Adaptive Control Allocation for Fault Tolerant Overactuated Autonomous Vehicles

Alessandro Casavola and Emanuele Garone

Dipartimento di Elettronica, Informatica e Sistemistica, Università della Calabria
87037 Arcavacata di Rende - (CS), Italy
casavola,egarone@deis.unical.it

Abstract

This paper presents an adaptive actuators allocation scheme 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. Two simple algorithms are proposed for nonlinear discrete-time systems. The main properties of the algorithms are summarized in the disturbance-free case and their effectiveness shown by means of two numerical examples.

1 Introduction

Actuators redundancy is an important issue to deal with in increasing the fault-tolerant and control properties of many real systems. For example, it is a very common matter in (autonomous) vehicle applications due to safety and performance reasons. A traditional way to handle overactuated systems (i.e. systems with physical actuator redundancy) is the use of optimal control design [1]. Such an approach achieves both regulation and control distribution amongst actuators at the same time. A different approach consists in using a (simpler) control law that specifies only 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 last decade and several methods have been proposed: Daisy Chaining [4], Direct Control Allocation [2], Convex Optimization Based algorithms [5]-[11] and PseudoInverse-Redistribution (PIR) methods [12]-[13].

One active area of research in overactuated systems is how to exploit their physical redundancy to develop effective reconfigurable control strategy so as 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 [14]-[16]. An alternative approach, that will be investigated here, is the so-called Reconfigurable Control Allocation (RCA) problem [17]-[19]. 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 $v(t)$. It is assumed that the virtual inputs are fully equivalent to the physical inputs $u(t)$ in generating a desired 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. Therefore, in the case of actuator's faults, control reconfiguration is possible in many cases 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 which making the system behaving in the same way.

In this paper a preliminary adaptive control allocation scheme is proposed able of solving RCA problems for non-linear discrete-time systems. Such systems are subject to actuator's faults or loss

Casavola, A.; Garone, E. (2007) Adaptive Control Allocation for Fault Tolerant Overactuated Autonomous Vehicles. In *UAV Design Processes / Design Criteria for Structures* (pp. 3.2-1 – 3.2-16). Meeting Proceedings RTO-MP-AVT-145, Paper 3.2. Neuilly-sur-Seine, France: RTO. Available from: http://www.rto.nato.int/abstracts.asp.

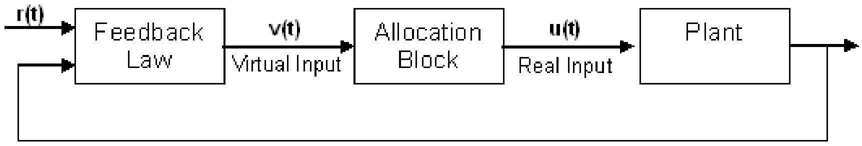

Figure 1: Control structure with allocation and control performed separately

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 the on-line computation of the allocation's rules possible by solving a standard constrained QP problem.

The main properties of the scheme are summarized and some indications on how to ensure persistence of input excitation are given in order to ensure good estimation properties. For simplicity all development is 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.

The paper is organized as follows: the problem is stated in Section II. In Section III an adaptive allocation scheme is presented, two different algorithms are proposed and their properties summarized. Finally, two numerical experiments are reported 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

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

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: $Rank(B_u(x)) = k < m, \quad \forall x$;

2) The input signal $u(t)$ lies into a compact set Ω, i.e.

$$u(t) \in \Omega := \left\{u \in \mathcal{R}^m \,|\, u^- \le u \le u^+\right\}, \tag{2}$$

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

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

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

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

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 $v(t) \in \mathcal{R}^k$ is the virtual control input. Hereafter, the state space equation (3) will be referred to as the *virtual plant* while (4) 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 command 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 ([4]-[13]). Note that:

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

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

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

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

- The **CAP** could not admit any solution due to the actuators saturation constraints (2). In such a case, the **CAP** can be relaxed by requiring to compute a command $u(t)$ such that $B_u(x(t))u(t)$ is somehow close to $B_v(x(t))v(t)$ (e.g. by evaluating at each time instant the numerical value of $\|B_u(x(t))u(t) - B_v(x(t))v(t)\|$);

- The analytical redundancy, i.e. $Rank(B_u(x)) = k < m$, implies that in principle there exists a set of admissible commands u solution for the **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

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

The slack-variable s is used to enlarge the set of solutions in the parity equation (4) and it allows the achievement of approximating allocation rules. 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 or to impose some preference 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 computational burdens, several efficient algorithms based on the semi-explicit solution have been proposed in the last years (see [11]-[13]). 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 (see [11]).

2.2 Fault Modeling

For simplicity we will restrict our attention on the class of faults describing actuator effectiveness variations. The effect of a fault event is then to change in percentage the nominal gain of some actuator signal. Such a kind of fault can be naturally formalized in a multiplicative fashion. Additive effects will be considered in a future work.

$$x(t+1) = a(x(t)) + B_u(x(t))\Delta(t)\, u(t), \tag{8}$$

where $\Delta(t) = diag\{\delta_1(t), \delta_2(t), \ldots, \delta_m(t)\}$ is the so-called Effectiveness Matrix and $\delta_i(t) \in \mathcal{R}, i = 1, ..., m$ 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

$$B_v(x(t))\, v(t) \;=\; B_u(x(t))\Delta(t)\, u(t). \tag{9}$$

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 A two-step procedure

It may simply be observed that the knowledge of the Effectiveness Matrix $\Delta(t)$ makes $\mathcal{F}$**-TCAP** to be reduced to a more simple **CAP**. This allows us to propose the following adaptive **two steps 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** defined by (2) and (9) by assuming (certainty equivalence hypothesis) $\Delta(t) = \hat{\Delta}(t)$,

There is an huge amount of literature both on online parameters estimation and allocation problems. Many of the existing algorithms solving the two problems 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: - Estimate of $\hat{\Delta}(t)$

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

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

defined as the diagonal matrix

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

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

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 $\hat{\Gamma}$. In principle, such a strategy corresponds to the sequential solution of the following two least-squares problems:

$$\begin{array}{l} \hat{s}_i(t) \triangleq \arg\min\limits_{s_i,\Gamma} \sum\limits_{i=1}^{N} \|s_i\|^2_{Q_i} \\ x(t-i+1) - a(x(t-i)) \\ -B_u(x(t-i)) \left[\Gamma + \hat{\Delta}(t-1)\right] u(t-i) = s_i,\ i = 1, \ldots, N \end{array} \tag{11}$$

and, once $\hat{s}_i(t), i = 1, \ldots, N$ are obtained,

$$\begin{array}{l} \hat{\Gamma}(t) \triangleq \arg\min\limits_{\Gamma} \|\text{vect}(\Gamma)\|^2_R \\ x(t-i+1) - a(x(t-i)) \\ -B_u(x(t-i)) \left[\Gamma + \hat{\Delta}(t-1)\right] u(t-i) = \hat{s}_i(t),\ i = 1, \ldots, N \end{array} \tag{12}$$

where $\text{vect}(\Gamma) = [\gamma_1, \gamma_2, \ldots, \gamma_m]^T \in \mathcal{R}^m$ and $s_i \in \mathcal{R}^n, i = 1, \ldots, N$ slack vectors and $R = R' > 0$ and $Q_i = Q_i' > 0, i = 1, \ldots, N$ consistent weighting matrices.

The choice of N has an important role in such a computation: picking a small value of N it means having few or no information at all and, in turn, bad parameters estimation results. On the contrary, a large value of N gives rise to long computation and reconfiguration times. A reasonable choice is $m/n \leq N \leq 2m$.

From a practical point of view, an approximate numerical solution to (11) and (12) can be obtained by combining those two optimization problems into a unique mixed weighted least-squares problem

$$\begin{array}{l} \hat{\Gamma}(t) \triangleq \arg\min_{s_i,\Gamma} \sum_{i=1}^{N} \|s_i\|^2_{Q_i} + \|\text{vect}(\Gamma)\|^2_R \\ x(t-i+1) - a(x(t-i)) \\ -B_u(x(t-i)) \left[\Gamma + \hat{\Delta}(t-1)\right] u(t-i) = s_i, \; i = 1, \ldots, N \end{array} \quad (13)$$

with $Q_i >> R, i = 1, \ldots, N$.

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

$$\begin{array}{l} u(t) \triangleq \arg\min_{s,u} \|s\|^2_{Q_s} + \|u\|^2_{R_u} \\ B_v(x(t))v(t) = B_u(x(t))\hat{\Delta}(t)u + s \\ u \in \Omega \end{array} \quad (14)$$

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 $Q_s >> R_u$ is chosen.

Remark 1 - In all cases in which a unique solution exists for problems (11), (12) and (13) an analytical expression can easily be determined. This would be beneficial for maintaining low the on-line numerical burden of the algorithm. However, in general one cannot ensure that a unique solution it exists unless special care in generating the inputs $u(t)$ is taken. See e.g. next section 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 faults or loss of effectiveness. To this end, the following fault at time t'

$$\begin{array}{ll} \Delta(t) = I & t < t', i = 1, \ldots, m \\ \Delta(t) = \Delta' & t \geq t', i = 1, \ldots, m \end{array} \quad (15)$$

is assumed where $\Delta' = diag\{\delta_1', \ldots \delta_m'\}$ 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

$$e_R(t) = ||\text{vect}(\hat{\Delta}(t) - \Delta')||_R \quad (16)$$

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 (see [14]).

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)-(12) perform under (15). 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 - Reported in [21]. □

Finally, thanks to Proposition 1, under a constant fault it is possible to conclude that:

Main results

1 - As 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 $rank\{(M(t^*)\} = m$, where

$$M(t) = \begin{pmatrix} B_u(x(t-1))diag\{u_1(t-1),...,u_m(t-1)\} \\ ... \\ B_u(x(t-N))diag\{u_1(t-N),...,u_m(t-N)\} \end{pmatrix} \tag{17}$$

Proof - Reported in [21]. □

3.3 A threshold two-step algorithm for linear systems

It has been shown that the proposed two-step algorithm to the $\mathcal{F}$**-TCAP** 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 by the introduction of artificial disturbances able to force the input signals to be persistently exciting the system. 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 $B_{u,j} \neq 0, j = 1, \ldots, m$, denote the j-th column of B_u. Then $rank\{M(t)\} = m$ for $N = m$ provided that

$$rank \left\{ \begin{pmatrix} u_1(t-1) & \ldots & u_m(t-1) \\ \ldots & \ldots & \ldots \\ u_1(t-m) & \ldots & u_m(t-m) \end{pmatrix} \right\} = m, \tag{18}$$

Proof - Reported in [21]. □

3.4 An on-line algorithm

It is possible then to present the following algorithm in which condition (18) is ensured in (19) by applying an appropriate perturbation to the allocated inputs $u(t)$.

Init
$forceConvergenceFlag = 0;$
$N = 0;$

Step 1 `- Residual generator and threshold`
$r = 0$
`if(`$N > 0$`)`

$r = x(t) - Ax(t-1) - B_u(\hat{\Delta}(t-1))u(t-1)$
if $(r^T Qr > \epsilon_{thr})$ AND $(forceConvergenceFlag == 0)$
$forceConvergenceFlag = 1;$
$N = 0;$
if $(forceConvergenceFlag == 1)$ AND $(N == m)$
$forceConvergenceFlag = 0;$

Step 2 - Parameter Estimation
Solve (13)

Step 3 - Allocation
Solve (14)

Step 4 - Adding Artificial Disturbances
if$(forceConvergenceFlag == 1)$

$$U_{old} = \begin{pmatrix} u^T(t-1) \\ \dots \\ u^T(t-N) \end{pmatrix}$$

if$(Ker(U_{old}) * u(t) == 0)$
Solve

$$\begin{array}{l} \min_{s,u} \|s\|^2_{Q_s} + \|u(t)\|^2_{R_u} \\ B_v(x)v = B_u(x)\hat{\Delta}(t)u(t) + s \\ u(t) = \sum_{i=1}^{N} \beta_i u(t-i) + \sum_{i=1}^{m-N} \alpha_i Ker_i\{U_{old}\}, \\ [\alpha_1, ..., \alpha_{m-N}]^T \neq 0 \\ u \in \Omega \end{array} \tag{19}$$

If$(N < m)$
$N = N + 1$
goto Step 1
where $Q \in \mathcal{R}^{n\times n}$ is an appropriate weighting matrix and ϵ_{thr} is an appropriate scalar threshold.

4 Numerical Examples

4.1 Linear unstable model

Consider the following linear model

$$x(t+1) = Ax(t) + B_u\Delta(t)u(t) \tag{20}$$

where $x \in \mathcal{R}^3$ is the state vector and $u = [u_1, u_2, u_3]$ the physical input vector subject to the constraints $-5 < u_i < 5 \quad i = 1, ..., 3$. Matrices A and B_u are

$$A = 1.2, \quad B_u = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}, \tag{21}$$

and $\Delta(t)$ is assumed as

$$\begin{array}{ll} \Delta(t) = diag\{1,1,1\} & t < 50 \\ \Delta(t) = diag\{1,1,0\} & 50 \le t < 225 \\ \Delta(t) = diag\{1,0.5,0\} & t \ge 225. \end{array} \tag{22}$$

consisting of a sequence of two faults. The first occurring at time $t = 50$, when the effectiveness of the third actuator becomes zero. This is followed, at time $t = 225$, by a 50% reduction of the effectiveness of the second actuator.

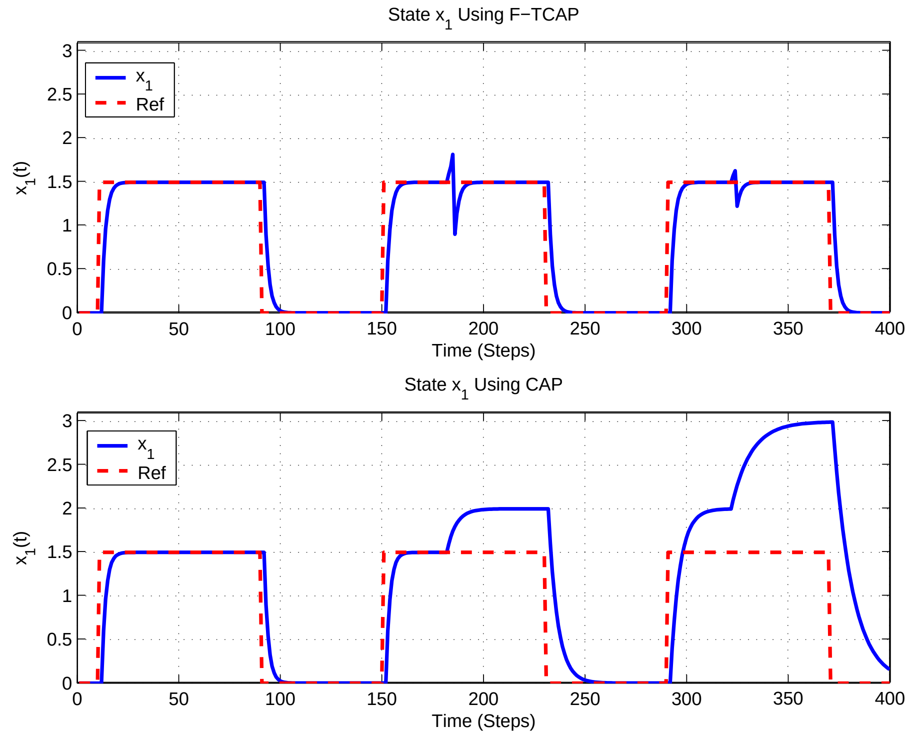

Figure 2: Output and reference with (UP) and without (BOTTOM) $\mathcal{F}$-**TCAP**.

The virtual input matrix $B_v = 1$ and the virtual control law $K = -0.6$ have been chosen. The threshold two-steps version of the ($\mathcal{F}$-TCAP) strategy was used with parameters: $\epsilon_{thr} = 10^{-5}$, $Q = Q_i = 10^5$, $i = 1, .., 3$, and $R = R_u = I_{3\times3}$.

Simulation results on plant evolutions, actuators' allocation and control reconfiguration are reported in next figures 3-6 for a tracking problem from an initial state $x(0) = 0$ and a square wave reference signal.

In order to show the the effectiveness of the adaptive strategy, two sets of simulations have been accomplished: with and without the use of the adaptive **($\mathcal{F}$-TCAP)** strategy. When **($\mathcal{F}$-TCAP)** is not used, a CAP problem is solved at time $t = 0$ and the corresponding allocation rule frozen afterwards.

Figs. 3-5 report respectively the output, the physical and virtual input closed-loop evolutions achieved with (UP) or without (BOTTOM) the use of the ($\mathcal{F}$-**TCAP**) strategy. In particular, in Fig. 3 it is easy to note the effectiveness of $\mathcal{F}$-**TCAP** in reconfiguring the actuators' allocation after a fault occurrence. Correspondingly, Figs. 4 and 5 report the physical and virtual input evolutions. It is worth noticing how signals related to failed actuators smartly change, coherently with the new estimated actuators effectiveness. On the contrary, under a frozen allocation, all input signals change uniformly and the tracking performance is lost. This behavior can be better explained in Fig. 5, where the virtual inputs v are shown in both cases. In the (UP) part is possible in fact to remark how, unlike in the (BOTTOM) part, the control law behavior is not influenced by the faulty events, apart around the time instants of fault occurrences, and the steady-state values of the control action remain unchanged.

Finally, in Fig. 6 , (UP) the estimation error $e_R(t) = ||Vect(\hat{\Delta}(t) - \Delta(t))||_R$ and (BOTTOM) the difference between the desired total control effort and the actual one, i.e. $B_u\Delta(t)u(t) - B_v v(t)$ are reported, both achieved under the proposed $\mathcal{F}$-**TCAP** algorithm. Those are two important indexes to evaluate the estimation and reconfiguration performances, the lower the better (at zero one has exact estimation and allocation). Notice, in particular, their finite-time convergence to zero.

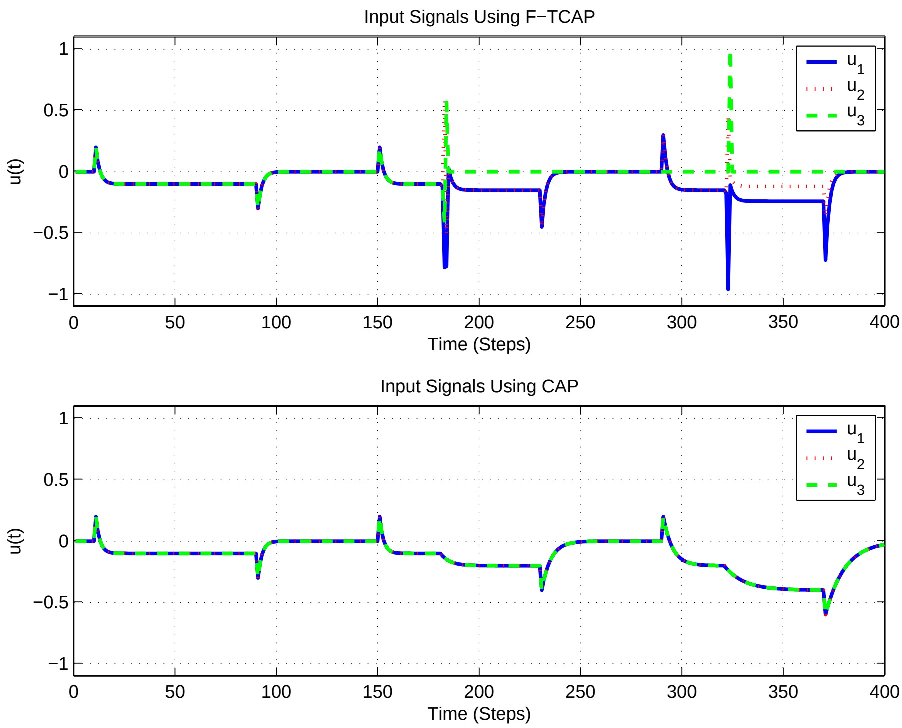

Figure 3: Physical inputs $u(t)$ with (UP) and without (BOTTOM) $\mathcal{F}$-**TCAP**.

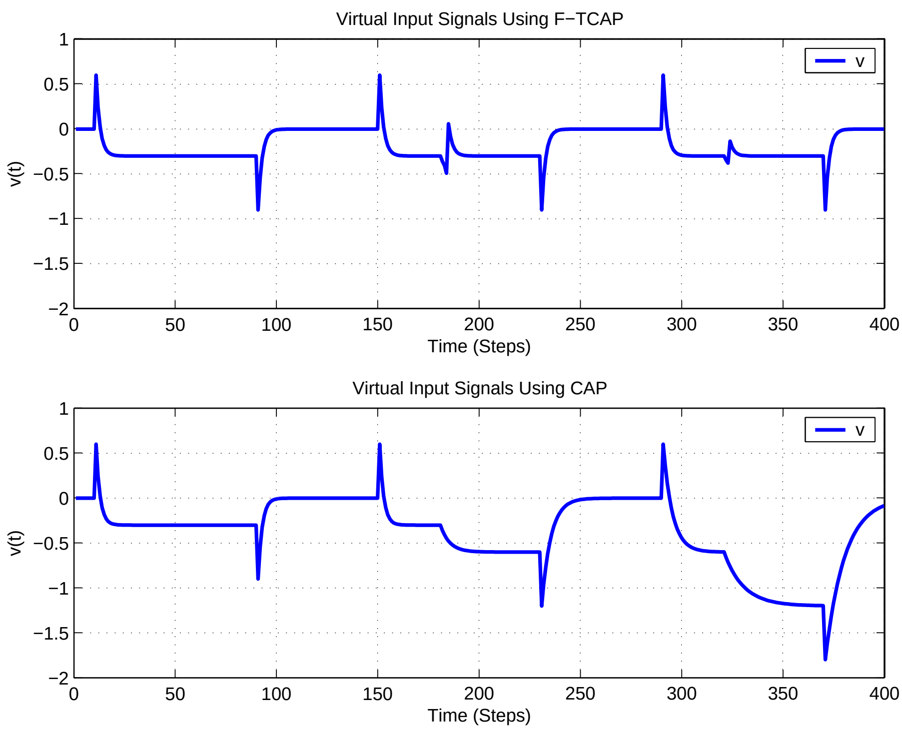

Figure 4: Virtual inputs $v(t)$ with (UP) and without (BOTTOM) $\mathcal{F}$-**TCAP**.

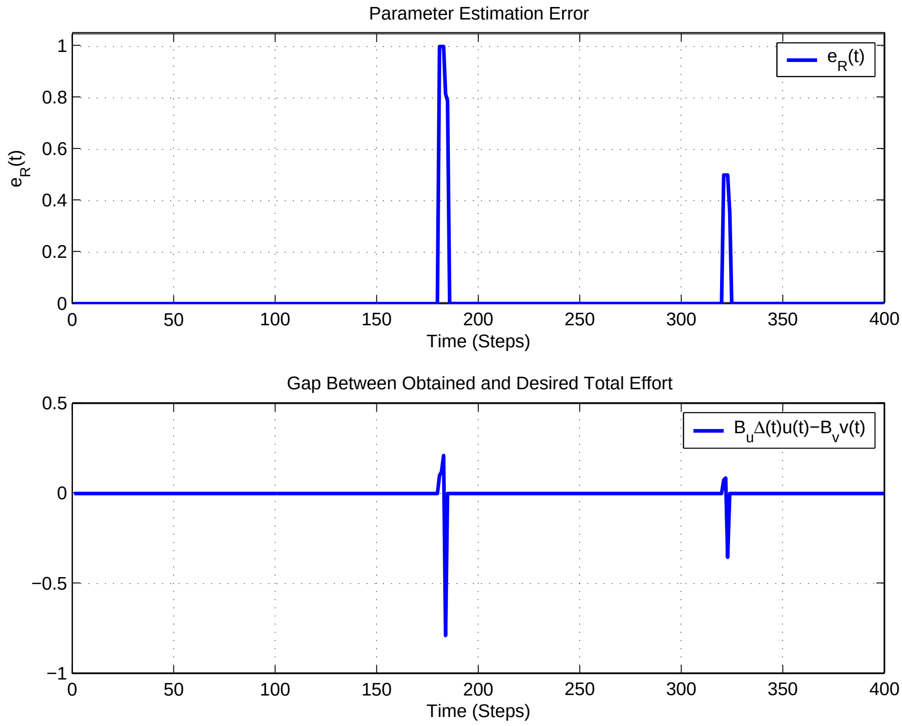

Figure 5: (UP) Parameter estimation error $e_R(t)$) and (BOTTOM) parity equation, viz. the gap between obtained total effort $B_u\Delta(t)u(t)$ and desired total effort $B_v v(t)$

4.2 Tracking of an overactuated autonomous marine vessel

We consider the ship model presented in [20] with actuator dynamics discarded for simplicity. In this model Earth-fixed positions (x, y) and yaw angle ϕ are represented by the vector $\eta = [x, y, \phi]^T$ and the body-fixed velocities are expressed by $\nu = [u, v, r]^T$, where u is forward velocity (surge), v is transverse velocity (sway) and r is the angular velocity in yaw (rate of turn). In order to normalize variables the following bis-scaling change of variables is accomplished:

$$\begin{aligned} \eta &= diag\{L, L, 1\}\eta'' \\ \nu &= diag\{\sqrt{gL}, \sqrt{gL}, \sqrt{gL}\}\nu'' \end{aligned} \tag{23}$$

where g is the gravity acceleration and L the length of the ship. Time was bis-scaled too. The resulting bis-scaled nonlinear ship model is as follows

$$\begin{aligned} \dot{\eta}''(t'') &= J(\eta''(t''))\nu''(t'') \\ M\dot{\nu}''(t'') + C(\nu'')\nu''(t'') + D\nu'' &= B_u\Delta(t'')u''(t'') \end{aligned} \tag{24}$$

where $u'' = [u''_1, ..., u''_6]^T$ are the input signals scaled in such a way that $|u''_i| \leq 1$. In particular, u''_1 and u''_2 represent the two identical main propellers whereas the other four inputs are thrusters acting as rudders. Moreover, M is the inertia matrix and $C(\nu)$ and D matrices taking into account Coriolis, centripetal and damping forces. Matrix J is the usual rotation matrix in yaw

$$J(\eta'') = \begin{pmatrix} cos(\phi'') & -sin(\phi'') & 0 \\ sin(\phi'') & cos(\phi'') & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{25}$$

In our simulation we consider a supply vessel with mass $m = 6.4\,10^6(Kg)$ and length $L = 76.2(m)$ with the following non-dimensional matrices [20]:

$$M = \begin{pmatrix} 1.1274 & 0 & 0 \\ 0 & 1.8902 & -0.0744 \\ 0 & -0.0744 & 0.1278 \end{pmatrix}, \tag{26}$$

$$C(\nu'') = \\ = \begin{pmatrix} 0 & 0 & -1.8902v''+0.0744r'' \\ 0 & 0 & 1.1274u'' \\ 1.8902v''-0.0744r'' & -1.1274u'' & 0 \end{pmatrix}, \qquad (27)$$

$$D'' = \begin{pmatrix} 0.0414 & 0 & 0 \\ 0 & 0.1775 & -0.0141 \\ 0 & -0.1073 & 0.0568 \end{pmatrix}, \qquad (28)$$

$$B_u = 10^{-3} \begin{pmatrix} 13.0 & 13.0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 11.6 & 11.6 & 6.0 & 6.7 \\ 0 & 0 & -4.6 & -4.6 & 2.7 & 2.2 \end{pmatrix}, \qquad (29)$$

The following fault occurrences

$$\begin{array}{ll} \Delta(t) = diag\{1,1,1,1,1,1\} & t < 72.3(s), \\ \Delta(t) = diag\{1,1,1,1,1,0\}, 72.3(s) \le t'' \le 211.6(s), & \\ \Delta(t) = diag\{0.5,1,1,1,1,0\} & t'' \ge 211.6(s). \end{array} \qquad (30)$$

are considered. The first fault corresponds to the complete failure of a thruster. The second subsequent event is an additional loss of effectiveness of one of the two main propellers. The virtual input matrix is chosen to be

$$B_v = 10^{-3} \begin{pmatrix} 13.0 & 0 & 0 \\ 0 & 11.6 & 6.0 \\ 0 & -4.6 & 2.7 \end{pmatrix}, \qquad (31)$$

and the following back-stepping control law [20]

$$B_v v = M\dot{\nu}_r + C(\nu'')\nu_r - J^T(\eta'')K_d s - J^T(\eta'')K_\eta \tilde{\eta} \qquad (32)$$

is considered for the virtual plant where

$$\begin{array}{rcl} \tilde{\eta} & = & \eta - \eta_d, \\ \dot{\eta}_r & = & \dot{\eta}_d - \Lambda\tilde{\eta}, \\ \ddot{\eta}_r & = & \ddot{\eta}_d - \Lambda\left[J(\eta'')\nu'' - \dot{\eta}_d\right], \\ \nu_r & = & J^{-1}(\eta'')\dot{\eta}_r, \\ \dot{\nu}_r & = & J^{-1}(\eta'')\ddot{\eta}_r, \\ s & = & \dot{\eta}'' - \dot{\eta}_r = J(\eta'')\nu'' - \dot{\eta}_r, \end{array} \qquad (33)$$

Notice that η_d, that is the bis-scaled reference trajectory, is chosen in such a way that $\eta_d, \dot{\eta}_d, \ddot{\eta}_d$ are smooth and bounded. Finally Λ, K_η, K_d are the following design matrices: $\Lambda = 0.1I_{3\times3}$ and $K_\eta = K_d = I_{3\times3}$.

The plant, the control law and the control allocator have been simulated with a non dimensional virtual sampling time of $h'' = 0.02$, corresponding to $h = 0.0557(s)$ in real time. Simulation results are shown for an initial position vector $\eta_0 = [-3, 0, 3/2\pi]$ and initial speed vector $\nu_0 = [0.01, 0, 0]$. To show the effectiveness of the proposed strategy this experiment has been simulated both with the proposed $\mathcal{F}$-**TCAP** scheme and under a fixed allocation provided by solving **CAP** at time $t = 0$. Fig. 7 shows the position of the ship in terms of Earth-fixed position coordinates x, y and yaw angle ϕ, both for the $\mathcal{F}$-**TCAP** (UP) and **CAP** (BOTTOM) cases. Improvements can be noticed in the ϕ tracking under $\mathcal{F}$-**TCAP**.

Figs. 8-9 report the physical and virtual commands related to main propellers only. All other commands have been omitted for brevity because their changes are modest in these experiments. In particular, Fig. 8 shows the physical commands u_1 and u_2 to the main propellers. Also in this example, it is possible to notice how under $\mathcal{F}$-**TCAP** any single physical command is reconfigured coherently with the fault event while under **CAP** all signals change uniformly, increasing their values. This last fact can be also noticed in Fig. 9 where the single virtual input v_1, representing the

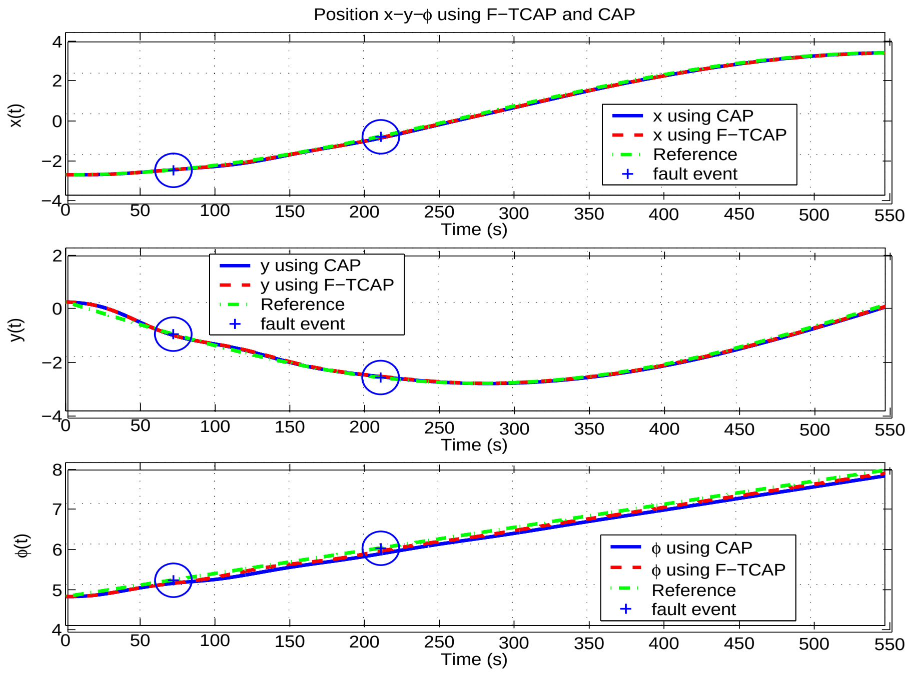

Figure 6: Ship Position in $x - y - \phi$ coordinates and reference with (UP) and without (BOTTOM) $\mathcal{F}$**-TCAP**.The circles indicates fault events.

total effort provided by the two propellers, is compared with the same virtual command corresponding to a fault-free experiment. By direct comparisons, it is possible to notice that under $\mathcal{F}$**-TCAP** the closed-loop performance is not substantially affected by the fault events. On the contrary, under **CAP** only, a control performance degradation usually results. Finally, Fig.10 reports the estimation error and parity equation histories: again their finite-time convergence to zero can be observed.

5 Conclusions

A preliminary adaptive scheme to perform fault tolerant control allocation for nonlinear discrete-time systems has been here proposed for disturbance free plats subject to loss of effectiveness. Two algorithms have been proposed and their properties investigated. The effectiveness of the proposed method is shown by means of two numerical experiments. Extensions to more general class of faults and in the presence of state and measurement disturbances are in progress.

Acknowledgment

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

References

[1] H. Kwakernaak, R. Silvan *Linear optimal control system*, Weley, 1972

[2] W.C. Durham, "Contrained Control Allocation", *Journal of Guidance, Control and Dynamics*, Vol. 16, No. 4, 1993 pp.717-725

[3] K.A. Bordignon, W.C. Durham, "Closed-Form Solutions to Contrained Control Allocation Problem", *Journal of Guidance, Control and Dynamics*, Vol. 18, No.5, 1995, pp.1000-1007.

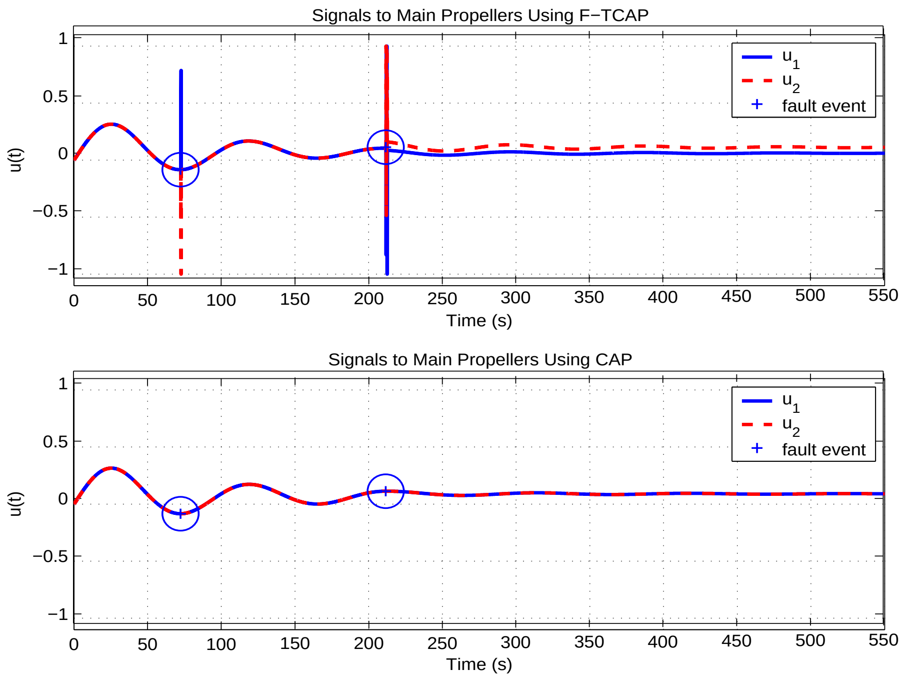

Figure 7: Physical inputs $u_1(t)$ (continuous) and $u_2(t)$ (dashed) to main propellers with (UP) and without (BOTTOM) $\mathcal{F}-TCAP$. The circles indicates fault events.

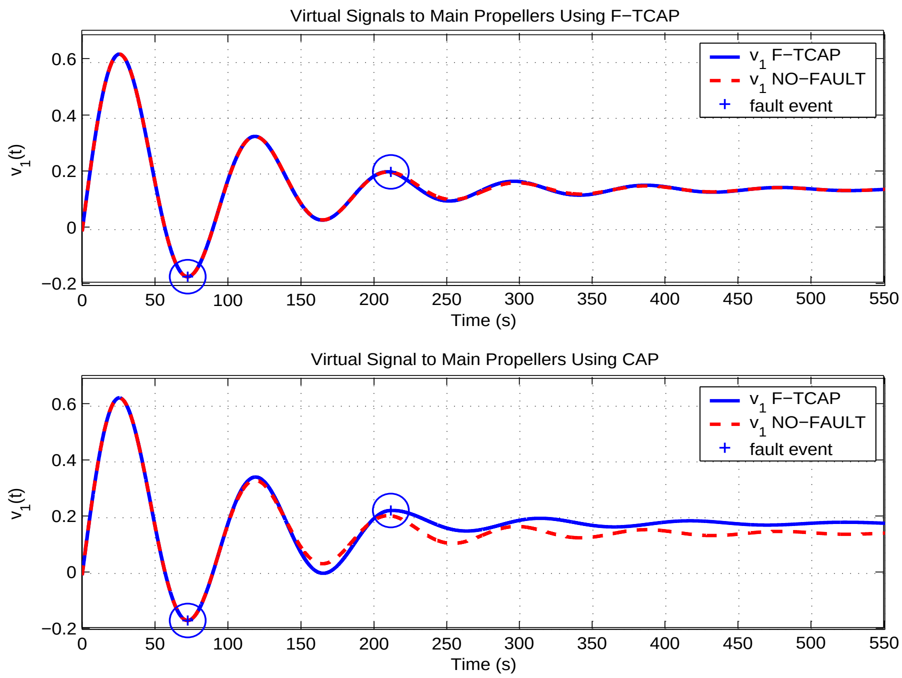

Figure 8: Virtual input $v_1(t)$ (continuous) to main propellers with (UP) and without (BOTTOM) $\mathcal{F}$**-TCAP**, both compared with the same signal for the fault-free case (dashed). The circles indicates fault events.

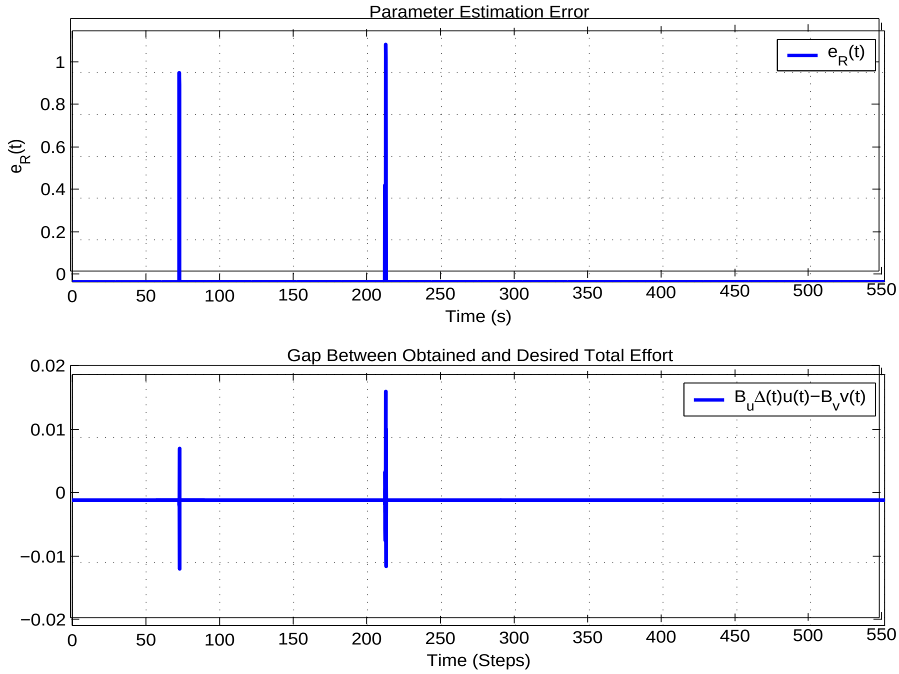

Figure 9: (UP) Parameter estimation error $e_R(t)$) and (BOTTOM) parity equation, viz. the gap between obtained total effort $B_u\Delta(t)u(t)$ and desired total effort $B_v v(t)$

[4] J. Buffington, D. Enn, "Lyapunov Stability Analysis of Daisy Chain Control Allocation", *Journal of Guidance, Control and Dynamics*, Vol. 19, No. 6, 1996 pp.1226-1230

[5] W.C. Durham, Efficient, "Near-Optimal Control Allocation", *Journal of Guidance, Control and Dynamics*, Vol. 22, No.2, 1998, pp.369-372.

[6] J.H. Oh, M.B. Jamoom, M.W. McConley, E.Feron, *Solving control allocation problems using semidefinite programming*, Vol 22, No.3, 1999, pp. 494-497.

[7] J.D. Bošković, B. Ling, R.Prasanth, 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.

[8] O. Härkegård, "Efficien Active Set Algorithms for Solving Contrained Least Squares Problem in Aircaft Control Allocation", *Proceedings of the 41st IEEE Conference on Decision and Control*, Las Vegas, 2002, pp. 1295-1300.

[9] K.P. Lindegaard, T.I. Fossen, "Optimal Contrained Control Allocation in Marine Surface Vessels with Rudder", *IFAC Conf. Menoeuvring and Control of Maine Craft*, Girona, 2003.

[10] J.A.M. Petersen, M. Bodson, "Interior-Point Algorithms for Control Allocation", *Journal of Guidance, Control and Dynamics*, Vol. 28, No.3, 2005, pp. 471-480.

[11] T.A. Johansen, T.I. Fossen, P.Tøndel, "Efficient Optimal Contrained Control Allocation via Multiparametric Programming", *Journal of Guidance, Control and Dynamics*, Vol. 28 No.3, 2005, pp. 506-514.

[12] M.Bodson, "Evaluation of Optimization Methods for Control Allocation", *Journal of Guidance, Control and Dynamics*, Vol. 25, No. 4, 2002, pp. 703-711.

[13] J.Jin, "Modified Peudoinverse Redistribution Methods for Redundant Controls Allocation", *Journal of Guidance, Control and Dynamics*, Vol.28, No.5, 2005, pp. 1076-1079.

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

[15] M. Bodson, J.E. Groszkiewicz, "Multivariable Adaptive Algorithms for Reconfigurable Flight Control", *IEEE Transactions on Control Systems Technology*, Vol. 5, No. 2, 1997, pp.217-229.

[16] G.Tao, S. Chen, S.M. Joghi, "An adaptive control scheme for sistems with unknown actuator failures", *Automatica*, Vol. 38, 2002, pp. 1027-1034.

[17] J. Buffington, P. Chandler, M. Pachter, "Integration of on-line System Identification and optimization-based Control Allocation", *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Boston, 1998, AIAA-98-4487.

[18] J.B. Davidson, F.J. Lallman, W.T. "Bundick, Real-Time Adaptive Control Allocation applied to a High Performance Aircraft", *5th SIAM Conference on Control and Its Applications*, 2001.

[19] M.A. Bolender, 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.

[20] T.I.Fossen, S.P. Berge, "Nonlinear Vectorial Bakstepping design for Global Exponential Tracking of Marine Vessels in the Presence of Actuator Dynamics", *Proceedings of the 36th Conference on Decision and Control*, San Diego 1997, pp. 4237-4242.

[21] A.Casavola, E.Garone, "Adaptive actuators allocation for fault tolerant overactuated autonomous vehicles", TR-7-07, Univrsity of Calabria, DEIS, 2007.