

Adaptive Fault Tolerant Control Allocation Strategies for Autonomous Overactuated Vehicles

Alessandro Casavola* Emanuele Garone**

* *casavola@deis.unical.it - DEIS, University of Calabria*

** *egarone@deis.unical.it - DEIS, University of Calabria*

Abstract: This paper presents a preliminary adaptive control allocation scheme for overactuated autonomous vehicles that is fault-tolerant with respect to actuator faults or loss of effectiveness. The main idea here is to use an ad-hoc online parameter estimator coupled with an allocation algorithm to perform on-line control reconfiguration. A simple algorithm is proposed for nonlinear discrete-time systems and its main properties are summarized for the disturbance-free case. Its effectiveness is also investigated by means of numerical simulations on an underwater vehicle.

Keywords: Control Allocation, Adaptive Control Allocation, Control Reconfiguration, Autonomous Overactuated Vehicles, Underwater Vehicles

1. INTRODUCTION

Actuators and sensors redundancy is an important issue to deal with for increasing the fault-tolerance of autonomous vehicles. It is a very common matter in airspace and underwater applications where, due to safety and performance reasons, a bank of redundant actuators and sensors is often used. Moreover, it is also common in gas, water and electrical distribution and production networks, where such a physical redundancy may be exploited to avoid interruption of services.

In this paper we focus on the problem of control allocation for overactuated systems (i.e. systems with physical actuator redundancy). One traditional way to solve it is to use multivariable optimal control design methods (Kwakernaak, 1972). Such an approach achieves both regulation and control allocation at the same time. A different approach consists of using a simpler control law that specifies the total control effort that has to be produced and in separately solving the so-called Control Allocation Problem (CAP) i.e. the one of optimally distributing the desired total control effort over the available actuators. Due to its relevance, especially in flight control systems, CAP has been deeply investigated in the last decade and several methods have been proposed: Daisy Chaining (Buffington, Enn, 1996), Direct Control Allocation (Durham, 1993), Convex Optimization Based algorithms (Durahm, 1998; Bošković et al., 2002; Härkegård, 2002; Petersen, Fossen, 2005; Johansen et al., 2005) and PseudoInverse-Redistribution (PIR) methods (Bodson, 2002; Jin, 2005).

A key necessity in the development of overactuated vehicular systems is to dispose of methods which exploit such a physical redundancy in the design of effective recon-

figurative control strategies capable to avoid, or at least mitigate, the effects of actuator failures. A popular way to ensure some level of control reconfiguration is the use of adaptive control laws (Bodson, Groszkiewicz, 1997; Tao et al., 2002). An alternative approach that will be investigated here is the so-called Reconfigurable Control Allocation (RCA) problem (Buffington et al., 1998; Bolender, Doman, 2005). The key idea is depicted in Fig. 1 where supposedly the control law has been designed on the basis of a virtual system with a minimal number of inputs $v(t)$, fully equivalent to the physical inputs $u(t)$ in generating a desired total control effort. Then, an allocation unit distributes at each time t the control $v(t)$ on the physical actuators $u(t)$, with a law that is allowed to be time-variant. Then, in the case of actuator faults, control reconfiguration is possible in many cases without altering the control law by simply modifying the distribution of the total control effort $v(t)$ to the remaining no-faulty actuators in $u(t)$. This does not perturb in principle the closed-loop system dynamics because there are several way to distribute the control amongst actuators, all of them making the system behaving in the same way.

In this paper a preliminary adaptive control allocation scheme is proposed able at solving RCA problems for non-linear discrete-time systems. Such systems are subject to actuator faults or loss of effectiveness. Unlike other works on the topic, here the algorithm is not assumed to know the occurrence of a fault. On the contrary, an adaptive mechanism is used to estimate possible loss of effectiveness and make possible the on-line computation of the allocation rules by solving a standard constrained QP problem. The main properties of the scheme are summarized. For simplicity all developments are done in a disturbance-free scenario without considering model uncertainty. All issues related to the robustness properties of the algorithm are demanded to future studies.

¹ This work has been supported by MIUR Project *Fault Detection and Diagnosis, Control Reconfiguration and Performance Monitoring in Industrial Process*.

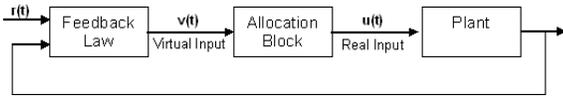


Fig. 1. Control structure with actuators allocation, sensors reconciliation and control computation performed separately

The paper is organized as follows: the problem is stated in Section II. In Section III an adaptive allocation scheme is presented and its properties are summarized. Finally, a numerical example dealing with a nonlinear model of an underwater autonomous vehicle is detailed in Section IV and some conclusions end the paper.

2. PROBLEM STATEMENT

2.1 Control Allocation Problem

Let us consider plants whose dynamics is described by the following nonlinear discrete-time state space equation

$$x(t+1) = a(x) + B_u(x)u(t), \quad (1)$$

where $x \in \mathcal{R}^n$ is the state vector and $u(t) \in \mathcal{R}^m$ the control input; $a(x) \in \mathcal{R}^n$ and $B_u(x) \in \mathcal{R}^{n \times m}$ are nonlinear state-dependent functions. The following assumptions are considered

- 1) The matrix $B_u(x)$ is column-rank deficient:
 $Rank(B_u(x)) = k < m, \forall x$;
- 2) The input signal $u(t)$ lies into a compact set Ω , i.e.

$$u(t) \in \Omega := \{u \in \mathcal{R}^m \mid u^- \leq u \leq u^+\}, \quad (2)$$

where $u^- := [u_1^-, u_2^-, \dots, u_m^-]^T \in \mathcal{R}^m$ and $u^+ := [u_1^+, u_2^+, \dots, u_m^+]^T \in \mathcal{R}^m$.

The assumption 1) allows one to define an equivalent representation of the plant (1)

$$x(t+1) = a(x) + B_v(x)v(t), \quad (3)$$

$$B_v(x)v(t) = B_u(x)u(t), \quad (4)$$

where $B_v(x) \in \mathcal{R}^{n \times k}$ is a full column-rank matrix such that its columns are a basis for the subspace defined by the columns of $B_u(x)$ and $v(x) \in \mathcal{R}^k$ is the virtual control input. Hereafter, the state space equation (3) will be referred to as the *virtual plant* while (4) is the *parity equation* of the system, which defines the analytical relationship between the virtual and applied commands.

Note that, in such a scheme, the virtual control input $v(t)$ represents the *desired total control effort* we want to apply to the plant. In the sequel, we will assume that such a signal $v(t)$ is provided at each time instant by the control law. On the basis of the overall system description (3)-(4) under the actuator constraints (2), the following problem can be stated:

Control allocation problem (CAP) - *Given a virtual input $v(t) \in \mathcal{R}^k$ compute a physical input $u(t) \in \mathcal{R}^m$ such that (2) and (4) are satisfied.*

Such a problem has been extensively studied in recent years and several numerical procedures for its solution have been proposed (Bodson, 2002; Johansen et al., 2005). Note that:

- Many previous works on the topic re-arrange the equation (4) as follows

$$\begin{aligned} v(t) &= B(x)u(t) \\ B_u(x) &= B_v(x)B(x) \end{aligned} \quad (5)$$

where $B(x) \in \mathcal{R}^{k \times m}$ is a factorization of $B_u(x)$

$$B(x) = (B_v^T(x)B_v(x))^{-1}B_v^T(x)B_u(x). \quad (6)$$

- The **CAP** could not admit any solution due to the actuator saturation constraints (2). In such a case, the **CAP** can be relaxed by requiring the computation of a command $u(t)$ such that $B_u(x)u(t)$ is somehow close to $B_v(x)v(t)$ (e.g. by evaluating at each time instant the numerical value of $\|B_u(x)u(t) - B_v(x)v(t)\|$);
- The analytical redundancy, i.e. $Rank(B_u(x)) = k < m$, implies that in principle there exists a set of admissible commands u all solutions of **CAP**. This fact can be exploited to comply with other specifications besides the **CAP** requirements.

A common way to solve **CAP** at time t is that of minimizing the quadratic optimization problem

$$\begin{aligned} u(t) &\triangleq \arg \min_{s,u} \|s\|_{Q_s}^2 + \|u\|_{R_u}^2, \\ B_v(x(t))v(t) &= B_u(x(t))u + s, \\ u &\in \Omega. \end{aligned} \quad (7)$$

The slack-variable s is used to enlarge the set of solutions in the parity equation (4) and it allows the achievement of approximating allocations. When zero, a perfect allocation is achieved. On the contrary, the penalty on u is optional and it is used to minimize the actuators' effort when many solutions are possible. It is well-known that an explicit solution to this optimization problem can be found in the unconstrained case while it does not exist in the general case. However, in order to reduce the computational burdens, several efficient algorithms based on the semi-explicit solution have been proposed in the last years (Johansen et al., 2005; Jin, 2005). For the purposes of this paper, it is important to notice here that computational efficiency obtained through explicit approaches is paid in term of a reduction of flexibility w.r.t. reconfiguration issues (Johansen et al., 2005).

2.2 Fault Modeling

Hereafter we suppose that the plant dynamics is corrupted by unpredictable events which alter the nominal behavior of the system. The aim is to make use of the input analytical redundancy to reconfigure the actuator allocation in such a way that the fault becomes ineffective.

We will focus onto the class of faults consisting in variations of actuator's effectiveness. Such a kind of fault can be naturally formalized in a multiplicative fashion

$$x(t+1) = a(x) + B_u(x)\Delta(t)u(t), \quad (8)$$

where $\Delta(t) = \text{diag}\{\delta_1(t), \delta_2(t), \dots, \delta_m(t)\}$ is the so-called Effectiveness Matrix and $\delta_i(t) \in \mathcal{R}, i = 1, \dots, m$ are piecewise constant sequences representing the effectiveness of any single actuator. Notice that, in absence of fault occurrences, $\Delta(t) = I$. Moreover, the parity equation (4) becomes

$$B_v(x)v(t) = B_u(x)\Delta(t)u(t). \quad (9)$$

Then, the problem we want to solve can be stated as follows

\mathcal{F} - Tolerant Control Allocation Problem (\mathcal{F} -TCAP) - Given the virtual plant (8) and a virtual input $v(t) \in \mathcal{R}^k$, find a command input $u(t) \in \mathcal{R}^m$ such that (2) and (9) hold true.

3. TWO-STEP PROCEDURE

It may simply be observed that the knowledge of the Effectiveness Matrix $\Delta(t)$ makes \mathcal{F} -TCAP be reduced to a more simple CAP. This allows us to propose the following adaptive **two-step method** to solve \mathcal{F} -TCAP at each time t :

Step 1: Compute the diagonal matrix $\hat{\Delta}(t)$, the best estimate of $\Delta(t)$ at time t , based on records of N past system's measures.

Step 2: Solve the CAP problem defined in (2) and (9) by assuming (certainty equivalence hypothesis) $\Delta(t) = \hat{\Delta}(t)$.

There exists an huge amount of literature on both online parameters estimation and allocation problems. Many of the existing algorithms can be arranged in this general scheme.

3.1 A simple two-step algorithm

Hereafter, a very simple two-step algorithm is proposed by using quadratic programming arguments.

Step 1: - Estimation of $\hat{\Delta}(t)$

In order to estimate the Effectiveness Matrix it is convenient to rewrite things in terms of the Incremental matrix

$$\hat{\Gamma}(t) \triangleq \hat{\Delta}(t) - \hat{\Delta}(t-1) \quad (10)$$

defined as the diagonal matrix

$$\hat{\Gamma} \triangleq \text{diag}\{\hat{\gamma}_1, \hat{\gamma}_2, \dots, \hat{\gamma}_m\} \in \mathcal{R}^m$$

of increments of actuator loss-of-effectiveness $\hat{\gamma}_i(t) = \hat{\delta}_i(t) - \hat{\delta}_i(t-1)$, $i = 1, \dots, m$.

We are especially interested in algorithms able at detecting constant or slow-varying actuator faults or loss of effectiveness, that is in determining matrices $\hat{\Delta}(t)$ that "matches as much as possible" the measured signals of the plant in the last N time instants, with N arbitrarily chosen. This corresponds to solutions which minimize the entries of $\hat{\Gamma}$.

A workable strategy corresponds to the solution of the following weighted least-squares problem

$$\hat{\Gamma}(t) \triangleq \arg \min_{s_i, \Gamma} \sum_{i=1}^N \|s_i\|_{Q_i}^2 + \|\text{vect}(\Gamma)\|_R^2$$

$$x(t-i+1) - a(x(t-i))$$

$$-B_u(x(t-i)) \left[\Gamma + \hat{\Delta}(t-1) \right] u(t-i) = s_i, \quad i = 1, \dots, N \quad (11)$$

with $Q_i \gg R$, $i = 1, \dots, N$, where $\text{vect}(\Gamma) = [\gamma_1, \gamma_2, \dots, \gamma_m]^T \in \mathcal{R}^m$, $s_i \in \mathcal{R}^n$, $i = 1, \dots, N$ are slack vectors and $R = R' > 0$ and $Q_i = Q'_i > 0$, $i = 1, \dots, N$ consistent weighting matrices. The choice of N has an important role in such a computation: picking a small value of N means having less or no information and, in turn, bad parameters estimation results. On the contrary, a large value of N yields to long computation and reconfiguration times. A

reasonable choice is $m/n \leq N \leq 2m$.

Step 2: - Given $\hat{\Gamma}(t)$, compute $\hat{\Delta}(t) = \hat{\Delta}(t-1) + \hat{\Gamma}(t)$ and solve the following CAP

$$u(t) \triangleq \arg \min_{s, u} \|s\|_{Q_s}^2 + \|u\|_{R_u}^2$$

$$B_v(x(t))v(t) = B_u(x(t))\hat{\Delta}(t)u + s \quad (12)$$

$$u \in \Omega$$

where $s \in \mathcal{R}^n$ is the parity slack vector and $Q_s = Q'_s > 0$ and $R_u = R'_u \geq 0$ consistent weighting matrices. In order to force slack vector to be as small as possible usually $Q_s \gg R_u$ is chosen.

Remark 1 - In all cases in which a unique solution exists for problem (11) an analytical expression can easily be determined. This would be beneficial for maintaining the on-line numerical burden of the algorithm low. However, in general one cannot ensure that a unique solution exists unless special care in generating the inputs $u(t)$ is taken. See e.g. subsection 3.3.

3.2 Properties of the two-step algorithm

In this section we will investigate the properties of the proposed algorithm with a particular regard to constant actuator's faults or loss of effectiveness. To this end, the following fault at time t'

$$\Delta(t) = I \quad t < t', i = 1, \dots, m$$

$$\Delta(t) = \Delta' \quad t \geq t', i = 1, \dots, m \quad (13)$$

is assumed where $\Delta' = \text{diag}\{\delta'_1, \dots, \delta'_m\}$ is the a constant diagonal matrix corresponding to the true loss of effectiveness.

In particular, we are interested to study the asymptotical properties of the R-weighted estimation error

$$e_R(t) = \|\text{vect}(\hat{\Delta}(t) - \Delta')\|_R \quad (14)$$

and the conditions of its convergence to zero. It is reasonable in fact to argue that, as many other parameters estimators, the convergence of the proposed one strongly depends on the nature of the input signals. Such a dependence, especially in a closed loop embedding, can yield to partially uncorrected estimations (Åström, Wittenmark, 1989). The following result on the monotonicity of the estimation error can be stated.

Proposition 1 - Given the overactuated physical plant (8) and the corresponding virtual plant (3)-(9), let the algorithm (11) perform under (13). Then, the weighted estimation error $e_R(t) = \|\text{vect}(\hat{\Delta}(t) - \Delta')\|_R$ is a monotonically non-increasing sequence, i.e. $e_R(t+1) \leq e_R(t)$, $\forall t > t' + N$.

Proof - See (Casavola, Garone, 2006).

Finally, by Proposition 1 and exploiting some arguments of its proof, under a constant fault it is possible to conclude that:

Main results

- 1 - As it was expected, in the general case the algorithm does not ensure that $e_R(t)$ converges to zero. In fact, the convergence strictly depends on the nature of the $u(t)$ history;

- 2 - Because $e(t)$ is monotonically non-increasing, if $\exists t^* > t' + N$ such that $e_R(t^*) = 0$ then $e_R(t) = 0, \forall t \geq t^*$;
- 3 - A sufficient condition for $e(t)$ to have zero value at some finite time $t^* > t' + N$ is that $\text{rank}\{M(t^*)\} = m$, where

$$M(t) = \begin{pmatrix} B_u(x(t-1)) \text{diag}\{u_1(t-1), \dots, u_m(t-1)\} \\ \dots \\ B_u(x(t-N)) \text{diag}\{u_1(t-N), \dots, u_m(t-N)\} \end{pmatrix} \quad (15)$$

Proof - See (Casavola, Garone, 2006).

3.3 A threshold two-step algorithm for linear systems

It has been shown that the proposed two-step algorithm does not guarantee the convergence of the estimation error to zero in general because of possible rank deficiency of the $M(t)$ matrix. A popular way to move around this obstacle is that of introducing an artificial disturbances able to force the input signals to be persistently exciting. Those disturbances obviously cause unwanted side effects on the system behavior. Here we will perform a different policy and we will exploit both parameter estimator properties and actuator redundancy to reduce side effects.

This is made by using the following two key ideas:

- 1- Because of the monotonicity of the estimation error, in order to have an exact estimate of $\Delta(t)$ is enough that matrix $M(t)$ have full rank at least for a single time instant t^*
- 2- It is possible to exploit actuator redundancy in order to reduce the side effects of artificial disturbances.

In order to easily perform the objective of having a full rank $M(t)$, the following sufficient condition for linear systems is proved.

Proposition 2 - Let $B_u(x) = B_u$ be a constant matrix and $B_{u,j} \neq 0, j = 1, \dots, m$, denote the j -th column of B_u . Then $\text{rank}\{M(t)\} = m$ for $N = m$ provided that

$$\text{rank} \left\{ \begin{pmatrix} u_1(t-1) & \dots & u_m(t-1) \\ \dots & \dots & \dots \\ u_1(t-m) & \dots & u_m(t-m) \end{pmatrix} \right\} = m, \quad (16)$$

Proof It is enough to notice that, for $N=m$ and $B_u(x(t)) = B_u$, one has that

$$M(t) = M_U(t) M_{B_u}, \quad \forall t \quad (17)$$

where:

$$M_U(t) = \begin{pmatrix} u_1(t-1)I_{n \times n} & \dots & u_m(t-1)I_{n \times n} \\ \dots & \dots & \dots \\ u_1(t-m)I_{n \times n} & \dots & u_m(t-m)I_{n \times n} \end{pmatrix} \quad (18)$$

is full rank whenever (16) holds true and

$$M_{B_u} = \begin{pmatrix} B_{u,1} & 0 & \dots & 0 \\ 0 & B_{u,2} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & B_{u,m} \end{pmatrix} \quad (19)$$

is full rank if $B_{u,j} \neq 0, j = 1, \dots, m$. \square

An on-line algorithm which applies one appropriate perturbation to the allocated inputs $u(t)$ so that condition (16) is ensured is proposed in (Casavola, Garone, 2006).

4. NUMERICAL EXAMPLES

Here we will deal with a spherical underwater vehicle described in (Fossen, 2006). The state of the system consists of positions $\eta = (x, y, z, \phi, \theta, \psi)$, where (x, y, z) are the positions of the CoG and (ϕ, θ, ψ) the Euler angles in the earth-fixed framework, and speeds $\nu = (u, v, w, p, q, r)$, that is the body velocities and the angular velocities in the body-fixed frame.

The whole system can be described by the following system of differential equations

$$\begin{cases} M\dot{\nu} + C(\nu)\nu + D(\nu)\nu = B_u(\nu)u \\ \dot{\eta} = J(\eta)\nu \end{cases}$$

where M is the inertia matrix

$$M = \begin{bmatrix} m - X_{\dot{u}} & 0 & 0 & 0 & 0 & 0 \\ 0 & m - Y_{\dot{v}} & 0 & 0 & 0 & 0 \\ 0 & 0 & m - Z_{\dot{w}} & 0 & 0 & 0 \\ 0 & 0 & 0 & I_x - K_{\dot{p}} & 0 & 0 \\ 0 & 0 & 0 & 0 & I_y - M_{\dot{q}} & 0 \\ 0 & 0 & 0 & 0 & 0 & I_z - N_{\dot{r}} \end{bmatrix}$$

$C(\nu)$ is a skew symmetric matrix taking into account the Coriolis effects

$$C(\nu) = \begin{bmatrix} 0 & -mr & mq & 0 & -Z_{\dot{w}} & Y_{\dot{v}}v \\ mr & 0 & -mp & Z_{\dot{w}}w & 0 & -X_{\dot{u}}u \\ -mq & mp & 0 & -Y_{\dot{v}}v & X_{\dot{u}}u & 0 \\ 0 & -Z_{\dot{w}}w & Y_{\dot{v}}v & 0 & (I_z - N_{\dot{r}})r & -(I_y - M_{\dot{q}})q \\ Z_{\dot{w}}w & 0 & -X_{\dot{u}}u & -(I_z - N_{\dot{r}})r & 0 & (I_x - K_{\dot{p}})p \\ -Y_{\dot{v}}v & X_{\dot{u}}u & 0 & (I_x - K_{\dot{p}})p & -(I_x - K_{\dot{p}})p & 0 \end{bmatrix}$$

$D(\nu)$ is a matrix taking into account hydrodynamic frictions

$$D(\nu) = \begin{bmatrix} \tilde{X}_u|u| & 0 & 0 & 0 & 0 & 0 \\ 0 & \tilde{Y}_v|v| & 0 & 0 & 0 & 0 \\ 0 & 0 & \tilde{Z}_w|w| & 0 & 0 & 0 \\ 0 & 0 & 0 & \tilde{K}_p|p| & 0 & 0 \\ 0 & 0 & 0 & 0 & \tilde{M}_q|q| & 0 \\ 0 & 0 & 0 & 0 & 0 & \tilde{N}_r|r| \end{bmatrix}$$

and $J(\eta)$ is the usual 6DOF kinematics transformation matrix:

$$J(\eta) = \begin{bmatrix} J_1(\eta) & 0 \\ 0 & J_2(\eta) \end{bmatrix}$$

where

$$J_1(\eta) = \begin{bmatrix} c\psi c\theta & -s\psi c\phi + c\psi s\theta s\phi & s\psi s\phi + c\psi c\phi s\theta \\ s\psi c\theta & c\psi c\phi + s\psi s\theta s\phi & -c\psi s\theta + s\psi s\psi c\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix}$$

and

$$J_2(\eta) = \begin{bmatrix} 1 & s\phi t\theta & c\phi t\theta \\ 0 & c\phi & -s\phi \\ 0 & s\phi/c\theta & c\phi/c\theta \end{bmatrix}.$$

Finally, the matrix B_u is used to describe the physical actuators. In this case there are 9 thrusters, organized in three families of three thrusters each disposed on the 3 planes orthogonal to each axes and passing from the centre of mass:

$$B(\eta) = B_u = \begin{bmatrix} K_{T_{11}} & K_{T_{12}} & K_{T_{13}} & 0 & 0 \\ 0 & 0 & 0 & -K_{T_{24}} & K_{T_{25}} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & l_4 K_{T_{24}} & -l_5 K_{T_{25}} \\ 0 & 0 & 0 & 0 & 0 \\ -l_1 K_{T_{11}} & l_2 K_{T_{12}} & l_3 K_{T_{13}} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ K_{T_{26}} & 0 & 0 & 0 & 0 \\ 0 & K_{T_{37}} & K_{T_{38}} & K_{T_{39}} & 0 \\ -l_6 K_{T_{26}} & 0 & 0 & 0 & 0 \\ 0 & -l_7 K_{T_{37}} & l_8 K_{T_{38}} & l_9 K_{T_{39}} & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Table 1. Physical parameters of the UAV

| | | |
|----------------------------------|------------------------------------|--------------------------------|
| $X_{\dot{u}} = -30 \text{ kg}$ | $X_u = -80 \text{ kg/s}$ | $X_{u u } = -120 \text{ kg}$ |
| $Y_{\dot{v}} = -110 \text{ kg}$ | $Y_v = -110 \text{ kg/s}$ | $Y_{v v } = -200 \text{ kg}$ |
| $Z_{\dot{w}} = -80 \text{ kg}$ | $Z_{\dot{w}} = -100 \text{ kg/s}$ | $Z_{w w } = -150 \text{ kg}$ |
| $K_{\dot{p}} = -15 \text{ kg m}$ | $K_{\dot{p}} = -30 \text{ kg m/s}$ | $K_{p p } = -35 \text{ kg m}$ |
| $M_{\dot{q}} = -20 \text{ kg m}$ | $M_{\dot{q}} = -40 \text{ kg m/s}$ | $M_{q q } = -40 \text{ kg m}$ |
| $N_{\dot{r}} = -1 \text{ kg m}$ | $N_{\dot{r}} = -10 \text{ kg m/s}$ | $N_{r r } = -15 \text{ kg m}$ |
| $x_G = 0 \text{ m}$ | $x_B = 0 \text{ m}$ | $\bar{V} = 0.0183 \text{ m}^3$ |
| $y_G = 0 \text{ m}$ | $y_B = 0 \text{ m}$ | $m = 180 \text{ kg}$ |
| $z_G = 0 \text{ m}$ | $z_B = 0.04 \text{ m}$ | $I_x = 28 \text{ kg m}^2$ |
| $I_x = 25 \text{ kg m}^2$ | $I_y = 29 \text{ kg m}^2$ | $I_z = 0 \text{ kg m}^2$ |
| $I_{xy} = 0 \text{ kg m}^2$ | $I_{xz} = 0 \text{ kg m}^2$ | |
| $\tilde{X}_u = X_u + X_{u u }$ | $\tilde{Y}_v = Y_v + Y_{v v }$ | $\tilde{Z}_w = Z_w + Z_{w w }$ |
| $\tilde{N}_r = N_r + N_{r r }$ | $\tilde{M}_q = M_q + M_{q q }$ | $\tilde{K}_p = K_p + K_{p p }$ |

Table 2. Parameters

| | |
|--|-----------------------------------|
| $K_{T11} = K_{T12} = K_{T13} = 7$ | $l_1 = 0.4, l_2 = 0.4, l_3 = 0.5$ |
| $K_{T24} = K_{T25} = K_{T26} = 7$ | $l_4 = 0.2, l_5 = 0.1, l_6 = 0.2$ |
| $K_{T37} = K_{T38} = K_{T39} = 7$ | $l_7 = 0.3, l_8 = 0.3, l_9 = 0.4$ |
| $\Delta_0 = I_{9 \times 9}$ | |
| $\Delta_f = \text{diag}(0.5, 1, .7, 1, 1, 1, 1, 1, 1)$ | |
| $K_1 = \text{diag}(-5, -5, -5, -5, -5, -5)$ | |
| $K_2 = \text{diag}(-4, -4, -4, -4, -4, -4)$ | |

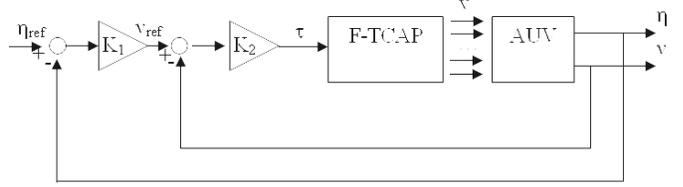


Fig. 2. Control Scheme of the AUV.

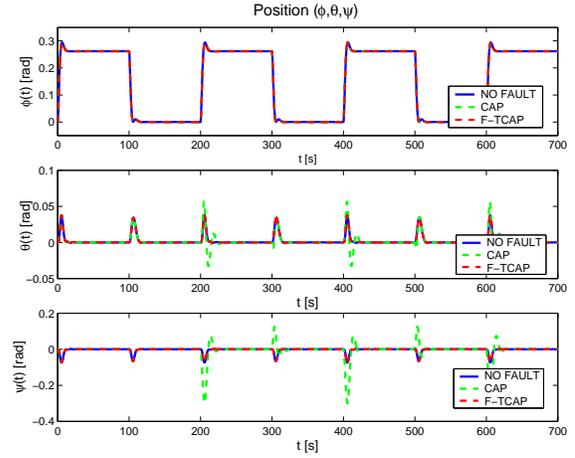
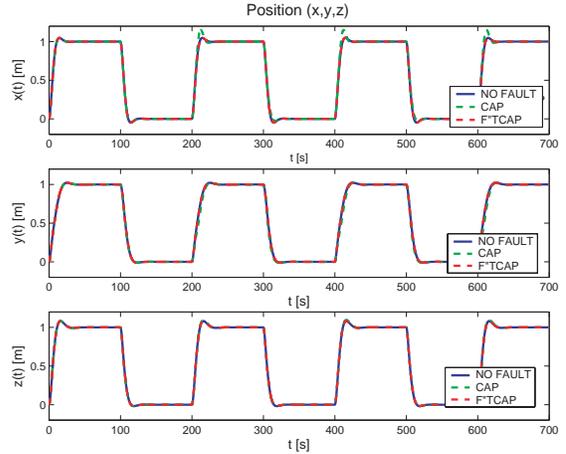
Moreover, we will define the virtual input matrix B_v as the identity matrix, $B_v = I_{6 \times 6}$. With such a choice, the virtual input v represents exactly the total forces and torques produced by the thrusters τ .

The overall control is the one shown in Fig. 2 composed of two P controller loops plus a F-TCAP module that performs an estimation of the effectiveness parameters and uses the latter to suitably allocate the control to the thrusters. In order to ensure the possibility to distinguish between the various actuator effects, the quadratic allocation is performed by using a time-varying weighting matrix W . The used plant and control parameters are in Table 1 and Table 2. The effectiveness matrix is supposed to be the following:

$$\Delta(t) = \begin{cases} \Delta_0 & t \leq 200 \text{ s} \\ \Delta_f & t > 200 \text{ s} \end{cases}.$$

For the sake of comprehension of the simulative results, Δ_f it is supposed to have variations on the family of thrusters acting along direction x and angle ψ (See Table 2). In Fig. 3 and 4 the evolutions of $\eta(t)$ are shown for the non-faulty case, for the case of using the non-adaptive CAP strategy as allocator and for using the adaptive \mathcal{F} -TCAP strategy instead. Notice that both CAP and \mathcal{F} -TCAP are executed at each time instant, with the only noticeable difference that no adaptation of the actuators' effectiveness is performed under CAP.

In this example it is possible to note that, after the fault event, the \mathcal{F} -TCAP is able to maintain the vehicle extremely close to his nominal path (no-faulty case) while without adaptation, the vehicle follows a different trajectory. This is evident in particular in the variables $x(t)$ and $\psi(t)$ that are those more directly affected by loss of effectiveness. This fact can also be observed in Fig. 5, where the differences between the corresponding trajectories are shown in the 3-dimensional space for $400 \text{ s} < t < 500 \text{ s}$. In Fig. 6 the virtual commands mainly affected by the fault event are shown: it is possible to see that while, for the non-adaptive CAP, the virtual inputs are changed substantially by the control law w.r.t. the no faulty-case, under the \mathcal{F} -TCAP scheme the control actually performs along its nominal behavior because the reconfiguration issue is completely managed by the allocation block. Finally in Fig.7 the parameter estimates are reported for completeness.

Fig. 3. ϕ - θ - ψ position, with and without \mathcal{F} -TCAPFig. 4. x-y-z trajectory between $t=200$ and $t=300$

5. CONCLUSIONS

A preliminary adaptive scheme to perform fault tolerant control allocation for nonlinear discrete-time dynamical systems subject to actuator faults or loss of effectiveness. A preliminary algorithm has been proposed and its proper-

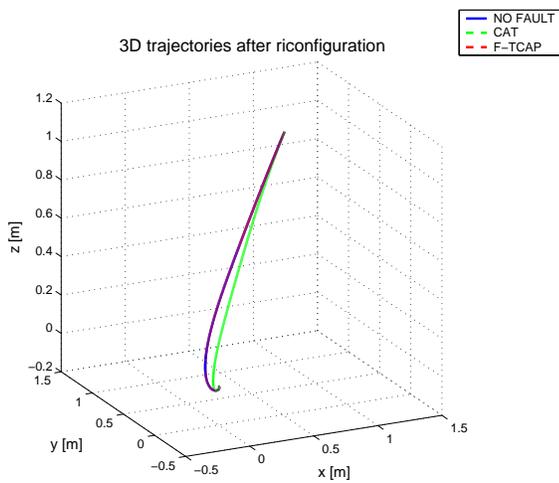


Fig. 5. x-y-z position, with and without \mathcal{F} -TCAP

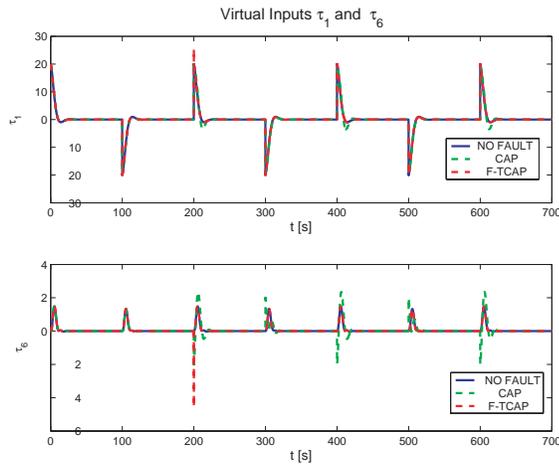


Fig. 6. Virtual Inputs τ_1 and τ_6

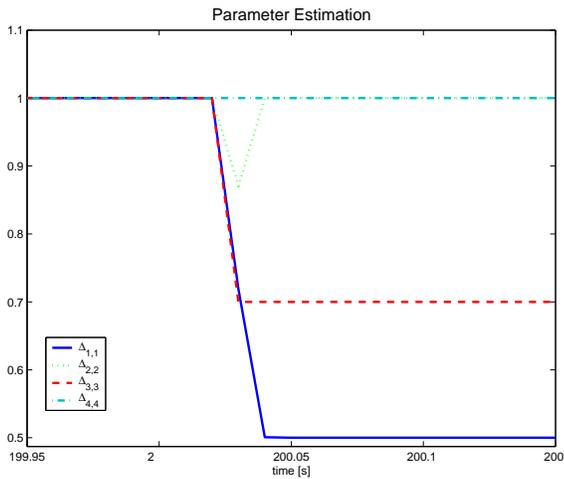


Fig. 7. Parameter Estimation

ties investigated. The effectiveness of the proposed method has been shown by means of a numerical experiment.

REFERENCES

Åström K.J., B. Wittenmark, *Adaptive Control*, Addison-Wesley Publishing Company, 1989

- Bodson M., J.E. Groszkiewicz, "Multivariable Adaptive Algorithms for Reconfigurable Flight Control", *IEEE Transactions on Control Systems Technology*, Vol. 5, No. 2, 1997, pp.217-229.
- Bodson M., "Evaluation of Optimization Methods for Control Allocation", *Journal of Guidance, Control and Dynamics*, Vol. 25, No. 4, 2002, pp. 703-711.
- Bolender M.A., D.B. Doman, "Nonlinear Control Allocation Using Piecewise Linear Functions: A Linear Programming Approach", *Journal of Guidance, Control and Dynamics*, Vol. 28, No.3, 2005, pp.558-562.
- Bordignon K.A., W.C. Durham, "Closed-Form Solutions to Constrained Control Allocation Problem", *Journal of Guidance, Control and Dynamics*, Vol. 18, No.5, 1995, pp.1000-1007.
- Bošković J.D., B. Ling, R. Prasad, R.K. Mehra, "Design of Control Allocation Algorithms for Overactuated Aircraft Under Constraints Using LMIs", *Proceedings of the 41st IEEE Conference on Decision and Control*, Las Vegas, 2002, pp. 1711-1716.
- Buffington J., D. Enn, "Lyapunov Stability Analysis of Daisy Chain Control Allocation", *Journal of Guidance, Control and Dynamics*, Vol. 19, No. 6, 1996 pp.1226-1230
- Buffington J., P. Chandler, M. Pachter, "Integration of on-line System Identification and optimization-based Control Allocation", *AIAA Guidance, Navigation, and Control Conference*, Boston, 1998, AIAA-98-4487.
- Casavola A., E. Garone, "Adaptive Schemes for Control Allocation", *DEIS-University of Calabria, Technical report*, DEIS-18/06, 2006.
- Durham W.C., "Constrained Control Allocation", *Journal of Guidance, Control and Dynamics*, Vol. 16, No. 4, 1993 pp.717-725
- Durham W.C., "Efficient, Near-Optimal Control Allocation", *Journal of Guidance, Control and Dynamics*, Vol. 22, No.2, 1998, pp.369-372.
- Fossen T.I., "Nonlinear Model and Control of Underwater Vehicles", *PhD Thesis, Norwegian Institute of Technology*, 2006.
- Härkegård O., "Efficient Active Set Algorithms for Solving Constrained Least Squares Problem in Aircraft Control Allocation", *Proceedings of the 41st IEEE Conference on Decision and Control*, Las Vegas, 2002, pp. 1295-1300.
- Jin J., "Modified Pseudoinverse Redistribution Methods for Redundant Controls Allocation", *Journal of Guidance, Control and Dynamics*, Vol.28, No.5, 2005, pp. 1076-1079.
- Johansen T.A., T.I. Fossen, P. Tøndel, "Efficient Optimal Constrained Control Allocation via Multiparametric Programming", *Journal of Guidance, Control and Dynamics*, Vol. 28 No.3, 2005, pp. 506-514.
- Kwakernaak H., R. Sivan, *Linear Optimal Control System*, Wiley, 1972.
- Petersen J.A.M., M. Bodson, "Interior-Point Algorithms for Control Allocation", *Journal of Guidance, Control and Dynamics*, Vol. 28, No.3, 2005, pp. 471-480.
- Tao G., S. Chen, S.M. Joghi, "An adaptive control scheme for systems with unknown actuator failures", *Automatica*, Vol. 38, 2002, pp. 1027-1034.