

Optimal de-tumbling of spacecraft with four thrusters

James D. Biggs, Hugo Fournier, Simone Ceccherini, Francesco Topputo

Abstract Motivated by a drive towards spacecraft miniaturisation and the desire to undertake more complex missions in deep-space, this paper tackles the problem of de-tumbling spacecraft using a minimal number of attitude thrusters. The control problem addressed is to drive high tip-off angular velocity rates that result from imperfect orbit injection to within a required tolerance using only the on-off switching of 4 thrusters. This paper presents three possible control solutions to this problem (i) a logic-based controller that is simple to implement and requires no tuning (ii) a projective control that aims to replicate an ideal continuous control as closely as possible with the available torques and (iii) a Neural-network-based Predictive Control (NNPC) that is adapted to nonlinear control systems with boolean inputs. The NNPC is based on a Recurrent Neural Network (RNN) using a Nonlinear AutoRegressive exogenous configuration for time propagation of the state in a finite-time horizon optimization. Commonly, for continuous systems, a back-propagation algorithm for the receding horizon optimization is used, but this is not applicable to systems with discrete inputs and so is replaced by a genetic algorithm. In addition a Multi-Layer Perceptron (MLP) is trained off-line with optimal control data obtained with the NNPC resulting in an optimal control that can be implemented on-line with a significantly reduced on-board computational cost. The NNPC performance is compared to the proposed logic-based and projective de-tumbling control laws in simulation of a 12 U CubeSat and is shown to be the most efficient in terms of total impulse requirement.

1 Introduction

Recent interest in CubeSat deep-space missions has created numerous and unique challenges related to the highly constrained nature of these spacecraft. One such example is the Lunar Meteoroid Impact Observer (LUMIO) which is a 12 U CubeSat (20cm \times 20 cm \times 30 cm with a mass of approximately 25kg) that intends to deploy on the Libration Point orbit L_2 of the Earth-Moon-Spacecraft system [1]. One major challenge of such a mission is to minimise the amount of mass and volume of the spacecraft as well as the amount of fuel expended for the purpose of de-tumbling the spacecraft post-launch (magnetic torquers typically used for

James D. Biggs

Department of Aerospace Science and Technology, Politecnico di Milano, e-mail: jamesdouglas.biggs@polimi.it

Hugo Fournier

Department of Aerospace Science and Technology, Politecnico di Milano e-mail: fournier.hugo@mail.polimi.it

de-tumbling CubeSats are only useful in Low Earth Orbit), orbit transfer, station-keeping and de-saturation maneuvers. The paper addresses minimizing the mass, volume and fuel expenditure related to the ADCS system by using a minimal thruster reaction control system and optimal control laws.

In this paper the problem of de-tumbling a spacecraft in a fuel-efficient way using only the on-off switching of four thrusters is considered. The four on-off thrusters are configured on a single face of the spacecraft with each delivering a constant force F with their vectors denoted F_1, F_2, F_3, F_4 as depicted in Figure 1. Each thruster is inclined by an angle α to the horizontal. The propulsive system can deliver only a set of specific torques with fixed values on each axis according to the different combination of thruster. A combination of two thrusters can be used to apply a torque with the required sign of each axis. Again, although a

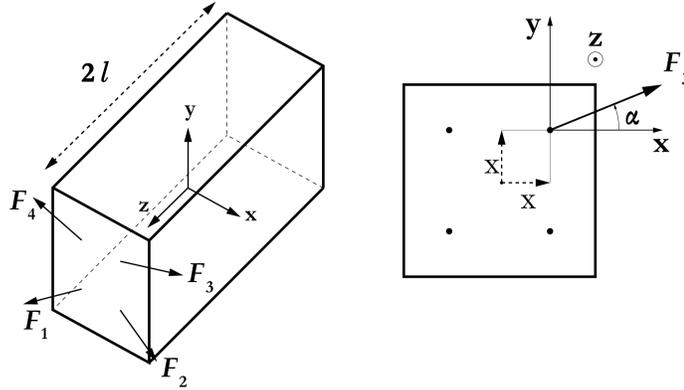


Fig. 1 Schematic representation of a spacecraft with 4 thrusters located on the same face

specific torque can be applied to a single axis at any instant in time it cannot be applied simultaneously to all axis. Controls designed for on-off thrusters have historically been dealt with by trying to replicate continuous control laws by discrete laws. Some of the first control algorithms useful for "on-off" actuators have been developed in [2] based on Lyapunov stability theory. More recent methods for controlling a spacecraft with on-off thrusters deal with the Thruster Selection Problem (TSP) by reducing the number of thrusters activated during a maneuver thus minimizing fuel usage. For example [3] develops a continuous "ideal" control law which is then mapped to the different thrusters depending on the configuration via a Linear Programming (LP) algorithm and finally a Pulse-Width Modulation (PWM) is then used, transforming a variable torque amplitude into a torque of variable duration. Different LP algorithms such as the primal-dual interior point and the simplex algorithm which provides a reliable linear optimization process with constraints can be used. The method combining a simplex algorithm and PWM has been implemented and adapted to the rototranslational dynamics of a spacecraft in [4]. The NASA Ames Center has also investigated optimal thruster selection using six and eight thrusters with fuel minimization [5] for a deep-space nano-satellite control configuration investigating both cold-gas and micro-electric propulsion devices for de-tumbling.

In this paper a nonlinear optimal controller is developed for the de-tumbling of a 4 thruster spacecraft using a Neural network-based predictive control adapted to boolean input systems. Classical Model-based Predictive Control (MPC) that utilises Linear Quadratic Regulator theory [8] [9] are not efficient for the highly nonlinear dynamics of a de-tumbling Spacecraft. Recent methods using Recurrent Neural Networks (RNN) [10] [11] have been used to replicate the nonlinear dynamics and can be fused with MPC to allow efficient time propagation of the state and dynamic Back-propagation for performing receding horizon opti-

mization. More recently MPC based on Neural networks has been adapted to nonlinear systems with boolean control inputs [12]. Here we use RNN to replicated the nonlinear dynamics so that the future state and cost function can be evaluated and optimized using a genetic algorithm and the optimal thruster selection chosen. Furthermore, the MLP is used to improve the computational efficiency whereby different MLPs are trained with optimal data obtained with the genetic algorithm-based predictive optimization.

2 Problem Statement

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

$$J\dot{\boldsymbol{\omega}} = -\boldsymbol{\omega} \times J\boldsymbol{\omega} + \mathbf{d} + \mathbf{u} \quad (1)$$

where \mathbf{d} is an internal or external disturbance, $\boldsymbol{\omega}$ the angular velocity in body fixed coordinates with respect to a fixed inertial frame, J the inertia matrix and \mathbf{u} a boolean input control defined by

$$\mathbf{u} = -T \cdot \mathbf{u}_{on/off} \quad (2)$$

where $\mathbf{u}_{on/off} \in \{0, 1\}^{4 \times 1}$ is a binary vector whose i -th element equals 1 when the i -th thruster is activated, and 0 when it is turned off and

$$T = \begin{bmatrix} lF \sin \alpha & lF \sin \alpha & -lF \sin \alpha & -lF \sin \alpha \\ -lF \cos \alpha & lF \cos \alpha & lF \cos \alpha & -lF \cos \alpha \\ xF \sin \alpha - xF \cos \alpha & xF \cos \alpha - xF \sin \alpha & xF \sin \alpha - xF \cos \alpha & xF \cos \alpha - xF \sin \alpha \end{bmatrix} \quad (3)$$

where $\alpha \in (0, \frac{\pi}{4})$ is defined by the thrusters orientation, x and l are geometric lengths of the spacecraft with its centre of mass assumed to be at the geometric centre. The control objective is to then drive $\boldsymbol{\omega}$ from a high initial angular velocity to within a required tolerance in a fuel-efficient (preferably optimal) way. In the following section three control laws are proposed with increasing degrees of implementation complexity.

3 De-tumbling control laws for four thrusters

In this section three control laws are presented based on (i) a logic-based control law which is simple to implement and requires no tuning (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. This control is simple to implement but requires tuning of the ideal controller and (iii) a neural-network based predictive control that efficiently propagates the state in a finite horizon optimization. The cost function considered is chosen to minimize the total impulse of the system. In NNPC (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. Here a genetic algorithm is used as an alternative to back-propagation which is applicable to discrete input systems.

3.1 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 and reliable control that reduces at each time t the error function in its maximum direction is given by a simple logic-based control:

$$\mathbf{u}_{sl} = -M_{i_{max}} \text{sign}(\omega_{i_{max}}) \mathbf{e}_{i_{max}} \quad (4)$$

where $\mathbf{e}_{i_{max}}$ with $i = 1, 2, 3$ form an orthonormal basis in the body frame.

$$i_{max} = \underset{i}{\text{arg max}} |\omega_i| \quad (5)$$

with

$$M = \left\{ \begin{array}{c} 2lF \sin \alpha \\ 2lF \cos \alpha \\ -2xF \sin \alpha + 2xF \cos \alpha \end{array} \right\} \quad (6)$$

Note that for a given set of thrusters and thruster configuration the control law requires no tuning and is therefore simple to implement.

3.2 Projection control

This control approach is based on a standard proportional continuous de-tumbling control $u^c = -kJ\boldsymbol{\omega}$ which is projected onto one of the 15 available torques u_i^p by selecting the one which is as close to the ideal torque as possible. 15 torques are available since 2^n combinations are possible with n thrusters and where the case of all thrusters being turned on is removed since it provides zero torque. In other words at each time-step the selected thruster configuration is chosen as the one that minimizes the cost function $J_c = \|u^c - u_i^p\|$. In this case the ideal control parameter k must be tuned to minimize the total impulse. Additionally, if there are significant internal disturbances (such as a time-varying inertia matrix due to fuel usage, sloshing, rotating solar panels or even actuator faults) then the ideal control can be amended to cancel these effects at each sampling period where

$$u^c = -kJ\boldsymbol{\omega} - \hat{d} \quad (7)$$

where \hat{d} is an estimate of the uncertain disturbance torques which can be obtained, for example, from a nonlinear extended state observer such as:

$$\begin{aligned} \frac{d\hat{\boldsymbol{\omega}}}{dt} &= J^{-1}(J\boldsymbol{\omega} \times \boldsymbol{\omega} + u + \hat{d}) + \beta_1 |\boldsymbol{\omega} - \hat{\boldsymbol{\omega}}|^\sigma \text{sgn}(\boldsymbol{\omega} - \hat{\boldsymbol{\omega}}) \\ \frac{d\hat{d}}{dt} &= \beta_2 |\boldsymbol{\omega} - \hat{\boldsymbol{\omega}}|^\sigma \text{sgn}(\boldsymbol{\omega} - \hat{\boldsymbol{\omega}}) \end{aligned} \quad (8)$$

where β_1, β_2, σ are tuning parameters. If the estimator is perfect the ideal continuous control (7) will asymptotically stabilize the origin even in the presence of disturbances. In practise the estimator is not perfect but

the control can be augmented to include an adaptive parameter that yields asymptotic stability to a tunable bounded region of the desired state [7] or alternatively the inclusion of a sliding-mode component ensures asymptotic stability even in the presence of an estimation error [6]. For attitude control in deep-space the effect of external disturbances such as solar radiation pressure should be considered, but over the short duration of a de-tumbling manoeuvre they have a negligible effect.

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

$$u_{on/off}^p = \underset{u_{on/off} \in \{0,1\}^{4 \times 1}}{\operatorname{argmin}} \Gamma(u_{on/off}) \quad (9)$$

where

$$\Gamma(u_{on/off}) = \|u^c + Tu_{on/off}\|. \quad (10)$$

In contrast to the simple-logic controller the projective control requires tuning of the ideal controller. For a simple proportional based ideal controller this may require only the tuning of 1 scalar gain, however, more complex ideal controls such as those which utilise active disturbance rejection would require the tuning of multiple parameters which significantly increases the complexity of implementation.

4 Neural network based predictive attitude control with four thrusters

In this section, a neural network-based predictive control is presented that is adapted to boolean inputs. It 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 neural controller. It is performed in two steps: (i) an optimal control is developed, it uses a 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 a second Multi-Layers Perceptron (MLP) off-line. 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.

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.

4.1 On-line optimization control with a genetic algorithm and Neural-Network prediction

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

$$F = \frac{1}{\Delta t} \int_t^{t+\Delta t} \omega^T K \omega + u^T R u dt \quad (11)$$

where K and R are weight matrices. The optimization variable is the 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 Neural Network Multi-Layers Perceptron is used whose structure is presented in Figure 2.

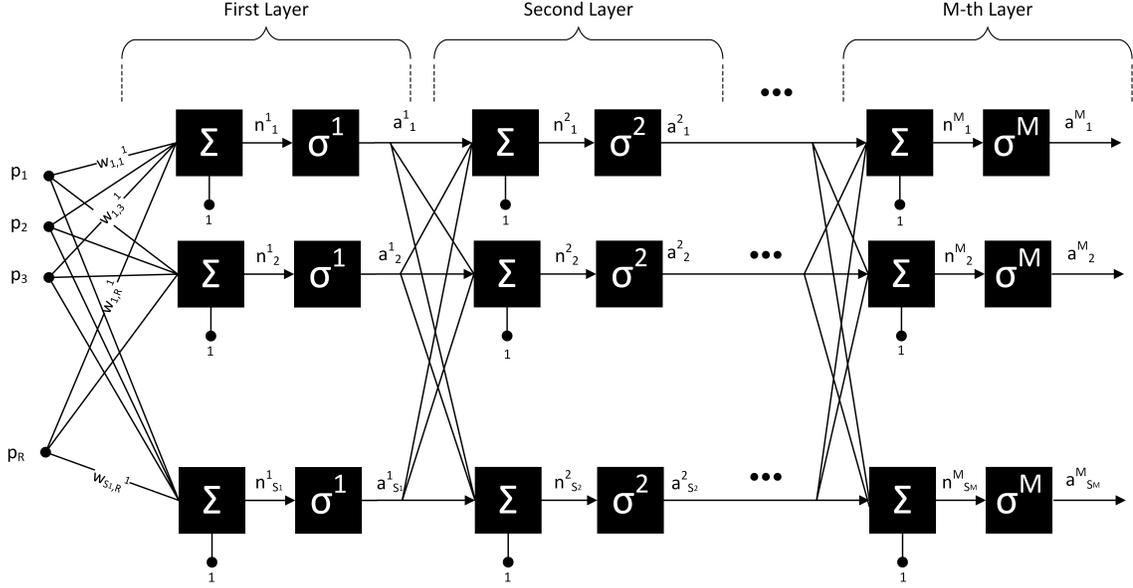


Fig. 2 Representation of a Multi-layers Perceptron

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

$$a^1 = \sigma^1(W^1 p + b^1) \quad (12)$$

$$a^{m+1} = \sigma^{m+1}(W^{m+1} a^m + b^{m+1}) \quad \text{for } m = 2 \dots M-1 \quad (13)$$

where $p \in \mathbb{R}^{S^0 \times 1}$ is the input, $a^m \in \mathbb{R}^{S^m}$ is the output of layer m , W^m is the weight matrix of layer m , σ^m is the activation function of layer m . S^m is the number of neurons of layer m . Layer M is the output layer, with S^M the number of outputs and S^0 the number of inputs of the Network.

In this paper, the activation functions are the log-sigmoid functions

$$\sigma(n) = \frac{1}{1 + e^{-n}} \quad (14)$$

in the hidden layers and identity functions in the outer layer.

Training a Multi-Layer Perceptron consists in providing it with a set of target inputs/outputs pairs and modifying its weights and biases so that the outputs of the network get as close as possible to the target outputs, for the same inputs. 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 those from the training set.

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

$$F^{tr}(x) = \frac{1}{Q} \sum_{q=1}^Q (t^q - a^q(x))^T (t^q - a^q(x)) \quad (15)$$

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

$$F^{tr} = v(x)^T v(x) \quad (16)$$

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

$$x(k+1) = x(k) - [\mathbf{J}^T(x_k)\mathbf{J}(x_k) + \mu_k I]^{-1} \mathbf{J}^T(x_k)v(x_k) \quad (17)$$

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

$$\mathbf{J} = \frac{\partial v}{\partial x} \in \mathbb{R}^{(S^M Q) \times (\sum_{m=0}^{M-1} (S^m+1)S^{m+1})} \quad (18)$$

is obtained by the backpropagation algorithm:

$$\begin{aligned} s^M &= \sigma^M \\ s^m &= s^{m+1} W^{m+1} \dot{\sigma}^m \quad \text{for } m = M-1 \dots 1 \end{aligned} \quad (19)$$

where

$$s^m = \frac{\partial a^M}{\partial n^m} \quad (20)$$

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

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

is the matrix composed of the derivatives of the activation function of layer m . In particular, a specific type of MLP with one hidden layer, a log-sigmoid activation function, and a linear activation function in the output layer is used based on a Nonlinear AutoRegressive exogenous configuration (NARX). 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) represented in Figure 3 by the tapped delay lines (TDL).

In case of continuous torques, this Neural Network can be used for computing the derivatives of the future states with respect to the current and future torques, hence allowing for a gradient-based or Jacobian

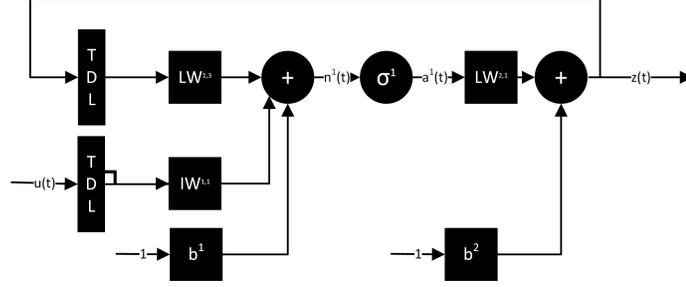


Fig. 3 Nonlinear AutoRegressive eXogenous (NARX) network used for predicting the future states of the system

based optimization. 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 $[u_{on/off}(q)]_{q \in [1;N]}$ where q is the discrete time.

An advantage of using a genetic algorithm is that no constraint exist on the form of the cost function. In this paper, we propose a cost function of the form

$$F = \frac{1}{N} \sum_{q=1}^N \|\omega\|_{2,K}^2 + K_{\infty} \|\omega\|_{\infty,K} + \|u_{on/off}\|_{2,R}^2 \quad (21)$$

where

$$\|\omega\|_{2,K} = \omega^T K \omega, \quad \|\omega\|_{\infty,K} = \max(K^{1/2} \omega), \quad \|u_{on/off}\|_{2,R} = u_{on/off}^T R u_{on/off} \quad (22)$$

The matrix K is 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. Thus, the quadratic norm of the state is dominant for high state errors, hence efficiently reducing the equivalent kinetic energy, and the infinity norm is dominant for low values of the state error, when accuracy in all directions is needed. The states are normalized with a value equal to the maximum expected value, so that the weights matrices in the cost function are close to unity.

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 corresponding to $N = 1$ is selected.

$$u_{on/off}^{opti} = \left[\underset{u_{on/off} \in \{0,1\}^{4 \times N}}{\operatorname{argmin}} F(u_{on/off}) \right]_{q=1} \quad (23)$$

5 Simulation results: De-tumbling of a 4-thruster 12U Cubesat

5.1 On-line optimization predictive control with genetic algorithm and Neural Network plant model

The case study considered is the LUMIO spacecraft which has a minimal thruster configuration within the ADCS architecture as shown in Figure 4. Mass, volume and power of the ADCS sub-system were mini-

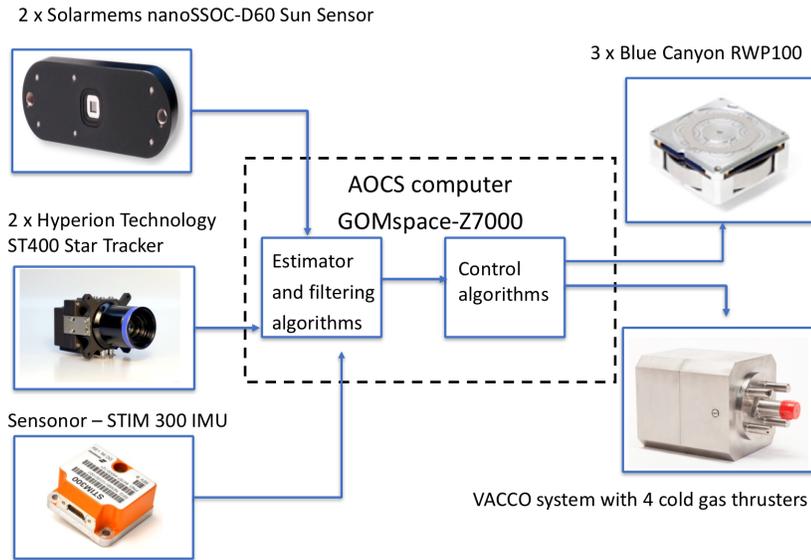


Fig. 4 ADCS architecture for the 12U LUMIO spacecraft

mized 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 Sensoror10. 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 [1]. 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 $J_1 = 26.66 \times 10^{-2} \text{kg.m}^2$, $J_2 = 26 \times 10^{-2} \text{kg.m}^2$, $J_3 = 16.66 \times 10^{-2} \text{kg.m}^2$. The initial angular velocity is $\omega(0) = [0.45; 0.52; 0.55]^T \text{rad/s}$. The thrusters have the same thrust level $F = 10 \text{mN}$, the geometric parameters are taken as $\alpha = 30^\circ$, $x = 0.05 \text{m}$, $l = 0.15 \text{m}$. The performance metric is the total impulse $I_{tot} = \sum_{i=1}^3 \sum_{q=1}^Q u_{on/off}$ which computes how many time steps a thruster is turned on, and can be multiplied by $F dt / (I_s g_0)$ to obtain the propellant consumption, where dt is

the time step, I_s is the specific impulse of the thrusters, and g_0 the gravity acceleration on Earth. De-tumbling is considered complete when all of the angular velocities have absolute values less than 0.002 rad/s . The angular velocity is normalized in the cost function with $\omega^{norm} = [2.0; 2.0; 2.0] \text{ rad/s}$. The Neural Network used for propagation of the state uses 50 neurons in its hidden layer, and 3 delays for the activation thrust vector and for the delayed states, resulting in 21 inputs.

In Figure 5, we show the results with a typical quadratic cost function ($K_\infty = 0$), a cost function with an infinity norm ($K_{quad} = 0$), and a linear combination of the two. The simple logic control (4) leads to a total

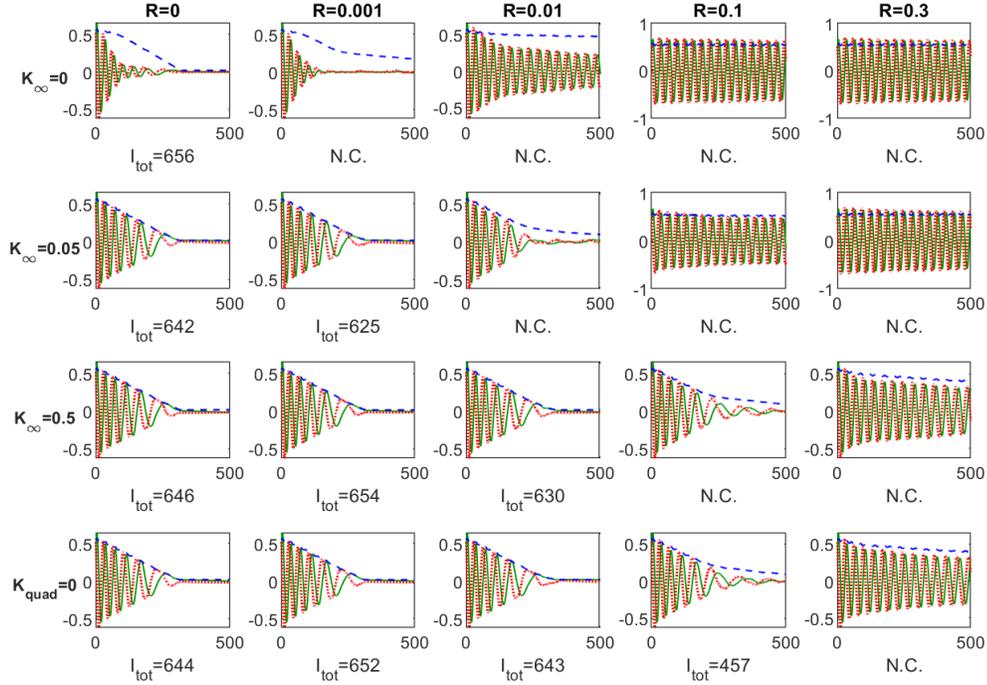


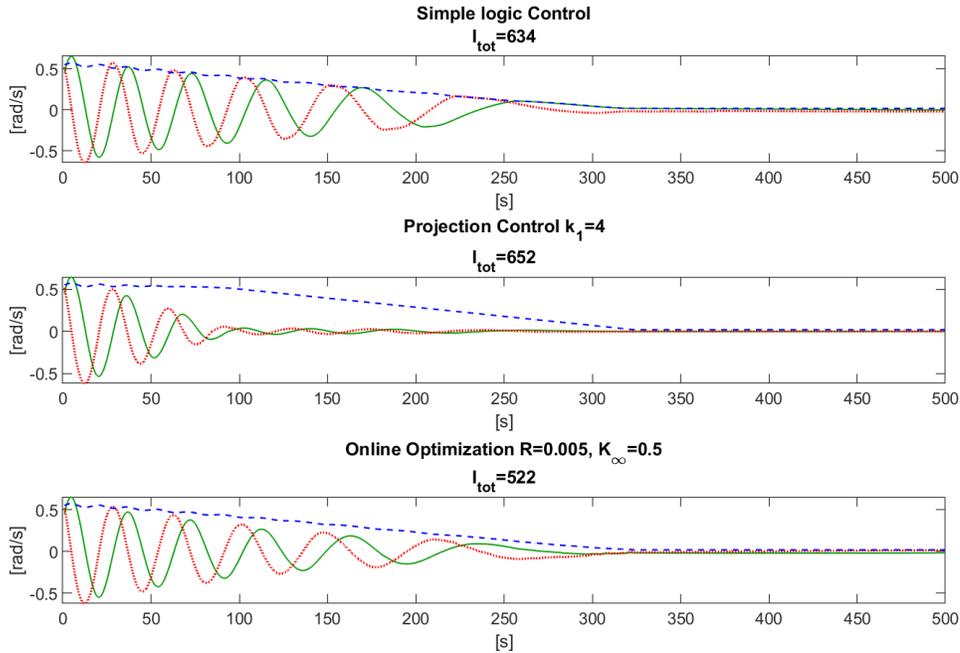
Fig. 5 De-tumbling of a 12U CubeSat with an optimal predictive control using the cost function $F = K_{quad} \|\omega\|_2^2 + K_\infty \|\omega\|_\infty^2 + R \|\mathbf{u}\|_2^2$. (N.C.=not converged)

impulse of 634, and the projection control (9), after selecting the best coefficient $k_1 = 4$, leads to a total impulse of 652. In the case of the predictive control the best impulse of 625 was achieved which offers only a small improvement to the simpler controls. However, using the modified cost function with a square value of the quadratic norm and a linear value of the infinity norm yields the results in Table 1. This shows a clear improvement in the total impulse reduction with approximately the same settling time.

In particular for values of $R = 0.005$ and $K_\infty = 0.5$ a total impulse of 522 is achieved leading to a 17.7% reduction in the fuel requirement. In Figure 6 the simulations using the on-line optimization method, the simple logic method and the projection control are compared. This shows a clear improvement of the de-tumbling cost by using the on-line optimization NNPC.

Table 1 Total Impulses for detumbling a 12U CubeSat with 4 thrusters, with an optimal predictive control using a modified cost function $F = \|\omega\|_2^2 + K_\infty \|\omega\|_\infty + R \|u\|_2^2$ (N.C.=non converged)

| | $R = 0$ | $R = 0.0003$ | $R = 0.001$ | $R = 0.003$ | $R = 0.01$ | $R = 0.1$ |
|------------------|---------|--------------|-------------|-------------|------------|-----------|
| $K_\infty=0.015$ | 637 | 643 | N.C. | N.C. | N.C. | N.C. |
| $K_\infty=0.05$ | 649 | 607 | 611 | N.C. | N.C. | N.C. |
| $K_\infty=0.15$ | 630 | 601 | 576 | N.C. | N.C. | N.C. |
| $K_\infty=0.5$ | 642 | 598 | 588 | 543 | N.C. | N.C. |
| $K_\infty=1.5$ | 641 | 633 | 636 | 604 | 550 | N.C. |
| $K_\infty=5$ | 643 | 661 | 630 | 648 | 609 | N.C. |

**Fig. 6** Detumbling a 12U CubeSat with 4 thrusters, comparison of 3 different methods

5.2 Off-line Training of a Neural Network Model Reference Control

The optimal control described so far is not computationally feasible on-line because of the necessity of performing an optimization at each time step. Another Neural Network, that takes as inputs the delayed values of the states and torques, and outputs the thrusters activation vector $u_{on/off}$ is trained using a set composed of random inputs and for each of them, the corresponding optimized torque obtained by the same optimization over a given time-horizon. Using this approach, with successful training, the process can be used to obtain optimal values on-line, without the computational cost of optimization. A Multi-Layers Perceptron is trained off-line for this purpose.

Different MLPs are trained with optimal data obtained with the genetic algorithm-based predictive optimization. Good results are obtained with 3 hidden layers. Every result presented has been obtained with 12 different Neural Network training, because the training process highly depends on the initial condition of the Network weights. Each training uses an early stopping: 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 behaviour of the neural controller in any condition. The values of $R = 0.005$ and $K_\infty = 0.5$ for the generation of the optimized data has been taken as the best in the on-line optimization. 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 2 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 table 2 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.

Table 3 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.

Table 4 shows the number of floating point operations (flops) required by the spacecraft's on-board computer for different number of neurons, in each of the 3 hidden layers of the MLP. Note that for de-tumbling, reducing the time step would lead to negligible improvements.

Table 3 Performance (generalization error, total impulse 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 | - |
| I_{tot} | 661 | 580 | 522 |
| Δt_s | 349s | 320s | 321s |

Table 4 Number of floating points operations required by the spacecraft's onboard computer for 3 hidden layers MLP controllers for different numbers of neurons

| S | flops |
|----|-------|
| 10 | 1070 |
| 20 | 2920 |
| 30 | 5570 |

6 Conclusion

This paper presents a Neural Network-based 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 neural controls, a logic-based control law and a control obtained by projection of a continuous control on the set of available torques is performed. It leads to the conclusion that the NNPCs are more fuel-efficient for de-tumbling maneuvers, leading to up to 17.7% fuel savings when compared to the logic-based and projective controller. Future work can include the development of optimal controls for slew motions and attitude tracking for spacecraft with four thrusters which could include micro-electric propulsion technology. In addition, a rigorous stability analysis is still required; with the logic based controls stability determined using passivity arguments while stability of a neuro-based controller remains a significant challenge.

References

1. 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)
2. 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)
3. Wie, B.: *Space Vehicle Dynamics and Control*, Second Edition. AIAA, pp. 450-457, (2008)
4. 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)
5. Nehrenz, M., and Sorgenfrei, M.: A Comparison of Thruster Implementation Strategies for a Deep Space Nanosatellite. AIAA Guidance, Navigation, and Control Conference, (2015)
6. 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)
7. 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)
8. 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)
9. Eren, U., Prach, A., Koer, B. B., Rakovic, S. V., Kayacan, E., Aikmese, 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)
10. 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)
11. 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)
12. Fornasini, E., Valcher, M. E.: Optimal Control of Boolean Control Networks, *IEEE Transactions on Automatic Control*, vol. 59, No. 5, pp. 1258-1270, (2014)