Time-Varying Lyapunov Control Laws with Enhanced Estimation of Distribution Algorithm for Low-Thrust Trajectory Design

Abolfazl Shirazi, Harry Holt, Roberto Armellin and Nicola Baresi

Abstract Enhancements in evolutionary optimization techniques are rapidly growing in many aspects of engineering, specifically in astrodynamics and space trajectory optimization and design. In this chapter, the problem of optimal design of space trajectories is tackled via an enhanced optimization algorithm within the framework of Estimation of Distribution Algorithms (EDAs), incorporated with Lyapunov and Q-law feedback control methods. First, both a simple Lyapunov function and a Q-law are formulated in Classical Orbital Elements (COEs) to provide a closed-loop low-thrust trajectory profile. The weighting coefficients of these controllers are approximated with various degrees of Hermite interpolation splines. Following this model, the unknown time series of weighting coefficients are converted to unknown interpolation points. Considering the interpolation points as the decision variables, a black-box optimization problem is formed with transfer time and fuel mass as the objective functions. An enhanced EDA is proposed and utilized to find the optimal variation of weighting coefficients for minimum-time and minimum-fuel transfer trajectories. The proposed approach is applied in some trajectory optimization problems of Earth-orbiting satellites. Results show the efficiency and the effectiveness of the proposed approach in finding optimal transfer trajectories. A comparison between the Q-law and simple Lyapunov controller is done to show the potential of the potential of the EEDA in enabling the simple Lyapunov controller to recover the finer nuances explicitly given within the analytical expressions in the Q-law.

Key words: Global Optimisation, Interpolation, Q-law, Estimation of Distribution Algorithms, Evolutionary Approach

A. Shirazi
Basque Center for Applied Mathematics - BCAM, Bilbao, 48009, Spain, e-mail: ashirazi@bcamath.org

H. Holt
Surrey Space Centre, University of Surrey, GU2 7XH, Guildford, United Kingdom, e-mail: h.holt@surrey.ac.uk

R. Armellin
Te Pūnaha Ātea - Auckland Space Institute, University of Auckland, Auckland, New Zealand, e-mail: roberto.armellin@auckland.ac.nz

N. Baresi
Surrey Space Centre, University of Surrey, GU2 7XH, Guildford, United Kingdom, e-mail: n.baresi@surrey.ac.uk

CHAPTER (Author's version)
Modeling and Optimization in Space Engineering, vol. 200, pp. 377–399 (2023)
DOI: 10.1007/978-3-031-24812-2_14

List of Acronyms

- EA - Evolutionary Algorithm
- EDA - Estimation of Distribution Algorithm
- EEDA - Enhanced Estimation of Distribution Algorithm
- GA - Genetic Algorithm
- PSO - Particle Swarm Optimization
- Q-law - Proximity Quotient Control Law

1 Introduction

Low-thrust many-revolution trajectory design and orbit transfer are becoming increasingly important with the development of high specific impulse, low-thrust engines such as electric propulsion (EP) systems. Finding an optimal transfer trajectory is a challenging task due to the non-linearity of the system's dynamics and the problem complexity. Many approaches have been developed so far to overcome the difficulty of finding optimal transfer trajectories of the space systems. In general, these can be divided into one of two categories: indirect and direct methods. Direct methods convert a continous optimal control problem into a parameter optimization problem, often via discretization and subsequent transcription, to find an approximate solution to the original problem. Indirect methods, on the other hand, use calculus of variation to reduce the optimal control problem to the solution of a two-point boundary value problem [1–3]. Both techniques can be used to solve low-thrust trajectory design problems, however, they are computationally intensive and still present many challenges. Direct methods generate a large optimisation problem and result in approximate solutions. Indirect methods can produce rigorously optimal solutions, but are intensive due to the unbounded the need for a good initial guess and the difficulties in handling discontinuous controls. Both methods provide point solutions, i.e. for the assumed initial condition, and cannot be used as guidance laws due to time limitations and difficulties in ensuring convergence [3].

In recent years, advances in artificial intelligence and evolutionary computations has shifted the attention of the aerospace community towards the employment of evolutionary algorithms (EAs) in spacecraft trajectory optimization [3]. The motivation for utilizing EAs is based on their ability in dealing with local optimal region of the solution domain and handling nonlinear constraints that naturally appear in nonlinear optimal control problems. Development of novel EAs in this applications covers the vast types of space missions. Such developments are mainly within the framework of well-known EAs. For instance, in [4] a fuzzy goal programming-based is developed. The proposed algorithm is a hybrid technique based on the combination of a gradient-based method and Genetic Algorithm (GA). This algorithm is used to solve an optimal flight-path design for a constrained multi-objective aero-assisted vehicle trajectory optimization problem. An improved NSGA-II algorithm is developed in [5] for solving non-coplanar orbits transfers in multi-impulse Lambert rendezvous problems. The proposed algorithm benefits from a self-adaptive differential evolution technique to increase the efficiency of the algorithm. In another recent research by Pontani [6], Particle Swarm Optimization (PSO) is utilized in an indirect approach based on Pontryagin principle. This approach is used to solve low-thrust Earth to Mars trajectory optimization problem. Englander and Conway [7] presented a modified GA, and incorporated it in low-thrust interplanetary trajectory optimization problem. Algorithm modification in this research is towards the elitism operator within GA which preserves the best members of the population and encourages diversity at the same time. Many other algorithm enhancements have been proposed for various type of space applications including, satellite formation [8], mission planning [9], asteroid exploration [10], and orbit determination [11].

Following the progress in evolutionary methods with application in astrodynamics, it can be observed that the algorithm improvements are mostly based on the traditional evolutionary algorithms. One of the frameworks that did not receive much attention in spacecraft trajectory optimization is Estimation of Distribution Algorithms (EDAs) [12]. EDAs are a class of EAs. EDAs are a class of EAs that work based on probabilistic models. In an EDA, a probabilistic model is learned at each iteration and new solutions are sampled from that model. The obtained solutions have similar characteristics as those used for learning the model. One of the characteristics of EDAs is to have an explicit description of the promising solutions in terms of probabilistic models. Due to this feature, they have a great potential for enhancement toward further improvements. This characteristic is the main motivation in this research and the effort here is to enhance the mechanisms of EDAs for obtaining higher quality solutions in spacecraft trajectory optimization.

The emergence of closed-loop feedback-driven (CLFD) controllers, particularly those based on Lyapunov control theory, has allowed the computation of sub-optimal trajectories with minimal computational cost [13–15]. The control profile is readily available as they only require knowledge of the current and target spacecraft states to compute it, making them suitable as initial guesses for indirect and direct methods [16,17]. However, as they treat the problem from a targeting perspective, Lyapunov controllers are inherently sub-optimal and often have many user-defined parameters which significantly affect their performance [18–20]. One method for improving their performance involves EA. Both Lee et al. 2005 [18] and Varga et al. 2016 [19] used a multi-objective genetic algorithm to optimise the Petropoulos's Lyapunov-based Q-law design parameters for a variety of Earth orbit transfers, with the design parameters remaining fixed throughout the transfer. Yang et al. [21] used an artificial neural network and improved cooperative evolutionary algorithm optimiser to make the design parameters of a Lyapunov-based Q-law state-dependent.

In this research, both a simple Lyapunov function and a Q-law are formulated in Classical Orbital Elements (COEs) to provide a closed-loop low-thrust trajectory profile. Both control formulations are considered in order to access whether the EEDA is capable of recovering the finer nuances embedded within the analytical expressions in the Q-law when using a simple Lyapunov controller. The optimal variations of weighting coefficients are interpolated via Hermite interpolation. The time series of weighting coefficients are turned into decision variables and a black-box optimization problem is formed. Having the transfer time and the fuel mass as the objective functions, an enhanced EDA (EEDA) is proposed to find the unknown weighting coefficients. The proposed optimizer benefits from a new learning mechanism based on mixture of Gaussian distribution. The mechanism prevents diversity loss of the population during the optimization process of EDA .The approach is tested in some time-optimal and fuel-optimal cases and the optimality of the obtained solutions are analyzed.

The outline of the chapter is as follows. In Section 2, the two-body dynamical model is given and the proposed simple Lyapunov Controller is presented alongside the Petropoulos Q-law. Section 3 is dedicated to the optimization process in finding the optimal weighting coefficients for minimum-time and minimum-fuel transfer trajectories. Numerical simulations are provided in Section 4, where several cases of the orbit transfer missions are considered as the benchmark problems. The problems are solved using the proposed approach with variety of settings as algorithm parameters. Finally, Section 5 contains the conclusions remarks and future works.

2 Low-thrust Trajectory Design

2.1 Two-body Dynamics

The spacecraft's motion about a central body is described in terms of the classical orbit element (COEs) semi-major axis a, eccentricity e, inclination i, right ascension of the ascending node (RAAN) $\varOmega$ and argument of periapsis ω. If the perturbing acceleration $\boldsymbol{a}_d$ is described in the radial, transverse and normal (RTN) frame, then the set of variational equations in $a, e, i, \varOmega, \omega$ and the true anomaly ν take Gauss's form of the Lagrange Planetary Equations [22]. These can be expressed as:

$$\begin{bmatrix} \dot{\boldsymbol{X}} \\ \dot{\nu} \end{bmatrix} = \begin{bmatrix} & \boldsymbol{B} & \\ \frac{p\cos(\nu)}{he} & \frac{-(p+r)\sin(\nu)}{he} & 0 \end{bmatrix} \boldsymbol{a}_d + \begin{bmatrix} \boldsymbol{0} \\ \frac{h}{r^2} \end{bmatrix}, \tag{1}$$

where $\dot{\boldsymbol{X}}$ represents the dynamics of the COE state, $p = a(1 - e^2)$ is the semi-latus rectum, μ the gravitational parameter and $h = \sqrt{\mu p}$. The matrix $\boldsymbol{B}$ represents the Gauss Variational Equations (GVEs) for the slow variables and is required later to compute to compute the control.

$$\boldsymbol{B} = \begin{bmatrix} \frac{2a^2}{h} e\sin(\nu) & \frac{2a^2}{h}\frac{p}{r} & 0 \\ \frac{1}{h} p\sin(\nu) & \frac{1}{h}((p+r)\cos(\nu) + re) & 0 \\ 0 & 0 & \frac{r\cos(\omega+\nu)}{h} \\ 0 & 0 & \frac{r\sin(\omega+\nu)}{h\sin(i)} \\ -\frac{1}{he} p\cos(\nu) & \frac{1}{he}(p+r)\sin(\nu) & -\frac{r\sin(\omega+\nu) cos(i)}{h\sin(i)} \end{bmatrix} \tag{2}$$

When examining Eq. (2), it is clear that singularities occur when $i = 0$ or $e = 0$. The modified equinoctial elements (MEEs) p, f, g, h, k and fast variable L are used in the dynamical integrator instead of COEs to avoid these issues [23]. However, the control remains in COEs in order to preserve the physical interpretation of the variables and extract insights from the observed behaviour.

2.2 Lyapunov Control

Lyapunov functions offer a method for computing continuous-thrust trajectories with minimal computational cost. Obtaining an estimate of the low-thrust profile required for the acquisition of the different target orbits is straightforward thanks to the computation ease and closed-loop nature. The most well known and widely used is perhaps the Petropoulos Q-law [14, 15]. We provide an outline of this control law here, although the interested reader should refer [15] for full details. We also propose a very simple Lyapunov feedback control law, formulated in Classical Orbital Elements (COEs). Although the Q-law is a more optimal Lyapunov feedback control laws exist, the simplicity gives the subsequent enhanced EDA more freedom and avoids potential singularity issues. As such it is interesting to compare the performance of these two controllers.

2.2.1 Simple Lyapunov Controller

The proposed simple Lyapunov control function can be written as

$$V = \sum_X W_X \delta(X, X_T)^2. \tag{3}$$

where $\delta(X, X_T) = X - X_T$ for $X = a, e, i$ whilst $\delta(X, X_T) = \arccos(\cos(X - X_T))$ for $X = \mathit{\Omega}, \omega$. The weights W_X can be used to prioritise which elements to target. As written above, the control law is a proportional function and as such $\boldsymbol{W}$ are normalised.

2.2.2 Proximity Quotient Controller (Q-law)

The Q-law is best thought of as a weighted, squared summation of the time required to change the current state $\boldsymbol{X} = [a, e, i, \mathit{\Omega}, \omega]^{\mathrm{T}}$ to the target state $\boldsymbol{X}_T = [a_T, e_T, i_T, \mathit{\Omega}_T, \omega_T]^{\mathrm{T}}$. It can be written as

$$V = Q = (1 + W_P P) \sum_X S_X W_X \left(\frac{\delta(X, X_T)}{\max_\nu(\dot{X})} \right)^2, \tag{4}$$

where Q and V can be used interchangable in the following formulas. Here W_P and P form a penalty function and S_X are scaling functions. These are functions of the state and can be found in Ref 15. Again, $\delta(X, X_T) = X - X_T$ for $X = a, e, i$ whilst $\delta(X, X_T) = \arccos(\cos(X - X_T))$ for $X = \mathit{\Omega}, \omega$. The expressions $\max_\nu(\dot{X})$ are the maximum rate of change of each COE over the current osculating orbit and can be calculated analytically for all elements except ω. For a more detailed breakdown of the components of the Q-law, the reader is encouraged to look at Refs. 15, 24 and 25. All user-defined parameters aside from the weights W_X (for instance, those in W_P, P and S_X) are assumed constant, as these terms are only activated to prevent particular behaviour that we also want to prevent. The values are given in Table 1.

Table 1: Table of standard Q-law parameters from Ref. 15

Parameter	Value	Parameter	Value	Parameter	Value
r	2	k	100	b	0.01
$r^{\text{p-min}}$	6,578 km	m	3	n	4

2.2.3 Control Direction

Lyapunov's second theorem states that for a system $\dot{\boldsymbol{Z}} = \boldsymbol{f}(\boldsymbol{Z})$, $\boldsymbol{Z} = \boldsymbol{X} - \boldsymbol{X}_{\mathrm{T}}$, the equilibrium point $\boldsymbol{X}_{\mathrm{T}}$ is asymptotically stable if there exists a scalar Lyapunov function $V(\boldsymbol{Z})$ such that $V(\boldsymbol{0}) = 0$; it is positive-definite ($V(\boldsymbol{Z}) > 0, \forall Z \neq \boldsymbol{0}$); the derivative is negative-definite ($\dot{V}(\boldsymbol{Z}) < 0, \forall Z \neq \boldsymbol{0}$); and $\lim_{|\boldsymbol{Z}| \to \infty}, V(\boldsymbol{Z}) = \infty$ [26]. A very thorough discussion on the implications of this for trajectory design using nonlinear control can be found in Ref. 27.

A stable control is therefore one that ensures $\dot{V} < 0$ throughout the transfer. One way of doing this is to select a controller that minimises the rate of change of the Lyapunov function (in this case the most negative value).

$$\dot{V} = \frac{\partial V}{\partial \boldsymbol{X}} \dot{\boldsymbol{X}} = \frac{\partial V}{\partial \boldsymbol{X}} \boldsymbol{B} \boldsymbol{u} \tag{5}$$

Given an engine thrust T and a spacecraft mass m, the control vector is computed as (using $f = T/m$):

$$\boldsymbol{u} = -f \frac{\boldsymbol{B}^{\mathrm{T}} \left(\frac{\partial V}{\partial \boldsymbol{X}}\right)^{\mathrm{T}}}{\left|\left|\left(\frac{\partial V}{\partial \boldsymbol{X}}\right) \boldsymbol{B}\right|\right|} \tag{6}$$

Coasting can be introduced using effectivity thresholds η^{thresh}. These attempt to quantify the effectivity of changing an orbital parameter at a given point in an orbit compared to the optimum point for changing the same orbital parameter. Definitions for both the absolute and relative effectivity parameters exist. Studies [18] have also shown that, when varying other control law parameters in addition to the effectivity, there is little difference in performance between relative and absolute effectivity, and instead the specific transfer will determine which is more applicable. Hence, in this work the absolute effectivity:

$$\eta_a = \frac{\min_{\alpha,\beta} \dot{V}}{\min_{\nu}(\min_{\alpha,\beta} \dot{V})} \tag{7}$$

is used. α and β are the in-plane and out-of-plane angles of the thrust vector, whilst $\min_{\nu}(\min_{\alpha,\beta} \dot{V})$ is computed numerically by scanning through the possible true anomaly ν values to find the maximum and minimum $\dot{V}$ for the particular osculating orbit. The authors note that an alternative approach first found in Ref. 19 can be used which now avoids any numerical derivations thanks to recent work in Ref. 25.

3 Optimization Approach

Following the proposed approaches including the Q-law and Lyapunov methods, achieving optimal transfer trajectory lies upon finding a suitable set of values for unknown weighting coefficients of the proposed techniques. In this research, the unknown weighting coefficients are defined as:

$$\vec{x} = [\eta_a \quad W_a \quad W_e \quad W_i \quad W_{\Omega} \quad W_{\omega}] \tag{8}$$

which include the normalized state weights $W_X = [W_a \quad W_e \quad W_i \quad W_{\Omega} \quad W_{\omega}]$ and the absolute effectivity threshold η_a. Tuning these parameters for a given orbit design problem yields time-optimal or mass-optimal transfer trajectories. Basically, constant values are the primary choice for these parameters. However in this research, these parameters are considered as functions of transfer time. Therefore, the problem turns into finding proper time-series for the weighting coefficient to achieve optimal transfer trajectories.

3.1 Direct Interpolation

Parameterizing the unknown weighting coefficients has a significant impact on the convergence of the optimization algorithm, and consequently, obtaining optimal transfer trajectories. In this research, the time series of decision variables $\vec{x}$ is approximated by considering N_p number of interpolation points in the desired time of flight interval $0 < t < t_f$. Noting the fact that all weighting coefficients are normalized, the weights' variation are in the limits of $0 < \vec{x} < 1$. Having N_p number of uniformly discretized points, the overall time span is divided into $N_p - 1$ sub-intervals. The interpolating polynomial for the time interval can be represented by:

$$\hat{x}(t) = \sum_{k=1}^{N} (\prod_{j \neq k} \frac{t - t_j}{t_k - t_j}) p_k \tag{9}$$

where $\hat{x}(t)$ denotes any of the unknown weighting coefficients in $\vec{x}$, t_k are the discretized times, and p_k are the discrete points within the time interval. Given the number of discrete points N_p for each decision variable, the time series of the corresponding weighting coefficient may be interpolated with different shapes. One of the most popular methods is using piecewise cubic Hermite interpolating polynomials [28]. Various types of splines can be obtained depending the choice of tangents in each node. One type of spline from the family of Hermite splines, which is frequently used in many applications, is illustrated in Fig. 1 for approximating the weighting coefficients.

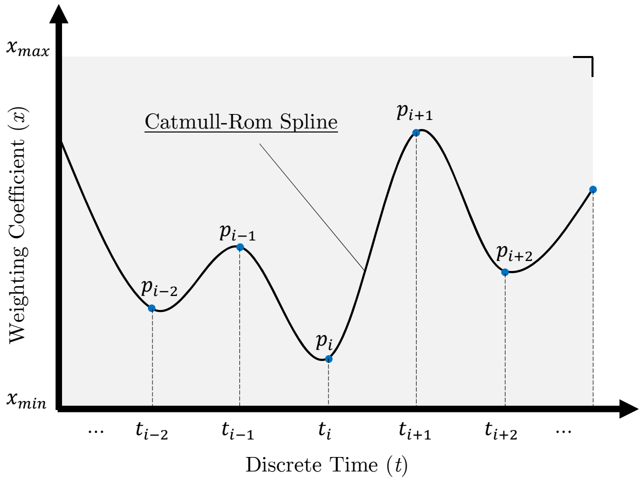


Fig. 1: Weghting coefficient interpolating with piecewise cubic Hermite splines.

This interpolation represents Catmull Rom spline [29], which has a continuous first derivative. The difference between this polynomial and other types of Hermite interpolation schemes is the choice of tangents in the internal and end nodes. This spline has a balanced smoothness and the slope of the spline at data points depends directly on the points before and after. The resulting piecewise cubic does not have a continuous second derivative and it does not always preserve shape. However, it can be evaluated quickly by a convolution operation. More details regarding the derivation of this spline and its difference from the other types of interpolations is beyond the scope of this research and the reader is urged to refer to the provided references for details [29–31].

Regarding this explanation, the low-thrust trajectory design can be defined as an unconstrained optimization problem. The general form of such an optimization problem is:

$$\begin{aligned} &Minimize \quad F(x) \quad x = (x_1, x_2, ..., x_n) \\ &Subject\, to \quad x_{min} < x_i < x_{max} \end{aligned} \tag{10}$$

where n is the number of decision variables, and $F(x)$ represents the objective function, which is the actual transfer time for time-optimal transfers, and fuel mass of the spacecraft for mass-optimal transfers. Based on the proposed formulation of the low-thrust space orbit transfer problem, the decision variables $x = (x_1, x_2, ..., x_n)$ are the interpolation points for unknown weighting coefficients of the proposed controller:

$$x = p_i \quad (i = 1, ..., N_v \times N_p) \tag{11}$$

where N_v is the number of unknown weighting coefficients (i.e. $N_v = size(\vec{x})$). As can be seen, equal number of interpolation points are dedicated to every unknown variable in this research. However, in a more general concept, one may consider different interpolation points for each of the weighting coefficients. To deal with this unconstrained continuous optimization problem, an EDA-based algorithm is proposed in the following section.

3.2 Enhanced Estimation of Distribution Algorithm

EDAs are a type of population-based optimization algorithms, designed for solving numerical optimization problems. Based on machine learning techniques, at each iteration, EDAs learn a probabilistic model from a subset of the most promising solutions, trying to explicitly express the interrelations between the different variables of the problem. Then, by sampling the probabilistic model learned in the previous iteration, a new population of solutions is created. In other words, EDAs work based on two major key methods: learning and sampling, where a probabilistic model that estimates the probability distribution of the selected solutions is learned and then utilized for sampling new individuals [32]. In this work, an improved learning mechanism is presented and applied to an EDA based on multivariate Gaussian distribution. It will be shown that the new enhanced algorithm outperforms traditional EAs within the proposed optimization problem.

The overall pseudo code of the proposed algorithm is presented in Algorithm 1. Following the pseudo code, the optimization process begins with N as the population size and M as the maximum number of iterations. Initially, the **SEEDING** mechanism is utilized to generate an initial population. Having the initial feasible solutions, with corresponding objective values f obtained from **EVALUATION**, the main optimization loop starts. At each iteration, the algorithm begins by selecting the top promising individuals in the current population according to the **SELECTION** method. Truncation selection method [32] is used in this research, with γ as the truncation factor. In this method, the γ fraction $\gamma \in (0, 1]$, of the best individuals are selected. All the individuals have the same selection probability defined as:

Algorithm 1: Overall pseudo code of EDA

```
Input: $F(x), x_{min}, x_{max}$
Parameters: $N, M, \gamma, \alpha, \lambda, N_c$
$x \leftarrow$ SEEDING($x_{min}$,$x_{max}$,$N$)
$f \leftarrow$ EVALUATION($x$,$F(x)$)
if $i < M$ then
    for $iter \leftarrow i$ to $M$ do
        $[x_{sel}, f_{sel}] \leftarrow$ SELECTION($x$,$f$,$\gamma$)
        $[\Phi, \phi] \leftarrow$ LEARNING($x_{sel}$,$f_{sel}$, $\alpha$,$\lambda$,$N_c$)
        $x_{sam} \leftarrow$ SAMPLING($\Phi$,$\phi$,$N$)
        $x_{rep} \leftarrow$ REPAIRING($x_{sam}$,$x_{min}$,$x_{max}$)
        $f_{rep} \leftarrow$ EVALUATION($x_{rep}$,$F(x)$)
        $[x, f] \leftarrow$ REPLACEMENT($x_{rep}$,$f_{rep}$,$x$,$f$)
        EXTRACT $[x_{best}, f_{best}]$ FROM $[x, f]$;
        if stopping criteria are met then
            BREAK;
        end if
    end for
else
    EXTRACT $[x_{best}, f_{best}]$ FROM $[x, f]$;
end if
Output: $x_{best}, f_{best}$
```

$$P_j = \begin{cases} \dfrac{1}{N_s} & 1 < j < N_s \\ 0 & N_n < j < N \end{cases} \tag{12}$$

where N_s is the number of selected individuals as $N_s = \gamma N$. Having the selected population x_{sel} and the corresponding objective values f_{sel}, a probability model is learned via the **LEARNING** mechanism. In the proposed learning mechanism, the selected population is divided into two types of clusters. These clusters include *parent clusters*, denoted by Φ, and *smart clusters* denoted by ϕ. The pseudo code of the learning mechanism is shown in Algorithm 2.

The main idea of the learning process is based on utilizing a mixture of Gaussian distributions as a probabilistic model whose density function is formalized as:

$$P(x) = \sum_{k=1}^{N_c} \pi_k f_k(x|\mu_k, \Sigma_k) \tag{13}$$

where each $f_k(x|\mu_k, \Sigma_k)$ component of the mixture is a multivariate Gaussian distribution, and μ_k and Σ_k are the mean value (the centroid) and the covariance matrix of the k model for $k = 1, \ldots, N_c$ respectively, with π_k as the mixing coefficient for the kth component.

In the proposed learning stage, the Gaussian mixture model is constructed in two steps. The first step consists of clustering the selected population according to N_c number of parent clusters. In this research, *k–means++* is chosen as the clustering method [33]. However, other methods could also be considered. Following the clustering process, the mixture of Gaussian distributions model

Algorithm 2: Pseudo code of the learning mechanism

```
Input: $x_{sel}$, $f_{sel}$, $\alpha$,$\lambda$, $N_c$
$[\iota, \mu] \leftarrow$ kmeans($x_{sel}$,$i$);
CONSTRUCT $\mathit{\Phi}$ FROM $[\mu, x_{sel}(\iota)]$
for $i \leftarrow 1$ to $N_c$ do
    EXTRACT $[\hat{x}, \hat{f}, \hat{\mu}, \hat{\sigma}]$ FROM $\mathit{\Phi}(i)$
    $[\hat{x}_{sel}, \hat{f}_{sel}] \leftarrow$ SELECTION($\hat{x}$,$\hat{f}$,$\alpha$)
    $d \leftarrow ||\hat{x}_{sel} - \hat{\mu}||$
    $j \leftarrow 0$
    if $d > \lambda \times \hat{\sigma}$ then
        $j \leftarrow j + 1$
        CONSTRUCT $\hat{\phi}$ FROM $[\hat{\mu}, \hat{x}_{sel}]$
        $\phi(j) \leftarrow \hat{\phi}$
end for
Output: $\mathit{\Phi}, \phi$
```

is learned by calculating the maximum likelihood estimators of the parameters of the components in this mixture, using the solutions in the respective clusters. This process is the first step of the learning process. Finalizing the process, the components $\mathit{\Phi}$, referred to as the parent clusters, are extracted, which contain corresponding solutions $\hat{x}$, objective values $\hat{f}$, centroids $\hat{\mu}$ and covariances $\hat{\sigma}$. In the next step of the learning process, more components are added to the model. This step is to compensate the covariance loss during the optimization process after the sampling stage. In this step, for each component $\mathit{\Phi}_i$, first, the top α percentage of the best solutions ($\hat{x}_{sel}$ and $\hat{f}_{sel}$) are selected. Then, the selected set of solutions is analyzed to see if they have outliers using the Z-score outlier detection method [34]. This method is represented as:

$$\frac{||\hat{x}_{sel} - \hat{\mu}||}{\hat{\sigma}} > \lambda \tag{14}$$

where λ is the distance threshold from the centroids $\hat{\mu}$. According to this mechanism, if an outlier solution is at the top α percentage of the best solutions, it will be considered as the centroid of a new component in the mixture $\hat{\phi}$, referred to as an *outlier-based cluster*. For the newly formed components, we assume an independent multivariate Gaussian distribution, where the variance of each dimension is calculated as half of the distance from the initial centroid in each component.

By the end of the learning mechanism, a mixture of models is learned, one component on top of each cluster, in such a way that the probability of sampling top quality solutions becomes high. Having the mixture of models, new solutions are sampled via the **SAMPLING** method as x_{sam}. Then, the **REPAIRING** method simply refines the newly sampled solutions based on the boundaries of the solution domain x_{min} and x_{max}. Following the repairing process, new individuals will be obtained as x_{rep}. After evaluating the objective value of the obtained solutions, f_{rep}, via the **EVALUATION** process, the new individuals are combined with the individuals from the previous population, and the **REPLACEMENT** mechanism is invoked to form the new population and the corresponding objective values f in the current iteration. Population aggregation method is used for this mechanism in this research. The process mentioned continues until at least one stopping criteria is met.

4 Numerical Simulations

The presented algorithm is incorporated with two aforementioned low-thrust trajectory optimization schemes including Lyapunov and Q-law feedback control methods. In this section the presented optimization approach is tested in two experiments, with respect to the orbit transfer missions from the literature. Both time-optimal and mass-optimal trajectories are considered in each experiment and the results are analyzed accordingly.

In the first experiment a low-thrust orbital maneuver from a Geostationary Transfer Orbit to Geostationary Orbit (GTO to GEO) is considered [21, 35]. Owing to the almost circular nature of the target orbit in this case, Ω and ω are free variables, and the algorithm needs to target only a, e and i. In this experiment, EEDA is incorporated in both Lyapunov and Q-law control methods in order to compare their performances. Results indicate that the proposed approach outperforms the other method based on averaging technique and another Q-law based method from the literature. Furthermore, we observe a similar performance between the EEDA+Lyapunov and EEDA+Q-law approaches in some circumstances. This shows the potential the EEDA to extract nuances in the behaviour that allows the simple Lyapunov controller to match the much more advance Q-law controller performance.

The second experiment is a transfer from a Geostationary Transfer Orbit to a retrograde, Molniya-type orbit (GTO to Molniya) [18,20,36]. In this case, the weighting coefficients correspond to all slowly varying orbital elements are considered to be optimized. The Lyapunov feedback control method incorporated with the proposed EEDA is implemented in this experiment. It will be shown that the obtained solutions have higher quality in terms of fuel consumption and transfer time in comparison to the results from other references based on nominal Q-law methods. Also, the quality of the obtained solutions and the convergence of the algorithm is compared with other EAs. However, when compared to other Genetic Algorithm Q-law solutions, there is a gap in performance that can arise due to the use of a simple Lyapunov controller instead of the Q-law.

In the case of time-optimal transfers, the cost function provided to the optimiser is simply the final time-of-flight. A penalty term is added for not converging to the target orbit, along with a residual on the orbit elements. The mass-optimal transfers are more complex. It is unclear if truly mass-optimal low-thrust transfers will converge to the target orbit within a practical time-frame and thus it is necessarily to impose a time restriction. This can be done by fixing the arrival time and incorporating this as a constraint in the optimisation process. However, relaxing this to a maximum time constraint is easier to solve whilst ensuring the same mass-optimal solution as long as the dynamics remain Keplerian. Hence, the cost function includes the propellant mass used and a penalty term when the time-of-flight exceeds the desired maximum arrival time. Again a penalty term on the residual is provided to encourage trajectories to converge.

In order to realize the best choice for the interpolation spline, several number of interpolation points have been considered. It is possible to consider very high number of interpolation points. However, increasing the number of interpolation points results in having more number of decision variables to be optimized and an increase in the optimization problem complexity. Having this insight, the number of interpolation points are considered as $N_p = 1$ to $N_p = 5$. The optimization has been run 10 times for each case. Therefore, total number of 50 optimization runs have been processed, including 10 runs for each $Np = 1, ..., 5$. Some of the EEDA parameters are randomly selected for each individual run, while some other settings have been set to constant values. The algorithm settings is provided in Table 2.

Table 2: EEDA parameter settings

Parameter	Value
Number of parent clusters	$3 \leqslant Nc \leqslant 8$
Truncation factor	$0 \leqslant \gamma \leqslant 1$
Outlier detection parameter	$0.01 \leqslant \alpha \leqslant 0.1$
Distance threshold	$\lambda = 1$
Number of iterations	$N = 100$
Population size	$M = 100$

According to the provided settings, the total number of decision variables $\vec{x}$ varies with the number of interpolation points and the number of weights that are considered for the problem. In GTO to GEO mission, we have $4 \times N_p$ decision variables to consider in the optimization, since two orbital elements are considered free, along with the absolute effectivity threshold. However, in GTO to Molniya mission, the number of decision variables is $6 \times N_p$ since all orbital elements are considered in optimization. Also, the boundaries of the decision variables are $x_{min} = 0$ and $x_{max} = 1$, since the weights are normalized as mentioned previously.

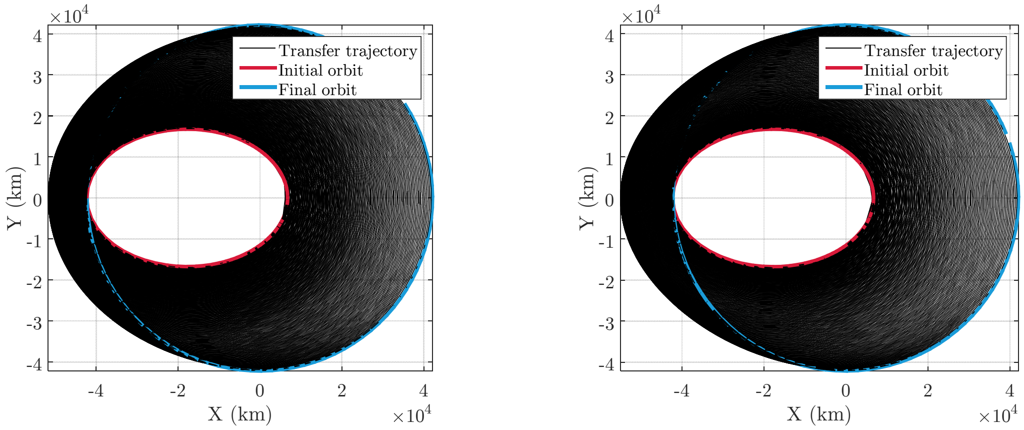
4.1 GTO to GEO

First, a GTO to GEO transfer with an inclination change is considered in Keplerian dynamics, the details for which are provided in Table 3.

Table 3: Initial and final orbital parameters for GTO to GEO transfer

	a (km)	e	i (deg)	Ω (deg)	ω (deg)	ν (deg)
Initial orbit	24505.9	0.725	7	0	0	0
Final orbit	42165	0	0	$free$	$free$	$free$
Convergence	36.0	8.5e-4	0.1	-	-	-

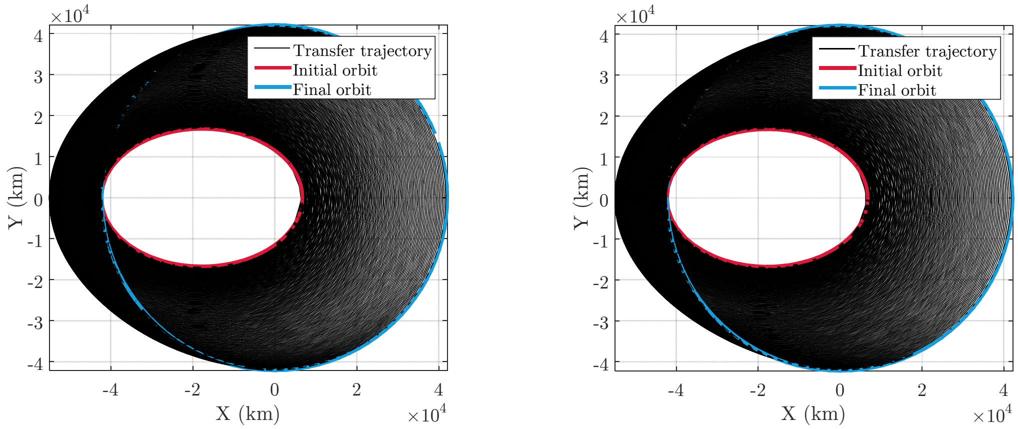
The parameters are chosen to compare with [21] and [35]. The modeled spacecraft has the mass of $2000kg$, the thrust level of $0.35N$ and the specific impulse (Isp) of $2000s$, giving an initial thrust-to-mass ratio of 0.000175 ms^{-2}. In the mass-optimal case, the upper bound on the time-of-flight is 150 days. Following the insights from the previous experiment, the incorporation of EEDA with Lyapunov method is competitive to Q-law methods. Therefore, in this experiment, the proposed algorithm is implemented in both Lyapunov and Q-law control methods in this space mission and time-optimal and fuel-optimal transfers are solved. The best obtained transfer trajectories for Q-law method are shown in Fig. 2a and Fig. 2b for fuel-optimal and time-optimal transfers respectively. Also, Fig. 3a and Fig. 3b show the best obtained solutions for Lyapunov method, similarly for fuel-optimal and time-optimal transfers respectively.



(a) Fuel-optimal transfer.

(b) Time-optimal transfer.

Fig. 2: 3D visualization of trajectories for GTO to GEO transfer based on Q-law method.



(a) Fuel-optimal transfer.

(b) Time-optimal transfer.

Fig. 3: 3D visualization of trajectories for GTO to GEO transfer based on simple Lyapunov method.

As the figures indicate, the transfer trajectories have small difference, mainly due to the very low amount of thrust level. In order to realize the differences within the transfer trajectories, the

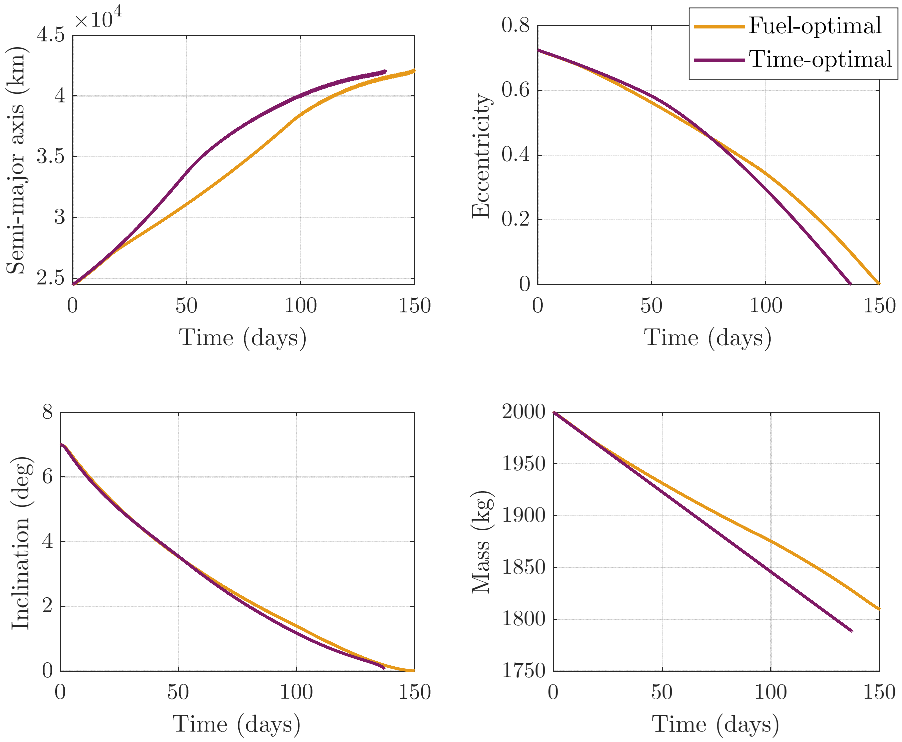


Fig. 4: State variables for fuel-optimal and time-optimal transfers based on Q-law method in GTO to GEO transfer

time variation of state variables are depicted in Fig. 4 and Fig. 5 for Q-law method and simple Lyapunov method respectively.

The exact values of the transfer time and the actual fuel mass consumption for this problem are provided in Table 4.

Table 4: Comparison of the obtained solutions for GTO to GEO transfer

	Min-time		Min-fuel	
	Time (day)	Fuel mass (kg)	Time (day)	Fuel mass (kg)
Q-law+EEDA method	137.34	211.76	149.99	190.40
Lyapunov+EEDA method	137.5	212.01	149.70	192.34
Averaging method [35]	137.5	212	150	192
Q-law method [21]	137.3	211.72	150	187.97

Several observations can be made from Table 4. First, the implementation of the proposed algorithm with Lyapunov method ends up in the solutions with almost the same quality of the averaging method in [35]. However, the implementation of the proposed algorithm with Q-law

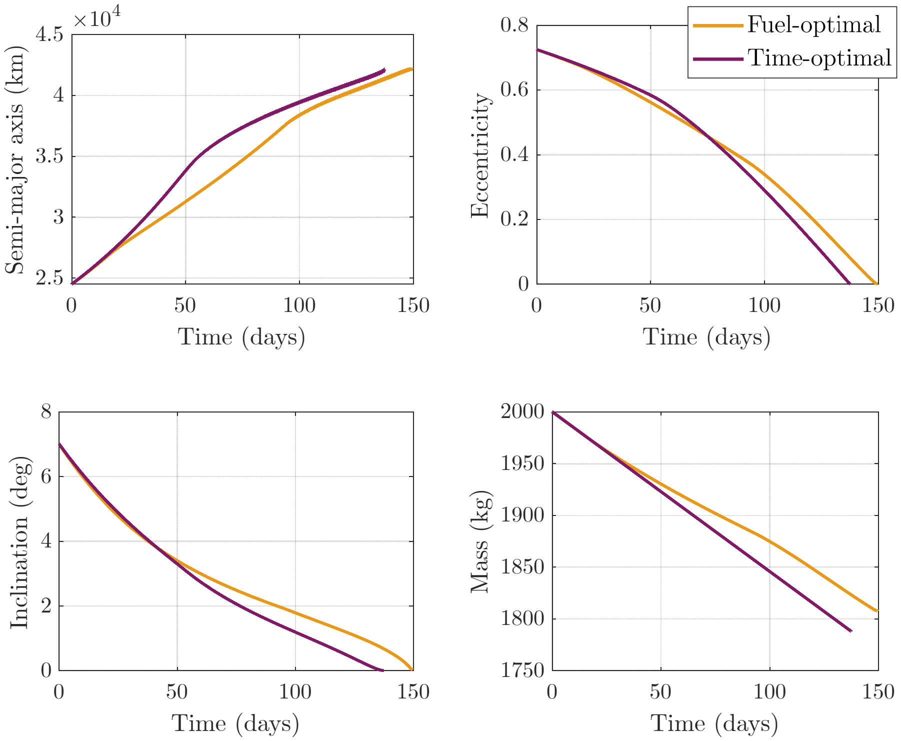


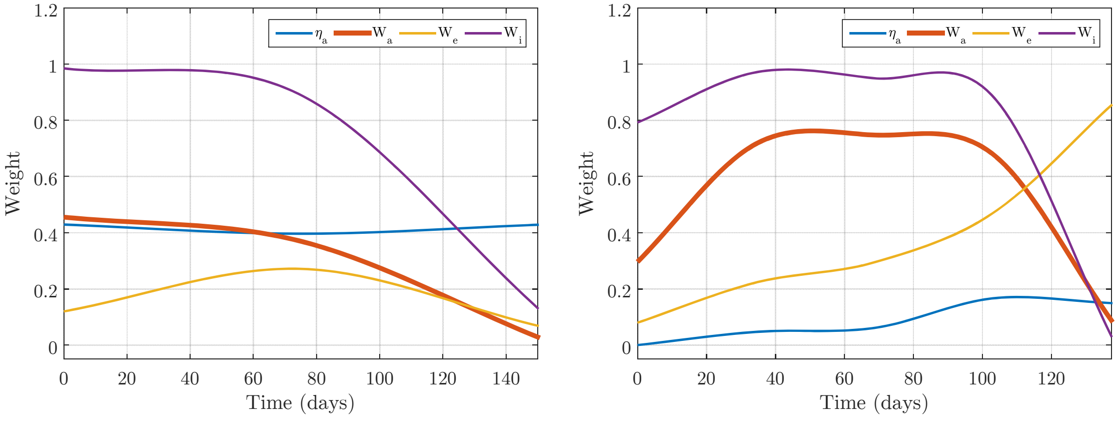
Fig. 5: State variables for fuel-optimal and time-optimal transfers based on simple Lyapunov method in GTO to GEO transfer

method results in a slightly better solutions in both time-optimal and fuel-optimal transfers. Comparing the obtained results from the Q-law implementation with the results from [21] shows that the proposed approach is able to find a solution close to the one presented in the literature in the time-optimal problem. However, in fuel-optimal problem the amount of fuel-mass is a bit higher than the solution from [21]. Overall, the proposed approach is shown to be competitive to the recently developed methods for obtaining low-thrust transfer trajectory design and optimization.

The corresponding variation of the optimized weights for Q-law method are depicted in Fig. 6a and Fig. 6b for fuel-optimal and time-optimal transfers respectively.

As shown in the figures, the variations of weights for semi-major axis, eccentricity, and inclination start with the values which are close in fuel-optimal and time-optimal transfers. The significant difference between the two is the mean value of the absolute threshold, which is higher in fuel-optimal transfer in comparison to time-optimal transfer. Similarly, the optimized variation of weights for Lyapunov method are illustrated in Fig. 7a and Fig. 7b

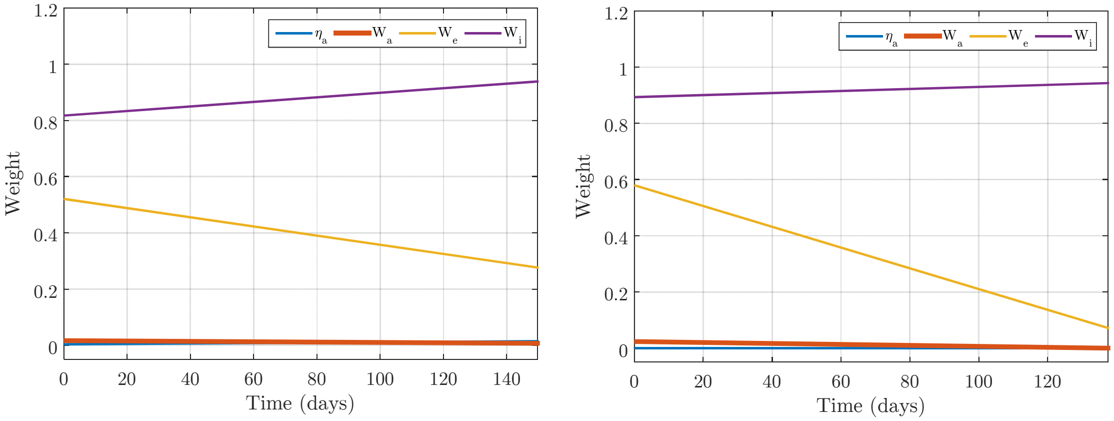
Analyzing the weighting coefficients in Lyapunov method reveals a noticeable difference. According to the time variation of the weighting coefficients for Lyapunov method, the low-point interpolation seems to be more beneficial for the optimizer to find high quality solutions in both fuel-optimal and time-optimal transfers. In this regard, the best obtained solution for time-optimal and fuel-optimal transfers for Lyapunov method are associated with $N_p = 2$, leading to conclude



(a) Fuel-optimal transfer.

(b) Time-optimal transfer.

Fig. 6: Weighting coefficient profiles for GTO to GEO transfer based on Q-law method.



(a) Fuel-optimal transfer.

(b) Time-optimal transfer.

Fig. 7: Weighting coefficient profiles for GTO to GEO transfer based on simple Lyapunov method.

that the Lyapunov method requires less variation of weights for achieving optimal solutions, while Q-law method is more sensitive to time-variation of weights.

4.2 GTO to Molniya

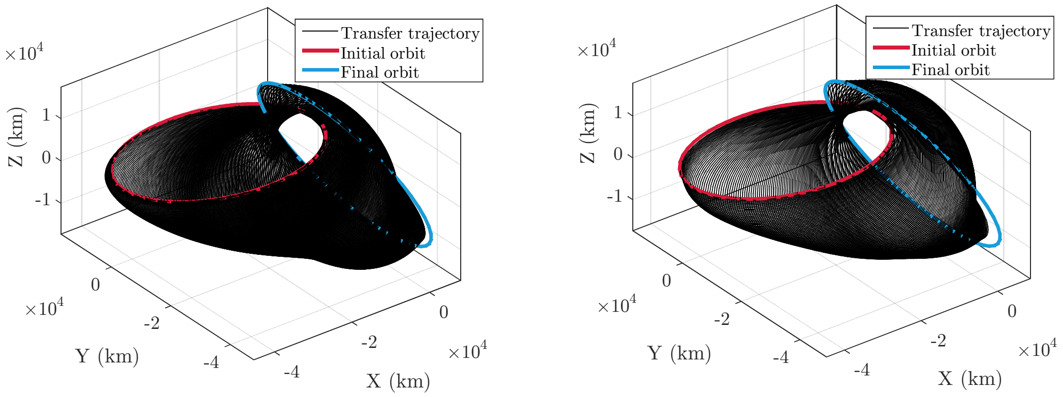
In the second experiment a GTO to Molniya transfer is considered and the orbital elements of the initial and final orbits are provided in 5.

Table 5: Initial and final orbital parameters for GTO to Molniya transfer

	a (km)	e	i (deg)	Ω (deg)	ω (deg)	ν (deg)
Initial orbit	24505.9	0.725	0.06	0	0	0
Final orbit	26500	0.7	116	270	180	$free$
Convergence	10.0	0.005	1.0	1.0	1.0	-

As is evident from the table, this case involves large changes. The required plane change is about 116°. The modeled spacecraft has the mass of = 2000 kg, the thrust level of $T = 0.35N$ and the specific impulse of $I_{sp} = 2000s$, giving an initial thrust-to-mass ratio of 0.000175 ms^{-2}. In the mass-optimal case, the upper bound on the time-of-flight is 150 days.

For this case, the proposed algorithm is implemented in Lyapunov control method and time-optimal and fuel-optimal transfers are solved. Fig. 8a and Fig. 8b show the initial and final orbits along with the transfer trajectory, corresponding to the best obtained solution in min-fuel and min-time problems respectively.



(a) Fuel-optimal transfer.

(b) Time-optimal transfer.

Fig. 8: 3D visualization of trajectories for GTO to Molniya transfer based on simple Lyapunov method.

As can be seen, the time-optimal transfer trajectory significantly has more revolutions to reach the final orbit relative to fuel-optimal transfer. The time variation of state variables for this example are shown in Fig. 9.

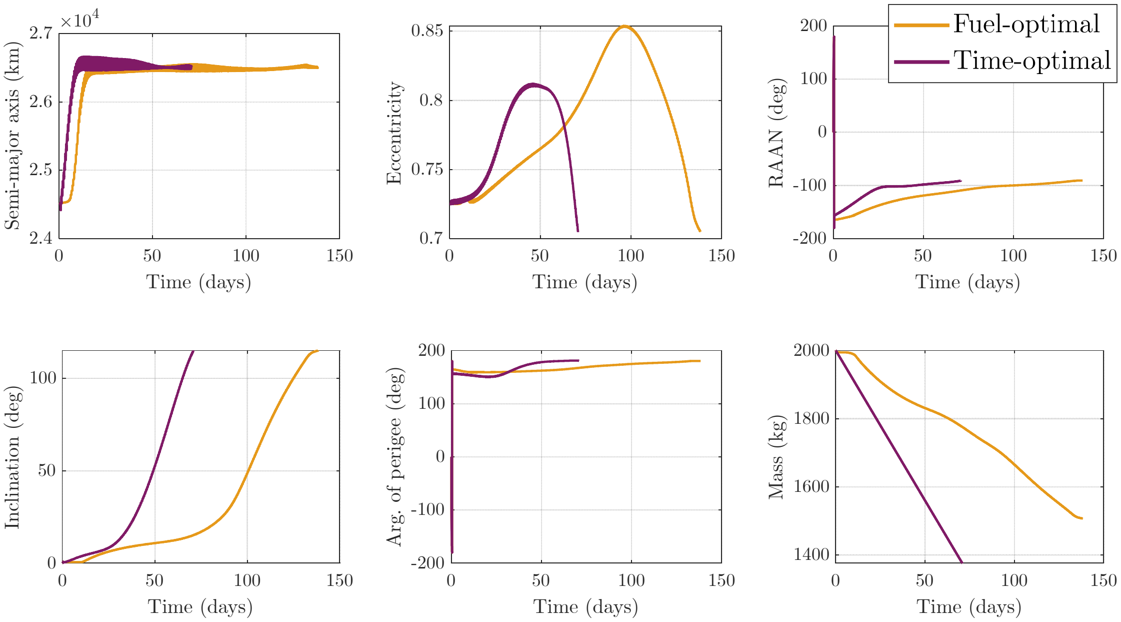


Fig. 9: State variables for fuel-optimal and time-optimal transfers based on simple Lyapunov method in GTO to Molniya transfer

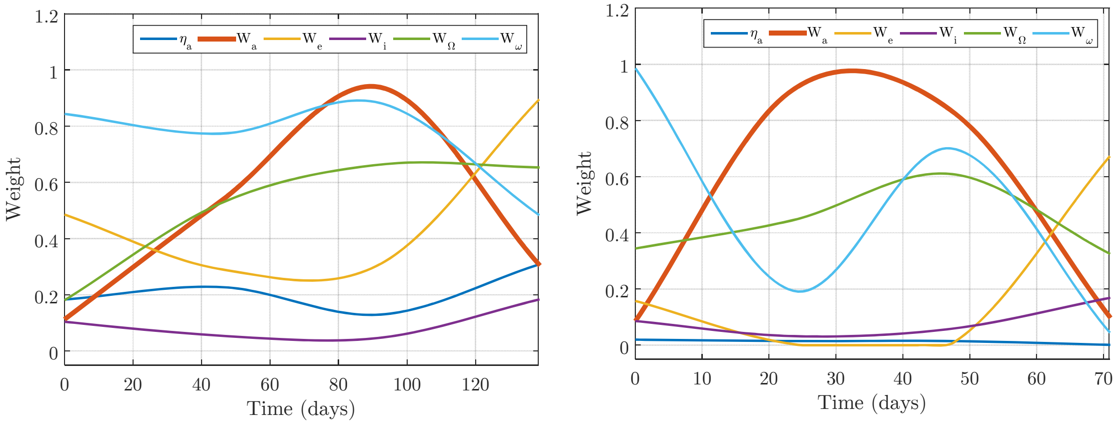
Details of the transfer time and the actual fuel mass consumption for this problem are provided in Table 6.

Table 6: Comparison of the obtained solutions for GTO to Molniya transfer

	Min-time		Min-fuel	
	Time (day)	Fuel mass (kg)	Time (day)	Fuel mass (kg)
Lyapunov+EEDA method	70.83	624.01	138.34	492.89
Lyapunov method [36]	96.6	677.2	N. A.	N. A.
Q-law method [20]	84	734	150	580

Table 6 also includes the corresponding transfer time and fuel mass of the obtained solutions from other methods in the literature. As can be seen, the obtained solutions by the proposed method in this research significantly have higher quality than the Lyapunov method in [36] and even the Q-law method in [20]. This comparison shows that the incorporation of the proposed optimizer within the Lyapunov control method is competitive to the Q-law method, which is a more complicated control technique. The variation of the weighting coefficients for the obtained transfers are illustrated in Fig. 10a and Fig. 10b.

Following the variation of the optimized weighting coefficients, some remarks can be highlighted. It can be observed that there are some small similarities between time-optimal and fuel-optimal



(a) Fuel-optimal transfer.

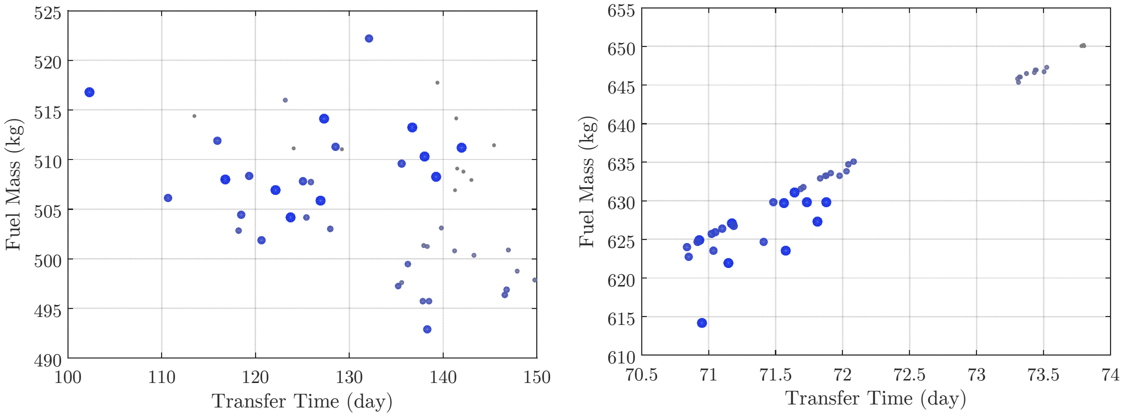
(b) Time-optimal transfer.

Fig. 10: Weighting coefficient profiles for GTO to Molniya transfer based on simple Lyapunov method.

transfers. For instance, the absolute threshold η_a begins with a high value in both transfers. However, it slightly changes based on the desired objective. The semi-major axis weight has similar variation, and the weights of inclination is low in both problems.

It is noteworthy that the best obtained solution in both time-optimal and fuel-optimal cases correspond to $N_p = 4$ in this scenario. Other solutions have been obtained as well out of 50 optimization runs and they have more or less similar objective values. However, the absolute best solutions (min-time and min-fuel) indicate that 4-points interpolation can interpret the optimal variation of weights better than other number of interpolation points. In other words, a simple increase in the number of interpolation points does not necessarily lead us to high quality solutions. The spread of the obtained solutions, depicted in Fig. 11a and Fig. 11b confirms this fact.

Fig. 11a and Fig. 11b show the relative position of the obtained solution in each optimization run. In these figures, solutions correspond to higher number of interpolation points are plotted with bigger markers. As can be observed, it can be highlighted that in fuel-optimal transfer (Fig. 11a), the optimizer did not find any high quality solutions with 5-points interpolation, and the majority of the points with lower fuel mass correspond to 3-points and 4-points interpolation. The existence of some of the points near the transfer time of 150 days (the maximum allowable simulation time), with fuel mass less than 500 kg indicates that this region has the potential for finding better solutions. Interesting remarks can be highlighted from the distribution of the obtained solutions in time-optimal transfer (Fig. 11b) as well. Unlike the previous case, the distribution of the 5-points interpolation shows that increasing the number of interpolation points leads the optimizer to find solutions with better quality. As can be seen, the solution with the lowest fuel mass in time-optimal solution belongs to $N_p = 5$ with transfer time of 70.95 days and the fuel mass of 614.1 kg. This solution can also be selected as a substitute for the absolute best solution provided in Table 6. Other remark in this figure is the stock of the solutions for $N_p = 1$ in a specific region far from the the desired region. This is another observation which confirms the fact that increasing the number of interpolation points eventually leads the optimizer to find high quality solutions.

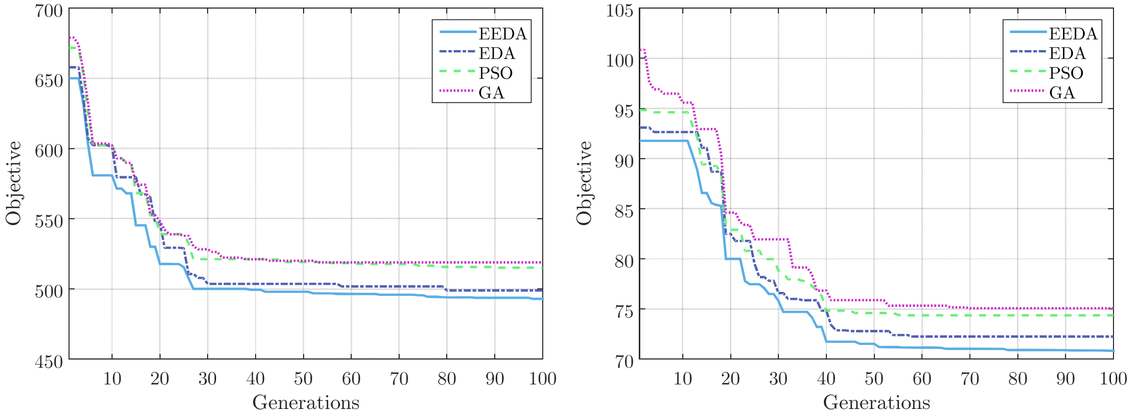


(a) Fuel-optimal transfer.

(b) Time-optimal transfer.

Fig. 11: Obtained solutions after optimization for GTO to Molniya transfer based on simple Lyapunov method.

The other analysis in this space mission is the comparison of the performance of EEDA with other well-known EAs. Such a comparison is shown in Fig. 12a and Fig. 12b.



(a) Fuel-optimal transfer.

(b) Time-optimal transfer.

Fig. 12: Convergence of the algorithms for GTO to Molniya transfer based on simple Lyapunov method.

In Fig. 12a and Fig. 12b the convergence of the proposed algorithm EEDA is compared with normal EDA, PSO, and GA for fuel-optimal and time-optimal transfers. Comparing EEDA and

EDA confirms the effectiveness of the proposed learning mechanism in enhancing the optimization process. Also, it is evident that EDA-based algorithms outperform traditional EAs such as PSO and GA in finding optimal transfer trajectories in this research.

5 Conclusion

In this chapter, the problem of optimal design of low-thrust Earth-orbiting trajectories for space missions is considered. When considering complex trajectory optimization problems with non-convexity and strong non-linearity, it has been demonstrated that intelligent optimization algorithms are effective in optimal trajectory design. To tackle the extreme complexity of the search space, an enhanced evolutionary algorithm within the framework of EDAs, named EEDA is proposed and applied to an approach based on Lyapunov and Q-law methods. In both methods, the unknown weighting coefficients are interpolated with Hermite polynomials and the optimizer is implemented to optimize the variation of weights for minimizing the transfer time and the fuel consumption. Simulation results are carried out to demonstrate the effectiveness of the proposed approach. It has been realized that the implementation of EEDA with simple Lyapunov function is competitive to the Q-law method. The comparison shows the potential of the EEDA in enabling the simple Lyapunov controller to recover the finer nuances explicitly given within the analytical expressions in the Q-law. Also, it has been discovered that increasing the number of interpolation points does not necessarily increase the chance of achieving the optimal solution. In other words, the choice of the proper number of interpolation points is problem-dependence and needs to be adjusted according to the type of the orbit transfer mission.

Current research can be extended in various aspects. Regarding the algorithm enhancement, various further improvements can be the aim for the future research. As for the seeding mechanism, the improvement can be towards obtaining initial feasible population in a more efficient method. For instance, the current mechanism does not use any information from the gradient of the solution domain. Therefore, future works can be conducted in considering gradient-based methods within the seeding mechanism to improve the process. Incorporation of such techniques, more specifically, gradient-based stochastic operators, in minimization of the objective function is also a new area for further research.

References

1. J. T. Betts, *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming.* Society for Industrial and Applied Mathematics, 2010.
2. B. A. Conway, *Spacecraft trajectory optimization.* Cambridge University Press, 2010.
3. A. Shirazi, J. Ceberio, and J. A. Lozano, “Spacecraft trajectory optimization: A review of models, objectives, approaches and solutions,” *Progress in Aerospace Sciences*, vol. 102, pp. 76–98, oct 2018.
4. R. Chai, A. Savvaris, A. Tsourdos, S. Chai, and Y. Xia, “Unified multiobjective optimization scheme for aeroassisted vehicle trajectory planning,” *Journal of Guidance, Control, and Dynamics*, vol. 41, no. 7, pp. 1521–1530, Jul. 2018. [Online]. Available: https://doi.org/10.2514/1.g003189
5. G. qun Wu, L.-G. Tan, X. Li, and S.-M. Song, “Multi-objective optimization for time-open lambert rendezvous between non-coplanar orbits,” *International Journal of Aeronautical and Space Sciences*, vol. 21, no. 2, pp. 560–575, Nov. 2019. [Online]. Available: https://doi.org/10.1007/s42405-019-00231-z

6. M. Pontani, “Optimal low-thrust hyperbolic rendezvous for earth-mars missions,” *Acta Astronautica*, vol. 162, pp. 608–619, 2019.
7. J. A. Englander and B. A. Conway, “Automated solution of the low-thrust interplanetary trajectory problem,” *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 1, pp. 15–27, 2017.
8. S. Sarno, J. Guo, M. D’Errico, and E. Gill, “A guidance approach to satellite formation reconfiguration based on convex optimization and genetic algorithms,” *Advances in Space Research*, vol. 65, no. 8, pp. 2003–2017, Apr. 2020. [Online]. Available: https://doi.org/10.1016/j.asr.2020.01.033
9. Z. Zheng, J. Guo, and E. Gill, “Distributed onboard mission planning for multi-satellite systems,” *Aerospace Science and Technology*, vol. 89, pp. 111–122, Jun. 2019. [Online]. Available: https://doi.org/10.1016/j.ast.2019.03.054
10. Z. Fan, M. Huo, N. Qi, Y. Xu, and Z. Song, “Fast preliminary design of low-thrust trajectories for multi-asteroid exploration,” *Aerospace Science and Technology*, vol. 93, p. 105295, Oct. 2019. [Online]. Available: https://doi.org/10.1016/j.ast.2019.07.028
11. X. Li, X. Wang, and Y. Xiong, “A combination method using evolutionary algorithms in initial orbit determination for too short arc,” *Advances in Space Research*, vol. 63, no. 2, pp. 999–1006, Jan. 2019. [Online]. Available: https://doi.org/10.1016/j.asr.2018.08.036
12. J. A. Lozano, P. Larrañaga, I. Inza, and E. Bengoetxea, *Towards a new evolutionary computation: advances on estimation of distribution algorithms.* Springer, 2006, vol. 192.
13. C. A. Kluever, “Simple guidance scheme for low-thrust orbit transfers,” *Journal of Guidance, Control, and Dynamics*, vol. 21, no. 6, pp. 1015–1017, 1998.
14. A. E. Petropoulos, “Simple control laws for low-thrust orbit transfers,” *AAS/AIAA Astrodynamics Specialists Conference*, 2003.
15. A. Petropoulos, “Refinements to the Q-law for low-thrust orbit transfers,” in *Advances in the Astronautical Sciences*, vol. 120, 2005, pp. 963–982.
16. Y. Gao and X. Li, “Optimization of low-thrust many-revolution transfers and Lyapunov-based guidance,” *Acta Astronautica*, vol. 66, no. 1-2, pp. 117–129, 2010.
17. J. L. Shannon, M. T. Ozimek, J. A. Atchison, and M. Christine, “Q-law aided Direct Trajectory Optimization for the High-fidelity, Many-revolution Low-thrust Orbit Transfer Problem,” in *AAS*, no. 19, 2019, p. 448.
18. S. Lee, A. Petropoulos, and P. von Allmen, “Low-thrust Orbit Transfer Optimization with Refined Q-law and Multi-objective Genetic Algorithm,” *Advances In The Astronautical Sciences*, 2005.
19. G. Varga and J. M. Sánchez Pérez, “Many-Revolution Low-Thrust Orbit Transfer Computation Using Equinoctial Q-Law Including J2 and Eclipse Effects,” *Icatt 2016*, pp. 2463–2481, 2016.
20. A. E. Petropoulos, Z. B. Tarzi, G. Lantoine, T. Dargent, and R. Epenoy, “Techniques for designing many-revolution electric-propulsion trajectories,” *Advances in the Astronautical Sciences*, vol. 152, no. 3, pp. 2367–2386, 2014.
21. D. L. Yang, B. Xu, and L. Zhang, “Optimal low-thrust spiral trajectories using Lyapunov-based guidance,” *Acta Astronautica*, vol. 126, pp. 275–285, 2016.
22. K. F. Wakker, *Fundamentals of Astrodynamics.* TU Delft Library, 2015.
23. R. H. Battin, *An Introduction to the Mathematics and Methods of Astrodynamcis.* New York: AIAA Education Series, 1987.
24. H. Holt, R. Armellin, N. Baresi, Y. Hashida, A. Turconi, A. Scorsoglio, and R. Furfaro, “Optimal q-laws via reinforcement learning with guaranteed stability,” *Acta Astronautica*, vol. 187, pp. 511–528, 2021.
25. J. L. Shannon, D. Ellison, and C. Hartzell, “Analytical Partial Derivatives of the Q-Law Guidance Algorithm,” in *AAS*, 2021, pp. 1–15.
26. H. Schaub and J. L. Junkins, *Analytical Mechanics of Space Systems, Fourth Edition.* American Institute of Aeronautics and Astronautics, Inc., 2018.
27. M. Pontani and M. Pustorino, “Nonlinear earth orbit control using low-thrust propulsion,” *Acta Astronautica*, vol. 179, pp. 296–310, 2021.
28. J. R. Rice, *Numerical Methods in Software and Analysis.* Elsevier, 2014.
29. E. Catmull and R. Rom, “A class of local interpolating splines,” in *Computer Aided Geometric Design.* Elsevier, 1974, pp. 317–326.
30. F. N. Fritsch and R. E. Carlson, “Monotone piecewise cubic interpolation,” *SIAM Journal on Numerical Analysis*, vol. 17, no. 2, pp. 238–246, 1980.
31. C. De Boor, C. De Boor, E.-U. Mathématicien, C. De Boor, and C. De Boor, *A practical guide to splines.* springer-verlag New York, 1978, vol. 27.

32. P. Larrañaga and J. A. Lozano, *Estimation of distribution algorithms: A new tool for evolutionary computation.* Springer Science & Business Media, 2001, vol. 2.
33. D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," Stanford, Tech. Rep., 2006.
34. V. Hodge and J. Austin, "A survey of outlier detection methodologies," *Artificial intelligence review*, vol. 22, no. 2, pp. 85–126, 2004.
35. S. Geffroy and R. Epenoy, "Optimal low-thrust transfers with constraints - Generalization of averaging techniques," *Acta Astronautica*, vol. 41, no. 3, pp. 133–149, 1997.
36. A. Petropoulos, "Low-thrust orbit transfers using candidate lyapunov functions with a mechanism for coasting," in *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, 2004, p. 5089.