doi: 10.3969/j.issn.1006-1576.2010.05.023

基于线程消息映射的虚拟仪表设计方法

徐君明，李国林，范绍里
（海军航空工程学院 7 系，山东 烟台 264001）

摘要：针对 Win32 控制台下单线程虚拟仪表设计与仿真系统融合存在的问题，提出了一种基于线程消息映射的虚拟仪表设计方法。该方法在主线程里运行主控程序，在每个辅线程里运行各虚拟仪表（含虚拟面板），通过线程消息映射机制实现仿真系统中主线程与各虚拟面板辅线程的控制与协同。实践表明，该方法可以大幅提升仿真系统的模块化设计效率、仿真的互操作性和模型的可复用性。

关键词：线程消息映射；虚拟仪表；线程间通讯；虚拟面板集成
中图分类号：TP391.9 **文献标识码**：A

Design Method of Virtual Instrument Based on Thread-Message Mapping

XU Jun-ming, LI Guo-lin, FAN Shao-li
(No. 7 Department, Naval Aeronautical & Astronautical University, Yantai 264001, China)

Abstract: Aiming at the problem during the fusion process of virtual instrument with simulation system in single thread under Win32 console, put forward a virtual instrument design method based on thread-message mapping mechanism. In this method, the main control program is running under primary thread, and each virtual instrument (including virtual panel) is running under every associate thread. By the thread-message mapping mechanism, the method implements the fusion and cooperation of primary thread with each associate thread running virtual instrument. The practice shows that the method can greatly promote the modularizing efficiency, the interoperability of simulation and the reusability of model in simulation system.

Keywords: Thread-message mapping; Virtual instrument; Inter-thread communication; Virtual panel integration

0 引言

GL Studio 软件是一个与平台无关的高效原型开发工具，可以快速地构建虚拟仪表和虚拟面板的仿真模型，生成 Win32 控制台的可执行文件或动态链接库文件[1]。因其可以生成所见即所得的实时交互的 2D/3D 仿真模型，且所生成的 C++代码与 Open GL 图形库完全兼容，该软件在虚拟仪器仪表（含虚拟面板，以下类同）开发方面的优势更为明显。但在某些建模与仿真领域，如弹道导弹、大型运输机等，单线程环境下的虚拟面板不足以完成相关的仿真和模拟训练任务[2]，而需要与处于同一进程地址空间的主控软件融合，协同运行以完成各项控制和仿真任务，这就涉及多线程环境下虚拟面板的设计技术及线程消息映射的问题。

1 虚拟仪表的一般设计

GL Studio 软件主要包括图形设计器、代码生成器等 GUI 工具。借助 Visual C++编译环境，设计人员无需高深编程知识就可以把 GL Studio 生成的 C++代码编译链接成控制台下的可执行文件。

虚拟仪表设计主要包括虚拟面板（仪表）规划、纹理设计、工程创建、虚拟仪表设计、代码生成和 Visual C++编译链接和发布虚拟面板等步骤，如图 1。该过程的设计结果是继承自 GL Studio 类库 disti::glsDisplayFrame 类或 disti:: ComponentBase 类虚拟面板类（*.cpp 和*.h），其中前者适用于发布成 Win32 可执行文件，后者用于发布成动态链接库。

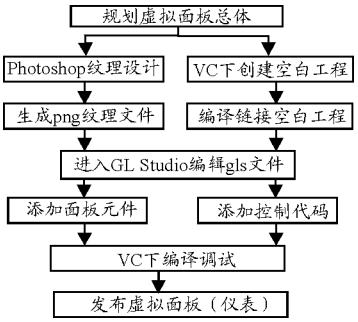

图 1 单线程虚拟仪表设计流程

单线程下虚拟仪表执行流程主要包括 GLS 库初始化、创建虚拟面板类对象、进入仿真循环等过程，在仿真主循环里逐帧绘制虚拟面板及其上的各种仿真元素，如电压表、电流表、按键和开关等。

```
int WINAPI WinMain(…)
{
…
```

收稿日期：2009-12-26；修回日期：2010-01-20
基金项目：总装备部十一五预研资助项目（513040301）；海军航空工程学院青年科研基金资助项目（200801C3）
作者简介：徐君明（1978-），山东人，男，讲师，博士研究生，从事弹道导弹指挥与控制系统工程研究。

```
glsCommandLine::Instance()-
>ReadCommandLine(argc,argv);//读取命令行参数
InitializeLibrary();        //初始化库
simplePanel = new SimplePanelClass("Simple
Panel",800,600);          //创建面板对象
simplePanel->CreateObjects();          //生成面板元素
simplePanel->SetRedraw();
Timer frameTimer;              //定义定时器
do{
    simplePanel->RecordCalculateStart();
simplePanel-
>Calculate(frameTimer.ElapsedSecondsDouble());
simplePanel->RecordCalculateEnd();
}while(simplePanel->FrameAnimate());       //仿真主循环
delete simplePanel;
…
}
```

单线程模式下的虚拟仪表一般设计对于诸如电源、万用表、示波器之类的小型仿真系统是够用的[7]。但当仿真实体涉及主控软件、仪器仪表操控台等多种类型的用户接口时，就无法完成预定的仿真任务。如果再考虑到仿真系统中网络通讯和数据库方面的因素，无论是从仿真软件设计效率上，还是从仿真系统运行开销上都很难接受。

2 多线程下的虚拟仪表设计

在涉及诸多虚拟面板或虚拟仪表的仿真系统设计中，可将多种类型用户接口、主控软件、网络通讯和数据库等多种仿真资源采用多线程机制来实现，如图 2。

图 2 基于多线程的仿真系统

采用这种设计结构，可在兼顾 GL Studio 虚拟面板和虚拟仪表设计性能的前提下，通过线程间的消息映射实现虚拟面板辅线程与其他仿真子线程的无缝集成和实时通讯。

进程主要包括一个内核对象和一个进程地址空间 2 项内容，其中，前者用于提供操作系统管理当前进程所必须的一些信息，后者主要包括可执行文件、动态链接库和数据等；线程是进程上下文中的执行序列，具体负责进程地址空间代码的执行，一个进程至少包括一个线程[8]，即主线程。

在系统设计中，创建一个进程需要包括内存地址空间、CPU 时间片等在内的庞大资源，而创建一个线程的系统开销要比创建一个进程小得多，因此，在单个仿真系统设计中应优先选择多线程设计思路。当然，这只是一般原则，具体实现方案要视仿真系统实际需求而定。在图 2 中，主线程的主控软件采用 Visual C++的 MFC 类库设计，其它各用户接口分系统均采用 GL Studio 软件设计成虚拟面板。

3 虚拟仪表中线程消息映射的实现

在多线程虚拟仪表设计中，基于 Windows 窗体的主控软件运行在仿真进程的主线程中，每个虚拟面板作为一个辅线程来运行。其设计过程主要包括单线程虚拟面板创建、测控软件设计及编译环境配置、多线程设计与集成等几个过程：

3.1 设计虚拟面板

遵循与实际装备外观和功能一致的要求设计各操控台虚拟面板，并将虚拟面板发布成继承自 disti::glsDisplayFrame 类的操控台虚拟面板类，并对虚拟面板的按键响应、电表指示等功能进行反复测试直至满足仿真需求。

3.2 编译环境配置

主线程中 Windows 测控软件的设计与基于 MFC 的一般程序设计无异，按照实际装备或仿真需求设计测控软件即可。之后，在当前 MFC 工程中配置 GL Studio 编译环境，主要包括：

1) 添加 GL Studio 编译所需头文件

在“Project Settings”的“C/C++”选项卡下的“Category | Preprocessor”下的“Additional include direcotries”里添加：

“$(glstudio)\plugins\include,$(glstudio)\include”；

2) 添加 GL Studio 编译所需库文件

在“Project Settings”的“Link”选项卡下的“Category | Input”下的“Additional library direcotries”里添加：

“$(glstudio)\plugins\lib,$(glstudio)\lib”；

3) 在当前工程中插入虚拟面板类

这个过程实际上就是把设计调试好的虚拟面板类的*.cpp 和*.h 文件添加到当前 MFC 工程中。注意，如果所设计的面板类中还内嵌其他*.gls 生成的面板类或动态链接库文件*.dll，要将其头文件或*.dll 文件一并添加到当前工程目录，否则编译后链接或运行时会报错。

3.3 线程消息定义和映射

为解决主线程和辅线程通讯的一致性问题，建立一个单独头文件定义线程消息如下：

```
…
    #define TM_ROLL WM_APP+100
    #define TM_PITCH WM_APP+101
    #define TM_GLS_START WM_APP+102
…
```

要在主线程头文件和虚拟面板类的头文件中引入这个线程消息定义头文件。

借助 MFC 的 ON_THREAD_MESSAGE 宏定义消息响应代码，主要包括：

1) 在主线程 CWinApp 的*.h 文件里声明消息响应函数：

```
    …
    afx_msg void
OnThreadMsgPitch(WPARAM,LPARAM);
    DECLARE_MESSAGE_MAP()
    …
```

2) 在主线程 CWinApp 的*.cpp 文件里设定线程消息映射函数：

```
BEGIN_MESSAGE_MAP(CIntegretApp, CWinApp)
    …
ON_THREAD_MESSAGE(TM_PITCH,OnThreadMsgPit
ch)
      …
END_MESSAGE_MAP()
```

3) 在主线程 CWinApp 的*.cpp 文件里定义线程启动函数和消息响应函数：

虚拟面板线程的启动函数定义为：

```
BOOL CIntegretApp::InitInstance()
{
    …
    AfxBeginThread( RunGlsPanel, 0 );
    …
}
```

添加线程消息响应代码为：

```
void CIntegretApp::OnThreadMsgPitch(WPARAM
wparam,LPARAM lparam)
{
    CString str;
    str.Format("%d",lparam);
    m_pDlg->m_editPitch.SetWindowText(str);
}
```

3.4　虚拟面板线程的启动与控制

主要包括虚拟面板辅线程函数的定义和启动 2 个过程：

1) 虚拟面板线程函数

虚拟面板线程函数的作用类似于单线程环境下虚拟面板主函数，用于在辅线程中启动虚拟面板：

```
UINT RunGlsPanel(LPVOID lpParam)
{
…
//以下同单线程下虚拟面板中 WinMain 函数体内的代码
return 0;
}
```

2) 线程消息控制

在虚拟面板类所有可能向主线程发送消息的回调函数处添加消息发送代码。这里主要用到 Win32 API 的 PostThreadMessage(DWORD，UINT，WPARAM，LPARAM)函数，示例代码如下：

```
…
if ( ObjectEventIs(ev, "DetentVal") )
{
_newPitch = (pitchGlsKnob->PositionVal()*0.9-45);
if ( !_testing )
{
::PostThreadMessage(theApp.m_nThreadID,TM_PITCH,
0,(LPARAM)_newPitch);
}
rval = true;
}
```

3.5　线程映射运行效果

基于线程消息映射的仿真系统运行效果如图 3。主线程首先启动辅线程的虚拟面板 ADI，在虚拟面板上启动开关，虚拟面板 ADI 的实时姿态值便通过线程消息映射进行通讯；与此同时，主线程可以在测控终端上对辅线程中虚拟面板 ADI 进行暂停、重启等操作。

图 3　运行效果

4　结束语

在系统建模与仿真中，采用多线程及线程消息映射机制实现虚拟仪表（或虚拟面板）与测控软件乃至整个仿真系统的通讯，对于提高仿真系统的分层设计效率、仿真的互操作性和模块的可复用性具有强有力的支撑作用。工程实践表明，只要设计好多个仿真线程的消息映射标准，则各分系统的具体设计可以在最大程度上保持相对独立性，从而提高整个仿真系统的集成设计效率。

（下转第 80 页）

算法实现简单，具有很好的工程应用前景。

该关联算法的关联正确率还有待于进一步提高，下一步的工作重点是更好地实现关联的正确率。反馈机制也是多传感器信息融合研究领域的一个重要方向，如何在多传感器航迹融合中引入反馈机制也是下一步迫切需要解决的问题。

参考文献：

[1] 韩红, 刘允才, 韩崇昭, 等. 多传感器融合多目标跟踪中序贯航迹关联算法[J]. 信号处理, 2004, 20(1): 30-34.

[2] 何子述, 夏威. 现代数字信号处理及应用[M]. 北京: 清华大学出版社, 2009.

[3] 杨露菁, 耿伯英 译. (美)霍尔(Hall,D.L.) 等 编著. 多传感器数据融合手册[M]. 北京: 电子工业出版社, 2008.

[4] 何友, 修建娟, 张晶炜, 等. 雷达数据处理及应用[M]. 北京: 电子工业出版社, 2006.

[5] Leigh A. Johnston and Vikram Krishnamurthy. An Improvement to the Interacting Multiple Model (IMM) Algorithm[J]. IEEE transactions on signal processing, 2001, 49(12): 2909-2923.

[6] 郭治. 现代火控理论[M]. 北京: 国防工业出版社, 1996.

[7] 乔向东, 李涛. 多传感器航迹融合综述[J]. 系统工程与电子技术, 2009, 31(2): 245-250.

****

（上接第 73 页）

参考文献：

[1] DiSTI. GL Studio Version 3.0 User's Guide, U.S.A http://www.disti.com/Products/glstudio, 2008.

[2] 汪乐舟, 张合新. 某型导弹模拟训练仿真系统的研究[J]. 兵工自动化, 2009, 28(2):10-12.

[3] 樊世友, 邸彦强, 朱元昌. GL Studio 软件在视景仿真建模中的应用[J]. 计算机工程, 2002, 28(3): 260-261.

[4] 高颖, 邵亚楠, 等. GL Studio 在飞行座舱模拟器中的仿真研究[J]. 弹箭与制导学报, 2008, 28(1): 257-260.

[5] 李哲煜, 张峋, 等. 大型运输机综合训练器虚拟仪表系统的研究[J]. 计算机仿真, 2007, 24(7): 247-250.

[6] 朱渊超, 车建国, 等. 基于 GL Studio 的某型雷达面板仿真[J]. 电脑开发与应用, 2006, 19(5): 21-25.

[7] 李栩冰, 郭喜庆, 冯祺. 一种基于 Lab Windows/CVI 的 GL Studio 插件调用方法[J]. 系统仿真技术, 2008, 4(1): 45-46.

[8] Charles Petzold. Programming Windows(Fifth Edition)[M]. Seattle: Microsoft Press, 2009:1198-1200.

****

（上接第 76 页）

6) 低功率高能量效率。低截获的一个核心思想就是进行严格的能量控制，因此发射信号的能量效率便是考核其低截获性能的一个重要标准。而根据文献发射信号能量效率可以定义为待测脉冲信号辐射能量与以待测信号频带为中心的单频信号的比值。不同的信号具有不同的发射效率，发射效率高的波形有利于信号的检测。

图 1 描述了多种主动声纳能量效率对比，表 1 给出了几种信号的能量效率。

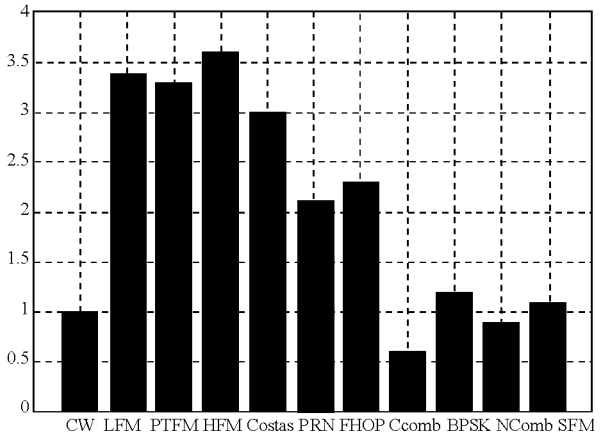

图 1 主动声纳信号能量效率对比

表 1 部分主动声纳波形能量效率

波形	能量效率	波形	能量效率
CW	中	Costas	中
LFM	高	PRN	中
HFM	高	New COMB	中
SFM	高	PTFM	高
FHOP	中	OC-PTFM	高
COX COMB	低	Costas-PTFM	高
BPSK	中	MCPC	高

3 结束语

分析了影响低截获主动声纳信号的诸因素，并从低频、大时宽带宽积、复合频（码）制、随机或非线性体制、时频捷变、低功率高能量效率等 6 个方面对低截获声纳的信号特征进行了剖析，对于今后选择 LPI 声纳的信号有一定指导作用。

参考文献：

[1] Stove, Hume, Baker. Low probability of intercept radar strategies[J]. IEEE Proc. Radar Sonar Navig., 2004, 151(5): 249-260.

[2] 张锡熊. 低截获概率（LPI）雷达的发展[J]. 现代雷达, 2003, 25(12): 1-4.

[3] 周胜, 林春生. 水雷主动声引信低截获特性分析[J]. 海军工程大学学报, 2009, 21(1): 91-95.