Residual Generation in Stochastic Systems -
A Polynomial Approach

Erik Frisk
email: frisk@isy.liu.se
Dept. of Electrical Engineering
Linköping University, Sweden

Abstract
This report describes a polynomial design algorithm for linear residual generation for stochastic systems in both continuous and discrete time. It is shown how the two main steps in the design algorithm is extraction of a polynomial basis for the left null-space of a polynomial matrix followed by a J-spectral co-factorization of a para-hermitian polynomial matrix. For both these operations there exists good numerical tools. The design algorithm is successfully demonstrated on a number of non-trivial examples. Full Matlab implementations is also provided.

Contents
1 Introduction 3
2 Problem formulation 3
3 Spectral factorization theory 6
  3.1 Note on the singular case 7
4 Introductory examples 7
  4.1 Example 1: Successful design 7
  4.2 Example 2: Zeros on the imaginary axis 8
  4.3 Example 3: Infinite zeros 8
5 Design algorithm 8
  5.1 Summary of design algorithm 12
    5.1.1 Innovations filter design - Theorem 11 12
    5.1.2 Whitening residual generator design - Theorem 8 12
6 Design examples 13
  6.1 Design Example: Aircraft Dynamics 13
    6.1.1 Process and measurement noise 13
    6.1.2 Only process noise 14
    6.1.3 Noise on all states and sensor 3 16
  6.2 Example with purely imaginary zeros 17
7 White residuals 17
  7.1 Whiteness tests 18
  7.2 Simulations and Comparisons 18
  7.3 Simulation conclusions 19
8 Time-discrete systems

9 The singular case
  9.1 Singular complications ........................................ 24
  9.2 When does the non-singular case occur? ..................... 25

10 Conclusions .......................................................... 25

A Matlab functions ....................................................... 27
  A.1 innovationsfilter.m ............................................. 27
  A.2 whiteresgen.m .................................................. 28
1 Introduction

This work deals with residual generation for fault diagnosis in linear systems. A residual is a fault-sensitive signal that is produced by filtering known signals, i.e. the control signals and measured signals. The residual should, ideally, be zero in the fault-free case regardless of any unknown disturbances and non-zero in case of a fault.

For deterministic systems, this is a well studied area (Wünneberg, 1990; Massoumnia et al., 1989; Nikoukhah, 1994; Chow and Willsky, 1984; Gertler, 1991) to name but a few. In previous works, (Frisk and Nyberg, 2001, 1999; Nyberg and Frisk, 1999), design algorithms and analysis tools were developed based on polynomial methods, instead of parity-space based approaches or observer-based approaches. The polynomial approach proved to be very well suited to answer many fundamental questions regarding e.g. complexity of residual generators, simple parameterization of residual generators.

For stochastic linear systems, i.e. noise affected linear systems, there is not as much work published. A common approach for these systems is to use Kalman-filters as residual generators which then produces residuals that is zero-mean and white with known covariance. The drawback of this approach is that systems subjected to unknown inputs which cannot, in any reasonable way, be modeled as random processes with known statistics, is not handled. This is often the case for the fault isolation task were a subset of the monitored faults must be, statistically or ideally, decoupled in the residual. Such detailed information about fault signals is rarely available and often, just modelling the fault influence on the process is difficult enough. This means that the diagnosis decision should not be based on any residual that is “corrupted” by these unknown signals, i.e. they should be decoupled in the residual.

A fundamental contribution to this problem is given in a nice paper by Nikoukhah (1994). Here, the aim is to extend the polynomial methods that proved beneficial in the deterministic case to the stochastic case and address problems posed in (Nikoukhah, 1994) and also extend the problem formulation and solve a slightly more general problem. The main algorithmic tool is J-spectral co-factorization which is shown to quite nicely handle the stochastic problem. Algorithms for spectral factorization of polynomial matrices has recently received much attention since it plays a fundamental role in the solution of polynomial $\mathcal{H}_\infty$- (Green et al., 1990) and $\mathcal{H}_2$-(Kwakernaak, 2000b) standard problems. Therefore, feasible and numerically appealing algorithms and implementations has been proposed (Kwakernaak and Šebek, 1994; Kwakernaak, 2000a).

2 Problem formulation

The system under consideration is described by the following class of models:

$$ y = G_u(s)u + G_d(s)d + G_f(s)f + G_n(s)n \quad (1) $$

where $y \in \mathbb{R}^{k_y}$ is the measurement vector, $u \in \mathbb{R}^{k_u}$ control signals, $d \in \mathbb{R}^{k_d}$ unknown disturbances, $f \in \mathbb{R}^{k_f}$ faults, $n \in \mathbb{R}^{k_n}$ noise, and $G_u(s)$, $G_d(s)$, $G_f(s)$, and $G_n(s)$ are proper transfer matrices of suitable dimensions. The difference between the disturbances $d$ and the noise $n$ is that the disturbances is assumed to have no stochastic description and must be decoupled while the noise is modeled as a white stationary stochastic process with unit covariance. The noise is not decoupled but is handled otherwise.

Before algorithms for design and analysis of residual generators can be presented, definitions of residual generators for stochastic systems is needed. In the noise-free case ($G_n(s) = 0$), residual generators can be defined as follows:

**Definition 1 (residual generator for deterministic systems).** A stable and proper linear filter $Q(s)$ is a residual generator for (1) with $G_n(s) = 0$ if and only if when $f \equiv 0$ it holds that

$$ r = Q(s) \begin{pmatrix} y \\ u \end{pmatrix} \equiv 0 $$

for all $u$, $d$.
Note that for the residual to be useful for fault detection it must also hold that the transfer function from faults to the residual is non-zero. Sometimes this requirement is also included in the definition of the residual generator. Also, the requirement that the residual should be zero in the fault-free case is too strict in the general case since perfect decoupling is not always possible even in the deterministic case and we have to resort to approximate decoupling of disturbances.

In spite of this, Definition 1 is used here as a basis for the stochastic case since and it is assumed that perfect decoupling of disturbances \( d \) is possible. In case perfect decoupling of \( d \) is not possible, signals may have to be transferred from \( d \) to \( n \) and the model augmented with stochastic descriptions of these signals.

For linear models with no unknown inputs, the innovations process associated with the Kalman filter is often used as a residual because of its zero-mean and whiteness properties in the fault-free case. Once the innovations is generated, the fault decision problem reduces to a whiteness test of the residual. Also, other more elaborate decision algorithms can be used based on more deep utilization of stochastic properties of the residual (Basseville and Nikiforov, 1993).

Trying to achieve the same properties but also including unknown disturbances in the system leads to the following definition.

**Definition 2 (Whitening residual generator).** A stable and proper linear filter \( Q(s) \) is a residual generator for (1) if and only if when \( f \equiv 0 \) it holds that

\[
r = Q(s) \begin{pmatrix} y \\ u \end{pmatrix}
\]

is zero mean and white for all \( u, d \).

Of course, for the residual generator to be useful for fault detection, when \( f \neq 0 \) the zero mean whiteness property need to be violated.

Finally, a restricted class of residual generators defined by Nikoukhah (1994), where the whiteness property of the residual is achieved without restricting the number of linearly independent residuals.

**Definition 3 (Innovations filter).** A finite-dimensional linear time-invariant system \( Q(s) \) is called an innovations-filter for system (1) if it is stable with the least number of outputs such that, in the absence of failure,

1. its output

\[
r = Q(s) \begin{pmatrix} y \\ u \end{pmatrix}
\]

is zero-mean, white and decoupled from \( u \) and \( d \),

2. if \( Q'(s) \) is any finite-dimensional linear time-invariant system such that

\[
r' = Q'(s) \begin{pmatrix} y \\ u \end{pmatrix}
\]

is decoupled from \( u \) and \( d \), then there exists a linear system \( L(s) \) such that \( Q'(s) = L(s)Q(s) \).

**Assumption:** From now on it is assumed that perfect decoupling of both the noise \( n \) and disturbances \( d \) is not possible. A brief discussion on the singular case that arises when the noise is perfectly decouplable is presented in Section 9.

**Problem formulation:** Under the assumption above, find synthesis algorithms for whitening residual generators(WRG) and innovation filters based on polynomial methods.

Now, characterization of residual generators and innovations filters will be derived and presented in Theorem 1 and Theorem 2. Let the fault-free system (1) be described by a state-space realization on the form

\[
\dot{x} = Ax + B_u u + B_d d + B_n n \\
y = C x + D_u u + D_d d + D_n n
\]
Previous works, e.g. (Frisk and Nyberg, 2001), gives that any deterministic residual generator \( Q(s) \) can be written as \( Q(s) = \varphi(s)N_{M_1}(s)P_z \) where

\[
M_x(s) = \begin{bmatrix} C & D_d \\ -(sI_{n_x} - A) & B_d \end{bmatrix} \quad \text{and} \quad P_z = \begin{bmatrix} I_{km} & -D_u \\ 0_{n_x \times km} & -B_u \end{bmatrix}
\]

where \( n_x \) is the number of states i.e. the size of \( x \) and \( N_{M_1}(s) \) is a minimal polynomial basis for the left null-space of \( M_x(s) \). All available design freedom lies in the choice of the rational row-vector \( \varphi(s) \) which here will be used to realize whitening residual generators and innovations filters. One constraint on \( \varphi(s) \) is that \( Q(s) \) must be realizable and stable. Details on the algorithm and ways of computing \( N_{M_1}(s) \) can be found in e.g. (Frisk and Nyberg, 2001; Nyberg and Frisk, 1999; Frisk and Nyberg, 1999).

**Theorem 1.** A transfer matrix \( Q(s) \) is a whitening residual generator for (2) if and only if there exists a \( \varphi(s) \) such that

\[
Q(s) = \varphi(s)N_{M_1}(s)P_z
\]

is proper, stable and it holds that

\[
\forall s, H(s)H^T(-s) = \Psi
\]

where \( H(s) = \varphi(s)N_{M_1}(s) \begin{bmatrix} D_n \\ B_n \end{bmatrix} \) and \( \Psi \) is a constant matrix.

**Proof.** As shown in (Frisk and Nyberg, 2001), all disturbance decoupling residual generators can be written as

\[
Q(s) = \varphi(s)N_{M_1}(s)P_z
\]

Insertion of (2) into \( r = Q(s) \left( \begin{array}{c} y \\ u \end{array} \right) \) gives, after some straightforward calculations

\[
r = \varphi(s)N_{M_1}(s)P_x \begin{bmatrix} y \\ u \end{bmatrix} = \varphi(s)N_{M_1}(s) \begin{bmatrix} C & D_d \\ -(sI - A) & B_d \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} = \varphi(s)N_{M_1}(s) \begin{bmatrix} D_n \\ B_n \end{bmatrix} n
\]

For \( r \) to be white it must hold that the spectrum of \( r \), \( \Phi_r(j\omega) \) is constant for all \( \omega \) which is equivalent to \( \Phi_r(s) \) is constant for all \( s \). The spectrum \( \Phi_r(s) \) can be written as

\[
\Phi_r(s) = \varphi(s)N_{M_1}(s)\Phi_n(s)N_{M_1}^T(-s)\varphi^T(-s)
\]

and since \( \Phi_n(s) \) was assumed to be \( \Phi_n(s) = I \), the theorem follows immediately. \( \Box \)

**Theorem 2.** A transfer matrix \( Q(s) \) is an innovations filter for system (2) if and only if there exists a \( \varphi(s) \) such that

\[
Q(s) = \varphi(s)N_{M_1}(s)P_z
\]

is proper, stable and it holds that

\[
\forall s, H(s)H^T(-s) = \Psi
\]

where \( H(s) = \varphi(s)N_{M_1}(s) \begin{bmatrix} D_n \\ B_n \end{bmatrix} \), \( \Psi \) is a constant matrix, and \( \varphi(s) \) is a square full-rank matrix.

**Proof.** To fulfill the first requirement of Definition 3, follow the same lines as in the proof of Theorem 1. Left to prove is that the second requirement of Definition 3 requires \( \varphi(s) \) to be square and full-rank matrix.

Any disturbance decoupling residual generator \( Q'(s) \) can be written

\[
Q'(s) = \varphi'(s)N_{M_1}(s)P_x
\]

Requirement 2 requires that for any \( Q(s) \) there must exist an \( L(s) \) such that \( Q'(s) = L(s)Q(s) \), i.e.

\[
\varphi'(s)N_{M_1}(s)P_x = L(s)\varphi(s)N_{M_1}(s)P_x
\]
Since this must hold for any \( \varphi'(s) \), there is a solution \( L(s) \) if and only if
\[
\varphi'(s) = L(s)\varphi(s)
\]
has a solution \( L(s) \) for any \( \varphi'(s) \) which gives that \( \varphi(s) \) must be invertible, i.e. square and full-rank.

Before the proposed algorithm can be described, some theory on polynomial J-spectral factorization is needed.

## 3 Spectral factorization theory

Material for this section mainly from (Kwakernaak and Šebek, 1994; Šebek, 1990), but also from (Ježek and Kučera, 1985; Callier, 1985). A corresponding discrete time version of this theory is also available and the time-discrete case is discussed in Section 8.

A polynomial matrix \( Z(s) \) is said to be para-hermitian if \( Z^T(s) = Z(s) \). Para-hermitian is sometimes abbreviated as p.h. Only matrices \( Z(s) \) with real coecients are considered in this work. A factorization
\[
Z(s) = P^T(-s)JP(s)
\]
is called a spectral factorization if \( J \) is a signature matrix and \( P(s) \) a square matrix with real coefficients such that \( \det P(s) \) is Hurwitz, i.e. all zeros in the closed left half plane. The signature matrix \( J \) has the following form
\[
J = \begin{bmatrix}
I_1 & 0 & 0 \\
0 & -I_2 & 0 \\
0 & 0 & 0
\end{bmatrix}
\]
A factorization on the form \( Z(s) = P(s)JP^T(-s) \) is called a spectral cofactorization.

No necessary and sufficient existence conditions are known for J-spectral factorization (Kwakernaak and Šebek, 1994). However, the following necessary condition due to (Jakubović, 1970) (referred in (Kwakernaak and Šebek, 1994)) gives a necessary condition.

**Theorem 3 (Existence of J-Spectral Factorization).** Suppose that the multiplicity of the zeros on the imaginary axis of each of the invariant polynomials of the para-hermitian polynomial matrix \( Z(s) \) is even, then \( Z(s) \) has a spectral factorization \( Z(s) = P^T(-s)JP(s) \).

A related issue is uniqueness of the factorization

**Theorem 4 (Non-uniqueness of J-Spectral Factorization).** Let the polynomial Matrix \( P(s) \) be a spectral factor of the full-rank para-Hermitian matrix \( Z(s) \) with corresponding signature matrix \( J \).

1. All other spectral factors of \( Z(s) \) are of the form \( U(s)P(s) \) with \( U(s) \) unimodular such that
\[
U^T(-s)JU(s) = J
\]
Matrix \( U(s) \) is said to be a J-unitary unimodular matrix.

2. If the factorization is canonical, i.e. \( P(s) \) is column reduced, any other spectral factor is on the form \( UP(s) \) with \( U \) constant J-unitary.

**Theorem 5.** Let \( Z(s) \) be positive definite on the imaginary axis, then the J-spectral factorization of \( Z(s) \) is canonical. Let \( P(s) \) be a spectral factor, then it holds that

1. \( P(s) \) is column reduced
2. The column-degrees of \( P(s) \) equals the half diagonal degrees, i.e. half the degrees of the diagonal entries, of \( Z(s) \).
3.1 Note on the singular case

Now follows a brief discussion on spectral (co-)factorization of singular para-hermitian matrices which will be used in discussions in Section 9. This presentation follows (Sebek, 1990).

Let \( Z(s) \) be a p.h. \( n \times n \) matrix of rank \( m < n \). First, find the greatest right divisor of \( Z(s) \), i.e. find \( Z_R(s) \) and a unimodular \( U(s) \) such that

\[
Z(s) = U(s) \begin{bmatrix} Z_R(s) \\ 0 \end{bmatrix}
\]

By symmetrical extraction of \( U(s) \) we get

\[
Z(s) = U(s) \begin{bmatrix} \hat{Z}(s) \\ 0 \end{bmatrix} U^T(-s)
\]

where \( \hat{Z}(s) \) is a square \( m \times m \) non-singular p.h. matrix. Let a \( \hat{P}(s) \) and \( \hat{J} \) be a spectral co-factor and the signature of \( \hat{Z}(s) \). Then a spectral co-factorization of \( Z(s) \) is given by

\[
Z(s) = U(s) \begin{bmatrix} \hat{P}(s) \\ 0 \end{bmatrix} \begin{bmatrix} \hat{J} \\ 0 \end{bmatrix} \begin{bmatrix} \hat{P}^T(-s) \\ 0 \end{bmatrix} U^T(-s)
\]

That is, a spectral co-factor and the signature of \( Z(s) \) can always be written as

\[
\hat{P}(s) = U(s) \begin{bmatrix} \hat{P}(s) \\ 0 \end{bmatrix}, \quad \hat{J} = \begin{bmatrix} \hat{J} \\ 0 \end{bmatrix}
\]

where \( \hat{P}(s) \) and \( \hat{J} \) is a spectral co-factor and signature of a non-singular p.h. matrix.

4 Introductory examples

Before going into detail, describing a design algorithm and existence conditions, three small illustrative examples are presented. The first describes a successful design and the last two illustrates the two cases when whitening residual generators do not exist.

4.1 Example 1: Successful design

Consider a system described by

\[
y = \begin{bmatrix} \frac{1}{s+1} \\ \frac{1}{s(s+1)} \end{bmatrix} u + \begin{bmatrix} 0 \\ 1 \end{bmatrix} d + \begin{bmatrix} 1 \\ 0 \end{bmatrix} f + \begin{bmatrix} n_1 \\ n_2 \end{bmatrix}
\]

The only residual generator that decouples \( d \) is parameterized by the free variable \( \varphi(s) \) as

\[
r = \varphi(s)[s + 1 \ 0 \ -1] \begin{bmatrix} y \\ u \end{bmatrix}
\]

Inserting (4) into (5) gives an internal form in the fault-free case

\[
r = \varphi(s)(s + 1)n_1
\]

It is clear that by letting \( \varphi(s) = \frac{1}{s+1} \) we get a white residual in the fault-free case by the stable and proper residual generator

\[
Q(s) = \begin{bmatrix} 1 \ 0 \\ -\frac{1}{s+1} \end{bmatrix}
\]
4.2 Example 2: Zeros on the imaginary axis
Consider the same example as above, but switch the positions of \( f \) and \( d \), i.e.
\[
y = \begin{bmatrix} \frac{1}{s+1} \\ \frac{1}{s+1} \end{bmatrix} u + \begin{bmatrix} 1 \\ 0 \end{bmatrix} d + \begin{bmatrix} 0 \\ 1 \end{bmatrix} f + \begin{bmatrix} n_1 \\ n_2 \end{bmatrix}
\]
In the same way as before it is clear that the all disturbance decoupling residual generators can be parameterized as
\[
r = \varphi(s)[0 \ s(s+1) - 1] \begin{bmatrix} y \\ u \end{bmatrix}
\]
for which the fault-free internal form is given by
\[
r = \varphi(s)s(s+1)n_2
\]
Here it is clear that no strictly stable \( \varphi(s) \) exists making \( r \) white, all because of the finite zero on the imaginary axis in the transfer function from \( n \) to \( r \).
This also shows a link to non strongly detectable faults (Chen and Patton, 1994). A zero at \( s = 0 \) will appear in the transfer function from \( n \) to \( r \) if \( n \) enters the system in the same way as a non strongly detectable fault \( f \) which was the case in the example above.

4.3 Example 3: Infinite zeros
Consider the scalar system
\[
y = \frac{1}{s+1} u + f + \frac{1}{(s+2)^2} n
\]
All residual generators can be written
\[
r = \varphi(s)[s+1 - 1] \begin{bmatrix} y \\ u \end{bmatrix}
\]
for which the internal form is
\[
r = \varphi(s) \frac{s+1}{(s+2)^2} n
\]
It is clear that for \( r \) to be white \( \varphi(s) = \frac{(s+2)^2}{s+1} \) which gives an improper, and thus non-realizable, residual generator
\[
r = [(s+2)^2 - \frac{(s+2)^2}{s+1}] \begin{bmatrix} y \\ u \end{bmatrix}
\]
And this was caused by the infinite zero of the transfer function \( \frac{s+1}{(s+2)^2} \) in (6).
Now, with these three cases in mind, a design algorithm is described in the next section.

5 Design algorithm
The main step in designing both residual generators for stochastic systems and innovations filters is to first compute \( N_M(s) \) in (3) and then find \( \varphi(s) \) such that \( Q(s) = \varphi(s)N_M(s)P_z \) is stable, proper, and the spectrum of \( r \) is constant for all \( s \). For the innovations filter, additional requirements of invertability of \( \varphi(s) \) is needed.
Now, existence conditions for proper and stable whitening residual generators and innovations filters will be derived. The proofs are constructive, outlining design algorithms that finds all possible WRG/innovations-filters.
First, let \( Z(s) \) denote
\[
Z(s) = N_M(s) \begin{bmatrix} D_n \\ B_n \end{bmatrix} \begin{bmatrix} D_n \\ B_n \end{bmatrix}^T N_M^T(-s) \in \mathbb{R}^{m \times m}(s)
\]
The following standard result is also needed
Theorem 6 (Division Theorem for Polynomial Matrices\textsuperscript{1}). Let $D(s)$ be an $m \times m$ non-singular polynomial matrix. Then, for any $p \times m$ polynomial matrix $N(s)$, there exists unique polynomial matrices $\{W(s), R(s)\}$ such that $N(s) = D(s)W(s) + R(s)$ and $D^{-1}(s)R(s)$ strictly proper.

Matrices $W(s)$ and $R(s)$ is called the polynomial matrix quotient and remainder of $D^{-1}(s)N(s)$.

Now follows a lemma characterizing the $\varphi(s)$ in Theorem 1 and 2.

Lemma 7. Assume $Z(s)$ full-rank. Then there exists a $\varphi(s)$ such that the linear time-invariant filter $Q(s) = \varphi(s)N_{M_z}(s)P_z$ produces white residuals if and only if $\varphi(s)$ can be written

$$\varphi(s) = \eta(s)P^{-1}(s)$$

where $P(s)$ is a spectral co-factor of $Z(s)$ and $\eta(s)\eta^T(-s) = \Psi$.

Proof. According to the proof of Theorem 1, the spectrum of $r$ can be written

$$\Phi_r(s) = \varphi(s)Z(s)\varphi^T(-s)$$

Note that $Z(s)$ is a p.h. polynomial matrix. Now, let $P(s)$ be a spectral co-factor and $J$ a signature of $Z(s)$, i.e.

$$Z(s) = P(s)JP^T(-s)$$

Since, according to assumption, $Z(s)$ is positive definite the signature of $Z(s)$ is $J = I_m$. Insertion of (9) into (8) and denoting $\eta(s) = \varphi(s)P(s)$ gives

$$\Phi_r(s) = \varphi(s)P(s)JP^T(-s)\varphi^T(-s) = \eta(s)\eta^T(-s)$$

which also gives that $\Phi_r(s) = \Psi$ according to the assumptions in the Theorem. The parameterization matrix $\varphi(s)$ is found by solving for $\varphi(s)$ in the equation

$$\eta(s) = \varphi(s)P(s)$$

which has only one unique solution $\varphi(s) = \eta(s)P^{-1}(s)$.

Remark: Any constant $\eta(s)$ yields a constant $\eta(s)\eta^T(-s)$ for all $s$. Note however that also non-constant $\eta(s)$ exists. One such example is $\eta(s) = \frac{1}{s+1}$. Insertion of (9) into (8) and denoting $\eta(s) = \varphi(s)P(s)$ gives

$$\Phi_r(s) = \varphi(s)P(s)JP^T(-s)\varphi^T(-s) = \eta(s)\eta^T(-s)$$

which also gives that $\Phi_r(s) = \Psi$ according to the assumptions in the Theorem. The parameterization matrix $\varphi(s)$ is found by solving for $\varphi(s)$ in the equation

$$\eta(s) = \varphi(s)P(s)$$

which has only one unique solution $\varphi(s) = \eta(s)P^{-1}(s)$.

A property of a whitening residual generator is that it must be strictly stable. The cause of non strictly-stable filters was, in the small scalar example in Section 4.2, the existence of zeros on the imaginary axis of the transfer function from noise to residual. To form necessary and sufficient conditions for the existence of whitening residual generators in the multidimensional case we need some notation. Assume $Z(s)$ has purely imaginary zeros $j\omega_i$, $i = 1, \ldots, r$ with multiplicities $2n_i$, $i = 1, \ldots, r$. Then, denote a polynomial

$$p_{\text{imag}}(s) = \prod_{i=1}^{r}(s - j\omega_i)^{n_i}$$

These are the poles we need to “factor out” to be able to form a strictly stable residual generator. Also denote

$$p_{\text{stab}}(s) = \prod_{i=1}^{q}(s - j\zeta_i)^{n_i}$$

where $\zeta_i, i = 1, \ldots, q$ are the strictly stable zeros of $Z(s)$ with multiplicities $2n_i$. This implies that for a spectral co-factor $P(s)$ of $Z(s)$ it holds that $\det(P(s)) = kp_{\text{stab}}(s)p_{\text{imag}}(s)$.

$$W(s) = \text{quotient } [P^{-1}(s)N_{M_z}(s)]$$

$$W = [W_1 \cdots W_d] \text{ where } W(s) = \sum_{i=0}^{d} s^iW_i$$

$$R(s) = \text{remainder } [(p_{\text{imag}}(s)I)^{-1}\text{adj } P(s)]$$

\textsuperscript{1}Theorem 6.3-15 in (Kailath, 1980).
Theorem 8. There exist an \( \eta(s) \) such that \( \eta(s)\eta^T(-s) = 1 \) and \( Q(s) = \eta(s)P^{-1}(s)N_{M_L}(s)P_x \) is proper and strictly stable if and only if
\[
N_L(W) \cap N_L(R(s)) \neq \emptyset
\]

Before the proof of the Theorem, a Lemma is needed

Lemma 9. Let \( \eta(s) \) be a \( 1 \times m \) row-vector of transfer functions such that \( \eta(s)\eta^T(-s) = \Psi \) with \( \Psi \neq 0 \). Then it holds that \( \eta(s) \) is proper and at least one element of \( \eta(s) \) is not strictly proper.

Proof. Since \( \eta(s)\eta^T(-s) = \Psi \), it holds that
\[
\Psi = \eta(j\omega)\eta^T(-j\omega) = \sum_{k=1}^{m} |\eta_k(j\omega)|^2
\]

Assume that element \( k \) of \( \eta(s) \) is improper, then the limit \( \lim_{\omega \to \infty} |\eta_k(j\omega)|^2 \) would not exist. Since for all \( l \), \( |\eta_l(j\omega)|^2 \geq 0 \), the improper assumption has lead to a contradiction, i.e. \( \eta(s) \) is proper.

Left to prove is that at least one element of \( \eta(s) \) is not strictly proper. Denote the constant coefficient matrix of \( \eta(s) \) with \( H_0 \), i.e. \( H_0 = \lim_{s \to \infty} \eta(s) \). The limit exists since \( \eta(s) \) is proper. Then,
\[
\eta(s) = H_0 + \tilde{\eta}(s)
\]
where it holds that \( \tilde{\eta}(s) \) is strictly proper. Now, assume \( H_0 = 0 \), i.e.
\[
\Psi = \eta(j\omega)\eta^T(-j\omega) = \tilde{\eta}(j\omega)\tilde{\eta}^T(-j\omega)
\]

Since all elements of \( \tilde{\eta}(s) \) is strictly proper, \( \lim_{\omega \to \infty} |\eta_k(j\omega)|^2 = 0 \) which is a contradiction, i.e. \( H_0 \neq 0 \) which means that at least one element of \( \eta(s) \) is not strictly proper. \( \square \)

Now, back to the proof of Theorem 8.

Proof. First, consider the properness requirement. Let \( H(s) = P^{-1}(s)N_{M_L}(s)P_x \) and let \( W(s) \) and \( E(s) \) be the polynomial matrix quotient and remainder of \( H(s) \) respectively, i.e.
\[
Q(s) = \eta(s)H(s) = \eta(s)W(s) + \eta(s)P^{-1}(s)E(s)
\]

First note that \( \eta(s) \) is proper according to Lemma 9. Also, since \( P^{-1}(s)E(s) \) is strictly proper it holds that \( Q(s) \) is proper if and only if \( \eta(s)W(s) \) is proper. Now, write \( \eta(s) \) as
\[
\eta(s) = d^{-1}(s)n(s)
\]
where \( d(s) \) is a scalar polynomial and \( n(s) \) is a polynomial row-vector of the same dimensions as \( \eta(s) \). Since \( \eta(s) \) is proper, but not strictly proper it must hold that
\[
\deg d(s) = \text{row-deg } n(s) \quad (14)
\]

Now, for \( \eta(s)W(s) \) to be proper it must hold that \( \deg d(s) \geq \text{row-deg } n(s) \). This together with but since \( (14) \) holds, the properness of \( Q(s) \) is equivalent to the existence of an \( n(s) \) such that
\[
\text{row-deg } n(s) = \text{row-deg } n(s)W(s) \quad (15)
\]

Now, let \( W_{\min}(s) \) denote a minimal polynomial basis for \( \text{Row-Im } W(s) \). Then the existence of an \( n(s) \) fulfilling \( (15) \) is equivalent to the existence of constant rows in \( W_{\min}(s) \). Denote such rows as of \( W_{\min}(s) \) with \( W_{\min,0} \). Then for any \( n(s) \) fulfilling \( (15) \) it holds that \( n(s) \in \text{Row-Im } W_{\min,0} \). Since \( W_{\min,0} \) is constant it can be easily computed by finding all constant \( \eta \) such that \( \eta W(s) \) is proper. Since \( W(s) \) is a polynomial matrix of degree \( d \) \( \eta W(s) \) can be written
\[
\eta W(s) = \eta W(0) + \sum_{i=1}^{d} \eta W_i s^i
\]
Thus, for $\eta W(s)$ to be proper it must hold that $\eta \in N_L(W)$, i.e.
\[
\text{Im } W_{\text{min,0}} = N_L(W)
\]
Now, consider the strict stability requirement. Let $V(s)$ and $R(s)$ be the polynomial matrix quotient and remainder of $(p_{\text{imag}}(s))^{-1}\text{adj } P(s)$. Direct calculations gives that
\[
\eta(s)P^{-1}(s) = \frac{1}{p_{\text{stab}}(s)p_{\text{imag}}(s)}\text{adj } P(s) = \eta(s)\frac{1}{p_{\text{stab}}(s)}\eta(s)V(s) + \frac{1}{p_{\text{stab}}(s)p_{\text{imag}}(s)}\eta(s)R(s)
\]
Thus, $\eta(s)P^{-1}(s)$ (and thereby $Q(s)$) is strictly stable if and only if $\eta(s)R(s) = 0$, i.e. $\eta(s) \in N_L(R(s))$.

Therefore, for $Q(s)$ to be proper and strictly stable it must hold that
\[
\eta(s)\eta^T(-s) = 1 \quad \land \quad \eta(s) \in N_L(R(s)) \cap N_L(W) \quad (16)
\]
If the intersection in (16) is non-empty there exist an $n(s)$ in $N_L(R(s)) \cap N_L(W)$. An $\eta(s)$ fulfilling (16) can then be found by letting $\eta(s) = d^{-1}(s)n(s)$ where $d(s)$ is a scalar polynomial selected to fulfill
\[
n(s)n^T(-s) = d(s)d(-s)
\]
which always exists.

Now, the following result is immediate.

**Corollary 10.** If $Z(s)$ is full rank with no zeros on the imaginary axis, a whitening residual generator exists if
\[
\exists i. \text{row-deg}_i N_{M_z}(s) \begin{pmatrix} D_n \\ B_n \end{pmatrix} = \text{row-deg}_i N_{M_z}(s)
\]

**Proof.** Since, according to assumption in the theorem, $Z(s)$ has no zeros on the imaginary axis, stability of the residual generator is assured. By Theorem 5, $P(s)$ is row-reduced and the row-degrees of $P(s)$ equals the row-degrees of $N_{M_z}(s) \begin{pmatrix} D_n \\ B_n \end{pmatrix}$. Thus, at least one row of
\[
P^{-1}(s)N_{M_z}(s)P_z
\]
is also proper which ends the proof.

**Theorem 11.** If $Z(s)$ is full rank, an innovation filter exists if and only if
\[
\forall i. \text{row-deg}_i N_{M_z}(s) \begin{pmatrix} D_n \\ B_n \end{pmatrix} = \text{row-deg}_i N_{M_z}(s)
\]
and $Z(s)$ has no roots on the imaginary axis.

**Proof.** According to Theorem 2 and Lemma 7, an innovations filter exists if and only if there exists an $\eta(s)$ such that
\[
Q(s) = \eta(s)P^{-1}(s)N_{M_z}(s)P_z
\]
is stable, proper, $\eta(s)\eta^T(-s)$ is constant, and $\varphi(s) = \eta(s)P^{-1}(s)$ is invertible.

Following the lines as in the proof for Theorem 8, for $Q(s)$ to be stable and proper it must hold that $\eta(s) \in N_L([W \ R(s)])$ and for $\varphi(s)$ to be invertible it must hold that $\eta(s)$ is square and invertible. Existence of such an $\eta(s)$ gives that $N_L([W \ R(s)]) = I$, i.e. $W = R(s) = 0$ which is equivalent to $Z(s)$ having no purely imaginary zeros and $P^{-1}(s)N_{M_z}(s)P_z$ proper.

Also, for $P^{-1}(s)N_{M_z}(s)P_z$ to be proper it must hold that since $P(s)$ is row reduced it must also hold that
\[
\text{row-deg}_i P(s) \geq \text{row-deg}_i N_{M_z}(s)P_z
\]
By Theorem 5, the row-degrees of $P(s)$ equals the row-degrees of $N_{M_{Lx}}(s)\begin{pmatrix} D_n \\ B_n \end{pmatrix}$, i.e. the properness condition is equivalent to
\[
\text{row-deg}_i N_{M_{Lx}}(s) \begin{pmatrix} D_n \\ B_n \end{pmatrix} \geq \text{row-deg}_i N_{M_{Lx}}(s) P_x
\]
which gives the theorem.

5.1 Summary of design algorithm
Now, the design procedures of innovations filters and whitening residual generators are summarized.

5.1.1 Innovations filter design - Theorem 11
1. Form $M_x(s)$ according to (3) and compute $N_{M_{Lx}}(s)$.
2. Form $Z(s)$ as in (7). An innovations filter exists if and only $Z(s)$ is strictly stable and
\[
\forall i \text{ row-deg}_i N_{M_{Lx}}(s) \begin{pmatrix} D_n \\ B_n \end{pmatrix} = \text{row-deg}_i N_{M_{Lx}}(s)
\]
3. All innovations filters, if any exists, is then given by:
\[
Q(s) = \Psi(s)P^{-1}(s)N_{M_{Lx}}P_x
\]
where $P(s)$ is a spectral co-factor of $Z(s)$ and $\Psi(s)$ is any invertible matrix such that $\Psi(s)\Psi^T(-s)$ is constant for all $s$.

5.1.2 Whitening residual generator design - Theorem 8
1. Form $M_x(s)$ according to (3) and compute $N_{M_{Lx}}(s)$.
2. Form $Z(s)$ as in (7) and compute a spectral co-factor $P(s)$. If $Z(s)$ has purely imaginary zeros, form the polynomials $p_{\text{imag}}(s)$ and $p_{\text{stab}}(s)$ according to (11) and (12).
3. Form $W$ and $R(s)$ according to (13). If $Z(s)$ has no purely imaginary zeros, let $R(s) = 0$. Existence conditions for whitening residual generators is given by Theorem 8.
4. All whitening residual generators can be formed as
\[
Q(s) = \eta(s)P^{-1}(s)N_{M_{Lx}}(s)P_x
\]
where $\eta(s) = d^{-1}(s)n(s)$ such that
\[
n(s) \in \mathbb{N}_L[W R(s)]
\]
and $d(s)$ is a scalar polynomial that fulfills
\[
n(s)n^T(-s) = \Psi d(s)d(-s)
\]
where $\Psi$ is a constant.
6 Design examples

This section includes 4 design examples that illustrates different aspects of the design problem and the proposed design algorithm. The first three examples is based around a linearized airplane model from (Maciejowski, 1989). This model has been used previously in e.g. (Frisk and Nyberg, 1999) to demonstrate the deterministic design problem. In the first example, a complete design of an innovations filter and a whitening residual generator is shown. In the second example the noise environment is changed and it is shown that no innovations filter or whitening residual generator exists. In the third example, using a third noise setup, it is shown that an innovations filter does not exists but a whitening residual generator exists that has acceptable fault sensitivity. The case with purely imaginary zeros is demonstrated in a final fourth example.

All calculations is done in Matlab using Polynomial Toolbox 2.0 for Matlab 5 (1998). All functions used is included in the toolbox. Included in Appendix A are full Matlab implementations of innovations filter/whitening residual generator design.

6.1 Design Example: Aircraft Dynamics

This model, taken from (Maciejowski, 1989), represents a linearized model of vertical-plane dynamics of an aircraft. The inputs and outputs of the model are

**Inputs**

| \( u_1 \): spoiler angle [tenth of a degree] |
| \( u_2 \): forward acceleration [ms\(^{-2}\)] |
| \( u_3 \): elevator angle [degrees] |

**Outputs**

| \( y_1 \): relative altitude [m] |
| \( y_2 \): forward speed [ms\(^{-1}\)] |
| \( y_3 \): Pitch angle [degrees] |

The model has state-space matrices:

\[
A = \begin{bmatrix}
0 & 0 & 1.132 & 0 & -1 \\
0 & -0.0538 & -0.1712 & 0 & 0.0705 \\
0 & 0 & 0 & 1 & 0 \\
0 & 0.0485 & 0 & -0.8556 & -1.013 \\
0 & -0.2909 & 0 & 1.0532 & -0.6859 \\
\end{bmatrix}, \quad B_u = \begin{bmatrix}
0 & 0 & 0 \\
-0.12 & 1 & 0 \\
0 & 0 & 0 \\
4.419 & 0 & -1.665 \\
1.575 & 0 & -0.0732 \\
\end{bmatrix}
\]

\[C = [I_3 \ 0] \quad D_u = 0_{3 \times 3}\]

Assume additive sensor-faults (denoted \( f_1, f_2, \) and \( f_3 \)), actuator-faults (denoted \( f_4, f_5, \) and \( f_6 \)). Also, assume that the process is influenced by additive white noise, both in the dynamic and measurement equations. The total model, including fault models then becomes:

\[
\dot{x} = Ax + Bu + Bf + Bn \\
y = Cx + Du + Df + Dn
\]

The noise is assumed white with unit covariance.

The design goal in all the three examples based on this model are a residual generator \( Q(s) \) that decouples faults in the elevator angle actuator, i.e. \( f_6 \), and produces a white residual in the fault-free case. The difference in the designs are different noise assumptions.

6.1.1 Process and measurement noise

In this first example, both measurement noise and process noise is considered and state-space matrices \( B_n \) and \( D_n \) is set to

\[
B_n = [I_5 \ 0_{5 \times 3}] \quad D_n = [0_{3 \times 5} \ I_3]
\]

\(2\)The spectral factorization procedure used is (Kwakernaak, 2000a) which is downloadable from http://www.polyx.com. The spectral factorization command included in the toolbox is also possible to use.
Calculations in MATLAB give

\[
N_M(s) = \begin{bmatrix}
0.0496s & 0.703s + 0.0378 & \cdots \\
0.421s^2 + 0.27s & -0.123 & \cdots \\
\cdots & 0.0643 & 0.0844 - 0.703 & 0 \\
\cdots & 0.0185s^2 - 0.0174s - 0.306 & 0.582 & 0 \\
\end{bmatrix}
\] (17)

This gives that the dimension of the null-space \(N_L(M(s))\) is 2, i.e. there exists two linearly independent numerators that decouples \(f_0\).

Step 2 from Section 5.1 was to compute matrix \(Z(s)\) and checking full-rank condition. Matrix \(Z(s)\) is shown to be

\[
Z(s) = \begin{bmatrix}
-0.5s^2 + 0.5 & 0.021s^3 - 0.012s^2 - 0.11s - 0.011 \\
-0.021s^3 - 0.012s^2 + 0.11s - 0.011 & 0.18s^4 - 0.26s^2 + 0.57
\end{bmatrix}
\]

which has full rank. Performing a J-spectral cofactorization gives:

\[
P(s) = \begin{bmatrix}
-0.59s - 0.61 & -0.39s - 0.37 \\
-0.26s^2 - 0.57s - 0.38 & 0.33s^2 + 0.75s + 0.66
\end{bmatrix}
\]

\(J = I_2\)

The spectral factor \(P(s)\) is strictly stable which can be seen by calculating the zeros of the invariant polynomials. Calculating the zeros in Matlab gives \(s = -1.0196\) and \(s = -1.1124 \pm j0.7305\).

Checking for existence of innovations filter according to Theorem 11 gives

row-deg \(N_{M_e}(s) = \{1, 2\}\)

row-deg \(N_{M_e}(s) \begin{bmatrix} D_n \\ B_n \end{bmatrix} = \{1, 2\}\)

i.e. a innovations filter exists and can be formed as \(Q(s) = P^{-1}(s)N_{M_e}(s)P_x\). Next, find an \(\eta(s)\) from Section 5.1.2 giving a scalar whitening residual generator achieving unit variance in the fault-free residual. One choice of \(\eta(s)\) is:

\[
\eta(s) = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}
\]

and the whitening residual generator can be formed as

\[
Q(s) = \eta(s)P^{-1}(s)N_{M_e}(s)P_x
\]

which is a 3rd order realizable and strictly stable residual generator. Figure 1 shows how the faults influence the residual and Figure 2 shows the fault-free spectrum \(\Phi_r(j\omega)\) which is 1 for all \(\omega\) as expected.

### 6.1.2 Only process noise

In this second example, only process noise is considered and state-space matrices \(B_n\) and \(D_n\) is set to:

\[B_n = I_5 \quad D_n = 0_{3 \times 5}\]

The null-space basis \(N_{M_e}(s)\) is identical as in the first example (17). The row-degrees of \(N_{M_e}(s)\) is \(\{1, 2\}\) and the row-degrees of \(N_{M_e}(s) \begin{bmatrix} D_n \\ B_n \end{bmatrix} = \{0, 1\}\), i.e. no innovations filter exists. Now, it will be shown that no whitening residual generator exists either.

Performing the spectral factorization gives a spectral co-factor:

\[
P(s) = \begin{bmatrix}
-0.70489 & 0.018563 \\
-0.018545s - 0.0011086 & 0.421333s + 0.6798
\end{bmatrix}
\]
Figure 1: Magnitude bode plots for the faults to the residual.

Figure 2: Spectrum $\Phi_r(j\omega)$. 
Matrix $W(s)$ from (13) then becomes
\[
W(s) = \begin{bmatrix}
-0.057s + 0.12 & 1s + 0.045 & -0.0056s + 0.11 & 0.12 & -1 & 0 \\
-1s + 0.96 & -0.057s - 0.071 & -0.044s + 0.11 & -0.0068 & 0.057 & 0
\end{bmatrix}
\]

It is then easy to verify that no proper residual generator exists by checking that $\mathcal{W}$ has full row-rank, i.e. $\mathcal{N}_L([R(s) \ W]) = \emptyset$.

### 6.1.3 Noise on all states and sensor 3

In this case the process is subjected to noise on all states and on sensor 3, i.e. the matrices $B_n$ and $D_n$ is given by
\[
B_n = [I_5 \ 0] \quad D_n = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1
\end{bmatrix}
\]

Now, computing $N_{M_x}(s)$ and $Z(s)$ as before gives that $Z(s)$ is strictly stable and that the row-degrees of $N_{M_x}(s)$ is (as before) $\{1, 2\}$ and row-degrees of $N_{M_x}(s) \begin{bmatrix} D_n \\ B_n \end{bmatrix}$ is $\{0, 2\}$. This gives that no innovations filter exists, but since the row-degree of the second row of $N_{M_x}(s)$ does not decrease when multiplied by the noise distribution matrices this indicates that there should exist a proper whitening residual generator. And sure it does, computing $W(s)$ gives that
\[
\mathcal{W} = \begin{bmatrix}
-0.14 & -2 & 0 & 0 & 0 \\
-0.0043 & -0.061 & 0 & 0 & 0
\end{bmatrix} \quad \text{rank } \mathcal{W} = 1
\]

Since $Z(s)$ is strictly stable, $R(s)$ from Theorem 8 is 0, and a whitening residual generator can be formed by
\[
Q(s) = \eta P^{-1}(s) N_{M_x}(s) P_x
\]
where $\eta$ is a constant row-vector in the (proven non-empty) null-space of $\mathcal{W}$. Figures 3 and 4 shows the fault influence on the residual and the spectrum of the fault-free residual. Here it is clear that the residual still is able to detect all faults even though an innovations filter didn’t exist. It is straightforward to realize that innovations filters preserve any fault detectability properties since $\varphi(s)$ is invertible. Whitening residual generators however may, if an innovations filter does not exist, lose desirable properties due to that $\varphi(s)$ is constrained to a subspace according to Theorem 8.

![Figure 3: Magnitude bode plots for the faults to the residual.](image-url)
6.2 Example with purely imaginary zeros

To study how purely imaginary zeros of $Z(s)$ influences the design procedure, consider the following system:

$$y = \left[ \frac{1}{s+1} \right] (u + f) + \left[ \begin{array}{c} 1 \\ 1 \\ 3 \\ 4 \end{array} \right] n$$

Simple calculations by hand gives that no innovations filter exists and all whitening residual generators can be written as

$$Q(s) = v(s) \frac{1}{\sqrt{5}} (y_1 - \frac{1}{s+1} u)$$

(18)

where $v(s)$ is any scalar all-pass link. Now, lets see how the design algorithm outlined in Section 5.1 arrives at the same conclusion.

Forming $M_x(s)$, computing $N_{M_x}(s)$ and $Z(s)$ gives

$$Z(s) = \begin{bmatrix} -2s^2 + 2 & -4.0166s^2 - 5.8424s - 1.8257 \\ -4.0166s^2 + 5.8424s - 1.8257 & -8.3333s^2 + 1.6667 \end{bmatrix}$$

which has zeros $\{0, 0, \pm 1\}$. This gives that $Z(s)$ has zeros on the imaginary axis and $p_{\text{imag}}(s) = s$.

The matrix $R(s)$ is then computed by left division of adj $P(s)$ with $p_{\text{imag}}(s)$

$$R(s) = \begin{bmatrix} -1.2 & -1.4 \\ 0.36 & 0.4 \end{bmatrix}$$

Matrix $\eta(s)$ is then computed according to step 4 in 5.1 with $\Psi = 1$. This gives

$$Q(s) = \begin{bmatrix} 0.4472 & 0 \\ -0.4472 \end{bmatrix} \frac{1}{s+1}$$

which, since $1/\sqrt{5} \approx 0.4472$ is identical to (18).

7 White residuals

Why is white residuals desirable and are there advantages with white residuals compared to low-pass filtered residuals to be thresholded?

One reason for white residuals is that basic change detection algorithms, e.g. cusum (Basseville and Nikiforov, 1993), is directly applicable. Although these algorithms are optimal, the algorithm relies on deep knowledge of amplitude distribution of the residual and also the type of change (step/ramp/etc.). Often, such knowledge is not available and a perhaps more important advantage with white residuals is that the simple thresholding test for testing if the residual is small can be replaced by a whiteness test of the residual.
The next section describes a whiteness test and Section 7.2 shows a few simulation results where a whiteness test based detection is compared with a low-pass and threshold approach. The simulation study does not provide any hard theoretical results on the merits of either method, but still indicates some properties.

7.1 Whiteness tests

A whiteness test is a test to decide between the hypotheses

\[ H_0 : n(t) \text{ is white} \quad H_1 : n(t) \text{ is coloured} \]  \hspace{1cm} (19)

where \( n(t) \) is the signal being analyzed. This section is mainly based on (Söderström and Stoica, 1989) and (Mehra and Peschon, 1971).

Typically, a test quantity to separate \( H_0 \) from \( H_1 \) is based on an estimate of the covariance function \( c_k \). A basic property of white noise processes is that \( c_k = \sigma^2 \) if \( k = 0 \) and \( c_k = 0 \) for \( k \neq 0 \). The covariance function of \( n(t) \) is defined (for zero-mean stationary stochastic processes),

\[ c_k = E\{n(t)n(t-k)\} \quad k = 0, 1, \ldots \]

and an estimate of \( c_k \) can be obtained from data as

\[ \hat{c}_k = \frac{1}{N} \sum_{t=1}^{N} n(t)n(t+k) \quad k = 0, 1, \ldots \]  \hspace{1cm} (20)

Let \( \hat{c} \) denote the vector

\[ \hat{c} = \begin{pmatrix} \hat{c}_1 \\ \hat{c}_2 \\ \vdots \\ \hat{c}_{n_k} \end{pmatrix}, \]

then it holds that (asymptotically)

\[ \frac{N\hat{c}^T\hat{c}}{c_0} \sim \chi^2(n_k) \]  \hspace{1cm} (21)

Thus, a test quantity for hypotheses (19) can be formed by estimating \( c_0 \) to \( c_{n_k} \) from \( M \) data points and calculate

\[ T(x) = \frac{M\hat{c}^T\hat{c}}{c_0} \]  \hspace{1cm} (22)

The null hypothesis is rejected when \( T(x) > J \). The thresholds are selected such that the false alarm rate is lower than a specified level \( \alpha \). The threshold selection is done by assuming asymptotic properties (21) of \( T(x) \), i.e. assume \( M \) large enough.

One nice property of the test \( T(x) \), assuming asymptotic properties, is that it is invariant to distribution and input noise power.

7.2 Simulations and Comparisons

In this section, a few simulations is done to compare the whiteness test from the previous section with a simple LP-and-threshold test. The thresholds in both tests is set to achieve a false-alarm rate of \( \alpha = 0.01 \).

In the simulations, the system under consideration is given by

\[ y = G_u(q)u + n + f \]

i.e. sensor noise and a sensor fault is considered. The “raw” residual is given by

\[ r_{raw} = y - G_u(q)u = n + f \]
In the simulations, \( n \) is white gaussian noise and \( y \) and \( u \) is collected during 100 seconds, sampled with sampling frequency 10 Hz. The residual is then evaluated, either with a white-noise test or a LP-and-threshold test.

The parameters in the whiteness is set to (without any excessive tuning) \( M = 80 \) and \( n_k = 20 \). In the LP-and-threshold test, a first order LP-link with unit DC-gain is used, i.e.

\[
H_{lp}(z) = \frac{1-a}{z-a}
\]

The parameter in the test quantity is therefore the placement of the pole in the filter. The threshold is set by (correctly) assuming gaussian distribution and unit variance. In the simulations, four different pole placements with different cut-off frequencies are used, \( a \in \{0.9,0.95,0.97,0.99\} \).

Three kinds of faults are simulated

1. Step fault. The fault signal is

\[
f(t) = \begin{cases} 
0 & t < 50 \\
0.85 & t \geq 50 
\end{cases}
\]

The simulation results is shown in Figures 5, 6, and 7.

2. Sinus fault. The fault signal is

\[
f(t) = \begin{cases} 
0 & t < 50 \\
1.5 \sin(\pi t) & t \geq 50 
\end{cases}
\]

The simulation results is shown in Figures 8, 9, and 10.

3. Ramp fault. The fault signal is

\[
f(t) = \begin{cases} 
0 & t < 50 \\
\frac{1.5}{50} (t - 50) & t \geq 50 
\end{cases}
\]

The simulation results is shown in Figures 11, 12, and 13.

### 7.3 Simulation conclusions

From this simulation study it is not possible to draw any hard theoretical conclusions regarding the effectiveness of a whitening test compared to a low-pass and threshold test. The simulation study does however indicate that, in the case studied, the whiteness test is generally better to handle a wide variety of fault signal characteristics than the lp-and-threshold approach.

### 8 Time-discrete systems

In the deterministic case, the time-discrete systems and time-continuous systems could be handled identically by just replacing \( s \) with \( z \) and proper with causal (Frisk and Nyberg, 2001). In the stochastic case, small but important differences exists which are briefly discussed below.

First, a few words on the discrete time spectral factorization problem. Let \( r = H(z)n \) with \( n \) unit covariance white noise, then the spectrum of \( r \) is given by

\[
\Phi_r(\omega) = H(e^{j\omega T})H^T(e^{-j\omega T})
\]

This indicates that \( A(z) \) is parahermitian if \( A(z) = A^T(z^{-1}) \).

Thus, a time-discrete version of (7) becomes:

\[
Z(z) = N_{M_x}(z) \begin{bmatrix} D_n \\ B_n \end{bmatrix} \begin{bmatrix} D_n \\ B_n \end{bmatrix}^T N_{M_x}^T(z^{-1})
\]
Figure 5: The raw residual and fault signal in the step-fault simulation.

Figure 6: Test quantity and the decision, i.e. thresholded test quantity, for the whiteness based test in the step-fault simulation.

Figure 7: Residuals and decisions for the LP-and-threshold test in the step-fault simulation.
Figure 8: The raw residual and fault signal in the sinus-fault simulation.

Figure 9: Test quantity and the decision, i.e. thresholded test quantity, for the whiteness based test in the sinus-fault simulation.

Figure 10: Residuals and decisions for the LP-and-threshold test in the sinus-fault simulation.
Figure 11: The raw residual and fault signal in the ramp-fault simulation.

Figure 12: Test quantity and the decision, i.e. thresholded test quantity, for the whiteness based test in the ramp-fault simulation.

Figure 13: Residuals and decisions for the LP-and-threshold test in the ramp-fault simulation.
and $Z(z)$ is para-hermitian. A J-spectral co-factorization of $Z(z)$ then becomes

$$Z(z) = P(z)JP^T(z^{-1})$$

where $P(z)$ have all zeros inside and on the unit circle. However, here only factorizations with signature $J = I$ and $Z(z)$ full rank on the unit-circle is considered. This because, to the authors knowledge, no numerically reliable factorization algorithms exists for the general case. In Polynomial Toolbox 2.0 for Matlab 5 (1998), an algorithm described in (Ježek and Kučera, 1985) is used.

The main difference between the time-continuous and time-discrete cases is that properness of the residual generator can always be achieved. This is immediate since a non-proper (non-causal) filter can always be made proper (casual) by inserting a number of time-delays and since $z^{-1}$ is an all-pass link, the whiteness property is not destroyed. Thus, it is immediate to prove that the existance conditions for full-rank innovations filters and whitening residual generators is identical to the time-continuous case where the properness condition has been removed, i.e.:

**Theorem 12.** If $Z(z)$ is full rank, an innovation filter exists if and only if $Z(z)$ has no roots on the imaginary axis.

**Theorem 13.** If $Z(z)$ is full-rank, there exist an $Q(z)$ such that $Q(z) = \Phi_r(z)N_L(R(z))P_x$ is proper and strictly stable if and only if

$$N_L(R(z)) \neq \emptyset$$

Therefore, for the innovations filter design, the algorithmic restriction to $Z(z)$ with no zeros on the imaginary axis is no restriction. However in the whitening residual generator case this is a restriction as illustrated by the following small example which shows how the continuous algorithm directly transfers to the time-discrete case.

$$y = \begin{bmatrix} \frac{1}{z-a} \\ \frac{1}{z-1}(z-a) \end{bmatrix} (u + f) + n$$

It is clear that a residual generator $r = y_1 - \frac{1}{z-a}u$ produces white residuals in the fault-free case, an innovations filter does however not exist. Straightforward calculations give that for any residual generator it holds that the transfer function from noise to residual is given by

$$r = \varphi(z) \begin{bmatrix} z-a & 0 \\ 0 & 1-z \end{bmatrix} n$$

This gives that $\Phi_r(z)$ is given by

$$\Phi_r(z) = \varphi(z)Z(z)\varphi^T(1/z) = \varphi(z)P(z)P^T(1/z)\varphi^T(1/z)$$

with $P(z) = \begin{bmatrix} z-a & 0 \\ 0 & z-1 \end{bmatrix}$

$$P^{-1}(z) = \frac{1}{z-a} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \frac{1}{(z-1)(z-a)} \begin{bmatrix} 0 & 0 \\ 0 & 1-a \end{bmatrix}$$

which gives that any

$$Q(z) = \eta(z)P^{-1}(z)N_M(z)P_x \quad \eta(z) \in N_L(R(z)) \quad \eta(z)\eta(1/z) = 1$$

One such choice is $\eta(z) = [-1 \ 0]$ which gives the residual generator

$$r = y_1 - \frac{1}{z-a}u$$

i.e. same as (23). However, to the authors knowledge, no computational tools seems to be available that can handle general cases with zeros on the unit circle.
9 The singular case

This section deals with the singular case. A discussion on when the singular case occurs and what the difficulties are and why this report doesn’t include a design algorithm also applicable to the singular case.

9.1 Singular complications

Now follows a singular version of Lemma 7. After that, a discussion on the difficulties of formulating singular versions of Theorem 8 and Theorem 11.

Lemma 14. There exists a \( \varphi(s) \) such that the linear time-invariant filter \( Q(s) = \varphi(s)N_{M_2}(s)P_x \) produces white residuals if and only if \( \varphi(s) \) can be written

\[
\varphi(s) = [\eta(s) \quad \zeta(s)]P^{-1}(s)
\]

where \( P(s) \) is a spectral co-factor of \( Z(s) \), \( \eta(s)\eta^T(-s) = \Psi \), and \([\eta(s) \quad \zeta(s)]\) is partitioned according to the signature of the spectral co-factorization.

Proof. According to the proof of Theorem 1, the spectrum of \( \varphi(s) \) results from \( \lambda_1 = \eta(s)P_1(s)^{-1} \eta(s) \) and \( \lambda_2 = \zeta(s)P_1(s)^{-1}\zeta(s) \) insertions into (24) and denoting \( \eta(s)\eta^T(-s) = \Psi \), the parameterization matrix \( \varphi(s) \) is found by solving for \( \varphi(s) \) in the equation

\[
\eta(s) = \varphi(s)P_1(s)
\]

which also gives that \( \Phi_r(s) = \Psi \) according to the assumptions in the Theorem. The parameterization matrix \( \varphi(s) \) is found by solving for \( \varphi(s) \) in the equation

\[
\varphi(s) = \eta(s)P_1(s)^{-1}\eta(s) + \zeta(s)N_{P_1}(s)
\]

where \( P_1^{-1}(s) \) is any left-inverse of matrix \( P_1(s) \) and \( N_{P_1}(s) \) is a basis for the left null-space of \( P_1(s) \). The existence of such a left-inverse is ensured by the full column-rank property of \( P_1(s) \). A stable inverse can be found by the inverse of \( P(s) \). Let

\[
P(s)^{-1} = \begin{bmatrix} P_{11}(s) \\ P_{21}(s) \end{bmatrix}
\]

then it holds that \( P_{1i}(s) \) is a stable left inverse of \( P_1(s) \) and \( P_{2i} \) is a basis for the null-space of \( P_1(s) \). Thus, in the rank deficient case, all \( \varphi(s) \) satisfying (26) can be written

\[
\varphi(s) = [\eta(s) \quad \zeta(s)]P^{-1}(s)
\]
**Remark 1:** An example of an inverse $P^{-1}(s)$ from the proof is the Moore-Penrose inverse. However, even though $P(s)$ is Hurwitz, the Moore-Penrose inverse of $P_1(s)$ is not necessarily stable.

**Remark 2:** One might argue that the parameterization from the theorem introduces unnecessary poles in $\phi(s)$ since whole $P(s)$ is inverted, not only $P_1(s)$ as needed. However, as shown in Section 3.1, the spectral factor $P(s)$ can always be written on the form

\[ P(s) = U(s) \begin{bmatrix} \hat{P}(s) & 0 \\ 0 & I \end{bmatrix} \]

where $P(s)$ is partitioned according to the signature $J$, matrix $U(s)$ is unimodular, and $\hat{P}(s)$ is a spectral factor of the non-singular part. Thus, all poles in $P^{-1}(s)$ origins from the non-singular part of the spectral factorization.

Now, existence conditions of innovations filters and whitening residual generators would be performed like in Section 5. Such conditions that are easily computable is difficult to state mainly because

1. The spectral factorization of a singular para-hermitian matrix is not canonical, i.e. we can not assume generic row-degrees of a spectral factor as in Theorem 5. To the authors knowledge, no canonical spectral factorization exists for the singular case which is also stated in the conclusions of (Sebek, 1990).

Thus, easily computable results like in Corollary 10 and Theorem 11 is not directly available.

2. The freedom included in the design with $\zeta(s)$ from Lemma 14 to ensure properness and stability of the residual generator is not yet fully understood.

### 9.2 When does the non-singular case occur?

Now, a result giving necessary condition for the problem to be non-singular and the results from Section 5 is directly applicable.

**Theorem 15.** A sufficient condition for $Z(s)$ to be non-singular is

\[ \text{rank} \begin{bmatrix} D_n \\ B_n \end{bmatrix} > k_y - \text{rank} \begin{bmatrix} D_d \\ B_d \end{bmatrix} \]

**Proof.** If $Z(s)$ is non-singular case it is possible to achieve full decoupling of the noise $n$. This is evident by following the proof of Lemma 14 and letting $\eta(s) = 0$ and $\zeta(s) \neq 0$ which will give a residual generator (always possible to get proper and stable) such that $n$ is decoupled in the residual.

Following the proof of Theorem 1 it is seen that this is achievable if and only if the left null-space of

\[ M(s) = \begin{bmatrix} C & D_d & D_n \\ -(sI - A) & B_d & B_n \end{bmatrix} \]

is non-empty. A dimensionality analysis of $M(s)$ immediately gives the theorem. ■

Then, the following corollary follows immediately

**Corollary 16.** Let all sensors in process (1) be subjected to measurement noise, then $Z(s)$ is non-singular.

### 10 Conclusions

This report describes a polynomial design algorithm for linear residual generation for stochastic systems in both continuous and discrete time. The problem formulated is based on innovations filters formulated by Nikoukhah (1994). The problem formulation is then also further developed to whitening residual generators.
The two main steps in the design algorithm is extraction of a polynomial basis for the left null-space of a polynomial matrix followed by a J-spectral co-factorization of a para-hermitian polynomial matrix. For both these operations there exists good numerical tools. The design algorithm is successfully demonstrated on a number of non-trivial examples. Full Matlab implementations is also provided.

References


M. Nyberg and E. Frisk. A minimal polynomial basis solution to residual generation for fault diagnosis in linear systems. IFAC, Beijing, China, 1999.


### A Matlab functions

#### A.1 innovationsfilter.m

```matlab
function Q=innovationsfilter(A,Bu,Bd, Bn,C,Du,Dd,Dn)
%INNOVATIONSFILTER - Design an innovationsfilter
% Given a fault-free system description
% x'= Ax + Bu u + Bd d + Bn n
% y = Cx + Du u + Dd d + Dn n
% Design (if one exists) an innovationsfilter Q(s).
% Any innovations filter W(s) can then be parameterized by
% W(s) = L(s)Q(s)
% where L(s) is invertible and satisfies L(s)L'(s)=I
% Syntax: Q=innovationsfilter(A,Bu,Bd, Bn,C,Du,Dd,Dn)

tol=1e-5; % Tolerance for detecting purely imaginary zeros
nx = length(A); % Number of states
nmeas = size(C,1); % Number of measurements
ndist = size(Bd,2); % Number of disturbances
Mx = [C Dd;-(s*eye(nx)-A) Bd];
Px = [eye(nmeas) -Du;zeros(nx,nmeas) -Bu];
Nmx = null(Mx.').'.';
Z = Nmx*[Dn;Bn]*[Dn;Bn]'*Nmx';
if rank(Z)<size(Z,1)
    error('Singular cases is not covered (yet) by this function');
end
if (max(deg(Nmx,'row')-deg(Nmx*[Dn;Bn],'.row'))>0) | ...
    (sum(abs(real(roots(Z)))<tol)>0)
    error('No innovations filter exists');
end
[P,J] = spfc((Z.');'); P = P.'; % Spectral factorization
```

27
[Qa,Qb,Qc,Qd] = lmf2ss(Nmx*Px,P);
Q = minreal(ss(Qa,Qb,Qc,Qd));

A.2 whiteresgen.m

function [Q,N]=whiteresgen(A,Bu,Bd, Bn,C,Du,Dd,Dn,any)
%whiteresgen - Design whitening residual generators
%
% Given a fault-free system description
% x' = Ax + Bu u + Bd d + Bn n
% y = Cx + Du u + Dd d + Dn n
%
% Design (if one exists) whitening residual generators Q(s).
% All whitening residual generators W(s) can be parameterized by
% W(s) = L(s)Q(s)
% where L(s) is in row-image N(s) and satisfies L(s)L'(s)=1
%
% Syntax: [Q,N]=whiteresgen(A,Bu,Bd,Bn,C,Du,Dd,Dn[,any])
% With 'any=1' the result is one strictly stable
% whitening residual generator Q(s) instead of a
% parameterization.

tol=1e-5; % Tolerance for detecting purely imaginary zeros

nx = length(A); % Number of states
nmeas = size(C,1); % Number of measurements
ndist = size(Bd,2); % Number of disturbances

Mx = [C Bd;-(s*eye(nx)-A) Bd];
Px = [eye(nmeas) -Du;zeros(nx,nmeas) -Bu];
Nmx = null(Mx.');.';

Z = Nmx*[Dn;Bn]*[Dn;Bn]'*Nmx';

if rank(Z)<size(Z,1)
    error('Singular cases is not covered (yet) by this function');
end

[P,J] = spfc((Z')..');; P = P.'; % Spectral factorization
Pzeros = roots(det(P));

if sum(abs(real(Pzeros))<tol)>0
    imagIdx = find(abs(real(Pzeros))<tol);
    stabIdx = setdiff([1:length(Pzeros)],imagIdx);
    pimag = prod(s-Pzeros(imagIdx));
    pstab = prod(s-Pzeros(stabIdx));
    k = lcoef(det(P)); % det(P) = k*pimag*pstab
    [V,R] = ldiv(adj(P),pimag);
    [Qa,Qb,Qc,Qd] = lmf2ss(V*Nmx*Px,k*pstab*eye(size(V,1)));
else
    R=0;
    [Qa,Qb,Qc,Qd] = lmf2ss(Nmx*Px,P);
end

if isa(Qd,'pol')
    W = Qd(1:deg(Qd));
end
Qd = Qd(0);
else
    W = zeros(size(Qd));
end
N = null([R W_='']);
if isempty(N)
    error('No whitening residual generator exist');
end
Q = ss(Qa, Qb, Qc, Qd);

if any
    n = ones(1, size(N, 1)) * N; % Example choice of n(s)
    if deg(n) < 1 % A constant n(s) is found
        n = n(0);
        Q = ss(Qa, Qb, n * Qc, n * Qd);
    else
        d = spf(n * n');
        [etaa, etab, etac, etad] = lmf2ss(n, d);
        Q = ss(eta, etab, etac, etad) * ss(Qa, Qb, Qc, Qd);
    end
end
Q = minreal(Q);