Fault Diagnosis in Heterogeneous Complex Systems

Björn Schieffer and Günter Hotz

Fachbereich 14 – Informatik, Universität des Saarlandes
Postfach 15 11 50, 66041 Saarbrücken, FRG

email: schieffer@cs.uni-sb.de
www: http://hamster.cs.uni-sb.de

Abstract

In this paper a new approach for the diagnosis problem of complex heterogeneous systems is presented. It needs no linearization of the system and allows arbitrary influences between the subsystems, while this is not the case for many approaches in literature. As a domain example, ballast tanks in offshore plants are used.

1 Introduction

Heterogeneous complex systems contain of several subsystems like electronic devices, mechanical elements or fluid dynamics. Although they influence each other, these subsystems work in parallel. The classical approaches for diagnosis consider homogeneous systems. When used for heterogeneous systems they have to deal with one of the following problems:

- The estimation of the fault parameters [CW84, Mas86, IF91] and observer based approaches [Fra93, GK94] need a linearizable system without any singularitys. But the interaction of subsystems prevents such a linearization.

- [Dav84, Rei87, dKW87] use directions of influences between the subsystems to calculate those which may be responsible for the systems misbehavior. Therefore, loops of influences are not allowed.

In this paper, we propose a new way to solve the diagnosis problem. The basic idea is to approximate the fault parameters of the system with search strategies based on simulations of the system. For simulation the system needs not to be linearizable and the influences between the subsystems do not matter. Therefore, this approach may be used for heterogeneous systems as well as for homogeneous systems.

The paper is organized as follows. We begin with a formal definition of the diagnosis problem. Then a domain example is introduced to demonstrate the methods. In section 4 we show how to reduce the search space with qualitative fault reduction. Finally, the search strategies for the remaining search space are sketched.

2 The diagnosis problem

Diagnosis is troubleshooting in a system that does not behave as expected. Let us start with a definition of such a system:

Definition 1 *A system $S = (X, Y, C, behave)$ consists of the input space X, the output space Y, the configuration space C and the behavior function $behave : X \times C \to Y$.*

The configuration space C describes the states of the system. If the the system is ok, the configuration is $c_0 \in C$. If faulty, the system turns into another configuration $c \in C$. Configurations that stand for the same kind of fault in different intensitys are grouped into a subspace C_i. To achieve that, C is divided into m disjoined subspaces:

$$C = \biguplus_{i=0}^{m} C_i$$

If we are not interested in the exact intensity of a particular fault, we may express the type of the fault by determining the subspace C_i of the current configuration c.

The distribution of C reflects the fault model we use. It may distinguish double faults or even multiple faults. But because of the complexity, it is often reduced to single faults.

There are two special subspaces: C_0 consists only of one element c_0 which is the state of the correct system. The last subspace C_m contains all the remaining configurations. So, if we restrict our interest to single faults, the multiple faults can all be found in C_m. The probability of the configurations in C_m should be very low, because it is of less information if we know that the current configuration is in C_m.

We are now able to define the main problems:

Definition 2 *A* fault reduction *for a system* $S = (X, Y, C, behave)$ *with the current input* $x \in X$ *and the current measurement* $y \in Y$ *is a set* $red \subseteq \{C_i \,|\, 0 \le i \le m\}$ *with:*

$$C_i \notin red \quad \Rightarrow \quad \nexists c \in C_i,\ y = behave(x, c)$$

Definition 3 *A* fault diagnosis *for a system* $S = (X, Y, C, behave)$ *with the current input* $x \in X$ *and the current measurement* $y \in Y$ *is a configuration* $c \in C$ *with* $y = behave(x, c)$*.*

We will see later on that these problems are related and it may be useful to deal with them at once.

3 Domain Example

There is no hope in a general tool for the diagnosis problem. For that reason, one restricts to special domains and trys to learn from them to carry over the experiences made. The domain we consider in this paper are offshore plants.

Ballast tanks are used to keep balance while dealing with heavy loads. They are filled with sea water. Therefore, large pumps suck in the filtered water from the sea. Valves select the used tanks and control the strength of the water flow.

Figure 1 shows the schematic of the example we consider, introduced by [DBMB93]. We find two float switches in each tank to report whether the tanks are completely filled or emptied. According to the position of the valves v_4, v_5, v_6 and v_7 the operation is filling or clearing. The control unit keeps the scope of the flow and pressures with the throttle valve v_8. Therefore, it reads the pressure sensors inside the tanks and nearby the pump. This combination of an electronic control unit with hydraulic elements builds a heterogeneous control loop for which a diagnosis is impossible with conventional methods.

Now, let us have a look at the typical faults of such a plant:

1. Vent pipes compensate the air pressure in each tank. They are closed if a tank is not in use. If one is still closed during an operation or if it is plugged because of a damage with a slipped load, the pressures inside the tanks are changed.

2. When air gets into the pump it leads to a degradation of its performance. This may be caused by a leak or by swirls in the tanks when they are emptied.

3. The filter accumulates seaweed and shells. So it slowly gets plugged. On the other side it may be plugged abruptly by large garbage.

4. The pressure sensors within the tanks measure the air pressure in some small pipes that lead to the ground of the tank. If these pipes are plugged, the sensors report wrong data.

5. The float switches may jam. Then the valve corresponding to the tank is not closed by the control unit and the tank is still filled or air is getting into the pipes below the tank.

6. The valves are opened and closed by little motors. If the position sensor is shifted, the valves are not opened or closed totally.

7. Some of the tanks are situated on top of others. So, their feed pipe leads through the lower tank. Water may flow between the tanks if it leaks.

We have to distinguish three type of faults: First there are the *binary faults* with fault parameter $\lambda \in \{0, 1\}$. The plugging of the vent pipes and the jam of the float switches are such binary faults. The plugging is only of interest when it is total, because a partial plugging still allows the compensation of the air pressure.

The second type are *one-dimensional faults.* All other faults but the last one are faults of that type. We use a fault parameter $\lambda \in [0, 1]$ to express the degree of such a fault. For example, the filter may be plugged with all degrees between none ($\lambda = 0$) and total ($\lambda = 1$).

The last fault is a *two-dimensional fault.* It is described by a fault parameter $\lambda \in [0, 1] \times [0, 1]$. The first dimension represents the size of the leak, whereas the second dimension stands for its location. The location is of relevance, if the lower tank is not totally filled.

In this paper we restrict to the following faults: The plugging of the vent pipe of tank 1, the plugging of the filter and some air inside of the pump. These are sufficient to demonstrate the ideas in our tools. The resulting configuration space is $C := \{0, 1\} \times [0, 1] \times [0, 1]$. If we are content with single fault recognition, we get

$$\begin{aligned}
C_0 &:= \{(0, 0, 0)\} \\
&\quad\ \text{the system is ok} \\
C_1 &:= \{(1, 0, 0)\} \\
&\quad\ \text{the vent pipe of tank 1 is plugged} \\
C_2 &:= \{(0, \lambda, 0) \,|\, 0 < \lambda \le 1\}
\end{aligned}$$

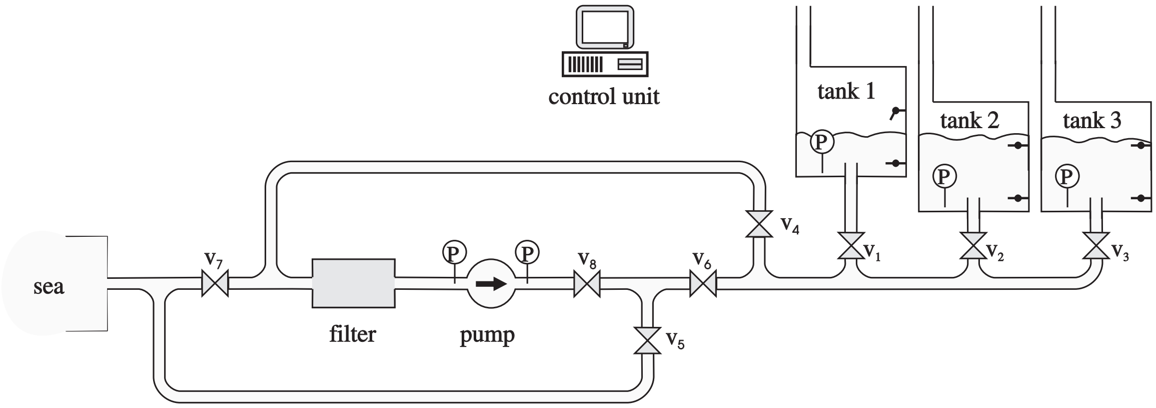


Figure 1: small tank ballast system

$$
\begin{array}{lll}
 & & \text{the filter is plugged} \\
C_3 & := & \{(0,0,\lambda)\,|\,0<\lambda\leq 1\} \\
 & & \text{there is air inside of the pump} \\
C_4 & := & C\setminus(C_0\cup C_1\cup C_2\cup C_3) \\
 & & \text{all multiple faults}
\end{array}
$$

A system input $x \in X$ contains information about the initial charge of the tanks, the valve positions and their manual changes. To continue the example, the tanks shall be empty and all the valves but v_4 and v_5 shall be open: the resulting operation is filling all tanks.

A system output $y \in Y$ contains the values of the pressure sensors and the positions of the valves and float switches. We assume that there are r measurements for any measurable variable during the whole operation. With s measurable variables Y is the space $\mathbb{R}^{r,s}$. As we never use a single value, we always mean the r values of one measurable variable if we speak of *one measurement*.

We assume to have a simulation tool for such a system so that we are able to calculate $behave(x, c)$ for $x \in X$ and $c \in C$.

4 Qualitative fault reduction

We qualify the r measured values of a measurement $m \in \mathbb{R}^r$ with predicates. Predicates of interest are *sign*, *monotony* and *curvature*. To compare two measurements, we apply the predicates to their difference. We have to make a suitable choice of all possible predicates over the available measurements. We call such a choice a *spot*. For our little example, it is sufficient to compare the pressures inside of the tanks with the expected values of $behave(x, c_0)$. That means, the spot consists of the signs of these differences.

The evaluation of a spot for a configuration $c \in C$ is called a *profile*. As we only use comparison in our spot, the possible qualitys are *less* ($-$), *similar* (0) and *greater* (+). One might be surprised that we don't use the term *equal*. But, as there are r values in one measurement, two measurements don't have to be equal, if we are not able to decide between *less* and *greater*.

For sure, the profile of c_0 is $p_0 := (000)$, as there is no difference in that case. The profile of $c = (1, 0, 0)$ is $p_1 := (+++)$: If the vent pipe of tank 1 is plugged the pressure increases in that tank. Therefore, there is more water left for the other tanks and their pressure increases, too.

To eliminate a subspace C_i, we need the profile of the whole subspace. That means, we need the profile of every configuration $c \in C_i$. In addition to the qualitys *less*, *similar* and *greater* for single configurations, we now have the quality *non-uniform* ($*$). It is used if there are configurations in the subspace with the quality *less* as well as configurations with the quality *greater*.

How can we get profiles of a subspace C_i? C_i may be infinite and we want to know, if a predicate is fulfilled for all of these configurations.

1. The first possibility is *qualitative simulation*. Thereby, the system is not simulated for a fixed configuration $c \in C$ but for the whole subspace $C_i \subset C$. Unfortunately, there are many contradictory effects and the qualitative simulation

does not know which one beats the other. Therefore, the results of qualitative simulation are not very satisfying.

2. External knowledge may also help to find profiles. People involved with the system often know how it behaves in typical faulty cases.

3. Our tools calculate the profiles with the following method: A lattice is used as an overlay of the subspace C_i. The system behavior is computed for each configuration of the lattice. If the predicate of interest is fulfilled for all these configurations and the lattice is close enough, we suspect that the predicate is fulfilled for every $c \in C_i$. Unfortunately, this method works only for low-dimensional subspaces, because in other cases, lattices with reliable results are to large. But for single and double faults it is a proper way to get the profiles.

The resulting profiles for C_2 and C_3 are $p_2 := p_3 := (- - -)$: if the filter is plugged or if there is air in the pump, the flow will decrease and therefore the pressure inside of the tanks. For the remaining configurations in C_4 an increase as well as a decrease of the pressures is possible: $p_4 := (* * *)$.

Assuming we measure $y \in Y$ during the operation of the system, we can determine a profile for y as well as we did for a single configuration $c \in C$. We discuss three cases:

- The profile of y is $p = (+ + +)$. The only fault candidates that are consistent with that profile are C_1 and C_4 and we get the fault reduction $red = \{C_1, C_4\}$. From that we know that either the vent pipe of tank 1 is plugged or a multiple fault occurs. The combination of all faults never can be denied. But as it is of lower probability, it usually will be ignored.

- The profile of y is $p = (- - -)$. This includes $red = \{C_2, C_3, C_4\}$ and we are not able to decide between C_2 and C_3. We will see later on, that the methods for exact diagnosis are able to eliminate subspaces and therefore may further reduce the set of fault candidates.

- The profile of y is $p = (+ - -)$. None of the single faults C_1, C_2 or C_3 is able to explain this profile. Therefore, we know that there is a multiple fault in our system: $red = \{C_4\}$.

5 Fault diagnosis

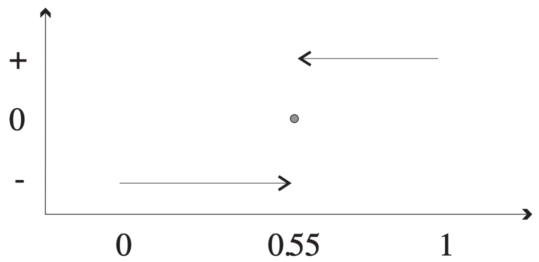
Except blind search, any search algorithm to find a configuration c that may explain the behavior y of the system has to make assumptions to the search space. We assume the following:

Assumption 1 *The behavior of the system S is changed by a fault in a (nearly) monotone way.*

That means, the intensity of an effect caused by a fault should not oscillate with the intensity of the fault, but should also increase. We do not demand this in a strict way. Because there are more than one measured variables, small exceptions may be compensated.

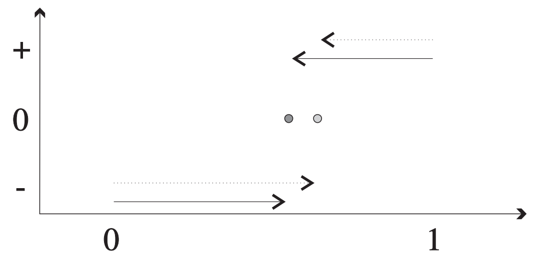
5.1 One-dimensional Subspace

Now we want to search for a configuration c with $y = behave(x, c)$ in a one-dimensional subspace C_i. To continue our example, let us chose $y = behave(x, (0, 0.55, 0))$. The profile of y is $p = (- - -)$ and according to the last section we have to search in the subspaces C_2 and C_3. Starting with C_2 let us consider the pressure inside of tank 3 compared to the corresponding pressure of $behave(x, (0, \lambda, 0))$. Again, we use the qualitys *less*, *similar* and *greater*. Assumption 1 assures us, that there is only one change from the quality *less* to *greater* or vice versa.



We approximate the solution $\lambda = 0.55$ with binary search: Starting with the lower bound $l = 0$ and the upper bound $u = 1$ we see that the middle $m = 0.5$ has the same quality as the lower bound, namely *less* and therefore, the solution lies in the interval $[0.5, 1]$. We iterate this with m as the new lower bound.

We have seen in the last section that the profile $p = (- - -)$ is fit for both subspaces, C_2 and C_3. Therefore, we have to search in the subspace C_3 as well. If we restrict our interest to the pressure of tank 3, the situation looks like in the last figure. But if adding the measurement of the pressure inside of tank 1, we get two different solutions:



From that we can conclude that there is no configuration $c \in C_3$ with $y = behave(x, c)$. So, the exact search is able to further reduce the fault candidates. Even, if we are confronted with a multiple fault, we can at least recognize with this method that none of the one-dimensional faults is able to explain the behavior of the system.

5.2 Two-dimensional Subspace

Up to now we considered only single faults. Now we focus our interest on double faults. Therefore, we reorganize the configuration space of our example:

$C_0 := \{(0,0,0)\}$
the system is ok

$C_1 := \{(1,0,0)\}$
the vent pipe of tank 1 is plugged

$C_2 := \{(0,\lambda,0) \mid 0 < \lambda \le 1\}$
the filter is plugged

$C_3 := \{(0,0,\lambda) \mid 0 < \lambda \le 1\}$
there is air inside of the pump

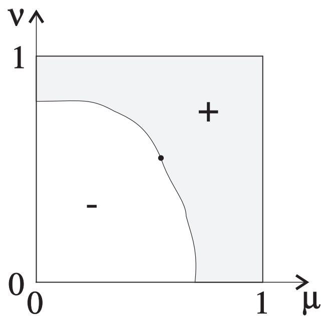
$C_4 := \{(1,\lambda,0) \mid 0 < \lambda \le 1\}$
vent pipe and filter plugged

$C_5 := \{(1,0,\lambda) \mid 0 < \lambda \le 1\}$
vent pipe plugged, air inside of the pump

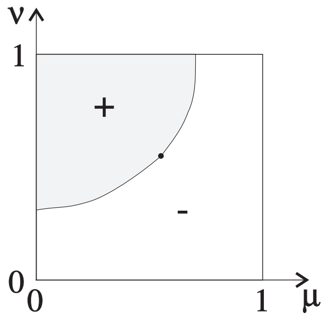
$C_6 := \{(0,\mu,\nu) \mid 0 < \mu, \nu \le 1\}$
filter pipe plugged, air inside of the pump

$C_7 := \{(1,\mu,\nu) \mid 0 < \mu, \nu \le 1\}$
triple fault

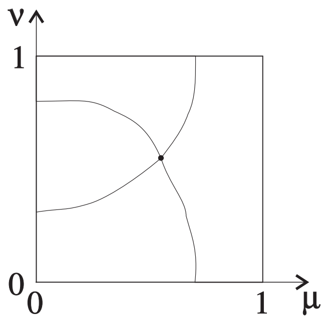
We search for $y = behave(x, (0, 0.55, 0.55)$. The binary search shown above may be used for the one-dimensional subspaces C_4 and C_5. But how about the two-dimensional subspace C_6? Graphically, it is a square. If we illustrate the area (μ, ν) where the pressure inside of tank 3 is *greater* compared to the one of $behave(x, (0, \mu, \nu))$ we get the following figure. Because the pressure is equal for the solution $(0.55, 0.55)$, the borderline between the area of *greater* and the area of *less* crosses this point.



The areas for the pressures of the other tanks look nearly the same. Therefore, they are not helpful to determine the solution. For that we need the pressure before the pump. The areas of that measurement look like that:

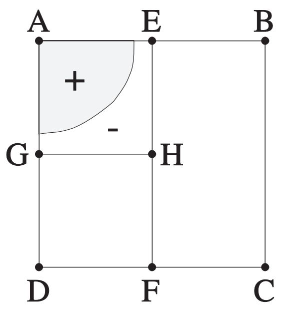


If combined with the pressure of tank 3, the solution is the point of intersection of the two borderlines:



How can we find this point of intersection? We localize it with horizontal and vertical cuts. They distribute the search space into subsquares. Each cut substitutes one square by two new ones. We give them a measure of the guess whether the solution lies in. The search is continued in that subsquare with the highest measure. We do not restrict to the both we created last, but choose the subsquare with the highest measure out of all current squares over the search space. Therefore, a wrong decision does not prevent us from finding the solution. Of course, it will delay the search.

The measure of a subsquare is given as the sum of the borderlines that cross it. We can decide whether the borderline of a measurement crosses the subsquare by observing the corners. At this point we use assumption 1: The borderline can only cross the subsquare if both qualitys – *less* and *greater* – appear at the corners. Let us have a look at the following figure as an example.



The points E and F cuts the square defined by A, B, C and D into two subsquares. The measure of the subsquare bounded by A, E, F and D is one, as the quality for point A is *greater* while it is *less* for the others. The measure of the subsquare with the corners E, B, C and F is zero, as the quality of all corners is *less*. Therefore, we cut the left square with the line between the points G and H. For the resulting subsquare defined by A, E, H and G the measure is one while zero for the subsquare with the corners G, H, F and D.

6 Conclusions

Up to now we concealed the problem of noise. The model used for simulation can't be a perfect image of the real world. And the measurement instruments don't work absolutely perfect. Therefore, the qualitative fault reduction and the fault diagnosis have to be robust against noise. To keep the ideas clean, we didn't mention all what we have done in our implementation. The basic idea is to use the redundance that is given with all the different measurable variables. Only in the case of the two-dimensional search we mentioned that it is necessary to cope with wrong decisions. These are based on noise and violations of assumption 1.

We implemented this system with all the typical faults, mentioned above. After the qualitative fault reduction, only few fault candidates are left. In many cases, only one single candidate remains. For fault diagnosis we implemented the one-dimensional and the two-dimensional search. The ideas should work with three-dimensional search spaces as well, if *search square* is changed into *search cube*.

This work is part of the Ph. D. thesis of the first author and is granted by the German Science Foundation (DFG) in its Special Collaborative Programme on VLSI-Design and Parallelism (SFB 124).

References

[CW84] E. Y. Chow and A. S. Willsky. Analytical redundancy and the design of robust failure detection systems. *IEEE Transactions on Automatic Control*, 29:603–614, 1984.

[Dav84] R. Davis. Diagnostic Reasoning based on structure and behavior. *Artificial Intelligence*, 24:347–410, 1984.

[DBMB93] O. Dressler, C. Böttcher, M. Montag, and A. Brinkop. Qualitative and Quantitative Models in a Model-based Diagnosis System for Ballast Tank Systems. In *Proc. of the International Conference on Fault Diagnosis TOOLDIAG '93, Toulouse, France*, pages 397–405, 1993.

[dKW87] J. de Kleer and B. Williams. Diagnosing multiple faults. *Artificial Intelligence*, 32:97–130, 1987.

[Fra93] P. M. Frank. Advances in observer-based fault diagnosis. In *Proc. of the International Conference on Fault Diagnosis TOOLDIAG '93, Toulouse, France*, pages 817–836, 1993.

[GK94] J. P. Gauthier and I. A. K. Kupka. Observability and observers for nonlinear systems. *SIAM Journal on Control and Optimization*, 33:975–994, 1994.

[IF91] R. Isermann and B. Freyermuth. Process fault diagnosis based on process model knowledge – part 1: Principles for fault diagnosis with parameter estimation. In *Trans. of the ASME*, volume 113, pages 620–626, 1991.

[Mas86] M. A. Massoumnia. A geometric approach to the synthesis of failure detection filters. *IEEE Transactions on Automatic Control*, 31:839–846, 1986.

[Rei87] R. Reiter. A theory of diagnosis from first principles. *Artificial Intelligence*, 32:57–96, 1987.