**MIL测试流程**

[一、 模型实现 1](#_Toc25880)

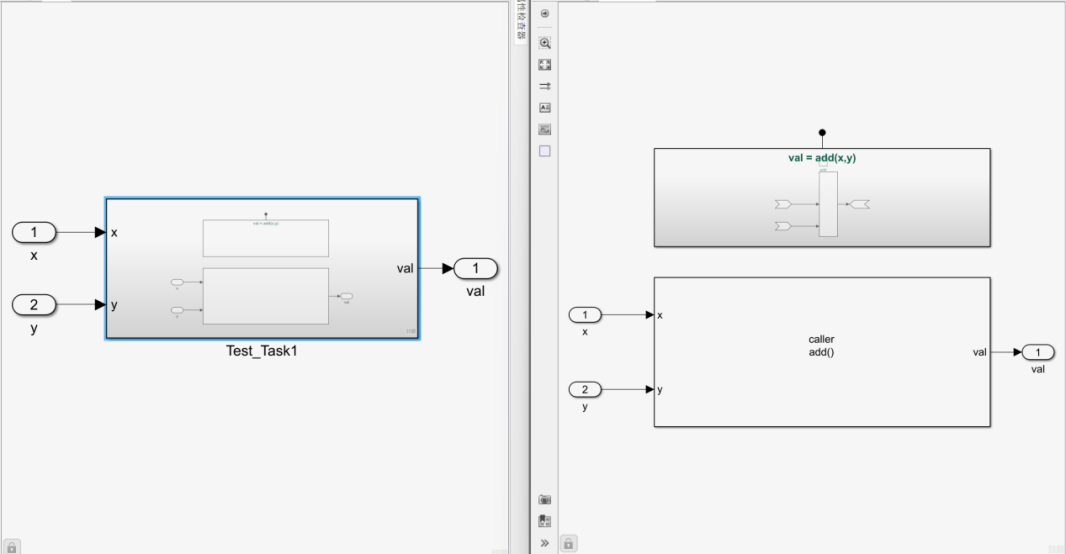
[二、 MIL测试方法一 1](#_Toc29468)

[三、 MIL测试方法二 1](#_Toc23540)

[四、 测试过程分析 1](#_Toc7475)

# **模型实现**

## **模型图片**



## **模型实现公式： val = x + y 的计算逻辑**

# **MIL测试方法1**

## **1、**右键点击模型

## **2、**找到测试框架

## **3、**为Model创建一个Test

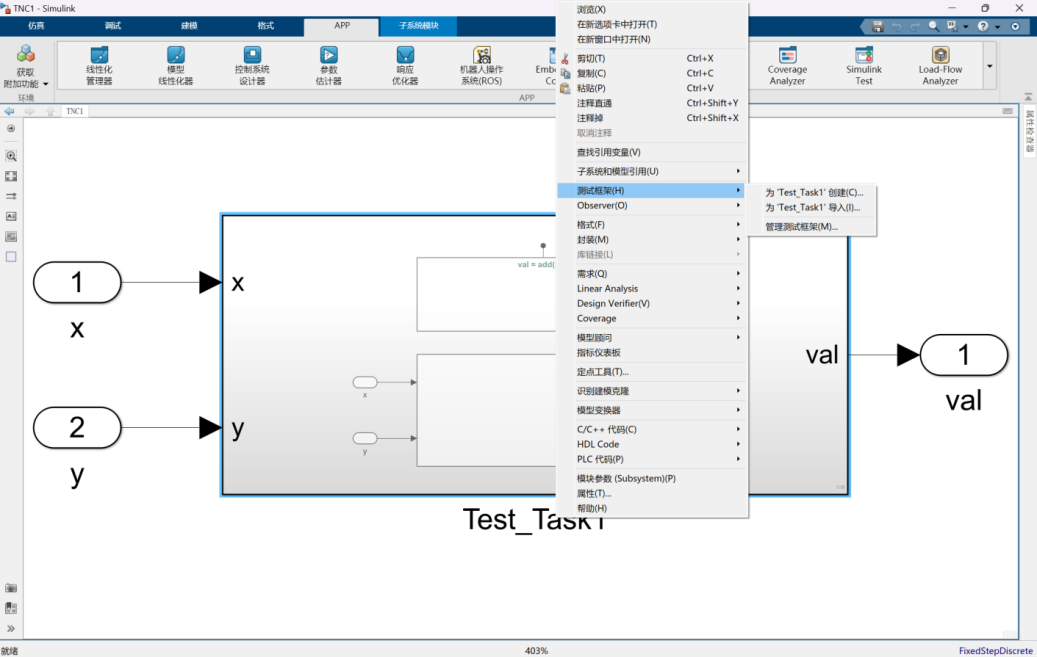
## **4、**在基本属性中确定受测组件的名字和存放路径

## **5、**源和信宿中选择：signal builder

## **6、**《使用以下项生成函数调用信号》选择None

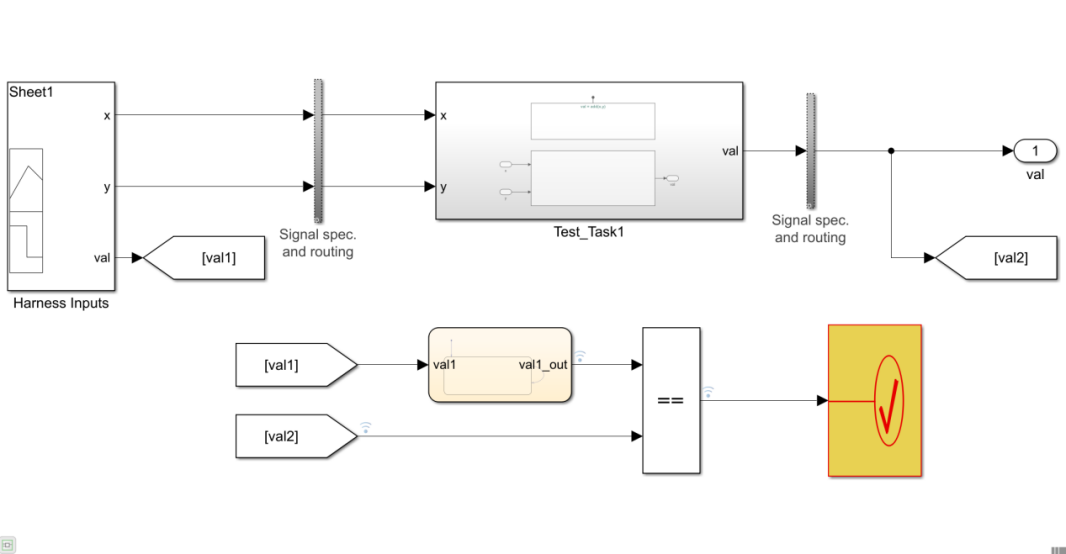
## **7、**确定

**过程如下图↓**



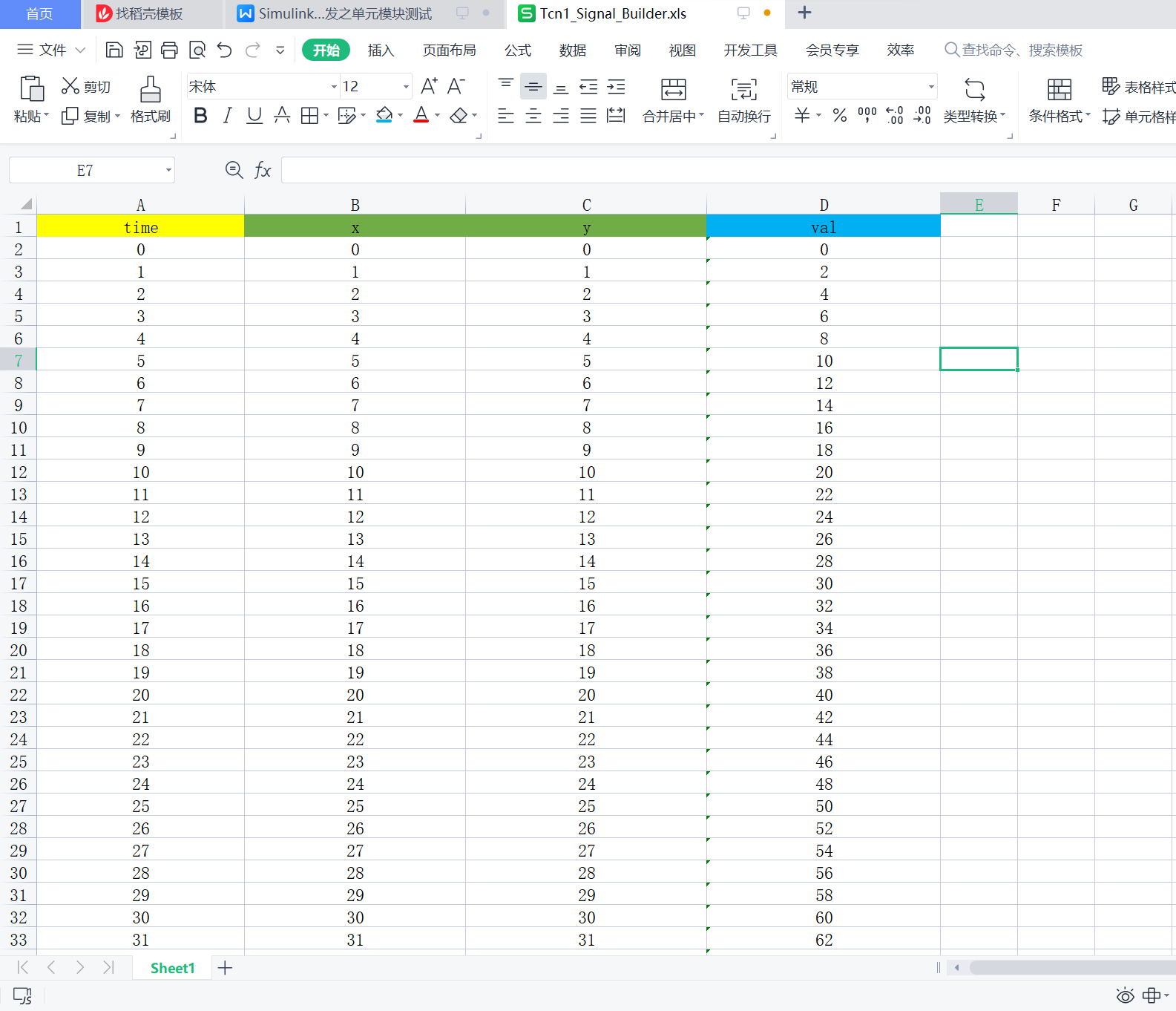


## 8、获得TCN1\_Harness2测试模型如下：



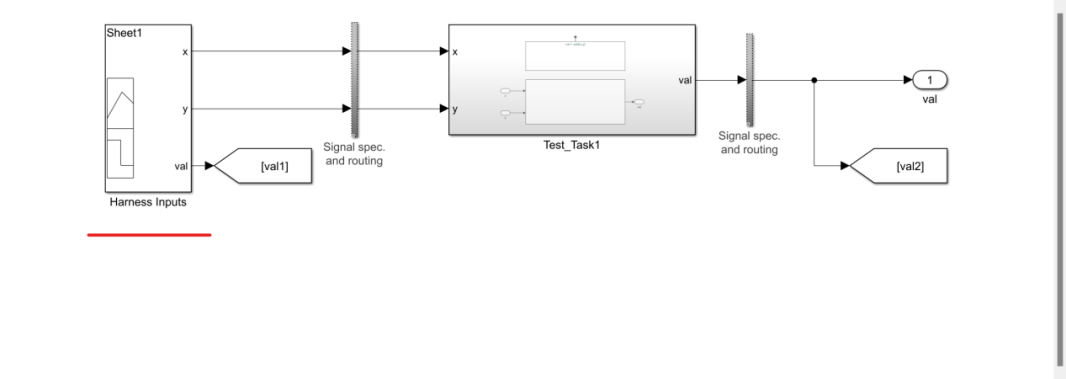
## 9、创建测试用例文档为Tcn1\_Signal\_Builder.xls,测试用例格式如下：

Time:为步长时间/X&y:为模型输入信号/Val:为模型输出信号

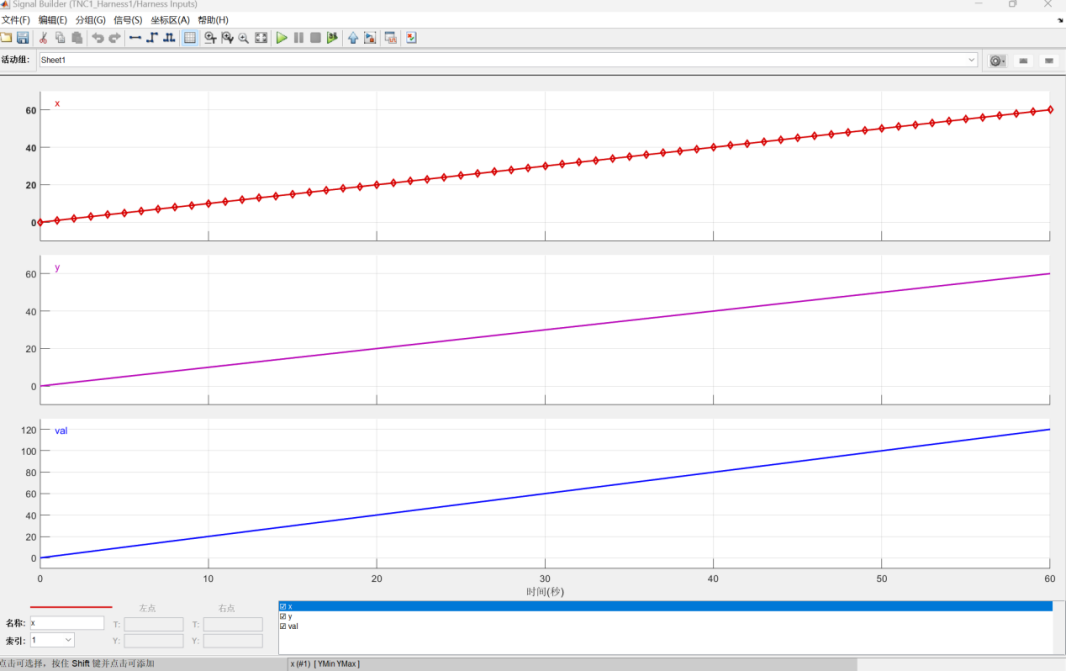


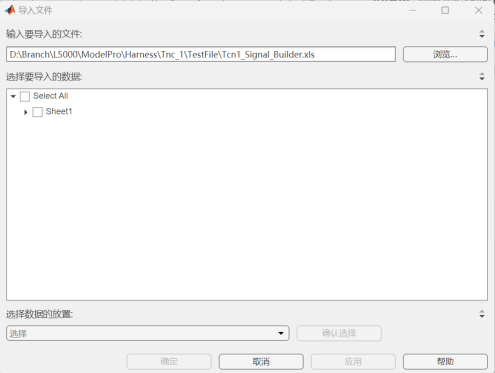
## **10、编辑测试用例：根据功能需求设定某一时刻输入x、y，输出val**

## **11、导入测试用例：点击Harness Input**



进入如下界面↓





**步骤：**点击文件 ⇒从文件导入 ⇒浏览进入：加载Tcn1\_Signal\_Builder.xls文件路径 ⇒勾选Select all ⇒选择数据数据的放置：替换现有数据集 ⇒应用 ⇒确定

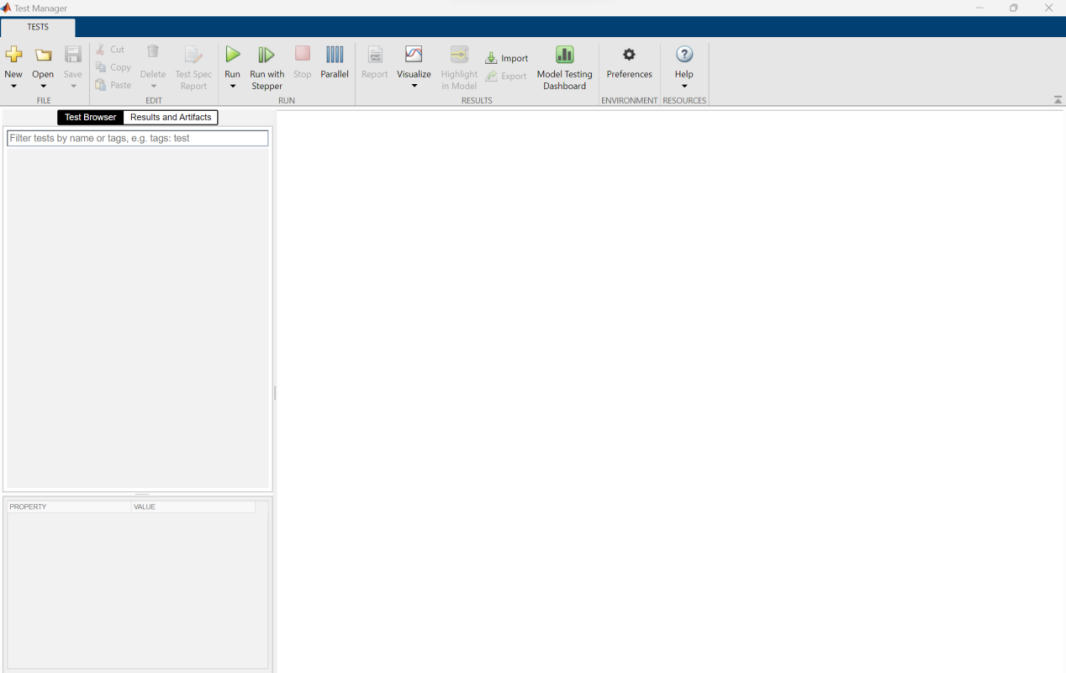
## **12、Run -> 分析逻辑分析仪中的逻辑时许是否满足功能需求和测试用例**

# **MIL测试方法2**

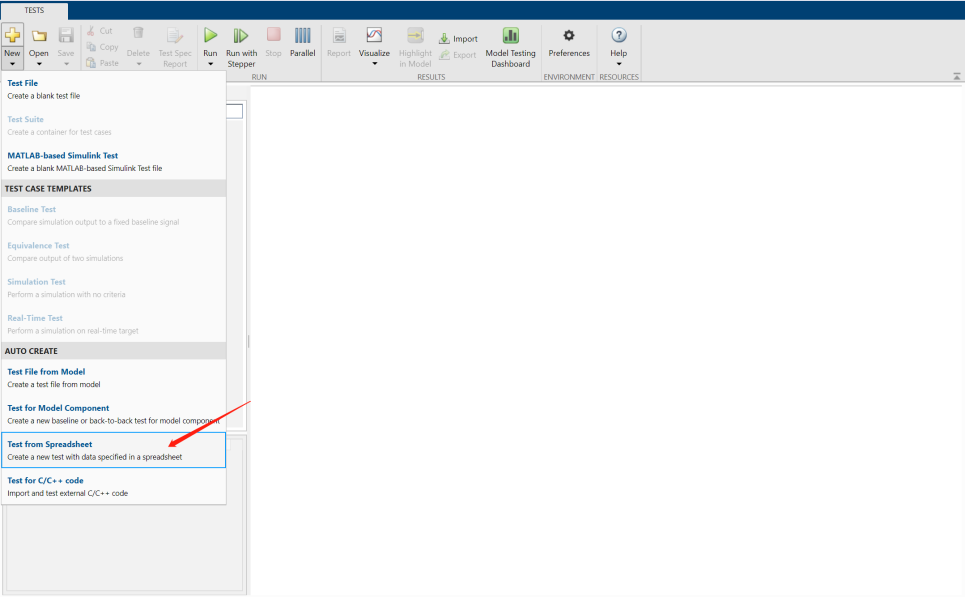
## 1、点击APP，寻找到Simulink Test,点击进入，出现如下工具栏↓



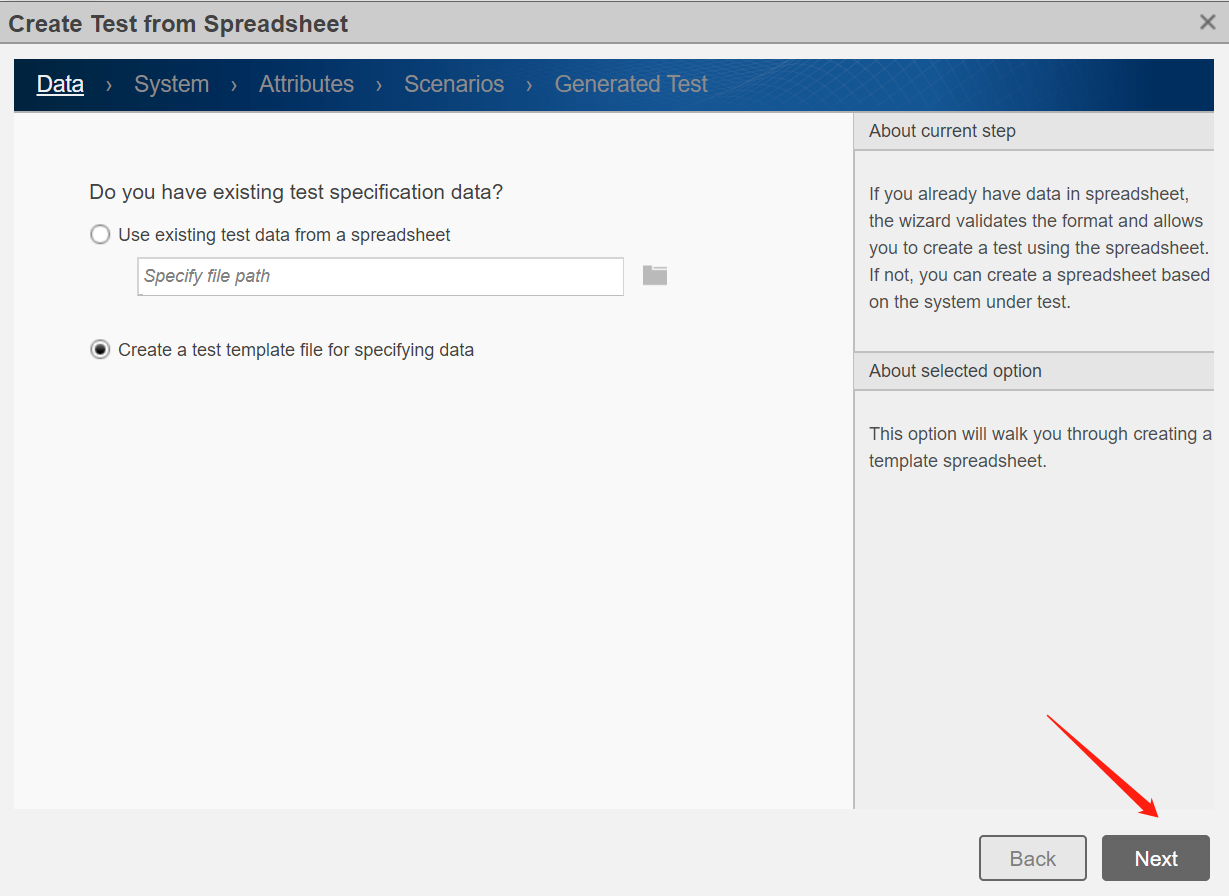
## **2、使用Simulink Test Manager ,点击进入，如下图↓**



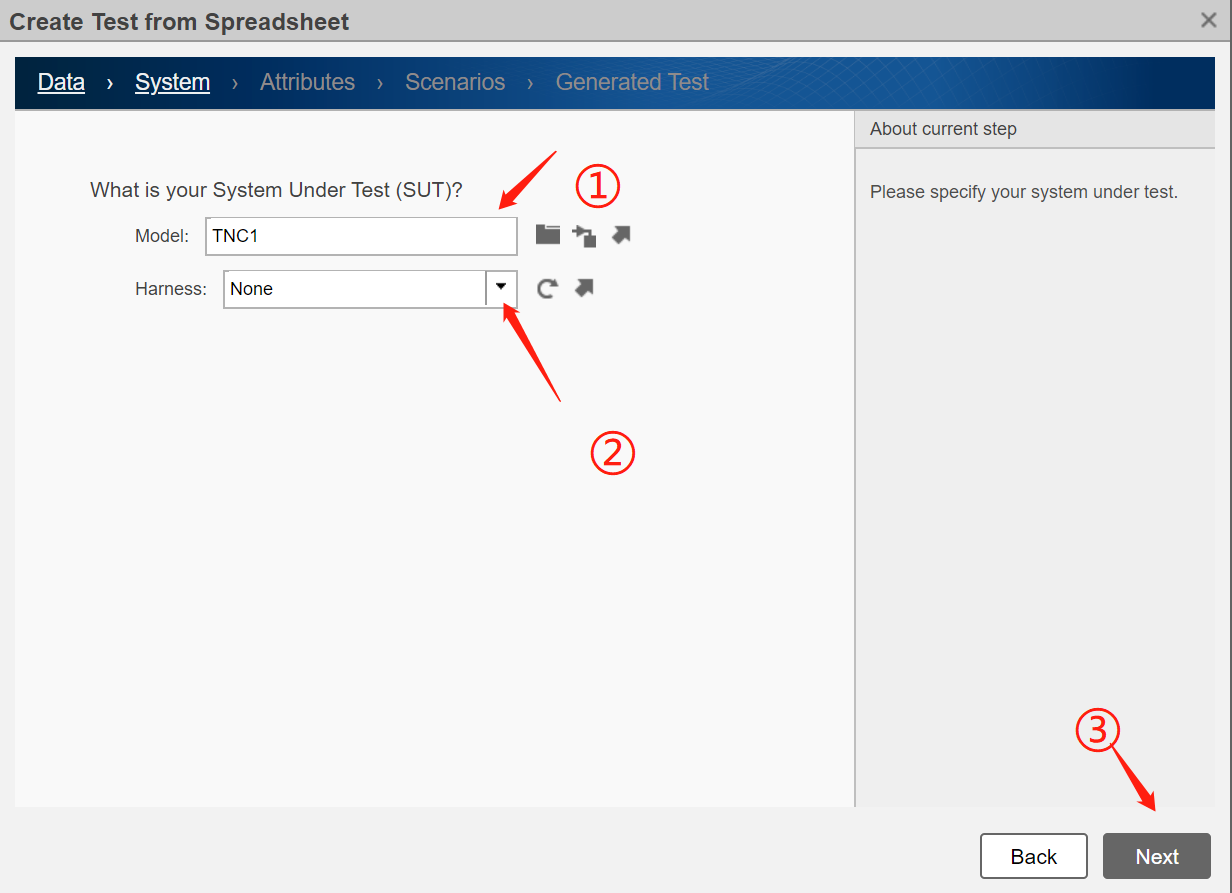
## **3、用电子表格中指定的数据创建一个新的测试：New -> Test from Spreadsheet**



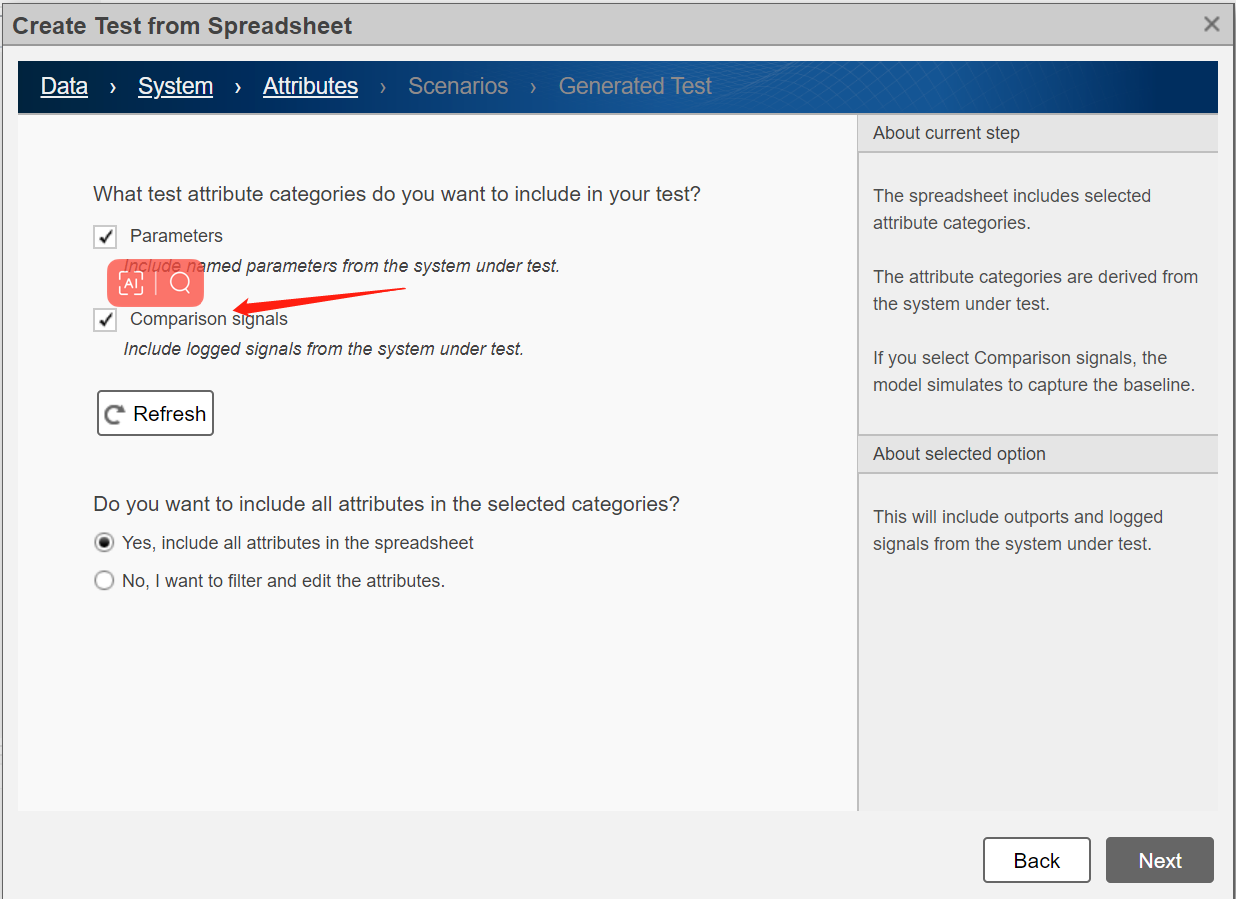
## **4、按照Data****⇒Systerm⇒Attributes⇒Scenarios⇒****Generated Test顺序添加**



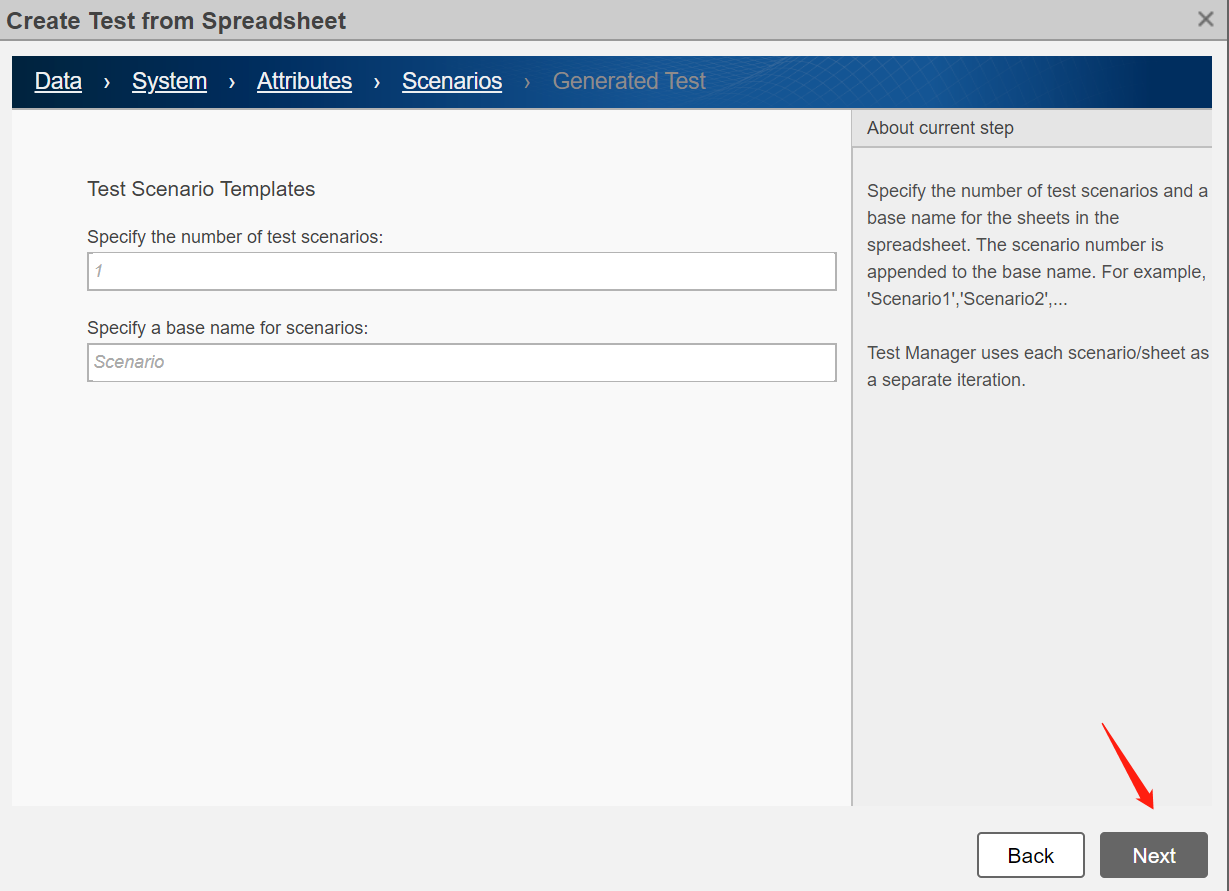
### 选择Model->选择Harness(也可不选，配置为None)->Next->



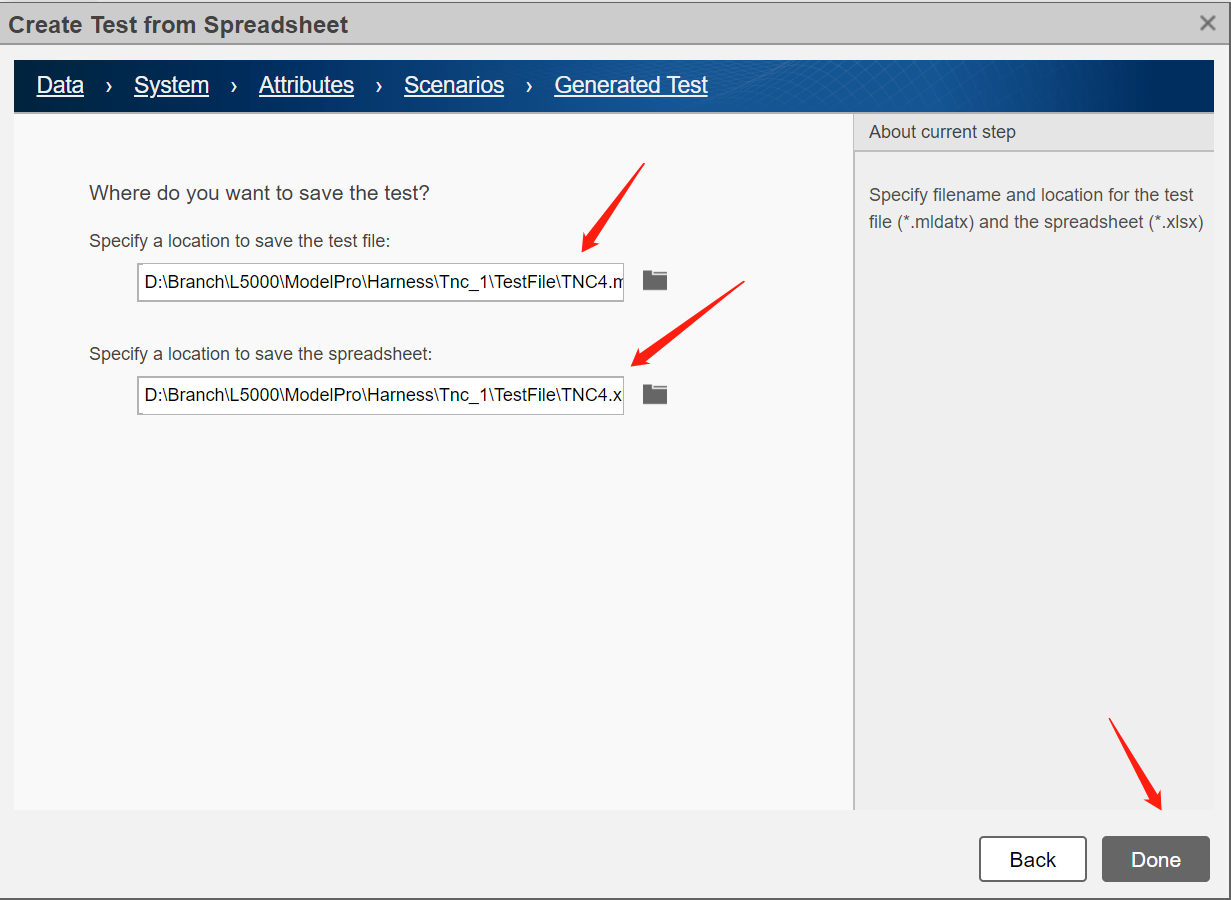
### 勾选Comparison Signals->Next



### **Next**

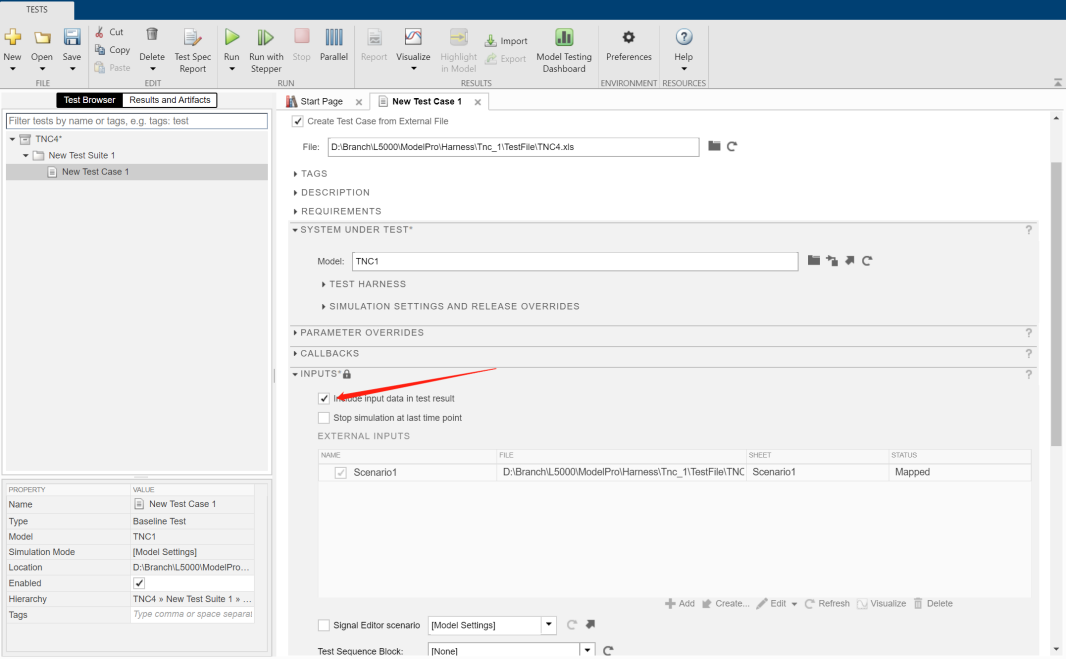


### **选择你的Test file 和 SpredatSheet -> Done**



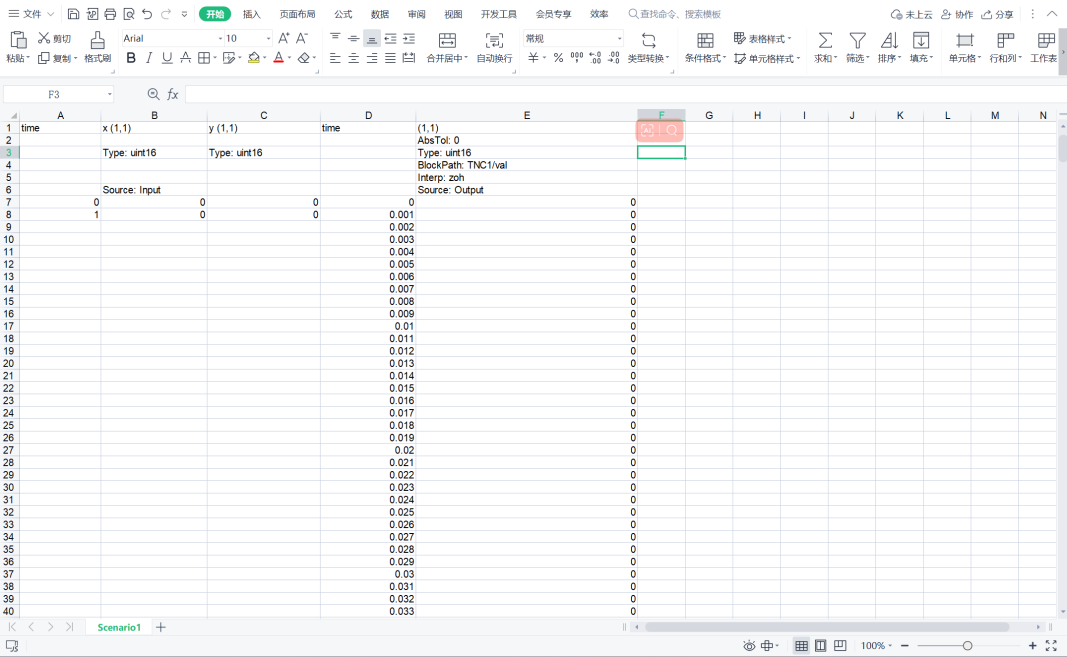
## **5、开始配置Test准备条件**

### **勾选INPUTS -> Include Input data test result**

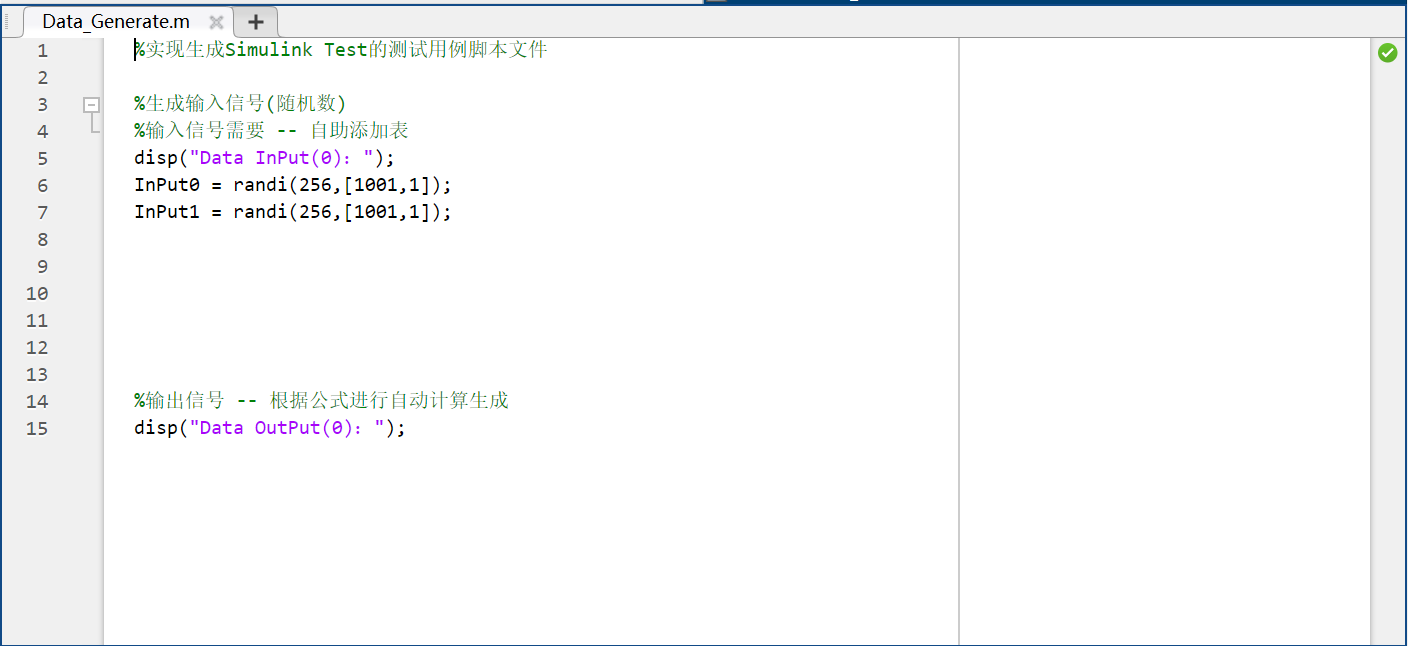


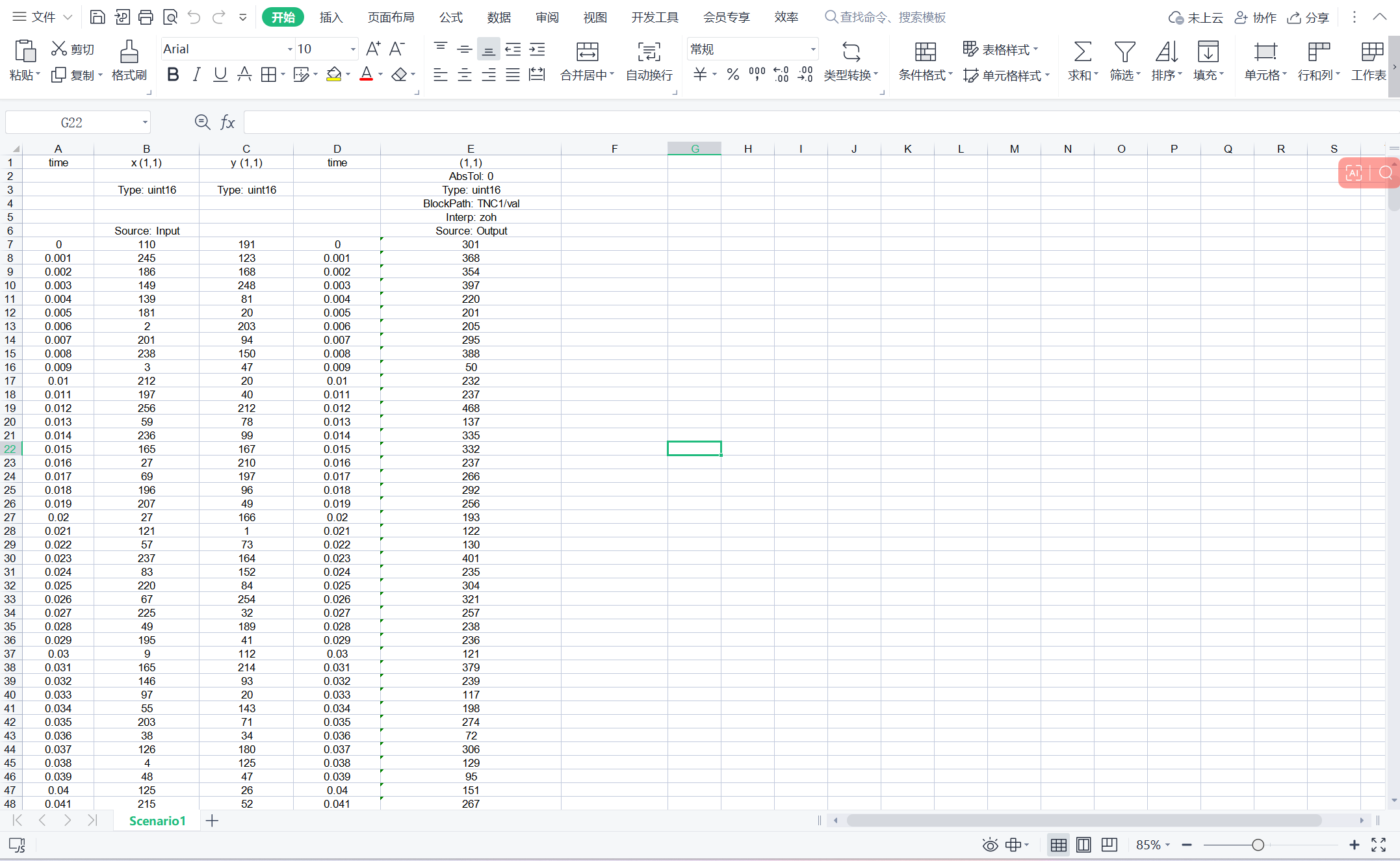
## **6、编辑测试用例(测试用例在创建时自动生成，文件路径参考第四步的”Generated Test”界面)**

### **测试用例初始情况：**



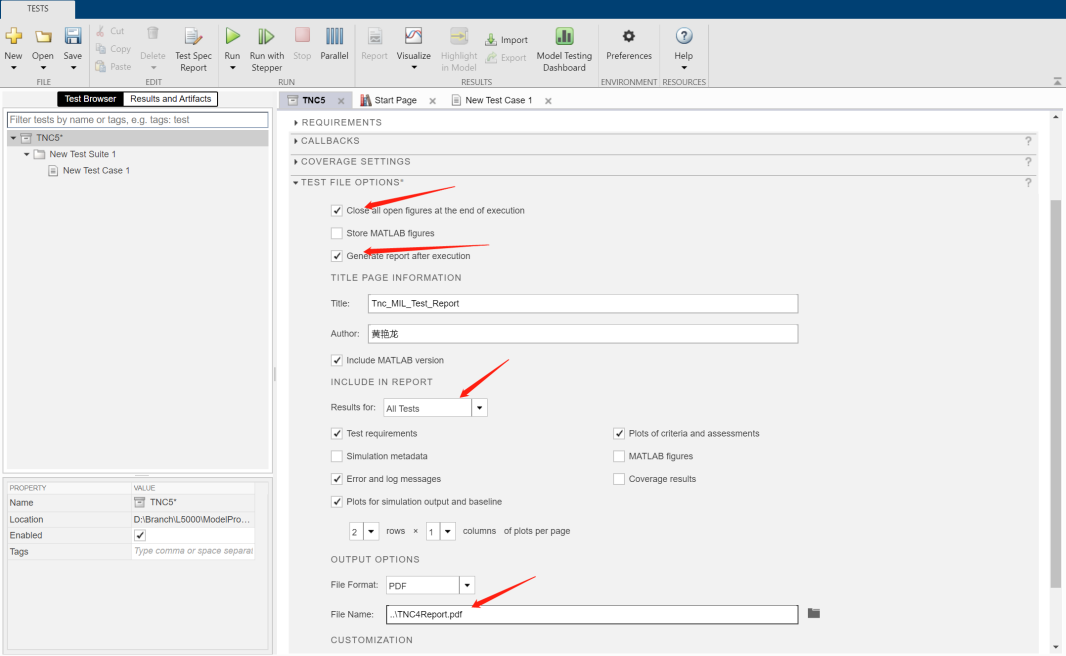
### **测试用例编辑完成状态：**可以通过脚本生成随机数据对模型公式进行验证



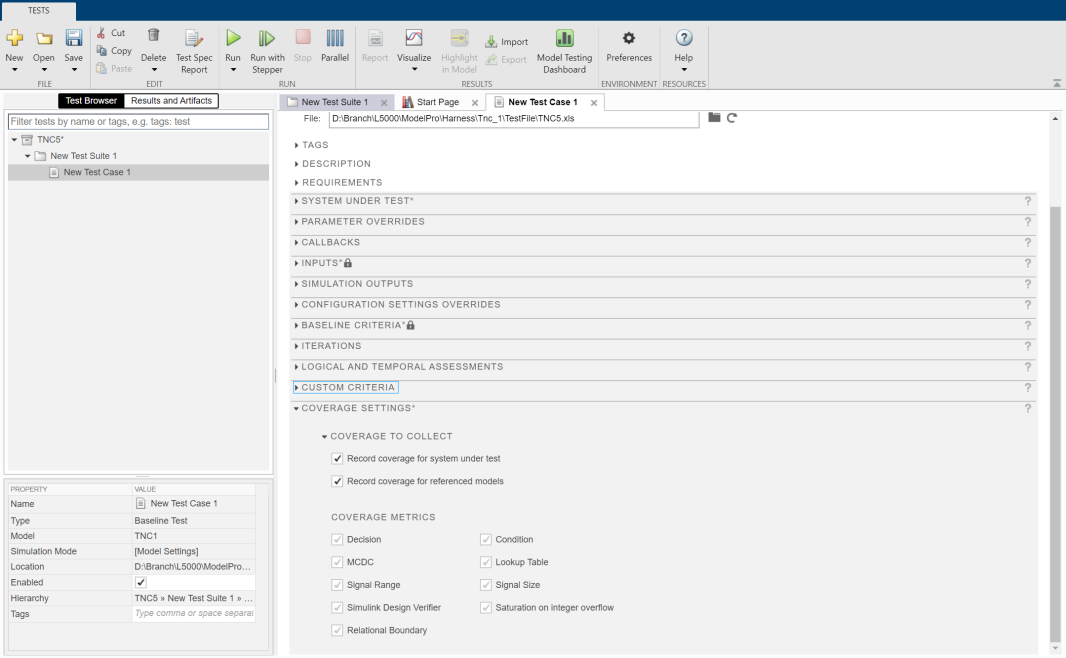


## **7、生成报告配置**

### **配置输入**

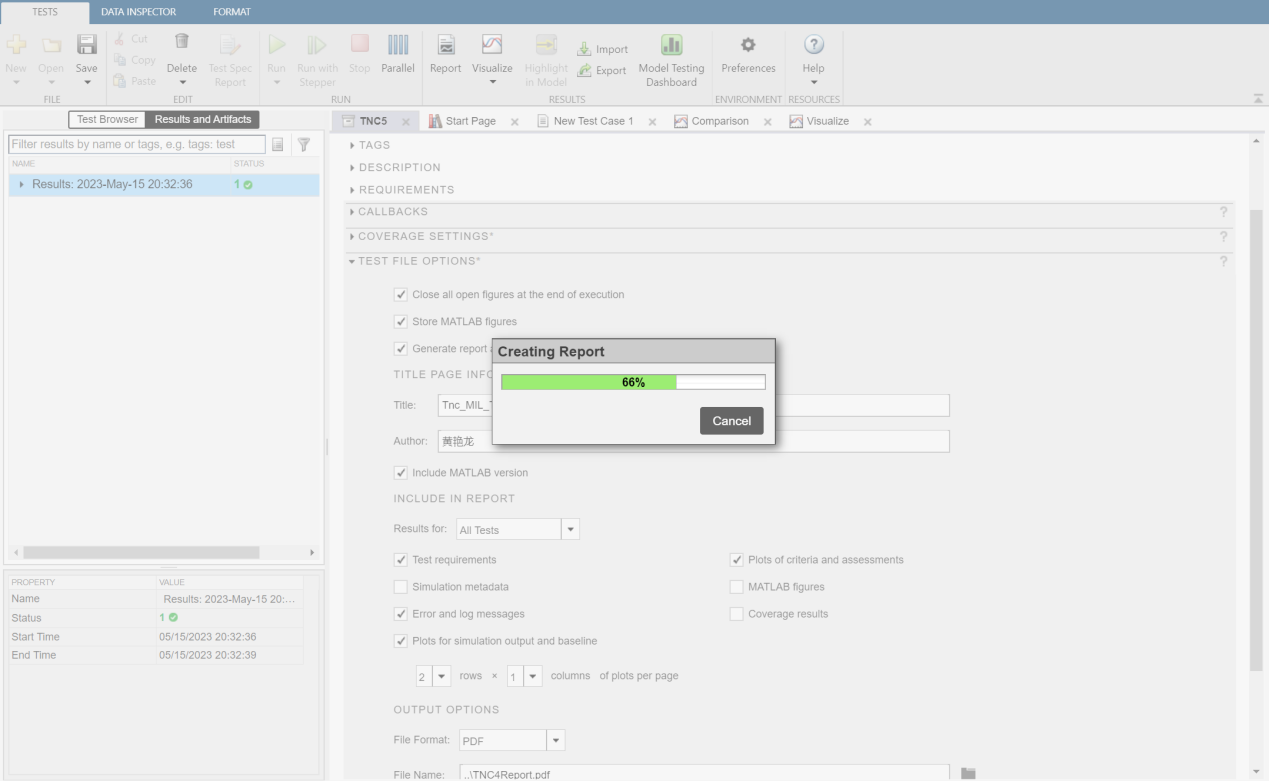


### **配置生成报告选项**



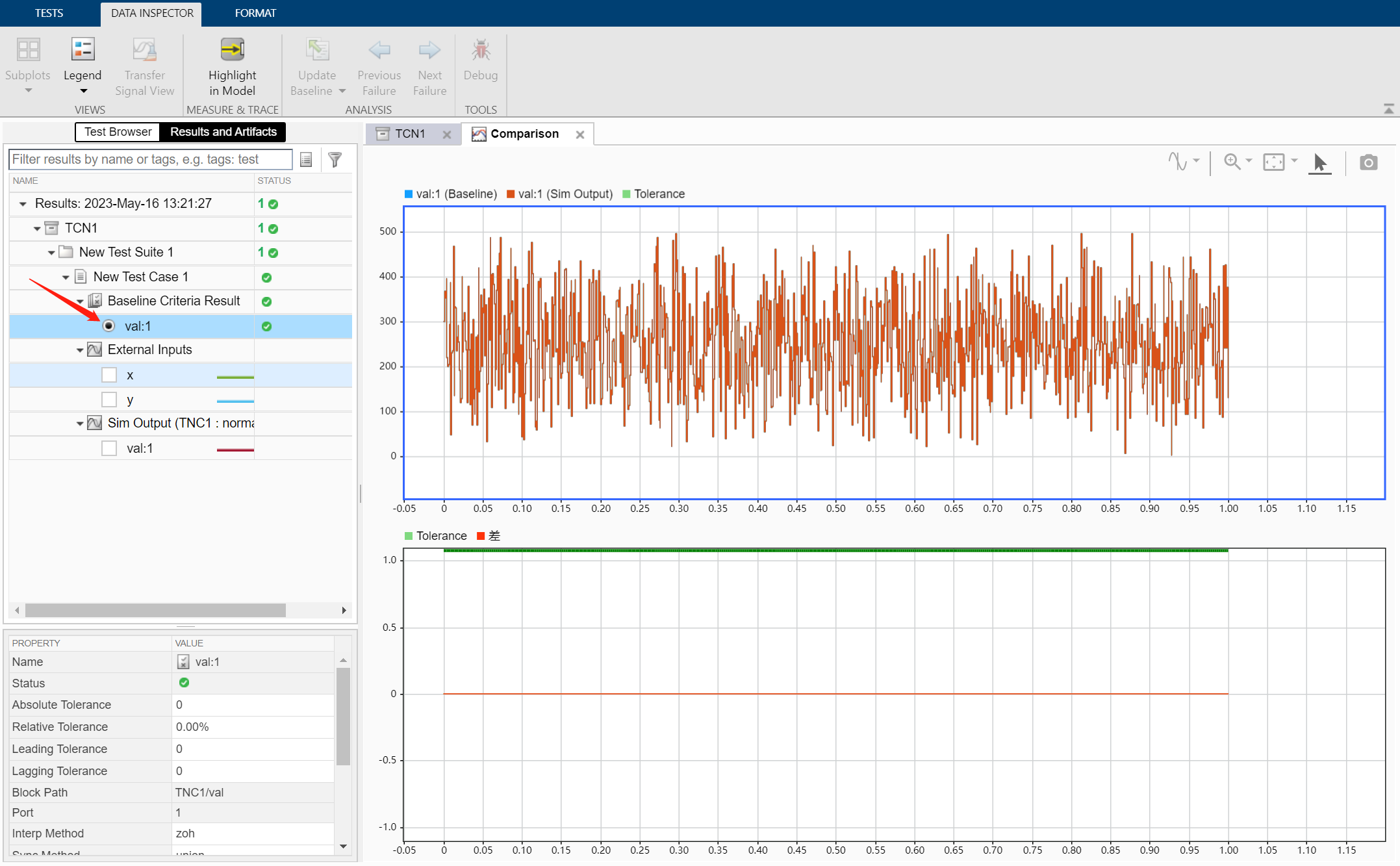
## **8、Run->**

### **点击Run按钮**

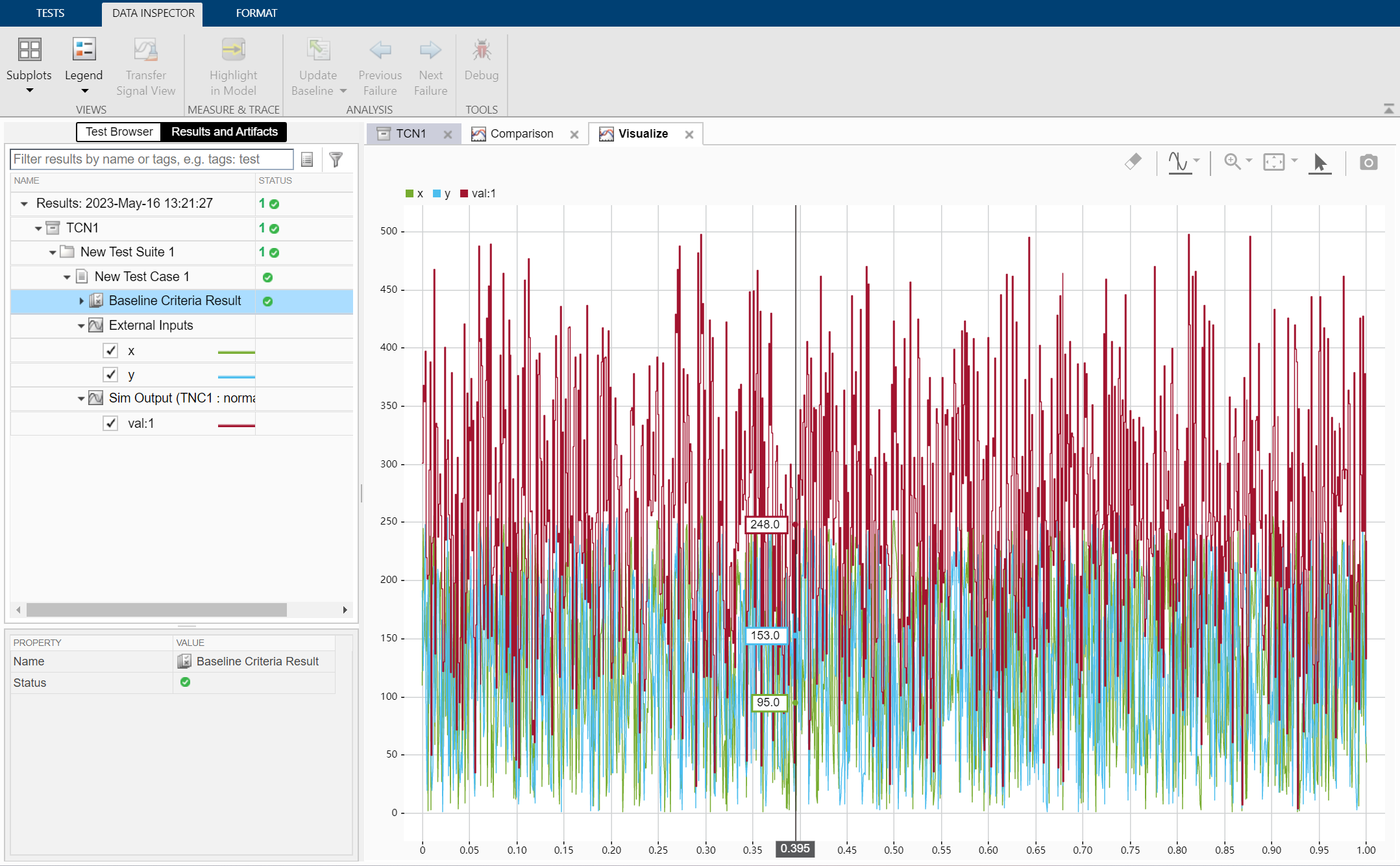


### **等待结果......**

# **测试过程分析**

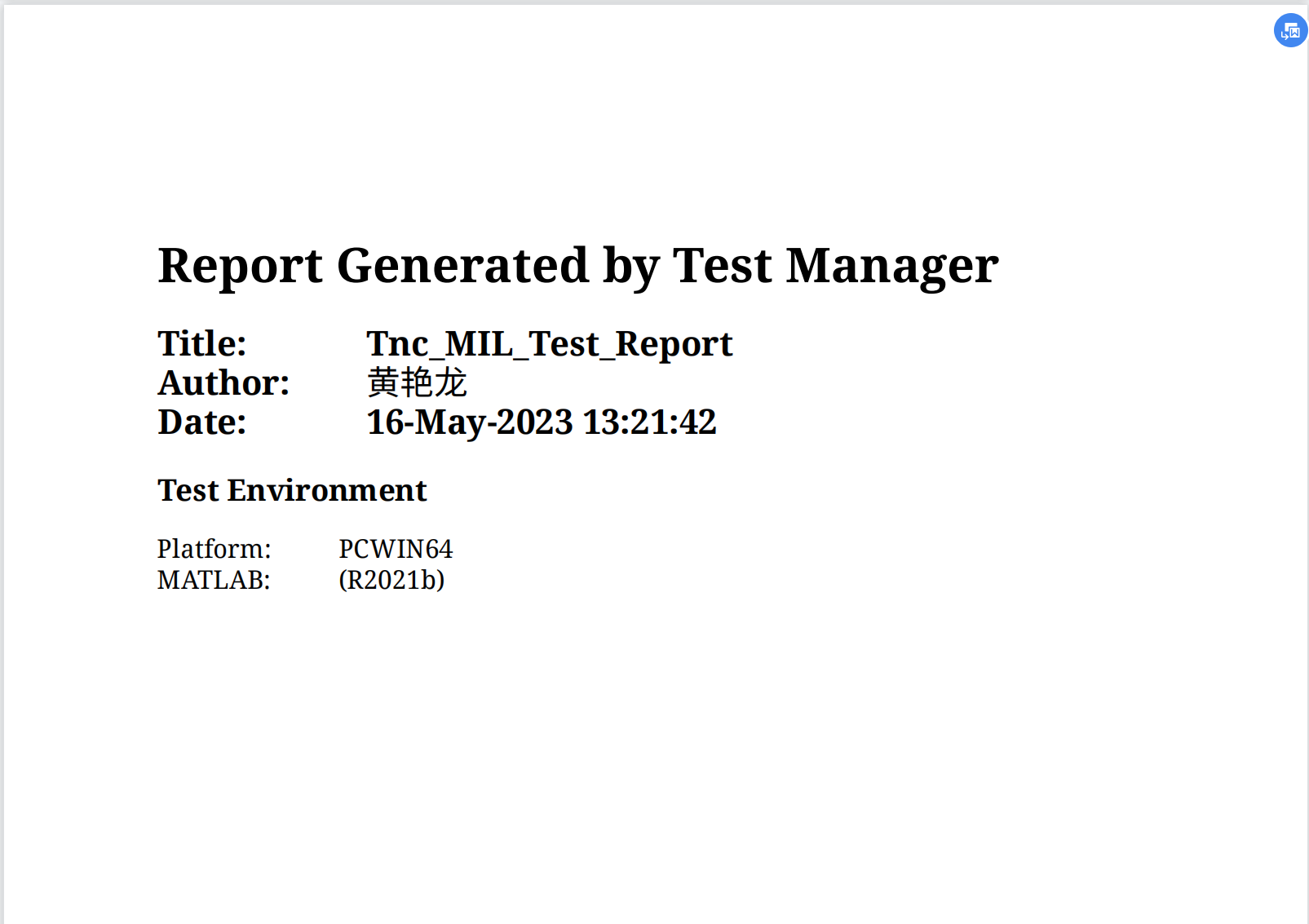


### 分析1：上图是输出结果的基线标准显示结果，从上面可以清楚看出计算公式val = x + y的输出结果val值的变化曲线，其与测试用例中的val的比较结果可以参考下面的曲线，结果显示差值一直保持在 = 0的水平线，可见测试用例中的每一时刻的输入x,y相加的结果都与模型得到的结果完全一致

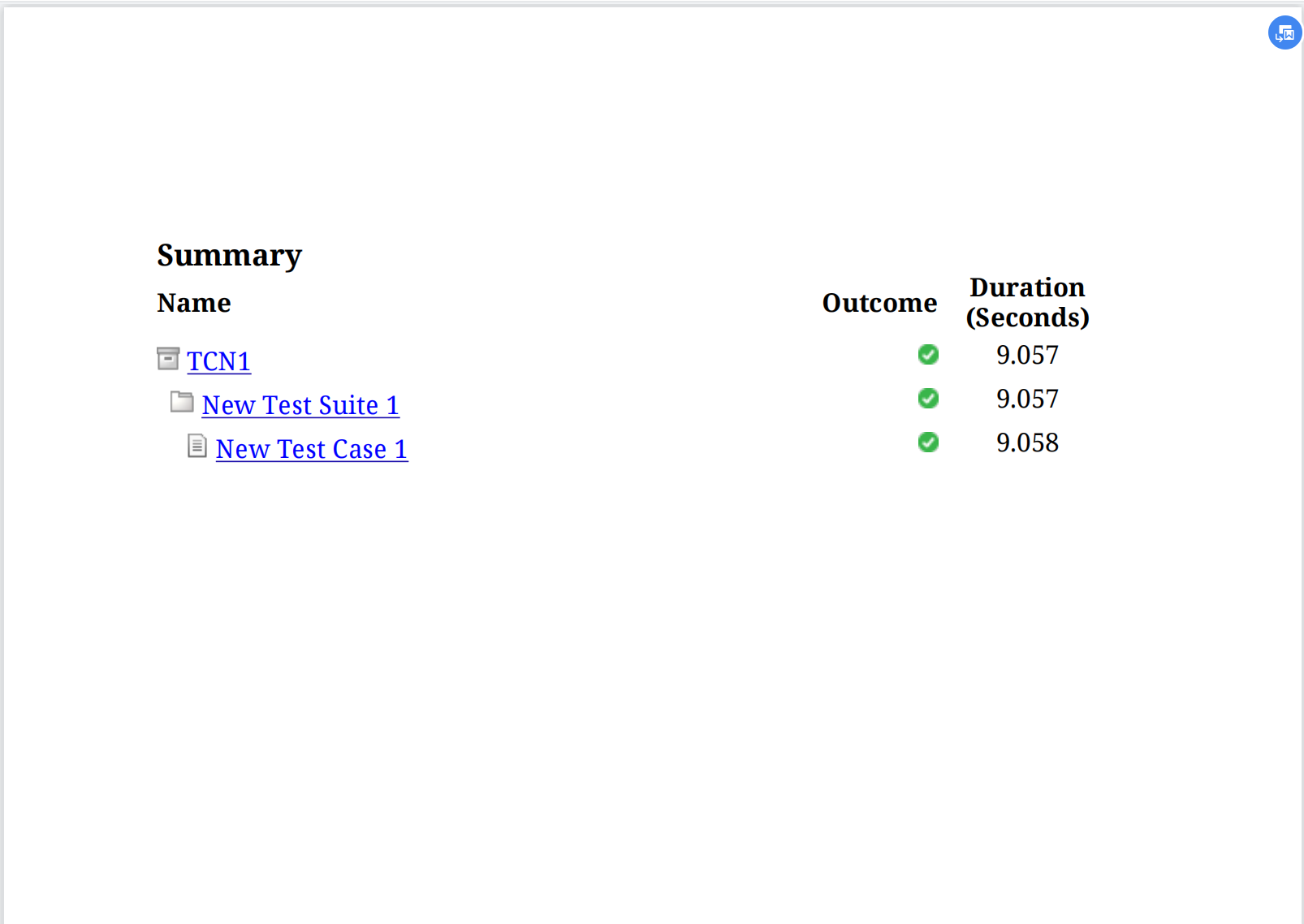


### 分析2：上图是外部输入和模拟输出的全局显示结果，通过上面左侧可以看到下，x,y,val的勾选控件，勾选后出现右面的曲线图，通过设置游标至某一处，可以看到：在time = 0.395s时，x = 95; y = 153; val = 248;可见模型对外部输入的计算结果和模拟输出的结果完全一直

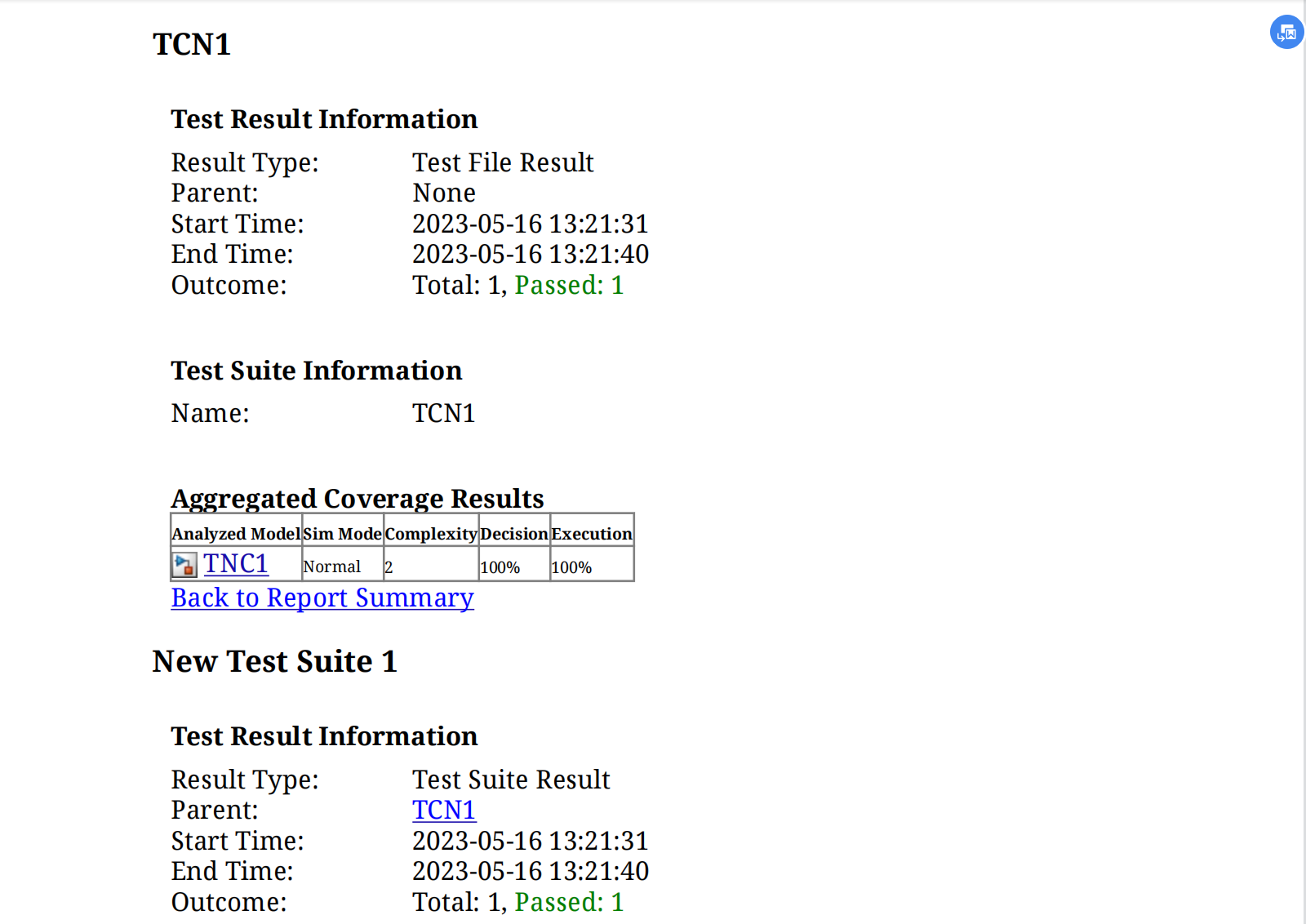
# **生成报告分析**



### 分析1：开头界面会提供基本信息，包括Title,文件头名、Author作者、Data创作的时间、以及测试的环境：平台环境Win64、Matlab的版本R2021b

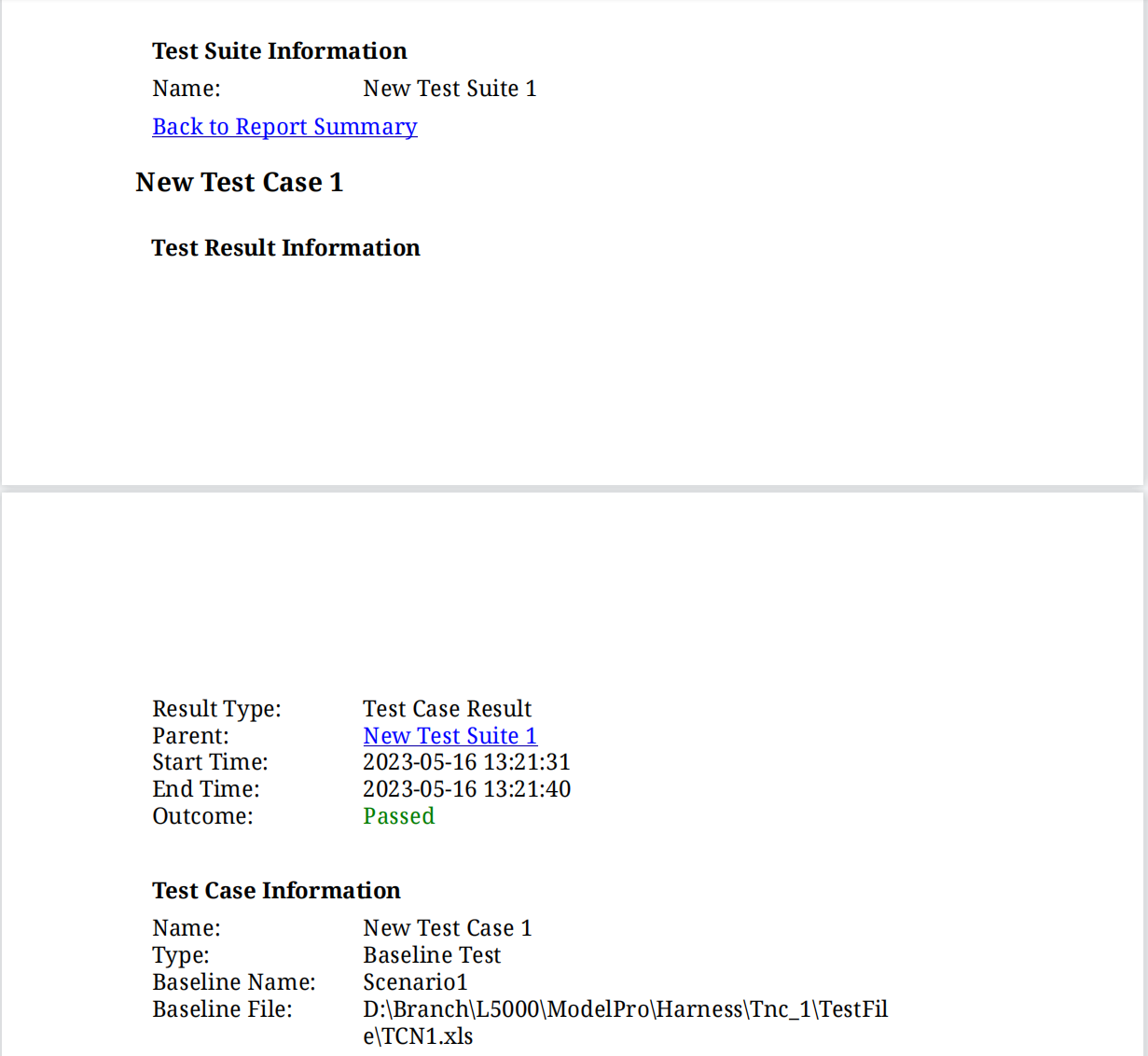


### 分析2：上图中可以见到：测试Tese Manager 提供的测试套件和用例在整个运行期间用的时间



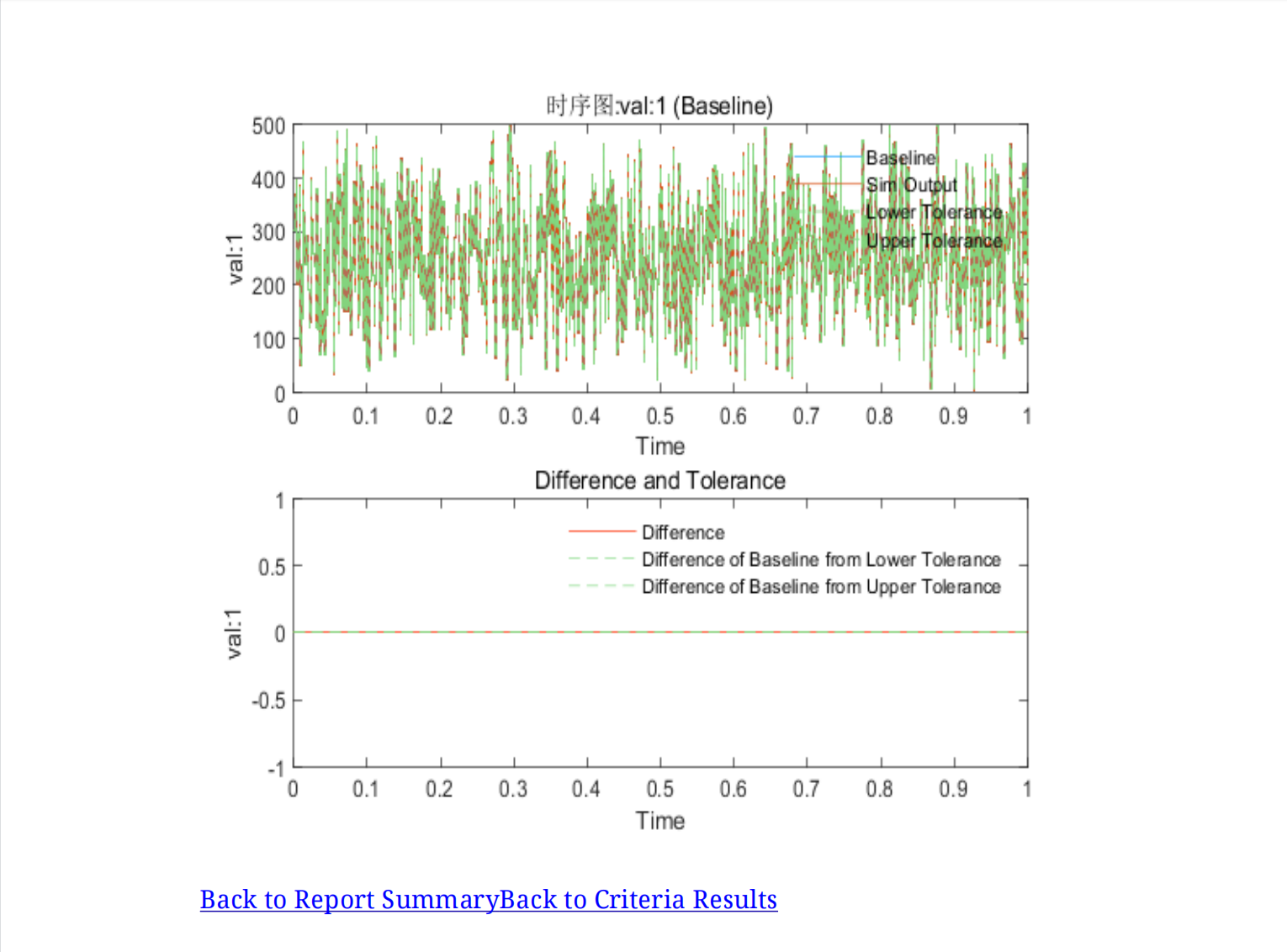
### 分析3：上图中提供TCN1这个模型的测试结果信息

* **测试的开始时间、结束时间、和测试结果的统计（数量、通过和失败的数量）**
* **比较重视的就是汇总的测试覆盖度结果：结果显示TCN1这个模型采用标准的模拟模式，其复杂度为2级，最后得到该模型在测试过程中：判断过程100%覆盖、执行实施过程100%覆盖**
* **Test Suit是测试套件，其父级时TCN1，该套件测试的信息有测试的开始时间、结束时间、和测试结果的统计（数量、通过和失败的数量）**



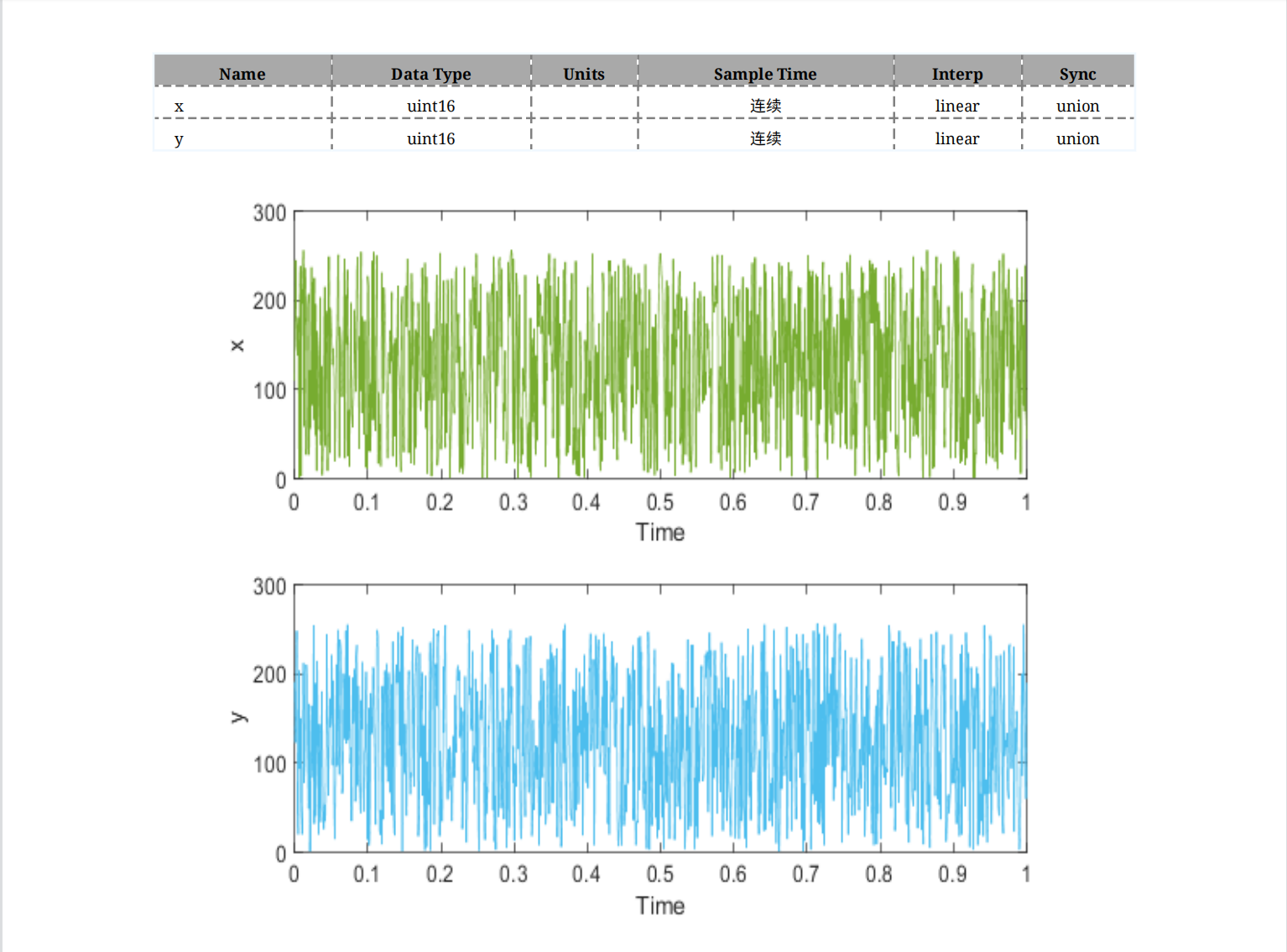
### 分析4：上图对测试用例加入测试运行的介绍：

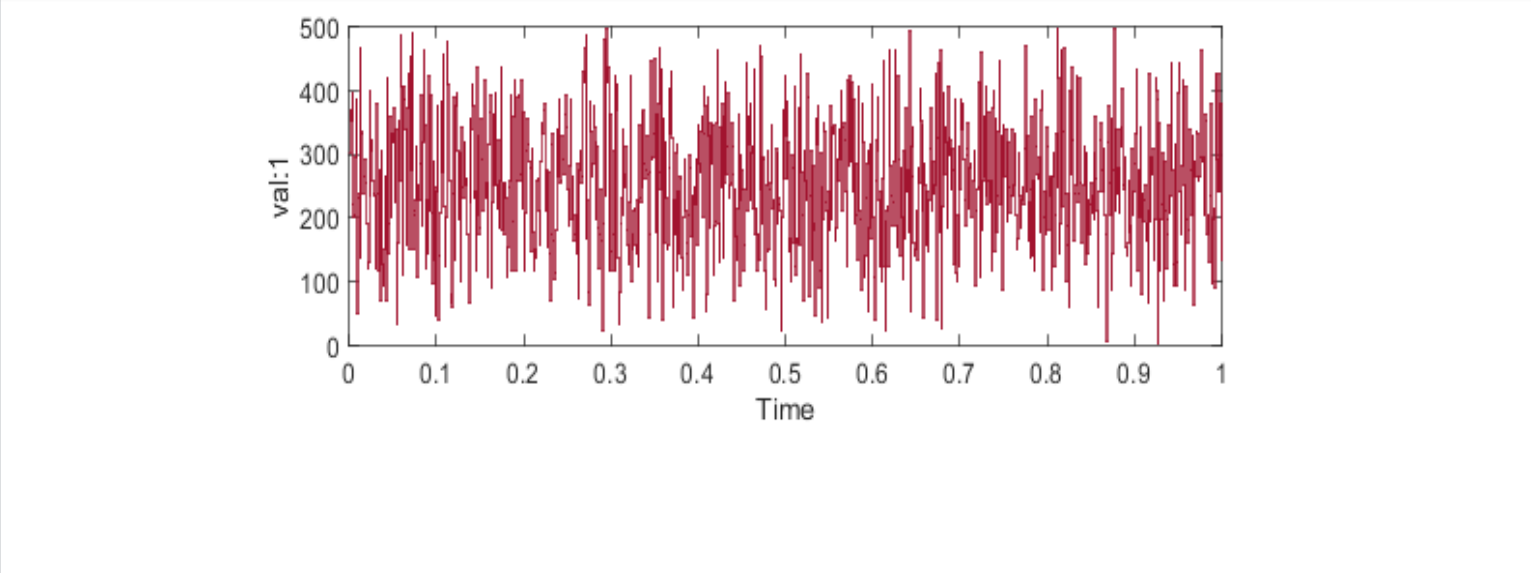
* **运行的开始时间、结束时间、和运行结果**
* **测试类型为BaseLine Test**
* **用例的名字**
* **用例的文件存放路径**



### 分析5：上图是Baseline Comparison 基线比较的结果：

* **时序图中蓝色的Baseline是测试用例的输出val，红褐色的sim Output是模拟输出val**
* **Different and Tolerance的图中给出三种结果**
  + **一是区别Different,代表Baseline中的输出和sim Output模拟输出的val之间的区别（差值）**
  + **Different of Baseline from Lower Tolerance指低容差下的差值**
  + **Different of Baseline from Upper Tolerance指高容差下的差值**





### 分析6：上图是测试用例中x,y,val的时序图，报告介绍其文件所在位置

### 系统报告信息

