THEORETICAL FOUNDATION OF COPERNICUS: A UNIFIED SYSTEM FOR TRAJECTORY DESIGN AND OPTIMIZATION

Cesar Ocampo[1], Juan S. Senent[2], and Jacob Williams[3]

[1]*Department of Aerospace Engineering & Engineering Mechanics, The University of Texas at Austin, Austin, Texas*
[2]*Odyssey Space Research, Houston, Texas*
[3]*ERC, Inc. Engineering and Science Contract Group, Houston, Texas*

ABSTRACT

The fundamental methods are described for the general spacecraft trajectory design and optimization software system called Copernicus. The methods rely on a unified framework that is used to model, design, and optimize spacecraft trajectories that may operate in complex gravitational force fields, use multiple propulsion systems, and involve multiple spacecraft. The trajectory model, with its associated equations of motion and maneuver models, are discussed.

Key words: Copernicus, Spacecraft Trajectory Optimization Software.

1. INTRODUCTION

Copernicus [17] is a general trajectory design and optimization system intended to solve a wide range of spacecraft trajectory problems in a robust and efficient manner. It facilitates the design and optimization of simple to complex spacecraft trajectories. Some examples from recent Project Constellation studies include: trans-lunar [3], trans-Earth [16] and trans-lunar abort return trajectories [14]. Interplanetary mission applications include: design of a 10 year 32-asteroid tour using a low-thrust propulsion system, Mars sample return missions and trajectory design for the Lunar Crater Observation and Sensing Satellite (LCROSS).

The three main problems addressed by the system are:

1. Modelling and Open Loop Simulation: The system can simulate open loop trajectories in any force field, using one or more spacecraft, and including impulsive and/or finite burn maneuvers.
2. General Targeting via Nonlinear Root Finding: The system can search for the values of the independent variables required to satisfy a set of constraint functions. This is required for trajectory targeting problems, such as orbital boundary value problems. But more generally, it is possible to formulate and solve general systems of nonlinear equations that depend on a preselected set of independent variables.
3. Optimization: The system can extremize the value of a function consistent with the constraints. Any computable function can be extremized. Typically, the optimization mode of the system is used for maneuver design and optimization. This includes determining the times and parameters that describe the maneuvers, which can be impulsive or finite burns.

A key component of the system is the *trajectory model* that describes how a complete trajectory for a single or set of spacecraft is modelled. The supporting components of the trajectory model include the equations of motion, the propulsion system, the control parametrization, the independent variables, and the constraint functions. This paper describes and summarizes some of the theoretical aspects of these components. Some of these along with the details of the system architecture have been documented previously as the development of the system has evolved over the years[8, 9, 10]. A discussion on future extensions and proposed formulations for the finite burn maneuver modeling is also given.

2. TRAJECTORY MODEL

The trajectory model uses a building block called a *segment*. A segment is a trajectory arc that can have velocity impulses and/or a finite burn maneuver. Depending on how the relationship between different segments is defined, single or multiple spacecraft trajectories that may interact can be modeled. The generality of how segments are related to one another facilitates the modeling of complex missions. The simplest mission may have only one segment. Complex missions may have tens or hundreds of segments.

The segment arc is bracketed by two node points. The node points are tagged with epochs that are referenced to a reference epoch, t_{epoch}. The epoch of the initial node is t_0 and the epoch of the final node is t_f. There is no restriction on the values for t_0 and t_f ($t_0 = t_f, t_0 < t_f$, $t_0 > t_f$). The segment duration is $\Delta t = t_f - t_0$. The segment nodes are uniquely defined by specifying any of the pairs $(t_0, t_f), (t_0, \Delta t), (\Delta t, t_f)$.

The segment state vector $\mathbf{x}$ is

$$\mathbf{x} = \begin{pmatrix} \mathbf{r}^\top & \mathbf{v}^\top & m \end{pmatrix}^\top . \tag{1}$$

The initial position vector is $\mathbf{r}_0$, the initial velocity vector is $\mathbf{v}_0^-$, and the initial mass is m_0^{--}. Both $\mathbf{v}$ and m can have discontinuities at the initial node. For velocity, the discontinuity is an impulsive maneuver $\Delta\mathbf{v}_0$. For mass, the discontinuities are due to the propellent mass consumed by the impulsive maneuver, and independent mass discontinuities before and after the impulsive maneuver. The independent mass discontinuities can be used to represent acquisition or removal of other spacecraft, and/or stage masses. After these discontinuities are accounted for, the segment is propagated from t_0 to t_f if $t_0 \neq t_f$. At the t_f node, velocity and mass have the same type of discontinuities as at the t_0 node. The propagation direction is considered as forward if $t_0 < t_f$ and backward if $t_0 > t_f$. If $t_0 = t_f$, then the propagation direction is specified to properly account for how the velocity and mass discontinuities are used. Let $i_d = +1$ for a forward propagation and $i_d = -1$ for a backward propagation.

Across either the t_0 or the t_f node, position remains constant. The velocity at the end of the node is

$$\mathbf{v}^+ = \mathbf{v}^- + (i_d)\,\Delta\mathbf{v} \tag{2}$$

The mass across the node evolves as

$$m^{--} \to m^{-+} \to m^{+-} \to m^{++}$$

Any one of these is specified, and the remaining three are determined by knowing the propagation direction, the independent mass discontinuity Δm^- before the impulse, the independent mass discontinuity Δm^+ after the impulse, the magnitude of the impulsive maneuver, and the exhaust velocity c of the engine doing the impulsive maneuver. Define the factor

$$f = e^{-\Delta v / c} \tag{3}$$

There are two main cases:

1. m^{--} or m^{-+} is specified; depending on which one is specified, the other is solved by using

$$m^{-+} = m^{--} + \Delta m^- \tag{4}$$

then

$$m^{+-} = \left(m^{-+}\right)(f)^{i_d} \tag{5}$$

$$m^{++} = m^{+-} + \Delta m^+ \tag{6}$$

2. m^{+-} or m^{++} is specified; depending on which one is specified, the other is solved by using

$$m^{++} = m^{+-} + \Delta m^+ \tag{7}$$

then

$$m^{-+} = \left(m^{+-}\right)(f)^{-i_d} \tag{8}$$

$$m^{--} = m^{+-} - \Delta m^- \tag{9}$$

As stated before, the segment initial state vector is

$$\mathbf{x}_0^- = \begin{pmatrix} \mathbf{r}_0^\top & \mathbf{v}_0^{-\top} & m_0^{--} \end{pmatrix}^\top . \tag{10}$$

After accounting for the potential velocity and mass discontinuities at the t_0 node, the initial condition for the state vector that is propagated from t_0 to t_f (if $t_0 \neq t_f$) is

$$\mathbf{x}_0^+ = \begin{pmatrix} \mathbf{r}_0^\top & \mathbf{v}_0^{+\top} & m_0^{++} \end{pmatrix}^\top . \tag{11}$$

The state vector after the propagation to t_f is

$$\mathbf{x}_f^- = \begin{pmatrix} \mathbf{r}_f^\top & \mathbf{v}_f^{-\top} & m_0^{--} \end{pmatrix}^\top . \tag{12}$$

The final value of the state vector after all possible velocity and mass discontinuities are accounted for at the t_f node is

$$\mathbf{x}_f^+ = \begin{pmatrix} \mathbf{r}_f^\top & \mathbf{v}_f^{+\top} & m_f^{++} \end{pmatrix}^\top . \tag{13}$$

Sets of segments are used to construct complete trajectories for one or more spacecraft. Segments can be connected to other segments by forcing the complete state or a subset of it to inherit the complete state or a subset of it from either node of another segment. Segments can also be completely disconnected to model independent trajectories representing other spacecraft, for example. These segments can later be constrained to be connected via continuity constraints to other segments if needed. A *mission* consists of the set of all segments.

A mission has a *base frame* which is required to be non-rotating and centered at a celestial body. All state transformations use the base frame as a hub through which all state, maneuver, and state function transformations are made. Each segment has a *state input frame*, a *propagation frame*, and a *function output frame*. The propagation frame is the frame used for the integration of the equations of motion. It is required to be non-rotating and centered at a celestial body. The initial position and velocity state data $(\mathbf{r}_0, \mathbf{v}_0^-)$ is given in the state input frame. The function data for a segment includes all functions that can be computed from the distinct segment states ($\mathbf{r}_0, \mathbf{r}_f, \mathbf{v}_0^-, \mathbf{v}_0^+, \mathbf{v}_f^-, \mathbf{v}_f^+$, and so on) and are computed in the function output frame. The state input and function output frames are required to be centered at a celestial body and can be either fixed or rotating. Each segment is allowed to have its own state input, propagation, and function output frame. A simple mission may have all of these frames be the same.

The individual segment impulsive maneuvers are each referenced to an *impulsive maneuver frame*. The impulsive maneuver frame can be either fixed (non-rotating) or osculating along the instantaneous trajectory path. In either case, the impulsive maneuver frame is referenced to a frame centered at a celestial body. A fixed impulsive maneuver frame is typically the same as the segment propagation frame. Alternatively, an example of a common osculating impulsive maneuver frame is one that uses the instantaneous velocity vector as a basis vector. The remaining two basis vectors can be constructed, for example, by choosing the instantaneous angular momentum vector, if it exists, as one of the basis vectors.

A segment vector $\mathbf{s}^i$ is defined for each segment i to be a mixed variable type vector containing at least all of the independent and dependent variables and functions associated with the segment. Additionally it contains all the frame definitions, how and what data is inherited from other segments, what functions need to be evaluated from the state vector at the distinct time tags, which variables are search variables, and so on. This list of information is large. As an example, assume that a segment is purely ballistic with impulsive maneuvers at both nodes. A subset of the segment vector $\mathbf{s}^i$ is

$$\mathbf{s}^i = \begin{pmatrix} t_0, \mathbf{r}_0^{\top}, \mathbf{v}_0^{-\top}, m_0^{--}, \Delta m_0^{-}, \Delta \mathbf{v}_0^{\top}, \Delta m_0^{+}, \\ t_f, \Delta m_f^{-}, \Delta \mathbf{v}_0^{\top}, \Delta m_f^{+}, \mathbf{r}_f, \mathbf{v}_f^{++}, m_f^{++}, ... \end{pmatrix}^{\top}. \tag{14}$$

The complete mission with n segments can be computed uniquely in open loop (simulated) by processing all of the information contained in $\mathbf{s}^i$ $(i = 1, ..., n)$.

Every segment i has a variable vector $\mathbf{s}_x^i$ and a function vector $\mathbf{s}_f^i$. $\mathbf{s}_x^i$ is a subset of $\mathbf{s}^i$ and contains only the variables that have been identified to be independent search variables to be determined. $\mathbf{s}_f^i$ is a subset of $\mathbf{s}^i$ and contains only the functions that have been identified to be constrained or added to an objective function.

The state vector for any segment is governed by the first-order vector equation of motion

$$\frac{d}{dt}\begin{pmatrix} \mathbf{r} \\ \mathbf{v} \\ m \end{pmatrix} = \begin{pmatrix} \mathbf{v} \\ \mathbf{g}(\mathbf{r}, \mathbf{v}, m, t, \mathbf{p}) + \frac{T(t)}{m}\mathbf{u}(t) \\ -\frac{T(t)}{c(t)} \end{pmatrix}^i \tag{15}$$

where $\mathbf{g}$ is the ballistic acceleration per unit mass, $\mathbf{p}$ is a vector of non-state or non-control parameters such as force model and vehicle parameters, T is the finite engine thrust force, $\mathbf{u}$ is the unit thrust direction, and c is the engine exhaust velocity. T, c, and $\mathbf{u}$ are possible control functions. The information needed to evaluate these functions are also contained in $\mathbf{s}^i$. These equations are solved as an initial value problem from t_0 and t_f. The only allowable discontinuities along the integrated solution are associated with the control terms and only if an optimal control formulation is used to determine the values of the control functions.

The natural ballistic acceleration vector $\mathbf{g}$ is expressed in the segment propagation frame which is centered on a main celestial body c_b and is non rotating. An example expression for $\mathbf{g}$ is

$$\begin{aligned} \mathbf{g}(\mathbf{r}, t) = & -\frac{G m_{c_b}}{r^3}\mathbf{r} \\ & -G\sum_{j=1}^{n_b} m_j \left(\frac{\mathbf{r} - \mathbf{r}_j(t)}{|\mathbf{r} - \mathbf{r}_j(t)|^3} + \frac{\mathbf{r}_j(t)}{r_j^3(t)} \right) \\ & + \mathbf{a}_{drag} + \mathbf{a}_{srp} + \mathbf{a}_{non-spherical} \end{aligned} \tag{16}$$

where G is the universal constant of gravitation, m_{c_b} is the mass of the main celestial body, n_b is the number of additional celestial bodies, and m_j is the mass of celestial body j. It is assumed that the positions of the n_b celestial bodies is known as an explicit function of time with respect to c_b. Depending on the problem, additional terms accounting for other common accelerations, such as atmospheric drag($\mathbf{a}_{drag}$), solar radiation pressure($\mathbf{a}_{srp}$), and non-spherical celestial bodies($\mathbf{a}_{non-spherical}$), are added to the vector function $\mathbf{g}$. The equations of motion are integrated numerically with an explicit numerical integration algorithm. The trajectory arc that connects the t_0 node to the t_f node is either ballistic or it is a finite burn arc. If it is a finite burn arc, one of two main finite burn models are used. These are described below.

3. FINITE BURN MANEUVER MODELS

The modelling of a finite burn arc is based on one of two possible models. The first model is the parameter model that uses a finite set of parameters to describe the time evolution of the control variables. This finite set of parameters form part of $\mathbf{s}_x^i$. The second model is the optimal control model which requires augmenting the state vector to include the costate vector which is the vector of Lagrange multipliers adjoined to the state. The initial conditions of the costate vector, or the initial values of a transformation vector used to uniquely determine the initial costate vector, form part of $\mathbf{s}_x^i$.

3.1. Simple Engine Model

The following discussion refers to the trajectory arc for a particular segment between t_0 and t_f $(t_0 \le t \le t_f)$. The finite burn acceleration is given by the thrust acceleration term

$$\mathbf{a}_{thrust} = \frac{T(t)}{m(t)}\mathbf{u}(t) \tag{17}$$

The mass rate is

$$\dot{m} = \frac{-T(t)}{c(t)} \tag{18}$$

and the minimal set of control constraints are

$$T_{\min} \le T(t) \le T_{\max} \tag{19}$$

$$|\mathbf{u}(t)| = 1\,. \tag{20}$$

$$c(t) > 0 \tag{21}$$

The type of engine being modelled determines which of these are the controls. A constant exhaust velocity engine that can only be turned on and off with $T = T_{\max}$ when it is on has $c(t) = c$ and T is piecewise constant (0 or $T_{\max}$). If the engine can be throttled then $T(t)$ needs to be determined consistent with its lower and upper bounds. A simple model assumes that $T(t) = T$, a constant, with $0 \leq T \leq T_{\max}$.

An engine that is limited by the input power constrains $T(t)$ and $c(t)$ to satisfy

$$P(t) = \frac{1}{2}T(t)c(t) \tag{22}$$

with

$$P_{\min} \leq P(t) \leq P_{\max} \tag{23}$$

along with bound constraints on either $T(t)$

$$T_{\min} \leq T(t) \leq T_{\max} \tag{24}$$

or $c(t)$

$$c_{\min} \leq c(t) \leq c_{\max} \tag{25}$$

In this model, any pair $(T(t), P(t))$, $(T(t), c(t))$, $(c(t), P(t))$ are the control variables. The power P can be obtained by an internal source or by external source such as a solar electric system and will depend in part on the radial distance from the Sun. Optimal control theory has been used to determine the values of the controls that affect directly the thrust acceleration magnitude, namely T, c, P, for this system[9].

3.2. Detailed Engine Model

For the detailed engine model the controls can be Thrust (T), specific impulse (I_{sp}), exhaust velocity (c), Power (P), mass flow rate ($\dot{m}$), Thrust acceleration (T_{acc}), or Thrust over weight (T/W). In addition, the efficiency (η) of the engine can be modeled which relates the input power (P_{input}) to the power available to the engine. Any valid combination of the engine controls can be selected (e.g., Thrust and I_{sp}, Power and c, etc.) Any of the control parameters can be specified as constants, or can be modeled using selected engine control laws. The selected engine controls must satisfy the equations:

$$\begin{aligned}
\eta &= \frac{P}{P_{input}} && (26)\\
P &= \frac{1}{2} \cdot T \cdot c && (27)\\
\dot{m} &= -T/c && (28)\\
c &= I_{sp} \cdot g_0 && (29)\\
T_{acc} &= T/m && (30)\\
T/W &= T/(m \cdot g_0) && (31)
\end{aligned}$$

The efficiency of an electric propulsion systems can be modeled as a function of specific impulse using the equation:

$$\eta = \frac{BB \times I_{sp}^2}{I_{sp}^2 + DD^2} \tag{32}$$

where BB and DD are specified coefficients. Alternately, polynomial expressions for mass flow rate and thrust as a function of power can be used:

$$\begin{aligned}
\dot{m} &= c_f \cdot [c_m(1) + c_m(2) \cdot P + c_m(2) \cdot P^2 && (33)\\
&\quad + c_m(4) \cdot P^3 + c_m(5) \cdot P^4] \\
T &= c_f \cdot [c_t(1) + c_t(2) \cdot P + c_t(2) \cdot P^2 && (34)\\
&\quad + c_t(4) \cdot P^3 + c_t(5) \cdot P^4]
\end{aligned}$$

Where c_f and c_t are coefficient arrays for modeling LILT effects and solar panel degradation, respectively; c_f is the duty cycle fraction.

The engine can use an internal power source, or be modeled as a solar electric engine. For a solar electric engine, a custom solar panel power model can be used:

$$\begin{aligned}
P &= \phi \cdot P_0 \cdot [\frac{g(1) + g(2)/r + g(3)/r^2}{1 + g(4) \cdot r + g(5) \cdot r^2} \frac{1}{r^2}] && (35)\\
&\quad \cdot [t(1) + t(2) \cdot e^{t(3) \cdot \Delta t} + t(4) \cdot \Delta t]
\end{aligned}$$

Where $\mathbf{g}$ and $\mathbf{t}$ are coefficient arrays, Δt is the accumulated time the vehicle has been in space, P_0 is the reference power at 1 AU at the start of the mission, r is the distance from the Sun, and ϕ is the fraction of the Sun that is visible. A variety of eclipse models can be used, from a simple cylindrical model to a complex model that includes the umbra, penumbra, and antumbra. The accumulated time in space can be tracked across multiple segments, allowing for multiple coast and thrust arcs in the mission.

The remaining discussion is limited to discussing the computation of the thrust direction unit vector $\mathbf{u}(t)$.

3.3. Parameter Steering Model

The thrust direction unit vector is parameterized by two spherical angles referred to the *finite burn control frame*. This frame can be the same as the segment propagation frame, but it can also be an osculating frame such as one that uses the instantaneous velocity vector as a basis direction. In terms of the spherical angles,

$$\mathbf{u}(t) = \begin{pmatrix} \cos\alpha(t)\cos\beta(t) \\ \sin\alpha(t)\cos\beta(t) \\ \sin\beta(t) \end{pmatrix} \tag{36}$$

The angles are time functions of the form

$$\begin{aligned}
\alpha(t) &= \alpha_0 + \dot{\alpha}_0(t - t_0) + \frac{\ddot{\alpha}_0(t - t_0)^2}{2} \\
&\quad + a_\alpha \sin(\omega_\alpha(t - t_0) + \phi_\alpha) && (37)\\
\beta(t) &= \beta_0 + \dot{\beta}_0(t - t_0) + \frac{\ddot{\beta}_0(t - t_0)^2}{2} \\
&\quad + a_\beta \sin(\omega_\beta(t - t_0) + \phi_\beta) && (38)
\end{aligned}$$

where ($t_0 \leq t \leq t_f$). Each of the constants in Eqs. 37-38 can be part of $\mathbf{s}_x^i$. As defined, the functions for the spherical angles α and β admit constant, linear, quadratic, and sinusoidal terms. For most practical applications, the constant and linear terms are sufficient. For longer finite burn durations, both the quadratic and the sinusoidal terms may be needed.

A more general model that attempts to emulate the potential complexity that the steering of a finite burn can exhibit is referred to as the Single Axis Rotation (SAR) Model. In this model a reference unit vector $\mathbf{u}_{ref}$ is rotated about a rotation axis $\mathbf{n}$ through a time varying angle γ. The reference unit vector $\mathbf{u}_{ref}$ is referenced to a control frame that in turn is defined by basis vectors defined in a frame centered at a given celestial body c_b. The spherical angles that define the rotation vector $\mathbf{n}$ and the rotation angle γ are functions of time similar to the time functions given by Eqs. 37-38. A special case exists if $\mathbf{u}(t)$ is confined to a plane normal to the rotation axis $\mathbf{n}$; here the new reference unit vector $\mathbf{u}'_{ref}$ that gets rotated is

$$\mathbf{u}'_{ref} = \frac{(\mathbf{n} \times \mathbf{u}_{ref}) \times \mathbf{n}}{|(\mathbf{n} \times \mathbf{u}_{ref}) \times \mathbf{n}|}$$

Depending on the parameters used to define $\mathbf{u}_{ref}$, $\mathbf{n}$ and γ, $\mathbf{u}(t)$ can be constant, or it can rotate at a constant rate in plane normal to $\mathbf{n}$, or it can precess and nutate relative to a cone whose central axis is the rotation axis $\mathbf{n}$. In general, the unit vector thrust direction is calculated:

$$\begin{aligned}\mathbf{u}(t) &= \mathbf{u}_{ref}(t)^{\top}\mathbf{n}(t)\,(1-\cos\gamma(t))\,\mathbf{n}(t) \\ &\quad + \cos\gamma(t)\mathbf{u}_{ref}(t) + \sin\gamma(t)\,(\mathbf{n}(t) \times \mathbf{u}_{ref}(t))\end{aligned}$$

3.4. Optimal Control Steering Model

In the Mayer form of the optimal control problem[1, 4] where the cost function to maximize is of the form

$$J = \phi(t_0, \mathbf{x}_0, t_f, \mathbf{x}_f) \tag{39}$$

the optimal control Hamiltonian is

$$H = \boldsymbol{\lambda}^{\top}\mathbf{f} \tag{40}$$

where $\boldsymbol{\lambda}$ is the costate vector of Lagrange multipliers adjoined to the state vector and $\mathbf{f}$ are the state equations,

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t, \mathbf{u}_c) \tag{41}$$

and $\mathbf{u}_c$ is a vector of control variables. A first-order necessary condition is the vector differential equation for $\boldsymbol{\lambda}$

$$\dot{\boldsymbol{\lambda}} = -\left(\frac{\partial H}{\partial \mathbf{x}}\right)^{\top} \tag{42}$$

Additionally, the control vector $\mathbf{u}_c$ is chosen to extremize H at all points on a solution consistent with the control constraints if they are present. The explicit costate vector is

$$\boldsymbol{\lambda} = \begin{pmatrix} \boldsymbol{\lambda}_r^{\top} & \boldsymbol{\lambda}_v^{\top} & \lambda_m \end{pmatrix}^{\top}. \tag{43}$$

For the force model considered,

$$\mathbf{f}(\mathbf{x}, t, \mathbf{u}_c) = \begin{pmatrix} \mathbf{v} \\ \mathbf{g}(\mathbf{r}, \mathbf{v}, m, t, \mathbf{p}) + \frac{T(t)}{m}\mathbf{u}(t) \\ -\frac{T(t)}{c(t)} \end{pmatrix} \tag{44}$$

the Hamiltonian is

$$H = \boldsymbol{\lambda}_r^{\top}\mathbf{v} + \boldsymbol{\lambda}_v^{\top}\left(\mathbf{g}(\mathbf{r}, \mathbf{v}, m, t, \mathbf{p}) + \frac{T}{m}\mathbf{u}\right) + \lambda_m\left(-\frac{T}{c}\right) \tag{45}$$

and the costate vector equations are

$$\dot{\boldsymbol{\lambda}}_r = -\left(\frac{\partial H}{\partial \mathbf{r}}\right)^{\top} = -\left(\frac{\partial \mathbf{g}}{\partial \mathbf{r}}\right)^{\top}\boldsymbol{\lambda}_{\mathbf{v}} \tag{46}$$

$$\dot{\boldsymbol{\lambda}}_v = -\left(\frac{\partial H}{\partial \mathbf{v}}\right)^{\top} = -\boldsymbol{\lambda}_r - \left(\frac{\partial \mathbf{g}}{\partial \mathbf{v}}\right)^{\top}\boldsymbol{\lambda}_{\mathbf{v}} \tag{47}$$

$$\dot{\lambda}_m = -\frac{\partial H}{\partial m} = -\boldsymbol{\lambda}_v^{\top}\left(\frac{\partial \mathbf{g}}{\partial m} - \frac{T}{m^2}\mathbf{u}\right) \tag{48}$$

Extremization of H with respect to the thrust unit vector results in the well known thrust unit vector steering law[5, 6].

$$\mathbf{u}(t) = \frac{\boldsymbol{\lambda}_v(t)}{\lambda_v(t)} \tag{49}$$

and is referred to as the Primer Vector.

The solution requires knowing the costate vector $\boldsymbol{\lambda}(t_0)$; and it forms part of $\mathbf{s}_x^i$. An estimate for $\boldsymbol{\lambda}(t_0)$ is obtained by using an adjoint-control transformation[2]. If an estimate for $\mathbf{u}_0$ is available,

$$\boldsymbol{\lambda}_{v_0} = \lambda_{v_0}\mathbf{u}_0 \tag{50}$$

where $\mathbf{u}_0$

$$\mathbf{u}_0 = \begin{pmatrix} \cos\alpha_0\cos\beta_0 \\ \sin\alpha_0\cos\beta_0 \\ \sin\beta_0 \end{pmatrix} \tag{51}$$

The time rate of change of $\boldsymbol{\lambda}_{v_0}$ is

$$\dot{\boldsymbol{\lambda}}_{v_0} = \dot{\lambda}_{v_0}\mathbf{u}_0 + \lambda_{v_0}\dot{\mathbf{u}}_0 \tag{52}$$

where

$$\dot{\mathbf{u}}_0 = \begin{pmatrix} -\dot{\alpha}_0\sin\alpha_0\cos\beta_0 - \dot{\beta}_0\cos\alpha_0\sin\beta_0 \\ \dot{\alpha}_0\cos\alpha_0\cos\beta_0 - \dot{\beta}_0\sin\alpha_0\sin\beta_0 \\ \dot{\beta}_0\cos\beta_0 \end{pmatrix} \tag{53}$$

Using Eq. 47

$$\boldsymbol{\lambda}_{r_0} = -\left[\dot{\lambda}_{v_0} + \lambda_{v_0}\left(\frac{\partial \mathbf{g}}{\partial \mathbf{v}}\right)^{\top}_{t_0}\right]\mathbf{u}_0 - \lambda_{v_0}\dot{\mathbf{u}}_0\,. \tag{54}$$

The basic transformation is summarized as

$$\begin{pmatrix} \lambda_{v_0} \\ \dot{\lambda}_{v_0} \\ \alpha_0, \beta_0 \\ \dot{\alpha}_0, \dot{\beta}_0 \\ \lambda_{m_0} \end{pmatrix} \Longrightarrow \begin{pmatrix} \lambda_{v_0} \\ \dot{\lambda}_{v_0} \\ \mathbf{u}_0 \\ \dot{\mathbf{u}}_0 \\ \lambda_{m_0} \end{pmatrix} \Longrightarrow \begin{pmatrix} \boldsymbol{\lambda}_{v_0} \\ \boldsymbol{\lambda}_{r_0} \\ \lambda_{m_0} \end{pmatrix}. \tag{55}$$

The necessity of explicitly specifying all of the components of the costate vector has been removed in favor of specifying the four control related variables $(\alpha, \beta, \dot{\alpha}, \dot{\beta})$ and three costate related variables $(\lambda_v, \dot{\lambda}_v, \lambda_m)$. Without any prior knowledge of the solution for $\mathbf{u}(t)$, obtaining an estimate for these remaining quantities is possible, but is problem dependent and requires further knowledge associated with the additional transversality conditions associated with the optimal control problem[9]. If a parameter model solution exists, a method to obtain these for a constant exhaust velocity engine is discussed in the last section of this paper. The use of the transformation replaces the explicit costate vector in $\mathbf{s}_x^i$ with the adjoint control variables.

It is important to note that even though an optimal control formulation leads to a well defined multi-point boundary value problem with functions that include transversality conditions on the states and costates at distinct times, these conditions are not used in the system. This is intentional. Instead, the method used is best described as a hybrid method where some of the unknown variables, which include the initial values of the costates or functions of them, form part of the variable vector and the cost function is optimized directly with an optimization algorithm. The advantage being that the control follows the control optimality condition, without explicitly deriving and enforcing the transversality conditions.

4. TARGETING AND OPTIMIZATION

Let $\mathbf{x}_p$ be the $n_p \times 1$ vector that contains the set of variables to be determined. $\mathbf{x}_p$ is an augmented vector that contains all of the segment variable vectors $\mathbf{s}_x^i (i = 1..n)$. The function vector is the $n_c \times 1$ function vector $\mathbf{c}$ that contains all of the segment function vectors $\mathbf{s}_f^i (i = 1..n)$. $\mathbf{c}$ is functionally dependent on $\mathbf{x}_p$,

$$\mathbf{c} = \mathbf{c}(\mathbf{x}_p) \tag{56}$$

The constraints on $\mathbf{c}$ are

$$\mathbf{c}^l \leq \mathbf{c}(\mathbf{x}_p) \leq \mathbf{c}^u \tag{57}$$

where $\mathbf{c}^l$ and $\mathbf{c}^u$ are constant vectors of lower and upper bounds, respectively. An equality constraint on an element j of $\mathbf{c}$ is defined by setting

$$\mathbf{c}_j^l = \mathbf{c}_j^u = \mathbf{c}_j^* \tag{58}$$

where $\mathbf{c}_j^*$ is the required target value. Simple one sided inequality constraints $\mathbf{c}_j$, are set by choosing

$$-\infty \leq \mathbf{c}_j \leq \mathbf{c}_j^u \text{ or } \mathbf{c}_j^l \leq \mathbf{c}_j \leq +\infty \tag{59}$$

If $n_p = n_c$ and all of the constraints on $\mathbf{c}$ are equality constraints, then the problem requires solving a nonlinear system of equations. If $n_p < n_c$ and all of the constraints on $\mathbf{c}$ are equality constraints, then a minimax solution is needed where it is necessary to find the local minimum of the function

$$\max \left| [\mathbf{c}(\mathbf{x}_p)]_j \right| , \; j = 1...n_c \, . \tag{60}$$

If $n_p > n_c$, the system is underdetermined, the minimum of the function in Eq. 60 zero, and a solution is the minimum norm solution from the initial value of $\mathbf{x}_p$. For an optimization problem, let the element k of $\mathbf{c}$, $\mathbf{c}_k$, be the objective function with the condition that $\mathbf{c}_k^l \neq \mathbf{c}_k^u$. The nonlinear constrained optimization problem is to minimize or maximize $\mathbf{c}_k$ subject to the constraints given by Eq. 57. The problem has been cast in a standard form currently used by state of the art nonlinear constrained optimization codes such as SNOPT[7, 15].

Regardless of whether a targeting or optimization problem is required to be solved, it is desirable to at least estimate as accurately as possible the $n_p \times n_c$ Jacobian matrix $\partial \mathbf{c} / \partial \mathbf{x}_p$. Numerical finite difference methods are the most common methods for this purpose, even though they have inherit difficulties especially for the functions in $\mathbf{c}$ that are highly sensitive to the elements in $\mathbf{x}_p$. For example, the (i, j) element of $\partial \mathbf{c} / \partial \mathbf{x}_p$ can be approximated using a central difference approximation

$$\frac{\partial c_i}{\partial x_{p_j}} \approx \frac{c_i(x_{p_j} + \Delta x_{p_j}) - c_i(x_{p_j} - \Delta x_{p_j})}{2 \Delta x_{p_j}} \tag{61}$$

where Δx_{p_j} is the positive perturbation stepsize for the x_{p_j} element of $\mathbf{x}_p$. A very accurate method to compute $\partial \mathbf{c} / \partial \mathbf{x}_p$ uses state transition matrices and analytical gradient expressions for all of the functions and the states required required to compute those functions[18, 11, 12]; however, this method is not practical for a generalized system because of the overhead required to derive all of the required relationships between the large number of search variables and constraint functions.

Examples that describe how to cast mission design and trajectory optimization problems into a targeting or optimization solution method for both impulsive and finite burn maneuvers are described in[13].

5. FUTURE CONSIDERATIONS

One of the areas currently being examined is an alternate formulation for the finite burn thrust steering model. For the parameter basic steering model, assuming only at most a quadratic function for the spherical angles used to represent $\mathbf{u}(t)$, an augmented state vector that includes the parameters that define the thrust unit vector $\mathbf{u}(t)$ and its rate $\dot{\mathbf{u}}(t)$ is being considered,

$$\mathbf{x} = \begin{pmatrix} \mathbf{r}^\top & \mathbf{v}^\top & m & \alpha & \beta & \dot{\alpha} & \dot{\beta} \end{pmatrix}^\top \tag{62}$$

with

$$\dot{\mathbf{x}}(t,\mathbf{x}) = \begin{pmatrix} \mathbf{v} \\ \mathbf{g}(\mathbf{r},\mathbf{v},m,t,\mathbf{p}) + \frac{T(t)}{m}\mathbf{u}(t) \\ -\frac{T(t)}{c(t)} \\ \dot{\alpha} \\ \dot{\beta} \\ \ddot{\alpha} \\ \ddot{\beta} \end{pmatrix}^i \tag{63}$$

In this model, the initial conditions for the spherical angles and their rates form part of the segment variable vector. With this model, a procedure is being investigated to automate the conversion of a parameter model finite burn solution to a finite burn solution that uses the primer vector for the thrust unit vector. As an example, assuming that $\mathbf{g}$ only depends on $\mathbf{r}$ and t, it can be shown that an augmented state vector of the form

$$\mathbf{x} = \begin{pmatrix} \mathbf{r}^\top & \mathbf{v}^\top & m & \alpha & \beta & \dot{\alpha} & \dot{\beta} & \lambda_v & \dot{\lambda}_v & \lambda_m \end{pmatrix}^\top \tag{64}$$

has the following first order equation of motion

$$\dot{\mathbf{x}}(t,\mathbf{x}) = \begin{pmatrix} \mathbf{v} \\ \mathbf{g}(\mathbf{r},\mathbf{v},m,t,\mathbf{p}) + \frac{T}{m}\mathbf{u} \\ -\frac{T}{c} \\ \dot{\alpha} \\ \dot{\beta} \\ \ddot{\alpha} \\ \ddot{\beta} \\ \dot{\lambda}_v \\ \ddot{\lambda}_v \\ \frac{T}{m^2}\lambda_v \end{pmatrix} \tag{65}$$

where $\dot{\alpha}, \dot{\beta}$ are determined uniquely from $\mathbf{u}$ and $\dot{\mathbf{u}}$; and $\ddot{\alpha}, \ddot{\beta}$ are determined uniquely from $\mathbf{u}, \dot{\mathbf{u}}$ and $\ddot{\mathbf{u}}$; and

$$\ddot{\mathbf{u}} = \mathbf{G}\mathbf{u} - \left\{\left[\left(\mathbf{u}^\top\mathbf{G}\right)(\mathbf{u})\right] + \dot{\mathbf{u}}^\top\dot{\mathbf{u}}\right\}\mathbf{u} - 2\frac{\dot{\lambda}_v}{\lambda_v}\dot{\mathbf{u}} \tag{66}$$

$$\ddot{\lambda}_v = \lambda_v\left[\left(\mathbf{u}^\top\mathbf{G}\right)(\mathbf{u}) + \dot{\mathbf{u}}^\top\dot{\mathbf{u}}\right] \tag{67}$$

where

$$\mathbf{G} = \frac{\partial \mathbf{g}(\mathbf{r},t)}{\partial \mathbf{r}} \tag{68}$$

With this formulation, an automated procedure is being developed that transforms a parameter model finite burn solution to an approximation of the nearby optimal control finite burn solution. For a constant exhaust velocity engine use is made of the associated switching function S that determines the value for T,

$$S = \frac{\lambda_v}{m} - \frac{\lambda_m}{c} \tag{69}$$

The cost function is minimization of the propellant consumed by the finite burn. With this, $S(t_0) = 0$ and t_0 is forced to be the switching time from zero thrust to $T_{\max}$. The scale for the costates is set by choosing

$$\lambda_v(t_0) = 1 \tag{70}$$

Using $S(t_0) = 0$,

$$\lambda_m(t_0) = c\frac{\lambda_v(t_0)}{m(t_0)} \tag{71}$$

The time history of $\lambda_v(t)$ is assumed to be quadratic with negative curvature ($\ddot{\lambda}_v < 0$) and with a maximum value near the middle of the burn. If t_m is the time of the midpoint of the burn then as an approximation set $\dot{\lambda}_v(t_m) = 0$. Having a parameter model finite burn solution, Eq. 67 is used at t_m to get an approximation for $\ddot{\lambda}_v$. $\ddot{\lambda}_v$ is assumed to be constant; this along with the conditions $\lambda_v(t_0) = 1, \dot{\lambda}_v(t_m) = 0$ the expression for $\dot{\lambda}_v(t_0)$ is

$$\dot{\lambda}_v(t_0) = -\lambda_v(t_m)\left[\left(\mathbf{u}^\top\mathbf{G}\right)(\mathbf{u}) + \dot{\mathbf{u}}^\top\dot{\mathbf{u}}\right]_{t_m}(t_m - t_0) \tag{72}$$

where $\lambda_v(t_m)$ is the value of λ_v at t_m and is given by,

$$\lambda_v(t_m) = \frac{1}{1 + \frac{1}{2}\left[\left(\mathbf{u}^\top\mathbf{G}\right)(\mathbf{u}) + \dot{\mathbf{u}}^\top\dot{\mathbf{u}}\right]_{t_m}(t_m - t_0)^2} \tag{73}$$

This procedure provides the initial estimate for all of the adjoint control variables needed in the augmented state vector, Eq. 64, at t_0.

Another area being considered for possible inclusion in Copernicus is a general architecture for guidance algorithms and control laws. Current finite burn models in Copernicus can be complemented with guidance algorithms for specific applications. Finding a general method for the implementation of such algorithms is still under research. The implementation of a general relative motion capability between segments is also under consideration. This will allow a more flexible modeling and optimization of spacecraft rendezvous and formation flying problems. Finally, a more complete vehicle model within the segment that incorporates the ability to model a spacecraft as a system of multiple masses, e.g.: tanks, consumables, dry mass, stages, propellant, etc. is being designed.

REFERENCES

[1] Bryson, A. E. Jr., and Ho, Y. C. (1975) *Applied Optimal Control,* Hemisphere Publishing Corporation, Revised Printing, New York, NY.

[2] Dixon, L. C., and Bartholomew-Biggs, M. C. (1981) Adjoint Control Transformations for Solving Practical Optimal Control Problems, *Optimal Control Applications and Methods,* **2**, 365–381.

[3] Garn M., Qu M., Chrone J., Su P., Karlgaard C. NASA's Planned Return to the Moon: Global Access and Anytime Return Requirement Implications on the Lunar Orbit Insertion Burns, *Proceedings of the AIAA Guidance, Navigation and Control conference,* August, 2008.

[4] Hull, D. G. (2003) *Optimal Control Theory for Applications,* Springer-Verlag, New York, NY.

[5] Lawden, D. E. (1963) *Optimal Trajectories for Space Navigation*, Butterworths, London.

[6] Marec, J. P. (1979) *Optimal Space Trajectories*, Elsevier Scientific Publishing Company, New York, NY.

[7] Gill, P. E. et. al., (1998) *Practical Optimization*, Academic Press, San Diego, CA.

[8] Ocampo, C. (2003) An Architecture for a Generalized Trajectory Design and Optimization System, in *Libration Point Orbits and Applications*, Edited by Gomez, G., Lo, M., and Masdemont, J. World Scientific Publishing Co., Singapore.

[9] Ocampo, C. (2004) Finite Burn Maneuver Modeling for a Generalized Spacecraft Trajectory Design and Optimization System, *Astrodynamics, Space Missions, and Chaos, Annals of the New York Academy of Sciences*, **1017**, 210–233.

[10] Ocampo C "Elements of a Software System for Spacecraft Trajectory Optimization," *Spacecraft Trajectory Optimization*, Ed. Bruce Conway, Cambridge University Press 2010

[11] Ocampo C. and Munoz JP. Variational Equations for a Generalized Trajectory Model, AAS-09-255, *Proceedings of the 19th Space Flight Mechanics Meeting*, Savannah, GA, 2009.

[12] Ocampo C. and Munoz JP. Variational Model for the Optimization of Constrained Finite-Burn Escape Sequences AAS 09-381, *2009 AAS/AIAA Astrodynamics Specialist Conference Proceedings*, Pittsburgh, PA, 2009.

[13] Ocampo, C. and Byrnes, D. "Mission Design and Trajectory Optimization," chapter to appear in *Encyclopedia of Aerospace Engineering*, Wiley Publishing, Dec 2010

[14] Senent J. An Optimal Initial Guess Generator for Entry Interface Targeters AAS 09-426, *2009 AAS/AIAA Astrodynamics Specialist Conference Proceedings*, Pittsburgh, PA, 2009.

[15] Stanford Business Software http://www.sbsi-sol-optimize.com/ asp/sol_product_snopt.html

[16] Williams J., Davis E., Lee D., Condon G., Dawn T., Qu M. Global Performance Characterization of the three burn trans-Earth injection maneuver sequence over the lunar nodal cycle, *2009 AAS/AIAA Astrodynamics Specialist Conference Proceedings*, Pittsburgh, PA, 2009.

[17] Williams J., Senent, J., Ocampo C., Mathur R., Davis E. Overview and Software Architecture of the Copernicus Trajectory Design and Optimization System, in *4th International Conference on Astrodynamics Tools and Techniques*, Madrid, May, 2010

[18] Zimmer, S., and Ocampo, C. (2005) Use of Analytical Gradients to Calculate Optimal Gravity-Assist Trajectories, *Journal of Guidance, Control, and Dynamics*, **28**, No. 2, 324–332.