$$\hat{T} = \left\{ \hat{\tau}_{j,k_i} : \hat{\tau}_{j,k_i} \in W_{j,N} \bigcap G_{\text{old}},\ i = 1, 2, \cdots, N_{\hat{\tau}} \right\} \tag{14}$$

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 $\hat{\mathbf{\Phi}}$ at $\hat{\tau}_{j,k_i} \in \hat{T}$ using the pth-order essentially non-oscillatory (ENO) interpolation.

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

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

If every element in $\mathbf{d}_{j,k_i}$ is below its threshold in $\boldsymbol{\varepsilon}$, 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. $\{\hat{\tau}_{j,k_i+l},\ l = -N_1, \cdots,\ N_1,\ l \neq 0\}$.

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

(e) Add the function values at all the newly added points to $\mathbf{\Phi}_{\text{int}}$. If the function value at any of the newly added points is not known, interpolate the function value at that point from Grid_{old} and $\mathbf{\Phi}_{\text{old}}$ using the pth-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 $j \leq J_{\max}$-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 $\mathbf{\Phi}_{\text{new}} = \mathbf{\Phi}_{\text{int}}$.

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 = $\tau^0 < \tau^1 < \ldots < \tau^{Ns} = 1$. 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^i to t^{i+1} using $\mathbf{x}_0$ as the initial condition and the ENO interpolation of the controls, $\mathbf{u}^{i+1}(t)$, $t \in [t^i, t^{i+1}]$ based on the discretized control solution obtained in step 2, where t^i is the physical time corresponding the scaled time τ^i in step 1. This step generates a continuous-time trajectory, $\mathbf{x}^{i+1}(t)$, $t \in [t^i, t^{i+1}]$. 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. $\{\mathbf{x}(t), t\in[t_0, t_f]\}=\{\mathbf{x}^1(t), t\in[t^0, t^1]; \mathbf{x}^2(t), t\in[t^1, t^2]; ...; \mathbf{x}^N(t), t\in[t^{N-1}, t^N]\}$. Similarly, the corresponding controls are given by $\{\mathbf{u}(t), t\in[t_0, t_f]\}=\{\mathbf{u}^1(t), t\in[t^0, t^1]; \mathbf{u}^2(t), t\in[t^1, t^2]; ...; \mathbf{u}^N(t), t\in[t^{N-1}, t^N]\}$.

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:

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

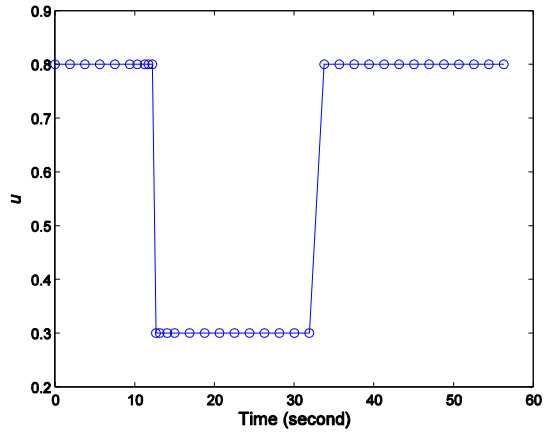
The boundary conditions are:

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

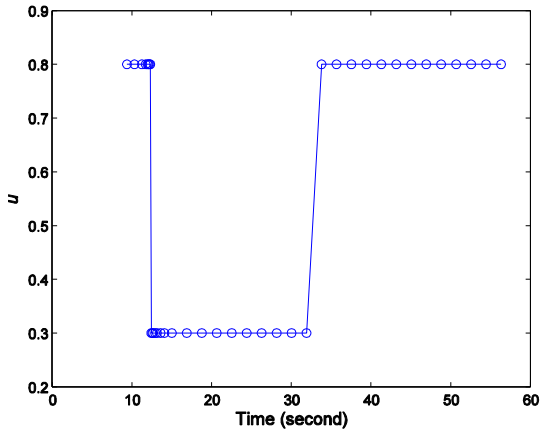
In this example, we choose N_s=5 and N=5, i.e. the time interval is divided into 5 segments, and 6 base nodes is used in each segment to start the mesh refinement. The control magnitude found in each sequential optimization is shown in Fig. 2~4. It is clear that the algorithms autonomously discretize 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.

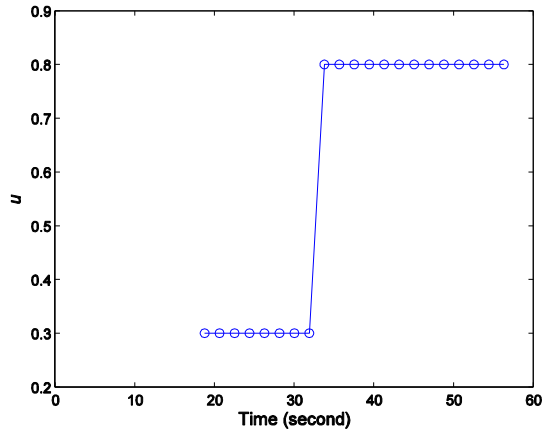
Currently the time for generating each sequential optimal trajectory in this case was about 1.6 seconds on a desktop computer. Note that the computation was all carried out in the MATLAB language for convenient. The run time can be significantly reduced by optimizing the code and coding the algorithms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



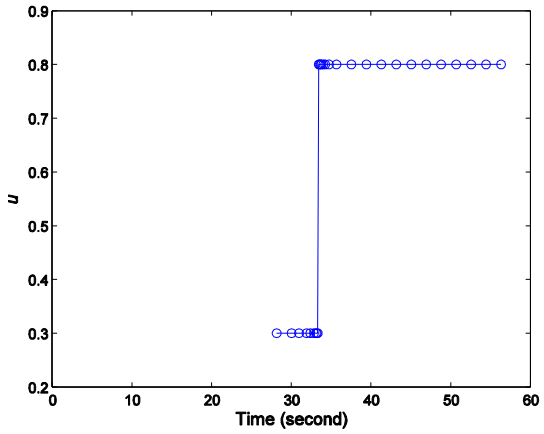
(a) Control magnitude for horizon 1



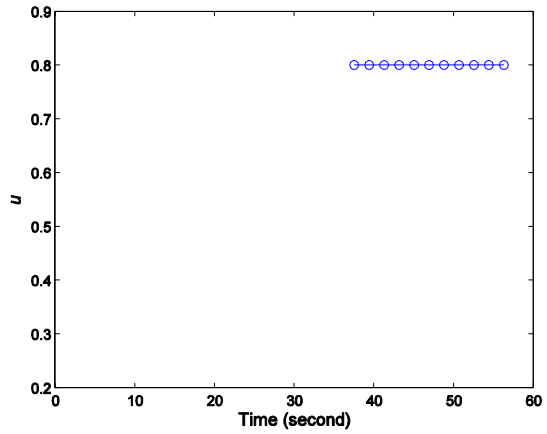
(b) Control magnitude for horizon 2



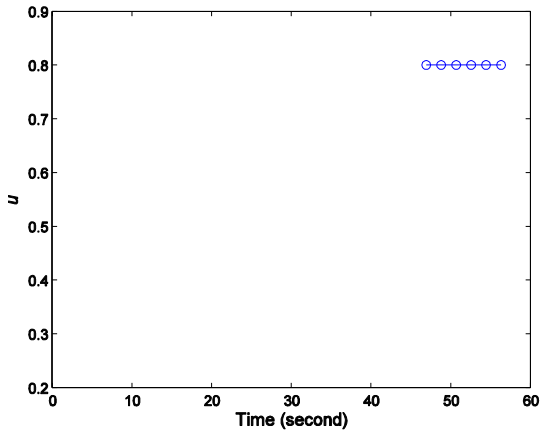
(c) Control magnitude for horizon 3



(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

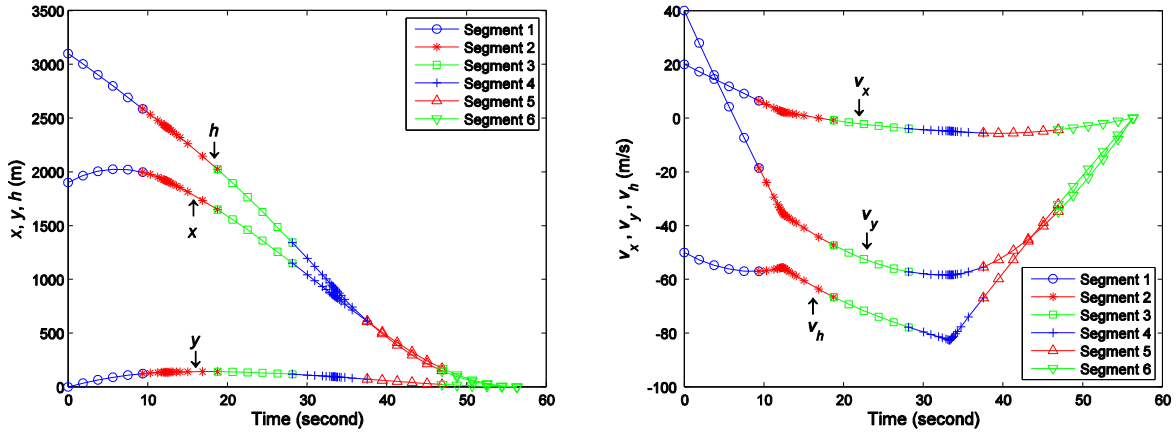


Fig. 3 Optimal states found on different horizons: (a) Position; (b) Velocity

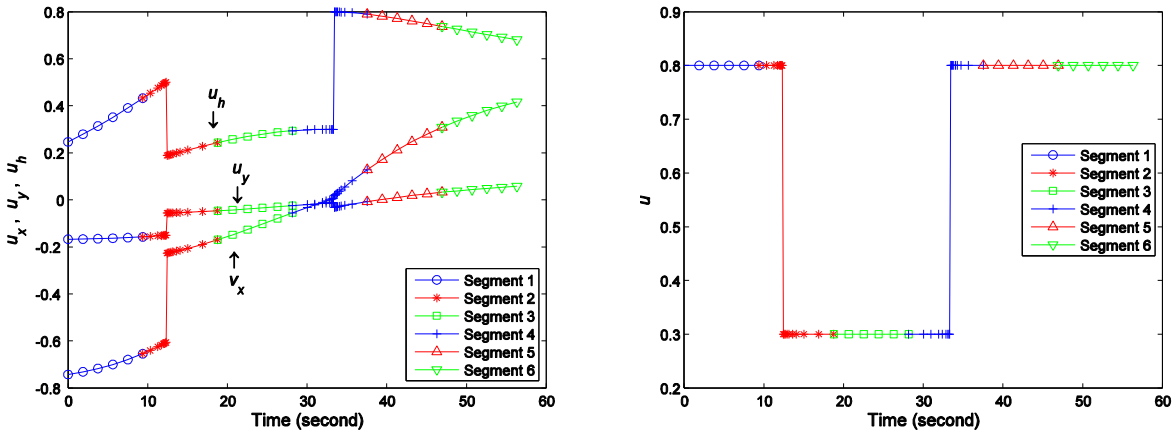


Fig. 4 Optimal controls found on different horizons: (a) Component; (b) Magnitude

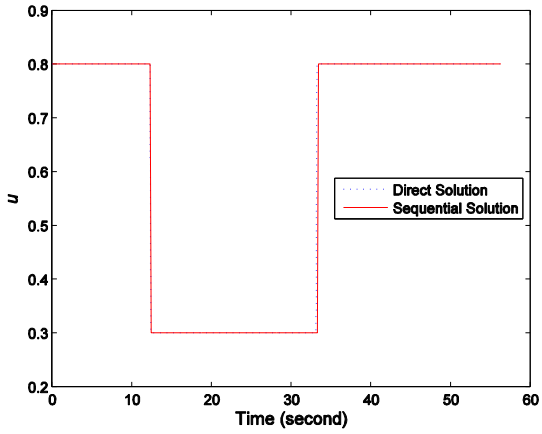


Fig. 5 Comparison of the thrust throttle profile

5. CONCLUSIONS

The sequential optimization technique is used to solve the minimal fuel Mars pinpoint 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 pinpoint 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.