

## 一种改进的 DV-HOP 算法在无线传感网中的定位研究

陈玲君

(绍兴职业技术学院, 浙江 绍兴 312000)

**摘要:** WSN 中的信息传递主要通过传感器来进行传递信号, 针对无线传感中 DV-Hop 算法在节点定位上存在精度低的问题, 首先提出建立双曲线二维模型用来确定锚节点与未知节点的距离关系, 其次设定误差系数使得传感器节点之间的误差降低, 最后采用斯蒂芬森迭代法 (Steffensen) 定位方法对传感器节点进一步进行定位修正; 仿真实验表明算法的在远程控制的中定位精度提高, 传感器之间能量消耗降低, 具有一定的推广价值。

**关键词:** 无线传感网络; 双曲线二维模型; 误差系数; 斯蒂芬森

## Research on Positioning of an Improved DV-HOP Algorithm in Wireless Sensor Networks

Chen Lingjun

(Shaoxing Vocational &amp; Technical College, Shaoxing 312000, China)

**Abstract:** In information transfer in WSN, signals are transferred mainly through the sensor. Aiming at the low accuracy of DV-Hop algorithm in node positioning in wireless sensor, this paper first proposes establishing hyperbolic two-dimensional model to determine the distance between anchor nodes and unknown nodes. Then, set the error coefficient to reduce errors between nodes of the sensor. Finally, Steffensen positioning method is used to modify the positioning of sensor's nodes. Simulation experiment shows that algorithm in this paper improves the positioning accuracy in remote control, and energy consumption between sensors is reduced, so it has certain promotion value.

**Keywords:** wireless sensor network; hyperbolic two-dimensional model; error coefficient; Steffensen

## 0 引言

伴随着微电子技术和计算无线通信技术的发展, 无线传感网络 (wireless sensor network, WSN) 得到了迅速的发展。传感器之间的信息传递主要依赖于节点位置的分布, 因此节点的准确定位是目前研究的主要的问题<sup>[1]</sup>。WSN 中的节点定位主要是指锚节点定位和未知节点定位, 其中锚节点是指少量的带有 GPS 定位装置的节点, 未知传感器节点位置的定位需要通过锚节点来进行自身的定位。国内外学者对于 DV-HOP 算法进行了广泛的研究, 文献 [2-3] 提出采用智能算法来对 DV-Hop 的定位结果进行了优化具有很好的稳定性; 文献 [4] 提出一种基于加权的三维 DV-Hop 定位算法, 该算法降低定位误差, 但是权值的设置需要进一步讨论; 文献 [5] 提出了将接收信号强度指示的改进算法, 降低了误差 30%~45%; 文献 [6] 提出根据邻居节点间的距离对跳数进行加权处理, 修正平均跳距, 缩小未知节点被包围的三角形区域, 筛选出最终进行坐标计算的锚节点, 其缺点是区域的设置受到其他条件的影响; 文献 [7] 提出了一种普适性反馈机制, 引入位置调整因子和基于 RSSI 的跳距优化算法, 实验表明, 在资源变化的网络计算环境中, 所提出的定位算法能够显著地提高定位精度, 且具有更强的稳定性; 文献 [8] 提出了对 DV-Hop 算法中的平均跳距值进行调整, 使得锚节点与未知节点之间的距离误差变小, 缺点是实际环境难以应用; 文献 [9] 提出了一种综合考虑所有信标节点的平均跳距并引入权值修正, 该算法使得网络平均跳距和节点位置估计值更加准确。文献 [10] 提出

从锚节点组合的有效性和节点早定位能减少因等待锚节点信息所耗的时间和能量两方面考虑, 并进行改进, 仿真实验表明算法效果良好。

为了进一步提高传感器定位的精度, 本文在 DV-HOP 算法的基础上引入斯蒂芬森迭代法 (Steffensen) 对传感器节点定位进行修正。分析 DV-Hop 算法存在的不足, 建立双曲线二维方程, 通过设定误差系数降低偏差, 采用 Steffensen 迭代法对 DV-Hop 算法定位结果进行检验和修正, 仿真实验表明本文的算法能够有效的提高定位性能。

## 1 DV-Hop 算法及误差分析

## 1.1 DV-Hop 定位算法

DV-Hop 算法的具体步骤如下:

- 1) 锚节点向周围的节点广播数据包, 未知节点收到锚节点的最小跳数, 进行加 1 后转发给下一个节点。
- 2) 锚节点收到其他节点的信息和跳数后, 计算平均每跳的距离进行估算, 按照公式 (1)。

$$HopSize_i = \frac{\sum_{j \neq i} \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}{\sum_{j \neq i} hopS_{ij}} \quad (1)$$

式中, 分别是锚节点  $i$  和  $j$  的坐标,  $hopS_{ij}$  是两个锚节点  $i$  和  $j$  间的最小跳数。

- 3) 锚节点与未知节点之间估计距离为:

$$L_i = S_i \times HopSize \quad (2)$$

根据未知节点与各锚节点间的距离, 采用多边形法计算未知节点的坐标, 具体为:

$$\begin{cases} (x_i - x_1)^2 + (y_i - y_1)^2 = L_1^2 \\ (x_i - x_2)^2 + (y_i - y_2)^2 = L_2^2 \\ \vdots \\ (x_i - x_j)^2 + (y_i - y_j)^2 = L_j^2 \end{cases} \quad (3)$$

收稿日期:2015-11-30; 修回日期:2015-01-04。

基金项目:国防预先研究项目(51327030104)。

作者简介:陈玲君(1983-),女,讲师,硕士,主要从事智能控制方向的研究。

式中,  $(x_i, y_i)$  为未知节点的坐标;  $(x_1, y_1), (x_2, y_2), \dots, (x_j, y_j)$  为该未知节点所记录锚节点的坐标。因此未知节点  $(x_i, y_i)$  通过公式 (4) 得到

$$P = (A^T A)^{-1} A^T B \quad (4)$$

$$\text{设 } A = -2 \begin{bmatrix} x_i - x_1 & y_i - y_1 \\ x_i - x_2 & y_i - y_2 \\ \dots & \dots \\ x_i - x_n & y_i - y_n \end{bmatrix}$$

$$B = \begin{bmatrix} d_n^2 - d_1^2 + x_1^2 - x_n^2 + y_1^2 - y_n^2 \\ d_n^2 - d_2^2 + x_2^2 - x_n^2 + y_2^2 - y_n^2 \\ \dots & \dots \\ d_n^2 - d_i^2 + x_i^2 - x_n^2 + y_i^2 - y_n^2 \end{bmatrix}$$

$$P = \begin{bmatrix} x \\ y \end{bmatrix}$$

1.2 误差分析

在 DV-HOP 算法中主要采用锚节点之间的距离来近似估算未知节点到锚节点之间的距离, 但具有一定的局限性, 主要是因为取决于锚节点是否分布均匀, 但实际情况恰恰相反, 锚节点的定位受到很多因素的影响, 当网络中的锚节点之间分布密集的时候, 网络中的未知节点之间的距离接近于实际距离, 但当锚节点分布稀疏的时候, 偏差就容易产生, 如图 1 所示。

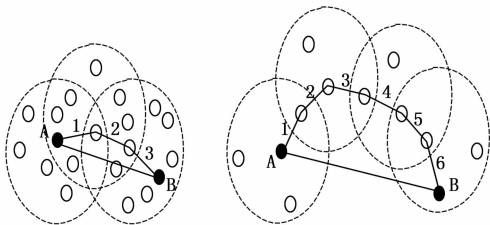


图 1 节点分布造成的误差

因此 DV-Hop 算法测量所得到距离  $L$  与实际距离之间必然存在一定误差, 假设  $f_n$  为未知节点到锚节点之间的测距误差系数, 则未知节点到锚节点的距离为公式 (5) 所示

$$(x_i - x_j)^2 + (y_i - y_j)^2 = L_j^2 + f_j \quad (5)$$

根据公式 (3) 得到未知节点坐标满足下式:

$$\begin{cases} (x_i - x_1)^2 + (y_i - y_1)^2 = L_1^2 + f_1 \\ (x_i - x_2)^2 + (y_i - y_2)^2 = L_2^2 + f_1 \\ \dots \\ (x_i - x_j)^2 + (y_i - y_j)^2 = L_j^2 + f_j \end{cases} \quad (6)$$

求解未知节点坐标  $(x_i, y_i)$ , 使得

$$f(x_i, y_i) = \sum_{j=1}^n (\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} - L_j) \quad (7)$$

当 (7) 式取得最小值时, 总误差最小, 未知节点的定位就能接近真实值。即满足下式的未知节点坐标:

$$f(x_i, y_i)_{\min} = \min(\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} - L_j) \quad (8)$$

通过上述方式把 DV-Hop 定位误差问题成功地转化一个非线性优化问题, 通过斯蒂芬森迭代法对式 (8) 进行求解, 进一步优化传感器定位精度。

2 WSN 中的改进的 DV-Hop 算法

2.1 基于双曲线的二维平面

为了降低研究的复杂度, 本文选择在二维平面作为研究对象, 假设在有  $M$  个传感器节点, 第  $i$  个未知节点坐标为  $M_{i,j}$

$= [x_i, y_i]^T (i = 1, 2, \dots, n, j = 1, 2, \dots, n)$ , 锚节点的坐标为  $S_p = [a, b]^T$ , 因此每一个节点发送源与传感器之间的距离为  $\|M_i - S_p\|$ 。以选取其中第  $j$  个未知节点作为参考节点, 通过观测值  $\kappa_{i,j} (i = 1, \dots, N)$  用来表示在锚节点与未知节点之间的关联关系, 设定关于锚节点与未知节点之间的关系为双曲线方程, 如图 2 所示。

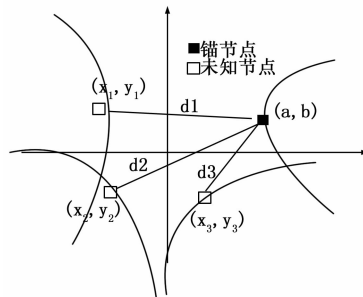


图 2 WSN 中的双曲线定位

设定信号之间的传播速度为  $v$ , 则锚节点与未知节点之间的距离为  $d = v\kappa_{i,j}$ , 构建以锚节点为中心的构建  $N$  个双曲线方程  $d_{i,j} = d_i - d_j$ , 通过对  $N$  个方程求解得到定位的节点坐标。由于该方程式非线性方程, 所以, 需要对其进行线性处理, 由  $d_i^2 = (d_{i,j} + d_j)^2$  进行拓展得到如下:

$$d_{i,j}^2 + 2d_{i,j}d_j + 3d_j^2 = M_{i,j}^2 + S_p - 2SM_{i,j} \quad (9)$$

对公式 (9) 进行局部处理, 得到公式 (10)

$$M_{i,j}^2 + S_p^2 - 2M_{i,j}S_p = d_i^2 \quad (10)$$

将公式 (10) 带入公式 (9) 得到如下:

$$M_{i,j}^2 - S_p^2 - 2(M_{i,j} - M_{1,1})^T S_p = d_{i,j}^2 + 2d_{i,j}d_j \quad (11)$$

通过公式 (11), 将问题转换为线性方程的求解, 代入 (8) 进行求解。

2.2 误差系数设定

由于锚节点与未知节点的实际距离与理论距离肯定存在误差, 因此通过设定误差系数来减少, 之前描述了未知节点与锚节点之间的实际距离为  $D_{i,j}$ ,

$$D_{i,j} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (12)$$

根据公式 (12) 和公式 (3) 得到未知节点到锚节点之间的总误差  $\epsilon$ , 如公式 (13), 误差系数的确定如公式 (14)

$$\epsilon = \sum_{i \neq j} D_{ij} - L_{ij} \quad (13)$$

$$f_{ij} = \frac{\epsilon}{n * (n - 1) / 2} \quad (14)$$

式 (14),  $n * (n - 1) / 2$  表示  $n$  个节点之间的跳数总和,  $f_{ij}$  表示误差系数

2.3 Steffensen 算法

对于非线性方程  $f(x) = 0$ , 可化成等价的方程:

$$x = \varphi(x) \quad (15)$$

式中,  $\varphi(x)$  称为迭代函数。

Steffensen 迭代法是一种改进的不动点迭代法, 具有二阶收敛速度、计算精度等优点, 迭代函数为:

$$\varphi(x) = x - \frac{(\varphi(x) - x)^2}{\varphi(\varphi(x)) - 2\varphi(x) + x} \quad (16)$$

Steffensen 迭代公式为:

$$\begin{cases} x_k = \varphi(x_k), z_k = \varphi(y_k) \\ y_{k+1} = x_k - \frac{(y_k - x_k)^2}{z_k - 2y_k + x_k} \end{cases}, k = 0, 1, 2, \dots \quad (17)$$

Steffensen 算法流程如下：

- 1) 设置最大迭代次数  $N$  和精度  $\epsilon$ , 令  $k = 0$ 。
- 2) 计算  $y_k = \varphi(x_k)$ ,  $z_k = \varphi(y_k)$ 。
- 3) 若  $|x_{k+1} - x_k| < \epsilon$  则停止计算。
- 4) 若  $k = N$ , 则停止计算; 否则, 置  $k = k + 1$ , 转到步骤 2)。

### 2.4 Steffensen 修正 DV-Hop 定位误差

除了对误差系数进行设定之外, 还需要 (8) 取得最小值的必要条件是:

$$\frac{\partial f}{\partial x_i} = \frac{\partial f}{\partial y_i} = 0, (m < i \leq n) \quad (18)$$

其中:

$$\begin{cases} \frac{\partial f}{\partial x_i} = \sum_{m < i \leq n} \sum_{1 \leq j \leq m} \left( 2(x_i - x_j) - \frac{2l_{ij}(x_i - x_j)}{\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}} \right) \\ \frac{\partial f}{\partial y_i} = \sum_{m < i \leq n} \sum_{1 \leq j \leq m} \left( 2(y_i - y_j) - \frac{2l_{ij}(y_i - y_j)}{\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}} \right) \end{cases} \quad (19)$$

$$\text{令: } \Delta_i = \sqrt{\left(\frac{\partial f}{\partial x_i}\right)^2 + \left(\frac{\partial f}{\partial y_i}\right)^2} \quad (20)$$

则求  $\min(\text{fitness}())$  的最优解问题转化为求使得  $\Delta_i$  取到最小值时未知节点的坐标  $(x_i, y_i)$  的问题。具体步骤为:

- 1) 按照 Steffensen 迭代公式 (17) 的第一步, 对变量  $x_i$  和  $y_i$  进行迭代, 有:

$$\begin{cases} \bar{x}_i = \varphi(x_i) = \frac{\sum_{1 \leq j \leq m} \left( x_j + \frac{l_{ij}(x_i - x_j)}{\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}} \right)}{m} \\ \bar{y}_i = \varphi(y_i) = \frac{\sum_{1 \leq j \leq m} \left( y_j + \frac{l_{ij}(y_i - y_j)}{\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}} \right)}{m} \end{cases} \quad (21)$$

$$\begin{cases} \hat{x}_i = \varphi(\bar{x}_i) = \frac{\sum_{1 \leq j \leq m} \left( x_j + \frac{l_{ij}(\bar{x}_i - x_j)}{\sqrt{(\bar{x}_i - x_j)^2 + (y_i - y_j)^2}} \right)}{m} \\ \hat{y}_i = \varphi(\bar{y}_i) = \frac{\sum_{1 \leq j \leq m} \left( y_j + \frac{l_{ij}(\bar{y}_i - y_j)}{\sqrt{(x_i - x_j)^2 + (\bar{y}_i - y_j)^2}} \right)}{m} \end{cases} \quad (22)$$

式中,  $x_i$  和  $y_i$  为采用 DV-Hop 算法的传感器节点估计位置, 将其作为迭代初值。

- 2) 根据式 (17) 对未知节点坐标进行更新迭代操作:

$$\begin{cases} x_i' = \bar{x}_i - \frac{(\bar{x}_i - \hat{x}_i)^2}{\bar{x}_i - 2\hat{x}_i + x_i} \\ y_i' = \bar{y}_i - \frac{(\bar{y}_i - \hat{y}_i)^2}{\bar{y}_i - 2\hat{y}_i + y_i} \end{cases} \quad (23)$$

- 3) 将新的未知传感器节点的坐标  $(x_i, y_i)$  代入式 (20) 中, 算出  $\Delta_i$  的值。

- 4) 当迭代次数达到所设的最大值, 迭代算法停止, 求出最接近真实值的未知节点的坐标, 使得定位误差最小。

### 3 算法仿真与分析

进行仿真实验的硬件选择 CPU 为酷睿 i3, 4GDDR3 和硬盘为 500 G, 软件采用 matlab2012。采用本文算法与基本 DV-HOP 算法以及文献 [2] 算法进行比较。选择  $100 \text{ m} \times 100 \text{ m}$  的区域, 选择 1000 个节点, 其中 220 个为锚节点, 780 为未知节点, 所有节点都进行随机分布。采用误差计算公式 (24)

作为比较的标准, 其中  $(x'_i, y'_i)$  和  $(x_i, y_i)$  分别表示估算未知和实际未知,  $R$  为节点通信半径,  $N$  为节点个数, 结果如下

$$f = \frac{\sum_{i \neq j} \sqrt{(x'_i - x_j)^2 + (y'_i - y_j)^2}}{R \cdot N} \quad (24)$$

图 3 表示在不同的节点密度与定位误差关系。从图中发现 3 种算法伴随着节点密度的增大都对定位误差进行了优化, 从中发现本文算法所产生的误差是最小, 相比与 DV-HOP 算法平均优化了 21.2%, 相比与文献 [2] 算法优化了 7.23%, 这说明本文算法能够获得更精确的定位结果。

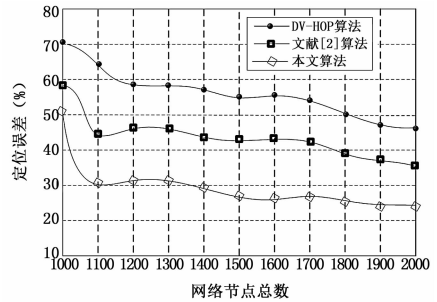


图 3 节点密度对定位误差的影响

图 4 表示在不同的节点通信半径与定位误差的关系。节点通信半径关系到未知节点定位准确性, 选择节点半径从 5 到 15 米的范围, 从中可以发现 3 种算法定位误差都伴随着节点通信半径不断增大而减少, 这说明 3 种算法都的改进都改善了与节点通信半径之间的关系, 本文算法曲线平缓, 说明节点半径的增大不影响算法的稳定性, 因此本文算法的定位误差相比与其他两种算法能够更好的完成更精确定位。

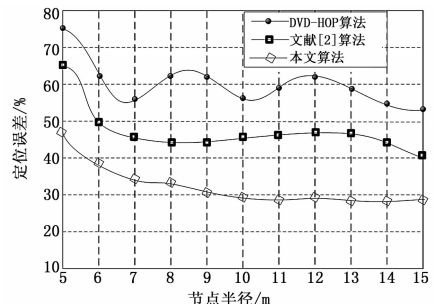


图 4 节点通信半径对定位误差的影响

图 5 表示锚节点比例与定位误差的关系。3 种算法中锚节点的比例按照 2% 到 20% 的顺序增加, 从图中发现本文的算法相比于其他两种算法在锚节点比例不断增大的情况下, 定位误差相比与 DV-HOP 算法降低了 12.3%, 相比与文献 [2] 算法降低了 5.2%, 这说明锚节点的数量增多的情况, 本文算法具有更加明显的定位优势。

图 6 表示节点能量消耗与定位误差的关系。从图中发现, 本文算法相比于 DV-Hop、文献 [2] 算法的网络能量消耗都伴随着节点数量的不断增多而降低, DV-Hop 算法的能量消耗曲线振幅最大, 说明能量消耗不稳定, 并且消耗最快, 文献 [2] 算法次之, 本文算法的能量消耗曲线平缓。说明本文算法在一定程度上可以有效提高网络能量利用率。

### 4 结束语

针对 WSN 中的 DV-Hop 算法定位的问题, 本文在 DV-

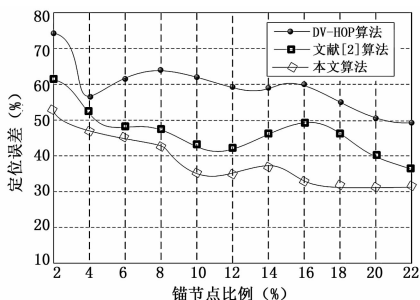


图 5 锚节点的比例对定位误差的影响

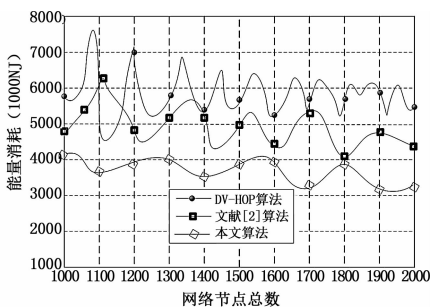


图 6 节点能量消耗

HOP 算法的基础上, 引入双曲线二维平面方程来定位锚节点与未知节点的关系, 其次通过引入误差定位系数降低距离误差, 采用 Steffensen 迭代法对 DV-Hop 算法定位结果进行分析

(上接第 304 页)

### 4 试验结果与分析

通过矢量信号源模拟超高频 RFID 读写器对 RFID 检测仪发送激励信号 (单次查询命令), 同时触发矢量信号分析仪接收并分析 RFID 检测仪回传信号, 对 RFID 检测仪的支持频率范围、接收灵敏度、最大发射功率、支持调制解调方式、反向链路频率误差等参数进行测试, 测试结果如表 1 所示。

表 1 测试项目及测试结果

测试项目	测试结果
支持频率范围/MHz	840~845, 920~925
接收灵敏度/dBm	-110
最大发射功率/dBm	30
支持调制解调方式	ASK, BPSK
反向链路频率误差	±12%

通过测试结果可以看出, 作为手持式设备的 RFID 检测仪在小体积、低功耗等条件制约下, 关键指标均满足或优于超高频 RFID 通信协议<sup>[11]</sup>的要求。

### 5 结束语

RFID 检测仪是一款针对超高频 RFID 读写器射频参数测试和协议一致性测试的现场专用检测设备, 其检测项目的完备性和方便性是其他检测手段所不具备的。该检测仪既可用于超高频读写器的研制生产过程中的检测环节, 又方便现场维护人员的携带和对超高频 RFID 读写器的现场测试。该产品大大减少测试仪器资源的占用量, 极大地降低测试设备的购置成本, 目前已经在超高频读写器的生产商和最终用户得到较好的应

和修正, 实现精确定位。通过在仿真实验中 3 种算法的比较, 本文的算法能够有效的提高了定位精度, 具有一定的实用性。

### 参考文献:

- [1] 黄毅, 胡爱群. 无线传感器网络定位算法综述 [J]. 电信科学, 2010, 20 (7): 69-74.
- [2] 曹欲晓, 张倩, 李艳冰. 基于蝙蝠算法的 DV-Hop 定位改进 [J]. 计算机测量与控制, 2015, 23 (4): 1273-1275.
- [3] 张万礼, 宋启祥. 遗传算法的 DV-Hop 算法改进 [J]. 重庆大学学报, 2015, 38 (3): 159-166.
- [4] 李琳, 赵可, 林志贵. 基于加权的三维 DV-Hop 定位算法 [J]. 控制工程, 2015, 22 (4): 761-764.
- [5] 郭春明, 张金强, 焦龙. 基于 RSSI 跳数连续的 DV-HOP 改进算法 [J]. 重庆邮电大学学报 (自然科学版), 2015, 27 (2): 184-187.
- [6] 蔡丽萍, 蒋晶晶, 洪利. 一种改进的 DV-HOP 定位算法 [J]. 微电子学与计算机, 2015, 32 (4): 120-124.
- [7] 柳飞, 鄢广增. 基于 RSSI 和反馈机制的无线传感器网络定位优化算法 [J]. 南京邮电大学学报 (自然科学版), 2015, 35 (2): 108-113.
- [8] 苏兵, 薛伟杰, 王洪元. 一种 WSN 非测距定位 DV-Hop 算法的误差改进方法 [J]. 计算机测量与控制, 2013, 21 (5): 1357-1359.
- [9] 周杭霞, 崔晨, 叶佳骏. 一种基于加权处理和误差修的 DV-Hop 定位算法研究 [J]. 传感技术学报, 2014, 27 (12): 1699-1703.
- [10] 郑远, 蔡宇, 赵锐. 一种兼顾性能与能耗的 DV-Hop 改进算法 [J]. 计算机应用与软件, 2014, 31 (4): 269-272.

用。手持式超高频 RFID 专用检测仪的出现, 对于超高频 RFID 读写器的发展起到很好的促进作用, 其必然会在工业生产、现场维护等领域呈现出良好的市场发展前景。

### 参考文献:

- [1] 游战清, 李苏剑. 无线射频识别技术 (RFID) 理论与应用 [M]. 北京: 机械工业出版社, 2004.
- [2] 张殿东. 无线射频识别 (RFID) 技术 [J]. 电信技术, 2005 (2): 85-89.
- [3] 文浩. 2007 中国 RFID 发展分析报告 [J]. 中国电子商情—RFID 技术与应用, 2008, (2): 50-54.
- [4] 李鹰, 李倩, 朱建红, 等. RFID 系统测试标准体系研究 [J]. 电子产品可靠性与环境试验, 2010, 28 (2): 43-47.
- [5] 李军, 何婷婷, 陈柯. 超高频 RFID 标准和测试技术演进 [J]. 国外电子测量技术, 2015, 34 (9): 13-16.
- [6] 程俊, Peter Chen. 泰克实时频谱仪在 RFID 测试中的应用 (上) [J]. 中国电子商情 (RFID 技术与应用), 2007, 2: 57-61.
- [7] Dobkin D M. The RF in RFID; passive UHF RFID in practice [M]. New York: Elsevier Inc. 2007.
- [8] 谈熙, 常若艇. 超高频射频识别系统信号特征分析及应用 [J]. 通信学, 2009, 30 (6): 89-91.
- [9] 王秀敏, 高凌鸿. 一种新的基于 ARM 的 RFID 测试系统设计研究 [J]. 计算机测量与控制, 2012, 20 (8): 2045-2047.
- [10] 黄安琪, 冯超, 孙建, 等. 基于软件无线电平台的 RFID 被动侦测技术 [J]. 计算机工程, 2014, 40 (4): 91-95.
- [11] EPC global. EPC™ Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860 MHz-960 MHz Version 1.0.9 [S]. EPC global, Jan. 2005.