

## Mars Pinpoint Landing Trajectory Optimization Using Sequential Multiresolution Technique

\* Jisong Zhao<sup>1)</sup>, Shuang Li<sup>2)</sup> and Zhigang Wu<sup>3)</sup>

<sup>1), 2)</sup> College of Astronautics, NUAA, Nanjing 210016, PRC

<sup>3)</sup> School of Aeronautics and Astronautics, DUT, Dalian 116024, PRC

<sup>1)</sup> [zhaojisong@nuaa.edu.cn](mailto:zhaojisong@nuaa.edu.cn)

### ABSTRACT

This paper aims at solving the optimal pinpoint landing trajectory on Mars, using the sequential multiresolution technique. In this technique, the optimal landing trajectory is sought merely in the immediate future whereas the full trajectory is sequentially determined by moving the initial conditions toward the final conditions. Thus, uncertainties and perturbations during the landing are absorbed into the sequential optimization. The benefits of this technique are demonstrated on the pinpoint Mars landing problem. The sequential optimal solutions are closed-loop solutions in fact, so it is possible to code the technique on an onboard computer for practical landing guidance.

### 1. INTRODUCTION

The landing accuracy on the surface of Mars has progressed steadily over the last four decades, from about 200 km of target for the Vikings to 150 km for the Mars Pathfinder to 35 km for the Mars Exploration Rovers. The Mars Science Laboratory (MSL) has an even high landing accuracy, with a delivery of within 10 km of the target. The next generation of Mars missions such as the sample return and human exploration missions will require Mars probes to perform tasks at specific points of interest on the Martian surface. Thus, the performance of pinpoint landings (defined as landing within 100 m of the target) will be of utter importance. Considerable work has been done to improve the landing accuracy (Braun et. al 2007, Shen et. al 2010).

This study focuses on generating a closed loop landing trajectory for powered descent stage using the sequential optimization technique. The sequential technique was originally proposed by Ross et. al in combination with pseudospectral method to solve multi-scale and long-horizon trajectory optimization problems (Ross et. al 2007, Yan et. al 2011). Another sequential optimization technique was developed by Jain et. al in

---

<sup>1)</sup> Lecturer

<sup>2)</sup> Professor

<sup>3)</sup> Professor

combination with multiresolution technique to solve problems with moving targets (Jain et. al 2009). In this technique, the optimal trajectory is sought merely in the immediate future whereas the full trajectory is sequentially determined by moving the initial conditions toward the final conditions. As move forward in time, the problem is solved again on the new horizons, any propagation error that occurs before the new horizon is absorbed automatically into the new initial conditions. Thus the sequential optimization technique is able to provide a closed loop optimal landing trajectory.

This paper is organized as follows. First, the formulation of minimum-fuel powered descent problem is described. Then, a brief overview of the sequential optimization technique is presented, including the generalized dyadic grids, the mesh refinement algorithm based on the generalized dyadic grids and the sequential optimization algorithm. The simulation results and the conclusions are presented in the end.

## 2. MINIMUM FUEL POWERED DESCENT PROBLEM

The powered descent phase of a Mars landing trajectory typically starts when the parachute and heat shield are jettisoned (the handoff), and from then on, thrusters are used to guide the lander to a safe landing site. A uniform gravitational field is used, and the aerodynamic forces and the rotation of the planet are neglected.

A surface-fixed coordinate system  $o-xyh$  is defined to describe the position and velocity of the lander. The origin is anchored at the planned landing point, the  $x$  and  $y$  axes span the horizontal plane, and the  $h$  axis points upward. Let  $v_x$ ,  $v_y$ , and  $v_h$  denote the velocities along the three axes, and let  $m$  denote the mass of the lander. Then the state vector is defined as  $\mathbf{x} = [x, y, z, v_x, v_y, v_h, m]^T$ . Let  $g$  denote the gravitational acceleration on the surface of Mars, and let  $g_0$  denote that at the sea level on Earth. Let there be  $n$  identical thrusters with specific impulse  $I_{sp}$ . Each thruster supplies a maximum thrust  $T$  and is throttled at the same level, which leads to all the thrusters producing the same thrust at any time. The thrusters are mounted such that they are canted at an angle  $\phi$  from the net thrust direction. Thus, the net thrust is  $n \cdot T \cdot \cos \phi$ .

The translational motion of the lander is controlled via modulating the throttle of the thrusters and the direction of the net thrust vector. Let  $\mathbf{u}$  be the vector that points in the direction of the net thrust and has a magnitude of the thruster throttle  $u$ . Let  $u_x$ ,  $u_y$ , and  $v_h$  be the three components of  $\mathbf{u}$  along the  $x$ ,  $y$ , and  $h$  axes respectively. Then  $\mathbf{u} = [u_x, u_y, v_h]^T$  can serve as the control vector. With the above definitions, the differential equations of the translational motion can be written as (Shen et. al 2010)

$$\begin{aligned} \dot{x} &= v_x, & \dot{y} &= v_y, & \dot{h} &= v_h, & \dot{v}_x &= \frac{u_x \cdot n \cdot T \cos \phi}{m} \\ \dot{v}_y &= \frac{u_y \cdot n \cdot T \cdot \cos \phi}{m}, & \dot{v}_x &= \frac{u_x \cdot n \cdot T \cos \phi}{m}, & \dot{m} &= -\frac{u \cdot n \cdot T}{I_{sp} \cdot g_0} \end{aligned} \quad (1)$$

Once the thrusters are switched on, they remain on throughout the descent. Thus, the engine throttle is bounded between two nonzero settings; i.e.,

$$0 < u_{\min} \leq u = \sqrt{u_x^2 + u_y^2 + u_z^2} \leq u_{\max}. \quad (2)$$

The position and velocity at both ends of the powered descent trajectory are specified. The initial condition, denoted by  $\mathbf{x}_0$ , is given at the hand-off, and the lander at the end of the descent is at rest at the planned landing point. The assumption of zero final position and velocity vectors are assumed. Thus

$$\mathbf{x}(t_0) = \mathbf{x}_0, \quad \mathbf{r}(t_f) = \mathbf{0}, \quad \mathbf{v}(t_f) = \mathbf{0} \quad (3)$$

where  $\mathbf{r}(t)$  and  $\mathbf{v}(t)$  denote the position and velocity vectors at time  $t$ .

During the powered descent, the spacecraft obviously cannot travel below the planet surface, which leads to the following state constraint:

$$h(t) > 0 \quad (4)$$

The objective of the optimal control is to guide the lander from the initial conditions to the final conditions with the minimum amount of fuel consumption; that is,

$$J = -m(t_f) \quad (5)$$

where the final time is free.

It is convenient for optimization to transform the physical time interval to the scaled time interval  $\tau \in [0, 1]$  via the following transformation

$$\tau = \frac{t - t_0}{t_f - t_0} \quad (6)$$

### 3. SEQUENTIAL OPTIMIZATION

#### 3.1 Generalized Dyadic Grid

A uniform generalized dyadic grid over the unit interval  $[0, 1]$  is a collection of points obtained by successively subdividing any uniformly distributed initial grid,

$$V_{j,N} = \{\tau_{j,k} \in [0, 1] : \tau_{j,k} = k/(2^j N), 0 \leq k \leq 2^j N\}, 0 \leq j \leq J_{\max} \quad (7)$$

where  $N$  is the number of intervals in the uniformly distributed initial grid,  $j$  denotes the resolution level,  $k$  is the spatial location, and  $J_{\max}$  is a positive integer which specifies the maximum resolution level.

We denote by  $W_{j,N}$  the set of grid points belonging to  $V_{j+1,N}$  but not  $V_{j,N}$ , that is,

$$W_{j,N} = \{\hat{\tau}_{j,k} \in [0, 1] : \hat{\tau}_{j,k} = (2k+1)/(2^{j+1}N), 0 \leq k \leq 2^j N - 1\}, \\ 0 \leq j \leq J_{\max} - 1 \quad (8)$$

Hence,  $\tau_{j+1,k} \in V_{j+1,N}$  if and only if

$$\tau_{j+1,k} = \begin{cases} \tau_{j,k/2}, & k \text{ is even} \\ \hat{\tau}_{j,(k-1)/2}, & \text{otherwise} \end{cases} \quad (9)$$

An example of a uniform generalized dyadic grid is shown in Fig. 1 ( $N=5, J_{\max}=5$ ).

The subspaces  $V_{j, N}$  are nested; that is,  $V_{0, N} \subset V_{1, N} \subset \dots \subset V_{J_{\max}, N}$ , with

$$\lim_{J_{\max} \rightarrow \infty} V_{J_{\max}, N} = [0, 1] \quad (10)$$

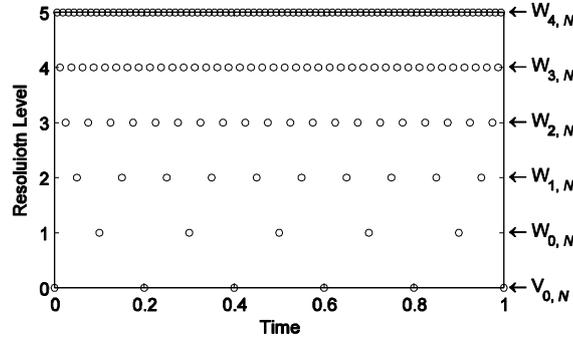


Fig. 1 Example of generalized dyadic grid ( $N = 5, J_{\max} = 5$ )

Furthermore, the sequence of subspaces  $W_{j, N}$  satisfies the property  $W_{j, N} \cap W_{l, N} = \emptyset$  for all  $j \neq l$ . Notice that these dyadic grids are constructed by successive subdivisions. Furthermore, the sets  $V_{j, N}$  and  $W_{k, N}$  for  $k \geq j$  are orthogonal to each other, as are the sets  $W_{j, N}$  and  $W_{l, N}$  for  $j \neq l$  (see Fig. 1). Such subdivision schemes of initial grids can be used to construct refined grids. The idea here is that one can keep only the even-indexed points in the grid and generate the odd-indexed points (at every level) using some interpolation. The use of subdivision schemes simplifies the computations.

The continuous optimal control problem can be easily discretized on a set of generalized dyadic grid (either uniform or nonuniform).

### 3.2 Mesh Refinement Algorithm

Let a nonuniform grid of the form

$$G_{\text{old}} = \left\{ \tau_{j_i, k_i} : \tau_{j_i, k_i} \in [0, 1], 0 \leq k_i \leq 2^{j_i} N, 0 \leq j_i \leq J_{\max}, \text{ for } i = 0, 1, \dots, N_{\tau} \right. \quad (11)$$

$$\left. \text{and } \tau_{j_i, k_i} < \tau_{j_{i+1}, k_{i+1}}, \text{ for } i = 0, 1, \dots, N_{\tau} - 1 \right\} \subset V_{J_{\max}, N}$$

Suppose  $\Phi: [0, 1] \rightarrow \mathbf{R}^{N_r}$  is specified on the grid  $G_{\text{old}}$  such that

$$\Phi_{\text{old}} = \left\{ \phi_l(\tau_{j_i, k_i}) : l = 1, \dots, N_r, \tau_{j_i, k_i} \in G_{\text{old}} \right\} \quad (12)$$

where  $N_r$  is the dimension of the function  $\Phi$ .

To refine the grid  $G_{\text{old}}$ , firstly initialize an intermediate grid  $Grid_{\text{int}} = V_{0, N}$ , with function values

$$\Phi_{\text{int}} = \left\{ \phi_l(\tau_{0, k}) \in \Phi_{\text{old}}, l = 1, \dots, N_r, 0 \leq k \leq N \right\} \quad (13)$$

and set  $j = 0$ . Then the procedures to refine the grid  $Grid_{\text{old}}$  are as follows:

- (1) Find the points that belong to the intersection of  $W_{j, N}$  and  $G_{\text{old}}$

$$\hat{T} = \left\{ \hat{\tau}_{j,k_i} : \hat{\tau}_{j,k_i} \in W_{j,N} \cap G_{\text{old}}, i = 1, 2, \dots, N_{\hat{\tau}} \right\} \quad (14)$$

where  $i$  is the index number of the point in  $\hat{T}$ , and  $N_{\hat{\tau}}$  is the number of nodes in  $\hat{T}$ .

(2) Set  $i = 1$ , and execute the following (a)~(f).

(a) Compute the interpolated function values  $\hat{\Phi}$  at  $\hat{\tau}_{j,k_i} \in \hat{T}$  using the  $p$ th-order essentially non-oscillatory (ENO) interpolation.

(b) Calculate the interpolative errors  $\mathbf{d}_{j,k_i}$  at the point  $\hat{\tau}_{j,k_i}$

$$\mathbf{d}_{j,k_i} = \left| \Phi(\hat{\tau}_{j,k_i}) - \hat{\Phi}(\hat{\tau}_{j,k_i}) \right| \quad (15)$$

If every element in  $\mathbf{d}_{j,k_i}$  is below its threshold in  $\epsilon$ , then reject  $\hat{\tau}_{j,k_i}$  and go to step (f); otherwise add  $\hat{\tau}_{j,k_i}$  to the intermediate grid  $Grid_{\text{int}}$  and move on to the next step. The thresholds are chosen so that they are small and constant in the immediate future, and increase exponentially approaching the final time (Jain et. al 2009).

(c) Add to  $Grid_{\text{int}}$   $N_1$  points on the left and  $N_1$  points on the right neighboring to the point  $\hat{\tau}_{j,k_i}$  in  $W_{j,N}$ , i.e.  $\{\hat{\tau}_{j,k_i+l}, l = -N_1, \dots, N_1, l \neq 0\}$ .

(d) In a similar way, add to  $Grid_{\text{int}}$   $2N_2$  neighboring points at the next finer level, i.e.  $\{\hat{\tau}_{j+1, 2k_i+l}, l = -N_2+1, \dots, N_2\}$ .

(e) Add the function values at all the newly added points to  $\Phi_{\text{int}}$ . If the function value at any of the newly added points is not known, interpolate the function value at that point from  $Grid_{\text{old}}$  and  $\Phi_{\text{old}}$  using the  $p$ th-order ENO interpolation.

(f) Increment  $i$  by 1. If  $i \leq N_{\hat{\tau}}$ , go to Step 2a, otherwise move on to the next step.

(3) Increment  $j$  by 1. If  $j \leq J_{\text{max}}-2$ , go to Step 1, otherwise move on to the next step.

(4) Terminate the algorithm. The final nonuniform grid is  $Grid_{\text{new}} = Grid_{\text{int}}$  and the corresponding function values are in the set  $\Phi_{\text{new}} = \Phi_{\text{int}}$ .

### 3.3 Sequential Optimization Algorithm

The basic idea behind the sequential algorithms is to solve the trajectory optimization problem at hand over a suitably horizon. As we continue to move forward in time, we solve the problem again on the new horizons, using the solution of the previous horizon as an initial guess (Ross et. al 2007). The procedures are as follows:

(1) Partition the scaled time interval  $[0, 1]$  into  $N_s$  segments with  $N_s+1$  nodes,  $0 = \tau^0 < \tau^1 < \dots < \tau^{N_s} = 1$ . These segments need not be uniformly spaced. Set  $i = 0$ .

(2) Solve the optimal control problem on the horizon  $\{t_0, t_f\}$ , with the grid is refined using the procedures given in Section 3.2, where  $t_f$  is either fixed or free.

(3) Propagate the differential equation from  $t^j$  to  $t^{j+1}$  using  $\mathbf{x}_0$  as the initial condition and the ENO interpolation of the controls,  $\mathbf{u}^{j+1}(t)$ ,  $t \in [t^j, t^{j+1}]$  based on the discretized control solution obtained in step 2, where  $t^j$  is the physical time corresponding the scaled time  $\tau^j$  in step 1. This step generates a continuous-time trajectory,  $\mathbf{x}^{j+1}(t)$ ,  $t \in [t^j, t^{j+1}]$ . This propagation is done numerically via some high-precision propagator, say the standard 4th/5th-order Runge–Kutta method.

(4) if  $i < N_s-1$ , set  $\mathbf{x}_0 = \mathbf{x}^{i+1}(t^{i+1})$  and  $t_0 = t^{i+1}$ , increase  $i$  by 1 and go to step 2; other-

wise terminate the algorithm. The optimal trajectory is given by the chain, i.e.  $\{\mathbf{x}(t), t \in [t_0, t_f]\} = \{\mathbf{x}^1(t), t \in [t^0, t^1]; \mathbf{x}^2(t), t \in [t^1, t^2]; \dots; \mathbf{x}^N(t), t \in [t^{N-1}, t^N]\}$ . Similarly, the corresponding controls are given by  $\{\mathbf{u}(t), t \in [t_0, t_f]\} = \{\mathbf{u}^1(t), t \in [t^0, t^1]; \mathbf{u}^2(t), t \in [t^1, t^2]; \dots; \mathbf{u}^N(t), t \in [t^{N-1}, t^N]\}$ .

#### 4. NUMERICAL SOLUTION

It is assumed that the lander carries 400 kg of fuel at the hand-off, with a total wet mass of 1905 kg. Other parameters of the lander used in the simulations are:

$$\begin{aligned} n &= 6, & \phi &= 27 \text{ deg}, & g &= 3.7114 \text{ m/s}^2 \\ g_0 &= 9.80655 \text{ m/s}^2, & T &= 3100 \text{ N}, & I_{sp} &= 255 \text{ s} \\ u_{\min} &= 0.3 & u_{\max} &= 0.8 \end{aligned}$$

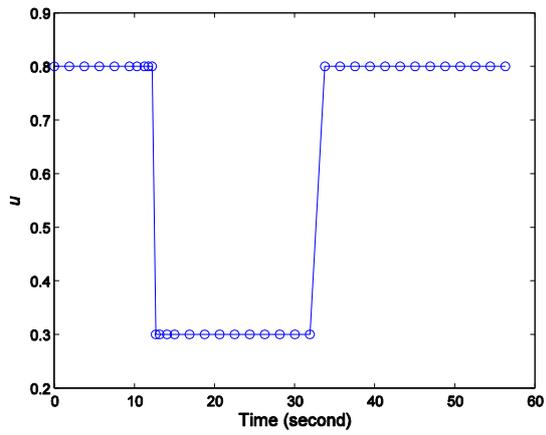
The boundary conditions are:

$$\begin{aligned} t = 0 : & \quad x = 1900 \text{ m}, y = 0 \text{ m}, h = 3100 \text{ m}, v_x = 40 \text{ m/s}, v_y = 20 \text{ m/s}, v_h = -50 \text{ m/s} \\ t = t_f : & \quad x = 0 \text{ m}, y = 0 \text{ m}, h = 0 \text{ m}, v_x = 0 \text{ m/s}, v_y = 0 \text{ m/s}, v_h = 0 \text{ m/s} \end{aligned}$$

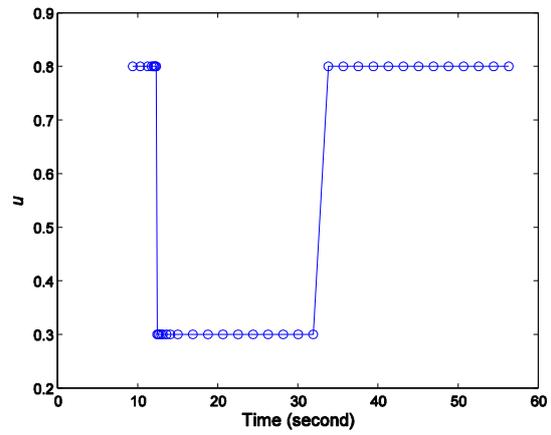
In this example, we choose  $N_s=5$  and  $N=5$ , i.e. the time interval is divided into 5 segments, and 6 base nodes is used in each segment to start the mesh refinement. The control magnitude found in each sequential optimization is shown in Fig. 2~4. It is clear that the algorithms autonomously discretize the trajectory with more nodes near the current time (not necessarily uniformly placed) while using a coarser grid for the rest of the trajectory to capture the overall trend. The combined trajectory and the control found on different horizons are shown in Fig. 3 and Fig.4. The optimal maximum–minimum–maximum throttle profile is captured accurately with little noise.

The minimal fuel consumption is 291.258 kg, which is very close to that of 291.255 kg obtained by solve the problem directly without using the sequential optimization, in which case the mesh is refined on the whole horizon. For further comparison, Fig. 5 plots the optimal thrust throttle profile along with that obtained from the direct optimization. The two profiles are so close that the differences are nearly invisible, which verifies the high accuracy of the sequential optimization in this case. However, the controls obtained from the sequential optimization are closed-loop controls by absorbing the accumulation errors in previous trajectories into the new initial conditions, preventing them from being carried over to the remaining part of the trajectory.

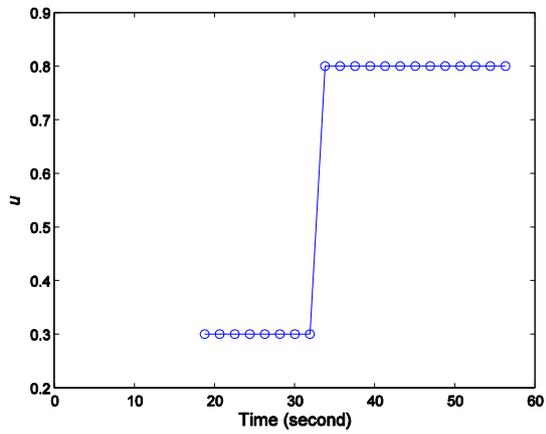
Currently the time for generating each sequential optimal trajectory in this case was about 1.6 seconds on a desktop computer. Note that the computation was all carried out in the MATLAB language for convenient. The run time can be significantly reduced by optimizing the code and coding the algorithms in C or FORTRAN.



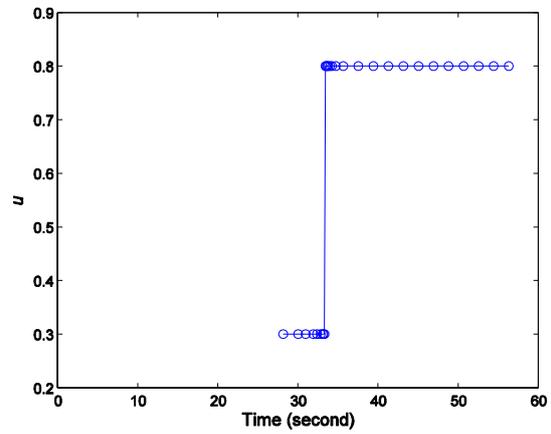
(a) Control magnitude for horizon 1



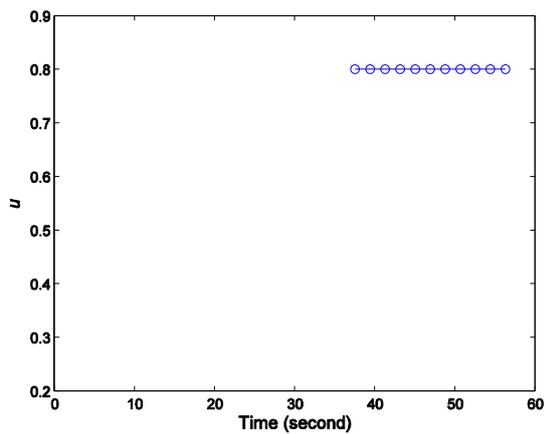
(b) Control magnitude for horizon 2



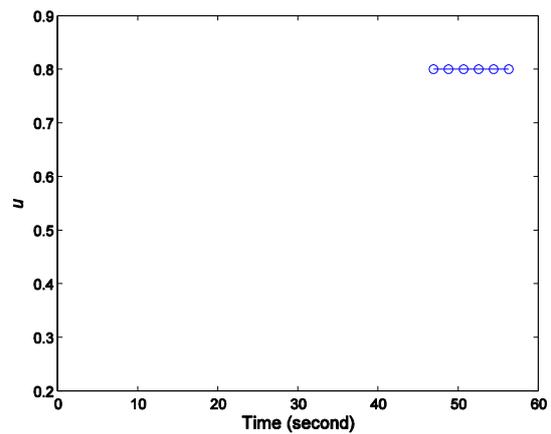
(c) Control magnitude for horizon 3



(d) Control magnitude for horizon 4



(e) Control magnitude for horizon 5



(f) Control magnitude for horizon 6

Fig. 2 Control magnitude for horizon 1, 2, 3, 4, 5 and 6

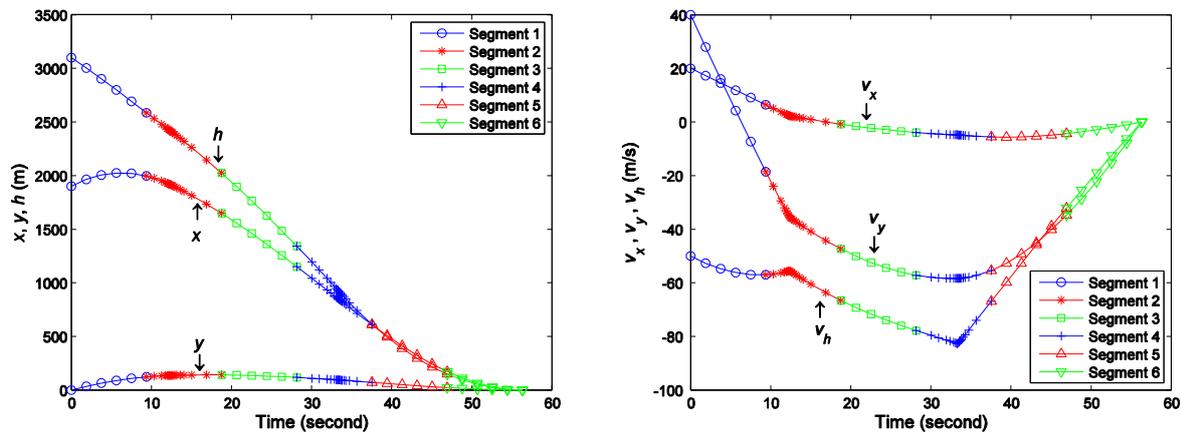


Fig. 3 Optimal states found on different horizons: (a) Position; (b) Velocity

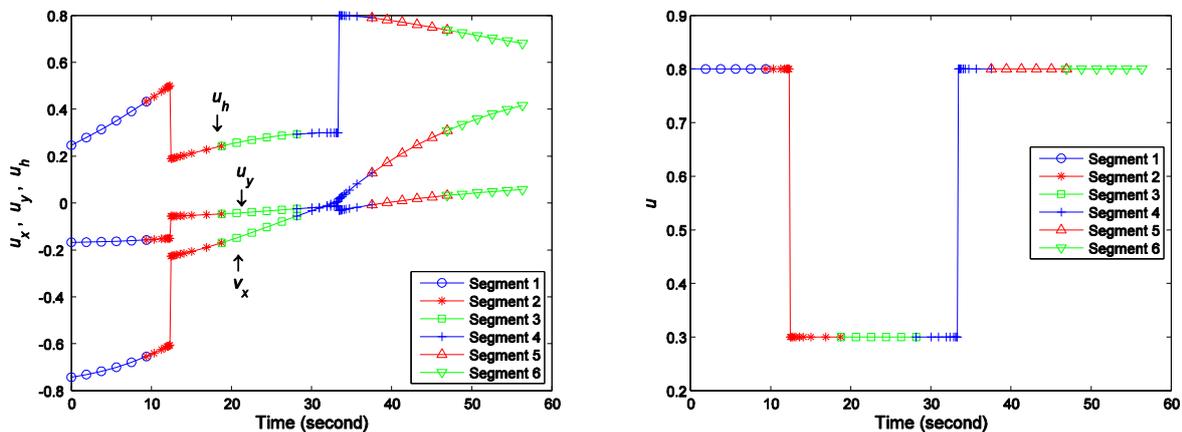


Fig. 4 Optimal controls found on different horizons: (a) Component; (b) Magnitude

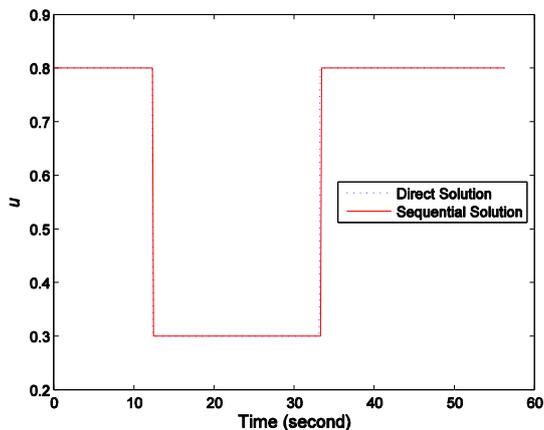


Fig. 5 Comparison of the thrust throttle profile

## **5. CONCLUSIONS**

The sequential optimization technique is used to solve the minimal fuel Mars pin-point landing problem in this paper. The multiresolution technique based on generalized dyadic grids is used to refine the mesh in each sequential optimization. Numerical simulation results indicate that the sequential technique is able to solve the Mars pin-point landing problem accurately, with the optimal maximum–minimum–maximum throttle profile captured exactly. Furthermore, the optimal solutions found by the sequential optimization technique are closed-loop solutions, so it is possible to code the technique on an onboard computer for practical landing guidance in the future.

## **Acknowledgements**

The work described in this paper was supported partially by “the Fundamental Research Funds for the Central Universities”, NO. NS2016087. The authors greatly appreciate the above financial support.

## **REFERENCES**

- Braun, R. D., and Manning, R. M. (2007), “Mars Exploration Entry, Descent and Landing Challenges,” *Journal of Spacecraft and Rockets*, **44**(2), 310–323.
- Shen, H., Seywald, H. and Powel, R.W. (2010), “Desensitizing the minimum-fuel powered descent for mars pinpoint landing”, *Journal of Guidance, Control, and Dynamics*, **33**(1), 108-115.
- Ross, I. M., Gong, Q., and Sekhavat, P. (2007), “Low-thrust, high-accuracy trajectory optimization”, *Journal of Guidance, Control, and Dynamics*, **30**(4), 921-933.
- Yan, H., Gong, Q., Park, C. D. et. al. (2011), “High-accuracy trajectory optimization for a trans-earth lunar mission”, *Journal of Guidance, Control, and Dynamics*, **34**(4), 1219-1227
- Jain, S. and Tsiotras, P. (2009), “Sequential Multiresolution Trajectory Optimization Schemes for Problems with Moving Targets”, *Journal of Guidance, Control, and Dynamics* **32**(2), 488-499.