IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 20 February 2023, accepted 17 March 2023, date of publication 27 March 2023, date of current version 30 March 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3262023

RESEARCH ARTICLE

Long Short-Term Memory-Based Neural Networks for Missile Maneuvers Trajectories Prediction*

DARIO GIUSEPPE LUI[1], GAETANO TARTAGLIONE[2], (Member, IEEE), FRANCESCO CONTI[3], GIANMARIA DE TOMMASI[1], (Senior Member, IEEE), AND STEFANIA SANTINI[1], (Member, IEEE)

[1]Department of Electrical Engineering and Information Technology, University of Napoli Federico II, 80125 Naples, Italy
[2]Department of Engineering, University of Napoli Parthenope, 80143 Naples, Italy
[3]MBDA Italia SpA, 00131 Rome, Italy

Corresponding author: Stefania Santini (stefania.santini@unina.it)

ABSTRACT Due to its extensive applications in different contexts, moving target tracking has become a hot topic in the last years, above all in the military field. Specifically, missile tracking research received a great effort, mainly for its importance in terms of security and safety. Herein, traditional solutions, e.g. Interacting Multiple Model (IMM) based on the Kalman estimation theory, achieve good performance under the main restrictive assumption of the *a priori* knowledge of the target model, so neglecting the unavoidable presence of model uncertainties and limiting the achievable tracking accuracy only by the presence of the measurement noise. With the specific aim of overcoming this narrowness, this work investigates the capability of deep neural networks in predicting the missile maneuvering trajectories in a model-free fashion. The idea is to leverage the Long-Short Term Memory (LSTM) net due to its excellent capability in learning long-term dependencies of temporal information. Two different LSTM-based architectures have been hence designed to predict both position and velocity of a missile using raw and noisy measurements provided by a realistic radar system, exploiting a large database abundant of realistic off-line data. Training results and theoretical derivations are verified through non-trivial scenarios in order to assess the capability of predicting unknown and realistic 3D missile maneuvers. Finally, the proposed approach has been also compared with a performing model-based IMM algorithm, suitably tuned to deal with realistic missile maneuvers, confirming the excellent generalization abilities of the developed data-driven architectures for different datasets.

INDEX TERMS Tracking problem, ballistic missile, long-short term memory neural-network.

I. INTRODUCTION

The target tracking problem has received great attention from the research community in the last years, due to its wide application in both military and civilian fields, such as space and weather monitoring, traffic control, remote sensing, autonomous vehicles, robotics, and so on [1]. Among them, the application in a secure field as missile fast-tracking has known a great effort over the last decades, since it is crucial to intercept and shoot down a flying enemy missile successfully.

Since the tracking of a moving target is performed by processing the measurements of the available sensors, such as radar, sonar and camera, corruption generated by random noise is unavoidable. Under the quite restrictive assumption of regular target motion and white Gaussian distributions for the process and the measurement noise, most solutions proposed for this problem are based on the Kalman Filter (KF) theory [2]. However, when target trajectories have been characterized by great complexity and diversity and vary

The associate editor coordinating the review of this manuscript and approving it for publication was Chao Tong.

*This work has been partially carried out in collaboration with MBDA Systems Inc.

unexpectedly, classical KF approaches, which are based on a single dynamical model, do not achieve satisfactory performance [3], [4]. To overcome this issue, different Multiple Model (MM) approaches have been proposed from the technical literature and the Interacting Multiple Model (IMM) is one of the most computationally efficient strategy [5]. The IMM approach, originally proposed in [6], is a suboptimal hybrid estimator which consists of a bank of multiple KFs, each of them matched on a specific target model, that tracks the target motion through a weighted average KFs estimation using a probability model. To exploit its potentiality, many studies have been carried out on tracking estimation through IMM, ranging from the classification of dynamical characteristics of different targets such as drones, jets, and civil aircrafts (see references in [3]) to the electric vehicles for the stable steering control problem [7] until to the multi-fault diagnosis of lithium-ion batteries in [8].

Focusing on the fast missile tracking problem, different IMM strategies have been proposed in the technical literature. For example, in [3] the authors design an IMM, disclosing better tracking performance w.r.t. a KF based on a single dynamic model; the Interacting Multiple Model Particle Filtering (IMMPF, [9]) is applied to track ballistic missile motion in [10], and, more recently, the same tracking problem is solved during the boost phase in [11], where a new modified IMM based on Unscented Kalman Filter (UKF) is proposed. Moreover, a novel state-dependent IMM based on Gaussian particle filtering is developed [12] to estimate the motion information describing the ballistic missile, such as the phase of flight, position, velocity, and parameters.

Most of the approaches based on MM algorithms provide acceptable tracking performances only when the dynamic target motion can be described by a small set of models, otherwise, a degradation appears with the computational burden increment [13]. Furthermore, being a model-based approach, the most shortcoming of the IMM algorithm arises when models used are not representative of the target dynamics [13], [14]. Indeed, since the system model is only an approximation of the real plant, practical problems are inevitably affect by errors modeling, which can significantly compromise the the estimation quality [4]. This implies that the estimation accuracy for IMM approaches is limited not only from measurement errors and noise, but also from the unavoidable presence of model uncertainties arising from neglected or simplified dynamics, as well as uncertain parameter values. Besides these facts, regardless of the adopted solution, in order to evaluate the most trustworthy filter, a crucial role in the definition and operation of the IMM algorithm is assumed by the underlying Transition Probability Matrix (TPM), whose tuning remains a difficult task to be accomplished by leveraging a priori information and/or dedicated analysis, in addition to base the setting on two strong assumptions: *i*) the time-varying probability of the TPM transitioning among models are well represented by a constant value; *ii*) this constant value is well known a priori [11].

On the other side, recently, the developments in the Deep Learning (DL) field have brought significant advantages in different areas, including computer vision, driverless car, speech processing, machine health monitoring, signal processing, and so on (e.g., the interested readers can consult [15] and the references therein). Within this wide context, different DL algorithms have been proposed with the aim of extracting features from complex and abstract data via general-purpose learning algorithm [15], without artificially designing feature patterns, as in traditional Machine Learning techniques [16].

Among the possible approaches, Deep Neural Networks (DNNs) have recently proved this claim [17], and, so, they have been exploited to reach better performance both with Convolutional Neural Networks, which currently represents the main approach in tasks as image classification [18] or object detection [19], [20], and Recurrent Neural Networks (RNNs), mainly used to process sequential data, as temporal series, audio signals, etc. [21].

Since the missile target tracking problem can be seen as a sequence problem [14], RNNs could be employed to handle this task [22] and, unlike conventional model-based methods, allows to learn the correct behavior from the available training data in a model-free fashion, facing both the issues of measurement noise and model uncertainties, and without any a priori knowledge on the probabilistic noise distribution [2]. However, the training of standard RNNs suffers for well-known problems of *vanishing gradient* and *exploding gradient*, due to the difficulties for the gradients to propagate far in a lot of time steps consistently with an acceptable range [23], thus considerably limiting the applicability of these standard nets. While the exploding gradient can be avoided placing strong constraints on the gradient norm [24], the vanishing gradient problem can be deal leveraging the Long Short-Term Memory (LSTM) networks [25], initially proposed by Hochreiter and Schmidhuber in 1997 in their seminal work [26] (without the forget gate) and, after, perfected by Graves and Schmidhuber in [27] with the introduction of an additional forget gate, thus realizing the most common LSTM architecture used nowadays. Note that, the excellent qualities in learning of the LSTM nets justify their application in different contexts, including Remaining Useful Life (RUL) estimation [23], [28], [29], fault detection and isolation [30], forecasting of vehicles' emissions [31], prediction of water table depth in agriculture [32], and stability prediction of a smart grid [33].

Very Recently, LSTM have been also proposed for solving a generic target tracking problem with the aim to have a powerful tool able dealing with the target motion uncertainties, conversely to traditional MM-based approaches, which suffer from serious degradation in the presence of model mismatches. More specifically, authors in [14] combine the theory of DNN and traditional tracking filters, proposing an LSTM-IMM model algorithm, while a bidirectional LSTM approach was considered in [34]. The authors in [13], instead, propose two different LSTM-based approaches to solve the tracking problem of a manoeuvring target, and a comparison with IMM shows that the proposed systems achieve better performance.

Inspired by the aforementioned facts in this paper, to the best of the authors' knowledge, we investigate for the first time the capability of an LSTM-based approach to predict the trajectory of a ballistic missile over a short time interval. Specifically, two alternative LSTM-based systems have been designed for the *online prediction* of the position and velocity of a missile. Moreover, a performance analysis is carried out for carefully characterizing the suitable LSTM-NNs systems, by varying their structure, i.e. number of hidden layers and neurons, activation functions, dimensions of the parameters hyperspace that define the training data set of missile trajectories, etc. More notably, in order to account for the real environment, we consider a careful characterization of the noisy radar measurements. In this way, the training and the performance assessment of the LSTM-based systems are performed by using the raw and noisy data provided by a radar system and parametric trajectories of a missile moving in a 3D space during the re-entry phase.

Finally, a comparative analysis with respect to a state-of-the-art IMM system has been carried out considering trajectories related to different classes of maneuvers.

The results of this analysis show confirm that data-driven LSTM approaches are able to tackle and solve the fast missile tracking problem overcoming the issue of both the measurement degradation and model uncertainties.

Based on the above discussion, the main contributions of this paper can be summarized as follows:

- first, an LSTM-based model-free approach is proposed in order to handle the fast-tracking problem of a ballistic missile for different truthful and non-trivial maneuvering scenarios, exploiting the raw measurement data provided by a realistic radar model [35];
- second, the excellent LSTM nets capabilities in learning long-term dependencies of temporal information are exploited to develop two different architectures, with the aim to overcome the traditional model-based drawback in dealing mainly with the measurement errors, neglecting the unavoidable model uncertainties;
- third, a comparison analysis w.r.t. a traditional model-based IMM solution, developed to deal with realistic and non-trivial scenarios considered [35], is provided to better highlight the powerful performance of the proposed data-driven methodology.

The rest of the paper is organized as follows. In Section II the problem statement is provided. In Section III a background on the LSTM theory is given to facilitate understanding of the work, while in Section IV the proposed data-driven solutions for the missile tracking problem are described. In Section V the tuning problem of the NNs parameters is tackled and the training problem is defined. Section VI is dedicated to the assessment of the performance of the proposed systems, through non-trivial simulations. Moreover, in Section VII a comparison analysis w.r.t. a classical IMM-based solution is provided, Finally, conclusions are drawn in Section VIII.

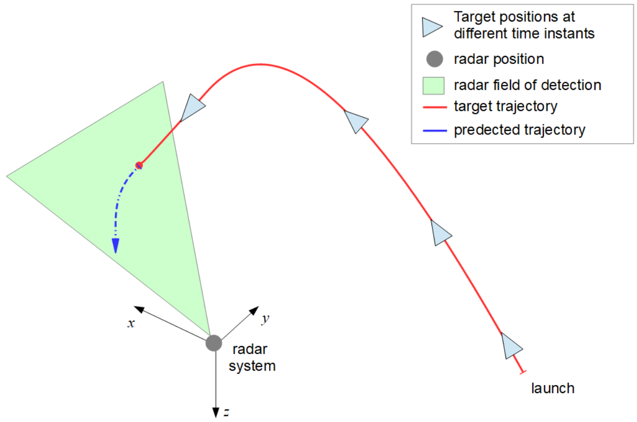

FIGURE 1. Schematic representation of the missile fast-tracking problem. Red solid line: missile trajectory from the launch to the actual position (red point). Green area: field of detection of the radar system. Blu dashed line: manoeuvring trajectory to be predicted.

II. PROBLEM STATEMENT AND MOTIVATION

The solution of the missile fast-tracking problem requires the correct online estimate of the states of the mobile target exploiting real-time measurements gathered from sensors. More in detail, let's consider an enemy ballistic missile during the flight *re-entry phase* as a moving target to intercept within the field of detection of the radar system (see Fig. 1). The aim is to predict in real-time the target trajectory within the radar field of view over a finite and short time interval by online processing raw and noisy radar measurements and uploading the prediction when the new measurements are available.

As also pointed out in the introduction section, traditional model-based tracking methods base their functionality on the assumption that the target motion, and corresponding measurements, can be represented with a sufficient accuracy by known mathematical models. Along this line, some approaches exploit a linear sets of state equations, while in a more general framework some nonlinear relations among the kinetic variables have to be exploited for emulating the underlying moving target trajectories, as well the relations among measurements. So, in this traditional missile tracking framework, the target is commonly assumed to be a point (e.g., see for instance [36], [37], [38]) whose dynamics belong to a finite set of m nonlinear models as:

$$x_{k+1} = \phi_{j_k} x_k + w_k \qquad j_k \in [1, \ldots, m], \tag{1}$$

where x_k is the state vector at the k-th timestep, the process noise w_k is assumed to be additive and ϕ_{j_k} is the nonlinear dynamics associated to the i_k model, while the measurements are assumed to be model dependent and related to the true

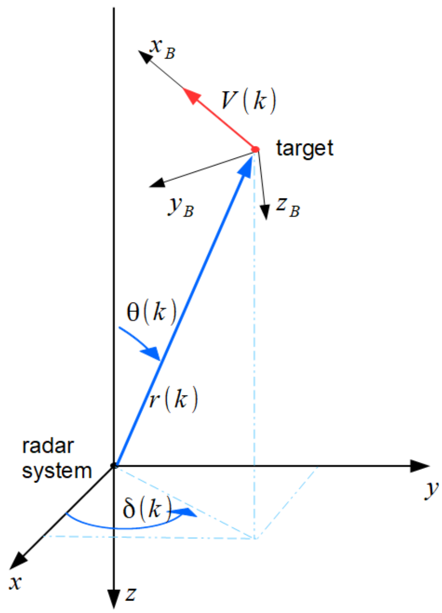

FIGURE 2. Measurements frames. Red dot and red arrow: position and velocity vector of the target in the radar frame, respectively. Blue arrows: target measurements provided by the radar system in terms of range $r(k)$, inclination $\theta(k)$ and azimuth $\delta(k)$.

states as:

$$z_k = \chi_k(x_k) + \nu_k \tag{2}$$

where z_k are the measurements gathered from sensors at time k, ν_k is the measurement noise and χ_k characterizes the sensing system behavior.

Despite model-based approaches could work quite well in general cases, some open challenges arise when specifically focusing on the design of a model-based tracking system for the missile defense. Herein it is necessary to track fast and unknown objects when no a priori information about the hostile missile is available. Indeed, without this knowledge, it is hard to make the model, as in (1)-(2), on which the design of the estimator is grounded. It follows that model-based approached do not perfectly fit the missile tracking problem, where not only accurate nonlinear target models are usually not available to the tracker, but the target can also exhibit a very wide and rich variety of nonlinear behaviors, often resulting in abrupt changes in its trajectory, such as sudden directional changes of the evasive target. Eventually they usually suffer from the presence of uncertainties and neglected or unknown dynamics (e.g., resulting from the very restrictive point-mass assumption), as well as from the fact that the very rich dynamical behavior of the underlying target can not be well approximated by a small set of linear or nonlinear models.

For the above reasons, in this study, we tackle the tracking problem of a ballistic missile by investigating the feasibility of a fully data-driven solution. In this context, we take into account that the target is able to perform a wide range of maneuvers, also belonging to the following main classes or any possible combination:

- *pull-up pull-down (PUPD)*: a maneuver performed with a nonzero lateral acceleration along the z_B axis of the body system reference (see Fig. 2 and Fig. 3);
- *cross*: a maneuver performed with a nonzero lateral acceleration along the y_B axis of the body system reference (see Fig. 2 and Fig. 4);
- *ballistic*: a maneuver performed with null lateral accelerations.

Hence, under the main assumption that the target missile is unknown, we design an LSTM-based strategy able to provide the short-term prediction of the target by only exploiting real-time measurements. Namely, the strategy processes in real-time the noisy information about the missile position and velocity in a local North, East, Down (NED) reference system in which the origin is fixed at the radar position (see Fig. 2), say $\tilde{p}(k) \in \mathbb{R}^3$ and $\tilde{v}(k) \in \mathbb{R}^3$, respectively, in order to obtain a prediction of the next position and velocity of the target over a short-time interval, say $\hat{p}(k+1|k) \in \mathbb{R}^3$ and $\hat{v}(k+1|k) \in \mathbb{R}^3$ respectively.

It is worth noting that the radar system only provides noisy measurements about the position of an object which moves in its field of perception, with a given sampling rate T [s], and that this measure is provided in a spherical reference system whose origin is fixed at the radar position [36] (see Fig. 2) as:

$$\tilde{r}(k) = r(k) + n_r(k)\,, \tag{3a}$$
$$\tilde{\theta}(k) = \theta(k) + n_\theta(k)\,, \tag{3b}$$
$$\tilde{\delta}(k) = \delta(k) + n_\delta(k)\,, \tag{3c}$$

where the normal distribution of the stochastic variables $n_r(k)$, $n_\theta(k)$ and $n_\delta(k)$ are characterized by setting the corresponding mean values, i.e. μ_r, μ_θ and μ_δ, and standard deviations, i.e. σ_r, σ_θ and σ_δ.

The noisy measurements of the position in the NED reference system, i.e. $\tilde{p}(k) = \left[\tilde{x}(k)\ \tilde{y}(k)\ \tilde{z}(k)\right]^T$ [m], can be than obtained from the corrupted spherical coordinates as:

$$\tilde{x}(k) = \tilde{r}(k)\cos\tilde{\delta}(k)\sin\tilde{\theta}(k)\,, \tag{4a}$$
$$\tilde{y}(k) = \tilde{r}(k)\sin\tilde{\delta}(k)\sin\tilde{\theta}(k)\,, \tag{4b}$$
$$\tilde{z}(k) = -\tilde{r}(k)\cos\tilde{\theta}(k)\,. \tag{4c}$$

Finally, the velocity information $\tilde{v}(k)$ can be derived from position as:

$$\tilde{v}(k) = \frac{\tilde{p}(k) - \tilde{p}(k-1)}{T}\,. \tag{5}$$

III. LONG SHORT-TERM MEMORY NETWORKS: BACKGROUND

LSTM neurons replace the traditional RNN structure, with the same repeating chain modules, but with a more complicated function for the presence of three layers and internal self-loops interacting among them in a specific way to determine what information should be recalled (see a generic

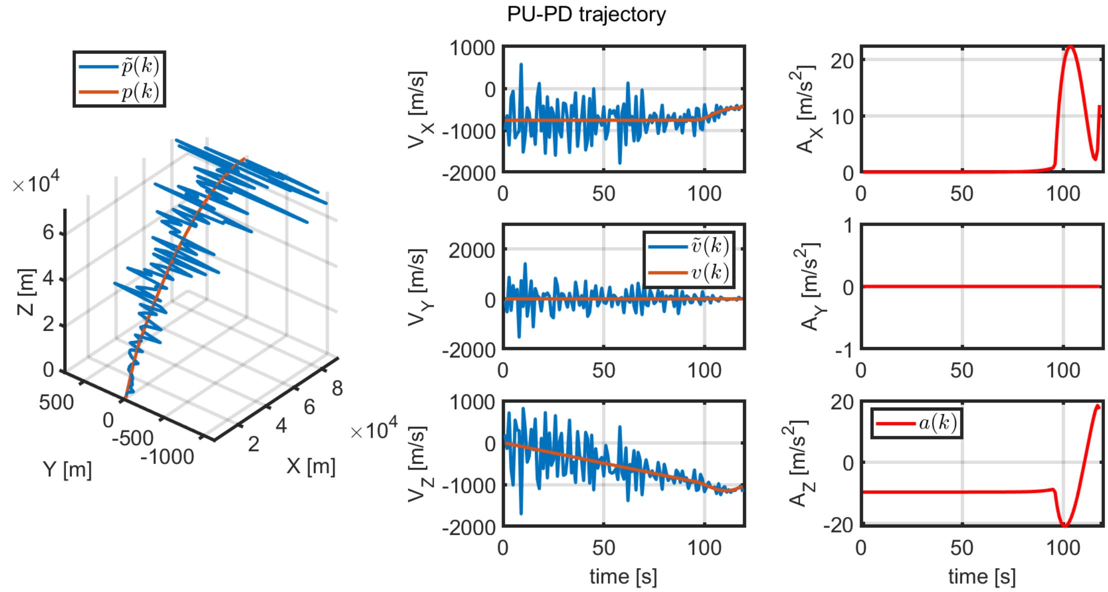

FIGURE 3. PUPD maneuver (Left frame: position. Central frame: velocity. Right frame: missile acceleration. Red: raw target 3D position data. Blue: noisy radar measurements.

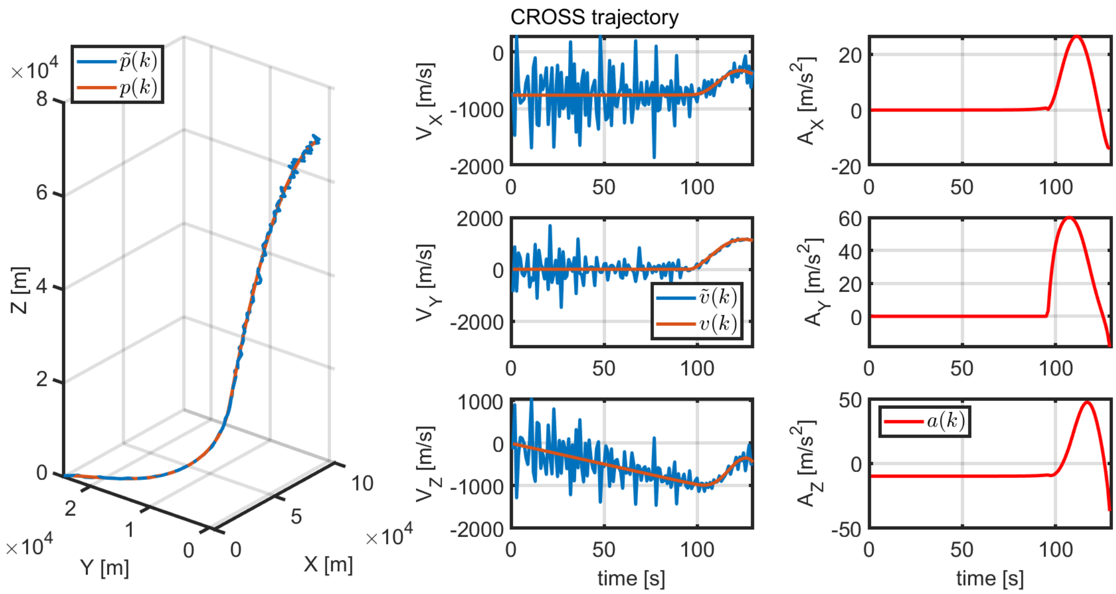

FIGURE 4. CROSS maneuver (Left frame: position. Central frame: velocity. Right frame: missile acceleration. Red: raw target 3D position data. Blue: noisy radar measurements.

LSTM module structure in Fig. 5). In so doing, the structure allows to remember previous information and to process arbitrary long-time sequences [32], making LSTM nets perfect for learning long-time dependencies. It is worth noting that alternative approaches such as RNNs cannot capture long-time dependencies in data [28] due to the well-known "vanishing gradient" problem affecting the backpropagation process during the training phase. On the contrary, in LSTM each repeating module has a memory block with self-connections, designed to store information over long time periods, allowing to learn long-term dependencies more easily than classical RNNs, since the information can flow more easily. More in detail, the *input gate* controls the flow of input activations within the memory cell and is able to select what information have to be stored in the internal state, while the *output gate* determines the output flow of cell activations into the rest of the network, identifying the output information. Finally, the *forget gate* is able to determine what information needs to be discarded from neuron states [39].

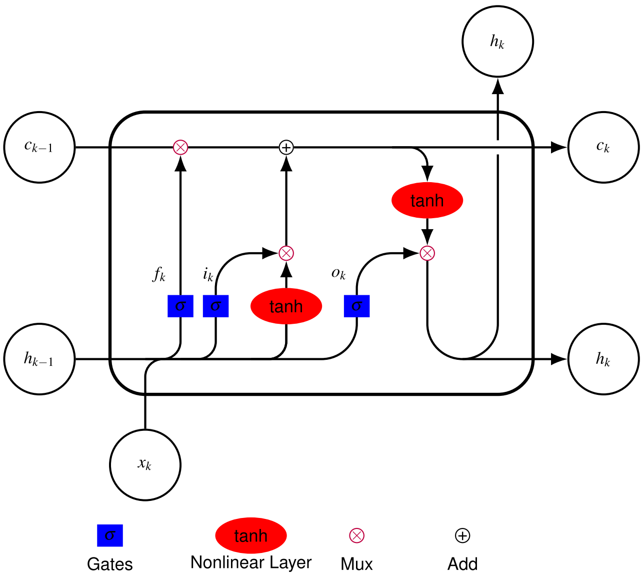

FIGURE 5. Schematic of a repeating module in an LSTM. The cell state c_k let the information to flow unchanged along the module. The forget, input and output gates (i.e. f_k, i_k and o_k, respectively) control information flows through the module. The hidden state h_{k-1} determines how much information to forget.

The LSTM neuron defines a mapping from an input sequence x_k to output by iteratively evaluating the following computational process, which allows us to determine the forget gate f_k, the input gate i_k, the output gate o_k, the cell state c_k and the hidden state h_k at time k, as [40]:

$$f_k = \sigma(W_f \cdot [h_{k-1}, x_k] + b_f), \tag{6a}$$
$$i_k = \sigma(W_i \cdot [h_{k-1}, x_k] + b_i), \tag{6b}$$
$$c_k = f_k \odot c_{k-1} + i_k \odot \tanh(W_c \cdot [h_{k-1}, x_k] + b_c), \tag{6c}$$
$$o_k = \sigma(W_o \cdot [h_{k-1}, x_k] + b_o), \tag{6d}$$
$$h_k = o_k \odot \tanh(c_k), \tag{6e}$$

where W_γ are learnable parameters and denotes the weight matrices from the forget, input, output and cell gates to the input ($\gamma = f, i, o, c$); b_γ are the forget, input, output and cell bias vectors ($\gamma = f, i, o, c$); the $\odot$ operator indicates the element-wise product; $\sigma(\cdot)$ is an element-wise nonlinear activation function, i.e. logistic sigmoid function. Along the line of the DL paradigm, which allows capturing more potential features from input data exploiting a multi-layer network structure [41], multiple LSTM hidden layers - each composed by numerous LSTM neurons as in Fig. 5 - can be stacked to construct a more complex module, named the Deep Long Short-Term Memory (DLSTM) network so to carry out a deep data fusion of sensor input. Fig. 6 shows an exemplar DLSTM architecture.

IV. LSTM-BASED PROPOSED APPROACHES FOR MISSILE TRACKING

Here we present two alternative LSTM architectures purposely designed with the aim of *on-line* solving the fast-tracking problem of an unknown missile during

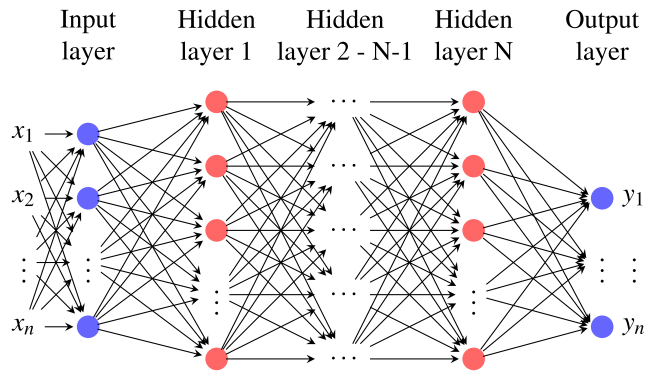

FIGURE 6. Exemplar DLSTM architecture. Each node has a structure as in Fig. 5.

maneuvers, described in Section II, by *sequentially on-line* processing the actual noisy radar measurements.

The idea behind the first architecture, named *Coupled Missile Fast-Tracker* (CMFT) and depicted in Fig. 7, is to leverage a single network (6 inputs - 6 outputs) for estimating and forecasting all the kinematic variables of the unknown moving target. Namely, given radar data collected at each sampling instant k, say $\tilde{p}(k)$ and $\tilde{v}(k)$, the single LSTM-system provides the prediction of both the position and velocity vectors at the next time step $k+1$, say $\hat{p}(k+1|k)$ and $\hat{v}(k+1|k)$.

Conversely, the second architecture, named *Decoupled Missile Fast-Tracker* (DMFT), follows a decoupled approach based on the *kinematic* separation of the output variables. This reflects in leveraging two different LSTM-based modules running in parallel (see Fig. 8). Each module (6 inputs - 3 outputs) is devoted to provide only one of the state variables that describe the missile motion. Namely, exploiting the noisy measurements about the target position and velocity collected from the radar at each discrete time instant k, Modules 1 and 2 are designed for predicting the position and velocity vectors at time $k+1$, $\hat{p}(k+1|k)$ and $\hat{v}(k+1|k)$.

Motivations for investigating the above two alternative architectures lie in the well-known complexity of the tracking problem. The idea is to understand if by exploiting the kinematic decoupling of the predicted variables, is possible to achieve a better forecast and gain greater accuracy for the same complexity of neural networks since the two different prediction problems are independently solved.

Further details about the design of both the architecture, such as the selection of a proper structure for the LSTM-NNs (i.e. number of hidden layers, number of neurons, activation functions, etc.), as well as about tuning, training and validation phases are provided in the following sections.

V. SETUP AND TUNING OF THE DATA-DRIVEN ARCHITECTURES

The LSTM-NNs structures of both the CMFT and DMFT architectures have been appropriately defined taking into

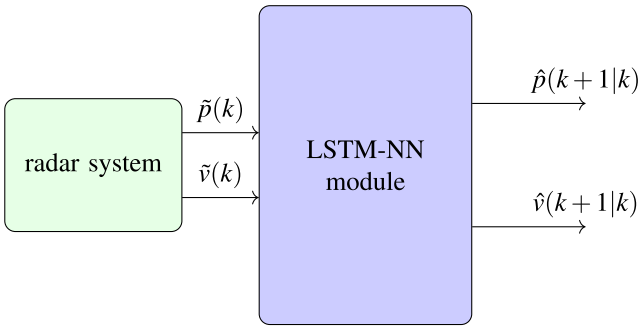

FIGURE 7. CMFT (Coupled Missile Fast Tracker). The architecture leverages a single network for the fast-tracking and prediction of all the missile kinematic variables.

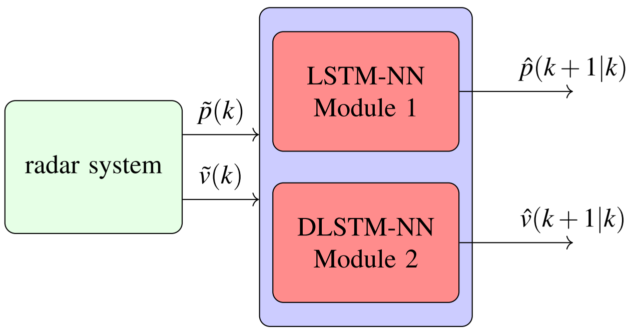

FIGURE 8. DMFT (Decoupled Missile Fast Tracker). The architecture leverages two LSTM-based Modules, each one devoted to the fast-tracking and prediction of a specific kinematic variable of the missile motion.

account a trade-off between complexity and performance with respect to the training time. The iterative procedure has involved tuning the hyperparameters in search of a combination that is most consistent and achieves the best model performance. To this aim, a careful analysis has been carried out for evaluating the impact on the performance of changes in the NNs features, i.e. the number of layers and neurons, the activation functions, and so on.

The results of this iterative procedure have been summarized in Table 1 for the CMFT, and in Table 2 and Table 3 for the DMFT. Note that, with respect to the design of the DMFT architecture, the two LSTM-NN modules are not identical, since a deeper structure is required to correctly reconstruct velocity information. Finally, the dropout technique, which is usually seen as a regularization technique, has been exploited for reducing the overfitting [13].

The sampling time T has been fixed at 1.0 $[s]$ according to the characteristic of the radar system [36].

Once the structure has been defined for both the proposed CMFT and DMFT architectures, their related weights matrices have been optimized during the training phase. This optimization problem has been defined as minimizing the prediction error with respect to a set of training trajectories, and then solved via the Stochastic Gradient with Momentum method [42] with optional parameters as in Table 4. This iterative method computes the neural network parameters

TABLE 1. CMFT: network architecture parameters.

Layer	Parameter	Value
sequence input layer	input size	6
LSTM layer	number of neurons state activation function gate activation function	5000 hyperbolic tangent sigmoid
dropout layer	dropout probability	1%
fully connected layer	number of neurons	5000
regression output layer	loss function output size	mean squared error 6

TABLE 2. DMFT: LSTM-NN Module 1 (position).

Layer	Parameter	Value
sequence input layer	input size	6
LSTM layer	number of neurons state activation function gate activation functions	5000 hyperbolic tangent sigmoid
dropout layer	dropout probability	1%
fully connected layer	number of neurons	5000
regression output layer	loss functions output size	mean squared error 3

TABLE 3. DMFT: LSTM-NN Module 2 (velocity).

Layer	Parameter	Value
Sequence input layer	input size	6
LSTM layer 1	number of neurons state activation function gate activation function	1200 hyperbolic tangent sigmoid
LSTM layer 2	number of neurons state activation function gate activation function	900 hyperbolic function sigmoid
LSTM layer 3	number of neurons state activation function gate activation function	600 hyperbolic function sigmoid
LSTM layer 4	number of neurons state activation function gate activation function	300 hyperbolic function sigmoid
dropout layer	dropout probability	1%
fully connected layer	number of neurons	300
regression output layer	loss function output size	mean squared error 3

by minimizing the loss function. Namely, at each iteration, the algorithm evaluates the gradient of the loss function using a subset of the training data, called *mini-batch*, and it updates the optimization variables by taking small steps in the direction of the negative gradient. Note that, commonly the stochastic gradient descent algorithm can oscillate along the path of steepest descent towards the optimum and so adding a momentum term to the parameter update helps reduce this oscillation [42].

The training and validation data-sets are composed of different kinds of target motion trajectories ensuring that the tracking of targets with different motions can be handled with the proposed LSTM-based architecture (e.g., the coupled or decoupled architecture, CMFT and DMFT respectively). The maneuver-based trajectories have been generated by using an advanced simulation tool for ballistic missiles, which has been developed and validated by MBDA-Italy [35]. In order to collect data from a wide range of trajectories, simulations have been started from different initial data and input signals,

TABLE 4. Parameters of the Stochastic Gradient with Momentum method.

Parameter	Value
maximum number of epochs	250
mini batch size	20
gradient threshold	1
shuffle	once
learn rate schedule	piecewise
initial learn rate	0.1
learn rate drop period	15
learn rate drop factor	0.5
validation frequency	100
validation patience	50

in order to perform different realistic maneuvers. Specifically, the trajectories of the data-set are characterized by the following dynamic features [35]:

- the missile ballistic coefficient is in the range

$$\left[20000\,,\ 35000\right]\ [kg/m^2]\,;$$

- starting maneuver altitude is in the range

$$\left[25000\,,\ 35000\right]\ [m]\,;$$

- the impact points of the PUPD trajectories are

$$IP_{PUPD} = \left[IP_x\ \ 0\ \ 0\right]^T\ ,$$

being $IP_x\ \in\ \left[-16500\ 4500\right]\ [m]$;
- the impact points of the cross trajectories are

$$IP_{cross}\ =\ \left[0\ IP_y\ 0\right]^T\ ,$$

being $IP_y \in \left[-25000\ 25000\right]\ [m]$,

while initial conditions in terms of position and velocities (p_0 and v_0) are in the following ranges:

- $p_0 = \left[100000\ 0\ 70000\right]^T\ [m]$;
- $\dot{p}_0 = \left[V_{x0}\ 0\ V_{z0}\right]^T$,

being

$$V_{x0} \in \left[-1200\,,\ -600\right]\ [m/s]$$

and

$$V_{z0}\ \in\ \left[-1200\,,\ -200\right]\ [m/s]\,.$$

A set of 36000 trajectories have been considered, including 12000 ballistic trajectories, 12000 PUPD trajectories, and 12000 cross ones. 30000 out of the considered 36000 trajectories have been included in the training data-set, while the remaining 6000 constitute the validation data-set. Note that each trajectory is a discrete-time signal defined by a sequence of N position, velocity and acceleration vectors (say $p(k)$, $v(k)$ and $\ddot{p}(k)$, respectively), where $k = 0, \ldots, N$, being N the discrete time instant when the missile hits the ground.

According to the characteristics of the radar system (3), position and velocity measurements are corrupted by an additive noise that in spherical coordinates can be expressed as a stochastic signal with zero mean and standard deviations as:

$$\sigma_r = 20\ [m]\,, \tag{7a}$$
$$\sigma_\theta = 6\cdot 10^{-3}\ [rad]\,, \tag{7b}$$
$$\sigma_\delta = 6\cdot 10^{-3}\ [rad]\,. \tag{7c}$$

Note that, although the radar noise directly affect only the position measurement, obviously it has also effect on the measured speed of the missile.

An exemplar three-dimensional cross trajectory is shown in Fig. 4, and in Fig. 3 is shown a PU-PD trajectory. Moreover, the above stochastic characterization of the radar noise allows to take into account worst conditions than the typical ones proposed in the current technical literature, see for instance [36], [43], allowing to analyze the prediction performances in a more realistic scenarios, as stated in [35].

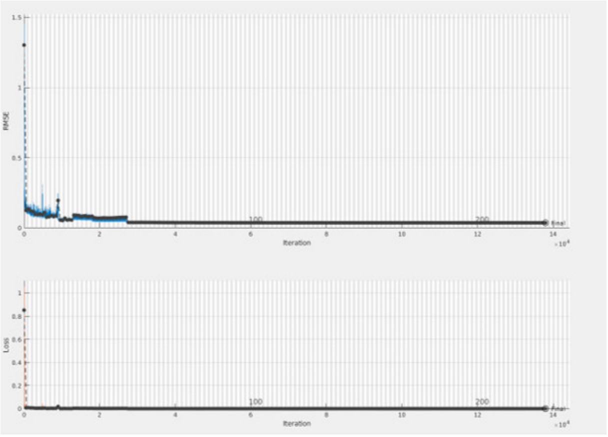

FIGURE 9. Training of the LSTM neural network of the CMFT. Figures show the time-variation of the root mean squared error and the loos function during the training.

VI. VALIDATION

In this section, the effectiveness of the proposed data-driven approach for fast-tracking of an unknown missile is assessed, and the performances of both the architectures are analyzed and compared by leveraging a novel data-set of 300 maneuver-based trajectories, of which 100 are ballistic, 100 are PU-PD and 100 are cross ones.

Validation results have disclosed the capabilities of both the proposed architecture in predicting the missile maneuvers without a thorough knowledge of the model and exploiting only raw and noisy measurement provided by the radar system, so confirming the robustness of LSTM approaches w.r.t. both uncertainty sources. See for example Figs. 10 and 11 where results confirm that, despite the measurement noise, both CMFT and DMFT are able to identify the dynamic of the target and correctly predict its position and velocity on a short time interval. Note that as expected, the maximum prediction error occurs at the beginning of the missile manoeuvre, where both CMFT and DMFT take just few samples to identify the maneuver. Moreover, generally the prediction error decreases

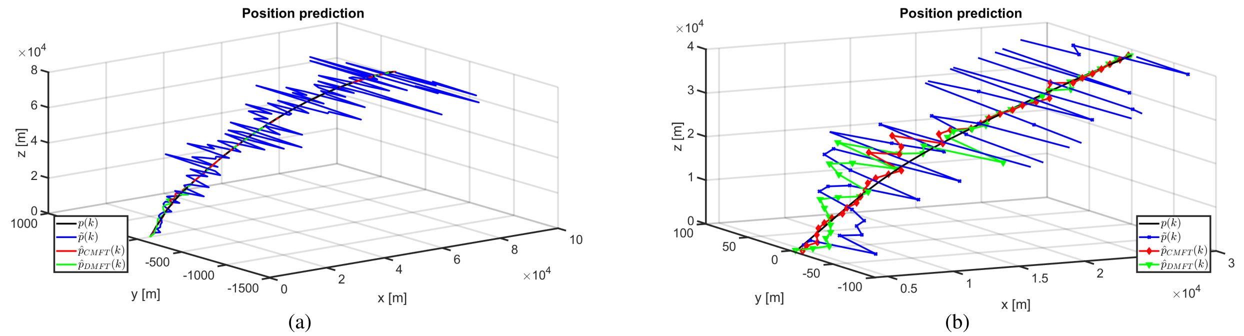

FIGURE 10. PUPD maneuver: Target 3D-Position Tracking. Black: raw target position data; Blue: noisy radar measurements; Red: CMFT prediction; Green: DMFT prediction. (a): entire PUPD maneuver. (b): focus on the last 40 seconds of the predicted trajectories.

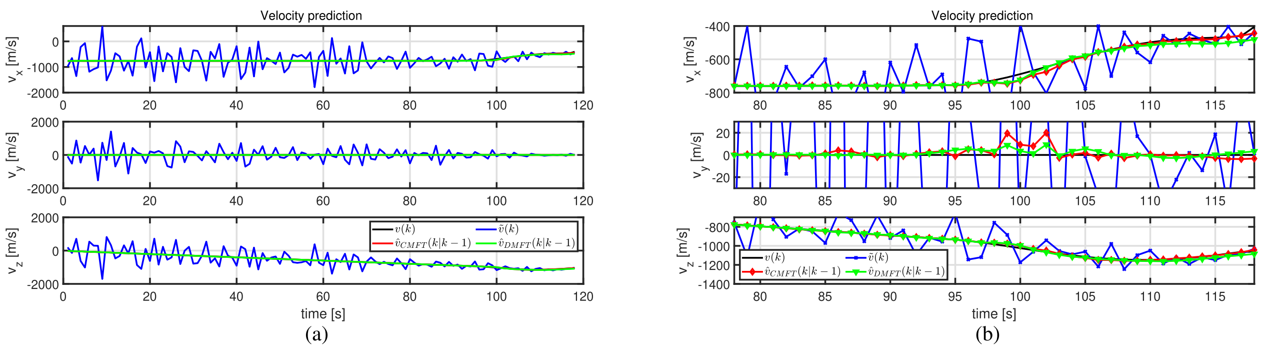

FIGURE 11. PUPD maneuver: Target velocity prediction. Black: raw target velocity data; Blue: noisy radar measurements; Red: CMFT prediction; Green: DMFT prediction. (a): entire PUPD maneuver. (b): focus on the last 40 seconds of the predicted velocities.

after the first time samples, as long as the manoeuvre does not change again.

A more in-depth characterization of the performance of the two proposed systems have been performed by processing all trajectory in the data-set and by analyzing the related results, choosing the Euclidean norm of the error vectors as performance index for assessing the tracking accuracy. In particular, the mean values of the errors for each validation trajectory were evaluated, and then the distribution of the errors for all the samples of the 300 trajectories has been analyzed. The results of this comparison analysis, depicted in Fig. 12, can summarized as follows.

- Compared to the DMFT, the CMFT discloses a better target position prediction capability for 282 of the 300 trajectories composing the test data-set (see Fig. 12a). More notably, the mean value of the prediction error of the CFMT is less than 400 [m] for 262 of the 300 trajectories composing the test data-set. Furthermore, the distributions of the prediction errors of DMFT is characterized by a greater number of samples in which the position error is greater than 100 [m], as shown in Fig. 12c.
- Focusing on the target velocity prediction, again the CMFT solution shows better performance than the DMFT one, with a smaller mean error for 286 of the 300 trajectories composing the test data-set (see Fig. 12b), that anyway is always below than 40 [m/s] for all the trajectories. Also in this respect, the distributions of the velocity target prediction errors show that the DMFT is characterized by a greater number of samples in which the velocity prediction error is greater than 100 [m/s], despite the latter is characterized by a greater number of samples in which the prediction errors is less than 25 [m/s], see Fig. 12d.

It follows that, although both architectures are able to solve the fast missile tracking problem, the obtained results lead to prefer the Coupled architecture rather than the Decoupled one. See, for example, Fig. 13 for an additional illustration of the prediction capabilities of the CMFT in the case of a 3D cross maneuver in the validation data-set.

So, to further investigate the potential of data-driven methods, in the next section the CMFT is compared to a traditional model-based Interacting Multiple Model (IMM) via Kalman filters.

VII. COMPARISON WITH AN IMM BANK OF KALMAN FILTERS

As already mentioned, prediction systems of the missile motion are usually based on classical IMM techniques (e.g. see [38] and references therein). So in this section, we compare the prediction capability of the CMFT system with an IMM system based on an array of nonlinear Kalman filters set to work with the same sampling time of the radar system, i.e. 1.0 [s], and able to take as output signals the predictions at

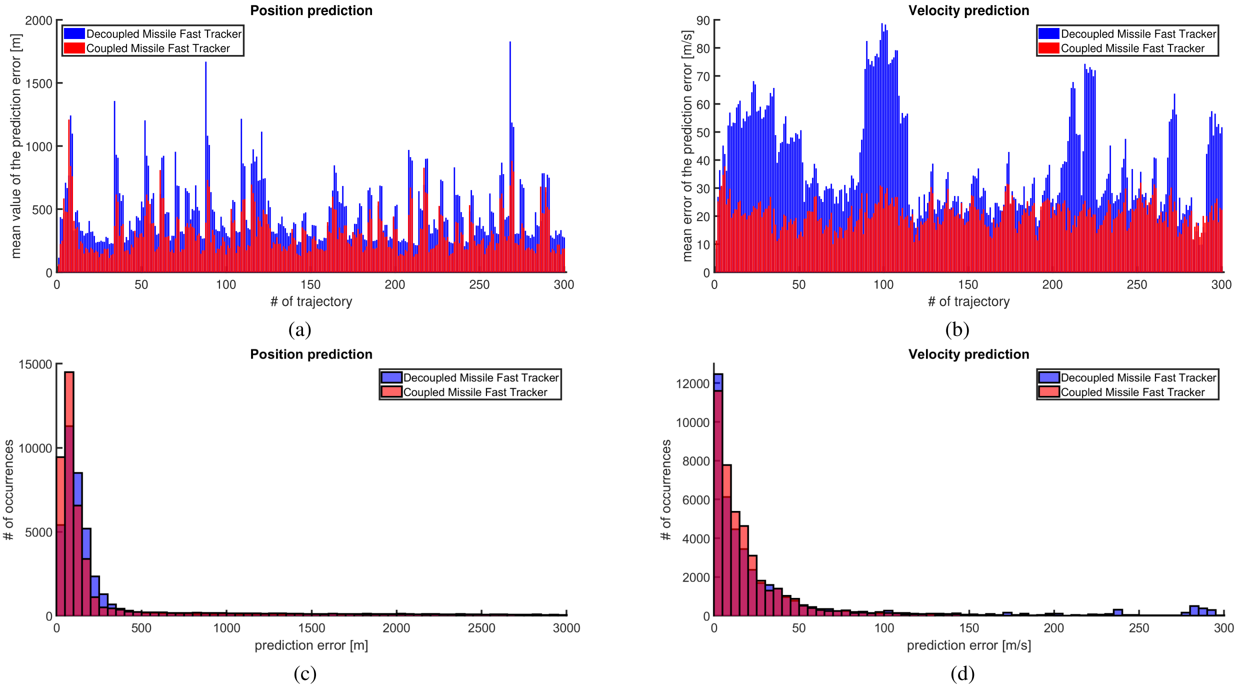

FIGURE 12. Comparison Analysis: CMFT (red) *v.s.* DMFT (blue). Top: mean values of errors for each trajectory in the test-set (left frame (a): position; right frame (b): velocity). Bottom: errors distributions over all the samples of the trajectories of the test data-set (left frame (c) : position; right frame (d): velocity).

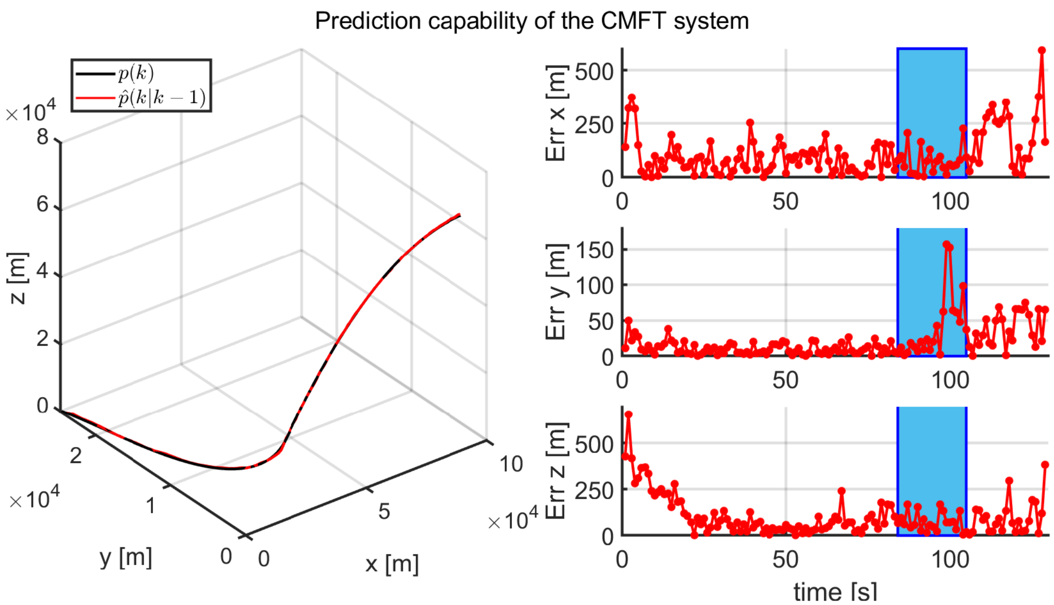

FIGURE 13. Cross maneuver: Target 3D-Position Tracking via CMFT. Left frame: actual trajectory performed by the target (blue line) *v.s.* the predicted one (red line). Right frame: time-history of the prediction errors. Note that the highlighted blue area refers to the set of samples characterized by an altitude ranging from 15 [*km*] to 35 [*km*], which have been averaged to compare the performance of the CMFT and IMM (see Section VII).

the next time instant of the missile position and velocity. Note that, the comparison analysis of proposed data-driven solutions has been performed with respect to an optimized IMM algorithm, suitably developed and validate by MBDA-Italy [35], in order to work in a realistic maneuvering scenarios. As a result, the performance of the proposed LSTM-based architecture and the efficient IMM method are evaluated in the same real-guidance environment, in order to make the comparison reliable and to assess the applicability of our solution in realistic situations.

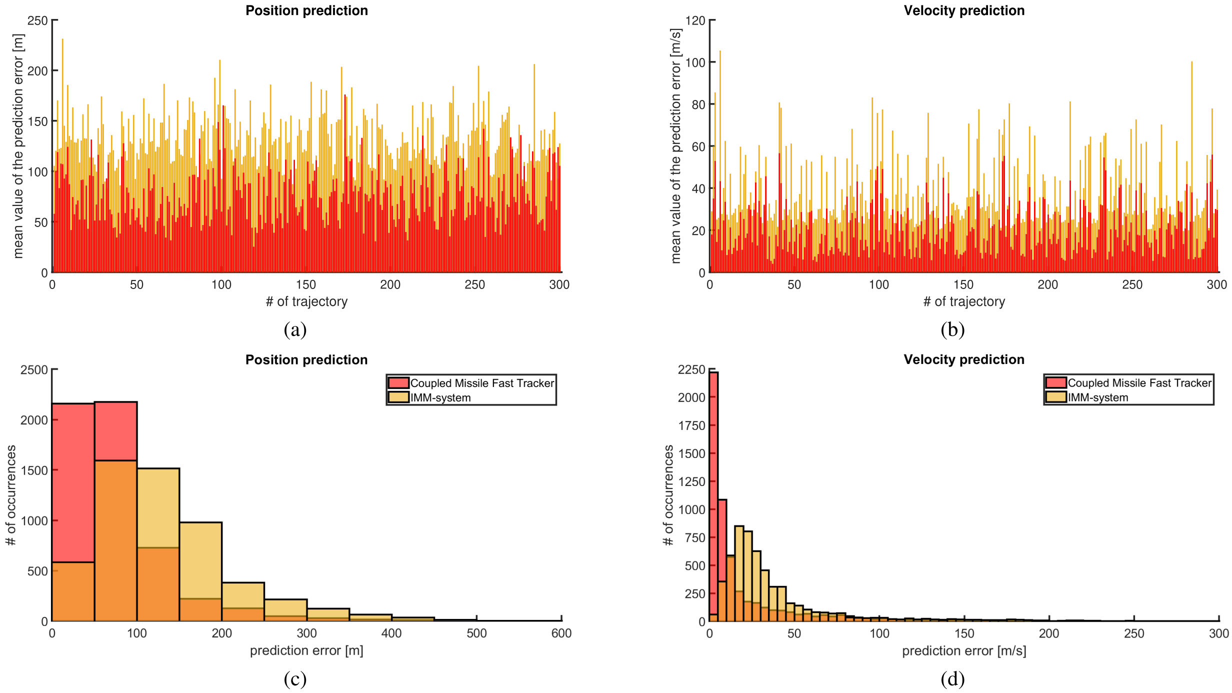

FIGURE 14. **Comparison Analysis: CMFT (red) *v.s.* IMM (light orange). Top: mean values of the errors for each trajectory of the test-set (left frame (a): position; right frame (b): velocity). Bottom: errors distributions over all the samples of the trajectories in the test data-set (left frame (c): position; right frame (d): velocity).**

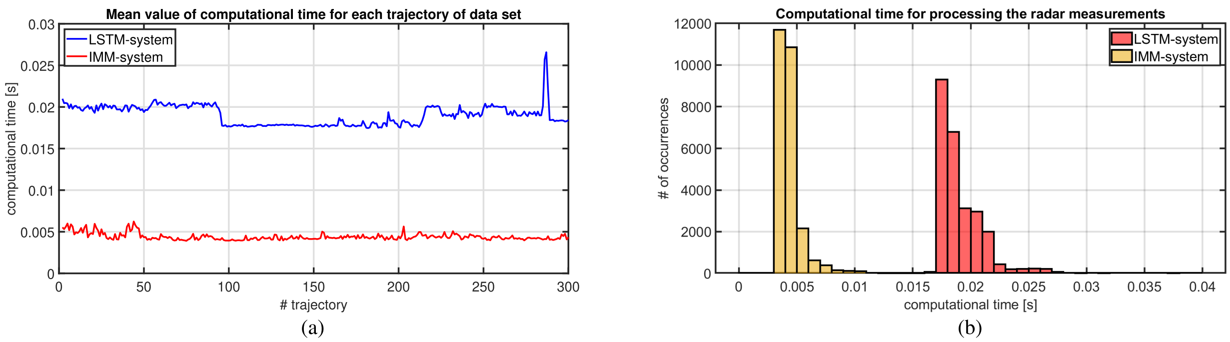

FIGURE 15. **Comparison of computational times. Left frame: avarage values of the computational times for each trajectory of the test-set. Right frame: times distribution over all the samples of the trajectories in the test data-set.**

The comparison between the two approaches in Fig. 14 has been carried out evaluating the errors in both position and velocity predictions. By considering the same test data-set of 300 trajectories exploited in the previous analysis, the tracking errors have been evaluated in the range of altitudes of possible interception of the target missile. In particular, for each trajectory of the data-set, we have averaged the errors related to the samples characterized by an altitude between 15 [*km*] and 35 [*km*], (see the blue area in Fig. 13). Results in Fig. 14a) and Fig. 14c) disclose that the CMFT has better target position prediction capability for 287 of the 300 trajectories composing the test data-set, while the distributions of the prediction errors of the IMM system are characterized by a greater number of samples in which the position error is greater than 100 [*m*]. Moreover, also in predicting the missile velocity the CMFT exhibits better capabilities w.r.t. the IMM one for 295 of the 300 trajectories composing the test data-set, as represented from the mean values of the prediction errors in Fig. 14b and from the distribution errors in Fig. 14d.

The very good performance of the CMFT are also confirmed by the overall comparison results summarized in Table 5. Finally, for comparison purposes, in Figure 15 we report the computational time required by both systems to accomplish the prediction task. The results point out a comparable time, with the IMM solution that exhibits slight improvements, however at the expense of the prediction's quality. Thus, the LSTM-based structure can be considered agile, with the possibility to decrease the computational burden by reducing adequately the number of units and retraining

TABLE 5. Average performance over 300 test trajectories (ballistic, PUPD and cross).

Metric	IMM-system	CMFT-system
ABS Err Position X [m]	76.97	43.06
ABS Err Position Y [m]	48.89	10.95
ABS Err Position Z [m]	59.03	31.92
Position RMSE [m]	118.83	62.10
ABS Err Velocity X [m/s]	17.15	6.70
ABS Err Velocity Y [m/s]	7.42	5.12
ABS Err Velocity Z [m/s]	13.18	5.04
Velocity RMSE [m/s]	25.71	11.61

the network, obviously at the cost of slightly worse performance [13].

VIII. CONCLUSION

In this paper, the possibility of addressing the tracking of uncertain ballistic missile trajectories leveraging data-driven Long-Short Term Memory nets architectures has been investigated. Comparison results suggest that data-driven algorithms can be considered as an effective alternative to classical Kalman-based approaches, such as Interacting Multiple Models (IMMs) methods. The proposed solution allows overcoming the narrowness of IMM in handling only the measurement noise that affects the raw radar data, by exploiting the ability of the LSTM in learning long-term dependencies of temporal information without any model knowledge, so to take into account both the measurement noise and the missile motion uncertainties. The built architectures have been tested in non-trivial missile motion scenarios and compared with a classical model-based IMM algorithm to investigate the effectiveness of the proposed solution in predicting unknown 3D missile maneuvers. Future works could include the real implementation of the proposed LSTM architectures on dedicated FPGA-based accelerators, which allows also to exploit parallelism and therefore improves the execution time of the proposed data-driven approach.

REFERENCES

[1] X. Wang, T. Li, S. Sun, and J. Corchado, "A survey of recent advances in particle filters and remaining challenges for multitarget tracking," *Sensors*, vol. 17, no. 12, p. 2707, Nov. 2017.

[2] F. Song, Y. Li, W. Cheng, L. Dong, M. Li, and J. Li, "An improved Kalman filter based on long short-memory recurrent neural network for nonlinear radar target tracking," *Wireless Commun. Mobile Comput.*, vol. 2022, pp. 1–10, Aug. 2022.

[3] K. Yoo, J. Chun, and J. Shin, "Target tracking and classification for missile using interacting multiple model (IMM)," in *Proc. Int. Conf. Radar (RADAR)*, 2018, pp. 1–6.

[4] W. Youn and H. Myung, "Robust interacting multiple model with modeling uncertainties for maneuvering target tracking," *IEEE Access*, vol. 7, pp. 65427–65443, 2019.

[5] E. Mazor, A. Averbuch, Y. Bar-Shalom, and J. Dayan, "Interacting multiple model methods in target tracking: A survey," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 34, no. 1, pp. 103–123, Jan. 1998.

[6] H. P. Blom, "An efficient filter for abruptly changing systems," in *Proc. 23rd IEEE Conf. Decis. Control*, Dec. 1984, pp. 656–658.

[7] K. Shi, D. Cheng, X. Yuan, L. Liu, and L. Wu, "Interacting multiple model-based adaptive control system for stable steering of distributed driver electric vehicle under various road excitations," *ISA Trans.*, vol. 103, pp. 37–51, Aug. 2020.

[8] Y. Wang, D. Meng, R. Li, Y. Zhou, and X. Zhang, "Multi-fault diagnosis of interacting multiple model batteries based on low inertia noise reduction," *IEEE Access*, vol. 9, pp. 18465–18480, 2021.

[9] H. A. P. Blom and E. A. Bloem, "Exact Bayesian and particle filtering of stochastic hybrid systems," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 43, no. 1, pp. 55–70, Jan. 2007.

[10] L. Gong and M. Yu, "A new interacting multiple model particle filter based ballistic missile tracking method," in *Proc. 3rd Int. Conf. Frontiers Signal Process. (ICFSP)*, Sep. 2017, pp. 81–85.

[11] H. M. T. Menegaz and S. Battistini, "Switching multiple model filter for boost-phase missile tracking," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 54, no. 5, pp. 2547–2553, Oct. 2018.

[12] M. Yu, L. Gong, H. Oh, W.-H. Chen, and J. Chambers, "Multiple model ballistic missile tracking with state-dependent transitions and Gaussian particle filtering," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 54, no. 3, pp. 1066–1081, Jun. 2018.

[13] C. Gao, J. Yan, S. Zhou, P. K. Varshney, and H. Liu, "Long short-term memory-based deep recurrent neural networks for target tracking," *Inf. Sci.*, vol. 502, pp. 279–296, Oct. 2019.

[14] L. Deng, D. Li, and R. Li, "Improved IMM algorithm based on RNNs," *J. Phys., Conf.*, vol. 1518, no. 1, Apr. 2020, Art. no. 012055.

[15] D.-T. Hoang and H.-J. Kang, "A survey on deep learning based bearing fault diagnosis," *Neurocomputing*, vol. 335, pp. 327–335, Mar. 2019.

[16] S. Jia, S. Jiang, Z. Lin, N. Li, M. Xu, and S. Yu, "A survey: Deep learning for hyperspectral image classification with few labeled samples," *Neurocomputing*, vol. 448, pp. 179–204, Aug. 2021.

[17] S. Wu, Y. Jiang, H. Luo, and S. Yin, "Remaining useful life prediction for ion etching machine cooling system using deep recurrent neural network-based approaches," *Control Eng. Pract.*, vol. 109, Apr. 2021, Art. no. 104748.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[19] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 21–37.

[20] Y. Chen, D. Zhang, and W.-A. Zhang, "MSWR-LRCN: A new deep learning approach to remaining useful life estimation of bearings," *Control Eng. Pract.*, vol. 118, Jan. 2022, Art. no. 104969.

[21] E. Marchi, G. Ferroni, F. Eyben, L. Gabrielli, S. Squartini, and B. Schuller, "Multi-resolution linear prediction based features for audio onset detection with bidirectional LSTM neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2014, pp. 2164–2168.

[22] T. Hu, K. Li, H. Ma, H. Sun, and K. Liu, "Quantile forecast of renewable energy generation based on indicator gradient descent and deep residual BiLSTM," *Control Eng. Pract.*, vol. 114, Sep. 2021, Art. no. 104863.

[23] Y. T. Wu, M. Yuan, S. Dong, L. Li, and Y. Liu, "Remaining useful life estimation of engineered systems using vanilla LSTM neural networks," *Neurocomputing*, vol. 275, pp. 167–179, Jan. 2018.

[24] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 1310–1318.

[25] X. Li, F. Duan, P. Loukopoulos, I. Bennett, and D. Mba, "Canonical variable analysis and long short-term memory for fault diagnosis and performance estimation of a centrifugal compressor," *Control Eng. Pract.*, vol. 72, pp. 177–191, Mar. 2018.

[26] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[27] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," *Neural Netw.*, vol. 18, no. 5, pp. 602–610, 2005.

[28] J. Wu, K. Hu, Y. Cheng, H. Zhu, X. Shao, and Y. Wang, "Data-driven remaining useful life prediction via multiple sensor signals and deep long short-term memory neural network," *ISA Trans.*, vol. 97, pp. 241–250, Feb. 2020.

[29] M. Sayah, D. Guebli, Z. Al Masry, and N. Zerhouni, "Robustness testing framework for RUL prediction deep LSTM networks," *ISA Trans.*, vol. 113, pp. 28–38, Jul. 2021.

[30] J. Yang, Y. Guo, and W. Zhao, "Long short-term memory neural network based fault detection and isolation for electro-mechanical actuators," *Neurocomputing*, vol. 360, pp. 85–96, Sep. 2019.

[31] Q. Zhang, F. Li, F. Long, and Q. Ling, "Vehicle emission forecasting based on wavelet transform and long short-term memory network," *IEEE Access*, vol. 6, pp. 56984–56994, 2018.
[32] J. Zhang, Y. Zhu, X. Zhang, M. Ye, and J. Yan, "Developing a long short-term memory (LSTM) based model for predicting water table depth in agricultural areas," *J. Hydrol.*, vol. 561, pp. 918–929, Jun. 2018.
[33] M. Alazab, S. Khan, S. S. R. Krishnan, Q.-V. Pham, M. P. K. Reddy, and T. R. Gadekallu, "A multidirectional LSTM model for predicting the stability of a smart grid," *IEEE Access*, vol. 8, pp. 85454–85463, 2020.
[34] J. Liu, Z. Wang, and M. Xu, "DeepMTT: A deep learning maneuvering target-tracking algorithm based on bidirectional LSTM network," *Inf. Fusion*, vol. 53, pp. 289–304, Jan. 2020.
[35] A. Candeli, G. De Tommasi, and S. Santini, "Design di algoritmi di guida airframe basati sull'intelligenza artificiale," Dept. Elect. Eng., Inf. Technol. Univ. Naples Federico II, Italy, Tech. Rep., MEp-IT/RM/LL/2019/530, 2019.
[36] W. J. Farrell, "Interacting multiple model filter for tactical ballistic missile tracking," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 44, no. 2, pp. 418–426, Apr. 2008.
[37] S.-H. Kim and H.-L. Choi, "An IMM-based method for reentry-phase tracking of unknown ballistic missiles," in *Proc. 13th Int. Conf. Control, Autom. Syst. (ICCAS)*, Oct. 2013, pp. 751–754.
[38] I.-C. Moon, K. Song, S.-H. Kim, and H.-L. Choi, "State prediction of high-speed ballistic vehicles with Gaussian process," *Int. J. Control, Autom. Syst.*, vol. 16, no. 3, pp. 1282–1292, Jun. 2018.
[39] H. Sak, A. W. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *Proc. INTERSPEECH*, 2014, pp. 338–342.
[40] G. S. Chadha, A. Panambilly, A. Schwung, and S. X. Ding, "Bidirectional deep recurrent neural networks for process fault classification," *ISA Trans.*, vol. 106, pp. 330–342, Nov. 2020.
[41] J. Wang, J. Zhang, and X. Wang, "Bilateral LSTM: A two-dimensional long short-term memory model with multiply memory units for short-term cycle time forecasting in re-entrant manufacturing systems," *IEEE Trans. Ind. Informat.*, vol. 14, no. 2, pp. 748–758, Feb. 2018.
[42] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. Cambridge, MA, USA: MIT Press, 2012.
[43] Z. Tian, K. Yang, M. Danino, Y. Bar-Shalom, and B. Milgrom, "Ballistic object trajectory and impact point estimation in the reentry phase from a moving passive sensor," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 58, no. 5, pp. 4540–4550, Oct. 2022.

DARIO GIUSEPPE LUI was born in Avellino, Italy, in March 1990. He received the master's degree (Hons.) in electronics engineering for automation and telecommunications (specialization in automation) and the Ph.D. degree in information technology for engineering from the University of Sannio, Benevento, Italy, in 2015 and 2020, respectively. He is currently a Research Fellow with the University of Naples Federico II. His work focuses on the distributed control of multiagent systems in the presence of communication impairments, with application to the automotive field and reinforcement learning.

GAETANO TARTAGLIONE (Member, IEEE) was born in Caserta, Italy, in March 1989. He received the master's degree (Hons.) in aerospace engineering from the Second University of Naples, Italy, in 2014, and the Ph.D. degree in information engineering from the University of Naples Parthenope, Italy, in 2018. Currently, he is a Researcher with the Engineering Department, University of Naples Parthenope. His research interests include control of nuclear fusion devices, multiagent cooperative control, and finite-time stability and stabilization for the class of linear time-varying stochastic systems.

FRANCESCO CONTI was born in Pontecorvo, Italy. He received bachelor's and master's degrees in communication engineering from the University of Roma La Sapienza, in 2017 and 2020, respectively. He held a Research Fellowship in signal and image processing with the Department of Computing Sciences, Tampere University, Finland. He held a Research Fellowship with the Department of Information Engineering, Electronics and Telecommunications (DIET), Sapienza University of Rome, Italy, in 2020. He was an Image Processing Engineer with the Guidance, Control and Navigation Department, MBDA, Italy, in 2021. In 2022, he joined the Data and Artificial Intelligence Center of Competence, Cy4gate SPA.

GIANMARIA DE TOMMASI (Senior Member, IEEE) was born in Milan, Italy, in 1975. He received the Laurea degree (summa cum laude) in electronic engineering and the Doctorate degree (Research) in computer and automatic engineering from the University of Naples Federico II, in 2001 and 2005, respectively. He is currently a Full Professor in automatic control with the Department of Electrical Engineering and Information Technology, University of Naples Federico II. Since 2002, he has been a Visiting Researcher with the Joint European Torus, U.K., ITER Organization, France, EAST Tokamak, China, and International Fusion Research Centre, Japan, where he has participated in various projects connected to the plasma magnetic control systems. His current research interests include control of nuclear fusion devices, fault detection, and identification of discrete event systems modeled with Petri nets and stability of hybrid systems. He has published more than 200 journal and conference papers on these topics and coauthored two monographs titled *Finite-Time Stability and Control* and *Finite-Time Stability: An Input-Output Approach*. He is a member of the IEEE Control System Society Conference Editorial Board. He is also the Guest Editor of the *Fusion Engineering and Design* Special Issue titled "Design and implementation of real-time systems for magnetic confined fusion devices," and an Editor of the 14th IFAC Workshop on Discrete Event Systems (WODES 2018).

STEFANIA SANTINI (Member, IEEE) received the M.Sc. degree in electronic engineering and the Ph.D. degree in automatic control from the University of Naples Federico II, Naples, Italy, in 1996 and 1999, respectively. She is currently an Associate Professor in automatic control. She is involved in many projects with industry, including small- and medium-sized enterprises operating in the automotive field. Her research interests include the area of the analysis and control of nonlinear systems with applications to automotive engineering, transportation technologies, computational biology, and energy systems.

Open Access funding provided by 'Università degli Studi di Napoli "Federico II"' within the CRUI CARE Agreement