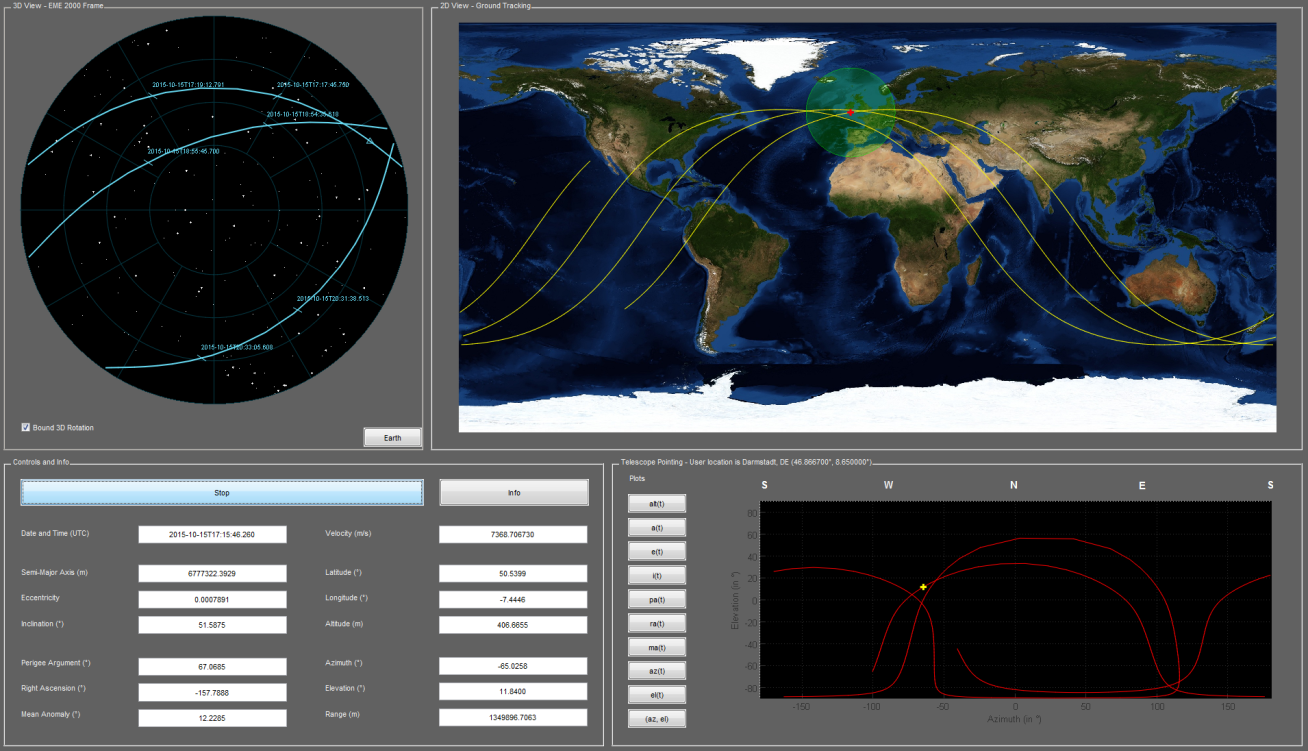
ISSTracker Software Description

ISSTracker v2

Rami Houdroge

October 2015



# Scope

The scope of this document is to present the real time International Space Station Tracker developed in Matlab.

# Acknowledgements

This project wouldn’t have been possible without the work of hundreds of men and women, whom I would like to thank most sincerely for making available to the general public an immense amount of scientific data and software. Parts and/or resources of the following projects were used in the development of this software:

1. OREKIT (ORbits Extrapolation KIT), C.-S. [***website***](https://www.orekit.org/forge/projects/orekit)
2. Apache Commons Math, Apache Commons [***website***](http://commons.apache.org/math/)
3. Solar System Dynamics, Jet Propulsion Laboratory, NASA [***website***](http://ssd.jpl.nasa.gov/)
4. Marshall Solar Activity Future Estimation, NASA [***website***](http://sail.msfc.nasa.gov/)
5. ISS orbital parameters, NASA Human spaceflight [***website***](http://spaceflight.nasa.gov/realdata/sightings/SSapplications/Post/JavaSSOP/orbit/ISS/SVPOST.html)
6. Blue Marble, Earth Observatory, NASA [***website***](http://earthobservatory.nasa.gov/Features/BlueMarble/)
7. International Center for Global Earth Models, IGFS and IAG [***website***](http://icgem.gfz-potsdam.de/ICGEM/modelstab.html)
8. Earth Orientation Center, OBSPM [***website***](http://hpiers.obspm.fr/eop-pc/index.php?index=leapsecond&lang=en)
9. International Earth Rotation and Reference Systems Service ***[website](http://www.iers.org/IERS/EN/Publications/TechnicalNotes/tn36.html?nn=94912)***

# License

Copyright 2015 Rami Houdroge

Licensed under the Apache License, Version 2.0, January 2004; you may not use this file except in compliance with the License.

You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and limitations under the License.

# Table of Contents

[Scope i](#_Toc373267365)

[Acknowledgements i](#_Toc373267366)

[License i](#_Toc373267367)

[Table of Contents ii](#_Toc373267368)

[1 Introduction 1](#_Toc373267369)

[2 Required 1](#_Toc373267370)

[3 First run 1](#_Toc373267371)

[4 Package Contents 2](#_Toc373267372)

[5 Software Architecture 3](#_Toc373267373)

[6 Dynamic Model 3](#_Toc373267374)

[6.1 Forces 3](#_Toc373267375)

[6.2 Reference Frames 3](#_Toc373267376)

[6.3 Footprint Computation 4](#_Toc373267377)

[7 Prediction Accuracy 4](#_Toc373267378)

[7.1 Values for V1.0.0 4](#_Toc373267379)

[7.2 Values for V1.1.0 5](#_Toc373267380)

[8 Future Features 5](#_Toc373267381)

# Introduction

The purpose of this open-source software is to provide Matlab users with the means to track the ISS in real-time, and to provide them with:

* Real-time Keplerian orbital parameters,
* Real-time plot of passes
* Real-time 3D visualisation of the location of the ISS,
* Ground trace and real-time 2D location of the ISS,
* Real-time telescope pointing coordinates from a given user location.

This is done by propagating the orbit bulletin provided by Nasa. The ISS 24h ephemeris data is regularly updated, and provides expected positions and planned manoeuvres for two weeks.

The low-level libraries Commons-Math and Orekit are used directly within Matlab to propagate the orbit bulletin and a Matlab Guide figure updated every 10 ms displays the real-time data.

# Versions

|  |  |
| --- | --- |
| **Version** | **Changes** |
| 2.0 | - Migrated to Orekit 7.0  - Added simulated time  - Added footprint  - Added plots for each orbital parameter  - Added sky map plot  - Updated UTC-TAI.history file |
| 1.0.0 | Initial version |

# Required

Matlab (No toolboxes required).

An internet connection.

A good PC. Note: one can disable rendering Earth in the 3D plot, by setting the renderEarth boolean to false, in the simulation\_parameters.m file. This will result in a sphere with a plot of the coast line.

# First run

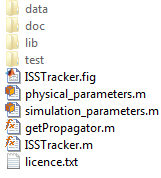
Just run the ISSTracker script.

A warning dialog that says “Loaded required libraries” will pop up. This is due to the fact that the commons-math3 and Orekit libraries need to be loaded prior to launching the ISSTracker. The warning box will load these libraries and the button “Run ISSTracker...” will run the GUI.

# Package Contents

The ISSTracker zip file is available on the Mathworks file exchange.

Upon extraction, the following files and folders are created:



The *data* folder holds all the data files used to propagate the orbit bulletin, such as the gravitational coefficients files, the planetary ephemeris, the Blue Marble files etc (, , , , ).

The *doc* folder contains the ISSTracker Software Description.

The *lib* data contains the Commons-Math 3.4.1 and Orekit 7.0 jars, downloaded directly from their respective websites (, ). Both, and the ISSTracker are under Apache License V2.0 of January 2004.

The *test* folder contains a test script (run.m) that propagates every orbit bulletin available in the ISS ephemeris and prints to the console the propagation errors after 24h (‎see §4).

The other files (listed hereunder) are the actual ISSTracker GUI files:

* *ISSTracker.fig* and *ISSTracker.m* are respectively, the Matlab GUI file and the corresponding script.
* *physical\_parameters.m* and *simulation\_parameters.m* areproperties classes that hold respectively, the physical model parameters and the simulation (e.g. ground trace length) parameters.

# Software Architecture

The ISSTracker relies heavily on the interaction of Matlab and Java libraries.

The Model-View-Control design pattern is applied here to isolate representation, interaction and computation of information and data. The *ISSTracker.m* script is organized with that in mind.

## General View

## Important Data Structures

### OrbitalData

OrbitalData is a structure produced by reading a given source, such as the NASA feed, and contains a number of given fields. It is structured as follows:

1x14 struct array with fields:

text

time

data

Where 14 represents the number of state vectors detected in the source. Each state vector can be accessed using the command orbitalData(k) where k is the number of the desired state vector.

For each of these entries, there are three fields which hold the data:

|  |  |  |
| --- | --- | --- |
| orbitalData(1).text | orbitalData(1).time | orbitalData(1).data |
| raw: [1x3002 char]  lines: {1x66 cell} | year: 2015  month: 10  day: 12  hour: 12  minute: 0  second: 0  msecond: 0 | X: -1.0209e+06  Y: -5.1410e+06  Z: 4.2898e+06  XDot: 6.6107e+03  YDot: 1.6442e+03  ZDot: 3.5377e+03  weight: 907028 |

The selected state vector is then fed into the propagator and used for propagating the orbit, after instantiating the relevant Java objects.

### OrekitData

### Main Handles

### Functions that change the handles

This list contains the functions that change the handles :

# Dynamic Model

## Forces

Forces taken into account include:

* Earth attraction,
* Moon and Sun attractions,

The Solar System Dynamics DE405 ephemeris ‎[R3] is used in order to compute the gravitational forces.

The Holmes Featherstone attraction model is used with the combined gravity field model EIGEN-5C ‎[R7], up to degree and order 10.

At the moment:

* Higher degrees and orders do increase accuracy, but not enough,
* Solar Radiation Pressure and Drag forces are not accounted for.
* Thrust manoeuvres are not accounted for.
* The attraction of both inner and outer solar system bodies was found not to affect the propagation error.

## Reference Frames

The Earth Mean Equinox frame of epoch 2000 (EME2000) is used for the 3D plot. The International Terrestrial Reference Frame is used for the 2D ground trace. At the moment, no EOP data is used.

The azimuth and elevation graph is drawn in a Topocentric Frame from a given user location on Earth’s surface. This location is configurable in the simulation parameters file (parameters userLat, userLon and userAlt).

For the orbit propagation, EME2000 is also used. It is also the frame, amongst others, in which the ISS ephemeris is given by NASA.

Transformations from one frame to another are handled internally by Orekit and use the IERS 2010 Conventions [R9].

## Footprint Computation

The footprint formula used is straightforward and assumes Earth is spherical. The angle between the edge of the footprint and the ISS is given by

Where *ae* is the equatorial radius of Earth and *h* is the altitude of the ISS. This translates as a list of longitudes and latitudes generated, for each *k=1…n*, as per:

The factor n is the number of dots that approximate the circular footprint, and corresponds to the parameter footPrint in simulation\_parameters.

# Prediction Accuracy

## Configuration

Unless otherwise stated, all tests are completed with an Intel Core i3 2100 3.10 GHz processor and 4.00 GB PC10600 DDR3.

The test consists of the *run.m* script that can be found in the test folder.

## Values for V2.0

Using the same dynamic model as for V1.0.0, the test run shows a significant degradation of the prediction accuracy after 24h. The position error is more than 8 km.

The run results can be found in the file titled *17072013.txt* in the test folder.

## Values for V1.0.0

At V1.0.0 release, in October 2013, a propagation error of about 5 km and 5 m.s-1 after 24 hours could be expected.

The run results can be found in the file titled *17072013.txt* in the test folder.

# Future Features

Future versions will, in no particular order:

* Be concerned with increasing the accuracy of the dynamic model,
* Implement a nicer 3D model of the ISS,
* Implement a nicer info figure, with links to project pages,
* Show how to install the underlying libraries into the static Matlab classpath,
* A better software architecture description.