Automotive Fault Diagnosis: Distributed Diagnostic Agent System

Professor Yi Lu Murphey
Intelligent Systems and Machine Vision Lab
Dept. Of Electrical & Computer Engineering
The University of Michigan-Dearborn
313-593-5028
yilu@umich.edu
Organization of This Workshop

- Introduction to Vehicle Diagnostic Problem
- DDAS: Distributed Diagnostic Agents System
  - Definition of agents
  - DDAS
    - Diagnostic agents
    - Major functions involved in agents
    - Applications of DDAS
  - Vehicle signal analysis
- Advanced techniques used in DDAS
  - Signal segmentation
  - Signal feature extraction
  - Fuzzy logic and its application in signal segment fault detection and signal detection
  - Case Based Reasoning used in Vehicle Fault Diagnosis
- Applications of DDAS in OBD and vehicle prognosis
Introduction to Vehicle Diagnostics

- As electronic control systems in vehicles become more advanced and sophisticated in the recent years, malfunctions have been increasingly more complicated.
- A typical modern vehicle has a large number of sensors, controllers and computer modules embedded in the vehicle that collect abundant signals.
- Vehicle fault diagnosis very much depends on vehicle signal diagnosis.
- Vehicle signals range from simple binary modes to extremely complex spark timing signals that interact with each other either directly or indirectly, and have dynamic ranges in magnitude, oscillation, frequency, slope, derivative, etc
- The most challenging problems in vehicle diagnosis include
  - Signal dependency
  - Signal noise
  - Intermittent problems
Signal Dependency Examples

RPM dropping when TP rises

TP != TPCT at idle
Engineering Knowledge on Signal Interaction

Key

- Closely Follows
- Significant Effect
- Small/Unknown Effect
- Key Signal
- Useful Signal

Control Signals (from PCM)

Sensor Input Signals

Diagram showing interactions between various signals such as LOAD, O2S xx, FUELPW xx, IAC, SPARKADV, RPM, MAF, TP, TPCT, and VSS, with arrows indicating the flow of information.
Vehicle Classes

- **TP Class**: TP, RPM, MAF, IAC, FUELPW
- **ECT Class**: ECT, IAT
- **SPARK Class**: SPARKADV, FUELPW, LOAD, MAF
- **MODE Class**: TR (Transmission Mode), GEAR (Current Gear)
Research Objectives

- Investigate advance technology for signal based fault diagnosis
  - Develop intelligent agents for
    - Single signal fault analysis
    - Multiple signals fault analysis
    - Vehicle fault diagnosis
  - Develop machine learning technology to automatically train agents
General Definition of Agents

- An agent is a computer system situated in some environment, and that is capable of autonomous action in this environment in order to meet its design objectives.
- Autonomy is a difficult concept to pin down precisely, but we mean it simply in the sense that the system should be able to act without the direct intervention of humans (or other agents), and should have control over its own actions and internal state.
- Examples of agents in real life
  - Real estate agents
  - Travel agents
  - Intelligence agents
- In vehicle diagnosis, we define agents that can perform certain fault detection that specified for the agents.
  - Signal agents and vehicle fault agents
DDAS
a Distributed Diagnostics Agents Model

A recording of signals

Agent_A: primary signal: A
reference signals: A1, A2, ...

Agent_B: primary signal: B
reference signals: B1, B2, ...

Agent_C: primary signal: C
reference signals: C1, C2, ...

Device_Agent:

- case library
- case inference engine

Diagnostic result of
signal A

Diagnostic result of
signal B

Diagnostic result of
signal C

Diagnostic result of
other

Fault diagnosis of signal A

Fault diagnosis of signal B

Fault diagnosis of signal C

Fault diagnosis of other signals

Diagnostic result of other signals

Other diagnostic agents

Device diagnosis results: ranked similar cases
Agents in DDAS

- **Multiple SDA (signal diagnostic agents)**
  - A SDA is responsible for the fault diagnosis of one particular signal using either a single signal or multiple signals depending on the nature of the signal.
  - Each SDA is developed using a common framework that incorporates technologies in signal segmentation, automatic signal feature selection, and machine learning.
  - The SDAs can concurrently execute their tasks.

- **One DDA (Device Diagnostic Agent)**
  - The DDA applies case based reasoning (CBR) technique to the output from the SDAs to obtain the fault diagnosis of the machine.
**Vehicle Fault Diagnosis**

**Vehicle Diagnostic agent**
- Major functions:
  - Case representation
  - Case matching
  - Adding and deleting cases

**Major Functions of a Diagnostic Agent**
- Signal feature extraction
- Optimal feature selection
- Signal segment fault detection
- Signal fault detection
- Fuzzy learning
- Fuzzy inference

**ADSAS:**

- **TP Agent**
- **RPM Agent**
- **IAC Agent**
- **IAT Agent**
- **ECT Agent**
- **MAF Agent**

**RPM(M) agent**
- Multiple reference signals: *FUEL, *MAF, *SPARKAD

**TP(M) agent**
- One reference signal: *TPCT

**IAC(M) agent**
- Multiple reference signals: *TP, *MAF

**MAF(M) agent**
- Multiple reference signals: *FUEL, *MAF, *SPARKAD

**Other Agents**
- Idlerolling agent
- Vacuum leak agent
- Misfire agent
Three Levels of Diagnostics

- Signal segment level
- Signal level
- Vehicle level
Three Levels of Functions in SDA

- **Level 1:**
  - automatic signal segmentation
  - automatic feature extraction and
  - automatic feature selection.

- **Level 2:**
  - an intelligent system that determines whether a signal segment is F, N/F, or U/C through segment-by-segment analysis.

- **Level 3:**
  - an intelligent system to detect whether the primary signal indeed is faulty.
    - uses the diagnostic features of all segments within the signal to make the decision.
Illustration of Major Functions in a SDA

- Signal segmentation
- Signal feature extraction
- Segment fault detection
- Diagnostic agent training
  - Automatic feature selection
  - Generate fuzzy agent-KB
  - Generate fuzzy seg-KB
- Training data: signals with segments marked as normal or abnormal
- Expert knowledge:
  -- Problem description
  -- Algorithm flowchart

Signal segments

<segment_normal, segment_abnormal, unknown>

<segments_1, segments_2, ..., segments_n>
ADSAS: Applications

- ADSAS: a complete system for research and development
  - Diagnosis
  - prognosis
- ADSAS-Training agents
- ADSAS-Operation
- ADSAS-WEB
  - ADSAS SERVER
  - ADSAS CLIENT
ADSAS
Training Diagnostic Agents

• **TP Agent**
  Signal features:

• **RPM Agent**
  Signal features:

• **IAT Agent**
  ..........

• **IAC Agent**
  ..........

• **ECT Agent**
  ..........

• **MAF Agent**
  ..........

• **IAC(M) agent**
  ..................

• **MAF(M) agent**
  ..................
ADSAS
Diagnostic Agents in Operation

- Matching cases repair suggestions
- Fault signals
- Fault signal segments
**ADSAS**
**WEB and Wireless Accessible**

**DDAS SERVER**

- Repair suggestions
  - Fault signals
  - *Fault signal segments*

- Repair instructions

- Wireless transmission
**ADSAS’s Data Visualization Tools (1)**

- **DDAS Data Visualization Function**
  - Displaying multiple signals, analog and/or digital, simultaneously
  - Displaying a fixed line at trigger point.
  - Displaying a floating/moving vertical data line and the values of each signal at the floating vertical line
  - Display signal names beside the plots. The text color of signal names matches the color of the signals. Click on a signal name will highlight that signal.
  - Analog signals on background remain unchanged and display the highlighted signal with its range/scale values.
  - Display full name of selected signal in status bar (lower right corner of the window).
Displaying multiple signals, analog and/or digital, simultaneously

- Displaying a fixed line at trigger point.
- Displaying a floating/moving vertical data line and the values of each signal at the floating vertical line
• Display signal names beside the plots. The color of signal names matches the color of the signals. Click on a signal name will highlight that signal.
• Signals on background remain unchanged and display the highlighted signal with its range/scale values.
• Display full name of highlighted signal in status bar.
In the tree control view, display the list of signals in three groups: Input, Output, and Other. Highlight the signals that are plotted.

Quick deselect feature (repeat double click on a signal name in the tree control list will remove that signal from plot).

Zoom feature and quick “Unzoom” button.

Define new signal dialog.

Define new signal agent dialog.

Monitor dialog.

Displaying signal segments and allowing engineers to mark segments as ABNORMAl or NORMAL.
• In the tree control view, display the list of signals in three groups: Input, Output, and Other. Highlight the signals that are plotted.
• Quick deselect feature (repeat double click on a signal name in the tree control list will remove that signal from plot).
• Zoom in and quick “Unzoom” button.
Monitor Dialog

In Tree Control View, highlighted signals belong to the selected monitor.
Define New Signal (second figure shows the dropdown list of ALIAS)
Define New Agent

Name
Reference
- ACCS - ACCS
- ACCS - ACCS_x1101-0_x
- ACP - ACP
- ACP - ACC_x1102-0_x
- AIR - AIR
Segmentation
- Multiscale, DB1, 0.150, 0.400, 4
Features
Edit 'Output' and 'OutDetail' information, and press 'Add' button to add to the list box below:
Output: COPY, 1.000, 5, Both
OutDetail: , FALSE
Add
Remove
OK
Cancel
Fuzzy Learning/Testing: Train/Test Agent(s) Dialog

-- Select one or more Agents to be trained or used for testing.
-- Select one or more Vehicle Recordings to test or use for training.
-- Choose Train Agents or Test Vehicles
-- Click BEGIN button to start.
Agent Diagnostic Step 1: 
Load a vehicle recording and save project by clicking “File->Save Project”.
Agent Diagnostic Step 2: Open “Learning->Train/Test Agent(s)…” dialog to do testing.
Agent Diagnostic Step 3: View agent testing results. In the tree structure, the signals agents in RED indicate signal faults detected, the signals agents in GREEN indicated normal signals detected.

For example: RPM agent detected a Dip in RPM signal which indicates a stall during acceleration, (RPM->0).
Major components in DDAS

- Signal segmentation
- Signal feature extraction and selection
- Fuzzy logic and its application in signal segment fault detection and signal detection
- Case Based Reasoning used in Vehicle Fault Diagnosis
Signal Segmentation

- Signals are often analyzed, by human or computer, segments by segments.
  - When analyzing signals, the question of window size always arises.
  - A common method is to divide a signal into a fixed size.
    - Simple
    - What is a proper window size?
    - What is the problem?
Signal Segmentation Based on Physical Events

- most vehicle signals are irregular, non-periodic signals composed mainly of sharp transients
  - often in the form of edges and
  - periods of relative stability.

- Most of the transients and stable sections of the signal have some physical significance and are related to the vehicle’s behavior over time.
  - Acceleration (rising)
  - Deceleration (falling)
  - Cruise (flat)
Signal Segmentation Using Wavelets(I)

- Objective is to detect sudden changes in a signal
  - Places where flat to rise, rise to fall, etc.
  - We referred to these places as edges or boundaries

- Discrete Wavelet Transformation (DWT)

\[ C(a,b) = C(j,k) = \sum_{n \in \mathbb{Z}} s(n) g_{j,k}(n) \]

where

\[ a = 2^j, b = k2^j, j \in \mathbb{N}, k \in \mathbb{Z} \]

- \( s(n) \) is a discrete signal function
- \( a \) is called scale, \( b \) is shift,
- \( C(a,b) \) are coefficients with various \( a \) and \( b \)
- \( g(n) \) is the wavelet filter such as Harr, Daubechies, etc.
Signal Segmentation Using Wavelets(II)

- Decomposition of a signal with DWT
  - Given a signal $s(n)$ of length $N$, the DWT consists of at most $\log_2 N$ stages.
  - Starting from $s(n)$, two sets of coefficients can be obtained,
    - Approximate coefficients $CA_1$, obtained by convolving $s(n)$ with a low-pass filter $LoF_D$ following by a downsampling of $2$.
    - Detail coefficients $CD_1$, obtained by convolving $s(n)$ with a high-pass filter $HiF_D$ a downsampling of $2$.
    - Both $LoF_D$ and $HiF_D$ are constructed based on the selected wavelet function $g(n)$.
  - $CA_i$, $i = 1, \ldots$, can be further decomposed using the $LoF_D$ and $HiF_D$ described above.
  - This procedure can be repeated at most $\log_2 N$ times.
Signal Segmentation Algorithm

- A signal is segmented based on multi-levels of details through a DWF using DB1 wavelet function

- Major steps in the algorithm
  - Finding approximate segment boundaries
    - This step uses the wavelet coefficients from a range, \([\text{Lmin, Lmax}]\), of detail coefficient levels to place segment boundaries very close to their correct location in the signal.
    - A recursive, multi-scale procedure is used to divide large segments into multiple smaller segments for further detailed analysis.
  - Combining segments of same state
    - adjacent segments that have the same state are merged.
  - Fine tuning segment boundaries
    - This step looks at a small neighborhood, typically one or two samples, around the segment boundaries and shifts the boundaries to more optimal locations.
    - This step also removes any steady states (idle or cruise) that are too short to be significant
Examples:
Displaying signal segments and allowing engineers to mark segments as **ABNORMAL** or **NORMAL**.
Multiple Signal Segmentation

- For multiple signal agents, each agent needs to detect certain faults using more than one signals.
  - We assume all signals belonging submitted to the vehicle at a given time are sampled simultaneously.

- How are we going to segment these signals?
Examples of Multiple Signal Segmentation
Features of Interests

- Abnormal signal value
- Rolling
- Noise
- Sharp transitions
Feature Representation

- Feature families:
  - Basic statistical features: maximum, minimum, average, value range, length of the segment, etc
  - Wavelet energy: normalized sum of square of detailed wavelet transform coefficients
  - Wavelet average: average of detailed wavelet transform coefficients
  - X-Centroid: horizontal gravity center of detailed wavelet transform coefficients
  - Convexity/concavity: ratio of the convex/concave detailed wavelet coefficients in the segment
Feature Selection and Pattern Recognition

- A pattern classification system often involves two stages of development:
  - extraction and selection of features that can be used to discriminate pattern classes,
  - classification that draws class boundaries in the selected feature space.

- Research has shown that many classification algorithms trained using supervised machine learning degrade in performance when provided with many features that are not necessary for prediction.
**Automatic Feature Selection**

- **Objective of feature selection**
  - From a large pool of features, automatically select a subset of features that are most useful in segment-based fault detection.

- **The most common framework for feature selection**
  - Define criteria for measuring the goodness of features
    - The most method is evaluate the performance of feature subsets on an available training data set.
  - Use a search algorithm to find an optimal/sub-optimal set of features in a feature space based on the criteria.
Two categories of methods

- The ultimate performance of the subsequent classification algorithm that uses the selected feature set to perform pattern classification.
A hybrid feature selection algorithm

- Use three methods to measure different statistics of data samples in the feature space $\Omega$
  - linear separability
  - pairwised interclass distance
  - histogram
- Derive a vector that represents a ranking order of all the features in $\Omega$
- Select the most promising features based on the rankings using a Bayesian classifier with EM (Expectation Maximization)
Block Diagram of a hybrid automatic feature selection algorithm

Original Feature Space

Features

Preliminary Statistical Selection
* Pairwise Distance
* Linear Separability
* Histogram Overlap

Feature Ranked Space

Backward Sequential Selection Algorithm

True
Better Classification Result?

False
Replace Feature, Throw away another

Partially Optimized Feature Space

Bayesian Classifier

False
Feature Set Small/Good Enough?

Add 1 Feature In

Forward Sequential Selection Algorithm

True
Better Classification Result?

Replace Feature, Add In another

Partially Optimized Feature Space

quasi Optimal Feature Set

False

Bayesian Classifier

False

Feature Set Big/Good Enough?

Add 1 Feature In
**Linear Separability Measurement**

- Derived from the concept of linear discriminant function

\[
Z = \frac{n_1 n_2}{n_1 + n_2} \frac{n_1 + n_2 - p - 1}{(n_1 + n_2 - 2)p} D^2
\]

- \( p \) is the dimension of the input vector
- \( D \) is the Mahalanobis distance between the centroids of two classes
- \( n_1 \) and \( n_2 \) are the number of samples of two classes \( C_1 \) and \( C_2 \) respectively within a training set \( X \)

- If \( Z \) is high, the selected feature set is considered to have good linear separability
Generating Ranking Vector using Linear Separability

- The ranking vector $R_z = \{r_z^1, r_z^2, \ldots, r_z^n\}$ is generated using an iterative procedure that evaluates all the features in $\Omega$ using the linear separability measure.

- The ranking $r_z^i$ for the $i$th feature, for $i = 1, \ldots, n$, is an integer such that $1 \leq r_z^i \leq n$ and the $i$th feature is selected to the feature set at the $r_z^i$th iteration in the above procedure.
An example of generating feature ranking using linear separability

- At the first iteration

<table>
<thead>
<tr>
<th>i</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z</td>
<td>1.2185</td>
<td>0.3123</td>
<td>0.3159</td>
<td>0.1966</td>
<td>1.1558</td>
<td>0.3824</td>
<td>0.4234</td>
<td>0.1836</td>
<td>0.0162</td>
<td>0.8770</td>
<td>0.0879</td>
<td>0.1391</td>
</tr>
</tbody>
</table>

- At the second iteration

<table>
<thead>
<tr>
<th>i</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
<th>20</th>
<th>21</th>
<th>22</th>
<th>23</th>
<th>24</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z</td>
<td>0.1617</td>
<td>0.0563</td>
<td>0.0312</td>
<td>0.0735</td>
<td>0.0156</td>
<td>0.0113</td>
<td>0.0106</td>
<td>0.0605</td>
<td>0.3647</td>
<td>0.0907</td>
<td>0.7552</td>
<td>0.2725</td>
</tr>
</tbody>
</table>

- At the third iteration

<table>
<thead>
<tr>
<th>i</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z</td>
<td>-</td>
<td>1.9600</td>
<td>12.398</td>
<td>7.9833</td>
<td>0.6177</td>
<td>1.4819</td>
<td>12.369</td>
<td>8.1867</td>
<td>3.7357</td>
<td>0.6531</td>
<td>6.4052</td>
<td>7.0864</td>
</tr>
</tbody>
</table>

- At the third iteration

<table>
<thead>
<tr>
<th>i</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
<th>20</th>
<th>21</th>
<th>22</th>
<th>23</th>
<th>24</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z</td>
<td>2.4568</td>
<td>3.0714</td>
<td>8.0787</td>
<td>2.6067</td>
<td>4.7748</td>
<td>3.9317</td>
<td>4.3681</td>
<td>6.6861</td>
<td>1.4977</td>
<td>2.1754</td>
<td>0.7190</td>
<td>1.8413</td>
</tr>
</tbody>
</table>

- At the third iteration

<table>
<thead>
<tr>
<th>i</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
</table>

- At the third iteration

<table>
<thead>
<tr>
<th>i</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
<th>20</th>
<th>21</th>
<th>22</th>
<th>23</th>
<th>24</th>
</tr>
</thead>
</table>

- $\Lambda = \{1, 3, 23\}$, $r_z^1 = 1, r_z^3 = 2, r_z^{23} = 3$
**Pairwised Interclass Distance Measurement**

Pairwised interclass distance measurement evaluates each feature \( i \) by calculating the interclass distance \( J_{-i} \) after the \( i \)th feature is removed from a feature set.

\[
J_{-i} = \frac{1}{2M} \sum_{r=1}^{m} P(\omega_r) \sum_{s=r+1}^{m} P(\omega_s) \frac{1}{N_r N_s} \sum_{k=1}^{N_r} \sum_{l=1}^{N_s} d(\xi_{rk}, \xi_{sl})
\]

- \( N_r \) is the number of samples belonging to class \( \omega_r \)
- \( m \) is the number of classes
- \( P(\omega_r) \) is the class probability of \( \omega_r \)
- \( d(\xi_{rk}, \xi_{sl}) \) is the distance between the \( k \)th sample of class \( \omega_r \) and the \( l \)th sample of class \( \omega_s \)
- \( M \) is the dimension of the feature vector and is used to eliminate the effect of the different dimensions of the feature set under evaluation.
Generating Ranking Vector using Pairwised Interclass Distance Measurement

- For a feature space of n dimensions, we generate a vector of $n$ between-class distances
  \[ J = [J_{-1}, J_{-2}, \ldots, J_{-n}] \]
- The ranking vector $R_d = \{r_d^1, r_d^2, \ldots, r_d^n\}$,
  - $r_d^i$ is an integer and $1 \leq r_d^i \leq n$ for any $1 \leq i \leq n$
  - for any $i$ and $j$, $r_d^i < r_d^j$ if and only if $J_{-i} < J_{-j}$. 
An example of generating feature ranking using Pairwised Interclass Distance Measurement

- Interclass distance after removing one dimension

<table>
<thead>
<tr>
<th>i</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>$J_{-i} (\times 10^{-3})$</td>
<td>4.9931</td>
<td>5.0212</td>
<td><strong>4.8933</strong></td>
<td>4.9718</td>
<td>4.9486</td>
<td>5.0001</td>
<td><strong>4.8582</strong></td>
<td>4.9771</td>
<td><strong>5.0392</strong></td>
<td>5.0111</td>
<td><strong>5.0302</strong></td>
<td>4.9435</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>i</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
<th>20</th>
<th>21</th>
<th>22</th>
<th>23</th>
<th>24</th>
</tr>
</thead>
<tbody>
<tr>
<td>$J_{-i} (\times 10^{-3})$</td>
<td>5.0111</td>
<td><strong>5.0372</strong></td>
<td>4.9597</td>
<td>4.9739</td>
<td>5.0101</td>
<td><strong>5.0272</strong></td>
<td>4.9911</td>
<td><strong>4.8954</strong></td>
<td>5.0181</td>
<td>5.0211</td>
<td>5.0091</td>
<td><strong>4.8101</strong></td>
</tr>
</tbody>
</table>

- The ranking vector is

$R_d = \{12, 20, 3, 8, 6, 13, 2, 10, 24, 17, 22, 5, 16, 23, 7, 9, 15, 21, 11, 4, 18, 19, 14, 1\}$. 
Overlapped Feature Histogram Method

- For feature i, the overlapped feature histograms between classes i and j, \( h_j^i (1) \), \( h_j^i (2) \), …, \( h_j^i (\eta^i) \) are calculated as follows:
  - Divide the range of feature i into \( \eta^i \) intervals,
  - \( h_j^i (x) \) is the number of data samples belonging to class j and whose ith feature values fall in the xth bin, for \( x = 1, 2, \ldots \eta^i \) and for \( j = 1, 2, \ldots, m \).

- The degree of overlapped classes in different bins for feature i is measured by

\[
O_i = \sum_{j=1}^{m-1} \sum_{k=j+1}^{m} \sum_{x} \min(h_j^i (x), h_k^i (x))
\]
Generating Ranking Vector using Overlapped Feature Histograms

- If the $ith$ feature is good for classification, then $O_i$ should be small.
- An Example, $O_i$ is 0.1
- Ranking vector $R_h = \{r_h^1, r_h^2, \ldots, r_h^n\}$, where $1 \leq r_h^i \leq n$, and for any $i$ and $j$, $r_h^i < r_h^j$ if and only if $O_i < O_j$. 
A Hybrid Feature Selection Method

- \( R = \{r^1, r^2, \ldots, r^n\} = (\omega_d R_d + \omega_z R_z + \omega_h R_h) \)
- where \( \omega_d + \omega_z + \omega_h = 1 \) and \( \omega_d > 0, \omega_z > 0, \omega_h > 0 \).

- Use the well-known Bayesian classifier with EM (Expectation Maximization) estimation to select a suboptimal subset of features \( \Lambda \) from \( \Omega \) based on the overall ranking vector \( R \).
Feature Selection using Bayesian EM Classifier

Let \( \Lambda' = \{ f_1, f_2 \ldots f_n \} \) and \( r^{f_i} \leq r^{f_{i+1}} \) for \( i = 1, 2, \ldots, n-1 \).

- Remove the lowest ranked feature \( f_k \) from the current feature set \( \Lambda^k = \Lambda' - \{ f_k \} \).

- Evaluate \( \Lambda^k \) using the Bayesian classifier with EM.
  - If the error rate is small, the procedure continues for \( k = k-1 \) and \( \Lambda^k = \Lambda^{k+1} - \{ f_k \} \).
  - The procedure stops when \( E_{-k} \) has a big error increase in comparison with the previous error.
  - The final feature set \( \Lambda = \Lambda^k \cup \{ f_k \} \).
The hybrid feature selection algorithm is evaluated using a fuzzy classification system.

- The system has the capability of automatically generating fuzzy rules and fuzzy membership functions from training data and optimizing fuzzy membership functions.
Experiments(2)

- Training and test data: signal segments:
  - In vehicle diagnosis, various signals recorded can be either normal or abnormal.
  - The data used in the experiments shown in this paper are segments of normal and abnormal RPM signals generated by an automatic signal segmentation algorithm.
  - The segments were labeled as normal or abnormal based on manual examination by several vehicle diagnostics experts.
Experiments(3)

- Each vehicle segment was originally represented in a feature space $\Omega$ of 48 dimensions
  - Features were derived from primarily wavelet transformation, signal magnitude, signal fluctuation, signal concavity and convexity.
- The training data TR contains 599 samples labeled normal and 181 abnormal.
- The test set TEST contains 188 normal samples and 72 abnormal ones.
The hybrid feature selection algorithm used TR to select a subset of $\Omega$. The fuzzy classifier used in the experiments was also trained on TR data.

Results generated from TEST data.

<table>
<thead>
<tr>
<th>Feature sets</th>
<th># of fuzzy rules</th>
<th>Classification accuracy on TR</th>
<th>Classification accuracy on TEST</th>
</tr>
</thead>
<tbody>
<tr>
<td>All features in $\Omega$, 48 features</td>
<td>895</td>
<td>97.08%</td>
<td>61.92%</td>
</tr>
<tr>
<td>Hybrid method 7 features</td>
<td>133</td>
<td>89.49%</td>
<td>83.84%</td>
</tr>
</tbody>
</table>
Result Analysis(1)

- When we trained the fuzzy diagnostic system with the original feature set $\Omega$, we obtained a fuzzy knowledge base containing 895 fuzzy rules.
  - The resulting fuzzy diagnostic system has very good performance on the training data, but it gave dismal performance on the test data.
Result Analysis (2)

- The hybrid feature selection algorithm selected 7 features out of 48.
  - The fuzzy diagnostic system generated 133 rules.
  - The resulting fuzzy diagnostic system has increased classification accuracy over 22% better classification result on the test set.
Result Analysis (3)

- The performance on the training data is very close to the performance on the test data.

  - A classification training using the feature set selected by the hybrid feature selection algorithm has a better generalization capability, which is extremely important in many applications of pattern recognition.
Example of Feature Selection result:

- In RPM multiple signal agent, our feature selection algorithm reduced a set of 220 possible to 11 final features.
- More results will be seen later.
FUZWIN

- An intelligent fuzzy system for building a fuzzy classification system.
  - learning mode
  - diagnostic mode

- In DDAS, it is used to train SDAs to perform
  - signal segment fault detection and
  - signal fault detection
Learning Diagnostic Knowledge

- Training Data
- Fuzzy Rule Generator
- Data Input Interface
- MSF Optimizer
- Fuzzy Rules
- MSF’s
- Engineer Experts
- Linguistic Interface
Fuzzy Engineering Diagnosis

- Testing Data
- Fuzzy Rules
- MSF’s
- Belief of Fault

Data Input Interface → Inference Engine → Data Output Interface
Characteristics of Automotive fault diagnosis

- Knowledge is incomplete and vague due to the complexity of the modern vehicles
- Different vehicle models have different engineering features
- When vehicles of a new model are being manufactured, we have very few data for learning.
- Data samples of good and bad vehicles having are often unbalanced
  - the number of good vehicles is overwhelmingly larger than the number of bad vehicles
A complete set of fuzzy rules contains \( n^m \) fuzzy rules.
- \( n \) is the number of fuzzy terms.
- \( m \) is the number of fuzzy parameters.

A fuzzy rule generation can be computationally expensive.
A fast fuzzy rule generation algorithm

- Initialize the critical parameters of the fuzzy membership functions
- The fuzzy rules and the membership functions are generated and optimized in an iterative fashion.
- Dominate rules and dominate samples are the key concepts in this algorithm
Computation at each iteration

- Compute dominant rules
- Use Winner-Take-All to extract fuzzy rules from dominant rules
- Compute priority value for each fuzzy rule
- Clustering
- Optimize fuzzy membership functions
Computing dominant rules

- Assume we have rule $r$ for a data sample $s$, if
  - its belief value for control variable $x_1$ in fuzzy term $L$ is larger than any other fuzzy terms, and
  - its belief for value control variable $x_2$ in fuzzy term $M$ is larger than its belief value for $x_2$ in any other fuzzy terms,
  - and its truth value is $y$ is $H$

- $S$ has a dominant rule
  - “if $x_1$ is $L$ and $x_2$ is $M$, then $y$ is $H$”
Extracting fuzzy rules

- Dominant rules that have the same antecedent may have different consequence
  - If $x_1$ is Low and $x_2$ is Low Then $y$ is Low
    - support samples 60, average belief value: 0.72
  - If $x_1$ is Low and $x_2$ is Low Then $y$ is Medium
    - support samples 100, average belief value: 0.8
  - If $x_1$ is Low and $x_2$ is Low Then $y$ is High
    - support samples 40, average belief value: 0.65

- Winning rule:
  - If $x_1$ is Low and $x_2$ is Low Then $y$ is Medium
Computing priorities of fuzzy rules

Each fuzzy rule is associated with a priority defined as an accumulation of the belief values of its dominated samples

$$\frac{1}{N} \sum_{s=1}^{N} b_s$$

where $N$ is the number of data samples in the whole training set, and $b_s$ is the belief value that $s$-th data sample fires this rule.
Updating Fuzzy rules

- If the newly generated dominant rule does not exist, it is added into the rule base.

- If the rule already exists
  - Add the priority value of the new rule to the existing one.

- If the new rule is conflicting with an existing rule
  - Subtract the priority of the new rule from this existing one.
    - If the priority value of the existing rule becomes negative, the rule is replaced with the new one.
Clustering

- For each fuzzy rule, "IF (x1 is c1 and x2 is c2 and ... and xn is cn), THEN (y is c)."
  - The samples in the training set that have truth value, y is c, form a hyper-cube called a cluster.
  - For each variable, each of its fuzzy terms has a range of value.
  - The ranges of all the fuzzy terms of all the control variables form the boundary of the cluster.
Optimize fuzzy membership functions

- Compare the critical parameters of the fuzzy membership functions with cluster centers.
- If the critical parameters of MBF are very close to the corresponding cluster centers, then stop the iteration.
- Otherwise:
  - The critical parameters of fuzzy membership functions are updated to the cluster centers.
  - Repeat the iteration.
Heap Structure to Store Fuzzy Rules

- Fuzzy rules are stored in a heap structure
  - Rules with the least priority are on the top of the ones with higher priority

- Advantages
  - Rule with the least priority can be discarded easily
  - Rule search is more efficient, since the rules are already sorted
Segment-based Fault Detection
Using Fuzzy Logic

- **Fuzzy learning**
  - Every agent must generate its own knowledge base through fuzzy learning
  - A knowledge base is a collection of rules and related fuzzy variables
    - IF WAverage is LOW AND Maximum is LOW THEN segment is GOOD
  - A training set contains feature vectors representing good segments

- **Fault detection**
  - For every segment of an input signal, an agent applies fuzzy inference to the segment and outputs one of the following results:
    - Good, bad, unknown
Customize a project (System Interface I):
Modify the membership function (System Interface II):
Modify the fuzzy rules (System Interface III):

![Image of a rule editor window with parameters and rules listed]
Report the data information (System Interface IV):
Intelligent Signal Segment Fault Detection using Fuzzy Logic

- We describe a fuzzy learning algorithm that
  - learns from good vehicle signals only and
  - the resulting knowledge base is used to detect GOOD and BAD signal segments.

- The fuzzy system has been implemented in all the
  SDAs in the ADSAS system for detecting signal segment faults.
Symbolic Representation of Signal Segments

- After signal segmentation, a signal is partitioned into a sequence of segments, each of which is labeled by a vehicle state corresponding to a meaningful physical event.

- *In general, a SDA* agent has a sequence of segments \( \{ s_1^p, s_2^p, \ldots, s_k^p \} \) to represent its primary signal, and a sequence of segments \( \{ s_1^{r_i}, s_2^{r_i}, \ldots, s_k^{r_i} \} \) to represent each reference signal \( i \).
Segment Feature Representation

- Each segment, $s_j^q$, $q = r_i$ or $p$, is represented by a feature vector calculated based on the features selected by feature extraction functions and a feature selection process.
- The feature vector of each signal segment is sent to a fuzzy intelligent system to detect segment faults.
Fuzzy Modeling

- Let a feature vector be $\Lambda = \{x_1, x_2, \ldots, x_m\}$
  - $X_i, i = 1, \ldots, m$, are called feature parameters
- Feature parameters are modeled as fuzzy control variables
- One solution variable $y$ is used to represents whether a segment is G/B/U.
- Each control variable $x_i$ is associated with a set of fuzzy terms $= \{\alpha_i^1, \ldots, \alpha_i^{p_i}\}$
- The solution variable $y$ is associated with fuzzy terms $= \{\tau_1, \ldots, \tau_q\}$. 
Fuzzy membership functions

- The fuzzy membership functions associated with a fuzzy variable can be collectively defined by a set of critical parameters that uniquely describe the characteristics of the membership functions.

- In this project we use triangular functions to model fuzzy membership functions.
Problem in Labeling Training Data

- The labeling of GOOD and BAD signal segments is usually a manual process, and it is not a trivial task since in many cases GOOD or BAD segments are subjective to individual engineers.
One solution

- Our solution to this difficult quandary is to train a fuzzy fault detection system with only GOOD data samples.
- This has several advantages:
  - Good data is relatively easy to collect and relatively reliable.
Challenges in fuzzy learning from GOOD data only

- A fuzzy learning algorithm usually makes fuzzy membership functions cover the entire input domain based on the training data.
  - But in the case of learning from GOOD class data only, this input domain is only the domain for good data. This results in fuzzy rules that will incorrectly classify most out-of-bounds fault data as good.

- When will the fuzzy system output “segment is ABNORMAL?”
We dealt these two issues by modifying the fuzzy membership functions and rule generation process and the use of the UNKNOWN output.

We use an example to illustrate this process.

Consider a single fuzzy variable called MAF_MIN (standing for MAF (Mass Air Flow) signal minimum value)

- with three fuzzy terms, LOW, MEDIUM and HIGH.
- Notice that the LOW and HIGH terms have shoulders that stretch out to infinity, thus covering all possible input values outside of the domain of the training data.
Data that are not seen by the fuzzy learning system and well outside the window shown here may be called

- Low when it is actually “very low” and
- High when it is actually “very high.”

However, these “very low” and “very high” terms are not known to the fuzzy system because it never saw these abnormal signal segments.
Now assume we have generated the following fuzzy rule:

- *IF MAF_MIN is LOW THEN segment_is_good is HIGH*

- If there was a MAF signal segment that had a MAF_MIN value of 0.

- The MAF value is too low because engineering knowledge tells us that MAF should never be 0, in fact it should always be above about 0.75 Volts at all times when the car is running.

- Therefore this signal segment should be detected as abnormal or BAD.

- However this existing fuzzy rule will give the classification result as “The signal segment is GOOD”
Our experience with vehicle diagnostics show that abnormal signals often show such out-of-bounds features.

The solution to this particular problem is to limit the coverage of the fuzzy terms to the input space inhabited by GOOD data.

Any part of the domain that is not covered by GOOD data may be assumed to hold another, unknown, class of data that the fuzzy system does not know the details during the training.
Algorithm that Modifies Fuzzy Membership Functions

- Given a set of fuzzy rules and membership functions generated from GOOD data samples
- For each of the left and right end shoulder term, replace it with a new term called such as LEFT_OUT_OF_BOUNDS or RIGHT_OUT_OF_BOUNDS respectively.
- The boundary of these new terms should not overlap the domain of the GOOD data so as not to cause unnecessary false alarms.
- For each of the new terms, generate the following new rule
  - IF fuzzy_variable is NEW_TERM THEN segment_is_good is LOW
    - where either LEFT_OUT_OF_BOUNDS or RIGHT_OUT_OF_BOUNDS can be substituted for NEW_TERM
An illustration of membership function modification

(a) shows the fuzzy end terms (shoulders) as they are normally generated by a fuzzy learning algorithm using data samples from good class only. (b) shows the new end terms and the modified end terms as well as the new end terms that are used to define rules with a consequence segment_is_good is low.
Summary of fuzzy membership function modification

- With these changes in fuzzy membership functions and rule generation data outside the fuzzy term boundaries is no longer misclassified.
- We should apply engineering knowledge about signals and vehicle systems, and scientific knowledge to segment features in terms of generating new appropriate fuzzy terms.
Fuzzy rule analysis(1)

- The second issue is that fuzzy learning from only good data results in a set of fuzzy rules with the same consequence: segment_is_good is HIGH, except those modified by the step above.

- In many fuzzy rule generation algorithms, the fuzzy rules may not cover all possible scenarios of input data due to the fuzzy rule pruning for efficiency.

- The areas in the multi-dimensional feature space that are not represented by the input data are not covered by fuzzy rules.
An illustration

Fuzzy Rules Cover
Good Data

No Fuzzy Rules,
Data marked
UNKNOWN

Solution:
If an input signal segment does not fire any fuzzy rule, the fuzzy system should interpret it as possibly a bad segment, i.e. segment_is_good = LOW.

For simplicity, the horizontal and vertical lines indicate the boundaries of fuzzy terms, although in most applications the boundaries are curved.

The fuzzy rules cover only clusters (shaded blocks) containing good data.

Any input data that falls in the blank areas does not fire any rules.
Fuzzy learning algorithm from GOOD training examples only (1)

- Step 1: Use a supervised fuzzy learning algorithm to generate a fuzzy knowledge base that contains a set of fuzzy rules, fuzzy variables and fuzzy membership functions.
- Step 2: For each control variable, generate at most two new end fuzzy terms as appropriate as described above.
Fuzzy learning algorithm from GOOD training examples only (2)

- Step 3: For each new fuzzy term NEW_TERM, add fuzzy rules:
  - IF fuzzy_variable is NEW_TERM THEN segment_is_good is LOW

- Step 4: For those combinations of fuzzy control variables and fuzzy terms that do not occur in the antecedence of any fuzzy rules, add new fuzzy rules that use these combinations as antecedence and use consequence such as segment_is_good is LOW or segment_is_good is UNCERTAIN.
### Experimental results(1)

<table>
<thead>
<tr>
<th>Agent</th>
<th># of GOOD segments in training data</th>
<th>Classified GOOD segments</th>
<th># of GOOD segments classified as UNKNOWN or BAD</th>
<th>% correctly classified segments</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECT</td>
<td>537</td>
<td>537</td>
<td>0</td>
<td>100%</td>
</tr>
<tr>
<td>MAF</td>
<td>389</td>
<td>389</td>
<td>0</td>
<td>100%</td>
</tr>
<tr>
<td>RPM</td>
<td>602</td>
<td>602</td>
<td>0</td>
<td>100%</td>
</tr>
</tbody>
</table>
# Experimental results (2)

<table>
<thead>
<tr>
<th>Agent</th>
<th>No. of GOOD segments</th>
<th>No. of Classified GOOD segments</th>
<th>No. of GOOD classified as UNKNOWN or BAD</th>
<th>% classified as GOOD</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECT</td>
<td>111</td>
<td>87</td>
<td>24</td>
<td>78.38%</td>
</tr>
<tr>
<td>MAF</td>
<td>138</td>
<td>128</td>
<td>10</td>
<td>92.75%</td>
</tr>
<tr>
<td>RPM</td>
<td>74</td>
<td>69</td>
<td>5</td>
<td>93.24%</td>
</tr>
</tbody>
</table>
## Experimental results(3)

<table>
<thead>
<tr>
<th>Agent</th>
<th>No. of BAD segments</th>
<th>No. of Classified BAD segments</th>
<th>No. of BAD segments classified as GOOD</th>
<th>% classified as BAD</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECT</td>
<td>69</td>
<td>65</td>
<td>4</td>
<td>94.20%</td>
</tr>
<tr>
<td>MAF</td>
<td>579</td>
<td>390</td>
<td>189</td>
<td>67.36%</td>
</tr>
<tr>
<td>RPM</td>
<td>127</td>
<td>54</td>
<td>73</td>
<td>42.52%</td>
</tr>
</tbody>
</table>
## Experimental results (4)

<table>
<thead>
<tr>
<th>Agent Name</th>
<th>GOOD signals</th>
<th>% correctly classified</th>
<th>BAD signals</th>
<th>% correctly classified</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECT</td>
<td>4</td>
<td>100 %</td>
<td>11</td>
<td>100 %</td>
</tr>
<tr>
<td>MAF</td>
<td>4</td>
<td>75 %</td>
<td>5</td>
<td>100 %</td>
</tr>
<tr>
<td>RPM</td>
<td>1</td>
<td>100 %</td>
<td>5</td>
<td>100 %</td>
</tr>
</tbody>
</table>

The entire signal fault diagnostic results of all three agents generated based on the segmentation fault detection results running on the test set.
Signal Fault Detection Using Fuzzy Logic

Fault detection at the signal level gives better understanding of faulty behavior of the entire signal:

- An abnormal signal usually looks different from a normal signal:
  - More segments labeled ABNORMAL
  - Abnormal segments cluster close together
  - Abnormal segment data has larger distance from the fired fuzzy rules if the case of learning from NORMA data only.

- Signal fault detection is developed based on the fault detection results from all the segments of the signal:
- Signal features used in the fault detection at the signal level:
  - Average Bad Belief Value
  - Average Good Belief Value
  - Average Distance to Good Rule
  - Maximum Bad Cluster Size
  - Number of Bad Clusters
  - Percentage of Signal that is Bad

- A fuzzy knowledge base, signal_KB, is generated from a training data of normal and abnormal signals

- Signal_KB is shared by all SDAs.
Vehicle Fault Diagnostics

-- A Case Based Reasoning Approach

- When a SDA reports a signal fault, we know there is a fault, but WHAT is the fault?
- Vehicle fault detection is performed by the VFD agent using a Case Based Reasoning (CBR) approach
- A CBR system has a knowledge base, case library, that stores known bad cases
- When a vehicle case submitted by a user, SDAs are being called, and the VFD searches this “case library” for the cases that best match the scenarios presented by the SDAs.

- Two Issues:
  - Descriptive data for each case, case representation
  - Similarity function for matching two cases
**CBR using Signal Signature**

- **Case Representation:**
  - *Case ID*: used for internal processing and tracking
  - *Recording Name*: Name of the recording signals attached to the case. This is stored with the case information for later reference by the technician or engineer.
  - *Vehicle Info*: This can be any information related to the vehicle; however, it is intended mainly for year/model/engine type information.
  - *Root Cause*: A brief description of the vehicle root cause of the fault.
  - *Description*: General description of the fault in a few lines of text.
  - *Signature Data*: Data describing the recording from a signal agent point of view.
Case Signature

- **Signature Data**
  - three choices available:
    - Raw signal data
    - Feature data
    - SDA outputs (Good, Bad, Unknown)
  - Signature form: \( \{ a_i \}_{i \in [1,N]} \), \( a_i \) is output from \( i \)th agent

- **Advantages**
  - Simple and easy to compute
  - Takes advantage of work done by SDAs
  - As the number of agents in the system grows, the descriptive power of the signature increases

- **Disadvantages**
  - Not descriptive enough for precise matching
Case Matching

- Case matching is performed based on comparing the signature data of the two cases.

- Let the vehicle currently under diagnosis be a “query case” and denote it as $C_q = \{a_1^q, a_2^q, \ldots, a_N^q\}$ and the cases in the case library as $C_m = \{a_1^m, a_2^m, \ldots, a_N^m\}$, where $m = 1, \ldots, K$.

- A distance function, $D(C_q, C_m)$, is defined as

$$D(C_q, C_m) = \sum_{i=1}^{N} d(a_i^q, a_i^m)$$

$d(\ldots)$ is a distance function of two signature vectors, $a_i^q$ and $a_i^m$. 


Signature Vector Distance
Constraints

- Signature vector distance is defined based on the following constraints:
  - $d(G, G) = d(B, B) = d(U, U) = d(NA, NA) = 0$
  - $0 \leq d(U, B) \leq d(U, G) \leq d(G, B)$
  - $d(NA, B) < d(G, B)$
  - $d(NA, B) \leq d(U, B)$
  - $d(NA, G) < d(G, B)$
  - $d(NA, G) < d(G, U)$
  - $d(NA, U) < d(G, U)$

- “NA” is an important feature that DDAS provide. This gives the flexibility to the user:
  - a vehicle case submitted to DDAS does not have to provide the signals to all the agents in DDAS.

- Two approaches to handling NA signature elements
  - Do not include them.
  - Develop a special distance function for NA comparison

$$D(C_q, C_m) = \sum_{i=A}^{N} d(a_i^q, a_i^m) + \sum_{i=NA}^{M} d_{NA}(a_i^q, a_i^m)$$
Signature Vector Distance Function

- The following table represents one distance function based on the constraints presented earlier.

<table>
<thead>
<tr>
<th></th>
<th>G</th>
<th>B</th>
<th>U</th>
<th>NA</th>
</tr>
</thead>
<tbody>
<tr>
<td>G</td>
<td>0.0</td>
<td>1.0</td>
<td>0.7</td>
<td>0.6</td>
</tr>
<tr>
<td>B</td>
<td>1.0</td>
<td>0.0</td>
<td>0.3</td>
<td>0.5</td>
</tr>
<tr>
<td>U</td>
<td>0.7</td>
<td>0.3</td>
<td>0.0</td>
<td>0.5</td>
</tr>
<tr>
<td>NA</td>
<td>0.6</td>
<td>0.5</td>
<td>0.5</td>
<td>0.0</td>
</tr>
</tbody>
</table>

- All distances are constrained to be between 0 and 1
- NAs are biased toward matching with BAD
Case Based Reasoning (1) – find matching cases

After testing, click “Learning->Search/Edit Case Base…” to open CBR dialog.
Select a Vehicle Recording as query case.
Select Matching method.
Click “Find Matches” button to find matching cases.
Case Based Reasoning (2) – result screen

Source cases in case library is displayed in the order of distance from query cases.
Powertrain Control Module (PCM) Signal Diagnosis

- Throttle Position (TP)
- Mass Air Flow (MAF)
- Engine RPM (RPM)
- Engine Coolant Temperature (ECT)
- Idle Air Control (IAC)
- Intake Air Temperature (IAT)
## Single Signal Agents

<table>
<thead>
<tr>
<th>ECT</th>
<th>IAC</th>
<th>IAT</th>
<th>MAF</th>
<th>RPM</th>
<th>TP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single Signal Agent</td>
<td>Single Signal Agent</td>
<td>Single Signal Agent</td>
<td>Single Signal Agent</td>
<td>Single Signal Agent</td>
<td>Single Signal Agent</td>
</tr>
<tr>
<td>Convexity</td>
<td>Max</td>
<td>Avg</td>
<td>Convexity</td>
<td>Max</td>
<td>Convexity</td>
</tr>
<tr>
<td>Max</td>
<td>Avg</td>
<td>WE DB15</td>
<td>Fluctuation</td>
<td>Min</td>
<td>Max</td>
</tr>
<tr>
<td>Avg</td>
<td>WE DB14</td>
<td>WE DB23</td>
<td>Min</td>
<td>Max</td>
<td>Fluctuation</td>
</tr>
<tr>
<td>WA DB12</td>
<td>WA DB13</td>
<td>XC DB12</td>
<td>WE DB30</td>
<td>WE DB12</td>
<td>WE DB10</td>
</tr>
<tr>
<td>WA DB31</td>
<td>WA DB14</td>
<td>XC DB13</td>
<td>XC DB30</td>
<td>WA DB15</td>
<td>WA DB33</td>
</tr>
<tr>
<td>XC DB13</td>
<td>XC DB32</td>
<td>XC DB23</td>
<td>XC DB32</td>
<td>WA DB23</td>
<td>XC DB33</td>
</tr>
<tr>
<td>XC DB20</td>
<td>XC DB33</td>
<td>WE DB25</td>
<td>WE DB31</td>
<td>WE DB31</td>
<td>WE DB35</td>
</tr>
</tbody>
</table>

- **ECT** Single Signal Agent:
  - Convexity
  - Max
  - Avg
  - WA DB12
  - WA DB13
  - WA DB31
  - XC DB13
  - XC DB20

- **IAC** Single Signal Agent:
  - Max
  - Avg
  - XC DB00
  - WA DB13
  - WA DB14
  - Concavity

- **IAT** Single Signal Agent:
  - Avg
  - WE DB34
  - XC DB31
  - XC DB32
  - XC DB33

- **MAF** Single Signal Agent:
  - Convexity
  - Fluctuation
  - Min
  - Max
  - Avg
  - WA DB15
  - WE DB14
  - WE DB23
  - XC DB12
  - XC DB13
  - XC DB23
  - XC DB30

- **RPM** Single Signal Agent:
  - Max
  - Fluctuation
  - XC DB22
  - XC DB33
  - Concavity
  - WA DB12
  - WA DB15
  - WA DB23
  - WE DB31
  - WE DB32
  - WE DB35

- **TP** Single Signal Agent:
  - Convexity
  - Max
  - Min
  - WA DB12
  - WA DB33
  - WE DB12
  - XC DB10
An Example of RPM Agent

Case 1: RPM Hesitation

Root Cause: Mis-aligned Cylinder Identification Sensor (CID) Causing Late Fuel Delivery
Step 1: Segmentation

- Segment states are labeled using colors
- Red: rising, green: falling, brown: steady
## Step 2: Feature Extraction/selection

- Over 60 Features Extracted

- Feature Selection (down to 12 features)

<table>
<thead>
<tr>
<th>Seg#</th>
<th>Rec...</th>
<th>Agent</th>
<th>S...</th>
<th>E...</th>
<th>State</th>
<th>State</th>
<th>Max</th>
<th>Flucuation</th>
<th>XC_DB22</th>
<th>XC_DB33</th>
<th>Concavity</th>
<th>WA_DB12</th>
</tr>
</thead>
<tbody>
<tr>
<td>✔️ 1</td>
<td>Rains2</td>
<td>RPM: OUTPUT</td>
<td>0</td>
<td>34</td>
<td>3</td>
<td>3.00</td>
<td>656</td>
<td>19.4</td>
<td>0.505</td>
<td>0.463</td>
<td>0.500</td>
<td>-1.16</td>
</tr>
<tr>
<td>✔️ 2</td>
<td>Rains2</td>
<td>RPM: OUTPUT</td>
<td>34</td>
<td>51</td>
<td>4</td>
<td>4.00</td>
<td>630</td>
<td>106</td>
<td>0.371</td>
<td>0.510</td>
<td>0</td>
<td>-17.7</td>
</tr>
<tr>
<td>✔️ 3</td>
<td>Rains2</td>
<td>RPM: OUTPUT</td>
<td>51</td>
<td>61</td>
<td>2</td>
<td>2.00</td>
<td>1776</td>
<td>567</td>
<td>0.481</td>
<td>0.187</td>
<td>0</td>
<td>-204</td>
</tr>
<tr>
<td>✔️ 4</td>
<td>Rains2</td>
<td>RPM: OUTPUT</td>
<td>61</td>
<td>66</td>
<td>4</td>
<td>4.00</td>
<td>1778</td>
<td>39.3</td>
<td>0.477</td>
<td>0.590</td>
<td>0.600</td>
<td>5.68</td>
</tr>
<tr>
<td>✔️ 5</td>
<td>Rains2</td>
<td>RPM: OUTPUT</td>
<td>66</td>
<td>101</td>
<td>3</td>
<td>3.00</td>
<td>2077</td>
<td>122</td>
<td>0.492</td>
<td>0.414</td>
<td>0</td>
<td>-24.7</td>
</tr>
<tr>
<td>✔️ 6</td>
<td>Rains2</td>
<td>RPM: OUTPUT</td>
<td>101</td>
<td>114</td>
<td>2</td>
<td>2.00</td>
<td>2256</td>
<td>59.4</td>
<td>0.476</td>
<td>0.327</td>
<td>0</td>
<td>-15.0</td>
</tr>
<tr>
<td>✔️ 7</td>
<td>Rains2</td>
<td>RPM: OUTPUT</td>
<td>114</td>
<td>131</td>
<td>3</td>
<td>3.00</td>
<td>2222</td>
<td>29.1</td>
<td>0.334</td>
<td>0.572</td>
<td>0.294</td>
<td>4.96</td>
</tr>
<tr>
<td>✔️ 8</td>
<td>Rains2</td>
<td>RPM: OUTPUT</td>
<td>131</td>
<td>172</td>
<td>2</td>
<td>2.00</td>
<td>2457</td>
<td>90.2</td>
<td>0.491</td>
<td>0.522</td>
<td>0</td>
<td>-16.0</td>
</tr>
<tr>
<td>✔️ 9</td>
<td>Rains2</td>
<td>RPM: OUTPUT</td>
<td>172</td>
<td>294</td>
<td>4</td>
<td>4.00</td>
<td>2515</td>
<td>635</td>
<td>0.488</td>
<td>0.496</td>
<td>0.385</td>
<td>31.4</td>
</tr>
<tr>
<td>✔️ 10</td>
<td>Rains2</td>
<td>RPM: OUTPUT</td>
<td>294</td>
<td>311</td>
<td>2</td>
<td>2.00</td>
<td>672</td>
<td>43.0</td>
<td>0.705</td>
<td>0.440</td>
<td>0.176</td>
<td>-13.3</td>
</tr>
<tr>
<td>✔️ 11</td>
<td>Rains2</td>
<td>RPM: OUTPUT</td>
<td>311</td>
<td>318</td>
<td>4</td>
<td>4.00</td>
<td>676</td>
<td>81.6</td>
<td>0.249</td>
<td>0.680</td>
<td>0</td>
<td>-17.4</td>
</tr>
<tr>
<td>✔️ 12</td>
<td>Rains2</td>
<td>RPM: OUTPUT</td>
<td>318</td>
<td>327</td>
<td>2</td>
<td>2.00</td>
<td>1847</td>
<td>581</td>
<td>0.523</td>
<td>0.236</td>
<td>0.333</td>
<td>-209</td>
</tr>
<tr>
<td>✔️ 13</td>
<td>Rains2</td>
<td>RPM: OUTPUT</td>
<td>327</td>
<td>355</td>
<td>4</td>
<td>4.00</td>
<td>1853</td>
<td>276</td>
<td>0.521</td>
<td>0.534</td>
<td>0.786</td>
<td>50.2</td>
</tr>
</tbody>
</table>
Step 3: Fuzzy Training

- Training occurs offline
- For this RPM agent:
  - 600 training segments from 12 good recordings
  - Generated 382 fuzzy rules

<table>
<thead>
<tr>
<th>State</th>
<th>Max Fluctuation</th>
<th>XC DB22</th>
<th>XC DB33</th>
<th>Conca-vity</th>
<th>WA DB12</th>
<th>WA DB15</th>
<th>WA DB23</th>
<th>WE DB31</th>
<th>WE DB31</th>
<th>WE DB35</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 (fuzzy terms)</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>
Step 4: Fuzzy Inference

<table>
<thead>
<tr>
<th>Seg#</th>
<th>Rec...</th>
<th>Agent</th>
<th>S...</th>
<th>E...</th>
<th>Stale</th>
<th>Cut</th>
<th>Exp Out</th>
<th>Bel Good</th>
<th>Bel Bad</th>
<th>Dist Good</th>
<th>Dist Bad</th>
<th>Clust#</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>✔</td>
<td>Rains2: OUTPUT</td>
<td>0</td>
<td>34</td>
<td>3</td>
<td>2.00</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4.44e-002</td>
<td>3.09e-005</td>
<td>1.000</td>
</tr>
<tr>
<td>2</td>
<td>✗</td>
<td>Rains2: OUTPUT</td>
<td>34</td>
<td>51</td>
<td>4</td>
<td>1.000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3.21e-002</td>
<td>1.05e-002</td>
<td>1.000</td>
</tr>
<tr>
<td>3</td>
<td>✗</td>
<td>Rains2: OUTPUT</td>
<td>51</td>
<td>61</td>
<td>2</td>
<td>1.000</td>
<td>0</td>
<td>2.73e-002</td>
<td>0.945</td>
<td>9.14e-002</td>
<td>2.90e-005</td>
<td>1.000</td>
</tr>
<tr>
<td>4</td>
<td>✔</td>
<td>Rains2: OUTPUT</td>
<td>61</td>
<td>66</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0.270</td>
<td>0</td>
<td>5.71e-003</td>
<td>2.49e-007</td>
<td>NA</td>
</tr>
<tr>
<td>5</td>
<td>✔</td>
<td>Rains2: OUTPUT</td>
<td>66</td>
<td>101</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0.320</td>
<td>0</td>
<td>1.00e-002</td>
<td>3.39e-005</td>
<td>NA</td>
</tr>
<tr>
<td>6</td>
<td>✔</td>
<td>Rains2: OUTPUT</td>
<td>101</td>
<td>114</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0.250</td>
<td>0</td>
<td>4.89e-003</td>
<td>2.50e-007</td>
<td>NA</td>
</tr>
<tr>
<td>7</td>
<td>✗</td>
<td>Rains2: OUTPUT</td>
<td>114</td>
<td>131</td>
<td>3</td>
<td>2.000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2.41e-002</td>
<td>3.17e-005</td>
<td>2.00</td>
</tr>
<tr>
<td>8</td>
<td>✔</td>
<td>Rains2: OUTPUT</td>
<td>131</td>
<td>172</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0.379</td>
<td>0</td>
<td>4.61e-003</td>
<td>2.50e-007</td>
<td>NA</td>
</tr>
<tr>
<td>9</td>
<td>✔</td>
<td>Rains2: OUTPUT</td>
<td>172</td>
<td>294</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0.493</td>
<td>0</td>
<td>6.01e-003</td>
<td>2.49e-007</td>
<td>NA</td>
</tr>
<tr>
<td>10</td>
<td>✔</td>
<td>Rains2: OUTPUT</td>
<td>294</td>
<td>311</td>
<td>2</td>
<td>2.00</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>6.65e-002</td>
<td>2.50e-007</td>
<td>3.00</td>
</tr>
<tr>
<td>11</td>
<td>✔</td>
<td>Rains2: OUTPUT</td>
<td>311</td>
<td>318</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0.396</td>
<td>0.208</td>
<td>0.136</td>
<td>4.64e-002</td>
<td>NA</td>
</tr>
<tr>
<td>12</td>
<td>✗</td>
<td>Rains2: OUTPUT</td>
<td>313</td>
<td>327</td>
<td>2</td>
<td>1.000</td>
<td>0</td>
<td>0</td>
<td>1.000</td>
<td>0.172</td>
<td>0</td>
<td>3.00</td>
</tr>
<tr>
<td>13</td>
<td>✔</td>
<td>Rains2: OUTPUT</td>
<td>327</td>
<td>355</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0.243</td>
<td>0</td>
<td>1.87e-002</td>
<td>2.49e-007</td>
<td>NA</td>
</tr>
<tr>
<td>14</td>
<td>✔</td>
<td>Rains2: OUTPUT</td>
<td>355</td>
<td>423</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0.462</td>
<td>0</td>
<td>8.37e-003</td>
<td>2.89e-005</td>
<td>NA</td>
</tr>
</tbody>
</table>

?- unknown
X: bad
√: good
Step 5: Signal Level Fault Detection

The detection of a signal fault is calculated over one or more columns of this inference result table generated by the segment level fault detection system.

<table>
<thead>
<tr>
<th>Seg#</th>
<th>Recording</th>
<th>Agent</th>
<th>Out</th>
<th>Exp Out</th>
<th>Bel Good</th>
<th>Bel Bad</th>
<th>Dist Good</th>
<th>Dist Bad</th>
<th>Clust#</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Rains2</td>
<td>RPM...</td>
<td>2.00</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4.44e-002</td>
<td>3.09e-002</td>
<td>1.000</td>
</tr>
<tr>
<td>X</td>
<td>Rains2</td>
<td>RPM...</td>
<td>1.000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.11e-002</td>
<td>1.05e-002</td>
<td>1.000</td>
</tr>
<tr>
<td>X</td>
<td>Rains2</td>
<td>RPM...</td>
<td>1.000</td>
<td>0</td>
<td>2.73e-002</td>
<td>0.945</td>
<td>3.14e-002</td>
<td>2.90e-002</td>
<td>1.000</td>
</tr>
<tr>
<td>X</td>
<td>Rains2</td>
<td>RPM...</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td></td>
</tr>
<tr>
<td>X</td>
<td>Rains2</td>
<td>RPM...</td>
<td>0</td>
<td>0</td>
<td>0.320</td>
<td>0</td>
<td>1.00e-002</td>
<td>3.39e-002</td>
<td>NA</td>
</tr>
</tbody>
</table>

Details of Fuzzy Inference on Signal Segments

<table>
<thead>
<tr>
<th>Avg Good Belief</th>
<th>Avg Bad Belief</th>
<th>Avg Distance to Good Rules</th>
<th>Maximum Cluster Size</th>
<th>Number of Clusters</th>
<th>Percentage of Signal Labeled Bad</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 (fuzzy terms)</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>0.203</td>
<td>0.154</td>
<td>0.0717</td>
<td>5</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>25%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fuzzy Inference

Signal-KB was trained on selected good and bad signals

This signal is classified as ABNORMAL
Other RPM Case

- Another recording from same vehicle
- We can see a stall where RPM = 0
ECT Cases

- In general, single signal analysis works well for many simple faults.
# Multi-signal Agents

<table>
<thead>
<tr>
<th>IAC Multiple Signal Agent</th>
<th>MAF Multiple Signal Agent</th>
<th>RPM Multiple Signal Agent</th>
<th>TP Multiple Signal Agent</th>
</tr>
</thead>
<tbody>
<tr>
<td>IAC XC DB33</td>
<td>MAF Fluctuation</td>
<td>RPM Convexity</td>
<td>TP XC DB30</td>
</tr>
<tr>
<td>IAC XC DB34</td>
<td>MAF Max</td>
<td>RPM Avg</td>
<td>TP XC DB25</td>
</tr>
<tr>
<td>MAF Avg</td>
<td>MAF WE DB25</td>
<td>RPM WE DB35</td>
<td>TP CT Avg</td>
</tr>
<tr>
<td>MAF WE DB25</td>
<td>MAF XC DB33</td>
<td>FUEL Fluctuation</td>
<td>TP CT XC DB30</td>
</tr>
<tr>
<td>MAF XC DB23</td>
<td>TP Max</td>
<td>FUEL Min</td>
<td></td>
</tr>
<tr>
<td>TP Concavity</td>
<td>RPM Min</td>
<td>FUEL Max</td>
<td></td>
</tr>
<tr>
<td>TP Flucatuation</td>
<td>RPM Avg</td>
<td>FUEL Avg</td>
<td></td>
</tr>
<tr>
<td>TP WA DB15</td>
<td>RPM XC DB10</td>
<td>MAF Max</td>
<td></td>
</tr>
<tr>
<td>TP WA DB25</td>
<td>IAC Max</td>
<td>MAF Avg</td>
<td></td>
</tr>
<tr>
<td>TP WA DB35</td>
<td>IAC XC DB30</td>
<td>MAF WE DB24</td>
<td></td>
</tr>
<tr>
<td>TP XC DB35</td>
<td>IAC XC DB34</td>
<td>SPARK Fluctuation</td>
<td></td>
</tr>
<tr>
<td>TP XC DB34</td>
<td>IAC XC DB35</td>
<td>SPARK Max</td>
<td></td>
</tr>
</tbody>
</table>
Multiple Signal Cases: RPM Hesitation Improved

- Single signal agent results

- Multiple Signal Agent Results: RPM, MAF, FUELPW, SPARK
Multiple Signal Agent TP

Case 1: Rough Idle

Root Cause: Incorrect TP Sensor Reading. TPCT Changes and TP ≠ TPCT. Engine Thinks Throttle Is Part Open: Wrong Fuel Delivery, Rough Idle

- TP single signal agent did not find this fault
**Advantages of DDAS(I)**

- **Scalable:**
  - allows user to authorize new agents and train new agents
  - allows user to define new signals

- **Versatile:** allows user to submit signals relevant to his problems.
  - DDAS would activate only these agents that are relevant to the submitted vehicle case
  - DDAS could be trained from NORMAL DATA ONLY at the segment-level fault detection
Advantages of DDAS(II)

- Fuzzy knowledge base can be generated from either
  - training data automatically
  - expert knowledge interactively, or,
  - a combination of the two sources above

- Fuzzy learning and test system, FUZWIN, has advanced learning capability
  - Learning from good data samples only
  - Learning from noise data
  - Accumulative learning

- When a new type of problems requires research, ADSAS can quickly learn to solve them By
  - allowing user to author new agents and re-train agents
  - Allowing user to develop add new feature functions into the system

- Easy to develop new types of agents
  - Fault specific agents
ADSAS’s Data Visualization Capabilities

- Displaying multiple signals, analog and/or digital, simultaneously
- Displaying a fixed line at trigger point.
- Displaying a floating/moving vertical data line and the values of each signal at the floating vertical line
- Displaying signal segments and allowing engineers to mark segments as ABNORMAL or NORMAL.
- and more.
Applications of DDAS in OBD and Vehicle Prognosis

- **OBD**
  - Train important SDAs from well-known cases
  - Perform accumulative learning during OBD
    - Learn from new data and update its KB

- **Prognosis**
  - Develop SDAs for import components of vehicles
  - Monitor the changes of signals by detecting the change of signal behaviors
  - Use case based reasoning to detect fatigue of components
Acknowledgment

- This work has been supported by a number of research contracts from the Ford Motor Company and the National Science Foundation DMI-9612190.
Related Publications

- Jacob Crossman, Hong Guo, Yi Lu Murphey and John Cardillo, “Automotive Fault Diagnosis Part II A Distributed Agent Diagnostic System,” to appear in IEEE Transaction on Vehicular

