AAS 14-272

AUTOMATED PATCH POINT PLACEMENT FOR SPACECRAFT TRAJECTORY TARGETING

Galen Harden*, Amanda Haapala†, Kathleen C. Howell‡, and Belinda Marchand§

Autonomous targeting and guidance, specifically that which relies on iterative gradient-based processes, is sensitive to the quality of the startup guess or estimate and the underlying dynamical sensitivities. These startup arcs are often delivered as a collection of "patch states", i.e., a discretized representation of the overall trajectory in the full dynamical model. The goal of this work is to devise a systematic process by which to quantify and characterize the dynamical sensitivity of these points, and use this scheme to devise a metric that facilitates the systematic determination of their placement along the path prior to the application of optimal or sub-optimal corrections processes. The purpose of this strategy, ultimately, is to improve the performance of gradient based targeting and guidance algorithms in offline autonomous scenarios where human intervention is not plausible. The approach leverages Local Lyapunov Exponents in devising the metric and process by which patch states are systematically located.

INTRODUCTION

Trajectory design in multi-body regimes is typically accomplished by an expert familiar with the gravitational fields and the perturbations expected along the path, and with an intuitive understanding of the arcs likely to yield a successful result. For trajectories dominated by the gravity of the Sun, planets, and well-modeled moons, this approach is sufficient for pre-planning a baseline path which is the primary focus in the development of guidance algorithms. If an Earth-orbiting spacecraft significantly departs from the baseline or nominal path, an expert on the ground can design a new trajectory that meets the mission requirements and transmit it to the spacecraft. Distant spacecraft potentially face a communications delay[1] that adds to the time required to re-design the path, and could result in relatively long periods of flight without a viable guidance strategy. The communications time lag may not be significant for long-term trajectory planning, but for short-term trajectory adjustments–such as collision avoidance–the communications lag could be critical and an alternative approach is warranted. Other scenarios may also benefit from an autonomous guidance strategy. Spacecraft in the vicinity of asteroids or other celestial objects with gravitational fields that are not

*Graduate Student, School of Aeronautics and Astronautics, Purdue University, 701 W. Stadium Ave. West Lafayette, IN, 47907-2045.

†Ph.D. Candidate, School of Aeronautics and Astronautics, Purdue University, 701 W. Stadium Ave. West Lafayette, IN, 47907-2045.

‡Hsu Lo Distinguished Professor of Aeronautics and Astronautics, School of Aeronautics and Astronautics, Purdue University, 701 W. Stadium Ave. West Lafayette, IN, 47907-2045

§Adjunct Professor, Purdue University, Austin, TX, 78701.

sufficiently well-modeled can introduce further complications. Thus, the guidance strategy must be more focused and robust. Autonomous, onboard spacecraft trajectory design would eliminate the time delays associated with communications as well as the requirement for ground analysis; however, the development of an automated process is particularly challenging in multi-body dynamical regimes. Multiple aspects of an expert analysis must be distilled into simple algorithms to automate the entire trajectory re-design process. Exploring specific aspects of trajectory design individually and then combining them into a single, cohesive, autonomous design algorithm is the ultimate goal.

Given that the ultimate goal is an autonomous targeting and design algorithm, the primary focus in this investigation is the capability to produce an alternative path for the vehicle, one that may be very different from the nominal design, and incorporating significant time intervals in non-Keplerian environments. The trajectory design process is based upon the decomposition of a candidate baseline trajectory into multiple segments and a targeting scheme that subsequently blends the arcs into a continuous path that meets the appropriate constraints. For simpler trajectories, the path may be adjusted as a single segment; however, for more complex scenarios, this approach is not feasible from a numerical perspective without a very good reference path or initial guess. To mitigate the numerical issues associated with targeting, it is common to use a multiple-shooting formulation of a differential corrector, which corrects multiple-segment trajectories. Continuity and other applicable constraints are enforced at the break points.[2,3,4] In addition to improving the convergence behavior of the targeter, multiple-segment formulations allow for a simple implementation of impulsive maneuvers. The patch points, therefore, serve the dual purposes of robust convergence and potential maneuver locations. Thus, the location of the patch points can influence the convergence behavior as well as the characteristics of the converged path. Anderson et al.[5] utilizes similar tools and summarizes observations concerning the relationship between maneuver placement and stationkeeping costs. The mutual trade-offs between convergence properties and propellant usage is a topic for future work. The focus here is the improvement of the convergence behavior by placement of the patch points such that an automated algorithm to converge the segments is successful.

METHOD

Circular Restricted 3-Body Problem

The Circular Restricted Three-Body Problem (CR3BP) serves as a reasonable approximation to model the dynamical environment that governs the motion of a small satellite in many regions in the solar system. Three bodies are incorporated in the model – two massive celestial bodies, i.e., the primaries, as well as the vehicle of interest. The motion of an infinitesimally small body in the vicinity of any planet and moon, or simultaneously under the influence of the Sun and a planet, is dominated by the gravitational fields of the two massive bodies. Motion under the influence of additional gravitational bodies can be modeled by transitioning between various CR3BP systems,[6,7] reshaping or retiming the trajectory to avoid additional bodies,[8] or incorporating a higher-fidelity model. The position of the spacecraft relative to the barycenter and each primary is depicted in Figure 1. The system mass parameter, μ, is the ratio of the mass of the smaller primary mass to the combined mass of both primaries. The nondimensionalization, scaling distances by the characteristic length between the two primaries and time by the characteristic time associated with the primaries rotating through one radian in the inertial frame, assuages difficulties associated with numerical integration of quantities on vastly different scales.

The equations of motion in the CR3BP are derived from Newton's second law of motion by introducing simplifying assumptions and incorporating convenient transformations.[9] The nondi-

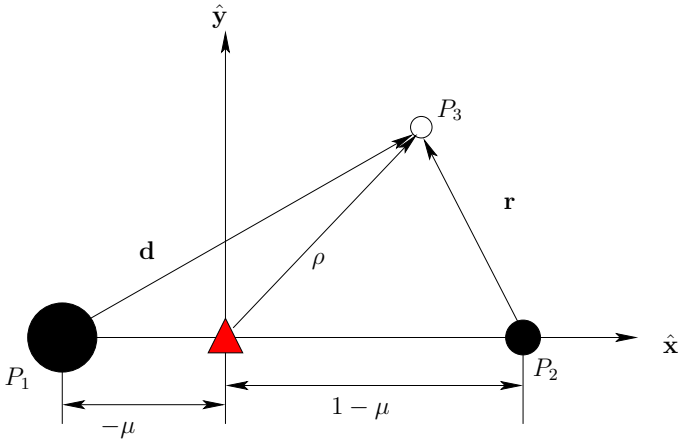


Figure 1. Position of Spacecraft Relative to Barycenter and Primaries

mensional equations of motion are

$$\ddot{x} - 2\dot{y} - x = -\frac{(1-\mu)(x+\mu)}{|\mathbf{d}|^3} - \frac{\mu(x-1+\mu)}{|\mathbf{r}|^3} \tag{1}$$

$$\ddot{y} + 2\dot{x} - y = -\frac{(1-\mu)y}{|\mathbf{d}|^3} - \frac{\mu y}{|\mathbf{r}|^3} \tag{2}$$

$$\ddot{z} = -\frac{(1-\mu)z}{|\mathbf{d}|^3} - \frac{\mu z}{|\mathbf{r}|^3} \tag{3}$$

where x, y, and z are Cartesian coordinates in a rotating frame in which the primaries appear fixed and centered at the barycenter; $\mathbf{d}$ and $\mathbf{r}$ are the position vectors of the spacecraft relative to the larger and smaller primaries, respectively, and the dot notation indicates a derivative with respect to nondimensional time.

Multiple-Shooting Targeters

Multiple-shooting differential correctors allow the reasonably simple method of differential corrections to be applied to relatively complex trajectories moving through nonlinear and dynamically sensitive environments. The underlying technique of a single-shooting differential corrector involves guessing an initial state and then iteratively updating the guess. The guesses for initial position and velocity states are stored in a design vector, $\mathcal{X}$, and constraints–represented by scalar equality expressions that evaluate to zero–are collected in a constraint vector, $\mathbf{F}$. For example,

$$\mathcal{X} = \begin{bmatrix} x(t_0) & y(t_0) & z(t_0) & \dot{x}(t_0) & \dot{y}(t_0) & \dot{z}(t_0) & \tau \end{bmatrix}^T \equiv \begin{pmatrix} \mathbf{X}(t_0) \\ \tau \end{pmatrix} \tag{4}$$

contains the initial states at time t_0 as well as the duration of propagation, τ; $\mathbf{X}(t_0)$ is the initial state vector corresponding to one trajectory segment. A sample constraint vector,

$$\mathbf{F} = \begin{bmatrix} y(t_0) & z(t_0) & \dot{x}(t_0) & \dot{z}(t_0) & y(t_0+\tau) & z(t_0+\tau) & \dot{x}(t_0+\tau) & \dot{z}(t_0+\tau) \end{bmatrix}^T \tag{5}$$

enforces perpendicular crossings of the $\hat{x}$-$\hat{z}$ plane at the beginning and end of the segment. Updates to the design vector are produced by propagating the initial guess to detect violations of the constraint vector and relating the errors to the initial conditions, typically employing the state transition matrix. The iterative update is evaluated as,

$$\mathcal{X}(i+1) = \mathcal{X}(i) - DF^T(DFDF^T)^{-1}\mathbf{F}, \qquad \text{where} \quad DF = \frac{\partial \mathbf{F}}{\partial \mathcal{X}}|_{\mathcal{X}(i)}, \quad \mathbf{F} = \mathbf{F}|_{\mathcal{X}(i)} \tag{6}$$

thus, DF is the Jacobian matrix of the constraint vector with respect to the design variable vector. Assuming the initial guess is sufficiently close to a solution, Equation (6) is applied until the norm of the constraint vector falls below an acceptable tolerance, i.e., the targeter converges. Multiple techniques have been developed for extending the single-shooting differential corrector to multiple-segment trajectories and enforcing continuity between the segments.

Parallel Shooting One strategy for implementation of a multiple-shooting targeter employs a type of parallel shooting,[10] which combines the states at all patch points into a single design vector that essentially redefines the multiple-shooting problem into a single-shooting problem with correspondingly more dimensions. For example, a parallel-shooting design vector may take the form,

$$\mathcal{X} = \begin{pmatrix} \mathbf{X}_1(t_1) \\ \mathbf{X}_2(t_2) \\ \vdots \\ \mathbf{X}_k(t_k) \\ \tau_1 \\ \tau_2 \\ \vdots \\ \tau_k \end{pmatrix} \tag{7}$$

where $\mathbf{X}_j(t_j)$ is the initial state vector associated with patch point j. Full-state continuity between segment j and $j+1$ is enforced by including a continuity expression in the constraint vector,

$$\mathbf{F} = \begin{pmatrix} \vdots \\ \mathbf{X}_j(t_j + \tau_j) - \mathbf{X}_{j+1}(t_{j+1}) \\ \vdots \end{pmatrix} \tag{8}$$

This approach is a conceptually simple extension of single-shooting corrections–employing the same update equation, Equation (6)–but requires inversion of large matrices that may not be feasible for onboard applications.

Two-Level Corrector The "two-level" corrector offers a less computationally-intensive alternative to parallel shooting and decomposes the problem into two levels. Level-I enforces position continuity between segments by modifying the initial velocity states on each segment independently; thus, Level-I consists of a sequence of single-shooting corrections. Level-II enforces velocity continuity, as well as additional constraints,[11] by modifying the initial positions of all patch points simultaneously. Level-II resembles a single iteration of a parallel shooting algorithm, but with approximately half as many elements in the design vector. Three implementations of Level-II have been developed [2,12,3] that offer different approaches for incorporating constraints. Level-I

introduces velocity discontinuities at patch points to achieve position continuity, and Level-II simultaneously shifts all positions, so the corrector must alternate between the levels until all constraints are satisfied.

For implementation of the targeting strategy, Harden employs a derivation that includes the DF matrix from parallel shooting;[3] thus, the method affords the flexibility to apply either parallel shooting or the two-level corrector with minimal additional effort. Let $\mathcal{X}^I$ denote the design variables in Level-I–velocity states; similarly, $\mathcal{X}^{II}$ is the design variable vector in Level-II–position states and times. Thus $\mathcal{X}^I$ and $\mathcal{X}^{\mathbf{II}}$ together form the $\mathcal{X}$ consistent with parallel shooting. Define,

$$DX = \frac{\partial \mathcal{X}}{\partial \mathcal{X}^{II}} \tag{9}$$

that is, the Jacobian of all available design variables with respect to the design variables available in Level-II. The M matrix,

$$M = (DF)(DX) = \frac{\partial \mathbf{F}}{\partial \mathcal{X}} \frac{\partial \mathcal{X}}{\partial \mathcal{X}^{II}} = \frac{\partial \mathbf{F}}{\partial \mathcal{X}^{II}} \tag{10}$$

relates small changes in the Level-II design variables to the resulting changes in the constraint vector. The Level-II update equation,

$$\mathcal{X}^{II}(i+1) = \mathcal{X}^{II} - M^T (M M^T)^{-1} \mathbf{F} \tag{11}$$

is effectively Equation (6) with $\mathcal{X}^{II}$ instead of $\mathcal{X}$ and M in place of DF. Thus, once a parallel-shooting targeter is derived, only the DX matrix is required to derive the Level-II corrector.

The DX matrix depends primarily on the state transition matrix, $\phi(t_i, t_{i-1})$; however, applying the chain rule to this Jacobian is nontrivial. It is beneficial to express the linear variational equations using a block matrix form for the state transition matrix,

$$\begin{aligned} \begin{Bmatrix} \delta\rho_i^- \\ \delta\mathbf{v}_i^- \end{Bmatrix} &= \phi(t_i, t_{i-1}) \begin{Bmatrix} \delta\rho_{i-1} \\ \delta\mathbf{v}_{i-1} \end{Bmatrix} + \begin{Bmatrix} \mathbf{v}_i^- \\ \mathbf{a}_i^- \end{Bmatrix} (\delta t_i - \delta t_{i-1}) \\ &= \begin{bmatrix} A_{i,i-1} & B_{i,i-1} \\ C_{i,i-1} & D_{i,i-1} \end{bmatrix} \begin{Bmatrix} \delta\rho_{i-1} \\ \delta\mathbf{v}_{i-1} \end{Bmatrix} + \begin{Bmatrix} \mathbf{v}_i^- \\ \mathbf{a}_i^- \end{Bmatrix} (\delta t_i - \delta t_{i-1}) \end{aligned} \tag{12}$$

where ρ_{i-1} and $\mathbf{v}_{i-1}$ are the position and velocity at the beginning of segment $i-1$, and ρ_i^-, $\mathbf{v}_i^-$, and $\mathbf{a}_i^-$ are the position, velocity, and acceleration vectors at the end of segment $i-1$. Then, $A_{i,i-1}$, $B_{i,i-1}$, $C_{i,i-1}$, and $D_{i,i-1}$ are 3-by-3 submatrices of the state transition matrix from time t_{i-1} to t_i. Solving this equation for $\delta\mathbf{v}_{i-1}$ and $\delta\mathbf{v}_i^-$, applying the position continuity achieved in Level-I, and differentiating with respect to Level-II design variables, yields

$$\frac{\partial \mathbf{v}_{i-1}}{\partial \rho_{i-1}} = -B_{i,i-1}^{-1} A_{i,i-1} \tag{13}$$

$$\frac{\partial \mathbf{v}_{i-1}}{\partial \rho_i} = B_{i,i-1}^{-1} \tag{14}$$

$$\frac{\partial \mathbf{v}_{i-1}}{\partial t_{i-1}} = B_{i,i-1}^{-1} \mathbf{v}_i^- \tag{15}$$

$$\frac{\partial \mathbf{v}_{i-1}}{\partial t_i} = -B_{i,i-1}^{-1} \mathbf{v}_i^- \tag{16}$$

$$\frac{\partial \mathbf{v}_i^-}{\partial \rho_{i-1}} = C_{i,i-1} - D_{i,i-1} B_{i,i-1}^{-1} A_{i,i-1} \tag{17}$$

$$\frac{\partial \mathbf{v}_i^-}{\partial \rho_i} = D_{i,i-1} B_{i,i-1}^{-1} \tag{18}$$

$$\frac{\partial \mathbf{v}_i^-}{\partial t_{i-1}} = D_{i,i-1} B_{i,i-1}^{-1} \mathbf{v}_i^- - \mathbf{a}_i^- \tag{19}$$

$$\frac{\partial \mathbf{v}_i^-}{\partial t_i} = \mathbf{a}_i^- - D_{i,i-1} B_{i,i-1}^{-1} \mathbf{v}_i^- \tag{20}$$

Constructing the DX matrix, then, consists primarily of arranging these partials appropriately. The arrangement depends on the structure of $\mathcal{X}$ and $\mathcal{X}^{II}$, but is independent of the constraints applied. Thus, once the structure of the design variables is selected, the DX matrix is only derived once.

Lyapunov Exponents

Corrections strategies are fundamentally based on predictions of the evolution of a dynamical system. Lyapunov exponents are a construct from dynamical systems theory that facilitate the process of characterizing the rate of separation of two initially infinitesimally close trajectories as they evolve in time. Generally, the mathematical definition may be represented as,

$$\lambda(t + \tilde{\tau}) = \lim_{\tilde{\tau} \to \infty} \frac{1}{\tilde{\tau}} \log \frac{||\delta \mathbf{X}(t + \tilde{\tau})||}{||\delta \mathbf{X}(t)||} \tag{21}$$

where

$$\delta \mathbf{X}(t + \tilde{\tau}) = \phi(t + \tilde{\tau}, t) \delta \mathbf{X}(t) \tag{22}$$

If $\mathbf{X} \in \mathbb{R}^n$, the full Lyapunov exponent spectrum is characterized by $\lambda_1, \lambda_2, \cdots, \lambda_n$, where each value is associated with one unique fundamental direction in the phase space. Each exponent defines the rate of expansion, or contraction, of the trajectory along its respective direction. In general, there is no analytical means of identifying the exponents and, thus, they are often approximated over a finite horizon. This simplification naturally leads to the term Finite-Time Lyapunov Exponents (FTLE). The accuracy of the approximation degrades with the shortening of the horizon time; the term FTLE still implies the full spectrum of exponents. However, a reasonable approximation for the local growth rate may still be acquired by considering only the largest of these exponents, or the Local Lyapunov Exponent (LLE). Since the stability of the system is dictated, ultimately, by the largest Lyapunov exponent, the LLE is sufficient for stability analysis in many applications. Short and Howell[13,14] use the FTLE and demonstrate the relationship of Lagrangian coherent structures and the dynamical structures in astrodynamics systems. Haapala[15] explored the relationship between the LLE values and targeter convergence. The local Lyapunov exponent proves to be a particularly useful approximation since it offers insights into possible autonomous placement of patch points along a baseline trajectory.

Relationship of LLE to Targeter Convergence The LLE value associated with a specific state along the trajectory quantifies the sensitivity to small perturbations, relative to that state, as the path evolves over some specified finite horizon, $\tilde{\tau}$. The finite horizon, here, is also sometimes denoted the "normalization time". Gradient-based targeting algorithms, whether optimal or sub-optimal, often depend on a user-selected set of nodes, sometimes labeled "patch states", along a candidate startup solution. If this candidate set of nodes, or patch states, includes states characterized as "dynamically

sensitive", the subsequent targeting processes can be adversely affected. Earlier investigations[15] suggested a correlation between the LLE values corresponding to a candidate set of nodes and the subsequent performance of targeting processes. Specifically, patch states in low-LLE regions exhibit better overall performance in targeting processes than patch states in high-LLE regions. This is not unexpected since the linear variational equations better approximate motion in low-LLE regions, as perturbations in high-LLE regions rapidly depart the vicinity of the reference path.

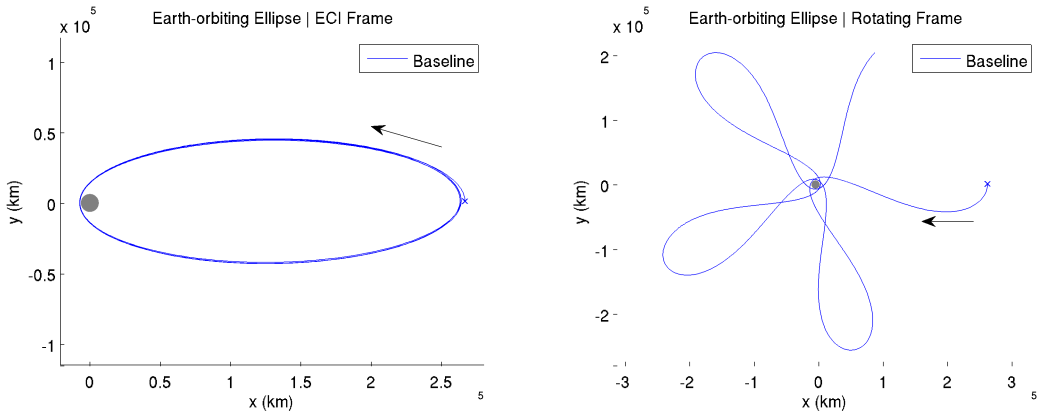
Though the LLE's apparently offer a quantifiable metric by which to characterize dynamical sensitivities along a path, one practical challenge remains for their use in an automated node selection process, that is, the determination of the LLE contour is a function of the normalization time, $\tilde{\tau}$. To illustrate this dependency, consider the trajectory in Figure 2, as viewed in the Earth-centered inertial and Earth-Moon rotating frames; this trajectory is nondimensionalized by the Earth-Moon distance and the time required by the primaries to move through one radian. Overall, the path illustrated in Figure 2 spans 5 non-dimensional time units. This arc is employed to illustrate the process of efficiently computing the LLE contour, based on the definition,[5]

$$\lambda_{max}(t+\tilde{\tau},t) = \frac{1}{\tilde{\tau}} \ln \sqrt{\max \quad \text{eigenvalue} \quad \text{of} \quad \phi(t+\tilde{\tau},t)^T \phi(t+\tilde{\tau},t)}. \tag{23}$$

For any trajectory spanning an interval from time unit t_0 to some future time unit t_f, i.e., for $t_0 \leq t \leq t_f$, the time horizon, $\tilde{\tau}$, must satisfy the constraint $t + \tilde{\tau} \leq t_f$ at any given t. The matrix term $\phi(t+\tilde{\tau},t)$ represents the state transition matrix that characterizes the sensitivity of the trajectory as it evolves from t to $t+\tau$. Note that this expression is explicitly dependent on the selected finite horizon value, $\tilde{\tau}$. Thus, any contour computed based on this expression will vary consistent with this selection.

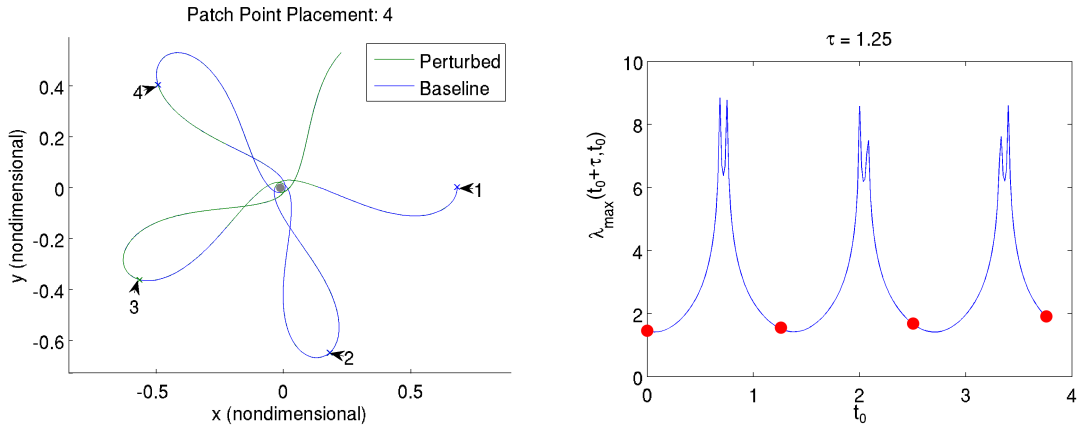
The sample trajectory in Figure 2 spans the interval from $t_0 = 0$ to $t_f = 5$. Two candidate patch state distributions are considered, each dictated by $\tilde{\tau} = (t_f - t_0)/M$ where $M = 4$ or $M = 5$. The distribution defined with four patch point states is illustrated in Figure 3 (a) while the five patch point state distribution appears in Figure 4 (a). These figures also include a perturbed path, computed by perturbing each patch state along the most unstable direction by a magnitude consistent with 10% of the distance from the center of the Earth. To the right in each figure is the LLE contour that results from evaluating Equation (23). The red circles on the LLE contour plots highlight the LLE values at each of the selected patch points. Note that the patch points in Figure 3 lie in troughs of the LLE contour, and the perturbed trajectory remains close to the baseline; in contrast, some of the patch points in Figure 4 lie near peaks of the LLE contour and the perturbed trajectory significantly departs from the baseline. Individual LLE contour plots are sufficient to assess a proposed set of patch points and evaluate the location of the nodes, that is, whether they occur on peaks or in troughs of the LLE contour. The LLE contour does not necessarily offer immediate insight into a process to shift the given patch points to deliver a more desirable result.

LLE Surface To employ a sequence of segments that are each propagated for different time intervals and to view the impact of different time horizons, multiple LLE contours must be visualized simultaneously. The *LLE surface* is a tool that allows visualization of a large set of full LLE contours on a single set of axes. The LLE surface for the trajectory that is plotted in Figure 2 appears now in Figure 5. This surface retains time along the horizontal axis; however, the LLE value is now mapped as a color intensity. The vertical axis of the LLE surface corresponds to the normalization time of a particular contour, $\tilde{\tau}$. The dark region in Figure 5 is representative of the previously stated constraint, requiring that $\tilde{\tau} \leq t_f - t$ for any given t along the trajectory. Recall that, in this particular example, $t_f = 5$. Thus, to evaluate the LLE equation at $t = 4$, for example, $\tilde{\tau} \leq 1$. Such a



(a) Inertial Frame (b) Rotating Frame

Figure 2. Earth-Centered Elliptical Orbit



(a) Perturbed Trajectory (b) $\tilde{\tau} = 1.25$ (nd) LLE Contour

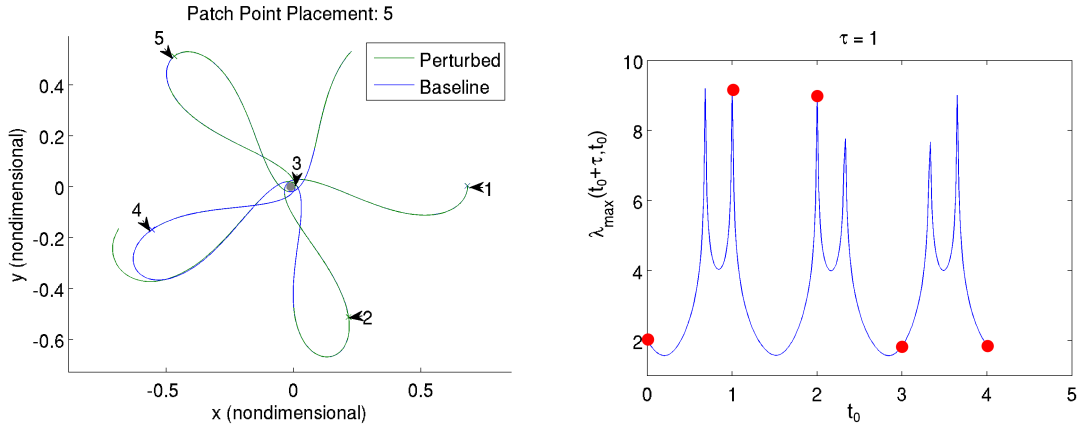
Figure 3. Four Patch Points Equally-Spaced in Time

constraint motivates the diagonal boundary, with a unity slope, as illustrated in Figure 5. For any given $\tilde{\tau}$, Equation (23) is evaluated from $0 \leq t \leq t_f - \tilde{\tau}$. The results of such an evaluation are represented as a horizontal contour of varying intensity. For each possible value of $\tilde{\tau}$, the collection of these contours forms the LLE surface.

White areas on the LLE surface correspond to peaks of the LLE contour, while darker regions indicate troughs. The LLE surface in Figures 5 and 6 can, in fact, be exploited to deliver a straightforward placement strategy for patch points along the Earth-centered trajectory in Figure 2.

Error Prediction

To develop an *automated* placement algorithm, it is beneficial to identify an easily computed metric that is useful in comparing two candidate sets of patch points. The candidate approach adopted in this investigation is based on a prediction of the severity of the constraint violations, or 'error', due to the initial guess for the path based on the placement of the patch points. Modeling the error associated with a trajectory in terms of the node times allows patch points to be placed,



(a) Perturbed Trajectory

(b) $\tilde{\tau} = 1$ (nd) LLE Contour

Figure 4. Five Patch Points Equally-Spaced in Time

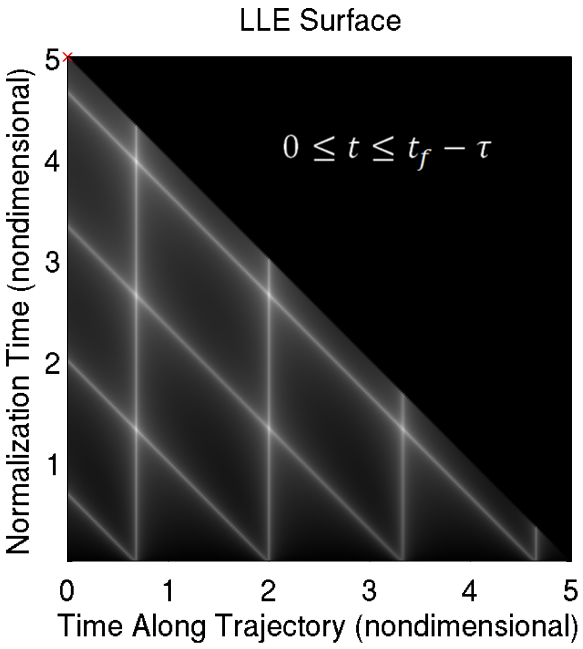


Figure 5. LLE Surface for baseline trajectory in Figure 2

potentially automatically, in a desirable configuration. Given a perturbation in the trajectory state, the error downstream will influence the corrections process through the patch point locations. The LLE plays a role in predicting the trajectory error, but there are also other factors that influence the magnitude of the constraint violations so the error model requires some simplifying assumptions.

A typical targeting strategy solves for a set of design variables that satisfies the constraints in the vector $\mathbf{F}$; for simplicity, parallel shooting is assumed in this derivation. Specifically, a parallel shooting strategy is applied assuming that the only constraints are state continuity. This distinction is significant because the Jacobian matrix employed in the targeting process is central to the error

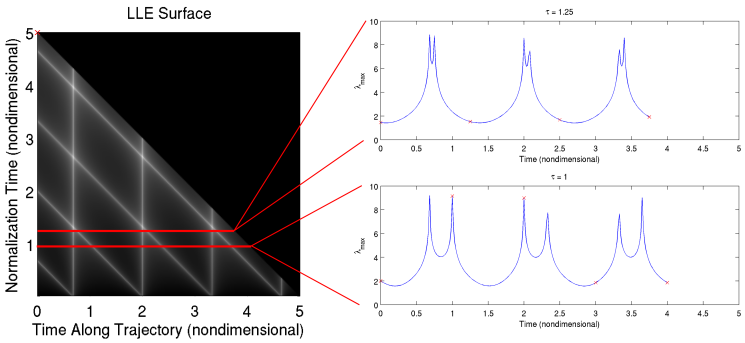


Figure 6. Interpreting the LLE Surface

prediction process. The simplifying assumptions that follow are specifically associated with this case and leverage the fact that the Jacobian matrix exhibits a unique and simple structure under these constraints. Naturally, the methodology requires modification to adapt to more complex cases. Still, this preliminary example suffices to illustrate the automated patch state selection process concept. The approach is not necessarily superior to other techniques or alternate implementations.

The goal of any corrections process is the determination of a nominal design vector, $\mathcal{X}^*$, that satisfies all the constraints, i.e., $\mathbf{F}(\mathcal{X}^*) = \mathbf{0}$. Of course, a precise nominal value is not available a-priori; only an initial guess, $\mathcal{X}$, is constructed, for which $\mathbf{F}(\mathcal{X}) \neq \mathbf{0}$. A linear corrections process approximates the relationship between $\mathbf{F}(\mathcal{X}^*)$ and $\mathbf{F}(\mathcal{X})$ by defining a perturbation variable,

$$\delta\mathcal{X} = \mathcal{X}^* - \mathcal{X} \tag{24}$$

and linearizing $\mathbf{F}(\mathcal{X}^*)$ relative to $\mathbf{F}(\mathcal{X})$,

$$\delta\mathbf{F} \approx \frac{\partial \mathbf{F}}{\partial \mathcal{X}}\delta\mathcal{X} = DF\delta\mathcal{X}. \tag{25}$$

Assume that $\delta\mathcal{X}$ is a zero-mean random vector with autocovariance

$$\mathrm{cov}_{\delta\mathcal{X}} = E[\delta\mathcal{X}\delta\mathcal{X}^T] \tag{26}$$

Thus, the expected value of the norm-squared of the constraint vector, from Equation (25), is

$$E[\|\mathbf{F}\|^2] = E[(DF\delta\mathcal{X})^T(DF\delta\mathcal{X})] = E[tr((DF\delta\mathcal{X})(DF\delta\mathcal{X})^T)] \tag{27}$$

where $tr(\cdot)$ denotes the matrix trace operator. The DF matrix is deterministic by the assumption that it is invariant near the solution, so the expected value can be rewritten,

$$E[\|\mathbf{F}\|^2] = tr(\mathrm{cov}_{\delta\mathcal{X}}DF^TDF) \tag{28}$$

Assuming that $\delta\mathcal{X}$ is independent, identically distributed, Equation (28) simplifies to

$$E[\|\mathbf{F}\|^2] = \sigma^2_{\delta\mathcal{X}}tr(DF^TDF) \tag{29}$$

where $\sigma^2_{\delta\mathcal{X}}$ is the variance of the random perturbation. Assuming that the guess is in the vicinity of a solution, and that the deviation from the solution is a zero-mean, independent, identically distributed random vector, yields a reasonably simple model for the norm-squared error.

The error associated with a trajectory inherently depends on the enforced constraints; assuming a single set of constraints further simplifies the error prediction model. Excepting impulsive maneuvers, full-state continuity is required for all trajectories and, therefore, is selected as a representative set of constraints. For this constraint set, given $t_i = t_{i-1} + \tilde{\tau}_{i-1}$,

$$tr(DF^T DF) = (k-1)6 + \sum_{i=2}^{k} tr(\phi(t_i, t_{i-1})^T \phi(t_i, t_{i-1})) + (\dot{\mathbf{X}}_i(t_i))^T (\dot{\mathbf{X}}_i(t_i)) \tag{30}$$

The first term of the above summation is rewritten as a sum of eigenvalues,

$$tr(\phi(t_i, t_{i-1})^T \phi(t_i, t_{i-1})) = \sum_{j=1}^{6} \lambda_j (\phi(t_i, t_{i-1})^T \phi(t_i, t_{i-1})) \tag{31}$$

where λ_j is the j^{th} eigenvalue of the enclosed matrix. Since the enclosed matrix is the square of the state transition matrix, it is inherently positive semi-definite; thus, all eigenvalues are nonnegative, and the maximal eigenvalue is a lower bound on the sum. The maximal eigenvalue is typically on the order of hundreds and, therefore, dominates the constant term in Equation (30). The final term in Equation (30) is associated with deviations in the time of each patch point. Many trajectories can be converged using fixed-time targeters, for which this term is irrelevant. Assuming that the maximal eigenvalue of the square of the state transition matrix dominates in Equation (30), and employing Equation (23), Equation (29) becomes

$$E[\|\mathbf{F}\|^2] \approx \sigma_{\delta\mathcal{X}}^2 \sum_{i=2}^{k} \exp(2\lambda_{max}(t_i, t_{i-1})\tilde{\tau}_{i-1}) \tag{32}$$

This result is employed for the error prediction model, i.e.,

$$\epsilon = \sigma_{\delta\mathbf{X}}^2 \sum_{i=2}^{k} \exp\left(2\lambda_{max}(t_i, t_{i-1})\tilde{\tau}_{i-1}\right) \tag{33}$$

A variety of error prediction strategies can be employed and the placement strategy does not depend upon the choice. This error prediction model incorporating the LLE value was validated for the design examples in this investigation.[3]

Patch Point Placement

The objective of the patch point placement algorithm is the identification, without excessive computation, of an arrangement for patch points along the path that yields a relatively small predicted error. Given a particular arrangement of the patch points, Equation (33) supplies a computationally efficient prediction of the error, relying primarily upon the LLE. Beginning with only one patch point at the beginning of the trajectory–and at any maneuver points–the algorithm iteratively adds a single patch point at the location that offers the greatest reduction in predicted error.

The process can be visualized against the backdrop of the LLE surface, and begins by placing a patch state at the initial state along the arc. Continuing with the sample trajectory in Figure 2, since $t_f = 5$, evaluation of the LLE equation at $t = 0$ requires that $\tilde{\tau} \leq 5$. This fact yields the vertical line, of varying intensities, at $t = 0$ on the LLE surface in Figure 7. A similar logic is applied to any vertical line along the LLE surface, noting that the upper bound is always identified as the

diagonal line defined by $\tilde{\tau} \leq t_f - t$. Because the accuracy of the LLE approximation hinges on a maximal value of the magnitude of $\tilde{\tau}$, boundary values along this diagonal line supply the most accurate representation of the LLE for any given point along the arc. Thus, in selecting the location of the next candidate patch state, it is most efficient to seek the darkest regions along this diagonal line. As such, a candidate point is selected at $t = 3$, which then splits the trajectory into two segments, one spanning for 3 time units, and the other for 2. The data in the plot is still valid along the first segment, but the diagonal boundary must be re-defined because the duration of this segment is shortened to 3. Thus, $\tilde{\tau} \leq 3 - t$ for $0 \leq t \leq 3$. Subsequently, in selecting the next candidate patch state, one along this first segment, dark regions along this new diagonal line are desirable. And so the process is repeated until the desired number of patch states is established. Naturally, this selection scheme, thus far, is graphical in nature, that is, the selection thus far assumes visual inspection of the LLE surface is possible. However, the goal is an automated process suitable for offline non-interactive determination of the patch point locations. Thus, the prior approximate error expression is incorporated. In an automated process, a series of candidate patch points are placed along the boundary defined by the initial diagonal in Figure 8. Then, the error associated with each point is estimated using Equation (33). The predicted errors corresponding to each candidate patch point are systematically compared, and the patch point that results in the greatest reduction in predicted error is automatically selected as the new patch point. The process is repeated in selecting the next candidate point, by re-defining the diagonal boundary accordingly, until (i) an acceptable overall predicted error is achieved; (ii) some maximum number of patch points are placed; (iii) all candidate patch points yield an increase in the predicted error.

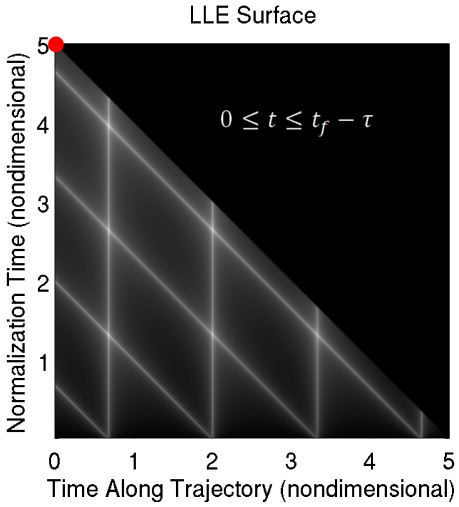


Figure 7. Single Patch Point on LLE Surface

As observed by Haapala,[15] configurations with all patch points located in troughs of the LLE contour–i.e., dark regions of the LLE surface–tend to yield more desirable convergence behavior than arrangements with one or more patch points located on peaks of LLE contours–or, in white regions on the LLE surface. Thus, arrangements, or configurations, with all patch points in darker regions of the LLE surface tend to yield smaller predicted errors, and the automated patch point algorithm converges on one such arrangement. A more exhaustive search, perhaps employing dynamic programming, is possible rather than placing one patch point at a time, but would be far more computationally intensive. For possible onboard applications, the optimality of a global search is

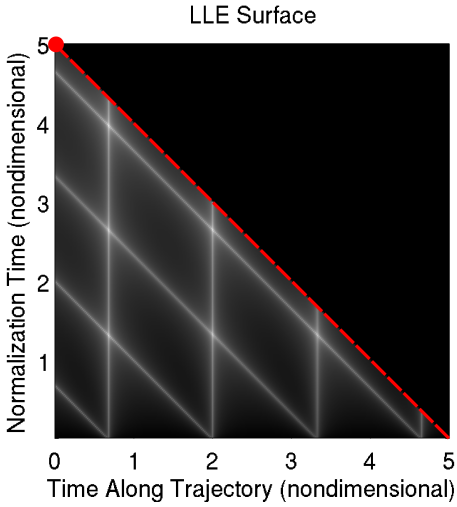


Figure 8. Possible Locations of New Patch Point

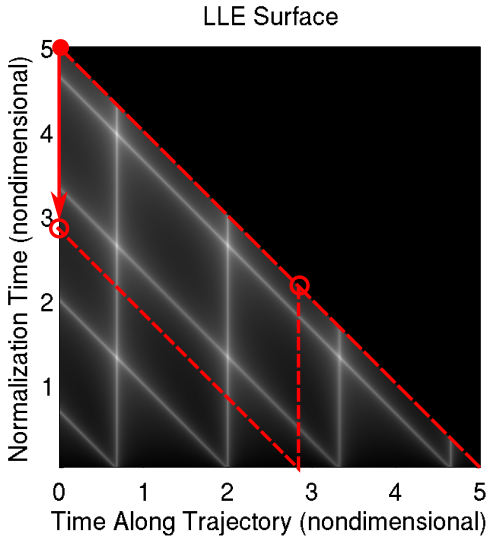


Figure 9. Possible Arrangement of Two Patch Points

outweighed by the necessity for computational efficiency.

RESULTS

The patch point placement algorithm was constructed for two different corrections schemes and applied to a number of design examples; two of these examples are discussed here. The first example supports the hypothesis that the patch point placement algorithm is effective for two-level correctors, although the placement algorithm was originally derived for a parallel-shooting targeter. The second design example employs a parallel shooter for a plausible design example, in which initial guesses are constructed in lower-fidelity models and corrected in the CR3BP.

Patch Point Placement and the Two-Level Corrector

The convergence behavior of a two-level corrector in the design example for the baseline path in Figure 2 is compared for different patch point distributions. One simple placement strategy is just a set of nodes placed at equal time intervals along the path. Convergence of the targeter is then compared to a distribution from the automated patch point placement algorithm. The goal is a path from the initial state to a specified location near the Earth after a given time interval. For the trajectory in Figure 2, the patch points are placed by (i) manual introduction of equal time intervals, consistent with the placement in Figure 4; and (ii) using the automated placement algorithm. Continuity is enforced, and the initial position is fixed. The terminal position is constrained to a specific altitude relative to the terminal altitude on the baseline–218,000 km–but is placed along the same radial vector from the center of the Earth; thus, the terminal position is also fully constrained.

This example indicates that the automated algorithm is much more reliable than equal-time placement; the results are reported in Table 1 in terms of the number of iterations of the targeting algorithm that are required for convergence to a specified tolerance. The number of patch points is constrained to five in one test, and eight in another test, to facilitate a direct comparison between the equal-time and automated placement approaches; note that this requirement supersedes the stopping criteria identified in the patch point placement discussion. The equal-time placement method is particularly sensitive to the number of patch points. Over a range of terminal altitudes, the equal time patch points frequently fail to converge within 15 permitted iterations. The automated algorithm is much more reliable. In cases where both sets yield convergence, the automated process requires an equal or fewer number of iterations. In the cases where the automated algorithm failed, patch points can be added to improve performance. Thus, this automated approach is promising for trajectory design in sensitive dynamical regimes.

Earth-Return Trajectory Design

A second design example demonstrates the benefit of the patch point placement algorithm for transitioning an initial guess from a two-body solution to a higher-fidelity, non-Keplerian dynamical model. The trajectory, inspired by a potential trajectory for the Orion capsule as described by Marchand et al.,[4] originates in a circular lunar orbit. A Moon-to-Earth return is accomplished in four steps: (1) the vehicle begins in the circular lunar orbit; (2) an apolune-raising maneuver establishes an elliptical orbit about the Moon; (3) a plane-change maneuver re-orients the orbital plane about the Moon for a more desirable lunar escape maneuver; (4) a third maneuver places the spacecraft on a transition arc toward the Earth and ensures that the trajectory satisfies Earth arrival constraints. This maneuver schedule specifies a minimum of four distinct trajectory segments, with the first segment fixed by the original spacecraft orbit. Prior to the final maneuver, initiating the lunar departure, the motion is dominated by lunar gravity. Thus, Moon-centered two-body arcs supply a strong initial guess for the first three segments. The final segment, however, transitions from an arc shaped most significantly by the Moon to one that is primarily influenced by the Earth's gravitational field.

An Earth-centered elliptical orbit is generated as an initial guess for the final segment, but propagating the state from apogee in the CR3BP rapidly impacts the Moon. While techniques exist that would produce a more accurate initial guess, it is possible to converge the Moon-centered and Earth-centered elliptical initial guesses if patch points are located appropriately. The initial guess for the state corresponding to the first patch point along the trans-Earth arc is defined from the lunar ellipse, as if no maneuver occurs, to avoid immediately impacting the Moon. Additional patch points are

Table 1. Convergence behavior of the two-level corrector employing automated placement algorithm; number of iterations required to meet a tolerance of $\|\mathbf{F}\| \leq 10^{-12}$ nondimensional units

	5 Patch Points		8 Patch Points	
Change in Altitude (km)	Equal-Time	Automated	Equal-Time	Automated
-8000	Failed	6	7	5
-7000	11	6	6	5
-6000	11	10	6	5
-5000	Failed	5	6	5
-4000	Failed	5	6	5
-3000	Failed	5	7	5
-2000	Failed	5	5	4
-1000	Failed	4	5	4
1000	Failed	13	6	4
2000	Failed	9	5	4
3000	Failed	5	10	5
4000	Failed	5	Failed	5
5000	Failed	6	16	5
6000	10	10	11	5
7000	Failed	Failed	9	5
8000	Failed	Failed	Failed	5

placed using the patch point placement algorithm, with their states sampled from the Earth-centered elliptical solution; however, additional information is required for $\sigma_{\delta\mathbf{X}}$. To obtain a model for $\sigma_{\delta\mathbf{X}}$, the ellipse is sampled at 100 points equally-spaced in time, propagated in the CR3BP, and then converged with a parallel shooter. After obtaining a converged solution, the difference between the converged state and the initial guess is computed and the norm of that difference is plotted in Figure 10; a function in terms of distance from the Moon is identified to resemble this curve and to serve as a model for $\sigma_{\delta\mathbf{X}}$. Applying the automated patch point placement algorithm, and allowing 20 patch points along the trans-Earth arc, converges to a continuous path after only 8 iterations of a parallel shooter. The initial guess, propagated in the CR3BP dynamical model, and the converged solution appear in Figures 11 and 12.

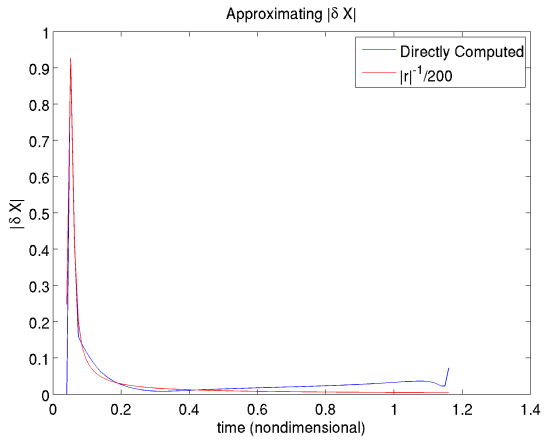
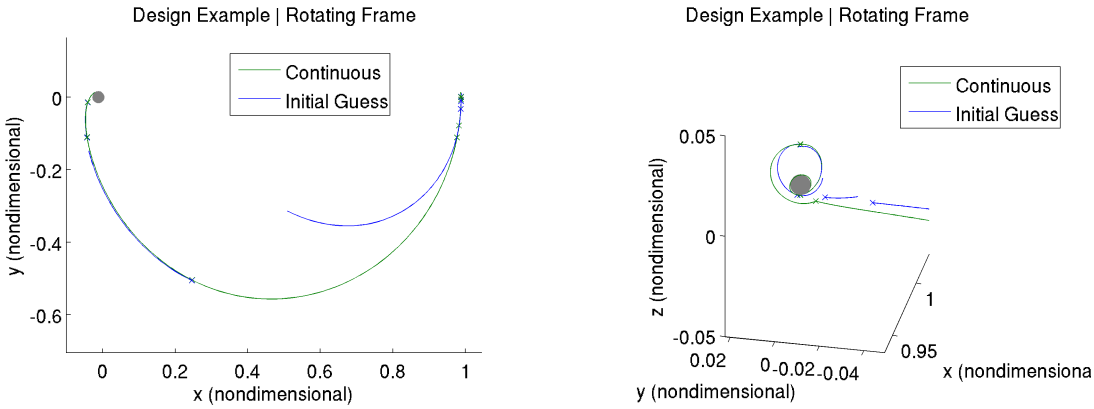
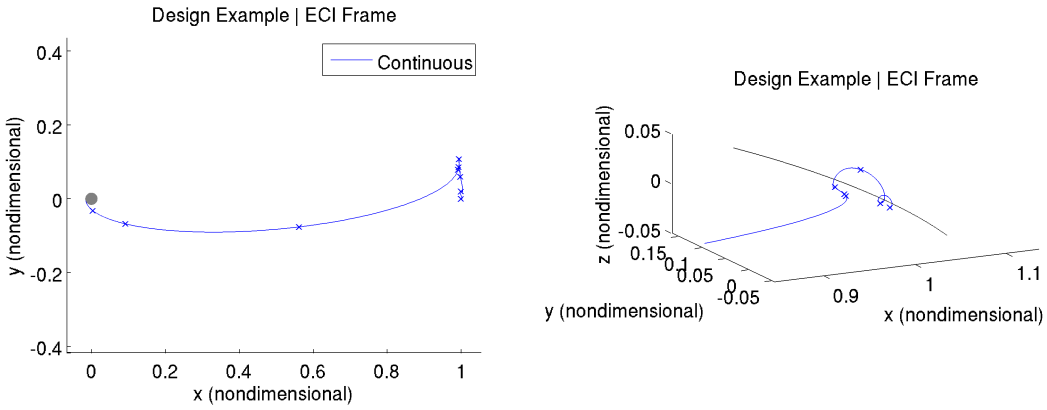


Figure 10. Approximating $\sigma_{\delta \mathbf{X}}$



(a) Rotating Frame (b) Rotating Frame - Near Moon

Figure 11. Initial Guess and Converged Solution



(a) Inertial Frame (b) Inertial Frame - Near Moon

Figure 12. Converged Solution Viewed in Inertial Frame

SUMMARY AND CONCLUSIONS

The preliminary results from this investigation offer a strong indication that an automated patch state selection strategy, based on LLE values, provides significant improvements to the overall performance of linear targeting processes. The candidate error prediction model selected, though not all-encompassing or in final form, further suggests such approximations are valid and simplify the automation process. The current approach is designed about a specific type of targeting process, which naturally suggests that further study is warranted before the results can be generalized. However, the improved selections that result even from this simplified model have, thus far, benefited both two-level targeting and parallel shooting strategies in problems as complex as the Orion trans-Earth injection.

ACKNOWLEDGMENTS

This work was completed, in large part, at Purdue University under NASA Grant No. NNX12AC02G. Support from the School of Aeronautics and Astronautics is acknowledged, as well as the computational facilities in the Eliasen Visualization Laboratory. The authors would like to further extend their gratitude to Ph.D. student Cody Short for his assistance with numerical approximations to Lyapunov exponents, and to Dr. Tom Pavlak for his insights into targeter design.

REFERENCES

[1] C. V. Samaras and V. Tsaoussidis, "Design of Delay-Tolerant Transport Protocol (DTTP) and its evaluation for Mars," *Acta Astronautica*, Vol. 67, 2010, pp. 863–880.

[2] K. C. Howell and H. J. Pernicka, "Numerical Determination of Lissajous Trajectories in the Restricted Three-Body Problem," *Celestial Mechanics*, Vol. 41, 1988, pp. 107–124.

[3] G. K. Harden, "Automated Node Placement for Autonomous Spacecraft Trajectory Targeting," Master's thesis, Purdue University, 2013.

[4] B. G. Marchand, M. W. Weeks, C. W. Smith, and S. Scarritt, "Onboard Autonomous Targeting for the Trans-Earth Phase of Orion," *Journal of Guidance Control and Dynamics*, Vol. 33, 2010, pp. 943–956.

[5] R. L. Anderson, M. W. Lo, and G. H. Born, "Application of local Lyapunov exponents to maneuver design and navigation in the three-body problem," *AAS/AIAA Astrodynamics Specialist Conference*, August 2003.

[6] K. C. Howell and M. Kakoi, "Transfers between the Earth-Moon and Sun-Earth Systems using Manifolds and Transit Orbits," *IAF 56th International Astronautical Congress*, October 2005.

[7] K. Howell and M. Kakoi, "Transfers between the Earth-Moon and Sun-Earth Systems using Manifolds and Transit Orbits," Vol. 59, 2006, pp. 367–380.

[8] F. C. F. Venditti, E. M. Rocco, A. F. B. d. Almeida Prado, and A. Suhkanov, "Gravity-assisted maneuvers applied in the multi-objective optimization of interplanetary trajectories," *Acta Astronautica*, Vol. 67, December 2010, pp. 1255–1271.

[9] V. Szebehely, *Theory of Orbits: The Restricted Problem of Three Bodies*. London, 1967.

[10] H. B. Keller, *Numerical Methods for Two-Point Boundary-Value Problems*. Waltham, Mass.: Ginn-Blaisdell Pub. Co., 1968.

[11] B. G. Marchand, K. C. Howell, and R. S. Wilson, "An Improved Corrections Process for Constrained Trajectory Design in the n-Body Problem," *Journal of Spacecraft and Rockets*, Vol. 44, July-August 2007, pp. 884–897.

[12] D. Grebow and K. C. Howell, "Generating Periodic Orbits in the Circular Restricted Three-Body Problem with Applications to Lunar South Pole Coverage," Master's thesis, Purdue University, 2006.

[13] C. R. Short, "Lagrangian Coherent Structures in the Circular Restricted Three-Body Problem," Master's thesis, Purdue University, 2010.

[14] C. Short and K. Howell, "Lagrangian Coherent Structures in Various Map Representations for Application to Multi-Body Gravitational Regimes," Vol. 94, No. 2, 2014, pp. 592–607.

[15] A. Haapala and K. Howell, "Automated Node Placement Strategies," Progress Report NNX12AC02G, prepared for NASA Johnson Space Center, November 2011.