# Adaptive Fault-Tolerant Control Allocation Schemes for Overactuated Systems with Actuator and Bias Faults

Waseem Akram, Francesco Tedesco and Alessandro Casavola
*Department of Computer Engineering, Modelling, Electronics and Systems (DIMES), University of Calabria, Rende, Italy*

Keywords: Control Allocation, Fault-Tolerant, Actuator Redundancy, Bias Fault, Marine Surface Vehicle.

Abstract: Fault-Tolerant control is of paramount importance in marine technology, especially for autonomously guided vehicles. It can be achieved by exploiting actuators redundancy, which adds flexibility to the system by guaranteeing maneuverability, even in the presence of actuator faults. The main idea in the control allocation scheme here proposed is at distributing the control effort among the remaining healthy actuators without changing the nominal control law. In this paper, we propose an enhanced adaptive control allocation algorithm for over actuated systems. The proposed algorithm works under actuators loss of effectiveness with possible thruster stuck situations under both input saturation and rate of change constraints. The effectiveness of the proposed scheme is shown by simulating a marine surface vehicle model on a path-following problem.

## 1 INTRODUCTION

The success of autonomous marine vehicle missions under actuator faults strongly depends on the effectiveness of control reconfiguration strategies. Such strategies are concerned with diving, hovering and safe movements of the vehicles under different circumstances, possibly in the case of faults or failures. Usually, actuator faults occur due to seaweeds or ropes that get stuck in the thruster or failures in the actuator power unit (He et al., 2012). Thus, fault tolerance is a key issue in the development of control strategies for autonomous marine vehicles (Khan et al., 2018).

The actuator redundancy concept is widely used in many control schemes to achieve system faults tolerability. In the design and implementation of resilient control strategies, control allocation algorithms are used to manage and distribute the control signals among redundant actuators, by using the degree of freedom provided by redundancy to accomodate faulty thrusters and increasing the system maneuverability, flexibility and safety.

In the past few decades, many scientific studies have been accomplished on fault-tolerant control schemes. Some of the key examples are briefly reviewed next. Authors of (Wang et al., 2015) worked on fault control and reconfiguration schemes in the presence of uncertainty, environmental disturbances, and thruster faults. Their study proposes the use of a sliding mode algorithm and a back-stepping technique. Authors of (Tohidi et al., 2017) proposed an adaptive control allocation scheme for over-actuated systems. In this work, actuator loss is dealt without estimating the control input matrix. Another adaptive fault-tolerant control allocation approach is proposed in (Casavola and Garone, 2010), where the authors consider over-actuated systems under actuator faults. In this work, the online parameter estimation algorithm is integrated with the control allocation algorithm. Authors of (Liu et al., 2018) studied the adaptive fault-tolerant control scheme for autonomous underwater vehicles under environmental disturbance and uncertainty. In this work, a closed-loop system is considered by using an adaptive fault-tolerant control scheme. The system deals with tracking problems. Authors of (Ismail et al., 2014) studied the fault-tolerant control of a kinematical redundancy thruster structure for autonomous underwater vehicles. In this work, the method is divided into thruster force allocation and control design for tracking problems. The redundant thruster concept is adopted for accomodate faulty thrusters in path-following control problems.

The authors of (Tohidi et al., 2016) proposed an adaptive correction approach for fault-tolerant control of autonomous underwater vehicles. In this work, the method does not require any estimation of the input matrix. The adaptive control is responsible to find control allocation parameters. Additionally, a sliding



Akram, W., Tedesco, F. and Casavola, A.
Adaptive Fault-Tolerant Control Allocation Schemes for Overactuated Systems with Actuator and Bias Faults.
DOI: 10.5220/0009894800810088
In *Proceedings of the 17th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2020)*, pages 81-88
ISBN: 978-989-758-442-8

mode controller is also used that provides system stability. In the study (Chu et al., 2018), authors studied diving movement and proposed an adaptive fuzzy sliding mode controller for autonomous underwater vehicles. In this work, the concept of partial saturation of the rudder angle is adopted. A partially known input gain was used and a single fuzzy logic was designed for online parameter estimation. Authors of (Zhang et al., 2017) worked on path tracking of fault tolerant control problems for the autonomous underwater vehicle by using the backstepping technique. The technique is capable to drive the vehicle under ocean current, unknown faults and rate constraints. Similarly, the utility of fault-tolerant control schemes has also been reported in (Patel and Shah, 2018; Wang and Zhang, 2017).

In this paper, we propose a new control allocation method that is capable to tolerate system faults for vehicles with actuator redundancy. The general architecture of control allocation scheme is shown in Figure 1 where it is assumed that the control law has been designed on the basis of a virtual system with a minimal number of inputs $v(t)$. Then, a control allocation unit is in charge of reallocating the desired control effort related to $v(t)$ among the physical actuators $u(t)$ on the basis of their current status. Here the allocation unit is designed by resorting to ideas presented in (Casavola and Garone, 2010). Anyway, in the current work we use additional actuator rate constraints and stuck (bias) faults. The online effectiveness matrix and the bias fault estimation are integrated with the control allocation algorithm. The proposed algorithm is capable to successfully perform the control allocation task under actuator rate constraints and stuck faults. Simplicity, accuracy, low computational cost and system stability are the main characteristics of the proposed control allocation method.

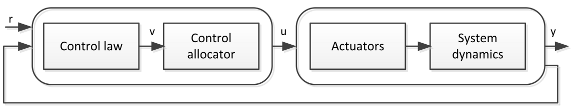


Figure 1: Modular structure of control and control allocation scheme.

## 2 PROBLEM STATEMENT

Let us consider the plant described by the following discrete-time state-space equation.

$$x(t+1) = A(x(t)) + B_u(x(t))u(t) \tag{1}$$

where $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$ are the system states and control input. $A(x(t))$ and $B_u(x(t))$ are nonlinear state-dependent matrices. It is assumed that the system has redundant actuators, thus control matrix is rank deficient and can be written as $rank(B_u((x(t)))) = r < m$ for all $x \in \mathbb{R}^n$. Moreover, the control inputs are subject to amplitude and rate constraints, that is:

$$u(t) \in \Omega(\tilde{u}) := \{u \in \mathbb{R}^m | u^- \leq u(t) \leq u^+, |u - \tilde{u}| \leq \bar{\Delta}u,\} \tag{2}$$

where $u^+ := [u_1^+, u_2^+, ..., u_m^+]^T$, $u^- := [u_1^-, u_2^-, ..., u_m^-]^T$, and $\bar{\Delta}u := [\bar{\Delta}u_1, \bar{\Delta}u_2, ..., \bar{\Delta}u_m]^T$ .

According to the definition of the control matrix, the system in (1) can be represented in an equivalent form as following:

$$x(t+1) = A(x(t)) + B_v(x(t))v(t) \tag{3}$$

$$B_v(x(t))v(t) = B(x(t))u(t) \tag{4}$$

where $B_v(x(t)) \in \mathbb{R}^{nxk}$ is a full column rank matrix, $v(t) \in \mathbb{R}^k$ is a virtual input to control the model. The virtual control input is the desired total effort that one wants to apply to the system. The system given in (3) is called the virtual plant and (4) is called the parity equation. This representation of the system shows the relationship between virtual and physical inputs.

In the sequel it is assumed that the system is subject to actuator faults and can be rewritten as:

$$x(t+1) = A(x(t)) + B_u(x(t))\Delta(t)u(t) + B_u(x(t))f(t) \tag{5}$$

where $f(t)$ represents possible bias faults and

$$\Delta(t) = diag\{\delta_1(t), \delta_2(t), ..., \delta_m(t)\}, 0 \leq \delta_i(t) \leq 1 \tag{6}$$

is called the Effectiveness matrix. If $\delta_i(t) = 1$,the actuator is working perfectly, if $\delta_i(t) < 1$,the actuator is faulty, and if $\delta_i(t) = 0$, then the actuator is failed. The overall control allocation problem can be stated as follows.

**Fault-Tolerant Control Allocation Problem (F-TCAP).** *Given the plant* (3) *and the virtual input* $v(t)$, *compute at each time instant* $t \geq 0$ *the physical input* $u(t)$ *such that*:

- the input constraints are satisfied

$$u(t) \in \Omega(u(t-1)) \tag{7}$$

- control allocation is successfully performed regardless of possible faults occurrences, i.e.

$$B_v(x(t))v(t) = B_u(\Delta(t)u(t) + f(t)) \tag{8}$$

**Remark.** The previous F-TCAP (Casavola and Garone, 2010) was not able to tolerate the actuator rate constraints and bias faults during the operation. In this paper we generalize the scheme to accommodate such new requirements.

# 3 PROPOSED SCHEME

In this section the above state problem is solved by means of the following a **three-step** procedure performed at each time instant $t$:

1. Estimate the contribution of the fault term $\hat{f}(t)$ by exploiting the current state measurement $x(t)$
2. Compute the diagonal matrix $\hat{\Delta}(t)$ as the best estimation of $\Delta(t)$ on the basis of previous state measurements and applied commands
3. Solve a control allocation problem by means of equation (8) by assuming (*certainty equivalence hypothesis*) $\Delta(t) = \hat{\Delta}(t)$ and $f(t) = \hat{f}(t)$

The first step is easily accomplished as $x(t)$, $A(x(t-1))$, $u(t-1)$ and $B_u(x(t-1))\hat{\Delta}(t-1)u(t-1)$ are known quantities. Then the computation of $\hat{f}(t)$ is performed as follows:

$$\begin{array}{rcl} B_u(x(t-1))\hat{f}(t-1) & = & x(t) - A(x(t-1)) \\ & - & B_u(x(t-1))\hat{\Delta}(t-1)u(t-1) \end{array} \tag{9}$$

The second step is performed by exploiting the idea of (Casavola and Garone, 2010) where a moving time-windowed least-squared parameter estimation scheme was introduced. It is based on the following optimization problem:

$$\hat{s}_i(t) \triangleq \arg\min \sum_{i=1}^{N} \|s_i\|^2 Q_i + \|vect(\Gamma)\|^2 R \tag{10}$$

subject to the following condition:

$$\begin{array}{r} x(t-i+1) - A(x(t-i)) - B_u(x(t-i))[\Gamma + \hat{\Delta}(t-1)] \\ u(t-i) - B_u(x(t-i))\hat{f}(t-i) = s_i, i = 1,..,N. \end{array} \tag{11}$$

where $Q_i >> R$ are weighting matrices, $s_i$ is a parity slack vector, and $\hat{\Gamma}(t) = \hat{\Delta}(t) - \hat{\Delta}(t-1)$ is $\hat{\Delta}(t)$ in incremental order.

The matrix $\hat{\Gamma}(t)$ is defined as:

$$\hat{\Gamma}(t) \triangleq diag\{\hat{\gamma}_1, \hat{\gamma}_2, .., \hat{\gamma}_m\} \in \mathbb{R}^{m\times n} \tag{12}$$

which is a diagonal matrix of actuator effectiveness loss.

Finally the third step is carried out by completing the control allocation task. In particular the following optimization problem is solved:

$$\begin{array}{r} u(t) \triangleq \arg\min\limits_{s,u} \|s\|^2 Q_s + \|u\|^2 R_u, \\ B_v(x(t))v(t) = B_u(x(t))\hat{\Delta}(t)u + B_u(x(t))\hat{f}(t), \\ u \in \Omega(u(t-1)), \end{array} \tag{13}$$

The whole scheme can be summarized in the following algorithm:

Algorithm 1: F-TCAP.

Initialization:

1: **set:** $x(0)$, $v(0)$, $\Delta(0)$, $f(0)$
2: **choose:** window horizon N for the parameter estimation
3: **store:** $u(0)$, $x(0)$, $v(0)$, $f(0)$, $\Delta(0)$

Online-Phase

1: **for** $t > 0$ **do**
2: **compute:** $B_u(x(t-1))\hat{f}(t-1)$ as in (9)
3: **get:** virtual input $v(t)$ from the controller
4: **compute:** $\hat{\Delta}(t)$ by solving (10)
5: **compute:** $u(t)$ as in (13)
6: **apply** $u(t)$

# 4 ILLUSTRATIVE EXAMPLES

## 4.1 Linear Unstable Model

In this section, we consider a linear unstable model in order to show the applicability and effectiveness of the proposed scheme. The model is taken from (Casavola and Garone, 2010). Let consider a state-space model as following:

$$x(t+1) = Ax(t) + B_u u(t) \tag{14}$$

where $x \in \mathbb{R}$ is the state vector and $u = u_1, u_2, u_3 \in \mathbb{R}^3$ physical input vector subject to the following constraints:

$$\begin{array}{rccl} -5 & \leq & u_i(t) & \leq 5 \\ -0.9 & \leq & u_i(t) - u_i(t-1) & \leq 0.9 \end{array} \tag{15}$$

Moreover: $A = 1.2$ and $Bu = [1, 1, 1]$ are considered. The actuator and bias faults occurrences are assumed as:

$$\begin{array}{l} \Delta(t) = diag\{1,1,1\} \text{ for } t < 12_s \\ \Delta(t) = diag\{0.9,1,1\} \text{ for } 12_s \leq t \leq 200_s \\ \Delta(t) = diag\{0.5,1,0\} \text{ for } t > 200_s \end{array} \tag{16}$$

$$\begin{array}{l} f(t) = [0,0,0]^T \text{ for } t < 12_s \\ f(t) = [0.01,0,0]^T \text{ for } 12_s \leq t \leq 200_s \\ f(t) = [0,0,0.01]^T \text{ for } t > 200_s \end{array} \tag{17}$$

The actuator faults consist of two fault sequence. The first fault is occurring at $12_s \leq t \leq 200_s$ when a partial fault occurs at the first actuator. This is followed by a reduction of 50% effectiveness at first actuator after

$t > 200_s$. In particular, please note that after $t > 200$, the third actuator gets stuck until the simulation.

The virtual input matrix $B_v = 1$ is used and the virtual signals are built as $v(t) = Kx(t) + K_r r(t)$, where $r(t)$ is a constant reference signal to be tracked, $K$ is such that $(A + B_v K)$ is a Schur matrix, and $K_r = ((I - A - B_v K)^{-1} B_v)^{-1}$. In the feedback control law, the gain $K = -0.6$ is used. The model is simulated under the CAP and F-TCAP algorithms.

In Figure 2, the virtual signals are shown in three different cases i.e No fault, CAP, and F-TCAP. This shows that the control law is not disturbed by the faulty actuator. In Figure 3, the position vector is plotted. Here, it can be noticed that the F-TCAP algorithm successfully tracks the reference position under actuator and bias faulty events. It is observed that a small perturbation is generated in the case of F-TCAP, while the behavior of the CAP method is highly disturbed by faulty events. Figure 4 reports the physical inputs. This shows that the signals of faulty actuator smartly changed- all signals are distributed among healthy actuators after the actuator got stuck. In last, Figure 5 shows the loss of effectiveness parameters. Notice in this viewpoint, the F-TCAP algorithm tolerates the fault better than the CAP algorithm.

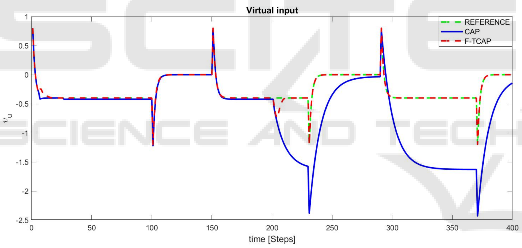


Figure 2: Virtual inputs.

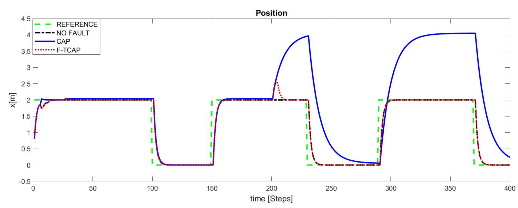


Figure 3: Position.

## 4.2 Marine Surface Vehicle

In this section, we consider the surface marine vehicle depicted in Figure 6. The model and the control structure are taken from (Folino, 2018; D'Angelo, 2018). There, the system position is denoted by $\eta = [x, y, \theta]^T$, where $x$ and $y$ are earth-fixed positions and $\theta$ is the yaw angle. The body-fixed velocities are denoted with $\rho = [u, v, r]^T$ where $u$ is forward velocity, $v$ is lateral velocity, and $r$ is yaw angular velocity. The thrust and force moment of the vehicle are represented by $\upsilon = [\upsilon_u, \upsilon_v, \upsilon_r]^T$ and are generated via the thrust vector $\tau = [\tau_1, \tau_2, \tau_3, \tau_4]^T$ as depicted in Figure 7. More formally the kinematic model of the system is given

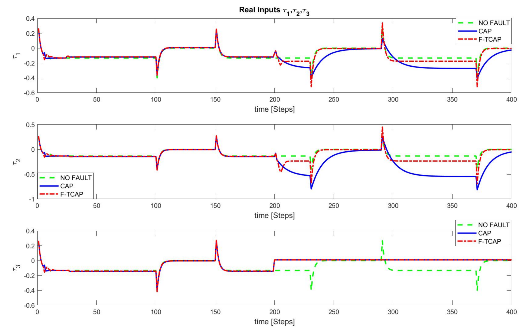


Figure 4: Physical inputs: $\tau_i = u_i + f_i$.

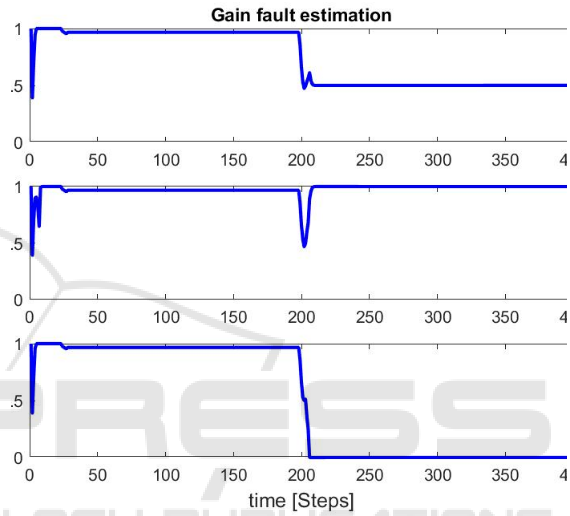


Figure 5: Fault loss of effectiveness profiles.

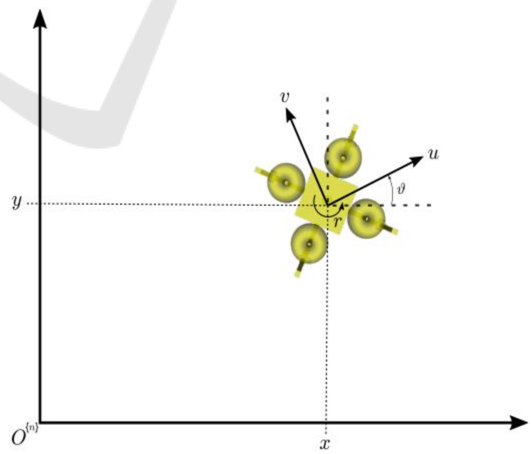


Figure 6: Schematic model of the ship.

Table 1: The system matrices.

| | |
|---|---|
| $M$ | Inertia matrix |
| $C$ | Coriolis/centripetal matrix |
| $D$ | Hydrodynamics damping matrix |

Table 2: Parameters definitions.

| | |
|---|---|
| $m = 1.1274$ [Kg] | Mass |
| $I_z = 1$[Kg m$^2$] | moment of Inertia |
| $\beta_u = 0.0414$[Kg/m] | friction force coefficient along $u$ |
| $\beta_v = 0.0414$[Kg/m] | friction force coefficient along $v$ |
| $\beta_r = 0.0568$[Kg m$^2$] | friction torque coefficient along $r$ |
| $\alpha = \pi/4$ | thruster angle in the body frame |

as follows:

$$\dot{\eta}(t) = R(\theta(t))\rho(t) \tag{18}$$

where $R(\theta)$ is the rotational matrix around the yaw angle defined as

$$R(\theta) := \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

while the dynamic model is derived as:

$$M\dot{\rho}(t) + C(\rho(t))\rho(t) + D(\rho(t))\rho(t) = \upsilon(t) \tag{19}$$

where the system matrices, whose meaning and numerical parameters are reported in Tables 1-2, take the following form:

$$M := \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I_z \end{bmatrix} C(\rho) := \begin{bmatrix} 0 & -mr & 0 \\ mr & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$D(\rho) := \begin{bmatrix} \beta_u u & 0 & 0 \\ 0 & \beta_v v & 0 \\ 0 & 0 & \beta_r r \end{bmatrix}$$

Such a model can be recast as:

$$\dot{\rho}(t) = A(\rho(t)) + \hat{B}_v \upsilon(t) \tag{20}$$

that is more similar to (3) although in continuous-time domain. Here the matrix $A(\rho(t)) := -M^{-1}(C(\rho(t)) + D(\rho(t))\rho(t)$, while $\hat{B}_v := M^{-1}$. Thus,the discrete-time state-space representation (3) takes the form:

$$\begin{bmatrix} \eta(t+1) \\ \rho(t+1) \end{bmatrix} = \begin{bmatrix} I & T_s R(\theta(t)) \\ 0 & I + T_s M^{-1}(C(\rho(t)) + D(\rho(t)) \end{bmatrix} \begin{bmatrix} \eta(t) \\ \rho(t) \end{bmatrix} + \begin{bmatrix} 0 \\ B_v \end{bmatrix} \upsilon(t)$$

where $T_s$ is the sampling period and $B_v = T_s \hat{B}_v$.

In particular input $\upsilon$ is produced by the 4 thrusters according to the following allocation law

$$\upsilon(t) = B_u \tau(t) \tag{21}$$

where the allocation matrix $B_u$ is defined as:

$$B_u := \begin{bmatrix} \sin(\alpha) & \sin(\alpha) & -\sin(\alpha) & -\sin(\alpha) \\ \cos(\alpha) & -\cos(\alpha) & \cos(\alpha) & -\cos(\alpha) \\ -l/2 & l/2 & l/2 & -l/2 \end{bmatrix} \tag{22}$$

In (21) $\tau_1$ and $\tau_2$ denote the signals for two identical main propellers and $\tau_3$ and $\tau_4$ are the control signals for the transverse thrusters. In this respect $\tau$ represents the signal generated by the actuators on the basis of the physical input $u(t) = [u_1, u_2, u_3, u_4]^T$ produced by the allocation unit according to the following expressions that accounts also for possible faulty events

$$\tau(t) = \Delta(t)u(t) + f(t) \tag{23}$$

where $\Delta \in \mathbb{R}^{4\times4}$ represents the control effectiveness matrix and $f(t)$ models bias faults.

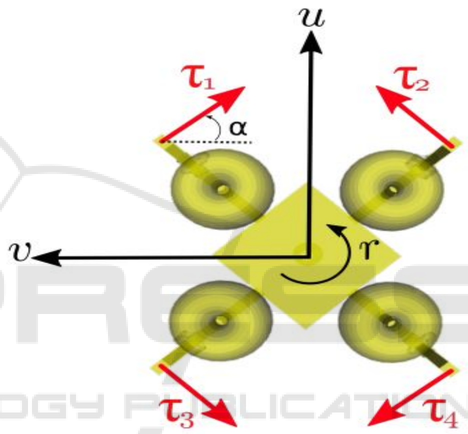


Figure 7: Thrust forces allocation.

In view of the previous discussion, the signal $\upsilon$ can be considered as the virtual control signal. It is determined by the following time-invariant control law

$$\upsilon(t) = K_\rho[R^T(\theta(t))K_\eta(\eta_d(t) - \eta(t)) - \rho(t)] \tag{24}$$

where the controller gains are $K_\rho = 0.75 I_{3\times3}$, $K_\eta = 0.63 I_{3\times3}$ and $\eta_d$(t) is the desired trajectory to track. The sampling period for both the control law and the control allocator is $T_s = 0.2827$.

The goal of the first simulation campaign is to successfully allocate virtual law (24) with desired reference $\eta_d$ evolving as

$$\eta_d = \begin{cases} [0,0,0]^T \text{for } t < 12_s \\ [2,0,0]^T \text{for } 12_s \le t \le 200_s \\ [2,2,\pi]^T \text{for } 200_s < t \le 500_s \\ [0,0,0]^T \text{for } t > 500_s \end{cases} \tag{25}$$

under the constraints

$$-0.1 \le u_i \le 0.1 \text{and} -0.09 \le u_i - u_i(t-1) \le 0.09 \tag{26}$$

and despite the matrix effectiveness and bias actuator faults assumed as:

$$\begin{aligned} \Delta(t) &= diag\{1,1,1,1\} \text{for } t < 12_s \\ \Delta(t) &= diag\{0.9,1,1,1\} \text{for } 12_s \le t \le 200_s \\ \Delta(t) &= diag\{0.8,1,0,1\} \text{for } t > 200_s \end{aligned} \quad (27)$$

$$\begin{aligned} f(t) &= [0,0,0,0]^T \text{for } t < 12_s \\ f(t) &= [0.01,0,0,0]^T \text{for } 12_s \le t \le 200_s \\ f(t) &= [0,0,0.01,0]^T \text{for } t > 200_s \end{aligned} \quad (28)$$

In particular note that after $t > 200_s$, the third actuator gets stuck until the simulation.

In order to show the effectiveness of the proposed scheme, we have simulated the model under the following scenarios:

- No Fault: the model is simulated without considering any fault.
- CAP: the algorithm is implemented without calculation of effectiveness matrix and bias fault estimation.
- F-TCAP: the algorithm is simulated with the calculation of effectiveness matrix and bias faults estimation.

The results of this first simulation are reported in Figures 8-12. In particular Figure 8 shows the total virtual control input needed to drive the vehicle. It is observed that the virtual control inputs are successfully tracked and the system remains bounded with smooth variations by using the F-TCAP algorithm. Figures 9 and 10 show the positions of the vehicle by using CAP and F-TCAP algorithm. The simulation results demonstrate that the variables continue to track their desired values after a fault occurs by using the F-TCAP algorithm while the CAP is not working correctly. It is observed that the system remains stable under faulty conditions with F-TCAP. The physical inputs are plotted in Figure 11. The proposed algorithm successfully allocates the control efforts to the vehicle under faulty scenarios and distributes the control to the healthy actuators. Figure 12 gain fault estimation. Here, the loss of effectiveness parameters is estimated and plotted. The system shows a small perturbation when the fault occurs. It is observed that the proposed algorithm performed better in case of reconfiguration and control allocation as compared to the CAP algorithm under actuator faults, rate constraints and bias faults.

A second simulation example has been performed to observe the behaviour of the CAP and F-TCAP algorithms when the vehicle fixed is instructed to keep position at a fixed point when an actuator is stuck. In

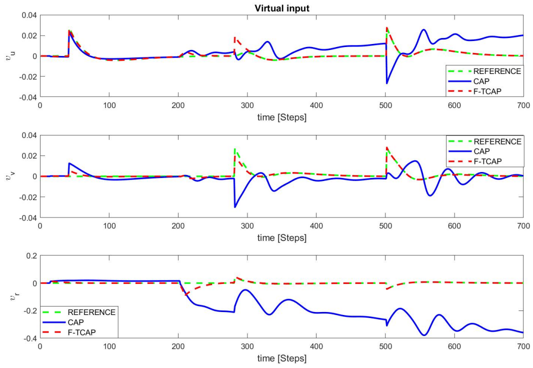


Figure 8: Example 1: Virtual inputs.

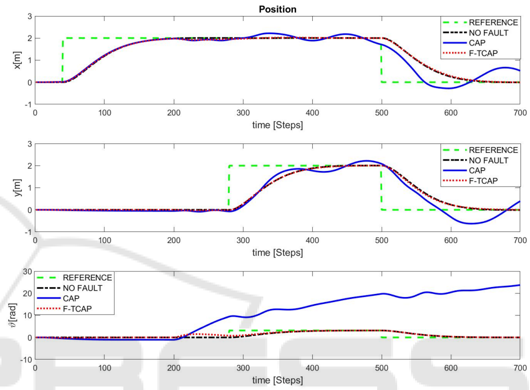


Figure 9: Example 1: Vehicle positions and angle.

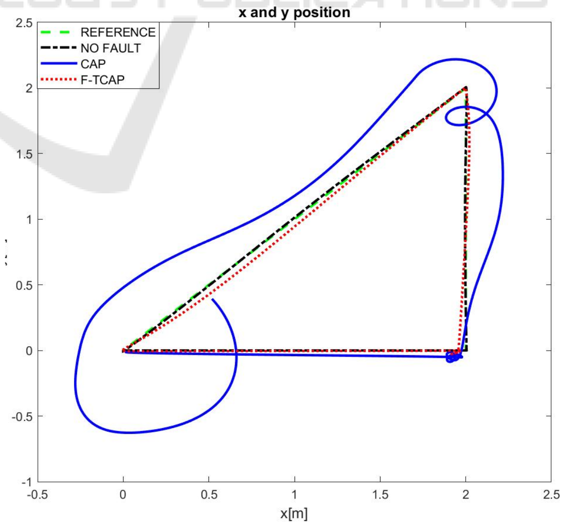


Figure 10: Example 1: x and y position of the vehicle.

this example, we assumed a reference path position as $\eta_d(t) = [0,0,0]^T, \forall t \ge 0$. During the simulation, the actuator fault $\Delta(t) = [1,0,1,1]^T$ and bias fault as $f(t) = [0,0.8,0,0]^T$ is imposed on the model in the

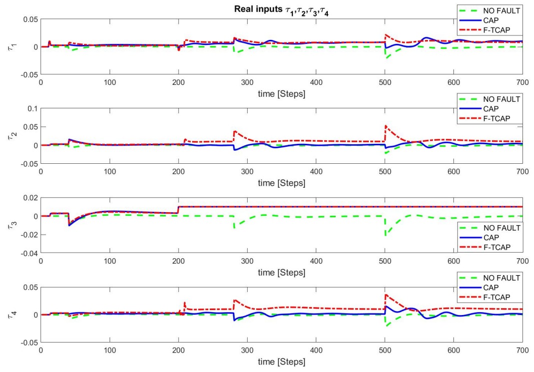


Figure 11: Example 1: Physical inputs.

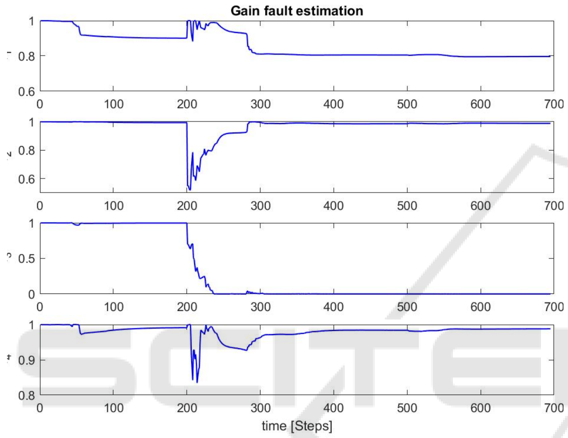


Figure 12: Example 1: Fault loss of effectiveness profiles.

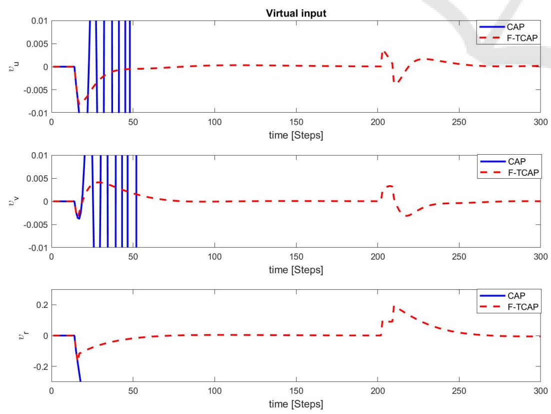


Figure 13: Example 2: Virtual inputs.

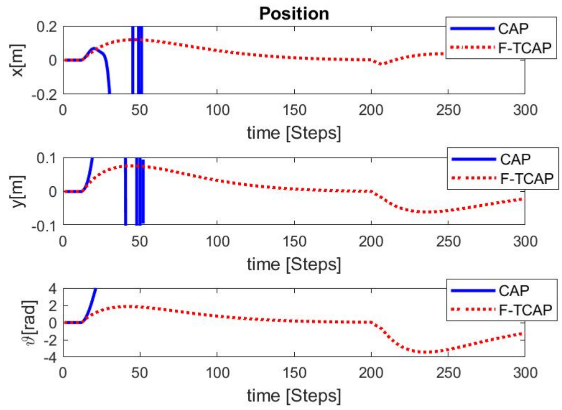


Figure 14: Example 2: Vehicle positions and angle.

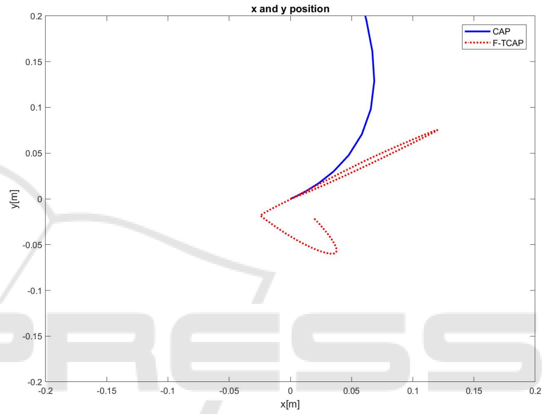


Figure 15: Example 2: x and y position of the vehicle.

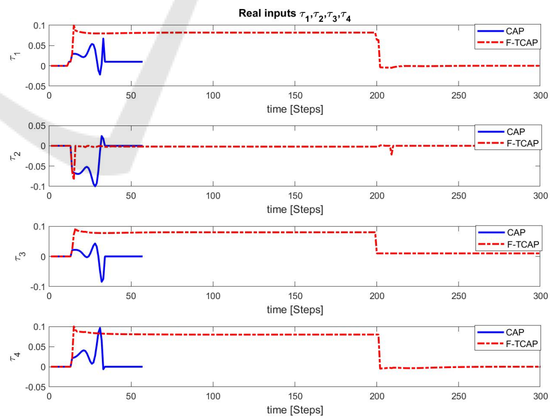


Figure 16: Example 2: Physical inputs.



time range $12_s \leq t \leq 200_s$. Figure 13 shows the virtual control input that is used to generate the physical inputs as shown in Figure 16. The position of the vehicle is shown in Figure 14 and 15. It can be noted that under CAP the vehicle position goes to infinity. On the contrary, under F-TCAP the vehicle remains close to the reference position thanks to the action of the allocation unit. This depicts that the vehicle is stable and input signals are bounded by using the F-TCAP algorithm under actuator and bias faults.

## 5 CONCLUSIONS

In this paper, a new adaptive fault-tolerant control allocation algorithm is proposed. The proposed algorithm integrates online effectiveness matrix and bias faults estimation with the control allocation algorithm. The actuator and bias faults are considered under actuator amplitude and rate constraints. The effectiveness of the proposal is shown by using a marine surface vehicle model where the proposed algorithm showed good performance in terms of control allocation. Future works foresees the implementation of this class of adaptive control allocation strategies on real redundant marine vehicles.

## ACKNOWLEDGEMENTS

The authors want to thank Ing. Paolo Folino and Ing. Vincenzo D'Angelo by AppliCon S.r.l for providing us with the mathematical model of the autonomous marine vehicle they developed in their master's theses and for useful discussions and assistance in their vehicle control and guidance aspects.

## REFERENCES

Casavola, A. and Garone, E. (2010). Fault-tolerant adaptive control allocation schemes for overactuated systems. *International Journal of Robust and Nonlinear Control*, 20(17):1958–1980.

Chu, Z., Xiang, X., Zhu, D., Luo, C., and Xie, D. (2018). Adaptive fuzzy sliding mode diving control for autonomous underwater vehicle with input constraint. *International Journal of Fuzzy Systems*, 20(5):1460–1469.

D'Angelo, V. (2018). Strategie di gestione delle missioni e controllo di un veicolo marino autonomo di superficie. Master's thesis, University of Calabria. Master's thesis in Automation Engineering.

Folino, P. (2018). Strategie di guida e navigazione per un veicolo marino autonomo di superficie. Master's thesis, University of Calabria. Master's thesis in Automation Engineering.

He, C., Feng, Z., and Ren, Z. (2012). Flocking of multi-agents based on consensus protocol and pinning control. In *Proceedings of the 10th World Congress on Intelligent Control and Automation*, pages 1311–1316. IEEE.

Ismail, Z. H., Faudzi, A., and Dunnigan, M. W. (2014). Fault-tolerant region-based control of an underwater vehicle with kinematically redundant thrusters. *Mathematical Problems in Engineering*, 2014.

Khan, H. Z. I., Rajput, J., Ahmed, S., Sarmad, M., and Sharjil, M. (2018). Robust control of overactuated autonomous underwater vehicle. In *2018 15th International Bhurban Conference on Applied Sciences and Technology (IBCAST)*, pages 269–275. IEEE.

Liu, X., Zhang, M., and Yao, F. (2018). Adaptive fault tolerant control and thruster fault reconstruction for autonomous underwater vehicle. *Ocean Engineering*, 155:10–23.

Patel, H. R. and Shah, V. A. (2018). A framework for fault-tolerant control for an interacting and non-interacting level control system using ai. In *ICINCO (1)*, pages 190–200.

Tohidi, S. S., Yildiz, Y., and Kolmanovsky, I. (2016). Fault tolerant control for over-actuated systems: An adaptive correction approach. In *2016 American Control Conference (ACC)*, pages 2530–2535. IEEE.

Tohidi, S. S., Yildiz, Y., and Kolmanovsky, I. (2017). Adaptive control allocation for over-actuated systems with actuator saturation. *IFAc-PapersOnLine*, 50(1):5492–5497.

Wang, B. and Zhang, Y. (2017). An adaptive fault-tolerant sliding mode control allocation scheme for multirotor helicopter subject to simultaneous actuator faults. *IEEE Transactions on Industrial Electronics*, 65(5):4227–4236.

Wang, Y., Wilson, P. A., Liu, X., et al. (2015). Adaptive neural network-based backstepping fault tolerant control for underwater vehicles with thruster fault. *Ocean Engineering*, 110:15–24.

Zhang, M., Liu, X., and Wang, F. (2017). Backstepping based adaptive region tracking fault tolerant control for autonomous underwater vehicles. *The Journal of Navigation*, 70(1):184–204.