Deutscher Luft- und Raumfahrtkongress 2004, Dresden, Germany

LOW-THRUST TRAJECTORY OPTIMIZATION AND INTERPLANETARY MISSION ANALYSIS USING EVOLUTIONARY NEUROCONTROL

B. Dachwald
Deutsches Zentrum für Luft- und Raumfahrt e.V.
Institut für Raumsimulation
Linder Höhe, 51147 Köln

ABSTRACT

The design and optimization of interplanetary transfer trajectories is one of the most important tasks during the analysis and design of a deep space mission. Due to their larger ΔV-capability, low-thrust propulsions systems can significantly enhance or even enable those missions. Searching low-thrust trajectories that are optimal with respect to transfer time or propellant consumption is usually a difficult and time-consuming task that involves much experience and expert knowledge, because the convergence behavior of traditional optimizers that are based on numerical optimal control methods depends strongly on an adequate initial guess, which is often hard to find. 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. Within this paper, trajectory optimization is attacked from the perspective of artificial intelligence and machine learning, which is a perspective quite different from that of optimal control theory. Inspired by natural archetypes, a novel smart method for global low-thrust trajectory optimization is presented that fuses artificial neural networks and evolutionary algorithms to so-called evolutionary neurocontrollers. This paper outlines how evolutionary neurocontrol works and how it could be implemented. Using evolutionary neurocontrol, low-thrust trajectories are optimized without an initial guess and without the attendance of an expert in astrodynamics and optimal control theory. For an exemplary mission to a near-Earth asteroid, its performance for low-thrust trajectory optimization and interplanetary mission analysis is assessed. It is demonstrated that evolutionary neurocontrollers are able to find spacecraft steering strategies that generate better trajectories – closer to the global optimum – because they explore the search space more exhaustively than a human expert can do by using traditional optimal control methods. Finally, the use of evolutionary neurocontrol for the analysis of a piloted Mars mission using a spacecraft with a nuclear electric propulsion system is demonstrated within this paper.

1 LOW-THRUST TRAJECTORY OPTIMIZATION

Innovative solar system exploration missions require ever larger velocity increments (ΔVs) and thus ever more demanding propulsion capabilities. Using the state-of-the-art technique of chemical propulsion in combination with (eventually multiple) gravity assist maneuvers for those high-energy missions results in long, complicated, and inflexible mission profiles. Low-thrust propulsions systems can significantly enhance or even enable high-ΔV missions, since they utilize the propellant more efficiently – like electric propulsion systems – or do not consume any propellant at all – like solar sails, large ultra-lightweight reflecting surfaces that utilize solely the freely available solar radiation pressure for propulsion. Consequently, they permit significantly larger ΔVs and/or larger payload ratios and/or smaller launch vehicles, while at the same time allowing direct trajectories with reduced flight times, simpler mission profiles, and extended launch windows, providing more mission flexibility.

This paper deals with the problem of searching optimal interplanetary trajectories for low-thrust spacecraft. In simple words, the spacecraft trajectory is the spacecraft's path in space from A (the initial body or orbit) to B (the target body or orbit). Optimality can, in general, be defined according to several objectives like transfer time or propellant consumption. Since solar sails do not consume any propellant, their trajectories are typically optimized with respect to transfer time alone. Trajectory optimization for spacecraft with an electric propulsion system is less straightforward, since 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 $\mathbf{r}_{\mathrm{SC}}$ and the velocity $\dot{\mathbf{r}}_{\mathrm{SC}}$ of the spacecraft must match that of the target body ($\mathbf{r}_{\mathrm{T}}$ and $\dot{\mathbf{r}}_{\mathrm{T}}$, respectively), one has a rendezvous problem. If only the position must match, one has a fly-by problem. A spacecraft trajectory is obtained from the (numerical) integration of the spacecraft's equations of motion, which contain terms for the external forces that are acting on the spacecraft (gravitational forces and "disturbing" forces) and for the thrust force. Besides the inalterable external forces, the trajectory is determined entirely by the variation of the thrust vector $\mathbf{F}(t)$, which is typically described by a control function $\mathbf{u}(t)$. Therefore, the actual optimization problem is to find the optimal spacecraft control function $\mathbf{u}^{\star}(t)$ that yields the optimal trajectory $\mathbf{x}_{\mathrm{SC}}[t] = (\mathbf{r}_{\mathrm{SC}}[t], \dot{\mathbf{r}}_{\mathrm{SC}}[t])$, where '$[t]$' denotes the time history of the preceding variable.

For spacecraft with high thrust propulsion systems like chemical rockets, optimal interplanetary trajectories can be found relatively easily[1], since only a few thrust phases are necessary. These thrust phases are very short as compared to the transfer time, so that they can be approximated by singular events that change the spacecraft's velocity instantaneously while its position remains fixed. In contrast to those high-thrust propulsion systems, low-thrust propulsion systems must operate for a significant part of the transfer to generate the necessary ΔV. Consequently, the thrust vector $\mathbf{F}(t)$ is a continuous function of time and the dimension of the solution space is infinite. $\mathbf{F}(t)$ is manipulated through the n_u-dimensional spacecraft control function $\mathbf{u}(t)$ that is also a continuous function of time. Therefore, the trajectory optimization problem is to find the optimal spacecraft control function $\mathbf{u}^\star(t)$ in infinite-dimensional function space. 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. For doing that, the infinite-dimensional problem must be converted into a finite-dimensional problem by numerical discretization. Dividing the maximum allowed transfer time interval $[t_0, t_{f,\max}]$ into τ finite elements, the approximate trajectory optimization problem is then to find an optimal spacecraft control history[2] $\mathbf{u}^\star[\bar{t}] \in \mathbb{R}^{n_u \tau}$, which gives the optimal trajectory $\mathbf{x}^\star_{\mathrm{SC}}[t]$.[3] Through discretization, the problem of finding $\mathbf{u}^\star(t)$ as an optimal function in infinite-dimensional function space is reduced to the problem of finding the optimal control history $\mathbf{u}^\star[\bar{t}]$ in a finite-dimensional parameter space, a space which is usually still very high-dimensional. In terms of optimal control theory, the discrete rendezvous problem, for example, can be stated formally as:

Discrete rendezvous problem from the perspective of optimal control theory:

Find a spacecraft control history $\mathbf{u}[\bar{t}]$ ($\bar{t} \in \{\bar{t}_0, \ldots, \bar{t}_{f-1} \leq \bar{t}_{\tau-1}\}$), which forces the state $\mathbf{x}_{\mathrm{SC}}(t) = (\mathbf{r}_{\mathrm{SC}}(t), \dot{\mathbf{r}}_{\mathrm{SC}}(t))$ of the spacecraft from its initial value $\mathbf{x}_{\mathrm{SC}}(\bar{t}_0)$ to the state $\mathbf{x}_{\mathrm{T}}(\bar{t})$ of the target body, along a trajectory that obeys the dynamic constraint $\dot{\mathbf{x}}_{\mathrm{SC}}(t) = \mathsf{G}(\mathbf{x}_{\mathrm{SC}}(t), \mathbf{u}(t))$ and the terminal constraint $\mathbf{x}_{\mathrm{SC}}(\bar{t}_f) = \mathbf{x}_{\mathrm{T}}(\bar{t}_f)$, and at the same time minimizes some cost function J.

The resulting state function $\mathbf{x}^\star_{\mathrm{SC}}[t]$ is the optimal trajectory for the given problem. Thus the trajectory optimization problem is actually a problem of finding the optimal control history $\mathbf{u}^\star[\bar{t}]$. If the propellant mass m_{P} is to be minimized, $J = m_{\mathrm{P}}(\bar{t}_f) - m_{\mathrm{P}}(\bar{t}_0)$ is an appropriate cost function, if the transfer time is to be minimized, $J = \bar{t}_f - \bar{t}_0$ is an appropriate cost function.

2 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 be generally 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 $\mathbf{u}[\bar{t}]$, 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]. Figure 1 illustrates how trajectory optimization is usually performed with a LTOM.

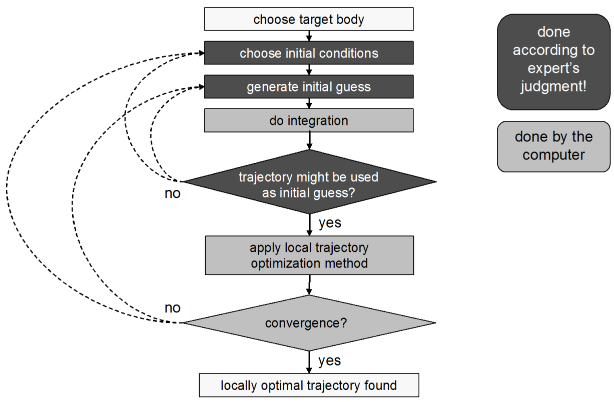

Figure 1: Low-thrust trajectory optimization using a local trajectory optimization method

First, the target body and the initial conditions (launch date, initial propellant mass, hyperbolic excess velocity vector, etc.) are chosen according to the mission objectives and the launcher restrictions. Although those parameters are crucial for mission performance, they are typically chosen according to an expert's judgment and are not part of the actual optimization process. After that, the initial guess for the control history (in the case of a NLP or gradient method) is generated and numerically integrated to obtain a trajectory. The objective is to come as close as possible to the target body, so that in the next step the LTOM is able to converge. If the generated trajectory does not come close to the target body, the initial guess has to be refined and – if several trial-and-error cycles yield no acceptable result – the initial conditions have to be modified (e.g. different launch date and/or more initial propellant and/or larger hyperbolic excess velocity). If the trajectory finally comes close enough to the target body, it is taken as the initial guess for the LTOM. If the LTOM does not converge, a new initial guess must be conceived and the above steps must be repeated, using eventually different initial conditions again. If the LTOM finally converges, a locally optimal trajectory is found, which is typically close to the initial guess that is rarely close to the global optimum. Unfortunately, the convergence behavior of LTOMs (especially of indirect methods) is very sensitive to the initial guess. Similar initial guesses often produce very dissimilar optimization results, so that trajectory optimization becomes sometimes "more art than science" [3]. Since all steps require frequent manual interactions and thus the permanent attendance of an expert in astrodynamics and

[1] as long as no gravity assist maneuvers are required
[2] the symbol $\bar{t}$ denotes that time is descrete
[3] note that only the spacecraft control function is discretized, whereas the trajectory is still continuous

optimal control theory, the search for a good trajectory can become very time-consuming and thus expensive.

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

Emanating from the drawbacks of traditional LTOMs, a *smart global* trajectory optimization method (GTOM), as it is sketched in figure 2, was developed in [4]. This method was termed "InTrance", which stands for "**In**telligent **Tra**jectory optimization using **n**euro**c**ontroller **e**volution".

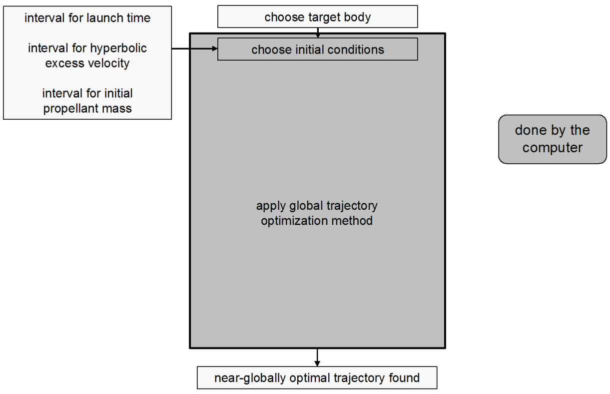

Figure 2: Low-thrust trajectory optimization using a smart global trajectory optimization method

InTrance requires only the target body and intervals for the initial conditions as input to find a near-globally[4] optimal trajectory for the specified problem. Implementing evolutionary neurocontrol as described in the sequel, it works without an initial guess and without the permanent attendance of a trajectory optimization expert. The remainder of this section will explain the motivation for evolutionary neurocontrol and the underlying concepts.

3.1 Motivation for Evolutionary Neurocontrol

Evolutionary neurocontrol (ENC) fuses artificial neural networks (ANNs) with evolutionary algorithms (EAs) to so-called evolutionary neurocontrollers (ENCs). Like the underlying constructs, 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. One brilliant example for this proposition is the smart flight control system of the housefly. The nervous system of the housefly comprises about 100 000 neurons. This natural neural network manages the flight control of the fly as well as many even more difficult tasks like finding food, finding a mate, producing offspring, etc. Nature has optimized the performance of the fly's neurocontroller on this tasks with respect to one single objective: survive to produce offspring. Nature has solved this problem through the recombination and mutation of the fly's genetic material and through natural selection, the famous so-called "survival of the fittest": smarter flies produce more offspring and there is a high probability that some of them are even smarter than their parents. This very elegant optimization process runs without initial guess and without employing the calculus of variations! So, if a natural evolutionary neurocontroller can 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? The remainder of this section explains how this could be done.

3.2 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), as it is defined by an associative mapping from situations to actions $\mathsf{S} : \mathcal{X} \mapsto \mathcal{A}$,[5] has to be learned solely through interaction with the environment, which gives an immediate or delayed evaluation[6] J (also called reward or reinforcement) of the agent's behavior [5, 6]. The optimal strategy $\mathsf{S}^\star$ 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, collectively for $(X, A)[t]$, the sequence of (X, A) pairs occurring in time during the agent's operation.

3.3 Low-Thrust Trajectory Optimization from the Perspective of Machine Learning

From the perspective of machine learning, a spacecraft steering strategy may be defined as an associative mapping S that gives the actual spacecraft control $\mathbf{u}(t)$ from some input $\mathbf{X}(t) \in \mathcal{X}$ that comprises the variables that are important for the optimal steering of the spacecraft (the actual state of the relevant environment). The trajectory can then be regarded as the result of the spacecraft steering strategy. The search for the optimal strategy is a delayed reinforcement problem, since a strategy can be evaluated only ex post, when the trajectory is realized and a reward can be given according to the fulfillment of the optimization objective(s). From the perspective of machine learning, the rendezvous problem, which was stated above from the perspective of optimal control theory, may now be reformulated:

Discrete rendezvous Problem from the perspective of machine learning:

Find a spacecraft steering strategy S, which forces the state $\mathbf{x}_{\mathrm{SC}}(t) = (\mathbf{r}_{\mathrm{SC}}(t), \dot{\mathbf{r}}_{\mathrm{SC}}(t))$ of the spacecraft from its initial value $\mathbf{x}_{\mathrm{SC}}(\bar{t}_0)$ to the state $\mathbf{x}_{\mathrm{T}}(\bar{t})$ of the target body, along a trajectory that obeys the dynamic constraint $\dot{\mathbf{x}}_{\mathrm{SC}}(t) = \mathsf{G}(\mathbf{x}_{\mathrm{SC}}(t), \mathbf{u}(t))$ and the terminal con-

[4] *near*-globally optimal, since for "real-world" problems global optimality can rarely be proved

[5] $\mathcal{X}$ is called state space and $\mathcal{A}$ is called action space

[6] this evaluation is analogous to the cost function in optimal control theory. To emphasize this fact, it will be denoted by the same letter, J

straint $\mathbf{x}_{\text{SC}}(\bar{t}_f) = \mathbf{x}_{\text{T}}(\bar{t}_f)$, and at the same time maximizes some reward J.

The resulting steering strategy $\mathsf{S}^\star$ is the optimal spacecraft steering strategy for the given problem. Thus the trajectory optimization problem is actually a problem of finding the optimal spacecraft steering strategy $\mathsf{S}^\star$. A very obvious way to implement spacecraft steering strategies is to use artificial neural networks, as they have been successfully applied to "learn" associative mappings for a wide range of problems.

3.4 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 parallel, analog, fault tolerant, and adaptive [7]. They are composed of processing elements (called neurons) that model the most elementary functions of the biological neuron. Linked together, those elements show some characteristics of the brain, like 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].

Since the neurons can be connected in many ways, ANNs exist in a wide variety. Here, however, only feedforward ANNs are considered. Typically, feedforward ANNs have a layered topology, where the neurons are organized hierarchically in a number of so-called 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 layered feedforward ANN, as it is used here, can be regarded as a continuous parameterized function (called network function)

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

that maps from a set of inputs $\mathcal{X}$ onto a set of outputs $\mathcal{Y}$. The parameter set $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_m)$ of the network function comprises the m internal parameters[7] of the ANN.

Operating within so-called neurocontrollers (NCs), ANNs have been successfully applied to reinforcement learning problems [8]. Neurocontrol approaches to solve reinforcement learning problems can be divided into two categories, *indirect* (or critic-based) ones and *direct* ones. The direct neurocontrol approach, which is used here, employs a single ANN, which is called the action model (or action network). The action network controls the dynamical system by providing a control $\mathbf{Y}(t) \in \mathcal{Y}$ from some input $\mathbf{X}(t) \in \mathcal{X}$ that contains the information that is relevant to perform the control task. To keep things simple, the term 'NC' is used here for the ANN that is precisely speaking 'the action network of the NC'.

NCs can be applied to low-thrust trajectory optimization, which is a delayed reinforcement learning problem: if a NC is used to direct the spacecraft's trajectory by determining the spacecraft control at each time step $\bar{t}_i$, then this NC receives a single reward for its control history $\mathbf{u}[\bar{t}]$ (i.e. for its behavior) at the final time step $\bar{t}_f$, when the trajectory can be evaluated. Note that the NC's behavior is completely characterized by its network function $\mathsf{N}_{\boldsymbol{\pi}}$ (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 network error (i.e. the difference between the actual output and the correct output) can be measured and utilized to learn the optimal network function $\mathsf{N}^\star := \mathsf{N}_{\boldsymbol{\pi}^\star}$ by adapting $\boldsymbol{\pi}$ in a way that the network error is minimized. A variety of learning algorithms has been developed for this kind of learning problems, the backpropagation algorithm – a gradient-based method – being the most widely known. However, learning algorithms for ANNs 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 method that may be used for determining $\mathsf{N}^\star$ in this case.

3.5 Evolutionary Algorithms and Evolutionary Neurocontrol

EAs (sometimes also called GAs, genetic algorithms) are proven to be robust methods for finding global optima in very high dimensional search spaces. They have been successfully applied as a learning method for ANNs [9, 10, 11] as well as for a wide range of other optimization problems. Therefore, they are expected to be an efficient method for finding the NC's optimal network function in the case of delayed reinforcement learning problems.

EAs use a vocabulary borrowed from biology. The key element of an EA is a population that comprises numerous individuals $\boldsymbol{\xi}_j$ $(j \in \{1, ..., q\})$, which are potential solutions to the given optimization problem. All individuals of the (initially randomly created) population are evaluated according to a fitness function[8] J for their suitability to solve the problem. The fitness value of an individual $J(\boldsymbol{\xi}_j)$ is crucial for its probability to reproduce and to create offspring into a newly created population, because a selection scheme (the environment) selects fitter individuals 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 the selection pressure of the environment, the individuals – which are also called chromosomes or strings – strive for survival. After some reproduction cycles the population converges against a single solution $\boldsymbol{\xi}^\star$, which is in the best case the globally optimal solution for the given problem.

The application of an EA to search for the NC's optimal network function makes use of the fact that a NC parameter set can be mapped onto a real valued string, which provides an equivalent description of the NC's network

[7] the weights and the biases of the neurons

[8] this fitness function is also analogous to the cost function in optimal control theory. To emphasize this fact, it will be denoted by the same letter, J

function. By searching for the fittest individual the EA searches for the NC's optimal network function. Figure 3 sketches the transformation of the optimal chromosome into the optimal trajectory.

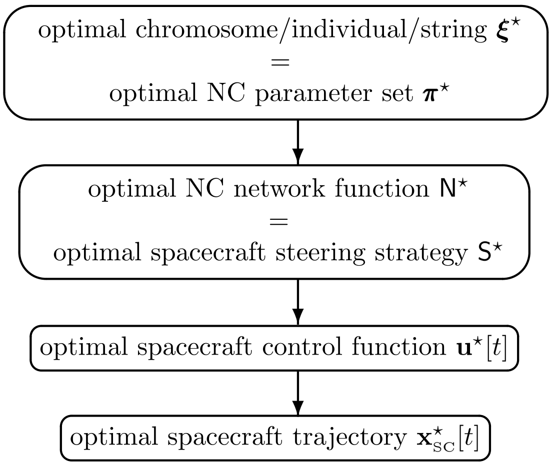

Figure 3: Transformation of the optimal chromosome into the optimal trajectory

3.6 Neurocontroller Input and Output

The 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 not depend *explicitly* on time. To determine the actual optimal spacecraft control $\mathbf{u}(\bar{t}_i)$, the spacecraft steering strategy should "have to know" – at *any* time step $\bar{t}_i$ – only the actual spacecraft state $\mathbf{x}_{\mathrm{SC}}(\bar{t}_i)$ and the actual target body state $\mathbf{x}_{\mathrm{T}}(\bar{t}_i)$:

$$\mathsf{S} : \mathcal{X} = \{(\mathbf{x}_{\mathrm{SC}}, \mathbf{x}_{\mathrm{T}})\} \mapsto \{\mathbf{u}\}.$$

If a spacecraft propulsion system other than a solar sail is employed, the actual propellant mass $m_{\mathrm{P}}(\bar{t}_i)$ might be considered as additional input:

$$\mathsf{S} : \mathcal{X} = \{(\mathbf{x}_{\mathrm{SC}}, \mathbf{x}_{\mathrm{T}}, m_{\mathrm{P}})\} \mapsto \{\mathbf{u}\}.$$

The number of potential input sets, however, is still large: $\mathbf{x}_{\mathrm{SC}}$ and $\mathbf{x}_{\mathrm{T}}$ may be given in coordinates of any reference frame and in combinations of them. Also the difference $\mathbf{x}_{\mathrm{T}} - \mathbf{x}_{\mathrm{SC}}$ may be used, again in coordinates of any reference frame and in combinations of them.

The number of potential output sets is also large: each output neuron gives a value in the range $(0, 1)$. There are many alternatives to define the spacecraft control $\mathbf{u}$, and there are again many ways to calculate $\mathbf{u}$ from the NC output. 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 in the desired thrust direction – the so-called direction unit vector $\mathbf{d}$ – is calculated via

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

For solar sailcraft, $\mathbf{u} = \mathbf{d}$ and thus

$$\mathsf{S} : \{(\mathbf{x}_{\mathrm{SC}}, \mathbf{x}_{\mathrm{T}})\} \mapsto \{\mathbf{d}\}.$$

For electrically propelled spacecraft, the NC output must include the engine throttle χ, so that $\mathbf{u} = (\mathbf{d}, \chi)$ and thus

$$\mathsf{S} : \{(\mathbf{x}_{\mathrm{SC}}, \mathbf{x}_{\mathrm{T}}, m_{\mathrm{P}})\} \mapsto \{\mathbf{d}, \chi\}.$$

3.7 Implementation of Evolutionary Neurocontrol

This section outlines, how ENC is applied within InTrance for low-thrust trajectory optimization. To find the optimal spacecraft trajectory, the ENC method is running in two loops (see figure 4).

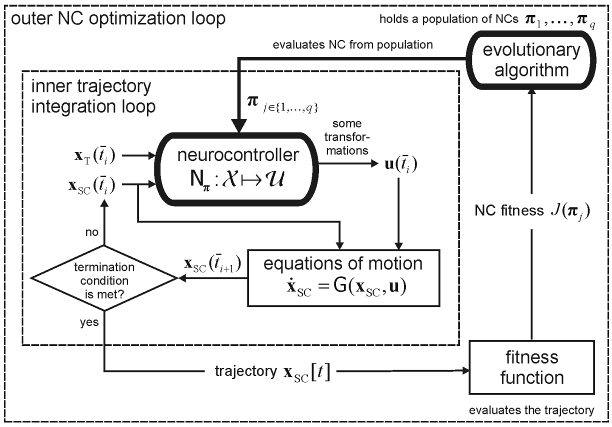

Figure 4: Low-thrust trajectory optimization using evolutionary neurocontrol

Within the inner trajectory integration loop, an NC steers the spacecraft according to its network function $\mathsf{N}_{\boldsymbol{\pi}}$ that is completely defined by the NC's parameter set $\boldsymbol{\pi}$, which is set and evaluated by the EA in the outer NC optimization loop. The EA holds a population $\{\boldsymbol{\pi}_1, \ldots, \boldsymbol{\pi}_q\}$ of NCs (i.e. NC parameter sets) and uses the trajectory integration loop to test all population members $\boldsymbol{\pi}_j$ for their suitability to generate an optimal trajectory. Within the trajectory optimization loop, the NC takes the actual spacecraft state $\mathbf{x}_{\mathrm{SC}}(\bar{t}_i)$ (starting from $\bar{t}_0$) and that of the target body $\mathbf{x}_{\mathrm{T}}(\bar{t}_i)$ as input, and maps from them onto some output, from which – after some transformations – the actual spacecraft control $\mathbf{u}(\bar{t}_i)$ is calculated. Then, $\mathbf{x}_{\mathrm{SC}}(\bar{t}_i)$ and $\mathbf{u}(\bar{t}_i)$ are inserted into the equations of motion, which are numerically integrated over one time step to yield $\mathbf{x}_{\mathrm{SC}}(\bar{t}_{i+1})$. This state is fed back into the NC. The trajectory integration loop runs until the accuracy of the trajectory with respect to the terminal constraint is sufficient or until some time limit is reached. Then, back in the NC optimization loop, the NC's trajectory is rated by the EA's fitness function. This fitness value is crucial for the individual's probability to reproduce and to create offspring. Under the selection pressure of the environment the EA breeds NCs that generate more and more suitable steering strategies that in turn generate better and better trajectories. The EA finally converges against a single steering strategy, which gives in the best case the globally optimal trajectory $\mathbf{x}^{\star}_{\mathrm{SC}}[t]$, or at least a near-globally optimal one.

If an EA is already employed for the evolution of the NC, it is manifest to employ this EA also for the parallel optimization of additional problem parameters, which can be done without major additional effort. Therefore, in InTrance, the following parameters are additionally encoded on the chromosome, making them an explicit part of the optimization problem: (1) the launch date, (2) the hyperbolic excess velocity vector, and (3) the initial propellant mass.

4 RESULTS

To assess the performance of ENC for low-thrust trajectory optimization, InTrance was used to re-calculate trajectories for problems, for which trajectories have been be found in the literature, providing the best benchmarks available (henceforth called reference problems/trajectories).

Within this paper, the convergence behavior of ENC and the quality of the obtained solutions is assessed for an exemplary rendezvous mission to a near-Earth asteroid (1996FG$_3$).[9] For solar sailcraft with a characteristic acceleration[10] of 0.14 mm/s^2 (ideally reflecting 50 m × 50 m solar sail, launch mass 148 kg, useful mass[11] 75 kg) an "optimal" trajectory was calculated in refs. [14, 15] using a LTOM. This reference trajectory launches at Earth at 13 Aug 06 and takes 1640 days to rendezvous 1996FG$_3$, if the solar sailcraft is inserted directly into an interplanetary trajectory with an hyperbolic excess energy of $C_3 = 4\,\mathrm{km}^2/\mathrm{s}^2$.

In the first experiment, InTrance was run five times – using different initial NC populations – for the reference launch date, but the hyperbolic excess energy was reduced to $C_3 = 0\,\mathrm{km}^2/\mathrm{s}^2$. The accuracy limit for the distance and the relative velocity at the target was set to 300 000 km and 100 m/s, respectively, which is compatible with the reference trajectory. The best found InTrance-trajectory (figure 5) is 135 days faster (9.0%) than the reference trajectory, while reducing at the same time the C_3-requirement from $4\,\mathrm{km}^2/\mathrm{s}^2$ to $0\,\mathrm{km}^2/\mathrm{s}^2$, thus permitting a reduction of the launcher requirements and eventually of launch costs. The final distance to 1996FG$_3$ is approx. 200 000 km and the final relative velocity to 1996FG$_3$ is approx. 65 m/s, both being better than the required values.

In the second experiment, InTrance was used to find the optimal launch date for the 1996FG$_3$ rendezvous problem (with $0\,\mathrm{km}^2/\mathrm{s}^2$). Since the optimal launch date is not evident, it was encoded additionally on the chromosome, leaving it to the EA to co-evolve it with the NC.

[9] the results for further reference missions can be found in [4, 12, 13]
[10] maximum acceleration of a solar sail at Earth distance from the sun
[11] spacecraft bus plus scientific payload

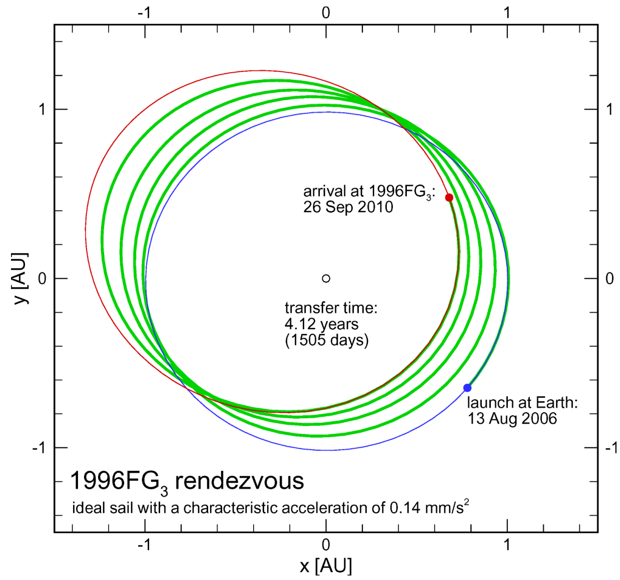

Figure 5: Best InTrance-trajectory for the 1996FG$_3$ rendezvous (reference launch date)

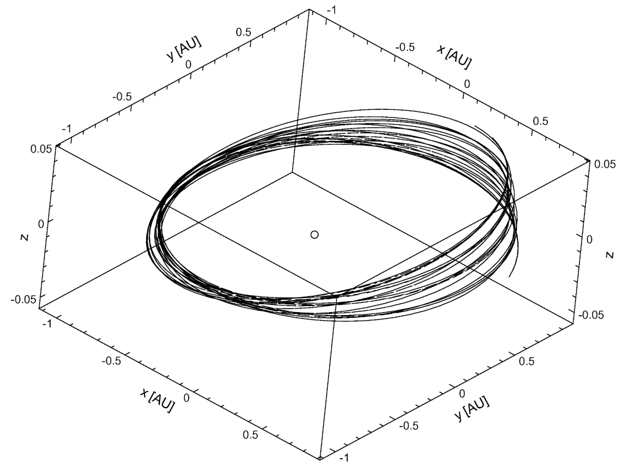

Figure 6: 1996FG$_3$ rendezvous (optimized launch date): Trajectories for five different initial NC populations

Figure 6 shows the results for five runs with different initial NC populations. The worst found trajectory takes only 1.7% longer to rendezvous 1996FG$_3$ than the best one (shown in figure 7). The small variance of the five results gives evidence for a good convergence behavior of ENC. Taking 1435 days to rendezvous 1996FG$_3$, the trajectory is now 205 days (14%) faster than the reference trajectory. The final distance to 1996FG$_3$ is approx. 267 000 km and the final relative velocity to 1996FG$_3$ is approx. 89 m/s, both being better than the required values. The optimal launch date was found to be 22 Oct 05, 295 days earlier than the reference launch date.

In the third experiment, to find out whether a given hyperbolic excess energy of $4\,\mathrm{km}^2/\mathrm{s}^2$ could be spent more efficiently than done by the reference trajectory, the optimal direction of the hyperbolic excess velocity vector was encoded additionally on the chromosome, leaving it to the EA to co-evolve it with the NC. The optimal launch date for this problem was found to be 12 Feb 06, a half year earlier than the reference launch date. Fig-

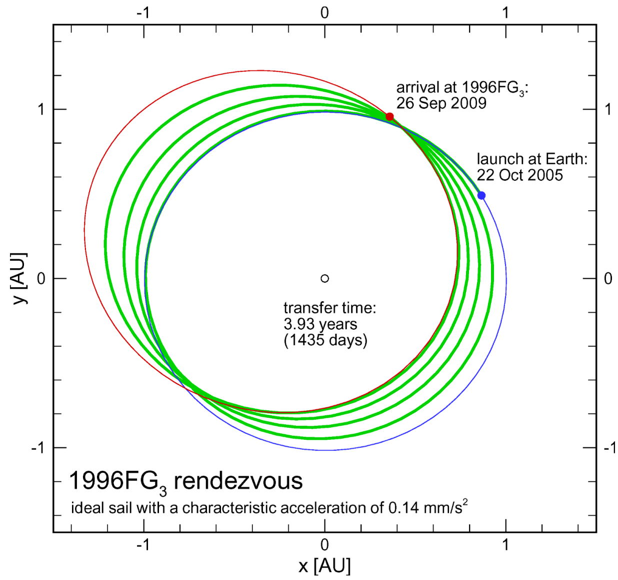

Figure 7: 1996FG$_3$ rendezvous (optimized launch date): Best InTrance-trajectory for $C_3 = 0\,\mathrm{km}^2/\mathrm{s}^2$

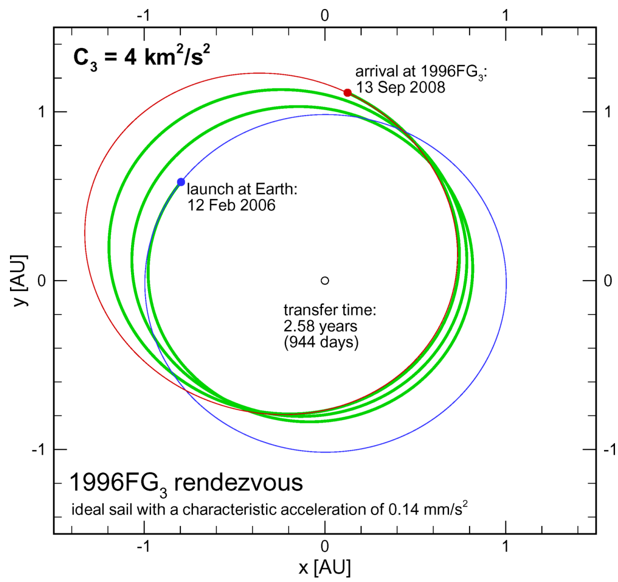

Figure 8: 1996FG$_3$ rendezvous (optimized launch date): Best InTrance-trajectory for $C_3 = 4\,\mathrm{km}^2/\mathrm{s}^2$

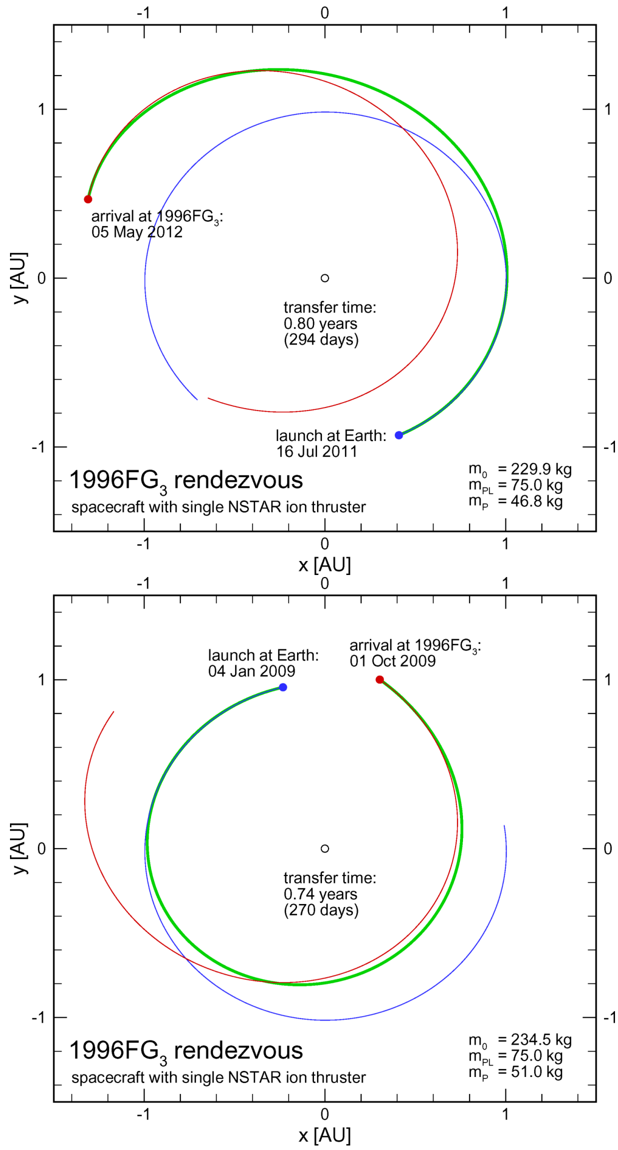

Figure 9: Trajectory options for 1996FG$_3$ rendezvous with SEP spacecraft

ure 8 shows the best found trajectory for this launch date. This trajectory takes only 944 days to rendezvous 1996FG$_3$, being 696 days (74%) faster than the reference trajectory.

To assess the trajectory optimization capability of ENC also for low-thrust propulsion systems other than solar sails, and to assess the capability of near-term solar sail propulsion for this NEA rendezvous mission, InTrance was used to calculate trajectories for spacecraft with an already existing solar electric propulsion (SEP) system, NASA's NSTAR ion thruster, which was flown on the Deep Space 1 mission. The mission objective was as before: deliver a useful mass of 75 kg to 1996FG$_3$.

In contrast to solar sailcraft trajectory optimization, SEP spacecraft trajectories may not only be optimized with respect to transfer time but also with respect to the required propellant mass, and usually a trade-off between both optimization objectives has to be made, so that an "optimal" solution is only one of many (Pareto-) optimal solutions. Figure 9 exemplifies two InTrance-solutions for this problem. Using an NSTAR thruster, the same useful mass of 75 kg could be delivered to 1996FG$_3$ within 294 days (with a propellant mass of 46.8 kg) or even within 270 days, if slightly more propellant (51.0 kg) is consumed.

The results demonstrate that at least for this mission, a near-term solar sail is outperformed by the SEP option, if only the transfer time is considered. This is not surprising, since the required ΔV for the transfer is moderate. The launch mass of the SEP option, however, is larger (229.9 kg and 234.5 kg, respectively) than for the solar sail option (148.0 kg), requiring eventually a heavier and thus more expensive launch vehicle. If ground operation costs can be kept low (e.g. due to a high on-board autonomy during transfer), and if the transfer time plays a subordinate role with respect to cost, the solar sail might be the favorable option for such a mission. In any case, on the way to more advanced solar sails, as they are required for high-ΔV missions, the development of near-term solar sails is an indispensable first stepping stone, even if their performance is not superior to that of state-of-the-art electric propulsion systems.

To perform trajectory optimization with ENC, the following parameters have to be fixed: (1) the NC's input set, (2) the NC's output set, (3) the NC's topology, (4) some EA parameters like population size, mutation rate, etc., and (5) the EA's fitness function. Various combinations of those parameters have been investigated. The performance of the ENC was found to be relatively insensitive with respect to different settings of (3) and (4). The dependency on the EA's fitness function (5) is reasonable, since this function has not only to decide autonomously which trajectories are good and which are not, but also which trajectories are promising for future "cultivation" and which are not. A fitness function that works well for the majority of problems was found. The dependency on the NC's input set (1) and output set (2) is also reasonable, since they determine what the NC knows about its environment and what the NC can do to steer the spacecraft. Again, input and output sets that work well for the majority problems were found.

For all trajectory calculations a standard feedforward ANN with 30 neurons in the hidden layer was used. The maximum number of integration steps was usually set to values between 100 and 400, allowing the NC to change the spacecraft control every $2-5$ days. Depending on the number of integration steps, the total computation time for one run of InTrance was in the order of $1-8$ hours on a 1.3 GHz personal computer.

The InTrance-generated trajectories are quite accurate with respect to the terminal constraint, however, they are *not* optimal solutions in the strict sense, since the terminal constraint is not *exactly* met. To improve the accuracy of the trajectories further, an InTrance-trajectory can be taken as the initial guess for a LTOM.

5 APPLICATION EXAMPLE: MISSION ANALYSIS FOR A PILOTED LOW-THRUST MARS MISSION

To provide a further example of how InTrance can be applied to support space mission analysis, it was used to analyze the feasibility of a piloted Mars mission for spacecraft using a nuclear electric propulsion (NEP) system.

Beyond the ISS and the Moon, Mars is the logical next step towards the manned exploration and conquest of space. Differing from "ordinary" robotic missions due to large payloads and restricted flight times, the feasibility of piloted Mars missions depends crucially on an adequate propulsion system. To reduce the risk for the crew, a short mission duration (of less than approximately two years) and a short stay time (of less than approximately three months) is desirable (fast mission). Such a requirement precludes the application of chemical propulsion systems, which necessitate in this case an immense effort (up to several thousand tons in LEO for an Earth return vehicle mass of about 75 t), since at least one trajectory leg requires a large ΔV. Due to their much larger specific impulse, low-thrust propulsion systems are expected to enable relatively short missions with reasonable effort. InTrance was employed to analyze mission opportunities for an exemplary spacecraft with a NEP system (300 N maximum thrust, 6000 s specific impulse, 160 t launch mass at Earth, 75 t Earth return vehicle mass), providing an illustrative example of how InTrance is recently used at DLR to analyze the capability of various low-thrust propulsion systems to enable fast and flexible piloted Mars missions [16, 17].

Using InTrance, time-optimal trajectories have been found to have three different topologies (A, B, and C, figure 10), depending (1) on the constellation of Earth and Mars at the respective departure, (2) on the closest tolerable solar distance ($r_{\min}$), and (3) on the maximum relative velocity at the target body.[12] Within this categorization, trajectories of type A neither cross the orbit of Earth nor that of Mars. They have short transfer times and require a moderate ΔV. However, Type A trajectories are only possible for favorable constellations of the two planets (type A phase). Trajectories of type B cross the orbit of Earth, moving thereby closer to the sun. They have longer transfer times and require a high to very high ΔV. Trajectories of type C move farther away from the sun than Mars, having long to very long transfer times and a moderate to high ΔV-requirement. Type B and C phases are defined accordingly as the time intervals, in which type B and C trajectories are time-optimal due to the constellation of Earth and Mars at departure. The phases alternate (A→C→B→A→... for the Earth-Mars transfer and A→B→C→A→... for the Mars-Earth transfer) as similar constellations recur.

Figure 11 shows for an Earth return trajectory, how the transfer time varies within one A→B→C-cycle. The left part of the downward slope is associated with type C trajectories, the right part of the downward slope is associated with type A trajectories, and the upward slope is associated with type B trajectories. Thus type C trajectories evolve gradually into type A trajectories, whereas there is a tremendous increase in flight time, when type B trajectories become non-optimal and type C trajectories provide the time-optimal option to return to Earth. As it can be seen, type B trajectories can also be flown later in time, if a closer solar approach is tolerated. In this case, a trade-off has to be made concerning the medical risk for the crew (long transfer-time vs. close solar flyby). A similar diagram can be drawn for the Earth-Mars leg of the mission. What is more meaningful, however, is to plot the transfer time for this leg against the *arrival* date at Mars, together with a plot of the Mars-Earth transfer time against the *departure* date at Mars, as it is done in figure 12 for $r_{\min} = 0.7$ AU.

Looking at the displacement of both curves, one can see that for a short stay at Mars, the combination of a short

[12] within this paper, a maximum relative (hyperbolic) velocity of 6 km/s was used for Mars entry and a maximum (hyperbolic) velocity of 8 km/s was used for Earth entry, in accordance with [18]

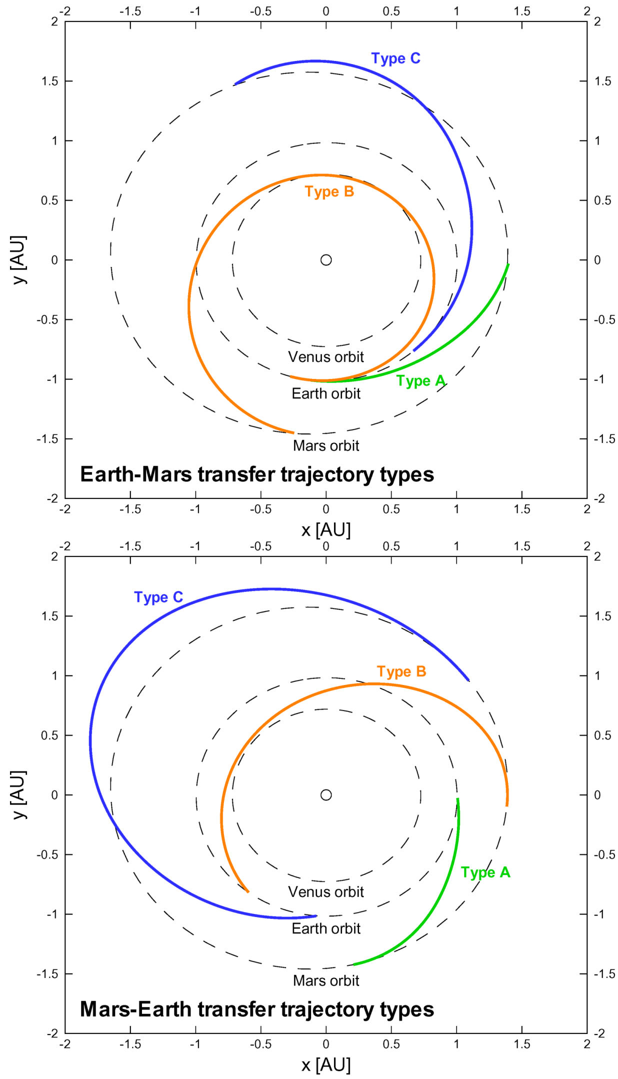

Figure 10: Trajectory types (A, B, and C, see text) for Earth-Mars and Mars-Earth transfers

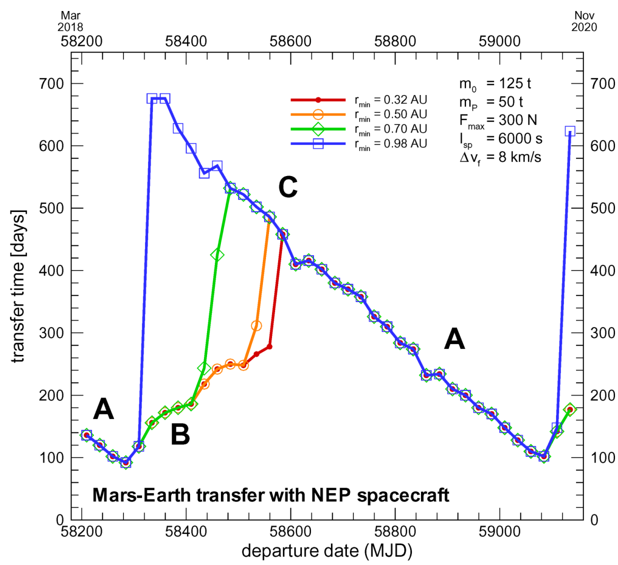

Figure 11: Transfer time for Earth return in dependency of the departure date at Mars and the minimum tolerable solar distance $r_{\min}$

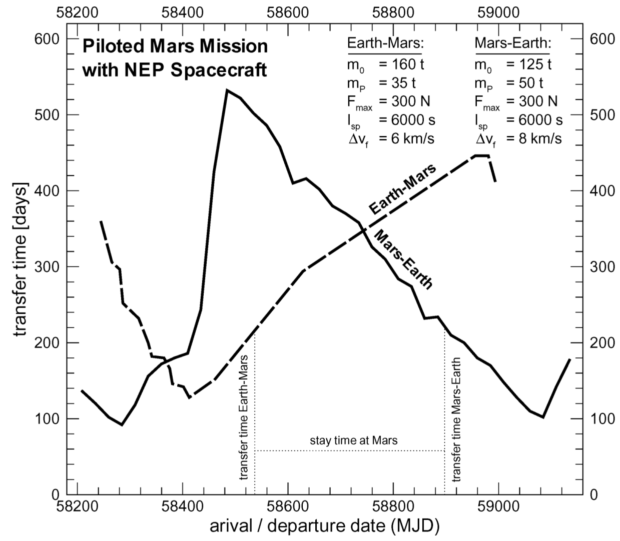

Figure 12: Transfer times against arrival/departure date at Mars ($r_{\min} = 0.7\,\mathrm{AU}$)

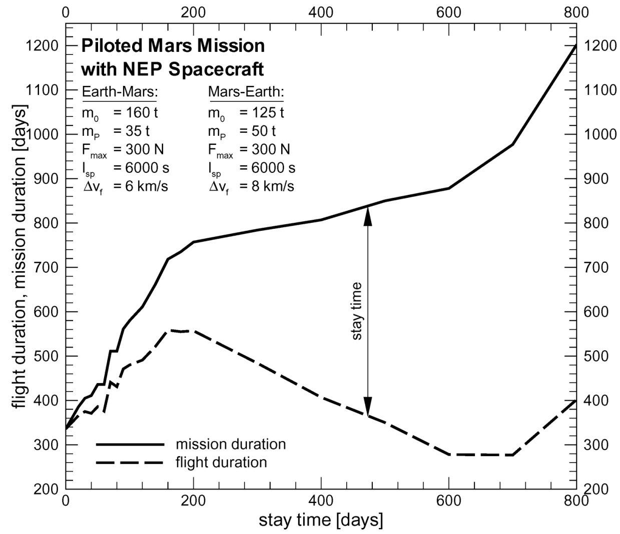

Figure 13: Mission duration against stay time at Mars ($r_{\min} = 0.7\,\mathrm{AU}$)

Earth-Mars leg with a short Mars-Earth leg (a type A-A trajectory pair) is not possible with the given propulsion system. On the basis of this diagram, different options for a piloted mission can be discussed. The horizontal bar of the "H" gives the stay time and the two vertical bars give the combined transfer time, so that the size of the "H" defines the total mission duration. For each stay time-value a minimal flight time exists, which can be plotted against the stay time, as it is done in figure 13. As this diagram shows – using the given spacecraft and propulsion system parameters – a stay time of three month can be realized within a total mission duration of 561 days (1.54 years). For a total mission duration of 2 years, the stay time at Mars can be extended to about 140 days (4.7 months). The diagram shows also that – using the given propulsion system – type A-A transfers with short total flight durations are only possible for long stay times at Mars of about 600 days.

6 SUMMARY AND CONCLUSIONS

Within this paper, low-thrust trajectory optimization was attacked from the perspective of machine learning. Inspired by natural archetypes, a smart global method for spacecraft trajectory optimization was proposed that fuses artificial neural networks and evolutionary algorithms to evolutionary neurocontrollers. This method was termed InTrance, which stands for "**In**telligent **Tra**jectory optimization using **n**euro**c**ontroller **e**volution". From the perspective of machine learning, a trajectory is regarded as the result of a steering strategy that manipulates the spacecraft's thrust vector according to the actual state of the spacecraft and the target body. An artificial neural network is used as a so-called neurocontroller to implement such a spacecraft steering strategy. This way, the trajectory is defined by the internal network parameters of the neurocontroller. An evolutionary algorithm is used for finding the optimal network parameters. The trajectory optimization problem is solved, if the parameter set that generates the optimal trajectory is found. Using an evolutionary algorithm for the optimization of the neurocontroller, this algorithm may be additionally used for finding optimal initial conditions.

Within this paper, InTrance was applied to an interplanetary low-thrust trajectory optimization problem, a rendezvous with a near-Earth asteroid, for which a reference trajectory was found in the literature. The recalculation of the reference problem revealed that the trajectory, which has been generated using a local trajectory optimization method, is quite far from the global optimum. Using InTrance, the transfer time was reduced by 74%. Since evolutionary neurocontrollers explore the trajectory search space more exhaustively than a human expert can do by using traditional optimal control methods, they are able to find spacecraft steering strategies that generate better trajectories, which are closer to the global optimum. The obtained InTrance-trajectories are sufficiently accurate with respect to the terminal constraint. If a more accurate solution is required, the InTrance-solution might be used as an initial guess for some local trajectory optimization method. Unlike the traditional methods, InTrance runs without an initial guess and without the permanent attendance of an expert in astrodynamics and optimal control theory. Within this paper, the use of evolutionary neurocontrol was also demonstrated for the analysis of a piloted Mars mission using a spacecraft with a nuclear electric propulsion system. 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] J.W. Hartmann. Low-thrust trajectory optimization using stochastic optimization techniques. Master thesis, University of Illinois at Urbana-Champaign, 1996.

[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 neurocontrol 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] B. Dachwald. Verwendung eines neuronalen Reglers und evolutionärer Algorithmen zur Berechnung optimaler interplanetarer Sonnenseglerbahnen. Stuttgart, Germany, September 2002. Deutscher Luft- und Raumfahrtkongress 2002. DGLR-JT2002-089 (in German).

[13] B. Dachwald. Optimization of interplanetary solar sailcraft trajectories using evolutionary neurocontrol. *Journal of Guidance, Control, and Dynamics*, 27(1):66–72.

[14] E.K. Jessberger, W. Seboldt, K.-H. Glassmeier, G. Neukum, M. Pätzold, G. Arnold, H.-U. Auster, D. deNiem, 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.

[15] 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.

[16] T. D. Schmidt, B. Dachwald, W. Seboldt, and M. Auweter-Kurtz. Flight opportunities from Mars to Earth for piloted missions using continuous thrust propulsion. Huntsville, USA, July 2003. 39th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit. AIAA-2003-4573.

[17] T. D. Schmidt, W. Seboldt, and M. Auweter-Kurtz. Propulsion options for manned Mars missions. Versailles, France, May 2002. 6th International Symposium on Propulsion for Space Transportation of the XXIst Century.

[18] L. E. George and L. D. Kos. Interplanetary mission design handbook: Earth to Mars opportunities and Mars to Earth return opportunities 2009–2024. Technical Report NASA/TM–1998-208533, NASA, 1998.