

EMTG Testatron Tutorial: Creating a Test

July 25, 2023

Revision Date	Author	Description of Change
June 30, 2023	Joseph Hauerstein	Initial revision.

Contents

1	Introduction	1
2	Reason for Creating a Test	1
3	Test Creation Process	1
3.1	Generating Files	1
3.2	Running the New Test	3

List of Known Issues

- | | | |
|---|--|---|
| 1 | Testatron does not update the paths of files in HardwareModels. | 3 |
| 2 | The output produced by running Testatron is different from PyEMTG. | 3 |

List of Acronyms

EMTG Evolutionary Mission Trajectory Generator

SNOPT Sparse Nonlinear OPTimizer

NLP Nonlinear Program

MBH Monotonic Basin Hopping

1 Introduction

Welcome to the second Evolutionary Mission Trajectory Generator (EMTG) Testatron tutorial. The goal of this tutorial is to familiarize users with the process of creating a new test to use with the Testatron system. Before starting this tutorial, you should have completed the first tutorial on how to run Testatron.

2 Reason for Creating a Test

Before creating a test, it is first important to consider whether creating a new test is necessary. At present, there should be tests for all EMTG features, so there should not be a need to create tests for existing features. However, if you are adding a new feature to EMTG, or discover that a test was not sufficient for an existing feature, you should create a test that makes sure that feature is working. Essentially, there should be an example case for each new feature that makes sure the feature is working properly.

3 Test Creation Process

If you need to create a test for a new feature, this section will explain the process of creating a test, and how to add the test to Testatron.

3.1 Generating Files

The *.emtgopt and *.emtg files are the two required files to create a test for Testatron. This tutorial will use the OSIRIS-REx mission from the EMTG Tutorials as an example to demonstrate the creation of a new test.

1. Make a copy of the OSIRIS-REx.emtgopt file from the EMTG tutorials and move it to a new directory called OSIRIS-REx_testatron.
2. Open the file in PyEMTG and rename the mission to OSIRIS-Rex_testatron.
3. Navigate to the Solver Options page and change the following options as shown in Figure 1:
 - **Inner-Loop Solver Mode:** “NLP with initial guess”
 - The test version of the OSIRIS-REx mission will not use Monotonic Basin Hopping (MBH) because it is a stochastic optimizer. Instead, it will use the Nonlinear Program (NLP) solver SNOPT, which is deterministic, to propagate the solution. Using these settings ensures the test results are reproduceable.
 - **Quiet NLP Solver:** “Off”
 - **Trial decision vector or initial guess:** path to a previous run of the OSIRIS-REx mission

Inner-Loop Solver Parameters	
Inner-loop Solver Mode	
NLP solver	NLP with initial guess
NLP solver mode	SNOPT
Quiet NLP solver?	Optimize
Enable NLP chaperone?	<input type="checkbox"/>
Stop NLP upon attaining goal?	<input checked="" type="checkbox"/>
Always write MBH archive file?	<input type="checkbox"/>
Print NLP movie frames at every major iteration?	<input type="checkbox"/>
Feasibility tolerance	1e-05
Optimality tolerance	1e-05
NLP max step	1
SNOPT major iterations limit	8000
SNOPT minor iterations limit	500
SNOPT maximum run time (s)	15
Check for new NLP solution to write to file every N seconds. Only works if using NLP chaperone.	1000000
Check derivatives via finite differencing?	<input type="checkbox"/>
Trial decision vector or initial guess	...

Figure 1: OSIRIS-REx_testatron Solver Options.

4. Update the path to the working directory so that output will go into the new folder. Then, run the Mission using PyEMTG (File->Run or Ctrl+r).

Now that *.emtgopt and *.emtg files have been generated, they need to be placed into the correct Testatron test folder. Imagine that the OSIRIS-REx_testatron.emtgopt file was using a new feature in the Solver Options tab. In this case, the new test would go in the “solver_options” folder in the Testatron tests directory.

5. Copy the *.emtgopt and *.emtg files into <EMTG_root_dir>\testatron\tests\solver_options.

NOTE: You should not copy tests into the “tests_that_dont_work” folder as this folder is for tests that are expected to fail with the current build of EMTG.

6. Copy the OSIRIS-REx mission default.emtg_spacecraftopt and default.emtg_propulsionsystemopt files from the OSIRIS-REx hardware_models folder into the <EMTG_root_dir>\testatron\HardwareModels folder. All other required files in the hardware_models folder and the Universe folder are already in the corresponding Testatron folders.

NOTE: There is no need to change the paths to “hardware_models” or the Universe folder in the *.emtgopt file, because Testatron will do this automatically. However, if you use a *.emtg_spacecraftopt or *.emtg_propulsionsystemopt file that contains a path to another file, such as a *.ThrottleTable file, this path will need to be updated manually within the spacecraft or propulsion system options file.

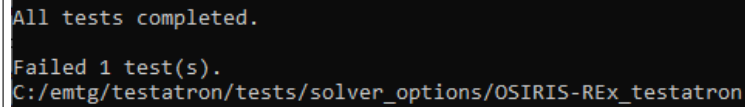
3.2 Running the New Test

Now that all the required files are added, the new test can be run using the run test case command:

```
python testatron.py --emtg <EMTG_root_dir>\bin\EMTGv9.exe --pyemtg <EMTG_root_dir>\PyEMTG\ -c <EMTG_root_dir>\testatron\tests\solver_options\OSIRIS-REx_testatron
```

The test should run quickly, but you will notice that it fails on the initial run as shown in Figure 2.

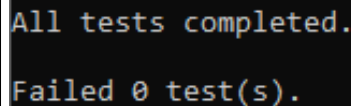
NOTE: This failure is due to Testatron slightly changes some of the numbers used in the OSIRIS-REx_testatron.emtgopt file, which causes EMTG to produce slightly different results. This is likely occurring from number to string conversions in python. A bug ticket has been created for this.

A terminal window with a black background and white text. The text reads: "All tests completed." followed by "Failed 1 test(s)." and then the file path "C:/emtg/testatron/tests/solver_options/OSIRIS-REx_testatron".

```
All tests completed.  
Failed 1 test(s).  
C:/emtg/testatron/tests/solver_options/OSIRIS-REx_testatron
```

Figure 2: Initial run output.

To address the slight numerical differences, navigate to the output folder for the test run (located in <EMTG_root_dir>\testatron\output\ <time-of-test>) and copying the OSIRIS-REx_testatron.emtg file into the solver_options folder. Run the test again using the same command and it should pass. Example output is shown in Figure 3.

A terminal window with a black background and white text. The text reads: "All tests completed." followed by "Failed 0 test(s).".

```
All tests completed.  
Failed 0 test(s).
```

Figure 3: Fixed run output.

Congratulations, you have successfully added a test to Testatron and completed the Testatron Tutorials!