基于重力方向的三维空间中传感器网络的分布式自定位算法

First A. Author*. Second B. Author, Jr.**
Third C. Author***

**National Institute of Standards and Technology, Boulder, CO 80305 USA (Tel: 303-555-5555; e-mail: author@ boulder.nist.gov).*
***Colorado State University, Fort Collins, CO 80523 USA (e-mail: author@lamar. colostate.edu)*
**** Electrical Engineering Department, Seoul National University, Seoul, Korea, (e-mail: author@snu.ac.kr)}*

Abstract: xx.

Keywords: Registration, Scan-matching, Lidar, Transformation.

1. INTRODUCTION

results.

2. 问题描述和预引理

2.1 预引理

首先我们将介绍一些图论和代数图论中的基本概念；有向图$G=[V,E]$包含非空节点集合 V 和边集合$E\in V\times V$；如果(j,i)是有向图 G 的一条边，那么节点 j 是节点 i 的内邻居，同时节点 i 是节点 j 的内邻居；我们定义节点 i 的内邻居集合为$N_i=\{j\in V,(j,i)\in E\}$；下面我们介绍复数 Laplacian 矩阵，这个矩阵和有向图有关；对于一个有向图，对该有向图的每条边$(j,i)\in E$都赋予一个复数的权重ω_{ij}，那么对应的复数 Laplacian 矩阵 L 中的元素L_{ij}定义为如下；

$L_{ij}=$

$-\omega_{ij}, \text{if } i\neq j \text{ and } j\in N_i$

$0, \text{if } i\neq j \text{ and } j\notin N_i$

$\sum_{j\in N_i}\omega_{ij}, \text{if } i=j$

2.2 问题描述

设Σ_g是一个公共的全局坐标系；下面我们讨论一个由两组传感器节点集合定义的传感器网络；第一组传感器集合称为锚节点（记为$1,\dots,m$）；在全局坐标系Σ_g中，这些节点的坐标定义为$p_a=[p_1,\dots,p_m]\in R^{3\times m}$，并且这些坐标是已知的；另一组传感器集合称为自由节点（记为$m+1,\dots,m+n$）；在全局坐标系Σ_g中，这些节点的坐标定义为$p_s=[p_{m+1},\dots,p_{m+n}]\in R^{3\times n}$，并且这些坐标是未知的；这些传感器节点的坐标$p_i\in R^3$位于三维空间；我们设$p=[p_a,p_s]\in R^{3\times(m+n)}$为所有传感器节点的累计坐标，并且将这个坐标称为传感器网络的 configuration；

在该论文中我们对所有传感器节点的累计坐标$p=[p_a,p_s]$（或者称为传感器网络的 configuration）有如下的假设；传感器网络的 configuration 是 generic；传感器网络的累加坐标 p（或者称为 configuration）是 generic 的条件是所有坐标（$p_1,\dots,p_{m+n}$）不满足任何的整数系数的非平凡方程[31]；直观描述就是一个 generic 的传感器节点的 configuration 是没有退化的；比如没有任何第三个点在同一条直线等；该假设的依据是传感器网络的节点位置是通过实际的传感器测量得到的（带有噪声）；

自由节点 i 的邻居节点（包括自由节点和锚节点）定义为集合N_i；自由节点 i 可以测量到每个邻居节点$j\in N_i$在局部坐标系Σ_i中的相对位置，记为$q_{ij}=\left[x_{ij},y_{ij},z_{ij}\right]^T$；局部坐标系$\Sigma_i$以节点 i 为原点，并且$\Sigma_i$与$\Sigma_g$之间的坐标轴角度之间存在一个任意的旋转矩阵 R；相对位置q_{ij}的含义是节点 j 在节点 i 的局部坐标系中的坐标是$q_{ij}=\left[x_{ij},y_{ij},z_{ij}\right]^T$；

我们构建了一个有向图$G=[V,E],V=\{1,\dots,m+n\}$来描述这些传感器的拓扑关系；其中，边$(j,i)\in E$表示节点 i 可以测量得到相对位置$q_{ij}$；那么传感器网络可以使用一个元组定义$(G,p_a,p_s)$；传感器网络$(G,p_a,p_s)$的相对位置定位问题定义如下；

定义 1，给定一个传感器网络(G,p_a,p_s)和基于局部坐标系的相对位置测量$q_{ij},(j,i)\in E$，寻找所有自由节点p_s的坐标；并且存在一个局部坐标系Σ_i相对于全局坐标系Σ_g的旋转矩阵R_i如下；

$$q_{ij}=R_i\left(p_j-p_i\right), j\in N_i$$

使用传感器网络(G,p_a,p_s)得到的测量信息，可以计算唯一解；

定义 2，一个传感器网络(G,p_a,p_s)是联合可定位的条件如下；给定锚节点p_a的坐标和局部坐标系中测量到的相对位置$q_{ij},(j,i)\in E$，该相对位置测量问题具有唯一解；

定义 2 可能意味着需要得到所有锚节点的位置和所有相对位置测量信息来求解该相对位置测量问题；然而为了

减少信息交换和使得每个自由节点能够使用分布式计算的方式求解自己的问题，我们提出另一个称为自定位性的概念；下面的定义中我们将一些几个新的概念；设U_s是任意自由节点的集合；然后所有其他传感器节点的集合定义为$U_a = V \backslash U_s$（包括锚节点）；然后我们定义p_{U_a}和p_{U_s}分别是传感器集合U_a和U_s的坐标向量；

定义 3，一个传感器网络(G, p_a, p_s)是自定位性的条件如下；在给定全局坐标系Σ_g中p_{U_s}的坐标和局部坐标系中的相对位置测量$q_{ij}, i \in U_s, (j, i) \in E$，对于任意的自由节点$U_s$，该传感器网络都要是联合可定位的；

定义 3 意味着如果某个自由节点得到他的内邻居在全局坐标系Σ_g中的坐标，那么该自由节点就必然可以计算得到自己在全局坐标系Σ_g中的坐标；我们之所以要提出自可定位性这个概念是因为这个概念在后续完全分布式定位的算法中将要使用到；

该论文主要讨论自可定位性问题，并且提出一种分布式的算法来计算每个自由节点在全局坐标系Σ_g中的位置（如果传感器网络是自可定位的）；

3. 二维平面上的相对位置定位算法

文献[1]给出了该问题在二维情况中的分布式迭代算法；当前论文中，我们使用该文献中的方法为基础，将算法推广到三维空间中；所以本节简要描述文献[1]中该问题在二维情况下的计算方式；

3.1 变量定义

相对位置定位算法在二维和三维情况中，问题描述大致上是相同的；下面我们给出该问题在二维情况下的一些变量定义；在二维情况下，传感器节点 i 的坐标表示为$p_i^{2d} \in C$；所有传感器节点的坐标使用复数表示，复数的实部表示传感器节点在 x 轴的坐标，复数的虚部表示传感器节点在 y 轴的坐标；在全局坐标系Σ_g，锚节点集合的坐标定义为$p_a^{2d} = \left[p_1^{2d}, \ldots, p_m^{2d}\right]^T \in C^n$；自由节点集合的坐标定义为$p_s^{2d} = \left[p_{m+1}^{2d}, \ldots, p_{m+n}^{2d}\right]^T \in C^m$；我们设$p^{2d} = \left[{p_a^{2d}}^T, {p_s^{2d}}^T\right]^T \in C^{m+n}$为所有传感器节点的累计坐标；自由节点 i 可以测量到每个邻居节点$j \in N_i$在局部坐标系Σ_i中的相对位置，记为$r_{ij} \in C$；

$$r_{ij} = \rho_{ij} e^{i\theta_{ij}}$$

其中ρ_{ij}是节点 j 到节点 i 的距离，θ_{ij}是节点 j 在局部坐标系Σ_i的角度；

3.2 自定位性条件

定理 1，在假设 1 的条件下，在二维空间中一个传感器网络(G, p_a, p_s)是自可定位性的充要条件如下；

(NS-1)，锚节点的数量$m \geq 2$；

(NS-2)，每个自由节点都是从锚节点 2-可达的；

3.3 权重计算

根据该论文的假设，每个自由节点可以在自己的局部坐标系得到相对位置测量；然后根据定理 1，如果一个传感器网络是可自定位的，那么每个自由节点都是从锚节点 2-可达的；那么每个自由节点必然有 2 个内邻居；因此，使用所有的内邻居数据，每个自由节点可以使用如下式子计算权重；

$$\sum_{j \in N_i} \omega_{ij} r_{ij} = 0$$

其中$r_{ij}, j \in N_i$是在节点 i 测量得到的数据；当$|N_i| = 2$，参数ω_{ij}和ω_{ik}如下；比如节点 j 是节点 i 的内邻居，那么参数如下；

$$\omega_{ij} = \frac{e^{-i\theta_{ij}}}{\rho_{ij}}$$

然后节点 k 是节点 i 的另一个内邻居，那么参数如下；

$$\omega_{ik} = -\frac{e^{i\theta_{ik}}}{\rho_{ik}}$$

当$|N_i| > 2$，那么公式(8)的解空间的数量是大于 1；此时我们可以随机选择一组满足式子[8]的参数$\omega_{i,j1}, \ldots, \omega_{i,jk}$，只是要求每个参数都不为 0 即可；例如我们可以设$\omega_{i,j1}, \ldots, \omega_{i,j(k-1)}$都为 1，然后使用公式[8]计算剩下的参数$\omega_{j,ik}$；只要$\omega_{j,ik}$不为零，这组参数就是可行的；

因为锚节点没有内邻居，那么锚节点没有测量其他节点的相对位置；然后我们可以将所有节点的的权重关系写成如下矩阵形式；

$$Lp^{2d} = 0$$

其中$p^{2d} = \left[{p_a^{2d}}^T, {p_s^{2d}}^T\right]^T$是所有节点的坐标向量；矩阵 L 是图 G 的 Lapalcian 矩阵，并且矩阵 L 的形式如下；

$$L = \begin{bmatrix} 0 & 0 \\ B & H \end{bmatrix}$$

上述方程展开如下；

$$Hp_s^{2d} + Bp_a^{2d} = 0$$

如果一个传感器网络是可自定位的，那么根据定理 1 和引理 1，矩阵 H 是非奇异的；

3.4 存在噪声测量时的自定位算法

在该节中，我们使用一种分布式算法来计算每个自由节点在全局坐标系Σ_g中的坐标（传感器网络是自可定位的）；当使用存在噪声的相对位置测量和存在舍入误差的情况下计算权重ω_{ij}，那么上述线性方程的形式如下；

$$(H - \Delta_H)p_s^{2d} + (B - \Delta_B)p_a^{2d} = 0$$

其中矩阵 H 和 B 是通过公式(8)计算得到；矩阵Δ_H和Δ_B是因为测量噪声和舍入误差造成的误差矩阵；我们定义$v = \Delta_H p_s + \Delta p_a$，那么上述方程变换为如下；

$$Hp_s^{2d} = -Bp_a^{2d} + v$$

为了简化分析，我们假设$v \sim CN(0,\Sigma)$；v 服从均值为零并且方差为Σ的复数高斯分布；因为不同节点的测量和计算都是独立的过程，方差Σ是对角矩阵并且每个节点知道对应的第 i 个节点在Σ中的分量σ_i；使用最小二乘估计理论，可以得到上述加权最小二乘估计的最优解如下；

$$p_s^{2d} = \Phi^{-1}\theta$$

$$\Phi = H^T\Sigma^{-1}H$$

$$\theta = -H^T\Sigma^{-1}Bp_a^{2d}$$

然后我们给出一种分布式定位方案；该算法可以从任意的起始位置开始，逐渐收敛到最优解；首先定义如下矩阵；

$$\widehat{H} = \Sigma^{\frac{1}{2}}B$$

$$\widehat{B} = \Sigma^{\frac{1}{2}}B$$

通过引入辅助变量$\zeta \in C^n$，我们提出如下的迭代算法；

$$\zeta\,(t+1) = \widehat{H}\hat{p}_s^{2d}\,(t) + \widehat{B}p_a^{2d}$$

$$\hat{p}_s^{2d}\,(t+1) = \hat{p}_s^{2d}\,(t) - \epsilon\widehat{H}^T\zeta\,(t)$$

上述迭代算法可以在每个节点中采用分布式的方式实现；该算法中要求每个节点与邻居节点之间存在互相通信；在实际场景中，通信范围通常大于传感器的测量范围；那么通常情况下，如果节点 i 可以测量到节点 j 的相对位置，那么节点 i 同样可以与节点 j 进行互相通信；

迭代算法的实现

1. 节点i从内邻居j得到p_j的估计值$\hat{p}_j$；

2. 节点i从外邻居得到加权辅助变量$\omega_{ki}\zeta_k(t)$；

3. 节点i使用辅助变量估计自己的位置如下；

$$\zeta_i\,(t+1) = \frac{1}{\sqrt{\sigma_i}}\sum\nolimits_{j\in N_i}\omega_{ij}\hat{p}_j\,(t)$$

$$\hat{p}_i\,(t+1) = \hat{p}_i\,(t) - \frac{\epsilon}{\sqrt{\sigma_i}}\sum\nolimits_{j\in N_k}\omega_{ki}\zeta_k\,(t)$$

4. 节点i发送自己的位置估计$\hat{p}_i\,(t+1)$给自己的外邻居；

5. 节点i发送自己的加权辅助变量$\omega_{ki}\zeta_i\,(t+1)$给自己的内邻居；

为了使上述迭代算法收敛，ϵ的取值范围需要满足如下；

$$0 < \epsilon < \frac{1}{\lambda_{max}(\Phi)}$$

使用下面的分布式的方式，可以在有限步骤中寻找参数 epsilon 的可用数值；

$$\lambda_{max}(\Phi) \le \left\|\widehat{H}\right\|_1\left\|\widehat{H}\right\|_\infty$$

那么我们可以采用$\frac{1}{\left\|\widehat{H}\right\|_1\left\|\widehat{H}\right\|_\infty}$作为参数$\epsilon$的上界；每个节点 i 都可以得到矩阵$\widehat{H}$的第 i 行的所有数值（$\widehat{H}$的第 i 行数值是节点 i 的所有内邻居的权重$\omega_{ij}$和方差$\sigma_i$计算）；具体公式为$r_i = \sum_{j=1}^n\left|\frac{1}{\sqrt{\sigma_i}}\omega_{ij}\right|$；然后$\left\|\widehat{H}\right\|_\infty = \max_i r_i$可以通过最大一致性算法[33]计算；每个节点 i 都可以得到矩阵$\widehat{H}$的第 i 列的所有数值（$\widehat{H}$的第 j 列数值是节点 i 的所有外邻居权重ω_{ji}和方差σ_j）；具体公式为$l_i = \sum_{j=1}^n\left|\frac{1}{\sqrt{\sigma_j}}\omega_{ji}\right|$；然后$\left\|\widehat{H}\right\|_1 = \max_i l_i$可以使用最大一致性算法计算；那么$\epsilon$可以在区间$\left(0,\frac{1}{\left\|\widehat{H}\right\|_1\left\|\widehat{H}\right\|_\infty}\right)$上取任意数值，并且都能够使得算法收敛；

4. 三维空间中的相对位置定位算法

4.1 变量定义

论文[1]中的算法可以计算二维平面上的相对位置定位问题；我们将该算法推广到三维空间中；在该节中我们首先定义一些需要使用的变量；

在问题描述中，我们已经定义三维空间中自由节点 i 测量得到内邻居$j \in N_i$在局部坐标系Σ_i中的相对位置为$q_{ij} = \left[x_{ij}, y_{ij}, z_{ij}\right]^T, j \in N_i$；通常传感器上都会带有 MEMS 的 IMU，这类 IMU 的精度虽然不高，但足够检测重力的方向；我们可以使用 IMU 传感器得到的重力方向，将局部坐标系Σ_i矫正为 z 轴与重力方向相反的局部坐标系Σ_{iz}（设使用旋转矩阵R_i^{cor}）；那么自由节点 i 的内邻居$j \in N_i$在矫正后的局部坐标系Σ_{iz}上的相对位置如下；

$$q'_{ij} = R_i^{cor}\left[x_{ij}, y_{ij}, z_{ij}\right]^T = \left[x'_{ij}, y'_{ij}, z'_{ij}\right]^T$$

对于每个自由节点$i \in [m+1, m+n]$，我们都进行同样的操作，使得每个自由节点 i 的局部坐标系Σ_i都矫正为 z 轴与重力相反的坐标系Σ_{iz}；此时，我们可以将节点 i 的内邻居$j \in N_i$的相对位置测量q'_{ij}投影到地平面上，得到$q_{ij}^{proj} = \left[x'_{ij}, y'_{ij}\right]^T, j \in N_i$；已知所有节点在地平面上的相对位置测量，那么任意自由节点 i 的坐标的 x 轴分量和 y 轴分量就可以使用论文[1]中的方式计算；

在三维空间中，传感器节点 i 的坐标$p_i = [x_i, y_i, z_i]^T \in R^3$；在二维平面上，我们重新定义变量用于表示节点 i 的坐标；我们将传感器节点 i 使用复数$p_i^{proj} = x_i + y_i i$表示；复数的实部表示节点 i 坐标的 x 轴分量，复数的虚部表示节点 i 坐标的 y 轴分量；

原本全局坐标系Σ_g是定义在三维空间中；经过投影后，得到平面上的全局坐标系Σ_g^{proj}；该坐标系与Σ_g相比只是去除了 z 轴，剩余的 x 轴和 y 轴方向与Σ_g坐标系相同；自由节点 i 在三维空间中的局部坐标系Σ_i，在之前步骤轴已经矫正为 z 轴与重力方向相反的坐标系Σ_{iz}；经过投影后，自由节点 i 在平面上的局部坐标系是Σ_i^{proj}；该坐标系与Σ_{iz}相比只是删除了 z 轴，剩余的 x 轴和 y 轴方向和Σ_{iz}相

同；然后自由节点 i 在平面上的局部坐标系Σ_i^{proj}与平面上全局坐标系Σ_g^{proj}之间存在未知的角度θ_i；

下面我们定义所有传感器节点在地平面上的坐标变量；第一组传感器集合称是锚节点（记为$1,\dots,m$）；在三维空间的全局坐标系Σ_g，这些节点的坐标是$p_a = [p_1,\dots,p_m] \in R^{3\times m}$；那么经过投影后，在二维平面的全局坐标系$\Sigma_g^{proj}$中，锚节点集合的坐标是$p_a^{proj} = \left[p_1^{proj},\dots,p_m^{proj}\right]^T \in C^m$；另一组传感器集合是自由节点（记为$m+1,\dots,m+n$）；在三维空间的全局坐标系$\Sigma_g$中，这些节点的坐标是$p_s = [p_{m+1},\dots,p_{m+n}]^T \in R^{3\times n}$；那么经过投影后，在二维平面的全局坐标系$\Sigma_g^{proj}$中，自由节点集合的坐标是$p_s = \left[p_{m+1}^{proj},\dots,p_{m+n}^{proj}\right]^T \in C^n$；然后我们设$p^{proj} = \left[{p_a^{proj}}^T, {p_s^{proj}}^T\right]^T \in C^{m+n}$为所有传感器节点在二维平面上的累计坐标；

在三维空间中，自由节点 i 测量得到内邻居$j \in N_i$在局部坐标系Σ_i中的相对位置是$q_{ij} = \left[x_{ij}, y_{ij}, z_{ij}\right]^T, j \in N_i$；在上述投影过程中，我们可以将节点 i 的内邻居$j \in N_i$的相对位置测量$q'_{ij}$投影到地平面上，得到$q_{ij}^{proj} = \left[x'_{ij}, y'_{ij}\right]^T, j \in N_i$；该相对位置测量可以表示为如下；

$$r_{ij}^{proj} = x'_{ij} + y'_{ij} i = \rho_{ij}^{proj} e^{i\theta_{ij}^{proj}}$$

$$\rho_{ij}^{proj} = \sqrt{{x'_{ij}}^2 + {y'_{ij}}^2}$$

$$\theta_{ij}^{proj} = \operatorname{arctan2}(y'_{ij}, x'_{ij})$$

我们构建了一个有向图$G^{proj} = [V^{proj}, E^{proj}], V^{proj} = \{1,\dots,m+n\}$来描述这些传感器的拓扑关系；其中，边$(j,i) \in E^{proj}$表示节点 i 可以测量得到相对位置$r_{ij}^{proj}$；那么传感器网络可以使用一个元组定义$(G^{proj}, p_a^{proj}, p_s^{proj})$；

4.2 投影计算

在三维空间中，自由节点 i 测量得到内邻居$j \in N_i$在局部坐标系Σ_i中的相对位置是$q_{ij} = \left[x_{ij}, y_{ij}, z_{ij}\right]^T, j \in N_i$；我们需要使用 IMU 传感器得到的重力方向，将局部坐标系Σ_i矫正为 z 轴与重力方向相反的局部坐标系Σ_{iz}（设使用旋转矩阵R_{cor}^i）；那么自由节点 i 的内邻居$j \in N_i$在矫正后的局部坐标系Σ_{iz}上的相对位置是$q'_{ij} = R_i^{cor}\left[x_{ij}, y_{ij}, z_{ij}\right]^T = \left[x'_{ij}, y'_{ij}, z'_{ij}\right]^T$；然后将相对位置测量$q'_{ij}$投影到地平面上，得到$q_{ij}^{proj} = \left[x'_{ij}, y'_{ij}\right]^T, j \in N_i$；

下面我们给出具体的投影计算方法；设自由节点 i 的内邻居是节点 j，并且节点 j 在局部坐标系Σ_j中的坐标是$q_{ij} = \left[x_{ij}, y_{ij}, z_{ij}\right]^T$；在局部坐标系$\Sigma_i$中，使用 IMU 测量得到重力所在的方向是$g_{ori} = \left[x_g, y_g, z_g\right]^T$；此处我们直接使用 IMU 测量得到重力方向作为$g_{ori}$；在实际应用中，我们还可以构建机器人的运动模型和重力加速度的数值大小，使用 Kalman 滤波器等方式估计出更加精确的重力方向；

我们设全局坐标系的 z 轴与重力方向相反；所以为了方便，使用变量g_{inv}表示重力相反方向，记为$g_{inv} = -g_{ori} = \left[-x_g, -y_g, -z_g\right]^T$；然后设传感器节点上相对位置检测的传感器的 z 轴与 IMU 的 z 轴相同，设为$z = [0,0,1]^T$；

我们需要将局部坐标系Σ_i进行旋转，使得旋转后的局部坐标系Σ_{iz}的 z 轴与重力方向相反；具体来说，需要将局部坐标系Σ_i沿着"IMU 的 z 轴-节点 i-重力反方向g_{inv}"三个点构建的平面的法向量（记为向量$n \in R^3$），旋转"IMU 的 z 轴-节点 i-重力反方向g_{inv}"的角度（记为角度$\alpha \in (-\pi, \pi]$）；在局部坐标系Σ_i中，IMU 的 z 轴和重力反方向分别如下；

$$z = [0,0,1]^T$$

$$g_{inv} = \left[-x_g, -y_g, -z_g\right]^T$$

那么法向量 n 可以使用向量积计算如下；

$$n = z \times g_{inv}$$

$$= \left[y_g, -x_g, 0\right]^T$$

旋转角度α可以使用数量积计算如下；

$$\cos(\alpha) = \frac{n * g_{inv}}{||n||||g_{inv}||}$$

$$= -\frac{z_g}{\sqrt{x_g^2 + y_g^2 + z_g^2}}$$

$$\alpha = \arccos\left(-\frac{z_g}{\sqrt{x_g^2 + y_g^2 + z_g^2}}\right)$$

然后我们可以计算旋转向量$v \in R^3$；

$$v = \frac{n}{||n||}\alpha$$

$$= \frac{\alpha}{\sqrt{y_g^2 + z_g^2}}\left[y_g, -x_g, 0\right]^T$$

计算得到旋转向量 v 之后，可以使用如下公式转换为旋转矩阵 R；

$$R = phi(v)$$

使用上述旋转矩阵 R，可以将局部坐标系Σ_i旋转为局部坐标系Σ_{iz}；在节点 i 的局部坐标系Σ_i，观测到内邻居节点 j 的相对位置是$q_{ij} = \left[x_{ij}, y_{ij}, z_{ij}\right]^T$；那么在局部坐标系$\Sigma_{iz}$上，内邻居节点 j 的相对位置如下；

$$q'_{ij} = R^{-1} q_{ij}$$

$$= \left[x'_{ij}, y'_{ij}, z'_{ij}\right]^T$$

因为局部坐标系Σ_{iz}的 z 轴方向与重力方向相反，所以将q'_{ij}投影到地平面上的坐标如下（在二维平面上的坐标）；

$$q_{ij}^{proj} = [x'_{ij}, y'_{ij}]^T$$

4.3 节点的 xy 分量计算

三维空间中的传感器网络(G, p_a, p_s)投影到地平面上，得到传感器网络$(G^{proj}, p_a^{proj}, p_s^{proj})$；然后使用有向图来描述这些传感器的拓扑关系$G^{proj} = [V^{proj}, E^{proj}], V^{proj} = \{1, \dots, m+n\}$；其中，边$(j, i) \in E^{proj}$表示节点 i 可以测量得到相对位置$r_{ij}^{proj}$；对于平面上的传感器网络$(G^{proj}, p_s^{proj}, p_s^{proj})$的相对位置定位问题，可以使用文献[1]中算法计算；

每个自由节点可以使用如下式子计算权重；

$$\sum_{j \in N_i} \omega_{ij}^{proj} r_{ij}^{proj} = 0$$

其中，$r_{ij}^{proj}, j \in N_i$是在节点 i 的内邻居 j 在投影后的相对位置；上述公式可以整理成如下形式；

$$H^{proj} p_s^{proj} + B^{proj} p_a^{proj} = 0$$

当使用存在噪声的相对位置测量和存在舍入误差的情况下计算权重ω_{ij}^{proj}，那么上述线性方程的形式如下；

$$(H^{proj} - \Delta_H^{proj}) p_s^{proj} + (B^{proj} - \Delta_B^{proj}) p_a^{proj} = 0$$

矩阵Δ_H^{proj}和Δ_B^{proj}是因为测量噪声和舍入误差造成的误差矩阵；我们定义$v^{proj} = \Delta_H^{proj} p_s^{proj} + \Delta_B^{proj} p_a^{proj}$，那么上述方程变换为如下；

$$H^{proj} p_s^{proj} = -B^{proj} p_a^{proj} + v^{proj}$$

为了简化分析，我们假设$v^{proj} \sim CN(0, \Sigma^{proj})$；$v^{proj}$服从均值为零并且方差为$\Sigma^{proj}$的复数高斯分布；第 i 个节点在$\Sigma^{proj}$中的分量$\sigma_i^{proj}$；使用最小二乘估计理论，可以得到上述加权最小二乘估计的最优解如下；

$$p_s^{proj} = {\Phi^{proj}}^{-1} \theta^{proj}$$

$$\Phi^{proj} = {H^{proj}}^T {\Sigma^{proj}}^{-1} H^{proj}$$

$$\theta^{proj} = -{H^{proj}}^T {\Sigma^{proj}}^{-1} B^{proj} p_a^{proj}$$

分布式算法中，矩阵$\widehat{H}^{proj}$和$\widehat{B}^{proj}$定义如下；

$$\widehat{H}^{proj} = {\Sigma^{proj}}^{\frac{1}{2}} H^{proj}$$

$$\widehat{B}^{proj} = {\Sigma^{proj}}^{\frac{1}{2}} B^{proj}$$

引入辅助变量$\zeta^{proj} \in C^n$，迭代算法公式如下；

$$\zeta^{proj}(t+1) = \widehat{H}^{proj} \hat{p}_s^{proj}(t) + \widehat{B}^{proj} p_a^{proj}$$

$$\hat{p}_s^{proj}(t+1) = \hat{p}_s^{proj}(t) - \epsilon^{proj} {\widehat{H}^{proj}}^T \zeta^{proj}(t)$$

为了使上述迭代算法收敛，ϵ的取值范围需要满足如下；

$$0 < \epsilon^{proj} < \frac{1}{\lambda_{max}(\Phi^{proj})}$$

我们使用下面的公式，寻找参数ϵ^{proj}的可用数值；

$$\lambda_{max}(\Phi^{proj}) \le \left\| \widehat{H}^{proj} \right\|_1 \left\| \widehat{H}^{proj} \right\|_\infty$$

4.4 节点的 z 分量计算

我们已经计算得到三维空间中传感器网络(G, p_a, p_s)每个节点的 xy 分量，剩余 z 分量没有计算；此时我们可以在地平面上任意找一个方向作为 y 轴（比如采用两个锚节点所在的方向），然后再次将所有传感器节点投影到 y-z 所在平面；因为我们已经得到每个传感器节点的 y 轴坐标，需要计算 z 轴坐标；同样可以使用二维平面上的相对位置定位算法计算；该方案中我们需要将地平面上选定的 y 轴，使用最大一致算法等方式广播到每个传感器节点，使得算法的计算量较大；对此，我们提出一种更加简便的方式计算每个节点的 z 轴分量；

在投影计算部分，我们可以使用 IMU 传感器得到的重力方向，将局部坐标系Σ_i矫正为 z 轴与重力方向相反的局部坐标系Σ_{iz}（设使用旋转矩阵R_i^{cor}）；在节点 i 的局部坐标系Σ_i，观测到内邻居节点 j 的相对位置是$q_{ij} = [x_{ij}, y_{ij}, z_{ij}]^T$，观测到内邻居 k 的相对位置是$q_{ik} = [x_{ij}, y_{ik}, z_{ik}]^T$；那么在局部坐标系$\Sigma_{iz}$上，内邻居节点 j 和节点 k 的相对位置如下；

$$q'_{ij} = R_i^{cor} q_{ij} = [x'_{ij}, y'_{ij}, z'_{ij}]^T$$

$$q'_{ik} = R_i^{cor} q_{ik} = [x'_{ik}, y'_{ik}, z'_{ik}]^T$$

节点 i 在三维空间中坐标是$p_i = [x_i, y_i, z_i]^T$，节点 j 的坐标是$p_j = [x_j, y_j, z_j]^T$，节点 k 的坐标是$p_k = [x_k, y_k, z_k]^T$；因为局部坐标系Σ_{iz}的 z 轴与重力方向相反，所以节点坐标z_i, z_j, z_k的 z 轴分量与观测值z'_{ij}, z'_{ik}满足如下关系；

$$z_j - z_i = z'_{ij}$$

$$z_k - z_i = z'_{ik}$$

每个自由节点的内邻居数量大于等于 2 个；如果节点 i 的内邻居数量为 2 个，那么可以构建如下线性方程；

$$\frac{1}{z'_{ij}}(z_j - z_i) - \frac{1}{z'_{ik}}(z_k - z_i) = 0$$

$$(z_j - z_i)\omega_{ij}^z + (z_k - z_i)\omega_{ik}^z = 0$$

其中参数$\omega_{ij}^z = \frac{1}{z'_{ij}}$，参数$\omega_{ik}^z = -\frac{1}{z'_{ik}}$；

如果节点 i 的内邻居$j \in N_i$大于 2 个，那么可以采用如下方程计算参数ω_{ij}^z；其中r_{ij}^z表示节点 i 和节点 j 在垂直方向上的距离差；因为节点 i 的内邻居数量大于 2 个，所以该方程不具有唯一解；我们只需要寻找一组非零参数即可；比如我们可以设$\omega_{i,j(1)}^z, \dots, \omega_{ij(|N_i|-1)}^z$为 1，然后使用该公式计算$\omega_{i,j(|N_i|)}^z$即可；

$$\sum_{j \in N_i} \omega_{ij}^z r_{ij}^z = 0$$

因为锚节点没有内邻居，那么锚节点没有测量其他节点的相对位置；然后我们可以将所有节点的的权重关系写成如下矩阵形式；

$$L^z p^z = 0$$

其中

$p^z = \left[{p_a^z}^T, {p_s^z}^T\right]^T, p_a^z = [z_1, \ldots, z_n]^T, p_s = [z_{n+1}, \ldots, z_{n+m}]^T$，$p^z$是所有节点的 z 轴坐标向量；矩阵$L^z$是 Lapalcian 矩阵，并且矩阵$L^z$的形式如下；

$$L^z = \begin{bmatrix} 0 & 0 \\ B^z & H^z \end{bmatrix}$$

上述方程展开如下；

$$H^z p_s^z + B^z p_a^z = 0$$

注释 2，如果一个传感器网络是可自定位的，那么根据定理 1 和引理 1，矩阵 H 是非奇异的；

上述方程的形式与二维平面中相对位置定位算法非常相似；在二维平面的定位算法中，向量p_a中每个分量是复数形式，用于表示节点在平面上的位置；在 z 轴定位的方程中，向量p_a^z的每个分量是单个数值，用于表示节点在 z 轴上的位置；同样的，我们使用一种分布式算法来计算每个自由节点在全局坐标系Σ_g中的 z 轴坐标；当使用存在噪声的相对位置测量和存在舍入误差的情况下计算权重ω_{ij}，那么上述线性方程的形式如下；

$$(H^z - \Delta_H^z)p_s^z + (B^z - \Delta_B^z)p_a^z = 0$$

其中矩阵H^z和B^z是通过公式(8)计算得到；矩阵Δ_H^z和Δ_B^z是因为测量噪声和舍入误差造成的误差矩阵；我们定义$v^z = \Delta_H^z p_s^z + \Delta^z p_a^z$，那么上述方程变换为如下；

$$H^z p_s^z = -B^z p_a^z + v^z$$

为了简化分析，我们假设$v^z \sim N(0, \Sigma^z)$；v^z服从均值为零并且方差为Σ^z的高斯分布；因为不同节点的测量和计算都是独立的过程，方差Σ^z是对角矩阵并且每个节点知道对应的第 i 个节点在Σ^z中的分量σ_i^z；使用最小二乘估计理论，可以得到上述加权最小二乘估计的最优解如下；

$$p_s^z = {\Phi^z}^{-1}\theta^z$$

$$\Phi^z = {H^z}^T {\Sigma^z}^{-1} H^z$$

$$\theta^z = -{H^z}^T {\Sigma^z}^{-1} B^z p_a^z$$

然后我们给出一种分布式定位方案；该算法可以从任意的起始位置开始，逐渐收敛到最优解；首先定义如下矩阵；

$$\widehat{H}^z = {\Sigma^z}^{\frac{1}{2}} H^z$$

$$\widehat{B}^z = {\Sigma^z}^{\frac{1}{2}} B^z$$

通过引入辅助变量$\zeta^z \in R^n$，我们提出如下的迭代算法；

$$\zeta^z(t+1) = \widehat{H}^z \hat{p}_s^z(t) + \widehat{B}^z p_a^z$$

$$\hat{p}_s^z(t+1) = \hat{p}_s^z(t) - \epsilon^z {\widehat{H}^z}^T \zeta^z(t)$$

上述迭代算法可以在每个节点中采用分布式的方式实现；该算法中要求每个节点与邻居节点之间存在互相通信；
在实际场景中，通信范围通常大于传感器的测量范围；那么通常情况下，如果节点 i 可以测量到节点 j 的相对位置，那么节点 i 同样可以与节点 j 进行互相通信；

迭代算法的实现

1. 节点i从内邻居j得到p_j^z的估计值$\hat{p}_j^z$；
2. 节点i从外邻居得到加权辅助变量$\omega_{ki}^z \zeta_k^z(t)$；
3. 节点i使用辅助变量估计自己的位置如下；

$$\zeta_i^z(t+1) = \frac{1}{\sqrt{\sigma_i^z}} \sum_{j \in N_i} \omega_{ij}^z \hat{p}_j^z(t)$$

$$\hat{p}_i^z(t+1) = \hat{p}_i^z(t) - \frac{\epsilon^z}{\sqrt{\sigma_i^z}} \sum_{j \in N_k} \omega_{ki}^z \zeta_k^z(t)$$

4. 节点i发送自己的位置估计$\hat{p}_i^z(t+1)$给自己的外邻居；
5. 节点i发送自己的加权辅助变量$\omega_{ki}^z \zeta_i^z(t+1)$给自己的内邻居；

为了使上述迭代算法收敛，ϵ^z的取值范围需要满足如下（证明见附录 1）；

$$0 < \epsilon^z < \frac{1}{\lambda_{max}(\Phi^z)}$$

使用下面的分布式的方式，可以在有限步骤中寻找参数ϵ^z的可用数值（证明见附录 2）；

$$\lambda_{max}(\Phi^z) \le \left\|\widehat{H}^z\right\|_1 \left\|\widehat{H}^z\right\|_\infty$$

那么我们可以采用$\frac{1}{\left\|\widehat{H}^z\right\|_1 \left\|\widehat{H}^z\right\|_\infty}$作为参数$\epsilon^z$的上界；每个节点 i 都可以得到矩阵$\widehat{H}^z$的第 i 行的所有数值（$\widehat{H}^z$的第 i 行数值是节点 i 的所有内邻居的权重$\omega_{ij}^z$和方差$\sigma_i^z$计算）；具体公式为$r_i^z = \sum_{j=1}^n \left|\frac{1}{\sqrt{\sigma_i^z}}\omega_{ij}^z\right|$；然后$\left\|\widehat{H}^z\right\|_\infty = \max_i r_i^z$可以通过最大一致性算法[33]计算；每个节点 i 都可以得到矩阵$\widehat{H}^z$的第 i 列的所有数值（$\widehat{H}^z$的第 j 列数值是节点 i 的所有外邻居权重ω_{ji}^z和方差σ_j^z）；具体公式为$l_i^z = \sum_{j=1}^n \left|\frac{1}{\sqrt{\sigma_j^z}}\omega_{ji}^z\right|$；然后$\left\|\widehat{H}^z\right\|_1 = \max_i l_i^z$可以使用最大一致性算法计算；那么$\epsilon^z$可以在区间$(0, \frac{1}{\left\|\widehat{H}^z\right\|_1 \left\|\widehat{H}^z\right\|_\infty})$上取任意数值，并且都能够使得算法收敛；

附录 1，参数ϵ范围证明

该算法的迭代公式如下；

$$\zeta^z(t+1) = \widehat{H}^z \widehat{p}_s^z(t) + \widehat{B}^z p_a^z$$

$$\widehat{p}_s^z(t+1) = \widehat{p}_s^z(t) - \epsilon^z {\widehat{H}^z}^T \zeta^z(t)$$

其中，矩阵$\widehat{H}^z = {\Sigma^z}^{\frac{1}{2}} H^z$，矩阵$\widehat{B}^z = {\Sigma^z}^{\frac{1}{2}} B^z$；并且定义$\Phi^z = {H^z}^T {\Sigma^z}^{-1} H^z$；我们需要证明，为了使算法收敛，$\epsilon^z$的取值范围需要满足如下；

$$0 < \epsilon^z < \frac{1}{\lambda_{max}(\Phi^z)}$$

将上述迭代算法的公式写成矩阵形式如下；

$$\begin{bmatrix} \zeta^z(t+1) \\ \widehat{p}_s^z(t+1) \end{bmatrix} = \begin{bmatrix} 0 & \widehat{H^z} \\ -\epsilon^z \widehat{H}^z & I \end{bmatrix} \begin{bmatrix} \zeta^z(t) \\ \widehat{p}_s^z(t) \end{bmatrix} + \begin{bmatrix} \widehat{B}^z \\ 0 \end{bmatrix} p_a^z$$

设λ^z是矩阵$A^z = \begin{bmatrix} 0 & \widehat{H^z} \\ -\epsilon^z \widehat{H}^z & I \end{bmatrix}$的特征值，并且设$\begin{bmatrix} x \\ y \end{bmatrix}$是对应的特征向量；那么我们得到如下方程；

$$\widehat{H}^z y = \lambda^z x$$

$$-\epsilon^z {\widehat{H}^z}^T x + y = \lambda^z y$$

上述式子中消去变量 x 得到如下方程；

$$\epsilon^z \Phi^z y = (-{\lambda^z}^2 + \lambda^z) y$$

上述式子中，向量 y 刚好符合Φ^z的特征向量的形式，对应的特征值是$(-{\lambda^z}^2 + \lambda^z)/\epsilon^z$；设矩阵$\Phi^z$的特征值是$\gamma^z$，因为$\Phi^z$是对称正定矩阵，所以$\gamma^z > 0$；

$$\gamma^z = \frac{-{\lambda^z}^2 + \lambda^z}{\epsilon^z}$$

该式子变换后得到如下；

$${\lambda^z}^2 + \lambda^z + \epsilon^z \gamma^z = 0$$

该式子是二次方程形式，可以使用ϵ^z和γ^z表示变量λ^z；

$$\lambda^z = \frac{1 \pm \sqrt{1 - 4\epsilon^z \gamma^z}}{2}$$

如果要使得迭代方程收敛，需要满足谱半径$\rho(A^z) < 1$，也就是任意的λ^z都需要满足λ^z的模长小于 1；因为参数ϵ^z是梯度下降的步长，所以ϵ^z大于 0；γ^z是对称正定矩阵的特征值，也是大于 0 的数值；那么$4\epsilon^z \gamma^z$部分是大于等于 0 的数值；对于$1 - 4\epsilon^z \gamma^z$部分我们可以分两种情况讨论，该部分大于等于零和小于零；如果$1 - 4\epsilon^z \gamma^z$大于等于 0，那么该部分必然小于等于 1；参数λ^z必然小于 1，那么谱半径$\rho(A^z) < 1$，因此这种情况下，参数ϵ^z没有特定的数值范围；然后如果$1 - 4\epsilon^z \gamma^z$部分小于 0，那么λ^z的模长如下；

$$\left|\left|\lambda^z\right|\right|^2 = \frac{1 + i\sqrt{4\epsilon^z \gamma^z - 1}}{2} \times \frac{1 - i\sqrt{4\epsilon^z \gamma^z - 1}}{2}$$

$$= \frac{1 + 4\epsilon^z \gamma^z - 1}{4}$$

$$= \epsilon^z \gamma^z$$

要使得迭代算法收敛，需要满足谱半径$\rho(A^z) < 1$，那么任意λ_z的模长需要满足如下；

$$\left|\left|\lambda^z\right|\right|^2 = \epsilon^z \gamma^z < 1$$

$$\epsilon^z < \frac{1}{\gamma^z}$$

因为ϵ^z需要满足任意的特征值γ^z的情况，所以$\epsilon^z < \frac{1}{\gamma_{max}^z}$，或者$\epsilon^z < \frac{1}{\rho(\Phi^z)}$；同时因为$\epsilon^z$是梯度下降算法的步长，需要满足$\epsilon^z > 0$；所以参数$\epsilon^z$的取值范围如下；

$$0 < \epsilon^z < \frac{1}{\rho(\Phi^z)}$$

REFERENCES

Brown, F., Harris, M.G., and Other, A.N. (1998). Name of paper. In Name(s) of editor(s) (ed.), *Name of book in italics*, page numbers. Publisher, Place of publication.

Smith, S.E. (2004). *Name of book in italics*, page or chapter numbers if relevant. Publisher, Place of publication.

Smith, S.E. and Jones, L.Q. (2008). Name of paper. *Name of journal in italics*, volume (number), page numbers.

【待补充，调整格式后加入】