文章编号:1671-4598(2020)01-0011-05 DOI:10.16526/j.cnki.11—4762/tp.2020.01.004 中图分类号:V1 文献标识码:A

基于 SPARC V8 的星载嵌入式软件全数字仿真平台设计与实现

张 涛,李瑞军,范延芳
(北京空间飞行器总体设计部,北京 100094)

摘要:为了提高星载嵌入式软件的可靠性和安全性,解决硬件测试环境构建困难、成本昂贵以及运行状态难以监控的局限性,提出了一种基于 SPARC V8 的星载嵌入式软件全数字仿真平台设计和实现方法;介绍了全数字仿真平台实现的关键技术,包括 CPU 指令集仿真、寄存器仿真、存储器仿真、中断控制器仿真、串口仿真、定时器仿真、虚拟外设模型仿真以及设备管理器和平台时序设计;全数字仿真平台与基于硬件的测试平台相比具有可重用性强、可快速搭建、成本低廉、高可控性、调试和测试手段丰富、支持故障注入等优点;该全数字仿真平台已在星载嵌入式软件型号研制中得到了应用,基于此平台可快速搭建虚拟目标机和虚拟外设环境,进行星载嵌入式软件运行仿真、调试验证等工作。

关键词:SPARC V8;嵌入式软件;全数字仿真

Design and Implementation of a Full-digital Simulation Platform for on-board Embedded Software Based on SPARC V8 Architecture

Zhang Tao, Li Ruijun, Fan Yanfang
(Beijing Institute of Spacecraft System Engineering, Beijing 100094, China)

Abstract: In order to improve the reliability and security of on-board embedded software, and to solve the limitations of hard to build, expensive and difficult to monitor the running state of the hardware test environment, this paper presents the design and implementation of a full-digital simulation platform for on-board embedded software based on SPARC V8 architecture. The key technologies of the digital simulation platform are introduced, including CPU instruction simulation, register simulation, interrupt controller simulation, serial port simulation, timer simulation, virtual peripheral model simulation and design of device management and platform timing. Compared with the hardware-based test platform, the full-digital simulation platform has the advantages of strong reusability, fast construction, low cost, high controllability, abundant debugging and testing means, support fault injection, etc. It has been used in the development of embedded software on satellite. Based on this platform, the virtual environment can be built quickly, and the embedded software can be simulated and tested.

Keywords: SPARC V8; embedded software; full-digital simulation

0 引言

近年来,随着航天技术的快速发展,星载嵌入式软件在型号中发挥着越来越重要的作用。空间站、深空探测等为代表的复杂航天器呈现出智能化、网络化的特点,越来越多的功能将通过软件实现,信息流、控制流交互复杂,任务、中断、IO 等时序要求及其严苛。面对软件研制任务激增、研制周期缩短的挑战和软件高可靠性和安全性的需求,传统的基于硬件的星载嵌入式软件测试已逐渐不能适应新形势的要求,存在以下问题:

1)硬件设备定制开发,致使型号研制成本高;软件测试设备都是根据各个型号定制研发的,不同系列型号间难以复用;同时由于型号内许多人员都有需求,重点型号和领域有时还需要配备多套设备,致使型号研制的成本非常高。

2)星载嵌入式软件测试依赖于硬件研制进度,无法实现软硬件同步开发和早期验证。航天器星载软件测试严重受制于航天器单机设备和地面测试设备的研制,测试启动较晚,导致许多接口和时序问题层层闯关,直到分系统级或者整星级测试的最后阶段才暴露出来,对型号研制进度产生极大影响。

3)传统的基于硬件的软件测试环境难以对运行状态进行控制和监视、故障注入困难,无法实现对各种飞行工况和异常工况的测试覆盖。

全数字仿真平台针对真实硬件设备生产设计周期长,成本昂贵、使用资源紧张、状态监视和故障注入困难等缺陷,利用软件仿真技术,对真实的星载计算机进行虚拟化建模,从而逼真的模拟硬件目标系统[1]。原来运行于星载计算机的嵌入式软件,可以不加修改直接在虚拟平台上运行,并且其运行的动态特性与真实目标机上一致。利用全数字仿真平台,可以在不具备目标硬件的情况下,进行星载嵌入式软件开发、调试、测试和验证。

SPARC V8 (scalable processor ARChitecture version 8)是采用精简指令集的 CPU 指令集架构,其设计的指令

收稿日期:2019-06-27; **修回日期**:2019-07-30。

作者简介:张 涛(1986-),男,河北保定人,硕士研究生,主要从事星载软件及其地面测试工具的研发工作方向的研究。

集能提高执行的效率以及优化编译器生成的代码，从而使代码执行起来更为高效、快速[2]。基于 SPARC V8 架构的处理器被广泛应用于航天领域。运行于 SPARC V8 处理器的星载软件在实时嵌入式操作系统的支持下，完成卫星遥测、遥控、总线通信、姿轨控和自主管理等功能。本文提出了一种基于 SPARC V8 的全数字仿真平台设计和实现方法，能够在不具备硬件环境的情况下，对星载嵌入式软件进行调试、测试和仿真验证。

1 全数字仿真平台架构设计

全数字仿真平台架构如图 1 所示。主要包括 ECLIPSE 集成开发环境、虚拟目标机、虚拟外设模型、设备管理器以及其他辅助工具和配置文件。

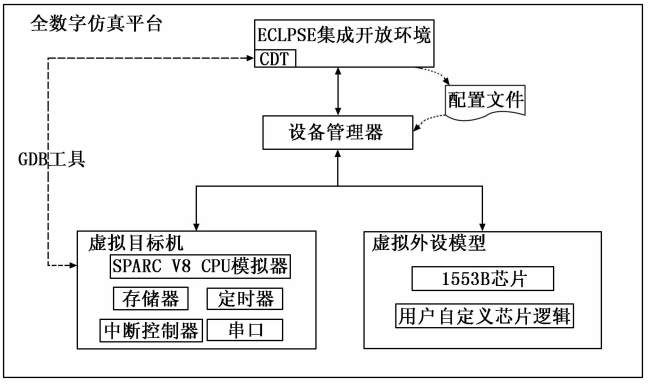

图 1 全数字仿真平台架构

集成开发环境基于 ECLIPSE 平台，结合 CDT（C Developing Toolkit）插件、调试插件和模拟器插件，结合编译器、调试器等工具，提供了模拟器配置、目标应用工程管理、目标应用源代码编辑、可配置编译和快捷调试、仿真器运行控制等功能。

底层的虚拟目标机与虚拟外设模型构成了功能趋近硬件的虚拟嵌入式硬件模型，提供了目标应用程序的仿真执行环境。虚拟目标机支持 SPARC V8 CPU 处理器的指令集仿真，同时对 CPU 寄存器、存储器、定时器、中断控制器和串口等内部结构进行仿真。虚拟外设模型包括通用外设和用户自定义外设，通用外设一般包含 1553B 总线，659 总线芯片，用户自定义外设一般包括遥测、遥控等专用 FPGA 逻辑芯片。CPU 模型还内嵌了 GDB 协议，与 ECLIPSE 的 CDT 插件调试模块和目标架构的 GDB 工具组成完整的调试环境。

作为集成开发环境与底层模型的适配器，设备管理器收集 IDE 提供接口、自身提供接口以及虚拟 CPU 提供接口，向上对 IDE 提供模拟器控制接口，解析模拟器配置文件；向下可依据配置信息调用并管理相应虚拟设备，并为虚拟设备提供统一的模型接口。

2 虚拟目标机设计与实现

虚拟目标机是全数字仿真平台的核心，为目标程序测试和验证提供虚拟的硬件平台。虚拟目标机通常包括 SPARC V8 CPU 模拟器、存储器仿真、片上外设如中断控制器、定时器等模型仿真。

2.1 SPARC V8 指令集模拟器

SPARC V8 CPU 模拟器使用了动态二进制翻译器翻译目标指令到本地 X86 指令，并在指令执行时按照指令的功能操作相应的寄存器和触发中断或异常。

2.1.1 寄存器模拟

虚拟 CPU 内核内部维持了一个处理器状态结构体变量（记为 CPUSTAT），成员包括寄存器变量定义和其他辅助变量定义。在虚拟 CPU 内核初始化时保存变量 CPUSTAT 中各目标寄存器变量的偏移；在目标代码执行开始时将变量 CPUSTAT 指针存入固定的本地寄存器（记为 LRx）中，LRx 不会再被用作它途；在需要使用目标寄存器（记为 TR）时，以 LRx 寄存器值为基地址加上保存的目标寄存器的偏移，则获取目标寄存器在本地内存中的地址，就可取得目标寄存器的值或向目标寄存器赋值。

2.1.2 指令翻译与执行

虚拟 CPU 内核使用动态二进制翻译技术将 SPARC V8 的指令翻译成本地 X86 指令，包括目标代码到中间代码的前端翻译和中间代码到本地代码的后端翻译两个翻译过程，如图 2 所示。

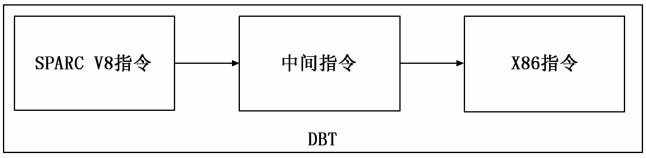

图 2 动态二进制翻译执行

动态二进制翻译以基本块为翻译和执行的基本单位，其中基本块为一段一般以跳转指令为结尾的一段目标代码；采用边翻译边执行的策略，只有在执行时翻译代码缓存中未发现待执行基本块才进行翻译和缓存；前端翻译中还进行内存管理和异常处理等。

2.2 片上外设模拟

2.2.1 存储器模拟

存储器的模拟是将虚拟目标存储器直接绑定到相同大小的本地内存上，虚拟目标机访问存储器实际是访问绑定的本地内存区域。同时，作为采用大端模式的处理器，在访问存储器数据时需要进行字节序的调整。

2.2.2 中断模拟

中断和陷阱是两种异常处理机制。陷阱是由与特定指令相关的硬件引起的异常，在引起异常产生的指令运行期间触发；中断是由处理器的外部事件产生并在指令执行期间发生。

对于陷阱的模拟是在相关指令翻译时插入陷阱条件判断逻辑，并在执行时依条件触发陷阱，陷阱触发后虚拟 CPU 内核进入异常处理逻辑；对于中断的模拟是通过置标志位使基本块退出执行，虚拟 CPU 内核进入异常处理逻辑。虚拟 CPU 内核进入异常处理逻辑后，依照硬件逻辑操作相关寄存器并根据特定的异常向量表完成程序的跳转。

虚拟中断控制器在初始化时以内存映射 MMIO（MEMORY－MAPPED IO）方式映射寄存器到虚拟 CPU 内核地址空间的指定地址区域，在虚拟 CPU 内核访问虚拟中断控制器的寄存器时触发相应的寄存器处理逻辑。同时，分配虚拟中断输入引脚和中断输出引脚，并注册中断信号输入回调。虚拟中断输入引脚根据硬件中断引脚连接分别接入虚拟片上外设中断引脚或虚拟外部中断引脚，虚拟中断输出引脚分别接到虚拟 CPU 内核的中断引脚，如图 3 所示。当中断输入信号到来时触发注册的中断信号输入回调函数，根据寄存器编程配置进行中断信号选择输出到虚拟 CPU 内核。

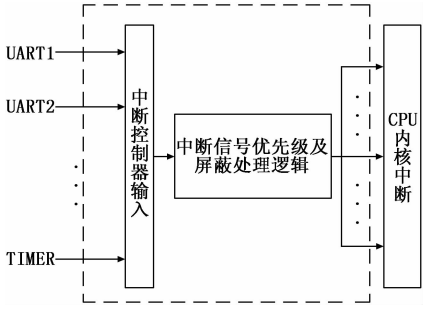

图 3　中断控制器结构示意图

2.2.3　定时器模拟

定时器的寄存器模拟使用虚拟 CPU 内核中的 MMIO 映射机制实现。当虚拟 CPU 内核访问到该虚拟定时器的寄存器地址区域时触发 MMIO 的访问回调，在回调函数中处理寄存器的访问逻辑。

定时器的中断模拟是在虚拟定时器创建时创建一个虚拟中断引脚，并将其连接到虚拟中断控制器相应引脚上。当虚拟定时器需要触发中断时，向虚拟中断引脚设值则会触发虚拟中断控制器的中断信号输入回调函数。

定时器的计时功能模拟依靠虚拟 CPU 内核中的精确减计数器机制实现。虚拟 CPU 内核中维护了一个减计数器集合，创建或删除虚拟定时器时会操作该集合增加或删减。虚拟 CPU 内核中有独立的线程循环检查减计数器是否计数到 0，到期后则触发创建定时器时注册的回调函数。

2.2.4　串口模拟

串口的寄存器模拟使用虚拟 CPU 内核中的 MMIO 映射机制实现。当虚拟 CPU 内核访问到该虚拟串口的寄存器地址区域时触发 MMIO 的访问回调，在回调函数中处理寄存器的访问逻辑。

串口的中断模拟是在虚拟串口创建时创建一个虚拟中断引脚，并将其连接到虚拟中断控制器相应引脚上。当虚拟串口需要触发中断时，向虚拟中断引脚设值则会触发虚拟中断控制器的中断信号输入回调函数。

串口的数据流可以定向到虚端口，也可基于虚拟 CPU 内核提供字符驱动选择定向到本地虚拟串口、TCP 端口或 UDP 端口。当数据流定向到虚端口时，可以在虚端口的连接端进行数据交互；定向到其他本地端口时，可以使用本地辅助工具进行数据交互。

本地端口作为数据流传输方式时，发送数据通过相应本地端口数据发送接口发送。当循环检查线程检测到本地端口有数据时，会向虚拟串口询问是否接收数据，允许后将接收的数据返回给虚拟串口，虚拟串口操作相关寄存器并根据用户编程配置决定是否向终端控制器投递中断信号。

2.3　CPU 内核与片上外设通信机制

对于虚拟 CPU 内核主动要求外设动作的通信，虚拟 CPU 内核访问外设寄存器时触发寄存器上的 MMIO 回调函数，在回调函数中实现外设的功能。

对于某些外设事件发生要通知虚拟 CPU 内核，虚拟 CPU 内核会主动循环检查。虚拟 CPU 内核内部使用循环检查线程来检查虚拟定时器到期和虚拟串口数据到来等事件，检测到这些事件后通知相关虚拟外设进行相应的处理，如寄存器改变和触发中断等操作。虚拟 CPU 内核通过查询相关寄存器或响应中断得知事件源并进行相应处理。

3　虚拟外设模型设计与实现

虚拟外设模型用来搭建虚拟目标机的外围仿真环境，实现虚拟目标机与外设模型的数据交互。为了保证被测软件的真实性，虚拟外设模型的逻辑功能必须与硬件逻辑功能保持一致：即虚拟外设模型接收到虚拟目标机的输入输出请求时，需要根据硬件逻辑改变自己的状态。虚拟外设模型可为虚拟目标机提供输入激励，如模拟上行遥控注入，也可以接收虚拟目标机输出数据，如接收下行遥测输出。因此，虚拟外设模型建模是全数字仿真平台开发的一个重要环节。

3.1　虚拟外设模型的通用接口设计

虚拟外设模型仿真接口作为虚拟目标机的扩展媒介，需要尽可能简便直观地将虚拟 CPU 内核与外设模型间的不同操作隔离。模型仿真采用单个函数以消息类型方式分隔操作，提高接口的扩展性、易使用性，也保证各接口的单一性。

虚拟外设模型导出函数接口如下：

MainProc（MESSAGE Msg，UINT32 uParam）

其中第一个参数 Msg 是消息类型，代表不同的操作，共有以下几种消息类型：

1）SETPROPERTY 消息用于设备管理器解析模拟器配置文件中的外设模型的属性；

2）INIT 消息用于设备管理器通知外设模型进行端口创建等初始化操作；

3）RESET 消息用于设备管理器通知外设模型复位操作；

4）READ/WRITE 消息用于设备管理器通知外设模型虚拟 CPU 内核正在进行 uParam 参数指定的 IO 读/写操作；

5）PINCONNECT 消息用于设备管理器通知外设模型虚拟引脚有电平信号输入；

6）VWRITE 消息用于通知外设模型大量逻辑数据输入；

7）CLOCK 消息用于通知外设模型 uParam 指定的时钟定时到期；

8）TERMINATE 消息用于通知外设模型虚拟目标机

即将终止，进行终止前的资源释放等操作。

3.2 虚拟外设模型端口设计

虚拟外设模型端口是全数字仿真系统中 CPU 内核与其他设备模型通信以及模型间通信的接口，包括 IO 端口、虚端口和 PIN 引脚。

3.2.1 IO 端口设计

IO 端口是虚拟外设模型与虚拟 CPU 内核间的通信媒介。IO 端口的设计包括 3 个部分：

1）IO 端口初始化。在模型初始化时，IO 端口初始化完成端口 ID、名称和类型初始化。

2）IO 端口配置。设备管理器会解析模拟器配置文件中 IO 端口配置，创建 MMIO 并在访问回调函数中触发 READ、WRITE 消息。IO 端口配置默认宽度为 4 字节，可配置其他尺寸表示寄存器区域。

3）IO 端口访问。虚拟 CPU 内核在访问 IO 端口所处地址区域时会触发 READ、WRITE 消息，在消息回调中进行 IO 的逻辑处理。

3.2.2 虚端口设计

虚端口是设备模型间大量数据的通信媒介。虚端口的设计包括 3 个部分：

1）虚端口初始化。在模型初始化时，虚端口初始化完成端口 ID、名称和类型初始化。

2）虚端口连接配置。设备管理器会解析模拟器配置文件中虚端口连接配置，保存虚端口连接关系。虚端口支持一对多连接配置。

3）虚端口写操作。在设备模型进行虚端口写操作时，设备管理器会依据虚端口连接关系向所有连接端虚端口写入数据。

3.2.3 PIN 引脚连接设计

引脚是设备模型间电平信号的通信媒介。引脚的设计包括 3 个部分：

1）引脚初始化。在模型初始化时，引脚初始化完成 ID、名称和类型初始化。

2）引脚连接配置。设备管理器会解析模拟器配置文件中引脚连接配置，保存引脚连接关系。引脚支持一对多连接配置。

3）引脚写操作。在设备模型进行引脚写操作时，设备管理器会依据引脚连接关系向所有连接端引脚写入电平信号。

4 设备管理器设计和平台时序调度

当全数字仿真平台启动时，设备管理器会解析模拟器配置文件，根据解析的信息启动相应的模拟器并执行指定的目标应用。设备管理器在模拟器启动时的主要工作如下：

1）解析全数字仿真平台构建所需的模拟设备，加载相应设备 DLL。

2）解析模型内存、IO、虚端口、引脚配置，保存所有 IO、虚端口和引脚连接关系。

3）以 RESET 消息复位 CPU 及各虚拟外设。

全数字仿真平台启动后，设备管理器在模拟器运行中的主要工作如下：

1）在虚拟 CPU 访问 IO 端口时，设备管理器创建 IO 端口时的回调函数会触发，在回调函数中向 IO 端口所属设别发送 READ/WRITE 消息。

2）在虚拟 CPU 或外设模型拉动引脚时，设备管理器查找引脚连接关系向连接端设备引脚发送 PINCONNECT 消息。

3）在虚拟 CPU 或外设模型写虚端口时，设备管理器查找虚端口连接关系向连接端虚端口发送 VWRITE 消息。

4）当虚拟外设模型创建的定时器到期时，设备管理器在定时器到期回调函数中以 CLOCK 消息通知所属设备到期定时器 ID。

为了尽可能保证虚拟目标机、虚拟外设间时序统一并与真实硬件时序保持一致，虚拟 CPU 内核时钟使用指令统计周期作为基础。虚拟 CPU 核在翻译指令时一般以跳转指令为结尾作为一个基本块，在每个基本块翻译时将基本块内所有指令的执行周期、取指周期和其他周期求和计算出基本块的总时钟周期并保存。在目标代码执行时，对所有执行的基本块进行时钟周期求和，再结合 CPU 内核的主频即得到所有执行的目标代码耗时，即得到模拟时间。

虚拟 CPU 核内部的减计数器功能就是基于模拟时间进行计数的，减计数器 Tick 为 1ns。在减计数器创建时，根据目标计数值和当前模拟时间设置到期计数值，并安装计数到期回调函数。虚拟 CPU 核使用循环检查方式检查当前模拟时间是否达到减计数器到期计数值，达到后触发到期回调函数进行到期逻辑处理。如果减计数器是单次计时则到期后就删除，否则减计数器的计数值复位并重新开始计数。仿真平台内部虚拟设备的定时功能均使用减计数器实现，这就保证了所有虚拟设备的时间是统一的。

5 应用实例

本文设计与实现的全数字仿真平台已在航天器多个型号的星载软件测试中得到应用，该平台可作为 SPARC V8 处理器的模拟运行平台，支持 RTEMS 和 VxWorks 操作系统，同时提供良好的人机界面，便于用户快速搭建虚拟目标机和虚拟外设环境，进行星载嵌入式软件运行仿真、调试验证等工作。

通过测试程序在硬件实物上运行结果与全数字仿真平台上的运行结果，遥测、遥控、总线通信等功能均正确仿真，表 1 为某个进程相同断点下目标机与全数字仿真环境寄存器和内存的值，两者保持一致，全数字仿真平台运行结果正确。

6 结束语

本文设计并实现了一个基于 SPARC V8 的星载嵌入式软件全数字仿真平台，该平台与基于硬件的测试平台相比具有可重用性强、可快速搭建、成本低廉、高可控性、调试和测试手段丰富、支持故障注入等优点。该全数字仿真平台已在星载嵌入式软件型号研制工作中得到了应用，可大

表 1　全数字仿真与目标机运行结果比对

寄存器/内存	全数字仿真平台	硬件平台
PSR	0xb39210c4	0xb39210c4
TBR	0x40000160	0x40000160
I0	2	2
O0	24	24
L0	0xb39210c4	0xb39210c4
G0	0	0
0x40011200	12	12
0x40011204	13	13

大提高软件的可靠性和安全性，从而保证型号软件的质量，推动航天事业的发展。

参考文献：

[1] 罗天成. 飞控系统虚拟仿真平台的研究与设计 [D]. 成都：电子科技大学，2013.

[2] 柳金龙. 基于 SPARC 处理器的虚拟飞控计算机的设计与实现 [D]. 成都：电子科技大学，2015.

[3] 王文杰，杨　坤，姜　强，等. 基于虚拟技术的飞控系统试验平台 [A]. 2015 中国无人机系统峰会论文集 [C]. 2015.

[4] 秦旭东. 飞控系统虚拟设计环境 [J]. 北京航空航天大学学报，2005 (12)：1274－1278.

[5] 关小川，陈朝晖，党纪红. 快速 SPARC V7 指令集模拟器的设计方法 [J]. 计算机工程与设计，2011 (32)：531－534.

[6] 李　勃，等. 嵌入式虚拟协同开发环境的设计与实现 [J]. 计算测量与控制，2011 (4)：931－934.

[7] 陈定君，郭晓东，张应辉，等. 嵌入式软件仿真开发系统的研究 [J] 电子学报，2000，28 (3)：137－139.

[8] 陈宗基，等. 民机飞控计算机系统虚拟样机验证平台研究 [J]. 系统仿真学报，2008 (15)：4057－4063.

[9] 秦旭东. 飞控系统虚拟设计环境 [J]. 北京航空航天大学学报，2005 (12)：1274－1278.

[10] 付鹏宇. 虚拟飞控计算机 SIO 板的建模与实现 [D]. 成都：电子科技大学，2016.

(上接第 10 页)

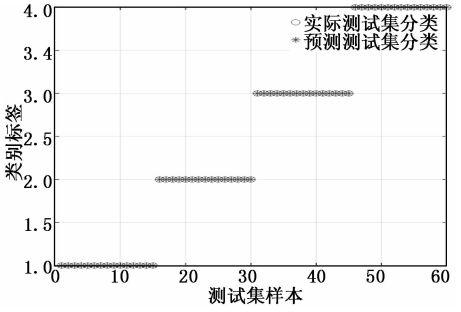

图 3　GASVM 故障诊断结果 (MFE)

5　结束语

本文采用 MFE－GASVM 方法对机载燃油泵系统振动信号进行分析，通过 MFE 提取不同故障状态振动信号的故障特征信息，解决了单尺度 FE 不足以完整概括特征信息的不足，有效提高了故障诊断精度，实现了对实验数据的故障诊断分析。实验结果表明：

1) MFE 能够有效概括振动信号在不同尺度下的特征信息，相比单尺度特征提取方法而言，所得到的故障信息更加完整，有效提高了故障诊断精度。

2) 采用 GA 优化 SVM 的核函数参数及惩罚参数，可以提高 SVM 的自适应性，避免了参数选择问题，提高 SVM 的分类能力。

参考文献：

[1] 焦晓璇，景　博，李　娟，等. 基于 ADEGWO－SVM 的机载燃油泵寿命预测研究 [J]. 仪器仪表学报，2018，39 (8)：43－52.

[2] 李　娟，景　博，焦晓璇，等. 基于 LSTAR 的机载燃油泵多阶段退化建模 [J]. 北京航空航天大学学报，2017，43 (5)：880－886.

[3] 李　娟，景　博，羌晓清，等. 基于样本分位数的机载燃油泵故障状态特征提取及实验研究 [J]. 航空学报，2016，37 (9)：2851－2863.

[4] 吴　参，李兴林，孙守迁，等. 混沌理论在滚动轴承故障诊断中的应用 [J]. 轴承，2013 (1)：60－64.

[5] 李晓娟，曲建岭，邹文栋，等. 航空发动机状态参数的关联维数分析 [J]. 计算机仿真，2013，30 (9)：56－59.

[6] 郑近德，程军圣，杨　宇. 基于 LCD 和排列熵的滚动轴承故障诊断 [J]. 振动. 测试与诊断，2014，34 (5)：802－806.

[7] Pincus S M. Approximate Entropy as a Measure of System Complexity [J]. Proceedings of the National Academy of Sciences of the United States of America，1991，88 (6)：2297－2301.

[8] Richman J S，Moorman J R. Physiological time－series analysis using approximate entropy and sample entropy. [J]. American Journal of Physiology Heart & Circulatory Physiology，2000，278 (6)：2039－2049.

[9] Chen W，Wang Z，Xie H，et al. Characterization of Surface EMG Signal Based on Fuzzy Entropy [J]. IEEE Transactions on Neural Systems and Rehabilitation Engineering，2007，15 (2)：266－272.

[10] 郑近德，姜战伟，代俊习，等. 基于 VMD 的自适应复合多尺度模糊熵及其在滚动轴承故障诊断中的应用 [J]. 航空动力学报，2017，32 (7)：1683－1689.

[11] 杨望灿，张培林，王怀光，等. 基于 EEMD 的多尺度模糊熵的齿轮故障诊断 [J]. 振动与冲击，2015，34 (14)：163－167.

[12] 姚文坡，刘铁兵，戴加飞，等. 脑电信号的多尺度排列熵分析 [J]. 物理学报，2014，63 (7)：427－433.

[13] Tong S，Koller D. Support vector machine active learning with applications to text classification [J]. Machine Learning Research，2002，2 (1)：999－1006.

[14] 陈　博，陈光雄. 基于 MEEMD 和 GA－SVM 的列车车轮多边形故障识别方法 [J]. 噪声与振动控制，2018 (3)：157－161.

[15] 李艳军，张　建，曹愈远，等. 基于模糊信息粒化和优化 SVM 的航空发动机性能趋势预测 [J]. 航空动力学报，2017，32 (12)：3022－3030.

[16] 郑近德，陈敏均，程军圣，等. 多尺度模糊熵及其在滚动轴承故障诊断中的应用 [J]. 振动工程学报，2014，27 (1)：145－151.