A DEEP NEURAL NETWORK BASED MANEUVERING-TARGET TRACKING ALGORITHM

Jingxian Liu[1,2], *Zulin Wang*[1], *Mai Xu*[1*], *Jie Ren*[1]

[1]School of Electronic and Information Engineering, Beihang University, Beijing, China
[2]School of Computer Science and Communication Engineering,
Guangxi University of Science and Technology, Liuzhou, China
[*]Correspondence: maixu@buaa.edu.cn

ABSTRACT

In the field of maneuvering-target tracking (MTT), the targets with changeable and uncertain maneuvering movements cannot be tracked precisely because there always exist time delays of maneuvering model estimation with traditional MTT algorithms. To solve this problem, we propose a deep MTT (DeepMTT) algorithm based on a deep neural network, which can quickly track maneuvering targets once it has been well trained by abundant off-line trajectory data from existent maneuvering targets. To this end, we first build a Large-scale trajectory database to offer abundant off-line trajectory data for network training. Second, the DeepMTT algorithm is developed based on a deep neural network, which consists of three bidirectional long short-term memory layers, a filtering layer, a maxout layer and a linear output layer. The simulation results verify that our DeepMTT algorithm outperforms other state-of-the-art MTT algorithms.

Index Terms— Maneuvering-target tracking, bidirectional long short-term memory network, multiple models, trajectory database

1. INTRODUCTION

Maneuvering-target tracking (MTT) is a critical issue in the field of target tracking [1, 2]. Traditional MTT algorithms [3, 4] combine multiple movement models with different weights to estimate the maneuvering model of the target, for improving the tracking performance. However, the maneuvering models are always unknown and changing in practical MTT scenarios, e.g., in civil airport surveillance. When traditional MTT algorithms accumulate sufficient information of target states to correctly estimate the previous maneuvering model, the current one has changed. As shown in Figure 1, in a 2 dimensional MTT scenario, the estimation of the acceleration model with the state-of-the-art MTT algorithm, i.e., HGMM algorithm [5], cannot timely catch the current model when it changes. This time-delay issue of model estimation causes the inaccuracy of target tracking when using traditional MTT algorithms.

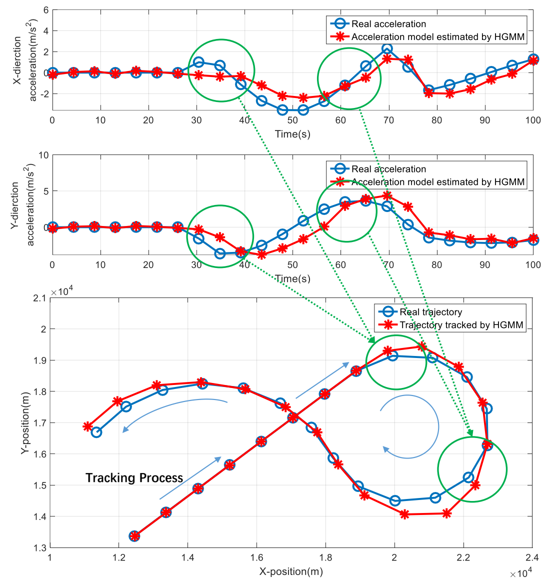


Fig. 1: Model estimation and target tracking results with the HGMM algorithm.

To solve the time-delay issue, we propose a new deep MTT (DeepMTT) algorithm from the data-driven perspective. In our DeepMTT algorithm, we design a "smart" model based on neural network, which can learn to timely and precisely predict trajectories of maneuvering targets with different observations. Although shallow neural networks have been applied to the field of target tracking since the 1990s [6], they are only complementary methods to provide more information in the tracking process [7, 8] and cannot solve the time-delay issue in MTT.

In contrast to the shallow neural networks of [7, 8], our DeepMTT algorithm learns the maneuvering models from observations based on a deep neural network, called DeepMTT network, which contains a filtering layer, three bidirection-

This work was supported by NSFC under Grants 61471022, 61876013, 61573037 and 61841202, by the Fok Ying Tung Education Foundation under Grant 151061, and by the Research Foundation of Educational Commission of Guangxi Zhuang Autonomous Region (2018KY0322).

Mai Xu is the corresponding author of this paper.

al long short-term memory (LSTM) [9–11] layers, a maxout layer [12] and a linear output layer. This DeepMTT network can directly estimate the trajectories of the maneuvering targets with no maneuvering model estimation procedure. Thus, the time-delay issue can be eliminated to a great extent. To this end, we first build a LArge-Scale Trajectory (LAST) database, which contains 10 million trajectories with the corresponding observations that cover normal maneuvering cases in civil airport surveillance. Using the LAST database, we design and train a deep network with a great amount of trainable parameters, which well fit different models of maneuvering movements. Finally, we directly and timely track the maneuvering target with the well-trained DeepMTT network. The simulation results show that our DeepMTT algorithm outperforms other state-of-the-art MTT algorithms. The main contributions of our work are summarized as follows:

- An LAST database is built to offer sufficient training data for tracking maneuvering targets.
- A DeepMTT network is proposed to track a maneuvering target by learning from the extensive trajectory data of the LAST database.

2. LARGE-SCALE TRAJECTORY DATABASE OF MANEUVERING TARGET

In this paper, the LAST database is established for MTT. The samples in our database contain the observation and ground-truth of trajectories, which constitute the input-output pairs of the DeepMTT network. In fact, it is difficult to obtain sufficient ground-truth data of maneuvering-target trajectory in civil airport surveillance. Instead, based on the state space model (SSM) [3], we design a trajectory generator that can simulate the segments of different maneuvering target trajectories, which are viewed as samples in our LAST database. Here, 10 million samples are generated by the trajectory generator to establish the LAST database, which cover all common cases of maneuvering targets in the civil airport surveillance [3, 13].

Specifically, the SSM of the trajectory generator is defined as follows:

$$\textbf{Transition equation}: \ \boldsymbol{x}_k = \boldsymbol{F}\boldsymbol{x}_{k-1} + \boldsymbol{n}, \tag{1a}$$

$$\textbf{Observation equation}: \ \boldsymbol{z}_k = \boldsymbol{h}(\boldsymbol{x}_k) + \boldsymbol{m}, \tag{1b}$$

where $\boldsymbol{x}_k$ and $\boldsymbol{z}_k$ are the target state and the corresponding observation at time step k, respectively. In (1a), $\boldsymbol{F}$ is the transition matrix, and $\boldsymbol{n}$ is the transition noise. In (1b), $\boldsymbol{h}(\cdot)$ is the nonlinear observation, and $\boldsymbol{w}$ is the observation noise. According to the SSM, the ground-truth $\boldsymbol{x}_{1:K} \triangleq \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_K\}$ of trajectories is generated by (1a), and the observations $\boldsymbol{z}_{1:K} \triangleq \{\boldsymbol{z}_1, \boldsymbol{z}_2, ..., \boldsymbol{z}_K\}$ are generated by (1b). In SSM, $\boldsymbol{x}_k$ is defined as $[d_{x,k}, \ d_{y,k}, \ v_{x,k}, \ v_{y,k}]^{\mathrm{T}}$, where $[d_{x,k}, \ d_{y,k}]^{\mathrm{T}}$ is the two-dimensional (2-D) position, and $[v_{x,k}, \ v_{y,k}]^{\mathrm{T}}$ is the corresponding velocity[1]. According to [14], $\boldsymbol{F}$ in (1a) is defined as

Table 1: Ranges of Maneuvering Target Trajectories

Content	**Ranges**
Distance from radar	$0.5 \sim 20$ $(nautical\ mile)$
Velocity of aircraft	$0 \sim 340$ (m/s)
Maneuvering turn rate	$-10 \sim 10$ $(^\circ/s)$

$$\boldsymbol{F} = \begin{bmatrix} 1 & 0 & s_\tau & 0 \\ 0 & 1 & 0 & s_\tau \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

for constant velocity (CV) movement, and $\boldsymbol{F}$ is defined as

$$\boldsymbol{F} = \begin{bmatrix} 1 & 0 & \frac{\sin(\alpha s_\tau)}{\alpha} & \frac{\cos(\alpha s_\tau)-1}{\alpha} \\ 0 & 1 & \frac{1-\cos(\alpha s_\tau)}{\alpha} & \frac{\sin(\alpha s_\tau)}{\alpha} \\ 0 & 0 & \cos(\alpha s_\tau) & -\sin(\alpha s_\tau) \\ 0 & 0 & \sin(\alpha s_\tau) & \cos(\alpha s_\tau) \end{bmatrix} \tag{3}$$

for constant turn (CT) movement. In the above equations, s_τ is the sampling interval of trajectories, which is set to be 0.1 s in our LAST database; α is the turn rate of maneuvering target. For radar tracking, $\boldsymbol{z}_k = [\theta_k, \ r_k]^{\mathrm{T}}$ is the radar observation vector, which contains azimuth value θ_k and distance value r_k. In addition, $h(\cdot)$ of (1b) is denoted as follows:

$$\underbrace{\begin{bmatrix} \theta_k \\ r_k \end{bmatrix}}_{\boldsymbol{z}_k} = \begin{bmatrix} \arctan \dfrac{d_{y,k}}{d_{x,k}} \\ \sqrt{d_{x,k}^2 + d_{y,k}^2} \end{bmatrix} + \underbrace{\begin{bmatrix} m_\theta \\ m_r \end{bmatrix}}_{\boldsymbol{m}}, \tag{4}$$

where $\boldsymbol{m} = [m_\theta, \ m_r]^{\mathrm{T}}$ is the noise of observation vector, which contains the azimuth and distance parts, i.e., m_θ and m_r.

Using the SSM, $\boldsymbol{z}_{1:K} \triangleq \{\boldsymbol{z}_1, \boldsymbol{z}_2, ..., \boldsymbol{z}_K\}$ and $\boldsymbol{x}_{1:K} \triangleq \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_K\}$ can be generated, given the parameters of the SSM, i.e., the number of time steps K, initial state $\boldsymbol{x}_0$, noises $\boldsymbol{n}$ and $\boldsymbol{m}$, and transition matrix $\boldsymbol{F}$.

To meet the need for covering all possible maneuvering cases, we first set the basic parameters in Table 1. As we see in this table, the distance from the radar to the target covers the main detection range of the common airport surveillance radar (ASR), i.e., 0.5 $\sim$ 20 nautical miles (NM) (about 926 m $\sim$ 37,040 m) according to [15]. The velocity of our maneuvering target in the range of 0-340 m/s because civilian aircrafts rarely exceed the sound velocity [16], i.e., 340 m/s. The turn rate α, which decides the maneuvering cases, ranges from $-10^\circ/s$ to $10^\circ/s$ because turn rate α is normally less than $10^\circ/s$ for the common civilian aircraft according to [3]. The smallest interval between turn rates is $0.1^\circ/s$, which provides sufficient differentiation to track the maneuvering target.

[1] In this paper, we only consider an X-Y plane coordinate of MTT.

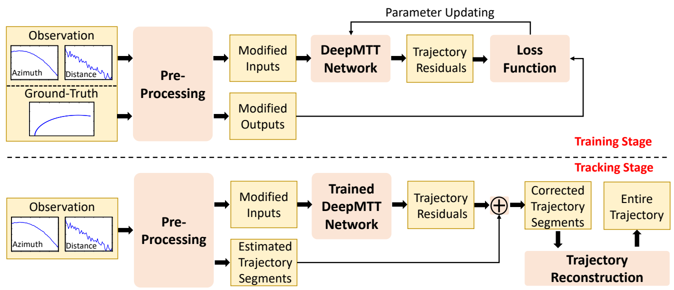


Fig. 2: The framework of DeepMTT algorithm.

According to Table 1, we derive the parameters of the SSM as follows: (1) The length of trajectory segments t_{segmet} is set to be 5 s because the fixed length of trajectory segments is required in our DeepMTT network. (2) The initial state $\boldsymbol{x}_0$ can be randomly sampled from the range of distance and velocity. (3) The deviation of accelerated velocity in transition noise $\boldsymbol{n}$ is randomly sampled in the range of $[8\ m/s^2, 13\ m/s^2]$ according to [3]. The deviations of azimuth noise σ_θ and distance noise σ_r are randomly sampled in the range of $[0.401°, 0.516°]$ and $[8\ m, 13\ m]$ according to [5, 17], respectively. (4) The maneuvering cases depend on the value of turn rate α, which is randomly sampled from $-10°/s$ to $10°/s$ with the sampling interval of $0.1°/s$ as discussed. If $\alpha = 0$, (2) is used as the transition matrix $\boldsymbol{F}$ to generate the trajectory. Otherwise, (3) is used as the transition matrix $\boldsymbol{F}$ to generate the trajectory.

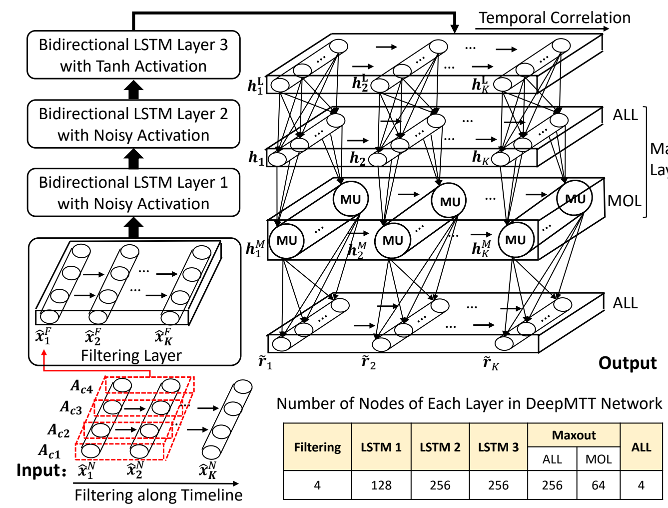
After all settings have been obtained for the SSM, we generate 10 million samples to build our LAST database, in which all samples are randomly and equally grouped into 100 thousand batches. In the training stage, we keep training our DeepMTT network by randomly selecting the batches in LAST database until the estimation deviation of target position is smaller than a threshold ϵ[2].

3. DEEP MANEUVERING-TARGET TRACKING ALGORITHM

In this section, we detail our DeepMTT algorithm. Unlike the traditional tracking algorithms, our DeepMTT algorithm tracks the target based on a DeepMTT network, which is trained off-line. The DeepMTT network can output residuals to directly correct the UKF tracking [18] results, by which the maneuvering trajectories are timely and precisely estimated. The DeepMTT algorithm contains two stages: training and tracking stages, as shown in Figure 2.

In the training stage, the original observations and ground truth are first pre-processed to generate the modified input-output pairs, which are suitable for training the DeepMTT network. Specifically, the azimuth and distance data are filtered by the UKF algorithm with the CV movement model (CV-UKF), to form the estimated trajectory segment: $\hat{\boldsymbol{x}}_{1:K} \triangleq \{\hat{\boldsymbol{x}}_1, \hat{\boldsymbol{x}}_2, ..., \hat{\boldsymbol{x}}_K\}$, which is similar to the ground truth

[2] ϵ is set to be 20 m in this paper.



Filtering	LSTM 1	LSTM 2	LSTM 3	Maxout		ALL
				ALL	MOL	
4	128	256	256	256	64	4

Fig. 3: Structure of the DeepMTT network.

of the trajectory. Then, $\hat{\boldsymbol{x}}_{1:K}$ is normalized to generate the final input data $\hat{\boldsymbol{x}}^N_{1:K} \triangleq \{\hat{\boldsymbol{x}}_1/C_{\max}, \hat{\boldsymbol{x}}_2/C_{\max}, ..., \hat{\boldsymbol{x}}_K/C_{\max}$, where $C_{\max}$ is the maximum absolute value in the elements of $\hat{\boldsymbol{x}}_{1:K}$. Meanwhile, the desired output of the DeepMTT network is the residual sequence $\boldsymbol{r}_{1:K}$, which is denoted as $\boldsymbol{r}_{1:K} = \boldsymbol{x}_{1:K} - \hat{\boldsymbol{x}}_{1:K}$, where $\boldsymbol{x}_{1:K}$ is the ground truth of the trajectory segment. Taking $\hat{\boldsymbol{x}}_{1:K}$ as a reference of the ground truth, $\boldsymbol{r}_{1:K}$ is the relative error between the ground truth and its reference.

Utilizing the modified input-output pairs, we train the DeepMTT network to minimize the loss $\mathcal{L} = \sqrt{\sum_{k=1}^{K} (\widetilde{\boldsymbol{r}}_k - \boldsymbol{r}_k)^2}$, where $\widetilde{\boldsymbol{r}}_k$ is the output of our DeepMTT network to predict the residual. The network structure is shown in Figure 3, which contains a filtering layer, three bidirectional LSTM layers, a maxout layer and a linear output layer. The filtering layer is used to smooth the input $\hat{\boldsymbol{x}}^N_{1:K}$ as follows:

$$\hat{d}^F_{x,k} = A_{1,1}\hat{d}^N_{x,k-4} + A_{2,1}\hat{d}^N_{x,k-3} \cdots + A_{5,1}\hat{d}^N_{x,k}, \quad (5)$$

$$\hat{d}^F_{y,k} = A_{1,2}\hat{d}^N_{y,k-4} + A_{2,2}\hat{d}^N_{y,k-3} \cdots + A_{5,2}\hat{d}^N_{y,k}, \quad (6)$$

$$\hat{v}^F_{x,k} = A_{1,3}\hat{v}^N_{x,k-4} + A_{2,3}\hat{v}^N_{x,k-3} \cdots + A_{5,3}\hat{v}^N_{x,k}, \quad (7)$$

$$\hat{v}^F_{y,k} = A_{1,4}\hat{v}^N_{y,k-4} + A_{2,4}\hat{v}^N_{y,k-3} \cdots + A_{5,4}\hat{v}^N_{y,k}, \quad (8)$$

where $A_{i,j}$ with $i \in [1:5], j \in [1:4]$ is the learnable parameters. Then, the filtering result $\hat{\boldsymbol{x}}^F_{1:K}$ is fed into three bidirectional LSTM layers to output $\boldsymbol{h}^L_{1:K}$, for learning the temporal information in maneuvering trajectories. At each time step k, $\boldsymbol{h}^L_k$ is passed through an all link layer (ALL) to generate a new tensor $\boldsymbol{h}_k$, which has been divided into several subsets. Each subset is mapped to the maxout output layer (MOL) with the maxout units (MU) to keep the maximal node. All the maximal nodes in $\boldsymbol{h}_k$ are combined to form $\boldsymbol{h}^M_k$. Finally, the prediction of the residual sequence $\widetilde{\boldsymbol{r}}_{1:K}$ is obtained when $\boldsymbol{h}^M_{1:K}$ passes through another ALL to restore itself to the shape of $\boldsymbol{r}_{1:K}$ with the linear activation function: $\phi_o(u) = u$.

In the tracking stage, the same pre-process step is used

Table 2: Setting of 3 trajectories

Trajectory	Prior target state x_0	The first part	The second part	The third part
1	[-18000 m, 2000 m, 150 m/s, 200 m/s]	30 s, CV model	40 s, CT model, α=3.18°	30 s, CT model, α=-6.54°
2	[-7000 m, -24000 m, 180 m/s, 220 m/s]	40 s, CT model, α=-1.08°	20 s, CV model	40 s, CT model, α=5.34°
3	[12000 m, 13000 m, 230 m/s, 190 m/s]	30 s, CV model	40 s, CT model, α=-7.16°	30 s, CT model, α=4.24°

Table 3: Means of the position tracking RMSE for all trajectories.

Trajectories		HGMM (m)	MIE-BLUE-IMM (m)	DeepMTT (m)
1	The first part	22.96	121.66	12.67
	The second part	138.29	55.24	12.98
	The third part	237.29	57.76	24.90
2	The first part	53.47	94.02	13.97
	The second part	23.67	56.45	12.12
	The third part	140.91	55.17	13.24
3	The first part	23.45	56.50	12.42
	The second part	278.06	59.89	17.63
	The third part	160.58	58.12	16.74

Table 4: Deviations of the position tracking RMSE for all trajectories.

Trajectories		HGMM (m)	MIE-BLUE-IMM (m)	DeepMTT (m)
1	The first part	2.92	13.99	2.58
	The second part	12.45	3.06(m)	2.77
	The third part	24.45	3.84	15.65
2	The first part	6.87	10.06	2.26
	The second part	3.72	4.61	2.65
	The third part	11.77	2.80	2.57
3	The first part	3.08	3.36	2.37
	The second part	31.43	3.82	12.35
	The third part	20.29	4.21	11.82

Table 5: Means of the velocity tracking RMSE for all trajectories.

Trajectories		HGMM (m/s)	MIE-BLUE-IMM (m/s)	DeepMTT (m/s)
1	The first part	12.38	159.55	3.91
	The second part	45.30	161.19	9.08
	The third part	84.74	162.34	13.88
2	The first part	18.57	178.53	4.59
	The second part	12.84	153.39	4.11
	The third part	68.83	183.68	7.85
3	The first part	13.86	190.53	8.45
	The second part	108.93	191.92	7.51
	The third part	59.96	189.75	7.71

Table 6: Deviations of the velocity tracking RMSE for all trajectories.

Trajectories		HGMM (m/s)	MIE-BLUE-IMM (m/s)	DeepMTT (m/s)
1	The first part	1.61	2.11	0.91
	The second part	1.30	1.53	1.03
	The third part	2.95	1.59	3.33
2	The first part	0.97	1.95	0.79
	The second part	1.74	2.59	1.04
	The third part	1.76	1.58	1.08
3	The first part	1.70	2.05	1.17
	The second part	2.96	1.54	2.56
	The third part	2.40	1.78	1.48

to estimate the trajectory segments. Those estimations of segments are corrected by the residuals predicted from the DeepMTT network. Finally, an entire maneuvering trajectory is estimated with the corrected segments using a reconstruction step, which connects all the segments together with overlaps according to the temporal relationship.

4. SIMULATION RESULTS

In our simulation, we followed [19] to consider the X-Y plane maneuvering tracking. The target trajectories and corresponding observations can be calculated by (1), (2), (3) and (4) of Section 2. Moreover, based on civil aircraft maneuvering parameters [3, 13], we designed 3 maneuvering-target trajectories in Table 2. Each trajectory lasts for 100 s and is sliced into three parts, in which the parameters are randomly selected according to the ranges in Table 1.

Our DeepMTT algorithm is used to track the three maneuvering-target trajectories in comparison with HGMM [5] and MIE-BLUE-IMM [20] algorithms. For each trajectory, 100 Monte Carlo simulations for tracking are run. We evaluate the performance of all three aforementioned algorithms with the means and deviations of the tracking RMSE in those 100 simulations. The results are shown in Tables 3, 4, 5 and 6. In these tables, the tracking RMSEs of the three algorithms are compared according to different parts of different trajectories in Table 2. The smallest tracking RMSE is colored sandy-brown in each row, which indicates that the corresponding algorithm performs best. Obviously, in Tables 3 and 5, all tracking RMSEs of our DeepMTT algorithm are colored sandy-brown. Hence, our DeepMTT algorithm provides the smallest tracking RMSEs for all different maneuvering trajectories in comparison with the other two algorithms. Our DeepMTT algorithm also performs well on the deviations of tracking RMSE, which presents the stability in tracking. Although for some parts of some trajectories, the tracking RMSEs of HGMM and MIE-BLUE-IMM are better, our DeepMTT algorithm retains a comparable performance. In summary, our DeepMTT algorithm outperforms the state-of-the-art HGMMM and MIE-BLUE-IMM algorithms for tracking maneuvering targets.

5. CONCLUSION

In this paper, we have proposed a novel MTT algorithm based on deep neural network. First, the LAST database of maneuvering target is built to offer sufficient samples of maneuvering trajectories. Then, a DeepMTT network is built and trained according to LAST database. Based on the DeepMTT network, our DeepMTT algorithm is proposed to estimate the maneuvering targets. The simulation results verify that in comparison with state-of-the-art MTT algorithms, our DeepMTT algorithm improves the MTT performance.

6. REFERENCES

[1] Y. D. Zhang, J. J. Zhang, M. G. Amin, and B. Himed, “Maneuvering target altitude tracking in over-the-horizon radars exploiting multipath doppler signatures,” in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 4056–4060.

[2] S. Xu, M. F. Bugallo, and P. M. Djuric, “Maneuvering target tracking with simplified cost reference particle filters,” in *Acoustics, Speech and Signal Processing (I-CASSP), 2006 IEEE International Conference on*, 2006, pp. IV–IV.

[3] X. Li and Y. Bar-Shalom, “Design of an interacting multiple model algorithm for air traffic control tracking,” *IEEE Trans. Control Syst. Technol.*, vol. 1, no. 3, pp. 186–194, 1993.

[4] X. R. Li and Y. Bar-Shalom, “Multiple-model estimation with variable structure,” *IEEE Trans. Automat.contr*, vol. 41, no. 4, pp. 478–493, 1996.

[5] L. Xu, X. R. Li, and Z. Duan, “Hybrid grid multiple-model estimation with application to maneuvering target tracking,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 52, no. 1, pp. 122–136, February 2016.

[6] L. Chin, “Application of neural networks in target tracking data fusion,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 30, no. 1, pp. 281–287, 1994.

[7] F. Amoozegar and S. H. Sadati, “Target tracking by neural network maneuver detection and input estimation,” in *Radar Conference, 1995., Record of the IEEE 1995 International*. IEEE, 1995, pp. 143–148.

[8] F. B. Duh and C. T. Lin, “Tracking a maneuvering target using neural fuzzy network,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 34, no. 1, pp. 16–33, 2004.

[9] S. S. Kim, S. Chien, C. K. An, S. D. Lee, and T. H. Lee, “Recurrent neural network,” *Kite Journal of Electronics Engineering*, vol. 5, no. 1, 1994.

[10] A. Graves and J. Schmidhuber, “Framewise phoneme classification with bidirectional lstm and other neural network architectures,” *Neural Networks*, vol. 18, no. 5-6, pp. 602–610, 2005.

[11] A. Graves *et al.*, *Supervised sequence labelling with recurrent neural networks*. Springer, 2012, vol. 385.

[12] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio, “Maxout networks,” *arXiv preprint arXiv:1302.4389*, 2013.

[13] E. Daeipour and Y. Bar-Shalom, “Imm tracking of maneuvering targets in the presence of glint,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 34, no. 3, pp. 996–1003, 1998.

[14] X. R. Li and V. P. Jilkov, “Survey of maneuvering target tracking. part i. dynamic models,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 39, no. 4, pp. 1333 – 1364, 2003.

[15] R. Weber and J. Schanne, *Airport Surveillance Radar Model 11 (ASR-11) FAA Test and Evaluation Master Plan (TEMP)*. U.s. Department of Transportation, 1998.

[16] J. Steele, “The fastest passenger jets in the sky,” https://thepointsguy.com/2017/03/fastest-passenger-jets/, accessed March 29, 2017.

[17] J. Lan and X. R. Li, “Tracking of maneuvering non-ellipsoidal extended object or target group using random matrix.” *IEEE Trans. Signal Processing*, vol. 62, no. 9, pp. 2450–2463, 2014.

[18] S. Julier, J. Uhlmann, and H. F. Durrantwhyte, “A new method for the nonlinear transformation of means and covariances in filters and estimators,” *IEEE Trans. Automat.contr*, vol. 45, no. 3, pp. 477–482, 2000.

[19] X. R. Li, Z. Zhao, and X. B. Li, “General model-set design methods for multiple-model approach,” *IEEE Trans. Automat.contr*, vol. 50, no. 9, pp. 1260–1276, 2005.

[20] H. Sheng, W. Zhao, and J. Wang, “Interacting multiple model tracking algorithm fusing input estimation and best linear unbiased estimation filter,” *Iet Radar Sonar and Navigation*, vol. 11, no. 1, pp. 70–77, 2017.