IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 19, 2019, accepted January 2, 2020, date of publication January 6, 2020, date of current version January 16, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2964285

A Novel Recurrent Convolutional Neural Network-Based Estimation Method for Switching Guidance Law

HUIBING SHAO, YEPENG HAN, CHANGZHU WEI, AND RUIMING WANG
School of Astronautics, Harbin Institute of Technology, Harbin 150001, China

Corresponding author: Changzhu Wei (weichangzhu@hit.edu.cn)

This work was supported in part by the National Nature Science Fund of China under Grant 61403100, and in part by the Fundamental Research Funds for the Central Universities under Grant HIT.NSRIF.2015037.

ABSTRACT This paper presents a recurrent convolutional neural network-based estimation method of guidance parameters of the pursuer under augmented proportional navigation (APN) guidance law with a time-varying switching navigation ratio. For the highly maneuvering pursuit-evasion process, realistic factors in the guidance law estimation are considered, such as the pursuer's estimation error and delay of the evader's acceleration in the APN guidance law. In view of the enormous measurement data, time dependency, transient change and unknown factors' disturbance in switching guidance law estimation, a novel neural network structure is built. 1-D CNN layer is used to extract features from enormous data obtained by the multiple previous measurements. The features extracted are processed by the recurrent cell to exploit the time dependency and eliminate the error caused by unknown factors. The result of ablation test shows the proposed RCNN's improved performance over single CNN or RNN. Compared to the multiple model guidance law estimation method, the proposed method can simplify the design of guidance law estimation system and reduce calculation load. The estimation result for switching guidance law shows the proposed method has higher accuracy and faster convergence rate than traditional interactive multiple model methods.

INDEX TERMS Guidance law estimation, pursuit-evasion process, recurrent convolutional neural network, switching navigation ratio, time dependency.

I. INTRODUCTION

With rapid development of interceptors, the penetration challenge in reentry warhead has become increasingly serious. The traditional programmed penetration does not consider the characteristics of interceptor, so it is difficult to penetrate successfully in the face of high performance interceptors. Since the publication of Imado's research on fighter evasive law against proportional navigation guidance law [1], building optimal evasion maneuver based on the interceptor's characters to realize successful penetration has become a hot research field. Jung *et al.* applied fuzzy rules to obtain a sub-optimal evasion law for aircraft [2]. Shima proposed a general framework for optimal evasive maneuvers against pursuers with linear guidance law. Matched optimal evasion maneuvers were derived based on the known guidance law [3]. As the precondition to building efficient optimal evasion maneuvers, accurate guidance law estimation of the interceptor is of great importance [4]. At present, multiple model guidance law estimation method is a commonly used approach.

Scholars have proposed several multiple model guidance law estimation methods and gained substantial results upon specific assumption. Shaferman and Shima used a static multiple-model adaptive estimator(MMAE) to estimate guidance law [4]. However, the parameter of each guidance law is assumed to be fixed and the transition between different guidance laws is not considered. Fondo and Shima developed a MMAE method with reduced-order approach [5], in which the guidance parameters were treated as unknown system parameters to be estimated. In the mentioned above works, guidance law is assumed to be fixed.

In practical pursuit-evasion process, the interceptors may switch the guidance law parameter [6]. MMAE methods may degenerate in this situation due to the static structure. Therefore, scholars put forward interactive multiple model

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

filter (IMM) method to fit the switching situation. IMM has a set of models representing different guidance laws of the pursuer. Each model corresponds to a filter. They run in parallel, estimate the motion of pursuer and interact with each other. In each filter cycle, the posterior probability of each model is updated. Yun and Ryoo estimated the guidance law with unknown navigation ratio by IMM method [7]. Xinguang *et al.* applied IMM to estimate proportional navigation guidance law with switching navigation ratio [8]. However, when the ratio switches rapidly and leaves a very narrow time window for IMM to converge, IMM degenerates and leads to a large estimation error.

Apart from slow convergence rate that weakens optimal maneuver effects, the work mentioned above ignore the dynamic characteristics such as the estimation delay of the evader's acceleration when attacking highly maneuvering evader. Onboard computation of multiple model filters leads to a large computation load for the processor of evader. In addition, the estimation performance depends highly on transient probability matrix and initialization [9].

The development of neural network provides new insight to tackle the guidance law estimation problem. Neural network can deal with complex nonlinear problems and performs well in analysis of time series and images [10]. Convolutional neural network can process a massive amount of data input parallelly, while recurrent neural network can handle the problems with time dependency. Scholars also explored the application of neural network in the field of aerospace guidance and control. Li *et al.* built BP neural network to realize online predictive guidance for high lifting vehicle and formed a data-driven online entry guidance framework [11]. Zhao *et al.* applied RBF neural network as disturbance observer for integrated guidance and control system [12]. This paper aims to build end-to-end guidance law estimation method based on neural network.

According to deep learning framework problems with complex mapping relationship can be solved by building appropriate network structure and training with sufficient data [13]. For guidance law estimation problem, pursuit-evasion simulations can provide enough data for training. The problem involves numerous data with time dependency and transient characteristic and is subjected to disturbance of unknown factors, so corresponding neural network structure shall be designed to realize accurate online estimation.

In this paper, a modified recurrent convolutional neural network method is presented for guidance law estimation. The design of guidance law estimation system can be greatly simplified. The proposed method can estimate the switching guidance law with measurement noise and uncertainty in the realistic environment.

The work's contributions are as follows:

1) Neural network is adopted for the first time to build an end-to-end method for guidance law estimation. Compared with multiple model filter estimation method, the end-to-end method has a simplified design of estimation system. The proposed method can greatly reduce calculation load and is more promising for online application.
2) In the proposed network, a novel combination of the feature extraction, pre-estimation and recurrent structure is adopted to process the past estimation result, so the measurement-associated time dependent characteristics can be exploited and the convergence time can be greatly reduced when facing switching navigation ratio compared to the traditional multiple model filter method.
3) The disturbance from the interceptor's unknown characters such as the time delay and system's nonlinearity is considered when building models and samples, so the proposed method can perform better in real environment.

The rest of the paper is organized as follows: Section 2 describes the guidance law estimation during the pursuit-evasion process by considering navigation ratio switching, the error and time delay in estimation of the evader's acceleration in the APN guidance law; Section 3 analyzes the adaptability of existing neural network methods to time series problem and the specific guidance law estimation problem; Section 4 presents the modified recurrent convolutional neural network-based estimation method of guidance law; Section 5 presents ablation test of proposed method in comparison to single CNN or RNN structure and carries out contrast simulation of proposed method in comparison to traditional IMM filter methods so as to verify the efficiency and accuracy of the proposed method; Section 6 gives the conclusions of this paper.

II. PROBLEM FORMULATION

In this paper, the interceptor is denoted as the pursuer, and the warhead is denoted as the evader. The initial line-of-sight (LOS) coordinate with x-axis pointing from evader to pursuer is taken as the scene inertial coordinate, as shown in Fig. 1. σ_e, σ_a are the LOS angles.

The state equation of the pursuit-evasion process is represented as:

$$\begin{cases} \dot{r}_x = v_x \\ \dot{r}_y = v_y \\ \dot{r}_z = v_z \\ \dot{v}_x = a_{xe} - a_{xm} \\ \dot{v}_y = a_{ye} - a_{ym} \\ \dot{v}_z = a_{ze} - a_{zm} \\ \dot{a}_{xm} = (u_{xm} - a_{xm}) \big/ \tau \\ \dot{a}_{ym} = \left(u_{ym} - a_{ym}\right) \big/ \tau \\ \dot{a}_{zm} = (u_{zm} - a_{zm}) \big/ \tau \end{cases} \tag{1}$$

where r_x, r_y, r_z are the relative position components in the scene inertial coordinate from the evader to the pursuer; v_x, v_y, v_z are the relative velocity components in the scene inertial coordinate from the evader to the pursuer; a_{xe}, a_{ye}, a_{ze} are the evader acceleration components in the scene

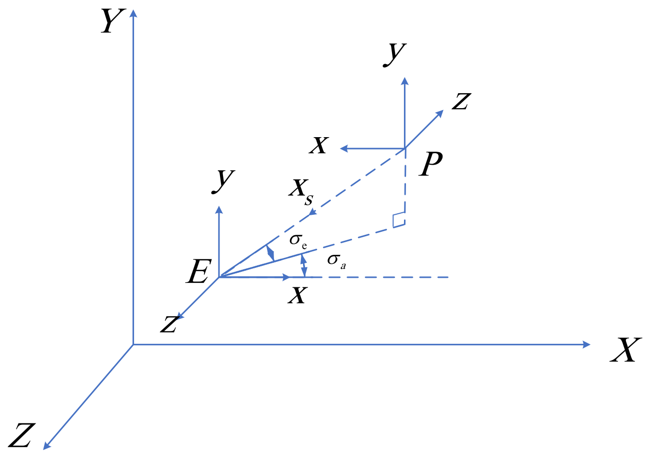

FIGURE 1. Engagement between the pursuer and evader.

inertial coordinate; u_{xm}, u_{ym}, u_{zm} are the pursuer's guidance command in the scene inertial coordinate; a_{xm}, a_{ym}, a_{zm} are the pursuer acceleration components in the scene inertial coordinate; τ denotes the time constant of the pursuer and is usually set to 0.1 s.

When attacking highly maneuvering evader, augmented proportional navigation (APN) guidance law shows good performance, while traditional proportional navigation guidance law cannot obtain satisfactory miss distance. APN is easy to realize and has been widely used in interceptor [14], [15]. The pursuer is assumed to be guided under an APN guidance law with an unknown navigation ratio N. This paper focuses on the estimation of the navigation ratio N.

Based on APN guidance law, the pursuer's acceleration command in the LOS coordinate is [16]:

$$\begin{cases} u_{yL} = Nv_r\dot{\sigma}_e + \dfrac{N}{2}\hat{a}_{TyL} \\ u_{zL} = Nv_r\dot{\sigma}_a + \dfrac{N}{2}\hat{a}_{TzL} \end{cases} \tag{2}$$

where N is navigation ratio, $\hat{a}_{TyL}, \hat{a}_{TzL}$ is the estimated normal and lateral acceleration of evader in the LOS coordinate, respectively. v_r is the approaching velocity, $\dot{\sigma}_e, \dot{\sigma}_a$ are the LOS angular rates, which are given by the following equations:

$$\begin{aligned} v_r &= \frac{v_x r_x + v_y r_y + v_z r_z}{\sqrt{r_x^2 + r_y^2 + r_z^2}} \\ R &= \sqrt{r_x^2 + r_y^2 + r_z^2} \\ \dot{\sigma}_e &= \frac{\left(r_x^2 + r_z^2\right) v_y - r_x r_y v_x - r_z r_y v_z}{R^2\sqrt{r_x^2 + r_z^2}} \\ \dot{\sigma}_a &= \frac{v_x r_z - v_z r_x}{r_x^2 + r_z^2} \end{aligned}$$

The pursuer cannot get the evader's acceleration directly, so the $\hat{a}_{TyL}, \hat{a}_{TzL}$ in APN guidance law need to be estimated, which will inevitably cause time delay [17]. According to the assumption in [17], the time delay is represented as an unknown constant τ_r, so $\hat{a}_{TyL}, \hat{a}_{TzL}$ in (2) are the delayed evader accelerations. In addition, the evader's uncertain maneuver with high amplitude may lead to great error in estimation of the evader's acceleration, thus influencing the pursuer's command [18]. These dynamic characteristics in turn bring uncertainty to guidance law estimation. The $\hat{a}_{TyL}, \hat{a}_{TzL}$ in (2) are given as:

$$\begin{aligned} [\hat{a}_{TxL}, \hat{a}_{TyL}, \hat{a}_{TzL}]^T &= \boldsymbol{C}_{ps}[\hat{a}_{Tx}, \hat{a}_{Ty}, \hat{a}_{Tz}]^T \\ \hat{a}_{Tx} &= a_{Tx} + w_{aTx} \\ \hat{a}_{Ty} &= a_{Ty} + w_{aTy} \\ \hat{a}_{Tz} &= a_{Tz} + w_{aTz} \\ w_{aTx} &\sim \mathcal{N}\left(0, r_{\mathrm{aTx}}^2\right) \\ w_{aTy} &\sim \mathcal{N}\left(0, r_{\mathrm{aTy}}^2\right) \\ w_{aTz} &\sim \mathcal{N}\left(0, r_{\mathrm{aTz}}^2\right) \end{aligned} \tag{3}$$

where a_{Tx}, a_{Ty}, a_{Tz} is the acceleration components of evader in the scene inertial coordinate, respectively. $\boldsymbol{C}_{ps}$ is a transform matrix from the scene inertial coordinate to LOS coordinate, and it is expressed as

$$\begin{aligned} \boldsymbol{C}_{ps} &= \begin{bmatrix} \cos\sigma_e & \sin\sigma_e & 0 \\ -\sin\sigma_e & \cos\sigma_e & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\sigma_a & 0 & -\sin\sigma_a \\ 0 & 1 & 0 \\ \sin\sigma_a & 0 & \cos\sigma_a \end{bmatrix} \\ \sigma_e &= \arctan\left(\frac{r_y}{\sqrt{r_x^2 + r_z^2}}\right) \\ \sigma_a &= \arctan\left(-\frac{r_z}{r_x}\right) \end{aligned}$$

During the terminal guidance phase, the guidance system of the pursuer may change the navigation ratio in order to obtain better performance [19]. In large relative range, navigation ratio is usually set as a small value to avoid the guidance command saturation caused by large thermal noise of pursuer seeker on the evader. Alternatively in smaller relative range, the level of thermal noise degrades and larger navigation ratio can improve the convergence of LOS angular rate [20]. In the convergence process after the guidance law switching, the estimation deviation will have a great influence on the optimal evasion maneuvers. The transient guidance parameter puts forward a higher requirement on the convergence rate of the estimation method.

The evader is assumed to be equipped with a radar seeker. It is assumed that the relative distance and LOS angle between the pursuer and the evader can be measured by radar seeker. The measurement equation of evader is given as:

$$\begin{aligned} \hat{z} &= [\hat{R}, \hat{\sigma}_e, \hat{\sigma}_a]^T \\ \hat{R} &= \sqrt{r_x^2 + r_y^2 + r_z^2} + w_R \\ \hat{\sigma}_e &= \arctan\left(\frac{r_y}{\sqrt{r_x^2 + r_z^2}}\right) + w_{\sigma e} \end{aligned}$$

$$\begin{aligned}\hat{\sigma}_a &= \arctan\left(-\frac{r_z}{r_x}\right) + w_{\sigma a}\\ w_R &\sim \mathcal{N}\left(0, r_R^2\right)\\ w_{\sigma e} &\sim \mathcal{N}\left(0, r_{\sigma e}^2\right)\\ w_{\sigma a} &\sim \mathcal{N}\left(0, r_{\sigma a}^2\right)\end{aligned} \tag{4}$$

where $w_R, w_{\sigma e}, w_{\sigma a}$ are zero-mean independent white Gaussian noise.

III. TYPICAL NEURAL NETWORK CHARACTER FOR TIME SERIES ANALYSIS

According to the above formulas, the guidance law estimation is basically a time series analysis problem involving strong time dependence, numerous data and unknown factors. Traditional multiple model estimation methods usually require accurate detailed system model and tedious derivation. In contrast, neural network method can form end-to-end estimation model with data driven training.

To deal with such kind of estimation problem with neural network, two typical neural networks, i.e. CNN and RNN are analyzed separately, which provides basis for further design of combined structure.

A. RECURRENT NEURAL NETWORK

Recurrent neural network is a normally used tool for time series analysis [21]. Traditional recurrent neural network may cause gradient vanishing problem for long time series. Long short-term memory(LSTM) recurrent neural network incorporates gating function to solve the vanishing gradient problem arisen from traditional recurrent neural network [22]. However, with the time goes on, the initial input has little influence on the output of the network, so the numerous measurement data cannot be fully exploited.

Another disadvantage is that recurrent neural network just processes the input one by one, so it takes longer calculating time when processing more input data. For the guidance law estimation problem, hundreds of data from the past several measurements are sent into the network, and recurrent neural network cannot meet real-time requirements.

B. CONVOLUTIONAL NEURAL NETWORK

To exploit parallel computation and prevent gradient vanishing, an alternative choice for time series analysis is to include enough data input and send them into the neural network together. As an end-to-end method, convolutional neural network can operate estimation in a succinct form [23]. To deal with high dimensional data, convolutional neural network shares weights for neurons in the same group [24]. This kind of neural layer can cover entire input series, and learn high-level abstract representation of numerous time series data. However, window-based neural network cannot adjust input number to fit the strike and jump. In addition, there will be several estimations during the flight, and convolutional neural network cannot exploit the time dependency among past estimation results.

C. RECURRENT CONVOLUTIONAL NEURAL NETWORK

In recent years, scholars have put forward the idea of combining recurrent neural network and convolutional neural network to treat different levels of information in process of pictures, texts and time series.

In Kalchbrenner's model for discourse composition, firstly a hierarchical convolutional neural network is adopted to treat sentence-level information, then the features of the extracted sentence are sent into recurrent neural network to capture the discourse's central properties [25].

Pinheiro *et al.* constructed a scene labeling system with recurrent convolutional neural network [26]. The network consists of a sequential series of convolutional networks with the same parameters. In each instance, an RGB image and the previous predictions are both sent to the layer. During training, the recurrent structure can effectively smooth the predicted labels. The proposed network can obtain contextual meaning from pictures.

Lai *et al.* applied a bi-directional recurrent structure to capture the contextual information when learning word representations. A max-pooling layer was further employed to judge key features and capture the text's key component [27].

Donahue *et al.* proposed a long-term recurrent convolutional neural network for visual recognition and description [28]. In the model, each frame is sent into a visual ConvNet to encode a deepstate vector, and then the vector is sent into an LSTM to be decoded into a natural language string. The model performs well in activity recognition, image description and video description.

Wu *et al.* applied recurrent convolutional neural network for abnormal condition diagnosis of fused magnesium furnace [29]. Convolutional structure is used to extract spatial feature of irregular highlighted regions on the furnace shell. Recurrent structure is used to exploit temporal feature of the irregular region, i.e. the brightness and area of such irregular region increase over time.

Karim *et al.* proposed a long short term memory fully convolutional network for time series classification [30]. The time series data are processed into univariate time series by convolutional block with multiple time steps. At the same time, the data is processed by dimension shuffle and sent into the LSTM block as a single-step multivariate time series. The outputs of two blocks are concatenated and passed onto a softmax layer to obtain the result. The model achieves better performance than several state-of-the-art algorithms on many UCR Benchmark datasets. Moreover, a series of ablation test were conducted to understand the reason of the improvement [31], and the method was further extended for multivariate time series classification [32].

From the works listed above, it can be seen the recurrent convolutional neural network has various structures to adapt to different application fields. In the next Section, a novel recurrent convolutional structure will be built for guidance law estimation.

IV. RCNN ESTIMATION NETWORK

A. PREPROCESSING OF NETWORK INPUT

Due to different initial states, navigation ratio and the time delay in estimation of evader's acceleration, multiple simulations of pursuit-evasion process are operated to get enough data for training. The obtained data is 3D tensor, each dimension represents sample number, time and data attribute (6 measurement attributes: $[x_t, y_t, z_t, \hat{R}, \hat{\sigma}_e, \hat{\sigma}_a]$). In practice, each estimation is operated based on the data measured at past several measurement points, so the simulation data is transformed into 4D tensor X_{sija} as inputs of network, where the subscripts s, i, j, a represent the sample number, the estimation number, the measurement number and the data attribute, respectively. Considering difference in magnitude and dimension of measurement, the input data is normalized by min-max method:

$$X_{ij}^{s,a} = \frac{X_{ij}^{s,a} - \min\left(X_{ij}^{s,a}\right)}{\max\left(X_{ij}^{s,a}\right) - \min\left(X_{ij}^{s,a}\right)} \quad \forall a = 1, \cdots, A \quad (5)$$

where A is the attribute number of measurement.

Data preprocessing is done in a user defined layer in Keras.

B. NETWORK STRUCTURE

The flight time is different from sample to sample, so the time length is variable. Zero-padding algorithm is performed to fit the batch parallel training. A user defined layer in Keras is built to guide the zero-padding position in convolutional part's product. LSTM cells applies masking to eliminate the influence of padded data.

Considering the massive amount of measurement data, the convolutional structure of estimation network is adopted to process the low-level data in each estimation while recurrent structure is used to exploit feature in time level. As shown in Fig. 2, the proposed guidance law estimation network consists of three parts, including the convolutional part for gathering the deep characters of present network input, the fully connected part which is responsible for pre-estimation of the navigation ratio, and LSTM part which is in charge of operating final estimation based on pre-estimation result, current value of the deep characters and previous estimation results.

1) CONVOLUTIONAL PART FOR FEATURE EXTRACTION

In each estimation, the data measured at past several measurement points is sent into the neural network.

To exploit numerous measurement data, the measurement series data is sent into the convolutional part to extract high level feature. The convolutional part consists of several modules, and each module is composed of CNN layer, maxpooling layer and batch normalization layer, as shown in Fig. 3.

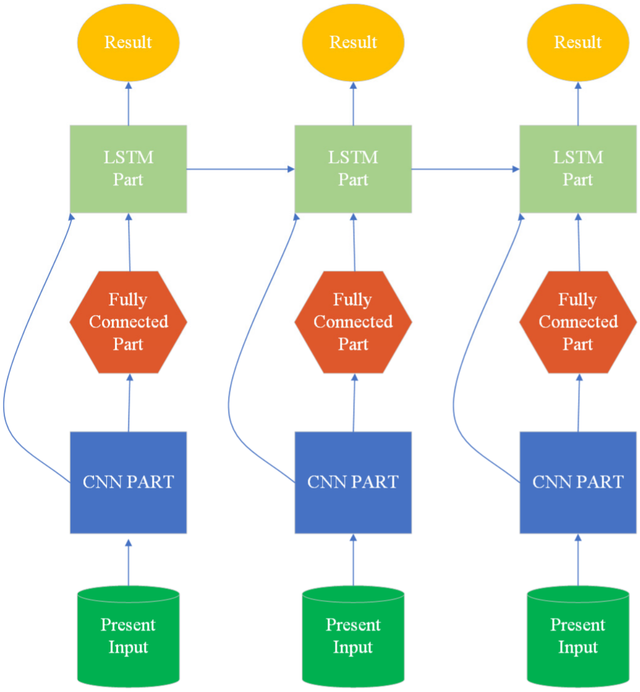

FIGURE 2. Structure of the estimation neural network.

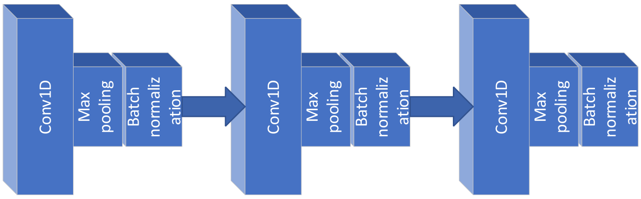

FIGURE 3. The feature extracting part of block.

The output of the jth feature in the ith CNN layer from the dth feature can be represented as [24]:

$$v_{ij}^{x,d} = \varphi\left(b_{ij} + \sum_{m=0} \sum_{p=0}^{P_i - 1} w_{ijm}^{p} v_{(i-1)m}^{x+p,d}\right) \quad \forall d = 1, \cdots, D \quad (6)$$

where φ is the activation function, b_{ij} is the bias, m is the index over the set of filters in the $(i-1)$th layer connected to the present featuremap, w_{ijm}^{P} is the weight at the position p of the convolutional kernel, and P_i is the length of the convolutional kernel.

The convolutional layer has a massive amount of data, and nonlinear activation function may greatly increase the training and online operating time of network. In view of this, each convolutional layer has linear activation function. The nonlinearity of feature extraction part nonlinearity is achieved by maxpooling and batch-normalization.

The convolutional neural layer's output is sent into a pooling layer so as to decrease the data scale. Mean pooling can relieve the intrinsic variance by limited convolution window and preserve the background information. In contrast, maxpooling can relieve the offset of data expectation and preserve the texture information. Under high variance of noise, max pooling also performs better than average pooling [33]. To extract the features that are invariant to background variables such as the evader's velocity and position, maxpooling is applied as follows:

$$v_{ij}^{x,d} = \max_{1\le q\le Q_i}\left(v_{(i-1)j}^{x+q,d}\right), \quad \forall d = 1, \cdots, D \tag{7}$$

To prevent overfit, the pooled data is accompanied by a batch normalization layer. The batch normalization process can be represented as [34]:

$$\begin{aligned}\mu_{\mathcal{B}} &\leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i\\ \sigma_{\mathcal{B}}^2 &\leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i-\mu_{\mathcal{B}})^2\\ \hat{x}_i &\leftarrow \frac{x_i-\mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2+\epsilon}}\\ y_i &\leftarrow \gamma\hat{x}_i+\beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i)\end{aligned} \tag{8}$$

where x_i is the input value of batch normalization layer over a mini-batch, γ, β are the batch normalization parameters to be learned, y_i is the normalized value, ϵ is a small floating point to avoid div/0 problem. During the training process, the batch normalization process smooths the loss variation in optimization problem and enables larger learning rate [35].

The end of the convolutional part is a flatten layer. The processed 2D arrays is flatten into 1D array. The extracted features are then sent into the fully connected part to operate pre-estimation. The detailed parameters of the convolutional part are shown in Fig. 4. The convolutional kernel size is 71-35-26-22-5-5.

2) FULLY CONNECTED PART FOR PRE-ESTIMATION

The fully connected part has multiple layers to fit the nonlinear mapping relation between extracted features and guidance law coefficient. The activation function can greatly influence the performance of fully connected network, and the Swish activation function is applied as follows [36]:

$$\begin{aligned}f(x) &= x\sigma(x)\\ \sigma(x) &= \frac{1}{1+e^{-x}}\end{aligned} \tag{9}$$

The Swish activation function can produce negative outputs for small negative inputs, so the robustness against extreme values can be improved.

The fully connected part has 7 hidden units. The output of fully connected part can be seen as a pre-estimation of the navigation ratio obtained based on the present input of the neural network. The final result is obtained by the LSTM part.

3) FINAL RESULT BY LSTM PART

According to the analysis of APN in Section 2, the navigation ratio may switch when the pursuer is near to the evader. It is worth noting that the measurement is time dependent. During training, the LSTM part at the end of the estimation network can capture the switching character of navigation ratio' and obtain more accurate estimation results online.

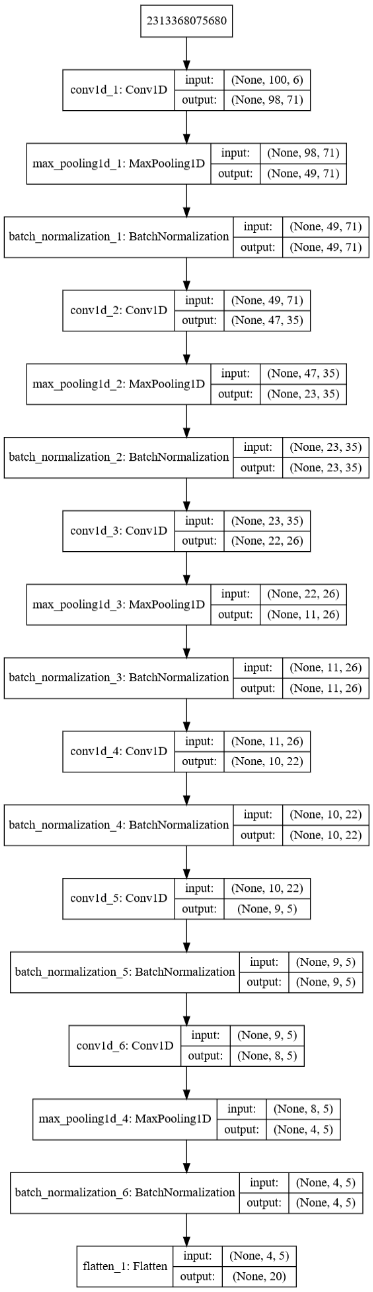

FIGURE 4. Detailed parameters of the convolutional part.

The research on deep reinforcement learning shows that additional recurrent unit of estimation network can aggregate past partial information to capture the influence of hidden states in the sequential information [37], [38]. For the guidance law estimation, the recurrent structure can capture the influence of unknown time delay and unmodelled relationship by analyzing the time dependence so as to eliminate the error caused by them.

The computation of LSTM cell at time step t can be represented as [39]:

$$\begin{aligned}
\boldsymbol{g}^{u} &= \sigma\left(\boldsymbol{W}^{u}\boldsymbol{h}_{t-1} + \boldsymbol{I}^{u}\boldsymbol{x}_{t}\right)\\
\boldsymbol{g}^{f} &= \sigma\left(\boldsymbol{W}^{f}\boldsymbol{h}_{t-1} + \boldsymbol{V}\boldsymbol{x}_{t}\right)\\
\boldsymbol{g}^{o} &= \sigma\left(\boldsymbol{W}^{o}\boldsymbol{h}_{t-1} + \boldsymbol{I}^{o}\boldsymbol{x}_{t}\right)\\
\boldsymbol{g}^{c} &= \tanh\left(\boldsymbol{W}^{c}\boldsymbol{h}_{t-1} + \boldsymbol{I}^{c}\boldsymbol{x}_{t}\right)\\
\boldsymbol{m}_{t} &= \boldsymbol{g}^{f} \odot \boldsymbol{m}_{t-1} + \boldsymbol{g}^{u} \odot \boldsymbol{g}^{c}\\
\boldsymbol{h}_{t} &= \tanh\left(\boldsymbol{g}^{o} \odot \boldsymbol{m}_{t}\right)
\end{aligned} \tag{10}$$

where $\boldsymbol{h}$ is the hidden state vector, $\boldsymbol{m}$ is the memory state vector, σ is the activation function, $\boldsymbol{W}^{u}, \boldsymbol{W}^{f}, \boldsymbol{W}^{o}, \boldsymbol{W}^{c}$ are recurrent weight matrices and $\boldsymbol{I}^{u}, \boldsymbol{I}^{f}, \boldsymbol{I}^{o}, \boldsymbol{I}^{c}$ are projection matrices.

The LSTM part's hidden number is set to 4. Under the limit of pursuer's overload, the navigation ratio is within the range of 2 to 6. In view of this, the output of neural network is limited within a reasonable range with hard-sigmoid activation:

$$\sigma_{\text{hard}}(x) = 2 + 4 * \max\left(0, \min\left(1, \frac{x+1}{2}\right)\right) \tag{11}$$

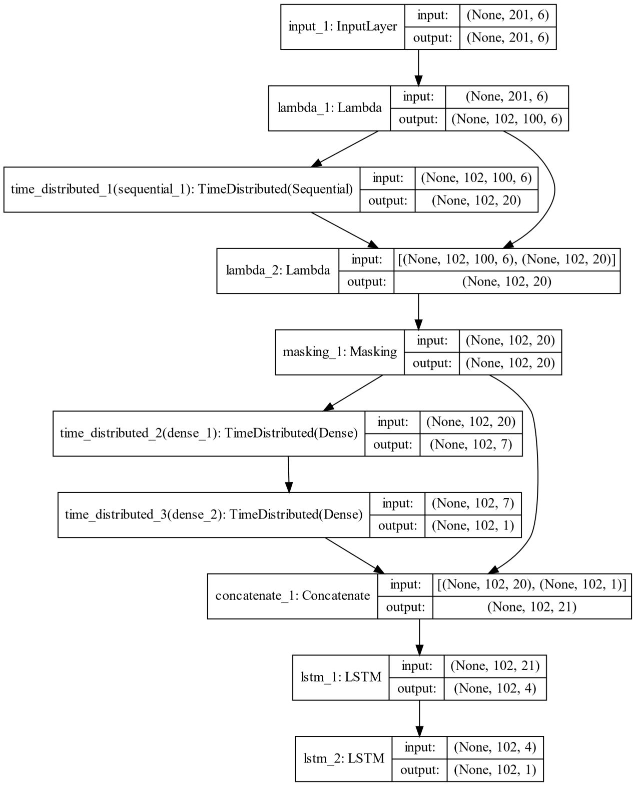

FIGURE 5. Detailed parameters of the modified RCNN.

The detailed parameters of the estimation network are shown in Fig. 5. "lambda_1" represents the user defined layer for data preprocessing. "lambda_2" represents the user defined layer that records zero-padding position in the output of convolutional process. "time_distributed_1(sequential_1)" represents the convolutional part.

C. OPTIMIZATION METHOD

Due to complex structure of RCNN, the estimation network may drop into local optimum when trained by traditional adaptive gradient optimization method. The newly developed AdaBound method can address this difficulty by adaptively tighten the clip interval and keep the optimizer steady [40].

The optimization process of AdaBound can be represented as [41]:

$$\begin{aligned}
g_t &= \nabla f_t(x_t)\\
m_t &= \beta_{1t} m_{t-1} + (1-\beta_{1t})\, g_t\\
v_t &= \beta_2 v_{t-1} + (1-\beta_2)\, g_t^2\\
V_t &= \operatorname{diag}(v_t)\\
\hat{\eta}_t &= \operatorname{Clip}\left(\alpha/\sqrt{V_t}, \eta_l(t), \eta_u(t)\right)\\
\eta_t &= \hat{\eta}_t/\sqrt{t}\\
x_{t+1} &= \Pi_{\mathcal{F},\operatorname{diag}\left(\eta_t^{-1}\right)}\left(x_t - \eta_t \odot m_t\right)
\end{aligned} \tag{12}$$

where $f(x)$ is the objective function, α, $\{\beta_{1t}\}_{t=1}^{T}$, β_2 are the initial step size, η_l is the lower bound function, η_u is the upper bound function, Clip function reduces the gradients larger than a threshold to avoid gradient explosion. Trained with AdaBound, the efficiency of proposed method will be verified in the next Section.

V. SIMULATIONS AND DISCUSSION

Due to different initial states, navigation ratio and time delay in estimation of the pursuer's acceleration, 4000 samples are obtained by trajectory simulation. Each sample has a flight time in the range of 8-12 s. The samples are separated into training and validating dataset of 2000 samples and testing dataset of 2000 samples. The trained network's performance is compared to that of the interactive multiple model estimation method using unscented Kalman filtering (UKF) technique [8]. The detail of the simulation is given as below.

A. PARAMETERS SETTING

1) PARAMETERS OF THE PURSUIT-EVASION PROCESS

The initial distance between the pursuer and the evader is set in the range of 90-110 km, and the initial velocity is set in the range of 6-8 km/s. The evader is assumed to operate a bang-bang maneuver or a sinusoidal maneuver with a maximum acceleration of $2g$ in each dimension.

2) PARAMETER OF MEASUREMENT

The measurement interval is 0.02 s, and the estimation interval is 0.02 s. In each estimation, the measurement data in past 2 seconds is sent into the network. The measurement standard deviation is set as: $r_R = 0.001R$, $r_{\sigma e} = 1$, $r_{\sigma a} = 1$, $r_{\text{aTx}} = 0.01a_{\text{Tx}}$, $r_{\text{aTy}} = 0.01a_{\text{Ty}}$, $r_{\text{aTz}} = 0.01a_{\text{Tz}}$. The time delay τ_r in estimation of evader's acceleration is set in the range of 0.06 s-0.1 s.

3) PARAMETERS OF NEURAL NETWORK FOR COMPARISON

To verify the performance of proposed recurrent convolutional structure, a single convolutional neural network and a single recurrent neural network are built for ablation tests. The proposed network consists of 6 1D convolutional layers, 2 fully connected layers and 2 LSTM cells. To ensure the fairness of comparison, each of the three neural networks has 10 layers. The structure of single CNN is set to 8 1D convolutional layers(kernel size 71-35-26-22-15-10-8-4)+ 2 fully connected layers(the same as the proposed RCNN's fully connected part). The structure of single RNN is set to 10 LSTM cells (number of units 500-300-150-80-50-25-13-8-4-1).

4) SUPER-PARAMETERS OF NEURAL NETWORK TRAINING

The training loss function is "mean_absolute_percentage_error"(MAPE) in Keras. The training is carried out on Nvidia Quadro2000. The environment is tensorflow1.14+ Keras2.24. The batch size is set to 128. The initial learning rate is set to 0.001. After 400 epochs of training, the network is used for guidance law estimation.

5) IMM PARAMETER

An IMM filter is designed based on three APN models (APN_3, APN_4 and APN_5) with navigation ratios set to 3, 4 and 5, respectively. The process noise matrix of the APN_3, APN_4 and APN_5 models is $\mathbf{Q}_1 = \mathbf{Q}_2 = \mathbf{Q}_3 = \text{diag}\left([1, 1, 1, 1, 1, 1]\right) \big/ 1000$. The measurement noise matrix is $\mathbf{R}_1 = \mathbf{R}_2 = \mathbf{R}_3 = \text{diag}([25, 0.01, 0.01])$. The prior probabilities are $\mu_0 = \left[0.333\ 0.333\ 0.334\right]$. The mode transition probability matrix is set as

$$p_{ij} = \begin{bmatrix} 0.996 & 0.002 & 0.002 \\ 0.002 & 0.996 & 0.002 \\ 0.002 & 0.002 & 0.996 \end{bmatrix}$$

The IMM is carried out on Intel Xeon Bronze 3104.

B. RESULT AND ANALYSIS

For CNN, RNN, and the proposed network, the training takes 6435 s, 18809 s and 16021 s and gets MAPE of 2.59%, 2.08% and 0.78% in the training dataset, respectively. The evaluation of the loss function's value is shown in Fig. 6.

The trained network is firstly tested in the test dataset. The whole course mean absolute percentage error(WMAPE) is defined as the mean of absolute percentage error of each estimation in a sample's whole flight course. Each sample's WMAPE is shown in Fig. 7. The Maximum WMAPE of CNN, RNN, and the proposed network is 20.72%, 13.45% and 4.44%, respectively; and their MAPE in test set is 4.0731%, 3.4792% and 1.42615%, respectively. The test result shows the proposed method's improvement upon single RNN or CNN structure.

For CNN, RNN and the proposed neural network, each estimation takes 0.0492 ms, 1.2473 ms, 0.0998 ms, respectively; while for IMM method, each estimation takes 8.837 ms. This indicates all three neural networks can reduce the computation load. Among the three networks, RNN consumes longer time compared with the other two.

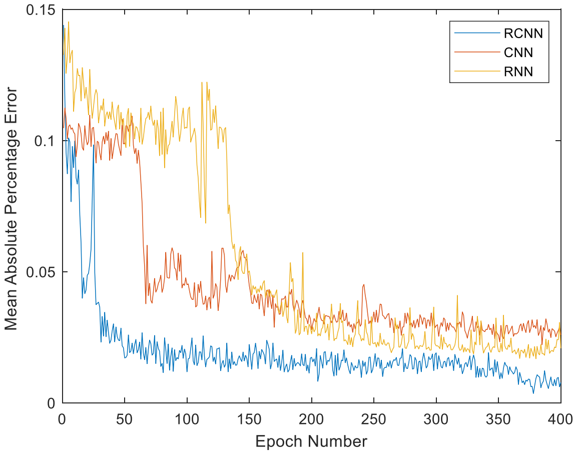

FIGURE 6. Loss function of the estimation neural network.

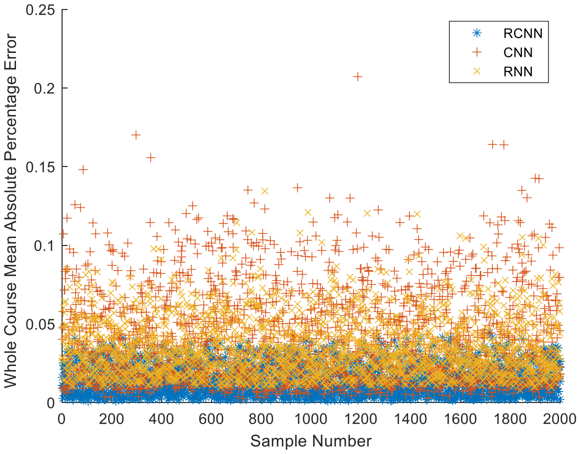

FIGURE 7. Sample whole course mean absolute percentage error of the estimation neural network in test dataset.

To compare the performance between proposed RCNN and the IMM method, the navigation ratio of a trajectory is estimated. In the scene inertial coordinate, the initial relative position is set to $[1 \times 10^5, 0, 0]$m, and the initial velocity is set to $[-7.5 \times 10^3, -1 \times 10^2, -8 \times 10^2] m/s$. When $t = 5$s, the navigation ratio switches from 3 to 5. The time delay τ_r in estimation of evader's acceleration is set to 0.08 s. The mode probability of the IMM method is shown in Fig. 8. The estimation result of each method are shown in Fig. 9.

The WMAPE of the IMM method is 3.8687%, while that of the CNN, RNN, and the proposed RCNN method is 4.6621%, 2.6471% and 1.3162%, respectively, indicating accuracy of proposed RCNN method is significantly higher than that of the IMM method or single neural network structure. The CNN without recurrent structure exhibits a large fluctuation in estimation result, which indicates single CNN is not suitable for the guidance law estimation with unknown characters. RNN can converge to 3 quickly at initially stage; however, with the time goes on, the performance becomes

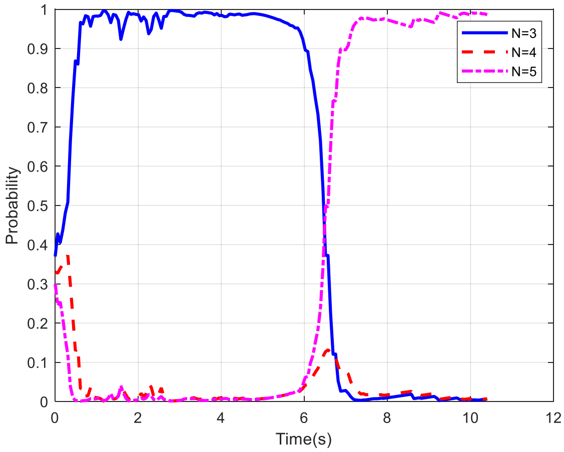

FIGURE 8. **Probability of the IMM method.**

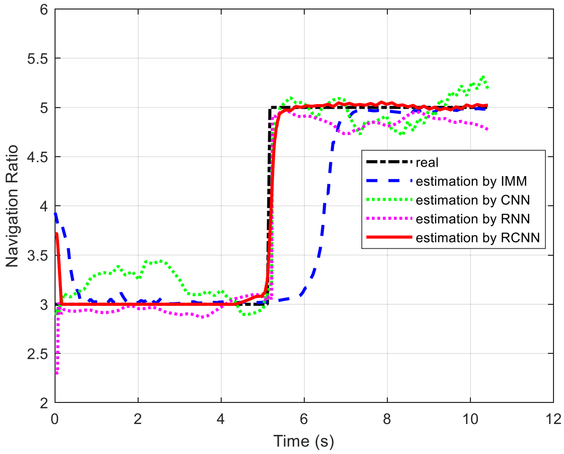

FIGURE 9. **Guidance law estimation by IMM, CNN, RNN and RCNN.**

worse. This shows the single RNN has disadvantage in processing numerous data measured in long period. The RCNN structure can exploit the time dependency, so it performs well at the switching point. At the beginning, the proposed method converges to 3 in 0.2 s and keeps steady, while IMM costs 0.5 s to converge. When the navigation ratio switches, the proposed RCNN method can track the rapidly changing navigation ratio and reach 5 in 0.5 s, while IMM converges after 1.2 s. This shows the proposed method can converge faster in face of navigation ratio switching compared with IMM method.

VI. CONCLUSION

This paper proposes a novel recurrent convolutional neural network-based method for a highly maneuvering evader to estimate the guidance law of a chasing pursuer. The time dependency, transient change and disturbance of unknown factors are considered. The convolutional structure of estimation network is adopted to extract features from the data obtained by multiple previous measurements, and then the features extracted are processed by recurrent structure to capture the time dependency and influence of unknown factors. With combined structure of convolutional layer and LSTM cell, the proposed method can realize efficient and accurate estimation of guidance law for the evader. The result of ablation test accords with the analysis of single CNN or RNN structure, indicating the proposed structure has a significantly improved performance over single CNN or RNN structure. Compared with IMM filter methods, the proposed method involves less computation load and can converge faster when facing navigation ratio switching. This paper applies RCNN to the field of aerospace and builds an end-to-end method for guidance law estimation of APN. Future work will focus on the neural network-based estimation of various guidance law forms. The structure optimization method for recurrent convolutional neural network shall also be further studied.

REFERENCES

[1] F. Imado and S. Miwa, “Fighter evasive maneuvers against proportional navigation missile,” *J. Aircraft*, vol. 23, no. 11, pp. 825–830, Nov. 1986.

[2] B. Jung, K.-S. Kim, and Y. Kim, “Guidance law for evasive aircraft maneuvers using artificial intelligence,” in *Proc. AIAA Guid., Navigat., Control Conf. Exhib.*, Aug. 2003. [Online]. Available: https://arc.aiaa.org/doi/abs/10.2514/1.51765

[3] T. Shima, “Optimal cooperative pursuit and evasion strategies against a homing missile,” *J. Guid., Control, Dyn.*, vol. 34, no. 2, pp. 414–425, Mar. 2011.

[4] V. Shaferman and T. Shima, “Cooperative multiple-model adaptive guidance for an aircraft defending missile,” *J. Guid., Control, Dyn.*, vol. 33, no. 6, pp. 1801–1813, Nov. 2010.

[5] R. Fonod and T. Shima, “Multiple model adaptive evasion against a homing missile,” *J. Guid., Control, Dyn.*, vol. 39, no. 7, pp. 1578–1592, Jul. 2016.

[6] D. K. Sang and M. J. Tahk, “Guidance law switching logic considering the seeker’s field-of-view limits,” in *Proc. Inst. Mech. Eng. G, J. Aerosp. Eng.*, vol. 223, no. 8, pp. 1049–1058, 2019.

[7] J. Yun and C.-K. Ryoo, “Missile guidance law estimation using modified interactive multiple model filter,” *J. Guid., Control, Dyn.*, vol. 37, no. 2, pp. 484–496, Mar. 2014.

[8] X. Zou, D. Zhou, R. Du, and J. Liu, “Active defense guidance law via cooperative identification and estimation,” *J. Guid., Control, Dyn.*, vol. 41, no. 11, pp. 2507–2512, Nov. 2018.

[9] G. Xie, L. Sun, T. Wen, X. Hei, and F. Qian, “Adaptive transition probability matrix-based parallel IMM algorithm,” *IEEE Trans. Syst., Man, Cybern., Syst.*, to be published, doi: 10.1109/tsmc.2019.2922305.

[10] Y. L. Bengio and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[11] Z. Li, X. Sun, C. Hu, G. Liu, and B. He, “Neural network based online predictive guidance for high lifting vehicles,” *Aerosp. Sci. Technol.*, vols. 82–83, pp. 149–160, Nov. 2018.

[12] B. Zhao, S. Xu, J. Guo, R. Jiang, and J. Zhou, “Integrated strapdown missile guidance and control based on neural network disturbance observer,” *Aerosp. Sci. Technol.*, vol. 84, pp. 170–181, Jan. 2019.

[13] X. Shi, Z. Chen, H. Wang, D. Y. Yeung, W. K. Wong, and W. C. Woo, “Convolutional LSTM network: A machine learning approach for precipitation nowcasting,” in *Proc. NIPS*, 2015. [Online]. Available: http://papers.nips.cc/paper/5955-convolutional-lstm-network-a-machine-learning-approach-for-precipitation-nowcasting

[14] R. R. Kumar, H. Seywald, and E. M. Cliff, “Near-optimal three-dimensional air-to-air missile guidance against maneuvering target,” *J. Guid., Control, Dyn.*, vol. 18, no. 3, pp. 457–464, May 1995.

[15] X.-T. Wu, X. Wang, J.-H. Yang, and X. Zhang, “Air-to-air missile tracking and guidance law identification based on CKF,” in *Proc. 4th Int. Conf. Mach., Mater. Comput. (MACMC)*, 2017. [Online]. Available: https://www.atlantis-press.com/proceedings/macmc-17/25888576

[16] M. Guelman, “Qualitative study of proportional navigation,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-7, no. 4, pp. 637–643, Jul. 1971.

[17] V. Shaferman, “Near optimal evasion from acceleration estimating pursuers,” in *Proc. AIAA Guid., Navigat., Control Conf.*, Jan. 2017. [Online]. Available: https://arc.aiaa.org/doi/pdf/10.2514/6.2017-1014

[18] Z. Hou, L. Liu, Y. Wang, J. Huang, and H. Fan, "Terminal impact angle constraint guidance with dual sliding surfaces and model-free target acceleration estimator," *IEEE Trans. Control Syst. Technol.*, vol. 25, no. 1, pp. 85–100, Jan. 2017.

[19] K. R. Babu, I. G. Sarma, and K. N. Swamy, "Switched bias proportional navigation for homing guidance against highly maneuvering targets," *J. Guid., Control, Dyn.*, vol. 17, no. 6, pp. 1357–1363, Nov. 1994.

[20] R. Du, X. Zou, D. Zhou, and J. Liu, "Interactive multiple model filter for tracking a pursuer with proportional navigation guidance law," *J. Dyn. Syst., Meas., Control*, vol. 140, no. 8, 2018, Art. no. 081003.

[21] J. Connor, R. Martin, and L. Atlas, "Recurrent neural networks and robust time series prediction," *IEEE Trans. Neural Netw.*, vol. 5, no. 2, pp. 240–254, Mar. 1994.

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

[23] Y. Kim, "Convolutional neural networks for sentence classification," 2014, *arXiv:1408.5882*. [Online]. Available: https://arxiv.org/abs/1408.5882

[24] J. Yang, M. N. Nguyen, P. P. San, X. L. Li, and S. Krishnaswamy, "Deep convolutional neural networks on multichannel time series for human activity recognition," in *Proc. Int. Joint. Conf. Artif. Intell (IJCAI)*, 2015. [Online]. Available: https://www.aaai.org/ocs/index.php/IJCAI/IJCAI15/paper/viewPaper/10710

[25] N. Kalchbrenner and P. Blunsom, "Recurrent convolutional neural networks for discourse compositionality," 2015, *arXiv:1306.3584*. [Online]. Available: https://arxiv.org/abs/1306.3584

[26] P. O. Pinheiro and R. Collobert, "Recurrent convolutional neural networks for scene labeling," in *Proc. ICML*, 2014. [Online]. Available: http://proceedings.mlr.press/v32/pinheiro14.html

[27] S. Lai, "Recurrent convolutional neural networks for text classification," in *Proc. AAAI*, 2015. [Online]. Available: https://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/viewPaper/9745

[28] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *Proc. CVPR*, 2015. [Online]. Available: http://openaccess.thecvf.com/content_cvpr_2015/html/Donahue_Long-Term_Recurrent_Convolutional_2015_CVPR_paper.html

[29] G. C. Wu, Q. Liu, T. Y. Chai, and S. J. Qin, "Abnormal condition diagnosis through deep learning of image sequences for fused magnesium furnaces," *Zidonghua Xuebao/Acta Autom. Sinica*, vol. 45, no. 8, pp. 1475–1485 2019.

[30] F. Karim, S. Majumdar, H. Darabi, and S. Chen, "LSTM fully convolutional networks for time series classification," *IEEE Access*, vol. 6, pp. 1662–1669, 2018.

[31] F. Karim, S. Majumdar, and H. Darabi, "Insights into LSTM fully convolutional networks for time series classification," *IEEE Access*, vol. 7, pp. 67718–67725, 2019.

[32] F. Karim, S. Majumdar, H. Darabi, and S. Harford, "Multivariate LSTM-FCNs for time series classification," *Neural Netw.*, vol. 116, pp. 237–245, Aug. 2019.

[33] Y-L. Boureau, F. Bach, Y. LeCun, and J. Ponce, "Learning mid-level features for recognition," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.205.8511&rep=rep1&type=pdf

[34] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015, *arXiv:1502.03167*. [Online]. Available: https://arxiv.org/abs/1502.03167

[35] S. Santurkar, D. Tsipras, A. Ilyas, and A. Madry, "How does batch normalization help optimization?" in *Proc. NIPS*, 2018. [Online]. Available: https://papers.nips.cc/paper/7515-how-does-batch-normalization-help-optimization

[36] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions," 2017, *arXiv:1710.05941*. [Online]. Available: https://arxiv.org/abs/1710.05941

[37] X. Li, L. Li, J. Gao, X. He, J. Chen, L. Deng, and J. He, "Recurrent reinforcement learning: A hybrid approach," 2015, *arXiv:1509.03044*. [Online]. Available: https://arxiv.org/abs/1509.03044

[38] A. El Sallab, M. Abdou, E. Perot, and S. Yogamani, "Deep reinforcement learning framework for autonomous driving," in *Proc. Int. Symp. Electron. Imag. Sci. Technol.*, 2015. [Online]. Available: https://chinesesites.library.ingentaconnect.com/content/ist/ei/2017/00002017/00000019/art00012

[39] A. Graves, *Supervised Sequence Labelling With Recurrent Neural Networks*, vol. 385. London, U.K.: Springer, 2012.

[40] D. Wu, Y. Yuan, and Y. Tan, "Optimize TSK fuzzy systems for regression problems: Mini-batch gradient descent with regularization, droprule and adabound (MBGD-RDA)," 2019, *arXiv:1903.10951*. [Online]. Available: https://arxiv.org/abs/1903.10951

[41] L. Luo, Y. Xiong, Y. Liu, and X. Sun, "Adaptive gradient methods with dynamic bound of learning rate," 2019, *arXiv:1902.09843*. [Online]. Available: https://arxiv.org/abs/1902.09843

HUIBING SHAO received the B.Eng. degree in electrical engineering and automation from Harbin Engineering University, in 1998, and the M.Eng. degree in navigation, guidance, and control from the Second Research Institute of Aerospace Science and Engineering Group, Beijing, China, in 2005. Since 2005, he has been working with the Beijing Institute of Control and Electronic Technology. He is currently pursuing the Ph.D. degree in aeronautical and astronautical science and technology, since 2013. His research interests include missile guidance, flight control, intelligent control, recurrent convolutional neural networks, and robust control.

YEPENG HAN received the B.Eng. degree in aircraft designing and the M.Eng. degree in aeronautical and astronautical science and technology from the Harbin Institute of Technology, Harbin, China, in 2014 and 2016, respectively, where he is currently pursuing the Ph.D. degree in aeronautical and astronautical science and technology. His research interests include missile guidance and control, intelligent control, and nonlinear control systems.

CHANGZHU WEI received the B.S. degree in flight vehicle design and engineering and the Ph.D. degree in aeronautical and astronautical science and technology from the Harbin Institute of Technology, Harbin, China, in 2005 and 2010, respectively. From 2013 to 2015, he was an Assistant Professor with the School of Astronautics, Harbin Institute of Technology, where he has been an Associate Professor, since 2016. He is the author of one book, more than 30 articles, and two inventions. His research interests include advanced guidance and control methods of flight vehicle, and intelligent control.

RUIMING WANG received the B.Eng. degree in automation from Northeastern University, Shenyang, China, in 2019. He is currently pursuing the Ph.D. degree in aeronautical and astronautical science and technology with the Harbin Institute of Technology, Harbin, China. His research interests include missile guidance and control, and neural networks.