Getting Started Processing SLR Data with ODTK

1 Introduction

ODTK can process *satellite laser ranging* (SLR) from a network of SLR tracking facilities. The SLR measurement is an emitted pulse of light that travels from the ranging station, is reflected from a retroreflector on a satellite, and returns to the ranging station. The corrected photon time-of-flight provides an extremely precise measure of the distance between the retroreflector and the ranging point on the Earth.

SLR measurements are used for precise geodetic research and tend to be readily available via the internet. Because of the high precision of laser ranging data, SLR measurements are useful for verifying the capable accuracy of an orbit determination system.

Satellite laser ranging is a relatively straightforward measurement to process compared to other data types available to ODTK. However, to fully correct the SLR measurement type, you need wavelength, calibration, and meteorological data from the tracking facility. Outside of a simulation environment, these data are usually retained as part of the SLR measurement.

2 ODTK configuration

ODTK requires at least one Facility object and one Satellite object equipped with a retroreflector to process SLR measurements. Simulation and estimation of SLR measurements is based on the construction of simple *tracking strands* between facilities and the retroreflector-equipped satellites. SLR measurements mostly depend on the following (bold indicates estimation is possible):

- The **location** of the ranging facility
- The ranging facility measurement **bias**
- The troposphere near the ranging facility
- The **orbits**, or trajectories, of the retroreflector
- The **location** of the retroreflector on the satellite
- The **retroreflector** delays

While it is actually possible to estimate the delay (or center-of-mass offset advance) for a retroreflector array using ODTK, this parameter is usually very difficult to separate from the ranging measurement bias of the facilities and perturbing effects on the satellite. If you are estimating those other effects with actual tracking data, it is best not to estimate retroreflector delay.

3 SLR tracking data: NPRange

A measurement type called NPRange is provided in ODTK to represent SLR. You can apply the Marini-Murray troposphere correction model to this type of range measurement. This model requires temperature, pressure, humidity, and laser wavelength values to be embedded with the tracking data.

NPRange is short for "normal-point range", a derived product computed from many ranging measurements sampled at a much higher, or "full", rate. Normal points have become the ILRS standard for SLR, although ODTK does not practically discriminate as to whether NPRange data are the (smoothed) normal-point product or the (unsmoothed) full-rate product.

Actual SLR measurements are obtained in either the Consolidated Laser Ranging Data (CRD) format, ILRS normal-point file format, or the older ILRS full-rate file format (formerly known as the Project MERIT-II format), all of which are straightforward ASCII text formats. Of these three formats, ODTK natively supports the CRD and the full-rate format, which are documented as part of the ODTK help system and the ILRS web pages. Since NPRange can refer to either normal-point SLR data or full rate SLR data in the full-rate format, the moniker "NPRange" in ODTK may be somewhat deceiving. Within the context of ODTK, you should understand that "NPRange" simply refers to a special ranging type intended to correctly model SLR, regardless of data-compression status or data format.

ODTK can simulate NPRange in the generic observation (gobs) format. The generic observation format is a binary format that has the benefit of maintaining the full precision of simulated measurements. This binary format is only useful, however, for simulation and estimation inside ODTK, since the format does not carry sufficient documentation for other uses. Facility attributes are provided to allow simulation of NPRange data by enabling you to specify default values for temperature, pressure, humidity, and laser wavelength. You can then enter these simulated data in the gobs format for estimation. ODTK does not have the capability to generate simulated data in any ILRS file format.

4 Orbit determination

ODTK can regard the ephemerides of retroreflector-equipped satellites in different ways while processing SLR measurements. Reference or "truth" ephemerides may be provided for the satellites, thus limiting estimation to facility location(s) and ranging biases only. Or, the satellite orbits may be simultaneously estimated with uncertain facility locations and ranging biases, as might be done during a colocation campaign verifying a new facility. Typically, all facility locations are well known, in which case, estimation of the orbit and its covariance is the primary objective.

5 Orbit determination example

Although the following presentation is detailed, it occasionally presumes that you have a basic understanding of orbit determination and prior familiarity with ODTK. If this tutorial is your first exposure to the ODTK user interface, you may want to first visit the tutorial "Getting Started with ODTK" by clicking the "Using the GUI" link within the LaunchPad Resources tab.

5.1 Obtaining and loading SLR tracking data

1. Start ODTK.

2. Create a new scenario. Change the name to "GettingStartedSLR".

3. Set the scenario's EarthDefinition EOPData filename to "EOP-All-v1.1.txt" and click "Apply".

4. Using your favorite FTP utility (or a web browser), download the file:

   ftp://cddis.gsfc.nasa.gov/pub/slr/data/npt_crd/lageos2/2019/lageos2_201906.npt

   This file contains tracking data for the LAGEOS2 satellite for the month of June, 2019 in the ILRS CRD format. If the file is unavailable or you do not have Internet access, you can obtain the file here:

   ```
   <Install Directory>\ODTK\UserData\TrackingData\lageos2_201906.npt
   ```

> **Tip**: For more about SLR tracking data availability and formats, visit https://cddis.nasa.gov/Data_and_Derived_Products/SLR/SLR_data_and_product_archive.html

5. Expose the scenario's Object Properties pane by double-clicking the scenario object in the Object Browser window. Under the "Measurements" category, beside the word "Files" (in shorthand, write this as Scenario.Measurements.Files), click where the panel shows "click to edit".

6. When the List pane appears, click the "Add" button on the right, then click where the panel says "click to edit" under the Filename column.

7. When the Open pane appears, select "Files of type" to be limited to "CRD Laser (*.crd, *.npt)".

8. For the File Name, go to the "lageos2_201906.npt" file. After selecting the file, verify that "true" appears in the Enabled column. Click OK to collapse the Open pane and the List pane.

9. Click "Apply" at the top of the Object Properties pane to update the scenario's Object Properties. The SLR data are now available for use by the scenario.

10. Save the scenario file before proceeding.

5.2 Initializing an SLR network

AGI supplies tracking-system files containing entire networks of facility locations. For the SLR network, two files are supplied: one for active SLR facilities, and one for inactive facilities.

> **Tip**: As SLR facilities can change their activity status, and new facilities are occasionally introduced, you are encouraged to maintain these files according to the network conditions during operational times of interest.

1. Highlight the scenario object in the object-browser pane. Click the "File" drop-down menu and select "Import -> Import Object".
2. In the file selection pane, first select "TrackingSystem Files (*.tso)" as the file type, then go to and select:

   ```
   <Install Directory>\ODTK\AppData\Databases\TrackingSystems\ILRS.tso
   ```

   ODTK will add a Tracking System object to the scenario, populated with SLR facilities.

> **Tip**: For more about SLR facility locations and their updates, visit http://itrf.ensg.ign.fr/ITRF_solutions/.

3. Open a few facility objects at random within the tracking system and verify that the RangingMethod = "Retroreflector". Verify that NPRange is an included data type in the MeasurementStatistics list. This should be true for all SLR facilities.

> **Tip**: Rather than manually reconfigure the RangingMethod setting for every facility, export the SLR tracking system object as a *.tso ASCII text file and use a text editor's "find-and-replace" function to globally substitute "Transponder" for "Retroreflector".

4. Save the scenario file before proceeding.

5.3 Initializing the SLR satellite

Your next step is to create a satellite and initialize its orbital state. Start by downloading the following file:

ftp://cddis.gsfc.nasa.gov/pub/slr/products/orbits/lageos2/190601/ilrsa.orb.lageos2.190601.v70.sp3.Z

This is an SP3 format ephemeris file that has been compressed with the UNIX compress utility (*.Z). Uncompress the file using 7-Zip or other utility. If the file is unavailable or you do not have internet access, you can obtain the file here:

```
<Install Directory>\ODTK\UserData\TrackingData\il-
rsa.orb.lageos2.190601.v70.sp3
```

Using a plain-text editor, open the file and find the following records for the date of Jun 1, 2019:

```
*  2019  6  1  0  0  0.00000000
PL52 -10158.955562  -6441.019876   2383.836878 999999.999999
VL52  21972.632227 -17251.694799  43646.819890 999999.999999
```

The first row of data is the state epoch time (UTC), the second row is the Cartesian position in km, and the following row is Cartesian velocity in decimeters per second (see the SP3c file format specification [1]). The coordinate frame is indicated at the top of the file as SRLF2008; for your purposes, this is effectively the WGS-84 ECF frame used in ODTK. When used with care, these initial conditions are good enough to start an orbit determination filter.

1. Create a Satellite object within the "GettingStartedSLR" scenario. Single-click the name of the object to rename it "LAGEOS2".

2. Verify or set the following values in the satellite's Object Properties white panel, in order. Pay close attention to the units, entering the “m” to indicate the values are in meters. By default, ODTK is displaying and assuming “km”. You also have to shift the decimal point in the velocity terms because the values are in decimeters/sec.

   OrbitState = "Cartesian"
   OrbitState.CentralBody = "Earth"
   OrbitState.CoordFrame = "Fixed" (since based on the SP3)
   OrbitState.Epoch = "1 Jun 2019 00:00:00.000000000 UTCG"
   OrbitState.XPosition = "-10158.955562 km"
   OrbitState.YPosition = “-6441.019876 km"
   OrbitState.ZPosition = “2383.836878 km"
   OrbitState.XVelocity = "2197.2632227 m*sec^-1"
   OrbitState.YVelocity = “-1725.1694799 m*sec^-1"
   OrbitState.ZVelocity = "4364.6819890 m*sec^-1"

   Apply the changes. The OrbitClass value should indicate “MEO”. Save the scenario file before proceeding.

[1] ftp://igs.org/pub/data/format/sp3c.txt

Tip: If you try to apply the coordinate frame after the position and velocity have been applied, ODTK will apply a coordinate transformation to the position and velocity, rather than reset the frame indicator. You must therefore enter the proper coordinate reference frame when the position and velocity values are applied.

3. Go to:

   https://ilrs.gsfc.nasa.gov/missions/satellite_missions/current_missions/lag2_general.html

   to review the physical properties of the LAGEOS 2 satellite. Make note of the mass and the physical diameter. Click the "Array Offset" tab that takes you to:

   http://ilrs.gsfc.nasa.gov/missions/satellite_missions/current_missions/lag2_com.html

   and make note of the standard center-of-mass correction.

Tip: For more about other active SLR satellites, as well as inactive ones, visit http://ilrs.gsfc.nasa.gov/missions/satellite_missions/index.html .

4. Set PhysicalProperties.Mass = 405.38 kg, as taken from the web page.
5. Open the "lageos2_201906.npt " measurement file with a plain-text viewer and examine line 3. It indicates the spacecraft is lageos2, along with the ILRS Satellite Identification (ID) number, based on the satellite COSPAR ID. In this example, the number is 9207002. You must set the Tracking ID equal to this seven-digit number.
6. Set MeasurementProcessing.TrackingID = 9207002.

Tip: Should you forget to set the satellite's Tracking ID, you may see an error message when you run the filter claiming that there are no measurements or that there are not enough measurements.

7. Set ForceModel.SolarPressure.Area = 0.28274 m^2.

   The circular cross-sectional area is based on the satellite's physical diameter (60 cm) from the web site.

Tip: The most significant nongravitational perturbing force for LAGEOS2 is solar radiation pressure.

8. Set the following unmodeled accelerations values:

   ForceModel.UnmodeledAccelerations.ProcessNoise.Frame = Gaussian (RIC)

   ForceModel.UnmodeledAccelerations.ProcessNoise.ZVelocitySigma = 0.002 cm*sec^-1

> **Tip**: At this preliminary stage, the force modeling is incomplete. Artificially adding process noise for "anomalous", unknown, or unmodeled accelerations can allow a better fit to tracking data.

9. The SP3 initial conditions are fixed-Earth coordinates that are not offset for polar motion. Therefore, the in-track (transverse) and cross-track (normal) components of these initial conditions will contain substantial error. The orbit is also "tuned" to a less accurate model. Since the ITRV-based initial orbit uncertainty is unknown, assume it is very large. Enter the following settings under OrbitUncertainty:

   R_sigma = 1000 m
   I_sigma = 10000 m
   C_sigma = 10000 m
   Rdot_sigma = 0.06 m*sec^-1
   Idot_sigma = 0.4 m*sec^-1
   Cdot_sigma = 0.4 m*sec^-1

10. Highlighting the satellite object in the Object Browser window and add a Retroreflector object to it.

11. Select the Retroreflector and verify that EstimateDelay is "false" and that PhaseCenter-Location is zero.

> **Tip**: Use PhaseCenterLocation values to establish the retroreflector location relative to the satellite center of mass. Refer to the ODTK Help system regarding the definition of the satellite body frame.

12. For the retroreflector, set the DelayData = “Distance Units” and Apply. This will convert the Delay from being expressed in time units to distance units, for ease of data entry based on the values you have from the website. Set the DelayModel.Constant = -0.502 m. This is based on the standard center-of-mass offset for LAGEOS2 multiplied by two. Save the scenario file before proceeding.

Tip: When the satellite itself is a spherical retroreflector array, like LAGEOS2, the phase center location is zero, and the Constant is set to the apparent center-or-mass offset. This delay should be *negative* since the observed range relative to center-of-mass is foreshortened in all directions by the center-of-mass offset. *The delay advance is a* ***two-way*** *value, and it should be double the physical center-of-mass offset!*

5.4 Running the filter

1. From the Utilities Menu, select "Data Update …" and go to the "Astro Data" tab. Verify the EOP, SpaceWeather, and leap second files are checked and then click the "Update Now" button.

Tip: SLR measurements are sensitive to inaccuracies in Earth orientation. This action updates Earth orientation parameters, ensuring that the EOP values are accurate.

2. Highlight the scenario object in the Object Browser window and insert a new Filter object.
3. Set or verify the following filter attributes:

   ProcessControl.StartTime = "1 Jun 2019 00:00:00.000000000 UTCG"
   Set ProcessControl.StopMode = "TimeSpan"
   Set ProcessControl.TimeSpan = 10 days

Tip: When using large amounts of tracking data, it may be helpful to shorten the time span of the filter run until the modeling and initial conditions are known to be adequate. This is also true when using Least Squares to initialize filter states.

4. Note the filter's Output.DataArchive.Filename. You will use this file shortly for analyzing the results of the filter run.
5. Set Output.SmootherData.Generate = "True" and note the filter's Output.SmootherData.Filename. You will use this file later for analysis with the smoother.
6. From the ODTK LaunchPad pane, select the ODTK Utilities tab. Look under the Installed Utilities directory and open the **StateContent** utility. Click the "Display State Contents" button to verify that the following states are being estimated:
   - Satellite position and velocity
   - Satellite solar-radiation pressure

- Individual ranging biases for each SLR facility (There will be a Range and Range LongTerm value since they are configured to use a Vasicek stochastic model for the bias.)

7. Save the scenario file before proceeding.
8. Run the filter. This may take several minutes.
9. On completion, check the Message Viewer to verify that there are no error or warning messages.

6 Analysis of results

6.1 Residual analysis

1. Open the Static Product Builder. Click the Inputs tab and add the *.filrun file from the filter run as a data source. Click the Outputs tab and select the Residual Ratios graph style. Click the Run Selected button to generate a new graph.
2. Right-click the legend box to hide it, if necessary.
3. Double-click the graph to expose the properties. Click the Axis tab and change the Y-axis scale to ±10 for better readability. Run the cursor over the various colored symbols to display the tracker to which they belong.
4. Return to the Static Product Builder pane and choose the Residuals graph style as the output product and run it. Hide the legend box if necessary for clarity by right-clicking the graph and select Legend Style | Hide Legend. Reduce the Y scale to ± 1 (meter) for better readability. The filter's orbit fits the majority of measurement residuals well below 1 meter.

Tip: Measurements with residual ratio magnitudes greater than the value set in the Filter’s MeasurementProcessing.ResidualEditing.NominalSigma are ignored in estimation. ODTK's default value is three (3). Rejected residuals appear on the graph with different colored symbols.

6.2 Graphical analysis

1. Add a Smoother object to the scenario.
2. Using the smoother's Input.Files, add the ".rough" file specified the filter generated when it ran. This is the file you noted when you configured the filter’s Output.SmootherData.
3. Note the smoother's Output.DataArchive.Filename. You will use this file shortly.

4. Set the smoother's Output.FilterDifferencingControls.Generate = "True". This will cause the smoother to automatically generate a difference run file with values corresponding to the difference between the filter and smoother outputs. Note the Output.FilterDifferencingControls.Filename. You'll use this file shortly.
5. Save the scenario file before proceeding.
6. Run the smoother. This may take several minutes.

Tip: "Filter-smoother consistency" is a test statistic equal to the difference between the filter estimate and the smoother estimate, divided by the variance of the difference. Theoretically, this statistic is normally distributed with unit variance, such that when the filter-smoother modeling is adequate, large values of the statistic (greater than, say, 3) should be rare.

7. Return to the Static Product Builder and add a new product. Choose the Inputs tab and add the .difrun file just created as the input data source.
8. Choose the Outputs tab of Static Product Builder and select the Pos Consistency graph style as the output product. Click the Run Selected button to generate a new graph.
9. Examine the position consistency graph. All component statistics demonstrate a larger variation than ±3. An experienced orbital analyst would consider the filter's total modeling as being inadequate.
10. Choose the Outputs tab of Static Product Builder and select the "SRP Coeff Consistency" graph style as the output product. Click the Run Selected button to generate a new graph.
11. Examine the SRP coefficient consistency graph. Note whether the statistic is well distributed within ±3. If so, an experienced analyst would consider the modeling of the parameter estimate to be adequate.

Tip: Residuals from the .filrun file are based on a predicted filter state. The filter needs sufficient data to "initialize" and provide optimum answers. If the filter is not fully initialized, some residuals may be deceptively large. Before taking remedial action for a facility reporting large residual biases or RMS, update the initial satellite state using a least squares fit.

7 Update the state

1. You'll use a Least Squares (LS) run to "clean up" the initial state that you got from the SP3 file. By doing the LS fit using the ODTK force model configuration, you ensure that the initial state the filter uses will be compatible with your force model selection.
2. Select the Satellite object and add a Least Squares object.

3. Using the LS object properties, update the Stages and set the start and stop time to be "1 Jun 2019 00:00:00" and "2 Jun 2019 00:00:00", respectively. This will give you a fit span of one day.
4. Set the Output.StatesToTransfer.PosVelCovariance = "true" and the corresponding PosCovSigmaScale and VelCovSigmaScale = 10.
5. Run the LS object. Watch the Message Viewer to see that the RMS converges after a few iterations.
6. Select the LS object and press the Transfer State button on the Toolbar. This will transfer the LS state *and* covariance (because of step 3) to the satellite. The position and velocity covariance will be multiplied by a factor of 10. The scaling enables you to account for the fact that the LS covariance is optimistic because it doesn't model the force model uncertainties. Otherwise, your initial covariance can be too tight and cause the filter to diverge.
7. Rerun the filter to update the .filrun file.
8. Rerun the smoother to update the .smtrun and .difun files.

8 Improving results

1. Return to the Satellite object. Activate the following model improvements:

   ForceModel.Gravity.Tides.SolidTides = "true"
   ForceModel.Gravity.Tides.TimeDependentSolidTides = "true"
   ForceModel.Gravity.Tides.OceanTides = "true"
   ForceModel.Gravity.GeneralRelativityCorrection = "true"
   ForceModel.Gravity.VariationalEquations.Degree = 6
   ForceModel.Gravity.ThirdBodies.UseInVariationalEquations = "true"
   ForceModel.SolarPressure.EclipsingBodies = "Moon"
   ForceModel.SolarPressure.UseInVariationalEquations = "true"
   ForceModel.CentralBodyRadiation.Albedo = "true"
   ForceModel. CentralBodyRadiation.ThermalRadiationPressure = "true"
   ForceModel. CentralBodyRadiation.Area = 0.28274 m^2
   ForceModel.UnmodeledAccelerations.ProcessNoise.Frame = "Gaussian (RIC)"
   ForceModel.UnmodeledAccelerations.ProcessNoise.ZVelocitySigma = 0.0003 cm*sec^-1

Tip: At this stage, the force modeling is more accurate. The artificially added cross-track process noise from "anomalous", unknown, or unmodeled accelerations is significantly reduced, since the known effects are now being accounted for in the variational equations. However, LAGEOS-2 has additional thermal accelerations that are difficult to model and require the remaining bit of cross-track process noise in order to pass the consistency test and have a realistic covariance.

2. Save the scenario file before proceeding.
3. Use the StateContent utility to verify that the following states are being estimated:

- satellite position and velocity
- satellite solar-radiation pressure
- individual ranging biases for each SLR facility

4. Rerun the filter to update the .filrun file. Notice that the filter runs more slowly. This is mainly due to the more complicated force modeling.
5. Rerun the smoother to update the .smtrun and .difrun files.

9 Assessing improvement

1. Repeat the Residual Analysis steps as above, creating a new Residual Ratio graph, Residual graph, and Residual Summary report. Are the residual RMSs about the means generally smaller than before? Are there any facilities for which the mean or RMS are now much bigger?
2. Repeat the Graphical Analysis. Are the overall position-consistency results improved by being closer to within ± 3? With the improved force modeling, things should be much better.

10 Improving results

1. On the Satellite object, turn off the cross-track unmodeled accelerations by setting:

   ForceModel.UnmodeledAccelerations.ProcessNoise.Frame = "Gaussian (RIC)"
   ForceModel.UnmodeledAccelerations.ProcessNoise.ZVelocitySigma = 0.0003 cm*sec^-1

2. Rerun the filter and smoother and create new Residual Ratios and Position Consistency graphs. The cross-track position consistency is consistently out of bounds, indicating that you likely still have a force model problem that was being mitigated when you had the additional cross-track process noise.
3. Turn on the empirical accelerations model to try to solve for and model the missing cross-track accelerations. Edit the ForceModel.EmpiricalForces properties and add a new item "RevRectifiedSinusoid1D" to the list. Accept the default values, but set

   ForceModel.EmpiricalForces(0).Direction = "CrossTrack".

   This will allow the filter to estimate a sinusoidal one-cycle-per-revolution acceleration in the cross-track axis. Rerun the filter and smoother.
4. Repeat the Graphical Analysis. Are the overall position-consistency results improved by being closer to within ± 3? Plot an Empirical Forces graph to see the magnitude of the solved for cross-track acceleration. Plot an "Empirical Forces Model Parameter" graph to see the solved for A_0, A_1, and A_2 terms in the force model.

EXTRA CREDIT:

Based on your analysis of the empirical force model parameters and their uncertainties, see if you can tighten up the model sigmas, error thresholds, and process noise (PN) step values.

Extend the fit span from 10 to 30 days by setting Filter.ProcessControl.TimeSpan = 30 days.

Tip: You can temporarily return to the operating system and make a backup copy of the scenario file while experimenting. If you make a mistake going forward, it may be easier to pick up your settings from the backup.