(Preprint) AAS 16-526

HYBRID DIFFERENTIAL DYNAMIC PROGRAMMING WITH STOCHASTIC SEARCH

Jonathan D. Aziz[*], Jeffrey S. Parker[†] and Jacob A. Englander[‡]

Differential dynamic programming (DDP) has been demonstrated as a viable approach to low-thrust trajectory optimization, namely with the recent success of NASA's Dawn mission. The Dawn trajectory was designed with the DDP-based Static/Dynamic Optimal Control algorithm used in the Mystic software.[1] Another recently developed method, Hybrid Differential Dynamic Programming (HDDP),[2,3] is a variant of the standard DDP formulation that leverages both first-order and second-order state transition matrices in addition to nonlinear programming (NLP) techniques. Areas of improvement over standard DDP include constraint handling, convergence properties, continuous dynamics, and multi-phase capability. DDP is a gradient based method and will converge to a solution nearby an initial guess. In this study, monotonic basin hopping (MBH) is employed as a stochastic search method to overcome this limitation, by augmenting the HDDP algorithm for a wider search of the solution space.

INTRODUCTION

DDP is a trajectory optimization algorithm that relies on quadratic expansions of the cost function nearby a nominal trajectory.[4] Iterates therefore only exhibit local improvements to the nominal trajectory within a region determined by the validity of the quadratic approximation. After converging on a solution it is difficult to claim anything more than local optimality, as is the case for any nonlinear optimization algorithm. Furthermore, for trajectory optimization in the context of spacecraft mission design it is often desirable to optimize decision variables that span a large trade space. Pairing trajectory optimization with a stochastic search method allows for the selection of the best design from many locally optimal solutions, while leveraging the full space of design variables.

The addition of stochastic search to trajectory optimization algorithms is not new to spacecraft mission design, but past applications have been augmentations to NLP solvers.[5,6] This work instead implements HDDP as an inner loop to compute the spacecraft trajectory. MBH is chosen as the stochastic search method in an outer loop.

The HDDP and MBH algorithms are detailed in the following sections, and implementation of HDDP is verified with a fixed time of flight Earth-Mars rendezvous transfer. The procedure for including variable departure and arrival times is discussed in the context of multi-phase HDDP and applied to the Earth-Mars rendezvous example. MBH is then applied to encourage larger steps in the time variables, including shifts in synodic period.

[*]Ph.D. Student, Colorado Center for Astrodynamics Research, University of Colorado, Boulder, CO 80309.
[†]Assistant Professor, Colorado Center for Astrodynamics Research, University of Colorado, Boulder, CO 80309.
[‡]Aerospace Engineer, NASA/GSFC, Code 595, 8800 Greenbelt Rd, Greenbelt, MD 20771, USA.

HYBRID DIFFERENTIAL DYNAMIC PROGRAMMING

The fundamental DDP iteration consists of a *backward sweep* and a *forward pass*. During the backward sweep, a quadratic model of the objective function about a nominal trajectory is minimized with respect to deviations in the control variables. A new trajectory is computed following the updated control law in the forward pass. If the cost improves then the new iterate is accepted as the nominal trajectory, otherwise the step size of the control update is reduced.

Backward Sweep

According to Bellman's principle of optimality, any sub-arc of an optimal trajectory is also optimal. This gives rise to the backward sweep in DDP, where optimal sub-arcs are computed recursively, proceeding backwards from the final state. For a trajectory discretized into N stages, a sequence of subproblems is solved from stage $k = N-1, ..., 0$. The first optimization finds the control update δu_{N-1} that minimizes J_{N-1}, the cost-to-go at stage $k = N-1$, yielding the optimal cost-to-go J_k^*. Solving the subproblem at stage k then minimizes the local stage cost L_k, given that the optimization downstream has already been performed.

$$J_k = L_k + J_{k+1}^* \tag{1}$$

$$J_k^* = \min_{\delta u_k}[J_k] = \min_{\delta u_k}[L_k + J_{k+1}^*] = \min_{\delta u_k}[L_k] + J_{k+1}^* \tag{2}$$

Here the appeal of DDP over popular NLP methods is made evident. For m-dimensional control, the DDP iteration solves N subproblems of size m, as opposed to a single optimization problem of size mN. This produces a stark contrast when solving a system of linear equations, as is typically the case. Inverting the $mN \times mN$ Hessian matrix, and populating it with the necessary derivatives across all stages, becomes increasingly difficult with larger problem sizes. The DDP subproblem size remains fixed, albeit there are more of them to solve.

A feasible subproblem is formed in DDP by quadratic expansions of the cost-to-go at each stage, accomplished by a second-order Taylor series expansion of both sides of Eq. (1).

$$\delta J_k \approx ER_k + J_{x,k}^T \delta x_k + J_{u,k}^T \delta u_k + \frac{1}{2}\delta x_k^T J_{xx,k} \delta x_k + \frac{1}{2}\delta u_k^T J_{uu,k} \delta u + \delta x_k^T J_{xu,k} \delta u_k \tag{3a}$$

$$\delta L_k \approx L_{x,k}^T \delta x_k + L_{u,k}^T \delta u_k + \frac{1}{2}\delta x_k^T L_{xx,k} \delta x_k + \frac{1}{2}\delta u_k^T L_{uu,k} \delta u + \delta x_k^T L_{xu,k} \delta u_k \tag{3b}$$

$$\delta J_{k+1}^* \approx ER_{k+1} + J_{x,k+1}^{*T} \delta x_{k+1} + \frac{1}{2}\delta x_{k+1}^T J_{xx,k+1}^* \delta x_{k+1} \tag{3c}$$

ER_{k+1} represents the expected reduction as a result of downstream optimization. Subscripts imply differentiation with respect to state x or control u at stage k. Because optimization at stage $k+1$ has already been performed, no δu terms are present in Eq. (3c) and ER_{k+1}, $J_{x,k+1}^*$ and $J_{xx,k+1}^*$ are known.

In order to match terms in Eq. (3), downstream deviations need to be known as a function of deviations in state and control upstream, $\delta x_{k+1} = f(\delta x_k, \delta u_k)$ In the standard DDP formulation, sensitivities and deviations at stage $k+1$ are integrated backwards in time to stage k. HDDP distinguishes itself by using the first-order state transition matrix Φ (STM) and second-order state transition tensor Φ^2 (STT) to perform this mapping.

$$\delta x_{k+1} = \Phi_k \delta x_k + \frac{1}{2}\delta x_k^T \cdot \Phi_k^2 \delta x_k \tag{4}$$

It is useful to proceed with the augmented state $X_k = [x_k \; u_k]^T$ and to partition the STM and STT into submatrices.

$$\Phi = \begin{bmatrix} \Phi_x & \Phi_u \\ 0 & 0 \end{bmatrix}, \quad \Phi^2 = \begin{array}{c} \Phi^2_{ux} \quad \Phi^2_{uu} \\ \Phi^2_{xx} \quad \Phi^2_{xu} \\ 0 \end{array} \tag{5}$$

Now derivatives of the cost-to-go at the current stage k are available by using Eq. (4) in Eq. (3) and matching terms.

$$J_{x,k} = L_{x,k} + \Phi^T_{x,k} J^*_{x,k+1} \tag{6a}$$

$$J_{u,k} = L_{u,k} + \Phi^T_{u,k} J^*_{x,k+1} \tag{6b}$$

$$J_{xx,k} = L_{xx,k} + \Phi^T_{x,k} J^*_{xx,k+1} \Phi_{x,k} + J^{*T}_{x,k+1} \cdot \Phi^2_{xx} \tag{6c}$$

$$J_{uu,k} = L_{uu,k} + \Phi^T_{u,k} J^*_{xx,k+1} \Phi_{u,k} + J^{*T}_{x,k+1} \cdot \Phi^2_{uu} \tag{6d}$$

$$J_{ux,k} = L_{ux,k} + \Phi^T_{u,k} J^*_{xx,k+1} \Phi_{x,k} + J^{*T}_{x,k+1} \cdot \Phi^2_{ux} \tag{6e}$$

Stage derivatives are used in the minimizing control law, that is straightforward to obtain by setting the derivative of Eq. (3a) with respect to δu_k equal to zero.

$$\delta u_k = -J^{-1}_{uu,k}(J_{u,k} + J_{ux,k}\delta x_k) \tag{7}$$

This unconstrained feedback control law is otherwise written as

$$\begin{aligned} \delta u_k &= A_k + B_k \delta x_k \\ A_k &= -J^{-1}_{uu,k} J_{u,k} \\ B_k &= -J^{-1}_{uu,k} J_{ux,k} \end{aligned} \tag{8}$$

A_k and B_k are stored in memory for application in the forward pass. Before proceeding to stage $k-1$, the expected reduction and derivatives of the cost-to-go are updated to reflect the new control law.

$$ER_k = ER_{k+1} + J^T_{u,k} A_k + \frac{1}{2} A^T_k J_{uu,k} A_k \tag{9a}$$

$$J^*_{x,k} = J_{x,k} + J^T_{u,k} B_k + A^T_k J_{uu,k} B_k + A^T_k J_{ux,k} \tag{9b}$$

$$J^*_{xx,k} = J_{xx,k} + B^T_k J_{uu,k} B_k + B^T_k J_{ux,k} + J^T_{ux,k} B_k \tag{9c}$$

Everything is in place for an unconstrained backward sweep. STMs and STTs are integrated and stored alongside the nominal trajectory, or preferably computed analytically if possible, as this is the most expensive part of the algorithm. The expected reduction and derivatives are initialized at the final state, $ER_N = 0$, $J^*_{x,N} = J_{x,N}$, $J^*_{xx,N} = J_{xx,N}$. Repeated application of Eqs. (6), (8) and (9) from stage $N-1, ..., 0$ yields a trial control law for the forward pass and an estimate of the expected reduction in the objective function.

Trust-Region Quadratic Subproblem

The unconstrained control update in Eq. (8) is likely to step beyond the valid region of the quadratic approximation. For nonlinear problems like spacecraft trajectory optimization, an unconstrained backward sweep and subsequent application of the new control law is likely to lead to divergence or infeasible iterates. Eq. (8) also requires $J_{uu,k}$ to be invertible, and positive definite so that δu_k is along a descent direction. HDDP overcomes these challenges by solving a trust-region quadratic subproblem (TRQP) at each stage. Now when minimizing Eq. (3a), δu_k is required to lie within the trust-region radius Δ.

$$\begin{aligned} &\min_{\delta u_k}[J_{u,k}\delta u_k + \frac{1}{2}\delta u_k^T J_{uu,k}\delta u_k] \\ &\text{s.t. } \|D\delta u_k\| \leq \Delta \end{aligned} \tag{10}$$

The methods of Reference 7 have proven robust in solving this subproblem in HDDP, yielding a positive definite $\tilde{J}_{uu,k}$ and constrained δu_k. However, the HDDP algorithm is sensitive to the selection of a scaling matrix D that determines the shape of the trust-region. For simple problems, setting D to the identity matrix is sufficient. When components of J_u and J_{uu} vary by orders of magnitude, a robust heuristic for selecting D becomes necessary. Reference 3 suggests several scaling methods and provides a performance comparison.

Performance of HDDP is also sensitive to the procedure for updating the trust-region radius with each iteration. If an iteration is unsuccessful, then the quadratic model is poor and the trust-region radius should be reduced. With an otherwise successful iteration, the trust-region radius is appropriate and could possibly be increased. Evaluating the ratio between the actual cost improvement and the expected reduction $\rho = \delta J/ER$ qualifies the validity of the model. Clearly, $\rho < 0$ is an unsuccessful iteration, and Δ should be reduced. For $\rho \in (0, 1)$ the model overpredicts the improvement, but underpredicts for $\rho > 1$. The model is accurate if ρ is close to one, i.e. $\rho \in [1-\epsilon, 1+\epsilon]$ for small ϵ. Ideally, Δ should be selected to keep ρ close to one, but such update rules have been found to prohibit convergence, where the trust-region collapses to Δ_{min} before terminal constraints are satisfied. In this study, all $\rho > 0$ are accepted as successful iterates, and the update for iteration $p+1$ is selected to favor frequent increases in Δ.

$$\Delta_{p+1} = \begin{cases} \min((1+\kappa)\Delta_p, \Delta_{\text{max}}), & \rho > C,\ C \in (0,1) \\ \max((1-\kappa)\Delta_p, \Delta_{\text{min}}), & \text{otherwise} \end{cases} \tag{11}$$

Selection of an initial trust-region radius Δ_0 and the limits $[\Delta_{\text{min}},\ \Delta_{\text{max}}]$ is straightforward as they typically represent physical quantities like thrust. Tuning the parameters κ and C can prove tedious.

The trust-region step $\delta u_k \leq \Delta$ still might violate stage constraints. Here the only stage constraint considered is maximum thrust T_{max} with control variables u as components of the thrust vector. During the backward pass it is easy to check for active thrust magnitude constraints, $u_k + A_k \leq T_{\text{max}}$. If the constraint is active then A_k is reduced and the feedback is zeroed. Challenging stage constraints warrant a more sophisticated approach. Success has been found with Fletcher's quadratic programming method as suggested for HDDP.[2,8]

Augmented Lagrangian Method

Terminal constraints are enforced in HDDP with the augmented Lagrangian method (ALM). Constraints on the final state of the form $\psi(x_N) = 0$ are introduced to the original objective function h

with Lagrange multipliers λ and penalty parameter σ.

$$J = h + \lambda^T \psi + \sigma f^2 \tag{12}$$

The quadratic term is $f^2 = \psi^T \psi$, the square of the norm of the constraint violations. An initial penalty parameter σ_0 is selected sufficiently large to compete with the contributions from h. Then σ must be continually increased to push the iterates towards feasibility. The HDDP update rule from Equation (13) is designed to prevent σ from becoming so large that it causes numerical difficulties.

$$\sigma_{p+1} = \max(\min(0.5\frac{|h|}{f^2}, k_\sigma \sigma_p), \sigma_p) \tag{13}$$

The rule is applied when an iterate fails to reduce the norm of the constraint violations, $f_{p+1} > f_p$. Selecting σ_0 and the factor $k_\sigma > 1$ has been the most tedious part of tuning HDDP. Values that are too small will lead to slow convergence or trust-region collapse, while large values effectively neglect h or cause numerical difficulties. Furthermore, these tuning parameters are largely problem dependent, so it is difficult to form a reliable heuristic. This challenge motivates the search for a penalty-free DDP method that will be the subject of future investigations.

The Lagrange multipliers serve to smooth the objective function and permit convergence proofs that do not require $\sigma \to \infty$.[9] HDDP is robust to initial guesses for λ, and $\lambda = 0$ has been sufficient thus far. The updates $\delta\lambda$ maximize the cost-to-go, and are computed by TRQP$(-J_\lambda, -J_{\lambda\lambda}, \Delta)$ after all stages have been minimized in the backward sweep. Here the trust-region radius is less intuitive, and sometimes it has helped to scale the multiplier trust-region radius to some factor of the control variable trust-region radius. The backward sweep equations must be reformed to account for sensitivities with respect to the multipliers, but the procedure is unchanged. A feedback term $C_k \delta\lambda$ is added to the control, $C_k = -J_{uu,k}^{-1} J_{u\lambda,k}$, and new stage equations arise for first, second, and cross-derivatives in λ. See Reference 2 for the complete treatment.

HDDP Acknowledgement

The body of this work results from efforts to implement HDDP as introduced by Lantoine and Russell.[2,3,10,11,12] The two part journal series[2,3] is a complete description of the algorithm with comprehensive theoretical discussion and example applications. Reference 10 is earlier work that first solves for controls in an inner loop, and then multipliers in an outer loop. This methodology has proven useful when struggling to update controls, multipliers and the penalty parameter simultaneously. Reference 11 dramatically reduces the computational effort required of HDDP by implementing analytic STMs and STTs. The preceding discussion should be consistent with their nomenclature, aside from neglecting to address multi-phase trajectories. Brief attention to multi-phase trajectories in HDDP will be given in a later next section. The implementation here differs by integrating a continuous thrust control, aggressive increases to the trust-region radius, and a naive approach to thrust magnitude constraints.

MONOTONIC BASIN HOPPING

The stochastic search method implemented here is based off of the MBH algorithm. MBH is a global optimization heuristic that seeks to explore the entire solution space and exploit local optima to find improvements in nearby solutions. MBH has been successfully coupled with NLP solvers by the European Space Agency's Advanced Concepts Team[5] and in NASA Goddard's Evolutionary Mission Trajectory Generator (EMTG).[6] Pseudocode in Algorithm (1) details the procedure.

First, an initial guess is provided to HDDP to generate a nominal solution. In this study all initial guesses are ballistic, i.e. all thrust vectors are zero magnitude. If the nominal solution is a feasible trajectory it is stored in memory. MBH then proceeds until meeting a prescribed stopping criteria such as computation time or N_{hop}, an MBH iteration limit.

The MBH iteration begins by introducing random perturbations to the decision variables of the nominal solution, and then HDDP is reinitialized. Small perturbations are expected to fall within the quadratic trust-region and HDDP will return the previous nominal trajectory. Thus, a global HDDP method benefits from drawing the random perturbations from long-tailed Cauchy and Pareto distributions as suggested by Englander and Englander.[6] The results presented here correspond to Pareto distribution sampling.

Introduction of departure and arrival times as decision variables adds many local minima to the trajectory optimization problem. Furthermore, if in the initial and final orbits lie in different orbital planes, then opportunities across different synodic periods are not equivalent. MBH tries different synodic periods with some percent likelihood $\rho_{time-hop}$. If a sample from the standard uniform distribution is below that value, the time variables are shifted forward or backward one synodic period.

With HDDP filling the role of the NLP solver, special consideration must be given to the penalty parameter σ that has been increased to some large value to produce the first nominal solution. It must be reset so that MBH iterations do not take σ to a large enough value to produce numerical instability. The initial parameter σ_0 is tied to an initial guess and likely insufficient, as the perturbed solution should be close to feasibility. The penalty parameter is reset at each MBH iteration so that HDDP restarts with more weight given to constraint violations than the objective in the ALM cost function. The Lagrange multipliers are carried through each hop.

$$\sigma_{i,hop} = \max(\sigma_0, 1.1\frac{|h|}{f^2}) \tag{14}$$

To clarify, a forward pass is first computed with the perturbed decision variables. A subsequent cost function evaluation yields the current objective h and constraint violation f. Then σ is reset and the cost is again evaluated before beginning the backward sweep.

Finally, if the iterate improves the objective on a feasible solution, or reduces the constraint violations of an infeasible solution, the iterate is accepted and stored. The next MBH iteration proceeds by perturbing this new nominal solution.

EXAMPLE EARTH-MARS RENDEZVOUS

Implementation of HDDP is validated by reproducing the Earth-Mars rendezvous transfer from Reference 3. However, here the control variable is continuous low-thrust in the velocity, normal, and co-normal directions, as opposed to approximating the low-thrust by impulsive maneuvers separated by Keplerian arcs in HDDP *standard*.[3] The equations of motion reflect two-body dynamics with continuous low-thrust to be consistent with the finite burn low-thrust model (FBLT) in EMTG.[13]

$$\begin{aligned} \ddot{x} &= -\mu x/r^3 + u_x/m \\ \ddot{y} &= -\mu y/r^3 + u_y/m \\ \ddot{z} &= -\mu z/r^3 + u_z/m \\ \dot{m} &= -\left\|u\right\|/(I_{sp}g_0) \end{aligned} \tag{15}$$

Algorithm 1 Monotonic Basin Hopping (MBH)

generate random point $\mathbf{x}$
run NLP solver to find point $\mathbf{x}^*$ using initial guess $\mathbf{x}$
$\mathbf{x}_{current} = \mathbf{x}^*$
if $\mathbf{x}^*$ is a feasible point **then**
  save $\mathbf{x}^*$ to archive
end if
while not hit stop criterion **do**
  generate $\mathbf{x}'$ by randomly perturbing $\mathbf{x}_{current}$
  for each time of flight variable t_i in $\mathbf{x}'$ **do**
    if $rand\,(0,1) < \rho_{time-hop}$ **then**
      shift t_i forward or backward one synodic period
    end if
  end for
  run NLP solver to find locally optimal point $\mathbf{x}^*$ from $\mathbf{x}'$
  if $\mathbf{x}^*$ is feasible **and** $f\,(\mathbf{x}^*) < f\,(\mathbf{x}_{current})$ **then**
    $\mathbf{x}_{current} = \mathbf{x}^*$
    save $\mathbf{x}^*$ to archive
  else if $\mathbf{x}^*$ is infeasible **and** $\|c\,(\mathbf{x}^*)\| < \|c\,(\mathbf{x}_{current})\|$)
    $\mathbf{x}_{current} = \mathbf{x}^*$
  end if
end while
return best $\mathbf{x}^*$ in archive

Note that the augmented state vector consists of the three position components, three velocity components, spacecraft mass, and three thrust components, $X = [r, v, m, u]^T$. Numerical integration of Equation (15) is performed with the adaptive step eight-stage Dormand-Prince method.[14]

A 1000 kg spacecraft with a .5 N, 2000 I_{sp} thruster departs Earth on April 10th, 2007 with a fixed time of flight of 348.79 days. The trajectory is discretized into 40 stages of equal time. The augmented Lagrangian is formed to maximize the final mass with a penalty on the position and velocity errors at Mars arrival.

$$\begin{aligned} h &= -m_f \\ \psi &= \begin{bmatrix} r_f - r_M \\ v_f - v_M \end{bmatrix} \end{aligned} \tag{16}$$

The resulting transfer is shown in Figure 1 with corresponding thrust profile in Figure 2. The trajectory is also computed in the EMTG – FBLT model for validation. HDDP obtains the expected bang-bang control profile, but differs slightly from EMTG at the switching points. Table 1 shows how final mass increases with the improved accuracy of a continuous low-thrust model. Lagrange multipliers from the continuous low-thrust solution and the reference HDDP *standard* in Table 2 differ but are seemingly related.

ALM Tuning

Successfully converging on a solution requires that the penalty weight and its rate of increase are appropriately selected. An additional benefit of the stochastic step is that MBH can cover for

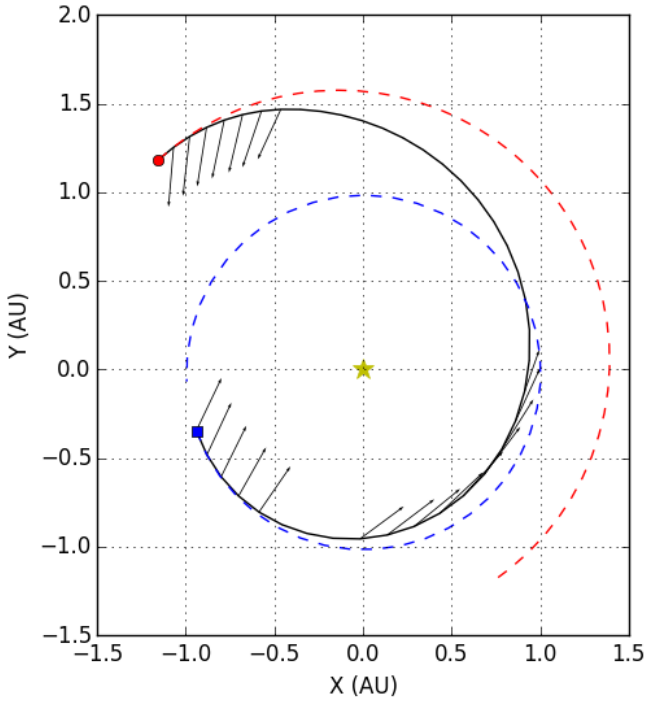

Figure 1: Ecliptic view of an example Earth-Mars rendezvous transfer with thrust vectors shown.

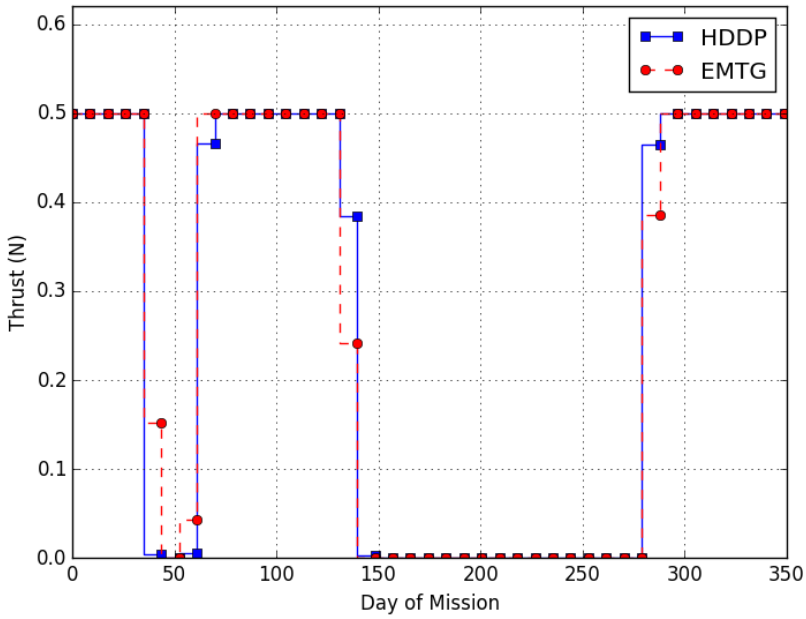

Figure 2: Thrust profiles from the FBLT implementation of HDDP and the EMTG FBLT model.

Table 1: Comparison of Spacecraft Final Mass

HDDP *standard*	$m_f = 598.66$ kg
HDDP – FBLT	$m_f = 603.29$ kg
EMTG – FBLT	$m_f = 603.45$ kg

Table 2: Comparison of Lagrange multipliers

HDDP *standard*	$\lambda = [0.5095, -1.2700, -0.2665, 0.1178, 2.0701, 0.13404]^T$
HDDP – FBLT	$\lambda = [1.0793, -2.3127, -0.5920, -0.1125, 2.9337, 0.0463]^T$

an improperly tuned HDDP setup. Table 3 presents a survey of HDDP results for different ALM parameters on the previous Earth-Mars rendezvous example, and the improvement after applying MBH. Time variables are fixed in this example so perturbations are applied to thrust variables only. MBH adds robustness to HDDP by both improving on current solutions and finding solutions when HDDP fails to converge. This added utility removes the very tedious task for the mission designer to obtain the proper parameters by trial and error. All of the failures indicated correspond to trust-region collapse before obtaining a feasible solution. When this occurs, it is often possible to drive HDDP toward feasibility with parameter resets alone, absent of perturbations.

VARIABLE TIME OF FLIGHT

Time of flight variables can be handled as static parameters in HDDP, $w = [t_0, t_f]^T$. Alternatively, the time of flight variable can be included in the augmented state vector.[11] Static parameters enable a multi-phase HDDP formulation.

$$x_{i,0} = \Gamma_i(w_i) \tag{17}$$

The initial state of phase i is determined by the initial function Γ_i. Dynamics and constraints may be functions of w, and those static parameters may be subject to additional constraints, e.g. upper and lower bounds on the time of flight.

A spacecraft trajectory may then be described by any number of phases, further discretized into stages. This multi-phase formulation is especially useful for discontinuities like impulsive maneuvers or planetary flybys. Static parameters might then be ΔV or flyby altitude. An additional use for static parameters might be spacecraft design variables like solar panel size, initial mass, or specific impulse. The backward sweep equations are again reformed, now to account for sensitivities with respect to the static parameters. The new feedback control term is $D_k \delta w$, $D_k = -J_{uu,k}^{-1} J_{uw,k}$. Furthermore, inter-phase constraints might require phase dependent ALM terms, λ_i, ψ_i. The backward sweep begins with the subproblem at the final stage of the final phase and proceeds recursively through the first stage of the final phase. Inter-phase subproblems TRQP$(-J_{\lambda,i}, -J_{\lambda\lambda,i}, \Delta)$ and TRQP$(J_{w,i}, J_{ww,i}, \Delta)$ are then solved before proceeding to the upstream stage. After solving subproblems of all of the stages of all the phases, the inter-phase subproblems at $i = 0$ complete the backward sweep.

Earth-Mars Rendezvous With Time Variables

Variable time of flight is introduced to the example Earth-Mars rendezvous problem by a two-phase representation of the trajectory. However, the second phase contains solely an initial function with zero stages, $x_{1,0} = \Gamma_1(w_1) = x_M(t_f)$. In other words, the second phase is the moment of Mars

Table 3: Survey of ALM tuning and improvements with MBH

σ_0	κ_s	Iterations	m_f (kg)	$m_f, N_{hop} = 30$
10	1.1	123	failed	603.29*
	1.5	541	failed	603.04
	2.0	643	601.87	602.80
100	1.1	231	failed	603.07
	1.5	181	602.93	603.17
	2.0	892	601.67	602.04
1000	1.1	352	failed	602.95
	1.5	1119	602.50	603.14
	2.0	940	601.75	602.21

*Solution in Figures 1- 2.

Table 4: Variable Time of Flight Results

δt_0	-23.89 days
δt_f	43.78 days
m_f	674.95 kg
λ	$[1.5042, -2.1115, -0.9008, -0.3042, 3.2379, 0.1306]^T$

arrival, and its initial function is simply a lookup of the state of Mars at the final time. Similarly, the initial function of the first phase is Earth's position and velocity at the time of departure. The inter-phase subproblems are then 1-dimensional and inexpensive to compute, for $w_0 = t_0$, and $w_1 = t_f$, and there are no additional ALM terms. Time derivatives for these subproblems are available from the equations of motion and application of the chain rule.

$$J_t = \dot{x}^T J_x \tag{18a}$$

$$J_{tt} = \dot{x}^T J_{xx} \dot{x} \tag{18b}$$

$$J_{tx} = \dot{x}^T J_{xx} \tag{18c}$$

$$J_{t\lambda} = \dot{x}^T J_{x\lambda} \tag{18d}$$

In the context of this example problem, Eqs. (18) are evaluated at t_f and δt_f is computed. The same cannot be done for the initial time, as the sensitivities must be carried through the backward sweep as stage minimizations are performed. Of course, this mapping is accomplished by the parameter STM, $\Phi_{w,k} = \dot{x}_k$.

The Earth-Mars rendezvous is again solved in HDDP, now with unbounded time variables. The resulting transfer is presented for ALM parameters $\sigma_0 = 100$ and $\kappa_\sigma = 1.5$ without applying MBH.

The new trajectory departs Earth 23.89 days earlier and arrive at 43.78 days Mars later, for a new time of flight of 416.46 days. Changing these mission dates saves 71.66 kg of propellant, for a new final mass of 674.95 kg. The thrust profile in Figure 4 suggests that the trajectory could be improved further. The mid-trajectory thrust arc has been shifted backwards in time but the switch

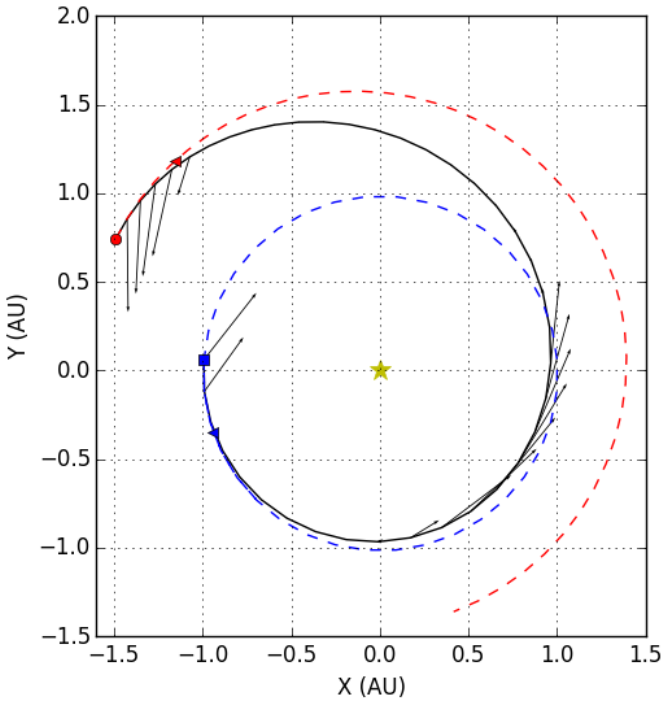

Figure 3: Resulting trajectory when time variables are included in the Earth-Mars rendezvous example. Triangle markers indicate departure and arrival corresponding to the fixed time of flight case.

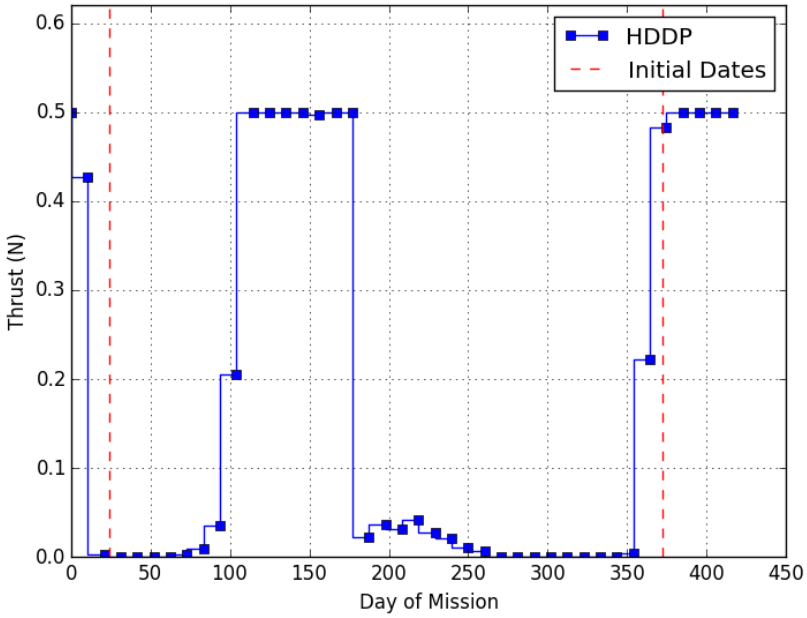

Figure 4: Thrust profile for the variable time example with reference dates from the fixed time of flight case.

Table 5: Results for $N_{hop} = 100$

δt_0	836.85 days
δt_f	894.95 days
m_f	745.57 kg
λ	$[4.3679, -1.1893, -2.1915, -4.2577, 6.0110, 1.5276]^T$

from max-thrust to coast is imperfect, with small thrust magnitudes from mission days 200-250. Restarting HDDP with this trajectory as an initial guess for fixed time of flight on the new mission dates should improve the result. This result is still nearby the fixed time of flight trajectory, so MBH is applied next to explore other local minima in the time variables, including the following synodic period.

APPLICATION OF THE STOCHASTIC STEP

Introduction of departure and arrival times as decision variables adds many local minima to the trajectory optimization problem, but HDDP iterates are constrained by the trust-region radius where the quadratic approximation is valid. MBH is now applied to the variable time Earth-Mars rendezvous example. Perturbations are applied to the initial and final time variables only, not control variables. Mars orbit lies outside of the ecliptic plane, so mission opportunities across different synodic periods are not equivalent. MBH hops are allowed to sample both the initial 2007 opportunity, and the following synodic period in 2009. A 20% likelihood of the synodic period hop is selected, $\rho_{time-hop} = 0.2$ and the best solution is accepted after 100 hops. HDDP is applied with first-order STMs only to quickly generate the results.

The new trajectory departs and arrives Earth and Mars in distinctly different phases of their respective orbits, and departs Earth during the 2009 opportunity. MBH reduces the time of flight to 392.86 days and improves the final mass to 745.57 kg. The resulting thrust profile is a Hohmann-like two burn solution. Like before, the thrust profile in Figure 6 suggest the trajectory could likely be improved further, as the first thrust arc ramps up and down instead of exhibiting a discrete on/off control switch. Furthermore, there is room for improvement in the discretization procedure, as the thrust profile only represents the final 28 of 40 stages. HDDP reduced the first 12 stages to coast arcs, effectively not departing Earth until the stage 13. Figure 7 tracks the progress of successful MBH iterations, and notes hops in synodic period. Those hops in synodic period are very successful, while the more local hops offer incremental improvement.

CONCLUSION

The HDDP and MBH algorithms have been described and paired as a two-loop procedure to spacecraft trajectory optimization, where HDDP computes trajectories in an inner loop and MBH works as a stochastic search step in an outer loop. Implementation of HDDP successfully reproduced the previous results for an example Earth-Mars rendezvous and was verified by EMTG. MBH was shown to overcome the limitations of poor tuning of the ALM parameters within HDDP.

Next, the inclusion of variable times of departure and arrival were presented as a multi-phase HDDP problem. An improved trajectory for the Earth-Mars rendezvous was computed for variable departure and arrival but nearby the initial dates. Stochastic search via MBH was then applied. MBH successfully guided HDDP through larger steps in the time variables and across synodic periods. The final solution departed in a later synodic period, with different phases of Earth and Mars. The

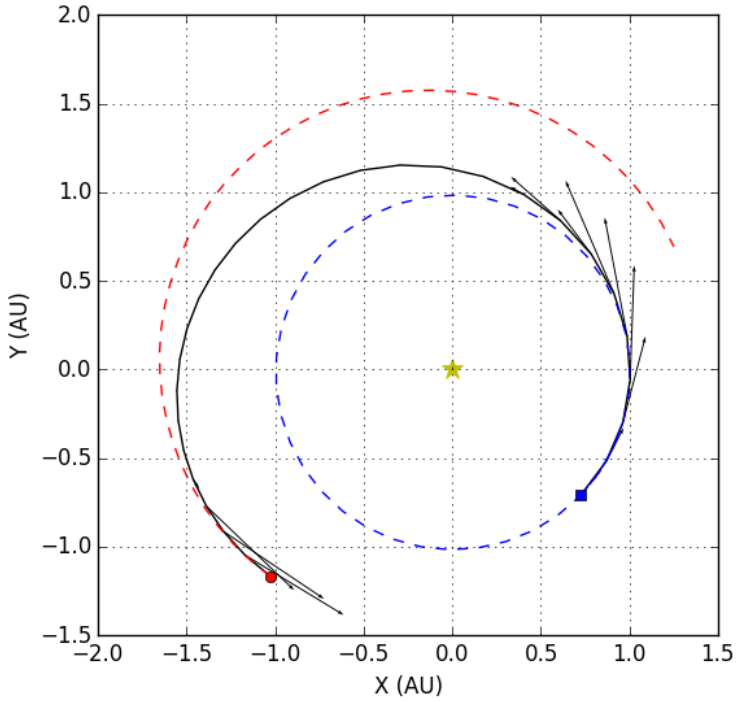

Figure 5: Resulting Earth-Mars rendezvous transfer, now departing in 2009 after applying MBH as a stochastic search step.

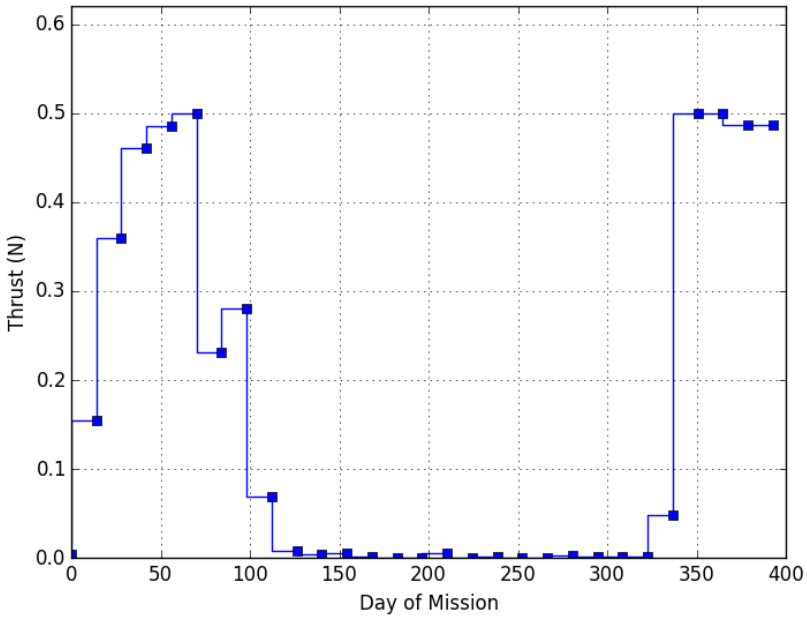

Figure 6: Thrust profile from the final MBH solution.

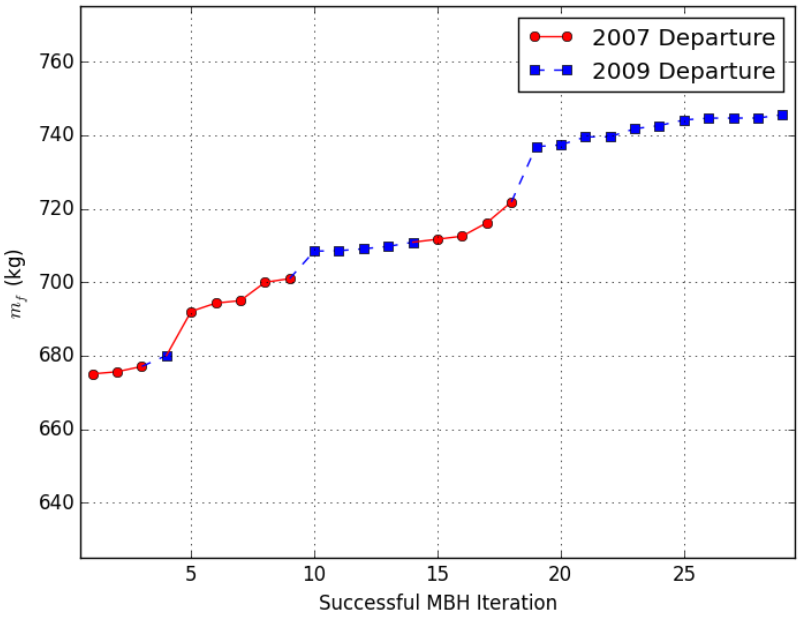

Figure 7: Progress of MBH iterations with indication of hops in synodic period.

three-bang structure of the original thrust profile was reduced to a two-bang Hohmann-like transfer that yielded significant propellant mass savings.

Future work intends to refine the HDDP implementation with several areas of improvement noted in the discussion. The primary challenges to address are premature trust-region collapse and ALM tuning. The addition of the stochastic step has been presented as an outerloop augmentation to HDDP, so inclusion of stochastic search within the HDDP algorithm will be considered. Lastly, the Earth-Mars rendezvous is a relatively simple example. Sophisticated transfers in complex dynamic environments will be the subject of later investigations.

ACKNOWLEDGMENT

Special thanks is extended to Gregory Lantoine for his initial help with HDDP, and to the low-thrust research group at the Colorado Center for Astrodynamics Research for their lively discussions. This work was funded by NASA Grant NNX14AM35H.

REFERENCES

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

[2] G. Lantoine and R. P. Russell, "A Hybrid Differential Dynamic Programming Algorithm for Constrained Optimal Control Problems. Part 1: Theory," *Journal of Optimization Theory and Applications*, Vol. 154, No. 2, 2012, pp. 382–417.

[3] G. Lantoine and R. P. Russell, "A Hybrid Differential Dynamic Programming Algorithm for Constrained Optimal Control Problems. Part 2: Application," *Journal of Optimization Theory and Applications*, Vol. 154, No. 2, 2012, pp. 418–442.

[4] D. H. Jacobson and D. Q. Maybe, "Differential Dynamic Programming," *American Elsevier Publishing Company, Inc.*

[5] C. H. Yam, D. D. Lorenzo, and D. Izzo, "Low-thrust trajectory design as a constrained global optimization problem," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, Vol. 225, 2011, pp. 1243–1251.

[6] J. A. Englander and A. C. Englander, "Tuning Monotonic Basin Hopping: Improving the Efficiency of Stochastic Search as Applied to Low-Thrust Trajectory Optimization," *24th International Symposium on Space Flight Dynamics*, May 2014.

[7] A. R. Conn, N. I. M. Gould, and P. L. Toint, "Trust-Region Methods," *MPS/SIAM*, 2000.

[8] R. Fletcher, "Practical Methods of Optimization," *2nd edn. Wiley,*.

[9] D. P. Bertsekas, "Dynamic Programming and Optimal Control," *Athena Scientific*.

[10] G. Lantoine and R. P. Russell, "A Hybrid Differential Dynamic Programming Algorithm for Robust Low-Thrust Optimization," *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, August 2008.

[11] G. Lantoine and R. P. Russell, "A Fast Second-Order Algorithm for Preliminary Design of Low-Thrust Trajectories," *59th International Astronautical Congress*, September 2008.

[12] G. Lantoine and R. P. Russell, "A Methodology for Robust Optimization of Low-Thrust Trajectories in Multi-Body Environments," *Ph.D. Thesis*, 2010.

[13] J. A. Englander, D. H. Ellison, and B. A. Conway, "Global Optimization of Low-Thrust Multiple-Flyby Trajectories at Medium and Medium-High Fidelity," *AIAA/AAS Astrodynamics Space Flight Mechanics Meeting*, 2014.

[14] P. Prince and J. Dormand, "High order embedded Runge-Kutta formulae," *Journal of Computational and Applied Mathematics*, Vol. 7, Issue 1, March 1981, pp. 67–75.