

IAC-04-A.6.03

OPTIMIZATION OF VERY-LOW-THRUST TRAJECTORIES USING EVOLUTIONARY NEUROCONTROL

Bernd Dachwald

German Aerospace Center (DLR), Institute of Space Simulation, Linder Hoehe, 51147 Cologne
bernd.dachwald@dlr.de

ABSTRACT

Searching optimal interplanetary trajectories for low-thrust spacecraft is usually a difficult and time-consuming task that involves much experience and expert knowledge in astrodynamics and optimal control theory. This is because the convergence behavior of traditional local optimizers, which are based on numerical optimal control methods, depends on an adequate initial guess, which is often hard to find, especially for very-low-thrust trajectories that necessitate many revolutions around the sun. The obtained solutions are typically close to the initial guess that is rarely close to the (unknown) global optimum. Within this paper, trajectory optimization problems are attacked from the perspective of artificial intelligence and machine learning. Inspired by natural archetypes, a smart global method for low-thrust trajectory optimization is proposed that fuses artificial neural networks and evolutionary algorithms into so-called evolutionary neurocontrollers. This novel method runs without an initial guess and does not require the attendance of an expert in astrodynamics and optimal control theory. This paper details how evolutionary neurocontrol works and how it could be implemented. The performance of the method is assessed for three different interplanetary missions with a thrust to mass ratio $< 0.15 \text{ mN/kg}$ (solar sail and nuclear electric).

INTRODUCTION

This paper deals with the problem of searching optimal interplanetary trajectories for very-low-thrust spacecraft. Two propulsion systems are considered: solar sails (large ultra-lightweight reflecting surfaces that utilize solely the freely available solar radiation pressure for propulsion), and a nuclear electric propulsion (NEP) system with constant thrust and specific impulse. The optimality of a trajectory can be defined according to several objectives, like transfer time or propellant consumption. Because solar sails do not consume any propellant, their trajectories are typically optimized with respect to transfer time alone. Trajectory optimization for spacecraft with a SEP system is less straightforward because transfer time minimization and propellant minimization are mostly competing objectives, so that one objective can only be optimized at the cost of the other objective. Spacecraft trajectories can also be classified with respect to the terminal constraint. If, at arrival, the position \mathbf{r}_{sc}

and the velocity $\dot{\mathbf{r}}_{\text{sc}}$ of the spacecraft must match that of the target (\mathbf{r}_{T} and $\dot{\mathbf{r}}_{\text{T}}$, respectively), one has a rendezvous problem. If only the position must match, one has a flyby problem. A spacecraft trajectory is obtained from the (numerical) integration of the spacecraft's equations of motion. Besides the inalterable external forces, the trajectory $\mathbf{x}_{\text{sc}}[t] = (\mathbf{r}_{\text{sc}}[t], \dot{\mathbf{r}}_{\text{sc}}[t])$ is determined entirely by the variation of the thrust vector ('[t]' denotes the time history of the preceding variable). The thrust vector $\mathbf{F}(t)$ of low-thrust propulsion systems is a continuous function of time. It is manipulated through the n_u -dimensional spacecraft control function $\mathbf{u}(t)$ that is also a continuous function of time. The trajectory optimization problem is to find the optimal spacecraft control function $\mathbf{u}^*(t)$ that yields the optimal trajectory $\mathbf{x}_{\text{sc}}^*[t]$. This problem can not be solved except for very simple cases. What *can* be solved, at least numerically, however, is a discrete approximation of the problem. Dividing the allowed transfer time interval $[t_0, t_{f,\text{max}}]$ into τ finite elements, the discrete trajectory optimization problem is then to find the optimal spacecraft control his-

tory $\mathbf{u}^*[t] \in \mathbb{R}^{n_u \tau}$ that yields the optimal trajectory $\mathbf{x}_{sc}^*[t]$ (the symbol t denotes a discrete time step; note that only the spacecraft control function is discrete, whereas the trajectory is still continuous). Through discretization, the problem of finding the optimal function $\mathbf{u}^*(t)$ in infinite-dimensional function space is reduced to the problem of finding the optimal control history $\mathbf{u}^*[t]$ in a $n_u \tau$ -dimensional parameter space (a space which is usually still very high-dimensional). For optimality, some cost function J must be minimized. If the propellant mass m_p is to be minimized, $J = m_p(\bar{t}_0) - m_p(\bar{t}_f) = \Delta m_p$ is an appropriate cost function, if the transfer time is to be minimized, $J = \bar{t}_f - \bar{t}_0 = T$ is an appropriate cost function.

TRADITIONAL LOCAL LOW-THRUST TRAJECTORY OPTIMIZATION METHODS

Traditionally, low-thrust trajectories are optimized by the application of numerical optimal control methods that are based on the calculus of variations. These methods can be divided into direct methods, such as nonlinear programming (NLP) methods, and indirect methods, such as neighboring extremal methods and gradient methods. All these methods can generally be classified as *local* trajectory optimization methods (LTOMs), where the term optimization does not mean finding *the best* solution but rather finding *a* solution [1]. Prior to optimization, the NLP methods and the gradient methods require an initial guess for the control history $\mathbf{u}[t]$, whereas the neighboring extremal methods require an initial guess for the starting adjoint vector of LAGRANGE multipliers $\boldsymbol{\lambda}(\bar{t}_0)$ (costate vector) [2]. Unfortunately, the convergence behavior of LTOMs (especially of indirect methods) is very sensitive to the initial guess, so that an adequate initial guess is often hard to find, even for an expert in astrodynamics and optimal control theory. Similar initial guesses often produce very dissimilar optimization results, so that the initial guess can not be improved iteratively and trajectory optimization becomes more of an art than science [3]. Even if the optimizer finally converges to an optimal trajectory, this trajectory is typically close to the initial guess that is rarely close to the (unknown) global optimum. Because the optimization process requires nearly permanent attendance of the expert, the search for a good trajectory can become very time-consuming and expensive. Another drawback of LTOMs is the fact that the initial conditions (launch date, initial propellant mass, hyperbolic excess velocity vector, etc.) – although they are crucial for mission performance – are generally chosen

according to the expert's judgment and are therefore not part of the actual optimization method.

EVOLUTIONARY NEUROCONTROL: A SMART GLOBAL LOW-THRUST TRAJECTORY OPTIMIZATION METHOD

To evade the drawbacks of LTOMs, a *smart global* trajectory optimization method (GTOM) was developed by the author [4]. This method was termed InTrance, which stands for **I**ntelligent **T**rajectory optimization using **n**euro**c**ontroller **e**volution. To find a near-globally¹ optimal trajectory, InTrance requires only the target body and intervals for the initial conditions as input. Implementing evolutionary neurocontrol (ENC), InTrance runs without an initial guess and does not require the attendance of a trajectory optimization expert. The remainder of this section will sketch the motivation for ENC and explain the underlying concepts, as well as the application of ENC to solve low-thrust trajectory optimization problems.

1. Motivation for Evolutionary Neurocontrol

ENC fuses artificial neural networks (ANNs) and evolutionary algorithms (EAs) to so-called evolutionary neurocontrollers (ENCs). Like the underlying concepts, it is inspired by the *natural* processes of information processing and optimization. Animal nervous systems incorporate *natural* evolutionary neurocontrollers to control their actions, giving them marvelous capabilities. The smart flight control system of the housefly might provide an example. The nervous system of the housefly comprises about 10^5 neurons. This small natural neural network manages the flight control of the fly, as well as many even more difficult tasks. Nature has optimized the performance of the fly's neurocontroller on this tasks through the recombination and mutation of the fly's genetic material and through natural selection: smarter flies produce more offspring and there is a high probability that some of them are even smarter than their parents. So, if a natural evolutionary neurocontroller is able to steer a housefly optimally from A to B, why should an artificial evolutionary neurocontroller not be able to steer a spacecraft optimally from A to B, which seems to be a much simpler problem? The remainder of this section will describe how such an artificial evolutionary neurocontroller could be implemented.

¹Near-globally optimal because global optimality can rarely be proved for real-world problems.

2. Machine Learning

Within the field of artificial intelligence, one important and difficult class of learning problems are reinforcement learning problems, where the optimal behavior of the learning system (called agent) has to be learned solely through interaction with the environment, which gives an immediate or delayed evaluation² J (also called reward or reinforcement) [5, 6]. The agent's behavior – its strategy – is defined by an associative mapping from situations to actions $S : \mathcal{X} \mapsto \mathcal{A}$ ³. The optimal strategy S^* of the agent is defined as the one that maximizes the sum of positive reinforcements and minimizes the sum of negative reinforcements over time. If, given a situation $X \in \mathcal{X}$, the agent tries an action $A \in \mathcal{A}$ and the environment *immediately* returns a scalar evaluation $J(X, A)$ of the (X, A) pair, one has an immediate reinforcement learning problem. A more difficult class of learning problems are delayed reinforcement learning problems, where the environment gives only a single evaluation $J(X, A)[t]$, collectively for the sequence of (X, A) pairs occurring in time during the agent's operation.

From the perspective of machine learning, a spacecraft steering strategy may be defined as an associative mapping S that gives – at any time along the trajectory – the current spacecraft control $\mathbf{u}(\bar{t})$ from some input $\mathbf{X}(\bar{t}) \in \mathcal{X}$ that comprises the variables that are relevant for the optimal steering of the spacecraft (the current state of the relevant environment). Because the trajectory is the result of the spacecraft steering strategy, the trajectory optimization problem is actually a problem of finding the optimal spacecraft steering strategy S^* . This is a delayed reinforcement problem because a spacecraft steering strategy can not be evaluated before its trajectory is known. Only then a reward can be given according to the fulfillment of the optimization objective(s) and constraints. One obvious way to implement spacecraft steering strategies is to use artificial neural networks because they have been successfully applied to learn associative mappings for a wide range of problems.

3. Artificial Neural Networks and Neurocontrol

Being inspired by the processing of information in animal nervous systems, ANNs are a computability paradigm that is alternative to conventional serial digital computers. ANNs are massively paral-

lel, analog, fault tolerant, and adaptive [7]. They are composed of processing elements (called neurons) that model the most elementary functions of biological neurons. Linked together, those elements show some characteristics of the brain, for example, learning from experience, generalizing from previous examples to new ones and extracting essential characteristics from inputs containing noisy and/or irrelevant data, so that they are relatively insensitive to minor variations in its input to produce consistent output [8].

Because the neurons can be connected in many ways, ANNs exist in a wide variety. Here, however, only feedforward ANNs are considered. Feedforward ANNs have typically a layered topology, where the neurons are organized in a number of neuron layers. The first neuron layer is called the input layer and has n_i input neurons that receive the network's input. The last neuron layer is called the output layer and has n_o output neurons that provide the network's output. All intermediate layers/neurons are called hidden layers/neurons. A feedforward ANN, as it is used here, can be regarded as a continuous parameterized function (called network function)

$$\mathbf{N}_{\boldsymbol{\pi}} : \mathcal{X} \subseteq \mathbb{R}^{n_i} \rightarrow \mathcal{Y} \subseteq (0, 1)^{n_o}$$

that maps from a set of inputs \mathcal{X} onto a set of outputs \mathcal{Y} . The parameter set $\boldsymbol{\pi} = \{\pi_1, \dots, \pi_m\}$ of the network function comprises the m internal parameters of the ANN (the weights of the neuron connections and the biases of the neurons).

ANNs have been successfully applied as neurocontrollers (NCs) to reinforcement learning problems [8]. An ANN controls a dynamical system by providing a control $\mathbf{Y}(t) \in \mathcal{Y}$ from some input $\mathbf{X}(t) \in \mathcal{X}$ that contains the relevant information for the control task. Note that the NC's behavior is completely characterized by its network function $\mathbf{N}_{\boldsymbol{\pi}}$ (that is again completely characterized by its parameter set $\boldsymbol{\pi}$). If the correct output is known for a set of given inputs (the training set), the difference between the given output and the correct output can be utilized to learn the optimal network function $\mathbf{N}^* := \mathbf{N}_{\boldsymbol{\pi}^*}$ by adapting $\boldsymbol{\pi}$ in a way that minimizes this difference for all input/output pairs in the training set. A variety of learning algorithms has been developed for this kind of learning, the back-propagation algorithm – a gradient-based method – being the most widely known. Unfortunately, learning algorithms that rely on a training set fail when the correct output for a given input is not known, as it is the case for delayed reinforcement learning problems. The next section will address a learning

²This evaluation is analogous to the cost function in optimal control theory. To emphasize this fact, it will be denoted by the same symbol, J .

³ \mathcal{X} is called state space and \mathcal{A} is called action space.

method that may be used for determining \mathbf{N}^* in this case.

4. Evolutionary Algorithms and Evolutionary Neurocontrol

EAs are robust methods for finding global optima in very high dimensional search spaces. They have been successfully applied as a learning method for ANNs, as well as for a wide range of other optimization problems [9, 10, 11]. EAs use a vocabulary borrowed from biology. The key element of an EA is a population that comprises numerous individuals $\xi_{j \in \{1, \dots, q\}}$, which are potential solutions for the given optimization problem. All individuals of the (initially randomly created) population are evaluated according to a fitness function⁴ J for their suitability to solve the problem. The fitness of each individual $J(\xi_j)$ is crucial for its probability to reproduce and to create offspring into a newly created population because fitter individuals are selected with a greater probability for reproduction than less fit ones. The selected parents undergo a series of genetic transformations (mutation, recombination) to produce offspring that consists of a mixture of the parents genetic material. Under this selection pressure, the individuals – also called chromosomes or strings – strive for survival. After some reproduction cycles, the population converges against a single solution ξ^* , which is in the best case the globally optimal solution for the given problem. EAs can be employed for searching the NC's optimal network function because a NC parameter set can be mapped onto a real-valued string that provides an equivalent description of the network function. By searching for the fittest individual, the EA searches for the optimal spacecraft trajectory. Fig. 1 sketches the transformation of a chromosome into a trajectory.

5. Neurocontroller Input and Output

Two fundamental questions concerning the utilization of a NC for spacecraft steering are: (1) "What *input* should the NC get?" (or "What should the NC *know* to steer the spacecraft?") and (2) "What *output* should the NC give?" (or "What should the NC *do* to steer the spacecraft?"). To be robust, a spacecraft steering strategy should be time-independent: to determine the currently optimal spacecraft control $\mathbf{u}(\bar{t}_i)$, the spacecraft steering strategy should

⁴This fitness function is also analogous to the cost function in optimal control theory. To emphasize this fact, it will be denoted by the same symbol, J .

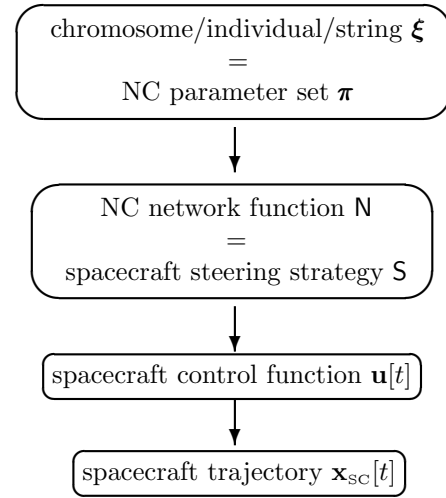


Fig. 1: Transformation of a chromosome into a trajectory

have to know – at *any* time step \bar{t}_i – only the current spacecraft state $\mathbf{x}_{SC}(\bar{t}_i)$ and the current target state $\mathbf{x}_T(\bar{t}_i)$, hence $S : \mathcal{X} = \{(\mathbf{x}_{SC}, \mathbf{x}_T)\} \mapsto \{\mathbf{u}\}$. If a propulsion system other than a solar sail is employed, the current propellant mass $m_P(\bar{t}_i)$ might be considered as an additional input, $S : \mathcal{X} = \{(\mathbf{x}_{SC}, \mathbf{x}_T, m_P)\} \mapsto \{\mathbf{u}\}$. The number of potential input sets, however, is still large because \mathbf{x}_{SC} and \mathbf{x}_T may be given in coordinates of any reference frame and in combinations of them. The difference $\mathbf{x}_T - \mathbf{x}_{SC}$ may be used as well, also in coordinates of any reference frame and in combinations of them. Two potential input sets are depicted in Figs. 2 and 3.

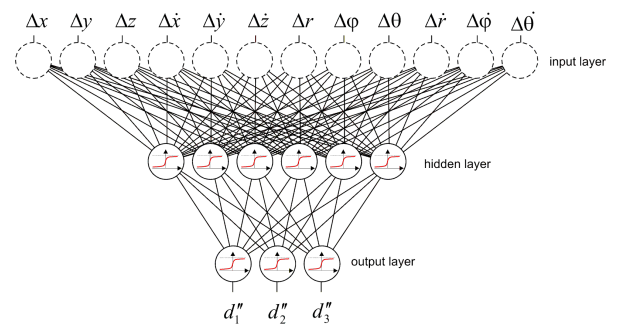


Fig. 2: Example for a NC that implements a solar sail steering strategy

Each output neuron gives a value $Y_i \in (0, 1)$. The number of potential output sets is also large because there are many alternatives to define \mathbf{u} , and to calculate \mathbf{u} from \mathbf{Y} . The following approach gave good results for the majority of problems: the NC provides a three-dimensional output vector $\mathbf{d}'' \in (0, 1)^3$ from which a unit vector \mathbf{d} in the desired thrust di-

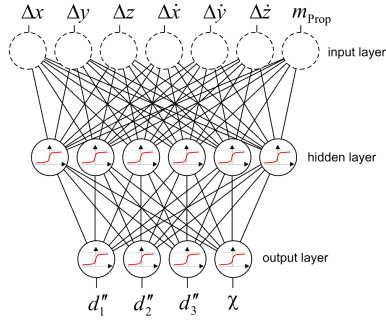


Fig. 3: Example for a NC that implements an SEP spacecraft steering strategy

rection (called direction unit vector) is calculated via

$$\mathbf{d}' = 2\mathbf{d}'' - \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \in (-1, 1)^3 \quad (1)$$

and

$$\mathbf{d} = \mathbf{d}' / |\mathbf{d}'| \quad (2)$$

For solar sails, $\mathbf{u} = \mathbf{d}$, hence $S : \{(\mathbf{x}_{SC}, \mathbf{x}_T)\} \mapsto \{\mathbf{d}\}$ (see Fig. 2). For SEP spacecraft, the output must include the engine throttle χ , so that $\mathbf{u} = (\mathbf{d}, \chi)$, hence $S : \{(\mathbf{x}_{SC}, \mathbf{x}_T, m_P)\} \mapsto \{\mathbf{d}, \chi\}$ (see Fig. 3).

6. Evolutionary Neurocontroller Design

Fig. 4 shows how an ENC may be applied for low-thrust trajectory optimization. To find the optimal spacecraft trajectory, the ENC method runs in two loops.

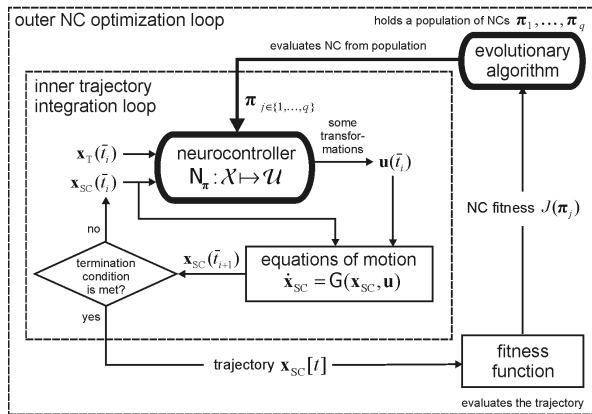


Fig. 4: Low-thrust trajectory optimization using ENC

Within the (inner) trajectory integration loop, a NC steers the spacecraft according to its network

function N_{π_j} that is completely defined by the NC's parameter set π_j . The EA in the (outer) NC optimization loop holds a population of NC parameter sets, $\Xi = \{\pi_1, \dots, \pi_q\}$, and examines all of them for their suitability to generate an optimal trajectory. Within the trajectory optimization loop, the NC takes the current spacecraft state $\mathbf{x}_{SC}(\bar{t}_i \in \{0, \dots, \tau-1\})$ and that of the target $\mathbf{x}_T(\bar{t}_i)$ as input, and maps them onto some output. For SEP spacecraft, the input includes the current propellant mass $m_P(\bar{t}_i)$ and the output includes the current throttle $\chi(\bar{t}_i)$. The first three output values are interpreted as the components of $\mathbf{d}''(\bar{t}_i)$, from which the direction unit vector $\mathbf{d}(\bar{t}_i)$ is calculated via Eq. (1) and (2). Now, the spacecraft control $\mathbf{u}(\bar{t}_i)$ is calculated from the NC output. Then, $\mathbf{x}_{SC}(\bar{t}_i)$ and $\mathbf{u}(\bar{t}_i)$ are inserted into the equations of motion and (numerically) integrated over one time step $\Delta \bar{t} = \bar{t}_{i+1} - \bar{t}_i$ to yield $\mathbf{x}_{SC}(\bar{t}_{i+1})$. The new state is fed back into the NC. The trajectory integration loop stops when the accuracy of the trajectory is sufficient or when a given time limit is reached. Then, back in the NC optimization loop, the NC's trajectory is rated by the fitness function $J(\pi_j)$. The fitness of π_j is crucial for its probability to reproduce and to create offspring. Under this selection pressure, the EA breeds more and more suitable steering strategies that generate better and better trajectories. Finally, the EA converges against a single steering strategy, which gives in the best case a near-globally optimal trajectory $\mathbf{x}_{SC}^*[t]$.

7. Additionally Encoded Problem Parameters

If an EA is already employed for the optimization of the NC, it is manifest to use it also for the co-optimization of additional problem parameters. This can be done without major additional effort. InTrance encodes the following parameters additionally on the chromosome, making them an explicit part of the optimization problem: (1) the launch date, (2) the launch velocity vector (hyperbolic excess velocity vector), and (3) the initial propellant mass (except for solar sails).

RESULTS

Within this section, ENC (InTrance) is applied to three interplanetary very-low-thrust trajectory optimization problems. The first problem is a near-Earth asteroid rendezvous using a solar sail (with a characteristic acceleration⁵ of $a_c = 0.14 \text{ mm/s}^2$), the second problem is a MESSENGER-like mission to Mercury but using a solar sail (with $a_c =$

⁵maximum acceleration at Earth distance from the sun

0.1 mm/s²), and the third problem is a Jupiter flyby using a NEP spacecraft (with an acceleration – or thrust to mass ratio – of 0.0357 mm/s² < $a(t) = F/m(t) \leq 0.05$ mm/s²).

1. Near-Earth Asteroid Rendezvous Mission Using a Solar Sail

Within this section, the convergence behavior of ENC and the quality of the obtained solutions is assessed for an exemplary rendezvous mission to a near-Earth asteroid (1996FG₃). For solar sailcraft with $a_c = 0.14$ mm/s² (ideal⁶ 50 m × 50 m solar sail, launch mass 148 kg, useful mass⁷ 75 kg) a trajectory was calculated in [12, 13] using a LTOM. This reference trajectory launches from Earth on 13 Aug 06 and takes 1640 days to rendezvous 1996FG₃, if the solar sailcraft is inserted directly into an interplanetary trajectory with an hyperbolic excess energy of $C_3 = 4$ km²/s².

In the first experiment, InTrance was run five times for the reference launch date but with zero hyperbolic excess energy. The maximum transfer time was set to $T_{\max} = 1800$ days. For discretization, this time interval was cut into $\tau = 360$ finite elements of equal length, so that the solar sail is allowed to change its attitude once every 5 days. The accuracy limit for the distance and the relative velocity at the target was set to $\Delta r_{f,\max} = 0.3 \cdot 10^6$ km and $\Delta v_{f,\max} = 0.1$ km/s, respectively, which is compatible with the reference trajectory. The best found InTrance-trajectory (Fig. 5) is 135 days faster than the reference trajectory, while reducing at the same time the C_3 -requirement from 4 km²/s² to 0 km²/s², thus permitting a reduction of the launcher requirements and eventually of launch costs. The final distance to 1996FG₃ is $\Delta r_f = 0.200 \cdot 10^6$ km and the final relative velocity is $\Delta v_f = 0.065$ km/s, both being better than the required values.

In the second experiment, InTrance was used to find the optimal launch date for the 1996FG₃ rendezvous problem (with $C_3 = 0$ km²/s²). Fig. 6(a) shows the solutions of five InTrance runs with different initial NC populations. The small variance of the five solutions gives evidence for a good convergence behavior of ENC. Taking 1435 days to rendezvous 1996FG₃, the best found trajectory (Fig. 6(b)) is 205 days faster than the reference trajectory ($\Delta r_f = 0.267 \cdot 10^6$ km and $\Delta v_f = 0.089$ km/s). The optimal launch date was found to be 22 Oct 05, 295 days earlier than the reference launch date.

⁶where the incident radiation is reflected specularly

⁷spacecraft bus plus scientific payload

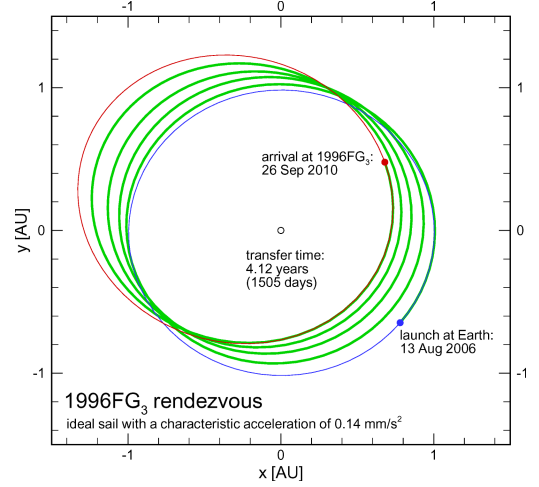
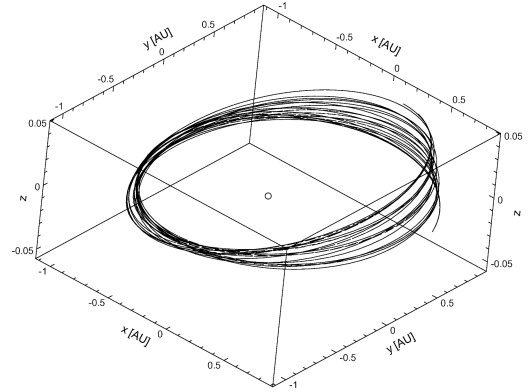
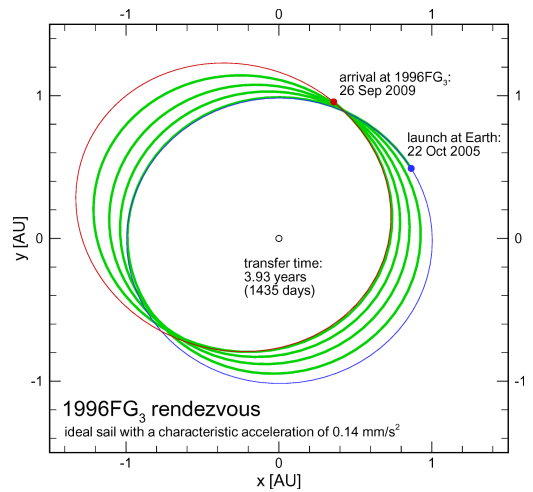


Fig. 5: Best InTrance-trajectory for the 1996FG₃ rendezvous (reference launch date)



(a) Trajectories for five different initial populations



(b) Best InTrance-trajectory

Fig. 6: 1996FG₃ rendezvous (optimized launch date)

The third experiment was to find out whether a given C_3 of $4 \text{ km}^2/\text{s}^2$ could be spent more efficiently than done by the reference trajectory. The optimal launch date for this problem was found to be 12 Feb 06, a half year earlier than the reference launch date. Fig. 7 shows the best found trajectory. It takes only 944 days to rendezvous 1996FG₃, being 696 days faster than the reference trajectory ($\Delta r_f = 0.272 \cdot 10^6 \text{ km}$ and $\Delta v_f = 0.091 \text{ km/s}$). For this calculation, the solar sail was allowed to change its attitude once every 4 days ($T_{\text{max}} = 1000 \text{ days}$, $\tau = 250$).

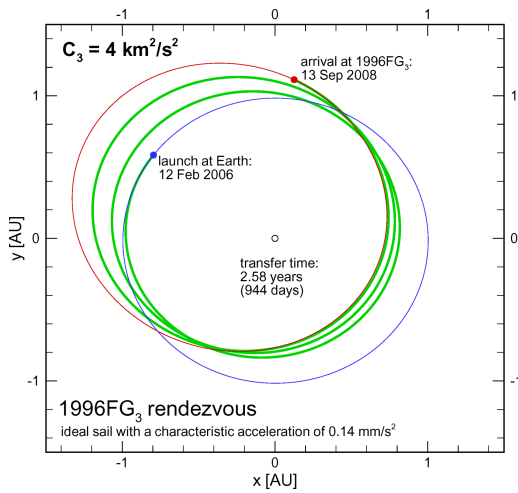


Fig. 7: 1996FG₃ rendezvous (optimized launch date): Best InTrance-trajectory for $C_3 = 4 \text{ km}^2/\text{s}^2$

2. Mercury Rendezvous Mission Using a Solar Sail

Within this section, a rendezvous mission to Mercury is assessed, comparable to the MESSENGER mission (same spacecraft dry mass to Mercury, approximately same transfer time) but using a solar sail instead of the chemical propulsion system. MESSENGER makes an Earth-Venus-Venus-Mercury-Mercury-Mercury gravity assist and five deep space maneuvers to reach the planet within about 6.6 years. Table 1 gives the most important mission parameters [14].

InTrance was used to derive the performance requirements for solar sails that would be able to transport a spacecraft with the dry mass of MESSENGER to Mercury within approximately the same transfer time. The maximum transfer time was set to $T_{\text{max}} = 2600 \text{ days}$. For discretization, this time interval was cut into $\tau = 520$ finite elements of equal length, so that the solar sail is allowed to change its attitude once every 5 days. The

Spacecraft dry mass	499.4 kg
Launch	02 Aug 04
Launch mass	1099.5 kg
Launcher	Delta II, Model 7925-H
C_3	$16.8 \text{ km}^2/\text{s}^2$
Gravity assists	Earth-(2×)Venus-(3×)Mercury
Transfer Time	6.623 years
Mercury orbit insertion	18 Mar 11

Table 1: MESSENGER mission parameters [14]

accuracy limit for the distance and the relative velocity at the target was set to $\Delta r_{f,\text{max}} = 10^6 \text{ km}$ and $\Delta v_{f,\text{max}} = 0.25 \text{ km/s}$, respectively. For a fair comparison, MESSENGER's C_3 was also used for the solar sail option. Fig. 8 shows a possible solar sail trajectory for such a mission.

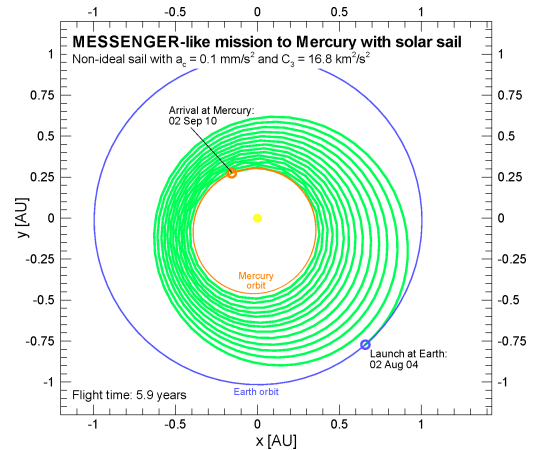


Fig. 8: Trajectory for MESSENGER-like mission to Mercury using a solar sail

Fig. 8 demonstrates that a solar sail with a moderate characteristic acceleration of 0.1 mm/s^2 is able to reach Mercury within approximately the MESSENGER transfer time, if it is injected with the same hyperbolic excess energy of $16.8 \text{ km}^2/\text{s}^2$. In contrast to the chemical mission scenario, no gravity assist maneuver is necessary to achieve the required ΔV , which results in a more flexible mission profile, with a practically permanent launch window [4]. Table 2 summarizes the significant mission parameters.

The launch date for the solar sail mission was not optimized but adopted from the actual MESSENGER mission. An optimization of the launch date might yield a slightly shorter flight time. The MESSENGER-trajectory, however, is also not glob-

Solar sail payload	499.4 kg
Launch	02 Aug 04
Characteristic acceleration	0.1 mm/s ²
C_3	16.8 km ² /s ²
Gravity assists	–
Transfer Time	5.914 years
Mercury orbit insertion	02 Sep 10

Table 2: Significant parameters for a MESSENGER-like mission using a solar sail

ally optimal because the originally envisaged launch window (10 Mar 04) was missed. A launch on 10 Mar 04 with $C_3 = 15.2 \text{ km}^2/\text{s}^2$ would have resulted in a shorter flight time of 5.5 years, leading to arrival at Mercury on 06 Apr 09 [15].

Assuming a quadratic solar sail, Fig. 9 gives the sail side length s that is required to achieve $a_c = 0.1 \text{ mm/s}^2$ for different sail assembly⁸ loadings σ_{SA} and payload masses m_{PL} , where the bold line relates to MESSENGER's dry mass of 500 kg.

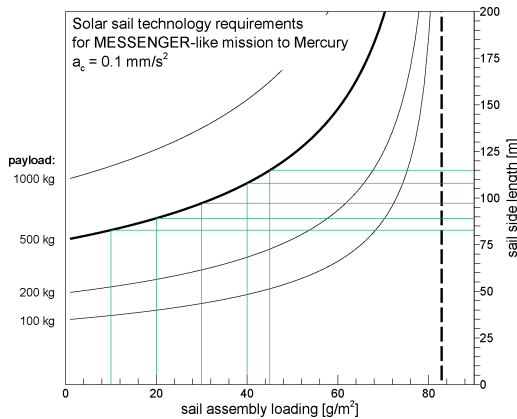


Fig. 9: Solar sail technology requirements for MESSENGER-like mission to Mercury using a solar sail

It can be seen that the required sail size increases drastically for $\sigma_{SA} \gtrsim 45 \text{ g/m}^2$, and that it approaches infinity for $\sigma_{SA} \approx 83 \text{ g/m}^2$ because the solar sail would be too heavy to achieve $a_c = 0.1 \text{ mm/s}^2$ even without any payload. Table 3 gives values for five points on the 500 kg-curve.

It can be seen that the sail assembly loading must only be $\lesssim 45 \text{ g/m}^2$ to yield a benefit in launch mass with respect to the MESSENGER mission. For an advanced solar sail with a low sail assembly loading of about 10 g/m^2 the launch mass is only about one half of the actual MESSENGER launch mass.

⁸the sail film and the required structure for storing, deploying and tensioning the sail

sail assembly loading $\sigma_{SA} [\text{g/m}^2]$	sail size $s [\text{m}]$	sail assembly mass $m_{SA} [\text{kg}]$	launch mass $m [\text{kg}]$
10	83	67	568
20	89	159	658
30	97	284	783
40	108	466	966
45	115	594	1093

Table 3: Trade-off between sail size s and sail assembly loading σ_{SA} for $a_c = 0.1 \text{ mm/s}^2$

3. Jupiter Flyby Mission Using Nuclear Electric Propulsion

InTrance was originally developed for the optimization of solar sail trajectories and later extended to optimize also trajectories for electric spacecraft. Within this section, it is applied for an exemplary Jupiter flyby mission with a NEP spacecraft, to assess the suitability of ENC also for the optimization of electric very-low-thrust trajectories. Unlike for the previous examples, the optimization objective was to minimize the used propellant mass for a transfer within a maximum flight time of $T_{\max} = 10\,000$ days (27.4 years, $\tau = 2000$). To find the absolute minimum – independent of the actual constellation of Earth and Jupiter – no flyby at Jupiter itself but only a crossing of Jupiter's orbit within a distance of less than 10^6 km was required, and InTrance was allowed to vary the launch date within a one year interval.

Without consideration of technical feasibility, a launch mass of $m = 14 \text{ t}$ (for $C_3 = 0 \text{ km}^2/\text{s}^2$) was chosen, including a minimum dry mass of $m_{\text{dry}} = 10 \text{ t}$. The propulsion system was assumed to provide a thrust of $F = 0.5 \text{ N}$, so that the thrust to mass ratio is $0.0357 \text{ mN/kg} < F/m(t) \leq 0.05 \text{ mN/kg}$. The specific impulse was assumed to be that of a state-of-the-art ion thruster, $I_{\text{sp}} = 4500 \text{ s}$ [16]. Fig. 10 shows a possible trajectory for this mission. The velocity increment of this trajectory is $\Delta V = 10.5 \text{ km/s}$, whereas a minimum ΔV HOHMANN-type trajectory to Jupiter⁹ would require $\Delta V = 8.6 \text{ km/s}$. Therefore, the gravitational loss of the NEP system is only about 21%. Due to the high specific impulse of the electric thruster, only about 3 t of propellant are required. In contrast, a propellant mass of about 94 t would be required for a HOHMANN-type transfer of a 11 t-spacecraft with a chemical propulsion system

⁹at perihelion

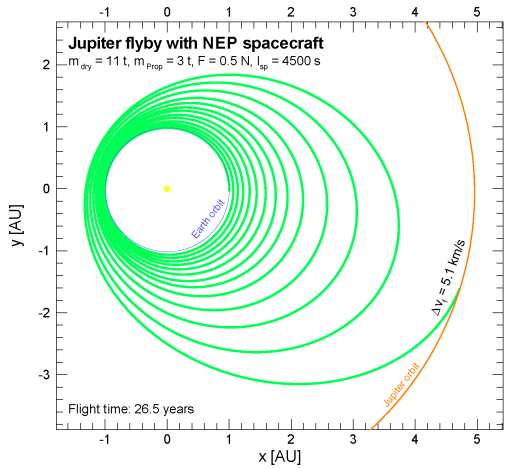


Fig. 10: Jupiter flyby using NEP

(H₂/LOX, $I_{sp} = 390$ s [17]). Note also that the relative velocity at Jupiter is larger for the HOHMANN-type transfer ($\Delta v_f = 5.9$ km/s) than for the very-low-thrust transfer ($\Delta v_f = 5.1$ km/s). This result demonstrates that a NEP system allows very good payload ratios for high- ΔV missions, if the transfer time plays a subordinate role, so that the gravitational losses can be kept small.

SUMMARY AND CONCLUSIONS

Within this paper, low-thrust trajectory optimization problems have been attacked from the perspective of artificial intelligence and machine learning. Inspired by natural archetypes, a novel method for spacecraft trajectory optimization was proposed. It fuses artificial neural networks and evolutionary algorithms into evolutionary neurocontrollers. Evolutionary neurocontrol was implemented within a program termed InTrance, which stands for **I**ntelligent **T**rajectory optimization using **n**euro**c**ontroller **e**volution. InTrance was applied to find near-globally optimal trajectories for three exemplary interplanetary very-low-thrust problems: a near-Earth asteroid rendezvous using a solar sail, a MESSENGER-like mission to Mercury but using a solar sail, and a Jupiter flyby using a spacecraft with a nuclear electric propulsion system. For the near-Earth asteroid rendezvous problem, InTrance found a trajectory that is considerably better than a reference trajectory found by a human trajectory optimization expert using a local trajectory optimization method. It was already shown in previous papers that evolutionary neurocontrollers are able to find spacecraft steering strategies that generate better trajectories, which are closer to the global

optimum because they explore the trajectory search space more exhaustively than a human expert can do by using traditional optimal control methods. Here, it was shown that evolutionary neurocontrol can also be applied successfully for very-low-thrust problems, where it is very difficult to generate an adequate initial guess for local trajectory optimization methods. Evolutionary neurocontrol runs without an initial guess and does not require the attendance of an expert in astrodynamics and optimal control theory. It may be also used to find the optimal initial conditions for the mission. Being problem-independent, the application field of evolutionary neurocontrol may be extended to a variety of other optimal control problems.

References

- [1] M. Vasile. A global approach to optimal space trajectory design. Ponce, Puerto Rico, February 2002. AAS/AIAA Space Flight Mechanics Meeting. AAS 03-141.
- [2] R. F. Stengel. *Optimal Control and Estimation*. Dover Books on Mathematics. Dover Publications, Inc., New York, 1994.
- [3] V. Coverstone-Carroll, J. W. Hartmann, and J. W. Mason. Optimal multi-objective low-thrust spacecraft trajectories. *Computer Methods in Applied Mechanics and Engineering*, 186:387–402, 2000.
- [4] B. Dachwald. *Low-Thrust Trajectory Optimization and Interplanetary Mission Analysis Using Evolutionary Neurocontrol*. Doctoral thesis, Universität der Bundeswehr München; Fakultät für Luft- und Raumfahrttechnik, 2004.
- [5] S. S. Keerthi and B. Ravindran. A tutorial survey of reinforcement learning. Technical report, Department of Computer Science and Automation, Indian Institute of Science, Bangalore, 1995.
- [6] R. S. Sutton and A. G. Barto. *Reinforcement Learning*. MIT Press, Cambridge, London, 1998.
- [7] R. Rojas. Was können neuronale Netze? In R.-H. Schulz, editor, *Mathematische Aspekte der angewandten Informatik*, pages 55–88. Wissenschaftsverlag, Mannheim, 1994. (in German).
- [8] D. C. Dracopoulos. *Evolutionary Learning Algorithms for Neural Adaptive Control*. Perspectives in Neural Computing. Springer, Berlin, Heidelberg, New York, 1997.
- [9] D. Whitley, S. Dominic, R. Das, and C. W. Anderson. Genetic reinforcement learning for neuro-control problems. *Machine Learning*, 13:259–284, 1993.
- [10] L. Tsinas and B. Dachwald. A combined neural and genetic learning algorithm. In *Proceedings of the 1st IEEE Conference on Evolutionary Computation*,

- IEEE World Congress on Computational Intelligence, 27–29 June 1994, Orlando, USA*, volume 2, pages 770–774, Piscataway (NJ), USA, 1994. IEEE.
- [11] X. Yao. Evolutionary artificial neural networks. In A. Kent et al., editor, *Encyclopedia of Computer Science and Technology*, volume 33, pages 137–170. Marcel Dekker Inc., New York, 1995.
 - [12] E. K. Jessberger, W. Seboldt, K.-H. Glassmeier, G. Neukum, M. Pätzold, G. Arnold, H.-U. Auster, D. deNiém, F. Guckenbiehl, B. Häusler, G. Hahn, N. Hanowski, A. Harris, H. Hirsch, E. Kührt, M. Leipold, E. Lorenz, H. Michaelis, D. Möhlmann, S. Mottola, D. Neuhaus, H. Palme, H. Rauer, M. Rezazad, L. Richter, D. Stöffler, R. Willnecker, J. Brückner, G. Klingelhöfer, and T. Spohn. ENEAS – exploration of near-Earth asteroids with a sailcraft. Technical report, August 2000. Proposal for a Small Satellite Mission within the Space Sciences Program of DLR.
 - [13] W. Seboldt, M. Leipold, M. Rezazad, L. Herbeck, W. Unkenbold, D. Kassing, and M. Eiden. Ground-based demonstration of solar sail technology. Rio de Janeiro, Brazil, 2000. 51st International Astronautical Congress. IAF-00-S.6.11.
 - [14] MESSENGER Web Site. messenger.jhuapl.edu.
 - [15] A. G. Santo, R. E. Gold, R. L. McNutt Jr., S. C. Solomon, C. J. Ercol, R. W. Farquhar, T. J. Hartka, J. E. Jenkins, J. V. McAdams, L. E. Mosher, D. F. Persons, D. A. Artis, R. S. Bokulic, R. F. Conde, G. Dakermanji, M. E. Goss, D. R. Haley, K. J. Heeres, R. H. Maurer, R. C. Moore, E. H. Rodberg, T. G. Stern, S. R. Wiley, B. G. Williams, C. L. Yen, and M. R. Peterson. The MESSENGER mission to Mercury: Spacecraft and mission design. *Planetary and Space Science*, 49:1481–1500, 2001.
 - [16] QinetiQ T6 ion propulsion system. Summary Information Issue 01, QinetiQ.
 - [17] P. Fortescue and J. Stark. *Spacecraft Systems Engineering*. John Wiley & Sons, Chichester, New York, Brisbane, second edition, 1995.