

# 一种火控雷达终端显示器的仿真实现

雷超, 陈伟

(厦门大学, 福建 厦门 361005)

**摘要:** 雷达系统的仿真模拟由于成本低、研发周期短、效果好等优点在军队中开始得到广泛应用, 雷达系统的模拟训练已成为某些部队军事训练的一种方式。雷达终端显示器的仿真是雷达系统仿真中的一个重要部分, 在三维坐标系中对飞行目标航迹的产生进行了建模, 并利用 C++ Builder 6.0 开发工具, 采用多线程技术, 对某一火控雷达终端显示器进行仿真, 实现了多飞行目标情况下, 单个目标捕获跟踪、杂波的模拟、雷达终端数据的实时录入等功能。本设计能很好应用于雷达课程教学和军队雷达手的操作训练中。

**关键词:** 火控雷达; PPI 显示器; 多线程; 雷达杂波; 坐标映射

中图分类号: TP391.9

文献标识码: A

文章编号: 1673-629X(2008)04-0195-04

## Realization of Simulation for a Fire Control Radar Terminal Display

LEI Chao, CHEN Wei

(Xiamen University, Xiamen 361005, China)

**Abstract:** Simulation of radar system has begun to be applied widely in army due to its low-cost, short period of development as well as good application results, and the training of simulation of radar system becomes a kind of military training in some units. Simulation of radar terminal display is a very important part of simulation of radar system. Constructs a model to produce routes of flying objects in three-dimensional coordinates. With C++ Builder 6.0 and multi-thread method, a simulation of a fire control radar terminal display is well realized, which includes the tracking of single object, simulation of radar clutter as well as real-time acquisition of data in radar terminals and so on. This design is well applied in radar textbook teaching and military training for radar operators.

**Key words:** fire control radar; PPI; multi-thread; radar clutter; coordinates conversion

### 0 引言

雷达作为现代化战争中的“电子眼”, 是影响现代化战争胜负的关键性因素。军用雷达中的火控雷达是一种精密跟踪雷达, 是控制火炮(或地空导弹)对空中目标进行瞄准攻击, 它的主要任务是实时测定飞行目标的坐标, 并迅速将射击数据传给火炮或地空导弹, 从而实现对方飞行目标的精确打击<sup>[1]</sup>。本设计通过利用 Borland 公司开发的 C++ Builder 6.0 开发工具, 采用多线程技术来实现某一火控雷达终端显示器的仿真模拟。

### 1 仿真界面构建与功能

本仿真软件主要界面由 PPI 显示区、目标跟踪数据显示区、雷达手操作区组成, 具体框架如图 1 所示。

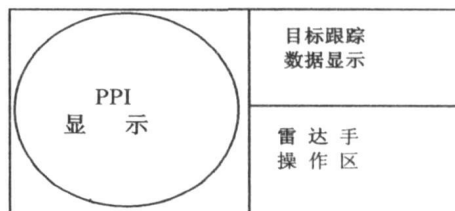


图 1 仿真界面框架图

PPI 显示区模拟雷达的 PPI 显示器, PPI 显示器是使用最广泛的雷达显示器, 它提供 360 度平面范围的目标分布情况, 通常也叫全景显示器或环视显示器, 简称 PPI 显示器或 P 显<sup>[2]</sup>。它主要显示目标的方位和距离(离雷达中心的斜距), 平面上的亮点表示目标的坐标位置, 其亮度强弱可以调节。

目标跟踪数据显示区显示当模拟目标被捕获跟踪后, 雷达提供的一系列关于被捕获跟踪的目标飞行参数, 包括目标的方位角、仰角、速度、距离、加速度等。

雷达手操作区主要模拟雷达启动操作、飞行目标的载入、目标飞行参数的初始设置和更改操作、杂波的产生和抑制操作、单一目标的捕获跟踪操作、目标实时

收稿日期: 2007-07-07

作者简介: 雷超(1983-), 男, 湖南常德人, 硕士研究生, 研究方向为检测技术与自动化装置; 陈伟, 副教授, 研究方向为检测技术与自动化装置。

数据载入等。

### 2 目标航迹的产生建模

目标航迹的产生是雷达终端显示器模拟的关键的一环,目标回波在显示器上何处显示出来取决于目标航迹,现实中的目标飞行航线是千变万化的,单一的直线或匀加速飞行规律无法逼真地模拟目标的飞行。有些仿真通过实时接受另一台计算机发送的飞行目标的飞行参数,根据接受到的飞行参数来确定飞行目标的位置和其他飞行状态,这种情况需要双机工作,一个负责发送数据,一个负责接受数据。还有的仿真将是相关数据存放在数据库中再通过定时访问数据库把数据调出来。本设计通过人机交互式的操作,设定飞行目标飞行参数初始值,然后通过改变飞行目标的加速度,从而改变飞行航迹,这种方法一方面能避免飞行目标按照初始的固定参数飞行,得到的是一个有规律的单调的飞行航迹;另一方面,由于是通过改变加速度(设置好加速度的范围)能够防止因飞行速度的大跃变导致航迹的突变,从而达到较好的模拟效果。目标各时刻的位置可以通过以下笛卡儿坐标系进行建模<sup>[3]</sup>:在这里假设目标离雷达中心站的斜距为  $S_0$ ,方位角为  $\theta_0$ ,高低角为  $\beta_0$ ,为方便计算,在这里假设目标的瞬时速度为  $V_0$ ,速度的方位角和高低角也分别为  $\theta_1$  和  $\beta_1$ ,假设此时的加速度为  $a_0$ ,方向角为  $\theta_1$ ,高低角为  $\beta_1$ 。如图 2 所示。

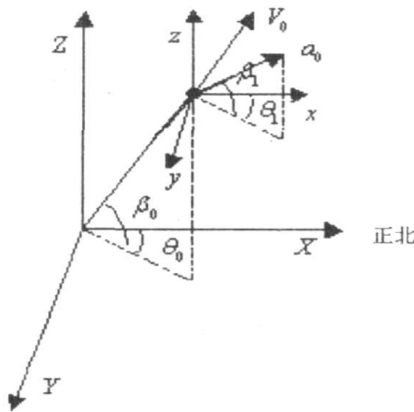


图 2 飞行目标三维航迹建模

设斜距、速度在  $X, Y, Z$  各个坐标轴的分量依次为  $s_{0x}, s_{0y}, s_{0z}, v_{0x}, v_{0y}, v_{0z}$ , 则由运动学公式可得其迭加算法:

$$v_{0x} = V_0 \cos\beta_0 \cos\theta_0 + a_0 \cos\beta_1 \cos\theta_1$$

.....

$$V_0 = \sqrt{v_{0x}^2 + v_{0y}^2 + v_{0z}^2}$$

$$s_{0x} = S_0 \cos\beta_0 \cos\theta_0 + v_{0x} + \frac{1}{2} a_0 \cos\beta_1 \cos\theta_1$$

$$S_0 = \sqrt{s_{0x}^2 + s_{0y}^2 + s_{0z}^2}$$

### 3 雷达杂波

雷达在工作过程中会受周围的自然地理环境(如高山、近处的高建筑物等)、天气气候(如雪、雨、云等)、飞禽(鸟类迁徙、昆虫的成群飞行等)以及敌方施放的金属丝干扰等外界因素的影响而接收到不需要的各种回波,这类回波称为雷达杂波<sup>[4]</sup>。这些产生杂波的实体大多是完全不动的或者运动缓慢的。根据多卜勒效应,即目标速度的不同引起回波信号频率产生的多卜勒频移不相等,就可以从频率上区分不同速度目标的回波,很多雷达中使用了各种滤波器,滤去固定杂波取出运动目标的回波,提高雷达的抗干扰的能力。为了比较逼真地模拟现实中的雷达工作情况,本仿真也简单地引入了两种运动状态实体所产生的杂波。一种是静止实体产生的杂波(位置不变),一种是运动实体产生的杂波(随时间位置发生变化),而且静止实体产生的杂波通过相应的操作可以被屏蔽掉。

### 4 仿真功能实现

本仿真采用多线程技术(共六个线程)来实时模拟八个飞行目标以及雷达杂波,由于 C++ Builder 6.0 提供的 Tthread 类是一个带有虚拟抽象对象方法的类,所以必须通过 C++ Builder 6.0 开发工具提供的 Tthread 类,产生派生类,再由派生类创建线程类的实例。本设计用一个线程专门用来绘图,并将它的优先级高于其它的线程,其它的线程分别用来计算飞行目标的位置和各种飞行参数,显示被捕获目标的位置、速度等各种参数并载入数据库。仿真中所有图像都在 PaintBox 组件中显示,图像的绘制主要通过调用 GDI 函数实现,并采用双缓技术消除画面抖动和闪烁现象<sup>[5]</sup>,雷达扫描线由画一组颜色不断变淡的半径来实现,杂波与目标的回波的显示采用“消隐技术”,即对同一回波用一组颜色明淡不同的图片或者绘制同一形状但颜色明淡不同的图形依次显示(回波亮度逐次变淡最后消失)来模拟现实中的回波。这样能达到较好的逼真效果。

#### 4.1 坐标映射

在绘制图形过程中,必须考虑两种坐标:设备坐标(又称物理坐标)和逻辑坐标,设备坐标指输出到设备上的坐标。设备坐标的原点在窗口的左上角, X 轴 Y 轴方向分别是向右和向下,以像素为单位。逻辑坐标指系统用做记录的坐标,在缺省的模式(MM\_TEXT)下,逻辑坐标和设备坐标的方向、单位和原点相同,由

于涉及到平面内 360 度范围内的距离、刻度、方向, 必须设置合理的坐标系(即坐标原点在显示窗口的中心,  $X$  轴方向向右,  $Y$  轴方向向上, 量程范围为 40000m)。完整的坐标映射包括以下三个步骤:

### ① 设置映射模式。

共有八种映射模式: MM\_TEXT, MM\_HIENGLISH, MM\_LOENGLISH, MM\_HIMETRIC, MM\_LOMETRIC, MM\_TWIPS, MM\_ANISOTROPIC 和 MM\_ISOTROPIC。

后面两种模式允许程序员自己设置视口和窗口范围, MM\_ANISOTROPIC 要求  $X$ 、 $Y$  轴范围必须相同, 而 MM\_ISOTROPIC 则不需要。

```
SetMapMode( Background -> Canvas -> Handle, MM_
ANISOTROPIC);
```

### ② 设置坐标轴的方向( $X$ 轴 $Y$ 轴方向)。

```
SetWindowExtEx( Background -> Canvas -> Handle, drawing
width, drawingheight, NULL);
```

```
SetViewportExtEx( Background -> Canvas -> Handle,
Background -> Width, - Background -> Height, NULL);
```

### ③ 设置逻辑坐标与设备坐标的原点。

```
SetWindowOrgEx( Background -> Canvas -> Handle, 0, 0,
NULL);
```

```
SetViewportOrgEx( Background -> Canvas -> Handle, Back
ground -> Width/2, Background -> Height/2, NULL);
```

## 4.2 线程应用

负责图像绘制的线程, 在绘制图像过程中, 采用双缓冲技术, 即先在内存中将图像画好, 再拷贝到 PaintBox1 组件中。

### ① 创建绘制图像的线程并完成对线程的初始化:

```
pThread2= new RadarScanThread( true);
pThread2 -> Priority= tpHigher; // 挂起方式创建雷达扫描线程
对象实例, 设置优先级在其构造函数中对其进行简单的初始化
FreeOnTerminate = true; // 线程终止时自动释放内存
```

② 判断是否扫描到目标并将扫描到的目标绘制在内存中:

```
if( Form1 -> x1+ 20- Form1 -> th2_ angle> = 0&& Form1 ->
x1+ 20- Form1 -> th2_ angle< = 4)
{
    Form1 -> object2_ flag0= 1; // 发现目标置标志为 1
    Form1 -> object2_ flag= 100;
    .....
}
else
    .....
```

```
Location= Rect( th_ k, th_ i, th_ k+ 600, th_ i+ 500); // 目标显示
位置
```

```
Form1 -> Background -> Canvas -> CopyMode= cmSrcCopy; //
```

设置拷贝模式

```
Form1 -> Background -> Canvas -> CopyRect( Location, Form1
-> panel -> Canvas, Form1 -> planeRect); // 将目标复制到内
存中
```

③ 用 Synchronize 方法将内存所有图像复制到 PaintBox 中:

```
Synchronize( all_ show);
```

Synchronize 是线程的同步方法, 只有通过 Synchronize 方法线程成员函数才能访问主线程中的 VCL 控件, 这个方法只有一个参数, 即指向线程成员函数类型指针, 并要求此成员函数必须无返回值也没有参数<sup>[6]</sup>。all\_show() 函数是此线程的成员函数, 功能是将内存中的图像拷贝到 PaintBox 控件中。

④ 其它线程产生和计算不同飞行目标的实时位置和有相关飞行参数, 并显示和载入追踪捕获到的目标的飞行信息。

```
th1_ z01= th1_ z01+ th1_ v01+ 0.5* th1_ a01; 目标在 X 轴方向
上的位移
```

.....

```
th1_ s0= sqrt( th1_ z01* th1_ z01+ th1_ x01* th1_ x01+ th1_ y01
* th1_ y01); 目标的距离
```

```
Form1 -> th3_ x1= th1_ s0* th1_ x01/sqrt( th1_ y01* th1_ y01
+ th1_ x01* th1_ x01);
```

```
Form1 -> th3_ y1= th1_ s0* th1_ y01/sqrt( th1_ y01* th1_ y01
+ th1_ x01* th1_ x01); 目标的方位
```

```
th1_ v01= th1_ v01+ th1_ a01; 速度在 X 轴方向上的迭加
```

```
..... 速度在 Y 轴方向上的迭加
```

.....

⑤ 一个线程专门用来对产生杂波的实体(静止和运动的实体)的位置进行初始化, 对于运动的实体还要对其速度、加速度进行初始化, 初始化通过调用 Random() 实现, 此线程还负责计算产生杂波的运动实体各个时刻的目标的位置和飞行参数。

## 4.3 目标的捕获跟踪

当启动捕获跟踪操作后, 鼠标光标在雷达显示区域变为“O”, 此时可以移动鼠标将目标“套住”进行捕获, 捕获成功后, 被捕获目标将被一个“□”套住, 并跟随目标一起移动, 与此同时在捕获目标的附近出“+”表示预测目标的未来位置, 目标跟踪数据显示区内将显示捕获目标的飞行参数和位置。

```
if( Form1 -> capture_ enabled= = 1) // 允许捕获
```

```
{
```

```
if( fabs( Form1 -> x_ capture - Form1 -> th1_ x1) <
800&& fabs( Form1 -> y_ capture - Form1 -> th1_ y1) < 800)
```

```
Form1 -> capture_ flag0= 1; // 成功捕获目标, 置捕获标志为
1
```

```
.....
```

### 5 仿真结果

图 3 是在引入杂波的环境下某一火控雷达显示器仿真, 图 4 为被捕获目标飞行参数的实时显示。

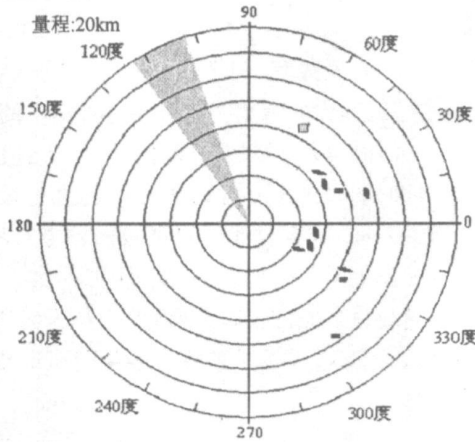


图 3 PPI 显示器仿真图

### 6 结束语

本设计通过交互式的操作来改变飞行目标的航行轨迹, 并采用多线程技术很好地实现了多目标的火控雷达终端显示器的仿真。此软件在雷达手模拟操作训练和雷达课程教学以及火控雷达仿真研究中有很好的

实用性。

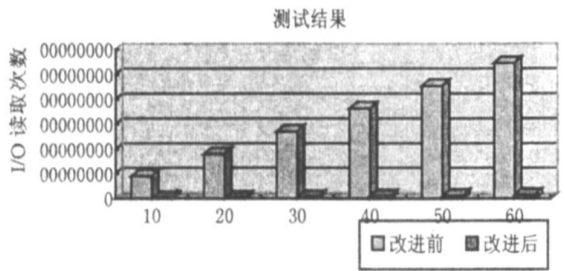


图 4 捕获目标跟踪数据显示

#### 参考文献:

- [1] 丁鹭飞, 耿富录. 雷达原理[M]. 西安: 西安电子科技大学出版社, 2003.
- [2] 刘 伟, 杨万海. 用 VB 实现 PPI 显示器的仿真[J]. 计算机仿真, 2004, 21(5): 191- 196.
- [3] 倪国旗, 李如年. 火控雷达的仿真目标建模及实现[J]. 火力与指挥控制, 2001, 26(3): 57- 59.
- [4] SkoInik M I. 雷达系统导论[M]. 左群生, 徐良国, 马 林, 王德纯, 等译. 北京: 电子工业出版社, 2001.
- [5] 张万里, 陈战林. C++ Builder 5. 0 高级开发技巧与范例[M]. 北京: 电子工业出版社, 2001.
- [6] Hughes C, Hughes T. C++ 面向对象多线程编程[M]. 周良忠译. 北京: 人民邮电出版社, 2002.

(上接第 194 页)

从表 1 可以看出, 改进后 I/O 读取次数大幅度下降, 约降低了 97%, 这样既保护了硬盘, 又提高了系统反映速度。

目 标 跟 踪 显 示	目标编号:	000111
	距 离:	10657
	速 度:	114. 93
	加 速 度:	0
	高低角:	43. 95
	水平角:	62. 23

图 7 并发呼叫量

### 3 结束语

文中实现的 IVR 系统所依赖的开发库 Openh323/PWlib 是跨平台的, 使得这个 IVR 具有良好的可移植性, 已经应用于福州市邮政 11185 呼叫中心系统中(Linux 环境下), 其运行状况良好。目前基于 VoIP 的 IVR 系统都是以 Voice XML 为基础的, 它们遵从 Voice XML 1. 0 标准, 文中实现的 IVR 系统同样遵从

这个规范, 但是系统中的 XML 解析器完全是笔者独立自主实现的, 并开发了相应的语音流程配置工具, 用户可以用这个图形化的工具自动生成语音流程配置文件(Voice XML 格式)。

#### 参考文献:

- [1] 张登银, 孙精科. VoIP 技术分析与系统设计[M]. 北京: 人民邮电出版社, 2003.
- [2] Collins D. VoIP 技术与应用[M]. 舒华英, 等译. 北京: 人民邮电出版社, 2003.
- [3] 周宽久, 王宏玲, 曲义飞. 交互式语音专家系统研究[J]. 哈尔滨工业大学学报, 2006(11): 1981- 1985.
- [4] 张 臻, 张仕军, 黄本雄. 可灵活扩展的 IVR 软件设计思想[J]. 计算机工程与设计, 2002(10): 47- 57.
- [5] ITU- T. Proposed Revision of Recommendation H. 323[S]. [s. l.]: [s. n.], 1997.
- [6] Ransome J F, Rittinghouse J W. Amsterdam VoIP security[M]. Boston: Elsevier Digital Press, 2005.
- [7] 张 荣, 武 波. SIP 协议的应用研究[J]. 计算机技术与发展, 2006, 16(6): 71- 73.