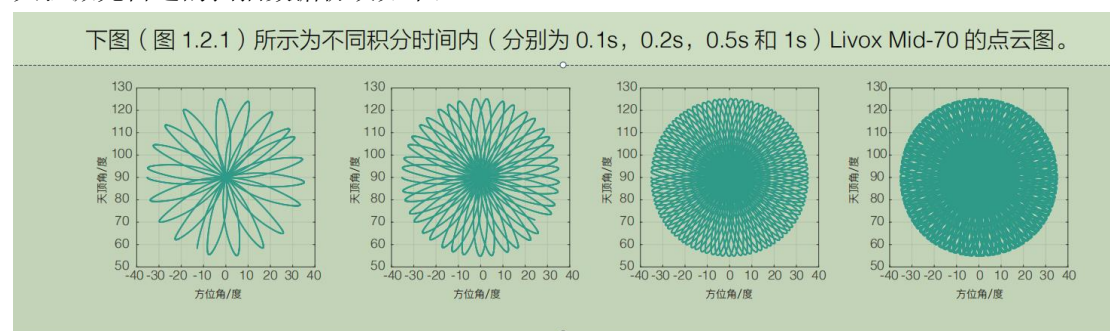


大疆 Livox 激光雷达的 json 参数配置内容如下：

```
{
  "SeqID":0,
  "TypeID":6,
  "TargetCopter":1,
  "TargetMountType":0,
  "DataWidth":250,
  "DataHeight":40,
  "DataCheckFreq":10,
  "SendProtocol":[0,127,0,0,1,9999,0,0],
  "CameraFOV":70.432,
  "SensorPosXYZ":[0,0,-0.3],
  "SensorAngEuler":[0,0,0],
  "otherParams":[600,2.956,1.4,1.18,10,5,1,0]
}
```

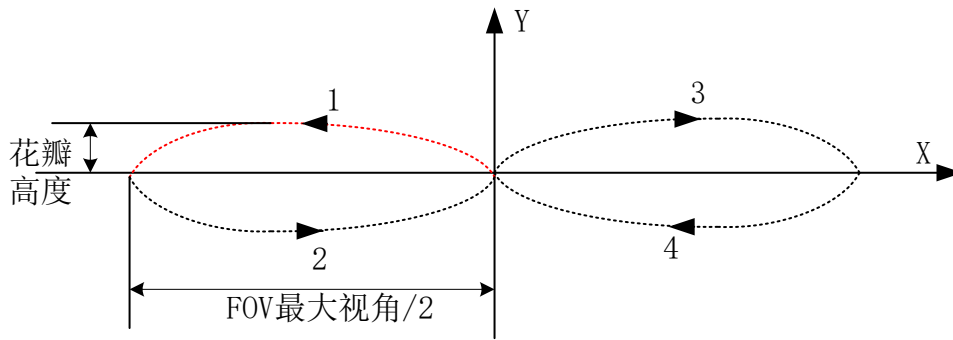
大疆激光雷达的扫描数据协议如下，



图：大疆激光雷达的实验数据展示

为了便于复现上面的扫描过程，进行如下假设：

- 1) 激光发射路径按如下图所示的 8 字型来扫描，扫描完一个 8 字型后，偏转一定角度去扫描下一个 8 字型。
- 2) 两个 8 字型之间错开的角度不能被 360 度整除，确保扫描一定周期内，不会出现重叠情况。
- 3) 每个八字形分为 2 个花瓣，4 个弧。每个弧上有若干个扫描点，扫描点之间是非均匀分布的（这里采用了幂函数分布形式），往外侧点越密，往内侧点越稀疏。但是，从上图的空间上看，这种内疏外密的方式，反而能在整个圆形视野扫描区域内，达到二维均匀分布。
- 4) 用 `DataWidth` 表示每段弧的点数，`DataHeight` 表示每一帧数据发出时，扫描的弧的数量。因此，每帧数据发出的点总数为 `DataWidth*DataHeight`
- 5) `DataCheckFreq` 表示每秒发送的数据帧数。



图：弧、花瓣、8 字图形的定义，数字和箭头表示扫描顺序

其他参数与 RGB 传感器相同。下面参数是定制内容：

TypeID: 6，分配给大疆激光雷达的传感器 ID 号

DataWidth: 半边花瓣（对应上图红色区域）的点数

DataHeight: 表示一帧需要扫描多少个 半边花瓣（上图是 4 个花瓣，形成一个 8 字形）

DataCheckFreq: 每秒有多少帧数据

因此：DataWidth * DataHeight * DataCheckFreq = 每秒雷达扫描的点数

CameraFOV: 对应了上图半边花瓣长度的 2 倍。单位度。

otherParams[0]: 激光雷达的最远扫描距离（单位米），例如这里是 600 米。注意：点云数据我们最终是以 int16 格式发布，这个值是与 otherParams[0] 参数是耦合的。例如，x_float 是激光雷达测量得到的点云的 x 轴浮点数值（单位米，数值小于 otherParams[0]），则 x_int = x_float / otherParams[0] * 32767。因此，拿到点云数据后，需要再根据最远距离 otherParams[0]，做一个逆向映射：x_float = x_int / 32767 * otherParams[0]。

otherParams[1]: 对应了上图的花瓣高度（单位度）。这里反映了扫描花瓣的弯曲程度。

otherParams[2]: 花瓣由中心向两边的点的 x 方向分布指数。指数为 1 说明均匀分布，指数为 1.5（默认值）说明靠近中心的点稀疏，靠近边上的点密集。

算法原理：

设图中红色半边花瓣有 DataWidth 个点，则第 i 个点的 X 坐标通过如下公式计算（浮

点数指数运算)

$X = -(i/\text{DataWidth})^{\text{otherParams}[2]}$

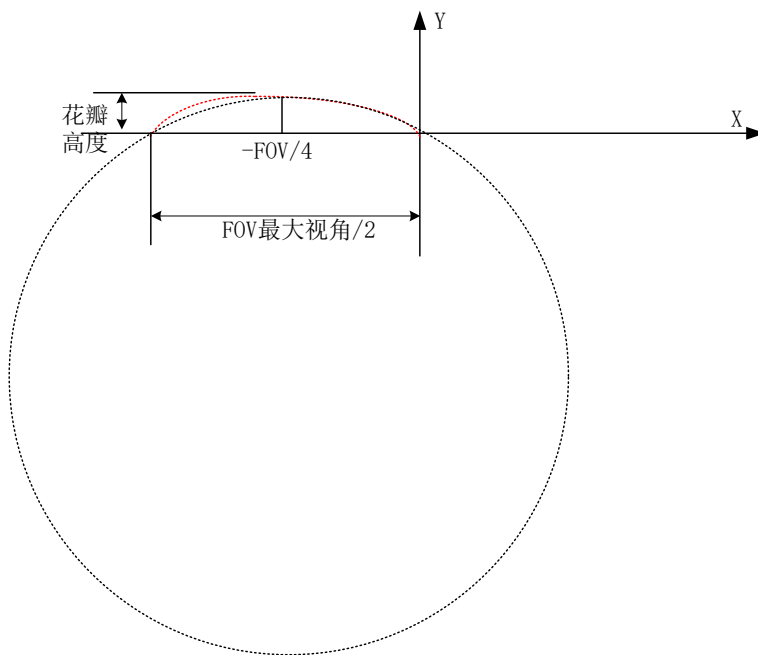
注：通常而言，中心的点要稀疏一些，越往两边越密，这样能在整个空间上使点云尽量均匀分布。

otherParams[3]： 红色花瓣左侧圆弧的校准指数，默认为 1.18。

原理：

花瓣的拟合原理如下：

首先根据 $\text{FOV}/2$ 作为弧端点长度 和花瓣高度 $\text{otherParams}[1]$ 作为弧高度，用这两个参数可以计算得到一个外接圆进行拟合。其次，将花瓣沿着中心 ($-\text{FOV}/4$ 的位置) 再分为两半，得到右侧圆弧图形。



图：对一条弧进行分段拟合

对于红色弧的左侧圆弧，我们直接用一个 $1/4\text{FOV}$ 半径的圆进行拟合，然后对 Y 轴进行压缩，同时对曲线进行校准。公式如下：

$X = -(i/\text{DataWidth})^{\text{otherParams}[2]}$

如果 $X < -0.5$ 也就是对应左侧圆弧

$X_{\text{ang}} = X * \pi$

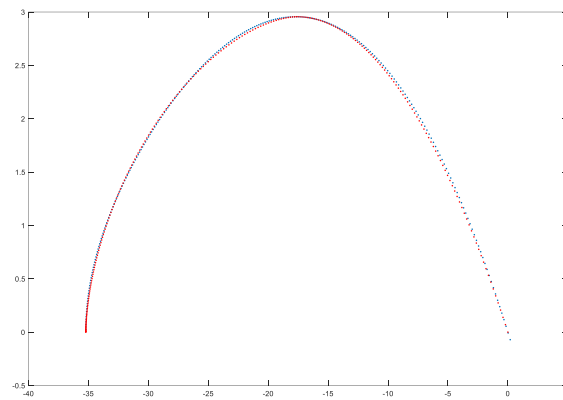
$R = 1/4 * \text{FOV}$

$X_{\text{point}} = -R * \cos(X_{\text{ang}})^{\text{otherParams}[3]} - R;$

$Y_{\text{point}} = \text{otherParams}[1] * \sin(X_{\text{ang}})^{\text{otherParams}[3]};$

从上面公式可以看出, $\text{otherParams}[3]$ 是红色花瓣左侧圆弧的校准指数, 默认为 1.18。上述拟合方式, 保证了在花瓣的左端点的斜率为竖直方向, 上端点的斜率为水平方向, 实现图形的连续性过渡。

`otherParams[2]=1.5` //沿径向的疏密分布幂指数系数
`otherParams[3]=1.18` //外弧图形的幂指数系数
以上面图形校准参数获得的与原始花瓣参数的拟合图如下

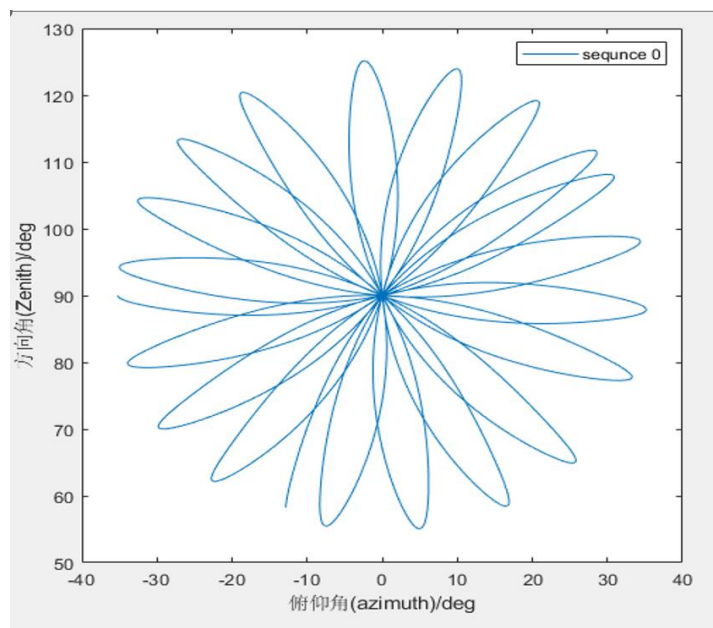


图：弧拟合函数的拟合效果

图中蓝色虚线为官网提供的角度扫描数据，红色点线为拟合函数输出曲线，可见图形与数据分布均拟合的非常好。

`otherParams[4]`：几个 8 字（一个八字是两片对称花瓣，4 段弧）组成一朵花，如下图所示是大疆激光雷达 0.1s 的数据，也就是 10 个八字（20 片花瓣）组成了一朵花。这样 `otherParams[4]=10`。也就是说，相邻两个花瓣之间的角度为 $180/10=18$ 度。

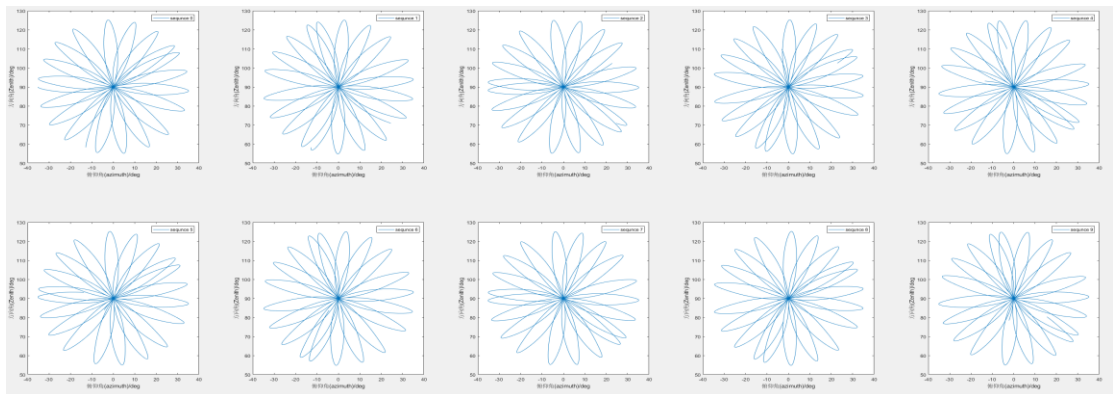
注意：为了保证扫描的均匀性，这几个花瓣并不是从左到右依次扫描的，扫描顺序是生成一个 0 到 N-1 的随机数列表，以此顺序来扫描这朵花的花瓣。



图：一朵花的定义

otherParams[5]: 几朵花（0.1s，20 个花瓣，10 个 8 字）组成一个图形。也就是说，对于上图的花的图像，需要均匀插入若干朵花，让他们进一步细分。这里默认值为 5，也就是说五朵花组成了一个图形。

注：这里的花朵是依次扫描的，如果每朵花有 20 个花瓣（10 个 8 字）则每两个花瓣之间夹角是 $360/20=18$ 度，每 5 朵花组成一个图形的话，那相邻两朵花的偏转为 $18/5=3.6$ 度。



otherParams[6]: 每扫描完一个图形，都会新引入一个随机错位角度，让两个图案不会完全重叠。随机错位角度为 $\text{rand}(0,1) * \text{otherParams}[6]$
其中， $\text{rand}(0,1)$ 是一个 0 到 1 之间的随机浮点数。

最终，采用参数

```
DataWidth=250 //一条弧（1/2 个花瓣，1/4 个 8 字图像）的点数
DataHeight=40 //每一帧图像一共多少条弧，这里 40 条弧，对应 20 个花瓣，10 个 8 字图形
DataCheckFreq=10//一秒有多少帧，也就是说这里每秒=250*40*10=10 万个点
CameraFOV= 35.216*2 //视场角，也就是花瓣的弦长，单位度
otherParams[0]= 600 //激光雷达最远距离
otherParams[1]= 2.9559 //花瓣版弧图形的高度，单位度。
otherParams[2]=1.4 //沿径向的疏密分布幂指数系数
otherParams[3]=1.18 //外弧图形的幂指数系数
otherParams[4]=10 // 一朵花包含多少个 8 字图形，这里是 10 个（每 0.1s 形成一个花朵）
otherParams[5]=5 //一个图案包含多少个花朵，这里是 5 个（每 0.5s 形成一个图案）
otherParams[6]=1//每扫描完一朵花，需要代入的噪声最大值，单位度
```