Implementing Low Thrust Combined Maneuvers via the Alfano Method

Colin Helms

Freelance Rocket Science

Author Note

This work is draft 1.4

23 May 2022

Abstract

This paper revisits the algorithm for long duration constant thrust circle to circle transfer first proposed in Edelbaum 1961 [1], as restated by Alfano and Wiesel, 1985 [2]. The method varies from Alfano and Wiesel in that it directly computes the control parameter, generating a table of controls as a function of orbit ratio over the range of Hamiltonian costates.

The Alfano and Wiesel derivation and resulting nomograph is included as an example of a low thrust control law in the textbook, Fundamentals of Astrodynamics and Applications by David A. Vallado [1] (pp. 382-388) and is important due to its influence on astrodynamics students. However, the treatment in the original paper and the text book is abbreviated and neither provides an intuitive basis for understanding the trajectory nor describes important details in the derivation; for instance, in the computation of elliptical integrals by Alfano and Wiesel the elliptic parameter ‘u’ is used rather than the elliptic modulus ‘k’ as in Edelbaum.

The development in this present paper returns to the original Edelbaum derivation in order to gain intuition, then completes provides an alternative derivation of controls, computing the inverse of the Alfano ‘Phi” function numerically rather than approximating it’s inverse as did the original paper. This alternate method enables direct computation of the thrust vector for each orbit in simulations of the Alfano and Wiesel family of trajectories. An ephemeris model covering a range of target orbit ratio is further developed to validate results by comparison with the famous Alfano and Wiesel Nomogram (Figure 1 of Alfano & Wiesel). To finalize the report a demonstration simulation is completed for transfers between Low Earth Orbit and Geosynchronous orbit using commercial Hall-Effect thrusters.

Keywords: Low-thrust, Geosynchronous, Orbit Transfer, Electric Propulsion, Inclination

Low Thrust Combined Maneuvers via the Alfano Method

Edelbaum and his team at United Aircraft derived the first closed form solution for continuous thrust circle-to-circle transfers [1]. Two decades later, Alfano and Wiesel extended the method, optimizing the Edelbaum yaw control law for many revolutions [2], other subsequent authors sought to improve the control law [3]. This paper seeks to implement the Alfano solutions in a software tool.

First, we revisit the original derivation with the objective of developing a direct computation of the Edelbaum/Alfano thrust controls.

# Initial Development of the Closed Form Solution

The Edelbaum team at United Aircraft started their development from the spherical form of the Lagrange planetary equations. Their 1961 paper surveyed a wide-ranging set of cases for satellite control, both impulsive and continuous thrust. It is worth noting that the Edelbaum development was very general in that the paper developed formulations for combined maneuvers changing any two orbital elements. Optimal combined maneuvers for circle-to-circle transfer using low-thrust propulsion were discussed in an appendix, for cases of change in eccentricity, altitude, and inclination. In this paper we are interested in combined maneuvers changing altitude and inclination.

The Edelbaum development proceeds by combining expressions for thrust vector components pitch () and yaw () with the Lagrangian equations of motion. With assumptions that time of the maneuver depends only upon the apogee radius (i.e. a circular orbit) and that the inclination change is small, Edelbaum proceeds to derive the steering law for producing the maximum change in inclination and semi-major axis (SMA) over the period of one orbit. The treatment yields the yaw steering angle to achieve the maximal combined altitude increase and inclination change.

The Edelbaum equations of motion are repeated in Equation 1 and Equation 2. These correspond to equations 18 in the original paper, where we have replaced thrust with specific acceleration.

Equation

Equation

Where:

* , starting semi-major axis
* , is specific acceleration, thrust divided by mass
* , is the initial circular velocity at start of the maneuver
* α, alpha, an in-plane thrust angle, known as pitch
* β, the out-of-plane thrust angle, known as yaw
* θ’, the angle along track, the sum of the argument of periapsis and the true anomaly

The thrust components are formulated in the Velocity-Normal-Binormal reference frame where unit vector is in the direction of the velocity vector, is in the direction of the angular momentum vector, and the binormal, is the cross-product .

Circular orbits with plane change should measure the along-track angle from the line of nodes, this is known as the Argument of Latitude, shown as in Edelbaum.

.

The thrust angle is the angle measured relative to the Normal axis and can have both in-plane (pitch) and out-of-plane (yaw) components. In-plane components are used for change of eccentricity or semi-major axis, and out-of-plane components affect inclination. The term , provides the tangential component of thrust and the term provides the out-of-plane component of thrust. The expression ensures that the resultant out-of-plane thrust is maximum at the line of nodes, and changes sign at the line of apsides.

Equation 3 shows the optimization can proceed by simply maximizing the change in orbital elements with change in pitch and yaw thrust angles and using an adjoint constraint for the amount if inclination change.

Equation

substituting a simple steering law,

And the following standard trigonometric substitutions,

The Edelbaum equations of motion then become,

Equation

Where the term under the radical can be recognized as the denominator of the elliptic integral. The control law becomes,

Equation

Note that if is the trigonometric substitution with , where is the maximum yaw angle for the revolution. When Equations 4 are integrated over one complete orbital period, Equation 6 and Equation 7 result.

Equation

Equation

Where:

* The complete elliptic integral of the first kind is, K(k).
* The complete elliptic integral of the second kind is, E(k).
* The argument k is the elliptic modulus.
* The square of the circular characteristic velocity = ,
* the semi-major axis at start of each revolution.

Using Equation 6 and Equation 7, solutions for k provide the optimum control angle. As the out-of-plane angle varies, inclination change is induced while the given change in semi-major axis is accomplished. Each value of lambda in Equation 3 provides a specific combination of altitude and inclination change for the circle-to-circle transfer in Edelbaum.

# Alfano & Wiesel Formulation

Alfano & Wiesel reformulated the Edelbaum equations for large changes in semimajor axis and inclination over many revolutions [2]. Their result selects a different optimum value for substitution into the control law, one which varies for change in semi-major axis per revolution.

Alfano and Wiesel factor the period out of Equation 6 and Equation 7 using,

Equation

Equation

Alfano and Wiesel have substituted = k in the Edelbaum equations, where k is the elliptic modulus and corresponds to the elliptic parameter [4]. This substitution alters the physical interpretation of the argument. Thus, Alfano and Wiesel restate the steering law Equation 5 as,

Equation 10

The denominator of Equation 10 is the form of an arctangent of an angle defined by . The parameter u is the sine of the maximum yaw thrust angle during the circle-to-circle transfer.

Equation 8 and Equation 9 are intended for use

over many revolution trajectories and the depletion of fuel affects the acceleration term.

Where the mass rate is, .

Alfano makes a differential substitution in order to remove the time dependence of the acceleration.

Where is the incremental change in velocity.

By direct substitution Equation 11 and Equation 12 result:

Equation

Equation

For compactness, Alfano bundles the elliptic integrals.

Equation

Equation

For direct calculation, the derivatives of these functions are needed,

Equation

Equation

The Hamiltonian is formed,

Equation

With the optimality condition:

Equation

Equation

These equations show that R(u) controls the amount of inclination change per orbit and P(u) controls the amount of change in orbit ratio. The quantity under the radical in Equation 18 is the Keplerian period and the like quantity in Equation 19 is the inverse of the characteristic velocity. This implies that P(u) has units of *1/sec* and R(u) has units of *km/sec*.

We can solve the value of lambda for a boundary condition of final inclination and final orbit ratio using the transversality conditions,

Equation

Equation

Equation 21 results since there is no dependency on inclination in the Hamiltonian, showing that is a constant. In addition, the Hamiltonian has no explicit dependency on τ.

Therefore, the Hamiltonian itself is a constant, and when the spacecraft reaches its final orbit, we may set .

Now using simple substitution, can be solved in terms of .

Equation

Equation 22 is used to find the constant value of for the transfer, where the values = k, the elliptic modulus in Equation 4.

Alfano simplifies Equation 22 by defining yet another function of u (note the sign change),

Equation

Alfano states that the function is monotonic, and its inverse function can be found, which given that is constant, will solve for the values of u proportional to the square root of orbit ratio, . One solution, given in the Appendix to [2] is to approximate the inverse using a Chebyshev polynomial. We choose a different approach.

# The Steering Law

By multiplying through, the steering law of Equation 10 can be re-arranged.

We can recognize the elliptic modulus, and the steering law becomes

From the definition of the elliptic integral,

where from Edelbaum,

Using a trigonometric identity, the steering law can be restated as,

Equation

Where is the arcsine of the maximum yaw angle in each orbit.

It still remains to find for each circle-to-circle transfer in this formulation, however Equation 24 provides the mental image that the tangent of the maximum yaw angle at any given incremental circular radius is modulated by the argument of latitude.

In the Edelbaum formulation, the transfer takes place between an initial circular orbit and a final circular orbit. Thus, there is a single pseudo-optimum steering angle over the entire transfer. In Alfano there is the assumption of many circle-to-circle transfers, and though the adjoint constrain for inclination, is also found to be constant, each circle-to-circle transfer has a different steering angle. The Alfano approach increases sophistication by optimizing a Hamiltonian formulation with adjoint constraints in both SMA and inclination. This means that the elliptic modulus, , in Equation 8 and Equation 9 has different solutions between Edelbaum and Alfano, yet still it is clear that identifies an angle which defines the optimum thrust vector. In Alfano and Wiesel, the must change in proportion to , and is the proportionality constant.

# Finding the Trajectory Directly

The quantity needed to control the thrust vector is u, the argument of . This suggests simply calculating for a series of linearly spaced u values. It is only necessary to compute the array of once; these values correspond to the values of for orbit ratio =1 and are referred to as canonical. The complete range of values of may be computed by multiplying with the inverse of the orbit ratio. Given the vector of computed these may be stored in a table and arrays of u created by sorting on the canonical values of . The sorted table effectively represents the inverse phi function.

Computing involves computing the complete elliptic integrals of the first and second kind, which may be done with the python SciPy special library[[1]](#footnote-1).

from scipy import special

special.ellipk(u)

special.ellipe(u)

Figure 1 shows the result of this computation for u = 0.1 to 1. The starting value of 0.1 is used to avoid another singularity in the elliptic integrals, which skews the plot magnitude.

|  |
| --- |
|  |

Figure , Plot of Complete Elliptic Integrals for u from 0 to 1

The Φ function defined by Alfano also requires that the derivatives of the first and second complete elliptic integral be taken.

Formulas for elliptic integral derivatives are found in the NIST Downloadable Library of Math Functions (DLMF) [5]. The form of the derivatives for K and E in the DLMF use the elliptic modulus as an argument, thus these formulas must be multiplied by ½, which results from taking the derivative of . The formulas used are as follows.

Equation

Equation

Equation 25 and Equation 26 can be used to compute and .

Equation

Equation

Plots of these functions are shown in Figure 2 and Figure 3.

Equation 27 and Equation 28 can be used to compute for a series of from 0 to 1 per Equation 23. A linear array of u is generated to four decimal place precision using the numpy linspace library function.

u = np.round(0.1 \* np.linspace(1, 10, ncols, endpoint=False), 4)

We exclude the endpoint in this array because the singularity in the recurring expression (1 - u) causes scaling difficulties. The resulting plot of Φ(u) is shown in Figure 4.

|  |
| --- |
| Chart, line chart  Description automatically generated |

Figure , Plots of R(u) and dR/du

|  |
| --- |
| Graphical user interface  Description automatically generated with medium confidence |

Figure , Plots of P(u) and dP/du

|  |
| --- |
| A picture containing chart  Description automatically generated |

Figure , Inverse Φ(u) over the Domain of u

The values of the costates are formed as combinations of the reciprocal of Φ(u) and the reciprocal square root of the orbit ratio and is shown in Figure 5.

The trajectory for any given value of can be visualized as a plane parallel to the u, R axes cutting through the Φ surface at a vertical offset equal to . Starting at a value of -0.496 Figure 5 shows the trajectory is cutoff at the right edge, where the u value approaches 1. This agrees with Alfano and Wiesel’s original figure as reprinted in Vallado Figure 6-24 [6], which shows that trajectories for are predominately inclination change.

Another observation is that for values of more positive than -0.3245 the trajectory is cutoff at the left edge where the u values are less than 0.2. This indicates that the optimal trajectory does not start until an orbit ratio > 1 has been achieved. This agrees with Vallado Figure 6-24 which shows trajectories less than this are predominately altitude change.

|  |
| --- |
| Chart  Description automatically generated |

Figure , Alfano Costates by Orbit Ratio

In our method, the end of the trajectory is easily identified because the number of u values associated with any canonical value of that are more negative than -0.4943 will simply run out at some value of orbit ratio less than 10. We mime the behavior of the Alfano and Wiesel in our code by initializing to 0 the rows of the table in which we store the values of u. The table is then overwritten with calculated values of u, leaving the 0 value as residue where no u value is associated with the costate. With this artifice, the control program simply returns a yaw angle of 0, implying that inclination change is complete, the thrust continues to raise the orbit with no inclination change.

The control program using this approach provides yaw angles over the range of orbit ratio 1.1 to 6.6 for various values of costate as shown in Figure 6.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |

Figure , Steering Angles from Costates

The approach for validation of the direct computation is to execute multiple trajectory simulations using the Goddard Mission Analysis Toolkit (GMAT) [7] which provides an Ephemeris model. The GMAT mission model is simple, shown in Figure 6. With the objective of obtaining computed values as similar to Alfano and Wiesel as possible, the eclipse model is turned off and there is no drag or Solar Radiation Pressure model used. The spacecraft model is a 4.5mT, 128kW SEP vehicle using an aggregate of Hall Effect Thrusters providing 6N thrust. Multiple models are executed in batch for orbit ratio 1.5 to 10 in increments of 0.5 and costates from -0.1 to -1.56 in increments of 0.01. The mission is executed starting at 0 degrees inclination, and the resulting final inclination is the maximum achievable orbit ratio and inclination for the given costate. The initial fuel mass is 1855kg which is precalculated to be sufficient for the theoretically maximum delta-v at orbit ratio 10.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37 | BeginMissionSequence;  Propagate 'Propagate to periapsis' TrimPropagator(EOTV) {EOTV.Earth.Periapsis};  BeginFiniteBurn 'GEOTransfer' ContinuousThrust(EOTV);  GMAT SMA\_INIT = EOTV.SMA;  GMAT REV\_LAST = EOTV.Earth.OrbitPeriod;  GMAT T0\_AT\_REV = EOTV.ElapsedSecs;  GMAT REV = REV + 1;  While EOTV.SMA <= SMA\_END    GMAT AOL = EOTV.TA + EOTV.AOP;  If AOL > 360.0  GMAT AOL = AOL - 360.0;  EndIf;    GMAT T\_REV = EOTV.ElapsedSecs - T0\_AT\_REV;    If T\_REV >= REV\_LAST  GMAT REV\_ERR = T\_REV - REV\_LAST;  GMAT REV\_LAST = EOTV.Earth.OrbitPeriod;  GMAT T0\_AT\_REV = EOTV.ElapsedSecs - REV\_ERR;  GMAT REV = REV + 1;    GMAT [CONTROL] = Python.YawAngles.get\_control\_onrev\  (COSTATE, AOL, EOTV.Earth.SMA, SMA\_INIT, MORE);  EndIf;    GMAT EOTV.HET1.ThrustDirection1 = CONTROL(1,1);  GMAT EOTV.HET1.ThrustDirection2 = CONTROL(1,2);  GMAT EOTV.HET1.ThrustDirection3 = CONTROL(1,3);    Propagate 'Propagate Steps' DefaultProp(EOTV);  EndWhile;  EndFiniteBurn 'GEOTransfer' ContinuousThrust(EOTV); |

Figure , Mission Model

A JSON control table containing rows of u values in order of orbit ratio and organized by keys of canonical is populated by a Python script generatecontrols.py which computes the elliptic integrals and their derivatives. A slice of the control table is shown as Table 1. Note that where the left edge of the plot occurs in Figure 5, the default value of 0 is shown.

Table , Summary of Control Table Output

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Orbit R** | **Costate Values** | | | | | | | | | |
| **-0.3666** | **-0.368** | … | **-0.4953** | **-0.4962** | … | **-1.5391** | **-1.5454** | **-1.5511** | **-1.557** |
| **1** | 0.1 | 0.1006 | … | 0.1673 | 0.1679 | … | 0.9969 | 0.9976 | 0.9982 | 0.9988 |
| **1.01** | 0.1008 | 0.1014 | … | 0.169 | 0.1695 | … | 0.9977 | 0.9984 | 0.999 | 0 |
| **1.02** | 0.1016 | 0.1022 | … | 0.1705 | 0.1711 | … | 0.9985 | 0.9992 | 0 | 0 |
| **1.03** | 0.1024 | 0.103 | … | 0.2239 | 0.2245 | … | 0.9993 | 0 | 0 | 0 |
| … | … | … | … | … | … | … | … | … | … | … |
| **9.97** | 0.9073 | 0.9091 | … | 0.9992 | 0 | … | 0 | 0 | 0 | 0 |
| **9.98** | 0.9076 | 0.9094 | … | 0.9992 | 0 | … | 0 | 0 | 0 | 0 |
| **9.99** | 0.9078 | 0.9095 | … | 0.9993 | 0 | … | 0 | 0 | 0 | 0 |
| **10** | 0.908 | 0.9098 | … | 0.9994 | 0 | … | 0 | 0 | 0 | 0 |

In the mission sequence, Figure 6, at line 25, a call to Python is made at the beginning of each orbit period. This call reads the *u* value from the JSON control table represented by Table 1, and computes both the yaw and pitch thrust vector per Equation 10. The GMAT variable CONTROL(1,1) is the returned pitch component of the thrust vector, and the returned yaw component is CONTROL(1,2). The CONTROL(1,3) element would be used for a roll component should it be necessary to compensate for solar beta, but is clamped to zero: the default solar power model in GMAT is simple and does not take solar beta angle into account.

# Results

|  |
| --- |
| Figure , Alfano Geosynchronous Trajectory |

Following the execution of the mission script “Alfano\_Test\_R12” the output report file recorded a Geosynchronous transfer from a 6878.136 km 28.5 deg initial SMA and inclination to 38280km and 0.2 deg inclination in 7074 revolutions and an elapsed time of 21days. The spacecraft started with a dry mass of 1024.5 kg and fuel mass of 2600kg, thrust was 11.28N at 1977sec Isp. The final mass of Xenon was 1679.5 kg of fuel remaining.

# Works Cited

|  |  |
| --- | --- |
| [1] | T. N. Edelbaum, "Propulsion Requirements for Controllable Satellites," *ARS Journal,* pp. 1079 - 1089, 1961. |
| [2] | A. a. Wiesel, "Optimal Many-Revolution Orbit Transfer," *Journal of Guidance and Control,* pp. 155-157, 1985. |
| [3] | C. Casolino, "AIAA-2004-5402, Improved Edelbaum Approach to Optimize LEO GEO Low-Thrust Transfers," in *AIAA/AAS Astrodynamics Specialist Conference*, Providence, Rhode Island, 2004. |
| [4] | E. W. Weisstein, "Elliptic Modulus."," 03 Jul 2019. [Online]. Available: http://mathworld.wolfram.com/EllipticModulus.html. |
| [5] | F. W. J. Olver and A. B. O. Daalhuis, "NIST Digital Library of Mathematical Functions.," 15 06 2019. [Online]. Available: htto:///dlmf.nist.gov. [Accessed 22 February 2018]. |
| [6] | D. A. Vallado, Fundamentals of Astrodynamcis and Applications, 4th ed., Hawthorne, CA: Microcosm Press, 2013, pp. 381 - 388. |
| [7] | NASA Goddard Spaceflight Center, "GMAT Central," 22 Nov 2012. [Online]. Available: http://gmatcentral.org/. [Accessed 31 May 2018]. |
| [8] | R. A. Broucke, "Low-Thrust Trajectory Optimization in an Inverse Square Force Field," in *AAS Paper 91-159*, Houston, 1991. |
| [9] | E. W. Weisstein, ""Complete Elliptic Integral of the First Kind."," 03 Jul 2019. [Online]. Available: http://mathworld.wolfram.com/CompleteEllipticIntegraloftheFirstKind.html. |
| [10] | E. W. Wesstein, "Elliptic Integral of the Second Kind.," 03 Jul 2019. [Online]. Available: http://mathworld.wolfram.com/EllipticIntegraloftheSecondKind.html. |

1. Wolfram Mathematica also provides these functions and is used as a check on results. [↑](#footnote-ref-1)