

MQTester 修改内容列表

2020.06 (版本 MQT2020(3.0))

1. bug fix: 一个 testobj 中存在多个 testbed 时或者多次模拟时覆盖度设置不同, 无法累加覆盖度数据
2. bug fix: 带 coverage 仿真后, 在 2014b 中无法再次打开覆盖度设置对话框
3. testbed 具有界面有效性设置后, 如果某些输入信号被改变或者输出信号被屏蔽, 产生评估函数时输入信号取改变的值, 输出信号不加入到评估函数中
4. 对 CSV 格式的支持, 增加参数 PropGUIDefault.tools.useExcelorCsv=1;
5. bug fix: 测试对象的顶层包含 enable trigger 时, 生成的 SIL 环境无法正确连接信号线
6. 在测试对象顶层输入端口包含 ifaction, functioncall 时, 无法生成 SIL 环境, 给出提示。注意, 输出端口是 ifaction 和 functioncall 时也无法生成 sil, 但是这无法提示, 需要测试人员注意
7. testbed 的文件名里面包含的测试对象名称长度在为测试对象生成 testbed 时为 21, 为测试组时为 13, 为测试用例时为 5.
8. 支持 subsystem reference: 自动为包含参考子系统的模型建立 testbed, 生成 SIL 测试环境等所有模型测试工作。
9. 建立 SIL 时删除 slprj 目录, 以避免因不同版本 matlab 产生的问题
10. 建立 SIL 前, 检查 CodeGenFolder 变量, 设置为当前被测对象目录, 以避免设置而带来建立 SIL 过程的中断
11. bug fix: 在 2016b 以前版本, 只能对整个 testbed 进行覆盖度仿真, 否则报告中无法出现覆盖度内容
12. 2010a 中使用 get_param(, ReferencedWSVars ‘) 无法得到所使用的 busobj 变量
13. 拷贝 modelworkspace 到 testbed 中, 如果被测模型有 sldd, 2019a 及以后则连接到 testbed 中, 以前的版本中则将 sldd 的变量添加到 baseworkspace 中。添加测试对象时, 保存当前模型所使用的变量, 但是这个命令在 2013b 之前不能发现使用的 busobject 及用于数据类型的变量, 使用者可以将这些变量添加到 testsetu.m 中。参考模型的 modelworkspace 及 sldd 的内容不拷贝。
14. sldd 的路径添加到 startupfile 中
15. bugfix: 2010a 中无法取得 local 变量的值。现在需要为 matlab2010a 添加一个补丁, 然后才能 local 变量值。
16. TL 模型中的对内部 bus 信号的 logging 的支持
17. 对于复杂 bus 的支持, 任意复杂 bus, 包括使用 concatenate 生成的多维 bus, 均可以自动识别并生成 testbed
18. bus 信号可以不设置信号名, 其中 element 的信号的名字可以重复, 系统自动判断并处理。
19. 自动检查 TargetLink 模型接口是否符合命名规范, 不符合时无法产生测试环境, 建立测试环境时如果测试对象的名称不符合命名规范, 则自动产生符合命名规范的名称, 避免代码生成的问题
20. TL bus 信号中的总线元素的 logging
21. TL 中重用函数模型支持
22. TL 中对 mask 子系统的支持, 避免在顶层系统中出现原模块的 mask 内容, 从而导致双击时出现 mask 的内容

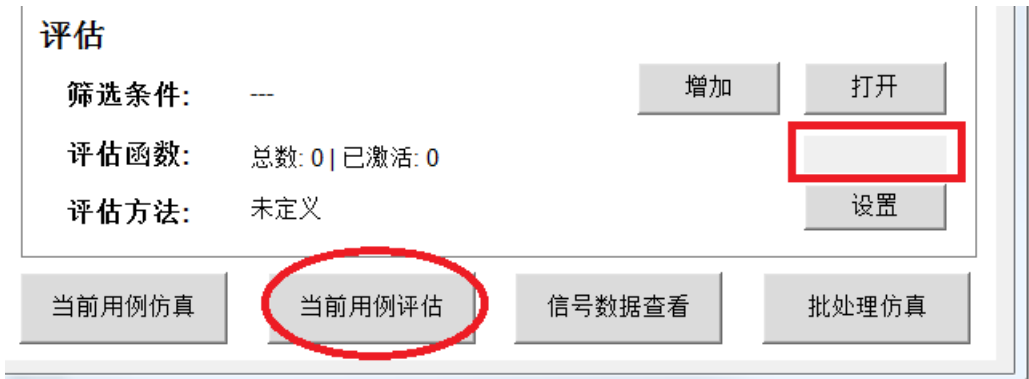
23. TL 模型中如果有外部 c 及 h 文件在建立 SIL 时需要，假设这些文件在 model 路径下，产生代码时拷贝到 testobj 所在目录
24. 模型中有 matrix 数据类型时，建立的测试模型中保持正确的 matrix 维数，以保证矩阵运算的正常进行。
25. TL 代码仿真时，支持对矩阵信号的 logging
26. 支持对包含 include DD 文件的模型的 MIL SIL 仿真，自动链接 include DD
27. TL 模型建立 SIL 时，参考模型和主模型使用相同的代码覆盖等级
28. 增加参数 execution.useOwnSolverSetting，为 1 时，不设置仿真参数 solver，使用 testbed 自身的设置
29. 增加参数 execution.defaultSolver，定义缺省的求解器设置，此参数影响生成的所有测试用例的设置。此参数设置新的值后，需要将原测试用例删除（帮助->清除数据->删除测试数据），否则原测试用例中的设置仍然被使用。
30. 增加参数 testbed.changeModelRefSimModeToNormal，设置为 1 时，在生成 Testbed 时将只被参考一次的 model block 中的仿真模式设置为 normal
31. 增加参数 import.checkInterfaceWhenNewTCSD，设置为 1 时，在新建测试 TCSD 格式的测试用例时重新检查被测对象的接口，否则则使用以前检查的结果

2020.03

1. 修复在进行期望值评估是无法找到 local 变量的问题
2. 生成信号比较配置文件时利用在 MQT 配置参数文件中的变量值定义缺省信号误差值
3. 在 MQT 配置文件中增加变量控制在进行期望值比较时强制使用误差值
4. 在 TCSD 中，在一个时间段中对一个变量多次赋值时，不相互覆盖的区域的值可以保存，如 In=set(1); In=set(2,0.2)，则 In 的值为 [1 1 2 2 2 2 2 2...]
5. 自动比较主模型和参考模型配置参数的不同，提供界面选择合适的统一的参数，并自动修改模型参数，使之一致
6. 增加检测接口信号名变化并自动修改相应测试用例评估函数的功能
7. 增加三个宏 \$AllInSig\$ \$AllOutSig\$ \$AllLocSig\$
8. 关闭 MQT 时关闭 MQT 打开的模型并删除 MQT 增加到 matlab 中的路径

2020.01

1. Workspace 参数增加不显示选项
模型中所用到的参数在生成测试用例时会自动添加到 initialization 部分，这方便了对个别参数值的修改，但是在参数很多时，会使 excel 表格变得很长，在不需要对参数进行修改时，可以调整 MOTester 的设置，不将这些参数添加到 Excel 表格中。
打开 MQTester\bin 目录下的 MQTester_GUIcommonProper.m 文件，找到参数 PropGUIDefault.import.addModelParameterToExcel，将此参数设置为 0，则参数名称不会出现在 Excel 表格中
2. 在主界面增加 当前用例评估 按钮，对仿真结果进行评估并在主界面显示评估结果
此功能方便使用者立即看到测试评估结果。



3. 测试用例中增加一列显示测试结果

新增菜单项 评估函数->拷贝评估结果到测试文档 将相应的评估结果拷贝到相应 Excel 文件的最后一列中。

Action	TCSD Work Status	Test Result
[+1.0s] // System initialization Tset = 10; // Assign a value to Tset Toutside=0; expDiff('Troom')=expExpress('@x>0 & @x,5'); [+10.0s] Tset=100; [+10.0s] //	reviewed	Error
Toutside=ramp(10); [+10s]		
[+1.0s] // System initialization Tset = 25; // Assign a value to Tset expDiff('Troom')=expRangeIn(1,3); [+10.0s] Tset=28; expDiff('Troom')=expValue(2); [+10.0s]	reviewed	Failed
Tset=30; [+10s]		
[+1.0s] // System initialization Tset = 25; // Assign a value to Tset pump_dir=expvalue(-1); pump_dir1=expvalue(-1); Troom=expvalue(10); pump_cmd=expvalue(1); [+10s] //Troom=expRangeIn(20,25); [+20.0s] //Troom=expExpress('Troom>10 & Troom<=20'); //pump_dir=expvalue(-1); [+5s]	reviewed	Failed

4. 增加 导数期望值函数 expDiff, 使用方法如下

此函数应用于定义测试用例的 Excel 表格的 action 部分, 用来定义给定信号的导数期望值。在导入测试用例后, 系统会自动产生针对此测试用例的评估函数。

函数调用	expDiff('signal') = expression;		
参数	描述	单位	默认值
signal	模型中的输入、输出或者 local 信号		
expression	定义信号 signal 的导数的期望值的表达式		
示例:	expDiff('Troom')=1;%Troom 信号的导数期望值 1 expDiff('Troom')=expValue(1);%Troom 信号的导数期望值 1 expDiff('Troom')=ramp(7);%Troom 信号的导数期望值为 ramp 函数的输出值 expDiff('Troom')= pulsewave (7, 2, 3);%Troom 信号的导数期望为 pulsewave 函数的输出值		

	<pre>expDiff('Troom')=expRange (1,3);%Troom的导数期望值在1 到 3 之间 expDiff('Troom')=expExpress (' expDiff('Troom')>3 && expDiff('Troom')<=10');%Troom的导数期望值在区间(3 10] expDiff('Troom')=expExpress ('@x>3 &&@x<=10');%功能同上, 可以用@x 代替 expDiff('Troom')</pre>
--	---

增加 diff 函数作为期望值函数，和 expDiff 区别是:diff 将某个输出信号的值设定为另一个信号的导数，expDiff 是设置某个信号的导数的期望值。

此函数应用于定义测试用例的 Excel 表格的 Action 部分，用来判断输出或者 local 信号是否为某个指定信号的导数。在导入测试用例后，系统会自动产生针对此测试用例的评估函数或者产生期望值。

函数调用	outSignal =diff(signal,duration, offset,tol);		
参数	描述	单位	默认值
outSignal	需要定义期望值的模型中的输出或者 local 信号		
signal	outSignal 的期望值被定义为 signal 的导数，signal 可以是输入信号，输出信号或者 local 信号。如果是输入信号，导入测试用例时会产生期望值，评估时可使用期望值比较进行评估，signal 是其它输入信号时，产生评估函数		
offset	相对本次时间区间的偏移量，默认为 0s		
duration	持续时间，默认为间隔时长		
tol	运行误差值，可是标量或者包含两个元素的向量，参考函数 MAF_SignalEquival中的误差使用介绍		
示例:	<pre>Troom=diff('T_outside' ,1, 4, [-1 2]); %Troom 信号的期望值为输入信号 T_outside 导数，时间偏移为 4s，持续 1s，误差设定范围为[-1 2] Troom=diff('loc_signal');%Troom 信号的期望值为 local 信号 loc_signal 的导数</pre>		

- 在导入需求时需要各项需求的可测性进行判断并作出审核，客户使用的模板和 MQTester 的标准模板会有不同，现在可以通过配置让系统根据用户模板中指定域的值对可测性进行判断，设置方法举例如下：

如，客户的模板中不包含可测性这一列，但是可以使用 需求类型 这一列的内容对需求的可测性进行判断，首先定义需求可测性的域名，选择需求分析->配置需求选项，出现如下对话框，此对话框中的域是需求文档(excel 表格)必须包含的内容，输入框中的是 Excel 表格中第一行的内容。输入框中的内容可以重复，如 需求类型域的名称和需求可测性的名称可以相同，这样，需求可测性的内容和需求类型相同。按确定后在 project 路径下产生 MQTester_DefineRCTPrjOption.m 文件，里面包含对话框设置的内容。如果 Excel 表格中不包含所有对话框中定义的域，则导入需求时会报错。此对话框中的 页名 域表示的是 Excel 中的 table 的名字。



然后设置文档可测性自动设置选项并定义可测性判定规则。打开在前面产生的MQTester_DefineRCTPrjOption.m 文件，设置参数 RCTopt.AutomaticTestabilitySettingAfterImport = true，然后定义自动判定可测性的规则。可测性由RSDTestability和RSDType两个域的值确定，这两个域之间的逻辑关系由下面第一行的参数确定，此例中二者之间是 or 的关系(可以是OR或者AND)，再下一行的参数定义了这两个域的可能值，只要有一行的值为真，则可测性被设置为 yes。如此例中，如果第一个域的值为 'Test'，第二个域的为 'functional requirement' 或者 'functional requirements'，则设置可测性为 yes。

```
RCTopt.AutomaticTestability_Operator = 'OR';
RCTopt.AutomaticTestability_Rules = {...
%RCT_Testability = RSDTestability RSDType
    'Yes', 'Test', 'functional requirement';
    'Yes', 'Test', 'functional requirements';
};
```

如果Excel中不包含RSDTestability，可以将这个域的值设置为其它域，例如 RSDType 域，这种情况下可以将规则设置如下(示例)

```
RCTopt.AutomaticTestability_Operator = 'OR';
RCTopt.AutomaticTestability_Rules = {...
%RCT_Testability = RSDTestability RSDType
    'Yes', 'functional requirements, 'functional requirement';
};
```

- 需求可测性审核状态由原始的 requirement 表格的 Staus 域统一获取并设置。
- 增加了需求文档及测试用例文档中 Status 域的关键字数量，并对关键字对应的工作量完成度进行了映射：

Specified	--	described	0.4
Draft	--	created	0.1

Analyzed -- compleed 0.8

Reviewed

Approved -- reviewed 1

8. 测试项目中的模型路径改为绝对路径，相对路径也能够识别。现在，MQT 不再向模型路径下写入任何文件，保证模型路径下模型文件不被影响。所有 MQT 的设置文件写到一个新的 startupFiles 目录下。模型路径和项目路径可以位于不同的盘符下，如项目在 D 盘，而模型在 C 盘。



9. 添加测试序列语法检查功能，选择下拉菜单 测试序列->文档有效性验证
在导入测试用例前，使用语法检查，保证测试用例中没有语法错误，发现错误后会指示错误的类型和具体位置

在表单: 'TCSD' 中, 发现下列语法错误:

在单元(6, 6) 中

'info init' 不是有效的赋值语句

在单元(8, 7) 中

表达式: '[+1. 0s]' 是无效的时间表达式

在单元(13, 7) 中

在表达式: 'pump_dir1=expvalue(-1)' 的左边部分的 'pump_dir1' 不是模型中的信号名(输入输出和局部信号名)

在参数 'Troom> 10 & Troom , 20' 中的子字符串 'Troom> 10 & Troom , 20' 的下列位置包含不允许的字符:

19 ','

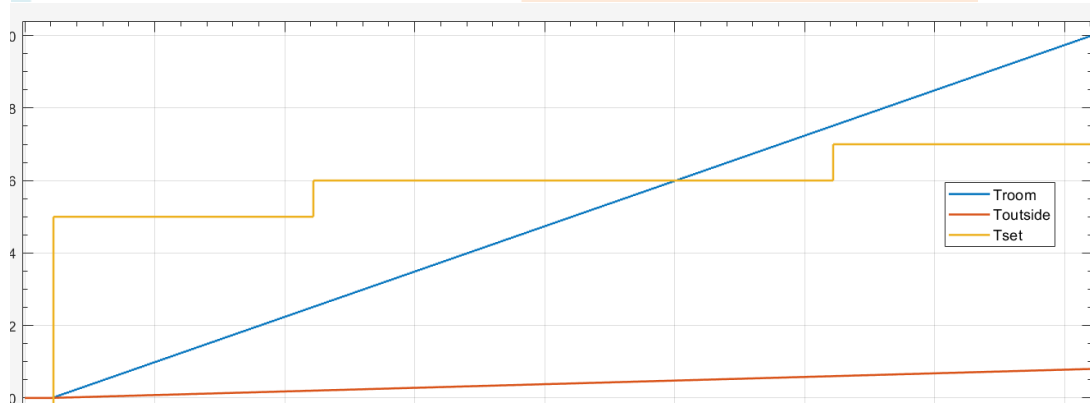
最后一行不是时间表达式, 这会导致最后时间表达式后面的语句无效

10. 所有的用于定义输入信号及定义输出信号期望值的函数(expPassFilter 除外)均提供 duration 参数, 使其变化不受间隔时长的影响, 下面的例子中, Troom 是输出信号, 需要定义其期望值, Toutside 和 Tset 是输入信号, 在初始化部分为 Troom 设定初始值, 在 action 部分定义期望值是梯度为 0.25 的函数的输出值, 定义持续时间为 50 秒, 此持续时间超过了整个测试序列的持续时间, 这样 Troom 的值按照梯度函数的定义持续到测试序列结束。Toutside 是室外温度, 定义在 1000 秒内从初始值 10 变化到 30, Tset 的值在每个时间间隔中有不同的值。这样使用下面的简单语句我们就得到这三个信号的输入值或期望值如下图。Duration 参数的使用简化了操作。

```

// Initialization of input signals (this
sequence)
Room=10;// Assign an initial value
[+1.0s] // System initialization
Tset = 15; // Assign a value to Tset
Room=gradient(0.25,50);
Toutside=ramp(30,1000);
[+10s]
Tset=16;
[+20.0s]
Tset=17;
[+10s]

```



11. 增加产生需求矩阵的函数，GenerateReqMatrix，该函数将需求和测试用例之间的对应关系以矩阵的形式填写到 Excel 表格中，方便需求和测试用例管理

Requirement ID	Requirement Description	Test Sequence (Group) ID	Test Sequence Name	Sequence Description
Heading	-			
Information	This document describes a set of requirements that apply to the altitude mode logic portion of an autopilot controller.			
Heading_req				
HP_req_01	Fan should start if the difference between room temp and set temp is bigger than a threshold	TS_001 (Test001)	Small and big time difference	Motivation Provide a short description of the requirement(s) the test is supposed to stimulate.
HP_req_02	pump and fan should both start if the difference between room temp and set temp is bigger than a bigger threshold	TS_001 (Test001)	Small and big time difference	Motivation Provide a short description of the requirement(s) the test is supposed to stimulate.
HP_req_03	if room temp is higher than the set temp and the pump is started, the direction of the pump should be negative	TS_002 (Test001)	heat or cool	Motivation Provide a short description of the requirement(s) the test is supposed to stimulate.
HP_req_04	if room temp is less than the set temp and the pump is started, the direction of the pump should be positive	TS_002 (Test001)	heat or cool	Motivation Provide a short description of the requirement(s) the test is supposed to stimulate.
HP_req_05	the set temp should be reached in abs(tset-room) seconds	TS_003 (Test001)	Small and big time difference	Motivation Provide a short description of the requirement(s) the test is supposed to stimulate.
		TS_004 (Test001)	Small and big time difference	Motivation Provide a short description of the requirement(s) the test is supposed to stimulate.

12. 对 TargetLink 参考模型及 disabled 参考模型模块的支持
全面支持对 TargetLink 模型中的参考模型模块及 disabled 参考模型模块的测试，其测试步骤和一般子系统一样。系统能自动识别模块属性并采取相应的解决方法。
13. 接口信号命名规范检查
在分析界面接口时自动检查接口信号名或者端口名是否符合命名规范，包括是否包含回车空格括号等不允许的字符，有不符合规范的端口，则不会生成 testbed，并指出错误之处。
14. 手册修改：修正错误，增加内容。