

# UMA ESTRATÉGIA DE FUSÃO DE ÁRVORES DE DIAGNÓSTICO PARA IDENTIFICAÇÃO DE FALHAS EM VEÍCULOS AUTOMOTORES

MARKOS F. B. G. OLIVEIRA, FERNANDO M. MORI, RICARDO LÜDERS

*DAINF/CPGEI, Universidade Tecnológica Federal do Paraná - UTFPR*

*Av. Sete de Setembro 3165, 80230-901 Curitiba-PR, Brasil*

*E-mails: markos.flavio@hotmail.com, fernandommori@gmail.com,  
luders@utfpr.edu.br*

**Abstract**— Due to a widespread using of microprocessors and a consequent increasing of the embedded systems complexity in vehicles, efficient fault diagnosis has become an important issue for reliability and reduction of the out of service time for maintenance. During vehicle utilization, several fault codes are stored when abnormal operation conditions are detected. The fault diagnosis discussed in this work is represented by a diagnostic tree for each fault code. Each node represents a test to be done in a tentative of failure cause identification. As several codes may be present simultaneously, this paper presents an initial strategy for integrating trees with active fault codes. This is done by isolating common tests and components in a single diagnostic tree which contributes for a time reduction of fault diagnosis. A computational tool for representing the considered trees is presented as well.

**Keywords**— Embedded intelligent systems, discrete event systems, fault diagnosis, decision trees, fault code.

**Resumo**— Devido à introdução de sistemas microprocessados e o consequente aumento na complexidade dos sistemas embarcados em veículos automotores, o diagnóstico eficiente de falhas torna-se importante para melhoria da confiabilidade e diminuição dos tempos de parada para manutenção. Durante a operação do veículo, vários códigos de falha são armazenados, representando situações anormais de funcionamento. O diagnóstico de falhas abordado neste trabalho é representado por uma árvore de diagnóstico para cada código de falha, tal que cada nó representa um teste a ser realizado na tentativa de se isolar a causa da falha. Como vários códigos de falha podem estar presentes simultaneamente, este trabalho apresenta uma estratégia inicial para integração das árvores com códigos ativos, isolando testes e componentes comuns com o objetivo de reduzir o tempo de diagnóstico. Uma ferramenta computacional para representação e cadastro das árvores de diagnóstico é também apresentada.

**Palavras-chave**— Sistemas inteligentes embarcados, sistemas a eventos discretos, diagnóstico de falhas, árvores de decisão, código de falha.

## 1 Introdução

De maneira geral, o diagnóstico de falhas pode ser genericamente definido como um conjunto de técnicas que, a partir de modelos de funcionamento normal e informações de anormalidades (efeitos observáveis da falha normalmente obtidas por sensores), identifica um ou mais componentes defeituosos (causas da falha). Em Avizienis (2004), os conceitos de falha, erro e defeito são precisamente definidos. Entretanto, optamos por designar genericamente uma falha como um código originado por um defeito, ou seja, por um componente ou subsistema defeituoso.

Existe na literatura uma ampla gama de aplicações e metodologias para diagnóstico de falhas. Em Struss and Price (2003) e Struss *et al* (2010) é apresentada uma abordagem de diagnóstico baseada em modelos, particularmente para a indústria automotiva. Segundo os autores, uma das dificuldades para a generalização das estratégias de diagnóstico está no fato de que os sistemas de diagnóstico são em geral projetados para casos particulares de domínios específicos. Os autores propõem então a criação de modelos que descrevem qualitativamente a interação entre os diversos componentes de um sistema. A partir deste modelo conceitual, uma estratégia de diagnóstico pode ser

construída ou mesmo reutilizada. Em Luo *et al* (2010) pode ser encontrada uma aplicação completa do diagnóstico baseado em modelo integrado com a aquisição de sinais de um sistema automotivo.

Existem ainda trabalhos cujo diagnóstico está baseado no registro de uma sequência particular de eventos observáveis que determinam a ocorrência de um evento não observável de falha, conforme Sampath *et al* (1995) e Sampath *et al* (1996). Estas sequências de eventos são modeladas em geral por autômatos de estados finitos e um ou mais eventos de falha. Em geral, esta abordagem é apropriada para o diagnóstico *on-line*, ou seja, que é realizado durante a operação do sistema.

Na indústria automotiva, um conjunto de códigos de falha são armazenados por unidades centrais de processamento durante a operação do veículo. O conjunto dos códigos de falha armazenados deve então ser analisado por especialistas com a ajuda de ferramentas de auxílio ao diagnóstico. Estas ferramentas computacionais estão disponíveis durante a parada do veículo em manutenções preventivas ou emergenciais, caracterizando um diagnóstico *off-board*. Tais ferramentas são normalmente projetadas segundo especificações de fabricantes de componentes ou durante a concepção do veículo e permitem analisar cada código de falha individualmente.

Este trabalho utiliza árvores de diagnóstico. Estas árvores determinam sequências de testes a serem executados por técnicos especializados de forma a excluir passo a passo as possíveis causas da falha até que se encontre o componente defeituoso. Portanto, são aplicadas em diagnósticos *off-board* e procuram não apenas identificar a ocorrência da falha, mas principalmente sua causa. Existem trabalhos dedicados à construção de árvores de diagnóstico através de diferentes metodologias (Assaf and Dugan, 2005; Majdara and Wakabayashi, 2009). Em Assaf and Dungan (2005) é apresentada uma metodologia baseada na análise de confiabilidade de árvores de falha dinâmicas (DFTs). Em Majdara and Wakabayashi (2009) é mostrado um método automático para a geração de árvores de falhas através da interação entre os componentes do sistema. Diferentemente destes trabalhos, o foco deste artigo não está na obtenção das árvores de diagnóstico, mas no seu uso integrado, uma vez que consideramos que as árvores individuais de diagnóstico já foram construídas, em geral por técnicos especializados.

A proposta é então fazer uso combinado dessas árvores quando vários códigos de falha estão presentes, pois aparentemente existe informação suficiente para reduzir o tempo de diagnóstico.

Para tanto, o artigo está organizado da seguinte forma. Na Seção 2 seguinte é descrito o problema de diagnóstico de falhas tratado neste artigo. A Seção 3 apresenta a metodologia utilizada, que se baseia nas árvores de diagnóstico para os diversos códigos de falha. Esta seção contém também a proposta deste trabalho e o protótipo de um software desenvolvido para auxílio ao cadastro das árvores. Os resultados são apresentados na Seção 4, através de um estudo de caso, seguida das conclusões na Seção 5.

## 2 Descrição do Problema

O problema de diagnóstico na indústria automotiva tem por objetivo responder de forma rápida e eficiente a uma situação de anormalidade de funcionamento do veículo, identificando precisamente o componente (ou componentes) defeituoso e assegurando um reparo adequado. Usualmente o diagnóstico é um processo custoso devido à complexidade crescente dos sistemas embarcados, à constante mudança no projeto dos veículos e diversidade dos componentes empregados, assim como à falta de mão-de-obra especializada.

O processo de diagnóstico envolve um produto, uma ferramenta de diagnóstico e um procedimento de reparo que pode ser entendido em cinco passos:

1. Monitoração das condições de funcionamento do veículo;
2. Identificação da falha;
3. Rastreamento da falha;
4. Reparo;

## 5. Verificação.

A monitoração do passo 1 acima se refere à monitoração de diversos componentes eletrônicos (sensores, atuadores, chaves ou relés) por uma ECU (*Electronic Central Unit*) que é capaz de processar estas informações, identificando e armazenando determinadas condições de falha. Isso é feito basicamente no passo 2 por um programa embarcado na ECU que codifica condições anormais de funcionamento em códigos de falha. Por exemplo, existem códigos de falha para condições elétricas anormais, tais como curto-circuito para fonte ou terra, ou circuito aberto, assim como erros de comunicação da rede CAN, apenas para citar alguns. Neste ponto, as falhas são codificadas de acordo com padrões internacionais, segundo as normas SAE J1939 (SAE International, 2012) e ISO14230-3 (ISO, 2012). Esta parte do diagnóstico é conhecida como diagnóstico *on-board*.

A partir dos códigos de falha, é possível identificar o componente (ou componentes) de falha no passo 3. Isso é feito por técnicos especializados guiados por uma ferramenta de diagnóstico *off-board*. Basicamente, esta ferramenta é resultado de uma compilação de informações que vão desde manuais técnicos dos componentes, dados de projeto do veículo e experiência de técnicos de pós-venda. Esta ferramenta guia o técnico através de testes previamente projetados para isolar o componente de falha. Cada código de falha dá origem a uma árvore de decisão, conhecida como árvore de falha ou de diagnóstico, onde cada nó corresponde a um teste específico. Por um procedimento de exclusão das causas da falha, chega-se ao componente defeituoso. Porém, existem casos onde o diagnóstico não é conclusivo, e o sucesso do diagnóstico (isolamento da causa da falha) depende da experiência do técnico.

Após a identificação da causa da falha, o componente ou sistema defeituoso deve ser reparado ou substituído no passo 4. Após o reparo, no passo 5 é feita uma verificação do veículo em funcionamento para se certificar de que a anomalia que originou o código de falha não está mais presente.

Este trabalho aborda os passos 2 e 3, pois parte de um conjunto de códigos de falha armazenados pela ECU e propõe uma estratégia de diagnóstico a partir da combinação das árvores de diagnóstico dos códigos de falha individuais, redefinindo a sequência de testes a serem executados. O diagnóstico eficiente de falhas torna-se importante para melhoria da confiabilidade (Berger, 2002) e diminuição dos tempos de parada para manutenção.

## 3 Metodologia

### 3.1 Árvores de Diagnóstico

Uma árvore de diagnóstico é uma árvore binária de decisão tal que cada nó representa um teste a ser realizado. O resultado deste teste pode resultar na

identificação do componente defeituoso (o que conclui o diagnóstico), no encadeamento de um novo teste ou numa não identificação da causa da falha. Para cada código de falha existe uma árvore de diagnóstico correspondente.

Uma árvore de diagnóstico é projetada a partir das características do produto a ser diagnosticado levando-se em conta:

1. O tipo de componente (sensores, atuadores, chaves, ECU) e sua probabilidade de falha;
2. A localização e acessibilidade do componente;
3. O tipo do teste: elétrico (curto-circuito, circuito aberto, resistência interna), mecânico (pressão, vazamento, acoplamento, ou químico (combustível, lubrificante, líquido refrigerante));
4. O ferramental necessário para o teste: por exemplo, aparelhos eletrônicos de medida, testes químicos, etc.

### 3.2 Proposta de Fusão das Árvores

Dadas as árvores de diagnóstico e vários códigos de falha ativos, a metodologia proposta sugere uma estratégia de fusão das árvores. Essa fusão baseia-se nos testes e componentes comuns presentes nas árvores originais. Isto pode ser entendido a partir da ideia de que um elevado número de códigos falhas ativos provavelmente indica a existência de causas comuns de falha que acionam mais de um código de falha, possivelmente em cascata. Desta forma, define-se uma prioridade para os testes comuns nas árvores de diagnóstico originais. Há ainda a possibilidade de que dois ou mais testes distintos estejam associados a um único componente comum. Logo este componente deve ser adicionado à sequência de testes da nova árvore.

Sugere-se então uma estratégia simples que identifica, em todas as árvores que possuem um código ativo (sinalizando uma falha), o número de ocorrências de testes comuns. Uma nova árvore é então criada com os testes de maior ocorrência ocupando níveis mais próximos da raiz. Como há possibilidade de que dois ou mais testes distintos estejam associados a um único componente comum, executa-se o procedimento acima considerando também os componentes comuns. Isso pode ser resumido em três passos:

1. Identificam-se as árvores com códigos de falha ativos;
2. Executa-se uma busca em árvore, retornando os testes e componentes em comum, assim como seus respectivos números de ocorrências;
3. Constrói-se a árvore resultante tal que os testes com maior número de ocorrências, ou que possuam um maior número de componentes comuns, ocupem nós mais próximos da raiz.

### 3.3 Software de Cadastro

De forma a padronizar a representação das árvores de diagnóstico utilizadas, um software em Java tem sido desenvolvido para o cadastro, navegação, visualização e composição dessas árvores. A estrutura das árvores sendo representadas em arquivos XML (*Extensible Markup Language*).

```
//Fault Code Reasoning Based on Diagnostic Trees
<FaultTrace id="FAULT_CODE">
  <Node id="TEST_001" Type="ELECTRICAL_TEST">
    <NodeStructure>
      <Position NodeVPos="0" NodeHPos="0"></Position>
      <Entry EntryId="Start"></Entry>
      <Exit Path="ELECTRICAL_TEST_002" Type="Pass"></Exit>
      <Exit Path=" " Type="Fail"></Exit>
    </NodeStructure>
    <Component id="X01" Caption="COMPONENT X01"></Component>
    <Measurement>
      <Tool Type="TOOL" Great="xx"></Tool>
      <ExpectedValue Central="5.0" Min="0" Max="10.0"></ExpectedValue>
      <Connector>
        <Conn1 state="enabled" connection="disconnected" ST_Id="xx">
          <ConnPin1 state="Enabled" ConnId="TOOL" Pin="1"></ConnPin1>
          <ConnPin2 state="Enabled" ConnId="TOOL" Pin="2"></ConnPin2>
        </Conn1>
      </Connector>
    </Measurement>
  </Node>
</FaultTrace>
<Node id="FID_001" Type="FAULT_IDENTIFIED">
  <NodeStructure>
    <Position NodeVPos="1" NodeHPos="0"></Position>
    <Entry EntryId="Start"></Entry>
    <Exit Path="ExitTree" Type="FID"></Exit>
    <Exit Path="ExitTree" Type="Error"></Exit>
  </NodeStructure>
  <Component id="X01" Caption="COMPONENT X01"></Component>
</Node>
```

Figura 1. Parte da estrutura da árvore de diagnóstico em XML.

Na Figura 1, tem-se uma imagem simplificada de parte da estrutura XML de uma árvore de diagnóstico construída pelo software de cadastro. A árvore é composta por estruturas independentes delimitadas por tags de início e fim que representam elementos do diagnóstico. Por exemplo, o tipo de nó, as características do teste ou a posição do nó de acordo com a posição hierárquica da tag na estrutura XML. Toda a estrutura e definição das tags, elementos e atributos foram feitas de forma a compor e representar de maneira robusta as árvores de diagnóstico. Na Figura 1, pode-se observar a primeira tag (FaultTrace id="FAULT\_CODE"), que é o elemento raiz do XML e representa o código de falha gerado. Mais abaixo, tem-se o que seria o primeiro nó da árvore, ou seja, um nó tipo "Teste Elétrico" (Id="TEST\_001" Type="ELECTRICAL\_TEST"). Informações como posição, componente de teste, informações de medição, entre outras, estão inseridas nesta subestrutura. Posteriormente, pode-se observar que há também uma tag do tipo "Falha Encontrada" (Id="FID\_001" Type="FAULT\_IDENTIFIED") que conclui o diagnóstico.

O subprograma relacionado a parte de comparação e criação da nova sequência de testes ainda está em desenvolvimento. Porém, o algoritmo relacionado ao cadastro e navegação das árvores já está finalizado. Na Figura 2, pode-se observar a tela

principal do software relacionado à navegação das árvores e criação dos nós. Entende-se por navegação uma visualização básica de cada nó da árvore e suas respectivas informações, associado a uma forma de percurso pelos nós da árvore.

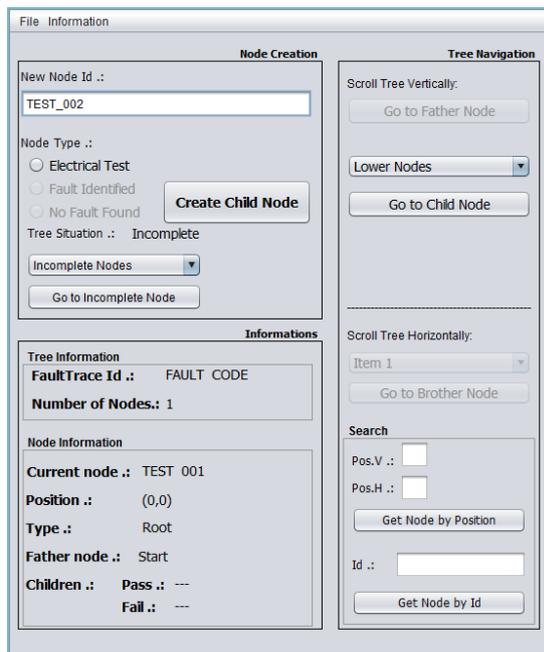


Figura 2. Painel principal do software de cadastro e visualização.

#### 4 Estudo de Caso

Esta seção ilustra a abordagem proposta a partir do diagnóstico do subsistema de monitoração das rodas de um veículo. Nesta caso, tem-se uma ECU que monitora quatro sensores, um para cada uma das rodas. A Figura 3 mostra o diagrama simplificado de conexões do subsistema. Caso ocorra uma falha, a ECU irá gerar o código de falha apropriado.

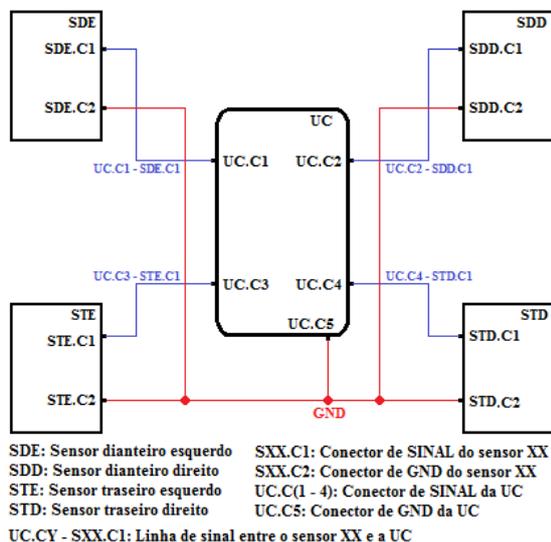


Figura 3. Diagrama de conexões do subsistema de monitoramento das rodas do veículo.

No esquema acima, consideram-se três tipos de falhas para cada um dos sensores. A falha de curto-circuito (CC) ocorre quando há uma leitura de 0V no pino de sinal da ECU. A falha de circuito aberto (CA) está relacionada a uma impedância infinita de entrada no pino de sinal da ECU. Já a falha de leitura inválida (LI) ocorre quando há uma leitura, limitada e não nula, fora da faixa de especificação ou operação do sensor no pino da ECU.

Para cada código de falha haverá uma árvore de diagnóstico específica, contendo determinados testes para a identificação do componente falho que gerou o código. Como exemplo, a Figura 4 ilustra as árvores de diagnóstico já aplicadas ao modelo automotivo da Figura 3, assim dispostas: a falha CA no sensor SDD (código CA\_SDD), a falha CC no sensor STE (código CC\_STE) e a falha LI no sensor STD (código LI\_STD).

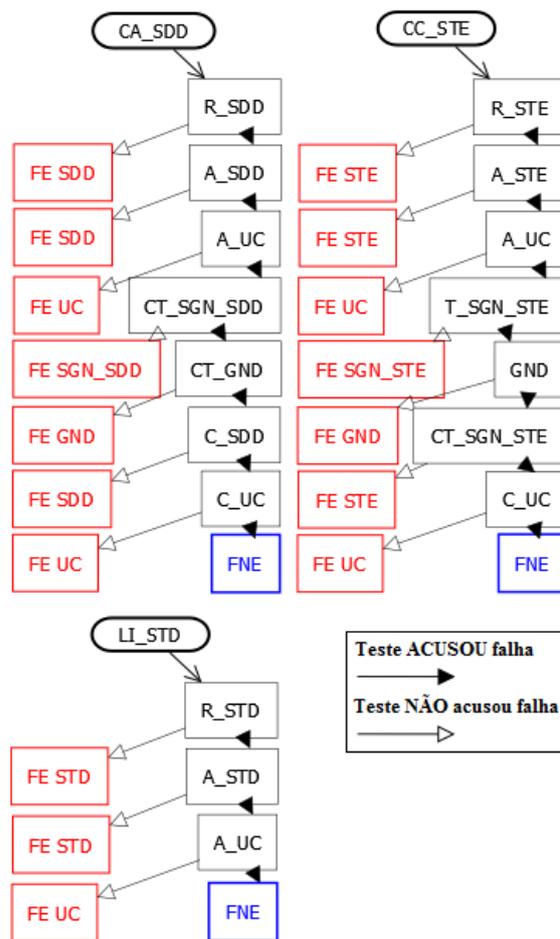


Figura 4. Árvores de Diagnóstico Ca\_SDD, CC\_STE, LI\_STD.

As árvores são compostas de testes sobre os componentes do sistema (representados pelos nós localizados à direita da árvore, os quais possuem um rótulo relativo a característica do teste); se o componente estiver OK, na sequência haverá outro teste, se não, o componente falho (representado pelos nós localizados mais à esquerda, com a abreviação 'FE' (Falha Encontrada) no rótulo) é a causa do

acionamento do código e deverá ser reparado. Caso todos os testes não acusarem falha, o nó 'FNE' (Falha Não Encontrada) é encontrado, sendo necessária uma análise mais peculiar do sistema. Mais abaixo, na Tabela 1, são apresentadas as descrições de cada teste que compõem as árvores. A descrição indica qual é o componente a ser testado (em **negrito**) e qual é o teste/verificação propriamente dito, entretanto o componente sobre o qual é realizado o teste e a identificação da falha já está implicitamente incluído no rótulo do mesmo.

Tabela 1. Descrição dos testes dos diagnósticos.

Teste	Descrição de verificação
A_UC	Alimentação/aterramento da UC
C_UC	Conectores da UC
A_SXX	Alimentação/aterramento do <b>sensor XX</b>
C_SXX	Conectores do <b>sensor XX</b>
R_SXX	Resistência interna do <b>sensor XX</b>
GND	Aterramento do sistema ( <b>GND = 0V?</b> )
CT_GND	Continuidade do <b>GND</b>
T_SNG_SXX	Tensão presente na <b>linha de sinal XX</b>
CT_SGN_SXX	Continuidade da <b>linha de sinal XX</b>

A partir das árvores da Figura 4 podemos verificar que apenas alguns componentes falhos podem ser identificados com a utilização dessas árvores. Essa limitação ocorre pois toda árvore de diagnóstico associada a um sistema possui uma certa resolução ou complexidade, que define quais falhas podem ser detectadas pelo diagnóstico e quais testes podem ser realizados para essa identificação. Para falhas em locais mais específicos são necessárias árvores mais complexas, com um maior número de testes.

Como mencionado anteriormente, a metodologia aqui apresentada é baseada nos casos onde mais de um código de falha ocorre simultaneamente. Desta forma, através de uma comparação entre testes e componentes comuns de árvores com códigos ativos, chega-se à construção de uma nova árvore com uma nova sequência de testes. Supondo por exemplo, que os códigos CA\_SDD e CC\_STE da Figura 4 estejam ativos, podemos construir uma nova árvore de diagnóstico baseada nos itens comuns (componentes ou testes) das árvores iniciais formando a árvore resultante da Figura 5. Os dois primeiros testes da nova árvore representam os testes comuns das árvores iniciais (teste A\_UC e do C\_UC. Os dois testes seguintes, apesar de não serem comuns nas árvores, possuem o mesmo componente provável de falha, no caso o GND. Neste caso, podemos afirmar que a construção da nova árvore leva em consideração tanto os testes em comum, como também os componentes prováveis de falha em comum das duas árvores.

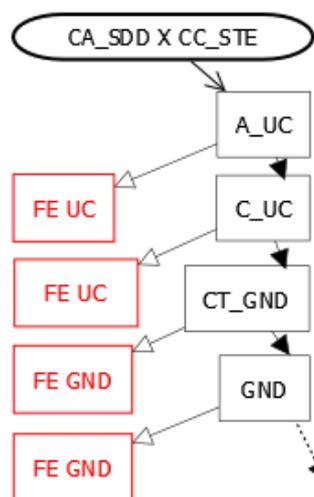


Figura 5. Árvore de diagnóstico Ca\_SDD X CC\_STE.

Apesar do estudo de caso apresentado possuir apenas dois testes em comum, a aplicação prevista da metodologia considera centenas de códigos de falha ativos. Este é o panorama típico de diagnóstico em veículos automotores.

## 5 Conclusão

A indústria automotiva dispõe de um grande volume de informações para diagnóstico de falhas em sistemas embarcados de veículos automotores. Muitas destas informações são disponibilizadas segundo padrões e normas pré-estabelecidas. Entretanto, devido à variedade e dinâmica do mercado de componentes e à frequente atualização dos projetos, o diagnóstico de falhas se pode tornar complexo e dispendioso. Para tanto, existem ferramentas de auxílio ao diagnóstico baseadas em árvores de diagnóstico para cada código de falha. Este trabalho apresenta uma estratégia de fusão das árvores de diagnóstico na presença de vários códigos de falha. O objetivo é reduzir o número e possivelmente a complexidade dos testes realizados para se isolar a causa da falha. A estratégia baseia-se na ideia de que muitos códigos de falha podem ser gerados por um conjunto reduzido de causas.

Futuramente, espera-se contar com um levantamento de dados reais de diagnóstico de campo. Isso possibilitará quantificar estatisticamente o ganho no tempo de diagnóstico pela aplicação da abordagem proposta em relação à utilizada atualmente.

## Agradecimentos

Ao Programa de Voluntariado em Iniciação Científica e Tecnológica da UTFPR.

## Referências Bibliográficas

- Assaf, T. and Dugan, J. B. (2005). Build Better Diagnostic Decision Trees. *IEEE Instrumentation & Measurement Magazine*, August, pp. 48- 53.
- Avizienis, A., Laprie J-C., Randell, B. and Landwehr, C. (2004). Basic concepts and taxonomy of dependable and secure computing. *IEEE Trans. on Dependable and Secure Computing*, Vol. 1, No.1, pp. 11-33.
- Berger, I. (2002). Can you trust your car? *IEEE Spectrum*, April, pp. 40-45.
- ISO (2012). ISO 14230-3:1999 Road Vehicles. [http://www.iso.org/iso/iso\\_catalogue/catalogue\\_tc/catalogue\\_detail.htm?csnumber=23921](http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=23921). Acesso em 02/06/2013.
- Luo, J., Namburu, M., Pattipati, K. R., Qiao, L. and Chigusa, S. (2010). Integrated model-based and data-driven diagnosis of automotive antilock braking systems. *IEEE Trans. on Systems, Man and Cybernetics – Part A: Systems and Humans*, Vol. 40, No. 2, pp. 321-336.
- Majdara, A. and Wakabayashi, T. (2009). A new approach for computer-aided fault tree generation. 3rd Annual IEEE International Systems Conf., Vancouver, pp. 1- 5.
- SAE International (2012). SAE J1939 Standards Collection on the Web: Content. <http://www.sae.org/standardsdev/groundvehicle/j1939a.htm>. Acesso em 02/06/2013.
- Sampath, M., Sengupta, R., Lafortune, S., Sinnamohideen, K. and D. Teneketzis, D. (1995). Diagnosability of discrete-event systems. *IEEE Trans. on Automatic Control*, vol. 40, pp. 1555–1575.
- Sampath, M., Sengupta, R., Lafortune, S., Sinnamohideen, K. and Teneketzis, D. (1996). Failure diagnosis using discrete-event models. *IEEE Trans. on Control Systems Technology*, Vol. 4, No. 2, pp. 105-124.
- Struss, P. and Price, C. (2003). Model-based systems in the automotive industry. *AI Magazine*, Winter, pp. 17-33.
- Struss, P., Provan, G., Kleer, J. and Biswas, G. (2010). Special issue on model-based diagnostics. *IEEE Trans. on Systems, Man and Cybernetics – Part A: Systems and Humans*, Vol. 40, No. 5, pp. 870-873.