AIAA/AAS Astrodynamics Specialist Conference and Exhibit
5-8 August 2002, Monterey, California

AIAA 2002-4989

OPTIMIZATION OF INTERPLANETARY RENDEZVOUS TRAJECTORIES FOR SOLAR SAILCRAFT USING A NEUROCONTROLLER

Bernd Dachwald * and Wolfgang Seboldt †
German Aerospace Center (DLR), Cologne, Germany

As for all low-thrust spacecraft, finding optimal solar sailcraft trajectories is a difficult and time-consuming task that involves a lot of experience and expert knowledge, since the convergence behavior of 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 converges to an "optimal trajectory", this trajectory is typically close to the initial guess that is rarely close to the global optimum. This paper demonstrates, that artificial neural networks in combination with evolutionary algorithms can be applied successfully for optimal solar sailcraft steering. Since these 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 steering strategies that generate better trajectories, which are closer to the global optimum. Results are presented for a Near Earth Asteroid rendezvous mission and for a Mercury rendezvous mission.

INTRODUCTION

Traditionally, solar sailcraft trajectories are optimized by the application of numerical optimal control methods that are based on the calculus of variations. The convergence behavior of these optimizers depends strongly on an adequate initial guess, which is needed prior to optimization. Therefore, depending on the difficulty and complexity of the problem, finding an optimal solar sailcraft trajectory usually turns into a time-consuming task that involves a lot of experience and expert knowledge. Even if convergence is achieved by the optimizer, the "optimal trajectory" is typically close to the initial guess that is usually far from the (unknown) global optimum.

Using artificial neural networks (ANNs) in combination with evolutionary algorithms (EAs) as evolutionary neurocontrollers (ENCs), this paper presents a novel method for solar sailcraft trajectory optimization, that does not depend on an initial guess and runs without the involvement of a trajectory expert. Although we have applied this method only to find optimal solar sailcraft trajectories for various interplanetary rendezvous problems, it is not limited to this problem class but can be adapted for other trajectory optimization problems (e.g. for different propulsion systems, for planetocentric motion, for fly-by trajectory optimization etc.).

*Research Engineer, Mission Architecture and Advanced Technologies Section, Institute of Space Sensor Technology and Planetary Exploration, Phone: +49-2203-601 3001, Fax: +49-2203-601 4655, E-Mail: bernd.dachwald@dlr.de

†Section Head, Mission Architecture and Advanced Technologies Section, Institute of Space Sensor Technology and Planetary Exploration, Phone: +49-2203-601 3028, Fax: +49-2203-601 4655, E-Mail: wolfgang.seboldt@dlr.de

SIMULATION MODEL AND PROBLEM STATEMENT

The magnitude and direction of the solar radiation pressure (SRP) force $\mathbf{F}_{\mathrm{SRP}}$ acting on a flat and perfectly reflecting solar sail (ideal sail) due to the momentum transfer from the solar photons is completely characterized by the sun-sail distance r and the sail attitude, which is generally expressed by the sail normal vector $\mathbf{n}$, whose direction is – according to Fig. 1 – usually described by the sail clock angle α and the sail cone angle β.

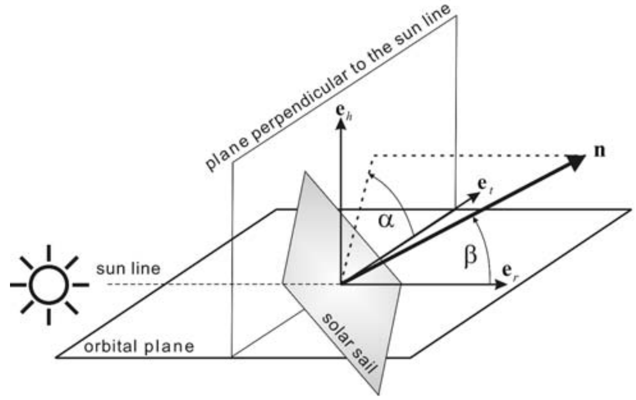

Fig. 1 Definition of the sail clock angle α and the sail cone angle β

For an ideal sail of area A, the SRP force that is acting on the sail,

$$\mathbf{F}_{\mathrm{SRP}} = 2P_0 \left(\frac{r_0}{r}\right)^2 A \cos^2\beta\, \mathbf{n}, \tag{1}$$

is always along $\mathbf{n}$ (where $P_0 \doteq 4.653\,\mu\mathrm{N/m^2}$ denotes the SRP at $r_0 = 1\,\mathrm{AU}$).

Since a *real* solar sail is neither flat nor a perfect reflector, a thorough trajectory analysis must take into account the optical properties of the real sail, which are also time-varying due to the erosive effects of the space environment. However, for preliminary trajectory analysis – as done in this paper – an ideal sail may be assumed and some further simplifications may be made:

- The solar sailcraft is moving under the sole influence of solar gravitation and radiation. The sun is a point mass and a point light source. Other celestial bodies are neglected. Also neglected are disturbing forces, which are much smaller than the sun's gravitational force and the SRP force (e.g. by the solar wind and the aberration of light).
- The sail attitude can be changed instantaneously.

The orbital dynamics of solar sailcraft is in many respects similar to the orbital dynamics of other spacecraft, where a small continuous thrust is applied to modify the spacecraft's orbit over an extended period of time. However, other continuous thrust spacecraft may orient its thrust vector in any desired direction and vary its thrust level within a wide range, whereas the thrust vector of solar sailcraft is constrained by equation (1) to lie on the surface of the "$\cos^2\beta$-bubble" that is always directed away from the sun (Fig. 2). Nevertheless, by controlling the sail orientation relative to the sun, solar sailcraft can *gain* orbital angular momentum (if $\mathbf{F}_{\text{SRP}} \cdot \mathbf{e}_t > 0$) and spiral outwards – away from the sun – or *lose* orbital angular momentum (if $\mathbf{F}_{\text{SRP}} \cdot \mathbf{e}_t < 0$) and spiral inwards – towards the sun.

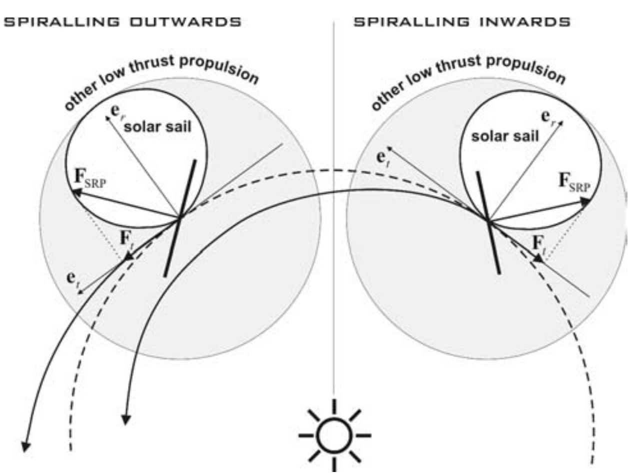

Fig. 2 Spiralling inwards and outwards

This paper deals with the problem of finding the optimal interplanetary solar sailcraft rendezvous trajectory to a given target body, that is in terms of optimal control theory:[1] find a sail normal vector (control vector) history $\mathbf{n}[t]$ (for $t_0 \le t \le t_f$) which forces the state $\mathbf{x}(t) = (\mathbf{r}(t), \dot{\mathbf{r}}(t))$ of the solar sailcraft from its initial value $\mathbf{x}(t_0)$ to the state $\mathbf{x}_T(t)$ of the target body (thus obeying the terminal constraint $\mathbf{x}(t_f) = \mathbf{x}_T(t_f)$) and, at the same time, minimizes the cost function $J = \int_{t_0}^{t_f} dt = t_f - t_0$.* The resulting state history $\mathbf{x}^\star[t]$ is the optimal trajectory for the given rendezvous problem. So the trajectory optimization problem is actually a problem of finding the optimal control vector history $\mathbf{n}^\star[t]$.

TRADITIONAL TRAJECTORY OPTIMIZATION

Traditionally, solar sailcraft 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. Prior to optimization, the NLP methods and the gradient methods require an initial guess for the control vector history $\mathbf{n}[t]$, whereas the neighboring extremal methods require an initial guess for the starting adjoint vector of Lagrange multipliers $\boldsymbol{\lambda}(t_0)$ (costate vector).[1] The convergence behavior of all of those methods is very sensitive to the respective initial guess,[2,3] so that trajectory optimization becomes sometimes "more art than science".[2] If convergence is achieved, a local optimum is found, which is typically close to the initial guess and far from the global optimum. If convergence could not be achieved, a new initial guess has to be conceived. Since similar initial guesses often produce very dissimilar optimization results, the initial guess can not be improved iteratively and the search for a good trajectory can turn into a very time-consuming task.

ARTIFICIAL NEURAL NETWORKS

Being inspired by the processing of information in animal nervous systems, artificial neural networks (ANNs) are a computability paradigm that is alternative to conventional serial digital computers. ANNs are massively parallel, analog, fault tolerant and adaptive.[4,5,6] 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, e.g. 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.[7]

Since the neurons can be connected in many ways, ANNs exist in a wide variety. However, within our research only feedforward ANNs have been considered. Typically, feedforward ANNs have a layered

*Since solar sailcraft do not consume any propellant, only the transfer time is minimized and – unlike for other spacecraft – the final mass $m(t_f)$ is not a part of the cost function.

topology, where the neurons are organized hierarchically in a number ℓ of so-called neuron layers. The first neuron layer is called the input layer and has n_1 input neurons, which receive the network's input. The last neuron layer is called the output layer and has n_ℓ output neurons, which provide the network's output. All intermediate layers/neurons are called hidden layers/neurons.* A layered feedforward ANN can be regarded as a continuous† parameterized function (called network function)

$$\Phi_{\boldsymbol{\pi}} : \mathcal{X} \subseteq \mathbb{R}^{n_1} \rightarrow \mathcal{Y} \subset \mathbb{R}^{n_\ell}$$

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

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 $\Phi^\star := \Phi_{\boldsymbol{\pi}^\star}$ by adapting the internal parameters in a way that the network error is minimized. For this kind of learning problems, a variety of learning algorithms have been developed to determine the optimal network parameters $\boldsymbol{\pi}^\star$, the backpropagation algorithm – a gradient-based method – being the most widely known.[6]

REINFORCEMENT LEARNING AND NEUROCONTROL

Learning algorithms for ANNs that rely on a training set fail, when the correct output for a given input is not known. This is the case for so-called reinforcement learning (RL) 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 scalar evaluation (reinforcement) of the agent's behavior.[8,9] The optimal behavior of the agent is defined as the one that maximizes the sum of positive reinforcements and minimizes the sum of negative reinforcements over time. Delayed reinforcement learning (DRL) problems commonly arise in the optimal control of dynamical systems.[8]

Operating within so-called neurocontrollers (NCs), ANNs have been successfully applied to this class of learning problems.[7] Neurocontrol approaches to solve RL problems can be divided into two categories, *indirect* ones and *direct* ones.[10,11] The direct neurocontrol approach, which we used within our research, employs a single ANN, which is called the action model. The action model controls the dynamical system by providing a control vector $\mathbf{Y}(t) \in \mathcal{Y}$ from some input vector $\mathbf{X}(t) \in \mathcal{X}$ that contains the information that is relevant to perform this task (system state, environmental state etc.).‡ Henceforth, to keep things simple, we will use the term 'NC' for the ANN that is precisely 'the action model of the NC'.

NCs can also be applied to the optimal control problem of solar sailcraft trajectory optimization, which is a DRL problem: if a NC is used to direct the solar sailcraft's trajectory by controlling the sail attitude $\mathbf{n}(t)$, then this NC receives a single reinforcement for its control vector history $\mathbf{n}[t]$ (i.e. for its behavior) at the final time t_f, when the trajectory can be evaluated. It is to note, that the NC's behavior is completely characterized by its network function $\Phi_{\boldsymbol{\pi}}$ (that is again completely characterized by its parameter vector $\boldsymbol{\pi}$). The next section will address a learning method for DRL problems that may be used for determining the NC's *optimal* network function $\Phi_{\boldsymbol{\pi}^\star}$

EVOLUTIONARY ALGORITHMS AND EVOLUTIONARY NEUROCONTROL

Evolutionary algorithms (EAs, sometimes also called genetic algorithms, GAs) 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[11,13,14,15] as well as for a wide range of other optimization problems. Therefore, they are also expected to be an efficient method for finding the NC's optimal network function.

EAs use a vocabulary borrowed from biology. The key element of an EA is a population that comprises numerous individuals A_k ($k \in \{1, ..., q\}$), which are potential solutions to the given optimization problem. All individuals of the (initially randomly created) population are evaluated according to a fitness function F (analogous to a cost function) for their suitability to solve the problem. Their allocated fitness value $F(A_k)$ is crucial for their probability to reproduce and to create offspring into a newly created population, since 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 sur-

*Fig. 7 shows an example for a layered feedforward ANN with three input neurons, one hidden layer with two hidden neurons and one output neuron.

†if a sigmoid activation function for the neurons is used, see mathematical appendix

‡The more commonly used indirect neurocontrol approach, which we did not use within our research, employs additionally a system model and a second ANN, which is called evaluation model. Based on the system model, the evaluation model provides a prediction of the evaluation of the action that is considered by the action model.[7,10,12]

vival. After some reproduction cycles, the population converges against a single solution $A^\star$, which is in the best case the globally optimal solution to the given problem.

The application of an EA to search for the NC's optimal network function makes use of the fact, that a set of NC parameters $\pi_1, \ldots, \pi_m$ can be mapped onto a real valued string of length m, which provides an equivalent description of the NC's network function. By searching for the fittest individual (string) $A^\star$, the EA searches for the NC's optimal network function $\Phi^\star$. Such NCs, which employ an EA for learning, are called evolutionary neurocontrollers (ENCs).

Before applying a NC to the optimization of solar sailcraft trajectories, the NC's input set $\mathcal{X}$ and output set $\mathcal{Y}$ have to be defined adequately, i.e. the questions "what does the NC get as input?" and "how do we interpret the NC's output?" or rather "what is the NC expected to do?" have to be answered. This is crucial for the NC's performance on the problem, since we can not expect the NC to "make something out of nothing". Before providing an answer to those questions, we have to address strategies for solar sailcraft steering.

SOLAR SAILCRAFT STEERING

A pure local steering law (PLSLs) may be defined as a steering law that changes (increases, decreases or adjusts to some given reference value) some actual osculating orbital element of solar sailcraft with a maximum rate. For obtaining PLSLs, we may use Lagrange's planetary equations in Gauss' form,[16] since these equations describe the rate of change of a body's osculating elements due to some small (disturbing and/or propulsive) acceleration. If we have defined n PLSLs, each PLSL $i \in \{1, \ldots, n\}$ gives a direction, along which the SRP force has to be maximized. This direction may be expressed by a unit vector $\mathbf{f}_i$, called the optimal thrust unit vector. From $\mathbf{f}_i$ the related sail normal vector $\mathbf{n}_{\mathbf{f}_i}$ (and thus the sail clock angle $\alpha_{\mathbf{f}_i}$ and the sail cone angle $\beta_{\mathbf{f}_i}$) can be calculated.

To change more than one orbital element at the same time, the n PLSLs can be mixed. For that reason, a vector $\mathbf{c} \in [0,1]^n$ of weight factors (called steering law weight vector) may be defined in a way, that each $\mathbf{c}$ defines a mixed local steering law (MLSL) by giving the (mixed) optimal thrust unit vector

$$\mathbf{f} = \frac{\sum\limits_{i=1}^{n} c_i \mathbf{f}_i}{\left| \sum\limits_{i=1}^{n} c_i \mathbf{f}_i \right|} \tag{2}$$

from the (pure) optimal thrust unit vectors $\mathbf{f}_i$. Again, the related sail normal vector $\mathbf{n}_\mathbf{f}$, sail clock angle $\alpha_\mathbf{f}$ and sail cone angle $\beta_\mathbf{f}$ can be calculated from $\mathbf{f}$.

A sailcraft steering strategy may now be defined as some function $S : \mathcal{X} \to [0,1]^n$, that gives the actual steering law weight vector $\mathbf{c}(t) \in [0,1]^n$ from some vector of input variables $\mathbf{X}(t) \in \mathcal{X}$. The trajectory optimization problem may then be reformulated: find a sail steering strategy $S : \mathcal{X} \to [0,1]^n$ (for $t_0 \leq t \leq t_f$) which forces the state $\mathbf{x}(t)$ of the solar sailcraft from its initial value $\mathbf{x}(t_0)$ to the state $\mathbf{x}_T(t)$ of the target body (thus obeying the terminal constraint $\mathbf{x}(t_f) = \mathbf{x}_T(t_f)$) and, at the same time, minimizes the cost function $J = t_f - t_0$. The resulting steering strategy $S^\star$ is the optimal steering strategy for the given rendezvous problem. So the trajectory optimization problem is actually a problem of finding the optimal steering strategy $S^\star$.

To use pure and mixed local steering laws is just *one* method for obtaining steering strategies. Since those steering strategies have implicit knowledge about how to change the orbital elements in an optimal way, they can be considered as *indirect* steering strategies. However, the implementation of steering strategies is also possible without the use of local steering laws, e.g. by providing the optimal thrust unit vector $\mathbf{f}$ directly. Since such steering strategies do *not* have implicit knowledge about what orbital elements are and how they can be changed, they can be considered as *direct* steering strategies.

SOLAR SAILCRAFT TRAJECTORY OPTIMIZATION USING EVOLUTIONARY NEUROCONTROL

For the implementation of solar sailcraft steering strategies, as defined above, an ENC may be used. In this case the NC's parameter vector $\boldsymbol{\pi}$ defines a steering strategy $S_{\boldsymbol{\pi}} : \mathcal{X} \to \mathcal{Y}$ and the EA is used to determine the optimal NC parameter vector $\boldsymbol{\pi}^\star$ that results in the optimal steering strategy $S^\star := S_{\boldsymbol{\pi}^\star}$ (that again results in the optimal sail normal vector history $\mathbf{n}^\star[t]$ that again results in the optimal solar sailcraft trajectory $\mathbf{x}^\star[t]$).

We have considered two different output sets $\mathcal{Y}$, one representing an indirect steering strategy and one representing a direct one:

- the NC provides the steering law weight vector $\mathbf{c}$, $S : \mathcal{X} \to \{\,\mathbf{c}\}$ (indirect steering strategy)
- the NC provides the optimal thrust unit vector $\mathbf{f}$, along which the SRP force has to be maximized,* $S : \mathcal{X} \to \{\mathbf{f}\}$ (direct steering strategy)

Since it is reasonable to assume for a robust steering strategy, that the actual optimal SRP force direction $\mathbf{n}(t)$ depends *at any time* t on the actual state of the solar sailcraft $\mathbf{x}(t)$ and the target body $\mathbf{x}_T(t)$, we have used $\mathcal{X} = \{(\mathbf{x}, \mathbf{x}_T)\}$ for the domain of solar sailcraft steering strategies, thus $S : \{(\mathbf{x}, \mathbf{x}_T)\} \to \mathcal{Y}$.†

*The optimal thrust unit vector $\mathbf{f}$ can be calculated from the NC output $\mathbf{Y} \in (0,1)^3$ via $\mathbf{f} = (2\mathbf{Y} - \mathbf{1})/|2\mathbf{Y} - \mathbf{1}|$.

†It is to note, that a steering strategy that is defined in this way, does *not* depend *explicitly* on time.

Now the final picture of solar sailcraft steering using an ENC can be drawn (Fig. 3): To find the optimal trajectory, the ENC method is running in two loops. Within the (inner) trajectory integration loop, a NC steers the solar sailcraft to fly the trajectory that is completely defined by the NC's parameters, which are set by the EA in the (outer) NC optimization loop.

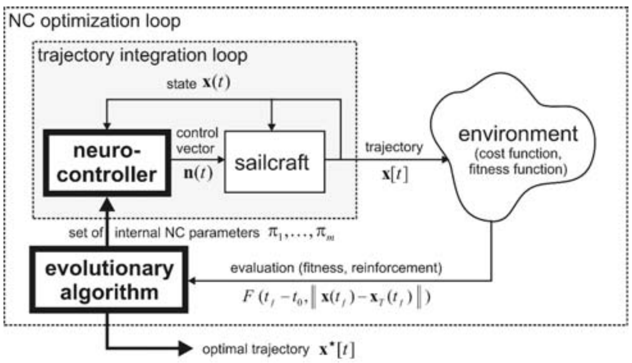

Fig. 3 Trajectory optimization using an evolutionary neurocontroller

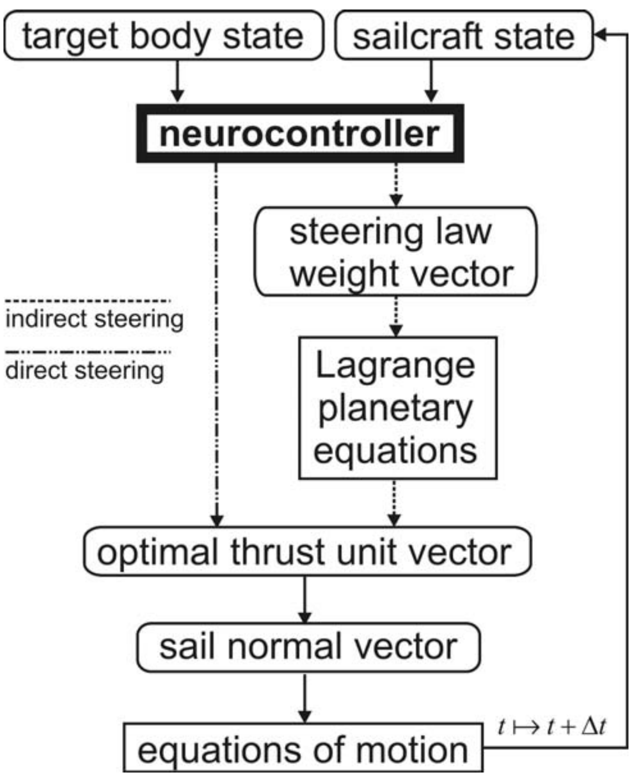

Fig. 4 Solar sailcraft steering using a neurocontroller (trajectory integration loop)

For solar sailcraft steering, the NC takes the actual state of the solar sailcraft $\mathbf{x}(t)$ and that of the target body $\mathbf{x}_T(t)$ as input values and maps from them onto some output values (Fig. 4). If the NC implements an indirect steering strategy, its n output values are interpreted as the required steering law weight vector $\mathbf{c}(t)$. Using this steering law weight vector, the optimal thrust unit vector $\mathbf{f}(t)$ can be calculated from Langrange's planetary equations. If the NC implements a direct steering strategy, its three output values are directly interpreted as the the required optimal thrust unit vector $\mathbf{f}(t)$. So we can calculate the required sail normal vector $\mathbf{n_f}(t)$ (or $\alpha_\mathbf{f}(t)$ and $\beta_\mathbf{f}(t)$) in both cases. Having done this, we insert the sail normal vector into the equations of motion and integrate over a time period Δt to get the solar sailcraft state $\mathbf{x}(t + \Delta t)$. This state is fed back into the NC. The trajectory integration loop stops, when the terminal constraint of the rendezvous problem is practically satisfied ($\|\mathbf{x}(t) - \mathbf{x}_T(t)\| \leq \boldsymbol{\epsilon}$) or if some time limit is reached. Then, in the NC optimization loop, the NC's parameter vector (i.e. its trajectory) is rated by the EA's fitness function. Since we can not expect the NC to generate a trajectory that strictly obeys the final constraint for rendezvous, the constraint has to be included into the fitness function, so that $F = F(t_f - t_0, \|\mathbf{x}(t_f) - \mathbf{x}_T(t_f)\|)$. As mentioned above, this fitness is crucial for the probability to reproduce and to create offspring. Under this selection pressure, the ENC generates more and more suitable trajectories. The ENC finally converges against a single steering strategy, which gives in the best case a near-globally* optimal trajectory for the rendezvous problem.

RESULTS

The method described above was applied to a variety of solar sailcraft rendezvous problems. Here, the results for two mission examples shall be presented, for which expert-generated trajectories are available.[3,17,18,19,20]

The first mission example (Fig. 5) is a Near Earth Asteroid (NEA) rendezvous with $1996FG_3$†, a mission that will not be too demanding for moderate performance sailcraft of the first generation (characteristic acceleration = maximum acceleration at Earth distance = $a_c = 0.14\,\mathrm{mm/s^2}$). The second mission example (Fig. 6) is a Mercury rendezvous with a more advanced solar sailcraft ($a_c = 0.55\,\mathrm{mm/s^2}$). Both mission examples reveal, that the trajectories found by traditional optimization are far from the global optimum. The ENC trajectory for the $1996FG_3$ rendezvous mission is 52 days (3%) faster than the conventionally generated trajectory[19,20] and reduces the C_3 requirement from $C_3 = 4\,\mathrm{km^2/s^2}$ to $C_3 = 0\,\mathrm{km^2/s^2}$ (at time of rendezvous the distance to the target body Δr is less than 52 000 km and the relative velocity Δv is less than 0.19 km/s). The ENC trajectory for the Mercury rendezvous mission is 94 days (14%) faster than the conventionally generated trajectory,[3,17,18] both with

* *near*-globally, since global optimality can rarely be proven except by complete enumeration, which is not feasible

† we have also investigated a sample return mission to $1996FG_3$, including a 140 kg lander and a 40 kg Earth return capsule ($a_c = 0.10\,\mathrm{mm/s^2}$)[21]

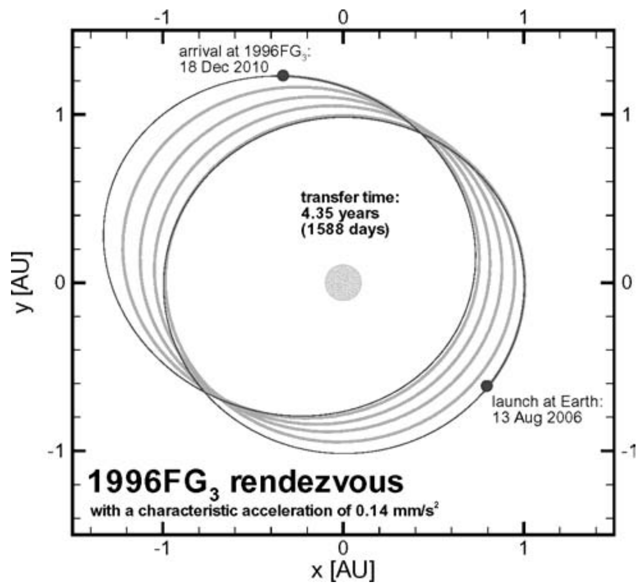

Fig. 5 1996FG$_3$ rendezvous trajectory

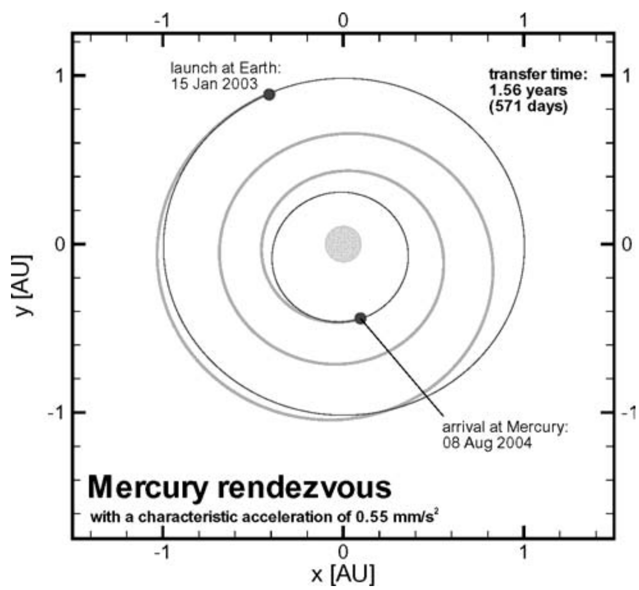

Fig. 6 Mercury rendezvous trajectory

$C_3 = 0\,\mathrm{km}^2/\mathrm{s}^2$ ($\Delta r < 44\,000\,\mathrm{km}$ and $\Delta v < 0.06\,\mathrm{km/s}$ at time of rendezvous). However, the ENC-generated trajectories are not optimal solutions in the strict sense, since the terminal constraint (the rendezvous condition) is not exactly met. To improve the accuracy of the trajectory further, the ENC trajectory can be taken as the initial guess for a direct numerical optimal control method such as NLP.

To perform trajectory optimization with an ENC, the following parameters have to be fixed:

- the NC's input set
- the NC's output set
- the NC's topology (number of hidden neuron layers and number of hidden neurons)
- some EA parameters (population size, mutation rate, selection scheme etc.)
- the EA's fitness function

We have investigated various combinations of those parameters. The performance of the ENC was found to be relatively robust with respect to different settings of most of the parameters. However, it depends strongly on the choice for the EA's fitness function. This 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. No final recommendation can yet be given for the most suitable set of steering strategies. For most problems, direct and indirect steering strategies produced similar results. However, there have been some problems, where direct or indirect steering was significantly superior. Regardless of the results, direct steering strategies are more elegant and have a broader applicability, since indirect strategies can not be used for trajectories that turn hyperbolic.

The NCs that we typically used had a single hidden layer with about 30 neurons. The maximum number of integration steps was usually set to values between 200 and 1 000, allowing the NC to change the solar sailcraft's attitude every $1 - 10$ days. Depending on the number of integration steps, the total computation time for one trajectory optimization run was in the order of one hour on a modern day (1.3 GHz) personal computer, during which the EA reproduced and tested about 10 000 trajectories.

CONCLUSIONS

The results shown above indicate clearly, that the novel method of using an evolutionary neurocontroller for solar sailcraft steering is a very promising approach for finding near-globally optimal trajectories. The obtained trajectories are fairly accurate with respect to the terminal constraint for rendezvous. If a more accurate trajectory is required, the evolutionary neurocontroller solution can be used as a proper initial guess for traditional trajectory optimization methods. However, before evolutionary neurocontrollers could be considered as a versatile and robust tool for generating near-globally optimal trajectories by someone without basic knowledge in astrodynamics, further research on convergence, stability and robust parameter settings should be done.

Evolutionary neurocontrol may be applied to a wide variety of low-thrust trajectory optimization problems, including different propulsion systems, fly-by trajectories and planetocentric trajectories. Future research will also focus on *multiple* rendezvous trajectories and the question, whether a *single* steering strategy exists, that yields near-globally optimal trajectories for *all* interplanetary rendezvous problems (for a given spacecraft).

MATHEMATICAL APPENDIX: LAYERED FEEDFORWARD ARTIFICIAL NEURAL NETWORKS

ANNs can be divided into *feedforward* ones and into *recurrent* ones, according to the connectivity of the neurons. An ANN is a feedforward one, if there exists a numbering method, which numbers all neurons in a way, that there is no connection from a neuron with a number i to a neuron with a number $j < i$. An ANN is a recurrent one, if such a numbering method does not exist.

Each neuron $i \in \mathcal{N}$ has a so-called activation function that maps from the neuron's input value(s) onto a single output value. The most commonly used activation function for feedforward networks is the sigmoid $s_\gamma : \mathbb{R} \to (0,1)$, defined by

$$s_\gamma(x) = \frac{1}{1+\mathrm{e}^{-\gamma x}}, \tag{3}$$

where the constant γ defines the slope of the function.

Typically, feedforward ANNs have a layered topology, where the set $\mathcal{N}$ of neurons is divided into ℓ subsets $\mathcal{N}_1, \ldots, \mathcal{N}_\ell$ (called neuron layers) in a way, that only connections from $\mathcal{N}_{k-1}$ go to $\mathcal{N}_k$ for all $k \in \{2, \ldots, \ell\}$. $\mathcal{N}_1$ is called the input layer and has n_1 input neurons, which receive the network's input $\mathbf{X} \in \mathbb{R}^{n_1}$. $\mathcal{N}_\ell$ is called the output layer and has n_ℓ output neurons, which provide the network's output $\mathbf{Y} \in (0,1)^{n_\ell}$. All other layers/neurons are called hidden layers/neurons (if $\ell > 2$). As an example, Fig. 7 shows an ANN with $\ell = 3$ layers, $n_1 = 3$ input neurons, one hidden layer with $n_2 = 2$ hidden neurons and $n_3 = 1$ output neuron (3-2-1-network).

Using the s_γ-activation function, a layered feedforward ANN can be described as a directed graph in which each node (neuron) i in a layer N_k ($k \in \{2, \ldots, \ell\}$) performs the function

$$y_i = \frac{1}{1+\mathrm{e}^{-\gamma(\sum_j w_{ij}x_j - \theta_i)}}, \tag{4}$$

where $y_i \in (0,1)$ is the output of neuron i, the $x_j \in (0,1)$ are the output values of the neurons j in the previous neuron layer $\mathcal{N}_{k-1}$, the $w_{ij} \in \mathbb{R}$ are the connection weights between the neurons j and neuron i, and $\theta_i \in \mathbb{R}$ is the so-called bias (or threshold) of neuron i. Each neuron i in the input layer $\mathcal{N}_1$ directly processes one component of the network's input values $X_i \in \mathbb{R}$ by performing the function

$$y_i = \frac{1}{1+\mathrm{e}^{-\gamma(X_i - \theta_i)}}. \tag{5}$$

Layered feedforward ANNs with the s_γ-activation function for the neurons can be regarded as a continuous parameterized function $\Phi_{\pi_1,\ldots,\pi_m} : \mathbb{R}^{n_1} \to (0,1)^{n_\ell}$,

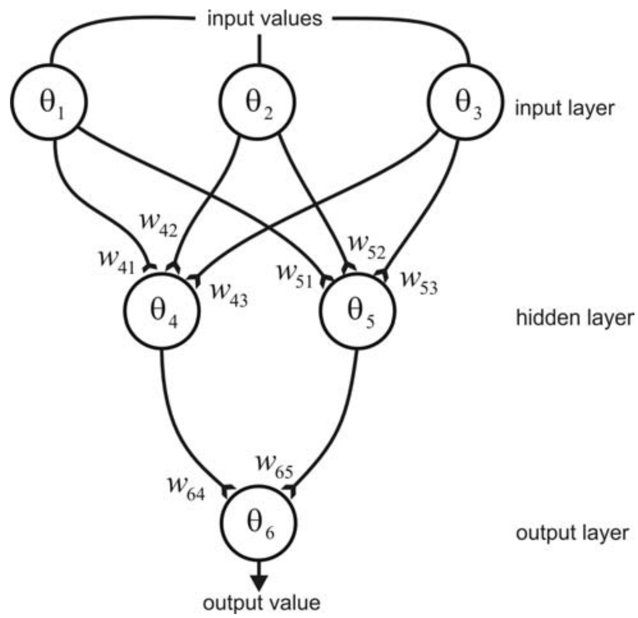

Fig. 7 Layered feedforward artificial neural network

called the network function, where the m parameters of the network function $\pi_1, \ldots, \pi_m$ are the connection weights (w_{ij}) and the biases (θ_i) of the neurons.*

Using Kolmogorov's theorem, it can be proven that any continuous function can be represented *exactly* by a finite network of computing units, though the general learning problem of determining the values for a given network's parameters is NP-complete.[6] In simple terms, this means that it is very improbable that an algorithm exists that is able to solve the problem in finite time (within the age of the universe) if the number of unknown variables gets large, though a guessed solution can be checked in finite time. However, in most practical cases no exact function representation is demanded but a finite approximation error is accepted for the network function, so that an approximate solution for the problem can be found in reasonable time.

REFERENCES

[1] R. F. Stengel. *Optimal Control and Estimation.* Dover Books on Mathematics. Dover Publications, Inc., New York, 1994.

[2] J. W. Hartmann. Low-thrust trajectory optimization using stochastic optimization techniques. Master thesis, University of Illinois at Urbana-Champaign, 1996.

[3] M. Leipold. *Solar Sail Mission Design.* Doctoral thesis, Lehrstuhl für Flugmechanik und Flugregelung; Technische Universität München, 1999. DLR-FB-2000-22.

[4] M. Caudill and C. Butler. *Naturally Intelligent Systems.* MIT Press, Cambridge, London, 1990.

[5] R. Rojas. Was können neuronale Netze? In R.-H. Schulz, editor, *Mathematische Aspekte der angewandten Informatik*, pages 55–88. Wissenschaftsverlag, Mannheim, 1994. (in German).

[6] R. Rojas. *Neural Networks. A Systematic Introduction.* Springer, Berlin, Heidelberg, New York, 1996.

*For the ANN in Fig. 7 we have $m = 14$, $\pi_1 = \theta_1$, $\pi_2 = \theta_2$, $\pi_3 = \theta_3$, $\pi_4 = w_{41}$, $\pi_5 = w_{42}$, …, $\pi_{14} = \theta_6$.

[7]D. C. Dracopoulos. *Evolutionary Learning Algorithms for Neural Adaptive Control*. Perspectives in Neural Computing. Springer, Berlin, Heidelberg, New York, 1997.

[8]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.

[9]R. S. Sutton and A. G. Barto. *Reinforcement Learning*. MIT Press, Cambridge, London, 1998.

[10]A. Likas and I. Lagaris. Training reinforcement neurocontrollers using the polytope algorithm. *Neural Processing Letters*, (9):119–127, 1999.

[11]D. Whitley, S. Dominic, R. Das, and C. W. Anderson. Genetic reinforcement learning for neurocontrol problems. *Machine Learning*, 13:259–284, 1993.

[12]A. G. Barto. Connectionist learning for control. In W. Miller, R. S. Sutton, and P. J. Werbos, editors, *Neural Networks for Control*, pages 5–58. MIT Press, Cambridge, London, 1990.

[13]Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer, Berlin, Heidelberg, New York, third, revised and extended edition, 1999.

[14]L. Tsinas and B. Dachwald. A combined neural and genetic learning algorithm. In *Proceedings of the First IEEE Conference on Evolutionary Computation*, pages 770–774, 1994.

[15]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.

[16]R. H. Battin. *An Introduction to the Mathematics and Methods of Astrodynamics*. AIAA Education Series. American Institute of Aeronautics and Astronautics, Reston, revised edition, 1999.

[17]M. Leipold, W. Seboldt, S. Lingner, E. Borg, A. Herrmann, A. Pabsch, O. Wagner, and J. Brückner. Mercury sun-synchronous polar orbiter with a solar sail. *Acta Astronautica*, 39(1-4):143–151, 1996.

[18]M. Leipold, E. Pfeiffer, P. Groepper, M. Eiden, W. Seboldt, L. Herbeck, and W. Unkenbold. Solar sail technology for advanced space science missions. Toulouse, 2001. 52nd International Astronautical Congress. IAF-01-S.6.10.

[19]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, 2000. 51st International Astronautical Congress. IAF-00-S.6.11.

[20]E. K. Jessberger and W. Seboldt et al. ENEAS – exploration of near-earth asteroids with a sailcraft. Proposal for a Small Satellite Mission within the Extraterrestric Sciences Program of DLR, August 2000.

[21]B. Dachwald, W. Seboldt, and B. Häusler. Performance requirements for near-term interplanetary solar sailcraft missions. Versailles, May 2002. 6th International Symposium on Propulsion for Space Transportation of the XXIst Century.