

Studies on Recorded Fault Data Analyzing System

Based On MS.NET COM Module Program

Major: Electric Power System And Automation

Graduate: You Feng Advisor: Cheng hao

As an important chain of operation and maintenance of power systems, relay protection is developing in the direction of computerization, networking and system integration. At present, computer-based relay and device of FRD are extensive using in power net. Recording and analyzing the fault record information effectively is useful to understand fault reason, type and state accurately in time, handle it rapidly and reduce losses. At the same time the data kept can be used in off-line analysis, help to research and predict the fault law and offer reference to design the power net and dispose the protect device in the future.

This text researches and develops the fault record information analytical system based on MS.NET and MATLAB interface programming. Use the mixed programming technology can give full play the advantage of different software and improve working efficiency. MS.NET has offered a kind of new running environment, i.e. .NET framework; enable to write out the fine, strong procedure code easily and rapidly, and can manage, dispose and revise codes conveniently. MATLAB is the most extensive used calculating software in the world, and it processes powerful ability of value calculation and data visibility. Because MATALB has already arranged the concrete algorithm of many kinds of mathematics questions into a available function, therefore simplified the algorithm procedure greatly. The system adopt the latest Object-oriented module programming technology, has the entire graph and the mouse operation contact surface, can read, enter and keep

standard COMTRADE record wave file , carry on wave form reveal with various functions analyzing conveniently automatically. The system has the good expansion ability, may realize good integrations with the other module of fault information management system.

This article based on MS. NET platform realizes the C# language and the MATLAB mixed programming and uses it in the analysis of fault record data and provides the new mentality and the method to the development of fault record analysis software for the power system.

Keyword: MS.NET, MATLAB, COM Program, Fault Record Analysis

第一章 绪论

§ 1.1 课题的背景与意义

随着我国电力行业的飞速发展，电网的规模日益扩大，电网的结构更加复杂，一方面，用户对供电质量的要求越来越高，另一方面，电网的扩大和运行条件更加复杂，使得电网发生故障的可能性也大大增加。为保证电网安全稳定地运行，必须能够迅速地查找故障源，判别故障类型并进行正确的故障处理以尽快地恢复电网运行，减小故障损失，因此，对电力系统的日常运行管理工作提出了更高的要求。

随着计算机技术地不断发展和深入，电力系统中的传统设备也正在被微处理器所替代。继电保护作为电力系统运行与维护的重要环节，也在向着微机化、网络化和系统集成化方向发展。目前微机型继电保护、故障录波器在电网中已广泛使用，有效地对故障录波信息进行记录、分析对于及时准确地了解故障原因、类型和状况，迅速地做出处理，减小故障损失有着重要的意义。同时保存下来的数据可以用于离线分析，有利于对故障规律进行研究和预测，对日后的电网设计和保护设备配置提供参考。面对系统故障产生的大量复杂信息，依靠人工的方式进行计算分析判断显然是十分困难的，因此，设计使用一套优良的信息分析处理系统是十分必要的。

随着电力系统自动化水平的不断提高，其软件系统的规模越来越庞大，结构也越来越复杂，传统的结构化程序设计方法面临着日益突出的维护、扩充和升级困难等问题。同时，针对故障信息分析研究，国内外许多文献都提出了大量的算法，但由于目前流行的操作系统开发语言数值计算的能力较弱，要实现这些算法是相当困难的；而专业的工程软件又不具备良好的操作平台和图形界面设计能力，不能很好地进行桌面应用程序开发。随着面向对象组件技术和混合编程技术的发展，为我们解决这一矛盾提供了新的方法。

面向对象编程技术是为解决传统程序设计方法所面临的日益突出的开发效率低、可重用性差、维护和扩充困难等问题而产生的。它的核心思想是模拟人的思维方式，尽可能地使程序的结构和实现与其所描述的现实世界保持一致，充分保证计算机领域的概念与问题域的概念之间的一致性。同以往的结构化技术相比，它可以更好地描述现实世界，具有提高开发效率和运行可靠性，

便于调试、维护和升级等优点。混合编程技术是在面向对象编程技术的基础上发展起来的，它是通过应用不同的软件分别进行设计开发，形成不同的组件，再通过集成的方法组合起来以满足系统不同方面的需要。应用混合编程技术可以充分发挥不同软件的优势，提高工作效率。

基于以上的考虑，本文研究开发了基于 MS.NET 和 MATLAB 接口编程的录波信息分析系统。系统采用最新的面向对象组件技术，具有全图形和鼠标的操作界面，能够自动读入和保存标准 COMTRADE 录波文件，方便地进行波形显示和各种功能分析。同时系统具有良好的扩展能力，可以和故障信息处理系统其它模块实现良好集成。

§ 1.2 面向对象程序设计

当软件工程达到一定的复杂程度时，要组织程序的源代码，记住函数内部的功能，这些的工作量就变得日益庞大，以至于超过了解决问题本身所需的工作量。以往面向过程的设计方法主要存在以下不足：1、功能与数据分离，不符合人们对现实世界的认识。要保持功能与数据的相容也十分困难；2、基于模块的设计方式，导致软件修改困难；3、自顶向下的设计方法，限制了软件的可重用性，降低了开发效率，也导致最后开发出来的系统难以维护。面向对象技术将数据和对数据的操作作为一个相互依赖、不可分割的整体，采用数据抽象和信息隐蔽技术，力图使对现实世界问题的求解简单化。它符合人们的思维习惯，同时有助于控制软件的复杂性，提高软件的生产效率。

§ 1.2.1 面向对象程序设计的基本思想^④

- ◆ 认为世界是由各种对象组成，任何事物都是某个对象类的实例，而复杂的对象可以由比较简单的对象以某种方式组成。
- ◆ 把所有对象都划分成各种对象类，每个对象类都定义了一组方法，所谓方法实际上是允许施加于该类对象上的各种操作。对象和传统的数据的本质区别，不是被动地等待对它执行某种操作，而是进行处理的主体。
- ◆ 对象之间除了互相传递消息的联系之外再没有其它联系。一切局部对象的信息和实现方法都被封装于相应的对象类的定义之中，在外界是不可见的。

- ◆ 对象类将按照类、子类与父类的关系构成一个层次结构的系统，在这种层次结构中，上层对象类所具有的性质可以被下层对象类继承，同时下层对象类也可以对相应的属性进行重新描述。

§ 1.2.2 面向对象程序设计的关键技术^[34.7]

1、对象和类。对象是面向对象技术的基本成分。每个对象可用它本身的一组属性和包含的一组操作来定义。对象可以是现实生活中的一个物理对象，还可以是某一类概念实体的实例。从分析和设计的角度来看，对象表示了一种概念，它们把有关的现实世界的实体模型化。类是一组具有相同数据结构和相同操作的对象的集合。类是对一系列具有相同性质的对象的抽象，是对对象共同特征的描述。在一个类中，每个对象都是类的实例，可以使用类中提供的方法。

2、继承性。继承允许构建类层次结构。类层次结构是一组相关的类，它们共享某些内容。派生类不仅继承了基类的所有成员，还可以给继承的类添加新行为。继承避免了代码的重复，允许重用代码，可以在已经定义的对象类型的基础上创建更加丰富和复杂的对象模型，从而避免了重复编程。

3、封装性。在程序设计中，封装是指将一个数据和与此数据有关的操作集合放在一起，形成一个能动的实体-对象模型。封装的目的在于将对象的使用者和设计者分开，用户不必知道对象行为的实现细节，只需要根据对象提供的外部特性接口访问对象。

4、多态性。多态性由继承支持。因为所有的派生类都继承了基类的基本功能，所以可以把派生类的实例看作基类的实例。因此可以提取出基类的一个实例，用派生类的一个实例替换它，而代码仍能继续工作。把一种类型的对象视作另一种类型的实例就叫做多态性。

§ 1.2.3 混合编程技术的使用

本文使用 MS.NET 下的 C#语言和 MATLAB 语言进行集成以实现混合编程。MS.NET 是微软公司推出的下一代软件运行开发平台。它提供了一种新的运行环境，即.NET 框架，使得程序能够更容易、迅速地写出优良、健壮的程序代码，并且能够方便地管理、部署和修改代码^[1]。MATLAB 是当今世界上使用最为广泛的数值计算软件之一，它具有强大的数值计算和数据可视化能力，

并已广泛应用于教学和科研。由于 MATLAB 已将多种数学问题的具体算法编成了现成的函数，因此大大简化了算法程序。同时它具有强大的矩阵计算功能，利用一般的符号和函数就可以对矩阵进行加减乘除运算以及转置和求逆运算等，还可以处理稀疏矩阵等特殊的矩阵^[5]，非常适合用于大型数值算法的编程。本文基于 MS.NET 平台实现 C# 语言和 MATLAB 的混合编程，并应用于故障录波数据的分析，为电力系统录波分析软件的开发提供了新的思路和方法。

§ 1.3 应用现状及发展趋势

随着面向对象技术的发展，它也被逐渐引入电力系统应用软件的设计开发之中，以提高软件的开发效率和可移植性。文献[6]建立了潮流计算的对象模型，设计了节点、注入电流等概念类，并具体说明了由各节点生成代表自身的 Jacobi 对角模块、支路生成非对角模块的得到全系统 Jacobi 矩阵的方法。文献[7]利用对象的消息传递的特点，将面向对象技术应用于网络拓扑分析和潮流计算，实现数据库、网络分析和图形界面的统一。文献[8]应用面向对象设计的概念和 C++ 语言建立电力系统仿真系统，针对 AC 潮流和 DC 潮流建立模型，实现了对任何数据类型的稀疏矩阵求解，并验证了其速度优于传统的潮流计算程序。文献[9]应用对象组件技术和面向对象的设计方法开发了分布式网络分析系统，并论述了应用此技术比较于传统的功能性设计方法在提高系统的适用性以及维护和升级能力方面的优势。

近年来面向对象技术及组件化程序设计在国内继电保护信息管理系统及故障录波分析中也得到了广泛的应用。

§ 1.3.1 在电网故障分析方面的应用

目前的故障分析软件普遍采用 VC++ 作为开发平台，采用面向对象的组件技术以模块化的方式设计实现。文献[10]以 Visual C++ V6.0 为开发工具，基于 IEEE 标准 COMTRADE 数据格式建立电网故障分析平台软件，采用模块化结构，具有较强的扩展能力。文献[11]将 COM+ 技术应用于录波分析软件，采取组件方式设计各功能模块，通过设计各服务组件的对象与接口，构架了三层分布式系统的框架，提高了系统开发效率，增强了系统的通用性和灵活性。文献[12]使用 VC++ 6.0 平台，借助于 CGraphFormX 控件，实现图形显示、图形编辑、频谱分析、打印等功能。软件具有启动判据全面、独立，人机界面友好，

故障数据记录完备, 图形显示和编程、分析功能强等特点。文献[13]采用基于OLE(对象的连接和嵌入)的ActiveX技术实现可切换的软件运行框架, 既可独立运行又可嵌入IE中运行, 使软件和网络实现了无缝连接, 开发了专门的COMTRADE读写引擎, 在全图形化的界面中实现故障分析的各种功能。文献[14]采用Visual C++6.0开发电力故障波形再现及分析系统, 主要包括数据远传、格式转换、信息管理、故障分析等模块, 实现格式转换、波形再现和分析、谐波和矢量分析、功率分析及故障分析等常用功能。

§ 1.3.2 在继电保护信息管理方面的应用

目前开发的信息管理系统已摒弃了过去的单机架构和胖客户端C/S(Client/Server)结构, 而普遍采用瘦客户端B/S(Browser/Server)结构设计开发。文献[15]应用Internet网络技术和数据库技术以及面向对象的设计方法, 设计了故障分析和管理系统以及通过Web同网络的连接, 具有远程访问、查询方便等特点。文献[16]提出了以浏览器、Web服务器和数据库服务器构成的基于Web的继电保护管理信息系统, 采用动态网页ASP技术实现, 具有在线管理、信息查询等功能, 具有开放性且易于使用、开发和维护。文献[17]设计了继电保护及故障信息管理系统的主站和子站系统, 子站系统接入保护及故障录波器并完成协议转换, 同时上传数据至调度端, 主站系统实现对上传数据的分类显示与管理, 并提供故障测距、录波数据分析等应用模块。文献[18]基于Web技术开发了多级电网继电保护整定和信息管理系统, 采用B/S结构和构件集成的设计模式实现保护整定和信息的Web发布。文献[19][20]分别采用ASP/ADO技术和J2EE技术建立了保护信息和故障计算的远程分布式系统。

§ 1.3.3 混合编程技术的应用

目前许多文献提出了多种MATLAB接口编程的方法, 主要是针对VB和VC++与MATLAB的集成, 如编写MEX文件、使用MATLAB引擎、安装MATCOM、应用MATALB Compiler和MATLAB Web Server等。文献[21]介绍了MATLAB软件的优缺点以及应用MTATLAB的API标准和Matcom工具实现与C/VC++的混合编程。文献[22]使用WindowsDLL技术, 利用MATLAB提供的C语言外部调用数学库建立起MATLAB与Winsows应用程序间的接口库, 从而实现系统的无缝联结。文献[23]介绍了应用MATLAB引擎、mcc编译器及

Matcom 工具实现 MATLAB 与 VC++混合编程的方法。文献[24]使用 MATLAB Add-in 工具实现与 VC/C++的接口技术。

总体来看，目前的图形化故障信息分析管理系统，还是普遍存在系统实现较为复杂，分布和配置较为困难等缺点。B/S 架构所有的计算都在服务器上完成，客户端只是充当一个浏览器的作用，因而不能支持脱机应用，而且不能充分利用本地计算机的计算能力，软件开发也比较复杂，响应速度较慢^[25]。VC++ 虽然可以实现良好的图形界面编辑设计，但其进行大型数值计算的能力相对较弱。有的系统虽然采用了基于 COM 技术或 ActiveX 的分布式组件方式，但是在实际应用中不同组件之间的接口实现十分复杂，同时其部署也很困难。全新的 MS.NET 框架和不断发展的混合编程技术为我们实现真正方便快捷的面向对象组件开发提供了新的方向。MS.NET 具有完全地面向对象设计及优秀的代码设计、管理和部署能力，其智能客户端（Smart Client）技术有效地结合了 C/S 结构和 B/S 结构的优点而摒弃其缺点，成为下一代客户端应用程序的代表。MATLAB6.5 以后提供了 COM 生成器，它也是 MATLAB 推荐使用的集成技术。使用 COM 组件使接口编程的实现更加简便，同时 MS.NET 对于 COM 组件也有很好的支持。应用 MS.NET 优秀的开发平台设计能力结合 MATLAB 强大的数值计算分析能力使软件的设计更加优良，算法的实现更加简便，同时系统的性能特别是可扩展性和可移植性也大大增强。

§ 1.4 本文所做的工作

1、介绍了开发工具的基本知识和混合编程的实现方法，运用面向对象的 COM 组件技术实现了 MS.NET 下 C#与 MATLAB 的集成和数据交换，同时利用 Windows Form 编程进行软件图形界面的设计，并阐述了故障信息分析软件开发的整体模型。COM 技术是目前广泛使用的接口标准，它使得系统的集成更加的方便，而且也为大多数的开发软件所支持。使用 Windows Form 设计的图形框架具有熟悉的界面和良好的操作性。

2、介绍了 ADO.NET 访问操作数据库的知识和具体实现过程。系统能自动读取标准 COMTRADE 文件，显示文件信息并对数据进行处理后显示波形。能够保存录波信息及数据文件，并提供了多种方式对历史数据的查询。能够自动添加和读取线路信息和线路参数，为分析计算提供方便。

3、研究了故障录波数据分析的算法以及各功能模块的设计和实现。软件

以多页面技术和全图形化的方式显示谐波、序分量等计算结果，自动进行故障起动判断和故障选相，根据故障选相结果自动计算阻抗并显示阻抗运行轨迹，同时建立保护的动作边界方程将保护的动作特性一同在阻抗平面上显示。

第二章 软件设计与实现

§ 2.1 系统开发工具简介

.NET 框架是一种新的计算平台，它简化了在高度分布式 Internet 环境中的应用程序开发，其特点表现为：

- ◆ 基于网络的应用服务，通过 XML 语言定义了简单对象访问协议(SOAP)、Web 服务描述语言 (WSDL)、Web 服务发现协议 (DISCO)，使未来的软件开发、维护和升级都将完全在互联网上进行，也使多种网络设备的访问变为可能。
- ◆ 面向对象的多语言平台，基于公共语言运行时 CLR(Common Language Runtime) 和类库的紧密结合，提供了强大的跨语言继承性、异常处理和调试功能，使不同系统和平台之间的交互和综合变得十分容易。
- ◆ 自动化的内存管理，垃圾回收器自动跟踪应用程序内存的分配、使用和释放，不必再编写执行内存管理任务的代码，解决以往常见的因内存访问而导致的错误。
- ◆ 安全的代码执行环境和版本控制，运行时环境的元数据包含了与其它模块相关联的所有信息，并强制执行安全性，避免了对注册表的操作，同时保证了版本的可靠性。
- ◆ 支持与 COM 的互操作，通过生成代理和其它技术，既可以在.NET 工程中使用 COM 组件，也可以在其它应用程序中使用.NET 组件。
- ◆ 按照工业标准生成所有通信，以确保基于.NET 框架的代码可与任何其它代码集成。

.NET 框架包含两个主要组件^[26]：公共语言运行库和.NET 框架类库。公共运行库提供核心服务（如内存管理、线程管理和远程处理），并且强制实施严格的类型安全以及可确保安全性和可靠性的其它形式的代码准确性。.NET 类库是一个综合性的面向对象的可重用类型集合，利用它可以开发包含从传统的命令行或图形用户界面 (GUI) 应用程序到基于 ASP.NET 所提供的最新应用程序（如 Web 窗体和 XML Web Services）在内的应用程序。

C#语言是微软公司专门为.NET 运行时高度兼容设计的简单的、现代的、面向对象的和类型安全的编程语言，它综合了 Visual Basic 的高效率和 C++ 功能的强大性及其熟悉的语法。与 VB 相比，它的代码可以实现与其它语言的交

互, 可以使用.NET 提供的丰富的基类, 语法也更为简洁。与 VC++相比, 它执行的效率更高, 放弃了容易出错的指针和指针算法(不安全代码除外), 使用完全面向对象的编程和 Windows Forms 快速开发图形化的用户界面。同时基于.NET 平台的强大功能, C#可以实现与其它语言的无缝集成。

C#支持面向对象编程的关键概念: 封装(Encapsulation)、继承(Inheritance)和多态性(polymorphism)。封装就是将数据或操作等集合在一个类中, 其目的在于保护或者防止代码被外部程序破坏。C#可以通过读写和属性的方式来实现封装。继承允许构建类层次结构, 定义基类和派生类。派生类可以继承基类的所有成员(包括属性、方法、域、事件、索引指示器等), 同时也可作为其它类的基类, 从而避免了代码的重复。多态性指同一操作可作用于不同的类的实例, 不同的类将进行不同的解释, 最后由系统来决定哪一个操作, 产生不同的执行结果。C#支持编译时的多态性和运行时的多态性两种类型。

§ 2.2 MATLAB 接口编程

MATLAB 的外部接口技术是它的一项重要功能。通过接口编程, 用户可以方便地利用 MATLAB 与其它语言实现交互和集成。本文着重阐述利用 MATLAB Builder for COM 实现接口编程的方法。

§ 2.2.1 COM 技术概述

COM (组件对象模型 Component Object Model) 组件技术是为实现面向对象化编程而产生的。软件的组件化设计使我们能够将大型的应用程序分为若干个相对独立的组件来分别进行开发、编译、调试和测试, 然后通过相互之间的接口组成实际需要的应用系统, 这样就可以通过协作缩短开发周期, 同时也便于进行系统的维护和升级。不难看出, 在实现组件化设计的过程中, 要保证系统在组合后能够有效地运行, 接口就成为最关键的技术。COM 就是基于 Windows 平台的广泛使用的组件接口标准之一。应用 COM 组件, 我们可以选择不同的开发语言来完成不同的工件, 然后再进行集成, 只要这些语言都支持标准的 COM 协议。MATLAB 从 6.5 版本开始提供 COM 生成器, 最新的版本为 1.1, 它可以将 MATLAB 算法生成 COM 组件, 直接被任何支持 COM 的语言所引用, 从而实现开发语言的集成。

§ 2.2.2 COM 生成器 (MATLAB COM Builder) 的应用^[27-31]

MATLAB COM Builder 是 MATLAB 编译器 (Compiler) 的扩展, 它提供了一个简单易用的图形化界面, 帮助用户把用 M 语言开发的算法自动而快速地转换为独立的 COM 组件对象。生成的 COM 组件对象可以在任何支持 COM 对象的应用中使用。通过 COM 组件, 可以同其它的用户共享算法, 并且 MATLAB 算法可以免费地随同 COM 应用程序发布。

在使用 MATLAB COM Builder 之前, 首先要做以下三个方面的工作:

- 进行编译器的配置。在 MATLAB 命令窗口敲入命令: `mbuild -setup` 和 `mex -setup`, 按照提示选择合适的编译器的设置, 注意应当选择支持 COM 标准的编译器。由于本文使用 VS.NET 2003 集成, 所以选择 Microsoft Visual C/C++ version 7.1 in “VS.NET dir”\Microsoft Visual Studio .NET 2003 选项。
- 设置系统路径。通常系统会自动设置系统路径。如果没有自动设置, 可以打开“我的电脑→属性→高级→环境变量→系统变量→path”选项, 增加以下路径:

头文件目录:

“MATLAB dir”\extern\include;

库文件目录:

“MATLAB dir”\extern\lib\win32\microsoft\msvc7;

DLL 文件目录:

“MATLAB dir”\bin\win32;

- 注册 `mwcomutil.dll`。该动态库包含了 MATLAB 定义的应用库, 主要是数据类型的声明等类库, 如果不进行注册将会出现数据类型转换无法完成、应用程序不能初始化等错误。通常在配置编译器时该动态库被自动注册, 如果没有被自己注册则需要进行手工注册。打开命令提示符窗口, 键入指令:

`‘MATLABROOT’\bin\win32>mwregsvr mwcomutil.dll;`

也可以在 MATLAB 命令窗口中按同样方法进行手工注册。完成预备工作后就可以开始使用 COM 生成器了。

§ 2.2.2.1 创建工程

在 MATLAB 命令窗口键入 comtool，打开“MATLAB COM Builder”对话框（注：本文使用 MATLAB Release 14，若使用 R13 会有所不同，可参阅有关资料）。

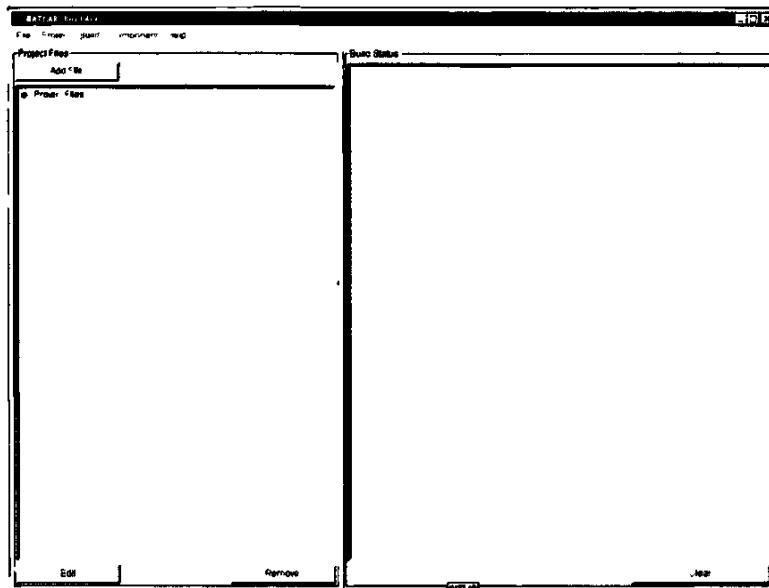


图 2.1 MATLAB COM Builder 对话框

对话框由工具条、项目文件（Project Files）和编译状态（Build Status）三部分组成。创建新项目时选择 FILE->New Project 选项，打开“New Project Settings”对话框，在 Component name 文本框中设置 COM 组件的名称，注意不要和任何 M 文件或 MEX 文件重名。在 Class name 框中设置类名，一般将鼠标点击空白位置系统会自动生成。Project version 是版本号，系统默认为 1.0。Project directory 是工程所在目录。

Complier options 是编译器配置

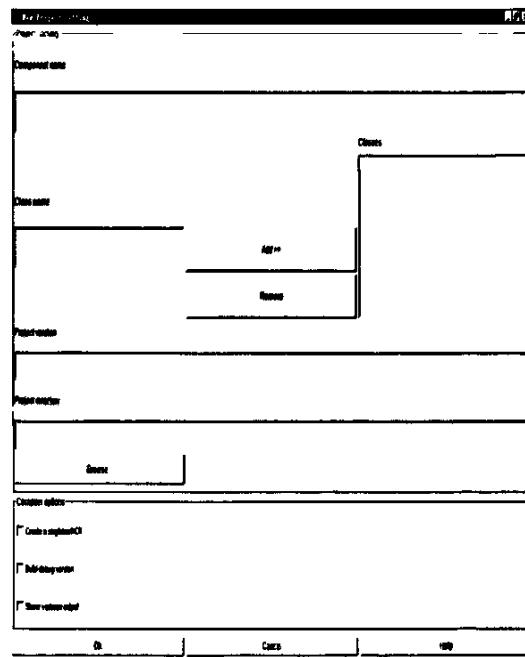


图 2.2 New Project Settings 对话框

选项，Create a singleton MCR 即创建独立运行 MCR (MATLAB Component Runtime)。MCR 是 MATLAB R14 新发布的技术，利用该技术可以将所有发布的应用需要的 MATLAB 语言特性、核心共享函数库等都集成起来。用户在实现算法发布的时候，只要将该组件一同发布给最终用户，则最终用户就可以在自己的计算机上建立 MATLAB 运行环境，使用发布的算法。MCR 替代了早期对话框版本的 MATLAB C/C++ Math Library 和 MATLAB C/C++ Graphic Library，并且提供了对所有 M 语言特性的支持。Build debug version 即创建具有调试信息的 COM 组件，如果开发的 COM 组件要参与应用程序的调试，则需要选择该选项。Show verbose output 即显示详细的信息，如果选择该选项，则所有编译过程的信息将显示在 COM Builder 工具的编译器状态信息栏中。完成设置后点击 OK，系统会出现对话框，询问是否创建工程目录，点击确定，回到 COM Builder 窗口。

§ 2.2.2.2 文件管理

创建工程后，生成器主窗口中的“Project”，“Build”和“Component”等 3 个菜单就变为可用。单击“Add File”按钮向工程中添加 M 文件或 MEX 文件，注意只能是函数文件，而不能是脚本文件，否则将无法编译。单击“Remove”按钮可以删除选定的 M 文件或 MEX 文件。选择 M 文件后单击“Edit”按钮或双击 M 文件名可以在 MATLAB 编辑器中打开该 M 文件并进行编辑和调试，但不能编辑 MEX 文件。

§ 2.2.2.3 组件生成

点击 Build->COM Object，此时 COM Builder 开始自动编译 COM 组件，生成所需的头文件、源文件、接口描述文件、动态链接库文件等，在右侧的 Build Status 显示框中显示编译的过程和信息。点击菜单 Component->Component Info 可以查找关于接口、类、库的信息。中间源文件写到“Project dir”\src 目录中，生成的 DLL 文件写到“Project dir”\distrib 目录中，并自动注册到系统。

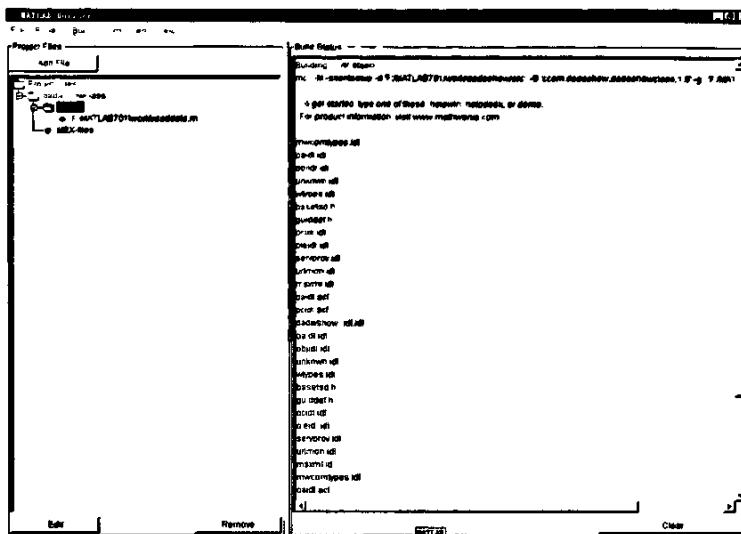


图 2.3 显示编译信息的 COM Builder 窗口

§ 2.2.2.4 打包分发

点击 Component->Package

Component 可以创建自解压可执行程序, 将文件打包并分发给终端用户。可以通过 Add File 按钮将需要打包的文件添加到发布包中。MATLAB R14 使用 MCR 作为最终的可执行应用程序的 MATLAB 运行环境, 因此用户要运行组件必须拥有 MCR。如果用户没有 MCR, 则需要将 MCR 一并打包进行发布, 即将 MATLAB Complier 提供的 MCRInstaller.exe 文件一并进行发布。

给用户进行安装，因此要选择将 Package Files 对话框中的 Include MCR 复选框打勾。MCRinstaller.exe 文件的位置也可由用户自行指定，点击 MCR Location 按钮，可在弹出的对话框中选择路径。设置完毕后点击 Create 按钮，COM Builder 将自动创建可自解压缩的文件，最终用户只要得到这个可执行文件，就可以解压缩出打包的 CTF 文件和 DLL 文件，同时还将完成安装 MCR，注册 COM 组件等工作。如果使用 MATLAB R13，用户只要直接执行打包发布的命令，就可

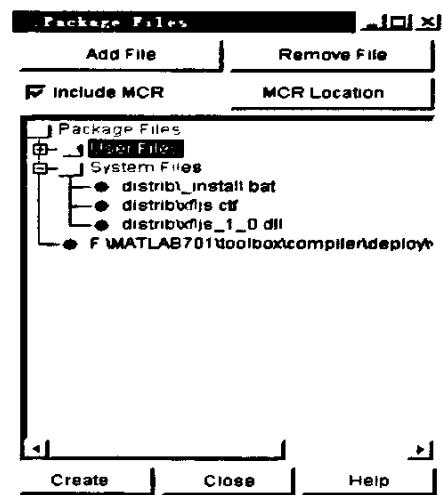


图 2.4 Package Files 对话框

以直接创建自解压缩可执行文件。该文件包括了所有 COM Builder 编译得到的目标文件以及 R13 版本下的 mglinstaller.exe，可以通过这个文件来配置 MATLAB 运行环境。

表 2.1 R13 自解压可执行程序包含的文件

| 文 件 | 功 能 |
|------------------------------------|--------------------|
| _install.bat | 由自解压可执行程序运行的脚本 |
| <componentname_projectversion>.dll | 编译后的组件 |
| Mglinstaller.exe | MATLAB 数学库和图形库安装器 |
| Mwcomutil.dll | COM 生成器工具库 |
| Mwregsvr.exe | 在计算机上注册 DLL 的可执行程序 |

§ 2.3 .NET 下 C# 调用 COM 组件

.NET 对于 COM 的互操作性使用“包装 (wrapper)”的类和“代理”组件。包装的类把依附于另一个系统的类包装起来，从而为不能识别被包装类原有接口的用户提供一个本地接口。同样地，客户可以使用代理组件去访问依附于不同系统的组件或在外地的组件^[32]。.NET 通过 RCW(Runtime Callable Wrapper 运行库可调用的封装器) 实现这种包装和代理。

表 2.2 RCW 使用的接口^[32]

| COM 接口 | 描 述 |
|----------|--|
| IUnknown | 当 .NET 使用早期绑定访问 COM 对象时，RCW 使用该接口。COM 的早期绑定是通过将 COM 类型库导出到 .NET 装配件中然后像普通 .NET 类型一样访问这些装配件类型实现的。当从这些装配件中一个在类型上调用成员时，RCW 就决定成 |

| | |
|---|--|
| | 员所属的接口。如果接口没有缓存到 RCW 内部接口表，RCW 调用 IUnknown-QueryInterface，传递 COM 接口的名称。如果接口存在，则调用 IUnknown-AddRef。如果接口不存在，会向客户产生一个异常 |
| Idispatch | .NET 客户使用后期绑定来访问 COM 对象的成员时，RCW 就使用这一接口。与 COM 对象的后期绑定是在 .NET 中通过所谓的映像（Reflection）技术来完成的 |
| ISupportErrorInfo 和 IerrorInfo | 如果 COM 对象实现这些接口，当 COM 方法返回一个失败的 HRESULT 时，RCW 就用它们获取与错误有关的扩展信息。RCW 把由这些接口提供的信息映射到那些映射至 .NET 客户的例外 |
| IConnectiionPoint 和 IConnectionPointContainer | 这些接口在 COM 中使用以支持 COM 事件属性。RCW 用这些接口将 COM 事件映射到 .NET 事件 |

§2.3.1 添加 COM 引用

在 MS. NET 下可以通过引用对话框添加 COM 组件。^[34]在解决方案资源管理器窗口中找到引用选项，鼠标右键点击添加引用项目，打开添加引用对话框。

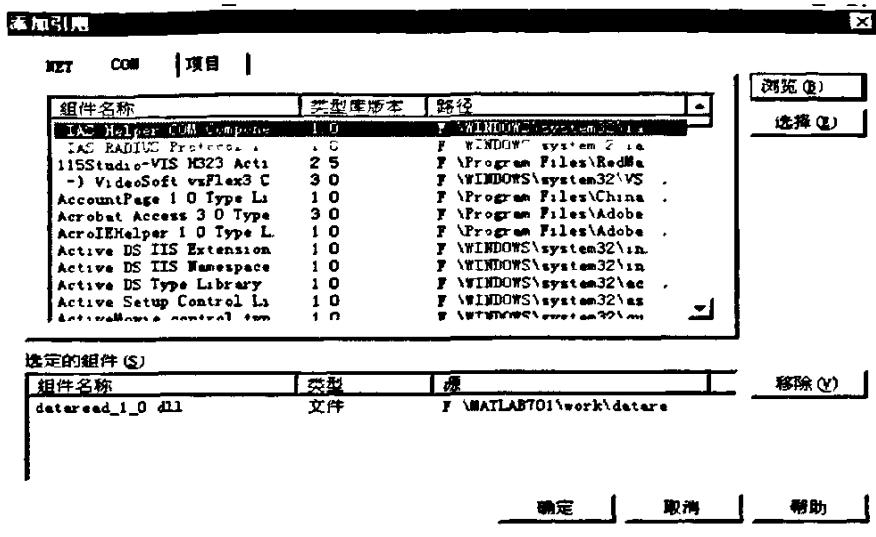
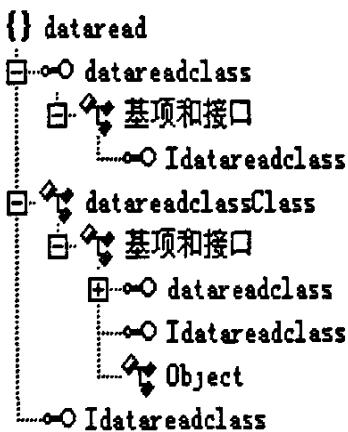


图 2.5 添加引用对话框

这个对话框有三个选项卡：第一个选项卡列出了一些 DLLs，这些 DLLs 对于.NET 运行时环境是至关重要和有补充作用的。最右边的选项卡是项目，它列出的是在当前解决方案中所引用的所有的.NET 工程。中间的选项卡是 COM，当我们要在.NET 工程中引入 COM 组件使用时，就需要使用这个选项卡。点窗口右边的浏览按钮，在弹出的对话框中查找到使用 MATLAB COM Builder 编译好的 COM DLL（这个 dll 文件保存在上文中提到的 distrib 文件夹中），并将其添加到 COM 选项卡上的组件列表当中（如上图所示）。在完成添加后，就可以在.NET 代码中使用 COM 组件了。MS.NET 创建的名称空间与原来的 COM 组件的同样的名称，由 COM 组件提供的类就放在该名称空间中。例如我们添加一个名为 dataread 的 COM 组件，打开对象浏览器，可以看到系统自动生成一个名为 interop.dataread 的装配件，其下包含命名空间、基项和接口等。这样，就可以使用与引用、实例化和调用 C# 固有对象相同的语言来创建一个引用，并实例化和调用 COM 对象的包装。



§2.3.2 数据交换与实例化

使用 MATLAB 作为 COM 端主要是利用其强大的数值计算功能,因此必须进行有效的数据及相关信息传递。MATLAB 用于数据交换的基本类型是字符型和双精度型。可以将 MATLAB COM 组件的调用定义为带引用型参数的方法从而实现 COM 的调用和数据的传递。以下 C#语句实现将打开的标准 COMTRADE 数据文件名字字符串传递给 MATLAB, 经数值处理后传回各相电压和电流数据。

```
public double[,]time,va,vb,vc,v0,
ia,ib,ic,i0;
.....
dataread.datareadclassClass dataArray=new dataread.datareadclassClass();
object time=null,va=null,vb=null,vc=null,v0=null,
ia=null,ib=null,ic=null,i0=null;
string b = file_name.Replace("CFG","DAT");
dataArray.readdata(9,ref time,ref va,ref vb,ref vc,ref v0,ref ia,ref ib,ref ic,ref i0,b);
```

MATLAB 生成的数据文件是双精度类型的二维数组,因此必须在 C#中作相应的定义和实例化。

§2.4 图形用户界面的设计

良好的图形用户界面能够为用户提供简明、直观的操作。.NET 提供了创建 WindowsForm 应用程序的可视化图形开发环境,同时也有许多基于.NET 的第三方控件可供选择。本软件图形界面主要包括菜单、文本框、状态栏、图形显示区等。

§2.4.1 事件驱动程序设计的概念

Windows Form 应用程序通过事件驱动来获得键盘、鼠标或其它设备的输入。事件驱动(Event Driven)指的是窗口应用程序运作的方式。所有的 Windows 应用程序都有一个公共的图形用户界面(GUI),多个 Windows 应用程序将会共享相同的硬件,由于它们的并发特性,一个程序在另外一个应用程序载入和运行之前执行和终止是不可能的。^[34]为此可以设计一个响应事件的应用程序,只有在特定的事件(Event)发生时,程序才会做出对应的处理,当处理完定义的相应程序内容后又将回到闲置状态。这里的事件通常包括键盘的输入或鼠标

点击、移动、按下左右键等。

§2.4.2 菜单与下拉菜单

菜单主要包括【文件】【历史记录】【波形分析】【工具】【帮助】等选项，并设置相应的下拉式菜单。

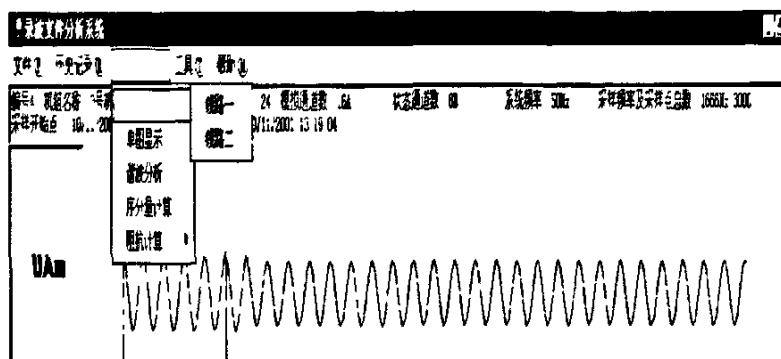


图 2.6 菜单工具栏

文件选项可以实现对录入文件的读取和保存，历史记录选项实现对数据库内保存的历史数据的操作，波形分析选项可以打开相应的分析功能。

§2.4.3 文本框

在文本框内显示读入的录波数据的相关信息，主要包括机组名称、录波通道、系统频率、采样频率以及采样开始和结束时间等。

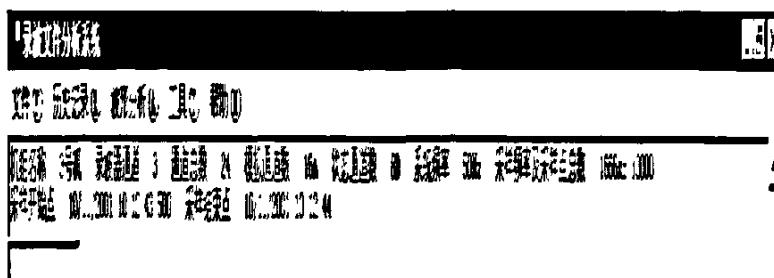


图 2.7 文本框

§2.4.4 信息框

信息框（Message Box）的目的在于提供应用程序与用户之间一种沟通的方法，可以通过它来告知用户应用程序运行的状态、错误信息、请用户再次确认等。如本系统中使用的信息框包括出错信息、要求确认信息等。



图 2.8 信息框

§2.4.5 状态栏

状态栏主要显示文件信息、线路信息以及红绿游标的采样点时刻。当打开文件时，文件信息栏显示“正在打开文件”，打开完毕显示“就绪”；打开波形时线路信息栏显示线路名称；红绿游标拖动时状态栏分别显示游标线所处的采样点时刻。

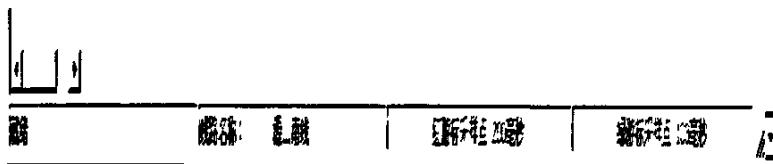


图 2.9 状态栏

§2.4.6 图形显示区

图形显示区是图形界面最重要的部分。为使各通道输出的波形能够在同一坐标平面显示而不至于重叠，系统自动计算输入数据的最大值，然后按照显示比例进行统一调整。在坐标横轴设置了放大功能和滚动条，可以对波形按比例进行放大并拖动滚动条观察波形细部。红绿游标可以任意进行拖动以准确观察采样点时刻，当鼠标距离游标线三个光标点内按下时，鼠标变为手型，拖动鼠标时状态栏显示当前的采样点时刻。

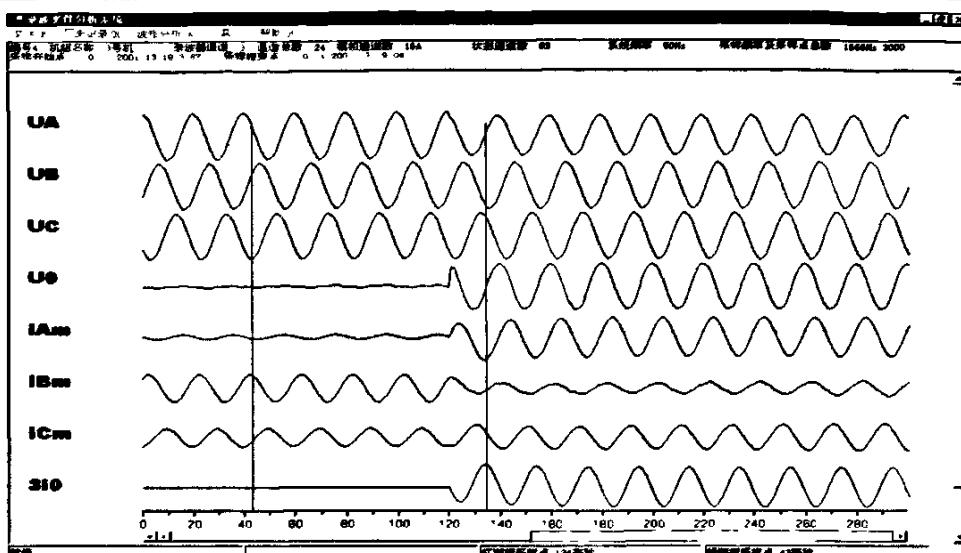


图 2.10 波形显示主页面

由于对同一坐标轴上显示的各通道波形数据进行了处理，所以并不能准确地反映出各个波形的幅值及变化，因此专门设计了单个波形显示页面，以准确地显示各个波形的数据。可以对波形进行放大，同时鼠标经过波形上方时自动显示所在点的坐标值。

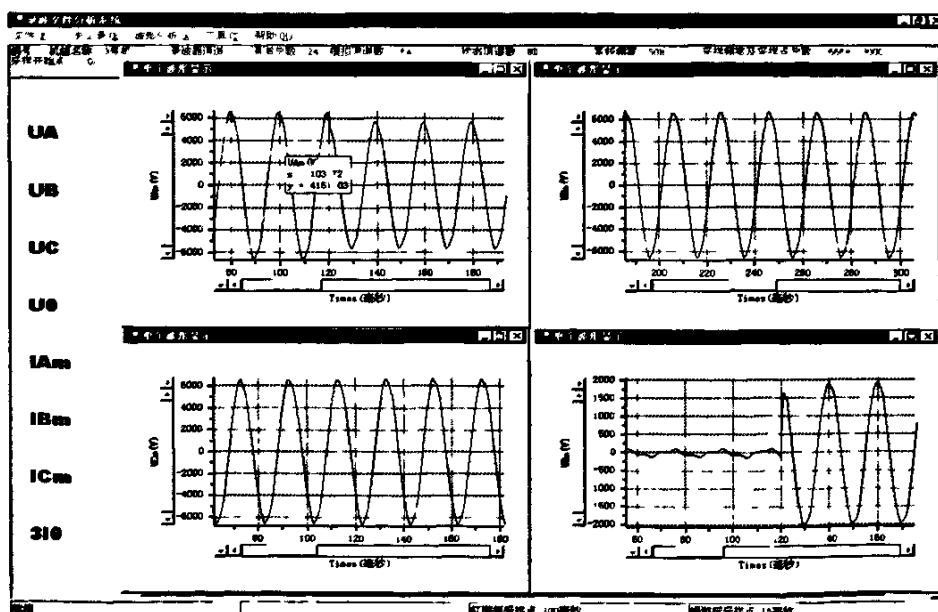


图 2.11 单个波形显示

§2.4.7 多窗体设计及数据传递

为能更加方便地调用和显示不同的功能, 可以使用 MDI (多文档界面) 技术, 在主窗口的基础上添加多个不同的子窗口, 以响应主窗口上的不同的事件。为此, 将主窗口定义为 MDI 容器, 添加以下代码:

```
this.IsMdiContainer = true;
```

子窗口可以继承父窗口的特性, 同时也可以拥有自己的控件和不同的事件。为了能够方便地进行数据计算和处理, 有时需要在主窗口和子窗口之间传递数据。主窗口向子窗口传递数据很简单, 例如, 定义一个子窗口为 Form5, 实现序分量计算的功能, 代码如下:

```
Form5 f5; //定义一个子窗口
//指定向子窗口传递的数据
f5 = new Form5(currentTime, currentPoint, time, Va, Vb, Vc, la, lb, lc, n);
f5.MdiParent = this.MdiParent; //指定父窗口容器为当前窗口
f5.TopMost = true; //使Form5窗口始终显示在最上方
f5.Show(); //显示指定窗口
```

在子窗口接收数据只需实例化即可:

```
public Form5(int currentTime, int currentPoint, double[] time, double[] Va,
double[] Vb, double[] Vc, double[] la, double[] lb, double[] lc, int n)
```

从子窗口向父窗口传递数据要复杂一些, 可以使用委托和事件来完成^[4]。委托可以看作是对象的一种新类型, 当要把方法传递给方法时, 需要使用它们。这里的方法带有参数, 用于传递数据并对数据进行一定的操作。定义委托的语法如下:

```
Delegate void VoidOperation(uint X);
```

这里定义了一个委托 VoidOperation, 并指定该委托的每个样例都包含一个方法的细节, 该方法带有一个 unit 参数, 并返回 void。

事件是应用程序接收某个事件发生的通知的方式。在申明事件之前, 首先要声明该事件的委托类型。在定义完委托类型后, 再根据委托类型申明事件本身。在 C# 中, 事件实际上是委托的一种特殊形式。

例如, 同样在 Form5 子窗口, 需要将故障起动和故障选相后的参数传递回主窗口, 则定义一个委托事件, 其代码如下:

```
public delegate void SendData(int k,int k1,int k2,int k3,int k4,int k5,int k6,int k7,int pi);
//定义委托类型
public event SendData Send; //定义一个事件类型
.....
Send(k,k1,k2,k3,k4,k5,k6,k7,pi); //指定事件传递的参数
```

同样在主窗口中也需要定义这个事件：

```
f5.Send +=new 录波文件分析系统.Form5.SendData(f5_Send);
```

其中 SendData 是定义的委托类型，并指定这个委托事件是位于 Form5 中。

现在就可以在主窗口中接收传递的参数并将其实例化和使用：

```
private void f5_Send(int k, int k1, int k2, int k3, int k4, int k5, int k6, int k7,int pi)
{
    ku=k;k1u=k1;k2u=k2;k3u=k3;k4u=k4;k5u=k5;k6u=k6;k7u=k7;p=pi;
}
```

第三章 数据库操作与应用

系统采用 SQL Server 2000 进行数据库设计。SQL (Structured Query Language) 是目前使用最为广泛的通用关系数据库查询语言。.NET 框架提供 ADO.NET 实现对数据的访问。ADO.NET 是由类、接口、枚举和其它工具所组成的库，这些工具为数据访问提供了结构化的、功能强大的、高效的工具包。

§3.1 标准 COMTRADE 文件的读取

§3.1.1 COMTRADE 格式简介^[40]

COMTRADE 的全称是 IEEE Standard Common Format for Transient Data Exchange for Power Systems，即电力系统暂态数据交换标准通用格式，由电气和电子工程师学会电力系统协会继电保护委员会制定。

IEEE COMTRADE 定义了一种通用标准的文件格式，用于存储从电力系统继电保护装置、故障录波器或暂态仿真模型获得的暂态数据和相关信息，其目的在于为交换数据提供标准的、方便解读的格式。

标准中包括三种类型的相关文件：

- 引导文件 (.HDR)。它是由故障数据源生成的字处理文件，为用户提供一个附加的描述文本，以更好地了解暂态记录的条件，为分析软件提供一些必要的参数。
- 组态文件 (.CFG)。它以预先定义的固定格式为分析软件提供暂态数据的相关信息，如通道信息、系统频率、采样点数和采样频率、采样起始时间等。
- 数据文件 (.DAT)。它以 ASCII 码格式保存各个采样通道采集的模拟量和开关量的具体数据值。数据同 CFG 文件中定义的格式完全一致，以方便程序读取。

§3.1.2 正则表达式应用

正则表达式 (regular expression, 简写为 regexes) 是用来操作和检验字符串数据的一种强大的工具。正则表达式是一串特殊的字符，它转换为某种算法，根据这个算法来匹配文本^[41]。因为 COMTRADE 具有标准的格式，所以应用正

则表达式来定义查找文件中字符串的数据就显得十分的方便和快捷。

C#中正则表达式的命名空间是 `System.Text.RegularExpressions`，包含下述八个类：

- `Regex`——包含了正则表达式，以及使用正则表达式的方法
- `MatchCollection`——包含了一个正则表达式找到的所有匹配项
- `Match`——包含了一次匹配中所有匹配的文本。
- `GroupCollection`——包含了一次匹配中所有的分组
- `Group`——包含了一个分组集合中一个组的细节
- `CaptureCollection`——包含一个组的所有捕获对象
- `Capture`——返回组内一次捕获所匹配的字符串
- `RegexCompilationInfo`——提供了把正则表达式编译为一个独立程序集所需的细节

表 3.1 以正则表达式语法表示的字符串

| 字符串 | 匹配的字符 |
|---------------------|---------------------------------|
| <code>\d</code> | 从 0~9 的任一数字 |
| <code>\D</code> | 非数字字符 |
| <code>\w</code> | 任一单词字符，如 A-Z、a-z、0-9，和下划线字符 |
| <code>\W</code> | 非单词字符 |
| <code>\s</code> | 任一空白字符，包括制表符、换行符、回车符、换页符和垂直制表符等 |
| <code>\S</code> | 任一非空白字符 |
| <code>.</code> | 任一字符 |
| <code>[...]</code> | 括号中的任一字符 |
| <code>[^...]</code> | 非括号中的任一字符 |

例如，CFG 文件中采样开始时间字符串为：

“10/12/2001, 10:41:42.000”

正则表达式定义为：

`(\d+)/(\d+)/(\d+), (\d+):(\d+):(\d+).(\d+)`

以下代码实现对此段字符串的捕获：

```

//定义正则表达式
string patternST = @"(\d+)(\d+)(\d+),(\d+):(\d+):(\d+).(\d+)";
string str08 = null, str18 = null;
//定义字符串匹配
MatchCollection Matches8 =
Regex.Matches(AllText, patternST, RegexOptions.IgnoreCase
| RegexOptions.ExplicitCapture);
foreach (Match NextMatch in Matches8) //循环查找
str08 = NextMatch.ToString();
str18 = str08.Replace(",", " ");
strtext8 = str18;

```

§3.1.3 ADO.NET 操作数据库

ADO.NET 库有两个核心的组件^[42]：DataSet 和.NET 数据提供者。DataSet 是非连接的、位于内存中的数据存储；而.NET 数据提供者是一套特有的组件，用于访问特殊类型的数据源，例如 OLEDB 数据源或 SQL Server 数据库等。

ADO.NET 体系结构

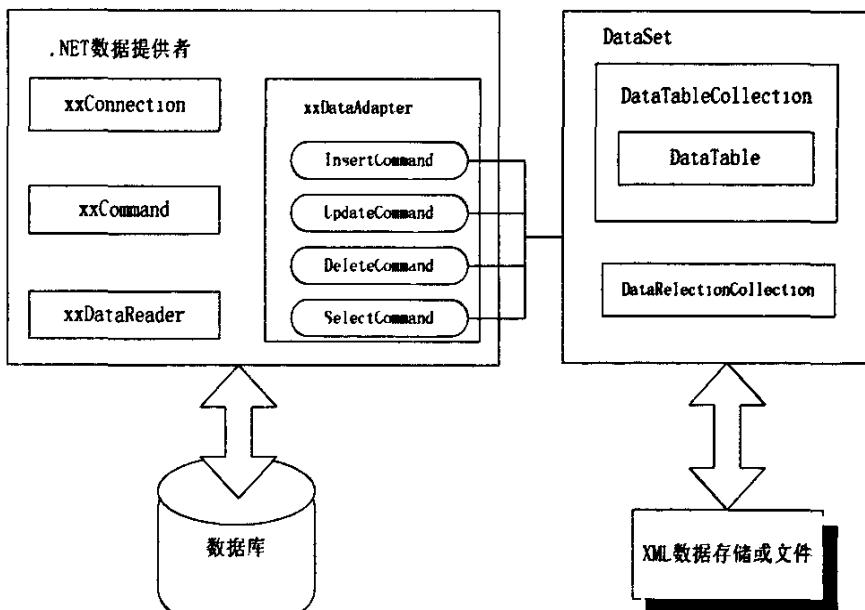


图 3.1 ADO.NET 体系结构

Microsoft SQL Server 的.NET 数据提供者定义为 Microsoft SQL Client，它使用快速的、本地的 TCP/IP 接口快速、健壮和可靠地访问 SQL Server 数据库。其命名空间为 System.Data.SqlClient，核心类包括：sqlConnection、sqlDataReader、sqlDataAdapter 和 sqlCommand。通过这个工具，我们可以在 C#语言中直接使用 SQL 语句来访问和操作 SQL Server 数据库。

§ 3.1.4.1 数据库的连接

SQL Server.NET Data Provider 提供了对 SQL Server7 和 2000 的优化访问。为了连接 SQL Server 必须实例化一个 SqlConnection 对象，并调用此对象的方法。当不再需要连接时，应当调用这个对象的 Close 方法关闭连接。SqlConnection 类定义两个构造函数，一个带参数，另一个接收连接字符串，可以使用两种方法中的任何一种实例化连接：

```
SqlConnection cn=new SqlConnection(ConnectionString);
cn.Open();
```

或：

```
SqlConnection cn = new SqlConnection();
cn.ConnectionString = ConnectionString;
cn.Open();
```

当使用 SQL Server 身份验证时，必须在连接字符串中显式地指定用户 ID 和密码。最后不要忘记关闭连接以节省服务器资源。

§ 3.1.4.2 执行数据库命令

ADO.NET 定义 Command 类去执行对数据库的访问和操作。

SqlCommand cmd=new SqlCommand(CommandText,ConnectionObject)
CommandText 是所要执行的命令文本，可以使用 SQL 定义的 Transact 语句编写。ConnectionObject 指数据库连接对象。SqlCommand 对象提供了 4 个执行方法：

- ExecuteNonQuery 不返回任何结果
- ExecuteScalar 返回单一值的结果
- ExecuteReader 使用结果填充 DataReader 对象
- ExecuteXmlReader 返回一个 XML 字符串的命令

§ 3.1.4.3 数据的读取

可以使用两种方法提取数据，第一是使用数据阅读器，第二是同时使用数据适配器和 DataSet。^[43]ADO.NET 的数据阅读器定义为 DataReader 类，使用与它相应的 Command 类从底层的数据存储上检索数据。以下代码使用数据阅读器实现从数据库中读取文件信息功能：

```

SqlCommand ch = new SqlCommand(selectID,cn); //定义数据库命令
ch.CommandType = CommandType.Text; //定义命令类型
SqlDataReader dr = ch.ExecuteReader(); //创建DataReader对象
dr.Read(); //打开阅读器
strtext0 = dr["ID"].ToString(); //使用提取数据集赋值
.....
strtext9 = dr["EndTime"].ToString();
//将文本显示在文本框中
textBox1.Text = "编号" + strtext0 + " 机组名称 " + strtext1 + " 录波器通道 " + strtext2 + " 通道总数 " + strtext3 + " 模拟通道数 " + strtext4 + " 状态通道数 " + strtext5 + " 系统频率 " + strtext6 + " 采样频率及采样点总数 " + strtext7 + "\n" + "采样开始点 " + strtext8 + " 采样结束点 " + strtext9;
dr.Close(); //关闭阅读器。(数据阅读器打开时，不能使用连接执行其它任务)
cn.Close(); //关闭连接

```

DataAdapter 的作用相当于内存绑定的 DataSet 对象和数据存储之间的通道。使用 DataAdapter 读取数据的步骤为：

- 创建一个 DataAdapter 去连接数据库
- 调用 DataAdapter 的 Fill 方法去执行 SQL SELECT 查询，然后用查询的结果填充 DataSet
- 从 DataSet 上访问所返回的数据

§3.1.4 文件的打开和存放

点击菜单中文件→打开选项，系统通过对话框打开 CFG 文件，读取其中信息后显示在文本框内，同时打开对应的 DAT 文件，读取其中的录波数据。如果需要保存数据信息，则点击菜单中文件→保存选项，系统自动在数据库

file_info 表中添加一行，以保存录波信息文件。

| ID | SetName | RecordID | AllchannelNumber | AnalogNumber | SwitchNumber | SystemRate | SamRatAndNu | StartTime | EndTime |
|----|---------|----------|------------------|--------------|--------------|------------|--------------|-----------------|-----------------|
| 1 | 34机 | 3 | 25 | 16A | 100 | 50Hz | 1666Hz;5000 | 10/12/2001 10:4 | 10/12/2001 10:4 |
| 2 | 34机 | 3 | 25 | 16A | 100 | 50Hz | 5000Hz;15000 | 10/12/2001 11:2 | 10/12/2001 11:2 |
| 3 | 34机 | 3 | 24 | 16A | 80 | 50Hz | 1666Hz;7500 | 10/10/2001 10:0 | 10/10/2001 10:0 |
| 4 | 34机 | 3 | 24 | 16A | 80 | 50Hz | 1666Hz;3000 | 10/11/2001 13:1 | 10/11/2001 13:1 |
| 5 | 34机 | 3 | 24 | 16A | 80 | 50Hz | 1666Hz;7500 | 10/10/2001 10:0 | 10/10/2001 10:0 |
| 6 | 34机 | 3 | 24 | 16A | 80 | 50Hz | 1666Hz;10000 | 10/10/2001 10:2 | 10/10/2001 10:2 |
| 7 | | | | | | | | | |

图 3.2 file_info 数据表

相关 C#语句如下：

```

string source = "server=localhost;uid=sa;pwd=sa;database=录波文件";
SqlConnection cn = new SqlConnection(source); //连接数据库
cn.Open(); //打开数据库连接
string addinfo = @"INSERT file_info(SetName,RecordID,AllchannelNumber,AnalogNumber,
SwitchNumber,SystemRate,SamRatAndNu,StartTime,EndTime)values
(@SetName,@RecordID,@AllchannelNumber,@AnalogNumber,@SwitchNumber,
@SystemRate,@SamRatAndNu,@StartTime,@EndTime)"; //存储命令文本
SqlCommand cm = new SqlCommand(addinfo,cn); //定义SQL命令
cm.CommandType = CommandType.Text; //定义命令类型
cm.Parameters.Add(new SqlParameter("@SetName",SqlDbType.Char,10)); //定义数据参数
cm.Parameters.Add(new SqlParameter("@RecordID",SqlDbType.Int));
.....
cm.Parameters.Add(new SqlParameter("@EndTime",SqlDbType.VarChar,50));
cm.Parameters["@SetName"].Value = strtext1; //为数据赋值
cm.Parameters["@RecordID"].Value = strtext2;
.....
cm.Parameters["@EndTime"].Value = strtext9;
cm.ExecuteNonQuery(); //指定无返回值
同时系统自动建立一个新的数据存储表单 data_**
```

| ID | IAs | IBs | ICs | IDs | IAs | IBs | ICs | IDr | IBs | ICs | IDr |
|-------|-------|-------|-------|------|-----|-----|-----|-----|-------|------|-----|
| 1 | 570 | -545 | -208 | -114 | 10 | 42 | -64 | -13 | 306 | -208 | |
| 200 | 6336 | -6337 | 1450 | -117 | 2 | 50 | -64 | -16 | 3365 | -254 | |
| 1200 | 6522 | 3832 | -2550 | -117 | -10 | 28 | -26 | -16 | 3477 | -201 | |
| 1800 | 6358 | -2613 | -3733 | -114 | -21 | 64 | -48 | -16 | 3421 | -144 | |
| 2400 | 8085 | -1386 | -645 | -114 | -24 | 56 | -27 | -13 | 3298 | -708 | |
| 3000 | 5712 | -210 | -5451 | -120 | -45 | 55 | -23 | -10 | 305 | -145 | |
| 3500 | 5232 | 938 | -6144 | -120 | -56 | 65 | -16 | -8 | 2789 | 458 | |
| 4200 | 4416 | 2160 | -6554 | -114 | -64 | 61 | -5 | -6 | 2328 | 1117 | |
| 4800 | 3298 | 3325 | -6597 | -114 | -72 | 53 | 5 | -8 | 1792 | 1714 | |
| 5400 | 2082 | 4320 | -6366 | -117 | -74 | 45 | 18 | -6 | 1130 | 2242 | |
| 6000 | 938 | 5114 | -6000 | -117 | -80 | 28 | 32 | -8 | 504 | 2574 | |
| 6600 | -236 | 5958 | -5539 | -120 | -80 | 16 | 46 | 0 | -114 | 3042 | |
| 7200 | -1509 | 8528 | -6929 | -117 | -80 | 5 | 50 | 0 | 776 | 3365 | |
| 7800 | -2577 | 8730 | -3960 | -120 | -72 | -10 | 61 | 0 | 1381 | 3465 | |
| 8400 | -3738 | 6621 | -2764 | -120 | -46 | -21 | 64 | 0 | 11955 | 3444 | |
| 9000 | -6442 | 6301 | -1568 | -114 | -46 | -40 | 66 | 0 | 2437 | 3238 | |
| 9600 | -5432 | 5998 | -361 | -117 | -46 | -53 | 22 | 5 | 2840 | 3082 | |
| 10200 | -6130 | 5365 | 827 | -117 | -37 | -64 | 22 | 5 | 3168 | 2775 | |
| 2800 | -5576 | 4861 | 1865 | -120 | -21 | -72 | 22 | 6 | 3465 | 2362 | |
| 400 | -6560 | 3597 | 3141 | -117 | -10 | -80 | 24 | 6 | 3472 | 1821 | |
| 12000 | -6463 | 2330 | 4173 | -120 | 2 | -95 | 25 | 0 | 3378 | 1104 | |

图 3.3 录波数据存储表单

相关 C#语句如下：

```

SqlCommand ce = new SqlCommand(@"SELECT Max(ID) FROM file_info",cn);
//在file_info表单中查找主键
int MaxID = (int)ce.ExecuteScalar(); //返回主键值
//创建一个空表
string cdtext = @"CREATE Table 录波文件.dbo.data_
(Times float NULL,UAs float  NULL,UBs float  NULL,UCs float  NULL,
U0s float  NULL,IAs float  NULL,IBs float  NULL,ICs float  NULL,
I0s float  NULL,UAr float  NULL,UBr float  NULL,UCr float  NULL,
U0r float  NULL,IAr float  NULL,IBr float  NULL,ICr float  NULL,
I0r float  NULL)
Use 录波文件";
//使创建的新表与信息文件相对应
string cdnewtext = cdtext.Replace("data_","data_"+MaxID.ToString());
SqlCommand cd = new SqlCommand(cdnewtext,cn);
cd.CommandType = CommandType.Text;
cd.ExecuteNonQuery();
//将数据存入表单
String dataadd = @"insert into data_(Times,UAs,UBs,UCs,U0s,IAs,IBs,ICs,I0s,

```

```

UAr,UBr,UCr,U0r,lAr,lBr,ICr,l0r)values(@Times,@UAs,@UBs,@UCs,@U0s,@lAs,@lBs,@l
Cs,@l0s,@UAr,@UBr,@UCr,@U0r,@lAr,@lBr,@lCr,@l0r");//定义存储命令文本
string dataaddnew = dataadd.Replace("data_","data_"+MaxID.ToString());
SqlCommand cf = new SqlCommand(dataaddnew,cn);
cf.CommandType = CommandType.Text;
cf.Parameters.Add(new SqlParameter("@Times",SqlDbType.Float));//定义数据参数
.....
cf.Parameters.Add(new SqlParameter("@l0r",SqlDbType.Float));
for(int i=1;i<=Time.Length;i++) //循环赋值
{
cf.Parameters["@Times"].Value = Time[i,1];
.....
cf.Parameters["@l0r"].Value = l0r[i,1];
cf.ExecuteNonQuery();
}
cn.Close(); //关闭数据库连接

```

§3.2 历史数据的查询与操作

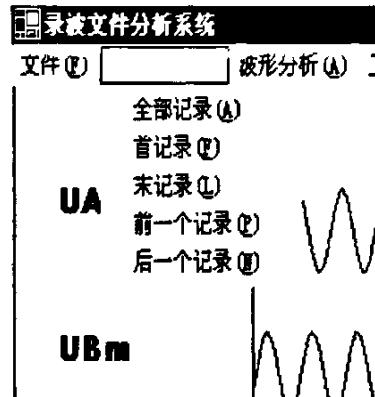
系统提供多种方式实现对历史数据的查询。打开历史数据下拉菜单，可以直接选择首记录或末记录，可以通过前翻或后退的方式查询记录，也可以选择打开全部记录表选择所要查询的记录数据。

利用 DataGrid 控件显示全部历史记录，DataGrid 采用数据网格绑定数据集，以行和列的方式来显示数据。可以任意选择查看或删除数据库中历史数据。在这里使用前述的数据适配器 DataAdapter 和 DataSet 来读取数据并进行绑定：

```

string source = "server=localhost;uid=sa;pwd=sa;database=录波文件";
SqlConnection cn = new SqlConnection(source); //定义数据库连接
cn.Open(); //打开连接

```



```

//定义数据适配器
SqlDataAdapter da = new SqlDataAdapter(@"SELECT
ID AS[记录编号],
SetName AS[机组名],
RecordID AS[录波器编号],
StartTime AS[开始时间],
EndTime AS[结束时间]
FROM  file_info",cn);
DataSet ds = new DataSet(); //定义DataSet
da.Fill(ds); //使用Fill方法填充DataSet
dataGrid1.DataSource = ds.Tables[0]; //指定DataGrid数据源并绑定数据集
cn.Close(); //关闭数据库连接

```

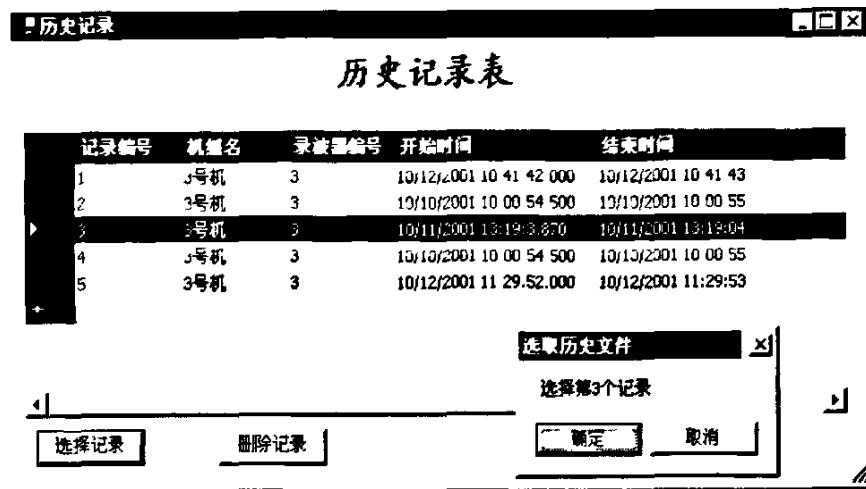


图 3.4 历史记录表页面

为使能够按行选择数据集，定义了一个 DataGridSelectedItems 的组件，其代码如下：

```

public class DataGridSelectedItems : DataGrid //指定继承类
{
    public IList SelectedItems //定义接口
    {

```

```

get
{
    CurrencyManager cm =
        (CurrencyManager) BindingContext[ DataSource ]; //指定数据源
    DataView dataView = (DataView) cm.List; //使用DataView获得列表
    ArrayList rows = new ArrayList(); //定义动态数组实现IList接口
    //获得列表值
    for ( int row = 0; row < dataView.Count; ++row )
        if ( IsSelected( row ) )
            rows.Add( row+1 );
    return rows;
}
}
}

```

最后返回选中行的数据集。

§ 3.3 添加线路参数

在后面的计算中将会用到一些线路参数。系统采用对话窗口的形式自动添加线路信息，并保存至数据库中。这里只列出了一些最基本的参数，若要添加其它的数据可以照此增加。选择线路时同样使用 DataGrid 数据网格，可以按行选择所要计算的线路参数。选定的线路参数同样传回主页面，使用委托事件的方法实现了页面向主页传递数据。

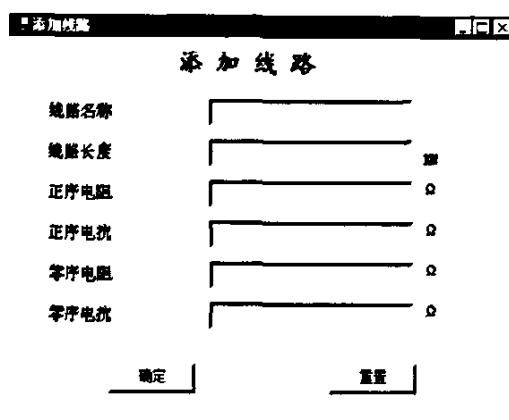
//定义委托事件

```

public delegate void SendData(string lineName, float length, float r1, float
x1, float r0, float x0);
public event SendData Send;

```

然后将从数据库中读取的数据封装在这个事件中：



```
Send(lineName, length, r1, x1, r0, x0);
```

最后将封装的数据传递给主页面：

```
private void f7_Send(string lineName, float length, float r1, float x1, float r0,  
float x0)  
{  
    lineNA=lineName;Len=length.R1=r1;X1=x1;R0=r0;X0=x0;  
}
```

第四章 基本算法及主要分析功能

§4.1 消除衰减直流分量的影响

在实际工程应用中, 傅立叶算法是微机继电保护中广泛采取的算法。但是傅立叶算法只能消除直流分量和整次谐波分量, 而当电网出现故障或扰动时将出现暂态过程, 产生衰减直流分量。由于衰减直流分量是典型的非周期信号, 对应的频谱为连续频谱, 从而与信号中基频分量的频谱产生混迭, 在利用傅立叶算法计算输入信号的基频分量时会产生很大的误差, 因此必须对傅立叶算法加以改进。

§4.1.1 已知衰减时间常数的傅立叶改进算法^[49]

设输入信号为

$$i(t) = I_0 e^{-kt_s/\tau} + \sum_{n=1}^{N/2} I_n \sin(\omega_n t + \varphi_n) \quad (4-1)$$

其中, $\omega_n = n\omega_1$, n 为正整数, 则第 k 次采样值为

$$i_k = I_0 e^{-kt_s/\tau} + \sum_{n=1}^{N/2} I_n \sin\left(\frac{2\pi}{n} nk + \varphi_n\right) = I_0 r^k + i_{ka} \quad (4-2)$$

其中, $r = e^{-kt_s/\tau}$, i_{ka} 为直流分量。交流分量一周波积分为零, 以矩形

积分近似, 可表示为:

$$\sum_{k=1}^N i_{ka} \cdot T_s = 0 \quad (4-3)$$

则有

$$\sum_{k=1}^N i_k \cdot T_s = \sum_{k=1}^N (i_{ka} + I_0 r^k) \cdot T_s = \sum_{k=1}^N I_0 r^k \cdot T_s = I_0 \sum_{k=1}^N r^k \cdot T_s \quad (4-4)$$

可求得

$$I_0 = \frac{\sum_{k=1}^N i_k}{\sum_{k=1}^N r^k} \quad (4-5)$$

因此, 由付氏算法求输入信号的正弦分量时, 有

$$\begin{aligned} I_{s1} &= \frac{2}{N} \sum_{k=1}^N i_k \sin\left(\frac{2\pi}{N} k\right) = \frac{2}{N} \sum_{k=1}^N i_{k0} \sin\left(\frac{2\pi}{N} k\right) + \frac{2}{N} \sum_{k=1}^N I_0 r^k \sin\left(\frac{2\pi}{N} k\right) \\ &= I_{s1(a)} + I_0 \cdot \frac{2}{N} \sum_{k=1}^N r^k \sin\left(\frac{2\pi}{N} k\right) \end{aligned} \quad (4-6)$$

则输入信号中周期信号的正弦分量为

$$\begin{aligned} I_{s1(a)} &= I_{s1} - I_0 \cdot \frac{2}{N} \sum_{k=1}^N r^k \sin\left(\frac{2\pi}{N} k\right) \\ &= I_{s1} - \frac{\sum_{k=1}^N i_k}{\sum_{k=1}^N r^k} \cdot \frac{2}{N} \sum_{k=1}^N r^k \sin\left(\frac{2\pi}{N} k\right) \\ &= I_{s1} - K_s \cdot \sum_{k=1}^N i_k \end{aligned} \quad (4-7)$$

由付氏算法求输入信号的余弦分量时, 有

$$\begin{aligned} I_{c1} &= \frac{2}{N} \sum_{k=1}^N i_k \cos\left(\frac{2\pi}{N} k\right) = \frac{2}{N} \sum_{k=1}^N i_{k0} \cos\left(\frac{2\pi}{N} k\right) + \frac{2}{N} \sum_{k=1}^N I_0 r^k \cos\left(\frac{2\pi}{N} k\right) \\ &= I_{c1(a)} + I_0 \cdot \frac{2}{N} \sum_{k=1}^N r^k \cos\left(\frac{2\pi}{N} k\right) \end{aligned} \quad (4-8)$$

则输入信号中周期信号的余弦分量为

$$\begin{aligned} I_{c1(a)} &= I_{c1} - I_0 \cdot \sum_{k=1}^N r^k \cos\left(\frac{2\pi}{N} k\right) \\ &= I_{c1} - \frac{\sum_{k=1}^N i_k}{\sum_{k=1}^N r^k} \cdot \frac{2}{N} \sum_{k=1}^N r^k \cos\left(\frac{2\pi}{N} k\right) \end{aligned}$$

$$\begin{aligned}
 &= I_{cl} - \frac{\sum_{k=1}^N r^k \cos\left(\frac{2\pi}{N} k\right)}{\frac{N}{2} \sum_{k=1}^N r^k} \cdot \sum_{k=1}^N i_k \quad (4-9) \\
 &= I_{cl} - K_c \cdot \sum_{k=1}^N i_k
 \end{aligned}$$

当衰减时间常数已知时, K_s , K_c 可以预先离线计算, 作为傅立叶算法的补偿系数, 此时傅立叶改进算法有较高的计算精度。但是, 在给出的衰减时间常数与实际的衰减时间常数有较大的差别时, 算法的计算误差也会较大。

§4.1.2 任意衰减时间常数的傅立叶改进算法

若衰减时间常数未知, 则需要使用任意衰减时间常数的傅立叶改进算法。这种算法可以提供较高的计算精度, 但也会增加算法的计算量。

在上一节的计算中, 求得

$$I_0 = \frac{\sum_{k=1}^N i_k}{\sum_{k=1}^N r^k} \quad (4-10)$$

经过一个采样间隔后, 有

$$I_0 e^{-\frac{t}{T}} = I_0 r = \frac{\sum_{k=2}^{N+1} i_k}{\sum_{k=1}^N r^k} \quad (4-11)$$

可求得

$$r = \frac{\sum_{k=2}^{N+1} i_k}{\sum_{k=1}^N i_k} \quad (4-12)$$

令 $K = K_c + jK_s$, 有

$$\begin{aligned}
 K &= \frac{\sum_{k=1}^N r^k \cos\left(\frac{2\pi}{N} k\right)}{\frac{N}{2} \sum_{k=1}^N r^k} + j \frac{\sum_{k=1}^N r^k \sin\left(\frac{2\pi}{N} k\right)}{\frac{N}{2} \sum_{k=1}^N r^k} \\
 &= \frac{1}{\frac{N}{2} \sum_{k=1}^N r^k} \sum_{k=1}^N (re^{\frac{j2\pi}{N}})^k \quad (4-13)
 \end{aligned}$$

其中，

$$\begin{aligned}
 \sum_{k=1}^N (re^{\frac{j2\pi}{N}})^k &= (re^{\frac{j2\pi}{N}}) \frac{1 - (re^{\frac{j2\pi}{N}})^N}{1 - (re^{\frac{j2\pi}{N}})} = \frac{r(1 - r^N)e^{\frac{j2\pi}{N}}}{1 - re^{\frac{j2\pi}{N}}} \\
 &= \frac{r(1 - r^N)(\cos \frac{2\pi}{N} + j \sin \frac{2\pi}{N})}{(1 - r \cos \frac{2\pi}{N}) - j \sin \frac{2\pi}{N}} \quad (4-14) \\
 &= r(1 - r^N) \frac{(\cos \frac{2\pi}{N} - r) + j \sin \frac{2\pi}{N}}{(1 + r^2 - 2r \cdot \cos \frac{2\pi}{N})}
 \end{aligned}$$

从而求得补偿系数为

$$K_s = r(1 - r^N) \frac{\sin \frac{2\pi}{N}}{(1 + r^2 - 2r \cdot \cos \frac{2\pi}{N})} \cdot \frac{1}{\frac{N}{2} \sum_{k=1}^N r^k} \quad (4-15)$$

$$K_c = r(1 - r^N) \frac{\cos \frac{2\pi}{N} - r}{(1 + r^2 - 2r \cdot \cos \frac{2\pi}{N})} \cdot \frac{1}{\frac{N}{2} \sum_{k=1}^N r^k}$$

而

$$\sum_{k=1}^N r^k = r \cdot \frac{1-r^N}{1-r} \quad (4-16)$$

进一步化简可得

$$K_s = \frac{2}{N} (1-r) \frac{\sin \frac{2\pi}{N}}{(1+r^2 - 2r \cdot \cos \frac{2\pi}{N})} \quad (4-17)$$

$$K_c = \frac{2}{N} (1-r) \frac{\cos \frac{2\pi}{N} - r}{(1+r^2 - 2r \cdot \cos \frac{2\pi}{N})}$$

以上推导均假定输入信号只含基频分量、衰减直流分量和各整次谐波分量。实际故障暂态信号中还含有大量的非整次谐波分量，因此用以上傅立叶改进算法计算基频分量时仍会产生一定的计算误差。

§4.1.3 仿真验证

利用 MATLAB 进行仿真实验，建立一个 500KV 输电线路模型，线路参数为：

正序阻抗： $Z_1 = 0.1272 + j0.2932 \Omega/km$

零序阻抗： $Z_0 = 0.3864 + j1.2957 \Omega/km$

线路对地正序电容： $C_1 = 0.01272 \mu F/km$

线路对地零序电容： $C_0 = 0.07751 \mu F/km$

等值系统参数为：

$$Z_M = 0.66 + j16.51 \Omega$$

设置为三相接地故障，接地电阻 100Ω ， $0.04s$ 故障开始， $0.12s$ 故障结束，

采样时间 $T_s = \frac{0.02}{96} s$ ，即每周采样 96 点。

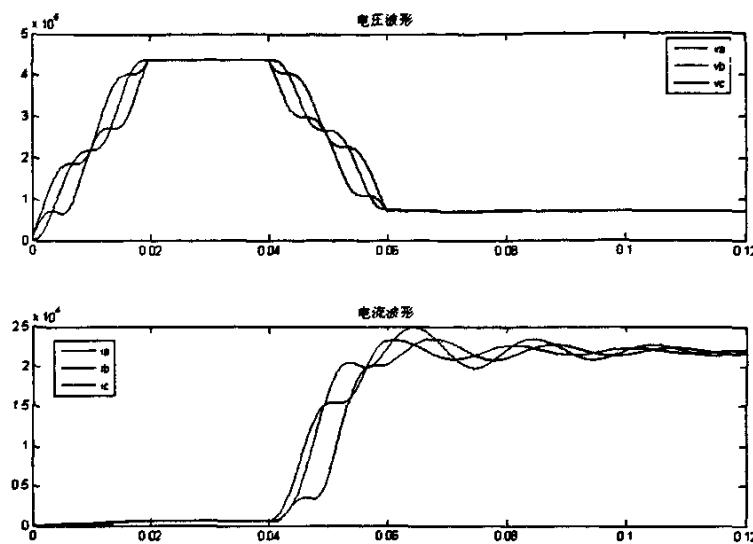


图 4.1 未经修正付氏算法的电压电流波形

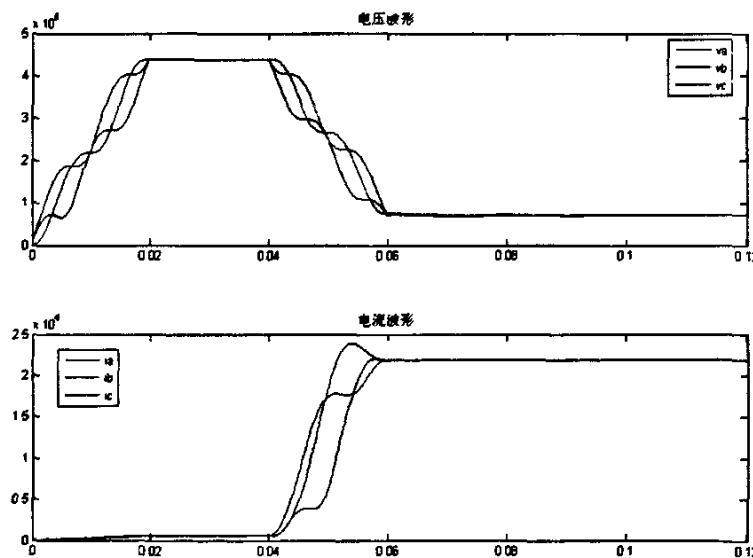


图 4.2 付氏改进算法的电压电流波形（电压未经修正）

由图可见，经过改进算法后的故障电流可以大大减少衰减直流分量的影响，经计算可得未修正的付氏算法电流分量相对误差为 0.1153，而改进算法相对误差为 0.0064，很好地提高了计算精度。

§4.2 计算谐波分量

§4.2.1 FFT 加窗插值修正算法^[50-53]

针对频谱泄漏的影响, 文献[45]提出了一种基于 FFT (快速付立叶变换) 的双峰谱线修正算法, 现引述如下: 假设一个频率为 f_0 幅值为 A 初相位为 θ 单一频率信号 $x(t)$, 在经过了采样率为 f_s 的模数变换后得到如下形式的离散信号:

$$x(n) = A \cdot \sin(2\pi \frac{f_0}{f_s} n + \theta) \quad (4-18)$$

如果所加窗函数的时域形式为 $w(n)$, 其连续频谱为 $W(2\pi f)$, 则加窗后该信号的连续付立叶变换为

$$\begin{aligned} X(f) &= \sum_{n=-\infty}^{\infty} x(n) \cdot w(n) \cdot e^{-j2\pi f n} \\ &= \frac{A}{2j} \left\{ e^{j\theta} W\left(\frac{2\pi(f - f_0)}{f_s}\right) - e^{-j\theta} W\left(\frac{2\pi(f + f_0)}{f_s}\right) \right\} \end{aligned} \quad (4-19)$$

如果忽略负频点 $-f_0$ 处频峰的旁瓣影响, 在正频点 f_0 附近的连续频谱函数可以表达为

$$\bar{X}(f) = \frac{A}{2j} e^{j\theta} W\left(\frac{2\pi(f - f_0)}{f_s}\right) \quad (4-20)$$

对式 (4-20) 进行离散抽样, 即可得到它的离散付立叶变换的表达式为

$$\bar{X}(k \cdot \Delta f) = \left(\frac{A}{2j} \right) e^{j\theta} W\left(2\pi(k \cdot \Delta f - f_0)/f_s\right) \quad (4-21)$$

式中离散频率间隔为 $\Delta f = f_s/N$, N 是数据截断长度。

峰值频率 $f_0 = k_0 \cdot \Delta f$ 很难正好位于离散谱线频点上, 即 k_0 一般不是整数。设峰值点左右两侧的谱线分别为第 k_1 和 k_2 条谱线, 这两条谱线也应该是峰值点附近幅值最大和次最大的谱线。显然, $k_1 \leq k_0 \leq k_2 = k_1 + 1$ 。在离散频谱中找到这两条谱线, 从而可确定 k_1 和 k_2 。令这两条谱线幅值分别是 $y_1 = |\bar{X}(k_1 \cdot \Delta f)|$ 和 $y_2 = |\bar{X}(k_2 \cdot \Delta f)|$, 则由 (4-21) 式可得

$$\frac{y_1}{y_2} = \left[\frac{W(2\pi \cdot (k_1 - k_0)/N)}{W(2\pi \cdot (k_2 - k_0)/N)} \right] \quad (4-22)$$

由于 $0 \leq k_0 - k_1 \leq 1$, 所以可以引入一个辅助参数 $\alpha = k_0 - k_1 - 0.5$ 。显然 α 的

数值范围是[-0.5,0.5], 它是以原点为对称的。这样将式(4-22)经过变换量代换和改写后, 可以得到

$$\frac{y_2 - y_1}{y_2 + y_1} = \frac{|W(2\pi \cdot (-\alpha + 0.5)/N)| - |W(2\pi \cdot (-\alpha - 0.5)/N)|}{|W(2\pi \cdot (-\alpha + 0.5)/N)| + |W(2\pi \cdot (-\alpha - 0.5)/N)|} \quad (4-23)$$

令 $\beta = (y_2 - y_1)/(y_2 + y_1)$, 并且当 N 较大时, 式(4-23)一般可以简化为 $\beta = g(\alpha)$, 其反函数记为 $\alpha = g^{-1}(\beta)$ 。当窗函数 $w(n)$ 为实系数时, 其幅频响应 $|W(2\pi f)|$ 是偶对称的, 因而函数 $g(\cdot)$ 及其反函数 $g^{-1}(\cdot)$ 都是奇函数。采用双峰谱线修正方法通过多项式逼近可得如下常用窗函数的修正公式:

(1) 矩形窗函数

$$w(n) = 1 \quad (n = 0, 1, \dots, N-1)$$

$$\alpha = 0.5 \cdot \beta$$

$$\theta = \arg[\bar{X}(k_i \cdot \Delta f)] + \frac{\pi}{2} - \frac{N-1}{N} \cdot \pi \cdot (\alpha - (-1)^i \cdot 0.5) \quad (i = 1, 2)$$

$$A = N^{-1} \cdot (y_1 + y_2) \cdot (1.57078827 + 1046939868 \cdot \alpha^2 + 0.85101985 \cdot \alpha^4 + 0.55454503 \cdot \alpha^6)$$

(2) 三角窗函数

$$w(n) = 1 - |2n - N|/N \quad (n = 0, 1, \dots, N-1)$$

$$\alpha = 1016494712 \cdot \beta + 0.08335791 \cdot \beta^3 + 0.04327764 \cdot \beta^5 + 0.03696048 \cdot \beta^7$$

$$\theta = \arg[\bar{X}(k_i \cdot \Delta f)] + \frac{\pi}{2} - \frac{N-2}{N} \cdot \pi \cdot (\alpha - (-1)^i \cdot 0.5) \quad (i = 1, 2)$$

$$A = N^{-1} \cdot (y_1 + y_2) \cdot (2.46740051 + 1.39754151 \cdot \alpha^2 + 0.44472069 \cdot \alpha^4 + 0.11621963 \cdot \alpha^6)$$

(3) 哈宁窗函数

$$w(n) = 0.5 - 0.5 \cdot \cos(2\pi n/N) \quad (n = 0, 1, \dots, N-1)$$

$$\alpha = 1.5 \cdot \beta$$

$$\theta = \arg[\bar{X}(k_i \cdot \Delta f)] + \frac{\pi}{2} - \pi \cdot (\alpha - (-1)^i \cdot 0.5) \quad (i = 1, 2)$$

$$A = N^{-1} \cdot (y_1 + y_2) \cdot (2.35619403 + 1.15543682 \cdot \alpha^2 + 0.32607873 \cdot \alpha^4 + 0.07891451 \cdot \alpha^6)$$

(4) 哈明窗函数

$$w(n) = 0.54 - 0.46 \cdot \cos(2\pi n / N) \quad (n = 0, 1, \dots, N-1)$$

$$\alpha = 1.21874943 \cdot \beta + 0.13349531 \cdot \beta^3 + 0.05301420 \cdot \beta^5 + 0.03656014 \cdot \beta^7$$

$$\theta = \arg[\bar{X}(k_i \cdot \Delta f)] + \frac{\pi}{2} - \pi \cdot (\alpha - (-1)^i \cdot 0.5) \quad (i = 1, 2)$$

$$A = N^{-1} \cdot (y_1 + y_2) \cdot (2.26557103 + 1.22719978 \cdot \alpha^2 + 0.37607775 \cdot \alpha^4 + 0.09767389 \cdot \alpha^6)$$

(5) 布莱克曼窗函数

$$w(n) = 0.42 - 0.5 \cdot \cos(2\pi n / N) + 0.08 \cdot \cos(4\pi n / N) \quad (n = 0, 1, \dots, N-1)$$

$$\alpha = 1.96043163 \cdot \beta + 0.15277315 \cdot \beta^3 + 0.07425838 \cdot \beta^5 + 0.04998548 \cdot \beta^7$$

$$\theta = \arg[\bar{X}(k_i \cdot \Delta f)] + \frac{\pi}{2} - \pi \cdot (\alpha - (-1)^i \cdot 0.5) \quad (i = 1, 2)$$

$$A = N^{-1} \cdot (y_1 + y_2) \cdot (2.70205774 + 1.07115106 \cdot \alpha^2 + 0.23361915 \cdot \alpha^4 + 0.04017668 \cdot \alpha^6)$$

§ 4.2.2 MATLAB 算法及仿真验证

数据截断取 4 个周波。MATLAB 算法文件如下：

```
function [A, B]=myfft(currentY, N) %数据传输
n=[0:1:N-1]; %循环变量
s=currentY;
w=0.42-0.5*cos(2*pi*n./N)+0.08*cos(4*pi*n./N); %布莱克曼窗函数
r=s.*w; %对信号加窗
v=fft(r, N), %快速付立叶变换
u=abs(v), theta=angle(v),
A=zeros(1, 11);B=zeros(1, 11);
for I=0:10
    y1=u(5+4*I); %找出双峰谱线
```

```

y2=u(6+4*I);
y3=u(4+4*I);
max=y2;
if y3>y2
    max=y3;
end
if max==y3
    t=y1;
    y1=max;
    y2=t;
end
b=(y2-y1)/(y2+y1);    %参数 β
a=1.96043163*b+0.15277325*b.^3+0.07425838*b.^5+0.04998548*b.^7; %参数 α
%修正公式计算幅值
A(I+1)=(y1+y2)*(2.70205774+1.071151*a.^2+0.23361915*a.^4+0.04017668*a.^6)
/N;
` B(I+1)=theta(5+4*I)+pi/2-pi*(a+0.5); %修正公式计算相角
End

```

仿真验证对形如 $x(n) = Ae^{\frac{t}{\tau}} + \sum_{i=1}^7 A_i \sin(2\pi \frac{i \cdot f_1}{f_s} n + \varphi_i)$ 的信号进行谐波分析，其中基波频率 f_1 为 50.5Hz，采样频率 f_s 为 1600Hz，截断信号的数据长度取 128 点，衰减直流分量幅值为 15， $\tau = 30$ 毫秒，基波、各次谐波的幅值（无单位数值）和相位计算结果对比如表 4.1 所示：

表 4.1 计算谐波与实际谐波对比表

| 仿真信号 | 谐波次数(次) | | | | | | |
|---------------|---------|--------|--------|--------|---------|---------|---------|
| | 基波 | 2 | 3 | 4 | 5 | 6 | 7 |
| 实际幅值(A_i) | 15 | 3 | 2 | 1.5 | 1 | 0.3 | 0.1 |
| 计算幅值(A) | 15.001 | 2.9973 | 1.9997 | 1.5004 | 1.0003 | 0.30048 | 0.10024 |
| 实际相角(rad) | -0.4 | 2 | 1.8 | 1.6 | 0.5 | 0.4 | 0.3 |
| 计算相角(rad) | -0.3996 | 1.9935 | 1.7991 | 1.6029 | 0.50335 | 0.41731 | 0.32952 |

可见算法能很好地控制计算精度，同时不受衰减直流分量的影响。

图形显示采用直棒图并附各次谐波详细幅值和相角数值表。

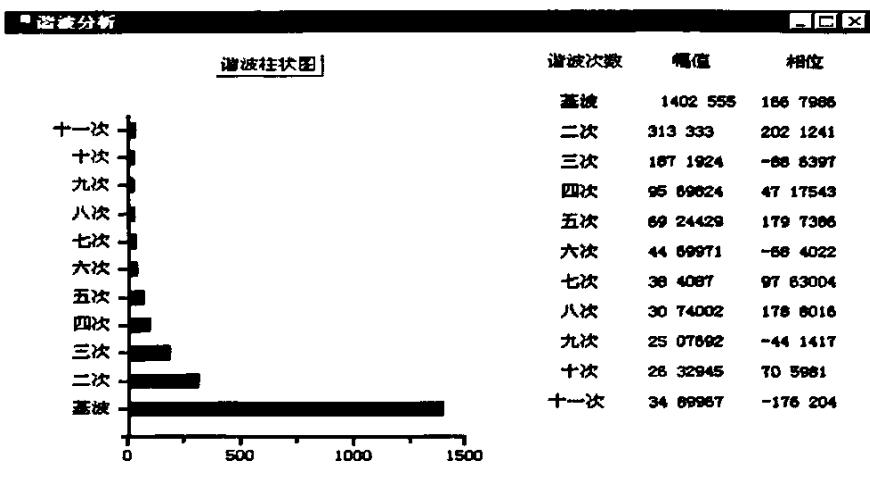


图 4.3 谐波分析页面

§4.3 序分量计算及选相元件的实现

§4.3.1 电流突变量起动元件^[54]

微机保护通常采用电流突变量作为起动元件。

$$\Delta i_k = i_k - i_{k-N} \quad (4-24)$$

式中, i_k ——电流在某一时刻 k 的采样值

N——一个工频周期内的采样点数

i_{k-N} ——比 i_k 提前一个周波的采样值

Δi_k ——k 时刻电流的突变量

系统正常运行时, 负荷电流保持稳定, i_k 和 i_{k-N} 应当接近相等。若在某一时刻发生短路, 故障相电流突然增大, 则在 k 点之后将有突变电流产生。为补偿电网频率变化引起的不平衡电流, 可采用下式计算:

$$\Delta i_k = |i_k - i_{k-N}| - |i_{k-N} - i_{k-2N}| \quad (4-25)$$

§4.3.2 序分量的移相算法

在分析线性三相不对称电路系统时, 对系统中的不对称电压、电流分量, 一般采用对称分量法进行分析计算, 就是将一组三相不对称电量, 分解成为三

个不同的对称序分量，即正序分量、负序分量和零序分量。引入运算符号 α ，令 $\alpha = 1 \cdot e^{j120^\circ}$ ，则对称序分量可表示为：

$$\begin{bmatrix} U_{a1} \\ U_{a2} \\ U_{a0} \end{bmatrix} = \frac{1}{3} \cdot \begin{bmatrix} 1 & \alpha & \alpha^2 \\ 1 & \alpha^2 & \alpha \\ 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} U_a \\ U_b \\ U_c \end{bmatrix}$$

微机保护中通常采用移相算法。零序分量的计算很简单：

$$3u_{0k} = u_{ak} + u_{bk} + u_{ck} \quad (4-26)$$

$$\text{正序分量 } U_{a1} = \frac{1}{3}(U_a + \alpha U_b + \alpha^2 U_c)$$

由于 $\alpha = 1 \cdot e^{j120^\circ}$ ， $\alpha^2 = 1 \cdot e^{j240^\circ}$ ，根据移相算法对于每周期采样 N 点，对应的相量采样值分别要延迟 $\frac{N}{3}T_s$ 和 $\frac{2N}{3}T_s$ ，用采样值表示的正序分量计算式为

$$u_{1k} = \frac{1}{3}(u_{ak} + u_{b(k+N/3)} + u_{c(k+2N/3)}) \quad (4-27)$$

$$\text{负序分量的计算式 } U_{a2} = \frac{1}{3}(U_a + \alpha^2 U_b + \alpha U_c)$$

同样由移相算法可得采样值表示的计算式为

$$u_{2k} = \frac{1}{3}(u_{ak} + u_{b(k+2N/3)} + u_{c(k+N/3)}) \quad (4-28)$$

§4.3.3 选相元件的实现

在计算出各相电流序分量后，即可按照正、负序故障分量的相位进行选相。选相元件相位比较动作判据为^[35]：

$$\text{DAG: } -60^\circ < \arg \frac{\dot{I}_{A1}}{\dot{I}_{A2}} < 60^\circ$$

$$\text{DBG: } -180^\circ < \arg \frac{\dot{I}_{A1}}{\dot{I}_{A2}} < -60^\circ$$

$$\text{DCG: } 60^\circ < \arg \frac{\dot{I}_{A1}}{\dot{I}_{A2}} < -180^\circ$$

$$\text{DBC(DBCG): } 120^\circ < \arg \frac{\dot{I}_{A1}}{\dot{I}_{A2}} < 240^\circ$$

$$\text{DAC(DACG): } 0^\circ < \arg \frac{i_{41}}{i_{42}} < 120^\circ$$

$$\text{DAB(DABG): } -120^\circ < \arg \frac{I_{A1}}{I_{A2}} < 0^\circ$$

由以上判据可以看出，两相短路与两相接地短路特征相同，而单相接地短路与两相接地短路有重叠区，因此需要设定两个限值 ε_1 和 ε_2 。 ε_1 根据相间短路时可能出现的最大不平衡电流决定， ε_2 根据两相短路和单相接地条件下可能出现的最大差值 $|I_{A1}| - |I_{A2}|$ 决定。选相元件框图如下：

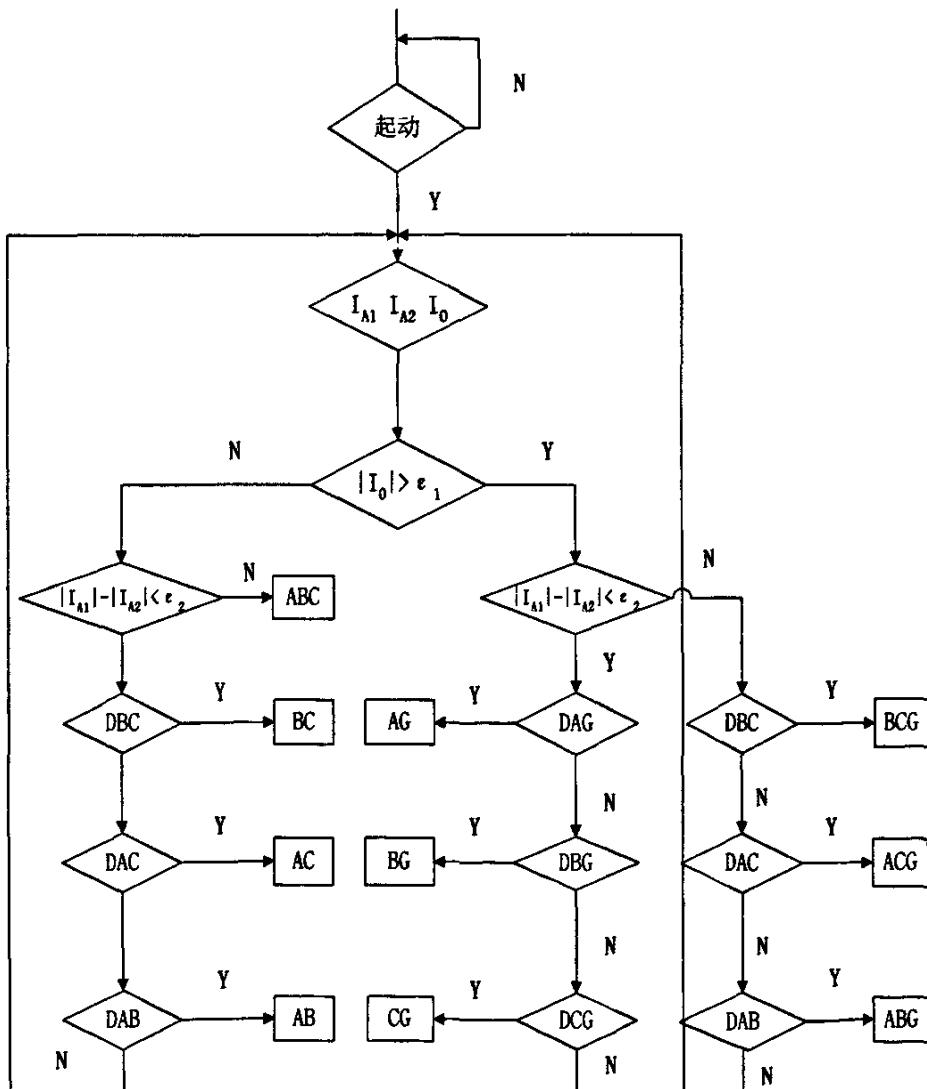


图 4.4 电流选相元件流程框图

§4.3.4 MATLAB 实现及图形界面

以下为 MATLAB 算法实现文件:

```

function [V1y,V1t,V2y,V2t,V0y,V0t,I1y,I1t,I2y,I2t,I0y,I0t,zzd,zd,p]=xfl(Va,Vb,Vc,Ia,lb,lc, N)
m=[Va;Vb;Vc;Ia;lb;lc]
[x,y]=size(m);
%故障突变量计算
c=zeros(x,N);
m1=[c m(:,1:y-N)];
m2=[c c m(:,1:y-2*N)];
m3=abs(m-m1)-abs(m1-m2);
M=[c m3(:,N+1:y)];           %故障突变量
mi=M(4:6,:);
ia1=M(4,:);ib1=M(5,:);ic1=M(6,:);
iab=ia1-ib1;ibc=ib1-ic1;ica=ic1-ia1;
%故障突变量单相定值
ma=max(abs(ia1));mb=max(abs(ib1));mc=max(abs(ic1));
zd=min([ma mb mc]);
%故障突变量相间定值
mab=max(abs(iab));mbc=max(abs(ibc));mca=max(abs(ica));
zzd=max([mab mbc mca])/50;
%故障起动判据
for i=N+1:y-1
    s=mi(:,(i-N):i);
    qd=max(max(abs(s)));
    if max(abs(mi(:,i+1)))> 5*qd
        p=i;
        break
    end
end
va=m(1,:);vb=m(2,:);vc=m(3,:);
ia=m(4,:);ib=m(5,:);ic=m(6,:);

```

```

C=zeros(1,2*N/3);
val=[va C];
vb=[vb C];
vc=[vc C];
ia=[ia C];
ib=[ib C];
ic=[ic C];
%移相算法计算序分量
for k=1:y
    v1a(k)=(val(k)+vb(k+N/3)+vc(k+2*N/3))/3;
    v2a(k)=(val(k)+vb(k+2*N/3)+vc(k+N/3))/3;
    i1a(k)=(ia(k)+ib(k+N/3)+ic(k+2*N/3))/3;
    i2a(k)=(ia(k)+ib(k+2*N/3)+ic(k+N/3))/3;
end
v0a=(Va+Vb+Vc)/3;
i0a=(Ia+Ib+Ic)/3;
i=1:N;a=1;
bc=sin(2*pi*i/N);bs=cos(2*pi*i/N);
%付立叶算法计算幅值和相角
V1r=filter(bc,a,v1a);V1i=filter(bs,a,v1a);
V2r=filter(bc,a,v2a);V2i=filter(bs,a,v2a);
V0r=filter(bc,a,v0a);V0i=filter(bs,a,v0a);
I1r=filter(bc,a,ia);I1i=filter(bs,a,ia);
I2r=filter(bc,a,ib);I2i=filter(bs,a,ib);
I0r=filter(bc,a,ic);I0i=filter(bs,a,ic);
V1z=complex(V1r,V1i);V2z=complex(V2r,V2i);V0z=complex(V0r,V0i);
I1z=complex(I1r,I1i);I2z=complex(I2r,I2i);I0z=complex(I0r,I0i);
V1y=abs(V1z);V2y=abs(V2z);V0y=abs(V0z);
V1t=angle(V1z);V2t=angle(V2z);V0t=angle(V0z);
I1y=abs(I1z);I2y=abs(I2z);I0y=abs(I0z);
I1t=angle(I1z);I2t=angle(I2z);I0t=angle(I0z);

```

操作时在波形显示页面拖动红游标选择采样点，点击序分量计算菜单打开序分量显示及故障选相页面。

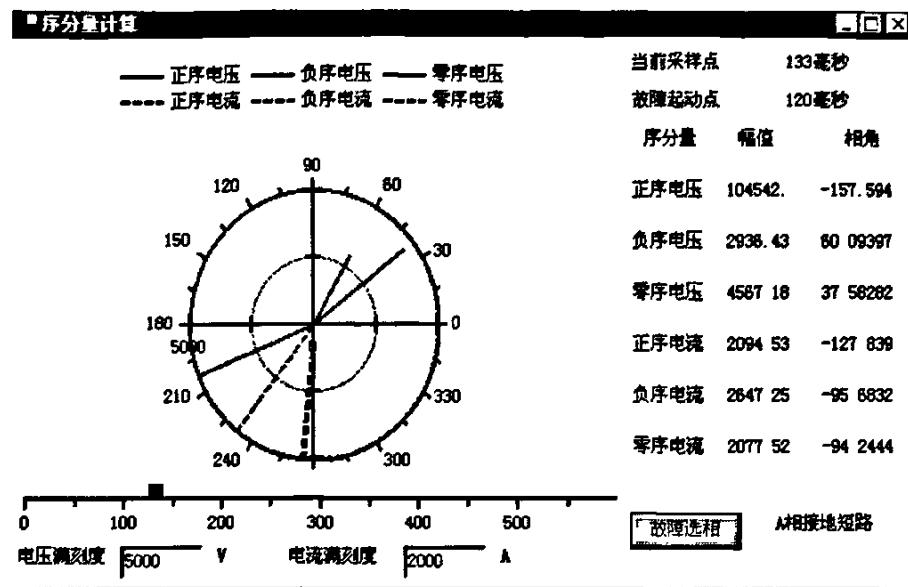


图 4.5 序分量计算页面

系统以极坐标方式显示各相电压和电流的序分量，下方设有最大刻度标尺，可选择合适的比例进行伸缩放大。左侧显示故障起动时刻、当前采样时刻和各相序分量幅值、相角的具体数值。拖动下方刻度尺上的滑块可以观察不同时刻的轨迹和幅值、相角。点击故障选相按钮调用选相元件自动进行故障选相。

§4.4 阻抗计算及保护特性分析^[56-58]

在确定了故障相后，即可代入相应的电流、电压值进行阻抗计算。本文采用付氏算法计算阻抗

设基波分量的电压和电流可表示为

$$\dot{U} = U_r + jU_i \quad (4-29)$$

$$\dot{I} = I_r + jI_i \quad (4-30)$$

式中 U_r 、 U_i 为基波电压的实部和虚部分量；

I_r 、 I_i 为基波电流的实部和虚部分量。

可求得阻抗为

$$\begin{aligned} Z &= R + jX = \frac{U}{I} = \frac{U_r + jU_i}{I_r + jI_i} \\ &= \frac{(U_r + jU_i)(I_r - jI_i)}{I_r^2 + I_i^2} = \frac{(U_r I_r + U_i I_i) + j(U_i I_r - U_r I_i)}{I_r^2 + I_i^2} \end{aligned} \quad (4-31)$$

所以可得

$$R = \frac{U_r I_r + U_i I_i}{I_r^2 + I_i^2} \quad (4-32)$$

$$X = \frac{U_i I_r - U_r I_i}{I_r^2 + I_i^2} \quad (4-33)$$

MATLAB 算法程序如下：

```
%从程序传入对应故障相电流电压、采样点N和故障启动点P并传回R、X值
function [R,X]=zkjfly(N,V,I,P)
v=V';i=I'; %矩阵转置
m=[v i];
[x,y]=size(m);
a=1;j=1:N; %付氏算法
bs=sin(2*pi*j/N);
bc=cos(2*pi*j/N);
ys=filter(bs,a,m);
yc=filter(bc,a,m);
%消除衰减直流分量影响
m1=m(P:P+N-1,:);m2=m(P+1:P+N,:);mm=m(P+1:x,:);
b=ones(1,N);a=1;
m0=filter(b,a,mm);
r=sum(m1,1)/sum(m2,1);
Ks=(ones(1,2)-r).*sin(2*pi/N)./(1+r.^2-2*r.*cos(2*pi/N));
Kc=(ones(1,2)-r).*(cos(2*pi/N)*ones(1,2)-r)./(1+r.^2-2*r.*cos(2*pi/N));
for i=1:x-P
    Ks1(i,:)=Ks; Kc1(i,:)=Kc;
end
```

```

ys1=ys(P+1:x,:)-Ks1.*m0;
yc1=yc(P+1:x,:)-Kc1.*m0;
ys(P+1:x,:)=ys1;
yc(P+1:x,:)=yc1;
vs=ys(:,1);is=ys(:,2);
vc=yc(:,1);ic=yc(:,2);
R=(vc.*ic+vs.*is)./(ic.^2+is.^2);    %计算R、X值
X=(vs.*ic-vc.*is)./(ic.^2+is.^2);

```

计算结果以折线图和阻抗图两种方式显示，保护特性采用阻抗圆特性和多边形特性。

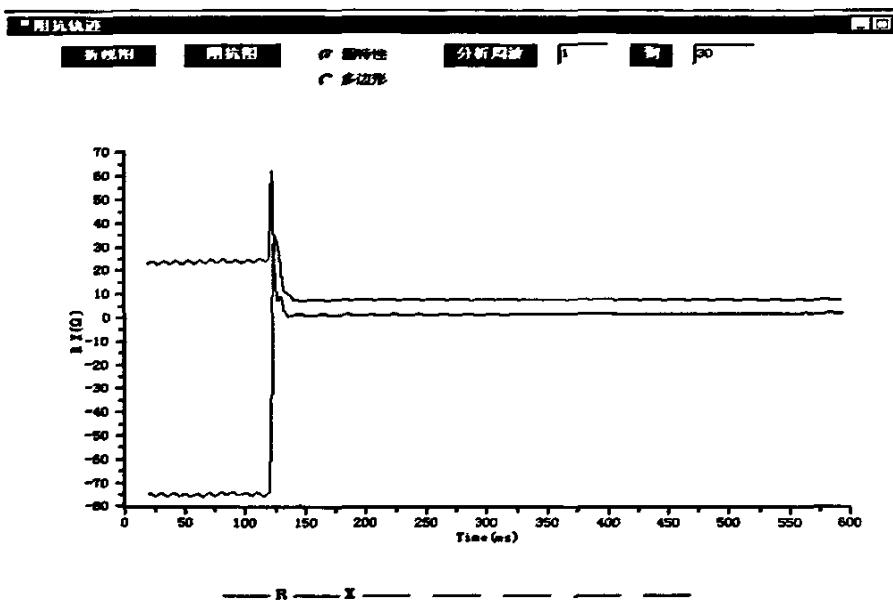


图 4.6 阻抗计算折线图

按照距离保护三段式时限特性进行保护特性分析。第 I 段为速断保护，反应于阻抗降低而瞬时动作时，不能保护本线路的全长，整定阻抗取线路正序阻抗的 80%~85%，整定值为：

$$Z_{dA} = K_k Z_{AB} \quad K_k = 0.8 \sim 0.85$$

第 II 段为限时保护，切除本线路速断范围以外的故障，保护本线路的全长，并作为速断的后备。按与相邻线路第 I 段配合，整定值为：

$$Z_{d\alpha A}^* = K_k (Z_{AB} + K_k \cdot Z_{BC}) \quad K_k = 0.8 \sim 0.85$$

第三段保护本线路及相邻下条线路的全长，作为相邻线路保护和断路器拒动的远后备或作为本线路 I、II 段的近后备。按与相邻线路第 II 段配合，整定值为：

$$Z_{d\alpha A}^* = 0.85 Z_{AB} + 0.8 Z_{d\alpha A}$$

以上整定未考虑分支系数。

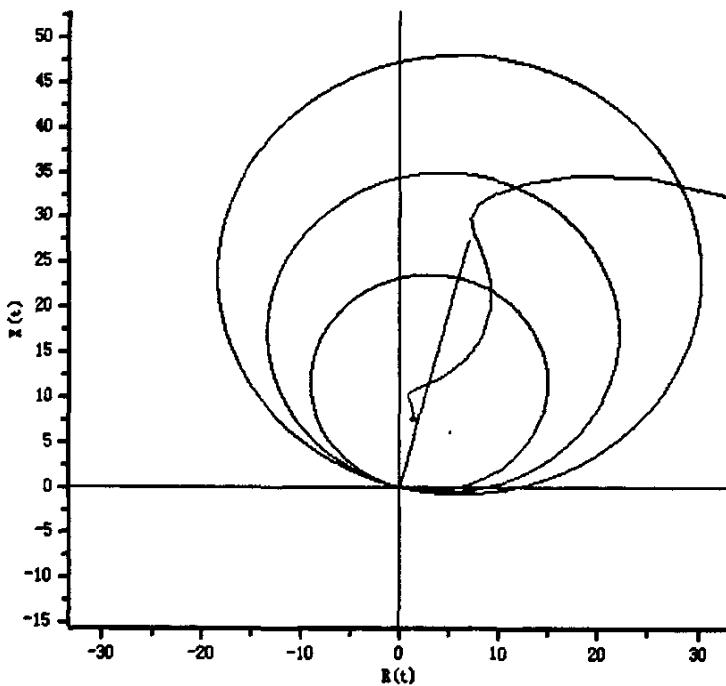
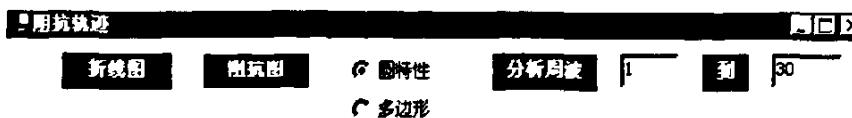


图 4.7 阻抗图及圆特性

多边形阻抗特性取 $\theta_1 = 7^\circ$, $\theta_2 = 15^\circ$, $\theta_3 = 60^\circ$, 测量阻抗在 A 区 (二象限) 的判别式为：

$$X_m > 0, \quad R_m < 0, \quad X_m \leq X_{d\alpha}, \quad |R_m| \leq \frac{1}{4} X_m$$

测量阻抗在 B 区 (一象限) 的判别式：

$$X_m > 0, \quad R_m > 0, \quad X_m \leq X_{d\alpha} - \frac{1}{8} R_m, \quad R_m \leq R_{d\alpha} + \frac{1}{2} X_m$$

测量阻抗在 C 区 (四象限) 的判别式:

$$X_m < 0, \quad R_m > 0, \quad R_m \leq R_{dz}, \quad |X_m| \leq \frac{1}{4} R_m$$

式中 R_{dz} 、 X_{dz} 为用户整定值。

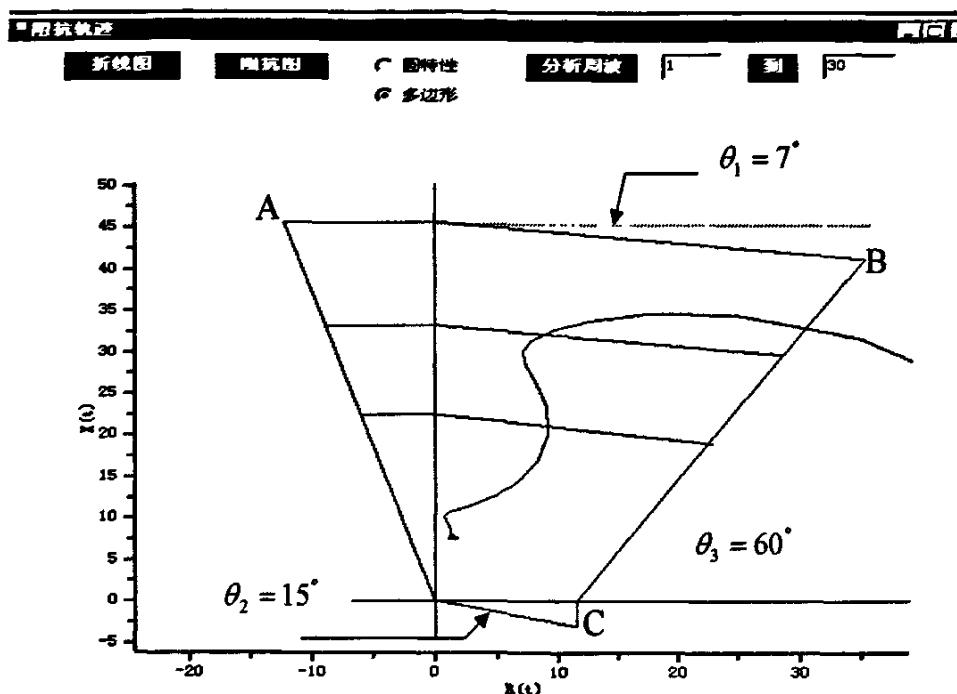


图 4.8 阻抗图及多边形特性

第五章 结束语

本文论述了基于 MS.NET 和 MATLAB 接口编程的录波信息分析系统的设计开发和相关技术。系统采用最新的面向对象组件技术，具有全图形和鼠标的操作界面，能够自动读入和保存标准 COMTRADE 录波文件，方便地进行波形显示和谐波分析、序分量计算、故障起动判断和故障选相、阻抗计算和简单的距离保护特性分析等功能。系统具有良好的扩展能力，可以和故障信息处理系统其它模块实现良好集成。

本文基于 MS.NET 平台实现 C#语言和 MATLAB 的混合编程，并应用于故障录波数据的分析，充分结合 MS.NET 优良的开发平台和 MATLAB 强大的数值计算能力，为电力系统录波分析软件的开发提供了新的思路和方法。

由于作者水平有限，错误和疏漏之处在所难免，恳请各位老师批评指正！

参考文献

- [1] Platt,D.S. Microsoft.NET 精髓 北京 机械工业出版社 2003
- [2] 周之英 现代软件工程(第二册) 北京 科学出版社 2003.1
- [3] Microsoft Corporation C# Language Specification White Paper 2001
<http://www.Microsoft.com/msdn>
- [4] Richard Conway, Teun Duynstee C#类设计手册 北京 清华大学出版社 2003.7
- [5] 苏金明,王永利 MATLAB 7.0 实用指南 北京 电子工业出版社 2004.11
- [6] Neyer A F, Wu F F, Imholz K. Object-Oriented Programming for Flexible Software: Example of a Load Flow IEEE Trans on Power System, 1990, 5(3): 689~696
- [7] Foley M, Bose A. Object-Oriented On-Line Network Analysis IEEE Trans on Power System, 1995,10(1): 125~132
- [8] Zhou E Z Object-Oriented Programming, C++ and Power System Simulation IEEE Trans on Power System, 1996, 11(1): 206~215
- [9] Zhu, J. Lubkeman, D.L. Object-oriented development of software systems for power system simulations IEEE Trans on Power System,1997,12(2): 1002~1007
- [10] 刘天斌,王永业,柳焕章,杨钢 基于 COMTRADE 格式的故障分析管理系统 继电器,2001,29(11): 47~49
- [11] 丁俊建,陆于平 基于 COM+技术的发电机变压器组故障录波分析系统软件 继电器,2002,30(1): 40~42
- [12] 陈佳胜,曾克娥,孙扬声,涂桂勤 大型发变机组故障录波装置分析软件的研制 电网技术 2002,26(5): 76~79
- [13] 桂勋,郭凯,谭永东,钱清泉 基于网络的全图形化故障录波分析软件系统 继电器 2004,32(24): 44~49
- [14] 李媛,刘涤尘,杜新伟,王和春,叶菁 电力故障波形再现及分析系统的开发 电网技术 2006,30(5): 106~109
- [15] Ren Jianwen, Zhou Ming, Li Gengyin. Analysis And management system for power system fault in information based on intranet network.. In: Proc. International Conference on Power System Technology Vol.2. USA, Oct.2002: 813~816

- [16] 李艳涛 栗然 赵敏 基于 Web 的继电保护管理信息系统研究与实现 电力自动化设备 2003, 23(11): 41~43
- [17] 刘志超 黄俊 承文新 电网继电保护及故障信息管理系统的实现 电力系统自动化 2003, 27(1): 72~75
- [18] 张征 房鑫炎 基于 Web 的多级电网继电保护自适应整定及信息管理系统 继电器 2004, 32(12): 55~58
- [19] 贾长朱 王剑峰 刘力丰 米增强 刘继平 基于 Web 的电力系统继电保护远程分布式信息管理系统的开发与设计 继电器 2000, 26(10): 65~68
- [20] 乐全明 郁惟镛 杜俊红 基于 J2EE 结构的分布式电网故障计算系统的实现 电网技术 2004, 28(11): 23~28
- [21] 陶红 袁金荣 MATLAB 与 C/Visual C++混合编程的实现 计算机工程与应用 2000, 10(28): 100~101+104
- [22] 王志鹏 马孝江 邹岩昆 钱志勤 MATLAB 数学库与 Windows 应用程序间接口库的设计 计算机工程 2001, 27(4): 95~97
- [23] 郭虹 薄云飞 VC++与 MATLAB 混合编程技术研究 计算机工程 2002, 28(9): 269~271
- [24] 穆玲玲 MATLAB6 与 Microsoft Visual C/C++的接口技术 微机发展 2003, 13(7): 3~4
- [25] 曾登高 .NET 系统架构与开发 北京 电子工业出版社, 2003. 4
- [26] 刘晓华 精通.NET 核心技术——原理与架构 北京 电子工业出版社 2002. 8
- [27] The MathWorks, Inc. MATLAB External Interfaces
- [28] 张威 MATLAB 应用程序集成与发布 西安 西安电子科技大学出版社 2005.7
- [29] 苏金明等 MATLAB 与外部程序接口 北京 电子工业出版社 2004. 1
- [30] The MathWorks, Inc. MATLAB Builder for COM
- [31] 岳玉芳 尤忠生 张玉双 基于 COM 的 VB 与 MATLAB 混合编程 计算机工程与设计 2005, 26(1): 67~68+71
- [32] Beyer, D. C# COM+编程指南 北京 中国水利水电出版社 2002. 3
- [33] Simon Robinson, K.Scott Allen C#高级编程 北京 清华大学出版社 2002. 8
- [34] Pappas, C. H. Murray, W. H. C# Windows 编程 北京 人民邮电出版社 2002. 10
- [35] 丁鹏 Inside C# —C#编程从入门到精通 北京 希望电子出版社 2002. 3

- [36] Charles Petzold, *Programming Microsoft Windows with C#* Microsoft Press
- [37] B-power 工作室 *C#窗口程序设计* 北京 中国铁道出版社 2001.11
- [38] 李兰友等 *Visual C#图形程序设计实例* 北京 国防工业出版社 2003.4
- [39] Ric White *GDI+程序设计* 北京 清华大学出版社 2002.12
- [40] IEEE Standard Common Format for transient data Exchange (COMTRADE) for power system, IEEE Standards Board, 1991
- [41] Francois Liger,Craig McQueen,Paul Wilton *C#字符串和正则表达式参考手册* 北京 清华大学出版社 2003.2
- [42] Kourosh Ardestani, Kevin Hoffman, Donald Xie *高效掌握 ADO.NET---C#编程篇* 北京 清华大学出版社 2003.3
- [43] Ferracchiati F.C. Jay Glynn 等 *.NET 数据服务 C#高级编程* 北京 清华大学出版社 2002.6
- [44] Nielsen, P. *Microsoft SQL Server 2000 宝典* 北京 中国铁道出版社 2004.1
- [45] 苏晓飞 *掌握 MATLAB 6.0 及其工程应用* 北京 科学出版社 2002.1
- [46] 陈小林 罗毅 王伟平 龙赛 莫京军 *基于故障录波数据的保护特性分析方法及其实现* *电网技术* 2005, 29(18): 70~74
- [47] 张杰 涂东明 张克元 *基于 COMTRADE 标准的故障录波的分析与再现* *继电器* 2000, 28(11): 20~22
- [48] Foley M, Bose A, Michell W et al, *An Object Based Graphical User Interface for Power System*, *IEEE Trans on Power System*, 1993, 8(1)
- [49] 陈德树 *计算机继电保护原理与技术* 北京 水利电力出版社 1992.11
- [50] 张伏生 耿中行 葛耀中 *电力系统谐波分析的高精度 FFT 算法* *中国电机工程学报* 1999, 19(3): 63~66
- [51] 庞浩,李东霞,俎云霄,王赞基 *应用 FFT 进行电力系统谐波分析的改进算法* *中国电机工程学报* 2003, 23(6): 50~54
- [52] 许珉 张鸿博 *基于 Blankman-harries 窗的加窗 FFT 插值修正算法* *郑州大学学报(工学版)* 2005, 26(4): 99~101
- [53] 张鸿博,蔡晓峰,许珉 *MATLAB 实现电网谐波测量加窗插值算法* *中原工学院学报* 2005, 16(2): 40~42
- [54] 杨奇逊 *微型机继电保护基础* 北京 中国电力出版社 1988.11
- [55] 葛耀中 *新型继电保护与故障测距原理与技术* 西安 西安交通大学出版社 1993.7

- [56] Dong Xingzhou Ge Yaozhong Xu Bingyin Fault position relay based on current traveling wave and wavelets[C] Proceedings of IEEE Winter Meetings Singapore 2000
- [57] 张举 微型机继电保护原理 北京 中国水利水电出版社 2004.9
- [58] 许建安 电力系统继电保护 北京 中国水利水电出版社 2004.1

作者在攻读硕士学位期间发表的论文

游锋, 陈皓 .NET 下基于 COM 组件的 C# 与 MATLAB 混合编程
电脑知识与技术 2005, 29(10): 65~67

声 明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。据我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得四川大学或其他教育机构的学位或证书而使用过的材料。与我一同工作过的同志对本研究的任何贡献均已在论文中做了明确的说明并表示谢意。

本学位论文成果是本人在四川大学读书期间在导师指导下取得的，论文成果归四川大学所有，特此声明。

研究生: 游峰

指导教师: 陈峰

致 谢

本文是在导师陈皓副教授的悉心指导、帮助和关心下完成的，倾注了陈老师大量的心血和智慧。

在三年的学习期间，陈老师对我进行了精心的教诲和耐心的指导，使我受益良多，有了长足的进步。陈老师渊博的知识、严谨的治学态度、敏锐的思维能力和谦和的品行为我树立了很好的榜样，在此谨向导师致以衷心的感谢。

在我学习期间得到了学院各位老师的授业、指导和帮助，他们传授给我的知识将终身受用。同时，也得到了许多同学的大力帮助，在此一并致以诚挚的谢意。

最后，要深深地感谢一直深爱和支持我的家人。谢谢！