Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

On-line references reshaping and control reconfiguration for non-minimum phase nonlinear fault tolerant control ⋆

M. Benosman, * **K.-Y. Lum** *

* *Temasek Laboratories, National University of Singapore, 5 Sports Drive 2, Singapore 117508 (E-mail: tslmb@nus.edu.sg)*

Abstract: In this paper we consider the problem of *graceful performance degradation*, for affine non-minimum phase nonlinear systems. The method is an optimization based scheme, that gives a constructive way to re-shape on-line the output reference for the post-fault system, and explicitly take into account the actuators and states saturations. The on-line output reference reshaping is associated with an on-line, MPC-based, controller reconfiguration, that forces the post-fault system to track the new output reference. The effect of FDD uncertainties on the on-line controller reconfiguration stability are studied, to ensure at least boundeness of the closed-loop system's states. The reshaping and reconfiguration schemes are applied to the Caltech ducted fan numerical example, which is described by a non-minimum phase nonlinear model.

Keywords: Nonlinear system, fault tolerant control, graceful performance degradation, trajectory reshaping, non-minimum phase.

1. INTRODUCTION

In this paper we present a simple idea to control a system in which component or actuator fault has occurred and is isolated/estimated. The idea is an extension of the paper by Jiang and Zhang [2006], where the authors proposed a fault tolerant control scheme for linear systems. There, the authors proposed a bank of reference models that the system should track, each of which corresponding to a particular faulty system. The references are chosen in such a way to be 'feasible' by the faulty system; i.e. output/state reference trajectories that can be tracked by the faulty system without saturation of the actuators. A controller, ensuring the closed-loop stability and reference model tracking, is computed off-line for each reference model. A reconfiguration mechanism is used on-line to switch between the different reference models and the corresponding controllers, depending on the fault detection and diagnosis (FDD) module. The 'faulty' reference model are designed by shifting the eigenvalues of the nominal-reference model towards the imaginary axis, using a state space realization scheme, and by adjusting the steady-state of the input commands. However, the selection of these faulty eigenvalues is not constructive. Moreover one*needs certain engineering insights into system performance limitations under different fault conditions*' (Jiang and Zhang [2006], page 286). Another, limitation of the scheme, is the fact that both the faulty reference models and their corresponding controllers are designed off-line, stored in a reference and controller bank and selected on-line through a switching module. The limitation comes from the fact that the number of reference-models/controllers will be directly proportional to the number of *expected* faults, which can rapidly lead to a cumbersome bank of models and controllers. Actually, the same authors stressed in Zhang and Jiang [2003], that a constructive way of dealing with what they call *graceful performance degradation* is still missing. To our knowledge this problem is still open, and we try here to extend the idea proposed in Jiang and Zhang [2006] to the nonlinear models case, through an optimization-based approach that gives a systematic and constructive way to design on-line a feasible output reference for the faulty system, explicitly taking into account the actuators and states saturations. When the output-reference trajectory has been reshaped on-line, the controller that tracks this trajectory is computed on-line as well, based on a nonlinear re-allocation scheme or pseudo-inverse. We present in this paper the reshaping scheme as well as the on-line control reconfiguration. First, we present in section 2 the models that we are studying here together with the adequate assumptions. In section 3, we present the reshaping method and the control reconfiguration scheme. We further discuss the stability of the closed-loop system. Then, in section 4, we apply the reshaping and the reconfiguration methods to a non-minimum phase flight system. Finally, we conclude with a discussion of the results obtained and some future research directions.

⋆ This work is supported by the Defence Science & Technology Agency POD51324.

2. PRELIMINARIES

2.1 Class of systems under study

We consider here affine nonlinear systems of the form:

$$\begin{aligned} \dot{x} &= f(x) + g(x)u \\ y &= h(x) \end{aligned} \tag{1}$$

where $x \in \mathbb{R}^n, u \in \mathbb{R}^{n_a}, y \in \mathbb{R}^m$ represent respectively the state, the input and the controlled output vectors. The vector fields f, columns of g and function h are supposed to satisfy the following classical assumptions.

Assumption(1): $f : \mathbb{R}^n \to \mathbb{R}^n$ and the columns of $g : \mathbb{R}^n \to \mathbb{R}^{n \times n_a}$ are smooth vector fields on a compact set X of $\mathbb{R}^n$ and $h(x)$ is a smooth function on X with $f(0) = 0, h(0) = 0$.

Assumption(2): System (1) has a well-defined (vector) relative degree $\{r_1, \ldots, r_m\}$ at each point $x^0 \in X$ (see e.g. Isidori [1989]).

Assumption(3): The system is fully or over-actuated, in the sense that the number of actuators is at least equal to the number of controlled outputs, i.e. $n_a \geq m$.

Assumption(4): The system is non-minimum phase, in the

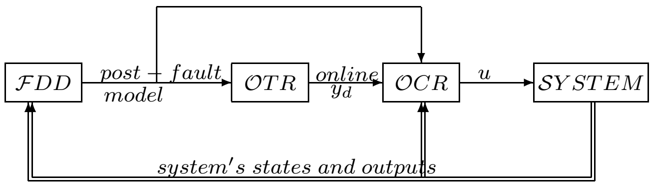

Fig. 1. Closed-loop system configuration

sense that it has internal dynamics, i.e. $|r| \triangleq \sum_{i=1}^{i=m} r_i < n$, and these dynamics are unstable in Lyapunov sense.
Assumption(5): We assume that assumptions 2-4 above are preserved after the occurrence of a fault in the system.

2.2 Control objectives

Find a controller u s.t. the nominal as well as the faulty system's output vector y tracks asymptotically a desired smooth feasible trajectory $y_d(t)$, while satisfying the actuators and states constraints:

$$
\begin{array}{l}
u \in \Omega \triangleq \left\{ u = (u_1, u_2, \cdots, u_{n_a})^T |\ u_i^- \le u_i \le u_i^+,\ i = 1, 2, \cdots, n_a \right\} \\
x \in X \triangleq \left\{ x = (x_1, x_2, \cdots, x_n)^T |\ x_i^- \le x_i \le x_i^+,\ i = 1, 2, \cdots, n \right\}
\end{array} \tag{2}
$$

where $u^- = (u_1^-, u_2^-, \cdots, u_{n_a}^-)^T$, $u^+ = (u_1^+, u_2^+, \cdots, u_{n_a}^+)^T$ and $x^- = (x_1^-, x_2^-, \cdots, x_n^-)^T$, $x^+ = (x_1^+, x_2^+, \cdots, x_n^+)^T$ are vectors of lower/upper actuators and states limits, respectively.
Assumption(6): We assume that the desired *nominal trajectory* is feasible by the *nominal safe system*, within its input/state limits.

3. NONLINEAR TRAJECTORY RESHAPING AND CONTROL RECONFIGURATION

The closed-loop configuration considered here consists of three main modules, as depicted on figure 1. The first one is the fault detection and diagnosis module (FDD) that detects and provides in real time a model of the actual post-fault system. The second one, *which is the main contribution of this work*, namely the on-line trajectory reshaping (OTR) module based on the post-fault model provided by the FDD, will generate on-line a suitable reference trajectory for the faulty system, within its actuator and state limits. Eventually, the on-line controller reconfiguration (OCR) block will use the new output reference provided by the OTR module and the actual system's model provided by the FDD block to reconfigure the controller on-line in such a way to track the new output reference, while ensuring closed-loop stability, i.e. at least boundeness of the states, subject to FDD uncertainties.

3.1 On-line trajectory reshaping

We present in this section the on-line OTR scheme. The idea is, whenever a fault is detected and estimated by the FDD block, the output trajectory will be reshaped on-line in such a way to be as close as possible to the nominal trajectory and to be feasible by the faulty system.
To do so, let's consider time output trajectories written in the canonical polynomial basis

$$
y_{dj}(t) = \sum_{i=1}^{i=l+1} a_{ij} \left(\frac{t - t_1}{t_2 - t_1}\right)^{(i-1)},\ j \in \{1, ..., m\} \tag{3}
$$

where l is the order of the polynomial, a_{ij} $i \in \{1, ..., l+1\}$ are the interpolation coefficients, and t_1, t_2 are the initial and the final interpolation times, respectively. The trajectory planner works as follows:
Assume that a fault (in the actuators or the components) occurs at $t = t_{1F}$ and that the FDD block detects this fault and identifies the equations of the faulty system as [1]

$$
\begin{array}{c}
\dot{x} = f_F(x) + g_F(x)u \\
y = h(x)
\end{array} \tag{4}
$$

where f_F, g_F hold for the modified vector field f and matrix g after the occurrence of the fault (see e.g.(Jiang and Zhang [2006], Zhang and Jiang [1999]) for some fault models). We stress that, regardless of the nature and the intensity of the components/actuators faults, we assume here that Assumption 5 still holds.
Trajectory generation consists in solving on-line the optimal problem

$$
\begin{array}{l}
\min\limits_{(a, t_{2F})} J = \min\limits_{(a, t_{2F})} \displaystyle\int_{t_{1F}}^{t_{2F}} (y_{nom}(t) - y_d(t))^T Q_1 (y_{nom}(t) - y_d(t)) dt \\
\qquad\qquad + \displaystyle\int_{t_{1F}}^{t_{2F}} u(t)^T Q_2 u(t) dt
\end{array} \tag{5}
$$

under the constraints

$$
\begin{array}{c}
\dot{x} = f_F(x) + g_F(x)u \\
y_d(t, b, t_{2F}) = h(x) \\
u^- \le u \le u^+ \\
x^- \le x \le x^+ \\
y^{(k)}(t_{1F}) \triangleq (y_1^{(k)}(t_{1F}), ..., y_m^{(k)}(t_{1F}))^T = y_{nom}^{(k)}(t_{1F}) \\
\triangleq (y_{nom_1}^{(k)}(t_{1F}), ..., y_{nom_m}^{(k)}(t_{1F}))^T, \\
y^{(k)}(t_{2F}) \triangleq (y_1^{(k)}(t_{2F}), ..., y_m^{(k)}(t_{2F}))^T = y_{nom}^{(k)}(t_{2nom}) \\
\triangleq (y_{nom_1}^{(k)}(t_{2nom}), ..., y_{nom_m}^{(k)}(t_{2nom}))^T,\ k = 0, ..., s \\
t_{2F} \ge t_{2nom}
\end{array} \tag{6}
$$

where
$y_d = (\sum_{i=1}^{i=l+1} a_{i1} (\frac{t - t_{1F}}{t_{2F} - t_{1F}})^{(i-1)}, ..., \sum_{i=1}^{i=l+1} a_{im} (\frac{t - t_{1F}}{t_{2F} - t_{1F}})^{(i-1)})^T$
$s \in \mathbb{N}^+$, $Q_1 \in \mathbb{R}^{m \times m}$, $Q_2 \in \mathbb{R}^{n_a \times n_a}$ are positive definite weight matrices, $a = (a_{(1)1}, .., a_{(l+1)1}, .., a_{(1)m}, ..., a_{(l+1)m})^T$ $\in \mathbb{R}^{m(l+1)}$ is the vector of the polynomials coefficients, t_{2F} is the final time motion for the optimal trajectory vector $y_d(t)$, and t_{2nom} is the final time motion for the nominal trajectory vector $y_{nom}(t) \triangleq (y_{nom_1}(t), ..., y_{nom_m}(t))^T$.
Let's consider now the optimization problem (5),(6), we can simplify its formulation by removing the $m(2s + 2)$ equality constraints in (6), as follows:
Rewrite the planned trajectories $y_{dj}(t)$ as

$$
y_{dj}(t) = \sum_{i=1}^{i=2s+2} a_{ij} \left(\frac{t - t_1}{t_2 - t_1}\right)^{(i-1)} + \sum_{i=2s+3}^{i=l+1} a_{ij} \left(\frac{t - t_1}{t_2 - t_1}\right)^{(i-1)} \tag{7}
$$

$\forall\ j \in \{1, ..., m\}$ and define the new vector

$$
b_j = (a_{2s+3j}, ..., a_{l+1j})^T.
$$

Then (7) writes as

$$
y_{dj}(t) = \sum_{i=1}^{i=2s+2} a_{ij} \left(\frac{t - t_1}{t_2 - t_1}\right)^{(i-1)} + \sum_{i=1}^{i=l-2s-1 \triangleq \tilde{l}} b_{ij} \left(\frac{t - t_1}{t_2 - t_1}\right)^{(i+2s+2)}. \tag{8}
$$

The trajectory $y_{dj}(t)$ can be expressed as functions of the vector b_j's components by using the $y_{dj}(t)$ given by (8) in

[1] We assume at the time being that the FDD block provides an exact model of the faulty system. However, later in sec. 3.2. we will study the effect of mismatches between the real system and the mathematical model due to FDD uncertainties, on the reconfigured systems's stability and performances.

the last $m(2s+2)$ equality constraints in (6), and solving for the coefficients $a_{ij},\ i \in \{1,...,2s+2\}, j \in \{1,...,m\}$. The trajectory reshaping problem writes now as

$$\min_{(b,t_{2F})} J(b, t_{2F}, u),\ b \in \mathbb{R}^{m(l-2s-1)} \tag{9}$$

under the constraints

$$\begin{array}{c} \dot{x} = f_F(x) + g_F(x)u \\ y_d(t, b, t_{2F}) = h(x) \\ u^- \leq u \leq u^+ \\ x^- \leq x \leq x^+ \\ t_{2F} \geq t_{2nom} \end{array} \tag{10}$$

Solution computation
To solve the optimization problem (9),(10), we first rewrite it as a problem depending directly on (b, t_{2F}). To do so, we write the control vector u as function of the optimization variables (b, t_{2F}) or equivalently, u as function of the planned output vector y_d.
Let's consider again the faulty system equations (4).
Under Assumptions 1,2,4 and 5, a diffeomorphism $\Phi(x) = (\xi, \eta)$ exists in a neighborhood of x^0, such that equations (4) can be transformed into

$$\begin{cases} y^{(r)}(t) = b_F(\xi(t), \eta(t)) + A_F(\xi(t), \eta(t))u(t) \\ \dot{\eta}(t) = q_F(\xi(t), \eta(t)) + p_F(\xi(t), \eta(t))u(t) \end{cases} \tag{11}$$

where

$$\begin{array}{l} y^{(r)}(t) \triangleq (y_1^{(r_1)}(t), \ldots, y_m^{(r_m)}(t))^T \\ \xi(t) = (\xi^1(t), \ldots, \xi^m(t))^T \\ \xi^j(t) = (y_j(t), \ldots, y_j^{(r_j-1)}(t)),\ \ 1 \leq j \leq m \end{array} \tag{12}$$

and $b_F,\ q_F,\ p_F$ write as functions of f_F, g_F, h and$A_F(x)$ is the nonsingular decoupling matrix (Isidori [1989], pp. 234-288). Next, computing the vector u from the first set of equations in (11) and substituting it into the internal dynamics equations associated with the desired references y_{dj}, yields

$$\dot{\eta} = q_F(\xi_d, \eta) + p_F(\xi_d, \eta)A_F^{-1}(\xi_d, \eta)(y_d^{(r)} - b_F(\xi_d, \eta)) \tag{13}$$

It is well known (Isidori [1989]) that, to analyze the behavior of the solutions of (13) associated with a given initial condition, we can analyze the behavior of the associated zero dynamics

$$\dot{\eta} = q_F(0, \eta) - p_F(0, \eta)A_F^{-1}(0, \eta)b_F(0, \eta) \tag{14}$$

If the equilibrium point $\eta = 0$ of (14) is asymptotically stable, the solutions of the initial-conditions problem (13) can be bounded by a decreasing function (Khalil [1996], p. 219, lemma 5.4), in which case the system is said to be minimum phase. In all other cases, the system is said to be non-minimum phase and no guarantee exists regarding the behavior of its internal dynamics, which is the case considered in this work.
Thus, u can be obtained directly from (11), and writes as

$$u(t) = A_F^{-1}(\xi_d(t), \eta(t))(y_d^{(r)}(t) - b_F(\xi_d(t), \eta(t))) \tag{15}$$

where $\eta(t)$ is a bounded time solution of the internal dynamics (13). However, particular care should be taken in obtaining bounded $\eta(t)$. Actually, the problem of finding a bounded solution to the internal dynamics for non-minimum phase systems is known as the stable inversion problem, and many solutions exists for the linear (Hunt and Meyer [1997], Benosman and Le Vey [2003]), and the nonlinear cases (Devasia et al. [1996], Benosman and Le Vey [2001]). By stable inversion, a bounded solution to (13) can be obtained, leading to bounded control (15). Now, the optimal problem (9),(10) can be written directly as a function of the optimization vector (b, t_{2F})

$$\begin{array}{c} \min\limits_{(b,t_{2F})} J(b, t_{2F}) \\ u^- \leq u(b, t_{2F}) \leq u^+ \\ x^- \leq x(b, t_{2F}) \leq x^+ \\ t_{2F} \geq t_{2nom}, \end{array} \tag{16}$$

which can be solved using available nonlinear constrained optimization codes.

3.2 On-line controller reconfiguration

We present here the approach for OCR. We first construct a nominal virtual control for the safe system. Next, assuming a fault occurs at $t = t_F$ and is instantly identified by FDD [2] , the virtual control is re-allocated online according to the post-fault model given by FDD, to the surviving actuators while minimizing tracking errors and satisfying actuator/state limits.
Based on equation (11), we can define a virtual input as [3]

$$b_N(\xi(t), \eta(t)) + A_N(\xi(t), \eta(t))u_N(t) = v(t). \tag{17}$$

Combining (11) and (17), we obtain the linear (virtual) input-output mapping

$$y^{(r)}(t) = v(t). \tag{18}$$

Based on the linear system (18), we propose the stabilizing output feedback

$$\begin{array}{l} v_s(t, \xi) = y_{nom}^{(r)} - K_r(y^{(r-1)} - y_{nom}^{(r-1)}(t)) - ... - K_1(y - y_{nom}(t)) \\ K_i > 0,\ i = 1, ..., r. \end{array} \tag{19}$$

Defining the tracking error vector as $e(t) = y(t) - y_{nom}(t)$, we obtain the tracking error dynamic

$$e^{(r)}(t) + K_r e^{(r-1)}(t) + ... + K_1 e(t) = 0. \tag{20}$$

By tuning the gain matrices $K_i,\ i = 1, ..., r$ such that all the polynomials [4]

$$e_i^{(r)}(t) + K_r(i,i)e_i^{(r-1)}(t) + ... + K_1(i,i)e_i(t),\ i = 1, ..., m$$

are Hurwitz, we obtain asymptotic convergence of $e(t)$ to zero. We then re-allocate on-line the virtual controller $v_s(t, \xi(t))$, to the actuators of the nominal system by solving the receding horizon optimal problem [5] (21) , where t_H is a finite integration time horizon, $Q_3 \in \mathbb{R}^{m\times m}$ is a positive definite weight matrix, and $Q_4 \in \mathbb{R}^{n-|r|\times n-|r|}$ is a positive definite weight matrix introduced with the internal dynamics tracking-error $e_\eta = \eta - \eta_d$. Indeed, we saw in section 3.1 that the internal dynamics are unstable in Lyapunov sense, and that a stable inversion technique should be used in this case to complete the output trajectory generation presented in section 3.1. This stable inversion will then provide us with the desired internal dynamics trajectories η_d when solving equation (13) in the output planning problem, as described in section 3.1.
Next, we consider a fault in the system at the instant $t = t_F$, from which point onward, the system will be described by equation (4). We assume that the FDD provides an estimate of the fault value noted $\tilde{f}$, i.e. either

[2] We stress here, that we do not consider in this note explicitly the FDD block synthesis for nonlinear systems, that can be found for example in Garcia and Frank [1997]. Instead we assume the availability of FDD module and we study both cases; first when the FDD provides us with a precise post-fault model and, the realistic case; when the FDD gives with some delay an imprecise post-fault model Zhang and Jiang [2006].
[3] Where the subscripts N indicate that we are dealing with the nominal safe system.
[4] Below $K_j(i,i)$ denotes the element $(.)(i,i)$ of the matrix K_j.
[5] Hereafter $||.||$ denotes the euclidian norm and $||x||_P = (x^T P x)^{1/2}$.

$$
\mathcal{P}_1(t_k,\xi_k,\eta_k)\begin{cases}\min\limits_{u_N}\displaystyle\int_{t_k}^{t_k+t_H}((b_N(\xi,\eta)+A_N(\xi,\eta)u_N(t)-v_s(t,\xi))^TQ_3(b_N(\xi,\eta)+A_N(\xi,\eta)u_N(t)-v_s(t,\xi))+(\eta-\eta_d)^TQ_4(\eta-\eta_d))dt\\ y^{(r)}(t)=b_N(\xi,\eta)+A_N(\xi,\eta)u_N(t)\\ \dot\eta(t)=q_N(\xi,\eta)+p_N(\xi,\eta)u_N(t)\\ u^-\le u_N(t)\le u^+\\ (\xi^-,\eta^-)=\Phi(x^-)\le(\xi,\eta)\le(\xi^+,\eta^+)=\Phi(x^+)\\ ||e(t_{k+1})||_{P_\xi}\le\alpha_\xi||e(t_k)||_{P_\xi},\ \ \alpha_\xi\in[0,1],\ P_\xi>0\\ ||e_\eta(t_{k+1})||_{P_\eta}\le\alpha_\eta||e_\eta(t_k)||_{P_\eta},\ \ \alpha_\eta\in[0,1],\ P_\eta>0,\ k=0,1,...\end{cases} \tag{21}
$$

$$
\mathcal{P}_2(t_k,\xi_k,\eta_k)\begin{cases}\min\limits_{u_F}\max\limits_{\tilde f\in\tilde F}\displaystyle\int_{t_k}^{t_k+t_H}((b_F(\xi,\eta)+A_F(\xi,\eta)u_F(t)-v_s(t,\xi))^TQ_3(b_F(\xi,\eta)+A_F(\xi,\eta)u_F(t)-v_s(t,\xi))+(\eta-\eta_d)^TQ_4(\eta-\eta_d))dt\\ y^{(r)}(t)=b_F(\xi,\eta)+A_F(\xi,\eta)u_F(t)\\ \dot\eta(t)=q_F(\xi,\eta)+p_F(\xi,\eta)u_F(t)\\ u^-\le u_F(t)\le u^+\\ (\xi^-,\eta^-)=\Phi(x^-)\le(\xi,\eta)\le(\xi^+,\eta^+)=\Phi(x^+)\\ ||e(t_{k+1})||_{P_\xi}\le\alpha_\xi||e(t_k)||_{P_\xi},\ \ \alpha_\xi\in[0,1],\ P_\xi>0\\ ||e_\eta(t_{k+1})||_{P_\eta}\le\alpha_\eta||e_\eta(t_k)||_{P_\eta},\ \ \alpha_\eta\in[0,1],\ P_\eta>0,\ k=0,1,...\end{cases} \tag{22}
$$

parameter or actuator fault. We then re-allocate on-line the virtual controller $v_s(t,\xi(t))$, to the actual actuators by solving the min-max optimal problem (22) where v_s writes in this case as

$$
\begin{array}{l} v_s(t,\xi)=y_d^{(r)}(t)-K_r(y^{(r-1)}(t)-y_d^{(r-1)}(t))-...-K_1(y(t)-y_d(t))\\ K_i>0, i=1,...,r\end{array} \tag{23}
$$

$y_d(t)$ is the on-line re-shaped trajectory vector, $\tilde F=[\tilde f-\triangle\tilde f,\ \tilde f+\triangle\tilde f]\subset\mathbb{R}$ where $\triangle\tilde f>0$ represents the tolerable fault value excursion, $Q_3,\ Q_4,\ \eta_d(t)$ defined as before.

Remark 1: The virtual controllers used for the nominal and for the faulty case differ only by the desired output trajectories, i.e. $y_{nom}(t)$ in (19) and $y_d(t)$ in (23). However, both controls lead to the same error dynamics (20), which means that we are imposing the same transient behavior for both systems. In this sense this controller can be considered as an extension of the well known pseudo-inverse control, e.g. Gao and Antsaklis [1991]. ♦
We can write now the reconfiguration algorithm.

Reconfiguration Algorithm The algorithm holds as follows:
1- Initialization:
- Initial conditions: (ξ_0,η_0) *at* t_0.
- Controller parameters: $K_1,K_2,...,K_r$ *in (19), weight matrices* Q_3,Q_4,P_ξ,P_η *in* $\mathcal{P}_1$, $\mathcal{P}_2$ *optimization problems, the optimization horizon* t_H *and the sampling time* T, *the fault value excursion* $\triangle\tilde f$, *the input/states bounds* $u^-,u^+,\xi^-,\xi^+,\eta^-,\eta^+$, *the contractive parameters* α_ξ,α_η.
2- Control
- Step 1: set $k=0$.
- Step 2: solve $\mathcal{P}_1(t_k,\xi_k,\eta_k)$, *this gives the solution* $u(t),\ t\in[t_k,t_k+t_H]$.
- Step 3: apply u for $t\in[t_k,t_{k+1}]$ *and measure* $\xi_{t_{k+1}},\eta_{t_{k+1}}$ *at* $t_{k+1}=t_k+T$
- Step 4: test the FDD block, if system safe put $k=k+1$ *and go to step 2.*
- Step 5: if fault $\tilde f$ *detected solve* $\mathcal{P}_2(t_k,\xi_k,\eta_k)$ *, this gives the solution* $u(t),\ t\in[t_k,t_k+t_H]$*, go to Step 3.*

Remark 2: In real applications, we expect that the FDD block reacts with a time delay, due for example to fault estimation delay (Zhang and Jiang [2000]), or that the post-fault model is imprecise. In these situations the algorithm can still ensure the boundeness of the system's states and inputs, due to the robustness of the optimal receding horizon reconfiguration, as we will see below (Lemma 2).♦

Stability analysis The stability analysis is straightforward, and is mainly due to the contractive constraints on the tracking errors, present in the optimization problems $\mathcal{P}_i,\ i=1,2$. Firstly, let's introduce some assumptions, necessary for the analysis that will follow.
Assumption(7): The system is controllable along the desired output trajectories.
Assumption(8): We assume that there exists a $\rho\in]0,\infty[$ such that for all $x_{t_0}\in\mathbf{B}_\rho\overset{\triangle}{=}\{x_{t_0}\in\mathbb{R}^n|\ ||e(x_{t_0})||_{P_\xi}=||h(x_{t_0})-y_d(t_0)||_{P_\xi}\le\rho\ and\ ||e_\eta(x_{t_0})||_{P_\eta}=||\eta(x_{t_0})-\eta_d(t_0)||_{P_\eta}\le\rho\}$, the optimal problems $\mathcal{P}_i,\ i=1,2$ have a solution.
Assumption(9): There exists constants $\beta_\xi,\beta_\eta\ \in]0,\infty[$ such that $||e(t)||_{P_\xi}\le\beta_\xi||e(t_k)||_{P_\xi},\ and\ ||e_\eta(t)||_{P_\eta}\le\beta_\eta||e_\eta(t_k)||_{P_\eta}\ \forall t\in[t_k,\ t_{k+1}],\ k=0,1,...$

- Stability in the nominal case: We can state now a Lemma summarizing the stability results for the control reconfiguration algorithm.

Lemma 1. Under Assumption 1 to 9, choosing $\alpha_\xi,\ \alpha_\eta\in]0,1[$ and $\rho,\beta>0$, the reconfiguration algorithm implies an exponential convergence of $e_\xi,\ e_\eta$ to zero.

Proof: refer to Benosman and Lum [2007].

Up to now, we have considered that the nominal as well as the faulty models were 'exact'. However, in real applications, we should expect some errors in the nominal model and the faulty models (Zhang and Jiang [2000]), due for instance to delays in the fault detection/estimation (Garcia and Frank [1997]) and fault estimation uncertainties (Zhang and Jiang [2006]). Thus we analyze hereafter the effect of mismatches between the system's model used in the reconfiguration algorithms and the actual system's plant.

- Stability in the uncertain case: Here we will use the subscript R to describe the real plant contrary to the mathematical model, which will be denoted by the subscript M. Note that the mathematical model will be either the nominal one or the post-fault model. Let's first rewrite the model equations in a more compact form. Let's define the vector $\tilde x=(\xi,\eta)^T$. Then, the equations of the model

used in the definitions of the optimal problems $\mathcal{P}_i,\ i = 1, 2$ can be rewritten as:

$$\dot{\tilde{x}} = l_M(\tilde{x}) + H_M(\tilde{x})u \tag{24}$$

where, $l_M,\ h_M$ write as function of[6] $b_{N\ orF},\ A_{N\ orF},\ q_{N\ orF},\ p_{N\ orF}$. Let's denote by $\tilde{x}_M$ the solution of (24) with the initial condition $\tilde{x}_R(t_0)$ the actual measured plant state.
The actual real plant will be described by

$$\dot{\tilde{x}} = l_R(\tilde{x}) + H_R(\tilde{x})u \tag{25}$$

and the real state value will be denoted by $\tilde{x}_R$ which are the solutions of (25) associated with the real-plant initial conditions $\tilde{x}_R(t_0)$.
The goal of this analysis is to evaluate the distance between the states trajectories obtained when applying the reconfiguration control algorithm to the model (24) and the states trajectories obtained when applying the same control algorithm to the real system described by (25). To do so we need the following assumption.
Assumption(10): The mathematical model and the real system's model satisfy the Lipshitz-like inequalities

$$\begin{aligned} ||l_R(\tilde{x}_1) + h_R(\tilde{x}_1)u_1 - l_R(\tilde{x}_2) - h_R(\tilde{x}_2)u_2|| \leq L_R||\tilde{x}_1 - \tilde{x}_2|| \\ +L_R||u_1 - u_2||,\ L_R > 0 \\ ||l_M(\tilde{x}) + h_M(\tilde{x})u - l_R(\tilde{x}) - h_R(\tilde{x})u|| \leq K||\tilde{x}|| + K||u||,\ K > 0 \end{aligned} \tag{26}$$

We then have the following Lemma:
Lemma 2. Under Assumption 10, for given $L_R,\ K > 0$, the reconfiguration algorithm implies that

$$||\tilde{x}_R(t) - \tilde{x}_M(t)|| \leq KT\delta e^{L_R T},\ \forall t > 0 \tag{27}$$

where $\delta = \max_{\Phi(x^-)\leq\tilde{x}_M\leq\Phi(x^+)} ||\tilde{x}_M|| + \max_{u^-\leq u\leq u^-} ||u||$, and T is the sampling time; i.e. $T = t_{k+1} - t_k,\ \forall k$.

Proof: refer to Benosman and Lum [2007].

4. NUMERICAL EXAMPLE

In this section we present simulation results obtained on a fast dynamic system, namely the Caltech ducted fan, which is an experimental test-bed that replicates qualities of actual flight UAVs (Dunbar et al. [2002]). We use here the planar model of these test-bed, as described in Dunbar et al. [2002].The dynamical model is given by

$$\begin{aligned} &m\ddot{x} - F_{xb}cos(\theta) - F_{zb}sin(\theta) = 0 \\ &m\ddot{z} + F_{xb}sin(\theta) - F_{zb}cos(\theta) = mg_{eff} \\ &J\ddot{\theta} - F_{zb}I_\tau = 0 \end{aligned} \tag{28}$$

where x and z represent horizontal and vertical inertial translations respectively, as depicted in figure 2. θ is the rotation of the ducted fan about the boom axis. F_{xb}, F_{zb} are the thrust vectoring body forces. $m = 12.5\ kg$ is the mass of the engine, $J = 0.25\ kg.m^2$ its moment of inertia about the boom, $I_\tau = 0.35\ m$ is the distance from the centre of mass along the X_b axis to the effective application point of the thrust vectoring force and $mg_{eff} = 7$ is the effective gravity. We define the state vector $\mathbf{x} = (x, \dot{x}, z, \dot{z}, \theta, \dot{\theta})^T$ and the control vector $u = (F_{xb}, F_{zb})^T$.
The control objective is to track a desired time trajectory for the output $y = x$ and to satisfy the state constraint $-1\ m \leq z \leq 1\ m$ and the input constraints $0\ N \leq F_{xb} \leq 13\ N,\ -6\ N \leq F_{xb} \leq 6\ N$.
The relative degree with respect to this output is clearly 2. We then have internal dynamics of dimension 4, which are basically the second and third ODE in equation (28). Several solutions may exist for these ODEs; we chose to fix

[6] N or F depending on if the mathematical model is the nominal or the faulty model.

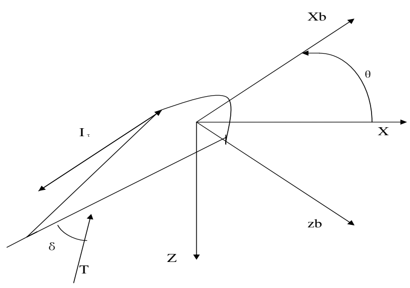

Fig. 2. Ducted fan frames

the vertical displacement and velocity $z, \dot{z}$ to zero and then study the behavior of the remaining internal dynamics $\theta, \dot{\theta}$. In this case, the zero dynamics are a 'pendulum-like' dynamics that have hyperbolic saddle points at $\theta = (4k + 1)\pi/2\ rad,\ \dot{\theta} = 0\ rad/sec,\ k \in N$, implying a non-minimum phase behavior.
Let's consider now the 'nominal' control problem, i.e. without faults. Our goal is to force the output $y = x$ to track the desired polynomial time trajectory

$$x_d = \begin{cases} x_f(6(\frac{t-t_0}{t_f})^5 - 15(\frac{t-t_0}{t_f})^4 + 10(\frac{t-t_0}{t_f})^3),\ t_0 \leq t \leq t_f, \\ x_f = 10\ m, \\ t_0 = 0,\ t_f = 20\ sec. \end{cases} \tag{29}$$

To solve the tracking problem in this case we will use the MPC formulation in equation (21). To formulate this problem we need first to solve a stable inversion problem to obtain the desired internal dynamics trajectories $\eta_d(t)$. As mentioned before, we chose here to force the first internal dynamics $z, \dot{z}$ to zero, i.e. the aircraft should stay at the constant zero altitude, and apply a stable inversion to the corresponding θ internal dynamics. We use here the stable inversion approach proposed in Benosman and Le Vey [2001] for Lagrangian systems, where the authors showed that stable internal dynamics trajectories can be obtained by formulating the problem as a two points boundary value problem. We chose here to solve the two points boundary value problem

$$\begin{cases} J\ddot{\theta} - I_\tau msin(\theta)\ddot{x}_d + I_\tau cos(\theta)mg_{eff} = 0, \\ \theta(t_0) = \theta(t_f) = \pi/2\ rad\ . \end{cases} \tag{30}$$

Next, the problem $\mathcal{P}_1(t_k, \xi_k, \eta_k)$ in equation (21), has been solved with the following parameters: $K_1 = 2500$, $K_2 = 0.3,\ Q_3 = 1,\ Q_4 = \begin{bmatrix} 3000 & 0 & 0 & 0 \\ 0 & 0.3 & 0 & 0 \\ 0 & 0 & 4000 & 0 \\ 0 & 0 & 0 & 0.3 \end{bmatrix},\ \alpha_\xi = 0.9,$ $\alpha_\eta = 0.9$.
To solve the receding horizon optimization problem, we have used the solver 'fmincon' of Matlab. To do so, we have first discretized the functional optimization problem into a finite-dimension optimization problem, by writing the states trajectories $x,\ z,\ \theta$ as polynomials of degrees $6,\ 4,\ 4$ respectively and solving the optimization problem with respect to the coefficients of these polynomials. It is worth noting that the first two coefficients of each polynomial were used to ensure continuity of the trajectories and their first derivatives, i.e. $x(t_{k-1}) = x(t_k),\ \dot{x}(t_{k-1}) = \dot{x}(t_k)$. Similarly for z and θ. Then effectively the optimization space dimension is 8, the number of the remaining coefficients. The receding horizon chosen is $t_H = T = 0.5\ sec$.
Fault scenario: We consider now a scenario where an actuator fault occurs at time $t_F = 10\ sec$, the fault being

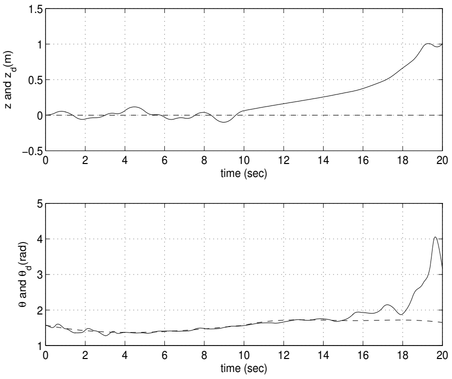

Fig. 3. Desired trajectory (dashed line) and the actual altitude (continuous line)- Faulty case without OTR

a loss of effectiveness in the body force F_{xb}, modelled by introducing the multiplicative coefficient α in the ducted fan model as follows:

$$\begin{aligned} &m\ddot{x} - \alpha F_{xb}cos(\theta) - F_{zb}sin(\theta) = 0 \\ &m\ddot{z} + \alpha F_{xb}sin(\theta) - F_{zb}cos(\theta) = mg_{eff} \\ &J\ddot{\theta} - F_{zb}I_{\tau} = 0. \end{aligned} \tag{31}$$

We have chosen to test fault $\alpha = 0.544$, which implies that the force F_{xb} will have to be almost double of the nominal one to achieve the same tracking quality of the nominal trajectory x_d. This means that if nothing is done, this actuator will saturate, and this may lead to the loss of the tracking or worse to the loss of the aircraft stability, as we will see below.

1- *With model switching and without trajectory reshaping:* We consider here the case where we assume that FDD module provides the exact value of the fault, after a delay of 2 *sec* (refer to Benosman and Lum [2007] for the case with an imprecise estimation). The MPC controller switches then to the faulty model[7] at the instant $t_s = 12$ *sec*. However, we do not use the OTR module here, and the MPC based on the post-fault model keeps trying to track the nominal desired output trajectory. The obtained results are presented in figure 3. We can see that the obtained internal dynamics are not acceptable, i.e. θ reaches the maximal value of 4 *rad*! and the altitude reaches its limit of 1 m.

2- *With model switching and with trajectory reshaping:* In addition to the setting of the previous case here The OTR module starts at $t_s = 12$ *sec* computing the best trajectory for the faulty system and the MPC based on the faulty model switches from tracking the nominal trajectory to tracking the optimal trajectory. We first start by presenting the results of the OTR block. We applied the reshaping scheme presented in section 3.1. Again to keep the optimization time acceptable for real time application, we used small values for the dimension of the vector b in equation (9). Figure 4 shows the obtained optimal trajectories for $dim(b) = 1$ and $dim(b) = 2$. The optimal trajectories are very close to the nominal trajectory, due to the first term in the optimization cost (5). However, the control forces necessary to track this optimal trajectories, computed by direct model inversion (equations (15) with the internal dynamics stable inversion of (13)), are all very close to the actuator limit of $13N$,

[7] We stress here that for computation time reason we did not use here the min-max MPC formulation of equation (22), we used instead a min formulation for equation (22), i.e. we do not take into account the excursion over $\tilde{f}$, but this does not affect the stability results that are solely due to the contractive constraints and not to the optimization cost.

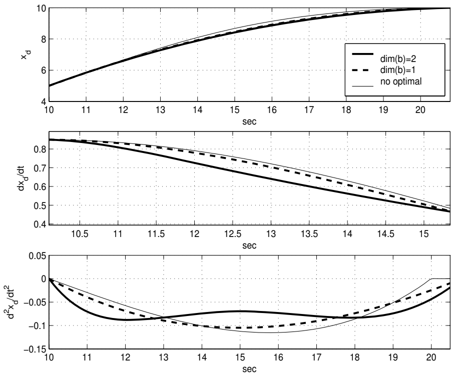

Fig. 4. Optimal trajectories for the faulty model: nominal trajectory (continuous thin line), optimal with $dim(b) = 1$ (dashed thick line), optimal with $dim(b) = 2$ (continuous thick line)

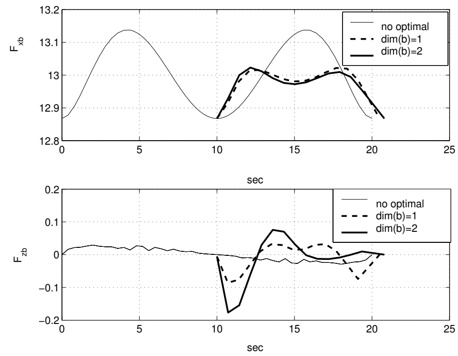

Fig. 5. Optimal control forces for the faulty model: nominal trajectory (continuous thin line), optimal with $dim(b) = 1$ (dashed thick line), optimal with $dim(b) = 2$ (continuous thick line)

as we can see in figure 5, whereas the nominal trajectory necessitates control forces which clearly violate this limit. We can expect then that the controller will be more able to reproduce these trajectories than the nominal one. The optimal value obtained are:

- for $dim(b) = 1$: $t_{2F} = 20.49$ *sec*, $b = 7e-6$, computation time= 1.54 *sec*.
- for $dim(b) = 2$: $t_{2F} = 20.79$ *sec*, $b = (0.47e-4,\ -38e-8)^T$, computation time= 2.025 *sec*.

For the tests we used $dim(b) = 2$, which means that the MPC based on the faulty model switches to the optimal faulty trajectory at $t_s = 14.025$ *sec*. The simulation results are reported in figures 6and 7. We clearly see that the tracking is achieved, since the ducted fan switches to the optimal faulty trajectory which is still very close to the nominal one, and thus reaches the same final point without exceeding the actuators limits, as we can see from figure 8, while keeping the system stable, as we can see from the internal dynamics plot 7.

5. CONCLUSION

The goal of this work was to extend previous results on acceptable performance degradation obtained in the linear case (Jiang and Zhang [2006]), to the class of non-minimum phase nonlinear systems affine in the control. We formulated the performance degradation problem as a nonlinear optimization problem, permitting to generate on-line output trajectories that are feasible by the faulty

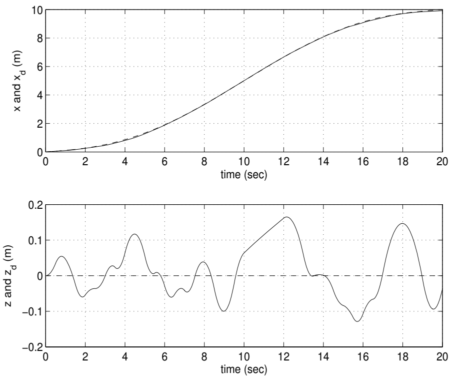

Fig. 6. Desired trajectory (dashed line) and the actual output (continuous line)- Faulty case with FDD delay and with OTR

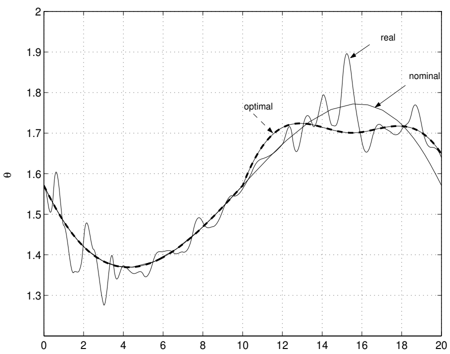

Fig. 7. Desired and actual θ trajectories- Faulty case with FDD delay and with OTR

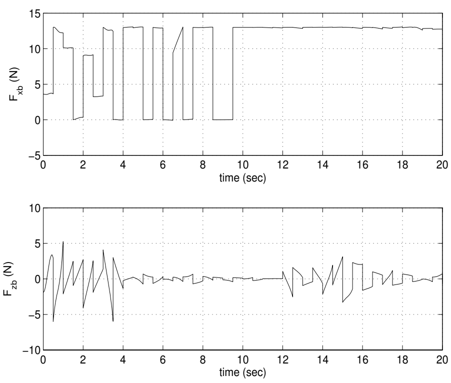

Fig. 8. Control force F_{xb}, F_{zb}- Faulty case with FDD delay and with OTR

system within its control and state limits. We showed how this on-line optimization can be solved for non-minimum phase models via stable inversion to ensure that the internal dynamics are feasible by the system. This on-line trajectory reshaping has been implemented with a contractive MPC algorithm to ensure the tracking of the generated trajectories. We proved that the contractive MPC can be used for trajectory tracking purposes and not just for regulation as is more common in the literature. We showed in a numerical example that both the trajectory generation and the MPC tracking-algorithm can be used for on-line application on fast dynamic systems, like aircrafts. Although the numerical tests were satisfactory, we expect better results when using faster solvers for the on-line optimization problems, thus we think that these algorithms can effectively be used for practical applications if some available fast optimization software like NPSOL (Dunbar et al. [2002]), are used together with more powerful computers. We also pointed out that the goal of this work was the performance degradation problem, thus we did not explicitly use an FDD block in this study. However, we have considered explicitly in the stability analysis the effect of a real FDD, namely the post-fault models uncertainties due to fault estimation delays and imprecisions. Future work concerns the explicit integration of FDDs for affine nonlinear systems, e.g. Garcia and Frank [1997], with the on-line reshaping and reconfiguration method proposed here.

REFERENCES

M. Benosman and G. Le Vey. Stable inversion of a non minimum phase system through nonlinear boundary value problem formulation. In *IFAC, NOLCOS 2001 Proc*, St Petersbourg, Russia, 2001.

M. Benosman and G. Le Vey. Stable inversion of SISO non minimum phase linear dynamic through output planning : an experimental application to the one-link flexible manipulator. *IEEE, Transactions on Control Systems Technology*, 11(4):588–597, 2003.

M. Benosman and K.Y. Lum. On-line references reshaping and control re-allocation for nonlinear fault tolerant control. Technical report, Temasek Laboratories-NUS, August 2007.

S. Devasia, D. Chen, and B. Paden. Nonlinear inversion-based output tracking. *IEEE, Transactions on Automatic Control*, 41(7):930–942, July 1996.

W.B. Dunbar, M.B. Milam, R. Franz, and R.M. Murray. Model predictive control of a thrust-vectored flight control experiment. In *IFAC*, Spain, 2002.

Z. Gao and P.J. Antsaklis. Stability of pseudo-inverse method for reconfigurable control systems. *Int. Journal of Control*, 53(3):717–729, 1991.

E.A. Garcia and P.M. Frank. Deterministic nonlinear observer-based approaches to fault diagnosis: A survey. *Control Engineering Practice*, 5(5):663–670, 1997.

L.R. Hunt and G. Meyer. Stable inversion for nonlinear systems. *Automatica*, 33(8):1549–1554, 1997.

A. Isidori. *Nonlinear Control Systems*. Communications and Control Engineering Series. Springer-Verlag, 2 edition, 1989.

J. Jiang and Y. Zhang. Accepting performance degradation in fault-tolerant system design. *IEEE, Transactions on Control Systems Technology*, 14(2):284–292, March 2006.

H.K. Khalil. *Nonlinear systems*. New York Macmillan, second edition, 1996.

Y. Zhang and J. Jiang. Design of integrated fault detection, diagnosis and reconfigurable control systems. In *IEEE, Conference on Decision and Control*, pages 3587–3592, 1999.

Y. Zhang and J. Jiang. Integrated design of reconfigurable fault-tolerant control systems. *Journal of Guidance, Control, and Dynamics*, 24(1):133–136, 2000.

Y. Zhang and J. Jiang. Bibliographical review on reconfigurable fault-tolerant control systems. In *Proceeding of the 5th IFAC symposium on fault detection, supervision and safety for technical processes*, pages 265–276, Washington DC, 2003.

Y. Zhang and J. Jiang. Issues on integration of fault diagnosis and reconfigurable control in active fault-tolerant control systems. In *6th IFAC Symposium on fault detection supervision and safety of technical processes*, pages 1513–1524, China, August 2006.