J.D. Biggs, H. Fournier
Neural-Network-based Optimal Attitude Control Using Four Impulsive Thrusters
Journal of Guidance Control and Dynamics, Vol. 43, N. 2, 2020, p. 299-309
doi:10.2514/1.G004226

Neural-Network-Based Optimal Attitude Control Using Four Impulsive Thrusters

James D. Biggs* and Hugo Fournier†
Politecnico di Milano

***Abstract*– This paper tackles the problem of optimal attitude control using a minimal number of attitude thrusters. Three possible control solutions to this problem are presented; (i) a logic-based controller that is simple to implement (ii) a projective control that aims to optimally replicate an ideal continuous control as closely as possible and (iii) an optimal neural predictive control (NPC) that minimizes the total impulse during a maneuver. The neural predictive control is based on a recurrent neural network using a nonlinear autoregressive exogenous configuration for state propagation in a finite-time horizon optimization. Typically, for continuous systems, a back-propagation algorithm for the receding horizon optimization is used, but this is not applicable to systems with discrete inputs. In this paper the neural predictive control is adapted to boolean input systems by employing a robust genetic algorithm to undertake the receding horizon optimization. An automatic selection of the parameters of the cost function is proposed which improves the performance of the neural predictive control and reduces the tuning to only one parameter. In addition, a multi-layer perceptron (MLP) is trained off-line with the optimal control data obtained, thus replacing the CPU-intensive cost function with a significantly less computationally expensive meta-model. The neural predictive control performance is compared to the proposed logic-based and projective control algorithms in simulation of a 12 U CubeSat and is shown to be the most efficient in terms of total impulse requirement at equal settling time and the least sensitive to the choice of parameters. The multi-layer perceptron control drastically reduces the online computational cost with performance approaching those of the neural predictive control.**

I. Introduction

Recent interest in CubeSat deep-space missions has created numerous and unique challenges related to the highly constrained nature of these spacecraft. One challenge related to attitude control in deep space, is in using thrusters for de-tumbling, momentum dumping and potentially for slew motions, while minimizing the mass and volume of the attitude control subsystem. This can be achieved by using optimal control algorithms that minimize fuel usage. In

*Associate Professor, Department of Aerospace Science and Technology, jamesdouglas.biggs@polimi.it
†Graduate student, Department of Aerospace Science and Technology, fournier.hugo@mail.polimi.it

addition, the number of thrusters used for attitude control can be minimized. For example, any control torque vector can be induced using only four thrusters (assuming no thruster saturation) [2]. In the case of four on-off thrusters a torque can be induced in any axis, but not simultaneously in all axes. This paper addresses the problem of designing attitude control algorithms that utilize only four reaction thrusters and that minimize fuel expenditure. The main contribution of this paper is a control based on a new neural network-based optimization for optimal attitude tracking, slew manoeuvres and de-tumbling. Although the approach in this paper is specialized to the case of four thrusters it can be applied to any controllable thruster configuration.

Some of the first control algorithms useful for "on-off" actuators [10, 11] were based on Lyapunov stability theory, where switching on or off of the thrusters were chosen to minimize the derivative of a Lyapunov function. These control algorithms are analogous to sliding mode controls, where the maximum torque in each axis is always used and the thrusters coupled, so that they are able to produce a torque in equal and opposite directions. However, these controls do not minimize a practically meaningful cost function of the propellant consumption and state error.

The majority of control algorithms for on-off attitude thrusters use a three phase procedure involving the design of a continuous "ideal control", a control allocation algorithm to map the ideal control to each thruster and then a pulsed width modulator (PWM) to convert the continuous signal to an on or off switching [3–9]. This three phase procedure aims to distribute a continuous "ideal" torque generated by a continuous feedback controller, to the thrusters, while minimizing the total impulse. This problem can be framed as a Linear Programming (LP) problem and solved using LP algorithms, such as the primal-dual interior point [12] and the simplex algorithm [13]. The NASA Ames Center has also adopted this approach to investigate using six and eight thrusters [14] for deep-space nano-satellite attitude control using cold-gas or micro-electric propulsion.

The linear programming based method for control allocation is a static method as it only optimizes the allocation based on the current control demand. Robust dynamic control allocation methods, in which the allocation also depends on the distribution in the previous sampling instant have been developed using quadratic programming [15]. However, these methods for thruster allocation are deemed relatively computationally expensive for on-board implementation [16]. In [16] the authors propose thruster allocation approaches based on Singular value decomposition (SVD) to reduce the computational expense of a minimum norm optimization. The neural-predictive control of this paper yields an optimal controller that avoids this three phase procedure and minimizes a cost-function of state-error and propellant use.

A neural-predictive control is proposed for optimal attitude tracking, slew maneuvers and de-tumbling with minimal fuel consumption using the minimum number of thrusters required for 3-axis controllability[1]. This neural network predictive control is adapted to systems with boolean inputs, that is, with on-off actuation. Furthermore, the data generated from this optimal controller is used to develop a meta-model of the optimal control problem, that can be implemented with much lower computational cost. Classical model-based predictive control applied to rototranslational dynamics of spacecraft [17][18] utilises linear optimal control theory [19] [20]. However, such controls are not efficient

when performing slew maneuvers or de-tumbling since these operations begin far from the equilibrium state and thus linearization does not yield a good approximation to the model for all time steps. Recent methods using Recurrent neural networks (RNN) [21] [22] have been used to replicate the nonlinear dynamics and can be fused with model-based predictive control to allow efficient time propagation of the state and dynamic back-propagation for performing receding horizon optimization. In particular, Nonlinear Auto-Regressive exogenous configuration (NARX) model are used where several delayed values of the states and inputs of the dynamic system are used as inputs of a neural network. In [23] the main engines of the Space Shuttle are modeled with a feed-forward NARX network, and in [24] the rototranslational dynamics of a flexible micro-satellite are model and controlled through two NARX networks. More recently model-based predictive control based on neural networks has been adapted to nonlinear systems with boolean control inputs [25]. Here RNN are used to replicate the nonlinear dynamics and to evaluate the future state and cost function. The cost function is then optimized using a genetic algorithm and the optimal thruster selection chosen. A simple adaptive tuning law of a weight of the cost function is proposed and shown to improve performance. Furthermore, a Multi-Layer Perceptron (MLP) is used to improve the computational efficiency by developing a meta-model of the optimal control problem. Moreover, an MLP is trained off-line before flight with optimal data obtained with the genetic algorithm-based predictive optimization. A similar approach to facilitate on-board optimization has recently been proposed by the European Space Agency (ESA) [26] for an optimal spacecraft landing problem: optimal control laws are generated for different initial states, and are used to train an MLP of 2 to 5 layers that outputs a control approaching the optimal response. To the authors knowledge this paper is the first to present a boolean control based on the association of an MLP with data generated by model predictive control using a recurrent neural network for the state propagation.

II. Problem statement

In this paper the control will be developed for the special case of four thrusters. The four thrusters are configured on a single face of the spacecraft with each delivering a constant force F with their vectors denoted $\boldsymbol{F_1}$, $\boldsymbol{F_2}$, $\boldsymbol{F_3}$, $\boldsymbol{F_4}$ as depicted in Figure 1. Each thruster is inclined by an angle α to the xz plane in different directions and as such can provide a torque in two directions of each axis at any given time, for example, by firing two thrusters simultaneously. In the configuration studied in this paper, the thrusters have the same thrust amplitude F and the same absolute angle α to the xz plane. Using four thrusters is considerably different to conventional thruster configurations as it does not allow a useful torque to be delivered in each axis simultaneously and it always induces a net force. In other words, conventional thruster configurations are symmetric and can always be operated in a way that induces zero net force. For a four thruster configuration a net force will always be induced and therefore minimizing the number of thruster firings has two practical objectives to not only minimize fuel but also the net force.

In the following, $\mathbb{1}$ and $\mathbb{0}$ stand for the identity matrix and the matrix with zero entries respectively, and for a given vectorial quantity $\boldsymbol{a} = [a_1; a_2; a_3]$, $\dot{a}$ designates its time derivative, $\boldsymbol{a}^T$ its transpose, and $\boldsymbol{a}^\times$ the skew symmetric matrix

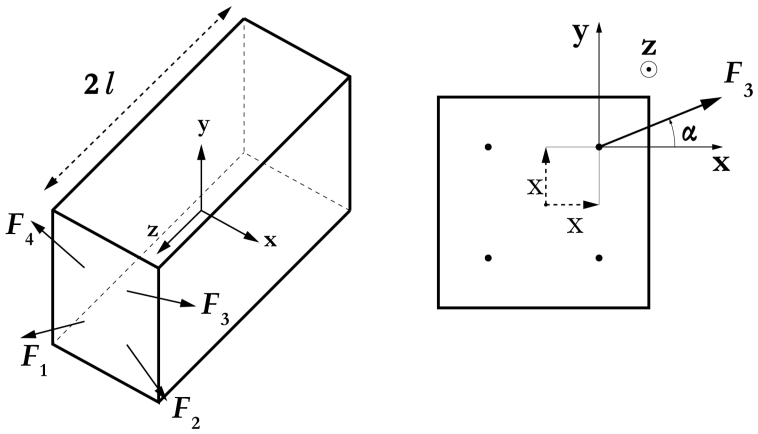

Fig. 1 Schematic representation of a spacecraft with 4 thrusters located on the same face

defined as:

$$\boldsymbol{a}^{\times} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}. \tag{1}$$

The attitude dynamics of a spacecraft are given by the equation of motion

$$I\dot{\boldsymbol{\omega}} = -\boldsymbol{\omega}^{\times} I\boldsymbol{\omega} + \boldsymbol{d} + \boldsymbol{u} \tag{2}$$

where $\boldsymbol{d}$ is an internal or external disturbance, $\boldsymbol{\omega}$ is the angular velocity expressed in the body frame, $\boldsymbol{u}$ is the control torque and I is the inertia matrix. The associated kinematic equations are:

$$\dot{A} = A\boldsymbol{\omega}^{\times} \tag{3}$$

where the spacecraft attitude matrix is $A \in SO(3)$ where $SO(3)$ is the 3D rotation group defined as:

$$SO(3) = \left\{ A \in \mathbb{R}^{3\times 3} \,:\, A^T A = \mathbb{1}_{3\times 3},\, det(A) = 1 \right\} \tag{4}$$

The attitude tracking control problem is to track a desired rotation A_d where $\dot{A}_d = A_d \boldsymbol{\omega}_d^{\times}$. The attitude error $A_e = A_d^T A$ is given by

$$\dot{A}_e = A_e \boldsymbol{\omega}_e^{\times} \tag{5}$$

where

$$\boldsymbol{\omega}_e = \boldsymbol{\omega} - A_e\boldsymbol{\omega}_d \tag{6}$$

where $\boldsymbol{\omega}_d$ is the desired angular velocity. The control objective is to develop a feedback control law $\boldsymbol{u}_{0/1}$ resulting in the torque:

$$\boldsymbol{u} = -T \cdot \boldsymbol{u}_{0/1} \tag{7}$$

where $\boldsymbol{u}_{0/1}$ is the thruster activation boolean vector, that is 1 when the thruster is activated, and 0 when it is turned off and

$$T = \begin{bmatrix} lF\sin\alpha & lF\sin\alpha & -lF\sin\alpha & -lF\sin\alpha \\ -lF\cos\alpha & lF\cos\alpha & lF\cos\alpha & -lF\cos\alpha \\ xF\sin\alpha - xF\cos\alpha & xF\cos\alpha - xF\sin\alpha & xF\sin\alpha - xF\cos\alpha & xF\cos\alpha - xF\sin\alpha \end{bmatrix} \tag{8}$$

where $\alpha \in (0, \frac{\pi}{4})$ is defined by the thrusters orientation, x and l are geometric lengths of the spacecraft with its center of mass assumed to be at the geometric center. The control objective is to then drive A and $\boldsymbol{\omega}$ to the desired trajectory defined by A_d and $\boldsymbol{\omega}_d$ in a fuel-efficient (preferably optimal) way. In total 16 combinations of thrusters activations exist, and the one in which all the thrusters are activated is excluded since it leads to a zero torque while consuming propellant.

The main contribution of the paper is the development of a receding horizon neural-predictive optimal control adapted to "on-off" thrusters that avoids the need for a three phase procedure. In addition, a neural network is trained on the optimal data generated by the neural predictive control to significantly reduce computational expense. The following section outlines some simple control laws that will be used as the benchmark controllers for comparison. These controllers are computationally efficient and avoid the need of pulsed-width modulation.

III. Control laws for non-symmetric thruster configurations

In this section two control laws are presented based on (i) a logic-based control law and (ii) a projection-based control, so called, because it projects an ideal continuous control onto the space of all possible torque profiles and selects the thruster combination that minimizes the norm of the error between the ideal torque and the possible torques.

The Lyapunov-control function is used to design a continuous "ideal" control that forms the basis of the logic-based and projection control laws. In this case we propose a classic tracking controller augmented to reject disturbances at each sampling period. This will reject both external and internal disturbances (such as a time-varying inertia matrix due to fuel usage, sloshing, rotating solar panels or even actuator faults). However, note that the ideal feedback control can be changed to one that is preferred or required by the control engineer. To develop the ideal control a Lyapunov

control-function is defined here as:

$$V = \frac{1}{2}\boldsymbol{\omega}_e^T I \boldsymbol{\omega}_e + k_2 tr(\mathbb{1} - A_e) \tag{9}$$

then differentiating this with respect to time yields

$$\dot{V} = \boldsymbol{\omega}_e^T I \dot{\boldsymbol{\omega}}_e - k_2 tr(\dot{A}_e) \tag{10}$$

and using the identity $tr(A_e \boldsymbol{\omega}_e^{\times}) = \boldsymbol{\omega}_e^T (A_e - A_e^T)^{\vee}$ yields:

$$\dot{V} = \boldsymbol{\omega}_e^T (I\boldsymbol{\omega} \times \boldsymbol{\omega} + \boldsymbol{u} + \boldsymbol{d} - I\boldsymbol{\alpha}_d + k_2 (A_e - A_e^T)^{\vee}) \tag{11}$$

where $\boldsymbol{\alpha}_d = (A_e \boldsymbol{\omega}_e^{\times} \boldsymbol{\omega}_d + A_e \dot{\boldsymbol{\omega}}_d)$. Defining $\hat{\boldsymbol{d}}$ to be an estimate of $\boldsymbol{d}$ then an "ideal" control that guarantees $\dot{V} \leq 0$ if the disturbances are perfectly estimated is:

$$\boldsymbol{u}^c = \boldsymbol{\omega} \times I\boldsymbol{\omega} - \hat{\boldsymbol{d}} + I\boldsymbol{\alpha}_d - k_1 I \boldsymbol{\omega}_e - k_2 (A_e - A_e^T)^{\vee}. \tag{12}$$

As $\dot{V} \leq 0$ it can be shown by an application of La Salles invariance Principle that the zero error state is locally asymptotically stable. The disturbance estimation can be obtained from an extended state observer (ESO) such as a linear [27, 28] or [29] nonlinear ESO. In practice the disturbance estimation error converges to a small bounded region of the zero error estimation. The control (12) can be augmented to include an adaptive parameter that yields asymptotic stability to a tunable bounded region of the desired state [28] or the inclusion of a sliding-mode component, to ensure asymptotic stability in the presence of an estimation error [27]. Convergence properties of ESO are detailed in [28, 29].

A. Simple Logic-based control

Since it is possible to create a torque at any instant in time in one axis and with the required sign, by activating the right combination of two thrusters simultaneously, a simple sliding mode control, with the sliding surface a function of the ideal control is proposed:

$$\boldsymbol{u}^{s.l.} = -M_{i_{max}} sgn(\boldsymbol{u}^c_{i_{max}}) \tag{13}$$

where the continuous control $\boldsymbol{u}^c$ is given by 12 and

$$i_{max} = \arg\max_i \left| \boldsymbol{u}_i^c \right| \tag{14}$$

where

$$\boldsymbol{M} = \begin{Bmatrix} 2lF\sin\alpha \\ 2lF\cos\alpha \\ -2xF\sin\alpha + 2xF\cos\alpha \end{Bmatrix} \tag{15}$$

is the vector of the absolute values of the torques that can be delivered in a single axis at a time by activating two thrusters simultaneously.

Note that the tuning of only one parameter is required in the general case because only the sign of the continuous control (12) is taken into account, k_1 can be arbitrarily set to 1 for example. For the application to de-tumbling in which $S = \omega$, the control (13) requires no tuning and is, therefore, extremely simple to implement. Note that this control can be applied to any thruster configuration that can induce a torque in each axis in equal and opposite directions. In this case the components of $\boldsymbol{M}$ are replaced with the maximum torque available in each axis.

B. Projection control

This control approach is based on a standard continuous control for the tracking of a desired trajectory which is projected onto one of the possible torques u_i^p. Here there are 16 possible combinations of thruster "on-off" states with both switching all four on or off resulting in zero torque. The resulting 15 possible torques can be stored on-board in a simple look up table. At each time-step one of a torque is chosen from the possible to minimize the cost function $\left\|\boldsymbol{u}^c - u_i^p\right\|$. In this case the ideal control parameters must be tuned to minimize the total impulse and the settling time.

The, so-called, projection control used can be stated as:

$$\boldsymbol{u}_{0/1}^p = \underset{\boldsymbol{u}_{0/1}\in\{0,1\}^{4\times1}}{arg\,min} \left(\left\|T\boldsymbol{u}_{0/1} - \boldsymbol{u}^c\right\|^2\right) \tag{16}$$

where and $\boldsymbol{u}^c$ is the continuous control defined in (12).

In the general case, this control requires the tuning of two parameters k_1 and k_2, which makes it more difficult to implement than the simple-logic control. For the application to the de-tumbling of a spacecraft, the tuning of one parameter is still required, in contrast to the simple-logic control. Note that this control is applicable to any thruster configuration T.

IV. Neural network based predictive attitude control

In this section, a neural-network based predictive control that efficiently propagates the state in a finite horizon optimization is presented. The cost function considered is chosen to minimize the total impulse of the system. The NPC

provides discrete torques based on an optimization over a receding time horizon that leads to propellant savings, with an affordable computational cost made possible by the off-line training of a Multi-Layers Perceptron (MLP) neural controller. It is performed in two steps: (i) an optimal control is developed, it uses a recurrent neural network to propagate the states forward in time, and to forecast the value of a cost function with a high accuracy. (ii) the optimal control of the first step is used to generate a set of optimal data, that is used for training an MLP off-line before flight. Once the MLP is trained it will only need to be forward propagated on-line, hence greatly reducing the computational cost of the predictive attitude controller.

In NPC (for continuous control systems) a back-propagation algorithm is typically used for the receding horizon optimization, however, this is not suitable for boolean input systems. This paper adapts a continuous neural-network-based predictive control to a nonlinear system with boolean inputs by implementing a genetic algorithm which does not require the assumption of continuity in the optimization procedure. In the following sub-section the optimal control development is described.

A. Finite horizon predictive control

A commonly used cost function for receding horizon control applications used is the Linear Quadratic Regulator (LQR) which has the following quadratic form:

$$L(t) = \frac{1}{\Delta t} \int_t^{t+\Delta t} \boldsymbol{z}_e^T K \boldsymbol{z}_e + \boldsymbol{u}^T R \boldsymbol{u} \, dt \tag{17}$$

where K and R are weight matrices. $\boldsymbol{z}_e$ is the state error, which includes the angular velocity error vector $\boldsymbol{\omega}_e$ and the rotation error whose expression depends on the used representation of the kinematics. The choice of the representation has a significant influence on the capability of the plant neural network to replicate the spacecraft kinematics. As will be discussed at the end of this section, the quaternions are preferred over the direct cosine matrix, Euler angles, and modified Rodrigues parameters. The state error is expressed as

$$\boldsymbol{z}_e = \begin{Bmatrix} \boldsymbol{\omega}_e \\ \bar{\boldsymbol{q}}_{\boldsymbol{e}} \\ 1 - q_e^0 \end{Bmatrix} \tag{18}$$

where the quaternions error is defined as:

$$\begin{Bmatrix} \bar{\boldsymbol{q}}_{\boldsymbol{e}} \\ q_e^0 \end{Bmatrix} = \begin{bmatrix} q_d^0 & -\bar{q_d}^3 & \bar{q_d}^2 & \bar{q_d}^1 \\ \bar{q_d}^3 & q_d^0 & -\bar{q_d}^1 & \bar{q_d}^2 \\ -\bar{q_d}^2 & \bar{q_d}^1 & q_d^0 & \bar{q_d}^3 \\ -\bar{q_d}^1 & -\bar{q_d}^2 & -\bar{q_d}^1 & q_d^0 \end{bmatrix} \cdot \begin{Bmatrix} \bar{\boldsymbol{q}} \\ q^0 \end{Bmatrix} \tag{19}$$

$\boldsymbol{q} = [\bar{\boldsymbol{q}}, q^0]^T$ and $\boldsymbol{q}_d = [\bar{\boldsymbol{q}_d}, q_d^0]^T$ are the quaternion and desired quaternion respectively. The optimization variable in Eq. 17 is the future control input $[u(t')]_{t' \in [t;t+\Delta t]}$. In order to efficiently and accurately predict the future state errors of the system, a recurrent neural network is used. The NPC minimizes a modified version of the cost function defined in Eq. 17, adapted to the particularity of using boolean inputs to the system, and the nonlinear dynamics and kinematics of the spacecraft are replicated by a neural network as explained in the following subsections.

B. Neural Networks

A neural network is a function that computes an output by forward-propagation through M layers of S^m neurons each, with the following expression:

$$\boldsymbol{a}^1 = \sigma^1(W^1\boldsymbol{p} + \boldsymbol{b}^1) \tag{20}$$

$$\boldsymbol{a}^{m+1} = \sigma^{m+1}(W^{m+1}\boldsymbol{a}^m + \boldsymbol{b}^{m+1}) \qquad \text{for } m = 2 \dots M-1 \tag{21}$$

where $\boldsymbol{p} \in \mathbb{R}^{S^0 \times 1}$ is the input, $\boldsymbol{a}^m \in \mathbb{R}^{S^m}$ is the output of layer m, $W^m \in \mathbb{R}^{S^m \times S^{m-1}}$ and $\boldsymbol{b}^m \in \mathbb{R}^{S^m \times 1}$ are the weights of layer m, σ^m is the activation function of layer m and S^m is the number of neurons of layer m where the 0 index stands for the input vector of the network. Figure 2 illustrates this equation.

In the hidden layers the activation functions are taken to be the log-sigmoid functions

$$\sigma(n) = \frac{1}{1+e^{-n}} \tag{22}$$

and identity function in the outer layer. A sufficient condition to the universal approximation theorem [30] [31] for neural networks is satisfied when the the activation functions are continuous and bounded. Other functions such as the arctangent and the error function can be equally used.

Training a neural network consists of providing it with a set of target inputs/output pairs and modifying its weights and biases, so that, for the same inputs, the outputs of the network get as close as possible to the target outputs. The ultimate goal is not to interpolate the points, but to be able to generalize to any new input belonging to the same range as

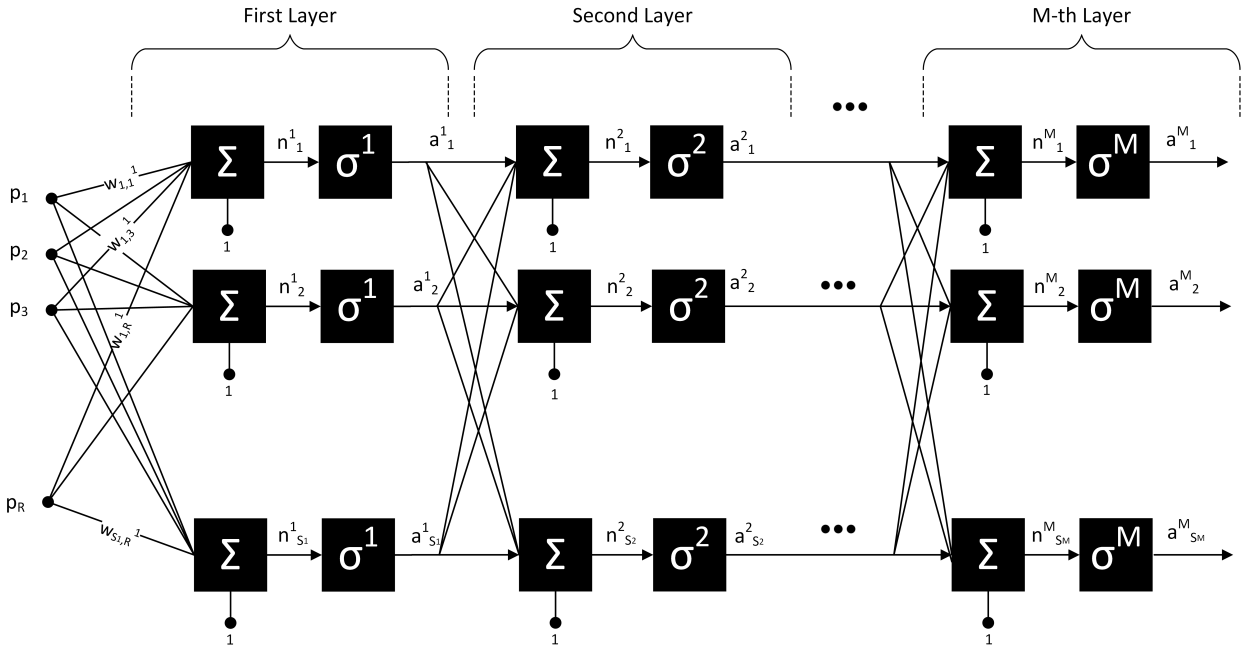

Fig. 2 Representation of a neural network

those from the training set.

For a given set $(p^q, t^q)_{q=1...Q}$ of Q training inputs/outputs, the following training cost function is used:

$$L^{tr}(\boldsymbol{\theta}) = \frac{1}{Q}\sum_{q=1}^{Q}(\boldsymbol{t}^q - \boldsymbol{a}^q(\boldsymbol{\theta}))^T(\boldsymbol{t}^q - \boldsymbol{a}^q(\boldsymbol{\theta})) \tag{23}$$

An optimization algorithm is implemented, in which the optimization variable is the vector $\boldsymbol{\theta} \in \mathbb{R}^{\sum_{m=0}^{M-1}(S^m+1)S^{m+1}}$ containing all the weights and biases of the Network,while $\boldsymbol{a} = \boldsymbol{a^M}$ is the network output. By writing

$$L^{tr}(\boldsymbol{\theta}) = \boldsymbol{\nu}(\boldsymbol{\theta})^T\boldsymbol{\nu}(\boldsymbol{\theta}) \tag{24}$$

where the vector $\boldsymbol{\nu} \in \mathbb{R}^{S^M Q\times 1}$ contains the Q error vectors, the cost function can be reduced by a Levenberg Marquardt algorithm as in [24], for which the update of the neural network's weights and biases at each cycle is obtained by

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \left[J^T(\boldsymbol{\theta}_k)J(\boldsymbol{\theta}_k) + \mu_k\mathbb{1}\right]^{-1}J^T(\boldsymbol{\theta}_k)\boldsymbol{\nu}(\boldsymbol{\theta}_k), \tag{25}$$

where the parameter μ_k is decreased if a weight update successfully decreases the cost function, and increased in the opposite case. When μ_k is low, the algorithm approaches the quadratic convergence velocity Newton Algorithm, while when it is high, the algorithm tends to a linear convergence velocity steepest gradient descent algorithm. The Jacobian

matrix

$$J = \frac{\partial \boldsymbol{\nu}}{\partial \boldsymbol{\theta}} \in \mathbb{R}^{(S^M Q)\times(\sum_{m=0}^{M-1}(S^m+1)S^{m+1})} \tag{26}$$

is obtained by the back-propagation algorithm:

$$\begin{aligned} \boldsymbol{s}^M &= \dot{\sigma}^M \\ \boldsymbol{s}^m &= \boldsymbol{s}^{m+1} W^{m+1} \dot{\sigma}^m \qquad \text{for } m = M-1 \ldots 1 \end{aligned} \tag{27}$$

where the sensitivities $\boldsymbol{s}^m$ are defined as:

$$\boldsymbol{s^m} = \frac{\partial \boldsymbol{a}^M}{\partial \boldsymbol{n}^m} \tag{28}$$

and $\boldsymbol{n}^m = W^m \boldsymbol{a}^m + \boldsymbol{b}^m$ is the net output of layer m.

$$\dot{\sigma}^m = \begin{bmatrix} \dot{\sigma}^m(n_1^m) & 0 & \ldots & 0 \\ 0 & \dot{\sigma}^m(n_2^m) & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \ldots & 0 & \dot{\sigma}^m(n_{S^m}^m) \end{bmatrix}$$

is the matrix composed of the derivatives of the activation function of layer m. In particular, a specific type of neural network with one hidden layer, a log-sigmoid activation function, and a linear activation function in the output layer is used based on a NARX model to replicate the dynamics of a system. The inputs are composed of the current torque applied to the system and its delayed values, and the delayed values of the output (the state). NARX networks allow to model continuous nonlinear systems with arbitrary accuracy. Intuitively, using different delayed values of the states allows the neural network to model several orders of time derivatives of the states in a similar way to a numerical integration.

C. Online neural predictive control (NPC)

In case of continuous torques, a NARX neural network can be trained offline to replicate the spacecraft dynamics and kinematics, and can be used online for computing the derivatives of the future states with respect to the current and future torques, hence allowing for a gradient-based or Jacobian based optimization of the control. However, in the case of thrusters the generated torques are discrete and therefore the assumption of continuous differentiability required for

these numerical optimization techniques is no longer valid. For this reason, different combinations of torques need to be successively computed in the optimization process. By discretizing the time, not all 15^N combinations of torques can be tried, where N is the receding horizon length, but a genetic algorithm can effectively find an optimal combination of thrusters $[\boldsymbol{u}_{0/1}(t')]_{t'\in[t+1;t+Ndt]}$ where t' is the discrete time in the optimization finite horizon.

An advantage of using a genetic algorithm is that no constraint exist on the form of the cost function, and shows good performance when the optimization variable can only have a discrete number of values. In this paper, we propose a cost function of the form

$$L(t) = \frac{1}{Ndt}\sum_{t'=t+1}^{t+Ndt} K_{quad}\left\|\boldsymbol{z_e}(t')\right\|_{2,K}^2 + K_\infty\left\|\boldsymbol{z_e}(t')\right\|_{\infty,K}^2 + \left\|\boldsymbol{u}_{0/1}(t')\right\|_{2,R}^2 \tag{29}$$

where K_{quad} and K_∞ are two scalar weights and

$$\|\boldsymbol{z_e}\|_{2,K} = \boldsymbol{z_e}^T K \boldsymbol{z_e}, \qquad \|\boldsymbol{z_e}\|_{\infty,K} = max(K^{1/2}\boldsymbol{z_e}), \qquad \left\|\boldsymbol{u}_{0/1}\right\|_{2,R} = \boldsymbol{u}_{0/1}^T R\, \boldsymbol{u}_{0/1} \tag{30}$$

The matrices R and K are positive definite symmetric, where $K = K^{1/2}K^{1/2}$ and where $K^{1/2}$ has real entries which can be computed by diagonalization since the diagonal terms will be positive. Here, the infinity norm is needed to achieve convergence of all the states of the system. The 2-norm, although not strictly needed, can improve the performance of the control by reducing the state errors in a more efficient way than the infinity norm.

In the particular case of de-tumbling a spacecraft, the state error is $\boldsymbol{z_e} = \boldsymbol{\omega}$, while for the slew maneuver it is taken as:

$$\boldsymbol{z_e} = \begin{bmatrix} \mathbb{1}_{4\times4} & \mathbb{0}_{4\times3} \\ \mathbb{0}_{3\times4} & K_2 \end{bmatrix}^{\frac{1}{2}} \begin{Bmatrix} \boldsymbol{q_e} \\ \boldsymbol{\omega} \end{Bmatrix} \qquad \text{and} \quad \boldsymbol{q_e} = \begin{Bmatrix} q_1 \\ q_2 \\ q_3 \\ 1-q_4 \end{Bmatrix} \tag{31}$$

where K_2 is a positive definite symmetric matrix, with a unique and symmetric square root. The quaternions were deemed the most efficient attitude representation for the plant model neural network used for prediction since they are computationally efficient relative to other global representations such as the direct cosine matrix.

In Eq. 29, the norm of the control cannot decrease proportionally with time since the thrust is either on or off with fixed magnitude, while the norm of the state error decreases during the maneuvre. For this reason, a variable value of R is required to improve performance. We define a function $E(t)$ which is chosen depending on the control objective and

define the time-dependent weight matrix $R(t)$ as:

$$R(t) = \frac{E(t)}{E_0}\bar{R} \tag{32}$$

where the weight matrix $\bar{R}$ is chosen as a positive definite matrix. This is initially defined as the identity matrix multiplied by a single, tunable, scalar with $E_0 = E(0)$. This modification prevents the control norm from becoming too high relative to the state error norm, which will lead to poor convergence. Explicitly $E(t) = E_1$ for the de-tumbling control problem and $E(t) = E_2$ for slew motions are defined by:

$$E_1 = \frac{1}{2}\left\|\boldsymbol{\omega}\right\|_{2,I}^2 \qquad \text{and} \qquad E_2 = \left\|\boldsymbol{q}\right\|_{2,I_4}^2 = 2(1 - q_4). \tag{33}$$

At each time step, the genetic algorithm computes the N future torque vectors of the time horizon by minimizing the cost function, and the first torque ($t' = t + 1$) is selected.

$$\boldsymbol{u}_{0/1}^{opti} = \left[\underset{\boldsymbol{u}_{0/1}\in\{0,1\}^{4\times N}}{arg\,min}\; L(\boldsymbol{u}_{0/1})\right]_{t'=t+1} \tag{34}$$

In addition, the use of the following time-variable law for $\bar{R}(t)$ is investigated and shown to improve convergence further with a reduced fuel consumption:

$$\frac{d\bar{R}}{dt} = \bar{R}\frac{\dot{E} - \dot{E}_r}{\dot{E}_r t_c} \tag{35}$$

where $t_c > 0$ is a, tunable, time constant which effects the sensitivity of $\bar{R}$, a high value leads to slower variations of $\bar{R}$. Practically t_c must be taken higher than the time step for stability. $\dot{E}_r$ is a reference energy rate that acts as a threshold for $\dot{E}$ under which $\bar{R}$ increases and above which $\bar{R}$ decreases. After normalization by the state error norm, $\dot{E}_r$ can be chosen within a finite range of possible values, regardless of the maneuver initial conditions. In particular, using the functions in Eq. 33, we write $\dot{E} = \dot{E}_1$ for the de-tumbling case and $\dot{E} = \dot{E}_2$ for a slew motion where $\dot{E}_1 = \boldsymbol{\omega}^T\boldsymbol{u}$ and $\dot{E}_2 = \boldsymbol{\omega}^T\bar{\boldsymbol{q}}$. The reference energy rate for the detumbling case is chosen as $\dot{E}_r = -u_r\left\|\boldsymbol{\omega}\right\|_2$ where u_r is a positive constant chosen within the finite range of possible values of $-\boldsymbol{\omega}^T\boldsymbol{u}/\left\|\boldsymbol{\omega}\right\|_2$. It leads to the following time law of $\bar{R}_1$ for the de-tumble maneuver:

$$\frac{d\bar{R}_1}{dt} = -\frac{\bar{R}_1}{u_r t_c}\left(u_r + \frac{\boldsymbol{\omega}^T\boldsymbol{u}}{\left\|\boldsymbol{\omega}\right\|_2}\right) \tag{36}$$

The higher u_r, the higher the average value of $\bar{R}$ hence lowering the fuel consumption and increasing the settling time. When no torque is produced, $\bar{R}$ decreases and when the energy rate is more negative than the threshold, $\bar{R}$ increases.

The time law of $\bar{R}$ in the performance of the slew maneuver is obtained in a similar way, after normalizing the

nominator and denominator by $\|\boldsymbol{\omega}\|_2 \|\bar{\boldsymbol{q}}\|_2$:

$$\frac{d\bar{R}_2}{dt} = -\frac{\bar{R}_2}{\beta\, t_c}\left(\beta + \frac{\boldsymbol{\omega}^T \bar{\boldsymbol{q}}}{\|\boldsymbol{\omega}\|_2 \|\bar{\boldsymbol{q}}\|_2}\right) \tag{37}$$

where $\beta \in (0; 1]$ is a threshold such that $\bar{R}$ increases when the cosine of the angle between $\boldsymbol{\omega}$ and $\bar{\boldsymbol{q}}$ is lower than $-\beta$ and decreases otherwise.

D. Multi-Layer Perceptron control trained with NPC data

The control described so far is has a high on-line computational cost because of the necessity of performing an optimization at each time step. We propose in this paper, in a similar way ESA used in [26] for landing control, a second neural network trained before flight that uses data obtained by NPC. This second neural network uses a multilayer perceptron (MLP) configuration. MLPs are feedforward neural networks (as opposed to recurrent neural networks such as the NARX network used for modeling the spacecraft's dynamics) with several hidden layers used for classification problems, by associating an output category to each input presented to the MLP. In the present case, the 15 categories are the different available torques, and the inputs are the states of the system and past torques, delayed several times. The training set is composed of a high number of random states and torques inputs, associated to the corresponding ideal thruster activation output obtained by the NPC algorithm. This way, by presenting a high number of situations to the MLP, it can produce a control similar to the NPC with a very reduced computational cost, since it only need to be forward propagated online with no additional training. Note that in this training only the NPC version with constant $\bar{R}$ can be used, and for this reason it has to be tuned accurately before being used in the training data generation.

V. Simulation results: De-tumbling and large angle slew maneuvers

In this section the three control laws are applied to: (i) perform a de-tumbling maneuver of a 12U CubeSat, as was preliminarily undertaken in [32] and (ii) perform a large angle slew maneuver. In both cases, another control strategy is obtained by training a Multi-Layer Perceptron off-line with a high number data generated by genetic NPC, which is used on-line to control the spacecraft with reduced computational cost.

The case study considered is the Lunar Meteoroid Impacts Observer (LUMIO) spacecraft [33] which has a minimal thruster configuration within the ADCS architecture. Mass, volume and power of the ADCS sub-system were minimized by selecting (from the available commercial off the shelf (COTS) options) sensors with the smallest mass, volume and power that satisfied the pointing requirements. The sensors shown are a nano SSOC-D60 Sun sensor manufactured by Solar MEMs technology, two ST 400 star trackers manufactured by Hyperion technology and Berlin technologies and STIM 300 ultra-high performance inertial measurement unit manufactured by Sensonor10. The on-board computer

is the GOMspace-Z7000. The actuators are 3 Blue Canyon RWP-100 reaction wheels which are used for fine 3-axis tracking of a reference attitude that is designed for Moon pointing and that enables maximum power generation [33]. Finally, the VACCO propulsion system is proposed for the de-tumbling and de-saturation manoeuvres of the mission which is a minimal set of four reaction control systems thrusters chosen to minimize mass and volume. The parameters used in the simulation are the principal moments of inertia $I_1 = 26.66 \times 10^{-2} kg.m^2$, $I_2 = 26 \times 10^{-2} kg.m^2$, $I_3 = 16.66 \times 10^{-2} kg.m^2$. The initial angular velocity is $\omega(0) = [0.45; 0.52; 0.55]^T rad/s$. The thrusters have the same thrust level $F = 10mN$, the geometric parameters are taken as $\alpha = 30°$, $x = 0.05m$, $l = 0.15m$. The performance metric is the number of thrust pulses $N_{pulses} = \sum_{i=1}^{3} \sum_{t'=t_0}^{t_f} \boldsymbol{u}_{0/1}$ which computes how many time steps a thruster is turned on. The total impulse $I_{tot} = N_p ulses\, F\, dt$ and the propellant mass consumption $m_p = I_{tot}/(I_s g_0)$ can be derived from it, where dt is the time step, I_s is the specific impulse of the thrusters, and g_0 the gravity acceleration on Earth. In the following, the settling times are accurate to within 1s, the time step used in the simulations. Although in the LUMIO mission it is likely reaction wheels will be used for the slew maneuver, the use of four thrusters is considered here. The use of four thrusters could be useful in case of reaction wheel failure or also in the design of all small electric spacecraft utilizing micro-electric thrusters.

De-tumbling is considered complete when the angular velocity relative to each body axis has an absolute value less than $0.002 rad/s$. The angular velocity is normalized in the cost function with $\omega^{norm} = [2.0; 2.0; 2.0]^T rad/s$. The neural network used for propagation of the state uses 50 neurons in its hidden layer. 3 delays are used for the activation thrust vector and for the delayed states so that the first order time derivative equations of the system's dynamics can be modeled with good accuracy, resulting in 21 inputs. For the slew maneuver, a rest-to-rest situation is assumed. The neural network takes the current state of the system composed of the angular velocity and the quaternions, using 3 delays, resulting in an input vector of 33 elements. The initial quaternions are taken as $q(0) = [0.8; -0.5; -0.1; 0.32]^T$. $N = 30$ times steps are taken for the time horizon, with a time step of 1s. We consider that the state has converged when the three first quaternions are lower than 0.05 and the angular velocity relative to each body axis has an absolute value less than $0.02 rad/s$ and stay lower until the end of the simulation.

The genetic algorithm is performed with a maximum of 50 generations, and a population size of 100. The future torques found at time t that minimize the cost function are used in the initial population of the genetic optimization at next time step.

A. De-tumbling of a 12U CubeSat

The simple logic control (13) leads to $N_{pulses} = 634$ thrust pulses with a settling time of $316s$, and the projection control (16), after selecting the best coefficient $k_1 = 4$, leads to $N_{pulses} = 652$ thrust pulses with a settling time of $323s$. The number of thrust pulses and settling time of the NPC control are shown in Tab. 1, for various values of the tunable weights parameters of Eq. 29, using the normalization of R of Eq. 32 with constant $\bar{R}$. In the results, $\bar{R}$ is

taken as a scalar multiplying the identity matrix since there is no reason to discriminate the thrusters which are all the same. The weights are chosen is such a way that $K_{quad} + K_\infty = 1$, and the ratio of the two parameters can vary. This way, only the parameter $\bar{R}$ has an influence on the ratio of the control norm and the state norm. At equal settling time the lowest number of thrust pulses obtained with NPC using fixed $\bar{R}$ is of $N_{pulses} = 508$, for $\bar{R} = 0.3$ and $K_{quad} = 0$, which corresponds to 20% savings with respect to the simple logic control. This shows a clear improvement in the total impulse reduction. The settling times using the different methods are very similar.

In Tab. 2 the results using the variable law of $\bar{R}$ described in Eq. 36 are shown for several values of u_r, a characteristic time of $20s$ and $K_{quad} = 0$. Depending on the settling time constraints of the mission, the parameter u_r can be tuned given the available torques, independently of the initial state of the system and of the system's characteristics. Very low fuel consumption ($N_{pulses} = 468$) is obtained in a finite time for $u_r = 3 \cdot 10^{-4} Nm$ but with a high settling time. By taking $u_r = 5 \cdot 10^{-4} Nm$, a slightly higher fuel consumption ($N_{pulses} = 481$) but still better than the other methods presented in the paper including the perfect tuning of NPC with constant $\bar{R}$ is obtained, with a settling time comparable to those obtained by the other methods.

In Figure 3 the simulations using the on-line NPC method with fixed and variable $\bar{R}$, the simple logic method and the projection control, and the corresponding numbers of thrust pulses, are shown. The control parameters have been chosen as to minimize the number of thrust pulses. While the differences in settling time are very low, the use of the on-line optimization NPC leads to a clear improvement of the de-tumbling cost.

Table 1 Number of thrust pulses (a) and settling time (b) for de-tumbling a 12U CubeSat with 4 thrusters, with the NPC control using fixed values of $\bar{R}$ (N.C.=non converged)

(a) Thrust pulses	$\bar{R} = 0.01$	$\bar{R} = 0.03$	$\bar{R} = 0.1$	$\bar{R} = 0.3$	$\bar{R} = 1$
$K_\infty/K_{quad} = 0.1$	545	510	N.C.	N.C.	N.C.
$K_\infty/K_{quad} = 0.3$	605	550	508	N.C.	N.C.
$K_\infty/K_{quad} = 1$	610	589	571	541	N.C.
$K_{quad} = 0$	618	595	544	508	N.C.

(b) Settling time [s]	$\bar{R} = 0.01$	$\bar{R} = 0.03$	$\bar{R} = 0.1$	$\bar{R} = 0.3$	$\bar{R} = 1$
$K_\infty/K_{quad} = 0.1$	307	455	N.C.	N.C.	N.C.
$K_\infty/K_{quad} = 0.3$	456	318	341	N.C.	N.C.
$K_\infty/K_{quad} = 1$	313	307	418	339	N.C.
$K_{quad} = 0$	309	308	312	338	N.C.

Table 2 Number of thrust pulses and settling time for de-tumbling a 12U CubeSat with 4 thrusters, with the NPC control using a time-variable $\bar{R}$

	$u_r = 3 \cdot 10^{-4}$ Nm	$u_r = 5 \cdot 10^{-4}$ Nm	$u_r = 8 \cdot 10^{-4}$ Nm
Thrust pulses	468	481	542
Settling time [s]	568	387	346

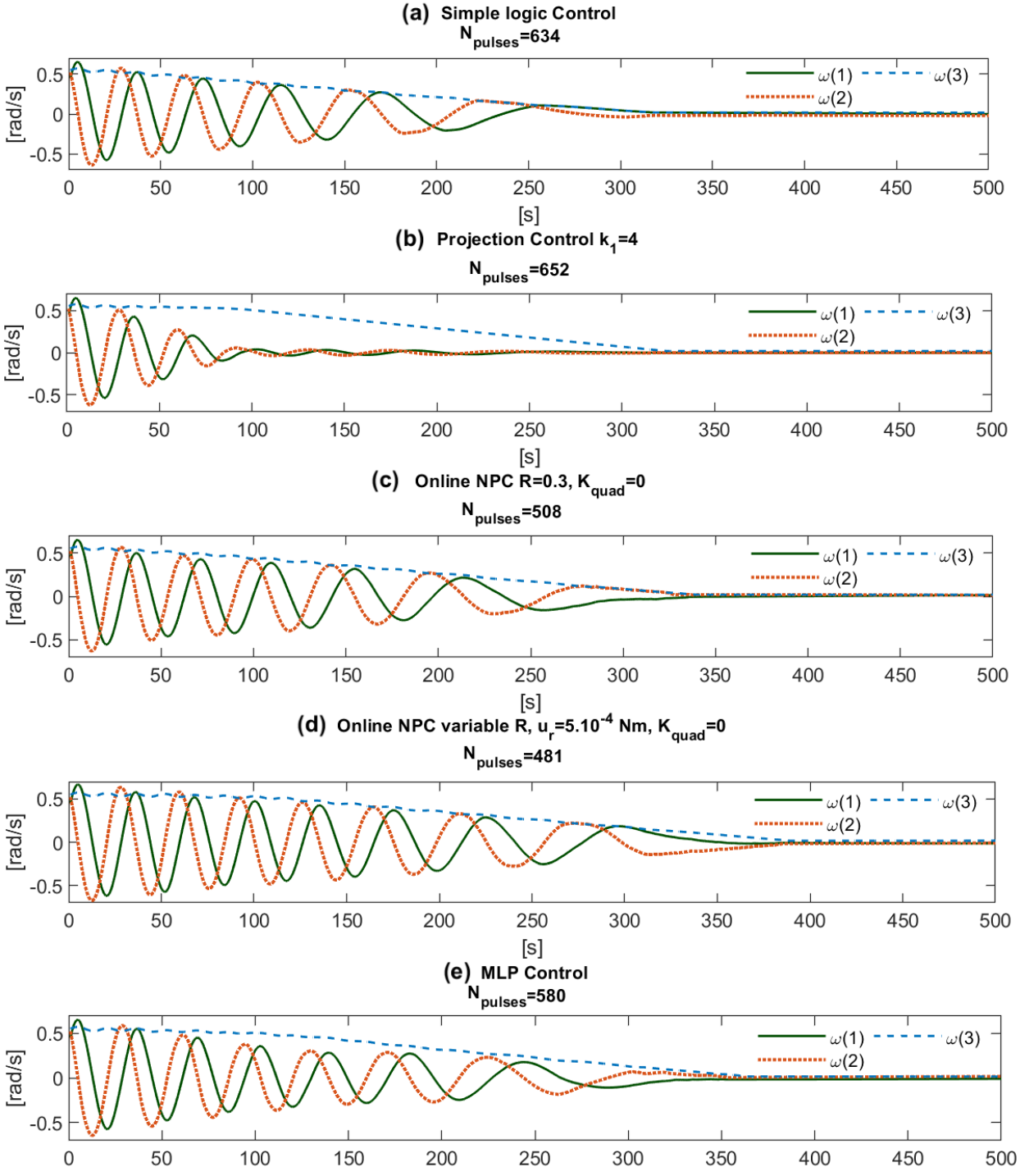

Fig. 3 Angular velocities during the detumbling of a 12U CubeSat with 4 thrusters, comparison of 5 different methods

Different MLPs are trained with data obtained with the genetic algorithm-based predictive optimization. Good results are obtained with 3 hidden layers. Each result presented has been obtained with 12 different neural network training processes. Each training uses an early stop procedure; if the training error keeps decreasing but a generalization

error obtained with a validation set starts increasing, the process is stopped and the best generalization error is retained. Each set of results has been tried with 10, 20 and 30 neurons in each of the 3 hidden layers, and the best number of neurons has been retained. Each training has been done with a mixed training set, composed of one set (70% of the full set) of high angular velocities, covering the maximum considered range $[-0.7; 0.7] rad/s$, and one set (30% of the full set) of low angular velocities $[-0.2; 0.2] rad/s$, in order to obtain a good behavior of the neural controller in any condition. The values for the generation of the optimized data has been taken as those that minimize the number of thrust pulses in the on-line NPC with fixed $\bar{R}$. The time step is 1s. The generalization error $E_{gen}^{\%}$ is after training the percentage of inputs of a test set, different than the training set, that lead to a wrong output by the trained MLP.

Table 3 Comparison of the generalization error (in %) using fully random or pseudo-random delayed state inputs, with 10 000 training data

	fully random inputs	pseudo-random inputs
best S	20	20
$E_{gen}^{\%}$	23.5	19.0

Results shown in Tab. 3 confirm that the use of pseudo-random delayed state inputs instead of fully random inputs improves the generalization capability of the Network trained in the same conditions $Q = 10000$ data and $S = 20$ neurons in each layer.

Tab. 4 evidences that increasing the number of training data efficiently reduces the generalization error, and leads to lower fuel usage, ultimately tending to the on-line optimization performances. With the neural networks used in this paper, clear savings are obtained with the off-line trained MLP with respect to the simple logic control and the projection control. The results also show that the best number of neurons increases when the data length increases.

Tab. 5 shows the number of floating point operations (flop) per time step required by the spacecraft's on-board computer for different number of neurons, assumed to be the same in each of the 3 hidden layers of the MLP, and is compared to the number of flop required by the other methods. The performance of the Neuro-controller tends to the NPC performance when the computational resources on earth increase, with a drastically reduced on-board computational cost. In comparison, the conventional controls show a very low number of flop per time step.

The net force originated during the detumbling does not vary much from a method to the other, and leads to a total impulse of $[0.101; 0.224; 0.031]^T Ns$ in the inertial referential defined by the inertia axes of the spacecraft in its initial position. Note that practically the sensors giving the angular position measurements are not available during the detumbling, and any attempt to integrate numerically the angular velocity would lead to exponentially increasing errors.

B. Large angle slew maneuver of a 12U CubeSat

A large angle slew maneuver is performed with the online optimization using a genetic algorithm and a neural network plant model, and the results are compared to those obtained with the simple logic 13 and the projection control

Table 4 Performance (generalization error, number of thrust pulses for de-tumbling and settling time) using different number of training data

	Q=2000	Q=10 000	Online Optimization
best S	10	20	-
$E^{\%}_{gen}$	26.1	19.0	-
N_{pulses}	661	580	522
Δt_s	349s	330s	321s

Table 5 Number of floating points operations required at each time step by the spacecraft's on-board computer for 3 hidden layers MLP controllers for different numbers of neurons and comparison to the other methods

Control law	flop per time step
simple logic	14
projection	59
online NMPC	246M
MLP 10 neurons	1070
MLP 20 neurons	2920
MLP 30 neurons	5570

16.

The simple logic requires the tuning of a single parameter because only the sign of the continuous control of Eq. 12 matters, the best results have been obtained for $k_2 = 0.043$ and taking arbitrarily $k_1 = 1$ It leads to 99 thrust pulses and a settling time of 49s.

Table 6 Number of thrust pulses (a) and settling time (b) for a large angle slew maneuver of a 12U CubeSat with 4 thrusters, with a projection control based on the continuous control $u^c = -k_1 I\omega_e - k_2 q_e$ (N.C.=non converged)

(a) Thrust pulses	$k_1 = 0.2$	$k_1 = 0.3$	$k_1 = 0.4$	$k_1 = 0.5$	$k_1 = 0.6$	$k_1 = 0.7$	$k_1 = 0.8$	$k_1 = 0.9$
$k_2 = 0.02$	N.C.	N.C.	N.C.	N.C.	59	N.C.	N.C.	N.C.
$k_2 = 0.03$	256	169	129	108	81	71	70	66
$k_2 = 0.04$	334	237	166	128	119	106	73	80
$k_2 = 0.05$	451	294	215	174	130	133	110	98

(b) Settling time [s]	$k_1 = 0.2$	$k_1 = 0.3$	$k_1 = 0.4$	$k_1 = 0.5$	$k_1 = 0.6$	$k_1 = 0.7$	$k_1 = 0.8$	$k_1 = 0.9$
$k_2 = 0.02$	N.C.	N.C.	N.C.	N.C.	59	N.C.	N.C.	N.C.
$k_2 = 0.03$	141	96	71	63	39	52	58	63
$k_2 = 0.04$	178	124	95	71	68	61	39	51
$k_2 = 0.05$	235	156	119	95	72	70	66	60

The projection control requires the tuning of two parameters k_1 and k_2, Tab. 6 show the number of thrust pulses and settling time of a batch computation when varying the two parameters in the range of minimal fuel consumption. It can be seen that the number pulses can be drastically reduced, but the control is very sensitive to the choice of parameters,

Table 7 Number of thrust pulses for a large angle slew maneuver of a 12U CubeSat with 4 thrusters, with the NPC control using fixed values of $\bar{R}$ (N.C.=non converged)

		$\bar{R} = 0.03$	$\bar{R} = 0.1$	$\bar{R} = 0.3$	$\bar{R} = 1$	$\bar{R} = 3$	best
$K_\infty/K_{quad} = 0$	$K_2 = 0$	57	63	N.C.	N.C.	N.C.	**52**
	$K_2 = 0.01$	54	64	N.C.	N.C.	N.C.	
	$K_2 = 0.1$	58	63	N.C.	N.C.	N.C.	
	$K_2 = 1$	57	65	N.C.	N.C.	N.C.	
	$K_2 = 10$	52	54	N.C.	N.C.	N.C.	
$K_\infty/K_{quad} = 0.3$	$K_2 = 0$	70	58	59	N.C.	N.C.	**56**
	$K_2 = 0.01$	68	63	59	N.C.	N.C.	
	$K_2 = 0.1$	68	60	58	N.C.	N.C.	
	$K_2 = 1$	67	62	56	N.C.	N.C.	
	$K_2 = 10$	77	65	65	N.C.	N.C.	
$K_\infty/K_{quad} = 3$	$K_2 = 0$	67	72	65	57	N.C.	**52**
	$K_2 = 0.01$	72	63	69	55	N.C.	
	$K_2 = 0.1$	72	70	64	52	N.C.	
	$K_2 = 1$	74	69	61	57	N.C.	
	$K_2 = 10$	84	83	71	63	N.C.	
$K_{quad} = 0$	$K_2 = 0$	71	73	64	52	N.C.	**52**
	$K_2 = 0.01$	75	68	68	54	N.C.	
	$K_2 = 0.1$	66	66	59	55	N.C.	
	$K_2 = 1$	75	66	67	57	N.C.	
	$K_2 = 10$	82	80	67	59	N.C.	

especially when low numbers of thrust pulses are obtained, and the settling time increases. Only a specific range is shown here in order to emphasize the sensitivity to the parameters, but other combinations lead to equivalent results.

The design of the NPC control for the slew maneuver requires tuning three parameters: K_∞/K_{quad} that can equal infinity, K_2 and $\bar{R}$, as in Eq. 29 and Eq. 32. After normalization of the state, K_2 and $\bar{R}$ are taken proportional to the identity matrix. In Tab. 7 the results of a batch computation while varying the three parameters are shown. In a similar way as for the detumbling case, the sum $K_{quad} + K_\infty$ is fixed to one so that only $\bar{R}$ has an influence on the ratio between the control norm and the state norm. Note that values of K_2 higher than 20 lead to lower performances. The settling time of the NPC is not shown because low variations are obtained, between $40s$ and $60s$. A clear improvement is obtained with respect to the simple logic of 47% fuel savings when taking the best parameters. Taking only an infinite norm of the state error in the cost function proves slightly more efficient. It can be noticed that the parameters shown in this table cover a wide range of values, and the variations of the number of thrust pulses are much lower than for the projection control whose results covered a narrow range, and the settling time is almost constant. This shows that the parameters used in the NPC control have a more universal meaning, and the results are less sensitive to them. When no constraint on settling time is considered, the projection control leads to the lowest fuel usage.

Table 8 Number of thrust pulses and settling time for performing a slew maneuver with a 12U CubeSat with 4 thrusters, with NPC control using a time-variable $\bar{R}$

	$\beta = 0.5$	$\beta = 0.6$	$\beta = 0.7$	$\beta = 0.8$	$\beta = 0.9$
Thrust pulses	43	44	42	49	38
Settling time [s]	61	60	53	55	53

It can be seen that by varying the right parameters, only $\bar{R}$ has a real influence on the results. In particular at a given ratio K_∞/K_{quad}, there exists a limit value of $\bar{R}$ above which there is no convergence, but that must be approached in order to reduce the number of thrust pulses, which justifies the implementation of a time-variable $\bar{R}$. The time law of $\bar{R}$ described in Eq. 37 is implemented with different values of the parameter β between 0 and 1 as shown in Tab. 8 where an upper bound of 4 has been used for $\bar{R}$. The dependance on β is very low, both for the number of thrust pulses and the settling time. For example for $\beta = 0.9$ we obtain 38 thrust pulses, which shows a clear improvement with respect to the NPC with constant $\bar{R}$, and with a comparable settling time of $53s$. This method, besides insuring convergence by diminishing $\bar{R}$ when it becomes too high, leads to the best performance by taking advantage of the natural motion of the spacecraft, and leads to 35.6% fuel savings with respect to the best projection control, and proves more robust and much less sensitive to the choice of the parameters.

As for the de-tumbling, different Multi-Layer perceptrons are trained off-line, using a set of pseudo-random inputs obtained by propagating an initial random state during three time steps, and the corresponding outputs obtained with the NPC control with parameters $\bar{R} = 1$, $K_2 = 0$, $K_{quad} = 0$. 10.000 pairs of inputs/outputs are generated to train a MLP of 3 hidden layers, with $S = 20$ neurons in each layer. Half of the training inputs are taken in the full range of quaternions (between -1 and 1) and expected range of angular velocities, and the other half are with lower quaternions. 24 neural networks are trained, the best has a generalization error of 22.5%, and leads to 69 thrust pulses (30% saving with respect to the simple logic control) and a settling time of $47s$. Figure 4 shows the quaternions and angular velocity obtained with the the five controls developed in the article. The results are shown at equal settling time of $55s$. The neural network control is close to the on-line optimization predictive control, evidencing the good generalization capability from the training process. It shows a clear improvement with respect to a simple logic in terms of fuel consumption, and is faster than the projection control.

Finally, the net force created by the control does not vary much from a method to the other, and leads to a total impulse of $[0.057; 0.158; 0.056]^T Ns$ in the inertial referential defined by the inertia axes of the spacecraft in its desired final position. Note that with methods reducing the state error such as those used in this paper, the net force can hardly be reduced, path planning methods are required if the net force must be minimized.

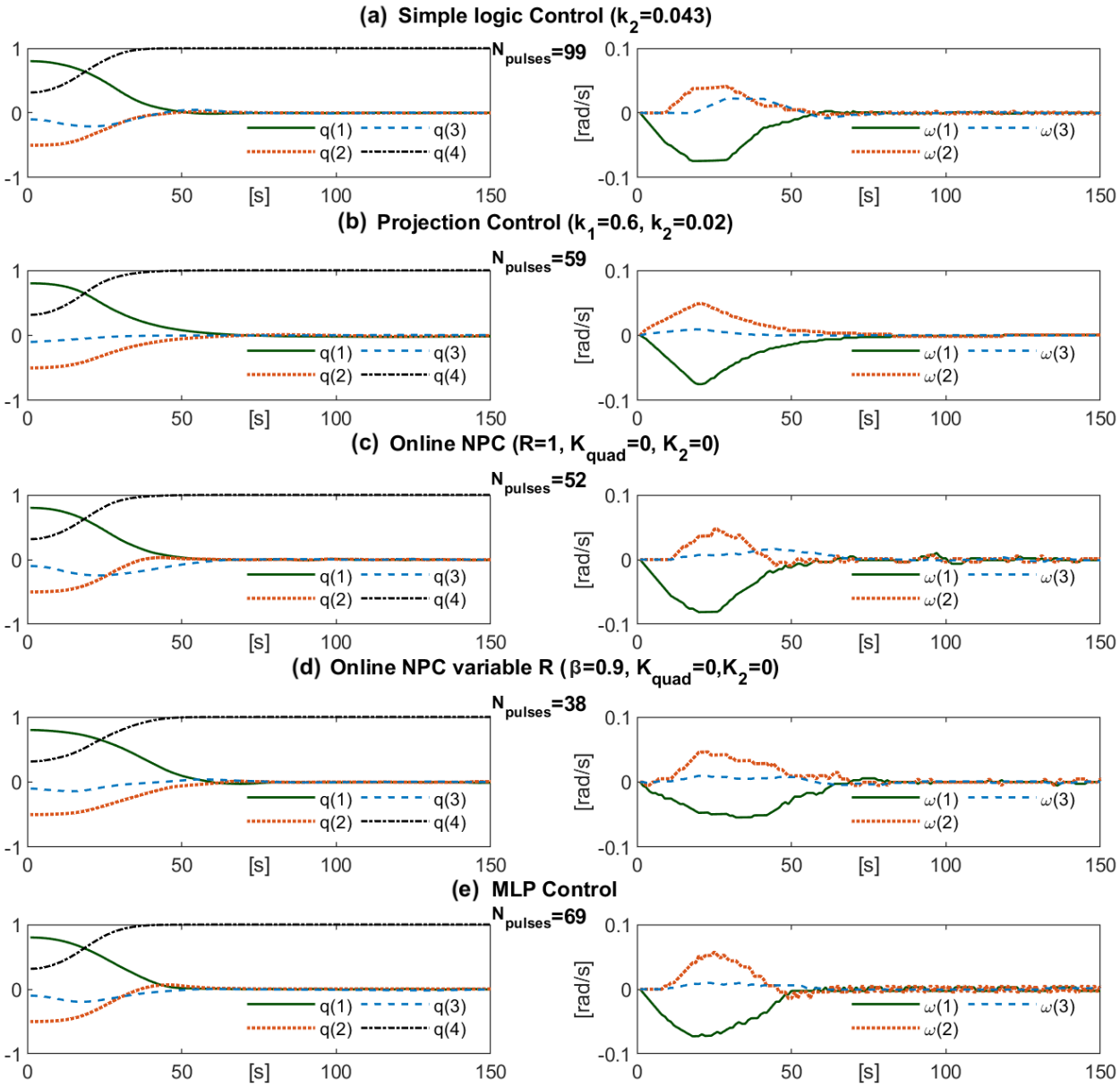

Fig. 4 Quaternions (left) and angular velocities (right) during the slew maneuver of a 12U CubeSat with 4 thrusters, comparison of 5 different methods at equal settling time

VI. Conclusion

This paper presents a Neural Predictive control adapted to nonlinear systems with boolean control inputs. The method is applied to the optimal control of a 12U CubeSat with four thrusters. An extensive comparison of the approach to a logic-based control law and a projection based control showed that the Neural Predictive Control is more fuel-efficient for de-tumbling maneuvers, with up to 25% fuel savings. In addition, for finite-time slew motions up to 36% fuel savings were observed. It is also shown that tuning the NPC control comes down to tuning one parameter, for which an automatic time law that insures convergence and leads to the best performance is proposed. However, the main

problem with the neural predictive control is the relatively high processing requirement. To overcome this a neural network meta-model for optimal thruster selection was trained based on generated data from the optimal controller. The performance of the meta-model for thruster selection tends to the optimal performance as the number of neurons and training time is increased. In conclusion neural networks have been shown as a useful tool in predictive control by replacing the dynamics and kinematics with a neural network model for attitude propagation and cost function evaluation. In addition, we show that neural networks could be useful to reduce computational expense of on-board algorithms by replacing optimization algorithms with a trained neural meta-model. However, the development of such meta-models requires a heavy off-line investment. Furthermore, in contrast to models based on differential equations, neural network controls currently have no analytical means of validation.

References

[1] Servidia, P., Sànchez Peña, R.: Practical Stabilization in Attitude Thruster Control, *IEEE Transactions on aerospace and electronic systems*, Vol. 41, No. 2, pp. 584-598, (2005)
doi.org/10.1109/TAES.2005.1468750

[2] Sanchez Pena, R. S. and Alonso, R. and Anigstein, P. A.: Robust Optimal Solution to the Attitude/Force Control Problem, *IEEE Transactions on aerospace and electronic systems*, vol. 36, No. 3, pp. 784-792, (2000)
doi.org/10.1109/7.869496

[3] Bernelli-Zazzera, F., Mantegazza, P., Nurzia, V.: Multi-Pulse-Width Modulated Control of Linear Systems. *Journal of Guidance, Control, and Dynamics*, vol. 21, No. 1, pp. 64-70, (1998)
doi.org/10.2514/2.4198

[4] Wie, B.,: Space Vehicle Dynamics and Control,Second Edition. AIAA, pp. 450-457, (2008)
doi.org/10.2514/4.860119

[5] Anthony, T., Wie, B. and Caroll, S.,: Pulse-modulated control synthesis for a flexible spacecraft. *Journal of Guidance, Control, and Dynamics*, 13(6), pp.1014-1022, (1990)
doi.org/10.2514/3.20574

[6] Ieko, T., Ochi, Y. and Kanai, K.,: New Design Method for Pulse-Width Modulation Control Systems via Digital Redesign. *Journal of Guidance, Control, and Dynamics*, 22(1), pp.123-128., (1999)
doi.org/10.2514/2.4358

[7] Sidi, M.,: Spacecraft dynamics and control. Cambridge: Cambridge University Press, pp.265–289, (1997)
doi.org/10.1017/CBO9780511815652

[8] Zimpfer, D., Shieh, L. and Sunkel, J.,: Digitally Redesigned Pulse-Width Modulation Spacecraft Control. *Journal of Guidance,*

Control, and Dynamics, 21(4), pp.529-534, (1998)
doi.org/10.2514/2.4276

[9] Curti, F., Romano, M., Bevilacqua, R.,: Lyapunov-Based Thrusters' Selection for Spacecraft Control: Analysis and Experimentation. *Journal of Guidance, Control, and Dynamics*, vol. 33, No. 4, pp. 1143-1160, (2010)
doi.org/10.2514/1.47296

[10] Robinett, R. D. and Parker, G. G., Schaub, H., Junkins, J.,: Lyapunov Optimal Saturated Control for Nonlinear Systems. *Journal of Guidance, Control, and Dynamics*, vol. 20, No. 6, pp. 1083-1088, (1997)
doi.org/10.2514/2.4189

[11] Holderbaum, W.: Control strategy for Boolean input systems. Proceedings of the 2002 American Control Conference (IEEE Cat. No.CH37301), (2002)
doi.org/10.1109/ACC.2002.1023193

[12] Wright S.J.: Applying New Optimization Algorithms To Model Predictive Control. Fifth International Conference on Chemical Process Control – CPC V, (1997)

[13] Dantzig, G.,: Linear Programming and Extensions. Princeton University Press, pp.94-146, (1998)
doi.org/10.7249/R366

[14] Nehrenz, M., and Sorgenfrei, M.,: A Comparison of Thruster Implementation Strategies for a Deep Space Nanosatellite. AIAA Guidance, Navigation, and Control Conference, (2015)
doi.org/10.2514/6.2015-0866

[15] Harkegard, O: Dynamic control allocation using constrained dynamic programming, *Journal of Guidance, Control, and Dynamics*, 27(6), (2004)
doi.org/10.2514/1.11607

[16] Garus, J: Optimization of thrust allocation in the propulsion system of underwater an vehicle, *Int. J. Appl. Math. Comput. Sci.*, 14(4), pp. 461-467 (2004)

[17] Weiss, A., Kalabic, U., Di Cairano, S: Model Predictive Control for simultaneous station keeping and momentum management of low-thrust satellites, 2015 American Control Conference (ACC), (2015)
doi.org/10.1109/ACC.2015.7171076

[18] Kronhaus, I., Schilling, K., Pietzka, M., Schein, J.,: Simple Orbit and Attitude Control Using Vacuum Arc Thrusters for Picosatellites, *Journal of Spacecraft and Rockets*, vol. 51, No. 6, pp. 2008-2015 (2014)
doi.org/10.2514/1.A32796

[19] Rawlings, J.B., Muske, K.R.: The stability of constrained receding horizon control. *IEEE Transactions on Automatic Control*, vol. 38, No. 10, pp. 1512-1516, (1993)
doi.org/10.1109/9.241565

[20] Eren, U., Prach, A., Kocer, B. B., Rakovic, S. V., Kayacan, E., Acikmese, B.,: Model Predictive Control in Aerospace Systems: Current State and Opportunities. *Journal of Guidance, Control, and Dynamics*, vol. 40, No. 7, pp. 1541-1566, (2017) doi.org/10.2514/1.G002507

[21] Atiya, A.F., and Parlos, A.G.,: New results on recurrent network training: unifying the algorithms and accelerating convergence. *IEEE Transactions on Neural Networks*, vol. 11, No. 3, pp. 697-709, (2000) doi.org/10.1109/72.846741

[22] Gori, M., Hammer, B., Hitzler, P., Palm, G.,: Perspectives and challenges for recurrent neural network training *IEEE Transactions on Neural Networks*, vol. 18, No. 5, pp. 617-619, (2009) doi.org/10.1093/jigpal/jzp042

[23] Saravanan, N., Duyar, A., Guo, T.-H., Merrill, W. C.: Modeling space shuttle main engine using feed-forward neural networks. *Journal of Guidance, Control, and Dynamics*, vol. 17, No. 4, pp 641-648, (1994) doi.org/1010.2514/3.21250

[24] Battipede, M., Gili, P., Massotti, L.: Neural and LQR Optimal Attitude Control of a Flexible Micro-Satellite. *AIAA Guidance, Navigation, and Control Conference and Exhibit*, (2003) doi.org/10.2514/6.2003-5434

[25] Fornasini, E., Valcher, M. E.,: Optimal Control of Boolean Control Networks, *IEEE Transactions on Automatic Control*, vol. 59, No. 5, pp. 1258-1270, (2014) doi.org/10.1109/TAC.2013.2294821

[26] Sánchez-Sánchez, C., Izzo, D.: Real-Time Optimal Control via Deep Neural Networks: Study on Landing Problems. *Journal of Guidance, Control, and Dynamics*, 41(5), pp.1122-1135.(2018) doi.org/10.2514/1.G002357

[27] Bai, Y., Biggs, J. D., Wang, X., Cui, N.,: A singular adaptive attitude control with active disturbance rejection. *European Journal of Control*, vol. 35, pp. 50-56, (2017) doi.org/10.1016/j.ejcon.2017.01.002

[28] Bai, Y., Biggs, J. D., Bernelli, F., Wang, X., Cui, N.,: Adaptive Attitude Tracking with Active Uncertainty Rejection. *Journal of Guidance, Control, and Dynamics*, vol. 41, No. 2, pp. 550-558, (2018) doi.org/10.2514/1.G002391

[29] Ye, Dong, Zhang, Jianqiao, Sun, Zhaowei.: Extended state observer–based finite-time controller design for coupled spacecraft formation with actuator saturation. *Advances in Mechanical Engineering*, (2017) doi.org/10.1177/1687814017696413

[30] Cybenko, G.: Approximations by superpositions of sigmoidal functions. *Mathematics of Control, Signals, and Systems*, (1989) doi.org/10.1007/BF02551274

[31] Hornik, K.: Approximation Capabilities of Multilayer Feedforward Networks. *Neural Networks*, (1991) doi.org/10.1016/0893-6080(91)90009-T

[32] Biggs, J. D., Fournier.H., Ceccherini, S.,Topputo, F.,: Optimal de-tumbling of spacecraft with four Thrusters. *5th CEAS Conference on Guidance, Navigation and Control*, April (2019). (to appear)

[33] Speretta, S., Topputo, F., Biggs, J.D., Di Lizia,. P., Massari, M., Mani, K., Dei Tos D., Ceccherini, S., Franzese, V, Cervone, A., Sundaramoorthy, P., Noomen, R., Mestry, S., Cipriano, A., Ivanov, A., Labate, D., Jochemsen, A., Furfaro, R., Reddy, R. Vennekens, J.,Walker, R.: LUMIO: achieving autonomous operations for Lunar exploration with a CubeSat. *SpaceOps Conference*, (2018) doi.org/10.2514/6.2018-2599