基于 RTPS 协议的网络传输服务

林晓辉，李 实，林 健
(哈尔滨工业大学卫星技术研究所，哈尔滨 150001)

摘 要：介绍了基于 RTPS (Real-time Publish/Subscribe) 协议的 NDDS 服务 (Network Data Delivery Service)，及其实现数据传输的原理、特点和应用范围，并给出了应用其进行网络数据传输的方法。NDDS 提供的特有属性和相关函数使其具有其它网络数据传输服务不可比拟的优势：可以应用到实时系统的数据传输之中,而且经过进一步地开发也可以应用到分布式实时仿真系统的协同运行方面。

关键词：网络数据传输；实时；同步；RTPS；NDDS
文章编号：1004-731X (2005) 05-1120-05 **中图分类号：**V411.8 **文献标识码：**A

Network Data Delivery Service Based on RTPS Protocol

LIN Xiao-hui, *LI Shi*, *LIN Jian*
(Research Institute of Satellite Engineering and Technology, Harbin 150001, China)

Abstract: It is described the Network Data Delivery Service Based on Real-time Publish/Subscribe protocol, the means of data transferring by NDDS, the features and the application range of NDDS. Especially because of the special properties and functions of NDDS, it has more ability than any other network data delivery software: it can be used in data delivery in real-time system, and it can also be used in distributed real-time simulation system after customizing development.
Keywords: network data delivery; real-time; synchronization; RTPS; NDDS

引 言

以太网是现在最常见的网络，它处于现场总线的最底层，积极采用以太网技术是目前现场总线的发展趋势；同时，TCP/IP 协议借助互联网的应用已成为国际共同使用的网络语言。如果可以在以太网上实现分布实时系统，那就可以充分利用这两种技术的优势。但是目前通过以太网建立分布式实时系统所遇到的困难主要来自于基本的介入算法 CSMA/CD 和 TCP/IP 协议[1]。原因简单说来是使用 CSMA/CD 算法的时候，在网络繁忙、节点众多的情况下会有多个节点空闲等待的可能；另外，TCP/IP 在传送数据的时候是以牺牲时间来保证传输数据的正确，比如如果丢失了一帧数据，它就会不停的重发，浪费了时间。这样显然满足不了实时系统对于通讯时间比较高的要求。当然，目前在采用以太网、TCP/IP 网络技术构成的分布式网络系统中，传输的实时性随着快速以太网和交换式以太网技术的发展而得到了提高，但底层的网络实时性能的提高只是为分布式实时系统奠定了基础，而通信协议的执行策略仍然极大地影响着系统的实时性和同步性。

为了解决问题，只有考虑使用基于 IP 的软件方法来实现实时通信。NDDS（Network Data Delivery Service）即网络传输服务正是这样一种软件方法。下面就详细介绍一下基于 RTPS 协议的 NDDS 原理、性能、使用方法等。

收稿日期: 2004-03-15 **修回日期:** 2004-08-11
基金项目：国家 863-701 项目（2002AA715021）
作者简介：林晓辉（1966-），男，黑龙江省人，硕士导师，研究方向为微小卫星技术。**李实**（1976-），女，黑龙江省人，硕士生，研究方向为分布式仿真系统的调度管理。

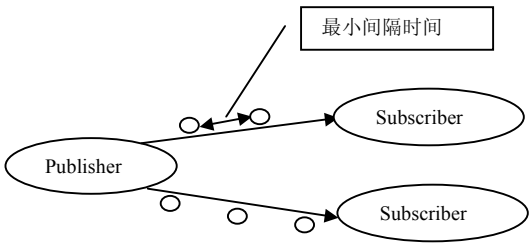
1 RTPS 和 NDDS

1.1 实时发布订阅协议 RTPS

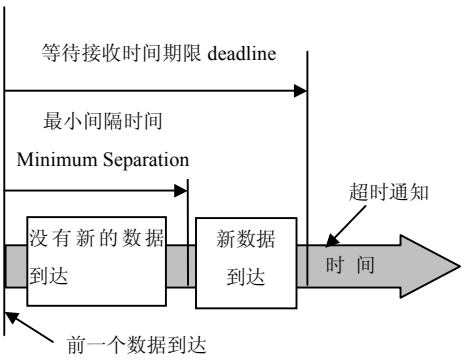
RTPS （Real-time Publish/Subscribe）所实现的是发布订阅（Publish/Subscribe）通信方式，这种针对实时通讯的协议所要完成的工作主要是：

- 对于每次数据通信控制用订阅者（Subscriber）和发布者（Publisher）进行发送接收。
- 允许应用程序衡量费时性和可靠性。
- 控制和设置内存的使用。
- 与实时操作系统环境进行配合。

图 1 是 RTPS 的 Subscriber 端工作原理[2]：Subscriber 是接收所订阅信息的程序；Subscriber 用最小间隔（Minimum Separation）这一属性来控制每一个数据发送的间隔；同时确定一个时间期限（Deadline），保证在此期限中如果没有新数据到达则要通知系统。



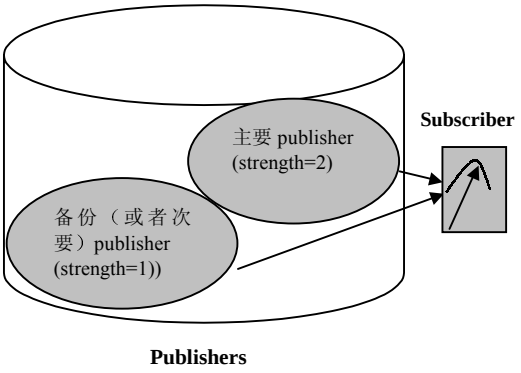
(a) Subscriber最小时间间隔属性图示



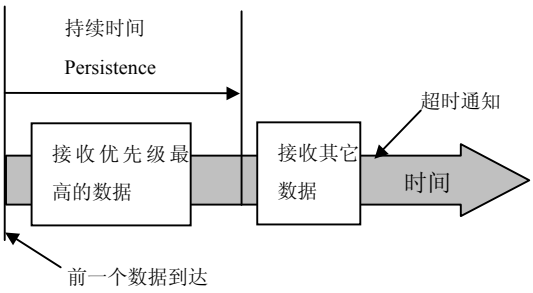
(b) Subscriber 的数据流控制方法

图 1 RTPS 中 Subscriber 的工作原理

图 2 是 RTPS 的 Publisher 端工作原理[2]：Publisher 是来发送 Subscriber 所订阅信息的程序；Publisher 的两个属性是权重（Strength）和持续时间（Persistence）， 权重即优先权，当多个 Publisher 发送数据时，Subscriber 按照优先权的高低来决定接收顺序；持续时间定义了数据的有效时间。



(a)多个 publishers 发送时的权重属性



(b) RTPS 按优先级接收的方法

图 2 Publisher 工作原理

这种 RTPS 协议的优点主要在于：

1) RTPS 比客户服务器模式在交换数据的延迟和带宽方面有显著的优势,它的带宽正好是服务器客户模式的两倍。而且是通讯过程是由数据驱动的，具有很好的实时性。

2)RTPS 对于建立多个节点，鲁棒性能要求高的可靠系统是理想的方案, 在一个新的节点加入或者废弃的节点离开的时候，RTPS 可自动调整网络，即所谓的 Plug and Play。

3)使用 RTPS 不需要网络编程, 数据传输中间件可以自行进行网络寻址。

1.2 数据传输服务 NDDS

NDDS（Network Data Delivery Service）是美国 Real-Time Innovations 公司研发的软件包[3]，它基于 RTPS 协议，具有通用的 API 接口，可以嵌入 VxWork 等实时系统的程序中来实现数据实时通信，也可以运行在 Windows 和 Unix 平台在不同主机上中传递信息。

NDDS 也可以称为一种网络中间件，它把应用程序和网络操作系统隔离开，形成接口程序，实现应用程序的独立性和对于各种网络操作系统的可移植性。

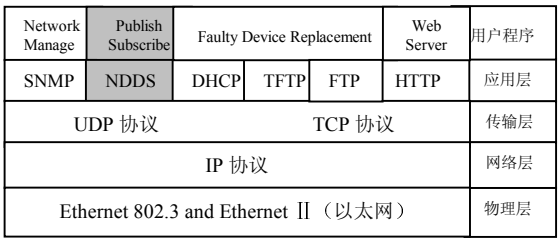


图 3 以太网络中常见的网络体系及协议

图 3 列出了在以太网络中常见的各种网络协议，可以看到 NDDS 在网络体系中的具体位置，它是建立在 UDP 之上的，不需要 TCP，这样就可以避免 TCP 的实时性不好的问题。而且有了 NDDS 之后传输层对于用户的应用程序就是透明了，用户的主要工作只是定义好传输的数据类型，而不用编制大量的网络程序。另外，从图 4[2]中可以看到 NDDS 在实时网络数据传输延时性能，是非常好的。

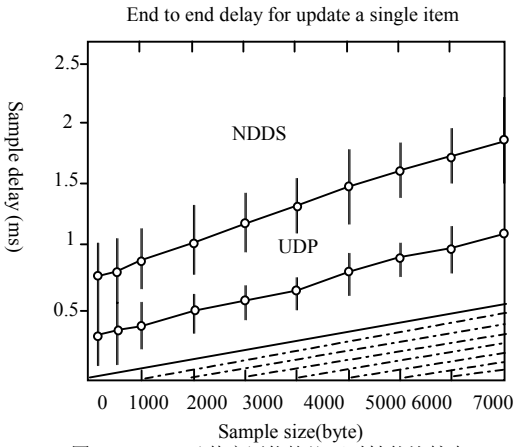


图 4 NDDS 及其它网络协议延时性能比较表

1.3 NDDS 在两种典型的分布式系统中的应用

1.闭环控制系统应用

使用 NDDS 可以实现两个或者更多的应用程序（即使应用程序在不同的节点上）以一个特定的速率传送数据，这一特性可以应用到分布式的闭环控制系统中[4]。

这种闭环控制系统主要的通信数据和其特性如下：

(1) 以一定采样循环的速率发送数据包。

(2) 数据的实时性非常严格。

(3) 敏感器要求敏感程度很高而且需要特殊的容错系统。

(4) 不同的节点上，处理过程必须是同步的而且延迟必

须很小，以避免相位丢失和不稳定。

(5) 如果数据包没有收到，通信必须提供故障处理。

2.远程监控系统应用

NDDS 可以用来远距离发送可靠的同步时间和命令，这样可以应用到需要通过一个激励而执行的有限状态的远程监控系统[4]。

这种远程监控系统主要的通信数据和其特性如下：

(1) 从图形用户界面到通信工具的通信必须是可靠的。

(2) 用户图形界面需要按照一定频率从通信工具那里得到数据。

(3) 命令必须很少。

(4) 发送命令的延迟必须很小。

在很多的分布式实时仿真系统构建中，都存在着两种典型系统的应用，完全可以应用 NDDS 实现所需要的仿真功能。

2 使用 NDDS 进行实时通信

安装了 NDDS 后，可以应用 MS VC++环境来实现 NDDS 的通信功能。即建立 Publication 发送一类数据给 Subscription，一个 Publisher 管理一系列的 Publications；建立 Subscription 向 Publication 订阅某类数据，而一个 Subscriber 管理一系列的 Subscriptions。NDDS 提供了一个数据库包含本节点（本地）一个域下面有效的 Publications 和 Subscriptions 的信息。所以要完成所需要的功能，就要编写代码创建和更新这个数据库。具体需要进行如下几步才能真正实现数据传输[5]：

(1)声明：即一个新的应用加入到网络中需要发送一个通知到所有和它进行 NDDS 数据传输的应用；这些和它相关的应用也都作为回答声明自己的 NDDS 对象(包括 Publications, Subscriptions 和 Servers)。当然一个已经在网络中的应用如果想创建一个新的 NDDS 对象也需要进行声明。

(2)创建 Publication:

a. 应该首先定义所要传输的数据类型。

b. 对于要传送的数据分配内存和初始化。

c. 需要使用 NDDSDomainDerivable() 函数构建新的 Domain 实例，并且赋予相应的值。

d. 使用 NDDSPublicationDerivable() 函数来创建 Publication，具体如下：

```
NDDSPublicationDerivable(
NDDSDomainBaseClass domain,
const char *nddsTopic,
class NDDSTypeClass *instance,
NDDSPublicationProperties *properties,
NDDSPublicationListenerClass *listener);
```

其中：Domain — 域名

NDDSTopic — 通信主题

Instance — 通信数据的实例

Properties — publication 的属性

Listener — 数据传输的监听类（可以在接收前获得数据传输的状态，捕捉一些有用的事件）

e. 设定 Publication 属性，其属性包括数据有效时间，权重，带宽等等。

f. 发送数据: Publication 发送数据使用的是 Send()函数。此函数的原型是

```
NDDSPublicationStatus NDDSPublicationDerivable::Send();
```

这个函数返回值如果是系统常数 NDDS_PUBLICATION_ERROR, 就是说此次发送数据失败了。

（3）创建 Subscription：

a. 应该创建所要传输的数据类型(如果还没有创建，如果已经创建过了就可以省却这一步骤)。

b. 对于要传送的数据分配内存和初始化。

c. 使用 NDDSDomainDerivable()函数构建新的 domain 类，并且赋予相应的值。

d. 使用 NDDSSubscriptionDerivable() 函数来创建 Subscription，具体如下：

```
NDDSSubscriptionDerivable(
NDDSDomainBaseClass *domain,
const char *nddsTopic,
NDDSTypeClass *instance,
NDDSIssueListenerClass *listener,
NDDSSubscriptionProperties *properties,
unsigned int multicastAddress);
```

其中各个参数的含义同 NDDSPublicationDerivable()。

e. 设定 Subscription 的属性，它的属性主要有订阅模式，最小时间间隔，接收队列大小等等

f. 开始接收订阅数据。

3 NDDS 提供的一些特殊函数和方法

3.1 Subscription 执行模式在数据交互通信中的应用

NDDS 提供了两种 Subscription 接收数据的执行模式立即模式(Immediate)和投票(Poll)模式。如果定义 Subscription 是投票模式来接收数据（即设置属性 Mode= NDDS_SUBSCRIPTION_POLL），数据发送出来时，NDDS 保存数据，Subsciption 并不是马上接收数据，而是当应用程序调用函数 NDDSSubscriptionDerivable::Poll() 时才接收数据。如果 Subscription 是立即模式（即设置属性 Mode= NDDS_SUBSCRIPTION_IMMEDIATE），订阅管理程序一旦接收到数据就产生中断给应用程序，应用程序立刻接收，保证数据的快速交换；相反如果等待的时间超过预设的期限，Subscription 的监听类就调用其相应的事件，按照用户需要处理。显然根据系统的需要，选择其中一种模式很容易实现实时同步通讯功能。

3.2 Publication 的特殊函数在交互通信中的应用

一个 Publication 可能需要确定存在订阅其数据的 Subscription 之后才发送数据，简而言之就是有接收者才发送。这个等待的动作可以通过使用下面的函数来实现：

NDDSPublicationDerivable::SubscriptionWait(RTINtpTime waitTime, unsigned int retries, unsigned int numberOfSubscriptions);

这个函数的三个参数具体介绍如下：

waitTime—连续两次发送数据间的间隔时间（秒级），为等待 Subscription 的而阻塞最大时间由这一公式定义：阻塞最长时间=retries · waitTime。

Retries—系统寻找 Subscription 的次数。

numberOfSubscriptions—期待的 Subscription 个数。

Publication通过此函数检查网络中已经声明的与其NDDS主题（Topic）相同的Subscription数量。如果至少有预先设置数量的Subscription，函数返回系统常数（实际为整数）NDDS_WAIT_SUCCESS，代表等待成功；否则函数阻塞，且不断查询系统直到Subscription数量满足再返回成功的标志；如果不停的尝试确定的次数，直到等待时间超过了预先设定的时间，这时尽管系统稳定函数也会返回系统常数NDDS_WAIT_FAILED代表等待失败；如果失败了，并且系统没有稳定下来（就是说如果设置一个更长的等待时间，也许会成功）函数返回系统常数NDDS_WAIT_PENDING代表可以补偿时间继续等待。另外一个返回值是NDDS_WAIT_INVALID表示出现异常，这种情况会指出错误的条件。

3.3 Subscription 数据监听类在数据交互通信中的应用

一个Subscription通过监听函数来得到数据传输的状态及新数据到达的通知，并且触发相应事件，完成指定的内容。这一函数是创建Subscription的时候由NDDS自动定义的，用户可以为要实现的功能修改代码，这样就可以控制数据到达后的操作。函数NDDSIssueListenerClass::OnIssueReceived()被调用的时间为：当一个Subscription到达网络时马上调用，或者应用程序向Subscription投票（Poll）请求数据时作为结果被调用。如果Subscription过了所定义的时间限制（deadline）后，还没有收到数据，此函数OnIssueReceived()也会得到通知。OnIssueReceived()的原型是：

RTIBool NDDSIssueListenerClass::OnIssueReceived(const NDDSRecvInfo *info,NDDSTypeClass *instance);

这样用户在编制自己的应用时，可以从得到的NDDSRecvInfo值判断数据接收的状态，采取相应的措施。NDDSRecvInfo的取值范围及其含义如表1所示[5]。

表1 NDDSRecvInfo的取值范围及其含义

数据状态	描述
NDDS_FRESH_DATA	执行正常；数据是新的
NDDS_NEVER_RECEIVED_DATA	超时；没收到任何数据；忽略数据
NDDS_NO_NEW_DATA	超时；没收到数据；缓存保持最后数据
NDDS_DESERIALIZATION_ERROR	一个数据到达，但是分解失败

3.4 可靠模式传输在数据交互通信中的应用

大多数的通信体系结构提供一个或者更多的可靠模式，但是不是所有的模式都适合实时系统。对于不同的系统，发布者程序和订阅者程序之间的通信对于实时性和数据传输可靠性之间的平衡要求不同。

NDDS针对这个问题提供了一种机制就是对于每一个基本订阅者都可以订制一个实时性和可靠性比例合适的有效模式。在NDDS可靠性模式下，压缩了内存的利用率，因为多余的内存缓存都用来重发、限时，所以NDDS提供的机制对于应用程序预测和控制内存的利用都很有必要。

NDDS的可靠性模式下，数据通信是由Subscription驱动的。一个发布者程序不能直接决定发送的数据是否可靠。但是，发布者程序的内存和所控制数据流需要一些特定的设置。这就是说如果有匹配的订阅者程序希望从这个发布者程序按可靠模式接收数据，NDDS会应用可靠性模式发送这些数据。

4 应用实例

Constellation是RTI公司研制的一种完整建模、仿真软件开发环境，同时也是设计和实现实时应用的开发工具箱。利用Constellation的基于模块化的计算方法、图形部件设计工具、部件库管理工具、代码生成器、运行执行器、运行调试工具可以快速又容易地构造和运行实时应用软件[6]。特别是这个软件集成了NDDS，即在Constellation中具有一些可以实现NDDS功能的组件。这些组件由C++代码写成，用户也可以自己创建组件在其中编写代码实现所需的NDDS功能。

利用Constellation和NDDS相结合可以实现分布式仿真系统的数据传输和运行同步。如下建立一个卫星姿态控制分布式仿真系统，由动力学模块、控制律模块和执行机构模块三部分组成。动力学模块主要是求解欧拉动力学和运动学方程，控制律采用角度和角速度增益反馈，执行机构使用一个增益作为参量输入。这三个模块之间数据传输就是利用NDDS；同时为了实现三个模块之间的同步与协调运行，利用NDDS的实时特性，在主控模块和其他三个模块之间传送同步时钟信号。在其中利用了前面提到的一些NDDS的特殊技术，来控制系统的状态转换和流程控制等功能。具体的系统结构如图5。

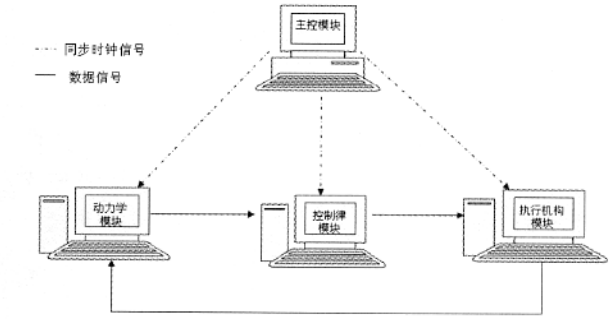


图 5 卫星姿态控制系统结构

经过对于此系统的仿真运行得到运行结果曲线如图6（角度输出值曲线，其中a为偏航角，曲线b为俯仰角，曲线c为滚动角）。为了验证仿真结果，利用Matlab中的Simulink建立的集中式的卫星姿态控制仿真系统，采用的数学模型和前面一样。得到其角度输出曲线和图6一致，而且通过记录实时仿真数据，证明了仿真结果是相同的。

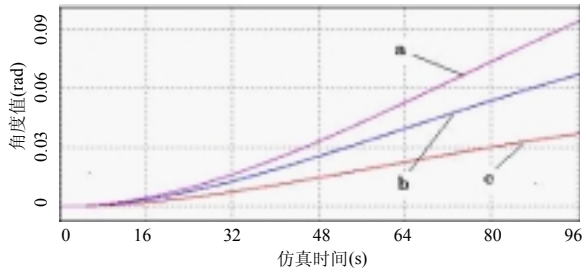


图 6　仿真结果曲线

5　结论

NDDS是一个有效的实时应用通信平台，它实现了RTPS协议，并且提供了一些特殊的方法和函数，使得在实时系统中实现交互通信工作更加容易，并且用户可以在这个平台上开发更加适合自己的实时通信程序。这样在分布式实时系统中采用NDDS作为网络中间件，对于通信和控制方面的系统研发将带来很大的贡献。

参考文献：

[1] Gerardo Pardo-Castellote，Stan Schneider，Mark Hamilton. Can Ethernet Be Real-Time[R]. Real-Time Innovations White Paper. 1998. 11.

[2] Gerardo Pardo-Castellote, Stan Schneider, Mark Hamilton. NDDS: The Real-Time Publish-Subscribe Middleware [EB/OL]. http://www.rti.com info@rti.com. 1999. 8.

[3] Jerey Biesiadecki Abhinandan Jain. A Recongurable Testbed Environment for Spacecraft Autonomy [EB/OL]. http://nasa.gov/pdf/estec96.pdf. 1996.

[4] Developing Distributed-Control Applications with NDDS and CORBA [Z]. Real-Time Innovations, Inc. 2002, (6).

[5] NDDS® Network Data Delivery Service. The Real-Time Publish-Subscribe Connectivity Solution User's Manual [Z]. Real-Time Innovations, Inc. 2002, (9).

[6] Constellation® User's Manual (Constellation Version 1.0)[Z]. Real-Time Innovations, Inc. 2002, (10).

（上接第1103页）

通常，实时仿真的仿真步长取值较小，有的仿真步长甚至在μS 级。如果$T_h = 100\mu s$，则一秒钟要仿真 10000 步。由于计算机字长的限制，在计算过程中不可避免地存在截断误差，长时间的仿真运行，截断误差的累积可能导致仿真不稳定[1]。

另一方面，仿真系统采用的时间函数的精度和分辨率制约着帧同步误差，如果采用闭环方式控制帧同步，帧同步精度可达到时间函数的精度。在仿真运行过程中，帧同步误差一般都正向累积[2,4]，长时间的仿真运行可能使得帧同步误差累积得不可容忍。以某火箭的助推段状态控制系统仿真为例，仿真时间为 128s，仿真步长为 4ms，帧同步精度为时间函数精度$T_{syna} = T_{funa} = 100\,ns$，仿真结束时帧同步误差累积量为 3.2ms，几乎等仿真步长[2,4]，如果仿真时间更长，时间函数的精度更差，帧同步误差的累积将更不容忽视。

稳定的实时仿真系统还要求基于同样的初始条件，仿真系统多次运行的仿真结论基本一致，至少不能产生相互矛盾的仿真结论，否则系统也不满足稳定性条件。

5　结论

实时仿真是仿真的高级模式，是仿真理论研究和工程运用的重要领域。随着实时仿真在航天、航空、国防等领域得到越来越广泛的应用，实时仿真系统的可信性问题也受到了更广泛的关注。实时仿真系统的可信性不仅取决于系统逻辑计算的正确性和系统的稳定性，而且取决于系统的实时性。本文从方法论的角度出发，讨论了实时仿真系统的验证问题，给出了实时仿真系统可信性的验证过程，重点研究了实时性、逻辑计算正确性及稳定性的验证，并给出了相关操作步骤和计算模型。本文对实时仿真理论研究和工程运用有一定的参考价值。

参考文献：

[1] K.H.(Kane) Kim, Juqiang Liu and Inho Kim. Distributed Object-Oriented Real-Time Simulation of Ground Transportation Networks with the TMO Structuring Scheme[C]. Proceedings of Computer Software and Applications.1999:130-138.

[2] 寥瑛，梁加红，姚新宇等. 实时仿真理论与支撑技术[M]. 湖南长沙：国防科技大学出版社，2002.

[3] Department of Defense Directive 5000.59-P. DoD Modeling and Simulation (M&S) Master Plan[Z].1995,10.

[4] 姚新宇，黄柯棣.半实物仿真系统的实时性分析[J].北京：计算机仿真.1999.16(4):51-54.

[5] Roger Champagne, Louis-A.Dessaint, Handy Fortin-Blanchette et al. Analysis and Validation of a Real-Time AC Drive Simulation[J]. IEEE Transactions on Power Electronics. 2004, 19(2): 336-345.

[6] Jens Herrmann. Guideline for Validation & Verification Real-Time Embedded Software Systems [EB/OL]]. http://www.dess-itea.org, 2002.

[7] D.paré, G.Turmel,-C.sourmagne et al. Validation Tests of The Hypersim Digital Real Time Simulator with a Large AC-DC Network[C]. International Conference on Power Systems Transients-IPST. 2003. NEW Orleans, USA:1-6.