



DOI:10.12404/j.issn.1671-1815.2307849

引用格式: 刘建娟, 李志伟, 姬森鑫, 等. 多策略改进麻雀搜索算法优化无迹卡尔曼滤波方法[J]. 科学技术与工程, 2025, 25(1): 227-237.

Liu Jianjuan, Li Zhiwei, Ji Miaoxin, et al. Multi-strategy improvement of the sparrow search algorithm for optimizing the UKF method[J]. Science Technology and Engineering, 2025, 25(1): 227-237.

自动化技术、计算机技术

多策略改进麻雀搜索算法优化无迹卡尔曼滤波方法

刘建娟^{1,2}, 李志伟^{1,2}, 姬森鑫^{1,2}, 吴豪然^{1,2}, 许强伟^{1,2}

(1. 河南工业大学电气工程学院, 郑州 450001; 2. 河南工业大学机电设备及测控技术研究所, 郑州 450001)

摘要 针对无迹卡尔曼滤波(unscented Kalman filter, UKF)中无迹变换(unscented transform, UT)在状态估计时采样点分布状态控制参数异常对滤波性能的影响问题,提出了一种利用多策略改进麻雀搜索算法(improved sparrow search algorithm, ISSA)对UT中采样点分布状态控制参数进行寻优调整的方法,从而优化Sigma点分布以提高非线性近似效果,改善滤波估计性能。同时针对传统麻雀搜索算法面临的易陷入局部最优和收敛速度慢等问题,首先利用Cubic混沌映射改善初始种群的多多样性;其次在发现者阶段引入非线性自适应收敛因子,提高平衡算法在全局探索和局部开发方面的能力;同时在追随者阶段利用小波变异策略,以避免追随者盲目追随而导致算法陷入局部最优;最后利用自适应 t 分布的扰动能力增强算法的全局搜索能力。通过测试函数对ISSA算法进行仿真实验,结果表明ISSA算法具有更好的收敛性和求解精度,同时验证ISSA优化UKF算法后的仿真结果,表明了ISSA-UKF算法相比于UKF算法的位置均方根误差降低了52.2%,速度均方根误差降低了21.9%,证明了改进方法的有效性和可行性。

关键词 无迹卡尔曼滤波; 麻雀搜索算法; Cubic混沌映射; 非线性自适应收敛因子; 小波变异策略

中图分类号 TP18 TN713; **文献标志码** A

Multi-strategy Improvement of the Sparrow Search Algorithm for Optimizing the UKF Method

LIU Jian-juan^{1,2}, LI Zhi-wei^{1,2}, JI Miao-xin^{1,2}, WU Hao-ran^{1,2}, XU Qiang-wei^{1,2}

(1. School of Electrical Engineering, Henan University of Technology, Zhengzhou 450001, China;

2. Institute of Electromechanical Equipment and Measurement & Control Technology, Henan University of Technology, Zhengzhou 450001, China)

[Abstract] A method for optimizing the control parameters of the sample point distribution state within the framework of the unscented transform (UT) for the unscented Kalman filter (UKF) was introduced. The issue of abnormal filtering performance arising from the state of sample point distributions was addressed by this method. A multi-strategy improved sparrow search algorithm (ISSA) was employed to finely tune the control parameters. The goal is to enhance the distribution of Sigma points, thereby improving the effectiveness of nonlinear approximations and ultimately enhancing the accuracy of filtering estimations. To address the shortcomings of traditional sparrow search algorithms, several refinements were implemented. Initially, a Cubic chaotic mapping was applied to diversify the initial population. Furthermore, during the exploration phase, a nonlinear adaptive convergence factor was introduced to balance the algorithm's capacity for global exploration and local exploitation. Additionally, a wavelet mutation strategy was integrated into the follower phase to prevent blind adherence to specific paths and mitigate the risk of becoming trapped in local optima. Lastly, an adaptive t -distribution perturbation capability was introduced to strengthen the algorithm's ability to perform wide-ranging global searches. The efficacy of the proposed ISSA was demonstrated through simulation experiments conducted on various test functions. The results consistently show that ISSA outperforms other methods in terms of convergence and solution accuracy. Furthermore, the benefits of ISSA are extended to the optimization of parameters within the UKF algorithm. Experimental outcomes indicate that the ISSA-UKF algorithm reduces the root mean square error (RMSE) of position by 52.2% and the RMSE of velocity by 21.9%, thus affirming the viability and effectiveness of the proposed enhancements.

[Keywords] unscented Kalman filter; sparrow search algorithm; Cubic chaotic mapping; nonlinear adaptive convergence factor; wavelet mutation strategy

收稿日期: 2023-10-09 修订日期: 2024-07-04

基金项目: 国家自然科学基金(62201199); 河南省科技攻关项目(232102320037); 河南工业大学自主创新基金(2021ZKCJ07)

第一作者: 刘建娟(1978—), 女, 汉族, 河南南阳人, 博士, 教授。研究方向: 智能感知与导航定位。E-mail: liujianhau@163.com。

投稿网址: www.stae.com.cn

随着无人驾驶技术的快速发展,如何克服复杂环境的影响成为确保智能汽车安全的重要前提^[1]。为了准确、实时地获取行驶状态的相关信息,目前常采用多传感器融合方法,将来自不同传感器的数据或信息进行集成和处理,以获得更全面、可靠的信息。多传感器融合定位依赖于各种滤波算法,其中常见的滤波算法包括卡尔曼滤波^[2](Kalman filter, KF)、扩展卡尔曼滤波^[3](extended Kalman filter, EKF)和无迹卡尔曼滤波^[4](unscented Kalman filter, UKF)。这些算法已被应用于目标跟踪、组合导航以及锂电池荷电状态估计等多个领域,并成为国内外学者研究的焦点之一^[5-7]。

UKF 针对非线性系统主要通过使用无迹变换(unscented transformation, UT)方法采用特定采样点来近似非线性函数,相比于 EKF 算法,UKF 算法在非线性的估计和滤波中具有更好的性能。Zheng 等^[8]针对同时存在异常值和不确定协方差矩阵的情况,提出了一种基于创新和残差的鲁棒自适应 UKF 滤波方法;Yang 等^[9]为提高全球卫星导航系统与惯性测量单元组合导航系统的鲁棒性,提出一种基于广义极大似然估计的鲁棒无迹卡尔曼滤波器,减弱了动态模型和测量异常值干扰的影响;Li 等^[10]提出一种鲁棒自适应 UKF 算法来解决非线性和非高斯问题,该方法能自适应估计非高斯和非线性情况下的过程噪声协方差和测量噪声协方差;Novi 等^[11]提出一种集成人工神经网络和无迹卡尔曼滤波的滤波器,用以估算车辆的侧滑角;Yang 等^[12]提出了一种改进的自适应衰减平方根无迹卡尔曼滤波器,通过利用衰减因子实时调整滤波器增益,改善算法对突变状态的适应性;Dunik 等^[13]提出一种自适应的取值方法,利用网格方法或随机搜索方法对缩放参数进行选择,缩放参数的选择与网格的密度密切相关,网格密度越小,能够搜索到的缩放参数数量就越少,网格密度越大,搜索到的缩放参数就越多;贺军义等^[14]提出一种利用和声差分进化算法对缩放参数进行最优选择的方法,将缩放参数作为优化目标,选择每时刻滤波误差最小的缩放参数。文献[8-12]的研究主要聚焦在 UKF 算法的噪声协方差方面,以及通过引入衰减因子改进其性能,然而在 UKF 算法中的 UT 变换理论的研究较为有限,其特点是通过缩放参数控制采样点的分布来影响近似的准确性,控制参数过小将使采样点过于集中在均值附近,导致滤波器低估状态的不确定性,使其对噪声和不确定性的抵抗能力下降;控制参数过大使得采样点分散太远,导致滤波器高估状态的不确定性,使其对噪声和不确定性的抵抗能力变强,但会导致估计值波动较大,影响滤波精度。文献[13]采用网格搜

索方法对缩放参数进行选择,但网格搜索方法计算成本高同时可能错过最优解;文献[14]利用和声差分进化算法对缩放参数进行优化选择,但其对初始解较为敏感且存在早熟等问题。

针对上述问题,现提出一种利用麻雀搜索算法改进 UKF 的方法。麻雀搜索算法具有全局搜索能力强和易于实现等优势,但同时存在收敛速度慢和易陷入局部最优等问题,从而可能出现错过最优解的情况,因此,现提出利用 Cubic 混沌映射、非线性自适应收敛因子、小波变异策略以及自适应 t 分布变异方法对其进行改进,然后利用改进后的麻雀搜索算法对采样点分布状态控制参数 α 进行优化选择,从而控制 UT 变换中采样点的分布,以提高 UKF 算法的性能和估计精度,最后通过实验进一步验证改进方法的可行性和有效性。

1 麻雀搜索算法及其改进

1.1 麻雀搜索算法

麻雀搜索算法^[15](sparrow search algorithm, SSA)主要灵感来自麻雀种群寻找食物和反掠食者捕食行为。该算法将麻雀按照不同职责分为发现者、追随者和预警者三大类,以实现高效的优化搜索。在 SSA 算法中,发现者负责寻找食物,并提供觅食区域和方向的信息,而追随者则依据发现者提供信息对食物进行获取,当预警者发现捕食者时会发出警报,此时位于边缘的个体会迅速飞往安全区域,即反捕食行为。值得注意的是,发现者和追随者可以相互转变身份,如果有更优的食物位置被发现,每只麻雀都可能是发现者,然而,在整个种群中,发现者和追随者的比重是不变的。

(1) 发现者的位置更新计算。

$$X_{i,j}^{t+1} = \begin{cases} X_{i,j}^t \exp\left(\frac{-i}{\sigma \text{iter}_{\max}}\right), & R_2 < ST \\ X_{i,j}^t + QL, & R_2 \geq ST \end{cases} \quad (1)$$

式(1)中: $X_{i,j}^t$ 为第 i 只麻雀在第 t 代麻雀种群中第 j 维的位置; σ 为 $(0,1]$ 中的随机数; Q 为正态分布随机数; R_2 为报警值,是取值范围 $[0,1]$ 中的随机数; ST 为预警阈值,取值范围为 $[0.5,1]$; L 为一个所有元素都为 1 的 $1 \times d$ 矩阵。 $R_2 < ST$ 表示当前位置没有捕食者,发现者可以继续觅食; $R_2 \geq ST$ 表示当前区域有捕食者,发现者需躲避危险,转移到安全区域。

(2) 追随者的位置更新计算。

$$X_{i,j}^{t+1} = \begin{cases} Q \exp\left(\frac{X_{\text{worst}}^t - X_{i,j}^t}{l^2}\right), & i > n/2 \\ X_p^{t+1} + |X_{i,j}^t - X_p^{t+1}| A^+ L, & \text{其他} \end{cases} \quad (2)$$

式(2)中: X_p^{t+1} 为麻雀种群第 $t+1$ 代的最优位置; X_{worst}^t 为第 t 代麻雀种群的最差位置; A^+ 为 $1 \times d$ 的矩阵, 矩阵内各元素被随机赋值为 1 或 -1, 且 $A^+ = A^T (AA^T)^{-1}$ 。当 $i > n/2$ 时, 麻雀没有得到食物, 需要寻找一个新的位置去觅食; 当 $i \leq n/2$ 时, 麻雀移动到当前最佳位置以获得更多的食物。

(3) 预警者的位置更新计算。

$$X_{i,j}^{t+1} = \begin{cases} X_{\text{best}}^t + \varphi |X_{i,j}^t - X_{\text{best}}^t|, & f_i > f_g \\ X_{i,j}^t + K \frac{|X_{i,j}^t - X_{\text{worst}}^t|}{f_i - f_w + \varepsilon}, & f_i = f_g \end{cases} \quad (3)$$

式(3)中: X_{best}^t 为麻雀种群第 t 代最优位置; φ 为符合标准正态分布的随机数; K 为 $[-1, 1]$ 的均匀随机数; f_i 为当前个体的适应度值; f_g 和 f_w 分别为全局最优和最差适应度值; ε 为最小常数, 避免分母为零。 $f_i > f_g$ 表示此麻雀处于觅食区域边缘位置, 易受到捕食者袭击; $f_i = f_g$ 表示处于中间位置的麻雀意识到危险, 通过向其他麻雀靠拢减少被捕食风险。

1.2 多策略改进麻雀搜索算法 (ISSA)

为了克服 SSA 算法收敛速度慢和易陷入局部最优等问题, 提出了一种多策略改进麻雀搜索算法, 算法首先引入 Cubic 混沌映射改善初始种群的多样性, 其次在发现者中引入非线性自适应收敛因子, 使算法在前期拓展搜索区域, 在后期提升收敛效率, 协调算法整体与局部的开发拓展能力; 同时在追随者引入小波变异策略, 降低对发现者的依赖性, 避免陷入局部最优; 最后, 利用自适应 t 分布方法, 对个体进行扰动, 以增强算法跳出局部最优解的能力。通过以上改进策略的融合, 使得改进后的麻雀搜索算法能有效提高算法性能, 改善求解的效率和精度。

1.2.1 Cubic 混沌映射

混沌映射是一种自然界中存在的非线性现象, 被广泛应用于各种优化算法。它的存在丰富了种群的多样性, 并且由于其具有随机性和遍历性, 使得算法能有效跳出局部最优解。其中, Cubic 是一种典型的混沌映射^[16], 其标准形式为

$$x_{n+1} = bx_n^3 - cx_n \quad (4)$$

式(4)中: b 和 c 为混沌影响因子。一般在 $c \in (2.3, 3)$ 时, Cubic 映射产生的序列处于混沌状态。通过文献[17]对 16 种常见一维混沌映射的最大 Lyapunov 指数计算和分析, 可以发现 Cubic 映射的混沌性与 Logistic 映射、Tent 映射的最大 Lyapunov 指数相近, 且相比于 Sine 映射、Circle 映射和 Singer 映射等一维映射具有很好的优势。修正后的 Cubic 映射特定表达式为

$$x_{n+1} = \rho x_n (1 - x_n^2) \quad (5)$$

式(5)中: $x_n \in (0, 1)$; ρ 为控制参数。Cubic 映射仿真结果如图 1 所示。

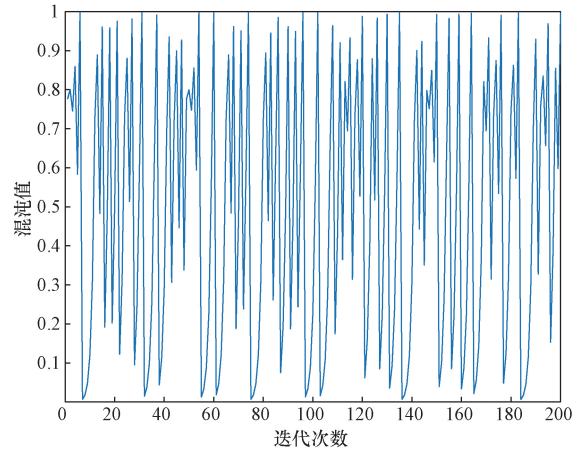


图 1 Cubic 混沌映射分布图

Fig. 1 Distribution of Cubic chaotic mapping

1.2.2 非线性自适应收敛因子

在麻雀捕食过程中, 发现者在搜寻最佳觅食位置时, 如果移动步长过大, 尽管可以缩短寻找最优解的时间, 但也存在容易错失全局最优解的风险。为了提升 SSA 算法的全局搜索性能, 发现者在早期迭代阶段应执行全局勘探, 这需要较大的收敛因子来扩大全局搜索范围; 而在后期迭代阶段, 发现者需较小的收敛因子, 以增强局部开发能力, 加快收敛速度, 并避免陷入局部最优。为了解决上述问题, 受灰狼优化算法启发, 引入了一个系数 A , 当 $|A| > 1$ 时, 灰狼群体会扩大搜索范围以寻找猎物, 即进行全局搜索, 从而加快收敛速度; 当 $|A| < 1$ 时, 灰狼群体会收缩搜索范围以攻击猎物, 即进行局部开发, 且收敛速度较慢, 因此, 系数 A 成为影响全局探索和局部开发能力平衡的关键因素^[18]。然而, 从式(6)可看出, 收敛因子 a 决定了 A 的取值大小, 但收敛因子是线性递减的, 无法准确反映复杂的非线性搜索过程。因此引入一种逆不完全 Γ 函数和符合 beta 分布的收敛因子, 以提高算法的搜索性能, 其具体表达式如式(7)所示。

$$A = 2ar - a \quad (6)$$

$$a = a_{\min} + \frac{a_{\max} - a_{\min}}{\xi} \Gamma\left(\xi, 1 - \frac{t}{\text{iter}_{\max}}\right) + \mu \text{betarnd}(p, q) \quad (7)$$

式(7)中: r 为区间 $[0, 1]$ 上的随机数; a_{\max} 、 a_{\min} 分别为收敛因子 a 的最大值和最小值; 逆不完全 Γ 函数为 $\gamma(\xi, a) = \int_0^{\xi} e^{-x} x^{a-1} dx$, ξ 为随机变量, 取 0.01; $\mu = 0.1$; $\text{betarnd}(p, q)$ 为服从贝塔分布的随机数, $p = 1$, $q = 2$ 。收敛因子 a 的变化曲线图如图 2 所示。

根据图 2 所示, 改进后的非线性收敛因子在迭代前期表现类似于线性递减, 这有利于算法进行全局探索; 而在算法迭代后期, 收敛因子开始呈指数形式下

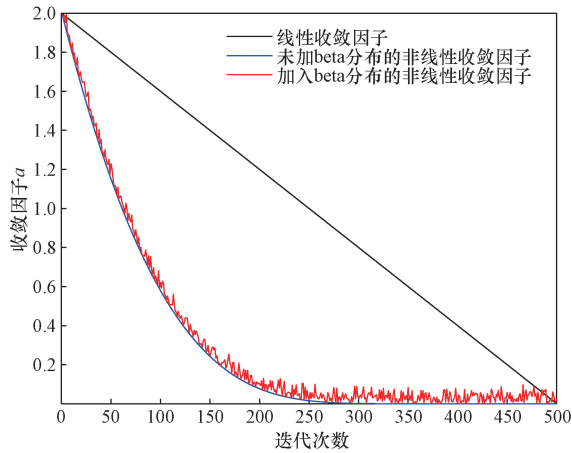


图2 非线性收敛因子曲线图

Fig. 2 Plot of nonlinear convergence factor

降,从而增强了算法的局部寻优能力。同时,引入了符合 beta 分布的随机数对非线性收敛因子进行局部扰动,实现了非线性因子的动态变化,意味着扰动了麻雀搜索步伐大小,提高了解的多样性。因此,改进后的非线性收敛因子有效协调了算法的全局探索和局部开发能力,提升了算法的优化性能。

改进后发现者的位置更新计算如下。

$$X_{i,j}^{t+1} = \begin{cases} AX_{i,j}^t \exp\left(\frac{-i}{\sigma_{\text{iter}_{\max}}}\right), & R_2 < ST \\ X_{i,j}^t + QL, & R_2 \geq ST \end{cases} \quad (8)$$

1.2.3 小波变异策略

标准 SSA 算法的追随者的更新方式是以占据最优位置的发现者为目标而进行位置更新,当追随者向最优位置靠近时,种群容易在短时间内迅速聚集,这种情况虽然实现了快速收敛,但也极大地增加了算法陷入局部最优的概率^[19]。为了让追随者更好跟随发现者进行位置更新,在追随者的更新方式中引入小波变异系数,这使追随者在后期能以较大的步长移动,防止因盲目追随发现者而错过更好的位置,通过有效地避免趋同性,算法可以改善过早收敛和陷入局部最优的问题。

小波变异的思想来源于小波函数,其主要优势在于能够根据迭代次数来调控振幅值,实现对扰动的动态微调效果,从而增强算法跳出局部最优的能力^[20]。小波变异的表达式为

$$\eta = \frac{1}{\sqrt{\tau}} e^{-\left(\frac{\varphi}{\tau}\right)^2} \cos\left(5 \frac{\varphi}{\tau}\right) \quad (9)$$

式(9)中: φ 为 $[-2.5\tau, 2.5\tau]$ 内的随机数,伸缩参数 τ 随着迭代次数的增加而减小,其作为分母反向调控变异系数,使得小波变异系数逐渐减小,从而能够自适应地调控种群的变异程度。

$$\tau = e^{-\ln g \left(1 - \frac{i}{\text{iter}_{\max}}\right)^{\delta} + \ln g} \quad (10)$$

式(10)中: δ 为形状参数,取 $\delta = 0.5$; g 为 τ 的上限,取 $g = 100$ 。

改进后追随者的位置更新计算如下。

$$X_{i,j}^{t+1} = \begin{cases} Q \exp\left(\frac{X_{\text{worst}}^t - X_{i,j}^t}{t^2}\right), & i > n/2 \\ \eta X_p^{t+1} + |X_{i,j}^t - X_p^{t+1}| A^+ L, & \text{其他} \end{cases} \quad (11)$$

1.2.4 自适应 t 分布

t 分布又称为学生分布,其曲线形态与自由度 n 密切相关,相比于正态分布,当自由度较小时,曲线较平坦,如果曲线峰值位置较低,则曲线两侧的尾部位置较高;当自由度为正无穷时,则逐渐接近标准正态分布^[21]。高斯分布和柯西分布是 t 分布边界处的两个特例分布, t 分布既有高斯分布的特点,也有柯西分布的特点,其概率密度函数为

$$p_t(x) = \frac{\Gamma\left(\frac{n+1}{2}\right)}{\sqrt{n\pi}\Gamma\left(\frac{n}{2}\right)} \left(1 + \frac{x^2}{n}\right)^{-\frac{n+1}{2}} \quad (12)$$

式(12)中: $\Gamma\left(\frac{n+1}{2}\right) = \int_0^{+\infty} x^{\frac{n+1}{2}-1} e^{-x} dx$ 为第二类欧拉积分。

为了提高传统麻雀算法的全局搜索能力并避免过早收敛,在发现者、加入者和预警者的位置更新之后,引入了自适应 t 分布的变异能力来扰动麻雀的位置。而自适应 t 分布的变异比柯西分布和高斯分布在扰动能力上更强,通过自适应 t 分布的变异扰动,可以有效提高麻雀算法的寻优性能,具体位置更新方式为

$$X_i^{t+1} = X_i^t + X_i^t t(\text{iter}) \quad (13)$$

式(13)中: X_i^{t+1} 为扰动后麻雀位置; X_i^t 为麻雀 i 在第 t 次迭代时的位置。式(13)在 X_i^t 的基础上引入了随机干扰项 $X_i^t t(\text{iter})$,充分利用了当前位置信息,并将自由度参数设置为迭代次数 t 。使算法在前期迭代次数较小时,具有全局探索能力,在算法后期,具有局部开发能力,而在算法中期, t 分布综合了柯西分布和高斯分布两者的优点,从而改进了算法的全局探索和局部开发能力。这种更新方式使得算法能够在不同阶段灵活的调整其探索和开发能力,以便更好地搜索全局最优解并避免陷入局部最优。

1.2.5 ISSA 算法具体步骤

ISSA 算法步骤如下。

步骤 1 初始化 ISSA 算法参数,包括种群大小、迭代次数、发现者比例、预警者比例等,然后使用式(5)初始化麻雀种群的初始位置。

步骤 2 计算各麻雀适应度值,寻找当前最优

适应度值和最差适应度值,以及它们的位置。

步骤3 发现者位置更新。根据式(8)更新发现者位置。

步骤4 追随者位置更新。根据式(11)更新追随者位置。

步骤5 预警者位置更新。随机选择一些麻雀作为预警者,并利用式(3)更新它们位置。

步骤6 利用式(13)进行自适应 t 分布变异,扰动当前最优值,生成新解。

步骤7 更新位置并计算适应度值。

步骤8 判断当前迭代次数是否达到最大迭代次数,若达到,则循环结束,算法输出最优解;否则,返回步骤2,直到满足结束条件。

2 ISSA 优化无迹卡尔曼滤波算法 (ISSA-UKF)

UKF 算法的关键在于 UT 变换,UT 变换中 Sigma 点的统计特性由缩放比例参数控制,通过调整缩放比例参数,可以选择合理的采样点分布状态控制参数,从而有效影响缩放比例参数,以改善 UKF 算法的滤波性能。

为了找到预测误差最小的分布状态控制参数,需设定正确的适应度函数,因此选择以实际值与滤波输出估计值的均方误差作为适应度函数,适应度函数表达式为

$$f_k = \frac{1}{M} \sum_{k=1}^M (\vartheta_k - \hat{\theta}_k)^2 \quad (14)$$

式(14)中: f_k 为第 k 次实际值与估计值的均方误差; M 为估计长度; ϑ_k 为实际值; $\hat{\theta}_k$ 为滤波输出估计值。

当适应度函数值越小时,均方误差越小,则ISSA算法寻找的采样点分布状态控制参数最优。算法前期设置初始的采样点分布状态控制参数,然后通过ISSA算法的迭代搜索过程,生成和调节具有最小适应度函数值的采样点分布状态控制参数,即最优的采样点分布状态控制参数,将其作为输入参数应用于UKF算法,从而改善UKF算法的性能。通过自适应更新采样点分布状态控制参数,UKF算法可以更准确地估计系统状态,并提高滤波精度。

对于一般的非线性离散动态系统,过程和测量模型可以描述如下。

$$\begin{cases} \mathbf{X}_k = \mathbf{f}(\mathbf{X}_{k-1}) + \mathbf{W}_{k-1} \\ \mathbf{Z}_k = \mathbf{h}(\mathbf{X}_k) + \mathbf{V}_k \end{cases} \quad (15)$$

式(15)中: $\mathbf{X}_k \in \mathbf{R}^n$ 为状态向量; $\mathbf{Z}_k \in \mathbf{R}^m$ 为测量向量; $\mathbf{f}(\cdot)$ 和 $\mathbf{h}(\cdot)$ 分别为已知的非线性状态转移函数和测量函数; \mathbf{W}_{k-1} 和 \mathbf{V}_k 为不相关的零均值高斯白噪声,其协方差分别为 \mathbf{Q}_{k-1} 和 \mathbf{R}_k 。算法的实现过程如下。

步骤1 初始化。

$$\begin{cases} \hat{\mathbf{X}}_0 = E[\mathbf{X}_0] \\ \mathbf{P}_0 = E[(\mathbf{X}_0 - \hat{\mathbf{X}}_0)(\mathbf{X}_0 - \hat{\mathbf{X}}_0)^T] \end{cases} \quad (16)$$

式(16)中: $\hat{\mathbf{X}}_0$ 为初始状态; \mathbf{P}_0 为初始误差协方差。

步骤2 Sigma点计算。

将通过ISSA算法寻优后获得的最优采样点分布状态控制参数 α 代入缩放比例参数计算公式。

$$\begin{cases} \mathbf{x}_{k-1}^{(0)} = \hat{\mathbf{X}}_{k-1} \\ \mathbf{x}_{k-1}^{(i)} = \hat{\mathbf{X}}_{k-1} + [\sqrt{(n+\lambda)\mathbf{P}_{k-1}}], i = 1, 2, \dots, n \\ \mathbf{x}_{k-1}^{(i)} = \hat{\mathbf{X}}_{k-1} - [\sqrt{(n+\lambda)\mathbf{P}_{k-1}}], i = n+1, n+2, \dots, 2n \end{cases} \quad (17)$$

式(17)中: n 为状态维数; $\lambda = \alpha^2(n + \kappa) - n$ 为缩放比例参数; α 和 κ 为调优参数,通常 α 反映样本点的分布。

步骤3 状态预测。

$$\begin{cases} \mathbf{x}_{k/k-1}^{(i)} = \mathbf{f}[\mathbf{x}_{k-1}^{(i)}, k-1], i = 0, 1, 2, \dots, 2n \\ \hat{\mathbf{X}}_{k/k-1} = \sum_{i=0}^{2n} \omega_i^{(m)} \mathbf{x}_{k/k-1}^{(i)} \\ \mathbf{P}_{XX} = \sum_{i=0}^{2n} \omega_i^{(c)} [\mathbf{x}_{k/k-1}^{(i)} - \hat{\mathbf{X}}_{k/k-1}] [\mathbf{x}_{k/k-1}^{(i)} - \hat{\mathbf{X}}_{k/k-1}]^T \\ \mathbf{P}_{k/k-1} = \mathbf{P}_{XX} + \mathbf{Q}_{k-1} \end{cases} \quad (18)$$

式(18)中: $\omega_i^{(m)}$ 和 $\omega_i^{(c)}$ 为权重,定义为

$$\begin{cases} \omega_0^{(m)} = \frac{\lambda}{n + \lambda} \\ \omega_0^{(c)} = \frac{\lambda}{n + \lambda} + (1 - \alpha^2 + \beta) \\ \omega_i^{(m)} = \omega_i^{(c)} = \frac{1}{2(n + \lambda)}, i = 1, 2, \dots, 2n \end{cases} \quad (19)$$

式(19)中:待选参数 $\beta \geq 0$ 是非负权重系数。

步骤4 测量预测。

$$\begin{cases} \mathbf{z}_{k/k-1}^{(i)} = \mathbf{h}[\mathbf{x}_{k/k-1}^{(i)}, k], i = 0, 1, 2, \dots, 2n \\ \hat{\mathbf{Z}}_{k/k-1} = \sum_{i=0}^{2n} \omega_i^{(m)} \mathbf{z}_{k/k-1}^{(i)} \end{cases} \quad (20)$$

步骤5 增益计算。

$$\begin{cases} \mathbf{P}_{XZ} = \sum_{i=0}^{2n} \omega_i^{(c)} [\mathbf{x}_{k/k-1}^{(i)} - \hat{\mathbf{X}}_{k/k-1}] [\mathbf{z}_{k/k-1}^{(i)} - \hat{\mathbf{Z}}_{k/k-1}]^T \\ \mathbf{P}_{ZZ} = \sum_{i=0}^{2n} \omega_i^{(c)} [\mathbf{z}_{k/k-1}^{(i)} - \hat{\mathbf{Z}}_{k/k-1}] [\mathbf{z}_{k/k-1}^{(i)} - \hat{\mathbf{Z}}_{k/k-1}]^T + \mathbf{R}_k \\ \mathbf{K}_k = \mathbf{P}_{XZ} \mathbf{P}_{ZZ}^{-1} \end{cases} \quad (21)$$

步骤6 滤波更新。

$$\begin{cases} \hat{\mathbf{X}}_k = \hat{\mathbf{X}}_{k/k-1} + \mathbf{K}_k (\mathbf{Z}_k - \hat{\mathbf{Z}}_{k/k-1}) \\ \mathbf{P}_k = \mathbf{P}_{k/k-1} - \mathbf{K}_k \mathbf{P}_{ZZ} \mathbf{K}_k^T \end{cases} \quad (22)$$

步骤7 重复步骤2~步骤6,以实现完整 UKF 算法。ISSA-UKF 算法流程图如图3所示。

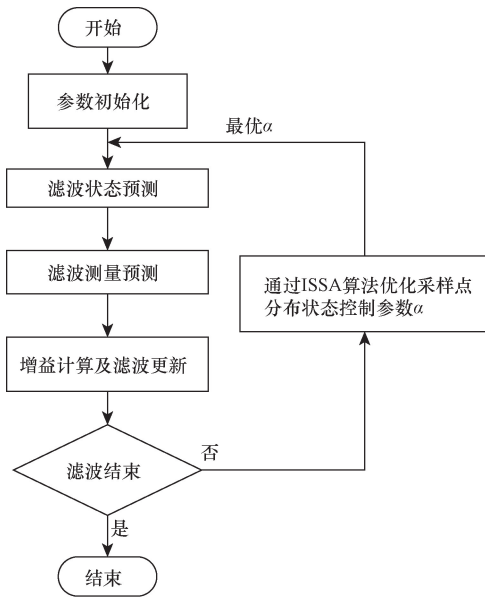


图3 ISSA-UKF 算法流程图

Fig. 3 Flow chart of the ISSA-UKF algorithm

3 仿真验证与分析

3.1 ISSA 算法性能测试

3.1.1 实验设置与测试函数

首先为验证 ISSA 算法相比其他智能优化算法的性能优势和可行性,选取 10 个常见的基准测试函数,将其与 SSA 算法、鲸鱼优化算法 (whale optimization algorithm, WOA)、粒子群优化算法 (particle swarm optimization, PSO) 以及灰狼优化算法 (grey wolf optimizer, GWO) 进行了实验对比。实验仿真平台基于 Windows10 系统, i7-8550U CPU @ 1.80 GHz, 1.99 GHz, 运行内存 8 GB, 仿真软件为 MATLAB R2018b。基准测试函数如表 1 所示。为保证实验的公平性,设置各对比算法的种群数量均为 30,迭代次数为 500,其余参数设置如表 2 所示,在 SSA 算法和 ISSA 算法中,ST 为安全阈值,PD 为发现者比例,SD 为意识到危险的麻雀占总数比例;WOA 算法中, a 为缩放因子;PSO 算法中, ω 为惯性权重, c_1, c_2 为学习因子;GWO 算法中, a 为收敛因子, r_1 和 r_2 为随机向量。

表 1 基准测试函数

Table 1 Benchmark function

函数	维数	范围	最优值
$F_1(x) = \sum_{i=1}^n x_i^2$	30	$[-100, 100]$	0
$F_2(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	30	$[-10, 10]$	0
$F_3(x) = \sum_{i=1}^n \left(\sum_{j=1}^i x_j \right)^2$	30	$[-100, 100]$	0
$F_4(x) = \max_i \{ x_i , 1 \leq i \leq n \}$	30	$[-100, 100]$	0
$F_5(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30	$[-30, 30]$	0
$F_6(x) = \sum_{i=1}^n -x_i \sin(\sqrt{ x_i })$	30	$[-500, 500]$	-418.982 8n
$F_7(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	30	$[-5.12, 5.12]$	0
$F_8(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left[\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right] + 20 + e$	30	$[-32, 32]$	0
$F_9(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	$[-600, 600]$	0
$F_{10}(x) = \frac{\pi}{n} \{10 \sin(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_n - 1)^2\} + \sum_{i=1}^n u(x_i, 10, 100, 4)$ $y_i = 1 + \frac{x_i + 1}{4}$	30	$[-50, 50]$	0
$u(x_i, a, k, m) = \begin{cases} k(x_i - a)m, & x_i > a \\ 0, & -a < x_i < a \\ k(-x_i - a)m, & x_i < -a \end{cases}$			

表2 算法参数设置

Table 2 Algorithm parameter setting

算法	参数设置
SSA	ST=0.8; PD=0.2; SD=0.2
ISSA	ST=0.8; PD=0.2; SD=0.2
WOA	$a \in [0, 2]$, 并从2线性递减至0
PSO	$\omega = 0.9; c_1 = c_2 = 1.49445;$
GWO	$a \in [0, 2]$, 并从2线性递减至0; $r_1, r_2 \in [0, 1]$

3.1.2 算法性能对比分析

为了验证 ISSA 算法在寻优结果方面的准确性和稳定性,对每个测试函数独立运行 30 次,根据实验结果计算各算法的平均值和标准差作为性能评价依据,结果如表 3 所示。其中, $F_1 \sim F_5$ 为单峰函数,可验证算法的局部搜索性能、收敛特性及稳定性, $F_6 \sim F_{10}$ 为多峰函数,可验证算法跳出局部最优的能力以及全局寻优性能。

由表 3 可以看出,ISSA 在寻优稳定性和寻优精度方面比其他算法有显著的性能提升,对于单峰函数 $F_1 \sim F_4$ 的求解,ISSA 能够稳定地找到其理论最

优值,并且在平均值和标准差的计算方面表现更出色,寻优效果远超其他对比算法;对于 F_5 , ISSA 算法的寻优性能提升不明显,但其平均值和标准差仍然优于其他对比算法,表明 ISSA 具有良好的鲁棒性。对于多峰函数 $F_7 \sim F_{10}$, ISSA 算法标准差和平均值均比其他算法更小,而且在函数 F_7, F_9 中均能达到理论最优值;在函数 F_8 中,ISSA 与 SSA、WOA 找到的最优值一致,但比 PSO 和 GWO 更接近理论最优值;对于函数 F_6 , ISSA 算法的平均值优于其他算法,而标准差略逊于 PSO 和 GWO,然而与传统 SSA 算法相比,ISSA 在寻优性能方面有所提升。综合来看,ISSA 算法在优化精度和鲁棒性方面均具有优势,在处理单峰函数时,ISSA 算法能快速找到理论最优值,有效改善了全局搜索能力;在处理多峰函数时,ISSA 算法具备跳出局部最优的能力。

为了直观地展示各个算法的收敛速度、收敛精度以及跳出局部最优值的能力,5 种优化算法在 10 个基准函数下的收敛曲线如图 4 所示。

表3 测试函数结果

Table 3 Results of the test function

函数	指标	ISSA	SSA	WOA	PSO	GWO
F_1	平均值	2.11×10^{-2}	4.02×10^0	6.90×10^2	4.83×10^2	6.51×10^2
	标准差	1.38×10^0	4.81×10^2	7×10^2	2.69×10^2	4.62×10^2
	最优值	0	8.63×10^{-65}	2.81×10^{-86}	1.23×10^1	3.88×10^{-29}
F_2	平均值	1.39×10^{-2}	3.33×10^6	6.62×10^{10}	2.92×10^{10}	4.90×10^{10}
	标准差	2.34×10^{-1}	4×10^8	3.76×10^{12}	9.12×10^{11}	3.69×10^{12}
	最优值	0	1.43×10^{-137}	1.21×10^{-60}	3.62×10^0	2.19×10^{-17}
F_3	平均值	2.39×10^0	2.33×10^1	8.08×10^4	3.20×10^3	3.27×10^3
	标准差	9.18×10^1	1.17×10^3	1.27×10^4	1.34×10^3	1.75×10^3
	最优值	0	0	8.74×10^3	3.51×10^2	7.66×10^{-9}
F_4	平均值	2.17×10^{-3}	9.18×10^{-3}	5.40×10^1	1.45×10^1	3.37×10^0
	标准差	8.75×10^{-2}	6.67×10^{-1}	1.07×10^1	6×10^{-1}	1.44×10^0
	最优值	0	3.07×10^{-155}	1.92×10^{-1}	6.38×10^0	6.58×10^{-8}
F_5	平均值	2.45×10^0	1.41×10^4	2×10^6	4.89×10^5	1.72×10^6
	标准差	1.32×10^2	1.70×10^6	4.14×10^6	1.17×10^6	2.93×10^6
	最优值	3.25×10^{-11}	1.40×10^{-11}	2.69×10^1	1.17×10^3	2.61×10^1
F_6	平均值	-1.14×10^4	-8.62×10^3	-9.60×10^3	-5.75×10^3	-4.04×10^3
	标准差	2.89×10^2	7.04×10^2	4.80×10^2	2.11×10^2	1.86×10^2
	最优值	-1.26×10^4	-1.26×10^4	-1.26×10^4	-7.72×10^3	-7.90×10^3
F_7	平均值	2.06×10^{-1}	5.72×10^{-1}	2.40×10^1	2.04×10^2	3.18×10^1
	标准差	3.43×10^0	6.22×10^0	1.58×10^1	1.05×10^1	5.09×10^0
	最优值	0	0	0	9.79×10^1	0
F_8	平均值	7.58×10^{-3}	1.53×10^{-2}	7.61×10^{-1}	8.69×10^0	7.44×10^{-1}
	标准差	9.04×10^{-2}	2.21×10^{-1}	3.24×10^{-1}	3.82×10^{-1}	1.24×10^{-1}
	最优值	8.88×10^{-16}	8.88×10^{-16}	8.88×10^{-16}	4.14×10^0	7.19×10^{-14}
F_9	平均值	2.47×10^{-3}	4.25×10^{-2}	6.17×10^0	1.34×10^1	6.23×10^0
	标准差	1.87×10^{-1}	4.89×10^0	6.44×10^0	4.52×10^0	4.55×10^0
	最优值	0	0	0	4.87×10^{-1}	0
F_{10}	平均值	2.26×10^{-3}	3.70×10^{-3}	4.49×10^6	1.66×10^5	3.44×10^6
	标准差	5.75×10^{-2}	1.17×10^{-1}	1.03×10^7	1.14×10^6	6.26×10^6
	最优值	1.77×10^{-11}	2.93×10^{-10}	5.56×10^{-3}	3.38×10^0	1.25×10^{-2}

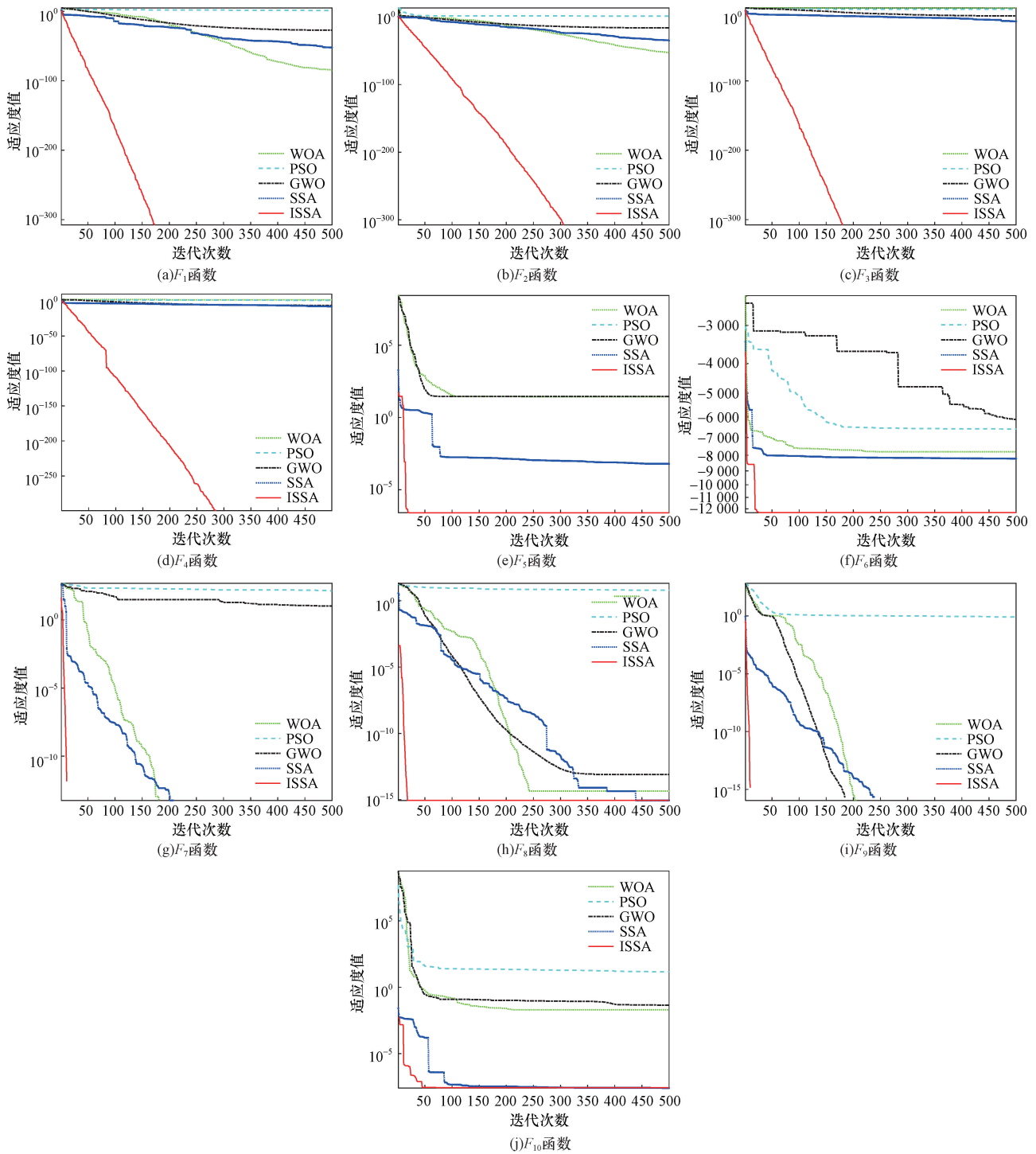


图4 测试函数收敛曲线图

Fig. 4 Convergence plot of the test function

从图4可知,ISSA算法相对于其他对比算法显示出最佳的收敛速度,能快速搜索空间,从而降低算法初始探索阶段,同时能在保持优化精度的情况下更接近最优解。对于单峰函数 $F_1 \sim F_4$,与其他算法相比,ISSA表现出收敛速度快和收敛精度高的性能,且具有良好的稳定性;对于函数 F_5 ,虽未找到最优解,但ISSA的收敛速度和精度明显优于其他

算法。在多峰函数 F_7 、 F_9 中,ISSA展现出较强的寻优性能和抗停滞能力,随着迭代次数的增加,ISSA的下降速度最快,并且保持良好的稳定性;在函数 F_8 中,ISSA的收敛精度与SSA、WOA相同,但可观察到ISSA具有更快的收敛速度且不容易陷入局部最优;而在函数 F_6 和 F_{10} 中,ISSA的收敛精度和收敛速度均优于其他算法。

综上所述,ISSA 在单峰函数和多峰函数中均展现出较强的寻优性能和抗停滞能力,证明了所提出的改进策略的有效性,即提高了算法的寻优性能,还降低了陷入局部最优的风险,使得 ISSA 整体性能优于 SSA 和其他对比算法,至此,ISSA 算法的有效性和可行性得以证明。

3.2 ISSA-UKF 算法仿真分析

假设车辆做匀加速直线运动,选取系统状态变量为 $\mathbf{X}_k = [x_k, v_{xk}, a_{xk}, y_k, v_{yk}, a_{yk}]^T$, 其中, x_k, v_{xk}, a_{xk} 分别为在 x 方向 k 时刻的位置、速度和加速度; y_k, v_{yk}, a_{yk} 分别为在 y 方向 k 时刻的位置、速度和加速度。

车辆运动模型利用文献[22]提出的“当前”统计模型,则状态方程为

$$\mathbf{X}_{k+1} = \boldsymbol{\varphi}_{k+1} \mathbf{X}_k + \mathbf{W}_k \quad (23)$$

式(23)中: \mathbf{W}_k 为状态噪声; $\mathbf{W}_k \sim N(0, \mathbf{Q})$, \mathbf{Q} 为过程噪声协方差矩阵。

$$\boldsymbol{\varphi}_{k+1} = \text{diag}[\boldsymbol{\varphi}_{x,k+1}, \boldsymbol{\varphi}_{y,k+1}] \quad (24)$$

$$\boldsymbol{\varphi}_{x,k+1} = \boldsymbol{\varphi}_{y,k+1} = \begin{bmatrix} 1 & T & \frac{T^2}{2} \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix} \quad (25)$$

式中: T 为采样周期。

量测方程为

$$\mathbf{Z}_k = \begin{bmatrix} x_{G,k} \\ y_{G,k} \\ \omega_k \\ s_k \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \\ \frac{v_{y,k} a_{x,k} - v_{x,k} a_{y,k}}{v_{x,k}^2 + v_{y,k}^2} \\ T \sqrt{v_{x,k}^2 + v_{y,k}^2} \end{bmatrix} + \begin{bmatrix} \nu_1 \\ \nu_2 \\ \nu_\omega \\ \nu_s \end{bmatrix} \quad (26)$$

式(26)中: ν_1, ν_2 分别为 GPS 接收机输出的 x 方向和 y 方向位置的量测噪声; ν_ω 为角速度陀螺的量测噪声; ν_s 为里程计的量测噪声。

将系统状态方程和量测方程代入 ISSA-UKF 算法模型中,实现对算法的仿真验证。

为验证 ISSA-UKF 算法的可行性和有效性,将其与 UKF 算法、自适应 UKF(AUKF)算法进行仿真对比,设定算法的初始状态为 $\mathbf{X}_0 = [0, 10, 0, 0, 10, 0]$, 采样周期 $T = 1$ s, 采样次数 $N = 300$, 智能优化算法种群数量为 10, 迭代次数为 50, 位置均方误差仿真结果如图 5 和图 6 所示,速度均方误差仿真结果如图 7 和图 8 所示。

可以看出,在 x 方向和 y 方向的位置均方误差中, AUKF 算法相比于 UKF 算法表现出更小的波动,而 ISSA-UKF 算法相比于 AUKF 算法和 UKF 算法效果更好,位置均方误差的波动较小,更加稳定,因此,ISSA-UKF 算法的滤波效果更好,精度较高,误

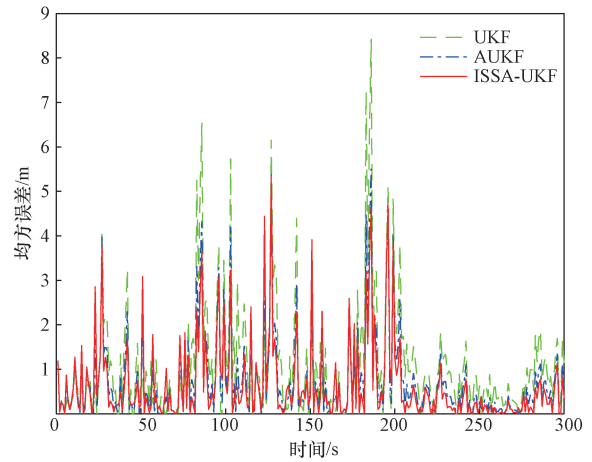


图 5 x 方向位置均方误差

Fig. 5 X -direction position mean square error

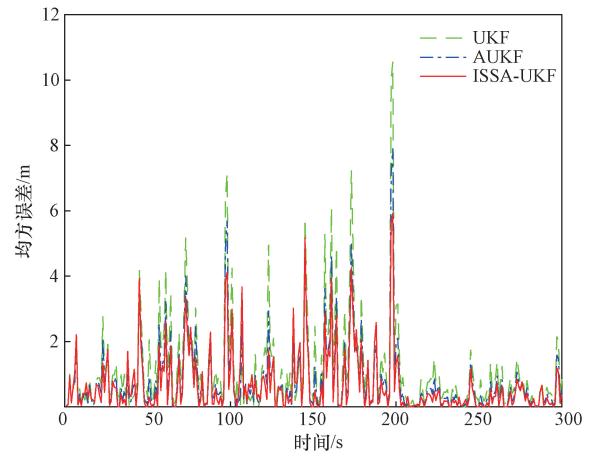


图 6 y 方向位置均方误差

Fig. 6 Y -direction position mean square error

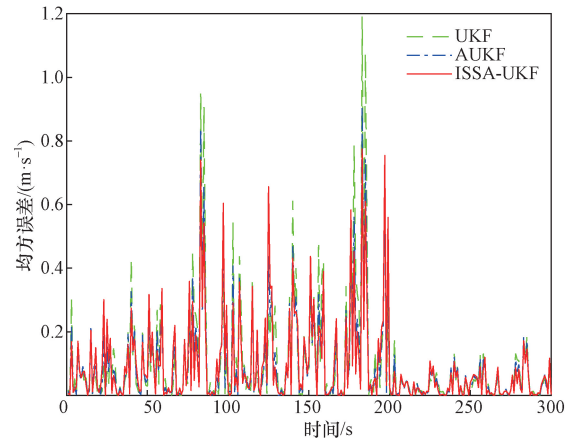


图 7 x 方向速度均方误差

Fig. 7 X -direction velocity mean square error

差相对较小;在 x 方向和 y 方向的速度均方误差中, ISSA-UKF 算法偶尔会得到较大的均方误差,但从整体效果仍然可看出,ISSA-UKF 算法的效果更好,而 UKF 算法和 AUKF 算法的均方误差波动大,较不稳

定。如表 4 所示,ISSA-UKF 算法相对于 UKF 算法的位置均方根误差减小了 52.2%,速度均方根误差减小了 21.9%,与 AUKF 算法相比,ISSA-UKF 算法在位置均方根误差上减小了 30.1%,在速度均方根误差上减小了 13%。综上可知,ISSA-UKF 相比于传统 UKF 算法和 AUKF 算法的性能更好,能够有效降低误差,进一步验证了 ISSA-UKF 算法的有效性和可行性。

为验证 ISSA-UKF 算法的稳定性,选取 10 组数据,并在每组数据中,分别使各对比算法进行 20 次独立运行,取均方根误差的均值作为结果,实验结果如图 9 和图 10 所示:

从图 9 和图 10 可知,AUKF 算法相较于 UKF 算法具有较小的位置均方根误差和速度均方根误差,说明在加入 SSA 优化后,滤波效果得到了一定程度

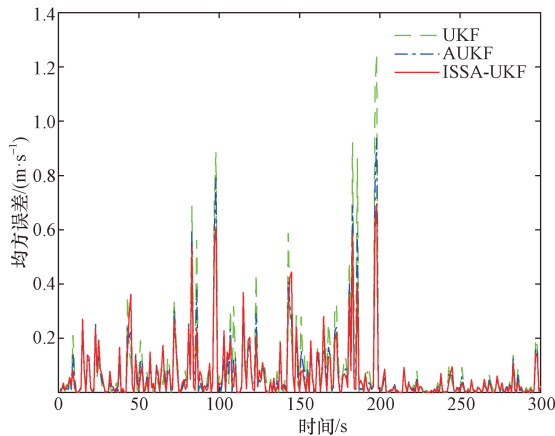


图 8 y 方向速度均方误差
Fig. 8 Y-direction velocity mean square error

表 4 均方根误差对比

Table 4 Comparison of root mean square error

算法	UKF	AUKF	ISSA-UKF
位置均方根误差/m	1.058 76	0.723 819	0.505 756
速度均方根误差/(m·s ⁻¹)	0.168 18	0.150 927	0.131 186

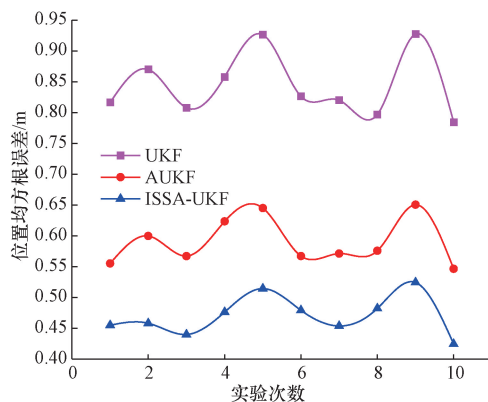


图 9 位置均方根误差统计
Fig. 9 Position root mean square error statistics

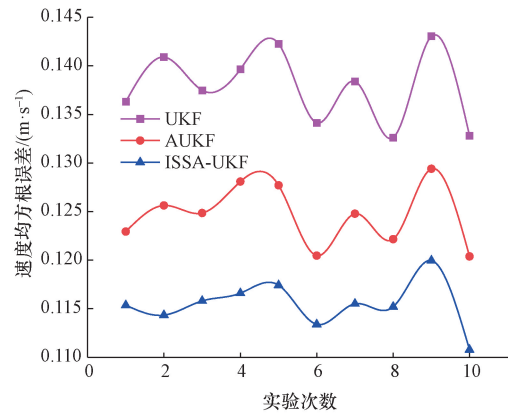


图 10 速度均方根误差统计

Fig. 10 Velocity root mean square error statistics

的改进,提高了滤波精度和稳定性;ISSA-UKF 算法的位置均方根误差和速度均方根误差同样均比 UKF 算法和 AUKF 算法小,表明 ISSA-UKF 算法在滤波精度方面优于传统的 UKF 算法和 AUKF 算法,验证了 ISSA-UKF 算法的有效性和可行性,同时,在多组数据下能够保持较低的误差,说明 ISSA-UKF 算法具备较好的稳定性。

4 结论

为了提高 UKF 算法的状态估计性能,提出利用麻雀搜索算法改进 UKF 的方法,但麻雀搜索算法具有收敛速度慢和易陷入局部最优等问题,因此针对麻雀搜索算法提出利用 Cubic 混沌映射策略、非线性自适应收敛因子、小波变异策略和自适应 t 分布等多种策略对其进行改进,以改善算法优化性能,然后通过改进后麻雀搜索算法对 UT 变换中的采样点分布状态控制参数进行寻找调整,以优化 Sigma 点分布,从而提高滤波效果。

通过测试函数仿真实验,表明了 ISSA 算法在寻优能力和收敛速度等方面相比其他算法具有显著优势。最后,验证了经 ISSA 优化后的 UKF 算法的效果,实验表明,相较于传统 UKF 算法,ISSA-UKF 算法的位置均方根误差降低了 52.2%,速度均方根误差降低了 21.9%,与 AUKF 算法相比,ISSA-UKF 算法的位置均方根误差降低了 30.1%,速度均方根误差降低了 13%,充分证明了 ISSA-UKF 算法的有效性,并为提升 UKF 系列滤波器性能提供了一定的参考价值。

参 考 文 献

- [1] Wang Y, Li Y, Zhao Z. State parameter estimation of intelligent vehicles based on an adaptive unscented Kalman filter[J]. Electronics, 2023, 12(6): 1500.
- [2] Wang J, Wang J, Zhang D, et al. Kalman filtering through the

- feedback adaptation of prior error covariance[J]. *Signal Processing*, 2018, 152: 47-53.
- [3] Gustafsson F, Hendeby G. Some relations between extended and unscented Kalman filters[J]. *IEEE Transactions on Signal Processing*, 2011, 60(2): 545-555.
- [4] 余成波, 成科宏. WiFi 与行人航迹推算自适应无迹卡尔曼滤波融合定位算法[J]. *科学技术与工程*, 2020, 20(27): 11155-11160.
- Yu Chengbo, Cheng Kehong. WiFi and pedestrian dead reckoning adaptive unscented Kalman filter fusion positioning algorithm[J]. *Science Technology and Engineering*, 2020, 20(27): 11155-11160.
- [5] Ge B, Zhang H, Jiang L, et al. Adaptive unscented Kalman filter for target tracking with unknown time-varying noise covariance[J]. *Sensors*, 2019, 19(6): 1371.
- [6] Al B N, Gavrilov A. A novel approach for aiding unscented Kalman filter for bridging GNSS outages in integrated navigation systems[J]. *Navigation*, 2021, 68(3): 521-539.
- [7] Wu Z, Wang G, Xie Z, et al. Lithium battery SOC estimation based on whale optimization algorithm and unscented Kalman filter[J]. *Journal of Renewable and Sustainable Energy*, 2020, 12(6): 065501.
- [8] Zheng B, Fu P, Li B, et al. A robust adaptive unscented Kalman filter for nonlinear estimation with uncertain noise covariance[J]. *Sensors*, 2018, 18(3): 808.
- [9] Yang C, Shi W, Chen W. Robust M-M unscented Kalman filtering for GPS/IMU navigation[J]. *Journal of Geodesy*, 2019, 93: 1093-1104.
- [10] Li W, Sun S, Jia Y, et al. Robust unscented Kalman filter with adaptation of process and measurement noise covariances[J]. *Digital Signal Processing*, 2016, 48: 93-103.
- [11] Novi T, Capitani R, Annicchiarico C. An integrated artificial neural network; unscented Kalman filter vehicle sideslip angle estimation based on inertial measurement unit measurements[J]. *Journal of Automobile Engineering*, 2019, 233(7): 1864-1878.
- [12] Yang Y, Li F, Gao Y, et al. Multi-sensor combined measurement while drilling based on the improved adaptive fading square root unscented Kalman filter[J]. *Sensors*, 2020, 20(7): 1897.
- [13] Dunik J, Simandl M, Straka O. Unscented Kalman filter; aspects and adaptive setting of scaling parameter[J]. *IEEE Transactions on Automatic Control*, 2012, 57(9): 2411-2416.
- [14] 贺军义, 李男男. 基于和声差分进化的 UKF 改进算法[J]. *控制与决策*, 2018, 33(4): 614-620.
- He Junyi, Li Nannan. Improved UKF algorithm based on harmony search differential evolution[J]. *Control and Decision*, 2018, 33(4): 614-620.
- [15] Xue J, Shen B. A novel swarm intelligence optimization approach: sparrow search algorithm[J]. *Systems Science & Control Engineering*, 2020, 8(1): 22-34.
- [16] Palacios A. Cycling chaos in one-dimensional coupled iterated maps[J]. *International Journal of Bifurcation and Chaos*, 2002, 12(8): 1859-1868.
- [17] Feng J, Zhang J, Zhu X, et al. A novel chaos optimization algorithm[J]. *Multimedia Tools and Applications*, 2017, 76: 17405-17436.
- [18] 徐明, 龙文. 基于多策略融合灰狼优化算法的特征选择方法[J]. *科学技术与工程*, 2021, 21(20): 8544-8551.
- Xu Ming, Long Wen. Feature selection method based on grey wolf optimizer algorithm integrated with multiple-strategies[J]. *Science Technology and Engineering*, 2021, 21(20): 8544-8551.
- [19] 付华, 刘昊. 多策略融合的改进麻雀搜索算法及其应用[J]. *控制与决策*, 2022, 37(1): 87-96.
- Fu Hua, Liu Hao. Improved sparrow search algorithm with multi-strategy integration and its application[J]. *Control and Decision Making*, 2022, 37(1): 87-96.
- [20] Ling S H, Iu H H C, Chan K Y, et al. Hybrid particle swarm optimization with wavelet mutation and its industrial applications[J]. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 2008, 38(3): 743-763.
- [21] 毛清华, 王迎港. 融合改进 Logistics 混沌和正弦余弦算子的自适应 t 分布海鸥算法[J]. *小型微型计算机系统*, 2022, 43(11): 2271-2277.
- Mao Qinghua, Wang Yinggang. Adaptive t -distribution seagull optimization algorithm combining improved Logistics chaos and sine-cosine operator[J]. *Small Micro Computer System*, 2022, 43(11): 2271-2277.
- [22] 周宏仁, 敬忠良, 王培德. 机动目标跟踪[M]. 北京: 国防工业出版社, 1991.
- Zhou Hongren, Jing Zhongliang, Wang Peide. *Motorized target tracking* [M]. Beijing: National Defense Industry Press, 1991.