

DOI: 10.19783/j.cnki.pspc.200334

基于PSCAD的保护逻辑C源代码级仿真方法

刘伟

(许昌学院, 河南 许昌 461000)

摘要: 数字仿真技术在继电保护开发过程中具有不可替代的作用。在PSCAD环境中利用自定义模块接口技术,将C源代码以自定义元件形式嵌入到仿真模型中,实现数字化保护的源代码级闭环仿真。针对实施C源代码级仿真方法效率低下的问题,改进了“调用接口”和“数据接口”,通过一个实例详细阐述了实现过程和效果。通过在自定义元件的Script脚本中调用Fortran接口函数实现调用接口,简化了C源代码的调用、修改及调试。直接利用自定义元件的输入、输出端口实现数据接口,避免使用复杂的公用数组。该方法简洁、灵活,可以完整验证实际装置的保护逻辑源代码,尤其适合于从保护原理研究到装置开发的过渡阶段,使开发者不依赖于物理装置即可验证软件代码,提高了开发效率。

关键词: PSCAD; 保护逻辑; C语言; 自定义模块; 源代码; 仿真

A PSCAD-based simulation method for protection logic at C source code level

LIU Wei

(Xuchang University, Xuchang 461000, China)

Abstract: Simulation technology plays an irreplaceable role in the development of relay protection. C source code can be embedded in the simulation model by the interface technology of custom modules in PSCAD. This can realize closed-loop simulation at source code level for digital relays. To solve the problem of low efficiency in the implementation of C source code level simulation, the method in this paper has improved the calling interface and the data interface. The realization process and effects are explained by an instance. The calling interface is implemented by calling a Fortran function in the Script of custom components. This makes debugging the C source code much simpler. The data interface is realized directly by the input and output ports of custom components, and this will avoid the use of complex public arrays. The proposed method can be used to verify the source code embedded in an actual protective relay flexibly and simply. It is particularly suitable in the transition stage between studying principles and developing a device, because developers can validate the source code independently of a physical device. This will greatly improve the efficiency.

This work is supported by Key Research and Development Program of Hunan Province (No. 2016GK1003) “Research and Application of Key Technology of Vanadium Flow Battery Energy Storage System”.

Key words: PSCAD; protection logic; C language; custom module; source code; simulation

0 引言

对继电保护原理和装置进行测试、仿真,是保障和提高继电保护水平的重要手段^[1-2]。继电保护的仿真、测试与电力系统暂态过程的分析、模拟密切相关,因此,实际应用的仿真方法种类繁多^[3-4]。在线仿真是根据实际运行的电力系统中电网数据采集

系统提供的实时数据进行仿真计算;离线仿真是对电力系统的物理过程建立数学模型,再根据所搭建的仿真模型进行仿真计算,它与实际运行的电力系统没有直接联系^[5]。由于在线仿真需要直接获取电网实时数据,存在影响电网安全运行的隐患,实施起来难度较大。在进行离线仿真时,所测试的物理装置或所研究的算法可以将逻辑结果直接反映到仿真系统中,则构成闭环仿真;如果逻辑结果不能直接反映到仿真系统中,则为开环仿真。

离线闭环仿真能直观反映保护装置的性能测试

基金项目: 湖南省科技重大专项资助(2016GK1003)“全钒液流电池储能系统关键技术研发及应用”

结果, 至今仍是不可替代的研究手段。传统的离线闭环仿真方法主要包括动态物理模拟^[6-7]和实时数字仿真^[8-10]两种。动态物理模拟所需设备投资巨大, 研究系统规模有限, 参数调整较复杂, 其广泛应用受到一定限制。实时数字仿真采用 RTDS 或类似的数字系统实现, 受到的限制相对较少。但两类传统的离线闭环仿真均需要专门的硬件环境支撑。

为弥补离线仿真数据不是来自电网实际运行数据的缺陷, 离线仿真也可采取获取现场录波文件等非实时数据、利用 Matlab 等软件来研究保护元件的动作情况。该类方法属于保护算法级的离线开环仿真方法。由于实际保护装置中的保护元件众多, 相互之间有着复杂联系, 因此这类方法无法准确、完整地反映保护动作行为。

随着基于 IEC 61850 标准的全数字化保护的发展, 出现了一种基于离线仿真软件的准闭环仿真方法^[11]。故障数据分别可以利用 ATP 软件^[12]或 PSCAD^[13]软件产生, 再由保护实验仪将 SV 数据包发给物理装置; 再根据保护装置发出的跳闸命令手动修改 ATP、PSCAD 模型, 可以实现包含物理装置的准闭环仿真, 但是需要在不同格式的暂态数据文件之间进行转换^[14], 并且无法根据保护装置的调整命令自动修改仿真模型, 仿真效率较低^[15]。

为了克服上述准闭环仿真方法的不足, 特别是为适应保护原理研究阶段、缺乏物理装置情况下的仿真需求, 纯软件环境下的离线闭环仿真方法获得了广泛的研究^[16-18]。该类方法采用 ATP、PSCAD 等电磁暂态仿真软件产生故障数据, 在上述软件环境下对保护原理进行验证, 可实现离线闭环仿真^[19-20]。由于上述仿真软件可以通过自定义模块嵌入源代码, 因此这类方法除了可以实现保护算法级离线闭环仿真外, 还可进行源代码级离线闭环仿真。

在进行源代码级离线闭环仿真时, ATP 软件的开放接口灵活性稍差, 只能用 Fortran 语言编写自定义元件代码或建立与 Matlab 接口实现保护仿真^[21]。相比之下, PSCAD 的开放接口更灵活, 可以与 Matlab 直接接口^[22], 也可以用 Fortran 语言^[23]或 C 语言^[24-25]编写自定义元件实现保护仿真。

PSCAD 与 Matlab 结合的方法虽简单, 但是与保护装置中采用 C 语言编写的保护逻辑有较大差异。PSCAD 与 C 语言接口的方法, 目前多通过 PSCAD 提供的接口数组^[26-27]实现, 较为繁琐。而且, 现有离线闭环仿真方法尚无法实现包括数据采集环节在内的保护装置 C 源代码级仿真验证^[28]。为此, 本文研究了一种利用 PSCAD 的 C 语言接口技术实现的源代码级闭环仿真方法。该方法简洁、灵活,

可全面验证保护装置的 C 源代码, 提高了保护算法、逻辑的开发效率。

1 利用 PSCAD 自定义模块的仿真方法

利用 PSCAD 提供的自定义模块可以实现保护逻辑的闭环仿真, 整体架构如图 1 所示。图中三绕组变压器仅用来表征典型一次系统, 并无特殊含义。在 PSCAD 的图形化界面中绘制一次系统图, 通过编译器和内核计算程序 EMTDC 获得故障暂态过程的电压、电流瞬时值, 为保护逻辑的仿真提供原始采样值。另一方面, 将待仿真的保护逻辑代码封装到 PSCAD 的自定义模块中。然后, 通过 EMTDC 内核与 PSCAD 的自定义模块之间的接口实现数据交换。需要交换的数据除了前述的电压、电流瞬时值外, 还有断路器位置状态以及保护逻辑的跳闸命令等。

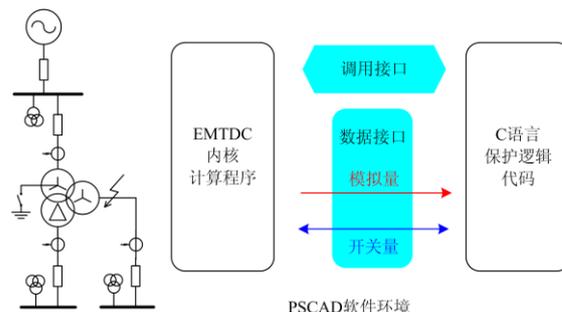


图 1 PSCAD 与保护逻辑自定义模块的接口

Fig. 1 Interface between PSCAD and custom module of protection logic

通过与物理动态模拟方法相对比, 不难发现图 1 方法完全在 PSCAD 软件环境中完成, 通过在 PSCAD 中直接嵌入完整的保护逻辑 C 语言代码, 实现了脱离保护装置硬件的保护逻辑源代码级闭环仿真。

图 1 方法的核心关键环节是 EMTDC 内核与 PSCAD 的自定义模块之间的接口^[20]。下面详细讨论实现该接口的方案, 对该接口的改进也是本文的核心工作。

1.1 PSCAD 自定义模块编程语言的选择

在 PSCAD 的 Protection 元件库中提供了过流、距离和方向元件等通用继电保护模块, 可以满足简单的教学、研究需求。但对于复杂保护原理的研究和工程实用算法验证还远远不够。

为此, PSCAD 提供了用户自定义模块接口, 用于实现自定义的保护算法, 并支持 Matlab、C 和 Fortran 等 3 种编程语言。由于目前成熟的数字化保护软件几乎全部用 C 语言编写, 为达到源代码级的

仿真，势必要选择 C 语言作为自定义模块的编程语言。

1.2 PSCAD 推荐的 C 语言模块接口方法

已有较多文献讨论采用 PSCAD 推荐方法实现与 C 语言自定义模块的接口，简述如下。

(1) C 语言模块嵌入到 PSCAD 内核计算程序的方法，即图 1 中所指的“调用接口”。

由于 PSCAD 的计算内核程序 EMTDC 是用 Fortran 语言编写的，因此无法将 C 代码直接嵌入其中。PSCAD 推荐方法是在编写好 C 源代码后利用 GCC 等编译器，编译链接成目标文件(*.o)或库文件(*.lib)，再通过在 PSCAD 增加外部引用，实现调用接口^[20]。

(2) C 语言模块与 PSCAD 的数据交换方法，即图 1 中所指的“数据接口”。

PSCAD 的推荐方法中，数据接口的实现依赖于内部封装的 4 个公用数组：STORF、STORI、STORL、STORC，及对应的访问指针：NSTORF、NSTORI、NSTORL、NSTORC。

数据交换的具体实现方法是：在每一步仿真计算完成后，自定义模块从 STORx 数组中读取所需要的数据；然后利用这些数据(相当于采样值和开关量)进行计算和逻辑判断；最后再将逻辑判断结果返回到 STORx 数组中。PSCAD 根据 STORx 数组的返回数据相应地修改模型，之后执行下一步的暂态仿真计算，如此反复，直至仿真结束。文献[18,20]给出该方法实现数据接口的实例，同时也可参考 PSCAD 中的示例工程。

1.3 推荐方法中存在的问题

上述 PSCAD 的推荐方法，在实际使用中存在诸多问题，包括：

(1) 调用接口以目标文件方式实现，在调试阶段频繁修改 C 源文件时，需要手动反复编译和链接目标文件，而且出现运行错误时调试不便。

(2) 自定义模块与 PSCAD 之间采用公用数组实现的数据接口，与实际保护装置的编程方式差别较大，因而无法直接验证保护装置输入、输出部分的源代码。

(3) 自定义模块的输出结果，包括跳闸信号等，也需要写入公用数组，提供给 PSCAD 修改仿真模型参数，用户无法直接显示公用数组中的信息，不便于调试。

(4) C 语言模块与 PSCAD 之间，以及 C 语言模块内部均存在数据交互，分别称为横向、纵向数据传递。按照 PSCAD 推荐方法，横向、纵向数据传递全部依赖公用数组及其指针实现，容易造成遗漏

或错位。由于 C 语言数组下标从 0 起始，而 Fortran 语言则是从 1 起始，在进行数组数据交换时，须时刻注意下标转换，极其繁琐。

2 保护逻辑的 C 源代码级仿真方法

由于 PSCAD 推荐方法存在上述不足，本文对调用接口和数据接口分别加以改进，使其更加简洁、灵活，并与保护装置实现方式保持一致。

2.1 C 语言模块的调用接口

除推荐方法外，还可以按照以下步骤实现调用接口：首先按照一定规范编写一个 Fortran 接口函数，在自定义模块的 Script 脚本中调用该函数；然后在上述 Fortran 接口函数中定义需要调用的 C 语言入口函数，完成与 PSCAD 的无缝链接。

以过流保护 C 源代码为例，描述本文方法实现的调用接口。首先，在 PSCAD 仿真工程中，右键选择“Create New Component...”命令，创建 1 个自定义元件。根据创建向导，设置自定义元件的名称“Over Current”。设置 1 个输入端“in”和 1 个输出端“out”，分别用于电流采样值输入和跳闸信号输出。右键点击自定义元件，选择“Edit Definition”命令，打开“Script”标签，输入该自定义元件的执行代码，即调用接口函数：

```
CALL AUX_CSUB($IN,$OUT)
```

AUX_CSUB()即为自定义元件的调用接口函数，函数的参数个数应与自定义元件的输入端口数量与输出端口数量的总和一致。

然后，右键选择“Add Component”命令，创建 2 个“File reference”元件，并在“File reference”元件的属性中设置外部引用的文件名和绝对路径，其中 1 个引用 Fortran 源文件 OC_df90.f，1 个引用 C 源文件 OC_c.c。

在 OC_df90.f 文件中除包含该接口函数的实现代码外，还要对自定义模块输入、输出的数据类型进行定义。虽然可任选 Fortran 支持的各种数据类型，但考虑到 C 数据类型与 Fortran 有一定差异，而且应尽可能反映保护装置实际情况，所以，选取输入变量为整型(对应保护装置的模数转换结果)，输出变量为整型或实型。

在 OC_df90.f 文件中实现的 Fortran 接口函数包含 2 段核心代码：第 1 段为 INTERFACE 段，声明需要调用的 C 语言入口函数名及参数变量；第 2 段调用 C 语言入口函数：

```
CALL OC_CSUB (in, out)
```

该 C 语言入口函数在 OC_c.c 文件中实现。

OC_c.c 文件即为用户自定义的 C 源代码文件,除入口函数名及参数变量必须与 OC_df90.f 文件的 INTERFACE 段声明一致外,其余的子函数皆可按照保护装置中的实际代码编写,包括相互之间的调用关系、宏定义、常量定义等。

2.2 C 语言模块的数据接口

实际保护装置的输入、输出,数量有限且相对固定,因此实现图 1 中的“数据接口”时,完全可以不使用 PSCAD 提供的公用数组,而仅利用自定义模块的输入、输出端口,再辅之以其他方式,来完成所有的数据交换。自定义模块的输入、输出端口可以看作保护装置的对外接线端子。由于用户可以直接获取到自定义模块输入、输出端口的数据,因此与 1.2 节中 PSCAD 推荐的 C 语言模块接口方法相比,本文所提的方法非常便于用户在 PSCAD 中对代码进行调试。

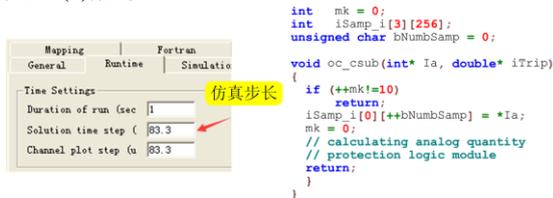
按照输入、输出的分类,数据接口可按照以下 3 类分别实现:

(1) 用于保护逻辑仿真的 C 语言自定义模块,输入量包括电流、电压等模拟量和断路器位置、手合信号等开入量。

开入量的实现较简单,可以采用整型变量直接读入,“1”代表闭合,“0”代表断开,直接参与保护逻辑判断。

为了与实际保护装置的模拟量接入方式保持一致,使得进入保护逻辑计算的数据采样率为 24 点/周波,需要针对不同应用场合分别采用如图 2 所示的 3 种方式实现模拟量输入接口。

a) 第 1 种方式适用于不进行频率调整的保护,例如变压器差动保护。这种方式实现相对简单,设置 PSCAD 模型的仿真计算步长为 83.33 μ s,然后在保护程序中增加一段每隔 10 个采样点的抽取代码,如图 2 (a)所示。



(a) 方式 1

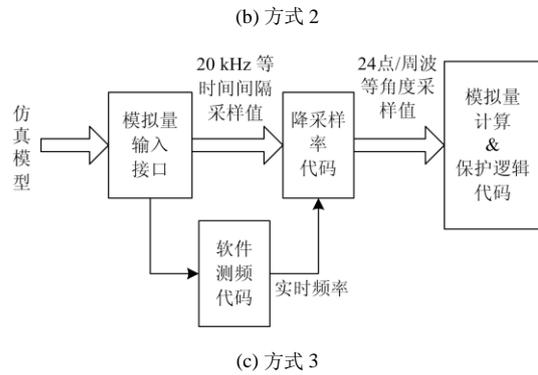
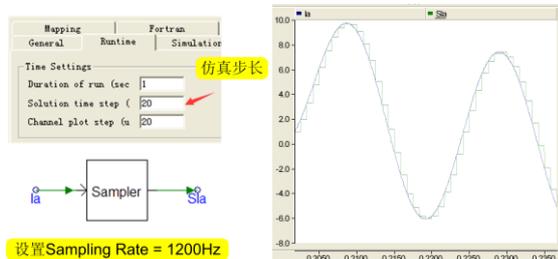


图 2 模拟量输入的数据接口

Fig. 2 Interface for analog data input

b) 第 2 种方式可任意设置 PSCAD 模型的仿真计算步长,例如 50 μ s,然后利用“Sampler”元件进行任意频率的降频“采样”,如图 2(b)所示。这种方式仍然属于不进行频率调整的方法,但是相对于第 1 种方式,仿真步长可灵活设置,有利于提高暂态过程的仿真精度。

c) 第 3 种方式设置 PSCAD 模型的仿真计算步长为 50 μ s,即 20 kHz 采样率,在自定义模块的源代码中调用变采样率预处理算法,得到 24 点/周波的采样值序列。图 2(c)给出了功能实现框图。在变采样率过程中需要同时实现频率跟踪,在改变采样率的同时,可将等时间间隔采样值序列转化为等角度采样值序列。这种方式的模拟量输入接口与智能变电站保护装置的重采样逻辑部分完全一致,可以直接验证这部分源代码。

(2) 用于保护逻辑仿真的 C 语言自定义模块,输出量包括保护跳闸信号和中间逻辑结果。前者通过一个延迟环节后,直接接入仿真模型的断路器控制端,实现保护跳闸的闭环仿真;后者用于调试保护元件的中间逻辑结果是否正确。

(3) 保护定值和投退控制字的输入方式有多种,本文通过在 C 源代码中直接赋值实现,可以避免使用 PSCAD 环境人机界面,同时最大限度地与实际保护装置中的保护定值数据结构保持一致。

2.3 保护逻辑的 C 源代码封装

与依据录波数据文件采用 Matlab 进行的保护算法级仿真不同,本文方法的最主要目的是实现保护逻辑的 C 源代码级仿真。因此在对自定义模块的 *.c 文件进行 C 源代码封装时,应尽量与保护装置中的代码完全保持一致。但应在以下方面加以注意,以适应 PSCAD 环境:

(1) 须将全部待仿真代码集中编写于同一个 *.c 文件中,并使用唯一入口函数,函数之间的调用关

系没有特殊要求，建议按照调用关系倒序在文件中出现，以省去函数声明。

(2) 由于 PSCAD 软件环境的原因，不能在 C 函数内部使用 static 型静态变量，必要时以全局变量代替，这与实际保护装置开发的代码有一定差异，但不会产生实质性影响。

(3) 尽量减少 C 语言库函数调用，部分库函数调用可能导致 PSCAD 软件崩溃，可用自主编写的子函数代替开方运算，使用查表法实现三角函数计算等。

(4) 实际保护装置的嵌入式软件通常会管理多个优先级不同的中断，在各中断服务程序之间分配保护功能。本文方法假定所有保护逻辑功能均在一个采样中断中完成，因此无法仿真各中断之间的配合关系，而各中断之间的配合关系对于保护软件的健壮性至关重要，所以需要通过其他手段进行验证。

2.4 C 源代码仿真过程中的调试

本文方法很大程度上依赖 PSCAD 软件环境，因此在调试过程中需要使用不同于一般嵌入式 C 语言调试的方法和技巧，可从以下几个方面入手提高仿真调试效率。

(1) PSCAD 提供的编译环境侧重于仿真模型，更加符合 Fortran 语言的编译习惯，编译错误的提示信息不直观。可以将待仿真代码用专门的嵌入式 C 语言集成开发环境先行编译、查错，之后再嵌入 PSCAD 环境。

(2) 保护逻辑 C 代码调试过程中，往往需要监视中间逻辑结果，为此可在开关量输出 Trip 跳闸信号之外，再增加若干调试输出端口，按照实型数据定义，可以输出模拟量中间计算结果或者保护元件中间判断结果，并以示波器绘图方式直观显示出来，方便调试。

(3) 为了保存仿真结果以便对比研究，在 PSCAD 仿真模型中可以增加 Recorder 元件，将模型的模拟量输出和保护逻辑自定义模块的跳闸输出等一并记录为 Comtrade 文件。Recorder 元件以 Comtrade 兼容方式记录数据时最多支持 12 路模拟量和 12 路开关量，需要记录的通道数较多时，只能采用多个 Recorder 元件记录，生成多个 Comtrade 文件后，再用处理软件将其合并。

3 仿真示例

本节给出完整的仿真示例。根据本文提出的方法，在 PSCAD 中对保护逻辑实现离线闭环仿真可分为 4 步进行：

(1) 创建输电线路模型及自定义元件。

(2) 编写 Fortran 与 C 源码接口函数代码文件，并在自定义模块的 Script 脚本中调用接口函数。

(3) 通过“File reference”元件引用接口函数代码文件和保护逻辑 C 源码文件。

(4) 对保护逻辑 C 源码进行离线闭环仿真。

按照以上 4 个步骤，在 PSCAD 中创建的线路模型、自定义元件以及仿真结果如下所述。

在 PSCAD 中建立 220 kV 同杆双回线模型如图 3 所示，按照图 4 建立距离保护的 C 语言自定义模块。

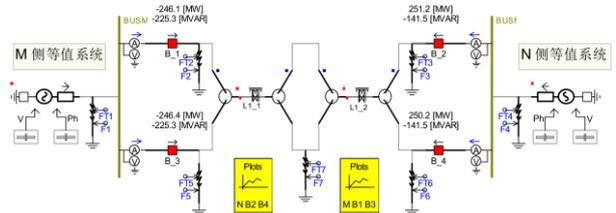


图 3 同杆双回线仿真模型

Fig. 3 Simulation model of parallel transmission lines

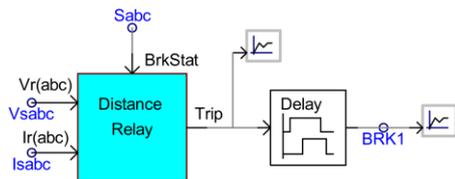


图 4 距离保护模块

Fig. 4 Distance relay module

图 4 的距离保护模块安装于开关 B_1 处，其中，Vr 和 Ir 输入端子分别接入电压和电流；BrkStat 作为断路器位置输入信号；Trip 即为保护跳闸信号；Delay 环节用于反映断路器固有跳闸时间，固定设为 0.05 s。设置仿真步长 50 μs，故障点位于 FT2 处，故障类型为 BC 相间接地故障。

图 5 给出了仿真结果，包括仿真模型输出的一次模拟量波形和保护逻辑自定义模块的跳闸信号，同时给出了保护控制的开关位置状态。其中模拟量计算和保护逻辑判断均由自定义模块内部采用实际保护装置的 C 源代码来完成，采样率为 24 点/周波。

距离保护模块经过模拟量计算、测量阻抗计算和距离 I 段、II 段逻辑判断后，在 0.228 s 时，即故障发生后 28 ms，发出 Trip 跳闸信号。这里的 28 ms 延时包括 3 部分：1) 保护算法中等待数据满窗的时间，大约 22 ms；2) 保护逻辑判断耗时，在 PSCAD 中该耗时与实际保护装置中嵌入式 CPU 的耗时可能不同，为准确反映实际保护装置的耗时，固定设置 4 个采样点的延时，约 2.5 ms；3) 跳闸延时确认时间，源码中设置为 4 个采样点，约 3.3 ms。

断路器 B_1 在收到保护跳闸信号后，经开关固

有跳闸延时 0.05 s 后,在 0.290 s 附近切断一次电流,随后距离保护收回跳闸信号,至此完成保护逻辑的闭环仿真。

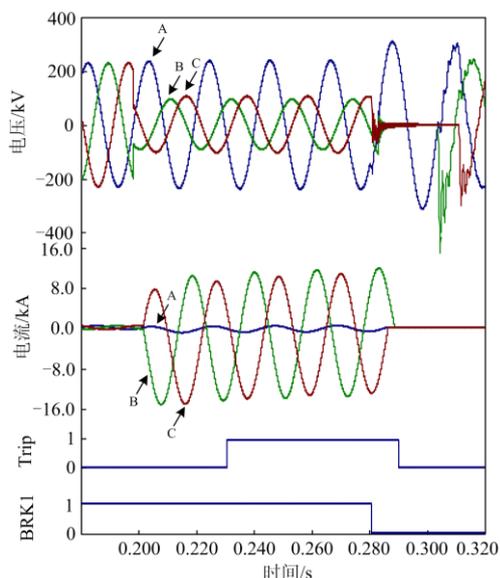


图 5 距离保护的测量及动作情况

Fig. 5 Measurement and operation of distance protection

图 6(a)给出了在故障发生和切除过程中,利用距离保护模块计算得到的测量电抗、电阻的变化情况。故障前,系统正常运行,测量电阻远大于测量电抗,测量阻抗呈现负荷阻抗特性;0.20 s 发生故障后,测量阻抗减小且电抗大于电阻,呈现线路阻抗特性;0.29 s 故障切除后,测量阻抗值再次变大。图 6(b)给出了在 0.20~0.24 s 之间,保护的测量阻抗在阻抗平面上的变化情况,同时绘制出来距离保护 I 段、II 段动作特性圆,以便比较。

按照文献[20]中方法对保护逻辑实现离线闭环仿真则需分为 5 步进行:

(1) 创建输电线路模型及自定义元件。

(2) 编写保护逻辑 C 源代码与公用数组进行数据交换的 Fortran 代码并与系统内核 Fortran 代码段进行整合。

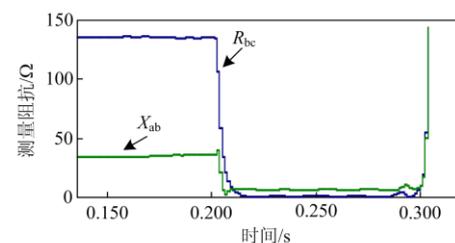
(3) 将保护逻辑 C 源码利用 EMTDC 提供的工具 GCC 编译成目标文件(*.o)。

(4) 引用编译好的目标文件(*.o)。

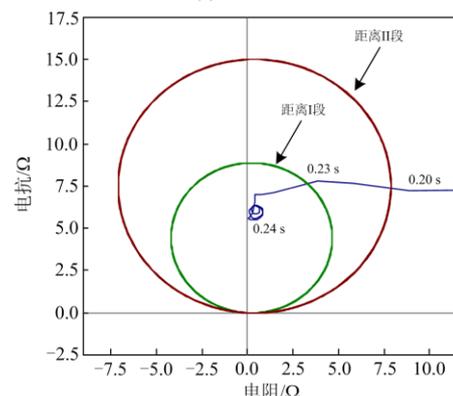
(5) 对保护逻辑 C 源码进行离线闭环仿真。

PSCAD 推荐方法中第(2)步编写的与公用数组进行数据交换的 Fortran 代码过于冗长、繁杂,同时还需避开 C 和 Fortran 因语法不同而导致的代码编写错误问题。本文提出方法步骤(2)中编写的接口函数代码文件仅有几行代码,简洁且模板化,易于修改。同时,本文中提出的方法不需要额外对保护逻辑

C 源码进行单独编译,通过“File reference”元件引用后,PSCAD 会在仿真过程中将保护逻辑 C 源码与 EMTDC 内核一起进行编译。



(a) 测量阻抗计算值



(b) 测量阻抗在阻抗平面上的变化

图 6 测量阻抗值及变化情况

Fig. 6 Measurement impedance and its variation

表 1 列出了本文方法与其他方法在完成保护逻辑 C 源代码级仿真时的执行效率对比情况。表 1 中列出了完成一次完整的距离保护代码闭环测试所需要时间,其中仿真时间统一设置为 5 s。由表 1 的对比结果可以看出本文方法相比传统方法大幅提高了调试效率。表 1 仅列出了执行仿真环节的耗时时间,尚未考虑中间停止仿真修改代码的时间。若再考虑到文献[20]方法修改代码反复生成 .o 文件的时间,以及本文方法可以直接查看中间变量输出结果的优点,本文方法在调试效率上的提高幅度会高于表 1 中的数据。

表 1 不同仿真方法的效率比较

Table 1 Efficiency comparison of different simulation methods

仿真方法	源代码闭环测试耗时/s	效率提高
PSCAD 推荐方法	25	—
文献[20]方法	15	40 %
本文方法	8	68 %

4 结论

本文研究了一种在PSCAD软件环境中实现的保护逻辑C源代码级闭环仿真方法,具有以下特点:

(1) 最大限度地实现纯软件环境下的源代码级

仿真,可以直接验证保护逻辑代码的正确性,相比保护逻辑算法级仿真,具有更高实用价值。

(2)从调用接口和数据接口两个方面对PSCAD的推荐接口方法进行了改进。避免使用复杂的公用数组,而是直接利用自定义模块的输入输出端口实现数据接口。

该方法尤其适用于从保护原理研究到装置开发的过渡阶段,使开发者不依赖于物理装置即可验证保护软件源代码,可大幅提高开发效率。

参考文献

- [1] 尹项根,曹文斌,潘远林,等.高阻抗变压器零模涌流特性与动模试验研究[J].电力系统保护与控制,2020,48(5):1-9.
YIN Xianggen, CAO Wenbin, PAN Yuanlin, et al. Study on zero-mode inrush current characteristics and dynamic physical simulation tests of high-impedance transformer[J]. Power System Protection and Control, 2020, 48(5): 1-9.
- [2] BASSEY O, CHEN Bo, BUTLER-PURRY K L, et al. Dynamic simulation of distribution systems and microgrids for reconfiguration studies using PSCAD/EMTDC[C] // 2019 North American Power Symposium (NAPS), October 13-15, 2019, Wichita, KS, USA: 1-6.
- [3] 张乔榆.三相暂态仿真与正序暂态仿真对比分析[J].广东电力,2018,31(10):59-64.
ZHANG Qiaoyu. Comparison analysis of three-phase transient simulation and positive sequence transient simulation[J]. Guangdong Electric Power, 2018, 31(10): 59-64.
- [4] 肖子龙,王丹,宋旭东,等.链式STATCOM的高效电磁暂态仿真模型[J].广东电力,2018,31(12):57-63.
XIAO Zilong, WANG Dan, SONG Xudong, et al. Efficient electromagnetic transient simulation model for cascade STATCOM[J]. Guangdong Electric Power, 2018, 31(12): 57-63.
- [5] 袁荣湘.电力系统仿真技术与实验[M].北京:中国电力出版社,2011:3-4.
- [6] 戚宣威,尹项根,李甘,等.一种电流互感器仿真分析平台构建方法[J].电力系统保护与控制,2015,43(22):69-76.
QI Xuanwei, YIN Xianggen, LI Gan, et al. A construction method for the simulation platform for the analysis of the current transformer[J]. Power System Protection and Control, 2015, 43(22): 69-76.
- [7] 姜运,彭红海,曾祥君,等.配电网故障行波定位动模实验平台[J].电力科学与技术学报,2017,32(3):81-85.
JIANG Yun, PENG Honghai, ZENG Xiangjun, et al. Dynamic simulation experimental platform for traveling-wave-based fault location in distribution network[J]. Journal of Electric Power Science and Technology, 2017, 32(3): 81-85.
- [8] 彭放,高厚磊,刘益青,等.基于RTDS-GTNET的智能变电站保护装置实用化测试技术及方案[J].电力自动化设备,2016,36(5):160-165.
PENG Fang, GAO Houlei, LIU Yiqing, et al. Practical test technology and solution based on RTDS-GTNET for protective equipment in smart substation[J]. Electric Power Automation Equipment, 2016, 36(5): 160-165.
- [9] ZHANG Rui, SONG Yankan, YU Zhitong, et al. A non-iterative switching status combination judgment algorithm for half-bridge sub-circuit based voltage-source converters in EMTP-type simulation program[C] // 2019 IEEE Innovative Smart Grid Technologies - Asia (ISGT Asia), May 21-24, 2019, Chengdu, China.
- [10] 段建东,汪鑫鑫,卢俊,等.基于ATP的HVDC系统建模及谐波分析[J].智慧电力,2019,47(11):9-15,21.
DUAN Jiandong, WANG Xinxin, LU Jun, et al. ATP based HVDC system modeling and harmonic analysis[J]. Smart Power, 2019, 47(11): 9-15, 21.
- [11] 华南理工大学.一种基于PC机平台的过程总线继电保护教学实验方法:中国,201110171659.X[P].2011-06-23.
- [12] 贾旭东,李庚银,赵成勇,等.基于RTDS的交直流混合系统实际故障再现分析[J].电力系统自动化,2009,33(10):54-58.
JIA Xudong, LI Gengyin, ZHAO Chengyong, et al. Real-time simulation studies of actual faults in AC/DC power system based on RTDS[J]. Automation of Electric Power Systems, 2009, 33(10): 54-58.
- [13] 姚洋,刘建锋,杨宏兵.基于IEC61850标准的智能变电站动模测试研究[J].电器与能效管理技术,2016,58(3):51-57.
YAO Yang, LIU Jianfeng, YANG Hongbing. Smart substation dynamic simulation test based on IEC 61850[C] // Electrical & Energy Management Technology, 2016, 58(3): 51-57.
- [14] 戴志辉,王增平.ATP/EMTP的保护模块内嵌技术与全闭环仿真[J].电力系统自动化,2009,33(20):73-77.
DAI Zhihui, WANG Zengping. Solution of building protection modules in ATP/EMTP and closed-loop simulation[J]. Automation of Electric Power Systems, 2009, 33(20): 73-77.
- [15] XIU Liancheng, KANG Zhiliang. Study on fault simulation of HVDC system based on PSCAD/EMTDC[C] // 2017 International Conference on Advanced Mechatronic Systems (ICAMEchS), December 6-9, 2017, Xiamen, China: 441-445.

- [16] 李志勇, 王伟, 庄良文, 等. 基于 RTDS 小步长仿真的配电网行波选线装置动模试验研究[J]. 电力系统保护与控制, 2019, 47(23): 108-114.
LI Zhiyong, WANG Wei, ZHUANG Liangwen, et al. Dynamic test of traveling wave line selecting device based on RTDS small time-step simulation[J]. Power System Protection and Control, 2019, 47(23): 108-114.
- [17] SU Sheng, DUAN Xianzhong, ZENG Xiangjun. ATP-based automated fault simulation[J]. IEEE Transactions on Power Delivery, 2008, 23(3): 1687-1689.
- [18] 安然然, 何茂慧, 王奕, 等. 复杂同塔线路继电保护动作特性仿真分析软件的设计与开发[J]. 电力自动化设备, 2015, 35(8): 156-162.
AN Ranran, HE Maohui, WANG Yi, et al. Design and development of operational characteristics simulation & analysis software for relay protections of complex transmission lines on same tower[J]. Electric Power Automation Equipment, 2015, 35(8): 156-162.
- [19] JEAN M, GABRIEL B, XAVIER L, et al. A link between EMTP and MATLAB for user-defined modeling[J]. IEEE Transactions on Power Delivery, 1998, 13(2): 667-674.
- [20] 蔡云峰, 徐洋, 潘琪. 基于自定义模型的变压器纵差动保护 PSCAD 仿真[J]. 电力系统保护与控制, 2015, 43(3): 118-122.
CAI Yunfeng, XU Yang, PAN Qi. PSCAD simulation of transformer longitudinal differential protection based on custom model[J]. Power System Protection and Control, 2015, 43(3): 118-122.
- [21] 肖异, 尹项根, 张哲, 等. PSCAD/EMTDC 程序与继电保护仿真模型接口技术及应用[J]. 电力自动化设备, 2006, 26(11): 67-70.
XIAO Yi, YIN Xianggen, ZHANG Zhe, et al. Interface technique between PSCAD/EMTDC and relay protection simulation model and its application[J]. Electric Power Automation Equipment, 2006, 26(11): 67-70.
- [22] Manitoba HVDC Research Centre Inc. PSCAD user's guide[M]. Manitoba: Manitoba HVDC Research Centre Inc, 2010.
- [23] 王玉婷, 李鹏. 基于 PSCAD/EMTDC 自定义建模的 MMC 电磁暂态仿真技术[J]. 高压电器, 2017, 53(10): 170-177.
WANG Yuting, LI Peng. Technique of electromagnetic transient simulation based on the user-defined model of PSCAD/EMTDC for MMC[J]. High Voltage Apparatus, 2017, 53(10): 170-177.
- [24] 周戴昊, 张紫凡, 肖燕, 等. 纯双侧电源输电线路继电保护的建模仿真[J]. 电力学报, 2019, 34(6): 564-570.
ZHOU Daihao, ZHANG Zifan, XIAO Yan, et al. Simulation modeling of relay protection for double-sided power transmission lines[J]. Journal of Electric Power, 2019, 34(6): 564-570.
- [25] PANG Y, XIANG Z, WONG M C. Comparison between PSCAD and RTDS hardware in-the-loop simulations system in power quality[C] // 2019 IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC) December 1-4, 2019, Macao, China: 1-4.
- [26] LIN Xuehua, GUO Qi, GUO Haiping, et al. Simulation modeling and AC system fault control strategy for the multi-terminal hybrid HVDC system based on RTDS[C] // 2019 IEEE Sustainable Power and Energy Conference (iSPEC), November 21-23, 2019, Beijing, China.
- [27] KIM M S, CHO G J, KIM C H, et al. New protective relay modeling scheme and analysis for AC electric railway feeding systems with Scott transformer[J]. International Journal of Electrical Power & Energy Systems, 2020, 123: 1-8.
- [28] 柏传军, 叶周, 程璐璐, 等. 基于实际工程及虚拟装置的直流输电控制保护模型设计方法[J]. 电力系统自动化, 2018, 42(14): 186-191.
BAI Chuanjun, YE Zhou, CHEN Lulu, et al. Design method of control and protection for DC transmission based on actual project and virtual device[J]. Automation of Electric Power Systems, 2018, 42(14): 186-191.

收稿日期: 2020-03-30; 修回日期: 2020-07-01

作者简介:

刘伟(1974—), 男, 硕士研究生, 高级工程师, 河南省高层次拔尖人才, 长期从事电力系统技术研究。E-mail: dswdliuwei@163.com

(编辑 魏小丽)