On Neural-Net Based Variable Structure Multiple Model Method : Neural-Net Design

Daebum Choi*, Monica Samal**, Byungha Ahn***
System Integration Lab. Dept. of Mechatronics, Kwang-ju Institute of Science and Technology.
1-Oryong-Dong, Puk-Gu, Kwang-ju, 500-712, South Korea
*oner@moon.kjist.ac.kr **monica@kjist.ac.kr, ***bayhay@kjist.ac.kr

ABSTRACT

In order to track a maneuvering target, multiple model (MM) methods have been researched. Almost MM algorithms have been developed based on Markov process. However, Markov based MM method is difficult to design and application-dependent. To solve this problem, Daebum Choi, et al proposed basic idea of neural-net based VSMM [5]. In this paper, we will show the design procedure of neural-net based MM and discuss how does it work in details.

Keywords: Single Maneuvering Target Tracking, Markov-Based MM, VSMM, Neural Net.

1. INTRODUCTION

Since the advent of Kalman filter that estimate the state from the contaminated measurements, various research achievements have been made in various application areas related to target tracking. However, a single Kalman filter cannot estimate exactly the state of maneuvering targets but that of slowly varying target movement [1]. Since 1970's, various researches have focused on maneuvering target Dimensional Filtering and so on. Nowadays, MM is no tracking: Input estimation, Multiple Model (MM) method, Variable w thought to be promising [2]. Various MM algorithms have been designed and can be classified into two categories based on mode variation in time: fixed MM and mode switching MM [3][4]. In fixed MM methods, each mode in previous scan cannot affect other modes except itself in current scan. In mode switching MM methods, every mode in previous scan has an influence on every mode in current scan and the relationships between previous and current modes are defined in terms of Markov process. In this letter mode switching MM is called Markov-based MM. From middle 1970's (when GPB [4] algorithms were developed), Markov-based MM methods has been focused on and intensively studied. However, sometimes Markov-based MM methods cannot estimate accurately dynamic state. To solve this problem, Daebum Choi, et al proposed basic idea of neural-net based VSMM [5]. In this paper, we will show the design procedure of neural-net based MM and discuss how does it work in details.

2. MARKOV-BASED MM

Variable Structure Multiple Model (VSMM) method is the most recent and sophisticated one in Markov-based MM methods.

Review of VSMM

The following model is a simplified VSMM. For mode set matched filtering, discrete dynamic and measurement equations are given as

$$x_{k+1}(M_{k+1}) == F_{k,k+1} x_k(M_k) + v_k(M_{k+1}) \quad (1)$$

$$z_k(M_k) = H_k(M_k) x_k(M_k) + w_k(M_k) \quad (2)$$

where x_k is a state vector, z_k is a measurement, v_k is a process noise vector, w_k is a measurement noise vector, H_k is a measurement sensitive matrix, $M_k(i)$ is the *i-th* mode set in N mode sets at scan k, and $F_{k,k+1}$ is a state transition matrix from scan k to $k+1$.

An assumption in VSMM is that the mode set transition is modeled based on the Markov process, whose transition matrix T is given by:

$$T = \{t_{i,j}\}, \text{ for } i,j = 1,\ldots,N \quad (3)$$

$$t_{i,j} = P\{M_{k+1}(j) \mid M_k(i)\} \quad (4)$$

where the predetermined $t_{i,j}$ is the probability that mode set i transfers to mode set j after one scan. Based on Markov process, the admissible mode set [3] is given by:

$$M_{k+1} = \{M \mid \exists x_{k+1}, P\{M_{k+1} \mid M_k, x_{k+1}\} > 0\} \quad (5)$$

where M is a mode set that is an element of total mode sets. Based on those two mode set jump assumptions, mode set matched estimation at scan $k+1$ is given as

$$\hat{x}_{k+1,k+1} = \sum_{j=1}^{N} P\{M_{k+1}(j)\} \hat{x}_{k+1,k+1}(M_{k+1}(j)) \quad (6)$$

where $\hat{x}$ is the overall state estimate, $P\{M\}$ is mode set probability, and $\hat{x}$ (M) is the mode set matched estimate based on mode M. In VSMM, choosing the appropriate $P\{M\}$ is the key for accurate estimation of x. Therefore the main idea of VSMM is how to calculate $P\{M\}$.

Limit of Markov-Based MM

Mode set sequence probability until scan $k+1$ in VSMM [3,4] is given by:

$$P\{M^{k+1} \mid Z^{k+1}\} = \frac{1}{c} P\{z_{k+1} \mid M^{k+1}, Z^k\} P\{M_{k+1} \mid M^k, Z^k\} P\{M^k \mid Z^k\} \quad (7)$$

where c is normalization constant, Z^k is the sequence of measurements and M^k is a sequence of the mode set until scan k.

In Eq. (7), the first term, $P\{z_{k+1} \mid M^{k+1}, Z^k\}$ is the likelihood of mode set sequence M^{k+1} given z^{k+1}, that is, the first term is the updated information form measurement. The second term, $P\{M_{k+1} \mid M^k, Z^k\}$ is the mode transition probability that is predetermined using Markov process assumption in VSMM. The meaning of second term is a priori that describes how a mode set jumps to another mode set in each scan. However, in practice, since a target does not move based on a predetermined mode jump pattern, we cannot define the second term exactly. As a result, Eq. (7) cannot be calculated exactly due to the second term. Since the mode set for state estimation is selected among admissible mode sets that satisfy Eq. (5), wrong mode set probability calculation of each mode set causes incorrect selection of the admissible mode set. However, as scans increase, the estimation mode set approaches the true mode set if the effect of the first term in Eq. (7) becomes more dominant than that of the second term. This is the main essence of the MJD phenomenon. During transient scans when the mode set for estimation approaches to the actual mode set, VSMM estimates the state based on wrong mode set. As a result, state estimation error increases during transient scans. This is the problem that springs from MJD phenomenon.

3.NEURAL NET BASED MM

State Estimate of Neural Net Based MM

To reduce MJD, we use Neural Net instead of Eq. (7). In Markov-based MM, score of each mode set is calculated using Eq. (7). Then the final state is estimated based on

Eq. (6). However, in neural net based MM, neural network gives the score to each mode set. The state estimate of neural net based VSMM is given as

$$\hat{x}_{k+1,k+1} = \sum_{j=1}^{N} f\left(I_{Update}, I_{prev}\right)\hat{x}_{k+1,k+1}\left(M_{k+1}(j)\right) \quad (8)$$

where I_{Update} is the updated information from the new-coming measurement, I_{prev} is the information that summarize the past situation, and f represents the neural net. Neural net searches the suitable output based on training process.

Design of Inputs and Outputs of Neural Net

What are the inputs and outputs of f, the neural network? Now define the moving patterns. To describe one moving pattern, three components are needed [5].

1) Previous mode set information (I_{prev}).
2) Current measurement information (I_{Update}).
3) Current target's mode set for estimation. (The desired output of f)

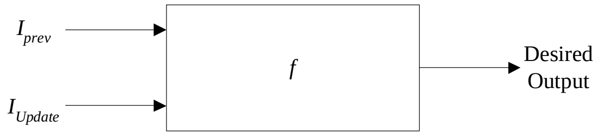

Figure 1 Block Diagram for Neural Net

I_{prev}: This value has the same format of the desired output. One difference is that I_{prev} was the desired output in the past.

I_{Update}: Measurement in each scan has information on target dynamics, but is also contaminated. Examples of this value are likelihood of measurement and distance between prediction and measurement.

The desired output of *f*: This value should imply the score for each candidate mode set. Two designing schemes are possible: (1) Parameter-based and (2) Mode-based. In parameter-based method, a mode is a specified value. For example, a mode set is defined based on process noise covariance.

The key component for neural net design is the desired output of f. In next two subsections, we show how to design the neural net based on both schemes.

Neural Net Design I: Parameter-based Method

In maneuvering periods, Kalman filter with big process noise covariance show better performance than that with small. Therefore filters with different process noise covariance can be modes in MM. This scheme is similar to AG (Adaptive Grid) [1] in VSMM.

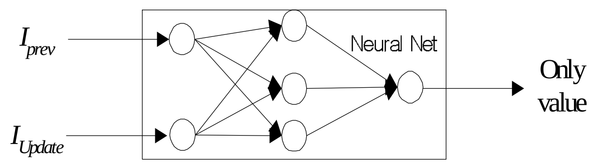

Figure 2 Parameter-Based Design

Figure 2 illustrates the basic idea of Parameter-based design. The output is the real number that represented a mode.

Neural Net Design I: Mode-based Method

In Model-based method, modes in a mode set is predetermined and fixed. In every scan neural net generate scores of all modes and select the modes based on a predefined rule like choosing k modes with maximum likelihood. This scheme is similar to AD (Active Digraph) [1] and DS (Digraph Switching) [1] in VSMM.

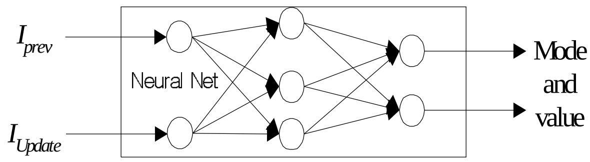

Figure 3 Mode-Based Design

Figure 3 shows the Mode-based design scheme. The output indicates specified mode or modes with its or their scores.

Training Pair Design

After design of inputs and outputs, training pair should be prepared. However, there are lots of trajectories of airborne targets, and all are not included in training pair. In this case, representative trajectories are selected for the training pair. In this section we show how neural net can handle the untrained input. Is there a pattern that cannot be interpolated? It is possible but the moving pattern is not extremely random and each object has own many but not infinite moving patterns. For example, an airplane cannot change their direction by 180° suddenly. Therefore neural network can treat the untrained input pattern adaptively.

4. SIMULATIONS AND RESULTS

In simulation, the proposed algorithm is compared to DS VSMM [1]. Simulation parameters and environments are as follows.

1) 2-D second order linear Kalman filter.
2) Measurement noise covariance of each sensor – 1 m for each coordinate.
3) Varying process noise covariance determines the mode. Mode set is constructed based on maneuvering index.
4) Range of process noise covariance: (1, 200) and spacing between neighbor modes: 5.
5) 200 Monte-Carlo runs.
6) Simulation Tool – Matlab 5.3.

Neural Net Design

1) Input- Measurement noise var. used in previous estimation, x-directional likelihood of the Kalman filter whose estimation error is smallest, and y-directional likelihood of the Kalman filter whose estimation error is smallest
2) Output- Process noise variance for current estimation (Parameter-based design)
3) One hidden layer with 16 neurons
4) For training data generation, 40 by 40 Kalman filters are running in parallel.

Training Scenarios

We predetermined two different but representative moving patterns: constant velocity linear movement and circular movement. We develop three training pairs: (1) 4 linear movement trajectories, (2) 4 circular movement trajectories and (3) 2 linear and 2 circular movement trajectories. Dynamic parameters for training scenarios are given in Table 1.

Training Scenarios	Descriptions
Scenario 1 Line Only	Initial position: (0,0) Velocities (200, 100), (-200, 100), (200, -100), (-200, -100)
Scenario 2 Circle Only	Angular velocity: $\pi/10$ Radius: 50, 100, 150, 200
Scenario 3 Line + Circle	Line - Initial position: (0,0), Velocities : (200, 100), (-200, -100), Circle - Angular velocity: $10/\pi$ Radius: 50, 100

Table 1 Descriptions of Training Scenarions

Test Scenarios

Four test scenarios are prepared. Figure 4~7 and Table 2 shows the test scenarios in detail.

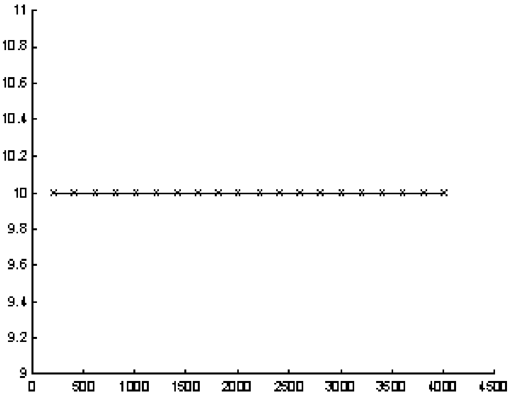

Figure 4 Test Scenario 1- Line

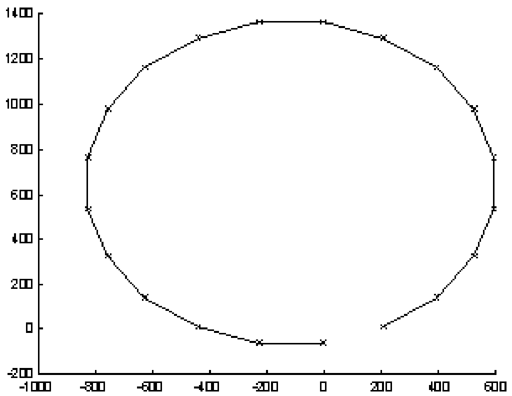

Figure 5 Test Scenario 2- Circle

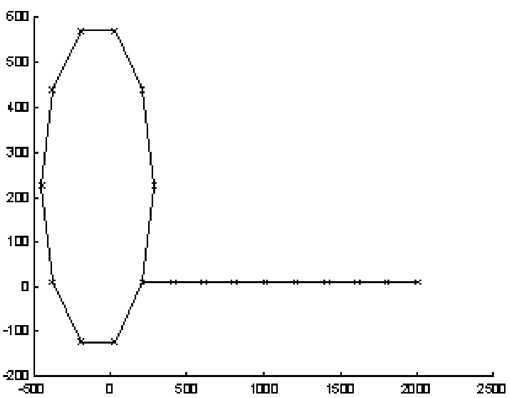

Figure 6 Test Scenario 3- Circle+Line

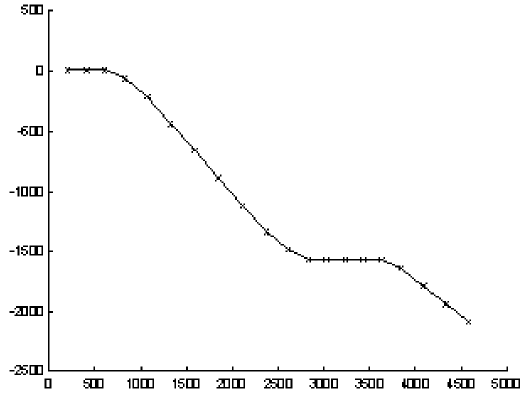

Figure 7 Test Scenario 4- Unknown

	Descriptions
Sc. 1	Initial position: (10, 10), Velocity: (200, 10)
Sc. 2	Angular velocity: π/10, Radius: 225
Sc. 3	1~10 scan (Circle): Angular velocity: π/5, Radius: 225 11~20 scan (Line): Velocity: (200, 10)
Sc. 4	Initial position: (10, 10), Initial velocity: (200, 0) 1~3 scan, 7~10 scan, 14~16 scan, 19~20 scan - No acceleration 4~6 scan – Acceleration (20, -75) 11~13 scan – Acceleration (-20, 75) 17~18 scan – Acceleration (20, -75)

Table 2 Descrition of Test Scenarios

Simulation Results

Table 3 summarizes the simulation results. NNVSMM trained with line movement (training scenario 1) show the biggest RMS error for different type (circle or line+circle) test scenario. This reflects that only simple training data is not enough for mode set selection. In circle trajectory training scenario (training scenario 2), the RMS errors are less than those of DSVSMM. In this case, circular movement is complex enough to cover all the test scenarios. However, for test scenario 4 (line+circle movement), RMS error of training scenario 2 is larger than that of training scenario3. From Figure 9 and 10, RMS error of training scenario 3 is more stable and smaller than that of training scenario 2. For unknown patterns the RMS errors of training scenario 2 and 3 are similar.

Test Sc.	Train Sc. / Filters	Sc 1 Line Only	Sc2 Circle Only	Sc 3 Line+ Circle
Sc. 1	NNVSMM	1.0301	1.1948	1.1154
	DSVSMM	1.1573	1.1469	1.1472
Sc. 2	NNVSMM	33.8047	3.1184	2.0502
	DSVSMM	5.9898	5.9647	5.9898
Sc. 3	NNVSMM	23.2437	6.0346	2.8477
	DSVSMM	5.2321	5.2337	5.2366
Sc. 4	NNVSMM	16.8466	1.5515	1.6559
	DSVSMM	2.5445	2.5178	2.5463

Table 3 Simulation Results (RMS errors)

NNVSMM using training scenario 2 and 3 shows better performances than DSVSMM. This indicates that Markov–based MM cannot cover all types of target motions and should be redesigned whenever the moving pattern is changed. Moreover, we verify that untrained input is inferred using the trained.

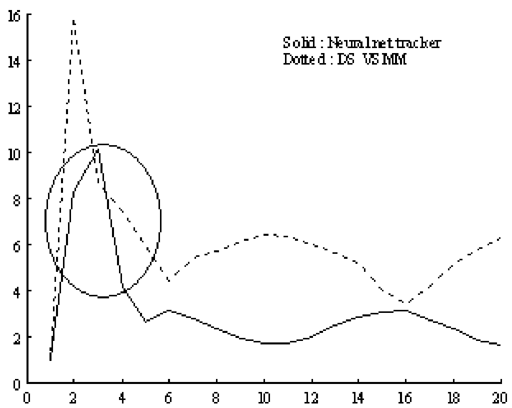

Figure 8 RMS Error - Train: Circle, Test : Circle

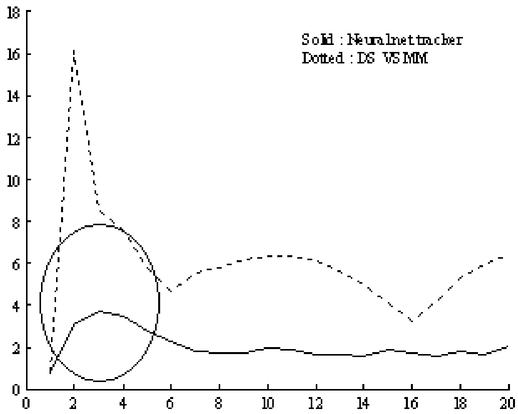

Figure 9 RMS Error - Train: Line+Circle, Test : Circle

5. CONCLUSIONS

In this paper, we propose the design procedure of neural-net based MM. For inputs and outputs of neural net two methods are proposed: Parameter-based and mode-based. We also discuss about training pair design. In simulations, we verify the proposed the training-pair-design methods.

6. REFERENCES

[1] X. Rong Li, ‘Hybrid estimation techniques’, Control and Dynamic Systems: Advances in Theory and Applications, Vol. 76, 1996, pp 2136-2287

[2] Samuel Blackman and Robert Popoli, Design and Analysis of Modern Tracking Systems, Artech House, Boston, 1999

[3] Bar-Shalom, Y. and X. Rong Li, Multitarget-Multisensor Tracking: Principles and Techniques, YBS Publishing, Connecticut, 1995

[4] Bar-Shalom, Y. and X. Rong Li, Estimation and Tracking: Principles Techniques ,and Software, Artech House, Boston, 1993

[5] Daebum Choi, Byungha Ahn, Hanseok Ko, ‘Neural Net Based Variable Structure Multiple Model Reducing Mode Set Jump Delay’, IEEE workshop on SSP, Singapore, 6~8 Aug 2001. pp. 142-145

[6] Philip D. Wasserman, Neural Computing: Theory and practice, Van Nostrand Reinhold, New York, 1989