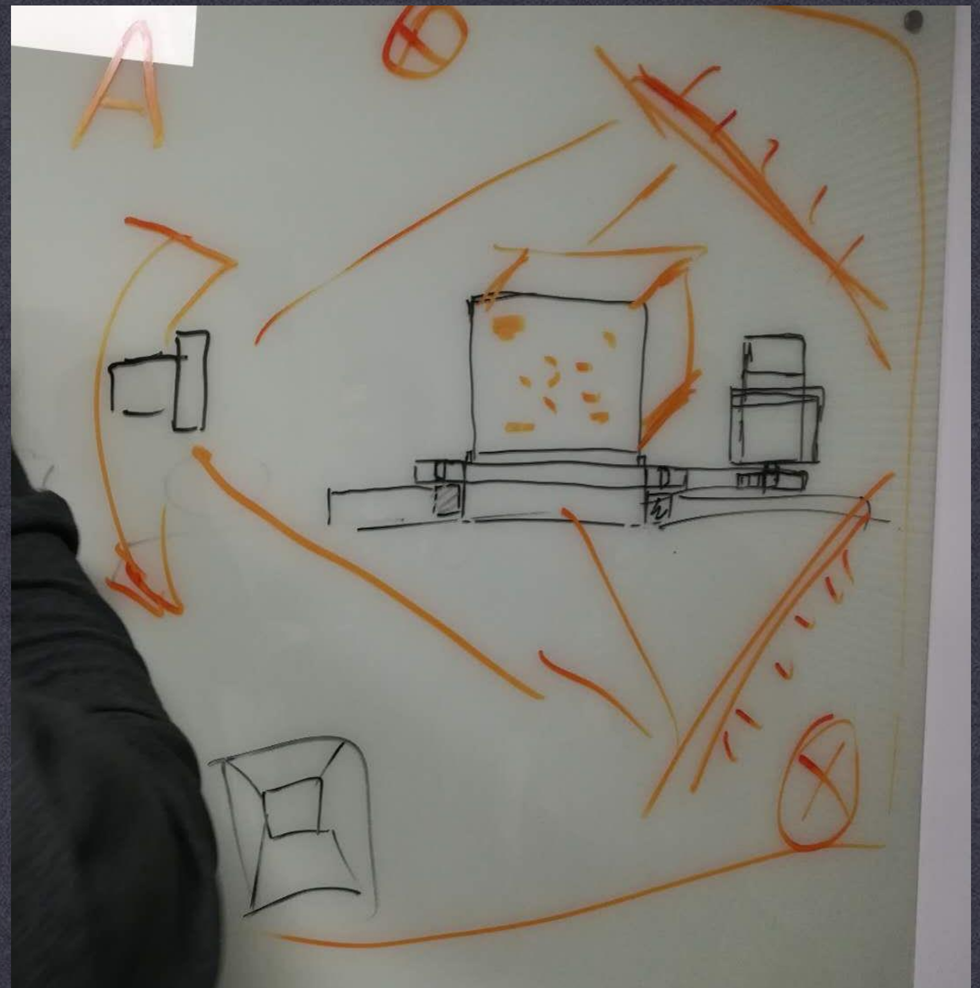


PROJECT

查无此队

TEAM 404 NO FOUND

DATE 1/25-2/3

CLIENT 张珈伟 王冠 廖洽源 贺贯齐 龚智礼 温传奇 华明浩

工作分配
Role&Task Assignment in Group

- 组长：张珈伟
- 机械：
  龚智礼
  温传奇
  华明浩
- 嵌入：
  贺贯齐
  廖洽源
- 算法：
  王冠
  张珈伟

团队特色及研发主旨
Team Characteristics and R&D Theme

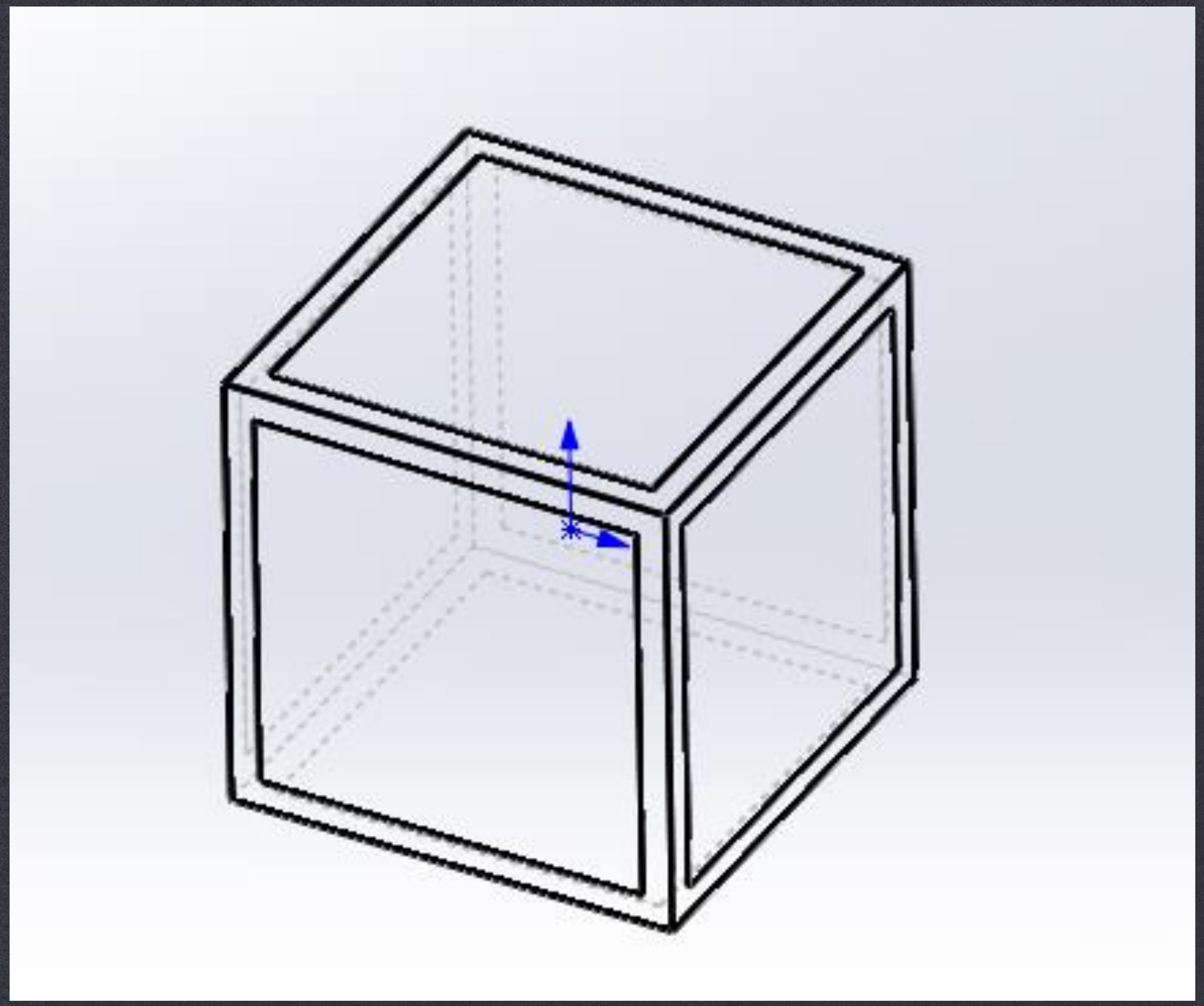
- 机械设计：Mechanic：
  优秀的结构实现能力及丰富的搭建知识Outstanding structure implementation capability and abundant construction knowledge
  考虑最大的冗余范围以及最小化不可控因素Consider maximum redundancy and minimize uncontrollable factors

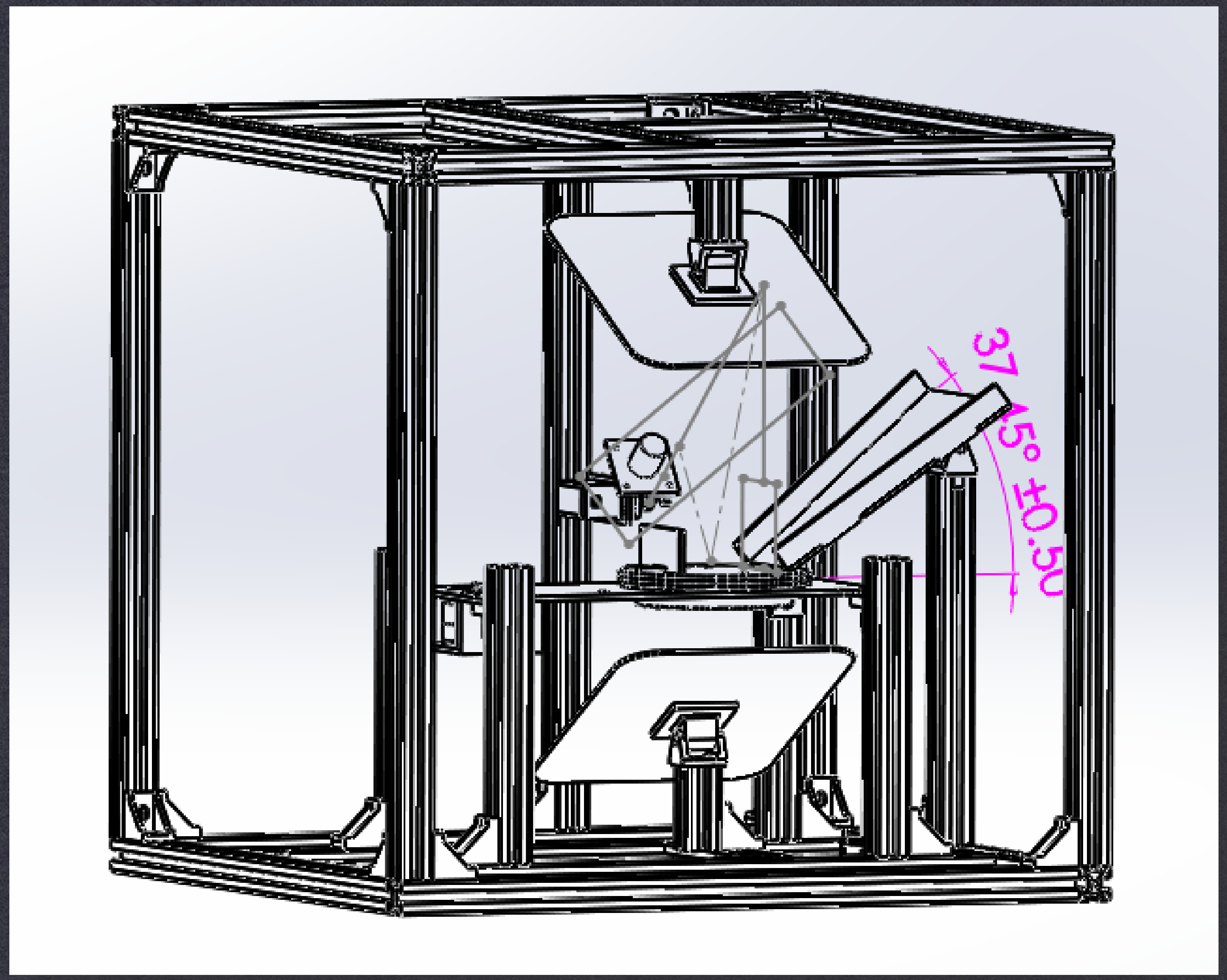
- 嵌入式：Embedded System
  崇尚学术，技术优先 Advocating academy, giving priority to technology
  自动化 Automation

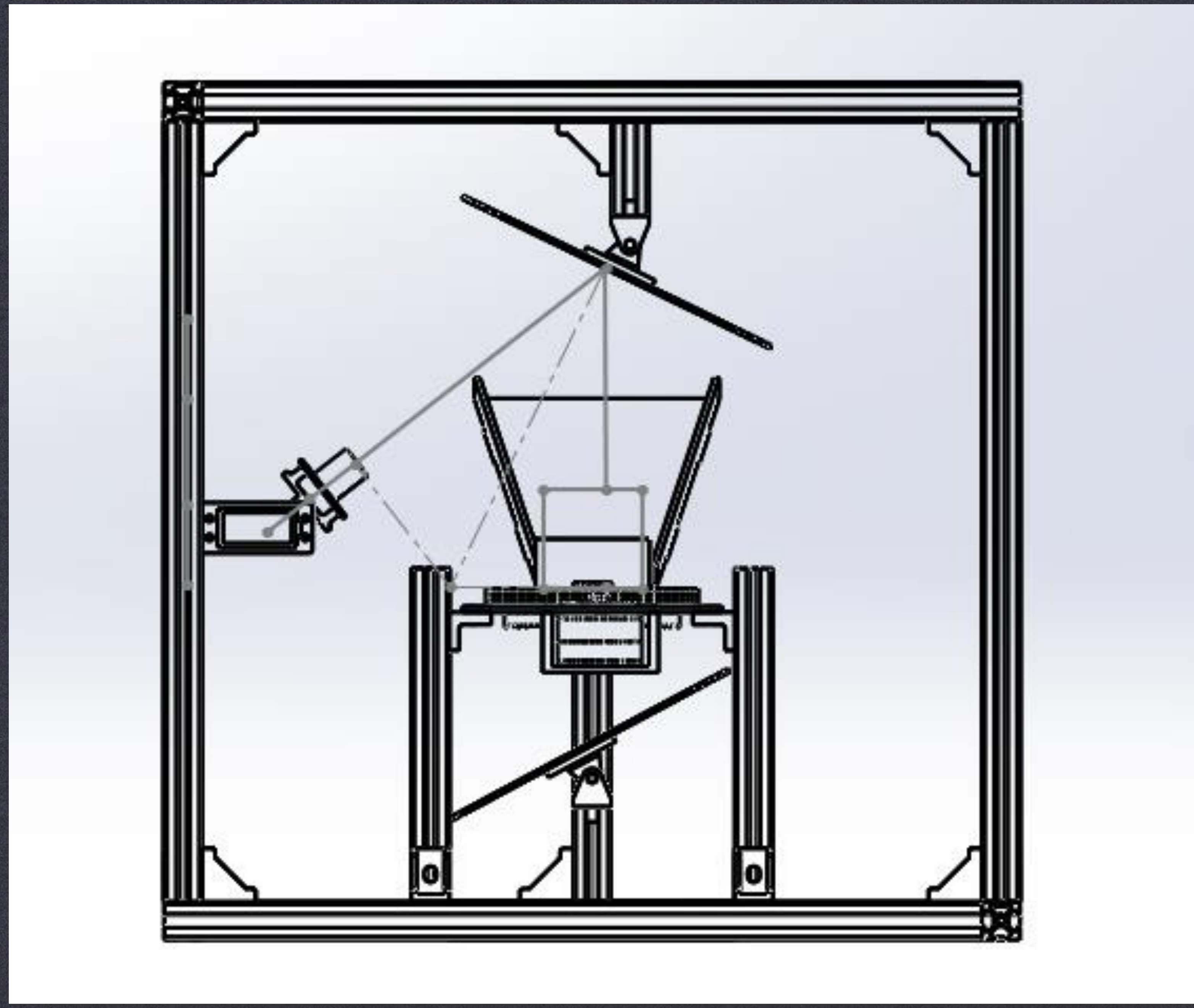
- 算法：
  完全自动化 用户体验良好 Fully Automation & User-Friendly
  超高稳定性 High Reliability
  使用条件闲置小 功能丰富 Wide Application Scenario

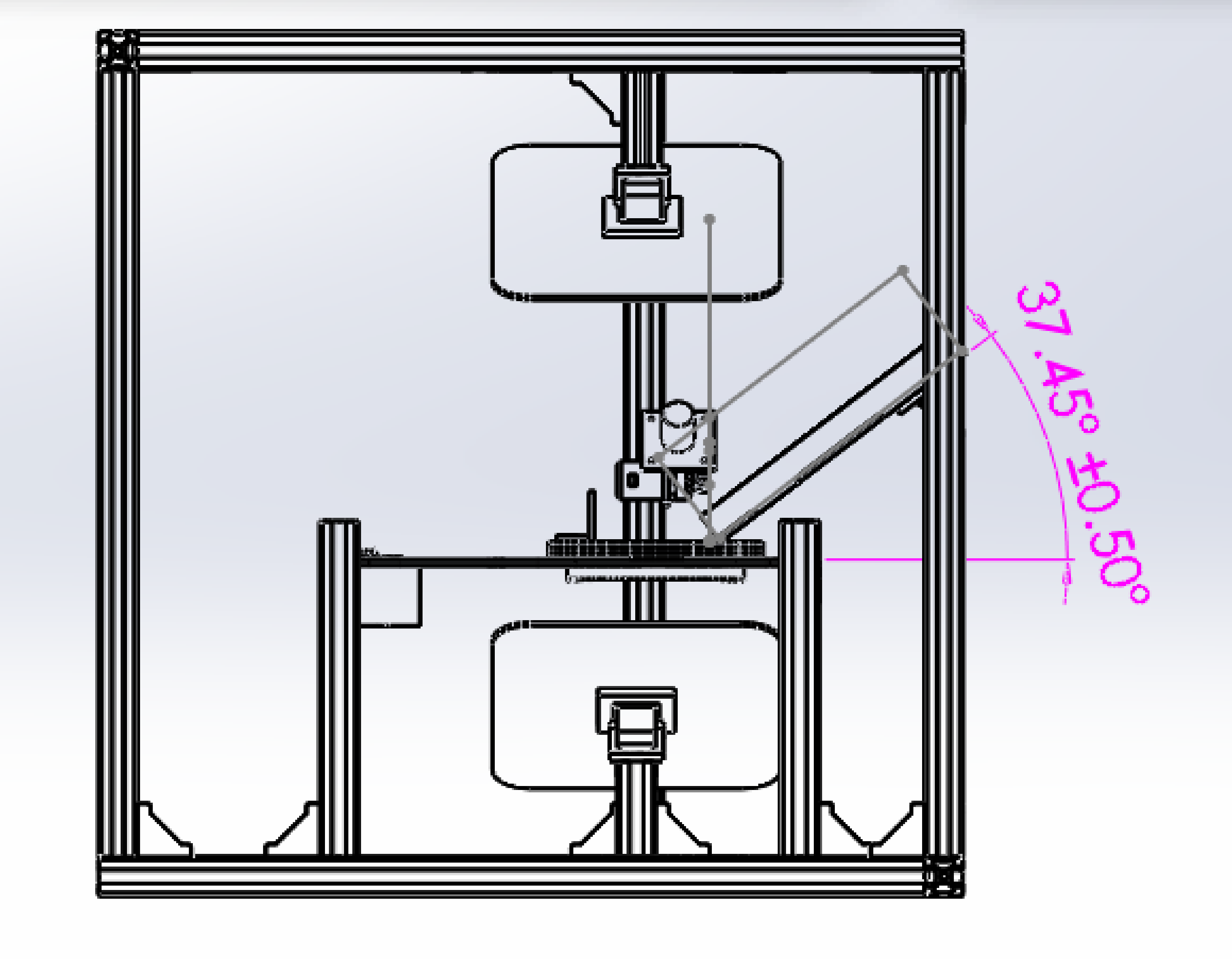
丰富的搭建知识

construction knowledge





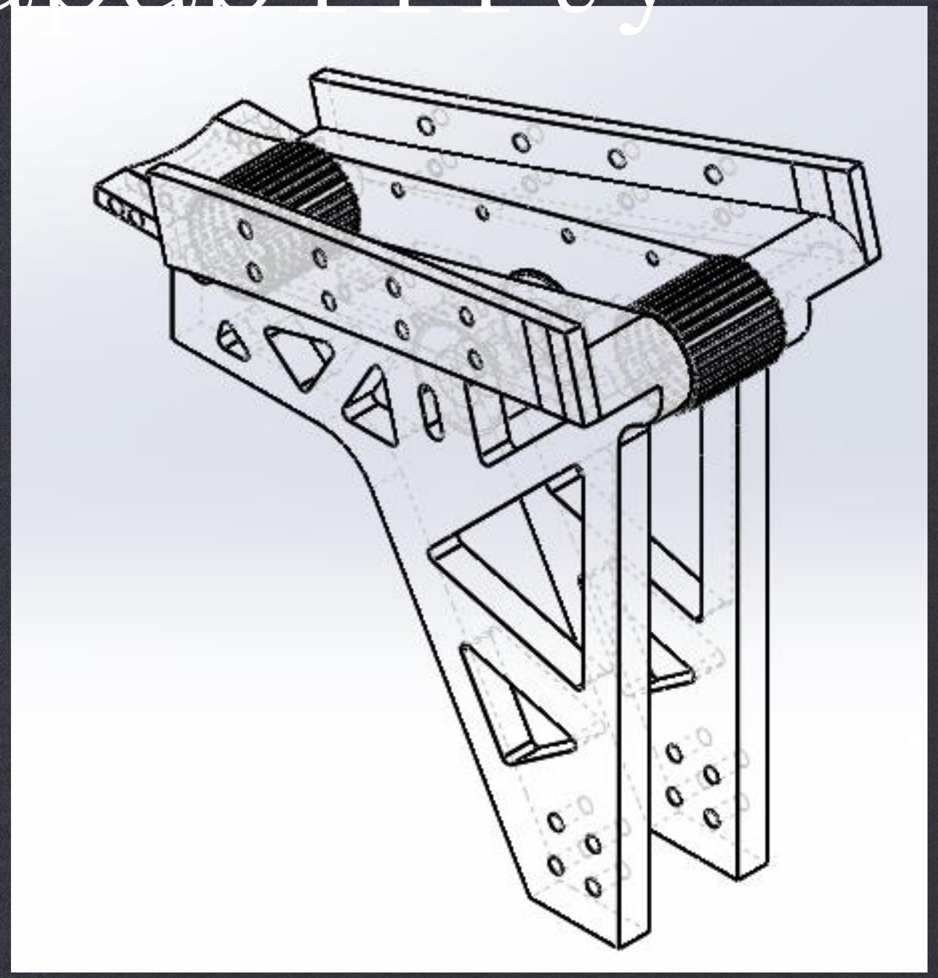


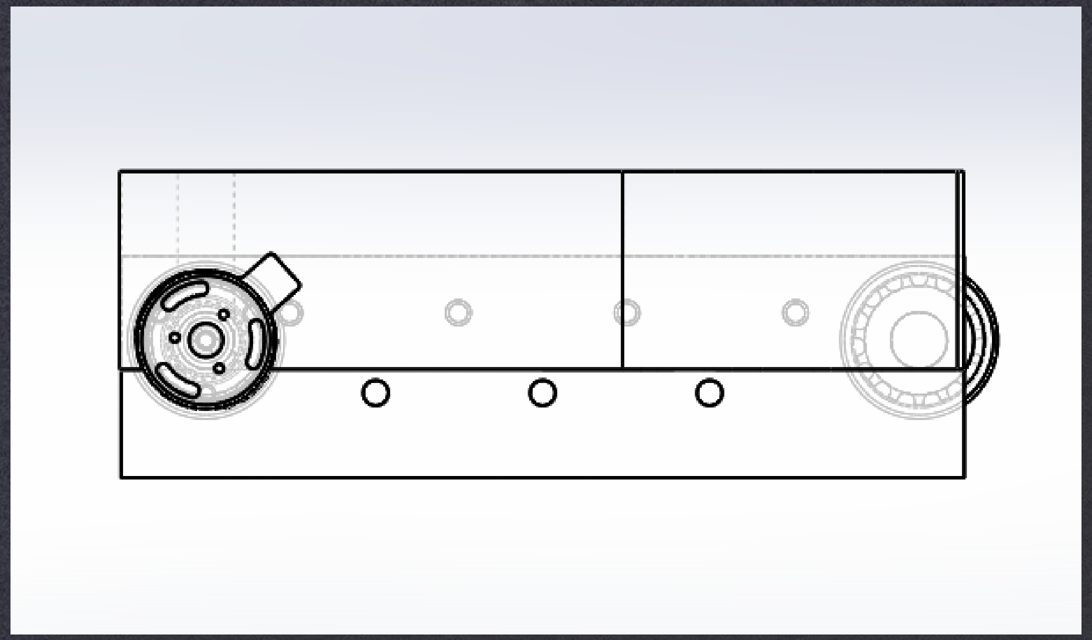


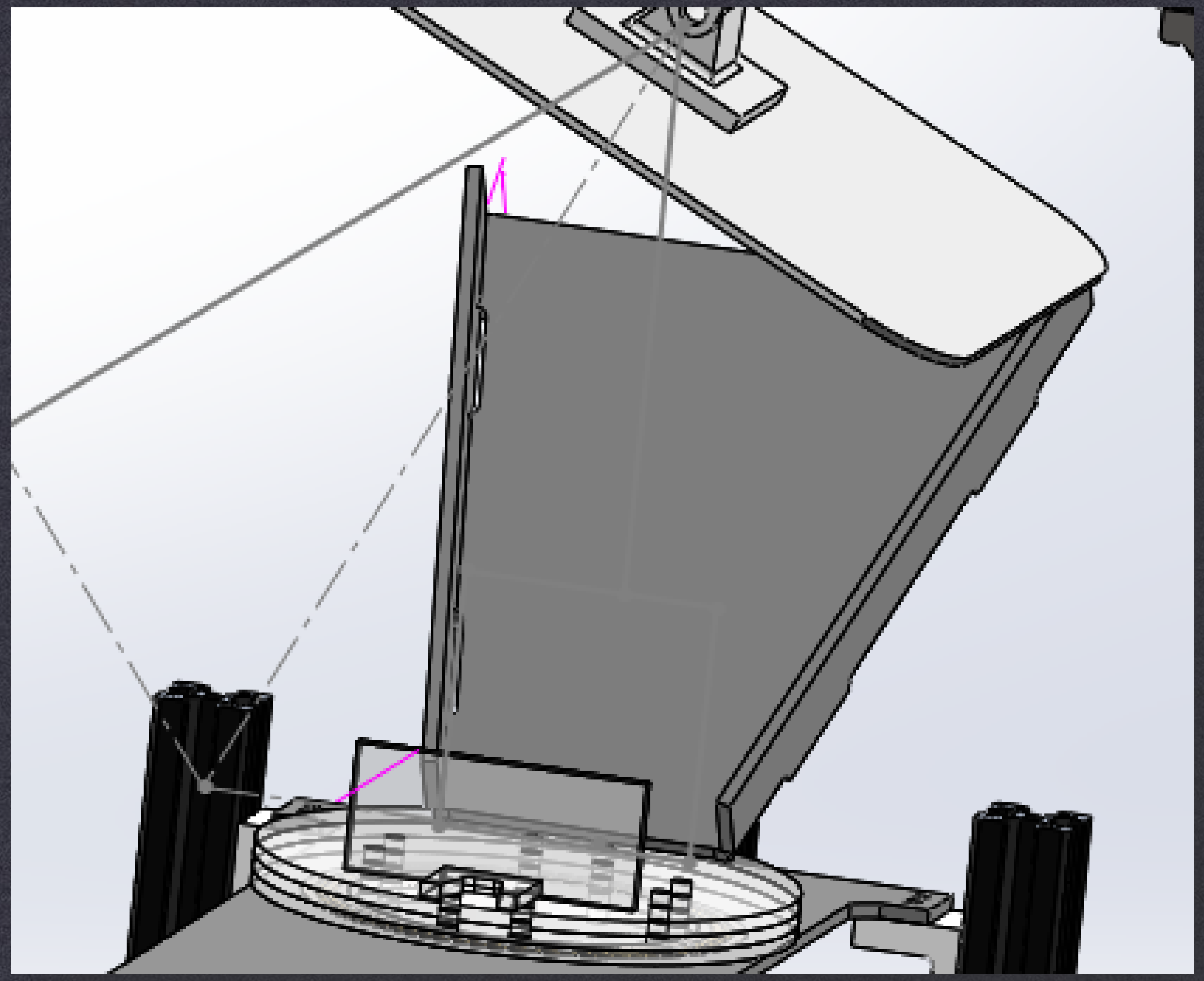
优秀的结构实现能力

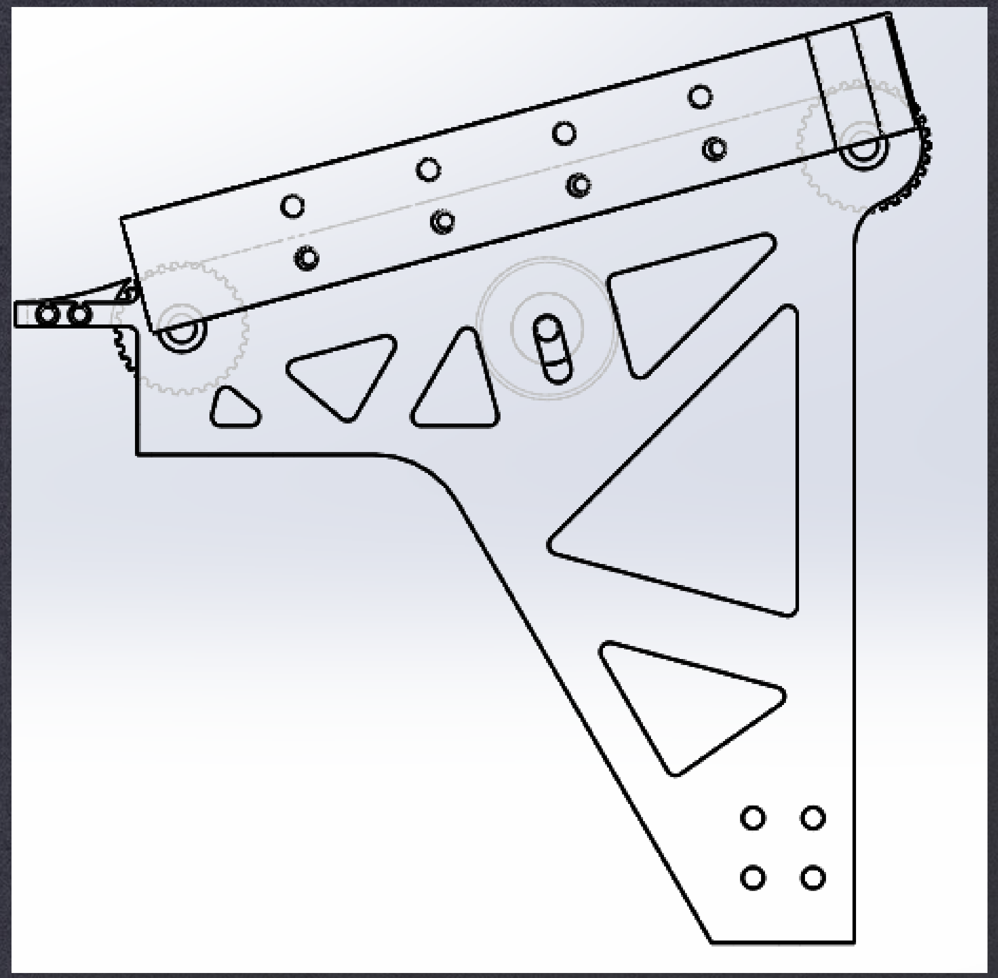
THE INTAKE SYSTEM

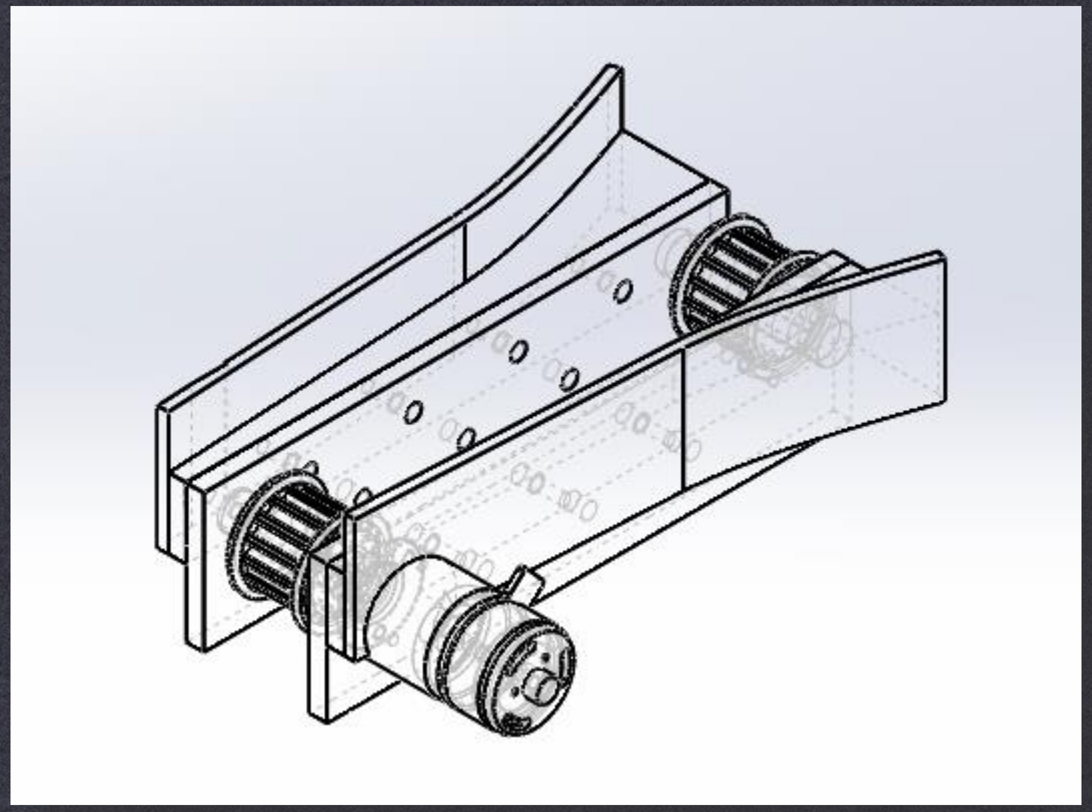
Outstanding structure implementation capability







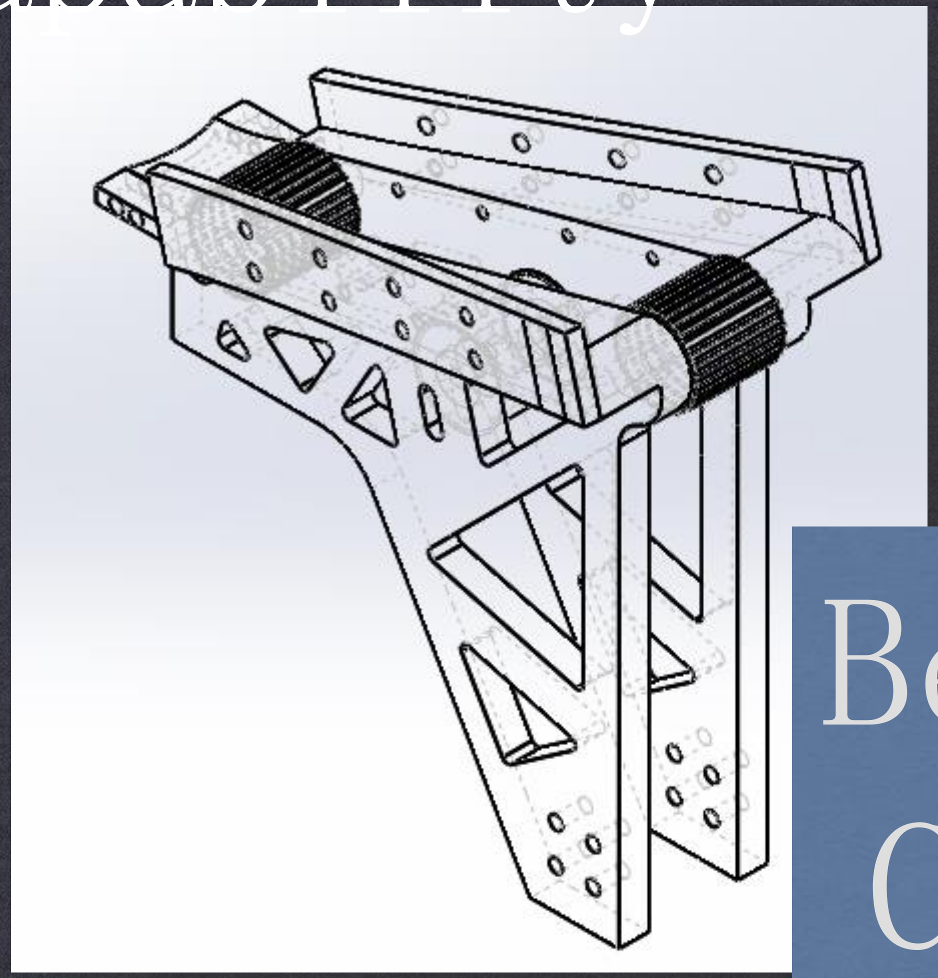


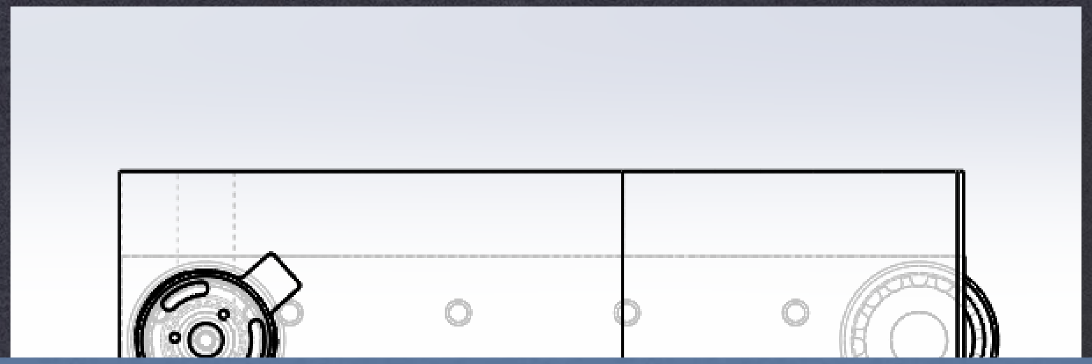


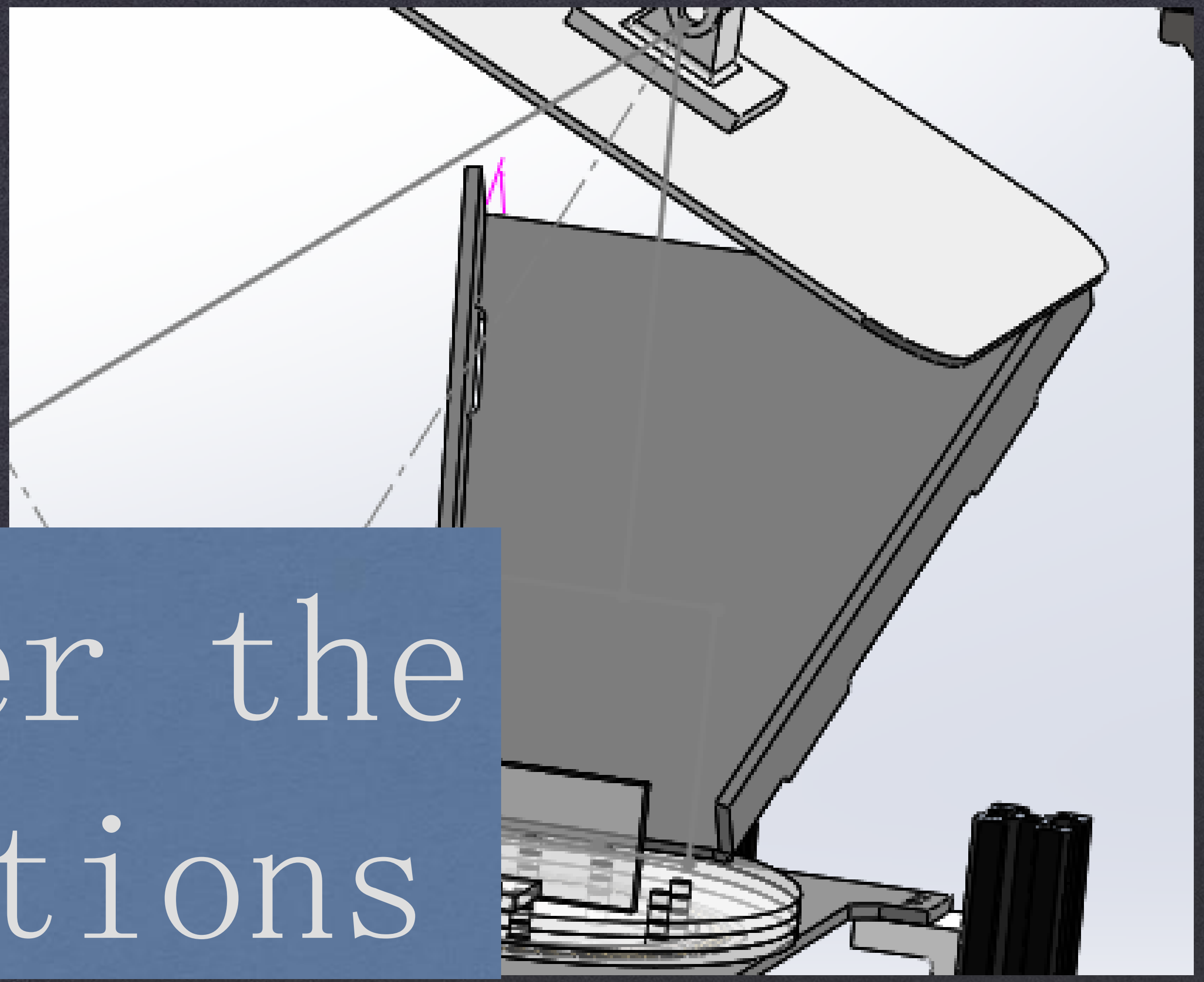
优秀的结构实现能力

THE INTAKE SYSTEM

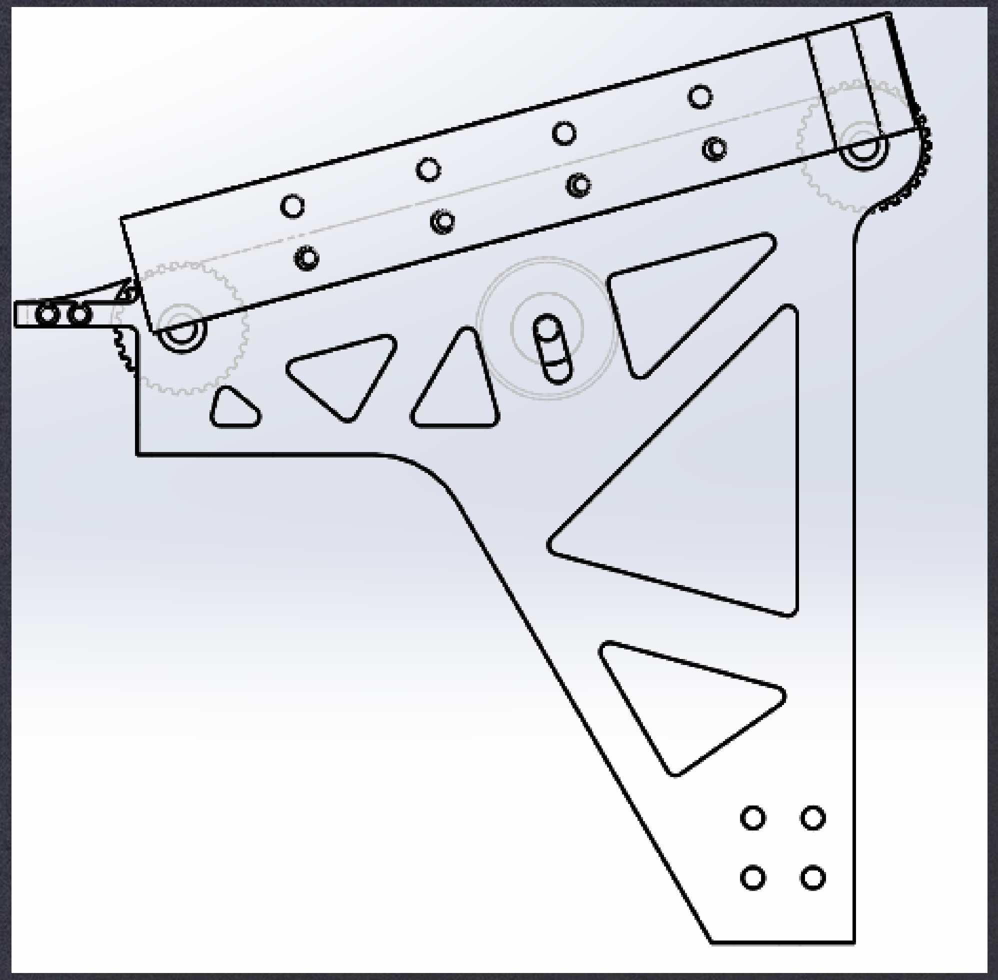
Outstanding structure implementation capability

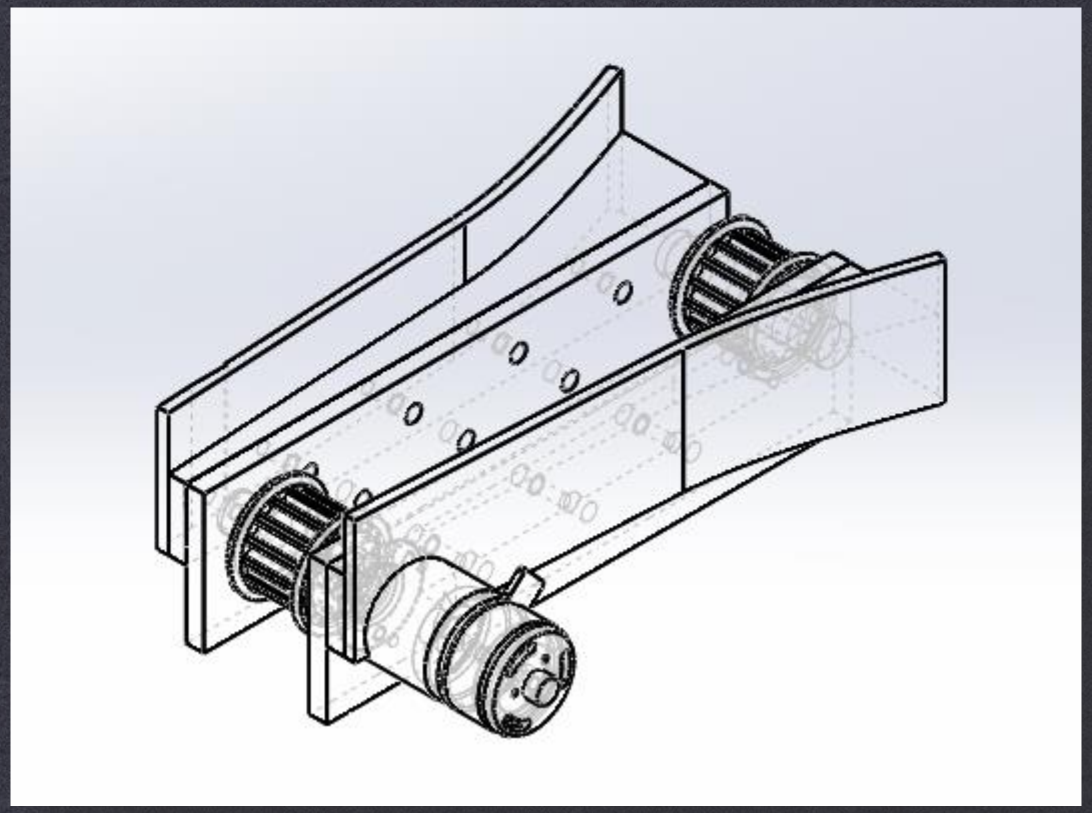


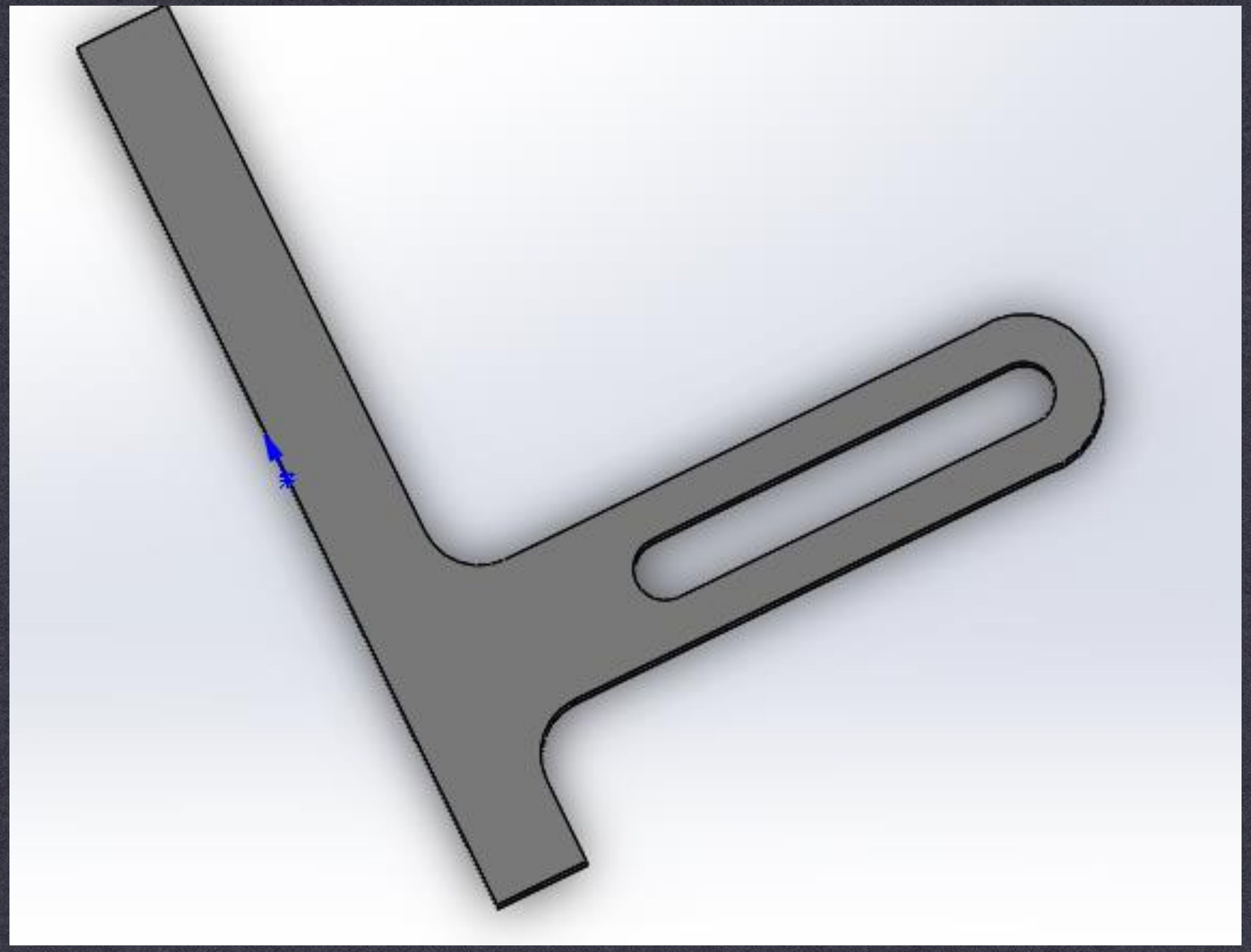




Being Able to Build Under the Current Machining Conditions

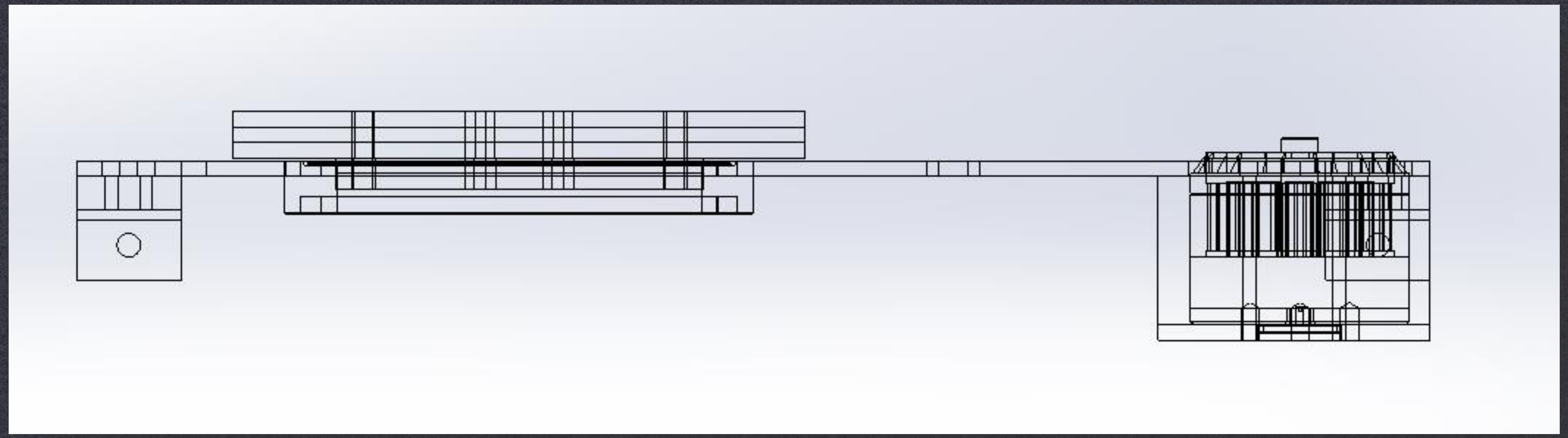


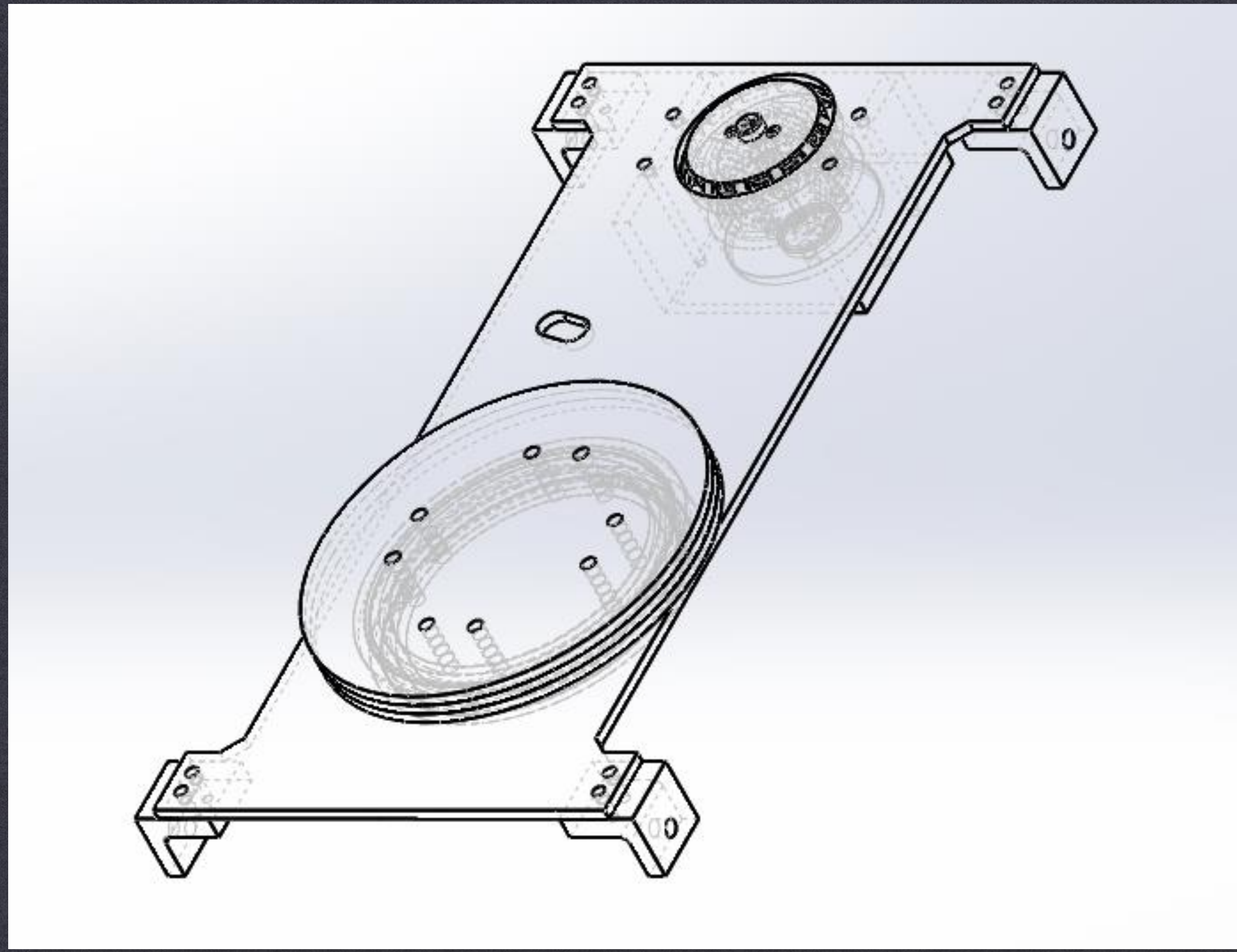


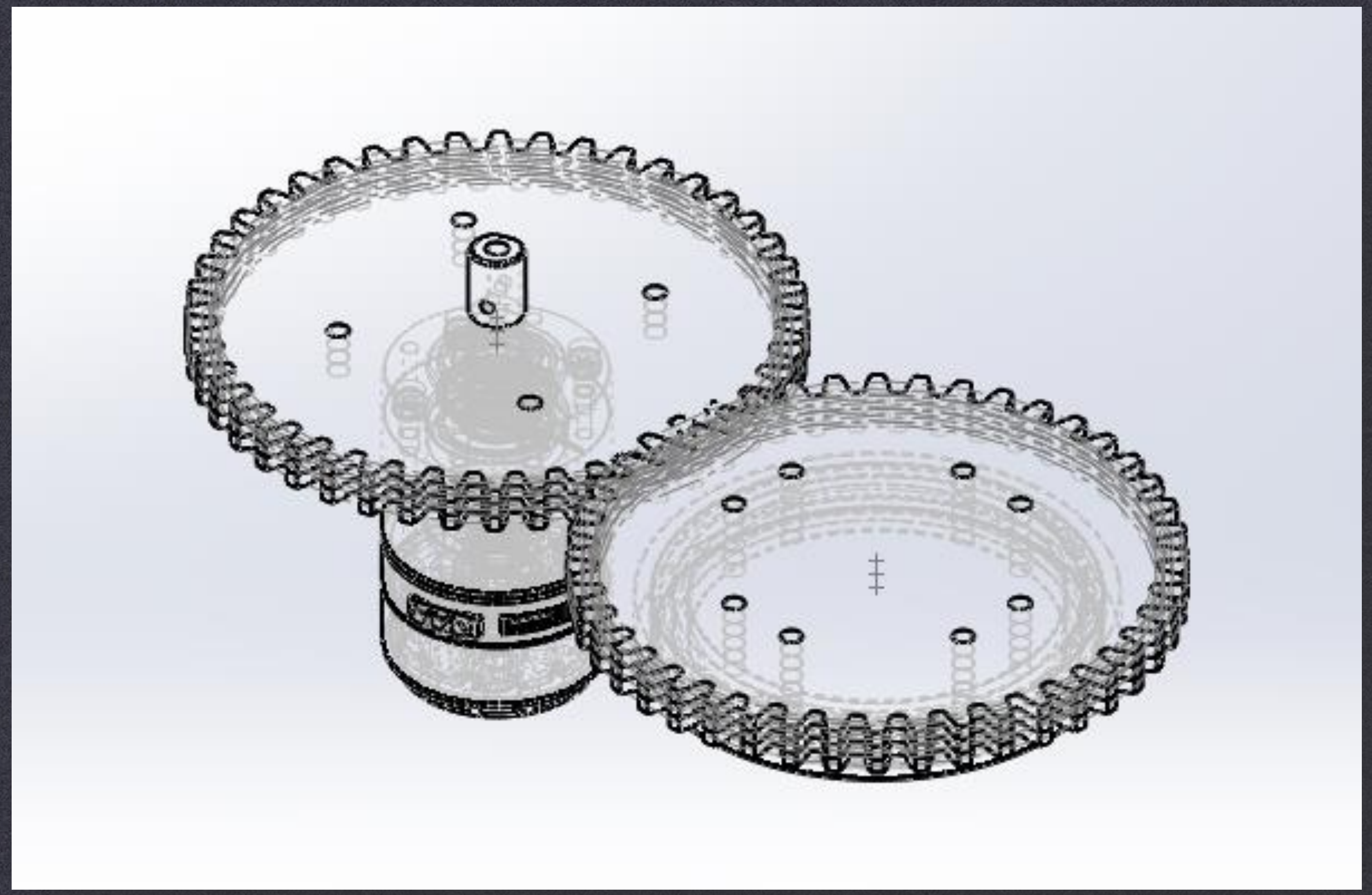


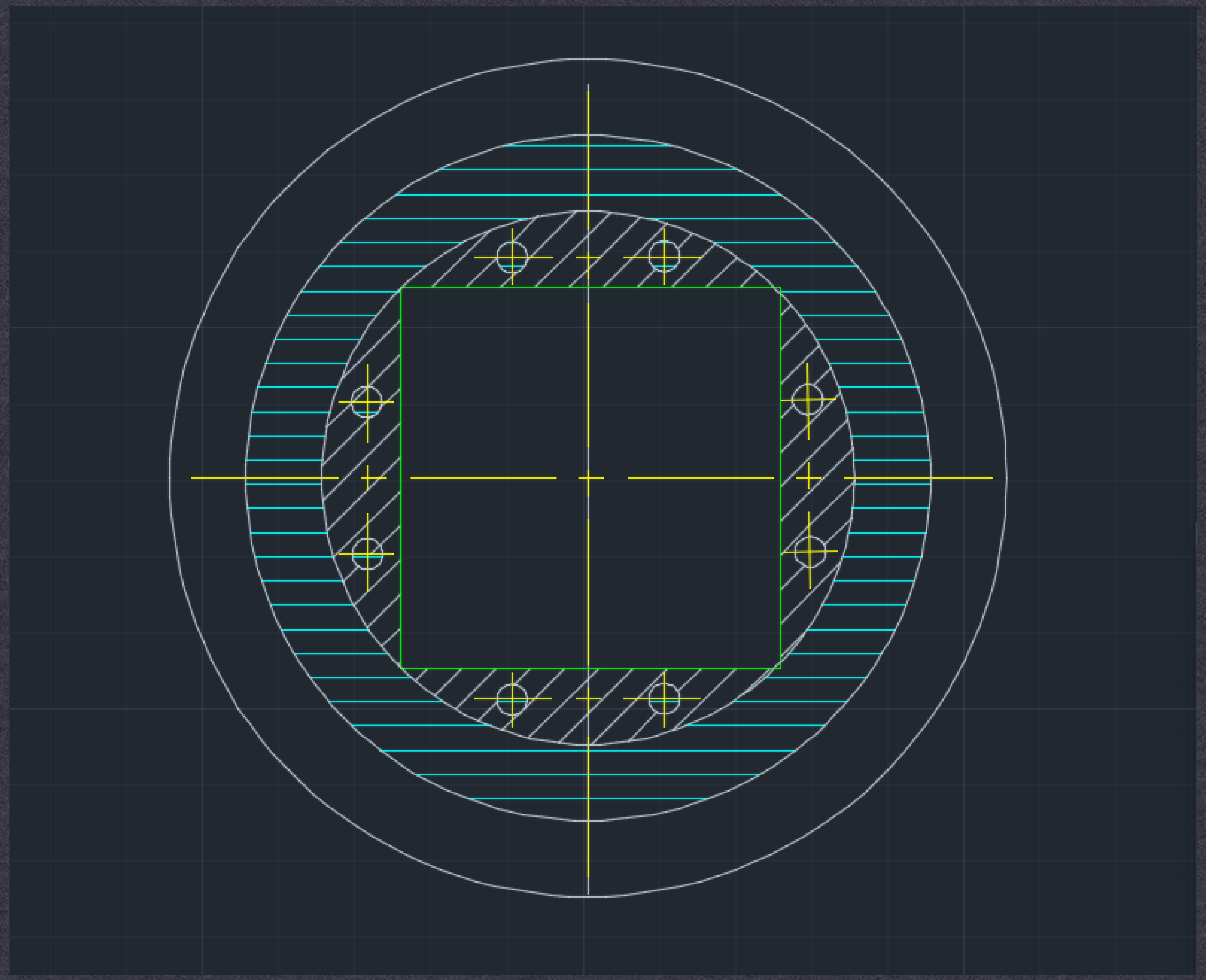
然而，我们还是遇到了我们的能力所无法解决的问题

However, we still run into a significant problem that can not be solved by our capability





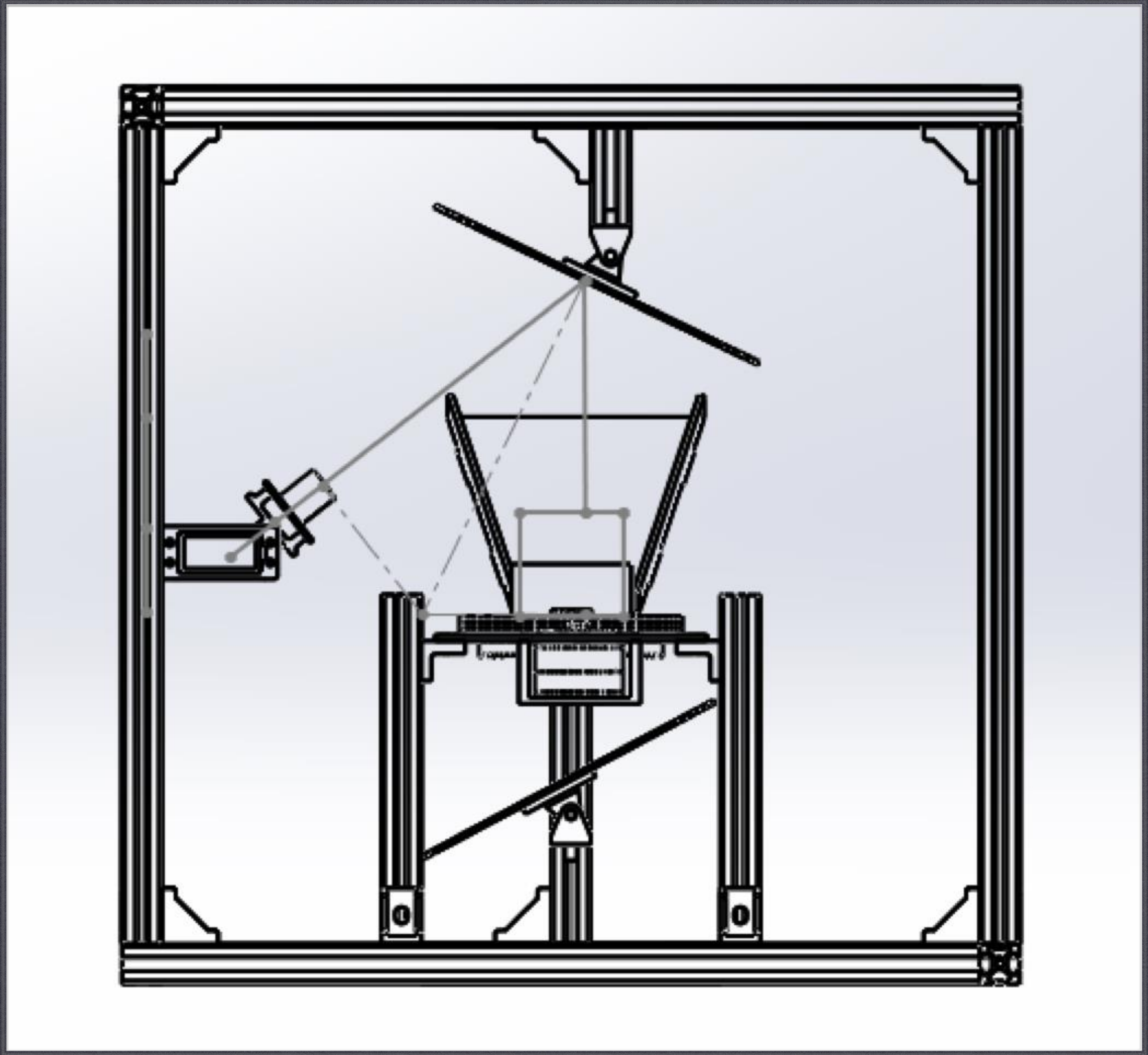




某种程度上，我们碰上的未预料到的加工问题把我们的结构实现能力完全压榨 To some extents, we need to thank to the problem as it push our capability to the limit.

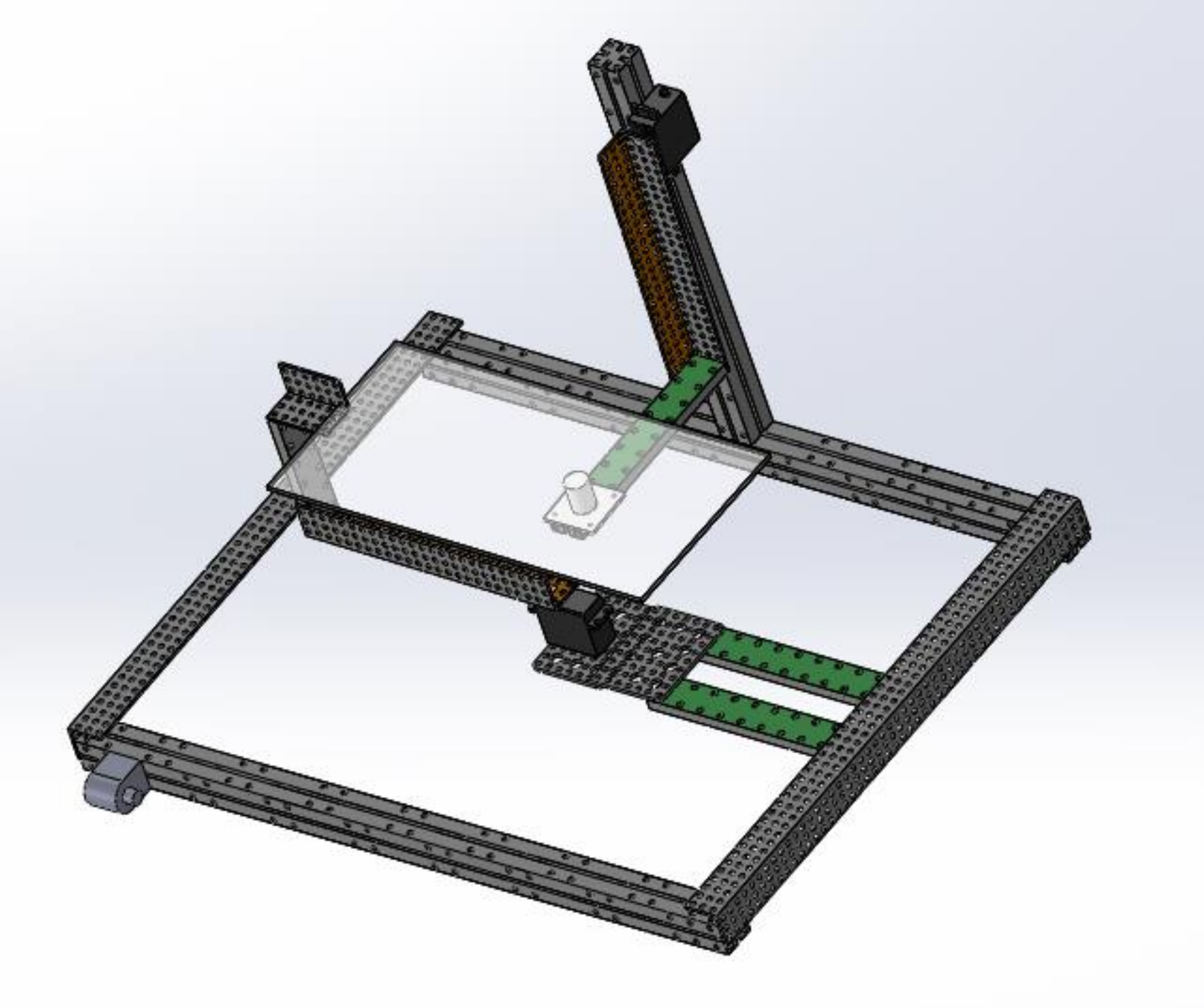
最小化不可控因素

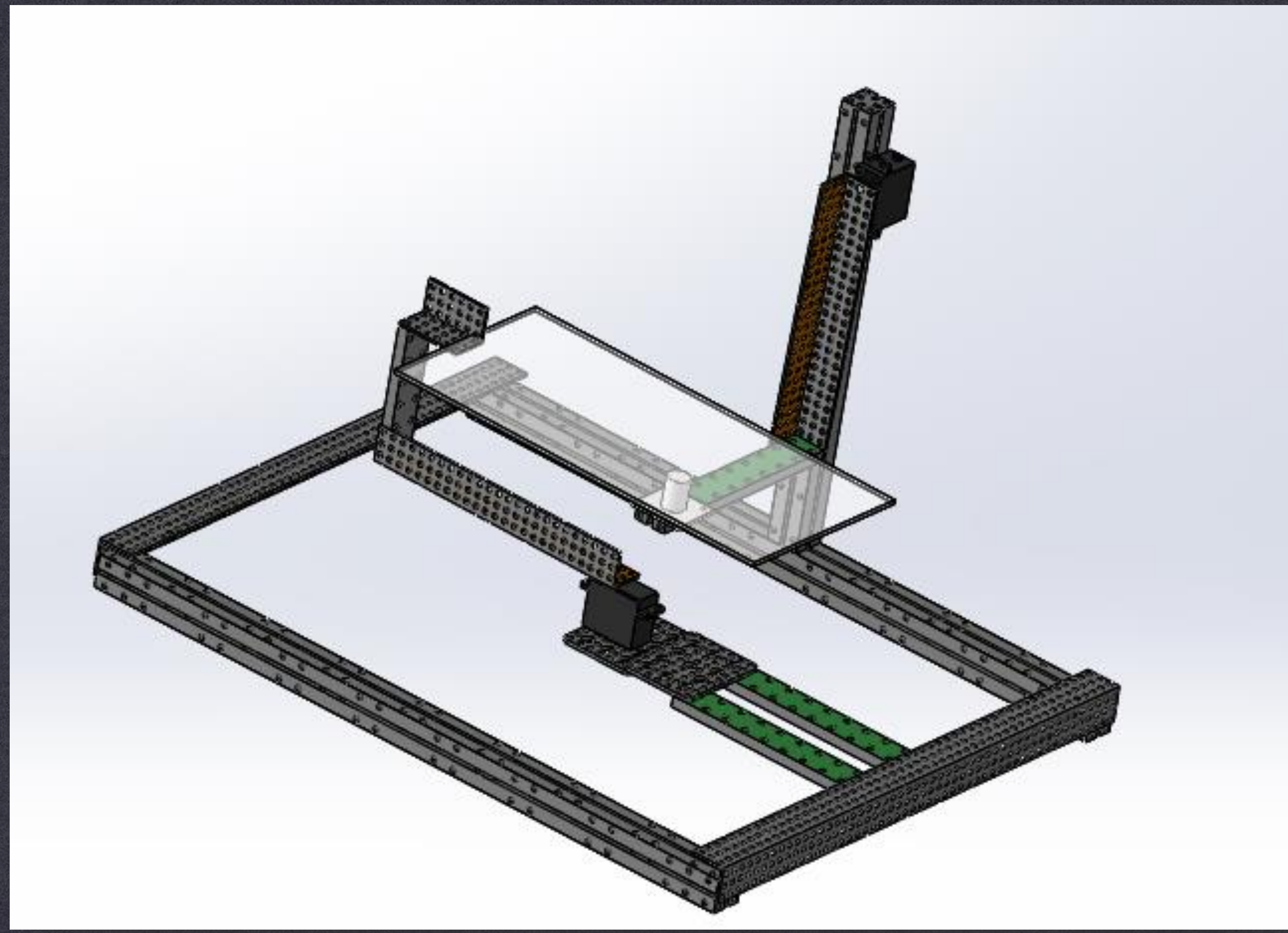
Minimize uncontrollable factors

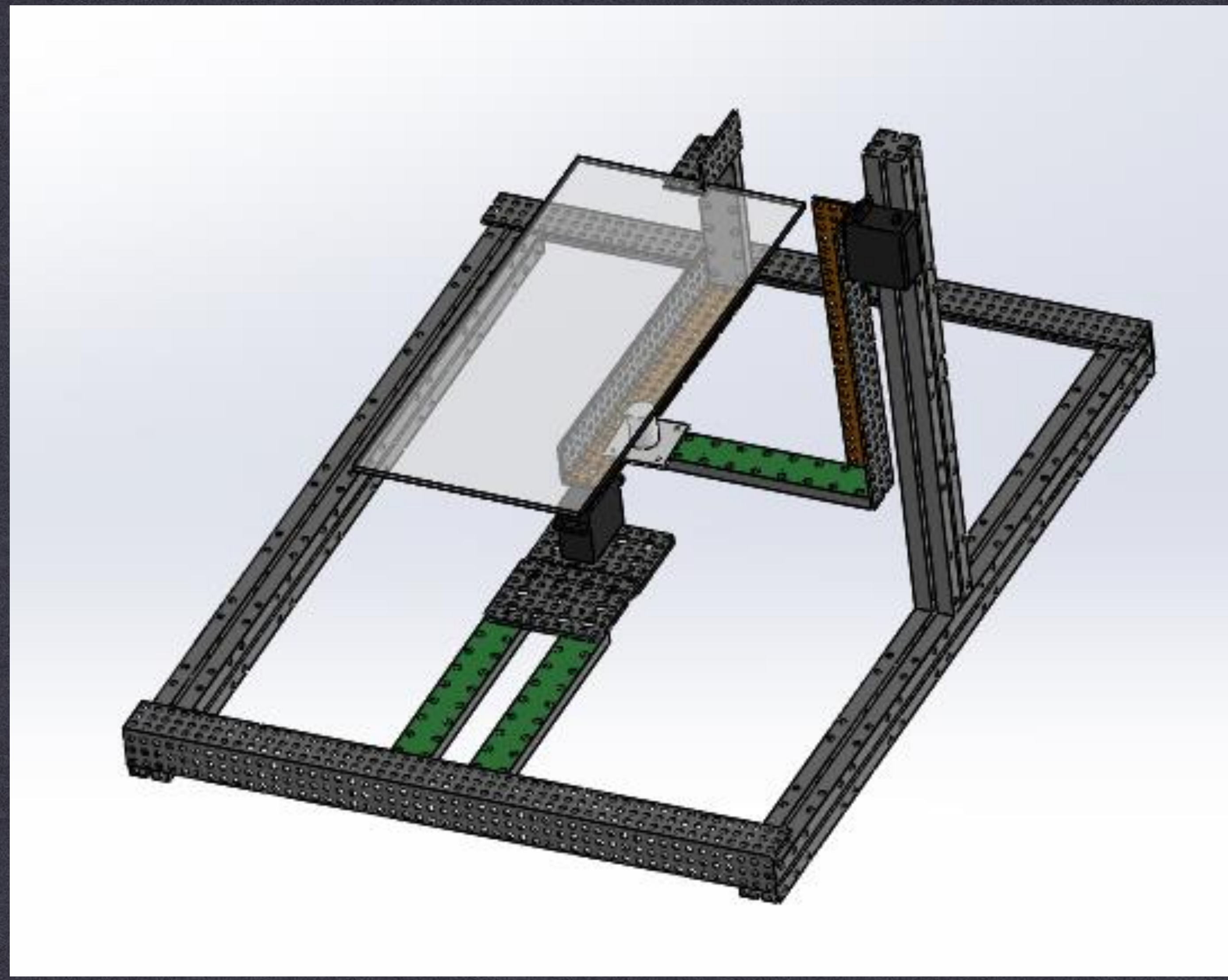


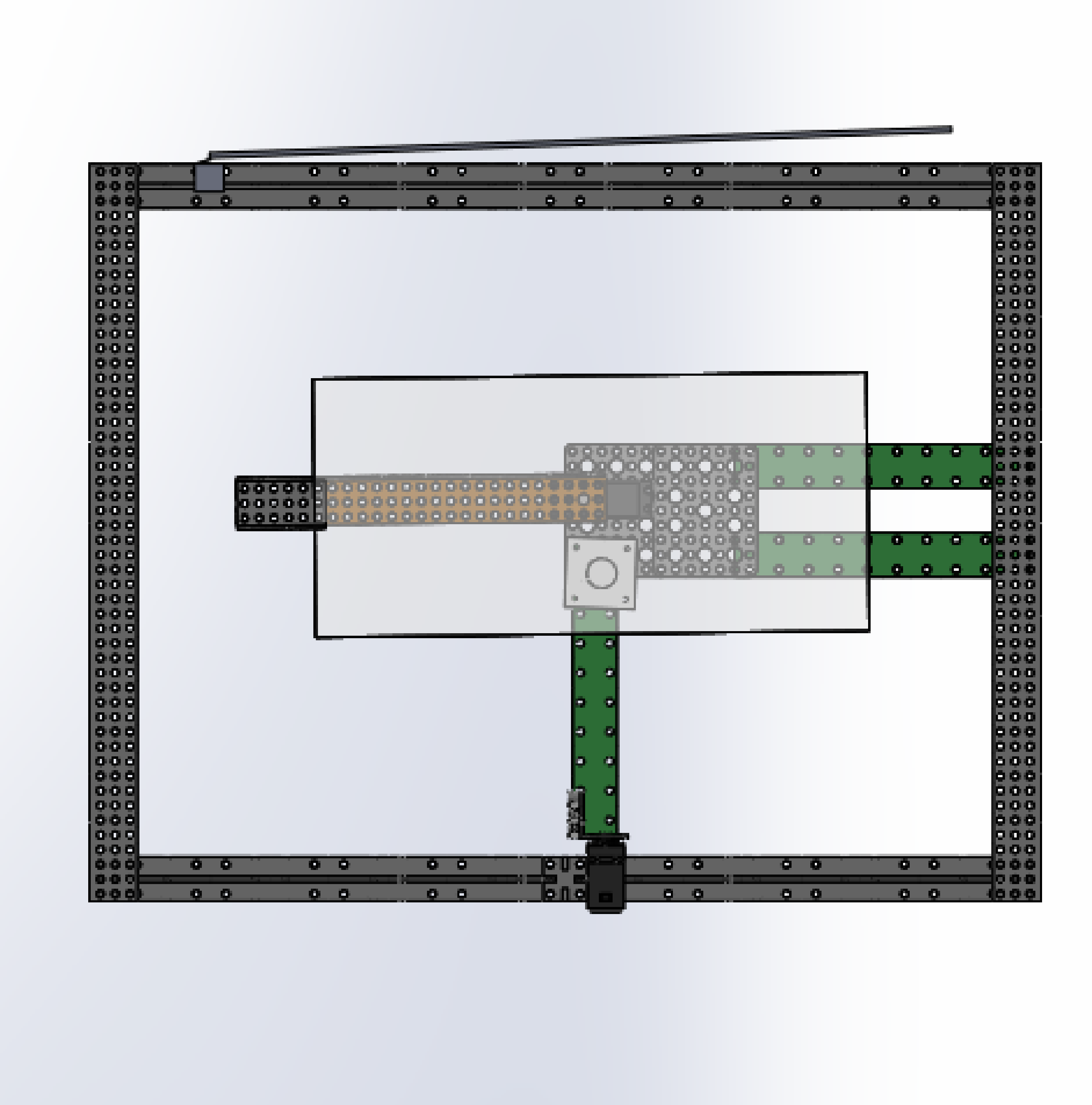
最大的冗余范围

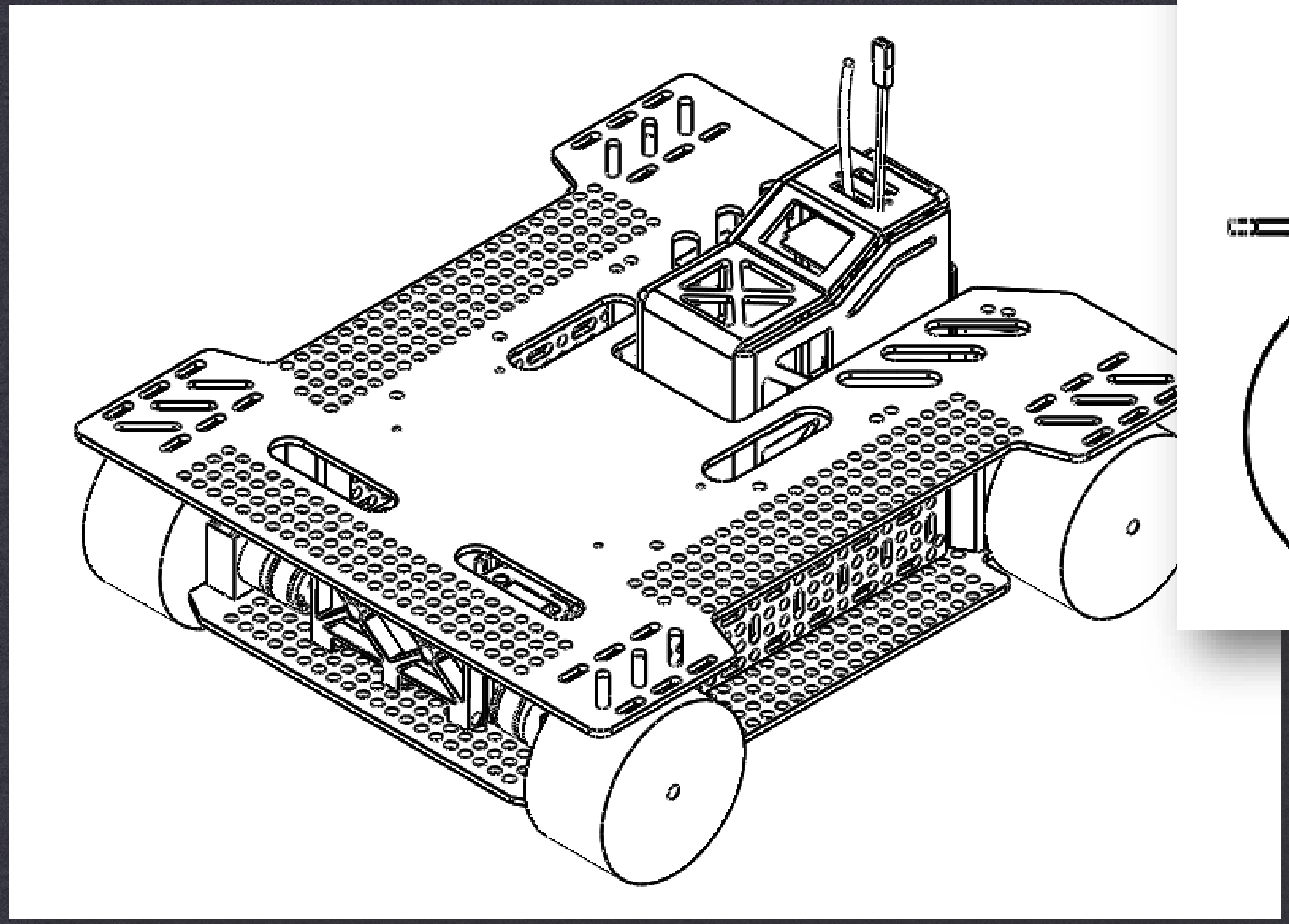
Maximum the Redundancy of the Whole Mechanism

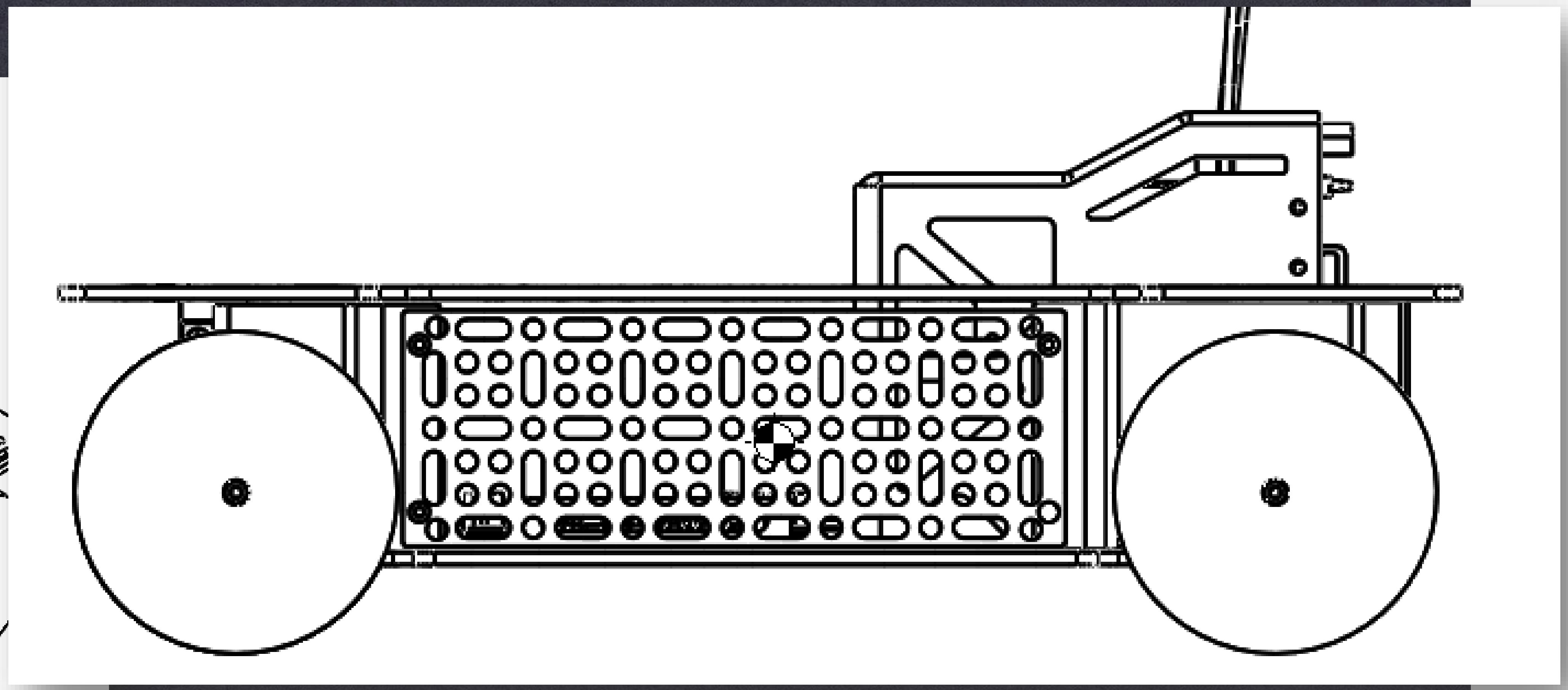


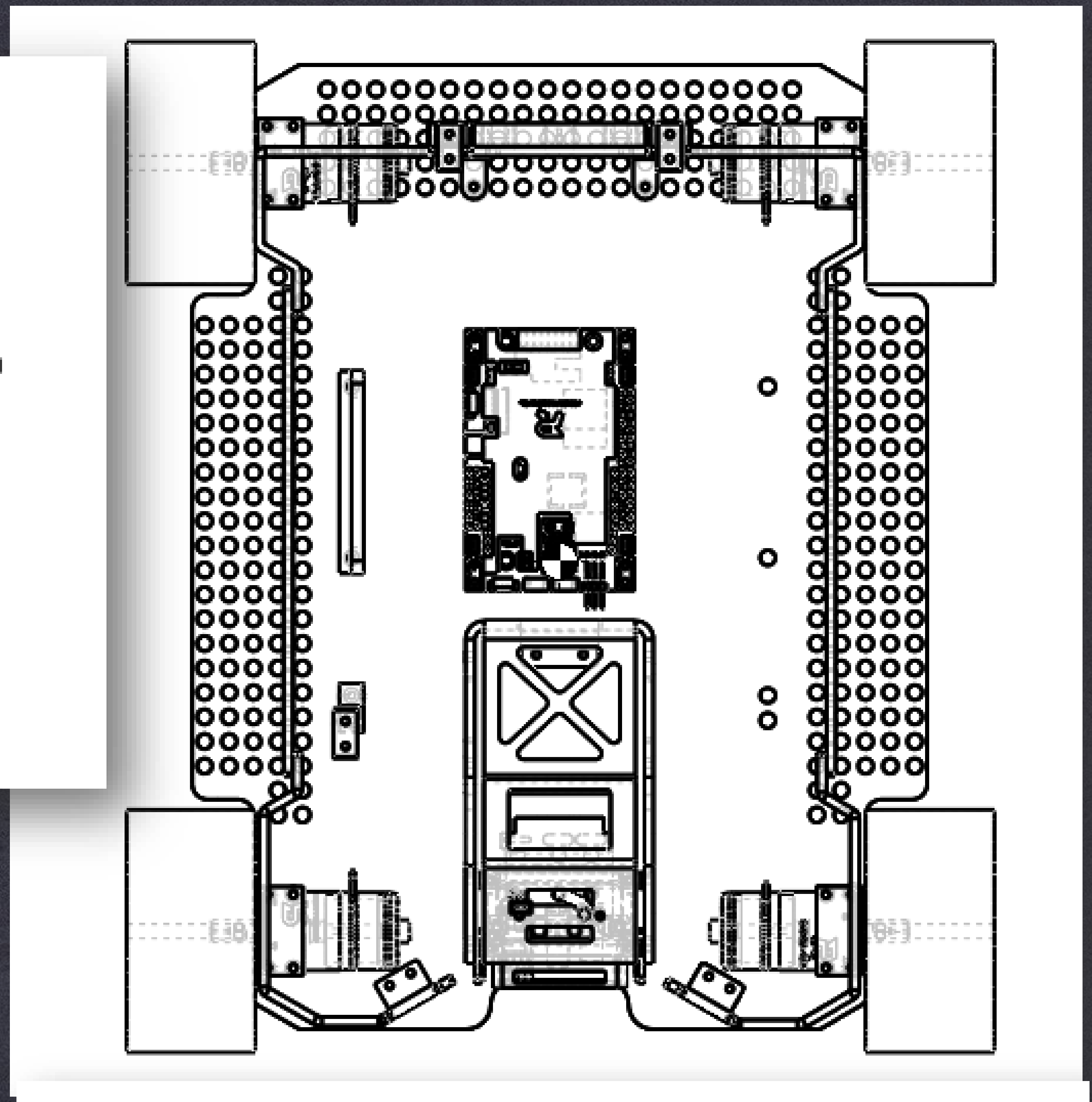








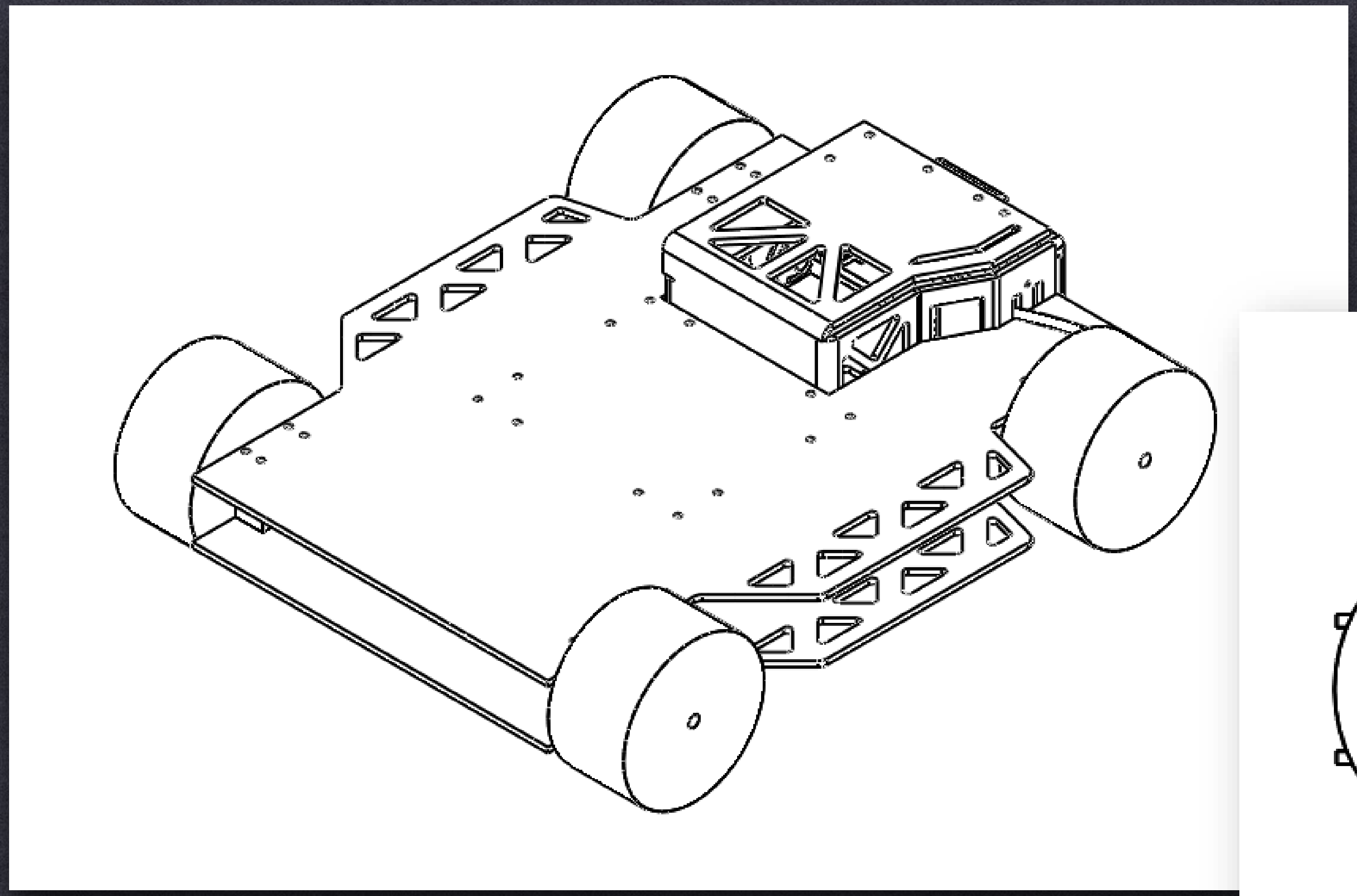


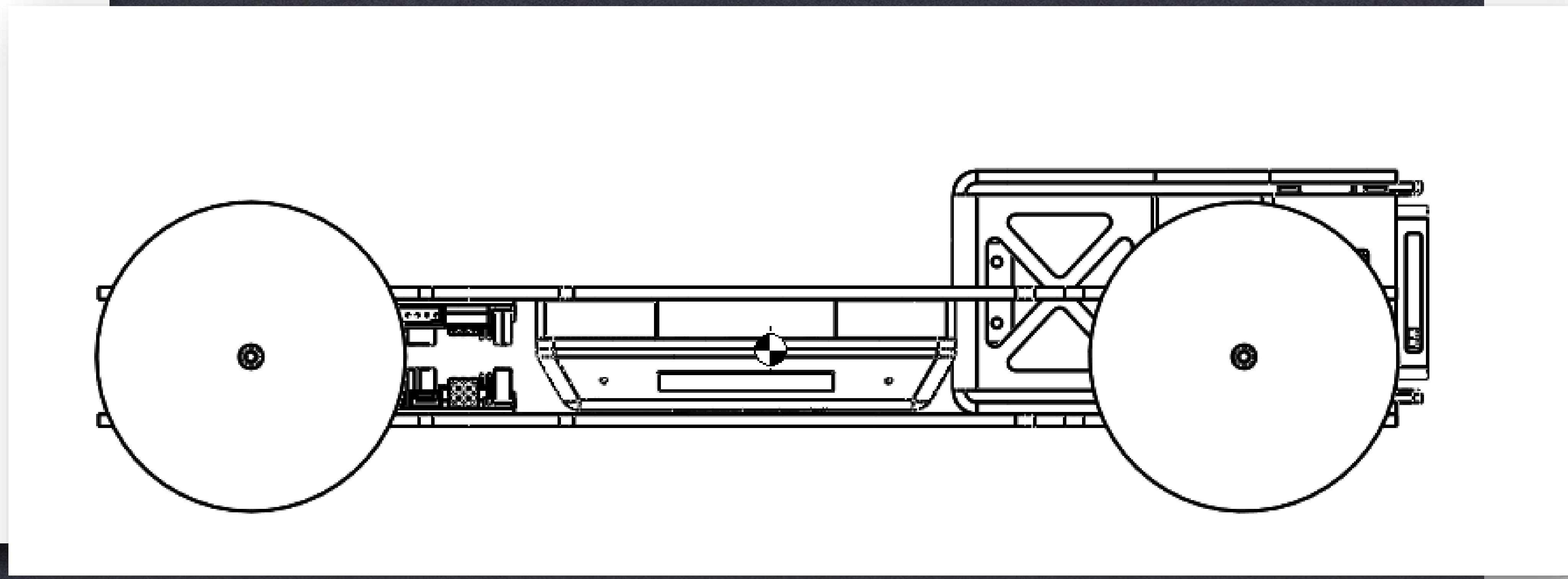


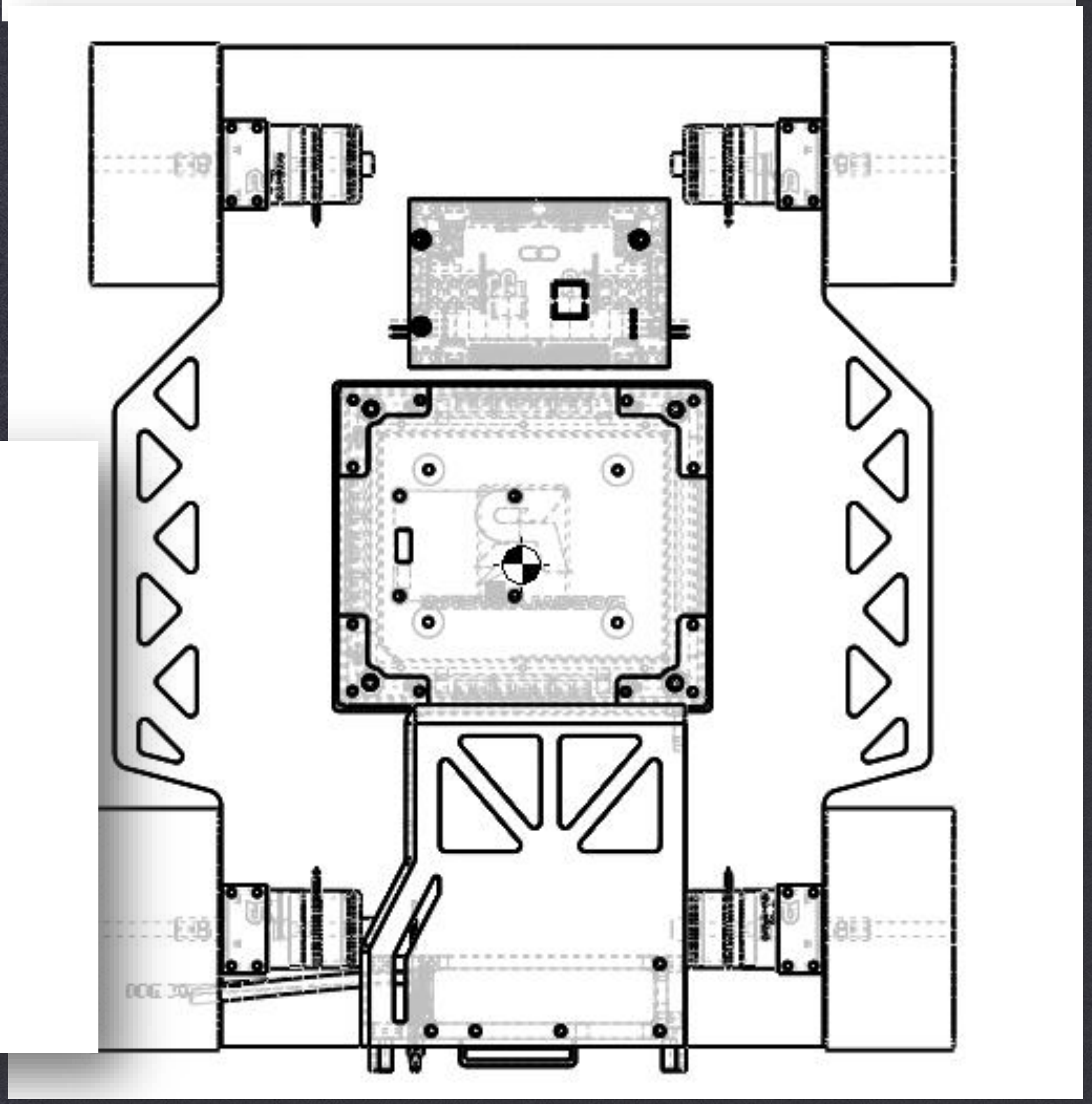
ROBOMASTERS DESIGN

vs.

PROJECT NOVA







嵌入式亮点一：机械臂控制

[Embedded]Highlight No.1: Manipulator Control

- 设计一套配合遥控器及用户体验的半自动抓取动作
  Designing a set of semi-automatic grabbing action with remote control and user-friendly experience

- 技术：组装调试机械臂，运用几何代数的三角函数以及余弦定理，构建机械臂运动的解算方程，将yz平面上的线性运动转换为舵机的角度变化，降低操作手难度.

Technology Point: Assemble and debug the manipulator, use the trigonometric function of geometric algebra and cosine theorem t construct the equation of the manipulator motion, transform the linear motion on the YZ plane into the angle change of the servers to reduce the difficulty of the driver.

- $\Delta x = L1\cos\theta 1 + L2\cos\theta 2$ （方程1）

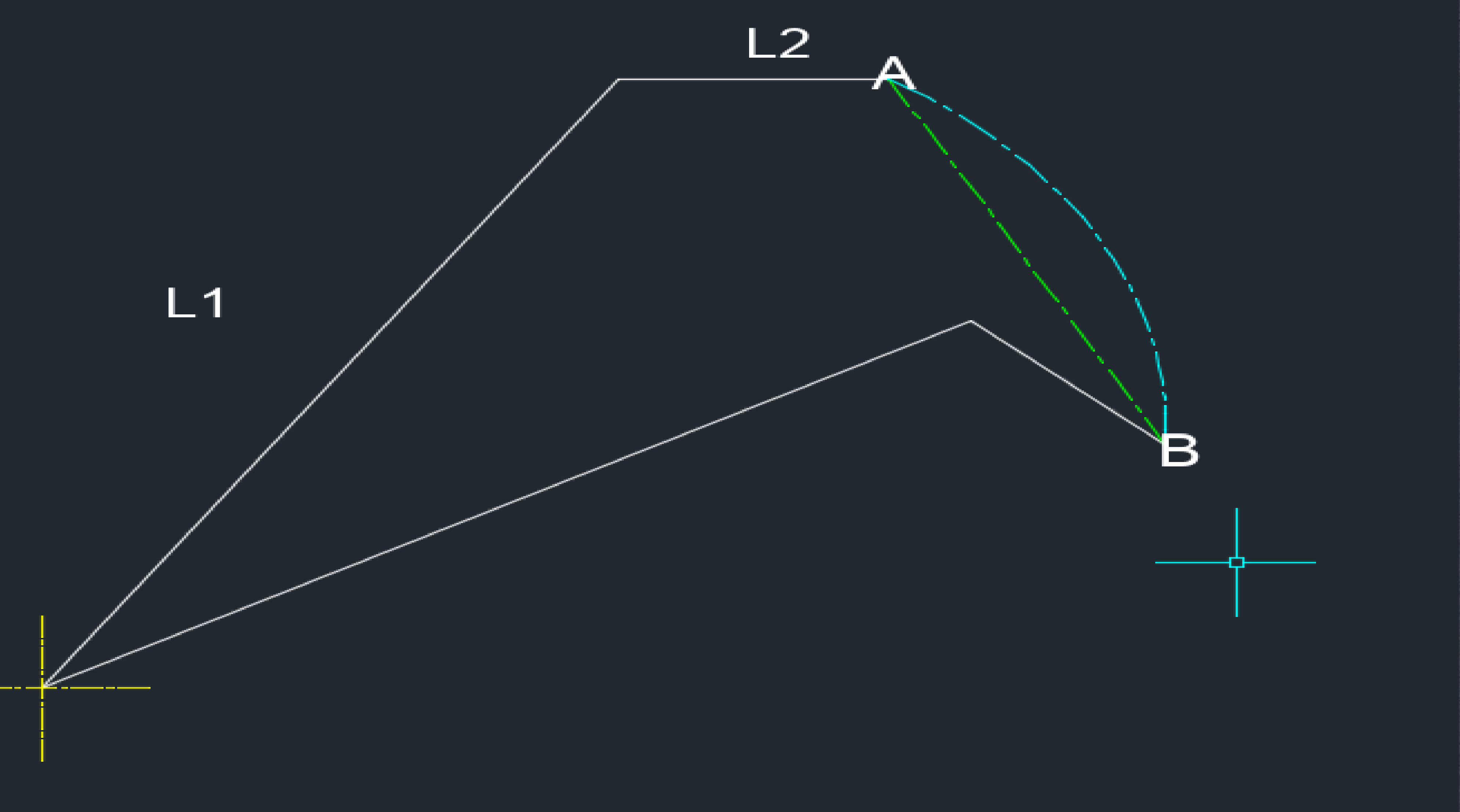
  $\Delta y = L1\sin\theta 1 + L2\sin\theta 2$

  $\theta 1 = a\cos((L1^{2}+x^{2}+y^{2}-r2^{2})/2*L1*(x^{2}+y^{2})^{0.5})$ （方程2）

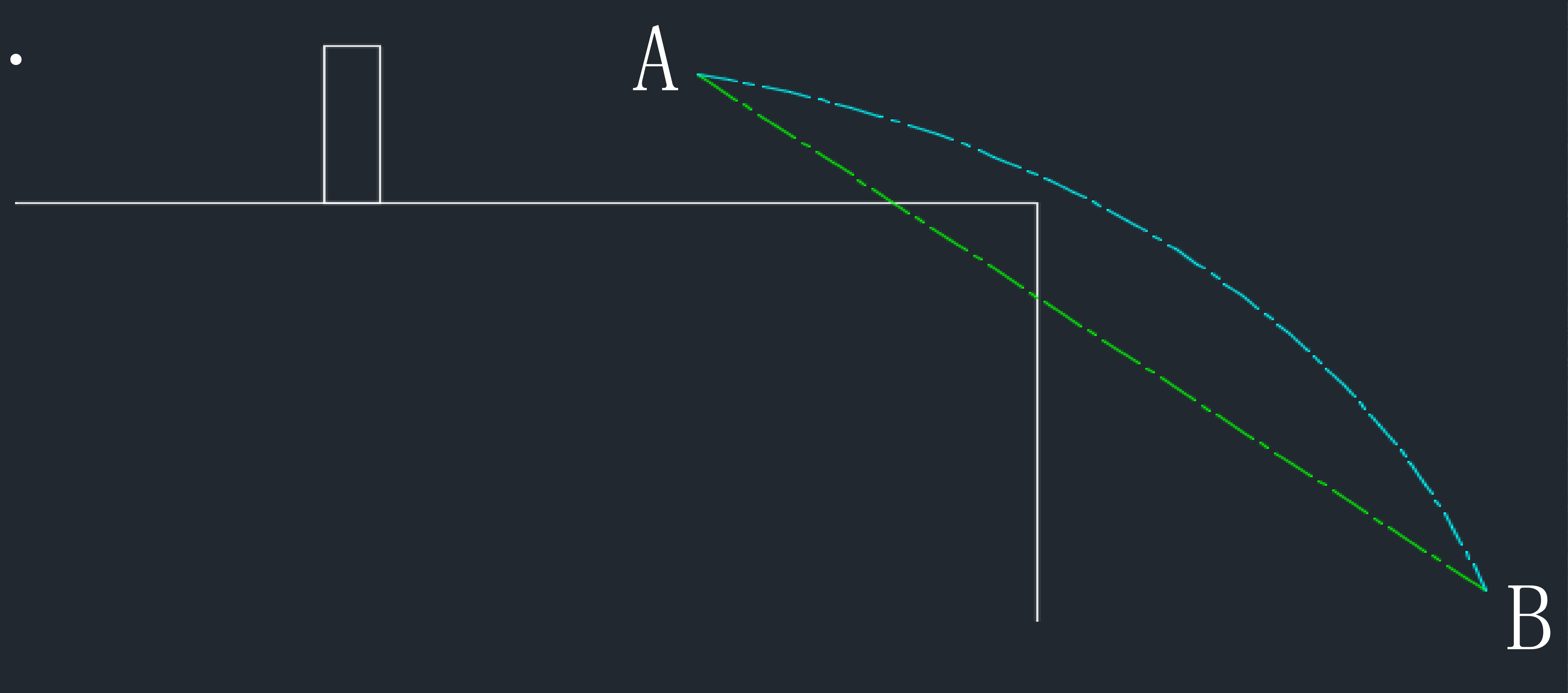
  $\theta 2 = (\Pi - a\cos(r2*r2+x*x+y*y-r1*r1/2*r2*sqrt(x*x+y*y)) + a\tan(y/x))$

机械臂运动解的限制：
1、因为结构限制，机械臂运动存在一定限制，所以要建立机械臂运动解的限制方程
因两舵机联动是复杂曲线（方程一），所以利用上述方程每次增加较小（ΔX，ΔY），化曲为直，将舵机的运动简化为直线。
现在舵机的运动干涉问题转化为平面坐标系中点与直线关系的判断，如图：（1.&2.）
计算两点间直线需要上一点坐标，所以SERVO结构体不但要存储目标位置，还要存储当前位置。
优化：假设舵机在相同时间内运动的角度相同，在曲线取大量差为ΔΘ的点，计算它们是否干涉，使运动方程更加精确。
2、上述方程因使用余弦定理，必须满足三角关系，即
(X^2+Y^2)^1/2<L1+L2
因此，平面上满足上述限制的点是可运动到的。(3.)

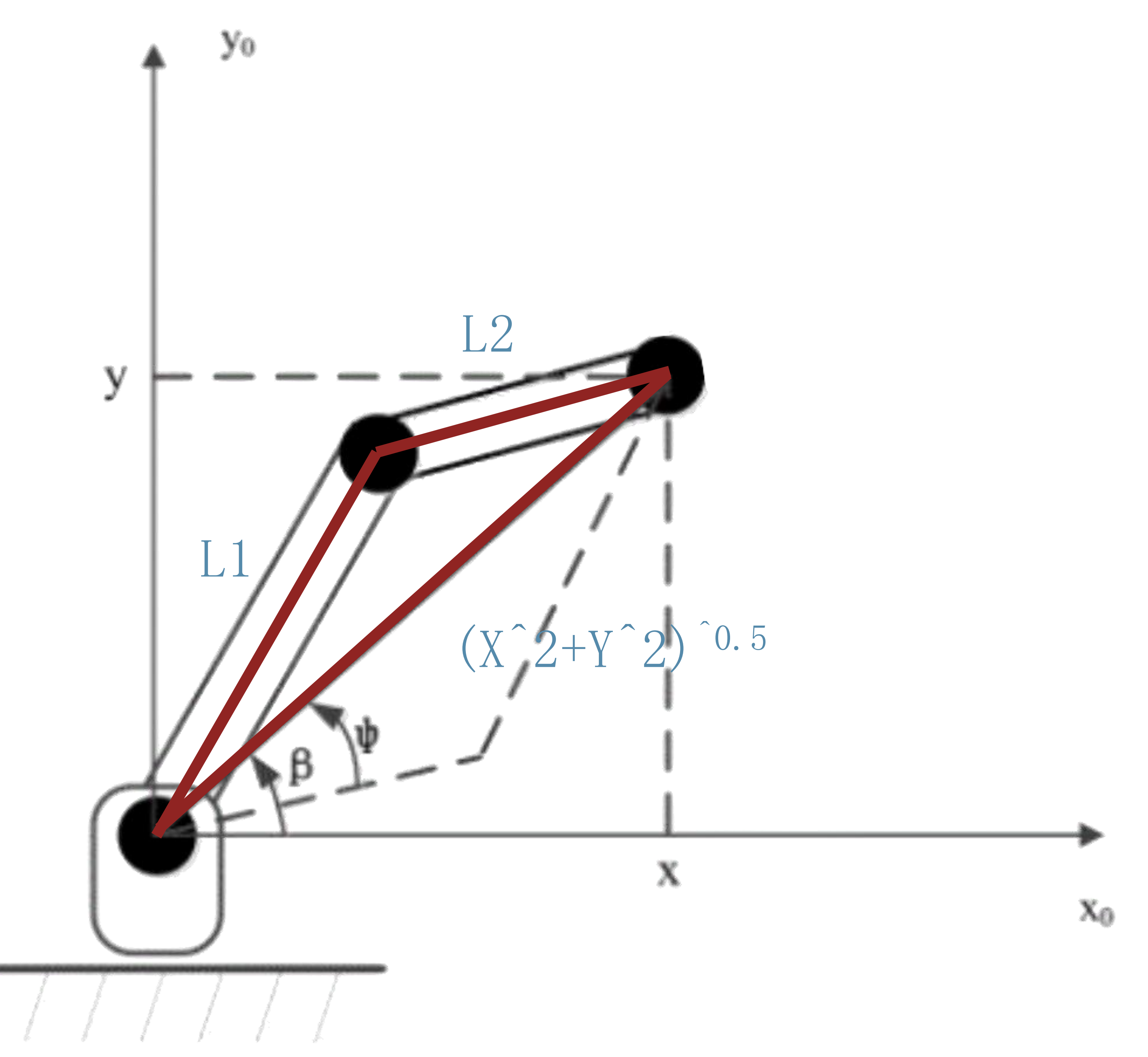
1.



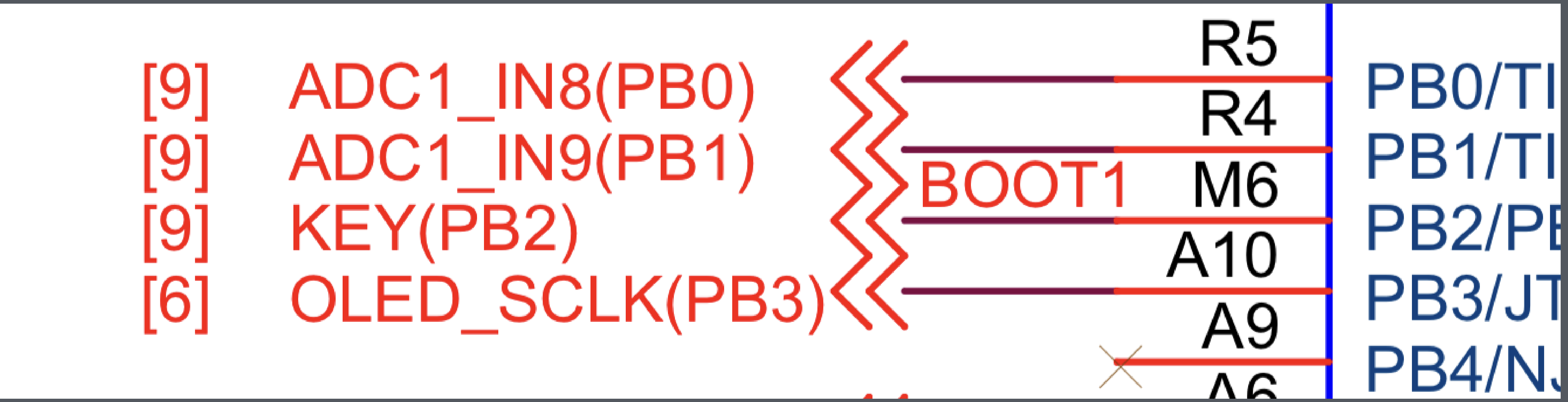
2.



3.



嵌入式亮点二：红外传感器模拟量读取[Embedded]Highlight No.2: Infrared ray Control



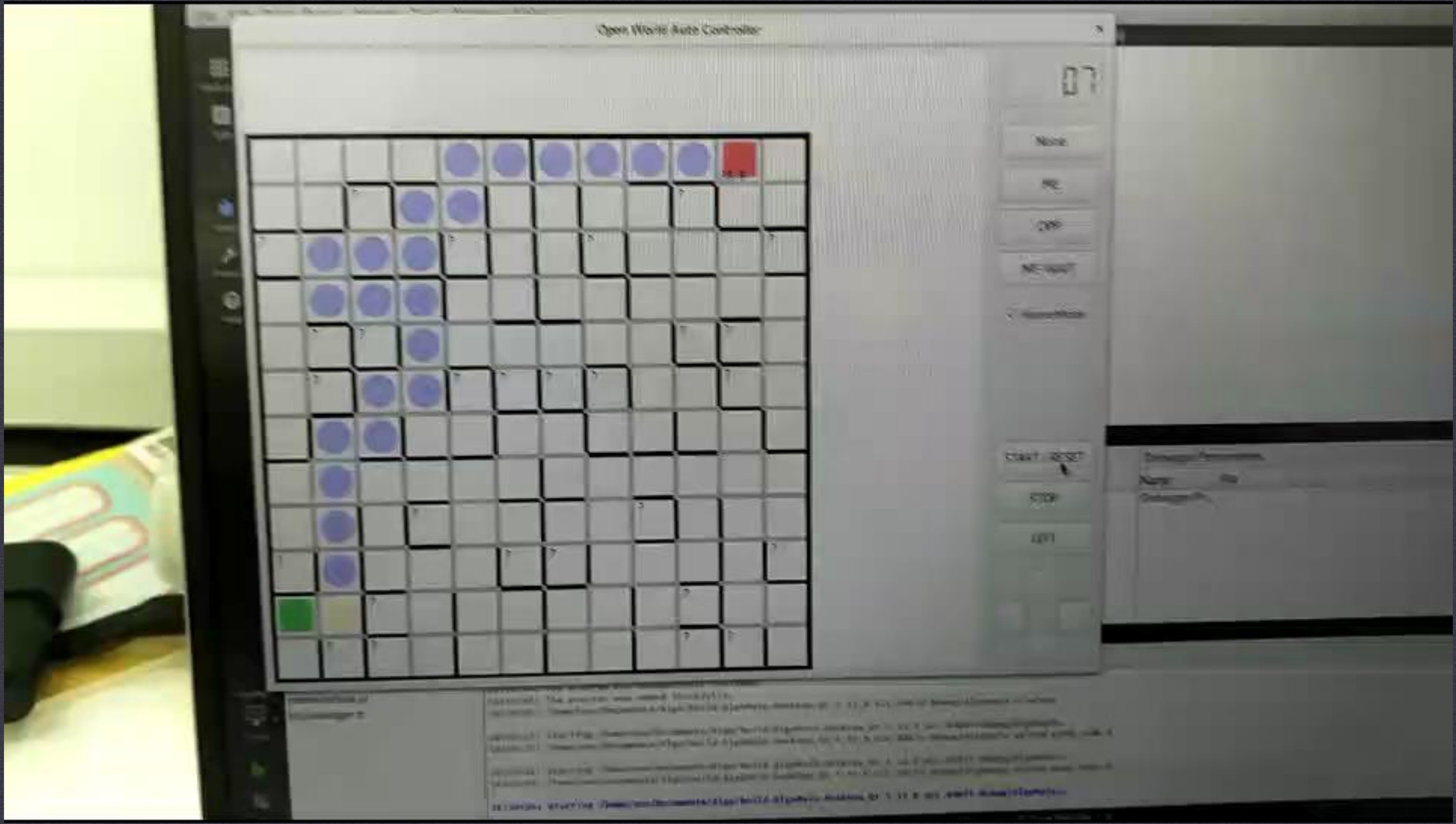
- 为提高巡线精度，尝试读取模拟量。因A型板的封装库不开放模拟量端口，通过查询数据手册，学习f427模拟量配置，读取PB0/L2口模拟量In order to improve the accuracy of line patrol, the analog data are read. Because the encapsulation Library of type A board does not open analog ports, the analog ports of PB0/L2 port can be read by inquiring the data manual, learning the configuration of f427 analog ports and reading the analog ports of PB0/L2 port.

```
void  Adc_Init(void)
{
  ADC_InitTypeDef ADC_InitStructure;
  GPIO_InitTypeDef GPIO_InitStructure;

  RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOA |RCC_APB2Periph_ADC1  , ENABLE );


  RCC_ADCCLKConfig(RCC_PCLK2_Div6);   //设置ADC分频因子6 72M/6

  //PB0 L2
  GPIO_InitStructure.GPIO_Pin = GPIO_Pin_1;
  GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AIN;
  GPIO_Init(GPIOB, &GPIO_InitStructure);

  ADC_DeInit(ADC1);  //复位ADC1,缺省

  ADC_InitStructure.ADC_Mode = ADC_Mode_Independent;
  ADC_InitStructure.ADC_ScanConvMode = DISABLE; //单通道
  ADC_InitStructure.ADC_ContinuousConvMode = DISABLE; //单次转换
  ADC_InitStructure.ADC_ExternalTrigConv = ADC_ExternalTrigConv_None; //
  ADC_InitStructure.ADC_DataAlign = ADC_DataAlign_Right;  //右对齐        别读错了orz
  ADC_InitStructure.ADC_NbrOfChannel = 8;
  ADC_Init(ADC1, &ADC_InitStructure);


  ADC_Cmd(ADC1, ENABLE);
  ADC_ResetCalibration(ADC1);
  while(ADC_GetResetCalibrationStatus(ADC1));
  ADC_StartCalibration(ADC1);
  while(ADC_GetCalibrationStatus(ADC1));
```

Autonomous Algorithm

UI

UI



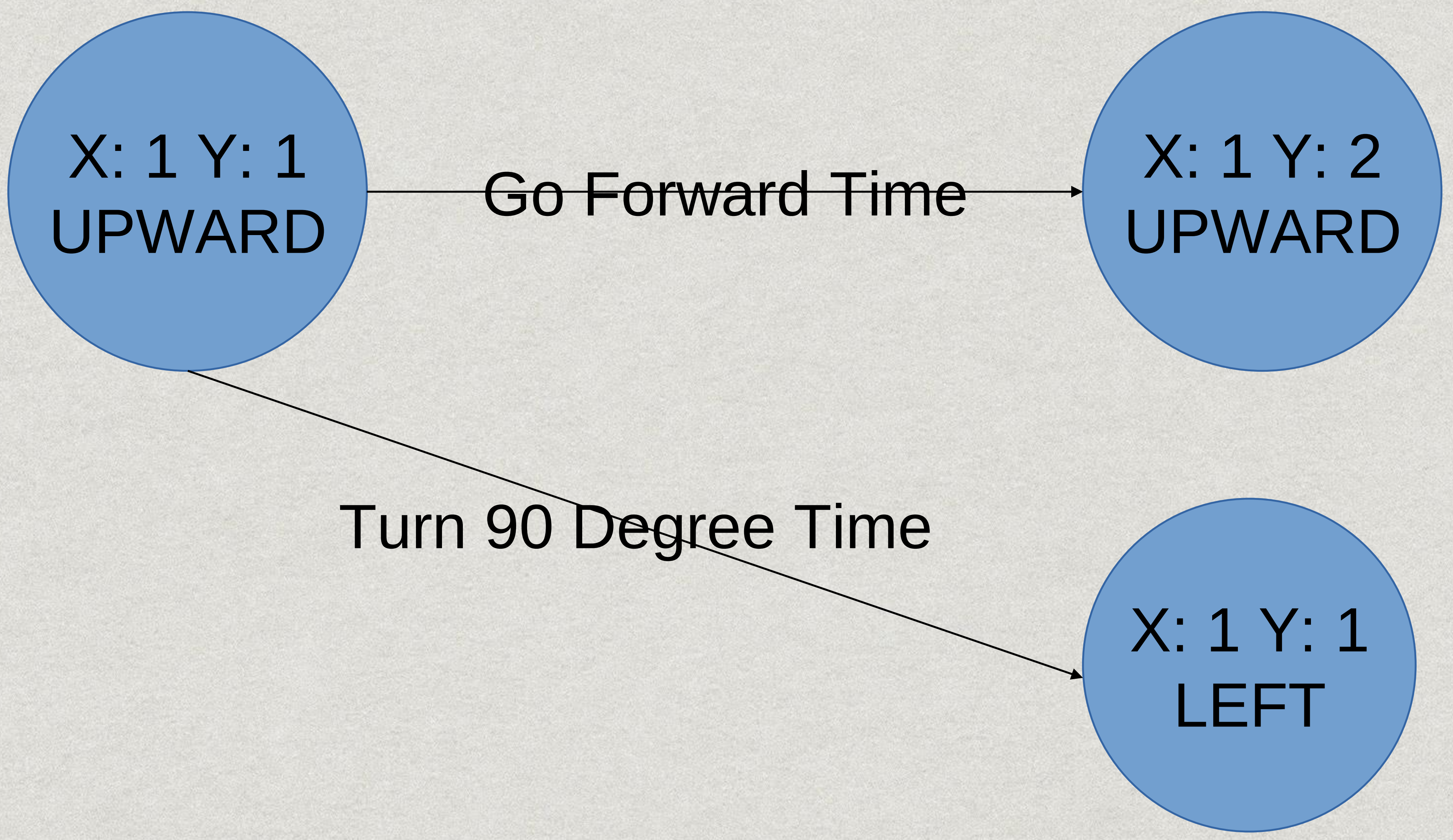
Path Planning

- Construct Graph Model(建模)

Robot Turn Time

Robot Go Forward Time

Every Node: Position and Direction (u, d)



Path Planning

- The Spinning Gate
- Map will change through time
  Consider whether wait for gate or go another path
  How to solve?

- A* Path Searching Algorithm
- Use Position and Time As State
  Convert Current Time(Continuous) To Integer(Discrete)

- time discretization
- (u, t): Current at position u, time t

- Avoid Frequent Path Changing & Going Round
- Using (u, t) = len
  Firstly Optimize time, Secondly Optimize path length

Successful Result

- Accurate maneuver in front of spinning gate

~1000 Lines Of Code

General Planning

- Shortest path to explore all hiding blocks
- Balance between exploring and catching block
- Using symmetry of map to infer block position

General Planning

- Strategy: 5 Stages
- 1 Get the key block and explore positions along the way
- 2 Go home and place the key block
- 3 Go back and catch the block found by exploring
- 4 Explore the positions that never reached
- 5 Catch all the blocks left

General Planning

- Randomized Algorithm for Exploring Path
- Simulated Annealing (SA) for first exploring path
- Pre-process the path and save it

- Searching Algorithm for Catching Path
- Optimal solution for the catching path

General Planning: The Optimal Solution

- Optimal Solution
- Modeling: Traveler Salesman Problem (TSP)
- Solution: Dynamic Programming (DP)
- Time complexity: $O(n * 2^n)$
- DP State Transfer: dp[u][s] -> dp[v][s | (1 << v)]

- $n = 26$
- Very fast, complete within 1 second
- Pre-process path length and DP optimization
- Solves the transfer cost change through time problem
- Use DP Value as Current Time

General Planning: Results

- Robot almost fully automatic running
- Operator only
- 1. identify the block using camera
- 2. catches the block

- All strategy & path planning & navigation is done by program

Algorithm: Real-Time & Stability

- Algorithms run on Real-Time 20Hz
- Frequent communication with robot
- Fast responding to changing situation
- Stable running & Robust

Totally ~2000 Lines of Code

Visual Debugging

Debugging Time ~5h

Autonomous Algorithm: Future Plan

- Time not enough to do
- Way to fully autonomous (without any human interaction)

- Two Computer Vision tasks:
- Automatic Block Identification
- Automatic Block Catching using Pose Estimation

Autonomous Control Logic

Robot Location

- Problem
- RFID Sometimes Fail To Read
- IR Sensor not accurate

- Sensor Fusion
- Detect black card for relative position
- Detect RFID for absolute poition

- Sensor data filtering
- Probalistic solution to fusion the data

Robot Location

- Trajectory & Odometry
- Motor Encoder
- Mathematical Trajectory Calculation
- Control Theory

- Move a certain distance accurately
- Starting & stopping accurately

Robot Navigation

- Gyroscope Direction Stablizing
- 16x Line Following Sensor for Horizontal Position Stablizing

- Data filtering
- PID Controller
- Paramter turning

Robot Navigation

- Navigation Control Logic
- How to go to certain position?
- Turn to specified direction and calculate distance
- Go specified distance
- Correction using robot location
- How to cancel path?
- Stop at the center of next cell using location system

Autonomous Stability

- Run very stable
- Resist to pushing & colliding
- 100% Accurate Location
- Never Collide the Wall

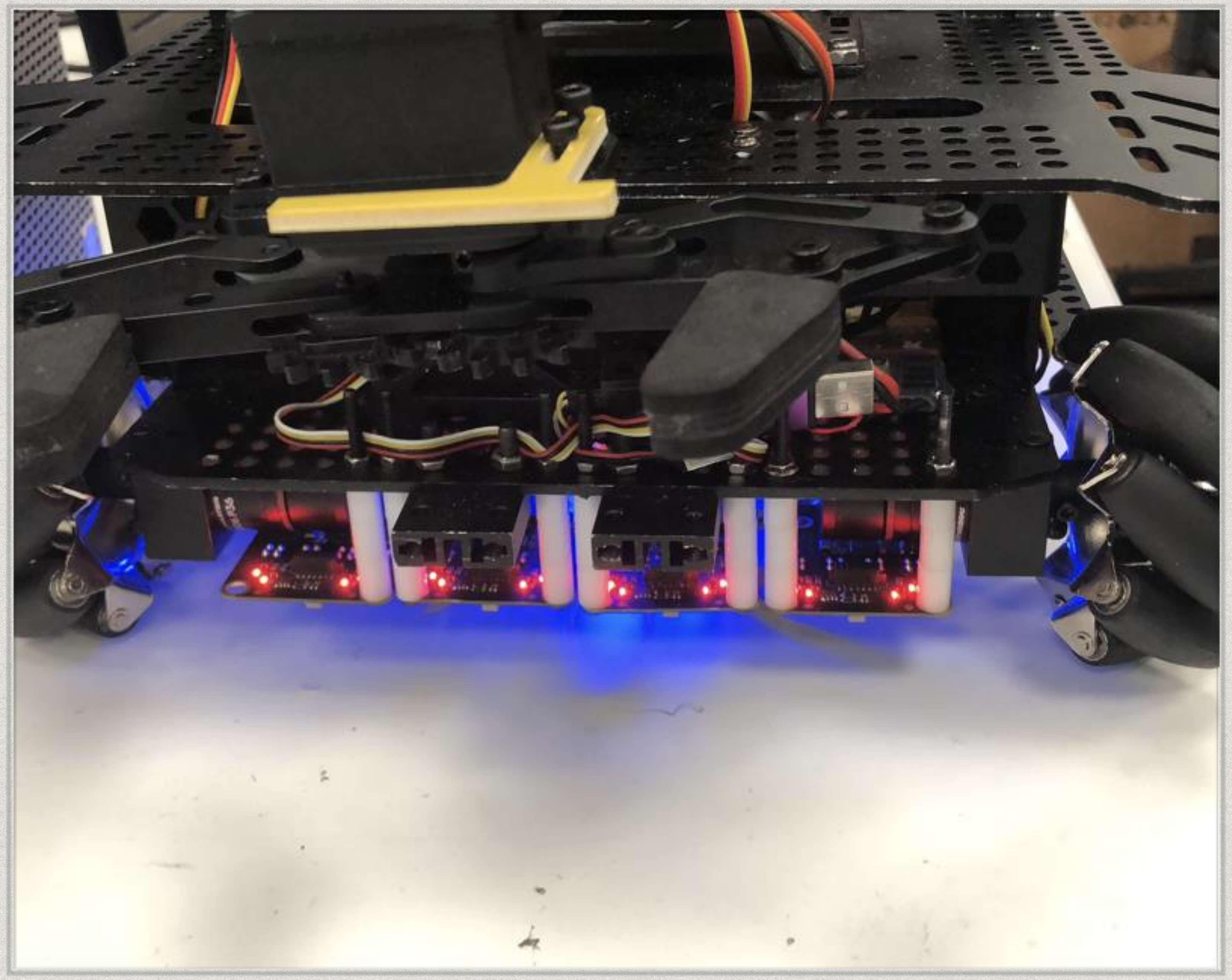
Communication

- Using PC for Algorithm Executing
  - Wireless Serial Port
  - Checksum

- Real-Time Feedback & Decision
  - Real-Time Algorithm Execution
  - 20Hz Position & Direction Feedback
  - 20Hz Navigation Target Receving

Localization

- Use infrared sensor to find black line, count the number of black line.
- 使用红外传感器寻找黑线，计算黑线的数量获得位置.
- Use RFID to get the absolute position in the same time.
- 同时使用RFID获得绝对位置.



Trajectory Generation

Use two quadratic functions and primary functions to generate the trajectory

$$t_{acc} = \frac{|e-s|}{2v} - \frac{\sqrt{a^2 \frac{|e-s|^2}{v} - 4a(e-s)}}{2|a|}$$

$$\theta_a(t) = s + \frac{a}{2}t^2$$

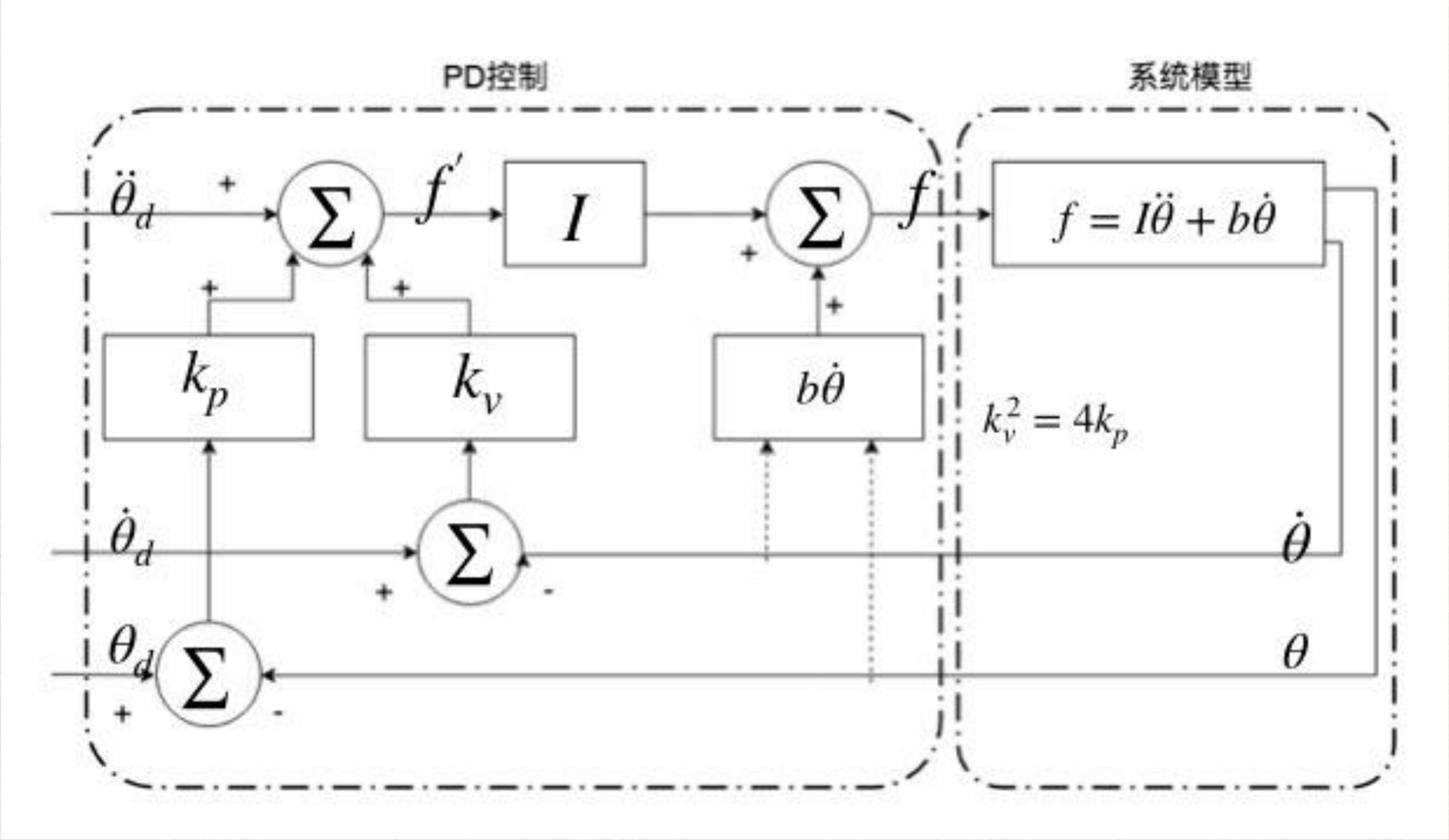
$$\theta_u(t) = at_{acc}(t - t_{acc}) + s + \frac{a^3}{2}$$

$$\theta_d(t) = e - \frac{a(e-s)^2}{v} + \frac{a^2(e-s)^2}{v}t + -\frac{t^2}{2}$$

Control

For small distance, use position
tracking controller for each whe

$$u_{wheel} = K_p(\theta_d - \theta) + Kd(\dot{\theta}_d -$$

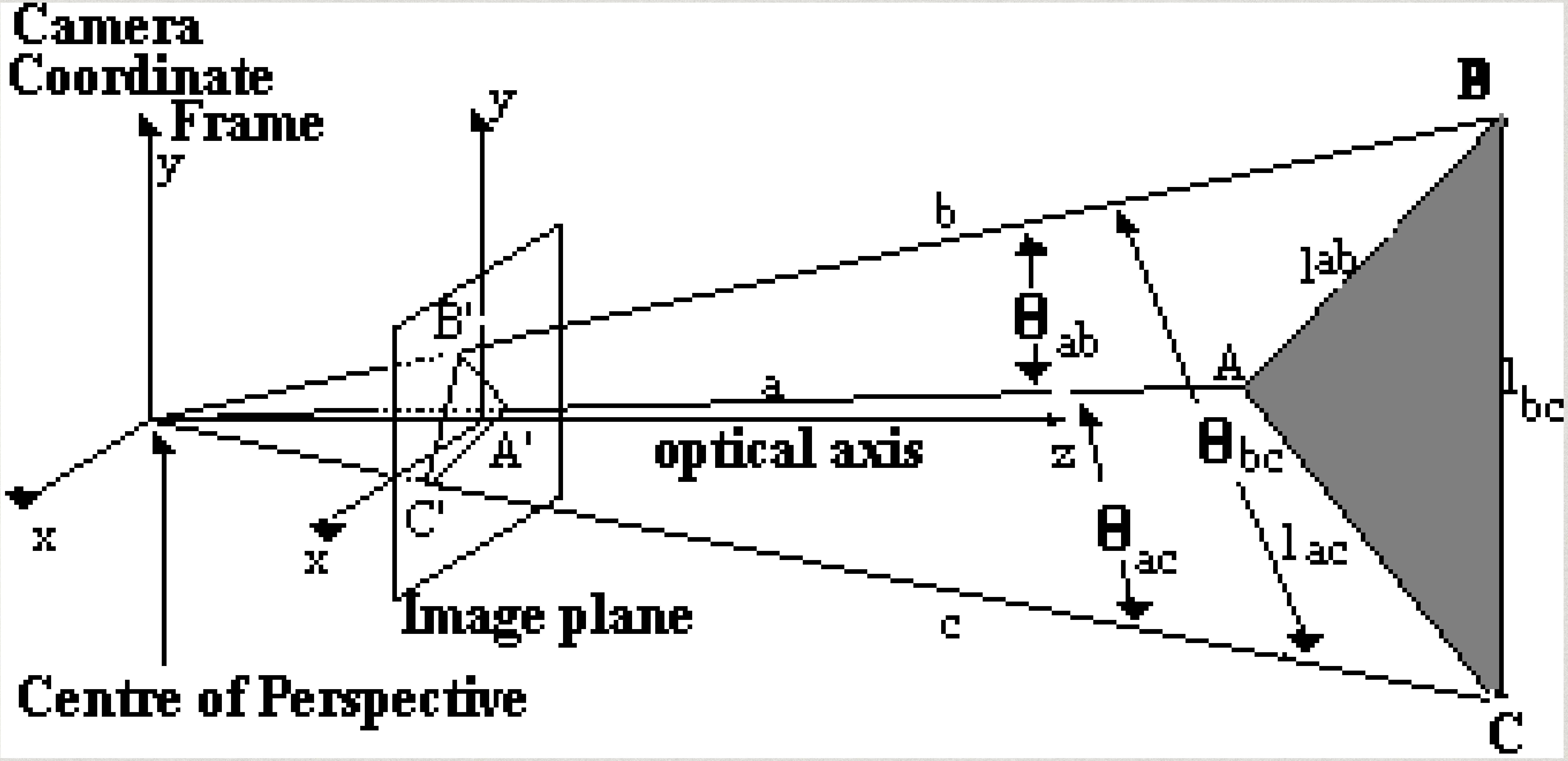


For long distance, we use speed
tracking controller for the hole car.

$$u_y = Kp(v_d - v) + k_d(\dot{v}_d - v)$$
$$u_x = K_p(x_d - x)$$

Fully Automatic(in doing)

- Automatic Localization and Manipulation
- Perspective-N-Point



团队主旨

可实践性与稳定性