

分布式实时仿真系统中基于 PN 的同步模型^{*}

Petri Nets Based Synchronization Model for Distributed Real-Time Simulation System

许建峰 胡宁宁 朱晴波 胡宁 谢立

(南京大学计算机科学与技术系 南京 210093)

(南京大学计算机软件新技术国家重点实验室 南京 210093)

Abstract Synchronization is still an open problem in distributed real-time simulation systems. In such kind of systems, application tasks are allocated on distributed processors. Because of such factors as communication delay and task failure, the multiple source data a task needs from several other tasks could not reach the task at required times. That is, they are not synchronized. Generally, time prediction is used to postpone the sending time of some data to achieve the goal. But this method is not adaptive because the real communication delay could not be predicted accurately. In this paper, the authors propose a new synchronization model—EGM (Entity Group Model). EGM abstracts the application tasks in a distributed real-time system as entities and proxy and uses proxy to take the advantages of both the central strategies and the distributed strategies. We also give the method to describe EGM formally using Petri Nets.

Keywords Synchronization, Petri nets, Real-time, Simulation, Distributed systems

1 引言

基于不同的应用背景与假设,人们已对分布式实时仿真系统中任务间的同步问题进行了大量研究,其主要思想是将系统看成一个离散事件的集合,事件之间存在某种相互依赖的关系,执行时遵循一定的时序约定,在不破坏这种约定的前提下,尽可能提高系统的并行度。基于此,人们提出了众多解决办法^[1],概括而言,可分为 Optimistic 方法^[2-9]与 Conservation 方法^[6-8]两大类。Time Wrap(TW)^[2]是前者的典型,可以很好地控制保存状态所需的开销,因而被广泛应用^[3-5],但它在出现因果关系倒置后所需的 Rollback 处理代价很高,仍不尽如人意,对此人们一直在进行多方面的改良工作^[13-14];后者面临的最大问题是难以避免死锁的产生,但仍可在某些特定问题上发挥较大作用。

不论是 Optimistic 方法还是 Conservation 方法,都有一个共同的缺陷,即它们并不能很好地应用于分布式实时仿真系统。在这类系统中,数据之间的匹配性十分重要,然而由于分布特性,一个节点从多个其它节点上获得的信息往往由于通信延迟等多种原因不能严格匹配,这便引起仿真系统计算精度的下降。因此对于实时仿真系统而言,在处理同步问题时不能简单地追求高并行度,而应更注重仿真的客观性或逼真性。时间预测机制是实时系统经常采用的同步方法^[10-12],这种方法通过对自身消息的延迟来逼近所要求的同步状态。但它也有明显的缺陷:①任务需要了解其他任务的处理状况,如处理时延信息等。②由于路由的不确定性,通信开销预测准确度往往很低。③容错性不强,无法处理某些任务突然死亡的情况。鉴于此,我们尝试通过对系统结构的改变来达到较好的同步效果。本文在介绍这一方法的同时,还给出它的形式化描述工具,以便于进一步的性能分析。

^{*} 本文得到国家“863”高科技计划资助。朱晴波 硕士研究生,主要研究方向为分布式系统。许建峰 博士研究生,主要研究领域为雷达软件工程和分布式系统。胡宁宁 硕士研究生,主要研究领域为分布与并行计算机系统、网络系统。胡宁 硕士研究生,主要研究方向为分布式操作系统。谢立 教授,博士生导师,主要研究领域为分布与并行计算机系统。

2 实体组模型

2.1 分布式实时仿真系统的构成

雷达系统仿真是分布式仿真领域的热点之一,这里我们给出一个分布式雷达仿真系统 RDSP,其框架如图 1 所示。系统硬件由若干台通过网络相连的计算机组成,每台计算机都可相对独立地仿真若干个系统模型。系统的运行单位称为任务,每次仿真一般由多个任务共同完成,这些任务根据调度策略分配在分布式环境中的各台处理节点上,按一定的时间顺序并行运行。任务之间的通信受系统中同步机制的管理,任务能够实时处理外部请求,形成处理和仿真结果传送到外设。各个外设是仿真系统的最终用户,仿真系统的各项性能将体现在每一外设所接收仿真数据的真实性和精确性上。

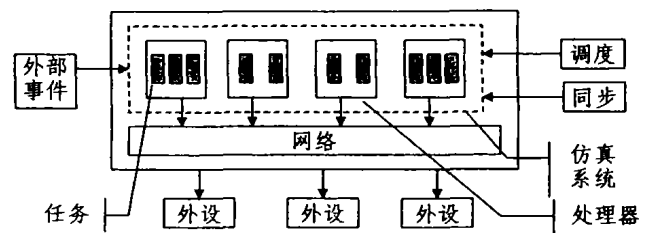


图 1 雷达仿真系统 RDSP

2.2 RDSP 中的同步问题

在引入 EGM 之前,我们首先描述一个在 RDSP 中运行的典型实例。图 2 是一个机载脉冲多普勒雷达主杂波跟踪系统的简化模型,其中飞机姿态模块提供飞机的姿态和速度信息(V),天线指向模块提供天线在机体坐标系中的角度指向信息(A),回波模块提供雷达回波信息,主要包含了地物回波的频率信息(f)。所有这些信息都要传送给雷达计算机 RC 进

行处理(如图2中实线箭头所示),且三者之间必须满足一定的关系式:

$$f = g(V, A) \quad (1)$$

但在系统仿真中,由于传输延迟及计算误差等因素,这些参数难以严格遵循这样的关系,即传送到 RC 的数据之间难以完全匹配,因此地物回波频率 f' 与 RC 根据 V, A 计算出的频率值 f 之间将会存在差值 $\Delta f = f - f'$, 这样的差值便直接影响到 RC 的客观处理效能,因此在仿真过程中 Δf 的大小是判断仿真系统运行质量的主要标准。

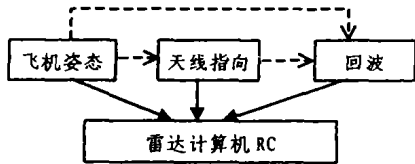


图2 同步实例

可见在 RDSP 中,必须引入同步机制以解决数据匹配问题,才能客观真实地实现系统仿真。对此,常用的一种方法是根据已知的数据依赖关系,用大量经验数据对各条路径的通信延迟进行预测,然后延迟相关数据的发送时间来改善它们之间的匹配关系,这便是时间预测同步模型 TPM 的基本思想。该模型实际上是一种分布式的处理方法,它的优点是简单易行,但确也存在前文述及的一些不足之处,特别是当系统中外设数量很多时,各种数据延迟将难以适从。一般地,集中处理是分布式系统中提高计算精度的最主要的方法,为此我们考虑将以上分布式处理的操作以集中式的方法实现,在 RD-SP 设计中提出了实体组模型(Entity Group Model, 简称为 EGM)。其基本思想为:系统中的所有模块仍以分布的方式运行在各自的处理节点上,同步只对系统中那些相互之间存在数据依赖关系的任务集合进行,同时在系统对每一外设设置一特殊任务,我们称为代理,专门接收相关任务组中任务向外设发送的数据,由它对这些数据进行匹配,最后按照实际物理系统中的时间关系将数据传送给外设,从而最大限度地模拟实际系统的运行。

2.3 实体组模型及其概念定义

在 EGM 中,所有任务称为“实体”(entity)。为了区别外部设备与仿真系统中的仿真任务,我们称外设为“外部实体”(External Entity),系统中的仿真任务为“内部实体”(Internal Entity)。EGM 中集中处理的只是那些具有数据依赖关系的任务组成的集合,这种集合关系实际上是由外设决定的,因为外设的运行往往依赖于多个内部实体同时提供数据,在此情形我们称这些实体相对于这个外部实体具有“配对关系”。具有配对关系的多个实体组成一个“实体组”(Entity Group, 简称 EG)。在图2的示例中,飞机姿态模块,天线指向模块和回波模块都是内部实体,它们之间具有配对关系,并且相对于 RC 构成了一个实体组。

尽管客观世界中的实际模型能够相对独立地工作,但在实时仿真系统中,总能选择一个相对特殊的任务,由它启动整个系统的工作。例如在图2中,飞机姿态模块就是这种特殊任务,因为其他两个任务的计算都要直接或间接地依赖于它的输出结果。在 EGM 中,称这样的特殊实体为“主动实体”。

在 EGM 中,我们还设置一个专门的特殊实体来管理由实体组中的内部实体向外部实体发送的数据,称其为“代理”(Proxy),通过它来完成具体的同步算法。具体地,Proxy 参与

系统的实时调度,所有实体向外部实体提供的数据都首先传送给 Proxy,由其根据外部实体对各种数据的时间特性要求进行输出。图2所示实例的 EGM 可用图3表达,其中 E_1, E_2, E_3 分别代表飞机姿态模块、天线指向模块与回波模块,EE 表示 RC, Proxy 为该实体组的代理。

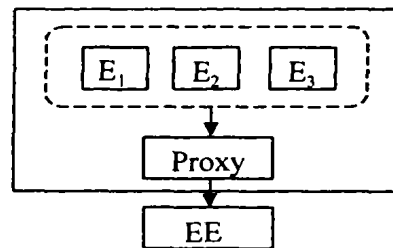


图3 EGM 实例

2.4 EGM 的形式描述

本节给出 EGM 所使用的形式描述工具-EGPN, 它以 PN 为工具对分布式实时系统中任务间的同步进行描述。

(1) 实体 PN 的描述 系统的同步机制都与其通信机制有着密切的关系,在 RDSP 中,任务之间的通信使用信箱机制,即发往某任务的数据存放在该任务的信箱之中,当其被调度运行时发现信箱非空后就进行处理并将计算结果通过信箱传送给其它相关任务。为了描述这种通信机制,我们用图4所示的 PN 表示实体, mp 代表实体内部的信箱, dp 代表计算结果,由于一个实体的结果可能送给多个其它内部实体或外部实体,故一个实体内可以包含多个 dp。

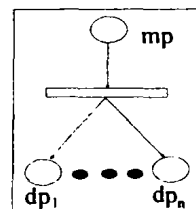


图4 实体的 PN 表示

(2) EGPN 的定义 定义 EGPN 如下:

$$EGPN = (P, T, A, M, R, C)$$

$$P = \cup P_{ie} \cup \{P_{ee}\}, P_{ie} = \cup \{mp_i, dp_{i1}, \dots, dp_{ik_i}\};$$

$$T = \{t_s\} \cup T_{prs} \cup T_{in_comm} \cup T_{ext_comm};$$

$$T_{prs} = \cup \{t_{prs_i}\}, T_{in_comm} = \cup \{t_{ij} | E_i \rightarrow E_j\}, T_{ext_comm} = \cup \{t_{ext} | E_i \rightarrow EE\};$$

$$A \subseteq \{P \times T\} \cup \{T \times P\}; M = \{m_p | \text{对所有 } p \in P\};$$

$$R = \cup \{(f(t_i), time(t_i)) | t_i \in T_{prs}\}; C = \cup \{cost(t_i) | t_i \in T_{ext_comm}\}.$$

其中 P_{ie} 是内部实体库所集合,每个 P_{ie} 都由一个信箱库所与若干个数据库所组成; P_{ee} 是外部实体库所。 t_i 是 EGPN 的激发变迁,即激活该主动实体的变迁,在这里我们暂时只考虑一个 EGPN 中只有一个主动实体的情况。 T_{prs} 是实体处理变迁集合,由各实体的 mp 激活,再由该变迁激活相应实体的 dp; T_{prs} 中的每一个变迁 t_i 都对应于 R 中的一对参数 $(f(t_i), tm(t_i))$, 其中 $f(t_i)$ 表示 t_i 所代表的实体操作, $dp_{ij} = f(t_i)(mp_{ij})$, $tm(t_i)$ 为其处理时间。 T_{in_comm} 是内部实体间的通信变迁集合。 T_{ext_comm} 是内部实体向外部实体发送仿真数据的变迁集合,对

应于该集合中每一变迁 t_{ext}, C 中都有一个元素 $cost(t_{ext})$, 表示其通信延迟。

2.5 EGM 实例

图 5 是用于描述图 3 中 EGM 的 EGPN。三个实体 E_1, E_2, E_3 分别用虚线圈出。EGPN 中的激发变迁作用于该系统的主动实体 E_1 。不失一般性, 这里我们只对 E_1 进行解释, 由图 2 知 E_1 自身的处理结果要发送给 E_2, E_3 以及该实体组的外部实体 EE, 因而 E_1 的实体 PN 中有三个 dp, 分别与 E_2, E_3 和 EE 对应。此外, E_1 中实体处理变迁参数 $(f(t_{pr1}), tm(t_{pr1}))$ 还给出 E_1 内部的处理细节。这样, 图 5 便清楚地刻画了 RDSP 中各任务之间的通信关系以及任务内部的处理细节, 为同步操作提供了详细信息。

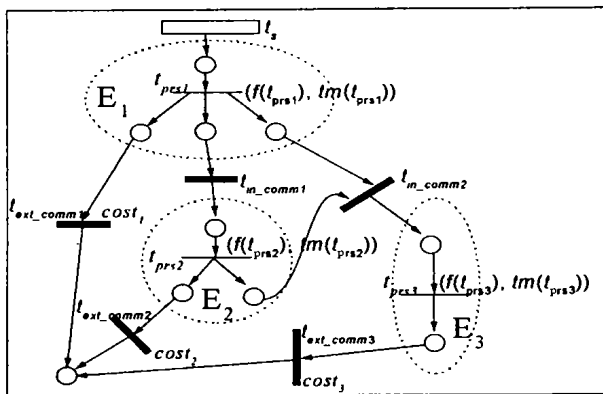


图 5 EGPN 实例

3 RDSP 中同步模型 EGM 的实现

在 RDSP 中, EGPN 所提供的信息可以用一些表格来存储以便于算法实现的简化, 我们称这些信息表为协同配置表, 它们又根据其任务种类的不同分为两类: 实体协同配置表与代理协同配置表。

3.1 实体协同配置表

在 RDSP 中, 每一个实体任务的协同配置表由 Input Table 与 Output Table 两张表组成。Input Table 保存了一实体进程 E 进行处理所需要的输入数据及这些数据的源实体, 由 EGPN 易知, 源实体可以表示为 $\{i | dp_i \in ('mp)\}$, 其中 $mp \in P_i$ 。同理, Output Table 指明该实体处理结果应传送的目标实体, 用 EGPN 表示为 $\{i | mp_i \in (dp)\}$, 对所有 $dp \in P_i$, 同时输出数据与输入数据的函数依赖关系也可由 EGPN 中 R 得到, 即 $dp = f(t)(mp)$ 。

3.2 代理协同配置表

在 RDSP 中, 对每一实体组创建一周期性任务来完成代理实体的功能, 由它统一处理实体组传往外部实体的数据, 协调各数据间的时差关系。代理任务的协同配置表 Proxy Table 根据 EGPN 所提供的信息创建, 它包括向外部实体提供数据的内部实体编号项及相应的数据传输延迟项, 即 $\{i | dp_i \in ('p_{ee})\}$ 与 $\{cost(t) | t \in T_{ext_comm}\}$, 代理任务则根据 $cost(t)$ 升序将实体在表中排列存放。

3.3 EGM 同步算法

RDSP 采用静态的周期调度策略, 每个任务被调度时, 首先查看自身信箱空否, 在非空时根据输入数据进行具体处理。由于代理也是一周期性任务, 并且对周期性任务而言, 其输入数据实际上的最小时间单位为一个调度周期, 因此代理配置

表中数据传输延迟项的单位是周期。

设 Proxy Table 中各实体按数据传输延迟项排序如下: $t_{e1} \leq t_{e2} \leq \dots \leq t_{en}$ 。于是 Proxy 在时间延迟上的仿真按如下方式实现: 当一个实体组中所有实体的数据都到达时, 首先将 E_1 的仿真数据 d_{e1} 传送给相应的外部设备, 相隔 $|t_{e2} - t_{e1}|$ 个周期后, 将 d_{e2} 输出, 依此类推。

值得注意的是, 由于多种原因, 一个 Proxy 的信箱中有可能同时出现一个实体组的两套(或多套)数据:

$$\textcircled{1} \{(d_{e1}, t_{11}), (d_{e2}, t_{12}), \dots, (d_{en}, t_{1n})\}$$

$$\textcircled{2} \{(d_{e1}, t_{21}), (d_{e2}, t_{22}), \dots, (d_{en}, t_{2n})\}$$

(其中 t_{iv} 为数据的到达时间)。并且由于分布式系统中通信的不确定性, 还有可能会出现 $t_{11} < t_{21}$ 但某些 $t_{1j} > t_{2j}$ 的情况, 因而不能简单地以到达时刻来判断多组数据间的配对关系。为此我们在 EGM 中引入触发时刻标识 STI (Stimulation Time Identity)。在 EGPN 中, 每个实体组都有一触发变迁 t , 由主动实体激活, 当主动实体发现本 mp 激活进而将激活整个实体组时, 就将当前周期号作为此次实体组工作的触发时刻标识, 即 STI。代理便能够利用 STI 来正确区分不同的配对关系, 对两组实体组数据 STI_1 和 STI_2 , 且 $STI_1 < STI_2$ 时, 第二组数据(STI_2)要在第一组数据(STI_1)全部发完后才可发送。对同一实体组数据, 代理实体将延迟组中数据的发送直到该组中所有数据都已到达, 然后再根据组中数据间的时间相对延迟依次发送给外部实体。由于篇幅所限, 具体的算法代码本文从略。

4 初步性能分析

首先我们定义一列自 t_s 至 t_{ext_comm} 的变迁所组成的序列为 t_{ext_comm} 的导引变迁链(简称导引链), 记为 l_{tlj} , 例如本例中

$$l_{tl31} = (t_s, t_{pr1}, t_{in_comm1}, t_{pr2}, t_{in_comm2}, t_{pr3}, t_{ext_comm3})$$

$$l_{tl32} = (t_s, t_{pr1}, t_{in_comm2}, t_{pr3}, t_{ext_comm3})$$

都是 t_{ext_comm3} 的导引链。同时, 我们记

$$LTL_i = \cup \{l_{tlj}\},$$

为 t_{ext_comm} 的导引链集, 下面我们就利用这些概念对 EGM 的主要同步性能进行估计。

4.1 响应时间

前文已经使用 $tm(t_{pr})$ 来表示变迁 t_{pr} 的处理延迟, 它实际上便是 t_{pr} 的响应时间。对其进行扩充, 我们用 $tm(t)$ 来表示 t 的响应时间, 这里 t 可以是变迁、导引链或导引链集。稍加分析可知, 对内部通信变迁 t_{in_comm} , 在不同的导引链中将有不同的响应时间, 即 $tm(t_{in_comm})$ 会由于所在导引链的变化而变化, 但这并不影响我们所关注问题的实质。进一步地, 我们定义在一个仿真周期内, 从变迁 t_s 被触发至库所 P_{ee} 收到仿真数据的时间延迟为系统响应时间, 记为 T_m , 它表征了仿真系统对于外部实体的反应能力。于是, 我们易于得到下列关系:

$$tm(l_{tlj}) = \sum tm(t), \text{ 对所有 } t \in l_{tlj},$$

$$tm(LTL_i) = \max \{tm(l_{tlj}) | l_{tlj} \in LTL_i\},$$

$$T_m = \max \{tm(LTL_i) | i = 1, 2, \dots, n\}.$$

由于一般情况下每一 $tm(t)$ 都是一随机时间序列, 因而这里的估计具有很大随机性。对典型情况, 如每一个处理和通信变迁响应时间均为一个周期时, T_m 便为所含变迁最多的导引链中的变迁数。

4.2 异常概率

在分布式实时系统中, 无论是处理机的内部处理还是机

间通信,都可能由于多种原因导致任务执行的夭折或传输数据的丢失,它们对应于 EGNP 中某一变迁失去作用,我们统称这样的现象为一次异常,将其出现的概率称为异常概率。对于大多数系统而言,异常并不导致灾难性后果,但必定引起系统性能的下降,因此异常概率的转移及其对系统性能的影响是我们关注的重点,有必要通过各个变迁的异常概率来分析导引链及其集合的异常概率,从而尽可能避免那些对系统性能至关重要的链路出现高的异常概率。在 EGNP 中,若每个变迁的异常是一系列相互独立的事件, $Pf(t)$ 为 t 的异常概率,于是我们有

$$Pf(tl_{i_0}) = 1 - \prod (1 - Pf(t)), \text{ 对所有 } t \in tl_{i_0}$$

$$Pf(LTL_i) \leq 1 - \prod (1 - Pf(tl_{i_0})), \text{ 对所有 } tl_{i_0} \in LTL_i$$

这里没有使用等号是因为考虑到不同的导引链之间也可能存在共同的处理或通信变迁。当各个变迁异常相互间不独立时,情况要复杂一些,需要根据具体情况得到特定结果。

4.3 内部实体间的同步

至此我们实际上仅讨论了内部实体对外部实体的同步问题,这是因为由一系列内部实体构成的仿真系统服务于客观存在的外部实体这一模式具有广泛的应用背景,所以 EGM 对于内部实体彼此之间的同步方法并未作严格的限制,我们可以根据具体的应用特点通过现有的 conservation 或 optimistic 方法解决内部的同步问题。

另一方面,内部实体与外部实体之间的关系本身就具有相对性,即一个内部实体 E_1 与另外一组内部实体 E_2 和 E_3 可以构成一实体组,作用于其它的外部实体;同时它也可相对于 E_2 和 E_3 成为一外部实体。所以 EGM 的基本思想适用于内部实体彼此之间的同步,问题仅在于我们需要根据各个导引链的结构确定相应的主动实体并加入必需的 Proxy,由于 Proxy 只执行数据的存储与延迟,因此由此增加的负担对大多数系统而言是可以承受的。

4.4 与处理机及任务数量的关系

根据前面的描述我们可以知道,决定 EGM 同步性能的是一系列变迁的响应特性。一般地,在导引链中,除 t_i 与 t_{ext_comm} 外,变迁 t_{pri} 与 t_{in_comm} 交替出现,每一变迁 t 的响应时间 $tm(t)$ 是一个随机时间序列。 $tm(t_{pri})$ 是仿真节点的内部处理延迟,按负载均衡的原则,系统内部处理机的个数越多,单一处理机的负载越轻, $tm(t_{pri})$ 的随机分布便越简单; $tm(t_{in_comm})$ 是各个内部实体之间的通信延迟,承担仿真任务的处理器个数越多,机间通信的负担便越重, $tm(t_{in_comm})$ 的随机分布将越复杂。所以,与其它分布式系统一样,EGM 同步性能与处理机数量及任务量之间的关系难以用简单的关系式表达,而是取决于系统结构及仿真模型的复杂性。对一具体系统,通过对每一 $LTL(t_{ext_comm})$ 中导引链的结构与响应时间分布特性的分析,我们能够进一步地分析其同步性能。

5 测试结果与分析

我们对 RDSP 分别使用 TPM 和 EGM 通过蒙特卡罗方法进行了测试与比较,在仿真过程中假设飞机如图 6 所示在一水平面内首先沿 X 轴方向由 A 点直线飞行到 B 点,然后从 B 点以匀角速度盘旋经 C 点到 D 点,最后沿 Y 轴方向由 D 点直线飞行到 E 点;机载雷达天线按图 7 所示的轨迹进行循环扫描;同时假设各任务异常概率符合二项分布。对 TPM,当任务出现异常时不能为外设提供新数据,由于其它任务此时仍提供新数据,故数据间不匹配性将增大。对 EGM,如果某一或

某些任务出现异常,Proxy 将延迟或舍弃同组数据以保证数组间的匹配性。

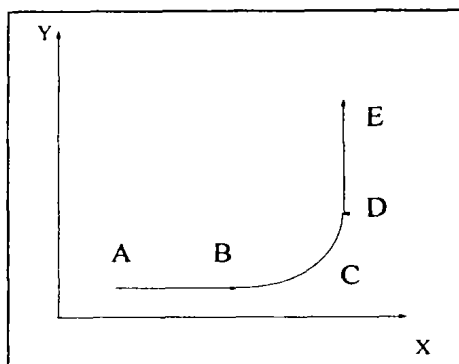


图 6 飞机的飞机轨迹

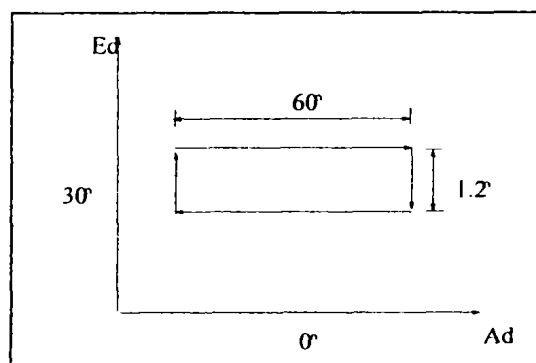


图 7 天线扫描轨迹

正如第二节所述,测试过程中我们主要以主杂波频率偏差 $|\Delta f|$ 为参照。图 8 以一航程中 $|\Delta f|$ 均值为指标,标注了任务异常概率由 0.01 逐渐变化到 0.15 的过程中 $|\Delta f|$ 均值的变化情况。由图可见,对各种任务异常概率 TPM 下频率偏差明显大于 EGM 下的频率偏差。并且随着任务异常概率的增大,TPM 下 $|\Delta f|$ 的均值也逐渐增大,而 EGM 下却没有明显变化。

图 9 是在 E_1, E_2, E_3 发生异常的概率都为 0.1 的条件下,两种模型中 $|\Delta f|$ 随时间的变化情况。其中时间段 $(0, t_1), (t_1, t_2), (t_2, t_3), (t_3, t_4)$ 分别对应于图 6 中飞机飞行的 (A, B), (B, C), (C, D), (D, E) 段。可以看出在 $(0, t_1), (t_1, t_2)$ 和 (t_3, t_4) 三个阶段 EGM 的优越性较明显。在 (t_2, t_3) 阶段,两种模型都有较大的偏差,这一点可从图 10 中找到原因。图 10 是飞机在整个过程中 f 的理论值随时间的变化图,从中可以看出图 9 中的 (t_2, t_3) 阶段对应于图 10 中频率的急剧变化阶段,此时 Proxy 难以提供与当前理论值差距较小的匹配数组,所以 EGM 对于频率剧变的适应性仍较差。

结论 本文针对分布式实时仿真系统中的同步问题应用 PN 提出了模型 EGM,与其它同步模型相比,它具有以下特点:①只对多任务进行整体上的考虑,任务之间的通信不影响同步处理;②不须使用全局时间信息,因为所有时间都是本地时间,同步隐含在不同实体间的数据传输之中;③该模型不仅具有较好的形式描述与分析能力,而且很容易进行实现中的转换,可以具体地应用于系统的开发与设计。进一步的工作可以从以下几个方面着手:①对 EGNP 进行扩充以实现多主动实体实体组;②推而广之使一个实体可以成为多个实体组的成员;③优化同步算法使仿真数据输出早于实体组数据的完

全到达以改善系统的实时性;④寻找更有效的算法使其更加适用于仿真参数剧烈变化的系统。

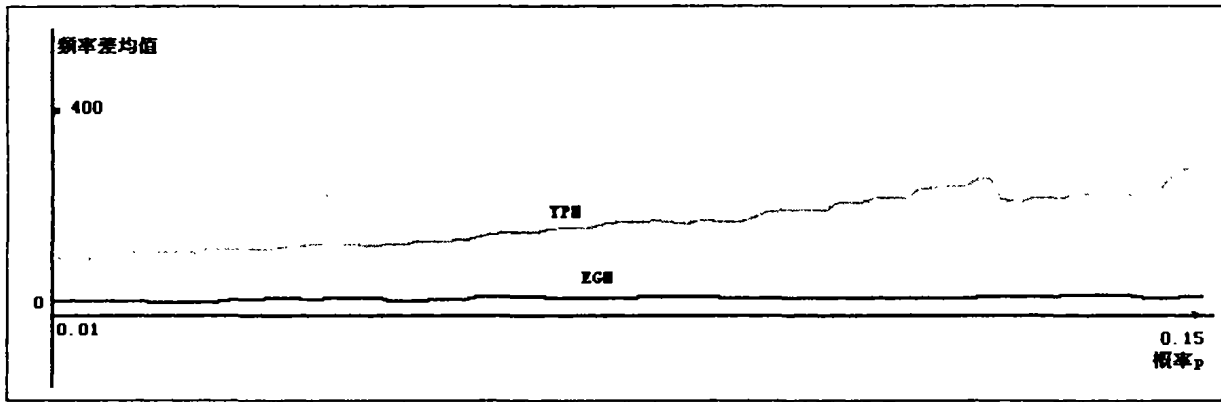


图8 两种模型中频率差均值随任务异常概率的变化

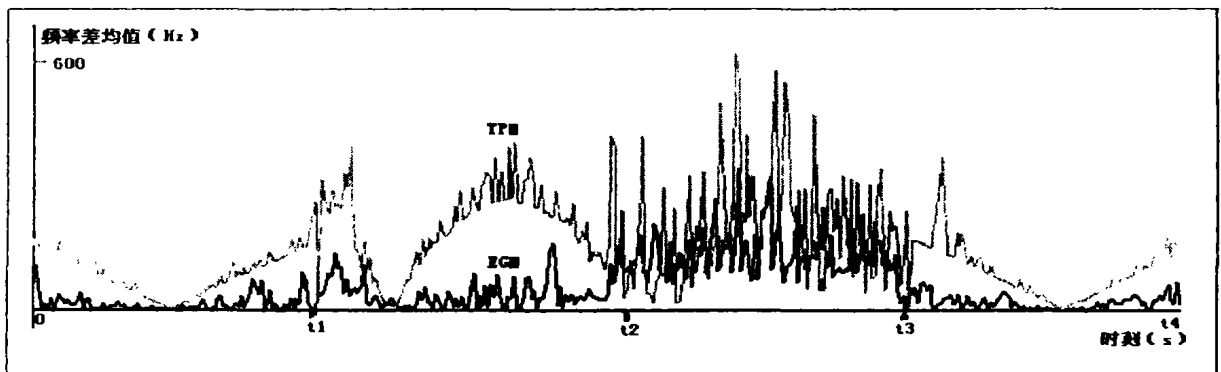


图9 恒定任务异常概率下频率差在整个飞行过程中的变化

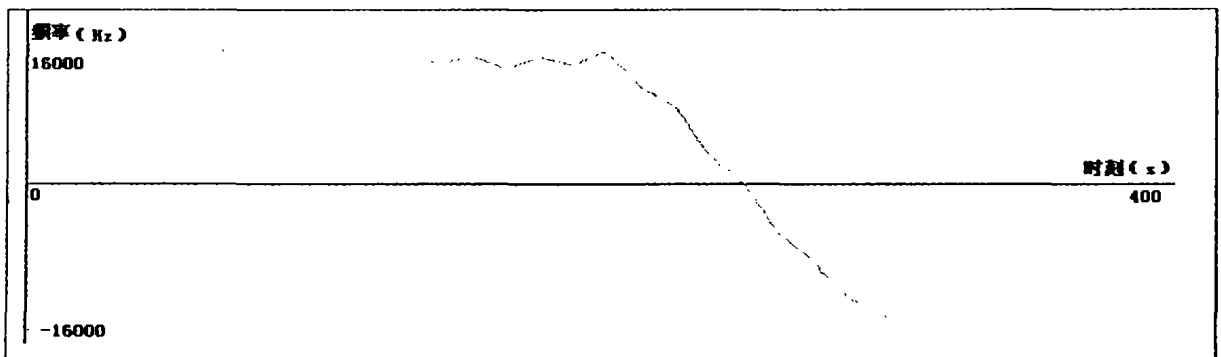


图10 回波频率值的典型变化曲线

参考文献

- 1 Fujimoto R M. Parallel Discrete Event Simulation. Communication of the ACM,1990,33(10)
- 2 Jefferson D R. Virtual Time. ACM Transactions on Programming Languages and Systems,1985, 7(3)
- 3 Wieland F,et al. Distributed Combat Simulation and Time Warp: The Model and Its Performance. In: Proc. of the SCS Multiconf. on Distributed Simulation 21, 1989. 14~20
- 4 Hontalas P,et al. Performance of the Colliding Pucks Simulation on the Time Warp Operating System. In: Proc. of the SCS Multiconf. on Distributed Simulation 21, 1989. 3~7
- 5 Presley M,et al. Benchmarking the Time Warp Operating System with a Computer Network Simulation. In: Proc. of the SCS Multiconf. on Distributed Simulation 21, 1989. 8~13
- 6 Bryant R E. Simulation of Packet Communications Architecture Computer Systems: [MIT-LCS-TR-188]. Massachusetts Institute of Technology, 1977
- 7 Cai W,Turner S J. An Algorithm for Distributed Discrete-Event Simulation -the "Carrier Null Message" Approach. In: Proc. of the SCS Multiconf. on Distributed Simulation 22,1990. 3~8
- 8 Chandy K M,Misra J. Distributed Simulation: A Case Study in Design and Verification of Distributed Programs. IEEE Trans. On Software, 1997,SE-5(5):440~452
- 9 Bellenot S. Global Virtual Time Algorithms. In: Proc. of the Multiconf. on Distributed Simulation,1990,21(1):122~127
- 10 Nicol D M. Parallel Discrete-Event Simulation of FCFS stochastic queueing networks. SIGPLAN, 1988(23):124~137
- 11 Tsai J J P, Yang S J, Chang Y-H. Timing Constraint Petri Nets and Their Application to Schedulability Analysis of Real-Time System Specifications. IEEE Transactions on software Engineering, 1995,21(1)
- 12 Avrunin G S,et al. Automated Derivation of Time Bounds in Uniprocessor Concurrent Systems. IEEE Transactions on Software Engineering, 1994,20(9)
- 13 Ferscha A. Adaptive Time Warp Simulation of Timed Petri Nets. IEEE Transactions on Software Engineering,1999,25(2)
- 14 Beraldi R,Nigro L. Distributed Simulation of Timed Petri Nets. IEEE Concurrency,1999,7(4)