**《机器人编程实践》学生实验报告**

**学院** 数计学院 **专业** 计算机科学与技术 **班级** 计科3班

**姓名** 周吉瑞 **学号** 20190521340 **日期** 2021/05/14

|  |  |  |  |
| --- | --- | --- | --- |
| **课程名称：** | 机器人编程实践 |  |  |
| **实验名称：** | 实验六 定点射击 | | |
| **指导老师：** | 孙建勇 |  |  |

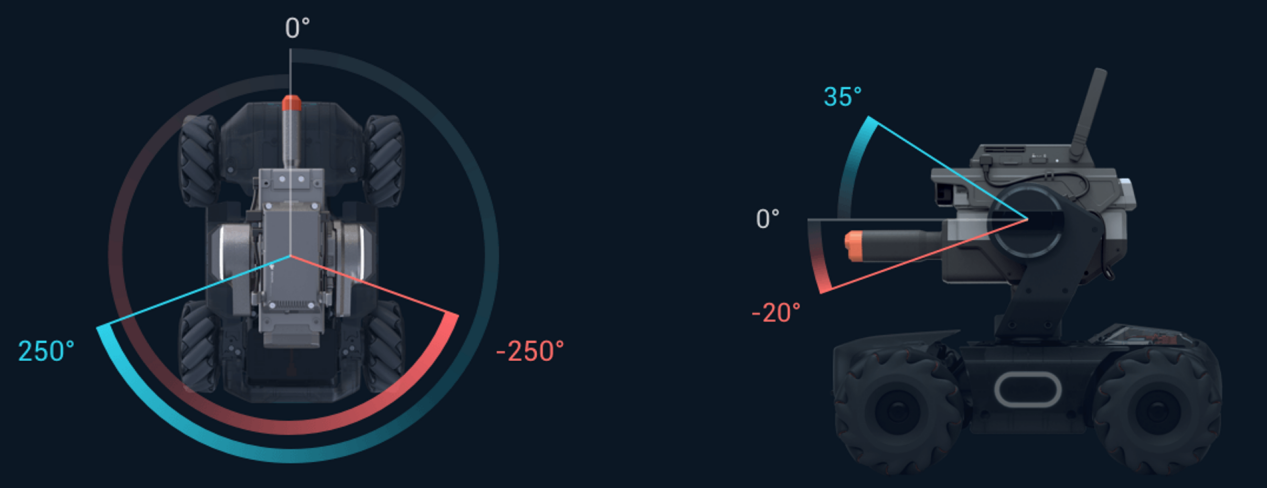
**目录**

1. **实验记录及总结**
2. **理论学习与总结**
3. **实践任务与设计**
4. **方案实践与记录**
5. **拓展任务**
6. **实践任务与设计**
7. **方案实践与记录**

## 实验记录及总结

1. **理论学习与总结**

**【云台旋转角度】**



航向轴：250°~ 250°

俯仰轴：-20°~ 35°

**【姿态角】**

航向轴电机是控制云台绕航向轴旋转，我们左看右看时，是云台绕着“航向轴”运动，所旋转的角度为航向角yaw。俯仰轴电机是控制云台绕俯仰轴轴旋转，我们低头抬头时，是云台绕着“俯仰轴”运动。所旋转的角度为俯仰角pitch。除此之外，当我们左歪脖右歪脖时，便是绕着“翻滚轴”运动。所转的角度为翻滚角roll。



**【编程模块】**

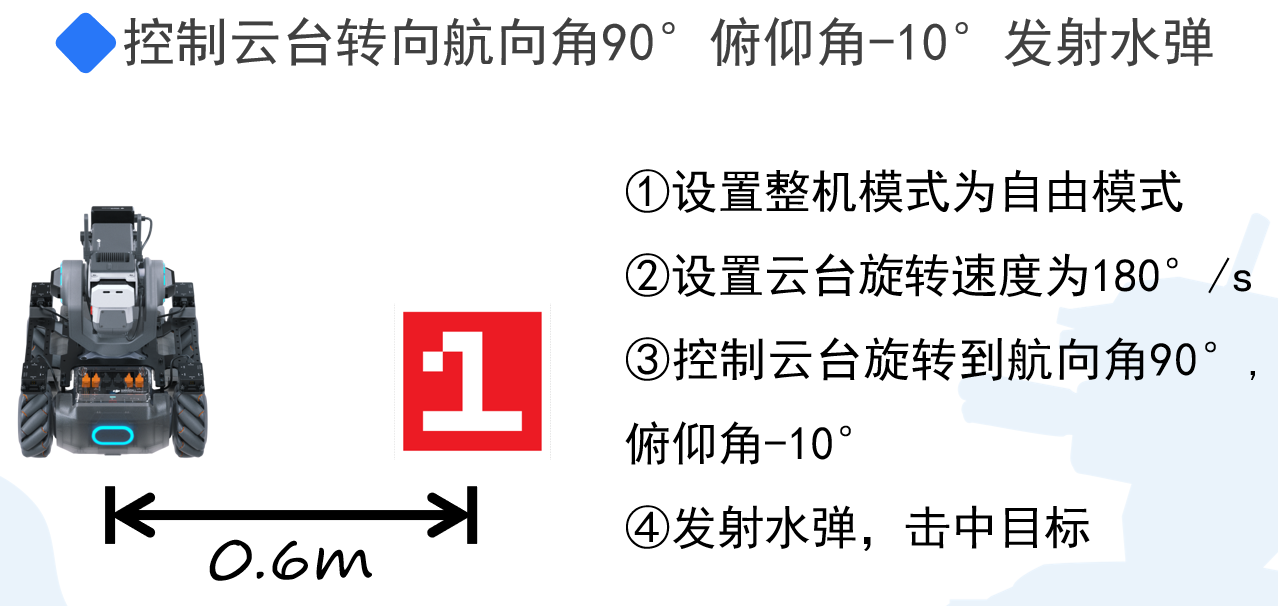






**2、实践任务与设计**

**（1）任务**



1. **方案设计**

- 设置整机运动为自由模式

- 设置云台旋转速度为180度/秒

- 控制云台旋转到航向轴90度，俯仰轴-10度

- 单次发射水弹

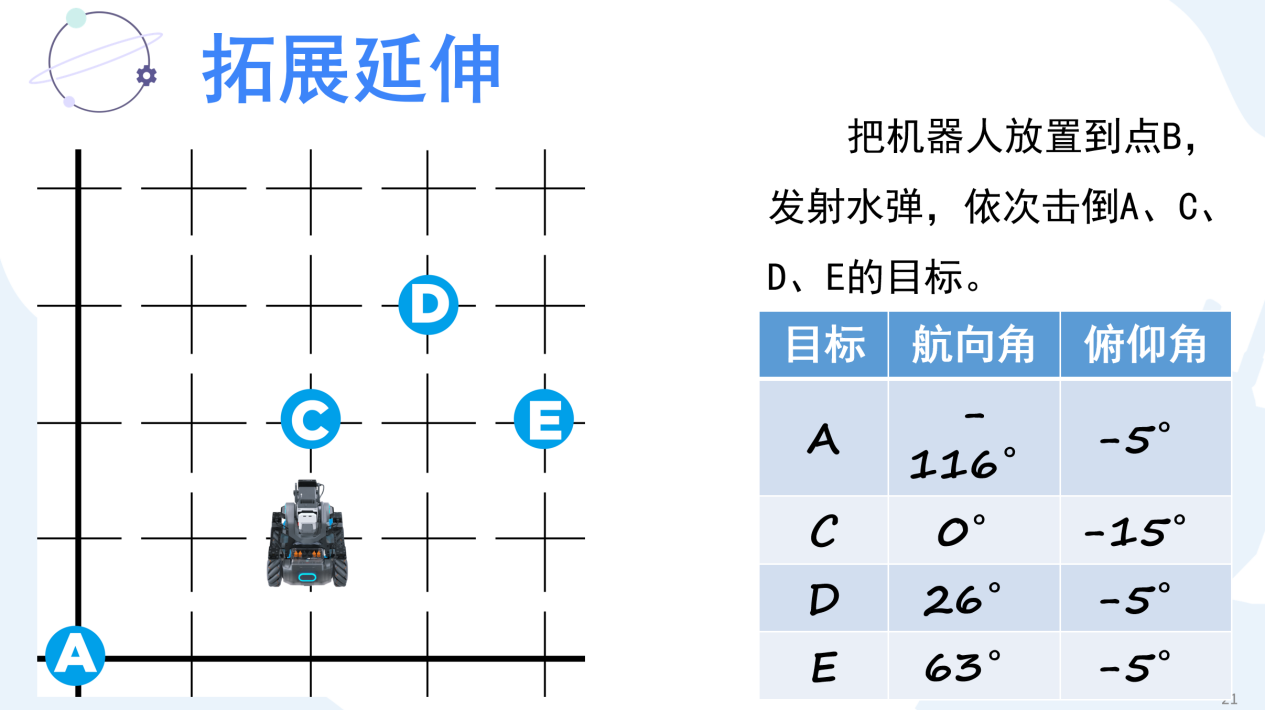
**3、方案实践与记录**



## 二、拓展任务

**1、实践任务与设计**

1. **任务**



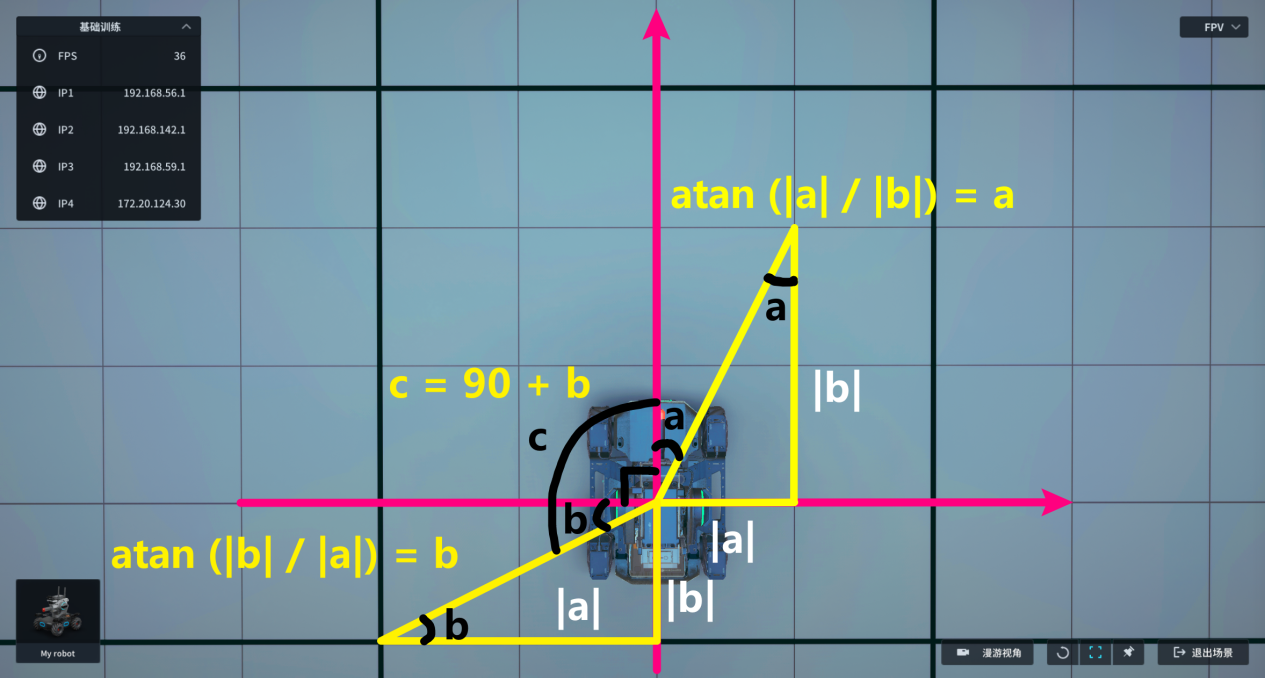
1. **方案设计**

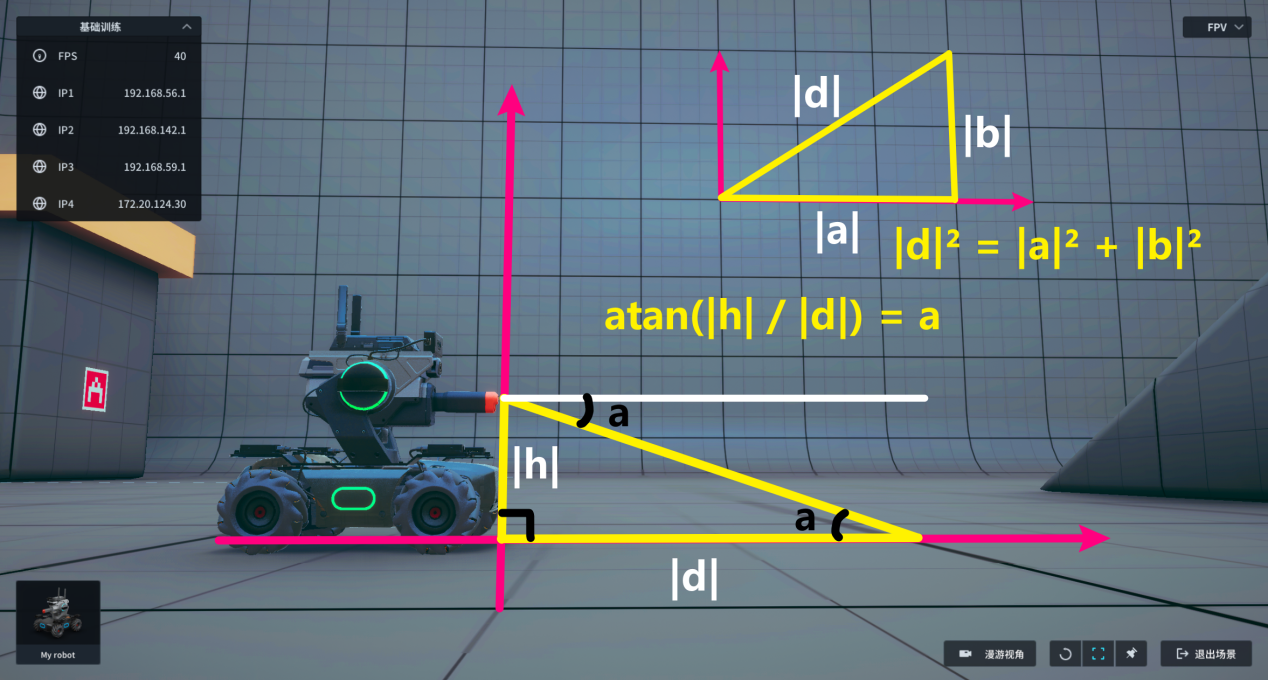
- 确定每个目标点与原点的x轴和y轴的差值的绝对值

- 根据坐标差利用 atan() 函数计算角度

- 根据目标点具体的象限值，确定具体的旋转角度

**【分析图】**

****

****

**【伪代码】**

    // DJI-EP-定点射击

    // 周吉瑞 2021/05/10

    // 伪代码（C语言风格）

    // 全局变量

*double* turret\_height **=** 0.30;    // 炮塔高度（30cm）

*double* x0 **=** 0;                  // 原点 x 坐标（默认为 0）

*double* y0 **=** 0;                  // 原点 y 坐标（默认为 0）

*double* x;                       // 目标点 x 坐标

*double* y;                       // 目标点 y 坐标

*double* a;                       // 原点与目标点的 x 坐标差值

*double* b;                       // 原点与目标点的 y 坐标差值

*double* h\_angle;                 // 待旋转的航向角

*double* f\_angle;                 // 待旋转的俯仰角

    main() {

        setSport(freedom);          // 设置整机运动（自由模式）

        setRotateSpeed(180);        // 设置云台旋转速率（180度/秒）

        where {

            // 默认每一个小方格的边长为50cm

            scanf("%lf%lf", **&**x, **&**y);               // 输入待射击的目标点坐标

            getDifference(*double* x, *double* y);     // 求原点与目标点的差值

            moveCannonAngle();                     // 调整炮台角度（航向角、俯仰角）

            shooting(once);                        // 射击水弹（单次发射）

            wait(3);                               // 等待（3秒）

        }

    }

    // 函数：求原点与目标点的差值

    getDifference(*double* *x*, *double* *y*) {

        // 默认原点为 (0, 0)

        a **=** x **-** x0;

        b **=** y **-** y0;

    }

    // 函数：调整炮台角度（航向角、俯仰角）

    moveCannonAngle() {

        // 航向角计算

        // 计算公式：航向角角度 = atan(对/邻)

        // 分类讨论：

        // 目标点位于原点的第一象限

**if** (a **>=** 0 **&&** b **>=** 0) {

            h\_angle **=** atan(a **/** b);

        }

        // 目标点位于原点的第二象限

**if** (a **<=** 0 **&&** b **>=** 0) {

            h\_angle **=** **-**1 **\*** atan(abs(a) **/** b);

        }

        // 目标点位于原点的第三象限

**if** (a **<=** 0 **&&** b **<=** 0) {

            h\_angle **=** **-**90 **-** atan(abs(b) **/** abs(a));

        }

        // 目标点位于原点的第四象限

**if** (a **>=** 0 **&&** b **<=** 0) {

            h\_angle **=** 90 **+** atan(abs(b) **/** a);

        }

        // 俯仰角计算

        // 计算公式：航向角角度 = -(atan(对/邻)

        f\_angle **=** **-**1 **\*** atan(turret\_height **/** (sqrt(abs(a) **\*** abs(a) **+** abs(b) **\*** abs(b))));

        // 调整炮台航向角

        moveCourseAngle(h\_angle);

        // 调整炮台俯仰角

        movePitchAngle(f\_angle);

    }

**2、方案实践与记录**

