ORBIT DESIGN AND ESTIMATION FOR SURVEILLANCE MISSIONS USING GENETIC ALGORITHMS

A Dissertation

by

OSAMA MOHAMED OMAR ABDELKHALIK

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

December 2005

Major Subject: Aerospace Engineering

ORBIT DESIGN AND ESTIMATION FOR SURVEILLANCE MISSIONS USING GENETIC ALGORITHMS

A Dissertation

by

OSAMA MOHAMED OMAR ABDELKHALIK

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Approved by:

Chair of Committee,	Daniele Mortari
Committee Members,	John L. Junkins
	Srinivas R. Vadali
	J. Maurice Rojas
Head of Department,	Helen Reed

December 2005

Major Subject: Aerospace Engineering

ABSTRACT

Orbit Design and Estimation for Surveillance Missions
Using Genetic Algorithms. (December 2005)

Osama Mohamed Omar Abdelkhalik, B.Sc., Cairo University;
M.Sc., Cairo University
Chair of Advisory Committee: Dr. Daniele Mortari

The problem of observing a given set of Earth target sites within an assigned time frame is examined. Attention is given mainly to visiting these sites as sub-satellite nadir points. Solutions to this problem in the literature require thrusters to continuously maneuver the satellite from one site to another. A natural solution is proposed. A natural solution is a gravitational orbit that enables the spacecraft to satisfy the mission requirements without maneuvering. Optimization of a penalty function is performed to find natural solutions for satellite orbit configurations. This penalty function depends on the mission objectives. Two mission objectives are considered: maximum observation time and maximum resolution. The penalty function poses multi minima and a genetic algorithm technique is used to solve this problem. In the case that there is no one orbit satisfying the mission requirements, a multi-orbit solution is proposed. In a multi-orbit solution, the set of target sites is split into two groups. Then the developed algorithm is used to search for a natural solution for each group. The satellite has to be maneuvered between the two solution orbits. Genetic algorithms are used to find the optimal orbit transfer between the two orbits using impulsive thrusters. A new formulation for solving the orbit maneuver problem using genetic algorithms is developed. The developed formulation searches for a minimum fuel consumption maneuver and guarantees that the satellite will be transferred exactly to the final orbit even if the solution is non-optimal. The results obtained demonstrate the feasibility of finding natural solutions for many case studies.

The problem of the design of suitable satellite constellation for Earth observing applications is addressed. Two cases are considered. The first is the remote sensing missions for a particular region with high frequency and small swath width. The second is the interferometry radar Earth observation missions. In satellite constellations orbit's design, a new set of compatible orbits, called the “Two-way orbits”, whose ground track path is a closed-loop trajectory that intersects itself, in some points, with tangent intersections is introduced. Conditions are derived on the orbital elements such that these Two-way Orbits exist and satellites flying in these orbits pass the tangent intersection points at the same time. Finally, the recently proposed concept of observing a space object from onboard a spacecraft using a star tracker is considered. The measurements of the star tracker provide directions to the target in space and do not provide range measurements. Estimation for the orbit of the target space object using the measurements of the star tracker is developed. An observability analysis is performed to derive conditions on the observability of the system states. The Gaussian Least Squares Differential Correction Technique is implemented. The results obtained demonstrate the feasibility of using the measurements of the star tracker to get a good estimate for the target orbit within a period of measurements ranging from about 20 percent to 50 percent of the orbital period depending on the two orbits.

To my father, my mother, my wife, and my kids Ganna and Noureddean

ACKNOWLEDGMENTS

I would like to express my thanks to my advisor, Dr. Daniele Mortari, for his advice throughout my Ph.D. journey. He was very close to me all the time, helping and advising.

I would like also to express my gratitude and thanks to Dr. John Junkins for his continuous support. He helped me join this department and helped me throughout the program.

I also thank Dr. Srinivas Vadali and Dr. J. Maurice Rojas for their support and suggestions.

Also, I would like to thank the Department of Aerospace Engineering at Texas A&M University for the support I received throughout my Ph.D. program. A special thanks goes to Ms. Karen Knabe for her administrative assistance.

I would like also to thank Dr. Sayed Hasan, who was my advisor in my master's program. He always encouraged me and sincerely advised me whenever I needed him.

Finally, many thanks to my family. I would like to thank my father, mother, and my bothers for their unlimited love and support. My love and appreciation go to my wife, Doaa, for her devotion.

TABLE OF CONTENTS

CHAPTER Page

LIST OF TABLES

TABLE Page

LIST OF FIGURES

FIGURE Page

FIGURE Page

CHAPTER I

INTRODUCTION

The orbits of space remote sensing systems usually control the ground resolution, area coverage, and the frequency of coverage parameters. The orbit hight affects the resolution and the swath width. The higher the spacecraft the more area that is covered with a corresponding decrease in the resolution of measurements. Lower altitudes enable a spacecraft to get higher resolution measurements but orbit perturbations will be non-negligible due to air drag [7]. Electric propulsion systems were proposed to enable such missions where the orbit altitude is low [1]. The air drag, in this case, is compensated using electric thrusters. Low altitude orbits, however, poses lack of coverage. To overcome this lack of coverage, strategies for maneuvering the spacecraft using the electric propulsion also were proposed in literature [15]. In this case, the thrusters are not only used to compensate the perturbations but also are used to continuously maneuver the spacecraft to achieve given coverage requirements. This solution, however, complicates the satellite system by adding a propulsion system to it. The motivation for this study is to search for a natural orbit that will enable a spacecraft to achieve the mission coverage requirements without the use of propulsion.

Ground surveillance can be defined as the observation of a discrete number of Earth surface locations from space. This kind of missions require that the satellite visits all the given sites within a given time frame. A site is said to be visited if the nadir point of the satellite passes through a given specified neighborhood of that target. The neighborhood size is determined based upon the mission scenario and sensors specifications. One example is to visit a set of 20 sites on the terrestrial

The journal model is *IEEE Transactions on Automatic Control.*

surface within a time frame, t_f, of 50 days. In other applications this time could be $t_f = 3$ days.

Throughout this study, the ground surveillance remote sensing missions are considered. For this type of missions, maneuvering strategies were developed in the literature to achieve the coverage requirements. Guelman and Kogan [15] developed an algorithm to provide the spacecraft the ability to pass over the target sites within the specified time frame. In this algorithm, the optimal trajectory is calculated on two steps, piecewise optimization and scheduling. The piecewise optimization is the calculation of the optimal trajectory that connects two sequential target sites. Pontryagin's principle is used to find the optimal trajectory. Scheduling is the selection of visiting time for each site. The simulated annealing method is used to solve the scheduling part.

The solution proposed in this study, however, does not use propulsion for maneuvers. A global search is performed to find a natural orbit that will enable the spacecraft to visit all the target sites within the assigned time frame. A penalty function is constructed to represent how far a given orbit is far from an ideal orbit. The global search will find the orbit that minimizes this penalty function in the invistigated regions of the design space. The search algorithm exploites the most fit solutions to direct the search into regions with higher potential for having the global optimal. This global search is performed using the genetic algorithms technique. The reason for using the genetic algorithms is because the penalty function has multiple local minima.

In case all the sites in the given set cannot be visited within the assigned time frame, then the given set is split into two subsets. The global search method is then used to find a natural orbit for each subset. The complete mission is then achieved by maneuvering the spacecraft between the two orbits or by having two spacecraft, one for each orbit. In case a single spacecraft is used, an orbit transfer algorithm is developed. The proposed algorithm implements the impulsive thrusters. The minimum fuel maneuver is calculated using genetic algorithms. A new formulation for this problem is introduced. This formulation guarantees that the satellite will arrive at the final orbit exactly even if the solution provided by the genetic algorithms is not the optimal solution.

Throughout this study, spacecraft are assumed to be moving in Keplerian orbits. A brief review for Keplerian orbits is described in section(A). The genetic algorithms technique is described in section(C). Section(D) describes the objectives of this dissertation. The organization of this dissertation is described in section(E).

A. Keplerian Orbits

The spacecraft is assumed to be subject only to the Newtonian attraction from the central body. In this case, the motion of the spacecraft can be described by [8]:

$$\ddot{\mathbf{r}} + \frac{\mu \mathbf{r}}{r^3} = \mathbf{0} \tag{1.1}$$

where $\mathbf{r}$ is the position vector of the spacecraft and μ is the gravitational constant of the central body. Throughout this dissertation, the central body is the Earth unless otherwise stated. The gravitational constant for the Earth is $\mu = 3.986004415 \times 10^5 Km^3/s^2$ [34].

Since the subject of this dissertation addresses orbits that are suitable for Earth observations, only elliptical orbits will be considered in this study. Elliptical orbits can be completely defined by one of the following sets:

1. The orbital elements

2. Position and velocity vectors for the satellite at a point on the orbit

3. Two positions on the orbit and the time of satellite travel from one point to the other.

The orbital elements description will be used throughout this study. Orbital elements are described in brief in this chapter. Given the position and velocity vectors of the satellite at one point on the orbit, these two vectors can be transformed to the orbital elements and vise versa. The determination of the orbit from two position vectors on the orbit given the time of satellite travel between them is known in literature as Lambert's problem. Lambert's problem will be explained in Chapter IV.

1. Orbital Elements

Orbital elements are widely used to describe the position of a spacecraft in space. The position of a spacecraft is described using six parameters or elements. Two of them are angles defining the plane of the orbit. These two angles are the inclination of the orbit plane on the equatorial plane, $\imath$, and the right assention of the ascending node, Ω, which is the angle of the line of nodes measured from the inertial vernal equinox in the equatorial plane. Two other parameters define the size and shape of the orbit, these are the semi-major axis a, and the eccentricity e. The fifth element defines the orientation of the orbit in the orbit plane. This is the angle, ω, of the semi-major axis measured from the line of nodes in the orbit plane. The last element is an angle defining the position of the spacecraft in the orbit. This angle is called the true anomaly, θ, and is the angle from the perigee to the satellite position measured in the orbit plane and from the center of the Earth.

2. Transformation from Orbital Elements to Position and Velocity Vectors

This transformation is used frequently throughout this work and is detailed here. The derivation in this section follows the work presented in [32]. Given the six orbital elements, it is required to find the position and velocity of the satellite in the inertial frame. Consider the orbit plane. Assume a coordinate system in this plane: **x**, **y**, and **z** as shown in Fig. 1. Unit vectors in this coordinate system are: $\hat{\mathbf{x}}$, $\hat{\mathbf{y}}$, and $\hat{\mathbf{z}}$. Note that in this coordinate system, the satellite position will always have $z = 0$. From Fig. 1:

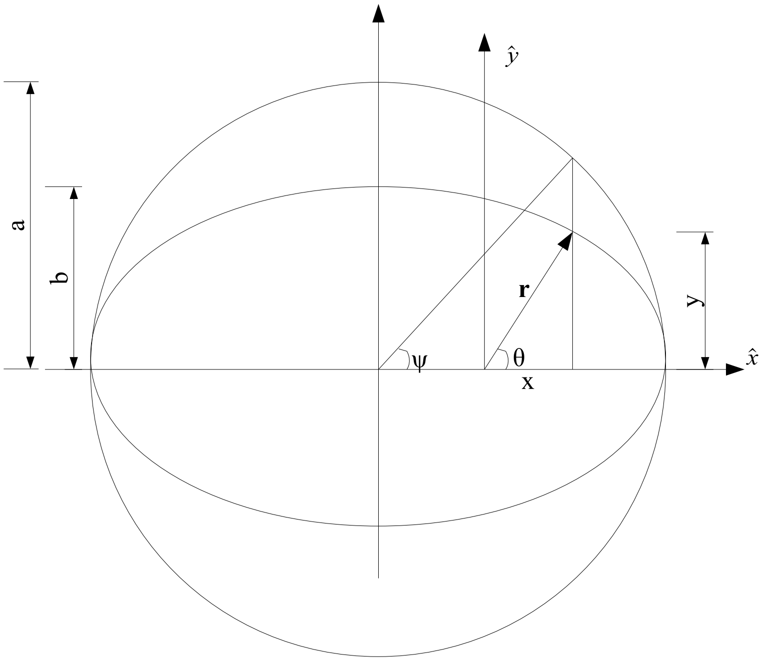

Fig. 1. Orbital plane coordinate system

$$x = a\cos(E) - c = a\cos(E) - ae \tag{1.2}$$

$$y = \frac{ab\sin(E)}{c} = a\sin(E)\sqrt{1-e^2} \tag{1.3}$$

$$\mathbf{r} = x\hat{\mathbf{x}} + y\hat{\mathbf{y}} \tag{1.4}$$

The eccentric anomaly, E, can be easily calculated from the mean anomaly, M, iteratively using the kepler equation:

$$E - e\sin(E) = M = nt \tag{1.5}$$

So, $\mathbf{r}$ is calculated in the orbital plane coordinate system. Next, the vector $\mathbf{r}$ is transferred to the inertial coordinate system [32]:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = [A_z(\Omega)]^T [A_x(\imath)]^T [A_z(\omega)]^T \begin{bmatrix} x \\ y \\ 0 \end{bmatrix} \tag{1.6}$$

where, X,Y, and Z are the inertial components of $\mathbf{r}$, and

$$[A_z(\omega)] = \begin{bmatrix} \cos(\omega) & \sin(\omega) & 0 \\ -\sin(\omega) & \cos(\omega) & 0 \\ 0 & 0 & 1 \end{bmatrix}, [A_x(\imath)] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\imath) & \sin(\imath) \\ 0 & -\sin(\imath) & \cos(\imath) \end{bmatrix}$$

The velocity vector is calculated first in the orbital plane coordinate system and then transferred to the inertial coordinate system.

$$\mathbf{v} = \frac{d\mathbf{r}}{dt} = \frac{d\mathbf{r}}{dE}\frac{dE}{dt} \tag{1.7}$$

From Kepler equation:

$$\frac{dM}{dt} = n = \frac{dE}{dt} - e\cos(E)\frac{dE}{dt} \tag{1.8}$$

Recall that:

$$r = a[1 - e\cos(E)] \tag{1.9}$$

where r is the magnitude of $\mathbf{r}$. Using (1.9)and (1.8) into (1.7):

$$\mathbf{v} = \frac{a^2 n}{r}[-\sin(E)\hat{\mathbf{x}} + \sqrt{1-e^2}\cos(E)\hat{\mathbf{y}}] \tag{1.10}$$

The transformation in (1.6) is then used to find the inertial coordinates of $\mathbf{v}$.

The transformation from position and velocity vectors to orbital elements follows a similar derivation and is detailed in [32].

B. Optimization Methods

This section describes common standard global optimization algorithms. This description summerizes the survey available in Ref. [13]. The focus is on general techniques that are applicable to a wide variety of combinatorial and continuous optimization problems.

1. *Branch and Bound*: Branch and Bound is a general search method. Suppose we wish to minimize a function f(x). To apply branch and bound, it is required to compute a lower bound on an instance of the optimization problem(bounding). It is also required to be able to divide the feasible region of a problem to create smaller subproblems(branching). The method starts by considering the original problem with the complete feasible region, which is called the root problem. The lower bound and upper bound are calculated to the root problem. If the bounds match, then an optimal solution has been found and the procedure terminates. Otherwise, the feasible region is divided into subregions, which together cover the whole feasible region. These subregions become children of the root search node. The algorithm is applied recursively to the subproblems, generating a tree of subproblems. If an optimal solution is found to a subproblem, it is a feasible solution to the full problem, but not necessarily globally optimal. Since it is feasible, it can be used to prune the rest of the tree: if the lower bound for a node exceeds the best known feasible solution, no globally optimal solution can exist in the subregion of the feasible region represented by the node. Therefore, the

node can be removed from consideration. The search proceeds until all nodes have been solved or pruned, or until some specified threshold is meet between the best solution found and the lower bounds on all unsolved subproblems. For the application of interest, this method can be applied. However, a bounding method must be found.

2. *Clustering methods*: Clustering methods are a modified form of the standard multistart procedure, which performs a local search from several points distributed over the entire search domain. In a multistart procedure, when many starting points are used, the same local minimum may be identified several times, thereby leading to an inefficient global search. In Clustering methods, points at which the local search is initiated are carefully selected such that this inefficiency is avoided. First, points are sampled in the search domain, then the sampled points are transformed to group them around the local minima. Finally, a clustering technique is applied to identify groups that represent neighborhoods of local minima. Redundant local searches can be avoided by simply starting a local search for some point within each cluster.

3. *Evolutionary algorithms*: Evolutionary algorithms (EAs) are search methods that take their inspiration from natural selection and survival of the fittest in the biological world. EAs involve a search from a population of solutions, not from a single point. The solutions with high fitness are recombined with other solutions by swaping parts of a solution with another. Solutions are also mutated by making a small change to a single element of the solution. Recombination and mutation are used to generate new solutions that are biased towards regions of the space for which good solutions have already been seen. genetic algorithms are one type of the EAs and is the method adopted to solve

the problem of interest. More details on the method is presented in section C.

4. *Tree annealings*: Simulated annealing was designed for combinatorial optimization, usually implying that the decision variables are discrete. A variant of simulated annealing called tree annealing was developed to globally minimize continuous functions.

   In simulated annealing, one is given an initial point x and randomly generates a new candidate solution, y. If f(y) ¡ f(x), then y replaces x and the process repeats. If f(y) ¿ f(x), y replaces x with a probability that is based on the Metropolis criteria and the progress of the algorithm. In tree annealing, candidate subspaces are accepted or rejected by evaluating the function at representative points.

   Application Domains It is not guaranteed to converge, and often the rate of convergence is very slow [13]. It is recommended to use tree annealing in the first part of optimization to find the region of the global optimum, then use a gradient descent or other method to hone in on the actual value. Given enough time, a sufficiently low final temperature, and a slow cooling rate, tree annealing should at least converge to a local minimum, and will often converge to a global minimum, especially for piecewise functions with discontinuities [13].

   Tree annealing has been used in applications such as signal processing and spectral analysis, neural nets, data fitting, etc. These problems involve fitting parameters to noisy data, and often it is difficult to find an optimal set of parameters via convential means.

C. Genetic Algorithms

The original concept of genetic algorithms was first introduced in the 1970s by John Holland of the University of Michigan. As the name denotes, genetic algorithms try to artificially imitate the concept of evolution observed in nature. A wide variety of genetic algorithm applications has been developed: control systems engineering, robotics, wire routing, game playing, cognitive modelling, seismology, structural dynamics, aircraft and ship design, spacecraft trajectory optimization, and other applications.

The current literature identifies three main types of search methods [12]:

Calculus-based methods: these methods have been extensively studied and used in literature. Calculus-based methods can be categorized into two categories:

- Direct methods: these are hill-climbing methods, where the optimum point is searched by evaluating the local gradient at a point and move in the steepest direction.
- Indirect methods: These method utilize the fact that the gradient at the optimum point is zero. By equating the gradient to zero, we get a set of equations. This set of equations is usually nonlinear.

Both methods are local. Starting with an initial guess point, the optimum they seek is the best in a neighborhood of the current point. If the optimized function has multi minima, then the solution that we get is the local optimum near the initial guess and there is no way to take off from that local minimum unless through a random restart. Another important disadvantage in calculus-based methods is that they require the existence of derivatives. Many practical parameter spaces are discontinues, some problems have integer variables in the design parameters.

Enumerated methods: the basic idea of these methods is to evaluate the objective function values at every point within a finite search space, one at a time. These methods are not efficient for problems with large search space.

Randomized methods: these methods processes direct search and performs random choices as a tool during the search. Genetic Algorithms (GAs) and Simulated Annealing Methods are two examples for randomized techniques.

GAs then have the advantage of being a robust technique among a wide range of problems. GAs differ from other optimization techniques in the way it works as follows. Calculus-based methods start with an initial point. GAs, however, start with a random initial population of points. Derivatives information are not needed in GAs. Only function evaluations are performed for the objective function. Thus, the continuity condition on the objective function is not required. The operations performed by the GAs are probabilistic not deterministic.

GAs work with coded parameters set. A binary representation for the parameters set is used rather than the parameters themselves. Assume that the parameters that we wish to optimize are x, y and z. Each variable is assigned a number of bits for coding. The number of bits is determined based on the range of this variable and the required accuracy for the variable. Assume the number of bits for x is 4, the number of bits for y is 5, and the number of bits for z is 4. Then the coding is to find the binary code for each variable and then stack them together in one string. This string is called a *member* or a *chromosome*. If the values of the variables are: $x = 6$, $y = 24$, and $z = 13$, then the corresponding chromosome is 0110110001101. The structure of chromosome is illustrated in Fig. 2.

GAs work on a group of chromosomes called a *population*. The initial population is randomly selected. GAs performs a series of probabilistic operations on the current population to generate a new *generation*. The new generation is then used as the

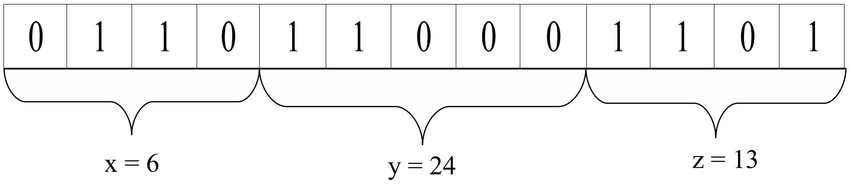

Fig. 2. Chromosome consists of a string of variables

current population and the GAs operations are performed again to produce a third generation and so on. Each generation is an iteration.

In a maximization problem, it is required to find the maximum of an objective function. The value of the objective function corresponding a member is called the *fitness* of the member. Literature presents many levels of genetic algorithms based on the complication of the algorithm. Basic operations that are used in all the algorithms are: reproduction, crossover, and mutation.

Reproduction: sometimes referred to as *selection*, is the operation of selecting some members out of the current generation to be transferred to the next generation. There is a number of techniques in literature that are used for reproduction. The simplest is to select the fittest 50% of the current generation to be transferred to the next generation. A more common technique is the fitness-proportional, or roulette wheel, technique where a member is selected for reproduction with a probability proportional to its fitness. The roulette wheel however does not guarantee that the fittest member in the current generation will be selected. It is possible that each time a new generation is produced, the fittest member in the previous generation is not selected. In this case the process may take longer time to converge to the best member. The speed of the process can be improved by not losing the best member from the current generation and transfer it as is to the next generation. This is called *elitism*. However, this is a trade off. If the elite member is not selected and the process takes more time until convergence, this will allow for more exploration for the search space leading to higher probability to locate the true global optimal [10].

Crossover: is a recombination operator where information from different members are melded to construct new members in a new portion in the design space. Pairs of members are selected to undergo a crossover with a probability P_c. The crossover is performed by cutting the pair of members at a random point and swapping the tails to generate two new members. The new members are called children, the original pairs are called parents. This is called single point crossover, and this is what is used throughout this study. Nature, however, have up to eight crossover points [2]. Example for the crossover operation is shown in table I, where the crossover point is selected to be after the third digit from the left.

Table I. Single point crossover operation

Parents	1101/001	1010/111
Children	1101/111	1010/001

Mutation: is the operation of flipping a 1 to a 0 or vice versa. This operation is applied to all bits in all members in a generation with a probability P_m. P_m usually takes a value of about 0.001 [12]. However, P_m value is problem dependant. Many have used $P_m = 1/L$, where L is the length of the member [10].

A flow chart for a simple GA is shown in Fig. 3. There is no guarantee that GAs converge to the true global optimal. To explain more about how the GAs work and why does it work, the concept of similarity templates is introduced.

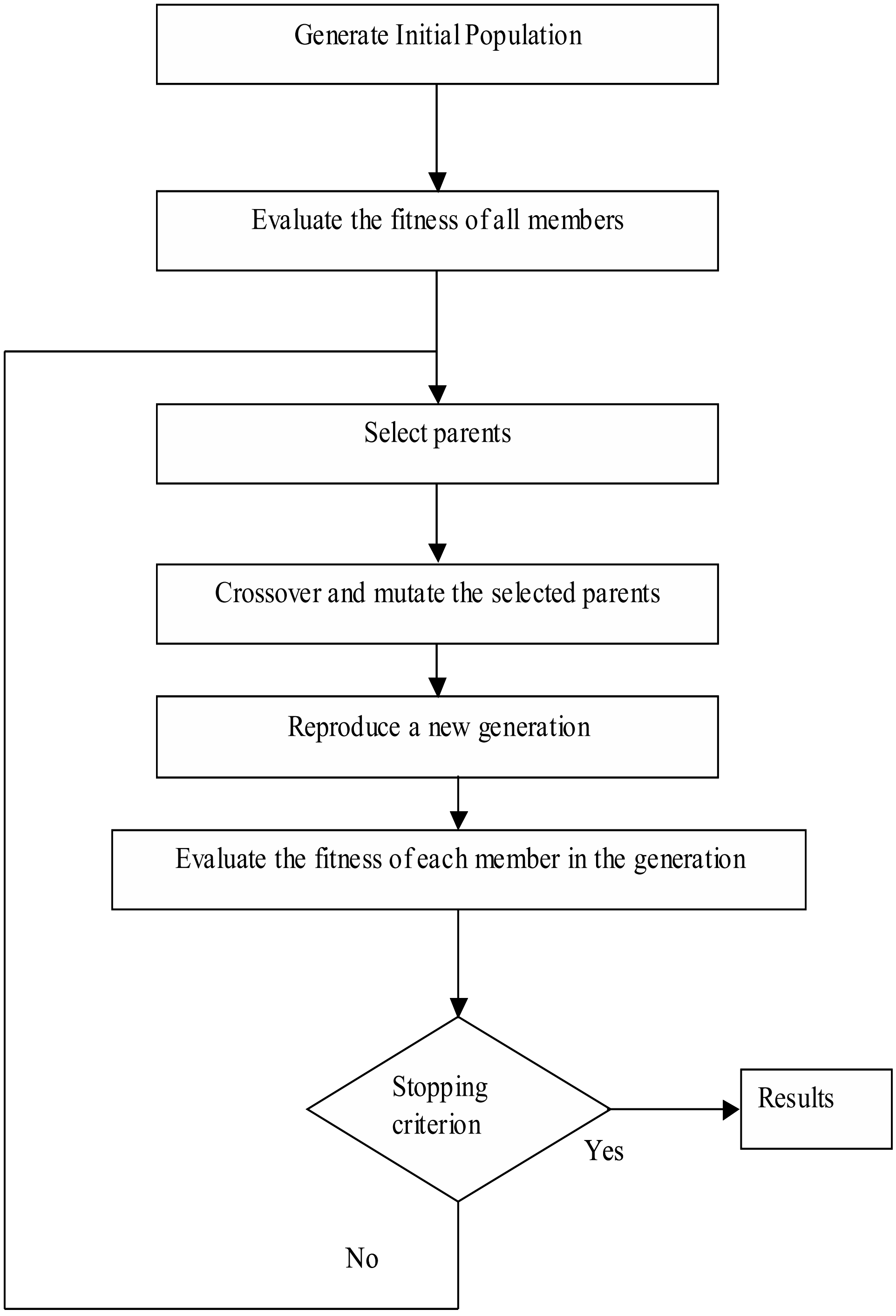
Generate Initial Population
Evaluate the fitness of all members
Select parents
Crossover and mutate the selected parents
Reproduce a new generation
Evaluate the fitness of each member in the generation
Stopping criterion
Yes
Results
No

Fig. 3. Flow chart for a simple genetic algorithm

1. Similarity Templates

Similarity templates are sometimes named as Schemata. A schema, the singular of schemata, is a subset of chromosomes, or members, with similarities at certain string positions. As an example, the schema $(*111*)$ describes a subset of 4 members: (01110), (01111), (11111), and (11110). The schema $(0 * 1 * *)$ describes a subset of 8 members that begin with a 0 and have 1 in the third position. To illustrate how the schemata are used to explain why GAs work, a common example in literature is introduced [10], [12]. Consider the function $f(x) = x^2$. In this example there is only one parameter to optimize, x. Consider the four chromosomes in table II and their fitness values:

Table II. Chromosomes and fitness values for given points of x

x-value	Chromosome	Fitness
13	01101	169
24	11000	576
8	01000	64
19	10011	361

Looking at the fitness of each member, the two most fit members are the second and the fourth members. Both has a 1 at the left most digit. We may conclude from this that the optimal solution will have a 1 in the left most digit. So, the basic idea is to look for similarities among members and then find a relation between these similarities and high fitness. The number of the members of a schema with high fitness will be increased from one generation to the next, while the number of members of a schema with lower fitness will decreased. To prove this statement, the following definitions are first introduced.

- The order of a schema H, denoted by $o(H)$, is the number of fixed positions in the template. e.g. $o(011 * 1 * *) = 4$.

- The defining length of a schema, $\delta(H)$, is the distance between the first and the last specific positions in the string. e.g. $\delta(011 * 1 * *) = 5 - 1 = 4$ and $\delta(* * 1 * * * *) = 0$.

Assume a string $A = a_1a_2a_3a_4a_5a_6a_7$, and let $\mathbf{A}(t)$ be the population at generation t. Suppose at a given time step t, there are m members, or examples, of the schema H, $m = m(H,t)$. During reproduction, a string is copied according to its fitness, so A_i is selected with probability $p_i = f_i/\Sigma f_j$. If the population size is n, then:

$$m(H, t+1) = \frac{m(H,t)nf(H)}{\Sigma f_j} \tag{1.11}$$

where $f(H)$ is the average fitness of the strings representing H. Let the population average fitness be $f_p = \Sigma f_j/n$ then:

$$m(H, t+1) = \frac{m(H,t)f(H)}{f_p} \tag{1.12}$$

From (1.12), the number of members in a particular schema grows as the ratio of the average fitness of the schema to the average fitness of the population. All schemata can grow or decay under the operation of reproduction alone. To investigate what is the rate of growth or decay for a particular schema, assume a schema H remains above average with amount cf_p where c is constant, then:

$$m(H, t+1) = \frac{m(H,t)(f_p + cf_p)}{f_p} = (1+c)m(H,t) \tag{1.13}$$

Starting at time $t = 0$, and assuming a stationary value of c, then:

$$m(H,t) = m(H,0) \times (1+c)^t \tag{1.14}$$

From (1.14), we can conclude that reproduction allocates exponentially increasing (decreasing) strings of schemata to above(below) average schemata.

So far, only the reproduction operation is considered. Next, we investigate the effect of crossover operation on the growth or decay of a schema members. To do that, consider a member $A = 011|1000$ that will undergo a crossover operation at the shown position. The member A belongs to many schemata, of them are $H1 = *1*\,|\,***0$ and $H2 = ***|10**$. Both schemata will be undergo the same crossover operation in the same position as A. Clearly, $H1$ will be destroyed while $H2$ will survive. It is clear that $H2$ is less likely to survive crossover than $H1$ because on average the cut point is more likely to fall between the extreme fixed positions. This previous observation can be quantified as follow: A schema is destroyed with the probability of having the crossover point within the fixed points of the schema, so the probability of destruction P_d is:

$$P_d = \frac{\delta(H)}{(l-1)} \tag{1.15}$$

where l is the number of bits in the member. The probability of Survival is then:

$$P_s = 1 - \frac{\delta(H)}{(l-1)} \tag{1.16}$$

If the crossover itself is performed by random choice, with probability P_c at a particular mating then:

$$P_s = 1 - P_c\frac{\delta(H)}{(l-1)} \tag{1.17}$$

Assuming the independence of crossover and reproduction operations, then:

$$m(H,t+1) = m(H,t)\left(\frac{f(H)}{f_p}\right)\left(1 - P_c\frac{\delta(H)}{(l-1)}\right) \tag{1.18}$$

Those schemata with both above-average performance and short defining lengths are going to be sampled at exponentially increasing rates.

To take into account the effect of mutation operation, recall that mutation is a random alternation of a single position with probability P_m. A single bit survives with probability $(1-P_m)$. A schemata H survives if all its specified positions survive. Each of the mutations are independent. A schema then survives with probability $(1 - P_m)o(H)$. If $P_m << 1$, then the survival probability can be approximated as: $1 - o(H)P_m$.

Combining the three operations effects in one formula, and neglecting the cross product terms: we conclude that a particular schema receives an expected number of copies in the next generation under reproduction, crossover and mutation as given by:

$$m(H,t+1) = m(H,t)\left(\frac{f(H)}{f_p}\right)\left(1 - P_c\frac{\delta(H)}{(l-1)} - o(H)P_m\right) \tag{1.19}$$

Equation (1.19) represents the *Fundamental Theorem of GA(schema theorem)*:

"Short, low-order, above-average schemata receives exponentially increasing individuals (trials) in subsequent generations."

Short, low-order, highly fit schemata are called *building blocks.*

It is important to note here that the concept of schemata is used to illustrate why the GAs work. Though the GAs does not use schemata for processing but rather uses the probabilistic operations defined earlier, reproduction, crossover and mutation.

This was a brief for the fundamentals of GAs necessary to understanding the developed work in the next chapters. More details on the principals of GAs can be found in [12], [10], [18], [28].

D. Objective of This Dissertation

The problem of optimal trajectories of spacecraft for remote sensing applications will be studied. First, current development in literature will be reviewed. For Ground surveillance missions, a proposed natural solution based on searching for the optimal orbit will be formulated. The genetic algorithms optimization technique will be studied and the feasibility of applying it to the problem will be investigated. Case studies will be considered and comparison with other solutions in literature will be presented. In cases where there is no solution that achieves the mission requirements exist, then a multi-orbit solution is proposed. In this case, the set of targets will be split into two groups. A solution is proposed for each group. The spacecraft will be maneuvered from one orbit to another. The problem of optimal orbit transfer is addressed. Genetic algorithms technique is used to solve this problem. A new formulation is proposed that has advantages to the current formulations in literature. The two-way set of orbits are introduced in this research. The concept is to have two spacecraft passing over the same Earth target at the same time and parallel to each other. Or, to have a spacecraft, that is maneuvered between two orbits, to pass over a particular target in the two orbits. Conditions on the orbit parameters for the two orbits will be derived.

E. Dissertation Organization

The first chapter of this dissertation is an introduction. This chapter presents in brief the fundamentals necessary to understand the developments in the next chapters.

Chapter II presents the solution to the problem of finding the optimal natural orbit for the Ground surveillance missions. It includes the development of the mathematical formulation of the problem. It also describes all the software subroutines developed to solve the problem. Many results and case studies are presented. This chapter was presented as a research paper.

Chapter III addresses the problem of splitting the set of target sites into two subsets. A mathimatical formulation for the problem is presented. An algorithm for splitting is developed in this chapter.

Chapter IV presents the problem of optimal orbit transfer using genetic algorithms. The mathematical formulation is developed. Validation to the technique is presented. Case studies are presented. Comparison to other formulations in literature is also discussed.

Chapter V introduces the concept of Two-way orbits. Analytical derivations are developed to find the conditions for the two-way orbits. A two-way orbit is an orbit which relative trajectory, with respect to the Earth, intersects itself, both branches of the trajectory at the intersection point are parallel. Conditions on the orbital parameters are derived for two-way orbits. If two spacecraft are flying in the two compatible two-way orbits, a condition is derived such that the two spacecraft will visit the intersection point at the same time. All these conditions are once more derived for the case of special two-way orbits. The special two-way orbits are special cases where the intersection point is at the perigee and apogee points of the orbit. This chapter was presented as a research paper.

Chapter VI presents the development of an estimator that uses star tracker measurements to estimate the orbit of a space object. Given a history of observations from a star tracker onboard an observing spacecraft to a space object, Kalman filter and a least squares differential correction methods are developed to estimate the object orbit.

CHAPTER II

ORBIT DESIGN FOR GROUND SURVEILLANCE MISSIONS

A. Introduction

Ground surveillance is the observation from space of a discrete number of sites of interest on the surface of the Earth. This kind of missions require that the satellite visits all the given sites within a given time frame. A site is said to be visited if the nadir point of the satellite passes through a given specified neighborhood of that target. One example is to visit a set of 20 sites on the terrestrial surface within a time frame, t_f, of 50 days. In other applications this time could be $t_f = 3$ days. Ground surveillance missions can be accomplished using active orbit control to allow the satellite to transfer from one target site to another [15]. In the contrary, the aim of this study is to solve the ground surveillance problem in a completely passive way. The goal is to find an orbit such that *all* the assigned sites will be visited within a prescribed time frame. In this case, the consumption of the control fuel is limited to that needed to just compensate the perturbations and not for accomplishing the mission task.

To accomplish this goal, the proposed method performs optimization among all possible orbits to find the best achievable orbit. This optimization is performed through minimizing an error function that quantifies how a particular orbit is far from an ideal solution. This error function is a function of the mission requirements. The optimization is performed through a hybrid method that implements both the Genetic Algorithms technique and a steepest descent technique.

Genetic algorithms have been already adopted to solve orbital mechanics problems. In particular, GAs have been used to find optimal solution for spacecraft rendezvous [17] and for orbit transfer [27] problems. The reason why GAs have been adopted to solve the ground surveillance problem is essentially because the ground surveillance problem is characterized by many local minima. Conventional optimization methods (e.g., gradient/derivative methods) are not suitable for this kind of problems because the final solution highly depends on the initial guess, and to obtain a good initial guess is usually an even more complicate problem. On the other hand, GAs provide global optimization solutions due to their randomized nature.

A local minimum solution is obtained through traditional optimization methodologies using the solution provided by the GA as a starting point. The final solution is then guaranteed to be at a local minimum and globalized through the randomized nature of the GA. The solution obtained by this approach as will be referred to as "optimal" throughout this chapter. Enumerated methods can provide good solutions to the problem. However, the efficiency of these algorithms is very low compared to genetic algorithms.

Two types of missions are considered. Each mission will have a different penalty function that we wish to minimize. The first mission is to get the highest resolution for each target site for a given imaging sensor. In this case, and since the resolution is proportional to the slant range, the objective is then to have the point of closest approach of the satellite to each of the target sites as close as possible to that site. The second mission requires that the observation time for each target site is a maximum. Depending on mission requirements, a combination between these two penalty functions may be used.

This solution presents a fuel-free mission except for the fuel needed to compensate the orbit perturbations. The amount of fuel needed for this purpose is relatively small. A comparison between the fuel required to maintain the orbit for this solution and the fuel required to achieve the mission by maneuvering the spacecraft from one site to another will be presented.

In the case that there is no natural solution satisfying the mission requirements, the multi-orbit solution is proposed. In this solution, the set of target sites is split into two groups. For each group, we use the same method to find the optimal orbit for the group. So we have two orbits in order to visit all sites in the targets set. The satellite will then be maneuvered from one orbit to another during mission operation. It is required then to find the optimal maneuver that achieves the orbit transfer with minimum fuel consumption. GAs is used to search for the optimal orbit transfer as detailed in Chapter IV.

B. Problem Formulation

1. Sites Position Vector

Assume we have n target sites to be visited. Each target is defined by its longitude and latitude, λ_k and ϕ_k, respectively, where $k = 1, \ldots, n$. The position vector for the k^{th} target site in the inertial frame is then

$$\mathbf{r}_k^I = R_{\mathrm{E}} \left\{ \begin{array}{c} \cos\phi_k \, \cos(\lambda_k + \omega_E \, t) \\ \cos\phi_k \, \sin(\lambda_k + \omega_E \, t) \\ \sin\phi_k \end{array} \right\} \tag{2.1}$$

The only variable in the position vector of the kth target site is the Greenwich Sidereal time, t, at which this site will be visited.

2. Satellite Position Vector

The satellite position r^O can be expressed in orbital fixed coordinate system as

$$\mathbf{r}^O = \frac{p}{1 + e\cos\varphi} \left\{ \begin{matrix} \cos\varphi \\ \sin\varphi \\ 0 \end{matrix} \right\} \tag{2.2}$$

where φ is the true anomaly and p is the semilatus rectum. This vector is then transformed to the inertial coordinate system through the transformation matrix, $R^{I/O}$ ($C \equiv \cos$ and $S \equiv \sin$)

$$R^{I/O} = \begin{bmatrix} C_\omega C_\Omega - C_i S_\omega S_\Omega & -S_\omega C_\Omega - C_i S_\Omega C_\omega & S_i S_\Omega \\ C_\omega S_\Omega + C_i S_\omega C_\Omega & -S_\omega S_\Omega + C_i C_\Omega C_\omega & -S_i C_\Omega \\ S_\omega S_i & C_\omega S_i & C_i \end{bmatrix} \tag{2.3}$$

by

$$\mathbf{r}^I = R^{I/O}\,\mathbf{r}^O \tag{2.4}$$

The time in Eq. (2.1) is coupled with the true anomaly in Eq. (2.2) through the Kepler time equation [32]

$$(t_k - t_p)\sqrt{\frac{\mu}{a^3}} = E_k - e\,\sin E_k \tag{2.5}$$

where $k = 1, \ldots, n$, t_p is the perigee time, and E is the eccentric anomaly which is related to the true anomaly by the relationship

$$\tan\left(\frac{E_k}{2}\right) = \sqrt{\frac{1-e}{1+e}}\,\tan\left(\frac{\varphi_k}{2}\right) \tag{2.6}$$

The objective is to minimize a penalty function L, that will depend on the above two position vectors which are, in turn, functions of the six orbital elements a, e, i, ω, Ω, and $\varphi(t)$. Consequently, the penalty function L is a function of a state vector whose elements are the orbital parameters a, e, i, ω, Ω, and all the visiting times t_k. A visiting time t_k is the time of closest approach to the target site k. The elements of the state vector constitute the design space.

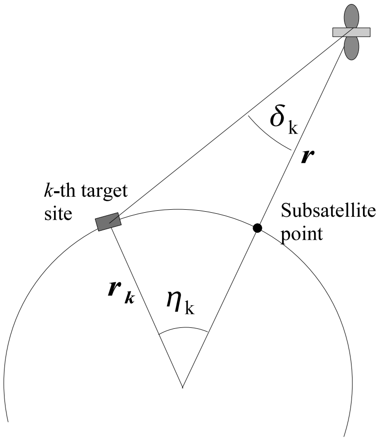

Fig. 4. The angle η_k as defined for the k-th target site

C. Optimality Definition

A target site is characterized by a relative weight which establishes its relative importance with respect to the other targets. Consider a satellite with an observing instrument (radar, camera, etc) with an aperture of $2\vartheta_{\mathrm{FOV}}$. Two cases are considered, one is to maximize the resolution and one is to maximize the observation time, respectively.

For the resolution, a candidate optimality criterion would be to minimize a weighted sum of squares of the distances between each site and the satellite at the nearest ground track point. The penalty function

$$L_R = \sum_k \alpha_k (\mathbf{r}_k - \mathbf{r})^{\mathrm{T}} (\mathbf{r}_k - \mathbf{r}) \tag{2.7}$$

will drive a solution orbit to pass as near as possible to each target site and also have the best achievable resolution since the resolution is proportional to $\|(\mathbf{r}_k - \mathbf{r})\|$.

For the observation time, the penalty function that will be minimized is

$$L_T = \sum_k \alpha_k \, H(\vartheta_{\mathrm{FOV}} - \delta_k) \left(t_f - \int_0^{t_f} \cos\eta_k \, dt \right) \tag{2.8}$$

where η_k is the angle between $\mathbf{r}$ and $\mathbf{r}_k$ (see Figure 4). $H(x)$ is the Heaviside unit step function $[H(x) = 0$ if $x < 0$, $H(x) = 1$ if $x > 0]$, and δ_k is the nadir angle, measured at the satellite from the nadir to the target. The cosine of the angle η can be expanded as follows:

$$\begin{aligned} \cos\eta_k &= \hat{\mathbf{r}}_k^I \cdot \hat{\mathbf{r}}^I = \\ &= \cos\phi_k \, \cos(\lambda_k + omega_E t) \, [\cos(\omega + \varphi_k) \, \cos\Omega - \cos i \, \sin(\omega + \varphi_k) \, \sin\Omega] + \\ &\quad + \cos\phi_k \, \sin(\lambda_k + \omega_E t) \, [\cos(\omega + \varphi_k) \, \sin\Omega + \cos i \, \sin(\omega + \varphi_k) \, \cos\Omega] + \\ &\quad + \sin\phi_k \, \sin(\omega + \varphi_k) \, \sin i \end{aligned} \tag{2.9}$$

The integral in Eq.(2.8) is evaluated numerically for general orbit solutions. However, the search can be directed to look for only near circular orbits, if that is a requirement. This can be achieved by constraining the range for eccentricity to be close to zero. In this case, the integral in Eq.(2.8) can be evaluated analytically.

In the special case of a circular orbit the true anomaly coincides with the mean anomaly

$$\varphi = \sqrt{\frac{\mu}{a^3}} \, (t - t_p) \tag{2.10}$$

Then an analytical expression for the integration in Eq. (2.8) can be derived

$$\int_0^{t_f} \cos\eta \, dt = I_1 + I_2 + I_3 \tag{2.11}$$

where the subscript k has been removed for simple writing, and

$$\begin{aligned}
I_1 &= \int_0^{t_f} \cos\phi\, \cos(\lambda + \omega_E t)\, [\, \cos(\omega + \varphi)\, \cos\Omega - \cos i\, \sin(\omega + \varphi)\, \sin\Omega\,]\, dt = \\
&= \frac{\cos\phi\, \cos\Omega}{2(c_1^2 - c_3^2)} \{(c_1 + c_3)[\sin(t_f c_1 - t_f c_3 + c_2 - c_4) - \sin(c_2 - c_4)] + \\
&\quad +(c_1 - c_3)[\sin(t_f c_1 + t_f c_3 + c_2 + c_4) - \sin(c_2 + c_4)]\} + \\
&\quad + \frac{\cos\phi\, \cos i\, \sin\Omega}{2(c_1^2 - c_3^2)} \{(c_1 + c_3)[\cos(t_f c_1 - t_f c_3 + c_2 - c_4) - \cos(c_2 - c_4)] + \\
&\quad +(c_1 - c_3)[\, \cos(t_f c_1 + t_f c_3 + c_2 + c_4) - \cos(c_2 + c_4)\,]\} \\
I_2 &= \int_0^{t_f} \cos\phi\, \sin(\lambda + \omega_E t)[\, \cos(\omega + \varphi)\, \sin\Omega + \cos i\, \sin(\omega + \varphi)\, \cos\Omega\,]\, dt \\
&= \frac{\cos\phi}{2(c_1^2 - c_3^2)} (\sin\Omega + \cos i \cos\Omega)\, \{(c_1 + c_3)[\, \cos(t_f c_1 - t_f c_3 + c_2 - c_4) \qquad (2.12) \\
&\quad - \cos(c_2 - c_4)\,] + -(c_1 - c_3)[\, \cos(t_f c_1 + t_f c_3 + c_2 + c_4) - \cos(c_2 + c_4)\,]\} \\
I_3 &= \int_0^{t_f} \sin\phi\, \sin i\, \sin(\omega + \varphi)\, dt = \frac{\sin\phi\, \sin i}{c_1} [\cos(\omega + c_2) - \cos(\omega + c_2 + c_1 t_f)]
\end{aligned}$$

where it has been set

$$c_1 = \sqrt{\frac{\mu}{a^3}}, \qquad c_2 = \omega - t_p\, c_1, \qquad c_3 = \omega_E, \qquad \text{and} \qquad c_4 = \lambda$$

D. Genetic Algorithms Implementation

Genetic Algorithms are search algorithms based on the mechanics of natural selection and natural genetics. They combine survival of the fittest among string structures with a structured yet randomized information exchange to form a search algorithm.

In GAs, a design point is called a *member*. A number of members is grouped in a *population*. Generations of this population are created using GA operations like *reproduction*, *crossover*, and *mutation* [12, 10, 18]. Each generation represents one iteration in the searching process. At each generation, members that are fittest are selected in a *parents pool*. The fitness of a member is determined according to the objective, which is constructed based on the penalty function. These parents are then used to create the *new generation*. This process leads to the evolution of populations of individuals that are more fit.

After expanding the expression for the penalty functions, the design variables turn out to be the five orbital elements plus the n visiting times. Each member, i.e. design point, is presented in a binary format as a binary string. This string contains the binary representation of the values of all the design variables at the corresponding design point, as shown in Fig. 5.

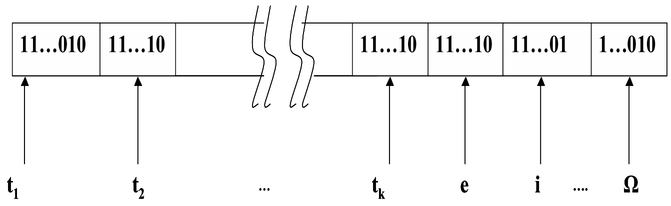

Fig. 5. bit string of a member, t_k is the visiting time of site k

The number of bits allocated for each variable determines its accuracy. The more number of bits, the higher the accuracy, however, the computational cost increases. The number of bits for a variable is selected according to the following inequality:

$$2^b \leq \frac{m}{a} \tag{2.13}$$

Where b is the number of bits representing a variable, m is the maximum value for the variable, and a is the required accuracy for the variable.

Based on this inequality, the number of bits allocated for the visiting time variables is 25, for the orbit angles is 13, and for orbit eccentricity is 7. The population size selection depends mainly on the size of the problem. For the results presented in this chapter, a population size of 100 members is used. The strings for all members in the initial population are selected randomly. The fittest members are selected as parents for the next generation. Genetic algorithm operations are then used to generate a new generation. This process is repeated until a stopping criteria is satisfied. In this problem the stopping criteria is to have the relative improvement of the fittest member in a number of successive generations less than a certain bound.

As a final step, the solution found by the GA is then used as an initial guess in an iterative optimization method to find a local *optimal solution* near the GA solution.

The fitness of each member is evaluated as a sum on all target sites. For the case of a maximum resolution mission, the fitness of a member is evaluated as follows. For each site, the mean anomaly is calculated as a function of the visiting time for this site. The visiting time is a design variable and is generated from the GA operations. The mean anomaly along with eccentricity are used to find the true anomaly by solving Kepler equation, Eq.(2.5) and Eq.(2.6). The true anomaly is then used to calculate the satellite position, Eq.(2.2). The satellite position is then compared to the target site position to calculate the penalty for this site.

1. Genetic Algorithm Parameters

The crossover probability, the mutation probability, and the population size parameters need to be carefully selected. The selection of these parameters is problem dependant [10]. Several run cases showed that a small change in the probability of crossover can cause a big change in the final result. Different references suggests different ranges for these parameters. Coly [10] suggests values for the crossover probability, P_c, of 0.4 to 0.9. He also states that a typical value for the mutation probability, P_m, is 0.001. However, many have used $P_m \approx 1/L$ or $P_m \approx 1/(N\sqrt(L))$ [31], where L is the length of the chromosome and N is the population size.

For the ground surveillance problem, in each case several values for the parameters are tested to find the best values for these parameters within the above mentioned ranges. For example, the case of 5 target sites has an optimal value for the crossover probability of 0.95. Fig. 6 show the variation of the optimal cost function with the crossover probability P_c. This results were generated with a population size of 200. The number of days for the mission is 3 days. The maximum number of generations is 100.

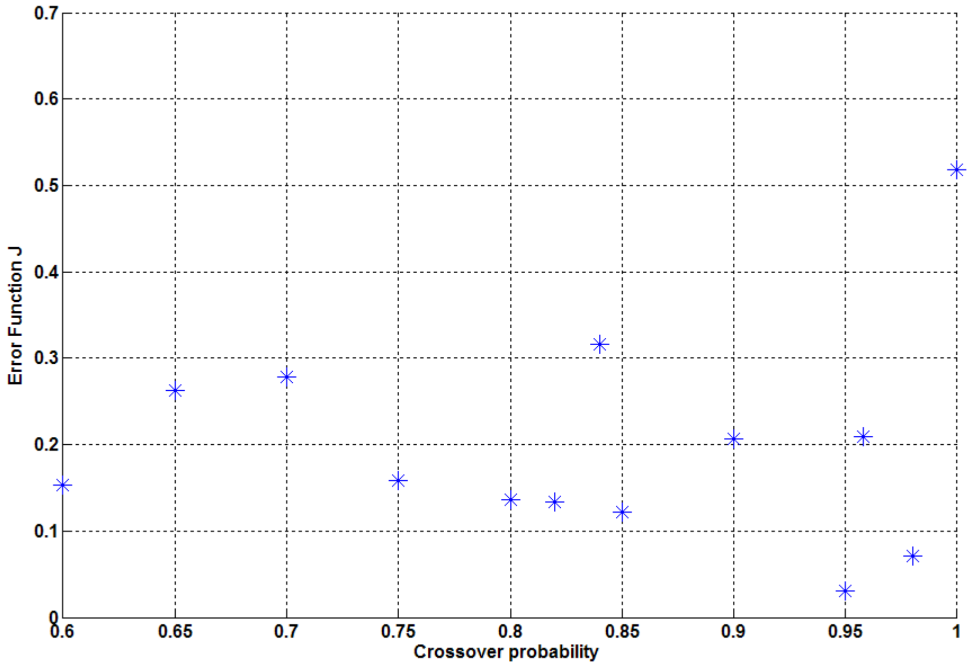

Fig. 6. Variation of the cost function with the crossover probability

E. Results

To demonstrate how good the algorithm is, the simple case of having only two target sites is examined. The mission duration time is 3 days in this example. The penalty function used is the maximum resolution mission penalty function, L_R. L_R is a non-dimensional measure for how far the solution is from an ideal solution. The expected optimal solution should then be an orbit whose ground track is as close as possible to the target sites. The semi-major axis, a, is not included in the design variables. Because resolution is proportional to slant range, the algorithm would find orbits too close to the ground if a was a design variable. The semi-major axis is set as an input parameter.

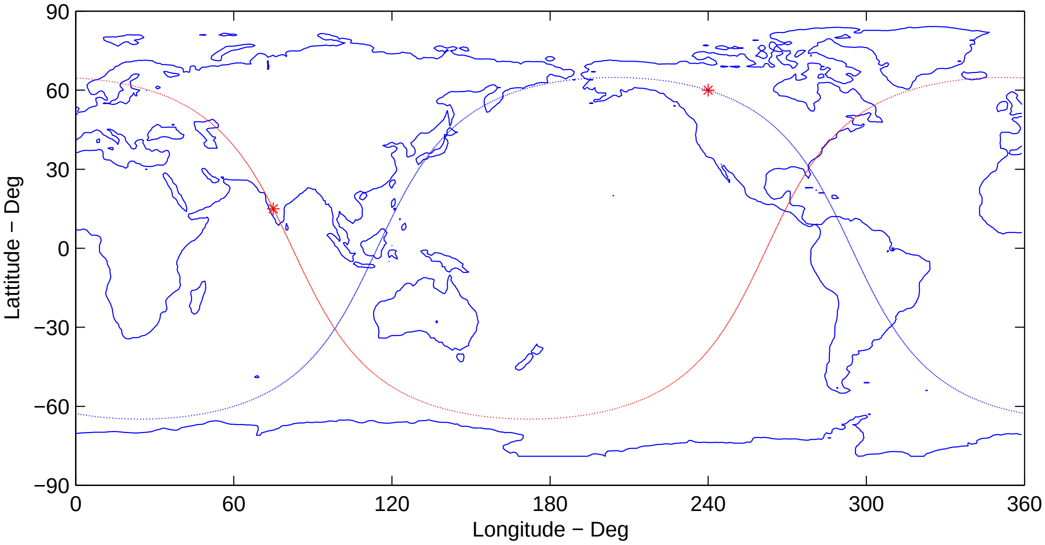

Fig. 7. Optimal orbit ($n = 2$ sites , $t_f = 3$ days)

Figure 7 shows the ground track of the optimal orbit for that case. The orbit parameters are $a = 7,753.5$ Km, $e = 0.7$, $i = 40°$, $\omega = 330°$, and $\Omega = 362°$, the penalty function, at convergence, reached the value of $L_R = 2.58 \cdot 10^{-9}$. As can be seen from the figure, the error is almost zero and the two sites are nearly perfectly visited. The penalty function is almost zero. A more complicated case of 15 target sites visited within 3 days is shown in Fig. 8.

For the case of 15 target sites, the penalty function value did not go to zero. The mission requirements, however, determines whether the resulting errors are acceptable

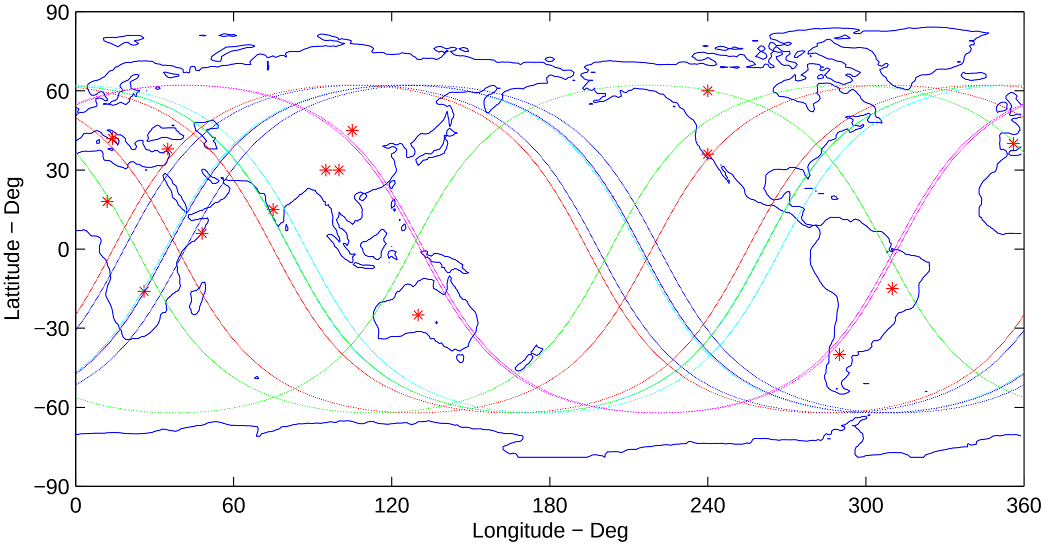

Fig. 8. Optimal orbit ($n = 15$ sites, $t_f = 3$ days)

or not. The orbital parameters for the solution orbit are $a = 7,753.5$ km, $e = 0.034$, $i = 76$ deg, $\omega = 207$ deg, and $\Omega = 275$ deg. Figures 9 and 10 show the results for a case study of 20 target sites with a mission duration of 3 days.

The solution presented for this case is achieved after 150 generations (iterations). The highest resolution is also required in this mission. The solution orbit semi-major axis is set to 6892.8 km. The resulting orbit parameters are $e = 0.045$, $i = 40.8$ deg, $\omega = 145.7$ deg, $\Omega = 86$ deg. The visiting times at which each target site is visited are calculated. Here we present this by indicating the index of the satellite revolution in which the site will be visited. The visiting orbit indices are shown in Table III.

The solution resulted from the GA has a penalty function of $L_R = 16$. The Guess-Newton optimization algorithm is used to refine the solution. The solution provided by the GA is used as initial starting point. The final value for the penalty function after refinement is $L_R = 3.23$. The solution resulted from the GA is shown in Fig. 10. Also the occurrences of the orbital elements in the last population as well as the history of convergence are shown. Assume that the resulting non-dimensional

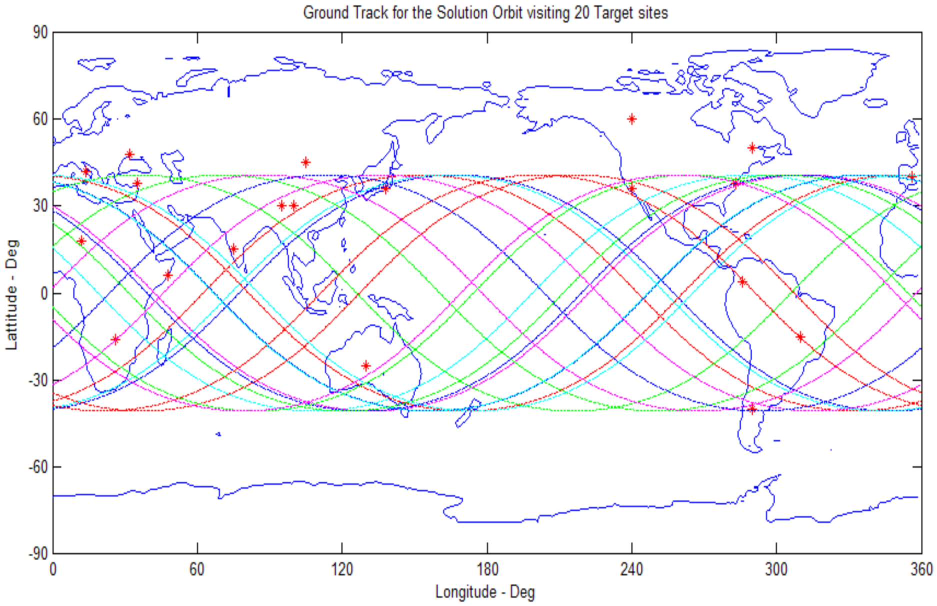

Fig. 9. High resolution mission ($n = 20$ target sites, $t_f = 3$ days)

error, $L_R = 3.23$, is not satisfactory for a particular mission. The solution is to split the set of target sites into two groups. Each group includes ten target sites. Because the satellite will be transferred from one orbit to the other, and because the out of plane maneuvers are expensive, constraints are added on the inclination and the right ascension of the ascending node (RAAN) of the orbit of the second group. The second orbit inclination and RAAN should be very close to those of the first orbit. This is achieved by limiting the design space of both angles to small ranges around those of the first orbit angles. As a result, the first group has a penalty function value of $L_R = 0.61$. The second group has a penalty function value of $L_R = 1.17$. The orbit of the first group has the following parameters: $a = 6892.8km$, $e = 0.02$, $i = 127.5^o$, $\omega = 8.9^o$, and $\Omega = 36.6^o$. The orbit of the second group has the following parameters: $a = 6892.8km$, $e = 0.01$, $i = 127.55^o$, $\omega = 248.4^o$, and $\Omega = 36.6^o$. Each orbit is required to visit all the target sites in its group in two days. The resulting two orbits have equal RAANs and very close inclinations.

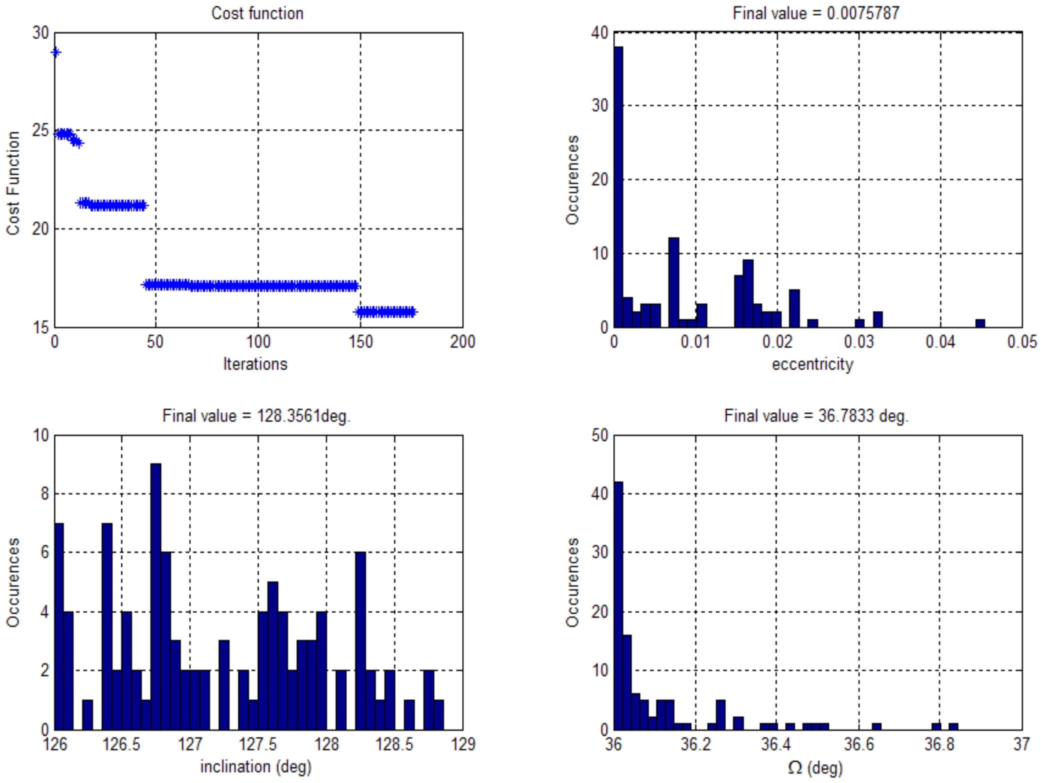

Fig. 10. High resolution mission (min L_R, $n = 20$ target sites)

Figures 11 and 12 show the results for the second optimality definition, L_T, where the objective is to have maximum observation time. The mission duration is also 3 days. Figure 11 shows the solution for the case of 20 target sites. The resulting orbit parameters are $a = 6,886.6$ Km, $e = 0.02$, $i = 56.1°$, $\omega = 0°$, $\Omega = 247.6°$. The visiting orbit indices are shown in Table IV. Figure 12 shows the results of the GA iterations.

The range for each design variable is controlled. The eccentricity is limited to be higher than zero and less than a maximum value e_{max}:

$$e_{max} = 1 - \frac{r_p}{a} \tag{2.14}$$

It can be limited to even lower values if high eccentric orbits are not required.

Table III. Observed orbits sequence ($\min L_R$ and $n = 20$)

Target	1	2	3	4	5	6	7	8	9	10
Orbit	9	28	6	23	3	17	19	28	15	1
Target	11	12	13	14	15	16	17	18	19	20
Orbit	25	30	37	38	29	15	38	20	9	15

Table IV. Observed orbits sequence ($\min L_T$ and $n = 20$)

Target	1	2	3	4	5	6	7	8	9	10
Orbit	8	3	1	34	1	37	38	1	37	35
Target	11	12	13	14	15	16	17	18	19	20
Orbit	1	1	1	1	1	18	1	5	15	35

Disturbances like the aerodynamic drag and solar radiation pressure cause the orbit to decay. The resulting orbit solution must be maintained during the mission period to cancel the effect of the disturbances. Assume electric propulsion is used for the orbit maintenance of the orbit obtained for the case of 20 target sites and duration time of 3 days. Algorithms for orbit maintenance and estimates for the required thrust and fuel mass using electric propulsion are available in the literature [1]. These algorithms are used to estimate the amount of fuel needed for orbit maintenance in this case. Assume the thruster specific impulse is 6000 sec and the spacecraft mass is 200 kg, then the amount of fuel needed for orbit maintenance in this case is on the order of 0.25 g/day. The maximum thrust needed is in the order of 0.3 mN. The thruster will not be working continuously all the time but rather with a duty cycle that ranges from 47% to 85%. On the other hand, if using the electric propulsion to maneuver the spacecraft continuously between sites, estimates [15] show that, for the case of 20 target sites and duration time of 50 days, the amount of fuel needed is 5 g/day. The maximum thrust needed is in the order of 1 mm/s^2. The thruster is working continuously all the time.

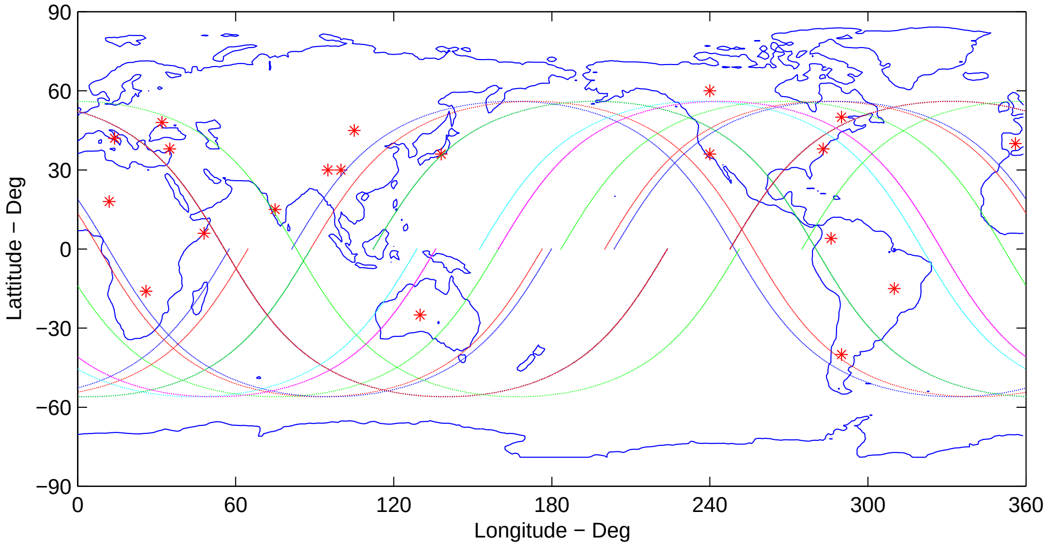

Fig. 11. Optimal orbit (min L_T and $n = 20$ sites)

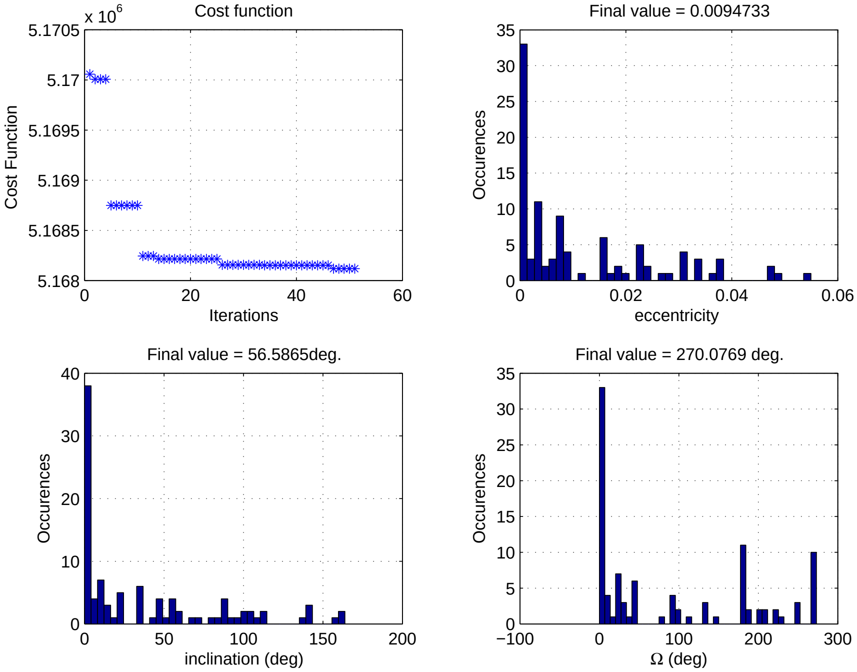

Fig. 12. Optimal orbit (min L_T, $n = 20$ sites)

F. Conclusions

The problem of the orbit design for optimal ground surveillance is considered. The GA method along with a steepest descent algorithm are used for optimization. The use of GA requires the determination of many genetic parameters. These parameters were determined using parametric analysis performed for each particular problem. The problem formulation has a big effect on the solutions obtained by the GA. The problem formulation is the set up of the member structure and the fitness function. Other formulations for the ground surveillance problem were investigated, the current formulation demonstrated to be the most efficient. However, no guarantee that it is the best possible formulation.

CHAPTER III

MULTI ORBIT SOLUTION TO THE GROUND SURVEILLANCE PROBLEM

Chapter (II) describes a method to find a natural orbit solution to the ground surveillance mission. This solution may or may not satisfy the mission requirements, in terms of the error associated with each site. In the case that there is no single orbit satisfying the mission requirements, a multi-orbit solution is proposed. The set of target sites is split into two subsets. For each subset, we find the orbit solution associated with this subset. The mission is completed by maneuvering the satellite from one orbit to another. Two subproblems need to be solved. The first is to develop an algorithm to split the set of targets into two subsets, this is the subject of this chapter. The second subproblem is to develop an algorithm for optimal orbit maneuver, this is the subject of chapter (IV).

The split can be done in many ways resulting in different subsets combinations. The optimal split is defined as the split that results in two subsets, the penalty function for them are the least that can be achieved among all possible splitting solutions. The first section presents a mathematical formulation for the problem. Next, an algorithm for splitting is introduced.

A. Mathematical Formulation

Let n be the number of target sites in a set S. Let the sites be indexed from 1 to n. Introduce a binary variable x_{ij} that is related to the two sites, i and j. x_{ij} is defined as follows:

$$x_{ij} = \begin{cases} 1 & \text{if } i \text{ and } j \text{ are in the same set} \\ 0 & \text{if } i \text{ and } j \text{ are not in the same set} \end{cases}$$

The range for i:

$$1 \leq i \leq n-1 \tag{3.1}$$

Noting that $x_{ii} = 1$ and $x_{ij} = x_{ji}$, then we can reduce the range of j to be:

$$i+1 \leq j \leq n \tag{3.2}$$

Specifying the variables x_{ij} for all values of i and j completely describes the splitting. As a simple example, consider the case of $n = 3$. Assume that x_{ij} are specified as follows:

$$x_{12} = 0, \quad x_{13} = 1, \quad x_{23} = 0 \tag{3.3}$$

This specification for x_{ij} tells us that we have two subsets, the first includes the sites 1 and 3 and the second subset includes the site 2. So we will call this specification a splitting. Recall that we are concerned only with splitting the set S into two subsets only, S_1 and S_2. So, in the above splitting, if the value for x_{13} was 0, then that would give us a splitting for S into 3 subsets each includes a site. So, The above splitting with $x_{13} = 0$ is not accepted. This condition can be written as follows:

$$If \qquad x_{jk} = 0 \qquad then \qquad x_{ji} \neq x_{ki} \tag{3.4}$$

The condition (3.4) guarantees that the set S will be split into two subsets only. Consider again the splitting (3.3), if the value of x_{23} was 1 rather than 0, this means that site 3 exists in both subsets. This can be accepted since it means a site will be visited by both orbits. However, since we know that adding a site to a given set of target sites results in an error function that is greater than or equal to the error function before addition, then this case should be excluded. In fact, the condition (3.4) implies also this new condition. It states that one site should not exist in two subsets and no more than two subsets is allowed.

So, the optimal splitting problem can be stated as follows: Given a set of target sites S, find the splitting x_{ij} that splits S to S_1 and S_2, such that the penalty functions, L_1 for S_1 and L_2 for S_2, are the minimum penalty functions among all possible splittings, subject to the constraint (3.4).

B. Splitting Algorithm

In this section, an algorithm to perform the splitting is introduced. The algorithm does not guarantee that the splitting is optimal, however, a satisfactory solution is achieved. Let the length, n_i, of a subset S_i be the number of sites in this subset. Two subsets S_1 and S_2 are initialized randomly with small number of sites, p, in each of them, selected from the main set S. Then, the rest of sites in S is distributed on S_1 and S_2. The solution orbits for the initial subsets are calculated. Then each of the remaining sites in S is added to either S_1 or S_2 according to whether it is closer to the solution orbit of S_1 or S_2. This process is repeated with different initializing for S_1 and S_2 until a satisfactory solution is achieved. This algorithm implements the GA method to find the orbit solution for the initial subsets only.

Consider the maximum resolution mission. Recall that the penalty function represents the sum of squares of errors on all the target sites from the orbit. So, the penalty function is always a positive scalar. Consider a subset S_1 that has a penalty function L_1, adding a new target site to this subset results in a new value for the penalty function that is greater than or equal to the current value for L_1. So, decreasing the number of targets in a subset leads to a lower penalty function for this subset. However, the other subset will have a higher number of targets. So, we will constraint the search to pairs of subsets which difference in length is less than 2 sites.

Given a set of n target sites S. It is required to split it into two target subsets S_1 and S_2 with penalty functions L_1 and L_2 respectively. Assume that the maximum size for any of the two subsets is n_m; where n_m is the least integer satisfying the condition:

$$n_m \geq \frac{n+1}{2} \tag{3.5}$$

The algorithm starts by giving a random indexing for all targets in S.

$$S = \{S(1), S(2), \ldots, S(n)\} \tag{3.6}$$

The first p targets in S are selected and added to subset S_1. The next p targets are selected and added to subset S_2:

$$\begin{aligned} S_1 &= \{S(1), S(2), \ldots, S(p)\} \\ S_2 &= \{S(p+1), S(p+2), \ldots, S(2p)\} \end{aligned} \tag{3.7}$$

The solution orbit for each subset is calculated along with the corresponding penalty functions L_1 and L_2. The number of initially selected targets, p, is a parameter that can be varied. From the results that were presented in the chapter II, the developed method for optimal ground surveillance can find a solution that is almost optimal if the number of sites is 2; the penalty function for this case is $L_R = 2.6 \times 10^{-9}$, see Fig. 7. If the number of sites is 3, the penalty function is small, $L_R = 0.00061$. These results motivate the selection of p to be 2 or 3. The initial penalty function for each subset will be almost zero. The next site in S is selected and added to both S_1 and S_2, to form ${S_1}'$ and ${S_2}'$. The new penalty functions, L_1' and L_2', are calculated. The calculation of L_1' and L_2' does not require the calculation of a new orbit solution but rather calculating the error between the new site and the current orbit solution. The increase in each of the penalty functions is:

$$\triangle L_1 = L_1' - L_1 \tag{3.8}$$

$$\triangle L_2 = L_2' - L_2 \tag{3.9}$$

Let k be an index such that $1 + 2p < k < n$. The new site, $S(k)$ is added to S_1 if

$$\triangle L_1 < \triangle L_2 \tag{3.10}$$

and is added to S_2 if

$$\triangle L_2 < \triangle L_1 \tag{3.11}$$

If both $\triangle L_1$ and $\triangle L_2$ are equal, it will be added to the subset with less number of sites in it. The process of selecting the next site from S and adding it to one of the two subsets repeats until all sites in S are selected. The resulting S_1 and S_2 are stored and the whole process is repeated again with a new random indexing for S. The process repeats until a satisfactory solution is achieved. For each indexing of S, the GA method is used only one time for each set. Not all possible subsets are checked but only those which sites are closer to a particular orbit, namely the solution orbits of the initial subsets. In the limit when all possible indexing orders are checked, the algorithm is equivalent to a numerical search. A flow chart for this algorithm is presented in Fig 13.

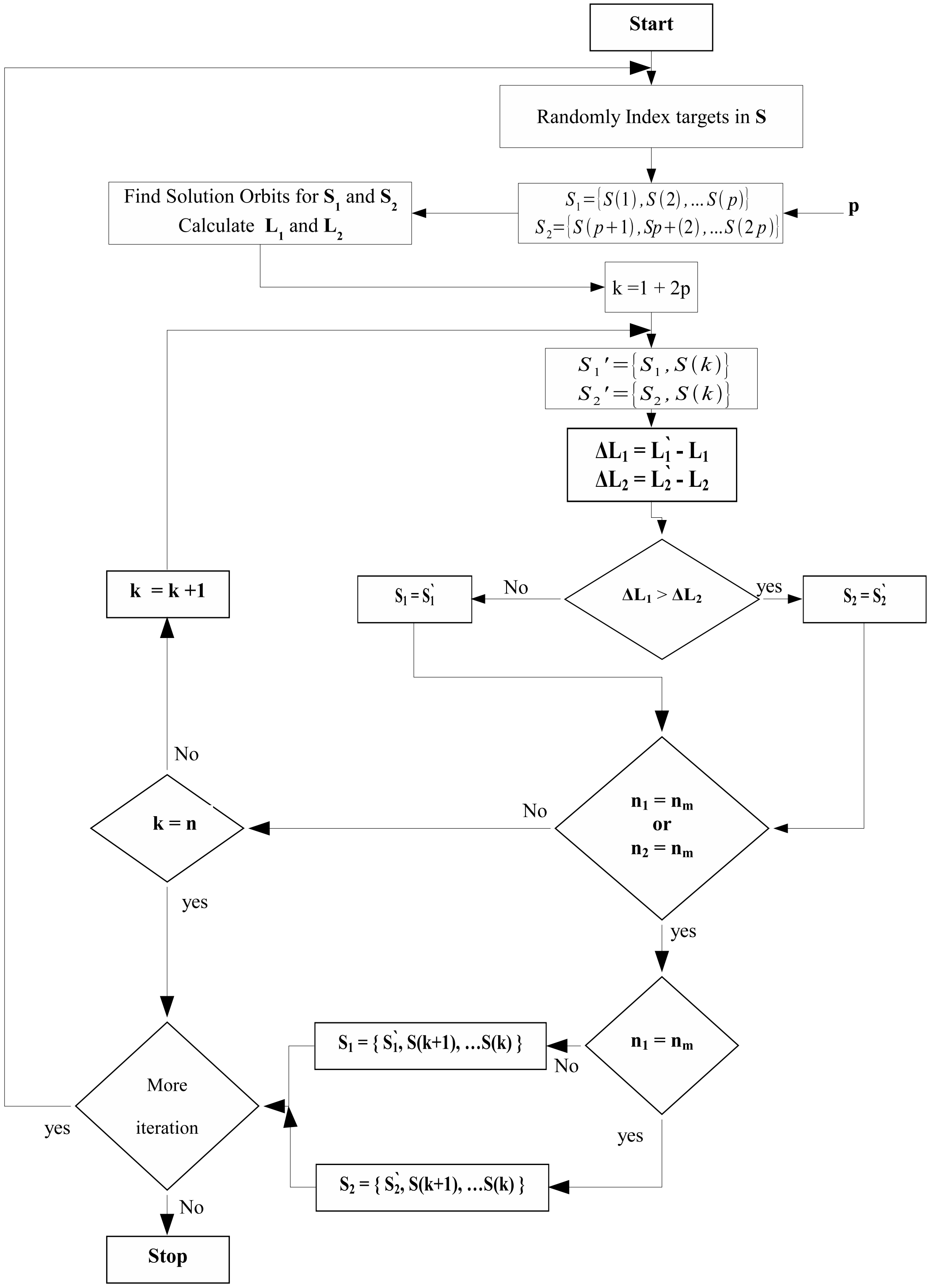
Start
Randomly Index targets in **S**
Find Solution Orbits for $\mathbf{S_1}$ and $\mathbf{S_2}$
Calculate $\mathbf{L_1}$ and $\mathbf{L_2}$
$S_1=\{S(1),S(2),\ldots S(p)\}$
$S_2=\{S(p+1),Sp+(2),\ldots S(2p)\}$
p
k =1 + 2p
$S_1'=\{S_1,S(k)\}$
$S_2'=\{S_2,S(k)\}$
$\Delta L_1 = L_1` - L_1$
$\Delta L_2 = L_2` - L_2$
No
$\Delta L_1 > \Delta L_2$
yes
$S_1 = S_1`$
$S_2 = S_2`$
k = k +1
No
k = n
No
$n_1 = n_m$
or
$n_2 = n_m$
yes
yes
$n_1 = n_m$
$S_1 = \{ S_1`, S(k+1), \ldots S(k) \}$
No
yes
More
iteration
yes
$S_2 = \{ S_2`, S(k+1), \ldots S(k) \}$
No
Stop

Fig. 13. Splitting algorithm flow chart

CHAPTER IV

OPTIMAL ORBIT MANEUVERS USING GENETIC ALGORITHMS

A. Introduction

The solution to the ground surveillance problem may not be achieved using a single orbit. In this case, the set of ground sites is split into two groups. Each group will be considered independently. A natural orbit solution for each group is calculated. An orbit transfer between the solution orbits is required. The problem of finding the optimal orbit transfer using genetic algorithms is addressed in this chapter; optimum in the sense of minimum fuel requirements.

McCue [19] solved the problem of finding the optimal two-impulse orbit transfer. His solution is a combination between numerical search and steepest descent optimization procedures. Numerical search is first used to present large amounts of optimum impulse information in a concise contour form. Initial conditions taken from the contour maps are used to find the local optimum. Although the steepest descent algorithm converges very fast due to the good initial conditions, the numerical search technique used to generate the impulse function contours has a higher computational cost.

The problem of finding the optimal orbit transfer using GAs is addressed in the literature. Kim [17] formulated this problem by constructing a member, or a chromosome, from six design variables in the co-planar orbits case. The six design variables are the true anomaly at the departure and arrival points, the required thrust magnitudes at the departure and arrival points, and the thrust direction at the departure and arrival points. This formulation can be easily generalized to a general non-coplanar case. The number of design variables will be eight in this case. This formulation produces optimal solutions to the problem, however, the satellite is not guaranteed to arrive exactly at the final orbit but rather close to it. Reichert [28] also addressed this problem for co-planar orbits only. He formulated the problem by constructing a member from three design variables. These design variables are the eccentricity, semi-latus rectum, and orientation angle of the transfer orbit. The required thrust to transfer the satellite is then calculated from the information of the transfer orbit, initial orbit, and final orbit. This formulation can be generalized to non-coplanar orbits, however, the number of design variables will increase. The solution in this case also does not guarantee that the satellite will reach exactly the required final orbit.

In this chapter, a new formulation to the problem is introduced. This formulation requires only three design variables for the general case of non-coplanar orbits. The solution obtained by this method is guaranteed to put the satellite on the final orbit exactly even if the GAs did not converge to the optimal solution. The design variables in this formulation are: the true anomaly at the departure and arrival points, and the time of flight on the transfer orbit from the initial position to the final position. This formulation requires solving Lambert problem to find the parameters of the transfer orbit for a given set of the three design variables. Lambert problem is reviewed in brief in this chapter. Validation to this formulation is performed by solving several case studies to which the optimal solution is known. This method is used to complete the solution for the Ground surveillance problem in the case of Multi-Orbit Solution.

1. Lambert Problem

Lambert was the first to form the solution to the problem of determining the orbit given two position vectors and the time of flight between them in 1761. Lambert's original problem was to find the orbital transfer time between two vector positions.

Lambert's theorem states that *the orbital transfer time depends only upon the semi-major axis, the sum of distances of the initial and final points of the arc from the center of force, and the length of the chord joining these points.* The proof of Lambert's theorem is detailed in Ref. [3], and is briefed as follows: Let $(t_2 - t_1)$ be the orbital transfer time, then Lambert's theorem states that:

$$\sqrt{\mu}(t_2 - t_1) = F(a, r_1 + r_2, c) \tag{4.1}$$

The theorem is applicable to any conic orbit, however in this section we will consider only elliptical orbits since the surveillance problem is restricted to only elliptical orbits.

Let E_1 and E_2 be the respective eccentric anomalies associated with the two position vectors $\vec{r_1}$ and $\vec{r_2}$. Then, from Kepler's equation, the time to traverse the arc from $\vec{r_1}$ to $\vec{r_2}$ is:

$$\sqrt{\mu}(t_2 - t_1) = 2a^{\frac{3}{2}}[\frac{1}{2}(E_2 - E_1) - e\sin(\frac{1}{2}(E_2 - E_1))\cos(\frac{1}{2}(E_2 + E_1))] \tag{4.2}$$

Define ψ and ϕ by:

$$\psi = \frac{1}{2}(E_2 - E_1) \qquad \cos\phi = e\cos(\frac{1}{2}(E_2 + E_1)) \tag{4.3}$$

Then the Kepler's equation becomes:

$$\sqrt{\mu}(t_2 - t_1) = 2a^{\frac{3}{2}}(\psi - \sin\psi\cos\phi) \tag{4.4}$$

The sum, $r_1 + r_2$, can be calculated from the equation of orbit [3],

$$r_1 + r_2 = 2a(1 - \cos\psi\cos\phi) \tag{4.5}$$

The cord, c, can also be expressed in terms of ψ and ϕ [3]:

$$c = 2a \sin\psi \sin\phi \tag{4.6}$$

From Eq.(4.5) and Eq.(4.6), ψ and ϕ can be expressed as functions of a, $r_1 + r_2$, and c. Substituting ψ and ϕ into Eq.(4.4), we get the mathematical expression for Lambert's theorem.

Lambert's problem can be stated as follows: given two position vectors $\vec{r_1}$ and $\vec{r_2}$ and the time of flight between them, find the transfer orbit for the spacecraft. Methods to Solve the Lambert's problem are available in the literature. One Method is the Universal Variable Method and is presented in Ref. [34]. The method evaluates the two velocity vectors, $\vec{v_1}$ and $\vec{v_2}$, to completely specify the transfer orbit.

$$\vec{v_1} = \frac{\vec{r_2} - f\vec{r_1}}{g} \qquad \vec{v_2} = \frac{\dot{g}\vec{r_2} - \vec{r_1}}{g} \tag{4.7}$$

where,

$$f = 1 - \frac{y_n}{r_1} \qquad \dot{g} = 1 - \frac{y_n}{r_2} \qquad g = A\sqrt{\frac{y_n}{\mu}} \tag{4.8}$$

and, $A = t_m\sqrt{r_1 r_2 (1 + \cos(\Delta\nu))}$, t_m is the transfer time, and y_n is a variable that is evaluated numerically after a loop converges to a solution.

B. Orbit Maneuver Algorithm

Consider a satellite in an initial orbit defined by the five orbital elements a_I, e_I, i_I, ω_I, and Ω_I. The final orbit is defined by the five orbital elements: a_F, e_F, i_F, ω_F, and Ω_F. Assume that the satellite is subject only to the Newtonian force from the central body. The true anomaly on the initial orbit at the time of satellite departure from the initial orbit is θ_I. The true anomaly on the final orbit at the time of satellite arrival to the final orbit is θ_F. The geometry for the orbit transfer is shown if Fig. 14.

Both θ_I and θ_F are design variables for which we need to find the optimal values that minimize the fuel consumption. For a given value of θ_I, we can calculate the position and velocity of the satellite on the initial orbit, $\vec{r}_I$ and $\vec{v}_I$ respectively as detailed in section (I.2). Similarly, for a given value of θ_F, we can calculate the position and velocity of the satellite on the final orbit, $\vec{r}_F$ and $\vec{v}_F$ respectively. The time of flight, t_f, on the transfer orbit from the initial position to the final position is a design variable. For given values for $\vec{r}_I$, $\vec{r}_F$, and t_f, we can solve the Lambert's orbital two point boundary value problem [3] to find the transfer orbit parameters. From the transfer orbit parameters, we can calculate the velocities on the transfer orbit at the initial and final positions, $\vec{v}_{It}$ and $\vec{v}_{Ft}$ respectively. The required $\triangle v$ can then be calculated as follow:

$$\triangle v = \|\vec{v}_I - \vec{v}_{It}\| + \|\vec{v}_F - \vec{v}_{Ft}\| \tag{4.9}$$

The problem formulation is, therefore, as follows. Given the initial orbit and the final orbit, find values for the time of flight t_f, the departure true anomaly θ_I, and the arrival true anomaly θ_F, such that the total $\triangle v$ is a minimum. A genetic Algorithm is used to search for the optimal values of t_f, θ_F, and θ_I.

1. Validation

In order to check the validity of using GAs to find a solution for the orbit transfer problem, some test cases are considered.

Case 1: Simple Coplanar Orbit Change

The first case tested is a simple case of transferring a spacecraft from a circular orbit of altitude $300Km$ to an elliptic orbit of perigee altitude equal to $300Km$, and an

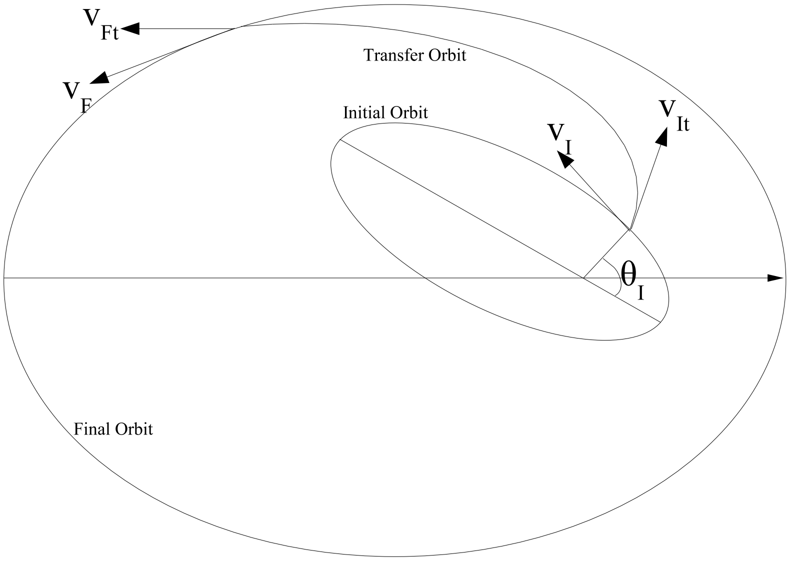

Fig. 14. Geometry for orbit transfer

apogee altitude of $3000Km$. Although the above algorithm was designed for the case of two impulse maneuvers, this example demonstrate that it can be used for the simple cases of one impulse maneuvers. The exact solution to this case can be easily calculated. Velocity on the initial circular orbit is:

$$v_I = \sqrt{\frac{\mu}{r}} = 7.726 \quad Km/sec \tag{4.10}$$

Velocity on the final orbit at the periapsis is:

$$v_F = \sqrt{\frac{2\mu}{r} - \frac{\mu}{a}} = 8.35 \quad Km/sec \tag{4.11}$$

So, the required increase in velocity is:

$$\Delta v = v_F - v_I = 0.624 \quad Km/sec \tag{4.12}$$

This Δv should be applied at the perigee of the final orbit.

Consider applying the developed GA algorithm to this problem. Since, the developed algorithm is designed for two impulse maneuver problems, the expected optimal solution should be as follows. The optimal solution should consist of two impulses, both at the perigee of the final orbit, and the time gab between them is the orbital period of the final orbit. The sum of the two impulses should equal to the value calculated in Eq.(4.12).

The optimal solution provided by the GA is shown in Figures 15 and 16. The spacecraft is transferred exactly to the final orbit with zero error. The first impulse is fired at a true anomaly angle of 0.038^o (the correct value is 0^o). The second impulse is fired at a true anomaly angle of 357.25^o (the correct value is 360^o). The total impulse as calculated from the GA solution is 0.62436 Km/sec, and the optimal value as calculated above is 0.624 Km/sec. The solution provided by the GA is very close to the optimal solution. The error in the final Δv is 0.57%. The convergence is very fast. From Fig. 16, the final solution was achieved after the third iteration.

Case 2: General Orbit Plane Change

Consider the case of transferring a spacecraft between two orbits that are not in the same plane. In general, non-coplanar orbits have different inclinations, Longitudes of ascending nodes, or both. In this case study, the general case of both angles are different is considered.

Consider an initial circular Earth orbit of altitude 275 Km, inclination 28.5^o, and $\Omega = 60^o$. The final orbit is also a circular orbit with altitude altitude 275 Km, inclination 10^o, and $\Omega = 100^o$. The analytical solution to this problem [4] is as follows. The intersection point of the two orbits is located at an anomaly angle of 162.45^o as measured along the initial orbit from its node point, and at an angle of 124.1^o as measured along the final orbit from its node point. These two angles correspond to

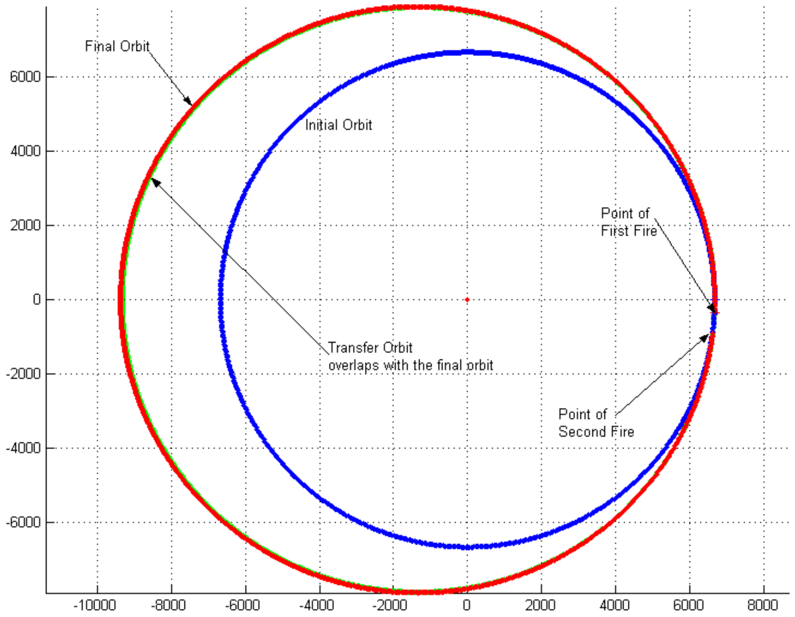

Fig. 15. Single imulse maneuver: initial, final, and transfer orbits and fire points

the departure and arrival true anomalies. At the intersection point, the angle between the two orbits is 21.73^o. The required velocity change is $\Delta v = 2.918 Km/sec$. This is also a single impulse maneuver. The optimal solution provided by the developed GA tool is expected to have two fires coincide on the same location and the time gap between them is equal to the period of the final orbit. The solution provided by the developed GA tool is presented in Fig. 17 and Fig. 18. As seen from Fig. 18, two impulses are fired, the total required velocity change is 2.9181 Km/sec. This value corresponds to the true optimal value of 2.918 Km/sec with an error of 0.0034%. The first impulse is fired at a departure true anomaly of 162.588^o, corresponding to 162.45^o. The second impulse is fired at an arrival true anomaly of 123.407^o, corresponding to 124.1^o. The time between the two impulses is 1.4552 hr which is 2.88 $min.$ less than the orbital period of the final orbit. So, it can be concluded that the developed GA

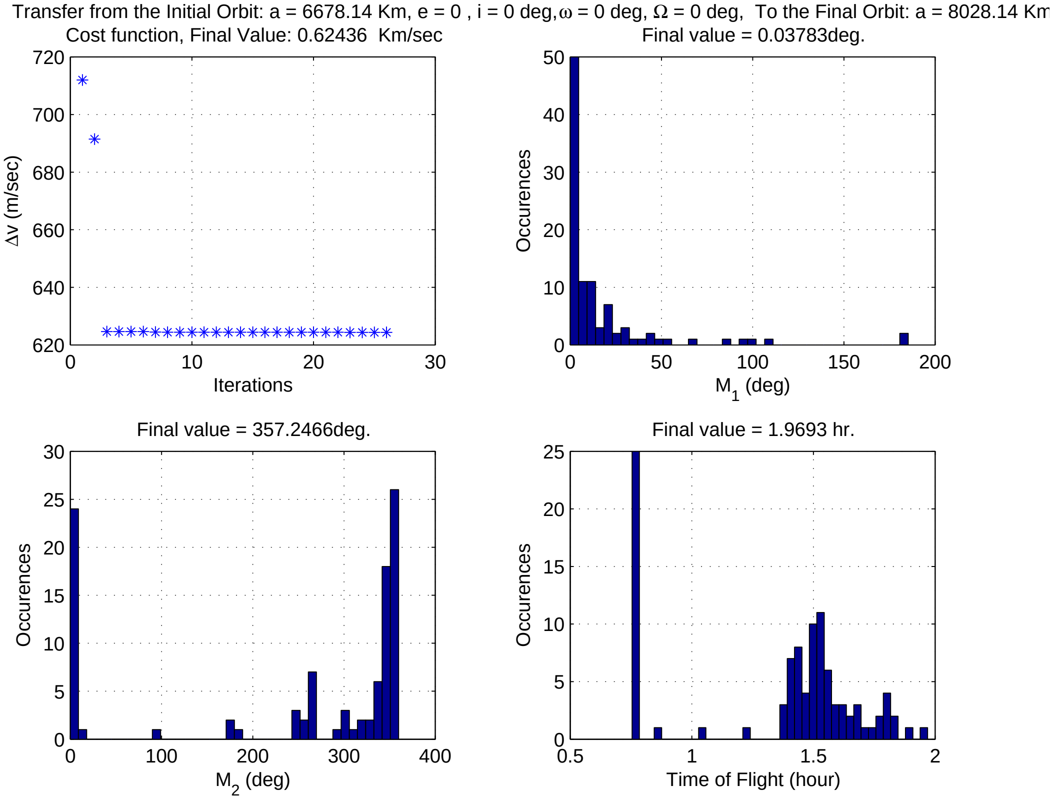

Fig. 16. Single imulse maneuver, design variables and cost function convergence

tool works well very the case of a general orbital plane change.

Case 3: The Hohmann Transfer

The Hohmann transfer maneuver is considered. In this case, both initial and final orbits are circular. The optimal solution to the Hohmann transfer problem is a two impulse maneuver. It is characterized by that the point of departure from the initial orbit, the point of arrival on the final orbit, and the center of the central body are all aligned. This can be written as: $\theta_F = \theta_I + \pi$. As a case study, consider the transfer from a circular Mars orbit of radius 8000 Km to a circular Mars orbit of radius 15000 Km. The optimal solution is the Hohmann transfer with a required total velocity

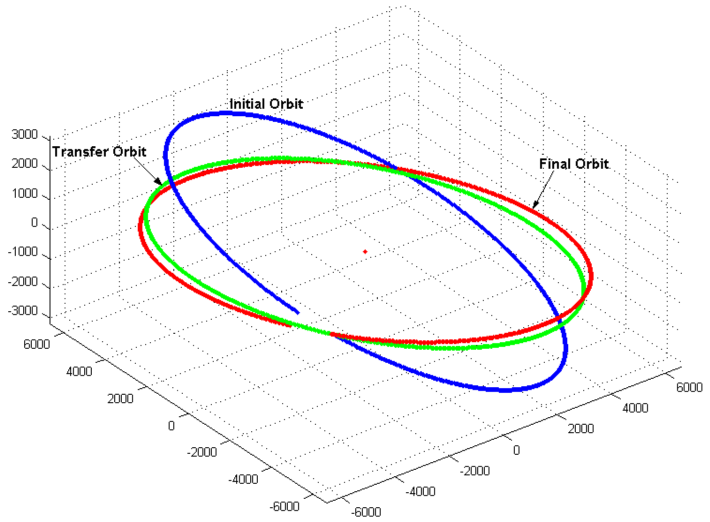

Fig. 17. General plane change: initial, final, and transfer orbits

change of 0.609 Km/sec. The time required for transfer is 5.08 hr [4].

The solution provided by the GA is shown in Fig. 19 and Fig. 20. Figure 19 shows a plot for the transfer orbit. The previously mentioned three points are almost aligned which implies that this is the optimal solution. Figure 20 shows the number of iterations and the final values for θ_I, θ_F, and t_f and their occurrences in the final generation. In Fig. 20, M_1 is θ_I and M_2 is θ_F. The required total velocity change is 0.60928 Km/sec. This is compared to the above optimal value of 0.609 Km/sec. The error is 0.045%. The true anomaly at departure is 127.85^o. The true anomaly at arrival is 306.73^o. The difference between them is 178.88^o, compared to the exact optimal value of 180^o. The calculated transfer time is 5.157 hr, compared to the exact optimal value of 5.08 hr. The satellite is transferred exactly to the final orbit with no error. The slight drift from the optimal solution appears as a slightly higher transfer time and consequently a slightly higher total Δv. The method converged to the optimal solution after about 20 iterations. This fast convergence can be explained

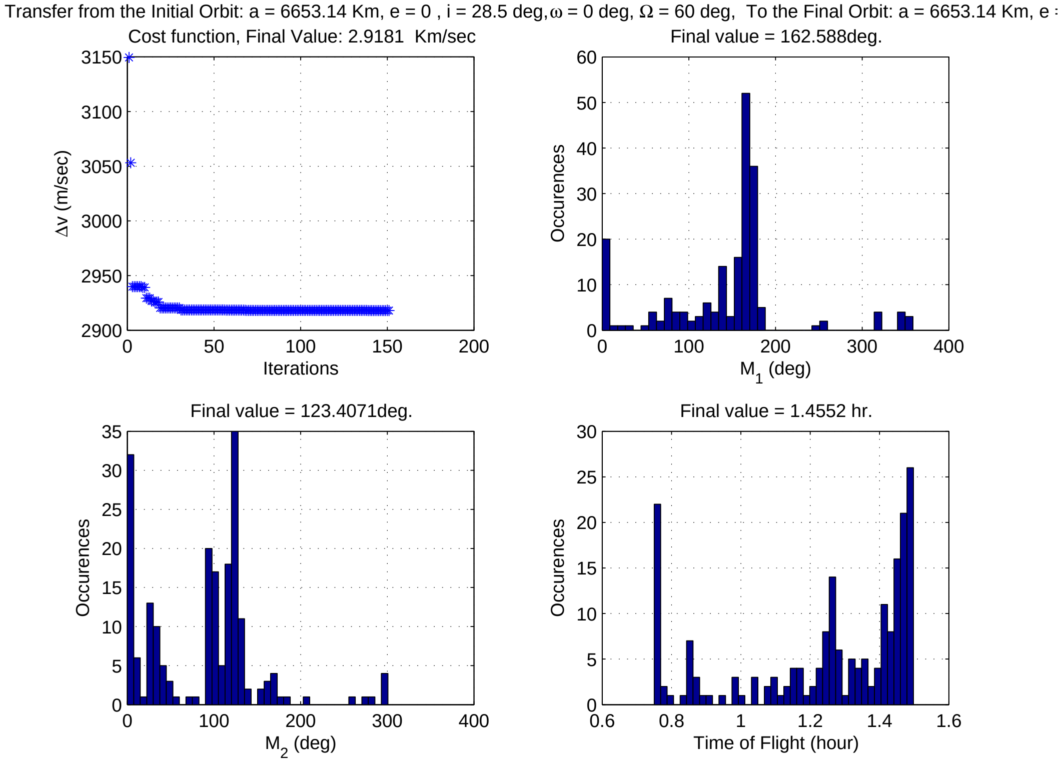

Fig. 18. General plane change: design variables and cost function convergence

as follows. The GAs look for a relation, or similarity, between the design variables among the more fit elements. So, if the optimal solution poses a certain similarity, or relation between some of the design variables, then a GA will figure out this relation very quickly. In this problem, since the optimal solution is characterized by: $\theta_F = \theta_I + \pi$, the GAs is able to figure out this relation and the remaining part is then to find the optimal value for t_f. Any value for θ_I is accepted as long as the above relation is satisfied.

Case 4: Transfer from parking orbit to geosynchronous orbit

Consider the case of transferring a spacecraft from a circular Earth parking orbit at altitude of 185 Km and inclination of 28.5^o, to a geosynchronous orbit of altitude of

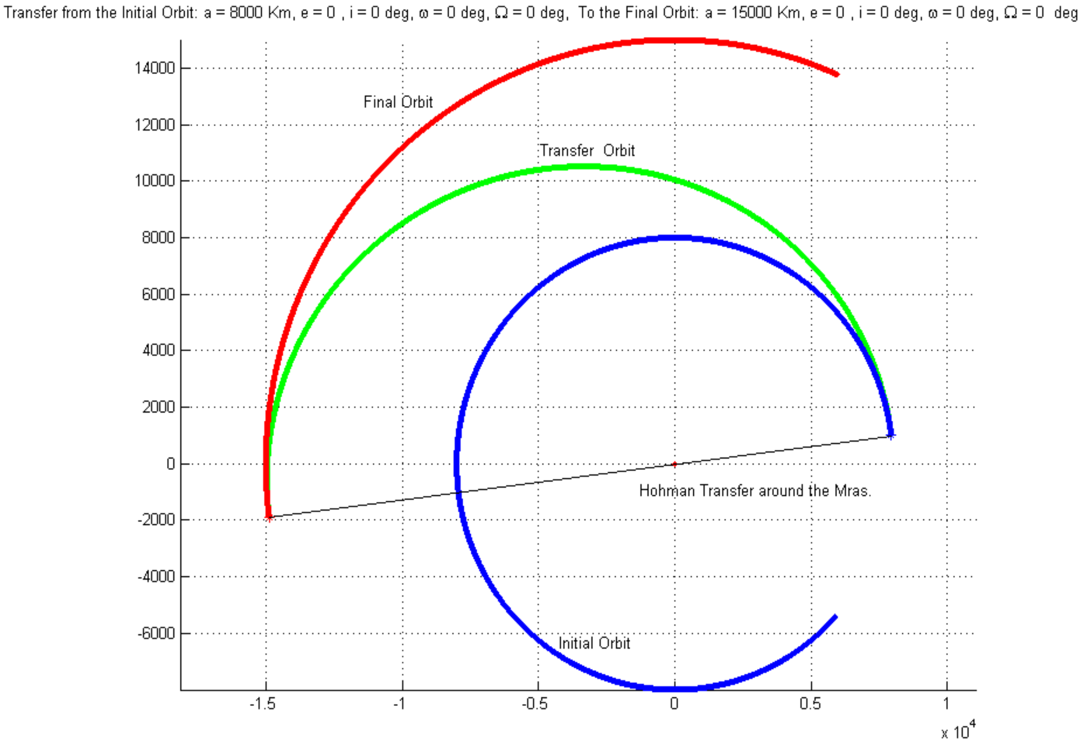

Fig. 19. Solution orbit of the Hohmann transfer problem using GA

35786 Km and zero inclination. First, the spacecraft need to be maneuvered to an elliptic transfer orbit to reach the altitude of the geosynchronous orbit. Then, it needs to be maneuvered to circulate the orbit. Finally, the plane inclination maneuver is performed to reach the final orbit. The previous three maneuvers can be done either separately or by combining the last two maneuvers to save fuel. If they are performed separately, the total cost is $\Delta v = 4.724\ Km/sec$ [14]. If combining the last two maneuvers together, the total cost will be reduced to $\Delta v = 4.297\ Km/sec$.

The developed GA tool is used to find the optimal solution to this problem. The

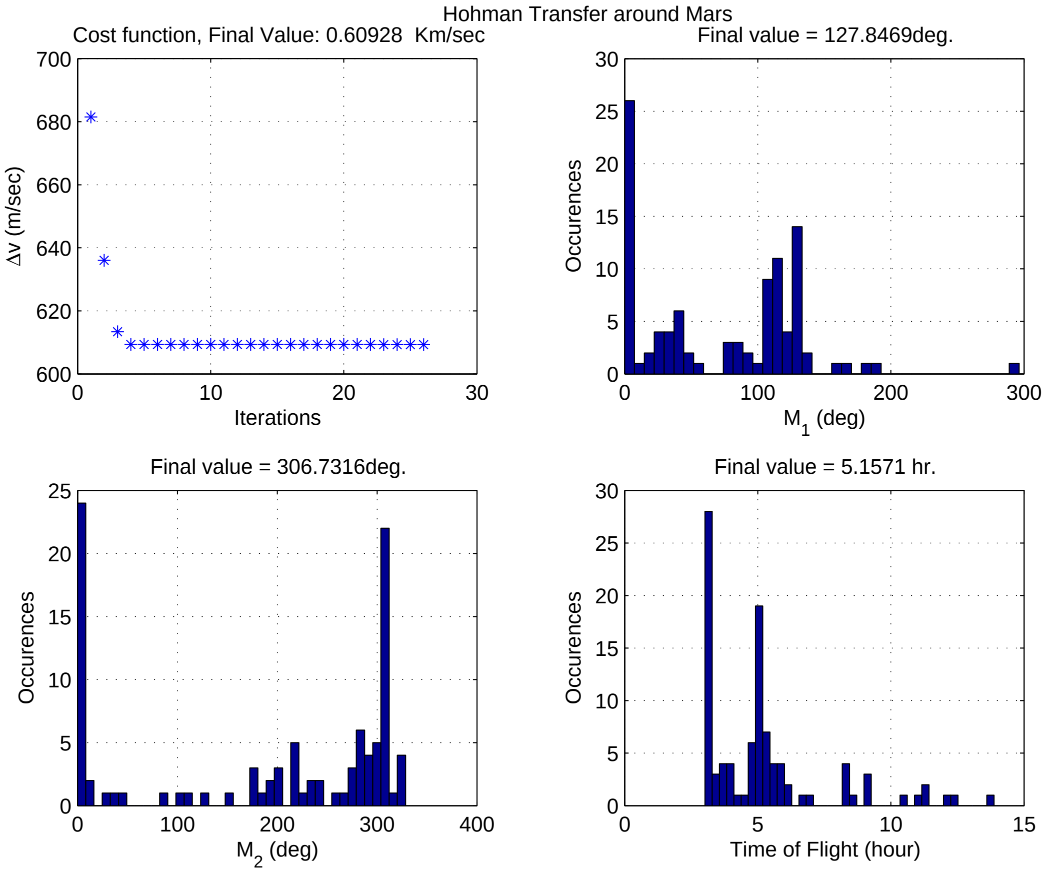

Fig. 20. The Hohmann transfer solution, no. of iterations and design variables values

results are shown in Figures 21-23. The solution consists of two impulses. The first is at the parking orbit, causing the spacecraft to go to the transfer orbit. The second impulse is at the apogee of the transfer orbit causing a plane change and circulating to the final required orbit. Fig. 21 shows the initial, transfer and final orbits. The transfer orbit is inclined to the final orbit. The apogee radius of the transfer orbit is equal to the radius of the final orbit. To better see this, a projection for the three orbits on the equatorial plane is plotted in Fig. 22. Fig. 23 shows the final values for the design variables and the cost convergence history. The total required change in velocity is 4.2742 Km/sec.

Since the initial and final orbits are circular, it does not matter where to depart

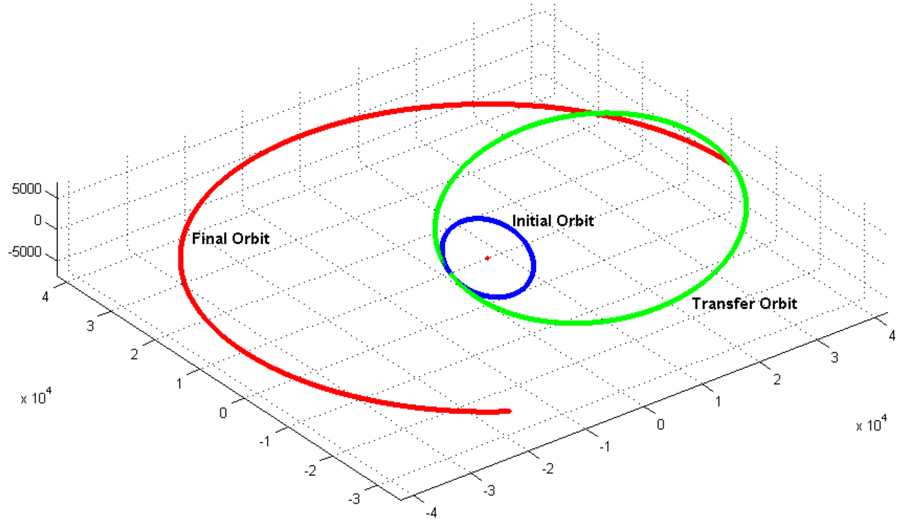

Fig. 21. Parking orbit to geosynchoronous orbit transfer-3D view

from the parking orbit as long as the spacecraft will depart at the perigee of the transfer orbit and depart from the transfer orbit at its apogee. From Fig. 23, the spacecraft departs at an angle equal 181.4^o on the parking orbit, and arrives the final orbit at angle of 359.9^o. The solution provided by the GA developed tool is almost optimal in this case.

2. Discussion

The formulation for the orbit transfer problem has the following advantages. First, the satellite is guaranteed to reach exactly the final orbit even if the transfer is not optimal. Any solution provided by the GA will put the satellite in the final orbit. The non optimality in a solution will appear as more fuel required to transfer the satellite from the initial to the final orbit and longer maneuver time. The reason relies on the fact that for each member in the population, the lambert problem is solved. The solution of the lambert problem yields the exact orbit transfer from the member's

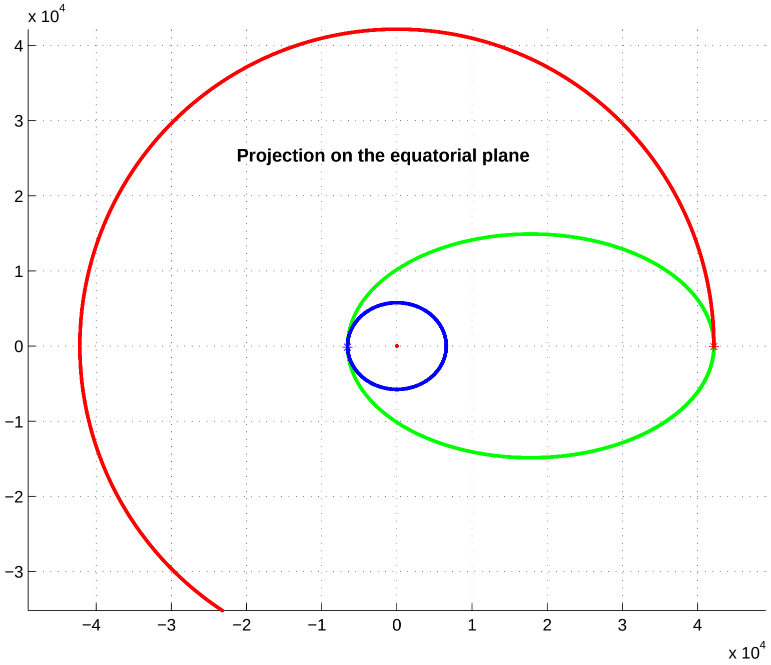

Fig. 22. Parking orbit to geosynchoronous orbit transfer-projected on the equatorial plane

initial position to its final position during its assigned time of flight. So, the most fit member in the last generation, which is the solution provided by the GAs, will exactly arrive the final orbit with no error in position.

Second, the number of design variables is only three, even for the case of non coplanar initial and final orbits. This is compared to six design variable in coplanar orbits and eight design variables in non coplanar orbits in the formulation introduced by Kim and Spencer [17] and compared to three design variables in only coplanar orbits in the formulation introduced by Reichert [28].

Finally, because of the lower number of design variables, the running time for this algorithm is relatively small. This small running time allows for a more exploration for the design space. Usually a GA has to trade off between two contradicting approaches. The first is to exploit the most fit members until the final solution is achieved through

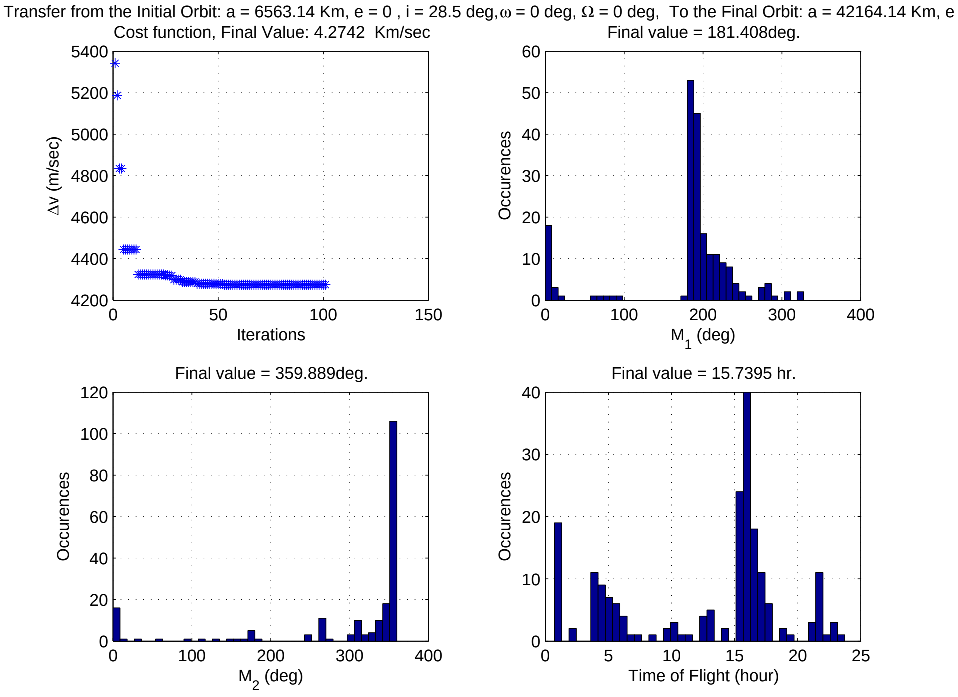

Fig. 23. Parking orbit to geosynchoronous orbit transfer: design variables and cost function convergence

successive GA operations on these most fit members. In the first population, the most fit members are selected and kept in the next generation. In the second generation, the most fit members are again selected and kept to the next generation and so on. Exploitation causes a fast convergence to the solution. The disadvantage of exploitation is that some regions in the design space may not be investigated at all, and the optimal solution could be in these regions. The second approach is to explore the design space. In this approach, most fit members in the current generation are not kept to the next generation, but rather new regions in the design space are investigated. Clearly this approach is slower in finding the final solution but has more probability to hit the true optimal solution. Usually a trade off between exploitation and exploration is tailored for each problem. In the orbital maneuver problem, and because of the small number of variables and fast running time, it is possible to give more time for exploring new solutions and so increase the probability of hitting the true optimal solution.

Having validated the use of the developed algorithm for solving the orbit transfer problems, it is then applied to the general orbit transfer problem as a required part to complete the solution of the Ground surveillance problem.

C. Conclusion

A new formulation for using the genetic algorithms to find the optimal orbit transfer using impulsive thrust was developed in this chapter. The new formulation has some advantages to other current formulations in literature. It has a few number of design variables. Only three design variables are optimized in the general case of a transfer between two non-coplanar orbits. This few number of design variables result in a very efficient algorithm in terms of the computational cost. This efficiency enables the user of this formulation to explore more in the design space searching for the global optimal and so increasing the probability of finding the global optimal solution. The developed formulation also guarantees that the optimal solution is achieved even if the final solution is not the global optimal. The non-optimality appears as more flight time during the transfer and more required fuel for the transfer but the spacecraft will arrive exactly at the final orbit. The developed formulation was validated through a number of cases and proved to find the optimal solution in all cases.

CHAPTER V

TWO WAY ORBITS

This chapter introduces a new set of compatible orbits, called the "Two-way orbits", whose ground track path is a closed-loop trajectory that intersect itself, in some points, with tangent intersections. The spacecraft passes over these tangent intersections once in a prograde and once in a retrograde mode. Motivations are found on the need of having simultaneous observations of the same target area for ground surveillance systems. The general mathematical model to design a Two-way orbit is presented for the specific case where the tangent points are experienced at the orbit extremes, perigee and apogee. As for the general case, the Two-way orbit conditions are formulated and numerically solved. Results show that, in general, Two-way orbits could be formed over any Earth point. Since Two-way orbits use compatible orbits, then all the theory of Flower Constellations can be applied to them. Using these Two-way orbits, this chapter also introduces the Two-way Flower Constellations, having one spacecraft prograde and one retrograde, passing simultaneously over the tangent intersections.

A. Introduction

With the introduction of the Flower Constellations theory (FC), a new unified family of satellite constellations and a new space object have been, simultaneously, created [22]. The Flower Constellations open a new frontier on complex satellite formations for two main reasons. Firstly, the Flower Constellations can be seen as constituted of two distinct parts: an "internal part", associated with the motion of all the satellites along a prescribed identical relative space track, and an "external part", associated with the dynamic of the whole constellation, as a rigid *object*, that spins about an axis with a prescribed angular velocity. Secondly, these new constellation-objects are used, as *building blocks*, to construct more complex configurations that allow to accomplish more complex tasks.

The Flower Constellations, and the more recently introduced Synodic and Relative Flower Constellations (see [20]), combine a number of new attractive features suitable for many potential classic applications (communications, Earth and deep space observation, coverage, navigation systems, etc.), as well as for new and advanced concepts.

A Flower Constellation is built using orbits that are compatible with respect to an assigned rotating reference frame. This implies that all the spacecraft, in this rotating frame, follow the same continuous closed-loop trajectory. For information on the Flower Constellations see also [5], [25], [24], [38], [39], and [40]. In particular, when the reference frame is chosen to be Earth-Centered Earth-Fixed (ECEF), then the FC spacecraft all follow the same relative trajectory (space track) in ECEF and, consequently, the same continuous closed-loop ground track.

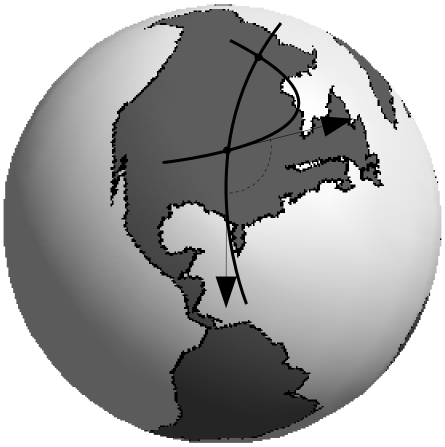

Fig. 24. Ground track intersecting angle.

In general, the ground track is made of prograde and retrograde parts, depending if the spacecraft ground track longitude increases or decreases with the time, respectively. Also, the ground track (for a compatible orbit) is a continuous closed-loop line that intersect itself in several points. These intersections can be characterized by the angle between the ground velocities along the two intersecting parts, Fig. 24. When this angle is equal to π, then the intersecting point is a tangent intersecting point and the two intersecting parts are, locally, one prograde and another retrograde, respectively, over the intersecting point on the Earth surface. This describes the concept of the Two-way orbits. Figure 25 shows an example of a Two-way orbit.

In the Two-way orbits the relative trajectory will have (at least) one tangent intersecting point. This implies that it is possible to build special Flower Constellations with one spacecraft moving along the tangent prograde direction and another spacecraft moving along the retrograde tangent direction. In particular, it is possible to phase the spacecraft in a such a way they will pass over the tangent point simultaneously.

Two cases will be considered here. The first is the “special” Two-way orbits where the tangency point is at the perigee of one spacecraft and the apogee of the other. The second case is the “general” Two-way orbits where the tangency point is any general point on the trajectories of the two spacecraft.

In this chapter, conditions on the orbits parameters such that they constitute a Two-way orbit are derived. The second section will briefly review the theory of Flower Constellations. The third section will review the orbit compatibility conditions. In the fourth section, the case of special Two-way orbits is considered. Condition on the orbits inclination is derived to have Two-way orbits. A plot is generated relating the inclination vs. the eccentricity for specified values of the semimajor axis. The fifth section will develop similar analysis for the general Two-way orbits. However, for

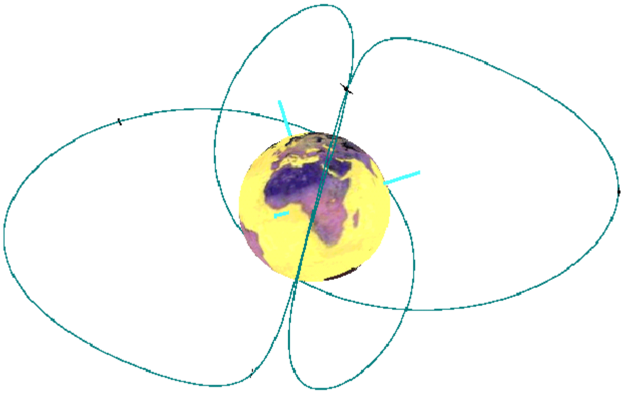

Fig. 25. Two-way orbits example.

this specific case the solution is obtained numerically. Algorithm for the numerical solution is presented in the sixth section. The last section considers the compatibility of the developed conditions with the Flower Constellation theory. Results show that the derived conditions are compatible with the FCs only in the special Two-Way Orbits.

B. Flower Constellations

We shall here briefly summarize some characteristics of the Flower Constellations. They are:

1. Axial-symmetric dynamics,

2. Compatible orbits,

3. Constellation axis can be re-oriented,

4. The whole constellation can be seen as a new space object (secondary path) spinning at constant rate about the constellation axis,

5. Multiple constellations (constellations),

6. Repeated-repeated ground track (repeated),

7. Relative Sun-Synchronous orbits,

8. Morphing constellations,

In order to design a Flower Constellation the "FC Visualization and Analysis Tool" (FCVAT) has been developed. FCVAT is a software written in JAVA and JAVA-3D, it runs on every machine and OS with 3D capabilities, it has easy input/output, it is STK compatible, and it can be specialized for any planet, Sun, and a fictitious planet.

FC Phasing: The closed-loop relative trajectory intersects the associated (general compatible) inertial orbit in many points. A subset of these points identifies the admissible positions for a spacecraft to belong to the same relative trajectory. The logic used to distribute the satellites in the admissible positions encompasses all the possible different distribution.

FC Secondary Paths: For some particular values of the design parameters or, equivalently, for some particular satellite distribution, the resulting Flower Constellation highlights a shape that is maintained during the entire repetition period. This shape, called *Secondary Path*, represents the contour of a fictitious rigid body that spin about the constellation axis with a prescribed angular velocity.

Consider an Earth-Centered Earth-Fixed (ECEF) system of coordinates identified by $\mathcal{E} = \{\mathcal{O}, \hat{e}_x, \hat{e}_y, \hat{e}_z\}$, where the origin $\mathcal{O}$ is at the center of the Earth, $\hat{e}_x$ on the equatorial plane at Greenwich meridian, $\hat{e}_z$ aligned with Earth's spin axis, and $\hat{e}_y = \hat{e}_z \times \hat{e}_x$ to form a right-handed reference frame.

A orbit is called *compatible* with respect to the Earth [6] when the spacecraft trajectory in $\mathcal{E}$ constitutes a closed-loop relative trajectory. A compatible orbit, which is often and inappropriately called *repeated ground track* orbit, is defined as the orbit whose orbital nodal period T_Ω (node to node) satisfies the relationship

$$N_p \, T_\Omega = N_d \, T_{\Omega G} \tag{5.1}$$

where N_p and N_d are two integer numbers indicating the number of orbit periods and the number of the Earth rotational periods to repeat, and where $T_{\Omega G}$ is the Greenwich nodal period, which has been defined by Carter [6] as

$$T_{\Omega G} = \frac{2\pi}{\dot{\alpha}_\oplus - \dot{\Omega}} \tag{5.2}$$

where $\dot{\alpha}_\oplus = 7.29211585530 \times 10^{-5}$ rad/sec is the rotation rate of the Earth and $\dot{\Omega}$ is the nodal regression of a satellite's orbit plane caused by perturbations such as the Earth's oblateness. In particular, $T_r = N_p \, T_\Omega$ is the period of repetition on the relative trajectory.

Equations (5.1) and (5.2) allow us to write

$$T_\Omega = \left(\frac{2\,\pi}{\dot{\alpha}_\oplus - \dot{\Omega}} \right) \frac{N_d}{N_p} = \left(\frac{2\,\pi}{\dot{\alpha}_\oplus - \dot{\Omega}} \right) \xi \tag{5.3}$$

where $\xi = N_d/N_p$ is the rational *compatibility parameter*. Equation (5.3) tells us that, for every distinct value of ξ, there is a different nodal period T_Ω, associated with Earth's compatible orbits. However, this equation can also be seen from a different perspective: for a given value of ξ, an arbitrary orbit (with nodal period T_Ω and nodal rate $\dot{\Omega}$) can be seen as *compatible* with a *fictitious Earth* that rotates with angular velocity

$$\dot{\alpha} = \dot{\Omega} + \frac{2\,\pi}{T_\Omega}\,\xi \tag{5.4}$$

Therefore, every orbit can be seen *compatible* with an associated Earth-Centered Rotating (ECR) system of coordinates that rotates at the angular velocity provided by Eq. (5.3). The final result is that any Earth-compatible orbit is compatible with infinite ECR reference frames. The compatibility concept is a relative concept, which is referring always to a rotating reference frame. Thus, if we consider a different rotating reference frame, then there will be a definition of *orbit compatibility* with respect to *this* reference frame.

C. The "Special" Two-way Orbits

The condition for two satellites to have tangent ground tracks at a point is to have parallel Earth-relative velocities at that point. The Earth-relative velocity, $\vec{V}^R$, is the velocity of the satellite with respect to an Earth rotating system of coordinates.

$$\vec{V}^R = \vec{V} - \vec{V}_E \tag{5.5}$$

where $\vec{V}$ is the satellite velocity in ECI and $\vec{V}_E$ is the local geographical velocity evaluated at radius $\vec{r}$ in ECI. The transformation matrix between inertial and orbital reference frames is ($C \equiv \cos$ and $S \equiv \sin$)

$$R^{\mathrm{T}} = \begin{bmatrix} C_\Omega C_\omega - C_i S_\Omega S_\omega & -C_\Omega S_\omega - C_i S_\Omega C_\omega & S_i S_\Omega \\ S_\Omega C_\omega + C_i C_\Omega S_\omega & -S_\Omega S_\omega + C_i C_\Omega C_\omega & -S_i C_\Omega \\ S_\omega S_i & C_\omega S_i & C_i \end{bmatrix} \tag{5.6}$$

This matrix allows to evaluate inertial (r_i) from orbital (r_o) directions, and vice versa

$$r_i = R^{\mathrm{T}}\, r_o \qquad \Longleftrightarrow \qquad r_o = R\, r_i \tag{5.7}$$

In particular, position and velocity are transformed accordingly with

$$r_i = \frac{p}{1 + e\,\cos\varphi} \left\{ \begin{array}{c} \cos\Omega\,\cos(\omega+\varphi) - \sin\Omega\,\sin(\omega+\varphi)\,\cos i \\ \sin\Omega\,\cos(\omega+\varphi) + \cos\Omega\,\sin(\omega+\varphi)\,\cos i \\ \sin(\omega+\varphi)\,\sin i \end{array} \right\} \tag{5.8}$$

while the velocity in orbital reference frame is expressed as

$$v_o = \sqrt{\frac{\mu}{p}} \left\{ \begin{array}{c} -\sin\varphi \\ e + \cos\varphi \\ 0 \end{array} \right\} \tag{5.9}$$

Let us consider, for simplicity, two eccentric orbits having the apsidal lines lying on the equatorial plane ($\omega = 0$). Under this condition, the velocity of the first at perigee is

$$\vec{V}_{p1} = (e+1)\sqrt{\frac{\mu}{p}} \left\{ \begin{array}{c} -\sin\Omega_1\,\cos i \\ \cos\Omega_1\,\cos i \\ \sin i \end{array} \right\} \tag{5.10}$$

while at apogee of the second orbit the velocity is

$$\vec{V}_{a2} = (e-1)\sqrt{\frac{\mu}{p}} \left\{ \begin{array}{c} -\sin\Omega_2\,\cos i \\ \cos\Omega_2\,\cos i \\ \sin i \end{array} \right\} \tag{5.11}$$

The local geographical velocity is

$$\vec{V}_E = \vec{\omega}_E \times \vec{r} \tag{5.12}$$

where $\vec{\omega}_E$ is the Earth angular velocity. Specializing Eq. (5.12) for the perigee position and orbit #1 we obtain

$$\vec{V}_{Ep1} = \begin{Bmatrix} 0 \\ 0 \\ \omega_E \end{Bmatrix} \times \frac{p}{1+e} \begin{Bmatrix} \cos\Omega_1 \\ \sin\Omega_1 \\ 0 \end{Bmatrix} = \frac{p\,\omega_E}{e+1} \begin{Bmatrix} -\sin\Omega_1 \\ \cos\Omega_1 \\ 0 \end{Bmatrix} \tag{5.13}$$

while at apogee of orbit #2 the Earth velocity is

$$\vec{V}_{Ea2} = \begin{Bmatrix} 0 \\ 0 \\ \omega_E \end{Bmatrix} \times \frac{p}{1-e} \begin{Bmatrix} -\cos\Omega_2 \\ -\sin\Omega_2 \\ 0 \end{Bmatrix} = \frac{p\,\omega_E}{e-1} \begin{Bmatrix} -\sin\Omega_2 \\ \cos\Omega_2 \\ 0 \end{Bmatrix} \tag{5.14}$$

Substituting Eqs. (5.10) and (5.13) into Eq. (5.5) we obtain

$$\vec{V}_p^R = (e+1)\sqrt{\frac{\mu}{p}} \begin{Bmatrix} -\sin\Omega_1\,\cos i \\ \cos\Omega_1\,\cos i \\ \sin i \end{Bmatrix} - \frac{p\,\omega_E}{e+1} \begin{Bmatrix} -\sin\Omega_1 \\ \cos\Omega_1 \\ 0 \end{Bmatrix} \tag{5.15}$$

while Eqs. (5.11) and (5.14) into Eq. (5.5) we obtain

$$\vec{V}_a^R = (e-1)\sqrt{\frac{\mu}{p}} \begin{Bmatrix} -\sin\Omega_2\,\cos i \\ \cos\Omega_2\,\cos i \\ \sin i \end{Bmatrix} - \frac{p\,\omega_E}{e-1} \begin{Bmatrix} -\sin\Omega_2 \\ \cos\Omega_2 \\ 0 \end{Bmatrix} \tag{5.16}$$

In general, in order to have tangent ground tracks at the intersection, the two velocity vectors and the vector pointing at the intersection, $\vec{R}_{eq}$, must be linearly dependent (they identify the plane passing through the origin of the coordinates and containing the two velocities $\vec{V}_a^R$ and $\vec{V}_p^R$). In our case, since we are looking for tangency at equator and experienced at perigee/apogee, then our tangency condition can be substituted with the condition for the two velocity vectors, $\vec{V}_a^R$ and $\vec{V}_p^R$, of being parallel. This implies that we can write

$$\vec{V}_a^R = k\,\vec{V}_p^R \tag{5.17}$$

where k is the proportionality constant whose value can be directly obtained from the third scalar identity of Eq. (5.17)

$$k = \frac{e-1}{e+1} \tag{5.18}$$

This condition yields to the relationship

$$\frac{-C_i S_{\Omega 1} V_p + \omega_E r_p S_{\Omega 1}}{C_i S_{\Omega 2} V_a - \omega_E r_a S_{\Omega 2}} = \frac{C_i C_{\Omega 1} V_p - \omega_E r_p C_{\Omega 1}}{-C_i C_{\Omega 2} V_a + \omega_E r_a C_{\Omega 2}} \tag{5.19}$$

and

$$\frac{-C_i S_{\Omega 1} V_p + \omega_E r_p S_{\Omega 1}}{C_i S_{\Omega 2} V_a - \omega_E r_a S_{\Omega 2}} = \frac{V_p}{-V_a} \tag{5.20}$$

With little manipulation, Eq. (5.19) is satisfied iff

$$\sin\left(\Omega_1 - \Omega_2\right) = 0 \tag{5.21}$$

or

$$V_a V_p C_i^2 - \omega_E C_i \left(r_p V_a + r_a V_p\right) + \omega_E^2 r_p r_a = 0 \tag{5.22}$$

The latter case is refused because it does not satisfy the condition in Eq. (5.20) The result in Eq. (5.21) states that either $\Omega_1 = \Omega_2$, which is a trivial case where the two orbits are identical, or

$$\Omega_1 - \Omega_2 = \pi \tag{5.23}$$

The latter gives the condition on the right ascension of ascending node for the two orbits. The condition in Eq. (5.20), after manipulation, implies that

$$C_i = \frac{\omega_E \left(\dfrac{r_p}{V_p} S_{\Omega 1} - \dfrac{r_a}{V_a} S_{\Omega 2}\right)}{S_{\Omega 1} - S_{\Omega 2}} \tag{5.24}$$

From Eq. (5.23), we have $\sin(\Omega_2) = -\sin(\Omega_1)$ then Eq. (5.24) simplifies to

$$\cos i = \frac{\omega_E}{2}\left(\frac{r_p}{V_p} + \frac{r_a}{V_a}\right) - \frac{2e\omega_E}{\sqrt{a(1-e^2)}}\sqrt{\frac{a^3}{\mu}} \tag{5.25}$$

where $V_p = \sqrt{\frac{2\mu}{r_p} - \frac{\mu}{a}}$ and $V_a = \sqrt{\frac{2\mu}{r_a} - \frac{\mu}{a}}$. Equations (5.23) and (5.25) constitutes the necessary and sufficient conditions to have Two-way orbits.

Fig. 26 shows *Two-Way* Orbits inclinations for different values of e and a.

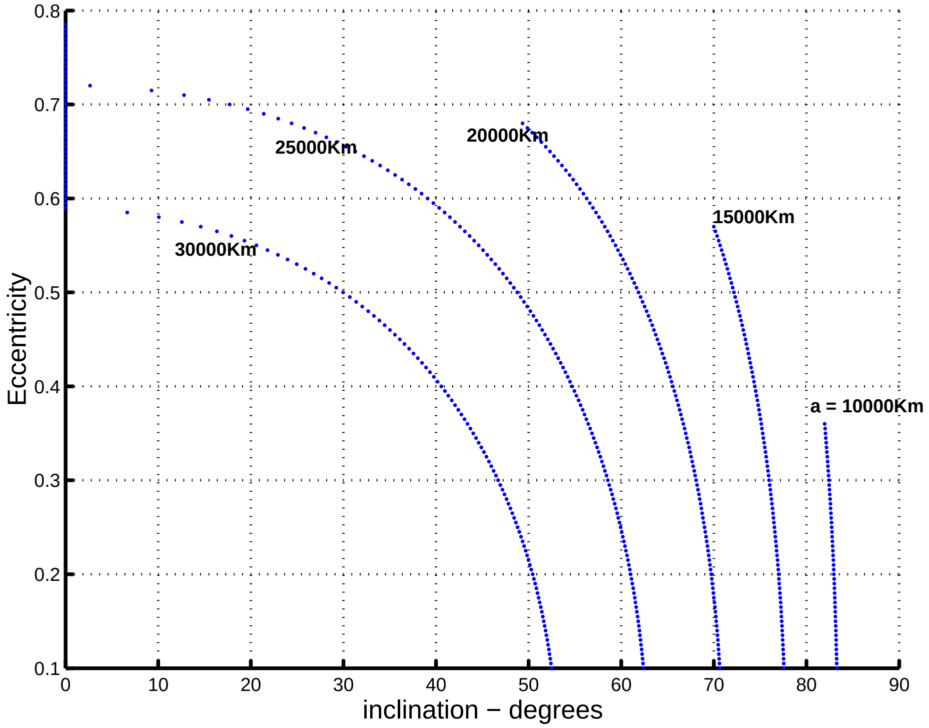

Fig. 26. Two-way orbits inclinations for different values of e and a

D. The "General" Two-way Orbits

We considered the case where the two orbits have similar shape, size and inclination. Moreover, we assumed zero argument of perigee and the intersection point occur at the perigee and apogee points of the two orbits. In this section we look at the same problem but the intersection point is any general point, not necessarily an apogee or perigee.

First we find the condition of having an intersection between the two ground tracks for two different orbits. An intersection between the ground tracks occurs if the two position vectors of the two satellites are parallel.

$$\vec{r}_1 = k_r \vec{r}_2 \tag{5.26}$$

Substituting for the vectors $\vec{r}_1$ and $\vec{r}_2$ from Eq. (5.8) then we can get the following two conditions for an intersection to occur

$$\frac{\cos\Omega_1 \cos(\omega+\varphi_1) - \sin\Omega_1 \sin(\omega+\varphi_1)\cos i}{\cos\Omega_2 \cos(\omega+\varphi_2) - \sin\Omega_2 \sin(\omega+\varphi_2)\cos i} = \frac{\sin(\omega+\varphi_1)}{\sin(\omega+\varphi_2)} \tag{5.27}$$

and

$$\frac{\sin\Omega_1 \cos(\omega+\varphi_1) + \cos\Omega_1 \sin(\omega+\varphi_1)\cos i}{\sin\Omega_2 \cos(\omega+\varphi_2) + \cos\Omega_2 \sin(\omega+\varphi_2)\cos i} = \frac{\sin(\omega+\varphi_1)}{\sin(\omega+\varphi_2)} \tag{5.28}$$

These two conditions can be simplified to the following form

$$\begin{bmatrix} C_i a_1 & -a_3 \\ a_3 & C_i a_1 \end{bmatrix} \begin{Bmatrix} S_{\Omega 1} \\ C_{\Omega 1} \end{Bmatrix} = \begin{bmatrix} C_i a_1 & -a_4 \\ a_4 & C_i a_1 \end{bmatrix} \begin{Bmatrix} S_{\Omega 2} \\ C_{\Omega 2} \end{Bmatrix} \tag{5.29}$$

where, $a_1 = \sin(\omega+\varphi_1)\sin(\omega+\varphi_2)$, $a_3 = \cos(\omega+\varphi_1)\sin(\omega+\varphi_2)$, and $a_4 = \sin(\omega+\varphi_1)\cos(\omega+\varphi_2)$.

1. Observations

1. We notice that for the special case where $a_3 = a_4$, then either $\Omega_1 = \Omega_2$, which is obvious case, or ${C_i}^2{a_1}^2 + {a_3}^2 = 0$ and this is satisfied only if $\omega + \varphi_2 = n\pi$ where, $n = 0, 1, \ldots$. This later case is the special case solved in the previous section.

2. If we eliminate C_i from Eq. (5.29) then we get

$$\frac{a_4 S_{\Omega 2} - a_3 S_{\Omega 1}}{\Delta C} = \frac{a_3 C_{\Omega 1} - a_4 C_{\Omega 2}}{\Delta S} \tag{5.30}$$

where $\Delta C = C_{\Omega 1} - C_{\Omega 2}$ and $\Delta S = S_{\Omega 1} - S_{\Omega 2}$. Then by rearrangement of Eq. (5.30)we can write

$$\frac{a_4}{a_3} \equiv \frac{\tan(\omega + \varphi_1)}{\tan(\omega + \varphi_2)} = \frac{C_{\Omega 1}\Delta C + S_{\Omega 1}\Delta S}{C_{\Omega 2}\Delta C + S_{\Omega 2}\Delta S} \tag{5.31}$$

This can be further simplified to the form

$$(a_3 + a_4)[\,\cos(\Omega_1 - \Omega_2) - 1\,] = 0 \tag{5.32}$$

Which means that either $\Omega_1 = \Omega_2$, which is a trivial solution, or $a_3 = a_4$. The latter can be written in the form

$$\omega_1 + \varphi_1 + \omega_2 + \varphi_2 = n\pi \qquad n = 0, 1, \cdots \tag{5.33}$$

3. For given Ω_1 and Ω_2, Eq. (5.31) gives one relation between φ_1 and φ_2 at the intersection point. Another relation is required to find the two anomalies. The other relation should relate them through the time equation. The above relation only states that the intersection is possible between the two tracks but not necessarily implies that the both satellites will pass by this point at the same time. A second condition should relate the two anomalies through the time such that the two satellite will pass by the intersection point at the same time.

4. For the special case where $\Omega_1 - \Omega_2 = \pi$, the above equation reduces to $\dfrac{a_4}{a_3} = -1$. This means that $\varphi_2 = \pi - \varphi_1$, which is the special case solved in the previous section.

The condition in Eq. (5.33) implies that the two trajectories of two satellites intersects at certain point; however, it does not imply that the two satellites will pass by this point both at the same time. To guarantee that both will pass by the intersection point at the same time, we introduce the following condition.

Assume the two satellites of interest intersect at time $t = t_i$, and writing the time equation for both satellites at the intersection point

$$(t_i - t_{p1})\, n = \psi_{1i} - e \sin(\psi_{1i}) \tag{5.34}$$

$$(t_i - t_{p2})\, n = \psi_{2i} - e \sin(\psi_{2i}) \tag{5.35}$$

where n is the mean motion. Then,

$$\psi_{2i} - e \sin(\psi_{2i}) = \psi_{1i} - e \sin(\psi_{1i}) - n(t_{p2} - t_{p1}) \tag{5.36}$$

In order to find t_{p1} and t_{p2}, we need to define the phasing between the two satellites. Recalling that for a Flower Constellation we have two phasing conditions [22]. The first, for a two body case

$$M_{k+1}(0) = M_k(0) + 2\pi \frac{F_n}{F_d} \frac{n}{\omega_E} \tag{5.37}$$

where F_n and F_d are integers defining the phasing of the two satellites. Now if we set, without loss of generality, $t_{p1} = -M_1(0)/n$ and $t_{p2} = -M_2(0)/n$, then the condition in Eq. (5.36) becomes

$$\psi_{2i} - e \sin(\psi_{2i}) = \psi_{1i} - e \sin(\psi_{1i}) + 2\pi \, \frac{F_n}{F_d} \, \frac{n}{\omega_E} \tag{5.38}$$

Eq. (5.33) and Eq. (5.38) completely determine the intersection point of the two satellites.

Now, we proceed to the condition of two way orbits. We proceed as in the previous section but with general orbit parameters.

Assume that the point of intersection occur at point *1* in the first orbit corresponding to a true anomaly, φ_1, and at point *2* in the second orbit corresponding to a true anomaly, φ_2. And assume also that the two orbits have common e, i, and ω. Then it can be shown that the velocity of point *1* relative to the Earth is

$$\vec{V}_1^R = \left\{ \begin{array}{c} \sqrt{\frac{\mu}{p_1}}\left(\tau_{11}\tau_{12} + \tau_{13}\tau_{14}\right) + \frac{\omega_E p_1}{1 + e\cos(\varphi_1)}\tau_{15} \\ \sqrt{\frac{\mu}{p_1}}\left(\tau_{21}\tau_{12} + \tau_{23}\tau_{14}\right) - \frac{\omega_E p_1}{1 + e\cos(\varphi_1)}\tau_{25} \\ \sqrt{\frac{\mu}{p_1}} S_i\left(S_\omega\tau_{12} + C_\omega\tau_{14}\right) \end{array} \right\} \tag{5.39}$$

where

$$\begin{aligned}
\tau_{11} &= C_{\Omega 1}C_\omega - C_i S_{\Omega 1} S_\omega \\
\tau_{12} &= \sin(\varphi_1)\left[\cos(\varphi_1)\left(1 - e\right) - 1\right] \\
\tau_{13} &= -C_{\Omega 1}S_\omega - C_i S_{\Omega 1} C_\omega \\
\tau_{14} &= 1 - \left[\cos(\varphi_1)\left(1 - e\right) - 1\right]\cos(\varphi_1) \\
\tau_{15} &= S_{\Omega 1}\cos(\omega + \varphi_1) + C_i C_{\Omega 1}\sin(\omega + \varphi_1) \\
\tau_{21} &= S_{\Omega 1}C_\omega + C_i C_{\Omega 1} S_\omega \\
\tau_{23} &= -S_{\Omega 1}S_\omega + C_i C_{\Omega 1} C_\omega \\
\tau_{25} &= C_{\Omega 1}\cos(\omega + \varphi_1) - C_i S_{\Omega 1}\sin(\omega + \varphi_1)
\end{aligned}$$

The vector $\vec{V}_2^R$ is defined similar to $\vec{V}_1^R$ with p_2, Ω_2 and φ_2 replacing p_1, Ω_1 and φ_1, respectively.

$$\vec{V}_2^R = \left\{ \begin{array}{c} \sqrt{\frac{\mu}{p_2}}\left(\sigma_{11}\sigma_{12} + \sigma_{13}\sigma_{14}\right) + \frac{\omega_E p_2}{1 + e\cos(\varphi_2)}\sigma_{15} \\ \sqrt{\frac{\mu}{p_2}}\left(\sigma_{21}\sigma_{12} + \sigma_{23}\sigma_{14}\right) - \frac{\omega_E p_2}{1 + e\cos(\varphi_2)}\sigma_{25} \\ \sqrt{\frac{\mu}{p_2}} S_i\left(S_\omega\sigma_{12} + C_\omega\sigma_{14}\right) \end{array} \right\} \tag{5.40}$$

where σ_{ij} correspond to τ_{ij}.

The condition for having two way orbits is that the vectors $\vec{V}_1^R$, $\vec{V}_2^R$ and the position vector of the intersection point, $\vec{r}_i$, belongs to the same plane. Then we can write this condition as follow:

$$\chi = \vec{r}_i \cdot \left(\vec{V}_1^R \times \vec{V}_2^R\right) = 0 \tag{5.41}$$

It is difficult to derive analytically an expression that gives the inclination of such orbits, however a numerical solution is developed.

E. Numerical Solution Algorithm

It is possible to introduce a numerical algorithm to find the intersection point of two satellites and the condition of the two orbits such that they constitute a Two-way orbit. This is done through two consecutive steps.

1. Determine the Intersection Point

Equations (5.33) and (5.38) can be solved numerically to fined φ_1 and φ_2 as follow:

1. Assume a value for φ_1,
2. Get the corresponding ψ_1,
3. Given the phasing parameters, F_n and F_d, evaluate ψ_2 using Eq. (5.38),
4. Evaluate the corresponding φ_2, and
5. Check if φ_1 and φ_2 satisfy Eq. (5.33), if not then repeat from step 1.

This will result in the values of φ_1 and φ_2 at the intersection point given the orbital shape, a, e, and ω.

2. Determine the Orbits Inclination

In this step, we use the Two-way orbit condition, Eq. (5.41), to find the orbit inclination. Given Ω_1, Ω_2 can calculated as follow

$$\Omega_2 = \Omega_1 - 2\pi \frac{Fn}{Fd} \tag{5.42}$$

we will then loop on all possible values inclination, and in each time we check if the derived condition is satisfied or not. This will result in all possible values for the inclination, i, completing the five orbital elements. There are many parameters to play with, one case is plotted in Figures 27 and 28 where $F_n = 1$ and $F_d = 4$. The condition $\chi = 0$ satisfaction is investigated and the variation of χ with different values for the eccentricity is plotted in Fig. 28.

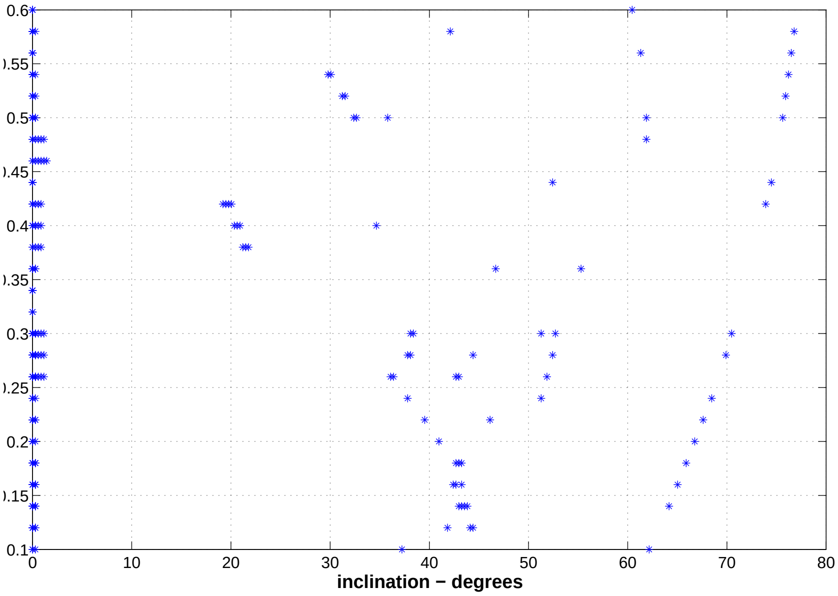

Fig. 27. "General" Two-way orbits: eccentricity vs. inclinations for different values of a

For a Flower Constellation [22], we have two phasing conditions. The first is

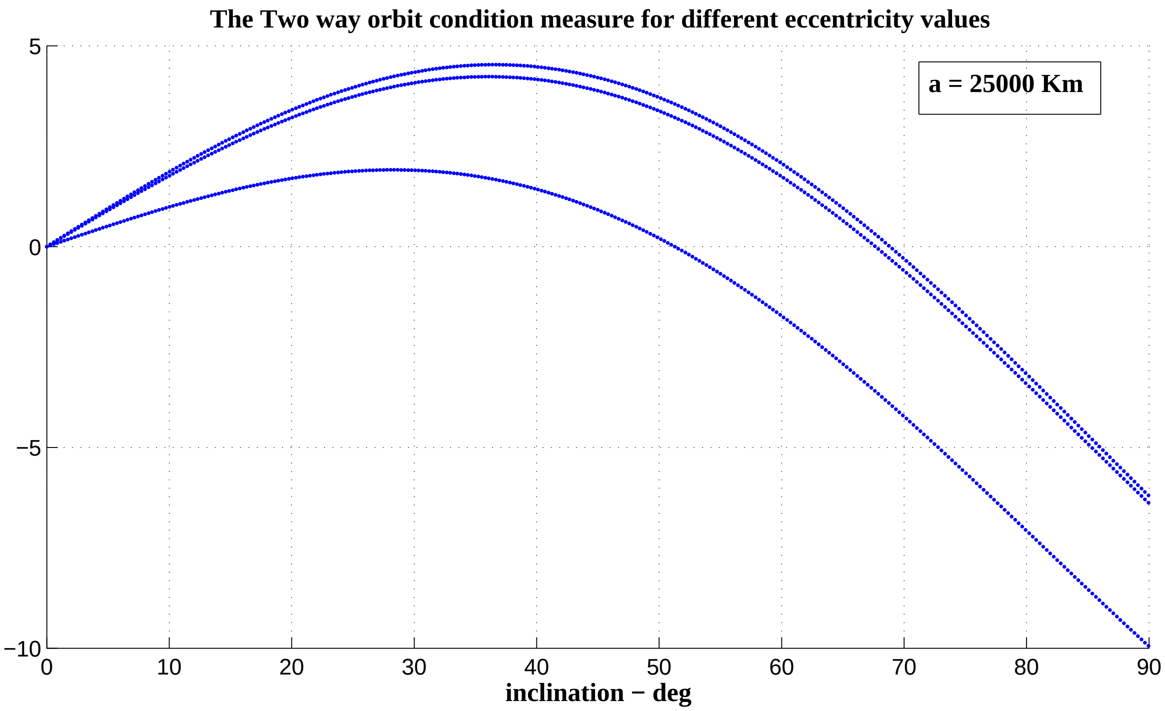

Fig. 28. "General" Two-way orbits: The χ values vs. inclination for different values of e

used in calculating the intersection point as discussed above. The second is used to calculate Ω_2, Eq. (5.42). So all the calculated satellites constitute a Flower Constellation.

F. Conclusions

In this chapter, the concept of Two-way orbits is investigated. The special case of two satellites intersecting at their perigee and apogee locations is solved analytically. The general case of the two satellites intersecting at two general points on their orbits with tangent ground tracks is formulated and solved numerically. The case of two satellites in a Flower Constellation is investigated and results demonstrated possible existence of general Two-way orbits in the Flower Constellation set.

CHAPTER VI

SPACE SURVEILLANCE WITH STAR TRACKERS ORBIT ESTIMATION

This chapter develops an estimator that uses star tracker measurements to estimate the orbit of space objects. Given a history of observations from a star tracker on-board an observing spacecraft to a space object, a batched least square approach is implemented to estimate the object orbit. A dual-use for wide field-of-view star trackers along with the Pyramid algorithm that provides their attitude determination capabilities can provide a measurement for the direction of the space object. As a matter of course, the Pyramid algorithm examines all objects that appear in the field of view of the star tracker. Objects that are identified as belonging to the star catalog are used for precision attitude determination. The star tracker, owing to the robustness of the Pyramid algorithm, can be used to obtain azimuth and elevation angles for the other unidentified space objects in the satellite's body frame. Additionally, star trackers are capable of taking measurements at frequencies approaching 100 Hz. Hence, star trackers that have adopted the Pyramid algorithm are capable of producing extremely dense short-arc in situ surveillance data of space objects within their field of view. Once the surveillance data has been collected on-board the satellite, it can be down-linked to Earth for use in orbit determination and object identification. The object identification process can be eliminated if the star tracker is deliberately pointed at a known space object for which we desire to have precision observations.

A. Introduction

Orbit determination is an integral part of current space operations. The ability to know precisely where a space object is orbiting the Earth is critical to mission planners for reaching science objectives as well as avoiding catastrophic collisions with other space objects or space debris. Currently, tracking information is provided by a number of facilities around the world including NASA's Satellite Laser Ranging (SLR) stations, the U.S. Air Force Space Surveillance Network, the U.S. High Accuracy Network Determination System (HANDS), the U.S. Navy Interferometer Fence, the Russian Space Surveillance Syste, the French Doppler Orbitography and Radio Position Integrated by Satellite system, as well as the U.S. Global Positioning System. The SLR stations provide some of the most accurate data that is often used to define "truth" orbits; however, its use is primarily limited to those space objects equipped with retro-reflectors.

Short-arc dense data sets are defined by Vallado and Carter [35] as having observations every second for at least two minutes. Long-arc data sets are in the five to six minute range. Dense data allows one to estimate a larger state that not only includes position and velocity but also error biases and the ballistic coefficient. Additionally, short-arc dense data provides a more realistic covariance matrix while the importance of the fit-span diminishes. Lastly, dense data sets allow for improved orbit determination of drag perturbed orbits.

In a study of the High Accuracy Network Determination System (HANDS), Sabol and Culp [29] found that supplementing traditional ranging observations with unbiased and well-calibrated angles only optical observations could improve the estimated error to the 10 m range. The Raven class telescope [30] has measurement standard deviations of less than $1''$ ($4.8 \cdot 10^{-6}$ rad) in azimuth and $0.5''$ ($2.4 \cdot 10^{-6}$ rad) in declination. However, it suffers from systemic errors [36] that are currently under investigation. Once these systemic errors are removed, the HANDS network will prove to be one of the most accurate and affordable ground-based tracking networks available. For comparison, the radar facilities at Eglin Air Force Base, located at N30.2316° latitude and W86.2147° longitude, have measurement error standard deviations of $\sigma_{\text{range}} = 34.3$ m, $\sigma_{\text{azimuth}} = 0.0149°$, and $\sigma_{\text{elevation}} = 0.0166°$. [37]

Now consider if one were able to take observations of space objects in situ. In situ data has the advantage of being taken outside the Earth's atmosphere. By using Gaussian best fitting to centroid the observed stars [21] and a state of the art CCD/CMOS digital imager, the centroiding of space objects can be provided with standard deviations below $2''$ (0.00083°). Additionally, star trackers have been shown to take reliable observations at frequencies approaching 100 Hz [33]. Therefore, wide FOV star trackers combined with the Pyramid algorithm described above can produce highly accurate, dense data sets for objects that cross its field of view. Clearly, a sensor that can provide dense, highly accurate angles only data sets in situ would be desirable.

Once these data sets have been obtained, there are two primary methods of orbit determination. The most widely used is the least squares method. Batch least squares has been implemented in most major orbit determination packages such as the NASA GSFC Ops Goddard Trajectory Determination System (Ops GTDS), the Naval Research Laboratory's Special-K, and the General Research Corporation's Analyst Workstation (AWS) Program to name a few. That being said, Kalman filtering presents new opportunities for orbit determination that should not be overlooked. Kalman filtering allows for the fact that our force models are imperfect, which may lead to a more accurate understanding of the orbit error.

In the following sections, the observability of the system is first investigated to check the possibility of estimating the system states from the measurements. For estimation purposes, the model of the system is here introduced and an analytical expression for the Jacobian matrix is derived. The Gaussian Least Squares Differential Correction (GLSDC) technique is implemented and results are presented.

B. Observability Investigation

In this section, the system observability is investigated to see whether the states of the system are independently observable from the measurements or not. For a linear system, this can be done by calculating the observability matrix and checking if it can be inverted or not. If it can be inverted, then all the states are observable from the measurements of the system [23]. For a nonlinear system, however, it is not that straight forward. The Newtonian gravitational attraction model is nonlinear. Even, if we assume small relative distance between the two spacecraft and use Hills equations as a model for the relative motion, the states equations are linear but the measurement is a nonlinear function of the states. Two cases will be considered to check the observability, the Hills equations model and the Newtonian attraction model.

1. Observability of the Nonlinear Model

In this nonlinear model, the system states are the object orbital elements and the true anomaly at the first measurement. The measurements are a series of directions from the observing spacecraft to the object spacecraft. The observability of the state vector from the measurements can be investigated by perturbing each of the states and check the resulting perturbation in the measured quantity [26]. If a perturbation of each of the states results in independent perturbation responses in the measured quantity, then all the states can be estimated from the measured quantity.

Each of the orbital elements for the object orbit is perturbed as well as the true anomaly at epoch, the first measurement. These perturbations result in a corresponding perturbation in the vector of object position relative to the observing spacecraft position. The relative position vector can be seen as two separate quantities, magnitude and direction. We are only concerned about the direction because this is the quantity that we can measure.

As a case study, it is assumed that the two spacecraft are moving in the same orbit with small eccentricity of 0.005 and an altitude of $550Km$. The object spacecraft, spacecraft A, is leading the observing spacecraft, spacecraft B. Figures 29-33 show the perturbation in the relative position magnitude and direction due to states perturbations. The direction is measured w.r.t. an inertial fixed axis.

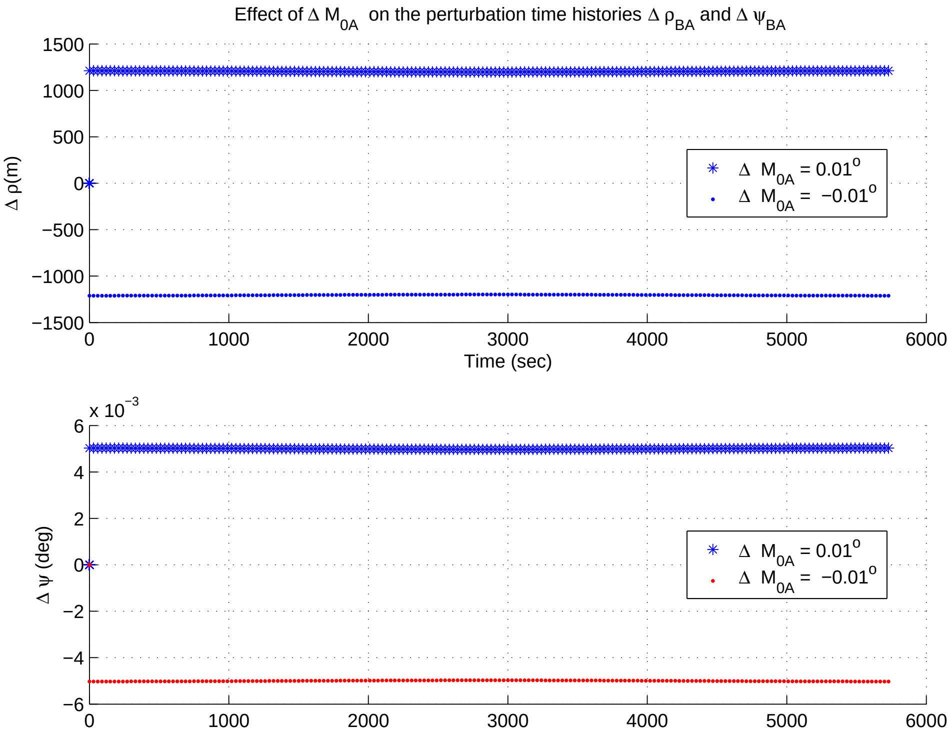

Fig. 29. Perturbations of range and angle caused by perturbation of the mean anomaly at epoch

As can be seen from Fig. 29, positive perturbation in the true anomaly at epoch results in constant positive perturbation in the direction of the relative position vector, ψ_{AB}. A negative perturbation causes also a negative constant perturbation in ψ_{AB}.

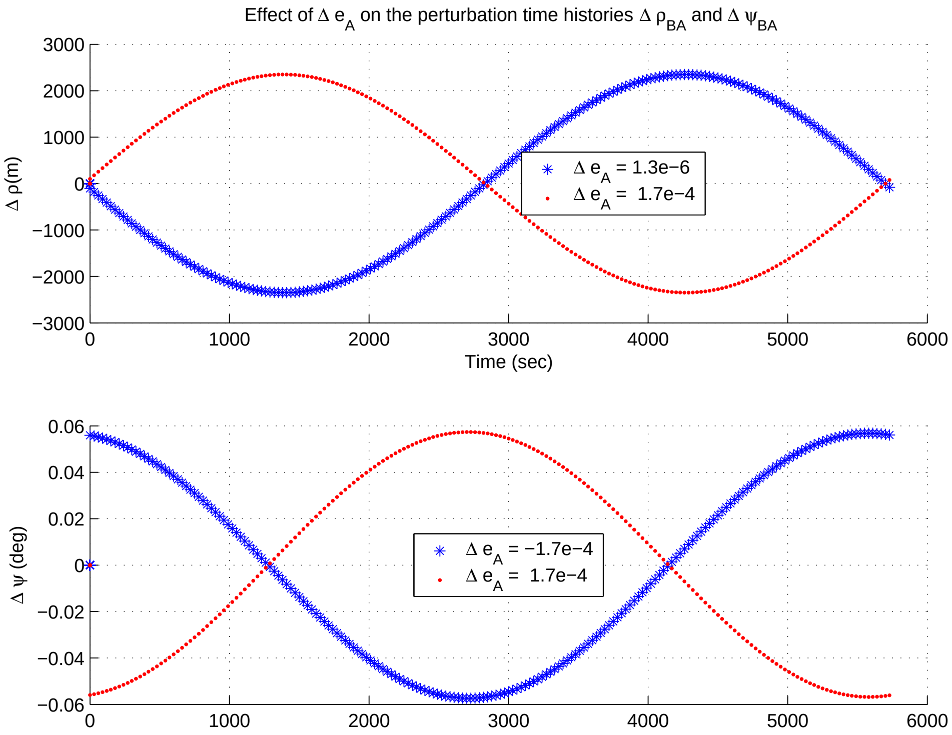

Fig. 30. Perturbations of range and angle caused by perturbation of the eccentricity

Eccentricity perturbation causes harmonic perturbation in ψ_{AB}, as shown in Fig 30. Positive and negative eccentricity perturbations can be distinguished from each other from the phase difference in the perturbed ψ_{AB}. Since eccentricity perturbations cause harmonic ψ_{AB} perturbations compared to a constant perturbation for the case of the true anomaly at epoch, then these two states can distinguished from each other based on the measured ψ_{AB}. Considering Fig. 31, any perturbation in the semi major axis will cause the two spacecraft to orbit the Earth with different orbital period. This will result in ψ_{AB} changing as a ramp or even faster. Positive and negative perturbations in a, or equivalently in n, are distinguishable from each other based on the sign of the perturbed ψ_{AB}. Both are distinguishable from previous states because of the ramp

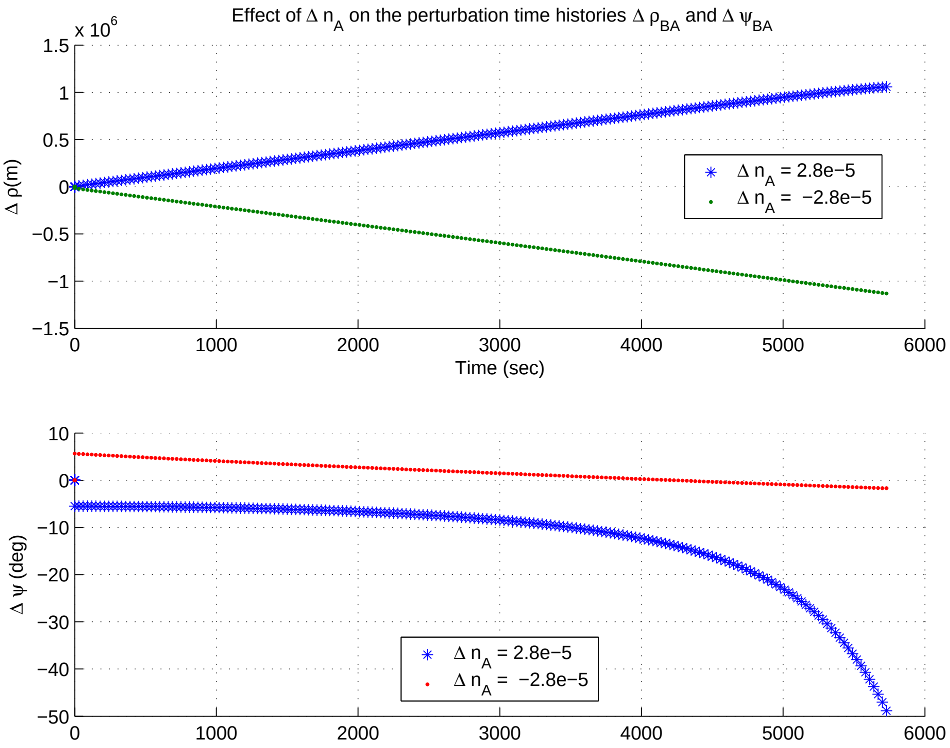

Fig. 31. Perturbations of range and angle caused by perturbation of the semi-major axis

nature of the ψ_{AB} response.

The orbit plane observability can be investigated geometrically as follows. First, consider the case of the two spacecraft are in the same plane. Then all the measured directions of the target spacecraft will be in the same plane. So, if all the measured directions are normal to the known observer plane normal vector, then the target spacecraft is in the observer plane. For the case of the two spacecraft are not in the same plane, consider the intersection point of the two orbits. The observer spacecraft orbit normal is known. The unit vector from the observer to the target at the intersection point is in the observer plane. So it is normal to the known observer plane normal vector. This means that the intersection points can be determined from the

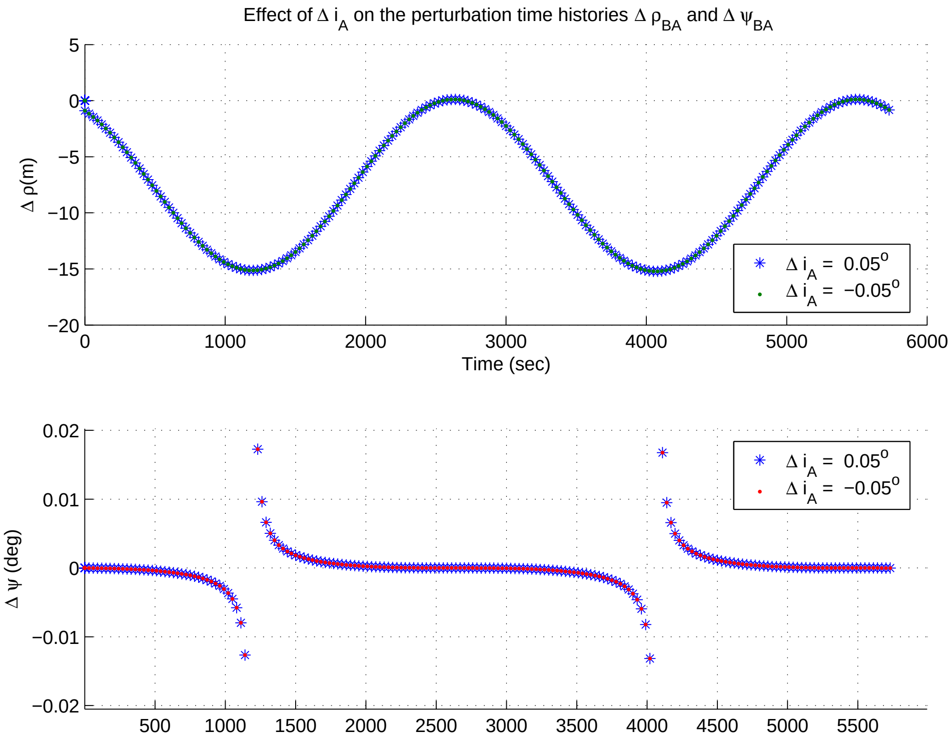

Fig. 32. Perturbations of range and angle caused by perturbation of inclination

measurements: If the target direction is normal to the observer plane normal, then the target is at the intersection point at this time. Given the intersection point of the two orbits, and the plane of one of them, the plane of the second orbit cab calculated geometrically.

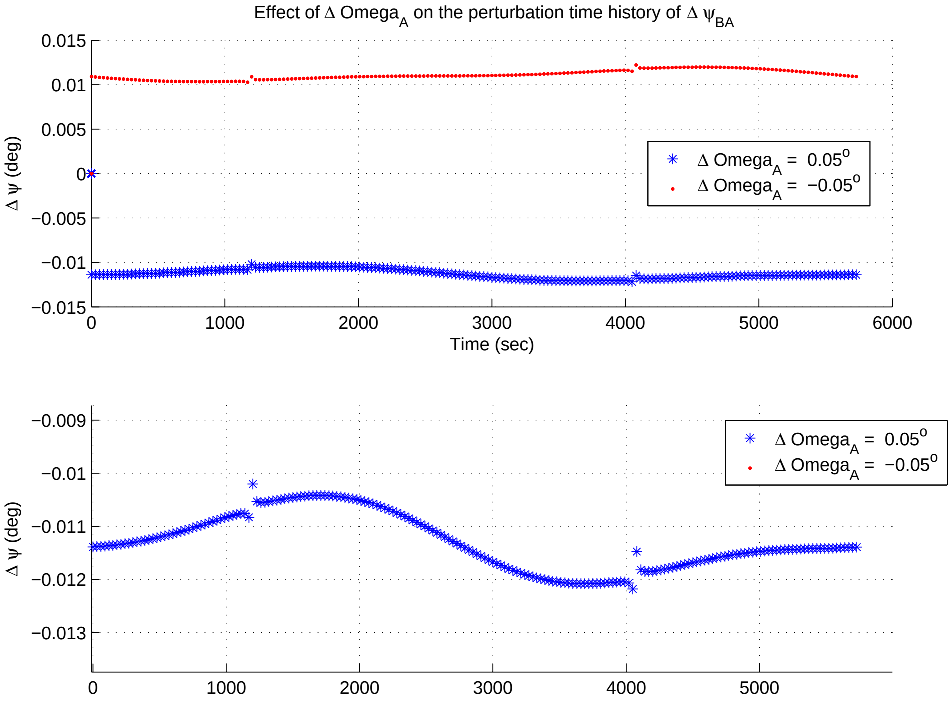

Fig. 33. Perturbations of angle caused by perturbation of Ω_A

From the above discussion, it can be concluded that all states can be estimated given a history of measurements. However, of importance is the duration of these measurements. Looking again at the previous figures, one can recognize that a short history of measurements, say 1000 *seconds*, may not tell whether the perturbed ψ_{AB} is, in this short period, a part of a ramp or a harmonic wave or even a constant with some noise added to it. It is concluded from that the measurements history should cover enough time to distinguish between different behaviors in the measured ψ_{AB}. This measurements period will depend on the two orbits of the two spacecrafts and some special cases may have special requirements. From the case studied above, one may conclude that a safe period of time is half the orbital period, or little less than that. As a demonstration for this concluded results; a GLSDC technique is used to estimate the states of a space object and the measurements are assumed ideal with no errors. Several cases are considered for different measurements duration period to see after how long measurements period can the estimator reach the true values. Consider the case of a LEO with eccentricity of 0.05, $h_p = 300\ Km$, and inclination of 95^o. Assume that the observing spacecraft is flying in an orbit with the same parameters but circular. Results are shown in figures 34-36. Figure 34 shows that the semi major length and eccentricity could not be estimated even from ideal measurements in 20 minutes. Even if with more measurements, 140 compared to 40 in the previous case, but still within the 20 minutes, we cannot estimate the true values, Fig. 35. However, if this period is increased to half of the orbit, we get the true values as shown in Fig. 36. Reasonable measurements rate is still needed to accurately detect the harmonic behaviors in the measurements.

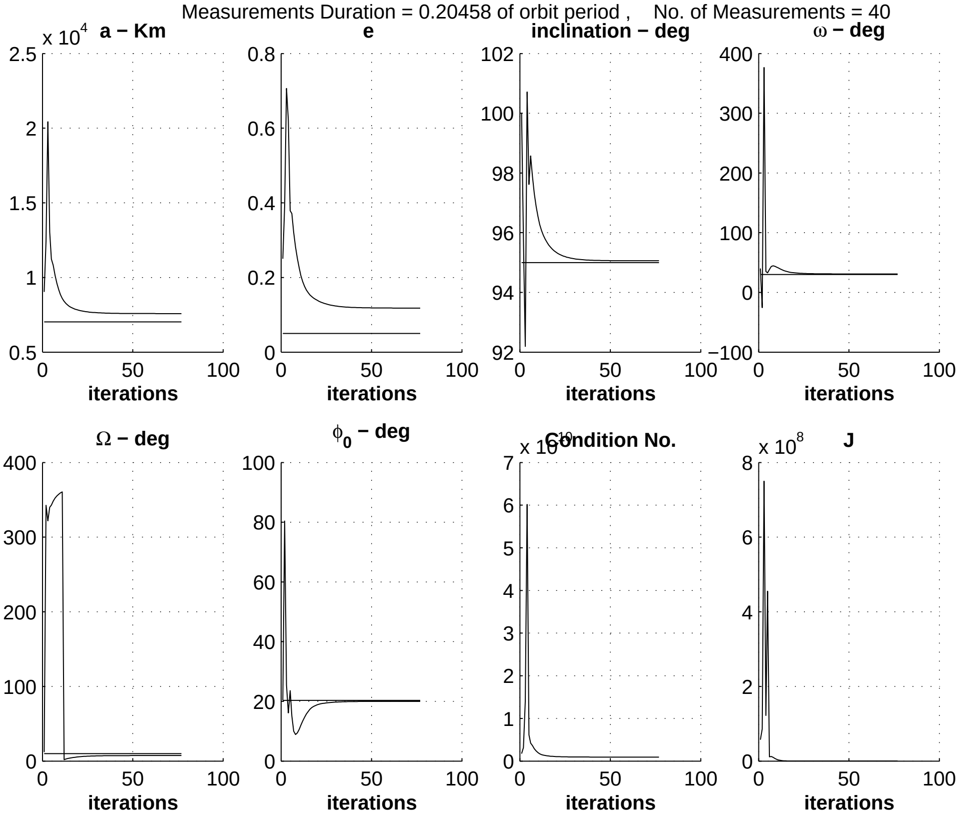

Fig. 34. Ideal measurements, duration = 20 min. no. of measurements = 40

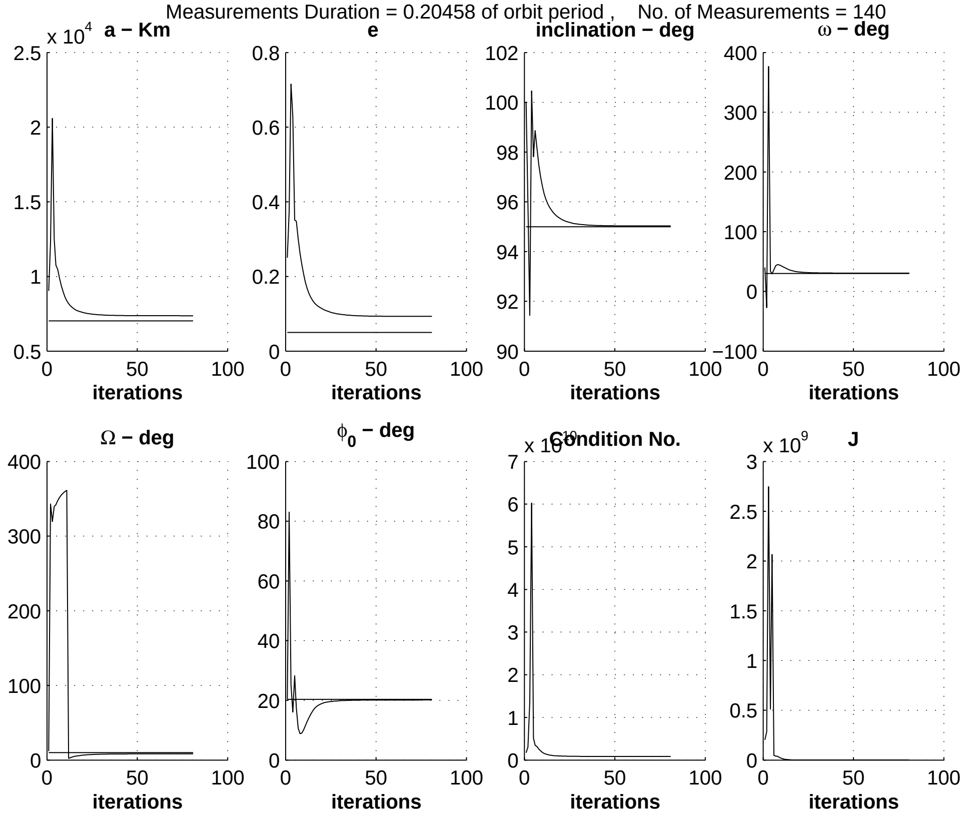

Fig. 35. Ideal measurements, duration = 20 min. no. of measurements = 140

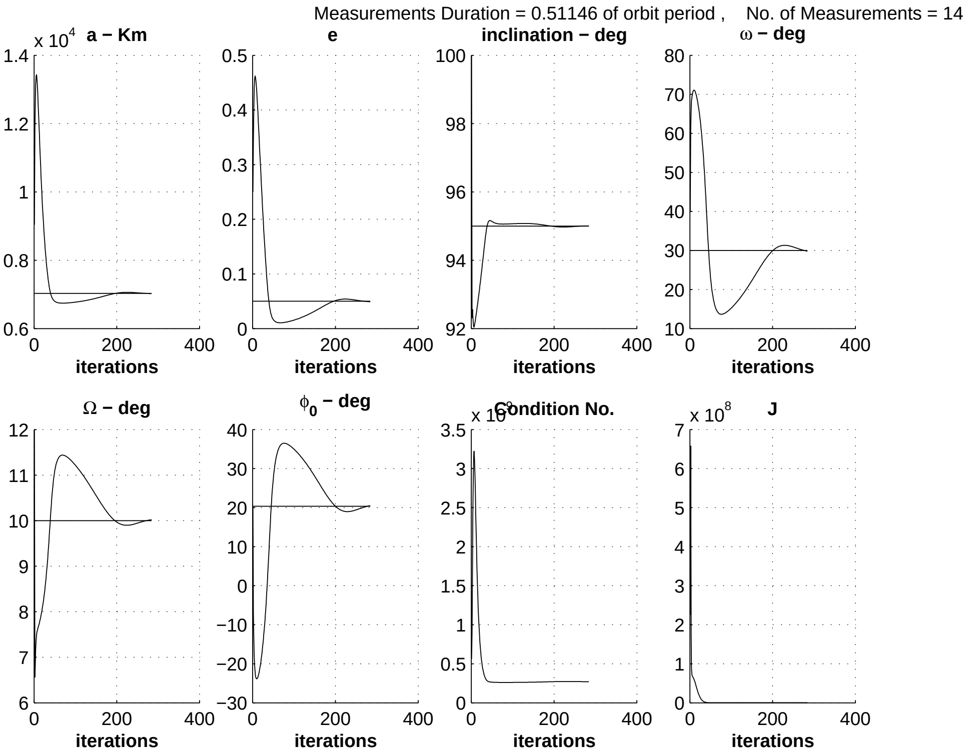

Fig. 36. Ideal measurements, duration = 50 min. no. of measurements = 140

2. Observability of the Linear States Model

If the distance between the two spacecraft is small compared to the orbit size, and the orbit of the spacecraft is near circular, then the linear Hills equations [9] can be used to represent the relative motion of the two spacecraft:

$$\begin{aligned} \ddot{x} - 2\omega\dot{y} - 3\omega^2 x &= f_x \\ \ddot{y} + 2\omega\dot{x} &= f_y \\ \ddot{z} + \omega^2 z &= f_z \end{aligned} \tag{6.1}$$

where x, y, and z are the three components of the object relative position w.r.t the observing spacecraft in the later's coordinate system. f_x, f_y, and f_z are the external forces applied to the object spacecraft. For the purpose of observability analysis, consider the solution of the above equations in the case of zero external forces [34] which can be written in a compact form as:

$$\begin{aligned} x(t) &= f_1(x_0, \dot{x}_0, y_0, \dot{y}_0, t) \\ y(t) &= f_2(x_0, \dot{x}_0, y_0, \dot{y}_0, t) \\ z(t) &= f_3(z_0, \dot{z}_0, t) \end{aligned} \tag{6.2}$$

The states that need to be estimated are the initial conditions: $x_0, \dot{x}_0, y_0, \dot{y}_0, z_0$, and $\dot{z}_0$. The measurement is the direction of the object spacecraft which can be considered as a unit vector in the direction of the object. So, a measurement $\mathbf{m}_k$ at time t_k is:

$$\mathbf{m}_k = \frac{1}{\sqrt{f_{1k}^2 + f_{2k}^2 + f_{3k}^2}} \begin{bmatrix} f_{1k} \\ f_{2k} \\ f_{3k} \end{bmatrix} \tag{6.3}$$

where, f_{ik} is the function f_i evaluated at time t_k.

Given one measurement vector, Eq.(6.3) will give you three equations in six unknowns. Given two measurements, then we will have six equations in six unknowns. This set of nonlinear equations may have a single solution or multiple solutions. Additional measurements can be used to solve the ambiguities in the solution. So, from a mathematical point of view, This linear model is observable if we have two or more measurements.

C. Gaussian Least Squares Differential Correction

The state variable vector $\mathbf{x}$ is selected to be:

$$\mathbf{x} = [a, e, \imath, \omega, \Omega, \phi_0]^T \tag{6.4}$$

where ϕ_0 is the true anomaly of the first measurement. The position of the object is $\mathbf{r}_o$. The position of the star tracker is $\mathbf{r}_s$. The position of the object with respect to the star tracker is $\mathbf{r}_{o/s}$. See Fig. 37 for geometry of the space surveillance problem. The model for m measurements, $\tilde{\mathbf{y}}_j$, is:

$$\mathbf{y}_j = \mathbf{f}_j(\mathbf{x}), \quad j = 1 \ldots m \tag{6.5}$$

$$\mathbf{f} = \frac{\mathbf{r}_{o/s}}{\rho}, \qquad \rho = \|\mathbf{r}_{o/s}\| \tag{6.6}$$

So, Given a set of measurements $\tilde{\mathbf{y}}$, it is required to find the state vector $\mathbf{x}$. The

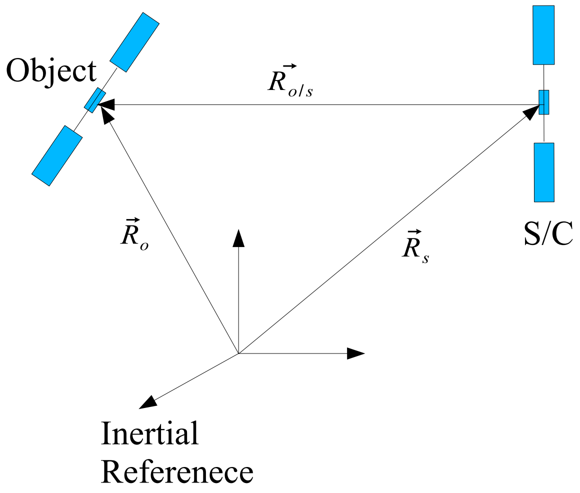

Fig. 37. Geomtery for space surveillance

motion of the object is assumed to follow a Keplarian orbit. The method of GLSDC is implemented.

The least squared error cost function,

$$J = \frac{1}{2}\,\mathbf{e}^{\mathrm{T}}\,W\,\mathbf{e} = \frac{1}{2}\,[\,\tilde{\mathbf{y}} - \hat{\mathbf{y}}\,]^{\mathrm{T}}\,W\,[\,\tilde{\mathbf{y}} - \hat{\mathbf{y}}\,],$$

is minimized by repeated state updates of the form

$$\Delta\mathbf{x} = (H^{\mathrm{T}}\,W\,H)^{-1}\,H^{\mathrm{T}}\,W\,[\,\tilde{\mathbf{y}} - \mathbf{f}(\hat{\mathbf{x}}_i)\,]$$

where $\hat{\mathbf{x}}_i$ is the current estimate of the state, $H = \left.\frac{\partial\mathbf{f}}{\partial\mathbf{x}}\right|_{\mathbf{x}_i}$ is the Jacobian matrix, W is the weighting matrix, and $(H^{\mathrm{T}}\,W\,H)^{-1}$ is the auto-correlation matrix. If the weighting matrix is the inverse of the measurement variances, then the auto-correlation matrix becomes the error covariance.

D. Jacobian Matrix Derivation

The jacobian matrix, H, is by definition:

$$H = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \tag{6.7}$$

Differentiating Eq.(6.6),

$$\rho \frac{\partial \mathbf{f}}{\partial \mathbf{x}} + \mathbf{f} \frac{\partial \rho}{\partial \mathbf{x}}^T = \frac{\partial \mathbf{r}_{o/s}}{\partial \mathbf{x}} \tag{6.8}$$

where,

$$\rho^2 = (\mathbf{r}_{o/s})^T \mathbf{r}_{o/s} \tag{6.9}$$

$$\frac{\partial \rho}{\partial \mathbf{x}} = \frac{1}{\rho} \left(\frac{\partial \mathbf{r}_{o/s}}{\partial \mathbf{x}} \right)^T \mathbf{r}_{o/s} \tag{6.10}$$

So, to evaluate H, we need only to calculate $\frac{\partial \mathbf{r}_{o/s}}{\partial \mathbf{x}}$. In fact, since $\mathbf{r}_s$ is not a function of $\mathbf{x}$, then:

$$\frac{\partial \mathbf{r}_{o/s}}{\partial \mathbf{x}} = \frac{\partial \mathbf{r}_o}{\partial \mathbf{x}} \tag{6.11}$$

The inertial position of the object at time t_j at which the measurement j was taken can be expressed as a function of the six orbital elements as follows [32]:

$$(\mathbf{r}_o)_j = \xi(\imath, \Omega, \omega) \begin{bmatrix} a\left(\cos(\psi_j) - e\right) \\ a\sqrt{1-e^2}\sin(\psi_j) \\ 0 \end{bmatrix} \tag{6.12}$$

where,

$$\xi(\imath, \Omega, \omega) = [A_z(\Omega)]^T [A_x(\imath)]^T [A_z(\omega)]^T \tag{6.13}$$

and

$$[A_z(\omega)] = \begin{bmatrix} \cos(\omega) & \sin(\omega) & 0 \\ -\sin(\omega) & \cos(\omega) & 0 \\ 0 & 0 & 1 \end{bmatrix}, [A_x(\imath)] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\imath) & \sin(\imath) \\ 0 & -\sin(\imath) & \cos(\imath) \end{bmatrix} \tag{6.14}$$

Differentiating Eq.(6.12) w.r.t. each of the six elements of $\mathbf{x}$:

$$\frac{\partial(\mathbf{r}_o)_j}{\partial a} = \xi(\imath, \Omega, \omega) \begin{bmatrix} (\cos(\psi_j) - e) \\ \sqrt{1-e^2}\sin(\psi_j) \\ 0 \end{bmatrix} \tag{6.15}$$

$$\frac{\partial(\mathbf{r}_o)_j}{\partial e} = \xi(\imath, \Omega, \omega) \begin{bmatrix} -a \\ \dfrac{-ae\sin(\psi_j)}{\sqrt{1-e^2}} \\ 0 \end{bmatrix} \tag{6.16}$$

$$\frac{\partial(\mathbf{r}_o)_j}{\partial \imath} = \frac{\partial \xi}{\partial \imath} \begin{bmatrix} a\,(\cos(\psi_j) - e) \\ a\sqrt{1-e^2}\sin(\psi_j) \\ 0 \end{bmatrix} \tag{6.17}$$

$$\frac{\partial(\mathbf{r}_o)_j}{\partial \omega} = \frac{\partial \xi}{\partial \omega} \begin{bmatrix} a\,(\cos(\psi_j) - e) \\ a\sqrt{1-e^2}\sin(\psi_j) \\ 0 \end{bmatrix} \tag{6.18}$$

$$\frac{\partial(\mathbf{r}_o)_j}{\partial \Omega} = \frac{\partial \xi}{\partial \Omega} \begin{bmatrix} a\,(\cos(\psi_j) - e) \\ a\sqrt{1-e^2}\sin(\psi_j) \\ 0 \end{bmatrix} \tag{6.19}$$

where,

$$\frac{\partial \xi}{\partial \Omega} = \frac{\partial [A_z(\Omega)]^T}{\partial \Omega} [A_x(\imath)]^T [A_z(\omega)]^T \tag{6.20}$$

$$\frac{\partial \xi}{\partial \omega} = [A_z(\Omega)]^T [A_x(\imath)]^T \frac{\partial [A_z(\omega)]^T}{\partial \omega} \tag{6.21}$$

$$\frac{\partial \xi}{\partial \Omega} = [A_z(\Omega)]^T \frac{\partial [A_x(\imath)]^T}{\partial \Omega} [A_z(\omega)]^T \tag{6.22}$$

and,

$$\frac{\partial[A_z(\omega)]^T}{\partial\omega} = \begin{bmatrix} -\sin(\omega) & -\cos(\omega) & 0 \\ \cos(\omega) & -\sin(\omega) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \frac{\partial[A_x(\imath)]^T}{\partial\imath} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -\sin(\imath) & -\cos(\imath) \\ 0 & \cos(\imath) & -\sin(\imath) \end{bmatrix} \tag{6.23}$$

The derivative of $(\mathbf{r}_o)_j$ w.r.t. the true anomaly at the first measurement can be evaluated as follows:

$$\frac{\partial(\mathbf{r}_o)_j}{\partial\phi_0} = \frac{\partial(\mathbf{r}_o)_j}{\partial\psi_j}\frac{\partial\psi_j}{\partial\psi_0}\frac{\partial\psi_0}{\partial\phi_0} \tag{6.24}$$

To calculate these derivatives, recall that:

$$\tan(\frac{\psi_0}{2}) = \sqrt{\frac{1-e}{1+e}}\tan(\frac{\phi_0}{2}) \tag{6.25}$$

differentiating this equation, then:

$$\frac{\partial\psi_0}{\partial\phi_0} = \sqrt{\frac{1-e}{1+e}}\frac{\sec^2(\frac{\phi_0}{2})}{\sec^2(\frac{\psi_0}{2})} \tag{6.26}$$

Recall also that:

$$n(t_1 - t_p) = \psi_0 - e\sin(\psi_0) \tag{6.27}$$

$$n(t_j - t_p) = \psi_j - e\sin(\psi_j) \tag{6.28}$$

Subtracting Eq.(6.27) from Eq.(6.28), then:

$$\psi_j - e\sin(\psi_j) - \psi_0 + e\sin(\psi_0) = n(t_j - t_1) \tag{6.29}$$

Differentiating Eq.(6.29) w.r.t. ψ_0 and taking into account that the difference in time between the two measurements, $t_j - t_1$, does not depend on ψ_0:

$$\frac{\partial\psi_j}{\partial\psi_0} = \frac{1 - e\cos(\psi_0)}{1 - e\cos(\psi_j)} \tag{6.30}$$

Differentiating Eq.(6.12) w.r.t. ψ_j:

$$\frac{\partial(\mathbf{r}_o)_j}{\partial\psi_j} = \xi(\imath,\Omega,\omega)\begin{bmatrix} -a\sin(\psi_j) \\ a\sqrt{1-e^2}\cos(\psi_j) \\ 0 \end{bmatrix} \tag{6.31}$$

Combining (6.30),(6.31), and (6.26):

$$\frac{\partial(\mathbf{r}_o)_j}{\partial\psi_0} = \xi(\imath,\Omega,\omega)\begin{bmatrix} -a\sin(\psi_j) \\ a\sqrt{1-e^2}\cos(\psi_j) \\ 0 \end{bmatrix}\frac{1-e\cos(\psi_0)}{1-e\cos(\psi_j)}\sqrt{\frac{1-e}{1+e}}\frac{\sec^2(\frac{\phi_0}{2})}{\sec^2(\frac{\psi_0}{2})} \tag{6.32}$$

Equations (6.15)-(6.19) and (6.32) construct $\frac{\partial\mathbf{r}_o}{\partial\mathbf{x}}$. The last in turn is substituted into equations (6.11), (6.10), and (6.8) to find H.

E. Results and Discussion

The above formulation is coded and the results show that this algorithm does not converge. The condition number for the matrix $(H^T H)$ starts in the first iteration with a value in the order of 10^8 and rapidly increases as the number of iterations increase.

A small modification to the above formulation is done to try to get it to converge. A virtual measurement vector is assumed to be the object inertial vector. This virtual measurements vector is calculated from the actual measurements vector as follows:

$$\begin{aligned}(\mathbf{r}_o)_j &= (\mathbf{r}_s)_j + (\mathbf{r}_{o/s})_j \\ &= (\mathbf{r}_s)_j + \rho_j\mathbf{y}_j \qquad j = 1\ldots m\end{aligned} \tag{6.33}$$

The length ρ_j in Eq.(6.33) is calculated based on the current estimate for the state vector $\mathbf{x}$. This enhances the previous model with more information, the range of the object spacecraft from the star tracker. However, this information is based on an estimate not actual measurements. The Jacobian matrix, H, is slightly modified. It is in this case, $H = \dfrac{\partial \mathbf{r}_o}{\partial \mathbf{x}}$. This modified algorithm converges. The solution provided by this algorithm is described and analyzed through the discussion of the following figures. The figures below show the convergence history and the number of iterations required for convergence. The true values of the states are plotted as horizontal lines in figures. In all figures presented, the model for the measurements errors is assumed to be random error in the satellite inertial position. The random error has zero mean. The maximum error is assumed to be a 100 Km in each direction of the three inertial coordinate system.

The first case presented is a case with a 2000 measurements, collected 5 times per second. This is about 7 minutes of spacecraft flight. So the duration of measurements in this case is about 7 minutes. The results for different values of iterations are shown in Fig. 38 and Fig. 39. In Fig. 38, the iterations were truncated after 40 iterations. As seen from figure, results encourage to continue with more iterations to converge to the true value. The number of iterations is increased to 150 iterations and the new results are plotted in Fig. 39. The results show a closer convergence to the true values. This can lead to the conclusion that with more iterations, a good estimate can be achieved. The drawback is the running time. With this big number of measurements, 2000 measurements, the Jacobian matrix becomes huge and the running time for more than 200 iterations may take a day on a personal computer.

After discussion with Dr. Junkines [16], he advised to increase the total time span of observations rather than worrying about the time between measurements. So, the number of measurements is reduced to be only 70 and the time between measurements is increased to be 33 seconds. The measurements duration is then about 39 minutes of observations. This resulted in decreasing the running time significantly since the

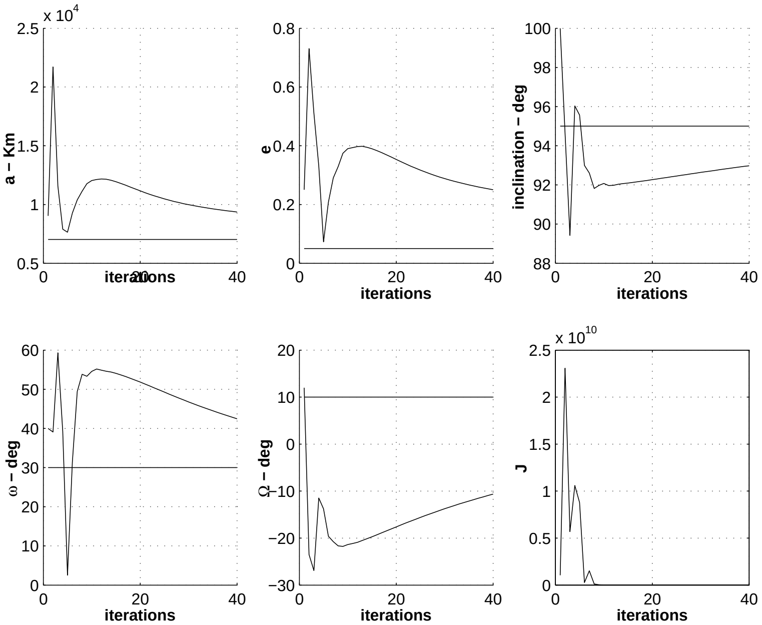

Fig. 38. GLSDC, measurements collected 5 times/second

number of measurements is much smaller. The results of this case are shown in Fig. 40 and Fig. 41. Figure 40 shows a good convergence to a close to the true values. The inclination of the orbit is almost exactly correct. Other states converge to a steady state value that is biased from the true value. For the purpose of analysis, the same case is run again but thus time with ideal measurements. Zero measurements errors are assumed. The only error is the initial guess. Results of this run are shown in Fig. 41. All states almost converges very close to the true values. It can be concluded that the biased estimate of the states is due to the model of measurements errors.

The duration of measurements was chosen to be 39 minutes in the previous case. To see how small can we go with this figure, the case of 15 minutes of measurements duration is considered. Several runs are performed and the results showed instability in the solution depending on the number of measurements and on the measurements

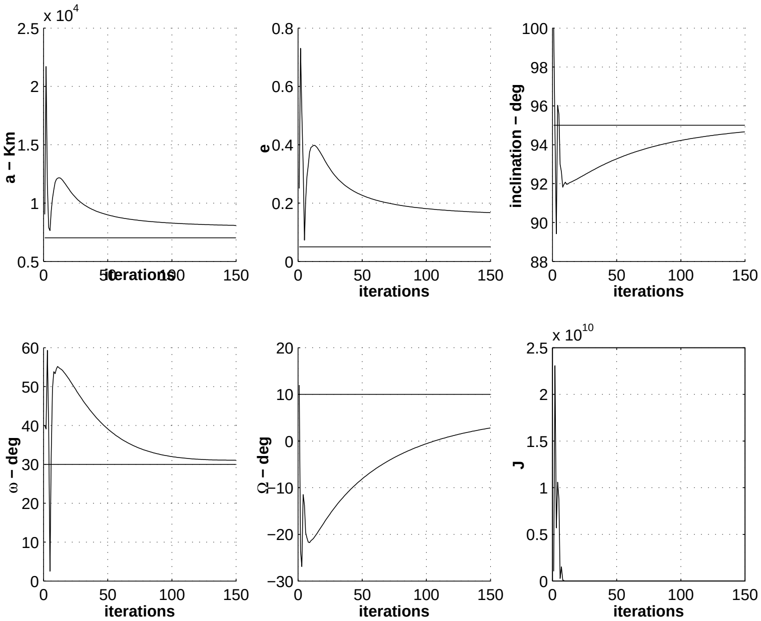

Fig. 39. GLSDC, measurements collected 5 times/second

errors which are random. As a result, with only 15 minutes of measurements, sometimes we can get a convergence and some times not. One example of a case that converged is shown in Fig. 42. If we increase the time of measurements duration to 20 minutes which is about 20% of the true orbital period, the algorithm will converge independent from the number of measurements. Three cases are plotted for a number of measurements of 30, 40, and 80. All converged to a close value to the true states. The condition number history for the matrix (H^TWH) is plotted. It is always below the order of 10^{10}. As the number of measurements increase, a better accuracy is achieved. This can be figured out by comparing Fig. 43, Fig. 44, and Fig. 45. The accuracy improvement is clear in the states ϕ_0 and ω. Orbit inclination is always accurately achieved.

If the error in the measurements is changed to be Gaussian with a maximum of

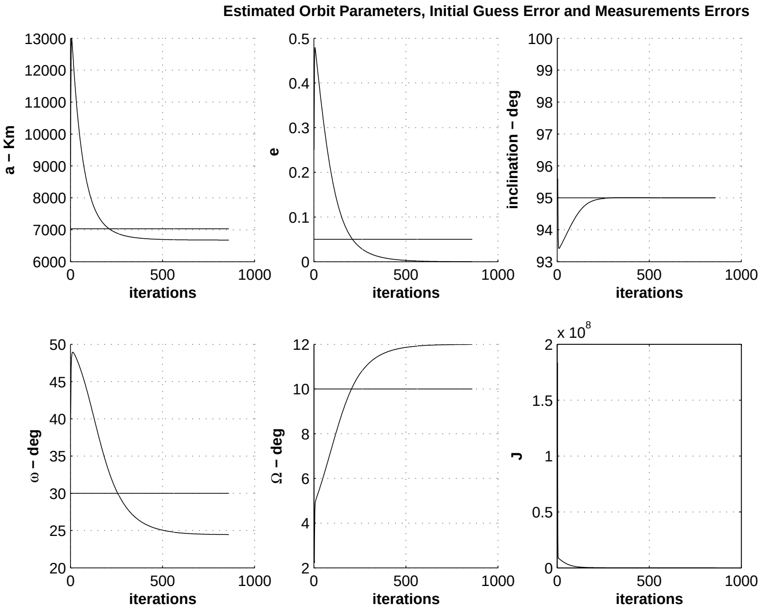

Fig. 40. GLSDC, measurements collected 2 times/min. for 39 minutes

1 km in each coordinate direction, then the residual error in the estimated orbit will be much improved. Fig. 46 shows the results for this case. Fig. 47 shows the results for the same case but with more measurements collected.

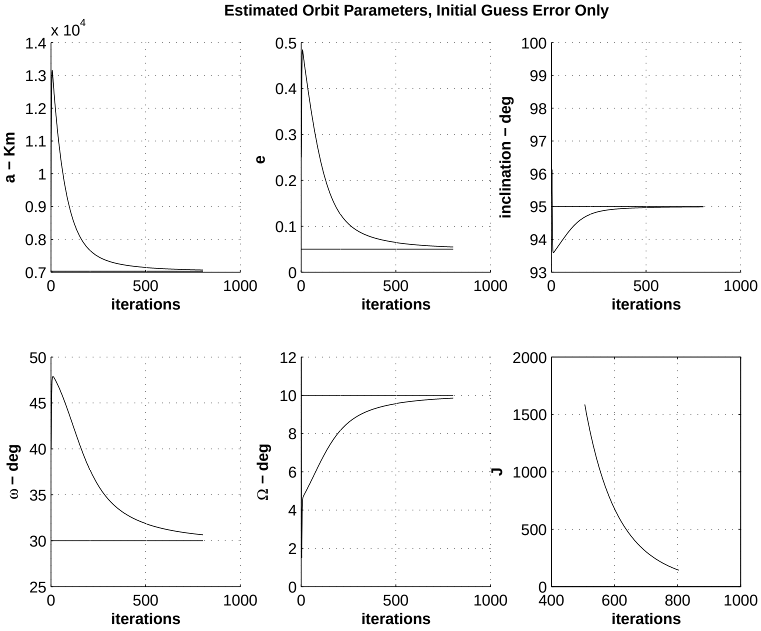

Fig. 41. GLSDC, measurements collected 2 times/min. for 39 minutes

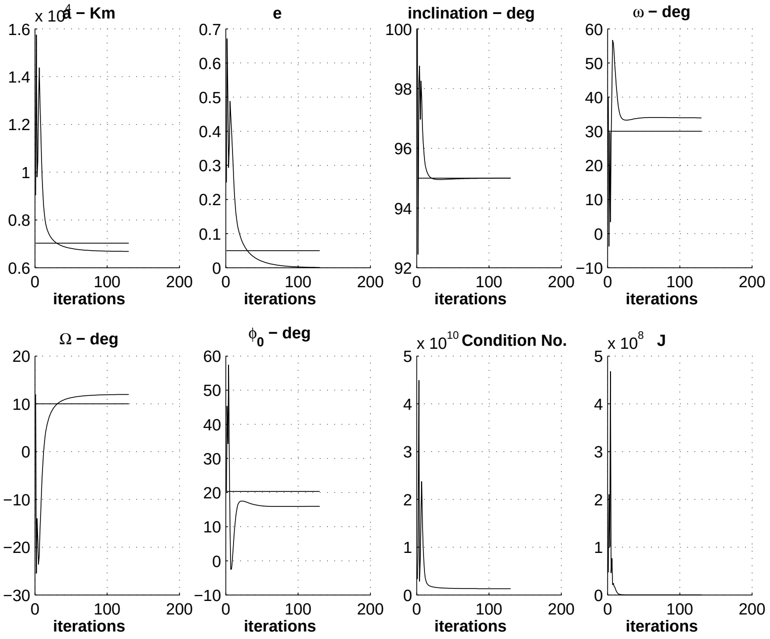

Fig. 42. GLSDC, measurements duration = 15 min. no. of measurements = 40

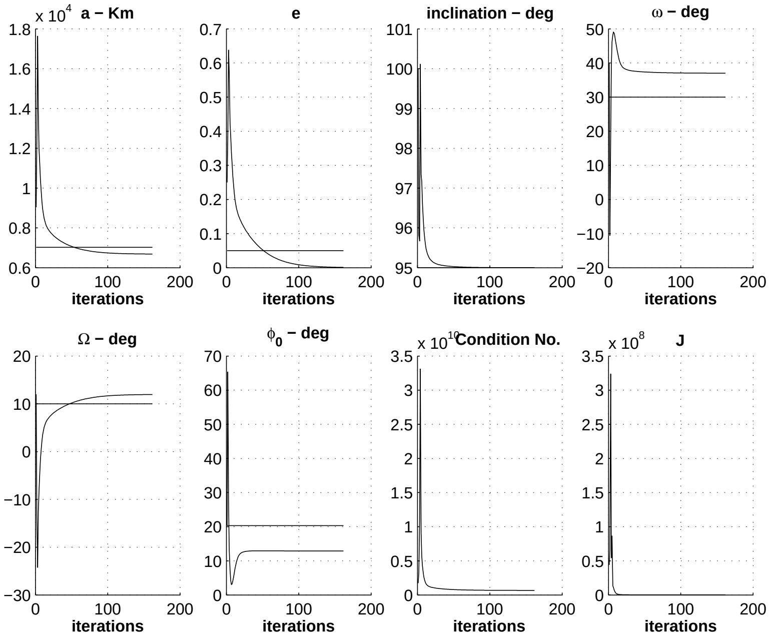

Fig. 43. GLSDC, measurements duration = 20 min. no. of measurements = 30

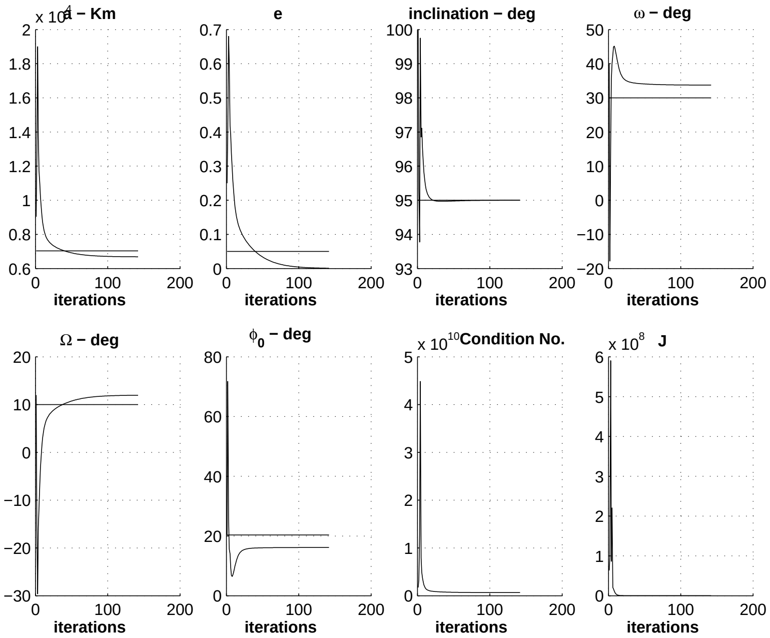

Fig. 44. GLSDC, measurements duration = 20 min. no. of measurements = 40

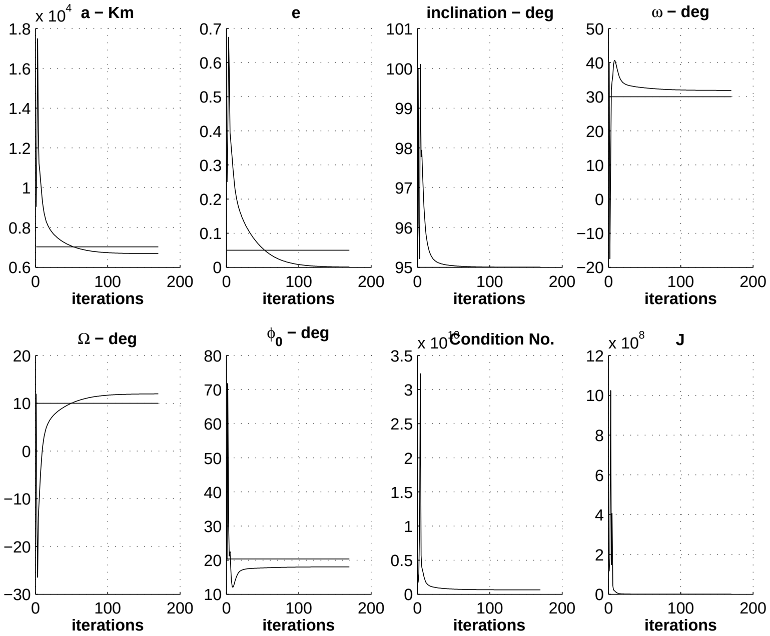

Fig. 45. GLSDC, measurements duration = 20 min. no. of measurements = 80

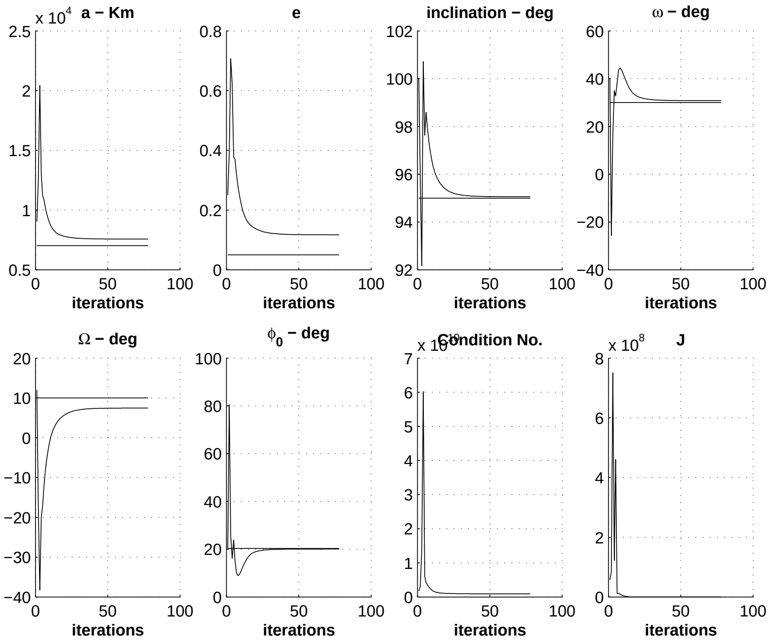

Fig. 46. GLSDC, measurements duration = 20 min. no. of measurements = 40

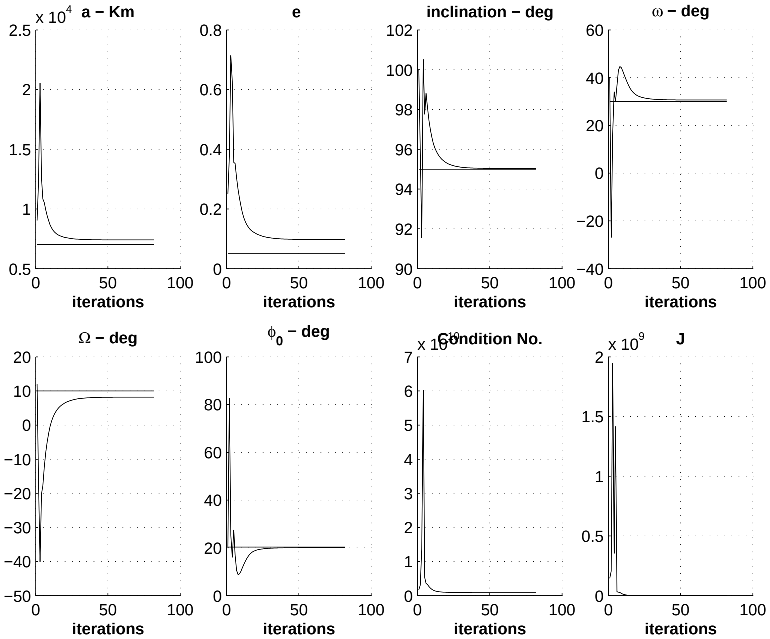

Fig. 47. GLSDC, measurements duration = 20 min. no. of measurements = 100

F. Iterated Kalman Filter

In this section the extended Kalman filter technique [11] is implemented for estimation. The state vector is chosen to be the initial position and velocity vectors. The truth model used is:

$$\ddot{\vec{r}} = -\frac{\mu}{\|\vec{r}(t)^3\|}\vec{r}(t) + w(t) \tag{6.34}$$

$w(t)$ is the process noise, which is assumed to be zero. The Extended Kalman Filter converges much faster than the least square technique implemented in the previous section. The extended Kalman filter is used to process data forward with guessed initial conditions, and then process the data backward. Initial conditions for the backward pass are the final states of the forward pass. Each iteration consists of a forward and a backward pass. Figure 48 shows the estimated initial position and velocity using the iterated Kalman filter.

G. Conclusions

The orbit of a space target can be estimated using the measurements of a star tracker. An important factor is how long is the time span in which the measurements are taken. Initial analysis performed in this study shows that the orbit parameters can be estimated if the measurements cover a time span of the order of 10% to 20% of the orbital period, depending on the target orbit. The target orbit plane however can be estimated in much less time span, which is in the order of 5% to 10% of the orbital period.

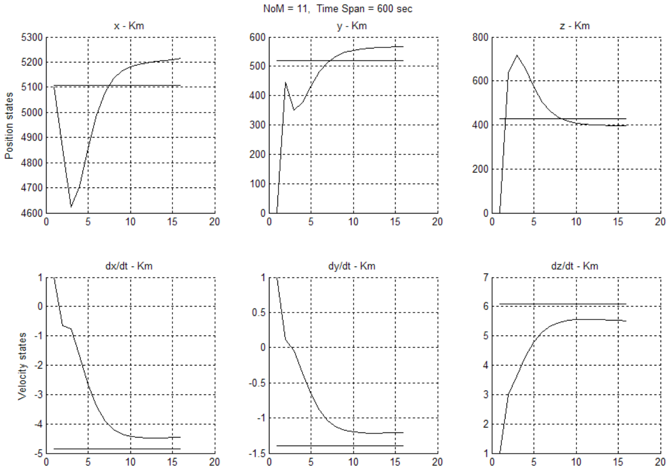

Fig. 48. Iterated Kalman filter. time span = 10 min, no. of measurements = 11

CHAPTER VII

CONCLUSIONS

In this dissertation, the problem of finding the optimal orbit for a remote sensing mission is addressed. In Chapter II, the optimal orbit that enables a spacecraft to visit a set of targets on the surface of the Earth within an assigned time frame is calculated. The genetic algorithms technique is implemented to optimize the multi minima cost function. The main advantage of the proposed solution is that it is a fuel free solution. Other solutions to this problem use thrusters to maneuver the spacecraft between sites. The solution to this problem mainly depend on the mission objectives. Two case studies are considered. The first is a mission which objective is to get a maximum resolution image for each visited site. The second is a mission which objective is to maximize the observation time for each visited site. The solution provided by the genetic algorithms is then used as initial guess in a gradient method to find the local optimum solution near the genetic algorithms solution. Results obtained demonstrate the effectiveness of the developed tool to find good solutions to the problem.

If the resulting solution is not satisfactory for a particular mission, a multi-orbit solution is proposed. This solution has two orbits to visit all the target sites. The satellite is maneuvered between the two orbits. Chapter IV addresses the problem of designing the optimal orbit maneuver using impulsive thrusters using genetic algorithms. A new formulation for the problem was developed. The new formulation has advantages to other current formulations in literature. It has a few number of design variables. Only three design variables are optimized in the general case of a transfer between two non-coplanar orbits. This few number of design variables results in a very efficient algorithm in terms of the calculations time. This efficiency enables to explore more in the design space, searching for the global optimal, and so increasing the probability of finding the global optimal solution. The developed formulation also guarantees that the optimal solution is achieved even if the final solution is not the global optimal. The non-optimality appears as more flight time during the transfer and more required fuel for the transfer but the spacecraft will arrive exactly at the final orbit. The developed formulation was validated through a number of cases and proved to find the optimal solution in all cases.

The problem of having two spacecraft visit the same target site at the same time is addressed in Chapter V. conditions on the orbits parameters are derived analytically such that: the two spacecraft move in compatible orbits that intersect each other, the two spacecraft pass through the intersection points at the same times, and the two ground tracks are tangent to each other at the intersection points. Two cases are considered. The first is the special case where the intersection points are the perigee and apogee of the two orbits. The second is a general case where the intersection points are any points on the orbits. The case of two satellites in a Flower Constellation is investigated and results demonstrated possible existence of general Two-way orbits in the Flower Constellation set.

The problem of estimating the orbit of a space object orbiting the Earth using only star tracker measurements is addressed in Chapter VI. This space object could be another spacecraft. Star trackers are able to identify spacecraft and provide a direction to the target spacecraft buy not the range. Given a history of observations from a star tracker onboard an observing spacecraft to a space object, a The Gaussian Least Squares Differential Correction technique is implemented to estimate the object orbit. First, the observability of the system is investigated to check the possibility of estimating the system states from the measurements. Two observability checks are performed. The first is the observability analysis for the nonlinear dynamical system, where the nonlinear model for spacecraft motions is implemented. The second is the observability analysis for the linear system of Hills equations assuming that the observing and the observed spacecraft are close to each other. The measurements in this model, however, are not linear functions in the states. The results of the observability analysis show that for the linear model, all states are observable. For the nonlinear model, the states are observable under the condition that enough time span of measurements is available. This enough time span will be no less than 20% of the orbital period. Analytical derivations for the Jacobian matrix is developed. Several cases are studied with different time span periods. The results of the developed estimator validates the results of the observability analysis.

REFERENCES

[1] Abdelkhalik, O., “Remote Sensing Satellites Orbits Design and Control,” Master’s thesis, Cairo University, Cairo, Egypt, 2001.

[2] Bäck, T., *Evolutionary Algorithms in Theory and Practice*, New York: Oxford University Press, 1996.

[3] Battin, R. H., *An Introduction to the Mathematics and Methods of Astrodynamics*, New York: AIAA, 1987.

[4] Brown, C.D., *Spacecraft Mission Design*, 2nd Edition, Reston, VA: AIAA, 1998.

[5] Bruccoleri, C., M. Wilkins, and D. Mortari, “On Sun-Synchronous Orbits and Associated Constellations,” in *6th conference of Dynamics and Control Systems and Structures in Space*, Riomaggiore, Italy, 2004, pp. 12–25.

[6] Carter, D.,“When Is the Groundtrack Drift Rate Zero,” in CSDL Memorandum ESD-91-020, Cambridge, MA: Charles Stark Draper Laboratory, 1991.

[7] Chen, H. S., *Space Remote Sensing Systems, An Introduction*, 16th edition, Orlando, FL: Academic Press, 1985.

[8] Chobotov, V. A., *Orbital Mechanics*, Reston, VA: AIAA, 2002.

[9] Clohessy, W.H. and Wiltshire, R.S., “Terminal Guidance System for Satellite Rendezvous,” *Journal of the Aerospace Sciences*, pp. 653–658, September, 1960.

[10] Coly, D. A., *An Introduction to Genetic Algorithms for Scientists and Engineers*, Singapore: World Scientific, 1999.

[11] Crassidis, J. L. and Junkins, J. L., *Optimal Estimation of Dynamic Systems*, Boca Raton, FL: Chapman&Hall/CRC, 2004.

[12] Goldberg, D.E., *Genetic Algorithms in Search, Optimization, and Machine Learning*, Reading, MA: Addison-Wesley, 1989.

[13] Gray, P. and Hart, W. and Painton, L. and Phillips, C. and Trahan, M. and Wagner, J. , "A Survey of Global Optimization Methods," http://www.cs.sandia.gov/opt/survey/main.html.

[14] Griffin, M.D. and French, J. R., *Space Vehicle Design*, Washington DC: AIAA, 1991.

[15] Guelman, M., and A. Kogan, "Electric Propulsion for Remote Sensing from Low Orbits," *Journal of Guidance, Control and Dynamics*, vol. 22, no. 2, pp. 313–321, 1999.

[16] Junkins, J., private communications, July 2005.

[17] Kim, Y.H. and D.B. Spencer, "Optimal Spacecraft Rendezvous Using Genetic Algorithms," *Journal of Spacecraft and Rockets*, vol. 39, no. 6, pp. 859–865, 2002.

[18] Koza., J.R., "Introduction to Genetic Programming," in *Advances in Genetic Programming*. Kinnear, E. K.,(ed.), Cambridge, MA: The MIT Press, 1994.

[19] McCue, C. "Optimum Two-Impulse Orbital Trnafer and Rendezvous between Inclined Orbits," *AIAA Journal,* vol. 1, no. 8, pp. 1865–1872, 1963.

[20] Mortari, D., O. Abdelkhalik, and C. Bruccoleri, "Synodic and Relative Flower Constellations with Applications for Planetary Explorations," in *Space Flight*

Mechanics Conference, AIAA/AAS, Copper Mountain, CO, AAS, January 2005, pp. 123–137.

[21] Mortari, D., C. Bruccoleri, S. La Rosa, and J. Junkins, “CCD Data Processing Improvements,” in *International Conference on Dynamics and Control of Systems and Structures in Space*, King College, Cambridge, England, July 14–18, 2002, pp. 67–79.

[22] Mortari, D., M. Wilkins,and C. Bruccoleri, “The Flower Constellations,” in *The John L. Junkins Astrodynamics Symposium*, Sponsoring Organization, College Station, TX, 2004, pp. 107–127.

[23] Ogata, K., *Modern Control Engineering*, 2nd edition, New Delhi, India: Prentice Hall of India, 1991.

[24] Park, K., M. Ruggieri, and D. Mortari, “Comparisons Between GalileoSat and Global Navigation Flower Constellations,” in *IEEE Aerospace Conference*, IEEE, Big Sky, MO, March 2005, pp. 43–56.

[25] Park, K., M. Wilkins, C. Bruccoleri, and D. Mortari, “Uniformly Distributed Flower Constellation Design Study for Global Positioning System,” in *Space Flight Mechanics Conference*, AIAA/AAS, Maui, HI, AAS, January 2004, pp. 87–106.

[26] Psiaki, M. L., “Autonomous Orbit Determination for Two Spacecraft from Relative Position Measurements,” *Journal of Guidance, Control, and Dynamics*, vol. 22 , no. 2, pp. 305–312, March-April, 1999.

[27] Rauwolf, G.A. and V. Coverstone-Carroll, “Near-Optimal Low-Thrust Orbit

Transfer Generated by a Genetic Algorithms," *Journal of Spacecraft and Rockets*, vol. 33, no. 6, pp. 859–862, 1996.

[28] Reichert, A. K., "Using a Genetic Algorithm to Determine the Optimum Two-Impulse Transfer Between Co-Planar, Elliptical Orbits," in *Industrial Applications of Geneteic Algorithms*. Karr, C. L. and Freeman, L. M., Danvers, (eds.), Boca Raton, FL: CRC Press, 1999, pp. 111–133.

[29] Sabol, C. and R. Culp,"Improved Angular Observations in Geosynchronous Orbit Determination," *AIAA Journal of Guidance, Navigation, and Dynamics*, vol. 24, no. 1, pp. 123–130, 2001.

[30] Sabol, C., T. Kelecy, and M. Murai,"Geosynchronous Orbit Determination Using the High Accuracy Network Determination System (HANDS)," in *AAS/AIAA Spaceflight Mechanics Conference*, AIAA/AAS, Maui, HI, AAS 04-216, February, 2004, pp. 61–73.

[31] Schaffer, J.D., R. A. Caruana,and L. J. Eshelman,"A Study of Control Parameters Affecting Online Performance of Genetic Algorithms for Function Optimisation," in 11^{th} *International Joint Conference on Artificial Intelligence*, Detroit, MI, 1989, pp750–755.

[32] Sidi, M. J., *Spacecraft Dynamics and Control, A Practical Engineering Approach*, New York: Cambridge University Press, 1997.

[33] Singla, P., T. Griffith, J. Crassidis, and J. Junkins, "Attitude Determination and Autonomous On-orbit Calibration of Star Tracker for the GIFTS Mission," AAS, Monterey, California, January, 2002, pp. 11–29..

[34] Vallado, D. A., *Fundamental of Astrodynamics and Applications*,2nd. edition, El Segundo, CA: STL, 2001.

[35] Vallado, D. A. and S. S. Carter, "Accurate Orbit Determination from Short-Arc Dense Observational Data," *Journal of the Astronautical Sciences*, vol. 46, no. 2, pp. 195–213, 1998.

[36] Visser, B., C. Sabol, and S. Dahlke,"Geosynchronous Orbit Determination Using High Accuracy Angular Obeservations," in *AAS/AIAA Spaceflight Mechanics Conference*, AIAA/AAS, Copper Mountain, CO, AAS 05-135, January, 2005.

[37] Wilkins, M., "Characterizing Orbit Uncertainty due to Atmospheric Uncertainty," Master's thesis, Texas A&M University, College Station, TX, 2000.

[38] Wilkins, M., "The Flower Constellations – Theory, Design Process, and Applications," Ph.D. dissertation, Texas A&M University, College Station, Texas, 2004.

[39] Wilkins, M.P., C. Bruccoleri,and D. Mortari,"Constellation Design Using Flower Constellations," in *Space Flight Mechanics Conference*, AIAA/AAS, Maui, HI, AAS, January 2004, pp. 27–42..

[40] Wilkins, M.P., and D. Mortari,"Constellation Design via Projection of an Arbitrary Shape onto a Flower Constellation Surface," in *AIAA/AAS Astrodynamics Specialist Conference*, Providence, Rhode Island, 2004, pp. 53–65.

APPENDIX A

CODE FOR ESTIMATION OF SPACE TARGETS ORBITS

This is the code for the Estimation of Space Targets Orbits using Star Tracker measurements.

```
function observ_check() clear ;
% This function invistigates the observability of the target orbit
% parameters from the star tracker measurements

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   DATA                             %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
global mu ;
Re = 6378.14 ;    % Earth Radius in Km
mu = 3.986012e5 ;   % EARTH GRAVITAIONAL CONSTANT Km^3/sec^2
d2r = pi/180 ; ref_direction = [1;0;0];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Nominal Observing Spacecrfat Orbit A%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
hp = 515 ; % Perigee Altitude in Km
ha = 585 ; % Apogee  Altitude in Km
a_A = Re + 0.5*(hp + ha) ;
e_A = 1 - (hp+Re)/a_A ; i_A = 90 ; omega_A = 10*d2r ;
Omega_A = 0 ; phi_0_A = 0*d2r ; % True anomay at epoch
psi_0_A = Tanom2Eanom(phi_0_A,e_A) ;
M_0_A = psi_0_A - e_A*sin(psi_0_A) ;
n_A = sqrt(mu/a_A^3) ; PeriodA = 2*pi/n_A ; % Orbit Period in seconds
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Nominal Observed Spacecrfat Orbit B%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
hp = 515 ; % Perigee Altitude in Km
ha = 585 ; % Apogee  Altitude in Km
a_B = Re + 0.5*(hp + ha) ; e_B = 1 - (hp+Re)/a_B ; i_B = 90 ;
omega_B = 10*d2r ;
Omega_B = 0 ; phi_0_B = 10*d2r ;  % True anomay at epoch
psi_0_B = Tanom2Eanom(phi_0_B,e_B) ;
 M_0_B = psi_0_B - e_B*sin(psi_0_B) ;
n_B = sqrt(mu/a_B^3) ; PeriodB = 2*pi/n_B ; % Orbit Period in seconds
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Orbit Propagation                 %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
time_step = 0.5*60  ; for time_count = 0 :time_step : PeriodA
```

```
    M_A = M_0_A + n_A*time_count  ;
    [psi_A, phi_A] = kepler2 (M_A, e_A) ;
    [r_A, v_A] =
    orb2eci([a_A ; e_A ; i_A; omega_A ; Omega_A ; phi_A]) ;
    M_B = M_0_B + n_B*time_count  ;
    [psi_B, phi_B] = kepler2 (M_B, e_B) ;
    [r_B, v_B] =
    orb2eci([a_B ; e_B ; i_B; omega_B ; Omega_B ; phi_B]) ;
    r_B_A = r_B - r_A ;
    rho(time_count+1) = norm(r_B_A);
    psi_angle = get_psi(r_B_A,ref_direction,d2r) ;
    psi_v(time_count+1) = psi_angle ;
    count(time_count+1) = time_count ;
end ;

DMO_observ(PeriodA,M_0_A,n_A,a_A ,e_A ,i_A,omega_A ,...
Omega_A,M_0_B,n_B,a_B,e_B,i_B,omega_B,Omega_B,d2r,...
time_step,rho,psi_v,ref_direction) ;

De_observ(PeriodA,M_0_A,n_A,a_A ,e_A ,i_A,omega_A ,...
Omega_A,M_0_B,n_B,a_B,e_B,i_B,omega_B,Omega_B,d2r,...
time_step,rho,psi_v,ref_direction) ;

Da_observ(PeriodA,M_0_A,n_A,a_A ,e_A ,i_A,omega_A ,...
Omega_A,M_0_B,n_B,a_B,e_B,i_B,omega_B,Omega_B,d2r,...
time_step,rho,psi_v,ref_direction) ;

Di_observ(PeriodA,M_0_A,n_A,a_A ,e_A ,i_A,omega_A ,...
Omega_A,M_0_B,n_B,a_B,e_B,i_B,omega_B,Omega_B,d2r,...
time_step,rho,psi_v,ref_direction) ;

DO_observ(PeriodA,M_0_A,n_A,a_A ,e_A ,i_A,omega_A ,...
 Omega_A,M_0_B,n_B,a_B,e_B,i_B,omega_B,Omega_B,d2r,...
time_step,rho,psi_v,ref_direction) ;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function Da_observ(PeriodA,M_0_A,n_A,a_A ,e_A ,i_A,omega_A ,...
Omega_A,M_0_B,n_B,a_B,e_B,i_B,omega_B,Omega_B,d2r,...
time_step,rho,psi_v,ref_direction) ;

global mu ;

n_A_temp = n_A ; n_B_temp = n_B ;
dn_A = 2.8e-5  ;
% dn_B =  dn_A   ;
dn_B =  0   ; n_A = n_A + dn_A ;
n_B = n_B + dn_B ; a_A = (mu/n_A^2)^(1/3) ;
a_B = (mu/n_B^2)^(1/3) ;

for time_count =0 : time_step : PeriodA
```

```
    M_A = M_0_A +  n_A*time_count  ;
    [psi_A, phi_A] = kepler2 (M_A, e_A) ;
    [r_A, v_A] =
    orb2eci([a_A ; e_A ; i_A; omega_A ; Omega_A ; phi_A]) ;
    M_B = M_0_B + n_B*time_count  ;
    [psi_B, phi_B] = kepler2 (M_B, e_B) ;
    [r_B, v_B] =
    orb2eci([a_B ; e_B ; i_B; omega_B ; Omega_B ; phi_B]) ;
    r_B_A = r_B - r_A ;
    rho_n(time_count+1) = norm(r_B_A);
    psi_vn(time_count+1) = get_psi(r_B_A,ref_direction,d2r) ;
    v1(time_count+1) =
    psi_v(time_count+1) -psi_vn(time_count+1) ;
    if abs(abs(v1(time_count+1)) - 360) < 10
        v1(time_count+1) = v1(time_count+1) - 360 ;
    end ;
    count(time_count+1) = time_count ;
end ; clear count ; dn_A = -2.8e-5     ;
% dn_B =   -dn_A   ;
dn_B =  0   ; n_A = n_A_temp + dn_A ; n_B = n_B_temp + dn_B ;

a_A = (mu/n_A^2)^(1/3) ; a_B = (mu/n_B^2)^(1/3) ;

for time_count = 0 : time_step : PeriodA
    M_A = M_0_A +  n_A*time_count  ;
    [psi_A, phi_A] = kepler2 (M_A, e_A) ;
    [r_A, v_A] =
    orb2eci([a_A ; e_A ; i_A; omega_A ; Omega_A ; phi_A]) ;
    M_B = M_0_B +  n_B*time_count  ;
    [psi_B, phi_B] = kepler2 (M_B, e_B) ;
    [r_B, v_B] =
    orb2eci([a_B ; e_B ; i_B; omega_B ; Omega_B ; phi_B]) ;
    r_B_A = r_B - r_A ;
    rho_n1(time_count+1) = norm(r_B_A);
    psi_vn1(time_count+1) = get_psi(r_B_A,ref_direction,d2r) ;
    v2(time_count+1) =
    psi_v(time_count+1)-psi_vn1(time_count+1) ;
    if abs(abs(v2(time_count+1)) - 360) < 10
        v2(time_count+1) = v2(time_count+1) + 360 ;
    end ;
    count(time_count+1) = time_count ;
end ;


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function De_observ(PeriodA,M_0_A,n_A,a_A ,e_A ,i_A,omega_A ,...
 Omega_A,M_0_B,n_B,a_B,e_B,i_B,omega_B,Omega_B,d2r,...
time_step,rho,psi_v,ref_direction) ;

e_A_temp = e_A ; e_B_temp = e_B ;
```

```
de_A = -1.7e-4  ;
% de_B =  -de_A   ;
de_B =  0   ; e_A = e_A + de_A ; e_B = e_B + de_B ;
for time_count = 0 : time_step : PeriodA
    M_A = M_0_A + n_A*time_count  ;
    [psi_A, phi_A] = kepler2 (M_A, e_A) ;
    [r_A, v_A] =
    orb2eci([a_A ; e_A ; i_A; omega_A ; Omega_A ; phi_A]) ;
    M_B = M_0_B + n_B*time_count  ;
    [psi_B, phi_B] = kepler2 (M_B, e_B) ;
    [r_B, v_B] =
    orb2eci([a_B ; e_B ; i_B; omega_B ; Omega_B ; phi_B]) ;
    r_B_A = r_B - r_A ;
    rho_n(time_count+1) = norm(r_B_A);
    psi_vn(time_count+1) =
    get_psi(r_B_A,ref_direction,d2r) ;
    count(time_count+1) = time_count ;
end ; clear count ; de_A = 1.7e-4  ;
% de_B =  de_A   ;
de_B =  0   ; e_A = e_A_temp + de_A ;
e_B = e_B_temp + de_B ;

for time_count = 0 : time_step : PeriodA
    M_A = M_0_A + n_A*time_count  ;
    [psi_A, phi_A] = kepler2 (M_A, e_A) ;
    [r_A, v_A] =
    orb2eci([a_A ; e_A ; i_A; omega_A ; Omega_A ; phi_A]) ;
    M_B = M_0_B + n_B*time_count  ;
    [psi_B, phi_B] = kepler2 (M_B, e_B) ;
    [r_B, v_B] =
    orb2eci([a_B ; e_B ; i_B; omega_B ; Omega_B ; phi_B]) ;
    r_B_A = r_B - r_A ;
    rho_n1(time_count+1) = norm(r_B_A);
    psi_vn1(time_count+1) =
    get_psi(r_B_A,ref_direction,d2r) ;
    count(time_count+1) = time_count ;
end ;


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function Di_observ(PeriodA,M_0_A,n_A,a_A ,e_A ,i_A,omega_A ,...
Omega_A,M_0_B,n_B,a_B,e_B,i_B,omega_B,Omega_B,d2r,...
time_step,rho,psi_v,ref_direction) ;

global mu ;

i_A_temp = i_A ; i_B_temp = i_B ; di_A = 0.05*d2r  ;
% di_B =  di_A   ;
di_B =  0   ; i_A = i_A + di_A ; i_B = i_B + di_B ;
```

```
for time_count = 0 : time_step : PeriodA
    M_A = M_0_A +  n_A*time_count  ;
    [psi_A, phi_A] = kepler2 (M_A, e_A) ;
    [r_A, v_A] =
    orb2eci([a_A ; e_A ; i_A; omega_A ; Omega_A ; phi_A]) ;
    M_B = M_0_B + n_B*time_count  ;
    [psi_B, phi_B] = kepler2 (M_B, e_B) ;
    [r_B, v_B] =
    orb2eci([a_B ; e_B ; i_B; omega_B ; Omega_B ; phi_B]) ;
    r_B_A = r_B - r_A ;
    rho_n(time_count+1) = norm(r_B_A);
    psi_vn(time_count+1) = get_psi(r_B_A,ref_direction,d2r) ;
    v1(time_count+1) = psi_v(time_count+1) -psi_vn(time_count+1) ;
%     if abs(abs(v1(time_count+1)) - 360) < 10
%         v1(time_count+1) = v1(time_count+1) + 360 ;
%     end ;
    count(time_count+1) = time_count ;
end ; clear count ; di_A = -0.05*d2r  ;
% di_B =   -di_A   ;
di_B =  0   ; i_A = i_A_temp + di_A ;
i_B = i_B_temp + di_B ;

for time_count = 0 : time_step : PeriodA
    M_A = M_0_A +  n_A*time_count  ;
    [psi_A, phi_A] = kepler2 (M_A, e_A) ;
    [r_A, v_A] =
    orb2eci([a_A ; e_A ; i_A; omega_A ; Omega_A ; phi_A]) ;
    M_B = M_0_B +  n_B*time_count  ;
    [psi_B, phi_B] = kepler2 (M_B, e_B) ;
    [r_B, v_B] =
    orb2eci([a_B ; e_B ; i_B; omega_B ; Omega_B ; phi_B]) ;
    r_B_A = r_B - r_A ;
    rho_n1(time_count+1) = norm(r_B_A);
    psi_vn1(time_count+1) = get_psi(r_B_A,ref_direction,d2r) ;
    v2(time_count+1) = psi_v(time_count+1)-psi_vn1(time_count+1) ;
    if abs(abs(v2(time_count+1)) - 360) < 10
        v2(time_count+1) = v2(time_count+1) - 360 ;
    end ;
    count(time_count+1) = time_count ;
end ;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function DM0_observ(PeriodA,M_0_A,n_A,a_A ,e_A ,i_A,omega_A ,...
Omega_A,M_0_B,n_B,a_B,e_B,i_B,omega_B,Omega_B,d2r,...
time_step,rho,psi_v,ref_direction) ;

dM_0_A = 0.01*d2r  ;
% dM_0_B =  dM_0_A   ;
dM_0_B =  0 ; for time_count = 0 : time_step : PeriodA
```

```
    M_A = M_0_A + dM_0_A + n_A*time_count  ;
    [psi_A, phi_A] = kepler2 (M_A, e_A) ;
    [r_A, v_A] =
    orb2eci([a_A ; e_A ; i_A; omega_A ; Omega_A ; phi_A]) ;
    M_B = M_0_B + dM_0_B + n_B*time_count  ;
    [psi_B, phi_B] = kepler2 (M_B, e_B) ;
    [r_B, v_B] =
    orb2eci([a_B ; e_B ; i_B; omega_B ; Omega_B ; phi_B]) ;
    r_B_A = r_B - r_A ;
    rho_n(time_count+1) = norm(r_B_A);
    psi_vn(time_count+1) =
    get_psi(r_B_A,ref_direction,d2r) ;
    count(time_count+1) = time_count ;
end ; clear count ; dM_0_A = -0.01*d2r  ;
% dM_0_B =  -dM_0_A   ;
dM_0_B =  0 ; for time_count = 0 : time_step : PeriodA
    M_A = M_0_A + dM_0_A + n_A*time_count  ;
    [psi_A, phi_A] = kepler2 (M_A, e_A) ;
    [r_A, v_A] =
    orb2eci([a_A ; e_A ; i_A; omega_A ; Omega_A ; phi_A]) ;
    M_B = M_0_B + dM_0_B + n_B*time_count  ;
    [psi_B, phi_B] = kepler2 (M_B, e_B) ;
    [r_B, v_B] =
    orb2eci([a_B ; e_B ; i_B; omega_B ; Omega_B ; phi_B]) ;
    r_B_A = r_B - r_A ;
    rho_n1(time_count+1) = norm(r_B_A);
    psi_vn1(time_count+1) =
    get_psi(r_B_A,ref_direction,d2r) ;
    count(time_count+1) = time_count ;
end ;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function DO_observ(PeriodA,M_0_A,n_A,a_A ,e_A ,i_A,omega_A ,...
Omega_A,M_0_B,n_B,a_B,e_B,i_B,omega_B,Omega_B,d2r,...
time_step,rho,psi_v,ref_direction) ;

global mu ;

Omega_A_temp = Omega_A ; Omega_B_temp = Omega_B ;
dOmega_A = 0.05*d2r  ;
% dOmega_B =  dOmega_A   ;
dOmega_B =  0   ; Omega_A = Omega_A + dOmega_A ;
Omega_B = Omega_B + dOmega_B ;

for time_count = 0 : time_step : PeriodA
    M_A = M_0_A +  n_A*time_count  ;
    [psi_A, phi_A] = kepler2 (M_A, e_A) ;
    [r_A, v_A] =
    orb2eci([a_A ; e_A ; i_A; omega_A ; Omega_A ; phi_A]) ;
    M_B = M_0_B + n_B*time_count  ;
    [psi_B, phi_B] = kepler2 (M_B, e_B) ;
    [r_B, v_B] =
    orb2eci([a_B ; e_B ; i_B; omega_B ; Omega_B ; phi_B]) ;
    r_B_A = r_B - r_A ;
    rho_n(time_count+1) = norm(r_B_A);
    psi_vn(time_count+1) = get_psi(r_B_A,ref_direction,d2r) ;
    v1(time_count+1) =
    psi_v(time_count+1) -psi_vn(time_count+1) ;
    if abs(abs(v1(time_count+1)) - 360) < 10
        v1(time_count+1) = v1(time_count+1) + 360 ;
    end ;
    count(time_count+1) = time_count ;
end ; clear count ; dOmega_A = -0.05*d2r  ;
% dOmega_B =   -dOmega_A   ;
dOmega_B =  0   ; Omega_A = Omega_A_temp + dOmega_A ;
Omega_B = Omega_B_temp + dOmega_B ;

for time_count = 0 : time_step : PeriodA
    M_A = M_0_A +  n_A*time_count  ;
    [psi_A, phi_A] = kepler2 (M_A, e_A) ;
    [r_A, v_A] =
    orb2eci([a_A ; e_A ; i_A; omega_A ; Omega_A ; phi_A]) ;
    M_B = M_0_B +  n_B*time_count  ;
    [psi_B, phi_B] = kepler2 (M_B, e_B) ;
    [r_B, v_B] =
    orb2eci([a_B ; e_B ; i_B; omega_B ; Omega_B ; phi_B]) ;
    r_B_A = r_B - r_A ;
    rho_n1(time_count+1) = norm(r_B_A);
    psi_vn1(time_count+1) =
    get_psi(r_B_A,ref_direction,d2r) ;
    v2(time_count+1) = psi_v(time_count+1)-psi_vn1(time_count+1) ;
    if abs(abs(v2(time_count+1)) - 360) < 10
        v2(time_count+1) = v2(time_count+1) - 360 ;
    end ;
    count(time_count+1) = time_count ;
end ;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function angle = get_psi(r_B_A,ref_direction,d2r) ;

angle = acos((dot(r_B_A,ref_direction)/(norm(r_B_A)*...
norm(ref_direction))))/d2r ;

v3 = cross(r_B_A,ref_direction) ;

if v3(3) > 0
    angle = -angle ;
end ;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
function eanom = Tanom2Eanom(tanom,e)

s_eanom = sin(tanom)*sqrt(1 - e^2)/(1+e*cos(tanom)) ;
c_eanom = (e + cos(tanom))/(1+e*cos(tanom)) ;

eanom = atan2(s_eanom,c_eanom) ;
```

VITA

Osama Mohamed Omar Abdelkhalik graduated from Cairo University, Egypt, with a Bachelor of Science degree in aerospace engineering in May 1996. After graduation he joined the National Authority for Remote Sensing and Space Sciences in Egypt as a research engineer. He received a Master of Science degree in aerospace engineering from Cairo University, Egypt in 2001. He joined the Department of Aerospace Engineering at Texas A&M University in fall 2003. His research interests are in the field of orbital mechanics, optimization and control systems. He can be contacted at: 34 Maher Haroun St., Ardellewaa, Embaba, Giza, Egypt. His email: ossama_omer@hotmail.com.

The typist for this thesis was Ossama Abdelkhalik.