# Prediction, Monitoring, and Diagnosis in Complex Dynamic Systems

Pieter J. Mosterman and Gautam Biswas

December 13, 1995

**Abstract**

Model-based monitoring and diagnosis of dynamic physical systems require well-contrained models of the system to prevent computational complexity. The key to modeling for diagnosis is the ability to generate models that accurately describe both steady state characteristics and the dynamics of system behavior. The bond graph modeling formalism provides a suitable framework that integrates and facilitates (i) gneration of behavioral constraints from energy-based physical laws, (ii) expresion of system dynamics in terms of energy transfer between constituent elements, and (iii) modeling of steady state behavior as a special case of dynamic behavior. Our work proposes a modeling scheme that combines bond graph principles with finite-state automata to provide a hybrid modeling scheme for complex engineering systems. The scheme allows for the application of qualitative reasoning methodologies for diagnostic analysis and prediction of system behavior. At the same time, more detailed quantitative information can be introduced to refine diagnostic and predictive analyses. A general methodology is developed for prediction, monitoring, and diagnosis in complex, dynamic physical systems.

# Contents


**1 Introduction** **3**

**2 Integrated Architecture for Prediction, Monitoring, and Diagnosis** **5**
2.1 The Model . . . . . 6
2.1.1 Generating the Temporal Causal Graph . . . . . 7
2.2 Candidate Generation . . . . . 8
2.2.1 Component Parameter Implication . . . . . 8
2.2.2 Structural Changes . . . . . 9
2.2.3 Generating Sets of Candidates . . . . . 11
2.3 Prediction . . . . . 11
2.3.1 Forward Propagation . . . . . 11
2.3.2 Structural Changes . . . . . 13
2.3.3 Steady State . . . . . 14
2.4 Monitoring . . . . . 15
2.4.1 Sensitivity . . . . . 16
2.4.2 Discontinuities and Progressive Monitoring . . . . . 17
2.4.3 Temporal Behavior . . . . . 19
2.5 Conclusions . . . . . 20

**3 Monitoring the Simulated Latched Bi-Tank System** **21**
3.1 Completely Observable System . . . . . 21
3.2 Not Completely Observable System . . . . . 22
3.3 Conclusions . . . . . 23

**4 Application to a Fast Breeder Reactor System** **24**
4.1 The Model . . . . . 26
4.1.1 The Hydraulic Domain . . . . . 27
4.1.2 The Thermal Domain . . . . . 27
4.2 Prediction . . . . . 28

**5 Summary and List of Tasks** **31**

**6 Schedule of Activities** **31**

# 1 Introduction

Recent advances in model-based and qualitative reasoning have led to researchers developing large scale models of complex, continuous systems, such as power plants, aircraft, and space station sub-systems. Configuration and individual component description changes cause multi-mode behavior in complex systems. Multiple configurations are achieved by modular, hierarchical compositional modeling schemes A primary reason for component description changes is the representation of complex non linear behaviors as simpler, discontinuous, piecewise linear behaviors.

To perform diagnosis of a system, a set of observable variables is continuously monitored and matched against their predicted values in the desired normal mode of operation. Deviations from these normal values indicate the occurence of one ore more faults in the system. First a number of candidate sets is generated by constraint analysis and propagation methods applied to the system model. These candidate sets are then used to predict future behaviors of the observed variables based on the system model. Monitoring of these variables then allows the continued validation or refutation of hypothesized candidate sets. The ultimate goal is to accrurately isolate problems and restore the system to normal operation by: (i) replacing faulty components, and (ii) making control changes to bring system behavior back to desired operating ranges.

Usually normal plant operation is well-understood, and can be analyzed using steady state models of system operation created by design and operations engineers. Well defined techniques can be employed to study multiple operating points of the plant that occur around predefined steady state or equilibrium behaviors. However, when faults occur in the system, behavior is dynamic and includes transients that are harder to model and analyze. This makes the task of diagnosing faulty behavior difficult, and it becomes hard to formalize behavior in a computationally efficient form. Furthermore, systematic automation of diagnosis methodologies often cannot match human performance, because humans use ad hoc methodologies and descriptions such as specific sounds and textures to characterize faults, but these are hard to incorporate into system models.

Faults cause deviations from steady state operations and cause the system to move into new steady state conditions, or after some transient behavior return to the original steady state. Therefore, it is essential to track and analyze system transients to identify and isolate faulty components in dynamic systems. As discussed, modeling, tracking, interpreting, and analyzing dynamic systems and transient behavior is a difficult task. To eliminate the modeling difficulties but to keep the dynamic, discriminative, information, several methods have been proposed that perform diagnosis based on deviations from a static model [9, 5]. However, these methods result in underconstrained process models. This especially causes problems for larger systems where the number and size of sets of fault candidates explode and the diagnosis problem becomes intractable.

After initial candidate generation for diagnosis, a prediction step is required to generate future behavior to help refine possible candidatates. Similarly to the candidate generation problems with underconstrained models, prediction of future behavior based on a set of fault candidates becomes increasingly unreliable if less constrained models are used. Notice that even for experienced plant operators it is hard to predict future behavior from given fault situations. However, prediction is crucial to diagnosis since the more accurate the prediction the easier is it for the diagnosis algorithm to quickly prune the search space and focus on a smaller number of final candidate sets.

Prediction drives the monitoring process, which is used for hypothesizing initial candidates and refining existing candidates. Therefore, the monitoring stage incorporates aspects that are closely involved with the other stages of the overall diagnosis process. Monitoring parameters such as sampling rates affect measurement interpretation, and, therefore, candidate generation and refinement. Note that depending on the monitoring implementation, certain faults may be distinguishable or not from others, and this will determine the overall diagnostic accuracy.

Both the candidate generation and prediction methodologies, rely on a model of the system that incorporates the various modes of operation of the system processes, and can reason about the dynamic behavior of the system. This dynamic behavior is by nature continuous. However, when faults occur the system may undergo structural changes, e.g., a wire may break in an electrical system. The model needs to have the capability to incorporate such changes. Note that such structural changes may affect the causal relations among parameters and variables in a system. In the case of the breaking wire, each of the new systems becomes current driven (value 0) at the point of breakage, whereas before the breaking one of the two was voltage, driven and the other was current driven.

With the emerging complexity of embedded control systems [13], i.e., physical systems that are controlled by digital computers, interest in methods to model interaction between the signals and power domain is becoming increasingly important. Process control operations, are now implemented as sophisticated Programming Logic Arrays and/or complex software modules run on embedded computer systems. These digital control mechanisms are discrete in nature, which requires that complete modeling schemes (control *and* process) to be hybrid. So the chosen modeling method has to allow for mixed continuous-discontinuous behavior based on constraints from the signals domain as well as the power domain.

Additionally, from a modeling perspective, it is desired to support compositional modeling techniques in combination with an organizational hierarchical paradigm. In case of larger systems, modeling each of the sub-systems independently each time becomes very inefficient. A library of sub-systems that can be connected through a rigorously specified interface remedies this. Moreover, to aid the modeler in keeping a comprehensive view of the entire process, the desired level of detail of the sub-system models may vary. This issue is addressed by hierarchical modeling which, in case of large systems, becomes of paramount importance. Furthermore, the sub-systems to be included may, and in general will, span several different physical domains. So, the modeling method has to be generic and flexible enough to allow for modeling these different domains, yet has to rigorously specify the interaction between the model fragments.

Summarizing, the set of desired features of a model-based prediction-monitoring-diagnosis architecture are:

- computational tractability
- dynamic models
- compositional, hierarchical, across domains
- structural changes, discontinuities in signals and power domain

- provide for additional, numerical, information
- imperative to consider prediction and monitoring

To satisfy these requirements, we use

- bond graphs as modeling formalism [10, 3], and
- extend this paradigm with support for compositional modeling [1] and a hybrid formalism [7].

The modeling aspects of these requirements are almost entirely fulfilled by the bond graph modeling formalism, which is based on the concept of energetic modeling. Since energy exchange is the basic form of interaction in physics, this method works well for modeling physical systems. However, because physical systems in principle do not exhibit discontinuous behavior, [1] this phenomenon is not embodied by the bond graph formalism. To eliminate this severe limitation on the practical use of the method, hybrid bond graphs are proposed. Though the notion of *divergence of time*[2] is not formally proved for this method yet, the mixed continuous/discontinuous formalism handles discontinuities well when modeling discretion is heeded.

In the prediction-monitoring-diagnosis architecture described in this proposal, hybrid bond graphs form the basis for developing system models. The prediction, monitoring, and diagnosis algorithms are also based on the same modeling methodology. A pragmatic implementation of the algorithms is given, yet more sophisticated techniques will result in better performance. Furthermore, the architecture implementation that is described applies the basic qualitative values $-, 0, +, ?$. Extension of this by applying interval algebra, symbolic manipulation, or numerical information is possible by merit of the hybrid bond graph formalism.

Goals of this project:

- Develop a general and systematic modeling scheme for hybrid systems and establish the divergence of time property formally for this modeling scheme.
- Develop sophisticated monitoring and diagnosis schemes for control, maintenance, and troubleshooting of complex processes and equipment.

# 2 Integrated Architecture for Prediction, Monitoring, and Diagnosis

Diagnosis is a process of close interaction between candidate generation, prediction, and monitoring. A schematic representation of the entire model-based diagnosis problem is presented in Fig. 1. In this diagram, the process is equipped with sensors that observe a number of variables in the models stored in the model base. These observables may be qualitative (*low,*

[1] Discontinuities in models are the result from either abstracting the time scale or component parameters.

[2] When a discontinuous change occurs, it may trigger a series of discontinuous changes to occur instantaneously. Divergence of time means that this series of changes does not continue infinitely, but some actual state is reached at one point.

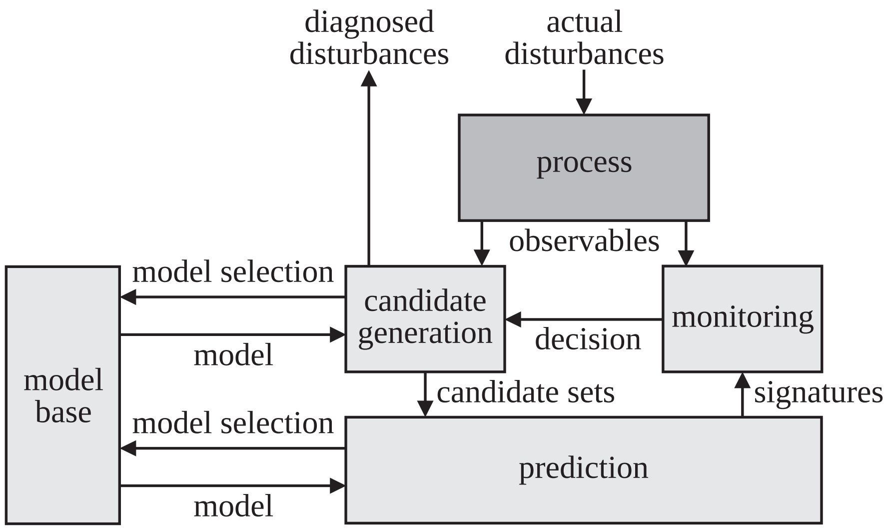



Figure 1: **Architecture for prediction, monitoring, and diagnosis.**

*normal, high, unknown*) or quantitative. When a deviation from steady state (*normal*) is detected, an alarm is activated and a number of sets of candidates is generated based on the process model that is available in the model base. This model base is then used by the prediction module to predict the future values of the observed variables, their *signatures*. The monitoring module takes these signatures and decides whether they are conform with the observations. It is then up to the candidate generation module to refute hypothesized sets of candidates.

This section develops the general architecture in more detail. The architecture is based on the use of a temporal causal graph which can be derived from hybrid bond graph models. Flow diagrams of each of the separate algorithms are given though the specific implementations algorithms may vary, e.g., feature detection may occur in a variety of ways. Which implementation to select is left to the designer's discretion and is situation dependent. For example, because of real-time constraints a full blown FFT may not be possible. However, fundamental issues of, e.g., structural changes, timing, and sensitivity are addressed and incorporated in the architecture.

So, the combination of the candidate generation module with prediction and monitoring provides a general framework for diagnosis of dynamic physical systems, specifically based on hybrid bond graph models. A basic qualitative implementation is established. However, depending on the a priori knowledge or sophistication of the diagnosis procedure the algorithms may involve more detailed information and become increasingly effective. The architecture inherently supports these extensions.

## 2.1 The Model

The model in the architecture has to satisfy a particular set of requirements. First, it has to describe the correct behavior of the process for normal behavior as well as fault behavior. This fault behavior may be the result of either parameter changes or structural changes. So, any possibly deviating parameters or activated sub-processes have to be included in the model. Furthermore, because of inherent structural changes, the model typically has to be based on a hybrid formalism. Second, the model has to allow for an algorithmic implication of elements based on deviating observations. Third, the model has to allow for predicting future behavior of the observed variables. Fourth, the model has to describe the dynamic, loading, behavior of the process.

Additional desired features include heavy constraints to limit the search space: Conservation of energy and power continuity are physical laws that render themselves prime candidates to be implemented to this end. Also, models that allow for use of qualitative as well as quantitative information and that can produce symbolical as well as numerical output are preferred. Their use prevents any loss of a priori information that may demonstrate to be a discriminating factor.

Since the hybrid bond graph modeling formalism [7] incorporates all the above features, it is chosen to serve as the formalism to construct the models in the model base. However, any other modeling formalism that supports the hybrid modeling paradigm and allows for the generation of a temporal causal graph can be used. Though strictly not required but to provide better insight in the candidate generation and prediction stage, a temporal causal graph is derived from the bond graph model. This temporal causal graph is used as the process model of a particular mode of operation. The underlying hybrid bond graph model is used to derive this specific mode.

### 2.1.1 Generating the Temporal Causal Graph

The causal graph extended with temporal behavior of relations is algorithmically derivable from the hybrid bond graph model of a physical system. To this end, the SCAP algorithm [11] is used to assign causality to all bond graph components based on local information. Then, this causality can be traced to find the causal relations between system variables.

In the temporal causal graph, vertices represent effort and flow variables whereas edges represent the relations between these variables. These relations are either the result from junctions or bond graph components. In case a relation results from a junction, its value is either $-1$, 1, or $=$. In case of a component parameter, the edge represents the component parameter's constituent relation. So, for a resistor with flow causality, the edge between effort and flow is labeled $\frac{1}{R}$. For a capacitor, for example, the edge is labeled $\frac{1}{C}dt$.

Notice that junctions, transformers, and resistors introduce magnitudinous relations whereas capacitances and inductances also introduce temporal effects. These temporal effects are integrative by nature and their associated rate of change is determined by the path that links an observed parameter to a fault.

For example, Fig. 2 shows a two tank system with a latch that possibly disconnects the both tanks. Also shown is the system's hybrid bond graph model [6]. Initially, the latch is standing upright (*closed* state) and so can open in either direction, as soon as its threshold pressure difference is exceeded. After the latch opens in one direction, it falls and cannot open in the opposite direction anymore. This latch is an illustration of a structural change and has to be modeled by sequential logic, in this case depicted by the finite state machine which controls the 1-junction to be ON or OFF. Fig. 3 shows the causal graph for this system when the latch is opened in either way, i.e., the junction is ON. The variable that is affected

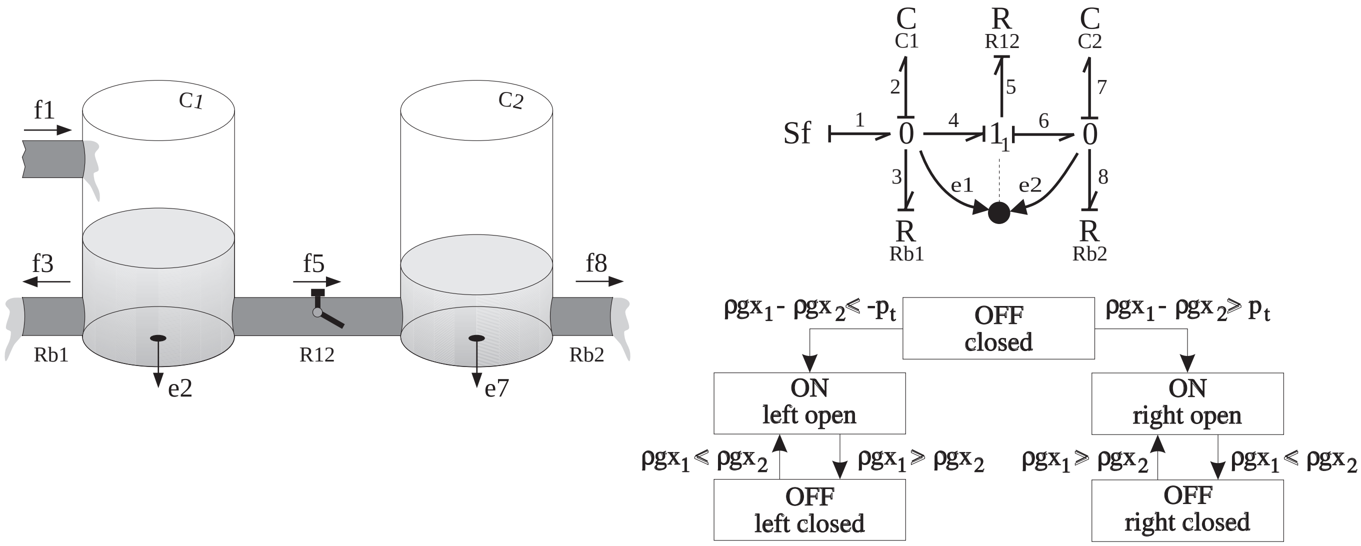



Figure 2: **Bond Graph and Causal Graph: Two connected tanks containing a fluid.**

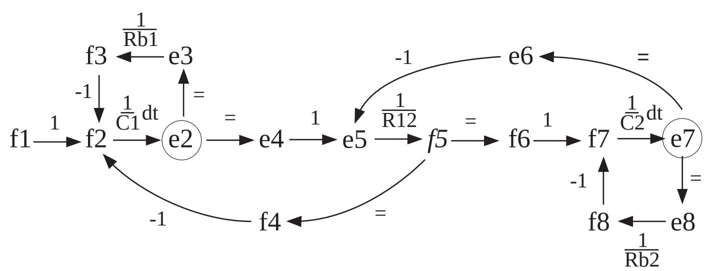



Figure 3: **Causal graph of the latched bi-tank system.**

by the controlled junction, in this case the flow $f_5$, is printed in italics. In the temporal causal graph model, this vertex has a *controlled* attribute which is used in candidate generation and prediction.

One of the advantages of bond graph models is the automatic deduction of not just particular modes of dynamic operation but also the steady state model of a system. In case of the latched bi-tank system, both the tank capacities in steady state can be replaced by flow sources with value 0, since no stored energy change takes place. The steady state bond graph and its resulting steady state causal graph are shown in Fig. 4. Notice the causality changes that have occurred with respect to the temporal causal graph as derived from the bond graph describing the dynamic behavior, Fig. 3.

## 2.2 Candidate Generation

### 2.2.1 Component Parameter Implication

When a deviation from normal is detected, a backward propagation step using the temporal causal graph implicates components, Fig. 5. This candidate generation step is executed for

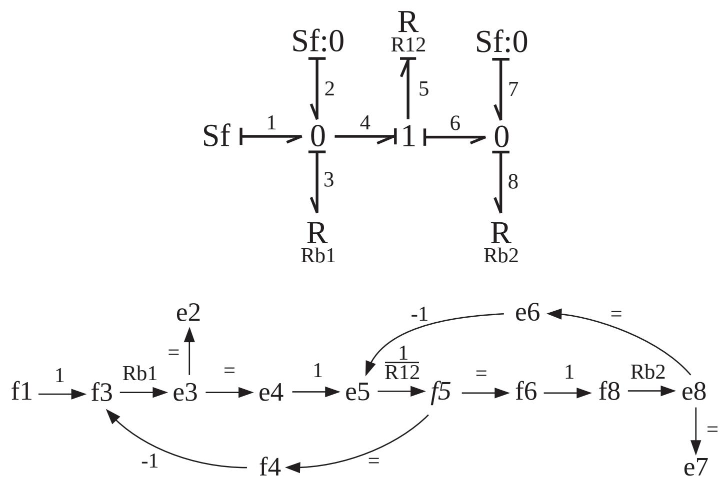



Figure 4: **Steady state bond graph of the latched bi-tank system and its corresponding causal graph.**

each observed variable that deviates seperately. The sets of candidates are merged later. In its most fundamental form, the backward propagation uses the qualitative values $-, 0, +$ to represent *below normal, normal*, and *above normal*. If more quantitative information is available about measurements or system parameters, this can be included. When backpropagating, the deviating value of an observed vertex is propagated backward, assigning corresponding deviations to all vertices along the way that have no assigned value yet. An example is shown in Fig. 6 for a deviating pressure, $e_7$, in the right tank of the two tank system in Fig. 2. The propagation ends along each path when a conflicting assignment is reached. Now, all component parameters along this path can be implicated as being possible faults.

Because of the consistency of physical systems, conflicts arise naturally from negative feedback, which is inherent to connections between passive components [12]. This step is implemented as a breadth-first search.

Vertices that are measured to be normal can be used to bound the search: If a vertex has an observed *normal* value, none of the component parameters that affect its value could have failed. Though this restriction reduces the set of implicated components, it may result in missing the actual fault if in contrast with the observation, in actuality the vertex *does* deviate. This deviation may not be observed at the moment of observation, e.g., because the value of the vertex has a wider range of margin or because the presence of noise. So, in this case, additional robustness measures or constraints have to be implemented.

### 2.2.2 Structural Changes

When backpropagating deviating variable values, so-called controlled junctions may be crossed. These controlled junctions represent discontinuities, or structural changes, of a physical system. Since these junctions have a parameter associated with them (i.e., ON or OFF), they can be implicated in the backward propagation step. To this end, the controlled variables (e.g., $f_5$ in Fig. 2) have to be part of the causal graph even when they are OFF. Now, if a controlled junction is ON and its associated variable (effort for a 0-junction and flow for a 1-

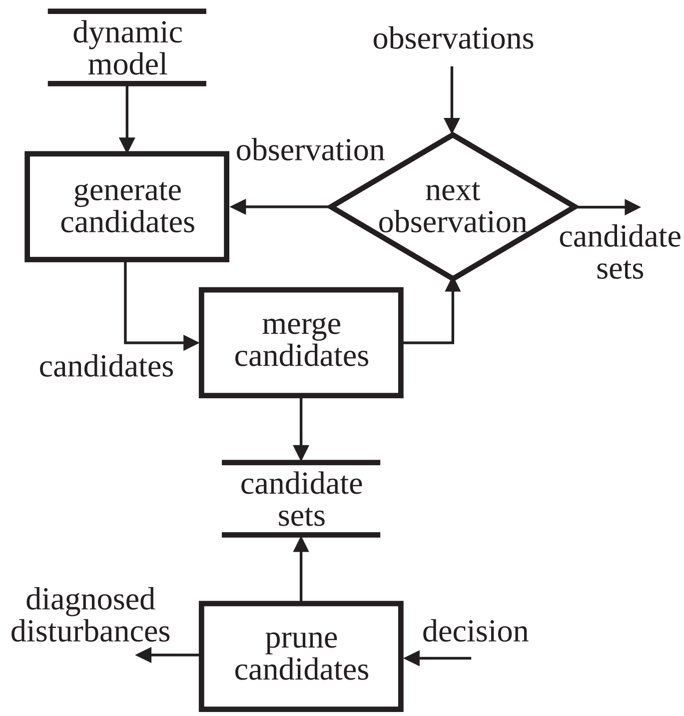



Figure 5: **Flow diagram of the diagnosis module.**

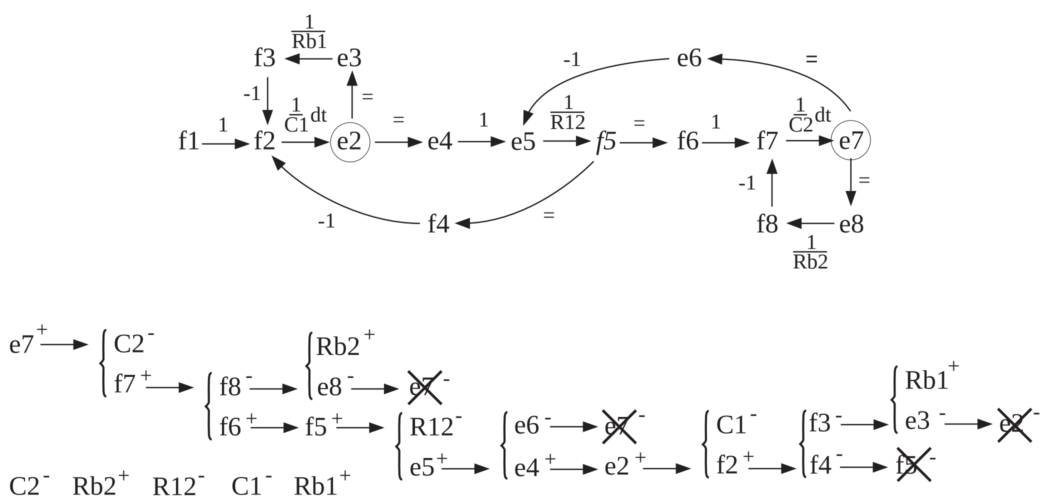



Figure 6: **Backward propagation of a deviation to implicate components.**

junction) is below normal, then the junction is implicated as possibly being OFF. Analogously, if a controlled junction is OFF and its associated variable is above normal, this junction is hypothesized to be ON.

### 2.2.3 Generating Sets of Candidates

After all candidates for each deviating observed variable separately have been compiled, these sets of candidates can be simply merged. This may result in conflicting candidates (e.g., $R_b^+$ and $R_b^-$). Monitoring will prune this set down to one or several candidates. Alternatively, it may be more effective to find all minimal sets of causes and monitor their combined effect. Generally, there are a number of faults that are consistent for several measurements. Additionally, there are causes for seperate observations that have conflicting effects. For example, $R_{b1}^-$ may explain one observation and $R_{b1}^+$ another. Only one of the two can be part of the set explaining both observations. Moreover, any other candidates that would have the same effect on an intermediate variable as the refuted candidate are not eligible either. In this case, if the outlow was observed to be high, but $R_{b1}^-$ was rejected then $C_1^-$ is inconsistent as well. So, generating sets of multiple faults requires to establish assumptions under which each of the candidates holds.

Issues involved are dealt with extensively in literature and can be implemented in this diagnosis framework.

## 2.3 Prediction

Next, these sets of candidates are fed into the prediction module. A flow-diagram of this module is shown in Fig. 7. One of the main functions of the prediction module is finding the correct mode of operation of the process: When a dramatic fault occurs, the process may undergo several structural changes instantaneously. The prediction has to first identify these changes before it can give a valid prediction of the process' dynamic initial behavior.

However, these structural changes occur not only at the moment of failure but also between this time and steady state. So, after the initial mode changes are found, a steady state analysis is performed to identify the eventual mode of the process, so as to give a steady state prediction. Notice that in a purely qualitative setting and a highly discontinuous system this is likely to be impossible and so *unknown* steady state predictions result.

### 2.3.1 Forward Propagation

After component and junction parameters are implicated, a prediction of future behavior for each of these can be determined. This prediction can then be used in the monitoring part to eliminate candidates.

In the bond graph framework, this future behavior in its most basic form can be described as behavior of derivatives. To this end the temporal dependency as incorporated in the causal graph, which results from elements with integrating behavior, is exploited in the prediction algorithm. A forward propagation of the deviating value as caused by each of the components, yields future behavior for all vertices of the system, including the set of observed vertices. Initially, the value of the deviation pertains to the 0-order derivative. Now, every time an

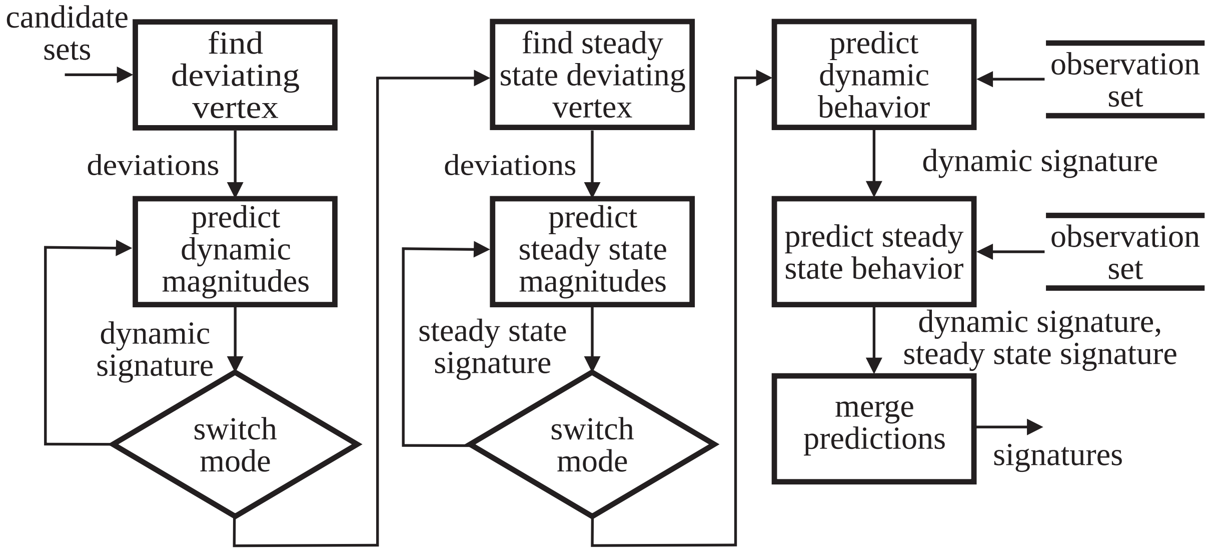



Figure 7: **Flow diagram of the prediction module.**

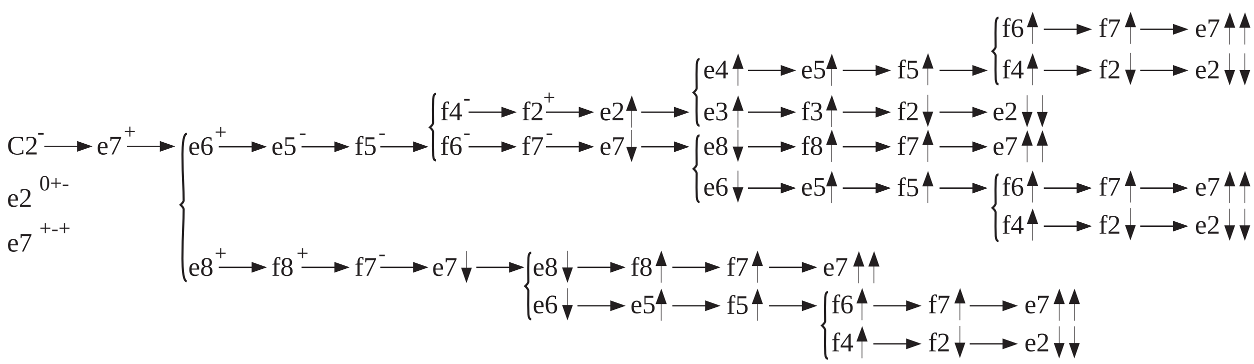



Figure 8: **Forward propagation of the effect of an implicated component to establish its signature.**

integrating edge is traversed, the order of the derivative is increased. Fig. 8 shows this as one or two arrows for respectively $1^{st}$ and $2^{nd}$ order derivatives.

This process of forward propagation with increasing derivative ends if a signature of sufficient order is achieved. The order that may be required depends on whether *progressive monitoring* is used. In that case, the maximally needed order depends on the maximal number of integrating edges in a specific path from a component parameter to an observed vertex. These derivatives are then used to find magnitude and derivative values of an observed variable. So, when the highest order of the prediction matches the maximal number of integrating edges in a path from a candidate set to an observed variable, this signature will eventually include a non-normal value. A detailed description of progressive monitoring is given in the monitoring section.

Progressive monitoring is best used if there is no reliable algorithm for detecting discontinuities available. If progressive monitoring is not used, the highest order of derivatives that can be used depends on those computable from the monitored values. In the basic form these include $0^{th}$ and $1^{st}$ order derivatives. So, in that case propagation to derive up till the first order derivative is sufficient.

When assigning values to derivatives, vertices may exist that have a particular assigned deviation for higher order derivatives, whereas the lower order derivatives are not marked. These lower order derivates are considered to be normal (i.e., non-deviating). Notice that for continuous physical systems this forward propagation is internally consistent. However, in case of discontinuities, a conflict may arise where opposing values are assigned to one vertex. In this case the vertex is marked as unknown and not used in the monitoring part. It is important to state that propagation of the *unknown* assignment takes precedence over any of the other qualitative values, *low*, *normal*, *high*.

As an example, consider the behavior of the latched bi-tank system in Fig. 2 when the latch is closed to the left and the right tank has a much larger capacity-outflow resistance combination than the left tank. In that case, the pressure in the right tank is slowly decreasing but let's consider this pseudo steady state. Then, if the pressure in the left tank suddenly decreases the latch might open. Now, in the causal graph the prediction shows $e_2$ to be low. So because of the $e_2 \rightarrow e_3 \rightarrow f_3 \rightarrow f_2$ loop, the first order derivative of $e_2$ is predicted to be rising. However, because the discontinuity of the latch turns $f_5$ ON, $f_5$ is high rather than low. Through $f_4$ and $f_2$ this results in a predicted first order derivative that is falling. Notice that this phenomenon is only observed because of the structural change that is present, which makes the physical system inconsistent.

Another effect of discontinuities applies to the $1^{st}$ order derivative propagation: A $1^{st}$ order derivative that is being propagated across a controlled junction that has turned OFF becomes 0, regardless of its value before crossing the junction.

### 2.3.2 Structural Changes

Structural changes may occur as the result of the forward propagation. Based on the assignment of deviating values, a controlled junction may switch ON or OFF. Whenever such a switch occurs, forward propagation from the implicated component under scrutiny has to be performed again until finally a stable (i.e., no more switches of controlled junction states) mode is reached. Now, forward propagation is executed once more to find the actual signature. Consistent with the hybrid bond graph theory [7], all intermediate modes are considered to have been virtual or mythical. So, they are never actually realized by the system.

Also, because of the inherent ambiguity in qualitative values, mode switching may not result in a particular mode. For example, consider the control specification of the latch in the bi-tank system. Because of the resistive effect of the connecting pipe, the pressure in the left tank in steady state is higher than in the right tank. Now, if the model predicts a decrease of the pressure in the left tank and an increase of the pressure in the right tank, then it is unknown whether the latch will close or not, Fig. 9. In this case the state of the controlled junction is ambiguous, it is marked as *unknown* and consequently used in the prediction algorithm.

The ordering between the pressures in each of the tanks is crucial for inferring mode switches. If it is not known whether $p_1 > p_2$ in steady state, $p_1 \uparrow$ would also result in an unknown situation. This ordering is derived from the steady state causal graph. All power

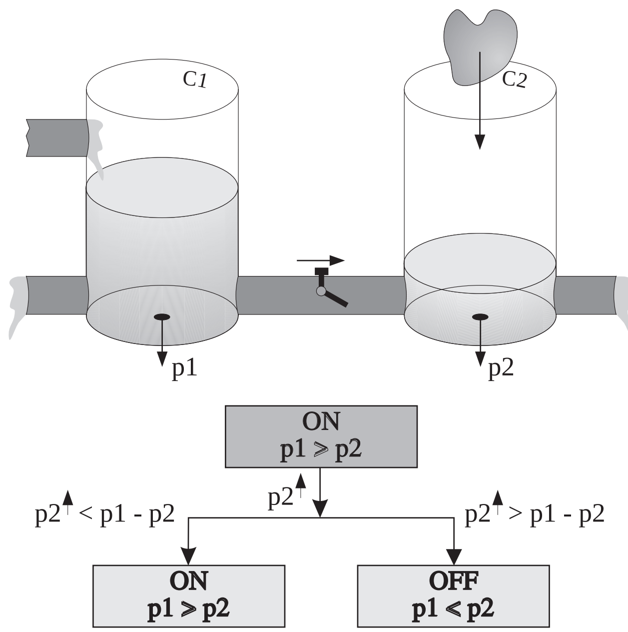



Figure 9: **No conclusive state because of the qualitative nature of the variables.**

flows are considered oriented in their positive direction, which means that the conjugating variables (effort and flow) have to both be either positive or negative. Now, starting at the sources, it is known whether a particular effort or flow is positive, this can then be propagated throughout the steady state graph. After establishing positive and negative magnitudes of variables, relations between them can be established based on the summing junctions that are part of their connecting path. If any of the above assignments is ambiguous, the ordering becomes ambiguous and the prediction loses power.

If, during the final forward propagation, a controlled junction is reached that has switched from OFF to ON (with regard to the initial state), the propagated value becomes *high*. If the controlled junction has switched from ON to OFF, the propagated value becomes *low*. If no switch has occurred, the junction is dealt with as a normal junction. Finally, if a controlled junction is reached that is marked *unknown*, the junction is marked as having a conflicting value and this value is then further propagated. Notice that in each of these cases, the *unknown* value is propagated throughout the temporal causal graph, taking precedence over any other assignment.

### 2.3.3 Steady State

After a signature of the derivatives of an implicated component is established. The steady state causal graph is used to find the final value of each observed variable for a fault. Again, the steady state graph may undergo a series of structural changes before the final graph is found. All intermediate graphs are not considered in determining the eventual steady state value. This predicted steady state value is now attached to the signature of each observed variable and can be used for monitoring purposes. Notice that because of the possibly qualitative nature of the propagated values, controlled junctions may be in an unknown state in which

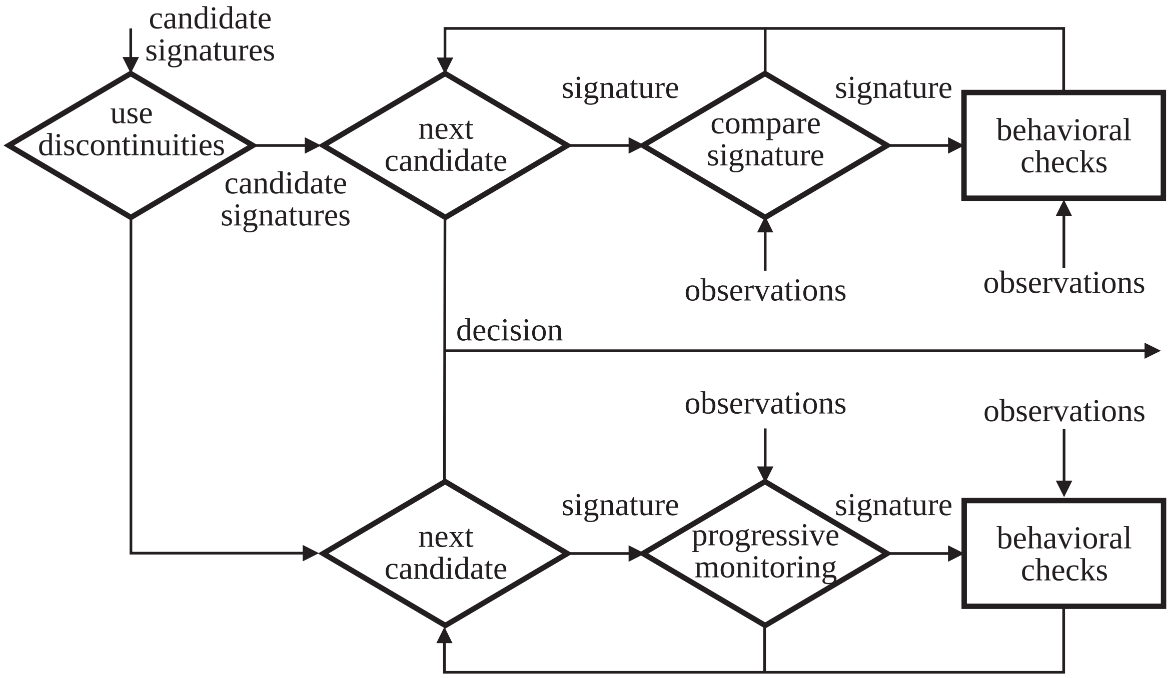



Figure 10: **Flow graph of the monitoring module.**

case the steady state value is not available. Now, monitoring revealing parameters of the system may yield valuable information about the actual mode of operation which then can be used to determine this very discriminative effect.

## 2.4 Monitoring

Monitoring of predicted behavior is essential to diagnosis. As inputs, the monitoring module requires the measured variables at each time step and their predicted behavior for each possible component (or set of components) failure. Then, it is required to find characteristics of the observed variables such as discontinuities and steady state to match against these prediction and to eliminate ineligible candidates. Because of the presence of noise or when discrete time steps are used, certain robustness measures have to be regarded.

The prediction of the future behavior of an observed variable is called its signature. These signatures are used in the monitoring module. Monitoring can be based on the assumption that discontinuities are detectable in the observed signals, Fig. 10. In this case, a straightforward match between observed and predicted behavior can be applied. However, sometimes this assumption is invalid, in which case the algorithm applies so-called *progressive monitoring*. In both of these cases, certain behavioral checks are included to correctly deal with, e.g., inverse responses. Notice that these responses do not have to be identified a priori, they can be checked for on-line. Based on (1) the characteristics of the signal, (2) the features of the prediction, and (3) the validity of the monitoring algorithms, a decision is made whether each candidate set is conform with the observations. This information is then sent to the diagnosis module so it can prune the overall set of candidate sets.

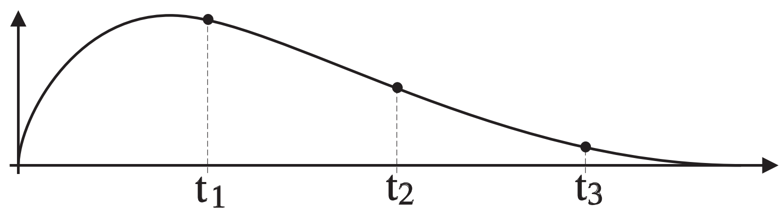



Figure 11: **Low frequency sampling of an inverse response.**

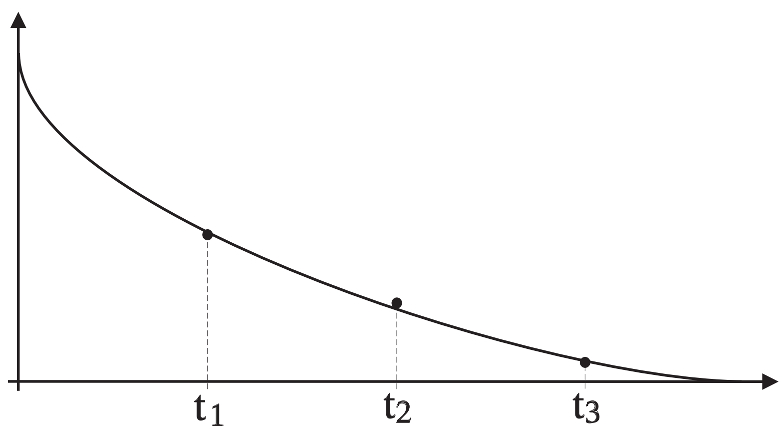



Figure 12: **Interpretation of low frequency sampled inverse response as a discontinuity.**

### 2.4.1 Sensitivity

When monitoring, selecting the correct sensitivity in discrimination is of paramount importance. This sensitivity is generally the result of different rates of responses of the system and the time step that is chosen in the monitoring module. Too small a time step may result in insensitivity to changes and too large a time step may cause incorrect inferences.

As an example, consider the signal as shown in Fig. 11. If the sample time for monitoring is chosen too large, this signal may be considered discontinuous (Fig. 12). Though decreasing the time step may detect a possible discontinuity, it may be too small to apply to a, relatively, slowly decreasing slope as shown in Fig. 13. Now, the signal may appear to have not deviated yet, i.e., it is monitored to be normal, or to have reached steady state whereas in actuality it is decreasing.

These phenomena emerge in the context of robustness as follows: Because of noise, in combination with a certain time step and margin of error, a signal may intermittently show a first order derivative that is *normal* while slowly decreasing or increasing. To refute a hypothesized candidate based on this intermittent observation could result in an incorrect diagnosis. To prevent this, a robustness option is used which attaches a certain belief value to a candidate based on the consecutive sample times at which the predicted values did not match the observations. This value is decreased whenever the predictions and observations match. When a preset threshold is crossed, the candidate is refuted. Notice that, e.g., sophisticated

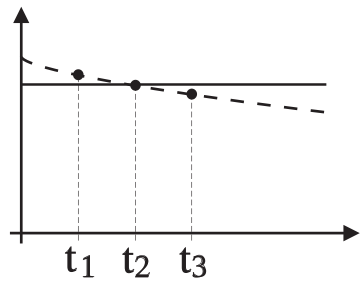



Figure 13: **Interpretation of high frequency sampled discontinuity may imply that a new steady state is reached when in actuality the signal is falling.**

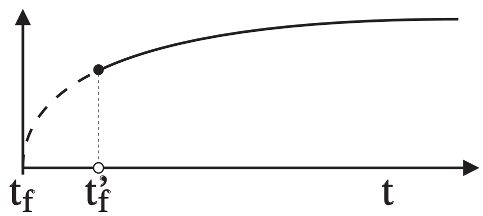



Figure 14: **Identifying a continuous response to a failure as discontinuous for lack of knowledge of the actual time of failure.**

Bayesian schemes for believe factors can be implemented at this stage as well.

Furthermore, the time step may affect correctly identifying a deviation from normal. Because of the differences in time scale of the system responses, a signal may appear to be normal over an initial range of samples. This data could be used to weed out suspect candidates. However if the system is known to include these differences in time scale, it may be preferable not to attach any diagnostic value to such information.

The opposite effect is noticeable when the actual moment of failure is not known a priori. In that case, a signal is assumed to deviate instantaneously whereas the underlying failure was present earlier, and therefore caused an initially normal signal value, Fig. 14. However, because this signal value was normal, it was not detected when the fault occured and the information about the moment of failure is unavailable. Again, sophisticated extrapolation methods may yield this information after several more samples.

### 2.4.2 Discontinuities and Progressive Monitoring

Discontinuities in signals have a very discriminative power in diagnosing abrupt failures. For example, consider the one-tank system in Fig. 15. While measuring the pressure at the bottom of this tank, a discontinuous change of this observable can only be caused by an abrupt change of tank-capacity. Any other change is reflected in pressure by an integrating effect, which smoothes the abrupt change. Notice that measuring the outflow would not yield this discriminative information since a discontinuous change can be caused by either an abrupt change of tank-capacity or a change of outflow resistance. However, if multiple outflow resistances are connected, no distinction can be made between which one of them has

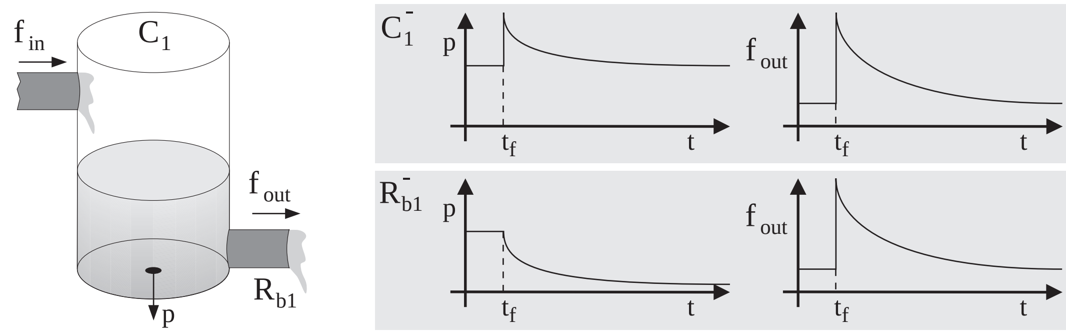



Figure 15: **Characteristic response of pressure and outflow to two different faults.**

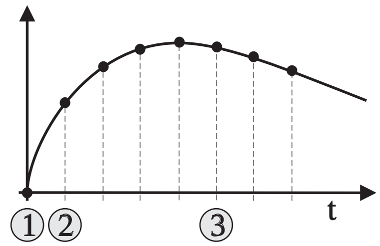
failed. On the other hand, in case of several resistances, steady state behavior *does* differ. So, the flow measurement can be used to discriminate if time permits.[3] Now, there is additional information about whether the resistance that is associated with the outflow resistance has failed or not. So, additional discriminative information becomes available and the importance of measurement selection becomes obvious.

Since discontinuous behavior can be inferred from the temporal causal graph, it can be applied to serve as great use in pruning the search space. However, detection of discontinuities is not trivial and is very much dependent upon the appropriate selection of the monitoring time step. In the general diagnosis framework described, the detection of discontinuities can be used and becomes an issue in the monitoring module. Pragmatically, it is implemented as detecting a $0^{th}$ order and $1^{st}$ order derivative which have values with opposing signs. As long as the $1^{st}$ order derivative is normal, no assertion is made. However, specific signal analysis techniques may be more appropriate to detect discontinuities [8].

Because of the difficulties detecting discontinuities, this phenomenon may have to be disregarded in certain situations. Now, the $0^{th}$ order derivative does not represent the value of a variable at the time of failure, but it represents the predicted value of the magnitude of the signal at any given time. Since this value changes over time, the signatures of the behavior of each of the observed variables for a candidate fault change dynamically. For example, a variable may have a $0^{th}$ order derivative which is normal and a $1^{st}$ order derivative which is above normal. So, over time, this variable will become above normal in magnitude, i.e., its $0^{th}$ order derivative becomes above normal. Including this effect of higher order derivatives is referred to as *progressive monitoring*. It replaces derivatives that do not match with the observed value with the value of derivatives of one order higher in their signature. An example of this is shown in Fig. 16, where at time stamps marked 1, 2, and 3 a lower order effect is replaced by a higher order effect that has become manifest. When the higher order derivative does not match the observed value either, the candidate is refuted.

Progressive monitoring is implemented as becoming effective when there is a discrepancy between a predicted value and a monitored value (this does not only apply to 0 order derivatives but also higher orders). At that time, it is checked whether the next higher derivative could make the prediction consistent with the observation. If this next higher derivative value

[3] It is obvious that in case of certain failures it is preferred not to wait for the process to reach steady state but to shut down immediately.




| dt | 0 | 1 | 2 |
|---|---|---|---|
| ① | 0 | + | - |
| ② | + | + | - |
| ③ | + | - | - |

Figure 16: **Progressive monitoring replaces signature values of higher order derivatives.**

is normal, it is moved on the next higher derivative value, until either a conflict in prediction and observation, a confirmation, or an unknown value is found.

### 2.4.3 Temporal Behavior

Signals that occur in response to disturbance demonstrate several characteristics that have to be included in the monitoring module either because they carry distinctive discriminative information or because they might lead to incorrectly refuting candidates. These phenomena include: (1) initial deviations, (2) inverse response, and (3) steady state.

When monitoring, it is important to know the initial deviation of a signal. However, since the time of failure may not be known, the initial deviation may not be known. Because of the improvements in the monitoring algorithm that stem from this knowledge, it is beneficial to invest some effort in measurement selection and sensor placement to acquire this knowledge. Also, signal processing techniques may be used to estimate this time of failure from a signal trace.

As was found by Kramer, a signal may exhibit an inverse response: After an initial increase or decrease, it reverses its direction. Because the predicted behavior of a fault entertains the behavior of the signal at the time of failure (Fig. 17) only, this information is not known and when monitoring it may lead to refuting a valid hypothesis. To prevent this, the monitoring module is furnished with a detection mechanism that cancels the refutation process based on a signal when an inverse response of that signal is detected. An algorithm that reveals more detailed information about signal behavior would be an improvement to this heuristic approach. Possible approaches to finding more detailed information about signal behavior as time progresses are:

- Mason's Loop rule, which applies well to bond graphs [4, 2].
- Estimating eigenvalues, which can be done from the structure of the generalized bond graph [14].
- Decoupling states by a transformation process that diagonalizes the system matrix.

Though Mason's Loop rule yields an exact transfer function, it becomes very extensive for larger scale systems. Especially, the characteristic polynomial of the system becomes large. Since this characteristic polynomial is determined by the system's eigenvalues, it may be more

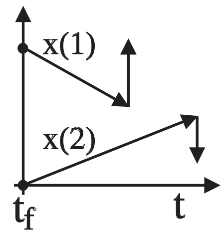



Figure 17: **Conceptual representation of diagnosis from time of failure, $t_f$, with $0^{th}$, $1^{st}$, and $2^{nd}$ order derivative.**

efficient to estimate these eigenvalues and use them in an approximation of system behavior. The disadvantage here is that, in order to apply the estimation technique, the bond graph has to be transformed into its generalized bond graph equivalent. Decoupling states results in only local feedback in which case the transfer function transforms into a summation of first-order effects. The inverse of the decoupling transformation then has to be applied to find the effect on the actual observable state variables. This inverse transformation may render the advantage of a summation of first-order effects void. These options need to be investigated.

The final concept in monitoring concerns steady state. Detection of steady state in the architecture is invoked when a $1^{st}$ order derivative with value 0 is detected. Then, there is a steady state counter that increases by 1 every time this combination is observed and decreases by 1 otherwise, until 0. Furthermore, every time the counter is increased, the observation time step is made larger to 'zoom out.' If the counter crosses a specific threshold, or incubation time, steady state is assumed to have been reached. More sophisticated methods are optional and can be implemented at this stage.

## 2.5 Conclusions

The prediction-monitoring-diagnosis architecture as put forward in this section establishes a coherent framework for model-based diagnosis of dynamic physical systems. The use of hybrid bond graphs for the modeling results in physically consistent and constrained models. The prediction features of

- discontinuity,
- higher order derivatives, and
- steady state,

are all automatically derivable from one hybrid bond graph model. These features have great merit in discriminating between fault candidate sets based on monitored values. Furthermore, the architecture leaves room for implementation of sophisticated algorithms where desired or required. Additionally, the bond graph models allow for basic qualitative operations as well as symbolic or quantitative implementation.

```
ACTUAL =>
   3: 0 . . .
  12: 1 . . .
==============
dt/C2+  =>
   3:  0 1 -1 0
  12:  1 -1 1 0
1/Rb2-  =>
   3:  0 0 1 1
  12:  0 1 -1 1
1/R12+  =>
   3:  0 -1 1 -1
  12:  0 1 -1 1
dt/C1+  =>
   3:  1 -1 1 0
  12:  0 1 -1 0
1/Rb1-  =>
   3:  0 1 -1 1
  12:  0 0 1 1
                (1)
```

```
ACTUAL =>
   3: 0 0 . 0
  12: 1 0 . .
==============
dt/C2+  =>
   3:  0 1 -1 0
  12:  1 -1 1 0
1/Rb2-  =>
   3:  0 0 1 1
  12:  0 1 -1 1
1/R12+  =>
   3:  0 -1 1 -1
  12:  0 1 -1 1
dt/C1+  =>
   3:  1 -1 1 0
  12:  0 1 -1 0
1/Rb1-  =>
   3:  0 1 -1 1
  12:  0 0 1 1
                (2)
```

```
ACTUAL =>
   3: 1 1 . 0
  12: 1 -1 . 1
==============
dt/C2+  =>
   3:  0 1 -1 0
  12:  1 -1 1 0
1/Rb2-  =>
   3:  0 0 1 1
  12:  0 1 -1 1
                (3)
```

```
dt/C2+  =>
   3:  0 1 -1 0
  12:  1 -1 1 0
                (4)
```

Figure 18: **Results of the prediction-monitoring algorithm when $C_2^-$ and $p_1$ and $p_2$ are measured.**

# 3 Monitoring the Simulated Latched Bi-Tank System

To study the effectiveness of the prediction-monitoring-diagnosis architecture as described before, it was applied in monitoring and diagnosing a number of faults introduced in the latched bi-tank system, Fig. 2. First, the system equations were derived based on the hybrid bond graph representation. These equations were used to simulate the behavior of the system. Next, a separate prediction and monitoring module was created which takes as input only the measured variables and their steady state values. For the entire algorithm, only the qualitative values of these observables are used $(-, 0, +)$. In steady state, the latch is opened to the right and there is a constant flow from the left tank to the right. Several scenario's can be considered, permutations of the observed variables and the introduced deviating component parameters.

## 3.1 Completely Observable System

First, the pressures in both of the tanks are observed. In this case both of the states of the system are known and monitoring of a prediction result in finding the introduced fault. In this situation, correctly diagnosing the fault does not hinge on the use of discontinuities. Moreover, to diagnose capacitance errors does not require sensitivity to normal observations either, neither does a faulty connecting resistance. However, if the algorithm is not sensitive to normal observations, it is unable to distinguish between faults in the outflow resistances: The effects of such faults are very similar, the discrimating factor being the delay time before the effect is observed in either of the tanks. This effect is incorporated in the sensitivity to normal observations. Fig. 18 shows an example of the prediction-monitoring output for a sudden decrease in tank capacity $C_2$. The output of the prediction-monitoring module shows the prediction of the behavior of the observed variables for each of the candidates. The numbers represent respectively their $0^{th}$ order, $1^{st}$ order and $2^{nd}$ order derivates, where $-1, 0, 1$ maps onto *low, normal, high* and a period means *unknown*. The fourth number shows the predicted steady state value of each signal.

Fig. 19 shows the diagnosis of the same fault with the same observables. However, now

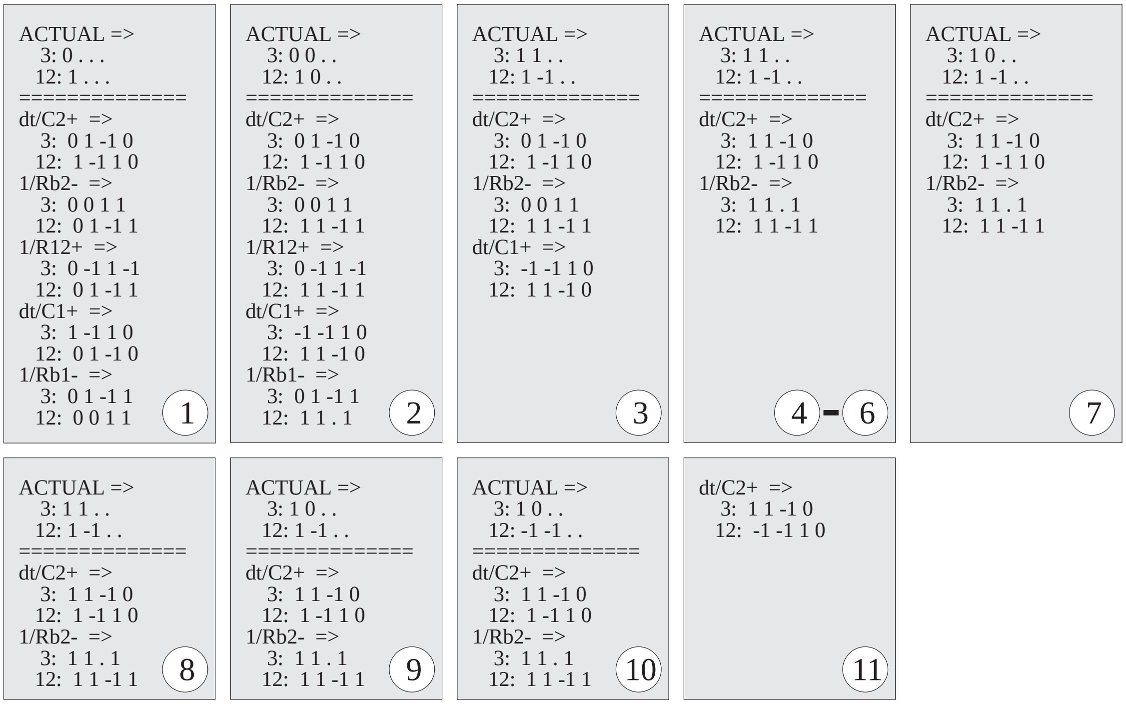



Figure 19: **Results of the prediction-monitoring algorithm when progressive monitoring is applied to a fault $C_2^-$ and $p_1$ and $p_2$ are measured.**

discontinuities are not used and therefore progressive monitoring is applied. Notice how at times the signatures of the observed variables are modified because of this. For example, the signature of $R_{b1}^+$ between step 1 and step 2 from $0, 0, 1, 1$ to $1, 1, ., 1$. So, the $2^{nd}$ order derivative, which is positive, is assumed to have affected the magnitude and $1^{st}$ order derivative to make the candidate consistent with the observation $1, ., ., .$. Notice that the fourth number represents steady state value and therefore does not partake in the progressive monitoring scheme. Furthermore, because discontinuities cannot be detected, the fourth number of the actual observation, which represents the initial deviation, is left unknown.

## 3.2 Not Completely Observable System

Now, using discontinuities and sensitivity to normal, if the pressure in only the left tank is measured, a sudden decrease in $C_2$ is also found. Notice that because of the sudden increase in pressure in the right tank the latch in the connecting pipe closes, so $p_1$ is subject to a structural change. In case the algorithm does not utilize disontinuities, thanks to this structural change, a decrease in $C_2$ is still correctly diagnosed. In case of any of the other capacity failures it cannot be distinguished between $C_1$ and $C_2$ anymore.

Worse, for the parameters given and when only observing the pressure in the right tank, an observation time step that is too large will lead to incorrectly diagnosing a decrease in $C_2$ when in actuality $C_1$ has decreased. In this case, the smallest time constant of the system results from the connecting resistance (10) and the failing capacity ($5*10^{-8}$). So, in $1.5*10^{-6}s$ the tank level has dropped within 5% from its original value. With a margin around *normal*

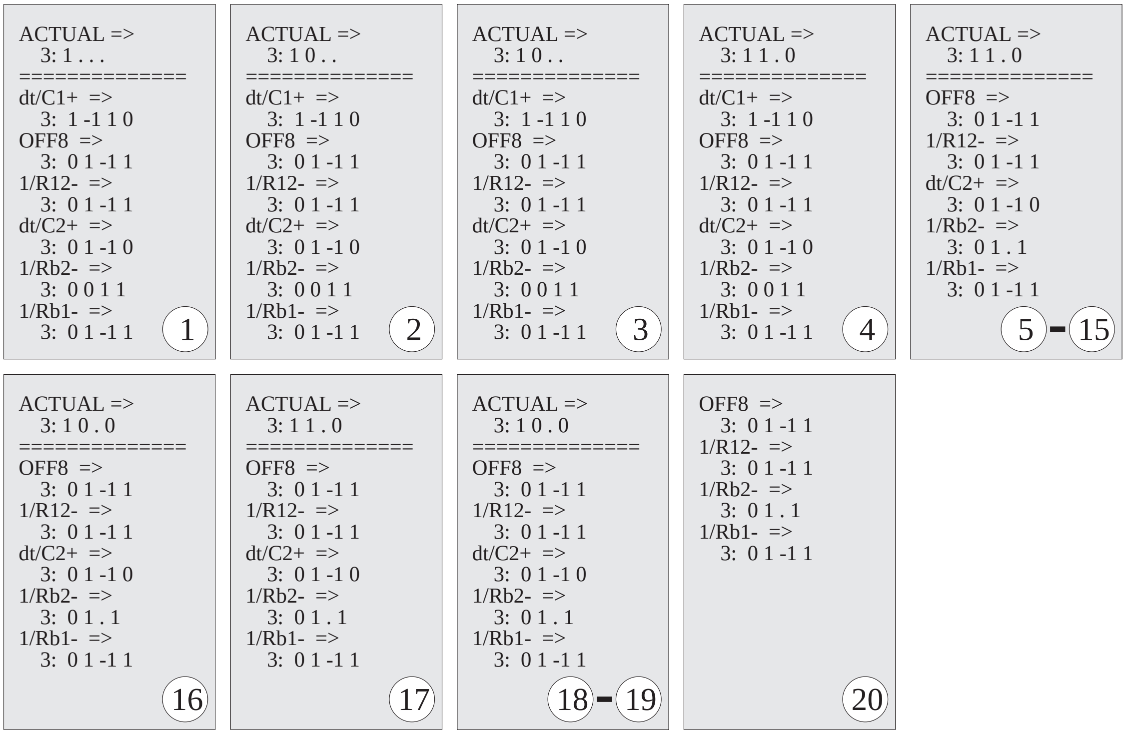



Figure 20: **Results of the prediction-monitoring algorithm when $R_{b1}^{+}$ and $p_1$ is measured only.**

of 5%, this means that an observation time step of $1 * 10^{-5}$ is too long to observe any of the falling effect. When doing the same experiment with a time step of $1 * 10^{-6}$ the fault is diagnosed correctly.

As shown by this example, it is important to establish the fastest time constant of a system to serve as a criterium by which the observation time step has to be chosen in order to guarantee the presence of the correct set of failing components amongst the sets of candidates.

In other cases where only one measurement is available, and the algorithm uses discontinuities and is sensitive to normal, prediction and monitoring may not reduce the set of candidates to one. In all cases, the actual fault is part of this set of candidates though. Fig. 20 shows the results for an increase of the outflow resistance $R_{b1}$ when measuring the pressure in the left tank only. Notice that based on this single observation, several faults are eligible candidates. In line with the discussion on measurement selection, making a distinction between resistance faults is not possible based on the qualitative information. However, the fault *can* be identified if the outflow $f_{out1}$ is measured instead of the pressure, Fig. 21.

## 3.3 Conclusions

From the results of the simulated latched bi-tank example, it is clear that the architecture as put forward in this paper provides good results. Even if just qualitative values *low, normal, high* are available, a single fault is still diagnosed correctly. As the observation time step is crucial to successfully diagnosing a fault, it is important to find the fastest time constant of

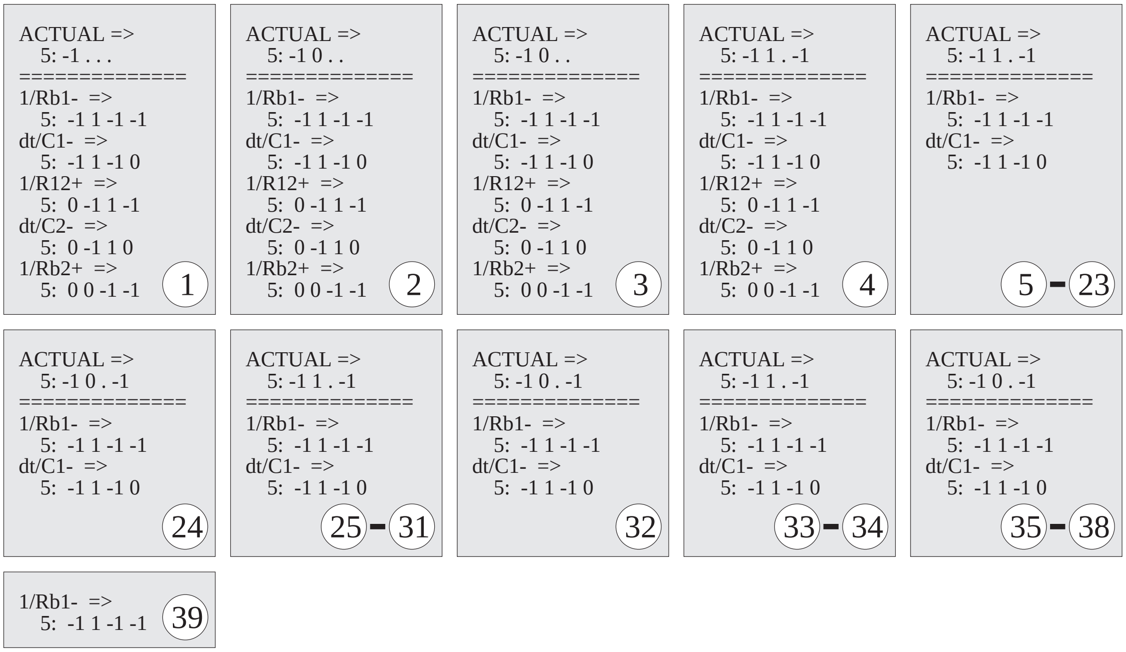



Figure 21: **Results of the prediction-monitoring algorithm when $R_{b1}^{+}$ and $f_{out1}$ is measured only.**

a fault that has to be captured by the diagnosis algorithm. This time constant can then be used to as a criterium for the observation time step. Measurement selection was also shown to make the difference between correctly diagnosing a fault or being unable to discriminate faults of the same type. Important issues are whether discontinuities can be detected and whether it is reasonable to wait until it is determined what the new steady state of the system will be.[4] Then, based on the causal strokes in the bond graph model, variables to be measured can be selected.

# 4 Application to a Fast Breeder Reactor System

In order to apply the described prediction-monitoring architecture to an industrial application, a sub-system of a plant is most appropriate. A comprehensive control approach involving all aspects of an entire plant is too ambitious at this point. Moreover, because of the compositionality of the method, each sub-system that is tackled is another step towards a complete prediction- monitoring system on a plant-wide scale.

For this initial study, the secondary sodium cooling system of a fast breeder nuclear reactor plant was selected, Fig. 22. This sub-system is transparant enough for an initial study, yet includes several difficult to model aspects (like convective energy flow and multi-port energy buffering) to be of interest. Furthermore, it contains mode switching capabilities where a safety, air cooling, system can be activated.

[4] At times it may be possible to establish this new steady state before it is actually reached, which saves valuable time.

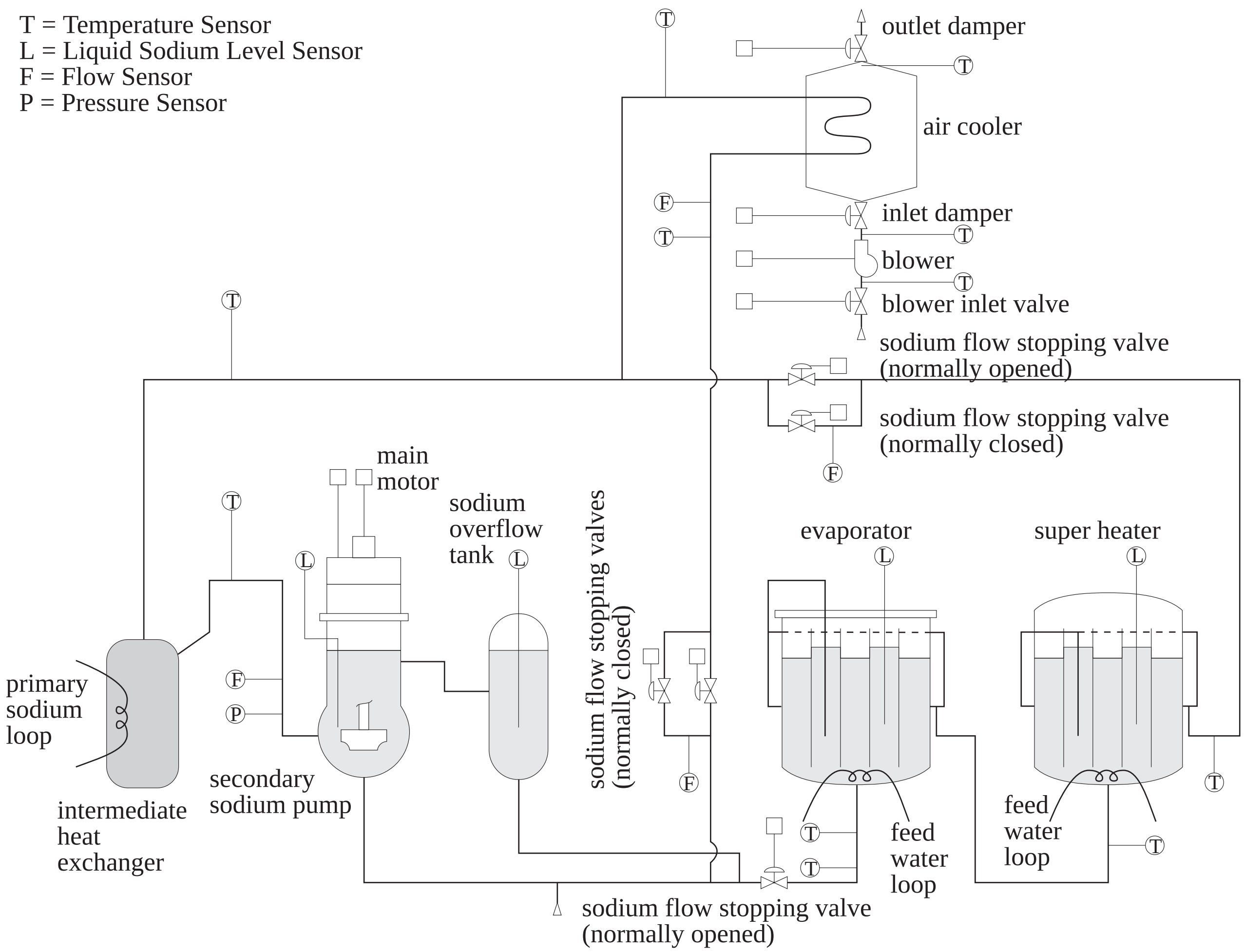



Figure 22: **Engineering representation of the secondary sodium cooling sub-system.**

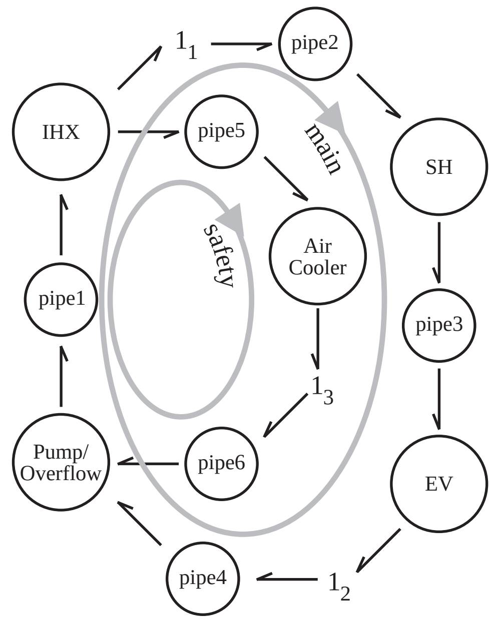



Figure 23: **Word bond graph of the secondary sodium cooling system.**

The most important variable in this process is the flow rate of the liquid sodium. Though this variable can be measured in pipes with a small diameter, it becomes difficult when the sub-system is scaled to a larger size. After modeling the hydraulic and thermal aspects of the process, this model can be used to predict the flow rate of the liquid sodium.

## 4.1 The Model

A word bond graph representation of the secondary sodium cooling system in Fig. 22 is given in Fig. 23. In this figure, the 1-junctions represent valves, or flow connections that can be turned off. In the following, the main loop is singled out and considered first.

Because of the cooling nature of the system, it involves two physical domains. First there is the hydraulics part which causes the liquid sodium to flow. Then, there is the thermal domain which applies to extracting heat from the primary loop and yielding it by the secondary loop.

Since the variable of primary interest is the flow rate of the liquid sodium, only a model of the hydraulics part could be applied. If this model proves to be inadequate, there are two options: (1) The hydraulic domain should be modeled in more detail, or (2) a model of the thermal domain should be included. In case of the first option, a more refined model of the connecting pipes can be established by using a multiple compartment approach. However, since the flow measurements are few, there is no need to find a more detailed distribution along these pipes. Because of the availability of several temperature measurements, the second option is more promising. The most important issue to resolve here is that the thermal energy flow involves convective energy transport which can be hard to model.

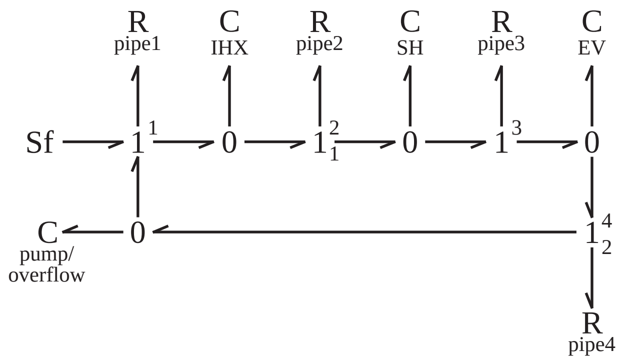



Figure 24: **Bond graph model of the hydraulics aspects of the main loop in the secondary sodium cooling system.**

### 4.1.1 The Hydraulic Domain

Fig. 24 shows a bond graph representation of the hydraulics of the main loop. In this bond graph model it is assumed that the flow into the superheater and evaporator enters at the bottom of the tank, and therefore is pressure dependent. Notice that since this is a model of the hydraulics of the main loop only, none of the thermal aspects are included.

Based on this bond graph model, prediction can proceed: The flow of the $1^1$-junction is observed in the actual process (Fig. 22), as well as the pressures on all 0-junctions except for the one connecting intermediate heat exchanger (IHX). However, because of the temperature measurements, a more detailed model can be obtained by including the thermal aspects.

### 4.1.2 The Thermal Domain

The energy flow in the thermal domain breaks down in two parts: (1) Conductive energy flow, and (2) convective energy transport. Conduction of thermal energy is much like energy flow in any other domain: Because of a potential difference between two intensive variables, an intensive variable tries to compensate by transporting energy (e.g., a difference in voltage causes a flow of charge). This conductive phenomenon is exhibited by the heat exchange between the primary sodium loop and the secondary sodium loop, as well as between all the capacities (including pipes) that contain liquid sodium and the surroundings. Furthermore, because of the lumped parameter assumption, there is conduction of energy between the capacity of the tanks and their connecting pipes (i.e., a lumped tank capacity temperature is different from the lumped pipe capacity temperature although at the actual point of connection there is no temperature difference). The bond graph in Fig. 25 models these energy conducting phenomena.

Convective energy does not flow based on a difference in potential of an intensive variable, rather it is transported by a flow of matter.[5] This matter flows because of a pressure (or any other) potential difference which relates to the amount of matter in terms of its volume, but not to the amount of thermal energy that the matter contains. To model this appropriately,

[5]For example, in a Van den Graaff generator, charge is moved by means of a belt rather than a difference in voltages.

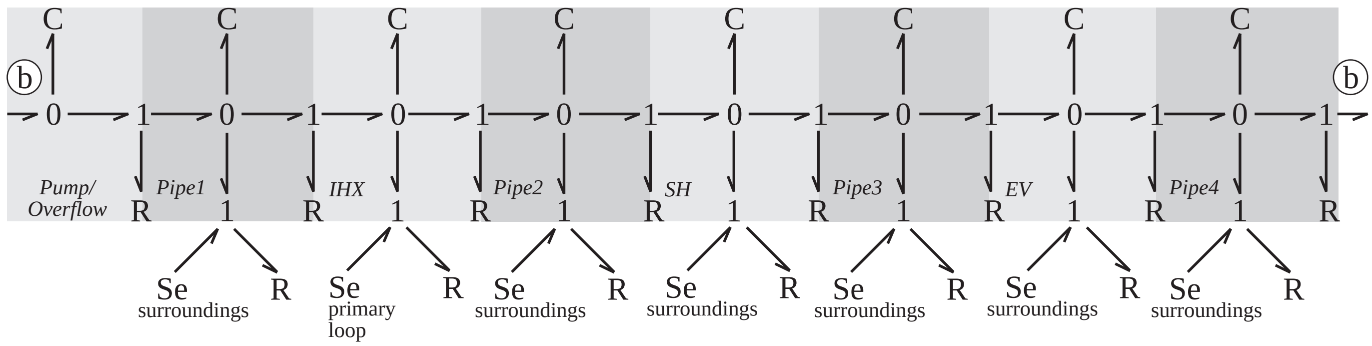



Figure 25: **Bond graph model of the thermal domain conductive phenomena of the second sodium cooling system.**

the flow of thermal energy is represented by modulated flow sources that are controlled by the amount of matter that is transported. The specific entropy of this matter determines the actual amount of thermal energy that is moved. Since this specific entropy is temperature dependent, the source is modulated by two signals. The specific entropy of flowing matter is based on the temperature of the surroundings that it exits. Alternatively, the temperature of the surroundings that it enters can be used, as long as this choice is consistent throughout the model.

A bond graph model of the entire thermal domain (conductive as well as convective phenomena) for the main loop of the secondary sodium cooling system is given in Fig. 26. At the top the convective energy flows are depicted by modulated flow sources marked with indices that correspond with the 1-junction superscripts in the hydraulic domain bond graph (Fig. 24). The model part below the capacities is concerned with the conductive phenomena of the thermal domain.

Notice that although the capacities in the thermal domain are represented seperate from the ones in the hydraulic domain, they truly represent the same physical component. Moreover, they represent the same matter and therefore their respecitive energies are interchangeable. Therefore, the appropriate depiction would be a multi-port buffer, or one C with several energy bonds connected. So, theoretically, these pairs of capacities ought to be represented as one with two energy bonds for each of the domains, plus an energy bond for the energy convection phenomenon. However, because of the assumed energetic independence between the hydraulic domain and the thermal domain in this process, the pairs of capacities are represented as such.

## 4.2 Prediction

Now that the model is established, causality can be used and prediction can proceed from there. For example, consider the situation where the temperature in the pipe connecting the intermediate heat exchanger (IHX) and the super heater (SH) is measured to be low. What does this mean in terms of the liquid sodium flow, and which faults could have caused this?

To globally work this problem, the conducting resistances between the liquid sodium in the pipe and in the respective tanks (IHX and SH) can be regarded very high. Now, Fig. 27

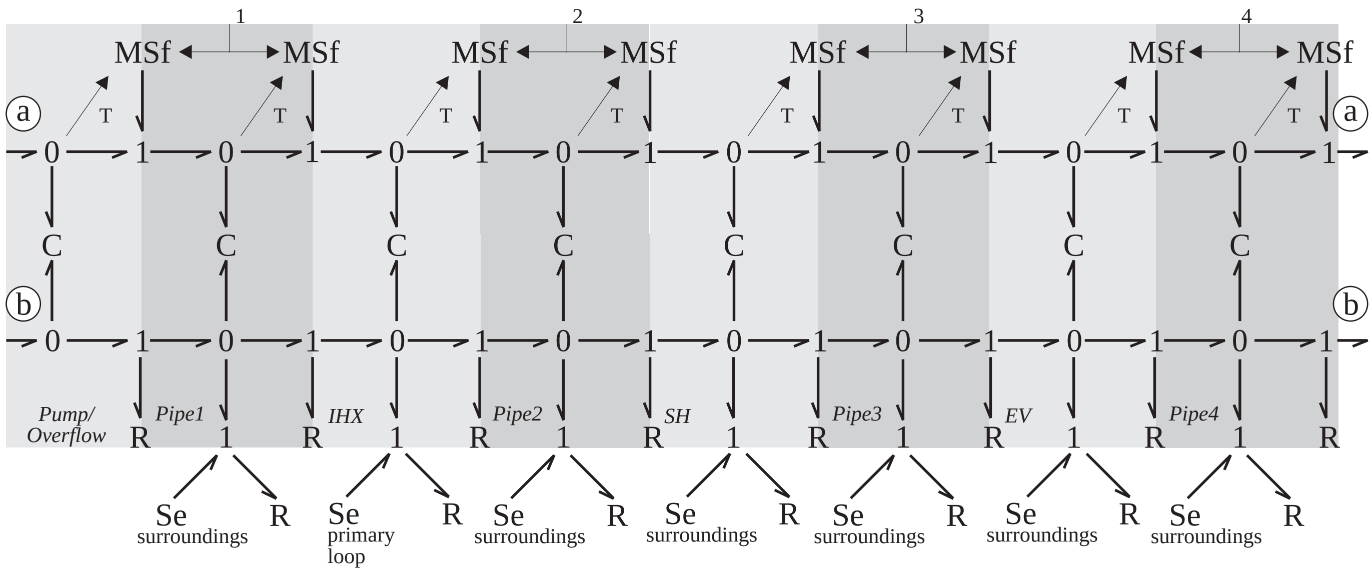



Figure 26: **Bond graph model of the thermal domain of the second sodium cooling system.**

shows the bond graph model of this situation, where the effort on bond 4 is measured low. To propagate this effect the causal graph of this model can be established, Fig. 28. Notice that when this causal graph is formed, two paths with opposite effects are found between two nodes. To effectively propagate any deviations, this path has to be reduced to one edge with the aggregated relation. Now, propagation depends on the sign of this relation, i.e., the ordering of the parameters. In this case the specific entropy of liquid sodium at the IHX end, $s_{i,IHX}$, is higher than the specific entropy at the end of the pipe, $s_{i,pipe}$. So, the aggregated relation is positive. As the remaining part of the temporal causal graph results in an academic exercise, it is not completed.

After the temporal causal graph is established, backward propagation is used to generate fault candidates. Based on $e_4^-$, either the thermal pipe capacity has increased, $C_{pipe}^+$, or any of the inflows has decreased, $f_4^-$, or $f_5^-$. In case of $f_4^-$ this results in $f_3^+ \rightarrow f_2^+ \rightarrow (R_{insulation}^-, e_2^+) \rightarrow e_3^+ \rightarrow e_4^+$ and propagation ends because of a conflict with the observation $e_4^-$. In case of $f_5^-$, either the aggregated relation $s_{i,IHX} - s_{i,pipe}$ has become low or $f_{10}^-$ (because $s_{i,IHX} - s_{i,pipe}$ is positive). This yields either $R_{pipe2}^+$ as a candidate or $e_{10}^-$. Further propagation then finds more candidates. Notice that normal measurements can be used to bound this search space drastically. So, if the flow as generated by the main motor is normal ($f_{18}$) as well as the pressure in the super heater ($e_{15}$), then the set of candidates is $C_{pipe}^+, R_{insulation}^-, (s_{i,IHX} - s_{i,pipe})^-, R_{pipe2}^+, C_{IHX}^+$.

Next, the behavior of the model variables can be predicted for each of the candidates. In terms of $f_{10}$, only the candidates $R_{pipe2}^+$ and $C_{IHX}^+$ have an effect. Both of these result in a flow that is low. However, in case of the resistance failure, the steady state will differ from its original, whereas in case of the capacitance failure the steady state that is reached after a transient phase is the same as before the failure.

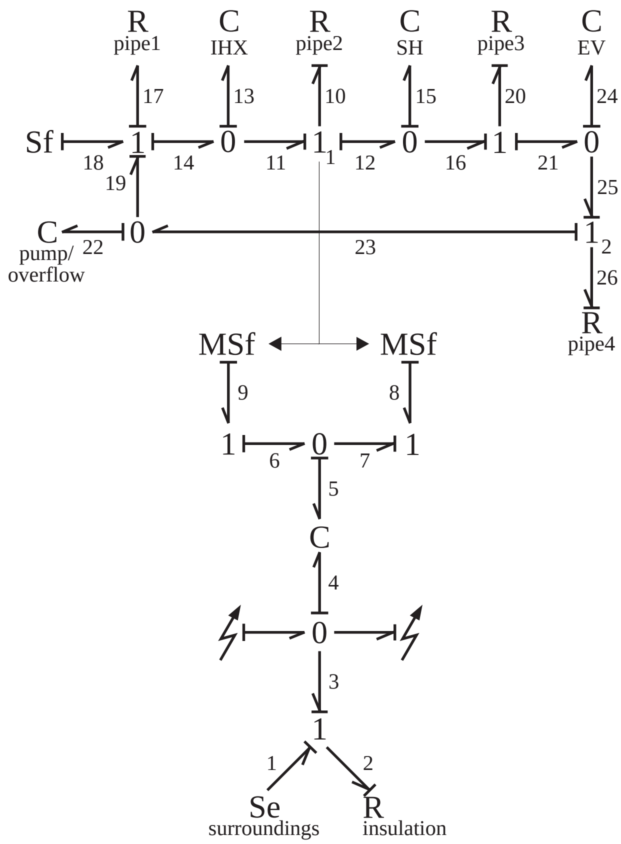



Figure 27: **Bond graph model of the hydraulic aspects of the second sodium cooling system with the thermal model of one pipe included.**

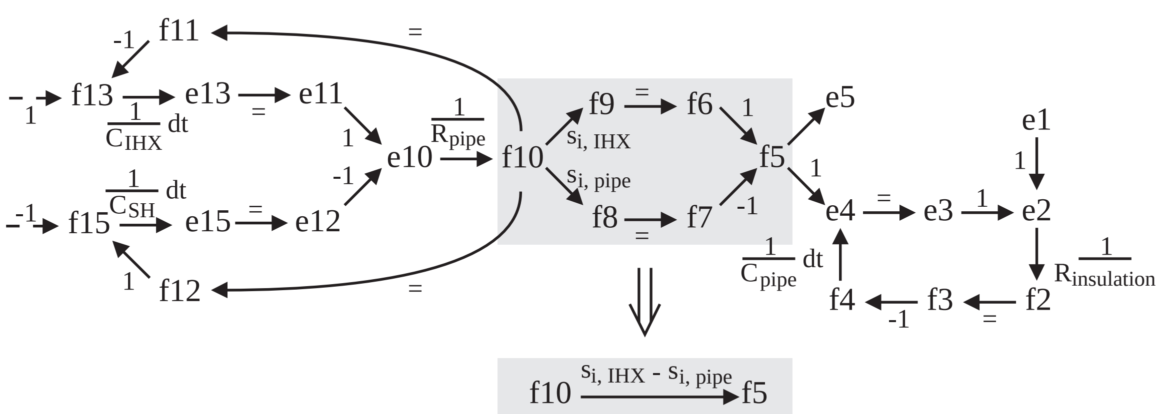



Figure 28: **Partial temporal causal graph of the second sodium cooling system.**

# 5 Summary and List of Tasks

As demonstrated, the hybrid bond graph formalism provides a solid basis for predicting system behavior. The introduction of convective energy flows can be dealt with within this formalism.

The finite state automata or the temporal causal graph may introduce relations that require a certain ordering of parameters to predict behavior.

Behaviors exhibit characteristics that have a very discrimating nature. Detection of these characteristics may be difficult or impossible. Based on which characteristics are detectable by the monitoring module, the prediction can be adapted and the monitoring procedure may change.

To implement a prediction and monitoring system for a secondary sodium cooling system in a nuclear reactor plant, first the bond graph model has to be established. Next, if complete numerical information is unavailable, certain parameter orders have to be established so qualitative prediction improves.

# 6 Schedule of Activities

The schedule of activities for this project are illustrated in Fig. 29. Note that the project is divided into two phases: (i) initial development based on the thermal and fluid system of the sodium cooling loop, and (ii) extension to other subsystems of the fast breeder reactor. At the end of part (ii), we will also develop general software systems for modeling, diagnosis, prediction, and monitoring of physical systems.

# References


[1] Gautam Biswas, Stefanos Manganaris, and Xudong Yu. Extending component connection modeling for analyzing complex physical systems. *IEEE Expert*, 8(1):48–57, 1993.

[2] F.T. Brown. Direct application of the loop rule to bond graphs. *Journal of Dynamic Systems, Measurement, and Control*, pages 253–261, 1972.

[3] Dean Karnopp and Ronald C. Rosenberg. *Analysis and Simulation of Multiport Systems.* John Wiley and Sons, New York, 1975. ISBN 0-471-45940.

[4] Samuel J. Mason. Feedback theory - further properties of signal flow graphs. In *Proceedings of the IRE*, pages 920–926, July 1956.

[5] Jakob Mauss and Bernd Neumann. Diagnosis by algebraic modeling and fault-tree induction. In Wolfgang Nejdl, editor, *DX-95 Sixth International Workshop on Principles of Diagnosis*, Goslar, Germany, October 1995.

[6] Pieter J. Mosterman and Gautam Biswas. Behavior generation using model switching a hybrid bond graph modeling technique. In François E. Cellier and José J. Granada, editors, *1995 International Conference on Bond Graph Modeling and Simulation (ICBGM '95)*, number 1 in Simulation, pages 177–182, Las Vegas, January 1995. Societey for Computer Simulation, Simulation Councils, Inc. Volume 27.

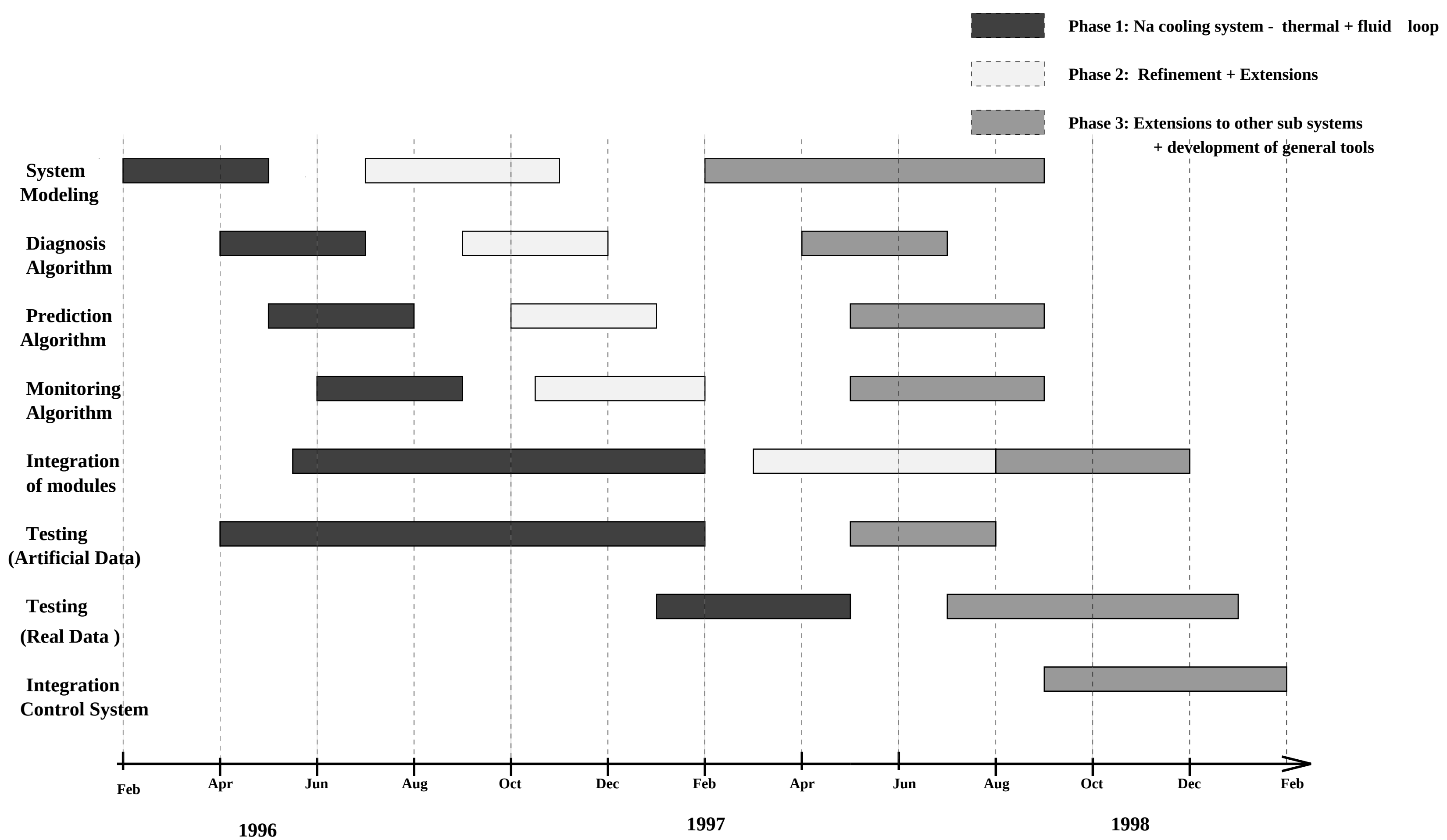



**Fig. 29: Schedule of Activities**

[7] Pieter J. Mosterman and Gautam Biswas. Modeling discontinuous behavior with hybrid bond graphs. In *1995 International Conference on Qualitative Reasoning*, Amsterdam, May 1995. University of Amsterdam.

[8] H. Nawab and V. Lesser. Integrated processing and understanding of signals. In A. Oppenheim and H. Nawab (eds.), editors, *Symbolic and knowledge-based signal processing*. Prentice-Hall, 1992.

[9] B.L. Palowitch. *Fault Diagnosis of Process Plants using Causal Models*. PhD dissertation, Massachusetts Institute of Technology, August 1987.

[10] Henry M. Paynter. *Analysis and Design of Engineering Systems*. The M.I.T. Press, Cambridge, Massachusetts, 1961.

[11] Ronald C. Rosenberg and Dean Karnopp. *Introduction to Physical System Dynamics*. McGraw-Hill Publishing Company, New York, New York, 1983.

[12] J.J. van Dixhoorn and P.C. Breedveld. *Technische Systeemleer*. University of Twente, Twente, Netherlands, fourth edition, February 1985. class pack.

[13] K.C.J. Wijbrans. *Twente Hierarchical Embedded Systems Implementation by Simulation: a structured method for controller realization*. PhD dissertation, University of Twente, CIP-DATA Koninklijke Bibliotheek, Den Haag, The Netherlands, 1993. ISBN 90-9005933-4.

[14] Ashraf Zeid and Ronald Rosenberg. Estimating eigenvalues for a class of dynamic systems. *Journal of the Franklin Institute*, 320(1):21–40, July 1985.