System identification and fault accommodation for thruster propelled UUVs

NWA Sowerby BSc, MSc, Mechatronics Research Centre, University of Wales, Newport, UK
E Omerdic BSc, MSc, PhD, Electronic and Computer Engineering, University of Limerick, Ireland and
GN Roberts BSc, MSc, PhD, Control Theory and Applications Centre, Coventry University, UK

This paper describes the current state of the work done at The University of Wales College Newport under the 'improve the performance of remotely operated vehicles' programmme (IMPROVES). The work incorporates advances in two interrelated parts of the project: it presents a technique to extract velocity from a low-cost MEMS inertial measurement unit for system identification and development of a combined control allocation and fault detection technique. The paper presents theoretical developments and results obtained on experimental trials.

INTRODUCTION

Improving the performance of unmanned underwater vehicles (UUVs) is an extremely active area of research. During the past few years it has been the goal of many research groups and UUV manufacturers to incorporate the benefits of system identification, fault accommodation, robust controller design, navigation and mission planning so as to enhance the capacity of the vehicles to achieve their missions with economy and reliability.

In early 2001, a new research programme was funded by the United Kingdom's Engineering and Physical Sciences Research Council (EPSRC): IMPROVES. This programme is a partnership between The University of Wales, Newport, The University of Plymouth, Southampton University and Seaeye Marine Ltd. The aim of this research project is to improve the dynamic performance of advanced, multi-mission UUVs through the design and development of new, robust, predictive control and on-line fault detection and handling subsystems. These subsystems are being developed and evaluated using existing comprehensive simulation packages for ROV dynamics and tested using two ROVs with contrasting configurations and, consequently, significantly different dynamic characteristics: *Subzero III* is a torpedo-shaped test vehicle developed at the University of Southampton, and *Falcon* is a

AUTHORS' BIOGRAPHIES

Nathan Sowerby graduated from Loughborough University's Department of Mechanical Engineering, UK in 2000 with a BEng in Engineering Science and Technology. This was followed by an MSc in Advanced Control from the University of Sheffield in 2001. He worked towards a PhD thesis covering the application of AI techniques to the system identification of underwater vehicles at the University of Wales, Newport up to 2004 and is currently working as design engineer at Seaeye Marine Ltd.

Edin Omerdic graduated from the University of Zagreb, Faculty of Electrical Engineering and Computing, Croatia in 1997 and received an M.S. degree in electrical engineering (automatic control) from the same University in 2001. He was awarded a PhD from the University of Wales, Newport in 2004 for a thesis concerning the development of fault detection and data handling systems for underwater vehicles. He is presently a Research Fellow at the University of Limerick continuing this line of research.

Geoff Roberts is Professor of Mechatronics in the Control Theory and Applications Centre, Coventry University. He has been active in the area of marine control systems for many years and has contributed to the development of control strategies for ship steering, rudder roll stabilisation, integrated rudder/fin roll stabilisation and guidance and control of underwater vehicles. He is presently working on fault detection and data handling for underwater vehicles, parallel multi mode controllers for ship stabilisation, and applications of intelligent control for guidance and control of land and marine vehicles. He is chairman of the Mechatronics Forum, a member of the IFAC Technical Committee on Marine Systems and a member of the IFAC Technical Committee on Mechatronic Systems.

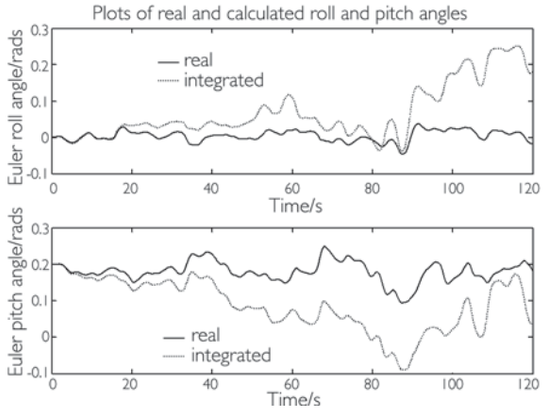
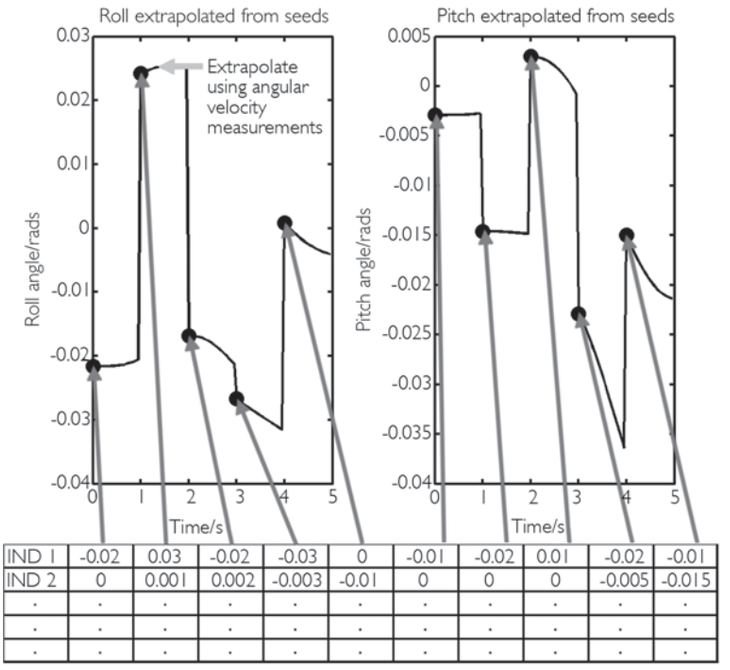


Fig 1: Earth-frame orientation calculated by integrating angular velocity



IND 1	-0.02	0.03	-0.02	-0.03	0	-0.01	-0.02	0.01	-0.02	-0.01
IND 2	0	0.001	0.002	-0.003	-0.01	0	0	0	-0.005	-0.015
.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.

Fig 2: Composition and use of individuals in EA

commercial vehicle for operations in coastal or inshore waters produced by Seaeye Marine Ltd.[1]

The material presented in this paper describes the state of current work on two related areas of the project; namely, *control allocation and fault accommodation and system identification*. The work is presented with application to the *Falcon* vehicle. The next section introduces a technique for extracting data from a MEMS inertial measurement unit that ultimately should allow system identification of an open-frame ROV in a cost-effective manner. This is followed by a novel method for allocating control signals to thrusters in over-actuated degrees-of-freedom while allowing accommodation of thruster faults of varying severity. These techniques have been developed and verified using simulations based on models taken from the literature and experiments performed during tank-testing trials.[2]

SYSTEM IDENTIFICATION USING A MEMS INERTIAL MEASUREMENT UNIT

Open-frame ROVs are becoming increasingly high-performance. As their power-to-weight ratio increases and size decreases, the hydrodynamic non-linearities become increasingly noticeable and due to the complicated shape of the open-frame the turbulence around the vehicle is obvious to the observer. Consequently these vehicles require practice before undertaking complicated missions; making velocity control a desirable feature to ease the burden on the pilot. The control problems that these ROVs generate are still unsolved and testing such a vehicle is expensive and time-consuming. An accurate dynamic model is required to run simulations for controller development; such a simulation also being desirable for pilot training and marketing. A number of work packages of the IMPROVES project were dedicated to this task.

The open-frame ROV presents a difficult system to identify; not only are there significant non-linearities but there is strong coupling between the outputs. It is further complicated by the fact that the desired output of the model is velocity and the linear velocity is a problematic state to measure when underwater. As this work has been undertaken in close collaboration with a commercial company, there is a very real-world and practical emphasis to the techniques developed so the sensors used had to be ones that would be cost-effective to use commercially. Due to the four degree-of-freedom (dof) control that the pilot has over the vehicle and the significant coupling between all 6 dof (pitch and roll are not controlled, but such rotation is induced by a change in surge and sway velocity), all three components of both linear and angular velocity must be measured and the least expensive method of measuring all six dof of a ROV is by means of an Inertial Measurement Unit (IMU) based on Micro-Electro-Mechanical System (MEMS) technology.[3] Unfortunately this unit measures linear acceleration (a mixture of Earth's gravity and the acceleration of the vehicle), from which the linear velocity must be inferred and as the MEMS sensors are corrupted by a non-Gaussian noise this is a significant problem. Data fusion techniques can be used to provide a more accurate representation of the dynamics by combining the IMU signals with information from GPS, compasses, acoustic modems, depth sensors, visual sensors or Venturi meters.[3,4,5,6] The addition of sensors adds cost and it was thought that, as the IMU was to be fixed to a system with largely predictable dynamics, an increase of the performance of the IMU would be found by looking at statistical and model-based methods. In this section a technique using an evolutionary algorithm is proposed to estimate the earth-frame orientation of the vehicle from the noisy angular velocity and linear acceleration measurements, thereby enabling the contribution of earth gravity to be subtracted from the accelerometer measurements facilitating the inference of linear velocity.

The procedures described in this section have been developed using a simulation based on the non-linear equations of motion of

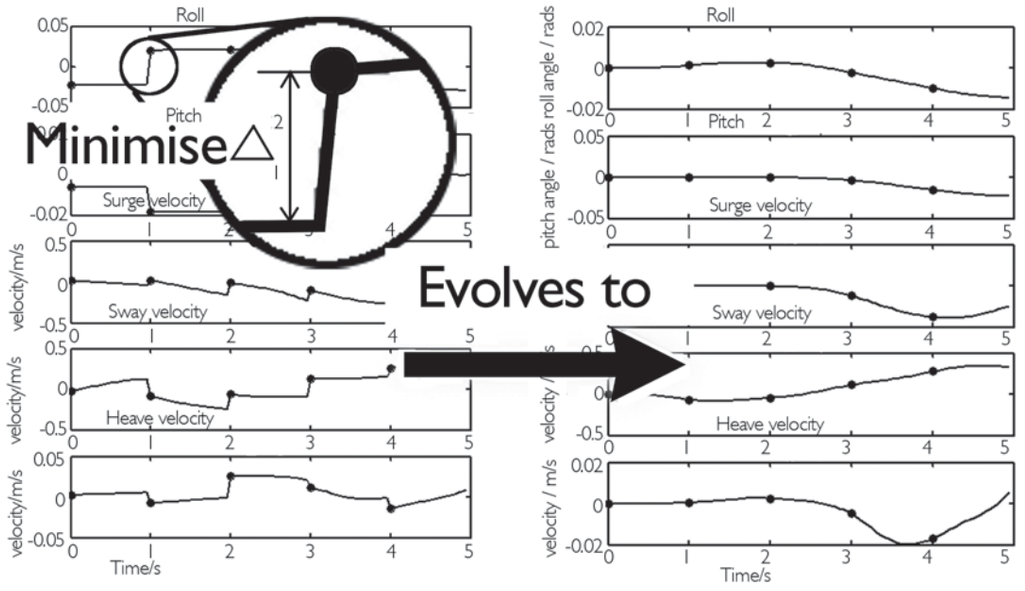


Fig 3: Operation of cost function J_1

an underwater vehicle and the thruster dynamics.[2,7] The simulation was not accurately parameter fitted but the outputs have been tuned to match, approximately, the output of a *Seaeye Falcon ROV* (in terms of PSD and magnitude). The acceleration and angular velocity outputs of the simulation were corrupted by noise measured from a stationary MEMS IMU. The procedures are verified using real-world test data: tank-trials were undertaken at the Ocean Basin Tank, QinetiQ, Haslar, UK; where random noise was used to drive the *Falcon* and the resulting motion measured by both a MEMS IMU and a highly accurate CDL MiniPOS Ring-Laser Gyro (RLG).[8] The MEMS unit was constructed around three Analog Devices ADXRS300 gyroscopes and a Crossbow CXL02LF3 triple-axis accelerometer giving an accuracy of 0.1deg/s and 0.3mg. It is worth noting that the MEMS IMU is a small and lightweight device compared to a *Falcon ROV* but the RLG added about 10% to the mass of the ROV and significantly changed the dynamics of the vehicle (drag, moment of inertia, centre of gravity, centre of buoyancy etc). This would not be suitable for a long term solution particularly as it would also triple the cost of the product. The purpose of this trial was to provide data to verify that orientation could be calculated accurately by the MEMS IMU on an underwater vehicle.

Calculation of linear velocity from inertial measurements

In order to remove the Earth's gravity from the accelerometer measurements, the orientation of the IMU must be found with respect to the Earth. As explained in Fossen[2] the integration of body-frame angular velocity ($\mathbf{v}_2=[p,q,r]^T$) to give earth-frame orientation ($\boldsymbol{\eta}_2=[\phi,\theta,\psi]^T$) is a numerical procedure. When the earth-frame orientation is represented by Euler angles, only the roll (ϕ) and pitch (θ) are required to calculate the body-frame linear acceleration ($\dot{\mathbf{v}}_1 = [\dot{u},\dot{v},\dot{w}]^T$),

$$\begin{bmatrix}\dot{u}\\ \dot{v}\\ \dot{w}\end{bmatrix}=\begin{bmatrix}\dot{u}_m + g.\sin\theta\\ \dot{v}_m - g.\cos\theta\sin\phi\\ \dot{w}_m - g.\cos\theta\cos\phi\end{bmatrix} \quad (1)$$

Where, u, v and w are the surge, sway and heave velocities, respectively, and the subscript m indicates the measured signal. g represents the magnitude of earth's gravity. ϕ and θ are calculated, sequentially, as follows,

$$\begin{bmatrix}\phi(k+1)\\ \theta(k+1)\end{bmatrix}=\begin{bmatrix}T_s p+\phi(k)+\dot{\psi}(k)\sin\theta(k)\\ T_s q\sec\phi(k)+\theta(k)-\dot{\psi}(k)\tan\phi(k)\cos\theta(k)\end{bmatrix} \quad (2)$$

Where T_s is the sampling period and $\dot{\psi}(k)$ is approximated by,

$$\begin{aligned}\dot{\psi}(k) &= (\psi(k+1)-\psi(k))/T_s\\ &= \frac{T_s(r(k)\cos\phi(k)+q(k)\sin\phi(k))}{\cos\theta(k)}\end{aligned} \quad (3)$$

The body-frame linear acceleration components can now be integrated to give linear velocity. When this procedure is applied to noisy simulation data (Fig 1), the effect of the noise is clear. Integration alone cannot be used to find orientation and linear velocity from this data even over just a couple of minutes. Linear filtering techniques do not help here as the PSD of the non-Gaussian noise overlaps with that of the angular velocity (the main component of the noise is very low frequency drift - <0.015Hz).

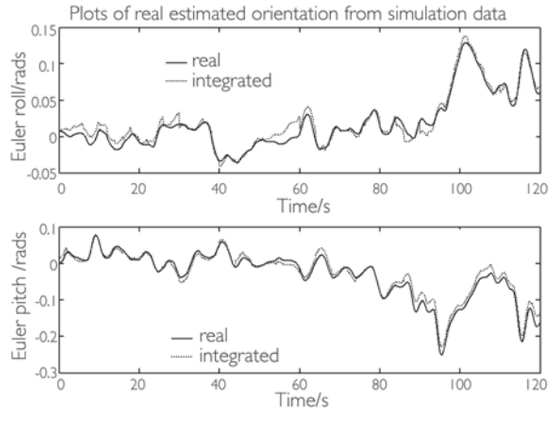


Fig 4: Earth-frame orientation calculated using the two-stage EA with simulation data

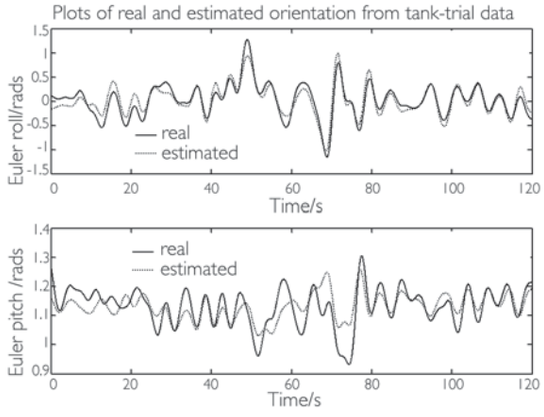


Fig 5: Earth-frame orientation calculated using the two-stage EA with tank-testing data

Introduction to Evolutionary Algorithms (EAs)

EAs are a form of gradient descent optimisation. Unlike most gradient descent techniques the EA approaches the cost landscape from many different points at once; increasing the likelihood of avoiding local minima. The basic principle by which the procedure works is based on natural selection: the 'chromosome' of an 'individual' is composed of the parameters that are to be optimised and a 'population' of these individuals are bred, with the probability of an individual breeding being dictated by its performance with the cost function. The breeding process involves mixing the chromosomes of two individuals in some way – splicing or taking a weighted sum of both are common techniques. During the breeding process mutation can be introduced to help avoid local minima. This is achieved by randomly replacing sections of chromosomes with a random number. A framework has been found to exploit the benefits of this approach to find the orientation of the IMU.

A framework for evolution of earth-frame orientation

From Fig 1 it can be seen that the integration procedure is reasonably accurate for about 2 seconds (40 samples for this system at a sample rate of 20Hz). Therefore, if an accurate sample can be found every 40 samples (call this sample a seed), integration of angular velocity can be used to fill in the gaps (Fig 2). This motivates the use of an EA to evolve individuals with chromosomes composed of 'seeds', from which orientation can be integrated using the angular velocity measurements, to provide an accurate progression of Euler angles.

Cost-functions for evolution of earth-frame orientation

Two cost functions have been found that lead effectively to the global minimum of the true roll and pitch orientation: one based on finding a progression of Euler angles that gives smooth signals and one based on minimising modelling error of linear velocity.

			Error (e) / rads	
			Max(abs(e))	std(e)
Simulation (50 000 samples)	Roll	J1	0.034	0.022
		J2	0.023	0.007
	Pitch	J1	0.038	0.022
		J2	0.029	0.011
Tank testing (24 000 samples)	Roll	J1	0.33	0.26
		J2	0.29	0.13
	Pitch	J1	0.34	0.063
		J2	0.13	0.047

Table 1: Statistics of the errors arising from applying the different cost functions to simulation and tank-testing data

Smoothing extrapolated sections

When the population is initialised with random numbers, the extrapolated sections are very mismatched. This effect is exacerbated when the extrapolated roll and pitch is used to calculate the linear velocity from the measured acceleration. As shown in Fig 3 the difference can be found between the end of one section and the start of another. Therefore by minimising the difference between the end of each section and the start of the following section a smooth signal can be found. Because the Euler roll and pitch and the three components of linear velocity are highly cross-coupled, this causes rapid convergence to a neighbourhood around the true orientation. This cost function actually evolves velocity as well and, as such, needs a population of extended individuals that include velocity seeds. Hence,

$$S = \sum_{k=1}^{n-1} abs(\Delta_k^{k+1}) \tag{4}$$

$$J_1 = S_\phi + S_\theta + S_u + S_v + S_w \tag{5}$$

Where, n represents the number of sections and Δ_k^{k+1} is the difference in magnitude between the end of section k and the start of section k+1.

Modelling linear velocity

A slower but more accurate cost function is to actually model the linear velocity (resulting from equation 1) and use the modelling error as the cost – if there is a component of earth-g in the estimated body-frame acceleration, it will cause unpredictability, increased complexity and, therefore, increased modelling error in the resulting linear velocity. The velocity is modelled using an auto-regressive with eXogenous inputs (ARX) structure[9] and the modelling error is calculated as the sum of squared error. Hence,

$$J_2 = (\mathbf{u} - \hat{\mathbf{u}})^T (\mathbf{u} - \hat{\mathbf{u}}) \times (\mathbf{v} - \hat{\mathbf{v}})^T (\mathbf{v} - \hat{\mathbf{v}}) \tag{6}$$

Where $\hat{\mathbf{u}}$ is a vector of the surge velocity model output and $\hat{\mathbf{v}}$ is a vector of the sway velocity model output. As the body-frame heave acceleration is calculated using the cosines of roll and pitch (see equation 1) and these angles rarely exceed 20 degrees, it is far less dependent on the orientation

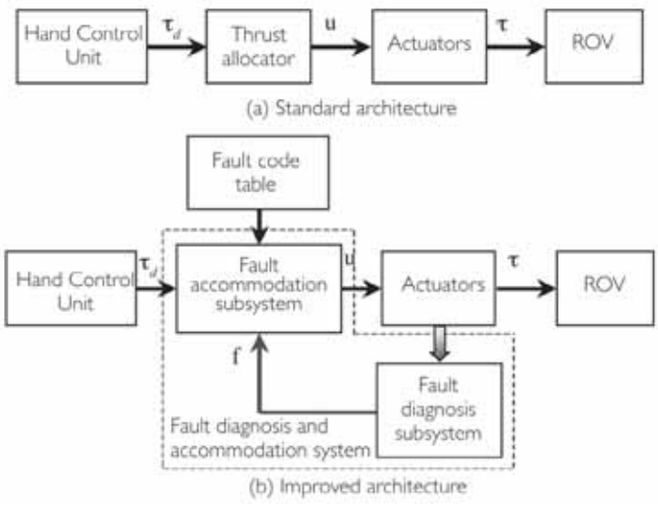


Fig 6: Open-loop control structure on ROVs

and thus contributes little to the cost function if its modelling error is added. However it does add considerably to the computation time so it is not used here.

A two stage process to find earth-frame orientation

Cost function J_1 (equation 5) causes rapid convergence to a neighbourhood around the true orientation from a completely random start but it takes a long time to converge to the exact orientation. Cost function J_2 (equation 6) takes a long time to converge from a completely random start point (if it does not get trapped in a local minimum), but will converge relatively quickly on the true result from a reasonable initial guess. Therefore, J_1 is used to build up a population of seeds in the neighbourhood of the true result and J_2 uses these values to get closer to the actual orientation.

Results

Fig 4 shows the results when J_1 and J_2 are applied successively to two minutes of noisy simulation data. When J_1 is applied the data is split up into 12 segments and for each, 10 sections (giving 10 seeds) of 20 samples are evolved from a population of 50 individuals. A high rate of mutation is used and the process is run for 500 generations. The individuals were initialised with completely random data spanning the full range of possible angles (+/-π rads). To initialise the J_2 stage, the output of the J_1 stage is taken and individuals are created by using random numbers in a band +/-0.1rads wide around the output. For this stage the data is split up into 6 segments, each being partitioned into 10 sections of 40 samples for evolution. Again, 50 individuals are used with a high rate of mutation but the algorithm is only run for 100 generations. All of these parameters have been found, heuristically, to obtain a minimum computation time and robustness.

Fig 5 shows the results when the two-stage process is applied to data from the tank-trials and Table 1 shows the results when the process is applied to 50 000 samples of simulation data and 24 000 samples of tank-trial data. These results show a significant discrepancy between simulation and system data. There are several reasons for this:

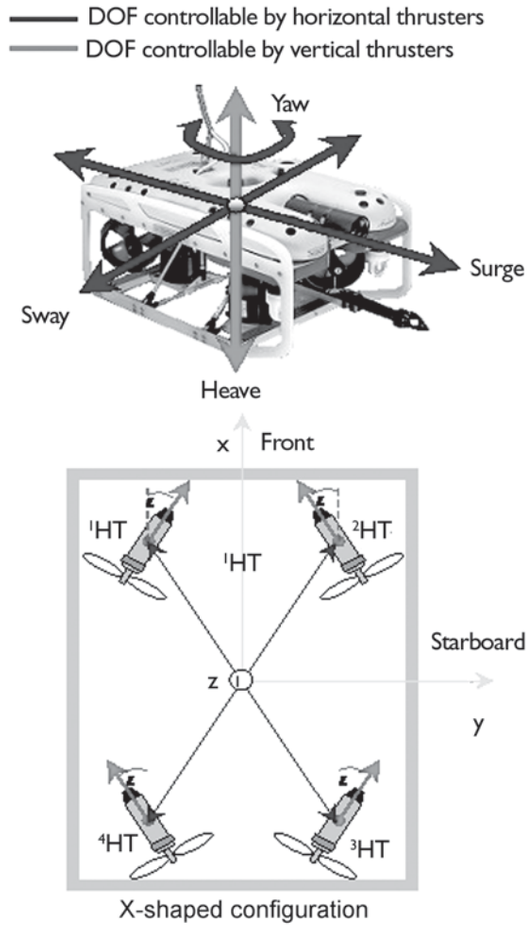


Fig 7: X-shaped configuration of horizontal thrusters

1. During the tank-trials, the two IMUs were mounted at different orientations with respect to the vehicle (hence the large pitch angle in Fig 5) and these positions could not be accurately measured at the tank. Therefore, the relative orientation was found by a least-squares technique using the calculated orientation.
2. The fastest sample rate of the RLG was 3Hz, whereas the sample rate of the MEMS IMU was 20Hz. Errors were introduced through interpolating and anit-alias filtering the RLG data. Also both units were interrogated by separate surface units so their time-difference had to be found by inspection of the results.
3. Finally due to the low mass of the ROV the umbilical cable exerts a significant influence on the dynamics of the ROV that is difficult to predict. If the cable has moved to a different position on the frame during the epoch of calculation, this may cause modelling error that the cost function tries to compensate for by evolving an erroneous orientation.

THRUSTER FAULT DIAGNOSIS AND ACCOMMODATION

A large number of open-frame underwater vehicles have no other actuators except thrusters. This section describes new approaches associated with optimal control allocation and thruster fault diagnosis and accommodation for this class of underwater vehicles. The work described herein is applicable

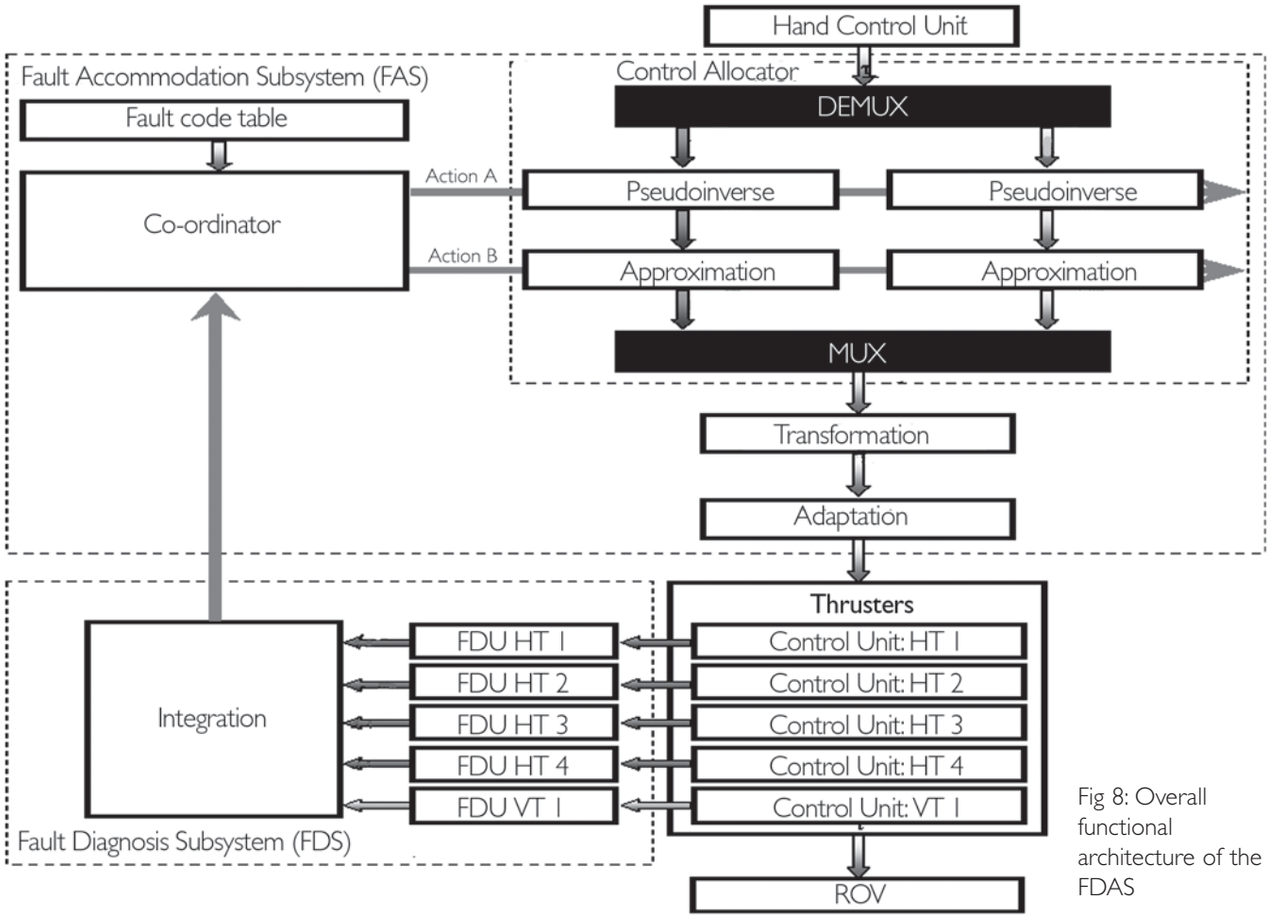


Fig 8: Overall functional architecture of the FDAS

to a wide class of open-frame underwater vehicles. However the application is focused on two ROVs with different thruster configurations. This section expands upon the work previously reported by various authors.[10,11,12,13]

Control architecture

A standard open-loop ROV control structure is shown in Fig 6(a). The ROV pilot uses the hand control unit (HCU) to generate vector $\boldsymbol{\tau}_d$, which can be interpreted as a desired vector of propulsion forces and moments among axes in the body-fixed frame (virtual control input). The control allocator maps the vector $\boldsymbol{\tau}_d$ into the vector (true control input) $\mathbf{u}$, representing control settings for individual actuators (thrusters). After input from $\mathbf{u}$, the actuators generate a vector of propulsion forces and moments (total control effect) $\boldsymbol{\tau}$, which is applied as the input to the ROV dynamics block and determine the behaviour of the vehicle. The main objective of the control allocation is to ensure that the condition $\boldsymbol{\tau}=\boldsymbol{\tau}_d$ is satisfied for all attainable $\boldsymbol{\tau}_d$.

An improved control architecture, which includes a novel fault diagnosis and accommodation system (FDAS) is depicted in Fig 6(b). The thruster allocator in the standard structure is replaced by the Fault Accommodation Subsystem (FAS) in the improved control structure. The FAS is a novel hybrid approach for thruster allocation based on the integration of the pseudoinverse and the fixed-point iteration method (Omerdic & Roberts, 2004b).[13] It is implemented as a two-step process. The pseudoinverse solution is found in the first step then the feasibility of the solution is examined analysing its individual components. If violation of actuator constraint(s) is detected the fixed-point iteration method is activated in the second step. In this way, the hybrid approach is able to allocate the exact solution, optimal in the l_2 sense, inside the entire attainable command set. This solution minimises a weighted control energy cost function; the most suitable criteria for underwater applications. The primary task of the thruster allocation is enhanced with the fault diagnosis subsystem (FDS), able to monitor the state of the thrusters and inform the FAS about any malfunctions using the total fault indicator vector $\mathbf{f}$, carrying the codes of faulty states for each thruster. The FDS uses fault detector units (FDUs) associated with each thruster to monitor their state. Robust and reliable FDUs are based on integration of self-organising maps and fuzzy logic clustering methods. These units are able to detect internal and external faulty states of thrusters. The FAS uses information provided by the FDS to accommodate faults by performing an appropriate reconfiguration, ie, to reallocate control energy among operable thrusters. Reconfiguration is performed by increasing the weight penalty of a faulty thruster and restricting the corresponding saturation bounds using pre-determined levels for each fault type. Actions are created and stored in the fault code table (FCT) in advance during the off-line training of FDUs. The overall fault diagnosis and accommodation process is very fast, due to the computational efficiency of the FDAS algorithm, where the heaviest numerical calculations are performed off-line in advance.

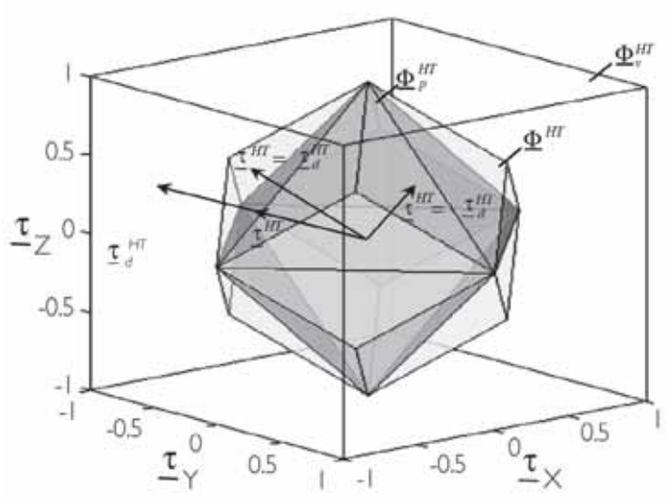


Fig 9: Partitions of the normalised virtual control space $\underline{\Phi}_v^{HT}$ for motion in the horizontal plane ($\underline{\Phi}_v^{HT}$- feasible region for pseudoinverse, $\underline{\Phi}_v^{HT}$- attainable command set, X-shaped thruster configuration)

Architecture of the FDAS

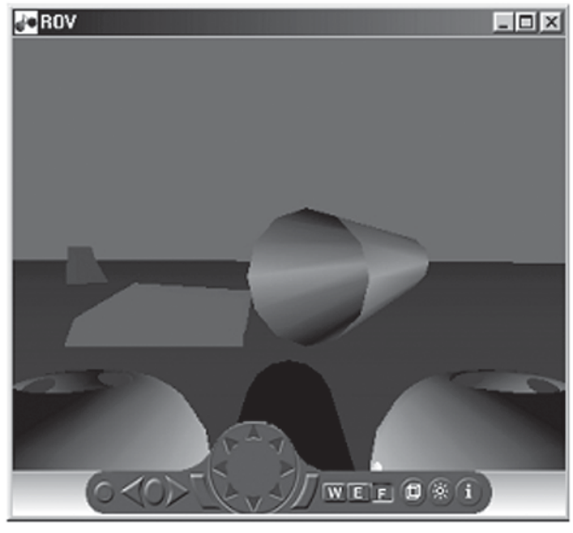
The overall functional architecture of the proposed FDAS shown in Fig 8, represents the expanded version of the improved architecture shown in Fig 6(b). The input to the FDAS is the vector $\boldsymbol{\tau}_d$ generated by the HCU. The output of the FDAS is the vector of desired thruster velocities **n**, transformed into a form that is compatible with the thruster control unit. There follows a short description of the individual components.

FDS

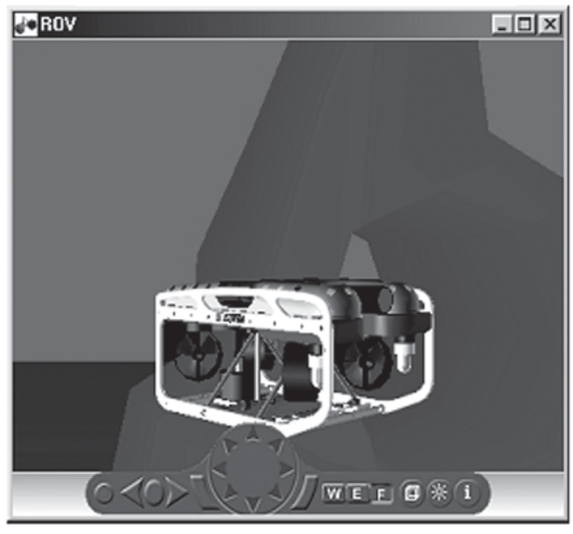
- FDU: The FDS uses FDUs to monitor the state of the thrusters. The FDU is a software module associated with the thruster, able to detect internal faults (eg, temperature of the windings exceeds limits) and external faults (eg, a jammed propeller. The output of the FDU is a fault indicator f_i, the code of the fault.
- Integration: The fault indicators f_i are integrated into the *total fault indicator vector* **f** inside this block. The vector **f** is a carrier of all thrusters' states.

FAS

- Demux: This block symbolically indicates the separation of the vector $\boldsymbol{\tau}_d$ into two parts: $\boldsymbol{\tau}_d^{HT}$, representing the dof (surge, sway and yaw) controllable by horizontal thrusters, and $\boldsymbol{\tau}_d^{VI}$, representing the dof (heave) controllable by vertical thrusters.
- Pseudoinverse: This block finds weighted pseudoinverse solutions $\mathbf{u}^{HT}$ and $\mathbf{u}^{VT}$ of the control allocation problem, separately for horizontal and vertical thrusters.
- Approximation: The weighted pseudoinverse solution $\mathbf{u}^{HT}$ can be feasible (satisfies all constraints) or unfeasible (violates some constraint(s)). The output $\mathbf{u}^{*HT}$ of this block must be a feasible solution all the time. Hence, if $\mathbf{u}^{HT}$ is feasible, then $\mathbf{u}^{*HT}=\mathbf{u}^{HT}$. Otherwise, truncation (clipping of components that exceed their constraints) or scaling is performed to find a feasible approximation $\mathbf{u}^{*HT}$.
- Co-ordinator: The role of this block is to undertake remedial actions in accordance to the context of the total fault indicator vector **f** and the instructions, stored in the fault code table. For each possible fault type corresponding actions A and B are stored in the FCT. The action A is related with the weight updates of weighting matrices, used to find the weighted pseudoinverse solution. The action B is related with a change of constraint bounds, in accordance to the fault type.
- Mux: This block performs the inverse role of the Demux block, ie, it merges feasible solutions $\mathbf{u}^{*HT}$ and $\mathbf{u}^{*VT}$ into composite solution vector $\mathbf{u}^{*}$.
- Transformation: The vector $\mathbf{u}^{*}$ cannot be directly applied to drive thrusters. It must be transformed into the vector of desired thruster velocities **n**. This block performs this function, using transformation $n_i = \operatorname{sgn} u_i \sqrt{|u_i|}$ for each component.
- Adaptation: Different thruster control units accept data in different format. For example, the desired angular velocity for the TCU of the *Falcon* must be represented as an integer number between -100 and +100. In contrast, the same variable must be converted into the voltage in order

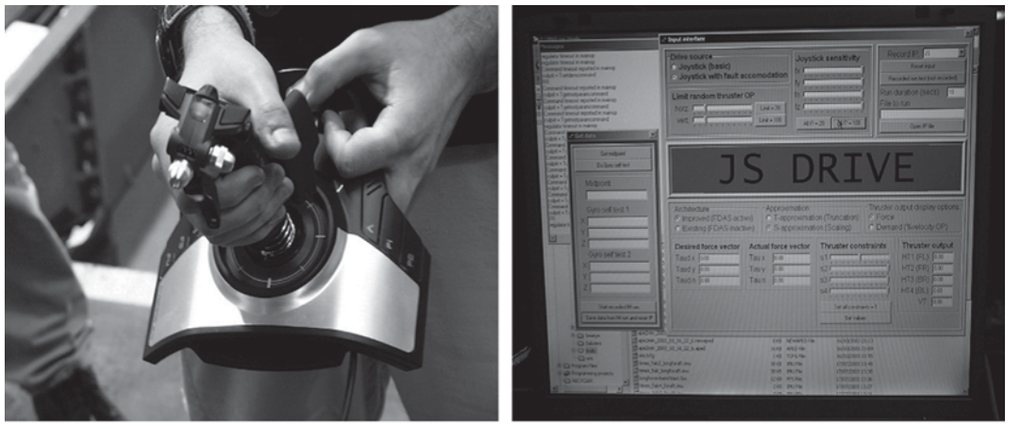


(a) Camera front view



(b) Camera side view

Fig 10: Virtual underwater world



(a) Joystick as input device

(b) control application

Fig 11: The ATC control software

to be applied to drive a thruster on *Uris*. This block transforms the vector $\underline{\mathbf{n}}$ into the vector $\mathbf{n}$, which has the form adapted for the particular TCU. Vector $\mathbf{n}$ is used to drive the thrusters, which generate a vector of propulsion forces and moments $\boldsymbol{\tau}$. A standard pseudoinverse method allocates a feasible solution only on a subset $\underline{\boldsymbol{\Phi}}_p^{HT}$ of the attainable command set $\underline{\boldsymbol{\Phi}}^{HT}$ (Fig 9).

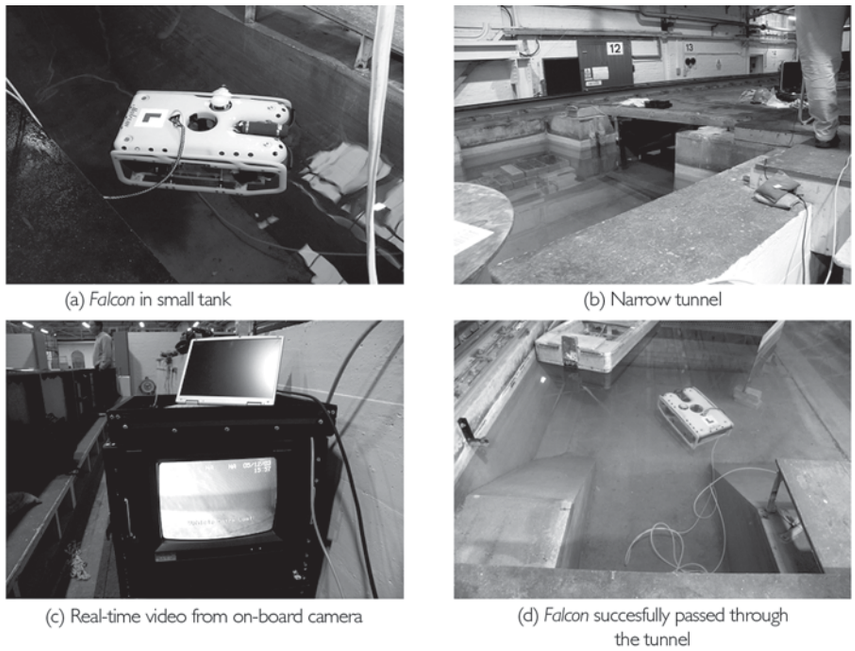
A hybrid approach for control allocation, implemented in the FDAS, integrates the pseudoinverse and fixed-point iteration method. By introducing fixed-point iterations, the feasible region is extended to the entire set and the obtained solution is optimal in the l_2 sense. That is the proposed FDAS guarantees that the condition $\boldsymbol{\tau}=\boldsymbol{\tau}_d$ is satisfied for all $\boldsymbol{\tau}_d^{HT}$ that lies inside $\underline{\boldsymbol{\Phi}}^{HT}$. Otherwise, the solution obtained by the FDAS is a very good approximation that lies on the boundary $\partial\left(\underline{\boldsymbol{\Omega}}^{HT}\right)$, which depends on design parameters (weighting matrices) and the type of approximation.

ROV simulator

Experimenting with a real ROV is time consuming and expensive. A dynamic model is useful for simulation purposes and for investigation of different control algorithms. In order to investigate different approaches for the design of a fault diagnosis and accommodation system, it is necessary to use a realistic model of the vehicle and propulsion system. A general ROV simulator has been developed in Matlab/Simulink environment. This model includes the nonlinear model of an ROV with 6 dof motion, propulsion systems, hand control unit and an interface for the fault detection and accommodation system. In order to enhance the user interface and improve understanding of the proposed fault detection and accommodation approach, a realistic virtual underwater world environment was developed. There are two view points: camera front view and camera side view. Camera front view is body-fixed and gives the picture similar to one from the on-board camera. Camera side view is a view of the vehicle from a distance with adjustable latitude and longitude. A joystick is used as the input device to generate command signals. Different fault conditions can be triggered while the simulation is running in order to investigate the performance of the FDAS in faulty situations.

Experiment results

A number of test trials with *Falcon* were scheduled for the period December 2003 to January 2004, performed at the



(a) *Falcon* in small tank

(b) Narrow tunnel

(c) Real-time video from on-board camera

(d) *Falcon* succesfully passed through the tunnel

Fig 12: Experimental set-up

QinetiQ Ocean Basin Tank at Haslar in December. Two-day's worth of experiments were dedicated to the evaluation of the proposed FDAS and comparison of the performance between the standard control architecture (FDAS not active) and improved control architecture (FDAS active). Preliminary results of these experiments are presented in this section.

In order to test the performance of the FDAS, the standard surface components (HCU and main surface unit) in the distributed control system of *Falcon* were replaced by the joystick and laptop. The laptop was connected to the control network via an RS232-RS485 converter. The joystick was used as the input device to generate command input signals (Fig 11(a)). Both control architectures were implemented using the control application, ATC (advanced thruster control); developed in the Borland Delphi programming environment (Fig 11(b)). This software implements the FDAS for real-time control of *Falcon*. The ATC was used as the master node to control thruster slave nodes. The HCU controls: SPEED, LAT SPEED and TURN RATE are implemented in the ATC as scaling factors for joystick sensitivity (slider controls): f_x, f_y and f_N, respectively. The hybrid approach for control allocation, implemented in the ATC, enables manual selection of saturation bounds s_i^{HT} for each horizontal thruster ${}^iHT$ using slider controls. Different faulty situations can be artificially generated by changing the values of these sliders. The slider values $s_i^{HT}=1$, $0<s_i^{HT}<1$ and $s_i^{HT}=0$ mean a fault-free state, partial fault and total fault in thruster ${}^iHT$, respectively. During the trial, the ATC enables selection of the desired architecture using radio buttons. The control architecture 'FDAS not active' (based on the existing control software for *Falcon*) does not provide for thruster fault accommodation and the only available solution in a faulty situation is to switch off a faulty thruster (regardless of fault type). In contrast, the control architecture 'FDAS active' uses the FDAS to accommodate thruster faults. Switching between different architectures is instantaneous, enabling direct comparison of their performances during the same test trial.

In the first experiment, *Falcon* was placed in a small tank (Fig 12(a)). Different test trials were performed with both architectures (separate and combined motion in four dof: surge, sway, yaw and heave). The limited tank size required a reduction of joystick sensitivity in order to prevent damage to the vehicle. Different faulty situations were injected on-line during the test trial. Poor performance was obtained in faulty situations in cases when the FDAS was not active, due to imperfect manual compensation of moment components induced by disabling a faulty thruster. In contrast, the control architecture with an active FDAS gave superior performance. All three dof in the horizontal plane were fully controllable. Although the manoeuvring space was limited due to the tank size, the control of *Falcon* was as easy as in the fault-free case. It is interesting to note that, from the ROV pilot's point of view, no change in the response of the vehicle was noticeable in faulty situations, despite the limited usage of the faulty thruster. The reason for invariant behaviour of the vehicle in fault-free and faulty situations is because reduced joystick sensitivity resulted in limitation in the size of the desired control vector, such that it lies inside the attainable command set throughout the experiment, whereby the FDAS is always able to allocate the exact solution of the control allocation problem.

In the second experiment, *Falcon* was driven through the narrow, shallow tunnel which links two parts of the tank (Fig 12(b)). A total fault in ${}^2HT$ was artificially generated by setting the slider value s_2^{HT} to zero; that is, ${}^2HT$ was disabled and removed from the thruster allocation process and the mission had to be performed with only three working horizontal thrusters. This experiment is similar to 'passing-through-the-pipe' experiments in the ROV simulator (Fig 10(a)). After the *Falcon* entered the tunnel, real-time video from an on-board camera was used as feedback to control the motion (Fig 12(c)). The experience gained from simulations resulted in successful completion of the mission (Fig 12(d)). It should be emphasized that an undesired drag effect of the umbilical cable was observable during the experiment. This effect was particularly noticeable for low-speed motion of the vehicle.

CONCLUSIONS

A technique has been found that will estimate the orientation of an MEMS IMU that relies on the inertial measurements alone. This technique possesses several important attributes for real-world use:

- It does not require any estimation of initial conditions; in a real operating environment, the vehicle will probably be switched on and launched from a ship which is obviously not a stationary platform.
- Although it is model-based, the model only has to remain valid for a short epoch; which suits an open-frame ROV, as the dynamics of the vehicle change over time (through a change in configuration as new tools are used or as the environment changes, eg, with a change in salinity or current).
- It is not susceptible to the low-frequency null-point drift that is characteristic of MEMS sensors.

This orientation calculation will enable the next step of velocity calculation to be undertaken: removing earth-g from the accelerometer measurements to allow body-frame acceleration to be integrated, giving the linear body-frame velocity required for system identification and control. Although the results do not show that this technique is suitable or accurate enough to allow real-time velocity measurement, it is expected that it will provide the foundation of a procedure that is capable of this.

A novel thruster fault detection and accommodation system for overactuated open-frame underwater vehicles has been presented. The proposed FDAS provides an optimal solution for the distribution of propulsion forces for a fault-free case and with a fault in a single thruster, which minimises a control energy cost function. This optimal solution is important as it means that a maximum operational time can be obtained from an AUV's battery, and minimum usage of a thruster means the maximisation of its life. The proposed FDAS is a part of the low-level control layer and its modular design enables easy integration into existing control laws. Future work will include implementation of the proposed approach and its integration into the existing control architecture of *Falcon*. Special attention will be devoted to the design of a universal controller, robust to a partial/total fault in a single thruster. An important part of this work will be the integration of the feasible region with the real-time video presented to the ROV pilot from the on-board camera.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the following:

- The EPSRC for providing funding for this project, under grant reference GR/N39456/01.
- CDL Ltd UK for the loan of a MiniPOS IMU.
- Everyone at Seaeye Marine Ltd. who contributed time and equipment to the testing of *Falcon*; particularly Jon Robertson and Miles Merckel.

REFERENCES

1. Seaeye. (2004). *Seaeye Falcon. [Online], http://www.seaeye.com/products10.html* (accessed 10th November 2004).

2. Fossen TI. (1994). Guidance and Control of Ocean Vehicles, John Wiley and Sons.

3. Love M, Hamilton A and Katebi R. *A comparison of single and multi axis attitude estimation in strapdown inertial navigation systems*, Proc. of the 2003 IFAC workshop on Guidance and Control of Underwater Vehicles, Newport, South Wales, UK, 9th to 11th April 2003. pp. 89-94.

4. Grewal MS and Andrews AP. *Kalman filtering theory and practice using MATLAB second edition*. John Wiley and Sons (2001).

5. *DVLNAV integrated navigation system for manned and unmanned submersibles*. 1st IFAC Workshop on Guidance and Control of Underwater Vehicles GCUV 03 (pp. 83-88). 9th –11th April 2003, Newport, South Wales, UK.

6. Loebis D, Dagliesh FR, Sutton R, Tetlow S, Chudley J and Allwood RL. *An integrated approach in the design of a navigation system for an AUV*. 6th IFAC Conference on Manoeuvring and Control of Marine Craft (MCMC 2003). (pp.319-324) Girona, Spain.

7. Fossen TI and Blanke M. (2000). *Nonlinear output feedback control of underwater vehicle propellers using feedback from estimated axial flow velocity*. IEEE Journal of Oceanic Engineering. 25(2) pp.241-255, 2000.

8. CDL. *CDL MiniPOS manual v1.03 [online], http://www.cdltd.net* (accessed on 23rd May, 2004).

9. Ljung L. *System identification theory for the user, (2nd Ed)*. Prentice Hall, 1999.

10. Omerdic E and Roberts GN. *Thruster fault accommodation for underwater vehicles*. 1st IFAC Workshop on Guidance and Control of Underwater Vehicles GCUV 03 (pp. 221-226). 9th –11th April 2003, Newport, South Wales, UK.

11. Omerdic E, Roberts GN and Ridao P. *Fault detection and accommodation for ROVs*. 6th IFAC Conference on Manoeuvring and Control of Marine Craft (MCMC 2003), Girona, Spain.

12. Omerdic E and Roberts GN (2004a). *Thruster fault diagnosis and accommodation for open-frame underwater vehicles*. Control Engineering Practice. 12 (12) pp. 1575-1598.

13. Omerdic E and Roberts GN (2004b). *Extension of feasible region of control allocation for open-frame underwater vehicles*. IFAC Conference on Control Applications in Marine Systems (CAMS 2004), Ancona, Italy. pp. 315-320.