Identifying Model-Based Reconfiguration Goals through Functional Deficiencies

Content Areas: planning, reconfiguration, diagnosis, model predictive control, hybrid system

Abstract

Model-based diagnosis is now advanced to the point autonomous systems nowadays face certain uncertain and faulty situations with success. The next step toward even more autonomy is to have the system recovering itself after faults occur, a process known as *model-based reconfiguration*. After faults occurred, given a prediction of the nominal behavior of the system and the result of the diagnosis operation, this paper proposes to automatically determine the *functional deficiencies* of the system. These deficiencies are characterized in the case of uncertain state estimates. A methodology is then presented to determine the reconfiguration goals based on the deficiencies. Finally, a recovery process interleaves planning and model predictive control to restore the deficiencies in prioritized order.

Introduction

Model-based autonomous systems already face faulty situations with some success: they detect and diagnose faults by either identifying potential candidates to their own physical state (Hofbaur and Williams 2002) or reasoning on their structural and behavioral knowledge (Hamscher *et al.* 1992). The next step toward even more autonomy is to have the system recovering itself after faults occur, a process known as *model-based reconfiguration*[1] (MBR). Automated reconfiguration comprehends three steps: goal identification, goal selection, recovery. *Goal identification* searches for a set of potential states of the system where the fault effects are inhibited; *goal selection* is the process of deciding the best of these states, denoted goal states; *recovery* searches for the chain of actions that may turn the physical system state into the desired goal states. Recent architecture design for autonomy (Muscettola *et al.* 1998) puts the goal identification and selection processes outside the scope of a model-based diagnoser, in the hands of upper decisional levels. The aim of this paper is to produce an automated goal identification/selection/recovery methodology that takes better advantage of the system model. Due to several factors, MBR is a challenging problem:

- The state of the system cannot be uniquely determined in all situations. Recent model-based monitoring/diagnosis systems tracks several potential non-faulty/faulty state estimates simultaneously (Nayak and Kurien 2000; Benazera and Travé-Massuyès 2003). Moreover, the set of state estimates is the result of a selection process as the total number of possible states is too large to be explored. The ambiguity is however mitigated by the fact that the number of state estimates is typically small.
- Faults effects may differ from one estimate to the other. For this reason, pre-compiled policies may fail recovering the system by proposing an improper command when the state is uncertain.
- Nowadays, embedded digitally controlled systems have complex behaviors characterized by a preeminence of discrete switches in their dynamics. They are modeled as hybrid systems, that exhibit both discrete and continuous dynamics.

The main idea that is developed in this paper is that when you lose your marbles, your first try is to recover them. Referring to the *faulty states* as the estimates that result from the diagnosis operation, as opposed to the nominally *predicted states*, we propose to compare the faulty states and the predicted states and thus determine the *functional deficiencies* caused by the faults. In this context, functional deficiencies are variable instances in one or more predicted states and that have been *lost* in one or more faulty states. Our approach seeks to minimize the size of a functionality to recover while maximizing its coverage of the estimates. The contributions of this paper are threefold. First, we show how this strategy leads to a finite set of disjoint functional deficiencies, and characterize them. Second, we propose a methodology to identify potential goals from the deficiencies based on a productive analogy with model-based diagnosis, reasoning at a single point in time, despite the system continuous dynamics. Third, we show how to interleave conformant planning and model predictive control to bring the system's hybrid dynamics from the initial potential faulty states to the potential goals states.

Hybrid Model-Based State Prediction and Diagnosis

In this section we introduce a comprehensive formalization of model, state and uncertainty. The autonomous system is

[1] For now, most embedded controllers include pre-compiled recovery policies as part of a rule-based system.

considered a model-based system, i.e. that has a structural and behavioral knowledge of itself.

Definition 1 (Model-Based System). *A model-based system A is a tuple $(\mathcal{C}, \mathcal{M}, \mathcal{T}, \mathcal{X}, E)$, where $\mathcal{C}$ is a set of modeled components, $\mathcal{M}$ a set of finite discrete variables as component behavioral modes, $\mathcal{T}$ a set of transitions among these modes, $\mathcal{X}$ the set of continuous variables and E a set of continuous static/differential equations over $\mathcal{X}$.*

In this paper we use a hybrid description of the physical system's state. The *hybrid state* s is the tuple (M, X). Instances of variables $v \in M \cup X$ are noted $(v = v^j)$, or v^j for short. The hybrid state's discrete side abstracts the physical system as a set of mode instances $M = \bigwedge_k C_k.m^{i_k}$ where $C_k.m^{i_k}$ is an instance of a variable $m \in \mathcal{M}$ of component $C_k \in \mathcal{C}$. The continuous state X is made of instances x^j of continuous variables of $\mathcal{X}$. Observed instances are noted Y, and $\tilde{Y}$ denotes the measured values. Commands are noted U. We consider a discrete-time model of the form:

$$E: \begin{cases} X(k+1) &= f\big(X(k), U(k)\big) \\ Y(k) &= g\big(X(k), U(k)\big) \\ 0 &\leq h\big(X(k), U(k)\big) \end{cases} \quad (1)$$

System A's behavior is described with rules of the form $\bigwedge_i e_i$ **if** ϕ, where $e_i \in E$ and ϕ is a conjunction of equalities/inequalities over functions of variables in $M \cup X$. A set $T = \{\tau_1, \cdots, \tau_{n_m}\}$ of transitions is specified for each mode m. Each transition τ is enabled according to a guard ϕ, and may trigger with probability $p(\tau)$ whenever the guard is satisfied. $T(s_i, s_j)$ denotes the set of transitions that moves A from s_i to s_j.

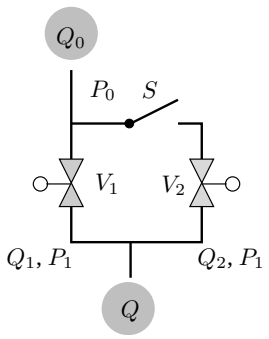
Given the ability A has to predict and diagnose its own behavior, we respectively note $\mathcal{P}(A)$ the prediction of the hybrid system's nominal states, and $\mathcal{D}(A)$ the diagnosis result after faults occur. Note that when fault modes are present, the diagnosis may become an identification problem, and $\mathcal{P}(A)$, $\mathcal{D}(A)$ may result from the same engine. Uncertainty on the physical system's state imposes to consider $\mathcal{P}(A)$ and $\mathcal{D}(A)$ as sets of hybrid states. We denote $\mathcal{S} = \big(\mathcal{P}(A), \mathcal{D}(A)\big)$.

Example (Pressure regulator). *Figure 1 pictures our case study: a two valves system that regulates water pressure between flow entry Q_0 and flow output Q. An electric switch S powers valve V_2 when pressure P_0 equals or exceeds threshold P^*. V_2 opens when powered. S, V_1 and V_2 have two nominal operational modes* open *and* closed*, and two faulty modes* stuck_closed, stuck_open*. Q_0 and Q are measured. P_0 is the single input to the system.*

Our scenario supposes faults occur when the prediction of the nominal state is uncertain[2], i.e. the uncertainty on the pressure does not allow to discriminate between two predicted states[3]:

[2]This corresponds to the general case of tracking multiple states simultaneously.

[3]Flows > 0 are abstracted from their real values for an improved readability.



$$V_i \begin{cases} P_i = P_{atm} \\ Q_i = k_i S_i \sqrt{P_0 - P_i} \textbf{ if } \phi_i \\ Q_i = 0 \textbf{ if } \neg\phi_i \\ \phi_i\text{: } P_0 \geq P_i \\ \quad \wedge (V_i.m = open \vee V_i.m = stuck\_open) \\ \\ \tau_1\text{: } V_2.m = closed \wedge \textbf{S.m=closed} \rightarrow V_2.m = open \\ \tau_2\text{: } V_2.m = open \wedge \textbf{S.m=open} \rightarrow V_2.\, = closed \end{cases}$$

$$S \begin{cases} \tau_3\text{: } S.m = open \wedge \textbf{(}P_0 \geq P^*\textbf{)} \rightarrow S.m = closed \\ \tau_4\text{: } S.m = closed \wedge \textbf{(}P_0 < P^*\textbf{)} \rightarrow S.m = open \end{cases}$$

Connection: $Q_0 = Q = Q_1 + Q_2$

Figure 1: Pressure regulator

$$s_N^1: \begin{cases} Q_0 > 0 \\ P_0 < P^* \\ V_1.m = open \\ S.m = open \\ V_2.m = closed \\ Q_1 > 0 \\ Q_2 = 0 \\ Q > 0 \end{cases} \text{ and } s_N^2: \begin{cases} Q_0 > 0 \\ P_0 \geq P^* \\ V_1.m = open \\ S.m = closed \\ V_2.m = open \\ Q_1 > 0 \\ Q_2 > 0 \\ Q > 0 \end{cases}$$

After observing $Q_0 > 0 \wedge Q = 0$, A returns diagnose, based on the knowledge of the nominal states above:

$$s_F^1: \begin{cases} Q_0 > 0 \\ P_0 < P^* \\ \mathbf{V_1.m = stuck\_closed} \\ S.m = open \\ V_2.m = closed \\ Q_1 = 0 \\ Q_2 = 0 \\ Q = 0 \end{cases}, s_F^2: \begin{cases} Q_0 > 0 \\ P_0 \geq P^* \\ \mathbf{V_1.m = stuck\_closed} \\ S.m = closed \\ \mathbf{V_2.m = stuck\_closed} \\ Q_1 = 0 \\ Q_2 = 0 \\ Q = 0 \end{cases}$$

$$\text{and } s_F^3: \begin{cases} Q_0 > 0 \\ P_0 \geq P^* \\ \mathbf{V_1.m = stuck\_closed} \\ \mathbf{S.m = stuck\_open} \\ V_2.m = closed \\ Q_1 = 0 \\ Q_2 = 0 \\ Q = 0 \end{cases}$$

s_F^1 is the faulty state diagnosed from s_N^1 while s_F^2 and s_F^3 have been deduced from s_N^2. Hybrid states in $\mathcal{P}(A) = (s_N^1, s_N^2)$ and $\mathcal{D}(A) = (s_F^1, s_F^2, s_F^3)$ contain enough information for the autonomous system to extract its *functional deficiencies*.

Functional Deficiencies

Given a belief on a model-based system A, we extend $\mathcal{P}(A)$ and $\mathcal{D}(A)$ by the states probabilities such that $\mathcal{P}(A) = ((s_N^1, p(s_N^1)), \cdots, (s_N^n, p(s_N^n)))$ is the set of the n nominally predicted states, and their associated probabilities, and $\mathcal{D}(A) = ((s_F^1, p(s_F^1)), \cdots, (s_F^f, p(s_F^f)))$ the set of f faulty states from diagnosis, and their attached probabilities. Given a variable v, we note $s(v)$ its value in state s. Any set of nominal and faulty states in $\mathcal{S}$ is denoted a *reconfiguration set*. We want to find a set $\mathcal{F}$ of prioritized variable instances in $M \cup X$ that are the functional deficiencies between states in $\mathcal{P}(A)$ and $\mathcal{D}(A)$, and thus need to be recovered. The general idea that is developed in this section has been inspired by the model-based reconfiguration of logical functions in (Stumptner and Wotawa 1999).

Deficient variable instances

Given two states (s_N, s_F) respectively from $\mathcal{P}(A)$ and $\mathcal{D}(A)$, and a variable v, we note $L\big(s_N(v), s_F(v)\big)$ the measure of the common ground of v's value in each state. We say that the value of v in s_N is *deficient* in s_F when $L\big(s_N(v), s_F(v)\big)$ is smaller than the mean measure of the estimates over *misbehaving* observed variables that correspond to the same states s_N and s_F, i.e.:

$$L\big(s_N(v), s_F(v)\big) \leq \frac{\sum_{y \in Y_{misb}} L\big(s_N(y), s_F(y)\big)}{nbr(Y_{misb})} \quad (2)$$

where $nbr(Y_{misb})$ is the number of misbehaving observed variables. A misbehaving y is an observed variable that led to the fault detection. Its value in s_F better fit $\tilde{y}$ than its value in s_N. When relation 2 is satisfied, we say $L\big(s_N(v), s_F(v)\big)$ is deficient. The definition of L depends on the nature of the variables and the expression of the uncertainty in the model.

In the case variables domains are discrete, as in (Williams and Nayak 1996), variable instances have attached boolean labels. Misbehaving variables are observables labeled 1 in s_N and 0 in s_F. We set up $L\big(s_N(v), s_F(v)\big) = 1 - \big(lab(s_N(v)) - lab(s_F(v)\big)$, where lab returns the label of a given instance. This case also applies to the measure of mode deficiencies.

In case variables instances are numerical intervals, as in (Benazera and Travé-Massuyès 2003), a misbehaving observed variable o is such that $s_N(y) \cap \tilde{y} = \emptyset$. We use $L\big(s_N(v), s_F(v)\big) = s_N(v) \cap s_F(v)$.

In case a variable estimate is represented with a Gaussian law, as in (Hutter and Dearden 2003), we say y is misbehaving if $p(\tilde{y} \mid s_F)p\big(T(s_N, s_F)\big) \geq p(\tilde{y} \mid s_N)$, i.e. if its likelihood is higher in the diagnosed estimate than in the nominally predicted one, given the probability of changing mode. Here $p\big(T(s_N, s_F)\big) = p\big(s_N(\phi_1, \cdots, \phi_r)\big) \prod_{i=1,\cdots,r} p(\tau_i)$. Given that $s_N \sim \mathcal{N}(m_N, \theta_N)$ and $s_F \sim \mathcal{N}(m_F, \theta_F)$, we define L as the measure of the common space enclosed by both density functions f_N, f_F. Given v^1, v^2 the two intersection points of these curves, and considering that $\theta_F \geq \theta_N$ (otherwise, the notations are inversed):

$$L\big(s_N(v), s_F(v)\big) = \int_{-\infty}^{v^1} f_N(v)dv + \int_{v^1}^{v^2} f_F(v)dv + \int_{v^2}^{+\infty} f_N(v)dv \quad (3)$$

v^1, v^2 are solutions of $f_N(v) = f_F(v)$, that is a second degree polynomial for f_N, f_F Gaussian densities. In the general case, at the curves intersection points, the Mahalanobis metric $(v - m)'\theta^{-1}(v - m)$ of both estimates is identical.

Functional Deficiencies

Based on deficient variables, we now form the functional deficiencies.

Definition 2 (Functional deficiency). *A functional deficiency F for a model-based system A over a set of hybrid states $\mathcal{S} = \big(\mathcal{P}(A), \mathcal{D}(A)\big)$ is a set of variable instances of $M \cup X$ that are realized in some states of $\mathcal{P}(A)$, and that are* deficient *in some states of $\mathcal{D}(A)$.*

We denote as $S(F)$ the *reconfiguration set* associated to F, and that is such that:

$$\forall\, (s_N^p(v) = v^j) \in F,\, L\big(s_N^p(v), s_F^q(v)\big) \text{ deficient, then } (s_N^p, s_F^q) \in S(F) \quad (4)$$

F is said to be *complete* w.r.t. a reconfiguration set S' iff $S' = S(F)$. The complete F over $\mathcal{S}$ is unique. From now on we consider a functional deficiency to be complete when not explicitly mentioned otherwise. Also, we sometimes write a functional deficiency as the conjunction of its elements. The tuple $\big(F, S(F)\big)$ is denoted a *reconfiguration tuple*. Finally, it is possible to prioritize a functional deficiency[4]:

$$pr(F) = \sum_{i=1}^{n} \sum_{j=1}^{f} p(s_N^i)p(s_F^j),\, (s_N^i, s_F^j) \in S(F) \quad (5)$$

Definition 3 (Core functional deficiency). *The core functional deficiency F^c has its elements satisfied in* all *states of $\mathcal{P}(A)$ and* deficient *in* all *states of $\mathcal{D}(A)$. The core function is unique for a given set $\mathcal{S}$, and its priority is equal to* 1.[5]

Note that at least all misbehaving variables in states of $S(F)$ do belong to the core deficiency, as does $Q = 0$ in our example.

Minimal functionalities over maximal reconfiguration sets

This section develops a characterization of functional deficiencies whose size is minimal, while deficient over the largest number of state estimates. The reason is that the autonomous system certainly wants to operate minimal

[4]Note that in this expression, there is no notion of fault criticality. Every faulty state is assumed to have equal criticality but the probability of the state is taken into account.

[5]Given that $\mathcal{P}(A)$ and $\mathcal{D}(\mathcal{A})$ have their states probabilities summing to 1.

changes while covering the maximum states. We begin by characterizing a complete functional deficiency of minimal size.

Definition 4 (Minimal functional deficiency). *A functional deficiency F is minimal if it exists no functional deficiency F' such that $F' \subset F$ and $S(F') \subset S(F)$.*

We then characterize the maximal reconfiguration set.

Definition 5 (Maximal reconfiguration set). *A functional deficiency F has a maximal reconfiguration set $S(F)$ if it exists no other functional deficiency F' such that $S(F) \subset S(F')$ and $F' \subseteq F$.*

The search for *minimal* functional deficiencies over *maximal* reconfiguration sets leads to a set of functional deficiencies We denote as being *minimax*.

Proposition 1. *Given two minimax functional deficiencies F and F' such that $F' \cap F \neq \emptyset$, then $S(F') = S(F)$.*

Proof. If $F'' = F' \cap F$ and $F'' \neq \emptyset$, then $F'' \subseteq F$ and from definition 4, applied to F, it comes $S(F) \subset S(F'')$. And from definition 5, $S(F'') \subset S(F)$. It follows $S(F'') = S(F)$. Similarly, $S(F'') = S(F')$, so $S(F) = S(F')$. □

According to relation 4, the completeness of two functionalities F and F' implies that if $S(F) = S(F')$, then $F = F'$. The previous proposition implicitly focuses the search on *distinct minimax functions*. Thus functional deficiencies may be characterize as disjoints sets of variables instances. This result brings flexibility to the reconfiguration process, but is mitigated as the disjoints functions are not independent from each other w.r.t. the equations in E/the transitions in $\mathcal{T}$. In other words, they may not be recovered independently. In reference to the recovery (planning) operation, these functionalities are no serializable goals.

Proposition 2. *The core functional deficiency F^c is minimax.*

Proof. This is trivial from definitions 4 and 5. F^c is also complete with $S(F^c) = \mathcal{S}$. □

Functional Deficiencies Computation

To work on reconfiguration tuples, we define the intersection and the union of two tuples $\big(F_1, S(F_1)\big)$ and $\big(F_2, S(F_2)\big)$:

$$\big(F_1, S(F_1)\big) \cap \big(F_2, S(F_2)\big) = \big(F_1 \cap F_2, S(F_1) \cup S(F_2)\big) \quad (6)$$

and :

$$\big(F_1, S(F_1)\big) \cup \big(F_2, S(F_2)\big) = \big(F_1 \cup F_2, S(F_1) \cap S(F_2)\big) \quad (7)$$

We note $F_1 \cap F_2$, $F_1 \cap F_2$ for short.

The computation of the minimax functional deficiencies is performed with algorithm 1. Its main principle is to progressively reduce simple non-minimax deficiencies. The first step updates the deficiencies for each combination of two states of $\mathcal{S}$ using the measure of relation 2, and computes the core function. Iterating through this set, step 3 prunes out any deficiency of its intersection with F^c. Step 4 prunes out non-disjoints functionalities of their intersection. Step 5 merges the reconfiguration sets of similar deficiencies.

1: Compute the *complete* F w.r.t. each reconfiguration set (s_N^p, s_F^q), compute F^c, and add them all to the agenda.
2: Iterate through the tuples (F_i, F_j) in the agenda.
3: If $F^c \cap F_i \neq \emptyset$, $F_i \longleftarrow F_i \setminus \{F_i \cap F^c\}$.
4: Else if $F_i \cap F_j \neq \emptyset$, create a new function $F' = F_i \cap F_j$ and add it to the agenda. Do $F_i \longleftarrow F_i \setminus F'$.
5: Else if $F_i = F_j$, $S(F_i) = S(F_i) \cup S(F_j)$ and remove the remaining function F_j from the agenda.
6: F_i is minimax when it does not intersect with other functions anymore. It is removed to the agenda and returned.

Algorithm 1: Computing minimax functional deficiencies

A word on complexity: given p nominal and q faulty states, resulting in f minimax deficiencies, the first step finds $pq + 1$ complete functions. Studying the loop that starts at step 2, we consider an iteration checks all intersections among the F_i currently in the agenda. Noting n_j the number of intersection checks at iteration j, we have $n_j = \lambda_j \sum_{i=1}^{n_{j-1}-1} i$, with $\lambda_j = \frac{n_{j-1}}{n_{j-1}-e_j}$, and e_j is the number of functions eliminated (or added, e being negative). Noting $\lambda = \frac{1}{\xi}\sum_{j=1}^{\xi} \lambda_j$, where ξ is the total number of iterations, we write $\lambda \approx \frac{pq}{f}$. It appears that if $\mathcal{D}(A)$ is computed w.r.t. $\mathcal{P}(A)$, then in general $f = pq$. From that it comes $\xi \approx \sum_{j=1}^{\xi} \lambda_j$. Finally, the total number of computed intersections is around $\sum_{j=1}^{\xi} n_j$, with $n_0 = pq + 1$.

The algorithm is better understood by developing our example. Step 1 gives:

$$\begin{aligned}
s_N^1, s_F^1 &: F_1 = (V_1.m = open) \wedge Q_1 > 0 \wedge Q > 0 \\
s_N^1, s_F^2 &: F_2 = P_0 < P^* \wedge (S.m = open) \\
&\quad \wedge (V_2.m = closed) \wedge Q_1 > 0 \wedge Q > 0 \\
s_N^1, s_F^3 &: F_3 = P_0 < P^* \wedge (S.m = open) \\
&\quad \wedge Q_1 > 0 \wedge Q > 0 \\
s_N^2, s_F^1 &: F_4 = P_0 \geq P^* \wedge (S.m = closed) \\
&\quad \wedge (V_1.m = open) \wedge (V_2.m = open) \\
&\quad \wedge Q_1 > 0 \wedge Q_2 > 0 \wedge Q > 0 \\
s_N^2, s_F^2 &: F_5 = (V_1.m = open) \wedge (V_2.m = open) \\
&\quad \wedge Q_1 > 0 \wedge Q_2 > 0 \wedge Q > 0 \\
s_N^2, s_F^3 &: F_6 = (S.m = closed) \wedge (V_1.m = open) \\
&\quad \wedge Q_1 > 0 \wedge Q_2 > 0 \wedge Q > 0 \\
&\quad \wedge (V_2.m = open) \\
s_N^1, s_N^2, s_F^1, s_F^2, s_F^3 &: F^c = (V_1.m = open) \wedge Q_1 > 0 \wedge Q > 0
\end{aligned}$$

We have $F_1 = F^c$ so F_1 can be eliminated. Then reducing other functions with F^c:

$$\begin{aligned}
F_2 &= P_0 < P^* \wedge (S.m = open) \wedge (V_2.m = closed) \\
F_3 &= P_0 < P^* \wedge (S.m = open) \\
F_4 &= P_0 \geq P^* \wedge (S.m = closed) \wedge (V_2.m = open) \wedge Q_2 > 0 \\
F_5 &= (V_2.m = open) \wedge Q_2 > 0 \\
F_6 &= (S.m = closed) \wedge Q_2 > 0 \wedge (V_2.m = open)
\end{aligned}$$

1. $F_2 \cap F_3 = P_0 < P^* \wedge (S.m = open)$, $F_7 \longleftarrow P_0 < P^* \wedge (S.m = open)$, $S(F_7) = (s_N^1; s_F^2, s_F^3)$, $F_2 = F_2 \setminus$

$F_7 = (V_2.m = closed)$, $S(F_2) = (s_N^1; s_F^2)$. F_7 is added to the agenda.

2. $F_2 \cap F_4 = \emptyset$, $F_2 \cap F_5 = \emptyset$, $F_2 \cap F_6 = \emptyset$, and $F_2 = V_2.m = closed$ is minimax.
3. $F_3 \cap F_4 = \emptyset$, $F_3 \cap F_5 = \emptyset$, $F_3 \cap F_6 = \emptyset$, $F_3 = F_7$, remove F_7, $S(F_3) = (s_N^1; s_F^2, s_F^3)$. $F_3 = P_0 < P^* \wedge (S.m = open)$ is minimax.
4. $F_4 \cap F_5 = F_5$, $F_4 \longleftarrow F_4 \setminus F_5 = P_0 \geq P^* \wedge (S.m = closed)$, $S(F_4) = (s_N^2; s_F^1)$.
5. $F_4 \cap F_6 = (S.m = closed)$, $F_8 = (S.m = closed)$, $S(F_8) = (s_N^2; s_F^1, s_F^3)$, $F_4 \longleftarrow F_4 \setminus F_8 = P_0 \geq P^*$, $S(F_4) = (s_N^2; s_F^1)$, and F_4 is minimax.
6. $F_6 \cap F_5 = F_5$, $F_6 \longleftarrow F_6 \setminus F_5 = F_8$. Remove F_8, $F_6 = (S.m = closed)$, $S(F_6) = (s_N^2; s_F^1, s_F^3)$. F_5, F_6 are minimax.

Finally, the minimax functions are:

$F^c = (V_1.m = open) \wedge Q_1 > 0 \wedge Q > 0$, $S(F^c) = (s_N^1, s_N^2; s_F^1, s_F^2, s_F^3)$

$F_2 = (V_2.m = closed)$, $S(F_2) = (s_N^1; s_F^2)$

$F_3 = P_0 < P^* \wedge (S.m = open)$, $S(F_3) = (s_N^1; s_F^2, s_F^3)$

$F_4 = P_0 \geq P^*$, $S(F_4) = (s_N^2; s_F^1)$

$F_5 = (V_2.m = open) \wedge Q_2 > 0$, $S(F_5) = (s_N^2; s_F^2)$

$F_6 = (S.m = closed)$, $S(F_6) = (s_N^2; s_F^1, s_F^3)$

At this point, a possible extension to the functional deficiencies is to distinguish the *continuous reduction* of F_i, that is its reduction to variables in X, from the *hybrid* deficiency (made of both discrete and continuous instances). Intuitively, as the modes are relaxed, there exist more states that satisfy the continuous reduction to a deficiency, than the hybrid deficiency. For this reason, we say the latter leads to *reset* solutions (as modes deficiencies are explicitly set up to be recovered), as opposed to *redundancy* solutions (modes are unspecified, various components may be activated to recover the continuous deficiencies). We note $\bar{F}$ the continuous reduction to F.

Reconfiguration of Functional Deficiencies

This section focuses on reconfiguring a functional deficiency by identifying a set of goal states, and planning a recovery to those states. Ideally, a goal state specifies a value to all component modes, and may be inferred from the functional deficiency. In the case of a hybrid uncertain state however, the constraints in the form of continuous static/differential equations prevent a unique identification of the modes at a single point in time. Instead we propose to rely on an intrinsic property of hybrid systems, that is that the conditional statements ϕ naturally partition their behavioral space into small regions that we refer to as *configurations*. We refer the reader to (Benazera and Travé-Massuyès 2003) for one among the several formalizations of these regions. Identifying the regions that enclose the values of F^* is sufficient as to form goals that we refer to as *configuration goals* (instead of goal states). They correspond to reduced sets of both component modes and equalities/inequalities over continuous variables.

Then, we must ensure that the goals are reachable by both the continuous and discrete dynamics, respectively equations E and transitions T.

In the following, we denote as the *goal functional deficiency* F^* the functional deficiency to be recovered. Its selection is part of the recovery process. A simple F^* is F^c as its priority is maximal, and it covers all state estimates.

Configurations identification

We first enhance the model representation, then determines the goal configurations through a process similar to the consistency approach to model-based diagnosis. Indeed, reconfiguration can be viewed as the problem of identifying components whose reconfiguration is sufficient to restore acceptable behavior, when diagnosis is the problem of identifying components whose abnormality is sufficient to explain observed malfunctions (Crow and Rushby 1991).

Causal-graph of influences A first difficulty lies in equations in E that may demand a time-analysis for determining continuous variable values that are not set in F^*. A second problem lies in the non-existence of a bijection between modes in M and a particular continuous region of the state-space, as constrained by E. These problems can be tackled by first enhancing the model-based formalism with a causal representation of E.

Definition 6 (Causal-Graph of Influences). *The causal-graph of influences of a set of equations E is an oriented graph $G = (X, I)$ where the variables in X form a set of nodes x_i, and I a set of arcs among these variables.*

The causal-graph is a representation of relations among variables in E that holds at any time step. Its structure allows reasoning at a single point in time.

Definition 7 (Causal Influence). *A causal influence in I, $I_{i,j} = (x_i, x_j, b, \phi)$, is a directed arc between two variables x_i and x_j, with b the* sign *of the influence and ϕ its* activation condition.

Influences are drawn from the implicit causality in E. Variables that are subject to no influence are referred to as the *inputs* of G. Figure 2 pictures the causal-graph of the pressure regulator system. In the following we replace equations in E with G.

In general some work is required to extract the causality from static relations (Travé-Massuyès and Pons 1997). $b = \{-1, 1\}$ (1 includes equality) stores the numerical *positive*/*negative* influence among variables. ϕ's truth value in the hybrid state determines the *activation*/*deactivation* of the influence in the graph. Unconditioned, the influence is permanently activated. The activation conditions represent the causality changes in the dynamics.

Definition 8 (Configuration). *A configuration for A is of the form $\bigwedge_i \phi_i$.*

A configuration delimits a region of behavior of A. In our example, $V_1.m = open \wedge V_2.m = open \wedge P_0 \geq P^* \wedge P_0 \geq P_1 \wedge P_0 \geq P_2 \wedge S.m = closed$ is a nominal configuration of the system.

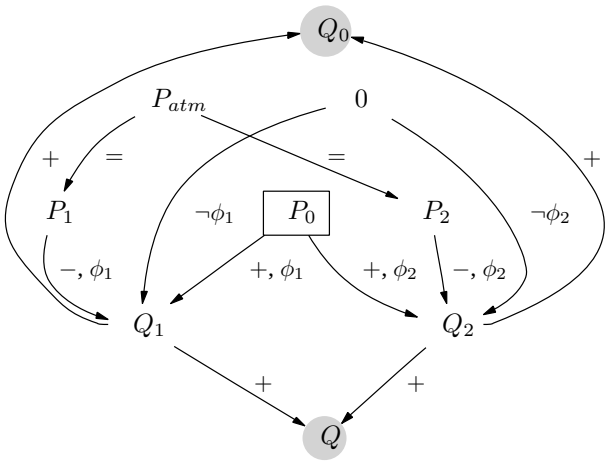


Figure 2: Pressure regulator causal-graph

Building goal configurations from reconfigurable functions We write the MBD theory based on consistency (Reiter 1987) where for the reconfiguration purpose, observations are replaced with functional deficiencies. A deficiency F_i has been characterized (min/max) w.r.t. the states uncertainty. We're now searching for the *minimal sets of conditions* that are sufficient to restore F_i.

Definition 9 (Reconfiguration candidate). *A reconfiguration candidate for A given F^* is defined as a minimal set $\Delta \subseteq I$ of influences such that*

$$A \cup F^* \cup \{\neg\phi_i \in \Delta\} \tag{8}$$

is consistent.

Definition 10 (Reconfiguration conflict). *A reconfiguration conflict for A given F^* is a set $\lambda = \{I_1, \cdots, I_k\}$ of influences such that*

$$A \cup F^* \cup \phi_1 \cup \cdots \cup \phi_k \tag{9}$$

is not consistent.

From $G \cup F^*$, we seek for reconfiguration conflicts in G that are such that influences in a conflict cannot be activated together given F^*. For a deficient variable (node) x_j of F^*, we call *ascending* influences the influences that belong to the paths from the inputs/other deficient variables, to x_j. An ascending influence for x_j is noted $\lambda_i^j = \{I_i, \phi_i\}$. A conflict for x_j is thus the set λ_j of its ascending influences $\{\lambda_i^j\}_{i=1,\cdots,n_j}$. $\Lambda = \left\{\{\lambda_j\}_{j=1,\cdots,n_{F^*}}\right\}$ is the collection of conflicts over all deficient variables of F^*. The minimal set of influences Δ that are candidates to the reconfiguration is obtained similarly to the diagnose in the MBD theory by computing the hitting sets (HS) over Λ (Reiter 1987). We note $\Delta_q = (\mathcal{I}_q, \wedge_{I_i \in \mathcal{I}} \phi_i)$ a diagnostic candidate, where $\mathcal{I}_q$ is a set of influences. Consequently, $\Delta = \left\{\{\Delta_q\}_{q=1,\cdots,n_q}\right\}$. We note $\neg\Delta = \left\{\{\neg\Delta_q\}_{q=1,\cdots,n_q} bigr\}$.

Consider our example again. Reconfiguring F^c with algorithm 2, it implies ϕ_1 is satisfied (step 1), and applying from $S_F(F^*)$, that $\neg\phi_2$ is satisfied (step 2). Activating influences

1: Apply F^* to G.
2: Apply $S_F(F^*)$ to $G \setminus F^*$.
3: Get the conflicts Λ.
4: Compute $\Delta = HS(\Lambda)$.
5: $\neg\Delta \wedge F^*$ are goal configurations.

Algorithm 2: Identifying reconfiguration candidates ($Goals$)

in the graph, it comes two sets of conflicts:

$$\begin{cases} \lambda_Q = \{Q \leftarrow Q_1, Q \leftarrow Q_2, Q_2 \overset{\neg\phi_2}{\leftarrow} 0, P_2 \leftarrow P_{atm}\} \\ \lambda_{Q_1} = \{Q_1 \overset{\phi_1}{\leftarrow} P_0, Q_1 \overset{\phi_1}{\leftarrow} P_1, P_1 \leftarrow P_{atm}\} \end{cases}$$

ϕ_1 is satisfied in F^c, and influences over Q, P_1 and P_2 are activated in all configurations, so it simplifies to:

$$\begin{cases} \lambda_Q = \{Q_2 \overset{\neg\phi_2}{\leftarrow} 0\} \\ \lambda_{Q_1} = \{\} \end{cases}, \Lambda = \{\lambda_Q, \lambda_{Q_1}\}$$

It comes $\Delta = \left\{\{\neg\phi_2\}\right\}$ and $\phi_2 \wedge F^c$ thus is a valid goal configuration (step 5).

Reconfiguring the continuous reduction $\bar{F}^c$ leads to more opportunities: ϕ_1 is no more satisfied and $\lambda_{Q_1} = \{\neg\phi_1\}$, thus $\Delta = \{\{\neg\phi_1, \neg\phi_2\}\}$ and goal configurations are given by $\phi_1 \wedge \phi_2 \wedge \bar{F}^c$.

Recovery

The recovery operation aims at bringing the system into the regions defined by the goal configurations. In our case, due to the hybrid dynamics, this process implies a chain of transitions exist to the component goal modes, while the continuous dynamics ensure the transition guards are successively satisfied. Sets of component transitions $T_0, \cdots, T_p$ must satisfy

$$A \cup \mathcal{D}(A) \cup T_0 \cup \cdots \cup T_p \cup F^* \cup \neg\Delta \tag{10}$$

is consistent, where we the current time of the system is set to 0 and the initial states belong to $\mathcal{D}(A)$. $Pl = \{T_0, \cdots, T_p\}$ is a *plan* for the recovery. Noting k_p the time transition T_p triggers, the continuous dynamics must satisfy

$$\begin{cases} X(0) \cup \phi_0 \\ E(X(0)) \cup \phi_1 \\ E(X(k_1)) \cup \phi_2 \\ \vdots \\ E(X(k_{p-1})) \cup \phi_p \cup F^* \end{cases} \tag{11}$$

are consistent, where $E(X(k_j))$ refers to the dynamics of relation (1), is conditioned by ϕ_{j+1}, and $X(0) = \sum_{s_F^i \in \mathcal{D}(A)} p(s_F^i) X_F^i(0)$. We say relations (10) and (11) define a *hybrid system planning* problem. To our knowledge, the planning of hybrid systems has received no attention yet. We believe that its development will be made necessary by several on-line applications.

Relation (10) poses a probabilistic conformant planning problem (Hyafil and Bacchus 2003), where a set of transitions must bring the system to a set of predetermined goals, under uncertainty and without observing the system state.

The plan maximizes the probability of the goal configuration given the initial belief state $\mathcal{D}(A)$. In our example, a stuck valve can't be re-opened, so no plan exists for functionalities F^c and $\bar{F}^c$. A plan exists to F_5 for some initial states, $Pl = \{\tau_3, \tau_1\}$. F_6 has a plan $Pl = \{\tau_3\}$.

Relation (11) poses a control problem where the continuous dynamics must be forced to successive ϕ_j through available inputs. A model predictive control problem (MPC) solves on-line a finite horizon open-loop optimal control problem subject to system dynamics and constraints involving states and controls. Based on measurements obtained at time k, the future dynamic behavior of the system is predicted over a fixed horizon, and the controller determines the input such that a performance criteria is optimized. This technique fits well within the model-based autonomous system framework, given two key elements are already present, the model A, and the state predictor (or estimator) $\mathcal{P}(A)$. By using control and measurement horizons of a single time step, a basic formulation of the MPC problem at time k is

$$\begin{aligned} U^*(k+1) &= \min_U J(X(k), U(k)) \\ J(X(k), U(k)) &= \int_k^{k+1} F(X(t), U(t))dt \\ F(X, U) &= (X - X_s)^T Q(X - X_s) \\ &\quad +(U - U_s)^T R(U - U_s) \\ X(k+1) &= f(X(k), U^*(k)) \\ 0 &\leq h(X(k), U(k)) \end{aligned}$$

where Q and R denote positive definite symmetric weighting matrices, and $U^*(k+1)$ is the optimal input used in the prediction at $k+1$. Considering ϕ over X is in the form ϕ: $l(X) \geq 0$, we note $\bar{\phi}$: $\bar{l}(X) + \epsilon = 0$ its reduction to an equality, where ϵ is a term that will ensure the threshold is later satisfied. The function is evaluated at k with $\bar{\phi}(k)$: $\bar{l}(X(k)) + \epsilon$, and we note its inverse $\bar{\phi}^{-1}(k)$. The MPC application to the control objective ϕ_j sets the setting point (X_s, U_s) to $(\bar{\phi}_j^{-1}(k), 0)$. In our example, τ_3's guard gives $\bar{\phi}_{\tau_3}^{-1}(k) = P^* + \epsilon'$.

Again, we're confronted to the fact that $\mathcal{P}(A)(k) = \{s^1, \cdots, s^q\}$ likely contains multiple state estimates. Thus the minimization must apply to each $F(X^i(k), U(k))$, returning $U^{*,i}(k+1)$. We merge the optimized input candidates according to the states estimated probabilities:

$$U^*(k+1) = \sum_{i=1,\cdots,q} p(X^i(k)) U^{*,i}(k+1) \qquad (12)$$

Finally, when ϕ_j is reached, transition T_p should trigger, and MPC then focuses on ϕ_{j+1}. The last MPC set-point is F^*.

This control problem however requires more research. First, the MPC community itself seeks for better state estimation integration within the loop (Morari and Lee 1997). Second, ϕ's inverse is problematic in practice. The control could focus on bringing the system state back to the geometrical center of the goal configuration region instead. This is yet to be explored. Third, optimality and especially, stability problems, if far out of the scope of this paper, must be tackled in the case of control based on multiple state estimates. Modern hybrid state estimators should be coupled with powerful techniques such as Quasi-Infinite Horizon NMPC (Chen and Allgwer 1998). Note that recent developments also pave the way for powerful stability and safety/reachability analysis of these controllers (Bemporad *et al.* 2001).

Reaching the goals: safety and convergence

Considering the context of a faulty system, the reconfiguration process should likely be safe, not making the situation worse. In our case, the goal configurations identification may produce multiple solutions, while not ensuring that any of them are reachable in the end. In this section we improve algorithm 2 by reducing the number of goal solutions while ensuring they are reachable under monotonous continuous dynamics. To ensure the latter, and given a variable $v \in F^*$, the sign of $(S_N(v) - S_F(v))$ is studied, where (S_N, S_F) is the reconfiguration set of F^*. Algorithm 2 is modified such that Λ becomes Λ^-, *the set of influences to be deactivated*, while Λ^+, *the set of influences to be activated* is constructed as follows:

- Given a path of ascending influences $\{I_{i,i_1}, \cdots, I_{i_n,j}\}$ from x_i to $x_j \in F^*$, if $x_i\big(S_N(x_j) - S_F(x_j)\big) \prod_{k=i_1,\cdots,i_n} b_k > 0$, then for all ϕ_k that is not satisfied, add $I_{i_k,i_{k+1}}$ to Λ^+.
- Otherwise, if ϕ_k is satisfied, add $I_{i_k,i_{k+1}}$ to Λ^-.

This corresponds to activating any ascendant path whose combined influences have a beneficial effect on the restoration of F^*. The approach is conservative as the test equality to 0 is not considered.

1: Apply F^* to G.
2: Apply $S_F(F^*)$ to $G \setminus F^*$.
3: Get the conflicts Λ^+, Λ^-.
4: Compute $\Delta^+ = HS(\Lambda^+)$ and $\Delta^- = HS(\Lambda^-)$.
5: Do $\Delta = \Delta^+ \bigotimes \neg\Delta^-$ and eliminate inconsistent configurations.
6: $\Delta \wedge F^*$ are goal configurations.

Algorithm 3: Identifying reconfiguration candidates ($SafeGoals$)

Back to our example, we reconfigure $\bar{F}_5 = Q_2 > 0$. Step 3 of algorithm 3 gives $\lambda^+_{Q_2} = \{Q_2 \overset{\phi_2}{\leftarrow} P_0\}$, $\lambda^-_{Q_2} = \{Q_2 \overset{\neg\phi_2}{\leftarrow} 0\}$, thus $\Delta^+ = \{\{\phi_2\}\}$, $\Delta^- = \{\{\neg\phi_2\}\}$. The solution is the same as returned by algorithm 2 but it is now ensured that opening V_2 brings the flow back into the right direction.

The safety may not be ensured when negative and positive effects to a variable are activated via the same condition, as over Q_2 in our example. If P_{atm} was not considered being a constant, a numerical analysis would have been required here.

Reconfiguring the Functional Deficiencies

Our general strategy to the reconfiguration of the functional deficiencies explores *reset* solutions first, then *redundancy* solutions (continuous reductions) in prioritized order. A

plan failure selects the next deficiency. Algorithm 4 sums up the process.

1: Compute functional deficiencies with algorithm 1
2: Identify goal configurations with algorithm 2 or 3.
3: Find a plan, in case of failure move to the next functionality, in prioritized order.
4: Apply MPC using $\mathcal{P}(A)$ as the predictor.

Algorithm 4: Reconfiguration of functional deficiencies

In our example, s_F^2 and s_F^3 have much lower probability than s_F^1 as they correspond to double faults. F^c is subject to plan failure. F_6: $S.m = closed$ is its own goal configuration and has a plan τ_3 which guard is $P_0 \geq P^*$. MPC generates the pressure input P_0 to reach that level. Note that depending on the real initial state, the reconfiguration may have no effect. The operation does not harm the system though (we consider maintaining a nominal level of pressure does not harm the system even when in a faulty state), and may help discriminate among the estimates. For example, if reconfiguring F_6 fails, s_F^1, and potentially s_F^2 are eliminated.

Summary, Existing works and Perspectives

We've presented a methodology to the automated reconfiguration of functional deficiencies. The deficiencies are identified by comparing predicted and diagnosed states, and then partitioned and prioritized over the state estimates. Goals are further identified from the deficiencies. Planning and MPC techniques are used in common to move the system toward the goals.

To our knowledge, automated MBR has not received a large attention. A pioneer work, (Crow and Rushby 1991), explores the analogy between the problems of diagnosis and reconfiguration. However, the approach does not deal with state uncertainty and provides no integration within a model-based loop. Goal identification and safe planning to the objectives have been studied in (Williams and Nayak 1997) in the case of qualitative models. We're not aware of any work on the planning of hybrid systems.

We hope making some improvements to the current approach in a near future. The $SafeGoal$ algorithm could be enhanced to tackle more complex dynamics. We also would like to participate to the integration of modern hybrid state estimator/diagnoser with non-linear MPC techniques. A priority is to explore the planning of hybrid systems and to search for stability and reachability results. Finally, we're considering a better integration of the functional deficiencies selection within the plan generation to reduce the loop over plan failures by using contingency branches (Meuleau and Smith 2003) instead of a mere probabilistic conformant planning.

References

A. Bemporad, W.P.M.H. Heemels, and B. De Schutter. On hybrid systems and closed-loop mpc systems. In *Proceedings of the 40th IEEE Conference on Decision and Control, Orlando, Florida, USA*, December 2001.

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

H. Chen and F. Allgwer. A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. *Automatica*, 34(10), 1998.

J. Crow and J. Rushby. Model-based reconfiguration: toward an integration with diagnosis. In *Proceedings of AAAI-91, Anaheim, CA*, volume 2, pages 836–841, 1991.

W. Hamscher, L. Console, and J. De Kleer. *Readings in Model-Based Diagnosis*. Morgan Kaufmann, San Mateo, CA, 1992.

M. Hofbaur and B.C. Williams. Mode estimation of probabilistic hybrid systems. *Hybrid Systems: Computation and Control, Lecture Notes in Computer Science (HSCC 2002)*, 2289:253–266, 2002.

F. Hutter and R. Dearden. The gaussian particle filter for diagnosis of non-linear systems. In *Proceedings of the Thirteenth International Workshop on Principles of Diagnosis DX-03*, 2003.

N. Hyafil and F. Bacchus. Conformant probabilistic planning via csps. In *Proceedings of the Thirteenth International Conference on Automated Planning and Scheduling (ICAPS 03)*, 2003.

N. Meuleau and D. E. Smith. Optimal limited contingency planning. In *Proceedings of the Thirteenth International Conference on Automated Planning and Scheduling (ICAPS 03)*, 2003.

M. Morari and J. H. Lee. Model predictive control: past, present, future. In *Joint 6th International Symposium on Process Systems Engineering (PSE'97)*, 1997.

N. Muscettola, P. Pandurang Nayak, Brian C. Williams, and B. Pell. Remote agent : To boldly go where no ai system has gone before. *Artificial Intelligence*, 103:5–47, 1998.

P. Nayak and J. Kurien. Back to the future for consistency-based trajectory tracking. In *Proceedings of AAAI-2000, Austin, Texas*, 2000.

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

M. Stumptner and F. Wotawa. Reconfiguration using model-based diagnosis. In *Proceedings of the Tenth International Workshop on Principles of Diagnosis DX-99*, 1999.

L. Travé-Massuyès and R. Pons. Causal ordering for multiple modes systems. In *Proceedings of the Eleventh International Workshop on Qualitative Reasoning*, pages 203 – 214, 1997.

B. C. Williams and P. Nayak. A model-based approach to reactive self-configuring systems. In *Proceedings of AAAI-96, Portland, Oregon*, pages 971–978, 1996.

B. C. Williams and P. Nayak. A reactive planner for a model-based executive. In *Proceedings of IJCAI-97*, 1997.