Synchronised Qualitative Simulation in Diagnosis

Qiang Shen & Roy Leitch

Intelligent Automation Laboratory

Dept. of Electrical & Electronic Engineering

Heriot-Watt University

31-35 Grassmarket

Edinburgh EH1 2HT

U.K.

Tel: +31 225 6465

Fax: +31 225 1137

e-mail: sqi@hw.ee.ac.uk

rrl@hw.ee.ac.uk

Submitted to: *Fifth International Workshop on Qualitative Reasoning about Physical Systems*

Synchronised Qualitative Simulation in Diagnosis

Qiang SHEN and Roy LEITCH

Intelligent Automation Laboratory
Department of Electrical and Electronic Engineering
Heriot-Watt University
EDINBURGH, SCOTLAND

ABSTRACT

The reliable model based diagnosis of faults on continuous dynamic systems requires that the model of the system synchronously tracks the evolution of the observations. We argue that to achieve this the system model must produce durations associated with the respective qualitative states. We present a system for Model Based Diagnosis of Dynamic Systems, based on the use of Fuzzy Qualitative Simulation, that provides five important extensions to previous approaches: 1) The temporal durations produced by FuSim effectively constrains the search within the tracking process of the discrepancy generation, without the need for heuristics. 2) The temporal information allows synchronous tracking between predictions and observations, thereby minimising the possibility of false negative matching based on spurious states. 3) The reduction in spurious behaviours resulting from the use of FuSim, without the need for numerical information or heuristics, reduces the possibility of false positive matching. 4) Adaptive sampling of the observations is used to minimise computation and retain important dynamic information. 5) The resulting diagnostic mechanism well matches the architecture of model-based diagnosis systems, allowing a generic framework to be developed. The operation of the system is demonstrated by application to a simple system under various fault conditions.

1. Introduction

The development of effective qualitative simulation techniques is intrinsically important, however, the real advantage comes when such techniques are used in application systems. *M*odel *B*ased *D*iagnosis of industrial *S*ystems (*MBDS*), although well established using numerical approaches to system modelling [4, 6], is currently flourishing within the Artificial Intelligence community. The use of *AI* approaches to system modelling and reasoning with incomplete and uncertain knowledge has resulted in a number of general and powerful diagnostic systems [1, 2, 10, 14]. Whilst important, this work has a crucial limitation that the system model within the MBDS is limited to algebraic models of the system in equilibrium. Further, the assumption of continuity of system variables is not used to allow the behaviour (response) of the system to be compared with the predicted response over the temporal evolution of the system. In this respect, the use of qualitative simulation techniques as the system model offers exciting prospects for extending MBDS to the diagnosis of continuous dynamic systems. Such systems would allow the early detection of incipient failures (during the transient) and reduce the requirement of accessibility of system states by diagnosing over time, an approach we term *Model Based Diagnosis of Dynamic Systems* (*MBDDS*). This work is only beginning. An important contribution to this work has been provided in [3], where a system called Mimic utilises the Qualitative Simulation algorithm, QSim [5], as the system model. However, this approach has a number of difficulties: 1) no temporal information on the evolution of the system is provided by QSim, thereby making the 'tracking' of the real behaviour difficult; 2) in addition to producing a qualitative description of the real behaviour, QSim may generate many spurious behaviours, again complicating the model tracking procedure, and, perhaps, making it intractable.

This paper utilises, as the system model within MBDDS, a *Fuzzy* qualitative *Sim*ulation (*FuSim*) algorithm, developed by the authors [11, 12] that has several features useful to synchronise the model evolution with the observations:

1) the temporal duration of the qualitative states is given;
2) stronger functional constraints can be utilised;
3) a substantial reduction in the spurious behaviours [7] is achieved as a product of 1) and 2);
4) fuzzy sets allow the subjective element in system modelling to be incorporated and reasoned with in a formal way.

We show that these attributes of FuSim are crucial to efficient and effective MBDDS. At the schematic level, MBDDS can be depicted by Fig. 1. Within this primitive architecture, the *System Model* explicitly describes the structure of the physical system (the plant) to be diagnosed and an algorithm for generating the dynamic behaviour from the plant's structure. The *Discrepancy Generator* compares observations from the plant with predictions from the system model and generates appropriate discrepancies, whilst the *Candidate Generator* uses the discrepancies to produce candidates, i.e., the possible faults, to be validated by the system model. The *Diagnostic Supervisor* contains the meta-knowledge necessary to control the diagnostic process. This paper discusses how this diagnostic method determines the fault conditions of a physical system. However, we shall not focus on the supervision task but leave it as an important future work. We first briefly review the FuSim algorithm, which produces a sequence of system states with related temporal durations, providing a significant advantage for refining diagnosis over time. Then, we describe an implementation of the primitive architecture. We present, in detail, the mechanism for the comparison between the predictions and the observations, the key technique for system-monitoring and fault evaluation and which is the basis for discrepancy generation within MBDDS. This shows the important contribution to diagnosing continuous dynamic systems by using the extended functionalities provided by FuSim. Finally, we shall give a simple example to demonstrate the method presented herein.

2. A brief review of FuSim

This section presents an overview of fuzzy qualitative representation and the FuSim algorithm. Principal concepts are given and basic properties shown. A more complete and formal treatment can be found in [11, 12].

2.1. Qualitative representation of values and constraints

The choice of representation of physical quantities plays a critical role in qualitative simulation. All qualitative modelling techniques describe quantities with a small set of symbols, called qualitative values, which are abstracted from the underlying field that the variables of a physical system take values from. In FuSim a qualitative value of a system variable is a fuzzy number chosen from a subset of normal convex fuzzy numbers [15]. This subset is generated by an arbitrary but finite discretisation of the underlying numeric range of the variable. For computational efficiency, such a fuzzy qualitative value can be characterised through a parametric representation of its membership function. The 4-tuple parametric representation of a qualitative value A, $[a, b, \alpha, \beta]$, is defined as

$$\mu_A(x) = \begin{cases} 0 & x < a - \alpha \\ \alpha^{-1}(x - a + \alpha) & x \in [a - \alpha, a] \\ 1 & x \in [a, b] \\ \beta^{-1}(b + \beta - x) & x \in [b, b + \beta] \\ 0 & x > b + \beta. \end{cases}$$

The arithmetic operations for these fuzzy numbers are well-developed, and we adopt this representation to form the quantity space: informally, the set of all the qualitative values that system variables can take.

Such a fuzzy quantity space maintains three desirable properties for performing qualitative simulation, namely, finiteness, granularity, and coverage of the quantity space. In fact, as the fuzzy quantity space is generated by a finite discretization of the underlying range of each system variable, the variable will, of course, have a finite number of associated qualitative values, and the whole underlying numerical range of interest can be covered by the fuzzy qualitative values. Also, if $x_1, x_2 \in R$ characterize "similar things" or stand for "similar properties" of a variable x, then the relevant qualitative values of x_1 and x_2 will be similar. Hence, we can translate a subset of a numeric range to one qualitative value according to what is needed in a particular modelling process.

FuSim adopts a constraint-centred ontology in system-modelling [12]. A model is derived from an underlying differential equation representation or from direct application of first order energy storage mechanisms. The sets of possible values which system variables can take are restricted by either algebraic, derivative or function relational constraints amongst the variables. The algebraic operations performed within qualitative values are those in the set of fuzzy numbers [11, 12].

As with any simulation language for dynamic systems, differential operation is essential for determining the transient behaviour of a system. A derivative constraint simply reflects that the qualitative value of a variable's magnitude must be the same as that of another variable's rate of change. Different from other qualitative simulation techniques, functional relationships within FuSim are represented by fuzzy relations [15]. Thus, a relation expressed by a rule with the form that, if x is A_i, then if y is B_i, then z is C_i, can be translated into

$$\mu_{L_i}(x, y, z) = \min(\mu_{A_i}(x), \mu_{B_i}(y), \mu_{C_i}(z)).$$

When a set of n fuzzy rules is available, the resulting relation L is the union of the n elementary fuzzy relations L_i, $i = 1, 2, \dots, n$.

$$\mu_L(x, y, z) = \max_{i \in \{1,2,\dots,n\}} \min(\mu_{A_i}(x), \mu_{B_i}(y), \mu_{C_i}(z)).$$

This enables imprecise and/or partial numerical information on functional dependencies between variables to be exploited if indeed such information is available (and necessary). Actually, this technique has been well-utilised for tasks like intelligent control of industrial processes.

2.2. FuSim algorithm

Based on the above qualitative representation of values and constraints, FuSim takes as input a set of system variables, a set of constraints relating the variables, and a set of initial values for the variables, and produces a tree of states with each path representing a possible behaviour of the system as output. Like other approaches to qualitative simulation, FuSim is based on the fundamental assumption that the variables of the physical system being modelled are continuously differentiable functions of time. Thus, transitions for variables to change from their current state to the successor states are governed by a set of rules -- called the possible state transition rules [11, 12]. From these rules, a set of transitions from one qualitative state description to its possible successor states can be generated. Further restrictions on the possible successor states are imposed by checking for consistency with the definition of the constraints and the consistency between constraints which share an argument -- called constraint filtering, and information on the rates of change of the system variables held as part of the fuzzy qualitative state -- called temporal filtering. In addition, other knowledge about the system may be used to produce so-called global filtering methods [5, 12].

More specifically, constraint filtering entails an operation, called *refine*, on each constraint and each argument of the constraint iteratively until no further changes are produced. In general, let $C : C(Q(x_i))$, $i = 1, 2, 3$, be a constraint among three arguments: $Q(x_i)$, and S_i be the set of qualitative values for the argument $Q(x_i)$. Then, the refine operation is defined by

$$refine(C, Q(x_i)) = \{A_i \in S_i \mid (A_j \in S_j, j = 1, 2, 3, j \neq i); C(A_k), k = 1, 2, 3\},$$

where $Q(x_i)$ denotes either the magnitude or the rate-of-change of the variable x_i.

The possible successor states survived from the constraint filtering are further checked by using the estimates of two temporal durations, named the persistence time and the arrival time respectively [11, 12]. A persistence time is indicated by the fuzzy magnitude and rate-of-change of a variable within a particular state, presenting a description of the amount of time that the variable may remain in this qualitative state. An arrival time, however, is determined by both a variable's current magnitude and rate-of-change and the magnitude and rate-of-change of its successor state, thereby reflecting the time that a variable takes to transition from one qualitative state to another. In general, the temporal filtering criterion requires:

> For any two system variables, x and y, if the persistence times of x and y within the current state are $\Delta T_p(x)$ and $\Delta T_p(y)$ and the arrival times for x and y to transition to the possible successor states are $\Delta T_a(x)$ and $\Delta T_a(y)$

respectively, then, unless x and y are independent of each other, they must satisfy the temporal constraint as follows:

$$\Delta T_p(x) + \Delta T_a(x) \cap \Delta T_p(y) + \Delta T_a(y) \neq \Phi.$$

It is important to notice that, associated with each sequence of states, i.e., each path of the output tree, FuSim also generates a sequence of temporal intervals to indicate how long the system will persist within a particular state. Such temporal information is determined by both the persistence time and the arrival time. This is a distinct advantage of FuSim. Especially, when used for diagnosing dynamic systems, FuSim is able to show which particular portion of the predicated behaviour should be matched by an observation at a particular time point or during a particular time interval, enabling a self-controlled mechanism for the prediction from FuSim and a self-guided mechanism for the comparison between the prediction and the observation.

3. Model Based Diagnosis of Dynamic Systems

The central idea of the model-based approach to diagnosis is the use of an explicit model of a system's structure, which can reflect either normal or abnormal behaviour of the system. Such a diagnostic mechanism determines some components (or constraints) of a physical system which account for the observed abnormalities, i.e., the discrepancies between the observed and designed (normal) behaviour of the system. This section presents an approach to MBDDS using FuSim as the system model. Our work adopts the basic concept of Mimic [3] in the sense that diagnosis is refined over time through a hypothesise and test cycle. However, the work presented herein fits a more general architecture of MBDS and, more importantly, it provides a way to automatically guide and control the predictions from the model and the comparisons to be made within the discrepancy detector without requiring further domain specific knowledge.

3.1. Framework

The basic structure of the diagnostic method using FuSim can be described by Fig. 2. Within this framework, the Normal and Fault Models are represented by a set of fuzzy constraints. The normal model describes the internal structure of the physical system being monitored with respect to its intended design specification, while each of the fault models represents a particular failure mode of a component within the system, with the other components performing normally, e.g., a leakage in a pipe or a stuck pressure regulator. The Discrepancy Generator detects the discrepancies between the observations and the predictions, and the resulting discrepancies are used to evoke the Decision Tree to generate possible candidates, which most likely reflect the current working characteristic of the plant.

Clearly, such a diagnostic system structurally matches the architecture shown in Fig. 1. It is worth pointing out that, although this method, in general, falls within the general MBDS framework, it combines associative and model-based reasoning. Actually, the associative element generates fault hypotheses and the model-based element tests these hypotheses. Nevertheless, from the discussion later it can be seen that the generation of the decision tree itself, which involves the associative reasoning, may also be based on qualitatively simulating particular structural models of the plant.

3.2. Discrepancy Generation

Diagnosing a physical system crucially relies upon the discrepancy detection between the observed and predicted behaviours of the system. Thus, a technique which is able to compare qualitative predictions against observations over time is required. For coherent detection the comparison between the observed behaviour and the predicted behaviour must be made at the same system state, i.e., the same absolute time (point or duration). In other words, the model must operate *synchronously* with the natural evolution of the plant. This brings a problem to the use of qualitative simulators as the behaviour predictors because of the need to guide and control the comparison between the observations and the predictions. Temporal information becomes essential to the maintenance of synchronous behaviour. Without this it is impossible, without resorting to heuristics, to control the evolution of the models, including spurious behaviours and fault models. It is in this respect that FuSim provides important advantages. FuSim produces a temporal duration sequence associated with the state sequence and substantially reduces the number of spurious behaviours generated, thereby enabling an effective and efficient discrepancy detection method to be developed.

The first step in the detection of discrepancies is to gather the observations and to interpret these onto the quantity space being used to qualitatively match the predicted and observed states. The key question here is how to determine the 'sampling' time. Two approaches to this are possible. The first leaves the time point (t_n) at which an observation is interpreted open and (presumably) under the control of the diagnostic supervisor. This is the approach taken in Mimic but avoids the question of how to select the sampling intervals. The second approach would be to use a constant sampling interval, related to the assumed system dynamics through *Shannon*'s sampling theorem. However, this may incur unnecessary computation if the system response is slowly varying. Also, dynamics under fault conditions are normally unknown and may be missed by a constant sampling interval. This latter requirement suggests 'adaptive' sampling dependent upon the rate-of-change of the observations. This is achieved by the third method, and the one adopted herein. In this case, the sampling intervals, and hence t_n, are determined directly from the observations. The continuous real-valued functions obtained from the transducers measuring the observations are continuously monitored and compared to the quantity space of the variables used in the current system model. Only when a transition of the (fuzzy) qualitative state is detected is the discrepancy detection mechanism activated. This produces an 'adaptive' sampling interval determined by the actual system behaviour and reduces the risk of missing important distinctions.

The rules used to guide and control the discrepancy detection are directly deduced from knowledge of the temporal durations of the qualitative states and can be stated as follows:

(1) Treat the current observation, $OBS(t_0)$, as the initial state of a model being evaluated and the time when the observation is made as the initial temporal point.

(2) From the next observation $OBS(t_1)$, generate the simulated behaviour of the model from $OBS(t_0)$ until the temporal upper bound of a qualitative state meets or covers t_1. Compare this last generated state with $OBS(t_1)$. If they are matched, redo (2) with a further observation and so on; otherwise, discard the model currently being simulated.

Notice that, when more than one possible behaviour results from FuSim (due to the theoretically unavoidable ambiguities of qualitative simulation [13]), a model remains to be further evaluated provided that one of the last states generated thus far within these behaviours matches with the observation. Also, the simulation will continue from the behaviour which contains the matched state.

Illustratively, the way to track a model can be depicted by Fig. 3, and explained in the following. Without losing generality, suppose that the first two observations from the plant are $OBS(t_0)$ and $OBS(t_1)$, FuSim then uses $OBS(t_0)$ as the initial state, $PRED(T_0)$, and starts running the simulation of the model. If it generates the possible next state, say, $PRED(T_1)$, which matches $OBS(t_1)$ under the condition that $t_1 \subseteq T_1$, no discrepancy is detected and hence the diagnostic system waits for another observation, $OBS(t_2)$, to be available. After this, FuSim continues predicting the successor state $PRED(T_2)$, however, this state's temporal information indicates that for any $t \in T_2, t < t_2$. Thus, FuSim keeps making further predictions from $PRED(T_2)$ and results in both $PRED(T_2')$ and $PRED(T_2'')$ as possible next states. Now that $PRED(T_2'')$ does not match with $OBS(t_2)$, though $t_2 \subseteq T_2''$, no further predictions will be made following the branch beginning at the $PRED(T_2'')$ whatever later observations are obtained. However, since $PRED(T_2')$ matches $OBS(t_2)$ and $t_2 \subseteq T_2'$, the model being evaluated remains as suspect and the prediction, detection, and decision-making cycle recurs, starting from the matched prediction $PRED(T_2')$. Otherwise, if $PRED(T_2')$ also conflicts with $OBS(t_2)$, this model is discarded and another fault is assumed. Clearly, due to the temporal information in FuSim only a small number of predictions starting from certain current state(s) are generated and, also, the discrepancy detector makes the comparison between a prediction and an observation only when the prediction is the last one generated with respect to the latest sampling time.

Within the above rules, discarding a model simply implies that this model is inconsistent with the current working condition of the physical system. If a suspected model is thus exonerated from the candidate set generated by the candidate generator, other hypotheses must be further tested. From this point of view the diagnostic method presented herein can be seen as a system identification tool. Methodologically, this is rather different from model-based troubleshooters [1, 10], where only the normal system's model is necessary and a particular faulty component is determined if a retraction of its corresponding correctness assumption makes the predicted behaviour consistent with observations.

It is important to notice that this diagnostic mechanism tracks a model without distinguishing if the model is the normal one or a fault one. Therefore, an on-line diagnostic system built in this way can actually perform system-monitoring with an identical structure. Once a discrepancy is detected, the normal model will be substituted by a fault model. The monitoring task then becomes a fault identification task.

3.3. Candidate generation

Within the basic framework shown in Fig. 2, the Decision Tree proposes candidates for further evaluation once there are some discrepancies produced. The tree is, in fact, a set of rules able to classify the discrepancies by mapping them onto particular fault hypotheses. The diagnostic knowledge embedded in the tree is induced from the results of simulating the fault models by FuSim. Such a model-based learning technique provides good coverage of the available knowledge of faults [8]. Although this method may be rather time-consuming it has only to be done once and is performed off-line.

This technique of developing rules for candidate generation has been reported in [3]. We utilise this technique to obtain the decision tree. Essentially, the off-line preparations are made through defining each particular fault model of the physical system in terms of a set of fuzzy qualitative constraints. Then, each such defined model is simulated using FuSim, starting from each possible initial state, i.e., the possible initial operating condition with respect to the basic architecture shown in Fig. 1, and generating a complete behaviour tree for the model. From the states of each behaviour tree, the training instances can be constructed by using the discrepancies of the states (expressed by the fuzzy qualitative magnitudes and fuzzy qualitative rates of change) of the observable variables against those of the normal model, generated from the same initial states, such that each instance is labelled with the fault model used for generating the tree. Of course, these discrepancies are subject to the temporal restriction. That is to say, the comparison between the states of the normal and fault models is only performed when both states occur at the same time. Finally, the training instances are compressed by an inductive learning program [9] to form the desired decision tree. Clearly, the structural model of a physical system plays the central role in such a machine learning process, wrong models result in incorrect diagnostic knowledge.

It is worth indicating that, when making the off-line preparations to obtain the decision tree for a particular physical plant to be diagnosed, the comparison between the behaviour of the normal model of the plant and the behaviour of a fault model is executed in a similar way to that described for the on-line disrepancy detection, except that the time used to guide and control the comparison is not the real time but that resulting from artificially simulating the normal model with the initial time point being zero.

An alternative approach to candidate generation is currently under investigation by the authors. This identifies the space of the possible (likely) model variation under fault conditions and systematically searches this space, using most likely variation first, until a 'matching' fault model is obtained. This requires that the possible model dimensions are identified and characterised. In our approach we have identified two dimensions based on the FuSim representation system [7]: 1) The modification of the functional relationships represented by fuzzy relations. This is a very common cause of faults, e.g., increased friction or reduced mass. 2) Structural changes due to physical effects, this typically reduces the system order, i.e., removes particular derivative constraints. The selection of these dimensions and assertion of possible fault models is under the control of the diagnostic supervisor which controls the 'search' for possible fault models. This approach avoids the need for explicit fault models, including the off-line compilation of these, and replaces it with systematic search. The success is, of course, very dependent upon choosing the right dimensions to search. However, this approach seems promising and is the one which we are currently pursuing.

4. Example

A simple example, of "a mass on a spring", as depicted in Fig. 4, is given to demonstrate how a physical system is modelled within FuSim, and to show how the diagnostic system based on FuSim determines faults. The physical system used herein consists of three variables: the displacement of the mass from the rest point of the spring, x, the velocity of the mass, v, and the acceleration of the mass, a. For simplicity, the following set of 4-tuple parametric fuzzy numbers is chosen to form the fuzzy quantity space:

$$Q_F = \{[-1, -0.7, 0, 0.1], [-0.6, -0.6, 0, 0], [-0.5, -0.1, 0.1, 0.1], [0, 0, 0, 0], [0.1, 0.5, 0.1, 0.1], [0.6, 0.6, 0, 0], [0.7, 1, 0.1, 0]\},$$

and is abbreviated to

$$Q_F = \{-b, -0.6, -s, 0, s, 0.6, b\};$$

with each value corresponding to a perceived meaning, for instance, $-b$ denoting *negative big* and s indicating *positive small*.

4.1. Normal system modelling and its qualitative behaviour

Suppose that the normal model assumes frictionless motion. Thus, the physical system can be characterised by two derivative and one functional constraints as follows

$$deriv\ x = v, \qquad deriv\ v = a,$$

$$\begin{array}{c|ccccccc} a \,\tilde{}\, x & -b & -0.6 & -s & 0 & s & 0.6 & b \\ \hline -b & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ -0.6 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ -s & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ s & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0.6 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ b & 1 & 1 & 0 & 0 & 0 & 0 & 0 \end{array} .$$

The first and second equations establish the ordinary derivative relationships holding amongst the distance, velocity, and acceleration of the mass. The third functional relation between a and x is a weak, but stronger than monotonic operator, form of Hooke's law represented as a degenerated fuzzy relation [12].

With such a system model and the initial state expressing that the mass is moved away from the equilibrium point, $x = 0$, to $x = 0.6 > 0$, and then let go. FuSim produces a unique behaviour for each system variable, as shown in Fig. 5 (one cycle only), with the following durations associated with the fuzzy qualitative states:

$$t_0 = 0, \quad t_1 \in [0.09, 1], \quad t_2 \in [0.63, 2.5], \quad t_3 \in [0.72, 3.5] \quad t_4 \in [1.26, 5],$$

$$t_5 \in [1.35, 6], \quad t_6 \in [1.89, 7.5], \quad t_7 \in [1.98, 8.5], \quad t_8 \in [2.52, 10].$$

4.2. Diagnosing faults

To be concise, presume that the system only has two kinds of possible faults: either a non-zero friction condition or a mass-stuck condition. The friction condition can be modelled as five fuzzy constraints, namely,

$$deriv\ x = v, \qquad deriv\ v = a, \qquad a = a_1 + a_2,$$
$$a_1 \sim x, \qquad a_2 \sim v;$$

where additional variables a_1 and a_2 are introduced to include the effectiveness of the friction. Two fuzzy relations, $a_1 \sim x$ and $a_2 \sim v$ are modelled in a very similar way to the weak form of Hooke's law shown above, and are not explicitly presented because of lack of space. Within this model we assume that the damping coefficient is *medium*, although in a real situation we may also need to consider cases with *small* or *big* damping coefficients. The stuck model is simple since such a system is equivalent to a friction system with an infinite damping coefficient. Thus, the model can be described by

$$x = x_0, \qquad v = a = 0,$$

with x_0 being the displacement where the mass is stuck.

Suppose the diagnostic system is initially used to monitor the normal system and starts monitoring from the same initial operating condition as the system. It is necessary, however, to point out that this starting condition is thus presumed purely for the reason of easy presentation, other observations from the system can also be used to serve as the initial working condition of the diagnostic system. When an observation is obtained at $t_1 = 0.5$ from observable variables (x, v) such that $OBS(T_1) = (0.3, -0.4)$, where the observed values are normalised with respect to variables' underlying numerical ranges. Based on the normal model, FuSim predicts the immediate next state from the initial one, $(s, -s)$, which matches with the observation under the condition that the predicted state temporally covers the observation time (see the result shown in the preceding sub-section). This indicates that the physical system is performing normally. In the same way, two further observations, $OBS(1.6) = (0, -0.6)$ and $OBS(2.7) = (-0.35, -0.45)$, also guide the system model to make the next two predictions that match with the observations. Now, the fourth observation is obtained such that $OBS(3.8) = (-0.45, 0)$, whose temporal information is used, as usual, to guide FuSim to produce a predicted state in a synchronous manner. However, from the normal model FuSim generates the state $(-0.6, 0)$ as the successor state of the previous state $(-s, -s)$ (matched with $OBS(2.7)$), thereby resulting in a discrepancy.

Now that the discrepancy between the current observation and prediction has been detected and FuSim produces unique behaviour from the normal model, this model is discarded and, hence, the MBDDS system alerts that the physical system is 'faulty'. Therefore, the original monitoring task is changed into a diagnostic one. The discrepancy detected is used to evoke the two possible fault models through the candidate generator. With the initial state being $(-0.45, 0)$, predictions can be generated from the fault models. Under the guidance of the temporal information from the next observation $OBS(4.9) = (-0.2, 0.35)$, two predictions from the two models are $(-s, s)$ and $(-s, 0)$, respectively. The latter state, $(-s, 0)$, of the stuck model can be, of course, expressed directly by $(-0{:}45, 0)$ if we do not intend to use any quantity space to qualitatively represent this model. Clearly, the prediction from the friction system's model matches with the observation but that from the mass-stuck system does not. Henceforth, the stuck model does not reflect the actual working condition of this mass-on-a-spring system and can then be exonerated from the suspect list. The diagnostic system thus returns the friction condition as the fault that the physical system is operating on. Alternatively, it may continue tracking the possible fault model if it is necessary to confirm that the physical system is suffering from this fault. For instance, from two further made observations, $OBS(5.0) = (0, 0.45)$ and $OBS(6.1) = (0.25, 0.3)$, FuSim will generate two matched states based on the friction model: $(0, s)$ and (s, s). The whole monitoring and diagnosing process explained above is shown in Fig. 6 (for the variable x), where the temporal points $(t_0, t_1, ..., t_7)$ satisfy those given in the previous sub-section.

5. Discussion and conclusion

This paper presents important extensions to model-based diagnosis of continuous dynamic systems by using Fuzzy Qualitative Simulation as the system model. Although diagnosing dynamic systems using Artificial Intelligence techniques is just at its beginning, FuSim makes such a task feasible, since it can generate a qualitative description of the dynamic behaviour of a system with related temporal durations, allowing the synchronous detection of discrepancies between the observations and predictions of the system.

The work presented here was inspired by Mimic, the pioneering approach to diagnosing continuous dynamic systems by the use of qualitative simulation [3]. However, approach presented herein provides five important extensions to previous work on MBDDS:

1) The temporal durations produced by FuSim effectively constrains the search within the tracking process of the discrepancy generation, without the need for heuristics.
2) The temporal information allows synchronous tracking between predictions and observations, thereby minimising the possibility of false negative matching based on spurious states.
3) The reduction in spurious behaviours resulting from the use of FuSim, without the need for numerical information or heuristics, reduces the possibility of false positive matching.
4) Adaptive sampling of the observations is used to minimise computation and retain important dynamic information.
5) The resulting diagnostic mechanism well matches the architecture of model-based diagnosis systems, allowing a generic framework to be developed.

Work is on-going in two directions. The first is to refine the diagnostic mechanism and take the diagnostic supervision task into consideration. The second, to seek a way to combine this mechanism with techniques developed for model-based troubleshooting, in particular, with those reported in [2, 14] that utilise fault models as well as normal model but are still restricted to static models. We believe, the outcome of the merger of the dynamic and static approaches would be beneficial to model-based diagnosis of physical systems. This is a major focus of a new *EEC ESPRIT* project on MBDS (P5143), named *ARTIST*, to which the authors are contributing.

References

[1] J. de Kleer and B. C. Williams, Diagnosing Multiple Faults, in: *Artificial Intelligence,* **32**, pp. 97 - 130, 1987.

[2] J. de Kleer and B. C. Williams, Diagnosis with Behavioural Modes, in: *Proceedings of the 11th International Joint Conference on Artificial Intelligence*, Vol. 2, pp. 1324 - 1330, 1989.

[3] D. Dvorak and B. Kuipers, Model-Based Monitoring of Dynamic Systems, in: *Proceedings of the 11th International Joint Conference on Artificial Intelligence*, Vol. 2, pp. 1338 - 1343, 1989.

[4] R. Isermann, Process Fault Detection Based on Modelling and Estimation Methods - A Survey, in: *Automatica*, Vol. 20, No. 4, pp. 387 - 404, 1984.

[5] B. J. Kuipers, Qualitative Simulation, in: *Artificial Intelligence,* **29**, pp. 289 - 338, 1986.

[6] Y. Landau, *Adaptive Control: The Model Reference Approach*, Marcel Dekker, 1979.

[7] R. R. Leitch and Q. Shen, Model Dimensions within Fuzzy Qualitative Simulation, to appear in: *Proceedings of the 13th IMACS World Congress on Computation and Applied Mathematics*, 1991.

[8] D. A. Pearce, The Induction of Fault Diagnosis Systems from Qualitative Models, in: *Proceedings of the 7th National Conference on Artificial Intelligence*, pp 353 - 357, 1988.

[9] J. R. Quinlan, Induction of Decision Trees, in: *Machine Learning*, **1**, pp. 81 - 106, 1986.

[10] R. Reiter, A Theory of Diagnosis from First Principle, in: *Artificial Intelligence,* **32**, pp. 57 - 95, 1987.

[11] Q. Shen and R. R. Leitch, Integrating Common-sense and Qualitative Simulation by the Use of Fuzzy Sets, in: *Proceedings of the 4th International Workshop on Qualitative Physics*, pp 220 - 232, 1990.

[12] Q. Shen and R. R. Leitch, Fuzzy Qualitative Simulation, submitted for journal publication, 1991.

[13] P. Struss, Mathematical Aspects of Qualitative Reasoning, in: *International Journal for Artificial Intelligence in Engineering,* 3(3), 1988.

[14] P. Struss and O. Dressler, Physical Negation - Integrating Fault Models into General Diagnostic Engine, in: *Proceedings of the 11th International Joint Conference on Artificial Intelligence*, Vol. 2, pp. 1318 - 1323, 1989.

[15] R. R. Yager, S. Ovchinnikov, R. M. Tong, and H. T. Nguyen (Eds.), *Fuzzy Sets and Applications: Selected Papers by L. A. Zadeh,* John Wiley & Sons, 1987.

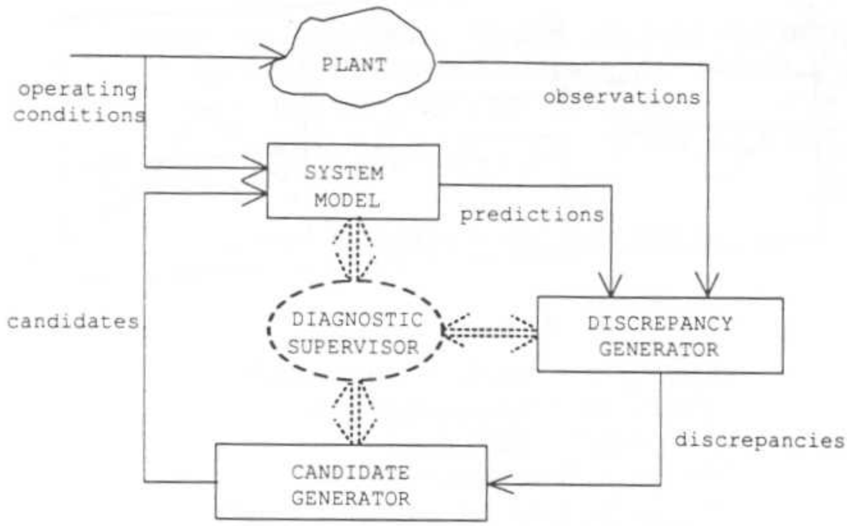

Fig.1. Architecture of MBDDS

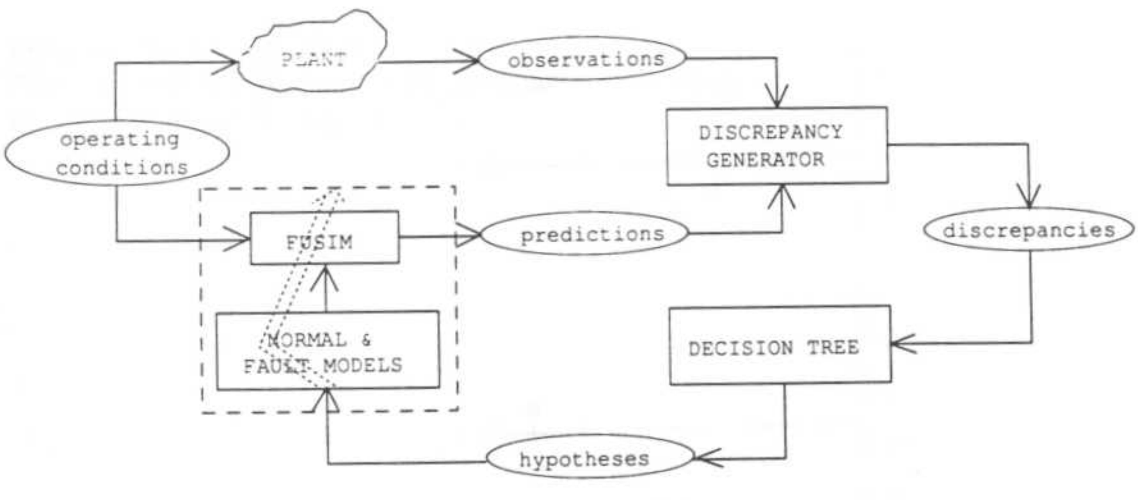

Fig.2. MBDDS Using FuSim

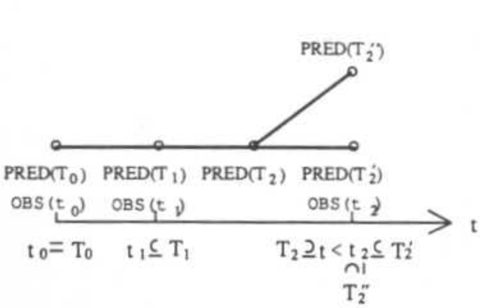

Fig.3. Tracking a Model

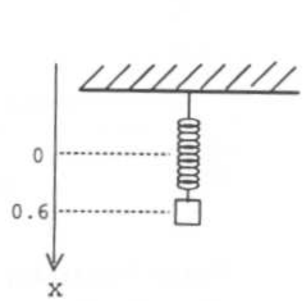

Fig.4. A Mass on a Spring

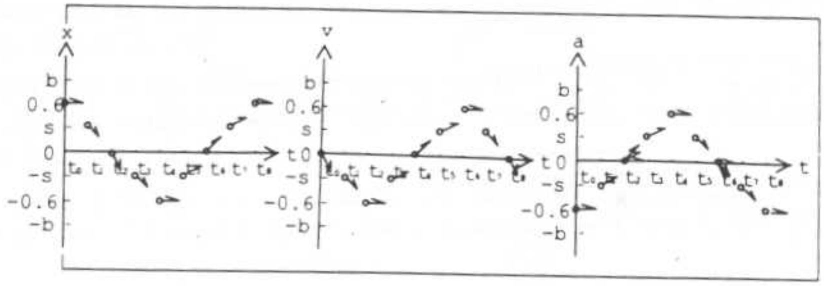

Fig. 5. Normal Behaviour

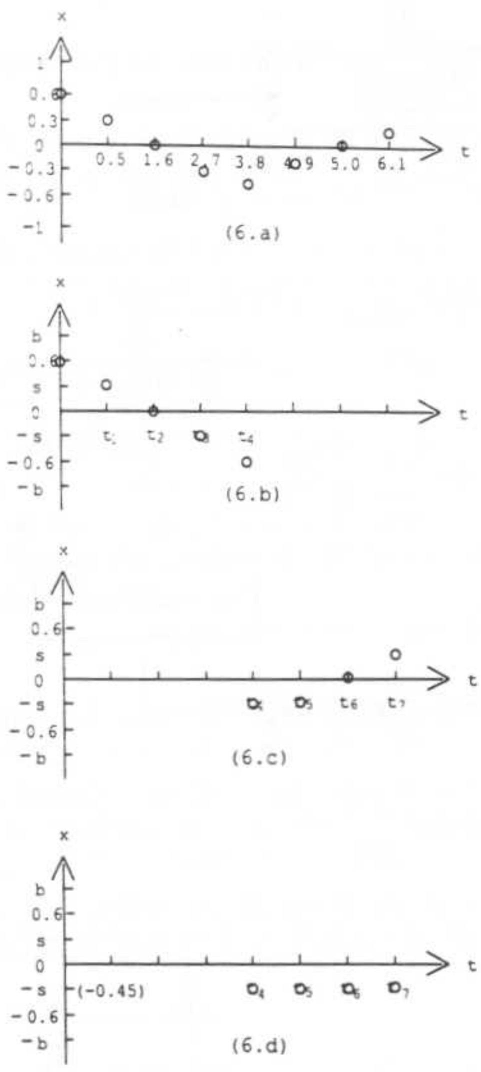

Fig.6. Monitoring and Diagnosing

(6.a) Observations

(6.b) Predictions from the Normal Model

(6.c) Predictions from the Friction Model

(6.d) Predictions from the Mass-Stuck Model