V. Pesce, S. Silvestrini, M. Lavagna
Radial Basis Function Neural Network Aided Adaptive Extended Kalman Filter for Spacecraft Relative Navigation
Aerospace Science and Technology, In press - Published online 31/10/2019
doi:10.1016/j.ast.2019.105527

Radial Basis Function Neural Network aided Adaptive Extended Kalman Filter for Spacecraft Relative Navigation

Vincenzo Pesce*, Stefano Silvestrini †, and Michèle Lavagna‡
Politecnico di Milano, via La Masa 34, Milano, Italy

This paper presents a novel technique, combining neural network and Kalman filter, for state estimation. The proposed solution provides the estimates of the system states while also estimating the uncertain or unmodeled terms of the process dynamics. The developed algorithm exploits a Radial Basis Function Neural Network that outputs an estimate of the disturbances that are included in the prediction step of an Adaptive Extended Kalman Filter. A recursive form of adaptation is used to limit the computational burden. The proposed solution is compared to classical navigation filter implementations. A realistic spacecraft relative navigation scenario is selected to test the filter performance. Simulations are performed with accurate tuning and also in off-nominal conditions to test the filter robustness.

I. Introduction

Spacecraft absolute and relative navigation are key tasks in the Guidance Navigation & Control (GNC) chain for current and future missions. Current navigation algorithms rely on the accurate knowledge of the system dynamics. This is possible whenever spacecrafts orbiting the Earth are considered, where the environment can be accurately modeled to a great extent of accuracy. Nevertheless, when dealing with relative approach with unknown bodies or interplanetary missions, the modelling of the system dynamics yields inevitable unmodeled uncertainties. This is mainly due to partial knowledge of the operative scenario, e.g. orbital disturbances acting on the target spacecraft. Furthermore, the growing interest towards micro-platforms, both for Earth and interplanetary missions, has significantly reduced the spacecraft available computational power; hence, very sophisticated models cannot be anymore handled on-board. Such limitation leads to a degradation of performance of the GNC subsystem [1]. In this framework, on one hand, the dynamical model employed in the on-board algorithms needs to be simplified, on the other hand, the accuracy of such model significantly deteriorates the GNC performance [1], due to the absence of nonlinear terms as well as disturbances. The Artificial Neural Networks (ANNs) are a powerful tool to bridge this gap. ANNs are becoming increasingly important when dealing with uncertain processes. In particular, their capability of approximating unknown functions can be employed to reconstruct system nonlinearities, as well as unmodeled environmental disturbances. The advantage of estimating

*PhD Candidate, Department of Aerospace Science and Technology, vincenzo.pesce@polimi.it
†PhD Candidate, Department of Aerospace Science and Technology, stefano.silvestrini@polimi.it
‡Associate Professor, Department of Aerospace Science and Technology, michelle.lavagna@polimi.it

such uncertainties benefits the whole GNC process chain. In this framework, Gurfil et al. [2] presented a nonlinear adaptive neural control method applicable to deep space formation flying. Bae and Kim [3] developed a neural network aided sliding mode control scheme for spacecraft formation flying. Recently, Zhou [4] proposed a neural-network based reconfiguration control for spacecraft formation in obstacle environments. Traditionally, the ANNs are solely employed for disturbance estimation, yet the aim of the Navigation filter is to estimate the system state. In past years, there have been attempts to couple ANNs with Extended Kalman Filters (EKF). In particular, the most common approach is to employ EKFs to train the ANNs [5]. In this configuration, the state of the Kalman filter is augmented with the ANN weights. For a large network this process increases the computational burden. Furthermore, the resulting coupled structure cannot provide an estimate of the uncertainties, unless the disturbance vector is added to the state vector and estimated as a constant parameter. An alternative solution is to use the estimated disturbance term, output of the ANN, directly in the dynamical propagation of the filter [6]. In this way, instead of the state vector, the dynamics of the EKF is augmented by an ANN that captures the unmodeled dynamics. The ANN learns online the function describing the disturbance, i.e. the mismatch between the measurement and the a-priori guess given by the model selected for the EKF. However, in this case, the *augmented* dynamical model accuracy changes in time and therefore, its covariance matrix has to be adapted at each step to capture this variation. In the past years, few solutions have been proposed to derive an efficient formulation for neural network aided filters. Gao et al. [7] derived a Radial Basis Function Neural Network (RBFNN) - Kalman Filter to improve the estimation accuracy for seam tracking during high-power fiber laser welding. They proposed a coupled formulation where the RBFNN is used to compensate for the model and noise uncertainties. However, they do not consider any online adaptation of the filter covariances. Similarly, Stubberud et al. [6] developed a neuro-observer based on an EKF and a multilayer feed-forward neural network. Their formulation involves two coupled Kalman Filters, one to estimate the state and the other to tune the neural network. Other authors also proposed neural network for system identification based on offline training [8, 9]. Jwo and Huang [10] presented a neural network aided EKF for DGPS positioning. The neural network is used for noise identification to adaptively tune the EKF. However, their neural network relies on an offline training using the steepest descent technique. Recently Harl et al. [11] developed a reduced-order modified state observer for uncertainties estimation in nonlinear systems. They also applied the proposed technique to estimate the uncertain disturbances caused by J_2 perturbation around the Earth. Also in this case, the gain of the observer is user-selected and there is not any kind of adaptation depending on the experienced scenario. In this paper, we propose a Radial Basis Function Neural Network aided Adaptive Extended Kalman Filter (RBFNN-AEKF) for state and disturbance estimation. RBFNN are selected for their simple structure and suitability for fast online training [12]. The neural network estimates the unmodeled terms which are fed to the EKF as an additional term to the state and covariance prediction step. Finally, a recursive form of the adaptive EKF is employed to limit the overall computational cost. The intended contributions of the paper are:

- to propose a filter to be employed with unknown or very uncertain environment;

- to develop a robust filter through the adaptation step;
- to demonstrate the feasibility of exploiting the proposed filter for relative navigation in perturbed orbits with a very simple dynamical model;
- to compare it with other common techniques for navigation filtering with various degrees of dynamics uncertainty.

The paper is organized as follows: the proposed algorithm architecture is presented in Section II; the neural network and the filter structure are detailed in Subsections II.A and II.B, respectively. Section III describes the applicative scenario of spacecraft relative navigation. Section IV presents the simulation environment and the results for the different filters in nominal and off-nominal case. Finally, in Section V, the conclusions are drawn.

II. Algorithm Architecture

The filter architecture is sketched in Figure 1. The neural network estimates the disturbances acting on the system, which are then adjuncted in the prediction step of the filter. The innovation term is used to carry out the adaptivity task. Whereas, the residual term, taking into account the estimation state at step k, is fed into the online learning algorithm of the network's weights. Each block of the RBFNN-AEKF is detailed in the following subsections. The system dynamics,

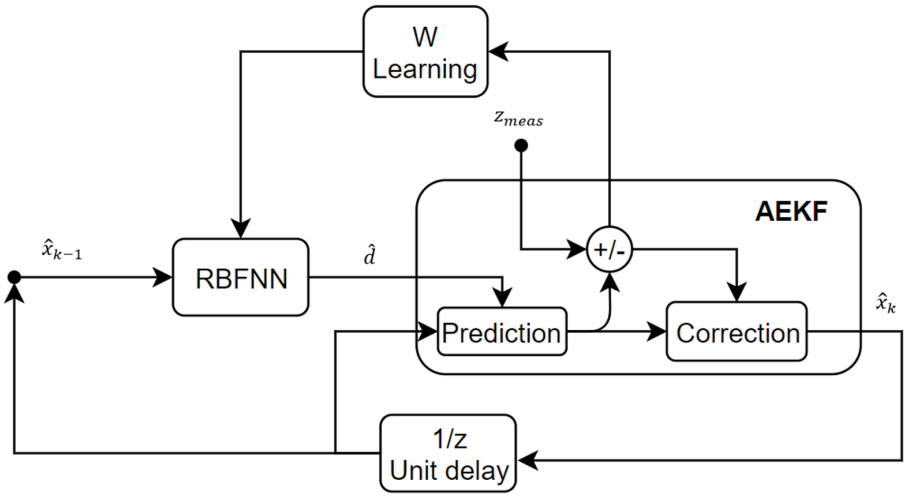

Fig. 1 Proposed architecture for the RBFNN-AEKF.

taking into account the process noise, is assumed to be described as:

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}) + \mathbf{w} \tag{1}$$

Alongside, the measurements are assumed to be perturbed by white noise as:

$$\mathbf{z_{meas}} = \mathbf{I}\mathbf{x} + \mathbf{v} \tag{2}$$

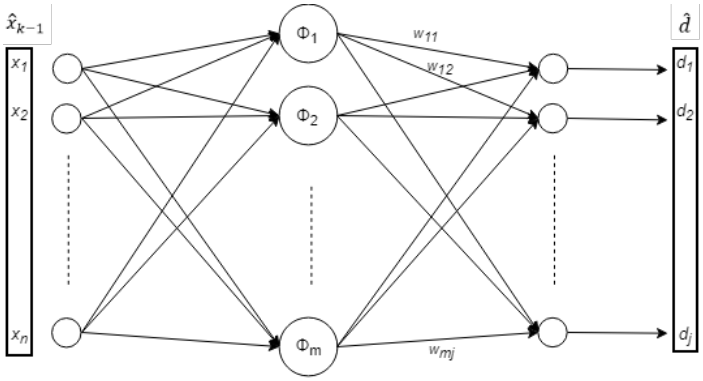

Fig. 2 Architecture of the RBF neural network. The network processes the estimated states yield an estimate of the disturbance term. The input, hidden, and output layers have n, m, and j neurons, respectively. $\Phi_i(\mathbf{x})$ denotes the radial Gaussian function at the hidden node i.

Note that in the derivation presented in the following sections, we assume that the observation model is $\mathbf{z_{meas}} = \mathbf{Ix} + \mathbf{v}$ as in [11]. Normally, an observation function, often nonlinear, is introduced as explained in Section II.B. Such assumption implies that the observation matrix $\mathbf{H} = \mathbf{I}$ or, more in general, $h(\mathbf{x}) = \mathbf{x}$. In other words, the state is assumed to be completely observable for sake of derivation but the approach is applicable to partially observable state and to nonlinear measurement models.

A. RBFNN

1. Neural Network Structure

The Radial Basis Function Neural Network (RBFNN) is a popular network topology, which has the capability of universal approximation [12] [13]. Due to its simple structure and much quicker learning process, it stands out compared to the classic Multi-Layer Perceptron (MLP), especially for function approximation applications [12]. The RBF neural network is a three-layer feedforward network, as seen in Figure 2. The RBF owns a single hidde-layer because it does not need multiple layers to obtain nonlinear behaviour clasification, as in Multi-Layer Perceptron. The neurons of RBF are nonlinear Gaussian function hence a shallow network can be used with the same results of multi-layer perceptron. Hence, referring to Fig. 1, the lightest network has been chosen, consisting of one input layer, one hidden and one output layer. The input layer processes the state vector $\hat{\mathbf{x}}_{\mathbf{k-1}} = [x_1\ x_2\ \dots\ x_n]^T$. The hidden layer performs a nonlinear mapping of the input, whereas, the output layer is a linear combination of the nonlinear hidden neurons transformed into the resultant output space. The output space is the disturbance vector $\hat{\mathbf{d}}\ [d_1\ d_2\ \dots\ d_n]^T$ A RBFNN is used to estimate the unmodeled disturbances, as well as the nonlinearities present in the system dynamics. The generic layout of the network is sketched in Figure 2. The network has a 3-layers structure, comprising an input, output and hidden layer. For the sake of derivation we call $\mathbf{x} \in \mathbb{R}^n$ the input vector. It is hereby remarked that the vector $\mathbf{x}$ is employed to derive the network structure: in the following sections the distinction between state vector and estimated state will be described and treated

accordingly. Similarly to the input vector, $\mathbf{\Phi} \in \mathbb{R}^m$ is the hidden layer vector and i the associated index, $\mathbf{d} \in \mathbb{R}^j$ is the output vector and l the associated index. In this derivation we assume that $n \equiv j$. Essentially, the hidden layer evaluates a set of m radial basis functions $\Phi : \mathbb{R}^n \rightarrow \mathbb{R}$, where n is the number of states, which are chosen as centered-Gaussian expression:

$$\Phi_i(\mathbf{x}) = e^{-\eta(||\mathbf{x}-\mathbf{c_i}||)^2} \tag{3}$$

for $i = 1 : m$, where m is the number of neurons and $\mathbf{c}_i$ is the randomly selected center for neuron i. The number of neurons m is a user-defined parameter: its value is application-dependent and it shall be selected by trading-off the reconstruction accuracy and the computational time. The same consideration holds for the parameter η, which impacts the shape of the Gaussian functions. A high value for η sharpens the Gaussian bell-shape, whereas a low value spreads it on the real space. On one hand, a narrow Gaussian function increases the responsiveness of the RBF network, on the other hand in case of limited overlapping of the neuronal functions due to too narrow Gaussian bells vanishes the output of the network. Hence, ideally, the parameter η is selected based on the order of magnitude of the exponential argument in Eq. 3. The output of the neural network hidden layer, namely the radial functions evaluation, is normalized:

$$\mathbf{\Phi}_{norm}(\mathbf{x}) = \frac{\mathbf{\Phi}(\mathbf{x})}{\sum_{i=1}^{m} \Phi_i(\mathbf{x})} \tag{4}$$

The classic RBF network presents an inherent localized characteristic; whereas, the normalized RBF network exhibits good generalization properties, which decreases the curse of dimensionality that occurs with classic RBFNN [12]. In the following derivation, the output vector of the hidden layer is simply called $\mathbf{\Phi}(\mathbf{x})$ without the subscript $norm$ for the sake of simplicity. For a generic input $\mathbf{x} \in \mathbb{R}^n$, the components of the output vector $\mathbf{d} \in \mathbb{R}^j$ of the network is:

$$d_l(\mathbf{x}) = \sum_{i=1}^{m} w_{il} \Phi_i(\mathbf{x}) \tag{5}$$

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

$$\mathbf{d}(\mathbf{x}) = \mathbf{W}^T \mathbf{\Phi}(\mathbf{x}) \tag{6}$$

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.

2. Online Learning Algorithm

The dynamical model can be described by a set of nonlinear differential equations:

$$\dot{\mathbf{x}} = f(\mathbf{x}) + \mathbf{d}_{ext} \tag{7}$$

where the term $\mathbf{d_{ext}}$ is representative of the unknown disturbance acceleration that is added to the known dynamics function $f(\mathbf{x})$. In particular the disturbance term gathers the contribution of all the environmental perturbations, and unmodeled terms. These uncertainties need to be estimated online. Hence, an online learning algorithm, which drives the update of the weights, is required. The weights update law is derived to guarantee the stability of the estimation algorithm and neural dynamics, hereby defined as the evolution of the weight matrix in time. In the following mathematical derivation we make use of the universal approximation theorem for neural networks that guarantees the existence of a set of ideal weights $\mathbf{W}$ that approximates a function with a bounded arbitrary approximation error [12]. Such weights are unknown, hence the algorithm is designed to obtain an estimate $\hat{\mathbf{W}}$ of the ideal weights by performing online learning. The neural network learning algorithm relies on the estimation error dynamics, targeting convergence and stability of the estimated weights matrix $\hat{\mathbf{W}}$ evolution towards the ideal weights and the error $\mathbf{e}$, calculated in the EKF. The symbol $\hat{(\cdot)}$ is used to refer to *estimated* quantities.

To derive the error dynamics, let us assume the actual system dynamics is described by Eq. 7, where $\mathbf{d_{ext}}$ is the unknown external disturbance term. The actual system dynamics can be rewritten as the following equation, assuming to include all the nonlinear terms into $d(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}^j, j \equiv n$, which is the vector-valued function equivalent to the RBFNN output vector:

$$\dot{\mathbf{x}} = \mathbf{A} \cdot \mathbf{x} + d(\mathbf{x}) \tag{8}$$

where the term $d(\mathbf{x})$ captures all the nonlinearities together with the unknown disturbances external to the system, namely $d(\mathbf{x}) = f(\mathbf{x}) - \mathbf{A}\mathbf{x} + \mathbf{d_{ext}}$. The matrix A is a stable, potentially time-varying, matrix representing the linear term, if any, of the original dynamics expression in Eq. 7. The expression of the continuous single-step Kalman filter can be written as:

$$\dot{\hat{\mathbf{x}}} = \mathbf{A} \cdot \hat{\mathbf{x}} + \hat{d}(\hat{\mathbf{x}}) + \mathbf{K_k}\mathbf{H}(\mathbf{x} - \hat{\mathbf{x}}) \tag{9}$$

where $\hat{d}$ is estimated using the radial-basis function neural network, $\mathbf{K_k}$ is the time-varying gain matrix of the Kalman filter (subscript k stands for Kalman here) and $\mathbf{H}$ is the observation matrix. In this paper, the observation matrix of the measurement model is assumed to be the identity matrix, nevertheless the derivation is not altered if one would consider a different expression for the measurement model. The measurements are assumed to be affected by white noise as in Eq. 2. Consider that the continuous form is employed for the sake of derivation, indeed the learning rule is then

discretized for the actual implementation. The error dynamics can be derived as:

$$\mathbf{e} = \mathbf{x} - \hat{\mathbf{x}} \tag{10}$$

$$\dot{\mathbf{e}} = \dot{\mathbf{x}} - \dot{\hat{\mathbf{x}}} = d(\mathbf{x}) - \hat{d}(\hat{\mathbf{x}}) + (\mathbf{A} - \mathbf{K_k}\mathbf{H})\mathbf{e} \tag{11}$$

Invoking the universal approximation theorem for neural networks [12], we can assume there exists an ideal approximation of the disturbance term $d(\mathbf{x})$:

$$d(\mathbf{x}) = \mathbf{W}^T \mathbf{\Phi}(\mathbf{x}) + \epsilon \tag{12}$$

where $\mathbf{W}$ is the neural weights matrix, $\mathbf{\Phi}(\mathbf{x})$ is the vector-valued function resulting from the evaluation of the Gaussian functions contained in each neuron of the RBFNN, ϵ is a bounded arbitrary approximation error. Consequently, the error in estimation can be written as:

$$d(\mathbf{x}) - \hat{d}(\hat{\mathbf{x}}) = \mathbf{W}^T \mathbf{\Phi}(\mathbf{x}) + \epsilon - \hat{\mathbf{W}}^T \mathbf{\Phi}(\hat{\mathbf{x}}) \tag{13}$$

by adding and subtracting the term $\mathbf{W} \cdot \mathbf{\Phi}(\hat{\mathbf{x}})$ and performing few mathematical manipulations, Equation 13 can be expressed as:

$$\tilde{d} = \tilde{\mathbf{W}}^T \Phi(\hat{\mathbf{x}}) + \epsilon' \tag{14}$$

where $\tilde{d} = d - \hat{d}$, $\tilde{\mathbf{W}} = \mathbf{W} - \hat{\mathbf{W}}$ and the bounded term $\epsilon' = \epsilon + \mathbf{W} \cdot [\mathbf{\Phi}(\mathbf{x}) - \mathbf{\Phi}(\hat{\mathbf{x}})]$. The aim of the learning rule is to drive the dynamics error to zero, as well as forcing the weights to converge to the ideal ones. Namely:

$$\mathbf{e} \to \mathbf{0}, \ \tilde{\mathbf{W}} \to [\mathbf{0}]$$

Similarly to [11], introducing the following scalar Lyapunov function for the feedback system, including the network weights and the estimation error, the weights update rule $\dot{\hat{\mathbf{W}}}$ is derived to guarantee the stability and convergence of the estimation algorithm:

$$V = \frac{1}{2} tr(\xi \tilde{\mathbf{W}}^T \tilde{\mathbf{W}}) + \frac{\eta}{2} \mathbf{e}^T \mathbf{e} \tag{15}$$

where $tr(\cdot)$ is the trace operator, $\xi, \ \eta > 0$ are user-defined coefficients. Recalling Eq. 11 and 14, the derivative of the

Lyapunov function can be written as:

$$\begin{aligned}
\dot{V} &= tr(\xi\tilde{\mathbf{W}}^T\dot{\tilde{\mathbf{W}}}) + \eta\mathbf{e}^T\mathbf{e} \\
&= tr(\xi\tilde{\mathbf{W}}^T\dot{\tilde{\mathbf{W}}}) + \eta\mathbf{e}^T(\tilde{\mathbf{W}}^T\boldsymbol{\Phi}(\hat{\mathbf{x}}) + \epsilon' + (\mathbf{A} - \mathbf{K_k}\mathbf{H})\mathbf{e}) \\
&= tr(\xi\tilde{\mathbf{W}}^T\dot{\tilde{\mathbf{W}}}) + \eta\mathbf{e}^T\tilde{\mathbf{W}}^T\boldsymbol{\Phi}(\hat{\mathbf{x}}) + \eta\mathbf{e}^T\epsilon' + \eta\mathbf{e}^T(\mathbf{A} - \mathbf{K_k}\mathbf{H})\mathbf{e} \\
&= tr(\xi\tilde{\mathbf{W}}^T\dot{\tilde{\mathbf{W}}} + \eta\tilde{\mathbf{W}}^T\boldsymbol{\Phi}(\hat{\mathbf{x}})\mathbf{e}^T) + \eta\mathbf{e}^T\epsilon' + \eta\mathbf{e}^T(\mathbf{A} - \mathbf{K_k}\mathbf{H})\mathbf{e} \\
&= tr(\tilde{\mathbf{W}}^T(\xi\dot{\tilde{\mathbf{W}}} + \eta\boldsymbol{\Phi}(\hat{\mathbf{x}})\mathbf{e}^T)) + \eta\mathbf{e}^T\epsilon' + \eta\mathbf{e}^T(\mathbf{A} - \mathbf{K_k}\mathbf{H})\mathbf{e} < 0
\end{aligned} \tag{16}$$

Recalling that $\dot{\tilde{\mathbf{W}}} = -\dot{\hat{\mathbf{W}}}$, the expression for the weights update rule that guarantees stability and convergence of the estimation algorithm and feedback system in Fig. 1 for weights update:

$$\dot{\hat{\mathbf{W}}} = \frac{\eta}{\xi}\boldsymbol{\Phi}(\hat{\mathbf{x}})\mathbf{e}^T \tag{17}$$

Indeed, by inserting Eq. 17 into Eq. 16, the expression for the derivative of the Lyapunov function reduces to the stability of the error estimation of the Extended Kalman Filter. The error term in Eq. 17 is defined as Eq. 10. Such term represent the residual between estimated and actual output of the observed system: in practical terms, the expression is the *innovation* of the estimation filter, which takes this form based on the assumption of Eq. 2. The ϵ' term is a bounded term that derives from the universal approximation theorem of artificial neural networks [12] that states that the term ϵ can be arbitrarily small. In practice, it represents an upper boundary for the derivative of the Lyapunov function, as in [11]. In the case of linear systems, the term $(\mathbf{A} - \mathbf{K_k}\mathbf{H})$ grants asymptotic stability of the Kalman Filter if $\mathbf{A}$ is reachable and $\mathbf{H}$ is observable. In the case of nonlinear systems, this is not always true. However, it has been proved [14] that the estimation error of an EKF is exponentially bounded if:

- $\mathbf{A}$ is nonsingular for every $t \geq 0$;
- there exist real constants $p_1, p_2 > 0$ such that $p_1 \cdot \mathbf{I} \leq \mathbf{P_k} \leq p_2 \cdot \mathbf{I}$, where $\mathbf{P_k}$ is the estimated state covariance matrix;
- the initial estimation error satisfies $||\hat{\mathbf{x}}_\mathbf{0} - \mathbf{x_0}|| \leq \epsilon$ and the process and measurements covariance matrices are bounded

where $\hat{\mathbf{x}}_\mathbf{0}$ and $\mathbf{x_0}$ are the estimated and true state vector at the initial step. Given the EKF asymptotic stability with exponential decaying error under the aforementioned conditions, i.e. the derivative of the Lyapunov function of the estimation error is negative, Eq. 16 is verified and hence the stability of the estimator is guaranteed.

The weights update rule can be discretized using a first-order Euler method, assuming the measurements interval is

small enough:

$$\hat{\mathbf{W}}_{k+1} = \psi\hat{\mathbf{W}}_k + h\dot{\hat{\mathbf{W}}}_k = \psi\hat{\mathbf{W}}_k + h\frac{\eta}{\xi}\mathbf{\Phi}(\hat{\mathbf{x}}_\mathbf{k})\mathbf{e}_k^T \tag{18}$$

where k is the time step index, ψ is a user-defined relaxation factor, $h = t_{k+1} - t_k$ is the time interval between two consecutive measurements.

B. Adaptive Extended Kalman Filter

The EKF is one of the most common techniques for nonlinear state estimation. In fact, it represents the extension of the linear Kalman Filter when dealing with nonlinear dynamical systems [15]. The EKF is the standard approach for relative navigation filters [16–18]. However, sometimes, accurate dynamical models are not available on-board. This is mostly due to computational limitations (e.g. for Cubesats) or unavailability of precise formulations (e.g. non-Keplerian dynamics). In these cases, a simpler model has to be used and the unmodeled effects have to be estimated. In this section the proposed approach combining the introduced RBFNN and a Kalman Filter is presented. The novelties lie in the formulation of a stand-alone estimator block containing a disturbance estimator and a filter. It is important to underline as the RBFNN and the filter are tightly coupled and they do not constitute two separate pieces. This is also a consequence of the implementation of an adaptive form of the process covariance matrix update.

Let's consider the system and measurement models:

$$\mathbf{x_k} = f(\mathbf{x_{k-1}}, \mathbf{u_{k-1}}, \mathbf{w_{k-1}}) = \mathbf{A}\mathbf{x_{k-1}} + d(\mathbf{x_{k-1}}, \mathbf{u_{k-1}}) + \mathbf{w_{k-1}}, \tag{19}$$

$$\mathbf{z_k} = h(\mathbf{x_k} + \mathbf{v_k}) \tag{20}$$

with $\mathbf{x}$ being the state vector, $\mathbf{u}$ the control input, $\mathbf{w}$ and $\mathbf{v}$ process and measurement noises, described by zero-mean white noise uncorrelated distributions with covariance matrices $\mathbf{Q}$ and $\mathbf{R}$ respectively. Please notice that the system dynamics is described by a linear part plus a non-linear disturbance function d, by assumption (see 8). In this case, the formulation of the RBFNN-AEKF is given by:

$$\hat{\mathbf{x}}_\mathbf{k}^- = \mathbf{A}\hat{\mathbf{x}}_\mathbf{k-1}^- + d(\hat{\mathbf{x}}_\mathbf{k-1}^-, \mathbf{u_{k-1}}) \tag{21}$$

$$\mathbf{P_k^-} = \tilde{\mathbf{F}}_\mathbf{k-1}\mathbf{P_{k-1}^+}\tilde{\mathbf{F}}_\mathbf{k-1}^\mathbf{T} + \mathbf{Q_{k-1}} \tag{22}$$

$$\mathbf{K_k} = \mathbf{P_k^-}\mathbf{H^T}(\mathbf{H}\mathbf{P_k^-}\mathbf{H^T} + \mathbf{R_k})^{-1} \tag{23}$$

$$\mathbf{P_k^+} = (\mathbf{I} - \mathbf{K_k}\mathbf{H})\mathbf{P_k^-}(\mathbf{I} - \mathbf{K_k}\mathbf{H})^\mathbf{T} + \mathbf{K_k}\mathbf{R_k}\mathbf{K_k^T} \tag{24}$$

$$\hat{\mathbf{x}}_\mathbf{k}^+ = \hat{\mathbf{x}}_\mathbf{k}^- + \mathbf{K_k}(\mathbf{z_k} - \mathbf{H}\hat{\mathbf{x}}_\mathbf{k}^-) \tag{25}$$

with

$$\tilde{\mathbf{F}} = \mathbf{A} + \left.\frac{\partial d}{\partial \mathbf{x}}\right|_{\hat{x}_k^-}; \qquad \mathbf{H} = \left.\frac{\partial h}{\partial \mathbf{x}}\right|_{\hat{x}_k^-} = \mathbf{I} \tag{26}$$

and

$$\mathbf{Q_k} = \alpha \mathbf{Q_{k-1}} + (1-\alpha)(\mathbf{K_k}\delta_\mathbf{k}\delta_\mathbf{k}^\mathbf{T}\mathbf{K_k^T}) \tag{27}$$

being $\mathbf{P}_k$ the estimation error covariance and $\mathbf{K}_k$ the Kalman gain, α is a forgetting factor and $\delta_\mathbf{k} = \mathbf{z_k} - \mathbf{H}\hat{\mathbf{x}}_\mathbf{k}^-$ is the filter innovation. The Jacobian in Eq. 26 of the vector-valued function reconstructed by the RBFNN is derived from Eq. 6:

$$\frac{\partial d}{\partial \mathbf{x}} = \frac{\partial \mathbf{W}^T \mathbf{\Phi}(\mathbf{x})}{\partial \mathbf{x}} = \mathbf{W}^T \frac{\partial \mathbf{\Phi}(\mathbf{x})}{\partial \mathbf{x}} \in \mathbb{R}^{n \times n} \tag{28}$$

In this formulation, the state is assumed to completely observable. This assumption can be relaxed but a different formulation has to be developed for the neural network (see [11]). The adaptation of $\mathbf{Q}$ is performed according to Eq. 27 as in [19] to limit the computational effort. The adaptation step is a fundamental aspect in the implementation of this estimation technique. In fact, the nonlinear term estimated by the RBFNN $d(\mathbf{x})$ directly affects the state estimation and, moreover, it influences the accuracy of the adopted dynamical model at each time step. The evolution in time of the model accuracy is very difficult to be established a-priori being dependent on the effectiveness of the disturbance term estimation, performed by the RBFNN time-varying. Although this aspect is often neglected [7], an online tuning of the process covariance matrix $\mathbf{Q}$ is fundamental to ensure filter accuracy and robustness. In fact, the adaptive formulation guarantees that, even when the neural network produces a completely wrong estimates of the disturbances, yielding a significantly biased dynamical model, the filter, at least, follows the available measurements. In fact, if the network yields a disturbance term that is significantly off, the dynamical model is no longer reliable. This implies a large value of the filter innovation δ, which delivers a high-valued process covariance matrix according to Eq. 27. As result, the filter does not diverge but simply follows the measurements (Eq. 22). The overall algorithm implementation is reported in Algorithm 1:

Algorithm 1: RBFNN-AEKF

begin

Random initialization of the *RBFNN*

RBFNN estimation step of $\mathbf{d}$ using the estimated state at the previous step $\rightarrow \mathbf{d}(\mathbf{x}) = \mathbf{W}^T \mathbf{\Phi}(\mathbf{x})$

RBFNN weight update $\rightarrow \dot{\hat{\mathbf{W}}} = \frac{\eta}{\xi}\mathbf{\Phi}(\hat{\mathbf{x}})\mathbf{e}^T$

EKF Prediction step, considering the computed $\mathbf{d}$ term $\rightarrow \hat{\mathbf{x}}_\mathbf{k}^- = \mathbf{A}\hat{\mathbf{x}}_\mathbf{k-1}^- + d(\hat{\mathbf{x}}_\mathbf{k-1}^-, \mathbf{u_{k-1}})$

EKF Update step $\rightarrow \hat{\mathbf{x}}_\mathbf{k}^+ = \hat{\mathbf{x}}_\mathbf{k}^- + \mathbf{K_k}(\mathbf{z_k} - \mathbf{H}\hat{\mathbf{x}}_\mathbf{k}^-)$

EKF $\mathbf{Q}$ adaptation step $\rightarrow \mathbf{Q_k} = \alpha \mathbf{Q_{k-1}} + (1-\alpha)(\mathbf{K_k}\delta_\mathbf{k}\delta_\mathbf{k}^\mathbf{T}\mathbf{K_k^T})$

Result: Estimated state $\hat{x}_k^+$

III. Application to Spacecraft Relative Navigation

In this section, we introduce one of the possible applications of the proposed RBFNN-AEKF. The relative navigation between two spacecrafts orbiting the Earth is considered for a dual motivation: it is a well-known scenario, hence sophisticated model can be employed to simulate the reality to evaluate the filter performances; also, there are many available dynamical models with increasing levels of accuracy that can be used for comparison. It is worth remarking that this is not the only application nor the most appealing one, since more uncommon scenarios are expected to emphasize the benefit of the proposed filter, such as interplanetary mission or non-keplerian orbits. Hereby, the different dynamical models for filter propagation are presented as well as the filter alternatives used for comparison.

A. Relative Dynamical Models

The paper uses a high-fidelity propagator as *truth* for algorithm validation. The accurate orbital simulator is used to test the filters in a realistic environment. In fact, the relative motion between target and chaser is obtained by integrating separately the chaser and the target orbital dynamics considering the perturbations acting on each spacecraft. In particular, the model considers irregularities in the gravitational potential due to non-spherical distribution of Earth's mass, the presence of the Moon and the Sun as third-body, the effect of the solar radiation pressure (SRP) and the atmospheric drag. The adopted Earth gravitational model is the EGM96 with harmonics up to the third degree and order. On the other hand, the atmospheric drag force is computed by using the Jacchia Reference Atmosphere model. The relative dynamics equations, used in the filter implementation as *design* models, are hereby presented.

1. Clohessy-Wilthsire Equations

The most common set of equations to describe the relative dynamics between two spacecrafts are the well-known Clohessy-Wiltshire equations, which are presented hereby. For a full derivation, the authors suggest to refer to [20]. With reference to Figure 3, the target spacecraft is in a nominal orbit at a distance $\mathbf{r_0}$ from the attractor. If the chaser is in close proximity of the target, the orbital radius can be expressed as $\mathbf{r} = \mathbf{r_0} + \delta\mathbf{r}$. The equation of motion of the chaser spacecraft is:

$$\ddot{\mathbf{r}} = -\mu\frac{\mathbf{r}}{r^3} \tag{29}$$

where μ is the gravitational constant of the central body. Let us attach a co-moving frame of reference centered on the target spacecraft center of mass, as shown in Figure 3. The x axis is aligned with $\mathbf{r_0}$, the z is orthogonal to the orbital plane along the positive angular momentum vector and the y axis completing the right-hand triad. Such reference frame is commonly known as Local-Vertical-Local-Horizontal (LVLH). Substituting the definition of $\mathbf{r}$ to Eq. 29 and

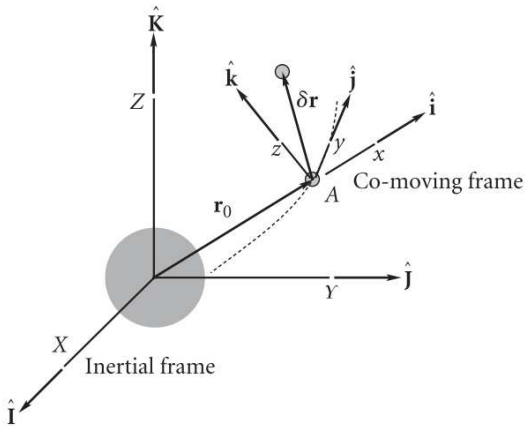

Fig. 3 Co-moving LVLH frame [20].

expressing everything in the LVLH, the Clohessy-Wiltshire equations are obtained for a circular reference target orbit:

$$\begin{aligned} \delta\ddot{x} - 3n^2\delta x - 2n\delta\dot{y} &= 0 \\ \delta\ddot{y} + 2n\delta\dot{x} &= 0 \\ \delta\ddot{z} + n^2\delta z &= 0 \end{aligned} \tag{30}$$

where $n = \frac{2\pi}{T}$ is the orbital mean motion and T is the reference orbital period.

2. Nonlinear Dynamical Model of J_2-Perturbed Relative Motion

The nonlinear dynamical model for J_2-perturbed relative orbit is described in [21]. Hereby, the fundamental equations are solely reported; for a thorough derivation, refer to [21]. With reference to Figure 3, the dynamics of the spacecraft can be written as:

$$\begin{aligned} \delta\ddot{x} &= 2\omega_z\delta\dot{y} - (n_j^2 - \omega_z^2)\delta x + \alpha_z\delta y - \omega_x\omega_z\delta z - (\zeta_j - \zeta)s_i s_\theta - r(n_j^2 - n^2) + a_x \\ \delta\ddot{y} &= -2\omega_z\delta\dot{x} + 2\omega_x\delta z - \alpha_z\delta x - (n_j^2 - \omega_z^2 - \omega_x^2)\delta y + \alpha_x\delta z - (\zeta_j - \zeta)s_i c_\theta + a_y \\ \delta\ddot{z} &= -2\omega_x\delta\dot{y} - \omega_x\omega_z\delta x - \alpha_x\delta y - (n_j^2 - \omega_x^2)\delta z - (\zeta_j - \zeta)c_i + \alpha_z \end{aligned} \tag{31}$$

where the contributing terms are:

$$
\begin{aligned}
n^2 = \frac{\mu}{r_0^3} + \frac{k_{J2}}{r_0^5} - \frac{5k_{J2}s_i^2 s_\theta^2}{r^5}, \quad n_j^2 = \frac{\mu}{r^3} + \frac{k_{J2}}{r^5} - \frac{5k_{J2}r_{JZ}^2}{r^7} \\
r_{JZ} = (r_0 + \delta x)s_i s_\theta + \delta y s_i c_\theta + \delta z c_i, \quad k_{J2} = \frac{3J_2\mu R_e^2}{2} \\
\omega_x = -\frac{k_{J2}s_{2i}s_\theta}{hr_0^3}, \quad \omega_z = \frac{h}{r_0^2} \\
\alpha_x = \dot{\omega}_x = \frac{k_{J2}s_{2i}c_\theta}{r_0^5} + \frac{3\dot{r_0}k_{J2}s_{2i}s\theta}{r_0^4 h} - \frac{8k_{J2}^2 s_i^3 c_i s_\theta^2 c_\theta}{r^6 h^2} \\
\alpha_z = \dot{\omega}_z = -\frac{2h\dot{r_0}}{r_0^3} - \frac{k_{J2}s_i^2 s_{2\theta}}{r_0^5}, \quad \zeta = \frac{2k_{J2}s_i s_\theta}{r^4}, \quad \zeta_j = \frac{2k_{J2}r_{JZ}}{r^5}
\end{aligned}
\tag{32}
$$

in which h is the orbital angular momentum, i orbital inclination, J_2 is the zonal harmonic coefficient $1.0826 \cdot 10^{-3}$ for Earth, R_e is the Earth radius, θ is the orbital true anomaly and $\mathbf{a} = [a_x, a_y, a_z]^T$ is the forced acceleration vector. Last, s_x and c_x stand for $\sin(x)$ and $\cos(x)$, where x is a generic angle. The spacecraft relative motion is actually described by 11 first-order differential equations, namely $(\delta x, \delta y, \delta z, \delta\dot{x}, \delta\dot{y}, \delta\dot{z})$ and $(r_0, \dot{r_0}, h, i, \theta)$. Nevertheless in this paper, the latter quantities are computed using the high-fidelity propagator previously described. This can be representative of an on-board absolute state estimator. Alternatively, these quantities can be included in the integration step of the dynamical model.

B. Tested Filters

In this subsection, we present the filters used for the comparison. Besides the RBFNN-AEKF, the new filter proposed in this paper, other filters are tested under the same simulation scenario:

- a state observer based on the formulation in [11];
- a standard, non-adaptive EKF aided with a RBFNN;
- an EKF exploiting a more accurate, nonlinear dynamical model.

1. Observer

The dynamics of the relative motion between the spacecrafts is reconstructed using a modified full-state observer [11]. In the same fashion as Section II.A, the state observer can be constructed as follows [22]:

$$
\dot{\hat{\mathbf{x}}} = \mathbf{A}_{cw} \cdot \hat{\mathbf{x}} + \hat{d}(\hat{\mathbf{x}}) + \mathbf{K_h}(\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}) \tag{33}
$$

where $\mathbf{A}_{cw}$ is the linear time invariant matrix of the Clohessy-Wiltshire dynamics presented in Section III.A.1, $\hat{d}$ is estimated using the radial-basis function neural network in Section II.A, $\mathbf{K_h}$ is the user-defined observer gain matrix and $\mathbf{H} = \mathbf{I}$ is the identity observation matrix, as already stated in Section II.

Table 1 Chaser-Target Orbital Parameters

	Chaser	**Target**
a [km]	8143.1	8143.1
e [-]	$1.4{\cdot}10^{-1}$	$1.4{\cdot}10^{-1}$
i [°]	98.2	98.2
ω [°]	85.9	85.9
Ω [°]	79.2	79.2
θ [°]	0	$1{\cdot}10^{-4}$
A_{sp} [m^2]	1.2	0.2

2. RBFNN-EKF

This filter is a standard EKF aided with RBFNN as presented in Section II. The only difference with respect to the proposed RBFNN-AEKF is that the value of the process covariance **Q** is fixed in time. It is worth underlying that this can be a very weak point because it is hard to a-priori establish the accuracy of the RBFNN-based disturbance estimation, especially for very uncertain dynamics. In fact, the matrix **Q** provides an indication of the accuracy of the dynamical model. Using a neural network to update the dynamical model and to increase its accuracy, it is therefore necessary to adapt the value of **Q** at each iteration step. The RBFNN-EKF formulation is based on Equations 21-25.

3. EKF - nonlinear

The last tested filter is an EKF with a different dynamical model. There is not any coupling with the neural network but the used dynamical model is nonlinear and accounting for J_2 perturbations. In particular, a standard EKF is employed where the evolution of the state vector is described by the nonlinear model introduced in Section III.A.2.

IV. Scenarios & Results

In this section, the numerical simulation environment to evaluate the filter performance is described. First, the selected scenario is presented. Subsequently, the capability of disturbance reconstruction of the RBFNN-AEKF is tested. Then, the filters presented in Section III.B are compared using a realistic orbital environment. At this point, the definition of measurement noise levels and filters tuning are introduced. Finally, the same simulation is performed using non-nominal filter tuning conditions to test the robustness of the navigation filters.

A. Orbital Scenario

The reference relative orbital motion is generated considering two spacecraft with the same initial orbital parameters except for the true anomaly. Table 1 reports the chaser and target initial orbital parameters along with the cross sectional area, important for disturbances evaluation. with A_{sp} being the cross sectional area. These orbital parameters result in

the following relative initial conditions, expressed in the target LVLH reference frame:

$$\rho_0 = [-0.0017 \;\; -12.2042 \;\; 4.7 \cdot 10^{-4}] \; m \tag{34}$$

$$\dot{\rho}_0 = [-0.0017 \;\; 3.9 \cdot 10^{-6} \;\; -5.6 \cdot 10^{-10}] \; m/s \tag{35}$$

Please note that the reference orbits are eccentric and that the cross sectional area are different, resulting in a different perturbation effect and potential dynamical mismodeling. In fact, the different cross sectional areas lead to a different differential drag and solar radiation pressure perturbations. It is worth underlying that at the selected altitude, the effect of the drag is not relevant but still present. A similar reasoning can be done for the solar radiation pressure. The most relevant effect is given by the eccentricity of the orbits: the design models neglecting such contribution (e.g. Clohessy-Wiltshire) deliver a modelling errors of tens of meters after one relative orbits. The presented scenario has been selected as representative of a leader-follower formation, separated along the orbit by a difference in the true anomaly. The truth model, as discussed before, is propagated true the high fidelity propagator considering all the perturbation effects. The dynamical models used by the filters depend on the selected architecture and are detailed in III.B.

B. Disturbance Reconstruction

The RBFNN disturbance approximation capability is assessed through the simulation of the scenario presented in Section IV.A. In order to have a quantitative disturbance term, which can be compared to the ANN estimation, the actual relative motion is propagated using the J_2-perturbed relative motion in Section III.A.2. Instead, the filter exploits a simple Clohessy-Wiltshire linearized model described in III.A.1. The normalized ANN consists of 60 hidden neurons with Gaussian-basis radial functions; the function centers are generated randomly. The number of neurons has been selected by trading-off the reconstruction accuracy and the computational time. In such framework, the disturbances that need to be estimated are caused by the following elements:

- J_2 zonal gravity perturbation
- $e \neq 0$, elliptical orbits;

together with the nonlinearities neglected in the derivation in Section III.A.1. Using the J_2-perturbed nonlinear model in Section III.A.2, the disturbance term is explicit in the form of $\mathbf{d} = [d_x \; d_y \; d_z]^T$ acceleration term. For the coherence of vectors dimensionality the estimation is actually performed for the vector $\mathbf{d}_{6x1} = \mathbf{G}\mathbf{d}$, where $\mathbf{G} = [\mathbf{0_{3x3}}; \mathbf{I_{3x3}}]$. The reference orbit is a LEO, which is incidentally assumed to be the orbit of the target spacecraft. Table 1 reports the orbital parameters of the two spacecrafts. It is assumed to have relative position and velocity measurements at 1Hz, with a noise level described by a Gaussian distribution with standard deviation $\sigma_{pos} = 10^{-2} m$ and $\sigma_{vel} = 10^{-4} m/s$. These values are representative of a relative RF metrology system (see [1]). The estimation of the disturbance acceleration term

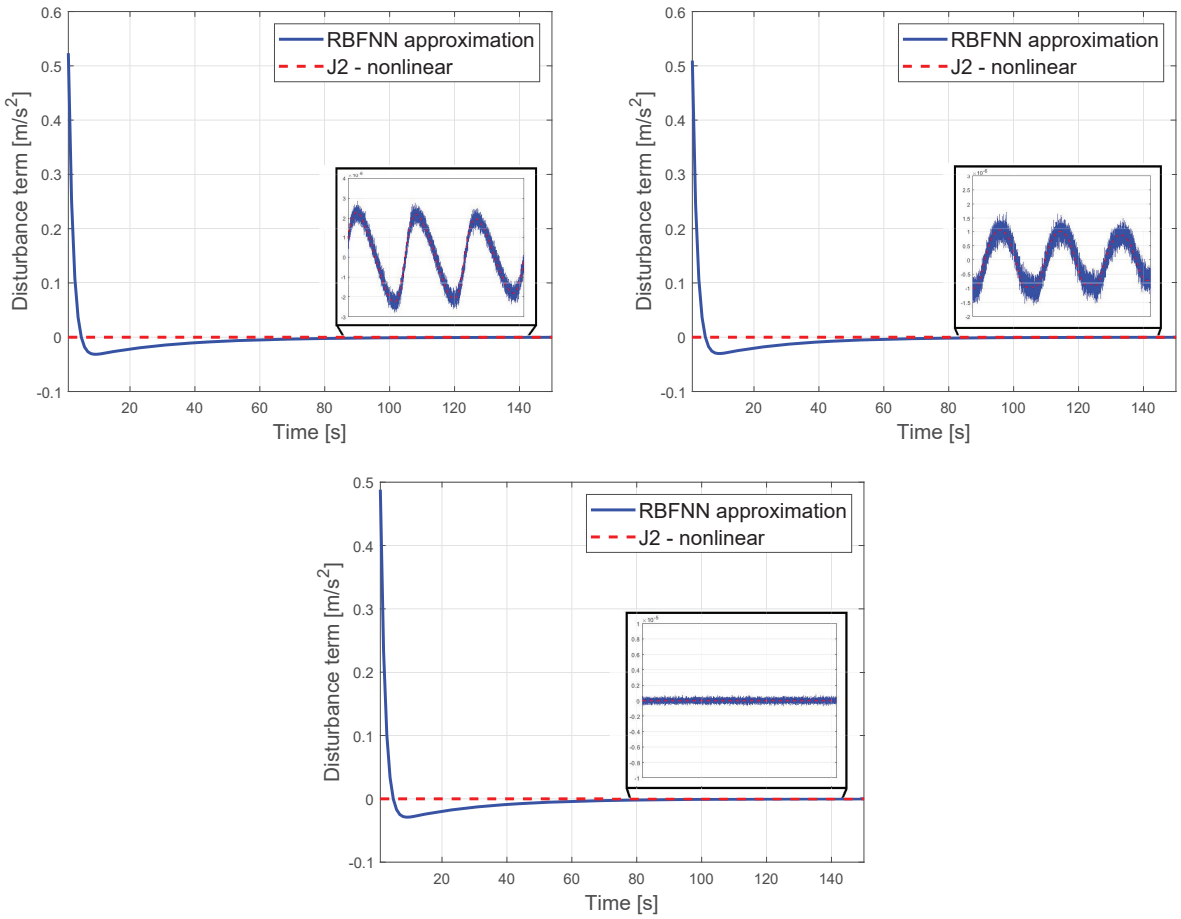

Fig. 4 Estimation of the disturbance acceleration term for LEO reference orbit. The perturbations are in the order of $10^{-5}\ \frac{m}{s^2}$. The plots show the initial phase of the neural network learning, regarded as the *main learning process*. From left to right: d_x, d_y and d_z.

converges after a transient time of nearly 350 s: this represents the *main learning process* of the randomly initialized network. The network is said to be converged when the estimation error is less than 10% of the nominal value. Figure 4 shows the learning curve of the network during the early phase of the orbital motion. The RBFNN is randomly initialized: indeed, the estimation is significantly off by almost ~ 6 orders of magnitudes during the initial phase. The disturbance acceleration components, after the *main learning process*, are shown in Figure 5. Despite the measurement noise, the estimation yields a RMSE reported in Table 2.

C. Relative Navigation - Nominal Case

An accurate orbital simulator is used to test the filters in a realistic environment, as described in Section III.A. The normalized neural network consists of 60 hidden neurons with Gaussian-basis radial functions; the function centers are generated randomly. This reference orbits are also used to generate relative measurements by adding a fictitious noise, representative of realistic sensors uncertainty. In particular, the noise level associated to relative position and

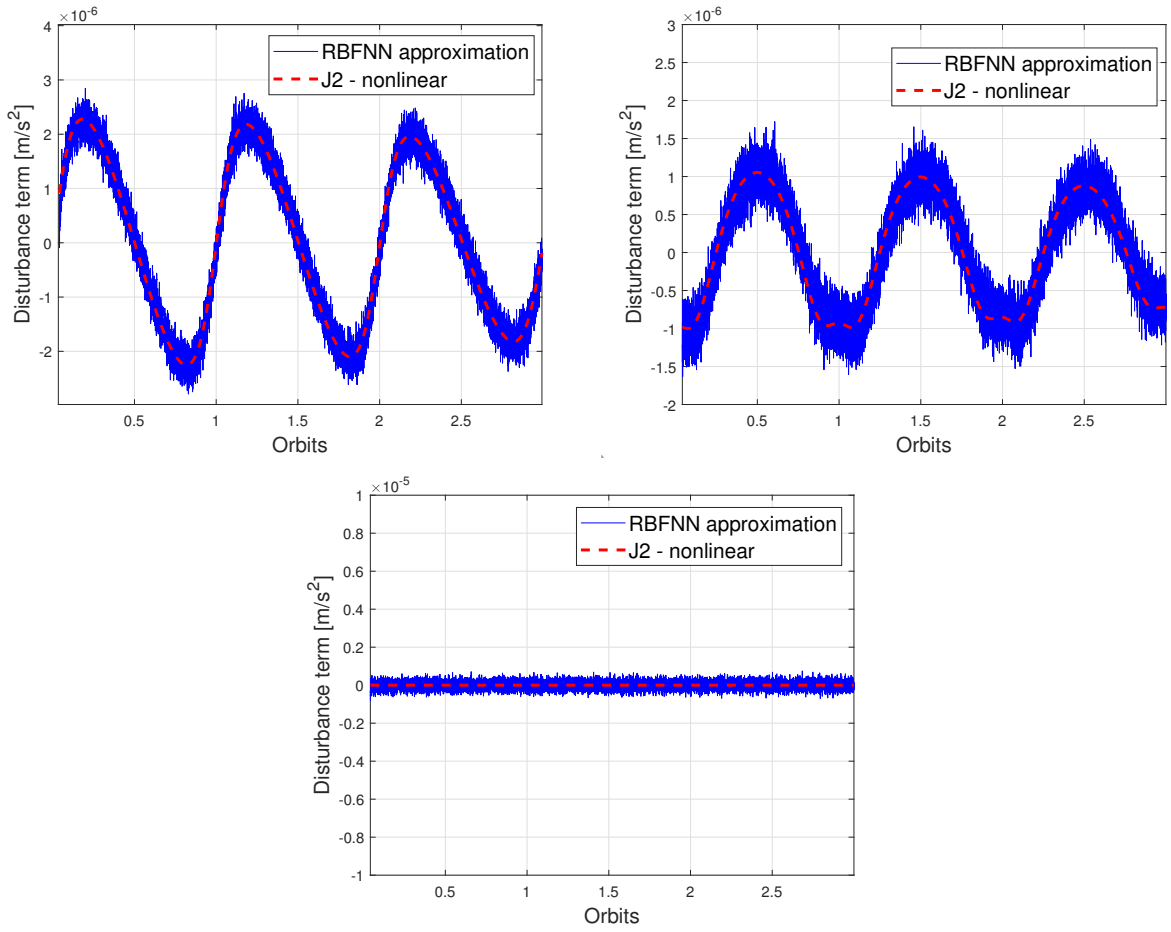

Fig. 5 Estimation of the disturbance acceleration term for LEO reference orbit after the *main learning process*. The plots show the estimation of the disturbance term by the neural network after the network has converged. From left to right: d_x, d_y and d_z.

velocity measurement respectively, is described by a Gaussian distribution with standard deviation $\sigma_{pos} = 10^{-2}m$ and $\sigma_{vel} = 10^{-4}m/s$, similarly to the previous case. It is important to remark that the orbits are eccentric and the cross sectional areas of the two spacecrafts are significantly different, yielding a strong differential perturbation effect due to the solar radiation pressure and drag. The estimation errors, used for performance assessment, are introduced. The relative position error is defined as:

$$e_\rho = \sqrt{(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 + (z_i - \hat{z}_i)^2} \tag{36}$$

where $\hat{x}, \hat{y}, \hat{z}$ are the position components estimates. Similarly, the relative velocity error is:

$$e_{\dot{\rho}} = \sqrt{(\dot{x}_i - \hat{\dot{x}}_i)^2 + (\dot{y}_i - \hat{\dot{y}}_i)^2 + (\dot{z}_i - \hat{\dot{z}}_i)^2} \tag{37}$$

Table 2 Root-mean-squared error of the disturbance estimation term for the LEO reference orbit

	Value
$\sigma_x \, [\frac{m}{s^2}]$	$7.2 \cdot 10^{-7}$
$\sigma_y \, [\frac{m}{s^2}]$	$7.6 \cdot 10^{-7}$
$\sigma_z \, [\frac{m}{s^2}]$	$5.9 \cdot 10^{-7}$

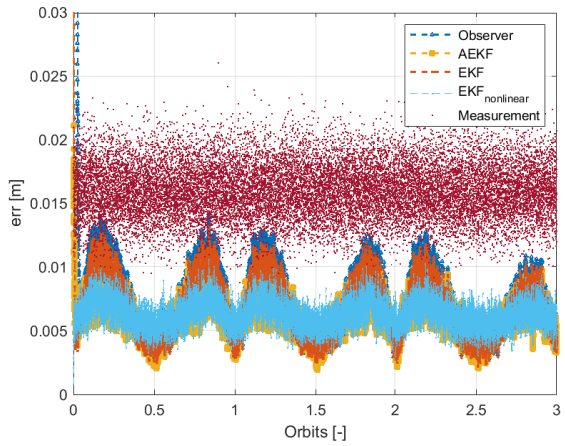

Fig. 6 Relative Position Error

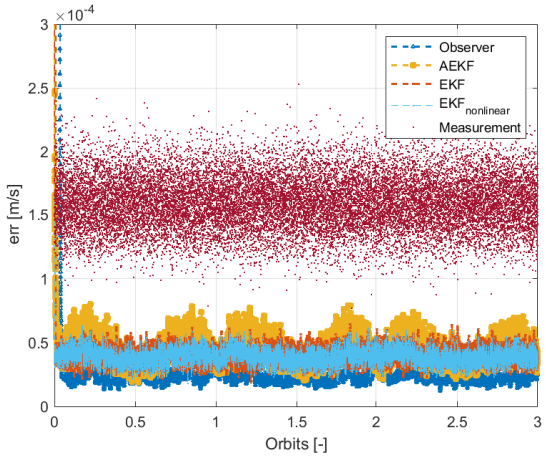

Fig. 7 Relative Velocity Error

with $\hat{x}, \hat{y}, \hat{z}$ are the velocity components estimates.

The measurement covariance matrix $\mathbf{R}$ for all the filters is tuned according to the imposed measurement noise level. The same process covariance matrix $\mathbf{Q}$ is used for the RBFNN-AEKF and RBFNN-EKF and, for the nonlinear EKF, it is properly selected to guarantee the best steady state error performance. Similarly, the observer gain $\mathbf{K_h}$ is tuned to guarantee the minimum steady state error. A statistical analysis of the filters has been performed over 100 runs for the described scenario. The filters run with a frequency of 1Hz and the simulation duration is set to three chaser orbits to appreciate the disturbances effect. Figures 6 and 7 show the relative position and velocity error averaged over 100 runs. For a more quantitative analysis of the results, the Root Mean Square Error (RMSE) starting from time step 300 (at ateady-state) are computed and reported in Table 3 to evaluate the steady state performance of the filters.

Figure 6 and 7 show the beneficial effect of the filters compared to the measurements error. The RBFNN-AEKF and the EKF-nonlinear show a similar behaviour for the relative position error (Figure 6) and, as in Table 3, they outperform the other alternatives. On the other hand, for what concerns the velocity estimation, the Observer, with this tuning, has better performance than the other filters. Despite these small differences, the compared filters show similar performance, and the order of magnitude of the RMSE, reported in Table 3, is the same.

Table 3 Filters RMSE Results

	RMSE - Position [m]	**RMSE** - Velocity [m/s]
Observer	0.0079	$2.39 \cdot 10^{-5}$
RBFNN - AEKF	0.0063	$4.49 \cdot 10^{-5}$
RBFNN - EKF	0.0074	$4.03 \cdot 10^{-5}$
EKF - nonlinear	0.0064	$3.92 \cdot 10^{-5}$

Table 4 Filters RMSE Results - Non-Nominal

	RMSE - Position [m]	**RMSE** - Velocity [m/s]
Observer	0.0149	$5.98 \cdot 10^{-5}$
RBFNN - AEKF	0.0064	$4.79 \cdot 10^{-5}$
RBFNN - EKF	0.0090	$9.34 \cdot 10^{-5}$
EKF - nonlinear	0.0110	$9.55 \cdot 10^{-5}$

D. Relative Navigation - Non-nominal Case

A proper tuning of the filter, however, is difficult to achieve when the process dynamics is not well known and time-varying. Moreover, it is very hard to a-priori determine the accuracy in the estimation that the RBFNN can achieve for that particular case. For this reason, we tested all the filters with off-nominal conditions. In particular, for each simulation, the value of $\mathbf{Q}$ and $\mathbf{K_h}$ were randomly selected according to a uniform distribution centered in the nominal value and spanning two order of magnitudes. This can be a very high uncertainty value for some applications, but we wanted to show how the tuning strongly affects the filter performance. Table 4 shows the relative position and velocity RMSE computed over 100 runs.

It is possible to appreciate how the estimation error of the RBFNN-AEKF is very similar to the nominal case. This is an evidence of high robustness of the proposed solution. On the contrary, all the other filters are badly affected from the inappropriate selection of $\mathbf{Q}$ or $\mathbf{K_h}$ respectively.

V. Concluding Remarks

An original approach for state estimation and uncertainties estimation has been presented. The proposed algorithm relies on a RBFNN coupled with an EKF. The proposed neural-network performs an online estimation of the disturbances acting on the spacecraft, which are included in the prediction step of the filter. The online learning algorithm exploits the state estimation worked out by the filter itself to update the neural network weights. Moreover, an innovation-based recursive filter architecture is employed. Preliminary numerical validation, performance assessment and comparison are carried considering a spacecraft relative navigation scenario. Realistic target/chaser relative dynamics are reproduced. Simulation results show the capability of the proposed solution to reconstruct the dynamics of a spacecraft in elliptic orbits with the J_2 perturbation in an Earth orbit environment. Furthermore, the filter performance is compared to more classical approaches and tested on realistic scenarios, through statistical simulations. Finally, the robustness over very

poor tuning of the state covariance matrix is considered. Satisfactory results are obtained for the proposed solution in all the presented cases and the sensitivity analysis demonstrated the algorithm robustness in non-ideal situations. Future developments aim at transforming the feed-forward neural network into a recurrent structure, which is expected to be significantly more performing when secular disturbance terms become predominant. This paper has shown the application of the navigation algorith, to a Earth-bounded motion but it is applicable to other scenarios.

References

[1] Di Mauro, G., Lawn, M., and Bevilacqua, R., "Survey on Guidance Navigation and Control Requirements for Spacecraft Formation-Flying Missions," *Journal of Guidance, Control, and Dynamics*, , No. December 2017, 2017, pp. 1–22. doi: 10.2514/1.G002868, URL https://arc.aiaa.org/doi/10.2514/1.G002868.

[2] Gurfil, P., Idan, M., and Kasdin, N. J., "Adaptive neural control of deep-space formation flying," *Journal of Guidance Control and Dynamics*, Vol. 26, No. 3, 2003, pp. 491–501. doi:10.2514/2.5072.

[3] Bae, J., and Kim, Y., "Adaptive controller design for spacecraft formation flying using sliding mode controller and neural networks," *Journal of the Franklin Institute*, Vol. 349, No. 2, 2012, pp. 578–603. doi:10.1016/j.jfranklin.2011.08.009.

[4] Zhou, N., Chen, R., Xia, Y., Huang, J., and Wen, G., "Neural network–based reconfiguration control for spacecraft formation in obstacle environments," *International Journal of Robust and Nonlinear Control*, Vol. 28, No. 6, 2018, pp. 2442–2456. doi:10.1002/rnc.4025.

[5] Simon, D., "Training radial basis neural networks with the extended Kalman filter," *Neurocomputing*, Vol. 48, No. 1-4, 2002, pp. 455–475.

[6] Stubberud, S. C., Lobbia, R. N., and Owen, M., "An adaptive extended Kalman filter using artificial neural networks," *Decision and Control, 1995., Proceedings of the 34th IEEE Conference on*, Vol. 2, IEEE, 1995, pp. 1852–1856. doi: 10.1109/cdc.1995.480611.

[7] Gao, X., Zhong, X., You, D., and Katayama, S., "Kalman Filtering Compensated by Radial Basis Function Neural Network for Seam Tracking of Laser Welding," *IEEE Transactions on Control Systems Technology*, Vol. 21, No. 5, 2013, pp. 1916–1923. doi:10.1109/tcst.2012.2219861.

[8] Stubberud, A., Wabgaonkar, H., and Stubberud, S., "A neural-network-based system identification technique," *Decision and Control, 1991., Proceedings of the 30th IEEE Conference on*, IEEE, 1991, pp. 869–870. doi:10.1109/cdc.1991.261441.

[9] Dah-Jing, J., and Chen, J.-J., "Neural network aided adaptive Kalman filter for GPS/INS navigation system design," *Proc. 9th IFAC Workshop*, 2011, pp. 1–7.

[10] Jwo, D.-J., and Huang, H.-C., "Neural Network Aided Adaptive Extended Kalman Filtering Approach for DGPS Positioning," *Journal of Navigation*, Vol. 57, No. 3, 2004, pp. 449–463. doi:10.1017/s0373463304002814.

[11] Harl, N., Rajagopal, K., and Balakrishnan, S. N., "Neural Network Based Modified State Observer for Orbit Uncertainty Estimation," *Journal of Guidance, Control, and Dynamics*, Vol. 36, No. 4, 2013, pp. 1194–1209. doi:10.2514/1.55711.

[12] Wu, Y., Wang, H., Zhang, B., and Du, K.-L., "Using Radial Basis Function Networks for Function Approximation and Classification," *ISRN Applied Mathematics*, Vol. 2012, No. March, 2012, pp. 1–34. doi:10.5402/2012/324194.

[13] Park, J., and Sandberg, I. W., "Universal Approximation Using Radial-Basis-Function Networks," *Neural Computation*, Vol. 3, No. 2, 1991, pp. 246–257. doi:10.1162/neco.1991.3.2.246.

[14] Reif, K., and Unbehauen, R., "The extended Kalman filter as an exponential observer for nonlinear systems," *IEEE Transactions on Signal processing*, Vol. 47, No. 8, 1999, pp. 2324–2328.

[15] Simon, D., *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*, John Wiley & Sons, 2006.

[16] Pesce, V., Lavagna, M., and Bevilacqua, R., "Stereovision-based pose and inertia estimation of unknown and uncooperative space objects," *Advances in Space Research*, Vol. 59, No. 1, 2017, pp. 236–251.

[17] Pesce, V., Opromolla, R., Sarno, S., Lavagna, M., and Grassi, M., "Autonomous Relative Navigation Around Uncooperative Spacecraft Based on a Single Camera," *Aerospace Science and Technology*, 2018.

[18] Pesce, V., Haydar, M. F., Lavagna, M., and Lovera, M., "Comparison of filtering techniques for relative attitude estimation of uncooperative space objects," *Aerospace Science and Technology*, Vol. 84, 2019, pp. 318–328.

[19] Akhlaghi, S., Zhou, N., and Huang, Z., "Adaptive adjustment of noise covariance in Kalman filter for dynamic state estimation," *arXiv preprint arXiv:1702.00884*, 2017.

[20] Curtis, H., *Orbital Mechanics for Engineering Students*, Elsevier, 2005.

[21] Wang, D., Wu, B., and Poh, E. K., *Satellite Formation Flying*, Vol. 87, Intelligent Systems, Control and Automation: Science and Engineering, 2017. doi:10.1007/978-981-10-2383-5.

[22] Burns, R., *Advanced control engineering*, Elsevier, 2001.