

# 利用专家系统判断故障位置

天津高压供电公司 刘琳 东北电力学院 王静茹

**摘要** 根据 Turbo Prolog 故障诊断推理系统, 利用以电力系统运行方式为依据的数据库及以继电保护原理为规则的规则库, 当输入电力系统故障时有关动作的继电保护和跳闸的断路器信息时, 能自动判断出故障位置, 同时还能判断拒动的保护装置和断路器。它给电力系统调度人员进行事故分析提供可靠依据。

**关键词** 专家系统 诊断 系统 故障

## 1 概述

近年来人工智能 (A. I) 特别是其中最有效分支—专家系统 (E. S) 在电力系统运行中的应用研究十分活跃, 而且取得了引人注目的进展, 它标志着计算机在这方面已由数值计算、实时控制发展到了智能性辅助决策阶段。

目前计算机技术在电力系统的各种处理和各方面得到广泛的应用, 但在整个系统水平上实现紧急操作的自动化仍是一大难题, 仍然以人工操作为主。当发生故障时, 必须通过一系列继电保护装置和断路器的动作, 把故障段测定出来, 并从系统中切除。测定故障区段是恢复过程的第一步。在实际中, 测定故障位置是很麻烦的, 如果还存在保护和断路器的拒动, 故障要由后备保护切除, 就更困难, 这时切除的区域 (即停电区) 是非常大的, 如果两个故障同时发生, 判断将变得更加复杂、困难。

本文提供了一个帮助调度员分析这些情况的专家系统, 它根据来自继电保护装置和断路器的信息测定可能的故障位置。这个专家系统用于调度中心, 帮助调度员判断紧急情况, 使恢复过程的第一步顺利完成, 即使停电区尽可能地缩小, 且准确判断。这个专家系统选用 “Prolog” 语言作为程序开发语言。Prolog 作为一种知识工程的计算机语言, 使用概念的知识与信息描述语言来表示系统, 建立数据库、知识库, 通过逻辑推理对故障段作出推断。这些知识和资料信息几乎与有经验的调度员水平相同。系统所必需的知识与输入信息是概念性知识, 它是关于保护继电器的类型、已动作的保护装置的信号、跳闸断路器的信号, 同时装入了专家调度员推断故障段的经验知识。要解决的问题是: 测定故障的区域, 判断拒动的保护装置和断路器。

## 2 基于知识的推理系统

这个专家系统用 Prolog 语言写成, Prolog 是一种基于一阶谓词计算的计算机语言, 程序由事实和规则构成, 电力系统的知识作为事实贮存在数据库中, 而继电保护原理作为规则在有关动作的继电器和跳闸的断路器的信息输入后, 推理和根据其动作情况计算故障位置。用 Prolog 语言的模式匹配和回溯功能检查事实库和规则库, 找出满足规则的故障段, 由推理机进一步确定故障位置, 这就是此专家系统推理的基本步骤。

### 2.1 推理计算机所用的数据库

数据库 1: 网络和保护系统的事实。这个数据库存储的信息有: 电力系统的结构、断路器和继电保护装置间的连接以及每个断路器的分合状态。

电力系统用节点和支路来描述。节点表示电力系统元件, 如线路、母线等; 支路代表断路器。图 1 是一个网络结构的例子, 图 2 是它的等值电路。这个等值电路有以下特点: 一个节点对应继电保护装置保护范围内的一个基本单元, 而整组保护装置对应一组节点。

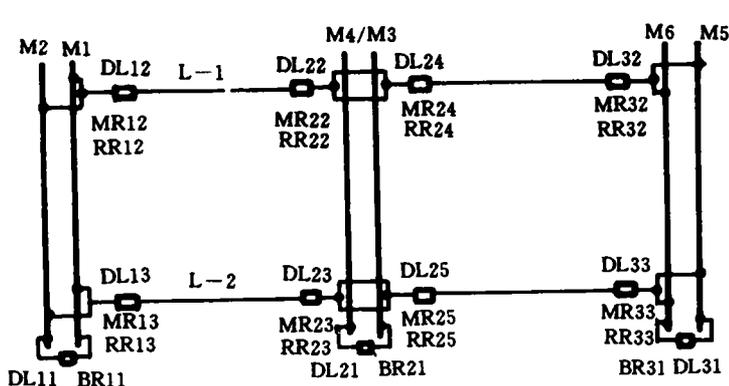


图 1 原型网络图

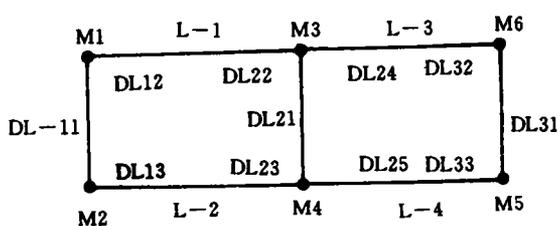


图 2 等值电路图

network (M<sub>1</sub>, DL<sub>12</sub>, L-1, on)  
network (M<sub>2</sub>, DL<sub>13</sub>, L-2, on)  
network (L-1, DL<sub>22</sub>, M<sub>3</sub>, on)  
network (M<sub>3</sub>, DL<sub>21</sub>, M<sub>4</sub>, on)  
network (L-1, DL<sub>22</sub>, M<sub>3</sub>, on)  
network (L-2, DL<sub>23</sub>, M<sub>4</sub>, on)  
network (M<sub>3</sub>, DL<sub>24</sub>, L-3, on)  
network (M<sub>4</sub>, DL<sub>25</sub>, L-4, on)  
network (M<sub>5</sub>, DL<sub>31</sub>, M<sub>6</sub>, on)  
network (L-3, DL<sub>32</sub>, M<sub>5</sub>, on)  
network (L-4, DL<sub>33</sub>, M<sub>6</sub>, on)

图 3 Prolog 语言

在这个数据库中, 用下述语句来描述一组事实所表达的图形关系:

```
network (<node name>, <DL name>, <node name>, on/off.)
```

注: network—网络; node—节点; DL—断路器; name—名称; on—合; off—分。

例如图 3 中语句第一行含义是: 在“M<sub>1</sub>”和“L-1”之间的断路器 DL<sub>12</sub>是闭合的。

数据库 2: 保护系统原理。

在实际网络中, 继电保护是很复杂的, 我们暂且将图 1 网络的保护简化为三种型式:

MR 型—差动保护装置, 它从两个及以上变电站取得信号, 作为线路的主保护 (即高频保护)。

RR 型—距离保护装置, 仅反映一个变电站信号, 作为后备保护。

BR 型—母线差动保护装置, 作为母线的主保护。

## 2.2 保护语句

```
relay (<name>, <type>, <DL>, <mode>)
```

注: <name>—保护名称; <type>—保护型式; <mode>—主保护保护范围。

## 2.3 继电保护动作规则

```
operate(Relay, Fault):—Protect(Relay, Fault)
```

```
Protect (Relay, Fault):—relay (Relay, MR, DL, Zones), member (Fault, Zones)
```

```
Protect (Relay, Fault):—relay (Relay, RR, DL, Zones), member (Fault, Zones)
```

Protect (Relay, Fault): —relay (Relay, BR, DL, Zones), member (Fault, Zones)

这些语句说明, 如果“Fault”节点在保护“Relay”的保护范围“Zones”之内, 那么“Relay”就保护“Fault”。谓项“relay”从“Relay”获得信息。当发生故障时, 变量“Relay”的值与实际动作的保护名称相对应。变量“Zones”由一组节点来表示, 这些节点就是保护范围内的电气元件。谓项:“member”摘出“Zones”中的故障元素, 用它替换进“Fault”中, 从而得出一个故障位置的答案。

#### 2.4 保护装置及断路器拒动的测定规则

保护装置及断路器拒动将造成信息短缺, 或使测定结果出现  $n$  个可能的故障位置, 给确定真正故障点带来困难, 因此需要发现拒动的保护装置和断路器。

规则 1: 如果保护装置已经动作, 断路器收到了来自保护装置的跳闸信号, 但未跳闸, 则断路器拒动。

规则 2: 如果测定的故障位置在某个未动作的主保护装置的保护范围内, 则该主保护拒动。

以上所述的知识规则贮存在不同的数据库中, 为适应具体电网的需要, 可以更换或修改数据库, 及加以扩展, 因而它将广泛应用于各种结构的电力系统中。

### 3 实例

现以图 1 所示网络为例, 应用上述专家系统进行故障诊断。根据继电保护和断路器的动作情况, 寻找故障位置及判断保护或断路器的拒动情况。

#### 3.1 故障位置的测定

##### 3.1.1 线路故障

设故障发生在图 1 中线路 L—1 上, 保护及断路器动作可有以下几种情况:

- (1) 线路两侧 MR 型保护动作, 断路器 DL<sub>12</sub>、DL<sub>22</sub>跳闸;
- (2) 线路两侧 RR 型保护动作, 断路器 DL<sub>12</sub>、DL<sub>22</sub>跳闸;
- (3) 线路一侧 MR 型保护动作, 另一侧 RR 型保护动作, 两端断路器 DL<sub>12</sub>、DL<sub>22</sub>跳闸;
- (4) 线路一侧断路器跳闸, 另一侧未跳(例如 DL<sub>12</sub>跳闸, DL<sub>22</sub>未跳), 而由相邻线路保护动作, 使断路器 DL<sub>13</sub>、DL<sub>32</sub>及 DL<sub>33</sub>跳闸。

将以上四种情况的任一种信息输入到 Prolog 程序中, 皆可判定为线路 L—1 发生故障。

##### 3.1.2 母线故障

当故障发生在图 1 中母线 M<sub>1</sub> 或 M<sub>2</sub> 上时, 保护及断路器动作有以下几种情况:

- (1) 母线保护 BR<sub>11</sub>动作, 断路器 DL<sub>11</sub>、DL<sub>12</sub>跳闸, Prolog 程序判定为母线 M<sub>1</sub> 故障;
- (2) 母线保护 BR<sub>11</sub>动作, 断路器 DL<sub>11</sub>、DL<sub>13</sub>跳闸, Prolog 程序判定为母线 M<sub>2</sub> 故障;
- (3) 母线保护 BR<sub>11</sub>动作, 断路器 DL<sub>11</sub>、DL<sub>12</sub>、DL<sub>13</sub>跳闸, Prolog 程序判定为母线 M<sub>1</sub> 或 M<sub>2</sub> 故障;
- (4) 母线保护 BR<sub>11</sub>动作, 断路器 DL<sub>12</sub>、DL<sub>23</sub>跳闸, Prolog 程序判定为母线 M<sub>1</sub> 故障;
- (5) 母线保护 BR<sub>11</sub>动作, 断路器 DL<sub>13</sub>、DL<sub>22</sub>跳闸, Prolog 程序判定为母线 M<sub>2</sub> 故障;
- (6) 若保护 RR<sub>22</sub>、RR<sub>23</sub>动作, 断路器 DL<sub>22</sub>、RR<sub>23</sub>跳闸, Prolog 程序判定为母线 M<sub>1</sub> 或 M<sub>2</sub> 故障。

#### 3.2 保护、断路器拒动情况的判定

##### 3.2.1 线路(在 MR 型保护不投入使用的前提下, 以线路 L—1 故障为例)

(1) 线路保护  $RR_{12}$ 、 $RR_{22}$  动作，跳闸的断路器为  $DL_{22}$ 、 $DL_{23}$ ，Prolog 程序判定为断路器  $DL_{12}$  拒动；若跳闸的断路器为  $DL_{12}$ 、 $DL_{13}$ 、 $DL_{32}$ 、 $DL_{33}$ ，则 Prolog 程序判定为断路器  $DL_{22}$  拒动。

(2) 线路一侧保护动作，另一侧保护拒动，与此类似，线路两侧保护动作，而有一侧断路器拒动，要由其相邻线路后备保护动作切除故障，这时就要判断是保护拒动，还是断路器拒动。如：线路保护  $RR_{12}$  动作，跳开的断路器是  $DL_{13}$ 、 $DL_{32}$ 、 $DL_{33}$ ，Prolog 程序判定为  $RR_{22}$  拒动。对应各种输入情况，Prolog 程序还可判断出是哪侧断路器拒动。

(3) 两侧保护拒动或两侧断路器都拒动。此时要由两侧相邻线路后备保护动作切除故障，如：动作的断路器是  $DL_{13}$ 、 $DL_{23}$ 、 $DL_{32}$ 、 $DL_{33}$ ，则 Prolog 程序判断为保护  $RR_{12}$ 、 $RR_{22}$  拒动。若输入信息中有一侧的保护动作，则程序判定是另一侧保护拒动。若两侧保护都有信息输入，程序则判定是断路器  $DL_{12}$ 、 $DL_{22}$  拒动。

### 3.3.2 母线

(1) 设母线  $M_1$  故障，母线保护  $BR_{11}$  动作，断路器  $DL_{12}$ 、 $DL_{23}$  跳闸，Prolog 程序判断是断路器  $DL_{11}$  拒动。

(2) 同上，若断路器  $DL_{11}$ 、 $DL_{22}$  跳闸，则程序判断是断路器  $DL_{12}$  拒动。

(3) 若动作的断路器为  $DL_{12}$ 、 $DL_{13}$ 、 $DL_{11}$ ，则程序判断是保护  $BR_{11}$  拒动，故障发生在母线  $M_1$  或  $M_2$  上。

## 4 结论

从上述分析可以看出，只要输入正确、完整的系统信息，Prolog 程序就能正确判定出故障位置及保护和断路器的拒动情况，可以说它具有了人的智能。又由于程序是由计算机进行推理判断，计算速度远快于人的手算，因此，它可以有效地帮助运行人员进行事故处理和事故分析，提高电力系统运行的可靠性，它给电力系统全盘自动化提供了可靠保证。

### 参考文献

- 1 Jones, M, and Poole, D. " An expert system for educational diagnosis based on default logic" (1985) 673—683
- 2 Reggia, J. A. NOV. D. S and wang. r" A formal model of diagnostic inference I. Problem formulation and decomposition IHf. set 37 (1985) 227—256
- 3 Eckart walther" Power Network Diagnosis Using Causal and Temporal Reasoning" second Symposium on Expert System Application Power System, July 17—20 1989, soattle USA