Sequential Scheduling of Observations in Diagnosis of Continuous Dynamic Systems

Roberto Zanotti[1] and **Alexander Feldman**[2] and **Marina Zanella**[1] and **Johan de Kleer**[2]
[1]Department of Information Engineering, University of Brescia, Brescia, Italy
e-mail: {r.zanotti003,marina.zanella}@unibs.it
[2]Palo Alto Research Center, Palo Alto, CA, USA
e-mail: {afeldman,dekleer}@parc.com

Abstract

A fault in a continuous dynamic system is a non nominal value of a physical parameter. Parameter values can usually be estimated with uncertainty given the values of the output variables of the system that have been observed over a time interval. A diagnosis consists of several probability density functions, one for each parameter. In order to reduce the uncertainty about the diagnosis, new observations are needed. However, in some contexts observations are expensive, thus there is a maximum number of observations that can be gathered. This paper faces the problem of choosing the best schedule of a limited amount of observations within a given time horizon, in order to minimize the uncertainty of the diagnosis. Several scheduling policies have been tried in order to experimentally compare them.

1 Introduction

Model based diagnosis of continuous dynamic systems is a challenging task [Hamscher *et al.*, 1992], and it is still more so in case little or no information about their internal state is available. The behavior of such systems over time can be described by means of Ordinary Differential Equations (ODEs). The coefficients of these equations are the physical parameters of the considered system, each of which can take a continuous value over a finite interval. If all such parameters take their nominal value, the behavior is normal; when the value of a parameter is not nominal, the actual behavior of the system differs from the expected one, that is, the system is faulty. Hence, a fault is a non nominal value of a parameter. However, based on the observations, one cannot infer the current value of each parameter; instead, one can compute a probability density function inherent to it. A diagnosis consists of several probability density functions, one for each parameter. Such a diagnosis may be noisy, especially if the considered system is usually affected by very few faults at the same time. In order to reduce the uncertainty about the diagnosis, new observations are needed. However, in some contexts observations are expensive, thus there is a maximum number of observations that can be carried out. The reduction in uncertainty is affected not only by the number of observations, but also by their temporal location. Thus, given that the number of affordable observations is fixed, what makes the difference is the time when the observations are performed. This paper faces the problem of choosing the best schedule of a limited amount of observations within a given time horizon, in order to minimize the uncertainty of the diagnosis. In particular, a single fault is assumed, and the (constant) length of the window within which to place the observations is such that no physical parameter can change its value over it.

2 Preliminaries

In this paper we address the problem of diagnosing dynamic systems, in particular, systems about which we have little or no information related to the internal state. The considered model of a dynamic system is a set of ODEs, as it is usually the case in Fault Detection and Isolation (FDI) methods, where a central concept is that of *residual* [Travé-Massuyès, 2014]. ODEs are supported by the Modelica language [Otter M., 2007], and consequently by the tools based on it. We focus on how scheduling can be used to improve the results of a diagnostic algorithm.

2.1 Running Example

In the remainder of this paper we use (a slightly modified version of) the Lotka-Volterra equations [Lotka, 1910], which are a pair of nonlinear, first order differential equations that model the interaction between two different species, a predator and a prey, within a biological system. We are particularly interested in diagnosing dynamic systems whose normal behavior is oscillatory, and the Lotka-Volterra model falls into this category.

2.2 Basic Definitions

In this section we provide the notions that will be exploited by the proposed approach.

Definition 1 (System Model)**.** *A system model M is a tuple $\langle \Gamma, I, OBS, COMPS \rangle$, where Γ is a set of ODEs, defined over a collection of variables that consists in I, which is the set of inputs, OBS, which is the set of outputs, and COMPS, which is the set of parameters.*

In the running example, Γ consists of the differential equations here below, while I is a single input whose value over time is given by *environment function* $\theta(t)$.

$$\frac{dx(t)}{dt} = p_1 x(t)\theta(t) - p_2 x(t) y(t),$$
$$\frac{dy(t)}{dt} = p_3 y(t) x(t) - p_4 y(t).$$

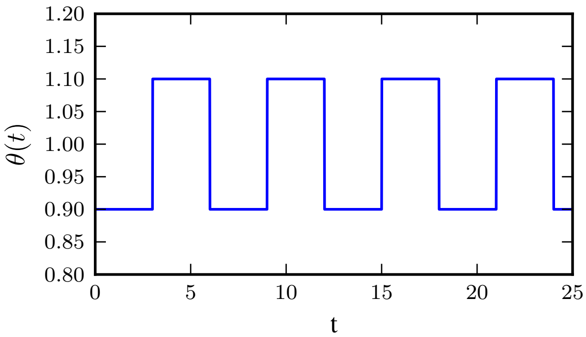

Figure 1: Environment function θ

Functions $x(t)$ and $y(t)$ constitute the outputs of the system; they represent the number of preys and the number of predators over time, respectively.
Set COMPS $= \{p_1, p_2, p_3, p_4\}$ contains the parameters, where p_1 (p_3) is the prey (predator) growth rate, and p_2 (p_4) is the prey (predator) death rate. Each parameter p_i in COMPS has a nominal value, denoted ω_i^*. In our example ω_1^* = 1.8, ω_2^* = 0.52, ω_3^* = 0.4, and ω_4^* = 0.4.
The environment function represents the effect of the season cycle on the prey growth rate. We assume to be able to predict the value of such a function over time, albeit we have no control over it. The predicted value of $\theta(t)$ has been used in the experiments in order to forecast the values of the output variables in OBS based on the equations in Γ.

In the running example, the adopted environment function is defined as follows:

$$\theta(t) = \begin{cases} 0.9 & \text{if } 0 + nk \leq t \leq \frac{k}{2} + nk \\ 1.1 & \text{otherwise} \end{cases}$$

where k is the period of the function, and $n \in \mathbb{N}$. Note that, given this specific $\theta(t)$, $x(t)$ and $y(t)$ are still oscillatory but not periodic. In Figure 1 an environment function with $k = 6$ is shown.

Definition 2 (Observation)**.** *Given a system model M, an observation is an assignment to the (output) variables in OBS at a time instant.*

Definition 3 (Fault Injection)**.** *Given a system model M, a fault injection is an assignment to parameters p in COMPS such that not all parameters are set to their nominal values. Specifically, a Single Fault Injection is an assignment where all the parameters but one take their nominal values.*

Definition 4 (Diagnosis)**.** *Let M be a system model, $\bar{I}[0,t]$ be an assignment relevant to the (input) variables in I over time interval [0, t], and $\bar{O}[0,t]$ be an assignment relevant to the (output) variables in OBS in the same time interval, where $\bar{O}[0,t]$ includes the values of the output variables at time t at least. A diagnosis $\Omega(M, \bar{I}[0,t], \bar{O}[0,t])$ is a set of probability density functions, one for each $p \in$ COMPS, where the probability density functions in Ω are compatible with both the observed values and the given inputs, based on the equations in Γ.*

The definition of diagnosis does not mention the state of system M at time 0, since it is assumed to be unknown. The probability density function for parameter p included in diagnosis Ω is denoted as Ω^p. In the following, it is assumed that Ω^p does not give the probability value over a continuous domain, instead it gives it over a finite discretized domain (a value for each subinterval the domain is split into). With a slight abuse of notation, $|\Omega^p|$ denotes the finite number of discrete values of parameter p encompassed by positive values of the probability density function Ω^p.

Definition 5 (Uncertainty Metrics)**.** *The uncertainty of a diagnosis is estimated by a function $U : \{\Omega\} \to [0,1]$, that maps each diagnosis Ω into a real value ranging from 0 (no uncertainty) to 1 (full uncertainty).*

In the context described above, the traditional entropy function [de Kleer and Williams, 1987] becomes:

$$U(\Omega) = -\frac{1}{|\text{COMPS}|} \sum_{p \in \text{COMPS}} \sum_{\omega \in \Omega^p} \pi(p,\omega) log_{|\Omega^p|} \pi(p,\omega),$$

where $\pi(p, \omega)$ is the probability that parameter p takes value ω.
Let us assume that a diagnosis Ω has already been found out at time t (it is beyond the scope of this paper to illustrate the methods to compute a diagnosis as of Definition 4; Algorithm 3 is the one we have adopted in our experiments). If such a diagnosis is uncertain, that is, if it does not point out that the system is affected by 0 or 1 fault, we have to perform some additional observations to reduce the uncertainty. Let us assume that only a limited amount of observations can be taken, which have to be evenly distributed within a time window of prefixed length τ, and that such time window has to be placed within a given time horizon, following t, this being $]t, t_{end}]$.

Definition 6 (Diagnostic Experiment)**.** *Given a system model M, an assignment $\bar{I}[0, t_{init} + \tau]$ relevant to the input variables in I, and two assignments, $\bar{O}[0,t]$ and $\bar{O}[t_{init}, t_{init} + \tau]$, relevant to the output variables in OBS over a pair of intervals, the latter being a time window, where $t_{init} > t$ and $t_{init} + \tau \leq t_{end}$, a diagnostic experiment consists in computing diagnosis $\Omega(M, \bar{I}[0, t_{init} + \tau], \bar{O}[0,t] \cup \bar{O}[t_{init}, t_{init} + \tau])$.*

We would like the uncertainty inherent to diagnosis $\Omega(M, \bar{I}[0, t_{init} + \tau], \bar{O}[0,t] \cup \bar{O}[t_{init}, t_{init} + \tau])$ to be lower than that inherent to $\Omega(M, \bar{I}[0,t], \bar{O}[0,t])$. In order to minimize the uncertainty relevant to the latter diagnosis, we have to perform an optimal scheduling of the time window within which the additional observations are gathered.

Definition 7 (Optimal scheduling problem)**.** *Given a system model M, and an uncertainty function U, the solution of an optimal scheduling problem is the best placement t_{init} of the time window in which to perform a diagnostic experiment $\Omega(M, \bar{I}[0, t_{init} + \tau], \bar{O}[0,t] \cup \bar{O}[t_{init}, t_{init} + \tau])$, in order to minimize its uncertainty $U(\Omega(M, \bar{I}[0, t_{init} + \tau], \bar{O}[0,t] \cup \bar{O}[t_{init}, t_{init} + \tau]))$.*

Solving an optimal scheduling problem relies on a forecast of the observations within each considered time window. Since the diagnosis $\Omega(M, \bar{I}[0,t], \bar{O}[0,t])$ at time t is available, this knowledge is exploited in order to forecast $\bar{O}[t_{init}, t_{init} + \tau])$. Such observation is used to 'simulate' a diagnostic experiment. This way, several diagnostic experiments, one for each considered time window, are simulated, each yielding a distinct diagnosis: the uncertainties of such diagnoses are compared so as to find out the time window that leads to the lower (estimated) uncertainty.

3 Algorithm

In this section we describe the algorithm for the optimal scheduling of the diagnostic experiment. Given a system model M, an existing diagnosis $\Omega_0 = \Omega(M, \bar{I}[0,t], \bar{O}[0,t])$ (which can be computed by Algorithm 3, based on the actual values of the outputs $\bar{O}[0,t]$), and a scheduling policy $\mathcal{P}$, function SCHEDULE computes the best time window placement in which to carry out the diagnostic experiment in order to minimize the diagnostic uncertainty. The scheduling policy $\mathcal{P}$ is aimed at selecting the values of t_{init} to try, and at deciding when to stop the (finite number of) trials.

Algorithm 1 Schedule a diagnostic experiment

Input: $\mathcal{M}$, system model
Ω_0, initial diagnosis
$\mathcal{P}$, scheduling policy
Returns: t_{init}, initial time of the window

1: **procedure** SCHEDULE($\mathcal{M}$, Ω_0, $\mathcal{P}$)
2: $\quad uncertainty \leftarrow 1$
3: $\quad$ **repeat**
4: $\quad\quad t_c \leftarrow$ SELECTTIME($\mathcal{P}$)
5: $\quad\quad \eta \leftarrow 0$
6: $\quad\quad$ **for all** $p \in$ COMPS **do**
7: $\quad\quad\quad \bar{I}[0, t_c + \tau] \leftarrow$ PREDICTINPUT(t_c)
8: $\quad\quad\quad \bar{O}[t_c, t_c+\tau] \leftarrow$ PREDICTOBS($\mathcal{M}, \Omega_0^p, p, t_c$)
9: $\quad\quad\quad \Omega_{(p,t_c)} \leftarrow$ DIAG($\mathcal{M}, \bar{I}[0, t_c + \tau], \bar{O}[0, t] \cup \bar{O}[t_c, t_c + \tau]$)
10: $\quad\quad\quad \eta \leftarrow \eta + \frac{1}{|COMPS|}$EVALUATE($\Omega_{(p,t_c)}$)
11: $\quad\quad$ **end for**
12: $\quad\quad$ **if** $\eta < uncertainty$ **then**
13: $\quad\quad\quad uncertainty \leftarrow \eta$
14: $\quad\quad\quad t_{init} \leftarrow t_c$
15: $\quad\quad$ **end if**
16: $\quad$ **until** TERMINATE($\mathcal{P}$)
17: $\quad$ **return** t_{init}
18: **end procedure**

Algorithm 1 works as follows: the estimated uncertainty of the diagnosis that will be computed based on the scheduled observations is initially set to its maximum value (1) at line 2. The main loop starts with the selection of a time stamp (line 4) by function SELECTTIME, based on policy $\mathcal{P}$. Function SELECTTIME is assumed to be stateful, which means that the selection of a time stamp depends on the previously chosen time stamps. The current iteration of the main loop is meant to evaluate the uncertainty of the diagnosis in case the time window within which the observations are gathered starts at time t_c. Such uncertainty (variable η), which is initially set to its minimum value (0) at line 5, is progressively updated by the iterations of the nested loop (lines 6-11). In fact, since the single fault assumption holds, but we do not know which is the faulty component, we have to take into account each single component and find a placement of the time window that can provide some help in disambiguating the diagnosis whichever is the fault. Hence, each iteration of the loop at line 6 assumes that component p takes the value having the highest probability in Ω_0^p (be it nominal or faulty), while the remaining components take their nominal values. Then, at line 7, it predicts the inputs from time 0 till the end of the considered window by calling function PREDICTINPUT (that, in the experiments computes $\theta(t)$ for t ranging in interval $[0, t_c+\tau]$), it predicts the observation relevant to the time window by invoking Algorithm 2 (line 8), computes the diagnosis, called $\Omega_{(p,t_c)}$, by invoking Algorithm 3 (line 9), evaluates the uncertainty of $\Omega_{(p,t_c)}$ by calling function EVALUATE, and combines this uncertainty with the current uncertainty value (line 10). The global uncertainty relevant to the choice of the time window starting at t_c is then compared (line 12) with the best value of the uncertainty relevant to all the time windows considered so far. If η is better, then both *uncertainty* and t_{init} are updated straightforwardly (lines 13-14). When the main loop stops according to policy $\mathcal{P}$, value t_{init} is returned (line 17).

Algorithm 2 Predict observations

Input: $\mathcal{M}$, system model
Ω, diagnosis
p, the only (possibly) faulty parameter
t_s, initial time of the window
Returns: $\bar{O}[t_s, t_s + \tau]$, the predicted observations

1: **procedure** PREDICTOBS($\mathcal{M}$,Ω, p, t_s)
2: $\quad max\_probability \leftarrow 0$
3: $\quad \omega_s \leftarrow 0$
4: $\quad$ **for all** $\omega \in \Omega^p$ **do**
5: $\quad\quad$ **if** $\pi(\omega) > max\_probability$ **then**
6: $\quad\quad\quad max\_probability \leftarrow \pi(\omega)$
7: $\quad\quad\quad \omega_s \leftarrow \omega$
8: $\quad\quad$ **end if**
9: $\quad$ **end for**
10: $\quad \bar{O}[t_s, t_s + \tau] \leftarrow$SIMULATE($\mathcal{M}$, p, ω_s, t_s)
11: $\quad$ **return** $\bar{O}[t_s, t_s + \tau]$
12: **end procedure**

Algorithm 2 is aimed at predicting the observations of system M within a time window, when parameter p takes the value with the highest probability in Ω^p. It starts by finding the value ω_s of parameter p with the highest probability according to the probability density function Ω^p (lines 2-9). Then, function SIMULATE is called (line 10). This function takes as an input model M, parameter p, the value ω_s of this parameter, the initial time of the window t_s. The function simulates system M using the nominal values of each parameter in COMPS, except for parameter p that takes value ω_s. It returns a set of h observations, evenly distributed in interval $[t_s, t_s + \tau]$, where h is the constant number of observations within a time window.

Algorithm 3 computes a diagnosis, given model M and its inputs and outputs, the latter possibly inherent to several time intervals included in the only interval relevant to the former. At each iteration of the main loop (lines 2-11) a probability density function for a parameter p is computed by first dividing the domain $[LB^p, UB^p]$ of the parameter itself into a constant number i of subintervals. At line 3 the size v of these subintervals is computed. Then, for each subinterval, a lower bound LBT and an upper bound UBT is determined (lines 5 and 6). These pair of values are the inputs of function MINRES (line 7), along with model M, parameter p, and the observation set α. This function determines which value of parameter p, within interval $[LBT, UBT]$, minimizes the square difference between the observations in α and the values of the variables in OBS obtained by simulating the model with the nominal values of every parameter, except for p, that can take any value

Algorithm 3 Compute a diagnosis

Input: $\mathcal{M}$, system model
$\bar{I}[0, t_f]$, inputs
α, set of observations within $[0, t_f]$
Returns: diagnosis $\Omega(\mathcal{M}, \bar{I}[0, t_f], \alpha)$

1: **procedure** DIAG($\mathcal{M}, \bar{I}[0, t_f], \alpha$)
2: **for all** $p \in COMPS$ **do**
3: $v \leftarrow \frac{UB^p - LB^p}{i}$
4: **for** $j \leftarrow 0$ **to** $i - 1$ **do**
5: $LBT \leftarrow LB^p + v \cdot j$
6: $UBT \leftarrow LB^p + v \cdot (j + 1)$
7: $(p^*, \rho) \leftarrow$ MINRES($\mathcal{M}, p, LBT, UBT, \alpha$)
8: $T^j \leftarrow (p^*, \rho)$
9: **end for**
10: $\Omega^p \leftarrow$ BUILD_PDF(T^p)
11: **end for**
12: **return** Ω
13: **end procedure**

between LBT and UBT. The function returns the square difference ρ, called *residual*, and p^*, the value of p to which that residual corresponds. Pair (ρ, p^*) is then stored in T^j. Once all the subintervals for a parameter domain have been considered, function BUILDPDF is called (line 10). This function takes as an input the set T of pairs (ρ, p^*), and returns a probability density function for parameter p. When a probability density function has been computed for each and every parameter, diagnosis Ω is returned (line 12).

4 Experimental Results

The approach described in the previous sections was implemented in a Python 2.7 program, and several experiments were run in a Windows 10 environment on an Intel i7-4700HQ 2.4 GHz platform endowed with 16 GB RAM. Three scheduling policies were considered: $\mathcal{P}_{rand}$, which randomly chooses the time windows to be tried; $\mathcal{P}_{seq}$, which tries windows whose t_{init} is placed at a fixed distance with respect to the initial time of the previous window (this way, windows may overlay); and $\mathcal{P}_{jump}$, according to which the distance between the initial times of the current window and that of the next one depends on the estimated uncertainty of the diagnosis relevant to the present window (the higher the uncertainty, the longer the distance).

The runtime of Algorithm 1 is assumed to be negligible with respect to the length of a time window, which in the experiments is set to 10 time units, while the time horizon is set to 100. In the experiments, the domain of each parameter is $[0, 2]$, while the values of constants h and i in Algorithm 3 are 10 and 4, respectively.

Some experimental evidence is recorded in Table 1, where each row is relevant to a distinct test. The 8 tests encompassed by the table have been generated by injecting two different faults in each component (specifically, one abnormal value of the parameter that differs from the nominal one for an absolute value equal to 0.1, and the other that differs for an absolute value of 1.0). The column relevant to each scheduling policy records the achieved uncertainty reduction, which is computed as $U(\Omega(M, \bar{I}[0, t], \bar{O}[0, t])) - U(\Omega(M, \bar{I}[0, t_{init} + \tau], \bar{O}[0, t] \cup \bar{O}[t_{init}, t_{init} + \tau]))$. Bold type indicates the best uncertainty reduction achieved for each test case.

The experimental results show that a better uncertainty reduction can be obtained through the two non-random policies ($\mathcal{P}_{seq}$ and $\mathcal{P}_{jump}$). In some cases, the uncertainty of the diagnosis computed using the additional observations chosen according to policy $\mathcal{P}_{rand}$ is higher than the uncertainty of the initial diagnosis.

$\mathcal{P}_{rand}$	$\mathcal{P}_{seq}$	$\mathcal{P}_{jump}$
-0.003	**0.253**	0.169
0.153	0.196	**0.226**
-0.16	-0.03	**0.114**
0.081	**0.17**	0.115
0.117	0.08	**0.125**
0.057	**0.149**	0.07
-0.022	0.042	**0.151**
-0.04	**0.062**	0.035

Table 1: Uncertainty reduction for 8 tests

Neither of the non-random policies $\mathcal{P}_{seq}$ and $\mathcal{P}_{jump}$ is clearly better than the other; in fact, in four test cases a more substantial uncertainty reduction is achieved using policy $\mathcal{P}_{seq}$ and in the other four test cases the better result is obtained through policy $\mathcal{P}_{jump}$. It looks like P_{jump} is more suitable when the difference of the value of the faulty parameter is 0.1 (that is, it is lower), however, this interleaving of results needs a deeper investigation.

5 Conclusions

This paper has presented some preliminary ideas about observation scheduling in diagnosis of continuous dynamic systems. The experimental activity is ongoing. All the tests related in the previous section use the same value of t and τ, and the same time horizon. In future tests such values will be changed. The only figure of the effectiveness achieved by a scheduling policy shown in Table 1 is the reduction in diagnosis uncertainty. A still more accurate estimate is the difference $U(\Omega(M, \bar{I}[0, t_{init} + \tau], \bar{O}[0, t] \cup \bar{O}[t_{init}, t_{init} + \tau])) - U(\Omega(M, \bar{I}[0, t_{init} + \tau], \bar{O}[0, t_{init} + \tau]))$: the smaller, the better. The proposals of new uncertainty functions and scheduling policies is a commitment for future work.

References

[de Kleer and Williams, 1987] J. de Kleer and B.C. Williams. Diagnosing multiple faults. *Artificial Intelligence*, 32(1):97–130, 1987.

[Hamscher *et al.*, 1992] W. Hamscher, L. Console, and J. de Kleer, editors. *Readings in Model-Based Diagnosis*. Morgan Kaufmann, San Mateo, CA, 1992.

[Lotka, 1910] A. J. Lotka. Contribution to the theory of periodic reaction. *Journal of Physical Chemistry A*, 14:271–274, 1910.

[Otter M., 2007] Mattsson S. E. Otter M., Elmqvist H. Multidomain modeling with modelica. In Paul A. Fishwick, editor, *Handbook of Dynamic System Modelling*, chapter 36. 2007.

[Travé-Massuyès, 2014] L. Travé-Massuyès. Bridging control and artificial intelligence theories for diagnosis: A survey. *Engineering Applications of Artificial Intelligence*, 27:1 – 16, 2014.