A Diagnosis Driven Self-Reconfigurable Filter

Emmanuel Benazera
Robotics Group
Bremen Universität
Robert-Hooke-Str. 5,
D-28359, Bremen, Germany
benazera@informatik.uni-bremen.de

Louise Travé-Massuyès
LAAS-CNRS
7, av. du Colonel Roche
31077 Toulouse Cedex4, France
louise@laas.fr

Abstract

Filtering consists in estimating the value of system state variables based on available noisy measurements. In Artificial Intelligence (AI), reasoning from first principles uses logic to trace back influences among variables and finds minimal sets that can be held responsible for a given measurement. Both theories rely on a model of the system, but while filtering implements an error feedback mechanism that closes on the measurements, reasoning from first principles provides the ability to localise the causes from the effects. In certain cases, when a system misbehaves, e.g. the motor in a robotic arm joint starts failing, the filter is able to detect the drift, but unable to locate the problem with precision in the state-space. The ability to break up the filter's feedback loop in such cases is exactly the purpose of our approach. We aim at coupling the localization ability of the theory of diagnosis from first principles with the state estimation achievement of Kalman filtering. The targeted result is a novel filter which localizes the subpart of the system that is misbehaving, isolates its effects, and keeps tracking a partial state.

Introduction

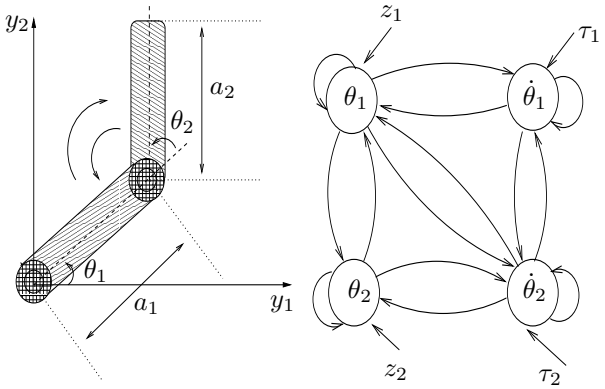
There exist numerous strategies for tracking the state of a possibly faulty system, using noisy measurements. The implied stochasticity of the system dynamics together with the number of faulty situations to account for makes it necessary to track a high number of behavioral hypotheses simultaneously. This is typically done by running either a bank of filters or a cloud of particles (Doucet *et al.* 2000). In most cases, the number of trajectories is untractable, or it is simply counter-productive to track them since many states are in fact never reached. For this reason, research has concentrated on ways to drive the filter's focus on the subset of relevant hypotheses (Hofbaur & Williams 2002a; Narasimhan, Dearden, & Bénazéra 2004) and to mitigate the blowup in tracked states (Hutter & Dearden 2003; Bénazéra & Travé-Massuyès 2003).

While these strategies are effective in practice, not all hypotheses can be modeled, of course, and more so in the case of fault hypotheses, whose number is potentially infinite. An alternative is to design a filter that tracks the potentially unmodeled behaviors. This can be done by fitting parameters to a skeleton model, e.g. using Generalized Likelihood Ratio or Expectation Maximization (Basseville & Nikiforov 1992). The problem is then to anticipate appropriate skeleton models.

In this paper, we adopt a different point of view on the problem of tracking the state of a system. Our approach is based on a reference behavior model (e.g. that of nominal behavior) but instead of closing on all the measurements, we propose to scale the filter so that it only closes on the part of the system that can be trusted to correspond to the reference model. It is necessary that such a filter correctly identifies the variables that fit the model, leaving the others in open loop. The filter naturally leaves the uncertainty to grow on these latter variables. The rational behind it is that the system upper controlling layers act locally on the estimated uncertainty, or *level of unknowingness*, instead of aiming at identifying a fully fitted model. Building the filtering loop to this end is challenging.

First, the subpart of the system and corresponding subset of variables whose behavior does not fit the reference model have to be identified. Although the numerical feedback loop that is natural to most filters makes it difficult to isolate these variables, we argue that they can actually be determined through causal analysis by logically tracing causes from their effects in the causal structure of the model. In AI, a logical theory of diagnosis does exist that can just do that. Diagnosis from first principles (DX) logically infers the minimal sets of elementary components that can be held responsible for a discrepancy in the system (Hamscher, Console, & J. de Kleer 1992). We use the power of this inference to break up the filter feedback loop after projecting the component sets on the corresponding sets of *untrusted* variables. Untrusted variables are hence decoupled from the filter loop. Second, the estimation step needs to be revised so that effects of untrusted variables are prevented from affecting themselves and *sane* variables, while discrepant measurements must not be used for updating the filter's innovation.

The aim of this paper is to bring a reasoning layer as well as a partial covariance minimization scheme into existing filters, starting with the Unscented Kalman filter that applies to nonlinear systems. The contribution stands on the idea of coupling a filtering technique well-known in the Control diagnosis community (FDI) with logical diagnosis inference from the AI diagnosis community (DX). It hence fits into the BRIDGE framework aiming at creating synergies between



(a) Support to the dynamic model.

(b) Causal System Description (CSD) for the robotic arm.

Figure 1: Two-link planar arm representation.

the FDI and DX communities (Gautam *et al.* 2004).

The paper is organised as follows. The next section presents our case study, which is a planar arm with two joints. The succeeding section overviews the principles of model based diagnosis and presents how causal models can be used. This is then interpreted in matricial form, bringing it back to the same framework as filtering methods, and the computation methods for deriving conflicts and diagnoses in this framework are presented. Following is the presentation of the Unscented Kalman Filter and how it can be modifed for partial state hypothesis filtering. Finally our semi-closed loop filter SCL-UKF that accounts for logical diagnosis inference is provided. Early results of the application of SCL-UKF to the planar arm are given and discussed. The paper ends with a section discussing related and future works.

Case study

A two-link arm example

Our case study is a two-link planar arm with two joints, at the shoulder and at the elbow. The state of the system is represented by a vector $x = (\ \theta_1\ \theta_2\ \dot{\theta}_1\ \dot{\theta}_2\ )$ where θ_1,θ_2 are the angular positions of the shoulder and elbow joints, respectively. The angular positions θ_1 and θ_2 are measured. m_1, m_2 are the respective masses of each link. Figure 1(a) pictures a schematic support to the arm dynamic model of figure 2. Our model of the arm includes a PD controller, which allows for the two angular position inputs to be translated into the input torques τ_1 and τ_2. While the model is simple enough, the number of possible faults is staggering. Component-wise, both joints can fail, the mass of the second limb can vary when used to pick up objects. Sensors and the controller may also fail. State-wise, this corresponds to 4 single discrepancies of angular positions and speeds, which yield 2^4 multiple faults, 2^6 with sensor faults, and 2^{10} with controller faults. Thus for such a small system, an exhaustive multi-hypothesis filter would require 2^{10} hypotheses to be modelled. In the following, we show how to build a single filter that does reconfigure itself instead of relying on hypotheses to be modelled.

Diagnosis from First Principles

Diagnosis oriented causal modelling

Reasoning about non-linear systems can be supported by a causal representation of influences among variables. Influences are a conceptualisation of the links established by the components between variables in a system. In fact, causal models have been proposed and shown to be suitable for diagnosis in several pieces of work (Biswas & Manders 2006; Travé-Massuyès *et al.* 2001; Travé-Massuyès & Calderón-Espinoza 2007). The model causal structure then acts as a substitute of dependency recording mechanisms. Causal models are generally supported by an oriented graph, also called *Causal Graph*, in which nodes represent variables and edges represent influences from variable to variable. An oriented edge from variable v_i to variable v_j exists if v_i has an influence on v_j, i.e. if a perturbation on variable v_i affects the value of variable v_j. v_i and v_j are called the *cause* and the *effect* variable of the influence, respectively. Three types of variables exist to model a system:

- *Input variables* are exogenous to the system. Their values are controlled by the system's environment and assumed to be known. n_u is the number of input variables.
- *Measured or output variables* are known, as provided by a sensoring device. n_z is the number of measured variables.
- *State variables* are internal to the model and their values are not known. n_x is the number of internal variables.

Definition 1 (Causal System Description (CSD))**.** *Let $CSD = \{V, \mathcal{I}\}$ be the causal system description where V is the set of variables that define the system, and $\mathcal{I}$ the set of oriented influences that model dependencies.*

Conflicts and Diagnoses

Let's assume that a fault detection mechanism is available and that it activates an alarm when the measured value (also called *observation*) of an output variable is not consistent with the expected value. Such a discrepancy for a measured variable z eventually indicates a misbehavior.

Definition 2 (Discrepant output vector)**.** *Let Z be the vector of output variables. The discrepant observation vector Z^f is a vector of size n_z such that $z_i^f = \begin{cases} 1 \text{ if } z_i \text{ is discrepant} \\ 0 \text{ otherwise.} \end{cases}$*

When one or several output variables misbehave, we can derive all sets of faulty influences that may explain the observations. The influences that may be at the origin of the misbehavior of a variable z_i are those related to the edges belonging to the paths going from the measured nodes to the node representing z_i, also called *ascending influences*. The set of such influences is a *conflict* set in the sense of (Reiter 1987). Conflict sets are sets of influences that cannot behave normally altogether according to the observations. A minimal conflict is a conflict that does not strictly include (in the sense of set inclusion) any conflict. (Reiter 1987) proved that minimal diagnoses can be computed from minimal conflicts.

$$M(x)\begin{bmatrix}\ddot{\theta}_1\\ \ddot{\theta}_2\end{bmatrix}+\begin{bmatrix}-m_2a_1a_2(2\dot{\theta}_1\dot{\theta}_2+\dot{\theta}_2^2)+\sin\theta_2\\ m_2a_1a_2\dot{\theta}_1^2\sin\theta_2\end{bmatrix}+\begin{bmatrix}(m_1+m_2)ga_1\cos\theta_1+m_2ga_2\cos\theta_1+\theta_2\\ m_2ga_2\cos\theta_1+\theta_2\end{bmatrix}=\begin{bmatrix}\tau_1\\ \tau_2\end{bmatrix}$$

where

$$M(x)=\begin{bmatrix}(m_1+m_2)a_1^2+m_2a_2^2+2m_2a_1a_2\cos\theta_2 & m_2a_2^2+m_2a_1a_2\cos\theta_2\\ m_2a_2^2+m_2a_1a_2\cos\theta_2 & m_2a_2^2\end{bmatrix}$$

Figure 2: Two-link planar arm dynamic model.

Proposition 1 (Minimal Diagnosis (Reiter 1987))**.** *Given a discrepant observation vector Z^f, $\Delta \subseteq \mathcal{I}$ is a (minimal) diagnosis for* (CSD, Z^f) *iff Δ is a (minimal) hitting set for the collection of (minimal) influence conflict sets.*

A hitting set of a collection of sets is a set intersecting every set of this collection.

Determining Candidate Diagnoses

In this section, we first interpret influence conflicts and diagnoses in a matricial form, suitable for coupling with the filtering framework. The computational methods for building *conflict* and *diagnosis matrices* are then presented.

Conflicts and diagnoses in a matrix framework

The causal graph associated to CSD can be equivalently represented by an incidence matrix $\mathcal{I}$, of size (n_c, n_c) with $n_c = n_x + n_u + n_z$:

$$\mathcal{I}=\begin{pmatrix}A & B & \emptyset\\ \emptyset & \mathbb{I}_u & \emptyset\\ H & \emptyset & \mathbb{I}_z\end{pmatrix}, \text{with } \mathcal{I}_{ij}=\begin{cases}1 \text{ if } x_i \text{ influences } x_j\\ 0 \text{ otherwise}\end{cases}$$

where A is of size (n_x, n_x), B of size (n_x, n_u), and H of size (n_z, n_x). These are incidence matrices that represent influences among state, input, and output variables, respectively. $\mathcal{I}$ reflects the natural hierarchy of influences: inputs on state, state on measures. $\mathbb{I}_u$ and $\mathbb{I}_z$ are identity matrices and account for effects due to external causes onto inputs (e.g. controller) and outputs (e.g. sensors).

Example. *Figure 1(b) shows the CSD=$\{V, \mathcal{I}\}$ for our case study, with $V = \begin{pmatrix}\theta_1 & \theta_2 & \hat{\theta}_1 & \hat{\theta}_2 & \tau_1 & \tau_2 & z_1 & z_2\end{pmatrix}$. We have: $A = \begin{pmatrix}1&1&1&1\\1&1&1&1\\1&1&1&1\\1&1&1&1\end{pmatrix}, B = \begin{pmatrix}0&0\\0&0\\1&0\\0&1\end{pmatrix}$,*

$$H=\begin{pmatrix}1&0&0&0\\0&1&0&0\end{pmatrix} \text{ and } \mathcal{I}=\left(\begin{array}{cccc|cc|cc}1&1&1&1&0&0&0&0\\1&1&1&1&0&0&0&0\\1&1&1&1&1&0&0&0\\1&1&1&1&0&1&0&0\\\hline 0&0&0&0&1&0&0&0\\0&0&0&0&0&1&0&0\\\hline 1&0&0&0&0&0&1&0\\0&1&0&0&0&0&0&1\end{array}\right)$$

For a given discrepant output vector Z^f, influence conflict sets may as well be represented in matrix form, as indicated by the following definition.

Definition 3 (Influence Conflict Matrix)**.** *Given a discrepant output vector Z^f, an influence conflict matrix Γ is an incidence matrix of size $n_c \times n_c$ whose entries correspond to ascending influences of the discrepant output variables of Z^f.*

In the above matrix, all conflicts are represented but it is difficult to identify each of them and relate them to their corresponding discrepant output variable. Now, conflicting influences naturally map onto variables and conversely. Indeed, influence conflict sets correspond to paths in the causal graph and a path may as well be represented by the edges (influences) or by the nodes (variables). This leads to the following definition.

Definition 4 (Variable Conflict Matrix)**.** *Given a discrepant output vector Z^f, a variable conflict matrix Λ is a boolean matrix of size $n_z \times n_c$ such that* $\begin{cases}\sum_j \Lambda_{i,j} > 0 \text{ if } z_i^f = 1\\ \Lambda_{i,\cdot} = 0, \text{ otherwise.}\end{cases}$

Considering a single row Λ_i of Λ we know that all state, input and output variables indicated by a non zero entry in Λ_i influence the discrepant output z_i^f. *This implies that at least one of these variables has to suffer a faulty influence to cause the discrepancy on z_i^f. Hence this set of variables can equivalently represent the influence conflict.* By *suffer a faulty influence* we mean that in the physical system, there must exist at least one influence on this variable whose effect on the discrepant output is incorrectly captured by the reference model. This set of variables is called a *variable conflict* set.A minimal variable conflict matrix is a matrix whose variable sets indicated by 1-valued entries on each row do not strictly include (in the sense of set inclusion) any variable conflict. Therefore a minimal conflict matrix indicates minimal variable conflicts only. Finally, we define the diagnosis matrix as follows.

Definition 5 (Diagnosis matrix)**.** *Given a discrepant measurement vector Z^f, a diagnosis matrix Δ is an influence incidence matrix of size $n_c \times n_c$ in which at least one faulty influence represented by a 1-value entry accounts for each discrepant measure of Z^f.*

Example. *Consider the arm's shoulder joint measure is discrepant, so $Z_f = (1\ \ 0)$.* $\Lambda = \begin{pmatrix}1&1&1&1&1&1&1&0\\0&0&0&0&0&0&0&0\end{pmatrix}$,

$$\Gamma=\left(\begin{array}{cccc|cc|cc}1&1&1&1&0&0&0&0\\1&1&1&1&0&0&0&0\\1&1&1&1&1&0&0&0\\1&1&1&1&0&1&0&0\\\hline 0&0&0&0&1&0&0&0\\0&0&0&0&0&1&0&0\\\hline 1&0&0&0&0&0&1&0\\0&1&0&0&0&0&0&0\end{array}\right)$$

are the conflict matrices of variables and influences, respectively. Δ *with all entries equal to 0 but* $\Delta_{1,1} = 1$ *is a possible diagnosis matrix.*

Computing Conflict Matrices

The discrepant output vector leads to the identification of the matrix of conflicting influences. This section is concerned with the computational methods for building the conflict and diagnosis matrices defined above.

We suppose a discrepant output vector Z^f. H^f (of size $n_z \times n_x$) is obtained by selecting the rows of H that correspond to positive values of Z^f and zeroing the others. H^f tells which state variables directly affect the discrepant outputs. Effects of state variables on other state variables are taken into account by the matrix A. Thus we dub $X^{f,1} = H^f A$ the *discrepant state influence matrix*. In other words, variable x_i influences output z_j iff $X^{f,1}_{j,i} \neq 0$. However, $X^{f,1}$ expresses direct influences of the state on the outputs. Upstream influences can be captured iterating on A, i.e. by $X^{f,2} = X^{f,1}A$. And so on for k steps, $X^{f,k} = H^f A^k$, until $(A)^{k+1} = (A)^k$. State variable conflicts are made of all influences from state variables onto outputs, thus

$$X^f = H^f + \sum_{i=1}^{k} H^f (A)^k \qquad (1)$$

Here k is such that $(A)^{k+1} = (A)^k$. We define the input influence matrix $B^f = X^f B$ where $B^f_{j,i} \neq 0$ implies that input u_i influences output z_j. Finally the matrix I^f (obtained from the identity matrix of size n_z by keeping the ones corresponding to Z^f) is used to account for sensor failures.

Example. *As before, consider* $Z^f = (1 \quad 0)$. *Therefore* $H^f = \left(\begin{smallmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{smallmatrix}\right)$. $A^2 = \mathbb{I}_x$, *and* $X^f = \left(\begin{smallmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{smallmatrix}\right)$, $B^f = \left(\begin{smallmatrix} 1 & 1 \\ 0 & 0 \end{smallmatrix}\right)$, $\mathbb{I}^f = \left(\begin{smallmatrix} 1 & 0 \\ 0 & 0 \end{smallmatrix}\right)$.

Now, we can build the variable conflict matrix Λ as the concatenation of matrices (X_f, B_f, I_f). Following the consistency-based theory presented above, Λ is the conflict matrix because each of its rows indicates an influence *conflict*.

Example. *Following up on our example:*

$$\Lambda = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Algorithm 1 sums up the steps of the automated generation of Λ.

1: Build H_f from the discrepant lines of H.
2: Compute powers of A.
3: Compute X^f.
4: Compute B^f.
5: Build $\Lambda \leftarrow (X^f, B^f, I^f)$.

Algorithm 1: Conflict generation.

Proposition 2 (Minimal Conflict matrix)**.** *Given a discrepant output vector* Z^f, Λ *is the minimal conflict matrix w.r.t.* Z^f.

Proof. The conflict matrix of variables Λ contains all variables that can held responsible for a discrepant variable. In a graph theoretic framework, the matrix of conflicting influences Γ contains all edges that belong to paths from input, state and output vertices to the discrepant vertices. Paths of increasing lengths correspond to the powers 1 to k of the incidence matrix A. Considering a path p_i in this graph, and assuming that one influence I_r is removed, leads to a subpath sp_i. Then sp_i is no conflict since if I_r is faulty, all the influences in sp_i can be normal, the discrepancy being hence explained by I_r only. The same applies to any subpath of p_i, meaning that p_i corresponds to a minimal conflict. □

Computing diagnosis matrices

Hitting sets based computation From the previous section it comes that the logical theory of diagnosis allows for the generation of the diagnosis candidates through the computation of the hitting sets.

Computing the diagnoses comes back to computing the hitting sets of the subset of variables indicated by each row of Λ. This computation returns the set of diagnosis matrices. An incremental algorithm to generate all the minimal hitting sets based on a set of conflicts was originally proposed by (Reiter 1987), then corrected by (Greiner, Smith, & Wilkerson 1989). This algorithm gives a means to compute diagnoses incrementally, under the permanent fault assumption. It builds a Hitting-Set tree (HS-tree) in which leaves contain the minimal diagnose. Like in (Travé-Massuyès & Calderón-Espinoza 2007), we refer to the algorithm version by (Levy 1991) which is more efficient than the original one because it uses less comparisons at each step. We implement a version of the algorithm where diagnoses are given by matrices, and where edges need not to be labelled.

Algorithm 2 begins with a tree HS consisting of a simple root, with an attached empty diagnosis matrix. Each tree node n supports a diagnosis matrix that records entries that solve the conflicts from the root node to n. The algorithm takes conflicts (vector rows of Λ) in an arbitrary order. For every conflict Λ_i and every element $\Lambda_{i,c}$ of the conflict, the algorithm builds two lists, *newleaves[c]* and *oldleaves[c]* (step 3). New leaves to a leaf l are created whenever Λ_i is not already into Δ_l. Intersection test is a matrix operation that maps influences diagnose onto conflicting variables (step 7). The conversion from state conflicts to influence conflicts is done at step 8. Step 10 creates the local diagnosis matrices, one per influence to a local conflict variable. A new leaf l is pruned if it already contains some conflicts that appear in some old leaf. At the end of the diagnosis procedure (step 19), the minimal hitting sets, and hence the minimal diagnoses that explain the system's misbehaviors, are given by the set of diagnosis matrices attached to the leaves. Note that a trivial diagnosis is one that accounts for simultaneous sensor failures.

The problem of exoneration Generating diagnoses as presented above is rather conservative since there are influences in the diagnoses that are not manifesting themselves thoroughly at the level of discrepant outputs. This occurs whenever an influence belongs to the path to several outputs

```
1: for Each conflict Λ_i in Λ (i.e. row) do
2:   for Each element Λ_{i,c} do
3:     Initialize the lists new-leaves[c]={} and old-leaves[c]={}.
4:   for leaf l of HS do
5:     Δ^l ← diagnosis matrix in leaf l.
6:     /* creating new leaves (intersection test). */
7:     if Δ^l.Λ_i = null vector then
8:       Build Γ_i from Λ_i.
9:       for Each positive Λ_{i,c} do
10:        for Each positive Γ_{c,j} do
11:          create Δ ← Δ^l. Δ_{c,j} = Γ_{c,j}.
12:          add new node (n, Δ) to l, and Δ to new-leaves[c].
13:    /* creating old leaves (intersection is singleton). */
14:    if Δ_l.Λ_i has a single positive value then
15:      add Δ_l to old-leaves[c].
16:    /* closing leaves (inclusion test). */
17:    for Each positive element c in Γ_i do
18:      for Each matrix Δ_n in new-leave[c] do
19:        if Δ_n contains some Δ_o in old-leaves[c] then
20:          close the branch of the node with Δ_n.
```

Algorithm 2: Minimal Hitting sets with diagnosis matrices.

and that not all of them are discrepant.

Example. *Given* $Z^f = \begin{pmatrix} 1 & 0 \end{pmatrix}$*, consider the reduced state diagnosis matrix* $\Delta^x = \begin{pmatrix} 0&0&0&0 \\ 0&1&0&0 \\ 0&0&0&0 \\ 0&0&0&0 \end{pmatrix}$*.* $\Delta^x_{2,2}$ *corresponds to the influence of* θ_2 *on itself. It can be held responsible for the first joint discrepancy, if a component in the second joint has failed. However, the second joint's measure is not discrepant so this makes this diagnosis unlikely.*

The elimination of such cases can be dealt with by adopting the *exoneration assumption* in contrast to the *no exoneration assumption* (Cordier *et al.* 2004) :

- no exoneration assumption: the influences that lie on the path to a discrepant output are potentially identified as faulty, i.e. they belong to a conflict;
- exoneration assumption: the influences that lie on the path to a non discrepant output are assumed to be normal.

```
1: Given Z^f, compute conflicts Λ (Alg. 1).
2: Exoneration:
   • ascending variable matrix Λ^ok on non-discrepant measures (Alg. 1).
   • Λ^exo = Λ ⊖ Λ^ok.
3: Compute Minimal Hitting sets on Λ^exo. (Alg. 2).
```

Algorithm 3: Computation of diagnosis matrices on exonerated conflicts

Note that the adoption of the exoneration assumption requires a thorough analysis of how the faults may manifest in a system. For instance, it may not be applicable to controlled systems in which the controller compensates for the faults or to highly non linear systems in which non linearities may hide the effect of the faults. The exoneration procedure can be efficiently implemented by removing from the conflicts the variables that affect non-discrepant outputs. This is done by generating ascending variables that influence non-discrepant outputs, i.e. gathering the variables that cannot suffer faulty influences for the outputs not to be discrepant. These variables are called *sane* variables.

Definition 6 (Matrix of sane variables)**.** *Given a discrepant output vector* Z^f*, a matrix of sane variables* Λ^{ok} *is a boolean matrix of size* $n_z \times n_c$ *such that* $\begin{cases} \sum_j \Lambda^{ok}_{i,j} > 0 \text{ if } z^f_i = 0 \\ \Lambda^{ok}_{i,.} = 0, \text{ otherwise.} \end{cases}$

The algorithm for determining Λ^{ok} is obviously the same as the conflict generation algorithm 1. The exoneration comes back to removing from the variable conflict matrix Λ all the entries that are 1 in Λ^{ok}, i.e. eliminating all the sane variables from the variable conflicts. This results in the exonerated variable conflict matrix $\Lambda^{exo} = \Lambda \ominus \Lambda^{ok}$. From there, the hitting set algorithm then performs normally on the exonerated set of conflicts. Algorithm 3 computes the diagnosis matrices on exonerated conflicts.

Partial State Hypothesis Filtering

In this section, we rely on the principles of the Uncented Kalman Filter (UKF) to build a filter that uses diagnoses to close only on those variables that can be considered unaffected by broken influences. It leaves the set of affected variables in open loop and lets the uncertainty naturally grow on these variables. This uncertainty is predicted from the model, and as such is theoretically sound. We hence derive a *semi-closed loop UKF* (SCL-UKF). This filter accurately combines the mininal state-space isolation of the previous section in open loop with a scaled a posteriori error minimization in closed loop.

Unscented Kalman filtering

Consider a discrete-time controlled process that is governed by a nonlinear stochastic difference equation (2) and a measurement equation (3).

$$x(t_i) = f(x(t_{i-1}), u(t_i), w(t_i)) \quad (2)$$
$$z(t_i) = h(x(t_i), v(t_i)) \quad (3)$$

$x(t_i)$, $u(t_i)$, and $z(t_i)$ have dimensions n_x, n_u, and n_z, respectively, and $w(t_i)$, $v(t_i)$ represent the process and measurement noise and are assumed to be independent, white and Gaussian with probability distributions $\mathcal{N}(0, Q)$, $\mathcal{N}(0, R)$ respectively. The Unscented Kalman filter (Julier & Uhlmann 1997) uses the Unscented Transform (UT) and fully captures the mean and covariance of the state vector with a minimal set of carefully choosen points, referred to as sigma points. The filter computes an unbiased estimate $\hat{x}$ of the state based on the optimal solution of the least-squares method (Kalman 1960). The state is a concatenation of the original state and noise variables $x^a = [x, w, v]$ of dimension n_a. The selection of a cloud of sigma points applies to the extended state to calculate the sigma matrix

$X^a = [X, X^w, X^v]$. Briefly, the state and error covariance are projected forward through the following equations:

$$\begin{aligned}
P^a(t_{i-1}) &= \begin{pmatrix} P(t_{i-1}) & 0 & 0 \\ 0 & P_w & 0 \\ 0 & 0 & P_v \end{pmatrix} \\
X^a(t_{i-1}) &= [\hat{x}^a(t_{i-1}) \hat{x}^a(t_{i-1}) + \sqrt{(n_a + \lambda) P^a(t_{i-1})}] \\
X(t_i^-) &= f(X^a(t_{i-1}), u(t_i), X^w(t_i)) \\
\hat{x}(t_i^-) &= \textstyle\sum_{j=0}^{2n_a} W_j^m X(t_i^-) \\
P(t_i^-) &= \textstyle\sum_{j=0}^{2n_a} W_j^c [X_j(t_i^-) - \hat{x}_j(t_i^-)][X_j(t_i^-) - \hat{x}_j(t_i^-)]^T \\
Z(t_i^-) &= h(X(t_{i-1}), X^v(t_i)) \\
\hat{z}(t_i^-) &= \textstyle\sum_{j=0}^{2n_a} W_j^m Z(t_i^-)
\end{aligned}$$

where t_i^- indicates *a priori* values, and W^m and W^c are the mean and covariance sigma point weight vectors respectively. An adaptive gain factor K minimizes (in the least-square sense) the error covariance. Noisy measurements are introduced to compute the *a posteriori* state and covariance estimates. These steps summarize as:

$$\begin{aligned}
P_z(t_i) &= \textstyle\sum_{j=0}^{2n_a} W_j^c [Z_j(t_i^-) - \hat{z}_j(t_i^-)][Z_j(t_i^-) - \hat{z}_j(t_i^-)]^T \\
P_{xz}(t_i^-) &= \textstyle\sum_{j=0}^{2n_a} W_j^c [X_j(t_i^-) - \hat{x}_j(t_i^-)][Z_j(t_i^-) - \hat{z}_j(t_i^-)]^T \\
K &= P_{xz} P_z^{-1} \\
\hat{x}(t_i) &= \hat{x}(t_i^-) + K(z(t_i) - \hat{z}(t_i^-)) \\
P(t_i) &= P(t_i^-) - K P(t_i^-) K^T
\end{aligned}$$

Partial variance minimization

For a given diagnosis, we produce a partial estimate that is not subjected to the effects of faulty influences. This implies:

- not using discrepant observations and therefore cancelling the measurement noise they introduce;
- cancelling the effects of faulty influences on sane variables, i.e. not influenced by a faulty influence;
- cancelling the effects of faulty influences on the untrusted variables, i.e. influenced by a faulty influence.

The first point is achieved by reducing the output matrix to non-discrepant observable dimensions only. Second and third points lead to the cancelling in the gain computation of the error introduced by the untrusted variables. However, effects of sane variables on the untrusted variables are preserved. In the following we denote by $\breve{x}, \breve{P}, \breve{K}, \cdots$ the elements (state, covariance matrix, gain, ...) of the partial filter. So we have

$$\breve{x}(t_i) = \breve{x}(t_i^-) + \breve{K}(t_i)\Big(\breve{z}(t_i) - \breve{H}(\breve{x}(t_i^-))\Big) \qquad (4)$$

where $\breve{z}$ are the non-discrepant outputs, $\breve{H}$ is the reduction of H to non-discrepant dimensions, $\breve{K}$ the gain that does not account for the error on the set of untrusted variables. It follows that the a posteriori partially estimated error $\breve{e}(t_i)$ is given by

$$\begin{aligned}
\breve{e}(t_i) &= x(t_i) - \breve{x}(t_i) \\
&= \breve{e}(t_i^-) + \breve{K}(t_i)\Big(v(t_i) - \breve{H}(\breve{e}(t_i^-) - e^f(t_i^-))\Big) \qquad (5)
\end{aligned}$$

where $e^f(t_i^-)$ is an n_x dimensional vector such that $e_j^f(t_i^-) = e_j(t_i^-)$ if x_j is affected by an influence of Δ, 0 otherwise. From there, the partially updated covariance is given by[1]

$$\breve{P}(t_i) = \sum_{j=0}^{2n_a} W_j^c [(\hat{X}_j(t_i) - \hat{x}_j(t_i)) - (\hat{X}_j^f(t_i) - \hat{x}_j^f(t_i))]^T$$

with

$$\begin{cases}
\hat{X}_j(t_i) &= \hat{X}_j(t_i^-) + \breve{K}(z(t_i) - \hat{Z}_j(t_i^-)) \\
\hat{x}_j(t_i) &= \hat{x}_j(t_i^-) + \breve{K}(z(t_i) - \hat{z}_j(t_i^-)) \\
\hat{X}_j^f(t_i) &= \hat{X}_j^f(t_i^-) + \breve{K}^f(z(t_i) - \hat{Z}_j^f(t_i^-)) \\
\hat{x}_j^f(t_i) &= \hat{x}_j^f(t_i^-) + \breve{K}^f(z(t_i) - \hat{z}_j^f(t_i^-))
\end{cases}$$

with $(z(t_i) - \hat{Z}_j^f(t_i^-)) = (z(t_i) - \hat{z}_j^f(t_i^-)) = 0$ since untrusted variables are predicted, and

$$\hat{X}^f(t_i^-) = \frac{\partial F}{\partial X^f}(\hat{X}(t_{i-1})), \hat{x}^f(t_i^-) = \sum_{j=0}^{2L} W_j^m X_j^f$$

where the X_j^f are sigma points for the affected variables.[2] This leads to

$$\begin{aligned}
\breve{P}(t_i) = \breve{P}(t_i^-) + \breve{P}^f(t_i^-) - T^f(t_i^-) - (T^f(t_i^-))^T \\
+ \breve{K}\breve{P}_z(t_i^-)\breve{K}^T - \breve{K}\breve{P}_{xz}^T(t_i^-) + \breve{K}\breve{P}_{xz}^f(t_i^-) \\
- \breve{P}_{xz}(t_i^-)K^T + \breve{P}_{xz}^f(t_i^-)\breve{K}^T \qquad (6)
\end{aligned}$$

where $\breve{P}^f(t_i^-) = E[e^f(t_i^-)(e^f(t_i^-))^T]$, $T^f(t_i^-) = E[\breve{e}(t_i^-)(e^f(t_i^-))^T]$ and $\breve{P}_{xz}$, $\breve{P}_{xz}^f$ are the cross-covariances. Minimizing the partial a posteriori error matrix leads to

$$\breve{K}(t_i) = (\breve{P}_{xz}(t_i^-) - \breve{P}_{xz}^f(t_i^-))\breve{P}_z^{-1} \qquad (7)$$

The a posteriori update is written

$$P(t_i) = P(t_i^-) - \breve{K}\breve{P}_z(t_i^-)\breve{K}^T \qquad (8)$$

Hypothesis Testing

The minimal candidate diagnoses generation procedure produces many hypotheses. Different hypotheses carry different levels of uncertainty. Observing that relation 6 rewrites

$$\begin{aligned}
\breve{P}(t_i) = \breve{P}(t_i^-) + \breve{P}^f(t_i^-) - T^f(t_i^-) - (T^f)^T(t_i^-) \\
+ (\breve{P}_{xz}^f - \breve{P}_{xz})^T \breve{K}^T \qquad (9)
\end{aligned}$$

and the error introduced by the untrusted state block is given by

$$P(t_i) - \breve{P}(t_i) = T^f(t_i^-) + (T^f)^T(t_i^-) - \breve{P}^f(t_i^-)$$

In general, we expect the correct diagnosis to best mitigate the growth of uncertainty on the system state. Whenever this is not the case, we expect a wrong diagnosis to lead to recurrent detection of the same error. Here we pose $\mathcal{D} = \frac{P(t_i) - \breve{P}(t_i)}{\breve{P}(t_i)}$ and hence look for the hypothesis with minimum trace $tr(\mathcal{D})$.

[1] This is for the UKF, the derivation of the partial minimization linear Kalman gain is given in (Bénazéra & Travé-Massuyès 2007).

[2] The partial filter requires the state projection's partial derivatives, that do not appear in the derivation of the original filter.

1: initialization: $CSD = \{x, I\}$.
2: $(\breve{x}(t_i), \breve{P}(t_i)) \leftarrow$ *Filter(CSD)*.
3: Compute $\delta(\breve{x}(t_i), \breve{P}(t_i))$ and Z^f.
4: **if** there is at least one discrepant observation **then**
5: Compute Λ, Γ and diagnoses (Algorithm 3).
6: Select diagnosis matrix $\Delta^* = \min_\Delta(\mathcal{D}(\Delta))$.
7: If $\Delta^* == 0$ Then *Filter* $\leftarrow$ *UKF*.
8: Else *Filter* $\leftarrow$ *UKF with partial minimization* using Δ^*.

Algorithm 4: Semi-closed loop filter (SCL-UKF).

Fault Detector

We define a simple fault detector based on a Mahalonobis distance which is the statistical distance of a point from a reference mean point. We characterize as discrepant the points that have 99% chances to lie outside $P(t_i^-)$.

Semi-closed loop filter

Our filter closes a loop on sane variables but runs a predictive open loop on unstrusted fragments of the system state. Growing, the uncertainty eventually re-captures the discrepant measures. When this occurs, it is possible to use the additional information to mitigate the growth of the a posteriori error. By scaling the observation space to the re-captured signals, diagnosing, and adapting optimal gains accordingly, we build the SCL-UKF (algorithm 4). This filter uses a mininal state-space isolation in open loop with a scaled a posteriori error minimization in closed loop.

Results

Our case study is the two-link planar robotic arm presented at the beginning of this paper. We used a numerical simulator of the arm movements.

Single fault and hypothesis

First, we study the SCL-UKF on a single fault and hypothesis. Figure 3 pictures its reaction to an incipient change in the second link mass m_2 at step 40 that leads to a discrepant measure of θ_2. The Hitting-Set algorithm produces 21 non-exonerated diagnose. The filter on figure 3 runs on a rejection of the measure of θ_2. Consequently, the filter trusts and closes on the first joint's angular position θ_1 (3(c)). This proves the newly derived gain is able to well decouple the uncertainty since state variables are otherwise tightly coupled. To estimate θ_2, $\dot{\theta}_2$, the SCL-UKF switches between the UKF and the UKF with partial gain (3(a), 3(b)). On the same scenario, a UKF with standard gain closes on the faulty signals with no bulge in the error covariance.

Hypothesis testing

Second, we study the hypothesis testing. Of the 21 diagnose (hypotheses), most correspond to broken influences on the four state variables. Figure 3(d) pictures $tr(\mathcal{D})$ for these four hypotheses and the 35 calls to the UKF with partial gain. Discrimination between $\dot{\theta}_1$ and $\dot{\theta}_2$ is easy: $\dot{\theta}_2$ introduces less uncertainty to the estimate. Hypothesis of a second arm joint positioning failure (θ_2) is eliminated.

Looking at the SCL-UKF as an hypothesis driven self-reconfigurable filter, it wears similarities with Rao-Blackwellized particle filters (RBPF) (Doucet *et al.* 2000) as it selects behavioral hypotheses. However, the RBPF samples hypotheses whereas the SCL-UKF logically draws them from the discrepancies. Also, the RBPF would need around 2^{10} hypotheses and a transition model to capture the arm multiple fault combinations. The SCL-UKF requires partial derivatives for all hypotheses[3], but remains more compact.

Future and related works

We have coupled diagnosis reasoning from first principles with Kalman filtering techniques for nonlinear systems. The result is a novel filter that opens and closes to estimation fragments of its state according to logical selection of diagnosis hypotheses.

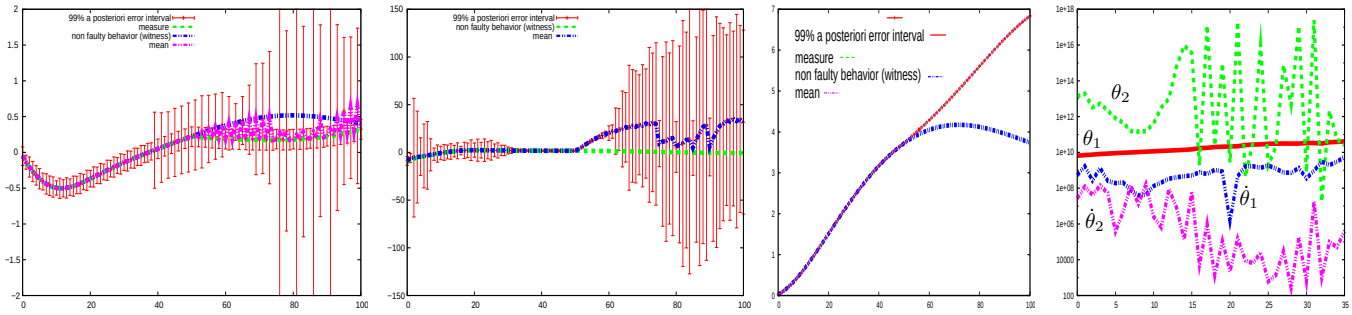
Related works

In (Hofbaur & Williams 2002b) a partial filter is presented that uses a decoupling based on causal and structural analysis of components. However, this scheme only produces independent filters on different subpart of the whole state, as it relies on a bidirectional decoupling of trusted/untrusted state and measured variables. (McIlraith *et al.* 2000) proposes a backward analysis of a causal-graph for producing diagnose and model fitting to adapt to discrepancies. Likewise, adaptive filtering enhances the filter to close on the observations. In that sense, they do not reveal the true uncertainty on the state. We believe that maintaining true uncertainty is key to the efficient control of stochastic systems since it permits for the exploration of a larger but accurately bounded space. While there are no works that we know of about intelligent semi-closed loop Kalman filtering, semi-closed loops have been studied in filtering with numerically bounded uncertainty in (Armengol *et al.* 2000; Benazera, Travé-Massuyès, & Dague 2002). Also, the self-reconfiguration through reasoning from first principles relates to logical filtering (Amir & Russel 2003) as the filtering distributes over disjunctions of the belief state (hypotheses).

Future work and possible extensions

We see at least two extensions to our coupling of diagnosis reasoning and filtering techniques. First, improvements of the RBPF have concentrated on the continuous space and a better use of observations (Hutter & Dearden 2003). However, the RBPF remains limited in the number of modes it can track. We believe that the subset of modes of interest can be reduced by using reasoning and decoupling techniques such as ours, and maintaining a hitting set tree of particle hypotheses for example. Second, we look forward embedding our partial filtering technique into the reinforcement learning framework, for decision and control, and building on existing work (Szita & Lorincz 2004);

[3]It is not too difficult to symbolically or numerically compute the derivatives online.



(a) Angle of the elbow joint (time, θ_2).

(b) Angular speed of the elbow joint (time, $\dot{\theta}_2$).

(c) Angle of the shoulder joint (time, θ_1).

(d) Hypothesis testing (calls, $log(tr(\mathcal{D}))$).

Figure 3: Case study: robotic arm effector mass changes at step 40 while moving its shoulder joint to a reference angle π.

Acknowledgements Emmanuel Benazera is supported by the DFG under contract number SFB/TR-8 (A3).

References

Amir, E., and Russel, S. 2003. Logical filtering. In *IJCAI-03*.

Armengol, J.; Vehi, J.; Travé-Massuyès, L.; and Sainz, M. 2000. Interval model-based fault detection using multiple sliding time windows. In *SAFEPROCESS*, 168–173.

Basseville, M., and Nikiforov, I. V. 1992. *Detection of abrupt changes: theory and application*. Prentice-Hall.

Bénazéra, E., and Travé-Massuyès, L. 2003. The consistency approach to the on-line prediction of hybrid system configurations. In *IFAC Conference on Analysis and Design of Hybrid Systems (ADHS-03)*.

Bénazéra, E., and Travé-Massuyès, L. 2007. A diagnosis driven self-reconfigurable filter (extended). In *Internal Report LAAS-CNRS, Toulouse, France. 9p*.

Benazera, E.; Travé-Massuyès, L.; and Dague, P. 2002. State tracking of uncertain hybrid concurrent systems. In *DX-02*, 106–114.

Biswas, G., and Manders, E.-J. 2006. Integrated system health management to achieve autonomy in complex systems. In *6th Symposium on Fault Detection, Supervision and Safety for Technical Processes*.

Cordier, M.-O.; Dague, P.; Lévy, F.; Montmain, J.; Staroswiecki, M.; and Travé-Massuyès, L. 2004. Conflicts versus analytical redundancy relations : A comparative analysis of the model-based diagnostic approach from the artificial intelligence and automatic control perspectives. *IEEE Transactions on Systems, Man and Cybernetics - Part B.* 34(5):2163–2177.

Doucet, A.; de Freitas, N.; Murphy, K.; and Russell, S. 2000. Rao-blackwellised particle filtering for dynamic bayesian networks. In *UAI-00*.

Gautam, G.; Cordier, M.; Lunze, J.; Staroawiecki, M.; and (Eds), L. T.-M. 2004. Diagnosis of complex systems: Bridging the methodologies of the fdi and dx communities. *IEEE Transactions on Systems, Man and Cybernetics - Part B, Special Issue.* 34(5).

Greiner, R.; Smith, B. A.; and Wilkerson, R. W. 1989. A correction to the algorithm in reiter's theory of diagnosis. *Artificial Intelligence* 41(1):79–88.

Hamscher, W.; Console, L.; and J. de Kleer, e. 1992. *Readings in Model-Based Diagnosis*. Morgan Kaufmann.

Hofbaur, M., and Williams, B. 2002a. Mode estimation of probabilistic hybrid systems. *HSCC-2002* 2289:253–266.

Hofbaur, M., and Williams, B. 2002b. Hybrid diagnosis with unknown behavioral modes. In *DX-02*, 97–105.

Hutter, F., and Dearden, R. 2003. The gaussian particle filter for diagnosis of non-linear systems. In *DX-03*.

Julier, S., and Uhlmann, J. 1997. A new extension of the Kalman filter to nonlinear systems. In *Int. Symp. Aerospace/Defense Sensing, Simul. and Controls.*

Kalman, Rudolph, E. 1960. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering* 82(Series D):35–45.

Levy, F. 1991. Reason maintenance systems and default theories. Technical report, Universit de Paris Nord. Internal report L.I.P.N., http://www-lipn.univ-paris13.fr/ levy/Publications/RMSaDT.pdf, 31p.

McIlraith, S.; Biswas, G.; Clancy, D.; and Gupta, V. 2000. Hybrid systems diagnosis. In *HSCC-2000*.

Narasimhan, S.; Dearden, R.; and Bénazéra, E. 2004. Combining particle filters and consistency-based approaches for monitoring and diagnosis of stochastic hybrid systems. In *DX-04*.

Reiter, R. 1987. A theory of diagnosis from first principles. *Artificial Intelligence* 32:57–95.

Szita, I., and Lorincz, A. 2004. Kalman filter control embedded into the reinforcement learning framework. *Neural Computation* 16:491–499.

Travé-Massuyès, L., and Calderón-Espinoza, G. 2007. Timed fault diagnosis. In *European Control Conference ECC-07*.

Travé-Massuyès, L.; Escobet, T.; Pons, R.; and Tornil, S. 2001. The Ca-En diagnosis system and its automatic modelling method. *Computacin y Sistemas Journal* 5(2):128–143.