Fault Aware Systems: Model-based Programming and Diagnosis

Outline

- Fault Aware Systems Through Model-based Programming
- Diagnosis as Detective Work
- Model-based Diagnosis

Fault Aware Systems:
Create embedded languages
That reason and coordinate
on the fly from models

Programmers are overwhelmed
by the bookkeeping of reasoning
about unlikely hidden states

Mars Polar Lander Failure

Leading Diagnosis:
- Legs deployed during descent.
- Noise spike on leg sensors
  latched by software monitors.
- Laser altimeter registers 50ft.
- Begins polling leg monitors to
determine touch down.
- Latched noise spike read as
touchdown.
- Engine shutdown at ~50ft.

Like Storyboards, Model-based Programs
Specify The Evolution of Abstract States

Embedded programs evolve actions
by interacting with plant sensors
and actuators:
- Read sensors
- Set actuators

Model-based programs evolve
abstract states through direct
interaction:
- Read abstract state
- Write abstract state

Programmer maps between state
and sensors/actuators.

Model-based executive maps
between state and sensors/actuators.

Descent Example

Turn camera off and engine on

EngineA EngineB
Science Camera

EngineA EngineB
Science Camera

Titan Model-based Executive

Generates target goal states
conditioned on state estimates

Tracks likely plant states
Tracks least cost goal states

System Model

Observations

Commands

Plant

 Vương Nguyễn
Model-based Programs

Control program specifies state trajectories:
- fires one of two engines
- sets both engines to 'standby'
- prior to firing engine, camera must be turned off to avoid plume contamination
- in case of primary engine failure, fire backup engine instead

Plant Model describes behavior of each component:
- Nominal and Off nominal
- qualitative constraints
- likelihoods and costs

Modeling Complex Behaviors through Probabilistic Constraint Automata

- Complex, discrete behaviors
  - modeled through concurrency, hierarchy and timed transitions.
- Anomalies and uncertainty
  - modeled by probabilistic transitions
- Physical interactions
  - modeled by discrete and continuous constraints

The Plant’s Behavior

- Assigns a value to each variable (e.g., 3,000 vars).
- Consistent with all state constraints (e.g., 12,000).
- A set of concurrent transitions, one per automata (e.g., 80).
- Previous & Next states consistent with source & target of transitions
arg max \( P_r(m') \)
s.t. \( M(m') \land O(m') \) is satisfiable

arg min \( R_r(m') \)
s.t. \( M(m') \) entails \( G(m') \)
s.t. \( M(m') \) is satisfiable

\[ \text{arg max } P_r(m') \]
s.t. \( M(m') \) \land \( O(m') \) is satisfiable

\[ \text{arg min } R_r(m') \]
s.t. \( M(m') \) entails \( G(m') \)
s.t. \( M(m') \) is satisfiable

Modeling Complex Behaviors through Probabilistic Constraint Automata

- Complex, discrete behaviors
  - modeled through concurrency, hierarchy and timed transitions.
- Anomalies and uncertainty
  - modeled by probabilistic transitions
- Physical interactions
  - modeled by discrete and continuous constraints

The Plant’s Behavior

- Assigns a value to each variable (e.g., 3,000 vars).
- Consistent with all state constraints (e.g., 12,000).
- A set of concurrent transitions, one per automata (e.g., 80).
- Previous & Next states consistent with source & target of transitions

Outline

- Fault Aware Systems Through Model-based Programming
- Diagnosis as Detective Work
- Model-based Diagnosis
Issue 1: Handling Hidden Failures Requires Reasoning from a Model: STS-93

Symptoms:
• Engine temp sensor high
• LOX level low
• GN&C detects low thrust
• H2 level possibly low

Problem: Liquid hydrogen leak

Effect:
• LH2 used to cool engine
• Engine runs hot
• Consumes more LOX

Compare Most Likely Hypothesis to Observations

Main Fuel tank
Engines
Helium tank
Oxidizer tank

Flow = zero
Pressure = nominal
Acceleration = zero

It is most likely that all components are okay.

Isolate Conflicting Information

Main Fuel tank
Engines
Helium tank
Oxidizer tank

Flow = zero
Pressure = nominal

The red component modes conflict with the model and observations.

Leap to the Next Most Likely Hypothesis that Resolves the Conflict

New Hypothesis Exposes Additional Conflicts

Model- ased Diagnosis as Conflict-directed Best First Search

When you have eliminated the impossible, whatever remains, however improbable, must be the truth.

- Sherlock Holmes. The Sign of the Four.

1. Test Hypothesis
2. If Inconsistent, learn reason for inconsistency (a Conflict).
3. Use conflicts to leap over similarly infeasible options to next best hypothesis.
Final Hypothesis Resolves all Conflicts

Outline
- Fault Aware Systems Through Model-based Programming
- Diagnosis as Detective Work
- Model-based Diagnosis

Model-based Diagnosis

Given a system with symptomatic behavior and a model of the system, find diagnoses that account for symptoms.

1. Generate candidates, given symptoms.
2. Test if candidates account for all symptoms.

Desired Properties:
- Set of diagnoses should be complete.
- Set of diagnoses should consider all available information.

Model-based Diagnosis

Given a system with symptomatic behavior and a model of the system, find diagnoses that account for symptoms.

Diagnosis as Hypothesis Testing

Issue 2: Failures are Often Novel:

Mars Observer: Explosion due to oxidizer/fuel leakage?
**Issue 2: How Should Diagnoses Account for Novel Failures?**

**Consistency-based Diagnosis:** Given symptoms, find diagnoses that are consistent with symptoms.

**Suspending Constraints:** Make no presumptions about faulty component behavior.

![Diagram](image1)

**Symptom**

1. Or1
2. Or2
3. Or3

**Issue 3: Multiple Faults Occur**

- Three shorts, tank-line and pressure jacket burst, panel flies off.
- **Divide & Conquer**
  - Diagnose each symptom.
  - Summarize (conflicts)
  - Combine

![Apollo 13 Image](image2)

**Diagnosis identifies consistent modes**

Adder(i):
- \( \text{G}(i) = \text{Out}(i) = \text{In}_1(i) + \text{In}_2(i) \)
- \( \text{U}(i) \)

Candidate: Assignment to all component modes.

![Diagram](image3)

**Diagnosis identifies All sets of consistent modes**

Adder(i):
- \( \text{G}(i) = \text{Out}(i) = \text{In}_1(i) + \text{In}_2(i) \)
- \( \text{U}(i) \)

Diagnosis = \{A1=G, A2=U, M1=G, M2=U, M3=G\}

- Diagnosis D: Candidate consistent with model Phi and observables OBS
- As more constraints are relaxed, candidates are more easily satisfied.
- Typically an exponential number of candidates.
Representing Diagnoses Compactly: Kernel Diagnoses

- Smallest sets of modes that remove all symptoms
- Every candidate that is a subset of a kernel diagnosis is a diagnosis.

Encoding Models In Propositional Logic

And(i):
- G(i): Out(i) = In1(i) AND In2(i)
- U(i): (¬(¬G(i)) v ¬(¬In1(i)) v ¬Out(i)) v (¬(¬G(i)) v ¬(¬In2(i)) v ¬Out(i))

Or(i):
- G(i): Out(i) = In1(i) OR In2(i)
- U(i): (¬(¬G(i)) v (¬In1(i)) v ¬Out(i)) v (¬(¬G(i)) v (¬In2(i)) v ¬Out(i))

Given model Phi and observations OBS
- Conflicts and Kernel Diagnoses
- Generating Kernels from Conflicts
- Finding Consistent Modes
- Estimating Likely Modes
- Conflict-directed A*

Outline

Testing Consistency

- Propostional Logic
  - DPLL Sat algorithm
  - Unit propagation (incomplete)

- Finite Domain Constraints
  - Backtrack Search w Forward Checking
  - AC-3/Waltz constraint propagation (incomplete)

- Algebraic Constraints
  - Sussman/Steele Constraint Propagation:
    - Propagate newly assigned values through equations mentioning variables.
    - To propagate, use assigned values of constraint to deduce unknown value(s) of constraint.

Summary: Consistency-based Diagnosis

And(i):
- G(i): Out(i) = In1(i) AND In2(i)
- U(i):

Diagnosis = \{A1=G, A2=U, O1=G, O2=U, O3=G\}

Summary: Consistency-based Diagnosis

- Component Model + Structure:
- And(i):
- Or(i):

Consistency-based Diagnosis

General Diagnostic Engine
[de Kleer & Williams, 87]
Conflicts Explain How to Remove Symptoms

Symptom: F is observed 10, but should be 12 if A1, M1 & M2 are okay.

Conflict: A1=G & M1=G & M2=G is inconsistent

A1=U or M1=U or M2=U removes conflict.

i.e., at least one is broken

Find Another Symptom

Symptom: G is observed 12, but should be 10 ...

Conflict: A1=G & M2=G & M1=G & M3=G is inconsistent

A1=U or A2=U or M1=U or M3=U removes conflict.

Summary: Conflicts

Conflict:

A set of component modes M that are inconsistent with the model and observations.

Properties:

• Every superset of a conflict is a conflict
• Only need conflicts that are minimal under subset
• Logically, not M is an implicate of Model & Obs
Summary: Kernel Diagnoses

Kernel Diagnosis
= \{ A_2=U \& M_2=U \}

Partial Diagnosis: A set of component modes M all of whose extensions are diagnoses.
- M removes all symptoms
- M entails Model & Obs (implicant)

Kernel Diagnosis: A minimal partial diagnosis K
- M is a prime implicant of model & obs

Outline

Model-based Diagnosis
- Conflicts and Kernel Diagnoses
- Generating Kernels from Conflicts
- Finding Consistent Modes
- Estimating Likely Modes
- Conflict-directed A*

Diagnoses Found by Mapping Conflicts to Kernels

Conflict: A set of component modes M that are inconsistent with the model and observations.
- not M is an implicate of Model & Obs

Kernel Diagnosis: A minimal set of component modes K that eliminate all symptoms.
- M is a prime implicant of Model & Obs

Conflicts map to Kernels by minimal set covering
(see “Characterizing Diagnosis,” de Kleer, Reiter, Mackworth)

Generate Kernels From Conflicts

\{ A_1=U, A_2=U, M_1=U, M_3=U \} conflict 1.
\{ A_1=U, A_2=U, M_1=U, M_3=U \} conflict 2
A_1=U or A_2=U or M_1=U or M_3=U removes conflict 1.
A_1=U or A_2=U or M_1=U or M_3=U removes conflict 2

Kernel Diagnoses = \{ A_1=U \}

“Smallest” sets of modes that remove all conflicts

Generate Kernels From Conflicts

\{ A_1=U, M_1=U, M_2=U \} conflict 1.
\{ A_1=U, A_2=U, M_1=U, M_3=U \} conflict 2
A_1=U or M_1=U or M_2=U removes conflict 1.
A_1=U or A_2=U or M_1=U or M_3=U removes conflict 2

Kernel Diagnoses = \{ M_1=U \}

“Smallest” sets of modes that remove all conflicts

Generate Kernels From Conflicts

\{ A_1=U, M_1=U, M_2=U \} conflict 1.
\{ A_1=U, A_2=U, M_1=U, M_3=U \} conflict 2
A_1=U or M_1=U or M_2=U removes conflict 1.
A_1=U or A_2=U or M_1=U or M_3=U removes conflict 2

Kernel Diagnoses = \{ A_1=U \}

“Smallest” sets of modes that remove all conflicts
**Generate Kernels From Conflicts**


A1=U or M1=U or M2=U removes conflict 1.
A1=U or A2=U or M1=U or M3=U removes conflict 2.

Kernel Diagnoses = \{A2=U, M2=U\}
\{M1=U\}
\{A1=U\}

"Smallest" sets of modes that remove all conflicts.

**Single Fault Diagnoses are the Intersection of All Conflicts**


A1=U or M1=U or M2=U removes conflict 1.
A1=U or A2=U or M1=U or M3=U removes conflict 2.

Single Fault Diagnoses = \{A1=U, M1=U\}

**Outline**

- Model-based Diagnosis
  - Conflicts and Kernel Diagnoses
  - Generating Kernels from Conflicts
  - Finding Consistent Modes
  - Estimating Likely Modes
  - Conflict-directed A*

**Diagnosis With Only the Unknown**

- Nominal and Unknown Modes

  - Inverter(i):
    - \(G(i)\): \(\text{Out}(i) = \text{not}(\text{In}(i))\)
    - \(U(i)\):
      - Isolates surprises
      - Doesn’t explain

Notational Note: \(G(i) = I = G\)

**Diagnosis With Only the Known**

- Exhaustive Fault Modes

  - Inverter(i):
    - \(G(i)\): \(\text{Out}(i) = \text{not}(\text{In}(i))\)
    - \(S1(i)\): \(\text{Out}(i) = 1\)
    - \(S0(i)\): \(\text{Out}(i) = 0\)

  - No surprises
  - Explains
Solution: Diagnosis as Estimating Behavior Modes

- Inverter(i):
  - $G(i)$: Out(i) = not(In(i))
  - $S1(i)$: Out(i) = 1
  - $S0(i)$: Out(i) = 0
  - $U(i)$: Isolates surprises
  - $U(i)$: Explains

Nominal, Fault and Unknown Modes

Example Diagnoses

- Diagnosis: [S1(A), G(B), U(C)]

1. Find Symptoms & Conflicts
- Conflict: not [G(A), G(B) and G(C)]

More Symptoms & Conflicts
- Not [S1(A), G(B), and G(C)]

More Symptoms & Conflicts
- not [S0(B) and G(C)]
More Symptoms & Conflicts

2. Constituent Diagnoses from Conflicts

- \(< S1(C) >
- \(\Rightarrow G(C), S0(C) \text{ or } U(C)\)
- \(< S0(B), G(C) >
- \(\Rightarrow G(B), S1(B), U(B), S1(C), S0(C) \text{ or } U(C)\)
- \(< S1(A), G(B), G(C) >
- \(\Rightarrow G(A), S0(A), U(A), S1(B), U(B), S1(C), S0(C) \text{ or } U(C)\)
- \(< G(A), G(B), G(C) >
- \(\Rightarrow S1(A), S0(A), U(A), S1(B), U(B), S1(C), S0(C) \text{ or } U(C)\)

3. Generating Kernel Diagnoses

- \([G(C), S0(C), U(C)]\)
- \([G(B), S1(B), U(B), S1(C), S0(C), U(C)]\)
- \([G(A), S0(A), U(A), S1(B), U(B), S1(C), S0(C), U(C)]\)
- \([S1(A), S0(A), U(A), S1(B), U(B), S1(C), S0(C), U(C)]\)

- \([U(C)]\)
- \([S0(C)]\)

3. Generating Kernel Diagnoses

- \([G(C), S0(C), U(C)]\)
- \([G(B), S1(B), U(B), S1(C), S0(C), U(C)]\)
- \([G(A), S0(A), U(A), S1(B), U(B), S1(C), S0(C), U(C)]\)
- \([S1(A), S0(A), U(A), S1(B), S0(B), U(B), S1(C), S0(C), U(C)]\)

- \([U(C)]\)
- \([S0(C)]\)
- \([U(B), G(C)]\)
3. Generating Kernel Diagnoses

- \([G(C),S0(C),U(C)]\)
- \([G(B),S1(B),U(B),S1(C),S0(C),U(C)]\)
- \([G(A),S0(A),U(A),S1(B),S0(B),U(B),S1(C),S0(C),U(C)]\)
- \([S1(A),S0(A),U(A),S1(B),S0(B),U(B),S1(C),S0(C),U(C)]\)

\([U(C)]\)
\([S0(C)]\)
\([U(B),G(C)]\)

\([S1(B),G(C)]\)

3. Generate Kernel Diagnoses

- \([G(C),S0(C),U(C)]\)
- \([G(B),S1(B),U(B),S1(C),S0(C),U(C)]\)
- \([G(A),S0(A),U(A),S1(B),S0(B),U(B),S1(C),S0(C),U(C)]\)
- \([S1(A),S0(A),U(A),S1(B),S0(B),U(B),S1(C),S0(C),U(C)]\)

\([U(C)]\)
\([S0(C)]\)
\([U(B),G(C)]\)

\([S1(B),G(C)]\)
\([U(A),G(B),G(C)]\)

3. Generating Kernel Diagnoses

- \([G(C),S0(C),U(C)]\)
- \([G(B),S1(B),U(B),S1(C),S0(C),U(C)]\)
- \([G(A),S0(A),U(A),S1(B),S0(B),U(B),S1(C),S0(C),U(C)]\)
- \([S1(A),S0(A),U(A),S1(B),S0(B),U(B),S1(C),S0(C),U(C)]\)

\([U(C)]\)
\([S0(C)]\)
\([U(B),G(C)]\)

\([S1(B),G(C)]\)
\([U(A),G(B),G(C)]\)

Outline

Model-based Diagnosis
- Conflicts and Kernel Diagnoses
- Generating Kernels from Conflicts
- Finding Consistent Modes
- Estimating Likely Modes
- Conflict-directed A*

Due to the unknown mode, there tends to be an exponential number of diagnoses.

But these diagnoses represent a small fraction of the probability density space.

Most of the density space may be represented by enumerating the few most likely diagnoses.
Candidate Initial (prior) Probabilities

\[ p(c) = \prod_{m} p(m) \]

Assume Failure Independence

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>p(G)</td>
<td>.99</td>
<td>.99</td>
<td>.99</td>
</tr>
<tr>
<td>p(S1)</td>
<td>.008</td>
<td>.008</td>
<td>.001</td>
</tr>
<tr>
<td>p(S0)</td>
<td>.001</td>
<td>.001</td>
<td>.008</td>
</tr>
<tr>
<td>p(U)</td>
<td>.001</td>
<td>.001</td>
<td>.001</td>
</tr>
</tbody>
</table>

\[ p(G(A),G(B),G(C)) = .97 \]
\[ p(S1(A),G(B),G(C)) = .008 \]
\[ p(S0(A),G(B),S0(C)) = .00006 \]
\[ p(S0(A),S1(B),S0(C)) = .0000005 \]

Posterior Probability, after Observation \( x = v \)

\[ p(c \mid x = v) = \frac{p(x = v \mid c)p(c)}{p(x = v)} \]

Bayes’ Rule

\[ P(x=v|c) \] estimated using Model:
- If previous obs, \( c \) and \( \Phi \) entails \( x = v \)
  Then \( p(x = v \mid c) = 1 \)
- If previous obs, \( c \) and \( \Phi \) entails \( x \neq v \)
  Then \( p(x = v \mid c) = 0 \)
- If \( \Phi \) consistent with all values for \( x \)
  Then \( p(x = v \mid c) \) is based on priors
  E.g., uniform prior \( = 1/m \) for \( m \) possible values of \( x \)

Observe out = 1:
- \( C = [G(A),G(B),G(C)] \)
- Prior: \( P(C) = .97 \)
- \( P(out = 1 \mid C) =? \)
  \( = 1 \)
- \( P(C \mid out = 0) =? \)
  \( = .97/p(x=v) \)

Observe out = 0:
- \( C = [G(A),G(B),G(C)] \)
- Prior: \( P(C) = .97 \)
- \( P(out = 0 \mid C) =? \)
  \( = 0 \)
- \( P(C \mid out = 0) =? \)
  \( = 0 \times .97/p(x=v) = 0 \)

Prior for Single Fault Diagnoses:
- \( p(S1) \)
- \( p(S0) \)
- \( p(U) \)
**Summary: Candidate Probabilities**

\[ p(c) = \prod_{m \in c} p(m) \]

Assume Failure Independence

\[ p(c \mid x = v) = \frac{p(x = v \mid c)p(c)}{p(x = v)} \]

Bayes’ Rule

\[ P(x=v|c) \text{ estimated using Model: } \]

- If previous obs, c and Phi entails \( x = v \)
  - Then \( p(x = v \mid c) = 1 \)
- If previous obs, c and Phi entails \( x \neq v \)
  - Then \( p(x = v \mid c) = 0 \)
- If Phi consistent with all values for \( x \)
  - Then \( p(x = v \mid c) \) is based on priors
  - E.g., uniform prior = \( \frac{1}{m} \) for \( m \) possible values of \( x \)

---

Due to the unknown mode, there tends to be an exponential number of diagnoses.

But these diagnoses represent a small fraction of the probability density space.

\[ \Rightarrow \text{ Most of the density space may be represented by enumerating the few most likely diagnoses} \]