# 六月工作报告

六月的工作内容主要围绕UWB定位中的**时间同步算法**开发、**定位算法**（三维坐标解算）开发展开。同时为验证产品性能（软件与硬件的最终效果），为今后产品迭代提供依据，确立了定位精度方面的各项**性能指标**，并建立一套定位精度的**测试流程**，编写**数据处理脚本**自动生成可视化数据。以下是各项工作的进度与细节：

## 基站坐标标定算法开发（已完成，可扩展）：

基于非线性最小二乘拟合，编写了对基站坐标进行**自动标定**的算法。现场人员布设基站时，通过手动布置基站坐标耗时耗力，且引入了基站坐标误差。最终标签的定位误差会比基站误差更大。使用该算法可以根据各基站间的双向测距值实现基站坐标的计算。该算法实现了在给定测距值下的最高坐标精度。

### **可扩展内容：**

若Controller加入自动进行各基站间的双向测距功能，与该标定算法结合，意味着基站可以自动进行自身坐标计算。这不仅解放了现场人员的工作量，更意味着**基站可以方便地进行移动**，可适应井下等需要基站频繁移动场景的部分需求。

## 定位算法（尚未完成，正常推进）：

### 预计目标：

核心是使用Levenberg-Marquardt算法进行非线性最小二乘优化，实现给定TDOA值下的最优坐标计算。

其中在spherical-intersection (SX)算法（原厂）或The Spherical-Interpolation（SI）中选择一种作为粗定位结果，用作Levenberg-Marquardt迭代计算的初值。

该算法可以实现在给定TDOA测量值下的整体误差最小化。但会带来一定的计算负荷。

### **目前进度**：

完成Python环境下的数值实验环境搭建。可自动生成TDOA数据，对比各算法性能。

完成原厂算法在python下的复现和测试。

完成Python环境下Levenberg-Marquardt算法与原厂算法的性能对比。

完成SI算法在CLE下的工程化和编译、运行。（已经可以出实物定位结果）。

正在进行SI算法与原厂算法在实物上的性能对比。

正在进行Levenberg-Marquardt算法的C++工程化。

### 预估时间点：

七月十五日前完成Levenberg-Marquardt算法的C++编写，八月一日之前在实物上调试通过，并进行性能测试。八月十五日前完成该算法下的计算负荷测试、容量测试、系统稳定性测试。

## 时间同步算法（尚未完成，正常推进）：

### 预计目标：

将无线传感器网络领域内的LSTS算法迁移至UWB基站时间同步中。该算法相比原厂的卡尔曼滤波算法具有**稳定性收敛性更强**，但**动态敏感度更弱**的特点。故选择将**两个算法进行融合**，利用融合算法达到时间同步性能稳定且动态性良好的特点。

### **目前进度**：

已完成Python环境下的融合算法实现和数值实验（使用真实运行数据）。

已完成融合算法的C++编写。

尚未完成该算法在CLE工程下的编译和调试。

### 预估时间点：

八月十五日前完成该算法在CLE工程下的编译、调试、性能测试。

## 定位精度测试流程建立（接近完成）

### 预计目标：

确立产品定位精度的相关指标。建立一套标准化的测试流程，留下一份可用作自动数据处理、生成可视化图表的Python脚本。

### **目前进度**：

已确立产品定位精度指标并编写文档。

已确立静态无遮挡环境（即完美环境）下的测量流程。

已编写对TDOA数据及最终定位数据进行处理并生成可视化图表的Python脚本。

尚未完成TDOA真值标定与坐标真值标定。原因是在处理数据过程中对上次手工激光测距的结果感到了不确定，准备进行更加标准化的测量。

### 预估时间点：

七月十五日前完成