

题 目： 基于 Matlab 的数字式
变压器差动保护仿真

2006 年 6 月 南 京

毕业设计说明书（论文）中文摘要

摘要：Matlab 高效处理矩阵运算的能力，使电力系统潮流计算的简化成为可能。作者利用 M 文件编写了电力系统变压器差动保护程序，并且利用 Matlab 的外部接口，结合 VB 编制界面友好、方便的长处，提出了基于 MATLAB 与 VB 混合编程的差动保护算法，以期为电力系统故障仿真提供参考。

本文结合实际输电线路仿真模型研究利用 Active X 技术实现 Matlab 与 VB 的数据交换问题。利用 Matlab 中的 Simulink 及 SPS(simpowersystem) 工具箱建立了 220kV 输电线路、变压器比率制动差动与二次谐波制动保护等仿真模型，获得了三相仿真电压、电流波形及保护动作波形，并将 Matlab 的仿真结果用 VB 界面显示。

以上两种新方法充分利用 Matlab 与 VB 的优点，便于仿真软件开发与使用。

关键词：Matlab；VB；变压器差动保护；仿真；继电保护

毕业设计说明书（论文）外文摘要

Title: Digital transformer differential action Protection
Simulation Based on Matlab

Abstract: Matlab' s high-efficient matrix calculation ability makes it possible to realize power flow analysis simplified. The author makes use of m-file to write differential action of transformer, then makes use of Matlab interface technology ,and to combine with the advantage of friendly and convenient on VB.To bring up a differential action protection analysis programs be blend-writed base on MATLAB and VB ,to expect to provide consultation with the fault simulation of the power system.

Based on an actual simulation model of power transmission system , using the Active X technology , this paper process a method which can be used to implement data change between Matlab and VB.First ,we build a simulation model which conclude 220kV transmission line power system , brake on ratio of differential action of transformer and twice harmonic waves to brake protection device .Then we simulate the single phase to ground permanent fault and single phase to ground instant fault in the model to get three phase voltage and current simulation waveforms and the protection operation waveform. Finally we show the simulation results on the Visual Basic software' s interface.

Two new methods processed in this paper integrate the advantages of Matlab and VB for the design and using of simulation software.

Keywords: Matlab ; VB ; differential action of transformer ; simulation ; protective relays

目 录

第一章 前言	6
第二章 Matlab 及其接口介绍.....	8
2.1 Matlab 编程基础.....	8
2.1.1 变量和数学运算	8
2.1.2 数组和数据	8
2.1.3 程序控制语句.....	9
2.1.4 辅助语句.....	11
2.2 M 文件及 M 函数.....	11
2.3 Simulink 介绍	12
2.3.1 设计中使用的模块	13
2.4 Matlab 外部接口介绍	14
2.4.1 Matlab 引擎应用程序	14
2.5 S-FUNCTION 简介.....	16
2.5.1 S-Function 的概念.....	17
2.5.2 建立 M 文件 S-Function.....	17
2.5.3 模块的封装与测试:	20
第三章 变压器差动保护	20
3.1 差动元件基本原理	20
3.1.1 差动元件的动作方程	20
3.1.2 差动电流及制动电流的取得	22
3.2 电流互感器二次电流相位补偿	23
3.2.1 电流互感器二次接线进行相位补偿（外转角）	23
3.2.2 用保护内部算法进行相位补偿（内转角）	24
3.3 CT 二次断线	27
3.4 逻辑构成框图	28
第四章 变压器差动保护仿真	30
4.1 保护仿真软件概述.....	30
4.2 软件设计步骤	31
4.3 一次供电系统模型.....	31
4.4 保护仿真模型	35
4.4.1 “差动保护” 模块子系统(Subsystem)	35
4.4.2. 变压器差动保护跳闸的实现	37
4.5 Matlab 与 VB 的接口.....	48
4.5.1 接口介绍	48
4.5.2 VB 界面设计.....	49
4.5.3 程序代码.....	50
4.5.4 设置故障与仿真结果	52

4.6 软件使用	54
第五章 结果分析	55
5.1 微机变压器差动保护的整定计算.....	55
5.2 模型仿真结果	56
工 作 总 结	63
致 谢 词	66
参考文献.....	66

第一章 前言

本设计的题目是基于 MATLAB 的电力变压器差动保护的仿真研究，研究包括对电力系统故障仿真和差动继电器仿真的研究。

电力变压器是电力系统普遍使用的重要电气设备。它的安全运行直接关系到电力系统的发、供电和稳定运行。随着现代电力系统继电保护的日益发展采用计算机仿真方法来分析研究电力系统继电保护是解决这类工程问题的一种有效工具。MATLAB语言是目前国际上流行的一种演算纸式的编程语言，它具有强大的矩阵分析与运算功能，并且是一个开放的环境。其中SIMULINK就是为MATLAB开发的一种优秀的控制系统仿真工具软件，它具有模块化、可重载、可封装、面向结构图编程及可视化等特点，可大大提高系统仿真的效率和可靠性。并且利用Matlab的外部接口，结合VB编制界面友好、方便的长处，提出了基于MATLAB与VB混合编程的差动保护算法，以期为电力系统故障仿真提供参考。

在变压器继电保护仿真方面，我们使用了 SimuLink 和 SimPowerSystems 工具箱，它常用于电力系统各个方面的仿真。例如 MATLAB 被用于发电机、变压器、线路和负荷等元件的建模和仿真，在输电线路研究中，有利用 Matlab 对输电线路进行故障定位数字仿真的研究。在继电保护仿真中也应用到了 Matlab，如基于 Matlab 开发平台的继电保护仿真系统。

本设计主要针对变压器的主保护——纵差动保护进行仿真研究。利用 MATLAB 及 VB 编写变压器差动保护的仿真程序（M 文件），使用了 SimuLink 和 SimPowerSystems 工具箱，构建了一个变压器的仿真模块。在系统中设定不同的故障点，分别进行了相应的仿真，且与变压器的整定计算值相配合，做到有选择性的动作，对仿真结果进行分析。

本文在第一章中为前言，介绍了本次设计的主要思路。

第二章介绍了 Matlab 基础和在设计中所用到知识。

第三章变压器保护的基本原理等基础理论知识。

第四章中对电力变压器保护仿真中所设计的差动保护仿真的程序，模型构成、参数设定及故障发生时保护装置的动作情况作了详细说明。并介绍了所设计的差动保护仿真与微机变压器保护整定值的配合。

第五章对仿真实验的结果进行了分析，分层面的做出分析，对于在整个设计过程的不同阶段做出具体的分析。

在本文的第六章中总结了此次设计并给出了结论。其中包括致谢词，参考文献等。

此次毕业设计在韩笑老师的指导下完成。

第二章 Matlab 及其接口介绍

本章简单介绍了此次设计用到的 Matlab 知识, 首先介绍 Matlab 软件编程基础, 再对计算机实现变压器差动保护程序设计中用到的 M 文件及 M 函数进行介绍, 然后介绍在进行继电器仿真设计中使用到的模块的来源、功能、参数等做详细说明。在变压器差动保护仿真过程中, 都使用了 Matlab 的外部接口功能, 所以在最后将介绍 Matlab 的各种外部接口功能。

2.1 Matlab 编程基础

2.1.1 变量和数学运算

首先从 Matlab 数学运算开始说明。例如要计算 $1+2+3$ 及 $1\times 10+2\times 10+3\times 10$ 这两个算式, 接在提示符号 `>>` 之后的是要键入的算式, Matlab 将计算的结果以 `ans` 显示。

2.1.2 数组和数据

Matlab 的运算事实上是以数组 (array) 及矩阵 (matrix) 方式在做运算, 而这两者在 Matlab 的基本运算性质上是不同的, 数组强调元素对元素的运算, 而矩阵则采用线性代数的运算方式。

在声明一个变量为数组或是矩阵时, 如果是要个别输入元素, 需要用中括号将元素置于其中。数组由一维元素构成, 而矩阵由多维元素组成。

在介绍矩阵运算之前首先介绍几个特殊的矩阵。Zeros 函数生成元素皆为 0 的矩阵; ones 函数生成元素皆为 1 的矩阵, eye 生成一个单位矩阵, 之所以称为 eye 是取其发音与原来单位矩阵符号 I 相同, 而又避免与复数定义中的虚部所用的符号 i 雷同, 所以改以 eye 替代。

上述三个函数的使用语法都相似, 如 `zeros (m)` 可以产生一个 $m\times m$ 的正方形矩阵, 而 `zeros (m, n)` 产生的是 $m\times n$ 的矩阵。也可以使用这三个函数将一个 $m\times n$ 矩阵原来元素全部取代成为 0, 1 或是单位矩阵的值, 不过要加上 `size`

指令来指出其矩阵大小是 $m \times n$ ，所以语法为：

`zeros (size (A))`

其中 A 为原来的矩阵

2.1.3 程序控制语句

For 循环语句：

For 循环允许一组命令以固定的次数重复。其一般形式是：

For 变数=矩阵，

运算式；

End

在 For 和 end 之间的运算式按数组中的每一列执行一次。在每一次迭代中，x 被指定为数组的下一列，即在第 n 次循环中，`x=array(:, n)`：

在使用 For 循环时需要注意以下两点：

为了加快速度，在 For 循环（while 循环）被执行之前应预先分配数组。建议最好先使用 `zeros` 或 `ones` 命令来预先设置所需的内存（即矩阵）大小；

可以利用 `break` 命令跳出 For 循环。

while 语句循环

while 语句循环以不定的次数求一组语句的值。其使用格式是：

while 条件句

运算式；

end

只要表达式里的所有元素的值为真，就执行 while 和 end 语句之间的运算式。通常，表达式的值是一个标量值，但数组值也同样有效。在数组情况下，所得到数组的所有元素的值必须都为真。也就是说，只要条件成立，运算就会一再被执行。

同样 while 循环也有以下两点需要注意：

可以利用 `break` 命令跳出 while 循环；

While 循环可按要求嵌套。

if-else-end 分支结构

最简单的 if-else-end 结构是：

```
if 条件句，  
    运算式；  
End
```

如果在表达式中的所有元素为真（非零），那么就执行 if 和 end 语言之间的语句。

如果有两个选择，if-else-end 结构是：

```
if 条件句  
    运算式  
else  
    运算式  
end
```

当有多种选择时，可以使用下面的语句：

```
if 条件式  
    运算式  
elseif 条件式  
    运算式  
elseif 条件式  
    运算式  
end
```

switch-case 语句：

Matlab 中 switch-case 的语句格式为：

```
switch num  
    case n1  
        command  
    case n2  
        command  
    case n3  
        command  
    .
```

```
.  
otherwise  
    command  
end
```

2.1.4 辅助语句

Matlab 中辅助语句包括：注释语句、中断语句、暂停语句和回显语句。Matlab 中的注释语句是由%开始，也就是说在%之后的任何文字都被视为程序的注释。注释的功能是简要的说明程序的内容，任何可能产生混淆的地方都应该使用注释，书写适量的注释可以在以后想了解程序时节省一些时间。

中断语句 break 用于终止一个循环语句的执行过程，特别可以利用 break 命令跳出 For 和 while 循环。

而暂停语句 pause 的作用是使用程序暂时停止运行，直到按下回车键，程序再继续执行程序。而 pause(n) 是中断 n 秒后，程序自动继续执行。另外使用 Ctrl+C 键可以用来中止执行中的 Matlab 的工作。

2.2 M 文件及 M 函数

Matlab 程序大致分为两类：M 脚本文件（M-Script）和 M 函数（M-Function），它们均是普通的 ASCII 码构成的文件。

M 脚本文件中包含一组有 Matlab 语言所支持的语句，它类似 DOS 下的批处理文件。它的执行方式很简单，用户只需在 Matlab 的提示符下输入该 M 文件的文件名，这样 Matlab 就会自动执行该 M 文件中的各条语句，并将结果直接返回到 Matlab 工作空间。使用 M 函数格式变成是 Matlab 程序设计的主流。

Matlab 的 M 函数是有 function 语句引导的，其基本格式如下：

Function [返回变量列表]=函数名(输入变量列表)

注释说明语句段，由%一道

输入、返回变量格式的检测

函数体语句

这里输入和返回变量的实际个数分别有 `nargin` 和 `nargout` 两个 Matlab 的保留变量来给出。只要进入该函数，Matlab 就将自动生成这两个变量，不论是直接使用这两个变量。如果返回变量多于一个，则应该用方括号括起来。输入变量和返回变量之间用括号分隔。注释语句段的每行语句都应该由百分号引导，百分号后面的内容不执行，只起注释作用。用户采用 `help` 命令则可以显示出来注释语句段的内容。此外、标准的变量树木检测也是必要的。如果输入或返回变量格式不正确，则应该给出相应的提示。

另外，因为 Matlab 是一种注释性语言，所以即使在某个或某些函数中存在语法错误，如果没执行到该语句时可能就不会发现该错误，这在严格的程序设计中是不容许的。要检查某目录中所有的 M 函数语法错误，首先应该用 `cd` 命令进入该目录，然后运行 `pcode*` 命令进行伪代码转换。因为该命令会将 Matlab 函数转换成伪代码，而在转换过程中该程序将自动翻译每一条语句，所以一旦发现有语法错误，将会停止翻译，给出错误信息。改正了该语法错误后，再重新执行 `pcode` 命令，直到没有错误为止。这样会保证目录下所有的程序不含有语法错误。

2.3 Simulink 介绍

Simulink 是以工具库的形式挂接在 Matlab 上的，不能独立运行，只能在 Matlab 环境中运行。Simulink 是一个用来对动态系统进行建模、仿真和分析的软件包，它支持连续、离散或两者混合的线性和非线性系统，也支持具有多种采样速率的多速率系统。

Simulink 是由模块库、模型构造及指令分析、演示程序等几部分组成。Simulink 提供了用方框图进行建模的图形接口。模块框图是动态系统的图形显示，由一组称为模块的图标组成，模块之间采用连线联接。每个模块代表了动态系统的某个单元，并且产生一定的输出。模块之间的连线表明模块的输入端口与输出端口之间的信号连结。模块的类型决定了模块输出与输入、状态和时间之间的关系。一个模块框图可以根据需要包含任何类型的模块。

模块代表了动态系统的某个功能单元，每个模块一般包括一组输入、状态和一组输出等几个部分。

Simulink 模块的基本特点是参数化的，许多模块都具有独立的属性对话框，在对话框中用户可以定义模块的各种参数。Simulink 包含 Sinks（输出方式）、Source（输入源）、Continuous（连续环节）、Nonlinear（非线性）、Discrete（离散环节）、Signals & System（信号与系统）、Math（数学模块）和 Functions & Tables（函数和查询表）等子模型库。SIMULINK 在诸如 Communication Toolbox, Nonlinear Control Design Blockset, Power System Blockset 等工具箱的配合下，还可以完成对诸如通行系统、非线性控制系统、电力系统的建模、分析和仿真。

2.3.1 设计中使用的模块

本设计中共使用了位于 Simpowersystems 模块库，和 Simulink 模块库中的 29 个不同的模块。分别为：

1) 位于 Simpowersystems 模块库中：

- ① 三相断路器模块(Three-Phase break)，
- ② 分布参数线路模块(Distributed parameter line block)，
- ③ 三相电压—电流测量器模块(Three-Phase V-I Measurement block)，
- ④ 三相变压器(双绕组)模块(Three-Phase Transformer (Two Windings) block)，
- ⑤ 三相电源模块(Three-Phase Source block)，
- ⑥ 三相串联 RLC 负载模块 (Three-Phase Series RLC Load block)，
- ⑦ 三相故障模块 (Three-Phase Fault block)，
- ⑧ 傅利叶变换模块 (Fourier block)，
- ⑨ 万用表模块 (Multimeter block)，
- ⑩ L 连接器模块 (L Connector block)，
- 接地模块 (Ground(input/output) block)，
- Powergui 模块 (Powergui block)；

2) 位于 Simulink 模块库中：

- ① 到工作空间模块 (To Workspace block)，
- ② 输入端口模块 (In1 block)，

-
- ③ 输出端口模块 (Out1 block),
 - ④ 分路器模块 (Demux block),
 - ⑤ 示波器模块 (Scope block),
 - ⑥ 常数模块 (Constant block),
 - ⑦ 乘法模块 (Product block),
 - ⑧ 符号模块 (Sign block),
 - ⑨ 增益模块 (Gain block)。

2.4 Matlab 外部接口介绍

Matlab是一个具有强大的矩阵运算、数据处理和图形显示功能的软件，其输出结果可视化，编程效率极高，极少的代码即可实现复杂的运算。然而其边解释边运行，运算效率较低，实现相同功能的代码跟其他编程平台相比运行时间相对较长，且不适合作为通用的编程平台。

MATLAB提供的应用程序接口一般来说可以分为以下三种：第一种接口利用了其他应用程序的优点，例如计算速度快和使用已有算法而制定的接口；第二种建立Matlab与其他应用程序间的数据交换；第三种接口拓广Matlab的应用范围和应用手段，开发的相应应用程序为MEX文件，MAT文件和Matlab引擎应用程序。其中MEX文件作为一种动态链接库文件，必须通过在Matlab的工作环境中调用才能运行；MAT文件用于数据交换，不能利用Matlab提供的功能来完成计算任务；Matlab引擎应用程序是一种可以独立执行的应用程序，但是在应用程序执行时，将在后台启动一个Matlab进程，然后按照要求返回计算结果。

新版本的Matlab软件包中还提供了C/C++的数学和图形库，通过其编译器支持可以将Matlab中编写的m文件转换成以C/C++代码的文件，而且可以将m生成dll库，甚至我们可以直接调用其中的库函数，生成并发布不必依赖Matlab的可执行文件。通过mcc编译器生成C/C++代码，进而可以在VC或者其他编译器生成可以独立执行的应用程序。

2.4.1 Matlab 引擎应用程序

Matlab 引擎库提供了一组以 eng 为前缀的函数，在 C++代码中使用这些函

数时，需要加入编译指令：

```
#include "engine.h"
```

这一小节将介绍这些函数的用法。

1) Engine * engOpen (const char * startcmd)

engOpen 函数开启 Matlab engine，创建一个对话 session。

这个函数的参数是一个命令字符串，而在 Windows 下这个字符串的值一般是 Null，而在 UNIX 下可以利用这个字符串指定主机名连接到远程的 Matlab engine 上。需要指出的是在 Windows 和 UNIX 下使用 engOpen 函数连接到 Matlab engine 的机制是不同的，UNIX 下使用输入输出两条管道来与 Matlab engine 进程通信，而在 Windows 下使用的是 COM 的 Server/Client 通信机制，这需要 Matlab 在安装时在本地计算机上进行注册，如果没有注册，可以使用下面命令注册 Matlab engine COM Server：

```
Matlab/regserver
```

函数返回值是一个指向 Matlab engine 句柄的指针。

这个函数类似于数据库连接中的打开数据源的操作。

2) Engine * engOpenSingleUse(const char * startcmd, void * dcom, int * retstatus)

这个函数与 engOpen 相似，不同之处在于它允许一个用户进程以独占的方式使用本地计算机的 Matlab Engine Server。

在 Windows 系统中使用这个函数时，前两个参数应该都置为空，如果出错，第三个参数返回一个可能的错误原因序号。

3) Int engClose(Engine * ep)

关闭 Matlab Engine，关闭连接 session。

其参数是一个指向 Matlab Engine 句柄的指针，返回一个整型的变量，如果是 0 则表示成功关闭链接，1 则表示出现错误，例如试图关闭一个已经关闭的连接。

4) Int engPutVariable (Engine * ep, const char * name, const mxArray * mp) 这个函数的作用是从应用程序向 Matlab Engine 发送一个 Matlab 矩阵 (mxArray 类型的参数)，用于数据交换。其中有三个参数，ep 是指向 Matlab Engine 句柄的指针，而且必须保证连接处于 Active 的状态，mp 是 Matlab 矩

阵的指针，name 表示在后台 Matlab 运行时这个参数的名称，即在 Matlab 命名空间中的标识。返回值为整型变量，其意义与 engClose 返回值相同。需要注意的是如果运行中的 Matlab 进程中没有与这个函数传入的参数名称相同的变量，Matlab 会创建一个升年的句柄，如果已经存在相同的变量，则会将原先的值覆盖。在老版本的 Matlab 头文件中，与这个函数功能类似的是 engPutArray。

5) mxArray * engGetVariable(Engine * ep, const char * name)

这个函数的作用与 engPutVariable 相反，是从 Matlab Engine 中获得一个 Matlab 矩阵。其中 name 参数是 Matlab 命名空间中的名称。

在老的版本的 Matlab 头文件中，与这个函数功能类似的是 engGetArray。

6) Int engEvalString(Engine * ep, const char * string)

这个函数通过打开的 Matlab Engine 连接执行一条 Matlab 命令，参数 String 即是用户所需要后台 Matlab 执行的指令，注意其中使用的参数名使用的不是 C++ 程序段中的命名，而是该连接里 Matlab 使用的名字空间。

7) Int engOutputBuffer(Engine * ep, char * p, int n)

这个函数的作用是创建字符缓冲区以获取 Matlab 文本输出。

参数 p 是接受输出的缓冲区指针，而参数 n 表示这个输出功能，可以使用下面的语句：

```
engOutputBuffer( ep, NULL, 0);
```

2.5 S-FUNCTION 简介

Simulink 为用户提供了许多内置的基本库模块，通过这些模块进行连接而构成系统的模型。对于那些经常使用的模块进行组合并封装可以构建出重复使用的新模块，但它依然是基于 Simulink 原来提供的内置模块。

而 Simulink s-function 是一种强大的对模块库进行扩展的新工具。

2.5.1 S-Function 的概念

- 1) s-function 是一个动态系统的计算机语言描述，在 MATLAB 里，用户可以选择用 m 文件编写，也可以用 c 或 mex 文件编写，在这里只给大家介绍如何用 m 文件编写 s-function。
- 2) S-function 提供了扩展 Simulink 模块库的有力工具，它采用一种特定的调用语法，使函数和 Simulink 解法器进行交互。
- 3) S-function 最广泛的用途是定制用户自己的 Simulink 模块。它的形式十分通用，能够支持连续系统、离散系统和混合系统。

2.5.2 建立 M 文件 S-Function

使用模板文件：sfuntmpl. m

- 1) 该模板文件位于 MATLAB 根目录下 toolbox/simulink/blocks 目录下。
- 2) 模板文件里 s-function 的结构十分简单，它只为不同的 flag 的值指定要相应调用的 m 文件子函数。比如当 flag=3 时，即模块处于计算输出这个仿真阶段时，相应调用的子函数为 `sys=mdloutputs(t,x,u)`。
- 3) 模板文件使用 switch 语句来完成这种指定，当然这种结构并不唯一，用户也可以使用 if 语句来完成同样的功能。而且在实际运用时，可以根据实际需要来去掉某些值，因为并不是每个模块都需要经过所有的子函数调用。
- 4) 模板文件只是 Simulink 为方便用户而提供的一种参考格式，并不是编写 s-function 的语法要求，用户完全可以改变子函数的名称，或者直接把代码写在主函数里，但使用模板文件的好处是，比较方便，而且条理清晰。
- 5) 使用模板编写 s-function，用户只需把 s-函数名换成期望的函数名称，如果需要额外的输入参量，还需在输入参数列表的后面增加这些参数，因为前面的 4 个参数是 simulink 调用 s-function 时自动传入的。对于输出参数，最好不做修改。接下去的工作就是根据所编 s-function 要完成的任务，用相应的代码去替代模板里各个子函数的代码即可。

-
- 6) Simulink 在每个仿真阶段都会对 s-function 进行调用，在调用时，Simulink 会根据所处的仿真阶段为 flag 传入不同的值，而且还会为 sys 这个返回参数指定不同的角色，也就是说尽管是相同的 sys 变量，但在不同的仿真阶段其意义却不相同，这种变化由 simulink 自动完成。
- 7) m 文件 s-function 可用的子函数说明如下：
- ①mdlInitializeSizes: 定义 s-function 模块的基本特性，包括采样时间、连续或者离散状态的初始条件和 sizes 数组。
 - ②mdlDerivatives: 计算连续状态变量的微分方程。
 - ③mdlUpdate: 更新离散状态、采样时间和主时间步的要求。
 - ④mdlOutputs: 计算 s-function 的输出。
 - ⑤mdlGetTimeOfNextVarHit: 计算下一个采样点的绝对时间，这个方法仅仅是在用户在 mdlInitializeSizes 里说明了一个可变的离散采样时间。
 - ⑥mdlTerminate: 实现仿真任务必须的结束。
- 8) 概括说来，建立 s-function 可以分成两个分离的任务：
- 9) 初始化模块特性包括输入输出信号的宽度，离散连续状态的初始条件和采样时间。
- 10) 将算法放到合适的 s-function 子函数中去。

定义 s-function 的初始信息

- 1) 为了让 Simulink 识别出一个 m 文件 s-function, 用户必须在 s-函数里提供有关 s-函数的说明信息，包括采样时间、连续或者离散状态个数等初始条件。这一部分主要是在 mdlInitializeSizes 子函数里完成。
- 2) Sizes 数组是 s-function 函数信息的载体，它内部的字段意义为：
 - ①NumContStates: 连续状态的个数（状态向量连续部分的宽度）
 - ②NumDiscStates: 离散状态的个数（状态向量离散部分的宽度）
 - ③NumOutputs: 输出变量的个数（输出向量的宽度）
 - ④NumInputs: 输入变量的个数（输入向量的宽度）
 - ⑤DirFeedthrough: 有无直接馈入
 - ⑥NumSampleTimes: 采样时间的个数
- 3) 如果字段代表的向量宽度为动态可变，则可以将它们赋值为 -1。

-
- 4) 注意 DirFeedthrough 是一个布尔变量，它的取值只有 0 和 1 两种，0 表示没有直接馈入，此时用户在编写 mdlOutputs 子函数时就要确保子函数的代码里不出现输入变量 u；1 表示有直接馈入。
 - 5) NumSampleTimes 表示采样时间的个数，也就是 ts 变量的行数，与用户对 ts 的定义有关。
 - 6) 需要指出的是，由于 s-function 会忽略端口，所以当有多个输入变量或多个输出变量时，必须用 mux 模块或 demux 模块将多个单一输入合成一个复合输入向量或将一个复合输出向量分解为多个单一输出。

输入和输出参量说明

- 1) S-function 默认的 4 个输入参数为 t、x、u 和 flag，它们的次序不能变动，代表的意义分别为：
 - ①t：代表当前的仿真时间，这个输入参数通常用于决定下一个采样时刻，或者在多采样速率系统中，用来区分不同的采样时刻点，并据此进行不同的处理。
 - ②x：表示状态向量，这个参数是必须的，甚至在系统中不存在状态时也是如此。它具有很灵活的运用。
 - ③u：表示输入向量。
- 2) flag：是一个控制在每一个仿真阶段调用哪一个子函数的参数，由 Simulink 在调用时自动取值。
- 3) S-function 默认的 4 个返回参数为 sys、x0、str 和 ts，它们的次序不能变动，代表的意义分别为：
 - ①sys：是一个通用的返回参数，它所返回值的意义取决于 flag 的值。
 - ②x0：是初始的状态值（没有状态时是一个空矩阵[]），这个返回参数只在 flag 值为 0 时才有效，其他时候都会被忽略。
 - ③str：这个参数没有什么意义，是 MathWorks 公司为将来的应用保留的，m 文件
 - ④s-function 必须把它设为空矩阵。
 - ⑤ts：是一个 $m \times 2$ 的矩阵，它的两列分别表示采样时间间隔和偏移。

2.5.3 模块的封装与测试:

将该程序以文件名*****.m 存盘, 编好 S 函数后, 就可以进行封装和测试了, 向模型编辑窗口中添加 S-FUNCTION 模块, 双击该模块, 打开参数设置对话框, 在其中输入 M 文件名和用户定义的参数。选择该模块, 按 CTRL+M 或右键单击该模块。从弹出的快捷菜单中添加用户定义的变量参数。在 Initialization 页中对变量初始化在 Documentation 页中添加模块的说明和帮助文档, 其具体操作与子系统的封装类似。

可以封装该 S 函数, 从而设计出相应的参数输入对话框, 选择 S 函数模块, 打开其封装编辑器, 在 Parameters 页中添加用户定义的变量参数。

S 函数模块被封装后, 双击它, 则有模块参数对话框。能有附加参数的一个提示。

第三章 变压器差动保护

变压器纵差动保护用于反映变压器绕组的相间短路故障、绕组的匝间短路故障、中性点接地侧绕组的接地故障及引出线的相间短路故障、中性点接地侧引出的接地故障。目前国内生产及应用的变压器(主变、厂高变、发变组、高压启备变)微机型差动保护, 主要由分相差动元件和涌流判别元件两部分构成。此外, 用于大型变压器的差动保护, 还有 5 次谐波制动元件, 以防止变压器过激磁时差动保护误动。

在变压器内部严重故障时, 为防止由于电流互感器(TA)饱和、电流波形畸变而致使差动元件拒动或延缓动作, 还设置有差动速断元件。

3.1 差动元件基本原理

3.1.1 差动元件的动作方程

在国内生产的微机型变压器差动保护中, 差动元件的动作特性较多采用具有二段折线型的动作特性曲线, 如图 3—1 所示

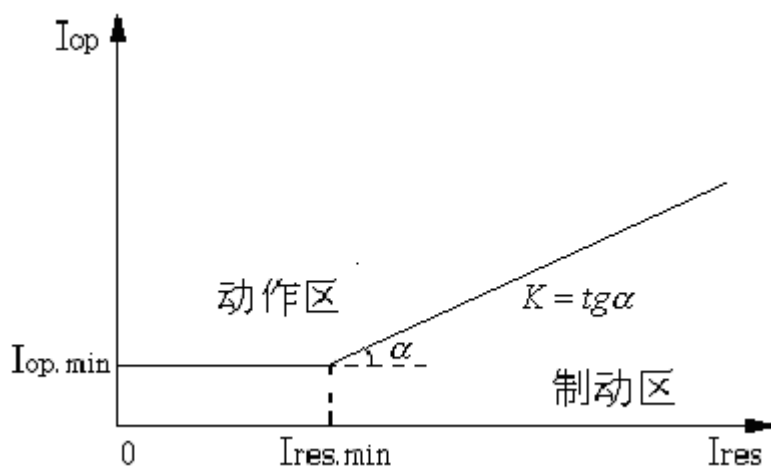


图 3-1 变压器差动保护的動作特性

在图 3-1-1 中, $I_{op.min}$ 为差动元件起始動作電流幅值, 也称为最小動作電流; $I_{res.min}$ 为最小制動電流, 又称为拐点電流; $K = tg\alpha$ 为制動特性斜率, 也称比率制動系数。微机变压器差动保护的差动元件采用分相差动, 其動作具有比率制動特性。動作方程为:

$$\text{拐点前 (含拐点): } I_{op} \geq I_{op.min} \quad (I_{res} \leq I_{res.min}) \quad (3-1-1)$$

$$\text{拐点后: } I_{op} \geq I_{op.min} + K(I_{res} - I_{res.min}) \quad (I_{res} > I_{res.min}) \quad (3-1-2)$$

式中 I_{op} —— 差動電流的幅值;

I_{res} —— 制動電流幅值。

二次谐波制動算法: 存在几种鉴别励磁涌流和内部故障的方法: 1) 直流制動。利用励磁涌流中含有大量直流分量这一特点, 但内部故障时暂态短路電流中也存在非周期分量, 这势必导致保护動作時間延迟; 2) 间断角比较原理。利用励磁涌流的间断角大于故障電流的间断角这一特点。该原理缺点之一就是受电源谐波影响较大, 之二就是由于超高压大容量变压器励磁涌流畸变率减少, 引起间断角减小; 3) 磁通制動原理。只要测知铁心是否饱和, 就可判别是涌流还是故障。但由于该法需要得到互感磁链的电压, 若得不到则无法采用; 4) 二次谐波制動原理。利用涌流信号比故障信号中含有较高百分比的二次谐波来制動。按三相“或”电路方式构成其谐波制動部分, 谐波制動比可整定为

15%~20%。因为对二次谐波的精度要求较高，从故障中精确滤出二次谐波，需要一个周波时间。这种延迟作用，不能适应大型变压器的快速要求。为加速其内部故障跳闸，利用微机的记忆功能，进行变压器涌流条件的快速判定。

3.1.2 差动电流及制动电流的取得

1) 差动电流(动作电流)的取得

变压器差动保护的差动电流（即动作电流），取各侧差动电流互感器 TA 二次电流相量和的绝对值。对于双卷变压器

$$I_{op} = |\dot{I}_h + \dot{I}_l| \quad (3-1-3)$$

对于三卷变压器或引入三侧电流的变压器

$$I_{op} = |\dot{I}_h + \dot{I}_m + \dot{I}_l| \quad (3-1-4)$$

式中 \dot{I}_h 、 \dot{I}_m 、 \dot{I}_l ——为变压器高、中、低侧 TA 二次电流。

2) 制动电流的取得

在微机保护中，变压器制动电流的取得方法比较灵活。对于双卷变压器，国内微机保护有以下几种取得方式：

①高、低压侧 TA 二次电流相量差的一半

$$I_{res} = |\dot{I}_h - \dot{I}_l|/2 \quad (3-1-5)$$

②高、低压侧 TA 二次电流幅值和的一半

$$I_{res} = (|\dot{I}_h| + |\dot{I}_l|)/2 \quad (3-1-6)$$

③高、低压侧 TA 二次电流幅值的最大值

$$I_{res} = \max\{|\dot{I}_h|, |\dot{I}_l|\} \quad (3-1-7)$$

④动作电流幅值与高、低压侧 TA 二次电流幅值之差的一半

$$I_{res} = (|I_{op}| - |\dot{I}_h| - |\dot{I}_l|)/2 \quad (3-1-8)$$

⑤低压侧 TA 二次电流的幅值

$$I_{res} = |\dot{I}_l| \quad (3-1-9)$$

对于三卷变压器，国内微机保护有以下取得方式：

①高、中、低压侧 TA 二次电流幅值和的一半

$$I_{\text{res}} = (|\dot{I}_h| + |\dot{I}_m| + |\dot{I}_l|) / 2 \quad (3-1-10)$$

②高、中、低压侧 TA 二次电流幅值的最大值

$$I_{\text{res}} = \max\{|\dot{I}_h|, |\dot{I}_m|, |\dot{I}_l|\} \quad (3-1-11)$$

③动作电流幅值与高、中、低压侧 TA 二次电流幅值之差的一半

$$I_{\text{res}} = (|\dot{I}_{op}| - |\dot{I}_h| - |\dot{I}_m| - |\dot{I}_l|) / 2 \quad (1-1-12)$$

④中、低压侧 TA 二次电流的幅值的最大值

$$I_{\text{res}} = \max\{|\dot{I}_m|, |\dot{I}_l|\} \quad (3-1-13)$$

注意，无论是双卷变还是三卷变，电流都要折算到同一侧进行计算和比较。

3.2 电流互感器二次电流相位补偿

电力系统中变压器常采用 Y, d11 接线方式，因此，变压器两侧电流的相位差为 30° 。如果不采取措施，差流回路中将会由于变压器两侧电流相位不同而产生不平衡电流。必需消除这种不平衡电流。方式有通过二次接线及通过保护装置内部两种，称为“外转角”及“内转角”两种方式

3.2.1 电流互感器二次接线进行相位补偿（外转角）

相位补偿的具体方法是：变压器星形侧的电流互感器二次绕组首尾相接成三角形，将变压器三角形侧的电流互感器二次绕组接成星形，如图 3-2-1 所示。

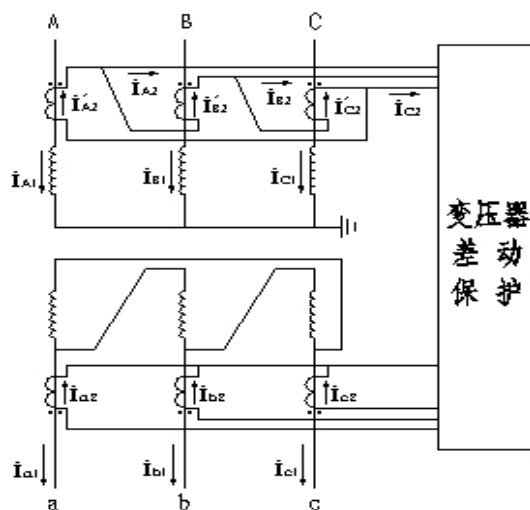


图 3-2 Y，d11 型变压器相位补偿接线图

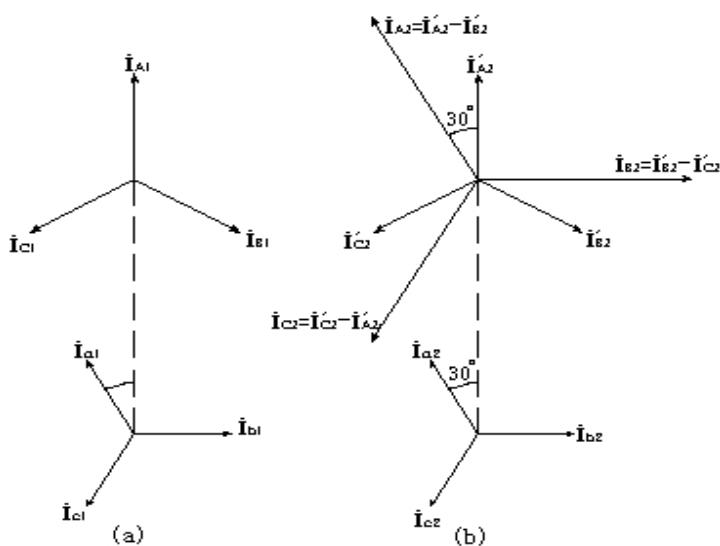


图 3-3 Y，d11 型变压器相位补偿相量图

采用相位补偿后，变压器星形侧电流互感器二次回路差动臂中的电流 i_{a2} 、 i_{b2} 、 i_{c2} ，刚好与三角形侧的电流互感器二次回路中的电流 i_{a2} 、 i_{b2} 、 i_{c2} 同相位，如图 3-3 所示。

3.2.2 用保护内部算法进行相位补偿（内转角）

当变压器各侧电流互感器二次均采用星型接线时，可简化 TA 二次接线，增加了电流回路的可靠性。如图 3-4 示，当变压器为 Y，d11 连接时，图 3-5 (a) 示出了 TA 原边的电流相量图，为消除各侧 TA 二次电流之间的 30° 角度

差，必须由保护软件通过算法进行调整，称为“内转角”。主要有由星形侧向角形侧调整以及角形侧向星形侧调整两种方式。

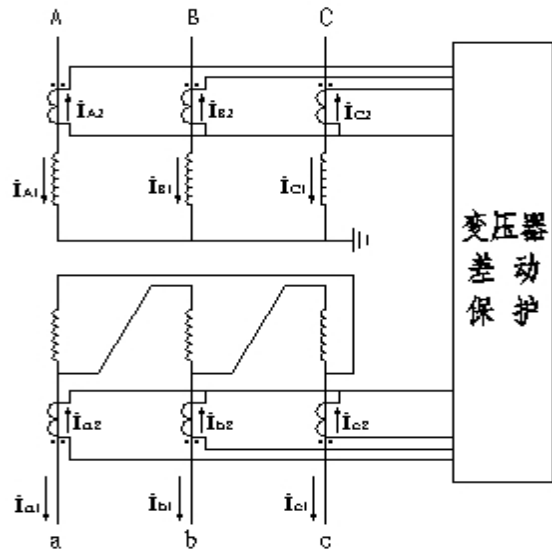


图 3-4Y，d11 变压器 TA 为 Y/Y 连接时差动保护原理接线

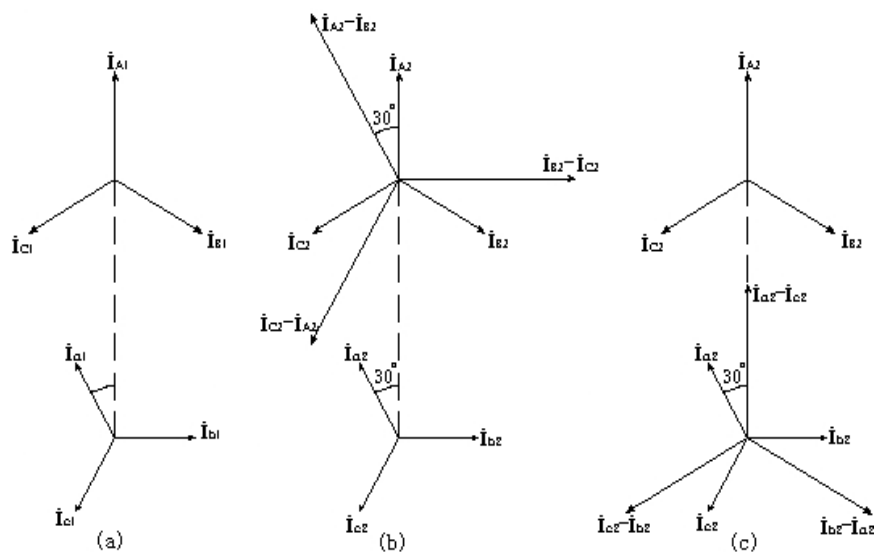


图 3-5 Y，d11 变压器 TA 为 Y/Y 连接时相位补偿相量图

(a) TA 原边电流相量 (b) 星型侧向角形侧调整 (c) 角型侧向星形侧调整

1) 星形侧向角形侧（称 Y→△）调整的算法

大部分保护装置采用星形侧向角形侧（称 Y→△变化）调整差流平衡，如四方的 CST31、南自厂的 PST-1200、WBZ-500H、南瑞的 LFP-972、RCS-985 等，其校正方法如下：

$$\begin{aligned}\text{星形侧: } \dot{I}'_{A2} &= (\dot{I}_{A2} - \dot{I}_{B2}) / \sqrt{3} \\ \dot{I}'_{B2} &= (\dot{I}_{B2} - \dot{I}_{C2}) / \sqrt{3}\end{aligned}\quad (3-2-1)$$

$$\dot{I}'_{C2} = (\dot{I}_{C2} - \dot{I}_{A2}) / \sqrt{3}$$

$$\begin{aligned}\text{角形侧: } \dot{I}'_{a2} &= \dot{I}_{a2} \\ \dot{I}'_{b2} &= \dot{I}_{b2}\end{aligned}\quad (3-2-2)$$

$$\dot{I}'_{c2} = \dot{I}_{c2}$$

式中： \dot{I}_{A2} 、 \dot{I}_{B2} 、 \dot{I}_{C2} 为星形侧 TA 二次电流， \dot{I}'_{A2} 、 \dot{I}'_{B2} 、 \dot{I}'_{C2} 为星形侧校正后的各相电流； \dot{I}_{a2} 、 \dot{I}_{b2} 、 \dot{I}_{c2} 为角形侧 TA 二次电流， \dot{I}'_{a2} 、 \dot{I}'_{b2} 、 \dot{I}'_{c2} 为角形侧校正后的各相电流。

经过软件校正后，差动回路两侧电流之间的相位一致，见图 3-5（b）所示。同理，对于三绕组变压器，若采用 Y，Y，d11 接线方式，星形侧的相位校正方法都是相同的。

2) 角形侧向星形侧（称 $\Delta \rightarrow Y$ ）调整的算法

南瑞 RCS-978 中电流互感器二次电流的相位校正方法与其它微机变压器保护有所不同，此保护装置采用角形侧向星形侧变化（称 $\Delta \rightarrow Y$ ）调整差流平衡，其校正方法如下：

$$\begin{aligned}\text{星形侧: } \dot{I}'_{A2} &= (\dot{I}_{A2} - \dot{I}_0) \\ \dot{I}'_{B2} &= (\dot{I}_{B2} - \dot{I}_0)\end{aligned}\quad (3-2-3)$$

$$\dot{I}'_{C2} = (\dot{I}_{C2} - \dot{I}_0)$$

$$\begin{aligned}\text{角形侧: } \dot{I}'_{a2} &= (\dot{I}_{a2} - \dot{I}_{c2}) / \sqrt{3} \\ \dot{I}'_{b2} &= (\dot{I}_{b2} - \dot{I}_{a2}) / \sqrt{3}\end{aligned}\quad (3-2-4)$$

$$\dot{I}'_{c2} = (\dot{I}_{c2} - \dot{I}_{b2}) / \sqrt{3}$$

式中： \dot{I}_{A2} 、 \dot{I}_{B2} 、 \dot{I}_{C2} 为星形侧 TA 二次电流， \dot{I}'_{A2} 、 \dot{I}'_{B2} 、 \dot{I}'_{C2} 为星形侧校正后的各相电流； \dot{I}_{a2} 、 \dot{I}_{b2} 、 \dot{I}_{c2} 为角形侧 TA 二次电流， \dot{I}'_{a2} 、 \dot{I}'_{b2} 、 \dot{I}'_{c2} 为角形侧校正后的各相电流。 \dot{I}_0 为星形侧零序二次电流。

经过软件校正后，差动回路两侧电流之间的相位一致，见图 3-5（c）所

示。同理，对于三绕组变压器，若采用 Y， Y， d11 接线方式，星形侧的软件算法都是相同的，角形侧同样进行相位校正。

3.3 CT 二次断线

CT 二次断线应考虑：

- 1) 单相、两相、三相完全和不完全断线及中性线完全断线；
- 2) 断线发生的时间是运行前还是运行中；
- 3) 断线发生时负荷大小对判据的影响。

单相、两相、三相不完全断线的特点：电流的下降有个过渡过程，且故障支路仍有部分电流流入保护装置。

中性线完全断线的特点：由于正常运行时，CT 中性线上的电流基本为零，故其断线时，流入保护装置的电流不发生变化。

几种不同的 CT 二次断线判据：

判据 1

- 1) 电流突变后，电流减小 $0.1I_n$ 且本侧三相中一相无电流，其它各侧三相电流无变化。
- 2) 本侧三相中一相无电流，其它两相电流与启动前相同

满足上述两个条件时，认为发生 CT 二次断线。

判据 2

当任一相差动电流大于 $0.15I_n$ 时，启动 CT 断线判别程序，满足下列条件认为 CT 断线：

- 1) 本侧三相电流中至少一相电流为零；
- 2) 本侧三相电流中至少一相电流不变；
- 3) 最大相电流小于 1.2 倍的额定电流。

判据 3

该判据具有延时 CT 断线报警及瞬时 CT 断线闭锁或报警功能，其动作判据如下：

- 1) 延时 CT 断线报警在保护每个采样周期内进行判别，在任一相差电流大于 $0.06I_n$ 的时间超过 10s 时，发出 CT 断线报警信号，此时不闭锁纵差保护，

这也兼起保护装置交流采样回路的自检作用。

2) 瞬时 CT 断线闭锁或报警功能在比率差动元件动作后进行判别。两侧电流同时满足下列条件则认为是 CT 断线：1，一侧负序电流大于 $0.1I_n$ ，另一侧负序电流小于 $0.06I_n$ ；2，负序电流大于 $0.1I_n$ 侧的最小相电流，远小于最大相电流。

3) 为防止瞬时 CT 断线的误闭锁，满足下述任一条件可不进行瞬时 CT 断线的判别：1，启动前各侧最大相电流小于 $0.08I_n$ ；2，启动后最大相电流大于对称过负荷报警定值；3，启动后电流比启动前增加。

判据 4

其动作判据为：

1) 增量 i （相分量） $\geq 0.1I_n$ 且 $1I_{H1} < 1I_{Q1}$ ；

2) 相电流 $\leq I_{W1}$ 且 $I_D \geq I_{W1}$ ；

3) 本侧 $1I_a + 1I_b + 1I_c \geq I_{W1}$ ；

4) $\text{Max}(I_{da}, I_{db}, I_{dc}) > 0.33I_{cd}$

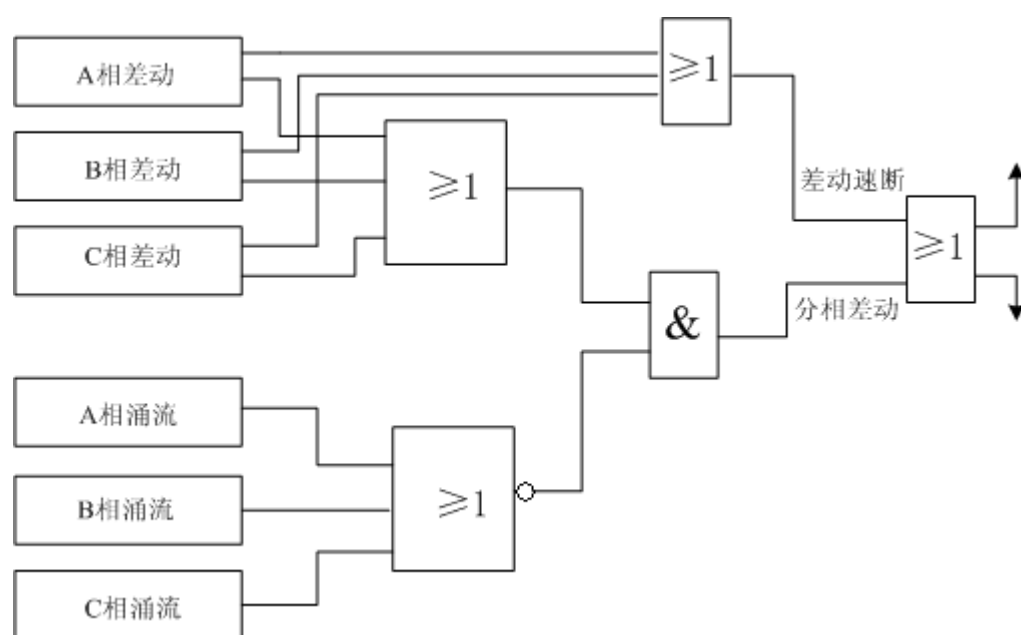
其中增量 I （相分量）为相电流突变量；为 A、B、C 三相差流值； I_{cd} 为纵差保护定值； I_n 为额定电流； I_Q 为前一次测量电流； I_H 为当前测量电流； I_D 为无流相的差动电流； I_{W1} 为无电流门槛值，取 0.04 倍的 CT 额定电流。

以上条件同时满足判 CT 断线，仅条件 4) 满足，判为差流越限。

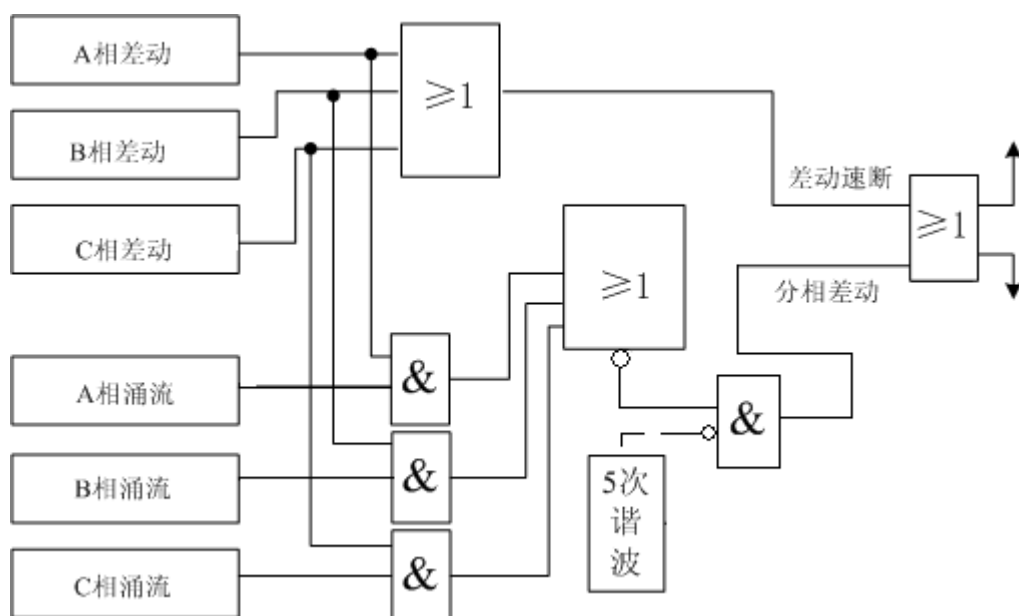
发生 CT 二次断线时，如不采取任何措施，当负荷电流大于纵差保护的启动定值时，纵差保护必然会动作。

3.4 逻辑构成框图

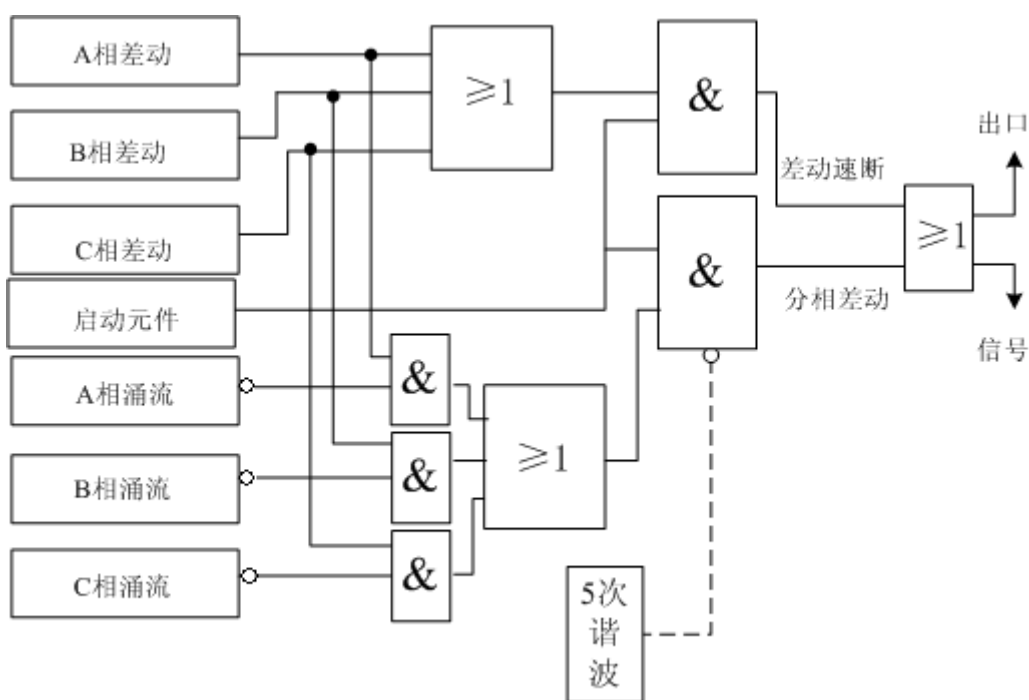
目前，在国内生产及应用的微机变压器保护装置，其差动部分逻辑框图如图 3-6 中的 (a)、(b)、(c) 所示。



(a) 差动逻辑 1



(b) 差动逻辑 2



(c) 差动逻辑 3

图 3-6 变压器差动保护逻辑框图

由图可见，图 3-6（a）表示 A、B、C 三相中任何一相都具有励磁涌流制动功能；图 3-6（b）表示涌流分相制动（或判别）；图 3-6（c）相对于图 3-6（b）增加了启动元件，图中虚线表示有的装置未设 5 次谐波闭锁。

第四章 变压器差动保护仿真

4.1 保护仿真软件概述

VB(Visual Basic) 语言是由Microsoft 公司开发的新一代面向对象编程工具，它既是一种编程语言，同时也是一个强大的集成开发环境，利用它可以编制出各种风格的软件产品。是Windows 下简单、易学、高效的可视化软件平台。然而用VB语言实现复杂的算法工作量十分庞大。Matlab 是MathWorks 公司推出的一套高性能的数值计算和可视化软件。Matlab 6. 5 中包含有 Simulink及功能强大的仿真电力系统(SimPowerSystems)模块库，它的功能非

常强大，含有电路、电力电子系统、电机系统、电力传输等电工科学中常用的基本元件和系统的仿真模型，建模只需点击和拖拉即可完成。利用MATLAB进行继电保护原理及装置的计算机仿真是当今高校及科研机构学习研究新型保护装置的重要手段之一，但限于 Matlab软件自身局限，利用电力系统工具箱所搭建的继电保护仿真模型界面不够友好，人机交互相对复杂。如将Matlab与VB的优势结合起来，取长补短，进行无缝链接可使编程更快捷、更专业、更灵活。

本文根据以上思路，利用ActiveX技术，成功实现Matlab与VB数据交互。这样我们利用Matlab强大的仿真功能，设计出仿真程序核心。利用VB编写界面简便的优势，绘制了仿真界面。这样就形成了界面友好，交互性强的工程仿真软件。

4.2 软件设计步骤

第一步，利用 Simulink 中的 SimPowerSystems 工具箱构建一个简单的单电源供电系统，并在 Matlab 环境中调试成功。

第二步，根据变压器差动保护的比率制动与二次谐波制动原理构建变压器差动保护 S-FUNCTION 模块，并结合前面构建的供电系统进行调试。再与整定值进行配合，使保护能有选择性的动作满足实际要求。

第三步，利用 ActiveX 技术使 Matlab 与 VB 进行链接，实现两者间的数据传递。在 VB 中编写仿真界面。

4.3 一次供电系统模型

图中变压器为例，在变压器区内，区外分别设置了故障发生器，并设置变压器差动保护，“差动保护”模块即为所建变压器差动保护模型。所建最外级模型如图 4-1 所示。

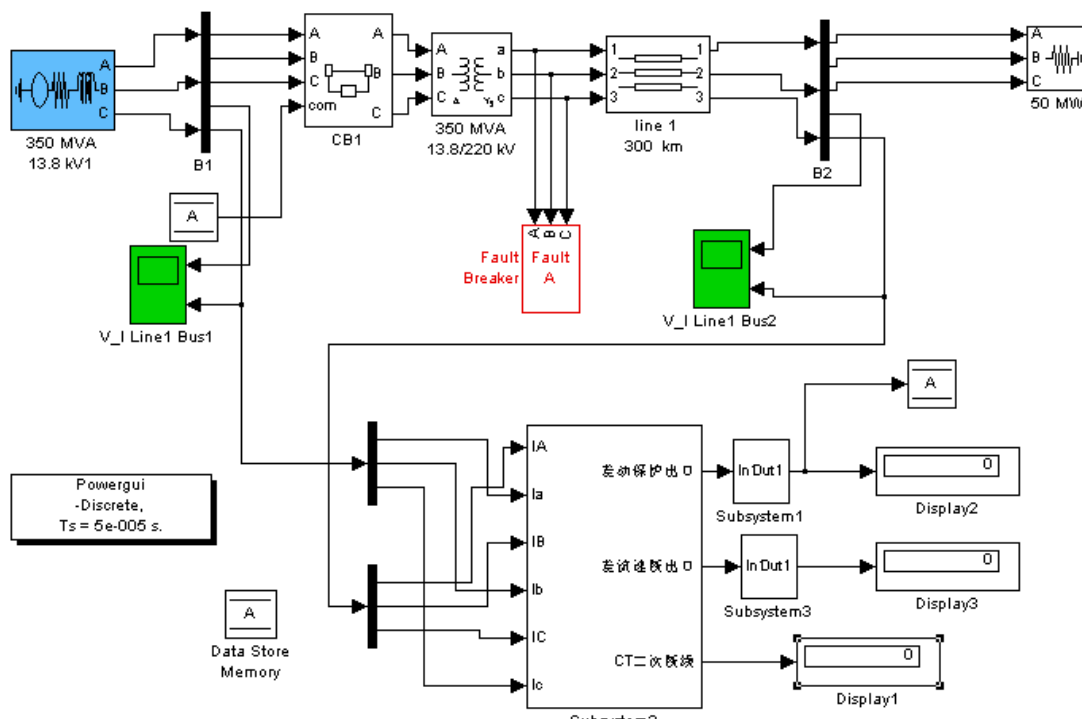


图 4-1 差动保护仿真最外级模型

该模型各个模块参数设置如下：（只需双击模块即可实现修改参数）

1) “无穷大系统”模块实际为一个三相电源模块(Three phase Source block),通过设定该模块的参数来模拟无穷大系统。三相交流电源模块参数(系统):

表 4-1

三相电源模块参数	无穷大系统
Phase-to-phase rms voltage(V)	$13.8\text{e}3/\text{sqrt}(6)$
Phase angle of phase A(degrees)	0
Frequency(Hz)	50
Internal connection	Yg
3-phase short-circuit level at base voltage(VA)	$350\text{e}6$
Bse voltage(Vrms ph-ph)	$13.8^2/350/10$
X/R ratio	$13.8^2/350/377$

2) 三相测量模块（B1）的参数如表 4-2:

表 4-2

三相测量模块参数	B1
Voltage measurement	phase-to-ground
Voltage in pu	<input checked="" type="checkbox"/>
Base voltage(Vrms phase-phase)	220e3
Current measurement	yes
Currents in pu	<input checked="" type="checkbox"/>
Base power(VA 3 phase)	100e6

3) 断路器模块“break”参数如表 4-3 所示。

表 4-3

断路器模块参数	Break
Initial status of breaks	close
Switching of phase A	<input checked="" type="checkbox"/>
Switching of phase A	<input checked="" type="checkbox"/>
Switching of phase A	<input checked="" type="checkbox"/>
External control of switching times	<input checked="" type="checkbox"/>
Breakers resistance Ron(ohms)	0.001
Snubbers resistance Rp(ohms)	1e-5
Snubbers capacitance Cp(Farad)	inf
Measurements	None

4) 线路模块参数如表 4-4 所示

表 4-4

线路模块参数	Line 1 300 km
Number of phases N	3
Frequenvy(Hz)	50
Resistance(Ohms/km)	[0.01273

	0.3864]
Inductance (H/km)	[0.9337e-3 4.1264e-3]
Capacitance (H/km)	[12.74e-9 7.751e-9]
Line length(km)	300
Measurements	None

5) 变压器模块参数如表 4-5 所示。

表 4-5

变压器模块参数	350 MVA 13.8/220kV
[Pn(VA) , fn(Hz)]	[350e6 , 50]
Winding1(ABC) connection	Delta(D1)
[V1 Ph-Ph(Vrms), R1(pu), L1(pu)]	[13.8e3 , 0.002 , 0.08]
Winding2(abc-2) connection	Yg
[V2 Ph-Ph(Vrms), R2(pu), L2(pu)]	[220e3 , 0.002 , 0.08]
Magnetization resistance Rm(pu)	500
Magnetization reactance Lm(pu)	500
Measurements	None

6) 负荷模块参数如表 4-6 所示。

表 4-6

三相串联 RLC 负载模块参数	50MW
Nominal phase-phase	220e3

voltage(Vrms)	
Niminal frequency(Hz)	50
P(W)	50e6
Q_L (var)	0
Q_C (var)	0

4.4 保护仿真模型

本设计中运用的逻辑是第三章中的图 3-6（a）差动逻辑 1.

4.4.1 “差动保护” 模块子系统 (Subsystem)

“差动保护” 模块的子系统如下图 4-2 所示：

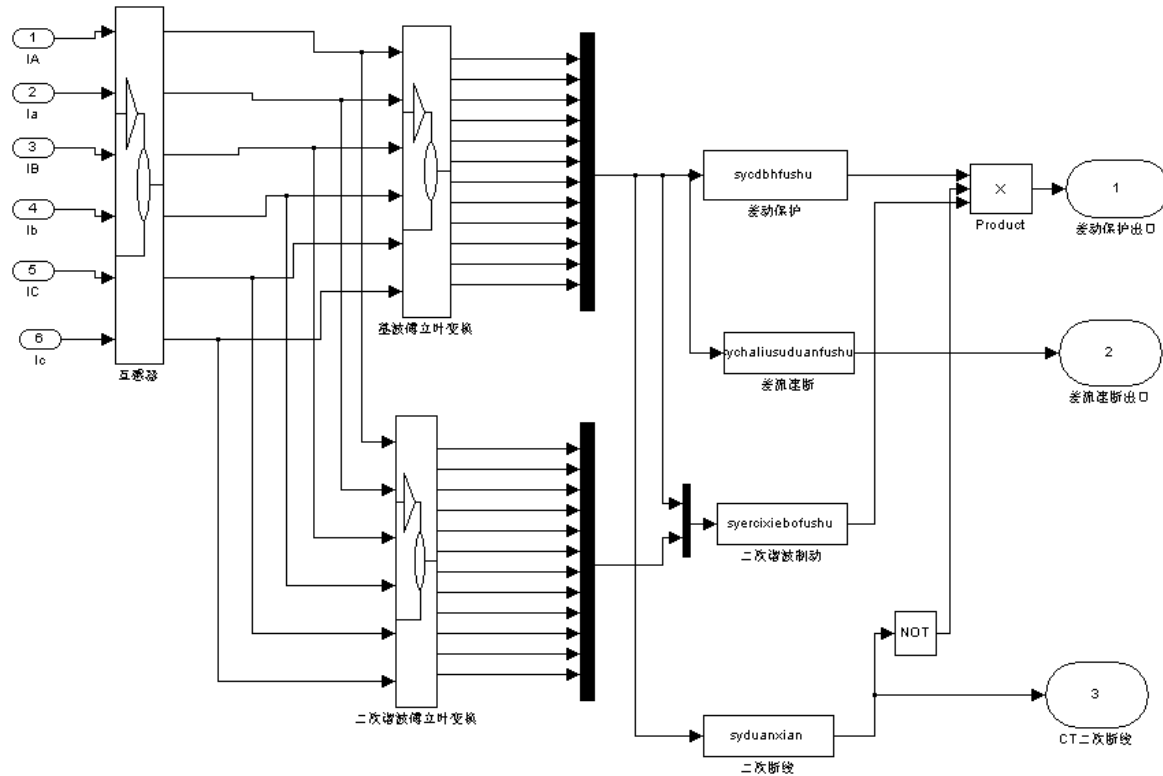


图 4-2 “差动保护” 模块的子系统

其中又包括了三个子系统（Subsystem）：互感器、基波傅立叶变换、二次谐波傅立叶变换。还有就是两个分路器（“Demux”）。三个子系统如下介绍：

1) 互感器

此子系统首先由三相 V-I 测量模块采集变压器高、低压侧流过的三相电流，分别经过一放大器（“Gain”），相当于电流互感器把变压器一次侧电流变换到二次侧。如图 4-3 所示。

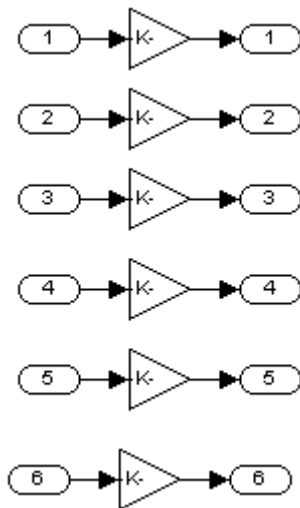


图 4-3 互感器

中间的模块即为放大器（“Gain”）。1、3、5 即 IA、IB、IC 高压侧电流的放大倍数为 1/0.69994，2、4、6 即 Ia、Ib、Ic 低压侧电流的放大倍数为 1/11.15868

2) 基波傅立叶变换

傅立叶模块的作用是输出各相电流的基波和幅值，从而输入 S-FUNCTION 模块中使运算中通过公式 $A + jB = I^* \cos \varphi + j I^* \sin \varphi$ ，把电流变换成复数形式进行运算。如图 4-4 所示。

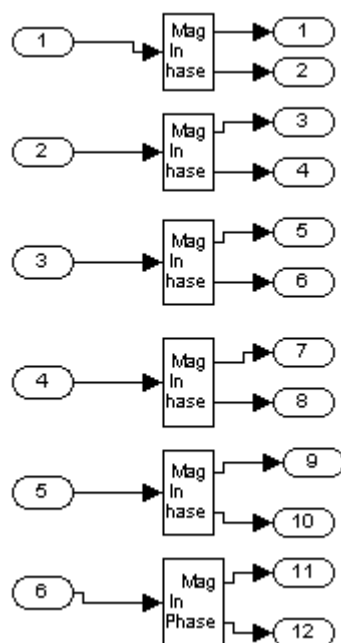


图 4-4 基波傅立叶变换

傅立叶变换模块的参数设置为：Fundamental frequency :50、Harmonic n :1、Initial input:[1.1 -30]、Sample time :50e-5。

3) 二次谐波傅立叶变换

二次傅立叶模块的作用是输出各相电流二次谐波的基波和幅值，从而输入 S-FUNCTION 模块中使运算中通过公式 $A + jB = I \cdot \cos \varphi + jI \cdot \sin \varphi$ ，把电流变换成复数形式进行运算。图形如图 4-4 与基波傅立叶变换一样。

参数设置为：Fundamental frequency :50、Harmonic n :2、Initial input:[1.1 -30]、Sample time :50e-5。

4.4.2. 变压器差动保护跳闸的实现

差动保护跳闸模型可分为主要的两个部分（四个 S-FUNCTION）：第一部分为实现主功能分相差动保护跳闸，其中包括比率制动分相差动保护和二次谐波制动以及 CT 二次断线闭锁三个 S-FUNCTION，三者都满足跳闸条件时实现跳闸；第二部分为差流速断保护跳闸。由于变压器的接线方式为 D/Y-1 形式的，所以在程序中对高压侧电流进行了“转角”，且此模型中为升压变，运用公式：

$$I_A' = (I_A - I_C) / \sqrt{3}$$

$$I_B' = (I_B - I_A) / \sqrt{3} \text{ 以使两侧电流相角一致。}$$

$$I_C' = (I_C - I_B) / \sqrt{3}$$

1) 第一部分（分相差动）

第一部分将变压器二次侧电流输入到封装有 M 文件的 S-Function 中，利用程序进行保护跳闸的判断。

此部分包括有三个封装不同功能的 M 文件程序的 S-Function，由一个比率制动分相差动、一个二次谐波制动、一个 CT 二次断线闭锁组成。

① 比率制动分相差动

比率制动分相差动的作用是利用分相比率制动判据实现差动保护跳闸。差动元件的动作特性较多采用具有二段折线型的动作特性曲线，动作判据为：

$$\begin{aligned} \text{拐点前（含拐点）：} & I_{op} \geq I_{op.min} \quad (I_{res} \leq I_{res.min}) \\ \text{拐点后：} & I_{op} \geq I_{op.min} + K(I_{res} - I_{res.min}) \quad (I_{res} > I_{res.min}) \end{aligned}$$

式中 I_{op} ——差动电流的幅值；

I_{res} ——制动电流的幅值。

M 文件程序为：

```
function [sys,x0,str,ts] = chadongbhsh(t,x,u,flag,Iopmin,Kres,Iresmin)
Iopmin=2; %给出整定值
Kres=0.5;
Iresmin=5;
switch flag, %六个子程序
case 0
    [sys,x0,str,ts]=mdlInitializeSizes(Iopmin,Kres,Iresmin);%初始化
case 2,
    sys=mdlUpdate(t,x,u); %更新离散状态、采样时间和主时间步的要求
case 3
    sys=mdlOutputs(t,x,u,Iopmin,Kres,Iresmin);% 计算 s-function 的输出
case 9,
    sys = []; % do nothing
```

otherwise

error(['Unhandled flag = ',num2str(flag)]);

end

function [sys,x0,str,ts] = mdlInitializeSizes(Iopmin,Kres,Iresmin)

sizes = simsizes;

sizes.NumContStates = 0;

sizes.NumDiscStates = 6;

sizes.NumOutputs = 1; % dynamically sized

sizes.NumInputs = 12; % dynamically sized

sizes.DirFeedthrough = 0; % has not direct feedthrough

sizes.NumSampleTimes = 1;

sys = simsizes(sizes);

str = [];

x0 = [0];

ts = [-1 0]; % inherited sample time

function sys=mdlUpdate(t,x,u)

%u 为输入量，共有 12 个输入量，x 为中间离散变量，共有 6 个，分别为：

%u1 为 a 相高压侧电流幅值,u2 为 a 高相位,u3 为 a 相低压侧电流幅值，

%u4 为 a 低相位,u5 为 b 相高压侧电流幅值，u6 为 b 高相位

%u7 为 b 相低压侧电流幅值,u8 为 b 低相位,u9 为 c 相高压侧电流幅值，

%u10 为 c 高相位,u11 为 c 相低压侧电流幅值,u12 为 c 低相位

%x1 为 a 相差动电流,x2 为 a 相制动电流,x3 为 b 相差动电流

%x4 为 b 相制动电流,x5 为 c 相差动电流,x6 为 c 相制动电流

sys(1)=abs((u(1)*cos(u(2))+u(1)*i*sin(u(2))-u(9)*cos(u(10))-u(9)*i*sin(u(10)))/

sqrt(3)-u(3)*cos(u(4))+u(3)*i*sin(u(4)));

%转化为复数形式进行运算后得到 A 相差动电流

```

sys(2)=abs((u(1)*cos(u(2))+u(1)*i*sin(u(2))-u(9)*cos(u(10))-u(9)*i*sin(u(10)))/
sqrt(3)+u(3)*cos(u(4))+u(3)*i*sin(u(4)))/2)
%转化为复数形式进行运算后得到 A 相制动电流
sys(3)=abs((u(5)*cos(u(6))+u(5)*i*sin(u(6))-u(1)*cos(u(2))-u(1)*i*sin(u(2)))/sqrt
(3)-u(7)*cos(u(8))+u(7)*i*sin(u(8)))
%转化为复数形式进行运算后得到 B 相差动电流
sys(4)=abs((u(5)*cos(u(6))+u(5)*i*sin(u(6))-u(1)*cos(u(2))-u(1)*i*sin(u(2)))/sqrt
(3)+u(7)*cos(u(8))+u(7)*i*sin(u(8)))/2)
%转化为复数形式进行运算后得到 B 相制动电流
sys(5)=abs((u(9)*cos(u(10))+u(9)*i*sin(u(10))-u(5)*cos(u(6))-u(5)*i*sin(u(6)))/s
qrt(3)-u(11)*cos(u(12))+u(11)*i*sin(u(12)))
%转化为复数形式进行运算后得到 C 相差动电流
sys(6)=abs((u(9)*cos(u(10))+u(9)*i*sin(u(10))-u(5)*cos(u(6))-u(5)*i*sin(u(6)))/s
qrt(3)+u(11)*cos(u(12))+u(11)*i*sin(u(12)))/2)
%转化为复数形式进行运算后得到 C 相制动电流

function sys=mdlOutputs(t,x,u,Iopmin,Kres,Iresmin)
%利用差动保护的判剧进行判别得到是否动作
if
(x(1)>Iopmin&(x(1)-Iopmin)>Kres*(x(2)-Iresmin))|((x(3)>Iopmin&(x(3)-Iopmin)
>Kres*(x(4)-Iresmin))|(x(5)>Iopmin&(x(5)-Iopmin)>Kres*(x(6)-Iresmin)))
    sys=1;
else
    sys=0;
end

```

②二次谐波制动

二次谐波制动的作用是当二次谐波占基波的 15%~35%时,使保护制动,不跳闸。二次谐波制动原理。利用涌流信号比故障信号中含有较高百分比的二次谐波来制动。按三相“或”电路方式构成其谐波制动部分,谐波制动比可整定为 15%~20%。

M 文件程序为：

```
function [sys,x0,str,ts] = ercixiebo(t,x,u,flag,K2)
K2=0.15;
switch flag,
    case 0
        [sys,x0,str,ts]=mdlInitializeSizes(K2);
    case 2,
        sys=mdlUpdate(t,x,u);

    case 3
        sys=mdlOutputs(t,x,u,K2);
    case 9,
        sys = []; % do nothing
    otherwise
        error(['Unhandled flag = ',num2str(flag)]);

end

function [sys,x0,str,ts] = mdlInitializeSizes(K2)

sizes = simsizes;
sizes.NumContStates = 0;
sizes.NumDiscStates = 6;
sizes.NumOutputs = 1; % dynamically sized
sizes.NumInputs = 24; % dynamically sized
sizes.DirFeedthrough = 0; % has not direct feedthrough
sizes.NumSampleTimes = 1;

sys = simsizes(sizes);
str = [];
```

```

x0 = [0];
ts = [-1 0]; % inherited sample time

% end mdlInitializeSizes

function sys=mdlUpdate(t,x,u)
%输入的有基波电流和二次谐波电流，分别为 u1a 相基波高压侧电流幅值，
%u2 为 a 高相角
%u3a 相基波低压侧电流幅值,u4 为 a 低相角
%u5b 相基波高压侧电流幅值,u6 为 b 高相角,u7b 相基波低压侧电流幅值，
%u8 为 b 低相角
%u9c 相基波高压侧电流幅值,u10 为 c 高相角,u11c 相基波低压侧电流幅值，
%u12 为 c 低相角
%u13a 相二次谐波高压侧电流幅值,u14 为 a 高相角，
%u15a 相二次谐波低压侧电流幅值,u16 为 a 低相角
%u17b 相二次谐波高压侧电流幅值,u18 为 b 高相角，
%u19b 相二次谐波低压侧电流幅值,u20 为 b 低相角
%u21c 相二次谐波高压侧电流幅值,u22 为 c 高相角，
%u23c 相二次谐波低压侧电流幅值,u24 为 c 低相角

%基波高低压侧电流
%x1 为 a 相基波差动电流,x2 为 b 相基波差动电流,x3 为 c 相基波差动电流
%x4 为 a 相二次谐波差动电流,x5 为 b 相二次谐波差动电流，
%x6 为 c 相二次谐波差动电流
%基波 A， B， C 三相的差动电流
sys(1)=abs((u(1)*cos(u(2))+u(1)*i*sin(u(2))-u(9)*cos(u(10))-u(9)*i*sin(u(10)))/sqrt(3)
-u(3)*cos(u(4))+u(3)*i*sin(u(4)))
sys(2)=abs((u(5)*cos(u(6))+u(5)*i*sin(u(6))-u(1)*cos(u(2))-u(1)*i*sin(u(2)))/sqrt(3)-u
(7)*cos(u(8))+u(7)*i*sin(u(8)));
sys(3)=abs((u(9)*cos(u(10))+u(9)*i*sin(u(10))-u(5)*cos(u(6))-u(5)*i*sin(u(6)))/sqrt(3)
-u(11)*cos(u(12))+u(11)*i*sin(u(12)));

```

%二次谐波 A, B, C 三相的差动电流

```
sys(4)=abs((u(13)*cos(u(14))+u(13)*i*sin(u(14))-u(21)*cos(u(22))-u(21)*i*sin(u(22)))  
/sqrt(3)-u(15)*cos(u(16))+u(15)*i*sin(u(16)))  
sys(5)=abs((u(17)*cos(u(18))+u(17)*i*sin(u(18))-u(13)*cos(u(14))-u(13)*i*sin(u(14)))  
/sqrt(3)-u(19)*cos(u(20))+u(19)*i*sin(u(20)));  
sys(6)=abs((u(21)*cos(u(22))+u(21)*i*sin(u(22))-u(17)*cos(u(18))-u(17)*i*sin(u(18)))  
/sqrt(3)-u(23)*cos(u(24))+u(23)*i*sin(u(24)));
```

```
function sys=mdlOutputs(t,x,u,K2)
```

%根据二次谐波制动条件进行判别 $I_{2cd} > K2 * I_{1cd}$ 时对差动保护进行闭锁

```
if (x(4)>K2*x(1))|(x(5)>K2*x(2))|(x(6)>K2*x(3))
```

```
    sys=0;
```

```
else
```

```
    sys=1;
```

```
end
```

③CT 二次断线闭锁

CT 二次断线闭锁的作用见第三章第三节 CT 二次断线。

M 文件程序为：

```
function [sys,x0,str,ts] =syduanxian(t,x,u,flag,Iopmin)
```

```
Iopmin=2;
```

```
switch flag,
```

```
    case 0
```

```
        [sys,x0,str,ts]=mdlInitializeSizes(Iopmin);
```

```
    case 2,
```

```
        sys=mdlUpdate(t,x,u);
```

```
    case 3
```

```
        sys=mdlOutputs(t,x,u,Iopmin);
```

```

    case 9,
        sys = []; % do nothing

    otherwise
        error(['Unhandled flag = ',num2str(flag)]);

end

function [sys,x0,str,ts] = mdlInitializeSizes(Iopmin)

sizes = simsizes;
sizes.NumContStates = 0;
sizes.NumDiscStates = 3;
sizes.NumOutputs = 1; % dynamically sized
sizes.NumInputs = 12; % dynamically sized
sizes.DirFeedthrough = 0; % has not direct feedthrough
sizes.NumSampleTimes = 1;

sys = simsizes(sizes);
str = [];
x0 = [0];
ts = [-1 0]; % inherited sample time

function sys=mdlUpdate(t,x,u)
%u1a 相基波高压侧电流幅值,u2 为 a 高相角
%u3a 相基波低压侧电流幅值,u4 为 a 低相角
%u5b 相基波高压侧电流幅值,u6 为 b 高相角,u7b 相基波低压侧电流幅值,
%u8 为 b 低相角
%u9c 相基波高压侧电流幅值,u10 为 c 高相角,u11c 相基波低压侧电流幅值,
%u12 为 c 低相角
%x1 为 a 相差动电流,x2 为 b 相差动电流

```

%x3 为 c 相差动电流

```
sys(1)=abs((u(1)*cos(u(2))+u(1)*i*sin(u(2))-u(9)*cos(u(10))-u(9)*i*sin(u(10)))/  
sqrt(3)-u(3)*cos(u(4))+u(3)*i*sin(u(4)));  
sys(2)=abs((u(5)*cos(u(6))+u(5)*i*sin(u(6))-u(1)*cos(u(2))-u(1)*i*sin(u(2)))/sqrt  
(3)-u(7)*cos(u(8))+u(7)*i*sin(u(8)));  
sys(3)=abs((u(9)*cos(u(10))+u(9)*i*sin(u(10))-u(5)*cos(u(6))-u(5)*i*sin(u(6)))/  
sqrt(3)-u(11)*cos(u(12))+u(11)*i*sin(u(12)));
```

```
function sys=mdlOutputs(t,x,u,Iopmin)
```

%不进行断线判别，满足下列任一条件不进行断线判别：

%A. 保护起动前某侧最大相电流小于 $(0.1 \sim 0.2) I_n$

%B. 起动后最大相电流大于 $(1.2 \sim 1.4) I_n$.

%进行断线闭锁判别

%某侧电流同时满足下列条件认为是 CT 断线

%A. 只有一相或两相电流为零。

%B. 其它两相或一相电流与起动电流相等。

if

```
(u(1)==0&u(5)==Iopmin&u(9)==Iopmin)|(u(5)==0&u(1)==Iopmin&u(9)==Iopmi  
n)|(u(9)==0&u(1)==Iopmin&u(5)==Iopmin)|(u(1)==0&u(5)==0&u(9)==Iopmin)|(  
u(5)==0&u(9)==0&u(1)==Iopmin)|(u(1)==0&u(9)==0&u(5)==Iopmin)|(u(3)==0&  
u(7)==Iopmin&u(11)==Iopmin)|(u(7)==0&u(3)==Iopmin&u(11)==Iopmin)|(u(11)  
==0&u(3)==Iopmin&u(7)==Iopmin)|(u(3)==0&u(7)==0&u(11)==Iopmin)|(u(7)==  
0&u(11)==0&u(3)==Iopmin)|(u(3)==0&u(11)==0&u(7)==Iopmin)
```

```
sys=1;
```

else

```
sys=0;
```

end

2) 第二部分（差流速断保护）

差流速断保护跳闸的判据为： $I_{cd} > K_{ds} \bullet I_n$ 。

M 文件程序为：

```
function [sys,x0,str,ts] = chadongbhsh(t,x,u,flag,Kds,In)
```

```
Kds=10;
```

```
In=5;
```

```
switch flag,
```

```
    case 0
```

```
        [sys,x0,str,ts]=mdlInitializeSizes(Kds,In);
```

```
    case 2,
```

```
        sys=mdlUpdate(t,x,u);
```

```
    case 3
```

```
        sys=mdlOutputs(t,x,u,Kds,In);
```

```
    case 9,
```

```
        sys = []; % do nothing
```

```
    otherwise
```

```
        error(['Unhandled flag = ',num2str(flag)]);
```

```
end
```

```
function [sys,x0,str,ts] = mdlInitializeSizes(Kds,In)
```

```
sizes = simsizes;
```

```
sizes.NumContStates = 0;
```

```
sizes.NumDiscStates = 3;
```

```
sizes.NumOutputs = 1; % dynamically sized
```

```
sizes.NumInputs = 12; % dynamically sized
```

```
sizes.DirFeedthrough = 0; % has not direct feedthrough
```

```
sizes.NumSampleTimes = 1;
```

```
sys = simsizes(sizes);
```

```
str = [];
```

```
x0 = [0];
```

```
ts = [-1 0]; % inherited sample time
```

```
function sys=mdlUpdate(t,x,u)
```

```
%u1 为 a 相高压侧电流幅值,u2 为 a 高相位,u3 为 a 相低压侧电流幅值,
```

```
%u4 为 a 低相位,u5 为 b 相高压侧电流幅值, u6 为 b 高相位
```

```
%u7 为 b 相低压侧电流幅值,u8 为 b 低相位,u9 为 c 相高压侧电流幅值,
```

```
%u10 为 c 高相位,u11 为 c 相低压侧电流幅值,u12 为 c 低相位
```

```
%x1 为 a 相差动电流,x2 为 b 相差动电流
```

```
%x3 为 c 相差动电流
```

```
sys(1)=abs((u(1)*cos(u(2))+u(1)*i*sin(u(2))-u(9)*cos(u(10))-u(9)*i*sin(u(10)))/  
sqrt(3)-u(3)*cos(u(4))+u(3)*i*sin(u(4))));
```

```
sys(2)=abs((u(5)*cos(u(6))+u(5)*i*sin(u(6))-u(1)*cos(u(2))-u(1)*i*sin(u(2)))/sqrt  
(3)-u(7)*cos(u(8))+u(7)*i*sin(u(8))));
```

```
sys(3)=abs((u(9)*cos(u(10))+u(9)*i*sin(u(10))-u(5)*cos(u(6))-u(5)*i*sin(u(6)))/  
sqrt(3)-u(11)*cos(u(12))+u(11)*i*sin(u(12))));
```

```
function sys=mdlOutputs(t,x,u,Kds,In)
```

```
%满足差流速断条件  $I_{cd} > K_{ds} * I_n$  则 0 秒跳闸
```

```
if (x(1)>Kds*In)|(x(2)>Kds*In)|(x(3)>Kds*In)
```

```
    sys=1;
```

```
else
```

```
    sys=0;
```

```
end
```

4.5 Matlab 与 VB 的接口

4.5.1 接口介绍

借助DDE技术：

DDE允许Microsoft Windows应用程序通过交换数据实现彼此的通信。由于VB支持DDE客户端功能，Matlab提供DDE服务功能，因而两者应用程序间通过建立一个DDE会话，VB端（客户端）对会话进行初始化，而Matlab（服务器端）对其作出响应。初始会话时VB应用程序必须指明Matlab的服务器名为Matlab，会话主题为Engine。这样使用Matlab的Engine主题支持的3种操作，VB应用程序可以向Matlab传送可执行的命令，从Matlab工作空间读取、传送数据。要注意的是，Matlab 的命令窗口必须在建立DDE会话之前已经运行。

借助ActiveX部件：

ActiveX部件是Microsoft公司提供的一种用于模块集成的新协议。Matlab支持ActiveX 部件的嵌入，它可以通过ActiveX 技术与VB作接口。Matlab提供了一个自动化对象，其外部名称为Matlab.Application，它支持COM技术。由于VB支持ActiveX 自动化控制端协议，Matlab支持ActiveX自动化服务器端协议，因此只要建立了VB应用程序和Matlab之间的ActiveX自动化连接，即可在一个应用程序中调用Matlab命令和直接从工作空间存取矩阵等。对Matlab自动化服务器的使用，必须通过服务器所提供的3种方法，即：Execute(用于执行一个合法的Matlab命令)、PutFullMatrix(用于从Matlab工作空间获取数据)和GetFullMatrix(用于向Matlab的工作空间输出一个指定名字的mxArray结构体)。

与DDE技术相比！它在运行时，不需要首先启动Matlab所以功能更为强大。在VB程序中创建ActiveX对象的具体步骤为：

```
Dim Matlab As Object
```

```
Dim Result As String
```

```
Set Matlab=CreatObject("Matlab.Application")
```

在VB应用程序中创建了ActiveX对象后，就可以使用ActiveX对象中所包含的方法来实现对Matlab的调用。ActiveX Automation， 对象中含有5个方法

(method)，它们分别是

(1)Excute: 执行Matlab命令;

(2)PutFullMatrix: 将VB应用程序中的一个数组的数值赋值给

CommandWindows中的一个矩阵变量;

(3)PutFullMatrix:将Matlab中的一个矩阵变量的值传送到VB应用程序的一个数组中;

(4)MinimizeCommandWindows: 使Matlab窗口最小化;

(5)MaxmimizeCommandWindows: 使CommandWindows窗口最大化。

应用上述方法可在VB应用程序中实现所有Matlab的功能。

4.5.2 VB 界面设计

文将采用ActiveX 部件的方法，从VB 中传递仿真命令给MATLAB 执行；然后借助DDE技术绘制出不同情况下的仿真波形。VB界面由1个PictureBox，1个框架frame，两个Option按钮和两个Commend按钮组成。PictureBox用来显示系统仿真图形Command1为执行键, Command2为系统退出键，界面如图4—5所示。Option 按钮组提供用户选择需要观察的发生单相故障类型过程三相电流、电压的波形变化以及动作情况。其中Option1为三相电压波形，Option2为三相电流波形，Option3为保护动作情况波形。图4-5为VB设计界面。



图 4-5 VB 界面

4.5.3 程序代码

在 command1 的双击事件中加入下面代码：

```
Private Sub Command1_Click()
    '创建 ActiveX 对象
    Dim Matlab As Object
    Dim result As String
    Set Matlab = CreateObject("Matlab.Application")
    '选择运行 Simulink 模型

    result = Matlab.Execute("sim('number2')")
    '初始化 Text1 与 MATLAB 的 DDE 会话
    '断开 Text1 与 MATLAB 服务器的连接
    Text1.LinkMode = 0
    '将控件 Text1 的 DDE 连接对象设置为 MATLAB|Engine
    Text1.LinkTopic = "MATLAB|Engine"
    Text1.LinkItem = "EngEvalString" "
```

```

'建立连接
Text1.LinkMode = 2
'根据用户选择的单选按钮，使 MATLAB 执行绘图命令
Select Case True
Case Option1.Value
number2 = " plot (Uabc ) ;axis([0,5000,-300000,300000]);title ('三相的电压
波形','Font size', 12 ,'FontName','黑体') )"
Case Option2.Value
number2 = " plot (Iabc ) ;axis([0,5000,-150000,150000]);title ('三相的电流
波形','Font size', 12 ,'FontName','黑体') )"
Case Option3.Value
number2 = " plot (A) ;axis([0,5000,-0.1,1.1]);title ('动作情况','Font size',
12 ,'FontName','黑体') )"
End Select
Text1.LinkExecute (number2)
'断开连接
Text1.LinkMode = 0
'执行命令的图形结果
Text1.LinkTopic = "MATLAB|Engine"
Text1.LinkItem = "EngFigureResult"
Text1.LinkMode = 2
Text1.LinkRequest
Text1.LinkMode = 0
'获取图形
If Text1.Text = "yes" Then
Picture1.Picture = Clipboard.GetData
End If
End Sub
Private Sub Command2_Click()
Unload Me

```

End Sub

4.5.4 设置故障与仿真结果

1) 故障设置

模拟变压器高压侧 A 相接地故障，总仿真时间为 0.2 秒，故障发生在 0.02 秒，故障发生器的模块参数设置见图 5-3 所示。

2) 保护动作结果

①在 VB 界面中选择第一个情况三相电压，且在 Text1 中键入 yes，然后单击仿真按钮，则可出现以下显示结果，如图 4-6 所示。

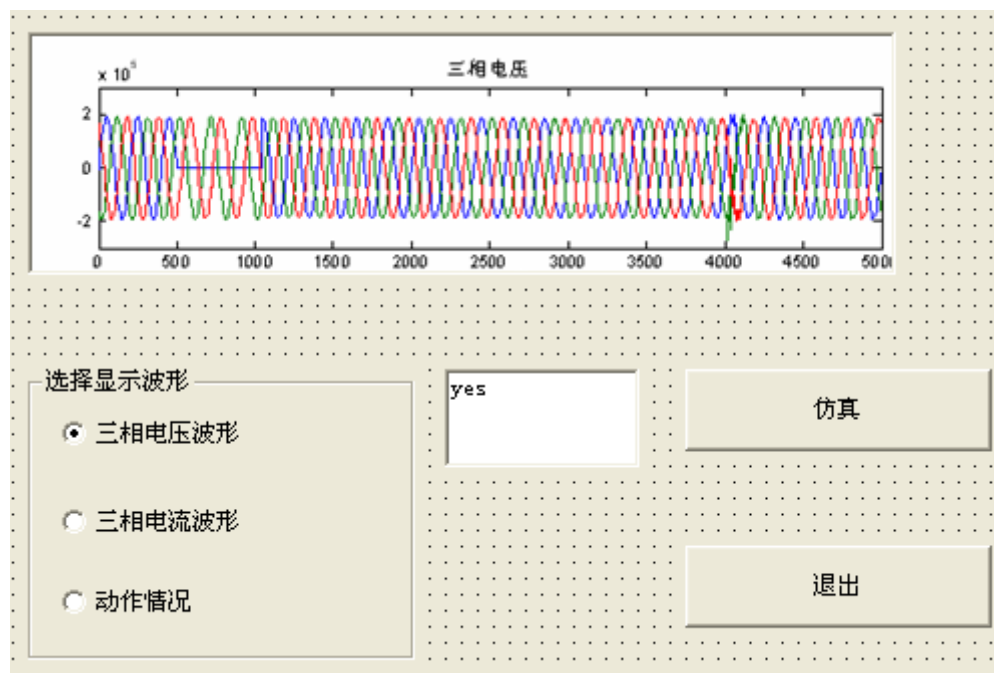


图 4-6 三相电压波形

②如果在 VB 界面中选择第二个情况三相电流，同样在 Text1 中键入 yes，然后单击仿真按钮，则可出现如图 4-7 所示。

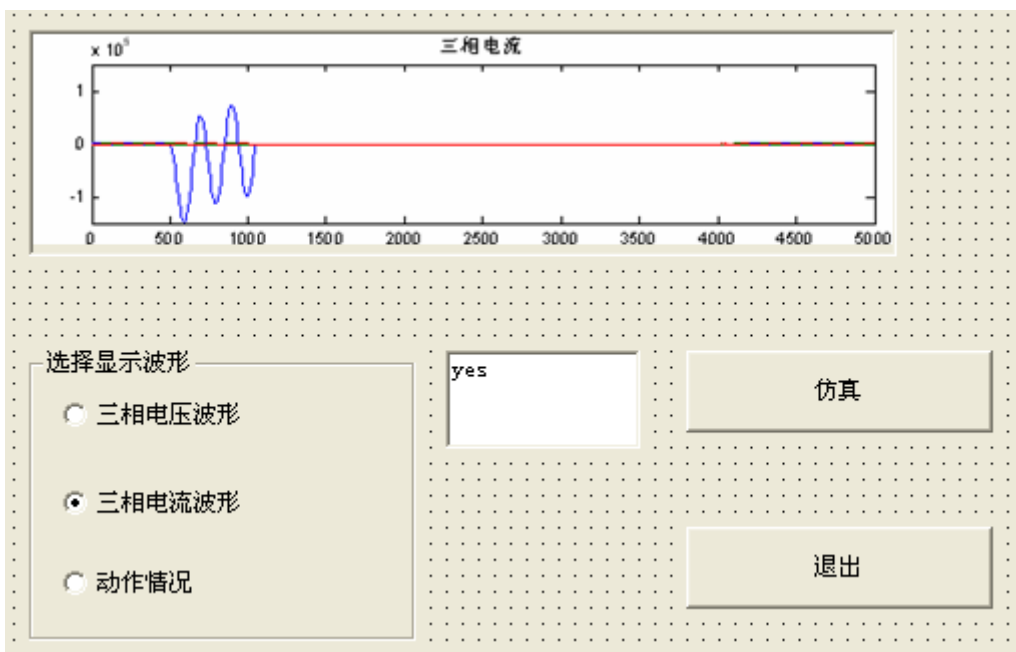


图 4-7 三相电流波形

③如果选择显示动作情况，则出现以下波形如图 4-8 所示。

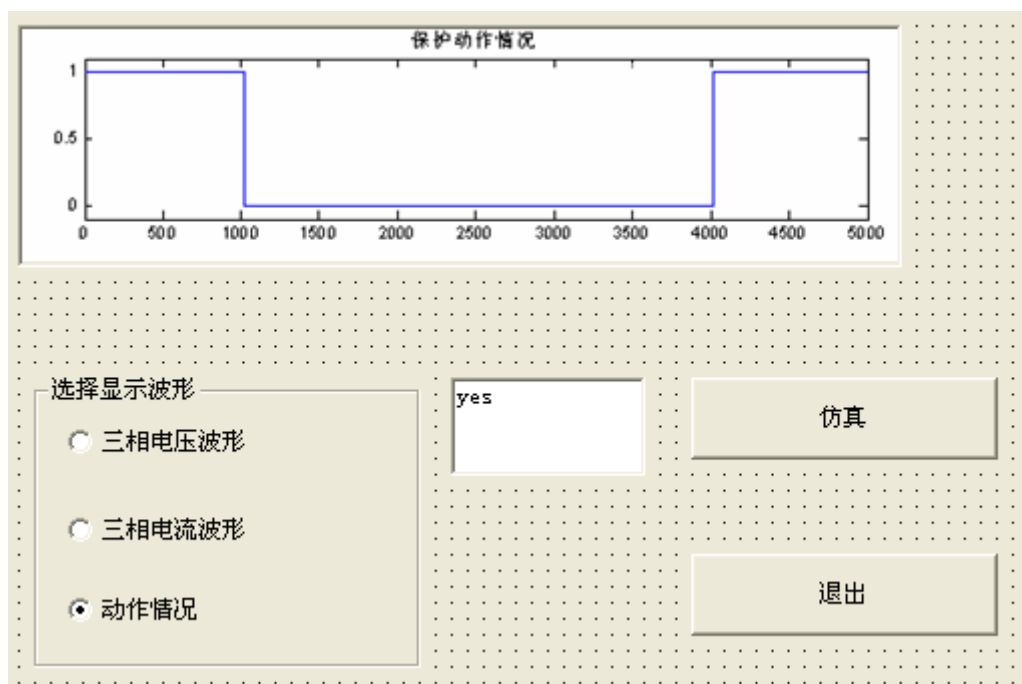
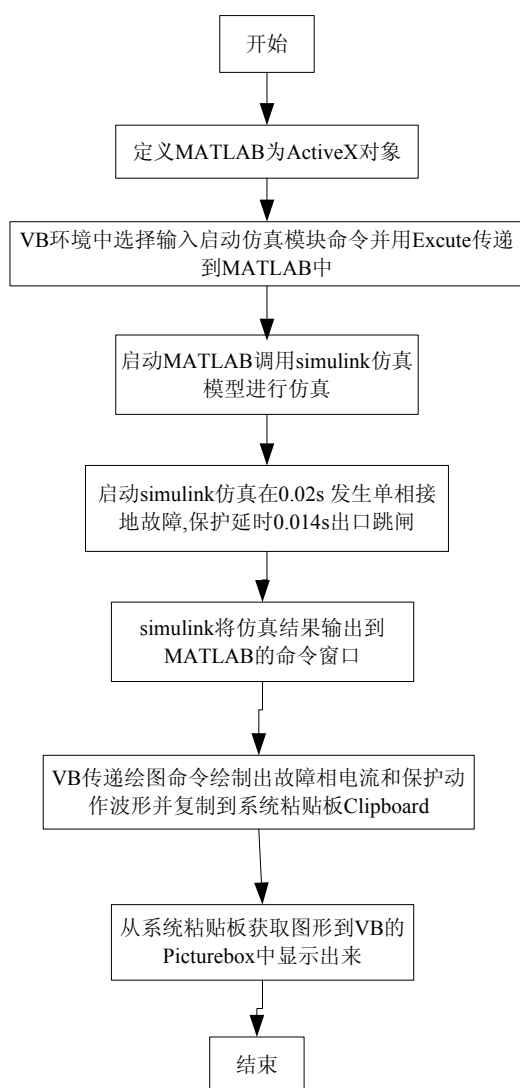


图 4-8 动作情况显示

动作过程：在故障出现后的仿真时间为 0.034s（即故障发生后 0.014s）时保护动作，差动保护出口，模型中的 A 从 1 跳变至 0。故障结束时保护返回，A 又从 0 跳变回 1，返回正常情况。

变压器差动保护仿真流程图如下：



4-9 变压器差动保护仿真程序流程图

4.6 软件使用

本次毕业设计所设计的软件当把这些 M 文件封装进 S-FUNCTION 中后，整个模型建成，只要改变故障发生器中的故障类型（双击故障发生器模块修改），就可模拟不同的故障情况，配合整定值可以判断故障是否发生在区内，发生在区内则动作，区外不动作。

仿真模型的运行只需进入 MATLAB 界面后，打开所建立的模型（保证

S-FUNCTION 所封装的 M 文件与模型在同一文件夹中), 在模型界面中的菜单栏中有一运行按钮, 单击此按钮即可, 可在菜单栏中选择仿真参数设置即可改变仿真时间。仿真结束后, 可双击示波器 (scope), 观察变压器两端电流电压波形。而差动保护是否动作则可观察显示 (display), 为 “1” 时则为动作, 为 “0” 是则为不动作。

我们还可根据实际情况改变仿真模型的采样时间。

根据以上所做的仿真, 证明可以使用 S-FUNCTION 来实现继电保护的仿真。我们还可以设计出双电源供电网络、三绕组变压器工作、自耦变压器工作等不同情况的仿真。只需根据这些工作方式的一些保护动作判据对 M 文件程序进行修改, 模块增添一部分, 配合整定值得出结果。

第五章 结果分析

5.1 微机变压器差动保护的整定计算

模型的参数为 220kV 的单电源供电网, 电源容量为 350MVA, 电压为 13.8kV, 变压器的参数为 13.8/220kV 的双绕组变压器。整个模型的基准电压为 220kV, 基准容量为 100e6VA。

1) 躲变压器的励磁涌流, 采用二次谐波制动, 制动系数整定在 0.15~0.17, 此处采用 0.15。

2) 躲外部故障, 采用两折线的比率制动原理。

由于变压器为两圈变, 额定容量为 $S=350\text{e6VA}$, 高压侧为 220kV, CT 变比为 $n_h/5$, 低压侧为 13.8kV, CT 变比为 $n_l/5$ 。

变压器一次侧的额定电流为:

$$\text{高压侧: } I_h = \frac{S_e}{\sqrt{3} * U_{eh}} = \frac{350\text{e6}}{\sqrt{3} * 220\text{e3}} = 0.9185\text{e3}$$

$$\text{低压侧: } I_l = \frac{S_e}{\sqrt{3} * U_{el}} = \frac{350\text{e6}}{\sqrt{3} * 13.8\text{e3}} = 14.643\text{e3}$$

$$\text{而模型的基准电流为: } I_B = \frac{S_B}{\sqrt{3} * U_B} = \frac{100\text{e6}}{\sqrt{3} * 220\text{e3}} = 0.2624\text{e3}$$

所以，高低压侧电流的标幺值分别为：

$$I_h^* = \frac{I_h}{I_B} = \frac{0.9185e3}{0.2624e3} = 3.5004$$

$$I_l^* = \frac{I_l}{I_B} = \frac{14.643e3}{0.2624e3} = 55.7934$$

从而高低压侧 CT 的变比分别为：（ $I_e = 5$ 为 CT 二次侧额定电流）

$$n_h = \frac{3.5004}{5} = 0.69994$$

$$n_l = \frac{55.7934}{5} = 11.51868$$

差动起动电流： $I_{op\min} = (0.4 \sim 0.8)I_e$ ，此模型中取 $I_{op\min} = 0.4I_e = 0.4 * 5 = 2$

起始制动电流： $I_{res\min} = (0.8 \sim 1.0)I_e$ ，此模型中取 $I_{res\min} = 1.0I_e = 5$

比率制动系数：计算式为 $K_{res} = K_k(K_{tx} * F_{wc} + \Delta U + \Delta F_{ph})$

式中： K_k 为可靠系数，取 $1.3 \sim 1.5$ 。

K_{tx} 为互感器同型系数，取 1.0。

F_{wc} 为互感器误差，取 0.1。

ΔU 为调压引起的误差，取调压范围的一半。

ΔF_{ph} 为由于电流互感器变比未完全匹配产生的误差，取 0.05。

一般 K_{res} 取值为 $0.3 \sim 0.7$ ，此模型中取为 0.5。

差动电流速断整定：当内部故障电流很大时，防止由于电流互感器的饱和引起差动保护延迟动作。整定值按躲过变压器初始励磁涌流或外部短路最大不平衡整定，一般整定为 6~12 倍的变压器二次侧额定电流。此模型中取为 10 倍的二次侧额定电流。

5.2 模型仿真结果

仿真参数设置为如图 5-1：

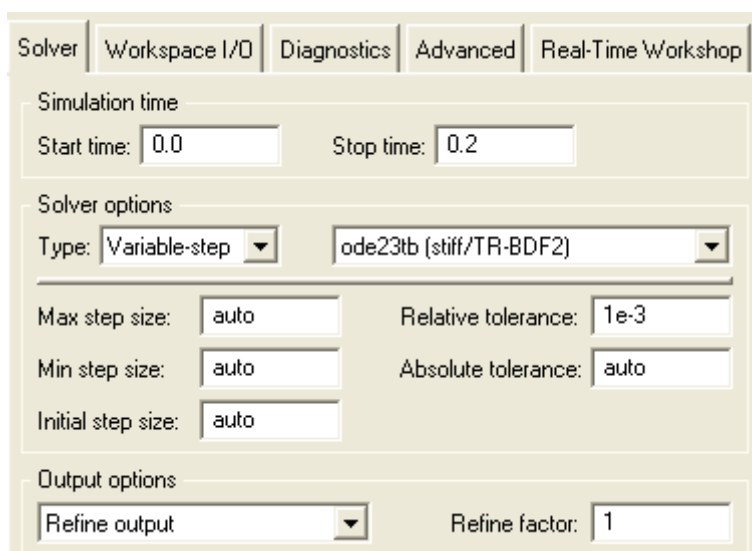
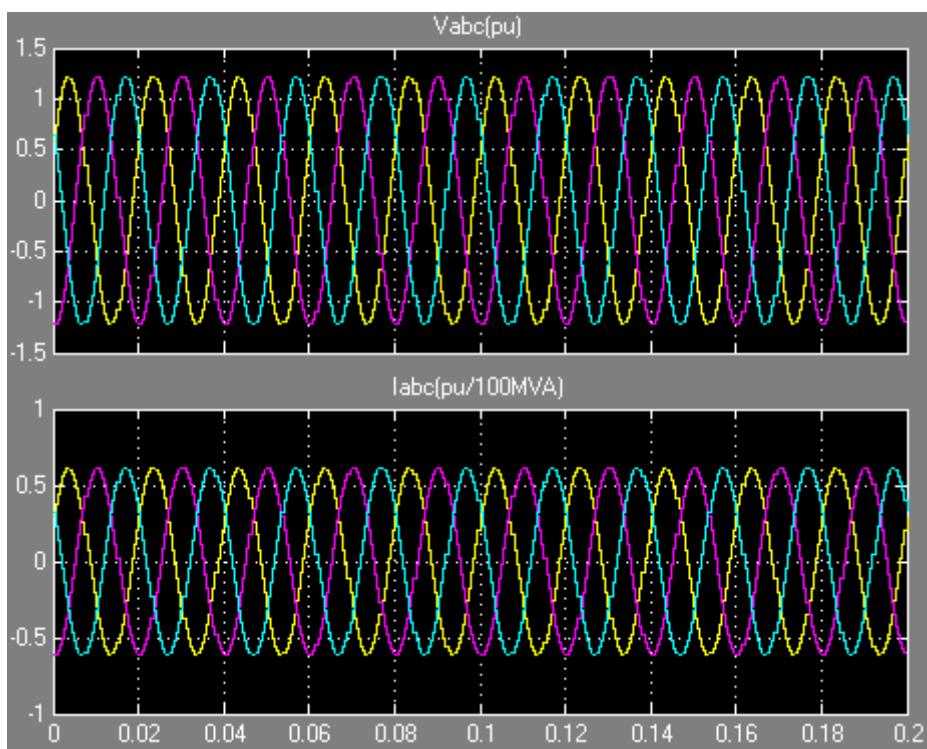
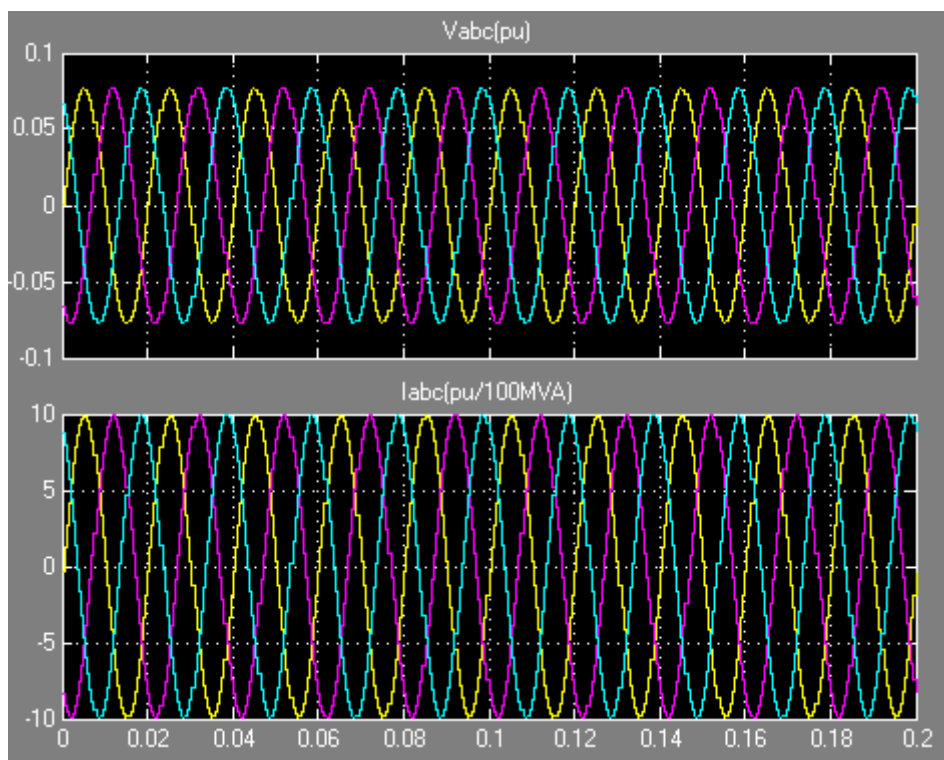


图 5-1 仿真参数设置

当系统正常运行（无故障）时，变压器两侧的电流波形如图 5-2：



(a) 高压侧



(b) 低压侧

图 5-2 正常运行高、低压侧电流波形

保护不动作，display 显示为 0。

当系统出现单相接地故障时，故障发生器设置参数为如图 5-3：

Three-Phase Fault (mask) (link)

Use this block to program a fault (short-circuit) between any phase and the ground. You can define the fault timing directly from the dialog box or apply an external logical signal. If you check the 'External control' box , the external control input will appear.

Parameters

☒ Phase A Fault
☐ Phase B Fault
☐ Phase C Fault

Fault resistances Ron (ohms) :

☒ Ground Fault
Ground resistance Rg (ohms) :

☐ External control of fault timing :
Transition status [1,0,1 ...):

Transition times (s):

Sample time of the internal timer Ts(s):

Snubbers resistance Rp (ohms) :

Snubbers Capacitance Cp (Farad)

Measurements

OK

Cancel

Help

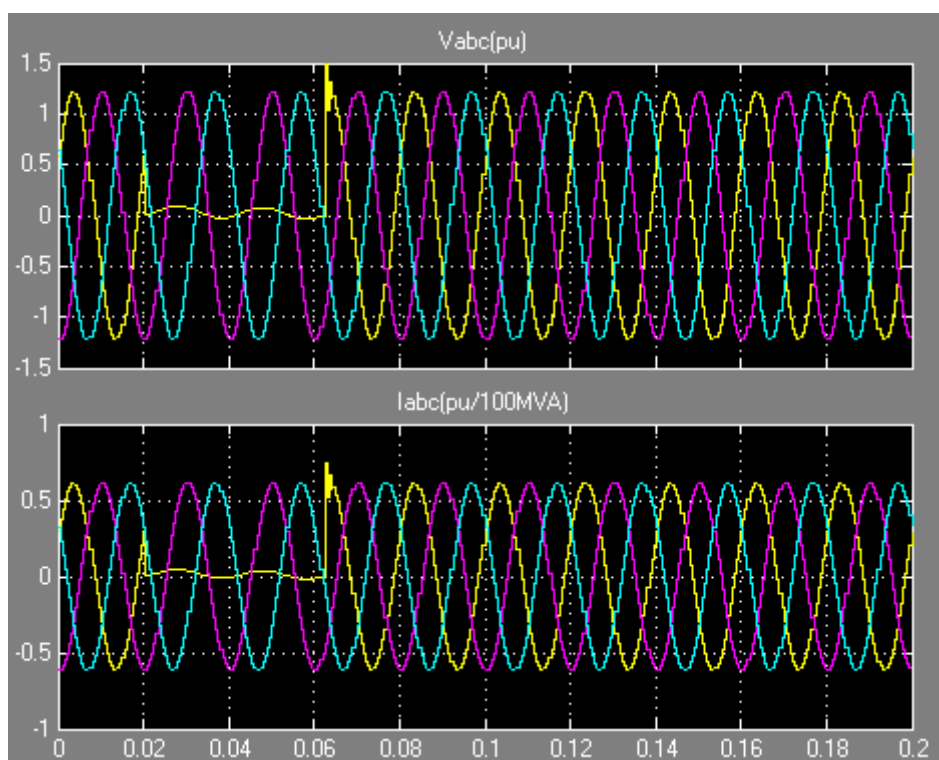
Apply

图 5-3 故障发生器设置

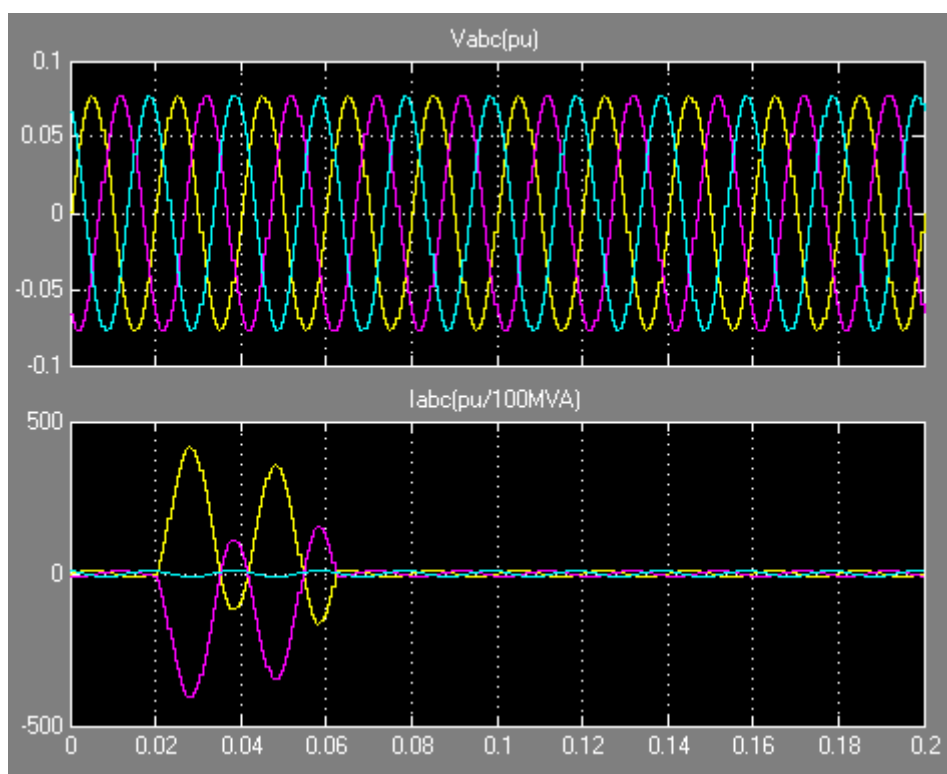
故障发生的时间段为：1/50~3/50s。

变压器两侧的电 流波形如图 5-4：

高压侧：



(a) 高压侧

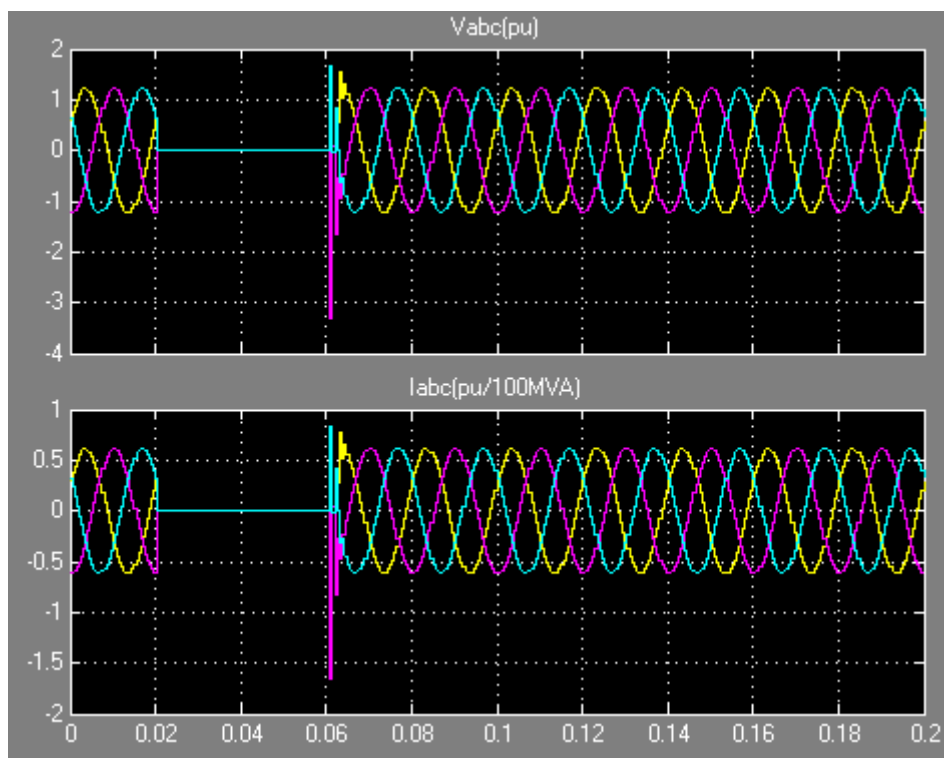


(b) 低压侧

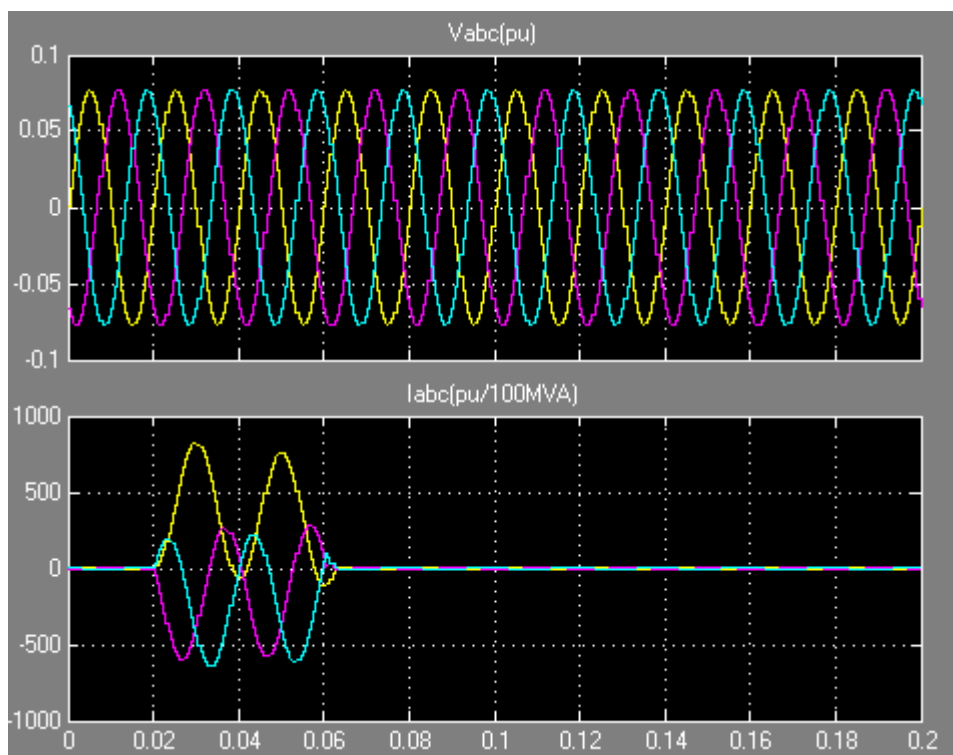
图 5-4 单相接地故障高、低压侧电流波形

在故障出现后的仿真时间为 0.034s（即故障发生后 0.014S）时保护动作，差动保护出口，display 显示为 1。故障结束时，0.065S（即故障结束后 0.005S 时保护返回，display 显示为 0。

当系统出现三相短路情况时，变压器两侧的电流量波形如图 5-5：



(a) 高压侧



(b) 低压侧

图 5-5 三相故障高、低压侧电流波形

在故障出现后的仿真时间为 0.037s 时（即故障发生后 0.017S）保护动作，差动保护出口，display 显示为 1。故障结束时，0.065S（故障结束后 0.005S）保护返回，display 显示为 0。

由此可得出此仿真模型动作正确，与变压器的整定参数配合，使得有选择性动作。也可改变仿真时间与采样时间，使得此仿真模型能够配合不同的仿真与采样时间。

工 作 总 结

本次毕业设计工作围绕“基于 MATLAB 的数字式变压器差动保护仿真”这一课题展开，在老师的悉心指导下，对所预想的问题都有了较满意的答案。毕业设计这一实践性环节培养了我针对实际问题确定研究方案，文献查阅、软件设计、系统仿真、缺陷处理等多方面能力，特别是培养了我独立学习和工作能力。对本次设计主要完成工作总结如下：

1) 阅读了大量相关的资料且温习了继电保护专业知识，学习了电力变压器数字式差动保护的基本原理，讨论了差动保护的各种判据，为编写比率制动、二次谐波制动差动保护计算程序奠定了基础，加深了对差动保护各种判据的原理和逻辑的了解。

2) 学习 Matlab 语言的编程基础，了解其强大的数值运算功能和大量的内置函数，以及矩阵运算的特性，掌握 M 文件的编写技巧。熟悉 Simulink 及 SimPowerSystems 工具箱。

3) 通过网络资源，了解 Matlab 接口技术的最新成果，利用 M 文件编写了电力系统差动保护计算程序，结合保护的仿真模型，进行运行调试，结果验证了程序的正确性、实用性。

4) 利用 Simulink 中的 SimPowerSystem 工具箱与 S-FUNCTION 模块，构建了一个 220kV 单侧电源供电系统的变压器差动保护的仿真模块。在系统中设定不同的故障点，分别进行了相应的仿真，得出了故障时的三相电压、三相电流波形，并对仿真结果进行分析。

5) 最后我们在毕业设计工作的基础上，完成了毕业设计说明书（论文）对毕业设计整个过程和毕业设计的成果进行了总结。

对论文评价：

本论文归纳总结了在整个毕业设计过程中，所学的各方面的知识和设计的成果。利用 MATLAB 对变压器的差动保护进行仿真，在此基础上利用 VB 的可视化的优点，实现 MATLAB 与 VB 的接口技术。利用 MATLAB 软件设计出不同采样周期的数字式变压器比率差动保护仿真 M 文件，且利用 Simulink 对数字式变压器保护建立仿真模型。且将利用 S-FUN 功能将 M 文件封装进 S-FUN 模块中，从而实现数字式变压器比率差动保护的仿真。本次仿真再结合对变压器微机保

护的整定方法，使之与整定值相配合，做到有选择性动作，同时进行各种故障模拟，利用 MATLAB 仿真程序自动处理采样数据，以做出正确的反应。利用 Matlab 和外部接口技术能够发挥 Matlab 和 VB 软件各自的优势，实现快速方便开发工程软件，这样既节约宝贵的开发时间和精力，从而满足教学与科研的需要。

论文完成的成果小结：

在本文中，通过学习 Matlab 软件参阅一些参考书和文献，在实际设计应用中，对于 Matlab 的 M 文件编写，对 Simulink 和 SimPowerSystems 工具箱中的大部分模块有了较详细的了解，并了解 Matlab 的外部接口技术。在对 Matlab 熟悉了解的基础上，利用 Matlab 外部接口技术和 M 文件编写技巧对电力系统变压器差动保护进行计算机实现；利用 Simulink/SimPowerSystems 工具箱对供电系统进行了仿真模型建立和仿真分析，并进行了变压器二次谐波制动差动保护的逻辑仿真。主要有以下结论：

1) Matlab 语言正是以处理矩阵见长，因此我们用语言 Matlab 编写了潮流计算核心。因此 Matlab 在电力系统潮流计算中有以下几个好处：①程序代码少；②程序调试方便，提供了丰富的错误信息方便寻找错误程序。

2) MATLAB 软件中的 SimPowerSystems 工具箱包含许多电力系统仿真模块以及很多其他功能模块，此次通过用这些模块对电力系统进行仿真，在所建模型中，可通过示波器观察各处的电流电压功率的波形均符合要求，用 Simulink 所建模型是正确的，验证了这些模块确实能用于电力系统的仿真。所设计的比率制动、二次谐波制动差动保护能正确判断是否存在故障并正确动作，起到了保护作用，通过对断路器进行外部控制，使保护动作后使断路器跳开。

3) 在差动保护程序设计中采用基于 Matlab COM Builder 的 Matlab 与 VB 的接口技术，即发挥了 Matlab 矩阵运算的长处，也发挥了 VB 界面设计方便的长处，在 VB 的界面上可方便的实现整定动作参数输入、算法选择、结果查看等功能，从而实现了差动保护的可视化，并且可生成脱离 Matlab 环境独立运行的安装包。为工程软件开发提供了一种新颖的、方便的途径。

4) 而在继电保护仿真中采用了基于 Active X 技术的 Matlab 与 VB 的接口技术，采用 Simlink 和 SimPowerSystems 工具箱可以建立多种多样的电力系统及其继电保护仿真模型，利用 Matlab 强大的仿真功能与 VB 在图形用户界

面开发方面的优势进行混合编程，能相互取长补短，实现快速开发工程软件，这样既节约宝贵的开发时间和精力，又能提高软件的整体性能，从而满足教学与科研的需要。

5) 本课题研究尚有不足之处，如界面的形式单调，需进一步发挥 VB 的多媒体控制功能。在继电保护仿真软件中，还应研究对故障类型的控制，使此次设计的软件功能更加完善。在教学与科研过程中发挥更大的作用。

致 谢 词

参考文献

- [1] 贺李,宋从矩. 电力系统继电保护原理.北京: 中国电力出版社
- [2] 许正亚. 变压器及中低压网络数字式保护.北京: 中国水利水电出版社
- [3] 刘万顺. 电力系统故障分析.北京: 中国电力出版社, 1998
- [4] 刘卫国. MATLAB 程序设计教程.北京: 中国水利水电出版社, 2005
- [5] 吴天明,谢小竹. MATLAB 电力系统设计与分析.北京: 国防工业出版社, 2004.1
- [6] VB 从入门到精通 北京希望电子出版社
- [7] 门爱东,杨波. 数字信号处理. 北京: 人民邮电出版社, 2003.4
- [8] 余啸海. Matlab 接口技术与应用[M].北京:国防工业出版社,2004
- [9] 章志刚,李兴源. 基于 MATLAB 开发平台的继电保护仿真系统[J]. 四川: 四川电力技术, 2003. 2
- [10] 罗士萍. 微机保护实现原理及装置. 北京: 中国电力出版

社，2001.8

- [11] 张志涌，徐彦琴. MATLAB 教程. 北京：北京航空航天大学出版社，2001.4