AAS-22-016

RAPID LOCAL TRAJECTORY OPTIMIZATION IN CISLUNAR SPACE

Spencer Boone* and Jay McMahon†

This paper presents an algorithm for rapid local trajectory optimization around a reference using the reference trajectory's higher-order state transition tensors (STTs) to approximate the local dynamics, and differential dynamic programming (DDP) to optimize the controls. The algorithm is applied to a variety of transfers in cislunar space within the context of the Earth-Moon circular restricted three-body problem. Continuous-thrust transfers both arriving at and departing near-rectilinear halo orbits are examined. The STT/DDP algorithm is shown to perform more accurately than linearized methods for these types of transfers, while requiring significantly less computational power than standard numerical optimization methods. The method is promising for rapidly conducting large-scale tradeoff analyses for transfers with varying or uncertain departure and arrival conditions. In addition, the proposed method could be used for on-board guidance applications with limited computational resources, or time-critical trajectory re-planning scenarios.

INTRODUCTION

There has been significant interest in the space community in operating spacecraft in cislunar space, with several future missions planned to operate in this regime, including the Lunar Gateway[1] and the various cubesats flying on the Artemis-1 mission[2], among others. There has been particular interest in operating satellites with low-thrust propulsion systems in this regime. However, due to the complex and highly nonlinear multi-body dynamics in these regimes, optimizing trajectories can be a time-consuming and sensitive process. In this work, we will present a method that can efficiently and accurately optimize trajectories in the vicinity of a reference, which could enable more efficient on-board guidance, or expedite large scale mission design analyses.

Currently, most continuous-thrust spacecraft trajectory optimization algorithms are run on the ground with powerful computers. For many missions, including the Dawn mission[3], any updates to the trajectory are generally also planned on the ground with numerically intensive optimization algorithms, and the commands are subsequently uploaded to the spacecraft. For the types of transfers that will be conducted in cislunar space, the timeframe for transfers between orbits is on the order of days (as opposed to weeks or months); as such, this ground-based procedure could be a limiting factor for replanning complex transfers in reaction to navigation or maneuver execution errors. An on-board maneuver planning capability may be required. The existing methods that have been used for on-board guidance have typically relied on linearizations around a reference[4]. However, in cislunar space, the dynamics are highly nonlinear, and the convergence region for linearized methods may not be large enough to account for the expected errors over the course of a trajectory.

[*] Ph.D. student, Smead Department of Aerospace Engineering Sciences, University of Colorado Boulder

[†] Assistant Professor, Smead Department of Aerospace Engineering Sciences, University of Colorado Boulder

M. Sandnas, D. B. Spencer (eds.), *Proceedings of the 44th Annual American Astronautical Society Guidance, Navigation, and Control Conference, 2022*, Advances in the Astronautical Sciences 179, https://doi.org/10.1007/978-3-031-51928-4_5

In addition, particularly for the large number of CubeSats that are proposed to be operating in cislunar space [5, 6, 7], there may be uncertain or varying departure and arrival conditions. This may necessitate conducting large-scale analyses such as Delta-V 99 (DV99) analyses in order to ensure that the proposed trajectories are sufficiently robust to the varying conditions of the transfer. Currently, these analyses are mostly conducted using either numerical Monte Carlo simulations [8], or linearized techniques such as JPL's ADAM maneuver analysis system [9]. For highly nonlinear systems such as the Earth-Moon system, using these numerical techniques may become prohibitively expensive (particularly if the trajectory design timeframe is relatively short, as it often would be for a CubeSat), while using the linearized methods may result in inaccurate or suboptimal trajectories. A capability to rapidly generate accurate and near-optimal trajectories in the vicinity of some reference could therefore be highly beneficial.

One common feature of many existing ground-based trajectory optimization algorithms is that they rely on the first (and often second) order derivatives of the dynamics to understand the local dynamical behavior and update controls guesses accordingly[10, 11]. When using full-fidelity models of the dynamics with realistic perturbations, no exact analytic representations of the dynamics and these derivatives are available. Running these algorithms therefore either requires repeated integrations of the dynamics (which can be prohibitively slow in real-time or on a flight computer), or the use of a lower-fidelity model to approximate the true dynamics.

These issues can be addressed through the use of higher-order state transition tensors (STTs) to approximate complex dynamics. STTs can be used to obtain accurate approximations of the sensitivities of a dynamical system around a reference trajectory; in fact, they can be thought of as a higher-order extension of the first-order state transition matrix (STM). The use of higher-order derivatives for spacecraft trajectory design and guidance is not new[12, 13]; however, as a novel development, the authors of this paper showed in Ref. [14] that the first and second-order derivatives of an STT-approximated dynamic system can be exactly expressed as a function of the STTs. This allows us to run a fully analytical approximation of any optimization algorithm that requires first and/or second-order derivatives of the dynamics. As minimal modifications are required to the nominal algorithm, it can accommodate most stage constraints and penalty methods that would be included in the standard algorithm. In addition, the method is agnostic to the dynamics; thus, any number of perturbations can be included in the model, and the method would be usable with a full-fidelity ephemeris model for the Earth-Moon system.

In Ref. [14], the STT approximation method was combined with differential dynamic programming (DDP), a second-order optimization algorithm which has been used in low-thrust trajectory optimization[10]. The resulting algorithm, referred to as STT/DDP, was shown to converge on similar solutions to the numerical DDP algorithm at a fraction of the computational cost. In this paper, the STT/DDP method is applied to a number of high-interest transfer in cislunar space. In particular, transfers arriving at and departing near-rectilinear halo orbits (NRHOs) are generated and analyzed. The STT/DDP method is shown to be a useful tool for expediting sensitivity analysis studies during the low-thrust mission design process, and could eventually find use on an on-board guidance system when linearized methods are not sufficiently accurate.

The paper is organized as follows. First, we describe the theory behind higher-order STTs in the context of spacecraft dynamics. We then provide a brief overview of the DDP algorithm for the optimization of spacecraft trajectories, and describe the modifications needed to replace numerical integration with evaluations of the reference STTs. We apply this method to the computation of near-optimal continuous-thrust trajectories in cislunar space. First we use the algorithm to generate a range of transfers from a distant retrograde orbit (DRO) to a near-rectilinear halo orbit (NRHO) in the Earth-Moon system. Next, we use the algorithm to generate transfers from an NRHO to a variety of geosynchronous orbit (GSO) configurations. The method is able to compute these new trajectories in a matter of seconds for very large deviations in both the initial conditions and the final target state.

STATE TRANSITION TENSORS

As stated in the introduction, state transition tensors (STTs) are effectively a higher-order extension of the commonly-used state transition matrix (STM); in fact, the STM can be thought of as the first-order STT. The basics of their derivation are repeated below, reproduced based on Park and Scheeres [15].

The solution to a dynamic system evolution can be described through its solution flow,

$$\mathbf{x}(t) = \boldsymbol{\phi}(t; \mathbf{x}_0, \hat{\mathbf{u}}_{(t,t_0)}, t_0) \tag{1}$$

where $\hat{\mathbf{u}}_{(t_f,t_0)}$ is the time history of all nominal (or reference) controls applied over the interval $[t_0, t_f)$. Note that, given a control history, the nominal controls can be regarded as part of the "natural" dynamical system so that their effect is wrapped up within the STTs. The STTs are partial matrices of this solution flow with respect to the initial conditions. Thus, the STTs of order p can be defined as

$$\phi_{(t,t_0)}^{i,\gamma_1 \dots \gamma_p} = \frac{\partial^p \boldsymbol{\phi}^i(t; \mathbf{x}_0, \hat{\mathbf{u}}_{(t,t_0)}, t_0)}{\partial x_0^{\gamma_1} \dots \partial x_0^{\gamma_p}} \tag{2}$$

Superscripts indicate components of a vector, matrix, or tensor. The order of a tensor can be determined by the number of superscript indices. Subscripts indicate the time of the vector, or for a matrix or tensor, (t, t_0) indicates a mapping from t_0 to t. Index (Einstein summation) notation is used heavily throughout this paper; repeated indices indicate summation, e.g.,

$$\frac{1}{2}\phi^{i,\gamma_1\gamma_2}\delta x_0^{\gamma_1}\delta x_0^{\gamma_2} = \sum_{\gamma_1=1}^{n}\sum_{\gamma_2=1}^{n}\frac{1}{2}\phi^{i,\gamma_1\gamma_2}\delta x_0^{\gamma_1}\delta x_0^{\gamma_2} \tag{3}$$

The solution of the variation of the state at time t_f around the nominal state can be approximated with STTs as [15]

$$\delta x^i \simeq \sum_{p=1}^{m}\frac{1}{p!}\phi_{(t,t_0)}^{i,\gamma_1 \dots \gamma_p}\delta x_0^{\gamma_1} \dots \delta x_0^{\gamma_p} \tag{4}$$

This corresponds to a Taylor series expansion about the reference trajectory including terms up to order m. As m increases, the approximation for δx^i will generally become more and more accurate, up to the limit $m \to \infty$, where the approximation converges on the true solution.

The integration of the higher-order STTs represents a significant computational burden, and is certainly not feasible to conduct at a large scale on a flight computer, let alone in real time. We can, however, integrate the higher-order STTs of a reference trajectory, and use the resulting STTs to analytically predict the effect of any state deviation on the final state. Because no further integrations of the dynamics are required, these mappings can be performed very efficiently. There will be a convergence region around the reference where the STT mappings are accurate approximations of the true dynamics; outside this region, the STTs will no longer be accurate. The size of this region will depend on the order of STTs included in the approximation.

Remark 1 (Derivatives of STT state deviation) *The 1st and 2nd-order derivatives of an STT-propagated state deviation $\delta\mathbf{x}$ at time t, with respect to the state deviation $\delta\mathbf{x}_0$ at time t_0, can be expressed analytically solely as a function of the reference STTs mapping from t_0 to t:*

$$\frac{\partial(\delta x^i)}{\partial(\delta x_0^{\gamma_1})} = \phi_{(t,t_0)}^{i,\gamma_1} + \sum_{p=2}^{m} \frac{1}{(p-1)!} \phi_{(t,t_0)}^{i,\gamma_1\gamma_2\cdots\gamma_p} \delta x_0^{\gamma_2} ... \delta x_0^{\gamma_p} \tag{5}$$

$$\frac{\partial^2(\delta x^i)}{\partial(\delta x_0^{\gamma_1})\partial(\delta x_0^{\gamma_2})} = \phi_{(t,t_0)}^{i,\gamma_1\gamma_2} + \sum_{p=3}^{m} \frac{1}{(p-2)!} \phi_{(t,t_0)}^{i,\gamma_1\gamma_2\gamma_3\cdots\gamma_p} \delta x_0^{\gamma_3} ... \delta x_0^{\gamma_p} \tag{6}$$

DIFFERENTIAL DYNAMIC PROGRAMMING

The analytical derivative properties described in Remark 1 can be used to construct an analytical optimization algorithm using differential dynamic programming (DDP). In this section we will briefly describe DDP, and outline the modifications and additions to the standard DDP algorithm that were used to generate the results in this paper. For conciseness we will only present equations for the components that are modified for the STT/DDP method. For a complete derivation of DDP, readers can refer to Jacobson and Mayne[16] and Lantoine and Russell[10]. In addition, a more thorough derivation of the STT/DDP algorithm can be found in Ref. [14].

DDP is a second-order local dynamic programming algorithm, in which a quadratic approximation of the cost-to-go around a trajectory is computed and correspondingly, a local linear-feedback controller of the form $\mathbf{u}_k = \bar{\mathbf{u}}_k + B_k\delta\mathbf{x}_k$ is obtained. DDP consists of successive backward and forward sweeps. The trajectory is discretized into $N+1$ stages, and the backward sweep solves the sequence of subproblems that minimizes the cost-to-go from stage $k = N, N-1, ..., 0$ to obtain a prediction for the control update at each stage $\delta\mathbf{u}_k$. In the forward pass, the dynamics are re-integrated, the new control updates are applied and a new quadratic expansion is performed around the resulting trajectory. Terminal constraints can be adjoined to the cost function using a constant vector of Lagrange multipliers λ. Iterations are repeated until the expected reduction in the cost function is below a pre-specified tolerance ϵ_{opt}, and terminal constraints are satisfied within the tolerance ϵ_{feas}.

This DDP implementation borrows several developments from the algorithm developed by Lantoine and Russell[10] for spacecraft trajectory optimization, deemed hybrid differential dynamic programming (HDDP), namely the use of the state transition matrix and tensor to propagate the cost-to-go through stages during the backward sweep. For clarity, we will combine the state vector $\mathbf{x}$ and control vector $\mathbf{u}$ into an augmented state vector $\mathbf{X}^T = [\mathbf{x}^T\mathbf{u}^T]$. We can then write the equations for the stage cost-to-go derivatives from the backward sweep for stage k as

$$J_{X,k} = L_{X,k} + \Phi^T J^{\star}_{X,k+1} \tag{7}$$

$$J_{\lambda,k} = J^{\star}_{\lambda,k+1} \tag{8}$$

$$J^{i,a}_{XX,k} = L^{i,a}_{XX,k} + J^{\star\gamma_1,\gamma_2}_{XX,k+1}\Phi^{\gamma_1,i}\Phi^{\gamma_2,a} + J^{\star\gamma_1}_{X,k+1}\Phi^{\gamma_1,ia} \tag{9}$$

$$J_{\lambda\lambda,k} = J^{\star}_{\lambda\lambda,k+1} \tag{10}$$

$$J_{X\lambda,k} = \Phi^T J^{\star}_{X\lambda,k+1} \tag{11}$$

where Φ is the augmented STM and STT, L_k is local cost at stage k, J_k is the cost-to-go at stage k, and $J^{\star}_k$ is the cost-to-go at stage k after the control update has been applied during the backward sweep, i.e.

$$J_k = L_k + J^*_{k+1} \tag{12}$$

$J_{X,k}$ refers to the derivative of the cost-to-go at stage k with respect to the augmented state vector $\mathbf{X}$, and $J_{\lambda,k}$ refers to the derivative with respect to the vector of Lagrange multipliers $\boldsymbol{\lambda}$.

Augmented Lagrangian Function

As in Lantoine and Russell[10], we add a quadratic penalty parameter σ to place additional weight on the terminal constraints. The augmented cost function then becomes

$$\tilde{J} = J + \boldsymbol{\lambda}^T\boldsymbol{\psi} + \boldsymbol{\psi}^T\sigma\boldsymbol{\psi} \tag{13}$$

where the vector $\boldsymbol{\psi}$ is the terminal constraint vector - the difference between achieved and desired final states. The terminal constraints will be satisfied when $\|\boldsymbol{\psi}\| < \epsilon_{\text{feas}}$.

Trust-Region Subproblem

DDP will often take large steps toward the minimum; if these are not constrained, the steps may lie outside the convergence region of the local quadratic approximation, or may lead to infeasible iterates. Many implementations [10, 17] address this issue by solving a trust-region quadratic subproblem (TRQP) at each stage. An extensive review of trust-region methods is available in Conn et. al.[18]; the methods from this source were found to be sufficient for the problems at hand. See Ref. [14] for a more thorough discussion of the trust-region subproblem. The trust region method also serves to keep successive iterations in proximity to the reference trajectory, and thus within the expected convergence region of the higher-order STTs.

Form of Control and Cost Function

The choice of control affects how the state transition matrices and tensors in Eqns. 7-11 are computed. If we consider a continuous control, additional differential equations would need to be integrated to obtain the augmented STM. To simplify the formulation for the demonstrations in this work, we treat the control vector as an impulsive change in velocity at the beginning of each stage.

$$\mathbf{u}_k = \Delta\mathbf{v}_{t_k} \tag{14}$$

Initially in this work we will use the minimum-energy cost function, which corresponds to the sum of the square of the magnitude of the velocity at each stage. The local cost L_k at stage k becomes

$$L_k = \|\mathbf{u}_k\|^2 = u_{k,x}^2 + u_{k,y}^2 + u_{k,z}^2 \tag{15}$$

Note that this form of the cost function will result in different optimal control policies than using the total fuel usage - the optimal policy will have the controls spread out over all stages as opposed to a bang-bang policy. We will also show that we can use the STT/DDP method to compute trajectories with a different cost function than the reference. For this we use the minimum-fuel formulation of the cost, which can be written as

$$L_k = \|\mathbf{u}_k\| = \sqrt{u_{k,x}^2 + u_{k,y}^2 + u_{k,z}^2 + \epsilon_{ml}} \tag{16}$$

where ϵ_{ml} is a small mass leak term which is necessary to resolve the singularity for the derivatives of the minimum-fuel cost function during coast arcs.

STATE TRANSITION TENSOR DDP

The computation of the first and second-order derivatives, used in Eqns. 7-11 for the backward sweep in each iteration, is the most computationally expensive portion of the numerical DDP algorithm. In order to alleviate this burden, we can integrate the higher-order STTs along a reference trajectory, and subsequently use the exact derivatives of the STT-approximated dynamics (see Remark 1 and Eqns. 5 and 6) to run an "approximation" of the numerical DDP algorithm, with significant improvements in computational time. In this paper we refer to this strategy as STT/DDP.

The formulation of the STT/DDP algorithm is as follows. Given a reference trajectory with an associated reference control history $\hat{\mathbf{u}}$, we can first divide the reference trajectory into $N + 1$ stages, and integrate the higher-order STTs for each stage along this reference. The numerical DDP algorithm is modified by replacing any integrations of the dynamics with evaluations of these STTs. For clarity, we use $\delta\hat{\mathbf{x}}_k$ to refer to state deviations from the *original reference trajectory*. Similarly, $\hat{\phi}^{i,\gamma_1\cdots\gamma_p}_{(t_{k+1},t_k)}$ refers to the reference trajectory STTs of order p, mapping from stage k to stage $k+1$.

For each iteration, the forward pass in the DDP algorithm can be restated as

$$x_{k+1}^i = \left[\hat{x}_{k+1}^i + \sum_{p=1}^{m} \frac{1}{p!} \hat{\phi}^{i,\gamma_1\cdots\gamma_p}_{(t_{k+1},t_k)} \delta\hat{x}_k^{\gamma_1} \ldots \delta\hat{x}_k^{\gamma_p} \right] + u_{k+1}^i \tag{17}$$

Eqns. 5 and 6 can then be used to obtain the first and second-order derivatives of the new trajectory computed in the STT-approximated dynamics during the forward pass:

$$\phi^{i,\gamma_1}_{(t_{k+1},t_k)} \simeq \hat{\phi}^{i,\gamma_1}_{(t_{k+1},t_k)} + \sum_{p=2}^{m} \frac{1}{(p-1)!} \hat{\phi}^{i,\gamma_1,\gamma_2\cdots\gamma_p}_{(t_{k+1},t_k)} \delta\hat{x}_k^{\gamma_2} \ldots \delta\hat{x}_k^{\gamma_p} \tag{18}$$

$$\phi^{i,\gamma_1\gamma_2}_{(t_{k+1},t_k)} \simeq \hat{\phi}^{i,\gamma_1\gamma_2}_{(t_{k+1},t_k)} + \sum_{p=3}^{m} \frac{1}{(p-2)!} \hat{\phi}^{i,\gamma_1,\gamma_2,\gamma_3\cdots\gamma_p}_{(t_{k+1},t_k)} \delta\hat{x}_k^{\gamma_3} \ldots \delta\hat{x}_k^{\gamma_p} \tag{19}$$

$$\tag{20}$$

No further modifications are required. We note again that while Eqns. 18 and 19 are approximations of the true state derivatives, they in fact correspond to the exact derivatives of the STT-approximated dynamics from Eqn. 17.

Enforcing accuracy of the STTs

The STT/DDP algorithm will yield a locally optimal trajectory within the STT-approximated dynamics. If this trajectory lies within the convergence region of the reference's STTs, it will correspond to a nearly-optimal trajectory in the true dynamics. However, if the optimal trajectory obtained from the STT/DDP algorithm is too far from the reference STTs, the STT approximations may not be sufficiently accurate. In this case, the algorithm is no longer useful as it is not an accurate representation of the actual dynamics. In order address this issue, a penalty method was derived in Ref. [14] that ensures that penalizes the trajectory if it is too far from the reference STTs. This method is simple to implement and does not require any significant additional computations. Note that the penalty method may result in sub-optimal trajectories if the optimal trajectory is far from the original reference trajectory, and would not be accurately approximated with the reference STTs. Appendix B contains the details of the derivation of the penalty method.

Resulting feedback policy

The STT/DDP method yields a feedback law of the form $\mathbf{u}_k = \bar{\mathbf{u}}_k + B_k \delta \mathbf{x}_k$. The open-loop component $\bar{\mathbf{u}}_k$ of the feedback law is optimal in the STT-approximated dynamics. However, when applying the resulting control law in the true dynamics, there will be approximation errors of $\mathcal{O}(\varepsilon^{m+1})$ at each stage. Over the course of an entire transfer, these errors may compound if they are too large, and potentially result in a large final error in reaching the desired target. This can be corrected for by applying the feedback law at each stage, where $\delta \mathbf{x}_k$ corresponds to the difference between the STT-predicted state at stage k and the observed or numerically integrated state. Thus, if the STTs perfectly approximate the true dynamics to within numerical precision, we would expect $\delta \mathbf{x}_k$ to be equal to 0 at each stage. Additionally, in an operational trajectory planning setting, the linear feedback law could be used to correct for small navigation errors, off-nominal performance, or unforeseen events.

APPLICATION: EARTH-MOON DRO-TO-NRHO TRANSFER IN THE CIRCULAR RESTRICTED THREE-BODY PROBLEM

We first apply the STT/DDP formulation to a continuous-thrust transfer from a distant retrograde orbit (DRO) to a near-rectilinear halo orbit (NRHO) in the Earth-Moon circular restricted three-body problem (CR3BP). Earth-Moon NRHOs are currently of very high interest for the civilian and military space communities, as it is the planned operating location for the Lunar Gateway [1]. On the other hand, DROs, which are planar in the CR3BP, have been shown have favorable long-term stability properties [19], and could be desirable for missions requiring very low station-keeping costs. Efficient transfers between the two families of orbits may be necessary in the near-future. The state data for the reference NRHO and DRO used in this example is shown in Table 1. The state data for the DRO was obtained from Ref. [20]. The target state for the NRHO is selected to lie at apolune, since this is the least sensitive location on the orbit. A reference trajectory was optimized using the numerical DDP algorithm and minimum-energy cost function, with 100 stages and a transfer time of 2.45 non-dimensional CR3BP units, corresponding to roughly 10.6 days. This reference transfer is shown in Fig. 1. The higher-order STTs of this reference trajectory (up to order $m = 4$) were then integrated separately for each stage and stored. For this case study, we set $\sigma = 10^4$, $\epsilon_{opt} = 10^{-10}$, and $\epsilon_{feas} = 5 \times 10^{-7}$ for both the numerical and STT/DDP algorithms, and used $W = 0.4$ for the penalty term to enforce accuracy of the STT method.

Table 1: DRO-to-NRHO transfer orbit scenario parameters

Orbit	x	y	z	$\dot{x}$	$\dot{y}$	$\dot{z}$	C
Initial DRO	0.983368093	-0.259208967	0.0	-0.351341295	-0.008333464	0.0	2.925
Target NRHO	1.021968177	0.0	-0.18206	0.0	-0.103140143	0.0	3.047

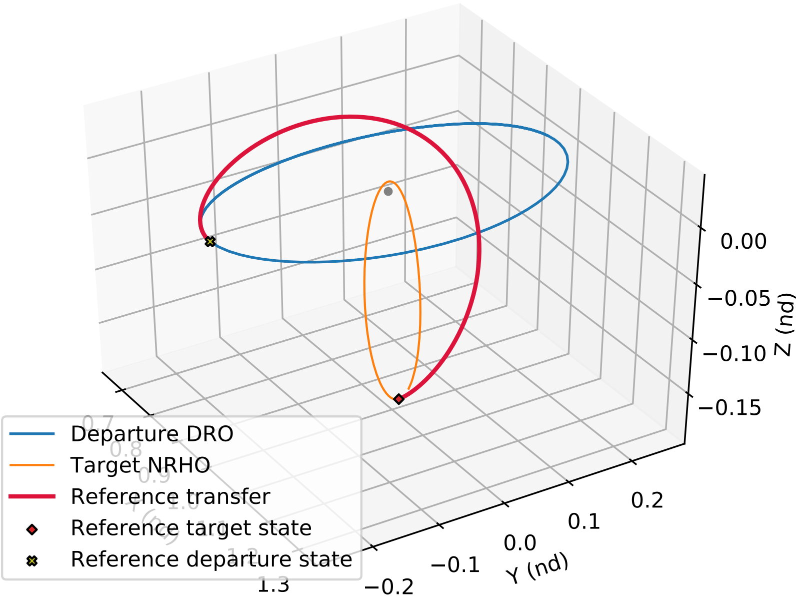

Figure 1: Reference DRO to NRHO transfer in Earth-Moon CR3BP

For the DRO-to-NRHO transfer example, we will demonstrate how the STT/DDP algorithm can be used as a “guidance” scheme, in order to steer a spacecraft with a perturbed initial state back towards the reference target state. Recall that the last iteration of the numerical DDP algorithm results in an optimal linear feedback law that can also be used as a guidance scheme. We will compare the accuracy regions for the linear feedback law and the re-optimized trajectories computed using the STT/DDP algorithm.

The initial DRO state was perturbed by selecting a range of 50 alternative departure points along the DRO, by changing the initial phasing on the orbit between $\delta\tau \in [-0.4, 0.6]$ non-dimensional CR3BP time units. These alternative departure ponits could be operationally desirable in order to achieve the correct phasing between the two orbits, if for example a spacecraft is seeking to rendezvous with another spacecraft along the NRHO. New trajectories were computed for the range of perturbed initial DRO states to reach the reference target NRHO state, using the linear feedback law obtained from numerical DDP, and the STT/DDP method for orders $m \in [2, 3, 4]$. As a comparison, optimal trajectories for each departure point were also computed using the numerical DDP method. The optimized trajectories computed using the STT/DDP method with $m = 3$, along with the reference transfer and departure and target orbits, are shown in Fig. 2.

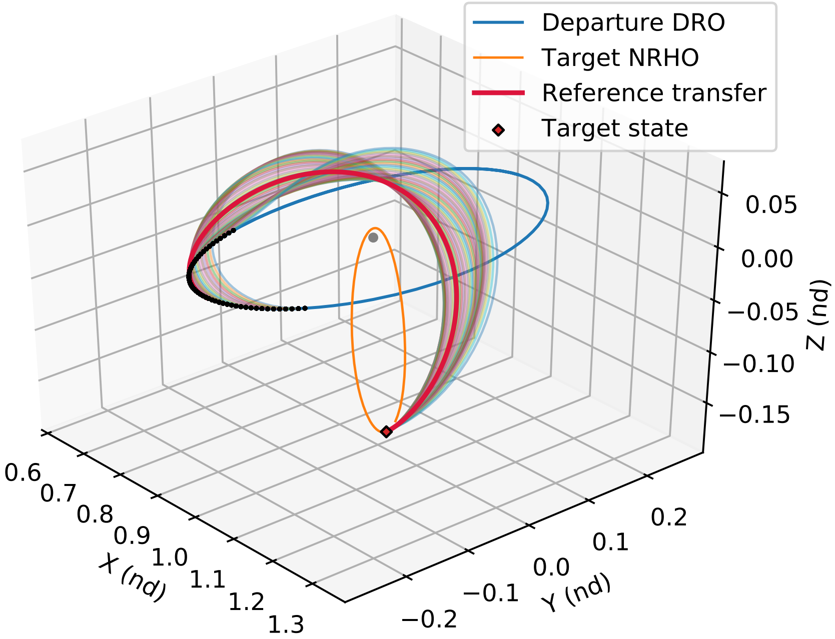

Figure 2: DRO to NRHO transfer with varying departure location in Earth-Moon CR3BP. Trajectories generated using STT/DDP method ($m = 3$)

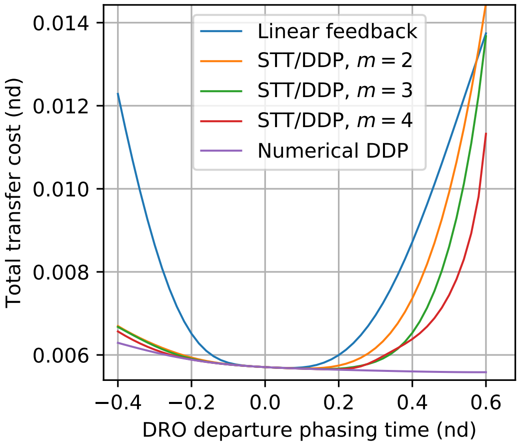

Figure 3: Minimum-energy transfer costs for DRO-to-NRHO transfers

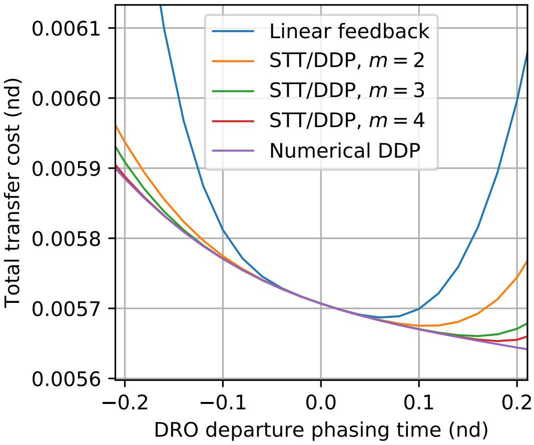

Figure 4: Minimum-energy transfer costs for DRO-to-NRHO transfers (zoomed view)

Table 2: STT/DDP algorithm performance at various orders m for perturbed initial state

Algorithm	Computation time (s)	Number of iterations
STT/DDP, $m = 2$	1.91	17.2
STT/DDP, $m = 3$	2.24	19.2
STT/DDP, $m = 4$	3.38	19.5
Numerical DDP	79.7	16.5

The total transfer costs for the range of alternative departure points for each method are shown in Figs. 3 and 4. The linear feedback law is shown to be accurate for a small region in the vicinity of the reference departure point, but rapidly loses accuracy and results in transfers with very large thrust requirements. The higher-order STT/DDP method clearly results in transfers with far lower thrust requirements. As expected, as the maxmimum order of STT included in the approximation increases, the trajectories approach the “true” optimal trajectories obtained through the numerical DDP.

The average computational time and number of DDP iterations to compute the 50 alternative transfers is shown in Table 2. All orders of the STT/DDP algorithm are at least an order of magnitude faster to run than the numerical DDP algorithm.

We note that all code was written in Python, which is a relatively slow language. Using a compiled language such as C or Julia would result in computational improvements for both algorithms. In addition, all code was run in serial - the computation of the derivatives at each stage could be parallelized to improve performance, though this may not be possible on all hardware configurations. The important point to note from these results is that, when using STTs to approximate the local dynamics, the computational requirements decrease significantly. In addition, the time required to evaluate the reference STTs will not increase as additional perturbations are added to the dynamics, whereas the time required for numerical integration certainly will. Thus, in a full-ephemeris model of the Earth-Moon system, the performance improvements will become even more notable.

APPLICATION: EARTH-MOON NRHO TO GEOSYNCHRONOUS ORBIT TRANSFER IN THE CIRCULAR RESTRICTED THREE-BODY PROBLEM

Next, we apply the STT/DDP method to a transfer from an NRHO to a geosynchronous orbit (GSO) around Earth. Since geosynchronous and geostationary orbits are currently in high use for a number of applications, efficient methods to compute transfers between NRHOs and these orbits are likely to be of interest in the near-future. Entering into geosynchronous orbit requires significant thrusting capability, resulting in highly sensitive and nonlinear trajectories. In addition, the precise geometry of the target geosynchronous orbits may not be exactly known *a priori* when designing a reference trajectory; thus, a method to rapidly generate transfers targeting a variety of orbital configurations could potentially be very useful.

Again, a reference transfer was generated using the numerical DDP algorithm, with a transfer time of 1.32 non-dimensional CR3BP time units, corresponding to roughly 5.7 days. The trajectory was segmented into 200 equally spaced stages - a larger number of stages is required for this transfer due to the high sensitivity of the geosynchronous orbit insertion portion of the transfer. The target reference geosynchronous orbit configuration was chosen to lie in the Earth-Moon plane. The initial and target orbit state parameters are given in Table 3. The higher-order STTs along this reference (up to order $m = 4$) were integrated separately for each stage and stored. We will investigate using these STTs to rapidly compute new trajectories around the reference.

Table 3: NRHO-to-GSO transfer orbit scenario parameters

Orbit	x	y	z	$\dot{x}$	$\dot{y}$	$\dot{z}$
Initial NRHO	1.021968177	0.0	-0.18206	0.0	-0.103140143	0.0
Target GSO	0.080981568	0.0	0.0	0.0	3.113042895	0.0

Varying target parameters

First, we will investigate using the STT/DDP method to target a new GSO configuration from the same initial NRHO state at apolune. The linear feedback law from the numerical DDP method cannot be used in this case because the target state is different from the reference target state. However, the STT/DDP method can easily be used.

50 different target geosynchronous orbits were selected, with the inclination varying from 0 to $15°$ relative to the Earth-Moon plane. The right ascension of the ascending node (RAAN) was allowed to vary between 0 and $360°$, again relative to the Earth-Moon plane. The STT/DDP methods at orders $m \in [2, 3, 4]$ were run to generate new optimal transfers to reach these new targets. The numerical DDP method was also run as a comparison. The 50 transfers are visualized in Fig. 6, for the $m = 4$ method. The thrust profiles for each of the 50 transfers are shown in Figs. 7 and 8. The resulting average cost, computational time, and final state errors are shown in Table 4.

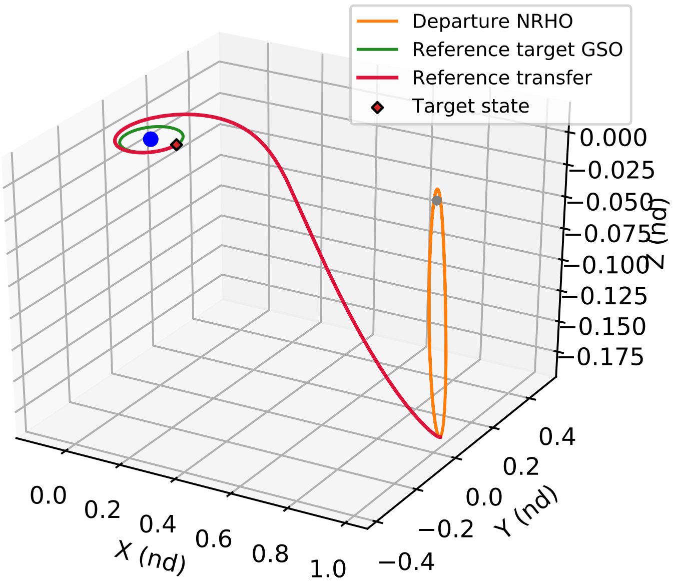

Figure 5: Reference NRHO to GSO transfer in Earth-Moon CR3BP

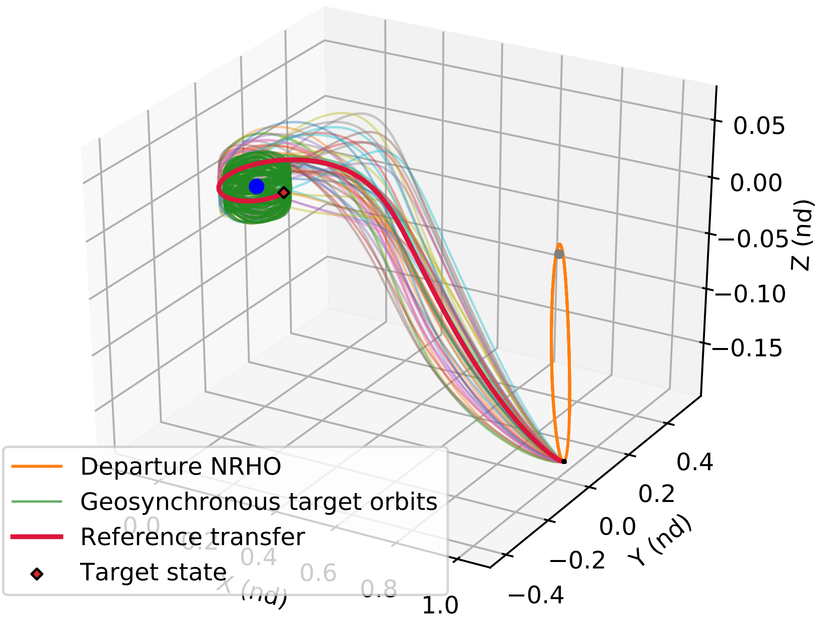

Figure 6: Transfers from NRHO to GSOs with varying parameters in Earth-Moon CR3BP

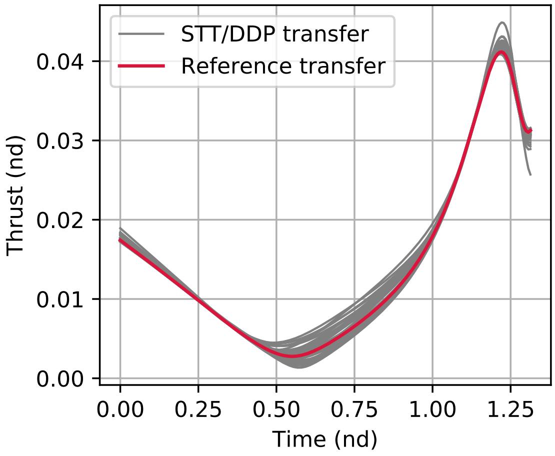

Figure 7: Thrust magnitudes for various NRHO to GSO transfers, generated using STT/DDP method with $m = 4$

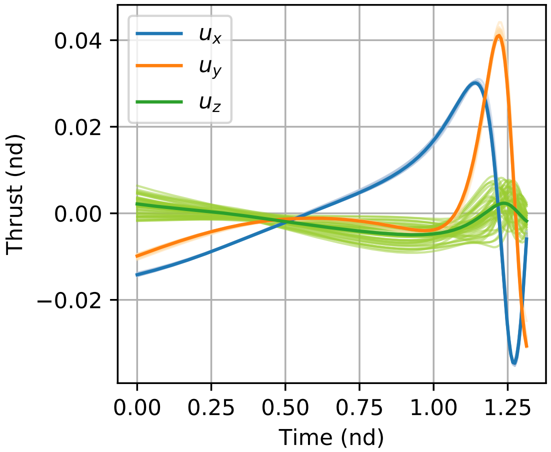

Figure 8: Thrust profiles for various NRHO to GSO transfers, generated using STT/DDP method with $m = 4$. Reference thrust profile in bold.

Table 4: STT/DDP algorithm performance at various orders m for targeting new geosynchronous orbit

Algorithm	Final cost J	Computation time (s)	Number of iterations	Final state error $\|\|\psi\|\|$ with feedback law
STT/DDP, $m = 2$	0.241557	6.71	33.4	2.42e-3
STT/DDP, $m = 3$	0.075993	7.90	31	5.24e-4
STT/DDP, $m = 4$	0.066361	10.94	36.8	9.96e-5
Numerical DDP	0.065921	330.9	36.3	1.38e-12

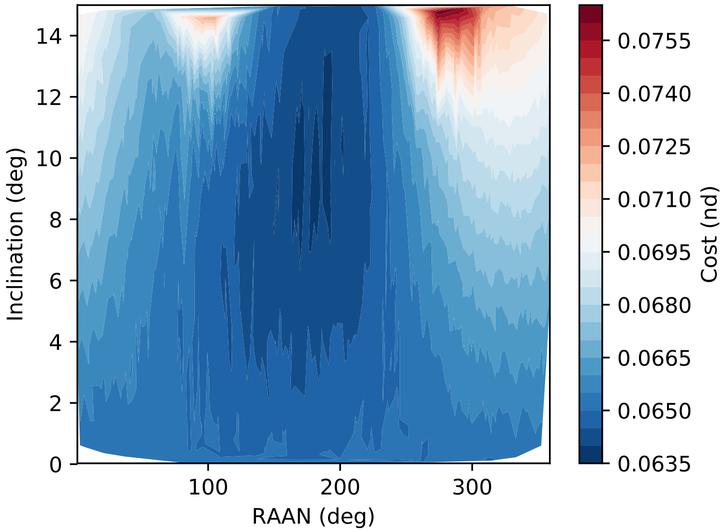

Figure 9: Cost to transfer from NRHO (apolune) to 1000 different GSO configurations, generated using STT/DDP with $m = 4$

As there is a less than 1% difference between the average transfer cost when using the 4th-order STT/DDP method and numerical DDP method, we can conclude that the STT/DDP method represents transfers that are an accurate approximation of the optimal transfers. We can use the STT/DDP method to run large-scale analyses that would be tedious to run with a standard numerical method. Using the same range of inclination and RAAN parameters as detailed above, the STT/DDP method was used to optimize trajectories for 1000 different target conditions. These parameters, and the resulting cost for each combination of parameters, are used to generate the contour plot shown in Fig. 9. This type of large-scale tradeoff analysis can allow a mission designer to rapidly identify reachable orbital configurations, and can be completed around $30\times$ faster when using the STT/DDP method instead of a numerical optimization method. For reference, optimizing these 1000 trajectories took around 3 hours to run using Python code and a standard laptop computed. The equivalent numerical DDP code would have taken nearly 4 days to run.

Transfer from different initial conditions

In order to demonstrate the flexibility of the proposed algorithm, the NRHO-to-GSO transfer scenario was also run with varying initial conditions. The phasing of the departure location along the NRHO was allowed to vary between $\delta\tau \in [-0.7, 0.5]$, in non-dimensional CR3BP units. The same range of target GSO parameters (inclination and RAAN) as in the previous section was used. The STT/DDP method ($m = 4$) was used to generate 50 new transfers with these characteristics. It was successfully able to optimize new trajectories for all 50 scenarios. These are shown in Fig. 10

TRANSFER USING A DIFFERENT COST FUNCTION

The STT/DDP method can also be used to optimize transfers using a different cost function than was used to generate the reference trajectory. For example, for all previous transfers computed in this work, the minimum-energy cost (Eq. 15) was employed. This form of the cost will result in

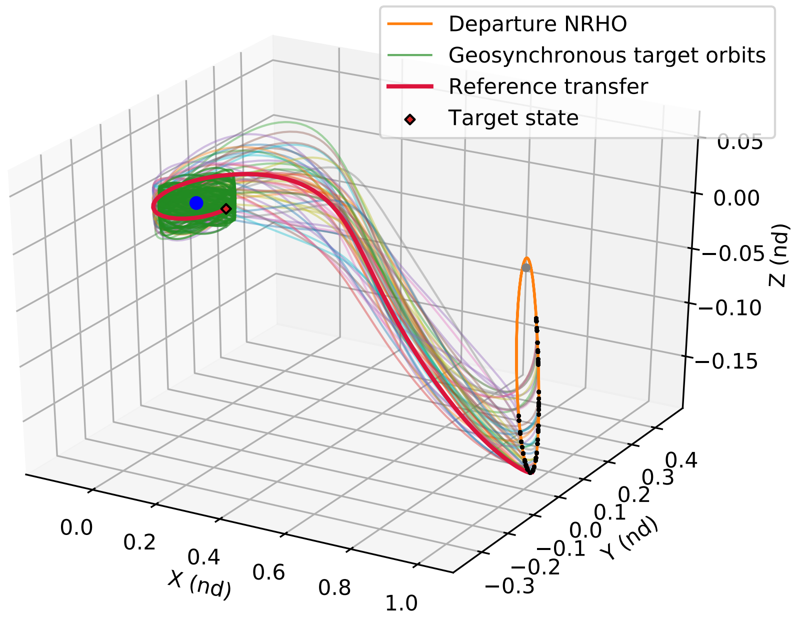

Figure 10: Low-thrust transfers from varying initial points along NRHO to various target GSOs, generated using STT/DDP $(m = 4)$

thrust profiles with the thrust spread relatively evenly over each stage. However, this form of the cost will not result in fuel-optimal trajectories, and is therefore not typically used in the design of optimal trajectories. Instead, the minimum-fuel form of the control (Eq. 16) is most often used. This form of the cost function results in bang-bang thrust profiles, with the control switching on and off over the course of the transfer.

The minimum-energy form of the cost results in a much more numerically stable optimization problem. Thus, an additional potential use for the STT/DDP method is to design a reference trajectory using an “easier” cost function (which can be solved with a small number of iterations). Then, the higher-order STTs of this reference can be integrated, and the STT/DDP method can be used to rapidly compute an optimal trajectory using a different cost function (e.g. minimum-fuel), which may require a far larger number of iterations to converge on a good solution. This could also be useful for operational situations where priorities shift during the course of a mission. In order to illustrate this strategy, the STT/DDP algorithm was run using the same reference STTs for the DRO-to-NRHO transfer as in the previous section, but using the minimum-fuel cost function from Eq. 16, with $\epsilon_{ml} = 1 \times 10^{-5}$. The target state was kept the same as the reference transfer, and the initial state was allowed to vary along the departure DRO. The thrust magnitude was constrained to be less than 0.015 nondimensional units. The tolerance and weighting parameters were set to $\epsilon_{opt} = 1 \times 10^{-6}$, $\epsilon_{feas} = 1 \times 10^{-6}$, $\sigma = 1 \times 10^{5}$, and $W = 1 \times 10^{3}$. The STT/DDP method was successfully able to compute optimal minimum-fuel trajectories for several different departure points along the reference DRO, using only the reference STTs from the minimum-energy transfer. The resulting orbit and thrust profiles are shown in Figs. 11 and 12; the bang-bang thrust profiles of the minimum-fuel trajectories are evident.

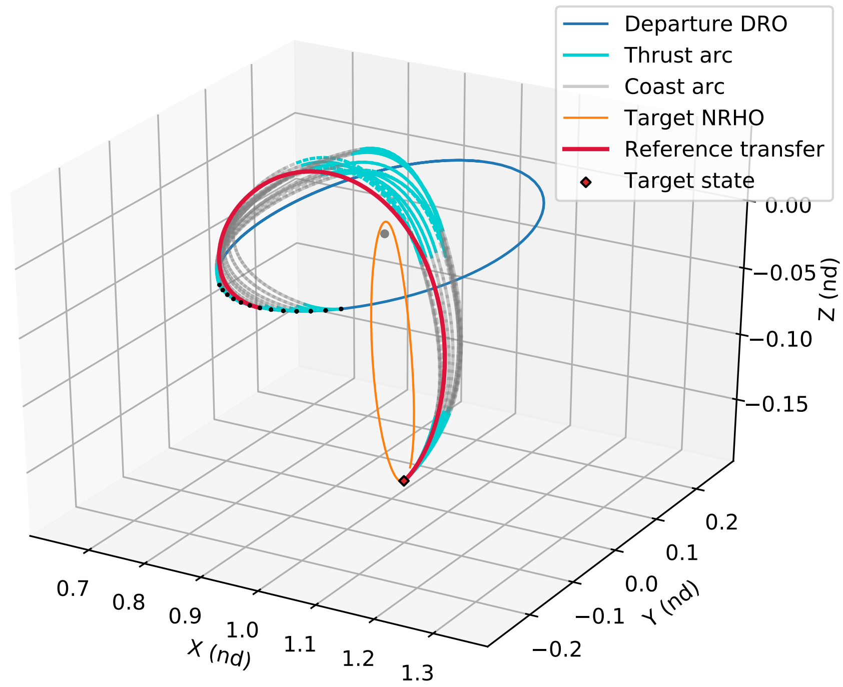

Figure 11: DRO-to-NRHO transfers using minimum-fuel cost function, computed using STT/DDP algorithm with $m = 4$. Turquoise arcs indicate thrust arcs.

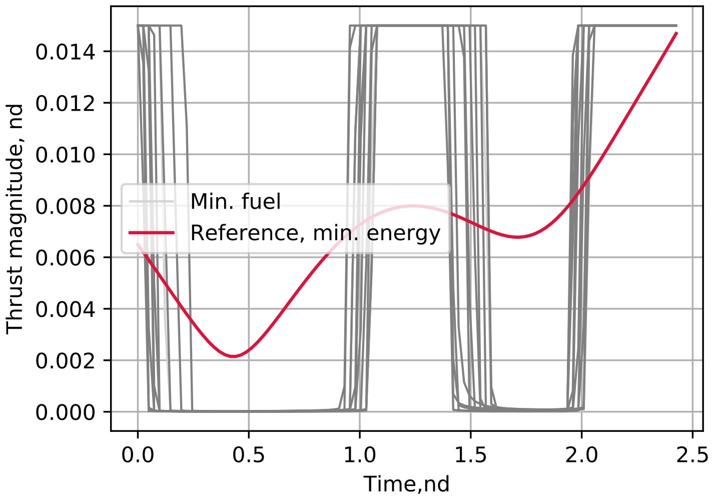

Figure 12: Thrust magnitude over time for transfers using minimum-fuel cost function, computed using STT/DDP algorithm with $m = 4$

DISCUSSION AND FUTURE WORK

The STT/DDP algorithm is promising for rapid local trajectory optimization around a reference. In an operational setting, one could integrate the higher-order STTs of a reference trajectory prior to mission execution, and use the STT/DDP algorithm to rapidly recompute near-optimal controls in response to changes in the initial or target states. As no further numerical integration is required, complex dynamical models can be incorporated at no additional cost (after the reference STT integration). The method could therefore be suitable for use on a flight computer with limited resources.

Nevertheless, there are several key rooms for improvement to the current implementation of the STT/DDP algorithm. Transitioning the method to a higher-fidelity dynamical model of the Earth-Moon system would further illustrate its computational benefits, since the time to evaluate the STTs does not increase as the dynamics become more complex. Allowing for a variable time-of-flight, following the procedure from Ref. [21], could further increase the flexibility of the proposed algorithm. Using a method to approximate the higher-order STTs with a fewer number of terms, such as the directional state transition tensor method proposed in Ref. [22], could improve the computational requirements of the STT/DDP algorithm, and would be particularly beneficial for unstable periodic orbits in cislunar space.

CONCLUSIONS

This paper presents an algorithm for rapid local trajectory optimization around a reference, and applies this method to efficiently optimize transfer between various high-interest orbits in cislunar space. Results show that the STT/DDP algorithm yields similar results to a numerical DDP algorithm when computing new trajectories in the vicinity of the reference, but at a fraction of the computational cost. The algorithm is particularly suitable to cislunar applications due to the highly nonlinear nature of orbits in the Earth-Moon regime, and the relatively short timescales which can present limitations for the traditional full-fidelity ground-based maneuver planning workflow. The proposed algorithm could also be beneficial for use on-board CubeSats with limited computational capabilities and lean flight dynamics operations teams. It could additionally be used to rapidly conduct large-scale tradeoff analyses for low-thrust enabled missions with variable departure, arrival and constraint conditions.

APPENDIX A: STT DIFFERENTIAL EQUATIONS

The differential equations for integrating the STTs up to fourth-order are listed in Park and Scheeres[15]. The first and second order equations are shown below.

$$\phi^{i,a}_{(t_f,t_0)} = \phi^{i,\alpha}_{fk}\phi^{\alpha,a}_{k0} \tag{21}$$

$$\phi^{i,ab}_{(t_f,t_0)} = \phi^{i,\alpha}_{fk}\phi^{\alpha,ab}_{k0} + \phi^{i,\alpha\beta}_{fk}\phi^{\alpha,a}_{k0}\phi^{\beta,b}_{k0} \tag{22}$$

$$\phi^{i,abc}_{(t_f,t_0)} = \phi^{i,\alpha}_{fk}\phi^{\alpha,abc}_{k0} + \phi^{i,\alpha\beta}_{fk}\left(\phi^{\alpha,a}_{k0}\phi^{\beta,bc}_{k0} + \phi^{\alpha,ab}_{k0}\phi^{\beta,c}_{k0} + \phi^{\alpha,ac}_{k0}\phi^{\beta,b}_{k0}\right) + \phi^{i,\alpha\beta\gamma}_{k0}\phi^{\alpha,a}_{k0}\phi^{\beta,b}_{k0}\phi^{\gamma,c}_{k0} \tag{23}$$

$$\begin{aligned}\phi^{i,abcd}_{(t_f,t_0)} = \;& \phi^{i,\alpha}_{fk}\phi^{\alpha,abcd}_{k0} + \phi^{i,\alpha\beta}_{fk}(\phi^{\alpha,abc}_{k0}\phi^{\beta,d}_{k0} + \phi^{\alpha,abd}_{k0}\phi^{\beta,c}_{k0} + \phi^{\alpha,ac}_{k0}\phi^{\beta,b}_{k0} + \phi^{\alpha,ab}_{k0}\phi^{\beta,cd}_{k0} + \phi^{\alpha,ac}_{k0}\phi^{\beta,bd}_{k0} \\ & + \phi^{\alpha,ad}_{k0}\phi^{\beta,bc}_{k0} + \phi^{\alpha,a}_{k0}\phi^{\beta,bcd}_{k0}) + \phi^{i,\alpha\beta\gamma}_{fk}(\phi^{\alpha,ab}_{k0}\phi^{\beta,c}_{k0}\phi^{\gamma,d}_{k0} + \phi^{\alpha,ac}_{k0}\phi^{\beta,b}_{k0}\phi^{\gamma,d}_{k0} + \phi^{\alpha,ad}_{k0}\phi^{\beta,b}_{k0}\phi^{\gamma,c}_{k0} \\ & + \phi^{\alpha,a}_{k0}\phi^{\beta,bc}_{k0}\phi^{\gamma,d}_{k0} + \phi^{\alpha,a}_{k0}\phi^{\beta,bd}_{k0}\phi^{\gamma,c}_{k0} + \phi^{\alpha,a}_{k0}\phi^{\beta,b}_{k0}\phi^{\gamma,cd}_{k0}) + \phi^{i,\alpha\beta\gamma\delta}_{fk}\phi^{\alpha,a}_{k0}\phi^{\beta,b}_{k0}\phi^{\gamma,c}_{k0}\phi^{\delta,d}_{k0} \quad (24)\end{aligned}$$

The integration of the STTs necessitates knowledge of the A tensors, which represent the partial derivatives of the state rates with respect to the state. These tensors can be analytically derived, though beyond the second order, this will generally become prohibitively tedious. For simple dynamics, these can be derived using symbolic manipulators such as Mathematica or SymPy. However, for complex dynamical systems with many perturbations, the resulting equations are often not optimally formulated and must be repeatedly re-derived each time the dynamics equations are modified. Thus, it is beneficial to employ some form of automatic differentiation. For this work, we made use of the freely-available PyAudi package developed by the European Space Agency[23], which uses the Taylor polynomial automatic differentiation method.

APPENDIX B: ENFORCING ACCURACY OF THE STTS

The STT/DDP algorithm will yield a locally optimal trajectory within the STT-approximated dynamics. If this trajectory lies within the convergence region of the reference's STTs, it will correspond to a nearly-optimal trajectory in the true dynamics. However, if the optimal trajectory obtained from the STT/DDP algorithm is too far from the reference STTs, the STT approximations may not be sufficiently accurate. In this case, the algorithm is no longer useful as it is not an accurate representation of the actual dynamics. It is therefore important to implement a method to enforce successive iterations to remain within the convergence region of the reference STTs.

Because the STTs represent a Taylor expansion up to order m integrated through time, it is difficult to explicitly predict the error from ignoring terms of $\mathcal{O}(\varepsilon^{m+1})$ (where $\varepsilon \ll 1$ and $\delta\mathbf{x}_0 \sim \mathcal{O}(\varepsilon)$). Nevertheless, we can use elements of perturbation theory[24] and knowledge of how a convergent series should behave to derive a penalty method to force these errors to be small. If we consider that an STT of order m contains secular terms that grow over time like $t, t^2, ..., t^m$[25], we can see that the expansion will break down when $t \sim \mathcal{O}(1/\varepsilon)$. In this case, the asymptotic ordering of the terms in the series breaks down and the series is no longer convergent.

However, if the order-m term (i.e. $\frac{1}{m!}\phi^{i,\gamma_1\cdots\gamma_m}\delta\hat{x}^{\gamma_1}_k...\delta\hat{x}^{\gamma_m}_k$ - the highest-order term in the expansion) is sufficiently small, and the integration time is not too long ($t \sim \mathcal{O}(1)$), then we can assume that the series is convergent, and that the approximation error from ignoring all terms of order $m+1$ (and greater) is of $\mathcal{O}(\varepsilon^{m+1}) \ll \mathcal{O}(\varepsilon^m)$. Thus, if the order-$m$ term is small, we can assume that the truncation error is smaller than this term, and that the series is sufficiently accurate. In order to enforce a small order-m term, we can apply a quadratic penalty at each stage on its magnitude. This is scaled with a weight W to ensure that the order m term is of the desired order of magnitude. The penalty parameter to be added to the local cost function at each stage k then becomes

$$L_k = W(\frac{1}{m!}\phi^{i,\gamma_1\cdots\gamma_m}\delta\hat{x}^{\gamma_1}_k...\delta\hat{x}^{\gamma_m}_k)(\frac{1}{m!}\phi^{i,\gamma_1\cdots\gamma_m}\delta\hat{x}^{\gamma_1}_k...\delta\hat{x}^{\gamma_m}_k) \quad (25)$$

where $\delta\hat{\mathbf{x}}_k$ represents the deviation from the *reference* trajectory. For this work, W was set to be the same for all state components, but it could be replaced by a vector of weights to place emphasis on specific components.

In order to include this penalty parameter in the DDP formulation, we must derive its first and second order partial derivatives with respect to the state vector X. Fortunately this form of penalty parameter is relatively straightforward to differentiate. First, we will define the i-th component of the order m term as β^i:

$$\beta^i = \frac{1}{m!}\phi^{i,\gamma_1 \dots \gamma_m}\delta\hat{x}_k^{\gamma_1}\dots\delta\hat{x}_k^{\gamma_m} \tag{26}$$

The first and second order derivatives of β^i with respect to the states can be expressed analytically as a function of the reference STTs:

$$\beta^{i,a} = \frac{1}{(m-1)!}\phi^{i,a\gamma_2 \dots \gamma_m}\delta\hat{x}_k^{\gamma_2}\dots\delta\hat{x}_k^{\gamma_m} \tag{27}$$

$$\beta^{i,ab} = \frac{1}{(m-2)!}\phi^{i,ab\gamma_3 \dots \gamma_m}\delta\hat{x}_k^{\gamma_3}\dots\delta\hat{x}_k^{\gamma_m} \tag{28}$$

Note that the number of superscripts after the i indicates the order of derivative of β^i with respect to the state vector. We can then write the stage quadratic penalty function from Eqn. 25 as

$$L_k = W\beta^i\beta^i \tag{29}$$

and the stage derivatives of this local cost function can be written compactly as

$$L^a_{X,k} = 2W\beta^{i,a}\beta^i \tag{30}$$

$$L^{ab}_{XX,k} = 2W\left[\beta^{i,a}\beta^{i,b} + \beta^i\beta^{i,ab}\right] \tag{31}$$

The weighting parameter W must be carefully chosen to be large enough to ensure that the terms of order m are maintained sufficiently small, but not too large, in which case the terms may be over-penalized to the point that they barely contribute to the approximation. For this work, this value was found through trial and error, but a more sophisticated heuristic should be derived in the future.

REFERENCES

[1] Williams, J., Lee, D. E., Whitley, R. J., Bokelmann, K. A., Davis, D. C., and Berry, C., “Targeting Cislunar Near Rectilinear Halo Orbits for Human Space Exploration,” *27th AAS/AIAA Space Flight Mechanics Meeting*, 2017.

[2] McIntosh, D., Baker, J., and Matus, J., “The NASA Cubesat Missions Flying on Artemis-1,” *34th Annual Small Satellite Conference*, 2020.

[3] Parcher, D., Abrahamson, M., Ardito, A., Han, D., Haw, R., Kennedy, B., Mastrodemos, N., Nandi, S., Park, R., Rush, B., Smith, B., Smith, J., Vaughan, A., and Whiffen, G., “Dawn Maneuver Design Performance at Vesta,” *23rd AAS/AIAA Space Flight Mechanics Meeting*, 2013.

[4] Riedel, J., Bhaskaran, S., Eldred, D., Gaskell, R., Grasso, C., Kennedy, B., Kubitscheck, D., Mastrodemos, N., Synnott, S., Vaughan, A., and Werner, R., “AutoNav Mark3: Engineering the Next Generation of Autonomous Onboard Navigation and Guidance,” 08 2006.

[5] Davis, D. C., Zimovan-Spreen, E. M., Power, R. J., and Howell, K. C., “Cubesat Deployment from a Near Rectilinear Halo Orbit,” *AIAA SCITECH 2022 Forum*, 2022.

[6] Diane Davis, Kenza Boudad, S. P. K. H., “Disposal, Deployment, and Debris in Near Rectilinear Halo Orbits,” *29th AAS/AIAA Space Flight Mechanics Meeting*, 2019.

[7] David Folta, Natasha Bosanac, A. C. K. H., "The Lunar IceCube Mission Challenge: Attaining Science Orbit Parameters from a Constrained Approach Trajectory," *27th AAS/AIAA SpaceFlight Mechanics Meeting*, 2017.

[8] Donald Dichmann, Cassandra Alberding, W. Y., "Stationkeeping Monte Carlo Simulation for the James Webb Space Telescope," *26th International Symposium on Space Flight Dynamics*, 2014.

[9] C. Chadwick, L. M., "An Overview of the ADAM Maneuver Analysis System," *1983 AAS/AIAA Astrodynamics Specialist Conference*, 1983.

[10] Lantoine, G. and Russell, R. P., "A Hybrid Differential Dynamic Programming Algorithm for Constrained Optimal Control Problems. Part 1: Theory," *J. Optim. Theory Appl.*, Vol. 154, No. 2, Aug. 2012, pp. 382417.

[11] Whiffen, G., "Mystic: Implementation of the Static Dynamic Optimal Control Algorithm for High-Fidelity, Low-Thrust Trajectory Design," *AIAA/AAS Astrodynamics Specialist Conference*, 2006.

[12] Di Lizia, P., Armellin, R., Bernelli-Zazzera, F., and Berz, M., "High order optimal control of space trajectories with uncertain boundary conditions," *Acta Astronautica*, Vol. 93, 2014, pp. 217 – 229.

[13] Boone, S. and McMahon, J., "Orbital Guidance Using Higher-Order State Transition Tensors," *Journal of Guidance, Control, and Dynamics*, Vol. 44, No. 3, 2021, pp. 493–504.

[14] Boone, S. and McMahon, J., "Rapid Local Trajectory Optimizaion Using Higher-Order State Transition Tensors and Differential Dynamic Programming," *2020 AAS/AIAA Astrodynamics Specialist Conference*, 2020.

[15] Park, R. and Scheeres, D., "Nonlinear Mapping of Gaussian Statistics: Theory and Applications to Spacecraft Trajectory Design," *Journal of Guidance, Control, and Dynamics*, Vol. 29, 2006.

[16] Jacobson, D. H. and Mayne, D. Q., *Differential dynamic programming*, American Elsevier Pub. Co New York, 1970.

[17] Aziz, J., Scheeres, D., and Lantoine, G., "Hybrid Differential Dynamic Programming in the Circular Restricted Three-Body Problem," *Journal of Guidance, Control, and Dynamics*, Vol. 42, 2019, pp. 1–13.

[18] Conn, A. R., Gould, N. I. M., and Toint, P. L., *Trust Region Methods*, Society for Industrial and Applied Mathematics, 2000.

[19] Bezrouk, C. and Parker, J., "Long term evolution of distant retrograde orbits in the Earth-Moon system," *Astrophysics and Space Science*, Vol. 362, No. 176, 2017.

[20] Parrish, N. L., *Low Thrust Trajectory Optimization in Cislunar and Translunar Space*, Ph.D. thesis, University of Colorado Boulder, 2018.

[21] Boone, S. and McMahon, J., "Variable Time-of-Flight Spacecraft Maneuver Targeting Using State Transition Tensors," *Journal of Guidance, Control, and Dynamics*, Vol. 44, No. 11, 2021, pp. 2072–2080.

[22] Boone, S. and McMahon, J., "Directional State Transition Tensors for Capturing Dominant Nonlinear Dynamical Effects," *2021 AAS/AIAA Astrodynamics Specialist Conference*, 2021.

[23] Izzo, D. and Biscani, F., "Audi/PyAudi," 2018.

[24] Bender, C. M. and Orszag, S. A., *Advanced mathematical methods for scientists and engineers: I: Asymptotic methods and perturbation theory*, Springer, 1999.

[25] Roa, J. and Park, R. S., "Efficient method for approximating nonlinear dynamics: applications to uncertainty propagation and estimation," *AIAA Scitech 2020 Forum*.