

神经过程和运动模型混合驱动的 机动目标跟踪算法

朱洪峰, 熊伟, 崔亚奇, 王子玲

(海军航空大学信息融合研究所, 264001, 山东烟台)

摘要: 针对基于贝叶斯滤波器模型的目标跟踪算法非常依赖先验知识, 在复杂场景中跟踪性能下降的问题, 提出了一种神经过程和运动模型混合驱动的机动目标跟踪算法。利用神经网络构建一个目标运动的随机微分方程来提高对目标运动的建模能力; 使用加速度模型作为领域知识约束目标状态的微分方程, 构造一个混合驱动的解码器; 利用所构成的混合驱动解码器替换神经过程的解码模块, 形成一种无监督学习的混合驱动滤波器。仿真实验结果表明: 所提算法同时具有数据驱动和模型驱动算法的优势, 依赖先验知识少, 能在不同场景下保持稳定的跟踪精度, 生成的轨迹较神经过程滤波器更加平滑且具有目标动力学特征; 与经典的贝叶斯滤波器相比, 所提算法的状态估计峰值误差平均降低了 20%。

关键词: 机动目标跟踪; 神经过程; 混合驱动; 无监督学习; 状态估计

中图分类号: TN953 **文献标志码:** A

DOI: 10.7652/xjtuxb202304016 **文章编号:** 0253-987X(2023)04-0152-10

Hybrid-Driven Maneuvering Target Tracking Algorithm Based on Neural Process

ZHU Hongfeng, XIONG Wei, CUI Yaqi, WANG Ziling

(Institute of Information Fusion, Naval Aviation University, Yantai, Shandong 264001, China)

Abstract: Aiming at the problem that the target tracking algorithm based on Bayesian filter model relies heavily on prior knowledge and the tracking performance degrades in complex scenarios, a hybrid-driven maneuvering target tracking algorithm based on neural process and motion model was proposed. Firstly, a stochastic differential equation of target motion was constructed by using neural network to improve the modeling ability of target motion. Secondly, the acceleration model was used as the differential equation of the domain knowledge constraint target state to construct a hybrid-driven decoder. Finally, the decoding module of the neural process was replaced by the hybrid-driven decoder to form an unsupervised learning hybrid-driven filter. The simulation results show that the proposed algorithm has the advantages of both data-driven and model-driven algorithms. It relies less on prior knowledge and can maintain stable tracking accuracy in a variety of scenarios. The generated trajectory is smoother and has more target dynamic characteristics than the neural process filter. Compared with the classical Bayesian filter, the state estimation peak error of the proposed algorithm is reduced by 20% on average.

Keywords: maneuvering target tracking; neural process; hybrid-driven; unsupervised learning; state estimation

目标跟踪技术在军用和民用两个领域有着广泛的应用^[1-2]。如今应用较为广泛的是 Kalman 滤波^[3]、交互式多模型(interactive multiple-model, IMM)^[4]以及在这两种框架上发展、改进而来的算法。但是,基于模型的经典算法非常依赖先验知识。随着科学技术的发展,目标的运动能力愈发强大,目标跟踪应用的环境也愈发复杂,经典目标跟踪算法越来越难以满足应用需求。近十年,深度学习技术得到了飞速发展,在许多领域有了广泛的研究和应用^[5]。深度学习作为一种数据驱动方法,能够从大数据中学习到显式模型难以表达的潜在规律和趋势。因此,许多学者将其应用到目标跟踪中,利用神经网络学习目标和环境的知识来提高经典算法的效能,形成数据与模型混合驱动的目标跟踪算法。随着计算机技术的不断发展以及未来智能化的需求,混合驱动成为了目标跟踪技术的重要的发展趋势之一^[6]。

目前,混合驱动的目标跟踪相关研究可以总结为对经典模型驱动算法的运动模型参数^[7]、中间参数^[8]以及估计输出^[9]这3个方面进行估计和修正,使得经典跟踪算法的跟踪性能有了一定的提升。但是,上述的同类文献中的神经网络几乎采用的都是有监督学习的方式,需要优化目标的真值或者标签,如运动模型和真实轨迹。数据集是深度学习应用面临的阻碍之一^[10],在实际的目标跟踪工程应用中,目标运动的成本和时机、电磁环境以及任务限制等客观因素限制了采集到的数据集规模和质量^[11],而且许多复杂场景难以获取理想的标签或者真值,这导致有监督学习的方式难以真正实施和部署。无监督学习的方式通常不需要优化目标的真值,对数据集的要求低。随着目标跟踪应用的环境越发复杂,研究探索无监督学习的混合驱动目标跟踪算法对于工程实用有着重要意义。

现有的无监督学习的目标跟踪算法大多采用时间序列处理领域的变分自编码器(variational auto-encoder, VAE)^[12]作为基本网络结构,之后结合其他网络进行发展而来。文献[13]使用循环神经网络和 VAE 构建了一个贝叶斯滤波器进行时间序列预测;文献[14]使用了 VAE 来进行自动驾驶中的轨迹重建和机动分类;文献[15]将神经网络建模随机过程,并利用 VAE 的架构对参数求解,提出了神经

过程等。这些方法能够通过无监督的方式学习时间序列的建模和生成,但是解码器直接使用神经网络输出序列,直接用于目标轨迹生成时,会导致输出的目标轨迹受量测影响严重,平滑性和动力学特征差,算法可解释性差,也没有充分利用已知的领域知识。因此,如何将领域知识与 VAE 充分结合,实现更加符合实际的时间序列处理器,也是当下一个热点研究问题^[16]。

为了进一步提高无监督学习的目标跟踪算法的跟踪效果,本文对神经过程的解码器使用了混合驱动的方式进行重构,提出了一种混合驱动的神经过程滤波器(hybrid-driven neural process filter, HD-NPF)。HD-NPF 在解码器部分使用神经网络作为数据驱动,估计每个滑动时间窗口内的目标运动的随机微分方程的初值以及机动参数,求解微分方程可得到目标任意时刻的状态估计,从而实现了混合驱动。目标的动力学模型引导和约束了目标轨迹生成过程,从而使得生成的轨迹更加平滑且具有动力学特征,也降低了神经网络的规模,提高了其学习效率。仿真结果表明,HD-NPF 依赖先验知识少,环境自适应能力较之经典目标跟踪算法更强。

1 问题描述

1.1 目标运动的随机微分方程

本文主要考虑目标跟踪中目标的状态估计(滤波)问题^[17]。在实际场景中,目标运动在时间上是连续的,加上环境中的扰动,目标运动可通过 Langevin 随机微分方程进行建模^[18]

$$\frac{dx}{dt} = \tilde{a}(x(t), t) + D(x(t), t) \frac{d\beta(t)}{dt} \quad (1)$$

式中: $\tilde{a}(x(t), t) \in \mathbb{R}^n$ 表示目标的动力学模型,表示了目标在无噪声和其他干扰条件下的真实状态变化趋势, t 为连续时间变量; $d\beta(t)$ 表示噪声或者干扰; $D(x(t), t) \in \mathbb{R}^{n \times n}$ 是扩散矩阵,作用是将 $d\beta(t)$ 通过某种方式作用于目标状态或者运动过程。

在经典的模型驱动目标跟踪算法中, $\tilde{a}(x(t), t)$ 通常被建模为一个或者一组固定的运动模型,如 IMM 通常采用匀速直线(constant velocity, CV)、匀加速(constant acceleration, CA)和协同转弯模型(coordinate turn, CT)^[19]的组合作为数据集。此外,为了方便计算, $d\beta(t)$ 通常被建模为高斯白噪声。

但是,目标往往是机动的。在与环境的交互中,目标机动具有随机、突发以及不可预测的特点,此时 $\tilde{\mathbf{a}}(\mathbf{x}(t), t)$ 是时变的,在缺乏目标机动信息时,模型驱动算法的跟踪精度就会下降甚至失跟。并且,在复杂的环境中, $\tilde{\mathbf{a}}(\mathbf{x}(t), t)$ 可能难以进行显式的数学建模。

对于 $d\boldsymbol{\beta}(t)$,从广义上来看,其不仅包含了环境噪声,还包含了当目标遇到未知情况进行的人工调整、规避以及目标跟踪算法机动判断延迟(算法需要累积一定量测才能判断出目标机动)带来的误差等,这些使得 $d\boldsymbol{\beta}(t)$ 以及 $\mathbf{D}(\mathbf{x}(t), t)$ 的数学形式非常复杂,甚至难以用数学模型明确表达。

因此,在复杂的机动场景中,难以获得准确先验知识,难以精确设置式(1)的各部分,导致跟踪受到影响。

1.2 神经过程网络

神经过程(neural process, NP)^[15]是一种将随机过程和神经网络相结合的模型,其本质是使用神经网络来参数化随机过程的映射函数,然后利用推导 VAE 的方式推导出 NP 的参数学习方式。NP 的主要原理如下。

首先,定义一个随机过程函数 $F: \mathcal{T} \rightarrow \mathcal{S}$,对于任意一个有限序列,表示为向量 $\mathbf{t}_{1:n} = (t_1, t_2, \dots, t_n)$, $t_i \in \mathcal{T}, i=1, 2, \dots, n$ (i 为序列索引),定义 ρ 为 $(F(t_1), F(t_2), \dots, F(t_n))$ 的边缘分布,则有

$$\rho_{1:n}(\mathbf{s}_{1:n}) = \rho_{t_1, \dots, t_n}(\mathbf{s}_1, \dots, \mathbf{s}_n) \quad (2)$$

上式满足文献[15]中的一致性和交换性时,对于 F 的一个实例化 f ,根据式(2)有

$$\rho_{1:n}(\mathbf{s}_{1:n}) = \int p(f) p(\mathbf{s}_{1:n} | f, \mathbf{t}_{1:n}) df \quad (3)$$

实际中的样本 $\mathbf{y}$ 是存在量测噪声的,类似高斯过程,NP 将量测噪声建模为 $\mathbf{S}_i \sim \mathcal{N}(F(t_i), \boldsymbol{\sigma}^2)$,即

$$p(\mathbf{s}_{1:n} | f, \mathbf{t}_{1:n}) = \prod_{i=1}^n \mathcal{N}(s_i | F(t_i), \boldsymbol{\sigma}^2) \quad (4)$$

将式(4)代入式(3),得到

$$\rho_{1:n}(\mathbf{s}_{1:n}) = \int p(f) \prod_{i=1}^n \mathcal{N}(s_i | F(t_i), \boldsymbol{\sigma}^2) df \quad (5)$$

然后,NP 采用神经网络 g 重新建模 $F(t_i)$,高维向量 $\mathbf{h}$ 作为输入提供序列信息,即 $F(t_i) = g(t_i, \mathbf{h})$,式(5)可重新表示为

$$p(\mathbf{h}, \mathbf{s}_{1:n} | \mathbf{t}_{1:n}) = p(\mathbf{h}) \prod_{i=1}^n \mathcal{N}(s_i | g(t_i, \mathbf{h}), \boldsymbol{\sigma}^2) \quad (6)$$

结合 VAE 的推导过程^[12],引入 $\mathbf{h}$ 的变分先验

分布 $q(\mathbf{h} | \mathbf{s}_{1:n}, \mathbf{t}_{1:n})$,则似然函数 $p(\mathbf{s}_{1:n} | \mathbf{t}_{1:n})$ 的证据下限(evidence lower-bound, ELBO)表示为

$$L_{p(\mathbf{s}_{1:n} | \mathbf{t}_{1:n})} = -\text{KL}(q(\mathbf{h} | \mathbf{s}_{1:n}, \mathbf{t}_{1:n}) \| p(\mathbf{h})) + \sum_{i=1}^n E_{q(\mathbf{h} | \mathbf{s}_{1:n}, \mathbf{t}_{1:n})} [\log p(s_i | \mathbf{h}, t_i)] \quad (7)$$

式中 KL 代表求取 KL 散度的函数。NP 为了便于处理预测问题和使网络更好地学习到正确的模型,其将数据集划分为上下文信息 $\mathbf{t}_{1:m}, \mathbf{s}_{1:m}$ 和目标信息 $\mathbf{t}_{m+1:n}, \mathbf{s}_{m+1:n}$,然后使用上下文信息 $q(\mathbf{h} | \mathbf{s}_{1:m}, \mathbf{t}_{1:m})$ 作为先验信息替代 $q(\mathbf{h} | \mathbf{s}_{1:n}, \mathbf{t}_{1:n})$,则预测分布 $p(\mathbf{s}_{m+1:n} | \mathbf{t}_{1:m}, \mathbf{s}_{1:m})$ 的 ELBO 为

$$L_{p(\mathbf{s}_{m+1:n} | \mathbf{t}_{1:m}, \mathbf{s}_{1:m})} = -\text{KL}(q(\mathbf{h} | \mathbf{s}_{1:n}, \mathbf{t}_{1:n}) \| q(\mathbf{h} | \mathbf{s}_{1:m}, \mathbf{t}_{1:m})) + \sum_{i=m+1}^n E_{q(\mathbf{h} | \mathbf{s}_{1:n}, \mathbf{t}_{1:n})} [\log p(s_i | \mathbf{h}, t_i)] \quad (8)$$

NP 通过最小化式(8)来完成序列的无监督学习。NP 通过神经网络参数化随机过程,灵活性强,并且根据上下文信息预测目标点的方式将部分训练转移到测试时间,提高了效率。本文采用 NP 作为无监督学习的网络框架,对目标的轨迹进行学习和估计。

2 基于神经过程的混合驱动滤波器

2.1 混合驱动解码器

1.2 小节分析了 NP 的基本原理,在应用中 $\mathbf{x}$ 为时间或者控制量, $\mathbf{y}$ 为时间序列。但是,由 NP 解码器的生成过程(式(6))可以看到,生成的 $\mathbf{y}_i$ 是相互独立的,如果将 $\mathbf{y}$ 作为目标轨迹序列,这显然是不符合实际的。目标状态的转移通常以一阶 Markov 链建模,为了符合目标跟踪的变量设定,这里强调: $\mathbf{x}$ 和 $\mathbf{z}$ 分别表示目标量测和状态向量。令 $\mathcal{M}$ 表示目标的运动模式或者对应的参数,可以得到联合分布

$$p(\mathbf{z}_{1:n}, \mathbf{x}_{1:n}, \mathbf{h}, \mathcal{M}) = p(\mathbf{h}) p(\mathcal{M}_0) p(\mathbf{x}_0) \cdot$$

$$\prod_{i=1}^n p(\mathbf{z}_i | \mathbf{x}_i) p(\mathbf{x}_i | \mathbf{x}_{i-1}, \mathcal{M}_i, \mathbf{h}) p(\mathcal{M}_i | \mathcal{M}_{i-1}, \mathbf{h}) \quad (9)$$

可以看出,目标的状态之间存在状态转移 $p(\mathbf{x}_i | \mathbf{x}_{i-1}, \mathcal{M}_i, \mathbf{h})$ 和运动模型转移 $p(\mathcal{M}_i | \mathcal{M}_{i-1}, \mathbf{h})$ 的关系。1.1 小节已经讨论过,目标的机动通常是未知的。但是,目标并不是时刻都在机动,也就是在某个长度 l 的滑动时间窗口内,目标的运动模型可以视为几乎不变,即 $p(\mathcal{M}_i | \mathcal{M}_{i-1}, \mathbf{h}) \approx 1$,即使在窗口中存在机动,也可以使用一个折中模型 $\overline{\mathcal{M}}$ 简化问题。因此,本文在滑动时间窗口内讨论目标轨迹估计问题。

神经网络是一种数据驱动的方法,具有强大的非线性拟合和建模能力^[20]。在复杂的机动场景中,目标运动和环境难以建模,采用神经网络参数化式(1)得到神经微分方程^[21]是较好的选择。但是,完全使用神经网络估计目标轨迹并没有利用好领域知识(如目标遵从一些基本的运动规律),可解释性差,且因为一般将网络的维度设置较大而导致计算消耗大。因此,本小节提出了混合驱动解码器,来替代 NP 的解码器。

虽然目标的机动未知,但是目标运动遵从基本的模型。由于机动几乎都可以描述为加速度变化^[22],所以采用加速度模型作为模型驱动,令目标状态向量为 $\mathbf{x}_t = [x_t, \dot{x}_t, y_t, \dot{y}_t]^T$ 。在实际的跟踪中,传感器获得的量测通常是离散的,因此本文采用 k 指代 t 中的离散时刻点(或量测时间)。令 NP 编码器编码第 k 个滑动窗口内的目标量测信息,然后采样得到高维向量 $\hat{\mathbf{h}}_{C/T}^k$ (下标 C 和 T 分别表示上下文和目标)。在解码阶段,利用多层感知机 MLP_{ini} 根据 $\hat{\mathbf{h}}_{C/T}^k$ 解码出当前窗口内目标初始(即 $k-l+1$ 时刻)状态 $\hat{\mathbf{x}}_{k-l+1}$ 和机动参数(加速度) $\hat{\mathbf{x}}_k$

$$[\hat{\mathbf{x}}_{k-l+1} \quad \hat{\mathbf{x}}_k] = \text{MLP}_{\text{ini}}(\hat{\mathbf{h}}_{C/T}^k) \quad (10)$$

根据加速度模型定义,窗口内任意连续时间 t 有 $\tilde{\mathbf{a}}(\mathbf{x}(t), t) = [\dot{x}_t, \ddot{x}_t, \dot{y}_t, \ddot{y}_t]^T$, 并使用神经网络 MLP_a 建模扰动 $\mathbf{D}(\mathbf{x}(t), t) d\boldsymbol{\zeta}(t)$

$$d\boldsymbol{\zeta}_t = \text{MLP}_a(\hat{\mathbf{h}}_{C/T}^k, \hat{\mathbf{x}}_t, t) \quad (11)$$

结合式(1)可以得到混合驱动的解码器微分方程

$$\frac{d\mathbf{x}(t)}{dt} = [\dot{x}_t, \hat{\mathbf{x}}_k(1) + d\boldsymbol{\zeta}_t(1), \dot{y}_t, \hat{\mathbf{x}}_k(2) + d\boldsymbol{\zeta}_t(2)]^T \quad (12)$$

正如对式(9)的分析,本文认为在一个滑动窗口

内目标的机动模式 $\overline{\mathcal{M}}$ 没有发生突变,因此在式(12)中,目标的加速度是在机动参数 $\hat{\mathbf{x}}_k$ 的基础上进行波动。使用神经微分方程求解器,结合式(10)~(12)便可以估计出时间窗内任意时刻目标状态

$$\hat{\mathbf{x}}_k = \text{SDESolver}(\hat{\mathbf{h}}_{C/T}^k, t_0, t_1, \boldsymbol{\theta}), \quad t, t_0, t_1 \in [k-l+1, k+1] \quad (13)$$

式中 SDESolver 为随机微分方程(stochastic differential equation, SDE)的求解器。

式(12)根据目标加速度模型,直接给定了目标状态与其微分的形式,这给神经网络的梯度优化过程提供了方向,提高了解码器的学习效率的同时减少了神经网络的规模,提高了算法的可解释性。

2.2 HD-NPF 框架结构

将 2.1 小节提出的混合驱动解码器与 NP 的解码器相结合,得到了基于神经过程的混合驱动滤波器(HD-NPF),其网络结构如图 1 所示。

图 1 中左边为 NP 根据目标量测和时间戳 $(\mathbf{Z}_{C/T}, \mathbf{T}_{C/T})$ 进行编码的过程,首先使用神经网络 MLP_e 编码得到高维向量 $\mathbf{r}_{C/T}^k$

$$\{\mathbf{r}_{C/T}^k\}_{i=1}^n = \text{MLP}_e(\mathbf{Z}_{C/T}^k, \mathbf{T}_{C/T}^k) \quad (14)$$

然后,使用平均函数对 $\mathbf{r}_{C/T}^k$ 聚合(图 4 中的 a 模块),目的是将不同长度的输入进行聚合,使得计算量与量测数目的关系为线性。通过 MLP_p 计算输出隐层向量 $\mathbf{h}_{C/T}^k$ 分布的一、二阶矩 $\boldsymbol{\mu}_{h_{C/T}^k}^k, \boldsymbol{\Sigma}_{h_{C/T}^k}^k$, 并使用高维高斯分布建模其概率分布,重采样得到 $\hat{\mathbf{h}}_{C/T}^k$

$$\begin{cases} [\boldsymbol{\mu}_{h_{C/T}^k}^k, \boldsymbol{\Sigma}_{h_{C/T}^k}^k] = \text{MLP}_p(\text{Mean}(\{\mathbf{r}_{C/T}^k\}_{i=1}^n)) \\ \hat{\mathbf{h}}_{C/T}^k \leftarrow q(\mathbf{h} | \mathbf{Z}_n, \mathbf{T}_n) = \mathcal{N}(\mathbf{h} | \boldsymbol{\mu}_{h_{C/T}^k}^k, \boldsymbol{\Sigma}_{h_{C/T}^k}^k) \end{cases} \quad (15)$$

式中的 $\hat{\mathbf{h}}_{C/T}^k$ 包含了初值、运动模型等目标的运动信

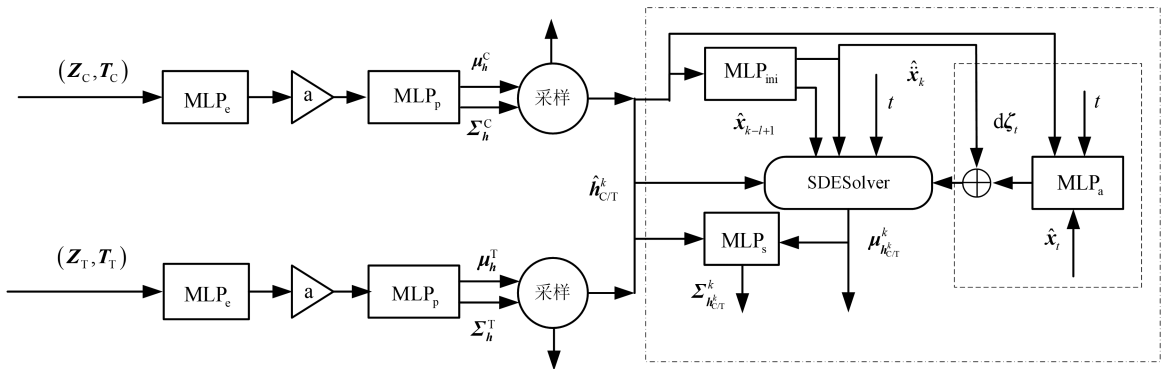


图 1 混合驱动神经过程滤波器结构

Fig. 1 Structure diagram of the hybrid-driven neural process filter

息。式(9)可简化为

$$p(\mathbf{z}_{1:n}, \mathbf{x}_{1:n}, \mathbf{h}) = p(\mathbf{h}) \prod_{i=1}^n p(\mathbf{z}_i | \mathbf{x}_i) p(\mathbf{x}_i | \mathbf{x}_{i-1}, \mathbf{h}) \quad (16)$$

根据 2.1 小节提出的混合驱动动力学解码器, $p(\mathbf{x}_i | \mathbf{x}_{i-1}, \mathbf{h})$ 的一阶矩 $\hat{\boldsymbol{\mu}}_x^k$ 可通过式(10)~(13)计算得到。为便于计算 ELBO, 使用网络 MLP_s 根据高维隐藏信息和目标状态均值估计目标状态的二阶矩 $\hat{\boldsymbol{\Sigma}}_x^k$

$$\hat{\boldsymbol{\Sigma}}_x^k = \text{MLP}_s(\hat{\mathbf{h}}_{\text{C/T}}^k, \hat{\boldsymbol{\mu}}_x^k) \quad (17)$$

令目标状态为高斯分布(即使不服从高斯分布, 神经网络会对其补偿), 同时令量测矩阵为

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}^T \quad (18)$$

根据高斯分布的性质, 得到似然函数

$$p(\hat{\mathbf{z}}_t^k | \hat{\mathbf{h}}_{\text{C/T}}^k) = p(\hat{\mathbf{z}}_t^k | \hat{\mathbf{X}}_t^k) \mathcal{N}(\hat{\mathbf{X}}_t^k | \hat{\boldsymbol{\mu}}_x^k, \hat{\boldsymbol{\Sigma}}_x^k) = \mathcal{N}(\hat{\mathbf{z}}_t^k | \mathbf{H}\hat{\boldsymbol{\mu}}_x^k, \mathbf{H}\hat{\boldsymbol{\Sigma}}_x^k\mathbf{H}^T) \quad (19)$$

式中 t^o 表示 HDF-NP 输出的目标状态估计的时刻点序列向量。

对比 1.2 小节和本小节 HD-NPF 和 NP 的原理可以发现, 两者的本质区别在式(6)和式(16)。即, HD-NPF 在解码器部分由于动力学约束 $p(\mathbf{x}_i | \mathbf{x}_{i-1}, \mathbf{h})$ 导致其可能不再满足 NP 的一致性和交换性, 不能直接利用 NP 的 ELBO 计算方式和结论。因此, 需要证明 HD-NPF 也是 NP。

利用文献[23]中的引理, 结合本文, 可以得到: 在第 k 个滑动窗口内, HD-NPF 的解码器输出函数 $\text{SDSolver}(\hat{\mathbf{h}}_{\text{C/T}}^k, t_0, t_1, \boldsymbol{\theta})$ 在给定的初值 $[\hat{\mathbf{x}}_{k-l+1} \quad \hat{\mathbf{x}}_k]$ 条件下, 可以被视为一个依赖时间 t 的函数 $\mathcal{F}_{\boldsymbol{\theta}}(t)$ 。

证明 根据 2.1 小节的论述, 本文假设在长度为 l 的滑动时间窗口内的目标运动模式 $\overline{\mathcal{M}}$ 是不变的, 因此在给定窗口目标状态和机动参数初值 $[\hat{\mathbf{x}}_{k-l+1} \quad \hat{\mathbf{x}}_k]$ 的条件下, 式(13)的输出由时间 t 确定, 即为关于时间的函数 $\mathcal{F}_{\boldsymbol{\theta}}(t)$, $[\hat{\mathbf{x}}_{k-l+1} \quad \hat{\mathbf{x}}_k]$ 由神经网络计算给出。

另外, $p(\mathbf{z}_i | \mathbf{x}_i)$ 依赖于解码器输出 $\hat{\mathbf{x}}_i^k$, 将其并入 $\mathcal{F}_{\boldsymbol{\theta}}(t)$ 中, 此时式(16)可表示为

$$p((\mathbf{z}_{1:n}, \mathbf{x}_{1:n}), \mathbf{h}) = p(\mathcal{F}_{\boldsymbol{\theta}}(t)) \cdot \prod_{i=1}^n p((\mathbf{z}, \mathbf{x})_i | \mathcal{F}_{\boldsymbol{\theta}}(t)) \quad (20)$$

此时, 式(20)与式(6)具有同样的形式, 因此 HD-NPF 也为一种 NP, 可以直接使用 NP 的结论。结合式(8)和式(16), 令在当前滑动窗口中算法生成的轨迹点数为 N_g , 可以得到 HDF-NP 的 ELBO 为

$$\mathcal{L}_k = -\text{KL}(q(\mathbf{h}_{\text{T}}^k | \mathbf{Z}_{\text{T}}^k, \mathbf{T}_{\text{T}}^k) \parallel q(\mathbf{h}_{\text{C}}^k, \mathbf{T}_{\text{C}}^k)) +$$

$$\sum_{j=1}^{N_g} E_{q(\mathbf{h}_{\text{T}}^k | \mathbf{Z}_{\text{T}}^k, \mathbf{T}_{\text{T}}^k)} [\log p(\mathbf{z}_i^k | \mathbf{h}_{\text{T}}^k, \mathbf{T}_{\text{T}}^k)] \quad (21)$$

对比式(21)和式(8)可以发现, HDF-NP 的 ELBO 或者优化目标较之 NP 没有变化, 但是其内部解码器生成过程具有了动力学特征, 目标状态相互之间具有 Markov 性, 算法的可解释性提高。而且, 相较 NP, 本文提出的 HDF-NP 在动力学约束下, 能够生成状态的二阶估计, 即速度、状态估计的维度得以增加。

3 HD-NPF 训练

为便于分析 HD-NPF 性能, 本文构造了一个二维的飞行目标数据集作为训练集。令目标在二维直角坐标系平面进行运动, 其运动由 CV、CA 以及 CT 随机组合, 形成机动轨迹。每条轨迹的起始和结束模型为 CV, 中间进行了两次随机为 CA 或者 CT 的机动。数据集其他参数见表 1。

表 1 数据集参数

Table 1 Parameters of dataset

参数	数值	参数	数值
轨迹长度/s	80	采样频率/Hz	2
最大转弯率/((°) · s ⁻¹) ± 15		最大速度/(m · s ⁻¹) ± 400	
最小转弯率/((°) · s ⁻¹) ± 4		最大转弯角度/(°) ± 270	

目标的初始位置和速度在满足表 1 情况下随机生成。令 $\text{rand}(a, b)$ 表示 a 和 b 之间的均匀分布。CV 运动持续时间为 $\text{rand}(10, 30)$ s; CA 运动持续时间为 $\text{rand}(10, 20)$ s(首要满足加速后不超过最大速度)。

根据表 1 随机生成 100 条轨迹, 每条轨迹随机抽选 20 条长度为 10 s(上下文信息为其中 2~7 s 的片段)的轨迹段, 则训练集的规模为 2 000 条。每个训练轮对数据集添加标准差为 50 m 的零均值高斯白噪声, 使得网络更好地学习目标轨迹的统计特征。

对于 HD-NPF 中的网络设置, 遵循尽可能简单的原则, 网络规模增大不仅会增加计算时间, 也会增加过拟合的风险。通过一定的测试, 本文设置 $\hat{\mathbf{r}}_{\text{C/T}}^k$ 、 $\hat{\mathbf{h}}_{\text{C/T}}^k$ 的特征维度分别为 64、16, HD-NPF 中的 5 个多层感知机 MLP_e 、 MLP_p 、 MLP_{ini} 、 MLP_s 、 MLP_a 的隐层维度(输入、输出维度由其在算法结构中的位置决定)分别为(8, 2)、(64, 16)、(6)、(16)以及(16), 激活函数均采用 Relu。训练批 batch 为 20, 采用式(21)在每批数据误差的均值作为损失, 即

$$\overline{\mathcal{L}}_k = \frac{1}{I_{\text{batch}}} \sum_{i=1}^{I_{\text{batch}}} \overline{\mathcal{L}}_k^i \quad (22)$$

式(15)的采样会使得式(22)计算的 ELBO 梯度难以求解。在 VAE 中,通常采用重参数化技巧将随机性与数据解耦,使得 ELBO 的梯度计算得以进行^[24]。HD-NPF 同样采用了该技巧,则隐藏向量 $\hat{\mathbf{h}}_{C/T}^k$ 第 i 次采样过程为

$$\hat{\mathbf{h}}_{C/T}^k(i) = \boldsymbol{\mu}_{C/T}^k + \boldsymbol{\varepsilon}^i \boldsymbol{\Sigma}_{C/T}^k, \boldsymbol{\varepsilon} \sim \mathcal{N}(0, \mathbf{I}) \quad (23)$$

目标量测数据随轨迹起始点变化范围很大,为便于神经网络学习,需要对其进行归一化处理。根据目标轨迹特点和表 1,本文将目标量测轨迹段归一化到 $[-1, 1]$

$$\mathbf{Z}_{t_k}^{\text{normal}} = \frac{(\mathbf{Z}_{t_k} - \text{mean}(\mathbf{Z}_{t_k}))f_s}{0.5(l + l_p) |v_{\max}|} \quad (24)$$

式中: $t_k = [k-l+1, k-l+2, \dots, k]$ 为第 t 个窗口内量测对应的时间戳向量; f_s 为传感器的采样频率; $v_{\max}$ 为预设的目标最大速度; l_p 为需要预测的时刻点数目,目标跟踪中通常关注滤波值和一步预测,因此 $l_p=1$ 。

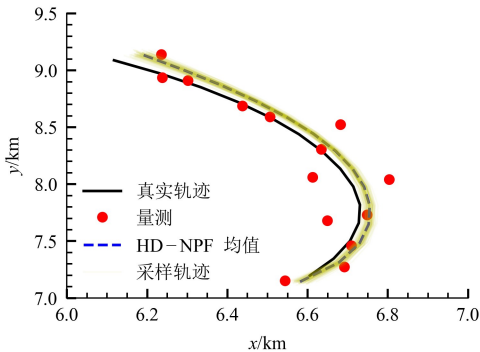
HD-NPF 初始学习率为 0.01,采用余弦退火法进行学习率衰减,在学习 500 轮后衰减到 0。经过训练,最终 $\bar{\mathcal{L}}_k$ 收敛在 -69.32。

4 仿真实验与分析

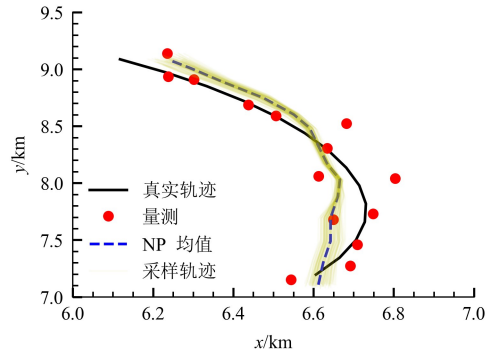
4.1 HD-NPF 和 NP 对比

首先进行 HD-NPF 的消融实验,验证分析 HD-NPF 中混合驱动解码模块的效果。NP 和 HD-NPF 采用同样的训练集轨迹长度进行训练。按照第 3 节所述随机生成一条轨迹段,分别使用 HDF-NP 和 NP 生成目标轨迹,结果如图 2 所示。

对比图 2 中 HD-NPF 和 NP 生成的轨迹可以发现,由于 HD-NPF 的解码器为混合驱动的动力学解码模块,使得其生成的轨迹较 NP 明显平滑了许多,具有了目标运动的特征,而 NP 学习的结果是朝着量测靠近,导致其估计的轨迹受数据影响大。由此可知,混合驱动的动力学模块正确发挥了预期作用。



(a) HD-NPF 生成轨迹



(b) NP 生成轨迹

图 2 HD-NPF 和 NP 估计轨迹对比

Fig. 2 Comparison of HD-NPF and NP estimated trajectories

4.2 机动目标跟踪实验

接下来进行 HD-NPF 的机动目标跟踪实验,进一步分析其性能。需要说明的是,HD-NPF 直接使用式(15)的隐藏状态的均值 $\boldsymbol{\mu}_{C/T}^k$ 到解码器生成目标轨迹,避免多次采样过程。HD-NPF 与两个模型驱动的算法(当前统计(current statistic, CS)模型^[25]和 IMM)和一个数据驱动的算法(最小平方滤波器(least square filter, LSF)^[26])进行了比较。LSF 采用与 HD-NPF 一样的滑动窗口设置。考虑到本文仿真估计的是固定间隔时刻的目标状态,HD-NPF 的积分/微分求解器采用较为简单的 Euler^[21],步长为 0.25 s。

由于本文考虑的是复杂机动场景且先验知识难以获知下的机动目标跟踪问题,因此对比算法的参数设置较为粗糙。LSF 算法采用 3 阶多项式来对目标每个维度的量测进行拟合。CS 模型的机动频率设置为 $\alpha=1/20$,最大加速度幅值为 $|a_{\pm\max}|=60 \text{ m/s}^2$ 。CS 和 IMM 的量测方差设置为 $\sigma_m=2500 \text{ m}^2$ 。IMM 采用了 3 个 Kalman 滤波器,运动模型和参数设置如下:过程噪声 $\sigma_p=0.01^2 \text{ m}^2$ 的 CV, $\sigma_p=0.01^2 \text{ m}^2$ 、 $\sigma_p=16 \text{ m}^2$ 的 CA,转移概率矩阵为

$$\mathbf{H} = \begin{bmatrix} 0.9 & 0.05 & 0.05 \\ 0.05 & 0.9 & 0.05 \\ 0.05 & 0.05 & 0.9 \end{bmatrix} \quad (25)$$

为使仿真结果更有说服力,采用两个与第 3 节所述的训练集不同构的场景进行跟踪实验,每条轨迹进行 200 次蒙特卡罗跟踪实验,噪声标准差为 50 m。使用均方根误差(root mean squared error, RMSE)作为评价指标,令 $\mathbf{X}_{i,k}$ 、 $\hat{\mathbf{X}}_{i,k}$ 分别为第 i 次实验目标的第 k 个位置点, n 为蒙特卡罗实验次数,则 RMSE 定义为

$$e_{\text{RMSE}} = \sqrt{\frac{1}{n} \sum_{i=1}^n (\mathbf{X}_{i,k} - \hat{\mathbf{X}}_{i,k})^T (\mathbf{X}_{i,k} - \hat{\mathbf{X}}_{i,k})}$$

(26)

然后,进行机动目标跟踪实验。对于 HD-NPF,当目标刚起始跟踪即量测数目小于 l 时,无法进行跟踪。这个阶段选择经典算法进行初始跟踪。

场景 1 在这个场景中,目标状态初值 $\mathbf{x}_0 = [7\ 000\ \text{m}, 200\ \text{m/s}, 6\ 000\ \text{m}, 220\ \text{m/s}]^T$ 。开始时直行 15 s,然后采取蛇形机动直到 80 s,在 81~90 s 时直行,16~80 s 阶段进行 5 次 CA 模型机动,其加速度相对于时间的变化情况见表 2。

表 2 场景 1 中 CA 模型参数序列

Table 2 Parameter sequence of CA model in scenario 1

时间/s	加速度/($\text{m} \cdot \text{s}^{-2}$)
16~35	[8, -20]
36~50	[-18, 36]
51~65	[14, -30]
66~80	[-16, 20]
81~90	[8, -12]

进行 200 次蒙特卡罗仿真实验,得到场景 1 中各算法跟踪的位置和速度 RMSE 对比,如图 3 所示。

计算场景 1 中各算法在整个轨迹上的位置和速度的平均均方根误差和峰值均方根误差,结果如表 3 所示。需要说明的是,上述计算从轨迹的第 12 s 开始算起,以减少算法初始化对计算指标的影响。此外,将 4 个算法 200 次仿真的运行时间也列于表 3 中。

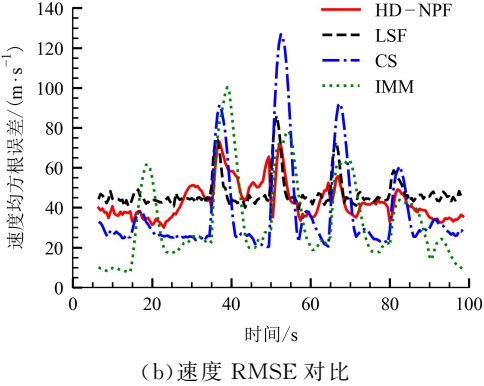
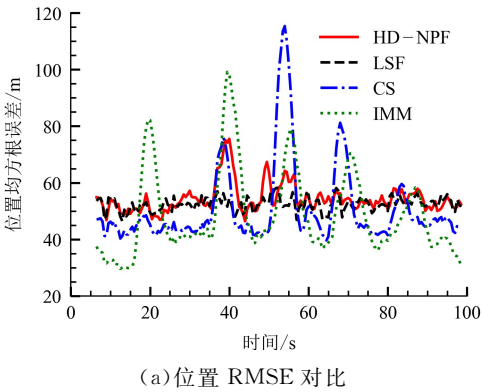


图 3 场景 1 中跟踪 RMSE 对比
Fig. 3 Tracking RMSE comparison in scenario 1

从图 3 和表 3 可以看出:本文提出的 HD-NPF 算法由于使用了混合驱动,同时具有了数据驱动和模型驱动算法的特征。HD-NPF 由于利用目标历史信息,位置和速度 RMSE 的曲线较模型驱动算法(CS、IMM)波动更小,峰值误差更低,较 CS 和 IMM 降低了约 20%,跟踪更为平稳、鲁棒性更强。较数据驱动算法(LSF),HD-NPF 由于加入了领域知识,使其在图 3(a)中能够对目标运动模型的变化发生产生一定反馈,在图 3(b)中在目标稳定运动期间,如 10~35 s,速度的估计明显优于 LSF 算法,可解释性更强,符合表 3 的数据。

从 200 次仿真中随机抽选一次实验,将各算法的轨迹进行对比,如图 4(a)所示。为方便观察,局部放大其机动部分,如图 4(b)所示。可以看出,由于 HD-NPF 解码器为混合驱动的动力学模块,其估计的轨迹平滑性较好。

但是,HD-NPF 是无监督学习算法,学习效率较低。同时,场景 1 的机动为加速度机动,较为符合 LSF、CS 和 IMM 的先验知识设置,因此在位置跟踪精度上,HDF-NP 较其他 3 种算法较差。

从表 3 中算法的运行时间来看,虽然 HD-NPF 使用了神经网络,但是增加的计算成本并不多,较 IMM,200 次仿真仅增加了 0.3 s 左右,效率较高。

表 3 场景 1 跟踪精度对比

Table 3 Tracking accuracy comparison in scenario 1

算法	位置平均均方根	位置峰值均方根	速度平均均方根	速度峰值均方根	运行时间/s
	误差/m	误差/m	误差/($\text{m} \cdot \text{s}^{-1}$)	误差/($\text{m} \cdot \text{s}^{-1}$)	
HDF-NP	54.288 2	75.713 1	42.855 0	72.488 3	1.028
LSF	52.696 9	60.883 7	47.375 6	87.271 0	0.056
CS	50.899 8	113.568 0	38.396 8	127.131 0	0.121
IMM	49.197 7	98.123 1	34.619 6	99.557 5	0.761

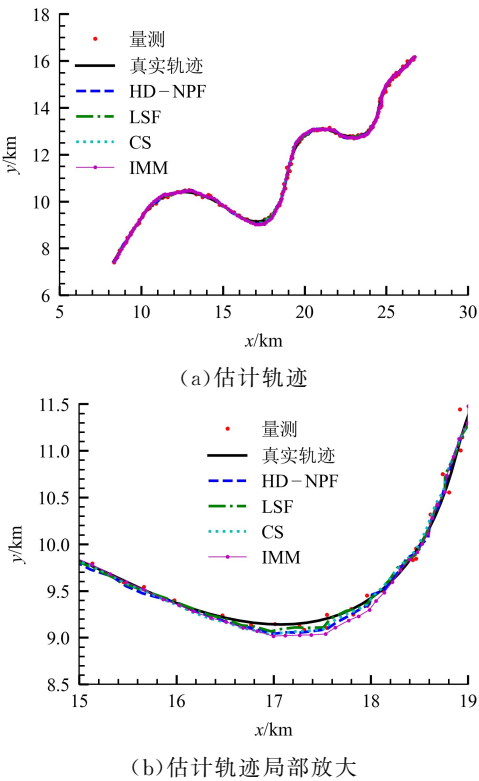


图 4 场景 1 估计的轨迹对比

Fig. 4 Comparison of estimated trajectories for scenario 1

场景 2 场景 1 的轨迹虽然与训练集不同构,但是其速度等关键参数范围大致在表 1 范围内。在实际场景中,可能在跟踪区域出现异常目标,其机动性明显超出训练集参数范围,此时重新训练算法是不现实的。对于本文提出的 HD-NPF,面对上述的异常目标,其策略为通过调整式(24)的轨迹段归一化参数 $v_{\max}$, 对异常目标的量测数据进行适当缩放,使得处理后的轨迹参数范围大致符合训练集,从而尽可能保证跟踪效果。

因此,场景 2 设置了一个异常目标的跟踪场景,其特点在于高速和高机动性。令目标初始状态向量为 $\mathbf{x}_0 = [4\,000\,\text{m}, 300\,\text{m/s}, 9\,000\,\text{m}, 600\,\text{m/s}]^T$, 目标在 1~15 s 期间进行匀速直线运动,然后以 $-20(^{\circ})/\text{s}$ 的转弯率右转弯 15 s, 31~40 s 期间以 $[-30, 10]^T\,\text{m/s}^2$ 的加速度机动,然后再以 $15(^{\circ})/\text{s}$ 的转弯率持续 10 s 的左转弯,之后再以 $[-40, 15]^T\,\text{m/s}^2$ 的加速度机动 10 s,最后 10 s 恢复匀速直线运动。另外,假设目标所在电磁环境较差,设置量测噪声标准差为 100 m。

此时,HD-NPF 在初始跟踪阶段判断其可能为异常目标后,对式(24)调整,本文设置本场景 $|v_{\max}| = 1\,000\,\text{m/s}$ 。此外,目标的高机动容易导致经典算法发散,将 CS 的加速度幅值调整为 $|a_{\pm\max}| = 80\,\text{m/s}^2$, IMM 算法 CA 模型的过程噪声从 $\sigma_p = 0.01^2\,\text{m}^2$ 调

整为 $\sigma_p = 1\,\text{m}^2$ 。进行 200 次蒙特卡罗仿真实验,得到仿真结果如图 5、图 6 及表 4 所示。

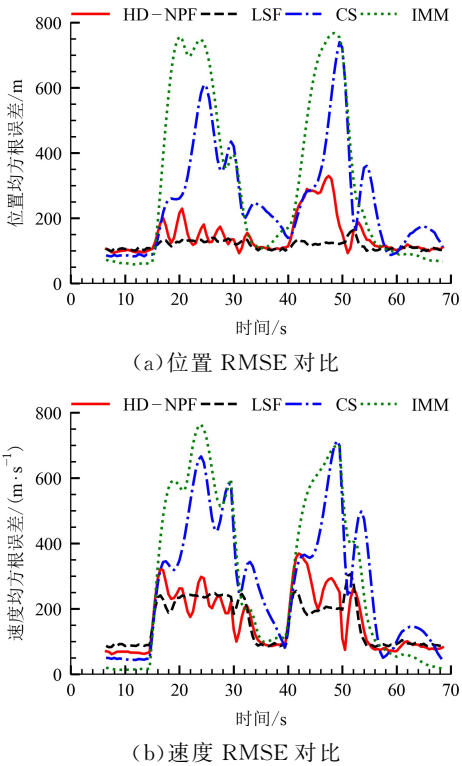


图 5 场景 2 中跟踪 RMSE 对比

Fig. 5 Tracking RMSE comparison in scenario 2

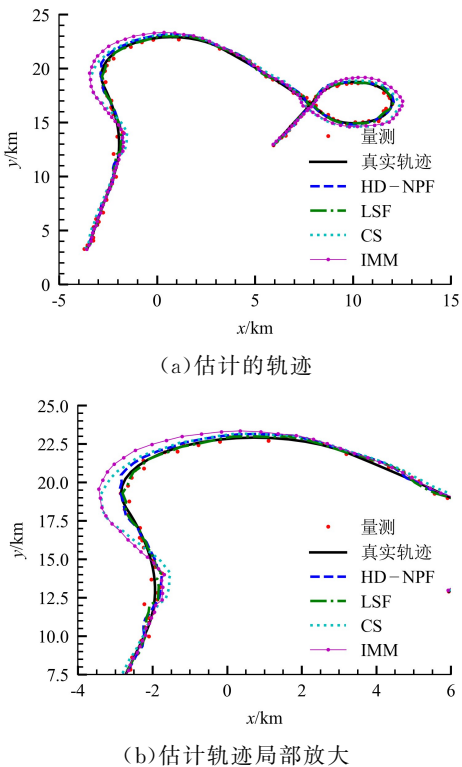


图 6 场景 2 估计的轨迹对比

Fig. 6 Comparison of estimated trajectories for scenario 2

表 4 场景 2 跟踪精度对比

Table 4 Tracking accuracy comparison in scenario 2

算法	位置平均	位置峰值	速度平均	速度峰值
	均方根	均方根	均方根误差/	均方根误差/
	误差/m	误差/m	($\text{m} \cdot \text{s}^{-1}$)	($\text{m} \cdot \text{s}^{-1}$)
HDF-NP	143.431	338.509	160.681	371.938
LSF	115.859	152.888	155.762	298.784
CS	252.975	732.794	277.853	709.117
IMM	304.531	770.584	293.927	758.737

在场景 2 中,由于是训练集之外的异常目标, IMM 和 CS 的参数只是略微修改以尽量避免发散,并没有足够的先验知识为其设置较为精确的模型参数。加上目标高机动、高速度的影响, IMM 和 CS 的跟踪效果误差很大(见图 5 和表 4),估计轨迹在机动部分已经偏离真实轨迹严重(见图 6)。本文提出的 HD-NPF 可以利用结构上的特点,通过归一化部分缩放的方式,尽可能适应异常情况。图 5 和表 3 表明, HD-NPF 仍然能够保持场景 1 中的部分优势,较 IMM 和 CS 具有更加稳定的 RMSE 和更高的跟踪精度。在实际应用中,对异常目标关注度更高,比如防碰撞等,此时 HD-NPF 的优势更加明显。

从图 5 和表 4 可以看出, LSF 的跟踪表现较 HD-NPF 更好,从图 5 可以观察到两者跟踪精度差异主要体现在 15~30 s 和 40~50 s 期间的转弯机动。这是由于目标的高速度使得转弯半径较大,导致 HD-NPF 易识别成为接近直线的运动,进而跟踪精度降低。因此,实际中应该尽可能使得训练集能够覆盖目标可能的运动模型和情景。

结合两个场景的仿真结果可以看出,虽然模型驱动方式跟踪精度较高,但是依赖精确的先验知识,一旦先验知识设置不准确,其跟踪精度急速下降。数据驱动的方法依赖先验知识少,且环境适应能力强,在各个场景均保持几乎一致的跟踪效果,但可解释性差,难以辨别目标的模型变化。本文提出的 HDF-NP 结合了两点特点,依赖先验知识少,环境适应能力强,且具有一定的可解释性,即对目标模型具有一定识别能力,且采用的是无监督学习的方式,对数据集要求低,更加易于实际应用。

在实际场景中,电磁干扰和通信等因素会使得量测可能出现乱序、缺失等现象, LSF 虽然在上述实验中跟踪表现一般,但是较经典跟踪算法,其在处理这类连续时间跟踪时具有优势^[26-27]。式(13)~(17)以及图 1 的结果表明, HD-NPF 可以接受无序的量测作

为输入,输出任意连续时刻的状态估计。因此,下一步工作将探索 HD-NPF 作为连续时间函数处理量测乱序、缺失等更加符合实际情况的目标跟踪问题。

5 结 论

本文基于神经过程框架提出了一种无监督学习的混合驱动机动目标跟踪算法。通过将目标运动的加速度模型和神经网络参数化的目标运动微分方程相结合,构造出了混合驱动的动力学解码模块,替换了 NP 的解码部分,使得 HD-NPFP 生成的轨迹更加平滑和具有动力学特征。通过仿真实验可以得到以下结论。

(1) HD-NPF 由于使用了混合驱动解码模块,估计的轨迹较 NP 在平滑性和动力学特征上明显得到增强。HD-NPF 同时具有了模型驱动和数据驱动算法的优势:环境自适应能力强,跟踪稳定,需要先验知识少,且能对目标的运动模型具有一定识别能力。

(2) HD-NPF 采用的是无监督学习方法,对数据集要求低,在实际情况中更加易于部署。下一步工作将探索 HD-NPF 在乱序、缺失和不等时间间隔量测等复杂场景中的目标跟踪问题。

参考文献:

[1] KUMAR M, MONDAL S. Recent developments on target tracking problems: a review [J]. Ocean Engineering, 2021, 236: 109558.

[2] 彭华甫, 黄高明, 田威, 等. 标签多伯努利机动目标跟踪与分类算法 [J]. 西安交通大学学报, 2019, 53(2): 157-162, 178.

PENG Huafu, HUANG Gaoming, TIAN Wei, et al. A tracking and classification algorithm for maneuvering targets with labeled multi-Bernoulli [J]. Journal of Xi'an Jiaotong University, 2019, 53(2): 157-162, 178.

[3] KALMAN R E. A new approach to linear filtering and prediction problems [J]. Journal of Basic Engineering, 1960, 82(1): 35-45.

[4] LI X R, JILKOV V P. Survey of maneuvering target tracking: part V multiple-model methods [J]. IEEE Transactions on Aerospace and Electronic Systems, 2005, 41(4): 1255-1321.

[5] DONG Shi, WANG Ping, ABBAS K. A survey on deep learning and its applications [J]. Computer Science Review, 2021, 40: 100379.

[6] JIN Xuebo, ROBERT JEREMIAH R J, SU Tingli, et al. The new trend of state estimation: from model-driven to hybrid-driven methods [J]. Sensors, 2021,

- 21(6): 2085.
- [7] DENG Lichuan, LI Da, LI Ruifang. Improved IMM algorithm based on RNNs [C]//Journal of Physics: Conference Series. Bristol, UK: IOP Publishing, 2020: 012055.
- [8] CHEN Changhao, LU C X, WANG Bing, et al. DynaNet: neural Kalman dynamical model for motion estimation and prediction [J]. IEEE Transactions on Neural Networks and Learning Systems, 2021, 32(12): 5479-5491.
- [9] LIU Jingxian, WANG Zulin, XU Mai. DeepMTT: a deep learning maneuvering target-tracking algorithm based on bidirectional LSTM network [J]. Information Fusion, 2020, 53: 289-304.
- [10] PALEYES A, URMA R G, LAWRENCE N D. Challenges in deploying machine learning: a survey of case studies [J]. ACM Computing Surveys, 2022, 55(6): 114.
- [11] 段续庭, 周宇康, 田大新, 等. 深度学习军事领域应用综述 [C]//2022 世界交通运输大会(WTC2022)论文集(交通工程与航空运输篇). 北京, 中国: 人民交通出版社股份有限公司, 2022: 417-425.
- [12] KINGMA D P, WELING M. An introduction to variational autoencoders [J]. Foundations and Trends® in Machine Learning, 2019, 12(4): 307-392.
- [13] LIM B, ZOHREN S, ROBERTS S. Recurrent neural filters: learning independent Bayesian filtering steps for time series prediction [C]//2020 International Joint Conference on Neural Networks (IJCNN). Piscataway, NJ, USA: IEEE, 2020: 1-8.
- [14] RAKOS O, BECSI T, ARADI S. Adversarial autoencoder for trajectory generation and maneuver classification [C]//2021 IEEE 25th International Conference on Intelligent Engineering Systems (INES). Piscataway, NJ, USA: IEEE, 2021: 000013-000018.
- [15] GARNELO M, SCHWARZ J, ROSENBAUM D, et al. Neural processes [EB/OL]. (2018-07-04)[2022-08-01]. <https://arxiv.org/abs/1807.01622>.
- [16] GIRIN L, LEGLAIVE S, BIE Xiaoyu, et al. Dynamical variational autoencoders: a comprehensive review [J]. Foundations and Trends® in Machine Learning, 2021, 15(1/2): 1-175.
- [17] 汪家宝, 陈树新, 吴昊, 等. 采用平方根高斯核积分滤波的目标跟踪算法 [J]. 西安交通大学学报, 2022, 56(4): 157-164.
- WANG Jiabao, CHEN Shuxin, WU Hao, et al. Target tracking algorithm using square-root Gaussian kernel quadrature filter [J]. Journal of Xi'an Jiaotong University, 2022, 56(4): 157-164.
- [18] CROUSE D. Basic tracking using nonlinear continuous-time dynamic models [J]. IEEE Aerospace and Electronic Systems Magazine, 2015, 30(2): 4-41.
- [19] 何友, 修建娟, 关欣, 等. 雷达数据处理及应用 [M]. 3 版. 北京: 电子工业出版社, 2013: 36-40.
- [20] 赵崇文. 人工神经网络综述 [J]. 山西电子技术, 2020(3): 94-96.
- ZHAO Chongwen. A survey on artificial neural networks [J]. Shanxi Electronic Technology, 2020(3): 94-96.
- [21] CHEN R T Q, RUBANOVA Y, BETTENCOURT J, et al. Neural ordinary differential equations [C]//Proceedings of the 32nd International Conference on Neural Information Processing Systems. Red Hook, NY, USA: Curran Associates Inc., 2018: 12.
- [22] XU Linfeng, LI Xrong, DUAN Zhansheng. Hybrid grid multiple-model estimation with application to maneuvering target tracking [J]. IEEE Transactions on Aerospace and Electronic Systems, 2016, 52(1): 122-136.
- [23] NORCLIFFE A, BODNAR C, DAY B, et al. Neural ODE processes [EB/OL]. (2021-08-17)[2022-08-01]. <https://arxiv.org/abs/2103.12413v2>.
- [24] JIN Xuebo, GONG Wentao, KONG Jianlei, et al. PF-VAE: a planar flow-based variational auto-encoder prediction model for time series data [J]. Mathematics, 2022, 10(4): 610.
- [25] 周宏仁. 机动目标“当前”统计模型与自适应跟踪算法 [J]. 航空学报, 1983, 4(1): 73-86.
- ZHOU Hongren. “Current” statistical model and adaptive algorithm for estimating maneuvering targets [J]. Acta Aeronautica et Astronautica Sinica, 1983, 4(1): 73-86.
- [26] LI Tiancheng, CHEN Huimin, SUN Shudong, et al. Joint smoothing and tracking based on continuous-time target trajectory function fitting [J]. IEEE Transactions on Automation Science and Engineering, 2019, 16(3): 1476-1483.
- [27] LI Tiancheng, FAN Hongqi. A computationally efficient approach to non-cooperative target detection and tracking with almost no a-priori information [EB/OL]. (2021-04-20)[2022-08-01]. <https://arxiv.org/abs/2104.11122>.

(编辑 陶晴 刘杨)