

6-4-2020

Design of Real-time Communication on Bus Using dSPACE System

Yan Cheng

DFH Satellite Co., Ltd, Beijing 100094, China;

Decong Huo

DFH Satellite Co., Ltd, Beijing 100094, China;

Xie Song

DFH Satellite Co., Ltd, Beijing 100094, China;

Follow this and additional works at: <https://dc-china-simulation.researchcommons.org/journal>



Part of the Artificial Intelligence and Robotics Commons, Computer Engineering Commons, Numerical Analysis and Scientific Computing Commons, Operations Research, Systems Engineering and Industrial Engineering Commons, and the Systems Science Commons

This Paper is brought to you for free and open access by Journal of System Simulation. It has been accepted for inclusion in Journal of System Simulation by an authorized editor of Journal of System Simulation.

Design of Real-time Communication on Bus Using dSPACE System

Abstract

Abstract: The analysis of the problem of real-time communication on CAN bus was proposed using dSPACE simulator, and the solution was proposed. *The CAN communication model based on dSPACE was designed using the methods of Interrupt or Enable to trigger the action.* The spacecraft subsystem simulation model was built on the dSPACE, and the real-time communication was performed following the spacecraft CAN bus communication protocol, to *implement the communication with large amount of data between subsystem simulation model and data handling subsystem.* The simulation result verifies the validity of the communication method.

Keywords

dSPACE, real-Time simulation, CAN communication, trigger method

Recommended Citation

Yan Cheng, Huo Decong, Xie Song. Design of Real-time Communication on Bus Using dSPACE System[J]. Journal of System Simulation, 2017, 29(6): 1297-1303.

基于 dSPACE 系统的总线实时通讯方法设计

阎诚, 霍德聪, 谢松

(航天东方红卫星有限公司, 北京 100094)

摘要: 对利用 dSPACE 仿真器进行 CAN 总线实时通讯时所面临的问题进行了分析, 提出了解决该问题的思路和方法。使用中断和使能两种触发方式, 设计了基于 dSPACE 的 CAN 通讯接口模型。构建了卫星分系统仿真模型, 按照星上 CAN 总线通讯协议, 进行了实时仿真, 实现了在分系统仿真模型与星务分系统之间进行大数据量通讯。仿真结果证明了本文设计的 CAN 通讯方法的有效性。

关键词: dSPACE; 实时仿真; CAN 通讯; 触发方式

中图分类号: TP337 文献标识码: A 文章编号: 1004-731X (2017) 06-1297-08

DOI: 10.16182/j.issn1004731x.joss.201706019

Design of Real-time Communication on Bus Using dSPACE System

Yan Cheng, Huo Decong, Xie Song

(DFH Satellite Co., Ltd, Beijing 100094, China)

Abstract: The analysis of the problem of real-time communication on CAN bus was proposed using dSPACE simulator, and the solution was proposed. The CAN communication model based on dSPACE was designed using the methods of Interrupt or Enable to trigger the action. The spacecraft subsystem simulation model was built on the dSPACE, and the real-time communication was performed following the spacecraft CAN bus communication protocol, to implement the communication with large amount of data between subsystem simulation model and data handling subsystem. The simulation result verifies the validity of the communication method.

Keywords: dSPACE; real-Time simulation; CAN communication; trigger method

引言

在当今的航天领域, 卫星研制的复杂性与难度逐年增加, 采用各种仿真方法进行先期地面验证实验、为卫星设计提供依据、降低卫星设计过程中的风险, 已经成为国际流行的趋势。借助于先进仿真技术不但能大大减轻设计人员的工作量, 而且能缩短研发周期, 减少研发的经费, 因

此高效便捷、高精度、高可靠仿真技术的研究越来越受到人们的重视。

随着计算机技术和仿真技术的高速发展, 实时仿真逐渐成为各种控制系统设计的重要手段。实时仿真又称半物理仿真, 可以更真实地模拟仿真对象的运行情况, 获得更为可靠的仿真结果。德国 dSPACE 公司开发的基于 Matlab/Simulink 环境的半物理仿真器 dSPACE 系统是最先进的实时仿真系统之一, 在汽车、电力、航空、航天等工业领域都已获得广泛应用^[1-2]。

控制器局域网(Control Area Network, CAN)总线通讯具有突出的可靠性、实时性和灵活性, 其总线规范已经成为国际标准, 被公认为几种最有前途



收稿日期: 2015-07-20 修回日期: 2015-11-02;
作者简介: 阎诚(1984-), 男, 重庆, 硕士, 高工, 研究方向为卫星总体设计与仿真; 霍德聪(1984-), 男, 河北, 硕士, 高工, 研究方向为卫星总体设计; 谢松(1983-), 男, 湖南, 硕士, 高工, 研究方向为卫星仿真。

<http://www.china-simulation.com>

• 1297 •

的总线之一^[3], 并已在我国航天器上获得广泛应用。在我国最近十年发射的卫星中, 约有 1/3 以上均采用了 CAN 作为星上总线。

目前, 国内航天领域对 dSPACE 实时通讯的研究主要集中在串口方面^[4-8], 利用 dSPACE 对 CAN 总线实时通讯进行仿真研究的相关文献极少且不够详尽^[9-10]。文献[9]所设计的方法是让 dSPACE 以串口方式与外接的串口/CAN 转换盒进行通讯, 再通过转换盒与其它节点进行 CAN 通讯, 无法保证整个通讯过程的实时性; 文献[10]介绍的是利用 dSPACE 直接进行 CAN 总线通讯的基础操作, 不涉及实时性问题。

本文以 dSPACE 为仿真平台, 搭建卫星星上分系统仿真模型, 按照星上 CAN 总线通讯协议, 重点针对实时性要求, 设计了能够实现星上 CAN 总线数据正确实时通讯的程序结构, 并以星上分系统为仿真对象, 对上述方法进行了仿真实验验证。

1 CAN 通讯模型的处理

1.1 星上 CAN 总线通讯协议

在介绍利用 dSPACE 进行 CAN 总线实时通讯之前, 有必要对星上使用的 CAN 总线通讯协议进行简述。

根据通讯协议的约定, 星上的 CAN 通讯采用“星务主动、分系统被动”的应答式通讯方式, 也即: 星上分系统只响应星务发送的 CAN 数据包, 不主动发送 CAN 数据包。具体为:

1. 当分系统接收到星务发送的遥控指令数据包时, 发送应答数据包;
2. 当分系统接收到星务发送的遥测参数轮询数据包时, 发送指定类型的遥测参数数据包。

收到的遥控指令存入接收缓冲区并导入分系统仿真模型执行。要发送的遥测参数从分系统仿真模型中读取、编码并存入发送缓冲区。

1.2 CAN 通讯触发方式的确定

利用 dSPACE 进行 CAN 总线通讯的接收和发

送, 必须通过 dSPACE 软件中自带的接收器模块(以下简称 RX)和发送器模块(以下简称 TX)来完成。RX 和 TX 模块均集成在 dSPACE 的 simulink 运行模块中。只要启动 RX, 其就会自动将收到的 CAN 总线数据存入接收缓冲区; 只要启动 TX, 其就会自动将发送缓冲区中的数据发到 CAN 总线上。

但是, RX 与 TX 每次仅能接收或发送(以下简称收发)一帧标准的 8 字节 CAN 数据。对于仅由一帧组成的“单帧”数据包, 这不是问题; 但对于帧数超过一帧的“多帧”数据包, 就需要不断通过触发的方式, 反复启动收发模块来完成多帧数据包的收发。在此, 触发方式的选择至关重要。一般而言, dSPACE 上可以使用的触发方式包括两种: 使能(Enable)触发与中断(Interrupt)触发。

如果采用使能触发方式, 则很难实现 CAN 总线通讯的实时性。以一个 5 帧的数据包为例, 在波特率约为 300 kbps 的 CAN 总线网络中, 其完全传输的理论最短时间不超过 5 ms。但如果对于每一帧的收发都采用使能触发方式, 则由于每一仿真步长内只能进行一次使能操作, 数据帧收发完成后会有一定的空闲时间。假如将 dSPACE 仿真步长设置为 5 ms, 则 5 帧的传输需要至少 25 ms, 远远高于理论最短时间, 不满足实时性要求。如果缩小仿真步长, 当仿真步长小到低于收发单帧的最小时间时, 会导致收发失败; 当仿真步长仍然高于收发单帧的最小时间时, 则仍会造成时间上的空闲和浪费。

采用中断方式则不存在上述问题。dSPACE 的程序模块中包括发送中断器 CAN TX Interrupt 和接收中断器 CAN RX Interrupt, 每次收发完成后, 中断器便被触发, 其输出的脉冲信号可以被立刻用于触发收发模块 RX 与 TX。在收发与中断之间几乎不存在等待空闲时间, 与仿真步长完全无关, 可以达到尽可能高的收发速度。

对于 RX, 只要有来自 CAN 总线上的数据, 即可进行接收, 并同时触发接收中断器产生一次中断, 该中断可以触发 RX 接收下一帧数据, 接收下

一帧数据的同时又触发接收中断器产生下一次中断, 如此反复循环。当全部数据接收完毕时, 接收中断器不再被触发, 中断即停止, RX 的数据接收动作同样停止。

但是对于 TX, 却存在如下问题: 在每次发送多帧 CAN 数据包时, 其第一帧的发送必须通过外部触发的方式。这是因为, 发送第一帧之前, 发送中断并没有被触发, 因此也就无从产生触发第一帧发送的信号。此外, 当多帧数据包发送完成后, 如果不采取关断措施, 则 TX 会一直被中断信号触发, 发送动作会一直延续下去, 进入死循环, 无法停止。

综上所述, 对于多帧 CAN 数据包的接收, 只需设立接收中断器作为触发信号源即可。对于多帧 CAN 数据包的发送, 必须设立两种发送触发信号源: 一种为使能触发, 用于启动和关闭发送过程; 一种为中断触发, 利用发送中断器操作除第一帧之外的后续帧的连续快速发送。

单帧 CAN 数据包的接收和发送较为简单, 直接采用使能触发即可。

1.3 CAN 通讯多帧中断发送模块的设计

本节以最为复杂的多帧 CAN 遥测参数数据包发送为例, 介绍利用 dSPACE 进行 CAN 通讯的方法和思想。

如前所述, 对于多帧发送, 存在两种不同的触发信号源。在 Simulink 程序中, 这两种信号源分别触发使能子系统(Enable Subsystem)和功能调用子系统(Function-call Subsystem)。但是需要特别注意的是: 在 Simulink 系统下, 这两种触发信号之间不能进行与、或、非等逻辑操作。这样的后果, 导致两种信号源必须分别触发并驱动不同的子系统模块。

因此, 在多帧中断发送程序的设计中, 对于同一个端口的多帧 CAN 数据包的发送, 必须设计两套不同的发送装置, 用两个 TX, 分别发送第一帧和后续帧。这是基于 dSPACE 环境下的 CAN 总线实时通讯的中断发送程序设计中核心的思想。

多帧 CAN 数据包发送程序模块的基本结构示意图如图 1 所示。

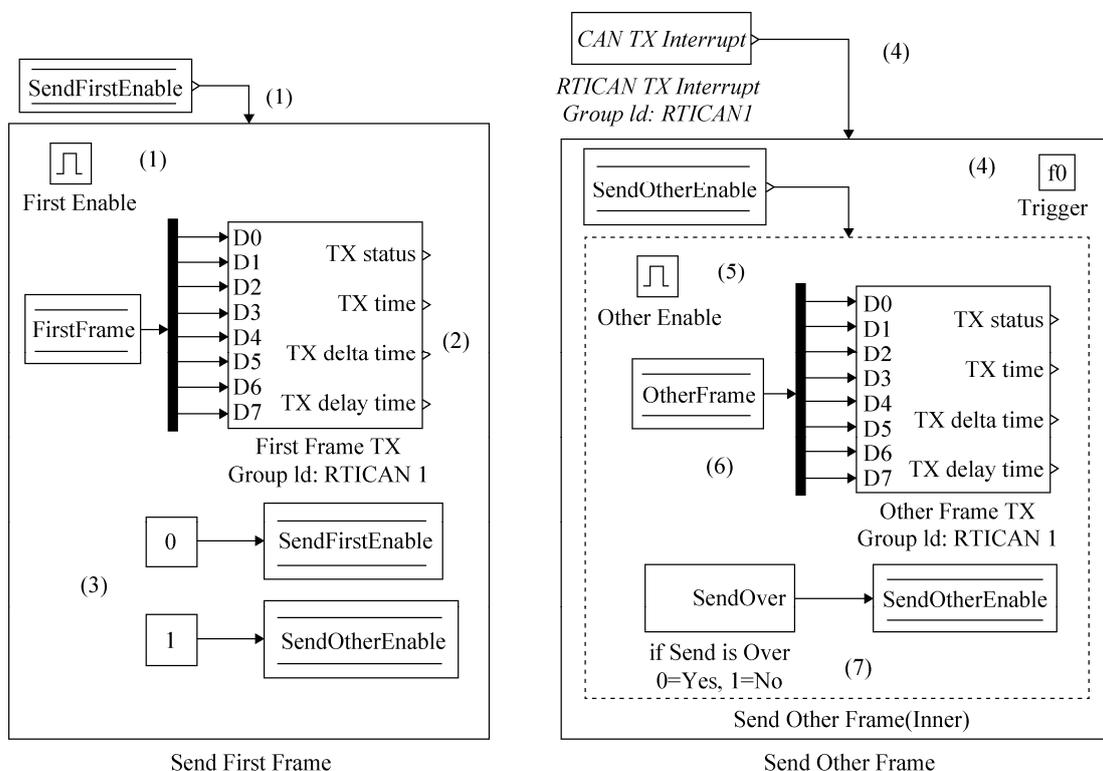


图 1 多帧 CAN 数据包发送程序模块的结构示意图
Fig. 1 Scheme of Multi-frame CAN Data Packet Transmission Program Model

该程序模块包括两部分。左侧为发送第一帧的程序模块，实线框为使能子系统 Send First Frame；右侧为发送后续帧的程序模块，由两层嵌套的子系统组成：实线框为功能调用子系统 Send Other Frame；点划线框为使能子系统 Send Other Frame (Inner)。发送中断器为 CAN TX Interrupt。前述要求的两个 CAN 发送器模块 TX 分别为 First Frame TX 和 Other Frame TX。整个程序模块的运行动作顺序说明如下。

(1) 状态字 SendFirstEnable 作为触发 CAN 发送的初始信号源，由分系统仿真程序中的其它部分为其赋值。当该值被置 1 后，触发了使能子系统 Send First Frame 的触发器 First Enable，使能子系统 Send First Frame 被激活，启动发送；

(2) 使能子系统 Send First Frame 被激活后，其中的 CAN 发送器模块 First Frame TX 执行发送操作，发送第一帧数据，该数据由发送缓冲区 FirstFrame 提供；

(3) 在使能子系统 Send First Frame 被激活、第一帧数据被发送的同时，状态字 SendFirstEnable 置 0，状态字 SendOtherEnable 置 1。也即，第一帧数据发送一次后就关闭了左侧的使能子系统 Send First Frame，同时为后续帧发送做好了准备；

(4) 在第一帧发送的同时，发送中断器 CAN TX Interrupt 收到发送中断信号，从而输出触发信号，触发了功能调用子系统 Send Other Frame 的触发器 Trigger，子系统 Send Other Frame 被激活；

(5) 由于状态字 SendOtherEnable 在步骤(3)中已经被置 1，因此在功能调用子系统 Send Other Frame 被激活后，使能子系统 Send Other Frame (Inner)的触发器 OtherEnable 也被触发，使能子系统 Send Other Frame (Inner)被激活；

(6) 使能子系统 Send Other Frame 被激活后，其中的 CAN 发送器模块 Other Frame TX 执行发送操作，发送后续帧数据，该数据由发送缓冲区 OtherFrame 提供。每发送一帧，发送中断器 CAN TX Interrupt 就中断并触发一次，因此发送不断进

行，发送缓冲区 OtherFrame 中的具体待发数据则不断变化，后续帧从而依次发出；

(7) 子系统 If Send is Over 用来判断多帧数据包的发送是否已完成，最基本的方法即通过计数器来监控这一次发送到底发了多少帧 CAN 数据。若 If Send is Over 判断发送未完成，则对状态字 SendOtherEnable 置 1，该状态字与发送中断器 CAN TX Interrupt 一起，共同保证了使能子系统 Send Other Frame (Inner)处于激活状态，从而使 CAN 发送器模块 Other Frame TX 能够继续发送；若发送完成，则 If Send is Over 对状态字 SendOtherEnable 置 0，置 0 后使能子系统 Send Other Frame (Inner)关闭，发送停止。由于发送停止，发送中断器 CAN TX Interrupt 也不再产生触发信号，功能调用子系统 Send Other Frame 关闭。此时整个 CAN 多帧遥测参数数据包发送完成并停止。

上述 7 个步骤利用了状态字切换和发送中断触发的方法，解决了利用 dSPACE 进行多帧 CAN 数据包实时发送的问题。其具体流程图如图 2 所示。

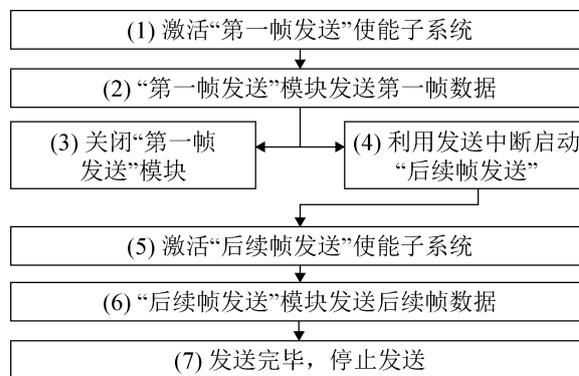


图 2 多帧 CAN 数据包发送流程图

Fig. 2 Flow of Multi-frame CAN Data Packet Transmission

1.4 CAN 通讯收发时序的设计

在 dSPACE 进行多帧数据包的发送过程中，存在着这样的隐患：发送进行到一半时，dSPACE 接收了一个与发送数据包采用同样通讯端口的 CAN 总线数据包，由于端口被占导致发送停止，发送中断失效，从而导致发送失败。因此，采用严格的顺序处理的时序设计，使接收和发送任务之间不产生

时间冲突,是保证分系统进行正确通讯的必要条件。

在图 3 中,左边的模块体现了顺序处理的收发时序的最基本设计思想:状态字 `SendFirstFrame` 和 `SendOtherFrame` 不会被同时置 0 或 1,而只会被同时置为两个不同的状态,这也就保证了对多帧数据包的发送,第一帧的发送和后续帧的发送不可能同时进行,不会产生时间冲突。

如前所述,仿真平台使用的 CAN 总线通讯协议规定:分系统采用应答式的发送方式,不会主动发送数据包。因此对接收和发送的动作时序按照如下要求设计:当每次的发送任务结束后,将接收状态字设为 1,发送状态字设为 0,保持随时准备接收的状态;只有在接收完成后,发送状态字才能被置 1,同时接收状态字应被同时设为 0,变成只发送不接收的状态。

改进后的程序模块基本结构示意图如图 3 所

示。与图 1 相比,图 3 增加了最左侧的接收模块 `Receive Frame`,其工作模式与最右侧的后续帧发送模块 `Send Other Frame` 基本类似,不再赘述。由于应答式发送方式的要求,只有在接收模块 `Receive Frame` 接收完 CAN 数据包之后,发送模块才会开始发送。从三个方框下方的置状态字环节可以看出,三个控制状态字 `ReceiveEnable`、`SendFirstEnable`、`SendOtherEnable` 在任何时刻都只会有一个被置 1,这也就保证了在任何时候,发送模块 `Receive Frame`、第一帧发送模块 `Send First Frame`、后续帧发送模块 `Send Other Frame` 这三个子系统在同一时间只会有一个被激活。因此,CAN 数据包的接收、第一帧发送与后续帧发送这三种动作在时间上不会发生冲突。在后续帧发送完成后,将接收状态字 `ReceiveEnable` 置 1,准备接收下一次 CAN 数据包,并将发送状态字置 0。

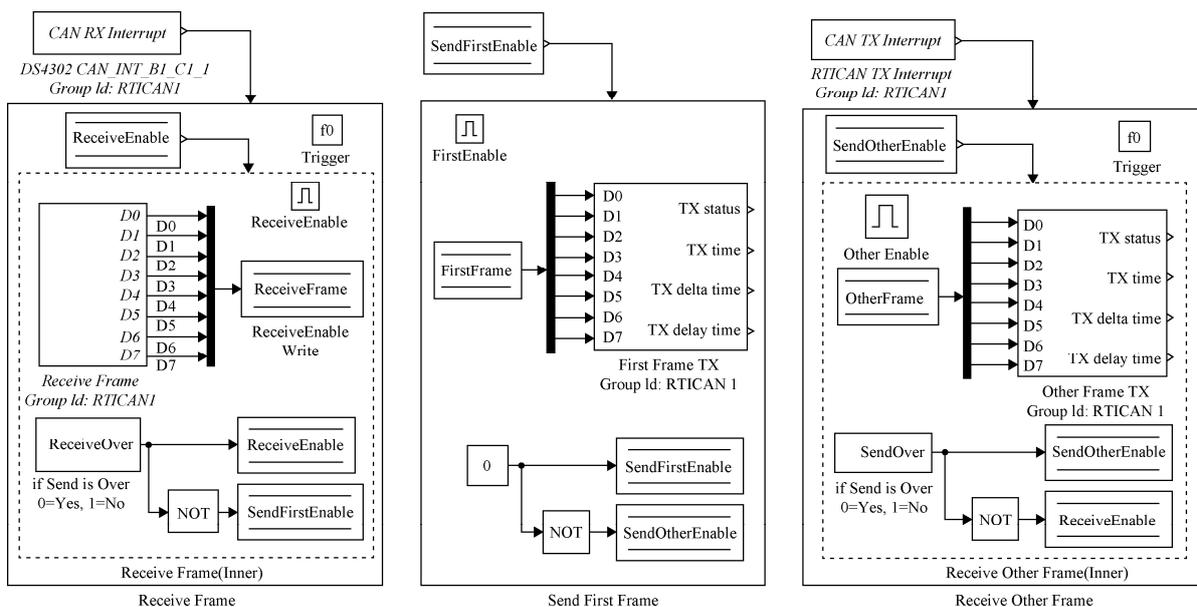


图 3 改进后的 CAN 数据包收发程序模块结构示意图

Fig. 3 Scheme of Updated Multi-frame CAN Data Packet Transmission Program Model

2 CAN 通讯的实时仿真

2.1 仿真硬件架构

仿真模型的硬件架构包括一台基于工控机的星务仿真机与一套基于 dSPACE 的分系统仿真机,两者之间通过 CAN 总线进行连接。前者用于模拟

星上星务分系统的工作,为分系统仿真提供支持。后者包括一块 DS1005 处理器板、一块 DS4302 板卡和一台分系统上位机。DS1005 是 dSPACE 仿真器中专门用于进行实时仿真运行计算的处理器板,其功能相当于 CPU;DS4302 则是 dSPACE 仿真器中用于进行 CAN 总线通讯的专用板卡,它与

DS1005 在仿真器内部通过 PHS 总线进行数据交互；分系统上位机用于进行仿真管理、数据显示与存储等功能，其与 dSPACE 仿真器之间通过 PCMCIA 总线相连。仿真模型的硬件架构如图 4 所示。

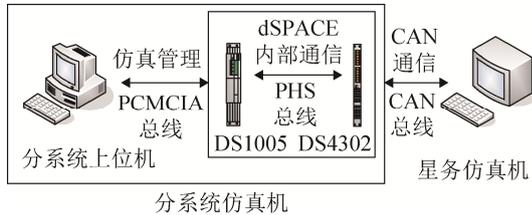


图 4 仿真系统硬件架构

Fig. 4 Scheme of Simulation System Hardware

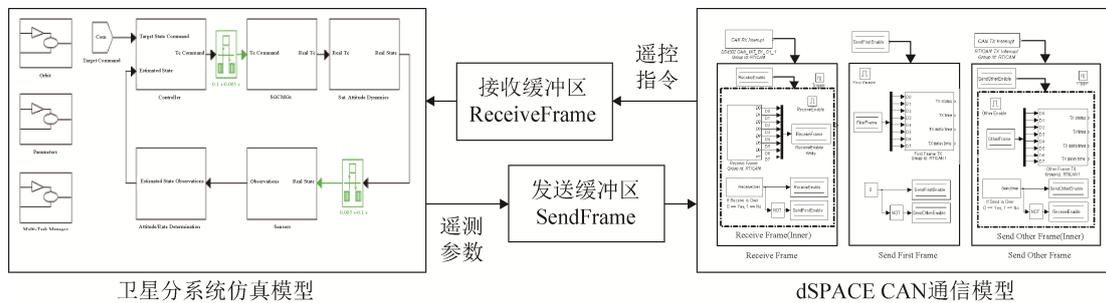


图 5 仿真系统软件模型

Fig. 5 Model of Simulation System Software

2.3 仿真设置

将仿真平台的仿真步长设为 5 ms。仿真时间为 60 000 s。CAN 总线通讯波特率为 300 kbps。按照星上 CAN 通讯协议，分系统与星务之间的 CAN 数据流如表 1 所示。

表 1 分系统 CAN 通讯数据流

Tab. 1 Data Flow of Subsystem CAN Communication

源节点	目的节点	数据内容	周期	数据量/帧
星务	分系统	遥控指令 1	事件	4
星务	分系统	遥控指令 2	事件	100
分系统	星务	遥控指令 1 应答	事件	1
分系统	星务	遥控指令 2 应答	事件	1
星务	分系统	遥测参数 1 轮询	奇数秒	1
		遥测参数 2 轮询	偶数秒	1
		遥测参数 3 轮询	4 s	1
		遥测参数 4 轮询	1 s	1
分系统	星务	遥测参数 1	奇数秒	28
		遥测参数 2	偶数秒	200
		遥测参数 3	4 s	18
		遥测参数 4	1 s	5

注：分系统模型搭建于 dSPACE 上。

2.2 仿真软件模型

仿真软件模型包括 CAN 通讯模型与分系统仿真模型两大部分。CAN 通讯模型用于与分系统仿真模型进行仿真数据交互，并将这些数据进行编码或解码处理后传递到 CAN 总线上，其基本结构如图 3 所示，已在 2.3 和 2.4 节进行了详细描述；分系统仿真模型以某在研卫星姿轨控分系统为基础，用于模拟该卫星在轨运行期间的姿态运动和控制算法。这两个模型之间通过程序中的 ReceiveFrame 接收缓冲区和 SendFrame 发送缓冲区进行数据交互，其基本结构如图 5 所示。

星务按固定的时间间隔每秒钟向分系统循环发送遥测参数轮询信号，分系统接收后将当前时刻指定类型的遥测参数发送给星务。

分系统遥控指令 1 和 2 采用随机方式发送，由星务发给分系统，发送时间间隔在 1~20 s 内随机变化，每次发送数据内容随机生成，分别发送 6 000 条，分系统每收到一次遥控指令后返回一次应答。所有的通讯过程都在 CAN 总线上进行。

2.4 仿真结果与分析

利用上述参数和仿真模型，在 dSPACE 仿真器上进行仿真。采用两种不同的方式生成仿真数据：A) 将 dSPACE 仿真器生成的仿真数据导入分系统上位机并保存；B) 将星务仿真机通过 CAN 总线从 dSPACE 处收到的遥测参数进行解码并保存。在仿真过程中，对 CAN 数据包通讯过程中的时间性能进行了测试。

对比了利用 A、B 两种方法生成的 60 000 s 时间内共 8 181 000 帧 CAN 数据, 结果如表 2 所示。

可见, 基于 dSPACE 模型搭建的分系统对所有遥控指令接收正确; 应答正确; 所有轮询信号接收正确; 所有遥测参数发送正确。证明所设计的基于 dSPACE 的 CAN 总线实时通讯方式正确可靠。

从 CAN 通讯时间性能测试结果来看, 基于

dSPACE 模型搭建的分系统对 100 帧的 CAN 数据包的接收传输时间仅略高于理论上的最短传输时间(约 30 ms); 对 28 帧和 200 帧的 CAN 数据包的发送传输时间仅略高于理论上的最短传输时间(分别为约 10 ms 和约 55 ms)。说明本文所设计的基于 dSPACE 的 CAN 总线通讯方式能够满足快速通讯的实时性要求。

表 2 CAN 通讯数据传输性能

Tab. 2 Data Transmission Performance of CAN Communication

数据内容	分系统预期通讯结果	分系统实际通讯结果	每包平均时间/ms
遥控指令 1	接收 6 000 条, 共 24 000 帧	接收 24 000 帧全部成功, 数据正确	6.46
遥控指令 2	接收 6 000 条, 共 600 000 帧	接收 600 000 帧全部成功, 数据正确	33.71
遥控指令 1 应答	发送 6 000 次, 共 6 000 帧	发送 6 000 帧全部成功, 数据正确	<1
遥控指令 2 应答	发送 6 000 次, 共 6 000 帧	发送 6 000 帧全部成功, 数据正确	<1
遥测参数轮询	共接收 遥测 1 轮询: 30 000 条, 共 30 000 帧	接收 135 000 帧全部成功, 数据正确	<1
	遥测 2 轮询: 30 000 条, 共 30 000 帧		<1
	遥测 3 轮询: 15 000 条, 共 15 000 帧		<1
	遥测 4 轮询: 60 000 条, 共 60 000 帧		<1
遥测参数发送	共发送 遥测 1: 30 000 次, 共 840 000 帧	发送 7 410 000 帧全部成功, 数据正确	13.97
	遥测 2: 30 000 次, 共 6 000 000 帧		61.29
	遥测 3: 15 000 次, 共 270 000 帧		11.81
	遥测 4: 60 000 次, 共 300 000 帧		8.39

3 结论

从基于 dSPACE 的实时仿真结果, 可以得出如下结论:

(1) 实时仿真的测试结果表明: 本文所采用的、基于 dSPACE 系统设计的 CAN 通讯接口模型是有效的, 单帧、多帧 CAN 数据包的接收、发送均能正确地完成。上述 CAN 通讯方法结合 dSPACE 强大的实时仿真与计算能力, 可以方便地构建利用 CAN 总线进行实时通讯的仿真系统, 为快速、灵活、可信地进行仿真试验提供有效途径。

(2) 本文提出的实时仿真系统和用 Simulink 设计 CAN 通讯接口模型的混合建模方法和思路, 对航空航天工程和其它领域, 在基于 dSPACE 环境下的实时系统设计中, 具有指导、参考和应用意义。

参考文献:

- [1] SPACE. ControlDesk Experiment Guide, Release3.2-3.3 [M]. Paderborn, Germany: dSPACE GmbH, 2009.
- [2] Chen P Z, Tang X Q. Hardware-in-the-loop simulation of parameters self-tuning for servo system [C]// Industrial Electronics and Applications, 2009. ICIEA 2009. IEEE Conference on. USA: IEEE Xplore, 2009: 3158-3162.
- [3] 饶运涛. 现场总线CAN原理与应用技术 [M]. 北京: 北京航空航天大学出版社, 2012: 194-199.
- [4] 翟坤. 基于dSPACE的挠性卫星姿轨控实时仿真系统 [J]. 航天控制, 2004, 22(1): 17-25. (Zhai K. Real Time Simulation System for Attitude and Orbit Control System of Flexible Satellite Based on dSPACE Multiprocessor System [J]. Aerospace Control, 2004, 22(1): 17-25.)
- [5] 朱承元. 卫星姿轨控实时仿真系统及串口通信建模 [J]. 哈尔滨工业大学学报, 2005, 37(1): 26-31. (Zhu C Y. The real-time simulation system of satellite's attitude and orbit control and modeling of serial-port communication [J]. Journal of Harbin Institute of Technology, 2005, 37(1): 26-31.)

(下转第 1310 页)

<http://www.china-simulation.com>