RESIDUAL GENERATOR DESIGN FOR NON-LINEAR, POLYNOMIAL SYSTEMS – A GRÖBNER BASIS APPROACH

Erik Frisk

Department of Electrical Engineering
Linköping University
Sweden
email:frisk@isy.liu.se

Abstract: Design and analysis of residual generators for polynomial systems is considered. This paper presents a systematic procedure, given an input-output description of system dynamics, to design residual generators for fault diagnosis. The design procedure is based on standard elimination theory. The design procedure is applied in a simulation study on a non-linear system, where it is showed how multiplicative and additive faults are detected and isolated. The example also shows how a fault detectability/isolability analysis can be made during the design.

Keywords: diagnosis, non-linear systems, parity functions, polynomial systems, elimination theory, water tanks

1. INTRODUCTION

Residual generation is a fundamental part of a diagnosis system. The residual generator filters known signals and generate a signal, a *residual*, that should be small (ideally 0) in the fault-free case and large when a fault is acting on the system.

A common approach is to design non-linear, unknown input observers, based on a state-space description of process dynamics. For linear systems, consistency relations/parity functions, instead of state-space descriptions, has successfully been used in the design and analysis (Gertler 1991, Nyberg & Frisk 1999, Nyberg 2000) of residual generators. Consistency relations has also been considered in the non-linear case (Krishnaswami, Luh & Rizzoni 1994, Krishnaswami & Rizzoni 1994, Guernez, Casar & Staroswiecki 1997, Zhirabok 1999). Section 4 presents a systematic design method for residual generators based on non-linear consistency relations of polynomial systems. The methodology has strong similarities with finding input-output descriptions of non-linear systems (Jirstrand 1998). Basic tools from standard elimination theory, needed for the design procedure, is introduced in Section 3. In Section 6, the proposed design procedure is applied to a non-linear model and it is shown by example how both multiplicative and additive faults are detected and isolated.

2. PROBLEM FORMULATION

One general principle for model based diagnosis is to detect and isolate faults by checking consistency between observed behavior and expected behavior for different fault situations. To perform this, signals called residuals, can be generated that indicates if a consistency relation is violated or not. A residual that is 0 or small in the fault-free case and deviates significantly from 0 when a fault occurs can be used for fault *detection*. By designing a set of such residuals, where a different set of faults are decoupled in each residual, fault detection and *isolation* is possible. This is the main idea of the well known *structured residuals* concept.

The model is assumed to be a set of differential equations:

$$g_i(\bar{y}, \bar{u}, \bar{d}, \bar{f}, \bar{x}) = 0 \qquad i = 1, \ldots, p \tag{1}$$

where u is the control vector, y the measurement vector, d disturbance vector, f the fault vector, and x unknown internal states. The notation $\bar{y}$ denotes y, $\dot{y}$, $\ddot{y}$, ... and $\bar{u}$, $\bar{d}$, $\bar{f}$, $\bar{x}$ corresponding for u, d, f, and x.

Here, only polynomial non-linearities are considered, i.e. the g_i in (1) are polynomials in $\bar{y}$, $\bar{u}$, $\bar{d}$, $\bar{f}$, and $\bar{x}$. This restriction is not as restrictive as it may seem, many non-polynomial non-linearities can be rewritten in polynomial form, e.g.

$$y = \sin u \quad \Leftrightarrow \quad \dot{y}^2 - \dot{u}^2(1-y^2) = 0$$

Therefore, quite general systems can be handled within the framework of polynomial systems. Also, any non-linearity can be described by polynomials to an arbitrary accuracy by Taylor expansion.

This model-class restriction is made to be able to form a systematic design procedure based on established elimination theory, i.e. modeling accuracy is traded against systematic design and analysis procedures. Such trade-offs is common, e.g. when using linear models in the design of controllers/residual generators to be able to use the powerful design tools available for linear systems.

The general design problem is then to find a filter that filters known signals and produces a residual where disturbances and a subset of the faults are decoupled in the residual. In the following, all faults that are to be decoupled and any modeled disturbances are collected in a vector d, and the faults we wish to detect are collected in a vector f.

The fundamental design problem studied here can then be stated formally as

Basic Design Problem:

> *Given a set of model equations* (1)*, find a computable quantity r that is a function of $\bar{y}$ and $\bar{u}$ only, such that when $\bar{f} = 0$ it holds that*
> $$r(\bar{y}, \bar{u}) = 0 \quad \forall \bar{d}$$

Of course, to be able to detect faults, it must also hold that $r(\bar{y}, \bar{u}) \neq 0$ for some $\bar{f} \neq 0$. The expression $r(\bar{y}, \bar{u})$ is called the computational form of the residual generator. The expression describing the fault response in the residual is called the internal form of the residual generator.

3. BASIC ELIMINATION THEORY

Some notation and a theorem from basic elimination theory (Cox, Little & O'Shea 1991) is needed to describe the design procedure. Let $k[x_1, \ldots, x_n]$ denote the set of polynomials in variables $x_1, \ldots, x_n$ with coefficients in k, e.g.

$$x_1x_2 + x_3^3 - 2x_1^2x_2x_3 \in k[x_1, x_2, x_3]$$

An important concept, *ideal*, is now defined:

Definition 1. (Ideal). Let $g_1, \ldots, g_s$ be polynomials in $k[x_1, \ldots, x_n]$. Then denote

$$I = < g_1, \ldots, g_s > = \left\{ \sum_{i=1}^{s} h_i g_i : h_i \in k[x_1, \ldots, x_n] \right\}$$

I is called the *ideal* generated by the polynomials $g_1, \ldots, g_s$.

This means that I is the set of all linear combinations of the polynomials g_i with polynomial coefficients h_i.

The main theorem used in the design is the well known elimination theorem:

Theorem 1. (Elimination Theorem).
Let $I \subset k[x_1, \ldots, x_n]$ be an ideal and let G be a Gröbner basis of I with respect to lex order $x_1 \succ x_2 \succ \cdots \succ x_n$. Then, for every $0 \leq k \leq n$, the set

$$G_k = G \cap k[x_{k+1}, \ldots, x_n]$$

is a Gröbner basis of the $k:th$ elimination ideal I_k

$$I_k = I \cap k[x_{k+1}, \ldots, x_n]$$

This means that all polynomials, where variables $x_1, \ldots, x_k$ has been eliminated, can be written as in Definition 1 where $g_1, \ldots, g_s$ are the polynomials in G_k.

4. DESIGN USING ELIMINATION THEORY

As discussed in Section 2, the model is assumed to be a set of polynomial equations on the form:

$$g_i(\bar{y}, \bar{u}, \bar{d}, \bar{f}, \bar{x}) = 0 \qquad i = 1, \ldots, p \tag{2}$$

The basic step in the design algorithm is to manipulate the model equations (2) such that a consistency relation is obtained where all disturbances $\bar{d}$ (including faults that are to be decoupled) and internal states $\bar{x}$ has been eliminated. This relation can then be used to form a residual generator where the computational form is a function of $\bar{y}$ and $\bar{u}$ only and the internal form is a function of $\bar{f}$ only. Calculating a Gröbner basis and using Theorem 1 provides a systematic procedure to perform such manipulations. All derivations in this section are made in the time-continuous case. However, corresponding results for the time-discrete case is immediate by exchanging the time differentiation operator with the time shift operator. More comments on the time-discrete case in Section 5.

The first step in the design procedure is to calculate the (reduced) Gröbner basis for the elimination ideal where $\bar{d}, \bar{x}$ has been eliminated. Denote this basis with

$$GB = < b_1, \ldots, b_r >$$

where b_i are polynomials in all variables but $\bar{d}$. This Gröbner basis GB means that any polynomial, analytical relation, inferred from the model equations (2) *without* using the differentiation operation, where the disturbances $\bar{d}$ are eliminated can be written as

$$\sum_{i=1}^{r} h_i b_i \tag{3}$$

for a set of polynomials h_i. Note that Gröbner bases is a *non-differential* tool, i.e. the differentiation operator is not considered. Of course, the differentiation operator is essential when analyzing dynamic systems. To be able to use non-differential tools like Gröbner bases, the differentiation has to be done "by-hand", i.e. differentiate the model equations to get new model equations. Therefore, when using the method outlined here, one is only considering residual generators and consistency relations up to a certain order, equal to the number of times the model equations were differentiated.

Another approach is to use differential tools like Ritt's characteristic sets (Ritt 1950) to perform the elimination. However, the differential theory is currently not as developed as the non-differential and differential bases is an active research topic.

Each of the r polynomials in GB, or any combination as in (3), can be used to form a residual generator where the computational and internal forms are given by

$$r^{comp}(\bar{y},\bar{u}) = \sum_{i=1}^{r} h_i b_i \big|_{f_i=0} \tag{4}$$

$$r^{int}(\bar{f},\bar{y},\bar{u}) = \sum_{i=1}^{r} h_i b_i - r^{comp} \tag{5}$$

The h_i polynomials are design variables available to the designer. These can be used e.g. to shape the fault response in the residual or select the residual structure, i.e. to get sensitivity in the residual to a desired set of faults. Note that, in general, the computational form r^{comp} includes higher order derivatives of y and u. This problem is further discussed in Section 5.

A design variable, apart from choosing h_i polynomials in (4), is the variable ordering when calculating the Gröbner basis. Different variable orderings highly influences the resulting basis. To the authors knowledge, no optimal way of ordering the variables exists. A heuristic to choose variable ordering can be stated as:

Heuristic 1. (Variable ordering).

(1) Order the variables to be decoupled as the largest variables, i.e. the faults to be decoupled and the disturbances should be largest.
(2) Order the faults not to be decoupled as the smallest variables.

The heuristic can be motivated as follows. According to Theorem 1, the variables to be eliminated should be ordered to be the largest set of variables. Other variables we wish to eliminate, e.g. time derivatives of output signals to get as low order consistency relations as possible, should therefore get a large variable ordering. For the same reason, to get good fault sensitivity, the fault variables f_i should be ordered lowest.

A nice property with the approach presented here is that there exists extensive support for the operations needed in symbolic algebra computing packages such as Mathematica and Maple. Design and analysis is easily automated and in the example in the next section, Mathematica was used extensively.

Another approach for the design of residual generators for non-linear systems is to design disturbance decoupling, non-linear observers. General disturbance decoupling in the observer approach is difficult and to the authors knowledge, no general method with strong computer support has been presented. In the approach presented here, decoupling is well covered. Stability issues is a major concern in observer based approaches and they are equally important in this, consistency relations based, approach. The stability issue will be addressed in the next section.

5. REALIZABLE RESIDUAL GENERATOR

Note that $\dot{y}$, $\dot{u}$ etc. normally appear in $r^{comp}(\bar{y},\bar{u})$. Normally, these are not known and a simple method to compute the residual, used in in Section 6, is to use a filter

$$\hat{\dot{y}} = \frac{s}{sT_d+1} y$$

with a suitably chosen T_d to estimate the derivatives. This may not be a satisfactory solution in many cases. In an off-line application or where fast detection time is not of importance, more time-consuming estimation algorithms can be used, e.g. fit splines to measured data and use the splines for derivative calculations.

Other possibilities than estimating the derivatives, not explored further here, are realization theory and time-discretization of the time-continuous model.

Realization theory

Add stable (possibly linear) low-pass dynamics, e.g.

$$r + \alpha_1 \dot{r} + \alpha_2 \ddot{r} = r^{comp}$$

and find an explicit state-space realization of the residual generator. This is very similar to the linear case where linear low-pass dynamics is added to make the residual generator realizable, see e.g. (Nyberg & Frisk 1999) among many. Note that the added dynamics is free to choose as long as it is stable. Unfortunately, realization theory with inputs is a difficult business, even when only polynomial systems is considered (Forsman 1991).

Transform the time continuous model to a time-discrete model

Transform the original, time-continuous model, to a time-discrete model. Then, time-derivatives of signals will be replaced by time delays which are known, i.e. the residual can be computed.

Realization of residual generators based on non-linear consistency relations is a promising topic where further study is needed.

6. SIMULATION EXAMPLE: COUPLED WATER TANKS

The model used to illustrate the approach is two coupled water tanks, shown in Figure 1. The process is equipped with four sensors, two sensors measuring the water level in each tank and two sensors measuring the outflow of water from each tank. The process is controlled by a pump.

6.1 *Modeling*

A first-principles model of the fault-free process, utilizing Bernoulli's law for the flows, is given by:

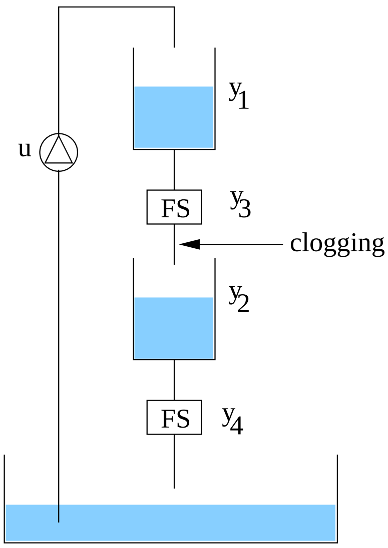

Fig. 1. The simulation process: Coupled water tanks equipped with flow and water-level sensors.

$$\begin{aligned}
\dot{h}_1 &= a_1 u - a_2\sqrt{h_1} \\
\dot{h}_2 &= a_3\sqrt{h_1} - a_4\sqrt{h_2} \\
y_1 &= h_1 \\
y_2 &= h_2 \\
y_3 &= a_5\sqrt{h_1} \\
y_4 &= a_6\sqrt{h_2}
\end{aligned} \tag{6}$$

where a_i are the model parameters, y_i the measurements, u the control signal to the pump, and h_i the height of water in each tank. The water level in the tanks can be between 0 and 10, i.e. $0 \leq h_1, h_2 \leq 10$.

The faults considered are faults in the actuator, sensors, and clogging in the pipe between the two water tanks at the point indicated by the arrow in Figure 1. Extending model (6) with straightforward fault models gives:

$$\begin{aligned}
\dot{h}_1 &= a_1(u + f_1) - a_2(1 - f_6)\sqrt{h_1} \\
\dot{h}_2 &= a_3(1 - f_6)\sqrt{h_1} - a_4\sqrt{h_2} \\
y_1 &= h_1 + f_2 \\
y_2 &= h_2 + f_3 \\
y_3 &= a_5(1 - f_6)\sqrt{h_1} + f_4 \\
y_4 &= a_6\sqrt{h_2} + f_5
\end{aligned} \tag{7}$$

where the signals $f_i, i = 1, \ldots, 6$ models the faults. Signal f_1 models an unknown additive fault on the actuator signal and f_2, f_3, f_4, f_5 additive faults on the four sensors. The clogging fault is modeled by f_6 where $f_6 = 1$ represents a completely clogged pipe and $0 < f_6 < 1$ represents partial clogging.

To make this example as short as possible, elimination of state-variables is made by hand by differentiating y_1 and y_2 and eliminate the state-variables. The model equations is then given by:

$$\begin{aligned}
\dot{y}_1 &= a_1(u + f_1) - a_2(1 - f_6)\sqrt{y_1 - f_2} \\
\dot{y}_2 &= a_3(1 - f_6)\sqrt{y_1 - f_2} - a_4\sqrt{y_2 - f_3} \\
y_3 &= a_5(1 - f_6)\sqrt{y_1 - f_2} + f_4 \\
y_4 &= a_6\sqrt{y_2 - f_3} + f_5
\end{aligned} \tag{8}$$

Here, only constant faults are considered, i.e. $\dot{f}_i = 0$. Note that this assumption is not required by the approach, it is only made here to limit the size of the example. Here, Taylor expansion of order 3 around operating point h_0 is used to approximate the square root expression with polynomials. The accuracy of the polynomial approximation can be seen in Figure 2. The point h_0, about which the square root Taylor expansions are made is selected to 5, i.e. in the middle of the operating range for both tanks. The model

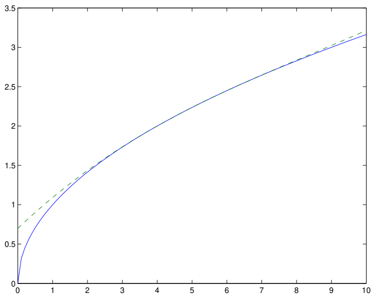

Fig. 2. Accuracy of the Taylor expansion. The solid line is $\sqrt{x}$ and the dotted line a third order polynomial approximation.

equations (8) can be stated on form (1) by moving the right hand side of each model equation in to the left side, and perform Taylor expansion of the square-root terms. The model equations is then given by:

$$g_i(u, y_1, \dot{y}_1, y_2, \dot{y}_2, y_3, y_4, f_1, \ldots, f_6) = 0 \qquad i = 1, \ldots, 4 \tag{9}$$

The first model equation g_1 becomes:

$$\begin{aligned}
\dot{y}_1 + \frac{1}{16\sqrt[5]{h_0}}\big(a_2(f_6 - 1)(f_2^3 - 5h_0^3 + \\
+ f_2^2(5h_0 - 3y_1) - 15h_0^2 y_1 + 5h_0 y_1^2 - y_1^3 + \\
+ f_2(15h_0^2 - 10h_0 y_1 + 3y_1^2))\big) - a_1(u + f_1) = 0
\end{aligned}$$

In the modeling done here, a polynomial model was achieved by Taylor expansion of non-polynomial parts of the model. Instead of Taylor expansion, the square root expressions could have been written in polynomial form by introducing new internal state-variables $z_i^2 = h_i$.

In addition, since the mathematical tool used is non-differential, the measurement equations was differentiated once by hand. It is important to note that by differentiating the model equations you get new model equations with new information. For a n:th order linear system, n differentiations is enough to extract all information in the model equations. For non-linear systems, no such limit exists, you may even have to differentiate infinitely many times to extract all information.

Therefore, to restrict the design to residual generators of maximum order k, differentiate the measurement equations k times. The higher order residual generators that is considered, the more of the dynamic model can be utilized, e.g. for fault isolation purposes. However, higher orders means a more difficult implementation problem.

6.2 *Design*

The object of the design is to find a set of residuals that form a fault isolating residual structure. For simplicity, 6 elimination ideals (and the corresponding Gröbner bases), are calculated where one fault is eliminated in each ideal. Residual generators are then selected among the basis polynomials of the calculated Gröbner bases.

The design is performed as described in Section 4. Using the variable ordering given by Heuristic 1, the following variable ordering is used when eliminating f_2:

$$\begin{aligned} f_2 \succ \dot{y}_1 \succ \dot{y}_2 \succ y_1 \succ y_2 \succ y_3 \succ y_4 \succ u \succ \\ \succ f_1 \succ f_3 \succ f_4 \succ f_5 \succ f_6, \end{aligned}$$

and corresponding ordering when eliminating the other variables. The variables $\dot{y}_1$ and $\dot{y}_2$ are given a high ordering since it is desirable to eliminate those variables to get simple computational forms of the residual generators.

Then, four consistency relations that creates a fault isolating structure is selected to form the residual generators. The computational form of the residual generators are:

$$\begin{aligned} r_1^{comp} =& \frac{1}{\sqrt[5]{h_0}}(-a_3(5h_0^3 + 15h_0^2 y_1 - 5h_0 y_1^2 + y_1^3) + \\ &+ a_4(5h_0^3 + 15h_0^2 y_2 - 5h_0 y_2^2 + y_2^3) + \\ &+ 16\sqrt[5]{h_0}\dot{y}_2) \\ r_2^{comp} =& -a_6(5h_0^3 + 15h_0^2 y_2 - 5h_0 y_2^2 + y_2^3) + \\ &+ 16\sqrt[5]{h_0} y_4 \\ r_3^{comp} =& a_5(5h_0^3 + 15h_0^2 y_1 - 5h_0 y_1^2 + y_1^3) - 16\sqrt[5]{h_0} y_3 \\ r_4^{comp} =& u a_1 a_5 - a_2 y_3 - a_5 \dot{y}_1 \end{aligned}$$

and the corresponding internal forms are:

$$\begin{aligned} r_1^{int} =& \frac{1}{\sqrt[5]{h_0}}(a_3(-\boxed{f_2}^3(-1 + \boxed{f_6}) - \\ &- \boxed{f_2}^2(-1 + \boxed{f_6})(5h_0 - 3y_1) - \\ &- \boxed{f_2}(-1 + \boxed{f_6})(15h_0^2 - 10h_0 y_1 + 3y_1^2) + \\ &+ \boxed{f_6}(5h_0^3 + 15h_0^2 y_1 - 5h_0 y_1^2 + y_1^3)) - \\ &- a_4 \boxed{f_3}(\boxed{f_3}^2 + 15h_0^2 + \\ &+ \boxed{f_3}(5h_0 - 3y_2) - 10h_0 y_2 + 3y_2^2)) \\ r_2^{int} =& -16\boxed{f_5}\sqrt[5]{h_0} + a_6 \boxed{f_3}(\boxed{f_3}^2 + \\ &+ 15h_0^2 + \boxed{f_3}(5h_0 - 3y_2) - 10h_0 y_2 + 3y_2^2) \\ r_3^{int} =& 16\boxed{f_4}\sqrt[5]{h_0} + a_5(\boxed{f_2}^3(-1 + \boxed{f_6}) + \\ &+ \boxed{f_2}^2(-1 + \boxed{f_6})(5h_0 - 3y_1) + \\ &+ \boxed{f_2}(-1 + \boxed{f_6})(15h_0^2 - 10h_0 y_1 + 3y_1^2) - \\ &- \boxed{f_6}(5h_0^3 + 15h_0^2 y_1 - 5h_0 y_1^2 + y_1^3)) \\ r_4^{int} =& a_1 a_5 \boxed{f_1} + a_2 \boxed{f_4} \end{aligned}$$

By inspection of the internal forms, the residual structure can be concluded to be as in Table 1. All instances of fault variables in the internal forms has been marked by shaded boxes. An X in column i and row j of the table means that fault i influences residual j. Thus, the table directly gives the decision logic. It is clear that, with the set of residuals chosen, there is no way

	f_1	f_2	f_3	f_4	f_5	f_6
r_1		X	X			X
r_2			X		X	
r_3		X		X		X
r_4	X			X		

Table 1. Residual structure

to separate f_2 and f_6. Examining the Gröbner bases obtained when eliminating f_2 and f_6 it can be seen that it is impossible to separate the two faults by only considering model equations (9). This because when eliminating f_2, also f_6 is eliminated and vice versa. By considering higher order residual generators, isolation of f_2 and f_6 is in fact possible.

6.3 *Simulations*

In the simulations, a simple proportional controller is used to control the water level in the upper tank to follow a square reference signal. Figure 3 shows the water levels in both tanks in a fault-free simulation. Note that the operating point deviates significantly from $h_0 = 5$. Figure 4 shows the residuals in the fault-

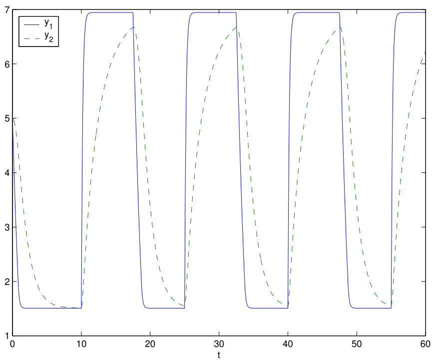

Fig. 3. Water level in the upper tank, y_1, and the lower tank y_2, during fault-free simulations. Note that the working point deviates significantly from $h_0 = 5$, therefore is it important to utilize the model of the non-linearities.

free case. All residuals are below the dotted thresholds. Here, only two fault scenarios are shown, a constant

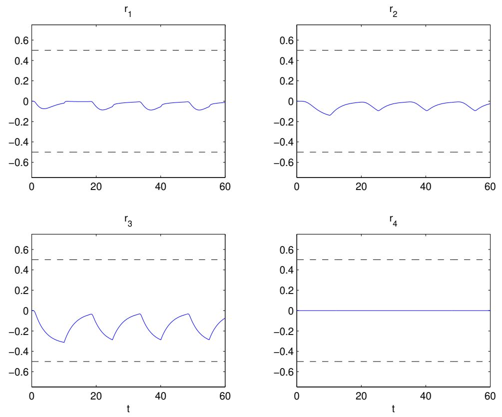

Fig. 4. Residuals in the fault-free case. All residuals are below the dotted thresholds, i.e. no false alarms during the simulation.

fault fault in the third sensor and a clogging in the pipe. Figure 5 shows the residuals when the sensor fault $f_4 = 1$ is induced at time $t = 30$ sec. It is clear that the residuals respond as expected by the residual structure in Table 1, i.e. residuals r_3 and r_4 respond to the fault while r_1 and r_2 does not. The fault is correctly isolated. Figure 6 shows the residuals when a

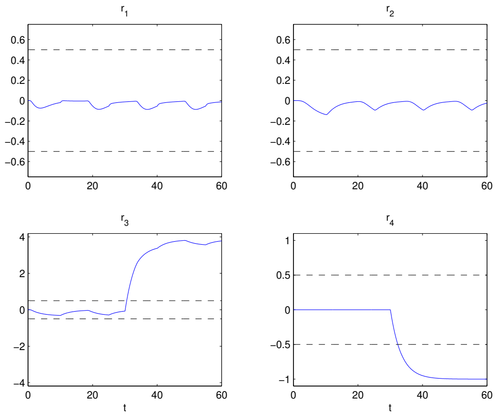

Fig. 5. Residuals when fault $f_4 = 1$ is induced at time $t = 30$ sec. Residuals r_3 and r_4 respond to the fault while r_1 and r_2 does not, i.e. f_4 is correctly isolated according to Table 1.

clogging appears abruptly, resulting in 10% decrease in water flow, i.e. $f_6 = 0.1$. The fault is also here induced at time $t = 30$ sec. Also here, the residuals respond as expected by the residual structure in Table 1, i.e. residuals r_1 and r_3 respond to the fault while r_2 and r_4 does not. The fault is correctly isolated.

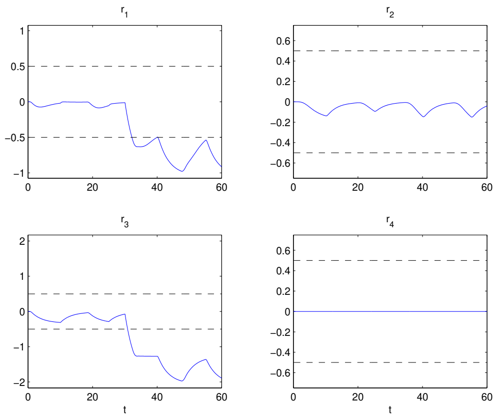

Fig. 6. Residuals when fault $f_6 = 0.1$ is induced at time $t = 30$ sec. Residuals r_1 and r_3 respond to the fault while r_2 and r_4 does not, i.e. f_6 is correctly isolated according to Table 1.

By careful examination of the model equations (8), it is clear that they can be used directly to form residual generators that can detect and isolate the faults, without the need to make polynomial approximations of non-linearities and eliminate faults. The residuals could be formed by just moving all known variables (and their derivatives) to the left hand side. The fault influence on the system is so structured that fault isolation is possible (apart from f_2 and f_6). However, the main point of the example is to show a systematic method to design residual generators for a non-linear system.

7. CONCLUSIONS

Design of residual generators for non-linear polynomial systems has been considered. It is shown how input-output relations of system dynamics is well suited for diagnostic purposes and some advantages compared to a state-space description is discussed.

A systematic approach for design and analysis of disturbance decoupling residual generators is presented. The basic design step is based on well established elimination theory for polynomial equations. The basic calculation step in the design procedure is to calculate a Gröbner basis for an elimination ideal where all disturbances has been eliminated. The available design freedom is discussed and a heuristic for variable ordering when calculating the Gröbner basis is given.

The approach is demonstrated on an example process, two coupled water tanks. The example shows how the design freedom can be used and how fault isolation properties of the model can be analyzed from the calculated Gröbner bases. The example also shows how both multiplicative and additive faults are handled equally in the design process. A simulation study shows how the faults are detected and isolated.

8. REFERENCES

Cox, D., Little, J. & O'Shea, D. (1991). *Ideals, Varieties, and Algorithms*, Springer-Verlag.

Forsman, K. (1991). *Constructive Commutative Algebra in Nonlinear Control Theory*, PhD thesis, Linköping University.

Gertler, J. (1991). Analytical redundancy methods in fault detection and isolation; survey and synthesis, IFAC Fault Detection, Supervision and Safety for Technical Processes, Baden-Baden, Germany, pp. 9–21.

Guernez, C., Casar, J. & Staroswiecki, M. (1997). Extension of parity space to non-linear polynomial dynamic systems, *IFAC Safeprocess 1997*, Hull, UK, pp. 861–866.

Jirstrand, M. (1998). *Constructive Methods for Inequality Constraints in Control*, PhD thesis, Linköping University.

Krishnaswami, V., Luh, G. & Rizzoni, G. (1994). Fault detection in IC engines using nonlinear parity equations, Proceedings of the American Control Conference, Baltimore, Maryland, pp. 2001–2005.

Krishnaswami, V. & Rizzoni, G. (1994). Nonlinear parity equation residual generation for fault detection and isolation, IFAC Fault Detection, Supervision and Safety for Technical Processes, Espoo, Finland, pp. 305–310.

Nyberg, M. (2000). Criterions for detectability and strong detectability of faults in linear systems, *submitted to Safeprocess'2000*.

Nyberg, M. & Frisk, E. (1999). A minimal polynomial basis solution to residual generation for fault diagnosis in linear systems, *Proc. of IFAC 1999 World Congress*, Vol. P, Beijing, P.R. China, pp. 61–66.

Ritt, J. (1950). *Differential Algebra*, Dover Publications.

Zhirabok, A. (1999). Parity space approach to fault diagnosis in nonlinear systems, *ECC'99*.