

文章编号:2095-6134(2021)03-0402-07

一种基于 RPC 的组态化卫星模拟器故障注入方法^{*}

杨善强^{1,2}, 李华旺^{1,3†}, 常亮¹, 高才栋¹, 虞业洙¹
(1 中国科学院微小卫星创新研究院, 上海 201210; 2 中国科学院大学, 北京 100049; 3 上海科技大学, 上海 201210)
(2019 年 8 月 9 日收稿; 2019 年 12 月 4 日收修改稿)

Yang S Q, Li H W, Chang L, et al. RPC-based configuration satellite simulator fault injection method[J]. Journal of University of Chinese Academy of Sciences, 2021, 38(3): 402-408.

摘 要 本着提高卫星故障注入模型的可复用性及缩短开发周期的目的, 为卫星仿真系统设计一种标准层次结构模型, 由卫星模拟器、故障注入前端和故障协议组成。卫星模拟器中的单机由通用单机模型配置而成, 实现卫星模拟器的组态化。故障前端通过 RPC(远程过程调用)调用模拟器中的各函数封装故障信息, 提高扩展性。实践结果表明, 基于该方案构建的标准层次模型能够组态实现不同型号卫星的仿真建模, 而且单机模型内嵌故障注入机制, 可提高卫星仿真模型的复用性, 有效缩短仿真建模的周期。

关键词 卫星; 仿真; 组态化; 故障注入; 远程过程调用; 仿真模型

中图分类号: TP391.9 **文献标志码**: A **doi**: 10. 7523/j. issn. 2095-6134. 2021. 03. 014

RPC-based configuration satellite simulator fault injection method

YANG Shanqiang^{1,2}, LI Huawang^{1,3}, CHANG Liang¹, GAO Caidong¹, YU Yeluo¹
(1 Innovation Academy for Microsatellites, Chinese Academy of Sciences, Shanghai 201210, China; 2 University of Chinese Academy of Sciences, Beijing 100049, China; 3 ShanghaiTech University, Shanghai 201210, China)

Abstract In order to improve the reusability of satellite fault injection model and shorten the development cycle, this paper contains a standard hierarchical model for satellite simulation system, which consists of a satellite simulator, a fault injection front end, and a fault protocol. Each individual machine in the satellite simulator is configured with a universal stand-alone model to make the satellite simulator configurable. The fault injection front end encapsulates the fault information by calling each function in the simulator by RPC (remote procedure call) to improve scalability. The actual results indicate that the standard hierarchical model constructed based on this scheme can be configured to create the simulation models of different types of satellites. Moreover, the fault injection mechanism is embedded in each stand-alone model, which improves the reusability of the satellite simulation model and effectively shortens the cycle time of simulation modeling.

Keywords satellite; simulation; configurable; fault injection; RPC; emulator model

^{*} 中国科学院战略性先导科技专项(XDA04040201)资助
[†] 通信作者, E-mail: lihw@microstate.com

卫星长时间运行在复杂、恶劣的太空环境中,不可避免地会出现故障,有的故障会对卫星的寿命和应用造成影响^[1-2]。故障注入能够有效检验系统的容错能力,已经在软件测试领域得到广泛应用。故障注入的基本思想是:应用和环境共同组成系统,环境的范围包含所有不属于应用程序的代码,应用和环境交互的接口处是故障的注入点,通过在故障注入点上模拟实现各类在真实环境中可能出现的故障,观察被测对象的反应,从而检验其对故障的耐受能力^[3-4]。

目前对故障注入不同角度和深度的研究已有很多成果,其实现方法和应用场景各不相同:

1) 为提高模拟程度,建立基于半物理的实时仿真平台,实现不同层次不同类型故障的单独注入,更加真实地得到故障信息。然而半实物仿真测试环境需要开发硬件设备,搭建硬件连接,测试的及时性和灵活性得不到保证^[5]。

2) 为精确有效地实现故障注入,通过在源码中添加故障注入代码,在有故障注入请求时就进入相应分支进行故障注入。然而故障注入工具的开放性欠好,很难在此基础上进行再开发和扩展^[6]。

3) 为注入任意位置任意类型的故障,提高仿真平台的通用性,根据系统配置信息构建存储所有可能的通信链路节点的故障库,通过多元组或文本模型实现对任意位置任意时间的故障注入。但该方法只实现了目标系统链路的离散故障,缺少对连续故障的注入^[7]。

以上方法与目标研究系统有较强的关联性,但在面对新的目标系统时故障注入模型的复用程度低^[8-11]。本文从提高故障仿真模型的可复用性角度,就卫星系统的仿真测试提出一种基于 RPC (remote procedure call, 远程过程调用) 的组态化模拟器故障注入方法,设计一个内含故障注入的通用卫星单机模型,可以通过通用卫星单机模型的组态化实现不同的卫星单机,同时在配置卫星单机模型时不必考虑故障注入。进一步地,该通用单机适用于不同型号的卫星单机,有效缩短卫星仿真建模的周期。

1 故障注入模型设计

1.1 故障注入模型结构

故障注入模型根据功能需要由通用单机模型、故障协议与故障注入前端 3 部分组成。故

障注入前端负责获取用户输入的故障信息和显示卫星的信息,并且通过 RPC 调用模拟器的功能函数来封装、组帧故障信息。通过模型拆分,各模块之间通过接口实现,可以有效地增加整个模型的可维护性和复用性。模型体系结构如图 1 所示。

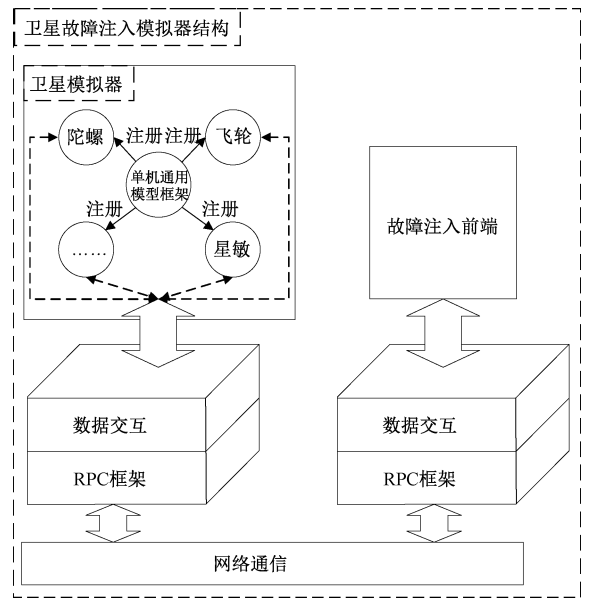


图 1 故障注入模型结构图

Fig.1 Structure chart of fault injection model

1) 故障注入是故障注入前端将故障信息传递给卫星模拟器,然后获取卫星模拟器对故障信息处理的返回值的过 程。为提高故障注入模型的复用性,使之适用于不同型号的单 机,设计一种通用单机模型。在通用单机模型中提供模型接口,当为某型号卫星建模时可以直接调用相应接口配置为实际的单机模型(如星敏感器、飞轮等)。

2) 为提高故障注入模型的可维护性、可扩展性和代码的易读性,将添加故障信息过程抽象为 RPC。使用 RPC 通信能够使故障注入前端就像进行本地调用一样去调用模拟器的相应函数来处理打包故障信息。

3) 同时设计故障注入协议用以对故障前端输出与接收故障信息进行控制,其规定了故障信息的封装方式以及故障注入前端与卫星模拟器之间通信链路的数据格式,是沟通故障注入前端与模拟器的桥梁。

1.2 通用单机模型设计

由于任何数据以及数据交互过程均可抽象为某种数据交互模型,在仿真环境中把数据交互模

型称为数据流模型^[12]。基于此,设计一个如图 2 所示的通用数据流单机模型框架,将单机的参数与数据打包动作进行分离,通过参数库、数据打包、逻辑控制 3 个功能模块描述该通用单机模型。功能模块之间通过接口进行信息交互,单机参数、数据打包以及逻辑动作都是由用户定义,同时故障注入在框架内部完成。

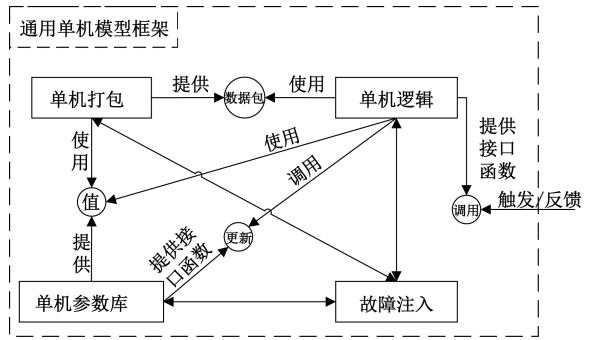


图 2 通用单机模型结构图

Fig. 2 General model structure diagram

为提高单机模型的复用性,对通用单机模型的内部模块做如下约定:

- 1) 参数库中使用三元组(标识符、数据、数据锁)定义单机的参数属性,分别表示该单机收到的指令或遥测数据的身份标识、该参数值和该值是否可以被修改,若此时值由于故障被锁定,则动力学数据无法写入参数库。
- 2) 数据打包模块从参数库中提取数据,进行打包封装、校验,并且当有故障激发,在打包时进行故障注入。

3) 逻辑模块读取和更新参数库的内容,同时决定数据包封装和发送的时机,还规定了发送次数。外部可以调用逻辑模块的外部接口。

单机通用模型的 3 个模块都提供外部接口函数,在搭建具体单机仿真模型时,只需调用相应的接口函数来创建自己的参数库、数据更新函数、数据发送函数即可快速完成单机建模。

1.3 故障协议设计

设计一组故障协议规定模拟器对故障注入前端的故障信息的打包封装格式。并将通用故障协议分为 3 层:链路层、任务数据层、业务控制层,如图 3 所示。

1) 链路层协议

链路层是整个故障协议的最底层,所有的上层模块都依赖于链路层,其不关心实际的远程调



图 3 故障协议层次图

Fig. 3 Fault agreement hierarchy

用逻辑,仅负责信息的收发和编解码,同时为上层模块提供与业务无关的基础服务。

在链路层,数据是以通信帧的形式在卫星单机各模块间流动的,各种数据从产生节点到目的节点需要指定一个传输链路,并且需要规定好通信帧的格式,本文设计的通信帧格式如表 1 所示。此处,卫星模拟器与故障注入前端之间通过 TCP (transmission control protocol, 传输控制协议) 进行通信。

表 1 通信帧格式描述

Table 1 Description of communication frame format		
序号	名称	描述
1	帧头	固定值
2	长度	通信帧的长度
3	数据类型	通信帧承载的数据的类型
4	出发点	发送该数据的单机
5	目的地	接收该数据的目标单机
6	数据	实际要发送的数据
7	校验	该通信帧的校验和
8	帧尾	固定值

2) 任务数据层协议

任务数据层主要记录故障注入的数据信息,同时对不同的故障信息设计不同的数据格式封装处理,这里设计 6 种(协议类、单机参数类、动力学数据类、仿真控制类、指令注入类、外设类)数据格式应对不同类型的故障信息。

这 6 种数据格式均分为有效载荷区和数据区两部分,有效载荷存放的是需要调用的数据交互方法和同类故障数据的组数计数器,数据区为故障前端输入的故障数据。对不同的故障信息,数据区内部的数据结构不同,如图 4 所示。

3) 业务控制层协议

业务控制层定义一组数据业务,又称为数据交互方法,卫星模拟器通过调用不同的业务来处理故障注入前端传来的故障信息,即将故障信息注入到卫星仿真模型中。

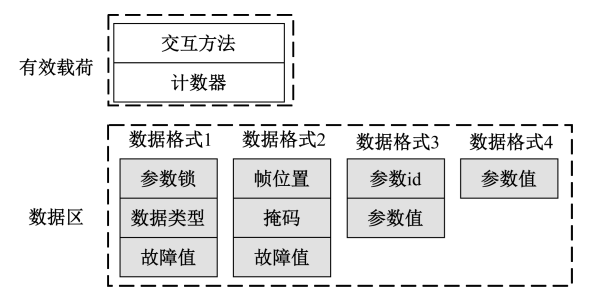


图 4 任务数据层数据格式

Fig. 4 Data format of task data layer

由于卫星中数据交互的任意接口位置都有可能发生故障,因此在本文中将所有数据交互方法统称为原子故障。原子故障是卫星模拟器的最底层故障,通过原子故障可以组合共同作用实现更加复杂的故障,根据卫星模拟器故障注入的需要,设计表 2 所示的 28 种原子故障实现卫星模拟器的故障。

1.4 基于 RPC 的通信机制

为了提供未来进一步扩展和维护的可能

表 2 原子故障类型表

Table 2 Atomic fault types

编号	业务名称	业务内容	编号	业务名称	业务内容
1	禁用外设	模拟外设故障	15	获取单机数据	获取参数库单机数据
2	动力学数据故障	模拟动力学数据异常	16	获取所有单机数据	获取参数库所有数据
3	动力学初始化	设置动力学模型初值	17	单机参数锁	锁死参数库某参数
4	填充 FIFO 数据	向双机 FIFO 中导入数据	18	单机数据报告	存储获取的数据
5	获取 FIFO 数据	导出双机 FIFO 中的数据	19	更新单机数据	更新参数库单机数据
6	中断	添加中断冲突	20	更新所有单机数据	更新参数库所有数据
7	暂停仿真	暂停卫星模拟器	21	获取遥测数据	获取遥测数据
8	仿真时	修改仿真时间	22	获取遥测故障信息	获取遥测故障信息
9	仿真速率	调整模拟器仿真速率	23	遥测缓冲区	获取遥测缓冲区数据
10	CAN 帧数据流	CAN 帧的处理规则	24	组遥测帧	将数据组帧
11	获取逻辑	获取单机所有的逻辑	25	更新遥测故障列表	获取遥测故障信息
12	逻辑列表	存储单机的逻辑	26	遥测校验	对遥测进行校验
13	逻辑锁	将发送次数 N 锁死	27	移除遥测故障	移除遥测的故障信息
14	重复发送次数	设置 1 s 内发送数据次数	28	注入遥测故障	添加遥测帧故障信息

性^[13],使用 RPC 完成对故障注入前端故障信息的封装。RPC 是一种基于客户/服务模式的函数调用过程,它将一个服务的请求和执行分别存放在客户端和服务端之中,其本质是在客户端要求远程服务端运行某项服务并获取其结果的过程,如图 5 所示。它是一种进程间的通信机制,一般承担系统协调与数据交互的责任,并未规定具体的实现方式、平台和语言^[14-15]。

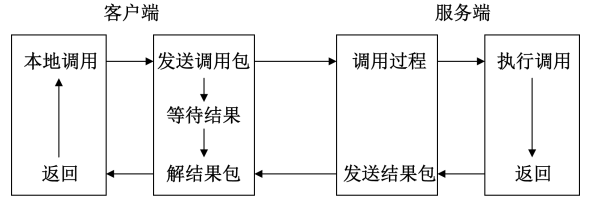


图 5 调用包在远程过程调用中的传输

Fig. 5 Transmission of the call package during a RPC

通过使用 RPC 的通信机制,可以将故障信息的封装与故障注入前端分离,这样故障注入前端只需要负责捕获用户输入的故障信息以及显示功能,易于扩展新的故障。

这里将故障注入前端设定为客户端,卫星模拟

器为服务端,提供客户端需要的故障信息打包封装函数。通过 RPC 封装故障前端的故障信息的过程为:当使用者在故障注入前端添加故障信息后,客户端将目标程序的规范和相关参数放置在调用信息包中,并请求 RPC 网络链路将调用包发送给被调用者,即服务端。一旦服务端接收到这些信息包,服务端会将它们解包获取任务数据,然后根据任务数据中有效载荷标识的业务方法去执行一个完全正常的本地调用,该本地调用会调用服务端中相对应的程序去处理任务数据中的数据。与此同时,服务端的调用进程将被挂起并等待结果包的返回。当服务端中的调用完成时,它将结果包发送给客户端,然后结果包传送回给客户端被挂起的进程,它们将被解包并返回客户端的初始调用函数。

1.5 故障注入流程

基于上述设计模块,故障注入实现流程如图 6 所示。

1)在故障注入前端的界面上给目标单机添加故障信息后,通过 RPC 调用数据交互方法将故障信息、故障单机信息添加到通信帧中,然后发送

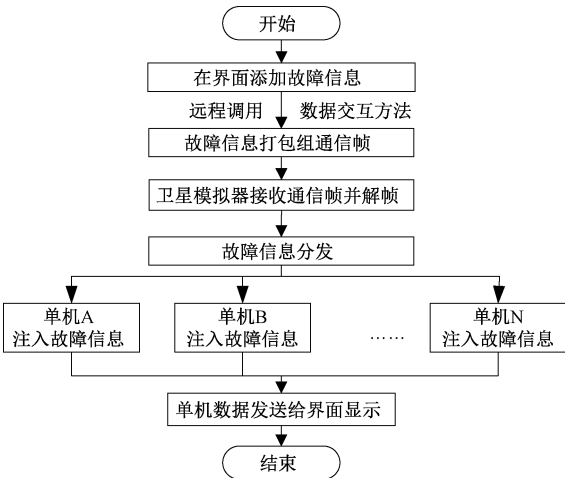


图 6 故障注入流程

Fig. 6 Flow chart of simulating faults of satellite

给模拟器。

2) 模拟器接受到通信帧后,分析通信帧的目的地(即目标单机),向目标单机发送故障数据,

而单机接收到数据后,在更新或打包单机数据时根据故障类型将故障信息封装到单机数据中。

3) 之后模拟器再将单机数据按照表 1 所示格式组帧,然后发送给故障注入前端,故障注入前端再将收到的通信帧解帧并且通过界面显示。

2 实践验证

2.1 组态化实现星敏感器

通过前文设计的卫星通用单机模型能够组态实现不同型号的单机模型,由于篇幅有限,本文仅以某型号的星敏感器为例讲述使用通用单机模型组态化为实际单机的过程,不同型号、不同单机只是它们的参数库、遥测控包和触发事件的逻辑不同,其他并无不同。基于上述通用单机模型实现单机模型的过程是向通用单机模型框架注册参数、数据包、逻辑,同时定义单机接口的过程。图 7 所示为组态实现的该型号星敏感器结构图。

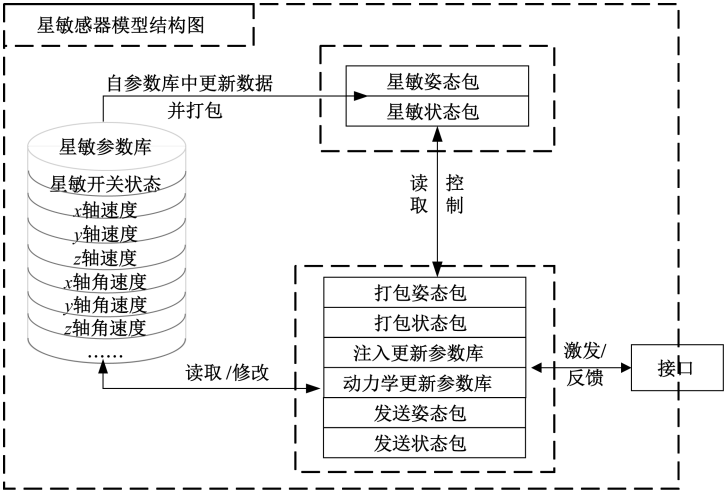


图 7 星敏感器结构图

Fig. 7 Star sensor structure diagram

基于通用单机模型实现星敏感器的过程为:

- 1) 搭建星敏感器参数库:输入星敏感器的所有参数,并给这些参数赋初值。
- 2) 构建星敏感器打包模块:首先设置该星敏感器的参数包列表;然后给出各参数包的组成元素,即每个参数包由参数库中的哪些元素组成,以及将这些参数打包的动作;最后设置对遥控指令帧的解帧、提取指令值的动作。
- 3) 注册星敏感器逻辑模块:设置单机参数更新的时机、连接指令与打包动作、连接指令与参数包发送动作。

基于此方法实现星敏感器的花费资源如表 3

所示。

表 3 新旧方法花费资源对比		
Table 3 Resource cost comparison of new and old methods		
	代码量/行	耗时/d
旧方法	872	2
组态化新方法	495	1

2.2 故障注入过程示例

表 2 描述的原子故障是最基本的故障,通过多个原子故障组合能够实现任意位置、任意组合的故障注入,表 4 列举了一些常见的故障类型。

表 4 故障示例

Table 4 The fault samples

项目	故障名称	故障场景	组成元素
姿轨控单机故障	单机参数故障	星敏传感器 x 轴角速度永久异常	原子故障 15、17、19
	数据“风暴”	星敏传感器每 1 s 内连续发送 N 帧数据	原子故障 11、12、13、14、15
星务计算机故障	中断冲突	星务软件触发假中断,或中断嵌套	原子故障 6
	禁用外设芯片	模拟外设芯片失效的故障	原子故障 1
遥测控故障	遥测帧故障	对星敏传感器遥测帧按位注入故障	原子故障 21、22、24、26、28
热控故障	单机参数故障	热敏电阻采样极值错误	原子故障 15、17、19

以 2.1 节组态实现的星敏传感器为例,进行故障注入仿真过程演示,该型号有 3 个星敏传感器,分别以 A、B、C 命名,除此之外并无差别,这里选取星敏传感器 A 进行展示。

ID	NAME	TYPE	VALUE
8	timestamp1	val_u16	0
0	开关状态	val_bool	true
1	星敏四元素Q0	val_f64	0.621099
2	星敏四元素Q1	val_f64	0.110083
3	星敏四元素Q2	val_f64	0.010321
4	星敏四元素Q3	val_f64	0.775894

(a) 注入参数故障前

ID	NAME	TYPE	VALUE
8	timestamp1	val_u16	0
0	开关状态	val_bool	true
1	星敏四元素Q0	val_f64	0.981559
2	星敏四元素Q1	val_f64	1000
3	星敏四元素Q2	val_f64	-0.152304
4	星敏四元素Q3	val_f64	-0.00739753

(b) 注入参数故障后

图 8 对星敏传感器注入参数故障的过程

Fig. 8 The process of injecting a parameter fault into a star sensor

在正常仿真状态下,星敏 A 的星敏四元素数值变化范围为 $(-1,1)$,如图 8(a) 所示;接下来模拟星敏四元素 Q1 未能正常更新的故障,在给星敏四元素 Q1 注入参数故障 1000 并锁死后,观察到 Q1 的值固定在 1000 而不再更新。当星务软件检测到星敏 A 发生了故障,根据内部算法放弃

星敏 A 采集到的数据,改用星敏 B 的数据。由此判定星务软件对星敏故障的应对措施有效。

2) 注入遥测帧故障

图 9 展示对星敏传感器 A 遥测姿态包的第 3、4 个字节中的某些位注入故障前后姿态包信息的变化。

姿态包	刷新	1	2	3
0000 a4 2b b1 f2 80 00 00 00 00 0a 08 26 ff				
0010 00 00 03 a1 00 00 04 32 00 00 23 2b 00				
0020 00 00 00 00 00 00 00 00 00 00 00 00 00				
0030 e0 00 00 0a 00 2c 14 a2 00 00 00 00 00				
0040 00 00 00 00 00 00 00 00 00 00 00 00 00				
0050 00 00 00 00 00 00 00 00 00 00 00 00 00				
0060 00 00				

(a) 注入遥测故障前

姿态包	刷新	1	2	3
0000 a4 2b b1 f7 0a 00 00 00 00 0a 08 26 00		3	a5	ff
0010 00 00 04 9c ff ff fe 87 00 00 0a e4 00		4	aa	f
0020 00 00 00 00 00 00 00 00 00 00 00 00 00				
0030 e0 00 00 0a 00 2c 13 fe 00 00 00 00 00				
0040 00 00 00 00 00 00 00 00 00 00 00 00 00				
0050 00 00 00 00 00 00 00 00 00 00 00 00 00				
0060 00 00				

(b) 注入遥测故障后

图 9 对星敏传感器注入遥测故障的过程

Fig. 9 The process of injecting a telemetry fault into a star sensor

对星敏传感器 A 发送的遥测姿态包按位进行故障注入,遥测帧故障注入格式为(帧位置、掩码、故障值),遥测帧故障按照下式规则进行注入。

$Value = f(value, mask, fault)$

$= (fault \& mask) \mid (value \& \sim mask), \quad (1)$

式中: $value$ 为遥测帧原值; $mask$ 为掩码值,按位屏蔽输入的故障值,若该位为 1 则屏蔽故障,为 0 则允许该位注入故障; $fault$ 为注入的故障值; $Value$ 为故障注入后值。

通过对姿态包第 3、4 字节的数值(16 进制表示)按照上述规则进行计算,计算结果如图 10 所示,与图 9 展示的故障注入后的数据是匹配的,说

明遥测故障注入有效。



图 10 数值计算结果

Fig. 10 Numerical calculation result

2.3 新方法使用前后对比

在设计使用本故障注入模型之前,对某通信卫星建立仿真模型以及故障注入模型时,该通信

卫星的陀螺、反作用轮、推力器、星敏、太敏等不同单机的故障源、故障类型都不相同,某一单机的不同故障也需要单独处理,导致该型号的仿真模型代码冗余,一般需要一个月的时间才能完成卫星仿真模拟器的搭建和调试。

而使用基于 RPC 的故障注入模型,只需调用通用单机模型的接口组态化为不同型号的不同单机模型,该方法已应用于对 SVOM、某通信卫星、某遥感卫星的仿真建模中,为这些型号搭建仿真模拟器分别使用了两周左右时间,为这些型号的方案设计验证、星务代码的测试提供了支持。

同时,使用基于 RPC 的故障注入方法具有故障类型丰富的特点,支持如数据风暴;同一时间点内发送大量有效数据;中断冲突;星务软件触发假中断或中断嵌套。

使用新旧方法参数对比如表 5 所示。

表 5 新旧方法对比

Table 5 Comparison of parameters of old and new methods

其他方法	基于 RPC 的故障注入方法
卫星平台单一 ^[6]	适用于多种卫星平台
不同的单机实现各自的故障 ^[8]	单机模型统一内嵌多种故障,可选择性使用
故障类型较少 ^[9]	故障类型丰富,可以方便地添加新的故障
建模任务繁重,代码工作量大 ^[5]	只需调用接口配置

3 结束语

本文设计一种基于 RPC 的组态化模拟器标准故障注入方法,通过对卫星单机建立通用仿真模型,对卫星系统进行故障注入。该卫星仿真模型实现的功能如下:

1) 虚拟化构建卫星的硬件环境和外部数据源,等效模拟实际卫星的运转状况。

2) 在没有故障注入时,故障注入前端向模拟器发送获取数据的请求,模拟器将由动力学更新的单机数据组包发送给故障注入前端,故障注入前端将通信帧解帧获得各单机参数并显示在界面上。

3) 在有故障信息注入时,故障注入前端通过 RPC 调用模拟器端的数据交互方法将故障信息组帧打包发送给模拟器端,模拟器解包通信帧获得故障信息,在单机数据更新和打包时将故障信息注入,然后向故障注入前端发送已被故障注入的该单机数据组包。

本方法投入实践使用后,能够有效缩短卫星

仿真模型开发周期,对型号的仿真建模从原来的一个月缩短到两周时间。在以后的研究工作中,应继续提炼形成更加标准化的通用模型,使得可以通过配置文本的方式建立卫星仿真模型,进一步提高模型组态化、复用性的程度;另一方面可以从动力学模型、能源热控模型等角度入手,提高模型的仿真精度。

参考文献

[1] Chen J Y, Hu X G, Zhang G F. Design and implementation of time management in simulation and test platform for satellite[C] //2016 12th IEEE International Conference on Control and Automation (ICCA). Kathmandu Nepal: IEEE, 2016: 232-235.

[2] Hu X G, Liang Q. Research on digital simulation of satellite integrated electronic system [C] //11th IEEE International Conference on Control & Automation (ICCA). Taichung, Taiwan, China: IEEE, 2014: 112-116.

[3] 徐璐, 吴际, 刘超. 一种模型驱动的协议测试故障注入引擎设计[J]. 计算机工程与应用, 2007, 43(16): 111-113, 145.

[4] 易锦, 罗响, 凹建勋, 等. 基于马尔科夫链的软件故障分类预测模型[J]. 中国科学院大学学报, 2013, 30(4): 562-567.

[5] 花良浩, 殷芝霞, 杨蒲. 无人机故障注入与故障诊断实时仿真平台研制[J]. 计算机应用与软件, 2013, 30(8): 106-108, 142.

[6] 杨海峰, 张发家, 刘吉霞, 等. 卫星在轨故障全数字仿真平台设计[J]. 航天器工程, 2012, 21(4): 21-25.

[7] 姚文斌, 赵玲, 王真, 等. 故障注入通用仿真平台的研究与设计[J]. 系统仿真学报, 2016, 28(2): 315-321.

[8] 姬祥, 董云卫, 王宇英, 等. 组态化嵌入式软件仿真系统设计[J]. 计算机工程, 2011, 37(21): 225-227, 231.

[9] Li G, Ying J, Wu M H. YSS2000: a configuration software system for industrial monitoring and controlling [C] // International Conference on Information Acquisition, 2004. Proceedings. Hefei, China: IEEE, 2004: 466-470.

[10] Fan J X, Hu X G, Ding D W, et al. A generic satellite simulation framework design and implementation[C]//2016 IEEE 11th Conference on Industrial Electronics and Applications (ICIEA). Hefei, China: IEEE, 2016: 2571-2574.

[11] Gao F, Deng F, Yan Y Q, et al. Full digital simulation testing of networked embedded software [C] //2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C). Lisbon, Portugal: IEEE, 2018: 45-50.

[12] 欧阳高翔, 李鑫, 董鑫, 等. 全数字仿真及其在卫星在轨维护中的应用[J]. 系统仿真学报, 2016, 28(4): 979-990.

[13] 于天, 黄祖. 一种高性能异步 RPC 框架的设计与实现[J]. 信息通信, 2018, 3: 127-129.

[14] 刘晓风. 多线程 RPC 分布式计算的实现[J]. 计算机工程与设计, 2003, 24(3): 24-26.

[15] Birrell A D, Nelson B J. Implementing remote procedure calls [J]. ACM Transactions on Computer Systems, 1984, 2(1): 39-59.