Keplerian Lambert Solver User Guide

Robyn Woollands & John Junkins
Texas A&M University
(robyn.woollands@gmail.com)

May 2016

Introduction

The Keplerian Lambert Solver (KLS) software is available in both MATLAB and C/C++ and is based on the Battin/Prussing [1,2] approach for solving the Keplerian elliptic Lambert problem over multiple revolutions. Parabolic and hyperbolic orbits are not considered. The KLS algorithm forms part of the Extremal Field Map (EFM) tool [3] in MATLAB and the Unified Lambert Tool (ULT) in C/C++ [4,5]. The ULT has the functionality to solver the perturbed Lambert's problem using high fidelity force models specified by the user.

The following command runs KLS where the inputs are ***initial state vector***, ***final state vector***, ***tolerance*** and an argument for ***plotting figures*** (YES: fig = 1, NO: fig = 0). For the multiple revolution problem 2*N+1 solutions exist, where N is the number of revolutions determined by the algorithm. If plotting is desired then all possible and feasible solutions are plotted. The position and velocity for each transfer is saved and the data is available to the user for further analysis.

```
>> keplerian_lambert_solver(r0,rf,tof,tol,fig)
```

Code Structure & Functions

keplerian_lambert_solver.m
This is the main function and all other functions are called from within this function or from sub-functions within this function.

consts.m
Loads some astrodynamics constants and also specifies the canonical distance and time units. All the computations are done using canonical units.

initialize.m
Initializes some variables for setting up the solution to Lambert's problem.

t_min_energy.m
This function computes the time-of-flight for the minimum energy transfer orbit.

t_parabolic.m
This function computes the time-of-flight for a parabolic orbit, which must be less than the desired transfer time for a solution to be computed.

n_max_newton.m
Determines the maximum number of possible revolutions over which Lambert's problem can be solved considering the user specified parameters. This function uses Newton's method, as specified by Prussing's paper [2]. If Newton's method fails then the bisection method (n_max_bisection.m) is used as a second attempt for solving the problem.

n_max_bisection.m
Determines the maximum number of possible revolutions over which Lambert's problem can be solved considering the user specified parameters. The bisection method is used only if Newton's method (n_max_newton.m) fails to converge.

a_transfer_newton.m
This function solves for the semimajor axis of the transfer orbit using Newton's method, as outlined in Prussing's paper [2]. If Newton's method fails then the bisection method (a_transfer_bisection.m) is used as a second attempt for solving the problem.

a_transfer_bisection.m
This function solves for the semimajor axis of the transfer orbit using the bisection method. This method is only utilized if Newton's method (a_transfer_newton.m) failed to converge.

velocity.m
This function computes the initial and final velocity of the transfer orbit.

retrograde.m
This function computes the z-component of the angular momentum vector. If it is negative then the orbit transfer was retrograde and the algorithm reruns the case using "theta = 2pi – theta" (see [2] for the definition of theta). Both the prograde and retrograde transfer orbits are saved and plotted if desired.

feasibility.m
This function checks the simulated transfer to determine if the transfer is physically possible. For example, trajectories that pass through the Earth are infeasible and are discarded.

FnG.m
The function computes the analytical F&G solution.

kepler.m
This function solves Kepler's equation.

store_data.m
This file builds data arrays as new data becomes available during the run.

store_data.m
This file saves the data arrays at the end of the run.

References

1. Battin, R., “An introduction to the mechanics and methods of astrodynamics”, AIAA education series, Restin, Virginia, 1999.

2. Ochoa, S., Prussing, J., “Multiple revolution solutions to Lambert’s problem”, AAS/AIAA spaceflight mechanics meeting, Kissimmee, 1992.

3. Woollands, R., Junkins, J., “Extremal Field Map User Guide”, Texas A&M University, College Station, Texas, 2016.

4. Woollands, R., Read, J., Hernandez, K., Probe, A., Junkins, J., “Unified Lambert Tool for Massively Parallel Applications in Space Situational Awareness”, Journal of Astronautical Sciences, accepted 2017.

5. Woollands, R., “Regularization and Computational Methods of Precise Solution of Perturbed Orbit Transfer Problems”, PhD Dissertation, Texas A&M University, College Station, Texas, 2016.