



(12) 发明专利

(10) 授权公告号 CN 113759245 B

(45) 授权公告日 2024.06.21

(21) 申请号 202111076643.0

(51) Int.Cl.

(22) 申请日 2021.09.14

G01R 31/327 (2006.01)

(65) 同一申请的已公布的文献号

(56) 对比文件

申请公布号 CN 113759245 A

CN 109374994 A, 2019.02.22

CN 110865262 A, 2020.03.06

(43) 申请公布日 2021.12.07

审查员 梁裕

(73) 专利权人 许昌开普检测研究院股份有限公司

地址 461000 河南省许昌市尚德路17号

(72) 发明人 姚致清 李蕾 周鹏鹏 李志勇
张彦兵 傅润炜 贾德峰 陈朋
李丹阳 张冉 胡卫东 王峥夏
陈明 闫黎明

(74) 专利代理机构 广州臻唯知识产权代理事务
所(普通合伙) 44991

专利代理师 赵蕊红

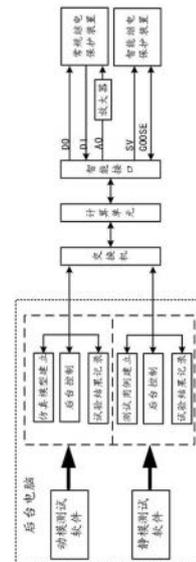
权利要求书2页 说明书5页 附图8页

(54) 发明名称

基于统一硬件平台的继电保护静模测试和动模测试方法

(57) 摘要

本发明提供一种基于统一硬件平台的继电保护静模测试和动模测试方法,包括如下步骤:后台电脑安装动模测试软件和静模测试软件,完成模型和测试用例建立、后台控制、试验结果记录功能,后台测试软件通过交换机与计算单元通信,计算单元负责测试命令处理、模型计算,并通过智能接口与继电保护装置实现模拟量、开光量、SV及GOOSE的交互,计算单元获取测试结果后返回给后台软件处理并呈现给测试人员查看。本发明方法使得硬件平台既能执行电力系统等值模型代码,又能执行静模测试用例脚本生成的测试命令。



1. 一种基于统一硬件平台的继电保护静模测试和动模测试方法,其特征在于:包括如下步骤:

后台电脑安装动模测试软件和静模测试软件,完成模型和测试用例建立、后台控制、试验结果记录功能,后台测试软件通过交换机与计算单元通信,计算单元负责测试命令处理、模型计算,并通过智能接口与继电保护装置实现模拟量、开光量、SV及GOOSE的交互,计算单元获取测试结果后返回给后台软件处理并呈现给测试人员查看;

静模测试实现方法包括如下步骤:

- a) 定义后台软件与计算单元的接口及接口参数;
- b) 建立测试用例;
- c) 后台控制及测试;

定义后台与计算单元的接口及接口的参数,接口的功能包括启动联机、下发配置参数、下发测试参数、开始测试、触发测试、停止测试、获取结果和获取异常信息,接口参数包括配置参数、联机参数、测试参数和结果参数;

测试用例采用xml语言,树形结构,整个测试用例包括4层结构,分别是测试大组、测试项目、测试序列、测试操作;

继电保护测试用例按照继电保护检测的条件和检测规程通过测试操作组合,通过基础操作设置试验各个时间段的状态以及中断条件,按照继电保护检测的条件和检测规程对各个试验状态和中断条件对测试操作进行排列组合,构成一个测试序列,最后生成测试项目,测试项目的内容包括装置参数修改、施加试验状态、读取表计数据环节,同一个保护的不同测试项目构成一个测试大组,不同保护的测试大组构成一个装置的测试用例;

计算单元采用高性能工控机,CPU采用Intel Core i7,搭载实时Linux操作系统;

Linux操作系统包括硬实时内核和常规Linux核心;

硬实时内核,用于运行实时核心;

常规Linux核心,用于运行非实时核心;

对Linux核心进行改动:将其与中断控制器隔离,不再允许Linux核心任意关中断,让实时核心控制中断控制器,所有的中断首先被实时核心所截获,实时核心首先进行相关的中断处理,然后再把中断“传”给Linux核心,这样Linux核心的一切活动都无法关闭中断,也就无从影响实时核心的任务调度,从而保证了实时核心的硬实时性;

改变时钟中断机制:RT-Linux通过将Linux操作系统的实时时钟设置为单次触发状态,可以提供十几个微秒级的调度粒度,RT-Linux中实时进程是运行在硬实时内核上,这样进程切换的开销远比常规的进程切换的开销要小得多,所以,RT-Linux最小实时进程周期可以达到100 μ s以内;

根据定义的接口、接口参数及测试用例建立方法开发静模测试软件,在测试软件上建立测试用例,以xml格式保存,在进行测试时,测试软件通过接口程序将测试用例转换为计算单元可以识别的接口命令并下发给计算单元,同时,计算单元的执行结果也通过接口程序翻译后返回测试软件;测试软件界面分为测试大组、测试模板及测试项目、已测试的结果文件及测试结果及事件报文四个部分,均以清单式方式呈现;

动模测试实现方法包括如下步骤:

- a) 建立电力系统等值模型;

- b) 生成代码和可执行程序;
- c) 后台控制及测试;

动模测试首先对电力系统建立一次等值建模,采用Matlab或Simulink作为模型开发软件,通过图形化建模的方式搭建电力系统一次仿真模型,接口控制部分通过自定义元件搭建,FromHMI功能块实现后台发送至仿真系统的信号的控制,ToHMI功能块实现仿真系统发送至后台的信号的控制,FromPKT功能块实现外部被测装置发送至仿真系统的信号的控制,ToPKT功能块实现仿真系统发送至外部被测装置的信号的控制,ToREC功能块实现记录录波通道生成Comtrade文件的功能,FromGPS功能块实现仿真系统与GPS的对时功能,Breaker功能块实现对开关的控制,Fault功能块实现对故障的控制,Signal功能块实现对信号的控制,Source功能块实现对系统电源的控制。

2. 如权利要求1所述的基于统一硬件平台的继电保护静模测试和动模测试方法,其特征在于:电力系统等值模型建立以后,通过Simulink代码生成工具将建立的模型生成ANSIC代码,然后将代码上传至计算单元生成可执行程序实时运行计算。

3. 如权利要求1所述的基于统一硬件平台的继电保护静模测试和动模测试方法,其特征在于:后台程序通过界面实现对测试项目的灵活配置,将具备相同触发条件的测试项目归为一个试验大组,分别设定试验大组的试验前状态、试验后状态、试验前等待时间、试验后复归时间、动作时间参考项;将模型的多个状态操作组合成一个操作来触发,以此来进行具体试验项目的设定;试验项目结合所属试验大组的试验属性共同构成了一个试验项目的完整信息,所有的试验大组及试验项目都通过清单的形式呈现。

基于统一硬件平台的继电保护静模测试和动模测试方法

技术领域

[0001] 本发明涉及继电保护装置测试技术领域,具体涉及一种基于统一硬件平台的继电保护静模测试和动模测试方法。

背景技术

[0002] 电力系统二次控制及保护设备在研发、出厂、入网和投运前都需要经过严格的静模测试和动模测试以保证功能和性能满足相关标准要求。当前,静模测试和动模测试主要是通过相关专用测试设备来实现的。静模测试设备主要有Omicron、昂立、博电、豪迈等厂家生产的测试设备,该类测试设备主要通过静态输出、状态序列、递变等方式输出特定的模拟量、开关量或数字量,进而实现继电保护装置基本性能的测试。动模测试设备主要有加拿大RTDS公司生产的实时数字仿真器和中国电科院生产的ADPSS实时数字仿真器,该类测试设备主要通过对电力系统等值数学模型实时计算并通过接口设备输入输出模拟量、开关量或数字量,进而实现继电保护装置功能逻辑的测试,测试设备需要具有较高的计算能力和实时性。

[0003] 目前,上述静模测试设备只能实现静模测试,其硬件的计算能力和实时性不能满足动模测试的需求。上述动模测试设备的软件功能只针对动模测试开发,只能开展动模测试,无法满足静模测试的需求。这两类测试设备硬件平台相互独立,互不通用。

[0004] 对于从事继电保护测试的实验室来说,既需要采购静模测试设备也需要采购动模测试设备,需要较大的经济投入。同时,在开展测试时,静模测试和动模测试需要分开接线,费时费力。

发明内容

[0005] 有鉴于此,本发明提供一种基于统一硬件平台的继电保护静模测试和动模测试方法,使得硬件平台既能执行电力系统等值模型代码,又能执行静模测试用例脚本生成的测试命令。

[0006] 为解决上述技术问题,本发明提供一种基于统一硬件平台的继电保护静模测试和动模测试方法,包括如下步骤:

[0007] 后台电脑安装动模测试软件和静模测试软件,完成模型和测试用例建立、后台控制、试验结果记录功能,后台测试软件通过交换机与计算单元通信,计算单元负责测试命令处理、模型计算,并通过智能接口与继电保护装置实现模拟量、开光量、SV及GOOSE的交互,计算单元获取测试结果后返回给后台软件处理并呈现给测试人员查看。

[0008] 进一步的,静模测试实现方法包括如下步骤:

[0009] a) 定义后台软件与计算单元的接口及接口参数;

[0010] b) 建立测试用例;

[0011] c) 后台控制及测试。

[0012] 进一步的,定义后台与计算单元的接口及接口的参数,接口的功能包括启动联机、

下发配置参数、下发测试参数、开始测试、触发测试、停止测试、获取结果和获取异常信息，接口参数包括配置参数、联机参数、测试参数和结果参数

[0013] 进一步的，测试用例采用xml语言，树形结构，整个测试用例包括4层结构，分别是测试大组、测试项目、测试序列、测试操作；

[0014] 继电保护测试用例按照继电保护检测的条件和检测规程通过测试操作组合，通过基础操作设置试验各个时间段的状态以及中断条件，按照继电保护检测的条件和检测规程对各个试验状态和中断条件对测试操作进行排列组合，构成一个测试序列，最后生成测试项目，测试项目的内容包括装置参数修改、施加试验状态、读取表计数据环节，同一个保护的不同测试项目构成一个测试大组，不同保护的测试大组构成一个装置的测试用例。

[0015] 进一步的，根据定义的接口、接口参数及测试用例建立方法开发静模测试软件，在测试软件上建立测试用例，以xml格式保存，在进行测试时，测试软件通过接口程序将测试用例转换为计算单元可以识别的接口命令并下发给计算单元，同时，计算单元的执行结果也通过接口程序翻译后返回测试软件；测试软件界面分为测试大组、测试模板及测试项目、已测试的结果文件及测试结果及事件报文四个部分，均以清单式方式呈现。

[0016] 进一步的，动模测试实现方法包括如下步骤：

[0017] a) 建立电力系统等值模型；

[0018] b) 生成代码和可执行程序；

[0019] c) 后台控制及测试。

[0020] 进一步的，动模测试首先对电力系统建立一次等值建模，采用Matlab/Simulink作为模型开发软件，通过图形化建模的方式搭建电力系统一次仿真模型，接口控制部分通过自定义元件搭建，“FromHMI”功能块实现后台发送至仿真系统的信号的控制，“ToHMI”功能块实现仿真系统发送至后台的信号的控制，“FromPKT”功能块实现外部被测装置发送至仿真系统的信号的控制，“ToPKT”功能块实现仿真系统发送至外部被测装置的信号的控制，“ToREC”功能块实现记录录波通道生成Comtrade文件的功能，“FromGPS”功能块实现仿真系统与GPS的对时功能，“Breaker”功能块实现对开关的控制，“Fault”功能块实现对故障的控制，“Signal”功能块实现对信号的控制，“Source”功能块实现对系统电源的控制。

[0021] 进一步的，电力系统等值模型建立以后，通过Simulink代码生成工具将建立的模型生成ANSI C代码，然后将代码上传至计算单元生成可执行程序实时运行计算。

[0022] 进一步的，后台程序通过界面实现对测试项目的灵活配置，将具备相同触发条件的测试项目归为一个试验大组，分别设定试验大组的试验前状态、试验后状态、试验前等待时间、试验后复归时间、动作时间参考项；将模型的多个状态操作组合成一个操作来触发，以此来进行具体试验项目的设定；试验项目结合所属试验大组的试验属性共同构成了一个试验项目的完整信息，所有的试验大组及试验项目都通过清单的形式呈现。

[0023] 进一步的，计算单元采用高性能工控机，CPU采用Intel Core i7，搭载实时Linux操作系统。

[0024] 本发明的上述技术方案的有益效果如下：

[0025] 1) 在同一个硬件平台上既可以实现静模测试又能实现动模测试；

[0026] 2) 静模测试和动模测试可以无缝切换，较少接线过程，提高测试效率；

[0027] 3) 因为采用同一平台就可实现静模测试和动模测试，无需采购额外的设备，可以

减少继电保护检测实验室设备采购投入。

附图说明

- [0028] 图1为本发明基于统一硬件平台的继电保护静模测试和动模测试方法的整体结构图；
- [0029] 图2为本发明测试用例的结构示意图；
- [0030] 图3为本发明后台软件与测试仪进行交互的结构示意图；
- [0031] 图4为本发明静模测试软件界面图；
- [0032] 图5为本发明动模测试电力系统仿真模型图；
- [0033] 图6为本发明动模测试软件界面图；
- [0034] 图7为本发明动模测试软件监控界面图；
- [0035] 图8为现有计算时间统计图；
- [0036] 图9为本发明多核任务协调机制对抖动的改善后计算时间统计图。

具体实施方式

[0037] 为使本发明实施例的目的、技术方案和优点更加清楚,下面将结合本发明实施例的附图1-9,对本发明实施例的技术方案进行清楚、完整地描述。显然,所描述的实施例是本发明的一部分实施例,而不是全部的实施例。基于所描述的本发明的实施例,本领域普通技术人员所获得的所有其他实施例,都属于本发明保护的范围。

[0038] 实施例一

[0039] 如图1-9所示:一种基于统一硬件平台的继电保护静模测试和动模测试方法,包括如下步骤:

[0040] 后台电脑安装动模测试软件和静模测试软件,完成模型和测试用例建立、后台控制、试验结果记录功能,后台测试软件通过交换机与计算单元通信,计算单元负责测试命令处理、模型计算,并通过智能接口与继电保护装置实现模拟量、开光量、SV及GOOSE的交互,计算单元获取测试结果后返回给后台软件处理并呈现给测试人员查看。

[0041] 根据本发明的一个实施例,如图2-4所示,

[0042] 静模测试实现方法包括如下步骤:

[0043] a) 定义后台软件与计算单元的接口及接口参数;

[0044] b) 建立测试用例;

[0045] c) 后台控制及测试。

[0046] 定义后台与计算单元的接口及接口的参数,接口的功能包括启动联机、下发配置参数、下发测试参数、开始测试、触发测试、停止测试、获取结果和获取异常信息,接口参数包括配置参数、联机参数、测试参数和结果参数

[0047] 测试用例采用xml语言,树形结构,整个测试用例包括4层结构,分别是测试大组、测试项目、测试序列、测试操作;

[0048] 继电保护测试用例按照继电保护检测的条件和检测规程通过测试操作组合,通过基础操作设置试验各个时间段的状态以及中断条件,按照继电保护检测的条件和检测规程对各个试验状态和中断条件对测试操作进行排列组合,构成一个测试序列,最后生成测试

项目,测试项目的内容包括装置参数修改、施加试验状态、读取表计数据环节,同一个保护的不同测试项目构成一个测试大组,不同保护的测试大组构成一个装置的测试用例。

[0049] 根据定义的接口、接口参数及测试用例建立方法开发静模测试软件,在测试软件上建立测试用例,以xml格式保存,在进行测试时,测试软件通过接口程序将测试用例转换为计算单元可以识别的接口命令并下发给计算单元,同时,计算单元的执行结果也通过接口程序翻译后返回测试软件;测试软件界面分为测试大组、测试模板及测试项目、已测试的结果文件及测试结果及事件报文四个部分,均以清单式方式呈现。

[0050] 在本发明的一个实施例中,如图5-7所示,

[0051] 动模测试实现方法包括如下步骤:

[0052] a) 建立电力系统等值模型;

[0053] b) 生成代码和可执行程序;

[0054] c) 后台控制及测试。

[0055] 动模测试首先对电力系统建立一次等值建模,采用Matlab/Simulink作为模型开发软件,通过图形化建模的方式搭建电力系统一次仿真模型,接口控制部分通过自定义元件搭建,“FromHMI”功能块实现后台发送至仿真系统的信号的控制,“ToHMI”功能块实现仿真系统发送至后台的信号的控制,“FromPKT”功能块实现外部被测装置发送至仿真系统的信号的控制,“ToPKT”功能块实现仿真系统发送至外部被测装置的信号的控制,“ToREC”功能块实现记录录波通道生成Comtrade文件的功能,“FromGPS”功能块实现仿真系统与GPS的对时功能,“Breaker”功能块实现对开关的控制,“Fault”功能块实现对故障的控制,“Signal”功能块实现对信号的控制,“Source”功能块实现对系统电源的控制。

[0056] 电力系统等值模型建立以后,通过Simulink代码生成工具将建立的模型生成ANSI C代码,然后将代码上传至计算单元生成可执行程序实时运行计算。

[0057] 后台程序通过界面实现对测试项目的灵活配置,将具备相同触发条件的测试项目归为一个试验大组,分别设定试验大组的试验前状态、试验后状态、试验前等待时间、试验后复归时间、动作时间参考项;将模型的多个状态操作组合成一个操作来触发,以此来具体试验项目的设定;试验项目结合所属试验大组的试验属性共同构成了一个试验项目的完整信息,所有的试验大组及试验项目都通过清单的形式呈现。

[0058] 在本发明的一个实施例中,如图8和图9所示,

[0059] 计算单元采用高性能工控机,CPU采用Intel Core i7,搭载实时Linux操作系统。

[0060] 同时实现静模测试和动模测试的基础是保证计算单元的实时性,这需要解决实时操作系统以及计算时间抖动的问题。

[0061] 通过研究实时操作系统的整体特征后发现,绝大多数实时应用都可分为实时和非实时两部分,而且这两部分分别具有如下特征:实时部分较少需要操作系统支持,甚至不需要这样的支持,至少应用程序的设计者可以做到这一点;而非实时部分一般则需相应的操作系统支持,基于这样的特征,构造一个简单的硬实时内核,而应用的实时部分作为实时进程直接运行在这个硬实时内核之上;原来的常规Linux核心这时就作为一个优先级最低的任务也为这个实时内核所调度,而应用的非实时部分作为非实时进程运行在Linux核心之上,从而可以获得Linux核心所提供的一切服务。

[0062] 对Linux核心进行改动:将其与中断控制器隔离不再允许它任意关中断。此时中断

控制器由实时核心来控制,所有的中断首先被实时核心所截获。核心首先进行相关的中断处理,然后再把中断“传”给Linux核心。这样Linux核心的一切活动都无法导致中断被关闭,也就无从影响实时核心的任务调度,从而保证了核心的硬实时性,同时容易看出,这种方法依然维护了Linux核心中数据结构的完整性。

[0063] 改变时钟中断机制:RT-Linux需用粒度更细的时钟,Linux系统中一般的定时精度为10 ms,而RT-Linux通过将系统的实时时钟设置为单次触发状态,从理论上说,可以提供十几个微秒级的调度粒度。RT-Linux中实时进程是运行在核心空间,这样进程切换的开销远比常规的进程切换的开销要小得多,所以,RT-Linux上有意义的最小实时进程周期可以达到100 μ s以内。

[0064] 实时操作系统保证了计算步长的稳定性,在不同的步长内模型的实际计算时间同样需要高度的稳定性,这就需要解决计算时间抖动的问题。

[0065] 基于intel架构的硬件并不是为实时计算设计的,由于节能变频、热量控制、中断共享、Cache抖动等原因,最终造成实时运算的时间抖动。时间抖动会造成每个计算步长内的实际模型计算时间有较大偏差,虽然没有计算溢出,但较大的抖动对实时仿真能力以及数字化报文输出的等间距性都有很大影响。

[0066] 从图8可以看出,在统计的50个计算步长中会比较频繁地出现大的抖动,抖动时间能达到30 μ s以上,通过反复研究比对,引入了多核任务协调机制,通过控制CPU非模型运算核的运算负载来降低模型运算核的时间抖动。

[0067] 图9中统计了50个计算步长的实际计算时间,可以看出,通过多核任务协调机制,计算时间抖动缩减到3 μ s以内,精确保证了计算的等时性。

[0068] 在本发明中,除非另有明确的规定和限定,例如,可以是固定连接,也可以是可拆卸连接,或成一体;可以是机械连接,也可以是电连接;可以是直接相连,也可以通过中间媒介间接相连,可以是两个元件内部的连通或两个元件的相互作用关系,除非另有明确的限定,对于本领域的普通技术人员而言,可以根据具体情况理解上述术语在本发明中的具体含义。

[0069] 以上所述是本发明的优选实施方式,应当指出,对于本技术领域的普通技术人员来说,在不脱离本发明所述原理的前提下,还可以作出若干改进和润饰,这些改进和润饰也应视为本发明的保护范围。

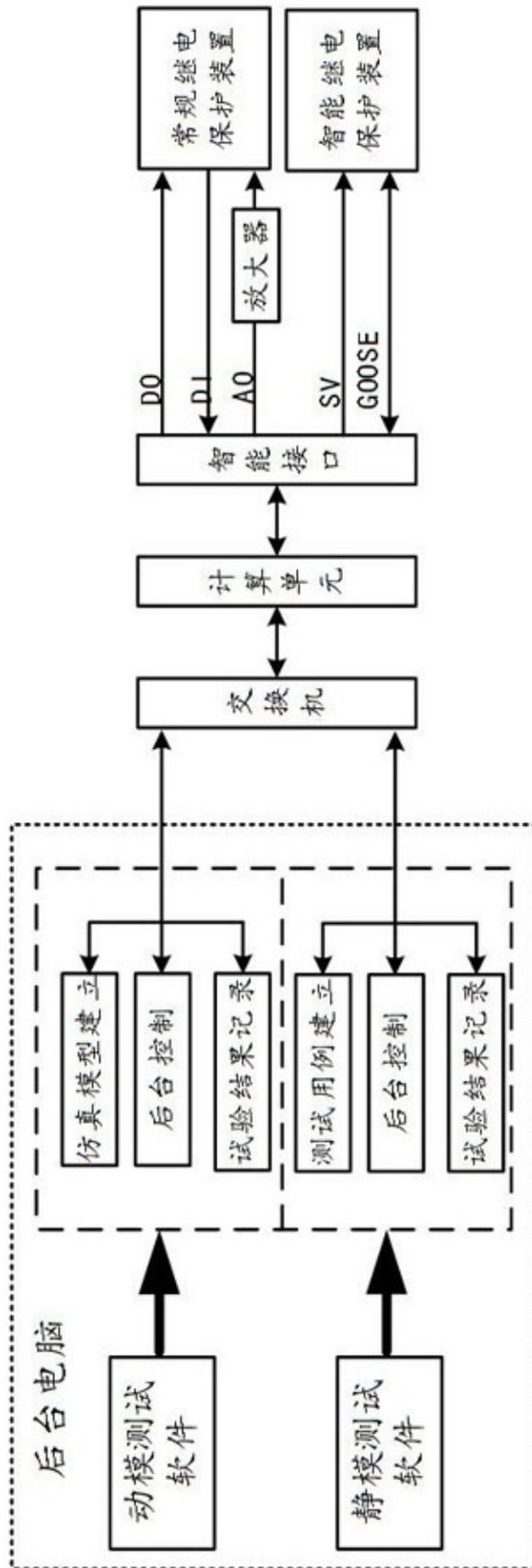


图1

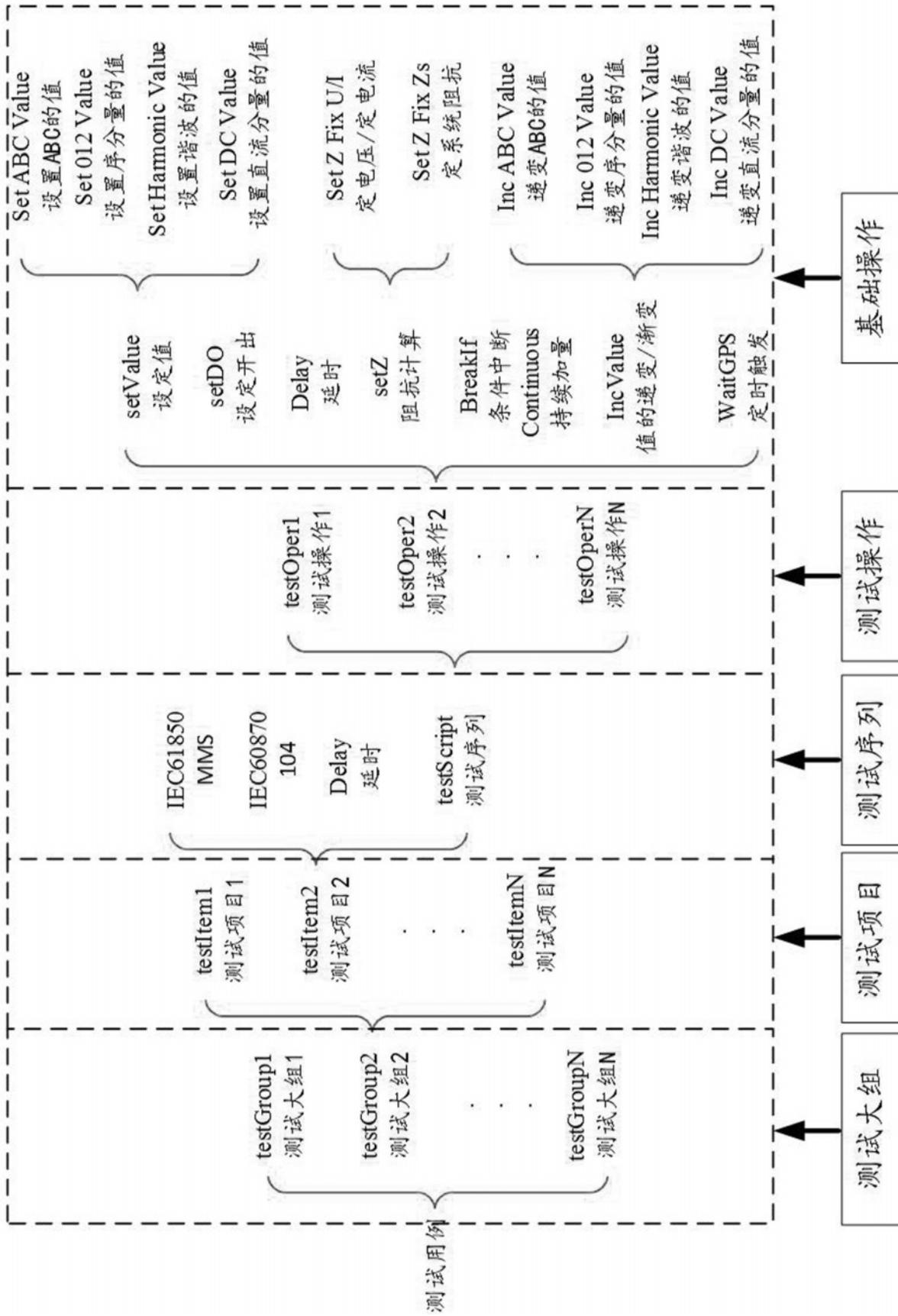


图2

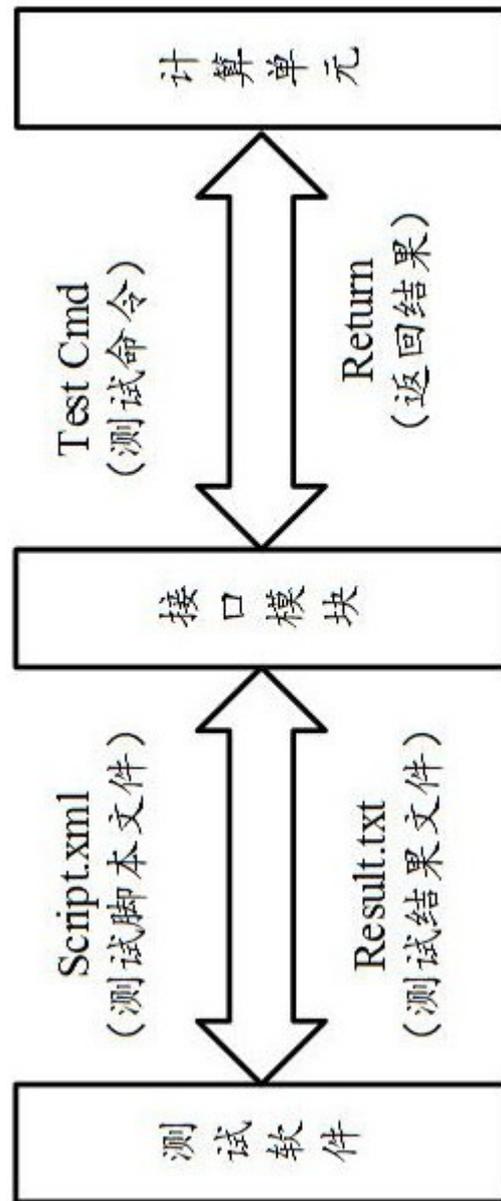


图3

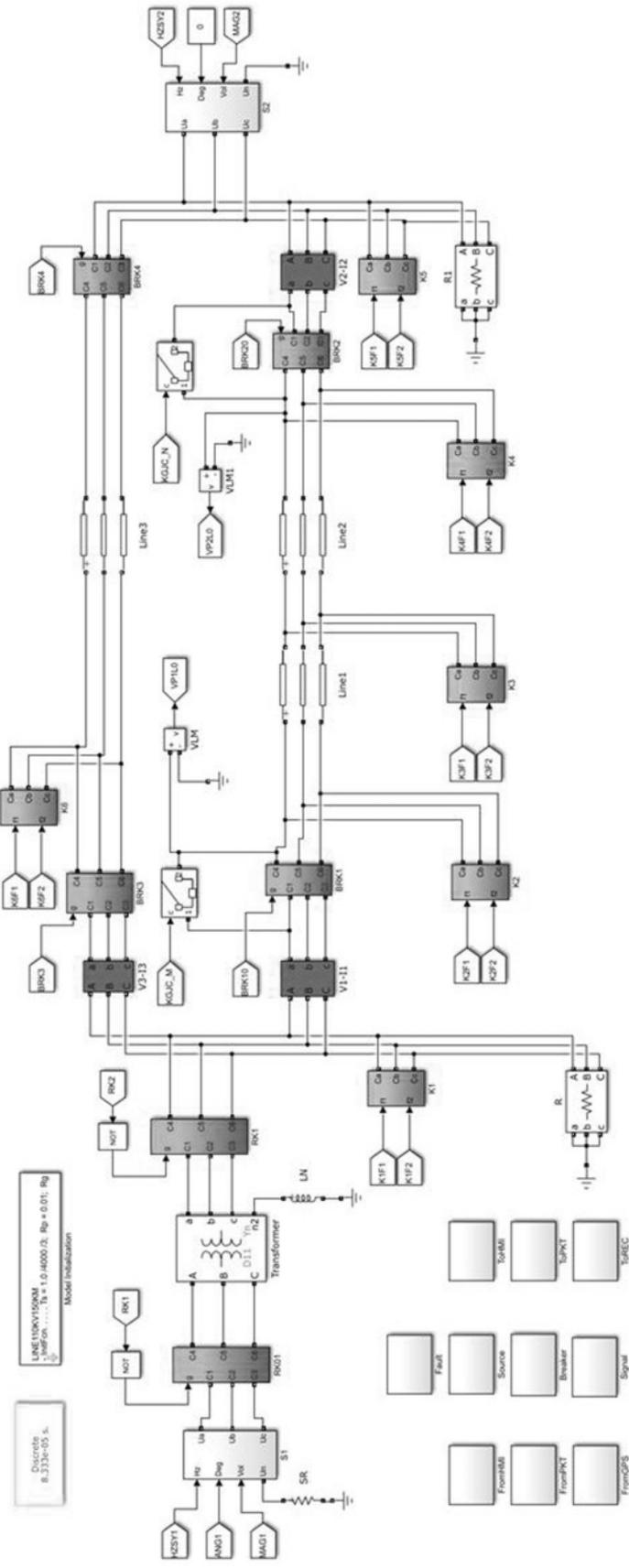


图5

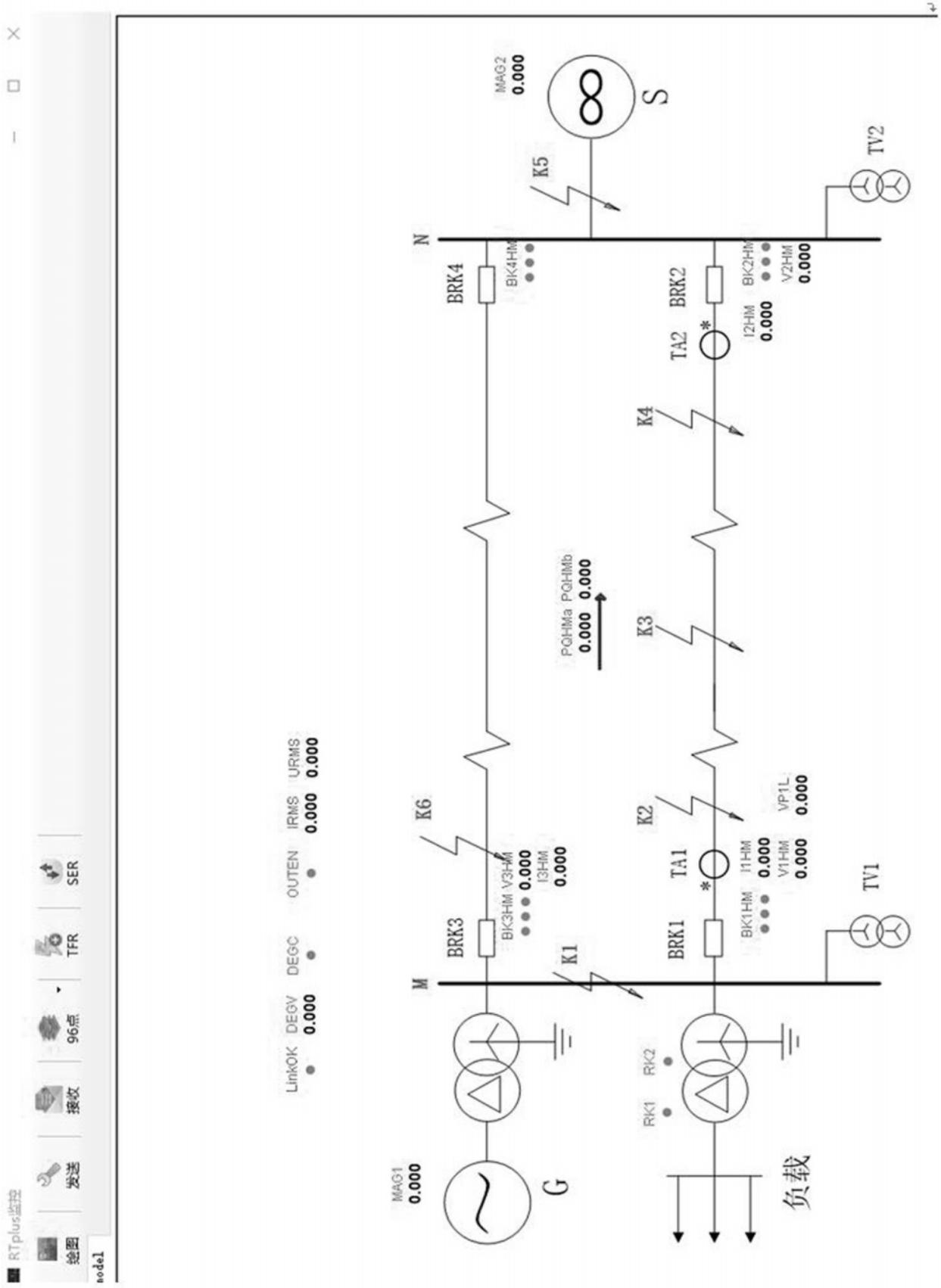


图7

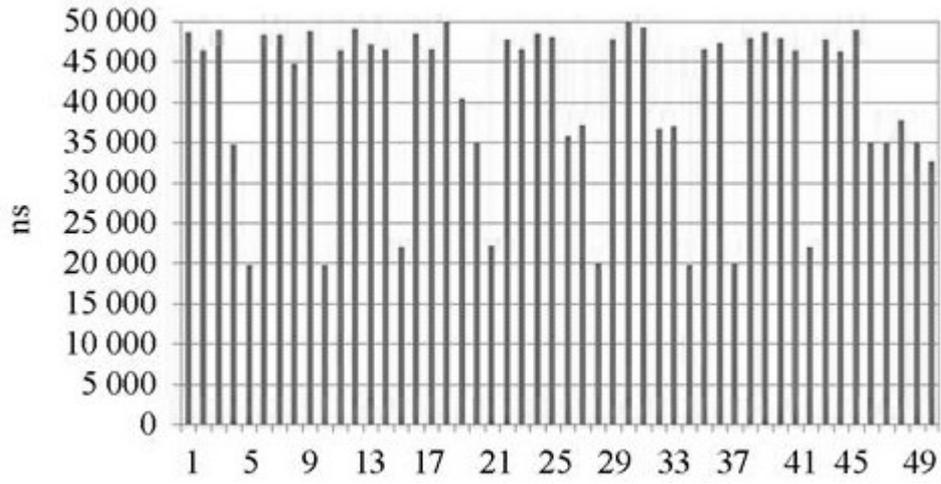


图8

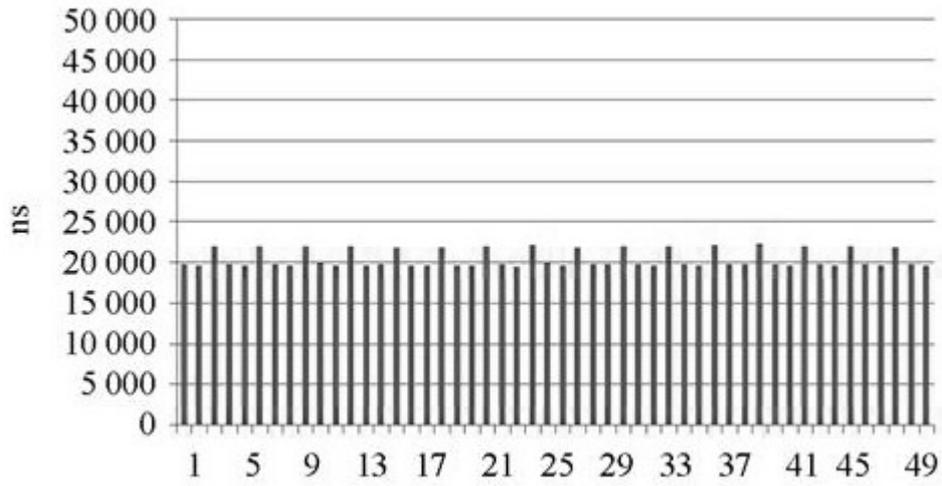


图9