MQTester 2020

概述

MQTester 2020 支持 Matlab2019 以后新增加的功能，如参考子系统，sldd 新功能特点等，使我们能够充分利用 Matlab 的新功能增加对模型测试的灵活性及易用性。考虑到分布式开发的重要性，增强了对参考模型的支持。全面升级了对信号的检查，支持任意复杂的总线、矢量及矩阵信号。在 matlab2010a 版本中支持对主模型及参考模型中的内部(local)信号的记录(Logging).

提高了对 TargetLink 模型的支持，自动处理 Add block， 支持对 SIL 仿真中复杂结构信号的记录，对所有函数及参考模型中代码复杂度进行一致性设置。改善了批处理中多个模型的 SIL 仿真效率。

测试用例编写方面，增加了新的信号编写函数及自动评估的函数，扩展了 TCSD 的语法。

新的版本提高了自动化功能，提高了对复杂建模方法的适应性，测试人员能够有更多的时间关注模型功能的测试，而不是花费到测试软件的使用上面。

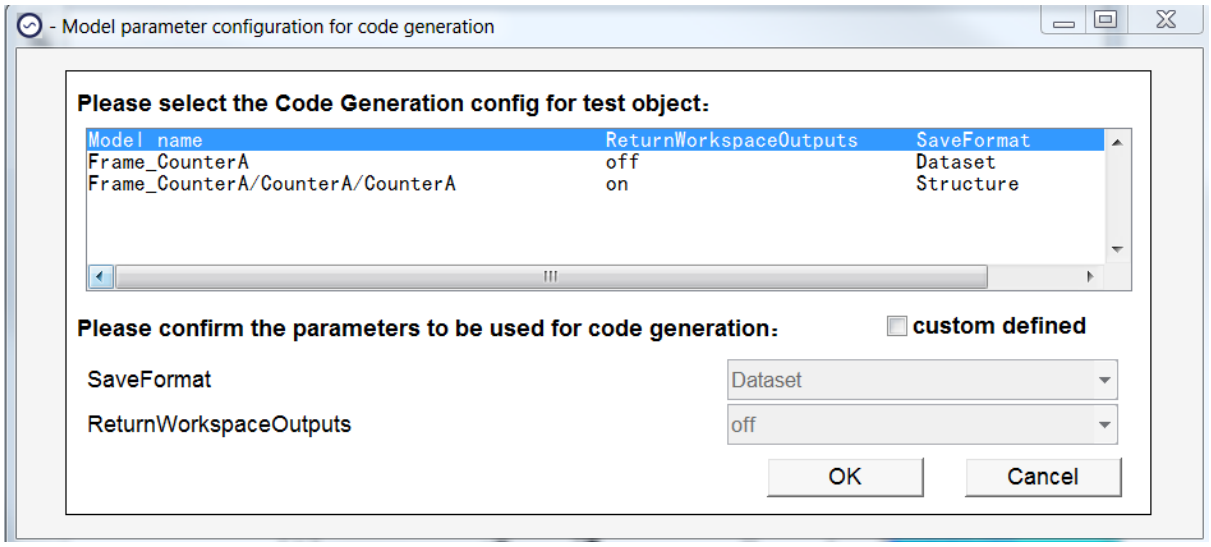
更多资讯，请访问我们的网站:www.model-soft.com

参考子系统

参考子系统是 matlab2019b 中增加的新的建模方式，可以更加灵活地进行模块的复用，但是这种方式和参考模型及库模块是有区别的，在建立测试环境及进行仿真时需要进行特殊设置，MQTester 2020 会自动识别并处理参考子系统。

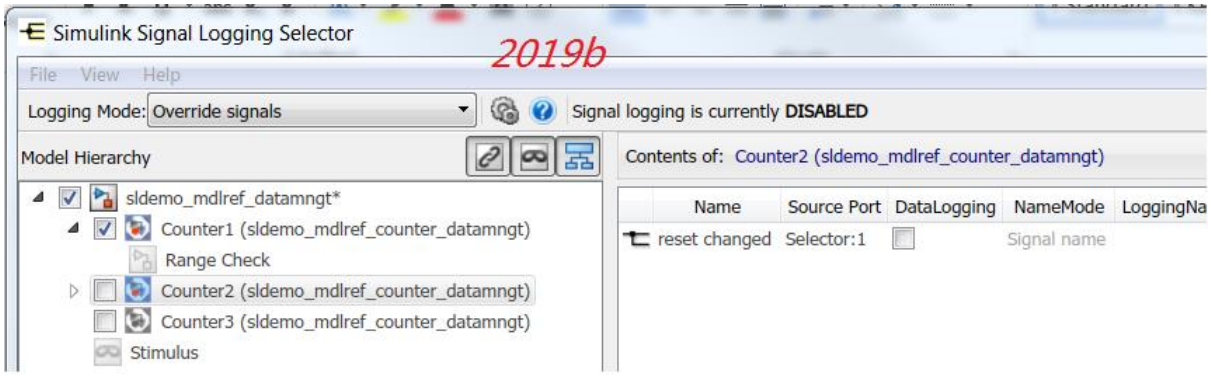
参考模型(EC)

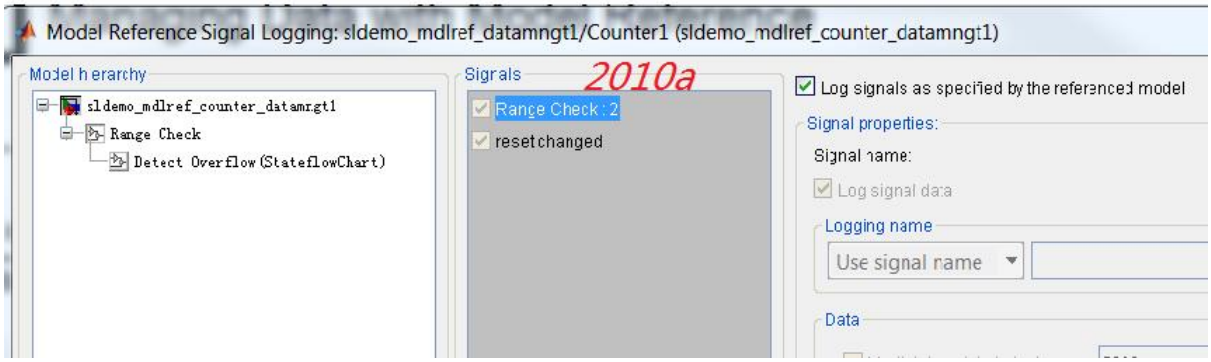
在生成 SiL 代码时，主模型和所有参考模型的部分参数设置需要保持一致，现在 MQTester 提供命令可以在生成代码之前进行一致性检查并统一修改。



生成代码时，所有参考模型不能有未保存的改动，不能处于编译状态，MQTester 在生成代码前自动保存所有参考模型、解除参考模型的编译状态并检查参数一致性。

参考模型中内部 logging 信号缺省情况下在主模型中不被记录，需要人工设置，在 matlab2010a 中需要为每一个 model 模块选择命令进行设置，在 2019 中可以统一进行设置，如下图。





MQTester 现在提供参数execution.ModelRefDataLogging, 此参数值为1 时，设置参考模型模块按照参考模型内部的设置进行信号记录。
如果参考模型的仿真模式没有被设置为 Normal，则内部信号也不会被记录，MQTester 提供参数 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。如果一个模型有多个参考，需要人工决定哪一个实例设置为 normal。

Model workspace 及 SLDD

被测模型的 Model workspace 中的所有参数被拷贝到 Testbed 的 model workspace 中。model workspace 中的变量不再和 base workspace 中的变量一起被写入参数文件中(仿真时自动调入)。被测模型的 SLDD 中的变量在 matlab2019a 之前和所使用的 base workspace 变量一起被写入参数文件，在 matlab2019a 及之后的版本被独立保存，并链接到建立的 Testbed 模型。

在添加测试对象时，MQTester将模型所使用的base workspace中的变量写入参数文件，但是有可能无法取得模型所使用的全部的base workspace中的变量，这时可以将遗漏的变量添加到配置文件中(选择界面中 测试对象 后面的 设置按钮 打开配置文件)或者将 MQTester_GUIcommonProper 中的变量testbed.getallbasevarOrRefvar设置为1，则将当前base workspace中的全部变量写入配置文件中。

在装载测试项目时，自动装载原始模型所需要的参数。

仿真数据存储

参数 testbed.simworkspace 控制测试用例数据在仿真时被保存的位置，设置为 1， 则保存到 current workspace，但是如果模型中被封装的模块中有封装参数时(mask parameter),这些参数的 storage class 只能被设置为 auto。将参数 testbed.simworkspace 设置为 0，则将测试用例数据保存到 base workspace 中，这可消除对封装参数的限制。缺省值是 0。

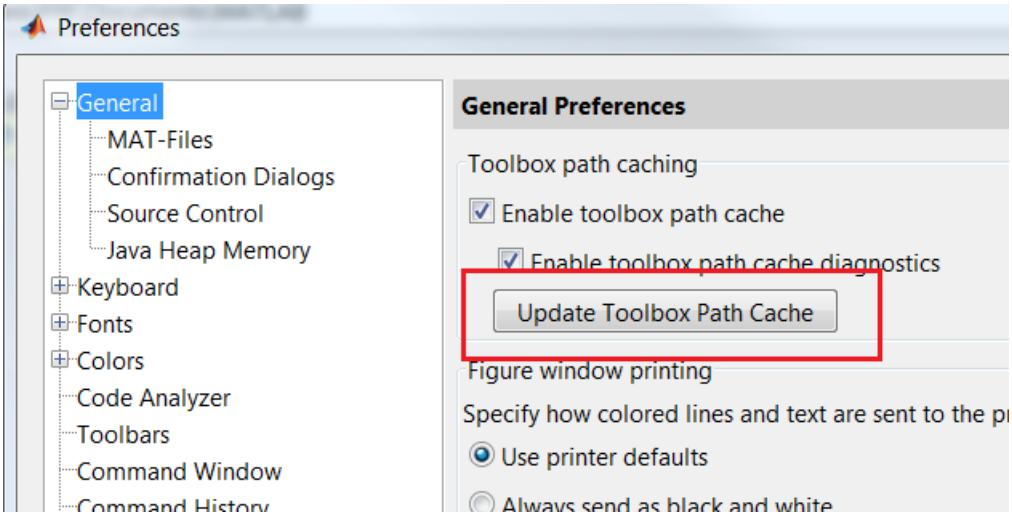
触发子系统(Trigger enable)

触发子系统在建立 SiL 时触发端口被移动到所有输入端口的下方，导致在建立 SiL 后信号线连接错误。现在 MQTester 能正确识别建立 SiL 后触发端口的端口号，避免信号线连接错误。

内部信号记录(2010a-2013a)

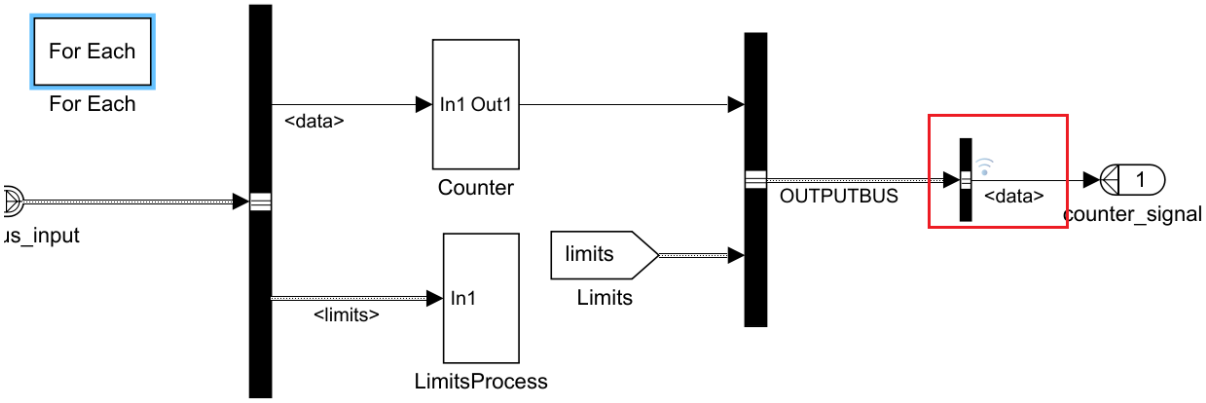
matlab 2013a 之前的版本使用旧的内部信号记录格式，MQTester 无法读取记录的内部信号内容，现在 MQTester 提供补丁，将此补丁安装到 matlab2010-2013a 中后，MQTester 能够读取内部记录信号的内容，内部信号同样可设置期望值并进行评估。

安装补丁到 matlab2010-2013a 中后，需要到 matlab preferences 中更新 matlab 工具箱路径缓存。

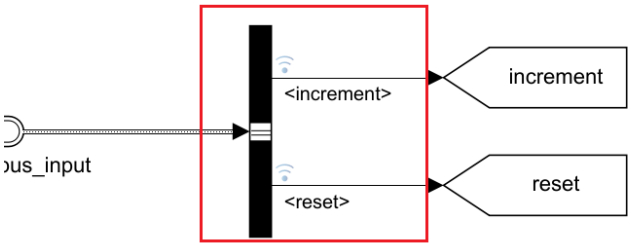


For Each 子系统的内部信号记录

自 matlab2016b 开始，支持 For Each 子系统内部的信号记录。For Each 子系统内部的每个信号包含多组数据，对应每次循环的结果，被记录的内部信号的命名方式采取 信号名_1 信号名_2 的方式，如果信号本身是矢量信号或者矩阵信号，则再增加 1 个或者两个下标，如 信号名_1_1, 信号名_1_2, 或者信号名_1_1_1, 信号名_1_2_1。



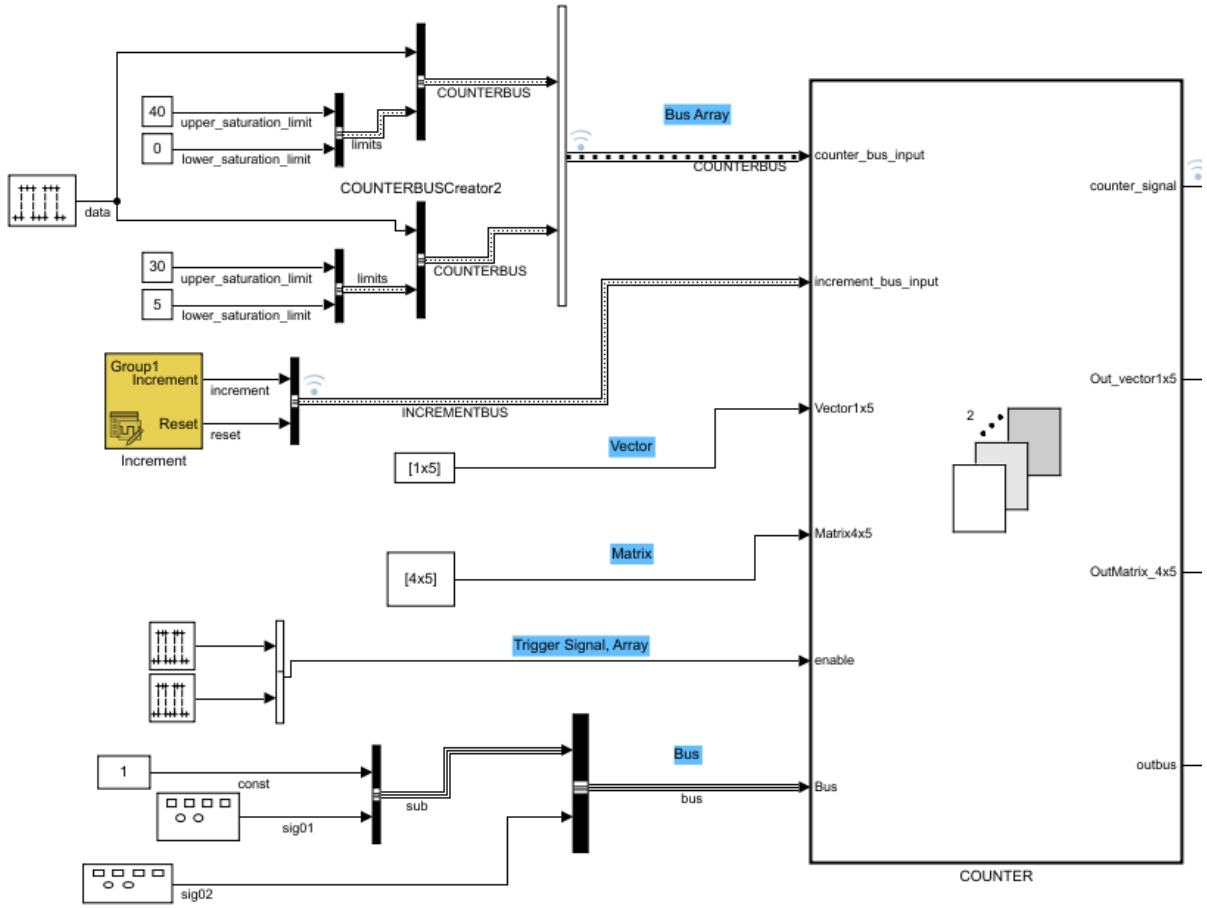
内部信号显示的名称



名称	值
data_1	131x1 double
data_2	131x1 double
increment_1	131x1 double
increment_2	131x1 double
reset_1	131x1 double
reset_2	131x1 double
t	131x1 double

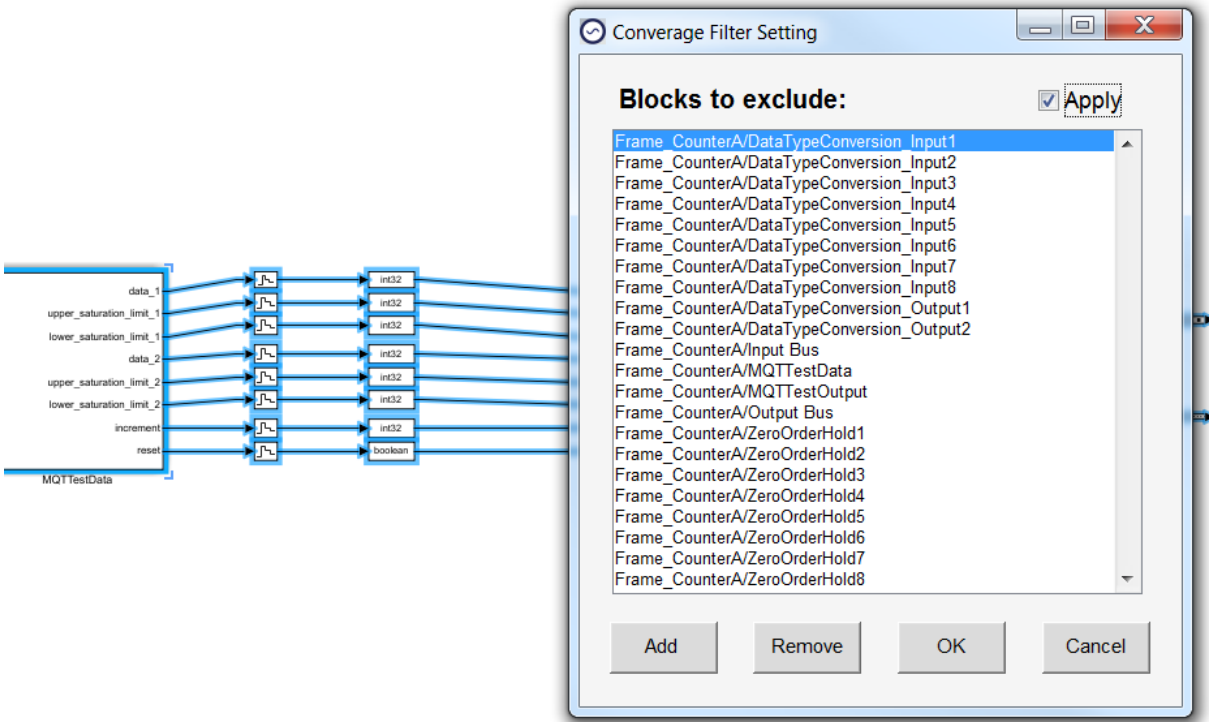
复杂信号的自动识别

在标量信号、矢量信号及总线信号之外，现在也支持矩阵信号及总线数组(bus array)信号,总线数组信号中的信号名(bus element name)允许重名。下图所示的所有输入信号均支持。



结构覆盖度

MQTester 以前的版本在 matlab2016b 以前只能对整个被测模型进行覆盖度仿真，否则无法在报告中显示覆盖度内容，新版本中在添加测试对象时可以选择两个子系统，第一个子系统是被测对象和辅助模块在一起的子系统，第二个子系统是被测的功能子系统，MQTester 会缺省选择被测功能子系统为覆盖度测试对象，建立 SiL 时也只为被测功能子系统产生代码。从 Matlab2016b 开始，matlab 限制只能选择对全部系统(主模型和参考模型)，或者只是对主模型或者参考模型进行覆盖度测试，无法和以前版本一样，在选择主模型中的子系统的同时还能选择参考系统，将 MQTester 的参数 execution.ModelRefDataLogging 设置为 1，设置参考模型模块按照参考模型内部的设置进行信号记录，这样就可以只选择设置主模型中的子系统进行覆盖度测试，不需要选择整个系统。另外，MQTester 现在提供覆盖度过滤命令，也可以在选择对全部系统进行覆盖度测试后，选择要过滤掉的模块。选择后会出现如下图对话框，选择要过滤的模块后添加到过滤列表中(可以同时选择多个模块)，再选择应用，则过滤列表生效。



覆盖度报告中列出所使用过滤文件及被过滤掉的模块。

Coverage Options

Analyzed model	Frame_CounterA
Logic block short circuiting	off
Filter filename	Frame_CounterA_covfilter.cvf

Objects Filtered from Coverage Analysis

#	Model Object	Rationale
	SubSystem block "Input Bus"	filtered block
	SubSystem block "MQTTestData"	filtered block
	SubSystem block "MQTTestOutput"	filtered block
	SubSystem block "Output Bus"	filtered block
	DataTypeConversion block "DataTypeConversion_Input1"	filtered block
	DataTypeConversion block "DataTypeConversion_Input2"	filtered block

在 matlab2015b 开始，覆盖度测试中增加执行度指标，缺省目标值定为 100%。选择命令 需求分析->配置需求选项 后，在 project 目录下产生的 m 文件 MQTester_DefineRCTPrjOption.m 中可以修改各项指标的目标值。

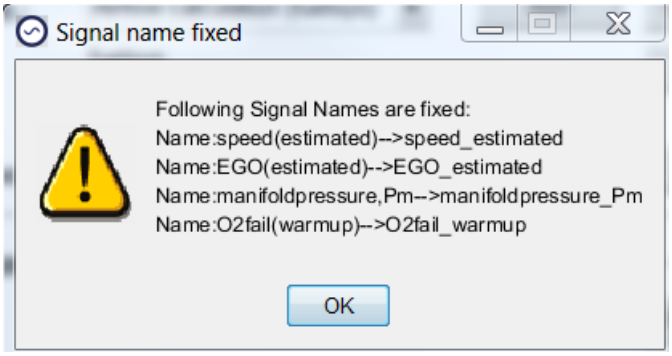
Tool	Metric	Percentage	Target	Total	Reached	Unreached
Model Coverage	Condition	61.9%	80%	42	26	16
	Decision	58.4%	80%	125	73	52
	Execution	91.7%	100%	265	243	22
	Lookup	75.0%	100%	8	6	2
	MCDC	27.3%	80%	11	3	8

CSV 格式文件

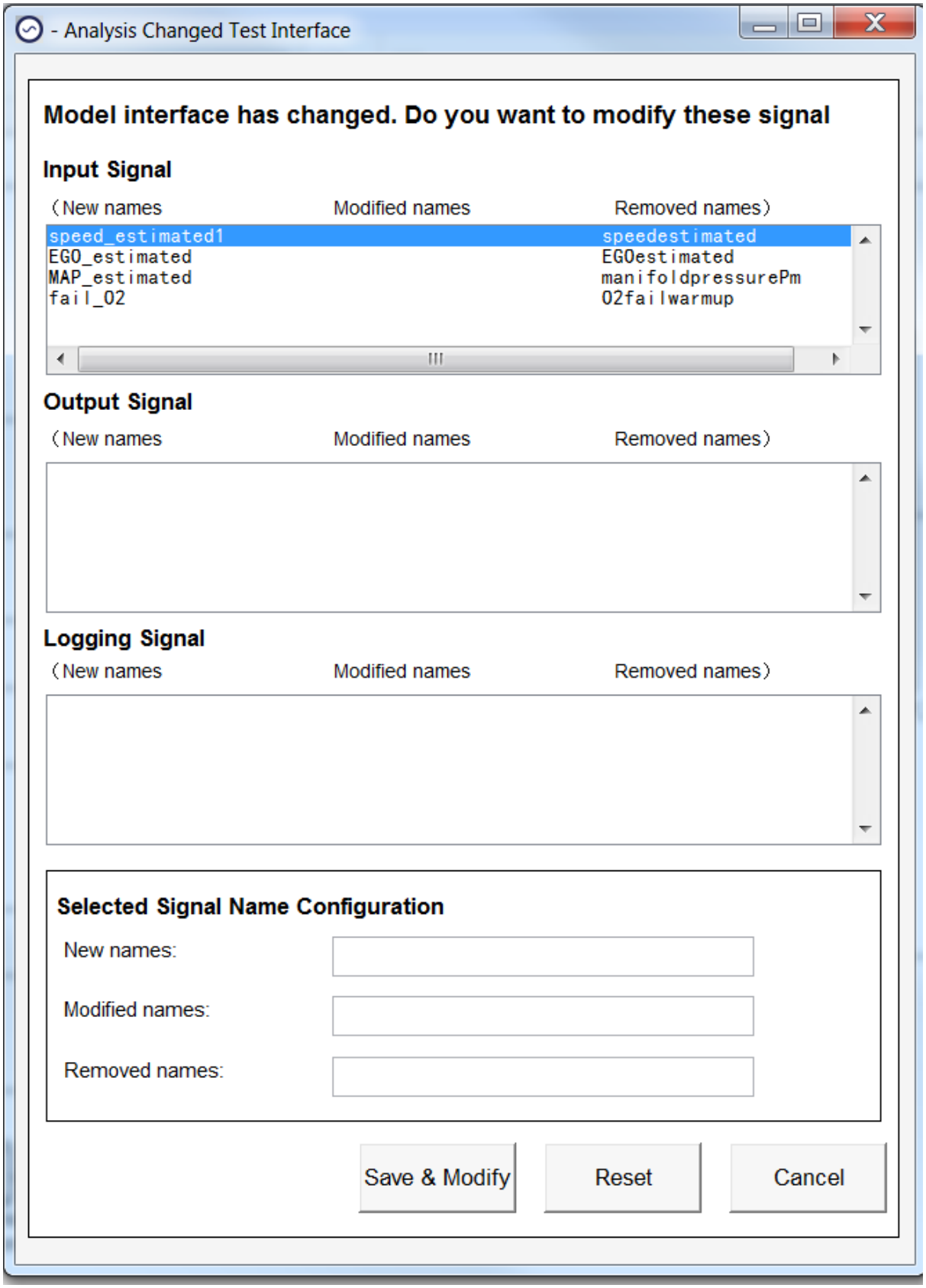
在没有安装Excel软件情况下，可以使用CSV格式的文件编写测试用例及需求文档，将MQTester_GUIcommonProper中的tools.useExcelorCsv设置为0， 则MQTester生成csv格式的测试用例文件，导入需求文档时也优先选择csv格式的文件。

接口自动修正

接口信号或者信号名中包含非 ASCII 字符时，现在 MQTester 可以自动修改为符合 ASCII 要求的名称，不再需要在原模型中人工修改，MQTeter 会提示对哪些信号名进行了修改，如果希望原模型和测试环境中的信号名保持一致，可以人工修改原模型，然后再产生 testbed。



该版本提供新的命令(文件->模型接口分析及变更)，检查接口信号名称的改变，检测到改变后可以自动对测试用例、测试环境及评估函数等进行修改。



Speedgoat HIL 仿真

使用 Speedgoat 硬件进行硬件在环仿真时，可以将使用 MQTester 建立的测试环境及测试用例直接应用到 HiL 中。为此，提供了三个下拉菜单命令:

仿真配置->生成 HiL(SG)环境

为测试对象生成能够在 speedgoat 硬件在环设备上运行的测试环境(该测试环境模型与 Simulink 下的有少许不同，因此使用单独的命令)

仿真配置->建立 HiL(SG)代码并下载

为使用上面的命令生成的测试环境模型建立在 speedgoat 硬件在环设备上运行的代码并下载到目标机

仿真配置->仿真当前用例 HiL(SG)

在 speedgoat 目标机运行当前测试用例(对于多个测试用例可以在批处理窗口中勾选相应选项进行批处理)。

以上菜单项自 MQTester2.6 开始提供。

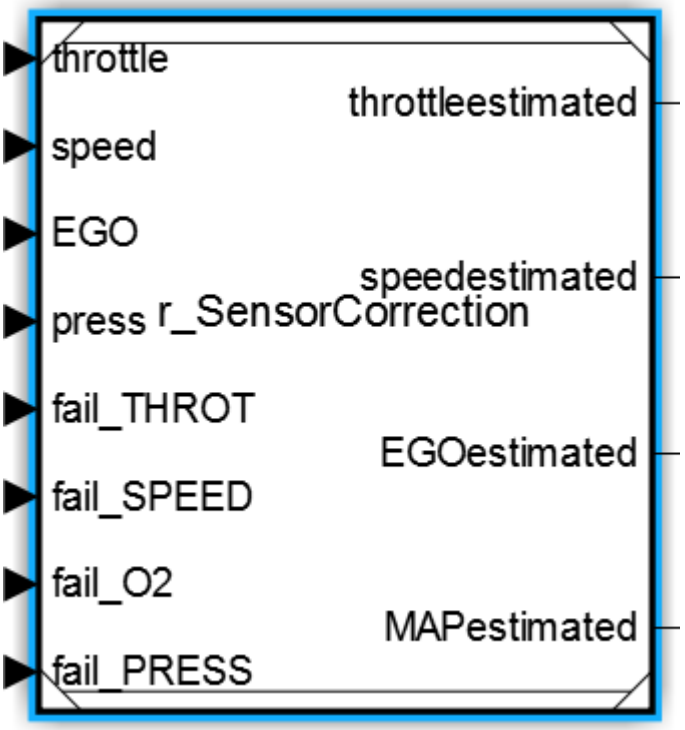
TL 中重用函数模型

TargetLink 模型中，如果包含 TL Function block 的子系统被设置为可重用函数子系统(设置 TL_Function block 的 make function reusable 属性)，在该子系统和 Library 的链接被断开时，无法生成代码，从而无法进行 SiL 仿真，MQTester 现在会保持子系统和 Library 的链接不变，从而保证代码的生成。

参考模型(TL)

提供三种在 Testbed 中处理 TargetLink 模型中的参考模型的方法，参数 execution.tldisablerefmodels2testbed 设置为 0，则在 testbed 中参考模型还是参考模型，设置为 1，则参考模型被拷贝到一个子系统中; 参数 execution.tldisablerefmodels4testbed 设置为 1，则 testbed 建立的最后步骤中将参考模型变为 disabled model reference, 与将参考模型转换为子系统功能相同，区别在于 disabled model reference 的外观和子系统不同，如下图，注意方块的四个角及上下两边的线。

建议将两个变量的值均设为 0，在产生代码有困难时，可以修改其中一个的值为 1.



TargetLink 参考模型中必需包含 TL Function block，在生成代码时，参考模型中的 TL function block 的代码覆盖等级被设置为和主模型一样。

TL 模型中需要的外部代码及 Include DD

TL 模型中包含的外部 c 代码及 h 文件，如果和模型在同一目录下，会被拷贝到 testbed 的目录下，包含的 include DD 会同样被拷贝到 testbed 环境，并被同样链接到 testbed 的 DD 中。这会减少生成代码时的错误。

TL 模型 SiL 仿真时复杂内部信号的记录

TL 模型 SiL 仿真中记录的内部信号需要根据 lsb 及 offset 计算其实际值，目前 MQTester 支持标量，矢量，各种结构的总线信号及矩阵信号。

TCSD 函数及宏

对内部(local)信号的评估

在编辑测试用例时，可以为内部信号(local 信号)定义期望值或者使用在期望值函数中，MQTester 会自动进行期望值比较或者生成评估函数。

TCSD 语法扩展

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

[+1s]

In=set(1);

In=set(2,0.2);//目标值 2， 时间偏移 0.2 秒

[+1s]

假设步长为 0.1s 则 In 的值为[1 1 2 2 2 2 2 2 2 2]。

宏定义

增加三个宏 $AllInSig$ $AllOutSig$ $AllLocSig$，代表所有的输入信号 输出信号及内部信号，可以使用在 TCSD 的 action 部分，用于在任意时刻为所有同类信号统一赋值。例外的信号可以在使用宏后单独赋值，如下所示语句，执行结果是除了 signal1 之外的所有输入信号的值为 1，signal1 的值为 2.

```
$AllInSig$=1;

signal1=2;
```

期望值函数语法扩展

期望值表达式函数 expExpress 扩展为里面的表达式可以使用任意的 matlab 函数，如下例中，输出信号 outMin 的期望值被设置为三个输入信号的最小值，表达式中使用了 matlab 函数 min().

```
[+1.0s]

a = sin(2,3);

b=sin(2,3,50,0.5);

c=ramp(5);

//期望值函数，持续 20 秒

outMin=expExpress('outMin==min(min(a,b),c)',20);

[+10.0s]

a=ramp(4);

[+5s]
```

跨时间段定义信号

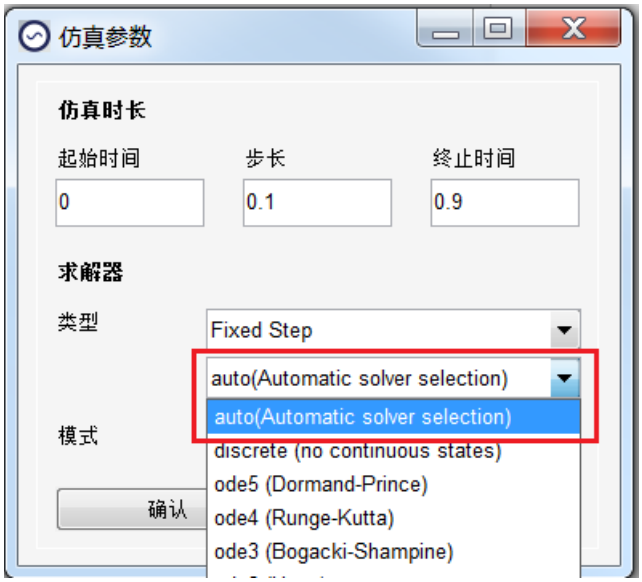
TCSD 中所用的函数中的持续时间参数对信号的影响不再受当前时间段长度的影响，可以持续到全部时间段的最后(如果持续时间大于等于后面所有时间段的时间长度总和)。例如上面的函数 outMin=expExpress('outMin==min(min(a,b),c)',20);中的持续时间是 20 秒，但是当前时间段是 10 秒，所以直到整个测试序列结束，outMin 的期望值都是被定义为 a b c 三个输入信号的最小值，实际持续时间是 15s。这样，对于规律变化的信号，在整个测试序列中只需定义一次。

求解器设置

在进行 MiL 仿真时，MQTester 设置 testbed 的求解器为固定步长求解器，但是有时需要对物理模型进行仿真，可能需要变步长求解器，为此，现在提供了参数 execution.useOwnSolverSetting，设置为 1 时 MQTester 不设置 testbed 的求解器，使用 testbed 的自身设置，设置为 0 时，MQTester 在仿真前设置求解器为固定步长求解器。

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。

在测试用例的仿真参数对话框中，求解器增加了FixedstepAuto选项。



导入测试用例时的缺省时间步长

在导入测试用例时 MQTester 以前给出缺省时间步长是 0.1 秒，现在 MQTester 检测测试模型中的所有时间步长，如果测试模型中的最小步长小于 0.1 秒，则取此最小值，否则取 0.1 秒作为推荐时间步长，测试中的最小步长也显示在窗口中。