

Diagnosis of an Alternator System Using Quantized Approaches

Sara Mohon¹ and Pierluigi Pisu²

^{1,2}*Clemson University International Center for Automotive Research, Greenville, SC, 29607, USA*

smohon@clemson.edu

pisup@clemson.edu

ABSTRACT

In this paper, the Generalized Cell Mapping (GCM) method for a linear system is compared with a new stochastic method for novel cell-to-cell mapping. The authors presented the new stochastic method in Mohon and Pisu (2013). The two methods are compared in an application example of a vehicle alternator. The alternator may experience three faults including belt slippage, a faulty diode connection, or incorrect controller reference voltage. Fault detection and isolation (FDI) is performed using the two cell-to-cell mapping methods. The results show that the new stochastic method is slower but yields better isolation results than the GCM method.

1. INTRODUCTION

Besides high performance, the other most important and desirable features of modern technological systems are safety and reliability. Owing to their increasing complexity, technological systems are becoming more and more vulnerable to faults. These faults, if not handled timely and properly, may lead to severe failures causing damage to property or even human lives. This is particularly true for the complex dynamic systems made of interconnected components where one faulty component can lead to malfunction of the overall system. Therefore, detection and isolation of the faults is of extreme importance in modern technological systems. Early detection and proper handling of faults essentially improve the dependability of the dynamic system ensuring safe operation.

An important tool for analyzing dynamic systems is cell-to-cell mapping as described by Hsu (1980). The dynamic state space of the system is quantized into cells that the system may occupy as time evolves. State variables are considered in intervals instead of a continuum of points. Such a system is justified due to the inherent inaccuracy of physical measurements. Using this framework, the probability of cell transitions can be computed using various approaches such as Monte Carlo, GCM, and Hyperbox-Mapping methods.

In the Monte Carlo method, repeated random samplings and deterministic computations are used to find possible outcomes and their associated probabilities (Kastner, 2010). Using this information, a state to state probability transition matrix for the system can be constructed (Wang, 1999). The more samplings performed, the more accurate the probability transition matrix (Sobol, 1994).

In the GCM method, the boundaries of image cells are important in determining state transition probabilities (Hsu, 1981). The image cell of the current cell are found first. Then the boundaries of the image cell are mapped back to locations on the current cell and when linearly connected form an area within the current cell. Now this area is known to transition to a particular image cell area. The probability associated with this transition is calculated given the total area of the current cell.

A similar approach to the GCM method is the Hyperbox-Mapping method. This method is also concerned with area transformations to obtain state transition probabilities especially for use with stochastic automata. In this approach, the cells are renamed hyperboxes. The method uses mappings between current state-input and future state-output plots to define areas and the measure of these areas produces probabilities. Higher accuracy for probabilities can be achieved by further sub-dividing the hyperboxes into smaller hyperboxes. A comparison between Monte Carlo method and Hyperbox-Mapping method shows the Monte Carlo method is an under-estimate of the true transition probabilities while the Hyperbox-Mapping method is an over-estimate of the true transition probabilities (Schröder, 2003).

The main motivation for formulating the GCM method was to analyze global dynamics of a system (Hsu, 1982). The purpose of the method was to find equilibrium states and periodic motions in the system that can be identified after many mapping steps are performed (Hsu & Chiu, 1986) (Chiu & Hsu, 1986). This global analysis can yield a stationary probability transition matrix that does not change with time. Stationary transition matrices allow the global behavior of the system to be analyzed through Markov Chain theory where the entire evolution of cell mapping

over time is determined by the stationary transition matrix (Hsu & Guttalu, 1980).

The Monte Carlo and GCM method each rely on repetitive simulations during each time step to calculate transition probabilities. Each method effectively uses information about the initial cell and image cell(s). The amount of computation involved could overwhelm a microcomputer trying to calculate transition probabilities in real-time. These methods are most suitable for offline approaches. Therefore, a new method that only uses information about the initial cell would be a beneficial step toward real-time applications.

The Monte Carlo and GCM approaches can also be computationally burdensome with respect to high dimensional nonlinear systems. Performing the Monte Carlo method on these systems requires huge sampling populations. The GCM method also requires many calculations in order to find image cell boundaries for a nonlinear system. Then all these image cell points must be inversely mapped into the original cell. The feasibility of these methods with nonlinear systems is severely limited.

The new stochastic method proposed by the authors uses the system vector field in quantized regions to calculate state transition probabilities as time evolves without computing image cells. In this paper, the new method will be called the Flow method. The flow in/out of a cell through its perimeter is analyzed similar to the Divergence Theorem. The total flow through a cell is comprised of summation of the flow through the sides of the cell. This flow directly impacts the probability of state transition. At each time step, the flow through each side of current state is calculated and then normalized to total flow through whole state perimeter. A time-varying probability transition matrix can be created from these calculations.

Once armed with the above methods for obtaining the probability transition matrices, they can be applied to FDI problems in quantized systems. For example, if an expected state transition has a very low probability, and then the state transitions to this state and possibly continues to transition to low probability states, then this could indicate a fault in the system. The probability transition matrices obtained from the above methods can also be used in conjunction with stochastic automata for quantized systems when the system output and system states are not identical (Schröder, 2003). This paper applies and compares the GCM and Flow methods for fault detection in an alternator system previously described by Mohon and Pisu (2013). Results show that the GCM method yields faster detection time with incomplete isolation of faults. On the other hand, the new stochastic method results in slower detection time and complete isolation.

The rest of the paper structure is as follows. The second section of this paper describes the GCM method. The third

section describes the Flow method. The fourth section describes the computational complexity of the Flow method and GCM method. The fifth section applies the two methods to an application example with a faulty automotive alternator and compares FDI results. Lastly, some concluding remarks about the usefulness of each method is provided.

2. GENERALIZED CELL MAPPING METHOD

The method for generalized cell mapping is described in a book by Hsu, Marsden, and Sirovich (1987). Unlike simple cell mapping, where one cell is mapped into a single image cell, generalized cell mapping allows one cell to be mapped to several image cells. Each image cells represents a fraction of the total probability.

Consider the following simple example. Suppose we have a system described by Eq 1. There are two states z_1 and z_2 and only z_2 is observable in output. We can illustrate the state space divided into quantized states 1 through 7 in Figure 1. We will also assume some maximum and minimum values for z_1 .

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \quad (1)$$

$$y = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}$$

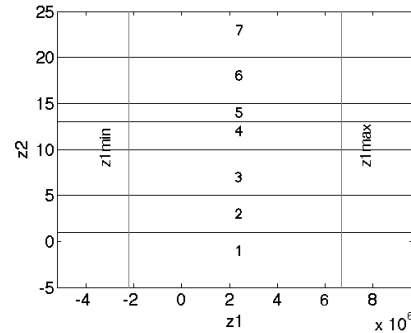


Figure 1: Quantized states in state space example

Obtaining the image cell boundaries can be thought of as a Monte Carlo exercise. By randomly choosing a large sample of random points within the initial cell (state 4) and applying the dynamic system equations, the new location of the points can be plotted on the 2D state space. Figure 2 and Figure 3 illustrate how the randomly sampled points move in time. A large number of points will clearly delineate the boundary of the new image cell.

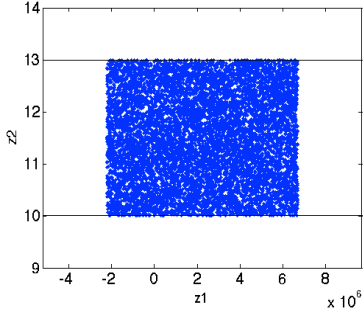


Figure 2: Initial cell with randomly sampled points

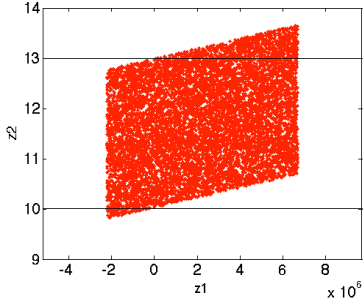


Figure 3: Image cell containing new position of sampled points after a finite time delta t

The new image cell in this example is clearly a quadrilateral with four vertices. These vertices represent the boundaries of the image cell. Note that the image cell is now spanning states 3, 4, and 5. By using the system dynamic equations, these vertices and other important points can be mapped back into the original cell shown in Figure 4. This will allow us to determine the regions of the original state that map into other states.

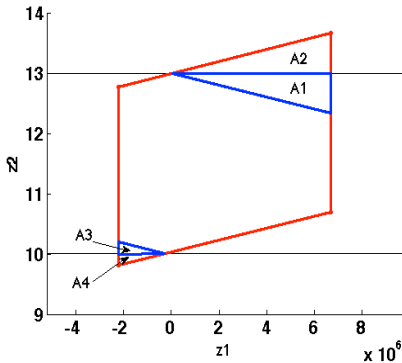


Figure 4: Inverse mapping important points on image cell back into original cell

The regions of the area now defined in the original cell can be used to calculate probabilities of transitioning up or down in the system. Region A2 is mapped back into region A1. Region A4 is mapped back into region A3. The probability to transition up, down, or stay in state 4 is given by the following.

$$\begin{aligned} \Pr(up) &= A1/A_{total} \\ \Pr(down) &= A3/A_{total} \\ \Pr(stay) &= 1 - \Pr(up) - \Pr(down) \end{aligned} \quad (2)$$

This process can be repeated as the system's state changes along with input values.

3. PROPOSED FLOW METHOD

The Flow method was proposed by the authors in a previous paper (2013). This method uses the system's vector field \vec{F} to determine flow into and out of the current state/cell. The method exploits the divergence theorem and determines the total potential of flow through the cell as the sum of flows through the perimeter of the cell.

A two-dimensional form of the divergence theorem is defined in Eq. (3). We define C as a closed curve, A as the 2D region in the plane enclosed by C , \vec{n} as the outward pointing normal vector of the closed curve C , and \vec{F} as a continuously differentiable vector field in region A . A graph of the 2D divergence theorem for the same 2D system in Eq. 1 is shown in Figure 5.

$$\iint_A (\nabla \cdot \vec{F}) dA = \int_C (\vec{F} \cdot \vec{n}) dr \quad (3)$$

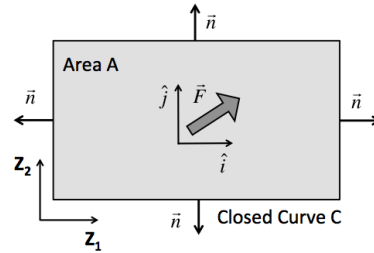


Figure 5. Graph of 2D Divergence Theorem for 2D state space system

We consider that the vector field \vec{F} describes transition flow in and out of the current state along the state boundaries. For the DC electric machine model, \vec{F} is defined as Eq. (4) where \hat{i} and \hat{j} are coordinates of vector field \vec{F} and functions f_1 and f_2 are defined by states z_1 and z_2 from the state space model in Eq. (1).

$$\begin{aligned} \vec{F} &= f_1 \hat{i} + f_2 \hat{j} \\ \dot{z}_1 &= f_1(z_1, z_2, u_1, u_2, u_3) \\ \dot{z}_2 &= f_2(z_1, z_2, u_1, u_2, u_3) \end{aligned} \quad (4)$$

The flow through the left and right sides of the area A in Figure 5 will be assumed zero for the alternator system shown in Figure 6. The line integrals along the state z boundaries will determine flow in and out of the state. The vector field \vec{F} is illustrated by grey slope field in Figure 6.

Flow out of state z is defined as a positive value φ^+ and flow into state z is a negative value φ^- . Since each side may have flow in and flow out sections, the flow transition point z^{**} or z^* is found if necessary and the appropriate limits of integration for flow in and flow out are integrated for each side. Transition points are shown in Figure 6. Without loss of generality assume $f_2 < 0$ if $z_1 < z^*, z^{**}$ and $f_2 > 0$ if $z_1 > z^*, z^{**}$ such that Eq. (5) holds. The upward and downward flow through each side of state z is given by Eq. (6).

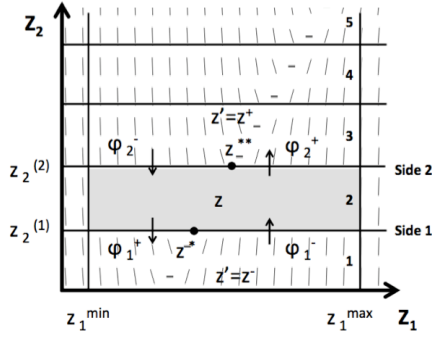


Figure 6. Graph of quantized DC electric machine system with flow definitions

$$\begin{aligned} f_2(z^*, z_2^{(1)}, u_1, u_2, u_3) &= 0 \\ f_2(z^{**}, z_2^{(2)}, u_1, u_2, u_3) &= 0 \end{aligned} \quad (5)$$

$$\begin{aligned} \varphi_1^+ &= - \int_{z_1^{\min}}^{z^*} f_2(z_1, z_2^{(1)}, u_1, u_2, u_3) dz_1 > 0 \\ \varphi_1^- &= - \int_{z^*}^{z_1^{\max}} f_2(z_1, z_2^{(1)}, u_1, u_2, u_3) dz_1 < 0 \\ \varphi_2^- &= \int_{z_1^{\min}}^{z^{**}} f_2(z_1, z_2^{(2)}, u_1, u_2, u_3) dz_1 < 0 \\ \varphi_2^+ &= \int_{z^{**}}^{z_1^{\max}} f_2(z_1, z_2^{(2)}, u_1, u_2, u_3) dz_1 > 0 \end{aligned} \quad (6)$$

Next we define φ_{in} , φ_{out} , and φ_{total} in Eq. (7) in order to build probabilities. The sum of the absolute value of all inward flow is defined as φ_{in} . The sum of all outward flow is defined as φ_{out} . The total flow φ_{total} is the sum of φ_{in} and φ_{out} .

$$\begin{aligned} \varphi_{in} &= |\varphi_1^- + \varphi_2^-| \\ \varphi_{out} &= \varphi_1^+ + \varphi_2^+ \\ \varphi_{total} &= \varphi_1^+ + |\varphi_1^-| + |\varphi_2^-| + \varphi_2^+ \end{aligned} \quad (7)$$

The notion of probability can be interpreted as counting types of occurrences and then normalizing the count of each

type by the total occurrences. Suppose the occurrences of outward and inward flow defined in Eq. (6) are normalized by the total flow defined in Eq. (7). For example, the probability to transition up will be defined as the outward flow through side 2, φ_2^+ , divided by the total flow φ_{total} . We can then define z^+ as the state above current state z and define z^- as the state below current state z . Equation (8) gives the probability to stay within the current state and the probability to transition up or transition down to an adjacent state. Uniform probability distribution is assumed along the borders of each state.

$$\begin{aligned} 1 &= \frac{\varphi_{in}}{\varphi_{total}} + \frac{\varphi_{out}}{\varphi_{total}} \\ 1 &= \frac{|\varphi_1^- + \varphi_2^-|}{\varphi_{total}} + \frac{\varphi_2^+}{\varphi_{total}} + \frac{\varphi_1^+}{\varphi_{total}} \\ 1 &= \Pr(z' = z | z) + \Pr(z' = z^+ | z) \\ &\quad + \Pr(z' = z^- | z) \\ 1 &= \Pr(stay) + \Pr(up) + \Pr(down) \end{aligned} \quad (8)$$

At each time step the probability to stay or transition up or transition down is calculated using the current state boundaries and the current input. This information builds a time-varying probability transition matrix named L that can be constructed as shown in Table 1 for the example of current state $z=2$ at time t .

		Future State z'			
		1	2	3	4
Current State z	1	0	0	0	0
	2	$\Pr(z' = z^- z)$	$\Pr(z' = z z)$	$\Pr(z' = z^+ z)$	0
	3	0	0	0	0
	4	0	0	0	0

Table 1: Example of probability transition matrix L for current state $z=2$ at a time t

4. COMPUTATIONAL COMPLEXITY

So far, the new stochastic method has been described in the context of a two dimensional state space system. The beauty of the new method is that it is based on the Divergence Theorem that already has a multi-dimensional framework. This means the new method can be easily extended to more dimensions and states using existing Divergence Theorem framework. In this section the multi-dimensionality of the new Flow method is investigated in terms of computational complexity and compared to the GCM method. Parameters of interest will include the number of states n in the system and the number of

sampling points on a quantized edge N . First, the computational complexity of the proposed Flow method is presented then followed by the GCM method.

4.1. Flow Method

Consider a system of $\dot{x} = f(x, u)$ such that $x \in R^n$ and N is the number of sampling points on a quantized edge on a state and n is the number of states. The calculation of the Flow method integral must be performed on each edge/face of a hypercube. The number of edges/faces F on a hypercube are a function of states n in the system.

$$F = 2n \quad (9)$$

Assume that the computational complexity for calculating elementary operations of $\dot{x} = f(x, u)$ is α . Also assume that the calculation of flow using divergence theorem can be approximated using the Euler method in Eq. (10).

$$\int_{x_1} \dots \int_{x_n} f_j(x, u) dt \approx \sum_{l_1=1}^{euler} \dots \sum_{l_n=1}^{no l_j} f_j(x_1^{l_1}, x_1^{l_2}, x_1^*, x_1^{l_n}, u) \Delta t \quad (10)$$

Therefore, numerically calculating the Flow method integral is $N^{n-1}\alpha$. The computational complexity for performing the Flow method on a n state system with N sampling points on the boundary of each state is given by

$$C = (2n)(N^{n-1})(\alpha) \quad (11)$$

4.2. GCM Method

The first step in the GCM method is obtaining the image cell boundaries and this can be achieved through the Monte Carlo method. The following analysis will show the computational complexity of the Monte Carlo method and how this first step of GCM method is already more computationally complex than the Flow method.

Now consider a system of x states such that $x \in R^M$ and N is the number of sampling points on a quantized edge on a state and n is the number of states. Again, α is assumed to be the computational complexity for calculating one state. Performing Monte Carlo on this system requires choosing initial points $x(t)$ and seeing where they evolve after some time to yield $x(t+\Delta t)$.

$$x(t + \Delta t) = x(t) + \dot{x}(t) * \Delta t \quad (12)$$

This equation requires another addition and multiplication to the computational complexity after calculating $\dot{x}(t)$ (which is essentially α), hence $\alpha+2$ computations are needed. This must also be performed inside all hypercubes in the n dimensional system with N sampling points on the boundary. Hence,

$$C = N^n(\alpha + 2) \quad (13)$$

4.3. Comparison of Computational Complexity

To illustrate the computational savings of the new Flow method compared to the Monte Carlo method, a 3 x 3 plot is given in Figure 7. Each row is number of sampling points N on state boundaries and each column is numbers of states n . For this figure, $\alpha=6$, $m=3$, $M=100$. The y axis in each plot is in log scale. Clearly the Flow method is less computationally burdensome than the GCM method.

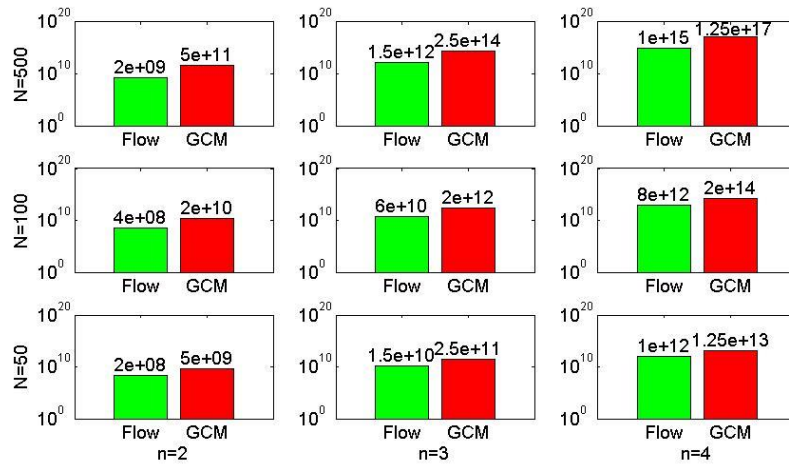


Figure 7: Comparison of computational complexity for different methods

5. APPLICATION EXAMPLE: ALTERNATOR SYSTEM

Today's vehicles require higher electrical demands than ever before due to more mandated safety technology and popular consumer technology integrated within the vehicle. The purpose of the vehicle's alternator system, also known as the electrical power generation storage (EPGS) system, is to maintain the necessary electrical power needed to start the vehicle and keep it running smoothly. A healthy EPGS system is crucial for proper operation of a vehicle and have been investigated in previous literature.

Scacchioli, Rizzoni, and Pisu (2006) proposed a fault isolation approach for an EPGS system using two equivalent alternator models. One equivalent model for a healthy alternator and one equivalent model for an alternator with one broken diode. Parity equations and three residuals with constant thresholds were used for fault isolation. The approach assumed a 3000 second Federal Urban Driving Schedule (FUDS) cycle.

Zhang, Uliyar, Farfan-Ramos, Zhang, and Salman (2010) proposed a fault isolation approach for an EPGS system using parity relations trained by Principal Component Analysis (PCA). Three residuals with constant thresholds were used for isolation. The approach assumed a staircase profile for both load current and alternator speed input, which is not a realistic scenario.

Hashemi and Pisu (2011a) proposed a fault isolation approach for an EPGS system using two observers and three residuals. The approach assumed a staircase profile for load current and a portion of the FUDS cycle for alternator speed. Adaptive thresholds were used for isolation. In other similar work, Hashemi and Pisu (2011b) showed the same approach but created a reduced order adaptive threshold model using Gaussian fit of data. The second approach was less computationally intensive.

Scacchioli, Rizzoni, Salman, Onori, and Zhang (2013) proposed a fault isolation approach for an EPGS system using one equivalent EPGS model that used parity equations to produce three residuals for fault isolation. The approach used a staircase profile for both load current and alternator speed input.

As stated, previous work for fault isolation in an EPGS system has included observers and parity relations. The approaches with observers were built for linear systems that approximate the nonlinear behavior of the EPGS system. These approaches cannot be extended for direct use on the nonlinear system itself. At least three residuals are required for all previous approaches. It is also concerning that some approaches were not validated using real driving situations. Therefore these approaches have limited scopes.

5.1. Model for Alternator System

This paper analyzes the EPGS system shown in Figure 8 as modeled by Scacchioli et al. (2006). It consists of a voltage controller, alternator, and battery. The controller can be an electronic control unit or a voltage controller on the alternator itself. In this paper, the controller is a part of the alternator to regulate field voltage. The alternator model consists of an AC synchronous generator, three phase full bridge diode rectifier, voltage controller, and excitation field.

The engine crankshaft mechanically spins the generator's rotor by use of a belt and pulley. The rotor is a ferrous metal wrapped with a single conductive winding. When the controller applies a small field voltage to the winding, a small field current flows through the winding. The flow of current through the winding produces a magnetic rotor with a north and south pole. However, the stator is composed of three phase stationary windings. As the magnetic rotor moves relative to the conductive stator windings, an electromotive force is induced in the stator windings. If the stator windings are connected to an electrical load, then AC current will flow in each of the three stator windings. The three currents are sent to a diode bridge rectifier to produce DC current for electrical loads or for recharging the battery. Therefore, the alternator takes mechanical energy of the engine and produces electrical energy for the battery or loads of the vehicle.

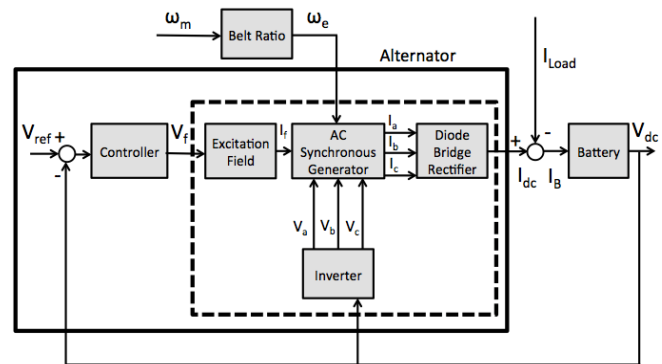


Figure 8. EPGS model

The model for the EPGS system results in a complex nonlinear system but can be more easily modeled by an equivalent DC electric machine as described by Scacchioli et al. (2006). The dashed line in Figure 8 encompasses the components represented by the DC model.

The DC electric machine is modeled by the state space system in Eq. (14) as shown by Hashemi (2011).

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \end{bmatrix} = \begin{bmatrix} 0 & a_{12}(\omega_e) \\ 1 & a_{22}(\omega_e) \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} + \begin{bmatrix} b_{11}(\omega_e) & b_{12}(\omega_e) & b_{13}(\omega_e) \\ 0 & b_{22}(\omega_e) & b_{23}(\omega_e) \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \quad (14)$$

$$y = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}$$

Equation (14) has two states z_1 and z_2 and inputs u_1 , u_2 , and u_3 . The system inputs represent the alternator field voltage V_f , angular frequency of alternator ω_e , and dc voltage of the battery V_{dc} also shown in Eq. (15). The coefficients a_{12} , a_{22} and $b_{11} \dots b_{23}$ are functions of engine speed and were found using system identification by Hashemi (2011) using test data at different constant engine speeds. In this model, state z_2 is the measurable quantity I_{dc} which is the rectified output current of the alternator.

$$\begin{aligned} y_2 &= I_{dc} = z_2 \\ u_1 &= V_f \\ u_2 &= \omega_e \\ u_3 &= V_{dc} \end{aligned} \quad (15)$$

5.2. Possible Faults in Alternator System

The EPGS system is important in every vehicle and faults in the system need to be detected and isolated as quickly as possible to prevent costlier damage. This paper considers three common faults that occur in an EPGS system. Possible fault locations in EPGS system are bolded in Figure 9.

1. **Voltage controller fault:** This fault occurs when the reference voltage V_{ref} is incorrectly raised or lowered by a percentage of the nominal V_{ref} . The fault can cause the alternator to overcharge or undercharge the battery.
2. **Diode rectifier fault.** This fault occurs when a diode in the diode bridge has a faulty connection. The fault results in a large ripple in battery voltage V_{dc} and alternator output current I_{dc} thereby decreasing the efficiency of alternator output.
3. **Belt slip fault.** This input fault occurs when the belt between the engine crankshaft and alternator pulley slips due to insufficient tension. The belt slip causes a decrease in alternator rotational speed ω_e and a decrease in alternator output voltage. To compensate, the voltage controller increases the field voltage and/or the battery must discharge more often to meet load demand. This can age the battery prematurely. Belt slip can signify the belt is worn and needs to be replaced.

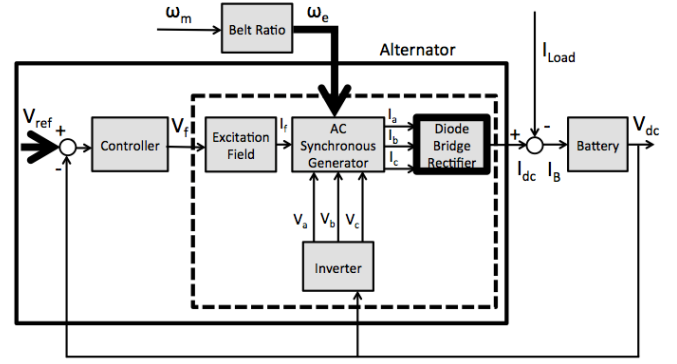


Figure 9: Possible faults in EPGS model

5.3. Simulation Results

Previous work by Scacchioli et al. (2006) yielded a complete nonlinear EPGS model. This nonlinear model uses ω_e , I_{load} , and V_{ref} as inputs and yields V_f , V_{dc} , and battery dc current I_{dc} as output. Diagnostics for the belt fault case, diode fault case, and voltage controller fault case are accomplished by using the Flow method and GCM method. The Flow method procedure is illustrated in Figure 10 and the GCM method procedure is illustrated in Figure 11.

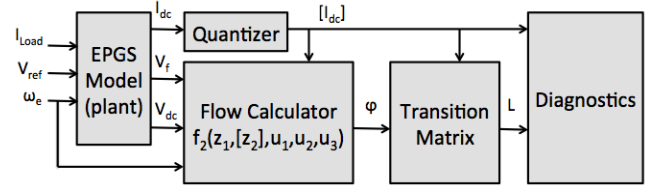


Figure 10: EPGS model with Flow method

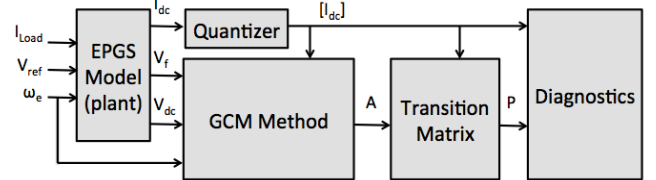


Figure 11: EPGS model with GCM method

The inputs for the nonlinear EPGS Simulink model are provided in Mohon and Pisu (2013).

Table 2 details the selected injection time and magnitude of fault relative to nominal that were injected during simulation. Depending on the fault cast (i.e. persistent fault, intermittent fault) the time of fault injection will change. The faults were simulated by modifying the nominal inputs of the system.

Fault	Injection time (s)	Modified Input	Resulting % drop with respect to nominal
Belt Slip	Varies	ω_e	80
Diode	Varies	V_{dc}	N/A (faulty connection)
Voltage Controller	Varies	V_{ref}	30

Table 2: Fault injection time and magnitude

Output z_2 range for nominal and faulty cases must be quantized into rectangles to find the probability transition matrix over time. Output z_2 is quantized into 12 states with names 1-12. The same boundaries and names will be used for faulty cases as well.

Given the z_1 range, the quantized states, and u_1 , u_2 , and u_3 , the probability transition matrix can now be calculated using the f_2 function from Eq. (14).

The probability transition matrix L contains probabilities of certain future state transitions as shown in Table 1. The largest probability will denote the most likely predicted state transition $z' = [I_{dc, predicted}]$ and its probability $P(z' = [I_{dc, predicted}])$. The most likely probability and most likely predicted state can be compared with the quantized output state $[I_{dc}]$ that is measured. If there is a relatively high probability of a particular state transition occurring and that state transition does not occur, then a fault may be present. An example of predicted state probabilities, predicted states, and output states over time for belt fault case is shown in Figure 12 and Figure 13.

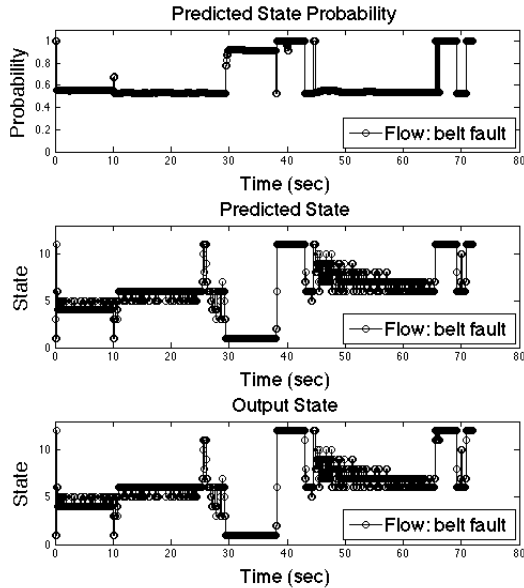


Figure 12. Belt fault outputs for Flow method

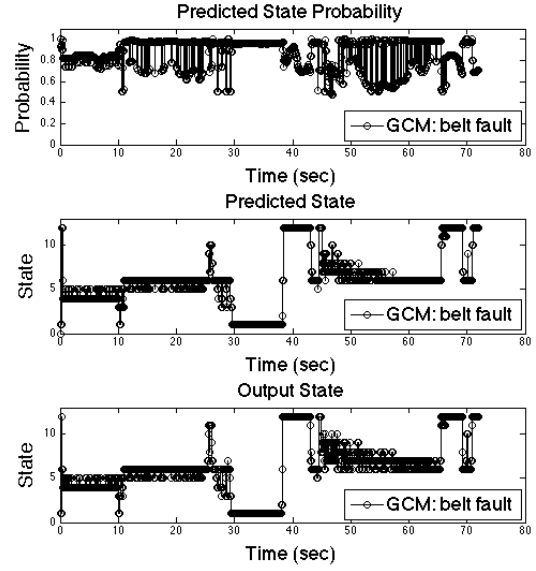


Figure 13: Belt fault outputs for GCM method

5.3.1. Persistent Faults and Threshold Analysis

Table 3 details the selected injection time and magnitude of fault relative to nominal that were injected during simulation. Once the fault is injected, it remains for the duration of the simulation. The faults are simulated by modifying the nominal inputs of the system.

Fault	Injection time (s)	Modified Input	Resulting % drop with respect to nominal
Belt Slip	10	ω_e	80
Diode	10	V_{dc}	N/A (faulty connection)
Voltage Controller	10	V_{ref}	30

Table 3: Persistent fault injection time and magnitude

Disagreement between predicted and output states are clear after calculating the difference of quantized output state $[I_{dc}]$ and the predicted state. This difference is defined as the residual r in Eq. (16). The residual results for each fault case using Flow method are shown in Figure 14 through Figure 16. The residual results for each fault case using GCM method are shown in Figure 17 through Figure 19.

$$r = [I_{dc}] - [I_{dc, predicted}] \quad (16)$$

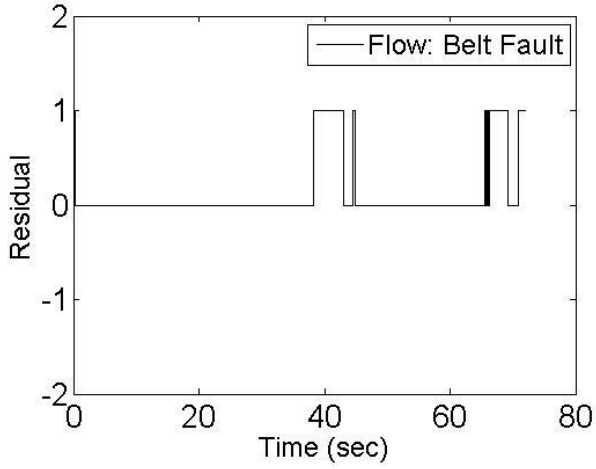


Figure 14. Belt fault residual for Flow method

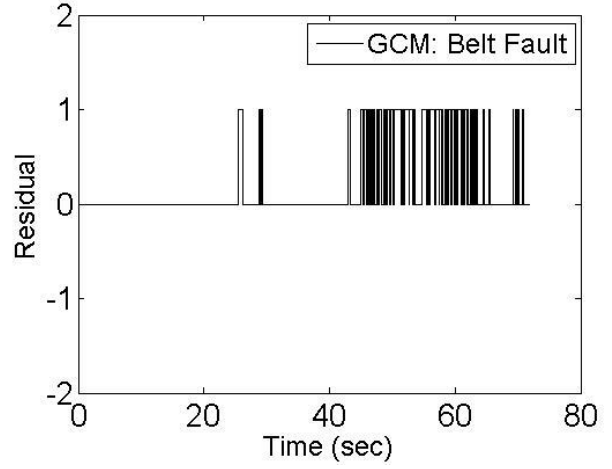


Figure 17: Belt fault residual for GCM method

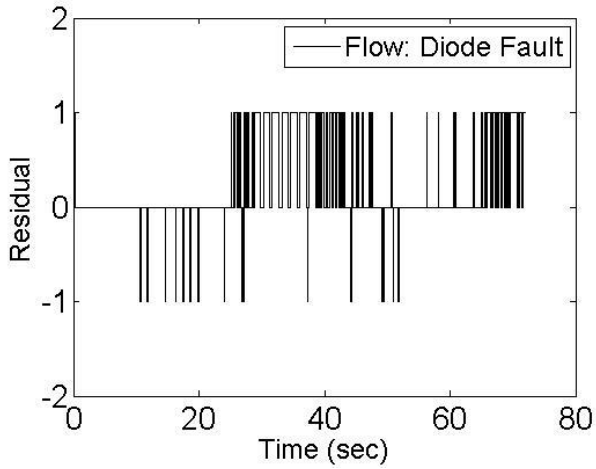


Figure 15. Diode fault residual for Flow method

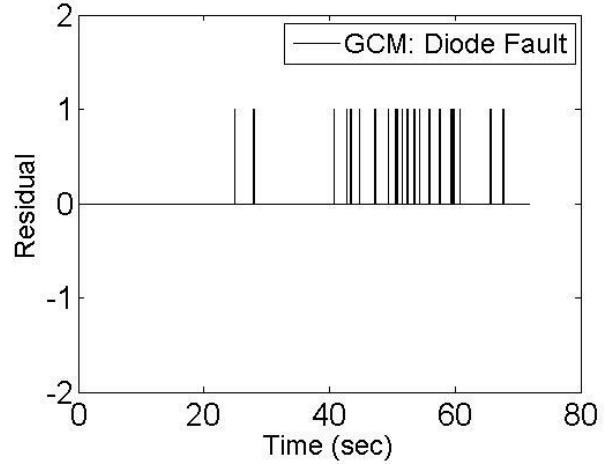


Figure 18: Diode fault residual for GCM method

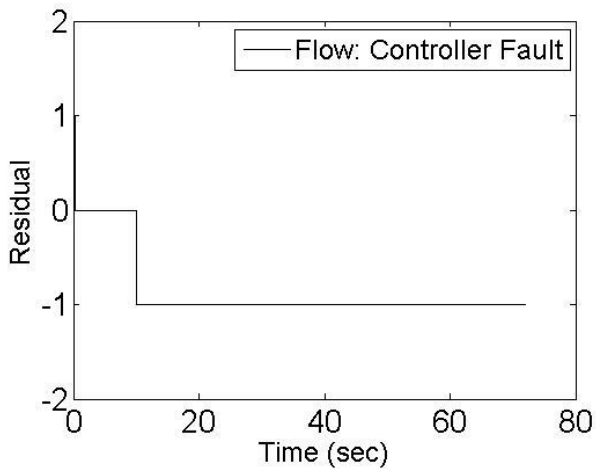


Figure 16. Voltage controller fault residual for Flow method

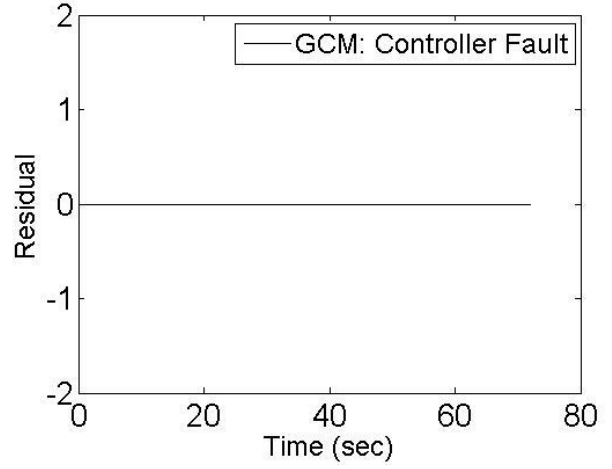


Figure 19: Voltage controller fault residual for GCM method

Analysis of Flow Method Results All three fault cases using the Flow method show a short-term disagreement $r \neq 0$ between predicted and output states at time $t=0.2$ seconds but returns to agreement $r = 0$ immediately at $t=0.3$ seconds. The disagreement occurs before a fault is injected at time $t=10$ seconds. This disagreement at $t=0.2$ could trigger a false alarm during fault detection. Similar rapid switching behavior also occurs in the diode fault residual in Figure 15. To distinguish between the similar switching behavior of false alarms with real faults and to build confidence in the diagnostic algorithm, a fault will only be detected if the residual shows disagreement for at least 0.2 seconds. The belt fault will be detected at $t=38.5$ seconds. The diode fault will be detected at $t=10.8$ seconds. The controller fault will be detected at 10.3 seconds.

Isolation of a detected fault is attempted by monitoring the mean of the residual during a finite time window T following detection. The time window T , chosen to be 10 seconds, is further divided into 5 subwindows and the mean inside each subwindow is computed as shown in Figure 20. Then a probability density function (pdf) of the means is computed using Matlab function `ksdensity`. This is done for all healthy and faulty cases. The pdfs from `ksdensity` are used for threshold analysis, so that probability of false alarm and misdetection are minimized as shown in Figure 21.

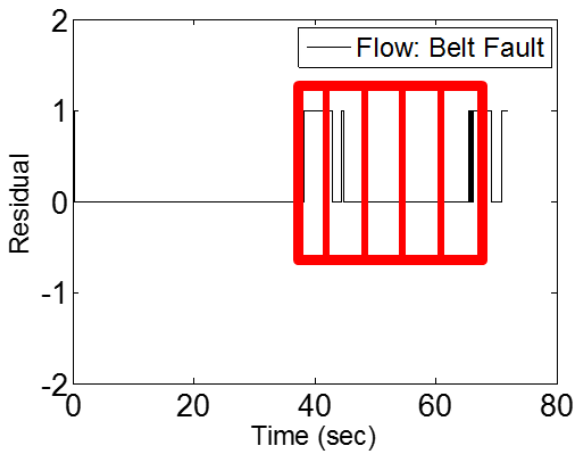


Figure 20: Window T and subwindows after fault detection

Figure 21 shows that the means of the controller and belt fault are isolatable but the diode fault is not. The mean of the diode fault and healthy pdfs are too similar to draw conclusions. Probabilities of false alarm and misdetection are tabulated in Table 4.

Based on the Flow method approach, the belt fault will be isolated at $t=48.5$ seconds and the controller fault will be isolated at time $t=20.3$ seconds. The diode fault will not be isolatable.

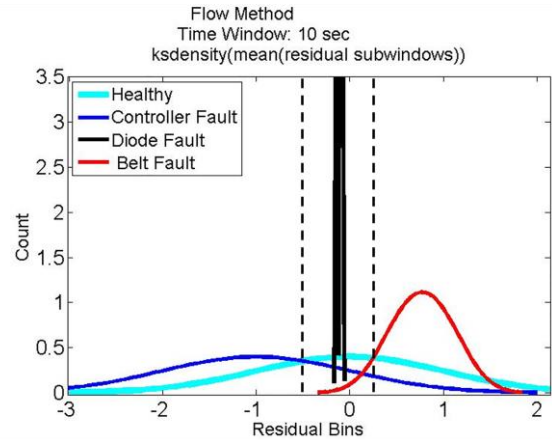


Figure 21: Persistent fault for Flow method with time window 10 seconds

	Threshold	P(False Alarm)	P(Misdetection)
Belt Slip	0.2732	0.3912	0.0749
Diode	N/A	N/A	N/A
Voltage Controller	-0.6061	0.2610	0.3455

Table 4: Threshold analysis for Flow method with persistent faults

Analysis of GCM Method Results The GCM method residuals show similar behavior compared to the Flow method residuals. The controller fault will not be detected or isolated because the residual never deviates from zero. The controller fault causes the output to transition to a nonadjacent cell and GCM method allows for nonadjacent cell transitions. Therefore, the residual of controller fault is always zero.

To distinguish between the similar switching behavior of false alarms with real faults and to build confidence in the diagnostic algorithm, a fault will only be detected if the residual shows disagreement for at least 0.2 seconds. The belt fault will be detected at $t=25.6$ seconds. The diode fault will be detected at $t=52.6$ seconds. The controller fault will not be detected.

Isolation of a detected fault is attempted by monitoring the mean of the residual during a finite time window T following detection. The time window T , chosen to be 10 seconds, is further divided into 5 subwindows and the mean inside each subwindow is computed as shown in Figure 20. Then a probability density function (pdf) of the means is computed using Matlab function `ksdensity`. This is done for all healthy and faulty cases. The pdfs from `ksdensity` are used for threshold analysis, so that probability of false alarm and misdetection are minimized as shown in Figure 22.

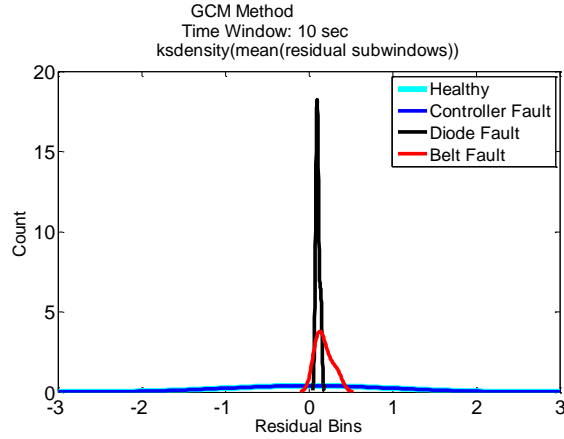


Figure 22: Persistent fault for GCM method with time window 10 seconds

Figure 22 shows that the means of the means of the pdfs are clustered on top of each other around residual value zero. The mean of the faulty and healthy pdfs are too similar to draw conclusions. Probabilities of false alarm and misdetection cannot be computed as shown in Table 5.

	Threshold	P(False Alarm)	P(Misdetection)
Belt Slip	N/A	N/A	N/A
Diode	N/A	N/A	N/A
Voltage Controller	N/A	N/A	N/A

Table 5: Threshold analysis for GCM method with persistent faults

Based on the GCM approach, the belt fault, diode fault, and controller fault will not be isolated.

5.3.2. Intermittent Faults with Noise and Threshold Analysis

Table 6 details the selected injection time and magnitude of fault relative to nominal that were injected during simulation. Intermittent faults have been considered where the faults are injected and then removed for a few cycles. The faults were injected for $10 < t < 20$, $30 < t < 40$, and $50 < t < 60$ seconds. The faults are simulated by modifying the nominal inputs of the system.

For the intermittent fault case, noise has been added to each fault according to Table 7.

Fault	Injection times (s)	Modified Input	Resulting % drop with respect to nominal
Belt Slip	10-20, 30-40, 50-60	ω_e	80
Diode	10-20, 30-40, 50-60	V_{dc}	N/A (faulty connection)
Voltage Controller	10-20, 30-40, 50-60	V_{ref}	30

Table 6: Fault injection time and magnitude

Fault	Input	Units	Mean	Variance
Belt Slip	ω_e	(rad/sec)	0	3
Diode	V_{dc}	(V)	0	0.03
Voltage Controller	V_{ref}	(V)	0	0.2

Table 7: Noise construction

Disagreement between predicted and output states are clear after calculating the difference of quantized output state $[I_{dc}]$ and the predicted state. This difference is defined as the residual r in Eq. (11). The residual results for each fault case using the Flow method are shown in Figure 23 through Figure 25. The residual results for each fault case using GCM method are shown in Figure 26 through Figure 28.

Analysis of Flow Method Results The belt fault residual again shows some switching behavior. The diode fault residual shows much less switching behavior compared to persistent fault case. The controller fault residual square wave behavior once the fault is injected at 10 seconds.

To distinguish between the similar switching behavior of false alarms with real faults and to build confidence in the diagnostic algorithm, a fault will only be detected if the residual shows disagreement for at least 0.2 seconds. The belt fault will be detected at $t=20.6$ seconds. The diode fault will be detected at $t=52.2$ seconds. The controller fault will be detected at 10.3 seconds.

Isolation of a detected fault is attempted by monitoring the mean of the residual during a finite time window T following detection. The time window T , chosen to be 10 seconds, is further divided into 5 subwindows and the mean inside each subwindow is computed as shown in Figure 20. Then a probability density function (pdf) of the means is computed using Matlab function `ksdensity`. This is done for all healthy and faulty cases. The pdfs from `ksdensity` are used for threshold analysis, so that probability of false alarm and misdetection are minimized as shown in Figure 29.

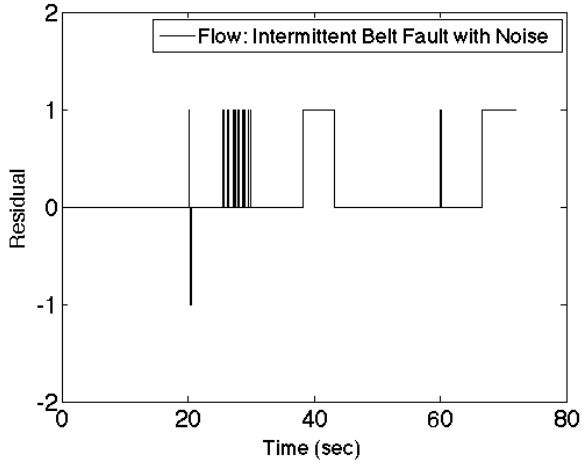


Figure 23: Noisy intermittent belt fault residual for Flow method

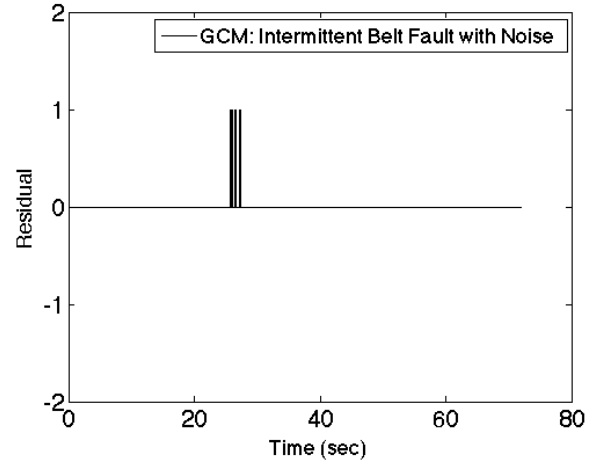


Figure 26: Noisy intermittent belt fault residual for GCM method

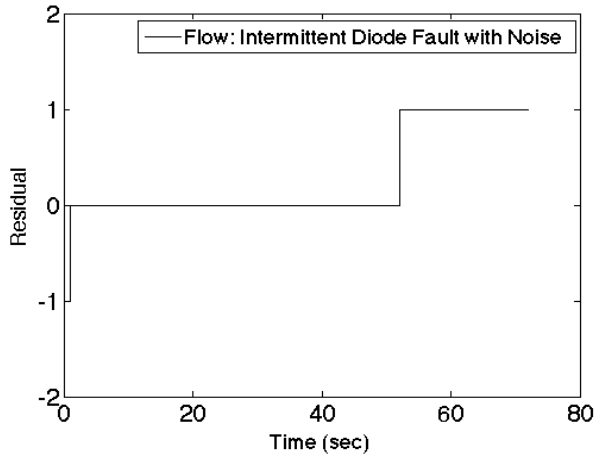


Figure 24: Noisy intermittent diode fault residual for Flow method

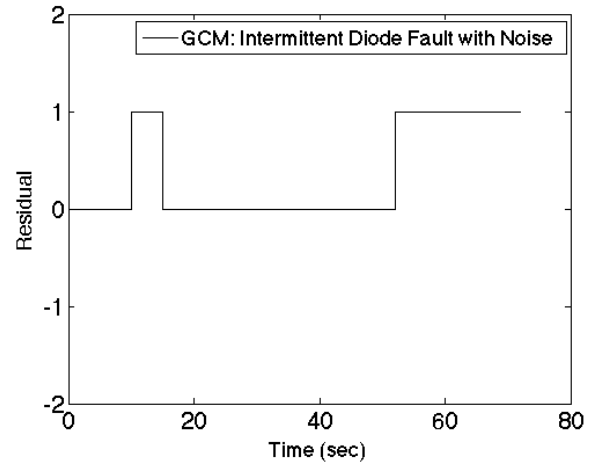


Figure 27: Noisy intermittent diode fault residual for GCM method

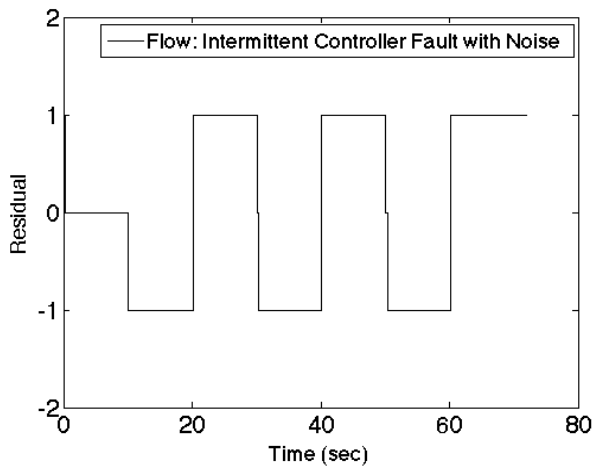


Figure 25: Noisy intermittent voltage controller fault residual for Flow method

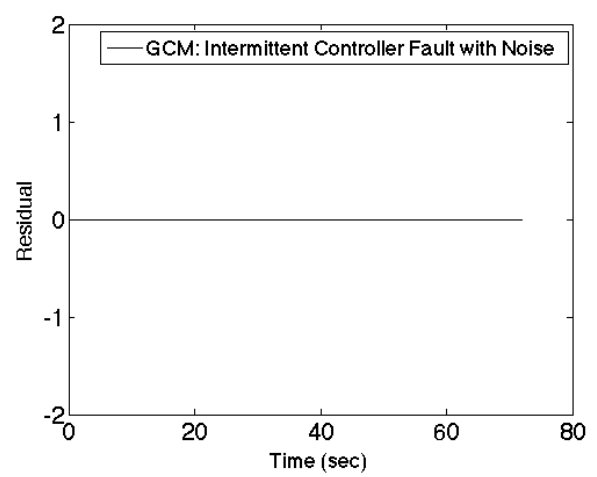


Figure 28: Noisy intermittent voltage controller fault residual for GCM method

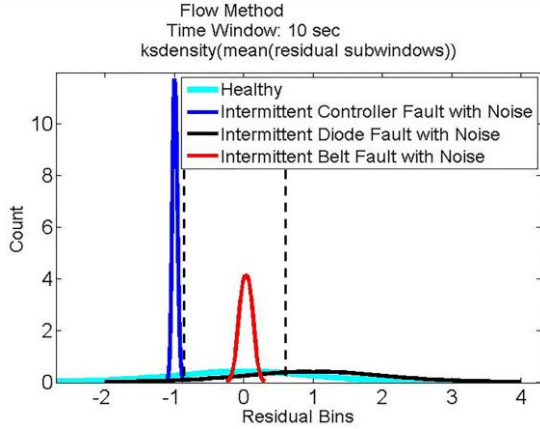


Figure 29: Noisy intermittent fault for Flow method with time window 10 seconds

Figure 29 shows that the means of the controller and diode faults are isolatable but the controller fault is not. The mean of the belt fault and healthy pdfs are too similar to draw conclusions. Probabilities of false alarm and misdetection are tabulated in Table 8.

	Threshold	P(False Alarm)	P(Misdetection)
Belt Slip	N/A	N/A	N/A
Diode	0.4848	0.3019	0.3019
Voltage Controller	-0.8929	0.1885	0.0028

Table 8: Threshold analysis for Flow method with noisy intermittent faults

Based on the Flow method approach, the diode fault will be isolated at $t=62.2$ seconds and the controller fault will be isolated at time $t=20.3$ seconds. The belt fault will not be isolated.

Analysis of GCM Method Results The belt fault residual shows minimal switching behavior. The diode fault shows much less switching behavior compared to the persistent fault case. The controller fault residual is zero similar to the persistent fault case.

To distinguish between the similar switching behavior of false alarms with real faults and to build confidence in the diagnostic algorithm, a fault will only be detected if the residual shows disagreement for at least 0.2 seconds. The belt fault will be detected at $t=25.9$ seconds. The diode fault will be detected at $t=10.2$ seconds. The controller fault will not be detected. The controller fault will not be detected or isolated because the residual never deviates from zero. The controller fault causes the output to transition to a nonadjacent cell and GCM method allows for nonadjacent cell transitions. Therefore, the residual of controller fault is always zero.

Isolation of a detected fault is attempted by monitoring the mean of the residual during a finite time window T following detection. The time window T , chosen to be 10 seconds, is further divided into 5 subwindows and the mean inside each subwindow is computed as shown in Figure 20. Then a probability density function (pdf) of the means is computed using Matlab function `ksdensity`. This is done for all healthy and faulty cases. The pdfs from `ksdensity` are used for threshold analysis, so that probability of false alarm and misdetection are minimized as shown in Figure 30.

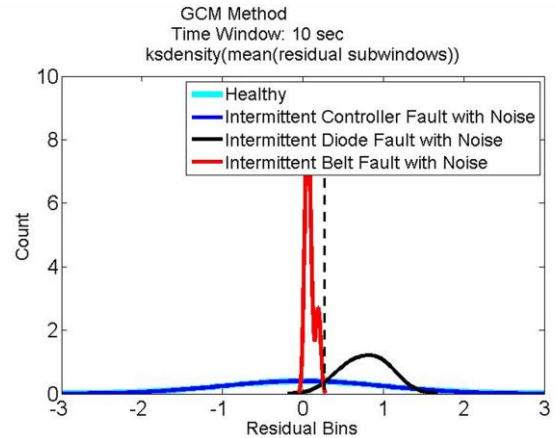


Figure 30: Noisy intermittent fault for GCM method with time window 10 seconds

Figure 30 shows that the mean of the diode fault is isolatable but the belt and controller fault are not. The mean of the belt and controller faults and healthy pdfs are too similar to draw conclusions. Probabilities of false alarm and misdetection are tabulated in Table 9.

	Threshold	P(False Alarm)	P(Misdetection)
Belt Slip	N/A	N/A	N/A
Diode	0.2827	0.3912	0.0536
Voltage Controller	N/A	N/A	N/A

Table 9: Threshold analysis for Flow method with noisy intermittent faults

Based on the GCM method approach, the diode fault will be isolated at $t=20.2$ seconds. The belt and controller faults will not be isolated.

5.3.3. FDI Summary

In summary, Table 10 and Table 11 list the probabilities of false alarm and misdetection for the three faults under persistent and noisy intermittent fault conditions. The results show that there is a high probability of false alarm and misdetection for the Flow method when compared to the GCM method.

	P(False Alarm)		P(Misdetection)	
	Flow	GCM	Flow	GCM
Belt Slip	0.56	0.31	0.15	0.01
Diode	0.05	0.56	0.13	0.02
Voltage Controller	0	N/A	0	N/A

Table 10: Threshold analysis with persistent faults

	P(False Alarm)		P(Misdetection)	
	Flow	GCM	Flow	GCM
Belt Slip	N/A	N/A	N/A	N/A
Diode	0.3019	0.3912	0.3019	0.0536
Voltage Controller	0.1885	N/A	0.0028	N/A

Table 11: Threshold analysis with noisy intermittent faults

Table 12 contains the detection and isolation times for both Flow and GCM methods under noisy intermittent fault condition. The Flow method can isolate two of the three faults while the GCM method cannot isolate any. The Flow method can detect the controller fault while the GCM method cannot. The Flow method can detect the diode fault faster than the GCM method. The GCM method can detect the belt fault faster than the Flow method. It is clear that the Flow method gives best results since more detection and isolation is achievable.

	Fault	Belt Slip	Diode	Voltage Controller
		Injection time (s)	10	10
Flow Method	Detection time (s)	38.5	10.8	10.3
	Isolation time (s)	44.5	N/A	20.3
GCM Method	Detection time (s)	25.6	52.5	N/A
	Isolation time (s)	N/A	N/A	N/A

Table 12: Fault detection and isolation times for three faults under persistent fault condition

Table 13 contains the detection and isolation times for both Flow and GCM methods under noisy intermittent fault

condition. The Flow method can isolate two of the three faults while the GCM method can only isolate one. The Flow method can detect and isolate the controller fault while the GCM method cannot. The Flow method can detect the diode fault faster than the GCM method. The GCM method can detect and isolate the diode fault faster than the Flow method. It is clear that the Flow method gives best results since more detection and isolation is achievable.

	Fault	Belt Slip	Diode	Voltage Controller
		Injection time (s)	10-20, 30-40, 50-60	10-20, 30-40, 50-60
Flow Method	Detection time (s)	20.6	52.2	10.3
	Isolation time (s)	N/A	62.2	20.3
GCM Method	Detection time (s)	25.9	10.2	N/A
	Isolation time (s)	N/A	20.2	N/A

Table 13: Fault detection and isolation times for three faults under noisy intermittent fault case

Different fault magnitudes might require different isolation thresholds. This paper only considers three discrete fault modes.

6. CONCLUSION

This paper compares the GCM method and a new stochastic Flow method for calculating state transition probabilities within a dynamic system. These methods are compared by detecting and identifying predetermined faults in a vehicle alternator system. The methods vary based on computational complexity and the ability to isolate all faults. The Flow method can isolate two of the three faults (belt and controller faults) while the GCM method cannot isolate any under the persistent fault condition. The GCM method could only detect but not isolate the controller reference fault. The Flow method can isolate two of the three faults (diode and controller faults) while the GCM method can only isolate the diode fault under the noisy intermittent fault condition. Overall, the new stochastic method is preferred since it can detect and isolate more than the GCM method.

ACKNOWLEDGEMENTS

Support for this research has been provided in part by the US Department of Energy GATE program and National Science Foundation under Grant No. 0825655.

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the

author(s) and do not necessarily reflect the views of the National Science Foundation.

NOMENCLATURE

ω_m	engine rotational speed
ω_e	alternator rotational speed
V_{dc}	battery DC voltage
V_f	field voltage
V_{ref}	voltage controller reference
I_{dc}	alternator output current
I_{load}	vehicle load current
I_B	battery charging current
z_1	first state space state
z_2	second state space state and output
u	state space input
$a(\omega_e)$	state space parameter dependent on alternator rotational speed
$b(\omega_e)$	state space parameter dependent on alternator rotational speed
z	current state
z'	possible future state
z_1^{\min}	minimum z_1 value
z_1^{\max}	maximum z_1 value
z^*	flow transition point on z_1 axis on side 1 of state z
z^{**}	flow transition point on z_1 axis on side 2 of state z
$z_2^{(1)}$	upper boundary of state z
$z_2^{(2)}$	lower boundary of state z
φ^+	flow up
φ^-	flow down
f	general function
\vec{F}	Field vector
\vec{n}	normal vector
C	general closed curve
A	area within curve C
r	line integral direction along curve C
φ_{in}	total flow into state z
φ_{out}	total flow out of state z
φ_{net}	net flow for given state z
z^+	state above state z
z^-	state below state z
L	time varying probability transition matrix
$[I_{dc}]$	quantized alternator output current
$[I_{dc, predicted}]$	predicted quantized alternator output current
r	residual

REFERENCES

- Chiu, H. M. & Hsu, C. S. (1986). A Cell Mapping Method for Nonlinear Deterministic and Stochastic Systems—Part II: Examples of Application. *ASME Journal of Applied Mechanics*. September Vol. 53. pp. 702-710. Doi: 10.1115/1.3171834
- Hashemi, A., & Pisu, P. (2011a). Adaptive Threshold-based Fault Detection and Isolation for Automotive Electrical Systems (pp. 1013-1018), *World Congress on Intelligent Control and Automation*. June 21-25, Taipei, Taiwan. doi: 10.1109/WCICA.2011.5970668
- Hashemi, A., & Pisu, P. (2011b). Fault Diagnosis in Automotive Alternator System Utilizing Adaptive Threshold Method. *Annual Conference of Prognostics and Health Management Society*. September 25-29, Montreal, Canada.
- Hashemi, A., (2011). *Model-Based System Fault Diagnosis Utilizing Adaptive Threshold with Application to Automotive Electrical Systems*. Masters dissertation. Clemson University, Clemson, South Carolina, USA. http://etd.lib.clemson.edu/documents/1314212419/Hashemi_clemson_0050M_11327.pdf
- Hsu, C. S. (1980). A Theory of Cell-to-Cell Mapping Dynamical Systems. *ASME Journal of Applied Mechanics*. December Vol. 47. pp. 931-939. Doi:10.1115/1.3153816
- Hsu, C. S. & Guttalu, R. S. (1980). An Unravelling Algorithm for Global Analysis of Dynamical Systems: An Application to Cell-to-Cell Mappings. *ASME Journal of Applied Mechanics*. December Vol. 47. pp. 940-948. Doi: 10.1115/1.3153817
- Hsu, C. S. (1981). A Generalized Theory of Cell-to-Cell Mapping for Nonlinear Dynamical Systems. *ASME Journal of Applied Mechanics*. September Vol. 48. pp. 634-642. Doi: 10.1115/1.3157686
- Hsu, C. S. (1982). A Probabilistic Theory of Nonlinear Dynamical Systems Based on the Cell State Space Concept. *ASME Journal of Applied Mechanics*. December Vol. 49. pp. 895-902. Doi: 10.1115/1.3162633
- Hsu, C. S. & Chiu, H. M. (1986). A Cell Mapping Method for Nonlinear Deterministic and Stochastic Systems—Part I: The Method of Analysis. *ASME Journal of Applied Mechanics*. September Vol. 53. pp. 695-701. Doi: 10.1115/1.3171833
- Hsu, C. S., John F., Marsden, J. E., & Sirovich, L. (Eds.). (1987). *Cell-to-Cell Mapping: A Method for Global Analysis for Nonlinear Systems*. New York: Springer.
- Kastner, M. (2010). Monte Carlo Methods in Statistical Physics: Mathematical Foundations and Strategies. *Communications in Nonlinear Science and Numerical Simulation*. June Vol. 15. pp. 1589-1602. Doi: 10.1016/j.cnsns.2009.06.011
- Mohon, S., & Pisu, P. (2013). A Stochastic Modeling Approach of Quantized Systems with Application to Fault Detection and Isolation of an Automotive Electrical Power Generation Storage System. *Annual Conference of the Prognostics and Health Management Society*, October 14-17, New Orleans, Louisiana.
- Scacchioli, A., Rizzoni, G., Salman, M. A., Li, W., Onori, S., and Zhang, X. (2013). Model-based Diagnosis of an Automotive Electric Power Generation and Storage System, *IEEE Transactions on Systems, Man and Cybernetics*, vol. 44, no. 1, pp. 72–85.

- Scacchioli, A., Rizzoni, G., & Pisu, P., (2006). Model-Based Fault Detection and Isolation in Automotive Electrical Systems, *ASME International Mechanical Engineering Congress and Exposition* (pp. 315-324), November 5-10, Chicago, Illinois, USA. doi: 10.1115/IMECE2006-14504
- Schröder, J. (2003). *Modelling, State Observation and Diagnosis of Quantised Systems*. Germany: Springer.
- Sobol, I. (1994). *A Primer for the Monte Carlo Method*. Boca Raton: CRC Press.
- Wang, J. (1999). Transition Matrix Monte Carlo Method. *Proceedings of the Europhysics Conference on Computational Physics (CCP)*. September Vol. 121. pp. 22-25. Doi: 10.1016/S0010-4655(99)00270-2
- Zhang, X., Uliyar, H., Farfan-Ramos, L., Zhang, Y., & Salman, M. (2010). Fault Diagnosis of Automotive Electric Power Generation and Storage Systems. *IEEE International Conference on Control Applications*. pp. 719-724, September 8-10, Yokohama, Japan.

Computer Engineering at Clemson University. He is also a faculty member at the Clemson University International Center for Automotive Research. His research interests are in the area of fault diagnosis with application to vehicle systems, and energy management control of hybrid electric vehicles; he also worked in the area of sliding mode control and robust control. He is member of the ASME and SAE, and a recipient of the 2000 Outstanding Ph.D. Student Award by the Ohio State University Chapter of the Honor Society of Phi Kappa Phi.

BIOGRAPHIES



Sara Mohon was born in Groton, Connecticut in 1987. She received her B.S. in Physics from the College of William and Mary (Williamsburg, VA, USA) in 2009 and M.S. in Automotive Engineering from Clemson University (Clemson, SC, USA) in 2012. She is currently a Ph.D. student at Clemson

University studying Automotive Engineering. She has completed summer internships at NASA Langley Research Center (Hampton, VA, USA) Thomas Jefferson National Accelerator Facility (Newport News, VA, USA), NOAA David Skaggs Research Center (Boulder, CO, USA), and Johns Hopkins University Applied Physics Laboratory (Laurel, MD, USA). She has completed a battery research project at BMW Manufacturing Company (Spartanburg, SC, USA) that resulted in filing a patent about methods to determine the condition of a battery. Her research interests are control, diagnostics, and prognostics for hybrid and electric vehicles. She is a member of ASME, SAE, SWE, and IEEE and received the national SEMA Top Student Award in 2012.



Pierluigi Pisu was born in Genoa, Italy in 1971. He received his Ph.D. in Electrical Engineering from Ohio State University (Columbus, Ohio, USA) in 2002. In 2004, he was granted two US patents in area of model-based fault detection and isolation. He is currently an Associate Professor in the Department of

Automotive Engineering at Clemson University and holds a joint appointment with the Department of Electrical and