CASI

An Investigation of Terminal Guidance and Control Techniques for a Robotic Mars Lander

by
Brian K. Birge
and
Gerald Walberg

Performance Report submitted under
NASA Grant NAG-1-2086
for the period 4/15/98 to 5/15/99

North Carolina State University
Mars Mission Research Center
May 13, 1999

Abstract.

Two separate Mars lander touchdown scenarios are considered and compared to a baseline study with the goal of minimizing the landed distance to a specified location on the Mars surface. This study considers a set of points from parachute handoff to touchdown on the surface. The first scenario examines the effect of thrust vectoring while the parachute is deployed and includes an algorithm for determining targeting initial guesses. The second considers a reverse gravity turn to a hover condition 500 meters above the surface and then uses lateral thrusting to minimize the range to target. The effects of both scenarios on fuel usage and targeting success are discussed.

Introduction.

For future planetary exploration missions, either robotic or manned, it is desirable to precisely target a lander's touchdown point. Perhaps there has been a previous robotic mission that requires a follow-up robotic mission in order to retrieve collected samples and return them to earth. Perhaps there are specific geographical features that require close-up study. Regardless, a need for precision landing within 100 meters of a specified geographic location exists. Current state of the art can only achieve positioning to within approximately 10 kilometers. This study examines two general methods for controlling a lander as it touches down on a specific point on the Mars landscape. All model parameters and constants are taken from and designed to be compatible with the M2001 specifications when possible.

Nomenclature.

alppc1	first coefficient of alpha equation (POST3D)
alppc2	second coefficient of alpha equation (POST3D)
asmax	maximum acceleration (POST3D)
betpc1	first coefficient of beta equation (POST3D)
betpc2	second coefficient of beta equation (POST3D)
critr	criterion used to activate events (POST3D)
depph	event at which targeting is to be satisfied (POST3D)
deptl	tolerances on dependent variables (POST3D)
depval	targets of dependent variables (POST3D)
depvr	names of dependent variables (POST3D)
diamp1	diameter of parachute #1 (POST3D)
dprng1	dot product range to reference long, lat (POST3D)
etapc1	throttling parameter polynomial coefficient one (POST3D)
gdalt	vertical altitude above oblate planet (POST3D)
gdlat	geodetic latitude (POST3D)
idepvr	type of constraint desired for dependent variables (POST3D)
ifdeg	allows degrees to be used in targeting (POST3D)
iguid(1)	guidance desired (POST3D)
iguid(10)	separate channel option for pitch (POST3D)
iguid(11)	separate channel option for bank angle (POST3D)
iguid(13)	relative yaw angle reference option flag (POST3D)
iguid(2)	selects either independent or identical channel steering (POST3D)
iguid(9)	separate channel option for yaw (POST3D)
indph	event that starts perturbing independent variables (POST3D)
indvr	names of independent variables (POST3D)
isp	specific impulse
ispv	vacuum specific impulse (POST3D)
long	longitude, degrees or meters (POST3D)
MarsGram	program modeling mars atmosphere
Matlab	high level programming language
neng	number of engines (POST3D)
npc(7)	acceleration limit option flag (POST3D)
opt	optimization flag (POST3D)
optph	optimize by this event (POST3D)
optvar	optimization variables (POST3D)
phi	angle between due North and target
POST3D	Program to Optimize Simulated Trajectories 3D version
rn	nose radius
sref	aerodynamic surface area
theta	angle between due North and velocity vector
ur	first component of lander horizontal velocity planet relative (POST3D)
velr	vehicle velocity relative to rotating planet (POST3D)
vr	second component of lander horizontal velocity planet relative (POST3D)
wgtsg	vehicle gross weight (POST3D)
wjett	jettisoned weight (POST3D)
wpropi	initial propellant weight (POST3D)
wr	vertical velocity planet relative (POST3D)

Lander Scenarios.

Scenario 1 – Thrust on Parachute.

At a set of initial (parachute handoff) conditions supplied by LaRC, the lander deploys a ballistic parachute to decelerate. Based on the handoff conditions and the desired touchdown point, thrust vectoring while on the parachute is used to minimize the range to the target. Once the parachute is jettisoned, the lander performs a reverse gravity turn with thrust vectoring to achieve desired terminal velocity components and altitude (touchdown).

Scenario 2 – Hover and Thrust Laterally.

At a set of initial (parachute handoff) conditions supplied by LaRC, the lander deploys a ballistic parachute for deceleration. Upon parachute jettison, a reverse gravity turn is performed to achieve a hover condition 500 meters above the Mars surface. At this point lateral thrusting is used to fly the lander to the desired target and achieve desired terminal velocity components.

Descent Simulations.

Simulation Algorithms.

The programs used in this study were Matlab, POST3D, and MarsGram. Matlab was used as a shell for controlling the batch runs as well as for visualization and pre/post processing of data. Program To Optimize Simulated Trajectories (POST3D) was used to perform the actual lander simulations on a case by case basis. MarsGram was used to build an atmosphere, which was then converted to tables for use in POST3D.

Scenario 1 – Thrust on Parachute

Input Deck Setup

There are three input decks used for this study:

1) bkb3firstguesses-noopt-nothrust.inp

2) bkb3firstguesses-nothrust.inp
3) bkb3firstguesses.inp

bkb3firstguesses-noopt-nothrust.inp

The first input deck (bkb3firstguesses-noopt-nothrust) is the baseline case. This takes the initial conditions, and pops open the parachute to slow down. There is no thrust applied while the parachute is deployed. Once the parachute is jettisoned, the control system kicks in and performs a reverse gravity turn to touchdown. The lander's touch down latitude and longitude is not targeted but ending velocity and altitude conditions are targeted.

These desired end conditions are:

```
0.1  <= ur    <= 2.1 m/s
0.1  <= vr    <= 2.1 m/s
        wr     = 2.0 m/s
2499 <= gdalt <= 2500m
```

(2500m is surface at landed latitude and longitude)

The above-discussed input deck gives us the baseline landed ellipse. It is desired to shrink the magnitude of this baseline ellipse as much as possible. Before discussing the two input decks that attempt to solve that problem, let's look in detail at the baseline case.

Event1 – initial setup and parachute deploy

- Atmosphere input as tables (from a previous MarsGram run)
- MarsGram winds with appropriate multipliers input as tables
- Initial pos/vel input in inertial coordinates from M2001 Monte Carlo analysis
- sref=4.5238934 (aero surface area)
- rn= 0.6638 (nose radius)
- Gravity model uses oblate planet with spherical harmonics j2 through j6
- guidance uses inertial aerodynamic angles
- wgtsg=2176.811 N (585.479 kg) [vehicle weight including parachute]
- wpropi=372.0 N (100 kg) [initial weight of propellant]
- neng=2, [2 engines]
- ispv(1)=553.9,553.9, [Mars Isp (Earth Isp = 210 sec, mono-propellant hydrazine rocket engine)]
- primary engine is pointed out X body axis
- secondary engine is pitched -90 degrees
- vehicle drag coefficient = 1.7
- parachute drag coefficient = 0.41
- Parachute deployment rate set so full deployment occurs in about 3 sec.

Event22 – parachute diameter limit

- at diampl=13
- wgtsg = 1937.297 N [after dropping heat shield]
- parachute diameter is limited to 13m

Event50 – jettison parachute and turn on primary engine (start of reverse gravity turn)

- at gdalt=3500m
- wjett = 276.702 N [weight jettisoned]
- sref=2.0 [surface area reflects heatshield jettison]
- turn on primary engine and start targeting (using relative aero-angles)
- vehicle drag coefficient = 2.0 [increased to reflect non-aerodynamic shape of lander]

Event70 – 2nd event during reverse gravity turn (allows adjustment of controls)

- at gdalt=3000m
- no action taken except gives guidance a chance to adjust values

Event80 – surface touchdown and turn off primary engine

- critr='wr'
- value = 2.0
- at this event gdalt is targeted to 2500m

Event500 – endproblem

bkb3firstguesses-nothrust.inp

The second case (bkb3firstguesses-nothrust.inp) is the same as the baseline case except now optimization is used to minimize the dot product distance to reference latitude and longitude. The reference (desired landed position) latitude and longitude points are:

```
Latitude:  15.18 degrees
Longitude: 264.760 degrees
```

The events are the same as the first case but now the optimization procedure is activated.

bkb3firstguesses.inp

The third case (bkb3firstguesses.inp) is the same is the second case except now there is controlled thrust applied during the parachute phase. The events are the same as the previous two cases with two events added:

Event25 – Turn on secondary engine while on parachute

- at same time as event22 (right when parachute is fully deployed) turn on secondary engine

- switch to relative euler angles for guidance

Event27 – Turn off secondary engine

- at gdalt=4000m turn off secondary engine
- switch back to relative aerodynamic angle guidance

Controls.

Baseline case - bkb3firstguesses-noopt-nothrust.inp.

For all three cases, there is a common set of controls used for the reverse gravity turn. These are included for control during events 50 and 70 (two part reverse gravity turn). As can be seen, the baseline case control inputs are relative aerodynamic angles and thrust magnitude. The initial guesses for events 50 and 70 are chosen to lie along the velocity vector but pointed backwards so that braking takes place. The targeting is designed to be achieved when the vertical velocity is 2.0 m/s (event 80). The relevant parts of the input deck are shown:

```
Independent   Control   inputs   (baseline
case)
     indph= 50,50,50,50,50,
            70,70,70,70,70,
     indvr= 'etapc1','alppc1','alppc2',
            'betpc1','betpc2',

            'etapc1','alppc1','alppc2',
            'betpc1','betpc2',

Dependent   Variable   inputs   (baseline
case):
     depph = 80,80,80,80,80,
     depvr ='gdalt','ur','vr','ur','vr',
     deptl = 1.0,0.1,0.1,0.1,0.1,
     depval= 2500,2.0,2.0,0.0,0.0,
     idepvr=0,1,1,-1,-1,/constraint type
```

Optimized baseline case - bkb3firstguesses-nothrust.inp

The controls are the same as for the baseline case except optimization is now used. The idea is to minimize the dotproduct range to the reference point.

```
     opt        = -1,
     optph      = 80,
     optvar     = 'dprngl',
```

Thrust on the parachute case - bkb3firstguesses.inp

The controls for this case are the exact same as the optimized baseline case with added controls at event 25. These added controls are relative euler angles and thrust magnitude. The initial guesses are taken from an algorithm specifically developed for this case.

```
Independent Control inputs
     indph     = 25,25,25,25,25,
                 50,50,50,50,50,
                 70,70,70,70,70,
     indvr= 'etapc1','pitpc1','pitpc2',
            'yawpc1','yawpc2',

            'etapc1','alppc1','alppc2',
            'betpc1','betpc2',

            'etapc1','alppc1','alppc2',
            'betpc1','betpc2',
*include '../../bkb3guess.dat',
```

First Guesses.

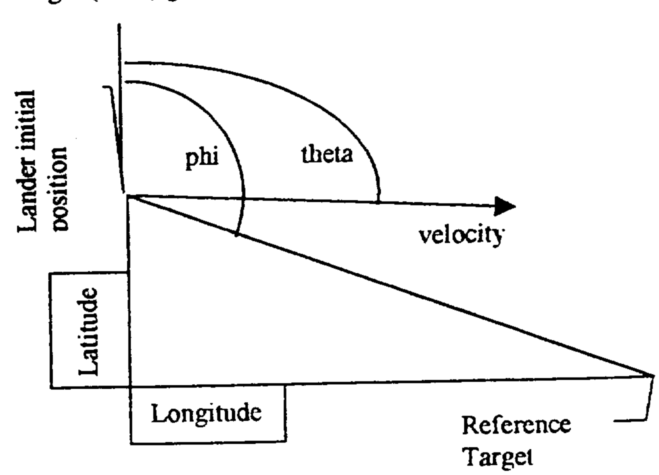
To get first guesses (tabulated in the include file 'bkb3guess.dat') for 'yawr' and 'pitr' used while thrusting on parachute we look at several cases. First we determine the initial conditions of the lander at parachute hand-off, then compare to the targeted reference point on the ground. In general, the initial conditions indicate that the lander is moving easterly and will overshoot the target. The initial conditions examined consisted of 1980 points supplied by LaRC from a Monte Carlo analysis studying the position ellipse at parachute handoff.

To determine the various conditions, the problem was split into two sub-problems. First it was considered how to correctly yaw the craft to achieve targeting and second it was considered how to correctly pitch the craft for same. After examining the geometry for both sub-problems, some heuristics were developed.

For yaw:

Given initial conditions of latitude, longitude, and azimuth heading (relative to geographic north) and given reference target of latitude, longitude we can build a right triangle with the initial and target positions at corners. The x-axis is east and y-axis is north.

Then, looking at the figure we can determine the angle of the line between initial and target positions (with respect to north). Comparing this angle (phi) with the azimuth-heading angle (theta) gives us the geometry.



To choose the initial guess of yawr used for the POST targeting the following rules were developed:

Case	verbal description	conditions	first guess choice
1	reference pt NW of lander track	1) phi>theta 2) reference longitude west of initial longitude	yawpc1 = azvelr+180+45 (thrust backwards and northwards)
2	reference pt SW of lander track	1) phi<theta 2) reference longitude west of initial longitude	yawpc1 = azvelr+180-45 (thrust backwards and southwards)
3	reference pt NE of lander track	1) phi>theta 2) reference longitude east of initial longitude	yawpc1 = azvelr-90 (thrust forward and northwards)
4	reference pt SE of lander track	1) phi<theta 2) reference longitude east of initial longitude	yawpc1 = azvelr+90 (thrust forward and southwards)
5*	reference pt E of lander track	1) phi>theta 2) reference longitude west of initial longitude	yawpc1 = azvelr+180 (thrust backwards)
6*	reference pt W of lander track	1) phi>theta 2) reference longitude west of initial longitude	yawpc1 = azvelr (thrust forward)

For pitch:
Similar methodology was used for calculating pitch except the cases were fewer since the yaw rotation will take care of whether we are too east or west initially. Coordinates of the formed right triangle were altitude (converted to degrees) and ground track in degrees. The right triangle was formed and the following heuristics developed:

Case	verbal description	conditions	first guess
7	lander will undershoot target	gammar more negative than angle to target	pitch up 30 degrees (pitpc1=30)
8	lander will overshoot target	gammar more positive than angle to target	pitch down 30 degrees (pitpc1=-30)
9*	no under/overshoot	gammar close to angle to target	follow current velocity vector (pitpc1=0)

*In case 5,6,9 an arbitrary threshold value was used to determine 'close'. For example, phi and theta are considered aligned if they are within some small deviation of each other.

Also, the initial guesses for the 2nd coefficients (time) of the yaw and pitch equation were set to zero. The initial guess for 'etapc1' for the reverse gravity turn (events 50,70) was set at 1.0 while the initial guess for etapc1 during the thrusting on parachute was set at 0.2.

Scenario 2 – Hover and Thrust Laterally

Input Deck Setup

There are three input decks used for this study:
1) target3000m.inp
2) hover20deg.inp
3) hover30deg.inp

target3000m.inp

This takes the initial conditions (same as in the 'thrust on parachute' study), and pops open the parachute to slow down. There is no thrust applied while the parachute is deployed. Once the parachute is jettisoned, the control system kicks in and performs a reverse gravity turn to hover at +500m above surface. The lander's hover latitude and longitude are not targeted but hover velocity and altitude conditions are targeted.

These desired hover conditions are:

```
 0.1 <= ur    <= 2.1 m/s
 0.1 <= vr    <= 2.1 m/s
        wr     = 2.0 m/s
2999 <= gdalt <= 3000m
```

(2500m is surface at landed latitude and longitude, this is +500m above surface)

The events are all the same as for the baseline case of 'thrusting on parachute' but the criteria is slightly changed for the following events:

Event50 – Jettison Parachute & turn on primary engine (start of reverse gravity turn)

- `this now happens at gdalt=4000m (was 3500m)`

Event70 – 2nd event during reverse gravity turn (allows adjustment of controls)

- `this now happens at 3500m (was 3000m)`

Event80 – Hover condition

- `this now happens at 3000m, i.e. +500m above planet surface (was 2500m - planet surface)`

hover20deg.inp (and hover30deg.inp)

The terminal states of the above case are then used as initial conditions for the next two input decks. Both decks are different from the 'thrust on parachute' scenario in that they target directly to the reference latitude and longitude rather than just trying to minimize the landed distance. There is no first guess algorithm such as for the first scenario. The targeting first guesses assume the reference point is somewhat south of the initial conditions (which indeed it is for most of the points tested). The targeting also starts immediately. The only difference between hover20deg.inp and hover30deg.inp is the choice of pitch angle used for lateral translation.

Event1 – initial setup

- `*include '../../hoverinitcond.dat', (this includes the end conditions from targ3000m.inp)`
- `inertial euler angles are used for steering`
- `npc(7) = 1, limits asmax to 1.06418`
- `asmax = 1.06418,`
- `pitpc(1) = -20.0, (this is -30.0 for hover30deg.inp)`
- `primary engine on (same one as used for lateral movement in scenario one, it is pitched -90 degrees)`
- `All other constants and parameters are basically the same as all other decks.`

The reason asmax is limited is to prevent the targeting algorithm from choosing a thrust that would send the lander back into orbit. At this point we want the lander's altitude to essentially remain constant and just maneuver laterally to the target point.

Event90 – this is when the command to start maneuvering laterally begins

```
 critr = 'tdurp',
 iguid(1)  = 2, / relative Euler angle
guidance
 iguid(2)  = 1, /separate steer opt for
each channel
 iguid(9) = 0, / carry over yawcp1
 iguid(10)= 1, /input all pc's for pitr
 iguid(11)= 0, /carry over rolpcl
 iguid(13)= 1,
```

Event95 – this event marks arrival at targeted latitude, longitude, altitude, and velocity

```
 event  = 95,
 critr  = 'tdurp',
 endphs = 1,
```

Event500 – end problem

Controls.

The controls for the Hover & Thrust Laterally scenario are set up as the following:

```
Dependent Variable inputs
     depph = 95,95,95,95,95,
     depvr = 'velr','velr','gdalt',
             'gdlat','long',
     ifdeg = 0,0,0,1,1,
     deptl = 0.1,0.1,1.0,
             0.00168818,0.00168818,
     depval= 0.0,2.0,2500.0,
             15.18,264.760,
     idepvr= -1,1,0,0,

Independent Variable inputs
     indph = 1,1,1,
             90,90,90,
             90,95,
     indvr ='yawpc1','yawpc2','pitpc2',
            'critr','yawpc2','pitpc1',
            'pitpc2','critr',

     u     = 180.0,0.0,0.0,
             100.0,0.0,20.0,
             0.0,100.0,
```

The dependent variable velr is targeted twice, once as a minimum bound and second as a maximum bound. The 0.00168818-degree tolerance on the latitude and longitude targets corresponds to 100 meters. That is, we wish the targeting algorithm to set the lander down no farther away than 100 meters from the reference latitude and longitude.

The independent variables are split up such that at event 1 we pick the initial horizontal direction and start thrusting towards it (while keeping altitude constant). Depending on what time the targeting algorithm determines is best (first guess is 100 sec) then at event 90 an update to pitch is chosen such that the lander is braking but still moving in the correct horizontal direction. The constant yaw term is carried over from event 1. Then at event 95 we have reached our desired targets. How long this takes is the variable (tdurp). The first guess is 100 sec. To summarize, at event 1 our horizontal movement begins towards the reference point. At event 90 (about halfway to the reference) we flip the engine so that it is now braking ensuring that by event 95 it sets down on target with a comfortable velocity.

First Guesses.
There was no special algorithm developed to improve first guesses. It was assumed as a first approximation that the target reference point generally was south of the initial point.

Results.

To test out the above scenarios a set of initial conditions was used. This set is a subset of the Monte Carlo analysis done by LaRC for the M2001 mission and includes the ellipse at parachute handoff (inertial coordinates). 235 points out of the full set of 1981 were used. For each case in each scenario a full 235 POST3D runs were made corresponding to the 235 initial parachute handoff conditions.

Thrust on Parachute.
For the first scenario (thrust during parachute) targeting success (out of 235) for each case is shown below:

Table 1 – Targeting Success for Thrust on Parachute

case	description	# targeted
bkb3firstguesses-noopt-nothrust.inp	baseline reverse gravity turn, distance to reference not minimized	231
bkb3firstguesses-nothrust.inp	baseline reverse gravity turn, distance to reference minimized	197
bkb3firstguesses.inp	thrust maneuver on parachute, baseline reverse gravity turn, distance to reference minimized	216

The following graphs show results for the three 'thrusting while on the parachute' cases (including baseline). The red asterisk shows the reference position.

For the baseline case shown first, the initial handoff ellipse magnitude is the same as the final ellipse magnitude (fig. 1). The final ellipse is just moved a bit east and south. We expect this since we are not trying to do anything but slow the lander down to an acceptable speed and reach the ground at 2500 meters. Propellant used is fairly constant at around 210 N (fig. 2).

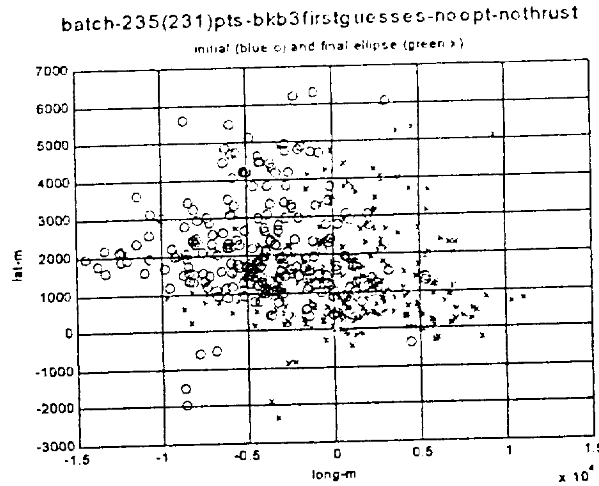


Fig. 1 – Baseline initial and final ellipses

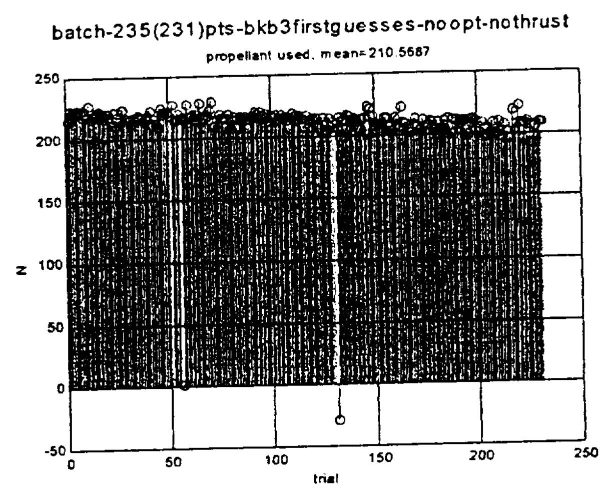


Fig. 2 – Baseline propellant used

The next graphs focus in on just the points that managed to fall within a 1 Km radius of the reference point (the baseline case only had 12 points that met this criteria but since it doesn't try to reach the reference it isn't shown). For the optimized baseline with final ellipse no more than 1 Km away from reference. 28 out of 235 made the cut (fig. 3).

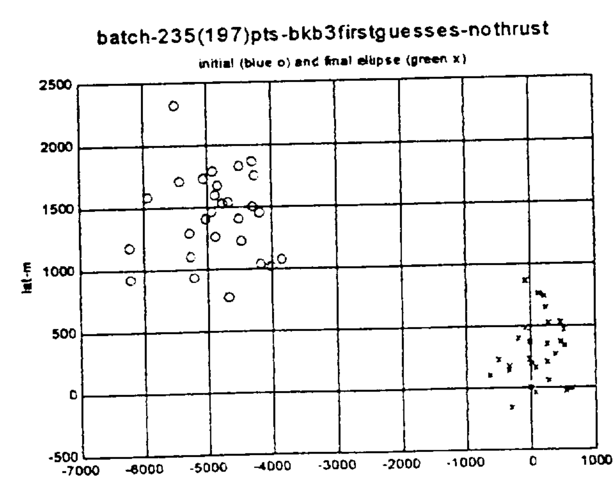


Fig. 3 – Optimized Baseline initial and final ellipses

Notice that the initial ellipse has a radius a bit larger than 1 Km while the final ellipse has a radius no larger than 900 meters. We have managed to shrink the ellipse some but not by much. The fuel usage for this case is very close to the baseline in the mean but fluctuates from point to point (fig. 4).

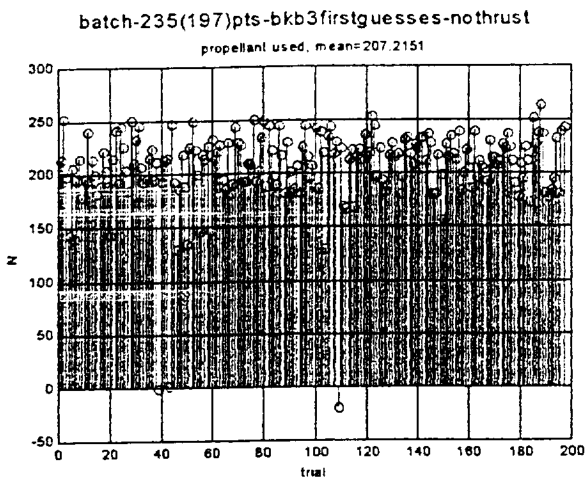


Fig. 4 – Optimized Baseline propellant used

Next is the thrusting on the parachute case, again focused in on only 1Km (61 points made the cut) but all points shown for fuel usage (fig. 5 and fig. 6).

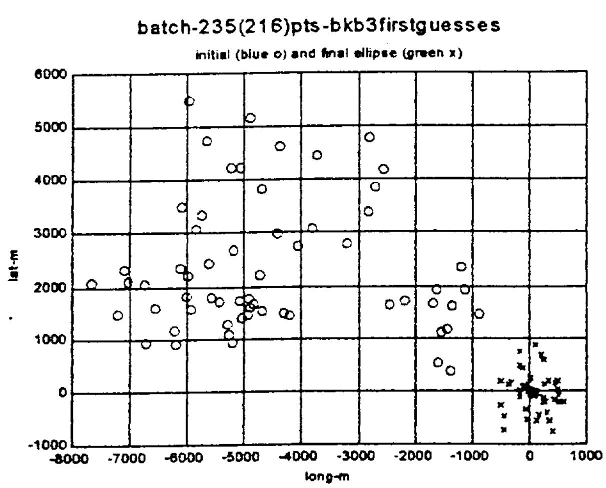


Fig. 5 – Thrust on Parachute initial and final ellipses.

Notice that the initial ellipse is over 4Km in radius yet the final ellipse is within 1Km radius. Clearly a vast improvement but still not as much as we'd like. The tradeoff comes in the form of fuel usage. The mean fuel usage is around 300 N.

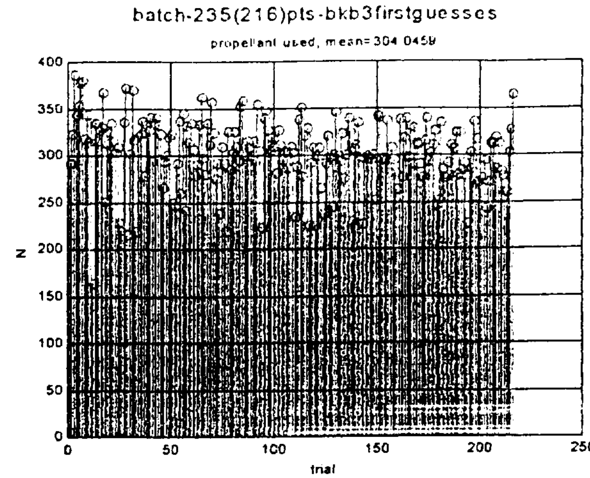


Fig. 6 – Thrust on Parachute propellant usage.

Hover and Thrust Laterally.
These two cases were targeted to within 100 meters of the actual latitude and longitude reference point. The targeting success (out of 235) is shown below and includes the targeting success of the input deck used to gather initial points.

Table 2 - Targeting Success for Hover and Thrust Laterally

case	description	# targeted
targ3000m.inp	reverse gravity turn to a hover condition at 3000m (+500m surface), used only as initial conditions for the hover cases.	234
hover20deg.inp	horizontal thrust & hover to target, reference point is targeted	208
hover30deg.inp	horizontal thrust & hover to target, reference point is targeted	180

It should be noted that these cases did not use an algorithm for choosing first guesses for the targeting algorithm in POST. It is assumed that a first guess algorithm such as that used in the first scenario would yield even better targeting results.

Here is a graph showing the initial handoff ellipse, then the initial hover point ellipse, and last the final ellipse (fig. 7).

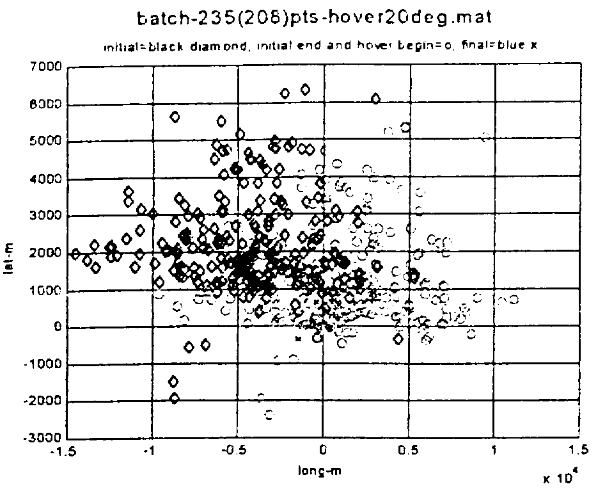


Fig. 7 – Initial to Hover ellipses.

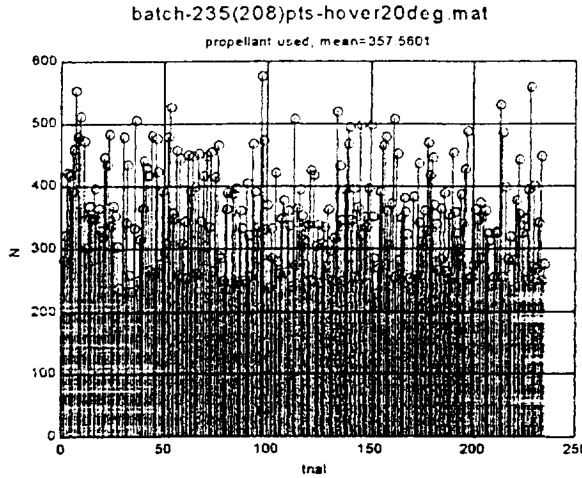


Fig. 9 – 20-degree case propellant used in lateral thrust.

Notice that the final ellipse is so small (centered at 0 reference) that it is hard to see. Most of the points targeted to within 100 meters. Also, the initial parachute handoff ellipse and the initial hover point ellipse are similar in magnitude with an offset of position. This is expected.

Here are graphs of all points targeted and untargeted for both hover cases with fuel usage. First the 20-degree case is shown. Only the final ellipse is shown (fig. 8).

The mean fuel usage was about 350 N for just the hover phase. When added to the reverse gravity turn average (210 N) this brings fuel usage up to 560 N. It can be seen however that a significant number of points were in the 250 N range for hover.

Next are the graphs for the 30-degree case (fig. 10 and fig. 11).

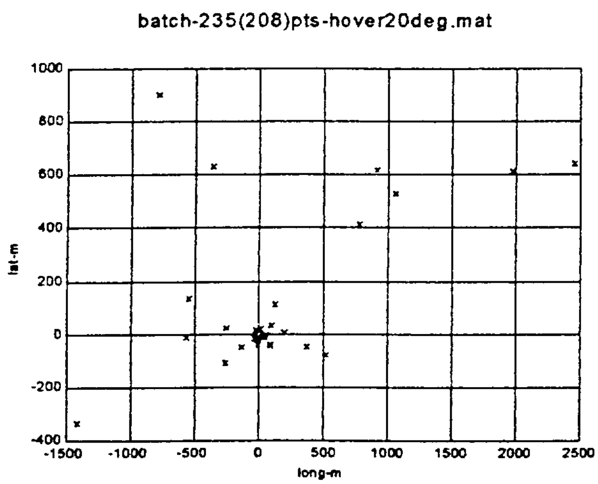


Fig. 8 – 20-degree lateral thrust final ellipse.

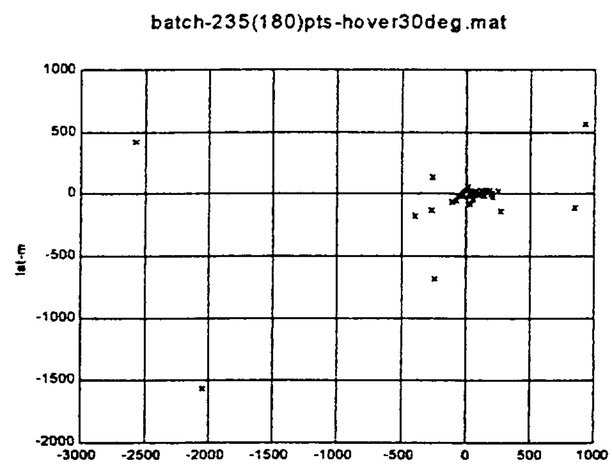


Fig. 10 – 30-degree lateral thrust final ellipse.

Notice that even the points that did not target to within 100m still fell within 2.5Km. Remember the initial ellipse was over 15Km across. The majority of points are so close to the reference that they are indistinguishable from it. This accuracy comes at a fuel cost though (fig. 9).

This is similar to 20-degree case but the fuel usage is a bit different (fig. 11).

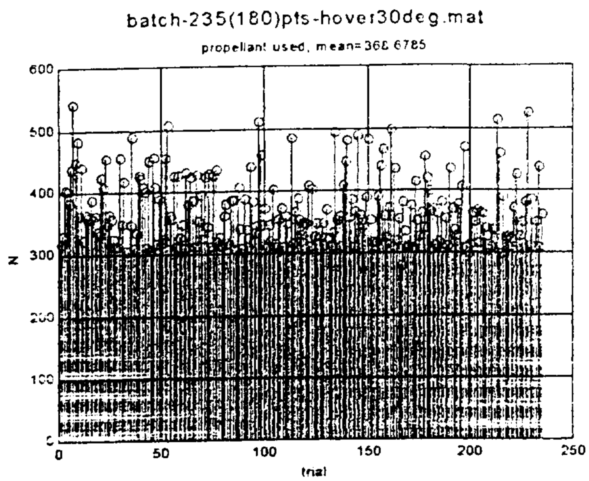


Fig. 11 – 30-degree case propellant used in lateral thrust.

Again, the mean fuel usage is close to the 20-degree case but there are very few cases under 300N. The effect of pitch angle on fuel usage needs to be looked at in depth.

Concluding Remarks.

Thrust on Parachute.

For a simple reverse gravity turn with not much intent to maneuver (baseline) the error at handoff equals error at touchdown. The reference point is in the middle of the final ellipse in terms of longitude but somewhat south in latitude.

For the optimized baseline the control choice appears adequate but does not help much except in some cases of fuel usage, though the average over all trials is similar to the baseline.

For the thrust on the parachute case (maneuver on parachute followed by gravity turn) an improvement in the error at touchdown was evident. This is at the cost of fuel usage, though amounts are within the expected availability.

Hover and Thrust Laterally.

Bringing the lander to a hover 500m above the surface before maneuvering to the reference point is fairly easily accomplished. The amount of horizontal maneuvering impacts heavily into fuel usage. It seems we can target to pinpoint precision as long as we have enough fuel.