Reliability Evaluation of Steering System Using Dynamic Fault Tree

Sungil Byun, Inseok Yang, Moo Geun Song, and Dongik Lee

Abstract— This paper addresses a dynamic fault tree analysis (DFTA) to predict the reliability of a steering system. Reliability evaluation is a vital task to prevent any potential failure in a system by identifying vulnerable parts of the system and managing them effectively. Safety-critical systems, such as electric vehicles, have many components whose failure may cause a catastrophic accident. The steering system in a vehicle is one of the most critical subsystems requiring a high level of reliability. In this paper, a DFTA method using the Simpson’s rule is proposed to evaluate the reliability of a steering system. A set of simulation results shows that the proposed method can overcome the problem of low accuracy with the existing approximation method while requiring no excessive calculation time of the Markov chain method.

I. INTRODUCTION

Recently, automotive industries have been paying much attention in the development of the advanced concept of electric vehicles. Thanks to the advances in electronics, some mechanical linkages/hydraulics in a system have been replaced into electric devices which offer benefits in terms of reduced wiring, costs for maintenance and implementation, and so on. For this reason, the automobiles adopt more and more electric devices. However, electric devices in a vehicle may increase the potential possibility of failure with sensors/actuators and controllers. Any failure in a vehicle can be critical to not only drivers but also pedestrian or other drivers. Therefore, a high level of reliability is required.

One of the useful methods that can evaluate the reliability of a system is the fault tree analysis (FTA) [1]. The FTA provides opportunities to diagnose the potential failure of a system by predetermining the reliability of the whole system. The FTA presents the causality of system failure logically, and improves the reliability by analyzing the probability of failure [1]. However, a standard fault tree model cannot represent some important dynamic behaviors in advanced fault-tolerant systems, such as fault and error recovery, sequence-dependent failures, and the use of spares. The Markov model approach is an alternative that can approximate such a dynamic system. However, the construction of a Markov model of a complicated system is tedious and error prone. To overcome this problem, dynamic fault tree (DFTA) was developed [2].

After constructing a dynamic fault tree, the evaluation of fault tree must be performed by calculating the top event failure probability. In a standard fault tree, a general numerical method is used to calculate the top event failure probability. On the other hand, Markov chains [3] are applied in the dynamic fault tree to get the probability. However, Markov chains have difficulties with an excessive calculation time. To overcome this problem, Han et al. developed an approximation algorithm based on trapezoidal rule [4]. This method calculates the integral of dynamic gates in a short time, but has low accuracy.

In this paper, a DFTA method that can achieve an accurate probability without excessive calculation time by adopting an approximation algorithm based on Simpson’s rule [5]. The effectiveness of the proposed method is then examined with a car steering system. The rest of the paper is organized as follows. Section II describes the dynamic fault tree of the steering system. Section III explains the evaluation method of dynamic fault tree. In section IV, simulation result of each evaluation methods using Matlab is presented. In the end, conclusions are given in section V.

II. FAULT TREE ANALYSIS MODEL OF STEERING SYSTEM

A car steering system is typical safety-critical part that has a significant impact on the safety of driver. Fig. 1 shows a conceptual design for the steering system, consisting of a controller, a steering wheel, and a road wheel subsystem [6]. The steering wheel system contains several sensors to measure the driver’s steering input. The road wheel subsystem contains some actuators to drive the wheels. In this paper, it is assumed that the steering system has two controllers, four sensors, and one actuator. Fig. 2 presents the dynamic fault tree of the steering system. The description of dynamic fault tree gates, such as CSP gate in Fig. 2, is represented in the Appendix.

1) Controller

In this paper, it is assumed that there are two local controllers for the steering wheel subsystem and road wheel subsystem, respectively. These two controllers have priority. The controller of steering wheel subsystem fails after the controller of road wheel subsystem fails, and then the whole control unit fails. Therefore, the control unit can be presented by a priority-AND gate.
2) **Sensor**
Each subsystem has its own sensor with a redundancy. Each redundant sensor works as a secondary. The sensor unit fails after the original sensor and spare sensor fail. Cold-spare gates and OR-gate can present the sensor unit.

3) **Actuator**
Failure of road wheel actuator can lead to the failure of the whole steering system.

III. **THE PROPOSED RELIABILITY EVALUATION METHOD**

There are several approaches to analyze the dynamic fault trees including Markov chains [7] and approximation algorithms [4].

**A. Markov chains approach**

The Markov chains (MC) approach is the basic method that can get the probability of top event failure exactly. However, building a Markov chain model for a complicated system takes very excessive time and is prone to an error. For applying the MC method, dynamic fault tree is separated to static and dynamic subtrees. Then, Binary Decision Diagram (BDD) and MC are applied in the DFT to process these static and dynamic subtrees, respectively. Fig. 3 depicts the process of Priority-AND gate to MC.
Transition is the change of system state in MC. Firstly, the first order transfer process is explained [4]. $\lambda_{0,1}$ is the transition rate from state “0” to failure state, and $\lambda_{0,NF}$ is the transition rate from state “0” to non-failure state. $P_{0}(t)$ is the transition rate from state $i$ to state $j$, then

\[
P(t) = \begin{pmatrix}
p_{00}(t) & p_{01}(t) & p_{0NS}(t) \\
p_{10}(t) & p_{11}(t) & p_{1NS}(t) \\
p_{NS0}(t) & p_{NS1}(t) & p_{NSNS}(t)
\end{pmatrix}
\]

satisfies the differential equations and the initial condition:

\[
\begin{aligned}
P(t) &= P(t)Q \\
(0) &= E
\end{aligned}
\]

where $E$ is identity matrix.

\[
Q = \begin{pmatrix}
-(\lambda_{0,1} + \lambda_{0,NF}) & \lambda_{0,1} & \lambda_{0,NF} \\
0 & 0 & 0 \\
0 & 0 & 0
\end{pmatrix}
\]

Solve (1) and yield to

\[
P_1(t) = p_{01}(t) = \frac{\lambda_{0,1}}{\lambda_{0,1} + \lambda_{0,NF}}(1 - e^{-(\lambda_{0,1} + \lambda_{0,NF})t}) \tag{2}
\]

where $\lambda_{0,1} \geq 0, \lambda_{0,NF} \geq 0$.

The calculating equations are changed based on the number of transition. The equation of calculating unreliability based on the transition number $n$ is expressed as follow:

\[
P_n(t) = p_{0n}(t) = \frac{1}{\prod_{i=1}^{n}(\lambda_{i-1,1} + \lambda_{i,NF})} \prod_{i=1}^{n} - e^{-(\lambda_{i-1,1} + \lambda_{i,NF})t} \tag{3}
\]

where, $\lambda_{i-1,1} > 0, \lambda_{i-1,NF} \geq 0, \lambda_{i-1,i}$ is probability of failure from state $(i-1)$ to state $i$, and $\lambda_{i-1,NF}$ is probability of failure from state $(i-1)$ to non-failure state.

**B. Approximation Algorithm Approach**

The approximation algorithm approach was developed to overcome the excessive time for converting of MC by calculating integral terms directly. Conventional approximation algorithms use trapezoidal rule to solve the integral of dynamic gates to improve the calculation time. For example, sequence-enforcing (SEQ) gate fails when bottom event occurred from left to right in sequence. The probability of top event failure of time $T$ is:

\[
P(t) = p\{T_1 \leq T_2 \leq \ldots \leq T_m \leq t\} = \int_0^t \cdots \int_0^{T_{m-1}} dp_{00}(x_m) \cdots dp_{12}(x_2) dp_{11}(x_1)
\]

Approximation algorithm is applied to (4), (5) yielding to

\[
P(t) \approx \sum_{i=1}^{m} \left[ \prod_{j=i}^{M} (P(i,j) - P(i,(j-1)h)) \right]
\]

where, $m$ is the number of bottom event, $M$ is the number of time interval, and $h = t / M$ is the steps.

The main drawback with an approximation algorithm based on the trapezoidal rule is a large error. For example, the approximation results of a system combined the PAND and two CSP gates have 12% of error [4].

**C. The Proposed Approximation Algorithm**

In order to overcome the problem of low accuracy with the existing approximation algorithms, this paper proposes a modified approximation algorithm. The proposed method calculates the equations of approximation algorithm using Simpson’s rule [5] without converting of MC. The Simpson’s rule, one of the approximate integration methods, can offer a higher accuracy than trapezoidal rule, and have a faster calculation than other approximation methods [8]. The Simpson’s rule calculates integral value of a curve graph as shown in Fig. 4. The integral in $[a, b]$ can be approximated as follows:

\[
\int_a^b f(x)dx \approx \Delta x \sum_{i=0}^{n-1} \left( y_0 + 4y_1 + y_2 + \cdots + 4y_{n-2} + y_n \right)
\]

where, $\Delta x = \frac{b-a}{n}$, $y_i = f(x_i)$, $x_i = a + i\Delta x$ for $i = 0, 1, \ldots, n$. The modified approximation algorithm is applied to (4), (5) yielding to

\[
P(t) \approx \sum_{i=1}^{m} \left[ \prod_{j=i}^{M} (P(i,j) - P(i,(j-1)h)) \right]
\]

where, $m$ is the number of bottom event, $M$ is the number of time interval, and $h = t / M$ is the steps.

The main drawback with an approximation algorithm based on the trapezoidal rule is a large error. For example, the approximation results of a system combined the PAND and two CSP gates have 12% of error [4].

**C. The Proposed Approximation Algorithm**

In order to overcome the problem of low accuracy with the existing approximation algorithms, this paper proposes a modified approximation algorithm. The proposed method calculates the equations of approximation algorithm using Simpson’s rule [5] without converting of MC. The Simpson’s rule, one of the approximate integration methods, can offer a higher accuracy than trapezoidal rule, and have a faster calculation than other approximation methods [8]. The Simpson’s rule calculates integral value of a curve graph as shown in Fig. 4. The integral in $[a, b]$ can be approximated as follows:

\[
\int_a^b f(x)dx \approx \Delta x \sum_{i=0}^{n-1} \left( y_0 + 4y_1 + y_2 + \cdots + 4y_{n-2} + y_n \right)
\]
Generally, approximation methods have a cumulative error [8]. To solve the problem and to get a precise probability, the standard error of Simpson’s rule is modified by presuming based on error between MC method and approximation algorithm based on Simpson’s rule. The modified error at $\xi$ in $[a, b]$ is given by

$$C = f(\xi) \left( e^{-2(162e^{-1})^2} b - a \right) \left( f(\xi) b - a \right)$$  

where, $a$ and $b$ are the end points.

IV. SIMULATION SET UP AND RESULTS

A. Simulation Set Up

As shown in Fig 2 In this paper, the steering system that has two controllers, four sensors, and one actuator is considered. In table 1, the probability of failure in each device is summarized.

B. Simulation Results

The DFT results with the steering system are compared between three methods: (i) MC, (ii) an approximation algorithm using the trapezoidal rule, and (iii) the proposed approximation algorithm. Table 2 shows the unreliability of the whole steering system. And Fig. 5 shows the errors between MC and the approximation algorithms using the trapezoidal rule and the proposed Simpson’s rule. In Fig. 5, the error result of the trapezoidal based approximation algorithm is around 13% maximally. However, the error of the proposed algorithm is less than 4%. Hence, the proposed approximation algorithm based on the Simpson’s rule achieves more accurate results than the existing trapezoidal based algorithm does.

<table>
<thead>
<tr>
<th>Component</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Controller 1</td>
<td>6.28E-06</td>
</tr>
<tr>
<td>Controller 2</td>
<td>2.60E-06</td>
</tr>
<tr>
<td>Actuator</td>
<td>7.90 E-07</td>
</tr>
<tr>
<td>Sensor 1</td>
<td>6.06 E-04</td>
</tr>
<tr>
<td>Spare Sensor 1</td>
<td>8.76 E-05</td>
</tr>
<tr>
<td>Sensor 2</td>
<td>6.06E-04</td>
</tr>
<tr>
<td>Spare Sensor 2</td>
<td>8.76 E-05</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Hour(h)</th>
<th>MC</th>
<th>Existing Alg</th>
<th>Proposed Alg</th>
</tr>
</thead>
<tbody>
<tr>
<td>500</td>
<td>3.3581 E-04</td>
<td>2.951 E-04</td>
<td>3.3010 E-04</td>
</tr>
<tr>
<td>1000</td>
<td>1.2111 E-03</td>
<td>1.0801 E-03</td>
<td>1.1796 E-03</td>
</tr>
<tr>
<td>1500</td>
<td>2.4655 E-03</td>
<td>2.2123 E-03</td>
<td>2.3891 E-03</td>
</tr>
<tr>
<td>2000</td>
<td>3.9821 E-03</td>
<td>3.5285 E-03</td>
<td>3.9303 E-03</td>
</tr>
<tr>
<td>2500</td>
<td>5.6748 E-03</td>
<td>4.9589 E-03</td>
<td>5.6237 E-03</td>
</tr>
<tr>
<td>3000</td>
<td>7.4806 E-03</td>
<td>6.6151 E-03</td>
<td>7.3160 E-03</td>
</tr>
<tr>
<td>3500</td>
<td>9.3529 E-03</td>
<td>8.2231 E-03</td>
<td>9.2033 E-03</td>
</tr>
<tr>
<td>4000</td>
<td>1.1259 E-02</td>
<td>1.0025 E-02</td>
<td>1.1135 E-02</td>
</tr>
<tr>
<td>4500</td>
<td>1.3174 E-02</td>
<td>1.1662 E-02</td>
<td>1.2858 E-02</td>
</tr>
<tr>
<td>5000</td>
<td>1.5081 E-02</td>
<td>1.3413 E-02</td>
<td>1.4764 E-02</td>
</tr>
<tr>
<td>5500</td>
<td>1.6967 E-02</td>
<td>1.4807 E-02</td>
<td>1.6662 E-02</td>
</tr>
</tbody>
</table>

Figure 5. The graph of error of evaluation results of steering system.

V. CONCLUSION

This paper presented a dynamic fault tree analysis on a safety-critical steering system of an electric vehicle. In order to achieve a higher accuracy without excessive calculation time, the proposed method is based on an approximation algorithm using the Simpson’s rule. The proposed method is applied to a steering system in order to evaluate its reliability. By comparing the evaluation results with the existing trapezoidal rule, it was shown that the proposed method provides a higher accuracy than the existing approximation method while requiring no excessive calculation time of MC.

APPENDIX

The static fault tree uses only AND and OR gates. However, DFT uses not only gates for the static fault tree but also specific gates to present the dynamic behaviors of a system. The following is the descriptions of dynamic gates [2].
Priority-AND gate (PAND): It is an AND gate with the condition that the events must occur in a specific order. The PAND gate has 2 inputs, A and B. If both events have not occurred, or if B occurred before A, then the gate does not fire. It makes output true only when B occurs after A.

Cold-Spare gate (CSP): It has a primary input and one or more alternate inputs. All inputs are basic events. The primary input is the one that is initially powered on, and the alternate inputs specify the (initially unpowered) components that are used as replacements for the primary unit. The CSP gate has one output which becomes true after all the input events occur.

Sequence-Enforcing gate (SEQ): The SEQ forces events to occur in a particular order. The input events are constrained to occur in the left-to-right order in which they appear under the gate. It allows the events to occur only in a specified order.

Functional-Dependency gate (FDEP): It has one trigger-input and some of dependent events depending on the trigger-input. When the trigger event occurs, the dependent basic events are forced to occur.

ACKNOWLEDGEMENT

This research was supported by the MKE(The Ministry of Knowledge Economy), Korea, under the CITRC(Convergence Information Technology Research Center) support program (NIPA-2013-H0401-13-1005) supervised by the NIPA(National IT Industry Promotion Agency.)

REFERENCES


TABLE III. THE SYMBOLS OF DYNAMIC GATES

<table>
<thead>
<tr>
<th>Name</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority-AND</td>
<td>![Priority-AND Diagram]</td>
</tr>
<tr>
<td>Cold-Spare</td>
<td>![Cold-Spare Diagram]</td>
</tr>
<tr>
<td>Sequence-Enforcing</td>
<td>![Sequence-Enforcing Diagram]</td>
</tr>
<tr>
<td>Functional-Dependency</td>
<td>![Functional-Dependency Diagram]</td>
</tr>
</tbody>
</table>