

## 多层神经网络在跟踪式卡尔曼滤波器中的应用\*

韩明华 袁乃昌

(国防科技大学电子技术系 长沙 410073)

**摘要** 本文将多层神经网络引入跟踪式卡尔曼滤波器,提高了估计的精确度。以前的跟踪式卡尔曼滤波器的估计精度与目标的运动状态有关,当目标的运动不能够用线性状态空间模型描述时,其估计精度将要下降。而多层网络的引入,改善了这一不足。多层神经网络经过训练以后,能够对卡尔曼滤波器的结果进行修正。仿真结果表明,多层神经网络的应用,使估计精度显著提高。

**关键词** 目标跟踪,卡尔曼滤波,多层神经网络

**分类号** TN956.51

### A Improved Tracking Karlman Filter Using a Multilayed Neural network

Han Minghua Yuan Naichang

(Department of Electronic Technique, NUDT, Changsha, 410073)

**Abstract** This paper present a method to improve the estimation accuracy of a tracking Karlman filter (TKF) by using a multilayed neural network (MNN). Estimation accuracy of the TKF is degraded due to the uncertainties which cannot be expressed by the linear state-space model given priori. The MNN capable of leaning an arbitrary so that realized a mapping from measurement to the corrections of estimation of TKF. Simullation results show that the estimation accuracy is much improved by using MNN.

**Key words** target tracking, Kalman filter, multilayed neural network.

跟踪滤波器已经被广泛用于估计动态目标的位置、速度及加速度等参数。文 [1, 2] 中在线性状态空间模型的基础上提出了跟踪式卡尔曼滤波器 (KTF)。状态空间模型

\* 1997年3月22日修订  
国家重点实验室基金资助项目

的提出是基于下面的假设：目标的加速度是一阶马尔可夫过程，背景噪声为白噪声。如果实际目标的运动符合线性状态空间模型，则 TKF 估计是在均方误差最小意义下的最优化。

但是，在实际应用中，由于诸多的不确定性因素，目标的运动往往无法用线性状态空间模型来近似。首先，噪声为相关的非平稳过程。其次，由于实际目标的运动为非线性过程，而且文 [1, 2] 中提出的模型中还包括模型误差。即使目标的运动能够用线性状态空间模型来描述，在进行 TKF 之前，仍然需要对马尔可夫过程的参数进行估计，而这些参数依赖于目标的运动。所以当目标的运动不确定时，估计精度将大大降低。因此，弥补这种不确定性成为 TKF 设计的关键问题。

多层神经网络(MNN)<sup>[3]</sup>是由几层神经元构成的系统。每一个神经元与相邻的网络层中的每一个单元相连。MNN 有两个显著的优点：(1) 通过调整权值，能够实现多输入、多输出的非线性映射；(2) 由于自身的非线性使 MNN 具有很强的适应能力。比如 MNN 对于相似的输入能够输出相似的结果。由于这些优良特性，MNN 被广泛应用于非线性信息处理的诸多领域。

本文提出了一种对非确定性模型的修正方法。将 MNN 与 TKF 结合起来，通过对 MNN 的训练，实现了对 FKT 估计结果的修正。计算机的模拟结果证明了这种方法的有效性。

## 1 跟踪式滤波器

在假定目标的加速度为一阶马尔可夫过程的基础上，文 [1, 2] 中从状态空间模型推导出 TKF。下面对 TKF 进行简单的回顾。

首先定义状态向量  $X_k$

$$X_k = [X_k, X'_k, X''_k]^T \quad (1)$$

其中  $X_k$  为笛卡尔坐标系下目标的位置向量， $X'_k$  为笛卡尔坐标系下目标的速度向量， $X''_k$  为笛卡尔坐标系下目标的加速度向量， $k$  是规一化的离散时间。在 TKF 中，目标的运动由以下的状态空间模型来描述：

$$X_{k+1} = \Phi(\rho, \Delta)X_k + W_k \quad (2)$$

$$\Phi(\rho, \Delta) = \begin{bmatrix} I & \Delta I & \Delta^2/2I \\ 0 & I & \Delta I \\ 0 & 0 & \rho I \end{bmatrix} \quad (3)$$

其中  $\Delta$  为抽样周期， $\rho$  是一阶马尔可夫过程的常量参数， $0 \leq \rho \leq 1$ 。  $\rho$  的选取与估计的精度有关。 $W_k$  是未知的激励信号。

雷达只能在极坐标下测出目标的范围、方位角、高度角参量。量测方程为：

$$Y_k = h(X_k) + V_k \quad (4)$$

这里， $h(X_k)$  是从  $X_k$  到位置的变换函数， $Y_k$  是雷达测得的数据， $V_k$  是测量时带来的测量误差。假设  $W_k$  和  $V_k$  满足广义平稳条件。

TKF 对状态向量  $X_k$  的估计为：

$$\hat{X}_{k+1|k} = \Phi \hat{X}_{k|k} \quad (5)$$

$$\hat{X}_{k|k} = \hat{X}_{k|k-1} + K_k[Y_k - h(\hat{X}_{k|k-1})] \quad (6)$$

其中  $\hat{X}_{k|k}$  是基于到时间  $t$  为止的测量值对  $X_k$  的估计, 增益矩阵  $K_k$  由下式给出

$$K_k = P_k H^T [H_k P_k H_k^T + R_k]^{-1} \quad (7)$$

其中

$$P_k = E\{(X_k - \Phi \hat{X}_{k|k-1})(X_k - \Phi \hat{X}_{k|k-1})^T\} \quad (8)$$

$$H_k = \left. \frac{\partial h(x)}{\partial x} \right|_{x = \hat{X}_{k|k-1}}$$

$P_k$  可以通过迭代方法计算

$$P_{k+1} = \Phi[P_k - K_k H_k P_k] \Phi^T + Q_k \quad (9)$$

## 2 TKF 与 MNN 的结合

### 1) 滤波器的结构

如果实际目标的运动严格遵循 (1) 式描述的线性状态模型, 则  $\hat{X}_{k|k}$  的估计是在  $E\{\|X_k - \hat{X}_{k|k}\|\}$  为最小意义下的最优估计。但是, 估计值由于下面的原因而使精确度降低:

(1) 在卡尔曼滤波理论中, 假设未知的激励信号  $W_k$  为白噪声。然而在实际应用中  $W_k$  是相关的, 而且为非平稳过程。

(2) 因为绝大多数运动目标的特性为非线性, 线性状态模型 (1) 中包含了线性近似误差。即使实际运动目标完全遵循模型 (1), 马尔可夫过程参量  $\rho$  以及  $Q_k$  的选取也要依赖于目标的运动<sup>[6,7]</sup>, 这样难免带来误差。

于是我们将 MNN 引入 TKF 中, 以克服模型不确定性而带来的不足。修正后的滤波器结构如图 1 所示。 $Z^{-1}$  表示单位延迟。MNN 是由多层神经元组成的反馈网络。每一个单元与相邻层的所有单元相连。这样每个单元的输入-输出关系为

$$\begin{aligned} \text{net}_i^n &= \sum_j \omega_{ij}^{n-1} O_j^{n-1} + \theta_i^n \\ O_i^n &= f(\text{net}_i^n) \end{aligned} \quad (10)$$

其中

$O_i^n$ : 第  $n$  层第  $i$  个单元的输出;

$\text{net}_i^n$ : 第  $n$  层第  $i$  个单元的总输入;

$\theta_i^n$ : 第  $n$  层第  $i$  个单元的偏差;

$\omega_{ij}^{n-1}$ : 第  $(n-1)$  层第  $j$  个单元与第  $n$  层第  $i$  个单元之间的权值。

在隐含层我们采用“S”函数  $f(x) = \tanh(x)$ , 而在输入、输出单元中采用线性函数  $f(x) = x$ 。在图 1 中  $\square$  和  $\circ$  分别表示线性函数和“S”函数。MNN 的输出向量  $\sigma \hat{X}_k$  是对  $\hat{X}_{k|k}$  的自动修正。修正后的估计值  $\hat{X}_k^*$  为

$$\hat{X}_k^* = \hat{X}_k + \sigma \hat{X}_{k|k} \quad (11)$$

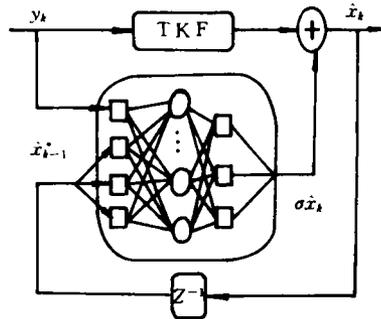


图 1 改进滤波器的结构

在状态估计理论中,要对时刻  $k$  做出估计必须知道  $(k-1)$  时刻的估计值及当前的测量值。所以, MNN 的输入向量为测量值  $Y_k$  和前一步估计值  $\hat{X}_{k-1}$  的集合。为了保证 TKF 的稳定性,在实际的应用中, TKF 与 MNN 是分别独立运行的。

## 2) MNN 的训练方法

在这一部分中,我们将给出 MNN 的训练方法。首先,为训练过程定义费用函数:

$$J_{TKF} = E\{\|X_k - \hat{X}_{k|k}\|^2\} \rightarrow \text{minimum} \quad (12)$$

其中  $\|\cdot\|$  表示欧几里得范数。在训练过程中,让费用函数达到最小:

$$\begin{aligned} J_k &= \frac{1}{2} \|X_k - \hat{X}_k\|^2 \\ &= \frac{1}{2} \|(X_k - \hat{X}_{k|k}) - \sigma X_k\|^2 \end{aligned} \quad (13)$$

在训练 MNN 时,训练信号必须是已知的,为典型目标运动的真实轨迹。本文考虑两种训练方法:

方法 1: 用单目标的运动轨迹进行训练; 方法 2: 用  $N$  个目标的运动轨迹进行训练。

在模型不确定性较小的情况下,我们采用方法 1,用常用的 BP 算法<sup>[3]</sup>训练 MNN。当目标的运动比较复杂,模型的不确定性较强而无法用线性空间模型来近似时,我们采用方法 2 来对 MNN 进行训练。用多目标的运动轨迹训练,能够增强 MNN 的适应性。

令  $Y_k^n$ 、 $X_k^n$  和  $\hat{X}_k^{n*}$  分别是在时刻  $k$  的雷达测量信号、实际状态向量和修正估计向量的第  $n$  个集合。新的滤波器可视为由一个 MNN 和  $N$  个 TKF 组成。令 TKFn 表示第  $n$  个 TKF。当  $Y_k^n$  和  $\hat{X}_k^n$  已知时,我们通过 TKFn 来计算  $\hat{X}_k^{n*}$ ,训练 MNN 直到费用函数达到最小:

$$J_n = \sum_{k=1}^N J_k^n$$

其中

$$J_k^n = \frac{1}{2} \|x_k^n - \hat{X}_k^{n*}\|^2 \quad (14)$$

在时刻  $k$ ,对连接权值的修改从  $n=1$  到  $N$  依次重复。令  $\omega_k^n$  和  $\Delta\omega_k^n$  分别为连接权值和修正量。当在时刻  $k$  用第  $n$  个状态向量集进行训练时,  $\Delta\omega_k^n$  由下式给出:

$$\Delta\omega_k^n = \begin{cases} -\eta \frac{\partial J_k}{\partial \omega} \Big|_{\omega} = \omega_k^{n-1} = -\eta \frac{\partial J_k^n}{\partial \omega} \Big|_{\omega} = \omega_k^{n-1}, 2 \leq n \leq N \\ -\eta \frac{\partial J_k}{\partial \omega} \Big|_{\omega} = \omega_k^{N-1} = -\eta \frac{\partial J_k^n}{\partial \omega} \Big|_{\omega} = \omega_k^{N-1}, n = 1 \end{cases} \quad (15)$$

## 3 仿真结果

我们通过计算机仿真来说明所提出方法的有效性:考虑一维跟踪问题,其中

$$\begin{aligned} \Phi(\rho, \Delta) &= \begin{bmatrix} 1 & \Delta & 0.5\Delta \\ 0 & 1 & \Delta \\ 0 & 0 & \rho \end{bmatrix}, R_k = \sigma_k^2 \\ h(X_k) &= H \quad X_k = [1 \quad 0 \quad 0]X_k \end{aligned} \quad (16)$$

注意到  $h$  为非线性传输算子, 可以通过加入非线性的 MNN 进行修正, 则经过训练的 MNN 能够利用角度的测量值求解复杂的跟踪问题。

在卡尔曼滤波理论中, 假设激励信号为白噪声, 我们必须选择方差  $Q_k$  的值保证 (1) 式描述的线性空间模型能够精确的描述实际目标的运动。在一维跟踪问题中,  $Q_k$  选取的最佳值为

$$Q_k = -2\sigma_M^2 \log \rho \begin{bmatrix} \Delta^4/20 & \Delta^3/8 & \Delta^2/6 \\ \Delta^3/8 & \Delta^2/3 & \Delta/2 \\ \Delta^2/6 & \Delta/2 & 1 \end{bmatrix} \quad (17)$$

其中  $\sigma_M^2$  为加速度的方差<sup>[1]</sup>。TKF 的初始化由下式给出<sup>[1,2]</sup>:

$$X_0 = \begin{bmatrix} Y_0 \\ (Y_0 - Y_{-1})/\Delta \\ 0 \end{bmatrix}, \quad (18)$$

$$P_0 = \begin{bmatrix} \sigma_R^2 & \sigma_R^2/\Delta & 0 \\ \sigma_R^2/\Delta & 2\sigma_R^2/2 + \sigma_M^2\Delta^2/4 & \sigma_R^2/2\Delta \\ 0 & \sigma_R^2/2\Delta & \sigma_M^2 \end{bmatrix} \quad (20)$$

在仿真中, 目标以恒定的速率和恒定加速度  $a$  运动。目标的运动轨迹如图 2 所示。我们假设  $V$  是已知的,  $a$  是未知的。实际的状态向量  $X_k$  可由下式给出:

$$X_k = \begin{bmatrix} X_k \\ X'_k \\ X''_k \end{bmatrix} = \begin{bmatrix} \frac{V^2}{a} \sin \frac{Ka\Delta}{V} \\ V \cos \frac{Ka\Delta}{V} \\ -a \sin \frac{Ka\Delta}{V} \end{bmatrix} \quad (20)$$

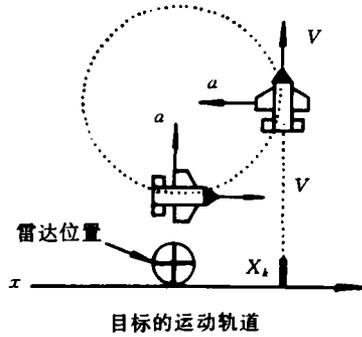


图 2 目标的运动轨迹

初始值为  $X_0 = [0 \ V \ 0]^T$ 。显然上面目标的运动不能够用 (1) 中的线性空间模型来描述。为了

尽可能精确的描述实际目标的运动, 参数  $\rho$  和  $\sigma_M^2$  必须事先估计, 估计值由下式给出:

$$\rho = \cos \frac{a\Delta}{V} \quad \sigma_M^2 = \frac{1}{2}a^2 \quad (21)$$

因为我们已经假设  $V$  是已知的常数而  $a$  是未知的。所以我们在运行 TKF 之前必须对  $a$  进行估计。我们定义横向加速度的估计值为  $a_k$ , 用加速度为  $a_k$  的目标的运动估计对 MNN 进行训练。

从  $k=1$  到  $k=k_s$ , 我们在  $a=a_k$  时运行 TKF 而不用 MNN, 这是因为当初始的估计值误差较大时, MNN 的收敛速度较慢。从  $k=k_s+1$  到  $k=k_E$ , 我们从初始状态  $X_{k_s}^* = X_{k_s|k_s}$  开始对 MNN 进行训练。

在本实例中, 令  $V=400$  (m/s),  $\Delta=10$  (ms),  $\sigma_R^2=400$  (m<sup>2</sup>),  $k_s=3000$ ,  $k_E=103000$ 。我们选择 3 层神经网络, 其中隐含层为 30 个单元。由于任意的非线性映射三层神经网络已经具备任意的精确度<sup>[4]</sup>, 因此本文采用的是三层神经网络。在估计过程中 TKF 占主要

作用而MNN只是产生较小的修正,因此MNN的连接权值不宜太大。初始权值可以随机产生,在 $\{-10^{-6}, 10^{-6}\}$ 之间,训练比率 $\eta$ 设置为 $\eta=10^{-5}$ 。除此之外,我们再在“S”函数上加一个固定的常数偏差0.001,这将加速训练的收敛速度。

训练完毕后,我们固定MNN的连接权值,对加速度为 $a_T$ 的目标的状态分别用TKF和带有MNN的TKF进行估计,执行20000次。利用均方误差(MSE)来评价估计的精度:

$$MSE = \frac{1}{20000} \sum_{k=1}^{20000} (X_k - \hat{X}_k)^T (X_k - \hat{X}_k) \quad (22)$$

首先,我们先看一下用方法1训练的带有MNN的TKF的性能。图3显示了当 $a_T = a_L$ 时的MSE。条件 $a_T = a_L$ 暗示了当训练MNN时为实际的横向加速度。在TKF中我们令 $a_k = 40, 50, 60 (m/s^2)$ ,而运行MNN时 $a_T = a_L = 40, 50, 60 (m/s^2)$ 。由图3可以看出,经过MNN修正的估计结果要明显优于只有TKF的情况。

然而,实际上,我们无法知道实际的加速度 $a^T$ 。下面看一下估计精度和 $(a_T - a_L)$  ( $a_k$ 固定为 $a_k = 50 (m/s^2)$ )的关系。仿真结果如图4所示。从仿真结果可以看出,带有MNN的TKF的估计精度要比单独的TKF高得多。同时我们看到估计的精度随 $(a_T - a_L)$ 的差值的增大而降低。这表明,仅仅经过方法1训练的MNN不能在一个大的范围内对模型的不确定性进行补偿。

下面比较一下方法1和方法2的结果。固定 $a_k = 50 (m/s^2)$ ,通过3个 $\{a_1^i, a_2^i, a_3^i\} = \{40, 50, 60\} (m/s^2)$ 轨迹来训练MNN。MNN的节点数和连接权值的设置同上, $k_s = 3000, k_E = 103000, \eta = 1.5 \times 10^{-6}$ 。仿真结果如图5所示。结果表明经过方法2训练的MNN能够在较大的范围内对模型的不确定性进行修正,估计精度明显优于方法1训练

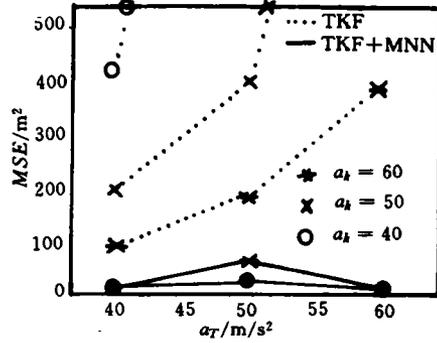


图3  $a_T = a_L$ 时MSE比较

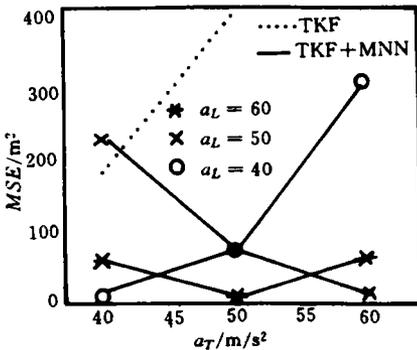


图4  $a_k = 50 (m/s^2)$  MSE比较

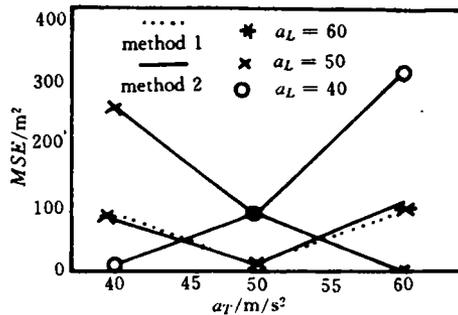


图5 方法1和方法2的比较  $a_k = 50 (m/s^2)$

练的MNN。

## 4 结论

本文中提出了用MNN对TKF进行修正的方法,提高了估计的精度,弥补了单纯的TKF估计中的不足。本文中仅限于对目标跟踪方面的讨论,经过推广,本方法可以广泛地应用于其它的领域内。同时,由于该方法需要大量样本的训练以及目前计算能力的限制,在实时性要求不高时可以很好的满足。但这一不足可以通过并行分布式计算的方式来解决,作者正在开展与此相关的算法。

## 参 考 文 献

- 1 R. A Singer. Estimating optimal tracking filter performance for maned maneuvering targets. IEEE Trans. Aerosp. Electron. Syst 1970 AES-6 (4), 473~483
- 2 R. A Singer and K. W. Behnke. Real-time tracking filter evaluation and selection for practical application. IEEE Trans. Aerosp. Electron. Syst 1971 AES-7 (1): 100~110
- 3 D. E Rumelhart, G. e. Hinton and R. J Williams. Learning internal representation by error propagation. In parallel Distributed Processing 1986 Chapter 8: 319~362
- 4 K. Funahashi. On the approximate realization of continuous mapping by neural networks. Neural Networks 1989 (3): 184~192
- 5 R. L. Moose and H. F. Valandingham. Modeling and estimation for tracking targets. IEEE Trans. Aerosp. Electron. Syst, 1979 AES-15 (3): 448~456
- 6 T. Ohshima and T. Nakamizo. A bearing-only adaptive tracking filter, (in Japanese. Trans, Soc, instr, Contr, Eng, 1987, 23 (8), 821~827
- 7 J. R. Cloutier, J. H. Evers and J. J. Feeley. Assessment of air-to-to air missile guidance and control technology. IEEE Contr, Sys, 1989 Mag 9 (6): 27~34 (October)
- 8 P. Haffner, A. Waibel, H. Sawal and K. Shilkano. Fast back-propagation learning for large phonemic neural networks. In Eurospeech 89, European Conf, Speech Commun. Tech 1989: 553~556

(责任编辑 潘 生)