

基于和声差分进化的UKF改进算法

贺军义¹, 李男男^{2†}

(1. 河南理工大学 计算机科学与技术学院, 河南 焦作 454000;

2. 河南理工大学 物理与电子信息学院, 河南 焦作 454000)

摘要: 无迹卡尔曼滤波(UKF)被广泛应用于工程实际中,但传统UKF在滤波过程进行无迹变换(UT)时的待选参数为固定值,这会带来一定误差.为了获取最优的待选参数,提出基于和声差分进化(HSDE)的UKF改进算法,并在目标跟踪中对该算法进行应用.和声差分进化算法对待选参数 κ 进行最优选择,跳出局部最优的现象还有很强的收敛性,通过改进可进一步提高UKF算法滤波精度. Matlab仿真结果表明,基于和声差分进化的UKF改进算法精度更高.

关键词: 无迹卡尔曼滤波; 无迹变换; 和声差分; 目标跟踪; 跟踪误差

中图分类号: TP273

文献标志码: A

Improved UKF algorithm based on harmony search differential evolution

HE Jun-yi¹, LI Nan-nan^{2†}

(1. School of Computer Science and Technology, He'nan Polytechnic University, Jiaozuo 454000, China; 2. School of Physics and Electronic Information Engineering, He'nan Polytechnic University, Jiaozuo 454000, China)

Abstract: Unscented Kalman filter(UKF) is widely used in practical engineering, but in the unscented transform(UT) of the filtering process of the traditional UKF, the parameters choose a fixed value, which can bring some errors. In order to obtain the best parameters to be selected, an improved UKF algorithm based on harmony search evolution differential(HSED) is proposed and applied to the target tracking. The harmony differential evolution algorithm can be used to select the optimal parameters, jump out of the local optimum, and has a strong convergence, which can further improve the accuracy of the UKF algorithm. The results of Matlab simulation show that the improved UKF algorithm based on harmonic differential evolution is more accurate.

Keywords: unscented Kalman filter; unscented transform; harmony search differential evolution; target tracking; tracking error

0 引言

自从UKF被提出以来,在许多应用领域得到了广泛的应用,如姿态估计^[1]、导航^[2]、容错控制^[3]、状态估计^[4]、定位、目标跟踪与拦截以及工业生产^[5]、地震预报等. UKF以UT变换为基础,使用UT变换来处理均值和协方差的非线性传递,直接利用非线性状态方程来估算状态向量的概率密度函数,无需计算Jacobian矩阵,无需忽略高阶项^[6]. 相对于扩展卡尔曼滤波(EKF)的一阶精确,UKF的估计精确度提高了对高斯数据的3阶精确和对任何非线性非高斯数据的2阶精确^[7],因而有较高的计算精度,能够有效地提高

估计精度和稳定性.

由于UKF算法的滤波效果优于其他很多滤波算法,UKF的研究及应用成为了很多国内外学者的研究对象,并对其进行了改进研究,使其能够应用于各个领域. 如欧县华等^[8]提出了在无线传感器网络中使用一种改进的UKF算法,即自适应迭代UKF算法; Lin等^[9]在目标跟踪中对UT变换的Sigma抽样方法进行了改进,使用调整因子确保半正定协方差矩阵的估计,采用最小二乘法算法拟合滤波更新后的误差,有效地提高了跟踪的准确性; Meng等^[10]针对导航卫星系统检测定位的准确性问题,运用正交三角

收稿日期: 2017-03-27; 修回日期: 2017-06-18.

基金项目: 河南省教育厅高校重点应用研究计划基金项目(13150051); 河南省教育厅高校博士基金项目(6609007023).

责任编辑: 林崇.

作者简介: 贺军义(1982—),男,副教授,博士,从事惯性测试与控制技术、智能信号处理的研究; 李男男(1991—),女,硕士生,从事智能信号处理的研究.

†通讯作者. E-mail: mayi1938736257@sina.com

模型增强的无迹卡尔曼滤波分解算法进行数据融合,以避免建模误差和噪声不确定等问题;Huang等^[11]根据遗忘因子的概念推导出了平方根遗忘因子的无迹卡尔曼滤波算法(SMRA-UKF),在迭代计算中采用平方根矩阵代替协方差矩阵,有效地克服了系统的发散问题,引入遗忘因子把普通滤波器转换为强滤波器,提高了系统的鲁棒性.这些研究者针对应用系统的需求对UKF算法进行改进,但对UT变化的研究相对较少.如Dunik等^[12]提出了一种自适应的取值方法,应用网格方法或随机搜索的方法对 κ 进行选择,但是 κ 的选择与网格的密度密切相关,网格密度越小,能够搜索到的 κ 就越少,难以有效地提高滤波精度.同时,网格密度越大,搜索到的 κ 就越多,虽然提高了滤波精度,但时间复杂度大幅升高.金瑶等^[13]提出了一种基于自适应的UKF算法,使用差分演化对缩放参数进行优化选择.此方法有效地克制了固定缩放参数带来的发散,提高了滤波精度,但优化目标被固定在0~10的范围内,提高的滤波精度是有限的.UKF算法本身得到的滤波精度很高但也存在缺点,在无迹卡尔曼滤波的UT变换中,缩放比例参数是用来降低总预测误差的.但在计算缩放比例参数时,由于待选参数一般都设计成固定值,导致计算出来的缩放比例参数不够精确,预测误差偏大.鉴于此,本文提出一种基于和声差分进化的UKF改进算法,并应用于目标跟踪中.虽然UKF算法和基于差分进化的UKF改进算法在目标跟踪系统中的应用使目标跟踪比较精确,但是基于和声差分进化的UKF改进算法在目标跟踪中的使用可以使跟踪精度更高. Matlab仿真实验结果表明,与UKF算法、基于差分进化的UKF改进算法在目标跟踪系统中的应用相比,基于和声差分进化的UKF改进算法的目标跟踪系统跟踪精度更高,误差更小.

1 相关算法研究

1.1 无迹卡尔曼滤波算法

与EKF算法不同,UKF算法利用UT变换在估计点附近确定采样点,并用这些样本点表示高斯密度近似状态的概率密度函数.使用UT变换方法得到的非线性变换后的均值和协方差精度最少具有2阶精度,对于高斯分布,可以达到3阶精度.UT变换中的采样点是基于先验均值和先验协方差矩阵的平均根相关列选择的^[6].

文献[6-7]介绍了无迹变换的基本原理,它是UKF算法的核心.假设非线性变化 $y = (x)$,状态向量 x 为 n 维的随机变量,其均值和方差分别为 \bar{x} 和 p .通

过UT变换得到 $2n + 1$ 个采样点,即Sigma(X 表示),再由采样点和其相对应的权值 w 来计算 y 的统计特征.下面是UT变换的具体步骤.

Step 1: 计算 $2n + 1$ 个采样点,即Sigma点

$$\begin{cases} X^{(0)} = \bar{X}, i = 0; \\ X^{(i)} = \bar{X} + (\sqrt{(n+\lambda)p})_i, i \in (1, n); \\ X^{(i)} = \bar{X} - (\sqrt{(n+\lambda)p})_i, i \in (n+1, 2n). \end{cases} \quad (1)$$

其中: $(\sqrt{p})_i$ 表示矩阵方根的第 i 列, $(\sqrt{p})^T(\sqrt{p}) = p$, n 为状态维数.

Step 2: 计算采样点的相应的权值

$$\begin{cases} w_m^{(0)} = \frac{\lambda}{n+\lambda}, \\ w_c^{(0)} = \frac{\lambda}{n+\lambda} + (1 - \alpha^2 + \beta), \\ w_m^{(i)} = w_c^{(i)} = \frac{\lambda}{2(n+\lambda)}, i \in (1, 2n); \end{cases} \quad (2)$$

$$\lambda = \alpha^2 \times (n + \kappa) - n. \quad (3)$$

其中: w 的上标表示第几个采样点,下标 m 表示均值,下标 c 表示协方差; λ 为缩放比例参数,影响总的预测误差; α 为采样点的分布状态控制参数; κ 为待选参数; β 为非负的权系数,其作用是合并方程中高阶项的动差,计算权值时将高阶项的影响考虑在内.

Step 3: 对采样点进行非线性处理得到采样点集 $\{y^{(i)}\}$ 为

$$y^{(i)} = f(X^{(i)}). \quad (4)$$

Step 4: 对采样点进行加权处理, y 为 $y^{(i)}$ 的加权均值, p_y 为加权协方差,表示为

$$y = \sum_{i=0}^{2n} w_m^{(i)} \times y^{(i)}, \quad (5)$$

$$p_y = \sum_{i=0}^{2n} w_c^{(i)} \times (y^{(i)} - y)(y^{(i)} - y)^T. \quad (6)$$

虽然UKF算法有很多改进类型,但针对UT变化进行改进的相对较少.

1.2 传统差分进化算法

差分进化(DE)算法是一种新兴的计算技术,在很多领域已经得到应用,譬如神经网络、电力系统、机器人、信号处理、生物信息学、现代农业与食品安全、环境保护等^[14],它的理论研究与实际应用具有重要的学术意义和实用价值.DE算法是一种特殊的遗传算法,保留了基于种群的全局搜索策略,记忆能力使得算法可以动态地跟踪当前的搜索情况来调整搜索策略,采用实数编码、基于差分的简单变异操作以及一对一的竞争生存策略,降低了遗传算法的复杂

性^[14],具有较强的全局收敛性和鲁棒性.

文献[14-17]对差分进化过程建立了数学模型,并对其描述和分析,给出了标准DE的3个核心操作:变异、交叉和选择.其中,交叉操作有指数交叉和二项交叉,本文中使用的二项交叉.下面是DE算法的具体步骤.

Step 1: 种群初始化.

$$\begin{aligned} x_{i,k}(0) &= l_k + \text{rand}(\cdot) \times (u_k - l_k), \\ k &= 1, 2, \dots, d, i = 1, 2, \dots, N. \end{aligned} \quad (7)$$

u_k 和 l_k 为搜索的上界和下界, $\text{rand}(\cdot)$ 为 0~1 之间均匀分布的随机数.

Step 2: 变异操作. 假设变异机制为

$$X_i(g) = x_{r_1}(g) + f \times [x_{r_2}(g) - x_{r_3}(g)]. \quad (8)$$

$X_i(g)$ 为变异的个体; f 为压缩比例因子,取值范围为 0~1; $x_{r_1}(g)$, $x_{r_2}(g)$ 和 $x_{r_3}(g)$ 为 3 个父代.

Step 3: 交叉操作. 本文采用二项交叉方式,二项交叉的执行方式为

$$y_{i,j}(g) = \begin{cases} X_{i,j}(g), & r < \text{cr} \text{ or } j = \text{rnd}; \\ x_{i,j}(g), & \text{other.} \end{cases} \quad (9)$$

其中: r 是 0~1 之间的随机数, cr 是变量的交叉概率, rnd 是 1~ d 之间的整数. 若 $r < \text{cr}$ 或 $j = \text{rnd}$, 则接受目标个体对应的分量, 否则保留当前个体对应的分量.

Step 4: 选择操作. 传统差分进化算法采用贪婪机制进行选择,操作如下所示:

$$x_i(g+1) = \begin{cases} X_i(g), & f[X_i(g)] < f[x_i(g)]; \\ x_i(g), & \text{other.} \end{cases} \quad (10)$$

使用贪婪机制使种群平均性能不断提高,逐步得到最优解.

DE 算法的搜索能力依赖于算法控制参数的选择,包括压缩比例因子 (f)、交叉概率 (cr) 等^[14-15]. 压缩比例因子是 DE 的主要参数之一,文献[14-15]使用联合概率密度函数和雅可比矩阵,在 $f = 0.2$ 、 $f = 1$ 、 $f = 2$ 等不同的情况下计算式(10)中 $X_i(g)$ 的概率密度函数. f 太大会使种群分散程度过大,导致过多的种群个体靠近边界,减弱有效区域的搜索; f 太小会使得到的随机变量的概率密度函数变化较小,种群中的变异不会产生很好的效果. 因此压缩比例因子的选择很重要,一般选择 0.5~1 之间的值^[15]. DE 中另一个主要参数为交叉概率 cr ,其本质是调整历史信息 and 进化信息的权重. cr 越大,保留的历史信息越少,当代进化信息越多,种群在空间中的搜索速度越快,但可能导致局部调整不完全; cr 越小,保留种群历史信

息的概率越大^[14-15].

1.3 和声差分进化算法

和声搜索 (HS) 算法由 Geem 等在 2001 年提出,是根据演奏家们为了演奏出完美的和声而不断调试自己乐器的过程而产生的优化搜索算法^[18]. HS 提出后的首次实际应用是解决了旅行商的问题,目前在很多工程中都有所使用,比如资源筹集^[19]、学生排课^[20]、结构工程^[21]等. 基本的和声搜索算法过程如下.

Step 1: 初始化参数. 初始化记忆库大小 HMS、记忆库选择概率 HMCR、局部调整概率 PAR、调整步长 bw、调整次数 MaxItr、维数 Dim 等参数,设置终止条件(本文以调整次数为终止条件).

Step 2: 初始化和声记忆库.

$$x_{i,j} = x_{\min,j} + (x_{\max,j} - x_{\min,j})\text{rand}(\cdot). \quad (11)$$

其中: $x_{\min,j}$ 、 $x_{\max,j}$ 为和声记忆库第 j 维的上界和下界, $\text{rand}(\cdot)$ 为 0~1 内的均匀分布随机数.

Step 3: 生成新和声. 新和声的产生有 3 种方法: 记忆库学习、局部调整或者随机创作. 通过记忆库学习获得新和声

$$x_{\text{new}} = \begin{cases} x_{\text{rand},j}, & \text{rand} < \text{HMCR}; \\ x_{\min,j} + (x_{\max,j} - x_{\min,j}) \times \text{rand}(\cdot); \\ j \in (1, \text{Dim}). \end{cases} \quad (12)$$

调整局部产生新和声

$$x_{\text{new}} = \begin{cases} x_{\text{new}} \pm \text{rand}(\cdot) \times \text{bw}, & \text{rand}(\cdot) < \text{PAR}; \\ x_{\text{new}}, & \text{other.} \end{cases} \quad (13)$$

Step 4: 更新和声记忆库. 将新生成的和声与记忆库中的最差和声进行比较,若新和声优于最差和声,则新和声替换最差和声.

Step 5: 判断终止条件. 判断算法是否满足终止条件,如果不满足则返回 Step 3,如满足则结束.

由于 DE 算法是由两个个体的差分量产生新个体,为了保证搜索的分散性,得到有效的新个体,必须拥有较大的种群,虽然影响计算效率,但优化能力较强. 和声搜索算法简单易于实现,通过独立的变量调节产生新的和声^[22-23],每次迭代只生成一个和声向量,有较高的计算效率,但优化能力并不理想,容易造成局部最优的现象. 和声差分进化算法不仅提高了计算效率,且优化能力很强,可以避免局部最优.

2 基于和声差分进化的 UKF 改进算法

基于和声差分进化的 UKF 改进算法,主要是通过和声差分进化算法选择出最优的 UKF 待选参数 κ ,

以此提高跟踪精度. 下面给出基于和声差分进化的UKF算法的模型及模型建立步骤.

Step 1: 建立系统模型

$$X(k+1) = f(x(k), W(k)), \quad (14)$$

观测方程

$$Z(k) = h(x(k), V(k)). \quad (15)$$

其中: $W(k)$ 为两方向上的加性系统噪声, $V(k)$ 为雷达加性观测噪声.

Step 2: 选定 κ 的搜索范围为 $(0, 0.9n)$, n 为状态维数. 使用和声差分进化算法对 κ 进行变异、交叉和选择, 训练出最优解用于计算采样时的缩放比例参数. 进行和声差分进化算法时, 设定压缩比例因子 f 、交叉概率 cr , 进行最优选择, 选择出最优解 κ ; 然后把最优解 κ 代入到式(18)中, 求出缩放比例因子; 最后将 κ 代入式(16)和(17)中, 获得一组采样点及其对应的权值. 计算 $2n+1$ 个采样点, 即 Sigma 点

$$\begin{cases} X^{(0)} = \bar{X}, i = 0; \\ X^{(i)} = \bar{X} + (\sqrt{(n+\lambda)p})_i, i \in (1, n); \\ X^{(i)} = \bar{X} - (\sqrt{(n+\lambda)p})_i, i \in (n+1, 2n). \end{cases} \quad (16)$$

计算采样点的相应的权值

$$\begin{cases} w_m^{(0)} = \frac{\lambda}{n+\lambda}, \\ w_c^{(0)} = \frac{\lambda}{n+\lambda} + (1 - \alpha^2 + \beta), \\ w_m^{(i)} = w_c^{(i)} = \frac{\lambda}{2(n+\lambda)}; i \in (1, 2n); \end{cases} \quad (17)$$

$$\lambda = \alpha^2 \times (n + \kappa) - n; \quad (18)$$

$$X^i(k|k) = \begin{bmatrix} \hat{X}(k|k) \\ \hat{X}(k|k) + \sqrt{(n+\lambda)P(k|k)} \\ \hat{X}(k|k) - \sqrt{(n+\lambda)P(k|k)} \end{bmatrix}^T. \quad (19)$$

式(19)为采样点集合.

Step 3: 根据下式计算 $2n+1$ 个 Sigma 点集的一步预测:

$$\begin{aligned} X^{(i)}(k+1|k) &= f[k, X^{(i)}(k|k)], \\ i &= 1, 2, \dots, 2n+1. \end{aligned} \quad (20)$$

Step 4: 根据下式计算系统的状态量的一步预测及协方差矩阵:

$$\hat{X}(k+1|k) = \sum_{i=0}^{2n} w^{(i)} X^{(i)}(k+1|k), \quad (21)$$

$$\begin{aligned} P(k+1|k) &= \\ &\sum_{i=0}^{2n} w^{(i)} [\hat{X}(k+1|k) - X^{(i)}(k+1|k)] \times \\ &[\hat{X}(k+1|k) - X^{(i)}(k+1|k)]^T + Q. \end{aligned} \quad (22)$$

其中: $w^{(i)}$ 为权值, 通过式(17)得到; Q 为系统噪声 $W(k)$ 的协方差.

Step 5: 根据下式再次使用 UT 变换, 生成新的 Sigma 点:

$$X^{(i)}(k+1|k) = \begin{bmatrix} \hat{X}(k+1|k) \\ \hat{X}(k+1|k) + \sqrt{(n+\lambda)P(k+1|k)} \\ \hat{X}(k+1|k) - \sqrt{(n+\lambda)P(k+1|k)} \end{bmatrix}^T. \quad (23)$$

Step 6: 将式(23)预测的 Sigma 点集代入如下所示的观测方程, 得到预测的观测量:

$$Z^{(i)}(k+1|k) = h(X^{(i)}(k+1|k)). \quad (24)$$

Step 7: 通过对式(24)得到的观测预测值进行加权求和得到系统预测的均值及协方差, 如下式所示:

$$\bar{Z}(k+1|k) = \sum_{i=0}^{2n} w^{(i)} Z^{(i)}(k+1|k), \quad (25)$$

$$\begin{aligned} P_{Z_k Z_k} &= \\ &\sum_{i=0}^{2n} w^{(i)} [Z^{(i)}(k+1|k) - \bar{Z}(k+1|k)] \times \\ &[Z^{(i)}(k+1|k) - \bar{Z}(k+1|k)]^T + R, \end{aligned} \quad (26)$$

$$\begin{aligned} P_{x_k z_k} &= \\ &\sum_{i=0}^{2n} w^{(i)} [X^{(i)}(k+1|k) - \bar{Z}(k+1|k)] \times \\ &[X^{(i)}(k+1|k) - \bar{Z}(k+1|k)]^T, \end{aligned} \quad (27)$$

其中 R 为观测噪声 $V(k)$ 的协方差.

Step 8: 根据下式计算增益矩阵:

$$K(k+1) = P_{x_k z_k} P_{Z_k Z_k}^{-1}. \quad (28)$$

Step 9: 根据下式计算系统的状态更新和协方差更新:

$$\begin{aligned} \hat{X}(k+1|k+1) &= \\ \hat{X}(k+1|k) &+ K(k+1)[Z(k+1) - \hat{Z}(k+1|k)], \end{aligned} \quad (29)$$

$$\begin{aligned} P(k+1|k+1) &= \\ P(k+1|k) &- K(k+1|k)P_{Z_k Z_k}K^T(k+1). \end{aligned} \quad (30)$$

通过建模可以清晰地了解到UKF算法以及基于和声差分进化的UKF算法的过程. 可以看到, 基于和声差分进化的UKF算法有以下优点: 1) 在处理非线性滤波时, 在估计点不需要作 Taylor 级数展开; 2) 在 UT 变换时, 缩放比例参数 λ 不再是固定值, 而是由它的待选参数 κ 决定; 3) 使用和声差分进化选择 κ 的最优的值.

3 基于和声差分进化的UKF改进算法在目标跟踪中的应用

假设质点 M 在 x - y 内作匀加速直线运动, 其 k 时刻的位置、速度和加速度由矢量 $X(k) = [x_k, y_k, v_{xk}, v_{yk}, a_{xk}, a_{yk}]^T$ 表示, 质点在 x 轴方向(水平方向)上作近似匀加速直线运动, 在 y 轴方向(垂直方向)上亦作匀加速直线运动. 假设雷达位置为 (x_0, y_0) , 对质点 M 进行跟踪, k 时刻雷达和质点的距离为 l_k , 角度为 a_k .

运动状态方程为

$$X(k+1) = FX(k) + W(k). \quad (31)$$

其中: $W(k)$ 为系统噪声; F 为状态转移矩阵,

$$F = \begin{bmatrix} 1 & 0 & T & 0 & \frac{T^2}{2} & 0 \\ 0 & 1 & 0 & T & 0 & \frac{T^2}{2} \\ 0 & 0 & 1 & 0 & T & 0 \\ 0 & 0 & 0 & 1 & 0 & T \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad (32)$$

T 为采样时间.

观测方程为

$$Z(k) = \begin{bmatrix} l(k) + V_r(k) \\ \alpha(k) + V_\alpha(k) \end{bmatrix} = \begin{bmatrix} \sqrt{(x(k) - x_0)^2 + (y(k) - y_0)^2} + V_r(k) \\ \arctan \frac{y(k) - y_0}{x(k) - x_0} + V_\alpha(k) \end{bmatrix}, \quad (33)$$

其中 $V_r(k)$ 和 $V_\alpha(k)$ 分别为雷达的距离观测噪声和角度观测噪声.

将系统状态方程和观测方程代入基于差分进化的UKF算法模型中, 实现对质点的目标跟踪.

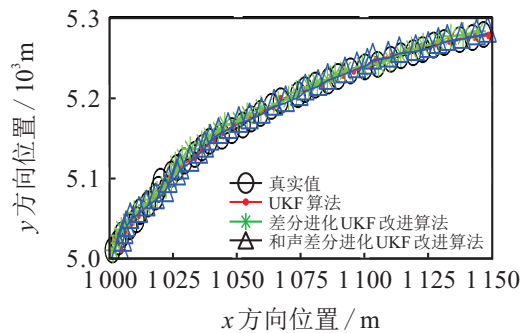
4 目标跟踪仿真实验

为了评估基于和声差分进化的UKF改进算法的精确性, 本文进行了3组实验: 分别使用UKF算法、基于差分进化的UKF改进算法和基于和声差分进化的UKF算法计算位置估计值及位置偏差、速度估计值及速度偏差和加速度估计值及加速度偏差, 并进行对比.

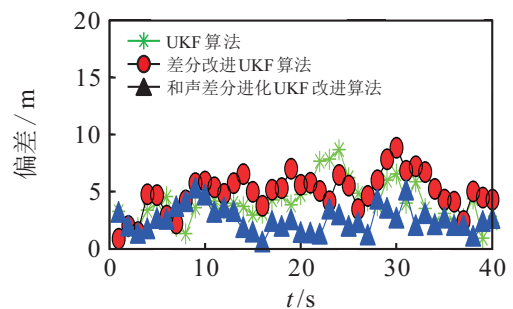
为了验证算法的可行性和有效性, 根据文献[6]来设定实验的数据, 这些数据是可靠的、完整的, 具有一定的说服力. 设定算法的初始状态为 $X(0) = [1000, 5000, 10, 50, 2, -4]$, 采样周期 $T = 0.2\text{s}$, 采样点的分布状态控制参数 $\alpha = 2e^{-2}$, 非负权系数

$\beta = 2$, 压缩比例因子 $f = 0.5$, 交叉概率 $cr = 0.2$.

图1是各算法对目标位置跟踪以及位置偏差的仿真对比图. 图1(a)是真实轨迹与3种算法对目标跟踪的轨迹对比图. 由图1(a)可知, 在 x 方向位置为 $1000 \sim 1050\text{m}$ 时, UKF算法跟踪到的目标轨迹明显地偏离真实轨迹; 在 x 方向位置为 $1050 \sim 1100\text{m}$ 时, UKF算法跟踪到的目标轨迹偏离真实轨迹; 基于和声差分进化的UKF改进算法跟踪到的目标轨迹与目标的真实轨迹基本吻合. 图1(b)是3种算法得到的目标跟踪的位置偏差对比图. 由图1(b)可知, 虽然基于和声差分进化的UKF算法得到的位置偏差偶尔比UKF算法、基于差分进化的UKF改进算法得到的位置偏差大, 但是从整体效果看, 基于和声差分进化的UKF改进算法效果更好, 位置偏差整体上波动比较小, 更加稳定. 由此可见, 基于和声差分进化的UKF改进算法得到的目标跟踪效果更好, 精度较高, 偏差相对较小, 这验证了基于和声差分进化的UKF改进算法的可行性、有效性和稳定性.



(a) 位置估计值

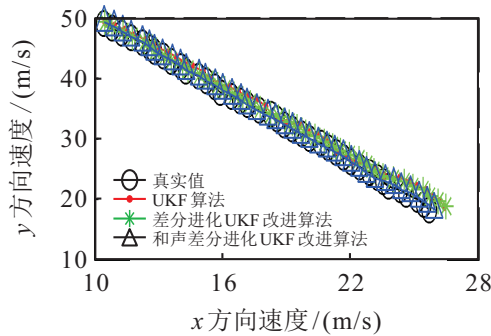


(b) 位置偏差

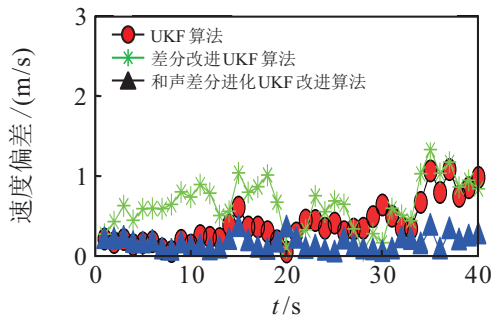
图1 位置估计值及位置偏差对比

图2是各算法对目标速度以及速度偏差的仿真对比图. 图2(a)中, 在 x 方向速度为 $22 \sim 26\text{m/s}$ 时, UKF算法和基于差分进化的UKF改进算法跟踪到的速度明显偏离目标的真实速度, 且随着时间的增加偏离的越明显; 而基于和声差分进化的UKF改进算法跟踪得到的速度和真实的速度基本一致. 由图2(b)可知, 基于和声差分进化的UKF改进算法得到速度

估计偏差很小而且很稳定,而UKF算法和基于差分进化的UKF改进算法得到的速度估计偏差波动大、不稳定,且随着时间的推移逐步增大,这验证了基于和声差分进化的UKF改进算法的优越性和稳定性。

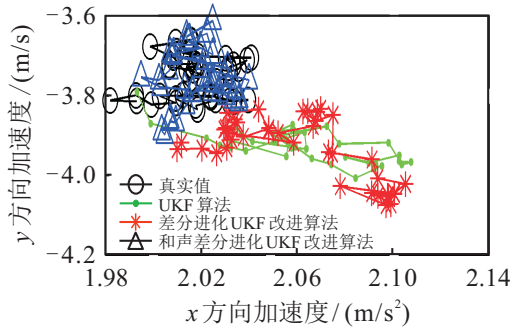


(a) 速度估计值

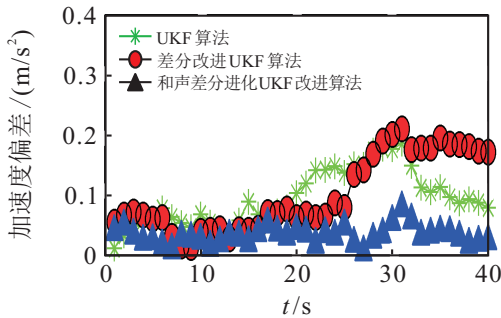


(b) 速度偏差

图2 速度估计值及速度偏差对比



(a) 加速度估计值



(b) 加速度偏差

图3 加速度估计值及加速度偏差对比

图3是各算法对目标的加速度以及加速度偏差的仿真对比图。图3(a)中,基于和声差分进化的UKF改进算法得到的加速度和目标的真实加速度相差较

小,而UKF算法和基于差分进化的UKF算法得到的加速度与真实加速度相差很大。图3(b)表明:基于和声差分进化的UKF改进算法得到的加速度估计偏差很稳定,偏差值很小,而且不会随着时间的推移而增大;UKF算法和基于差分进化的UKF改进算法得到的加速度偏差值比较大,不稳定,且随着时间的推移偏差值逐渐增大。实验表明,基于和声差分进化的UKF改进算法得到的加速度更加准确,接近于真实的加速度,偏差小,随着时间的变化偏差值变化小且稳定。

综合3组实验的结果可以看出,基于和声差分进化的UKF改进算法提高了滤波精度,使目标跟踪更加精确,不管是位置、速度还是加速度的滤波精度都有很大提高,为定位跟踪起到很好的铺垫作用。在目标跟踪中的成功应用验证了此算法的可行性、有效性和优越性。

5 结论

本文根据UKF算法的缺点,提出了基于和声差分进化的无迹卡尔曼滤波改进算法,此算法主要是在UKF的无迹变换中使用和声差分进化算法对其参数进行最优化选择。因为算法的参数是最优参数,所以其目标跟踪精确度大幅度提高。为了验证所提出算法的精确度,将基于和声差分进化的UKF改进算法应用到目标跟踪中,进行了3组仿真实验。实验结果表明,相比于UKF算法、基于差分进化的UKF改进算法,基于和声差分进化的UKF改进算法在位置、速度和加速度中精确度更高,稳定性更强,由此可见本文所提出的算法可行、有效。

参考文献(References)

- [1] Soken HE, Hajiyev C. UKF-based reconfigurable attitude parameters estimation and magnetometer calibration[J]. IEEE Trans on Aerospace Electronic Systems, 2012, 48(3): 2614-2627.
- [2] Li W, Wang J, Lu L, et al. A novel scheme for DVL-aided SINS in-motion alignment using UKF techniques[J]. Sensors, 2013, 13(1): 1046.
- [3] Qi J, Jiang Z, Zhao X, et al. Adaptive UKF and its application in fault tolerant control of rotorcraft UAV[C]. AIAA Guidance, Navigation, and Control Conf. Hilton Head Island, 2013: 43-57.
- [4] Xu W, Bao S, Liu Z. The state estimation of UAV based on UKF[C]. IEEE Workshop on Advanced Research and Technology in Industry Applications. Ottawa: IEEE, 2014: 402-405.
- [5] 林菜, 黄超. 采用UKF算法估计路面附着系数[J]. 哈

- 尔滨工业大学学报, 2013, 45(7): 115-120.
(Lin F, Huang C. Unscented Kalman filter for road friction coefficient estimation[J]. J of Harbin Institute of Technology, 2013, 45(7): 115-120.)
- [6] 黄小平, 王岩. 卡尔曼滤波原理及应用——Matlab仿真[M]. 北京: 电子工业出版社, 2015: 77-102.
(Huang X P, Wang Y. Kalman filtering principle and Application — Matlab simulation[M]. Beijing: Publishing House of Electronics Industry, 2015: 77-102.)
- [7] Julier S J, Uhlmann J K, Durrant-Whyte H F. A new approach for the nonlinear transformation of means and covariances in filters and estimators[J]. IEEE Trans on Automatic Control, 2000, 45(3): 477-482.
- [8] 欧县华, 武宪青, 何熊熊. 基于AIUKF的WSN节点定位算法[J]. 传感技术学报, 2015(2): 234-238.
(Ou X H, Wu X Q, He X X. Adaptive iterated UKF based node localization for WSN[J]. Chinese J of Sensors Actuators, 2015(2): 234-238.)
- [9] Lin T, Liu Y A. Application of improvement UKF algorithm in over-the-horizon radar[J]. Computer Simulation, 2014, 36(6): 6-9.
- [10] Meng F C, Wang S, Tao L, et al. QR-UKF fusion of satellite selection and RAIM for GNSS performance enhancement in multi-constellation[C]. Proc of the 28th Int Technical Meeting of The Satellite Division of the Institute of Navigation. Tampa, 2015: 764-776.
- [11] Huang X C, Lin R W. SRMA-UKF-based design of the sensorless system for permanent magnet synchronous motors[J]. Electrical Automation, 2014, 36(4): 4-6.
- [12] Dunik J, Simandl M, Straka O. Unscented Kalman filter: Aspects and adaptive setting of scaling parameter[J]. IEEE Trans on Automatic Control, 2012, 57(9): 2411-2416.
- [13] 金瑶, 蔡之华, 梁丁文. 基于差分演化算法的自适应无迹卡尔曼滤波[J]. 电子与信息学报, 2013(4): 838-843.
(Jin Y, Cai Z H, Liang D W. Adaptive unscented Kalman filter based on differential evolution algorithm[J]. J of Electronics Information Technology, 2013(4): 838-843.)
- [14] 王凌, 钱斌. 混合差分进化与调度算法[M]. 北京: 清华大学出版社, 2012: 1-14.
(Wang L, Qian B. Hybrid differential evolution and scheduling algorithm[M]. Beijing: Tsinghua University Press, 2012: 1-14.)
- [15] Ali M M. Differential evolution with preferential crossover[J]. European J of Operational Research, 2007, 181(3): 1137-1147.
- [16] Torn R, Price K. Minimizing the real functions of the ICEC'96 contest by differential evolution[J]. Proc of IEEE Int Conf on Evolutionary Computation. Nagoya: IEEE, 1996: 842-844.
- [17] Wang L, Huang F Z. Parameter analysis based on stochastic model for differential evolution algorithm[J]. Applied Mathematics Computation, 2010, 217(7): 3263-3273.
- [18] 薛亚娣. 和声搜索算法综述[J]. 现代妇女(理论前沿), 2014(3): 197.
(Xue Y D. Summarize of harmony search algorithms[J]. Modern Women, 2014(3): 197.)
- [19] 达列雄, 刘杰. 和声搜索算法在油田事故应急资源筹集中的应用[J]. 电子测试, 2014(2): 19-20.
(Da L X, Liu J. Harmony search algorithm application in oilfield accident emergency resource mobilization[J]. Electronic Test, 2014(2): 19-20.)
- [20] Al-Betar M A, Khader A T. A harmony search algorithm for university course timetabling[J]. Annals of Operations Research, 2012, 194(1): 3-31.
- [21] 邹德旋, 高立群, 吴建华, 等. 混合差分进化-和声搜索算法在结构工程中的应用[J]. 东北大学学报: 自然科学版, 2010, 31(6): 769-772.
(Zou D X, Gao L Q, Wu J H, et al. Application of HDEHS algorithm to structural engineering[J]. J of Northeastern University: Natural Science, 2010, 31(6): 769-772.)
- [22] 雍龙泉. 和声搜索算法研究进展[J]. 计算机系统应用, 2011, 20(7): 244-249.
(Yong L Q. Advances in harmony search algorithm[J]. Computer Systems Applications, 2011, 20(7): 244-249.)
- [23] 欧阳海滨, 高立群, 邹德旋, 等. 和声搜索算法探索能力研究及其修正[J]. 控制理论与应用, 2014, 31(1): 57-65.
(Ouyang H B, Gao L Q, Zou D X, et al. Exploration ability study of harmony search algorithm and its modification[J]. Control Theory & Applications, 2014, 31(1): 57-65.)

(责任编辑: 齐 霖)