

代 号 10701

学 号 1010121048

分 类 TN955

密 级 公开

题 (中、英文) 目

可视化技术在雷达仿真系统中的应用

Visualization Technology in the Application

of the Radar Simulation System

作 者 姓 名 刘昕玮 指导教师姓名、职称 张林让 教授

学 科 门 类 工学 学科、专业 信号与信息处理

提交论文日期 二〇一三年三月

西安电子科技大学 学位论文独创性（或创新性）声明

秉承学校严谨的学风和优良的科学道德，本人声明所呈交的论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢中所罗列的内容以外，论文中不包含其他人已经发表或撰写过的研究成果；也不包含为获得西安电子科技大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中做了明确的说明并表示了谢意。

申请学位论文与资料若有不实之处，本人承担一切的法律责任。

本人签名：_____

日期_____

西安电子科技大学 关于论文使用授权的说明

本人完全了解西安电子科技大学有关保留和使用学位论文的规定，即：研究生在校攻读学位期间论文工作的知识产权单位属西安电子科技大学。学校有权保留送交论文的复印件，允许查阅和借阅论文；学校可以公布论文的全部或部分内容，可以允许采用影印、缩印或其它复制手段保存论文。同时本人保证，毕业后结合学位论文研究课题再撰写的文章一律署名为西安电子科技大学。

（保密的论文在解密后遵守此规定）

本学位论文属于保密，在____年解密后适用本授权书。

本人签名：_____

日期_____

导师签名：_____

日期_____

摘要

随着雷达仿真系统研究的不断发展研究，其规模不断扩大、功能持续增强，以往的雷达仿真软件存在使用复杂、不能直观、重用率低扩展难等缺陷，已不能很好的适应当前雷达仿真系统的发展要求。

可视化技术能够在三维图形世界中直接对具体目标进行操作，和计算机直接交流，赋予计算机一种仿真的，三维的并且具有实时交互的能力。

本文首先给出了三维可视化雷达仿真系统的研发方案，该方案采用 Windows 界面编程、三维数字球、3D 成像和组件技术等关键技术。然后在 C#语言环境下开发软件操作界面，并嵌入了三维数字球地图空间;在此基础上研发了雷达模型参数的设置组件、具有雷达设置、添加探测目标、目标运动趋势预判等功能的场景组件和协调控制的子组件。其间仿真参数运用 XML 参数表传递、使用内存共享传递仿真数据使用内存共享传递、用 UDP 技术报文。最后组成了一套完整的具有三维可视化特性的雷达仿真系统软件，并使用实例雷达仿真系统对特定几个场景仿真使用，验证该系统的正确性与稳定性。

关键字：雷达仿真，三维建模，可视化，数字球，OpenGL

Abstract

With the development of radar simulation system, the scale of its research is continually expanding and the function of is increasing strengthen, the past radar simulation software have some defects such as the use is complex, can't intuitive, and reuse rate is low. This situation has not been able to adapt well to the development of radar simulation system requirements.

The visualization technology in 3D graphics can directly operate on specific goals, and can communicate with computer directly. It gives the computer simulation, three-dimensional and real-time interaction ability.

This paper gives a design scheme of visualization of radar simulation system firstly, this design scheme use the some key technology such as graphical interface programming, 3D digital ball, 3D imaging and forming technology, has the characteristics of easy to use, intuitive simulation information, reusable, easy to expand. Then, use the visual form to develop the simulation software interface, and embedded in 3D digital ball map space; based on this research, this paper developed the radar parameter setting assembly which to set simulation parameters, and developed subassembly include scene simulation setting assembly and simulation control assembly which have the function of radar setting, add object detection, and moving tendency of object. Each component is developed independently and has the strong reusability. The development use the XML parameter table to transfer simulation parameters, use shared memory to transfer the simulation data, use UDP technology to report message, and finally constitute a set of integrated radar simulation system software which has the characteristics of visualization and open. This paper verifies the correctness of the system through the simulation use of examples system in the specific scene.

Keywords:Radar stimulation,3D modeling,Visualization,Digital ball, OpenGL

目录

第一章 绪论	1
1.1 研究背景及意义.....	1
1.2 国内外研究现状.....	2
1.3 本文主要内容.....	3
第二章 可视化雷达仿真系统方案设计	5
2.1 仿真系统需求分析.....	5
2.1.1 可视化需求分析	5
2.1.2 开放式需求分析	6
2.2 系统组成框架.....	7
2.3 系统工作流程.....	8
2.4 关键技术分析.....	11
2.4.1 三维数字地理信息系统	11
2.4.2 三维建模与 OpenGL	11
2.4.3 3D 主动立体成像技术	12
2.4.4 组件技术	13
2.4.5 组件通信技术	15
2.5 本章小结.....	17
第三章 可视化雷达仿真系统开发.....	19
3.1 系统组件窗体编程.....	19
3.1.1 创建 Windows 窗体.....	20
3.1.2 添加控件	20
3.1.3 事件处理	24
3.2 仿真地图 MapX 嵌入	26
3.2.1 添加 MapX 控件.....	26
3.2.2 更改显示区域和地图标题	27
3.2.3 常用地图功能工具集	28
3.2.4 地图重置功能	29
3.2.5 获取鼠标位置经纬度	30
3.2.6 添加和显示仿真单元图标	31
3.2.7 绘制航迹	32
3.3 VVP 三维数字球与三维模型的嵌入	33

3.3.1	搭建三维数字球	34
3.3.2	三维模块装载	35
3.3.3	模型装载模块配置与设置数据路径	36
3.4	利用 OpenGL 生成 3D 主动立体显示效果	36
3.4.1	3D 主动式立体	37
3.4.2	用 OpenGL 编写立体显示程序	37
3.5	参数数据 XML 表	39
3.5.1	XML 的语法	39
3.5.2	雷达模型仿真参数表	41
3.6	仿真数据内存共享	42
3.7	UDP 仿真报文	45
3.8	本章小结	46
第四章	功能级仿真雷达模型建模	48
4.1	功能级仿真的流程	48
4.2	目标的 RCS	49
4.3	无干扰条件下信噪比的计算	50
4.4	有干扰条件下信干比的计算	50
4.5	雷达检测概率分析及计算	51
4.6	探测信息的误差计算	52
4.7	雷达仿真的实现	54
4.7.1	雷达类	54
4.7.2	位置类	55
4.7.3	集合 CollectionBase	56
4.8	本章小结	57
第五章	可视化开放式雷达仿真系统应用实例	59
5.1	软件的使用	59
5.1.1	雷达仿真集中控制平台	59
5.1.2	雷达系统的建模组件	60
5.1.3	雷达参数设置及其组件	60
5.1.4	雷达场景设置组件	61
5.1.5	雷达仿真控制组件	65
5.1.6	雷达仿真结果组件	68
5.1.7	其他组件	68
5.2	雷达仿真系统分析	68
5.3	雷达仿真系统举例	69

5.3.1 雷达仿真示例参数	69
5.3.2 雷达仿真示例结果	70
5.4 本章小结	72
结束语	73
致谢	75
参考文献	77

第一章 绪论

1.1 研究背景及意义

随着扩大其范围雷达应用的规模和复杂性的增加，雷达系统的设计，开发和测试正日益成为一个金融和费时的的工作。传统的雷达测试主要是外场，场测试模拟真实环境困难的测试昂贵和有限数量的测试等缺点，这是难以全面反映在各种复杂的环境中的性能的雷达系统中的，但是，如果该字段的测试结果不符合要求的，必须修改设计，反复的实地测试和设计修改，周期延长造成的巨大浪费和^[1]。

仿真技术的引入，可以更好地解决这些问题，其低成本，低风险，低，方便和灵活的特点，以满足雷达系统的设计，开发和测试要求。模拟使用的模型，该模型实际上是在系统中发生的复杂性的过程中，通过实验研究了系统模型的存在下或系统的设计。计算机技术的仿真技术提供了先进的工具，以加速仿真技术的发展，使得计算机模拟仿真技术中起着重要的作用。雷达系统采用计算机仿真技术的强大计算能力和良好的用户界面，完整的雷达系统数学模型仿真。雷达系统仿真已成为各种雷达系统的研究，设计和性能评估的一个功能强大的辅助工具。经过系统仿真，以确保产品的最高水平精度，可以找到产品的设计参数的最佳值，以确保获得最佳性能。不仅如此，还可以利用雷达系统仿真性能进行评估。

在近几年的快速发展，计算机硬件和软件技术，模拟雷达技术研究的不断扩大深入，雷达仿真系统，规模日益扩大，功能不断强化模拟软件系统的复杂性不断增加，从而在软件设计方法提出了更高的要求。软件设计的雷达系统仿真的作用越来越大，在很大程度上决定了仿真结果，甚至会影响到仿真系统的成功。现在越来越多的仿真研究人员也认识到这一点，开始更加注重仿真软件^[2-4]的设计和

实施。

从软件工程的角度来看，这种复杂的软件系统，目前有两个要解决的问题。首先是使用一个复杂的问题，与雷达仿真软件系统日益复杂，使用的复杂性不断增加，可用的信息量也越来越大，如果交互式会话的开发是不够友好，将在很大程度上增加了用户的困难。可视化软件，往往决定了其用户的质量水平的评价，高度可视化软件的用户，不仅容易理解，而且软件能够直观和友好提供信息。可以看出，增加雷达仿真系统的可视化是一个主要的要求。

但是软件开发和维护非常困难，传统的雷达仿真软件一般是单独开发的，每个仿真系统采用了不同的设计方法和实现结构的特定功能独特的系统。这将导致大量的重复的发展，不仅开发周期长，资金的严重浪费，也使得系统的维护和升

级变得非常困难，阻碍了雷达仿真技术的进一步发展。为了提高雷达仿真系统的开发效率，促进可重用的软件模型和互操作性，建模和仿真，以减少开支，有必要吸取一些大型软件系统的发展思路，引入基于组件的软件设计思路，实现专有系统，开放式系统架构的过渡^[5-10]。

对此，本文提出了一种雷达仿真系统的研究项目为背景，结合实际应用需求的视觉开雷达仿真系统的相关概念，综合分析，结合图形界面编程和软件组件，如技术，可视化开放雷达仿真系统，一系列的关键技术和理论进行了研究，并给出了项目的实施和应用实例。

1.2 国内外研究现状

在早期的雷达仿真技术，重点是利用数学模型来解决问题，注重过程的数学模型建模的结构特点和运行数学模型，用数学工具和手段，建立功能仿真模拟信号两种主要的数学模拟方法^[11-12]。随着计算机技术的飞速发展，利用计算机模拟复杂系统的分析可能。除了继续研究使用抽象的数学模型来描述系统，而且还充分利用新功能的计算机建模与仿真方法，并建立完善的雷达仿真软件，计算机仿真建模和仿真的雷达已经成为主要技术手段。多种计算机软件的设计思路也被广泛应用于仿真系统开发，可以简化复杂的建模过程^[13]。

20 世纪 70 年代到 20 世纪 90 年代早期的雷达仿真系统的基本模块化程序设计^[14]。从面向过程的软件设计采用模块化设计方法，包含输入/输出函数建模方法，它的好处是可以专注于建模模型的逻辑，而不必担心模型的实现细节。不足之处是基于模块化的方法仿真系统库功能仅支持通过庞大而复杂的系统仿真模型的描述不能满足需求。

20 世纪 90 年代开发的面向对象技术，雷达仿真系统的面向对象的设计方向，促进发展。模拟理论突破传统的方法和概念，使接近人类的思维方式，建模过程中，该模型具有固有的可扩展性和可重用性，面向对象的可视化建模和仿真环境的建立，有利于模拟大型，复杂系统的分析提供了一个方便的手段。然而，面向对象的设计方法也有缺点，如对象模型的元素和仿真软件和仿真系统和应用程序之间的耦合性，独立性和可扩展性的密切关系不强，缺乏仿真对象，仿真系统模型的语义信息仿真系统的对象不容易支持新的仿真应用的发展，不利于多域联合仿真系统的开发^[15]。

在可视化，图形界面编程作为一个成熟的软件开发技术，已经得到了广大用户的认可和好评。近年来，研究人员在雷达模拟雷达仿真软件的发展，也开始使用图形界面编程技术。但因为其所知只用几个简单和基本的技术，造成的雷达仿真软件的发展，软件开发技术方面的不足，虽然优于传统的字符界面的 Win32 控

制台应用程序,但仍然很复杂的操作,模拟信息不直观,更加难使用。

此外,在最近几年的地理信息系统(GIS)已经发展成为一个成熟的地图软件开发技术,被广泛用于许多商业软件。嵌入式地理信息系统中的雷达仿真软件,模拟场景将变得非常直观的设置和显示,用户可以直接在地图上图标来自模拟单元,并得到他们的位置和运动轨迹的趋势信息^[16-19],这将大大提高雷达仿真系统的可视化。

综上所述,对于当前状态的雷达仿真技术和缺点,引入图形界面编程技术,地理信息系统和开放系统的设计理念,促使雷达仿真系统设计走向可视化,人性化,通用化,易扩展,易保持方向,是非常必要的。国外在 20 世纪 90 年代,已经开始研究开放式系统的雷达领域,而国内在这方面的研究工作较少。可视雷达仿真系统的概念引入不久,工作才刚刚开始。因此,在这方面的研究,促进中国的开放式雷达系统的发展也是非常有意义的。

1.3 本文主要内容

随着扩大雷达仿真,雷达仿真软件设计和深入研究提出了更高的要求,雷达系统仿真软件设计系统的开发已成为一个重要组成部分。开放的软件设计模式可以很好地解决目前的雷达仿真软件设计,可重用性差,缺少可扩展性将成为雷达仿真软件开发的缺陷。另一方面,图像界面编程技术和地理信息系统的可视化系统的开发提供技术支持,可视化编程的雷达仿真软件将提供更好的人机交互界面,提升用户体验的雷达仿真软件,这样的雷达仿真软件专业软件更加人性化,具有更广泛的用途。允许开放的雷达仿真系统作为一种可行的技术路线开发的可视化。

本文将进入该领域的雷达仿真研究视觉的开放式雷达仿真系统的可视化技术和开放式的软件设计思路,是一个关键的技术和设计方法,实现了一系列相关支持软件,结合项目开发的例子系统。

论文各章的主要内容如下:

第一章介绍了这项研究需要的背景和意义仿真技术和雷达系统的研究现状总结,提出了研究思路:基于可视化技术和开放式的软件设计思路,采用图形化编程和组件技术的雷达仿真系统,雷达仿真系统,以改善用户体验、可重用性和可扩展性。

第二章雷达仿真系统的可视化程序设计。章节开始对雷达仿真系统可视化和开放的需求进行分析,以确定系统架构和 workflows。引入信息图形编程,地图嵌入和组件间通信问题,确定了技术手段来解决这个问题。首先介绍了图形界面的编程功能强大的 C# 语言,以及语言环境中使用 MapX 地图控件;其次组件相关的概念和特征进行分析,并总结了软件重用的想法和利益体现在组件技术;最后提出

解决通信问题组件之间的 XML，内存共享和 UDP 三种技术的。

第三章可视雷达仿真系统的开发方法进行了研究。首先，通过分析 c# 语言环境的 Windows 窗体应用程序的开发过程中，一个清晰的可视雷达仿真系统是一个关键的技术领域，其次引入 MapX 控件的编程环境和使用，尤其是雷达仿真系统特定的模拟单元图图标和目标跟踪实施问题。接着，XML 语法的基础上，并给出了雷达模型模拟参数表。最后，内存共享技术和的 UDP 技术开发的方法和关键代码。

第四章雷达功能仿真方法进行了研究。首先确定功能仿真算法流程，并给出了目标 RCS 分类和算法。随后，在一定条件下无干扰噪声比和干扰条件 SJR 的计算方法来分析和计算概率的检测，以确定雷达的探测结果。和真实的信息将被添加在目标测量误差的检测结果更符合实际情况。最后，仿真算法是关键代码的一部分。

第五章给出可视化开放式雷达仿真系统的实例，介绍它的使用和操作。在此基础上使用可视化开放式雷达仿真系统，对特定场景的模拟、仿真。

第二章 可视化雷达仿真系统方案设计

随着计算机技术的飞速发展，雷达仿真技术研究的不断扩大深入，雷达仿真系统的持续增强，规模日益扩大，日益复杂的仿真软件系统，软件设计方法也取得了更高的要求。近年来，在一个非常大的程度越来越大的作用的雷达系统仿真软件设计确定模拟的结果，甚至会影响到仿真系统的成功。现在越来越多的仿真研究人员也认识到这一点，开始更专注于模拟软件的设计和实施。

2.1 仿真系统需求分析

首先，雷达仿真系统软件必须首先完成系统建模与仿真。其次，新一代雷达模拟场景，并及时反馈信息也是必要的功能，性能评估软件和播放软件支持数据也经常被用来作为支持系统软件。此外，本研究新的雷达仿真与可视化系统和开放的需求，下面将分析这两个要求。

2.1.1 可视化需求分析

由于雷达仿真软件和通用软件不同的特性，雷达仿真软件的需求主要集中在三个方面，即接口，现场信息可视化和仿真结果。

1. 操作界面

C# 是微软推出的一个框架基于 MICROSOFT.NET 的面向对象的高级编程语言，它具有强大的图形界面编程。MICROSOFT.NET 为 C# 提供了一系列的工具和服务，以最大程度地满足开发者使用 C# 进行开发，并将它的使用在计算机领域。另外 C# 是一种非常相似 C++ 的语言，风格接近到 Java，不论 C++ 程序员或是 Java 程序员，你可以很容易地应用到 C#。因此，程序员可以像使用 Java 开发软件的过程一样，使用 C / C++ 原有的强大的功能^[21-23]。

2. 场景信息

传统的雷达仿真软件的用户是不直观的，因为许多不同的坐标，纬度和经度来反应单元的位置信息，用户可根据需要转换之前的场景的相关信息和实时仿真。如果它可以三维地理信息系统软件的嵌入式数字雷达仿真系统，用户可以直接在地图上设置的场景，可以直观地了解掌握实时仿真模拟细胞的位置信息和位置的相对运动走向。

MapInfo 的 MapX 的地图提供强大的分析功能的 ActiveX 控件产品。由于它是

基于 Windows 的操作系统标准控件，它可以支持绝大多数标准的可视化开发环境的 Visual C# 和 Visual C++，Visual Basic，PowerBuilder 的 Visual Studio.NET 等，如。使用 MapX 的嵌入式软件在软件地图功能，增强软件的空间分析功能。MapX 的 MapInfo Professional 中使用相同的基于地图的技术，可以实现 MapInfo Professional 的一个地图编辑器和空间分析功能^[25-30]。

3. 仿真结果

仿真结果是雷达仿真系统的重要组成部分，将直接影响整个系统的质量。然而，传统的雷达仿真软件往往是不直观的结果表明，其中大部分是大而复杂的特定的数据。这对用户来说是非常不直观的，对使用该软件造成了影响，也对于非专业用户制造了使用障碍。

雷达显示仿真软件开发是一个比较实用的方法。例如 PPI 显示，显示屏显示的雷达探测信息，使用户可以直观地观察目标和雷达目标的运动趋势的相对位置。

此外，Vega Prime 的使用和其他三维视景仿真软件，建立三维模型和场景模拟单元，结合实时数据，产生雷达反馈播放。这种技术还可以提高雷达仿真系统软件的可视化。

综上所述，为了开发一个可视雷达仿真系统，本文使用 C# 作为开发语言的软件基础架构和嵌入式的 MapX 的信息控制，实现了地图显示，同时 PPI 显示仿真结果显示仿真软件的发展。

2.1.2 开放式需求分析

雷达仿真系统发展面临的另一个重要问题，大量的重复开发周期长，资金的严重浪费，使系统的维护和升级成本提高。为了有效地解决传统发展模式的缺点，专门给“软件危机”，雷达仿真系统的开发，以提高效率，软件模型促进可重用性和互操作性，减少资金的建模和仿真支持，有必要使用一些大型软件系统的发展思路，采用基于组件的软件设计思路，专有系统实现开放式系统架构过渡。

开放式雷达仿真系统，需要软件开发技术为支撑。相对于结构化设计的早期应用比较广泛的当前的面向对象设计，组件设计为开放式系统更有效。支持软件重用组件技术作为当前的核心技术的飞速发展，一直受到高度重视。基于组件技术的发展历程基础上的模块化，信息隐藏等概念基础上，系统可以分解成独立的，可重复使用，可替代的组件。通过组件重用的模式可以有效地减少新代码的应用程序数量，缩短开发周期，提高软件应用程序的适应性，灵活性。可以说，基于组件的设计思路完全符合开放式系统的要求，基于组件技术的开放式雷达仿真系统的开发模式也将成为一个重要的未来技术路线图^[31-36]。

基于组件的软件组件独立，若想要工作的各个组成部分、软件正常运行，需

要在各个组件之间的通信来解决问题。多组件之间的通信常用的方法有是共享内存、命名管道和匿名管道、发送消息等。它还可以通过 socket 口，配置文件和注册表与其他组件来间接实现数据之间的通信任务。这些方法都有优点和缺点，需要选择最合适的软件通信本文开发的技术能够得到最好的软件性能。

在本文中，需要使用软件模块之间的数据通信分为以下三种：

1.参数数据。包括仿真过程中所有的数学模型的参数和软件配置参数在使用时，这种类型数据的数据量不大，但关闭后的软件的要求仍然可以保存在设备中，为了方便下次使用。例如，保留用户习惯的仿真模型和软件的默认参数的配置参数。在当前的软件开发领域，被称为 XML 数据交换技术很受欢迎，因为它具有很好的可扩展性，且易于使用，许多开发商将与它进行交互作为一个参数的数据格式。本文还使用 XML 技术作为一种通信手段。

2.模拟数据。主要是在仿真过程中生成和模拟模块之间进行交换的仿真数据，这种类型数据的数据量快速更新。这种情况是大量的数据的快速数据交换的问题。例如雷达子模块之间的信号数据模型。考虑到庞大的数据量，这些数据传输很快，并要求快速通信，通信技术的选择不多。通过直接操纵内存映射文件，共享内存和内存映射文件是独立的数据共享机制的最低水平。所以，你可以使用一个较小的高性能共享内存开销，在一般情况下，这是一个大的数据量的快速数据交换的最佳解决方案。文章将选定这种数据为数据通信模拟技术。

3 上报数据。主要是在模拟的结果和工作状态的雷达数据，这些数据需要向上级机构报告。这种数据量不大，但可以与其他计算机通信的要求。例如，雷达目标检测结果报告的数据。实现通信技术等数据的难度并不大，但由于不同的计算机之间的数据通信，因此可以考虑使用先进的网络化交流手段。通信与网络 UDP（用户数据报协议）和 TCP（传输控制协议），也许能够满足需求的常用手段。但是，由于 UDP 无与伦比的速度优势，在实际的实施过程 TCP 占用了大量的系统开销，所以，UDP 技术对本文上报的数据通信需求会更加适合。

2.2 系统组成框架

通过可视化开雷达仿真系统的要求进行分析，以确定系统的整体设计如下：

- 1 实现建模和仿真能力的雷达；
- 2 具有良好的人机交互界面且易于使用；
- 3 设置和场景信息显示可视化；
- 4 系统由组件构成，可以扩展系统功能；
- 5 组件是相互独立的，标准统一，组件能够取代；
- 6 完成的仿真过程中可以进行评估，并回放。

根据上述系统的设计，以确定插入到框架中的系统组件和系统组件，如图 2.1 所示。

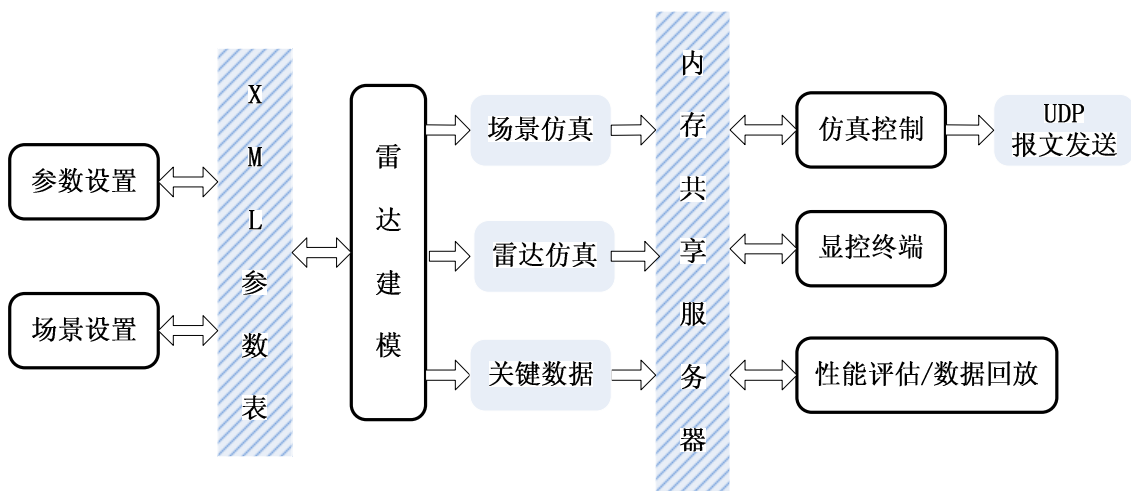


图 2.1 系统组成框架

1. 雷达建模和仿真能力，包括相关的场景模拟，杂波的仿真与干扰仿真建模型参数，建立了良好的雷达组件可以设置和修改模型参数，从而实现特定的雷达模拟。

2 场景设定元件来产生一个特定的模拟场景，包括模拟的选择区域，雷达位置设定的目标的起始位置，所述运动信息被设置，如参数设置的干扰的位置和运动信息。

3 仿真控制组件来控制仿真过程和显示仿真的情况，包括模拟的场景信息和雷达工作条件和探测信息。

4 显示和控制终端组件完全模拟真实雷达显示和控制设备，用户可以很容易地查看状态和雷达探测信息，并进行雷达操作。

5 性能评估/数据回放的组件用来存储关键数据，为整个仿真过程中的功能和回放功能，可以实现模拟雷达性能评价。

此外，参数设置和场景设置两个组件之间的组件和雷达建模作为参数表的仿真参数传递到 XML 雷达建模组件可以实现场景和雷达仿真，通过共享存储器，实时的信息将被发送到服务器仿真控制组件及显示和控制终端的组件，从而实现模拟显示功能；同样表现评估/数据播放组件共享服务器通过内存建模组件获得从雷达数据和保存，根据用户的行动数据回放性能评价工作仿真控制仿真和实时信息到用户指定的信息的 UDP 报文的形式发送到父组件。

2.3 系统工作流程

本文按照已有的系统功能设定和框架设计，确定了整个系统的工作流程。系

统已经预置可使用雷达仿真模型。当系统启动时，用户可以决定是否要使用的模型。如果你不使用时，用户将需要创建一个新的雷达仿真模型。确定的仿真模型，您需要设置模型参数，并编辑场景的信息，从而产生了情景模拟场景。启动仿真，仿真模型计算和控制，显示控制和仿真数据终端互动。这样就可以实现场景的信息显示和雷达探测结果显示和模拟控制。完成仿真后，用户可以选择是否进行性能评估和数据重放工作。最后关闭系统。系统框图如图 2.2 所示。

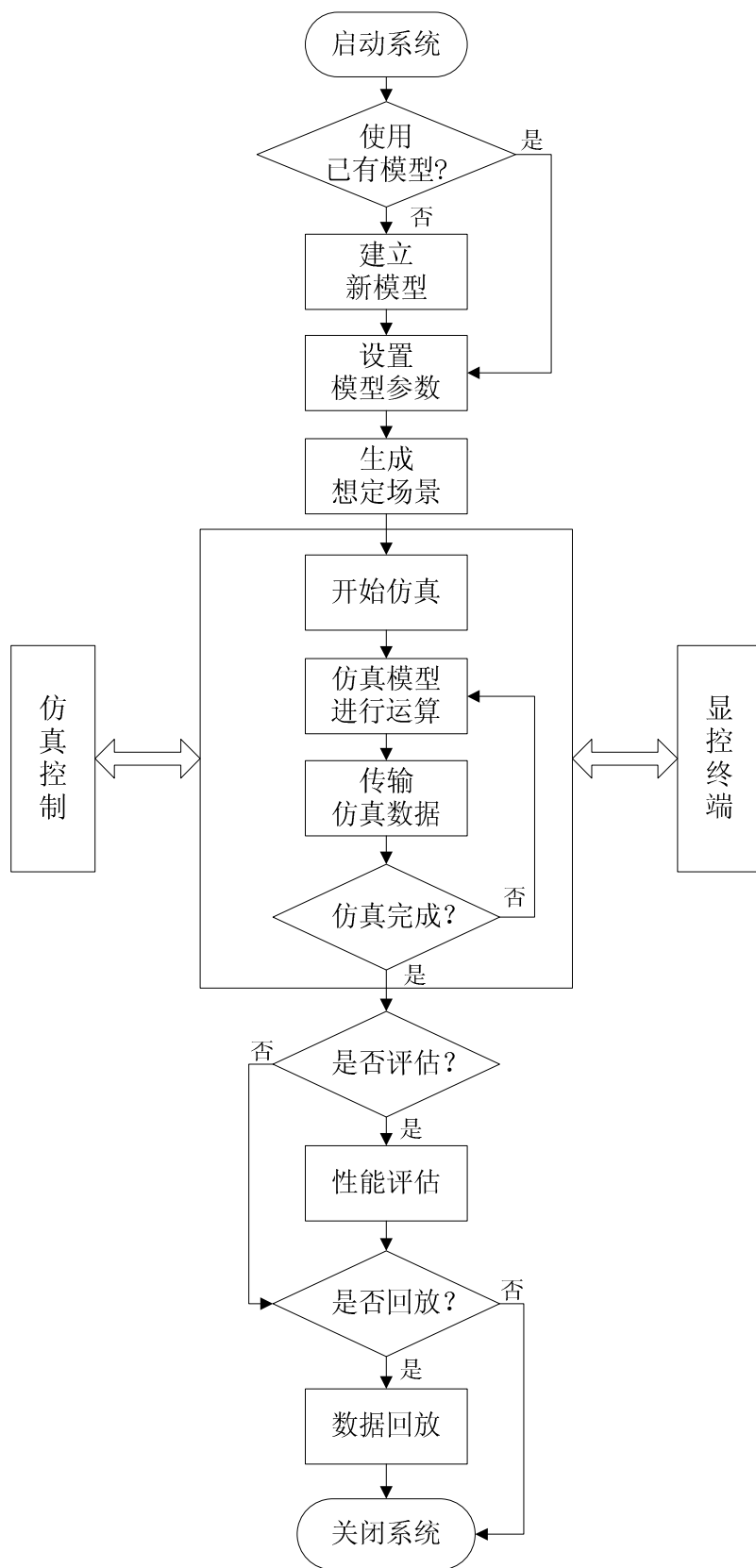


图 2.2 系统工作流程图

2.4 关键技术分析

2.4.1 三维数字地理信息系统

地理信息系统（GIS）是一个收集，存储，管理，描述，分析地球表面和空间分布的相关数据和地理信息系统。该系统是基于地理空间数据库对空间相关数据的收集，管理，操作，分析，模拟和显示的计算机硬件和软件环境的支持。地理模型分析方法和使用，提供多种空间和动态的实时的地理信息，为地理研究，评估，管理，定量分析和决策设立一个类型的计算机应用系统。随着计算机技术的飞速发展，GIS 已广泛赞赏和应用。定位和 GIS 集成也已经深入到各行各业，尤其是在军事领域。这项技术可以实现地图可视化，整合和集成，实时动态跟踪目标，并显示其地理位置在地图上的目标信息。决定允许用户得到了极大的方便，尤其是高科技军事作战任务，提供高效，快捷的决策方案。此外，这个系统船舶航行调查区调查，设计，搬迁阵营的其他方面也具有重要的应用价值。

三维 GIS 空间信息不仅打破了单调的二维平面上的限制，同时也为信息的解释和空间分析提供了一个更好的办法，同时也为业界提供一个更直观的决策支持。因此，空间信息服务的迫切需要社交应用支持二维 GIS，三维 GIS 地理信息系统已成为一个日益重要的发展方向。该系统提供了更丰富，逼真的平台，使得它很难理解抽象的空间信息可视化和可视化，并用自己的相关经验的人可以理解，从而作出准确快速的判断。

2.4.2 三维建模与 OpenGL

所谓的三维建模，是在计算机中重建真实物体或场景的三维数据，并最终在计算机上模拟的真实的三维物体或场景的使用。三维数据是使用各种三维数据采集仪获得的收集的数据，它记录了有限的离散点的表面上的各种物理参数。此数据包括的基本信息是对象的离散点的三维坐标，其表面还可以包括颜色，透明度，纹理等。遗产，建筑，医疗图像的三维建模，三维动画游戏，电影特效制作在其他领域中扮演一个重要的角色。一种三维模型的建筑过程中，其中包括一个三维的初始数据采集，如噪声的去除，简化过程中，根据以不同的方式组织的三维数据，并最终绘制在计算机中的初始数据点模型的三维特征。

三维空间数据模型主要有三种：数字高程模型（DEM），数字地形模型（DTM）和轮廓。

1. TIN（不规则三角网，简称 TIN），是指使用不规则的三角形曲面拟合，TIN 模型采用离散采样点获得的数据，按照优化组合的原则，这些离散点连接到对

方连续的三角形。任意点落在三角形顶点，边或三角形。如果该点不是顶点，高程值可以通过线性插值的方法。TIN 模型的三角面的形状和大小取决于不规则分布的样品，或节点的位置和密度。如果需要更复杂的地形变化，采样点的密度越大。TIN 三角形密集的地方，这意味着陡峭;相反，坡度较缓。地理信息系统可以是多种类型的三维数据来生成三维 TIN 模型。

2. 数字高程模型 (DEM)，通常由地面规则高程网格单元矩阵来表示的地面形势起伏。格网 DEM 数字高程模型是最常见的形式之一。其组织结构是类似的图像的光栅数据，但是每个像素值的正视矩阵。通常情况下，DEM 数据源在以下方面：首先，总台或在现场直接通过 GPS，激光测距仪测量;第二，离散数据，遥感影像航拍图像或间接通过插值，得到地形图上。地理信息系统可以是多种类型的二维数据生成三维 DEM 模型。

OpenGL (开放图形库) 是一个三维计算机图形和模型库。作为图形工作站的功能强大的 3D 图形标准，其开发的 IRIS GL 来自 SGI 图形工作站，OpenGL 的跨平台移植的发展过程中，最初是美国 SGI 公司。二维和三维的 OpenGL 实现先进的图形处理技术，性能表现极为出众，其中包括建模，转化，光处理，色彩处理，动画和更先进的功能，如纹理映射，运动模糊和其他对象。这些功能的 OpenGL 实现逼真的三维渲染，打造交互式三维景观提供优秀的软件工具。实现 OpenGL 的硬件，窗口，是相互独立的操作系统，许多计算机公司 (IBM, HP, SUN, MS) 已被纳入 OpenGL 和操作系统的各种视窗。操作系统，包括 UNIX, Windows 系列，X Window 系统的窗口时，Windows 等。OpenGL 已经成为开发的标准的三维图形，是从事三维图形开发的技术人员必须掌握的开发工具。其实是一个 OpenGL 的图形和硬件的接口，包括 120 图形功能，开发人员可以使用这些功能来创建三维模型和三维实时交互。和其他图形编程接口不同，OpenGL 提供了一个非常清晰的图形功能。因此，新手程序员可以利用 OpenGL 的图形处理能力，快速设计出三维图形和三维互动软件。很长一段时间，三维图形开发者写程序中的矩阵变换，外部设备访问脑子，和 OpenGL 等功能，同时提供了一个直观的编程环境，极大地简化了三维图形程序。如：OpenGL 提供三维图形单元，用于开发商调用的范围内。OpenGL 提供了一系列的图形变换功能。OpenGL 提供一系列的外部设备访问功能，您可以轻松地访问鼠标，键盘，空间球，数据手套。

2.4.3 3D 主动立体成像技术

如果你想提供立体 3D 显示卡的显示，你必须有两方面的技术：一维带动带有两个 3D 液晶快门。其中，三维驾驶的一部分，显示驱动器，3D (或 OpenGL) 兼容的应用软件能够直接从平板显示成一个实时三维显示。我们看到表面上的现象

是正常显示成一个双阴影显示，所以它包含一个双图像显示了各自的左眼和右眼的图像显示在显示器上，3D 液晶快门是一种高速的电子液晶快门，您可以三维显示的同步调整开关的驱动信号，以实现分离的左和右眼图像在显示器上，并立即发送眼睛周围的。因此，我们看到双重影像，显示器将会成为三维显示。当然，这种双重图像的图像必须有一定角度的偏差和缺省情况下，调整的角度。

通常情况下，由此产生的三维驾驶图像周围的隔行扫描格式，垂直格式，三种方式来改变对显卡内存的页面格式存储。隔行格式是指后半各自的左图像和右图像，然后交错合成类似的分辨率的图像。格式被吸入了个别合成为与图像的垂直格式的压缩图像的一半左右。更改网页格式是不作任何处理，左侧和右侧图像的原始分辨率，分别存储在内存中。这三个页面格式的图像格式为最佳的图像质量。本文中使用的是更改页面格式，这是饲料扫描模式。这种模式将交替左和右图像在显示屏上的扫描仪，它的第一左眼图像扫描仪，然后扫描右眼图像，如此反复。由于其扫描速度足够快（如 100Hz 的），再加上人类的视觉现象的持久性，这会让人觉得屏幕上同时显示。右眼图像时，在这个时候，一个连续的显示卡有一个三维 LCD 快门眼镜的同步信号，即，当左眼图像，3D 液晶眼镜离开镜头的开度，关闭右眼镜片；3D 液晶眼镜右镜片开闭左眼镜片，等等，左眼和右眼看到自己的形象，并在大脑中形成立体影像。

实时渲染的三维数据转换是指在三维数据可以被显示在屏幕上的数据，使用 OpenGL 或 DirectX 应用程序接口功能。在 OpenGL 应用程序接口，提供了一个多角度呈现四缓冲立体（立体声四方缓存）功能，和专业的四缓冲 OpenGL 图形卡提供了硬件支持。例如，3Dlabs 的野猫，ATI FIRE GL 图形工作站的 NVIDIA Quadro。三维显示，在图形驱动程序启用高级选项设置立体声选项，并做出相应的调整，按照相关的硬件或交换左，右眼等。

在 OpenGL 中，你必须首先使用 GL_STEREO 宏观测试显卡是否支持 stereo。

2.4.4 组件技术

从英语的“分量”的组件中，一个组件可以被看作是一个单独的模块，可提供一定的功能，通过一个公共接口的信息，并使用它，而是隐藏在内部的工作机制。相对于目前广泛使用的面向对象的技术，组件被认为是超越的对象的步骤。超越对象的地方是，它是致力于更好的需求替代，使粒度更大，更高层次的抽象的软件组件重用的组件是可能的。另一方面，也可从面向对象的角度来理解组件的概念。组件可以被认为是一种有效的方式来实现被包装的对象，你可以用它们

组装成一个较大的软件系统。

组件有三个要素：**接口 (Interface)**：组件提供服务的声明，获得用户界面的组件的功能；**实现 (Implementation)**：定义组件如何工作的，组件运行代码；**描述 (Description)**：组件的应用环境和约束条件的描述。

在一般情况下，需要将组件具有以下属性：

1. **可重用性 (Reusability)**：组件是可以重复使用的，这是基本的组件属性。仅仅是存在的价值，只可能是可重用的组件可重用的组件有自己的应用需求。可重用重用的重用性和易用性包含两方面的含义；

2. **独立 (Independence)**：组件是自成体系，独立于具体的应用，可以独立运行，分散和独立；

3. **封装性 (Encapsulizability)**：组件有一个二进制包，组件设计和隐藏实现细节，外面的世界，只有通过信息交换与外界接口的特点；

4. **组装性 (Compositability)**：组件被组装时，组件可以通过组装形成一个更大的实体，该组件来实现多路复用装置。

组件技术是一种基于组件的软件开发技术，其基本思想是一个组件组装应用软件系统的方式来建立。组件技术体现在两个方面的过程：组件的开发和基于组件的开发。组件开发生产的可重用的软件组件，基于组件的开发是利用现有的组件，开发软件系统。

组件技术支持的软件开发模式，包括元件制造，组件库管理，架构设计，组件装配和测试过程如图 2.2 所示。

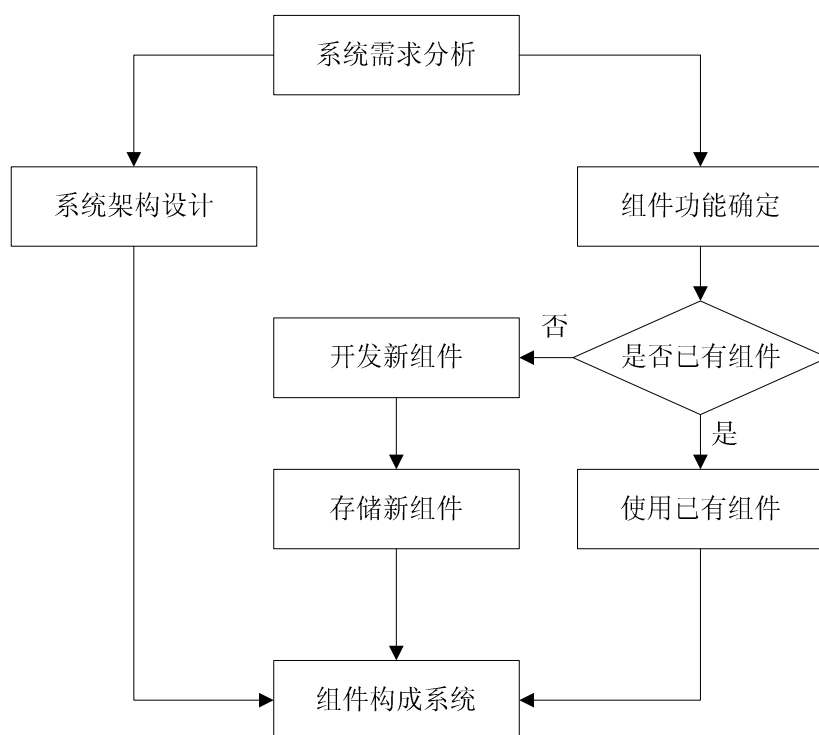


图 2.3 基于组件技术的软件开发模型

在本文中，选择的雷达仿真系统可执行文件的形式作为系统的组成部分，按照此背景下研究的规划六个部分组成：雷达建模，参数设置，场景生成，仿真控制，显示和控制的功能要求终端，性能评估/数据播放。与此同时，支持开发的软件所使用的各种组件，从而构成一个完整的雷达仿真系统。

2.4.5 组件通信技术

进程间的数据交换和共享是一个非常重要和实用技术。大多是由众多的设计师合作的程序，通常的设计是一个过程，其中只负责一个或几个模块的开发，这些模块也可以是一个动态链接库，或者可以在其他应用程序的中等规模的软件开发和设计的程序组件的形式。这些独立开发的程序模块最终将作为一个整体，即组成一个系统上运行。在该系统中，在操作过程中，每个模块往往需要频繁交换数据和共享数据。对于动态链接库和它的旋律在应用程序之间的数据交换是非常容易实现的，但之间的两个应用程序或动态链接库及其应用以外调用的应用程序进行数据交换更加困难。

以下建议的三个组成部分中第 2.1.2 通信技术进行介绍。

1 可扩展标记语言 (XML)

可扩展标记语言 (eXtensible Markup Language, 可扩展标记语言, XML), 是一种标记语言, 共同的语言数据。它允许开发人员数据结构, 从许多不同的应用程序来传输数据, 从而进行本地计算和上介绍的。XML 允许为特定的应用程序

创建一个独特的数据格式。这种语言或转让的结构化数据服务器之间的理想格式。从 XML 名以下特点分析如下：

1) 可以扩展

XML (eXtensible Markup Language, 可扩展标记语言) 的第一个字扩展“可扩展性”，这是一个强大而灵活的 XML 原因。XML，你可以创建自己的文件标记 (TAG)。在 XML 中，您可以创建任何标记，你可以充分发挥想象力，以文件一些朗朗上口的标签名称。例如，文件中包含学生的信息，我们可以创建一个文件称为<Students>标签，然后在“学生”>“创建其他的标签。只要清晰，容易理解，你可以创建任意数量的自己的标志。

2) 标记

第二个字是 XML 可扩展标记语言标记“标签”，其目的是为了说明 XML 标记文档中的元素。标记很容易理解的精髓，如果没有标明，在计算机上的文件似乎只是一个很长的字符串，每个字符看起来是一样的，有没有点上的重点。通过标记，文档，使得它容易阅读和理解，可以分成几段，指定标题。XML 可以使用它的可扩展性，以建立更适合文档标记。

3) 语言

XML 可扩展标记语言，是第三个词语言“的语言。”任何语言都有它自己特定的语法规则，它显示了在 XML 文档的编制必须遵循特定的语法。

2 内存共享

在数据交换过大，过于频繁的情况下，数据通信交换将难以实现，实际应用中可以实现大量数据的快速交换进程之间通过共享内存。要执行的操作，直接通过共享内存是内存映射文件，内存映射文件是独立的数据共享机制的最低水平。因此，您可以使用共享内存开销较小的高性能是携带大量数据的快速数据交换的最佳解决方案。

在 Windows 操作系统中，这个过程不会允许任何读，写或修改另一个进程中的数据（包括变量，对象和内存分配等），但在一个进程创建一个视图文件映射对象能够映射的其他进程数。这些进程共享相同的物理内存页。因此，当一个进程将数据写入到共享文件映射对象视图，其他进程可以立即得到的数据来改变这种状况。为了进一步提高数据交换的速度，但也通过系统页面文件支持，直接使用内存映射文件中使用的存储器区域，这个进程之间共享内存进行了充分满足大量的数据需要快速的数据传输任务。

首先通过的 CreateFileMapping () 函数创建一个内存映射文件对象，如果创建成功通过 MapViewOfFile () 函数查看此文件映射对象映射到地址空间，同时也得到了第一个地址本地视图。

数据传输实际上是写入的数据从发送者到共享内存中，然后你就可以读取接

收程序。数据传输完成后，将退出程序，你需要要映射的内存映射文件对象视图卸载和释放资源和其他治疗。这部分的工作由 `UnmapViewOfFile()` 和 `CloseHandle` 的 `()` 的功能，如完整的。

`OpenFileMapping()` 需要在接收过程中函数的名称，打开“数据映射”规定的文件映射对象，如果执行成功，继续用 `MapViewOfFile()` 函数查看此文件映射对象映射到接收应用程序的地址空间，并获得它首先解决的问题。

实际测试中，使用共享内存来处理大量的数据，快速的数据交换表现出优异的性能，可靠性等方面的数据应该是远远高于其他常见的方法。这种大容量，高速的数据共享高速数据传输的通信软件的设计方法，具有良好的使用效果。

3 用户数据报协议 (UDP)

用户数据报协议 **UDP** 是缩写代表，网络用户数据报协议 **UDP** 数据包进行处理，是一种无连接的协议。该协议用来支持那些需要计算机网络应用程序之间的数据传输。包括网络视频会议系统，包括众多的客户机/服务器模式的网络应用需要使用 **UDP** 协议。如 **ICQ**，**QQ** 聊天，使用 **UDP** 协议使用。

UDP 协议的主要作用是将网络数据流量压缩成数据包的形式。一个典型的数据包是一个二进制的数据传输单元，每个数据分组包含使用的标头信息的前 8 个字节中剩余的字节被用于包含在特定的传输数据。

UDP 报头由四个结构域，其中每个域占用两个字节，如下所示：

- **UDP** 源端口号
- 目的端口号
- 数据报长度
- 校验和值

UDP 协议使用端口号为不同的应用保留其各自的数据传输通道。因为 **UDP** 报头使用两个字节存储的端口号，所以端口号的有效范围是从 0 到 65535。 **UDP** 协议头的校验，以确保数据的安全性。然而，当检测到错误时，**UDP** 做纠错，而是简单地扔掉损坏的部分的消息，或消息给应用程序提供了一个警告。

虽然 **UDP** 是不可靠的协议，但它是一个理想的传播信息的协议。例如，在屏幕上的报告，股市，航空信息显示在屏幕上，并依此类推。 **UDP** 也使用路由信息协议 **RIP**（路由信息协议）修改路由表。在这些应用中，如果消息是失去了在几秒钟后，另一个新的消息取代。

2.5 本章小结

本章首先对可视化开放式雷达模拟可视化系统的需求和开放式的要求进行了分析，以确定分工的系统组件和系统组件的框架，并确定系统的工作流程。并引

入信息的图形界面编程，地图嵌入和组件间通信问题，确定技术手段来解决这个问题。首先介绍了图形界面的编程功能强大的 C# 语言，以及语言环境使用 MapX 地图控件;其次实现发式思想组件技术相关的概念和特点进行分析，总结了组件技术体现了软件复用的想法和优势;最后介绍了 XML、内存共享和 UDP 来解决组件间通信问题。

第三章 三维可视化雷达仿真系统开发

这些技术在不同领域有不同的功能和用法，了解每一项技术的基础知识，结合雷达仿真系统中详细描述每个技术在应用中的雷达仿真系统和方法的特点和需求。本章将首先通过分析 C# 语言环境的 Windows 窗体应用程序的开发过程中，一个清晰的可视雷达仿真系统的关键技术领域，其次是引进的 MapX 控件的编程环境和使用，尤其是雷达模拟系统的具体目标跟踪模拟单元图标和绘图功能的实现。接着，XML 语法的基础上，并给出了雷达模型的仿真参数表的设计。最后，内存共享技术和的 UDP 技术开发的方法和关键代码。

3.1 系统组件窗体编程

GUI（图形用户界面 GUI）是程序与用户进行交互的方法，使用它可以接受用户输入的用户输出的运行结果。用户界面的设计和施工，软件开发是一项重要任务。用户界面功能完善，方便使用，将直接影响到用户的应用软件。图形用户界面图形化菜单，按钮，和其他标准界面元素和鼠标操作，帮助用户轻松地到计算机系统发出指令，启动操作，并运行在同一系统的结果，在以图形方式显示给用户。生动的图形用户界面屏幕，操作方便，已成为事实上的几乎所有标准的应用软件。

。NET 中，提供了一系列的书面形式的基于 Windows 的应用程序类别。专注于这些类 System.Windows.Forms 和 System.Drawing 命名空间。其中包含了超过 200 个的类和接口。许多 System.Windows.Forms 命名空间中的类的描述 Windows 图形用户界面元素，如按钮，列表框，菜单，和共同的对话，包括表格和控制是最重要的。表格的形式，描述了一个窗口或对话框，这是所有窗口的基础类。控制是控制类，它是可视化组件的基类，所以它形成一个图形用户界面的基础。

图形用户界面的设计和实施工作主要集中在以下四点：

- 1 创建一个表单（form）：创建一个表单，以适应各种其他的界面元素；
- （2）建立的控制（控制）：创建不同的元件，如按钮，文本框等组成的接口；
- 3 指定布局（布局）：安排他们的位置关系的具体需求；
- 4 响应的事件（事件）：定义图形接口事件的响应不同的事件中，为了实现一个图形用户界面和用户交互的用户界面元素。

3.1.1 创建 Windows 窗体

一般 Windows 窗体应用程序启动的一种形式，在 Microsoft Visual Studio 中，创建一个新的 C# 解决方案使用“Windows 应用程序”模板中可以得到一个空白的 Windows 窗体。

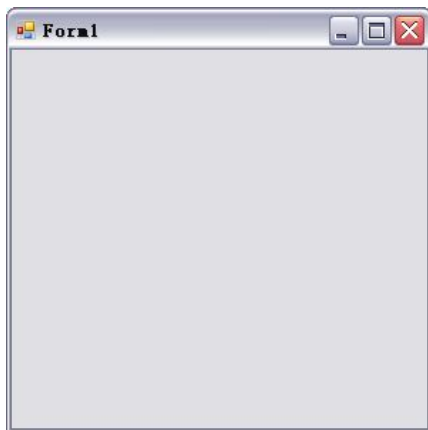


图 3.1 空白 Windows 窗体

创建一种形式的过程中，申报 Form 类，然后实例化这个类过程。Windows 窗体设计器生成的代码，为用户完成的过程中，用户在可视化编辑器也记录在代码中所做的更改，也由 Windows 窗体设计器生成的代码，它是最好不要手动编辑。

3.1.2 添加控件

添加窗体上的控件，其实在这个过程中，产生一个控制实例，并添加到表单的过程。一些常用的控件，如文本框（TextBox 的），按钮（按钮），标签（Label）和其他相应的类。控制可以使用 new 运算符，如：

```
this.textBox1 = new System.Windows.Forms.TextBox();
```

控件的生成，可以设置相应的属性，如：

```
this.textBox1.Location = new System.Drawing.Point(98, 105);
```

```
this.textBox1.Name = "textBox1";
```

```
this.textBox1.Size = new System.Drawing.Size(100, 21);
```

```
this.textBox1.TabIndex = 0;
```

其中 TabIndex 为索引顺序，，Size 为尺寸，Name 为名字，Location 为位置。

生成控件实例将它添加到表单中，你可以使用 Form 类的实例方法 Add 或 AddRange，增加了控制前者，这是增加了多个控件（控件数组）。的条件，如

```
this.Controls.Add (this.textBox1) ;
```


Microsoft Visual Studio 中提供了一个可视化的方法来添加控件在 Windows 窗体中的可视化编辑器，可以通过拖放控制“工具箱”添加控件。工具箱图 3.2 左，包括所有的系统可以用在控制资源。



图 3.2 工具箱和属性对话框

将窗体上的控件，你需要设置好多个控件的位置，大小，顺序关系，也就是说，每一个空间的布局，达到美观的目的。

每个控件都有一个属性值，指定不同的属性，包括外观，布局，行为的不同方面的数据控制。如图 3.2 所示。

您可以使用四个种类的财产的形式定位控制和调整大小控制：

1. 位置，以像素为单位，设置控件的 X 和 Y 坐标；
2. 尺寸，像素，设置控件的宽度和高度；
3. 锚，连接到一个或多个边境上的控件的形式；
4. 码头，控件和窗体的一个或多个连接的边界。

本文研究，将结合这些控件的具体使用上的某些控件的使用以下软件：

1. 按钮，按钮控制。按钮控件是一个图形界面编程中最常用的控件通常被用作一些功能函数入口。该软件使用了大量的按钮控件实现的功能调用。此外，如果按钮控制“CheckOnClick”属性赋值为 True，按钮被点击切换其选中状态，类似的 CheckBox。“仿真开始”按钮，以这种方式使用，点击按钮来改变后，已

被按下的状态，表示正在进行中该仿真。



图 3.3 按钮控件的使用

2. 文本框，文本框控件。文本框控件一般用来接收用户输入的各种信息，也可以用来显示给用户的软件的特定信息。该软件可实时显示使用一个文本框控制鼠标位置的纬度和经度信息，波束指向模拟当前仿真时间信息和其他信息。

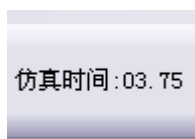


图 3.4 文本框控件的使用

3. 的 ComboBox, ListBox 控件可以进行编辑。可编辑的下拉列表框控件提供数据给用户组合框选项供用户选择，在下拉列表框中的顶部是一个文本框，允许用户类型，而且还支持用户自定义数据。在某些参数设置，一些常用软件的默认值供用户选择，同时还接受用户输入自定义值。

4. 标签控制标签栏。标签栏控件来显示一些特定的静态文本，通常用来识别和指示。

5. 面板，面板控制。控制面板是一个容器类控制，其作用通常是提供其他控件分组的形式来实现分割功能。此外，该软件将关闭边界，在表格上为的“MouseHover”发件人的情况下，当鼠标悬停时，请更改侧栏“Visable 的”属性（也侧边栏面板控制）以达到活化左边，上下两个侧边栏自动隐藏功能。

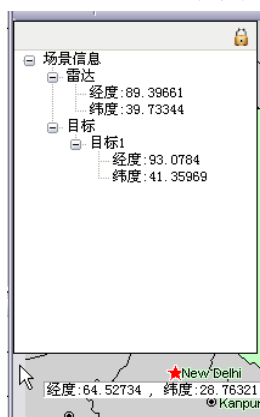


图 3.5 可自动隐藏的侧边栏

6. 树视图，树视图控制。使用树状视图控件，您可以按下 Windows 窗体分层树形结构显示数据，在 Windows 操作系统资源管理器，在左窗格中显示相同的文件和文件夹。树视图中的每个节点都可以包含子节点，用户可以展开或折叠显示父节点。该软件用于在树视图中显示每个数值模拟的实时位置信息单元的经度和

纬度。

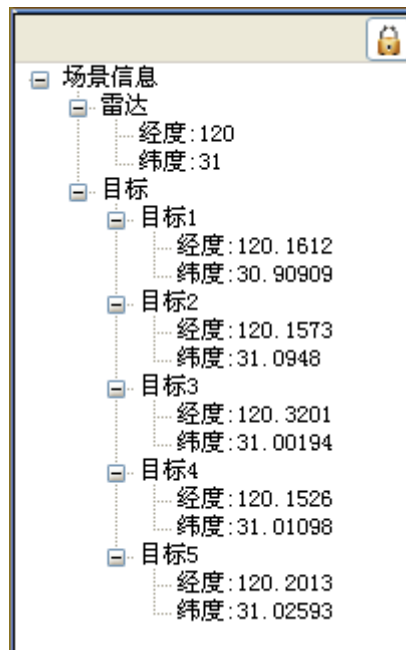


图 3.6 树状视图控件的使用

7. **ListView** 中，列表视图控制。显示列表视图控件是一家集空间的项目中，有四种显示方式，即，大图标，小图标，列表，显示的细节。在选择列表中的软件，在列表视图控制装置，用于显示雷达信息。

序号	航迹编号	距离 (m)	方位角 (°)	俯仰角 (°)	X轴坐标 (m)	Y轴坐标 (m)	Z轴坐标 (m)
131	3	20194.2271695362	31.0446028714137	0.0033107951056...	17301.7292436611	10414.2678471508	1.16690878468114
130	2	35211.2859840241	0.311226930230041	0.144927659099576	35210.6538721441	191.263852078544	89.0856144861269
129	2	35205.2259598228	0.308495130796282	0.147635744027708	35204.5987861731	189.552395474283	90.7142554104455
128	1	20351.0417232238	-29.512780156127	0.0794918280423739	17710.3922800993	-10025.2734274651	28.2349067227201
127	1	20341.0478382664	-29.5220589180373	0.0804138134016071	17700.0717770434	-10023.2166113053	28.5483625873857
126	3	20189.4979578651	31.0236947085307	0.0057598129214...	17301.4756510138	10405.5160356767	2.02960378594989
125	2	35203.1279349654	0.300679512449582	0.155950068454356	35202.5127927233	184.739114146081	95.8172395820502
124	3	20183.0725362287	31.0037136600004	-0.010224546763799	17299.5957325639	10396.1719647288	-3.6017097531839
123	2	35196.8188875263	0.30684823468759	0.17340670869825	35196.152880976	188.495221193385	106.544506578531
122	1	20337.0886569447	-29.535035789898	0.0859363676959146	17694.3540012009	-10025.2721202659	30.5030296612861

图 3.7 列表视图控件的使用

8. **MenuStrip** 中，菜单栏控制。菜单栏控件可以为用户提供一个菜单栏，开发人员可以自定义菜单项号，标题，和隶属关系。现有的 Windows 窗体应用程序菜单栏包含了基本的控制，但也最常用的控件之一。

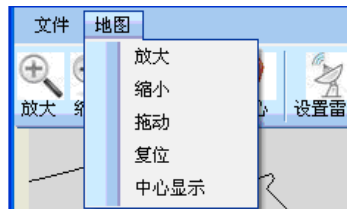


图 3.8 菜单栏控件的使用

9. 工具条，工具栏控制。工具栏控制一般用于集成应用程序的各种常用工具，实际上是一个容器。它可以包含 **ToolStripButton** 的，**ToolStripComboBox**，**ToolStripSplitButton**，**ToolStripLabel** 上，**ToolStripSeparator**，

ToolStripDropDownButton, ToolStripProgressBar ToolStripTextBox 控件。该软件使用一个 ToolStripButton 的, ToolStripDropDownButton, ToolStripSeparator ToolStripTextBox 控件。



图 3.9 工具栏控件的使用

10. 的 OpenFileDialog, 打开对话框控件。Windows 窗体 OpenFileDialog 控件是一个必要的预配置对话框中, 配置开放的条件, 就可以在“打开文件”对话框, 同样的操作和功能与 Windows 操作系统。

11. 定时器, 定时器控制。定时器控制是一个 Windows 窗体环境设计引发该事件的周期性定时器控制。属性“间隔”的决定, 其值以毫秒为单位的时间间隔长度, 如果控制开启时, 每次间隔时间引发了“蟀”事件。软件仿真使用定时器, 定时器控制仿真速度。

3.1.3 事件处理

事件处理的图形界面的应用程序的开发, 是最重要的方面, 这是程序与用户交互的机制。如果用户单击该按钮时, 程序将响应这个点击事件, 并调用相应的函数来执行任务。

窗体和控件使用。NET 的事件机制与其他形式和控制, 通信, 事件仅仅是一个形式或控制的通知, 这让程序知道有一些事件, 如点击按钮, 文本框, 列表框条目被选中, 移动鼠标或按下的形式, 等等。

“事件发件人”(事件发件人)表明该事件源产生的事件。“事件接收机”(事件 RECEIVER)的活动感兴趣。人们希望收到通知的收件人, 发件人, 注册一个处理程序在适当的时候, 发件人请求处理程序。

每个控件都有一个“事件”, 由用户添加事件处理程序, 如下图 3.10 所示。

我们给一个按钮控件“Button1”的事件“, 点击”添加事件处理程序为“button1_Click 的”。点击这里点击鼠标事件, 该事件源是 button1 时, button1_Click 事件处理函数。到 button1_Click 作为系统的功能将在用户点击按钮, 处理它。

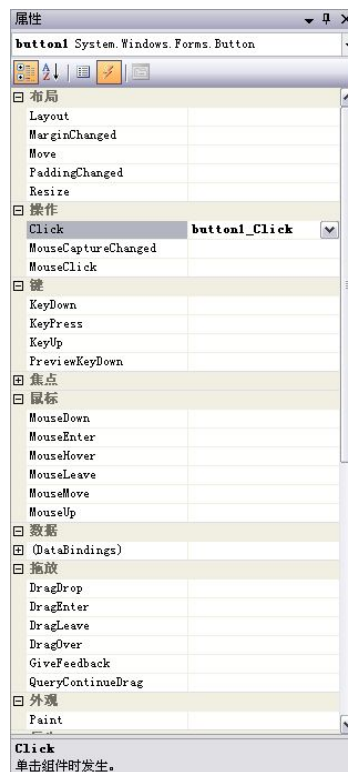


图 3.10 事件对话框

如本文所用，这些事件包括：

1. **Click**，单击事件。是用户控件的 **Click** 事件，用鼠标左键控制区发生的事件时，点击，单击“操作”是指新闻和电梯的组合。每个控制最常用的按钮点击事件控制是，但 **click** 事件不只是一个事件源按钮控制，可以一按事件的事件源。该软件添加一种新型的模拟单元的选择，你可以单击任何地方在地图上添加它的位置，这是在使用地图 **MapX** 的控制的点击事件作为触发添加一个新进入模拟单元。使操作非常人性化，而不是通过输入数字来确定的地理位置。

2. **MouseMove**，鼠标移动事件。当用户移动鼠标在控制区域，将触发“的 **MouseMove**”事件，与事件系统还提供实时位置的鼠标，供开发人员使用。该软件采用了鼠标光标移动的文本框中显示实时的鼠标位置经纬度。这是地图控制“的 **MouseMove**”事件响应函数来完成。

3. **MouseDown**，鼠标按下事件。事件与鼠标的动作，最常用的是点击事件，但是当开发者需要获得位置的鼠标点击事件发生时，有肯定有错误的。因为鼠标点击事件是一个“的 **MouseDown**”事件和“的 **MouseUp**”事件组合，有意或无意的，因为用户移动鼠标时，两个次子事件的位置会有所不同，系统将采取“**MouseUp**”事件位置点击事件位置。因此，本文使用的软件“**MouseDown**”事件添加新的模拟单元为用户指定的位置。

4. **MouseHover**，鼠标悬停事件。鼠标悬停事件是指用户控制鼠标光标在很短的时间内当休息区事件。该软件采用了鼠标悬停事件激活软件窗口的左侧，下，

右三个侧边栏。

5. **Load**，窗体加载事件。当软件被加载时，其形式，作为一个软件初始化功能，该软件使用一些事件响应功能的筹备工作。

6. **Tick**，时间间隔事件。“嘀”事件被设计为 **Windows** 窗体环境定期提高事件定时器控制定时器控制特定事件。如果控制开启时，每个时间间隔引发了“嘀”事件。属性“间隔”的决定，其值以毫秒为单位的时间间隔长度。软件仿真使用定时器，定时器控制仿真速度。

3.2 仿真地图 MapX 嵌入

MapInfo 公司的 MapX 的开发，能够为用户提供强大的分析功能映射的 ActiveX 控件产品。由于它是基于 **Windows** 的操作系统，标准控件，可以支持最标准的 MapX 的可视化开发环境，如：**Visual Basic** 中的 **Visual C++**，**Visual C#** 中，**Delphi** 和 **PowerBuilder** 的。利用 MapX，用户可以简单，快速地嵌入在自己的应用程序映射功能，增强了应用的空间分析功能，实现增值应用。空间技术的一个结果，所以应用程序可以从 MapInfo 软件平台运行。

安装 MapX 控件，安装和注册控制用户的计算机实际上为用户提供了一个 Map 对象，它具有多种属性和方法提供给用户。

3.2.1 添加 MapX 控件

使用 MapX 中，首先需要手动添加 MapX 控件到 Microsoft Visual Studio 的“工具箱”。点击微软 Visual Studio 的“工具”菜单项下的“选择工具箱项”

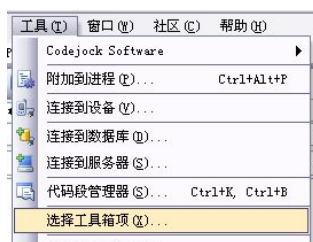


图 3.11 “选择工具箱项”菜单位置

选择“COM 组件”选项卡，找到“MapInfo MapX 的”选择“项，然后单击”确定“。

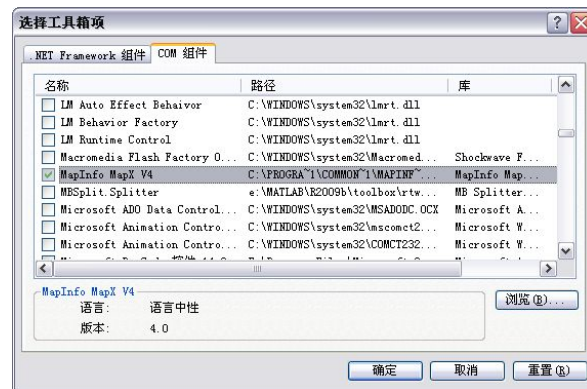


图 3.12 “选择工具箱项”对话框

MapX 控件将被添加到“工具箱”，以后可以直接调用相同的其他控件。

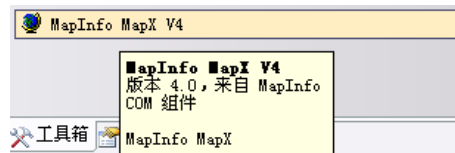


图 3.13 工具箱中的 MapX 控件

向现有窗体增加一个 MapX 控件。



图 3.14 窗体中的 MapX 控件

用户可以容易改变的属性或利用 MapX 提供的功能来达到目的的可定制功能，基于 MapX 二次开发完成的应用 MapX 控件。

3.2.2 更改显示区域和地图标题

可以应用更改 “Geoset” 属性来改变软件中的地图

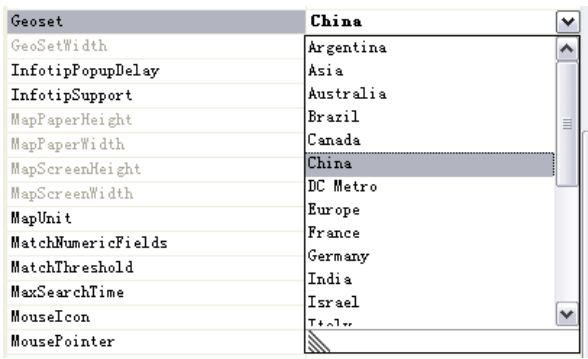


图 3.15MapX 控件的“Geoset”属性

更改的 MapX 控件“TitleText”属性，你可以改变软件地图标题，地图标题，如果没有，你可以指定一个空值的属性，下面的代码：

```
axMap1.TitleText = "";
```

3.2.3 常用地图功能工具集

MapX 的控制中心显示，漫游，提供了一套工具，常用的地图功能，如变焦，添加标签等。用户可以改变其的“的 CurrentTool”属性指定所需的地图工具，如图所示。

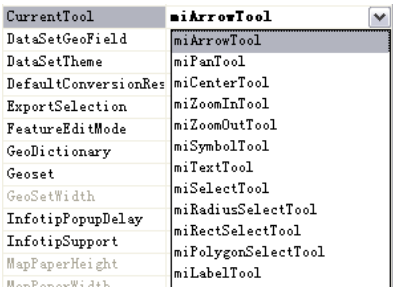


图 3.16MapX 控件的“CurrentTool”属性

- miArrowTool: 箭头工具，没有特殊的功能，可显示的位置;
- miPanTool: 漫游工具，地图漫游功能，你可以拖动地图;
- miCenterTool: 中心显示工具，该中心显示，移动鼠标在地图上点击窗口中央;
- miZoomInTool: 缩放工具，缩放地图功能，以鼠标点击地图放大地图显示的中心;
- miZoomOutTool: 狭窄的工具，在地图上的功能，以鼠标点击的位置为中心在地图上显示;
- miSymbolTool: 添加符号工具，添加鼠标点击的位置在地图上预设的符号，符号样式 MapX 控件 ActiveX 属性对话框修改;



图 3.17 MapX 控件的符号样式对话框

iTextTool: 添加文本工具，添加鼠标点击地图上的位置，并允许用户在文本框中输入文本风格的文本框中。

miSelectTool: 选择“工具”，选择鼠标指针地图区域;



图 3.18 使用选择工具选择区域

miRadiusSelectTool: 圆形选区工具，鼠标指针的位置的中心鼠标点击，鼠标位移半径的圆圈，选择圆形内区域的所有地图映射;

miRectSelectTool: 矩形选择工具，将鼠标指针在地图上的位置点击鼠标顶部的矩形，矩形的对角线位移鼠标离开顶点，选择的矩形区域内所有的地图;

miPolygonSelectTool: 多边形选择工具，将鼠标指针，鼠标点击地图顶点的多边形，双端多边形选择每个位置，选择地图区域之内的所有多边形;

miLabelTool: 添加标签工具，将鼠标指针鼠标指针地图区域添加姓名标签;

在开发接口属性对话框，选择“仅在程序启动时的默认工具，可以指定的 **CurrentTool**”属性。在实际应用中，用户往往需要改变在运行命令映射工具，开发人员可以编写当前地图工具做出改变。

3.2.4 地图重置功能

通过 MapX 控件的三个属性值，以确定当前的位置和规模所显示的地图，他们是“项 CenterX”，“项 CenterY”和“缩放”。其中，“项 CenterX”和“项 CenterY”的中心，将当前显示的位置的经度和纬度值，“缩放”是当前的规模为所显示的值。

地图复位功能的实现，可以节省这三个属性的值在程序启动时，程序运行时通过调用 MapX 控件 ZoomTo（双变焦，双 X，双 Y）函数来指定显示位置控制的

规模，为了实现该功能，即可重设地图。

除这三个属性的数字代码：

```
double resetZoom, resetX, resetY;
```

```
resetZoom = axMap1.Zoom;
```

```
resetX = axMap1.CenterX;
```

```
resetY = axMap1.CenterY;
```

在需要地图重置的事件响应函数中添加如下代码即可：

```
axMap1.ZoomTo(resetZoom, resetX, resetY);
```

3.2.5 获取鼠标位置经纬度

为了获得当前鼠标位置经纬度点需要获得鼠标当前位置的像素值，然后用 MapX 的提供转换功能进行转换，转换为纬度和经度值。MapX 的转换函数：

虚空 ConvertCoord（文献，楼盘浮浮 screenX screenY，双 MapX 的文献，文献，双 mapY 短的方向）

这个函数有没有返回值，一共有五个参数，即转换和转换方向转换后的笛卡尔坐标，纬度和经度值的像素的位置。

在本文中，使用的软件，我们需要到有随时鼠标的位置，并显示用户的理解现场的信息，所以这个选择 MapX 的控制“鼠标”事件响应函数为这个事件来添加的代码来实现上述功能。MapX 控件的“鼠标”的时间响应函数的形式：

```
void axMap1_MouseMoveEvent(object sender,
```

```
AxMapXLib.CMapXEvents_MouseMoveEvent e)
```

参数 AxMapXLib.CMapXEvents_MouseMoveEvent E 包含当前鼠标的位置信息，将这一信息，改变方向使用 MapX 的命名空间的提供 MapXLib.ConversionConstants.miScreenToMap 的，并将它显示在文本框中（文本框），用户可以实时了解当前鼠标点的经度和纬度。下面的代码实现。

```
void axMap1_MouseMoveEvent(object sender,
```

```
AxMapXLib.CMapXEvents_MouseMoveEvent e)
```

```
{
```

```
double mLongitude = 0;
```

```
double mLatitude = 0;
```

```
axMap1.ConvertCoord(ref e.x, ref e.y, ref mLongitude, ref mLatitude,
```

```
(short)MapXLib.ConversionConstants.miScreenToMap);
```

```
label1.Text = "经度:" + Convert.ToString(mLongitude).PadLeft(6) + ", 纬度:" +
```

```
Convert.ToString(mLatitude).PadLeft(6);
```

```
}
```

为了提升用户体验的软件，本文将显示的经度和纬度的文本框设置为跟随鼠标光标，添加上述功能的最后一段中的代码：

```
= label1.Top (INT) e.y+90;
```

```
= label1.Left (INT) e.x+15;
```

结果如图 3.19 所示。



图 3.19 经纬度浮动文本框

3.2.6 添加和显示仿真单元图标

根据可视雷达系统仿真软件的使用要求，需要添加地图和显示模拟单元的自定义图标，如雷达，目标和干扰。这是必要的使用添加 MapX 控件开发的技术元素。添加一个原始的需要添加自定义层。

```
LayerInfo pLayerInfo = new LayerInfoClass();
```

```
Fields pFields = new FieldsClass();
```

```
object o = Type.Missing;
```

```
pFields.AddSmallIntField("ID", 1);
```

```
pLayerInfo.Type = (short)LayerInfoTypeConstants.miLayerInfoTypeTemp;
```

```
pLayerInfo.AddParameter("Name", "strTempLayerName");
```

```
pLayerInfo.AddParameter("Fields", pFields);
```

```
axMap1.Layers.Add(pLayerInfo, 0);
```

然后，我们需要定义风格的元素加入实例 MapXLib.Style 的。以下 pStyle 在这里被定义软件的基本风格：

```
MapXLib.Style pStyle;
```

```
pStyle = new StyleClass();
```

```
pStyle.LineStyle = (short)PenStyleConstants.miPenSolid;
```

```
pStyle.LineColor = Convert.ToUInt32(ColorTranslator.ToOle(Color.Blue));
```

```
pStyle.LineWidth = 3;
```

```
pStyle.SymbolType = (short)SymbolTypeConstants.miSymbolTypeBitmap;
```

```
pStyle.SymbolBitmapName = "PIN2-32.BMP";
```

```
pStyle.SymbolBitmapSize = 32;
```

```
pStyle.SymbolBitmapTransparent = true;
```

开发人员可以使用这种风格的添加元素，绘制图形，定义包含一个原始的设置信息和平局线样式信息。实心线线条样式，线条的颜色是蓝色，3 为位图文件，原始类型的线宽，图元文件称为“的 PIN232.BMP”，像素的大小是 32 和透明的像素设置为透明。

在这里我们就可以创建层描绘出它们的内容。要添加一个雷达图标，例如：

```
pStyle.SymbolBitmapName = "radar_1.BMP";
```

```
MapXLib.Point tempPoint = new MapXLib.Point();
```

```
tempPoint.Set(radar.movingInfo.startPosition.longitude,
```

```
radar.movingInfo.startPosition.latitude);
```

```
MapXLib.Feature newSymbol = axMap1.FeatureFactory.CreateSymbol(tempPoint,
pStyle);
```

```
objLayer.AddFeature(newSymbol, Type.Missing);
```

objLayer 添加提前层实例。首先，原始风貌位图文件到一个自定义的雷达图标“radar_1.BMP”，然后定义一个的 MapXLib.Point 点的实例，并分配到初始位置的雷达的位置和使用的图元定义的样式点产生 MapXLib。功能实例，并加了一层事先实例 objLayer 添加。需要注意的是用户定义的位图文件需要存储在 C: \ Program Files 文件\ Common Files 文件\ MAPINFO 共享\ MapX 的通用\ CUSTSYMB 目录，你可以使用正常使用。

3.2.7 绘制航迹

在地图上绘制的模拟单元是一个可视化的雷达跟踪仿真系统的一个重要特征。不同于传统的模拟坐标数值信息，运动信息，通过在赛道上显示的地图信息，软件用户可以非常直观地了解每个模拟单元的运动趋势和相对位置，使得模拟更加生动，更加友好的人机交互。

在 MapX 控件还需要绘制一个自定义层的轨道上，开发人员可以使用的功能 CreateLine（对象点对象样式）函数来生成 MapXLib.Feature 线实例，并的使用功能 AddFeature（对象 SourceFeature，对象 rowValues 参数）功能添加在自定义层。

首先定义的 MapX 控件 MapXLib 命名空间提供的集合点实例类航迹点，经度和纬度值分配到各组，漆线数据和曲目信息。然后使用画线定义的样式 pStyle 的加了一层：

```
MapXLib.Feature newLine = axMap1.FeatureFactory.CreateLine(trackPoints, pStyle);
```

```
objLayer.AddFeature(newLine, Type.Missing);
```

在信号级仿真，我们可以用不同的颜色在地图上绘制的真正目标轨迹和目标

探测雷达跟踪比较两个错误评估雷达的性能。在这种情况下，我们需要更改线条颜色的绘画风格 pStyle，在平局。假设目标的真实运动轨迹和雷达目标检测点集轨迹 MapXLib.Points 的类实例 realPoints detectPoints，线条，颜色是蓝色和红色，那么实现的代码如下：

```
pStyle.LineColor = Convert.ToUInt32(ColorTranslator.ToOle(Color.Blue));
MapXLib.Feature newLineReal = axMap1.FeatureFactory.CreateLine(realPoints,
pStyle);
objLayer.AddFeature(newLineReal, Type.Missing);
pStyle.LineColor = Convert.ToUInt32(ColorTranslator.ToOle(Color.Red));
MapXLib.Feature newLineDetect =
axMap1.FeatureFactory.CreateLine(detectPoints, pStyle);
objLayer.AddFeature(newLineDetect, Type.Missing);
```

效果如图 3. 20。

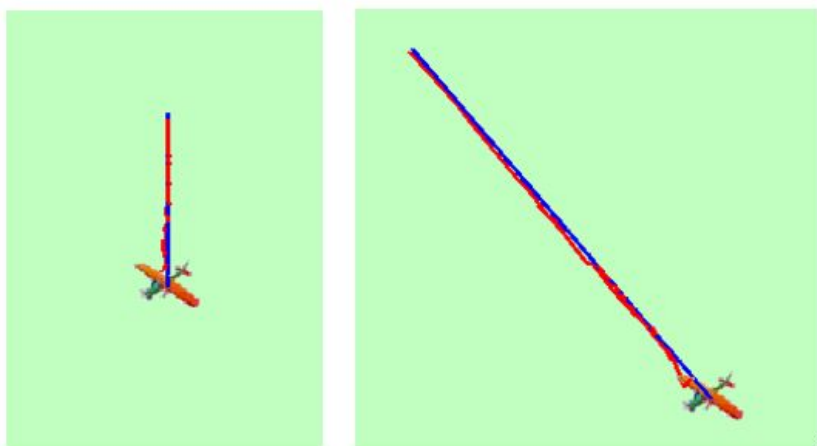


图 3. 20 目标轨迹和探测航迹的显示

3.3VVP 三维数字球与三维模型的嵌入

VVP-3D 平台是北京航空航天慧海公司开发的三维图形图像显示平台。代码使用 C++ 和 OpenGL Direct9X 书面从地上爬起来。全球局部范围的各种数据管理，可视化地图平台支持。支持查询，测量，分析，绘图等操作。支持矢量，栅格，影像地图，正射影像阴影，3D 地图显示，可以实现各种数据的混合显示。支持一机三屏或者多屏显示，多机协作，通过网络渲染，并且可以同时让多台机器。平台是基于插件技术的发展，具有高度的可塑性。平台支持二次开发，给用户带来了极大的灵活性。

3.3.1 搭建三维数字球

VVP-3D\ WIN32\3.0\ MFC 目录下创建一个 MFC 单文档的样品
如图 3.21 所示

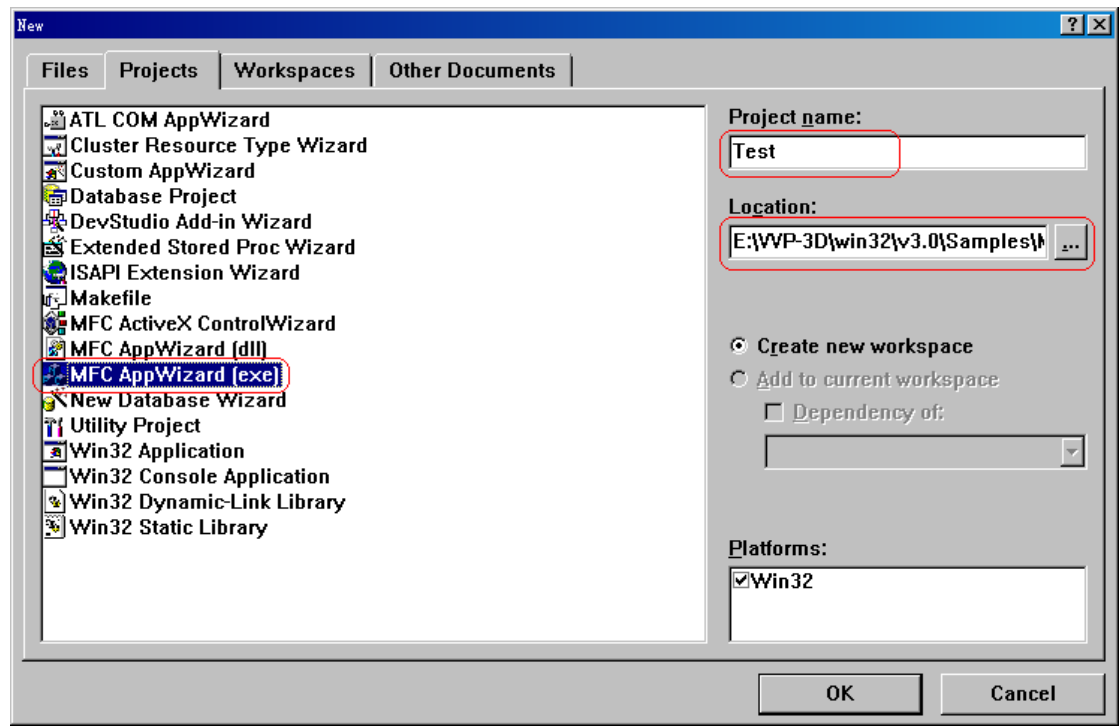


图 3.21 创建 MFC 程序

然后，你需要到配置 VVP-3D\ WIN32\ V3.0\ 调试 \ SysSet\3DMap\ VBF_3DConfig.ini。可以运行 VVP-3D\ WIN32\3.0\调试\ VBF_3DConfigD.exe 配置一个程序的源代码位于 VVP-3D\ WIN32\3.0\样本\ MFC \ VBF_3DConfig VBF_3DConfig.ini 的文件，和在表 3.1 中描述的内容：

表 3.1VBF_3DConfig.ini 文件内容及说明

内 容	说 明
[3DDataPath]	
3DTerrainPath = F:\VVP_Data\demx	地形数据
3DTexturePath = F:\VVP_Data	纹理数据存放在该目录的子目录下
3DNotePath = F:\VVP_Data\notx	注释数据
[3DPlotPath]	
3DMeshPath=F:\TEST\Lib\SysSet\模型	模型（导弹、地面站……）所在目 录

程序执行效果如图 3.22

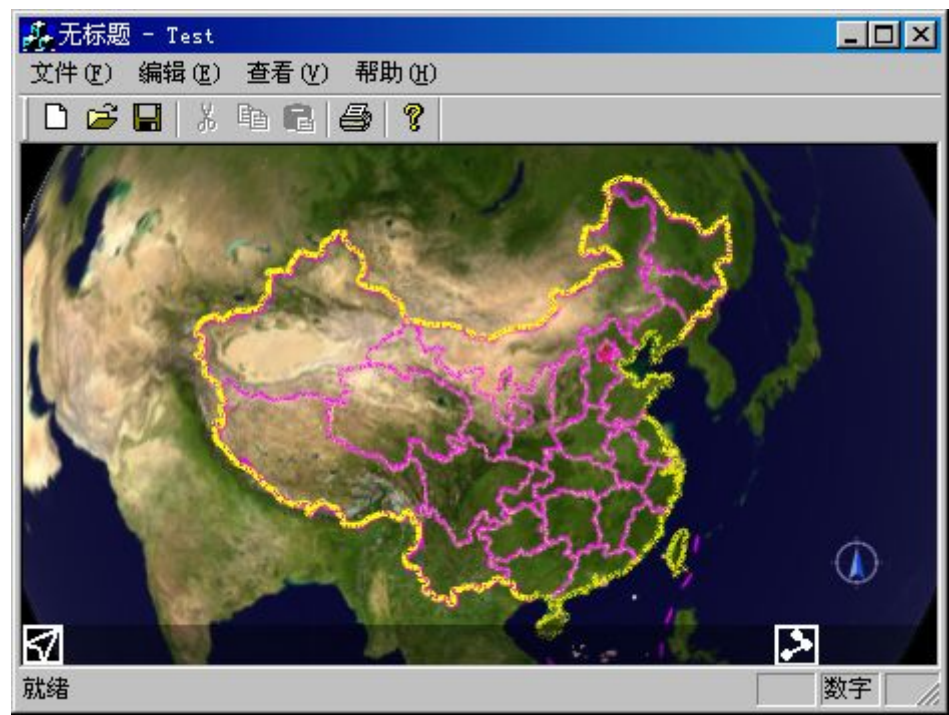


图 3.22 程序运行效果

3.3.2 三维模块装载

VVP-3D 是由多个组件组成。表 3.2 是一些指示的重要组成部分。粗体字地形组件所必需的显示模块。

表 3.2 VVP-3D 组件库及其说明

组 件	说 明
VVP_3DEngine.dll	三维引擎，完成实际的绘制工作
VVP_3DDisplay.dll	完成场景的显示控制
VVP_3DDB.dll	数据库，用于实际的装载数据
VVP_3DDBAgent.dll	数据库代理，用于缓存数据
VVP_3DSky.dll	天空
VVP_3DTerrain.dll	用于地形的 LOD 调度和管理
VVP_3DNote.dll	文字注记
VVP_3DCompass.dll	指北针
VVP_3DRadar.dll	雷达，实现了以矢量的方式绘制的雷达

VVP_3DMesh.dll	模型转载与管理，依赖模型装载配置
VVP_3DAnalysis.dll	量算，实现了坐标、距离、面积、坡度、通视
VVP_3DPlot.dll	标图，实现了栅格模型、3DS 模型等的渲染，依赖 VVP_3DMesh.dll
VVP_3DFlyDll.dll	飞行，依赖 VVP_3DPlot.dll
VVP_3DCoorGridDll.dll	经纬网绘制
VVP_3DPatrolDll.dll	巡逻，依赖 VVP_3DPlot.dll
VVP_3DParticleDll.dll	粒子系统

3.3.3 模型装载模块配置与设置数据路径

加载系统模型，通过插件方式实现，在系统初始化根据 SysSet\3DRegisterMeshLoaders.cfg 的动态加载“模型加载模块”里的动态库文件名。目前，该系统支持两种格式的模型加载。

VVP_OBJLoader.dll
VVP_3DSLoader.dll
VVP_3DXLoader.dll

配置参数文件为：SysSet\VVP_3DTerrainConfig\VVP_3DTerrain.ini，其格式为 [3DDataPath]

3DTerrainPath = F:\VVP_Data\Terrain\DDS
3DTexturePath = F:\VVP_Data\Terrain\DDS
3DNotePath = F:\VVP_Data\Terrain\DDS

其中，

3DTerrainPath：地形几何数据的路径，地形几何数据文件的后缀为.demx
3DTexturePath：地形纹理数据的路径，地形纹理数据文件的后缀为.texx
3DNotePath ：地形注记数据的路径，地形注记数据文件的后缀为.Notx

3.4利用 OpenGL 生成 3D 主动立体显示效果

立体显示的是一个虚拟现实的实现。主要在以下方面：二向色反射镜，活性立体显示，被动立体投影设备同步，三维显示，真正的三维显示，其他更先进的

设备的立体显示的。由于硬件设备的要求我们采取主动立体显示。

3.4.1 3D 主动式立体

在这种模式下，驱动程序将关闭左眼和右眼的图像的绘制，如用于左眼的第一帧图像，然后在下一帧的右眼图像，然后在下一帧重新渲染图像的左眼，交替渲染。观察者将使用快门眼镜。时，通过有线或无线的方式与图形卡和显示器同步，快门眼镜的左眼用图像被显示在显示屏上，打开左眼镜镜片的右透镜快门同时关闭快门，右眼的图像被显示在显示屏上时，打开右眼镜镜片的快门同时关闭左镜片的快门。隐形眼图像只能由大脑根据的屏幕只是一个暂时的保留，只要任何人在此范围内，把我们的眼镜可以观看三维图像的视觉效果。

3.4.2 用 OpenGL 编写立体显示程序

Off-axis 是一种正确的产生立体图像对的方法，它不会产生垂直视差并且产生不紧迫的立体图像对。需要注意的是它的投影是必须产生一个非均匀的相机平截头体，OpenGL 中支持这种投影方式。在这种情况下视觉向量仍然是平行，并且使用 `glFrustum()` 函数去描述透视投影。原理如图 3.23

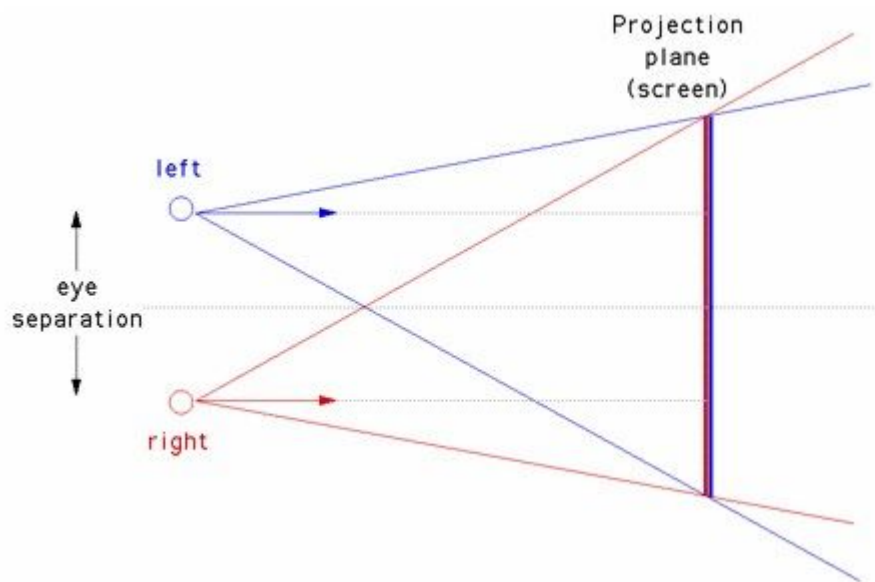


图 3.23 Off-axis 方法图

计算非均匀的视维补偿 D:

```
widthdiv2 = camera.near * tan(camera.aperture/2)
```

$$D = 0.5 * \text{camera.eyesep} * \text{camera.near} / \text{camera.fo}$$

上下左右前后的透视投影值是：

```
top    = widthdiv2;
```

```
bottom = - widthdiv2;
```

```
left   = - aspectratio * widthdiv2 ± 0.5 * camera.eyesep * camera.near  
/ camera.fo;
```

```
right  =  aspectratio * widthdiv2 ± 0.5 * camera.eyesep * camera.near  
/ camera.fo;
```

```
near = camera.near
```

```
far = camera.far
```

相机的摆放如图 3.24:

```
gluLookAt(camera.pos.x + right.x, camera.pos.y + right.y, camera.pos.z  
+ right.z,
```

```
camera.pos.x + right.x + camera.dir.x,
```

```
camera.pos.y + right.y + camera.dir.y,
```

```
camera.pos.z + right.z + camera.dir.z,
```

```
camera.up.x, camera.up.y, camera.up.z);
```

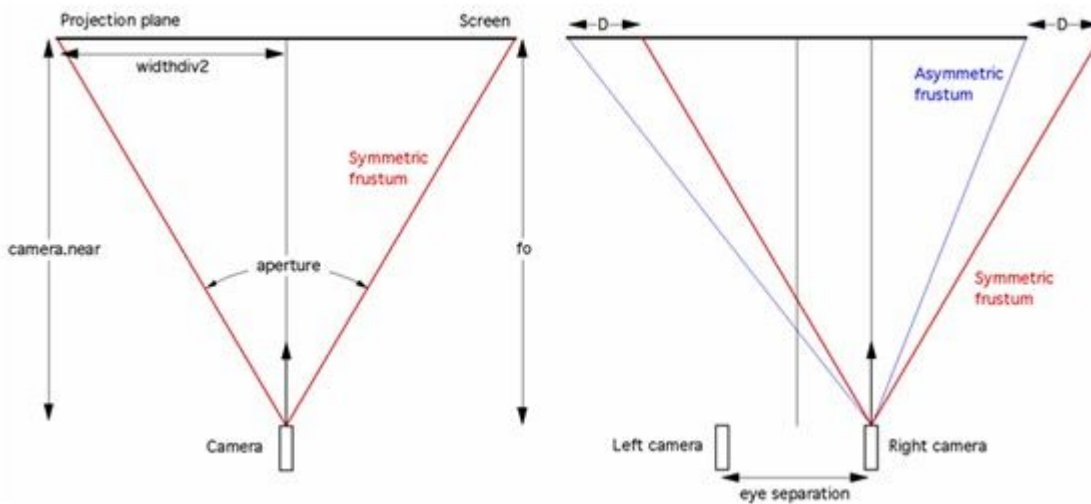


图 3.24 相机摆放位置

效果如图 3.25



图 3.25 主动立体效果图

3.5 参数数据 XML 表

本文使用的 XML 的文件系统模型参数和仿真软件系统的用户配置参数。

3.5.1 XML 的语法

一个简单的 XML 的参数表之后，例如，XML 语法规则的简要说明。
下面以一个简单的 XML 参数表为例，简要说明 XML 的语法规则。

```
<?xml version="1.0" encoding="GB2312"?>
```

```
<雷达参数>
```

```
  <雷达工作状态>
```

```
    <值>false</值>
```

```
    <单位>无</单位>
```

```
    <变量类型>bool</变量类型>
```

```
  </雷达工作状态>
```

```
  <雷达峰值功率 值="1000" 单位="瓦" 变量类型="double"/>
```

```
</雷达参数>
```

XML 文档的语法主要包含以下几个点

1. 文件声明，处理指令和注释

<? XML 版本=“1.0” 编码=“GB2312” ? >是一个文档声明部分。版本用于指示该文件的版本;编码是用来指定在文档中的字符的编码。此外，还有一个可选的属性，可以用来描述是否独立的外部文件。是指没有调用外部文件，没有表示，叫做外部文件。处理指令表示的引进处理 XML 文档的应用程序。随着<? > 标签

2. XML 元素

XML 元素是表示文档文件中包含的结构和数据的 XML 文档的核心部分。XML 文档中的根元素的元素被划分为两种类型的非根元素。

- 根元素：XML 文档有一个且只有一个根元素。在上面的例子中下面的部分的根元素。

<Radar 参数>

</雷达参数>

- 非根元素：包括开始标签，内容和结束标签。中学非根元素可以是嵌套元素，形成的亲子关系。上述的非 XML 文档的根元素的例子包括以下内容：

<Radar 工作 condition></雷达的工作状态>

<VALUE></值>

<Unit></单位>

<variable 类型></变量类型>

<雷达峰值功率..... />。

3. XML 元素的属性

XML 元素可以设置属性，用来代表元素了解更多信息。元素属性和元素来描述数据是不同的。定义的格式属性。

<Start 标签 属性命名="attribute value" 物业名称 2 ="attribute value"...></结束标记>

雷达峰值功率以下信息记录使用的元素属性。

<Radar 峰值功率值="1000"单位="W"变量类型="double"/>

这些 XML 文件的 XML 解析使用一种特殊的软件来打开如下：

| | |
|--------|--------|
| 雷达参数 | |
| 雷达工作状态 | |
| 值 | false |
| 单位 | 无 |
| 变量类型 | bool |
| 雷达峰值功率 | |
| 值 | 1000 |
| 单位 | 瓦 |
| 变量类型 | double |

图 3.26XML 参数表示例

你可以看到，这显然是一个 XML 结构，内容和明确的，所以使用的 XML 解析软件，可以有效地改善开发人员的 XML 技术。

3.5.2 雷达模型仿真参数表

在本文中，所使用的软件雷达模型如图 3.27 所示的仿真参数表。

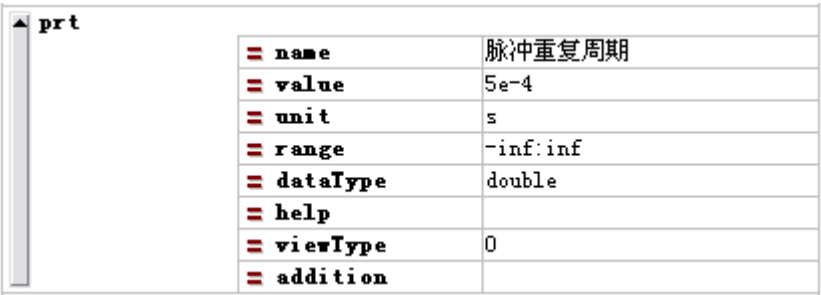
| | | | | | | | | | | | | | | | | | | | | | | | |
|--|---|------------------|--|-------------|---|-----------------|--|------------|---|------------------|---|-------------|---|--------|---|----------|--|--------|---|--------|---|------|---|
| XMLSpy v2010 (http://www.altova.com) by MESMERiZE (MIZE) | | | | | | | | | | | | | | | | | | | | | | | |
| rdParameters | | | | | | | | | | | | | | | | | | | | | | | |
| TransmitterParams | | | | | | | | | | | | | | | | | | | | | | | |
| classInformation | id= fileName= className= name= baseClass= help= author= | | | | | | | | | | | | | | | | | | | | | | |
| interface | | | | | | | | | | | | | | | | | | | | | | | |
| parameters | <table><tr><td>maxPower</td><td>name=峰值功率 value=20000 unit=W range=-inf:inf dataType=...</td></tr><tr><td>pulseWidth</td><td>name=脉冲宽度 value=1e-6 unit=m range=-inf:inf dataType=...</td></tr><tr><td>pri</td><td>name=脉冲重复周期 value=5e-4 unit=s range=-inf:inf dataType=d...</td></tr><tr><td>bandWidth</td><td>name=发射信号带宽 value=5e6 unit=Hz range=-inf:inf dat...</td></tr><tr><td>pulseResidentNum</td><td>name=驻留脉冲数 value=16 unit=个 range=-inf:in...</td></tr><tr><td>fc</td><td>name=中频频率 value=1e9 unit=Hz range=-inf:inf dataType=double...</td></tr><tr><td>fs</td><td>name=采样频率 value=2e7 unit=Hz range=-inf:inf dataType=double...</td></tr><tr><td>fworking</td><td>name=工作频率 value=1e10 unit=Hz range=-inf:inf dataType=...</td></tr></table> | maxPower | name=峰值功率 value=20000 unit=W range=-inf:inf dataType=... | pulseWidth | name=脉冲宽度 value=1e-6 unit=m range=-inf:inf dataType=... | pri | name=脉冲重复周期 value=5e-4 unit=s range=-inf:inf dataType=d... | bandWidth | name=发射信号带宽 value=5e6 unit=Hz range=-inf:inf dat... | pulseResidentNum | name=驻留脉冲数 value=16 unit=个 range=-inf:in... | fc | name=中频频率 value=1e9 unit=Hz range=-inf:inf dataType=double... | fs | name=采样频率 value=2e7 unit=Hz range=-inf:inf dataType=double... | fworking | name=工作频率 value=1e10 unit=Hz range=-inf:inf dataType=... | | | | | | |
| maxPower | name=峰值功率 value=20000 unit=W range=-inf:inf dataType=... | | | | | | | | | | | | | | | | | | | | | | |
| pulseWidth | name=脉冲宽度 value=1e-6 unit=m range=-inf:inf dataType=... | | | | | | | | | | | | | | | | | | | | | | |
| pri | name=脉冲重复周期 value=5e-4 unit=s range=-inf:inf dataType=d... | | | | | | | | | | | | | | | | | | | | | | |
| bandWidth | name=发射信号带宽 value=5e6 unit=Hz range=-inf:inf dat... | | | | | | | | | | | | | | | | | | | | | | |
| pulseResidentNum | name=驻留脉冲数 value=16 unit=个 range=-inf:in... | | | | | | | | | | | | | | | | | | | | | | |
| fc | name=中频频率 value=1e9 unit=Hz range=-inf:inf dataType=double... | | | | | | | | | | | | | | | | | | | | | | |
| fs | name=采样频率 value=2e7 unit=Hz range=-inf:inf dataType=double... | | | | | | | | | | | | | | | | | | | | | | |
| fworking | name=工作频率 value=1e10 unit=Hz range=-inf:inf dataType=... | | | | | | | | | | | | | | | | | | | | | | |
| AntennaParams | | | | | | | | | | | | | | | | | | | | | | | |
| classInformation | id= fileName= className= name= baseClass= help= author= | | | | | | | | | | | | | | | | | | | | | | |
| interface | | | | | | | | | | | | | | | | | | | | | | | |
| parameters | <table><tr><td>aziWidth</td><td>name=方位波束宽度 value=3 unit=" range=-inf:inf dataTy...</td></tr><tr><td>aziGain</td><td>name=方位天线增益 value=1e3 unit= range=-inf:inf dataTyp...</td></tr><tr><td>aziSideLobe</td><td>name=方位旁瓣电平 value=0 unit= range=-inf:inf dataT...</td></tr><tr><td>eleWidth</td><td>name=俯仰波束宽度 value=3 unit=" range=-inf:inf dataTy...</td></tr><tr><td>eleGain</td><td>name=俯仰天线增益 value=1e3 unit= range=-inf:inf dataTyp...</td></tr><tr><td>eleSideLobe</td><td>name=俯仰旁瓣电平 value=0 unit= range=-inf:inf dataT...</td></tr><tr><td>aziMin</td><td>name=方位扫描下限 value=10 unit=" range=-inf:inf dataTyp...</td></tr><tr><td>aziMax</td><td>name=方位扫描上限 value=45 unit=" range=-inf:inf dataTyp...</td></tr><tr><td>eleMin</td><td>name=俯仰扫描下限 value=10 unit=" range=-inf:inf dataTyp...</td></tr><tr><td>eleMax</td><td>name=俯仰扫描上限 value=45 unit=" range=-inf:inf dataTyp...</td></tr><tr><td>Gain</td><td>name=天线增益 value=1e3 unit= range=-inf:inf dataType=double...</td></tr></table> | aziWidth | name=方位波束宽度 value=3 unit=" range=-inf:inf dataTy... | aziGain | name=方位天线增益 value=1e3 unit= range=-inf:inf dataTyp... | aziSideLobe | name=方位旁瓣电平 value=0 unit= range=-inf:inf dataT... | eleWidth | name=俯仰波束宽度 value=3 unit=" range=-inf:inf dataTy... | eleGain | name=俯仰天线增益 value=1e3 unit= range=-inf:inf dataTyp... | eleSideLobe | name=俯仰旁瓣电平 value=0 unit= range=-inf:inf dataT... | aziMin | name=方位扫描下限 value=10 unit=" range=-inf:inf dataTyp... | aziMax | name=方位扫描上限 value=45 unit=" range=-inf:inf dataTyp... | eleMin | name=俯仰扫描下限 value=10 unit=" range=-inf:inf dataTyp... | eleMax | name=俯仰扫描上限 value=45 unit=" range=-inf:inf dataTyp... | Gain | name=天线增益 value=1e3 unit= range=-inf:inf dataType=double... |
| aziWidth | name=方位波束宽度 value=3 unit=" range=-inf:inf dataTy... | | | | | | | | | | | | | | | | | | | | | | |
| aziGain | name=方位天线增益 value=1e3 unit= range=-inf:inf dataTyp... | | | | | | | | | | | | | | | | | | | | | | |
| aziSideLobe | name=方位旁瓣电平 value=0 unit= range=-inf:inf dataT... | | | | | | | | | | | | | | | | | | | | | | |
| eleWidth | name=俯仰波束宽度 value=3 unit=" range=-inf:inf dataTy... | | | | | | | | | | | | | | | | | | | | | | |
| eleGain | name=俯仰天线增益 value=1e3 unit= range=-inf:inf dataTyp... | | | | | | | | | | | | | | | | | | | | | | |
| eleSideLobe | name=俯仰旁瓣电平 value=0 unit= range=-inf:inf dataT... | | | | | | | | | | | | | | | | | | | | | | |
| aziMin | name=方位扫描下限 value=10 unit=" range=-inf:inf dataTyp... | | | | | | | | | | | | | | | | | | | | | | |
| aziMax | name=方位扫描上限 value=45 unit=" range=-inf:inf dataTyp... | | | | | | | | | | | | | | | | | | | | | | |
| eleMin | name=俯仰扫描下限 value=10 unit=" range=-inf:inf dataTyp... | | | | | | | | | | | | | | | | | | | | | | |
| eleMax | name=俯仰扫描上限 value=45 unit=" range=-inf:inf dataTyp... | | | | | | | | | | | | | | | | | | | | | | |
| Gain | name=天线增益 value=1e3 unit= range=-inf:inf dataType=double... | | | | | | | | | | | | | | | | | | | | | | |
| ReceiverParams | | | | | | | | | | | | | | | | | | | | | | | |
| classInformation | id= fileName= className= name= baseClass= help= author= | | | | | | | | | | | | | | | | | | | | | | |
| interface | | | | | | | | | | | | | | | | | | | | | | | |
| parameters | <table><tr><td>noiseCoefficient</td><td>name=噪声系数 value=1 unit= range=-inf:inf dat...</td></tr><tr><td>activeRange</td><td>name=动态范围 value=0 unit= range=-inf:inf dataType=...</td></tr><tr><td>minAvailableSNR</td><td>name=最小可检测信噪比 value=13 unit=dB range=-i...</td></tr><tr><td>systemLoss</td><td>name=系统损耗 value=4 unit=dB range=-inf:inf dataType=...</td></tr></table> | noiseCoefficient | name=噪声系数 value=1 unit= range=-inf:inf dat... | activeRange | name=动态范围 value=0 unit= range=-inf:inf dataType=... | minAvailableSNR | name=最小可检测信噪比 value=13 unit=dB range=-i... | systemLoss | name=系统损耗 value=4 unit=dB range=-inf:inf dataType=... | | | | | | | | | | | | | | |
| noiseCoefficient | name=噪声系数 value=1 unit= range=-inf:inf dat... | | | | | | | | | | | | | | | | | | | | | | |
| activeRange | name=动态范围 value=0 unit= range=-inf:inf dataType=... | | | | | | | | | | | | | | | | | | | | | | |
| minAvailableSNR | name=最小可检测信噪比 value=13 unit=dB range=-i... | | | | | | | | | | | | | | | | | | | | | | |
| systemLoss | name=系统损耗 value=4 unit=dB range=-inf:inf dataType=... | | | | | | | | | | | | | | | | | | | | | | |
| JammerParams | | | | | | | | | | | | | | | | | | | | | | | |
| SignalProcParams | | | | | | | | | | | | | | | | | | | | | | | |
| TargetParams | | | | | | | | | | | | | | | | | | | | | | | |

图 3.27 雷达模型仿真 XML 参数表

的 XML 文档的根节点是 rdParameters 的组件模块的雷达参数，包括发射模块参数，天线模块参数，接收器模块参数，干扰模块参数，信号处理模块的参数和数据处理模块，每一个节点是根节点参数等等。该雷达组件模块节点有三个子节点：classInfomation 节点是该模块的类信息节点仿真内核算法开发人员使用，包括的类 ID，类名，文件名，类名，帮助和其他信息；接口节点与其他模块的模块

接口目前还没有启动，占位符模块节点的参数，每个参数的一个子节点，节点的模块依赖于多少的具体算法。

每个参数节点参数变量命名节点的名称，它是用来存储 XML 元素的属性信息的具体参数，每个参数有 8 个属性，分别是：名，参数名值，参数值，单位，参数的单位，范围，参数范围，数据类型，参数数据类型，参数类型的变量在算法；帮助，参数的帮助信息；viewType，参数类型，如直接输入类型和列表来选择项目类型，addition，占位符。



| | | |
|------------|-----------------|----------|
| prt | name | 脉冲重复周期 |
| | value | 5e-4 |
| | unit | s |
| | range | -inf:inf |
| | dataType | double |
| | help | |
| | viewType | 0 |
| | addition | |

图 3.28 具体雷达参数属性

在一般情况下，模拟需要使用的雷达 80-100 各种参数，使用参数表可以是非常直接的视图和管理如此庞大的数量的参数。树结构也可以表示模块的参数隶属关系，使用更方便用户使用和提升用户体验，使得人机交互更加友好的联系。

3.6仿真数据内存共享

此功能使用共享内存来实现内存共享技术，自定义包装功能的二次开发，打包其他程序调用的动态链接库（DLL）。动态链接库包含一个内存共享类的类用于创建和访问共享内存区域的功能，也有两名重要成员已经建立了一个共享内存区域。

```
LPVOID CreateMemory(char* name="YourName", size_t len=1)
```

```
LPVOID ShareMemory(char* name="");
```

按照成立 CreateMemory 功能共享内存区域的具有指定名称和长度，返回创建的内存指针；ShareMemory 功能已建立了具有指定名称查找共享内存区域，并返回指针。

创建共享内存区域的服务器，创建一个 Windows 窗体应用程序根据表创建服务器配置的内存区域。

| | | |
|-----------------|--------|--------|
| ▲ beam_memory | = name | beam |
| | = len | 1024 |
| ▲ detect_memory | = name | detect |
| | = len | 1024 |
| ▲ object_memory | = name | object |
| | = len | 1024 |
| ▲ radar_memory | = name | radar |
| | = len | 1024 |
| ▲ ctrl_memory | = name | ctrl |
| | = len | 1024 |
| ▲ echo_memory | = name | echo |
| | = len | 1e6 |

图 3.29 内存共享配置表

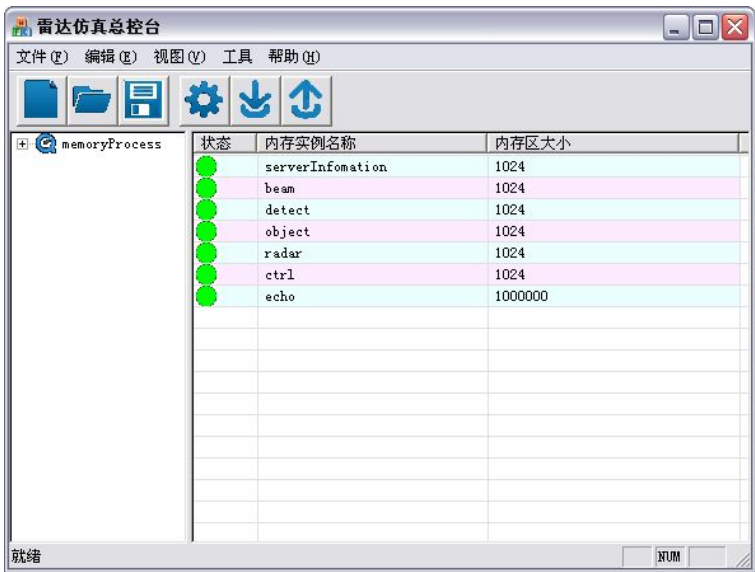


图 3.30 内存共享界面窗体

为预定的通信格式，使用不同类型的数据结构。

```
struct beamParameters
{
    double widthAngle;
    double centerAngle;
    double widthRange;
    double centerRange;
    double elemationAngle;
    double widthElemationAngle;
};

struct detectedParameters
{
    int type;
```

```
    char name[256];
    double range;
    double Azangle;
    double Pangle;
    double velocity;
};

struct ctrlParameters
{
    int modeSwitch;
    int start_flag;
    int stop_flag;
    int expand_flag1;
    int expand_flag2;
};

struct objectParameters
{
    double x;
    double y;
    double z;
    double Vx;
    double Vy;
    double Vz;
    double Ax;
    double Ay;
    double Az;
    double azi;
    double ele;
    double roll;
    double state;
    double life;
    double time;
};
```


3.7UDP 仿真报文

UDP 协议的主要作用是将网络数据流量压缩成数据包的形式。一个典型的数据包是一个二进制的数据传输单元。传输的数据是一个字节数组的数据类型（字节[]），但往往在用户的数据传输类型是不是字节，因此，数据发送方首先需要按照一定的顺序的数据转换成字节，并存储在一个阵列内的，数据接收方的发送方，根据接收到的数据存储阵列中的数据类型转换，为了得到正确的数据传输的数据序列。

在实践中，双方通常会同意传输数据的存储顺序。在本文中，软件报，移动目标的信息包的一部分，例如，双方同意消息内容是：经度，纬度，高度，速度，方向和新的目标是否标志。表 3.3 所示的数据格式，数据长度和数据开始位置。

表 3.3 报文数据格式示例

| 数据内容 | 数据格式 | 数据长度 | 起始位置 |
|-------|------|------|------|
| 经度 | long | 8 | 0 |
| 纬度 | long | 8 | 8 |
| 高程 | int | 4 | 16 |
| 速度 | int | 4 | 20 |
| 方向 | int | 4 | 24 |
| 是否新目标 | byte | 1 | 28 |

在数据传输之前和之后的数据转换工作需要做，本文使用。NET 提供了数据转换类 BitConverter 来完成这个任务，这个类可以完成基本的数据类型和字节数组转换。类的成员函数 BitConverter.GetBytes（）可以被转换成字节或字节数组中的数据类型各种通用的数据类型，子类还提供其他的可以被转换成字节数据类型之外的其他普通类型的数据的功能，如 ToDouble（字节[]值，INT startIndex 开始）的值是一个数组，从指定的起始位置 startIndex 开始转换为 Double 类型的数据，ToChar（byte []的值，INT startIndex 开始）的值是一个数组，从指定的起始位置 startIndex 开始转换为 Char 类型的数据。

.NET 环境也提供了 C# 语言的使用用户数据报（UDP）网络服务类 System.Net.Sockets 命名空间。类的实例，你可以使用它的端口号是变量构造的 UdpClient 特定端口号（INT 端口），提高开发人员的可控性的数据传输。

其成员函数发送（字节[] DGRAM，int 字节的字符串主机名，INT 端口）一个函数发送 UDP 包，四个参数是要发送的字节数组，数组的长度，的传输目的 IP 和端口号。功能的 byte []接收（参考的 IPEndPoint remoteEP 的）是接收 UDP 数据包的成员函数，参数是数据发送方的地址，返回值是一个接收的字节数组，也

就是说，传输的数据。通过定义一个任意地址的 `IPEndPoint` 实例收取任何地址发送 UDP 包来实现下面的代码。

发送方：

```
UdpClient udpClient = new UdpClient(50000);  
udpClient.Send(sendBytes, sendBytes.Length, 127.0.0.1, 50001);
```

接收方：

```
UdpClient receivingUdpClient = new UdpClient(50001);  
IPEndPoint RemoteIpEndPoint = new IPEndPoint(IPAddress.Any, 0);  
Byte[] receiveBytes = receivingUdpClient.Receive(ref RemoteIpEndPoint);
```

其中，发送方为 Windows 默认的本地 IP 地址 127.0.0.1，端口号是 50000，接收方的端口号是 50001，因此目标地址是 127.0.0.1:50001。

3.8本章小结

本章提供了一个开放的可视雷达仿真系统开发方法进行了研究。首先，通过分析 c# 语言环境的 Windows 窗体应用程序的开发过程中，一个清晰的可视雷达仿真系统是一个关键的技术领域，其次是引入 MapX 控件的编程环境和使用，尤其是雷达仿真系统特定的模拟单元图标和目标轨道的绘图功能实现。接着，XML 语法的基础上，并给出了雷达模型的仿真参数表的设计。最后，内存共享技术和的 UDP 技术开发的方法和关键代码。

第四章 功能级仿真雷达模型建模

雷达性能功能仿真是一种有效的手段来评估，因为它不是基于模拟信号流等等功能的仿真速度比相干视频模拟快得多，功能仿真的核心是利用雷达方程，雷达环境下的应用信号功率和噪声或杂波功率，并计算其信噪比计算功率的雷达范围内，检测概率，测量精度。

4.1 功能级仿真的流程

在功能模拟系统，没有回波的模拟信号的处理，但只计算的回声信号功率与噪声或干扰比，然后根据检测概率曲线的检测概率，相比，随机数检查目标被检测到，测量值的最终目标或误报和假目标。功能级别的雷达仿真算法流程如下图 4.1 所示：

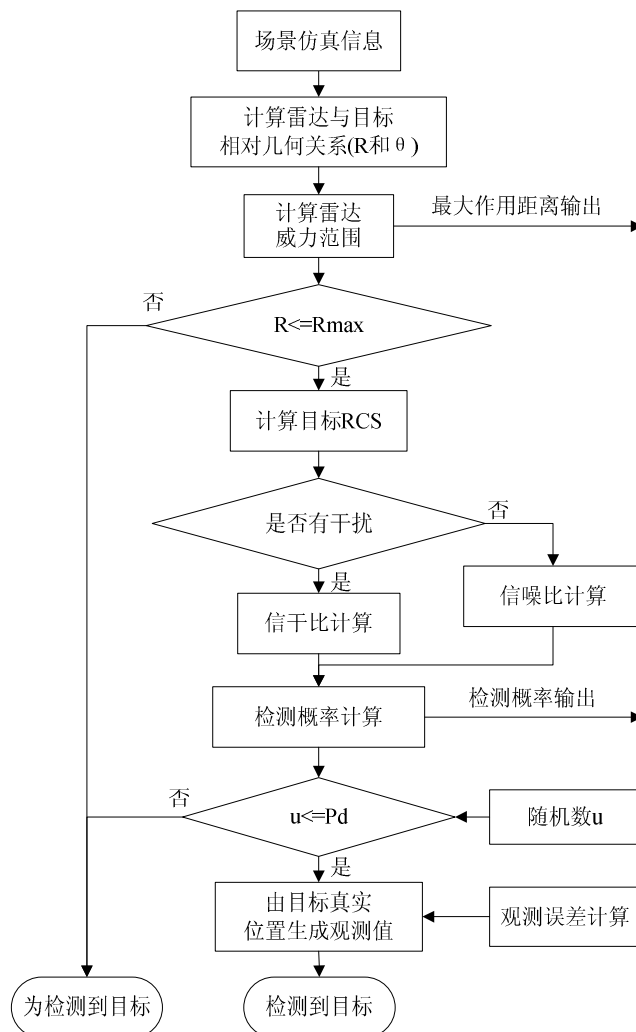


图 4.1 功能级仿真流程图

功能级功能仿真过程是通过从控制接收器，发射器，目标运动平台，获取相关信息计算天线模块信噪比 SNR 计算根据当前目标的探测概率的位置。雷达实际工作中外部环境的变化，因为随机的，即使目标的雷达范围的范围，有可能不被检测，以便检测由雷达目标的概率大于一定的阈值时，仅发现的目标，考虑到实际情况，阈值的选择是不是一个固定值，因此在治疗过程中使用的随机数，生成一个随机区间[0,1]均匀分布的随机数，以下方式获得的检测概率电流值比较，当该随机数小于或等于目标被检测到的检测概率，输出检测到的目标信息，并输出目标的测量值信息，当随机数为大于的概率检测没有检测到目标，即产生漏警，输出的探测信息为零信息。

4.2 目标的 RCS

根据目标相对于雷达分辨单元的大小，可以分为点目标和扩展目标的目标。点目标是指实际尺寸相比是非常小的雷达分辨单元的一类对象，点目标后向散射雷达截面为一般用途（RadarCrossSection - RCS）的特点，和雷达截面的特点目标本身，国外有关，也与目标的态度，雷达波长和偏振。因此，当目标处于运动状态，或的雷达波长或偏振变化，目标的雷达散射截面将跟随起伏。

如果目标是大于雷达分辨单元，用于延长的目标的目标，那么，不仅是目标的雷达横截面面积的完整的描述，但还需要表征的空间特性，将目标。简单的办法是延长几个不干涉目标散射中心是分开识别组件，每个散射中心作为一个孤立的点目标为蓝本。

现在常用的模型是点目标 Swerling（斯威林）模型，Swerling 模型统计分布的 RCS 和相关功能的五个标准的统计模型。通常情况下，Swerling 0 被称为非波动目标，Swerling 1 模型到 Swerling4 模型，包括两个不同的概率密度函数，同时有两种不同的有关情况，其中一个是回波的天线扫描期间的波动是完全相关的，而并扫描之间是完全不相干的，即所谓的缓慢起伏目标，另一种是快速变化的目标，他们的起伏在脉冲回波之间是完全不相干的。四种起伏模型区分如下：

a) Swerling 1 型，瑞利分布，慢起伏。其截面积 σ 的概率密度函数为以下分布：

$$p(\sigma) = \frac{1}{\bar{\sigma}} e^{-\frac{\sigma}{\bar{\sigma}}}, \sigma \geq 0 \quad (4-1)$$

$\bar{\sigma}$ 为目标起伏全过程的平均值，所表示的指数函数分布的横截面面积，目标的横截面面积的公式是成比例的回波的功率，回波振幅瑞利分布。

Swerling 2 型，瑞利分布，快起伏。目标截面积的概率分布与 Swerling 1

相同，但为快起伏。假定脉冲与脉冲之间的起伏是统计独立的。

b) Swerling 3 型，慢起伏。其截面积 σ 的概率密度函数为

$$p(\sigma) = \frac{4\sigma}{\sigma^2} e^{-\frac{2\sigma}{\sigma}} \quad (4-2)$$

这类截面积起伏所对应的回波振幅满足主加瑞利分布。

c) Swerling 4 型，快起伏，截面积的概率分布同 3 型。

首先，对于复杂的情况下的概率分布的两种类型的横截面面积是约等于由大量的散射体（如飞机）的靶细胞。三，四个横截面面积的情况下的概率分布为目标反射体，并且具有较大数目的小反射镜的合成，或大型的反射镜在一个小的变化的取向。

4.3 无干扰条件下信噪比的计算

作为主要的功能仿真雷达理论的是雷达方程。根据雷达距离方程，斜距为 R 的距离，目标再次被反射的雷达回波信号的接收功率：

$$P_R = \frac{P_t G_t G_r \lambda^2 \sigma_t}{(4\pi)^3 R_t^4 L_s} \quad (4-3)$$

其中， λ 为载波波长， σ_t 是目标的 RCS， R_t 是目标到雷达之间的距离， L_s 为系统损耗， P_t 为发射机峰值功率， G_t 为雷达发射天线的增益， G_r 是接收天线的增益。

考虑脉冲压缩比和相干积分增益脉冲积累，无干扰，在前面的雷达系统检测信噪比：

$$\frac{S}{N} = \frac{P_t G_t G_r \lambda^2 \sigma_t N_p D_0}{(4\pi)^3 R_t^4 K T_s B F L_s} \quad (4-4)$$

其中， D_0 为脉压比（其值等于脉冲宽度和采样率的乘积）， K 为波尔兹曼常数（ $K = 1.38054 \times 10^{-23} J/K$ ）， T 为电阻温度（ $T = 290K$ ）， B 为带宽， N_p 为积累脉冲数， F 为噪声系数， $K T_s B F$ 合起来为接收机噪声功率。

4.4 有干扰条件下信干比的计算

干扰机释放干扰时，干扰信号进入雷达接收天线的增益为 G_{rj} ，干扰发射功率为 P_j ，干扰机发射天线增益为 G_j ，干扰机距离雷达的距离为 R_j ，则雷达接收到的干扰功率为：

$$S_j = \frac{P_j G_j G_{rj}}{(4\pi)^2 R_j^2} \quad (4-5)$$

在考虑相参积累、脉冲压缩和抗干扰改善因子之后，雷达接收机接受到的信干比为：

$$\frac{S_i}{S_j} = \frac{P G_i G_r \lambda^2 \sigma_i}{(4\pi)^3 R_i^4 L_s} \cdot \frac{(4\pi)^2 R_j^2}{P_j G_j G_{rj}} \cdot N_p D_0 I_j \quad (4-6)$$

4.5 雷达检测概率分析及计算

标准已经知道后，可以得到一个理论上的雷达探测概率公式。让接收到的雷达信号和高斯白噪声，信号的幅值 A 的一个正弦信号，滤波器的中心频率的 IF 信号的频率，由检测器输出的信号的概率密度函数之后：

$$P_d(r) = \frac{r}{\sigma^2} \exp\left(-\frac{r^2 + A^2}{2\sigma^2}\right) I_0\left(\frac{rA}{\sigma^2}\right) \quad (4-7)$$

其中 r 为信号加了噪声之后通过检波器后所得到的输出。 I_0 为第 0 阶的修正贝塞尔函数，它的定义为

$$I_0(x) = \sum_{n=0}^{\infty} \frac{x^{2n}}{2^{2n} \cdot n! n!} \quad (4-8)$$

积分表示为：

$$I_0(x) = \frac{1}{\pi} \int_0^{\pi} e^{-x \cos \theta} d\theta \quad (4-9)$$

上述式表示的被称为广义瑞利分布的概率密度函数。找到目标的概率超过检测阈值的概率，对雷达的检测概率

$$P_D = \int_{r_T}^{\infty} P_D(r) dr = \int_{r_T}^{\infty} \frac{r}{\sigma^2} \exp\left(-\frac{r^2 + A^2}{2\sigma^2}\right) I_0\left(\frac{rA}{\sigma^2}\right) dr \quad (4-10)$$

在上式中

$$\frac{A}{\sigma} = \left(\frac{2S}{N}\right)^{\frac{1}{2}} = (2 \cdot SNR)^{\frac{1}{2}} \quad (4-11)$$

也可用 $\sigma(2 \cdot SNR)^{\frac{1}{2}}$ 代替 A 来计算雷达的检测概率。当雷达用的脉冲信号或其他信号，雷达的检测概率公式仍然可以由上述式表示。

从上面的公式可以计算的虚警概率，由此可以得出在一个雷达系统的干扰或不干扰的条件下，最大的检测范围内检测到某些因素的概率必须找到。

4.6 探测信息的误差计算

目标检测模块的输出目标的检测信息的发现目标的距离，速度，方位和仰角信息，具体的实施的目标的数据的一些纠错的理论，添加过程中的实用应用程序错误生成的条目应用程序的过程，以使其产生的数据更真实的。

1. 距离误差

雷达数据的测量误差通常是由多个方面，距离引起的测量误差的主要因素的是 SNR，采样频率，量化晶体的稳定性，系统定时和由大气折射引起的误差的距离。

1) 信噪比波动引起的误差

由于随机噪声，由噪声引起的测量误差将不为零，雷达设计，信号的信噪比由轧制引起的测距误差可以使用下列公式计算：

$$\sigma_{R_n} = \frac{c}{2B\sqrt{2S/N}} \quad (4-12)$$

式中， B 为雷达信号带宽， c 为自由空间光速， S/N 为所测距离的平均信噪比，由噪声引起的误差是雷达的极限误差。

1) 采样频率和距离量化引起的误差

接收到雷达信号，需要进行数字化处理，根据目标采样频率提取的信号振幅值，这个过程称为采样，采样频率误差为：

$$\sigma_{R_{fs}} = \frac{c}{2\sqrt{12}fs} \quad (4-13)$$

量化装置，从最大范围的雷达，用于表示一个二进制数，二进制“1”的比特数表示的量化单元，由脉冲宽度表示的距离大致相等的距离。量化误差引起的距离可以计算公式为：

$$\sigma_{R_q} = \frac{cR_{\max}}{2\sqrt{12} \times 2^n} \quad (4-14)$$

或

$$\sigma_{R_q} = \frac{c\tau}{2\sqrt{12}} \quad (4-15)$$

式中， n 为二进制位数， τ 为采样间隔， R_{\max} 为最大作用距离。在工程应用中，根据第二类型直接计算。然而，因为距离通常是大于或等于给量化单元的采样间隔，从而计算距离误差在分析中只考虑距离的量化误差时，可不再考虑采样频率误差。

2) 接收机延时引起的误差

传统接收机的延时为

$$\Delta t_d = \frac{\sqrt{m}}{B} \quad (4-16)$$

式中, m 是放大器级数, B 是接收机带宽。一般地取 $\Delta t_d = 3/B$, 但这是个固定的延时, 其补偿和校正后的残差作为随机误差

$$\sigma_{R_r} = \frac{c}{30B} \quad (4-17)$$

综上所述, 若认为距离误差的各因素是等权系统, 则距离总误差可表示为:

$$\sigma_R = \sqrt{\sigma_{R_n}^2 + \sigma_{R_r}^2 + \sigma_{R_q}^2} \quad (4-18)$$

2. 角度误差

相控阵雷达接收系统的结构和方法测量角度的相控阵雷达, 因为你要跟踪多个目标, 数据传输速率的要求, 在一个方向或跟踪搜索雷达波束的方向, 停留时间是很短, 只有少数的重复周期, 它通常是用于单脉冲方法。单脉冲主要有三种方法: 幅度比较法, 单相和差分脉冲方法, 振幅比较差单脉冲方法。如果没有有效的方法, 单脉冲测角误差会存在。

1) 回波信号信噪比 S/N 能够引起的测角极限出现误差

$$\sigma_n = \frac{K_m}{\sqrt{(S/N)N}} \theta_{1/2} \quad (4-19)$$

单脉冲测角时, $K_m = 0.45$, 比幅度测量的角度, 高斯型半功率点相交, $\theta_{1/2}$ 指半功率波束宽度, N 为天线半功率波束宽度内的脉冲数, 在波束最大点 $K_m = 1.04$, 在波束交点 $K_m = 0.51$, K_m 是系数。

2) 目标闪烁引起的测角误差

目标在探测方位向的尺寸 L_x 与目标闪烁引起的测角误差和雷达的探测距离 R 有关, 可以表示为

$$\sigma_c = 0.35 \frac{L_x}{R} \quad (4-20)$$

一般地, σ_c 小于 1mrad 。

3) 目标位置误差

间隔的一系列脉冲的脉冲重复周期, 在此期间中的目标被认为是相同的位置, 这个间隔所代表的方位值时 $\theta_{1/2}/N$, 假设目标位置在此间隔内是均匀分布的, 其造成的误差为:

$$\sigma_D = \frac{\theta_{1/2}}{N\sqrt{12}} \quad (4-21)$$

其中, $\theta_{1/2}$ 指半功率波束宽度, N 为天线半功率波束宽度内的脉冲数。

一般情况下, 所需的最大信号噪声比的噪声检测的角度测量误差引起的天线的波束角的测量精度本身有密切的关系的最终宽度成反比。的总的角误差, 近似为:

$$\sigma_\theta = \sqrt{\sigma_n^2 + \sigma_D^2 + \sigma_C^2} \quad (4-22)$$

3. 速度误差

通过测量多普勒频率来计算目标相对于雷达的径向速度, 它的测量误差主要和雷达系统的发射频率 f_0 , 脉宽 T_f 有关。

$$\sigma_v = \frac{K_R c}{2T_f f_0} \quad (4-23)$$

式中, f_0 发射频率 (Hz); K_R 为分辨因子, 一般取 1~2 之间的数; T_f 照射目标的时间或脉冲有效宽度或脉冲串有效长度 (s)。

准确性的速度误差的来源, 如量化, 脉冲抖动, 多径和大气折射闪烁, 它们的影响是非常小的, 一般远小于 1, 从而使总的测量误差引起的多普勒径向速度。

4.7 雷达仿真的实现

根据面向对象的编程思想, 功能级别雷达模拟雷达, 目标等仿真装置的包装类, 关键类别的定义进行了介绍。

4.7.1 雷达类

雷达类是功能级别雷达仿真功能的实现类, 预先定义的发射机类, 天线类型, 接收器类型, 光束控制类, 一流的信号处理, 数据处理类和运动信息类的形式作为其成员变量实例化。与此同时, 该类有四个成员函数, 分别实现类的实例化参数的读取, 添加目标检测仿真和测量误差函数。

```
public class xdRadar
{
    public RadarWorkingMode workingMode;
    public rdTransmitter transmitter;
    public rdAntenna antenna;
    public rdReceiver receiver;
```

```

public rdBeamControl beamControl;
public rdSignalProc signalProc;
public rdDataProc dataProc;
public ObjectBasic movingInfo;

public xdRadar(){...}
public void ReadParamsFromXML(string path) {...}
public bool DetectTarget(xdTarget curTarget, bool jamFlag) {...}
public Point AddError(Point realPosition){...}
}

```

以下对各成员函数做一说明：

1. 构造函数

xdRadar()，此功能需要系统使用实例化类的构造函数时，内部的初始化操作，包括雷达，包括默认的参数分配，工作状态，复位等。

2. 参数读取函数

void ReadParamsFromXML(string path)，XML 参数表的路径作为参数，并返回没有任何价值。此功能用于指定位置的雷达仿真从 XML 参数表中读取和雷达参数分配类的成员变量，包括雷达功率，天线增益等雷达子模块的所有参数。

3. 目标探测函数

bool DetectTarget(xdTarget curTarget, bool jamFlag)，待目标实例和干扰检测标志的功能参数，并返回一个 bool 值探测结果表明，true 为检测到，false 为未检测到。这个函数首先以计算目标被检测到的目标实例，如果小于雷达范围内的功率，计算出目标 RCS，选择的干扰标志或信号干扰比 SNR 的计算，最终得到检测的概率和该检测结果，并且将结果返回 bool 值。

4. 增加误差函数

Point AddError(Point realPosition)，检测到真正的目标位置的参数，返回值是增加目标检测后的位置误差。如果目标检测函数返回真，即探测到目标，然后调用该函数在目标真实位置，雷达探测结果的基础上增加的错误。

4.7.2 位置类

位置类作为功能级雷达仿真中各仿真单元运动信息的基类，包含三套雷达位

置坐标，分别是经纬高坐标系、雷达直角坐标系和雷达极坐标系。并提供了一个坐标系转换功能。

```
public class Point
{
    public int index;
    public double time;
    public double longitude;
    public double latitude;
    public double height;
    public double xInRadarCoord;
    public double yInRadarCoord;
    public double zInRadarCoord;
    public double rangeInRadarCoord;
    public double azimuthInRadarCoord;
    public double elevationInRadarCoord;

    public Point(){...}
    public void LonLatH2Radar(Point radarPostioin, double scale) {...}
    public void Radar2LonLatH(Point radarPostioin, double scale) {...}
    public void RightAngle2Polar(){...}
    public void Polar2RightAngle(){...}
}
```

实际使用中，在使用不同的坐标的情况下，需要使用不同的坐标位置。对于这样的对象的例子，没有很多的转换工作。只需调用该对象的成员函数的对象，可以实现坐标转换，然后使用对象的成员变量对应的坐标。

4.7.3 集合 CollectionBase

在雷达仿真系统中，一般包含多个靶细胞或干扰单元，但其数量是在用户指定一个开发好的系统，可以在编程时无法确定，这给编写代码不工作是一个很大的不确定性。在这种情况下，一个通用的解决方案的基础上的仿真程序的实际需要，目标的数量或干扰的上限值设置，定义目标数组的长度的上限，或干扰与类。尽管此方法可用于解决一个暂时性的问题，但有明显的限制，限制了可扩展性雷达仿真系统不能满足打开雷达仿真系统。

在 C# 中的集合类可以是一个很好的解决了这个问题。集合类一般用于处理的对象列表，它的功能更不是一个简单的数组，这些功能的实现 System.Collections 命名空间中获得的接口。收集功能，可通过该接口实现，接口不仅没有限制我们使用基本的集合类，如 System.Array 中。相反，我们还可以创建自己的自定义集合类，如目标类和干扰类。

一种可能的实现是从 `System.Collections.CollectionBase` 集合类派生自己的类。您可以使用这个抽象类的成员变量和成员函数实现集合成员动态变化。

上面介绍的一类基于自定义集合类下面的代码的位置。这个定义包括删除（点 `oldPoint`）的成员函数 `public void` 添加（点 `newPoint`）和公共无效的动态变化，并添加类似于数组索引访问方法操作符。

```
public class Points : CollectionBase
{
    public Points(){}
    public void Add(Point newPoint)
    {
        List.Add(newPoint);
    }
    public void Remove(Point oldPoint)
    {
        List.Remove(oldPoint);
    }
    public Point this[int pointIndex]
    {
        get
        {
            return (Point)List[pointIndex];
        }
        set
        {
            List[pointIndex] = value;
        }
    }
}
```

4.8 本章小结

本章雷达功能仿真方法进行了研究。首先确定功能仿真算法流程，并给出了目标 **RCS** 分类和算法。随后，在一定条件下无干扰噪声比和干扰条件 **SJR** 的计算方法来分析和计算概率的检测，以确定雷达的探测结果。和真实的信息将被添加在目标测量误差的检测结果更符合实际情况。最后，仿真算法是关键代码的一部

分。

第五章 三维可视化雷达仿真系统应用实例

基于早期的研究打开可视雷达仿真系统的方法，笔者的雷达模拟实验室的研究背景，开发了一套系统的实例。下面将详细的介绍其过程，演示结果和分析。

5.1软件的使用

5.1.1 雷达仿真集中控制平台

在雷达仿真系统的基础上，基于组件的设计，开发了具有一定的通用性集控雷达仿真平台，用于集中控制和管理各个组件。



图 5.1 雷达仿真集控平台

图 5.1 显示的雷达仿真集中控制平台，该平台放置了系统的各组成部分的图标，可以实现通过点击图标来调用各种系统组件。从左至右：系统建模，参数设置，场景设置，仿真控制，显示和控制终端，性能评估以及数据回访。

该软件界面上的图标信息通过 XML 文件进行存储，随着用户鄂修改其数据将自动更新。

| | |
|-------------|------------------|
| root | |
| modeling | |
| name | 系统建模 |
| path | simEngine.exe |
| image | 0 |
| author | |
| type | |
| description | |
| size | |
| createtime | |
| changetime | |
| parameters | |
| name | 参数设置 |
| path | ParameterSet.exe |
| image | 2 |
| author | |
| type | |
| description | |
| size | |
| createtime | |
| changetime | |

图 5.2 集控平台配置表

5.1.2 雷达系统的建模组件

此组件预计将发展成一个类似可视雷达和 Simulink 建模组件，通过拖动和连接结构雷达模块模型。

目前，第四章为理论基础，采用 C# 实现代码编写的雷达模型建模知识。创建“RadarSimulation”命名空间命名空间作为仿真内核，内部类注册的“rdTransmitter”发射器类，类“rdAntenna”作为天线类，类的“rdReceiver”光束控制类接收器类，类“rdBeamControl”，类的“rdSignalProc”信号处理类，类的“rdDataProc”作为数据处理类，类的“rdJammer”类别以及目标类的干扰类的“rdTarget”。下课后，发射器，天线类型，接收器类型，光束控制类，信号处理和数据处理类类封装组成的“xdRadar”类别的，因为程序使用模拟类。考虑到场景设置，目标数和干扰可以改变，C# 语言“CollectionBase 的”收集技术的引进。使用“CollectionBase 的”技术来定位类和干扰类打包为的“xdTargets”类和“xdJammer”的类，实现了用户设置的场景可以定制其个数。

使用“Timer”控制“Tick”事件控制的仿真速度，“作为一个仿真步时间间隔”属性。在用户发出命令启动仿真，雷达仿真模型，模拟操作，直到用户发出停止命令。

5.1.3 雷达参数设置及其组件

雷达模型建立后，需要每个模块的参数设置和修改雷达正常工作。在启动平台的接口的参数设置通过点击图标，就可以打开雷达参数设置软件，如图 5.3 所示。

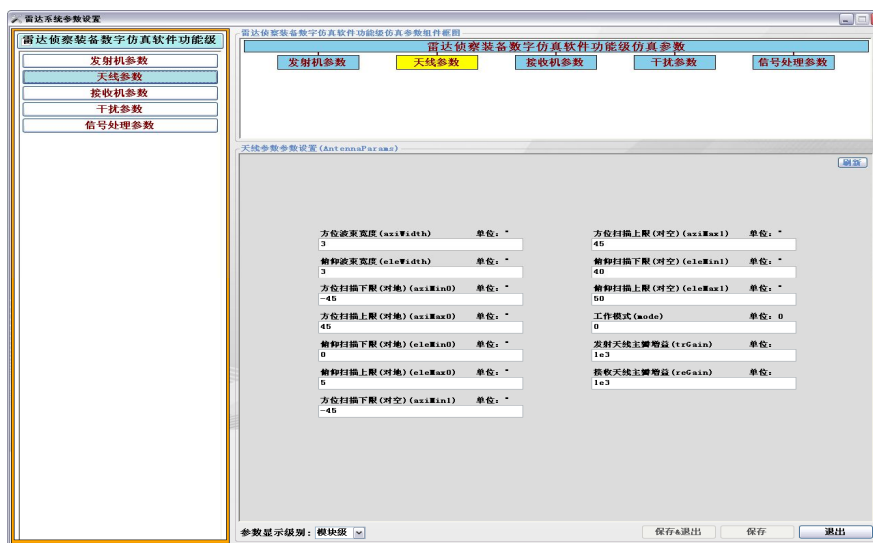


图 5.3 雷达参数设置界面

参数数据通过前文研究过的 XML 参数表在雷达建模组件和参数设置组件之

间进行传输。

| | |
|-------------------|---|
| rdParameters | |
| TransmitterParams | |
| classInformation | id= fileName= className= name= baseClass= help= author= |
| interface | |
| parameters | <div><div>maxPower</div><div>name=峰值功率 value=20000 unit=W range=-inf:inf dat...</div></div> <div><div>pulseWidth</div><div>name=脉冲宽度 value=1e-6 unit=m range=-inf:inf d...</div></div> <div><div>pri</div><div>name=脉冲重复周期 value=5e-4 unit=s range=-inf:inf dataT...</div></div> <div><div>bandWidth</div><div>name=发射信号带宽 value=5e6 unit=Hz range=-inf:inf...</div></div> <div><div>pulseResidentNum</div><div>name=驻留脉冲数 value=16 unit=个 range=-i...</div></div> <div><div>fc</div><div>name=中频频率 value=1e9 unit=Hz range=-inf:inf dataType=do...</div></div> <div><div>fs</div><div>name=采样频率 value=2e7 unit=Hz range=-inf:inf dataType=do...</div></div> <div><div>fworking</div><div>name=工作频率 value=1e10 unit=Hz range=-inf:inf dat...</div></div> |
| AntennaParams | |
| classInformation | id= fileName= className= name= baseClass= help= author= |
| interface | |
| parameters | <div><div>aziWidth</div><div>name=方位波束宽度 value=3 unit=" range=-inf:inf da...</div></div> <div><div>eleWidth</div><div>name=俯仰波束宽度 value=3 unit=" range=-inf:inf da...</div></div> <div><div>aziMin0</div><div>name=方位扫描下限(对地) value=45 unit=" range=-inf...</div></div> <div><div>aziMax0</div><div>name=方位扫描上限(对地) value=45 unit=" range=-inf...</div></div> <div><div>eleMin0</div><div>name=俯仰扫描下限(对地) value=0 unit=" range=-inf:inf...</div></div> <div><div>eleMax0</div><div>name=俯仰扫描上限(对地) value=5 unit=" range=-inf:inf...</div></div> <div><div>aziMin1</div><div>name=方位扫描下限(对空) value=45 unit=" range=-inf...</div></div> <div><div>aziMax1</div><div>name=方位扫描上限(对空) value=45 unit=" range=-inf...</div></div> <div><div>eleMin1</div><div>name=俯仰扫描下限(对空) value=40 unit=" range=-inf...</div></div> <div><div>eleMax1</div><div>name=俯仰扫描上限(对空) value=50 unit=" range=-inf...</div></div> <div><div>mode</div><div>name=工作模式 value=0 unit=0 range=-inf:inf dataType=in...</div></div> <div><div>trGain</div><div>name=发射天线主瓣增益 value=1e3 unit= range=-inf:inf...</div></div> <div><div>reGain</div><div>name=接收天线主瓣增益 value=1e3 unit= range=-inf:inf...</div></div> |
| ReceiverParams | |
| classInformation | id= fileName= className= name= baseClass= help= author= |
| interface | |
| parameters | <div><div>noiseCoefficient</div><div>name=噪声系数 value=1 unit= range=-inf:inf...</div></div> <div><div>minAvailableSWR</div><div>name=最小可检测信噪比 value=13.1 unit=dB r...</div></div> <div><div>systemLoss</div><div>name=系统损耗 value=4 unit=dB range=-inf:inf dat...</div></div> |
| JammerParams | |
| SignalProcParams | |

图 5.4 仿真示例参数 XML 表

5.1.4 雷达场景设置组件

平台在启动时的场景设置界面，单击该图标打开场景设置软件。仿真软件负责设置现场将通过场景和场景设置信息的 XML 表传递给仿真内核的。它的界面分为四个部分：菜单栏，工具栏，左侧场景单元树状列表以及地图部分。

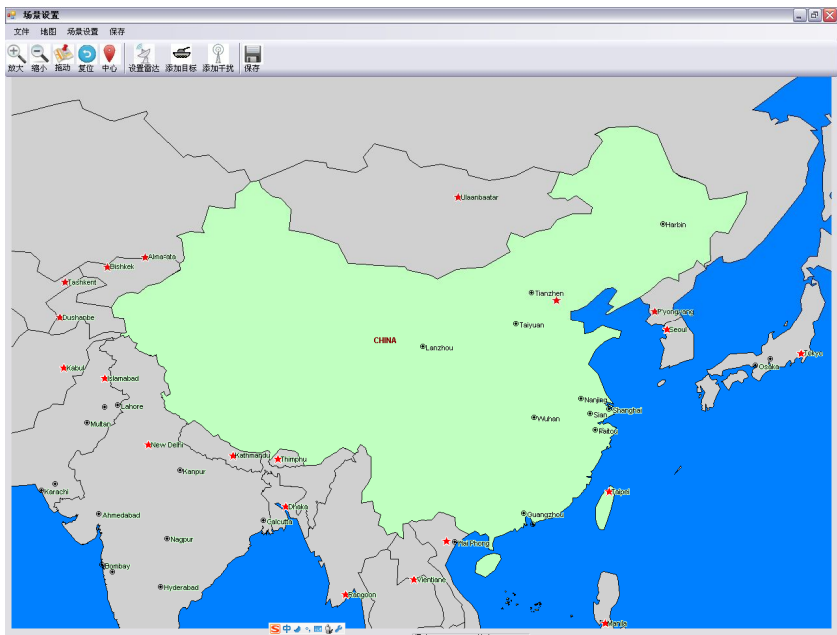


图 5.5 场景设置界面

5.1.4.1 地图

地图 MapX 控件的部分由经纬度和浮动文本框组成的文本框跟随鼠标指针移动，并实时显示当前的纬度和经度坐标。坐标显示如图 5.6 所示的坐标浮动的文本框。



图 5.6 场景设置界面地图部分

5.1.4.2 场景信息树状列表

用户添加设置场景信息时，系统将在仿真单元树状列表中实时添加仿真单元的设置经纬度信息，如图 5.7。

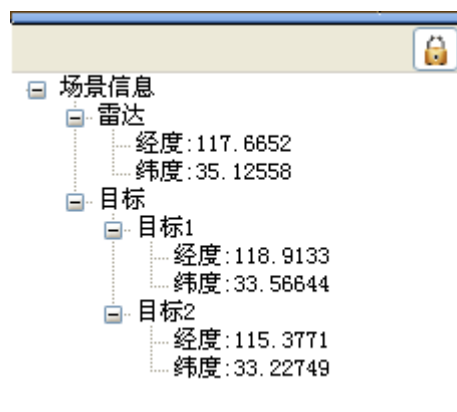


图 5.7 场景信息树状视图

此列表会自动隐藏在软件中打开，当用户将鼠标悬停时会出现一个窗口，在左侧列表中附近时，用户可以选择右上角的列表框中，点击锁图标，从而使其不再自动隐藏。

5.1.4.3 菜单栏和工具栏

菜单栏包含所有组件菜单提供的功能，包括四个菜单，包括文件，地图，场景设置和保存。根据“文件”菜单中的菜单项退出地图地图操作菜单包括五个菜单项，即，放大，缩小，拖动，重设，中心显示功能;场景设置菜单中有两个菜单项添加雷达和目标函数;保存所有的用户设置“菜单项，仿真的场景。场景信息将被写入预定一个很好的模拟控制组件 XML 参数表中。

放在工具栏上的常用工具，



图 5.8 场景设置工具栏

从左到右依次为 5 种地图工具（放大，缩小，拖动，复位以及中心显示），两种场景设置工具（设置雷达和添加目标），最后是场景保存按钮。

5.1.4.4 设置雷达

单击设置雷达之后，鼠标会变为下图的形状，

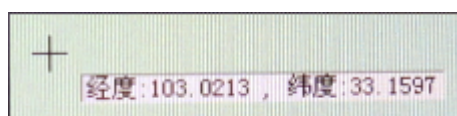


图 5.9 “设置雷达”光标

在地图上需要放置雷达位置处单击鼠标，则会出现雷达详细参数的对话框，



图 5.10 雷达详细参数对话框

该对话框显示用户只需点击的确切的经纬度位置，用户可以调整数值，同时可以设置雷达的高度值、经纬高参数值。单击“确定”，然后在地图上添加成功雷达。

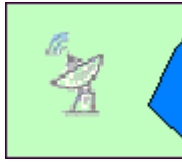


图 5.11 成功添加雷达

5.1.4.5 添加目标和干扰

选择添加目标或添加雷达之后，鼠标在地图区域会变为下图的样子，

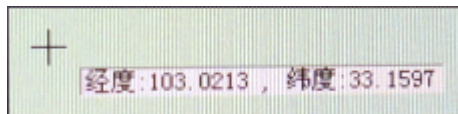


图 5.12 “添加目标” 光标

在地图上需要添加目标的位置处单击鼠标，则会出现详细参数对话框，

| 详细参数 | | | |
|---------|------------------|----------------------------|------|
| 经度 (°): | 119.370903418925 | X轴速度 (m/s): | 200 |
| 纬度 (°): | 33.3969648279763 | Y轴速度 (m/s): | -200 |
| 高度 (°): | 12000 | Z轴速度 (m/s): | 0 |
| | | X轴加速度 (m/s ²): | 0 |
| | | Y轴加速度 (m/s ²): | 0 |
| | | Z轴加速度 (m/s ²): | -10 |
| | | 截面积 (m ²): | 7.5 |
| | | 确定 | 取消 |

图 5.13 “添加目标” 对话框

该对话框显示用户只需点击的确切经纬度位置，用户可以调整数值。此外，用户可以输入目标的高度，速度，加速度和目标横截面面积的参数值。设定目标值的详细参数，单击“确定”，然后在地图上添加成功的目标。显示如图 5.14:



图 5.14 成功添加目标

5.1.4.6 保存场景

用户情景模拟现场设置完毕后，单击“保存”图标，以保存设置场景信息，保存成功后，系统会提示保存成功。场景信息将被写入预定好的模拟控制组件 XML 参数表中。

5.1.5 雷达仿真控制组件

平台界面，点击开始仿真控制图标，打开仿真控制软件。不可分割的组成部分的模拟控制软件的菜单栏，工具栏，地图，场景树列表的左侧和下检测信息列表列的信息。

仿真控制组件启动时，它会加载 XML 参数表已设置读取场景信息，并将其显示在 MapX 地图控制，仿真开始后实时访问内存共享技术的仿真内核将用于模拟场景的信息和探测信息，在用户的模拟操作时，该组件会相应的控制仿真内核。图 5.15 模拟仿真过程控制组件界面。

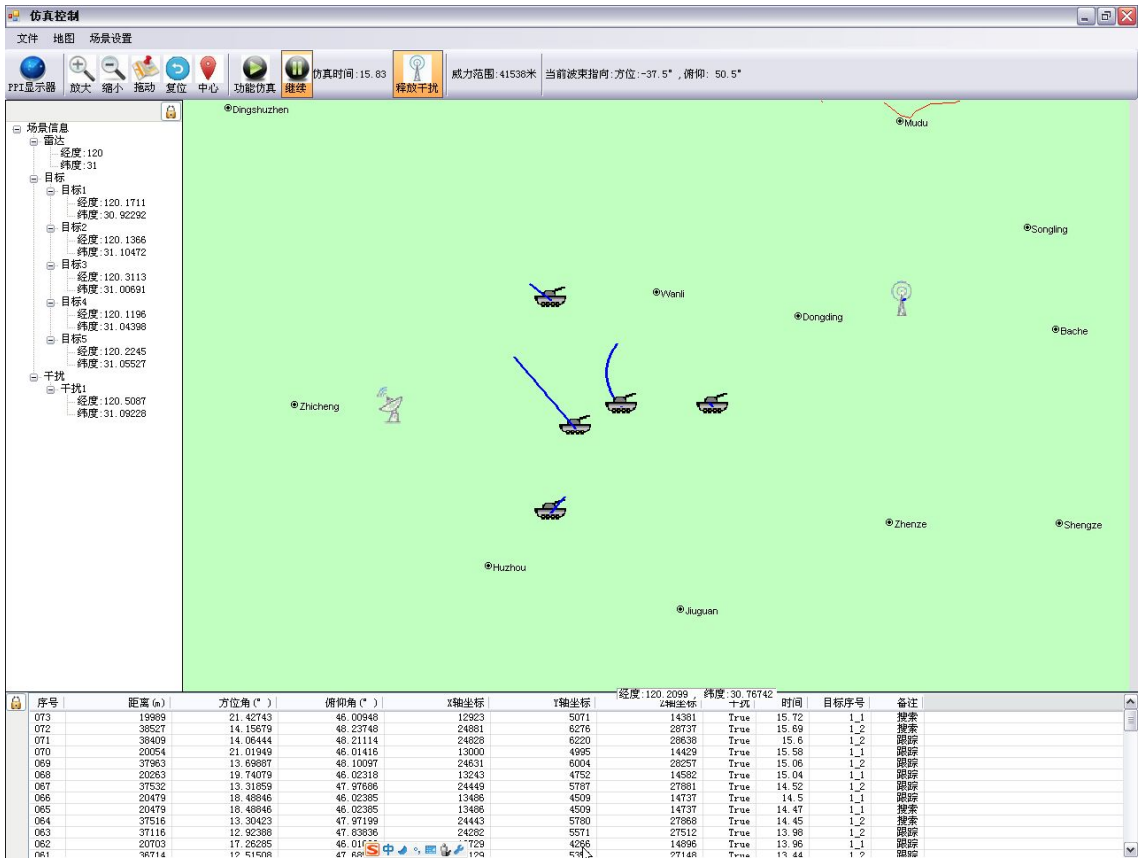


图 5.15 仿真控制界面

5.1.5.1 地图与航迹

软件开启时，会自动加载的 XML 表已设置的场景信息，并显示在地图上，如图 5.15 所示，系统会自动加载并显示一个雷达单元，五个目标的单元和干涉单元。仿真开始时，系统会读取内存中，并得到了真正的运动每个模拟单元信息的，然后它的轨迹在地图上绘制，蓝线是目标轨迹。图 5.15 仿真控制接口

同时，在信号级仿真中，该雷达能够检测到目标时，系统将读取目标轨道检

测单元绘制，以方便用户的结果进行比较和评价的雷达的探测性能的雷达信息的通信和信息的其他组成部分。如图所示，真正的目标轨迹为蓝色，红色轨迹是雷达探测信息。

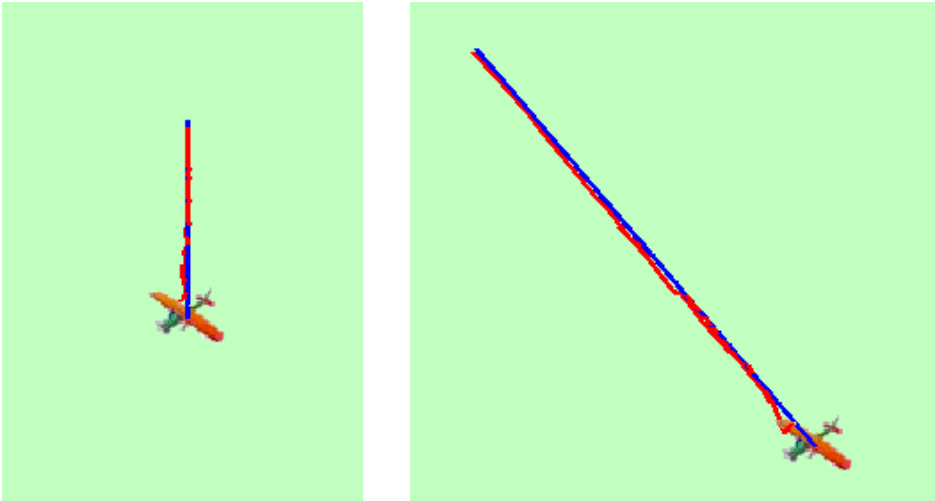


图 5.16 目标运动轨迹和探测航迹的绘制

5.1.5.2 探测信息列表

跟踪显示，可更直观地反映雷达探测结果，，足以弥补检测信息列表是不准确的缺点。下面列出检测软件会检测所有批次的雷达信息显示，其内容包括航迹编号、距离(m)、方位角($^{\circ}$)、俯仰角(m)、X 轴坐标(m)、Y 轴坐标(m)和 Z 轴坐标(m)。通过这些信息，用户可以对每次的雷达探测信息进行细致的评估。图 5.17 为软件运行时的截图。

| 序号 | 距离 (m) | 方位角 ($^{\circ}$) | 俯仰角 ($^{\circ}$) | X轴坐标 | Y轴坐标 | 经度: 120.2099 | 纬度: 30.75742 | 时间 | 目标序号 | 备注 |
|-----|--------|--------------------|--------------------|-------|------|--------------|--------------|-------|------|----|
| 073 | 19989 | 21.42743 | 48.00948 | 12923 | 5071 | 14381 | True | 15.72 | 1_1 | 搜索 |
| 072 | 38527 | 14.15679 | 48.23748 | 24881 | 6276 | 28737 | True | 15.69 | 1_2 | 搜索 |
| 071 | 38409 | 14.06444 | 48.21114 | 24828 | 6220 | 28638 | True | 15.6 | 1_2 | 跟踪 |
| 070 | 20054 | 21.01949 | 46.01416 | 13000 | 4995 | 14429 | True | 15.58 | 1_1 | 跟踪 |
| 069 | 37963 | 13.89887 | 48.10097 | 24631 | 6004 | 28257 | True | 15.06 | 1_2 | 跟踪 |
| 068 | 20263 | 19.74079 | 46.02318 | 13243 | 4752 | 14582 | True | 15.04 | 1_1 | 跟踪 |
| 067 | 37532 | 13.31859 | 47.97686 | 24449 | 5787 | 27881 | True | 14.52 | 1_2 | 跟踪 |
| 066 | 20479 | 18.48846 | 46.02385 | 13486 | 4509 | 14737 | True | 14.5 | 1_1 | 跟踪 |
| 065 | 20479 | 18.48846 | 46.02385 | 13486 | 4509 | 14737 | True | 14.47 | 1_1 | 搜索 |
| 064 | 37516 | 13.30423 | 47.97199 | 24443 | 5780 | 27868 | True | 14.45 | 1_2 | 搜索 |
| 063 | 37116 | 12.92388 | 47.83836 | 24282 | 5571 | 27512 | True | 13.98 | 1_2 | 跟踪 |
| 062 | 20703 | 17.26285 | 46.010 | 129 | 4268 | 14896 | True | 13.96 | 1_1 | 跟踪 |
| 061 | 36714 | 12.51508 | 47.850 | 24282 | 5571 | 27148 | True | 13.44 | 1_2 | 跟踪 |

图 5.17 探测信息列表

检测信息表在软件中打开时自动隐藏，用户可以通过移动鼠标附近的下边框的软件，以便查看列表框中，如果用户需要很长时间才能查看列表框中，可以点击顶部左上角的锁定键锁定和取消自动隐藏。

5.1.5.3 场景信息树状表

场景信息树表为用户提供每个模拟场景实时模拟单元纬度和经度坐标，使得用户可以查看实时仿真装置运动。如图 5.18 所示的具体的显示形式。

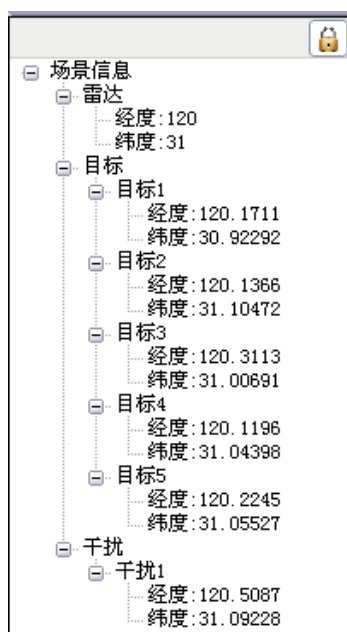


图 5.18 场景信息树状列表

场景信息树表在软件中打开自动隐藏，只有用户移动鼠标靠近左侧的软件边界只发生在用户需要很长时间才能查看列表框中，您可以点击右上角锁定按钮来锁定和取消自动隐藏。

5.1.5.4 菜单栏和工具栏

菜单栏为用户提供了退出功能和 5 种地图查看功能。而工具栏包含了地图工具，仿真控制工具，雷达模式切换工具和 PPI 显示打开。图 5.19 显示了。



图 5.19 仿真控制工具栏

使用鼠标点击 PPI 显示器按键可以打开 PPI 显示器；期间随时可以用于仿真的五种地图工具（放大、缩小、拖动、复位以及中心显示），使用户能够查看实时仿真场景信息，启动和停止按钮控制的模拟开始和停止功能接收和释放按钮干扰指定是否申请干扰的模拟过程。此外，该工具栏还设置了一个文本框显示当前功率的雷达范围和波束指向。

5.1.5.5 报文发送

按照项目的要求，软件的仿真进行了定期自动检测雷达侦察信息通过 UDP 报文发送到上级单位。考虑到数据拥塞问题，传输时间间隔设置为 10 秒，时间通过

发送“Timer”控制和“Tick”活动的实施。

此外，用户还可以发送邮件到指定位置报告的具体信息。点击工具栏上的按钮指定位置的数据包，鼠标光标会变成交叉，指定地图上的位置，单击“详细参数”窗口中会出现确认参数，并单击“确定”后，系统会发送消息到上级单位。

5.1.6 雷达仿真结果组件

在启动平台的界面中，单击“显示和控制终端的图标，能够使用雷达显示和控制终端软件。在菜单栏的“工具”子菜单中提供了 PPI 显示变焦功能，减少功能，复位功能，拖动功能和显示软件参数设置功能。

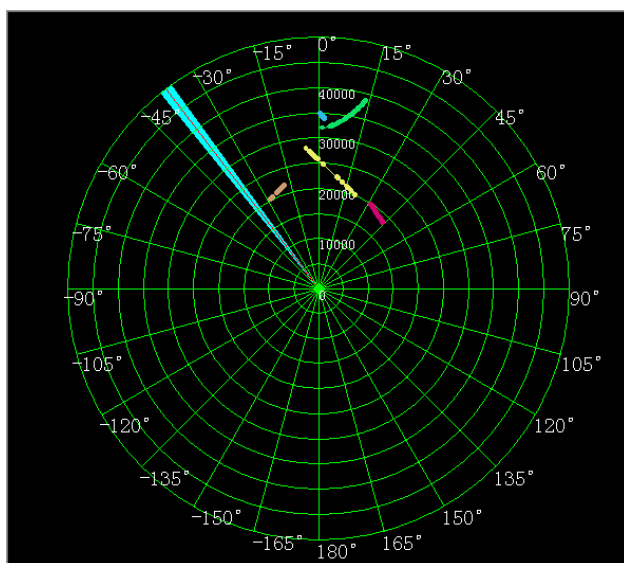


图 5.20PPI 显示器

5.1.7 其他组件

在平台界面点击单击性能评估图标或者数据回放图标，打开性能评估软件或数据的回放软件。

5.2 雷达仿真系统分析

首先，在视觉方面，本套系统具有友好的人机界面，操作方便，易于理解。并引进雷达 MapX 地图控制仿真系统，使系统设置的场景和场景的信息显示链接从过去传统的“数字化”的做法，从看到的结果，模拟单元可以直接显示在地图上相应的位置。用户可以直接看到地图附加的模拟单元，而不是转换成纬度和经度或直角坐标系和其他要设置的值，可以直观地看到每个模拟单元和运动趋势和

模拟单元的相对位置的当前位置显示在地图上信息，而不是通过系统的各种数值模拟的行为给出估计。

其次，基于组件技术的开放式雷达仿真系统的最大特点是，以反映其优势在于在模拟系统中是最好的地方是开放的，可扩展性，可重用的。该仿真系统的每个组件的实例是独立的可执行程序，他们一致的通信的装置，通过数据交换和形成一个系统，在该系统中，任何一个组件的位置是开放的。开发人员可以使用新的组件，以取代现有的组件或添加新的组件来扩展系统，你也可以使用现有的组件，建立一个新的系统。

下面的雷达仿真系统基于开放的软件系统的基本功能相结合，具体实施过程中，从以下几个方面来分析的雷达仿真系统的开放性：

组件：整个雷达仿真系统组件化的功能要求和能力的基础上通过粒度。

标准化：模拟系统组件的基础上，根据使用的通信标准的设计，一个标准的接口中的每个组件。

可重复使用：每个组件都是一个独立的可执行程序，你可以很容易地实现部分系统组件的重用，甚至完全可重复使用。

可扩展：由于成分分离接口和实现设计的，他们也不会改变组件的接口，一个单一的组件实现的变化不影响整体系统的运行，更新组件单元可能还需要重新编译整个软件系统将采取效果。

参数：仿真参数在 XML 配置参数表和记录的形式，通过一个可视化界面参数可以设置和保存，你可以随时调用。

互换性：组件都有一个标准化的接口设计，只要一个一致的界面，不管如何实施，相互之间是可以互换的。

基于上述分析，雷达仿真系统实例完全满足的可视化开放的雷达仿真系统的要求，以达到设计目标。

5.3 雷达仿真系统举例

为验证本文研究的可视化开放式雷达仿真系统的使用是否能满足需求，特设定以下仿真实例，对其进行验证。

5.3.1 雷达仿真示例参数

雷达主要参数如下。

表 5.1 雷达参数表

| 参数名称 | 参数数值 | 参数单位 |
|------|------|------|
|------|------|------|

| | | |
|----------|-----|-----|
| 峰值功率 | 20 | kW |
| 脉冲宽度 | 10 | us |
| 脉冲重复周期 | 500 | us |
| 发射信号带宽 | 5 | MHz |
| 驻留脉冲数 | 16 | 个 |
| 采样频率 | 20 | MHz |
| 工作频率 | 10 | GHz |
| 天线主瓣增益 | 30 | dB |
| 系统损耗 | 4 | dB |
| 干扰发射功率 | 50 | kW |
| 干扰发射天线增益 | 30 | dB |

场景信息:

- 雷达: 经度 120.00, 纬度 31.00, 高度 0m。
- 目标 1: 经度 120.16, 纬度 30.91, 高度 1000m; 初速度分量 $V_x=100\text{m/s}$, $V_y=100\text{m/s}$, $V_z=130\text{m/s}$; 加速度分量 $A_x=0\text{m/s}^2$, $A_y=0\text{m/s}^2$, $A_z=-10\text{m/s}^2$ 。
- 目标 2: 经度 120.16, 纬度 30.09, 高度 1000m; 初速度分量 $V_x=-150\text{m/s}$, $V_y=100\text{m/s}$, $V_z=100\text{m/s}$; 加速度分量 $A_x=0\text{m/s}^2$, $A_y=0\text{m/s}^2$, $A_z=-10\text{m/s}^2$ 。
- 目标 3: 经度 120.32, 纬度 31.00, 高度 2000m; 初速度分量 $V_x=-50\text{m/s}$, $V_y=50\text{m/s}$, $V_z=50\text{m/s}$; 加速度分量 $A_x=0\text{m/s}^2$, $A_y=0\text{m/s}^2$, $A_z=-10\text{m/s}^2$ 。
- 目标 4: 经度 120.18, 纬度 30.98, 高度 20000m; 初速度分量 $V_x=-450\text{m/s}$, $V_y=450\text{m/s}$, $V_z=436\text{m/s}$; 加速度分量 $A_x=0\text{m/s}^2$, $A_y=0\text{m/s}^2$, $A_z=10\text{m/s}^2$ 。
- 目标 5: 经度 120.23, 纬度 31.00, 高度 20000m; 初速度分量 $V_x=-400\text{m/s}$, $V_y=400\text{m/s}$, $V_z=400\text{m/s}$; 加速度分量 $A_x=50\text{m/s}^2$, $A_y=0\text{m/s}^2$, $A_z=20\text{m/s}^2$ 。
- 干扰 1: 经度 120.51, 纬度 31.10, 高度 12000m; 初速度分量 $V_x=10\text{m/s}$, $V_y=10\text{m/s}$, $V_z=0\text{m/s}$; 加速度分量 $A_x=0\text{m/s}^2$, $A_y=0\text{m/s}^2$, $A_z=-15\text{m/s}^2$ 。

5.3.2 雷达仿真示例结果

读取雷达模型参数, 和模拟场景信息, 在控制单元得到操作的结果, 并传输到仿真终端显示和控制元件, 分别在上图显示在监视器上的模拟和 PPI。图 5.21 中, 共有 1 个雷达, 5 个目标和 1 个干扰, 蓝线为实际的轨迹。短的干扰轨迹, 表明移动速度较慢的目标, 有两个高速移动的目标, 三个低速运动目标。地图中的目标轨迹为曲线, 分析表明, 有较大的加速度在 X 轴的正方向上, 查看场景参数, 为目标 5。

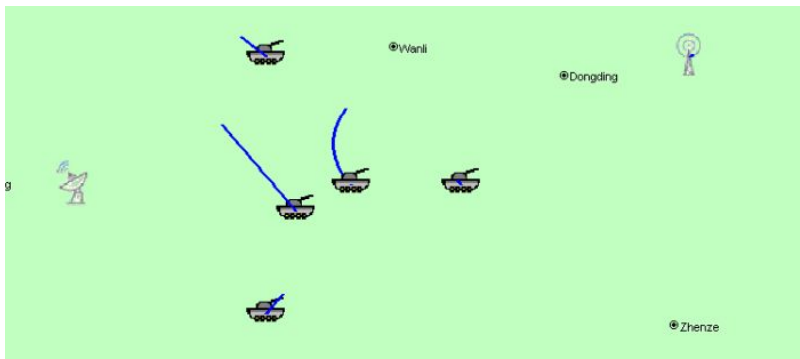


图 5.21 仿真实例场景显示

从场景信息树状列表可以看到具体的仿真单元运动信息，如图 5.23 所示。



图 5.22 场景信息树状列表

由上述可见，模拟单元的运动信息与设置相符，运算是正确的。

该雷达的信息时，系统将显示它更常用的在实际的雷达 **PPI** 显示。图 5.23 所示。**PPI** 显示绿色的线的经纬线，青色风扇的扫描光束和标有白色的角度和距离值。不同的颜色代表不同的跟踪收集目标检测，跟踪和场景真实轨迹基本上是相同的，一共有五种，即两个高速目标和三个低速目标，包括一个曲线运动，符合基本设置。

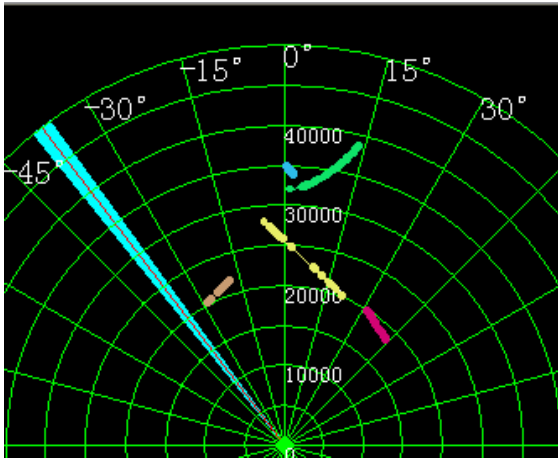


图 5.23 仿真实例仿真结果

具体信息可从雷达仿真模拟控制元件低于检测获得的信息列表所示，在 5.24 以下。设置目标的运动参数作为基础的信息通过以下方式获得探测雷达的信息是正确的，以验证模拟的正确性检查。

| 序号 | 距离 (m) | 方位角 (°) | 俯仰角 (°) | X轴坐标 | Y轴坐标 | Z轴坐标 | 十进 | 时间 |
|-----|--------|----------|----------|-------|------|-------|------|-------|
| 073 | 19989 | 21.42743 | 46.00948 | 12923 | 5071 | 14381 | True | 15.72 |
| 072 | 38527 | 14.15679 | 48.23748 | 24881 | 6276 | 28737 | True | 15.69 |
| 071 | 38409 | 14.06444 | 48.21114 | 24828 | 6220 | 28638 | True | 15.6 |
| 070 | 20054 | 21.01949 | 46.01416 | 13000 | 4995 | 14429 | True | 15.58 |
| 069 | 37963 | 13.69887 | 48.10097 | 24631 | 6004 | 28257 | True | 15.06 |
| 068 | 20263 | 19.74079 | 46.02318 | 13243 | 4752 | 14582 | True | 15.04 |
| 067 | 37532 | 13.31859 | 47.97686 | 24449 | 5787 | 27881 | True | 14.52 |
| 066 | 20479 | 18.48846 | 46.02385 | 13486 | 4509 | 14737 | True | 14.5 |
| 065 | 20479 | 18.48846 | 46.02385 | 13486 | 4509 | 14737 | True | 14.47 |
| 064 | 37516 | 13.30423 | 47.97199 | 24443 | 5780 | 27868 | True | 14.45 |
| 063 | 37116 | 12.92388 | 47.83836 | 24282 | 5571 | 27512 | True | 13.98 |

图 5.24 探测信息列表

5.4本章小结

本章以视景仿真系统开放式雷达为例，分析了其仿真分析需求，以及系统组件的分工。在此基础上，给出了比较全面的分析和可视化仿真系统和开放系统性能的例子。本章研究证明了该可视化的开放式雷达仿真系统的可行性和优越性。

结束语

目前，雷达仿真系统的规模迅速扩大，增长的功能，系统变得越来越复杂，仿真软件的要求也越来越高。传统仿真软件的使用，不仅容易，但开发周期长，维护和升级也很困难，难以适应逐渐雷达仿真要求。在这方面，基于纸张的软件工程可视化和开放式的软件设计思路，结合雷达领域的一个可视化的开放式雷达仿真系统的设计和实现方法的特点，以提高仿真软件的用户体验和可重用，可扩展容量的雷达仿真系统，具有一定的理论意义和较强的应用价值。

基于分析的可视化和开放式的软件设计的基础上，结合雷达领域，本文提出了基于一个开放的可视化雷达模拟系统的设计方法和实现进行研究，主要工作体现在：

1. 雷达仿真技术和组件复用的研究现状，系统总结，这使得开雷达仿真系统的研究思路，开放的软件构件技术和设计理念融入到雷达仿真，雷达仿真系统的开发开放；

2. 学习先进的可视化应用程序开发工具，编程技术，使用图形界面开发的雷达仿真系统，实现了视景仿真系统的新功能。成功将 MapX 地图控件嵌入雷达仿真系统进一步实现了雷达模拟可视化新功能

3. 实现一个可视化的开放式雷达仿真系统支持软件，结合项目要求的背景，发展应用系统的例子，介绍的可视化开放式雷达模拟系统的主要工作过程，其可视化功能和开放性综合分析。

在本文中，开放的雷达仿真可视化系统，更全面和深入的研究，提出了一些改进，并取得了一定的成果和结论。由于时间和空间限制的努力和研究论文，不是所有的问题得到有效解决。总结起来，你需要检查的深度和方向包括：

1. 雷达模拟组件标准化

该系统由一个独立的可执行程序文件的组成部分，虽然该组件构成，以满足当前的使用要求，但仍限制不能被忽略。下一步应该使用更多标准化部件，如 COM 标准，凭借微软操作系统平台的优势，广泛的影响力。COM 组件的运行效率比较高，并且有一系列的更复杂的支持工具，且易于实施。同时，COM 标准的组件和组件之间的通信机制规范，这些规范是不依赖于任何特定的编程语言和操作系统。因此，在目前的技术条件下，雷达仿真系统的开发基于 COM 的标准是可行的。

2. 实施雷达模拟组件库

在基于组件的开发模式，雷达模拟元件和重用雷达模拟组件的使用效率还需要合理的管理，需要雷达仿真组件库。雷达仿真组件库，用于存储和管理的雷达

模拟组件领域，一般包括模拟组件的存储，查询，修改，删除，下载等功能。此外，需要雷达原理分类，存储，检索方案，以提高雷达模拟组件库开发的生产力和效率的设计特色区域。

3. 雷达仿真系统，以进一步改善用户体验

技术指标与客观的评估标准不同，可视化在一定程度上，一个人的主观感受，你需要不断进步，为了满足用户的需求。无论是从软件的操作或运行经验反馈和仿真结果表明，本文研究的软件有很多的增长空间。提高技术标准，人机交互方面也将带来改善软件质量的改善。

致谢

两年多的硕士研究生学习生活即将结束，在这期间，老师、同学、朋友和家人帮助我不断地学习与进步，包括论文工作的顺利完成，都离不开他们对我的关心和协助。

首先衷心感谢我的导师张林让教授。张老师精深的学术水平、严谨的治学态度、求实创新的工作作风使我受益匪浅，更为我今后的工作与生活树立了良好的学习榜样。两年多来张老师不仅在学习中对我悉心指导，更在生活中给予了极大帮助在此向张老师表示深深的感谢和诚挚的敬意。

感谢张娟、刘楠、刘高高、刘昕、周宇、张波、虞敬璠、郭伟、胡子军、张静强、胡贵生等师兄师姐在我读研期间所给予的帮助和关心。

感谢缪鹏志、薛俊杰、陈娟、郑娇、王胜喜、朱保安、张杰、熊瑶、陈锦宇、李奇、柏龙等同窗，在与他们的探讨和交流中，不断地开阔我的视野，加深了我们之间的友谊，这对我来说是十分珍贵的财富，在今后一定会倍加珍惜。

最后，我要衷心地感谢我的父母、家人，是他们辛苦养育了我，给了我学习的动力，祝你们身体健康、万事如意！

感谢所有给予我关心和帮助的人！

参考文献

- [1] 杨万海著. 雷达系统建模与仿真. 西安: 西安电子科技大学出版社, 2007.01.
- [2] 郭金良. 基于构件技术的开放式雷达仿真系统研究[D]. 长沙: 国防科技大学, 2010.
- [3] 王恒霖, 曹建国等编著. 仿真系统的设计与应用. 北京: 科学出版社, 2003.
- [4] 张媛. 雷达目标环境建模与仿真[D]. 西安电子科技大学, 2006.
- [5] 李斌, 周云飞, 唐小琦. 基于 COM 技术和 OPC 规范的开放式结构数控系统研究. 中国机械工程, 2003, 14(4): 310~313.
- [6] 杨虔静. 数字阵列雷达自适应通道均衡算法研究[D]. 西安电子科技大学, 2011.
- [7] 胡卫东, 宋锐. 雷达目标识别开放系统结构与应用. 现代雷达, 2008, 30(3): 58~60.
- [8] 曹亚峰. 主被动雷达制导系统仿真与评估 [D][D]. 西安电子科技大学, 2009.
- [9] 覃征等著. 软件体系结构. 西安: 西安交通大学出版社, 2002.
- [10] 王健. 雷达半实物仿真系统的设计与实现[D]. 西安电子科技大学, 2009.
- [11] 王国玉, 肖顺平, 汪连栋. 电子系统建模仿真与评估. 长沙: 国防科技大学出版社, 1996.
- [12] 李延飞. 机载雷达多目标跟踪技术研究[D]. 西安电子科技大学, 2006.
- [13] 王国玉, 汪连栋等. 雷达系统数学仿真与评估. 北京: 国防工业出版社, 2004.
- [14] 孙玉婷. 数字通信信号参数估计与自动识别[D]. 西安电子科技大学, 2009.
- [15] 王雪松, 肖顺平, 冯德军等. 现代雷达系统建模与仿真. 北京: 电子工业出版社, 2010.
- [16] 郭伟. 雷达系统仿真软件的设计与实现[D]. 西安电子科技大学, 2012.
- [17] 李连营、李清泉、李汉武、颜辉武著. 基于 MapX 的 GIS 应用开发. 武汉: 武汉大学出版社, 2003.06.
- [18] 夏烨巍. 雷达导引头抗干扰系统的建模和评估[D]. 西安电子科技大学, 2010.
- [19] Yisak Abdella ; Knut Alfredsen. A GIS toolset for automated processing and analysis of radar precipitation data. Computers & geosciences. 2010, 36(4): 422~429.
- [20] 仇启明. 飞行器着陆导航敏感器信号处理方案设计和仿真[D]. 西安电子科技大学, 2009.
- [21] Bassem R. Mahafza 著, 陈志杰、罗群等译. 雷达系统分析与设计 (Matlab 版) (第二版). 北京: 电子工业出版社, 2008.10.

- [22] 黄海光.主被动复合雷达导引头建模与仿真[D].西安电子科技大学,2010.
- [23] 李泽等编著. C#函数实用手册. 北京: 冶金工业出版社, 2006.
- [24] (美) Chris H. Pappas, William H. Murray 著; 周良忠译. C#精髓. 北京: 人民邮电出版社, 2002.
- [25] 张文博.基于半实物仿真的雷达系统性能评估[D].西安电子科技大学,2010.
- [26] 雷刚.反舰导弹干扰建模与仿真[D].西安电子科技大学,2010.
- [27] ZhiRen LIU XiangHua YUE. Research and Development of GIS Airport Pavement Management System Based on MapX. International Conference on Intelligent Computation Technology and Automation (ICICTA 2009) 2nd. 2009: 407~409.
- [28] 李丽霞.单脉冲跟踪雷达杂波抑制技术及其 DSP 实现[D].西安电子科技大学,2010.
- [29] 赵泉, 钱志亚. 基于 MapX 实现雷达信号模拟器中态势图显示功能. 舰船电子对抗, 2010.04: 94~96.
- [30] 郭金良. 基于组件技术的开放式雷达仿真系统研究. 北京: 国防科学技术大学硕士学位论文, 2010.
- [31] 赵行云. 软件重用技术在雷达对抗仿真系统分布式数据处理中的应用. 成都: 电子科技大学硕士学位论文, 2006.
- [32] David M Dikel et al. Software Architecture: Organizational Principles and Patterns. China Machine Press,2002.
- [33] Liliana Dobrica and Eila niemela. A Survey on software Architecture Analysis Method[J] . IEEE Transactions on Software Engineering,2002,28(7):638-653.
- [34] 丁鹭飞、耿富录、 陈建春著. 雷达原理(第 4 版) . 北京: 电子工业出版社, 2009.03.
- [35] 张娟. 雷达系统仿真、评估与实测数据分析. 西安: 西安电子科技大学硕士学位论文, 2006.
- [36] Karli Watson、Christian Nagel、齐立波、黄静著. C#入门经典(第 5 版) . 北京: 清华大学出版社, 2010.
- [37] LI Lei, XIE Yong-jun, LI Xiao-feng, CHEN Bo-tao, CHEN Xiao. Pulse Doppler radar simulation system for detecting low altitude targets. Journal of Systems Engineering and Electronics 2010, 32(6) 1131-1135.
- [38] Eyung W. Kang. Radar System Analysis, Design and Simulation. Artech House Publishers, 2008.
- [39] Schleher, D.C.. Solving radar detection problems using simulation. IEEE Aerospace and Electronic Systems Magazine. 1995, 10(4): 36~40.

-
- [40] 雪松、肖顺平、冯德军著. 现代雷达电子战系统建模与仿真. 北京: 电子工业出版社, 2010.03.
- [41] Wu Feng, Shi Lin, Yang Wang-hai. Modeling and simulation of PD radar systembase on the SystemView. CIE International Conference on radar. 2001: 246~249 .

可视化技术在雷达仿真系统中的应用

作者: [刘昕玮](#)

学位授予单位: [西安电子科技大学](#)

引用本文格式: [刘昕玮](#) [可视化技术在雷达仿真系统中的应用](#)[学位论文]硕士 2013