

Article

Geometric Insight into the Control Allocation Problem for Open-Frame ROVs and Visualisation of Solution

Edin Omerdic *^{ID}, Petar Trsljic, Admir Kaknjo, Anthony Weir, Muzaffar Rao^{ID}, Gerard Dooly and Daniel Toal^{ID}

Department of Electronics and Computer Engineering, Centre for Robotics and Intelligent Systems, University of Limerick, V94 T9PX Limerick, Ireland; petar.trsljic@ul.ie (P.T.); kaknjo.admir@gmail.com (A.K.); anthony.weir@ul.ie (A.W.); muzaffar.rao@ul.ie (M.R.); gerard.dooly@ul.ie (G.D.); daniel.toal@ul.ie (D.T.)

* Correspondence: edin.omerdic@ul.ie; Tel.: +353-061-202-355

Received: 8 November 2019; Accepted: 21 January 2020; Published: 29 January 2020



Abstract: The overall control system for an open-frame Remotely Operated Vehicle (ROV) is typically built from three subsystems: guidance, navigation and control (GNC). The control allocation plays a vital role in the control subsystem. Typically, open-frame underwater vehicles have p actuators (thrusters) for the motion in the horizontal plane, and the control allocation problem, in this case, is very complex and hard to visualise, because the normalised constrained control subset is a p -dimensional unit cube. The aim of this paper is to give a clear picture and a geometric interpretation of the problem and to introduce a hybrid method, based on the integration of a weighted pseudoinverse and the fixed-point method. The main idea of the hybrid method is visualised, and the deep geometric insight is provided using a “virtual” ROV in low-dimensional control spaces, including visualisation of the attainable command set, solution lines, control energy spheres and the role of pseudoinverse and fixed-point iterations. The same concepts are then extended to higher-dimensional cases, for open-frame ROV with four X-shaped (vectored) horizontal thrusters, which is one of the most common thruster configurations for commercial ROVs. The proposed hybrid method has been developed, integrated into a generic fault-tolerant ROV control system and evaluated in virtual and real-world environments off the west coast of Ireland using observation-class ROV Latis and work-class ROV Étaín.

Keywords: fault-tolerant control; control allocation; ROV

1. Introduction

In general, a fault is the primary cause of changes in the system structure or parameters that eventually leads to degraded system performance or even the loss of the system function. A Fault-Tolerant Control (FTC) system continuously analyses the behaviour of the plant in order to identify faults and changes the control law to hold the closed-loop system in a region of acceptable performance. Fault tolerance includes two steps [1]: (i) Fault diagnosis (the existence of faults has to be detected, and the faults have to be identified); (ii) Control re-design (the controller has to be adapted to the faulty situation so that the overall system continues to have satisfactory performance). Like faults, disturbances and model uncertainties change the plant behaviour and have similar effects on the system. Disturbances and model uncertainties are nuisances which are known to exist, but whose effects on the system performance are handled by appropriate measures like filtering or robust design. On another side, the FTC system is designed to detect the faults and remove their effects by remedial actions, i.e., it is aimed to change the control law in order to cancel the effects of the faults or to attenuate them to an acceptable level.

Control allocation is an effective approach to implement fault-tolerant control. An overview of control allocation methods is given in [2] (for aerospace applications) and [3] (for marine applications). In survey article [4], authors intended to encourage the cross-disciplinary transfer of ideas between various application fields, with emphasize on optimization-based methods.

Unmanned Underwater Vehicles are subject to faults or failures during underwater missions, and thrusters are one of the most important and most common sources of faults. In all but the most trivial cases, the existence of a fault may lead to mission abortion. Ship time is very expensive, and the implication of small faults could be very costly and time-consuming.

Various fault-tolerant control strategies for unmanned underwater vehicles have been reported in the literature. A fault-tolerant motion control method based on reconstructive approach has been proposed in [5], which is used to ensure the X-rudder Autonomous Underwater Vehicle (AUV) complete the mission in the case of motion actuators failure. An adaptive fault-tolerant controller utilising Radial Basis Function (RBF) neural network sliding mode method was addressed in [6] for a class of Remotely Operated Vehicles (ROVs) with redundant thrusters. Adaptive fault-tolerant control and fault reconstruction problem for AUV subject to ocean current disturbance and modelling uncertainty is investigated in [7]. 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. A control allocation method for compensation for the effect of thrusters' dead zones is proposed in [8]. The Dynamic Surface Control (DSC) method for underwater ROVs has been proposed in [9]. 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. Design of thruster configuration and thrust allocation control for an underactuated ROV is addressed in [10]. The paper proposes the one-norm algorithm and modified singular value method to find appropriate thruster configuration and thrust allocation. Energy-aware fault-mitigation architecture for underwater vehicles has been proposed in [11]. The proposed architecture is capable of detecting faults and monitoring the performance of the thruster subsystem in modern AUVs by observing the use of the onboard resources. Event-Triggered Active Fault-Tolerant Control System is proposed in [12], based on event-triggered transmission mechanism to reduce the amount of data sent to the observer module. The attractive fault-tolerant control scheme for an omni-directional mobile robot with four mecanum wheels, based on the integration of Nonlinear Model Predictive Control module and basic Fault Detection and Isolation scheme, is proposed in [13]. Thrust Distribution Control System for ROVs, based on adaptive back-stepping controller, is proposed in [14]. An overview of existing fault detection and fault-tolerant schemes is presented in [15] and [16].

This paper introduces a geometric insight into the control allocation problem for over actuated open-frame underwater vehicles. The work presented herein is applicable to a wide class of control allocation problems, where the number of actuators is higher than the number of objectives. The paper is aimed to provide deep insight into the geometry of the control allocation problem, and authors hope that it will highlight the problem from a different perspective, based on geometric reasoning.

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 [17]. The extension of the approach with the inclusion of fixed-point iterations is proposed in this paper, and a hybrid method for control allocation is introduced, based on the integration of a weighted pseudoinverse and the fixed-point method. The main idea of the hybrid method is visualised, and the deep geometric insight is provided using a "virtual" ROV in low-dimensional control spaces. Then, the same concepts are extended to higher-dimensional cases, for open-frame ROV with four X-shaped (vectored) horizontal thrusters. The hybrid method for control allocation has been used as a foundation to build the thruster Fault-Tolerant Control system (FTC). The performance of FTC has been evaluated and validated in a virtual and real-world environment. Selected test cases from simulation and field

trials demonstrate the effectiveness of the method and the ability of the FTC to complete the mission in the presence of a single (partial and total) fault in a single thruster.

2. Control Allocation Problem

2.1. Problem Definition

The interconnection between Guidance, Navigation and Control (GNC) subsystems for open-frame ROV is shown in Figure 1.

- **Guidance:** subsystem that continuously computes the reference (desired) position, velocity and acceleration of an ROV to be used by the motion control system.
- **Navigation:** subsystem to determine position/attitude, course, travelled distance and (optionally) velocity and acceleration of an ROV.
- **Control:** subsystem to determine necessary control forces and moments to be provided by the ROV to satisfy certain control objective (in conjunction with the guidance system).

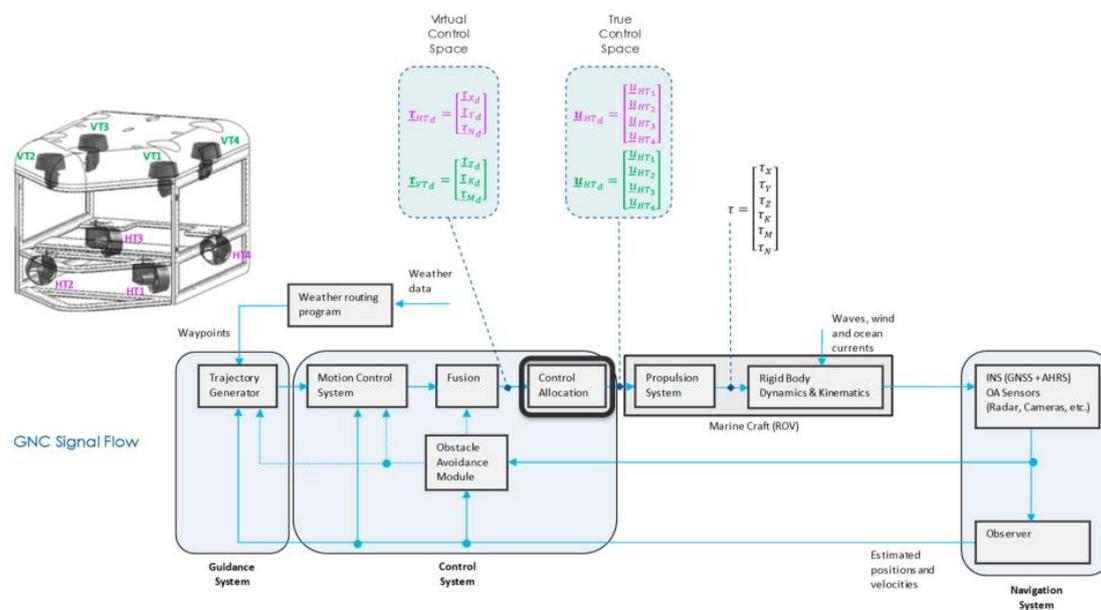


Figure 1. GNC signal flow for open-frame ROV (inspired by [18]).

The control design problem consists of two steps [19]:

- STEP 1 (*Regulation Task*): Design a control law, which specifies desired virtual control input (normalised vector of forces and moments $\boldsymbol{\tau}_d$) in the virtual control space;
- STEP 2 (*Actuator Selection Task*): Design control allocator, which finds the “best” feasible true control input \mathbf{u} (normalised command vector to be applied to individual actuators) in the true control space.

In the GNC Signal Flow shown in Figure 1, the *Motion Control System* and the *Control Allocation* components execute STEPS 1 and 2, respectively. After actuation with the control vector \mathbf{u} , the *Propulsion System* block will generate a total control effect (vector $\boldsymbol{\tau}$ in Figure 1). After normalisation, this vector should be equal to the desired virtual control input $\boldsymbol{\tau}_d$ in the case of successful control allocation.

Decomposition of the virtual control input $\boldsymbol{\tau}_d$ into the horizontal and vertical planes is presented in Table 1.

Table 1. Decomposition into horizontal and vertical planes.

Virtual Control Input	Horizontal Plane τ_{-HT_d}	Vertical Plane τ_{-VT_d}
$\tau_d = \begin{bmatrix} \tau_{X_d} \\ \tau_{Y_d} \\ \tau_{Z_d} \\ \tau_{K_d} \\ \tau_{M_d} \\ \tau_{N_d} \end{bmatrix}$	$\tau_{HT_d} = \begin{bmatrix} \tau_{X_d} \\ \tau_{Y_d} \\ \tau_{N_d} \end{bmatrix}$ τ_{X_d} —Surge Force τ_{Y_d} —Sway Force τ_{N_d} —Yaw Moment	$\tau_{VT_d} = \begin{bmatrix} \tau_{Z_d} \\ \tau_{K_d} \\ \tau_{M_d} \end{bmatrix}$ τ_{Z_d} —Heave Force τ_{K_d} —Roll Moment τ_{M_d} —Pitch Moment

The main focus of the discussion in the following will be based on the control allocation problem in the horizontal plane. However, a similar approach applies to the vertical plane with minor modifications.

The open-frame ROV with four horizontal thrusters in *X-shaped (vectored)* configuration and three controllable DOF is shown in Figure 2a, while the top view of thruster configuration in the horizontal plane is shown in Figure 2b. This configuration is the most dominant configuration of horizontal thrusters for commercial ROVs. However, the approach can be easily extended to other configurations.

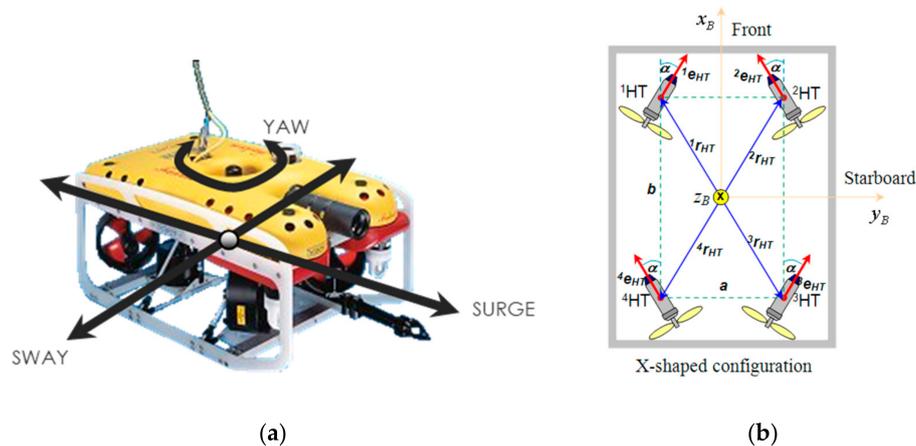


Figure 2. ROV with 4 HT (*X-shaped* configuration): (a) Controllable DoF; (b) Thruster configuration.

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).

For open-frame ROV, shown in Figure 2, dimensions of virtual and true control spaces are $k = 3$ and $m = 4$, respectively. For full geometric insight into underlying control allocation problem, the main difficulty is the inability to easily visualise what is happening in 4D true control space during the execution of control allocation solution steps.

To overcome this issue and to provide a clear picture into the geometry of the control allocation solution steps, an artificial “virtual” ROV is introduced (Figure 3), with very specific star configuration of the horizontal thrusters, such that the dimensions of control spaces are reduced by one, i.e., dimensions of virtual and true control spaces are $k = 2$ and $m = 3$, respectively. Introduction of the “virtual” ROV provides a framework to visualise concepts and all steps of the solution using 2D/3D figures, including the attainable command set, the shape of control energy spheres, solution lines, pseudoinverse solution, and cases for triggering the fixed-point iterations. The most important feature of the star configuration is the inability of thrusters to create a moment and rotate “virtual” ROV since

all thruster axes intersect in a common point, as indicated in Figure 3. Hence, two controllable DoFs are Surge and Sway, while Yaw DoF is not controllable.

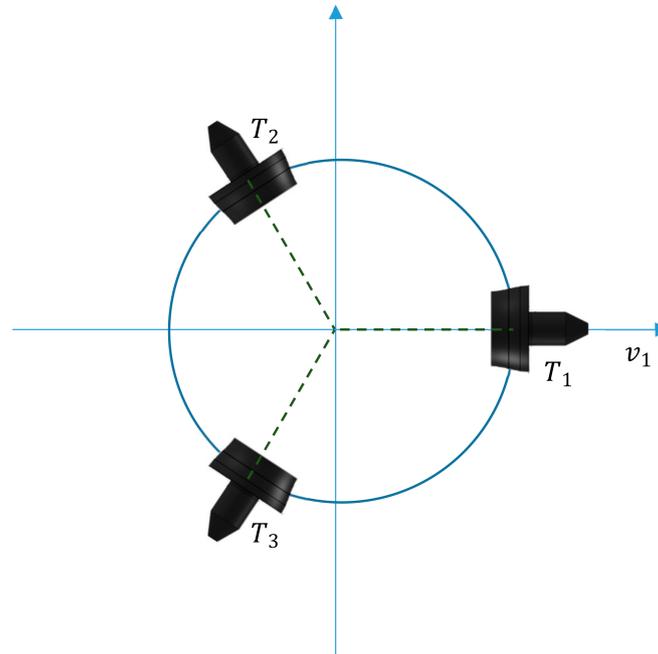


Figure 3. “Virtual” ROV with 3 HT (star thruster configuration).

To simplify notation, the *underline* and subscript *d* will be removed from normalised variables in the following text. When vectors/variables are not normalised, it will be stated explicitly. Based on the normalisation method described in [7], the general constrained problem for the “virtual” ROV, and the open-frame ROV can be formulated using normalised variables as indicated in Table 2.

Table 2. Control Allocation Problem for “virtual” ROV and open-frame ROV.

	Virtual ROV	Open-Frame ROV
Virtual Control Input	$v = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \in \mathbb{R}^2 (k = 2)$	$\tau = \begin{bmatrix} \tau_X \\ \tau_Y \\ \tau_N \end{bmatrix} \in \mathbb{R}^3 (k = 3)$
True Control Input	$u = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \in \mathbb{R}^3 (m = 3)$	$u = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} \in \mathbb{R}^4 (m = 4)$
Control Effectiveness Matrix	$B = \begin{bmatrix} \frac{1}{2} & -\frac{1}{4} & -\frac{1}{4} \\ 0 & \frac{3}{5} & -\frac{2}{5} \end{bmatrix}$	$B = \begin{bmatrix} \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & -\frac{1}{4} & \frac{1}{4} & -\frac{1}{4} \\ \frac{1}{4} & -\frac{1}{4} & -\frac{1}{4} & \frac{1}{4} \end{bmatrix}$
Actuator Position Constraints	$\underline{u} = \begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix}, \bar{u} = \begin{bmatrix} +1 \\ +1 \\ +1 \end{bmatrix}$	$\underline{u} = \begin{bmatrix} -1 \\ -1 \\ -1 \\ -1 \end{bmatrix}, \bar{u} = \begin{bmatrix} +1 \\ +1 \\ +1 \\ +1 \end{bmatrix}$
	For a given v , find u such that	For a given τ , find u such that
Control Allocation Problem (Inequalities (2) and (4) apply component-wise.)	$Bu = v$ subject to constraint $\underline{u} \leq u \leq \bar{u}$	$Bu = \tau$ subject to constraint $\underline{u} \leq u \leq \bar{u}$

$$Bu = v \tag{1}$$

$$\underline{u} \leq u \leq \bar{u} \tag{2}$$

$$Bu = \tau \tag{3}$$

$$\underline{u} \leq u \leq \bar{u} \tag{4}$$

In general case, the equation $Bu = \tau$ defines the set of hyperplanes in the true control space \mathbb{R}^m . The intersection of these hyperplanes is a convex set, denoted by \mathfrak{R} . The set of inequalities $\underline{u} \leq u \leq \bar{u}$ represents the hyper box in the same space. This hyper box is called constrained (admissible) control subset and denoted by Ω . The solution set \mathfrak{S} is given by the intersection of \mathfrak{R} and Ω . Three cases are possible:

- \mathfrak{S} is empty (i.e., no solution exists),
- \mathfrak{S} has exactly one element (i.e., there is one unique solution),
- \mathfrak{S} has more than one element (i.e., there are many solutions).

The following discussion is limited to Control Allocation Problem for the “virtual” ROV. Equation (1) represents a system of equations

$$\begin{aligned} \frac{1}{2}u_1 - \frac{1}{4}u_2 - \frac{1}{4}u_3 &= v_1 \\ 0u_1 + \frac{3}{5}u_2 - \frac{2}{5}u_3 &= v_2 \end{aligned} \tag{5}$$

Each Equation in (5) represents a plane in \mathbb{R}^3 . Consequently, (5) can be rewritten as

$$\begin{aligned} \pi_1 : N_1^T \cdot u &= v_1 \\ \pi_2 : N_2^T \cdot u &= v_2 \end{aligned} \tag{6}$$

where $N_1^T = \left[\frac{1}{2} \quad -\frac{1}{4} \quad -\frac{1}{4} \right]$ and $N_2^T = \left[0 \quad \frac{3}{5} \quad -\frac{2}{5} \right]$ are normal vectors, orthogonal on planes π_1 and π_2 , respectively. Since the determinant $\Delta = 0.1925 \neq 0$, these planes are not parallel, and their intersection is a line ℓ :

$$\ell : p = \begin{bmatrix} \frac{104}{77}v_1 + \frac{10}{77}v_2 + \frac{1}{4}t \\ -\frac{40}{77}v_1 + \frac{85}{77}v_2 + \frac{1}{5}t \\ -\frac{60}{77}v_1 - \frac{65}{77}v_2 + \frac{3}{10}t \end{bmatrix} \tag{7}$$

where t is the parameter of the line. The line ℓ is a convex set, denoted by \mathfrak{R} in the previous discussion. The constrained control subset Ω , which satisfies actuator position constraints (2), is a unit cube in \mathbb{R}^3 :

$$\Omega = \{u \in \mathbb{R}^3 : \|u\|_\infty \leq 1\} \tag{8}$$

Geometric interpretation of the control allocation problem can be obtained by reformulating the problem as follows:

$$\text{For a given } v, \text{ find the intersection } \mathfrak{S} \text{ of } \ell \text{ (7) and } \Omega \text{ (8)} \tag{9}$$

Three cases are possible:

- If the intersection is a segment, there is an infinite number of solutions (each point that belongs to the segment is a solution),
- If the intersection is a point, there is only one solution,
- If the intersection is an empty set, no solution exists.

The output of the Control Allocation subsystem in Figure 1 must be a single vector for all three cases. In the first case above, a secondary criterion (like minimum control energy) is introduced to

select a unique solution. In the third case above, the “best” approximation is used as the output of the algorithm. All three cases are discussed in the following in more details.

2.2. Nomenclature

The following nomenclature is adopted for referring to Ω , shown in Figure 4 [20]:

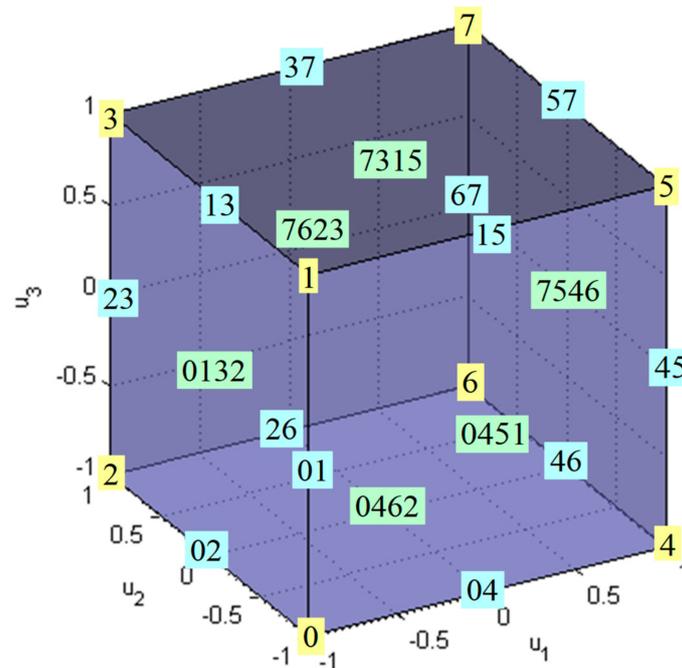


Figure 4. The constrained control subset $\Omega \subset \mathbb{R}^3$ for “virtual” ROV.

The boundary of Ω is denoted by $\partial(\Omega)$. A true control vector belongs to $\partial(\Omega)$ if and only if at least one of its components is at a limit (-1 or $+1$). Vertices are the points on $\partial(\Omega)$ where each component receives a limit (min or max). In Figure 4, vertices are denoted as $0, 1, \dots, 7$. In the general case, the number of vertices is equal to 2^m . Vertices are numerated using the following rule: if the vertex is represented in a binary form, then “0” in the k^{th} position of this representation indicates that the corresponding component u_k is at a minimum $\bar{u}_k = -1$, while “1” indicates it is at a maximum $\bar{u}_k = +1$. For example, binary representation for the vertex 1 is 001. Using the rule it can be decoded as $\underline{u}_1 \underline{u}_2 \bar{u}_3$, which refers to the vertex generated by $u_1 = -1, u_2 = -1, u_3 = +1$. Edges are lines that connect vertices and that lie on $\partial(\Omega)$. In Figure 4 edges are denoted as 01, 02, \dots , 67. They are generated by varying only one of the m components, while the remaining $m - 1$ are at their limits, associated with the two connected vertices. In the general case, the number of edges is equal to $2^{m-1} \binom{m}{1}$. Two vertices are connected by an edge if and only if their binary representations differ in only one bit. For example, vertices 0 and 1 are connected by an edge 01, because their binary representations (000 for 0, 001 for 1) differ in only one (last) bit. In contrast, vertices 0 and 3 are not connected by an edge, because their binary representations (000 for 0, 011 for 3) differ in more than one bit.

Facets are plane surfaces on $\partial(\Omega)$ that contain two adjacent edges, i.e., two edges that have a common vertex. In Figure 4, facets are denoted as 0132, 0451, \dots , 7623. In the general case, the number of facets is equal to $2^{m-2} \binom{m}{2}$. A facet is defined as the set in the true control space obtained by taking all but two components at their limits and varying the two free components within their limits. For example, for facet 0132 binary representations of its vertices are 000 for 0, 001 for 1, 011 for 3 and 010 for 2. It can be seen that the first digit in these representations is fixed, while the other digits are not

fixed, indicating that on this facet the first component is fixed at the limit $u_1 = -1$, while the other two components u_2 and u_3 are free to vary between their limits $[-1, +1]$.

In the virtual control space \mathbb{R}^2 the allowable virtual control subset Φ (Figure 5) is defined as a set of all allowable virtual control inputs (unit square in \mathbb{R}^2):

$$\Phi = \{v \in \mathbb{R}^2 : \|v\|_\infty \leq 1\} \tag{10}$$

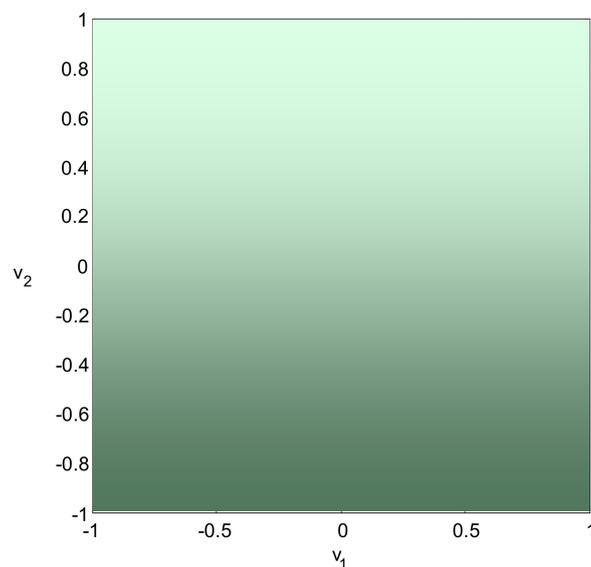


Figure 5. The allowable virtual control subset $\Phi \subset \mathbb{R}^2$ for “virtual” ROV.

2.3. Geometric Insight into Problem

Equation (1) defines a linear transformation B from the true control space \mathbb{R}^3 to the virtual control space \mathbb{R}^2 , which maps (projects) the constrained control subset (unit 3D cube) Ω into attainable command set $\Phi_v \subset \Phi$ (Figure 6).

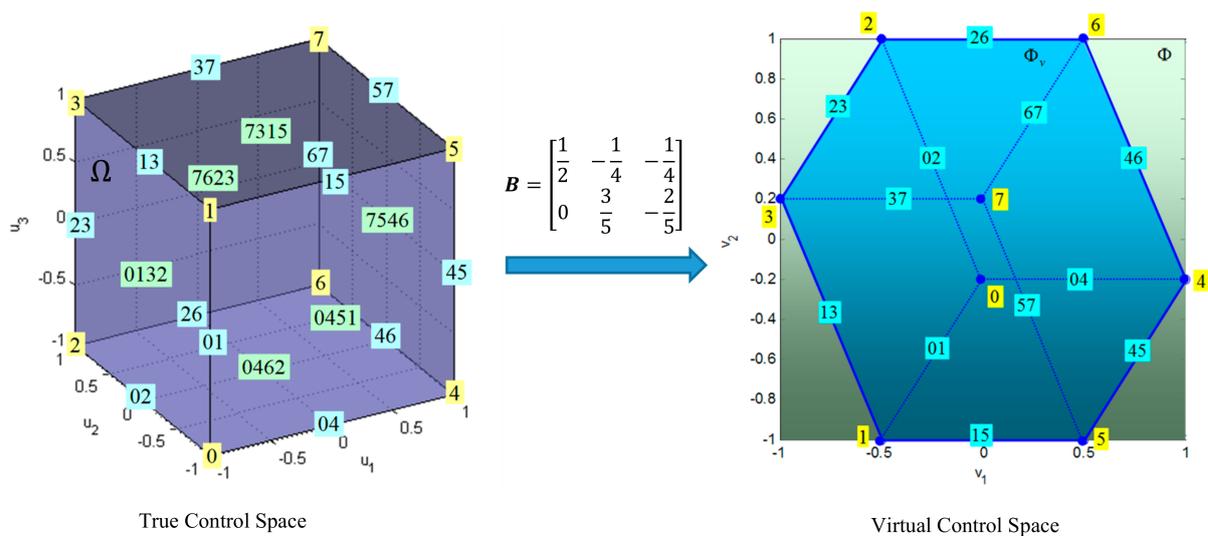


Figure 6. The control effectiveness matrix B maps (projects) the constrained control subset Ω into attainable command set $\Phi_v \subset \Phi$.

In the general case, the optimal true control input is given by the solution to a two-step optimisation problem [9]:

$$u = \arg \min \|W_u(u - u_p)\|_p, u \in \Psi \tag{11}$$

$$\Psi = \arg \min \|W_v(Bu - v)\|_p, u \in \Omega \tag{12}$$

where u_p represents preferred positions of the actuator inputs (preferred true control input) and W_u and W_v are weighting matrices. The problem defined with (11)–(12) can be interpreted as follows: given Ψ , the set of feasible control inputs that minimise $Bu - v$ (weighted by W_v), find the true control input u that minimises $u - u_p$ (weighted by W_u). In Equations (11) and (12) u_p , W_u , and W_v are design parameters. The choice of u_p may correspond, for example, to minimum control deflections in aerospace applications. The non-actuated state of the horizontal thrusters is a preferred state for an ROV, i.e., preferred true control input in (11) is $u_p = 0$. Matrix W_u can be used for actuator prioritisation, i.e., which actuator should be used primarily. Similarly, matrix W_v allows for prioritisation among the virtual control inputs when the problem (11)–(12) has no exact solution. Geometric interpretation of (11)–(12) for “virtual” ROV is given in the following example.

Example 1. Consider the control allocation problem (1)–(2) and let $v = [-0.5 \ 0.6]^T$ is the desired virtual control input (Figure 7). Let the optimisation objective be given by (11)–(12), with $u_p = 0$, $W_u = I_3$, $W_v = I_2$.

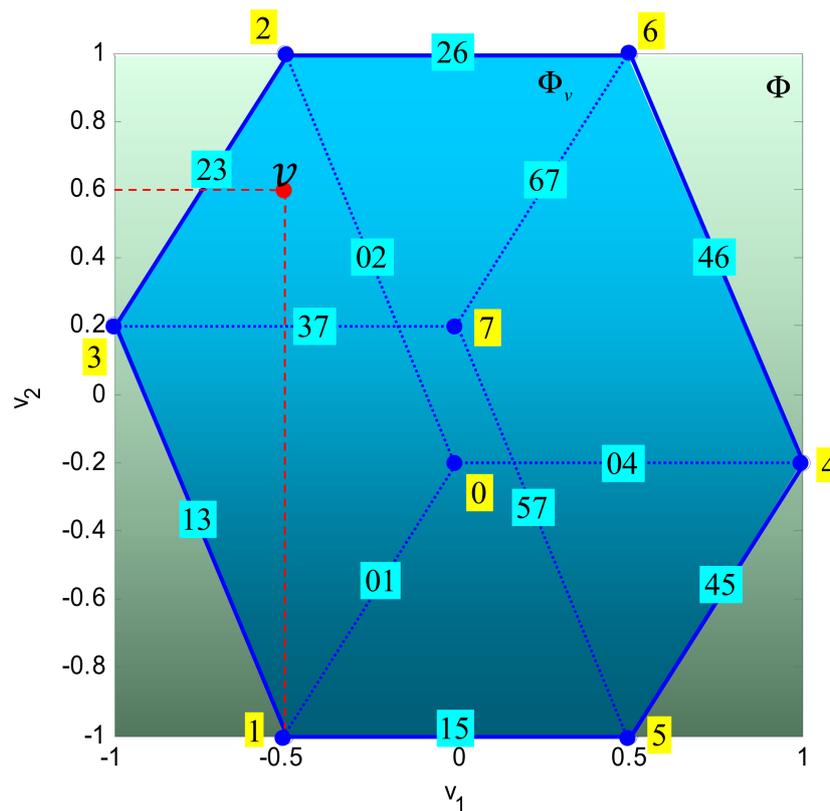


Figure 7. Position of the desired virtual control input v in Φ_v .

Since $v \in \Phi_v$, i.e., v is attainable, the problem (11)–(12) can be reduced to

$$\min_u \|u\|_p \tag{13}$$

subject to

$$B\mathbf{u} = \begin{bmatrix} -0.5 \\ 0.6 \end{bmatrix} \tag{14}$$

$$\underline{\mathbf{u}} = \begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix} \leq \mathbf{u} \leq \begin{bmatrix} +1 \\ +1 \\ +1 \end{bmatrix} \tag{15}$$

Substituting $v_1 = -0.5$ and $v_2 = 0.6$ in (7) yields the following solution line

$$\ell: \mathbf{p} = \begin{bmatrix} -\frac{46}{77} + \frac{1}{4}t \\ \frac{71}{77} + \frac{1}{5}t \\ -\frac{9}{77} + \frac{3}{10}t \end{bmatrix} \tag{16}$$

The intersection (solution set) \mathfrak{S} of ℓ (16) and Ω (8) is a segment $\mathfrak{S} = P_1P_2$, where $P_1 = \begin{bmatrix} -1 & \frac{3}{5} & -\frac{3}{5} \end{bmatrix}^T$ (for $t = t_1 = -123/77$) and $P_2 = \begin{bmatrix} -\frac{1}{2} & 1 & 0 \end{bmatrix}^T$ (for $t = t_2 = 30/77$) (see Figure 8). The points P_1 and P_2 belong to facets 0132 and 7623, respectively. The solution of the problem is a point on the segment P_1P_2 that minimises $\|\mathbf{u}\|_p$. Hence, the solution depends on the choice of the norm.

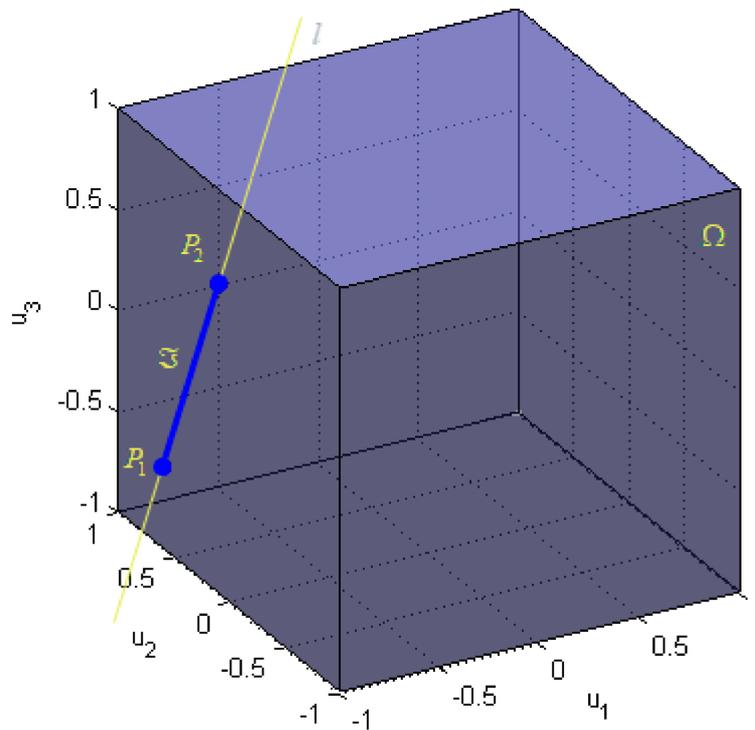


Figure 8. Solution line ℓ and solution set $\mathfrak{S} = \ell \cap \Omega = P_1P_2$.

2.4. Choice of Norm

It is useful to analyse the relationship between the choice of the norm l_p , the weighting matrix 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 $W_u(\mathbf{u} - \mathbf{u}_p)_p \leq r$ is called a weighted sphere $S_{W_u}(\mathbf{u}_p, r)_p$. A family of spheres can be obtained by varying radius r . The optimal solution \mathbf{u}^* is a point where a family of these spheres, centred at \mathbf{u}_p and weighted by W_u , “touch” the set $\mathfrak{S} = P_1P_2$.

Figure 9 illustrates the situation for l_1 norm. The family of spheres is represented by concentric diamond-shaped bodies. If the radius of spheres is increased, for certain value r_1 the sphere $S_{I_3}(\mathbf{0}, r_1)_1$ will touch the segment P_1P_2 . The touching point is a solution, which minimises $\|\mathbf{u}\|_1$ and $\min_{\mathbf{u}} \|\mathbf{u}\|_1 = r_1$.

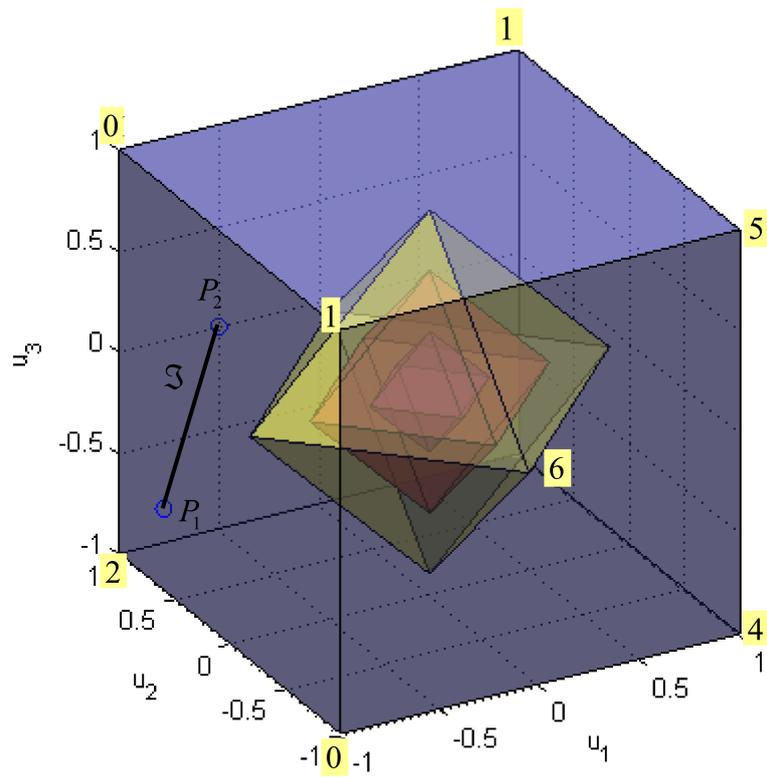


Figure 9. Family of spheres $S_{I_3}(\mathbf{0}, r)_1$ for l_1 norm.

Case $p = 2$ is illustrated in Figure 10. Now the shape of the sphere is familiar from the Euclidian metric. The same procedure, as described above, can be applied again: if the radius of spheres is increased, for certain value r_2 the sphere $S_{I_3}(\mathbf{0}, r_2)_2$ will touch the segment P_1P_2 . The touching point is a solution, which minimise $\|u\|_2$ and $\min_u \|u\|_2 = r_2$.

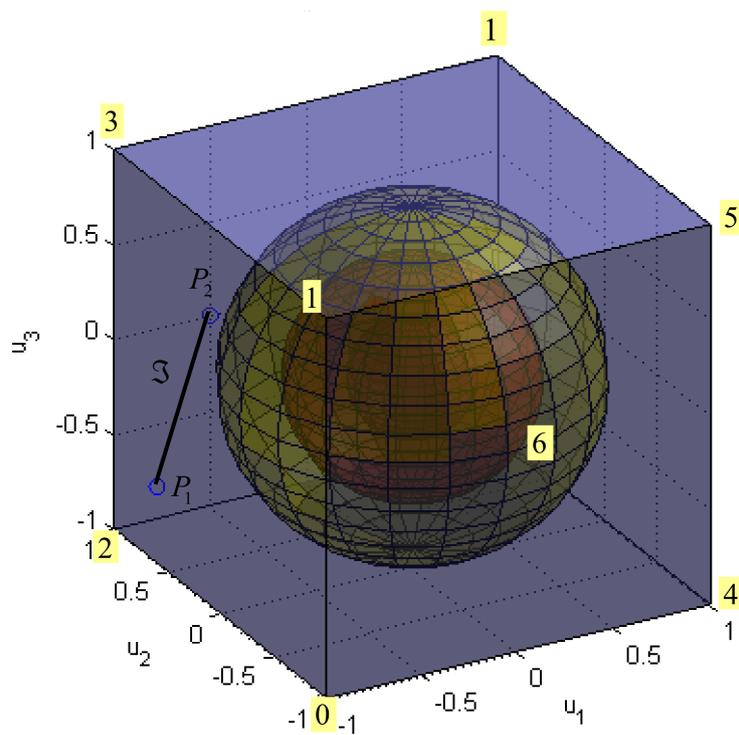


Figure 10. Family of spheres $S_{I_3}(\mathbf{0}, r)_2$ for l_2 norm.

Remarks 1.

- In the case when W_u is the unity matrix, the l_2 norm distributes the virtual control demand among the control inputs in a uniform way, while the l_1 solution utilises as few control inputs as possible to satisfy the virtual control demand.
- The l_2 solution varies continuously with the parameters (elements) of B , while the l_1 solution does not. Change in a parameter (element) b of B will produce the change in the slope of ℓ . The l_2 solution will vary continuously with b , while it can be shown that the l_1 solution will have discontinuity for some value of $b = b^*$ and the solution in the breakpoint b^* is not unique.
- If W_u is a non-singular, the problem $\min_u \|W_u u\|_p$ has a unique solution for $p = 2$. For $p = 1$, this is not always the case, as discussed above. The reason lies in the fact that the sphere $S_{W_u}(\mathbf{0}, r)_2$ is a strictly convex set, while this is not the case for $S_{W_u}(\mathbf{0}, r)_1$.

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 (11) can be interpreted as control energy cost function.

2.5. Choice of the Weighting Matrix W_u

The matrix W_u is a design parameter typically used for actuator prioritisation. If all actuators have the same priority, then W_u is equal to the 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 W_u . For example, if $W_u = \text{diag}(w_1, w_2, w_3) = \text{diag}(1, 1, 1 + \Delta w_3)$, then the faulty horizontal thruster HT₃ is penalised for $\Delta w_3 > 0$ and its contribution to the total control effort is less than in the fault-free case ($\Delta w_3 = 0$). Consider the same problem as before, but this time $\Delta w_3 = 1$, i.e., $W_u = \text{diag}(1, 1, 2)$. Since $w_3 = 2 > w_1 = w_2 = 1$, the third actuator has a lower priority than the other two. Geometrically, weighted spheres in the true control space are compressed along the third control axis and become “flattened” spheres, i.e., ellipsoids, as indicated in Figure 11.

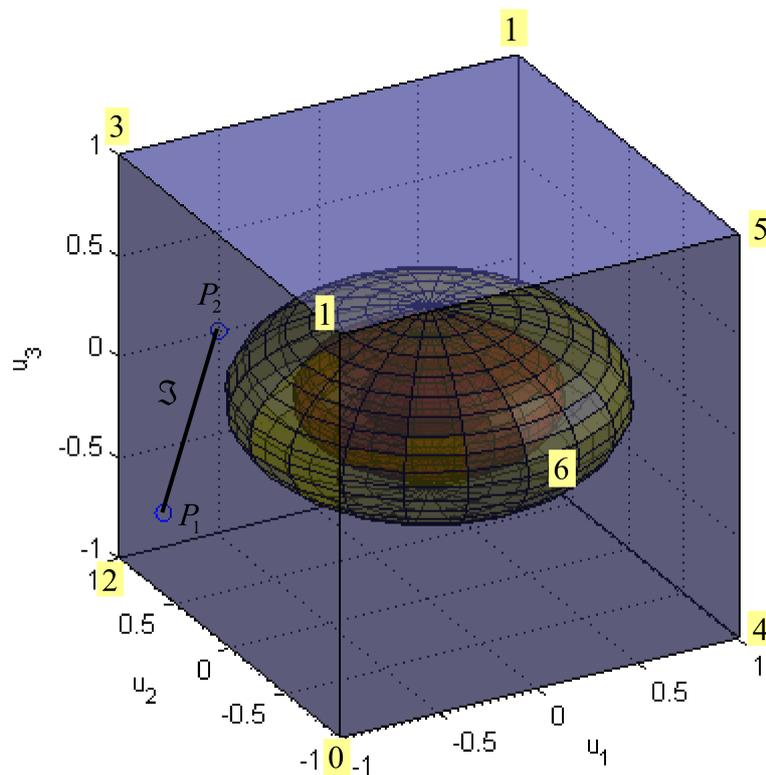


Figure 11. Family of spheres $S_{I_3}(\mathbf{0}, r)_2$ for l_2 norm and case $W_u = \text{diag}(1, 1, 2)$.

In this way, the solution (touching point of the family of ellipsoids with the segment P_1P_2) exhibits a lower contribution of the third control and higher participation of the other two controls, compared to the previous case, when all actuators had the same priority. The additional introduction of nonzero off-diagonal elements into the symmetric matrix W_u yields to the rotation of these ellipsoids around the centre.

3. Control Allocation Solution: Hybrid Method

3.1. Description

The hybrid method for control allocation is based on the integration of the weighted 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 method is activated in the second step, which results in a guaranteed feasible solution. In this way, the hybrid method 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. Weighted Pseudoinverse

3.2.1. Introduction

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

$$\min \|W_u u\|_2 \tag{17}$$

subject to $Bu = v$ is given by

$$u = B_{W_u}^\dagger v \tag{18}$$

where the matrix $B_{W_u}^\dagger$

$$B_{W_u}^\dagger = W_u^{-1}(BW_u^{-1})^\dagger = W_u^{-1}B^T(BW_u^{-1}B^T)^{-1} \tag{19}$$

is the weighted pseudoinverse of B [20]. For “virtual” ROV the matrix $B_{W_u}^\dagger$ is given by

$$B_{W_u}^\dagger = \frac{1}{25w_1 + 16w_2 + 36w_3} \begin{bmatrix} 8(4w_2 + 9w_3) & -10(2w_2 - 3w_3) \\ -40w_1 & 5(5w_1 + 12w_3) \\ -60w_1 & -5(5w_1 + 8w_2) \end{bmatrix} \tag{20}$$

where

$$W_u = \begin{bmatrix} w_1 & 0 & 0 \\ 0 & w_2 & 0 \\ 0 & 0 & w_3 \end{bmatrix} \tag{21}$$

For case $W_u = I_3$ the weighted pseudoinverse (20) is given by (the change of weights in W_u produces the change in slope of the parallelogram Ω_v and, consequently, the change in shape of Ω_p)

$$B_{I_3}^\dagger = \begin{bmatrix} b_{11}^\dagger & b_{12}^\dagger \\ b_{21}^\dagger & b_{22}^\dagger \\ b_{31}^\dagger & b_{32}^\dagger \end{bmatrix} = \begin{bmatrix} 1.3506 & 0.1299 \\ -0.5195 & 1.1039 \\ -0.7792 & -0.8442 \end{bmatrix} \tag{22}$$

The weighted pseudoinverse (18) is mapping from $k = 2$ -dimensional virtual control space to $m = 3$ -dimensional true control space (Figure 12). The allowable virtual control subset (unit square $\Phi = V_0V_1V_3V_2$) is mapped by the weighted pseudoinverse to a parallelogram $\Omega_v = U_0U_1U_3U_2$.

intersection of the parallelogram Ω_v with the constrained control subset Ω is a convex polygon $\Omega_p = R_{13}R_{15}R_{45}R_{46}R_{26}R_{23}$, where the vertex R_{ij} lies on the edge ij of Ω .

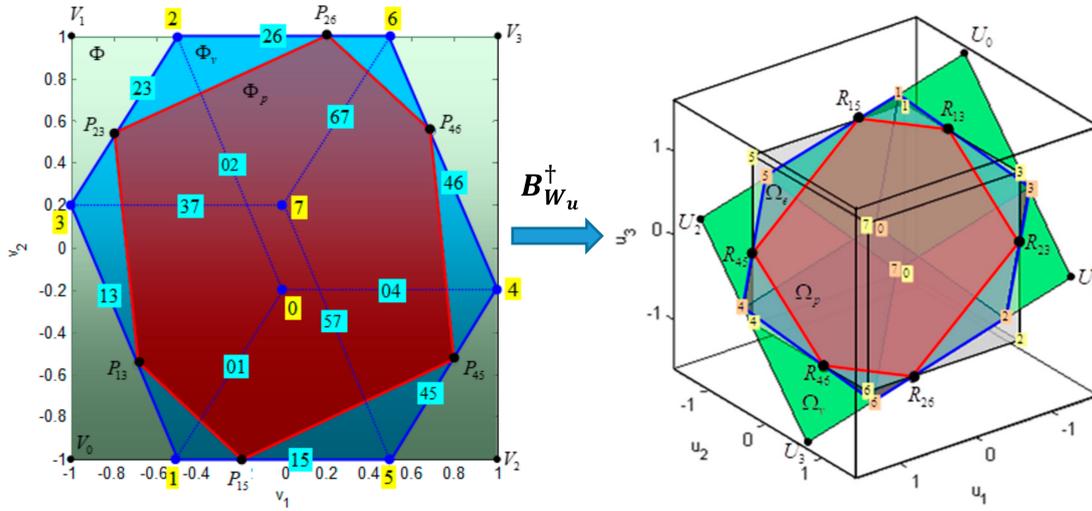


Figure 12. The Weighted Pseudoinverse as a mapping from Virtual Control Space to True Control Space. Partitions of the Virtual Control Space are: allowable virtual control subset Φ , attainable command set Φ_v and feasible region for the weighted pseudoinverse Φ_p . Images of these partitions in the true control space are: $\Omega_v = B_{W_u}^\dagger(\Phi)$, $\Omega_e = B_{W_u}^\dagger(\Phi_v)$ and $\Omega_p = B_{W_u}^\dagger(\Phi_p)$.

The following discussion will address these issues:

- Find $\Phi_p \subset \Phi_v$ such that $B_{W_u}^\dagger(\Phi_p) = \Omega_p$,
- Find $B_{W_u}^\dagger(\Phi_v)$.

To find Φ_p it is sufficient to find points P_{ij} such that

$$R_{ij} = B_{W_u}^\dagger(P_{ij}) \tag{23}$$

Let $P_{ij} = [v_1^{ij} \ v_2^{ij}]^T$ and $R_{ij} = [u_1^{ij} \ u_2^{ij} \ u_3^{ij}]^T$. Then (23) can be rewritten as

$$\begin{aligned} b_{11}^\dagger v_1^{ij} + b_{12}^\dagger v_2^{ij} &= u_1^{ij} \\ b_{21}^\dagger v_1^{ij} + b_{22}^\dagger v_2^{ij} &= u_2^{ij} \\ b_{31}^\dagger v_1^{ij} + b_{32}^\dagger v_2^{ij} &= u_3^{ij} \end{aligned} \tag{24}$$

The fact that $R_{ij} \in ij$ of Ω means that R_{ij} is bounded to the edge ij defined by vertices i and j , i.e., two coordinates (controls) of R_{ij} are fixed to their limits, while one is free to vary. Recall that the nomenclature for edges, introduced in Section 2.2, enables easy detection of free and fixed controls for ij : binary representations of vertices i and j differ in only one bit and the position of this bit indicates a free control for ij . For example, the edge 13 is determined by vertices 1 and 3. The binary representations of 1 (001) and 3 (011) differ in the second bit, which means that free control is u_2^{13} , while fixed controls are $u_1^{13} = \underline{u}_1 = -1$ and $u_3^{13} = \bar{u}_3 = +1$. Once the fixed and free controls for ij are obtained, it is easy to find coordinates v_1^{ij} and v_2^{ij} of P_{ij} from (24) by removing the equation that corresponds to free control and replacing the right-hand sides of other equations with corresponding limits for fixed controls. For example, for R_{13} system (24) can be rewritten as

$$\begin{aligned} b_{11}^\dagger v_1^{13} + b_{12}^\dagger v_2^{13} &= u_1^{13} \\ b_{21}^\dagger v_1^{13} + b_{22}^\dagger v_2^{13} &= u_2^{13} \\ b_{31}^\dagger v_1^{13} + b_{32}^\dagger v_2^{13} &= u_3^{13} \end{aligned} \tag{25}$$

Free control is u_2^{13} so the second equation is removed, while the right-hand sides of the first and the third equation are replaced with $u_1^{13} = \underline{u}_1 = -1$ and $u_3^{13} = \bar{u}_3 = +1$, respectively:

$$\begin{aligned} b_{11}^\dagger v_1^{13} + b_{12}^\dagger v_2^{13} &= -1 \\ b_{31}^\dagger v_1^{13} + b_{32}^\dagger v_2^{13} &= +1 \end{aligned} \tag{26}$$

which yields the solution $v_1^{13} = -0.6875$ and $v_2^{13} = -0.5500$. Other vertices can be found similarly, and results are shown in Table 3 (vertices P_{ij}) and Table 4 (vertices R_{ij}). Hence, the subset $\Phi_p \subset \Phi_v$ such that $B_{W_u}^\dagger(\Phi_p) = \Omega_p$ is a convex polygon $P_{13}P_{15}P_{45}P_{46}P_{26}P_{23}$, whose vertex P_{ij} lies on the edge ij of Φ_v (Figure 12).

Table 3. Vertices of Φ_p .

	P_{23}	P_{45}	P_{13}	P_{46}	P_{15}	P_{26}
v_1	-0.7917	0.7917	-0.6875	0.6875	-0.2000	0.2000
v_2	0.5333	-0.5333	-0.5500	0.5500	-1.0000	1.0000

Table 4. Vertices of $\Omega_p = B_{W_u}^\dagger(\Phi_p)$.

	R_{23}	R_{45}	R_{13}	R_{46}	R_{15}	R_{26}
u_1	-1.0000	1.0000	-1.0000	1.0000	-0.4000	0.4000
u_2	1.0000	-1.0000	-0.2500	0.2500	-1.0000	1.0000
u_3	0.1667	-0.1667	1.0000	-1.0000	1.0000	-1.0000

The pseudoinverse image of Φ_v is a convex polygon $\Omega_e = B_{W_u}^\dagger(\Phi_v)$. Vertices 1, 2, 3, 4, 5 and 6 of Φ_v are mapped to vertices 1, 2, 3, 4, 5, 6 of Ω_e that lie outside Ω , while nodes 0 and 7 of Φ_v are mapped to nodes 0 and 7 of Ω_e that lie inside Ω . The virtual control space can be partitioned into three characteristic regions (Figure 12): feasible region for the pseudoinverse Φ_p (polygon $P_{13}P_{15}P_{45}P_{46}P_{26}P_{23}$), $\Phi_v \setminus \Phi_p$ (union of triangles $1P_{15}P_{13}, 3P_{13}P_{23}, \dots, 5P_{45}P_{15}$) and $\Phi \setminus \Phi_v$ (union of triangles $V_0 1 3, \dots, V_2 4 5$). The weighted pseudoinverse image of each partition lies inside the parallelogram Ω_v (Figure 12). Table 5 displays cross-relation between these polygons in the virtual and the true control space.

Table 5. Cross-relation between the virtual and the true control space.

Virtual Control Space		True Control Space	
Partition	Polygon	Partition	Polygon
Φ_p	$P_{13}P_{15}P_{45}P_{46}P_{26}P_{23}$	Ω_p	$R_{13}R_{15}R_{45}R_{46}R_{26}R_{23}$
$\Phi_v \setminus \Phi_p$	1 $P_{15}P_{13}$	$\Omega_e \setminus \Omega_p$	1 $R_{15}R_{13}$
	3 $P_{13}P_{23}$		3 $R_{13}R_{23}$
	2 $P_{23}P_{26}$		2 $R_{23}R_{26}$
	6 $P_{26}P_{46}$		6 $R_{26}R_{46}$
	4 $P_{46}P_{45}$		4 $R_{46}R_{45}$
$\Phi \setminus \Phi_v$	5 $P_{45}P_{15}$	$\Omega_v \setminus \Omega_e$	5 $R_{45}R_{15}$
	$V_0 1 3$		$U_0 1 3$
	$V_1 3 2$		$U_1 3 2$
	$V_3 6 4$		$U_3 6 4$
	$V_2 4 5$		$U_2 4 5$

Subset Φ_p represents a part of the allowable virtual control subset Φ attainable (feasible) by weighted pseudoinverse. That is, if $v \in \Phi_p$, then $u^* = B_{W_u}^\dagger(v) = \ell \cap \Omega_p$ and $\|u\|_2^* = \min_{Bu=v} \|u\|_2$ (see Figure 13). The sphere with the centre at $\mathbf{0}$ and radius $\|u\|_2^*$ is also shown in Figure 13. This sphere touches the solution line ℓ at u^* . In other words, if the virtual control input lies in Φ_p , then the weighted

pseudoinverse solution is the intersection of the solution line ℓ and parallelogram $\Omega_p = B_{W_u}^\dagger(\Phi_p)$, it is feasible and has a minimal l_2 norm compared to any other solution. Otherwise, if $v \notin \Phi_p$, then $u = B_{W_u}^\dagger(v) \notin \Omega$, which means that the virtual control input v is not feasible with weighted pseudoinverse, although some other choice of general inverse or application of other methods, like the fixed-point method, could find a feasible solution. This aspect of the relationship between the position of the virtual control input and the feasibility of the solution is important and discussed in the following example.

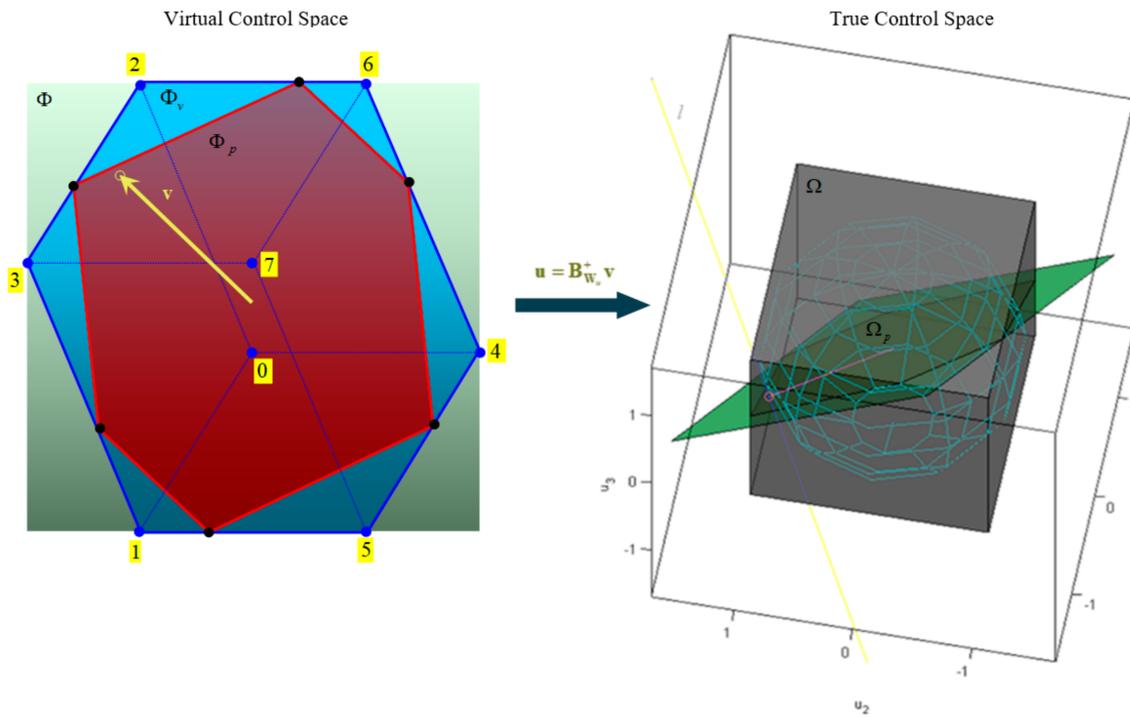


Figure 13. Geometric interpretation of pseudoinverse: for a given $v \in \Phi_p$, the weighted pseudoinverse solution is an intersection of the solution line and $\Omega_p = B_{W_u}^\dagger(\Phi_p)$ and has a minimal l_2 norm compared to any other solution.

Example 2. Consider the control allocation problem (1)–(2) and let $S = [v_1 \ v_2]^T$ denotes an arbitrary point from the allowable virtual control subset Φ . Recall that a total solution set \mathfrak{S} is given by the intersection of the line ℓ (7) and Ω . When a point S moves inside Φ , the corresponding line ℓ moves in the true control space. For a given S , the weighted pseudoinverse will select the solution from \mathfrak{S} where the line ℓ intersects the parallelogram Ω_v . Three characteristic cases are possible, regarding the position of S relative to the partitions of Φ (Figure 14):

1. $S = S_1 \in \Phi_p$,
2. $S = S_2 \in \Phi_v \setminus \Phi_p$,
3. $S = S_3 \in \Phi \setminus \Phi_v$.

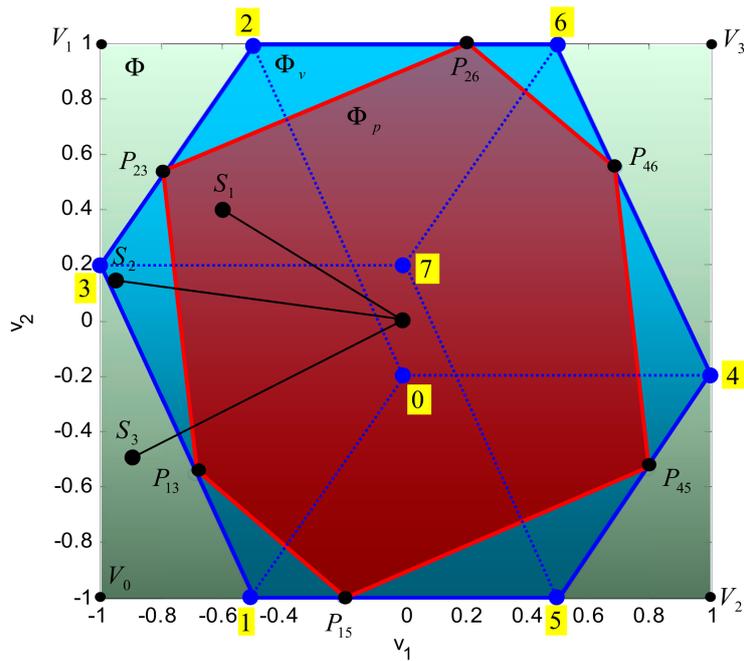


Figure 14. Three typical cases for the position of virtual control inputs relative to Φ_p and Φ_v .

Case 2.1: In this case, point S_1 lies inside Φ_p and the weighted pseudoinverse solution is $T_1 = B_{W_u}^+(S_1)$ that lies inside Ω_p (Figure 15). A solution set is a segment $\mathfrak{S}_1 = \ell_1 \cap \Omega$. This segment intersects the parallelogram Ω_v in the point $T_1 = \mathfrak{S}_1 \cap \Omega_v$. From all solutions in \mathfrak{S}_1 , the solution T_1 , selected by weighted pseudoinverse, is optimal in l_2 sense.

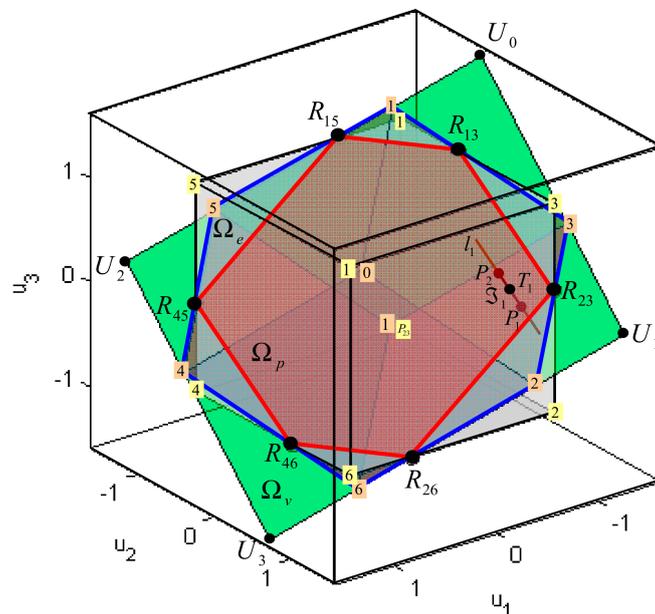


Figure 15. Case $S_1 \in \Phi_p$ yields to the pseudoinverse solution $T_1 \in \Omega_p \subset \Omega$ that is optimal in l_2 sense.

For example, if $S_1 = \begin{bmatrix} -0.6 & 0.4 \end{bmatrix}^T$, then the solution set is a segment $\mathfrak{S}_1 = P_1P_2 = \ell_1 \cap \Omega$, $P_1 = \begin{bmatrix} -1 & 14/25 & -4/25 \end{bmatrix}^T$, $P_2 = \begin{bmatrix} -9/20 & 1 & 1/2 \end{bmatrix}^T$ (Figure 15). The weighted pseudoinverse solution ($T_1 = B_{W_u}^+(S_1) = \begin{bmatrix} -0.7584 & 0.7532 & 0.1299 \end{bmatrix}^T$) represents the point where the segment P_1P_2 intersects the parallelogram Ω_v . This solution is feasible since it belongs to Ω .

Case 2.2: In this case, point S_2 lies outside Φ_p , but inside Φ_v . The image $T_2 = B_{W_u}^\dagger(S_2)$ lies outside Ω_p , but inside Ω_e (Figure 16). Geometrically, a solution set is the segment $\mathfrak{S}_2 = \ell_2 \cap \Omega$ that does not intersect with Ω_v , which means that the weighted pseudoinverse solution T_2 lies on $\mathfrak{S}_2 = \ell_2$ but outside \mathfrak{S}_2 , i.e., $T_2 \notin \Omega$. Hence, the virtual control input S_2 is unfeasible (unattainable) by weighted pseudoinverse, but, because \mathfrak{S}_2 is not empty, some other methods (like fixed-point method) can allocate solution from \mathfrak{S}_2 , optimal in l_2 sense.

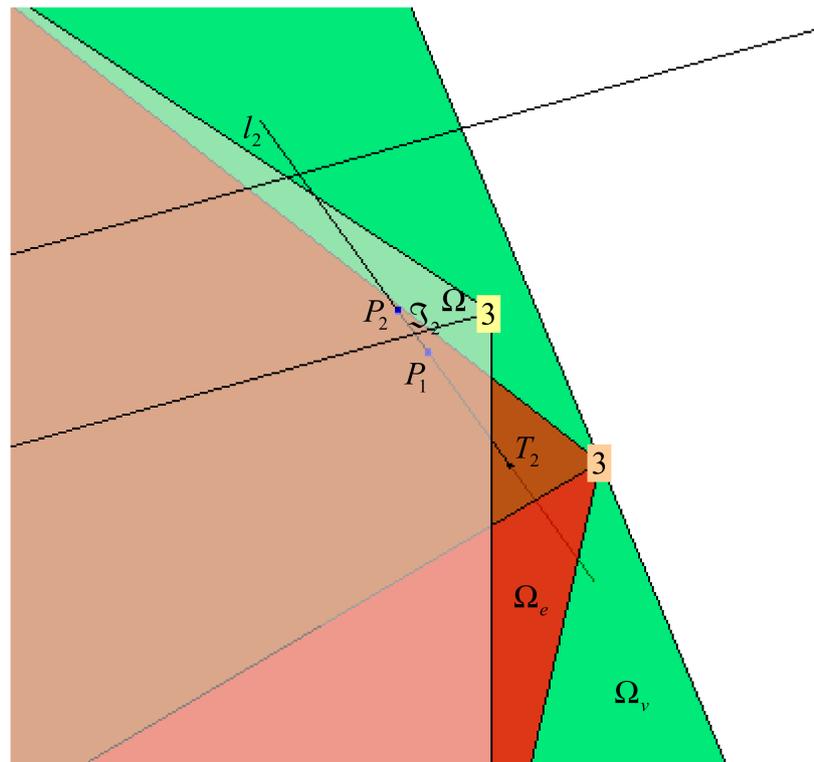


Figure 16. Case $S_2 \in \Phi_v \setminus \Phi_p$ yields to the unfeasible weighted pseudoinverse solution $T_2 \in \Omega_e \setminus \Omega_p$ that lies outside Ω .

For example, if $S_2 = \begin{bmatrix} -0.9375 & 0.1600 \end{bmatrix}^T$, then the solution set is a segment $\mathfrak{S}_2 = P_1P_2 = \ell_2 \cap \Omega$, $P_1 = \begin{bmatrix} -1 & 43/50 & 89/100 \end{bmatrix}^T$, $P_2 = \begin{bmatrix} -9/20 & 1 & 1/2 \end{bmatrix}^T$ (Figure 16). The weighted pseudoinverse solution ($T_2 = B_{W_u}^\dagger(S_2) = \begin{bmatrix} -1.2455 & 0.6636 & 0.5955 \end{bmatrix}^T$) represents the point where the line ℓ_2 intersects the parallelogram Ω_v . This solution is unfeasible since it lies on the line ℓ_2 outside \mathfrak{S}_2 , and does not belong to Ω .

Case 2.3: Finally, in the last case point S_3 lies outside Φ_v , but inside Φ (Figure 17). The image $T_3 = B_{W_u}^\dagger(S_3)$ lies outside Ω_e , but inside Ω_v (Figure 17). In this case line ℓ_3 does not intersect with Ω , i.e., \mathfrak{S}_3 is an empty set, and the exact solution of the problem does not exist. The weighted pseudoinverse solution T_3 is unfeasible, since it does not belong to Ω .

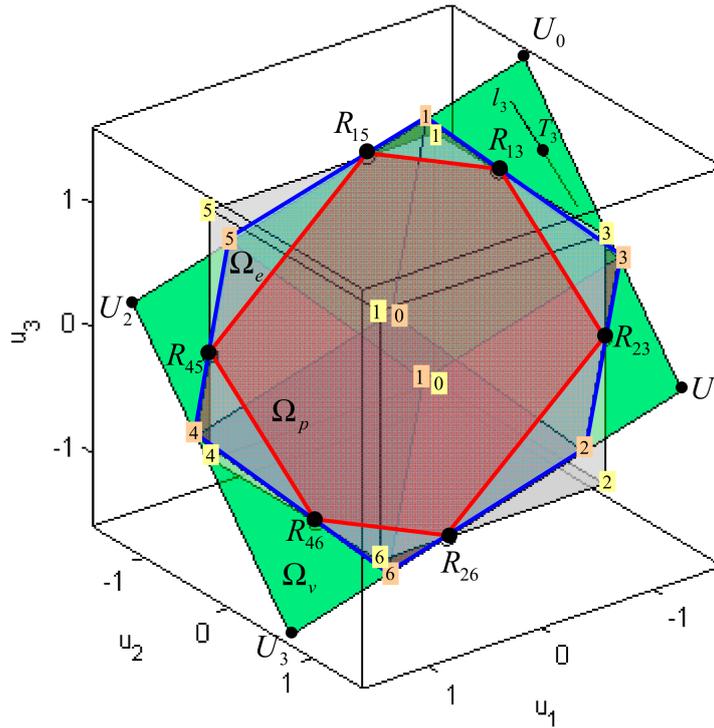


Figure 17. Case $S_3 \in \Phi \setminus \Phi_v$ yields to the unfeasible weighted pseudoinverse solution $T_3 \in \Omega_v \setminus \Omega_e$ that lies outside Ω .

For example, if $S_3 = \begin{bmatrix} -0.9000 & 0.5000 \end{bmatrix}^T$, then the solution line $\ell = \ell_3$ (7) is given by

$$\ell_3 : p = \begin{bmatrix} -\frac{493}{385} + \frac{1}{4}t \\ -\frac{13}{154} + \frac{1}{5}t \\ \frac{173}{154} + \frac{3}{10}t \end{bmatrix} \tag{27}$$

The line ℓ_3 does not intersect Ω , and the exact solution of the control allocation problem does not exist (Figure 17). However, the intersection of ℓ_3 and Ω_v is the weighted pseudoinverse solution $T_3 = \mathbf{B}_{W_u}^\dagger(S_3) = \begin{bmatrix} -1.2805 & -0.0844 & 1.1234 \end{bmatrix}^T$. This solution lies outside Ω and is not feasible.

This example demonstrated that the weighted pseudoinverse could allocate the exact solution (optimal in l_2 sense) only if the virtual control input v lies in Φ_p . Otherwise, if $v \in \Phi \setminus \Phi_p$, the weighted pseudoinverse finds solutions that lie outside Ω , i.e., that violate control constraints. The approximation of these solutions is the topic of the following section.

3.2.2. Approximation of Unfeasible Solution

If a virtual control input v lies outside Φ_p (for example, S_2 and S_3 in Figure 14), then the weighted pseudoinverse solution $u = \mathbf{B}_{W_u}^\dagger v$ is unfeasible and lies outside Ω , i.e., it violates control constraints (T_2 in Figure 16 and T_3 in Figure 17). In this case, it is necessary to approximate unfeasible $u \notin \Omega$ with feasible $u^* \in \Omega$ such that $\mathbf{B}u^* \approx v$.

Definition 1. (Approximation Error): The approximation error is defined as $e = v - v^*$, where v is the virtual control input, $v^* = \mathbf{B}u^*$ is an approximation of v and u^* is an approximation of $u = \mathbf{B}_{W_u}^\dagger v$. To be able to compare different approximations, two scalar errors are introduced: direction error $\theta = \cos^{-1} \frac{v^T \cdot v^*}{\|v\|_2 \|v^*\|_2}$ and magnitude error $\|e\|_2 = \|v - v^*\|_2$. The direction error represents the angle between v and v^* , while the magnitude error represents the module of the approximation error vector e (Figure 18). In the case when $\theta = 0$, the approximation v^* preserves the direction of the original vector v .



Figure 18. Approximation error $e = v - v^* = v - Bu^*$: (a) Case $\theta \neq 0$; (b) Case $\theta = 0$.

Two common approximations (Truncation and Scaling) are defined in the following:

Truncation (T-approximation): In this case approximation u_t^* is obtained from u by truncating (clipping) all controls that exceed their control constraints.

Scaling (S-approximation): In this case approximation u_s^* is obtained from u by scaling all components by factor f such that $u_s^* = fu \in \partial(\Omega_p)$.

T-approximations (T_{2t}^* and T_{3t}^*) and S-approximations (T_{2s}^* and T_{3s}^*) of unfeasible weighted pseudoinverse solutions T_2 (Case 2.2) and T_3 (Case 2.3) are shown in Figure 19, respectively.

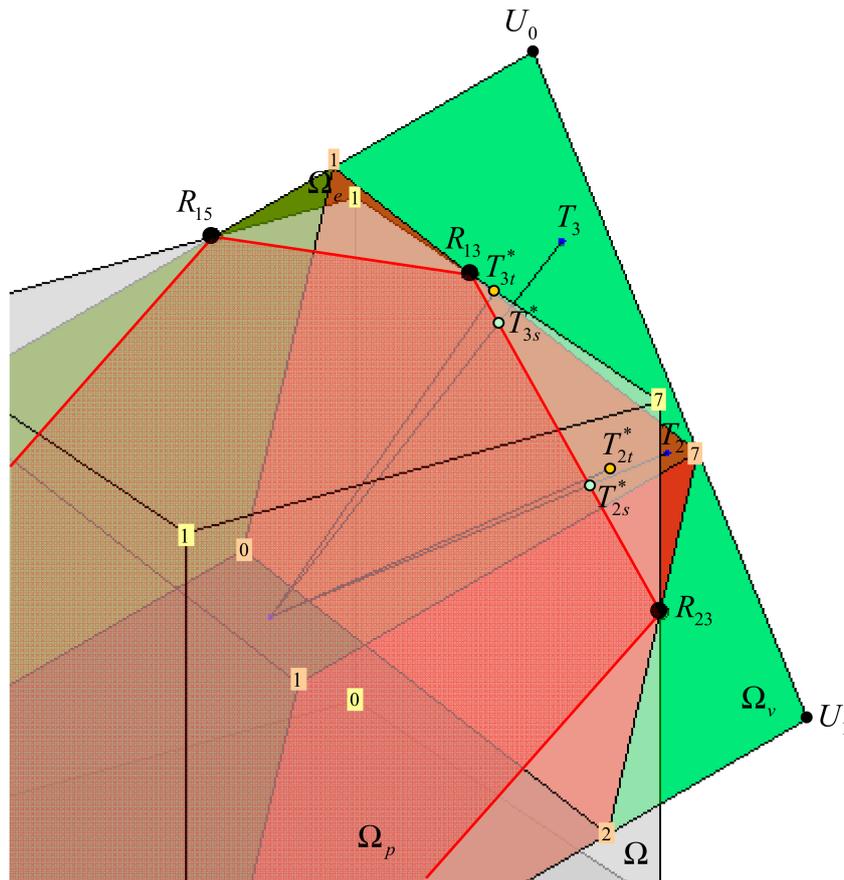


Figure 19. Approximation of unfeasible pseudoinverse solutions in the true control space.

Remarks 2.

- T-approximation u_t^* of the unfeasible pseudoinverse solution, u introduces direction error $\theta_t \neq 0$, i.e., vectors v and v_t^* have not the same direction. At the same time, the direction error $\theta_s = 0$ for S-approximation u_s^* ,

i.e., vectors \mathbf{v} and \mathbf{v}_s^* always have the same direction, but the magnitude error $\|\mathbf{e}_s\|_2 = \|\mathbf{v} - \mathbf{v}_s^*\|_2$ is greater than $\|\mathbf{e}_t\|_2 = \|\mathbf{v} - \mathbf{v}_t^*\|_2$.

- The fixed-point method (Section 3.3) is able to improve the T- or S-approximation of the unfeasible weighted pseudoinverse solution \mathbf{u} . Approximations \mathbf{u}_t^* or \mathbf{u}_s^* can be used as the initial iteration \mathbf{u}_0 and the algorithm will find the solution \mathbf{u}_f^* such that $\mathbf{v}_f^* = \mathbf{B}\mathbf{u}_f^*$ is a better approximation of \mathbf{v} than \mathbf{v}_t^* or \mathbf{v}_s^* . This feature is the main idea of the hybrid approach for control allocation.

3.3. Fixed-Point Method

Introduction

The fixed-point method [21] finds the control vector \mathbf{u} that minimises

$$J(\mathbf{u}) = (1 - \varepsilon)\mathbf{W}_v\|(\mathbf{B}\mathbf{u} - \mathbf{v})\|_2^2 + \varepsilon\|\mathbf{W}_u\mathbf{u}\|_2^2 \tag{28}$$

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

$$\mathbf{u}_{k+1} = \text{sat}\left[(1 - \varepsilon)\eta\mathbf{B}^T\mathbf{Q}_1\mathbf{v} - (\eta\mathbf{H} - \mathbf{I})\mathbf{u}_k\right] \tag{29}$$

where

$$\mathbf{Q}_1 = \mathbf{W}_v^T\mathbf{W}_v \tag{30}$$

$$\mathbf{Q}_2 = \mathbf{W}_u^T\mathbf{W}_u \tag{31}$$

$$\mathbf{H} = (1 - \varepsilon)\mathbf{B}^T\mathbf{Q}_1\mathbf{B} + \varepsilon\mathbf{Q}_2 \tag{32}$$

$$\eta = 1/\|\mathbf{H}\|_2 \tag{33}$$

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

$$|J(\mathbf{u}_{k+1}) - J(\mathbf{u}_k)| < \text{tol} \tag{34}$$

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 efficiency, authors [22] 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 .

Example 3. In this example, the fixed-point method is used to find a feasible solution for Case 2.2 from Example 2, optimal in l_2 sense. Design parameters are $\mathbf{W}_u = \mathbf{I}_3$, $\mathbf{W}_v = \mathbf{I}_2$, $\varepsilon = 10^{-6}$ and $\text{tol} = 10^{-6}$. Results are shown in Figure 20 and Table 6. Since the (unfeasible) pseudoinverse solution $\mathbf{u}_2 = T_2$ has been already found (Figure 16), it is natural to choose the feasible approximation of this solution as the initial point for the iteration. Two choices are available: the T-approximation $\mathbf{u}_{2t}^* = T_{2t}^* = \begin{bmatrix} -1.0000 & 0.6636 & 0.5955 \end{bmatrix}^T$ and S-approximation $\mathbf{u}_{2s}^* = T_{2s}^* = \begin{bmatrix} -1.0000 & 0.5328 & 0.4781 \end{bmatrix}^T$, obtained by scaling T_2 by factor $f_2 = 0.8029$. Fixed-point iterations in the True Control Space are shown in Figure 20a. Individual iterations are shown as black dots if they start from the T-approximation, and as red dots, if they start from the S-approximation. Details about iterations are given in Table 6. If $\mathbf{v}_2 \in \Phi_v \setminus \Phi_p$, the fixed-point algorithm converges toward the exact solution $T_{2f}^* = P_1$, which lies on the solution set \mathfrak{S}_2 and has a lower l_2 norm than any other point in \mathfrak{S}_2 . The corresponding sequence in the virtual control space converges toward the desired $S_{2f}^* = S_2$ (Figure 20b).

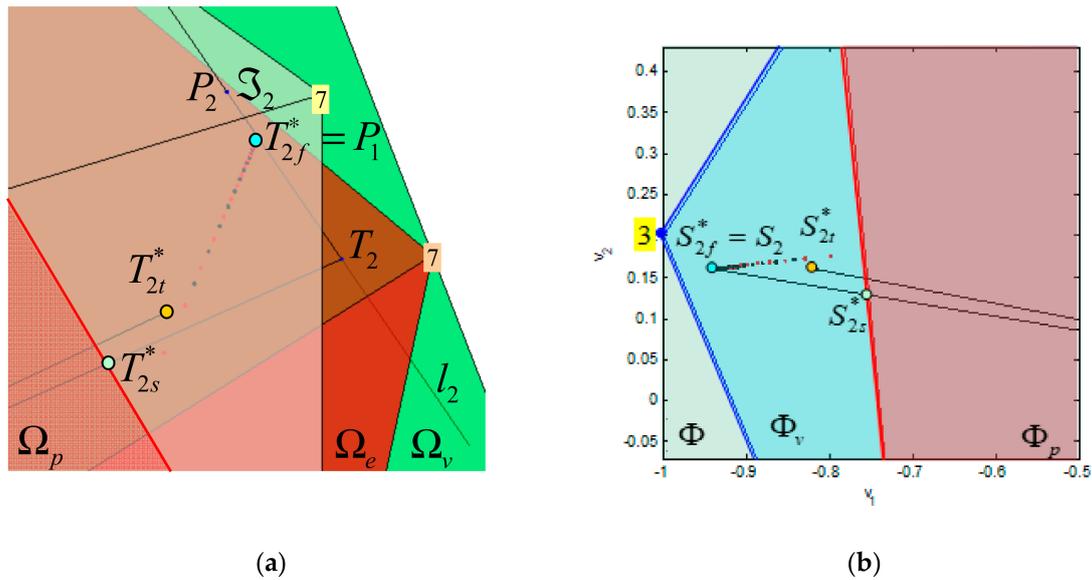


Figure 20. Fixed-points iterations for $v_2 \in \Phi_v \setminus \Phi_p$: (a) True Control Space: Black dots (Red dots) represent iterations starting at T_{2t}^* (T_{2s}^*); (b) Virtual Control Space: Black dots (Red dots) represent virtual control inputs related to iterations starting at T_{2t}^* (T_{2s}^*).

Table 6. Iterations of the fixed-point method for $v_2 \in \Phi_v \setminus \Phi_p$.

Initial Point	# Of Iterations	Last Iteration	Limit	Obtained Virtual Control Input	Desired Virtual Control Input	Direction Error	Magnitude Error
u_0	k	u_{f_k}	u_f	v_{f_k}	v	θ_k	$\ e_k\ _2$
u_{2t}^*	19	$\begin{bmatrix} -1.0000 \\ 0.8585 \\ 0.8874 \end{bmatrix}$	$\begin{bmatrix} -1.0000 \\ 0.8600 \\ 0.8900 \end{bmatrix}$	$\begin{bmatrix} -0.9365 \\ 0.1601 \end{bmatrix}$	$\begin{bmatrix} -0.9375 \\ 0.1600 \end{bmatrix}$	0.0181°	0.0010
u_{2s}^*	20	$\begin{bmatrix} -1.0000 \\ 0.8582 \\ 0.8870 \end{bmatrix}$	$\begin{bmatrix} -1.0000 \\ 0.8600 \\ 0.8900 \end{bmatrix}$	$\begin{bmatrix} -0.9363 \\ 0.1601 \end{bmatrix}$	$\begin{bmatrix} -0.9375 \\ 0.1600 \end{bmatrix}$	0.0208°	0.0012

3.4. Algorithm (Hybrid Method)

- Step 1:** For a given $\tau_d(v_d)$, 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 weighted 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 $201^4 = 1.632240801 \times 10^9$ 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.

3.5. Extension of Concepts from “virtual” ROV to Open-Frame ROV

In previous sections, the main idea of the hybrid method is visualised, and the deep geometric insight is provided using a “virtual” ROV in low-dimensional control spaces, including visualisation of the attainable command set, solution lines, control energy spheres and the role of pseudoinverse and fixed-point iterations. In this section, the same concepts are extended to higher-dimensional cases, for open-frame ROV with four X-shaped (vectored) horizontal thrusters.

Condition (4) for open-frame ROV determines the constrained control subset Ω , that is, the unit 4D cube (the tesseract) in true control space:

$$\Omega = \{u \in \mathbb{R}^4 : \|u\|_\infty \leq 1\} \tag{35}$$

The tesseract Ω (35) has 8 cells, $2^{4-2} \binom{4}{2} = 24$ faces, $2^{4-1} \binom{4}{1} = 32$ edges and $2^4 = 16$ vertices. Equation (3) defines a linear transformation from the true control space \mathbb{R}^4 to the virtual control space \mathbb{R}^3 , which maps (projects) the tesseract Ω into attainable command set Φ_v (see Figure 21), a subset of the allowable virtual control subset Φ (the unit 3D cube), defined as

$$\Phi = \{\tau \in \mathbb{R}^3 : \|\tau\|_\infty \leq 1\} \tag{36}$$

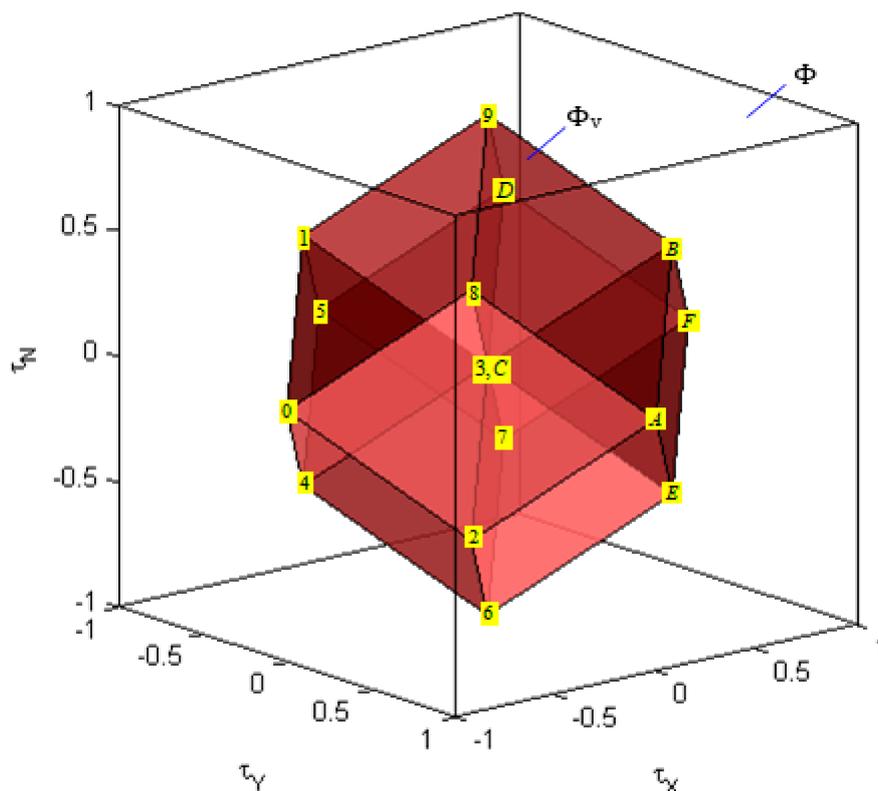


Figure 21. The allowable virtual control subset $\Phi \subset \mathbb{R}^3$ and the attainable command set Φ_v for open-frame ROV.

Coordinates of vertices of Ω and Φ_v are shown in Table 7. Recall from Section 2.2 that the component values (controls) of u in a vertex can be decoded from its binary representation. For example, decoding the binary representation 1000 of the vertex 8 yields $u_1 = +1, u_2 = -1, u_3 = -1$ and $u_4 = -1$.

Table 7. Coordinates of vertices of Ω and Φ_v .

Vertices of Ω					Vertices of Φ_v			
Label	Coordinates				Label	Coordinates		
0	-1	-1	-1	-1	0	-1.0	0.0	0.0
1	-1	-1	-1	+1	1	-0.5	-0.5	0.5
2	-1	-1	+1	-1	2	-0.5	0.5	-0.5
3	-1	-1	+1	+1	3	0.0	0.0	0.0
4	-1	+1	-1	-1	4	-0.5	-0.5	-0.5
5	-1	+1	-1	+1	5	0.0	-1.0	0.0
6	-1	+1	+1	-1	6	0.0	0.0	-1.0
7	-1	+1	+1	+1	7	0.5	-0.5	-0.5
8	+1	-1	-1	-1	8	-0.5	0.5	0.5
9	+1	-1	-1	+1	9	0.0	0.0	1.0
A	+1	-1	+1	-1	A	0.0	1.0	0.0
B	+1	-1	+1	+1	B	0.5	0.5	0.5
C	+1	+1	-1	-1	C	0.0	0.0	0.0
D	+1	+1	-1	+1	D	0.5	-0.5	0.5
E	+1	+1	+1	-1	E	0.5	0.5	-0.5
F	+1	+1	+1	+1	F	1.0	0.0	0.0

Considering Equation (3) (system of three linear equations with four unknown variables) from a row perspective, the solution set is the hyper line ℓ (the intersection of three hyperplanes in 4D space). The overall solution of the control allocation problem (3)–(4) is a solution set \mathfrak{J} , the intersection of hyper line ℓ and Ω . The existence and uniqueness of the solution u depends on the position of the virtual control input τ relative to the attainable command set Φ_v :

- If τ lies inside Φ_v , then \mathfrak{J} 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_v , then \mathfrak{J} is a single point, the unique solution for the control allocation problem.
- If τ lies outside Φ_v , then \mathfrak{J} is an empty set, i.e., no exact solution exists, only approximation.

For the X-shaped configuration of horizontal thrusters the matrix $B_{W_u}^+$ is given by

$$B_{W_u}^+ = \frac{2}{\sum_{i=1}^4 w_i} \begin{bmatrix} w_3 + w_4 & w_2 + w_3 & w_2 + w_4 \\ w_3 + w_4 & -(w_1 + w_4) & -(w_1 + w_3) \\ w_1 + w_2 & w_1 + w_4 & -(w_2 + w_4) \\ w_1 + w_2 & -(w_2 + w_3) & w_1 + w_3 \end{bmatrix} \quad (37)$$

Feasibility of the weighted pseudoinverse solution

$$u = B_{W_u}^+ \tau \quad (38)$$

depends on the position of virtual control input τ inside the allowable virtual control subset Φ . For a fault-free case, when W_u is the identity (unit) matrix, the shape of the feasible region for the pseudoinverse Φ_p , defined as a set of virtual control inputs τ that satisfy (38) subject to $u \in \Omega$, is shown in Figure 22. 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 (38) 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 (38) satisfies $u_k = +1$ (-1).

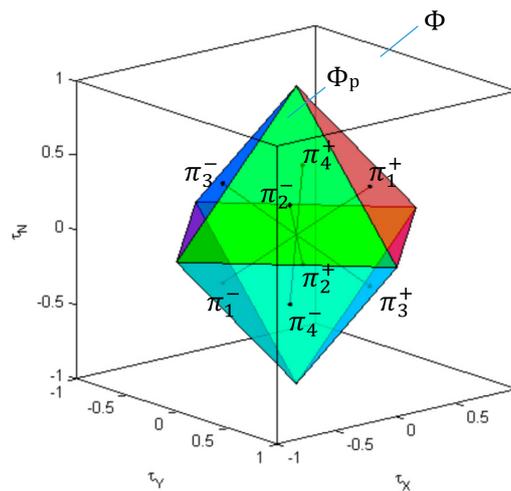


Figure 22. The feasible region for the weighted pseudoinverse Φ_p inside the allowable virtual control subset Φ for X-shaped configuration of horizontal thrusters (fault-free case).

The weighted pseudoinverse can find the exact feasible solution of the control allocation problem, optimal in the l_2 sense, only if $\tau_d \in \Phi_p$. Geometrically, Equation (38) defines a linear transformation from the virtual control space to the hyperplane Π in the true control space \mathbb{R}^4 . The intersection of this hyperplane with the hyper line ℓ (solution of $Bu = \tau$) is the point u^* where the family of weighted hyperspheres “touch” the hyper line ℓ . Since $\tau \in \Phi_p$, this intersection determines a feasible solution $u^* = B_{W_u}^\dagger \tau \in \Omega$, optimal in l_2 sense. The main issue with the weighted pseudoinverse is that the feasible region for the weighted pseudoinverse Φ_p is a subset of attainable command set Φ_v ($\Phi_p \subset \Phi_v$), and the weighted pseudoinverse cannot find an optimal feasible solution on the entire Φ_v .

Extending ideas from Section 3.2, the allowable virtual control subset Φ can be partitioned into characteristic regions, as indicated in Figure 23. The two characteristic regions inside Φ are Φ_p and $\Phi_v \supset \Phi_p$. The weighted pseudoinverse is able to find the exact feasible solution of the control allocation problem, optimal in l_2 sense, only if $\tau \in \Phi_p$. Otherwise, for $\tau \in \Phi_v \setminus \Phi_p$, the solution obtained by weighted pseudoinverse is unfeasible. However, the fixed point method can find the solution, optimal in l_2 sense, for case $\tau \in \Phi_v \setminus \Phi_p$. Finally, for case $\tau \in \Phi \setminus \Phi_v$, the solution does not exist, and approximation must be used (for example, T-approximation (truncation) or S-approximation (scaling), as described in Section 3.2.2).

It is useful to visualise the change in shapes of the attainable command set Φ_v and the feasible region for the weighted pseudoinverse Φ_p depending on faulty thruster states. In the fault-free state, all thrusters are equally prioritised, and the actuator position constraint in (4) 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 fault.

Change in the constraint bound yields to increase of corresponding weight in the weighting matrix W_u

$$w_i = 1 + \Delta w_i \tag{39}$$

where

$$\Delta w_i = 2 \left(\frac{1}{s_i} - 1 \right) \tag{40}$$

In the 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 u . The image of the “clipped” Ω under mapping (3) is reduced attainable command set Φ_v . Hence, the feasible region for the pseudoinverse Φ_p and the attainable command set Φ_v decrease in the case of a fault in the single thruster. In particular, Figure 24a displays Φ_p only and Figure 24b shows $\Phi_p \& \Phi_v$ for the case of a partial fault in HT_2 ($s_2 = 0.5$, $w_2 = 3$). Analysing (37) and (38), 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_v , as shown in Figure 24b. It can be seen that the relation $\Phi_v \supset \Phi_p$ is still valid, i.e., the weighted pseudoinverse is not able to find a feasible solution for case $\tau_d \in \Phi_v \setminus \Phi_p$. Similar to the fault-free case, the fixed-point method is triggered 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 optimally, despite the limited (restricted) usage of a faulty thruster.

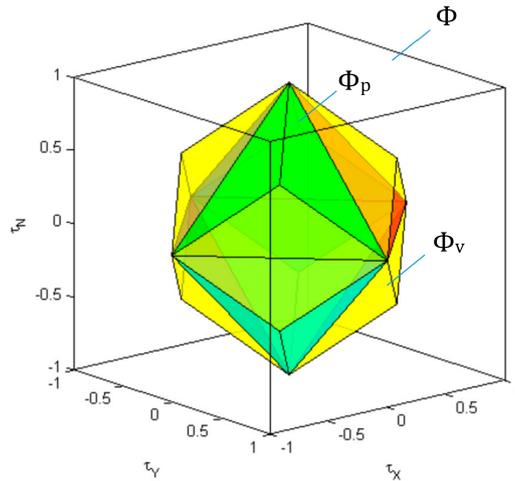


Figure 23. Partitions of the allowable virtual control subset Φ for X-shaped configuration of horizontal thrusters (fault-free case).

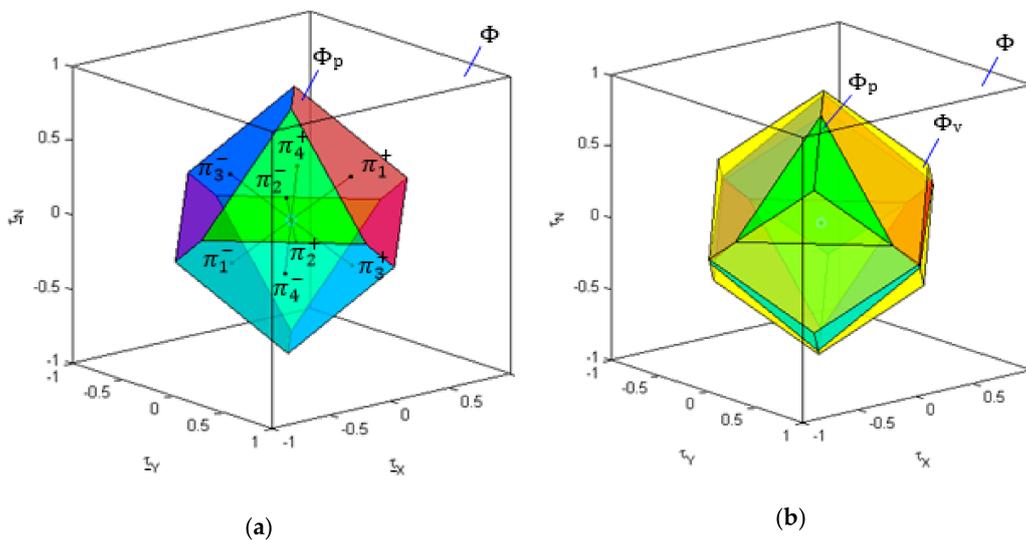


Figure 24. Partitions of the allowable virtual control subset Φ for X-shaped configuration of horizontal thrusters (partial fault in HT_2 , $s_2 = 0.5$, $w_2 = 3$): (a) Φ_p ; (b) $\Phi_p \& \Phi_v$.

Partitions of the allowable virtual control subset for the extreme case (shutdown of HT_2) is shown in Figure 25. In particular, Figure 25 displays $\Phi_p \equiv \Phi_v$. for the case of a total fault in HT_2 ($s_2 = 0.0$, $w_2 \rightarrow \infty$). This means that the thruster HT_2 is disabled and the redundancy is eliminated by removing the variable u_2 and the second column of matrix B from (3). The modified thruster control matrix 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 Figures 24b and 25 relative to the full size Φ_v (Figure 23), can be used as a measure of loss in manoeuvring capabilities.

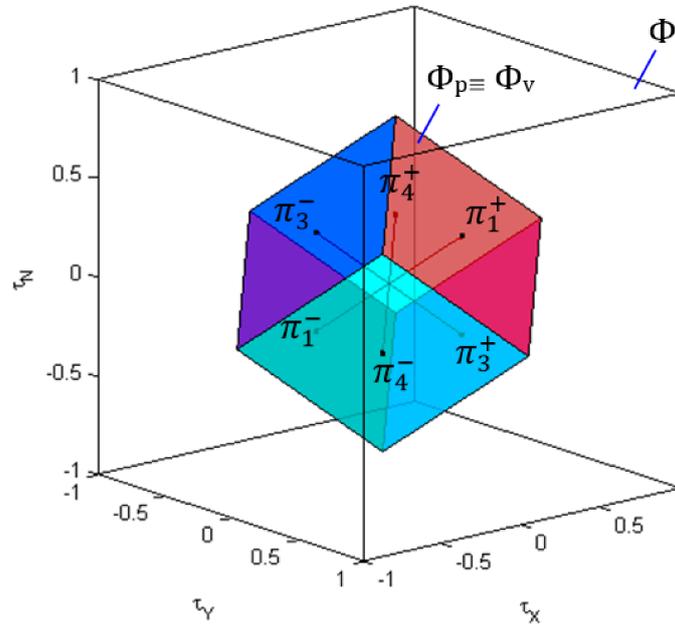


Figure 25. Partitions of the allowable virtual control subset Φ for X-shaped configuration of horizontal thrusters (total fault in HT_2 , $s_2 = 0.0$, $w_2 \rightarrow \infty$): $\Phi_p \equiv \Phi_v$.

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

$$\Omega = \{(u_1, 0, u_3, u_4) \in \mathbb{R}^4 : |u_1| \leq 1, |u_3| \leq 1, |u_4| \leq 1\} \tag{41}$$

Ω is mapped by modified B to Φ_v , which coincides with Φ_p , as indicated in Figure 25.

4. Testing and Validation

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 in a virtual and real-world environment.

4.1. Evaluation of the FTC in Virtual Environment

In the virtual (simulated) environment thruster faults are simulated by varying properties of the dynamic thruster model (load, friction, etc.) inside the propulsion subsystem of the ROV dynamics simulator. More information about simulation models can be found in (Omerdic, 2009). To enhance the graphical presentation of simulation results, a virtual underwater world has been developed with open-frame ROV model moving in an underwater environment. The relative position of different objects in the underwater world is shown in Figure 26. Three particular objects (the rock with a hole in the middle, long pipe and "Stonehenge"-like group of rocks) were used throughout the number of test cases to evaluate the manoeuvring capabilities and performance of the ROV when equipped with the FTC. Selected test cases will demonstrate the ability of the FTC to complete the mission in the presence of a single (partial and total) fault in HT_2 . Diagrams for fault-free and faulty cases are shown next to each other, to provide an easier comparison. Command inputs in test cases were generated using pre-defined signals read from a file.

Two criteria are introduced to measure control effort over time:

$$\text{Weighted criterion : } J_{W_u} = u^T W_u u \tag{42}$$

and

$$\text{Normal criterion : } J_n = \mathbf{u}^T \mathbf{u} \tag{43}$$

The weighted criterion J_{W_u} , minimised by the hybrid method, can be interpreted as a weighted control energy cost function. In contrast, the normal criterion J_n represents the actual control energy cost function, that is, a real measure of control effort. In fault-free case $J_{W_u} = J_n$, whereas in faulty situations, $J_{W_u} \geq J_n$. Criteria J_{W_u} and J_n are denoted in diagrams as J_W and J_n , respectively.

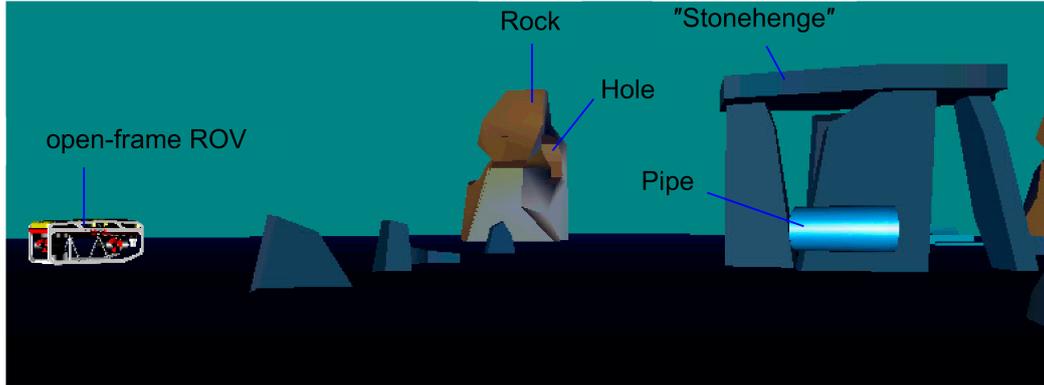


Figure 26. Virtual underwater world.

4.1.1. Partial Fault in HT₂

This test case compares the performance of the FTC for two cases. In the first case, a simulation was performed assuming fault-free states in all thrusters. The second case considered fault-free states in thrusters HT₁, HT₃ and HT₄, and a faulty state in thruster HT₂ (partial fault, $s_2 = 0.5$). The same command input vector $\underline{\tau}_d(t)$ was used to drive the ROV model in both cases. The mission objective was to move the vehicle from the start point A through the pipe along the straight line. Figure 27a displays time diagrams of $\underline{\tau}_d(t)$, $\underline{\tau}(t)$, $\mathbf{u}(t)$, $J_{W_u}(t)$, $J_n(t)$, $v(t)$ and $\Psi(t)$, the plan view of the trajectory $(x_E(t), y_E(t))$ and partitions Φ_p & Φ_v of the allowable virtual control subset Φ for the first case. Diagrams for the second case are shown in Figure 27b. It can be seen that, in the fault-free case, the vehicle performs the perfect straight-line motion, vector $\underline{\tau}_d(t)$ lies inside Φ_v for all time and $\underline{\tau}_d(t) \equiv \underline{\tau}(t)$, $J_{W_u}(t) \equiv J_n(t)$, $\forall t$. The situation is different for the second case, where the limited usage of HT₂ reduction in the size of the Φ_p and Φ_v (see Figure 24), and the trajectory $\underline{\tau}_d(t)$ becomes partially unfeasible, i.e., $\underline{\tau}(t)$ lies partially outside Φ_v . The FTC uses the S-approximation to approximate unfeasible solutions in these cases, which leads to an approximation $\underline{\tau}(t) = \underline{\tau}_s(t)$ with the same direction as unattainable command input $\underline{\tau}_d(t)$, but lower magnitude. The unfeasible part of trajectory $\underline{\tau}_d(t)$ is characterised by relationship $\underline{\tau}(t) = \underline{\tau}_d(t)$ and indicated as shadowed regions in time responses shown in Figure 27b. Pairs of thrusters HT₁ & HT₂ and HT₃ & HT₄ are equally actuated for straight-line motion ($u_1(t) = u_2(t)$ and $u_3(t) = u_4(t)$, respectively). The velocity profile $v(t)$ follows the profile of $\underline{\tau}_X(t)$, which is now different than $\underline{\tau}_{Xd}(t)$, and the vehicle moves slower in shadowed region than in the fault-free case. However, heading $\Psi(t)$ is constant in both cases, but trajectories $(x_E(t), y_E(t))$ are slightly different. The first difference stems from the fact that the higher forward velocity in the fault-free case means that the vehicle covers a longer distance than in the faulty situation, i.e., $\overline{AB}_{\text{fault-free}} > \overline{AB}_{\text{partial fault in HT}_2}$. The second difference comes from the shapes of the trajectories $(x_E(t), y_E(t))$. In the first case, it is a perfectly straight line, whereas, in the second case, although symmetrical pairs of thrusters have opposite spin direction coefficients, the vehicle is shifted from the straight line, such that the drift at the endpoint B is approximately 2.7cm. The drift is caused by unequal actuation of symmetrical thrusters in the second case, which leads to non-zero angular moments from these pairs, resulting in non-perfect straight-line motion of the vehicle.

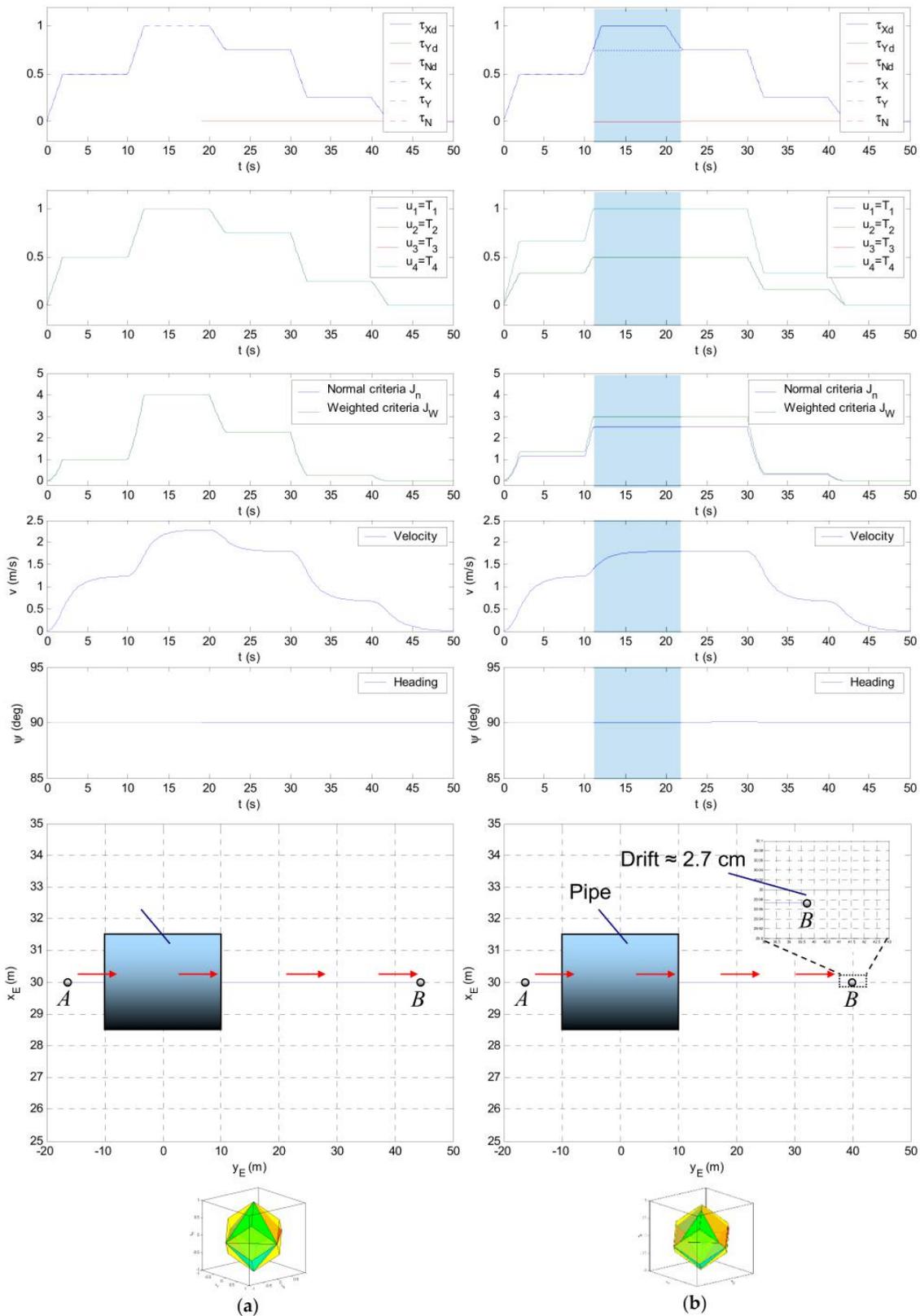


Figure 27. Partial fault in a single thruster: (a) All thrusters healthy; (b) HT₂ is faulty ($s_2 = 0.5$).

This test case demonstrates that, in the presence of a single partial fault in HT₂, the faulty vehicle is able to perform straight-line motion with satisfactory performance. An unavoidable effect is a drop in forward velocity for cases when the command input vector $\underline{\tau}_d(t)$ lies outside reduced Φ_v .

4.1.2. Total Fault in HT₂

This is an extreme case, where HT₂ is switched off, and the mission must be accomplished with three remaining horizontal thrusters. Recall from Figure 25 that Φ_p and Φ_v have the same shape (small parallelepiped inside the virtual control space), as indicated in Figure 28b. The volume of Φ_p and Φ_v is smaller than in Section 4.1.1, which yields the wider shadowed region, i.e., the larger unfeasible part of the trajectory $\tau_d(t)$. Besides, a drop in velocity $v(t)$, is larger for this test case. The faulty vehicle covers the shorter distance ($\overline{AB}_{fault-free} > \overline{AB}_{total\ fault\ in\ HT2}$) and the drift at the endpoint B is approximately 2.7cm. Nevertheless, again the heading $\Psi(t)$ is constant for all time.

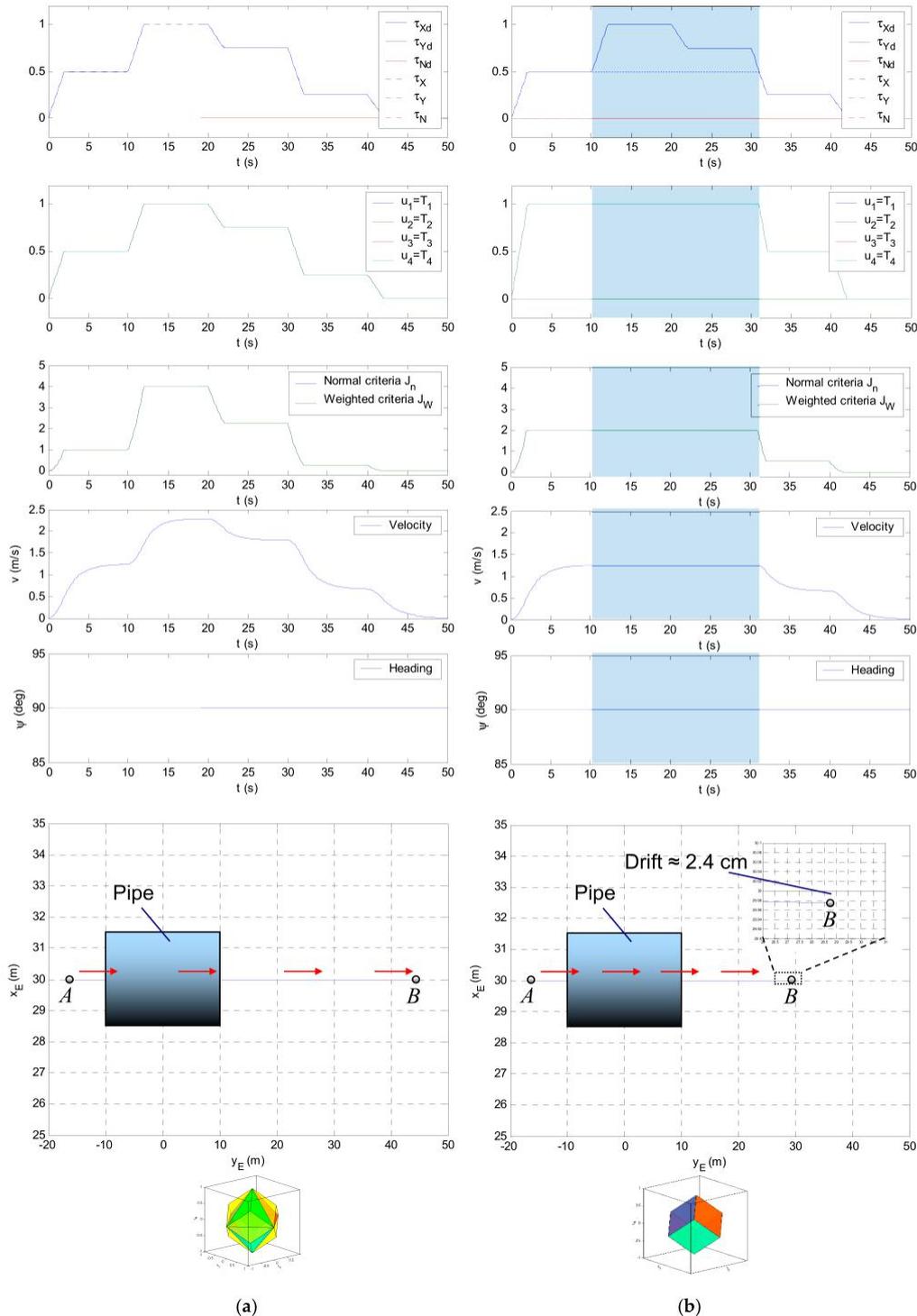


Figure 28. Total fault in a single thruster: (a) All thrusters healthy; (b) HT₂ is faulty ($s_2 = 0.0$).

This test case reveals an important feature of the FTC; that is, despite the presence of a total fault in HT_2 , which is switched off, the vehicle, equipped with the FTC, is able to continue the straight-line motion and complete the mission with acceptable performance. As in previous cases, the unavoidable consequence is a drop in velocity for cases when the command input vector $\underline{\tau}_d(t)$ lies outside reduced Φ_v .

Materials and Methods should be described with sufficient details to allow others to replicate and build on published results. Please note that publication of your manuscript implicates that you must make all materials, data, computer code, and protocols associated with the publication available to readers. Please disclose at the submission stage any restrictions on the availability of materials or information. New methods and protocols should be described in detail while well-established methods can be briefly described and appropriately cited.

4.2. Evaluation of the FTC in Real-World Environment

The FTC has been used as a foundation for the development of OceanRINGS, a set of smart technologies for subsea operations. The OceanRINGS [23] is a generic fault-tolerant ROV control system, built to make complex subsea tasks simple, and which combines state-of-the-art navigation equipment with the best control algorithms and emerging VR technologies to provide smart, intuitive and easy to use user interface, enabling average pilots to achieve exceptional results. The performance of FTC has been evaluated and validated during several research cruises in period 2009–2019 off the West coast of Ireland using observation-class ROV Latis and work-class ROV Étaín. Smart, multiple modes of operation observation-class ROV Latis [23] (Figure 29a) has been designed at UL in period 2006–2009 and used as a prototype platform to design and develop OceanRINGS modules. The work-class ROV Étaín (modified ROV Comanche by SubAtlantic, Figure 29b) has been extended with OceanRINGS control suite. Dual-mode of operation provides an opportunity to evaluate the performance of both control systems under the same conditions. Selected results from these trials, related to testing of fault-tolerant control system features, are described in the following.

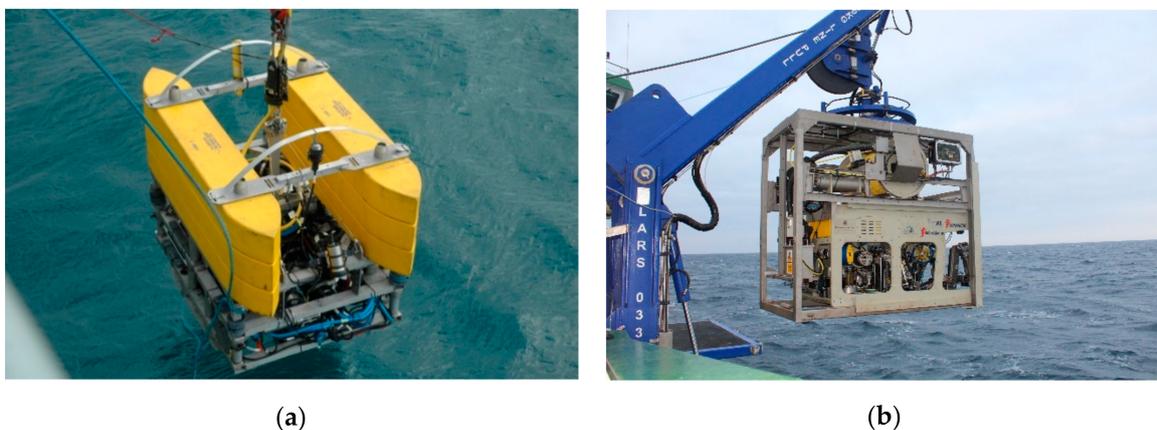


Figure 29. Underwater robots: (a) observation-class ROV Latis; (b) work-class ROV Étaín.

4.2.1. Path Following: Simulated Faults

The path following test with simulated faults has been performed during the research cruise in Bantry Bay (Figure 30) when ROV Latis has been deployed from support vessel Celtic Voyager in free-swimming mode. Since the vessel Celtic Voyager was not equipped with Dynamic Positioning (DP) system, it used an anchor to hold the position during the test.

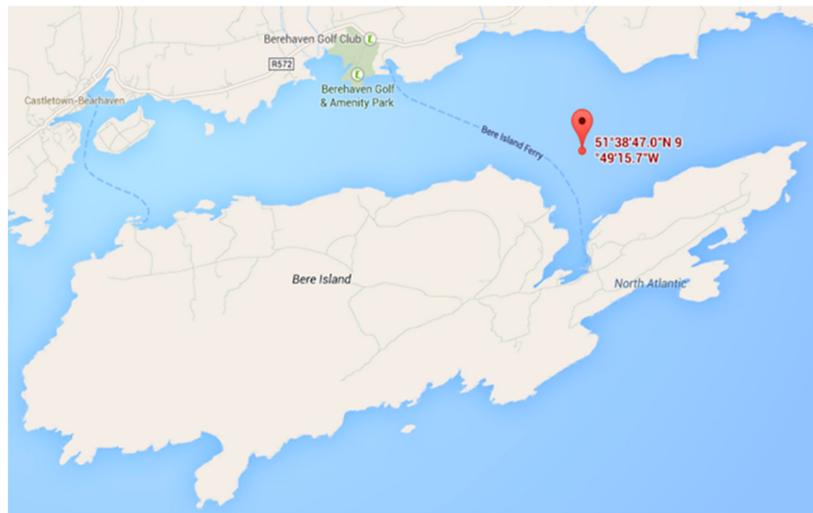


Figure 30. Location of the research cruise.

The path was defined as a set of waypoints 0, 1, 2, 3, 4 & 5, connected with straight-line segments 01, 12, 23, 34 & 45, as indicated in Figure 31. The main objective was an evaluation of the FTC performance in path following, with simulated faults in single thrusters along the path. Thruster states along path segments are given in Table 8. The main sensor system for navigation, integrated on ROV Latis, was iXBlue PHINS, with several aiding sensors: GAPS USBL, RDI DVL, GPS and Digiquartz pressure depth sensor.

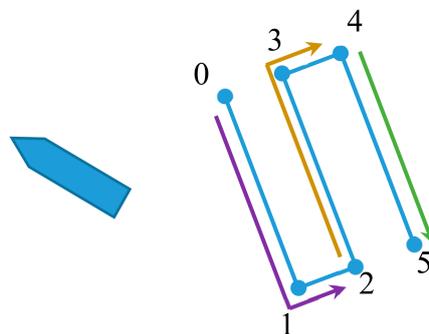


Figure 31. Path segments.

Table 8. Thruster states, along with path segments.

Segment	HT ₁	HT ₂	HT ₃	HT ₄
01	ON	ON	ON	ON
12	ON	ON	ON	ON
23	ON	ON	ON	OFF
34	ON	ON	ON	OFF
45	ON	OFF	ON	ON

Figure 32 shows path-following performance along segment 01, with all thrusters ON. Desired speed was set to 0.3 m/s. 3D real-time view is shown in the upper right corner, while schematic showing thruster states is shown in the lower right corner. Above ROV 3D model, desired and actual velocity vectors are displayed as green and red cylinders, respectively. Raw ROV position estimation data obtained from the USBL system are shown as a red trace, while the position estimation from PHINS is shown as a yellow trace. The PHINS output has been used for path-following algorithm. “Jumps” in USBL raw data are due to multi-path issues while operating in shallow water (7 m depth).

The anchored support vessel Celtic Voyager was moving in lateral direction oscillating around the fixed position due to high winds. This motion introduced significant pull out forces on ROV via umbilical and the FTC had to compensate this disturbance while performing path following.

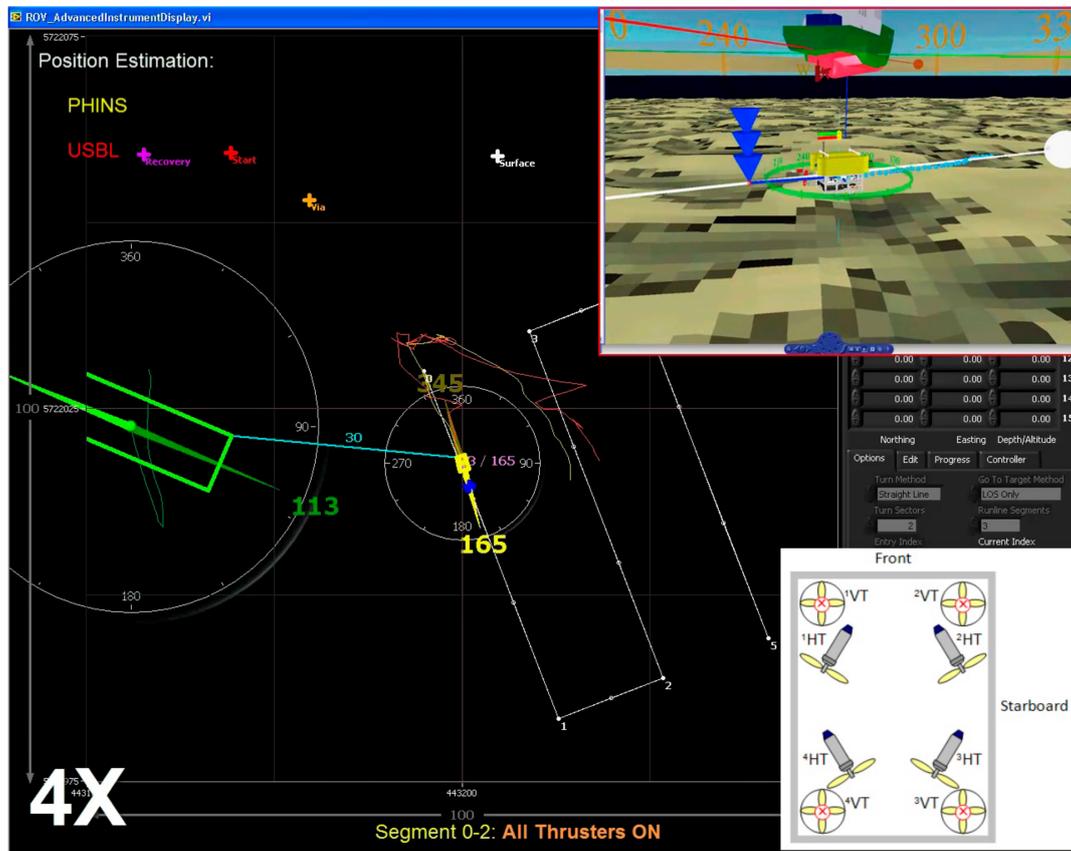


Figure 32. Path Following: segment 01, all thrusters ON (healthy).

After the way-point 2 has been reached, the total fault in HT₄ has been simulated by setting the saturation bound $s_4 = 0.0$. Figure 33 shows path-following performance along segment 23, with thruster HT₄ disabled. No significant deviation from the desired path was noticeable during the transition stage, and ROV continued to follow the path without the change in speed, as indicated in 3D view in the upper right corner (the actual velocity vector (red) has the same module and direction as the desired velocity vector (green)). This performance has been predicted in Section 4.1.2, since the command input vector $\underline{\tau}_d(t)$ required to follow desired speed 0.3m/s lies inside reduced Φ_v (Figure 25) all the time on the path segments 23 and 34.

After the way-point 4 has been reached, thruster HT₄ has been re-enabled ($s_4 = 1.0$), but the total fault in HT₂ has been simulated by setting the saturation bound $s_2 = 0.0$. Figure 34 shows path-following performance along segment 45, with thruster HT₂ disabled. Similar to the previous case, no significant deviation from the desired path was noticeable during the transition stage, and ROV continued to follow the path without the change in speed, as indicated in 3D view in the upper right corner.

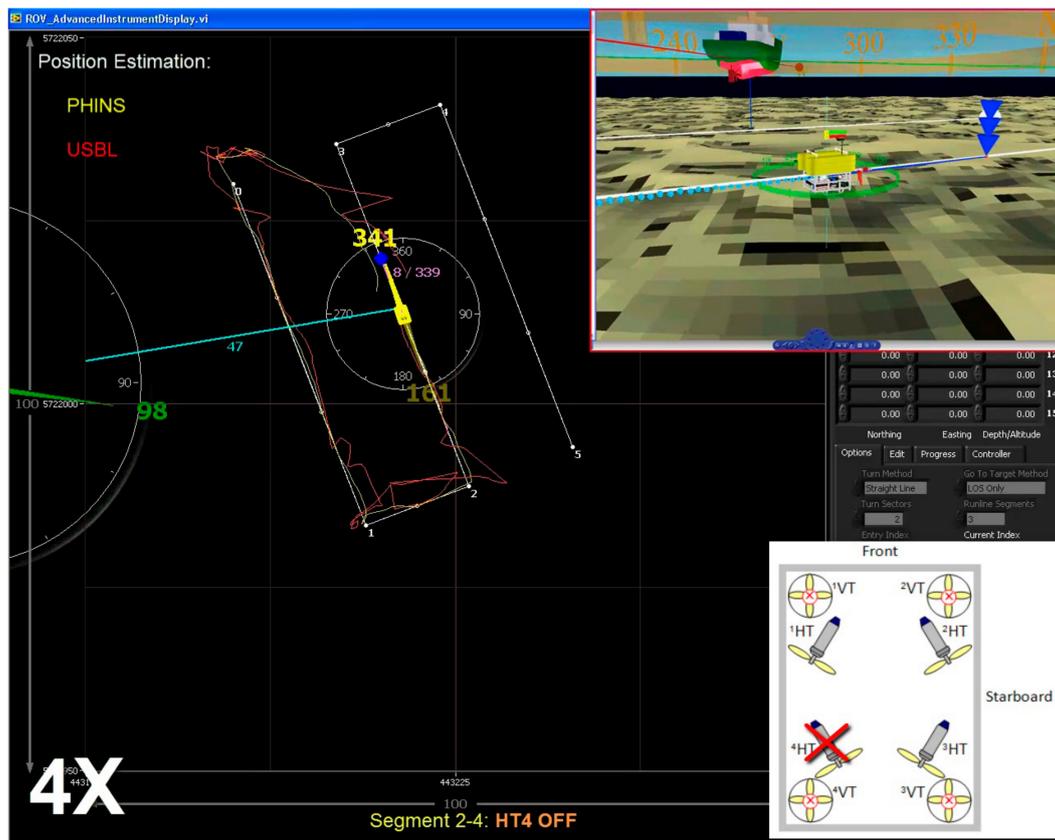


Figure 33. Path following: segment 23, HT₄ OFF (simulated total fault).

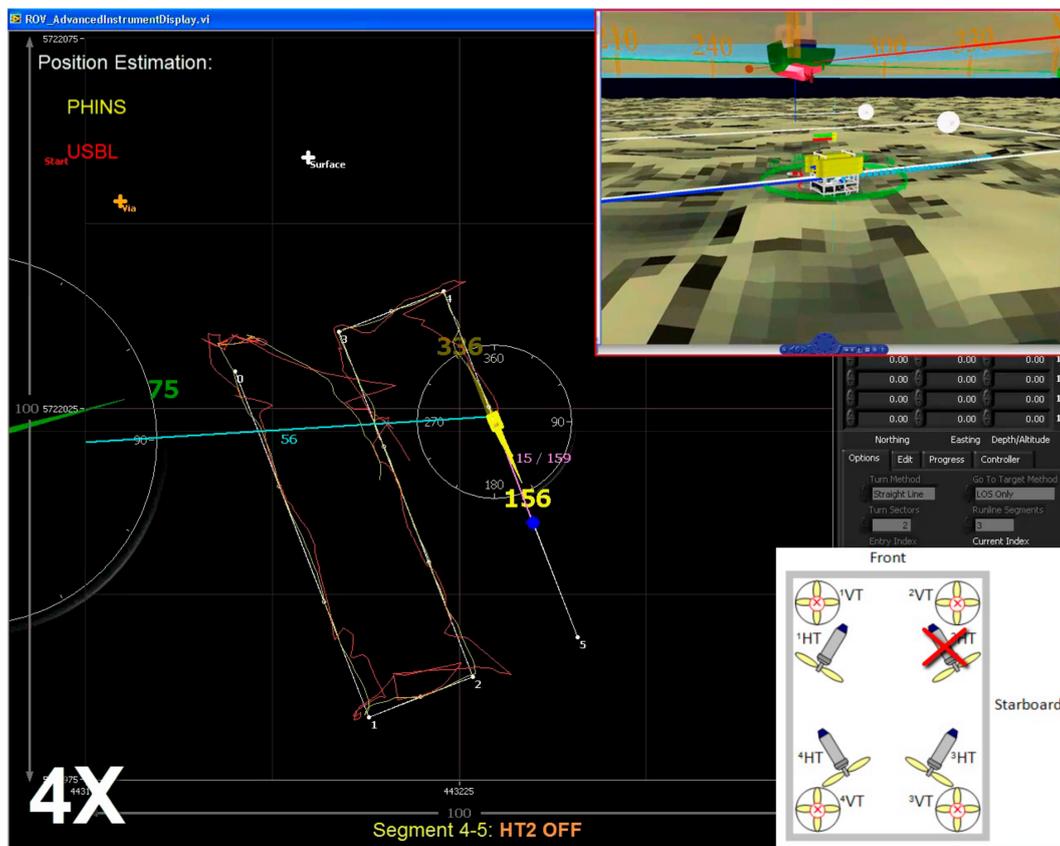


Figure 34. Path following: segment 45, HT₂ OFF (simulated total fault).

4.2.2. Complex Tasks with Faulty Thruster

The performance of FTC has been evaluated and validated during research cruise CE-19001 in January 2019, when the work-class ROV *Étaín* (Figure 29b), 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 HT_1 ($s_1 = 0.0, w_1 \rightarrow \infty$) (Figure 35c).

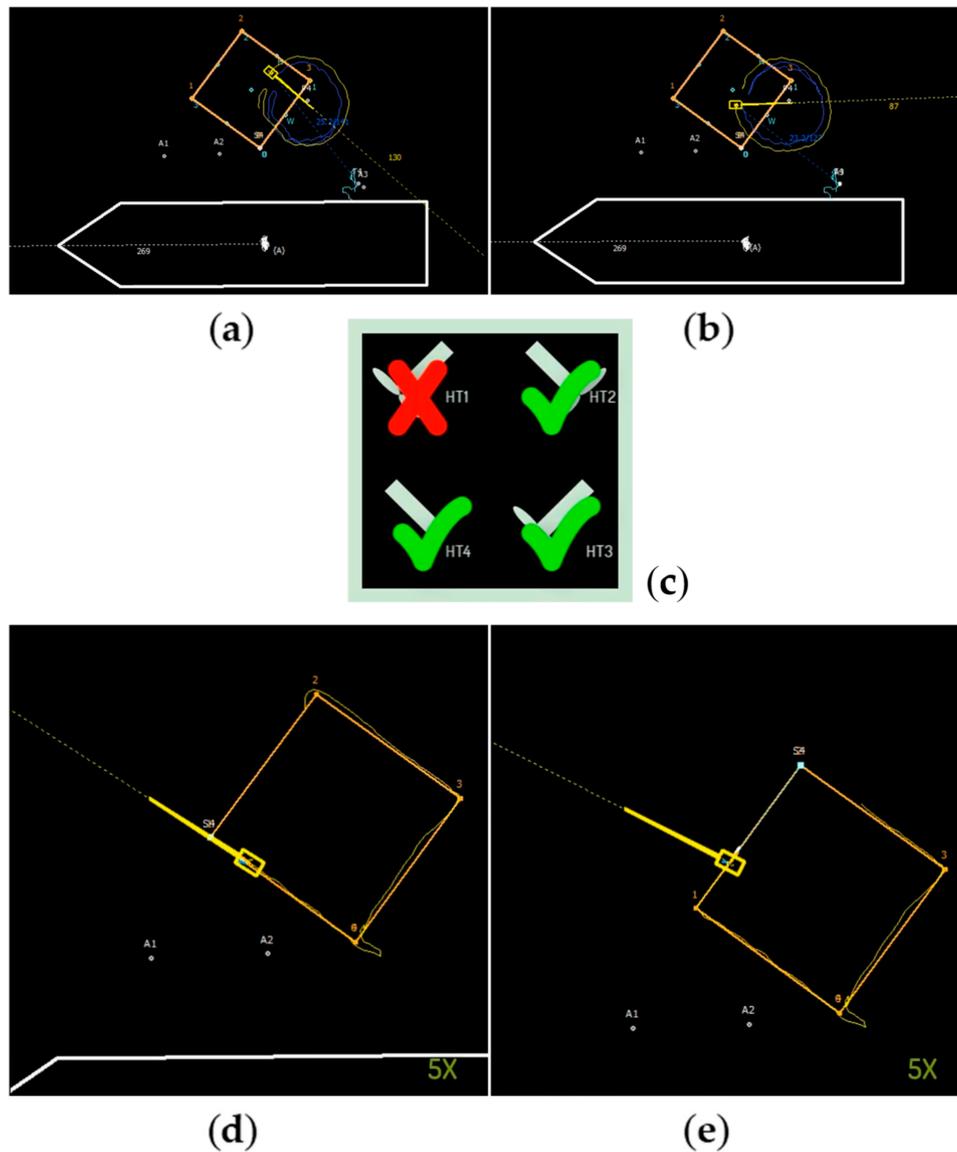


Figure 35. Testing performance with a fault in single thruster HT_1 (c); Circular arc: (a) CCW motion, (b) CW motion; Path following: (d) before WP1, (e) after WP1.

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 (Figure 35a,b). The screenshots of CCW and CW trajectories are shown in Figure 34a,b respectively. Desired surge speed was set to 0.0m/s, while desired sway speed was set to +0.2m/s (CCW motion) and -0.2m/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 FTC, but the performance was not satisfactory.

Path Following

For this test, ROV had to follow the path (square) defined with four points 0, 1, 2 & 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 are shown in Figure 35d,e. In particular, the ROV position approaching the point 1 is shown in Figure 35d. After reaching waypoint 1, the vehicle continued to move toward 2 along the straight-line segment (Figure 35e). The same task has been given to a professional ROV pilot to execute under the same fault conditions, with disabled FTC (Figure 36). 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, as shown in Figure 36.

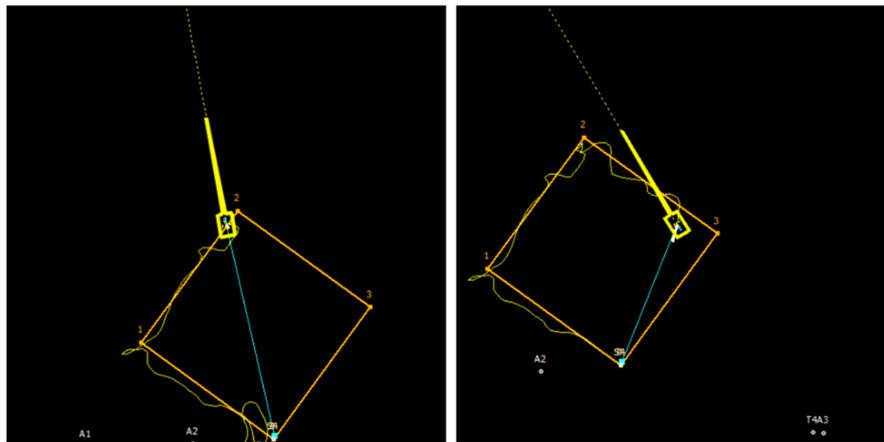


Figure 36. Path following performed by professional ROV pilot in manual mode, with disabled FTC.

5. Conclusions

This paper presented a hybrid method for control allocation, able to allocate a feasible solution of the control allocation problem, optimal in l_2 sense, on the entire attainable command set in real-time. The paper provides a clear geometric interpretation of the underlying control allocation problem and deep insight into each step of the solution using a “virtual” ROV in low-dimensional control spaces. Then, the same concepts are extended to higher-dimensional cases, for open-frame ROV with four X-shaped (vectored) horizontal thrusters.

The hybrid method for control allocation is based on the integration of the weighted pseudoinverse and the fixed-point method. The weighted pseudoinverse method is a member of a family of generalised inverses and is the one that yields minimum control energy. The main disadvantage of the weighted pseudoinverse method is its inability to find the exact solution of the control allocation problem on the entire attainable command set, i.e., the feasible region for pseudoinverse is a subset of the attainable command set. In contrast, the fixed-point method can find the exact solution on the entire attainable command set. The price paid is the necessity to perform iterations at each program step. The number of iterations depends on design parameters and choice of initial iteration. The hybrid approach for control allocation originates from the integration of features of the weighted pseudoinverse and fixed-point method: the weighted pseudoinverse method is used for cases when control inputs lie inside the feasible region for pseudoinverse, and the fixed-point method is used otherwise.

The performance of the thruster FTC system, whose main pillar is the hybrid method, has been evaluated not just in simulation, but in a real-world environment with observation-class ROV and work-class ROV through a series of complex, challenging subsea tasks in order to examine the performance of the FTC in fault-free and faulty conditions. Simulation results show that the FTC provides adequate 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 a satisfactory

way. Using different indicators and visualisation tools, the FTC can inform the ROV pilot about the position of actual command inputs relative to attainable command set. Using this information, even an inexperienced ROV pilot can detect the situation when thruster velocity saturation occurs and to correct the command inputs such that it becomes attainable. Results of real-world tests show that piloting of faulty ROV in pure manual mode was too hard task for a professional ROV pilot, i.e., disabling the faulty thruster without appropriate reallocation led to poor tracking performance and oscillatory character of yaw response during the way-point path following task. Execution of the same task with the FTC enabled led to a redistribution of available control energy among remaining operable thrusters, resulting in significant improvements in the global control performance of the vehicle in faulty situations.

Author Contributions: Conceptualization, E.O.; methodology, E.O. and D.T.; software, E.O. and P.T.; validation, M.R., A.K. and G.D., formal analysis, A.K.; investigation, P.T.; resources, A.W.; data curation, M.R.; writing—original draft preparation, E.O.; writing—review and editing, E.O., G.D. and D.T.; visualization, E.O.; supervision, D.T.; project administration, A.W.; funding acquisition, G.D. All authors have read and agreed to the published version of the manuscript.

Funding: This publication has emanated from research supported by the Science Foundation Ireland under the MaREI Centre research programme (Grant No. SFI/12/RC/2302, 15/RI/3232 and SFI/14/SP/2740) and LERO Science Foundation Ireland grant 13/RC/2094. It is also co-funded under the European Regional Development Fund through the Southern and Eastern Regional Operational Programme to Lero and MaREI centre. The development of ROV Latis and fault-tolerant control system has been supported by funding under the Irish Marine Institute and the Marine RTDI Measure, Productive Sector Operational Programme, National Development Plan 2000–2006 (PhD -05-004, INF-06-013 and IND-05-03); Science Foundation Ireland under Grant 06/CP/E007; HEA PRTL 3 (MSR3.2 project -Deep Ocean Habitat Mapping using ROV; HEA PRTL 4 Environment & Climate Change Impacts and Responses Project/ Environment Graduate Programme; an Enterprise Ireland Commercialisation Fund Technology Development 2007 projects—MPPT Ring (CFTD/07/IT/313, “Multi-Purpose Platform Technologies for Subsea Operations) and PULSE RT (CFTD/07/323, “Precision Underwater Accelerated Sonar Emulation in Real-Time) with ERDF funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Blanke, M.; Kinnaert, M.; Lunze, J.; Staroswiecki, M. *Diagnosis and Fault-Tolerant Control*; Springer: Berlin/Heidelberg, Germany, 2006; ISBN-13 978-3-540-35652-3.
2. Oppenheimer, M.; Doman, D.; Bolender, M. Control allocation. In *The Control Handbook, Control System Applications*, 2nd ed.; Levine, W.S., Ed.; CRC Press: Boca Raton, FL, USA, 2010; Chapter 8.
3. Fossen, T.I.; Johansen, T.A. A survey of control allocation methods for ships and underwater vehicles. In Proceedings of the 14th Mediterranean Conference on Control and Automation, Ancona, Italy, 28–30 June 2006.
4. Johansen, T.A.; Fossen, T.I. Control allocation—A survey. *Automatica* **2013**, *49*, 1087–1103. [[CrossRef](#)]
5. Zhang, Y.; Zeng, J.; Li, Y.; Sun, Y.; Wan, L.; Huang, S. Research on reconstructive fault-tolerant control of an X-rudder AUV. In Proceedings of the OCEANS 2016 MTS/IEEE Monterey, Monterey, CA, USA, 19–23 September 2016.
6. Chu, Z.; Luo, C.; Zhu, D. Adaptive fault-tolerant control for a class of remotely operated vehicles under thruster redundancy. In Proceedings of the IEEE 8th International Conference on Underwater System Technology: Theory and Applications (USYS), Wuhan, China, 1–3 December 2018. [[CrossRef](#)]
7. Liu, X.; Zhang, M.; Yao, F. Adaptive fault-tolerant control and thruster fault reconstruction for an autonomous underwater vehicle. *Ocean Eng.* **2018**, *155*, 10–23. [[CrossRef](#)]
8. Ropars, B.; Lasbouygues, A.; Lapierre, L.; Andreu, D. Thruster’s dead-zones compensation for the actuation system of an underwater vehicle. In Proceedings of the European Control Conference (ECC), Linz, Austria, 15–17 July 2015; pp. 741–746.
9. Baldini, A.; Ciabattini, L.; Felicetti, R.; Ferracuti, F.; Freddi, A.; and Monteriù, A. Dynamic surface fault-tolerant control for underwater remotely operated vehicles. *ISA Trans.* **2018**, *78*, 10–20. [[CrossRef](#)]
10. Chin, C.S.; Lau, M.W.S.; Low, E.; Seet, G.G.L. Design of Thrusters Configuration and Thrust Allocation Control for a Remotely Operated Vehicle. In Proceedings of the IEEE Conference on Robotics, Automation and Mechatronics, Bangkok, Thailand, 1–3 June 2006. [[CrossRef](#)]

11. De Carolis, V.; Maurelli, F.; Brown, K.E.; Lane, D.M. Energy-aware fault-mitigation architecture for underwater vehicles. *Auton. Robot.* **2017**, *41*, 1083–1105. [[CrossRef](#)]
12. Davoodi, M.; Meskin, N.; Khorasani, K. Event-triggered fault estimation and accommodation design for linear systems. In Proceedings of the Second International Conference on Event-based Control, Communication, and Signal Processing (EBCCSP), Krakow, Poland, 13–15 June 2016.
13. Karras, G.C.; Fourlas, G.K. Model Predictive Fault Tolerant Control for Omni-directional Mobile Robots. *J. Intell. Robot. Syst.* **2019**. [[CrossRef](#)]
14. Liu, H.; Wei, Y.; Zhou, X.; Li, G. Operated ROV Thrust Distribution Control System Based on Adaptive Back-stepping controller. In Proceedings of the 35th Chinese Control Conference, Chengdu, China, 27–29 July 2016.
15. Antonelli, G.; Fossen, T.I.; Yoerger, D.R. Underwater Robotics. In *Springer Handbook of Robotics*; Siciliano, B., Khatib, O., Eds.; Springer: Berlin/Heidelberg, Germany, 2008; pp. 987–1008.
16. Antonelli, G. *Underwater Robots*, 2nd ed.; STAR 2; Springer: Berlin/Heidelberg, Germany, 2006; pp. 79–91.
17. Omerdic, E.; Roberts, G.N. Thruster fault diagnosis and accommodation for open-frame underwater vehicles. *Control Eng. Pract.* **2004**, *12*, 1575–1598. [[CrossRef](#)]
18. Fossen, T.I. *Handbook of Marine Craft Hydrodynamics and Motion Control*; John Wiley & Sons: Hoboken, NJ, USA, 2011.
19. Härkegård, O. Backstepping and Control Allocation with Application to Flight Control. Ph.D. Thesis, Department of Electrical Engineering, Linköping University, Linköping, Sweden, 2003.
20. Omerdic, E. *Thruster Fault-Tolerant Control*; VDM Verlag: Riga, Latvia, 2009; ISBN 978-3-639-11739-4.
21. Bodson, M. Evaluation of Optimisation Methods for Control Allocation. *J. Guid. Control Dyn.* **2002**, *25*, 703–711. [[CrossRef](#)]
22. Burken, J.; Lu, P.; Wu, Z.; Bahm, C. Two Reconfigurable Flight-Control Design Methods: Robust Servomechanism and Control Allocation. *J. Guid. Control Dyn.* **2001**, *24*, 482–493. [[CrossRef](#)]
23. Omerdic, E.; Toal, D.; Nolan, S.; Ahmad, H. ROV LATIS: Next Generation Smart Underwater Vehicle. In *IET Book: Further Advances in Unmanned Marine Vehicles*; Inst of Engineering & Technology: London, UK, 2002; Chapter 2, ISBN-10 1849194793, ISBN-13 978-1849194792.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).