

基于多算法融合的汽车 UWB 数字钥匙定位方法研究

王朝阳, 李 辉

(河南天海电器有限公司, 河南 鹤壁 458031)

摘 要:汽车超宽带数字钥匙定位方法大多采用最小二乘法,该方法在解算过程中会出现奇异矩阵,导致模型失效,进而造成定位精度严重下降。本文提出采用列文伯格-马夸尔特模型解决最小二乘模型的病态问题,并构建泰勒级数模型进一步提升定位精度。结果表明,该融合算法能够极大增强钥匙定位系统的准确性及鲁棒性,且单次迭代时间满足实时性要求。

关键词:多算法融合; 汽车 UWB 数字钥匙; 定位方法

中图分类号:U463. 67

文献标志码:A

DOI:10. 15917/j. cnki. 1006-3331. 2025. 04. 002

Research on Positioning Method of Automotive UWB Digital Key Based on Multi-algorithm Fusion

WANG Chaoyang, LI Hui

(Henan Tianhai Electric Appliance Co., Ltd., Hebi 458031, China)

Abstract: Aiming at the problem that most of the vehicle ultra-wideband digital key positioning methods use the least square method, but the singular matrix will appear in the solution process, which will lead to the failure of the model and the serious decline of the positioning accuracy. This paper applies a Levenberg-Marquardt model to solve the ill-conditioned problem of the least squares model and establishes a Taylor series model to improve positioning accuracy further. The results show that this fusion algorithm can greatly enhance the accuracy and robustness of the key positioning system, and the single iteration time meets the real-time requirements.

Key words: multi-algorithm fusion; vehicle UWB digital key; positioning method

1 概 述

汽车钥匙经历了从机械钥匙、遥控钥匙、芯片钥匙到蓝牙-NFC 钥匙的发展阶段。如今随着“新四化”席卷汽车行业,汽车钥匙正式迈入 3.0 数字时代——超宽带(Ultra-wideband, UWB)技术方案。相比于蓝牙-NFC 技术方案,UWB 具有抗干扰能力强、穿透性强、测距精度高等优点,能够实现对汽车钥匙高精度定位的功能,拓展了更多的使用场景。故研究基于 UWB 技术的车辆数字钥匙定位系统具有重要的工程应用意义。

目前,如何在较小的软硬件资源消耗下实现高精

度、强鲁棒性的 UWB 定位算法是学者们当前研究的重点。文献[1]将标签和各锚点之间的信号强度作为算法输入,结合三边测量法来解算标签位置,但接收信号强度指示值通常存在很大的波动性,导致定位偏差较大。文献[2]将神经网络与极端梯度提升算法相结合,定位精度可达 5.08 cm,但增加了软硬件资源消耗。文献[3]提出了最小二乘法与泰勒级数融合的二维定位算法,但无法避免计算过程出现奇异矩阵导致定位点精度严重下降的情况。文献[4]提出了粒子群优化的极限学习机定位模型,但模型涉及多个参数标定,应用较为复杂。文献[5]提出构建

收稿日期:2024-11-05。

第一作者:王朝阳(1994—),男,硕士,助理工程师,主要从事车辆控制器软硬件系统开发和应用层算法软件开发工作。E-mail: 252011502@qq. com。

BP(Back Propagation)神经网络模型输出测距误差,但模型需要大数据进行训练。

本文提出一种多算法融合的 UWB 定位方法。首先,构建一种整车多锚点 UWB 数字钥匙物理架构,确定各 UWB 锚点坐标。其次,依据锚点与钥匙间的测距值建立最小二乘(Least Square,LS)模型,初步解算钥匙坐标。若解算过程中出现矩阵接近奇异的情况,立刻结束解算,将缓存中的钥匙坐标代入寻优模型重新解算;若未出现矩阵接近奇异的情况,则基于 LS 模型得到的钥匙坐标建立泰勒级数(Taylor Series,TS)模型,优化出钥匙的精确坐标。最后,通过 MATLAB/Simulink 仿真与硬件在环测试对融合算法进行验证。

2 UWB 数字钥匙融合算法原理

2.1 UWB 数字钥匙整车架构

本文基于某 SUV 车型的尺寸构建 UWB 数字钥匙的整车物理架构,如图 1 所示,共部署了 4 个 UWB 从锚点和 1 个 UWB 主锚点。其中,主锚点⑤布置在车内顶棚的中央位置,从锚点①~④布置在车辆前后保险杠的角点位置。这种分散布置的方式能够保证钥匙发射的无线信号被尽可能多的锚点捕捉,从而提升数字钥匙系统对钥匙的定位精确度。主从锚点分时依次与钥匙完成测距后,由主锚点整合各锚点与钥匙间的距离值,并传递至主锚点内软件算法的定位模型中,最终解算出钥匙坐标。

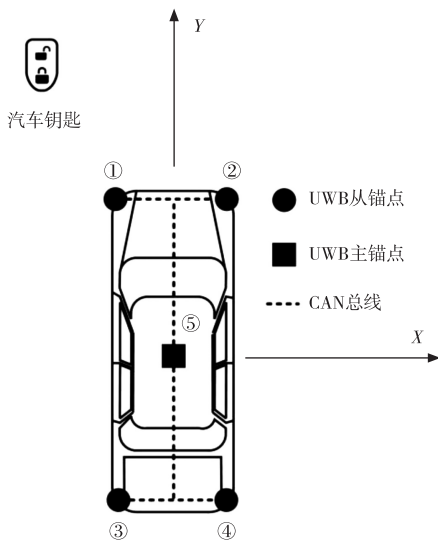


图 1 整车锚点部署架构

基于以上需求,主锚点一般采用较高算力的微控制单元(Micro Controller Unit, MCU)芯片,从锚点则采用集成 UWB 功能的系统级芯片(System on Chip, SoC)。整合计算完成后,主锚点通过车内总线将得到的钥匙坐标值传递给车身控制器(Body Control Module, BCM)和座椅控制器(Seat Control Unit, SCU)。根据钥匙坐标位置和功能定义,BCM 和 SCU 决定是否执行迎宾大灯闪烁或熄灭、双侧电滑门自动开关以及记忆座椅自动调整等操作。

以 UWB 主锚点为圆心构建二维坐标系,则锚点①至锚点⑤的坐标分别为: $P_1 = [-0.915, 2.3]$, $P_2 = [0.915, 2.3]$, $P_3 = [-0.915, -2.3]$, $P_4 = [0.915, -2.3]$, $P_5 = [0, 0]$,单位为 m。CAN 总线主要用于主锚点和各从锚点间的时钟同步,是测距会话开始的标志。主从锚点和钥匙之间主要采用飞行时间测算距离值。有关测距会话建立和执行的详细内容可参阅 CCC3.0(Car Connectivity Consortium)数字钥匙协议。

2.2 融合算法流程图

在 UWB 数字钥匙实际使用场景中,当进入测距过程时,车上 5 个锚点与钥匙间的无线通信信号可能会因金属车身的遮挡和多径反射而受到影响,导致某些锚点与钥匙间的测距值异常偏大、异常偏小或为 0(钥匙发射信号被严重遮挡)。这种情况称为无效测距值,反之则称为有效测距值。

本文构建的融合算法流程如图 2 所示。当有效测距的锚点数量不少于 3 个时,运行 LS 模型。若 LS 模型解算过程中未发现求解矩阵奇异,则接着运行 TS 模型来优化 LS 模型输出的钥匙坐标。若发现求解矩阵奇异,LS 模型解算立即终止,并检查钥匙缓存区中是否存在先前解算成功的钥匙坐标。若存在,则运行寻优模型对钥匙坐标进行求解,并在每次成功解算后更新钥匙坐标缓存区;若不存在,则不满足融合算法运行条件,直接进入下一次测距过程。该算法流程在测距过程中持续运行,直至数字钥匙系统检测到钥匙与车辆之间的距离超出某一预设值,或驾驶员点火启动车辆后,主锚点向各从锚点发出测距停止命令。

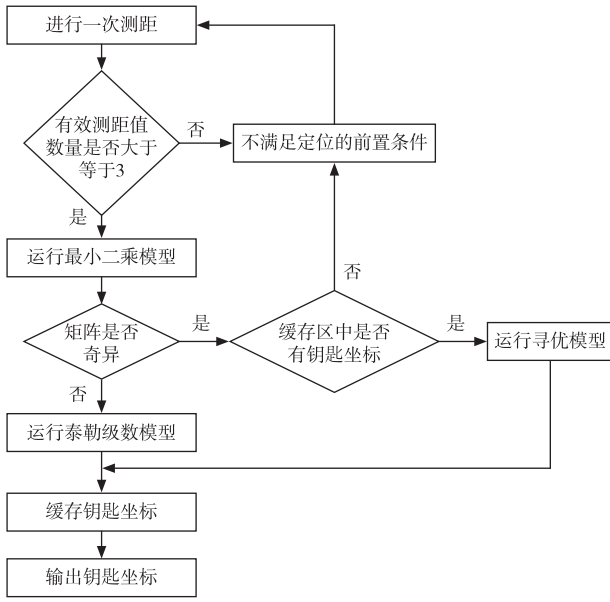


图2 融合算法流程图

2.3 数字钥匙定位圆交模型

假设在某一测距轮次下,数字钥匙系统主锚点和从锚点共同得到 n 个有效测距值 ($3 \leq n \leq 5$), 分别为 $R_1, R_2, \dots, R_n$ 。各锚点相对于钥匙的坐标分别为: $P_1 = [x_1, y_1], P_2 = [x_2, y_2], \dots, P_n = [x_n, y_n]$ 。设待求数字钥匙坐标 $P_{\text{Key}} = [x, y]$, 则可建立如下数学方程:

$$(x-x_i)^2 + (y-y_i)^2 = R_i^2, i \in [1, 5] \quad (1)$$

将式(1)逐项展开后相减,最后整理成 $A \cdot P_{\text{LS}} = B$ 矩阵,式中各组成元素如下:

$$A = \begin{bmatrix} 2(x_1-x_2) & 2(y_1-y_2) \\ 2(x_1-x_3) & 2(y_1-y_3) \\ \vdots & \vdots \\ 2(x_1-x_n) & 2(y_1-y_n) \end{bmatrix} \quad (2)$$

$$B = \begin{bmatrix} R_2^2 - R_1^2 - x_2^2 + x_1^2 - y_2^2 + y_1^2 \\ R_3^2 - R_1^2 - x_3^2 + x_1^2 - y_3^2 + y_1^2 \\ \vdots \\ R_n^2 - R_1^2 - x_n^2 + x_1^2 - y_n^2 + y_1^2 \end{bmatrix} \quad (3)$$

$$P_{\text{LS}} = [x_{\text{LS}}, y_{\text{LS}}]^T \quad (4)$$

因为待求解的未知数共有两个 (x, y) , 所以矩阵 A 的维度应至少为 2×2 。换言之,只有当 $n \geq 3$ 时,式(1)才可以进行解算,这就是要求有效测距值个数不少于3的原因。而且有效测距值越多,定位结果越准

确。

当数字钥匙发出信号时,由于金属车身反射及多径传输的影响,各锚点与钥匙的 UWB 测距值会存在厘米级或分米级的误差,导致各锚点测距圆并不会交于一点(理论上这一点即为钥匙所在位置),而会交于一个区域^[6],如图3所示。此时上述构建的定位圆交模型无法得出唯一解,故需建立 LS 模型计算其最优解。

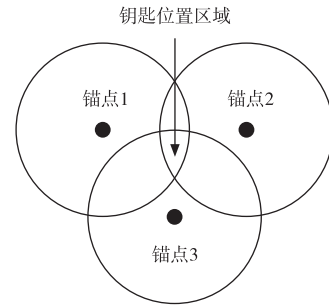


图3 定位圆交模型示意图

2.4 数字钥匙 LS 模型

定位模型首先运行 LS 模型,步骤如下^[7]:

1) 设置误差向量 ϵ :

$$\epsilon = A \cdot P_{\text{LS}} - B \quad (5)$$

2) 取误差向量平方 E :

$$E = \epsilon^T \cdot \epsilon = (A \cdot P_{\text{LS}} - B)^T \cdot (A \cdot P_{\text{LS}} - B) \quad (6)$$

3) 对 E 求偏导数,令其为0:

$$\frac{dE(P_{\text{LS}})}{dP_{\text{LS}}} = 2A^T A P_{\text{LS}} - 2A^T B = 0 \quad (7)$$

4) 解方程(7)可得:

$$P_{\text{LS}} = (A^T A)^{-1} A^T B \quad (8)$$

求解钥匙坐标 P_{LS} 时会先对 $A^T A$ 的奇异性进行判断,如果 $A^T A$ 奇异,则此计算无法继续进行,立即结束计算 P_{LS} ,转而运行 2.6 节中的寻优模型;若 $A^T A$ 非奇异,则先继续计算 P_{LS} ,接着运行 2.5 节中的 TS 模型对 P_{LS} 进行优化。

2.5 数字钥匙 TS 模型

TS 本质上是利用函数在某一点的无穷阶导数信息来构建多项式级数逼近原函数,在实际运用中,由于三阶及以后的导数值非常小,所以一般只取到二阶导数项^[8]。TS 的基本公式如下:

$$f(x) = f(a) + f'(a)(x-a) + \frac{f''(a)}{2!}(x-a)^2 \quad (9)$$

结合本文所要解决的问题, 钥匙坐标点 P_{LS} 即为式中的点 a , 优化后的钥匙坐标点 P_{TS} 即为待求未知点 x , 目标函数 $f(x)$ 根据测距值平方与 P_{LS} 到相应测距圆心距离平方的差值来建立:

$$f_i(x_{LS}, y_{LS}) = R_i^2 - [(x_i - x_{LS})^2 + (y_i - y_{LS})^2] \quad (10)$$

假设 TS 优化后的钥匙最优坐标 $P_{TS} = (x_{TS}, y_{TS})$, 代入式 (10) 可得:

$$f_i(x_{TS}, y_{TS}) = R_i^2 - [(x_i - x_{TS})^2 + (y_i - y_{TS})^2] = 0 \quad (11)$$

假设 P_{LS} 与 P_{TS} 在 x 和 y 维度方向相差 $[\alpha, \beta]$, 利用 TS 基本公式对其进行展开得到式 (12):

$$f_i(x_{TS}, y_{TS}) = f_i(x_{LS}, y_{LS}) + \frac{\partial f_i}{\partial x_{LS}}\alpha + \frac{\partial f_i}{\partial y_{LS}}\beta \quad (12)$$

进一步整理为矩阵相乘的形式:

$$\begin{bmatrix} \frac{\partial f_i}{\partial x_{LS}} & \frac{\partial f_i}{\partial y_{LS}} \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = -f_i(x_{LS}, y_{LS}) \quad (13)$$

其中,

$$\frac{\partial f_i}{\partial x_{LS}} = 2(x_i - x_{LS}) \quad (14)$$

$$\frac{\partial f_i}{\partial y_{LS}} = 2(y_i - y_{LS}) \quad (15)$$

将各基站坐标代入式 (13) 可得:

$$2(x_i - x_{LS})\alpha + 2(y_i - y_{LS})\beta = d_i \quad (16)$$

其中:

$$d_i = (x_i - x_{LS})^2 + (y_i - y_{LS})^2 - R_i^2 \quad (17)$$

以上即为 TS 模型的建立过程。模型在每次迭代后均会得到一组 $[\alpha, \beta]$, 但模型不能长时间计算, 否则会影响数字钥匙系统的实时性。本文设置阈值 $H = \alpha^2 + \beta^2$, 如果 $H > 0.001$, 则 TS 模型停止计算, 认定此刻解算结果为最优解, 即 $P_{TS} = \begin{bmatrix} x_{LS} + \alpha \\ y_{LS} + \beta \end{bmatrix}$, 最后将 P_{TS} 存储到钥匙坐标缓存区中。

2.6 数字钥匙寻优模型

2.6.1 LS 模型的局限性

从 2.4 节可知, LS 模型钥匙坐标解算公式为 $P_{LS} = (A^T A)^{-1} A^T B$, 其中, $A^T A$ 在计算过程中会存在接近奇异的情况 (即行列式值接近于 0), 进而导致解算出

的钥匙坐标和真实坐标相差较大, 严重影响数字钥匙系统功能。为解决上述问题, 每当 LS 模型运行至 $A^T A$ 处, 立即对行列式值进行计算, 一旦发现行列式绝对值小于 0.001, 则立即结束 LS 模型, 由寻优模型 (该模型的一些参数应事先初始化) 接管钥匙坐标的解算任务。下面分别构建列文伯格-马夸尔特 (Levenberg-Marquardt, LM) 和粒子群优化 (Particle Swarm Optimization, PSO) 两种典型寻优模型进行仿真, 选取两者中鲁棒性更强的算法作为本文融合算法最终的寻优模型。

2.6.2 LM 模型

LM 模型是一种结合了牛顿法和梯度下降法的优化算法^[9], 通过近似目标函数的海森矩阵来估计参数的更新方向。本文构建 LM 模型目标函数的方法与 TS 模型目标函数的方法一致, 均是利用误差建立模型, 直接列出目标方程^[9]:

$$S(x^{(k)}, y^{(k)}) = \sum_{i=1}^n f_i^2(x^{(k)}, y^{(k)}) \quad (18)$$

其中:

$$f_i(x^{(k)}, y^{(k)}) =$$

$$\sum_{i=1}^n [\sqrt{(x^{(k)} - x_i)^2 + (y^{(k)} - y_i)^2} - R_i] \quad (19)$$

式中: $S(x^{(k)}, y^{(k)})$ 为 k 时刻的目标函数; $(x^{(k)}, y^{(k)})$ 为 k 时刻待求钥匙的横坐标和纵坐标; x_i, y_i 为第 i 个 UWB 锚点的横坐标值和纵坐标值, i 的取值范围为 $[1, 5]$; k 为系统运行时刻, 取值范围为 $[0, T]$, T 为系统结束时刻; n 为一个测距循环内的有效测距值数量。

由于 LM 模型在求解目标函数的过程中引入了正定对角阵 μI (其中 μ 为阻尼系数, I 为单位矩阵), 所以 LM 模型修正了 LS 模型中因矩阵奇异而导致方程无解的问题。但是 LM 模型需要相对准确的初始值作为迭代输入。若初始值离目标解较远, 则模型迭代时间较长, 不满足数字钥匙系统的实时性要求。

针对以上情况, 本文建立了先进先出的钥匙坐标缓存区, 用于保存过往时刻成功解算出的钥匙坐标。每当运行 LM 模型时, 则从缓存区中取出最新的钥匙坐标作为模型输入, 以完成最优值的搜索求解。LM

模型的关键迭代步骤如下:

假设数字钥匙系统运行至 k 时刻, 先从缓存区中取出历史钥匙坐标 $P_{LM}^k = [x_{Taylor}^{k-1}, y_{Taylor}^{k-1}]$, 接着按照以下步骤运行 LM 模型^[10]:

- 1) 确定初始点 P_{LM}^k , 选取 $\beta \in (0, 1)$, μ 和 ν 大于 1 且精度大于 0, 其中 β 为置信度因子, μ 为阻尼系数, ν 为放大系数。
- 2) 计算 $f(x^{(k)}, y^{(k)})$, $S(x^{(k)}, y^{(k)})$ 。
- 3) 计算偏导数 $\nabla f(x^{(k)}, y^{(k)})$ 。
- 4) 计算 $\nabla S(x^{(k)}, y^{(k)}) = [\nabla f(x^{(k)}, y^{(k)})]^T \cdot f(x^{(k)}, y^{(k)})$ 。
- 5) 令 $Q = [\nabla f(x^{(k)}, y^{(k)})]^T \cdot \nabla f(x^{(k)}, y^{(k)})$, 解方程 $[Q + \mu I](\Delta x, \Delta y) = -\nabla S(x^{(k)}, y^{(k)})$ 。
- 6) 令 $x^{(k+1)} = x^{(k)} + \Delta x$, $y^{(k+1)} = y^{(k)} + \Delta y$ 。计算终止条件是否满足, 若满足则结束 LM 模型, 输出钥匙坐标 $x^{(k+1)}$ 和 $y^{(k+1)}$, 若不满足则转步骤 7)。
- 7) 若 $S(x^{(k+1)}, y^{(k+1)}) < S(x^{(k)}, y^{(k)}) + \beta [\nabla S(x^{(k)}, y^{(k)})]^T (\Delta x, \Delta y)$, 令 $\mu = \mu / \nu$, 转至步骤 8) 运行, 否则 $\mu = \mu \cdot \nu$, 转至步骤 5) 运行。
- 8) 令 $k = k + 1$, 转步骤 2) 及其后续步骤, 直至步骤 6) 输出模型的钥匙坐标 $x^{(k+1)}$ 和 $y^{(k+1)}$ 。

2.6.3 PSO 模型

PSO 模型是通过模拟群体中个体的协作和信息共享来寻找问题最优解的一种算法^[11]。尽管与 LM 模型同为寻优模型, 但 PSO 模型并非利用梯度信息进行模型迭代, 而是通过寻找空间中的最优粒子来调整模型的迭代方向。在 PSO 模型中, 粒子代表了问题的一个潜在解, 可以看作是搜索空间中的一个点。每个粒子都有自己的位置和速度, 具体到本文, 粒子是指钥匙的可能坐标, 速度是指钥匙坐标移动的方向和快慢。以下是 PSO 模型的迭代步骤:

- 1) 初始化粒子群体(群体规模、随机位置及速度等)。
- 2) 根据适应度方程, 评价每个粒子的适应度。
- 3) 根据每个粒子的适应度更新个体历史最佳位置与全局最佳位置。
- 4) 更新每个粒子的速度与位置。
- 5) 判断迭代次数是否满足要求, 若不满足则转至步骤 2) 继续运行, 直至满足要求为止。

3 UWB 数字钥匙融合算法的仿真验证

3.1 钥匙轨迹数据集

验证融合算法定位精确度的前提是构建一条绕车移动的钥匙轨迹数据集, 再结合 5 个锚点的位置生成相应的测距值, 最后整理成特定的坐标数据格式导入 MATLAB 的工作空间, 以便 Simulink 模型调用。本文使用 WebPlotDigitizer 工具生成一条闭合轨迹, 将数据点导入 MATLAB 中进行可视化显示, 如图 4 所示。

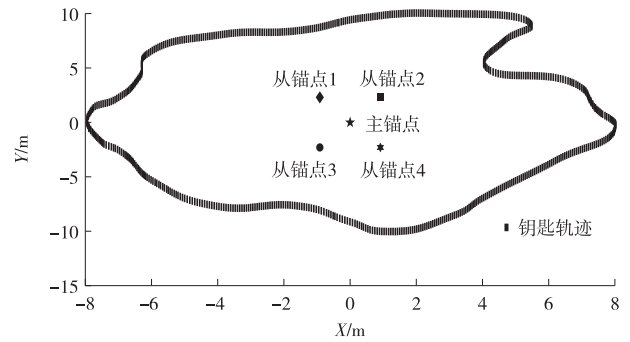


图4 钥匙仿真移动轨迹

由于 UWB 芯片本身存在约 10 cm 的测距误差, 因此需要在各锚点的测距值中加入一个在 $[-10, +10]$ cm 范围内的随机误差。以下简要描述将钥匙轨迹数据集分解为各锚点测距值数据集的步骤:

- 1) 取钥匙轨迹的第 k 个点 $P_{key_move}^k = [x_{key_move}^k, y_{key_move}^k]$, k 的初始值取 1。
- 2) 计算各锚点坐标与 $P_{key_move}^k$ 之间的直线距离 $d_{key_move}^k(i) = \sqrt{(x_{key_move}^k - x_i)^2 + (y_{key_move}^k - y_i)^2}$ 。
- 3) 加入随机测距误差, 得到第 k 个点的测距值数组 $D_{key_move}^k = [D_{key_move}^k(1), \dots, D_{key_move}^k(5)]$, 其中 $D_{key_move}^k(i) = d_{key_move}^k(i) + \text{random}(-0.1, 0.1)$, $\text{random}(-0.1, 0.1)$ 表示生成一个在 $[-0.1, 0.1]$ 内的随机数。
- 5) 使用 TestCopy 存储 $D_{key_move}^k$, 更新 $k = k + 1$, 转至步骤 2) 及后续步骤, 直至解析完所有的轨迹点。

3.2 融合算法 MATLAB/Simulink 仿真模型

在 MATLAB/Simulink 平台中搭建 UWB 数字钥匙定位融合算法模型, 使用 M 语言在 MATLAB Function 模块中实现融合算法。整个模型包含参数初始化模块和主程序模块, 如图 5 所示。

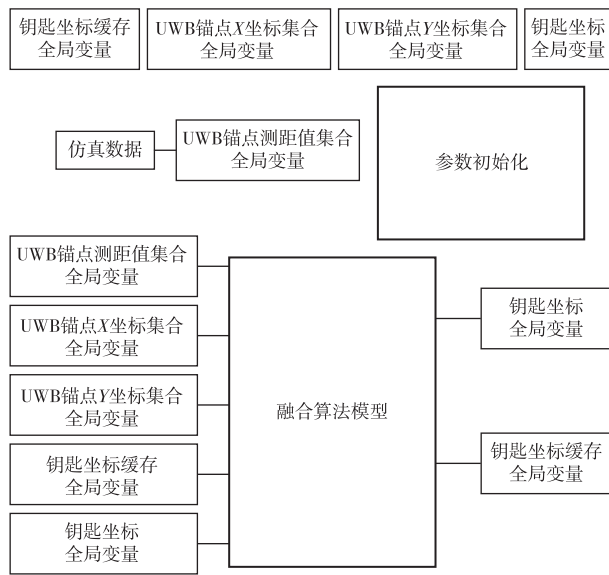


图5 融合算法 MATLAB/Simulink 模型

3.3 仿真试验

3.3.1 试验内容

UWB 数字钥匙系统实际运行过程中,由于信号遮挡及多径干扰因素的影响^[12],不同测距轮次下可能只会得到3个或4个有效测距值,故需要对3.1节中的 TestCopy 进行进一步裁剪,使其更具真实性。本文采用以下两种方式对 TestCopy 进行裁剪,两种裁剪方式各对应一组仿真试验:

1) 仿真试验一。由于锚点距离钥匙越近,其对应的测距值成为有效测距值的概率越大,故遍历 TestCopy 每个时刻下的测距值数组,保留5个锚点测距值中相对较小的3个测距值。

2) 仿真试验二。在某些位置,钥匙处于4个锚点的视距范围内,即某些测距轮次下会得到4个有效测距值。但由于某些异常情况(如锚点硬件故障或被障碍物遮挡)会导致某时刻或某段时间内其中一个锚点的有效测距值短暂丢失。为了模拟以上情况,选取一组连续时间段内的测距值数组,并选定一个锚点将该锚点的测距值置零,表示该锚点在该时段因异常导致无法获得有效距离值。

3.3.2 仿真试验一结果

仿真试验一中 LS 与 LS-TS 模型的输出结果如图6所示。从图6可以看出,LS 模型输出的钥匙轨迹点坐标与实际轨迹点坐标相差较大,尤其在轨迹曲

率较大处,位置坐标的误差较大,且轨迹点显得杂乱不平滑。在 X 方向最大偏差为 0.481 5 m, Y 方向最大偏差为 0.320 2 m。由于 LS-TS 融合模型在 LS 模型基础上增加了约束方程,解算出的钥匙轨迹更加平滑,即使在轨迹曲率较大处,钥匙轨迹点坐标也并未发生杂乱,在 X 方向最大偏差减小为 0.275 8 m, Y 方向最大偏差减小为 0.235 7 m。与 LS 定位模型相比,LS-TS 融合模型的定位精度更高。

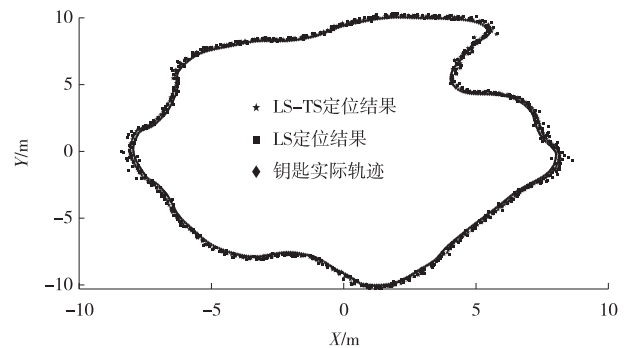


图6 仿真试验一中 LS 与 LS-TS 模型的输出结果

3.3.3 仿真试验二结果

当钥匙沿着图7所示矩形框内的轨迹移动时,正常情况下能够测得锚点1、3、4及主锚点与钥匙间的有效距离值。假设在这段时间内,锚点3由于硬件异常或信号遮挡,无法测得与钥匙间的有效距离值,PSO 模型与 LM 模型的输入只有锚点1、4及主锚点的测距值。

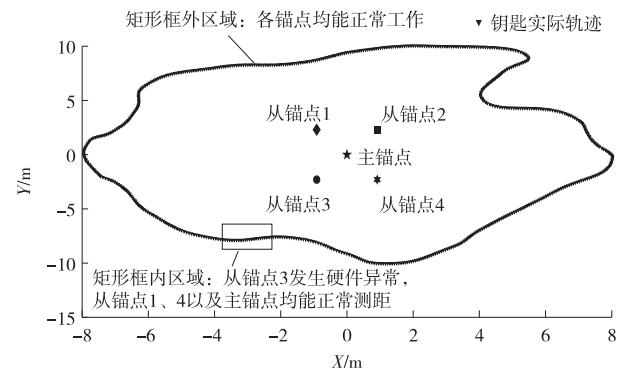


图7 仿真试验二测试区域

在各锚点正常工作的情况下,LS-TS 融合模型能够准确定位出钥匙坐标,如图8中矩形框以外的输出轨迹所示。一旦发生遮挡导致锚点数据异常,LS-TS 融合模型就会出现矩阵奇异,从而导致整段钥匙坐标

点的漂移失真, X 方向最大偏差达到 11.631 2 m, Y 方向最大偏差达到 6.064 8 m, 如图 8 中矩形框内的输出轨迹及其漂移轨迹。

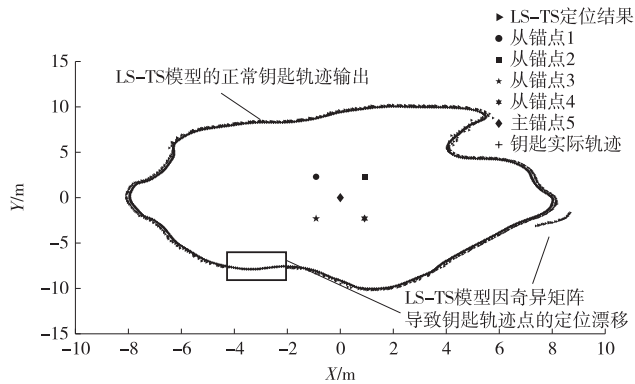


图8 仿真试验二中 LS-TS 模型的输出结果

由于仿真试验一的数据不存在矩阵接近奇异的情况, 因此只对比了 LS-TS 融合模型与 LS 模型的定位结果。在上述仿真试验二中, 对比了 LS-TS-PSO 融合算法与 LS-TS-LM 融合算法的定位结果。

3.3.4 融合模型

后面分别采用 PSO 模型和 LM 模型对异常情况下的 LS-TS 融合模型进行优化。在进行 PSO 仿真试验之前, 需确定模型的迭代次数以及种群规模。其中, 种群规模决定了空间中同时探索最优解的点的数量, 较大的种群规模意味着有更多的粒子在搜索空间中分布, 能够更广泛地覆盖搜索区域, 增加找到全局最优解的可能性。考虑到算法实时性与准确性问题, 最终选取迭代次数 (Iterations) = 50, 种群规模 (Scale) = 30 (即本文中设置钥匙的可能性坐标数为 30 个, 以搜索钥匙的最优坐标值)。

本文在 Iterations = 50, Scale = 30 的情况下进行了钥匙坐标的多次仿真试验, 发现每次仿真结果均有所不同, 其中误差最大的达到 1.27 m。选取其中 3 次进行展示, 如图 9 所示, 将图 9 中矩形区域内的轨迹点进行放大得到局部放大图 (图 10)。本文推理产生该结果的原因如下: PSO 模型在迭代开始时随机初始化每个粒子的位置, 即不同仿真试验开始时, 以及随后的迭代均会从不同的位置开始寻优, 在相同迭代次数的情况下就会发生最优解不同的情况。本文选取 10 次仿真试验中相对较好的 PSO 输出结果与 LM

模型输出的结果进行对比, 如图 11 所示。

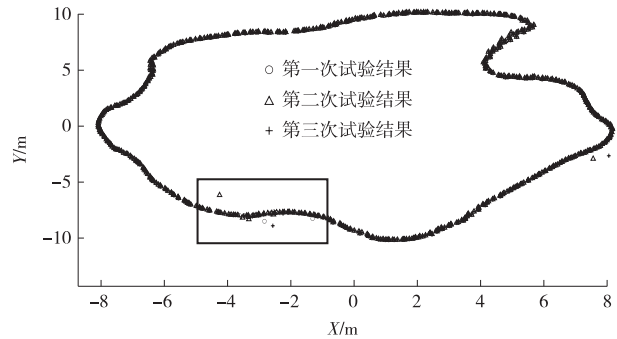


图9 相同 PSO 模型参数下的不同结果

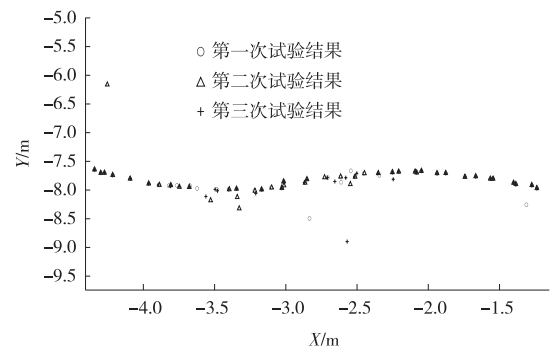


图10 图9中矩形框放大图

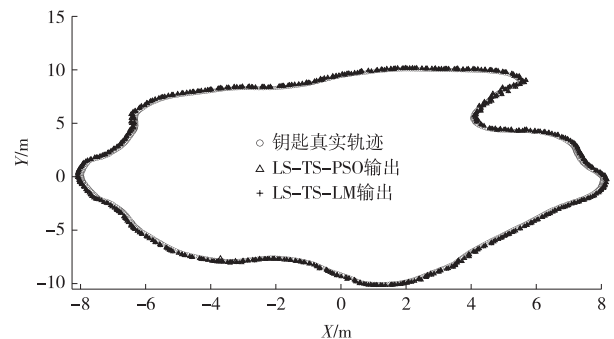


图11 PSO 与 LM 模型输出对比

从图 11 可以看出, 经两种寻优模型优化后, LS-TS 的定位稳定性得到极大增强。对两种模型逐点进行定位误差统计, LS-TS-LM 融合模型在 Y 方向上的最大误差为 0.113 01 m, 在 X 方向上的最大误差为 0.054 35 m; LS-TS-PSO 融合模型在 Y 方向上的最大误差为 0.268 61 m, 在 X 方向上的最大误差为 0.163 84 m。因此, LM 模型性能更优, 其基于梯度的寻优方式避免了 PSO 因随机初始化导致的结果波动, 鲁棒性更强。故接下来基于 LS-TS-LM 融合模型

进行硬件在环(Hardware-in-the-Loop,HIL)验证。

4 UWB 数字钥匙融合算法的 HIL 验证

结合 MATLAB/Simulink 与 UWB 数字钥匙电子控制单元(Electronic Control Unit,ECU)搭建 HIL 试验平台。ECU 的主控芯片采用 GDF103C8T6,系统运行时钟为 72 MHz,UWB 芯片采用 DWM1000 模组。具体试验过程如下:基于 MATLAB 平台编写串口通信脚本,实现与 ECU 的握手连接、时间同步、数据收发、数据存储等功能。脚本实现测距值集合由 MATLAB 传输至 ECU,ECU 每接收到 5 个有效测距值就迭代一次算法模型,迭代结束后再将钥匙坐标回传至 MATLAB,待 HIL 测试完成后,将所有结果在 MATLAB 中进行可视化显示。

基于仿真试验二的数据进行 HIL 验证,得到的结果如图 12、图 13 和图 14 所示。图 12 为 LS-TS 融合模型的仿真和 HIL 结果对比,图 13 为 LS-TS-LM 融合模型的仿真和 HIL 结果对比,图 14 为 LS-TS 模型和 LS-TS-LM 模型在仿真试验二数据下的迭代时间对比图。

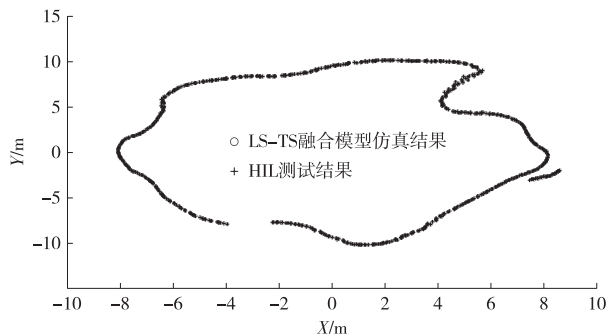


图 12 LS-TS 融合模型仿真结果和 HIL 测试结果

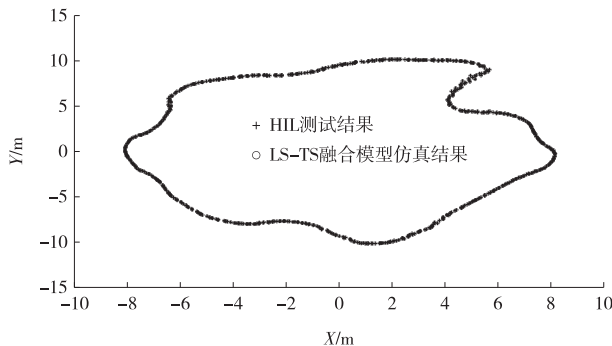


图 13 LS-TS-LM 融合模型仿真结果和 HIL 测试结果

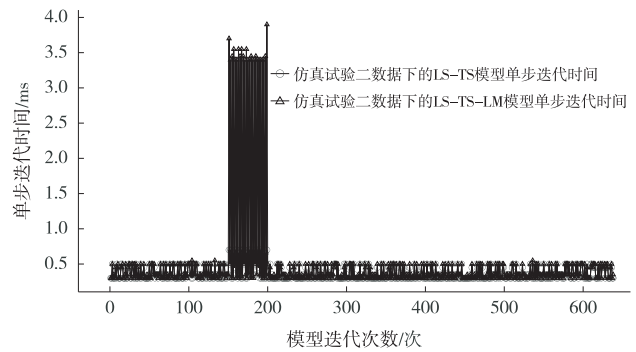


图 14 LS-TS 与 LS-TS-LM 迭代时间对比

1) 从图 12 可以看出,LS-TS 模型的 HIL 结果与仿真环境下的结果基本一致,在正常数据环境下能够保持较高的定位精度,但是一旦进入异常数据环境,模型的定位精度大幅下降。

2) 从图 13 可以看出,LS-TS-LM 模型的 HIL 结果与仿真环境下的结果基本一致,在异常数据环境下,依然能够保证钥匙坐标的精确输出。

3) 从图 14 可以看出,LS-TS 模型和 LS-TS-LM 模型在正常数据下的迭代时间基本一致,大约保持在每 0.55 ms 完成一次模型迭代。当系统运行至 1.51 s 时刻时,进入异常数据区域,LS-TS 模型和 LS-TS-LM 模型的最大迭代时间分别上升至 0.7 ms 和 3.9 ms。

4) 虽然 LM 算法的加入使系统资源消耗增加了约 3 ms,但由于车规级芯片运行频率普遍在百兆赫兹以上,而本次 HIL 试验所采用的系统板运行频率为 72 MHz,所以 LS-TS-LM 模型在极端数据环境下的单次迭代时间会 ≤ 3.9 ms。而人的平均步行速度为 0.001 46 m/ms,在毫秒级别下,人体移动状态基本可视为静止^[13]。因此,不论从定位精度还是定位实时性来看,本文构建的 LS-TS-LM 融合算法是有效的。

5 结束语

相对于传统采用 LS 模型实现钥匙定位的方法,本文提出一种融合 TS 和 LM 模型的定位解算方法,不仅能保证钥匙坐标解算精度,还能在个别锚点运行异常时维持数字钥匙定位解算系统的稳定性,具有重要的工程应用意义。

(下转第 23 页)

- [7] 中华人民共和国工业和信息化部. 电动汽车能量消耗量和续驶里程试验方法 第2部分: 重型商用车辆: GB/T 18386.2—2022[S]. 北京: 中国标准出版社, 2022: 1-15.
- [8] 中华人民共和国工业和信息化部. 重型混合动力电动汽车能量消耗量试验方法: GB/T 19754—2021[S]. 北京: 中国标准出版社, 2021: 1-26.
- [9] 企业标准“领跑者”工作委员会, 中国节能协会标准化专业委员会. “领跑者”标准评价要求 纯电动城市客车: T/CECA-G 0122—2021、T/CSTE 0118—2021[S]. 2021: 1-25.
- [10] 陶志军, 彭磊, 秦孔建, 等. 重型混合动力客车转鼓试验的阻力设定方法研究[J]. 汽车工程, 2011, 33(4): 361-364.
- [11] 尹涛, 郑义, 李晶, 等. 底盘测功机行驶阻力设定方法比较[J]. 小型内燃机与车辆技术, 2014, 43(1): 54-56.
- [12] 嵯爰红, 尚勇, 刘剑, 等. 重型电动汽车能耗测试阻力选取方法研究[J]. 客车技术与研究, 2019, 41(1): 42-44.
- [13] 马艳飞, 杨凡, 陈石人, 等. 底盘测功机不同模式对纯电动汽车经济性影响的研究[J]. 汽车实用技术, 2020(10): 20-25.
- [14] 唐保龙, 赵广成, 马超. 电动汽车的续驶里程及影响因素分析[J]. 汽车实用技术, 2022, 47(21): 25-31.
- [15] 李兰, 刘一鸣, 陈莹, 等. 基于缩短法城市客车能耗试验阻力选取方法研究[J]. 重型汽车, 2023(6): 11-12.
- [16] 周炳松, 吕良远, 罗彦, 等. 基于滑行阻力的轻型纯电动汽车续驶里程提升分析[J]. 客车技术与研究, 2023, 45(5): 27-30.
- [17] 中华人民共和国工业和信息化部. 重型商用车辆燃料消耗量测量方法: GB/T 27840—2011[S]. 北京: 中国标准出版社, 2011: 1-34.
- [18] 中国石油和化学工业联合会. 汽车轮胎滚动阻力试验方法 单点试验和测量结果的相关性: GB/T 29040—2012[S]. 北京: 中国标准出版社, 2012: 1-26.

(上接第14页)

参考文献:

- [1] 肖玉龙. 基于UWB技术的煤矿井下定位系统研究与设计[J]. 机械管理开发, 2022, 37(1): 201-203.
- [2] 张爱琳, 刘辉, 王小海, 等. 信号干扰下的超宽带精确定位问题研究[J]. 吉林大学学报(信息科学版), 2024, 42(2): 193-199.
- [3] 白晓娟, 道伟, 关露, 等. 室内复杂环境处理及泰勒公式改进定位算法研究[J]. 电子测量技术, 2021(3): 55-59.
- [4] 缪希仁, 范建威, 江灏, 等. 基站异常情况下基于改进极限学习机的超宽带室内定位方法[J]. 传感技术学报, 2020, 33(10): 1457-1466.
- [5] 肖庭杨, 章迪. 基于BP神经网络的厘米级超宽带测距误差改正模型设计与实验[J]. 全球定位系统, 2021, 46(4): 52-58.
- [6] 王成蓉. 基于超宽带定位系统的解析算法设计[J]. 电脑编程技巧与维护, 2019(1): 139-141.
- [7] 陈磊, 王菲菲, 焦良葆, 等. 基于最小二乘和泰勒级数的超宽带定位算法[J]. 信息化研究, 2020, 46(2): 30-35.
- [8] 任斌, 徐会彬. 基于总体最小二乘的泰勒级数展开的TOA的UWB定位方法[J]. 科学技术与工程, 2013, 13(21): 6129-6133.
- [9] 刘卫波, 张旭. 基于UWB技术的变电站三维定位系统设计与实现[J]. 粘接, 2024, 51(6): 142-145.
- [10] 龚纯, 王正林. 精通MATLAB最优化计算[M]. 北京: 电子工业出版社, 2009: 212-215.
- [11] 马光辉. 基于UWB的隧道内人员定位系统设计研究[D]. 石家庄: 石家庄铁道大学, 2023.
- [12] 周旷宇. 基于到达时间的UWB室内定位方法研究[D]. 重庆: 重庆交通大学, 2024.
- [13] 雷雪琪, 白玉, 张建波, 等. 面向步行交通品质提升的人行道宽度设计实证研究[J/OL]. 城市交通. <https://doi.org/10.13813/j.cn11-5141/u.2025.0010>.