THE PENNSYLVANIA STATE UNIVERSITY
SCHREYER HONORS COLLEGE

DEPARTMENT OF AEROSPACE ENGINEERING

APPLICATION OF PARTICLE SWARM OPTIMIZATION TO LYAPUNOV PERIODIC ORBITS

CORY C. CALDWELL
SPRING 2016

A thesis
submitted in partial fulfillment
of the requirements
for a baccalaureate degree
in Aerospace Engineering
with honors in Aerospace Engineering

Reviewed and approved* by the following:

Robert G. Melton
Professor of Aerospace Engineering
Director of Undergraduate Studies
Thesis Supervisor

David B. Spencer
Professor of Aerospace Engineering
Faculty Reader

George A. Lesieutre
Professor and Head of the Department of Aerospace Engineering
Faculty Reader

* Signatures are on file in the Schreyer Honors College.

ABSTRACT

The particle swarm optimization (PSO) technique is a stochastic population-based method motivated by the unpredictable behavior of bird flocks searching for food. This method utilizes information sharing within the population to influence the swarm in finding the optimal solution for the unknown parameters of the problem being considered. PSO has various applications; this research applies the optimization method to space trajectories. Specifically, PSO is applied to find the unknown parameters for the Lyapunov periodic orbits around the collinear interior and exterior Lagrange points in the context of the Circular Restricted Three-Body Problem (CR3BP) in the Earth-Moon system. In this problem, the unknown parameters are represented by the initial position and period which define these orbits. The research starts with application of PSO to the interior Lagrange point of the Earth-Moon system to predict the planar periodic orbits in a synodic reference frame. The orbits for the exterior Lagrange point are examined with PSO in a similar manner. This research concludes PSO is effective in finding Lyapunov periodic orbits defined by low values of the Jacobi constant (the only integral of the motion in the CR3BP).

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Introduction

The study of the behavior of a body with a negligible mass in the gravitational Earth-moon system is referred to as the Circular Restricted Three-Body Problem (CR3BP). In the Earth-Moon system, there are five points where the combined gravity from the Earth and Moon provide the centripetal force needed to maintain circular motion about the common mass center. These locations are known as Lagrange points, identified as L_1, L_2,…L_5.

There are planar periodic orbits about the Lagrange points, also known as Lyapunov periodic orbits. These orbits are considered a potentially useful method of placing refueling stations in space. These orbits would allow payloads to be less dedicated to carrying fuel. Instead, fuel can be retrieved at the orbiting refueling stations already in space. In addition, a satellite placed in a Lyapunov periodic orbit at L_2 would be able to observe the surface of the far side of the moon with limited energy and have frames of visibility from Earth to transmit the data collected during observation.

With the application of Particle Swarm Optimization (PSO), this research examines the behavior of a body with negligible mass in the Lyapunov periodic orbit about the collinear interior and exterior Lagrange points, L_1 and L_2, in context of the CR3BP. Once the PSO structure is established for calculating Lyapunov periodic orbits at the interior Lagrange point, L_1, the PSO is reconfigured to calculate the same parameters for similar orbits at the exterior point, L_2.

Chapter 2

Methodology Used

The Particle Swarm Optimization (PSO) technique is a stochastic population-based method motivated by the unpredictable behavior of bird flocks searching for food. In this method, birds are characterized as particles each having a position to represent a possible solution for the unknown parameters of the problem under consideration. PSO uses information sharing within the population to find the global optimal solution for the entire swarm[1].

At the initialization of PSO, each particle's position is randomly generated. During each iteration, the particle with the best current position is determined and a correction velocity value is applied to the other particles to follow the best current solution. The number of iterations, parameters, and particles in the population can be set as desired. A greater number of particles and iterations can help improve accuracy, however, execution time may suffer.

This research applies the PSO algorithm to space trajectories. Specifically, PSO is applied to find the unknown parameters for the Lyapunov periodic orbits around the collinear interior and exterior Lagrange points. In this problem, the unknown parameters are the initial position and period which define these orbits. First, PSO is tested at a various population settings. Afterwards, PSO is used to determine the initial position and period that form the periodic orbit around the interior Lagrange point, L_1. Lastly, the PSO algorithm is applied to find the periodic orbits at exterior point, L_2.

Chapter 3

Defining the Problem

Before Particle Swarm Optimization can be used, the problem must be viewed in a form to which the technique can be applied. Because the problem is in the context of the CR3BP, the mass of third body, m_3, is considered negligible compared to the primary bodies, m_1 and m_2. Furthermore, the motion of the third body is described using a synodic coordinate system rotating with the primary bodies. With this coordinate system, canonical units are used to allow for convenient normalized units[1]. This problem assumes the distance between the Earth and the moon are constant at 1 distance unit (DU) = 384,400 km. Therefore, a time unit (TU) describes the synodic period as 2π TU (where TU = 375,190 s). The gravitational parameters of Earth and the moon are denoted as μ_E and μ_M, respectively. Starting the definition, $\mu_E + \mu_M = 1$ DU^3/TU^2, it can be rewritten as

$$\mu = \frac{\mu_M}{\mu_M + \mu_E} \tag{1}$$

where μ is introduced to relate the gravitational parameters of the primary bodies. As a result, $\mu = 0.01215510$ DU^3/TU^2. This constant becomes useful for the following equations of motion for the third body in the CR3BP:

$$\dot{x} = v_x \tag{2}$$

$$\dot{y} = v_y \tag{3}$$

$$\dot{v}_x = \Omega_x + 2v_y \tag{4}$$

$$\dot{v}_y = \Omega_y + 2v_x \tag{5}$$

where

$$\Omega = \frac{x^2 + y^2}{2} + \frac{(1-\mu)}{\sqrt{(x+\mu)^2 + y^2}} + \frac{\mu}{\sqrt{(x+\mu-1)^2 + y^2}} \tag{6}$$

Because these equations are used in context of the CR3BP, the Jacobi integral can be used to introduce the Jacobi constant, C, in order to relate the equations of motion in the dynamical system[1]:

$$C = 2\Omega - ({v_x}^2 + {v_y}^2) \tag{7}$$

The Jacobi constant, in units of DU^2/TU^2, reflects the energy of the system, as this constant is a defining quantity for the motion of the third body in the system. The flight path angle, γ, measured from the x-axis in the synodic coordinate system, allows the velocity components to be written as[2]:

$$v_x = \sqrt{2\Omega - C}\cos\gamma \tag{8}$$

$$v_y = \sqrt{2\Omega - C}\sin\gamma \tag{9}$$

Equation (8) and (9) are related by the following:

$$tan\,\gamma = \frac{v_y}{v_x} \tag{10}$$

After taking the derivative with respect to time of Eq. (10) and substituting Eqs. (8) and (9), the following useful equation is derived:

$$\dot{\gamma} = \frac{\Omega_y \cos\gamma - \Omega_x \sin\gamma}{\sqrt{2\Omega - C}} - 2 \tag{11}$$

where Ω_x and Ω_y are the partial derivatives of Eq. 6. Due to the existence of the Jacobi integral, Eq. (11) can be used to replace Eqs. (4) and (5). The new set of equations, including Eqs. (2), (3), and (11), is called *Birkhoff's Equations* which are used to determine the Lyapunov period orbits of the third body in the CR3BP[1].

For this application of the PSO methodology, the optimization is based on the principle that the body in the *periodic* orbit returns to the initial values after period *T* passes:

$$x(T) = x(0), \qquad y(T) = y(0) \qquad \gamma(T) = \gamma(0) + 2\pi n \tag{12}$$

where *n* is the given number of completed periodic orbits. Therefore, the objective (or fitness) function, J, is the following:

$$J = |x(T) - x(0)| + |y(T) - y(0)| + \min\{mod[|\gamma(T) - \gamma(0)|, 2\pi], mod[-|\gamma(T) - \gamma(0)|, 2\pi]\} \tag{13}$$

Equation (13) allows the problem to be approached as an optimization problem for the PSO algorithm. The PSO technique minimizes J when finding the unknown parameters for the Lyapunov periodic orbits. In this problem, the initial conditions and period for the state equations determine the orbits. These orbits are symmetrical with the *x* axis and the initial velocity is always parallel to the *y* axis. Therefore, the following constraints for the initial conditions apply for a periodic orbit about L_1 at $(x_{L1}, y_{L1}) = (0.836893, 0)$ DU and L_2 at $(x_{L2}, y_{L2}) = (1.1156, 0)$ DU:

$$x_{min} < x(0) < x_{L1,L2} \quad y(0) = 0 \qquad (14)$$

$$\gamma(0) = \frac{\pi}{2}$$

where x_{min} is 0.75 DU for L_1 and 1.05 DU for L_2. These initial conditions allow the objective function to be written as

$$J = |x(T) - x(0)| + |y(T)| + \min\{mod\left[\left|\gamma(T) - \frac{\pi}{2}\right|, 2\pi\right], mod\left[-\left|\gamma(T) - \frac{\pi}{2}\right|, 2\pi\right]\} \qquad (15)$$

In conclusion, the PSO technique minimizes Eq. (15) in order to the find the values for initial position, $x(0)$, and period, *T*, which define the Lyapunov periodic orbits.

Chapter 4

Results

PSO Study

The PSO algorithm is applied with a set of MATLAB scripts. The first study explores the convergence speed and accuracy of the PSO while varying the number of iterations and the number of particles in the population. First, the number of iterations is set at 200, 350 and 500 while keeping the population size at 30 and the Jacobi constant, at C= 3.00. Because a closed periodic orbit is consistently produced for C= 3.0 by the algorithm, this setting is used to conveniently examine the iteration dependency. Fig. 1 shows the plot of the optimized objective as a function of iteration, j, for all three iteration settings. All three settings appear to converge at approximately the same optimized objective function value between 10^{-6} and 10^{-7}.

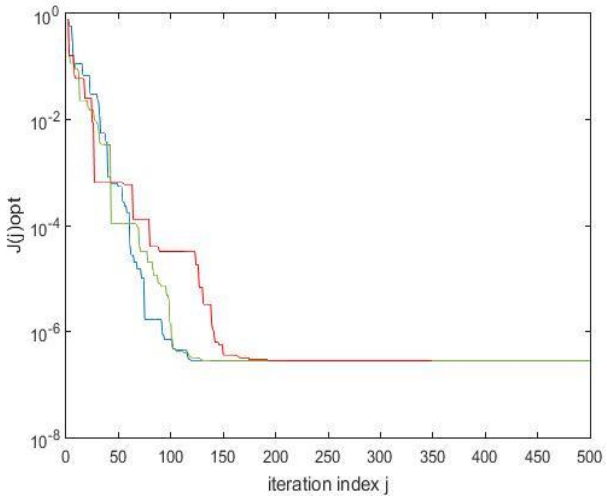

Figure 1. Objective evolution as function of iteration number, *j*.

Although setting the iteration number to a lower value would save time, all PSO runs for periodic orbits about L_1 and L_2 are set to 500 iterations to account for any unforeseen "jumps" in the objective function that would increase the accuracy of results.

Fig. 2a), b), and c) show the resulting trajectories for a population set at 10, 20, and 30 particles, respectfully.

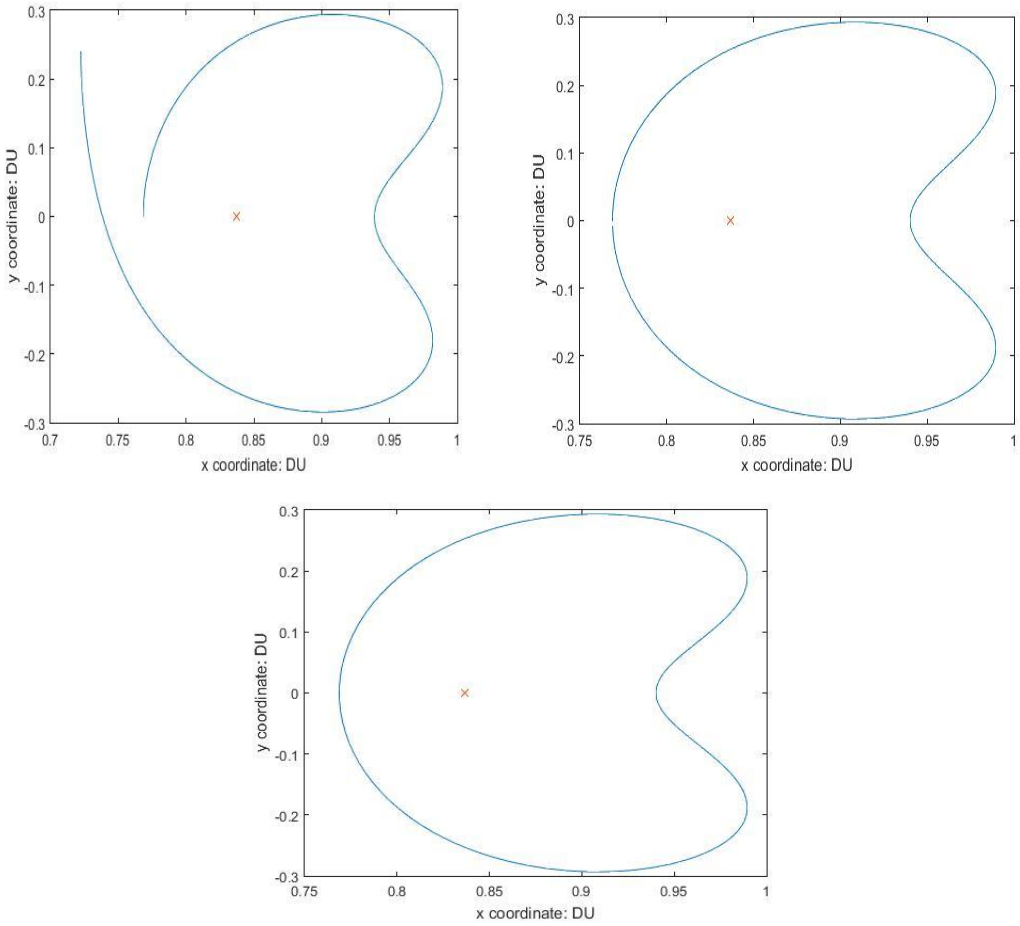

Figure 2 a-c). Results of varying population size.

Fig. 2c) is the only plot of the three varying population sizes that forms a closed curve. Fig. 2a) and 2b) form an orbit with a geometry similar to the plot with 30 particles, however, Fig. 2a) diverges near the end of the orbit and Fig. 2b) does not fully complete the trajectory at the

end of the curve. For this reason, 30 particles is used as the default setting for PSO application in this research.

PSO Application at L_1

Table 1 shows the solutions calculated by the PSO algorithm for L_1. In the previous section, Fig. 2c) shows the PSO algorithm is capable of determining periodic orbits about L_1 for C = 3.00, however, as the Jacobi constant is increased, the parameters tend to not converge to a suitable solution. Table 1 reflects this behavior.

Table 1. Parameters produced by PSO algorithm for varying Jacobi constants – L_1

C	$x(0)$, DU	T, TU	J_{opt}
3.00	0.7687	4.3349	2.7863×10^{-7}
3.02	0.8279	4.8608	1.3963×10^{-9}
3.04	0.7896	3.6857	4.7348×10^{-9}
3.06	0.7961	3.4378	0.0138
3.08	0.8262	4.9997	0.0033
3.10	0.8237	4.6548	0.0098
3.12	0.8227	4.9992	0.0332
3.14	0.8301	5.0000	0.1273
3.16	0.8185	2.2057	0.1293
3.18	0.8347	4.9937	0.2775

Table 1 shows an appropriately minimized final objective function, J_{opt}, for smaller values of C. Although the first three Jacobi constants result in reasonable optimized J_{opt} values, only C = 3 and C = 3.04 produce periodic orbits of expected behavior. These orbits are shown in Fig. 3.

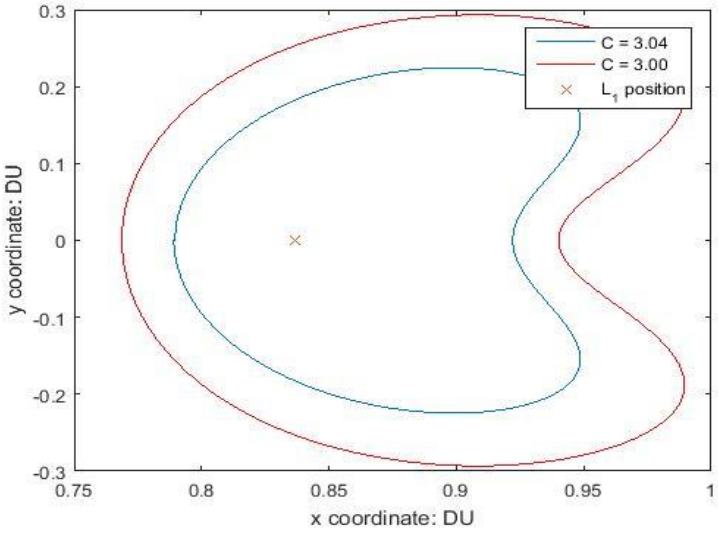

Figure 3. Successfully determined Lyapunov orbits produced by PSO algorithm.

Fig. 4 shows a few undesirable trajectories resulting from the parameters produced by the PSO algorithm. Fig. 4a) is a closed periodic orbit, however, this trajectory moves in close proximity of the moon which is located at (1, 0) DU. Fig. 4b) displays a similar trajectory to those in Fig 3., but is not considered periodic. The satellite does not return to its initial conditions after one period. Therefore, the orbit would not be repeated indefinitely.

Fig. 4c) shows a trajectory with a promising start, but eventually strays away to an unclosed orbit. In addition, Fig. 4d) shows oscillatory behavior, but is not classified a periodic orbit due to not returning to the initial conditions.

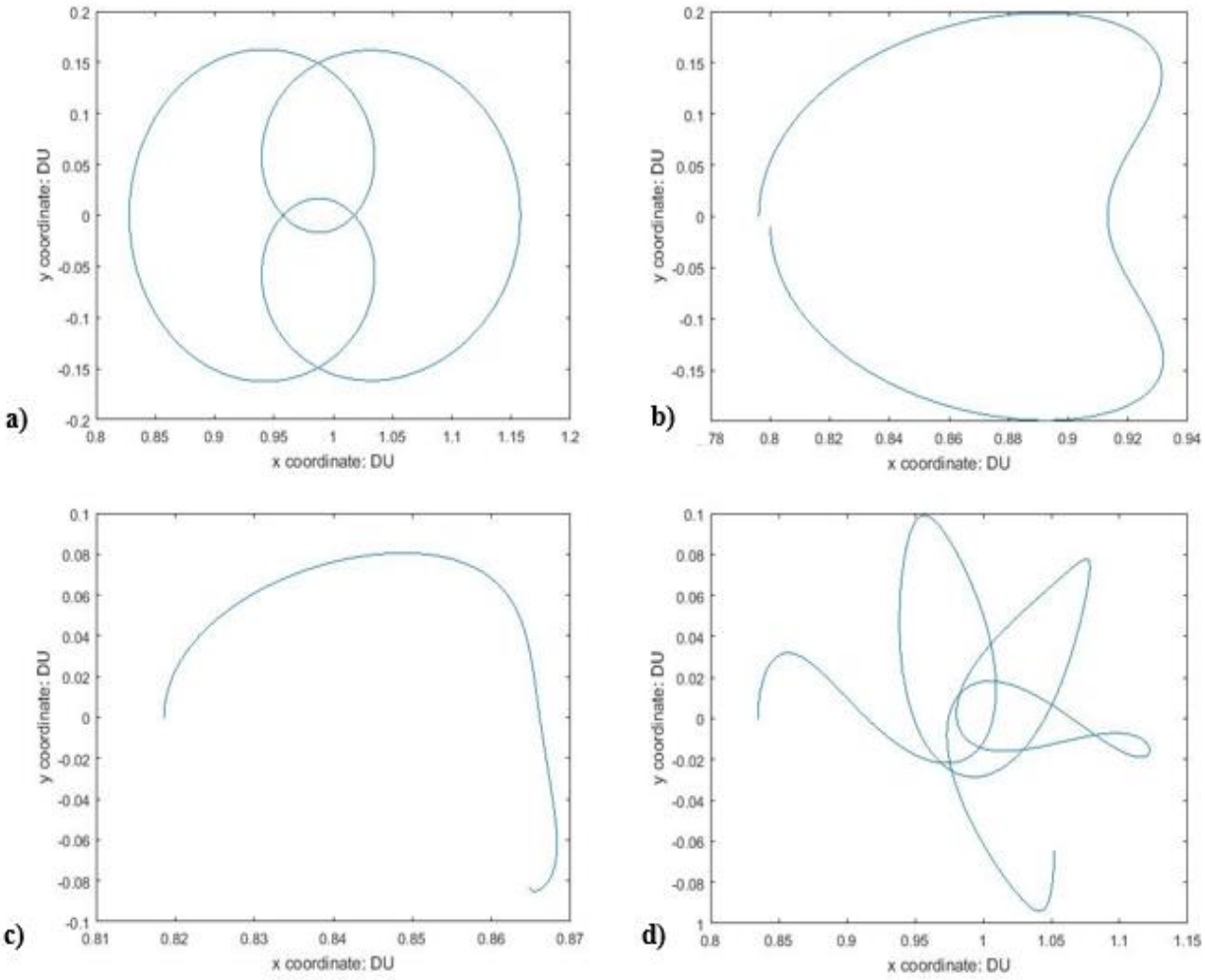

Figure 4. Successfully determined Lyapunov orbits produced by PSO algorithm.

PSO Application at L_2

Although there is limited success in determining periodic orbits for L_1 throughout a wide range of C values, attempts are made to produce periodic orbits at L_2. The PSO algorithm MATLAB scripts are adjusted so the appropriate bounds are established for the exterior Lagrange point.

Table 2. Parameters produced by PSO algorithm for varying Jacobi constants – L_2

C	$x(0)$, DU	T, TU	J_{opt}
3.18	1.0823	4.3445	1.3265×10^{-7}
3.20	1.0646	2.1864	0.0175
3.22	1.0500	3.5598	0.0161
3.24	1.1546	2.9563	0.2117
3.26	1.1547	2.9803	0.2299
3.28	1.0504	4.8976	0.0487
3.30	1.0698	2.5698	0.0859

Table 2 presents the parameters calculated by the PSO algorithm. C = 3.18 is the only setting that appears to have a successfully minimized objective function. This a familiar trend that is also occurs with L_1; the lowest C value produces parameters that form a periodic orbit about L_2. The resulting trajectory is displayed in Fig. 5. It is worth noting that the body does not orbit around L_2, but rather, it orbits around the location the moon would be positioned (1, 0) DU.

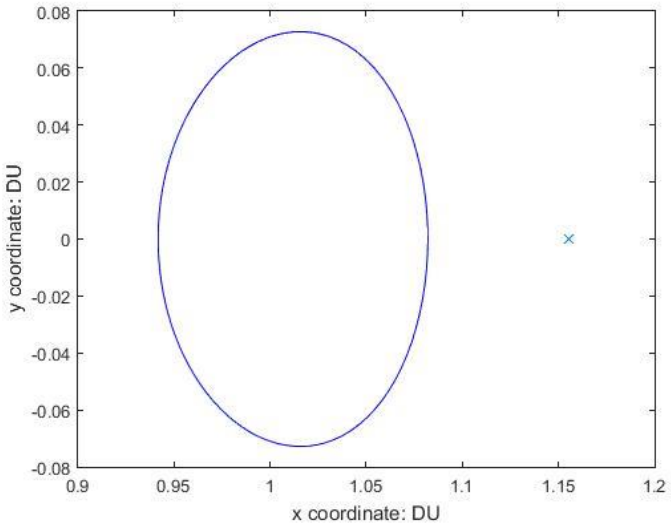

Figure 5. L_2 periodic orbit (C=3.18) determined by PSO algorithm

Fig. 6 shows two sample trajectories that form non-periodic orbits. Fig. 6a) seems to follow the same geometry as Fig. 5, but the satellite never returns to its initial values. Fig. 6b) shows a more sporadic oscillating orbit that appears to be even further from its initial values.

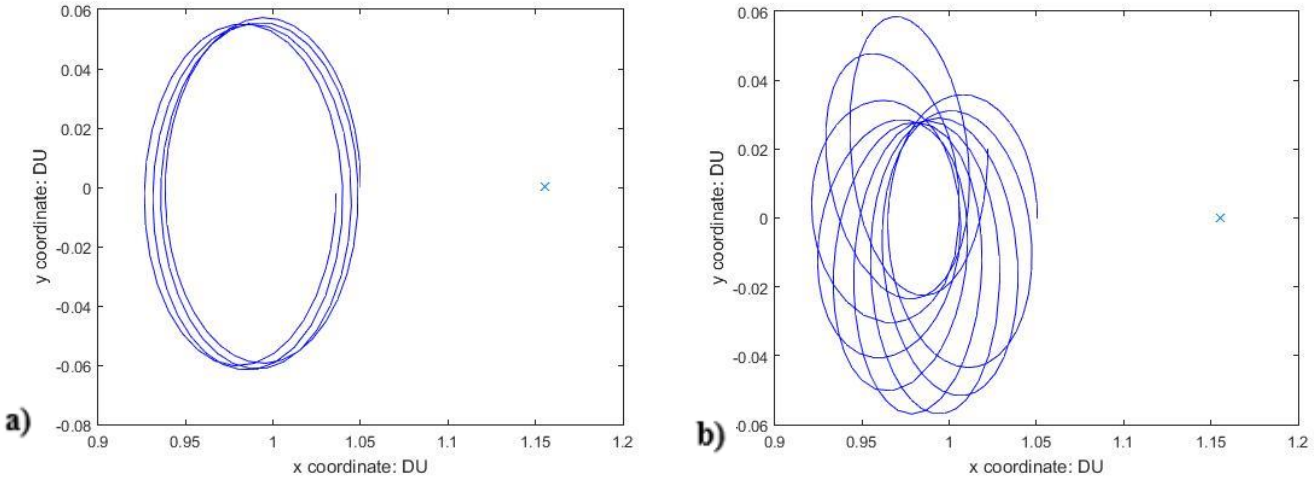

Figure 6. L_2 - Incorrect trajectories produced by PSO: a) C=3.22 b) C=3.28

Chapter 5

Observations

There are a few observations worth noting. As previously discussed, the PSO algorithm tends to be more successful in producing periodic orbits about each Lagrange point for lower Jacobi constant values. The amount of time needed to complete the PSO algorithm also increases as the Jacobi constant is increased. The periodic orbits seen in Fig. 3 and 4a) take approximately five minutes to obtain. When the Jacobi constant is set to values ranging between 3.14 to 3.18 for L_1, results take about fifteen minutes. During application of the PSO algorithm for L_2, which requires a greater range of Jacobi constant values than L_2, run time extends up to about thirty minutes. For L_2, most of the time is spent in the early iterations.

In some cases, the PSO algorithm has a tendency to converge to incorrect values for the parameters being optimized. For $C = 3.00$ and $C = 3.04$, multiple attempts are conducted before the PSO algorithm converges to a suitable solution. For this reason, global objective function values are output during each iteration to track the progress of the optimization. The PSO algorithm is stopped, and then restarted if there is no significant progress for the objective function minimization.

Chapter 6

Conclusions and Recommendations

Lyapunov orbits are sought for their ability to maintain an indefinitely repeating orbit. This study attempts to identify these periodic orbits about the interior and exterior Lagrange points with the Particle Swarm Optimization algorithm. This research concludes that the number of particles in the population is more significant than the number of iterations. As long as there is an adequate number of iterations, a greater number of particles leads to more accurate results. Based on this information, the algorithm is used to determine the values of the initial position and period with respect to the synodic reference frame. This research concludes Particle Swarm Optimization is more effective in determining periodic orbits for lower values of the Jacobi constant. In addition, the non-periodic solutions produced by the Particle Swarm Optimization algorithm in this study might repeat their complicated forms if given enough time. The solutions remain within bounded regions of the Lagrange points, so they might still be useful as communications relays between Earth and the far side of the Moon during certain time intervals.

There are a few suggestions for future work related to this study. For higher Jacobi constants, a greater population size should be tested to see if more suitable solutions can be obtained. However, this leads to the next issue of obtaining results in a timely manner. By increasing the number of particles, the Particle Swarm Optimization algorithm is expected to run even slower than described in this paper. Therefore, adjustments of other factors such as ODE solvers, tolerances, and parameter bounds should be tested as well. The Particle Swarm Optimization algorithm has proven potential to determine Lyapunov periodic orbits, but more

adjustments may be necessary to improve overall results and consistency. Lastly, other heuristic methods should be applied to this problem and compared to the effectiveness of Particle Swarm Optimization.

Appendix A

PSO_adaptive

```
%% PSO With Variable Accelerator Coefficients
%% Cory Caldwell
clc;
clear all;
format long;
tic

global P J JBest PBest GG N_particles N_elements V BLv BUv BLp BUp
global N_iterations mu C
N_particles = 30;
N_elements = 2;
N_iterations = 500;
mu = 0.01215510;

C = 3.00; % vary this

%% create random initial population
%% and set lower and upper bounds on unknowns (particle elements)

%%%%%%%% for L1
```

```
% BLp = [0.75 2];
% BUp = [0.836893 5];
%
%     for i=1:N_particles
%       P(i,1) = 0.75 + rand()*(0.836893-0.75) ;
%
%       P(i,2) = 2 + rand()*3.0;
%     end


% % %%%%%%%%%% for L2 AS MAX
BLp = [1.05 2];
BUp = [1.1556 5];
for i=1:N_particles
P(i,1) = 1.05 + rand()*(1.1556-1.05) ;
P(i,2) = 2 + rand()*3.0;
end


% initialize PSO variables
PBest = zeros(N_particles,N_elements);
J = zeros(N_particles); JBest = zeros(N_particles);
GGstar = zeros(N_iterations);
V = zeros(N_particles, N_elements);
```

```
%% determine velocity bounds
BUv = BUp - BLp;
BLv = -BUv;
for i = 1:N_particles
    JBest(i) = inf;
end
GG = inf;

for j = 1:N_iterations
    updated_EvalJ(j);
    EvalPGBest;
    UpdateV(j);
    UpdateP;
    GGstar(j) = GG;
    GG
    disp(j)
end
save('PSO_adapt.mat','C','P','GGstar','N_iterations','-append')

% plot GG behavior
figure
num_elements =  length(GGstar);
```

```
jspan = linspace(1,N_iterations,num_elements);

semilogy(jspan,GGstar)

xlabel('iteration index j')

ylabel('J(j)opt')

Cstr=num2str(C);

GGstr=num2str(GG);

title(['C=',Cstr,' GG=',GGstr])


toc
```

EvalJ

```
function updated_EvalJ(j)

%% EvalJ evaluates J for each particle in current iteration

global P J N_particles N_iterations C


y0 = 0;

gamma0 = pi/2;


for i = 1:N_particles
```

```
    %use last parameter values
    x0 = P(i,1);
    T = P(i,2);

    X_0 = [x0 y0 gamma0];

    tspan = [0,T];

    %integrate to find trajectory
    options = odeset('RelTol', 1e-6,'AbsTol',1e-6);
    [t,X] = ode45('ODEsolve',tspan,X_0,options);

    J(i) = abs(X(end,1) - x0) + abs(X(end,2)) + min(mod(abs(X(end,3)-gamma0),(2*pi)),...
        mod(-abs(X(end,3)-gamma0),(2*pi)));


end

%plots last iteration x vs y
if j == N_iterations
    xplot = X(:,1);
    yplot = X(:,2);
```

```
    save('PSO_adapt.mat','xplot','yplot')



    figure
    plot(xplot,yplot,'b')
    hold on

    %Lagrange point coordinates
    %L1x = 0.836893;
    L2x = 1.1556;
    L_y = 0;



    plot(L2x,L_y,'x')
    xlabel('x coordinate: DU')
    ylabel('y coordinate: DU')
    x0str = num2str(P(end,1));
    Tstr = num2str(P(end,2));
    Cstr = num2str(C);
    title(['C =',Cstr,' x(0)=',x0str,' T=',Tstr])
end
```

EvalPGBest

```
function EvalPGBest()
%% EvalP&GBest determines the best position visited by particle i up through
%%   the current iteration)
global P J JBest PBest GG GBest N_particles
for i = 1:N_particles  %@question: best position visited by PARTICLE?
   if J(i) < JBest(i)
      PBest(i,:) = P(i,:);
      JBest(i) = J(i);
   end
end
for i = 1:N_particles %@question: best position visited by SWARM?
   if J(i) < GG
     GG = J(i)
      GBest = P(i,:);
   end
end
```

UpdateP

```
function UpdateP()
```

```
%% UpdateP updates the position vector
%%
global P N_particles N_elements V BLp BUp
for i =1:N_particles
   P(i,:) = P(i,:) + V(i,:);
   for k = 1:N_elements
      if P(i,k) < BLp(k)
         P(i,k) = BLp(k);
         V(i,k) = 0;
      end
      if P(i,k) > BUp(k)
         P(i,k) = BUp(k);
         V(i,k) = 0;
      end
   end
end
```

UpdateV

```
function UpdateV(j)
%%
%% UpdateV updates the velocity vector V
```

```
%% Variable accelerator coeffs.
%%
global P PBest GBest N_particles N_elements V BLv BUv
global N_iterations
c_I = (1 + rand)/2;
c_C = 0.01 + 1.49445*rand*j/N_iterations;
c_S = 0.01 + 1.49445*rand*j/N_iterations;
%% c_C = 1.49445*rand;
%% c_S = 1.49445*rand;
c_S = 1.3*rand;
for i =1:N_particles
   V(i,:) = c_I*V(i,:) + c_C*(PBest(i,:) - P(i,:)) + c_S*(GBest - P(i,:));
   for k = 1:N_elements
      if V(i,k) < BLv(k)
         V(i,k) = BLv(k);
      end
      if V(i,k) > BUv(k)
         V(i,k) = BUv(k);
      end
   end
end
```

ODEsolve

```
function xdot = ODEsolve(t,x)

global mu C

xdot = zeros(3,1);
omega = 0.5*(x(1)^2 + x(2)^2) + (1-mu)/sqrt((x(1)+mu)^2 + x(2)^2) + mu/sqrt((x(1)+mu-1)^2 +
x(2)^2);
omega_x = -mu*(mu+x(1)-1)/((mu+x(1)-1)^2+x(2)^2)^(3/2) + (mu-
1)*(mu+x(1))/((mu+x(1))^2+x(2)^2)^(3/2) +x(1);
omega_y = -mu*x(2)/((mu+x(1)-1)^2+x(2)^2)^(3/2) - (1-mu)*x(2)/((mu+x(1))^2+x(2)^2)^(3/2)
+x(2);

xdot(1) = sqrt(2*omega - C)*cos(x(3));
xdot(2) = sqrt(2*omega - C)*sin(x(3));
xdot(3) = (omega_y*cos(x(3)) - omega_x*sin(x(3)))/sqrt(2*omega-C) - 2;
```

BIBLIOGRAPHY

[1]Pontani, M., & Conway, B. A. (2010). *Particle Swarm Optimization Applied to Space Trajectories.* JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS Vol. 33, No. 5, September–October 2010. American Institute of Aeronautics and Astronautics, Inc.

[2]Curtis, H. D. (2013). *Orbital Mechanics for Engineering Student. Third Edition.* Butterworth-Heinemann.

Academic Vita of Cory Caldwell

c.caldwell93@gmail.com

Bachelor of Science in Aerospace Engineering
Pennsylvania State University
Honors in Aerospace Engineering

Thesis Title: Application of Particle Swarm Optimization to Lyapunov Periodic Orbits
Thesis Supervisor: Robert G. Melton

Work Experience:
Lockheed Martin – Missiles and Fire Control
May 2015-Aug 2015
Worked on simulation development for internal research and development missile
Grand Prairie, TX
Supervisor: Chris Dorosky

Awards: Dean's List

Professional Memberships:
Sigma Gamma Tau Honors Society, Secretary
AIAA National and Penn State Chapter

Presentations: 2016 Region I AIAA Student Conference
Community Service Involvement: THON Operations Committee Member