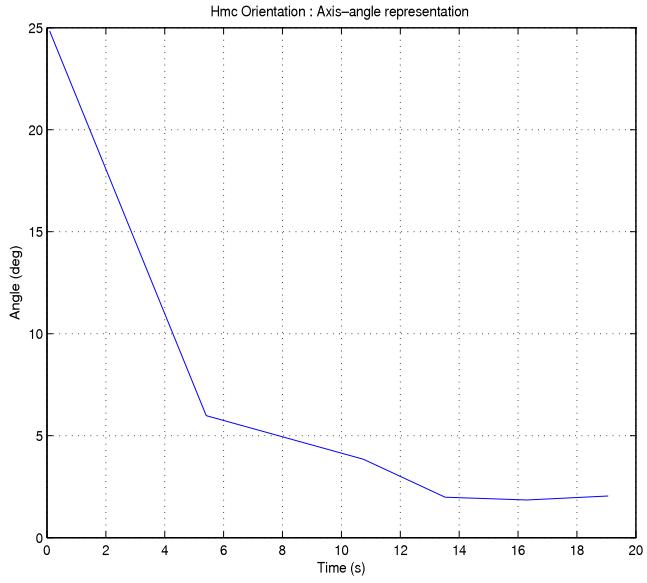


Fig. 16. Orientation error during the execution of the valve turning intervention task.

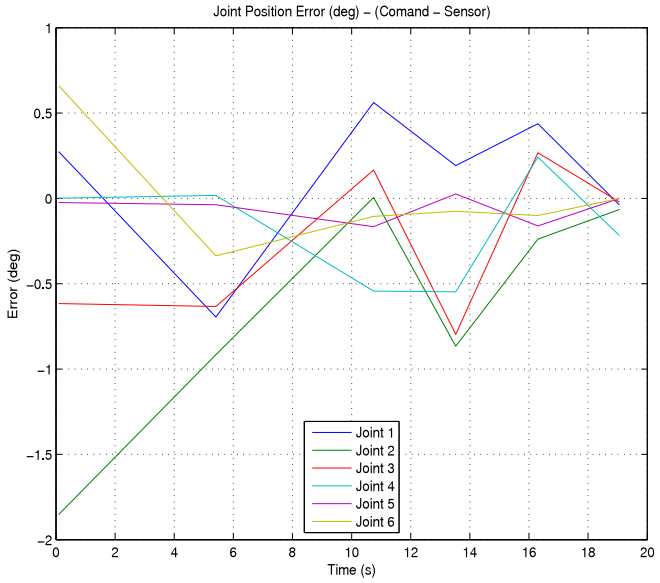


Fig. 18. Joint position error during the execution of the valve turning intervention task.

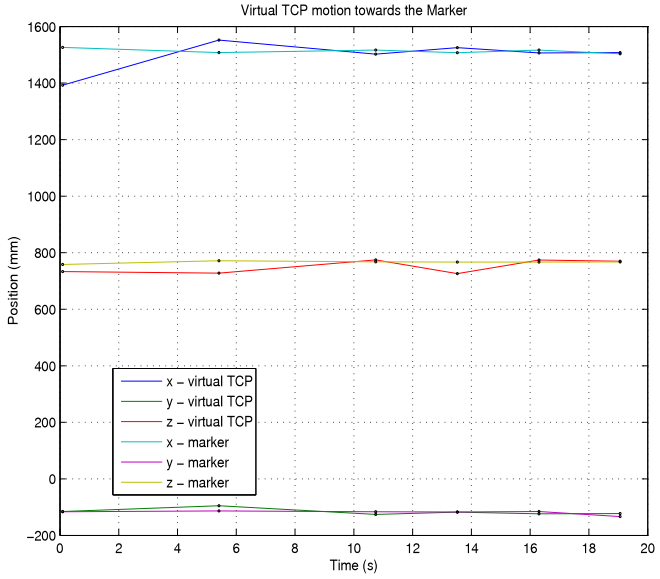


Fig. 17. Convergence of the TCP towards the estimated marker pose during the execution of the valve turning intervention task.

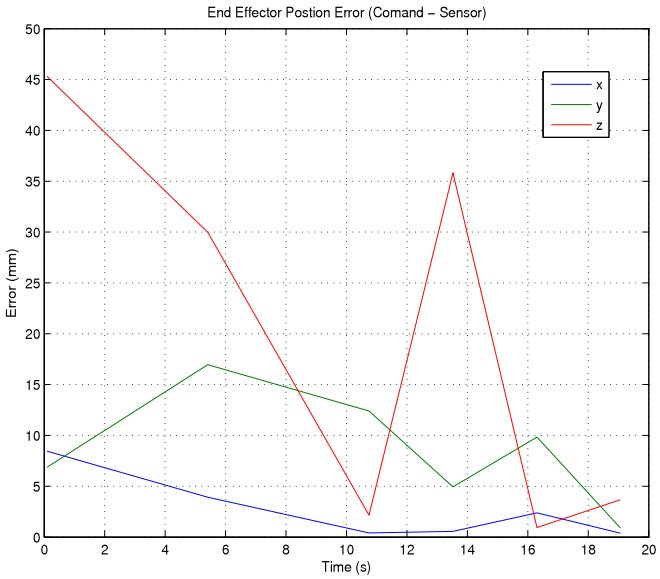


Fig. 19. End-effector position error as a consequence of the joint position error during the execution of the valve turning intervention task.

The control software was initially developed to completely replace the Titan 2 master controller. However, in order to provide the possibility for the pilot to choose either traditional teleoperation mode using the miniature master controller or the automatic and semi-automatic functions proposed in this paper, the overall system (hardware and software) was modified to include a software switch enabling one of the two input devices, with dual operation modes available. The benefit of this is that it eases the installation on the existing manipulator system and facilitates a smooth transition for pilots.

4.2. Field trials

The developed software was validated within two experiments, one of which was conducted within the University of Limerick laboratory in dry conditions and another which was performed in real world underwater environment in a flooded quarry in Portroe (Ireland) where the manipulator was mounted on the Marine Institute's work-class ROV, Holland 1.

All tasks described in the previous sub-section with the stationary target were carried out in dry (laboratory) condition experiments. The same tasks except the "Plugging the Connector" were also performed in the underwater environment. Additionally, the visual servoing algorithm validation task was carried out underwater with the target in motion. All visual servoing tasks were tested with different speed settings as well as with different initial target locations. For all experiments the ROV was "parked" in a fixed position. Marine work-class ROVs often use a basic secondary grabber arm for fixing the ROV to the underwater structure to be worked on. This case is basically identical with the case where the ROV is parked, since the target is in both cases static relative to the base of the manipulator.

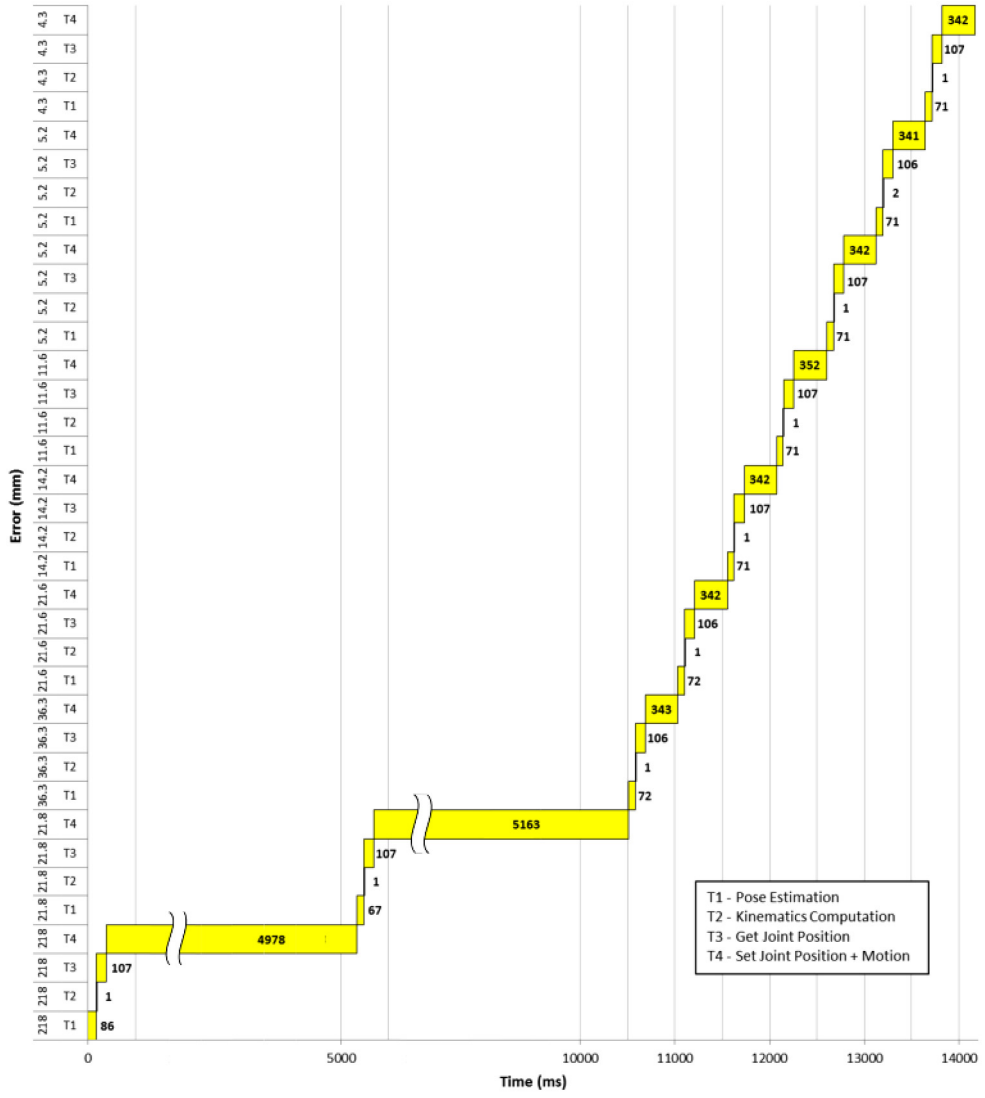


Fig. 20. Partitioned execution time of the PBVS algorithm during the visual servoing task.

This paper presents the detailed results of the "Turning the Valve" intervention task which was carried out in the following phases:

1. Preparation phase—where the Titan 2 manipulator is manually moved so that the target is in its workspace.
2. Coarse approach phase—where PBVS algorithm is utilized and the end-effector moves to place the camera right in front of the fiducial marker with a certain distance offset. In this phase, the homogeneous transformation which represents the reference for the motion controller ($H_{des}{}^{B_2}_{EE_2}$) is additionally modified so to include a certain offset in the Z direction. This effectively forces the manipulator to move a pose such that the gripper is in front of and pointing straight at the target. This phase is active until the position and orientation encapsulated in the error vector are below the specified threshold values. The error vector is formed as the difference between the desired and the actual values of the position and orientation vectors. The threshold values are relaxed as the accuracy is not too relevant in this phase, i.e. the goal of this phase is to acquire the satisfactory initial conditions for the next phase in a timely manner.
3. Fine approach phase—where the manipulator is guided by the PBVS algorithm to approach the target in a pose convenient for gripping. The reference homogeneous transformation matrix ($H_{des}{}^{B_2}_{EE_2}$) is not modified at all in this phase and the thresholds are set to much lower values in order to ensures that the gripper reaches the target with sufficient accuracy.
4. End phase—where the manipulator grabs the T-bar handle by closing the jaws, rotates the wrist through 90°, opens the jaws and finally pulls back by moving in a straight line, respectively. This phase is performed "blindly", without visual feedback.

Fig. 13 shows images taken by the Point Grey camera during the execution of this task. These are the same images which were used for the fiducial marker pose estimation within the PBVS algorithm. Images from the camera mounted on the ROV taken in the same time as the images from Fig. 13 are presented on Fig. 14. The total time required for the task execution was 30 s, where the coarse approach phase was accomplished in 5 s in a single iteration, fine approach phase in 14 s in 4 iterations and the end phase in 11 s. The position and orientation error vectors, during the visual servoing phases are presented in Figs. 15 and 16 respectively. The position error is formed by extracting the translation vector from the H_{error}, and the orientation error by extracting the rotation matrix from the same homogeneous transformation, and converting it into the axis-angle representation (Siciliano, Sciavicco, Villani, & Oriolo, 2009, p. 52). It can be noticed that the position error converges below the threshold value of 15 mm, halting the visual

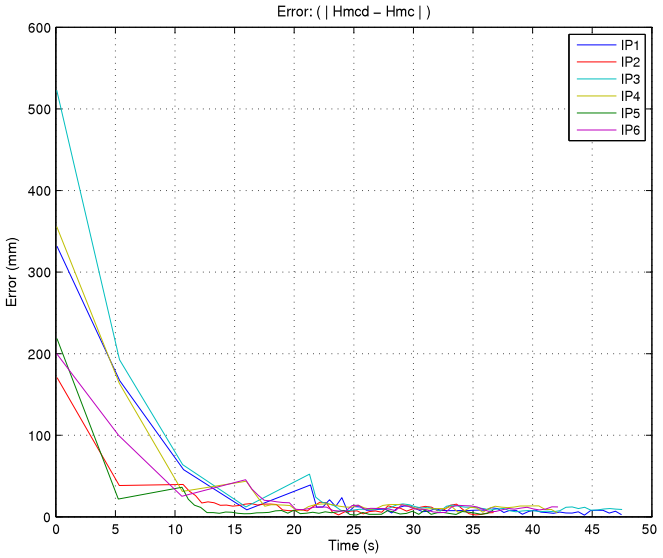


Fig. 21. Norm of position error during the visual servoing task with stationary target. Six experiments with different Initial Positions (IP1,..,IP6).

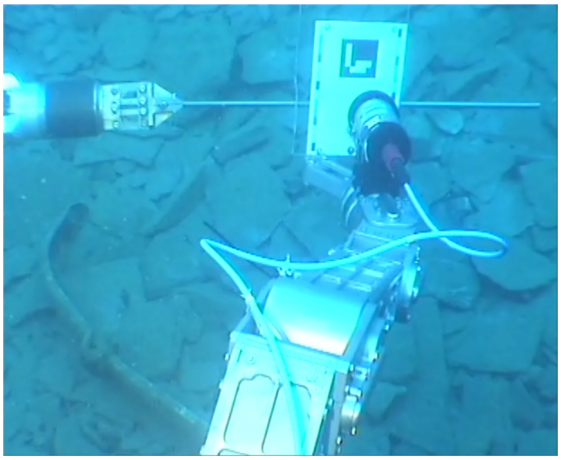


Fig. 22. Image taken by the ROV mounted camera during the visual servoing task with the target in motion, mounted on the Schilling Orion 7P manipulator.

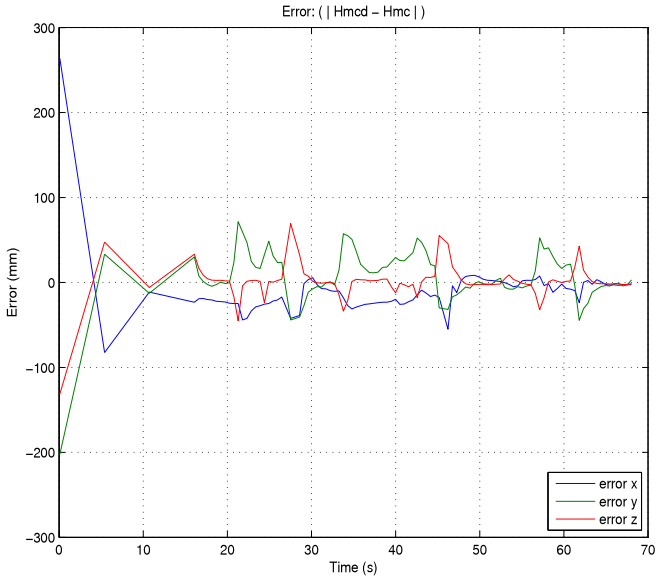


Fig. 23. Position error during the visual servoing task with the target in motion.

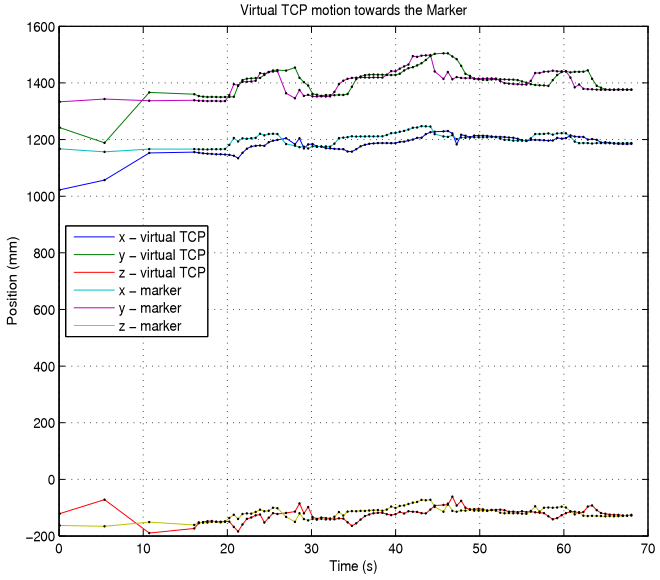


Fig. 24. Convergence of the TCP towards the estimated marker pose during the visual servoing task with the target in motion.

servoing approach phase and triggering the end phase. Fig. 17 represents the estimated marker pose and how the TCP converges towards it. Even though the joint calibration procedure was conducted, certain error was still present between the joint position commands which are the output of the developed algorithm and the joint position sensed by the resolvers (see Fig. 18). As a consequence of this, error in the end-effector position was constructed using forward kinematics (see Fig. 19). Despite the significant influence of the insufficient absolute joint controller accuracy on the Cartesian space error variable, the intervention task was successfully completed. Additionally, it can be noticed that the state feedback is sparse and hence the experiment consists of very few data points, with noticeably low control update rate. This is due to the limitations of the off-the-shelf SCU. Nonetheless, despite the sparse feedback satisfactory performance is achieved. Fig. 20 depicts the execution time of the PBVS algorithm during the visual servoing validation task. Each control loop is split into four distinct segments. The pose estimation part, which takes approximately 72 ms encapsulates image acquisition, image processing, and planar homography computation. The segment that includes the calculation of the homogeneous transformation matrix equations, and the numerical inverse kinematics solution is negligible compared to the other ones. The following segment shows the execution time required to acquire angular joint position readings. The Titan 2 low level motion controller does not support continuous stream of angular sensor position data. Therefore, getting this requires issuing a serial communication request and await a reply, which takes more than 100 ms. The last segment of the control loop includes another pair of serial messages, a command to move and a confirmation that the message is received, as well as the time it takes for the manipulator to perform a desired motion. The duration of this segment is variable and depends on the distance the manipulator's end-effector needs to traverse. The facts that clearly point out the limitations of the commercial underwater manipulator systems are that each loop requires the exchange of four serial messages, and that it is not possible to continuously read position sensor data and broadcast position commands. However, despite these technical shortcomings, our visual control approach works well. The developed PBVS algorithm is

evaluated for multiple initial positions, which can be seen from Fig. 21 which represents the position error for six different experiments.

Additionally, the results of visual servoing experiment with the target in motion are presented. In order to see how well the manipulator can track the target utilizing the developed PBVS algorithm, the panel with the fiducial marker was mounted on the second underwater manipulator (Schilling Orion 7P) which was operated manually by the ROV pilot, moving it up to 100 mm up and down, forth and back with relatively slow speed (see Fig. 22). Experiment was conducted for 70 s and manipulator managed to track the target, although with noticeable delays between the movements. From Fig. 23, it can be seen that every time the target is moved the position error vector increases but eventually converges to zero after approximately 2 s. This can also be noticed from Fig. 24 which presents the estimated fiducial marker pose and how the TCP converges towards it as well as the aforementioned tracking delay.

The intervention task "Turning the Valve" with the stationary target was also successfully conducted using two manipulators where Schilling Orion 7P was fixed holding the target while the Schilling Titan 2 utilized the PBVS algorithm. With this experiment the possibility for implementing simple visual guided cooperative tasks using two subsea manipulators was validated.

5. Conclusion and future work

By adapting algorithms common for industrial robotics, the developed kinematics engine which enables generating reference kinematics parameters for motion both in joint and Cartesian space has been successfully demonstrated and tested. Additionally, we developed visual servoing algorithms for existing commercial subsea hydraulic manipulators and encapsulated them into our software. For the first time, a proven, field-tested solution is presented that works with the global fleet of industry standard commercial work-class ROV systems as a software upgrade rather than a hardware replacement. Encouraging results have been presented based on unique real world underwater experiments, demonstrating the ability of the visual servo control scheme. Viewed from an industrial robotics perspective, the proposed solution may be rather simple as it only considers kinematics, however, it has been proven experimentally to be an entirely suitable approach and represents a very significant advancement in commercial subsea manipulator capabilities. The system presented is effective in replicating, entirely autonomously, what an ROV pilot does by traditional means of teleoperation for the tasks addressing stationary targets, using unmodified industry standard subsea manipulator systems. Using the developed control system, we believe that ROV pilots would, in a supervisory role, be able to execute typical underwater manipulation task with greater ease, faster, and with reduced cognitive load. This could provide significant cost savings for subsea intervention operations and significantly reduce pilot fatigue and associated errors. We believe that we have achieved excellent initial results with very promising applications and potential for uptake in the field of automated IRM of oil and gas, and MRE installations.

Ongoing work is continuing development of the visual servoing algorithms, focused on improvement of the performance of target in motion tracking and addressing intervention on a target/ROV in motion. The plan is to extend the control strategy from the sole manipulator control to the control of a unified ROV-manipulator system.

Acknowledgements

This material is based upon works supported by Science Foundation Ireland (SFI) under the Research Centres Award 2012, SFI Centre for Marine & Renewable Energy Research (12/RC/2302 and 14/SP/2740). The MaREI project is also supported by the following industrial partners: Resolve Marine Group, Shannon Foynes Port Company, Teledyne Blueview, Teledyne Reason and The Commissioners of Irish Lights (12/RC/2303 and 14/SP/2710).

References

Agarwal, A., Jawahar, C., & Narayanan, P. (2005). *A survey of planar homography estimation techniques, Centre for Visual Information Technology, Tech. Rep. IIIT/TR/2005/12*.

Antonelli, G. (2014). *Underwater rocbots* (Vol. 96) (3rd ed.). ISBN: 978-3-319-02876-7, Switzerland: Springer International Publishing. http://dx.doi.org/10.1007/978-3-319-02877-4.

Baumann, M., Léonard, S., Croft, E. A., & Little, J. J. (2010). Path planning for improved visibility using a probabilistic road Map. *IEEE Transactions on Robotics*, (ISSN: 1552-3098) *26*(1), 195–200. http://dx.doi.org/10.1109/TRO.2009.2035745.

Cavnar, B. (2010). *Disaster on the horizon: high stakes, high risks, and the story behind the deepwater well blowout*. Chelsea Green Publishing.

Chang, C. C., Chang, C. Y., & Cheng, Y. T. (2004). Distance measurement technology development at remotely teleoperated robotic manipulator system for underwater constructions. In *Proceedings of the 2004 international symposium on underwater technology (IEEE Cat. No.04EX869)* (pp. 333–338). http://dx.doi.org/10.1109/UT.2004.1405598.

Chesi, G., Hashimoto, K., Prattichizzo, D., & Vicino, A. (2004). Keeping features in the field of view in eye-in-hand visual servoing: a switching approach. *IEEE Transactions on Robotics*, (ISSN: 1552-3098) *20*(5), 908–914. http://dx.doi.org/10.1109/TRO.829456.

Christ, R., & Wernli, R. (2014). *The ROV manual: A user guide for remotely operated vehicles* (2nd ed.). Oxford, UK: Elsevier Science, ISBN 9780080982884, http://store.elsevier.com/The-ROV-Manual/Robert-Christ/isbn-9780080982885/.

Cieslak, P., Ridao, P., & Giergiel, M. (2015). Autonomous underwater panel operation by GIRONA500 UVMS: A practical approach to autonomous underwater manipulation. In *2015 IEEE international conference on robotics and automation, vol. 2*. ISBN: 978-1-4799-6923-4, (pp. 529–536). IEEE. http://dx.doi.org/10.1109/ICRA.2015.7139230.

Coleman, D. F., Ballard, R. D., & Gregory, T. (2003). Marine archaeological exploration of the black sea. In *Oceans 2003 celebrating the past... teaming toward the future, IEEE Cat. No.03CH37492, vol. 3* (pp. 1287–1291). http://dx.doi.org/10.1109/OCEANS.2003.178043.

Corke, P. (1996). *Robotics and mechatronics series: Vol. 2. Visual control of robots: high-performance visual servoing*. Taunton, England: Research Studies Press, ISBN: 0863802079.

Corke, P. (2011). *Springer tracts in advanced robotics: Vol. 73. Robotics, vision and control*. ISBN: 978-3-642-20143-1, Berlin, Heidelberg: Springer.

Dam, E. B., Koch, M., & Lillholm, M. (1998). *Quaternions, interpolation and animation* (Vol. 2). Datalogisk Institut, Københavns Universitet.

Davey, V. S., Forli, O., Raine, G., & Whillock, R. (1999). *On-destructive examination of underwater welded steel structures* (Vol. 1372). ISBN: 978-1-85573-427-2, Cambridge, UK: Woodhead Publishing.

Deng, L. (2004). *Comparison of image-based and position-based robot visual servoing methods and improvements*. University of Waterloo.

Djapic, V., Na, Đ., Ferri, G., Omerdic, E., Dooly, G., & Toal, D. (2013). Novel method for underwater navigation aiding using a companion underwater robot as a guiding platforms. In *2013 MTS/IEEE OCEANS - Bergen* (pp. 1–10). http://dx.doi.org/10.1109/OCEANS-Bergen.2013.6608153.

Evans, J., Redmond, P., Plakas, C., Hamilton, K., & Lane, D. (2003). Autonomous docking for Intervention-AUVs using sonar and video-based real-time 3D pose estimation. In *Oceans 2003. Celebrating the past. teaming toward the future, IEEE Cat. No.03CH37492, vol. 4* (pp. 2201–2210). http://dx.doi.org/10.1109/OCEANS.2003.178243.

Fletcher, B. (2000). *Worldwide undersea MCM vehicle technologies, Tech. Rep.*. DTIC Document.

Garrido-Jurado, S., Muñoz-Salinas, R., Madrid-Cuevas, F., & Marín-Jiménez, M. (2014). Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, *47*(6), 2280–2292. http://dx.doi.org/10.1016/j.patcog.01.005.

Hildebrandt, M., Kerdels, J., Albiez, J., & Kirchner, F. (2009). A multi-layered controller approach for high precision end-effector control of hydraulic underwater manipulator systems. In *OCEANS 2009* (pp. 1–5). (ISSN:0197-7385).

ISO 13628-8:2002, (2002). *Petroleum and natural gas industries —Design and operation of subsea production systems —Part 8: Remotely Operated Vehicle (ROV) interfaces on subsea production systems, Standard*. International Organization for Standardization, Geneva, Switzerland.

Jones, D. O. B. (2009). Using existing industrial remotely operated vehicles for deep-sea science. *Zool. Scr.*, (ISSN: 1463-6409) *38*, 41–47. http://dx.doi.org/10.1111/j.1463-6409.2007.00315.x.

Kazemi, M., Gupta, K., & Mehrandezh, M. (2009). Global path planning for robust visual servoing in complex environments. In *2009 IEEE international conference on robotics and automation* (pp. 326–332). (ISSN:1050-4729) http://dx.doiorg/.10.1109/ROBOT.5152453.

Lane, D., Davies, J., Casalino, G., Bartolini, G., Cannata, G., Veruggio, G., et al. (1997). AMADEUS: advanced manipulation for deep underwater sampling. *IEEE Robotics & Automation Magazine*, (ISSN: 1556-4967) *4*(4), 34–45. http://dx.doi.org/10.1109/100.637804.

Marani, G., & Yuh, J. (2014). Introduction to autonomous manipulation case study with an underwater robot, SAUVIM. In *Springer tracts in advanced robotics*: *Vol. 102*. ISBN: 978-3-642-54612-9, Berlin Heidelberg: Springer-Verlag. http://dx.doi.org/10.1007/978-3-642-54613-6.

Noé, S., Beck, T., Foubert, A., & Grehan, A. (2006). *Surface samples in belgica mound province hovland mound province, west rockall bank and northern porcupine bank, tech. rep.*. Universität Bremen, in: Ratmeyer, V. Hebbeln, D. & Shipboard Party: Report and Preliminary Results of RV Meteor Cruise M61/3.

O'Connor, M., Lewis, T., & Dalton, G. (2013). Weather window analysis of irish west coast wave data with relevance to operations & maintenance of marine renewables. *Renewable Energy*, (ISSN: 0960-1481) *52*(Suppl. C), 57–66. https://doi.org/10.1016/j.renene.2012.10.021.

Omerdic, E., Toal, D., Dooly, G., Miller, L., & Coleman, J. (2012). Smart ROV LATIS in action: Sea trials. *IFAC Proceedings Volumes*, (ISSN: 1474-6670) *45*(5), 281–286 3rd IFAC Workshop on Navigation, Guidance and Control of Underwater Vehicles https://doi.org/10.3182/20120410-3-PT-4028.00047.

Omerdic, E., Toal, D., & Leahy, M. (2010). Assistive tools for system integration, deployment, monitoring, and maintenance of ocean energy devices. *Proceedings of the Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment*, *224*(3), 155–172.

Ribas, D., Ridao, P., Turetta, A., Melchiorri, C., Palli, G., Fernandez, J. J., et al. (2015). I-AUV mechatronics integration for the TRIDENT FP7 project. *IEEE/ASME Transactions on Mechatronics*, *PP*(99), 1–10 (ISSN: 083-4435). http://dx.doi.org/10.1109/TMECH.2395413.

Rossi, M., Scaradozzi, D., Drap, P., Recanatini, P., Dooly, G., Omerdić, E., et al. (2015). Real-time reconstruction of underwater environments: From 2D to 3D. In *OCEANS 2015 - MTS/IEEE washington* (pp. 1–6).

Sanderson, A., & Weiss, L. (1980). Image-based visual servo control using relational graph error signals. *Proceedings of the IEEE*, *1074*, 1074–1077.

Schettini, R., & Corchs, S. (2010). Underwater image processing: State of the art of restoration and image enhancement methods. *EURASIP Journal on Advances in Signal Processing*, (ISSN: 1687-6180) *2010*(1), 746052. http://dx.doi.org/10.1155//746052.

Siciliano, B., Sciavicco, L., Villani, L., & Oriolo, G. (2009). *Robotics modelling, planning and control*. London: Springer-Verlag.

Simetti, E., Casalino, G., Torelli, S., Sperindé, A., & Turetta, A. (2014). Floating underwater manipulation: Developed control methodology and experimental validation within the TRIDENT project. *Journal of Field Robotics*, (ISSN: 1556-4967) *31*(3), 364–385. http://dx.doi.org/10.1002/rob.21497.

Sivčev, S., Coleman, J., Adley, D., Dooly, G., Omerdić, E., & Toal, D. (2015). Closing the gap between industrial robots and underwater manipulators. In *OCEANS 2015 - MTS/IEEE Washington*. (pp. 1–7).

Thuilot, B., Martinet, P., Cordesses, L., & Gallice, J. (2002). Position based visual servoing: keeping the object in the field of vision. In *Proceedings 2002 IEEE international conference on robotics and automation, Cat. No.02CH37292, vol. 2* (pp. 1624–1629). . http://dx.doi.org/10.1109/ROBOT.1014775.

Toal, D., Omerdic, E., & Dooly, G. (2011). Precision navigation sensors facilitate full auto pilot control of Smart ROV for ocean energy applications. In *2011 IEEE SENSORS proceedings* (pp. 1897–1900). (ISSN:1930-0395). http://dx.doi.org/10.1109/ICSENS.6127381.

Tsai, R., & Lenz, R. (1989). A new technique for fully autonomous and efficient 3D robotics hand/eye calibration. *IEEE Transactions on Robotics and Automation*, (ISSN: 1042296X) *5*(3), 345–358. http://dx.doi.org/10.1109/70.34770.

Wang, H., Rock, S., & Lees, M. (1995). Experiments in automatic retrieval of underwater objects with an AUV. In *Challenges of our changing global environment conference proceedings. OCEANS '95 MTS/IEEE, vol. 1* (pp. 366–373). IEEE. http://dx.doi.org/10.1109/OCEANS.1995.526796. ISBN:0-933957-14-9, ISSN:01977385.

Yuh, J. (2000). Underwater robotics. In *Proceedings 2000 ICRA millennium conference. IEEE international conference on robotics and automation. symposia proceedings, Cat. No.00CH37065, vol. 1*. ISBN: 0-7803-5886-4, (pp. 932–937). IEEE. http://dx.doi.org/10.1109/ROBOT.2000.844168.

Zhang, Z. (2000). A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*(11), 1330–1334. http://dx.doi.org/10.1109/34.888718. 01628828.