

应用CD++对继电保护进行建模研究¹

米增强, 赵洪山, 张纪平, 刘靖辉

华北电力大学, 071003

zhaohsh2002@yahoo.com.cn

摘要: 本文提出一种利用 CD++工具包针对电力系统继电保护动作行为进行建模的新方法, 该方法能够对具有离散特点的继电保护进行直观、高效、安全的建模与仿真。用这种方法所建的模型有利于对保护有更深入的理解, 为开发完善的保护装置提供了一种方法, 且能够为人员培训提供更直观、简明的仿真环境。我们以数字式三段距离保护为例, 为其建立理想的模型, 并详细阐述建模步骤, 最后验证了模型的合理性和可行性。

关键词: 继电保护; 建模; DEVS; CD++

1. 引言

对继电保护建模与仿真的研究, 不但有利于理解在系统故障时保护继电器的动态行为, 而且能够减少厂家开发新产品的周期和费用^[1]。通过建模与仿真, 可以揭示电力系统大停电事故中由保护动作引起的连锁事故发生的机理和电力系统的动态演化过程^[2]。当前在电力系统中, 研究数字式继电保护的建模与仿真的工具, 如利用基于EMTP、ATP或MATLAB等, 它们大多都不够方便, 可读性差, 尤其对于复杂逻辑的保护继电器建模更为复杂。所以有必要利用新的方法进行研究的建模方法。

保护继电器表现为一种离散的动态行为, 即逻辑的动态关系, 而CD++正是一个针对离散事件系统建模与仿真的工具包^[3]。为此, 我们提出了应用基于DEVS (Discrete EVents Systems specifications) 的CD++工具包对继电保护进行建模与仿真的思想与步骤。

2. 继电保护 CD++建模的思路

在电力系统中, 继电保护装置是电网中的重要环节, 它对于电网的可靠及安全运行有着重要而实际的意义。因此对继电保护建立较为精确、实用的模型是十分必要的。对于理想的保护模型, 根据保护动作行为的特性, 我们一般可以将其分为三种运行模式: 正常模式 (Norm)、启动后模式 (StartPost)、动作后模式 (OperPost)。三种动态模式的转换机制简要描述如下: 在正常模式 (Norm) 下, 系统没有发生任何故障, 保护装置不会有任何动作; 如果经判断, 系统发生故障, 那么此时三段保护的動作行为都将进入启动后模式 (StartPost), 根据三段各自的動作机制及動作范围来决定保护装置是動作, 从而进入動作后模式 (OperPost), 还是返回到正常模式。

¹ 本课题得到高等学校博士学科点专项科研基金资助课题 (20030079002) 资助。

3.1 继电保护三种模式的转换关系

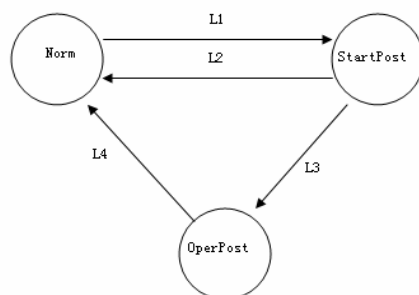


图 1: 继电保护三种运行模式的转化关系

Fig. 1 The relation of three modes of relay behavior

在图 1 中: 对于继电保护的三段来说, 四条弧 L1、L2、L3、L4 有着不同的意义。比如, 对于 I 段, L1 表示本条线路全长的某处发生故障, 此时 I 段必须要启动, 从而进入 StartPost 模式。如果经判断是 I 段所管辖的范围内故障, 那么保护会经 L3 进入 OperPost 模式, 表示 I 段动作, 然后经 L4 返回到 Norm; 如果经判断是线路末端故障, 即在 I 段管辖范围以外, I 段保护会经 L2 返回到 Norm。而对于保护的 II 段来说, L1 表示本条线路全长或相邻线路的一部分发生故障, 保护都会进入 StartPost, 如果此时经判断是本条线路末端故障, 那么保护经 L3 进入 OperPost 模式。因此, 对于保护的各段, 每条弧都代表着不同的判别机制。

3.2 CD++中保护继电器的 DEVS 模型

为了实现如上所述的这些关系, 我们建立了保护的 CD++模型。CD++是根据 DEVS 所开发的工具包, 用它建立的模型完全符合 DEVS 理论标准。为了使所建模型的动态行为更加灵活, CD++工具包允许用户使用 C++语言对其进行定义。但由于这种纯代码的方式较为复杂, 并不适合所有用户, 所以在 CD++中, 可以使用基于图形符号的方式进行更为直观的建模与仿真。本文便使用图形符号的方式为保护进行建模。

作为一个基于 DEVS 的工具包, 其模型必须按照 DEVS 理论的标准来建立。DEVS 体系是为了对复杂的离散动态系统进行直观及高效的建模与仿真而提出的一个理论框架。应用 DEVS 理论所建的模型不但具有层次化结构, 而且能够通过独立于模型本身的仿真机制运行。在该理论中, 模型被分为最基础的原子模型 (Atomic Model) 和更复杂的包括原子模型在内的组合模型^[4, 5]。

一个 DEVS 原子模型被描述为:

$$M = \langle X, S, Y, \delta_{int}, \delta_{ext}, \lambda, D \rangle$$

其中, X 是输入事件集合, S 是状态集合, Y 是输出事件集合。而其它的几个函数的意义: δ_{int} 用于处理内部变迁, δ_{ext} 处理外部变迁, λ 控制输出, D 为时间片。

一个 DEVS 组合模型被定义为:

$$CM = \langle I, X, Y, D, \{M_i\}, \{I_i\}, \{Z_{ij}\} \rangle$$

其中, X 是输入事件集合, Y 是输出事件的集合。 D 为模型组件的编号, $i \in D$, M_i 是基本的 DEVS 模型。其中

$$M_i = \langle I_i, X_i, S_i, Y_i, \delta_{int_i}, \delta_{ext_i}, tai \rangle$$

I_i 是第 i 模型的响应集。对于每个 $j \in I_i$, Z_{ij} 是从 i 到 j 的转换函数。

CD++ 用于实现 DEVS 的建模理论。在 CD++ 中, DEVS 的图形被定义为:

$$GGAD = \langle X_M, S, Y_M, \delta_{int}, \delta_{ext}, \lambda, D \rangle$$

$X_M = \{(p, v) | p \in IPorts, v \in X_p\}$ 输入端口集;

$Y_M = \{(p, v) | p \in OPorts, v \in Y_p\}$ 输出端口集;

$S = B \times P(V)$ 模型的状态;

$B = \{b | b \in Bubbles\}$ 模型状态的集合。

$V = \{(v, n) | v \in Variables, n \in R_0\}$ 模型的中间状态变量及其值。

$\delta_{int}, \delta_{ext}, \lambda$, 和 D 与传统 DEVS 模型的意义相同。

由此, 我们可以通过以下思路来创建模型: 首先, 可分别为保护的每段建立原子模型 (Atomic Model)。在原子模型中, 用一个状态 (Bubble) 代表一种模式, 因此, 首先建立三种运行状态, 并定义每种状态所持续的时间段; 按照逻辑关系, 用内/外部变迁函数 (虚线/实线) 参考图 1 中描述的关系, 将三种状态连接起来, 并在 CD++ 中定义这些函数所要进行的具体操作; 除了用来表示三段的原子模型外, 还需创建第四个原子模型 (Break), 该模型表示保护可以分别通过 I、II、III 段控制断路器进行跳闸; 最后, 将这四个原子模型通过一定的逻辑关系进行连接, 形成一个组合模型 (Coupled Model), 该模型就是一个较为完整的保护模型。

3. 距离保护的 CD++ 模型

按照以上思路, 我们以数字式三段距离保护为例建立理想的保护模型。在 CD++ 的 GGAD 建模环境中, 对于原子模型, 当输入端口接收到某一事件时, 如果满足一定的条件, 模型会通过内/外部变迁函数来决定进入另一种状态。一种状态有其默认的时间片 (保持该状态的事件), 如果满足外部变迁函数的条件, 那么模型会立即转入相应的状态而不管时间片是否已用尽。如果任何外部变迁函数的条件均不满足, 那么模型会等到时间片耗尽, 首先将状态转换的信息通过输出端口送出, 然后通过内部变迁函数转入其它状态。以下给出较为详细的建模过程。

3.1 三段距离保护的原子模型及 Break 模型

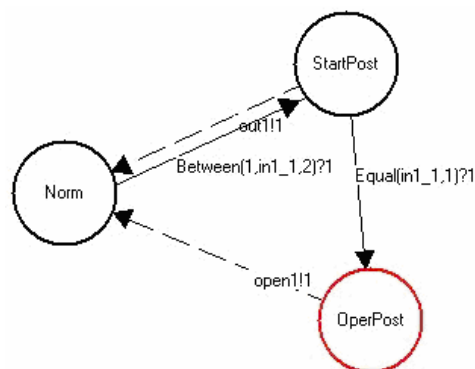


图 2. 保护 I 段的原子模型
Fig. 2 Atomic model of zone I

由图 2 可知,在保护 I 段的原子模型中,规定初始状态为 Norm.。I 段有一个输入端 in1_1,从 in1_1 端口接收到的事件,可用来判断本条线路的故障是否在 I 段范围内。in1_1 可能的数值有 0、1 和 2,0 表示本条线路无故障;1 表示本条线路 I 段的保护范围之内发生故障;2 表示本条线路末端发生故障,但位于 I 段范围以外。从图中可看出,如果输入端口 in1_1 接收到的数值为 1 或者 2(满足外部变迁函数 $Between(1,in1_1,2)?1$),那么 I 段都会启动,从而进入 StartPost。然后进一步判断如果 in1_1 是 1 (满足 $Equal(in1_1,1)?1$),则 I 段动作进入 OperPost。等到短时间的延时后,保护 I 段会发送 1 到输出端口 open1,接着返回到 Norm,表示 I 段发送跳闸信号给断路器。

以下分别给出 II 段和 III 段以及 Break 的 GGAD 模型:

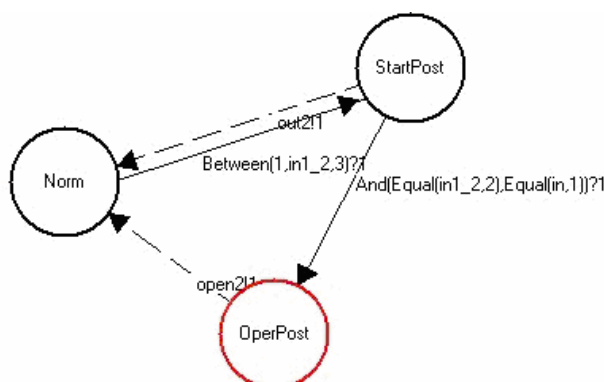


图 3. 保护 II 段的原子模型

Fig.3 Atomic model of zone II

与 I 段类似, II 段的原子模型中的各函数意义基本相同,该模型有两个输入端口 in1_2、in 以及两个输出端口 out2、open2 只是其中端口号的明成有区别。in1_2 仍然描述故障类型,它有四个值:0、1、2、3。其中 0~2 的意义和 I 段的输入端口 in1_1 相同;而当 in1_2 的值为 3 时,表示相邻线路某处发生故障,且该故障位于本线路 II 段保护范围之内。in 端口用来判别 I 段是否动作,它将与 I 段的 out1 端口连接(将在后面的组合模型中讨论)。如果 in (out1) 的值是 1 且 in1_2 为 2,就表明 I 段已启动但没有动作,故障属于本条线路末端故障且在 II 段范围内。此时 II 段动作,模型发送数值 1 给输出端口 open2,表示断路器通过 II 段跳闸,接着进入 OperPost 状态。

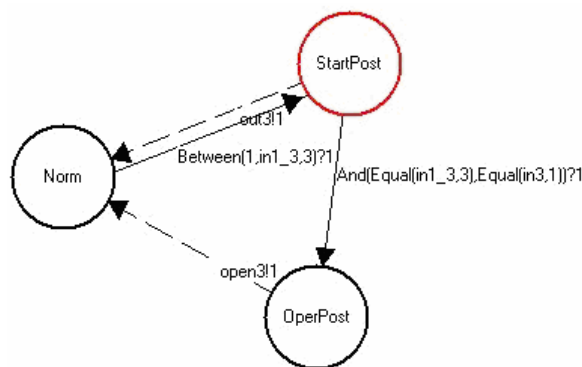


图 4. 保护 III 段的原子模型

Fig. 4 Atomic model of zone III

在图 3 所示的 III 段的原子模型中，同样有两个输入端 $in1_3$ 、 $in3$ 。分别用来区分故障类型以及判断 II 段的动作情况，从而决定其动作行为。 $in1_3$ 也有四个值：0~3。其中 0~2 的意义与 II 段相同，而当 $in1_3$ 的值为 3 时，则表示除“0”~“2”外的所有 III 段范围内的故障类型。输入端口 $in3$ 将与 II 段的输出端口 $out2$ 相连接。如果 $in3$ ($out2$) 的值为 1，表明故障位于相邻线路，保护会经过一段延时而动作，这段延时是通过 II 段 StartPost 状态的延时体现出来的（这将在组合模型中讨论）。如果 III 段进入 OperPost 状态，首先向端口 $open3$ 发送信号 1，表示断路器通过 III 段跳闸，然后会返回到 Norm，。

其次，建立 Break（跳闸模块）的原子模型：

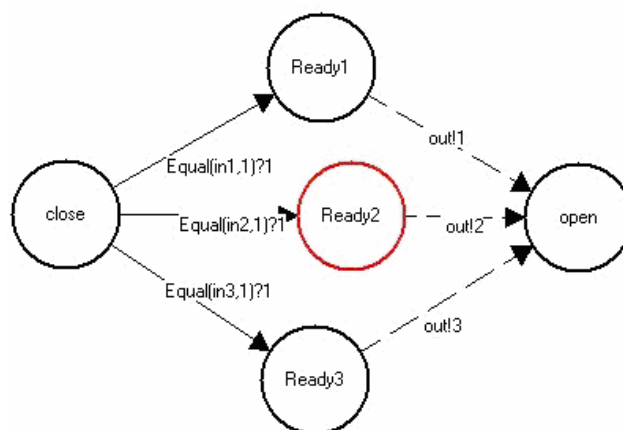


图 5. 跳闸模块的原子模型

Fig. 5 Atomic model of Break

为了便于说明，我们为 Break 设计了三个基本状态：close、Ready、open。分别用来表示开关闭合状态、控制开关的断路器准备跳闸、开关的断开状态。其中，Ready1~Ready3 用来表示断路器分别接收到 I、II、III 段的跳闸信号。三个输入端 $in1$ 、 $in2$ 、 $in3$ ，分别用来接收 I、II、III 段的输出信号。根据这三个端口来决定是否将开关断开（open），跳闸后通过输出端口 out 的值有三个，1 表示断路器接受到 I 段的跳闸信号而使开关断开；2、3 则分别表示接收到 I、II 段的跳闸信号。开关三种状态的设置完全是为了更好的说明它与 I、II、III 段保护的关系。

3.2 继电器的组合模型

最后，根据各段以及 Break 等模型的关系，将四个原子模型连接起来，形成保护的组合模型，见图 6。

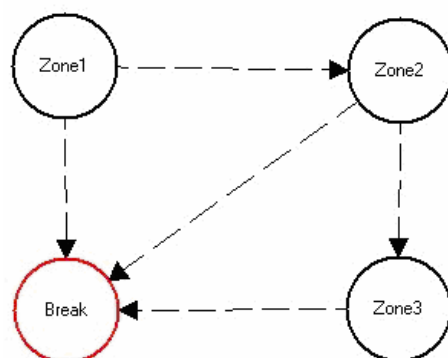


图 6. 继电保护的组合模型

Fig. 6 Coupled model of relay protection

对于组合模型，我们创建了三个输入端口：port1、port2、port3，分别用来连接 I、II、III 段的输入端 in1_1、in1_2、in1_3，表示将线路的故障信息分别送到各段保护中，从而根据这些信息进行相应的动作。而组合模型唯一的输出端口 out 将与 Break 的输出端口 out 相连接。为了让四个原子模型及组合模型相互建立连接（每一个连接用一个箭头表示），需要用以下规则来进行：源模型的输出端口同目的模型的输入端口连接；而组合模型的输入/输出端口与其中一个原子模型的输入/输出端口连接。比如连接 Zone1.out1-Zone2.in 表示模型 Zone1 的 out1 端口连接 Zone2 的 in 端口，箭头的方向指向 Zone2；而 MODEL.port1-Zone1.in1_1 表示组合模型的输入端 port1 对 Zone1 的 in1_1 端口进行输入。图 7 给出了组合模型的所有连接。

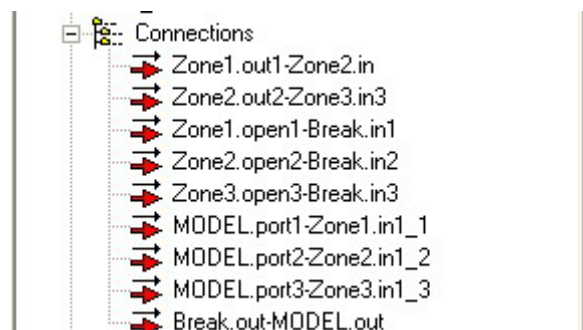


图 7. 继电保护组合模型的相关连接

Fig. 7 Connection of the coupled model of relay protection

这样，我们可以通过对组合模型进行输入，这时可将整个保护看成一个模块，这个模块有三个输入端口和一个输出端口。三个输入端口分别用来告知保护各段故障信息，输出端口可以得到两个信息，一是线路开关是否断开，二是开关经过保护的哪一段而断开的。

4. 继电保护模型的验证及分析

为了验证保护模型的合理性，我们用以下两类故障来分析该模型的动作情况。

(1). 本条线路的 I 段保护范围内发生故障：

在正常状况下，保护模块的输入和输出端口的值是零。而当(1)故障发生时，保护模块

将会从三个输入端口收到故障信息,此时三个端口的数值是: port1 (in1_1)、port2 (in1_2) 以及 port3 (in1_3) 均为 1。满足各段的外部变迁函数 `Between()` 的条件,于是 I、II、III 段都会启动,进入 `StartPost` 状态。由于 I 段满足 `Equal(in1_1,1)` 的条件,所以 I 段会动作,从而进入 `OperPost` 状态,经 5ms 延时后,通过内部函数 `open1!1` 返回到 `Norm`。此时由于 `open1` 连接着 `Break` 模块的 `in1` 端口,接收到进入该端口的数值 1 后, `Break` 进入 `Ready1` 状态,经过 30ms 的延时进入 `open`,同时发送跳闸信号 `out=1`,表明断路器是因为接收到 I 段的信号而跳闸。因此,对于保护的 I 段,从发现故障到切除故障共用了 35ms 的时间。对于 II 段,当它与 I 段同时进入 `StartPost` 时,要判断 `in(out1)` 端口的值是不是 1,此时由于 I 段已进入 `OperPost`,所以不会发送 1 到 `out1` 端口,因此,II 段不会动作,此时 II 段发送 1 给 `out2` 并返回。III 段启动后,由于不满足 `in1_3=3` 这一条件,所以也不会动作。

(2). 本条线路的 I 段保护范围外故障:

此时每一段会同时启动,进入 `StartPost`,但由于 I 段不满足动作条件而延时(500ms)返回。在这期间,II 段由于没有收到 I 段发送到 `out1` 端口的值而保持在 `StartPost` 状态,其延时大于 500ms,这里我们设置为 1s。等到 500ms 后,I 段返回到 `Norm` 同时发送信号 1 给 `out1`(II 段的端口 `in`),由于 II 段的延时还没有耗尽,此时接收到 I 段发来的信号,经过判断,输入信号 `in1_2` 和 `out1` 的值同时满足动作条件,所以 II 段动作。同 I 段动作类似,最后发送信号给 `Break` 的 `in2` 端口进行跳闸。对于 II 段,从发现故障到切除故障共用了 535ms。III 段没有收到 II 段的 `out2` 信号,所以不动作。

其它类型的故障可进行类似的分析。

最后,CD++同时提供了仿真平台,通过对继电器模型的仿真我们可以验证所建模型的合理性以及所关心的执行结果。

5. 结论

本文阐述了为电力系统继电保护建立理想模型的新方法的思路,并较为详细、直观地描述了建模过程。通过使用 CD++工具包,我们可以为继电器定义较为严格、精确地模型。但在建模过程中,仍有一些缺陷需要进一步研究,比如在电力系统实际运行过程中,系统存在振荡的可能,因此继电保护必须对此加以考虑。为了使保护模型更加精确完整,我们正在研究考虑振荡之后建立保护模型的思路与方法。

参考文献

- [1] 米增强, 赵洪山, 王海萍等. 数字保护继电器的抽象建模方法研究. 中国电机工程学报. 2005, 25(1):51-56.
MI Zeng-qiang, ZHAO Hong-shan, and WANG Hai-ping. Abstract Modeling Method of Digital Protection Relay. *Proceedings of the CSEE*, 2005, 25(1):51-56..
- [2] 赵洪山, 米增强, 牛东晓, 等. 利用混杂系统理论进行电力系统建模研究. 中国电机工程学报. 2003, 23(1): 20-25.
Zhao Hongshan, Mi zengqiang, Niu Dongxiao et al. Power system Modeling Using Hybrid system theory. *Proceedings of the CSEE*, 2003, 23(1): 20-25.
- [3] WAINER, G. CD++: a toolkit to define discrete-event model. G. Wainer. Software, Practice and Experience. Wiley, 2002, 32(3):1261-1306.

- [4] Hong, G. P., T. G. Kim. A framework for verifying discrete event models within a DEVS-based system development methodology. *Transactions of the Society for Computer Simulation*. 1996, 13(1):19-34.
- [5] Gastón Christen, Alejandro Dobniewski, Gabriel Wainer. Modeling State-Based DEVS Models in CD++. Military Government and Aerospace Simulation Symposium. ISBN: 1-56555-279-2, 105-110.

Modeling for Relay Protection with the CD++ Toolkit

MI Zeng-qiang, ZHAO Hong-shan, ZHANG ji-ping, LIU jing-hui

North China Electric Power University, 071003

Abstract

This paper presents a new method of modeling and simulation for relay protection by using a DEVS-based toolkit named CD++. With this toolkit, the modeling of relay will be more easy and straightforward. We takes the digital three zones distances protection as the example, studies the approach of ideal protection relay modeling, and also obtain related executive data of the relay internal states and the dynamic process of logic relation.

Keywords: *modeling, relay protection, DEVS, CD++*