

# Software Design on Carrier Rocket Fault Trees Analysis System

Zhimin Zhang, Hongmei Li, Ting Li

Ministry of Technology, China Jiu Quan Satellite Launch Center, Jiuquan Gansu  
Email: zzm\_fletcher@163.com

Received: Sep. 23<sup>rd</sup>, 2019; accepted: Oct. 5<sup>th</sup>, 2019; published: Oct. 12<sup>th</sup>, 2019

---

## Abstract

Fault Trees Analysis (FTA) is very important when a carrier rocket is flying abnormally. It is often time-consuming and less efficient, however, for people to improvise a fault tree hurriedly when necessary. Based on the Visual Studio 2010 software development platform, a carrier rocket FTA system software is designed and implemented which makes use of the object-oriented programming technology. It enables us to be able to collect, analyze and manage a great deal of complicated carrier rocket fault trees. First, a recursive algorithm for FTA is introduced which allows for operations on a fault tree such as searching, editing, adding/removing, zooming and branch expanding/collapsing, saving, loading. Second, the C++ standard template library brings us great efficiency in software development and maintenance. Finally, the GDI+ technology provides us the ability to easily export a fault tree in terms of BMP, JPEG, GIF, TIFF as well as PNG. In addition to effectively saving a lot of time in fault trees construction, this software system can also improve fault diagnosis efficiency, help determiner locate fault quickly and exactly, which makes it a great significant role in carrier rockets' flights.

## Keywords

Carrier Rocket, Fault Diagnosis, Fault Trees Analysis, Software Design, Recursive Algorithm, Object Orientation

---

# 运载火箭故障树分析系统软件设计

张志民, 李红梅, 李 婷

中国酒泉卫星发射中心技术部, 甘肃 酒泉  
Email: zzm\_fletcher@163.com

收稿日期: 2019年9月23日; 录用日期: 2019年10月5日; 发布日期: 2019年10月12日

## 摘要

故障树分析是运载火箭飞行异常情况下故障诊断的重要环节,人工现场建立故障树费时费力效率低下。本文基于Visual Studio 2010软件开发平台,采用面向对象程序设计技术开发运载火箭故障树分析系统软件,实现对火箭大量而复杂的故障树的收集、分析与管理。采用递归算法实现故障树查找、编辑、增减、视图缩放、分支收展、保存和载入等功能;使用C++标准模板库提高系统开发效率和软件可维护性;采用GDI+技术方便地将故障树导出为BMP、JPEG、GIF、TIFF、PNG格式。该软件系统可有效节省人工现场建树时间,提高故障诊断效率,帮助决策者快速、准确、有效地定位故障,对运载火箭的正常飞行具有十分重要的意义。

## 关键词

运载火箭, 故障诊断, 故障树分析, 软件设计, 递归算法, 面向对象

Copyright © 2019 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

## 1. 引言

故障诊断是指系统在一定工作环境下查明导致系统某种功能失调的原因或性质,判断劣化状态发生的部位或部件以及预测状态劣化的发展趋势等,它包括故障检测、故障定位和故障预测。故障树在本质上是一种特殊的逻辑图,它提供了一种自顶向下、由果到因的故障诊断分析方法。利用故障树能从系统的故障状态出发,逐级进行推理分析,最终确定故障发生的基本原因、影响程度和发生概率[1] [2] [3] [4]。

就世界范围看,美国是最早研究故障诊断技术的国家。1967年,在美国宇航局和海军研究所的倡导和组织下成立了美国机械预防小组,开始有计划地对故障诊断技术分专题进行研究。英国对故障诊断技术的研究始于20世纪60年代末、70年代初,以R. A. Collacott博士为首的英国机器保健中心在宣传、培训、咨询以及诊断技术的开发方面做了大量的工作,取得了很好的效果。日本的诊断技术研究始于20世纪70年代中期,1971年新日铁以丰田利夫教授为首率先开展对故障诊断技术的研究,到1976年已达到实用的阶段。相比之下,我国故障诊断技术的研究起步较晚,从20世纪70年代末期开始,广泛的研究则从20世纪80年代开始发展起来,随后在各领域分别确定了故障诊断的目标、方向和试点单位。目前,故障诊断技术在我国的化工、冶金、电力和铁路等行业得到了广泛的应用[1] [2] [3] [4]。

航天技术的迅猛发展使得运载工具变得日益复杂,运载工具发生故障的概率也随之增大。某航天发射场对运载火箭飞行试验智能分析评估,特别是对火箭飞行异常情况下智能故障诊断的需求越来越迫切,对智能故障诊断方法的研究成为一项亟待解决的课题,而构建故障树是智能故障诊断方法中的重要环节。故障树分析是运载火箭飞行异常情况下故障分析与定位的重要环节,通过故障树分析可以准确定位故障。以往由于技术条件的限制,故障诊断转入故障树分析环节后都是通过人工现场建立故障树。故障树建好后,故障诊断人员才能基于所建立的故障树进行逐层分析排查,找出导致故障发生的一种或几种故障树底事件。建故障树大体上需要1到2天的时间,费时费力,效率低下;此外整个故障树分析过程也非常复杂,持续时间往往较长。

为解决航天发射场故障树分析中的问题, 本文设计开发运载火箭故障树分析系统软件。通过将提前设定的故障模式存入故障树系统库, 节省现场建立故障树时间, 为运载火箭飞行异常情况下的故障诊断工作做好充分的准备。软件系统具备动态建树、查询、分析、管理与维护等功能, 可实现对火箭大量而复杂的故障树的分析与管理。利用该系统可及时发现和定位故障, 给相关技术人员提供决策依据, 提高运载火箭飞行试验的保障能力。

下面先给出运载火箭故障树分析系统总体设计, 接下来介绍递归算法、C++标准模板库、GDI+三种关键技术, 以此为基础开发故障树分析系统软件, 并对软件性能做出评估。

## 2. 运载火箭故障树分析系统总体设计

运载火箭故障树分析系统的核心是故障树库, 任何与火箭飞行相关的故障树都可以纳入库中, 作为后续故障诊断时的技术储备。因此故障树库中累积的已有故障模式越丰富, 故障树覆盖的范围越广泛, 火箭飞行异常条件下进行故障分析能用到库中既有故障树的概率就越大, 就愈能发挥故障树库的作用, 故障诊断的效率就越高。另一方面, 一旦从故障树库中检索到某一故障树, 往往还需要现场对该故障树进行分析并能对它做出适应性修改, 以便定位故障。基于此故障树库必须具备两大功能: 1) 存储容量要大, 以便尽可能多地存储既有故障树; 2) 针对某具体故障树, 必须具备对其进行动态修改的功能诸如故障树节点增减、分支收展、大型复杂故障树的屏蔽、添加说明文字等, 确保最大程度地减少人力参与程度, 提高故障诊断效率。故障树分析系统软件总体设计方案如图 1 所示, 它包含故障树分析与故障树管理两大模块。其中故障树分析模块主要包括门事件、节点增减、分支收展、分支屏蔽、添加说明性文字、故障树视图缩放功能; 故障树管理模块主要包括故障树查询、故障树创建、故障树存储、故障树增减、故障树导入与导出(导出为 BMP、JPEG、GIF、TIFF 和 PNG 格式的图像文件)、打印功能。

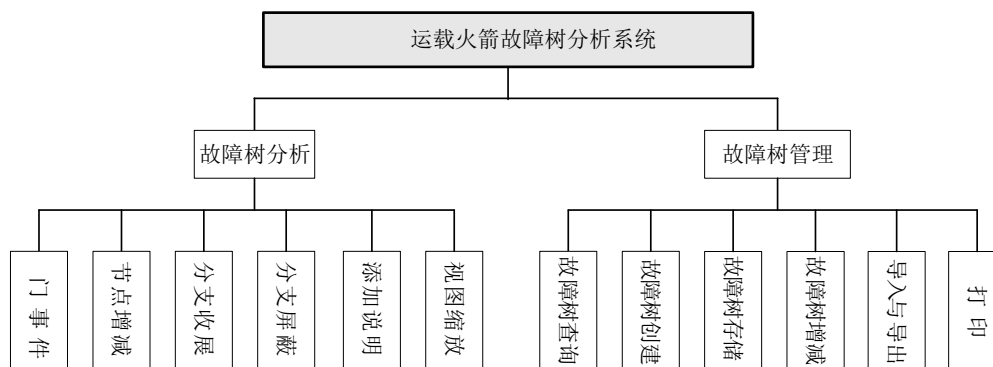


Figure 1. Block diagram of carrier rocket fault trees analysis system  
图 1. 运载火箭故障树分析系统总体设计

## 3. 关键技术

根据图 1 所示的运载火箭故障树分析系统总体设计方案, 软件系统开发涉及三种主要关键技术: 递归算法、C++标准模板库和 GDI+技术。

### 3.1. 递归算法

在数学和计算机科学中, 递归是指一个函数直接或间接调用自身的现象。作为一种算法, 递归在程序设计语言中得到了广泛的应用, 它通常把一个大型复杂的问题层层转化为一个与原问题相似但规模较小的问题来求解。使用递归, 只需少量程序代码就可以描述解决问题所需的多次重复操作, 因此大大减

少了代码量。在编辑或修改故障树时，需要进行查找故障树节点、删除节点分支、计算各节点外围矩形尺寸，及保存和载入故障树等操作，这就涉及到故障树节点的遍历问题。容易理解对于这种树形数据结构，使用递归遍历是一种行之有效的方法[5] [6] [7] [8]。

下面以故障树的保存和载入为例，说明递归算法的使用。如图 2 所示，当需要将故障树保存到永久存储介质中时，首先需判明故障树是否存在，这取决于故障树文档根节点指针是否有效。若故障树不存在，将标记数“0”保存；否则保存标记数“1”。在故障树存在的情况下，保存标记数“1”后将根节点序列化，之后进入递归算法处理：每次递归过程中，都先写入子节点数目 N，接着对 N 进行判断：若 N 为 0，返回到上一递归处理层；否则将当前子节点下的所有子节点序列化，并进一步对其所有子节点进行相同的遍历，也就是进入嵌套的递归处理过程。待全部节点序列化完毕后，故障树保存工作结束。

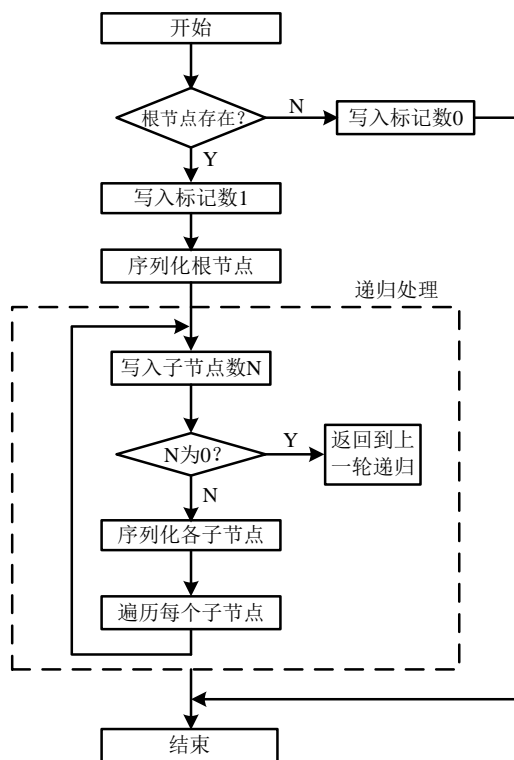


Figure 2. Fault tree saving on a recursive algorithm  
图 2. 递归算法保存故障树

从永久存储介质中载入故障树与保存故障树的过程相对应：首先读入故障树标记 bStart。若 bStart 为 0，说明载入的是空故障树，流程直接返回；在 bStart 为 1 时，先序列化根节点 m\_pTrunk，之后进入递归载入过程。每次递归过程都先判断当前节点下是否有子节点——没有子节点时流程返回到上层递归，否则进入嵌套递归处理，以便对当前子节点的所有子节点序列化。待全部节点序列化完毕，标明故障树已成功读入，接下来就可以对故障树实施查看、编辑、导出等操作了。

### 3.2. C++标准模板库技术

C++标准模板库(Standard Template Library, STL)是一个与标准类与函数模板相关的库，利用它可以创建用途广泛、功能强大的组织数据的通用类及函数，可以大大简化很多软件开发中的编程工作。STL 是用 C++标准定义的，因此总可以在合适的编译器中使用到它们。具体地讲 STL 包括容器、容器适配器、

迭代器、算法、函数对象和函数适配器等六种组件[9] [10]。鉴于运载火箭故障树分析系统每个故障树节点下的子节点数目都可动态增减，因此将数据结构取为链表形式是合理的。STL 中的 `list<T>` 容器模板封装了双向链表，使用它可带来的优点是当从列表序列中的任意位置插入或删除元素时所用的时间均相同。

#### 1) 创建列表容器双向链表

语法：`list<shared_ptr<CNode>> m_listNodes;`

其中 `CNode` 代表自定义故障树节点对象，`m_listNodes` 为要创建的双向列表容器对象，列表中元素类型为指向 `CNode` 类对象的 `shared_ptr` 类型的智能指针。智能指针是封装裸指针(即通常意义下的指针)的模板类型的对象，智能指针的行为类似于裸指针，但二者有区别——前者是智能的，譬如在分配堆内存时使用了智能指针，那么智能指针就会负责在内存使用后释放它们，这意味着代码中不再需要使用 `delete` 关键字。也就是说使用 `shared_ptr` 或 `unique_ptr` 智能指针替代裸指针，可省去释放内存的操作，从而消除了内存泄漏的风险，增加了程序的健壮性和可维护性。

#### 2) 往节点中添加子节点

语法：`m_listNodes.push_back(std::shared_ptr<CNode>& pNewNode);` 这里智能指针 `pNewNode` 指向要添加到列表容器尾部的子节点对象。

#### 3) 删除子节点

语法：`m_listNodes.remove(std::shared_ptr<CNode>& pDelete);` 其中智能指针 `pDelete` 指向要从列表中删除的子节点对象。

#### 4) 获得子节点数目

语法：`m_listNodes.size();`

返回列表中元素数目，即故障树中某节点下的子节点数。

#### 5) 遍历列表

在 `for` 循环中使用 `iterator` 迭代器可以方便地遍历列表，语法如下：

```
for(list<shared_ptr<CNode>>::iterator iter = begin(
    pNode->GetListNodes());
    iter != end(pNode->GetListNodes()); iter++)
{
    .....
}
```

其中 `iter` 为迭代器，`GetListNodes` 函数返回列表对象。`begin` 和 `end` 为 STL 中提供的全局函数，它们分别返回指向列表中第一个元素和最后一个元素下一个位置的双向迭代器。

### 3.3. GDI+技术

在 Windows 操作系统中，绝大多数具有图形界面的应用程序都离不开 GDI。GDI 是 Graphics Device Interface 的缩写，其含义是图形设备接口，它的主要任务是负责操作系统与绘图应用程序之间信息的交换，处理所有 Windows 程序的图形输出。利用 GDI 中提供的大量函数，可以方便地在屏幕、打印机及其它设备上输出图形、文本等。GDI 使程序员无需关心硬件设备及其驱动程序是什么就可以将应用程序的输出转化为硬件设备上的输出，从而实现程序开发者与硬件设备间的隔离，大大方便了软件开发工作[5]。

GDI+是一组通过 C++类实现的应用程序编程接口，是对 GDI 的更高层次的封装和对原有 GDI 在功能上的升级。它对 GDI 进行了优化并添加了许多新的功能，这些新功能涉及渐变画刷、基本样条函数、独立的路径对象、变换矩阵对象、可伸缩的区域、Alpha 混合和多种图像格式支持等。使用 GDI+技术，

图形和艺术字将具有更加平滑的轮廓以及可调整的用真实颜色混合的透明度级别。用 GDI+提供的诸多方法可以直接调用显示器或打印机，真正具备了与设备无关的特点，图 3 给出 GDI+的体系。

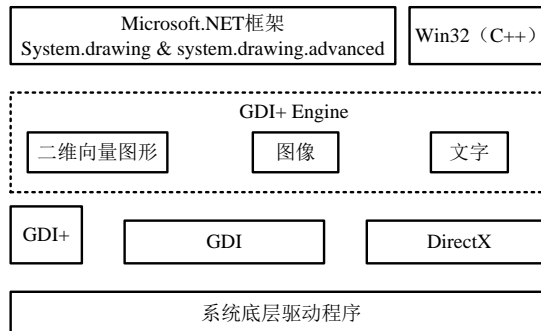


Figure 3. The GDI+ system

图 3. GDI+体系

为在运载火箭故障树分析软件系统使用 GDI+技术，必须在开始时调用 GdiplusStartup 函数，并在不需要 GDI+时调用 GdiplusShutdown 函数。在 GDI+提供的众多类对象中，Graphics 对象是最重要的，它封装了大量方法，其地位与作用与 GDI 中设备上下文句柄 HDC 相当。为将故障树导出为图像格式的文件，首先需要利用 GetImageDecodersSize 和 GetImageDecoders 这两个函数确定 GDI+所支持的图像解码器的类型。解码器的类型包括“image/bmp”、“image/jpeg”、“image/gif”、“image/tiff”、“image/png”，它们分别对应于 BMP、JPEG、GIF、TIFF、PNG 五种图像格式。将故障树保存为图像格式的文件时，使用的是直接派生于 Image 的 Bitmap 类对象提供的 Save 方法：

```
Bitmap::Save(const WCHAR* filename, const CLSID* clsidEncoder, const EncoderParameters* encoderParams);
```

其中 filename 为要保存的图像文件的文件名；clsidEncoder 标识了保存图像所用的解码器类型：保存成 BMP 图像时为“image/bmp”，保存成 JPEG 图像时为“image/jpeg”，以此类推；encoderParams 常为 NULL。如果能成功保存图像文件，Save 函数就返回 0，否则返回的是一个与具体错误相关的非零值。

## 4. 运载火箭故障树分析系统软件开发与应用

### 4.1. 开发环境选择

面向对象程序设计(Object Oriented Programming, OOP)兴起于上世纪 90 年代，现已是软件开发的主流方法。实际上面向对象的概念和应用已超越了程序设计和软件开发的范畴，扩展到很宽的范围如数据库系统、交互式界面、应用结构、应用平台、分布式系统、网络管理结构、CAD 技术、人工智能等领域。使用面向对象方法开发软件，可提高代码的重用性，增强软件的可靠性和可维护性。

根据运载火箭故障树分析系统软件需求，以图 1 总体设计方案和上述三种关键技术为基础，基于 Visual Studio 2010 软件开发平台，采用 Visual C++语言和面向对象程序设计技术开发。

### 4.2. 软件设计

系统功能设计如图 4 所示，包括故障树创建、门事件开发、故障树编辑与修改、保存、打印、图像导出、动态视图缩放、故障树库查询与维护等功能模块。其中，保存和载入故障树使用递归算法实现；对故障树创建、编辑和修改使用 C++标准模板库技术；将故障树以 BMP、JPEG、GIF、TIFF、PNG 五种格式输出为图像文件时使用 GDI+技术实现，三者的具体实现过程已在前述关键技术部分给出。

软件界面采用用户熟悉的 Windows 风格设计，界面包括菜单栏、工具栏和故障树编辑区域。菜单栏包括“文件”、“编辑”、“视图”、“插入”、“数据库”、“窗口”、“帮助”等 7 个功能，各菜单中还包含相应的子菜单，为节省篇幅不予一一介绍。工具栏包括常见的新建文件、打开文件、保存文件、视图缩放、各种门事件功能按钮及其它辅助操作选项。故障树编辑区域是用户主要的操作区域，对故障树的各种操作都主要在此区域中进行。

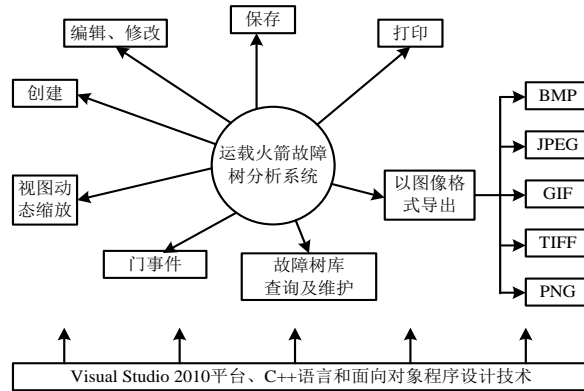


Figure 4. The functionality of carrier rocket fault trees analysis system  
图 4. 运载火箭故障树分析系统的功能

### 4.3. 系统应用与性能评估

过去，某航天发射场火箭故障诊断时基本采用手工创建故障树的方法，自采用故障树分析系统软件以来，故障分析时间与原来相比减少了一半以上，从而提高了故障诊断效率，同时故障诊断的可靠性也提高了。

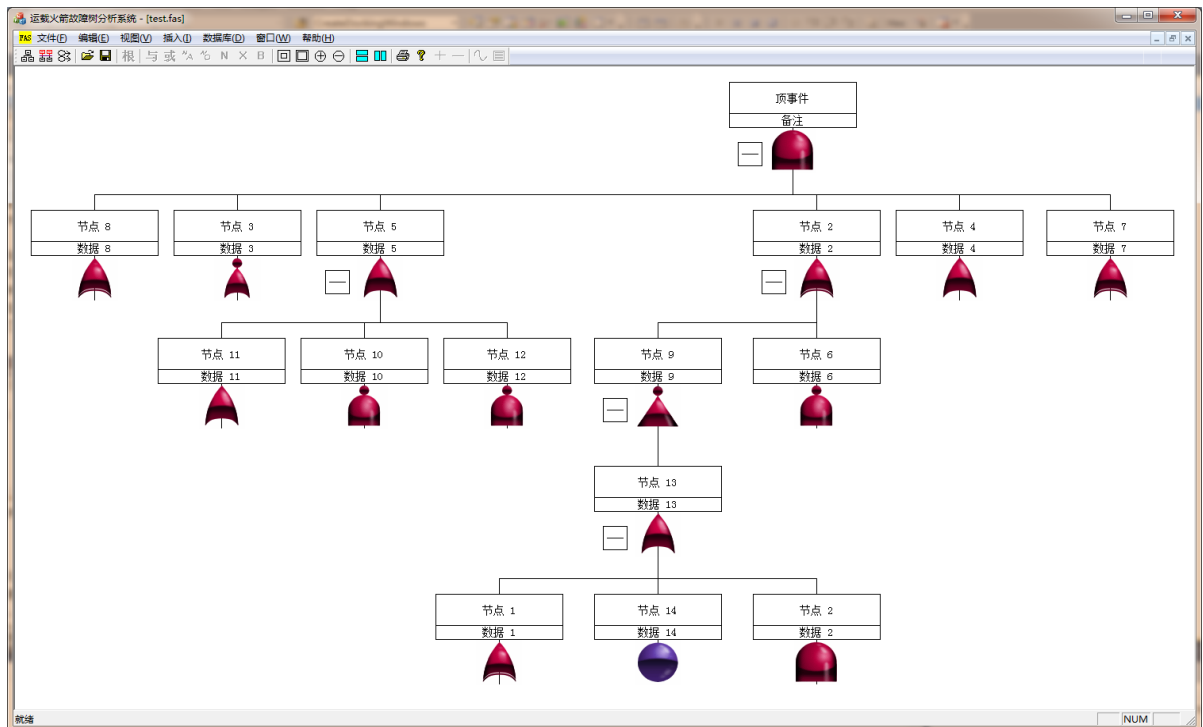


Figure 5. The application of a carrier rocket fault trees analysis system  
图 5. 运载火箭故障树分析系统应用举例

这里以运载火箭故障树分析系统在“火箭飞行弹道异常”故障诊断中的应用为例,说明对该软件系统性能的评估情况,图5为编辑该故障树时的情形。从图中可以看出,该故障树文件名称为 test.fas,位于软件窗口标题栏上。软件界面包含文件、编辑、插入、数据库、窗口等菜单,还包括一个快捷工具条。故障树则由一个顶事件、三个中间层事件和三个底事件组成。

利用图5所示功能,“火箭飞行弹道异常”故障诊断过程中故障树的建立、编辑和修改可以在10分钟内完成,同时可以将修改后的故障树导出为前述五种格式的图像文件。克服了过去人工建立故障树时存在的耗时长、效率低、不易维护的缺点和不足,证明了运载火箭故障树分析软件系统的有效性。

## 5. 结束语

针对某航天发射场高密度、常态化、多任务并行的工作特点,为解决故障诊断中人工建立故障树存在的速度慢、效率低、不易维护的缺点,基于面向对象软件设计模式,采用递归算法、STL和GDI+关键技术设计开发了运载火箭故障树分析软件系统。除具有常规的建立故障树的功能及对故障树库管理和维护外,故障树分析系统还能将故障树以BMP、JPEG、GIF、TIFF、PNG五种格式输出为图像文件,提高和拓展了软件系统在故障诊断中的应用范畴。对该系统的性能评估结果表明,该系统满足设计需求,目前已投入使用。特别地,运载火箭飞行试验智能故障诊断所处的地位日益重要,本系统为火箭飞行试验分析评估与智能故障诊断提供了良好的技术支撑。对决策者快速、准确、有效地定位故障,对运载火箭的正常飞行具有较为重要的意义。

## 参考文献

- [1] Chen, G.Z., Wu, Y.F., Fu, L. and Bai, N. (2018) Fault Diagnosis of Full-Hydraulic Drilling Rig Based on RS-SVM Data Fusion Method. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, **40**, 1-11. <https://doi.org/10.1007/s40430-018-1067-y>
- [2] Qu, Y.Z., Zhang, Y., He, M., et al. (2019) Gear Pitting Fault Diagnosis Using Disentangled Features from Unsupervised Deep Learning. *Proceedings of the Institution of Mechanical Engineers*, **233**, 719-730. <https://doi.org/10.1177/1748006X18822447>
- [3] 张保山, 周峰, 张琳, 等. 基于最优神经网络结构的故障诊断模型[J/OL]. 兵器装备工程学报: 1-6. <http://kns.cnki.net/kcms/detail/50.1213.TJ.20190905.1651.014.html>, 2019-09-05.
- [4] 吕琛, 栾家辉, 王立梅. 故障诊断与预测——原理、技术及应用[M]. 北京: 北京航空航天大学出版社, 2012.
- [5] 杨硕, 周霜菊, 张志杰. 基于状态机的递归算法非递归化框架[J]. 计算机应用与软件, 2018, 35(4): 122-128.
- [6] Liberios, V., Zuzana, B. and Daniel, M. (2017) Hanoi Towers in Resource Oriented Perspective. *15th International Symposium on Applied Machine Intelligence and Informatics*, Herl'any, Slovakia, 26-28 January 2017, 179-184.
- [7] Cheng, Y. and Zou, D. (2019) Complementary Ensemble Local Means Decomposition Method and Its Application to Rolling Element Bearings Fault Diagnosis. *Proceedings of the Institution of Mechanical Engineers*, **233**, 868-880. <https://doi.org/10.1177/1748006X19838129>
- [8] Bukhman, A.K., Nsengimana, V.J.P., Lipsitz, M.C., et al. (2019) Diagnosis and Management of Acute Heart Failure in Sub-Saharan Africa. *Current Cardiology Reports*, **21**, 120. <https://doi.org/10.1007/s11886-019-1200-2>
- [9] 刘泽潮, 张兵, 易彩, 等. 高阶频率加权能量算子在列车轴箱轴承故障诊断中的应用[J/OL]. 西安交通大学学报: 1-10. <http://kns.cnki.net/kcms/detail/61.1069.T.20190904.1156.004.html>, 2019-09-05.
- [10] 闵军, 罗泓. C++11 实现可变参数泛型抽象工厂[J]. 软件工程, 2017, 20(5): 18-22.