S. Silvestrini, M. Lavagna
Neural-Aided GNC reconfiguration algorithm for distributed space system: development and PIL Test
Advances in Space Research, In press - Published online 11/01/2021
doi:10.1016/j.asr.2020.12.014

Neural-aided GNC Reconfiguration Algorithm for Distributed Space System: Development and PIL test

Stefano Silvestrini[a,1,*], Michèle Lavagna[a,2]

[a] *Politecnico di Milano, Via La Masa 34, 20156, Milano*

Abstract

This paper presents a neural-aided Guidance, Navigation & Control algorithm for reconfiguration of distributed space systems. The guidance algorithm is based on Artificial Potential Fields (APF) in the Relative Orbital Elements (ROE) space. Since the relative orbit determination measurements are typically referred to the Cartesian metrics (e.g. range or range rate), a linear mapping between the set of ROE and the Cartesian coordinates expressed in the Local-Vertical-Local-Horizontal (LVLH) reference frame is derived. The navigation and control algorithms rely on the relative dynamics expressed in the same ROE set of coordinates. To cope with uncertainties and nonlinearities of the system, a Radial Basis Function Neural Network (RBFNN) is employed to reconstruct the perturbed dynamics. The Artificial Neural Network (ANN) is coupled with an adaptive Extended Kalman Filter for state estimation. A feedback control is designed to track the desired state, whose stability is analyzed using Lyapunov theory. The guidance, navigation and control algorithms are tested in a high-fidelity numerical orbit propagator. Moreover, the algorithm is tested in relevant Processor-In-the-Loop (PIL) simulations using a TI C2000-Delfino MCU F28379D. The results demonstrate the effectiveness of the algorithm for relative reconfiguration maneuvers involving relative distances $\sim 10^2$

*Corresponding author

[1] PhD Candidate, Department of Aerospace Science and Technology, stefano.silvestrini@polimi.it

[2] Full Professor, Department of Aerospace Science and Technology, michelle.lavagna@polimi.it

m with limited fuel consumption and constrained available thrust ($\leq 1mN$). In particular along-track maneuvers, relative plane change and formation enlargement are analysed in the paper, showing the comparison between the proposed algorithm and the legacy one without the neural network. The benefit of implementing a neural network is particularly highlighted when the nonlinearities or unmodelled terms in the on-board dynamics become prominent.

Keywords: GNC, ANN, spacecraft, distributed, formation flying, PIL test

---

1. Introduction

The concept of distributed space systems, composed of several micro-satellites flying in formation, is becoming increasingly attractive due to the increased robustness of the resultant mission architecture. Moreover, it is claimed that better performance, coupled with a cost and time-to-flight reduction, can be achieved. Different mission objectives can be achieved throughout the mission by reconfiguring the satellite formation. A high-level of autonomy, and consequent complexity, is required in such mission concept. Indeed, the satellites are expected to react autonomously to unforeseen events. In particular, the collision avoidance task is critical in formation reconfiguration especially when the number of satellites increases. The Guidance Navigation & Control (GNC) algorithms can be implemented following a centralized (Sarno et al., 2020), decentralized or distributed architecture (Silvestrini et al., 2019). On one hand, the centralized architecture presents two different issues: first, it presents a single failure point due to the presence of a master spacecraft; furthermore, the GNC commands are sent to all the agents of the system, inserting complexity on the communication link between the master and the other spacecrafts. On the other hand, the decentralized approach solves the failure point shortcoming but it lacks of a system-level perception, as each satellite is limited to its own data. The distributed architecture is selected to cope with the aforementioned shortcomings of the centralized and decentralized approaches. Even though formation flying missions are not so numerous in literature (Di Mauro et al., 2018;

Chu, 2015), the path-planning is often tackled as an optimum problem: the optimal trajectory of each satellite is found taking into account the collision hazard constraint. Both centralized and distributed architecture have been studied by Chu (2015). Nevertheless, the computational burden is high and difficult to handle, especially when on-board computational resources are very limited. Mixed Linear Programming and Particle Swarm Optimization were proposed by Di Mauro et al. (2018) to solve an optimal continuous control law for satellite reconfiguration. Such approach is hardly fitting the constraints imposed by the micro-platforms and does not take into account any collision avoidance strategy. A strategy for convexifying the collision constraint in optimal control has been reported by Chu (2015). Chernick and D'Amico (2016) presented an optimal control based on impulsive maneuver leveraging Keplerian dynamics to determine optimal and predictable maneuvering schemes, without taking into account the collision avoidance constraint. An impulsive strategy based on the state transition matrix of the system is also presented by Vadali and Alfriend (2013). Several authors have partially solved the task of collision-free path-planning using behaviour-based algorithms. Izzo and Pettazzi (2005) presented a behaviour-based algorithm, where the guidance desired velocity is determined as a result of the summation of identified behaviour (target approach, collision avoidance, etc.). Similar to the behaviour-based approach is the calculation of the Artificial Potential Field (APF) (Li et al., 2018). The major difference is that APF outputs desired acceleration, contrary to the behaviour-based that works with guidance velocities. Steindorf et al. (2017) proposed a guidance and control approach based on the Artificial Potential Field using relative orbital elements. The authors include a passive collision avoidance strategy applicable to satellite formations composed of two satellites. In general, the literature is still poor with respect to algorithms that can be implemented in a distributed architecture with low computational power, actively managing the collision avoidance constraint between more than two satellites.

The relative motion between spacecrafts flying in formation is typically reconstructed by state measurement expressed in Cartesian coordinates in the

Hill frame. Hence, in this paper, a linear mapping is developed to transform the Cartesian state to the relative orbital elements $\delta\chi$, and vice versa. Schaub and Alfriend (2002) used a similar approach adopting the mean orbital elements difference.

This paper extends the APF algorithm in ROE space to a distributed architecture with more than two satellites. In particular, mutual repulsion for each satellite pair is calculated to prevent collisions. Differently from Steindorf et al. (2017), the repulsive term is derived directly from Cartesian measurements, which is more compliant with a realistic scenario.

The Guidance, Navigation & Control algorithms are highly influenced by the dynamical models that are implemented on-board. For the sake of limiting computational burden, often these dynamical models are linearized or they intentionally neglect disturbance terms, which might play a role in the motion evolution. Such limitation leads to a degradation of performance of the GNC subsystem. This paper introduces a light and flexible algorithm based on Artificial Neural Networks to be able to estimate the unmodelled terms efficiently. In this way, the GNC system relies on accurate on-board dynamics without increasing the computational burden. Several researchers have utilized Artificial Neural Networks universal approximation theorem to work out the task of estimating disturbances and unmodelled terms (Pesce et al., 2020). Nevertheless, most of the algorithms work off-line using large amount of data to be trained on. The neural reconstruction of disturbances/unmodelled terms have proven the possibility to enhance the guidance and control for space mission (Gurfil et al., 2003; Bae and Kim, 2012; Zhou et al., 2018). However, an online strategy to reconstruct the dynamics, which can be used in the whole GNC chain, is still unexplored in literature. Indeed, most of the researchers do not refine the on-board dynamical model based on the neural network reconstruction.

To summarize, this paper presents an autonomous formation reconfiguration GNC algorithm based on Neural Network - Artificial Potential Field. It includes a distributed active collision avoidance based on repulsive potential contribution derived from Cartesian state measurements, suitable for microsatellite applica-

tions. A tracking feedback controller, based on Lyapunov theorem, guarantees the artificial potential dynamics to be followed. Additionally, the proposed algorithm exploits an online Radial-Basis Function Neural Network (RBFNN) to reconstruct the disturbances or unmodelled terms in the dynamics to enhance the whole Guidance Navigation & Control (GNC) system performance. Finally, the paper presents the Processor-in-The-Loop (PIL) validation performed with relevant hardware. Such testing procedure increases the Technology Readiness Level (TRL) of the algorithm with respect to Model-In-the-Loop (MIL) simulations. The intended contributions of the paper are:

- to develop a fully online algorithm based on Radial-Basis-Function Neural Network (RBFNN) for dynamics reconstruction that can benefit the whole GNC architecture. The refinement of the on-board dynamics improves the navigation performance and guarantees better control accuracy to reach the target state;

- to develop a GNC algorithm that exploits the RBFNN refinement of the on-board dynamics while flying. The paper presents a methodology that could potentially be exploited in several different environments. The disturbances and nonlinear terms may be caused by different sources. In this paper, a fully nonlinear Cartesian J2-perturbed with relative drag is used for the *ground-truth*, whereas the on-board dynamics used in the GNC is based on ROE dynamics.

- to propose a full GNC algorithm for spacecrafts formation reconfiguration, suitable for micro platforms implementation;

- to extend the application of the Artificial Potential Field to dynamically control multiple agents (≥ 2) assuring the collision avoidance constraint is respected;

- to develop a collision avoidance procedure that takes as input Cartesian relative measurements and process them in the Relative Orbital Element space

- to develop a Functional Engineering Simulator to test and validate the full GNC algorithm (Model-In-the-Loop MIL);
- to test the algorithms in relevant Processor-In-the-Loop (PIL) simulations with limited computational power to increase the TRL. The PIL validation is typically not considered in research papers. The TRL of the algorithm after PIL validation is increased with respect to simpler MIL validation. The computational power of the MCU is compatible with microsatellites on-board resources.

The paper is structured as follows: the dynamical model employed and the coordinates transformation are presented in Section 2; in Section 3 the Artificial Potential Field (APF) based on Relative Orbital Elements (ROE) is presented; in Section 4 the neural-aided navigation and control is presented; in Section 5 and 6 the numerical and PIL simulations results are presented; last, in Section 7 conclusions are drawn.

2. Dynamical Model

The spacecraft formation dynamics is described using the relative orbital elements, following the work done by D'Amico (2010). The following quasi-singular relative orbital elements are adopted:

$$\delta\chi = \begin{pmatrix} \delta a \\ \delta\lambda \\ \delta e_x \\ \delta e_y \\ \delta i_x \\ \delta i_y \end{pmatrix} = \begin{pmatrix} \frac{a_f - a_r}{a_r} \\ (M_f + \omega_f) - (M_r + \omega_r) + (\Omega_f - \Omega_r)\cos(i_r) \\ e_f \cos(\omega_f) - e_r \cos(\omega_r) \\ e_f \sin(\omega_f) - e_r \sin(\omega_r) \\ i_f - i_r \\ (\Omega_f - \Omega_r)\sin(i_r) \end{pmatrix} \tag{1}$$

where the subscript f stands for any follower spacecraft orbit, whereas the subscript r indicates the reference orbital elements. M is the mean anomaly, a the semimajor axis, e the eccentricity, i the orbit inclination, ω the argument of perigee and Ω the right ascension of the ascending node. It is important to

remark that in this paper the reference orbit is the same for the n spacecraft building up the formation. The benefit of using such model is that, if the perturbations are neglected, the geometry of the relative motion with respect to a reference orbit is uniquely determined by a set of invariant relative orbital elements (ROE), except for the relative true anomaly, which follows the Keplerian propagation. Indeed, the natural evolution of the dynamic system can be described as:

$$\dot{\delta\chi} = \mathbf{A_k} \cdot \delta\chi \tag{2}$$

where

$$\mathbf{A_k} = \begin{bmatrix} 0 & \vdots & \\ -1.5n & \vdots & \mathbf{0}_{6\times5} \\ \mathbf{0}_{4\times1} & \vdots & \end{bmatrix} \tag{3}$$

Guffanti et al. (2017) and Koenig et al. (2017) later expanded the model to a J_2 perturbed dynamics. The complete dynamical model can be expressed as:

$$\dot{\delta\chi} = (\mathbf{A_k} + \mathbf{A_{J2}}) \cdot \delta\chi + B\mathbf{u} \tag{4}$$

$$\mathbf{A_{J_2}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{7}{2}(1+\eta)(3\cos^2 i_r - 1) & 0 & e_x GFP & e_y GFP & -FS & 0 \\ \frac{7}{2}e_y Q & 0 & -4e_x e_y GQ & -(1+4Ge_y^2)Q & 5e_y S & 0 \\ -\frac{7}{2}e_x Q & 0 & (1+4Ge_x^2)Q & 4e_x e_y GQ & -5e_x S & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{7}{2}S & 0 & -4e_x GS & -4e_y GS & 2T & 0 \end{bmatrix} \tag{5}$$

where the terms in Eq. 5 are:

$$\begin{aligned} k = \gamma a_r^{-\frac{7}{2}} \eta^{-4},\ \eta = \sqrt{1-e_r^2},\ \gamma = \frac{3}{4} J_2 R_e^2 \sqrt{\mu},\ e_x = e_r \cos\omega_r, \\ e_y = e_r \sin\omega_r\ E = 1+\eta,\ G = \frac{1}{\eta^2},\ F = 4+3\eta,\ P = 3\cos^2 i_r - 1, \\ Q = 5\cos^2 i_r - 1,\ S = \sin 2i_r,\ T = \sin^2 i_r \end{aligned} \tag{6}$$

where $J2$ is the zonal harmonic coefficient $1.0826 \cdot 10^{-3}$ for Earth, R_e is the Earth radius, $\mu = 3.986 \cdot 10^{14} m^3 s^{-2}$ is the Earth gravitational constant. The control input for the model is defined by the accelerations in radial, along-track and normal direction of the LVLH orbital frame, derived from the inertial Gauss Variational Equations (GVE). The control matrix is derived from Gauss Variational Equation (GVE) as in Schaub et al. (2000):

$$\mathbf{B} = \frac{1}{a_r n_r} \begin{bmatrix} \frac{2}{\eta} e_r \sin\nu_r & \frac{2}{\eta}(1 + e_r \cos\nu_r) & 0 \\ -\frac{2\eta^2}{1+e_r\cos\nu_r} & 0 & 0 \\ \eta_r \sin\omega_r + \nu_r & \eta\frac{(2+e_r\cos\nu_r)\cos(\omega_r+\nu_r)+e_x}{1+e_r\cos\nu_r} & \frac{\eta e_y}{\tan i_r}\frac{\sin(\omega_r+\nu_r)}{1+e_r\cos\nu_r} \\ -\eta_r \cos\omega_r + \nu_r & \eta\frac{(2+e_r\cos\nu_r)\sin(\omega_r+\nu_r)+e_y}{1+e_r\cos\nu_r} & \frac{-\eta e_x}{\tan i_r}\frac{\sin(\omega_r+\nu_r)}{1+e_r\cos\nu_r} \\ 0 & 0 & \eta\frac{\cos(\omega_r+\nu_r)}{1+e_r\cos\nu_r} \\ 0 & 0 & \eta\frac{\sin(\omega_r+\nu_r)}{1+e_r\cos\nu_r} \end{bmatrix} \tag{7}$$

where ν_r is the true anomaly.

2.1. Coordinates Transformation

The active collision avoidance maneuvers depend on the relative metric distance between two agents. The relative distance is naturally expressed in the Cartesian Local-Vertical-Local-Horizontal (LVLH) reference frame. The mapping between the Hill $\mathbf{X} = [x\ y\ z\ \dot{x}\ \dot{y}\ \dot{z}]$ state to the ROE $\delta\chi$ is required to process the measurements and compute the guidance and control output. The transformation matrices are derived by using the classical orbital elements difference $\Delta OE = [\Delta a\ \Delta M\ \Delta\omega\ \Delta e\ \Delta i\ \Delta\Omega]$ as follows:

$$\mathbf{J}^{\mathbf{X}}_{\delta\chi} = \frac{\partial \mathbf{X}}{\partial \Delta OE} \cdot \frac{\partial \Delta OE}{\partial \delta\chi}, \qquad \mathbf{J}^{\delta\chi}_{\mathbf{X}} = \frac{\partial \delta\chi}{\partial \Delta OE} \cdot \frac{\partial \Delta OE}{\partial \mathbf{X}} \tag{8}$$

where a is the semimajor axis, M is the mean anomaly, ω the argument of perigee, e the eccentricity, i the inclination and Ω the right ascension of the ascending node. The first-order approximation of the mapping between the Hill state and classical osculating orbital elements yields (D'Amico, 2010; Lane and

Axelrad, 2006):

$$
\begin{aligned}
x &= \frac{r}{a}\Delta a - a\cdot\cos\nu\Delta e + \frac{ae\sin\nu}{\sqrt{1-e^2}}\Delta M \\
y &= \Big(a + \frac{r}{1-e^2}\Big)\sin\nu\Delta e + \frac{a^2}{r}\eta\Delta M + r\Delta\omega + r\cos i\Delta\Omega \\
z &= r\sin(\nu+\omega)\Delta i - r\sin i\cos(\nu+\omega)\Delta\Omega
\end{aligned}
\tag{9}
$$

Differentiating Eq. 9 the full transformation is obtained:

$$
\begin{aligned}
\dot{x} =& -\frac{ne\sin\nu}{2\sqrt{1-e^2}}\Delta a + n\sin\nu\sqrt{1-e^2}\Big(\frac{a^3}{r^2}\Big)\Delta e + en\cos\nu\frac{a^3}{r^2}\Delta M \\
\dot{y} =& \left[n\sqrt{1-e^2}\Big(1+\frac{r}{a(1-e^2)}\Big)\Big(\frac{a^3}{r^2}\Big)\cos\nu + \frac{aen\sin^2\nu}{(1-e^2)^{\frac{3}{2}}}\right]\Delta e + \\
& - en\sin\nu\frac{a^3}{r^2}\Delta M + \frac{aen\sin\nu}{\sqrt{1-e^2}}\Delta\omega \\
\dot{z} =& \frac{an}{\sqrt{1-e^2}}\Big(\sin i\Big[\sin(\nu+\omega)+e\sin\omega\Big]\Delta\Omega + \\
& + \Big[\cos(\nu+\omega)+e\cos\omega\Big]\Delta i\Big)
\end{aligned}
\tag{10}
$$

Combining Eq. 9 and 10, the transformation matrices between Hill state $\mathbf{X}$ and classical orbital elements ΔOE, namely $\frac{\partial\mathbf{X}}{\partial\Delta OE}$ and its inverse in Eq. 8, are found. To formulate the complete transformation, the Jacobian of the transformation between classical orbital elements and relative orbital elements $\delta\chi$ is required.

Such transformation is obtained from the definition of $\delta\chi$ for $\Delta OE \to 0$:

$$\frac{\partial \Delta OE}{\partial \delta\chi} = \begin{bmatrix} a & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & \frac{\sin\omega}{e} & -\frac{\cos\omega}{e} & 0 & \frac{\cos i}{\sin i} \\ 0 & 0 & -\frac{\sin\omega}{e} & \frac{\cos\omega}{e} & 0 & 0 \\ 0 & 0 & \cos\omega & \sin\omega & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sin i \end{bmatrix}$$

$$\frac{\partial \delta\chi}{\partial \Delta OE} = \begin{bmatrix} \frac{1}{a} & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & \cos i \\ 0 & 0 & -e\sin\omega & \cos\omega & 0 & 0 \\ 0 & 0 & e\cos\omega & \sin\omega & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sin i \end{bmatrix} \tag{11}$$

2.2. Bounded Orbits

This paper is focused on the presentation of a GNC architecture for spacecraft formation flying. For this reason, the natural motion is not deeply analyzed. Nevertheless, realistic formation are taken into account to generate initial and final configurations. In order to reduce the fuel consumption for formation-keeping actions, the configurations are chosen among the bounded relative orbits group. To avoid the relative drift, it is critical that the relative motion of the spacecrafts remains bounded. A fundamental concept in spacecraft Formation Flying is the orbital energy-matching method to generate bounded formations. The orbital energy of the satellites is a function of the semi-major axis only:

$$\mathcal{E} = -\frac{\mu}{2a} \tag{12}$$

Hence, it is sufficient to match the orbital energy of the reference orbit to generate bounded formations. In order to work out relevant initial conditions, being either in the Cartesian space or $\delta\chi$, we refer to the $\delta\chi$ relative space, cfr. Eq. 1. Clearly, it is sufficient that $\delta a = 0$ for the relative orbital elements defining the

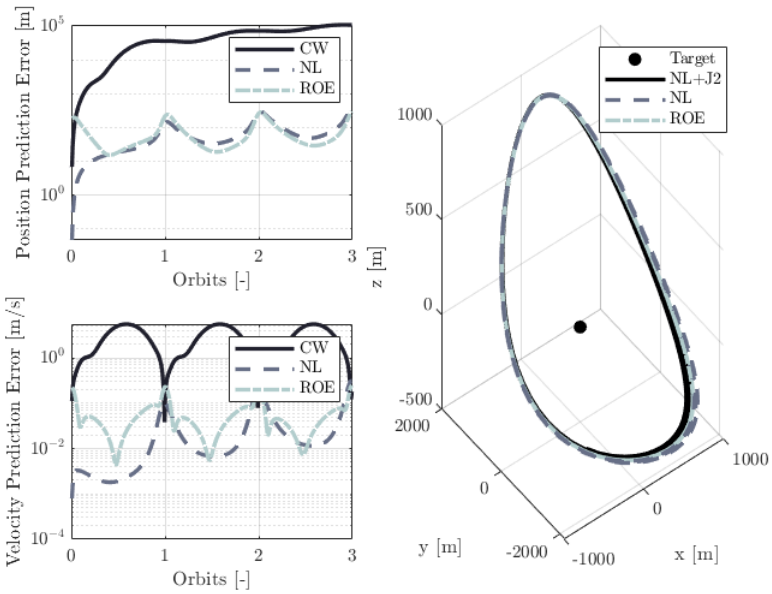

Figure 1: Relative bounded orbits in perturbed models. NL stands for nonlinear model, ROE for relative orbital elements dynamics.

formation to match the orbital energies. Fig. 1 shows the trajectories propagation based on energy-matching initial conditions in perturbed models.

3. Artificial Potential on ROE space

The distributed guidance algorithm processes locally all the state estimations of the satellite formation members. Each satellite knows the relative position with respect to the rest of the formation. The guidance strategy relies on artificial potential functions designed in the relative orbital elements space in $\mathbb{R}^6$ (Steindorf et al., 2017). The idea is to build a point-wise global potential based on the contribution of attractive and repulsive potential sources, namely the target relative orbits and any other satellite located in close neighbouring areas. The attractive potential is directly expressed in terms of relative orbital elements, being a convenient way to express relative orbits geometry. Indeed, a set of relative orbital elements uniquely define one particular formation con-

figuration. On the other hand, the natural way to express the vicinity between two satellites is using the Cartesian distance, expressed in the LVLH reference frame in this particular application. To obtain a uniform expression of the global potential, the Jacobian of the transformation is derived, based on the results presented in Section 2.1. The output of the guidance, for each satellite, is what we call *guidance state* and indicate as $\delta\chi_g$. The guidance algorithm forces the following dynamics for each satellite i:

$$\dot{\delta\chi_{\mathbf{g}}} = -\nabla\Phi_{glb} \tag{13}$$

where Φ_{glb} is the global potential:

$$\nabla\Phi_{glb} = \nabla\Phi_a + \nabla\Phi_r \tag{14}$$

where Φ_a is the attractive potential, whereas Φ_r is the repulsive one.

3.1. Attractive Potential: Configuration Target

The reconfiguration objective is to drive the satellites to a predefined relative configuration, expressed in terms of relative orbital elements. The set of ROE to be achieved is called *reference state* and indicated as $\delta\chi_r$. The attractive contribution to the global potential is determined as:

$$\Phi_a(\delta\chi) = \frac{1}{2}\xi_a \left\| \delta\chi_g - \delta\chi_r \right\|^2 \tag{15}$$

where the parameter ξ_a is a user-defined variable, which can be used to tune the guidance according to the scenario. The gradient in the guidance ROE space is defined as:

$$\nabla_{\delta\chi_g}(\cdot) = \left(\frac{\partial}{\partial\delta a}, \frac{\partial}{\partial\delta\lambda}, \frac{\partial}{\partial\delta e_x}, \frac{\partial}{\partial\delta e_y}, \frac{\partial}{\partial\delta i_x}, \frac{\partial}{\partial\delta i_y} \right)_g \tag{16}$$

Consequently, the dynamic contribution to Eq. 14 given by the attractive potential is:

$$\nabla_{\delta\chi_g} = \xi_a(\delta\chi_g - \delta\chi_r) \tag{17}$$

3.2. Repulsive Potential: Active Collision Avoidance

The repulsive potential is useful to calculate the trajectory in presence of other satellites, avoiding collision between agents. As previously stated, to achieve an efficient active collision avoidance maneuver, the potential is best representative in terms of the Cartesian state X in the Hill frame, where the metric distance is defined. Given two satellites, i and j respectively, the repulsive potential to be computed for Eq. 14 for satellite i is defined as:

$$\Phi_{r_{ij}} = \begin{cases} \frac{1}{2}\xi_r e^{-\frac{d_{ij}^2}{\eta}} = \frac{1}{2}\xi_r e^{-\frac{\left\|\mathbf{x_i}-\mathbf{x_j}\right\|^2}{\eta}} & \text{if } d_{ij} < d_{lim}, \\ 0 & \text{if } d_{ij} > d_{lim} \end{cases} \tag{18}$$

where d_{lim} is the threshold distance beyond which the collision maneuver is not required. The state of the relative position of the spacecrafts is known, thus it is possible to calculate the distance vector as the difference between $\mathbf{X_i} - \mathbf{X_j}$. The gradient of the potential is calculated using the chain-rule, which involves the coordinate transformation from Cartesian state $\mathbf{X}$ to ROE $\delta\chi$:

$$\nabla_{\delta\chi_g}\Phi_{r_{ij}} = \nabla_X \Phi_{r_{ij}} \cdot J_{\delta\chi}^X \tag{19}$$

where $\mathbf{J}_{\delta\chi}^{\mathbf{X}}$ is the Jacobian of the coordinate transformation, derived in Section 2.1. The gradient in the Cartesian space is defined as:

$$\nabla_X(\cdot) = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z}\right) \tag{20}$$

Hence, the gradient of the repulsive potential between agents i and j, below the threshold, can be expressed as:

$$\nabla_{\delta\chi_g}\Phi_{r_{ij}} = -\frac{\xi_r}{\eta} e^{-\frac{d_{ij}^2}{\eta}} \cdot (\mathbf{X_i} - \mathbf{X_j}) \cdot J_{\delta\chi}^X \tag{21}$$

The repulsive potential takes into account all the mutual distances between the formation agents; coherently, the repulsive contribution to the global potential for satellite i is the summation of the mutual repulsive potential between satellite i and all the other satellites:

$$\Phi_r = \sum_{j \neq i}^{n} \Phi_{r_{ij}} \tag{22}$$

where n is the number of spacecrafts in the formation.

3.3. Natural Dynamics: Action Smoothing

The described derivation of the global artificial potential does not take into account the natural dynamics of the system, even though the artificial potential is derived in the ROE space. By including the natural dynamics of Eq. 4 into Eq. 13, the guidance law is smoothed with respect to the global potential. Hence, the terms expressing the natural dynamics can be integrated to smooth the effect of the guidance law resulting in:

$$\dot{\delta\chi_{\mathbf{g}}} = -\nabla\Phi_{glb} + (\mathbf{A_k} + \mathbf{A_{J2}}) \cdot \delta\chi \tag{23}$$

Thus, the guidance dynamics evolves according to Eq. 23. Such dynamics is used to generate the desired ROE state, which is subsequently used to determine the control action to steer the actual trajectory of the spacecraft. The natural dynamics becomes dominant when the spacecraft is very close to the target configuration. This eliminates the oscillatory chattering caused by the artificial potential when very close to the target state.

4. Radial-basis Function Neural Network approximation

An artificial neural network is employed to approximate the dynamical terms encompassing all the unmodelled nonlinearities and disturbances. The paper presents a methodology that could potentially be exploited in several different environments. The disturbances and nonlinear terms may be caused by different sources, such as gravity harmonics, solar radiation pressure, drag, etc. In this paper a fully nonlinear Cartesian J2-perturbed with relative drag is used for the *ground-truth*, whereas the *on-board* dynamics used in the GNC is based on ROE dynamics. The ROE dynamics for unperturbed motion is equal to the Keplerian matrix in Eq. 4. If the difference in the semimajor axis is null, the Keplerian dynamics in ROE space vanishes. This makes the perturbations relevant even if the magnitude is very low. For instance, the J2 perturbation term in the ROE dynamics is $\sim 10^{-10}s^{-}1$.

The universal approximation theorem of artificial neural networks guarantees the existence of a set of ideal weights $\mathbf{W}$ that approximates a function with

a bounded arbitrary approximation error (Wu et al., 2012). A Radial-Basis-Function Neural Network is a single-layer shallow network, whose neurons are Gaussian functions. As detailed in Pesce et al. (2020), such network architecture possesses a quick learning process, which makes it suitable for online dynamics identification and reconstruction. The highlights of the mathematical expression of the RBFNN are reported here for clarity. For a generic state input $\delta\chi \in \mathbb{R}^n$, the components of the output vector $\gamma \in \mathbb{R}^j$ of the network is:

$$\gamma_l(\delta\chi) = \sum_{i=1}^{m} w_{il}\Phi_i(\delta\chi) \tag{24}$$

In a compact form, the output of the network can be expressed as:

$$\gamma(\delta\chi) = \mathbf{W}^T\mathbf{\Phi}(\delta\chi) \tag{25}$$

where $\mathbf{W} = [w_{il}]$ for $i = 1, ..., m$ and $l = 1, ..., j$ is the trained weight matrix and $\mathbf{\Phi}(\mathbf{x}) = [\Phi_1(\mathbf{x})\ \Phi_2(\mathbf{x})\ \cdots\ \Phi_m(\mathbf{x})]^T$ is the vector containing the output of the radial basis functions, evaluated at the current system state.

The online learning algorithm is derived using Lyapunov stability theorem, as in Pesce et al. (2020). The RBFNN weights matrix is updated as follows:

$$\dot{\hat{\mathbf{W}}} = \frac{\eta}{\xi}\mathbf{\Phi}(\hat{\delta\chi})\mathbf{e}^T \tag{26}$$

where $\mathbf{e} = \delta\chi - \hat{\delta\chi}$. The latter term is the estimated state output of the navigation filter, as explained in the following section.

4.1. Neural Navigation

The navigation filter is a Radial-Basis-Function Neural Network (RBFNN) Adaptive-Extended Kalman Filter (AEKF). The derivation and mathematical proof for stability and convergence has been reported in Pesce et al. (2020). The navigation algorithm scheme is depicted in Fig. 2. The formulation is basically an Extended Kalman Filter coupled with a Radial-Basis-Function Neural Network. The neural network complements the dynamics with an additional acceleration term, which contains all the nonlinearities or unmodelled terms

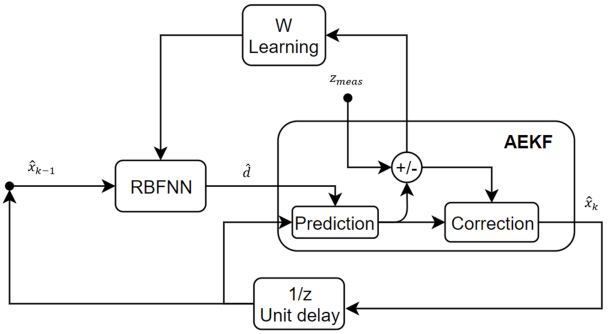

Figure 2: Overview of adopted navigation filter scheme (Pesce et al., 2020).

that are missing in the dynamical model used in the EKF propagation. For instance, for linear dynamics neglecting perturbations, the neural network would reconstruct an additional term that takes into account unmodelled harmonics, drag accelerations and nonlinear terms. The adaptivity is required to update the process covariance matrix as the neural-disturbance term is refined. The prediction is based on the augmented dynamics taking into account the contribution of the RBFNN. The measurements are the Cartesian state in the LVLH frame of the reference satellite. In this case, according to Pesce et al. (2020), the formulation of the RBFNN-AEKF is given by:

$$\hat{\delta\chi}_{\mathbf{k}}^{-} = \mathbf{A_k}\hat{\delta\chi}_{\mathbf{k-1}}^{-} + \gamma(\hat{\delta\chi}_{\mathbf{k-1}}^{-}, \mathbf{u_{k-1}}) \tag{27}$$

$$\mathbf{P}_{\mathbf{k}}^{-} = \tilde{\mathbf{F}}_{\mathbf{k-1}}\mathbf{P}_{\mathbf{k-1}}^{+}\tilde{\mathbf{F}}_{\mathbf{k-1}}^{\mathbf{T}} + \mathbf{Q_{k-1}} \tag{28}$$

$$\mathbf{K_k} = \mathbf{P}_{\mathbf{k}}^{-}\mathbf{H}^{\mathbf{T}}(\mathbf{H}\mathbf{P}_{\mathbf{k}}^{-}\mathbf{H}^{\mathbf{T}} + \mathbf{R_k})^{-1} \tag{29}$$

$$\mathbf{P}_{\mathbf{k}}^{+} = (\mathbf{I} - \mathbf{K_k}\mathbf{H})\mathbf{P}_{\mathbf{k}}^{-}(\mathbf{I} - \mathbf{K_k}\mathbf{H})^{\mathbf{T}} + \mathbf{K_k}\mathbf{R_k}\mathbf{K}_{\mathbf{k}}^{\mathbf{T}} \tag{30}$$

$$\hat{\delta\chi}_{\mathbf{k}}^{+} = \hat{\delta\chi}_{\mathbf{k}}^{-} + \mathbf{K_k}(\mathbf{z_k} - \mathbf{H}\hat{\delta\chi}_{\mathbf{k}}^{-}) \tag{31}$$

with

$$\tilde{\mathbf{F}} = \mathbf{A_k} + \left.\frac{\partial d}{\partial \delta\chi}\right|_{\hat{\delta\chi}_k^-}; \qquad \mathbf{H} = \mathbf{J}_{\delta\chi}^{\mathbf{X}} \tag{32}$$

and

$$\mathbf{Q_k} = \alpha\mathbf{Q_{k-1}} + (1-\alpha)(\mathbf{K_k}\delta_{\mathbf{k}}\delta_{\mathbf{k}}^{\mathbf{T}}\mathbf{K}_{\mathbf{k}}^{\mathbf{T}}) \tag{33}$$

being $\mathbf{A}$ the dynamics matrix, γ the output of the network representing the approximation of all the unmodelled terms (relative drag and nonlinearities), $\mathbf{P}_k$ the estimation error covariance, $\mathbf{Q}$ the process covariance matrix, $\mathbf{H}$ the measurement matrix, $\mathbf{R}$ the measurements covariance matrix, $\mathbf{K}_k$ the Kalman gain, α is a forgetting factor, $\mathbf{z_k}$ the measurements and $\delta_\mathbf{k} = \mathbf{z_k} - \mathbf{H}\hat{\delta\chi}_\mathbf{k}^-$ is the filter innovation. The Jacobian in Eq. 32 of the vector-valued function reconstructed by the RBFNN is derived from Eq. 25:

$$\frac{\partial \gamma}{\partial \delta\chi} = \frac{\partial \mathbf{W}^T \mathbf{\Phi}(\mathbf{x})}{\partial \mathbf{x}} = \mathbf{W}^T \frac{\partial \mathbf{\Phi}(\mathbf{x})}{\partial \mathbf{x}} \in \mathbb{R}^{n \times n} \tag{34}$$

In this formulation, the state is assumed to be completely observable. In the simulations, the Gaussian noise is used to corrupt the measurements to be representative of actual sensors output. The noise affects only the EKF update.

4.2. Neural Control

The output of the guidance algorithm is a set of ROE, which may differ from the target reference ones. To guarantee that the forced guidance dynamics in Eq. 23 is followed, a feedback control law is employed. The control law is derived using the Lyapunov stability theorem. In the distributed architecture, each spacecraft processes the guidance law, including state of the other agents. The reference signal to track is calculated by the guidance algorithm and follows the dynamics in Eq. 23. The current error between the desired guidance state and true state for each satellite is:

$$\mathbf{e}_{\delta\chi} = \delta\chi_g - \delta\chi \tag{35}$$

its temporal evolution can be described as:

$$\dot{\mathbf{e}}_{\delta\chi} = \dot{\delta\chi}_g - \dot{\delta\chi} = -\Big(\nabla\Phi_a + \nabla\Phi_r + A(\nu)\delta\chi\Big) - \Big(A(\nu)\delta\chi + \gamma(\delta\chi) + B\mathbf{u}\Big) \tag{36}$$

If we introduce the following positive semi-definite Lyapunov function:

$$V = \frac{1}{2}\mathbf{e}_{\delta\chi}^T \mathbf{e}_{\delta\chi} \rightarrow \dot{V} = \mathbf{e}_{\delta\chi}^T \dot{\mathbf{e}}_{\delta\chi} \tag{37}$$

$$\dot{V} = \Big(\delta\chi_g - \delta\chi\Big)^T \cdot \Big[- \Big(\nabla\Phi_a + \nabla\Phi_r + \gamma(\delta\chi) + B\mathbf{u}\Big)\Big] \tag{38}$$

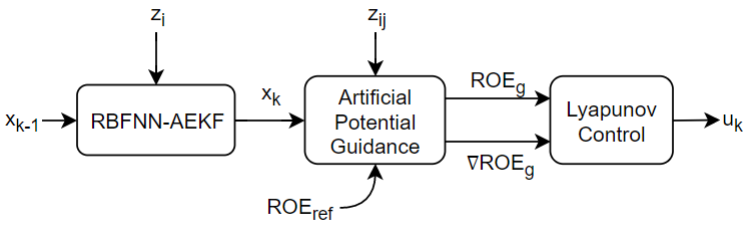

Figure 3: GNC architecture overview.

The control term can be solved to make the derivative of the Lyapunov function negative. The above strategy yields the following control law:

$$\mathbf{u} = \mathbf{B}^{-1}\left[\left(\delta\chi_g - \delta\chi\right) - \left(\nabla\Phi_a + \nabla\Phi_r\right) - \gamma(\delta\chi)\right] \tag{39}$$

In this way the derivative of the Lyapunov function is negative semidefinite, vanishing only when $\delta\chi = \delta\chi_r$, which is within the validity of the Lyapunov theorem. This approach is similar to the one adopted by Steindorf et al. (2017), with the exception of including the gradient of artificial potential and the RBFNN output in the control law. By including the gradient of the potential, which forces the dynamics, the control law calculates the action taking into account the derivative of the $\delta\chi_g$ determined by the guidance algorithm. The GNC architecture is reported schematically in Fig. 3.

Algorithm 1 Neural Distributed GNC for satellite i

1: Initialize RBFNN network $\mathcal{N}_{RBF}$
2: **while** $\delta\chi \neq \delta\chi_r$ **do**
3: Observe state $\delta\chi_i$ through relative measurements
4: Feed $\delta\chi$ to $\mathcal{N}$ (Eq. 25)
5: Estimate system state through AEKF (Eq. 31)
6: Generate guidance path using APF approach (Eq. 13)
7: Execute RBFNN-aided Lyapunov control (Eq. 39)
8: **end while**

5. Numerical Simulations

Three reconfiguration scenarios are treated in the following section involving a formation of four satellites orbiting the Earth. The scenarios are selected to be representative of different reconfiguration maneuvers that may occur during a formation flying mission. The simulations are provided to testify the efficiency of the proposed NN-APF reconfiguration algorithm. The system model parameters are chosen and described in a previous work by Silvestrini et al. (2019). The RBFNN neurons are Gaussian functions, whose centres are initialized randomly. The number of neurons is 60. Such number is the results of a trade-off between RBFNN accuracy and complexity. Tab. 1 reports the orbital parameter of the reference orbits. The reference orbits are also used to generate relative measurements by adding a fictitious noise, representative of realistic sensors uncertainty (e.g. ranging, optical or Doppler). In particular, the noise level associated is described by a Gaussian distribution with standard deviation $\sigma_{pos} = 10^{-1}m$ and $\sigma_{vel} = 10^{-3}\frac{m}{s}$ for position and velocity respectively. The Gaussian noise is used to corrupt the measurements to be representative of actual sensors output. The noise affects only the EKF update. The neural network here is used to complement the dynamics with an additional acceleration term, which contains all the nonlinearities or unmodelled terms that are missing in the dynamical expression used in the EKF propagation. Spacecraft *S/C 1* evolution is not reported for analysis as it is assumed to be controlled and to follow the reference absolute orbit. In other words, it is assumed that the absolute navigation and control guarantees the nominal reference orbit to be tracked by the *S/C 1*. This is not a major limitation since the algorithm deals with relative motion with respect to the *S/C 1*. *S/C 1* could also be thought as the virtual center of the defined LVLH frame. The natural motion is propagated using the absolute dynamics of each spacecraft. The dynamical model comprises relative drag acceleration (here considering different drag coefficients $C_{d,SC1} = 1.5$ and $C_{d,SC2-3-4} = 0.8$) and major Earth spherical harmonics. Since the satellites are close to the same position, the gravitational perturbations will affect both

Table 1: Reference orbits for numerical simulations

	$\boldsymbol{a}$ [km]	$\boldsymbol{e}$ [-]	$\boldsymbol{i}$ [^{o}]	$\boldsymbol{\omega}$ [^{o}]	$\boldsymbol{\Omega}$ [^{o}]
R1	7975	0.1	10	0	0
R2	10254	0.3	10	0	0

equally and not create noticeable drift in the ROEs. The neural-aided algorithm is beneficial when the dynamics is highly uncertain. Given that the *ground-truth* model is Cartesian and the on-board dynamics is based on ROE, it is difficult to compare the disturbance terms. For Cartesian models, the validation was performed in a previous work by the author (Pesce et al., 2020). Nevertheless, a dedicated simulation has been performed to show and report the reconstruction capabilities of the neural network in the next section.

5.1. Disturbance Reconstruction

The RBFNN is used to reconstruct the dynamical terms, which are not directly included in the on-board dynamics representation. As mentioned, the method is quite general and can be applied to different unknown environments regardless of the sources of the dynamical disturbances. In this paper the *ground-truth* dynamics is a fully nonlinear Cartesian model including J2 perturbation, whereas the dynamical model used by the GNC algorithms is based on ROE. For this reason, it is difficult to show a comparison between the *true* disturbances and the *reconstructed* ones. A dedicated simulation is here reported to show the reconstruction capabilities of the RBFNN, similarly to Pesce et al. (2020).

In the reported simulation, the *ground-truth* natural dynamics follows Eq. 4, whereas the *on-board* dynamics is simply the Keplerian term A_k of Eq. 4 coupled with the RBFNN reconstructed term. Fig. 4 shows the network approximation of the disturbance term due to J2 in ROE dynamics. The root mean squared error of the approximation is $\sim 10^{-13} s^{-1}$ once the network has converged to a steady-state.

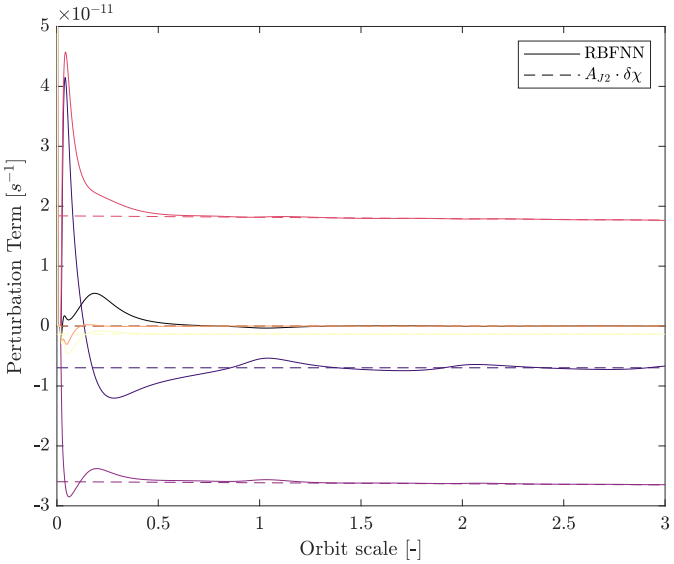

Figure 4: RBFNN network approximation of the disturbance terms due to J2 in ROE dynamics. The disturbance term is a vector $\gamma \in \mathbb{R}^6$.

5.2. Planar to Along-track

The Planar to Along-track (ALO) reconfiguration changes the relative position of the spacecraft such that they no longer belong to a symmetric configuration around the reference orbit but they are placed with a certain along-track offset. Fig. 5 shows the controlled neural reconfiguration. The initial and final state, expressed in relative orbital elements, are reported in Tab. 2. The reconfiguration is successful and the neural enhanced GNC architecture converges to stable estimation of the unmodelled terms (relative drag and nonlinearities), given that no unstable behavior is experienced, see Section 5.1. Fig. 6 shows the control effort and the navigation results in terms of estimation accuracy. The repulsive potential aided by the RBFNN network guarantees that the minimum distance between the agents is respected, as shown in Fig. 7. The relative maneuver initiates after one orbital period. The along-track reconfiguration is the most demanding in terms of collision avoidance, as the orbital state are very similar in the final configuration, roughly 100 m apart.

Table 2: Relative orbital elements of each spacecraft in the ALO reconfiguration

$a\delta\chi^a[m]$	S/C 1	S/C 2	S/C 3	S/C 4
$a\delta a$	$0 \to 0$	$0 \to 0$	$0 \to 0$	$0 \to 0$
$a\delta\lambda$	$0 \to 0$	$0 \to 300$	$0 \to 400$	$0 \to 500$
$a\delta e_x$	$0 \to 0$	$2000 \to 0$	$4000 \to 0$	$6000 \to 0$
$a\delta e_y$	$0 \to 0$	$300 \to 0$	$600 \to 0$	$900 \to 0$
$a\delta i_x$	$0 \to 0$	$500 \to 0$	$500 \to 0$	$500 \to 0$
$a\delta i_y$	$0 \to 0$	$0 \to 0$	$0 \to 0$	$0 \to 0$

[a] dimensional using the semimajor axis of the reference orbit

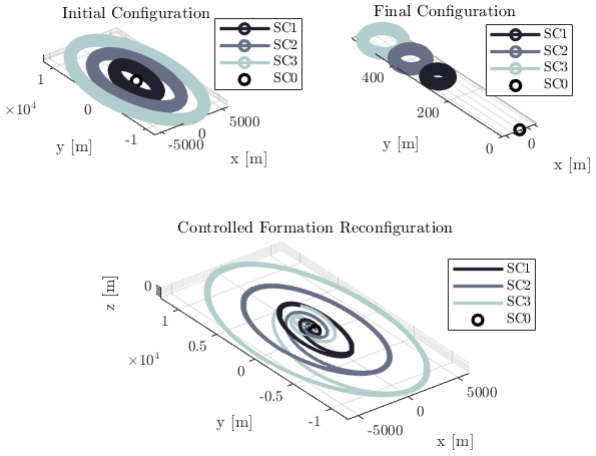

Figure 5: Planar to Along-track (ALO) neural reconfiguration.

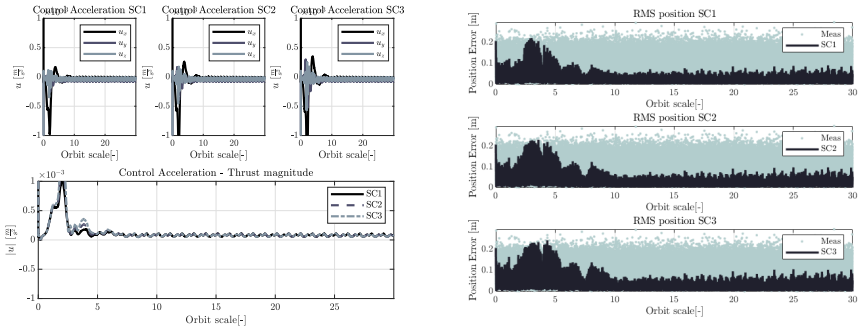

Figure 6: Control effort and estimation accuracy for the ALO neural reconfiguration.

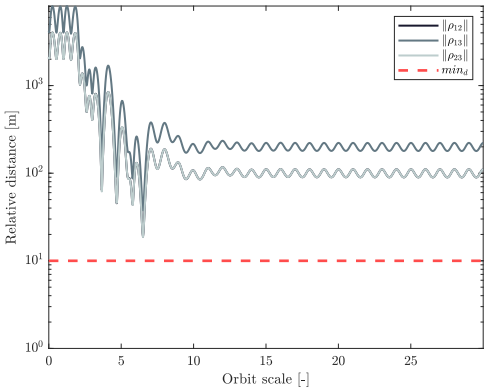

Figure 7: Relative distances between the formation spacecraft for the ALO scenario. The dotted line represents the minimum safe distance set for the simulations.

5.3. Planar Synthetic Aperture Variation

Formation flying missions are foreseen to be employed in configuration that allow synthetic aperture of the distributed system. A Planar Synthetic Aperture Variation (SAV) is a reconfiguration maneuver in which the equivalent instrument diameter is varied. Fig. 8 shows the controlled neural reconfiguration. The initial and final state, expressed in relative orbital elements, are reported in Tab. 3.

The reconfiguration is successful and the neural enhanced GNC architecture converges to stable estimation of the unmodelled terms, given that no unstable

Table 3: Relative orbital elements of each spacecraft in the SAV reconfiguration

$a\delta\chi^a[m]$	S/C 1	S/C 2	S/C 3	S/C 4
$a\delta a$	$0 \to 0$	$0 \to 0$	$0 \to 0$	$0 \to 0$
$a\delta\lambda$	$0 \to 0$	$0 \to 0$	$0 \to 0$	$0 \to 0$
$a\delta e_x$	$0 \to 0$	$100 \to 1000$	$300 \to 3000$	$600 \to 6000$
$a\delta e_y$	$0 \to 0$	$100 \to 1000$	$300 \to 3000$	$600 \to 6000$
$a\delta i_x$	$0 \to 0$	$0 \to 0$	$0 \to 0$	$0 \to 0$
$a\delta i_y$	$0 \to 0$	$0 \to 0$	$0 \to 0$	$0 \to 0$

[a] dimensional using the semimajor axis of the reference orbit

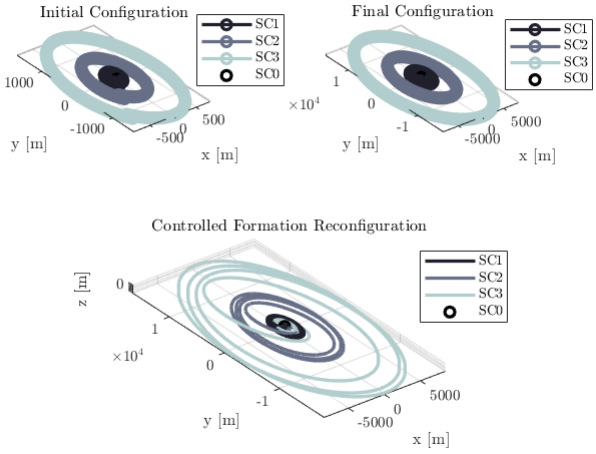

Figure 8: Planar Synthetic Aperture Variation (SAV) neural reconfiguration.

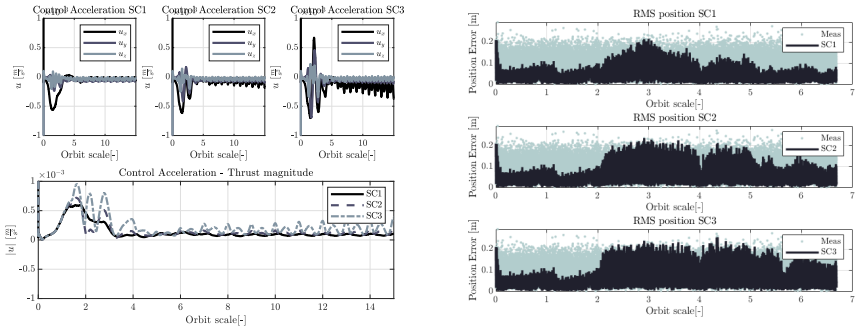

Figure 9: Control effort and estimation accuracy for the SAV neural reconfiguration.

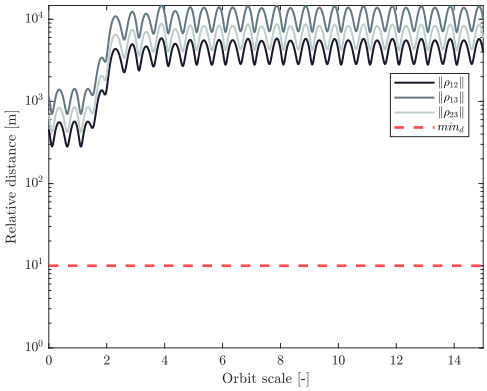

Figure 10: Relative distances between the formation spacecraft for the SAV scenario. The dotted line represents the minimum safe distance set for the simulations.

behavior is experienced, see Section 5.1. Fig. 9 shows the control effort and the navigation results in terms of estimation accuracy. The repulsive potential aided by the RBFNN network guarantees that the minimum distance between the agents is respected, as shown in Fig. 10. The relative maneuver initiates after one orbital period.

5.4. Relative Plane Change

The Relative Plane Change (RPC) reconfiguration is essentially an inversion of the relative inclination vector. The component δi_x, see Eq. 1, is actually the algebraic difference of the spacecraft orbital inclination. Hence, the reconfigura-

Table 4: Relative orbital elements of each spacecraft in the RPC reconfiguration

$a\delta\chi^a[m]$	S/C 1	S/C 2	S/C 3	S/C 4
$a\delta a$	$0 \to 0$	$0 \to 0$	$0 \to 0$	$0 \to 0$
$a\delta\lambda$	$0 \to 0$	$0 \to 0$	$0 \to 0$	$0 \to 0$
$a\delta e_x$	$0 \to 0$	$200 \to 200$	$400 \to 400$	$600 \to 600$
$a\delta e_y$	$0 \to 0$	$300 \to 300$	$600 \to 600$	$900 \to 900$
$a\delta i_x$	$0 \to 0$	$500 \to -500$	$300 \to -500$	$500 \to -500$
$a\delta i_y$	$0 \to 0$	$0 \to 0$	$0 \to 0$	$0 \to 0$

[a] dimensional using the semimajor axis of the reference orbit

tion shown in this section is equivalent to an inclination change maneuver. Such reconfiguration has been chosen because of its complexity on control, involving along-radial-cross track control. Fig. 11 shows the controlled neural reconfiguration. The initial and final state, expressed in relative orbital elements, are reported in Tab. 4.

The reconfiguration is successful and the neural enhanced GNC architecture converges to stable estimation of the unmodelled terms, given that no unstable behavior is experienced, see Section 5.1. Fig. 12 shows the control effort and the navigation results in terms of estimation accuracy. The repulsive potential aided by the RBFNN network guarantees that the minimum distance between the agents is respected, as shown in Fig. 13. The relative maneuver initiates after one orbital period.

5.5. Formation Position Swap

The reconfiguration is specifically designed to include a collision avoidance maneuvers for the sake of demonstration. Basically, the agents $2,\ 3,\ 4$ are asked to swap relative orbits with respect to spacecraft 1, once again fixed along the reference orbit. The threshold d_{lim} in Eq. 18 is set to 50 m and an exit condition for the simulation due to collisions is set to 10 m. In particular, the maneuver

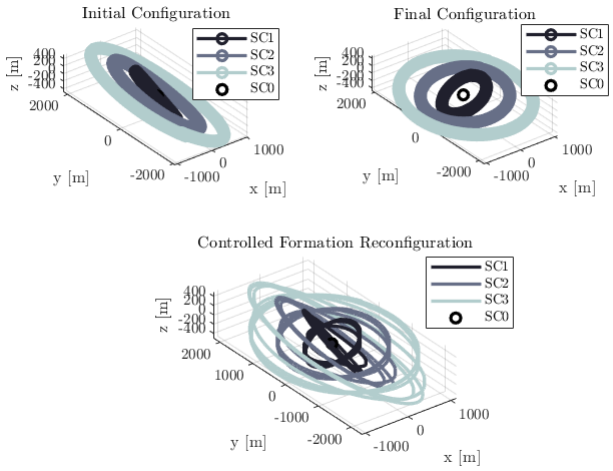

Figure 11: Relative Plane Change (RPC) neural reconfiguration.

is:

$$\delta\chi_{ref,2} = \delta\chi_{0,3},\ \delta\chi_{ref,3} = \delta\chi_{0,4},\ \delta\chi_{ref,4} = \delta\chi_{0,2}, \tag{40}$$

Fig. 14 shows the controlled neural reconfiguration. The initial and final state, expressed in relative orbital elements, are reported in Tab. 5.

The reconfiguration is successful and the neural enhanced GNC architecture converges to stable estimation of the unmodelled terms, given that no unstable behavior is experienced, see Section 5.1. Fig. 15 shows the control effort and the navigation results in terms of estimation accuracy. The repulsive potential aided by the RBFNN network guarantees that the minimum distance between the agents is respected, as shown in Fig. 16. The relative maneuver initiates after one orbital period.

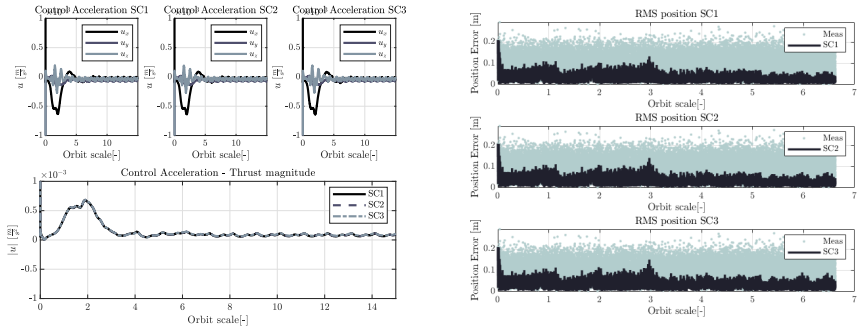

Figure 12: Control effort and estimation accuracy for the RPC neural reconfiguration.

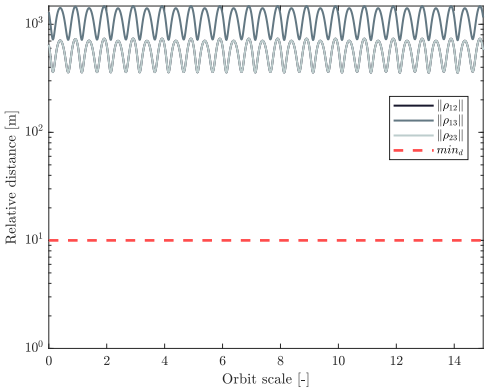

Figure 13: Relative distances between the formation spacecraft for the RPC scenario. The dotted line represents the minimum safe distance set for the simulations.

5.6. Comparison

The Neural-aided algorithm for reconfiguration has been compared with the standard APF algorithm presented in Steindorf et al. (2017) and extended in Silvestrini et al. (2019). Tab. 6, 7 and 8 present the results for the relevant reconfigurations in low-eccentricity reference orbits. In particular, Tab. 6 reports the ΔV effort of the Neural-Lyapunov control and the standard Lyapunov control presented in Section 4.2. The control effort is generally higher in the case of neural control. This may be due to the fact that during the initial phase of the reconfiguration, the main learning process of the network occurs (Pesce

Table 5: Relative orbital elements of each spacecraft in the PS reconfiguration

$a\delta\chi^a[m]$	S/C 1	S/C 2	S/C 3	S/C 4
$a\delta a$	$0 \to 0$	$0 \to 0$	$0 \to 0$	$0 \to 0$
$a\delta\lambda$	$0 \to 0$	$0 \to 0$	$0 \to 0$	$0 \to 0$
$a\delta e_x$	$0 \to 0$	$200 \to 400$	$400 \to 600$	$600 \to 200$
$a\delta e_y$	$0 \to 0$	$300 \to 600$	$600 \to 900$	$900 \to 300$
$a\delta i_x$	$0 \to 0$	$500 \to 300$	$300 \to 100$	$100 \to 500$
$a\delta i_y$	$0 \to 0$	$0 \to 0$	$0 \to 0$	$0 \to 0$

[a] dimensional using the semimajor axis of the reference orbit

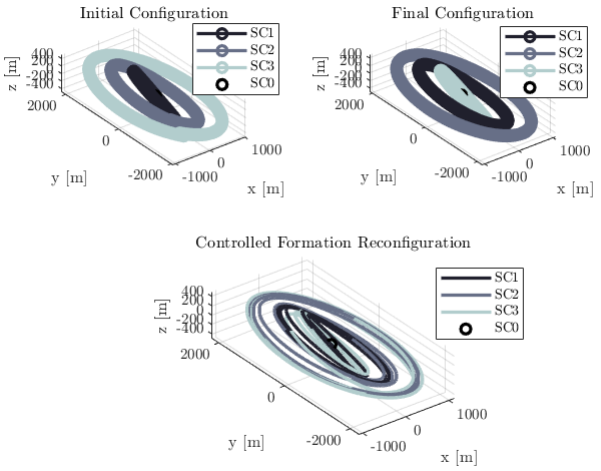

Figure 14: Position Swap (PS) neural reconfiguration.

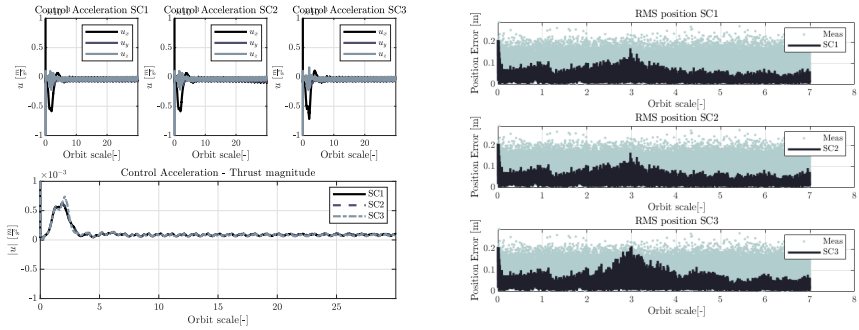

Figure 15: Control effort and estimation accuracy for the PS neural reconfiguration.

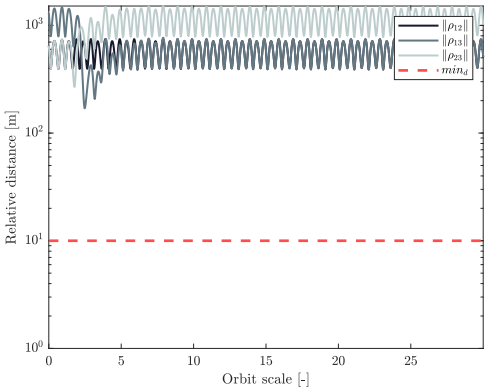

Figure 16: Relative distances between the formation spacecraft for the PS scenario. The dotted line represents the minimum safe distance set for the simulations.

et al., 2020). In this phase the controller is affected by the non-converged term $\gamma(\delta\chi)$. Nevertheless, as reported in Tab. 8 and 9, the achieved accuracy of the final configuration using the neural controller is superior, which is linked to the higher required Δv. The dynamics reconstruction based on RBFNN benefits the navigation also. The position estimation mean error is always lower when RBFNN-AEKF is used. The dynamical model refinement allows the filter to generate a more precise *a-priori* estimation, which is then corrected with the same measurements in both methods. The difference in estimation error increases as the nonlinearities become more relevant in the dynamics. Indeed,

Table 6: Comparison of control effort between standard APF reconfiguration algorithm and the proposed NN-APF. Low-eccentricity scenario.

Low-e	**NNAPF Control**			**APF Control**		
Scenario	$\Delta V_1[\frac{m}{s}]$	$\Delta V_2[\frac{m}{s}]$	$\Delta V_3[\frac{m}{s}]$	$\Delta V_1[\frac{m}{s}]$	$\Delta V_2[\frac{m}{s}]$	$\Delta V_3[\frac{m}{s}]$
ALO	19.81	21.39	23.55	19.48	2.15	23.39
SAV	12.09	14.81	22.64	11.88	14.84	23.08
RPC	12.13	12.43	12.76	11.69.	11.89	12.20
PS	20.42	21.64	19.67	20.18	21.14	19.29

for high-eccentricity reference orbits, the enhancement in state estimation using RBFNN-AEKF is remarked. Tab. 10, 11, 12 and 13 present the results for the analysed reconfigurations in high-eccentricity reference orbits. The NNAPF algorithm outperforms the standard APF, in terms of accuracy and navigation estimate. The ALO and RPC reconfigurations are challenging due to the involved relative distances, the change of plane and the limited maximum thrust available. However, also in this case, the NNAPF strategy delivers better final configuration in terms of accuracy. One important result is that, whenever the neural-improvement in the target accuracy is not significant, the NN-APF control effort to deliver the same accuracy of APF is considerably lower. The high-eccentricity scenario is more demanding since nonlinearities are prominent. The accuracy is poorer because both the controller struggle to handle such disturbance. Nevertheless, the RBFNN achieves a better accuracy with respect to the traditional APF algorithm. In addition, it is important to remark that the RBFNN can be tuned (number of parameters, weights, learning function) to improve the performance, whereas the APF is less flexible.

Table 7: Comparison of navigation accuracy between standard APF reconfiguration algorithm and the proposed NN-APF. Low-eccentricity scenario.

Low-e	**NNAPF Navigation**			**APF Navigation**		
Scenario	σ_1 [m]	σ_2 [m]	σ_3 [m]	σ_1 [m]	σ_2 [m]	σ_3 [m]
ALO	0.02	0.03	0.03	0.03	0.04	0.04
SAV	0.04	0.05	0.06	0.05	0.06	0.07
RPC	0.03	0.03	0.03	0.03	0.04	0.04
PS	0.03	0.03	0.03	0.04	0.04	0.05

Table 8: Comparison of target configuration accuracy between standard APF reconfiguration algorithm and the proposed NN-APF. Low-eccentricity scenario.

Low-e	**Neural Accuracy**[%]			**Accuracy**[%]		
Scenario	$\frac{\lvert\Delta\delta\chi\rvert}{\lvert\delta\chi_{ref}\rvert}_1$	$\frac{\lvert\Delta\delta\chi\rvert}{\lvert\delta\chi_{ref}\rvert}_2$	$\frac{\lvert\Delta\delta\chi\rvert}{\lvert\delta\chi_{ref}\rvert}_3$	$\frac{\lvert\Delta\delta\chi\rvert}{\lvert\delta\chi_{ref}\rvert}_1$	$\frac{\lvert\Delta\delta\chi\rvert}{\lvert\delta\chi_{ref}\rvert}_2$	$\frac{\lvert\Delta\delta\chi\rvert}{\lvert\delta\chi_{ref}\rvert}_3$
ALO	2.25	1.88	1.69	3.26	2.88	2.35
SAV	5.11	5.08	4.76	5.11	5.02	4.76
RPC	10.77	7.87	6.99	10.96	8.86	7.78
PS	7.66	13.98	7.02	4.20	3.80	5.30

Table 9: Norm of the Relative Orbital Elements error (dimensionless) with respect to the target ROE state. Low-eccentricity scenario. A rough order of magnitude in meters can be obtained dimensionalizing the ROEs by multiplying by the semimajor axis.

Low-e	**Neural Accuracy**			**Accuracy**		
Scenario	$\lvert\Delta\delta\chi\rvert_1$	$\lvert\Delta\delta\chi\rvert_2$	$\lvert\Delta\delta\chi\rvert_3$	$\lvert\Delta\delta\chi\rvert_1$	$\lvert\Delta\delta\chi\rvert_2$	$\lvert\Delta\delta\chi\rvert_3$
ALO	$8.5 \cdot 10^{-7}$	$9.4 \cdot 10^{-7}$	$1.1 \cdot 10^{-6}$	$1.2 \cdot 10^{-6}$	$1.4 \cdot 10^{-6}$	$1.5 \cdot 10^{-6}$
SAV	$6.7 \cdot 10^{-6}$	$1.7 \cdot 10^{-5}$	$3.2 \cdot 10^{-5}$	$7.02 \cdot 10^{-6}$	$1.9 \cdot 10^{-5}$	$3.4 \cdot 10^{-5}$
RPC	$8.3 \cdot 10^{-6}$	$8.6 \cdot 10^{-6}$	$1.1 \cdot 10^{-5}$	$8.5 \cdot 10^{-6}$	$9.8 \cdot 10^{-6}$	$1.16 \cdot 10^{-5}$
PS	$7.5 \cdot 10^{-6}$	$2.7 \cdot 10^{-5}$	$5.4 \cdot 10^{-6}$	$8.8 \cdot 10^{-6}$	$3.1 \cdot 10^{-5}$	$6.3 \cdot 10^{-6}$

Table 10: Comparison of control effort between standard APF reconfiguration algorithm and the proposed NN-APF. High-eccentricity scenario.

High-e	**NNAPF Control**			**APF Control**		
Scenario	$\Delta V_1[\frac{m}{s}]$	$\Delta V_2[\frac{m}{s}]$	$\Delta V_3[\frac{m}{s}]$	$\Delta V_1[\frac{m}{s}]$	$\Delta V_2[\frac{m}{s}]$	$\Delta V_3[\frac{m}{s}]$
ALO	16.25	23.67	31.91	15.29	20.64	28.19
SAV	16.72	33.56	61.42	21.48	38.78	66.65
RPC	12.99	13.66	14.43	14.76	15.35	16.02
PS	26.23	30.28	23.56	29.66	33.55	23.13

Table 11: Comparison of navigation accuracy between standard APF reconfiguration algorithm and the proposed NN-APF. High-eccentricity scenario.

High-e	**NNAPF Navigation**			**APF Navigation**		
Scenario	σ_1 [m]	σ_2 [m]	σ_3 [m]	σ_1 [m]	σ_2 [m]	σ_3 [m]
ALO	0.05	0.06	0.07	0.06	0.08	0.08
SAV	0.05	0.05	0.07	0.06	0.08	0.08
RPC	0.04	0.04	0.04	0.05	0.05	0.05
PS	0.05	0.05	0.05	0.07	0.07	0.07

Table 12: Comparison of target configuration accuracy between standard APF reconfiguration algorithm and the proposed NN-APF. High-eccentricity scenario.

High-e	**Neural Accuracy**			**Accuracy** [%]		
Scenario	$\frac{\lvert\Delta\delta\chi\rvert}{\lvert\delta\chi_{ref}\rvert}_1$	$\frac{\lvert\Delta\delta\chi\rvert}{\lvert\delta\chi_{ref}\rvert}_2$	$\frac{\lvert\Delta\delta\chi\rvert}{\lvert\delta\chi_{ref}\rvert}_3$	$\frac{\lvert\Delta\delta\chi\rvert}{\lvert\delta\chi_{ref}\rvert}_1$	$\frac{\lvert\Delta\delta\chi\rvert}{\lvert\delta\chi_{ref}\rvert}_2$	$\frac{\lvert\Delta\delta\chi\rvert}{\lvert\delta\chi_{ref}\rvert}_3$
ALO	15.11	17.08	20.44	19.35	23.40	27.22
SAV	16.25	12.61	11.84	17.65	13.87	12.94
RPC	54.55	34.44	28.42	65.60	38.52	32.90
PS	23.31	32.75	19.23	24.80	35.75	20.68

Table 13: Norm of the Relative Orbital Elements error (dimensionless) with respect to the target ROE state. High-eccentricity scenario. A rough order of magnitude in meters can be obtained dimensionalizing the ROEs by multiplying by the semimajor axis.

High-e	**Neural Accuracy**			**Accuracy**		
Scenario	$\lvert\Delta\delta\chi\rvert_1$	$\lvert\Delta\delta\chi\rvert_2$	$\lvert\Delta\delta\chi\rvert_3$	$\lvert\Delta\delta\chi\rvert_1$	$\lvert\Delta\delta\chi\rvert_2$	$\lvert\Delta\delta\chi\rvert_3$
ALO	$4.2 \cdot 10^{-6}$	$6.6 \cdot 10^{-6}$	$9.9 \cdot 10^{-6}$	$5.6.8 \cdot 10^{-6}$	$9.1 \cdot 10^{-6}$	$1.3 \cdot 10^{-5}$
SAV	$2.2 \cdot 10^{-5}$	$5.2 \cdot 10^{-5}$	$9.8 \cdot 10^{-5}$	$2.4 \cdot 10^{-5}$	$5.7 \cdot 10^{-5}$	$1.1 \cdot 10^{-4}$
RPC	$3.3 \cdot 10^{-5}$	$2.9 \cdot 10^{-5}$	$3.3 \cdot 10^{-5}$	$3.9 \cdot 10^{-5}$	$3.3 \cdot 10^{-5}$	$3.8 \cdot 10^{-5}$
PS	$1.7 \cdot 10^{-5}$	$3.5 \cdot 10^{-5}$	$1.1 \cdot 10^{-5}$	$1.9 \cdot 10^{-5}$	$3.8 \cdot 10^{-5}$	$1.2 \cdot 10^{-5}$

5.6.1. Thrusters Misalignment

An additional simulation set is presented here to tackle another practical problem in spacecraft GNC. Thruster misalignment can be regarded as uncertain terms that affect each satellite differently. Hereby, for the purpose of assessing NN-APF orbit control performance, a thruster on each satellite axis that has a random angular misalignment from the desired thrust axis is considered. This results in a Gaussian random noise with zero-mean and $\sigma_u = 10^{-4} \frac{m}{s^2}$. When an acceleration is commanded, the misalignment causes the satellite to accelerate in a direction that is not perfectly aligned with the desired direction. This is a practical concern on real satellites and is an area where feedback control must be utilized to keep the satellites on the desired orbit. In this simulation set, the neural-aided algorithm always performs better in terms of control accuracy, increasing only marginally the control cost, as shown in Tab. 14 and 15.

5.6.2. Highly Perturbed Environment

The comparison presented in Section 5.6 showed how the NNAPF algorithm guarantees a more accurate navigation and control, at the cost of slightly higher Δv. The results show that also the legacy algorithm was able to bring the relative maneuvers to completion. This may be due to the fact the the disturbance terms included in the *ground-truth* relative dynamics are still handled by the

Table 14: Comparison of control effort between standard APF reconfiguration algorithm and the proposed NN-APF. Thruster misalignment scenario.

Thrust Mis.	**NNAPF Control**			**APF Control**		
Scenario	$\Delta V_1[\frac{m}{s}]$	$\Delta V_2[\frac{m}{s}]$	$\Delta V_3[\frac{m}{s}]$	$\Delta V_1[\frac{m}{s}]$	$\Delta V_2[\frac{m}{s}]$	$\Delta V_3[\frac{m}{s}]$
ALO	12.95	14.82	17.23	12.42	14.48	16.96
SAV	12.04	14.89	22.76	11.61	14.95	23.21
RPC	12.15	12.42	12.71	11.66	11.91	12.20
PS	20.45	20.56	19.78	20.05	21.13	19.43

Table 15: Comparison of target configuration accuracy between standard APF reconfiguration algorithm and the proposed NN-APF. Thruster misalignment scenario.

Thrust Mis.	**Neural Accuracy**[%]			**Accuracy**[%]		
Scenario	$\frac{\lvert\Delta\delta\chi\rvert}{\lvert\delta\chi_{ref}\rvert}_1$	$\frac{\lvert\Delta\delta\chi\rvert}{\lvert\delta\chi_{ref}\rvert}_2$	$\frac{\lvert\Delta\delta\chi\rvert}{\lvert\delta\chi_{ref}\rvert}_3$	$\frac{\lvert\Delta\delta\chi\rvert}{\lvert\delta\chi_{ref}\rvert}_1$	$\frac{\lvert\Delta\delta\chi\rvert}{\lvert\delta\chi_{ref}\rvert}_2$	$\frac{\lvert\Delta\delta\chi\rvert}{\lvert\delta\chi_{ref}\rvert}_3$
ALO	11.06	8.12	6.86	11.96	9.08	7.54
SAV	3.07	2.05	1.92	3.72	3.13	2.93
RPC	11.01	8.03	7.07	11.78	8.48	7.57
PS	6.04	11.4	10.01	7.85	12.34	11.05

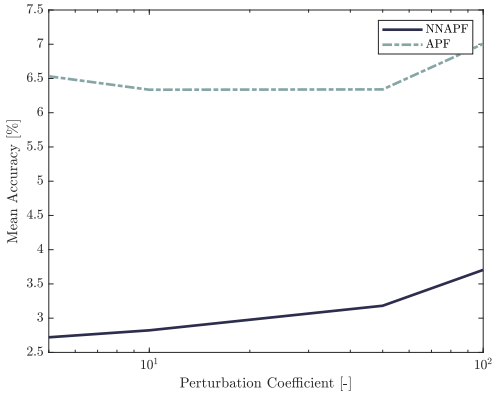

Figure 17: Mean accuracy of NNAPF and APF in highly perturbed environment. The perturbation coefficient is the multiplicative term of J_2 perturbation to generate the fictitious disturbance.

APF control. In order to showcase the quality of the NNAPF algorithm, a set of numerical simulations have been performed adding fictitious perturbations to the *ground-truth* dynamics to challenge both algorithms. In particular, the *perturbation coefficient* k_{pert} is introduced. The *perturbation coefficient* k_{pert} is a multiplicative term used to increase the J_2 perturbation to generate the fictitious disturbance:

$$\mathbf{F} = k_{pert} \cdot \mathbf{F}_{J_2} \tag{41}$$

Figure 17 reports the mean accuracy of the final formation configuration for different values of k_{pert}. The results are averaged over the scenarios and over the satellites. The results show that a constant increase in performance is achieved by the NNAPF when perturbations become more dominant. As already noted in Section 5.6, the better NNAPF accuracy is achieved at the cost of slightly higher Δv. The increase in Δv is within the range $[0.5 - 1.0]\frac{m}{s}$ in all the simulations.

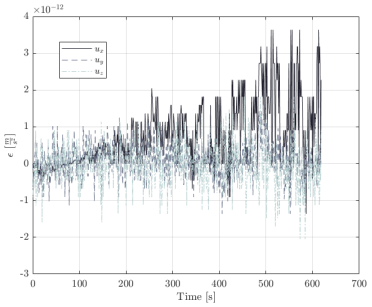

Figure 18: Discrepancy in calculation between PIL and SIL simulations.

Table 16: Average and maximum execution time of GNC routines using a single core TMS320C28x 32-Bit CPUs @200 MHz of TI C2000-Delfino MCUs F28379D

Routine	Average $[ms]$	Max $[ms]$
rbfnn(.)	0.89	0.90
aekf(.)	0.81	0.82
apf(.)	0.32	0.32
ctrl(.)	0.28	0.33

6. Processor-In-the-Loop Validation

The presented simulations based on the NNAPF algorithm, comprising the whole RBFNN-GNC architecture, were simulated using a Desktop computer using Intel® Core^{TM} i5-3470 CPU @3.20 GHz. The computational time for a single step execution of the NNAPF is $< 50ms$.

The NNAPF has been developed for an on-board applications, hence it is critical to evaluate its computational burden on relevant processor and hardware. For this reason, Processor-In-the-Loop simulations are performed using a TI C2000-Delfino MCUs F28379D, which is a dual-core TMS320C28x 32-Bit CPUs @200 MHz. The algorithm is developed in Matlab/Simulink®. The software is then ported to the MCU using the dedicated coder. The algorithm is run in external mode PIL, interfacing with the orbital simulator running on a Desktop computer using Intel® Core^{TM} i5-3470 CPU @3.20 GHz. The average execution times of the autonomous GNC routines are limited, as reported in Tab. 18. Fig. 18 shows the discrepancy in the calculated control by the MCU and the Desktop simulator. The values are in the order of $\sim 10^{-12}\frac{m}{s^2}$, which can be assumed to be numerical error, thus validating the PIL test. The MCU presented in this analysis is representative of the hardware integrated in the frictionless 5DoF facility installed at Politecnico di Milano (Visconti et al., 2018; Ottolina

et al., 2019). Naturally, the algorithm development follows the same workflow adopted here: the numerical algorithms are coded in Matlab/Simulink®, tested in Desktop computers and finally automatically ported into the MCU using the dedicated coder. This enables fast prototyping and hardware validation of the selected algorithms.

7. Conclusion

A neural Guidance, Navigation & Control algorithm based on Artificial Potential Field and Radial-Basis Function Neural Network is presented. The Artificial Potential Field algorithm has been extended to smooth the guidance dynamics and manage reconfigurations with more than two agents. The algorithm takes as input the Cartesian measurements of relative states between satellites. The developed navigation filter and controller is coupled with a neural reconstructed term that encompass all the disturbances or nonlinearities not modelled in the on-board dynamical model. Such refinement of on-board dynamics enhance the whole GNC system performance. Radial-Basis-Function Neural Networks have been employed for the inherent nonlinear neurons structure, which yields a faster learning process. The dynamics reconstruction works online, meaning that offline training is not needed. The Neural Network Artificial Potential Field (NNAPF) algorithm has been tested numerically and in Processor-In-the-Loop (PIL) simulations using a single core TMS320C28x 32-Bit CPUs @200 MHz of TI C2000-Delfino MCUs F28379D unit. The numerical and PIL tests demonstrated:

- the neural reconstructed dynamics yields more accurate navigation and final target configuration with respect to Artificial Potential Field algorithm.
- the algorithm can be executed using limited computational power, thus making it suitable for on-board applications.

The future work will focus on solving the drawback of Artificial Potential Field algorithms, which is the potential instability that may arise with the collision

avoidance constraints with several agents approaching the same area simultaneously. In addition, the algorithms will be tested in the 5-Dof frictionless facility at Politecnico di Milano (Visconti et al., 2018; Ottolina et al., 2019).

References

Bae J, Kim Y. Adaptive controller design for spacecraft formation flying using sliding mode controller and neural networks. Journal of the Franklin Institute 2012;349(2):578–603.

Chernick M, D'Amico S. New Closed-Form Solutions for Optimal Impulsive Control of Spacecraft Relative Motion. Journal of Guidance, Control, and Dynamics 2016;41(2):301–19. URL: https://arc.aiaa.org/doi/10.2514/1.G002848.

Chu J. Dynamics, Distributed Control And Autonomous Cluster Operations Of Fractionated Spacecraft, 2015.

D'Amico S. Autonomous formation flying in low earth orbit. Ph.D. thesis; 2010.

Di Mauro G, Spiller D, Bevilacqua R, Curti F. Optimal Continuous Maneuvers for Satellite Formation Reconfiguration in J2-perturbed Orbits. 2018 Space Flight Mechanics Meeting 2018;0216(January):1–20.

Guffanti T, D'Amico S, Lavagna M. Long-term analytical propagation of satellite relative motion in perturbed orbits. Advances in the Astronautical Sciences 2017;160:2387–417.

Gurfil P, Idan M, Kasdin NJ. Adaptive neural control of deep-space formation flying. Journal of Guidance Control and Dynamics 2003;26(3):491–501.

Izzo D, Pettazzi L. Autonomous and Distributed Motion Planning for Satellite Swarm. European Space Agency, (Special Publication) ESA SP 2005;30(603):727–36.

Koenig AW, Guffanti T, D'Amico S. New State Transition Matrices for Spacecraft Relative Motion in Perturbed Orbits. Journal of Guidance, Control, and Dynamics 2017;40(7):1749–68.

Lane CM, Axelrad P. Formation Design in Eccentric Orbits Using Linearized Equations of Relative Motion. Journal of Guidance, Control, and Dynamics 2006;29(1):146–60.

Li Q, Zhang B, Yuan J, Wang H. Potential function based robust safety control for spacecraft rendezvous and proximity operations under path constraint. Advances in Space Research 2018;62(9):2586–98. doi:10.1016/j.asr.2018.08.003.

Ottolina D, Silvestrini S, Lavagna M. DANCE: Design and Characterization of a 5 DOF Facility for Relative GNC. In: ASTRA. 2019. .

Pesce V, Silvestrini S, Lavagna M. Radial basis function neural network aided adaptive extended Kalman filter for spacecraft relative navigation. Aerospace Science and Technology 2020;1:105527. doi:10.1016/j.ast.2019.105527.

Sarno S, Guo J, D'Errico M, Gill E. A Guidance Approach to Satellite Formation Reconfiguration Based On Convex Optimization and Genetic Algorithms. Advances in Space Research 2020;URL: https://doi.org/10.1016/j.asr.2020.01.033. doi:10.1016/j.asr.2020.01.033.

Schaub H, Alfriend KT. Hybrid Cartesian and Orbit Element Feedback Law for Formation Flying Spacecraft. Journal of Guidance, Control, and Dynamics 2002;25(2):387–93.

Schaub H, Vadali SR, Junkins JL, Alfriend KT. Spacecraft formation flying control using mean orbit elements. Journal of the Astronautical Sciences 2000;48(1):69–87.

Silvestrini S, Pesce V, Lavagna M. Distributed Autonomous Guidance , Navigation and Control loop for Formation Flying Spacecraft Reconfiguration. In: 5th CEAS Conference on Guidance, Navigation and Control. 2019. p. 1–19.

Steindorf LM, D'Amico S, Scharnagl J, Kempf F, Schilling K. Constrained low-thrust satellite formation-flying using relative orbit elements. Advances in the Astronautical Sciences 2017;160:3563–83.

Vadali SR, Alfriend KT. Formation Establishment, Maintenance and Control. In: Distributed Space Missions for Earth System Monitoring. 2013. p. 1–675.

Visconti P, Silvestrini S, Lavagna M. Dance: a Frictionless 5 DOF Facility For GNC Proximity Maneuvering Experimental Testing And Validation. In: IAC. Number October; 2018. p. 1–5.

Wu Y, Wang H, Zhang B, Du KL. Using Radial Basis Function Networks for Function Approximation and Classification. ISRN Applied Mathematics 2012;2012(March):1–34.

Zhou N, Chen R, Xia Y, Huang J, Wen G. Neural network–based reconfiguration control for spacecraft formation in obstacle environments. International Journal of Robust and Nonlinear Control 2018;28(6):2442–56.