

On computing achievable fault signatures

Andras Varga

German Aerospace Center, DLR-Oberpfaffenhofen
 Institute of Robotics and Mechatronics
 D-82234 Wessling, Germany
 Email: andras.varga@dlr.de

Abstract: The knowledge of achievable fault signatures is a valuable information in designing residual generators providing structured residual sets for fault detection and isolation. We propose an efficient computational approach to determine the achievable fault signatures for a given additive fault model. The proposed procedure relies on recently developed numerically reliable nullspace updating techniques involving orthogonal reductions to Kronecker-like forms. The new procedure is general, being applicable to both proper as well as non-proper systems, and is significantly more efficient than an exhaustive search based approach.

Keywords: Fault detection, linear synthesis, structured residuals, computational methods.

1. INTRODUCTION

Consider additive fault models described by input-output representations of the form

$$\mathbf{y}(\lambda) = G_u(\lambda)\mathbf{u}(\lambda) + G_d(\lambda)\mathbf{d}(\lambda) + G_f(\lambda)\mathbf{f}(\lambda), \quad (1)$$

where $\mathbf{y}(\lambda)$, $\mathbf{u}(\lambda)$, $\mathbf{d}(\lambda)$, and $\mathbf{f}(\lambda)$ are Laplace- or Z-transformed vectors of the the p -dimensional system output vector $y(t)$, m_u -dimensional control input vector $u(t)$, m_d -dimensional disturbance vector $d(t)$, and m_f -dimensional fault vector $f(t)$, respectively, and where $G_u(\lambda)$, $G_d(\lambda)$ and $G_f(\lambda)$ are the *transfer-function matrices* (TFMs) from the control inputs to outputs, disturbance inputs to outputs, and fault inputs to outputs, respectively. According to the system type, the frequency variable λ is either s , the complex variable in the Laplace-transform in the case of a continuous-time system or z , the complex variable in the Z-transform in the case of a discrete-time system. For most of practical applications, the TFMs $G_u(\lambda)$, $G_d(\lambda)$ and $G_f(\lambda)$ are proper rational matrices. However, for complete generality of our problem settings, we will allow that these TFMs are general non-proper rational matrices for which we will not *a priori* assume any further properties (e.g., stability, full rank).

A linear residual generator (or fault detection filter) processes the measurable system outputs $y(t)$ and control inputs $u(t)$ and generates the residual signals $r(t)$ which serve for decision making on the presence or absence of faults. The input-output form of this filter is

$$\mathbf{r}(\lambda) = Q(\lambda) \begin{bmatrix} \mathbf{y}(\lambda) \\ \mathbf{u}(\lambda) \end{bmatrix} \quad (2)$$

where $Q(\lambda)$ is the TFM of the filter. For a physically realizable filter, $Q(\lambda)$ must be *proper* (i.e., only with finite poles) and *stable* (i.e., only with poles having negative real parts for a continuous-time system or magnitudes less than one for a discrete-time system). The (dynamic) *order* of $Q(\lambda)$ (also known as *McMillan degree*) is the dimension

* This work was supported via the ONERA-DLR Project IMMUNE: Intelligent Monitoring and Management of Unexpected Events.

of the state vector of a minimal state-space realization of $Q(\lambda)$. The dimension q of the residual vector $r(t)$ depends on the fault detection problem to be solved, and can be either given or determined during the solution process.

The residual signal $r(t)$ in (2) generally depends via the system outputs $y(t)$ of all system inputs $u(t)$, $d(t)$ and $f(t)$. The residual generation system, obtained by replacing in (2) $\mathbf{y}(\lambda)$ by its expression from (1), is given by

$$r(\lambda) = R_f(\lambda)\mathbf{f}(\lambda) + R_d(\lambda)\mathbf{d}(\lambda) + R_u(\lambda)\mathbf{u}(\lambda) \quad (3)$$

where

$$[R_f(\lambda)|R_d(\lambda)|R_u(\lambda)] := Q(\lambda) \begin{bmatrix} G_f(\lambda) & G_d(\lambda) & G_u(\lambda) \\ 0 & 0 & I_{m_u} \end{bmatrix}$$

For a successfully designed filter $Q(\lambda)$, the corresponding residual generation system is proper and stable and achieves specific fault detection requirements (e.g., decoupling of control and disturbance inputs from the residuals).

The *fault detection problem* (FDP) can be formulated as follows: Determine a physically realizable linear residual generator filter having the general form (2) such that for all $d(t)$ and $u(t)$ we have:

- (i) $r(t) = 0$ when $f(t) = 0$; and
- (ii) $r(t) \neq 0$ when any $f_j(t) \neq 0$, for $j = 1, \dots, m_f$.

The more advanced functionality of fault isolation (i.e., exact location of faults) can be often achieved by designing a bank of scalar output fault detectors [Gertler, 1998], where each detector solves a *structured* FDP. To solve such a problem, the designed detectors must be sensitive to some faults and insensitive to others. The basic approach to solve the *structured* FDP is to reformulate it as a *standard* FDP, where formally the faults to be rejected in the residual are redefined as fictive disturbances to be decoupled.

For a given detector with a $q \times (p + m_u)$ TFM $Q(\lambda)$, let $R_f(\lambda)$ be the corresponding fault-to-residual TFM in (3). We denote $R_{f_j}^i(\lambda)$ the (i, j) entry of $R_f(\lambda)$. Let Ω_s be a set of frequency values, which characterize one or

several persistent fault modes (e.g., constant faults). We can define a $q \times m_f$ structure matrix S corresponding to a set of q residuals as

$$\begin{aligned} S_{ij} &= 1 \text{ if } R_{f_j}^i(\lambda_s) \neq 0 \forall \lambda_s \in \Omega_s \\ S_{ij} &= -1 \text{ if } \exists \lambda_s \in \Omega_s \text{ such that } R_{f_j}^i(\lambda_s) = 0 \\ &\quad \text{and } R_{f_j}^i(\lambda) \neq 0 \\ S_{ij} &= 0 \text{ if } R_{f_j}^i(\lambda) = 0 \end{aligned}$$

For example, for constant faults we need to check $R_f(\lambda)$ for nonzero DC-gains, thus one chooses $\Omega_s = \{0\}$ for a continuous-time system and $\Omega_s = \{1\}$ for a discrete-time system.

If $S_{ij} = 1$ then we say that the fault j is *strongly* detected in residual i . If $S_{ij} = -1$ then the fault j is only *weakly* detected in residual i . The fault j is not detected in residual i if $S_{ij} = 0$. We refer to the i -th row of S as the i -th *specification*, while the j -th column of S as the *signature* (or *code*) of fault f_j . This and related nomenclature used later is borrowed from [Gertler, 1998].

The following *fault detection and isolation problem* (FDIP) can be now formulated: Given a $q \times m_f$ structure matrix S determine a bank of q stable and proper scalar output residual generator filters

$$\mathbf{r}_i(\lambda) = Q^i(\lambda) \begin{bmatrix} \mathbf{y}(\lambda) \\ \mathbf{u}(\lambda) \end{bmatrix}, \quad i = 1, \dots, q \quad (4)$$

such that, for all $u(t)$ and $d(t)$, and for $i = 1, \dots, q$ we have:

- (i) $r_i(t) = 0$ when $f_j(t) = 0, \forall j$ with $S_{ij} \neq 0$;
- (ii) $r_i(t) \neq 0$ when $f_j(t) \neq 0, \forall j$ with $S_{ij} \neq 0$;
- (iii) $r_i(t)$ is asymptotically bounded.

Each scalar output detector $Q^i(\lambda)$ achieves a signature structure representing the i -th specification in S . For example, to achieve the complete isolation and estimation of maximum q simultaneous faults the choice $S = I_q$ is necessary. In many practical applications this *strong isolation* requirement can not be achieved due to the lack of sufficient number of measurements. If we can enforce a structure matrix with distinct fault signatures, then a so-called *weak isolation* of faults is possible. For example, if for 3 fault inputs the structure matrix

$$S = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

can be achieved, then the occurrence of a single fault f_j can be detected if all residuals (excepting the j -th residual) are non-zero. More insight on how to specify fault signatures can be found in [Gertler, 1998, 2000].

2. COMPUTATION OF ACHIEVABLE SPECIFICATIONS

For the solution of FDIPs, the knowledge of achievable fault signatures is very helpful in formulating FDIPs which have guaranteed solutions. Surprisingly, the generation of the achievable complete structure matrix for a given system (1) has apparently not been addressed in the literature from a procedural point of view. Traditionally this aspect is addressed by a brute force approach, by generating a complete set of $2^{m_f} - 1$ possible specifications and determining which specifications are feasible by trying

to design an appropriate detector. A checked specification is achievable if the corresponding design was successful. In this way, all feasible specifications can be determined.

The design of a detector for a given specification can be reformulated as a nullspace basis computation problem [Frisk and Nyberg, 2001]. The typical approach is to redefine all faults with zero signatures in the checked specification as disturbances. In this way, the design problem becomes to determine the solution of a standard FDP and all potential solutions can be constructed as linear combinations of the nullspace basis vectors of the TFM of a certain extended system. Therefore, a detector formed from all nullspace basis vectors will provide the information if the given specification is feasible or not.

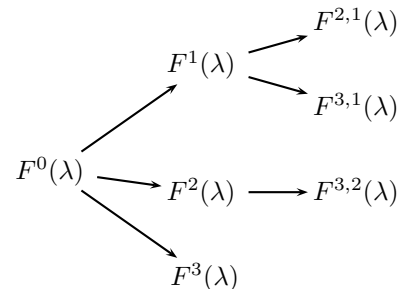
Based on these ideas, we devise a recursive procedure to generate in a systematic and computationally efficient way suitable nullspace bases to serve for the determination of all achievable specifications. We illustrate the core computation with two generic $p_e \times m$ and $p_e \times m_f$ TFMs $G(\lambda)$ and $F(\lambda)$, respectively. The basic computational step consists of successively determining left annihilators $N_i(\lambda)$ of $G(\lambda)$ (i.e., $N_i(\lambda)G(\lambda) = 0$) such that the signature of $N_i(\lambda)F(\lambda)$ has up to $\min(m_f, p_e - r) - 1$ zero columns, where $r = \text{rank } G(\lambda)$. For example, to initialize the procedure for the system (1), we define these TFMs as

$$G(\lambda) = \begin{bmatrix} G_u(\lambda) & G_d(\lambda) \\ I_{m_u} & 0 \end{bmatrix}, \quad F(\lambda) = \begin{bmatrix} G_f(\lambda) \\ 0 \end{bmatrix} \quad (5)$$

with $p_e = p + m_u$ and $m = m_u + m_d$.

To describe the nullspace generation process more in details, let $N_l^0(\lambda)$ be the $(p_e - r) \times p_e$ proper minimal left nullspace basis of $G(\lambda)$ and let S^0 be the structure matrix of $F^0(\lambda) := N_l^0(\lambda)F(\lambda)$, constructed according to its definition in the previous section. If $\min(m_f, p_e - r) > 1$, then for each $i = 1, \dots, m_f$, determine the left nullspace $N_l^i(\lambda)$ of the i -th column of $F^0(\lambda)$ and let S^i be the structure matrix corresponding to $F^i(\lambda) := N_l^i(\lambda)F^0(\lambda)$. Each S^i will have the i -th column zero. If the i -th column is zeroed with $N_l^i(\lambda)$, then $N_l^i(\lambda)$ is a $(p_e - r - 1) \times (p_e - r)$ TFM. If now $p_e - r - 1 > 1$, we continue by computing for each j -th column of $F^i(\lambda)$, $j > i$, the corresponding left nullspace $N_l^{j,i}(\lambda)$ and the corresponding structure matrix $S^{j,i}$ of $F^{j,i}(\lambda) := N_l^{j,i}(\lambda)F^i(\lambda)$. Each $S^{j,i}$ will have the i -th and j -th columns zero. This process continues in a similar way until all $S^{k,\dots,j,i}$ with a single row have been generated. The resulting S is formed by concatenating row-wise the determined $S^0, S^1, \dots, S^{m_f}, S^{2,1}, \dots, S^{m_f,1}, \dots, S^{m_f,m_f-1}, \dots$

The following tree illustrates the performed computations for a system with $m_f = 3$ and $p_e - r = 3$.



In this case, the resulting S is formed as

$$S = \begin{bmatrix} S^0 \\ S^1 \\ S^{2,1} \\ S^{3,1} \\ S^2 \\ S^{3,2} \\ S^3 \end{bmatrix}$$

where each S^i has the i -th column zero, while each $S^{j,i}$ has the i -th and j -th columns zero. It can be observed that the computation of $F^{1,2}(\lambda)$ is not necessary because the same information is provided by $F^{2,1}(\lambda)$. Similarly, the computation of both $F^{1,3}(\lambda)$ and $F^{2,3}(\lambda)$ is not necessary, because the corresponding information is provided by $F^{3,1}(\lambda)$ and $F^{3,2}(\lambda)$, respectively.

In view of software implementation, the proposed computational procedure is formulated as a recursive procedure, which for the given matrices $G(\lambda)$ and $F(\lambda)$ computes the complete structure matrix S . Formally, this procedure can be executed as $S = \text{FDISPEC}(G, F)$. For example, the fault signature matrix for the system (1) can be computed with $G(\lambda)$ and $F(\lambda)$ defined as in (5).

FDISPEC Procedure:

Achievable Fault Signature Computation

Inputs: $p_e \times m$ $G(\lambda)$, $p_e \times m_f$ $F(\lambda)$; *Output:* $q \times m_f$ S .

Procedure $S = \text{FDISPEC}(G, F)$

- 1) Compute a left nullspace basis $N_l(\lambda)$ of $G(\lambda)$;
exit with empty S if $N_l(\lambda)$ is empty.
 - 2) Compute $N_f(\lambda) = N_l(\lambda)F(\lambda)$.
 - 3) Compute the signature matrix S of $N_f(\lambda)$;
exit if S is a row vector.
 - 4) **For** $i = 1, \dots, m_f$
 - 4.1) Form $\tilde{G}_i(\lambda)$ as column i of $N_f(\lambda)$.
 - 4.2) Form $\tilde{F}_i(\lambda)$ from the columns $1, \dots, i-1, i+1, \dots, m_f$ of $N_f(\lambda)$.
 - 4.3) **Call** $\tilde{S} = \text{FDISPEC}(\tilde{G}_i, \tilde{F}_i)$.
 - 4.4) Partition $\tilde{S} = [\tilde{S}_1 \tilde{S}_2]$ such that \tilde{S}_1 has $i-1$ columns.
 - 4.5) Define $\hat{S} = [\tilde{S}_1 \ 0 \ \tilde{S}_2]$ and update $S \leftarrow \begin{bmatrix} S \\ \hat{S} \end{bmatrix}$.
-

The above procedure can be easily implemented such that it performs the minimum number of nullspace computations and updating. This number is given by the combinatorial formula

$$k_S = \sum_{i=0}^{i_{max}} \binom{m_f}{i}, \quad (6)$$

where $i_{max} = \min(m_f, p_e - r) - 1$ and r is the rank of the initial $G(\lambda)$. As it can be observed, k_S depends of the number of initial basis vectors $p_e - r$ and the number of faults m_f , and, although the number of distinct specifications can be relatively low, still k_S can be a large number. For the example considered above, $m_f = 3$ and $p_e - r = 3$, thus $k_S = 7$ nullspace computations are necessary.

The **FDISPEC Procedure** can be interpreted as an exhaustive search for a canonical set of achievable spec-

ifications, having up to $p_e - r - 1$ nonzero columns. The rest of columns of each specification may also contain additional zero entries, depending on the existing concrete algebraic dependencies in the underlying system. Having this in mind, it is clear that the final S obtained by the above procedure may not explicitly contain all possible achievable row signatures, but only implicitly, via suitable linear combinations of the rows of S . Since each row of S can be generated by a scalar output detector, the linear combination of the outputs of two or more detectors produce a signature which can be interpreted as a linear combination of the corresponding rows of S according to min-max algebra rules. Using such linear combinations, the full signature list can be explicitly obtained by generating all possible such linear combinations.

The main computational ingredients necessary to implement the **FDISPEC Procedure** are the determination of a left nullspace basis $N_l(\lambda)$ of $G(\lambda)$ at Step 1 and the computation of $N_f(\lambda) = N_l(\lambda)F(\lambda)$ at Step 2. Therefore, the efficient implementation of this procedure heavily benefits of the state space updating techniques recently developed in [Varga, 2008a]. In the next section, we present the application of these techniques to our problem and show that a substantial gain in efficiency can be achieved when compared with an exhaustive search based brute force procedure.

3. COMPUTATIONAL ISSUES

In this section we address the computational issues underlying the proposed **FDISPEC Procedure**. For computations we use an equivalent *descriptor* state space realization of the compound TFM $[G(\lambda) \ F(\lambda)]$ as

$$[G(\lambda) \ F(\lambda)] := \begin{bmatrix} A - \lambda E & B_G & B_F \\ C & D_G & D_F \end{bmatrix} \quad (7)$$

with state dimension n (i.e., the order of square matrices A and E). According to the notation (7), $G(\lambda)$ and $F(\lambda)$ can be expressed as

$$\begin{aligned} G(\lambda) &= C(\lambda E - A)^{-1}B_G + D_G \\ F(\lambda) &= C(\lambda E - A)^{-1}B_F + D_F \end{aligned} \quad (8)$$

In general, E can be singular in the case of an improper system, but we will still assume that the linear pencil $A - \lambda E$ is regular to guarantee the existence of the above TFMs. For systems with proper TFMs in (8), we can always choose a *standard* state space realizations where $E = I$. In general, we can assume that the representation (7), or equivalently the pair $(A - \lambda E, C)$ is *observable*.

The computational method described in [Varga, 2003] to determine a proper rational left nullspace basis $N_l(\lambda)$ of $G(\lambda)$ exploits the simple fact that $N_l(\lambda)$ is a left nullspace basis of $G(\lambda)$ if and only if for a suitable $M_l(\lambda)$

$$Y_l(\lambda) := [M_l(\lambda) \ N_l(\lambda)] \quad (9)$$

is a left nullspace basis of the system matrix

$$\mathcal{S}(\lambda) = \begin{bmatrix} A - \lambda E & B_G \\ C & D_G \end{bmatrix}. \quad (10)$$

To compute $N_l(\lambda)$, we determine first a left nullspace basis $Y_l(\lambda)$ of $\mathcal{S}(\lambda)$ and then obtain $N_l(\lambda)$ as

$$N_l(\lambda) = Y_l(\lambda) \begin{bmatrix} 0 \\ I_{p_e} \end{bmatrix}.$$

$Y_l(\lambda)$ and thus also $N_l(\lambda)$ can be computed by employing linear pencil reduction algorithms based on orthogonal transformations. Let Q and Z be orthogonal matrices (for instance, determined by using the algorithms of Beelen [1987] or Varga [1996]) such that the transformed pencil $\tilde{\mathcal{S}}(\lambda) := Q\mathcal{S}(\lambda)Z$ is in a Kronecker-like staircase form

$$\tilde{\mathcal{S}}(\lambda) = \left[\begin{array}{c|c} A_r - \lambda E_r & A_{r,l} - \lambda E_{r,l} \\ \hline 0 & A_l - \lambda E_l \\ \hline 0 & C_l \end{array} \right] \quad (11)$$

where the descriptor pair $(A_l - \lambda E_l, C_l)$ is observable, E_l is non-singular, and $A_r - \lambda E_r$ has full row rank excepting possibly a finite set of values of λ (i.e. the invariant zeros of $\mathcal{S}(\lambda)$). We can choose a left nullspace $\tilde{Y}_l(\lambda)$ of $\tilde{\mathcal{S}}(\lambda)$ in the form

$$\tilde{Y}_l(\lambda) = [0 \mid C_l(\lambda E_l - A_l)^{-1} \mid I]. \quad (12)$$

Then the left nullspace of $\mathcal{S}(\lambda)$ is

$$Y_l(\lambda) = \tilde{Y}_l(\lambda)Q \left[\begin{array}{c} 0 \\ I_{p_e} \end{array} \right].$$

If we partition row-wise

$$Q \left[\begin{array}{c} 0 \\ I_{p_e} \end{array} \right] = \left[\begin{array}{c} B_{r,l} \\ B_l \\ D_l \end{array} \right]$$

to comply with the column partitioning of $\tilde{Y}_l(\lambda)$ in (12), we obtain a minimal order proper left nullspace $N_l(\lambda)$ of $G(\lambda)$ with the descriptor realization

$$N_l(\lambda) = \left[\begin{array}{c|c} A_l - \lambda E_l & B_l \\ \hline C_l & D_l \end{array} \right] \quad (13)$$

An important aspect of this computational procedure is that we can explicitly write down also a realization of $N_f(\lambda) = N_l(\lambda)F(\lambda)$ [Varga, 2008a]. It is easy to show that

$$Y_l(\lambda) \left[\begin{array}{c|c} A - \lambda E & B_F \\ \hline C & D_F \end{array} \right] = [0 \mid N_l(\lambda)F(\lambda)]$$

and thus

$$N_l(\lambda)F(\lambda) = Y_l(\lambda) \left[\begin{array}{c} B_F \\ D_F \end{array} \right] = \tilde{Y}_l(\lambda)Q \left[\begin{array}{c} B_F \\ D_F \end{array} \right].$$

We compute now

$$Q \left[\begin{array}{c} B_F \\ D_F \end{array} \right] = \left[\begin{array}{c} * \\ \tilde{B}_F \\ \tilde{D}_F \end{array} \right], \quad (14)$$

where the row partitioning of the right hand side corresponds to the column partitioning of $\tilde{Y}_l(\lambda)$ in (12). The realization of $N_f(\lambda) = N_l(\lambda)F(\lambda)$ results as

$$N_f(\lambda) = \left[\begin{array}{c|c} A_l - \lambda E_l & \tilde{B}_F \\ \hline C_l & \tilde{D}_F \end{array} \right]. \quad (15)$$

Note that both $N_l(\lambda)$ and $N_f(\lambda)$ share the same observable pair $(A_l - \lambda E_l, C_l)$.

To obtain the state space realizations of both $N_l(\lambda)$ and $N_f(\lambda)$, we performed exclusively orthogonal transformations on the system matrices. We can prove that all computed matrices are exact for a slightly perturbed original system matrix. It follows that the algorithm to compute these realizations is *numerically backward stable*.

The structure matrix S corresponding to $N_f(\lambda)$ results by examining the transfer functions corresponding to individual entries of $N_f(\lambda)$. This can be done, for example,

by checking for non-zero values at a random frequency (for *weak detectability*) or at all values $\lambda_s \in \Omega_s$ (for *strong detectability*). This merely involves evaluating $N_f(\lambda)$ at several frequency values.

To avoid numerical problems in evaluating $N_f(\lambda_s)$ (e.g., due to poles nearby to λ_s), we can easily enforce the stability of the resulting nullspace basis. This comes down to additionally perform a similarity transformation on the reduced pencil $\tilde{\mathcal{S}}(\lambda)$ with a left transformation matrix of the form

$$U = \left[\begin{array}{ccc} I & 0 & 0 \\ 0 & I & K \\ 0 & 0 & I \end{array} \right]$$

and compute $\hat{\mathcal{S}}(\lambda) := UQ\mathcal{S}(\lambda)Z$, which is still in a Kronecker-like staircase form

$$\hat{\mathcal{S}}(\lambda) = \left[\begin{array}{c|c} A_r - \lambda E_r & A_{r,l} - \lambda E_{r,l} \\ \hline 0 & A_l + KC_l - \lambda E_l \\ \hline 0 & C_l \end{array} \right] \quad (16)$$

If we form now

$$UQ \left[\begin{array}{c} 0 \\ I_p \end{array} \right] = \left[\begin{array}{c} B_{r,l} \\ B_l + KD_l \\ D_l \end{array} \right],$$

we obtain an alternative minimal proper rational basis in the form

$$\tilde{N}_l(\lambda) = \left[\begin{array}{c|c} A_l + KC_l - \lambda E_l & B_l + KD_l \\ \hline C_l & D_l \end{array} \right] \quad (17)$$

In a similar way as above, we can compute the corresponding realization of $\tilde{N}_f(\lambda) = \tilde{N}_l(\lambda)F(\lambda)$ as

$$\tilde{N}_f(\lambda) = \left[\begin{array}{c|c} A_l + KC_l - \lambda E_l & \tilde{B}_F + K\tilde{D}_F \\ \hline C_l & \tilde{D}_F \end{array} \right]$$

Since the pair $(A_l - \lambda E_l, C_l)$ is observable, the generalized eigenvalues of the pair $(A_l + KC_l, E_l)$ can be arbitrarily assigned. For practical purposes, a randomly generated small norm K is also sufficient to check the fault detectability conditions.

For the example considered in the previous section, to compute $F^{2,1}(\lambda)$, the updating formulas above are employed three times, first to determine $F^0(\lambda)$, then $F^1(\lambda)$ and finally $F^{2,1}(\lambda)$. The first two computations can be reused to compute $F^{2,1}(\lambda)$. Similarly, to compute $F^{3,2}(\lambda)$, the updating formulas are employed only two times, provided the expression computed for $F^0(\lambda)$ can be reused. Finally, only $F^3(\lambda)$ need to be computed, and once again $F^0(\lambda)$ can be reused. This leads to a substantial saving of computational effort compared to the case when an exhaustive list of possible specifications is checked in a sequential manner.

We can easily compare the necessary computational efforts of performing an exhaustive search using a brute force sequential checking of all potential specifications and the proposed recursive procedure relying on the above explicit realizations. In a brute force procedure, each possible specification is checked by determining a corresponding left nullspace basis. This can be done using the above approach, where we redefine some of faults as fictive disturbances. If the corresponding FDP is solvable (this check can be done easily by just checking for nonzero columns in the input or feedthrough matrices of the state space realization of $N_f(\lambda)$), then the tested specification is

achievable. The minimum number of specifications to be tested is k_S given by (6). However, to discover non-generic structures with larger number of zeros, usually a much larger number of specification must be checked (see the example in Section 5). For each test, we need to perform the reduction to a Kronecker-like form of the original system, for which we need to perform $O(n^3)$ operations.

When performing the **FDISPEC Procedure**, the state dimensions of the realizations of subsequent annihilators in a chain like $N_l^0(\lambda)$, $N_l^i(\lambda)$, $N_l^{j,i}(\lambda)$, \dots , $N_l^{k,\dots,j,i}(\lambda)$ are non-increasing and therefore the nullspace computations involves systems of orders $n \geq n^i \geq n^{j,i} \geq \dots \geq n^{k,\dots,j,i}$, respectively. It follows that even in the case when a minimum number of specifications are checked in the brute search procedure, the difference in computational efforts can be still huge, taking into account the cubic dependence of the operation count of the problem dimension. Thus, the usage of the proposed procedure leads to a substantial saving of computational effort compared to the case when an exhaustive list of possible specifications is sequentially checked. An important feature of the new approach is that the whole recursive computations can be performed by using exclusively well-conditioned similarity transformations. A potential disadvantage of the new approach is the large storage requirements, because of the need to store the results of many intermediary computations.

4. COMPUTATION OF ASSOCIATED INFORMATION

Besides computing the signature information, often additional information is useful for a successful synthesis of a bank of residual generators. An important aspect is the determination of the least order scalar output detector corresponding to a given achievable fault signature. This information can be determined using minimal dynamic cover techniques as described in [Varga, 2007]. Here additionally a suitable vector h and a corresponding K are determined such that

$$h\tilde{N}_l(\lambda) = \left[\frac{A_l + KC_l - \lambda E_l}{hC_l} \mid \frac{B_l + KD_l}{hD_l} \right] \quad (18)$$

has the least McMillan degree. The choice of h corresponds to selecting linear combinations of basis vectors (e.g., rows of $\tilde{N}_l(\lambda)$) up to a given order and checking the agreement with the corresponding signature matrix. For details on how to select candidate h vectors, see also [Varga, 2008a].

Several performance measures for fault signatures are described in [Gertler, 1998, page 353]. These measures can be employed to choose among several possible specifications when defining a desirable fault signature to solve a FDIP. Of particular interest is the so-called *sensitivity condition*, which for a given $N_f(\lambda)$ can be defined as

$$\xi := \max_j \|N_{f_j}(\lambda)\|_\infty / \min_j \|N_{f_j}(\lambda)\|_\infty$$

where the minimum is taken over those columns of $N_f(\lambda)$ which correspond to non-zero entries of a selected specification. For strong detectability assessment, a similar *sensitivity condition* can be defined in terms of the gains at a selected frequency λ_s as

$$\xi^s := \max_j \|N_{f_j}(\lambda_s)\|_2 / \min_j \|N_{f_j}(\lambda_s)\|_2$$

A large value of the sensitivity condition indicates potential difficulties in detecting faults due to a substantial gap between the maximum and minimum gains. The definitions used in [Gertler, 1998] are more complicated, involving the minimum values of detectable faults and even the threshold values used for detection. The definitions used above correspond to minimum faults normalized to unity.

5. EXAMPLE.

This is a fourth order continuous-time system with output dimension $p = 3$, and input dimensions $m_u = 1$, $m_d = 0$ and $m_f = 8$ which has been used by Yuan et al. [1997] in a study to solve a FDIP for the complete achievable structure matrix S . For completeness we reproduce the state-space matrices here

$$A = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 0 & 1 & -2 & 1 \\ 0 & 0 & 1 & -2 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$B_u = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad B_f = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & -1 & 1 \end{bmatrix}$$

and $D_u = 0$, $D_f = 0$. This academic example is primarily intended to illustrate the main advantages of the proposed algorithm over brute force search based techniques and the ability to easily determine useful associated information. More realistic examples are studied, for example, in [Varga, 2009].

First we illustrate the performance of our algorithm by comparing it to an exhaustive search over all possible specifications. The resulting timing values have been obtained on a 3.4 GHz Pentium IV based PC with MATLAB running under Windows XP. In a first run, we checked all $2^8 - 1 = 255$ possible specifications. Only 18 specifications out of 255 are feasible. This computation took 7.4 seconds. In comparison, the time required by the proposed **FDISPEC Procedure** was only 1.1 seconds, which is about twice as much as that for just checking the 18 achievable specifications. Thus, even for this very low order system, the computational effort is about 7 times larger for exhaustive checking than for our optimized search. For larger order system, it is expected this difference is substantially higher.

For the considered example, we can try to reduce the number of specifications to be checked taking into account the concrete problem dimensions. According to (6), there are $k_S = 93$ specifications containing at most 3 zeros. Checking these 93 specifications takes 2.7 seconds only, but only 16 specifications out of 93 are feasible. To discover the missing 2 specifications, we need to check additionally 126 specifications having 4 or 5 zeros. Thus the minimum amount of time required to detect all feasible specifications is 16.3 seconds and corresponds to check $93+126 = 219$ specifications out of 255.

This example illustrates that discovering non-generic specifications can substantially increase the necessary computational effort for the exhaustive search, and therefore the only reasonable approach is to check all $2^{m_f} - 1$ possible specifications. In contrast, the procedure proposed in this paper is able to discover also non-generic specifications, because at each step several specifications are determined which contain the non-generic specifications too.

For $\Omega_s = \{0\}$, we computed the achievable structure matrix S , which in comparison to that in [Yuan et al., 1997], contains supplementary information on weekly detectable specifications. For the resulting specifications we computed the achievable least orders n_i for the i -th specification as well as the corresponding sensitivity conditions ξ_i and ξ_i^s . The results are presented in Table 1, where groups of specifications with the same number of zeros are delimited. As can be observed, there are 6 groups of specifications, where each group contains 0, 1, ..., 5 zeros. In each group there are respectively $\{1, 6, 5, 4, 1, 1\}$ specifications (to be compared with the generic case $\{1, 8, 28, 56, 0, 0\}$).

Table 1. Achievable specifications S

	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	n_i	ξ_i	ξ_i^s
#1	1	1	1	1	1	1	1	1	2	3.0	3.0
#2	0	1	1	1	1	1	1	1	2	2.0	2.0
#3	1	0	1	1	1	1	1	1	1	2.0	2.0
#4	1	1	0	1	1	1	1	1	2	2.0	2.0
#5	1	1	1	1	0	-1	1	1	2	1.7	∞
#6	1	1	1	1	-1	0	1	1	2	2.5	∞
#7	1	1	1	1	1	1	0	1	2	3.0	3.0
#8	0	1	1	1	1	1	0	1	2	2.0	2.0
#9	1	0	1	1	1	1	0	1	1	1.0	1.0
#10	1	1	1	0	1	1	1	0	2	3.0	3.0
#11	1	1	1	1	0	-1	0	1	2	1.5	∞
#12	1	1	1	1	-1	0	0	1	2	1.5	∞
#13	0	1	1	0	1	1	1	0	2	2.0	2.0
#14	1	0	1	0	1	1	1	0	1	1.0	1.0
#15	1	1	1	0	0	-1	1	0	2	1.5	∞
#16	1	1	1	0	-1	0	1	0	2	1.5	∞
#17	1	1	0	0	1	1	0	0	2	2.0	2.0
#18	0	0	0	1	0	0	1	1	1	1.0	1.0

Each line of S has been realized by a detector of order 1 or 2 with eigenvalues $\{-1\}$ or $\{-1, -2\}$. The corresponding sensitivity conditions correspond to the best values of h in (18). As it can be observed, several lines of S can not be used for strong detection of all detectable faults. Of interest for designing least order detectors is the fact that although the total order of the resulting bank of detectors is 32, however, the McMillan degree of the global detector is 6. Thus, with a detector of order 6 the complete set of 18 distinct specifications can be achieved. We claim that this is the least achievable order among all possible detectors which provides the full structure matrix for this example. For comparison, the "least" order of the global detector determined in Yuan et al. [1997] was 11.

6. CONCLUSIONS

An efficient and numerically reliable procedure has been proposed to determine the achievable structure matrix of a system with additive faults. The gain of efficiency compared to a brute force search over all possible specifications results both from checking a minimal number of specifications, and also from reusing previous results by

formulating our algorithm as a recursive procedure. The main qualitative difference between these two approaches is that while the brute force search is done for all possible specifications, the proposed algorithm performs a search over all possible detectors expressed as chains of nullspaces of decreasing dimensions and decreasing McMillan degrees. The basis for that are efficient updating techniques of the state-space realizations of the detectors and of faults-to-residual TFMs. For standard state space systems, similar updating techniques based on orthogonal reductions have been employed by Nikoukhah [1994] as basis for solving stochastic fault detection problems of increased complexity.

The performance of the proposed method has been illustrated only on an academic example. For this purpose, an implementation of the **FDISPEC Procedure** in MATLAB based on reverse communication techniques available in the latest version of a Fault Detection Toolbox for Matlab [Varga, 2008b]. This tool also represented a valuable support for the synthesis of structured residuals for a complete fault diagnosis system of primary actuator faults for a large transport aircraft [Varga, 2009].

REFERENCES

- Th.G.J. Beelen. *New Algorithms for Computing the Kronecker structure of a Pencil with Applications to Systems and Control Theory*. Ph. D. Thesis, Eindhoven University of Technology, 1987.
- E. Frisk and M. Nyberg. A minimal polynomial basis solution to residual generation for fault diagnosis in linear systems. *Automatica*, 37:1417–1424, 2001.
- J. Gertler. Designing dynamic consistency relation for fault detection and isolation. *Int. J. Control*, 73:720–732, 2000.
- J. Gertler. *Fault Detection and Diagnosis in Engineering Systems*. Marcel Dekker, New York, 1998.
- R. Nikoukhah. Innovations generation in the presence of unknown inputs: application to robust failure detection. *Automatica*, 30:1851–1867, 1994.
- A. Varga. On computing least order fault detectors using rational nullspace bases. *Proc. of IFAC Symp. SAFEPROCESS'2003, Washington D.C.*, 2003.
- A. Varga. On designing least order residual generators for fault detection and isolation. *Proc. 16th Internat. Conf. on Control Systems and Computer Science, Bucharest, Romania*, pp. 323–330, 2007.
- A. Varga. On computing nullspace bases – a fault detection perspective. *Proc. IFAC 2008 World Congress, Seoul, Korea.*, pp. 6295–6300, 2008a.
- A. Varga. Linear FDI-Techniques and Software Tools. FAULT DETECTION Toolbox V0.8 – Technical Documentation IB 515-08-18, German Aerospace Center (DLR), Institute of Robotics and Mechatronics, 2008b.
- A. Varga. Monitoring actuator failures for a large transport aircraft – the nominal case. *Proc. SAFEPROCESS'09, Barcelona, Spain*, 2009.
- A. Varga. Computation of Kronecker-like forms of a system pencil: Applications, algorithms and software. *Proc. CACSD'96 Symposium, Dearborn, MI*, pp. 77–82, 1996.
- Z. Yuan, G. C. Vansteenkiste, and C. Y. Wen. Improving the observer-based FDI design for efficient fault isolation. *Int. J. Control*, 68(1):197–218, 1997.