

EMTG Testatron Tutorial: Running Testatron

July 25, 2023

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

Contents

1	Introduction	1
2	Setup	1
3	Configuration	2
3.1	Test Suites	2
3.2	Other Options	4
4	Running Testatron	4
5	Output	5

List of Known Issues

- | | | |
|---|--|---|
| 1 | An NLP test is incorrectly listed as failed | 4 |
| 2 | Some tests fail if EMTG is not in the C:\emtg\ directory | 1 |

List of Acronyms

EMTG Evolutionary Mission Trajectory Generator

SPICE Spacecraft Planet Instrument Camera-matrix Events

NLP Nonlinear Program

1 Introduction

Welcome to the Evolutionary Mission Trajectory Generator (EMTG) Testatron tutorial, which is designed to get new users familiar with the Testatron regression testing system. Testatron allows users to execute various tests to ensure any changes made to the code did not adversely affect its results. Before starting this tutorial, it is expected that you have installed EMTG and have completed the “Tutorials and Intro”, “OSIRIS-REx”, and “LowSIRIS-REx” EMTG tutorials. This tutorial will assume that you set up EMTG following the installation guides located in `<EMTG_root_dir>/docs/0_Users/build_system`, where `<EMTG_root_dir>` is the directory where you downloaded/cloned the EMTG repository. Any commands (denoted by the use of **monospace font**) shown in this tutorial that utilize `<EMTG_root_dir>` syntax require you to replace it with the path to your EMTG root directory.

NOTE: It is recommended to use `C:\emtg\` for `<EMTG_root_dir>` because some tests depend on files being in this directory.

2 Setup

Follow the steps below to set up your terminal to run Testatron.

1. Navigate to the `<EMTG_root_dir>/testatron` folder. This directory contains the `testatron.py` file, which is the primary file used to run testatron. It also includes tests for all the features of EMTG, which can be found in the `<EMTG_root_dir>/testatron/tests` folder.
2. Open the Miniforge Prompt used during EMTG installation.
3. Navigate to the testatron folder using the command:

```
cd <EMTG_root_dir>\testatron
```
4. Activate your EMTG python environment created during installation using the command:

```
mamba activate PyEmtgEnv
```

3 Configuration

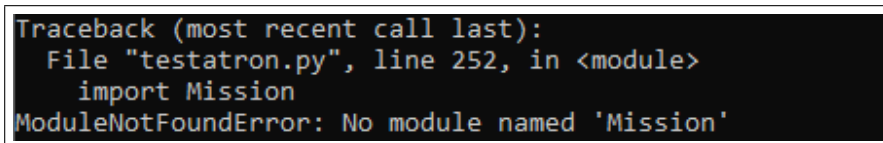
The various configuration options for Testatron can be seen by running the command:

```
python testatron.py --help
```

By passing the “--unit” option to testatron.py, you can run the unit tests. Try to run the Testatron unit tests using the command:

```
python testatron.py --unit
```

Without further configuration, it is likely that this command will cause an error, similar to Figure 1.



```
Traceback (most recent call last):
  File "testatron.py", line 252, in <module>
    import Mission
ModuleNotFoundError: No module named 'Mission'
```

Figure 1: Testatron error without correct file paths.

This error occurs because Testatron needs to be told where your EMTGv9.exe file and PyEMTG directory are if they are not in the default location (“C:\emtg\bin\EMTGv9.exe” and “C:\emtg\PyEMTG\” respectively). These can be set on the command line when Testatron is run, or you can change the defaults by editing the testatron.py file directly. This tutorial will focus on the command line approach. To use the command line to set the location of the EMTGv9.exe file, use the “-e” or the “--emtg” option followed by the path to the executable. To set the location of the PyEMTG directory, use the “-p” or “--pyemtg” command followed by the path to the folder. For example, the following command would run the unit tests with the correct EMTG and PyEMTG locations:

```
python testatron.py --emtg <EMTG_root_dir>\bin\EMTGv9.exe --pyemtg <EMTG_root_dir>\PyEMTG\ --unit
```

These two options must be set each time Testatron is run unless your EMTGv9.exe and PyEMTG directory are in the default location.

3.1 Test Suites

While it is most common to run the unit tests, there are a few other sets of tests that can be run. When running Testatron, only one of these sets should be specified. The available options are explained in Table 1. Note that examples do not include paths to EMTG or PyEMTG for brevity.

Test Type	Option	Explanation
Unit tests	“-u” or “--unit”	<p>Runs all tests that are not expected to fail. It is recommended to run these tests when adding a new feature to make sure EMTG is still working properly.</p> <p>Ex: <code>python testatron.py -u</code></p>
Failed tests	“--failure <path-to-failed_tests.csv>”	<p>Reruns all tests in a “failed_tests.csv” file produced by a previous testatron run. Do not include “failed_tests.csv” in the path. You can include multiple paths in a space separated list.</p> <p>Ex: <code>python testatron.py --failure output\Thu_Jun_15_121445_2023</code></p>
Mission tests	“-m” or “--mission”	<p>Runs the tests associated with specific missions that used EMTG. (Not available in the public release.)</p> <p>Ex: <code>python testatron.py -m</code></p>
Test cases	“-c <list-of-test-cases>” or “--cases <list-of-test-cases>”	<p>Runs the test cases specified in the space separated list following the command. This must be the full path to the test and does not include a file extension.</p> <p>Ex: <code>python testatron.py -c <EMTG_root_dir>\testatron\tests\output_options\output_options_frameICRF <EMTG_root_dir>\testatron\tests\spacecraft_options\spacecraft_options_Chemmargin</code></p>
Test folders	“-f <list-of-folders>” or “--folders <list-of-folders >”	<p>Runs all the test cases in each folder specified in the space separated list of folders following the command.</p> <p>Ex: <code>python testatron.py -f global_mission_options journey_options</code></p>
Update truths	“--update-truths”	<p>Replaces the truth cases (*.emtg files) for all tests with the output of this Testatron run. This is useful if an update has caused every test to fail (e.g., a new option has been added).</p> <p>Ex: <code>python testatron.py --update-truths</code></p>
All tests	“-a”, “--all”, or none of the above options	<p>Runs all tests, including those in the “tests_that_dont_work” folder.</p> <p>Ex: <code>python testatron.py -a or</code> <code>python testatron.py</code></p>

Table 1: Available Testatron test suites.

3.2 Other Options

You can ignore specific Mission, Journey, or MissionEvent attributes by using the “--ignore” option followed by a list of attributes to ignore. Mission attributes should be preceded by “M.”, Journey attributes by “J.”, and MissionEvent attributes by “E.”. When defining this and other lists you should avoid putting them in quotes or brackets and using commas. The list of attributes can be found in the class definitions for Mission, Journey, and MissionEvent. As an example, to run the Testatron unit tests ignoring the chemical_oxidizer_used Mission attribute, you can run the command:

```
python testatron.py --emt看 <EMTG_root_dir>\bin\EMTGv9.exe --pyemt看 <EMTG_root_dir>\PyEMTG\ --unit --ignore M.chemical_oxidizer_used
```

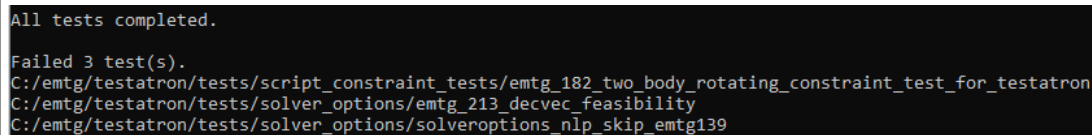
4 Running Testatron

If you have not already done so, run the Testatron unit tests by running the following command explained in the previous section:

```
python testatron.py --emt看 <EMTG_root_dir>\bin\EMTGv9.exe --pyemt看 <EMTG_root_dir>\PyEMTG\ --unit
```

Testatron will execute the unit tests, which can take up to half an hour. Do not worry if some errors occur during specific EMTG runs. These are most often normal SPICE errors and do not indicate a failed test. When Testatron finishes, it will list any failed tests and the path to the test as shown in Figure 2.

NOTE: The “solveroptions.nlp_skip.emtg139” test is currently expected to fail unless the “--ignore M.number_of_solution_attempts” argument is utilized.



```
All tests completed.  
Failed 3 test(s).  
C:/emt看/testatron/tests/script_constraint_tests/emtg_182_two_body_rotating_constraint_test_for_testatron  
C:/emt看/testatron/tests/solver_options/emtg_213_decvec_feasibility  
C:/emt看/testatron/tests/solver_options/solveroptions_nlp_skip_emtg139
```

Figure 2: Testatron output with some failed tests.

5 Output

The results from the Testatron run will be located in the `<EMTG_root_dir>/testatron/output/<time-of-test>` folder. Where `<time-of-test>` is the date and time the test was run so that each run goes into a separate folder. This folder will contain the `*.emtg` file from each run of EMTG, along with the corresponding `*.emtgopt` file, and any other files, such as `*.emtg_spacecraftopt` files. It also includes the “failed_tests.csv” file and a “test_results.csv” file. The “failed_tests.csv” file lists only the failed tests and the amount of error and tolerance for each test. If a test is listed with no other information, it is likely that the test caused an error while running EMTG. As discussed in the Configuration section, the “failed_tests.csv” file can be used with the “--failure” option to rerun the failed tests. The “test_results.csv” file lists all the tests that were run and whether they were successful. These files can be used as a starting point to troubleshoot any issues with EMTG introduced by new updates.