Nullspace-based input reconfiguration architecture for over-actuated aerial vehicles

Tamás Péni and Bálint Vanek, *Member, IEEE,* György Lipták, Zoltán Szabó, *Member, IEEE,* József Bokor, *Fellow, IEEE*

***Abstract*—A dynamic input reconfiguration architecture is proposed for over-actuated aerial vehicles to accommodate actuator failures. The method is based on the dynamic nullspace computed from the linear parameter-varying model of the plant dynamics. If there is no uncertainty in the system, then any signal filtered through the nullspace has no effect on the plant outputs. This makes it possible to reconfigure the inputs without influencing the nominal control loop and thus the nominal control performance. Since the input allocation mechanism is independent of the structure of the baseline controller, it can be applied even if the baseline controller is not available in analytic form. The applicability of the proposed algorithm is demonstrated via the case study of designing a fault tolerant controller for the Rockwell B-1 Lancer aircraft.**

***Index Terms*—fault tolerant control, null space computation, linear parameter varying systems, control input reallocation, aircraft control**

I. Introduction

The most stringent requirements for improved reliability and environmental sustainability of safety critical flight control systems could only be satisfied with the most advanced fault tolerant control (FTC) techniques [1], [2]. The FTC system is required to detect and identify the failure and then to compensate its effect by reconfiguring the control system [3]. Focus on the environmental impact of the aircraft triggers the need for higher performance flight control systems, which leads to a paradigm shift from robust passive FTC towards active methods relying on switching, gain scheduled or linear parameter-varying (LPV) methods with certifiable algorithms [4]. In the past few years a wide variety of FTC design approaches have been proposed [5], [6].

This paper focuses on the reconfiguration task in the case of actuator failures. The approach considered here is the control input reallocation, where the aim is to compensate the actuator failures by reconfiguring the remaining flight control surfaces such that the forces and moments required by the flight controller can be generated. A broad picture on the state-of-the-art control allocation methods can be found in [7]. One class of the several approaches developed so far is formed by the nullspace (or kernel) based methods that can be applied if control input redundancy is available in the system [8]. As in aerospace applications this is often the case [9], this approach is a promising method for fault tolerant flight control design.

The classical kernel-based algorithms assume constant input direction matrices and thus use static matrix kernels [7], [8]. This concept is extended in [10] and [11] by using dynamic annihilators. Though the reconfiguration architecture in these papers is similar to that is proposed here, the algorithms in [10] and [11] highly depend on the linear time invariant (LTI) framework and thus their extension to the parameter varying case rises several theoretical problems. This paper proposes a different approach, which is promising for LPV applications as well.

The main component of the proposed reconfiguration architecture is the nullspace of the LPV system to be controlled. Although the nullspace of a dynamical system bears significant importance on several other fields of control as well [12],[13],[14] its numerical computation has been solved only partially so far. In [15] an algorithm based on matrix pencils is proposed to compute the dynamic kernel of an LTI system. Although this approach is computationally efficient, it is based on frequency domain formulation, which prevents its extension to LPV systems. In [12] the nullspace of a parameter-dependent, memoryless matrix is addressed in connection with controller design. The paper uses linear-fractional representation (LFR), but it does not consider the case of dynamic kernels. Moreover, the method of [12] does not analyze and thus cannot guarantee the well-definedness of the kernel basis, which are also necessary to use the kernel in any further design process. The LFR-Toolbox [16] also provides a method for nullspace computation. The algorithm is similar to the one given in [12] and it can be applied for dynamical systems as well, but it does not work for the general case as it requires certain rank conditions to be satisfied. In this paper we revise the existing methods and assemble a complete tool for kernel computation that can be applied for parameter-dependent matrices and LTI/LPV dynamical systems as well.

The paper is structured as follows, Section II is devoted to the numerical computation of the parameter varying nullspace. The proposed actuator reconfiguration architecture is discussed in Section III. The simulation example, using the B-1 aircraft is presented in Section IV, while the paper is concluded in Section V.

The authors are with the Systems and Control Lab, Institute for Computer Science and Control of Hungarian Academy of Sciences, H-1111, Kende u. 13-17, Budapest, Hungary. Corresponding author: Tamás Péni, e-mail: peni.tamas@sztaki.mta.hu

The research leading to these results has received funding from the European Union's Seventh Framework Programme (FP7/2007 − 2013) under grant agreement no. FP7-AAT-2012 − 314544. This paper was also supported by the János Bolyai Research Scholarship of the Hungarian Academy of Sciences.

II. Nullspace computation for parameter-varying systems

In this section the definition of the nullspace of a linear map is defined and the problem related to its numerical computation is addressed.

A. Nullspace of a linear map

Definition 1 (nullspace and its generator)**.** Let L denote a linear map assigning to the elements of a linear input space $\mathcal{U}$ the elements of a linear output space $\mathcal{Y}$. The (right) nullspace (kernel) of L, denoted by $\mathcal{N}(L)$, is a subset (subspace) of $\mathcal{U}$ that are mapped to 0, i.e. $\mathcal{N}(L) = \{u \in \mathcal{U} \mid Lu = 0\}$. If for some linear map $N : \mathcal{W} \to \mathcal{U}$, $\mathcal{N}(L) = \mathrm{Im}(N)$, then N is called the *generator* of the nullspace.

The following lemma gives the basis of the nullspace construction algorithms derived afterwards.

Lemma 1. *Let $L : \mathcal{U} \to \mathcal{Y}$ and $Q : \mathcal{U} \to \mathcal{Z}$ be two linear maps such that $\left[\begin{smallmatrix} L \\ Q \end{smallmatrix}\right] : \mathcal{U} \to \mathcal{Y} \times \mathcal{Z}$ is invertible. Then $\mathcal{N}(L) = \mathrm{Im}\left(\left[\begin{smallmatrix} L \\ Q \end{smallmatrix}\right]^{-1} \left[\begin{smallmatrix} 0 \\ I \end{smallmatrix}\right]\right)$.*

Proof. $\Leftarrow$) Let $z \in \mathrm{Im}\left(\left[\begin{smallmatrix} L \\ Q \end{smallmatrix}\right]^{-1} \left[\begin{smallmatrix} 0 \\ I \end{smallmatrix}\right]\right)$, i.e. $z = \left[\begin{smallmatrix} L \\ Q \end{smallmatrix}\right]^{-1} \left[\begin{smallmatrix} 0 \\ I \end{smallmatrix}\right] r$, with some r. If $[M \ \ N] = \left[\begin{smallmatrix} L \\ Q \end{smallmatrix}\right]^{-1}$, then $Lz = L[M \ \ N] \left[\begin{smallmatrix} 0 \\ I \end{smallmatrix}\right] r = [I \ 0] \left[\begin{smallmatrix} 0 \\ I \end{smallmatrix}\right] r = 0$.

$\Rightarrow$) Let z s.t. $Lz = 0$. Then with $r = Qz$, $\left[\begin{smallmatrix} L \\ Q \end{smallmatrix}\right] z = \left[\begin{smallmatrix} 0 \\ Qz \end{smallmatrix}\right] = \left[\begin{smallmatrix} 0 \\ I \end{smallmatrix}\right] r$, which implies $z \in \mathrm{Im}\left(\left[\begin{smallmatrix} L \\ Q \end{smallmatrix}\right]^{-1} \left[\begin{smallmatrix} 0 \\ I \end{smallmatrix}\right]\right)$. □

We consider now two special classes of linear maps – the parameter varying memoryless matrices and LPV systems – and construct their nullspace by using Lemma 1.

B. Memoryless matrices

Let $M(\rho)$ be a parameter-dependent memoryless matrix such that $M : \Omega \to \mathbb{R}^{n_y \times n_u}$ where $\rho : \mathbb{R}_+ \to \mathbb{R}^{n_\rho}$ denotes the vector of time-varying parameters. Assume $0 \in \Omega$, $n_y < n_u$. Assume also that $M(\rho)$ has full row rank for all $\rho \in \Omega$. (We focus only on this case, because it is enough to construct the nullspace of an LPV system in the next section). If the parameter dependence is linear fractional, then $M(\rho)$ can be given in LFR as follows

$$\begin{aligned} M(\rho) &= M_{21}\Delta(\rho)(I - M_{11}\Delta(\rho))^{-1}M_{12} + M_{22} \\ &= \mathcal{F}_u\left(\left[\begin{smallmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{smallmatrix}\right], \Delta\right) \end{aligned} \tag{1}$$

where M_{ij} are constant matrices, $\Delta : \Omega \to \mathbf{\Delta} \subset \mathbb{R}^{r_\Delta \times c_\Delta}$ is a parameter dependent matrix whose entries are linear in ρ and M_{22} is of full row rank. The representation is *well-defined* if $I - M_{11}\Delta(\rho)$ is invertible for all $\rho \in \Omega$.

According to Definition 1 the kernel of $M(\rho)$ is the set of all input vectors $u \in \mathbb{R}^{n_u}$ satisfying $M(\rho)u = 0$, $\forall \rho \in \Omega$. Based on Lemma 1 the generator of the nullspace can be constructed by finding a (generally) parameter dependent matrix $Q(\rho)$ such that $[M(\rho)^T \ Q(\rho)^T]^T$ is invertible for all $\rho \in \Omega$. Since some additional smoothness properties are also prescribed in general for the parameter dependence of the nullspace generator, thus finding an admissible $Q(\rho)$ is difficult in practice. Therefore, based on the idea in [16], the following algorithm can be applied: choose some parameter value ρ^* and find (a parameter-independent) Q such that $[M(\rho^*)^T \ Q^T]^T$ is invertible. (Without loss of generality, we can choose $\rho^* = 0$, i.e. $M(\rho^*) = M_{22}$ and Q can be chosen to be a basis of the nullspace of M_{22}.) Then use Q over the entire parameter domain, i.e. consider the matrix $[M(\rho)^T \ Q^T]^T$. Based on [16], the *formal* inverse of $[M(\rho)^T \ Q^T]^T$ and thus by Lemma 1 the *formal* generator of $\mathcal{N}(M(\rho))$ can be obtained in LFR form as follows:

$$N(\rho) = \mathcal{F}_u\left(\begin{bmatrix} M_{11} - M_{12}XM_{21} & M_{12}Y \\ -XM_{21} & Y \end{bmatrix}, \Delta\right) \tag{2}$$

where $\left[\begin{smallmatrix} M_{22} \\ Q \end{smallmatrix}\right]^{-1} = [X \ Y]$ such that $M_{22}X = I$ and $M_{22}Y = 0$.

We called the inverse and the generator *formal*, because if $[M(\rho)^T \ Q^T]^T$ is not invertible for all $\rho \in \Omega$, then (2) is not well-defined. Well-definedness and thus the invertibility can be achieved by finding a factorization $N(\rho) = N_r(\rho)M_r(\rho)^{-1}$ such that $N_r(\rho)$ is well-defined. It is clear that $N_r(\rho)$ spans the same nullspace as $N(\rho)$, so $N(\rho)$ can be replaced by $N_r(\rho)$. Unfortunately there is no general algorithm that finds the factorization above in all cases. Instead, there are approaches that generally work in the practical situations.

M1) Before trying any factorisation method, it is advisable to minimize the realisation of $N(\rho)$ first. This means finding an equivalent representation for N with smaller Δ. The observability and reachability decomposition proposed in [17], [18] and implemented in the LFR Toolbox[1] [16] is a possible algorithm to minimize the realisation. Although the algorithm is efficient in several cases it has non negligible limits: it can work only with diagonal Δ and seeks only transformations, which commute with the parameter block.

M2) Find a matrix F such that $N_r(\rho)$ and $M_r(\rho)$ defined by

$$\begin{bmatrix} M_r(\rho) \\ N_r(\rho) \end{bmatrix} = \mathcal{F}_u\left(\left[\begin{array}{c|c} N_{11} + N_{12}F & N_{12} \\ \hline F & I \\ N_{21} + N_{22}F & N_{22} \end{array}\right], \Delta\right) \tag{3}$$

are well-defined. It follows from the construction that $N(\rho) = N_r(\rho)M_r(\rho)^{-1}$, so (3) is a suitable factorization of the kernel. Unfortunately, this approach cannot be applied in all cases: there are ill-defined systems (even in minimal realisation), for which there does not exist a suitable F, but a well-defined left-right factorisation can still be found. See e.g. the example in [19].

M3) It is clear that the entries of $N(\rho)$ are rational functions of δ_i-s. If $p(\rho)$ denotes the product of terms, which are responsible for the poles of $N(\rho)$, a possible factorisation can be obtained by choosing $N_r(\rho) = N(\rho)\frac{p(\rho)}{q(\rho)}$ and $M_r(\rho) = \frac{p(\rho)}{q(\rho)}$, where $q(\rho)$ is an arbitrary polynomial, which does not have zero in Ω. In this way, the ill-

[1] `minlfr` function

definedness can be fully eliminated. The only weak point of this algorithm is the difficulty of finding the poles of $N(\rho)$ in larger dimensional cases.

C. LPV systems

Consider now an LPV system in the usual state-space form:

$$G: \begin{array}{l} \dot{x} = A(\rho)x + B(\rho)u \\ y = C(\rho)x + D(\rho)u \end{array}, \tag{4}$$

where $\rho : \mathbb{R}_+ \rightarrow \Omega \subseteq \mathbb{R}^{n_\rho}$ collects the time-varying scheduling parameters; $u : \mathbb{R}_+ \rightarrow \mathbb{R}^{n_u}$, $y : \mathbb{R}_+ \rightarrow \mathbb{R}^{n_y}$, $x : \mathbb{R}_+ \rightarrow \mathbb{R}^{n_x}$ are the input, output and the state vector, respectively. If $x(0) = 0$, (4) defines a linear map that uniquely assigns to any input signal $u(t)$ an output signal $y(t)$. Moreover, this map, denoted by $G(\rho, \mathcal{I})$ can be given in LFR as follows:

$$G(\rho, \mathcal{I}) = \mathcal{F}_u \left(\left[\begin{smallmatrix} A(\rho) & B(\rho) \\ C(\rho) & D(\rho) \end{smallmatrix} \right], \mathcal{I} \right). \tag{5}$$

$\mathcal{I}$ is used to denote an n_x-dimensional diagonal matrix of integral operators such that $v = \mathcal{I}z$ means that $v_i(t) = \int_0^t z_i(t)dt$, $i = 1 \dots n_x$ with $v(0) = x(0) = 0$. Using Definition 1, we can now define the nullspace of $G(\rho, \mathcal{I})$ as a set of all input signals that are mapped to the constant 0 output. Due to the zero initial condition the defninition of the nullspace makes sense even if G is an unstable system. In the remaining part of the section we give an algorithm to construct the generator of the nullspace.

In order to proceed, two additional assumptions are made on G: (i) the entries of $D(\rho)$ are linear fractional functions of the parameters and (ii) $D(\rho)$ is a full row rank matrix for all $\rho \in \Omega$. Assumption (i) is not restrictive and necessary to use LFR formulation. Assumption (ii) facilitates the discussion and will be relaxed at the end of the section. The two assumptions imply that there exists a well-defined generator $N_D(\rho)$ for the nullspace of $D(\rho)$ such that $D'(\rho) := \left[\begin{smallmatrix} D(\rho) \\ N_D(\rho) \end{smallmatrix} \right] = \mathcal{F}_u \left(\left[\begin{smallmatrix} D'_{11} & D'_{12} \\ D'_{21} & D'_{22} \end{smallmatrix} \right], \Delta' \right)$ is invertible for all $\rho \in \Omega$. The inverse can be determined by applying the following formula [16] with similar considerations as in the previous section:

$$D'(\rho)^{-1} = \mathcal{F}_u \left(\left[\begin{array}{cc} D''_{11} & D''_{12} \\ D''_{21} & D''_{22} \end{array} \right], \Delta' \right)$$

$$\begin{array}{ll} D''_{11} = D'_{11} - D'_{12}(D'_{22})^{-1}D'_{21}, & D''_{12} = D'_{12}(D'_{22})^{-1} \\ D''_{21} = -(D'_{22})^{-1}D'_{21}, & D''_{22} = (D'_{22})^{-1} \end{array}$$

The invertibility of D'_{22} is guaranteed by the fact that $D'(\rho)$ is invertible for all $\rho \in \Omega$ and by assumption, $0 \in \Omega$. Let $D'(\rho)^{-1}$ be partitioned as $[X(\rho)\ Y(\rho)]$ such that $D(\rho)X(\rho) = I$ and $D(\rho)Y(\rho) = 0$. Then, by using (2) the generator of $\mathcal{N}(G(\rho, \mathcal{I}))$ can be computed as

$$N(\rho, \mathcal{I}) = \mathcal{F}_u \left(\left[\begin{array}{cc} A_o(\rho) & B_o(\rho) \\ C_o(\rho) & D_o(\rho) \end{array} \right], \mathcal{I} \right) \tag{6}$$

$$\begin{array}{ll} A_o(\rho) = A(\rho) - B(\rho)X(\rho)C(\rho), & B_o(\rho) = B(\rho)Y(\rho) \\ C_o(\rho) = -X(\rho)C(\rho), & D_o(\rho) = Y(\rho) \end{array}$$

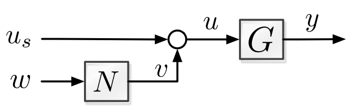

Fig. 1. A typical interconnection of G with its nullspace generator G_o in a reconfiguration architecture

Note that $N(\rho, \mathcal{I})$ is an LFR of integrators, thus it is always well-defined [19]. By construction, the linear mapping $v = N(\rho, \mathcal{I})w$ assigns to the input signal $w(t)$ the output response $v(t)$ of the dynamical (LPV) system

$$N: \begin{array}{l} \dot{x}_o = A_o(\rho)x_o + B_o(\rho)w \\ v = C_o(\rho)x_o + D_o(\rho)w \end{array}, \tag{7}$$

where $x_o(0) = 0$. Due to practical applicability we are interested in stable nullspace generator, therefore if N is not stable, it has to be stabilized. For this, we can use a similar factorization as (3). To proceed, we make assumption (iii): there exists a stabilizing state feedback gain $F(\rho)$ such that $A_o(\rho) + B_o(\rho)F(\rho)$ is asymptotically stable. Then with

$$\left[\begin{array}{c} M_r(\rho, \mathcal{I}) \\ N_r(\rho, \mathcal{I}) \end{array} \right] = \mathcal{F}_u \left(\left[\begin{array}{c|c} A_o(\rho) + B_o(\rho)F(\rho) & B_o(\rho) \\ \hline F(\rho) & I \\ C_o(\rho) + D_o(\rho)F(\rho) & D_o(\rho) \end{array} \right], \mathcal{I} \right)$$

the equality $N(\rho, \mathcal{I}) = N_r(\rho, \mathcal{I})M_r(\rho, \mathcal{I})^{-1}$ holds so $N_r(\rho, \mathcal{I})$ defines a stable generator system.

In the fault tolerant control framework the role of N is to generate an input reconfiguration signal for G that does not influence its output. The general interconnection is depicted in Fig. 1. G is assumed to be connected in a closed loop such that u_s is a stabilizing input generated by some (baseline) controller. Let us examine the system behavior in a practical situation, when the initial state of G differs from zero and is not known either. Assume that N is stable and starts from zero initial state. Since the systems are linear, G is stabilized via $u_s(t)$ and N is stable, the state transients generated due to the nonzero initial condition die out and the output of G converges to the response corresponding to $u_s(t)$. Consequently, $w(t)$ does not influence $y(t)$.

Finally, assume that condition (i) does not hold, i.e. $D(\rho)$ is not a full row rank matrix. We show that in this case the full row rank property can still be achieved by iteratively replacing the outputs causing rank deficiency by their time derivatives until a full row rank D' matrix is obtained. This idea is the same as that is used in [20] for constructing the inverse of a linear system. The main steps of the procedure are summarized in the following algorithm. For simplicity, we present the procedure for LTI systems. The parameter-varying case can be similarly handled.

Algorithm 1. Assume for simplicity that $[C\ D]$ has full row rank, i.e. the trivial output redundancy is excluded. Let $C_1 = C$, $D_1 = D$, $y_1 = y$ and $k = 1$.

1) If $D_k \neq 0$ compute its SVD as $D_k = [U_{k,1} \quad U_{k,2}] \left[\begin{smallmatrix} \Sigma_k & 0 \\ 0 & 0 \end{smallmatrix} \right] \left[\begin{smallmatrix} V^*_{k,1} \\ V^*_{k,2} \end{smallmatrix} \right]$, and perform a linear transformation using the invertible matrix U^*_k: $\tilde{y}_k = U^*_k y_k = \left[\begin{smallmatrix} U^*_{k,1}C_k x + \Sigma_k V^*_{k,1} u \\ U^*_{k,2}C_k x \end{smallmatrix} \right] = \left[\begin{smallmatrix} C_{k,1}x + D_{k,1}u \\ C_{k,2}x \end{smallmatrix} \right] = \left[\begin{smallmatrix} \tilde{y}_{k,1} \\ \tilde{y}_{k,2} \end{smallmatrix} \right]$. Trivially, if $D_k = 0$,

then $\tilde{y}_{k,2} = y_k$ and there is no $\tilde{y}_{k,1}$.

2) Replace $\tilde{y}_{k,2}$ by its time derivative, i.e. let the new output equation be defined as follows: $y_{k+1} = \begin{bmatrix} C_{k,1} \\ C_{k,2}A \end{bmatrix} x + \begin{bmatrix} D_{k,1} \\ C_{k,2}B \end{bmatrix} u = C_{k+1}x + D_{k+1}u$.
3) Set $k := k+1$. If D_k is of full row rank, then terminate and go to step 4. Otherwise, go to step 1.
4) Let the transformed system be defined as follows:

$$G' : \quad \begin{aligned} \dot{x} &= Ax + Bu \\ y &= C_k x + D_k u \end{aligned} \tag{8}$$

If the algorithm terminates in finite steps, then G' can be constructed. Since only derivations and invertible transformations are applied in each step, G' has the same kernel as of G and, by construction, D_k has full row rank. Consequently, (6) can be applied. The main concept of the algorithm can be extended to LPV systems, but in that case the transformed system will depend on the time derivatives of the scheduling parameters as well. As the generator computed from G' inherits this dependence, the time derivatives of the parameters have to be known (measured or precisely computed) to evaluate the generator system.

III. Control input reconfiguration architecture for compensating actuator failures

In this section an input reconfiguration method is proposed to compensate actuator failures. The concept extends the main idea of [8] to parameter varying systems that do not necessarily have static nullspace. The architecture presented in Figure 2 is similar to that proposed in [10], but the concept we follow is basically different.

Before proceeding, we introduce the following notational convention: If S denotes a dynamical system, then its ℓ-th input and output will be denoted by $u_{S,\ell}$ and $y_{S,\ell}$, respectively. Let the n_u input – n_y output LPV plant be denoted by G. Assume the actuators driving the inputs of G are modeled by LTI systems collected in the n_u input – n_u output block A. Let K_B denote the baseline controller designed to guarantee the stability and control performance in fault-free case. For simplicity, consider now the case of one actuator fault affecting the ℓ-th control input such that $u_{G,\ell} = y_{A,\ell} + f = A_\ell(y_{K_B,\ell}) + f$, where f is an external fault signal, A_ℓ is the ℓ-th actuator dynamics. Suppose that $u_{G,\ell}$ (or equivalently f) is available for measurement or can be reconstructed by a suitable FDI filter, e.g. [21]. Let $\bar{A}$ be an n_u input – n_u output LTI shaping filter defined later and let N be the (stable) generator for the nullspace of $G \cdot A \cdot \bar{A}$. The procedure of input reconfiguration is the following. If controller K_N is designed such that $v_{3,\ell}$ (the ℓ-th component of v_3) tracks the fault signal f, then $u_{A,\ell} = y_{K_B,\ell} + v_{2,\ell}$, i.e. $y_{A,\ell} = A_\ell(y_{K_B,\ell}) + A_\ell(v_{2,\ell}) = A_\ell(y_{K_B,\ell}) + f = u_{G,\ell}$, i.e. the ℓ-th output of the actuator block equals to the faulty input to the plant. Therefore it looks as if the faulty input had been generated by the baseline controller. The input reconfiguration is performed via v_2, because it modifies the other control inputs such that they adapt to the faulty input. Since v_2 is generated from the nullspace, thus it does not have any effect on the plant's behavior, i.e. the baseline control loop remains intact and the nominal control performance is preserved.

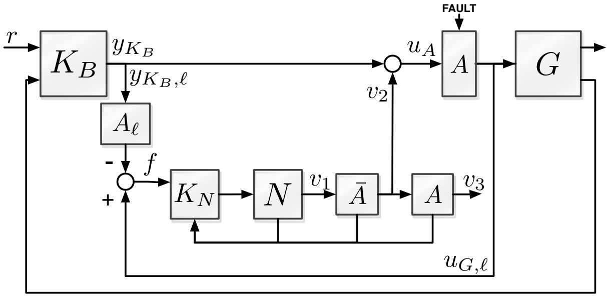

Fig. 2. Input reconfiguration architecture

To complete the discussion, the role of the shaping filter $\bar{A}$ remains to be clarified. This dynamical block can be chosen by the designer to shape the actuator dynamics in order to simplify the nullspace computation. For example, if A is invertible then one can choose $\bar{A} = A^{-1}$ and thus the actuators are eliminated from the reconfiguration procedure. If A is not invertible, it may be possible to find an invertible $\hat{A}$, which approximates A over the frequency range of its bandwidth. Then $\hat{A} = \hat{A}^{-1}$ can be chosen. If the approximation does not work either, $\bar{A}$ can still be chosen to ”advantageously shape” the actuator dynamics. For example, [10] proposes to choose $\bar{A}$ such that $A \cdot \bar{A} = \kappa(s)I$, where $\kappa(s)$ is a user-defined transfer function. This unification simplifies the generator system and can improve the conditioning of the numerical computations related to the construction of N.

Finally, we add some comments and remarks to the architecture above.

1) It is clear that the method works perfectly only if the exact model of the plant is known, since the kernel and the plant should perfectly match in order that the effect of the input reconfiguration on the control performance could be perfectly cancelled. If there is modelling uncertainty in the system, then the robustness of the reconfigured system has to be checked either by simulations or by using a classical analysis tool based on $\mathcal{L}_2$ gain computation and Integral Quadratic Constraints (IQC), see e.g. [22]. It is a possible option as well to improve the robustness properties of the reconfigured system by tuning the tracking controller K_N.

2) The design of the trajectory tracking controller K_N is not difficult in general as the states of every dynamical model in the reconfiguration block (i.e. N, $\bar{A}$ and A) are fully available. Also, when modelling uncertainty is present the tracking controller can also be used to attenuate the performance degradation caused by the mismatch of the (nominal) nullspace and the (uncertain) plant. To handle this additional objective, this requirement has to be transformed to a performance target considered in the design of K_N.

3) The number and type of faults that can be compensated by the proposed method depend on the dimension and dynamical properties of the kernel space. If the dynamical properties are not appropriate (e.g. the kernel dynamics contain too fast or too slow dynamical components) the stabilizing state feedback gain $F(\rho)$ or the tracking controller K_N can be used to shape this dynamics and set the required behaviour.

IV. Reconfigurable fault tolerant control of the B-1 aircraft

The Rockwell B-1 Lancer is a supersonic bomber that was introduced in the 1970s. It has four turbofan engines and variable wing sweep. In order to analyze the aeroelastic issues occuring at subsonic speeds a high fidelity simulator was developed in [23].

A. The nonlinear flight simulator

To test the reconfigurable architecture in a realistic environment, the high-fidelity mathematical model is used. The simulator consists of the following main components: the nonlinear dynamics of the B-1 aircraft, the dynamics of the actuators and the stability augmentation system (SAS). In this case study the flexible components were neglected, only the rigid body dynamics were used. The simplified structure of the simulator is presented in Fig. 3.

The inputs to the nonlinear model are the left (L) and right (R) horizontal tails ($u_{HR},\ u_{HL}$), wing upper-surface spoilers ($u_{SPR},\ u_{SPL}$), splitted upper (RU) and lower (RL) rudders ($u_{RU},\ u_{RL}$) and control vanes ($u_{CVR},\ u_{CVL}$). All are measured in [*deg*]. Since the flexible part was neglected the control vanes appear as free inputs that can be used in actuator reallocation. This configuration corresponds to the subsonic, clean configuration case. The model has 10 states: $x = [\phi, \theta, \psi, p, q, r, u, v, w, h]$, where ϕ, θ, ψ are the roll-, pitch- and yaw angles in [*rad*], p, q, r are the corresponding angular rates in [*rad/s*], u, v, w denote the body axis velocities in [*ft/s*] and h is the height in [*ft*]. The model is scheduled with height and the Mach number M. The inner loop system has 4 outputs, $y = [p,\ q,\ r,\ a_y]^T$, where a_y is the side acceleration measured in [g], which is often referred as side load factor n_y. The structure of the dynamics can be given as follows:

$$\dot{x} = F(\rho, x, u)$$
$$y = \begin{bmatrix} E_{4,5,6}x \\ h(\rho, x) \end{bmatrix} + \begin{bmatrix} 0 \\ \ell(\rho, x) \end{bmatrix} u$$

where matrix $E_{4,5,6}$ selects p, q, r from the state vector and $h(\rho, x), \ell(\rho, x)$ are nonlinear functions defining a_y. The scheduling variable ρ is a two dimensional vector comprising of the height and the Mach number, i.e. $\rho = [h,\ M]$. We assume that the flight envelope is defined such that ρ takes values only from polytope Ω, defined by the following 4 vertices: $[10000,\ 0.55], [10000,\ 0.7],\ [20000,\ 0.65], [20000,\ 0.8]$.

The actuators are modeled by first order linear systems with output saturation. The poles of the LTI models and the saturation limits are collected in Table I. The sensors dynamics are neglected from the present investigation due to their high bandwidth.

The SAS is responsible for the inner-loop rigid body control of the aircraft, providing pitch-, and roll-rate tracking and a yaw damper function using the measurements defined above and the reference stick commands from the pilot. In our reconfigurable control architecture the SAS is considered as the baseline controller K_B.

Actuator	Poles	Lower Lim.	Upper Lim.
Horizontal tails	$-10,\ -57.14$	-20 deg	10 deg
Wing u/s spoilers	$-10,\ -20$	0 deg	45 deg
Lower rudder	$-20,\ -80$	-10 deg	10 deg
Upper rudder	-10	-10 deg	10 deg
Control vanes	-50	-20 deg	20 deg

TABLE I
Properties of the actuators

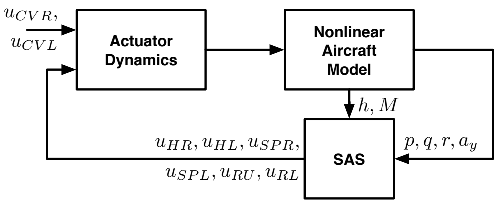

Fig. 3. Basic configuration of the closed loop system

B. Construction of the LPV model

In order to construct the LPV model, the nonlinear dynamics are trimmed and linearized at different points of the flight envelope. By analyzing several configurations, we finally select 12 points equidistantly covering the Ω domain. After removing the unstable spiral mode from the model, least-squares interpolation is used to connect the 12 LTI systems into the following affine LPV model:

$$G(\rho, \mathcal{I}) = \mathcal{F}_u \left(\begin{bmatrix} A(\rho) & B(\rho) \\ C(\rho) & D(\rho) \end{bmatrix}, \mathcal{I} \right), \text{ where} \tag{9}$$
$$A(\rho) = A_0 + A_1 h + A_1 M \qquad B(\rho) = B_0 + B_1 h + B_1 M$$
$$C(\rho) = C_0 + C_1 h + C_1 M \qquad D(\rho) = D_0 + D_1 h + D_1 M$$

The results are verified by computing the gap- and ν-gap metrics [24], [25] between (9) and the linearized models at the 14 grid points. Since none of the gap metrics exceed 0.025 among the points, the LPV model is accepted as a good approximation of the original dynamics.

C. Actuator inversion and nullspace computation

Since the actuator dynamics are stable, they can be augmented with suitably fast left half plane zeros to make them biproper, and hence invertible. It is clear that the faster the zeros, the better the approximation is. On the other hand, arbitrarily fast zeros cannot be chosen because that would lead to high transients in the inverse dynamics. Therefore, as a compromise, the zeros are chosen to be four times faster than the poles in each actuator. Since the zeros are much faster than the bandwidth of the controller and the plant, the use of approximate inverses do not significantly influence the performance of the reconfigured control loop.

The next step is the computation of the nullspace of $G(\rho, \mathcal{I})$. For this, the first three outputs p, q, r have to be derived because $D(\rho)$ is not a full row rank matrix. After derivation, the rank condition is satisfied so no more derivations are necessary to determine the kernel. Since the output equations defining p, q and r do not depend on ρ, the new outputs $\dot{p}$, $\dot{q}$ and $\dot{r}$ and consequently the kernel do not depend on $\dot{\rho}$. The kernel $N(\rho, \mathcal{I})$ finally obtained is unstable, so it has to be stabilized. For this, following the formal coprime factorization

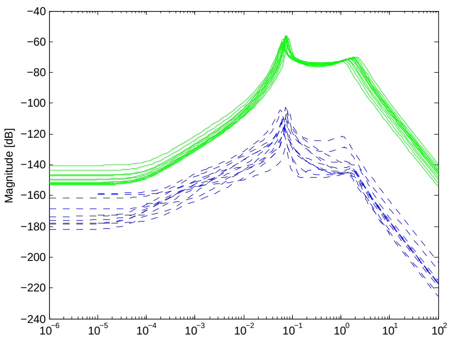

Fig. 4. Bode plot of the systems $G(\rho,\mathcal{I})N(\rho,\mathcal{I})$ (blue) and $G(\rho,\mathcal{I})N(\rho)$ (green) from the first input to the output q. Different lines correspond to different flight speed and height

method, a stabilizing state feedback $F(\rho)$ is designed by solving the feasibility problem as follows:

$$\begin{aligned} A(\rho)Q + QA(\rho)^T +& \\ +B(\rho)Y + Y^T B^T(\rho) + 2cQ &\prec 0 \\ \begin{bmatrix} u_{lim}^2 I & Y \\ Y^T & Q \end{bmatrix} &\succ 0 \end{aligned} \tag{10}$$

where $Q = P^{-1}$, $V(x) = x^T P x$ is the Lyapunov function and constants c, u_{lim} are used to control the location of the poles (at fixed ρ values) and the norm of the feedback gain matrix, respectively [26]. The free variables in (10) are Q and Y from which the feedback gain is computed as $F := YQ^{-1}$, which is now parameter-independent. (Parameter-dependent feedback gain can also be constructed by choosing Q and Y parameter-dependent.) To solve (10) the matrix inequalities are reduced to a finite set of LMI-s by evaluating them over a suitable dense grid over Ω [27].

We found in this particular example that the parameters c and u_{lim} can be chosen such that the kernel dynamics become significantly slower than the closed loop formed by the plant and the baseline controller. This makes it possible to truncate the states and substitute the dynamical kernel by a static, parameter dependent feedthrough gain, i.e. $N(\rho) := D_o(\rho)$. In order to analyze how $N(\rho)$ "approximates" the true generator system the bode plots of $G(\rho,\mathcal{I})N(\rho,\mathcal{I})$ and $G(\rho,\mathcal{I})N(\rho)$ are computed for different values of ρ. The worst results are obtained from the first input of the kernel to the output q. This input-output combination is shown in Fig. 4. It can be seen that $G(\rho,\mathcal{I})N(\rho)$ is very close to zero in this case as well, so $N(\rho)$ is a suitable approximation of the nullspace. It is important to emphasise that $N(\rho)$ can not be determined without the dynamical kernel, because the augmented input matrix $[B(\rho)^T \;\; D(\rho)^T]^T$ has full column rank for all $\rho \in \Omega$ so it does not have a nullspace.

D. Fault signal tracking

The next component of the reconfigurable control architecture is the construction of the K_N controller that is responsible for tracking the fault signal $f = u_\ell - A_\ell(s)y_{K_B,\ell}$, where u_ℓ is the output of the ℓ-th (faulty) actuator and $y_{K_B,\ell}$ is the ℓ-th output of the baseline controller. Since the kernel is static, this trajectory tracking problem is equivalent to finding for each time t a suitable $u_N(t)$ such that the ℓ-th entry of $N(\rho(t))u_N(t)$ be equal to $f(t)$. As we have strict and asymmetric constraints for the actuator outputs (Table I), this problem is solved by parameter-dependent quadratic programming (QP):

$$\begin{aligned} &\min_{u_N(t)} (r(t) - N(\rho(t))u_N(t))^T W (r(t) - N(\rho(t))u_N(t)) \\ &\text{w.r.t.} \quad N(\rho(t))u_N(t) \geq L - y_{K_B} \\ &\phantom{\text{w.r.t.} \quad} N(\rho(t))u_N(t) \leq U - y_{K_B} \end{aligned} \tag{11}$$

where the vector r of reference signals is defined as $r_i = 0$ if $i \neq \ell$ and $r_\ell = f$. Vectors L and U collect the lower and upper limits of the actuators and the diagonal matrix W is used to weight the tracking errors [8], [13]. By defining r in the way above we prescribe 0 reference for the non faulty inputs. This expresses our intention to keep the healthy inputs close to the value determined by the baseline controller. By varying the entries of W the contribution of each control input to the reconfiguration signal y_N can be tuned. (In the numerical simulations below $W = \mathrm{diag}(w_1, \ldots, w_{n_u})$ is used with $w_\ell = 1000$ and $w_i = 1$ for $i \neq \ell$, i.e. precise tracking is demanded for the fault signal but no preferences are prescribed for the use of other control inputs.

E. Simulation results

To demonstrate the applicability of the proposed reconfigurable control architecture the following fault scenario is presented: the pilot performs a pitch-rate doublet maneuver, by giving a q_{ref} command as shown in Fig. 6. The maneuver starts at $t = 0sec$. It is assumed that at $t = 1sec$ the right horizontal tail is jammed at its actual position. By assuming an ideal fault detection and isolation algorithm, the reconfiguration starts also at $t = 1sec$.

Simulation results are shown for the healthy (green), faulty (blue), and for the reconfigured (red) cases in Figs. 5-7.

The roll rate response of the three cases are shown in Figure 5, where the ideal case of healthy behavior is constantly zero again, in case of a right horizontal tail fault roll rate excursions of $4\ deg/s$ occur due to the asymmetrical deflections of the control surfaces. The proposed control allocation scheme is able to reduce it by at least a factor of four and the steady state error is zero, showing the clear advantage of the proposed method. The nonzero transient error is due to the nonlinear aircraft model. By testing the method on the LPV model only a negligibly small (less than 10^{-5}) error can be experienced due to the numerical inaccuracy of the simulation and the use of approximate actuator inverse.

The main objective of pitch rate tracking in healthy, faulty and reconfigured cases are simulated in Figure 6. The healthy and reconfigured responses are very close to each other, while the uncompensated case have more than 50% error and nonzero steady state error. This shows that, although stability is not lost, the performance of the uncompensated case in not acceptable.

The yaw rate response of the three cases are shown in Fig. 7, where it is clear that the healthy behavior does not

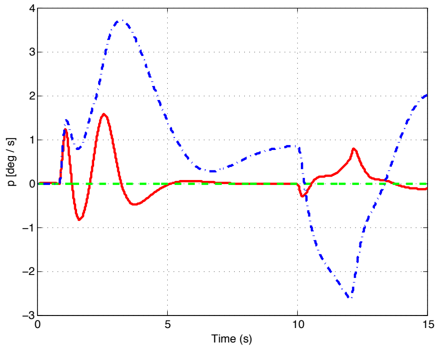

Fig. 5. Simulation result of the angular rate p in three different cases: without fault (green dashed), with fault and reconfiguration (red solid), and with fault (blue dot-dashed)

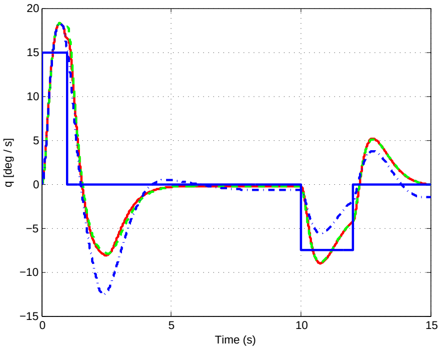

Fig. 6. Simulation result of the angular rate q in three different cases: without fault (green dashed), with fault and reconfiguration (red solid), and with fault (blue dot-dashed). The reference signal q_{ref} is plotted by solid blue.

leave the longitudinal plane, while the uncompensated case have small but non negligible yaw rate response. When the reconfiguration scheme is applied a similar fourfold transient response decrease can be seen, while the steady state error is improved by an order of magnitude. The nonzero error is again due to the nonlinear characteristics of the simulation, since in the LPV simulations the difference between healthy and compensated behaviors are not visible.

Fig. 8 - 10 show the inputs u_{HR}, u_{HL}, u_{SPR} and u_{SPL} with and without reconfiguration. The contribution of the other inputs are small in all the simulation scenarios. It can be seen that the compensation of the faulty right horizontal tail required the active use of the left horizontal tail (u_{HL}) and the two spoilers (u_{SPR}, u_{SPL}). The latter two are not used in normal operations.

F. Robustness analysis

As our reconfigurable control is based on perfect cancellation of the plant dynamics, the performance of the baseline controller can be perfectly preserved only if the exact plant model is known. Since in a real environment this condition is never met the performance of the reconfigured system always has to be checked for a set of models. Therefore, simulation are performed with different values of inertia and mass parameters of the aircraft. The nullspace based allocation and the tracking controller are held the same as in the nominal case. The actual values of the parameters are selected randomly according to uniform distribution in a $\pm 20\%$ interval around their nominal values of: $I_{xx} = 950000$, $I_{xz} = -52700$, $I_{yy} = 6400000$, $I_{zz} = 7100000$, $mass = 8944$, where the inertia is measured in $[sl.ft^2]$ and the mass is given in $[sl]$. This way 500 different scenarios are generated and tested. The results are summarized in Fig. 11, where the angular rates are plotted for the 7 most representative parameter values. It can be seen that the results are very similar in all cases, leading to a conclusion that

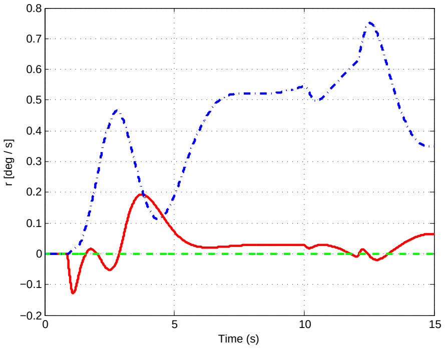

Fig. 7. Simulation result of the angular rate r in three different cases: without fault (green dashed), with fault and reconfiguration (red solid), and with fault (blue dot-dashed)

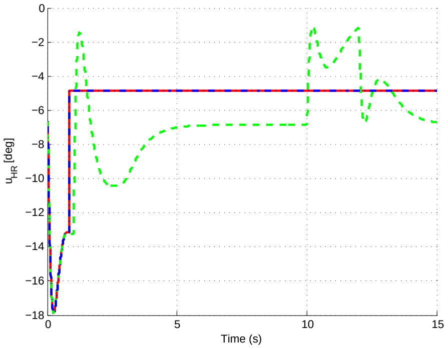

Fig. 8. Simulation result of the input u_{HR} in three different cases: without fault (green dashed), with fault and reconfiguration (red solid), and with fault (blue dot-dashed). The fault occurs at 1 sec

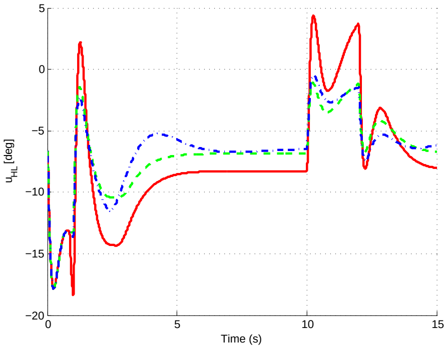

Fig. 9. Simulation result of the input u_{HL} in three different cases: without fault (green dashed), with fault and reconfiguration (red solid), and with fault (blue dot-dashed). The fault occurs at 1 sec

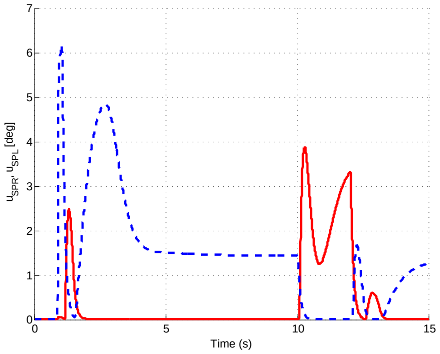

Fig. 10. Simulation result of the two spoilers u_{SPR} and u_{SPL} in the case of reconfiguration. These inputs are not used by the SAS. The fault occurs at 1 sec

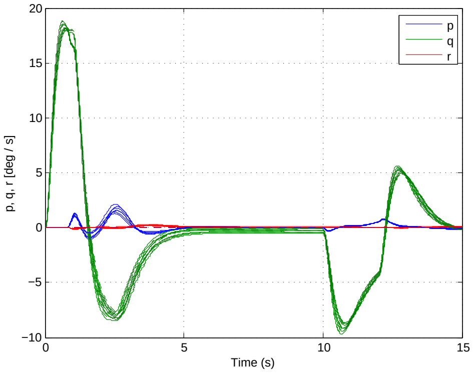

Fig. 11. Robustness analysis of the angular rates p (blue), q (green), r (red)

the fault-tolerant controller is robust to the mass and inertia parameters.

V. Conclusion

A dynamic kernel based input reallocation framework has been proposed to accommodate actuator faults in LPV systems. The method improves and extends the concept presented in [8]. Though the paper provides a complete design method, there are several points where the proposed approach can be further improved. First of all, the properties of the nullspace and the generator system have to be further analyzed in order to provide clear conditions for the stabilizability and well-definedness of the generator system and to find a systematic method for the right (or left) factorization. Second, the reconfiguration architecture provides large freedom for the designer: the generator dynamics can be shaped by state feedback and the design method used to construct the tracking controller can be freely chosen. This freedom can be exploited to achieve further performance objectives, e.g. to improve the robustness properties of the closed-loop against modeling uncertainties. These points assign research directions that should be focused on in the future.

Acknowledgment

The authors would like to thank Peter Seiler (University of Minnesota) and Arnar Hjartarson (formerly MUSYN Inc.) for providing insight related to the B-1 simulator model, originally developed by David Schmidt.

References

[1] P. Goupil and A. Marcos, “Industrial benchmarking and evaluation of addsafe fdd designs,” in *Fault Detection, Supervision and Safety of Technical Processes (8th SAFEPROCESS)*, 2012.

[2] T. J. J. Lombaerts, Q. P. Chu, J. A. Mulder, and D. A. Joosten, “Modular flight control reconfiguration design and simulation,” *Control Engineering Practice*, vol. 19, no. 6, pp. 540–554, 2011.

[3] I. Hwang, S. Kim, Y. Kim, and C. Seah, “A survey of fault detection, isolation, and reconfiguration methods,” *Control Systems Technology, IEEE Transactions on*, vol. 18, no. 3, pp. 636–653, 2010.

[4] P. Goupil, J. Boada-Bauxell, A. Marcos, E. Cortet, M. Kerr, and H. Costa, “Airbus efforts towards advanced real-time fault diagnosis and fault tolerant control,” in *19th IFAC World Congress, Cape Town, South Africa*, 2014, pp. 3471–3476.

[5] S. Ganguli, A. Marcos, and G. Balas, “Reconfigurable LPV control design for Boeing 747-100/200 longitudinal axis,” in *American Control Conference*, vol. 5, 2002, pp. 3612–3617.

[6] H. Alwi, C. Edwards, and C. Tan, *Fault Detection and Fault-Tolerant Control Using Sliding Modes*. Springer-Verlag, 2011.

[7] T. A. Johansen and T. I. Fossen, “Control Allocation - A survey,” *Automatica*, vol. 49, no. 5, pp. 1087–1103, 2013.

[8] L. Zaccarian, “Dynamic allocation for input redundant control systems,” *Automatica*, vol. 45, pp. 1431–1438, 2009.

[9] D. Enns, “Control allocation approaches,” in *AIAA GNC*, 1998.

[10] A. Cristofaro and S. Galeani, “Output invisible control allocation with steady-state input optimization for weakly redundant plants,” in *Conference on Decision and Control (CDC)*, 2014, pp. 4246–4253.

[11] S. Galeani, A. Serrani, G. Varanoa, and L. Zaccarian, “On input allocation-based regulation for linear over-actuated systems,” *Automatica*, vol. 52, pp. 346–354, 2015.

[12] F. Wu and K. Dong, “Gain-scheduling control of LFT systems using parameter-dependent lyapunov functions,” *Automatica*, vol. 42, pp. 39–50, 2006.

[13] B. Vanek, T. Peni, Z. Szabó, and J. Bokor, “Fault tolerant LPV control of the GTM UAV with dynamic control allocation,” in *AIAA GNC*, 2014.

[14] A. Varga, “The nullspace method-a unifying paradigm to fault detection,” in *IEEE Conference on Decision and Control*, 2009, pp. 6964–6969.

[15] ——, “On computing nullspace bases - a fault detection perspective,” in *17th IFAC World Congress*, 2008, pp. 6295–6300.

[16] J.-F. Magni. (2006) Linear Fractional Representation Toolbox for use with Matlab. [Online]. Available: http://w3.onera.fr/smac/?q=lfrt

[17] R. D’Andrea and S. Khatri, “Kalman decomposition of linear fractional transformation representations and minimality,” in *American Control Conference (ACC)*, 1997, pp. 3557–3561.

[18] C. Beck and R. D’Andrea, “Noncommuting multidimensional realization theory: Minimality, reachability and observability,” *IEEE Transactions on Automatic Control*, vol. 49, no. 10, pp. 1815–1820, 2004.

[19] Z. Szabó, T. Péni, and J. Bokor, “Null-space computation for qLPV Systems,” in *1st IFAC Workshop on Linear Parameter Varying Systems*, 2015.

[20] L. M. Silverman, “Inversion of multivariable linear systems,” *IEEE Transactions on Automatic Control*, vol. AC-14, no. 3, pp. 270–276, 1969.

[21] H. Alwi, C. Edwards, and A. Marcos, “Actuator and sensor fault reconstruction using an LPV sliding mode observer,” in *AIAA GNC*, 2010.

[22] H. Pfifer and P. Seiler, “Robustness analysis of linear parameter varying systems using integral quadratic constraints,” *International Journal of Robust and Nonlinear Control*, vol. 25, pp. 2843–2864, 2015.

[23] D. Schmidt, “A nonlinear simulink simulation of a large flexible aircraft,” MUSYN Inc. and NASA Dryden Flight Research Center, Tech. Rep., 2013.

[24] T. T. Georgiou, “On the computation of the gap metric,” *Systems Control Letters*, vol. 11, pp. 253–257, 1988.

[25] G. Vinnicombe, “Measuring robustness of feedback systems,” Ph.D. dissertation, Department of Engineering, University of Cambridge, 1993.

[26] S. Boyd, L. E. Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*, ser. Studies in Applied Mathematics. SIAM, 1994, vol. 15.

[27] F. Wu, “Control of linear parameter varying systems,” Ph.D. dissertation, University of California at Berkeley, 1995.