International Journal of Mechanical, Industrial and Aerospace Sciences
ISSN: 2517-9950
Vol:11, No:5, 2017

Minimum-Fuel Optimal Trajectory for Reusable First-Stage Rocket Landing Using Particle Swarm Optimization

Kevin Spencer G. Anglim, Zhenyu Zhang, Qingbin Gao

***Abstract*—**Reusable launch vehicles (RLVs) present a more environmentally-friendly approach to accessing space when compared to traditional launch vehicles that are discarded after each flight. This paper studies the recyclable nature of RLVs by presenting a solution method for determining minimum-fuel optimal trajectories using principles from optimal control theory and particle swarm optimization (PSO). This problem is formulated as a minimum-landing error powered descent problem where it is desired to move the RLV from a fixed set of initial conditions to three different sets of terminal conditions. However, unlike other powered descent studies, this paper considers the highly nonlinear effects caused by atmospheric drag, which are often ignored for studies on the Moon or on Mars. Rather than optimizing the controls directly, the throttle control is assumed to be bang-off-bang with a predetermined thrust direction for each phase of flight. The PSO method is verified in a one-dimensional comparison study, and it is then applied to the two-dimensional cases, the results of which are illustrated.

***Keywords*—**Minimum-fuel optimal trajectory, particle swarm optimization, reusable rocket, SpaceX.

I. Introduction

ACCESS to space has traditionally been barred by the high cost of purchasing space on a launch vehicle, which is typically about one hundred million dollars or more for a primary payload. This steep price is partially attributed to the expendable nature of traditional chemical rockets: the entire multi-million-dollar vehicle is discarded after the mission is completed, which is extremely wasteful and inefficient. To bring down the cost per launch, aerospace companies are now looking at making their launch systems partially reusable. One such vehicle is the SpaceX Falcon 9 rocket, which currently has an expendable second stage and an experimental reusable first stage.

While the Falcon 9 is the first of its kind in the rocket industry, the powered descent and landing problem has been around since the Apollo era and has been solved using a multitude of methods for many different scenarios. To give two examples, the Moon landing is one of history's most famous feats, while landing on Mars is a problem that has been of increasing interest in the past decade or so. Ahn et al. [1] present a solution to the two-dimensional lunar landing problem using pseudo-spectral methods and sequential quadratic programming, while Ramanan and Lal [2] use Pontryagin's Maximum Principle and a controlled random search method. Ocampo [3], on the other hand, presents a comparison of several solution methods to the lunar landing problem. Acikmese and Ploen [4] and Carson et al. [5] present two studies of powered descent guidance for a Mars Landing using lossless convexification and formulating the problem as a second-order cone problem. This type of formulation guarantees a convergence on the globally optimal solution, but requires an extensive reformulation of the original problem. For example, Acikmese and Ploen [4] introduce slack variables and use changes in variables to ensure that the cost function, state constraints, and control constraints are always convex.

The motivation for studying this problem is that RLVs still pose a relatively new challenge that is not very well understood yet. There is plenty of literature that solve powered descent trajectories on Mars and on the Moon, as previously mentioned, but few studies have analyzed landing trajectories on Earth. This is likely because Earth's environment is significantly different from that of Mars or the Moon due to the presence of a substantial atmosphere. Atmospheric effects can typically be neglected for studies involving Mars or the Moon because the Martian atmosphere is thin and can often be ignored, whereas the Moon lacks an atmosphere altogether. However, the studies that have used Earth as its setting also tend to disregard the atmospheric effects to avoid the difficulties associated with the equations of motions becoming highly nonlinear when drag and lift become involved. This paper aims to add to the discussion of in-atmosphere optimal landing trajectories on Earth. In these types of problems, it is desired to find a feasible trajectory that satisfies initial and terminal conditions and remains within specified bounds and constraints while minimizing an objective such as the fuel consumed or the terminal condition errors. The numerical optimization methods that are often used to find these trajectories typically fall into two different classifications: indirect or direct. Rao [6], Stryk and Bulirsch [7], and Betts [8] describe the advantages and disadvantages of these two methods in greater detail. However, there exists another class of optimization methods, called derivative-free methods, that are becoming increasingly popular for solving complex

K. S. G. Anglim was a graduate student with California State University Long Beach, Long Beach, CA 90840, USA (e-mail: kevinspencer.anglim@student.csulb.edu).

Z. Zhang is with the Servo Department, Western Digital Corporation, Irvine, CA 92614 USA (e-mail: zhenyu.zhang@wdc.com).

Q. Gao is with the Mechanical and Aerospace Engineering Department, California State University Long Beach, Long Beach, CA, 90840 USA (corresponding author, phone: 475-200-0351; e-mail: qingbin.gao@csulb.edu).

problems such as the ones commonly faced by the aerospace industry. As the name suggests, these methods do not require derivative information that is typically needed for direct and indirect methods. This makes formulating the optimal control problem and generating a numerical solution much easier. One such method that falls under this category, PSO, has been selected as the method of choice for this paper for its ease of implementation and acceptable accuracy. The PSO method was originally proposed by Kennedy and Eberhart in 1995, and is based on the cooperative movement of a flock of flying birds [9]. Birds can learn and coordinate their behavior based upon their individual experience as well as the shared experience of the entire flock. PSO emulates this by iteratively searching the search space and updating each particle's movement based on the history of other particles. These iterations continue until some stopping criterion is met, such as a prescribed error between consecutive iterations or a maximum number of iterations.

This paper is organized as follows. In Section II, the bang-off-bang control profile is explained and justified, followed by an introduction to the expected phases of flight during a typical reusable rocket trajectory. The vehicle and environment parameters are then specified. In Section III, modeling assumption are made, which are then used to define the dynamics equations of the system that move the vehicle from the initial boundary conditions to the terminal boundary conditions, subject to state and control constraints. Then, the minimum-fuel, minimum-landing error cost function is defined. In Section IV, a brief overview of the MATLAB PSO iteration steps is presented, as well as a summary of how this function is used in the author's MATLAB scripts. These scripts are then verified in a one-dimensional study, and the results of this verification are illustrated and can be compared to the same study conducted by Ocampo. In Section V, the three terminal landing cases are simulated, the results of which demonstrate that the solution method is capable of finding minimum-fuel trajectories. Lastly, Section VI presents a few concluding remarks.

II. Problem Statement

The problem of interest for this paper is determining the optimal control history and two-dimensional state trajectories that allow the booster to move from the point of stage separation to the designated terminal conditions and perform a successful landing. The selection of this optimal trajectory will be determined by the optimizing switching times for the thrust throttle control and the total time of flight, thereby minimizing the fuel consumption and terminal condition errors. The switching times and final time are chosen as the decision variables because it has been demonstrated by Kirk [10] that a minimum-fuel trajectory utilizes "bang-off-bang" control. The boostback burn serves the specific purpose of limiting the downrange translation of the rocket, thus the fuel expenditure for this maneuver is essential for targeting the desired landing location. Following this, using Kirk's fuel-optimal strategy, the vehicle will coast with the throttle off until the landing burn is initiated, possibly using atmospheric drag to slow it down slightly. The landing burn is then used to complete a gravity turn maneuver and reduce the vehicle's velocity to zero at the exact moment it reaches zero altitude. Using this switching system, it can be assumed that the throttle is at 100% for the boostback phase, 0% for the coasting phase, and 100% for the landing phase.

Although SpaceX employs the use of a reentry burn to adjust the vehicle's velocity profile and to protect the engines, it has been omitted from this paper due to lack of available information regarding the heating and stress limitations of actual launch vehicles, as well as the complicated aerodynamic interactions between the rocket's exhaust and the atmosphere. Additionally, based on Kirk's "bang-off-bang" requirement, the inclusion of a reentry burn would constitute a non-optimal control for minimizing fuel. This is due to the fact after the reentry burn is completed and some kinetic energy is dissipated, gravity continues to accelerate the vehicle. This reduction and increase in velocity prior to the landing burn constitutes a waste in fuel from a minimum-fuel optimization standpoint.

At the time of writing, the only operational reusable rocket is the SpaceX Falcon 9. Because of this, the Falcon 9 v1.1 launch vehicle specifications will be used for this study [12]. A summary of the vehicle's performance parameters, as well as some system parameters, is detailed in Table I.

TABLE I
Vehicle and Model Parameters

Vehicle Parameter	Value
Specific Impulse, I_{SP} (s)	282
Liftoff Thrust, T (kN)	5,886
Drag Coefficient, C_D	0.75
Diameter, d (m)	3.66
Gravity, g_0 (m/s^2)	9.80665
Air Density at Sea Level, ρ_0 (kg/m^3)	76,501
Scale Height, h_0 (m)	7,500

The hypothetical mission will be to insert a 5,000-kg payload into orbit using a gravity turn trajectory, which would require an 89.92° kick maneuver at 400 m altitude. The equations of motion used in this launch simulation are described by Curtis [13]. This launch phase is not optimized and is only used to determine practical values for the position and velocity conditions at stage separation, and it is assumed that the second stage of the rocket can complete the primary mission. The launch simulation also assumes that 10% of the first-stage fuel available at launch is reserved for the boostback and landing burns. If desired, the launch phase and amount of reserved propellant can potentially be optimized as well, but their optimization depends on the primary mission objective, i.e. the mass of the payload and the desired orbit, and the desired first-stage landing location. The states at stage separation, now being used as the initial conditions, are included with the boundary conditions in Table II.

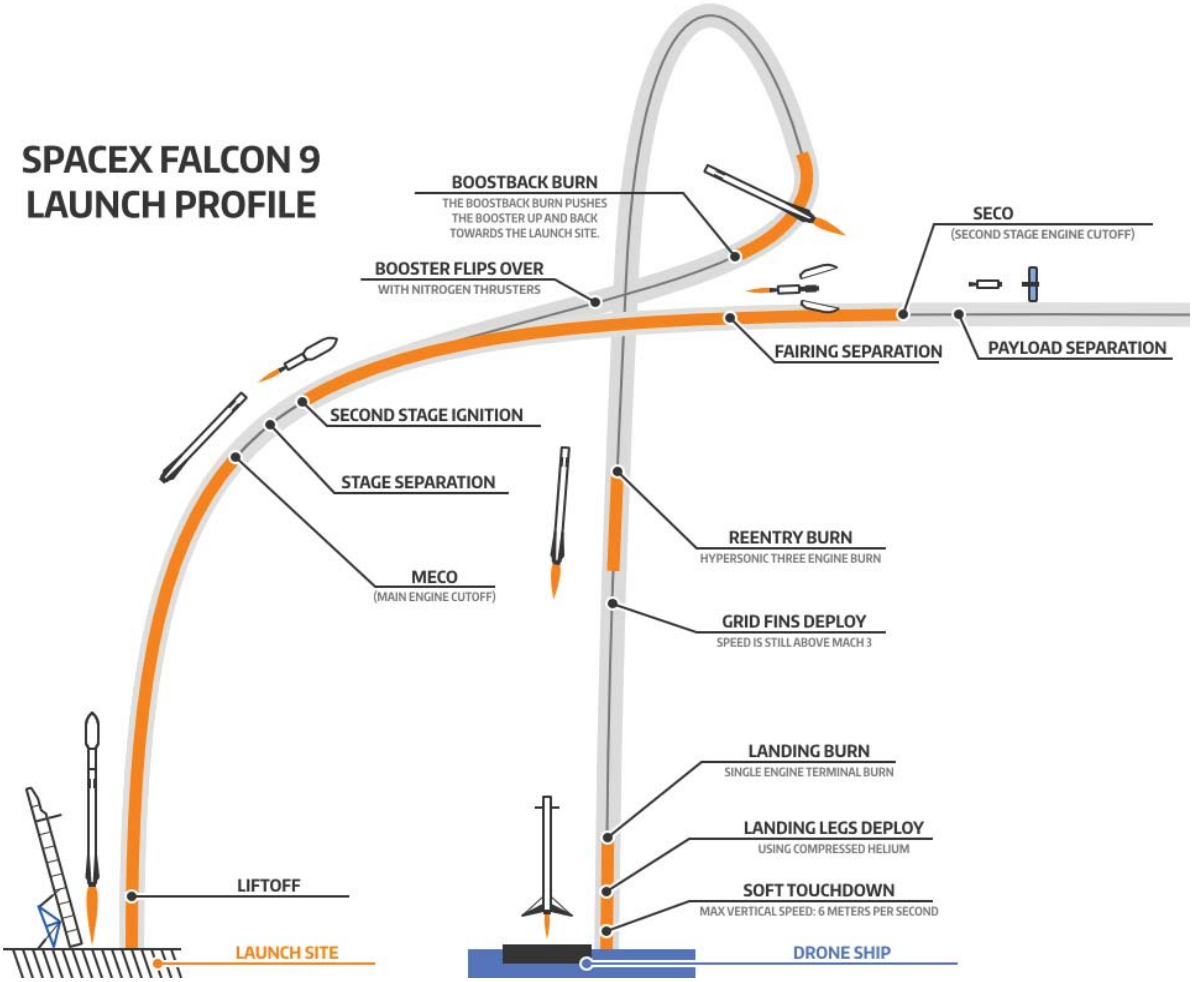

Fig. 1 Typical mission profile for Falcon 9 reusable rocket [11]

III. Optimization Modeling

An accurate system model is required to properly optimize the landing trajectory. This begins with some simplifying assumptions, which are then used to build the dynamic equations that described the motion of the rocket. As previously mentioned, when dealing with an in-atmosphere analysis, the drag terms in these equations are highly nonlinear, which typically makes them difficult to work with. This is due to drag being quadratically dependent on the vehicle's velocity, as well as exponentially dependent on the vehicle's altitude.

A. Modeling Assumptions

To begin, several assumptions are made to simplify the problem and make finding a solution more feasible.

- A point-mass dynamical model, with specified thrust directions during the boostback and landing phases, is used.
- Drag effects are considered in the equations of motion, but lift effects are neglected due to the gravity turn trajectory producing a zero angle of attack.
- The flight path is constrained to the two-dimensional plane, where altitude is the positive y-axis and downrange is positive the x-axis, over a non-rotating flat earth and constant gravitational field.
- The throttle control system is perfect and experiences zero-lag and zero-error during flight.
- The maximum thrust is assumed to be constant and independent of atmospheric pressure.

Although these assumptions should not be applied to real-world simulations, the resulting simulation and solution can still be used to gain a better understanding of the system's behavior and can be used as a stepping stone for more accurate models.

B. Dynamic Model

The forces acting on the rocket and the two-dimensional Cartesian coordinate system that the rocket resides in are illustrated in Fig. 2. The thrust force T has a constant magnitude, as previously mentioned, and points at an angle $\theta(t)$ measured counterclockwise from the positive x-direction. The velocity $v(t)$ of the rocket points along the flight path angle $\gamma(t)$ measured counterclockwise from the positive x-

direction. The drag force $D(t)$ points opposite to the velocity vector and has a magnitude that is nonlinearly dependent on altitude $y(t)$ and velocity $v(t)$, defined by

$$D(t) = \frac{1}{2}\rho_0 e^{-\frac{y(t)}{h_0}} C_D A v(t)^2 \quad (1)$$

where $\rho_0 = 1.225\frac{kg}{m^3}$ is the atmospheric density at sea level, $h_0 = 7{,}500\,m$ is the scale height, $C_D = 0.75$ is the drag coefficient, and A is the circular cross sectional area defined by a radius of 3.66 meters. The weight $W(t)$ of the rocket always points in the negative y-direction and has a magnitude dependent on the variable mass $m(t)$ of the vehicle and Earth's gravity constant $g_0 = 9.80665\frac{m}{s^2}$.

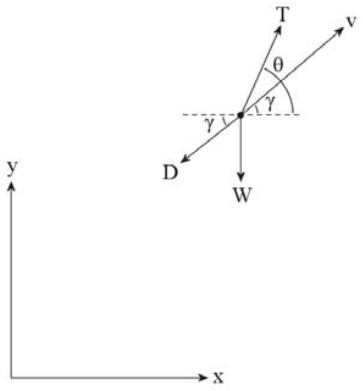

Fig. 2 Force-body diagram and Cartesian coordinate system

Using Newton's second law, the thrust, drag, and weight forces can be resolved into components in the x- and y-directions:

$$\begin{aligned} m(t)a_x(t) &= k(t)T\cos\theta(t) - D(t)\cos\gamma(t) \\ m(t)a_y(t) &= k(t)T\sin\theta(t) - D(t)\sin\gamma(t) - g_0. \end{aligned} \quad (2)$$

Replacing the acceleration terms in (2) with the rates of change of the velocities in the x-direction $\dot{v_x}(t)$ and y-direction $\dot{v_y}(t)$, the velocity dynamic equations can be defined as:

$$\begin{aligned} \dot{v_x}(t) &= \tfrac{1}{m(t)}(k(t)T\cos\theta(t) - D(t)\cos\gamma(t)) \\ \dot{v_y}(t) &= \frac{1}{m(t)}(k(t)T\sin\theta(t) - D(t)\sin\gamma(t)) - g_0. \end{aligned} \quad (3)$$

The dynamics of the downrange distance $x(t)$ and the altitude $y(t)$ are defined as:

$$\begin{aligned} \dot{x}(t) &= v_x \\ \dot{y}(t) &= v_y. \end{aligned} \quad (4)$$

Lastly, the mass flow rate of the rocket is defined by:

$$\dot{m}(t) = -\tfrac{k(t)T}{I_{SP}g_0} \quad (5)$$

where $k(t)$ is the bounded throttle and I_{SP} is the vehicle's specific impulse, which is a measure of the rocket engine's performance.

From observation, the constants in the drag term can be consolidated into a new constant β for convenience. Additionally, the terms that include flight path angle $\gamma(t)$ can be expressed in terms of the x and y components of velocity, thus removing an extra variable. Using this, these terms can then be described as:

$$\beta = \frac{1}{2}\rho_0 C_D A$$

$$v(t)^2 = v_x(t)^2 + v_y(t)^2$$

$$\cos\gamma(t) = \tfrac{v_x(t)}{\sqrt{v_x(t)^2 + v_y(t)^2}} \quad (6)$$

$$\sin\gamma(t) = \frac{v_y(t)}{\sqrt{v_x(t)^2 + v_y(t)^2}}.$$

Applying (6) to (3)-(5), the equations of motion that describe the dynamics of the reusable rocket are presented as:

$$\dot{x}(t) = v_x$$

$$\dot{y}(t) = v_y$$

$$\dot{v_x}(t) = \tfrac{1}{m(t)}(k(t)T\cos\theta(t) - D(t)\cos\gamma(t)) \quad (7)$$

$$\dot{v_y}(t) = \frac{1}{m(t)}(k(t)T\sin\theta(t) - D(t)\sin\gamma(t)) - g_0$$

$$\dot{m}(t) = -\frac{k(t)T}{I_{SP}g_0}.$$

The set in (7) is used during the boostback phase because it allows for a thrust direction $\theta(t)$ to be specified. During this phase, the thrust direction is set to 180° since only the downrange translation of the vehicle must be limited, while the "upwards" translation is inconsequential. The vehicle's state is represented by the five-element vector $[x(t), y(t), v_x(t), v_y(t), m(t)]$, which is composed of the downrange distance, altitude, horizontal velocity, vertical velocity, and mass, respectively. The throttle control $k(t)$, as previously mentioned, is equal to 1 during the boostback phase. Additionally, the thrust level T is equal to one-third of the total lift-off thrust, simulating the use of only three of the nine engines for the boostback burn.

Alternatively, it is desired to use a different set of equations for the coasting phase and the landing phase. These equations reflect a modified version of the gravity turn maneuver described by Curtis, and can be expressed as:

$$\dot{x}(t) = v_x$$

$$\dot{y}(t) = v_y$$

$$\dot{v}(t) = \frac{1}{m(t)}\left(-k(t)T - \beta e^{-\frac{y(t)}{h_0}} v(t)^2\right) - g_0 \sin\big(\gamma(t)\big) \quad (8)$$

$$\dot{\gamma}(t) = -\frac{1}{v(t)}\left(g_0 - \frac{v(t)^2}{R_e + y(t)}\right)\cos\big(\gamma(t)\big)$$

$$\dot{m}(t) = -\frac{k(t)T}{I_{SP} g_0}.$$

where $R_e = 6{,}378\ km$ is the radius of the Earth. The only difference between Curtis' equations and the equations presented here is the negative sign on the thrust term, which forces the thrust vector to always point opposite to the velocity vector. This maneuver is fuel-optimal due to all the energy being used to slow the vehicle rather than being used for steering; steering is achieved passively by using gravity, hence the name "gravity turn".

The vehicle states for these phases is represented by the five-element vector $[x(t), y(t), v(t), \gamma(t), m(t)]$, which is composed of the downrange distance, altitude, velocity, flightpath angle, and mass, respectively. The throttle control $k(t)$ is equal to 0 during the coasting phase, and is then equal to 1 during the landing phase. The thrust level T for the landing phase is equal to one-ninth of the total liftoff thrust, simulating a one-engine landing burn.

C. Mission Boundary Conditions

The objective of the mission is the bring the launch vehicle from its initial conditions at stage separation, as previously mentioned, to three different sets of terminal conditions, which are detailed Table II. The first landing case represents the desire to simply land anywhere downrange, so no downrange terminal condition has been specified. The second landing case represents the desire to land at a specified location downrange, such as on a barge out at sea. Lastly, the third landing case represents the desire to land back at the launch site. These three cases were chosen for being possible objectives for companies seeking to use reusable rockets.

TABLE II
BOUNDARY CONDITIONS

State	Fixed Initial	Case 1 Terminal	Case 2 Terminal	Case 3 Terminal
Downrange (km)	36.022	*Free*	200	0
Altitude (km)	60.708	0	0	0
X-Velocity [km/s]	1.052	0	0	0
Y-Velocity [km/s]	1.060	0	0	0
Mass [kg]	76,501	*Free*	*Free*	*Free*

D. Constraints

For the system to behave accurately, constraints must be placed. In this case, the altitude must remain greater than or equal to zero to ensure the vehicle remains above ground, the mass must remain greater than or equal to the structural mass to ensure that fuel is still available throughout the flight, and the throttle must be bounded to zero or one.

$$y \geq 0\ m$$

$$m \geq 25{,}600\ kg \quad (9)$$

$$k = 0\ or\ k = 1$$

Due to the formulation of the optimal control problem using bang-bang control, the throttle constraint will automatically be met. However, the state constraints cannot be explicitly met by the PSO method. Even so, the constraints on mass and altitude will still be checked in the final solution to determine whether a feasible trajectory has been found.

E. Objective Function

Properly formulating the cost function is crucial to having the optimization solver provide an accurate solution. It can be stated that minimizing the fuel consumption over the entire flight is equivalent to maximizing the terminal mass $m(t_f)$ at the terminal time t_f. This is done by having a negative sign with the mass term; the more mass that remains at touchdown, the more negative the value is, thus minimizing the cost function value. Also, since the problem is formulated in an initial value problem framework, minimizing the error between the states at the terminal time $x(t_f)$, $y(t_f)$, $v_x(t_f)$, and $v_y(t_f)$ and the desired terminal conditions is required for a feasible solution. To penalize the errors, their values are positive, thus increasing the cost function value rather than minimizing it. This can be expressed as:

$$\min J = -m(t_f) + s_1\left(x_f - x(t_f)\right)^2 + s_2\left(y(t_f)\right)^2 + s_3\left(v_x(t_f)\right)^2 + s_4\left(v_y(t_f)\right)^2. \quad (10)$$

For the terminal downrange distance error $x_f - x(t_f)$, the term will be omitted completely for the unspecified downrange landing. Otherwise, x_f will be equal to 200 km for the specified downrange landing, and 0 km for the return-to-launch-site landing. The terminal altitude error term, as well as the terminal velocity terms, does not include a desired terminal condition because the desired terminal condition will always be zero at touchdown. If the altitude at the terminal time is not zero, then the landing is not successful. Likewise, if the terminal velocities are not zero, then the landing is unsuccessful. These error terms are squared since both positive and negative deviations are equally undesirable.

The s terms are constant weighting factors that must be individual tuned by the user to achieve the desired optimization behavior. In this case, the weighting factors should be selected to force the terminal error terms to be of the same magnitude as the terminal mass or larger, otherwise the algorithm will prioritize the remaining fuel rather than achieving the desired terminal position and velocity.

IV. PSO PROCEDURE

To begin the optimization process, the MATLAB particle swarm function generates the initial population with random

positions and velocities and calculates the cost function for each particle [14]. In this case, the position of each particle is a decision vector that contains the switching time(s) and the terminal time. The function then iteratively updates the randomly-generated swarm population using the steps outlined in this section. At an arbitrary iteration n for a particle i, which is at position $x(i)$:

- Choose a random subset of S of N particles other than particle i.
- Find $fbest(S)$, the best overall objective function value of all neighbors, and $g(S)$, the best position of the particle with the best objective function.
- Update the velocity:

$$v(i) = Wv(i-1) + y_1u_1\big(p(i) - x(i-1)\big) + y_2u_2\big(g - x(i-1)\big) \quad (11)$$

where $Wv(i-1)$ is the inertia component, $y_1u_1(p - x(i-1))$ is the cognitive component, and $y_2u_2(g - x(i-1))$ is the social component. The variable W is the inertia weighting factor, the variable y_1 is the self-adjustment weighting factor, and the variable y_2 is the social-adjustment weighting factor. The vectors u_1 and u_2 are random vectors whose entries are chosen from a uniform distribution between 0 and 1. Lastly, p is the best solution found by particle i, whereas g is the best solution found so far by the current neighborhood.

- Update the position:

$$x(i) = x(i-1) + v(i) \quad (12)$$

using the velocity found in the previous step.

- Enforce the bounds. If any component of x violates a bound, then set it equal to that bound.
- Evaluate the cost function $f = fun(x)$.
- If $f < fun(p)$, then set $p = x$. This ensures that p contains the best solution that the particle has seen so far.
- If $f < fun(b)$, then set $b = f$ and $d = x$. This ensures that b contains the best cost function value found by the swarm, and d has the best solution.
- If the best function value was lowered in the previous step, then set $flag = true$. Otherwise, set $flag = false$.
- Update the neighborhood of particles. If $flag = true$:

a. Set $c = max(0, c-1)$.
b. Set N to $minNeighborhoodSize$.
c. If $c < 2$, then $W = 2W$.
d. If $c > 5$, then $W = W/2$.

If $flag = false$:

a. Set $c = c + 1$.
b. Set $N = min(N + minNeighborhoodSize, SwarmSize)$.

The variable c is the stall counter and initially starts equal to zero when MATLAB initiates the function. The variable N is initially equal to the $minNeighborhoodSize$, which is initiated by MATLAB by default. Expanding on the particle velocity update in (11), the movement of the population of solution candidates is based the three components previously mentioned. The inertia component represents the influence of the particle's velocity from the previous iteration, the cognitive component represents the tendency for the particle to move towards its best remembered solution, and the social component represents the tendency for the particle to move towards the best solution found so far by the entire swarm.

Although the use of PSO does not necessarily guarantee convergence on the globally optimal solution as convex optimization does, the global optimal can still be achieved by having a large swarm size to sample the entire search space. The drawback to this is that it can become computationally expensive if the search space is very large, and the number of particles is very high.

A. Program Procedure

Incorporating the vehicle and model parameters in Table I and the initial conditions in Table II, the main MATLAB script calls upon the particle swarm function to evaluate the objective function script for the initial swarm population. The objective function script takes each of the decision vectors of switching time(s) and terminal time, uses the MATLAB $ode45$ numerical integrator function to simulate the trajectory using (7) and (8), and evaluates the cost function in (10) using the terminal mass and terminal state errors. This process continues iteratively using the steps in the previous section until one of MATLAB's default stopping criteria is met.

B. PSO Validation

To test the validity of the PSO method and written MATLAB scripts, the optimization procedure has been applied to the solved one-dimensional lunar lander problem presented by Ocampo [3]. In her paper, Ocampo wishes to find the minimum-fuel landing trajectory for a spacecraft initially having a mass $m(t) = 224\,kg$, 204 kg of which is propellant. The spacecraft is also initially at an altitude $h(t) = 100{,}000\,m$ and moving upwards at a velocity $v = 100\,m/s$. The vehicle has a maximum thrust $T = 400\,N$, a mass flow rate $\dot{m} = 1.1332\,kg/s$, and is only affected by the Moon's gravity $g = 1.622\,m/s^2$. The equations of motion that describe the dynamics of the spacecraft are defined as:

$$\dot{h}(t) = v$$

$$\dot{v}(t) = -g + \frac{Tu(t)}{m(t)} \quad (13)$$

$$\dot{m}(t) = -ku(t)$$

Ocampo determines that the minimum-fuel optimal control history follows a bang-bang control structure, and then determines the switching time and the final time using three

different methods: explicitly, Newton's shooting method, and finite difference method using MATLAB's *bvp4c* function.

Applying the PSO scripts, the results in Figs. 3-7 were obtained. Fig. 3 illustrates the control history achieved by the switching time of 312.853 seconds and the total time of flight of 481.858 seconds found by the PSO method. The blue portion of the plot represents when the throttle is off, and the red portion represents when the throttle is on. Fig. 4 illustrates the altitude history, while Fig. 5 illustrates the velocity history. During the unpowered phase, the altitude follows a ballistic trajectory as the velocity linearly decreases under the effects of constant gravity. Once the switching time occurs and the throttle turns on, the velocity follows a nonlinear trajectory to zero, which correlates to the nonlinear altitude trajectory reaching zero as well. Fig. 6 illustrates that the mass of the vehicle remains constant while the throttle is off, as expected, but begins to decrease linear when the throttle is turned on. Lastly, Fig. 7 represents the altitude versus velocity trajectory of the vehicle, where the velocity becomes more negative as the altitude decreases, but then both states reach zero after the switching time.

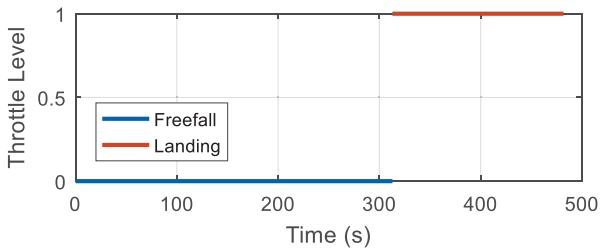

Fig. 3 Bang-bang throttle control history from optimizing switching time and terminal time

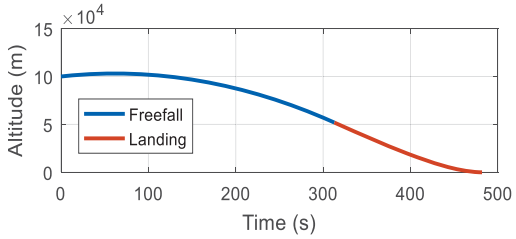

Fig. 4 Altitude history from optimizing switching time and terminal time

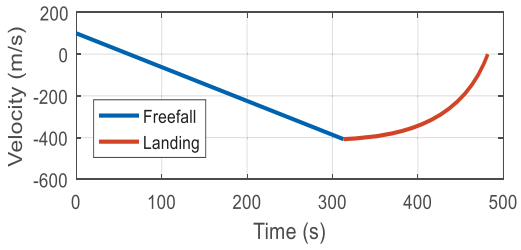

Fig. 5 Velocity history from optimizing switching time and terminal time

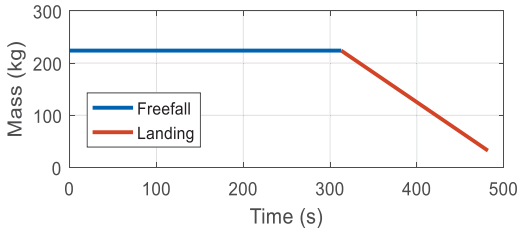

Fig. 6 Mass history from optimizing switching time and terminal time

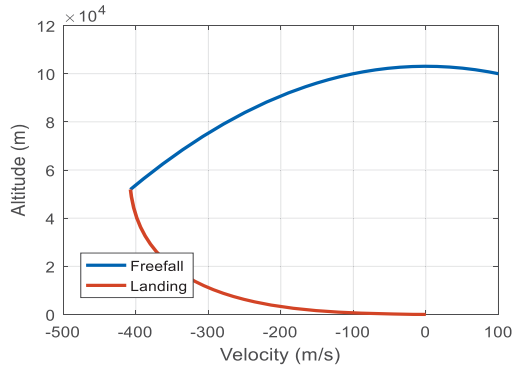

Fig. 7 Altitude versus velocity from optimizing switching time and terminal time

These results are comparable to the results obtained by Ocampo, which proves the viability of the PSO method.

V. Results and Discussion

With the PSO method verified, it can now be used to minimize the cost function in (10), subject to the dynamic equations of (7) and (8). The switching times and terminal times found by the PSO method will determine the two-dimensional in-atmosphere landing trajectories for three separate cases. This first case is the unspecified downrange landing, where it is desired to simply land anywhere downrange. Second, a specified landing will be made, where it is desired to land at a specific location downrange. Lastly, a return-to-launch-site will be analyzed, where the rocket must halt and reverse its downrange translation to return to the launch pad. The results of these cases will now be presented.

A. Unspecified Downrange Landing

The simulation for the unspecified downrange landing case will be presented first. This case was chosen to act as a proof-of-concept for applying the PSO algorithm to a two-dimensional case. Similar to the lunar lander problem, only two decision variables exist: the switching time and the terminal time. The switching time, total time of flight, and terminal states of this simulation are summarized in Table III.

As demonstrated, the terminal altitude and velocities are within a negligible error of zero, and can thus be said to have reached the desired terminal state. The mass also satisfies the mass constraint by being greater than 25,600 kg. The bang-bang throttle control history is illustrated in Fig. 8, where the dotted line represents the coasting phase when the throttle is

off, and the blue line represents the landing phase when the throttle is on. Figs. 9-11 illustrate the mass history, velocity history, and two-dimensional altitude versus downrange distance trajectory, respectively.

TABLE III
UNSPECIFIED DOWNRANGE LANDING RESULTS

Variable	Value
Boostback Burn Length (s)	0
Landing Burn Start Time (s)	159.6261
Time of Flight (s)	350.7243
Final Downrange Distance (m)	280,401.6715
Final Altitude (m)	-4.2007×10^{-5}
Final Horizontal Velocity (m/s)	-8.5879×10^{-7}
Final Vertical Velocity (m/s)	-0.0021327
Final Mass (kg)	31,309.0949

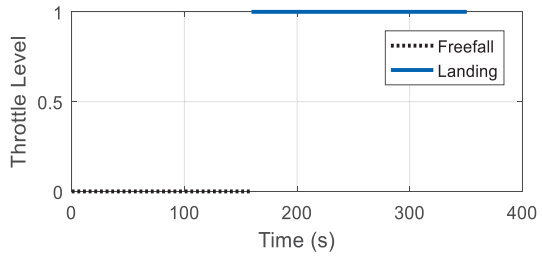

Fig. 8 Throttle control history for the unspecified downrange landing

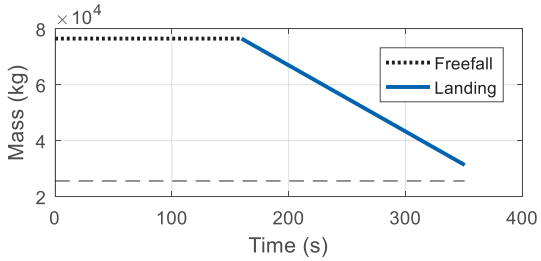

Fig. 9 Mass history for the unspecified landing

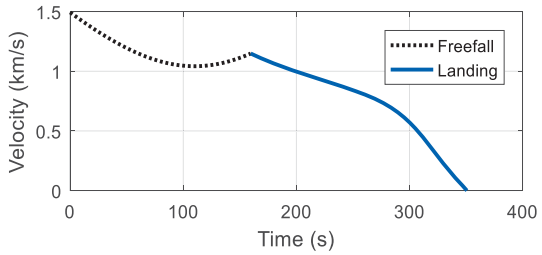

Fig. 10 Velocity history for the unspecified downrange landing

B. Specified 200 km Downrange Landing

Next, the simulation for the 200-km downrange landing case will be presented. For this case, a boostback burn has been added to the previous case. The 200-km location has been selected for only requiring a minor boostback burn. The results of this simulation are summarized in Table IV.

TABLE IV
200 KM DOWNRANGE LANDING RESULTS

Variable	Value
Boostback Burn Length (s)	14.0972
Landing Burn Start Time (s)	185.2656
Time of Flight (s)	336.6412
Final Downrange Distance (m)	200,000
Final Altitude (m)	-2.4906×10^{-6}
Final Horizontal Velocity (m/s)	-5.4074×10^{-9}
Final Vertical Velocity (m/s)	-0.00010518
Final Mass (kg)	30,701.5304

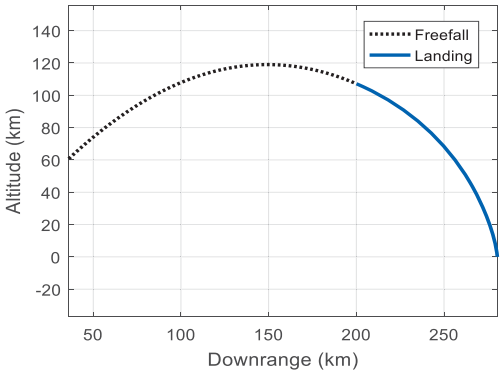

Fig. 11 Altitude versus downrange distance for the unspecified downrange landing

As demonstrated, the terminal altitude and velocities are again within a negligible error of zero, and can thus be said to have reached the desired terminal state. Similarly, the desired downrange distance of 200 km has been hit exactly. The mass once again satisfies the mass constraint by being greater than 25,600 kg. The control history is illustrated in Fig. 12 where the blue line represents the boostback burn when the throttle is on, dotted line represents the coasting phase when the throttle is off, and the blue line represents the landing phase when the throttle is on. Figs. 13-15 illustrate the mass history, velocity history, and two-dimensional altitude versus downrange distance trajectory, respectively.

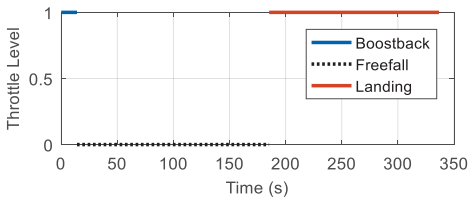

Fig. 12 Throttle control history for the 200-km downrange landing

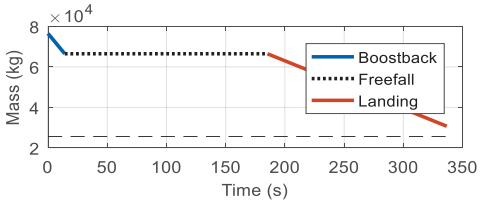

Fig. 13 Mass history for the 200-km downrange landing

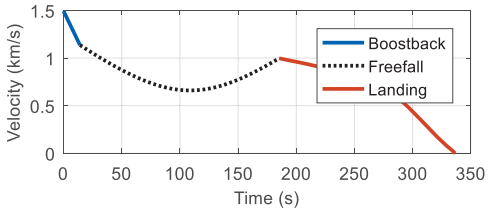

Fig. 14 Velocity history for the 200-km downrange landing

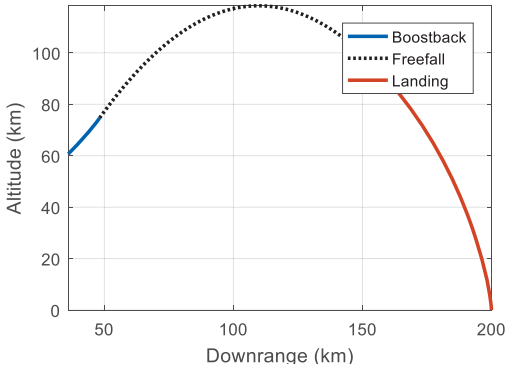

Fig. 15 Altitude versus downrange distance for the 200-km downrange landing

C. Return-to-Launch-Site Landing

Lastly, the simulation for the return-to-launch-site landing case will be presented. For this case, a significantly longer boostback burn than the previous case is required to complete the return. This location has been selected for being a potential landing destination for companies in the future. The results of this simulation are summarized in Table V.

TABLE V
RETURN-TO-LAUNCH-SITE LANDING RESULTS

Variable	Value
Boostback Burn Length (s)	40.5684
Landing Burn Start Time (s)	216.9112
Time of Flight (s)	307.8189
Final Downrange Distance (m)	2.5571×10^{-6}
Final Altitude (m)	-4.5998×10^{-6}
Final Horizontal Velocity (m/s)	-2.6498×10^{-8}
Final Vertical Velocity (m/s)	-7.6642×10^{-5}
Final Mass (kg)	26,221.1488

As demonstrated, the terminal position and velocities are all essentially zero, and can thus be said to have reached the desired terminal state. The final mass satisfies the mass constraint by being greater than 25,600 kg, but with an extremely small fuel margin remaining. The control history is illustrated in Fig. 16, where the blue line represents the boostback burn when the throttle is on, dotted line represents the coasting phase when the throttle is off, and the blue line represents the landing phase when the throttle is on. Figs. 17-19 illustrate the mass history, velocity history, and two-dimensional altitude versus downrange distance trajectory, respectively.

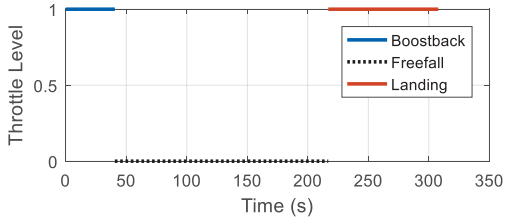

Fig. 16 Throttle control history for the return-to-launch-site landing

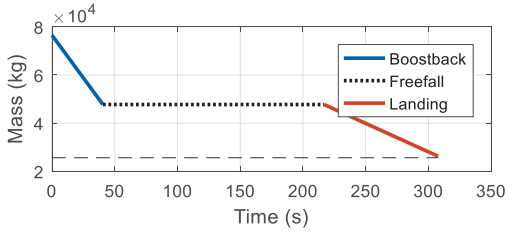

Fig. 17 Mass history for the return-to-launch-site landing

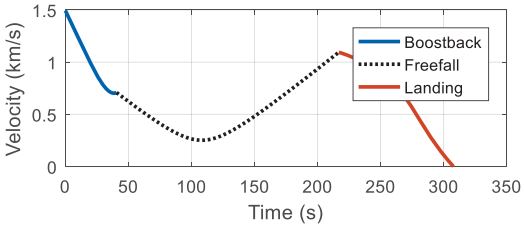

Fig. 18 Velocity history for the return-to-launch-site landing

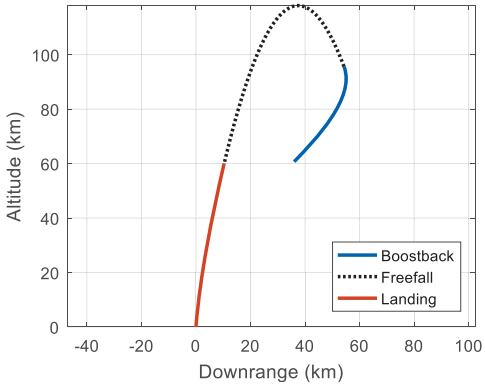

Fig. 19 Altitude versus downrange distance for the return-to-launch-site landing

VI. Conclusion

A successful optimization method for finding the minimum-fuel reusable rocket landing trajectories in three different cases has been presented. The method uses PSO to determine the switching time(s) and terminal times for the throttle control. During the boostback phase (if one is used), the thrust points in the negative x-direction in order to limit the downrange translation. During the landing phase, the vehicle uses a gravity turn maneuver where the thrust always points opposite to the velocity vector. The trajectories that result from these decision variables satisfy the two-point boundary value problems in an initial value problem framework, where the terminal state errors are minimized using a cost function. Additionally, the terminal mass is maximized in the cost function.

The use of PSO is advantageous because of its ability to converge on accurate solution in relatively large search spaces despite having no prior knowledge or approximation of what the optimal solution might be. Having an accurate initial guess is typically required for other optimization methods, which can be a difficult task to accomplish. Additionally, the absence of derivative information, such as the costate equations or the Hamiltonian equation, is beneficial in easily setting up the optimization problem.

While this paper can potentially provide useful insights into the behavior of reusable rocket landings, future work can greatly improve the accuracy results of this study by removing some, or all, of the simplifying assumptions. Changing the coordinate system from a two-dimensional plane using a flat, non-rotating Earth to a three-dimensional system using a circular, rotating Earth would only require a change in the equations of motion, which can be done relatively easily. Incorporating rigid-body dynamics would also be beneficial, but would require significantly more work.

Acknowledgment

K. S. G. Anglim would like to thank Dr. Eric Besnard, a professor at California State University Long Beach, for introducing him to the reusable rocket landing problem and for guiding him through the initial research and problem formulation.

References

[1] J. Ahn, B. Park, and M. Tahk, "Two-dimensional Trajectory Optimization of a Soft Lunar Landing from a Parking Orbit Considering a Landing Site," IFAC Proceedings Volumes, 43(15), 178-183, doi:10.3182/20100906-5-jp-2022.00031, 2010.

[2] V. Ramanan, and M. Lal, "Analysis of Optimal Strategies for Soft Landing on the Moon from Lunar Parking Orbit," Journal of Earth System Science J Earth Syst Sci, 114(6), 807-813, doi:10.1007/bf02715967, 2005.

[3] L. P. Ocampo, "Solving the Optimization Control Problem for Lunar Soft Landing Using Minimization Technique," M.S. thesis, Dept. Math., Univ. of Texas at Arlington, Arlington, TX, 2013.

[4] B. Acikmese, and S. R. Ploen, "Convex Programming Approach to Powered Descent Guidance for Mars Landing," Journal of Guidance, Control, and Dynamics,30(5), 1353-1366, doi:10.2514/1.27553, 2007.

[5] J. M. Carson, B. Acikmese, and L. Blackmore, "Lossless Convexification of Powered-Descent Guidance with Non-Convex Thrust Bound and Pointing Constraints," Proceedings of the 2011 American Control Conference, doi:10.1109/acc.2011.5990959, 2011.

[6] A. V. Rao, "A Survey of Numerical Methods for Optimal Control," AAS/AIAA Astrodynamics Specialist Conference, AAS Paper 09-334, 2009.

[7] O. V. Stryk, and R. Bulirsch, "Direct and Indirect Methods for Trajectory Optimization," Annals of Operations Research. 37(1), 357-373, 1992.

[8] J. T. Betts, "Survey of Numerical Methods for Trajectory Optimization," Journal of Guidance, Control, and Dynamics, 21(2), 193-207, doi:10.2514/2.4231, 1998.

[9] R. Eberhart, and J. Kennedy, "A New Optimizer Using Particle Swarm Theory," MHS'95, Proceedings of the Sixth International Symposium on Micro Machine and Human Science. doi:10.1109/mhs.1995.494215, 1995.

[10] D. E. Kirk, Optimal Control Theory: An Introduction. Mineola, NY: Dover, 2004.

[11] J. Gardi, and J. Ross, "The Future of Space Launch is Near: An Illustrated Guide to SpaceX's Launch Vehicle Reusability Plans," retrieved from http://justatinker.com/Future/, June 1, 2016.

[12] Falcon 9 v1.1 & F9R Launch Vehicle Overview," retrieved from http://spaceflight101.com/spacerockets/falcon-9-v1-1-f9r/, June 1, 2016.

[13] H. D. Curtis, Orbital Mechanics for Engineering Students, Second Edition. Oxford, UK: Elsevier, 2010.

[14] "Particle Swarm Optimization," retrieved from http://www.mathworks.com/help/gads/particleswarm.html, June 1, 2016.