

ULRR

Application of thruster fault-tolerant control system based on the hybrid method for control allocation in real-world environment

Item Type	Article
Authors	Omerdić, Edin;Toal, Daniel;Dooly, Gerard
Citation	IFAC-PapersOnLine;52 (21), pp. 277-282
Publisher	Elsevier
Download date	2026-03-11 06:51:51
Item License	https://creativecommons.org/licenses/by-nc-sa/1.0/
Link to Item	https://hdl.handle.net/10344/8563

Application of Thruster Fault-Tolerant Control System based on the Hybrid Method for Control Allocation in Real-World Environment

Edin Omerdic, Daniel Toal, Gerard Dooly

Centre for Robotics & Intelligent Systems, University of Limerick, Ireland
(e-mail: edin.omerdic@ul.ie, daniel.toal@ul.ie, gerard.dooly@ul.ie).

Abstract: Fault-tolerant control system for marine robots, developed by researchers at Centre for Robotics & Intelligent Systems (CRIS), University of Limerick, utilises the hybrid method for control allocation, based on the integration of the pseudoinverse and the fixed-point iteration method. The algorithm is implemented as a two-step process. In the first step, the pseudoinverse solution is found, and the feasibility of the solution is examined analysing its 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 method for control allocation can allocate the exact solution, optimal in the l_2 sense, inside the entire attainable command set. This solution minimises a control energy cost function, the most suitable criteria for underwater applications. The performance of the thruster fault-tolerant control system has been validated during the research cruise CE19001, where the work-class ROV *Étaín*, deployed from support vessel Celtic Explorer via Tether Management System (TMS), successfully performed several complex subsea tasks with a faulty horizontal thruster. This paper describes the algorithm, provides geometric interpretation and presents the results from field trials.

Keywords: fault-tolerant control, fault accommodation, control allocation.

1. INTRODUCTION

Various strategies for fault-tolerant control systems for AUVs and ROVs have been proposed in the literature. Fault tolerant control allocation using unknown input observers is proposed in (Cristofaro & Johansen, 2014). The proposed method extended ideas from (Chen *et al.* 1996) by representing faults as time-varying efficiency factors, active use of control allocation and input redundancy, unique identification of faulty actuators/effectors without need for fault estimation and robust handling of disturbances and/or output noise. The Dynamic Surface Control (DSC) method for underwater ROVs has been proposed in (Baldini *et al.* 2018). Exhaustive simulations have been carried out to compare the performance of the DSC method with respect to different control techniques (PID, backstepping and sliding mode approaches) in fault-free and faulty conditions, with included non-linear effects, such as saturation, actuator dynamics, sensor noises etc. Adaptive fault tolerant control and fault reconstruction problem for AUV subject to ocean current disturbance and modelling uncertainty is investigated in (Liu *et al.* 2018). A virtual closed-loop system is introduced to deal with the influence of the initial tracking error in an ideal environment. Then, the second-order sliding mode observer is constructed to estimate the thruster fault effect.

The paper expands upon the work previously reported by the authors. Thruster fault diagnosis and accommodation system for open-frame underwater vehicles has been proposed in Omerdic & Roberts (2004). The extension of the approach with the inclusion of fixed-point iterations has been proposed

in Omerdic *et al.* (2004), with the clear geometrical interpretation of the approach demonstrated on the low-dimensional control allocation problem, with two-dimensional virtual control space and the three-dimensional true control space. However, in this paper, the concepts are extended to ROV applications, which have three-dimensional virtual space (three DOF in the horizontal plane) and four-dimensional true control space (four horizontal thrusters).

The paper outline is given in the following. Problem formulation is given in the second section. The third section describes the hybrid approach for control allocation. Section 4 presents the testing and evaluation results of the proposed FTC in a real-world environment. Finally, the fifth section summarises the concluding remarks.

2. PROBLEM FORMULATION

2.1 Thruster Configuration

Plan view of thruster configuration in the horizontal plane is shown in Fig. 1. The axes are chosen to coincide with the principal axes of inertia, and they are defined as:

- x_b - longitudinal axes (directed to front side),
- y_b - transversal axes (directed to starboard),
- z_b - normal axes (directed from top to bottom).

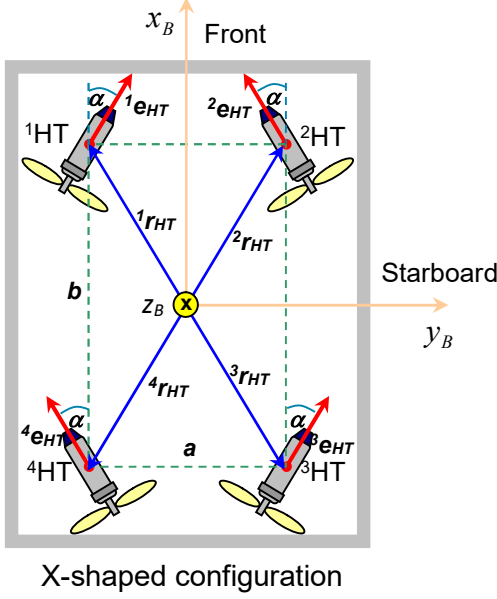


Fig. 1. X-shaped configuration of horizontal thrusters for open-frame underwater vehicles.

2.2 Control Allocation

Using the normalisation method described in (Omerdic & Roberts, 2004), the general constrained problem for open-frame underwater vehicles with X-shaped thruster configuration of horizontal thrusters can be formulated using normalised variables as follows:

For given τ , find \mathbf{u} such that

$$\mathbf{B}\mathbf{u} = \tau \quad (1)$$

subject to constraint

$$\underline{\mathbf{u}} \leq \mathbf{u} \leq \bar{\mathbf{u}} \quad (2)$$

In the following, the problem is analysed in more detail from the general control allocation perspective.

$$\mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} - \text{true control input} \quad (3)$$

$$\tau = \begin{bmatrix} \tau_X \\ \tau_Y \\ \tau_N \end{bmatrix} - \text{virtual control input} \quad (4)$$

$$\mathbf{B} = \begin{bmatrix} \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & -\frac{1}{4} & \frac{1}{4} & -\frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & -\frac{1}{4} & -\frac{1}{4} \\ \frac{1}{4} & -\frac{1}{4} & -\frac{1}{4} & \frac{1}{4} \end{bmatrix} - \text{control effectiveness matrix} \quad (5)$$

$$\underline{\mathbf{u}} = \begin{bmatrix} -1 \\ -1 \\ -1 \\ -1 \end{bmatrix}, \bar{\mathbf{u}} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} - \text{actuator position constraints} \quad (6)$$

Condition (2) determines the constrained control subset Ω , that is, the unit 4D cube in true control space:

$$\Omega = \{\mathbf{u} \in \mathcal{R}^4: \|\mathbf{u}\|_\infty \leq 1\} - \text{constrained control subset} \quad (7)$$

Equation (1) defines a linear transformation from the true control space \mathcal{R}^4 to the virtual control space \mathcal{R}^3 , which maps (projects) the unit 4D cube (the tesseract) Ω into attainable

command set Φ (see Fig. 2), a subset of the virtual control space Φ_v (the unit 3D cube), defined as

$$\Phi_v = \{\tau \in \mathcal{R}^3: \|\tau\|_\infty \leq 1\} \quad (8)$$

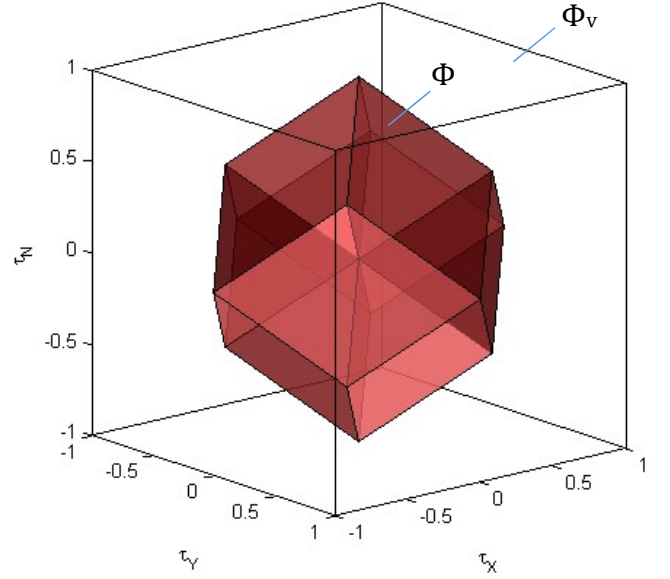


Fig. 2. The attainable command set Φ for the X-shaped configuration of horizontal thrusters (fault-free case).

Considering equation (1) (system of three linear equations with four unknown variables) from a row perspective, the solution set is the hyper line ι (the intersection of three hyper planes in 4D space). The overall solution of the control allocation problem (1) - (2) is a solution set \mathfrak{S} , the intersection of hyper line ι and Ω . The existence and uniqueness of the solution \mathbf{u} depends on the position of the virtual control input τ relative to the attainable command set Φ :

- If τ lies inside Φ , then \mathfrak{S} has an infinite number of points and the control allocation problem has an infinite number of solutions.
- If τ lies on the boundary of Φ , then \mathfrak{S} has a single point, the unique solution for the control allocation problem.
- If τ lies outside Φ , then \mathfrak{S} is an empty set i.e. no exact solution exists, only approximation.

To obtain a unique, "best" solution \mathbf{u}^* from a solution set \mathfrak{S} in case of multiple solutions, it is necessary to introduce criteria, which is minimised by the chosen solution \mathbf{u}^* . In the general case, the optimal true control input is given by the solution to a two-step optimisation problem (Härkegård, 2003):

$$\mathbf{u} = \arg \min \|W_u(\mathbf{u} - \mathbf{u}_p)\|_p, \mathbf{u} \in \Psi \quad (9)$$

$$\Psi = \arg \min \|W_v(\mathbf{B}\mathbf{u} - \tau)\|_p, \mathbf{u} \in \Omega \quad (10)$$

where \mathbf{u}_p represents preferred positions of the actuators (preferred true control input) and W_u and W_v are weighting matrices.

The problem defined with (9) - (10) can be interpreted as follows: given Ψ , the set of feasible control inputs that

minimise $\mathbf{B}\mathbf{u} - \boldsymbol{\tau}$ (weighted by \mathbf{W}_v), find the true control input \mathbf{u} that minimises $\mathbf{u} - \mathbf{u}_p$ (weighted by \mathbf{W}_u). In (9) – (10) \mathbf{u}_p , \mathbf{W}_u , and \mathbf{W}_v are design parameters. It is useful to give a geometric interpretation of the relationship between the choice of the weighting matrix \mathbf{W}_u and the solution of the control allocation problem for fault-free and faulty situations. A set of points \mathbf{u} that satisfy the condition $\|\mathbf{W}_u(\mathbf{u} - \mathbf{u}_p)\|_p \leq r$ is called a weighted hyper sphere $S_{\mathbf{W}_u}(\mathbf{u}_p, r)_p$. The optimal solution \mathbf{u}^* is a point where a family of these hyperspheres, centred at \mathbf{u}_p and weighted by \mathbf{W}_u , “touch” the set Ψ . Shape of these hyperspheres depends on the choice of norm l_p in (9). The non-actuated state of the horizontal thrusters is a preferred state, i.e. preferred true control input in (9) is $\mathbf{u}_p = \mathbf{0}$. The matrix \mathbf{W}_u is a design parameter typically used for actuator prioritisation. If all actuators have the same priority, then \mathbf{W}_u is equal to unit matrix. Otherwise, the weight of the actuator with less priority is increased. In this way, it is possible to accommodate actuator faults by changing the weighting matrix \mathbf{W}_u . For example, if $\mathbf{W}_u = \text{diag}(w_1, w_2, w_3, w_4) = \text{diag}(1, 1 + \Delta w_2, 1, 1)$, then the faulty horizontal thruster HT₂ is penalised for $\Delta w_2 > 0$ and its contribution to the total control effort is less than in the fault-free case ($\Delta w_2 = 0$). Geometrically, hyperspheres in the virtual control space are compressed along the second control axis in faulty case and become “deformed” hyper spheres, i.e. hyper ellipsoids. Additional introduction of nonzero off-diagonal elements into the symmetric matrix \mathbf{W}_u yields to the rotation of these hyper ellipsoids. The matrix \mathbf{W}_v allows for prioritisation among the virtual control inputs when the problem (9) – (10) has no exact solution. In the following, it is assumed that $p = 2$, i.e. that the l_2 norm is used as a measure of how good a solution (or approximation) is. This norm represents a measure of control effort and l_2 norm of solution (9) can be interpreted as control energy cost function.

3. HYBRID METHOD

3.1 Description

The hybrid method for control allocation is based on the integration of the pseudoinverse and the fixed-point method and implemented as a two-step process. The weighted pseudoinverse solution is found in the first step. Then the feasibility of the solution is examined analysing individual components of the solution. If violation of actuator constraint(s) is detected, the fixed-point iteration method is activated in the second step, which results in a guaranteed feasible solution. In this way, the hybrid approach can allocate the exact solution, optimal in the l_2 sense, inside the entire attainable command set. This solution minimises the control energy cost function, which is the most suitable criteria for underwater applications.

3.2 Pseudoinverse

The first step of the hybrid method relies on the fact that an explicit solution to the unconstrained control allocation problem

$$\min \|\mathbf{W}_u \mathbf{u}\|_2 \quad (11)$$

subject to $\mathbf{B}\mathbf{u} = \boldsymbol{\tau}_d$ is given by

$$\mathbf{u} = \mathbf{B}_{\mathbf{W}_u}^\dagger \boldsymbol{\tau}_d \quad (12)$$

Where the matrix $\mathbf{B}_{\mathbf{W}_u}^\dagger$

$$\mathbf{B}_{\mathbf{W}_u}^\dagger = \mathbf{W}_u^{-1}(\mathbf{B}\mathbf{W}_u^{-1})^\dagger = \mathbf{W}_u^{-1}\mathbf{B}^T(\mathbf{B}\mathbf{W}_u^{-1}\mathbf{B}^T)^{-1} \quad (13)$$

is the weighted pseudoinverse of \mathbf{B} (Omerdic, 2009). For the X-shaped configuration of horizontal thrusters the matrix $\mathbf{B}_{\mathbf{W}_u}^\dagger$ is given by

$$\mathbf{B}_{\mathbf{W}_u}^\dagger = \frac{2}{\sum_{i=1}^4 w_i} \begin{bmatrix} w_3 + w_4 & w_2 + w_3 & w_2 + w_4 \\ w_3 + w_4 & -(w_1 + w_4) & -(w_1 + w_3) \\ w_1 + w_2 & w_1 + w_4 & -(w_2 + w_4) \\ w_1 + w_2 & -(w_2 + w_3) & w_1 + w_3 \end{bmatrix} \quad (14)$$

Feasibility of pseudoinverse solution (12) depends on the position of virtual control input $\boldsymbol{\tau}_d$ inside the virtual control space Φ_v . For a fault-free case, when \mathbf{W}_u is the unit matrix, the shape of the feasible region for the pseudoinverse Φ_p , defined as a set of virtual control inputs $\boldsymbol{\tau}_d$ that satisfy (12) subject to $\mathbf{u} \in \Omega$, is shown in Fig. 3. The boundary of a convex polyhedron Φ_p represents a set of all virtual control inputs for which at least one component of the pseudoinverse solution (12) receives extreme value. This boundary is determined by the intersection of the four pairs of parallel planes π_k^+ (π_k^-), $k = \overline{1,4}$, where the plane π_k^+ (π_k^-) represents a set of all virtual control inputs for which k^{th} component of solution (11) satisfies $u_k = +1$ (-1).

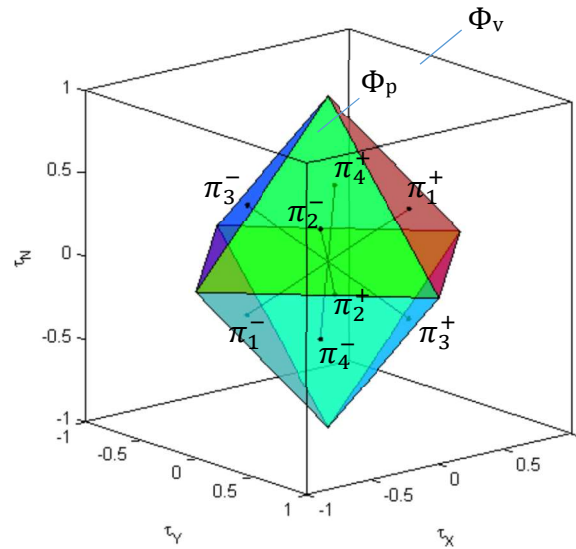


Fig. 3. The feasible region for pseudoinverse Φ_p inside the virtual control space Φ_v for X-shaped configuration of horizontal thrusters (fault-free case).

The weighted pseudoinverse can find the exact feasible solution of the control allocation problem, optimal in the l_2 sense, only if $\boldsymbol{\tau}_d \in \Phi_p$. Geometrically, equation (12) defines a linear transformation from the virtual control space Φ_v to the hyper plane Π in the true control space \mathfrak{R}^4 . The intersection of

this hyper plane with the hyper line ι (solution of $\mathbf{B}\mathbf{u} = \boldsymbol{\tau}_d$) is the point \mathbf{u}^* where the family of weighted hyper spheres “touch” the hyper line ι . Since $\boldsymbol{\tau}_d \in \Phi_p$, this intersection determines a feasible solution $\mathbf{u}^* = \mathbf{B}_{W_u}^\dagger \boldsymbol{\tau}_d \in \Omega$, optimal in l_2 sense. The main issue with the weighted pseudoinverse is that the feasible region for pseudoinverse Φ_p is a subset of attainable command set Φ ($\Phi_p \subset \Phi$), and the pseudoinverse cannot find an optimal solution on the entire Φ .

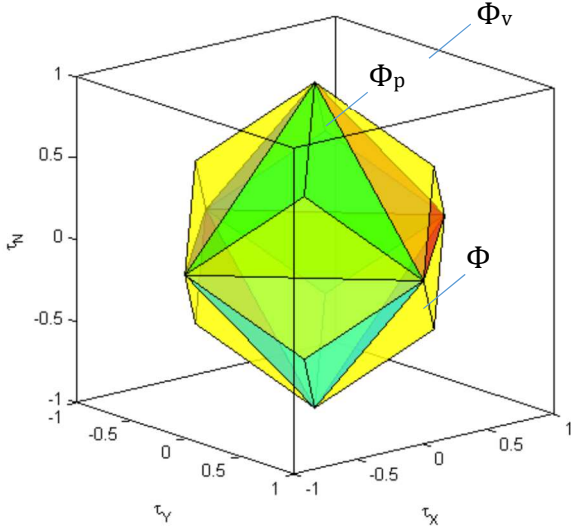


Fig. 4. Partitions of the virtual control space Φ_v .

The virtual control space Φ_v can be partitioned into characteristic regions, as indicated in Fig. 4. The two characteristic regions inside Φ_v are Φ_p and $\Phi \cap \Phi_p$. The weighted pseudoinverse is able to find the exact feasible solution of the control allocation problem, optimal in l_2 sense, only if $\boldsymbol{\tau}_d \in \Phi_p$. Otherwise, for $\boldsymbol{\tau}_d \in \Phi \setminus \Phi_p$, the solution obtained by pseudoinverse is unfeasible. However, the fixed point iteration method can find the solution, optimal in l_2 sense, for case $\boldsymbol{\tau}_d \in \Phi \setminus \Phi_p$. Finally, for case $\boldsymbol{\tau}_d \in \Phi_v \setminus \Phi$, the solution does not exist and approximation must be used (for example T-approximation (truncation) or S-approximation (scaling), as described in Omerdic (2009)).

3.3 Fixed-Point Method

The fixed-point method (Bodson, 2002) finds the control vector \mathbf{u} that minimises

$$J(\mathbf{u}) = (1 - \varepsilon) \|\mathbf{W}_v(\mathbf{B}\mathbf{u} - \boldsymbol{\tau}_d)\|_2^2 + \varepsilon \|\mathbf{W}_u \mathbf{u}\|_2^2 \quad (15)$$

subject to (2), where $|\varepsilon| < 1$. The algorithm proceeds by iterating on the equation

$$\mathbf{u}_{k+1} = \text{sat}[(1 - \varepsilon)\eta \mathbf{B}^T \mathbf{Q}_1 \boldsymbol{\tau}_d - (\eta \mathbf{H} - \mathbf{I})\mathbf{u}_k] \quad (16)$$

where

$$\mathbf{Q}_1 = \mathbf{W}_v^T \mathbf{W}_v \quad (17)$$

$$\mathbf{Q}_2 = \mathbf{W}_u^T \mathbf{W}_u \quad (18)$$

$$\mathbf{H} = (1 - \varepsilon) \mathbf{B}^T \mathbf{Q}_1 \mathbf{B} + \varepsilon \mathbf{Q}_2 \quad (19)$$

$$\eta = 1/\|\mathbf{H}\|_2 \quad (20)$$

and $\text{sat}[\mathbf{u}]$ is the saturation function that clips the components of the vector \mathbf{u} to their limits. The condition for stopping the iteration process is

$$|J(\mathbf{u}_{k+1}) - J(\mathbf{u}_k)| < \text{tol} \quad (21)$$

The fixed-point algorithm is very simple, and most computations need to be performed only once before iterations start. The only condition for the initial point is $\mathbf{u}_0 \in \Omega$. However, the number of iterations depends on the desired accuracy and the choice of the initial point (solution). To improve the efficiency, Burken, *et al.* (2001) suggest selecting the initial point \mathbf{u}_0 as the true control input calculated at the previous time sample, i.e. $\mathbf{u}_0(t) = \mathbf{u}(t - T)$. Design parameters of the fixed-point method are \mathbf{W}_u , \mathbf{W}_v , ε and tol .

For case $\boldsymbol{\tau}_d \in \Phi \setminus \Phi_p$ the pseudoinverse solution (intersection of pseudoinverse hyper plane Π and hyper line ι) is unfeasible i.e. it lies outside Ω . However, the solution set \mathfrak{S} is not empty and the fixed-point method is triggered in this case, which is able to find the optimal solution in l_2 sense in \mathfrak{S} . The choice of the parameter ε is delicate, since it affects the trade-off between the primary and the secondary optimisation objectives, as well as the convergence of the iteration algorithm (16).

3.4 Algorithm (Hybrid Method)

Step 1: For given $\boldsymbol{\tau}_d$, find the weighted pseudoinverse solution.

Step 2: If the solution is feasible, go to Step 4.

Step 3: Use the fixed-point method (section 3.3) to find a feasible solution.

Step 4: Perform additional modifications (correction for non-symmetrical thrust-speed curves, scaling, type casting, etc.).

Step 5: Deliver the output to the Thruster Interface module.

It should be emphasized that the number of fixed-point iterations, performed to find the feasible solution for cases when the pseudoinverse solution is unfeasible, depends on the desired accuracy and the choice of the design parameters. The desired accuracy is closely related to requirements imposed by the Thruster Interface module. For example, some thruster control protocols require desired velocities to be presented as integer numbers between -100 and +100. This means that the true control space for motion in the horizontal plane is discretised by the uniform grid of $201^4 = 1.632240801 \cdot 10^9$ discrete control vectors and each solution must be rounded to the closest point in the grid. Design parameters must be chosen to take into account these issues.

It is useful to visualise the change in shapes of the attainable command set Φ and the feasible region for the pseudoinverse Φ_p depending on thruster faulty states. In the fault-free state, all thrusters are equally prioritised, and the actuator position constraint in (6) is $|u_k| \leq 1$ for each component. In faulty case, faulty thruster HT_i is penalised such that $|u_i| \leq s_i < 1$.

The numerical value of the constraint bound s_i depends on the type of the fault. Change in the constraint bound yields to increase of corresponding weight in the weighting matrix \mathbf{W}_u

$$w_i = 1 + \Delta w_i \quad (22)$$

where (Omerdic, 2009)

$$\Delta w_i = 2 \left(\frac{1}{s_i} - 1 \right) \quad (23)$$

In faulty case, restriction of constraints bounds reduces the size of the constrained control subset Ω , i.e. Ω is “clipped” to new actuator position constraint s_i for an i -th component of \mathbf{u} . The image of the “clipped” Ω under mapping (1) is reduced attainable command set Φ . Hence, the feasible region for the pseudoinverse Φ_p and the attainable command set Φ decrease in the case of a fault in the single thruster. In particular, Fig. 5. (a) displays Φ_p only and Fig. 5. (b) $\Phi_p \& \Phi$ for the case of a fault in HT₂ ($s_2 = 0.5, w_2 = 3$). Analysing (12) and (14), the geometrical interpretation of change in the shape of Φ_p can be obtained by observing that the change of weight w_2 yields the change of slopes of the planes π_k^- and π_k^+ , $k \in \{1,3,4\}$, while the planes π_2^- and π_2^+ with actuator position constraints -0.5 and $+0.5$ respectively moved closer to the origin, staying parallel, without changing their slopes. The image of the “clipped” constrained control subset Ω is the reduced attainable command set Φ , as shown in Fig. 5 (b). It can be seen that the relation $\Phi \supset \Phi_p$ is still valid, i.e. the pseudoinverse is not able to find a feasible solution for case $\boldsymbol{\tau}_d \in \Phi \setminus \Phi_p$. Similar to the fault-free case, the fixed-point method is activated in this case, able to find a feasible solution optimal in l_2 sense. In this way, the hybrid method allocates the entire attainable command set in an optimal way, despite the limited (restricted) usage of a faulty thruster.

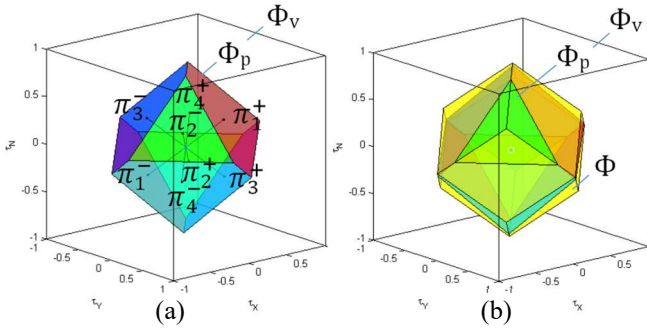


Fig. 5. Partitions of the virtual control space Φ_v for the case of a partial fault in thruster HT₂: (a) - Φ_p , (b) - $\Phi_p \& \Phi$.

Partitions of the virtual control space for the extreme case (shutdown of HT₂) is shown in Fig. 6. In particular, Fig. 6. (a) displays Φ_p only and Fig. 6. (b) $\Phi_p \& \Phi$ for the case of a total fault in HT₂ ($s_2 = 0.0, w_2 \rightarrow \infty$). This means that the thruster HT₂ is disabled and the redundancy is eliminated by removing the variable u_2 and the second column of matrix \mathbf{B} from (1). The modified thruster control matrix \mathbf{B} is a non-singular 3×3 square matrix, and the problem can be solved in a standard way. The ratio of volumes of attainable command sets in Fig. 5. (b) & 6. (b) relative to the full size Φ (Fig. 2.), can be used as a measure of loss in manoeuvring capabilities.

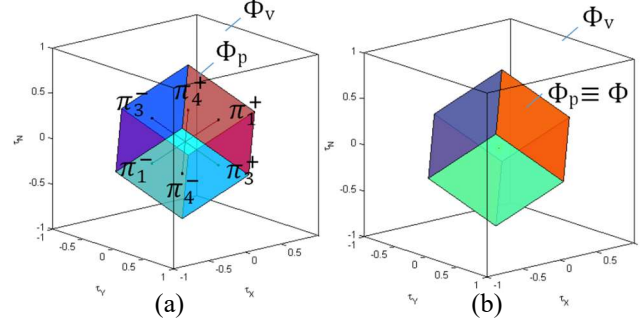


Fig. 6. Partitions of the virtual control space Φ_v for the case of a total fault in thruster HT₂: (a) - Φ_p , (b) - $\Phi_p \& \Phi$.

The constrained control subset Ω becomes the 3D unit cube, a subset in 4D virtual control space, defined as

$$\Omega = \{(u_1, 0, u_3, u_4) \in \mathbb{R}^4: |u_1| \leq 1, |u_3| \leq 1, |u_4| \leq 1\} \quad (24)$$

Ω is mapped by modified \mathbf{B} to Φ , which coincides with Φ_p , as indicated in Fig. 6. (b).

4. TESTING IN REAL-WORLD ENVIRONMENT

The hybrid method for control allocation, described in this paper, has been used as a foundation to build the thruster fault-tolerant control system (FTC). The performance of FTC has been evaluated and validated during research cruise CE19001 in January 2019, when the work-class ROV *Étain* (Fig. 7.), deployed from support vessel Celtic Explorer via Tether Management System (TMS), successfully performed several complex subsea tasks with a simulated fault in the horizontal front-left thruster HT₁ ($s_1 = 0.0, w_1 \rightarrow \infty$) (Fig. 8. (c)).

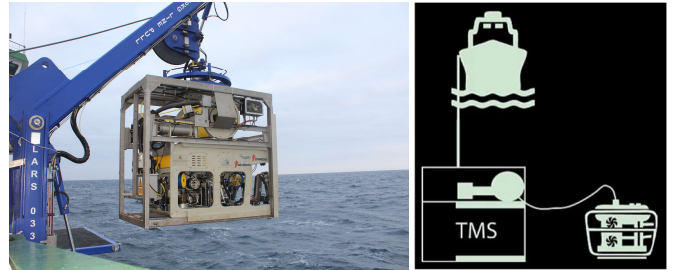


Fig. 7. Work-class ROV *Étain* onboard RV Celtic Explorer.

4.1 Circular Arc

The main task was to move the ROV with a constant velocity of 0.2m/s along the circular arc, with heading pointing toward the centre P_4 (Fig. 8. (a)-(b)). The screenshots of CCW and CW trajectories are shown in Fig. 8. (a) and Fig. 8. (b), respectively. Desired surge speed was set to 0m/s, while desired sway speed was set to $+0.2$ m/s (CCW motion) and -0.2 m/s (CW motion). The heading controller used the “Fixed Point” mode, with the ROV heading directed toward the fixed point P_4 . In both cases, ROV trajectories were almost perfect circles, demonstrating a high quality of the thruster FTC in performing complex subsea manoeuvring. The same task has been trialled in the manual mode (with a joystick as an input device) by professional ROV pilot with disabled Fault Accommodation, but the performance was not satisfactory.

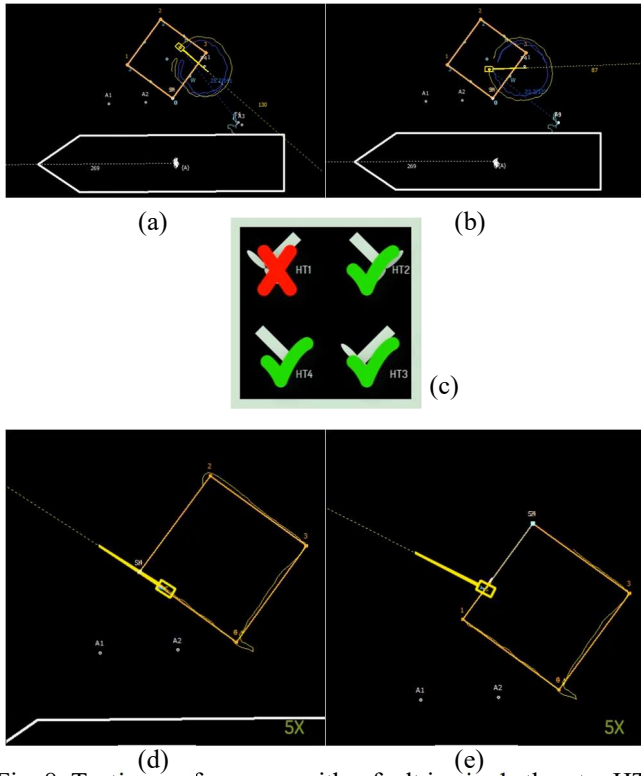


Fig. 8. Testing performance with a fault in single thruster HT_1 (c); circular arc: (a) CCW motion, (b) CW motion; path following: (d) before corner WP_1 , (e) after corner WP_1 .

4.2 Path Following

For this test, ROV had to follow the path (square) defined with four points WP_0 , WP_1 , WP_2 , and WP_3 with constant speed 0.2m/s and constant heading 300° . The screenshots of the main pilot screen during the subsea path following with a fault in a single thruster is shown in Fig. 8. In particular, the ROV position approaching the point WP_1 is shown in Fig. 8. (d). After reaching WP_1 , the vehicle continued to move toward WP_2 along the straight-line segment. The same task has been given to a professional ROV pilot to execute under the same fault conditions, with disabled Fault Accommodation module (Fig. 9). The ROV pilot had to maintain the heading and speed by applying manual compensation for unbalanced moment components, due to lack of contribution from the faulty thruster. Non-perfect compensation led to the poor tracking performance and oscillatory character of the heading response.

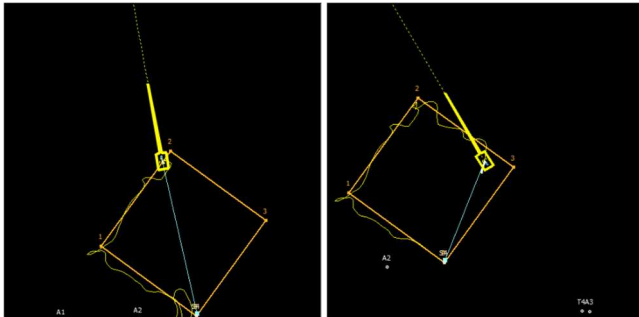


Fig. 9. Path following performed by professional ROV pilot in manual mode, with disabled Fault Accommodation module.

6. CONCLUSIONS

This paper presented a hybrid method for control allocation, able to allocate feasible solution of the control allocation problem, optimal in l_2 sense, on the entire attainable command set in real-time. The performance of the thruster FTC system, whose main pillar is the hybrid method, has been evaluated with work-class ROV through a series of complex, challenging subsea tasks. Results show that the FTC provides automatic reallocation in faulty situations, keeping all three DOF in the horizontal plane fully controllable, making it possible to control the motion of the faulty vehicle in satisfactory way.

ACKNOWLEDGEMENT

This paper is based upon works supported by the Science Foundation Ireland under Grant No. 12/RC/2302 for the Marine Renewable and Energy Ireland (MaREI) centre and Infrastructure Grant No. (15/RI/3232).

REFERENCES

- Baldini, A., Ciabattoni, L., Felicetti, R., Ferracuti, F., Freddi, A., and Monteriù, A. (2018). Dynamic surface fault tolerant control for underwater remotely operated vehicles. *ISA Transactions* 78, pages 10-20.
- Bodson, M. (2002). Evaluation of Optimisation Methods for Control Allocation. In *Journal of Guidance, Control and Dynamics*, 25(4), pp. 703-711.
- Burken, J., Lu, P., Wu, Z., and Bahm, C. (2001). Two Reconfigurable Flight-Control Design Methods: Robust Servomechanism and Control Allocation. In *Journal of Guidance, Control and Dynamics*, 24(3), pp. 482-493.
- Chen, J., Patton, R.J., and Zhang, H.Y. (1996). Design of unknown input observers and robust detection filters. *International Journal of Control*, 63, 85-105.
- Cristofaro, A., and Johansen, T.A. (2014). Fault tolerant control allocation using unknown input observers. *Automatica* 50 (7), pages 1891-1897.
- Härkegård, O. (2003). Backstepping and Control Allocation with Application to Flight Control. PhD thesis, Department of Electrical Engineering, Linköping University, Sweden.
- Liu, X., Zhang, M., and Yao, F. (2018). Adaptive fault tolerant control and thruster fault reconstruction for autonomous underwater vehicle. *Ocean Engineering* 155, pages 10-23.
- Omerdic, E.; & Roberts, G.N. (2004). Thruster fault diagnosis and accommodation for open-frame underwater vehicles. *Control Eng. Pract.* 2004, 12, 1575-1598.
- Omerdic, E., Roberts, G.N., and Toal, D. (2004). Extension of Feasible Region of Control Allocation for Open-frame Underwater Vehicles. *CAMS 2004*. Ancona, Italy.
- Omerdic, E. (2009). *Thruster Fault-Tolerant Control*. VDM Verlag, ISBN: 978-3-639-11739-4.