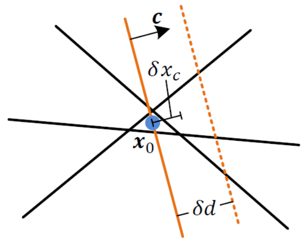


Fig. 2. Definition of *degeneracy factor*. We insert an additional constraint (orange line) as a disturbance and inspect movement of the solution $\boldsymbol{x}_0$.

SLAM中退化场景的判别

FrankDellaert
酷的像风

定位退化是什么?

对于SLAM而言，传感器需要纹理信息决定特征点，在特征点缺乏时，导致约束减少，比如在狭长的走廊上或者隧道环境，状态估计方法会进行退化，很难进行定位，这时候在工程上来说，要求机器人的系统，需能进行判断出来并做对应的预处理，如果实在处理不了，再将退化后的结果返回给机器人并通知使用者；

1 导入问题: LIO_SAM中退化的判断

关于LIO_SAM其中这块的代码，没有看的很明白，起始下面的代码部分并不是LIO_SAM作者Tixiao Shan写的，而是他复用了LOAM作者Zhang Ji的思想

```
cv::Mat matA(laserCloudSelNum, 6, CV_32F, cv::Scalar::all(0));
cv::transpose(matA, matAt);
matAtA = matAt * matA;
float eignThre[6] = {100, 100, 100, 100, 100, 100};

cv::eigen(matAtA, matE, matV); //matE is eigenvalue : matV is eigenvector
for (int i = 5; i >= 0; i--)
{
  if (matE.at<float>(0, i) < eignThre[i]) //只要有一个特征值小于阈值，就算激光定位退
  {
    for (int j = 0; j < 6; j++)
     {matV2.at<float>(i, j) = 0;}
    isDegenerate = true;
  }
  else {
   break;
  }
}
```

打印了一次他们的数值如下所示:

```
j=5; matE.at<float>(0, j)=1474.81 j=4; matE.at<float>(0, j)=2074.43 j=3; matE.at<
j=2; matE.at<float>(0, j)=54927.5 j=1; matE.at<float>(0, j)=99595.9 j=0; matE.at<

j=5; matX.at<float>(j, 0)==-0.00313341 j=4; matX.at<float>(j, 0)==-0.00498093 j=3
j=2; matX.at<float>(j, 0)==0.000204836 j=1; matX.at<float>(j, 0)==0.00122446 j=0;
```

2 前置知识:特征值和特征向量

A矩阵是n阶方阵，x为非零向量，若存在数 λ 使得下式成立，那么数 λ 称为矩阵A的特征值(eigenvalue),非零向量 x 称为矩阵A的对应于λ的特征向量(eigenvector),这里特征值只限定在实数范围内；

$$A \bullet x = \lambda \bullet x$$

特征值和特征向量几何意义的结论：

某个非零的向量，乘以矩阵A后，实现了线性变换后的结果，有时候方向相对于线性变换前会发生改变，还有的运算结果的向量是没有发生改变的，那些没有发生方向改变的向量称为特征向量，变换前后的**伸缩比**叫做特征值；

至于是SVD分解还是QR分解不应该，也不能成为研究SLAM退化问题的主线，因为分解方式有好多种，我们当前任务是聚焦将特征值求解出来，然后加以利用上，如何求解不是重点！这一点是当下的我和以前的我，在思考方式上很大的不一样！最近也有种起飞的感觉，哈哈哈

3 退化会和矩阵的条件数⁺有关么？

然后又根据matAtA=matAt * matA；感觉这个乘法有点儿是在判断矩阵的条件数，进而确定矩阵是不是病态矩阵，后来分析不是条件数，因为定义都不对！如果是条件数的运算那么一定是对他们各自求矩阵范数，然后相乘，而不是先求矩阵的乘法，再求解矩阵的范数；

还有一个原因和条件数没有关系的是：

```
float eignThre[6] = {100, 100, 100, 100, 100, 100};
if (matE.at<float>(0, i) < eignThre[i])
```

如果是和条件数有关系，那么上面的代码内容不至于说是小于某个固定的数值，原文中定义的是100；如果是条件数，那么代码中应该这样写：

```
float 条件数;
if (matE.at<float>(0, i) ≤ 条件数)
```

下面就补充一下条件数的知识点，了解的可以掠过

《数值分析》第2版 Timothy Sauer P77中介绍方阵A的条件数cond(A)为求解Ax=b时，对于所有右侧向量b，可能出现的**最大**误差放大因子；

计算方法：cond(A)= $||A|| \bullet ||A^{-1}||$

其中计算 $||A||$ 这个叫做**矩阵范数**的东西=矩阵每行元素绝对值之和，然后取其中的最大值；

如此定义条件数是对应现实场景中最糟糕的情况，对于系统 $A \bullet X = b$ 来说，对于任何其他的b，误差放大因子将 $\leq$ 这个条件数的值，因为条件数考虑的情况是最糟糕的情况会有多么的差，这在工程上是非常有帮助的，因为工程上不像学术研究在乎你的顶级表现是怎么样的，而是在乎系统**稳定发挥**的水准是怎么样的！

下面是另外一位同学 @独鹿 关于条件数的分享：

> **病态矩阵**：即求解方程组时对数据的微小扰动比较敏感的矩阵。比如在求解方程组 $Ax = b$ 时，如果对系数矩阵 A 或右端项 b 进行一个微小的扰动，算出的解和原方程组的解差别很大，这样的矩阵 A 被称为病态矩阵。

- 用**条件数**来判断 A 矩阵是否为病态矩阵，对于这种一个微小扰动对数值求解造成很大干扰的病态矩阵直接跳过。
- 条件数定义为**矩阵的范数乘以其逆矩阵的范数**，即 $cond(A) = |A| \cdot |A^{-1}|$，条件数越大，矩阵越病态，越接近不可逆，从而也越难得到精确解。

4 Zhang Ji关于SLAM退化的判定

后来我是如何想到这篇文章的，以前阅读过一些文章，记得zhang ji有一篇文章是讲SLAM的Degeneracy，乘坐完火车回家，就打开电脑查看这篇文章，果然有关于SLAM退化的内容，所以说并不是为了看这篇文章而写的知乎，而是因为痛点引入；

分享Zhang Ji文章《On Degeneracy of Optimization-based State Estimation Problems》

这是激光SLAM中防止非线性退化最早的一批论文。重要性不言而喻，原论文链接如下：

下文是比较早的了，有点儿**概述**的味道：

下面真的是对雅可比一个最经典的解释了！雅可比的本质就是为了**线性化**：

```
Most nonlinear optimization methods locally linearize the problem by computation
```

之前**李群和李代数**不就是为求雅可比嘛？！然后再去学李群就会更有目的性，之前高中学的微分都是不work的，因为李群和李代数是高等线性代数的积分和微分；

4.1 解的张成空间

如果面对的是非线性问题建模，那么可经过雅可比**线性化**为求最小二乘方程 arg\ \mathop{min}\limits_{x}\sum_{i=1}^{n}({\left||Ax_{i} -b\right||)^{2} } 其中n表示方程组的数量，也是方程组矩阵的行数目；上面的方程最后求出的结果应该是关于 x_{i} 的；

因特征点约束的数量至少都在10个以上，但想要求解的目标函数值，只有表达位姿的**6个**值的根，所以上面的方程数量大于根的数量，所以是**超定方程**，所以这个方程组**没有解析解**(代入6个根的值让方程组中的每1个方程都为0)，只有最小二乘的最优解(代入方程组，让这几个方程组结果的平方和最小)；

其中方程组中的每个方程 Ax_{i} -b=0的解(solution) x_{i} 都能够形成一个平面(3维度是平面，>3维就叫**超平面**)，或者叫做解集的**张成(span)空间**;

```
张成(span)空间含义：矩阵的最大无关组作为基，构成的向量空间；
```

所求得解x的几何意义是：在解空间中找到一个点，应满足到每个超平面的**欧式距离最小**；

4.2 构建虚拟约束

4.2.1为什么要添加那条桔色的面(线)?

构建虚拟约束：是为了"测试"系统相对于disturbances(扰动)产生优化结果的鲁棒性(稳定性)；

先是浅浅介绍一下，4.2.4有详细解释；

4.2.2添加的这条桔色的面(线)是唯一的么？

we insert an additional constraint passing through x_{0} as the orange line

c^{T}(x-x_{0})=0, \left|| c \right|| =1,

where c is a n * 1 vector indicating the **normal** of the constraint (the black arrow in Fig. 2).

本质上它是穿过已知点 x_{0} 桔色的面(线)，然后c向量是穿过并且是这个面的**单位**法向量；

所以穿过x_{0}的平面不是唯一的，是一簇平面的集合(他们旋转的自由度没有约束，所以可自由旋转形成**平面簇**);

4.2.3 这是作者没有思考到的么？还是刻意为之？

答：是刻意为之

只有当这个平面的旋转方向没有限制的时候，随意抽出一个曲面才具有一般性,可代表这个平面簇;而且作者定义的退化因子degeneracy factor D 非常好，分母是这个平面簇的最大偏移量，也就是说某个位置的平面旋转360°后，其中产生最大的偏移量作为分母，作者思考的是，在最"**糟糕**"的场景作为此算法的"底线"！

得到 J=\partial f(x)/\partial x (注意偏导数一定要带着偏导符号 \partial)，下文还定义了一个退化因子,注意退化因子就是除法而**不是偏导**了!The degeneracy factor D = \delta d/ \delta x_{c}^{*}；

\delta x_{c}^{*} =\mathop{{max}}\limits_{c}{(\delta x_{c}^{})} 用虚拟约束的超平面 移动的距离去 **除以** 解的**最大位移量**;

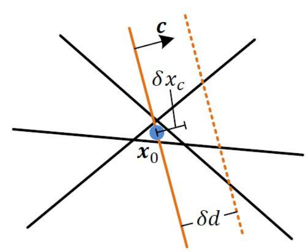
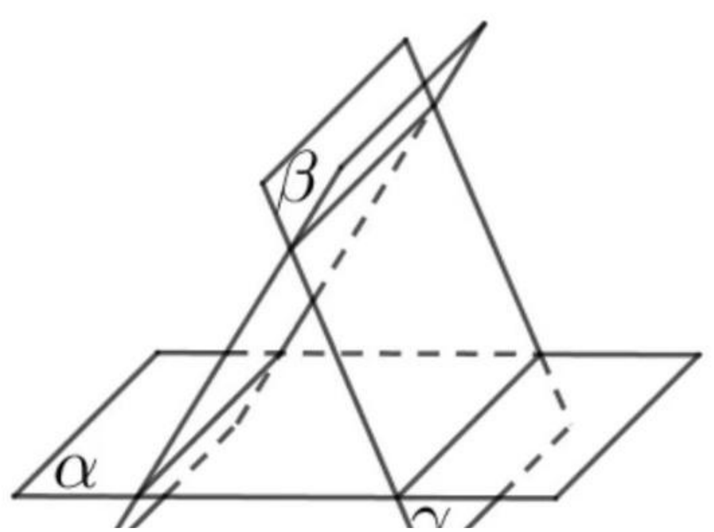


Fig. 2. Definition of *degeneracy factor*. We insert an additional constraint (orange line) as a disturbance and inspect movement of the solution $\boldsymbol{x}_0$.

Definition of degeneracy factor. We insert an additional constraint(orange line) as a disturbance and inspect movement of the solution x0

实际上3个平面的交叉，**还原**为3维立体场景应该是下面这样：只不过我们是从其中一个面看到的，像直线！



桔色线条表示该超平面的法向量，这个超平面经过解x0，所以添加这个约束并不改变原问题的解; 然后通过桔色的法向量的移动来"测量"整个优化系统的稳定性，作者起名是退化因子，也可以叫做不稳定性因子；这个参数的设计可以说是这个文章最大的**靓点**也是最不容易理解的地方；

4.2.4 为什么要加入这个**桔色面(线)**呢?

不失一般性，它只是一个**代表**，当然还有其他类别的约束会因为disturbances导致优化结果的变化，到底这个变化对优化结果影响有多大，其实这个桔色面约束就是起到了对整个结果的"测试"作用！该值越大说明桔色超平面的扰动并不会对最终的结果产生的影响偏小，系统比较稳定，不会因为其中一个约束的变化而让整个求解产生较大的波动；反之就是对系统影响偏大，说明系统不稳定！(来自于原文中下面的那段话)

```
By maximizing  ，we find a direction in which the solution is the least stable
```

算法应用**具象化了数学**，如果按照课本捋一遍线性代数之后，是不是都lost(迷失)了,所以问题导向，让你插上翅膀，也让数学变得更有**力量**而不是更难缠！

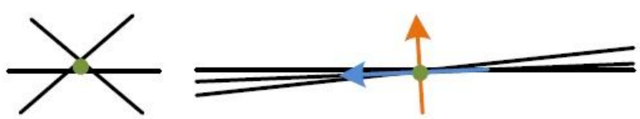
4.3 退化因子D 只与矩阵A有关，而与b无关

退化因子是**人为构建**出来的，是判断矩阵是否退化的**重要指标**！

4.3.1 先说这个结论是不是违反常识?

上面的结论说明退化方向只与原始的约束方向有关，与原始约束的位置无关。这和前面的推测一致，当原始约束方向平行的时候，就会引入退化。如下图所示，并不是喂饭常识的，是否退化和约束的方向有关，和这些约束所处的position无关，下面就进行proof！

先铺垫一下左(右)伪逆矩阵的概念：



(a) (b)

Fig. 1. Intuition of the proposed method in determining state estimation degeneracy. The black lines represent constraints in a state estimation problem. (a) illustrates a well-conditioned problem. The solution (green dot) is constrained in different directions. (b) gives an example of degeneracy. The constraints are mostly parallel such that the problem is degenerate along

图(b)就是退化场景

c^{\dagger}=c_{left}^{-1}=(c^{T}c)^{-1}c^{T}=c^{T}

将这个左伪逆矩阵分别左乘原文章公式(6)和(7)的两侧，然后(6)和(7)左侧就只剩下各自的解(solution)了，所以这里引入**左伪逆矩阵**目的就是**为约掉**左侧除未知数外多余的矩阵，让等式一侧只是剩下待求解的未知数，核心思想和初中时候的多项式原理是类似的；

左乘原文章公式(6)和(7)的两侧，再用(7)subtracting(6),运算过程中**没必要展开**，因为有的矩阵相乘多项式直接可以约掉的！

在经过左伪逆矩阵(left pseudo inverse)运算后得到：

and subtracting (6) from (7), we can compute $\delta \boldsymbol{x}$,

$$\delta \boldsymbol{x} = (\mathbf{A}^T\mathbf{A} + \boldsymbol{c}\boldsymbol{c}^T)^{-1}\boldsymbol{c}\delta d. \qquad (9)$$

Recall that $||\boldsymbol{c}|| = 1$, the shift in the direction of $\boldsymbol{c}$, δx_c can

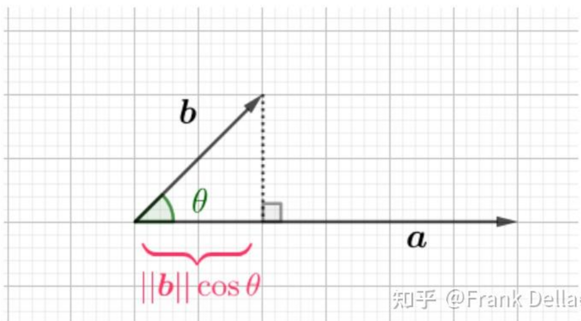
Eq. (10) tells us that δx_c is a function of $\mathbf{A}$, $\boldsymbol{c}$, and δd, but not a function of $\boldsymbol{b}$. Hence, we complete the proof. ∎

上面式子是如何从(9)->(10)的是因为下面的dot product:

根据点积可推出:

$$cos\theta = \frac{\boldsymbol{a}\cdot\boldsymbol{b}}{||\boldsymbol{a}||||\boldsymbol{b}||} \implies \boldsymbol{a}\cdot\boldsymbol{b} = ||\boldsymbol{a}||||\boldsymbol{b}||\cos\theta$$

其中$||\boldsymbol{b}||\cos\theta$可以看作$\boldsymbol{b}$在$\boldsymbol{a}$上的投影:



因为 |\left| c \right||=1 所以上面式子就变成了向量 \delta x 在平面单位法向量c上面的投影了，投影之后是标量；

根据公式推导得到： \delta x_{c}=c^{T}(A^{T}A+cc^{T})^{-1}c\delta d

上式告诉我们退化因子与A，c， \delta d 有关，但是和b是无关的；

4.4 退化因子D=λmin+1

其中 \lambda_{min} 是求得 A^{T}A 众多特征值(eigenvalue)当中的最小值；

但是zhang ji论文里使用的是右伪逆的转置：需要经过一个**硬核的变换**：

c_{right}^{-T}=(c^{\dagger})^{T}=(c^{T})^{\dagger}=((c^{T})_{right}^{-1})

然后将上面左伪逆和右伪逆矩阵一起代入到论文的公式(10)中进行化简，这一切还是比较简单，最难的是下面：

构建一个**Rayleigh商**：

7.6 Rayleigh 商 **443**

7.6.1 Rayleigh 商的定义及性质

在研究振动系统的小振荡时，为了找到合适的广义坐标，Rayleigh 于 20 世纪 30 年代提出了一种特殊形式的商[425]，被后人称为 Rayleigh 商。下面是现在被广泛采用的 Rayleigh 商定义。

定义 7.6.1 Hermitian 矩阵 $\boldsymbol{A}\in\mathbb{C}^{n\times n}$ 的 Rayleigh 商或 Rayleigh-Ritz 比 $R(\boldsymbol{x})$ 是一个标量，定义为

$$R(\boldsymbol{x}) = R(\boldsymbol{x},\boldsymbol{A}) = \frac{\boldsymbol{x}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{x}}{\boldsymbol{x}^{\mathrm{H}}\boldsymbol{x}} \tag{7.6.1}$$

其中，$\boldsymbol{x}$ 是待选择的向量，其目的是使 Rayleigh 商最大化或者最小化。

Rayleigh 商的重要性质如下[393, 394, 101, 224]。

《矩阵分析与应用（第二版）》 张贤达

然后用**Rayleigh-Ritz** 定理论述它的有界性：

$$\lambda_{\min} = \lambda_1 \leqslant \lambda_2 \leqslant \cdots \leqslant \lambda_{n-1} \leqslant \lambda_n = \lambda_{\max} \tag{7.6.6}$$

排列，则

$$\max_{x \neq 0} \frac{x^H A x}{x^H x} = \max_{x^H x = 1} \frac{x^H A x}{x^H x} = \lambda_{\max}, \quad \text{若 } Ax = \lambda_{\max} x \tag{7.6.7}$$

和

$$\min_{x \neq 0} \frac{x^H A x}{x^H x} = \min_{x^H x = 1} \frac{x^H A x}{x^H x} = \lambda_{\min}, \quad \text{若 } Ax = \lambda_{\min} x \tag{7.6.8}$$

更一般地，矩阵 A 的所有特征向量和特征值分别称为 Rayleigh 商 $R(x)$ 的临界点 (critical point) 和临界值 (critical value)。

444 第 7 章 特征分析

《矩阵分析与应用（第二版）》张贤达

当我们知道瑞利(Rayleigh)商的形式可以使得整个函数取得最大化或者最小化时候，接下来的工作就有了方向，就是构建这种Rayleigh商的的形式，求解最值；

然后按照目标构建等式： c^{*}=arg\ \mathop {min}\limits_{c}\ \frac{c^{T}A^{T}Ac}{c^{T}c}

然后如何求取最值呢？按照下面的内容：就是求 AA^{T} 的特征值

```
Since  is a symmetric matrix,
the minimum of the quotient is equal to the minimum eigenvalue of
,namely
```

张贤达老师的电子书，但还是推荐大家买纸质书，查询起来更方便！

链接: pan.baidu.com/s/1uWMIMt...

提取码: 7c69

5 Zhang Ji关于SLAM已退化的状态进行地纠正

因为内容篇幅太长了，准备再单独开辟一个文章，记得催更哦！

福利: 知乎的公式编辑器

Tex：是一种具有编译和排版功能的基础语言，相当于 C 语言。

LaTex：LaTex是 Tex 的扩展版本，拥有多种宏包，能实现比 Tex 更多的功能。

知乎中使用的是Tex类型的公式编辑器(可以粗浅理解为Latex)；但知乎不支持 Markdown；

Latex如何引入空格

下文先在CSDN垃圾堆文章中看到的，然后引流到古月居，下面还有从油管搬运到B站的视频，真是太棒了！