

基于动态粒子群优化的目标跟踪算法

郝振兴, 胡朝晖

(空军工程大学 航空航天工程学院, 西安 710038)

摘要: 目标跟踪问题的关键在于如何寻找与目标运动状态匹配的运动模型; 交互式多模型算法的模型集是根据先验信息确定的, 它不随时间变化而变化, 并且要求在模型集中任意时刻都存在描述目标运动模型; 在实际中需要大量模型来描述运动; 将粒子群优化和变结构多模型算法相结合, 不仅能充分利用系统的实时量测信息, 还能根据其先验信息调节优化算法结构; 仿真表明, 运用动态自适应粒子群优化算法实现模型集自适应, 可以提高目标跟踪的精度和实时性。

关键词: 目标跟踪; 交互式多模型算法; 变结构多模型算法; 动态优化; 粒子群优化算法

Target Tracking Algorithm Based on Dynamic Particle Swarm Optimizer

Hao Zhenxing, Hu Zhaozhui

(Engineering College of Aeronautics and Astronautics, Air Force Engineering University, Xi'an 710038, China)

Abstract: The key to target tracking problem is how to find and match the motion model of target motion state. Interactive multiple model algorithm of the model set is set according to the prior information, it does not changes over time, and requires concentration at any time in the model are described target motion model. In practice, need a lot of model to describe the motion. Particle swarm optimization combined with variable structure multiple model algorithm, can not only make full use of the information system of real-time measurement, can also according to the prior information structure optimization algorithm. Simulation shows that the use of dynamic adaptive particle swarm optimization algorithm adaptive implementation model set, can improve the accuracy and real-time performance of target tracking.

Keywords: target tracking; interacting multiple models (IMM); variable structure multiple models; dynamic optimizer; particle swarm optimizer (PSO)

0 前言

在目标跟踪问题中, 描述目标运动的数学模型是否准确对目标跟踪性能有极大的影响。目标跟踪的难点在于其运动的不确定性。一般来说, 建立的数学模型越接近真实运动模式, 跟踪效果越好。大多数的目标跟踪算法基于目标运动模型, 他们假设目标的运动及测量都可以用一个精确的数学模型表示出来, 比如匀速直线、匀加速和圆周转弯运动模型以及 Singer 模型^[1]等。

当运动模型不匹配时, 所估计出的目标状态结果与实际相比会出现极大偏差。为此, Bar Shalom 等人于 1988 年提出交互式多模型算法^[2]。该算法以广义伪贝叶斯算法为基础, 假设不同模型之间的转移服从已知转移概率的有限马尔科夫链, 引入马尔科夫转移概率矩阵分析各模型间的切换。算法的核心是, 根据各运动模型与目标运动过程的匹配程度, 系统为每个模型分配一定的概率, 当模型与目标运动状态匹配度高时, 该模型被分配的概率大, 反之模型被分配的概率小, 最后根据每个模型的似然函数进行数据融合, 输出结果。但在实际环境中, 很难用一组固定的模型集去准确描述目标机动运动, 大型的模型集必然导致运算量成倍增加, 模型切换延迟过长, 算法实时性下降; 算法中马尔科夫转移矩阵、系统的过程噪声等参数需要人为设定, 影响了算法的客观性。

为了提高目标跟踪系统的自适应性, Li. X. R, Bar Shalom 等提出变结构多模型算法实时调整模型集^[3-4]。变结构多

模型算法根据实际需要选择性的加权融合模型集中的模型, 核心技术是模型集自适应算法, 即新模型集的激活与当前模型集中不匹配模型的删减。

Kennedy 和 Eberhart 提出的粒子群优化算法是一种基于群体智能的并行随机优化算法^[5]。在粒子群优化算法中, 每个粒子都有一个由被优化的函数所决定的适应值和一个决定寻优方向和距离的速度, 所有粒子通过追随当前的最优粒子在解空间的可行域中进行搜索。动态粒子群算法的特点是该算法具有很好的跟踪动态极值的能力, 只要外部环境发生一定程度的变化, 以上算法就能监测出变化, 并且根据相应策略自动地初始化种群中的一些粒子, 然后根据更新公式重新计算适应度值, 得到最新的最优值。

本文基于粒子群优化算法的优点, 运用动态自适应粒子群优化设计变结构多模型算法, 将动态优化和多模型算法结合起来, 以提高目标跟踪的精度和实时性。

1 目标跟踪理论

1.1 目标跟踪原理

典型的目标跟踪算法包括: 测量目标运动参数、建立目标运动模型、检测并辨别目标机动、外推滤波。

目标运动状态参数主要包括位置、速度、加速度、角速度等。在此假设由机载传感器获得原始测量数据是目标位置, 则其速度、加速度等信息均可计算得到。研究目标跟踪问题时通常用一个时间序列表示其状态空间, 一般情况下, 系统的状态方程和测量方程为:

$$x(k+1) = f(x(k), u(k)) + w(k)$$

$$z(k) = h(x(k), u(k)) + v(k)$$

其中: $x(k)$ 为 k 时刻的目标运动状态向量, 通常包括位

收稿日期: 2016-03-28; 修回日期: 2016-04-14。

作者简介: 郝振兴(1991-), 男, 河北唐山人, 硕士研究生, 主要从事航空指挥控制引导。

假设测量状态是线性的，那么该问题的状态方程和测量方程分别如下：

$$x(k+1) = \mathbf{F}(k)x(k) + G(k)w(k)$$

$$z(k) = \mathbf{H}(k)x(k) + v(k)$$

其中： $\mathbf{F}(k)$ 为目标运动状态矩阵； $\mathbf{H}(k)$ 为测量矩阵。

对于非机动目标而言,目标运动状态几乎不发生改变,可以直接对实际测量量进行滤波处理,实施跟踪。对于机动目标而言,由于自身需求或因外界干扰,目标有意或无意地做与之前运动轨迹不同的运动,如匀速直线运动变为匀加(减)速、匀速转弯、爬升、俯冲、螺旋运动等。机动目标跟踪实质上就是实际测量量和状态预测量的差值进行机动判别,然后利用滤波算法估计目标状态量。目标跟踪原理如图1所示^[6]。

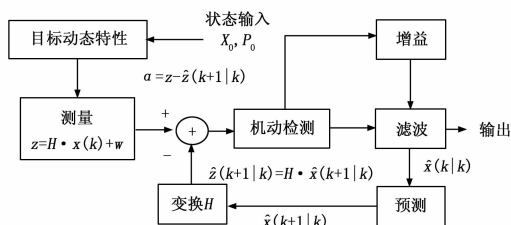


图 1 目标跟踪原理图

1.2 目标运动数学模型

目标运动模型包括状态转移矩阵和测量矩阵。状态转移矩阵建立目标在当前时刻状态与前一时刻状态的函数关系, 测量矩阵描述实际测量量与状态变量的函数关系。

本文主要应用圆周转弯模型^[7]，描述如下。

在离散情况下, 圆周转弯模型的时间离散状态方程为:

$$X(k+1) = F(k)X(k) + G(k)w(k)$$

假设在二维坐标系中, k 时刻目标的状态向量为 $x(k) = [x \ \dot{x} \ y \ \dot{y} \ \omega]^T$, $x, \dot{x}, y, \dot{y}$ 依次对应 X 和 Y 方向上位置和速度分量; ω 为运动目标的角速度。

若采样间隔为 T ，则状态转移矩阵为：

$$\mathbf{F}(k) = \begin{bmatrix} 1 & \frac{\sin(\omega T)}{\omega} & 0 & -\frac{1 - \cos(\omega T)}{\omega} & 0 \\ 0 & \cos(\omega T) & 0 & -\sin(\omega T) & 0 \\ 0 & \frac{1 - \cos(\omega T)}{\omega} & 1 & \frac{\sin(\omega T)}{\omega} & 0 \\ 0 & \sin(\omega T) & 0 & \cos(\omega T) & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

状态方程噪声输入矩阵为:

$$\mathbf{G}(k) = \begin{bmatrix} \frac{T^2}{2} & 0 & 0 \\ T & 0 & 0 \\ 0 & \frac{T^2}{2} & 0 \\ 0 & T & 0 \\ 0 & 0 & T \end{bmatrix}$$

当 $\mathbf{F}(k)$ 中 $\omega = 0$ 时, 圆周转弯模型转化成了匀速直线模型。另外, 当 ω 作为输入变量时, 它的值会随时间而变化, 因

此一般情况下圆周转弯模型是非线性模型。

1.3 交互式多模型算法

交互式多模型 (IMM) 算法包含一个交互式作用器, 多个滤波器, 一个模型概率估计器和一个估计混合器。算法包含 4 个步骤: 模型交互运算、滤波、模型概率更新、状态估计。

在设置目标运动模型集时,考虑到滤波器的性能和解算时间上的要求,应顾及目标的主要运动特征,不宜过细。因此,IMM 算法中目标运动模型至少包括一个非机动模型(如常速运动模型)和一个机动模型(如转弯模型)。

假设在 IMM 算法中一共设置了 r 个目标运动模型, 其离散的状态方程如下:

$$X(k+1) = F_i X(k) + w_i(k), i = 1, 2, \dots, r$$

其中: F_i 是系统的状态转移矩阵, $w_i(k)$ 是均值为零, 协方差矩阵为 Q 的过程噪声, 模型 i 的量测方程为:

$$Z(k) = H_i X(k) + v_i(k)$$

其中: $\mathbf{H}_i$ 为系统测量矩阵; $v_i(k)$ 为高斯测量噪声, 其协方差为 R 。

系统的马尔科夫转移矩阵为:

$$\mathbf{P} = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1r} \\ p_{21} & p_{22} & \cdots & p_{2r} \\ \vdots & \vdots & \vdots & \vdots \\ p_{r1} & p_{r2} & \cdots & p_{rr} \end{bmatrix}$$

其中： p_{ij} 是从模型 i 到模型 j 的转移概率。

这里的跟踪问题就是在获得 $k-1$ 时刻的各模型的概率 $\{\mu_i(k-1)\}_{i=1}^M$ ，模型转移概率矩阵 $\mathbf{P}_i(k-1)$ ，状态估计量 $\hat{X}_i(k-1|k-1)$ ，协方差 $\hat{P}_i(k-1|k-1)$ 和 k 时刻的测量量 $Z(k)$ ，估计各模型 k 时刻的状态 $\hat{X}_i(k|k)$ ，协方差 $\hat{P}_i(k|k)$ ，以及更新各模型该概率 $\{\mu_i(k-1)\}_{i=1}^M$ ，最后以此估计目标状态 $\hat{X}(k|k)$ 。

(1) 模型交互运算。输入交互是在已经获得给定模型上一时刻的状态估计值, 协方差的估计值, 并获取新的量测值 $Z(k)$ 之后, 对模型进行重新的初始化运算。假设, 对于模型 i , 在 $k-1$ 时刻交互运算的状态估计值和协方差矩阵分别为:

$$\bar{X}_i(k-1 | k-1) = \sum_{i=1}^M \hat{X}_i(k-1 | k-1) \cdot \mu_{ij}(k-1)$$

$$\bar{P}_i(k-1 \mid k-1) =$$

$$\sum_{i=1}^M \{ [\hat{X}_i(k-1 | k-1) - \bar{X}_i(k-1 | k-1)] \cdot [\hat{X}_i(k-1 | k-1) - \bar{X}_i(k-1 | k-1)]^T + \hat{P}_i(k-1 | k-1) \} \cdot \mu_{ij}(k-1), i = 1, 2, \dots, M$$

其中： $\mu_{ij}(k-1) = \frac{1}{c_i(k-1)} \cdot p_{ij} \cdot \mu_j(k-1)$ 是混合概率，

而 $c_i(k-1) = \sum_{j=1}^M p_{ij} \cdot \mu_j(k-1)$ 是归一化因子。

(2) 滤波。在交互运算的结果和测量量 $Z(k)$ 的基础上, 通过各匹配滤波器, 得到各模型的状态估计量 $\hat{X}_i(k|k)$ 和协方差估计量 $\hat{P}_i(k|k)$ 。一般, IMM 算法采用 Kalman 滤波器。对应于 i 模型, 进行卡尔曼滤波, 状态预测值:

$$\hat{X}_i(k|k-1) = F_i \bar{X}_i(k|k-1)$$

状态预测误差协方差为:

$$P_i(k | k-1) = F_i \bar{P}_i(k-1 | k-1) F_i^T + Q_i$$

预测测量：

$$Z_i(k|k-1) = h(\hat{X}_i(k|k-1))$$

卡尔曼滤波增益为:

$$K_i(k) = P_i(k|k-1)H_i^T S_i^{-1}(k)$$

k 时刻的滤波值为:

$$\hat{X}_i(k|k) = \hat{X}_i(k|k-1) + X_i(k)^T Z(k) - H_i S_i^{-1}(k) \hat{X}_i(k|k-1)$$

滤波协方差为:

$$P_i(k|k) = [1 - K_i(k)H_i]P_i(k|k-1)$$

残差协方差:

$$S_i(k) = H_i P_i(k|k-1) H_i^T + R$$

(3) 模型概率更新。模型概率更新是 IMM 模型中至关重要的环节, 它决定着每个模型在输出结果中所占据的比重。模型概率 $\{\mu_i(k-1)\}_{i=1}^M$ 随目标运动时刻变化, 需要及时更新。一般模型概率更新采用最大似然函数法, 计算出每个模型和当前运动模式的匹配程度。

计算各模型的似然函数:

$$\Lambda_i(k) = N(\delta_i(k); 0, S_i(k)), i = 1, 2, \dots, M$$

其中: $\delta_i(k)$ 为模型 i 的残差。

$$\delta_i(k) = Z(k) - \hat{Z}_i(k|k-1)$$

则得到模型概率:

$$\mu_i(k) = \frac{1}{c(k)} \cdot \Lambda_i(k) \cdot c_i(k-1), i = 1, 2, \dots, M$$

$$\text{其中: 归一化因子 } c(k) = \sum_{i=1}^M \Lambda_i(k) \cdot c_i(k-1)$$

(4) 状态估计。求各模型估计量的均值, 得到状态的估计量和协方差矩阵:

$$\hat{X}(k|k) = \sum_{i=1}^M \mu_i(k) \cdot \hat{X}_i(k|k)$$

$$\hat{P}(k|k) = \sum_{i=1}^M \mu_i(k|k) \{ p_i(k|k) + [\hat{X}_i(k|k) - \hat{X}(k|k)] \cdot [\hat{X}_i(k|k) - \hat{X}(k|k)]^T \}, i = 1, 2, \dots, M$$

至此, 完成了交互式多模型算法的一次递推。在下次滤波时将本次得到的输出值作为交互输入的输入值, 开始新的滤波输出。

2 基于动态自适应粒子群优化的变结构模型

2.1 变结构多模型算法

变结构多模型算法能根据目标的机动情况来调整每一个滤波时刻的模型集合。当出现以下这些情况时, 尽可能考虑使用该算法: 1) 目标运动模式未知或者系统模式时变; 2) 目标运动模式空间很大或者目标运动模式的先验信息比较复杂; 3) 计算机软硬件能力有限, 导致数据处理能力或资源受限。

变结构多模型算法增添了模型集自适应的内容, 步骤包括模型交互运算; 滤波; 模型概率更新; 模型集更新; 状态估计。由于变结构多模型算法是交互多模型算法的改进, 下面的变结构多模型算法, 仅讨论模型集的设计和自适应切换。

2.2 粒子群算法的基本描述

标准 PSO 算法中, 每个粒子代表一个可能的解, 所有的粒子组成群体。算法首先初始化为一群随机粒子 (随机初始解), 然后通过迭代更新其速度和位置。在每一次迭代中, 粒子通过跟踪两个“最优值”来更新自己的状态: 第一个是粒子个体本身在历史上所找到的最优解, 称作个体位置最优值; 另

一个是整个群体到目前为止找到的最优解, 称作群体位置最优值。粒子在解空间中根据上述的自身历史信息和群体信息共同决定其“飞翔”的速度和方向, 以此来寻找最优解。

假设在 D 维搜索空间中进行问题求解, 群体由 m 个粒子组成, $Swarm = \{x_1^{(k)}, x_2^{(k)}, \dots, x_m^{(k)}\}$ 。 k 时刻第 i 个粒子在搜索空间中的位置向量为 $x_i^{(k)} = (x_{i1}^{(k)}, x_{i2}^{(k)}, \dots, x_{iD}^{(k)})$, $i = 1, 2, \dots, m$, 这是粒子个体在搜索空间中的位置, 也代表问题的一个可能解。与该个体位置向量相对应的是其速度向量 $v_i^{(k)} = (v_{i1}^{(k)}, v_{i2}^{(k)}, \dots, v_{iD}^{(k)})$, 分别描述了该粒子在空间每一维上的运动情况。变量带括号的上标表示迭代周期, 如 $x_{id}^{(k+n)}$ 表示第 $k+n$ 个周期。

PSO 算法的邻域函数在每一个迭代周期根据个体自身位置向量、速度向量、个体历史信息、群体信息和扰动来产生新的位置状态。第 i 个粒子在是 $k+1$ 时刻的第 d 维邻域函数计算公式为:

$$\begin{cases} v_{id}^{(k+1)} = \omega v_{id}^{(k)} + c_1 r_1 (p_{id}^{(k)} - x_{id}^{(k)}) + c_2 r_2 (p_{id}^{(k)} - x_{id}^{(k)}) \\ x_{id}^{(k+1)} = x_{id}^{(k)} + v_{id}^{(k+1)} \end{cases}$$

其中: ω 为惯性因子, 是粒子群算法中最重要的参数, ω 较大时全局搜索能力比较强, ω 较小时局部搜索能力增强; k 为迭代周期; c_1, c_2 为加速因子; r_1, r_2 为 $(0, 1)$ 之间服从均匀分布的随机数; $x_{id}^{(k)}$ 为粒子的当前位置; $v_{id}^{(k)}$ 为粒子运动速度; $p_{id}^{(k)}$ 为粒子个体位置最优值; $p_{id}^{(k)}$ 为群体位置最优值。

Shi, Clerc 等建议 $\frac{(c_1 + c_2)}{2}$ 取值为 1.494, 惯性因子 ω 取值为 0.729^[8-9]。

2.3 动态单目标优化问题的定义

相对于传统静态环境下的优化问题的概念而言, 动态优化问题是指在动态环境下的优化问题。具体来讲是指适应度函数、约束条件或参数变量等因素随时间而变化的优化问题^[10]。这种变化, 或是离散变化, 或是连续变化, 甚至有可能两种变化交替进行。

单目标动态优化问题可以定义为:

$$\begin{cases} \max/\min f(x, t) = f_1(x) \\ \text{s. t. } g_i(x, t) \leq 0, i = 1, 2, \dots, q; \\ h_j(x, t) = 0, j = 1, 2, \dots, p \end{cases}$$

其中: $x = (x_1, x_2, \dots, x_n)^T \in \Omega \subset R^n$, 为 n 维决策 (变量) 向量, 即可行解, Ω 为 n 维决策空间, 即搜索空间; $f(x, t)$ 为目标函数, 也称为评价函数; $g_i(x, t) \leq 0, i = 1, 2, \dots, q$, 定义了 q 个不等式约束条件, $h_j(x, t) = 0, j = 1, 2, \dots, p$, 定义了 p 个不等式约束条件。

动态优化问题的目标就是寻找一系列随时间变化的最优个体的集合。对于一个极小化动态优化问题有任意 $x \in R^n$, 存在 $x^*(t)$, 使得 $t \in T$ 满足 $f(x^*(t), t) \leq f(x(t), t)$, 则 $x^*(t)$ 为 t 时刻的最优值。

在任意一个固定时刻, 评价函数 $f(x, t)$ 、约束函数 $g_i(x, t)$ 与 $h_j(x, t)$ 的值都是确定的。但是, 他们的值依赖于时间 t , 会随着时间的变化而不断变化, 因此会导致问题的最优解也随着时间变化。求解动态优化问题的算法需要能够持续地跟踪不断变化的最优解, 而不是当发生动态环境变化后再重新开启一次优化过程。求解动态优化问题的挑战就是需要利用以前环境变化复用信息来加速变化后的优化过程。

PSO 算法及其众多的改进算法在众多静态问题的优化上已被成功运用。对于动态问题的优化, 不仅要求算法能够对环境的变化快速做出反应, 并且不再是单一地要求算法在解空间中寻找最优值, 而是要尽可能地在解空间中跟踪运动变化的最优点的动态轨迹, 时刻获得当前最优解。这就涉及动态优化问题中 PSO 算法的搜索策略、关于环境变化的检测方法、动态响应策略。

2.4 针对动态优化问题的 PSO 改进搜索策略

动态优化问题中关于 PSO 算法的标准搜索策略为粒子在搜索空间中不断通过自身信息 $p_{id}^{(k)}$ 和群体信息 $p_{gd}^{(k)}$ 向目标点运动。动态 PSO 算法在随机搜索策略基础上, 增加了环境检测过程, 并对检测结果做出响应, 调整动态 PSO 算法的搜索策略。

Carlisle 使用的自适应 PSO, 在搜索空间中随机选择一个点作为环境变化量测点, 而且当且仅当它的适应值变化超过一个设定的范围时, 算法进行动态环境响应操作, 以激发群体多样性^[11]。

使用带线性递减惯性因子, 有益于增加群体多样性, ω 随着迭代周期递减, 使得粒子动态系统的参数稳定域变宽, 从而保证了群体将最终收敛到解空间中的某一个极值区域。

自适应权重能够使粒子朝向较好的搜索区域, 它的表达式如下:

$$\omega = \begin{cases} \omega_{\min} - \frac{(\omega_{\max} - \omega_{\min})(f - f_{\min})}{f_{\text{avg}} - f_{\min}}, & f \leq f_{\text{avg}} \\ \omega_{\max}, & f > f_{\text{avg}} \end{cases}$$

其中: $\omega_{\max}$, $\omega_{\min}$ 分别为 ω 的最大值和最小值; f 为粒子当前的目标函数值; f_{avg} 和 $f_{\min}$ 为当前所有粒子的平均目标值和最小目标值。

根据目标机动强弱, 模型集包含非机动, 弱机动, 强机动, 转弯等不同运动模式对应的特征值。对于该算法需要首先在目标的运动模式空间中得到目标的机动值, 再根据该机动值在模式空间中的模型集合中选择不同的模型进行模式匹配。动态 PSO 算法预先设定一个可以描述目标整个运动模式的模型集合 (搜索空间), 模型集合中的每个模型都可看作搜索域中的可能解。根据滤波信息在一定的准则下决定是否切换最优解用于下一时刻的滤波。具体的数学描述如下。

对于时间状态变量 t , 变量值 $\omega^*(t)$, 存在 $f(t, \omega^*(t))$ 并且满足:

$$\min J(t) = |\tilde{f}(t, \omega^*(t)) - \tilde{f}(t, \omega(t))|$$

则 $\omega^*(t)$ 为动态函数在 t 时刻的最优解。

2.5 变结构多模型问题的动态 PSO 算法描述

在动态环境当中, 粒子的个体最优值和群体最优值都可能会随着时间而不断变化, 且粒子容易停滞于前一环境的寻优状态, 因此对粒子群算法做了一定程度的改进, 使其更好地跟踪动态极值。

1) 引入探测粒子, 通过设置一部分粒子为探测粒子使其感知外部环境是否变化;

2) 设置响应机制, 当指定的探测粒子发现环境变化时, 采取一定的响应措施对种群进行更新来适应变化后的环境。

采用分段划分粒子法, 首先将初始种群中的所有粒子均匀地划分成 n_1 个子集, 然后在每个子集中随机地初始化 n_2 个探测粒子, 这些粒子就是用来监测环境是否发生变化的。每次迭

代时计算所有探测粒子所对应的适应度函数值 f_i , 然后由式 (1) 计算最邻近两次迭代后的适应度值差值 Δf_i , 最后对所有差值 Δf_i 绝对值求总和 (F_{Δ})。

$$\Delta f_i = f_i(k+1) - f_i(k) \quad (1)$$

$$F_{\Delta} = \sum_{i=1}^n |\Delta f_i| \quad (2)$$

环境变化的依据就是判断 F_{Δ} 是否等于 0, 如果 $F_{\Delta} \neq 0$ 表示认为外部环境已发生了一定程度的变化, 但是如果发生微小的变化都进行重新计算将会造成计算量的增加, 因此需要设定一个界限, 即响应极限值 δ_F 。当变化后的值 F_{Δ} 超过了极限值 δ_F 时将发生响应的触发。采取的响应机制就是选择一定比例的粒子随机重新初始化或者全部重新初始化。如果 $F_{\Delta} > \delta_{\Delta}$, 那么:

$$\begin{cases} v_{id}^{(k)} = v_{id}^{(k)} \cdot rand \\ x_{id}^{(k)} = x_{id}^{(k)} \cdot rand \end{cases}$$

其中: $rand$ 为 $[0, 1]$ 内任意的随机数。

具体算法流程为:

1) 把可行空间划分为 n_1 个均匀子集, 在每个子集内随机初始化 n_2 个探测粒子, 共生成 $n_1 \times n_2$ 个探测粒子, 计算探测粒子适应度 $J_i^m (i = 1, 2, \dots, n_1 \times n_2)$ 。

2) h 初始化参数。惯性权重 ω_i , 学习因子 c_1, c_2 , 动态出发响应阈值 δ_F , N 个 m 维粒子的种群 $x_{id}^{(k)}$, 初始化粒子速度 $v_{id}^{(k)}$;

3) 根据适应度函数计算每个粒子的适应度值 J_i , 根据适应度值得到个体极值 $p_{id}^{(k)}$ 和群体极值 $p_{gd}^{(k)}$;

4) 根据 PSO 更新方程定义的迭代公式产生新群体 $x_{id}^{(k)}$, 根据适应度函数计算新群体中每个粒子的适应度值 f_i ;

5) 把新种群个体极值适应度值 $p_{id}^{(k)}$ 同原个体最优值适应度值 $p_{id}^{(k)}$ 、群体最优值的适应度值 $p_{gd}^{(k)}$ 做比较, 根据新种群的适应度值更新个体极值 $p_{id}^{(k)}$ 和群体极值 $p_{gd}^{(k)}$;

6) 计算探测粒子适应度值 J_i^m , 根据公式 (2) 求 F_{Δ} , 若 $F_{\Delta} > \delta_{\Delta}$, 则按比例初始化种群和粒子速度, 转 3);

7) 若满足结束条件, 则算法结束, 否则转 3)。

新激活模型的初始化包括为新激活模型分配转移概率和给出新激活模型的初始状态估计及误差协方差两个方面。

由于系统模型之间的切换符合 Markov 链, 所以给新激活模型分配概率时只考虑那些可以跳变过来的模型的概率。即新激活模型的初始值由前一时刻模型根据局部转移概率来分配。某时刻该模型的初始值仅由上一时刻与它相邻的模型根据一定的转移概率加权而来。其中局部模型转移概率矩阵为:

$$p_{ij} = \begin{bmatrix} p_{i-1|i-1} & p_{i-1|i} & 0 \\ p_i|i-1 & p_i|i & p_i|i+1 \\ 0 & p_{i+1|i} & p_{i+1|i+1} \end{bmatrix}$$

模型切换时原有模型的状态向量和协方差可以继续沿用上时刻各模型滤波的结果; 对于新激活模型的状态向量和协方差的初始化, 采用上一时刻各模型的概率加权组合来进行, 即:

$$\hat{X}_{new}(k-1|k-1) = \sum_{i=1}^M \hat{X}_i(k-1|k-1) \cdot u_i(k-1)$$

$$\hat{P}_{new}(k-1|k-1) = \sum_{i=1}^M \{\hat{P}_i(k-1|k-1) +$$

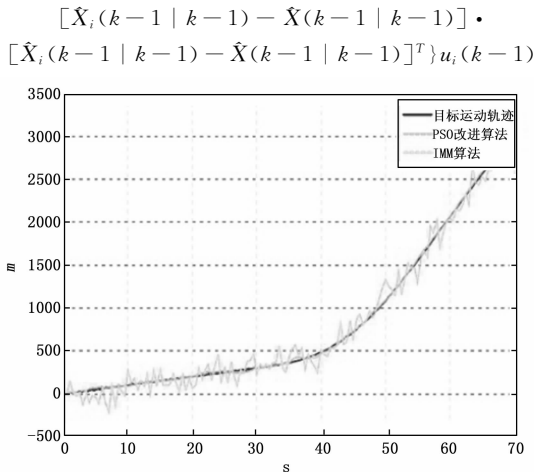


图 2 目标运动轨迹跟踪效果比较

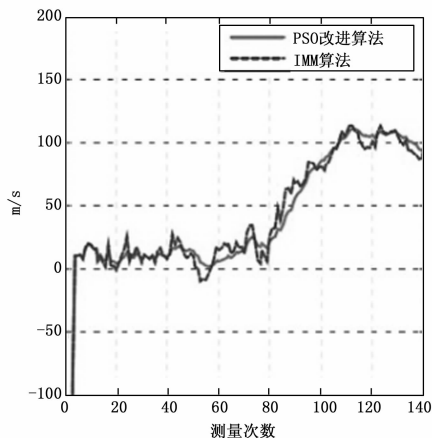


图 3 目标速度跟踪效果比较

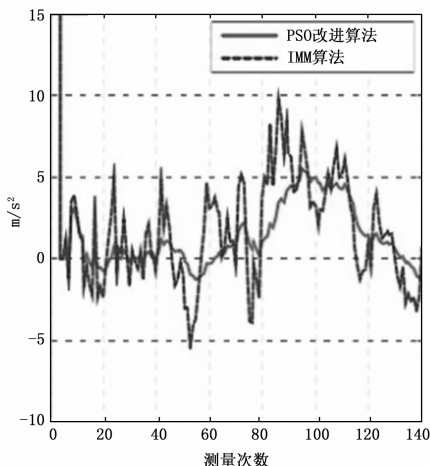


图 4 目标加速度跟踪效果比较

3 仿真与分析

假设目标以角速度 ω 机动运动, $\omega \in (-12^\circ, 12^\circ)$ 。将该范围作为搜索域, 每一个可能解作为模型的特征参数。将初始角速度代入圆周转弯模型中, 得出转移矩阵, 然后按照算法流程

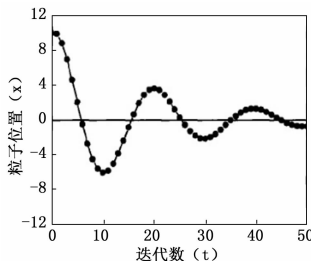


图 5 IMM 算法迭代次数与收敛速度

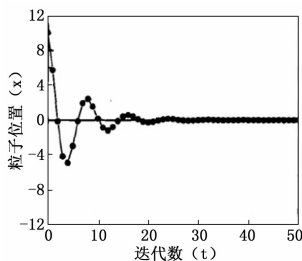


图 6 PSO 改进算法迭代次数与收敛速度

进行仿真。

由图 2~图 4 可知, 改进后算法的位置、速度、加速度的误差值都小于 IMM 算法, 表明了目标跟踪效果的优越性。由图 5 和图 6 知, 算法的运行时间缩短, 提高了解算的实时性。

4 结论

本文基于粒子群优化算法, 利用动态自适应 PSO 解决了目标跟踪变结构多模型算法的模型集自适应问题。在机载目标跟踪系统测量信息十分有限的情况下, 算法的结构对目标发生的跳变有着良好的跟踪效果, 有助于提高目标跟踪的精度和实时性。与传统交互式多模型算法相比, 该算法通过在线跟踪测量信息获取模型参数并且建立可变的模型集限定系统模型集中模型的寻优, 保证跟踪精度的同时其计算量也不会明显增加。

参考文献:

- [1] Singer R A. Estimating optimal tracking filter performance for manned maneuvering targets [J]. IEEE Transactions on Aerospace and Electronic Systems, 1970, (4): 473-483.
- [2] Blom H, Bar Shalom. The interacting multiple model algorithm for systems with markov switching coefficients [J]. IEEE Trans on Automatic Control, 1988, 33 (8): 780-783.
- [3] Li X R, Bar-Shalom Y. Design of an interacting multiple model algorithm for air traffic control tracking [J]. IEEE Transactions on Control Systems Technology, 1993, 9 (1): 186-194.
- [4] Li X R, Bar-Shalom Y. Performance prediction of the interacting multiple model algorithm [J]. IEEE Transactions on Aerospace and Electronic Systems, 1993, 7 (29): 755-771.
- [5] Kennedy J, Eberhart R C. Particle swarm optimization [J]. IEEE Service Center, Piscataway, NJ, 1995, 6: 1942-1948.
- [6] Asseo S J. Effect of monopulse signal thresholding on tracking multiple targets [J]. IEEE Trans on AES, 1974, 10 (4): 501-509.
- [7] 夏佩伦. 目标跟踪与信息融合 [M]. 北京: 国防工业出版社, 2010.
- [8] Shi Y, Eberhart R. Parameter selection in particle swarm optimization [A]. Proceedings of the 7th International Conference on Evolutionary Programming [C]. London, UK: Springer, 1998, 591-600.
- [9] Clerc M, Kennedy J. The particle swarm: explosion, stability and convergence in a multi-dimensional complex space [J]. IEEE Transactions on Evolutionary Computation, 2002, 6 (1): 58-73.
- [10] Branke J. Evolutionary optimization in dynamic environments [M]. Kluwer Academic Publishers, 2001: 95-113.
- [11] Carlisle A, Dozier G. Adapting particle swarm optimization to dynamic environments [C]. Proceedings of International Conference on Artificial Intelligence, Las Vegas, USA, 2000: 429-434.