Predicting Dynamic Behavior Using Physical System Models with Algebraic Loops

Pieter J. Mosterman[1] Eric J. Manders[2]
Gautam Biswas[3]
[1]Institute of Robotics and System Dynamics
DLR Research Center Oberpfaffenhofen
P.O. Box 1116
D-82230 Wessling, Germany
Pieter.J.Mosterman@dlr.de
[2] Department of Electrical and Computer Engineering,
Vanderbilt University
400 24th Avenue South, Room 218
Nashville, Tennessee 37235
manders@vuse.vanderbilt.edu
[3]Knowledge Systems Lab
Gates Computer Science Bldg, 2A
Stanford University
Stanford, CA 94305
biswas@ksl.stanford.edu

February 4, 1999

Abstract

Our system for monitoring and diagnosis of abrupt faults in complex dynamic systems, TRANSCEND, relies on system models to predict dynamic behavior in response to abrupt faults. The ability to express fault transients in a qualitative framework mitigates complexity issues and convergence problems that arise in numerical diagnosis approaches. Predicted behavior for a fault is captured as a *signature*, an ordered tuple of magnitude and higher order changes. A progressive monitoring scheme uses uses these signatures to refine the hypothesized causes by dropping candidates whose signatures are inconsistent with the measurements. A key to this methodology is to make the level of detail in the model and the measurement bandwidth correspond. If the model contains higher order effects beyond the measurement bandwidth, it will predict continuous transients that appear as discontinuous changes in the measured signal. However, when the small parameter values responsible for the higher order transients are abstracted away, algebraic dependencies may occur between system variables, i.e., their variable values may have instantaneous relations without intervening integrating states to introduce time delays in the relation. When these algebraic dependencies appear in a compensating loop, prediction of future behavior in a qualitative framework becomes ambiguous. This paper describes an algorithm that recognizes such dependencies, and propagates compensating effects in a manner that avoids a number of ambiguities. The effectiveness of the approach is demonstrated by diagnosing a punctured hose in the cooling system of an automobile combustion engine.

1 Introduction

TRANSCEND uses a model based approach fault detection and isolation of abrupt faults in engineered systems [3]. It applies models of dynamic system behavior to obtain accurate predictions for measured transients and compares predictions with actual observations to distill the true cause for the faulty behavior. To successfully perform diagnosis, TRANSCEND uses a model of dynamic behavior for the system. A qualitative framework is employed for hypothesis generation, prediction, and fault refinement to avoid numerical estimation and convergence problems. But a qualitative framework, unless properly constrained, tends to predict spurious behaviors and leads to combinatorial explosion in the search space [2]. We have used the bond graph modeling approach to generate physically correct and well constrained models in our modeling and diagnosis applications [4, 6].

An important aspect of monitoring and diagnosis tasks is to ensure that the level of granularity in the model corresponds to the bandwidth or the sampling rate of the measured signals. The time constants associated with the system behaviors determine the sampling rate for the measurement siganls. Conversely, if a cetain sampling rate is chosen or desired, the model needs to be adjusted so that it does not include parameters (and, therefore, physical phenomena) that result in much faster time constants. For example, the inertial effects of fluid flow in a pipe may not be of much consequence in terms of the overall behaviors of interest. Therefore, the sampling rates for the pressure and flowrate measurements would be chosen so that the inertial effects do not show up as transients. Similarly, the analysis model should be designed so that the inertial effects are not explicitly modeled.

When creating lumped parameter models of physical systems, abstracting away small parameters may have little consequence on overall behavior. If they are included in more detailed models, these parameters would temporally decouple signals by introducing an integrating effect, but they also lead to an increase in the size of the state vector. A consequence of abstracting them away is that it creates instantaneous relations between system variables [12]. If these variables are such that they are in a compensating loop, the negative feedback causes a prediction of conflicting behaviors in a qualitative framework, where changes are represented as $\pm$ values. Therefore, they cannot be applied for refinement of fault candidates in our diagnosis framework.

In numerical simulation, this problem is solved by algebraically manipulating the equations to achieve an equation structure where certain parameters are aggregates of the original parameter values [1]. In a diagnosis framework this is inconvenient because a deviating aggregate parameter implicates several components. It would be advantageous to handle the implicit equations directly, resulting in more discriminating information on a component level.

This paper describes an algorithm that is implemented in the TRANSCEND diagnosis system that addresses this issue. It relies on first detecting the algebraic loops and then applies augmented prediction criteria to select the initial deviation and terminate compensating effects before they overwrite the prediction. The effectiveness of the approach is demonstrated by diagnosing a punctured hose in the cooling system of a combustion engine.

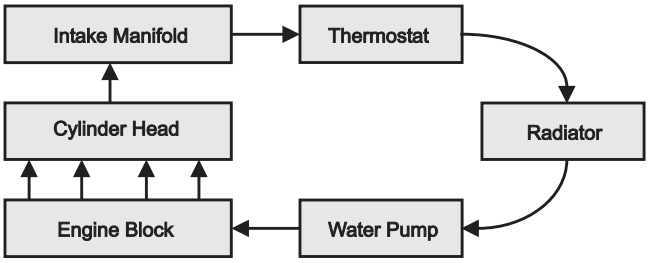

Figure 1: The engine cooling system block diagram.

2 Modeling Complex Systems for Diagnosis

TRANSCEND uses a diagnosis model that captures qualitative dependency relations between component parameters and the observed variables in the form of a temporal causal graph (TCG). Predictions are symbols that represent (i) discontinuous changes, (ii) magnitude deviations, (iii) slope deviations, and (iv) steady state behavior. A progressive monitoring scheme compares actual observations against the predicted transients for each hypothesized fault. Only those faults whose transients agree with the observations are retained. The goal is to continue monitoring till the true fault is isolated. A detailed description of the modeling scheme and diagnosis methodology with examples can be found in [5, 6]. A critical part for successful diagnosis in this scheme is the model used to compute nominal operating values and predict future behavior in case of faults. This is discussed next.

2.1 The Automotive Engine Cooling System

We have constructed a testbed at Vanderbilt University around an Chevrolet V8 internal combustion engine. The system is engineered to include a set of pressure and temperature sensors in the cooling system.

2.1.1 The System

An automotive cooling system uses a liquid coolant pumped through passageways in the engine block by a centrifugal pump that is driven by the crank axis. Fig. 1 shows a block diagram of the fluid path. The hot coolant flows through the radiator (heat exchanger) where it is cooled by air blown through by a fan. The coolant is then pumped back into the lower part of the engine block where it moves towards the back, flows through the passageways in the cylinder, and heads up into the intake manifold toward the front of the engine. If the engine is operating at its desired temperature, the thermostat is open and the coolant circulates back to the radiator. If the engine has not reached its desired temperature, the thermostat is closed and the warm coolant is pumped back into the engine block to facilitate quick warm up of the engine.

2.1.2 Bond Graphs for Diagnosis

We have adopted the bond graph methodology [11] to system modeling. Bond graphs provide a systematic framework for building consistent and well constrained models of dynamic physical

systems across multiple domains. Topological causality constraints, systematically derived from component descriptions, and structural connectivity expressed in bond graph models are exploited for effective and efficient diagnosis. The bond graph representation is directly amenable to qualitative reasoning. A fault is represented as a deviation of a component parameter in the bond graph model. In our qualitative framework this means a component parameter deviating above (+) or below (−) its nominal value.

2.1.3 The Cooling System Bond Graph

The bond graph of the cooling system, presented in Fig. 2, describes a lumped parameter model that encompasses the mechanical, hydraulic, and thermal subsystems. The hydraulic part, by forcing coolant through the engine blocks, conveys heat away by convection. The heat transfer rate is a function of the fluid flow rate. The influence from the thermal subsystem to the hydraulic subsystem, such as fluid expansion with increasing temperature, is not modeled. This implies that discrepancies detected on measurements in the thermal part cannot implicate parameters in the hydraulic part. The primary components of the system are shown as shaded boxes in Fig. 2.

In the bond graph, the crank axis is represented by an ideal source of torque, Se. This source drives the centrifugal pump, represented by the modulated gyrator, MGY, that models the conversion of mechanical energy to fluid energy [4, 6]. The inertia of the rotor in the pump is represented by mass m_1. It is assumed that all hoses and other similar fluid paths provide a linear resistance to flow, shown as fluid resistances, R, in the figure. Fluid storage is represented by capacitances. Resistance R_{l-hose} represents the fluid energy losses that occur during coolant flow through the lower hose. The small outlet of the radiator introduces a fluid inertia, $I_{rad-out}$. A T-split coupling in the lower hose is used to introduce a leak. Even when it is closed, it allows a small flow of coolant, modeled by resistance R_{leak}. On entering the engine block, the coolant is channeled into passageways, whose fluid resistance is represented by parameter R_{hy-blk}. The fluid capacitance is represented as a capacitor, C_{hy-blk}. The thermostat valve is modeled as a combination of a fluid inertia, I_{tstat}, and fluid resistance, R_{tstat}. When the thermostat valve is open, the coolant flows through the upper hose, represented by fluid resistance, R_{u-hose}, to the radiator modeled as capacitance, C_{hy-rad}.

In our lumped parameter model, conductive heat transfer occurs at two primary locations. First, heat is transferred from the combustion chamber of the engine to the liquid coolant. This is modeled as a constant flow of entropy, modeled by the source, S_f. The heat capacitance of the coolant in the cylinder head and block, modeled by C_{th-blk}, is a function of the specific heat of the coolant, the thermal capacitance per unit mass. The overall capacity to absorb heat is a function of the specific heat and the mass flow rate of the coolant. The mass flow rate is accounted for by a modulated transformer, MTF_2, whose modulation factor is a function of the coolant flow rate determined in the fluid subsystem. The heat capacitance of the coolant in the radiator, modeled by C_{th-rad} and MTF_1, in a similar fashion is a function of the specific heat of the coolant and its mass flow rate, determined by the fluid flow rate in the radiator. Heat transfer from the engine block to the radiator is attributed to the mass transfer of coolant from the engine block to the radiator, and is modeled by a modulated flow source, MSf_1, which is a function of the fluid flow rate in the upper hose. Analogously, convective heat transfer from the radiator to the engine block through the lower hose is modeled by MSf_2. The second

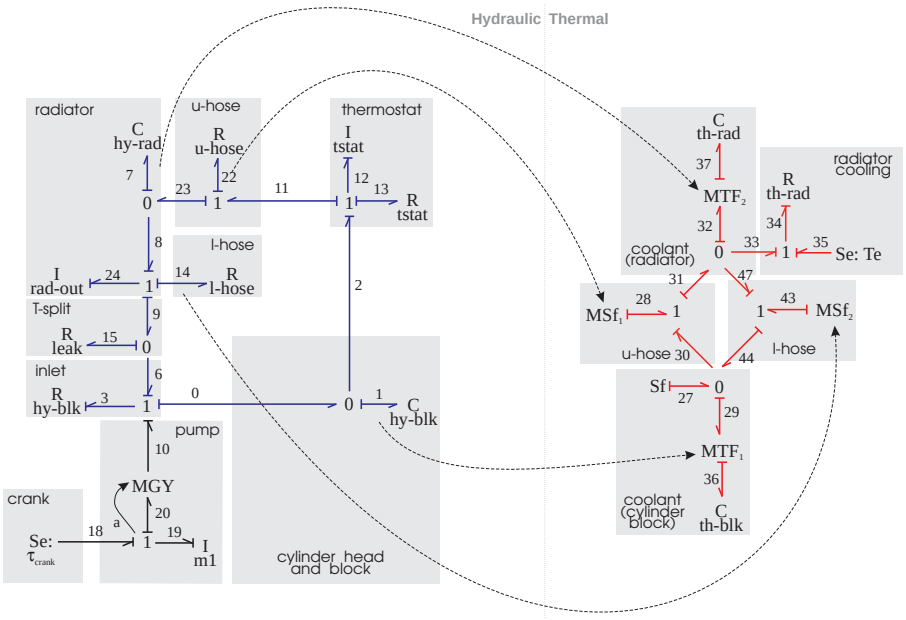

Figure 2: Bond graph model for the engine cooling system.

location where conductive heat transfer takes place is in the radiator, where the mass of coolant liquid, is cooled by the air blown through the radiator by the fan. The heat transfer occurs through the resistive junction R_{th-rad}, which is a function of thermal conductivity and radiator geometry. The outside air is represented as an infinite sink at constant temperature, T_a, and is modeled by an effort source. The non-linearity that results in unknown edge relations in the MGY model of the pump ([6]) are resolved by assuming straight veins for the water pump.

2.2 The Temporal Causal Graph

Dynamic characteristics of system behavior are represented as a *temporal causal graph* (TCG), algorithmically derived from the bond graph model [5, 6].

2.2.1 Causal Augmentation of the Bond Graph Model

A well defined algorithm, Sequential Causality Assignment Procedure (SCAP) [11, 12] is applied for assigning causality to the bonds in the bond graph model. This algorithm proceeds in the following steps: (i) fixed causality associated with exogenous variables is propagated, (ii) the preferred causality of the energy storing elements (state variables) is imposed, and (iii) resistive

elements are assigned a consistent causality. If it turns that there is more than one choice of causality assignment that can be made to certain elements, there are circular dependencies that imply the presence of *algebraic loops*. In such cases, arbitrary causality assignments are often made. Several heuristic methods have been suggested [12].

Causal assignment to the coolant loop bond graph (Fig. 2) reveals an algebraic loop that involves resistances R_{hy-blk} and R_{leak}. The algebraic loop can be removed by including small energy storage effects, such as the fluid capacitance of the lower hose. This results in added state variables to the system model, and introduces fast behaviors associated with these variables because of the small time constants involved. If the time constants introduced are beyond the measurement bandwidth, we have introduced modeled behaviors that cannot be observed, and, this may lead to incorrect diagnoses.

This implies that rather than add small energy storage effects to eliminate algebraic loops in a model, the implicit instantaneous relations between the flows and pressure drops associated with R_{hy-blk} and R_{leak} have to be handled by other means to avoid ambiguity in the qualitative prediction of system behaviors. In simulation systems, the two resistances would be aggregated into an equivalent resistance, but this is not useful for diagnosis. The fault isolation task requires that faulty parameters be linked back to physical components, and that becomes difficult when parameters are aggregated. In this paper, we present an augmented algorithm that overcomes the ambiguity problems introduced by algebraic loops, and predicting correct behavior.

2.2.2 Generating the Temporal Causal Graph

The TCG expresses causal and temporal relations between the variables of the system. The variables, *effort* and *flow* variables of the system, are the nodes of the graph, and the directed edges labeled by component parameters or by ± 1 and $=$ relations, express the functional relations that components and the junction structure (i.e., the system topology) impose on pairs of variables. Energy storage elements, like capacitances and inductances, impose an additional temporal relation between the variables. In addition, there may be *signal* vertices that correspond to modulated variables and other constraints that exist between system variables. The TCG representation provides a rich but uniform framework for representing magnitude and temporal constraints among system variables. The hypothesis generation and prediction algorithms for the fault isolation task are then developed as standard graph traversal algorithms.

The TCG of the automotive cooling system model is presented in Fig. 3. The hydraulic part is shown at the bottom and physical components are marked by shaded areas. The TCG is automatically derived from the bond graph model in Fig. 2 by tracing the causal strokes using the hybrid bond graph modeling and simulation tool HyBrSim [7].[1]

2.2.3 Predicting Future Behavior

The prediction algorithm employs the system model to compute the qualitative transient behavior of the observed variables under the individual fault conditions. We constrain the problem space here by making the assumption that faults do not cause changes in system configuration, and the system model remains valid even after faults occur in the system. Transient behavior is expressed as a tuple of qualitative values for magnitude, 1^{st} order time-derivative and higher

[1] Also see http://www.op.dlr.de/~pjm/hybrsim.

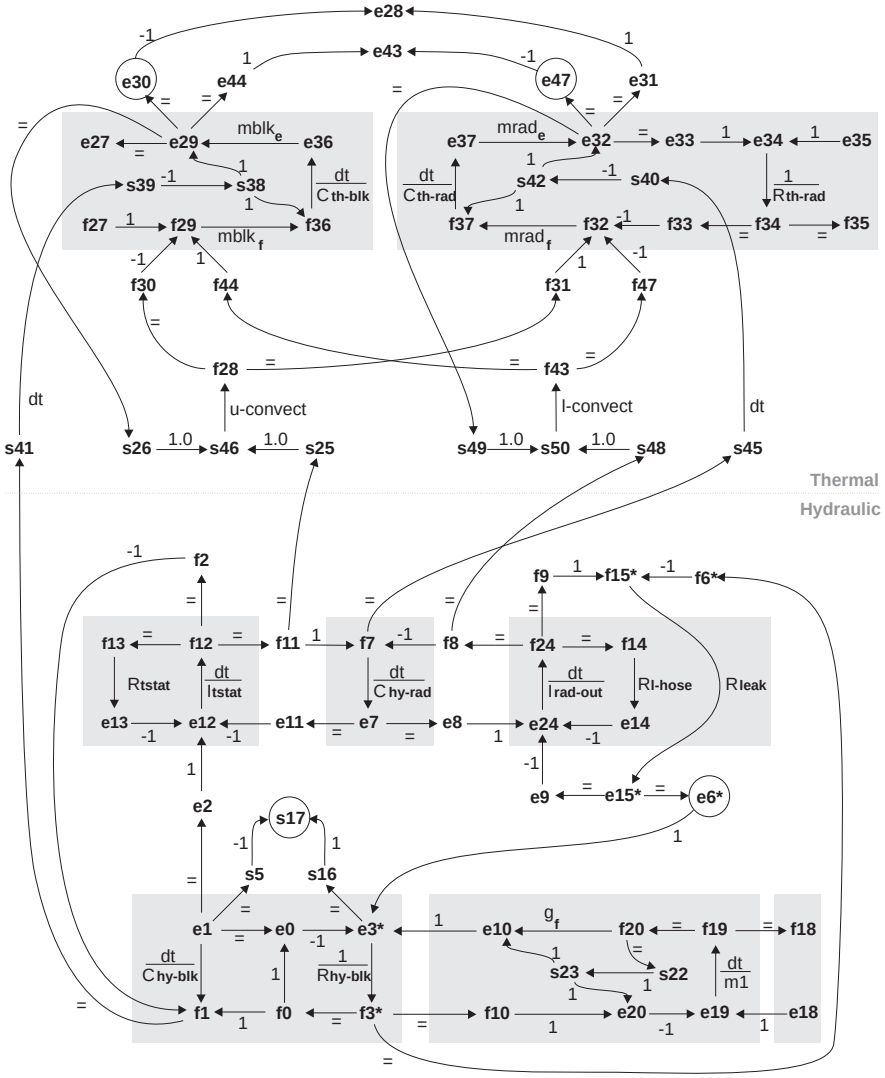

Figure 3: Temporal causal graph for the engine cooling system. Vertices that correspond to measured variables are circled.

order effects. The qualitative values are similar to those of the measured values: '+', '−', '0' or '·'. The '·' implies that the value is unknown, a result of opposite qualitative influences. The tuple is called the *signature* for the variable [5, 7].

The algorithm propagates the effects of a hypothesized fault to establish a signature for all observations. Energy storage elements, because of their integrating effects, introduce temporal delays in the propagation. This is expressed as teh cause variable affecting the derivative of the effect variable. Propagation of a deviation starts with a 0^{th} order effect, i.e., a magnitude change. When an integrating edge is traversed, the magnitude change becomes a 1^{st} order change, i.e., the first derivative of the affected quantity changes. Similarly, a first order change propagating across an integrating edge produces a second order change, and so on. The highest predicted derivative order required is a design consideration [4].

For example, consider a fault which changes the thermostat outlet dissipation (because of partial blockage). Theefore, R_{tstat} deviates in the positive direction, i.e., R^{+}_{tstat}. for the TCG

in Fig. 3, if opening), The effect of this deviation is now propagated in the TCG (Fig. 3) to predict future behavior.

$$R^{+}_{tstat} \rightarrow e^{+}_{13} \rightarrow e^{-}_{12} \rightarrow f^{\downarrow}_{12} \rightarrow \begin{cases} f^{\downarrow}_{13} \rightarrow e^{\downarrow}_{13} \rightarrow e^{\uparrow}_{12} \rightarrow f^{\uparrow\uparrow}_{12} \\ f^{\downarrow}_{2} \rightarrow f^{\downarrow}_{1} \rightarrow e^{\downarrow\downarrow}_{1} \\ f^{\downarrow}_{11} \rightarrow \begin{cases} f^{\downarrow}_{22} \rightarrow e^{\downarrow}_{22} \rightarrow e^{\uparrow}_{11} \rightarrow e^{\downarrow}_{12} \rightarrow f^{\downarrow\downarrow}_{12} \\ f^{\downarrow}_{23} \rightarrow f^{\downarrow}_{7} \rightarrow e^{\downarrow\downarrow}_{7} \end{cases} \end{cases} \tag{1}$$

First, R^{+}_{tstat} would cause the dissipative pressure to deviate above normal, i.e., e^{+}_{13}, causing the pressure across the thermostat to be low, i.e., e^{-}_{12}. The temporal, integrating, effect of I_{tstat} now causes a first order effect, i.e., a decrease in the rate of flow through the thermostat, $f^{\downarrow}_{12}$. This, in turn affects three fluid paths: (i) the dissipative flow decreases, $f^{\downarrow}_{13}$, (ii) the flow out of the block decreases, $f^{\downarrow}_{2}$, and (iii) the flow through the upper hose decreases, $f^{\downarrow}_{11}$. Each of these branches is traced till a sufficiently high order signature is generated. In Eq. (1) the prediction is terminated when a second order derivative is reached, which results in a prediction for observed variables of order 1.

3 The Algebraic Loop

In physical systems, feedback effects between passive components result in a negative gain [13]. Therefore, an initial deviation caused by a fault, diminishes because of these compensating effects. Typically, there are energy storage elements associated with these compensating feedback loops, and the compensating effects tend to manifest as higher order temporal terms. In situations, such as the one described in the last section, where the energy storage effects are small, the higher order effects may have very fast time constants. In designing a monitoring and observer system, the engineer may decided that the time constants of the fast effects are beyond the bandwidth of interest, and to achieve simplicity in the model and in computation, these effects are dropped from the model. This may result in compensating effects with no temporal delay. A loop in the temporal causal graph that does not have at least one integrating edge is called an *algebraic loop*.

In our qualitative analysis framework, algebraic loops produce conflicting predictions, therefore, future behavior expressed as higher order temporal effects become unknown. However, as discussed above, compensating effects have higher temporal order, but with very small time constants. With this knowledge, it makes sense not to mark the qualitative predictions as unknown, but to ensure that the compensating effects do not affect the initial magnitude deviations. To design an algorithm that implements this notion, vertices that are in an algebraic loop are tagged with a special value in the TCG. In Fig. 3, the vertices in the loop below are marked by a *.

$$e_6 \xrightarrow{1} e_3 \xrightarrow{\frac{1}{R_{hy-blk}}} f_3 \xrightarrow{=} f_6 \xrightarrow{-1} f_{15} \xrightarrow{R_{leak}} e_6 \tag{2}$$

As discussed earlier, this loop has a negative gain, but no temporal (dt) edges. Therefore, a straight propagation of a deviating value in this loop would cause all values along the loop to become unknown, an this, in turn, would affect predicted deviations in the rest of the model.

Since the causality assignment procedure in the bond graph can detect algebraic loops, the propagation algorithm can be modified to handle such situations. The modification is based on the observation based on physical principles, that the compensating effects in the algebraic loop affect the rate of change of the initial deviation, but not its magnitude. In our qualitative framework, an initial + deviation associated with a node in an algebraic loop remains +, because the compensating − effect generated in the loop only decreases the + magnitude. A similar statement can be made about an initial − deviation, with a + compensating effect generated in the algebraic loop. This holds only for compensating effects generated within the algebraic loop. When a vertex in a an algebraic loop is initially assigned a + value, and a − deviation is propagated along a path other than the algebraic loop (i.e, it is a non compensating effect), the predicted deviation for that vertex has to be listed as unknown.

The algorithm needs to keep track of the entry point of deviations into an algebraic loop. When deviations with opposite signs are assigned to a vertex, and they have the same entry point, the original deviation is retained, and the propagation along this path is terminated. If their entry points differ, the deviation value is marked as *unknown*, and the unknown value continues to be propagated forward. The pseudo code for the algorithm is presented in Algorithm 1.

To illustrate the algorithm, consider a hypothesized fault R^+_{l-hose} in the engine cooling system in Fig. 2. The TCG in Fig. 3 produces the following propagation trace:

$$
\begin{array}{l}
R^+_{l-hose} \rightarrow e^+_{14} \rightarrow e^-_{24} \rightarrow f^{\downarrow}_{24} \rightarrow \left\{ \begin{array}{l} f^{\downarrow}_{8} \rightarrow f^{\uparrow}_{7} \rightarrow e^{\uparrow\uparrow}_{7} \\ f^{\downarrow}_{9} \rightarrow f^{\downarrow,f_{15}}_{15} \rightarrow e^{\downarrow,f_{15}}_{15} \rightarrow \end{array} \right. \\
\left\{ \begin{array}{l} e^{\downarrow}_{9} \rightarrow e^{\uparrow}_{24} \rightarrow f^{\uparrow\uparrow}_{24} \\ \\ e^{\downarrow,f_{15}}_{6} \rightarrow e^{\downarrow,f_{15}}_{3} \rightarrow \left\{ \begin{array}{l} f^{\downarrow,f_{15}}_{3} \rightarrow \left\{ \begin{array}{l} f^{\downarrow}_{0} \rightarrow f^{\downarrow}_{1} \rightarrow e^{\downarrow\downarrow}_{1} \\ f^{\downarrow,f_{15}}_{6} \rightarrow f^{\uparrow,f_{15}}_{15} \Rightarrow \times \\ f^{\downarrow}_{10} \rightarrow e^{\downarrow}_{20} \rightarrow e^{\uparrow}_{19} \rightarrow f^{\uparrow\uparrow}_{19} \end{array} \right. \\ \\ s^{\downarrow}_{16} \rightarrow s^{\downarrow}_{17} \end{array} \right. \end{array} \right.
\end{array}
\tag{3}
$$

When the algebraic loop is entered at vertex f_{15}, this is stored and added as an attribute to each assigned deviation in the algebraic loop. When a vertex with a previously assigned deviation is reached, in this case f_{15} along the branching vertices f_9, e_6, f_3, and f_6, it is checked whether the current deviation that is propagated is caused by the same entry point deviation. For both the $\uparrow$ and $\downarrow$ prediction this entry point is f_{15}, and, therefore, no conflict is generated. Instead, the $\uparrow$ prediction is identified as a compensating effect, and the $\downarrow$ prediction is maintained.

Note that the entry point is stored for each order of the propagated deviations. This mechanism is illustrated with Fig. 4. When the fault in C is C^-, the predicted deviation in e_1 is e^+_1, which is propagated along e_2, e_3, f_3, f_4 to predicts the deviation $f^+_5(e_3)$, where e_3 indicates this was the algebraic loop entry point for the vertex deviation. Along e_8, e_7, f_7, f_6 the algorithm predicts $f^{\uparrow}_5(f_5)$. Now, when the f^+_5 deviation is propagated along e_5, e_4, e_3, f_3, f_2, f_1, e_1, e_2, e_3, f_3, f_4 a deviation $f^{\downarrow}_5(e_3)$ is predicted. This is a conflict because the entry points for f^+_5 is not the same as the entry point for $f^{\uparrow}_5$. The first order derivative of f_5, is, therefore, unknown.

Algorithm 1 Predict future behavior for a fault

```
add initial vertex, i.e., immediate consequence of the fault to list v_list
mark vertex 0^th order derivative with qualitative value
while v_list is not empty do
  v_current ← the first vertex in v_list
  while v_current has successors not determined to sufficient order do
    if successor relation includes a time integral effect then
      increase current derivative order
    end if
    if derivative order ≤ maximum order and successor derivative is not conflict then
      if successor derivative is no_mark then
        successor derivative value ← new_value(current value, relation)
        if successor in algebraic loop then
          if v_current in algebraic loop then
            add v_current loop entry point to set of successor loop entry points
          else
            add v_current to set of successor loop entry points
          end if
        end if
      else if successor derivative value is opposite of current value then
        if relation is inverse then
          if successor in algebraic loop then
            if current loop entry point not in set of successor loop entry points then
              if v_current in algebraic loop then
                add current loop entry point to set of successor loop entry points
              else
                add v_current to set of successor loop entry points
              end if
            end if
          else
            if successor in algebraic loop then
              if current loop entry point not in set of successor loop entry points then
                successor derivative value ← conflict
              end if
            else
              successor derivative value ← conflict
            end if
          end if
        else
          if relation is inverse then
            if successor in algebraic loop then
              if current loop entry not in set of successor loop entry points then
                successor derivative value ← conflict
              end if
            else
              successor derivative value ← conflict
            end if
          else
            if successor in algebraic loop then
              if current loop entry point not in set of successor loop entry points then
                if v_current in algebraic loop then
                  add current loop entry point to set of successor loop entry points
                else
                  add v_current to set of successor loop entry points
                end if
              end if
            end if
          end if
        end if
        if attributes of successor changed then
          add the successor to end of v_list
        end if
      end if
    end while
  end while
  for all vertex derivatives do
    if value = no_mark and any higher order derivative ≠ no_mark then
      replace no_mark with normal
    end if
    if value = conflict then
      replace conflict with no_mark
    end if
  end for
```

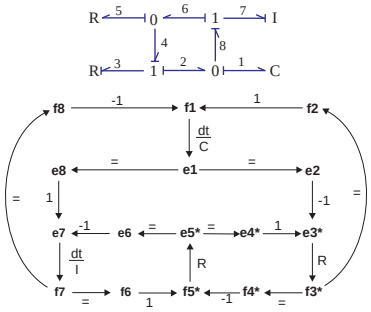

Figure 4: Algebraic loops require entry-point for each derivative.

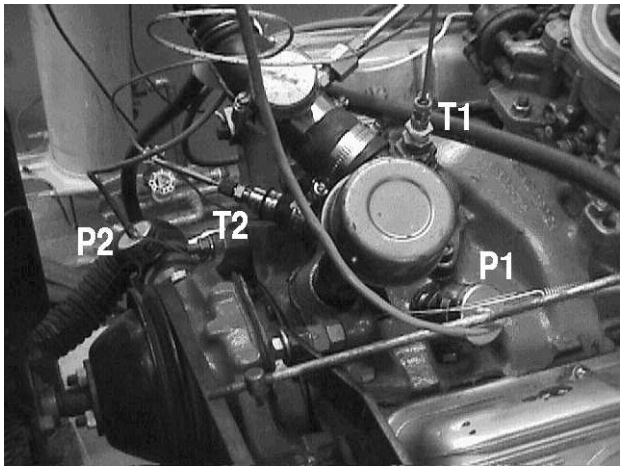

Figure 5: Chevrolet V8 engine with sensors. The fan is on the bottom left, the carburetor on the top right (the air cleaner has been removed).

4 Experimental Results

This section describes results of the cooling system. Details of the data acquisition system and signal processing techniques are presented elsewhere [3]. The measured variables are depicted in Fig. 5 where $\{T1, T2, P1, P2\}$ corresponds to the vertices $\{e47, e30, s17, e6\}$ in the TCG in Fig. 3, respectively.

4.1 Running Diagnosis Experiments

We describe a scenario where the engine system is run to a steady state operation (or close to that), after which a punctured hose fault is introduced. To simulate a punctured hose, we have inserted a T-split coupling in the lower radiator hose that allows us to drain coolant from the system by attaching a valve to the open end of the coupling. The coupling has a large inner diameter to enable a large outflow. A large hose puncture is simulated by a lever operated gate valve attached to the coupling. This valve can be switched from closed to open very quickly to make the fault look abrupt. We do not drain all the coolant from the engine during the experiment to prevent damage. The valve is closed after a large quantity of coolant is drained.

Since the nominal steady state values for the measured signals in our test bed are well

time	T1	T3	P1	P2
4	(0,.)	(0,.)	(0,.)	(0,.)
5	(0,.)	(0,.)	(0,.)	(0,.)
6	(0,.)	(0,.)	(-,.,*)	(-,.,*)
7	(+,+)	(+,+)	(-,-)	(-,+)
8	(+,+)	(+,+)	(-,-)	(-,0)

Table 1: Results of the signal to symbol transformation on the data around the occurrence of the fault.

known, we use these as the reference values for diagnostic analysis. Fault data captured by our data acquisition system is processed by a median filter to remove outliers, and the statistical discontinuity detection and slope estimation techniques are applied to process the faulty data, and convert them into a sequence of symbol values [3]. The processed signals are then analyzed by our diagnosis algorithms.

4.2 Results

Table 2 shows the results of the signal to symbol transformation from step 4 to step 8 (the fault occurs at step 5). The slope values are marked as *unknown* (indicated by a '.') till an initial deviation has occurred. The tuple is extended with a '$\star$' to indicate that an abrupt change was detected during a step. Fig. 6 shows the results generated by the hypothesis generation and refinement procedures. The nonlinearities and the algebraic loop in the engine model resulted in unknown signatures for second and higher order derivatives for all of the predicted faults, except for R^+_{tstat} and R^-_{tstat}. Therefore, predictions were made up till the second order derivative for this model. The steps are indexed starting at 0, which is when the initial discrepancy is detected (in both P1 and P2). This is actually time step 6 in Table 4.2.

To illustrate, consider the prediction for R^+_{l-hose}. According to the trace in Eq. (3) the observed deviations are $e^{\downarrow}_6$ and $s^{\downarrow}_{16}$. This matches the predicted $-$ deviation in Fig. 6 of the first order behavior.

The resulting candidates generated by the hypothesis generation algorithm and their corresponding fault signatures generated by the prediction algorithm are listed in the first column of Fig. 6. For some parameters, both an increase and decrease in fault magnitudes are hypothesized. This is because each deviating measurement is used to generate possible faults. So, R^+_{tstat} is generated in response to s^-_{17} and R^-_{stat} in response to e^-_6.

4.2.1 Overall Performance

Detection of abrupt changes results in the elimination of a large number of hypothesized faults at step 1. At step 2 the R^-_{hy-blk} fault is eliminated because the predicted *slope* is negative for measurement e_{30}, which does not match the observed magnitude deviation. Without abrupt change detection, R^+_{l-hose} and C^+_{hy-rad} cannot be eliminated at step 1 nor at any later time step and will stay in the set of hypothesized faults.

The final result at step 2 in Fig. 6 shows that the diagnosis is accurate because it includes the actual fault R^-_{leak}. Only one spurious candidate, $I^+_{rad-out}$, is generated and cannot be distinguished from R^-_{leak} given this set of observations. In other work [8, 10], we have shown

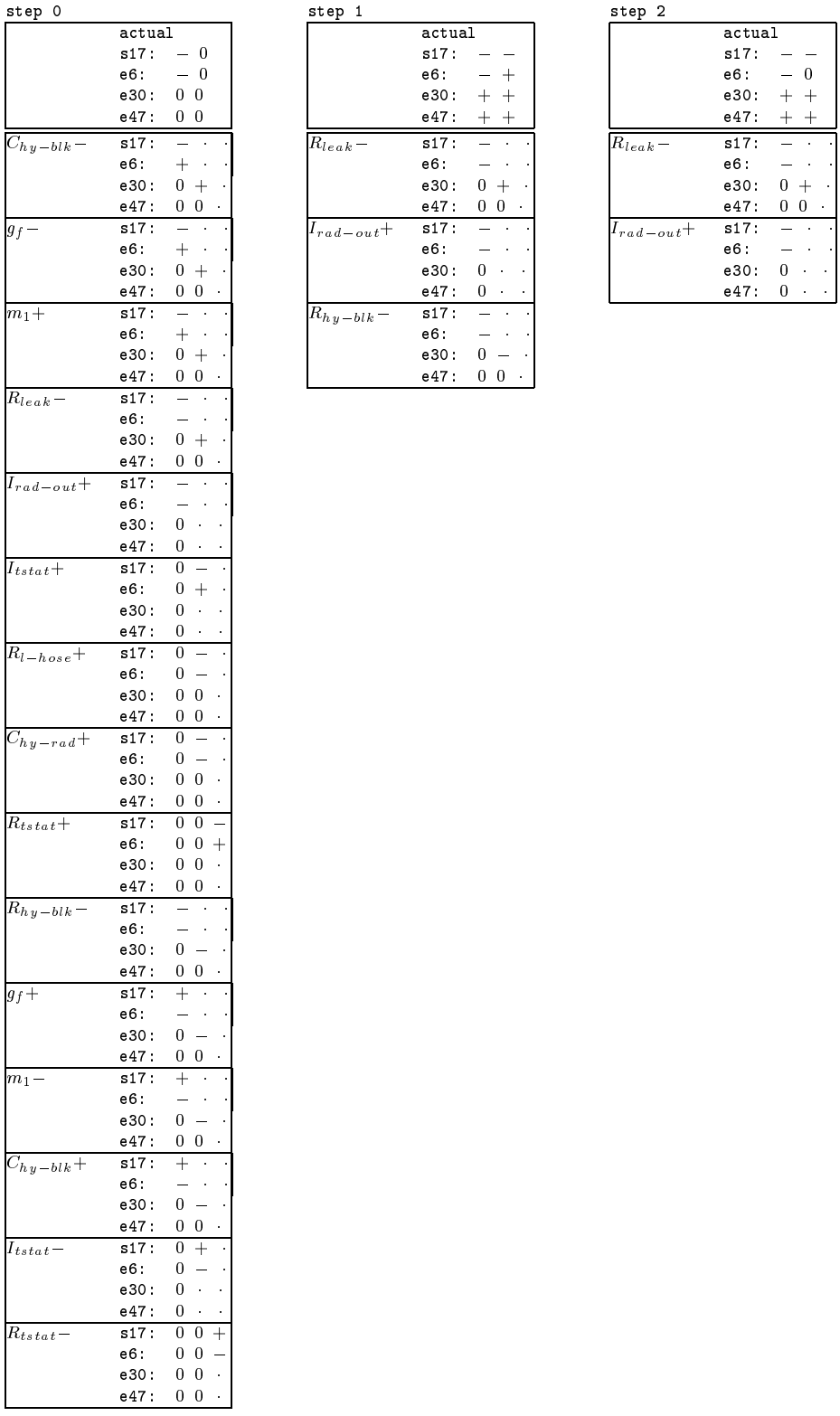

Figure 6: Fault detection and isolation for a punctured lower hose fault ($R_{leak}-$).

how a systematic measurement selection algorithm can be employed to establish diagnosability of the system, i.e., which fault sets can be discriminated. Furthermore, this measurement selection algorithm can be used to generate the minimal set of required measurements for complete diagnosability, i.e., all faults can be uniquely discriminated. This analysis has not yet been performed on this system.

4.2.2 The Prediction Algorithm

The prediction of R^+_{tstat} shows a first deviation in its second order derivatives. This is generated from the TCG in Fig. 3 by tracing the $e^{\downarrow\downarrow}_{12}$ in Eq. (1) further.

$$e^{\downarrow\downarrow}_{12} \rightarrow \left\{ \begin{array}{l} s^{\downarrow\downarrow}_{5} \rightarrow s^{\downarrow\downarrow}_{17} \\ e^{\downarrow\downarrow}_{0} \rightarrow e^{\downarrow\downarrow,e_3}_{3} \rightarrow f^{\downarrow\downarrow,e_3}_{3} \rightarrow \end{array} \right. \left\{ \begin{array}{l} f^{\downarrow\downarrow}_{0} \rightarrow f^{\downarrow\downarrow}_{1} \rightarrow e^{\downarrow\downarrow\downarrow}_{1} \\ f^{\downarrow\downarrow,e_3}_{6} \rightarrow f^{\uparrow\uparrow,e_3}_{15} \rightarrow e^{\uparrow\uparrow,e_3}_{15} \rightarrow \left\{ \begin{array}{l} e^{\uparrow\uparrow,e_3}_{6} \rightarrow e^{\uparrow\uparrow,e_3}_{3} \Rightarrow \times \\ e^{\uparrow\uparrow}_{9} \ldots \end{array} \right. \\ f^{\downarrow\downarrow}_{10} \rightarrow e^{\downarrow\downarrow}_{20} \rightarrow e^{\uparrow\uparrow}_{19} \rightarrow f^{\uparrow\uparrow\uparrow}_{19} \end{array} \right. \tag{4}$$

This shows how the second order derivative predictions of s_{17} and e_3 are derived. Note the feedback along the algebraic loop that is terminated when a second order prediction with opposing sign for e_3 is generated that has the same entry point in the algebraic loop.

An example of a conflict in the algebraic loop is given by $I^+_{rad-out}$, which results in the following trace (partially expanded)

$$I^+_{rad-out} \rightarrow f^-_{24} \rightarrow \left\{ \begin{array}{l} f^-_8 \ldots \\ f^-_{14} \rightarrow e^-_{14} \rightarrow e^+_{24} \rightarrow f^{\uparrow}_{24} \left\{ \begin{array}{l} f^{\uparrow}_{8} \ldots \\ f^{\uparrow}_{14} \ldots \\ f^{\uparrow}_{9} \rightarrow f^{\uparrow,f_{15}}_{15} \rightarrow e^{\uparrow,f_{15}}_{15} \rightarrow \left\{ \begin{array}{l} e^{\uparrow}_{9} \ldots \\ e^{\uparrow,f_{15}}_{6} \rightarrow e^{\uparrow,f_{15}}_{3} \Rightarrow \times \end{array} \right. \end{array} \right. \\ f^-_9 \rightarrow f^{-,f_{15}}_{15} \rightarrow e^{-,f_{15}}_{15} \left\{ \begin{array}{l} e^-_9 \ldots \\ e^{-,f_{15}}_{6} \rightarrow e^{-,f_{15}}_{3} \rightarrow f^{-,f_{15}}_{3} \rightarrow \left\{ \begin{array}{l} f^{-,f_{15}}_{6} \ldots \\ f^-_{10} \ldots \\ f^-_0 \rightarrow f^-_1 \rightarrow e^{\downarrow}_1 \rightarrow e^{\downarrow}_0 \rightarrow e^{\downarrow,e_3}_3 \Rightarrow \times \end{array} \right. \end{array} \right. \end{array} \right. \tag{5}$$

This trace shows that de $I_{rad-out}$ deviation propagates along two different paths to predict the first order derivative of e_3. The top trace generates a first order effect by propagating through R_{l-hose} and back into $I_{rad-out}$ to generate a first order effect that enters the algebraic loop at f_{15}. This predicts $e^{\uparrow,f_{15}}_3$. The bottom trace propagates the f^-_{24} effect first through the algebraic loop and then through C_{hy-blk} to generate a first order effect that enters the algebraic loop at e_3 and predicts $e^{\downarrow,e_3}_3$. This prediction conflicts with the prediction along the other path, and, because the entry points are different, the first order prediction of e_3 is unknown. Note that this unknown value now is propagated further, causing a large number of first and higher order effects to become unknown (see Fig. 6).

4.2.3 Evaluation

For comparison we tested the prediction algorithm without the algebraic loop facility. The final diagnosis result for the punctured hose in Fig. 7 shows that this algorithms has much less discriminative power, and, therefore, more spurious candidates are generated.

$C_{hy-blk}-$	s17: · · ·	$I_{tstat}+$	s17: 0 · ·	g_f+	s17: · · ·
	e6: · · ·		e6: 0 · ·		e6: · · ·
	e30: 0 · ·		e30: 0 · ·		e30: 0 · ·
	e47: 0 0 ·		e47: 0 · ·		e47: 0 0 ·
g_f-	s17: · · ·	$R_{l-hose}+$	s17: 0 · ·	m_1-	s17: · · ·
	e6: · · ·		e6: 0 · ·		e6: · · ·
	e30: 0 · ·		e30: 0 0 ·		e30: 0 · ·
	e47: 0 0 ·		e47: 0 0 ·		e47: 0 0 ·
m_1+	s17: · · ·	$C_{hy-rad}+$	s17: 0 · ·	$C_{hy-blk}+$	s17: · · ·
	e6: · · ·		e6: 0 · ·		e6: · · ·
	e30: 0 · ·		e30: 0 0 ·		e30: 0 · ·
	e47: 0 0 ·		e47: 0 0 ·		e47: 0 0 ·
$R_{leak}-$	s17: · · ·	$R_{tstat}+$	s17: 0 0 ·	$I_{tstat}-$	s17: 0 · ·
	e6: · · ·		e6: 0 0 ·		e6: 0 · ·
	e30: 0 · ·		e30: 0 0 ·		e30: 0 · ·
	e47: 0 0 ·		e47: 0 0 ·		e47: 0 · ·
$I_{rad-out}+$	s17: · · ·	$R_{hy-blk}-$	s17: · · ·	$R_{tstat}-$	s17: 0 0 ·
	e6: · · ·		e6: · · ·		e6: 0 0 ·
	e30: 0 · ·		e30: 0 · ·		e30: 0 0 ·
	e47: 0 · ·		e47: 0 0 ·		e47: 0 0 ·

Figure 7: Signature generation without use of the algebraic loop extension to the algorithm

To test the implementation, we performed two important tests. First of all, the algorithm should generate the same results for breadth first and depth first propagation. Secondly, because of the arbitrary choice of causality assignment in the algebraic loop, the prediction should produce the same results for different assignments. Both of these test were successfully conducted.

5 Conclusions

TRANSCEND is a monitoring and diagnosis system that relies on a process model to generate qualitative prediction. This process model takes the form of a temporal causal graph. In certain cases, the underlying system equations cannot be formulated in an explicit form because of circular dependencies between variables. In the corresponding bond graph model, these circular dependencies may arise because of algebraic loops. The inherent negative feedback gain of such loops causes a large number of predictions to be unknown, therefore, it is desirable to either generate an explicit form of the algebraic loop variables, or to augment the prediction algorithm to handle such loops. The first option relies on introducing

References

[1] M. Andersson. *Object-Oriented Modeling and Simulation of Hybrid Systems*. PhD dissertation, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden, 1994.

[2] B. Kuipers. *Qualitative Reasoning: Modeling and Simulation with Incomplete Knowledge*. MIT Press, Cambridge, MA, 1994.

[3] Eric J. Manders, Pieter J. Mosterman, and Gautam Biswas. TRANSCEND: A system for robust monitoring and diagnosis of complex engineering systems. *International Journal of Adaptive Control and Signal Processing*, 2000. in review.

[4] Pieter J. Mosterman. *Hybrid Dynamic Systems: A hybrid bond graph modeling paradigm and its application in diagnosis*. PhD dissertation, Vanderbilt University, 1997.

[5] Pieter J. Mosterman and Gautam Biswas. Monitoring, Prediction, and Fault Isolation in Dynamic Physical Systems. In *AAAI-97*, pages 100–105, Rhode Island, August 1997. AAAI Press, 445 Burgess Drive, Menlo Park, CA, 94025.

[6] Pieter J. Mosterman and Gautam Biswas. Diagnosis of continuous valued systems in transient operating regions. *IEEE Transactions on Systems, Man, and Cybernetics*, 1998. in review.

[7] Pieter J. Mosterman and Gautam Biswas. A Java Implementation of an Environment for Hybrid Modeling and Simulation of Physical Systems. In *ICBGM99*, pages 157–162, San Francisco, January 1999.

[8] Pieter J. Mosterman, Gautam Biswas, and Narasimhan Sriram. Measurement Selection and Diagnosability of Complex Dynamic Systems. In *Eighth International Conference on Principles of Diagnosis*, pages 79–86, Mont St. Michel, France, September 1997.

[9] Pieter J. Mosterman, Gautam Biswas, and Janos Sztipanovits. Hybrid modeling and verification of embedded control systems. In *Proceedings of the 7^{th} IFAC CACSD '97 Symposium*, pages 21–26, Gent, Belgium, April 1997.

[10] Sriram Narasimhan, Pieter J. Mosterman, and Gautam Biswas. A Systematic Analysis of Measurement Selection Algorithms for Fault Isolation in Dynamic Systems. In *Ninth International Workshop on Principles of Diagnosis*, pages 94–101, Cape Cod, MA, May 1998.

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

[12] Johannes van Dijk. *On the role of bond graph causality in modelling mechatronic systems*. PhD dissertation, University of Twente, CIP-Gegevens Koninklijke Bibliotheek, Den Haag, The Netherlands, 1994.

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