

# 大型运输机模拟器仪表系统<sup>①</sup>

李哲煜 刘国庆 张维军 (空军航空大学 吉林 长春 130022)

**摘要:** 目前模拟器仪表系统主要是实表系统, 存在驱动复杂、价格昂贵等缺点。结合大型运输机综合训练器的研制, 提出并采用了实物仪表与虚拟仪表相结合的飞机仪表系统, 给出了实物仪表和虚拟仪表相结合的耦合仪表系统的实现方法, 为飞机模拟器的仪表仿真提出了新的解决方案。

**关键词:** 模拟器; 真实仪表; 虚拟仪表

## Instrument System of a Large-Scale Transporter Simulator

LI Zhe-Yu, LIU Guo-Qing, ZHANG Wei-Jun

(Aviation University of Air-Force, Changchun 130022, China)

**Abstract:** The Instrument system of simulator is mainly real instrument system at present. It has shortcomings like having complex driving and high price. According to the development of a large-scale transporter simulator, an integrated plane instrument system of real instrument and virtual instrument is given and the developing methods of integration instrument system is presented. It provides new solution for instrument simulation of flight simulator.

**Keywords:** simulator; real instrument; virtual instrument

### 1 引言

计算机技术、图形图像技术和网络技术不断进步, 促进了模拟器技术的高速发展。模拟训练具有节能、安全、不受场地与气象条件限制、训练效率高、培训时间短、训练费用低等特点, 因此在军事训练中得到了广泛的应用<sup>[1]</sup>。目前在模拟器研制过程中, 仪表显示都采用实表系统, 即通过采用真实飞机的仪表, 通过相应的驱动电路来模拟显示飞机状态, 但这种方案的缺点是造价昂贵, 尤其是对于大型运输机而言。根据大型运输机仪表数量大, 仪表价格昂贵的特点, 本文介绍了一种在大型运输机半实物训练器的研制过程中采用虚拟仪表和实物仪表相耦合的仪表系统进行仪表仿真的方法。

案, 其体系结构如图 1 所示。

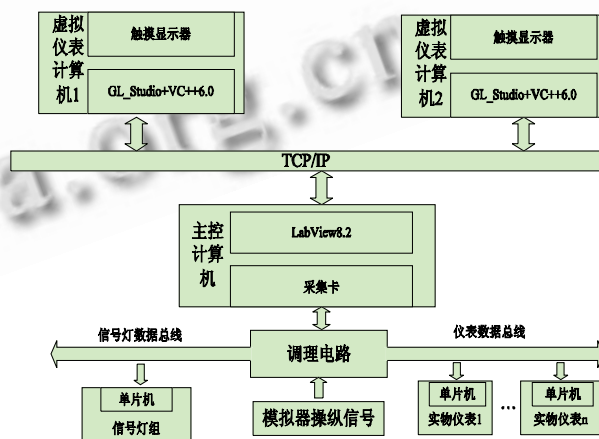


图 1 系统体系结构

### 2 大型运输机综合训练器的体系结构

大型运输机具有大量的控制面板, 而且在面板中装有大量的显示仪表和控制按钮、开关, 为了减少开发的费用和时间, 本模拟器采用半虚拟半实物仿真方

整个体系采用分布式结构, 虚拟仪表计算机主要用来显示利用 VC++6.0+GL\_Studio 开发的虚拟仪表, 为了实现与虚拟仪表进行人机交互, 虚拟仪表计算机采用了具有触摸功能的显示器。主控计算机是整

① 收稿时间: 2009-12-12;收到修改稿时间:2010-02-06

个系统的控制中心，主要接收来自外部电路和虚拟仪表计算机的控制信号，并根据这些信号进行飞机状态解算，然后将解算后的飞机状态信号传输给实物仪表盘单片机、信号灯控制单片机和虚拟仪表计算机，实现仪表和信号灯的显示。

模拟器工作时，模拟器上的各种操纵信号(如推油门、打开电门等)通过调理电路进行信号调理后，通过板卡采集到主控计算机，同时，虚拟仪表计算机上的触摸控制信号也通过网络实时的传递给主控计算机，主控计算机根据这些控制信号进行当前状态的解算，一方面将解算后的飞机的状态通过仪表数据总线和信号灯数据总线传输给仪表驱动单片机和信号灯驱动单片机，实现实物仪表的指示和信号灯的显示。另一方面，通过网络将飞机的仪表状态参数传送给虚拟仪表计算机，实现虚拟仪表的指示。

### 3 虚拟仪表的开发

#### 3.1 虚拟仪表的开发路线

对于模拟器的虚拟仪表的建模，我们采用的是世界上最流行的虚拟仪表建模软件 **GL Studio** 软件，它是基于 **OpenGL** 的三维图形仪表软件，具有开发效率高，成本等特点<sup>[2]</sup>。其开发路线如图 2 所示：

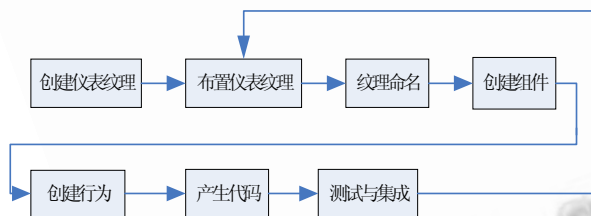


图 2 虚拟仪表开发路线

第一步：创建仪表纹理，即采用图像处理软件制作出所需纹理的图片，如表盘纹理和表针纹理等；

第二步：根据仪表的显示要求，在 **GL Studio** 工作区中创建多边形并贴上纹理；

第三步：为了便于管理和编程，为图形起一个有意义的名字；

第四步：为了便于仪表重复使用，将其创建成仪表组件；

第五步：为虚拟仪表创建行为，来动态的显示指示值，或响应鼠标、键盘的消息；

第六步：将编写的行为代码和图形数据转化为

C++类代码；

第七步：在 **VC++** 平台上进行编译和测试，察看显示效果，如果不满意，回到第二步。

#### 3.2 典型虚拟仪表的开发

座舱仪表主要包括柱形仪表、盘式仪表和矩形仪表。

##### 3.2.1 盘式仪表的开发

盘式仪表是指带有细分刻度和数字标记的圆盘状仪表，下面以高度表为例，论述盘式仪表的建模过程。

如图 3，高度表主要由表针和数字滚轮和调整旋钮组成，通过表针和滚轮联合指示来表示飞机的高度，滚轮表示高度的前几位，指针表示高度的后几位。旋钮调整飞机起飞前的机场的气压值。

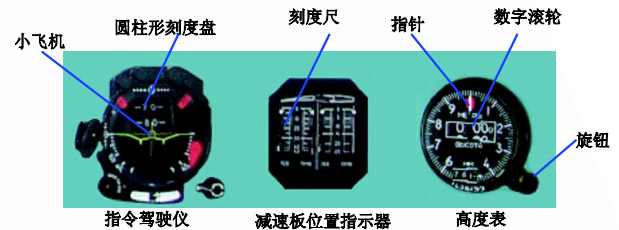


图 3 典型的飞机仪表盘

指针的旋转是比较容易制作的，首先在 **GL Studio** 软件当中定义高度表类的一个成员函数，这个函数的所带的参数为所要显示的值，然后在函数体中添加代码，将数值转化为指针的角度值，通过 **API** 函数 **DynamicRotate** 将指针指向所显示的位置。**DynamicRotate** 的函数原型为<sup>[3]</sup>：

```
GLS_EXPORT void DynamicRotate ( float angle, int axis)
```

在函数体中添加代码，将数值转化为指针的角度值，转换公式为： $\theta = \frac{D}{R} \alpha$

其中  $\theta$  为指针实际旋转角度， $\alpha$  为指针旋转的角度范围， $R$  为刻度范围， $D$  为高度值。

数字滚轮是广泛应用在飞行仪表上的数字显示设备，它通过数字条的移动来实现数字的滚动效果，它的制作是通过类来实现的，首先，在定义一个类变量，类类型为 **OdometerOperatorClass**，然后在对所创建的类变量进行初始化，最后在 **InputDevice** 对象的回调函数或者在类循环函数中添加如下的动态数字滚轮显示函数<sup>[3]</sup>。

Altitude\_display->Value(\_altitude);

其中，Altitude\_display 为 OdometerOperatorClass 类类变量，\_altitude 为所显示的数值。以下是 OdometerOperatorClass 类原型函数：

```
OdometerOperatorClass(Group *group, int
magnitude, int number_of_wheels, float
texture_scale, float min_val, float max_val, float
rollover,
```

```
OdometerWheelEnum wheel0_type,
```

...

```
OdometerWheelEnum wheel10_type)
```

第一个参数表示数字滚轮组的名称，第二个参数表示滚轮组所表示的数值的最高位数，第三个参数表示滚轮的数目，第四个参数表示数字条纹理刻度，它表示数字轮变化一个单位，数字条材质移动的距离，如图 4，它的计算公式为：

$$\xi = \frac{(h_2 - h_1) / l}{10}$$

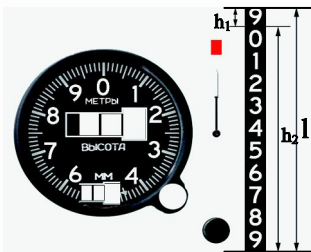


图 4 高度表材质

其中， $h_1$  表示数字条 0 的起始位置， $h_2$  表示数字条中 9 的终止位置。 $l$  表示数字条材质的长度，它们都用像素大小来表示。第五个参数表示数字滚轮组所能表示的最小值，第六个参数表示数字滚轮组所表示的最大值，第七、八...个参数表示轮的滚动类型，参数的个数要根据滚轮的数目来决定。

对于旋钮的开发，首先在 GLstudio 建模软件内将旋钮的材质转化为 Knob 输入设备，然后在其旋钮的回调函数 Knob-ObjectCallback(GLsKnob \*self, DisplayEvent \*ev) 添加响应的代码即可。

```
Int Knob-ObjectCallback(GLsKnob*self,
DisplayEvent *ev)
```

```
{ bool rval=false;
```

```
if(ObjectEvents(ev,"PositionVal"))
```

```
{ //响应代码
```

```
rval=true;
```

```
}
```

```
return rval;
```

```
}
```

将上面用 GL Studio 开发的高度表模型在 VC++6.0 平台上进行编译，即可实现虚拟高度表的动态显示。

### 3.3.2 柱形仪表开发

形仪表是指在圆柱体的外表面具有数字标志和刻度标尺的一种计量仪表。下面以指令驾驶仪为例，论述具体建模过程。

如图 3 所示：它通过水平指针一小飞机的转动来表示飞机的横滚值，通过带刻度的圆柱的转动来实现飞机俯仰值。对于横滚值的指示是通过小飞机的转动来实现的，所以是比较容易实现的，对于俯仰值的指示是通过带不均匀刻度的圆柱或球的转动来实现，它的建模是比较复杂的。它的建模过程如下：

(1) 利用图形处理工具准备好刻度材质。

(2) 利用 GL Studio3.0 的三维建模工具，创建一个圆柱，软后将刻度材质赋予此圆柱上，使刻度材质包裹整个圆柱表面，通过圆柱的转动来表示俯仰值。

(3) 由于在贴图过程中，不能保证包裹的刻度能与真实的仪表刻度相同，也就是当圆柱体旋转  $\theta$  角，可读盘所指示的位置不是  $\theta$  角。同时由于刻度盘的刻度分布不是均匀的，即 0~5 度 2.5 度一个格，5~40 度 5 度一个格，40~80 为 10 度一个格，所以为了正确的显示飞机的俯仰值，本文采取了如下的分段函数的数学模型：

$$\theta = \begin{cases} \frac{\theta_1}{5^0} \times \alpha & 0 \sim 5^0 \\ \frac{\theta_2 - \theta_1}{35^0} \times (\alpha - 5^0) + 5^0 & 5^0 \sim 40^0 \\ \frac{\theta_3 - \theta_1}{40^0} \times (\alpha - 40^0) + 40^0 & 5^0 \sim 40^0 \end{cases}$$

其中， $\theta_1$ 、 $\theta_2$ 、 $\theta_3$  分别表示刻度盘分别表示 5°、40°、80° 时，圆柱体旋转的角度。表示刻度指示角度时圆柱体应旋转的角度。

(4) 在 GLStudio 代码区中创建地平仪类的一个

成员函数，在此成员函数中，根据上面的公式，编写代码实现飞机俯仰值的正确指示。

### 3.3.3 矩形仪表开发

矩形仪表是指一种带有数字标志和水平刻度标尺的垂直型计量仪表。飞机上的矩形仪表主要有：升降舵、平尾位置指示器等。下面以图 3 减速板位置指示器为例，论述开发过程。具体开发步骤为：

- ①建模过程与上述仪表建模方法相同；
- ②控制滑块移动。在初始化位置调用函数

GetLocation( )获得滑块初始位置，然后在代码添加窗口添加如下代码：

```
void JSBZSQGClass::Lsjd_angle (const float&
value)
{ Vertex v;
  v.x=Lsjd_init.x;
  v.z=Lsjd_init.z;
  v.y=(Lsjd_angle)*(68.0/25.0)+Lsjd_init.y;
//滑块沿 Y 轴移动
  up_slide1->Location(v);
}
```

- ③生成代码，在 VC 环境下进行调试。

### 3.3.4 虚拟仪表仿真效果

图 5 是上述方法开发的某型运输机的虚拟仪表仿真效果图。



图 5 虚拟仪表仿真效果图

## 4 实物仪表的驱动

对于模拟器中的实物仪表，采用了 LabView 测控软件，它是一种用图标代替文本创建应用程序的图形化编程语言<sup>[4]</sup>。通过总线技术，将控制信息发送给各个仪表驱动单片机，单片机接收到控制信息后，驱动响应的仪表指针转动，从而实现实物仪表的指针的驱动指示，如图 6 所示：

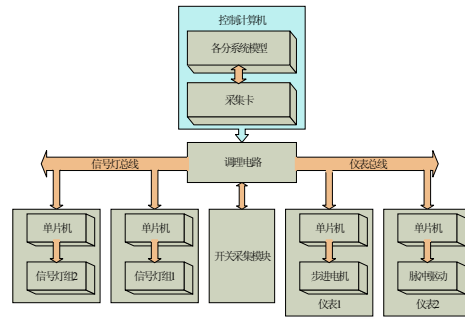


图 6 实物仪表驱动原理图

### 4.1 仪表驱动主控软件的设计

对于实物仪表驱动总线，采用了 10 位的数字总线，其中高 2 位是地址或数据的标志位，后 8 位为地址或数据信息。因此，应用 LabView 开发的仪表驱动主控软件的流程图如图 7 所示：

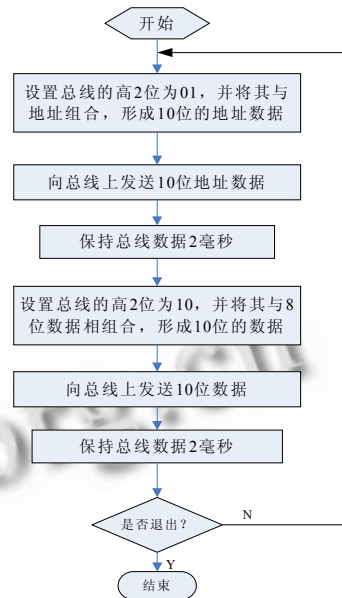


图 7 仪表驱动主控程序

### 4.2 单片机控制软件的设计

实物仪表控制采用总线方式，由控制计算机通过采集卡发出控制指令，然后通过调理电路，输入总线，在总线上的各个单片机接受到信号后，进行地址匹配，如果地址正确，解算数值，驱动仪表指示。

在控制实物仪表的单片机中，均建立了每个仪表的数学模型。但单片机接受到仪表数值后，能够自动进行模型解算，根据仪表的特性，进行驱动。

实物仪表采用了步进电机驱动和脉宽驱动两种方式。对于指针变化速率慢或者转动角度大的仪表，采

用步进电机驱动；对于指针变化速率快的仪表，采用脉宽驱动方式。当单片机解算完数据，对于采用步进电机驱动的仪表，对步进电机发出步进数，驱动步进电机指示到相应位置。而对于采用脉宽控制的仪表，单片机发出占空比变化的 TTL 信号，通过线圈驱动指针偏转，它们的程序流程如图 8 所示：

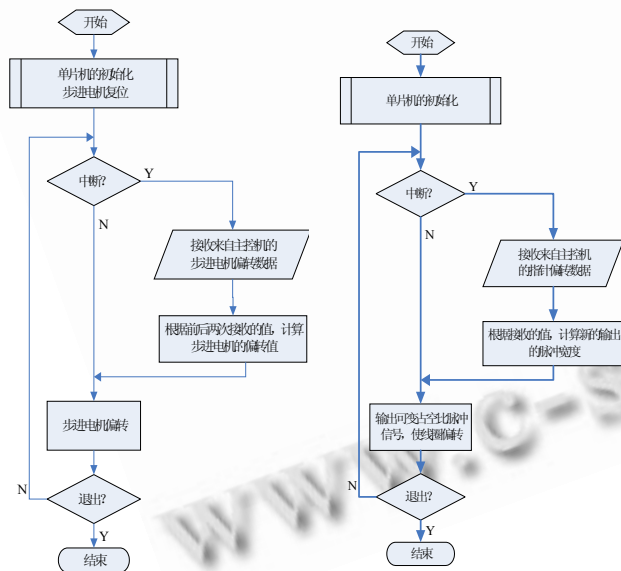


图 8 步进电机和线圈驱动流程图

## 5 结语

本文结合空军资助的项目《某型飞机综合训练模拟器》的研制，重点介绍了虚拟仪表和实物仪表的开发过程。并将此技术应用到的某型模拟器的研制，通过实验表明，利用虚拟仪表来代替部分真实仪表制作模拟器，实现虚拟仪表与实物仪表耦合仪表系统完全可以达到实表模拟器的仿真效果。不仅能够提高开发效率，而且开发费用也会大大的降低。随着大尺寸显示器技术和大尺寸触摸屏技术的发展，虚拟仪表和实物仪表相耦合的显示方案将会在模拟器中得到广泛的应用。

## 参考文献

- 1 康凤举.现代仿真技术与应用.北京:国防工业出版社, 2001.4-20.
- 2 樊世友.GL Studio 软件在视景仿真建模中的应用.计算机工程, 2002,28(3):260-261.
- 3 DISTI, Inc.G5studio Programming Guide(version 3.0). USA: DISTI, Inc,2002.
- 4 陈锡辉.LabView 8.02 程序设计从入门到精通.北京:清华大学出版社, 2007.
- 5 David J.Kruglinski. Visual c++6.0 技术内幕.北京:北京希望电子出版社, 2001.