

项目编号： Yz2023110

2023 年大学生创新创业项目

# 设计说明书

作品名称： 葱行者-小型智能大葱联合收获机

设计者： 刘俊杰

指导教师： 袁新梅

## 目录

一. 综述 .....	1
1.1 市场调研 .....	1
1.2 设计目的 .....	2
1.3 设计思路 .....	3
二. 设计内容 .....	3
2.1 松土模块 .....	4
2.2 夹持运输模块 .....	4
2.3 清土模块 .....	6
2.4 剪切模块 .....	7
2.5 视觉识别筛选模块 .....	10
2.6 动力及辅助转向模块 .....	16
三. 工作流程图 .....	19
四. 理论计算与仿真分析 .....	19
4.1 理论计算 .....	19
4.2 关键零件仿真分析 .....	24
五. 总结 .....	26
5.1 创新点总结 .....	26
5.2 展望 .....	27
六. 参考文献 .....	28
附录 1 产品三维图 .....	29
附录 2 视觉识别程序 .....	30

# 一种小型大葱联合收获机

## 一.综述

### 1.1 市场调研

大葱是我国的传统种植经济作物，大葱收获工作是一个繁重的工作。虽然我国大葱的种植面积、总产量以及出口总量均居世界首位，但我国大葱收获机械起步较晚，收获方式主要还是以人工收获和半机器半人工收获为主，极大的增加了作业成本，导致种植大葱经济效益降低，直接影响人们种植大葱的积极性。

市场上的大葱为了保持新鲜性，一般保留大葱根部，并剪去大葱上部多余的葱叶，如图 1.1 1



图 1.1 1 市场上贩卖的大葱

通过市场调查和查阅资料，我们发现市场上的大葱收获产品主要存在两种问题。

- (1) 机器功能较单一，只能完成部分功能。如图 1.1 2、图 1.1 3 所示，该产品搭载于拖拉机上，通过机械转动或抖动实现对大葱根部的松土，之后再通过人工进行采集收获，后期再对根部土壤进行清理，对葱叶进行剪切，从而进入市场，这样对人力需求大，且收获过程繁琐，集成化程度较低。



图 1.1 2 大葱收获机



图 1.1 3 传统大葱收获方式

- (2) 部分大葱收获机虽然实现了机械化，但是收获时容易对大葱造成伤害或者误收到腐烂、枯萎或患病的大葱，以致于整体收获质量较低，提高了后期筛选成本。如图 1.1 4、图 1.1 5 。



图 1.1 4 枯萎的大葱个体



图 1.1 5 患病的大葱个体

## 1.2 设计目的

机械收获大葱可以大大减少人工操作的时间和劳动成本，提高收获效率。一台大葱收获机器可以完成多个人的工作量，大幅提高大葱的产量。其次，人工收获大葱需要弯腰、下蹲等较为繁重的体力劳动，容易导致劳动者的腰椎、膝关节等受到损伤。机械收获可以取代人力劳动，减轻人工劳动的强度，降低

农民的劳动压力和身体损伤的风险。在收获质量方面，大葱收获机械可以实现精准的收获，避免人工操作带来的误伤、漏收等问题，确保大葱的完整性和质量。机械化收获技术可以减少劳动力成本和时间成本，从而降低生产成本。这对农民来说是一种经济效益的提升，也有助于提高农业产业的竞争力。

但目前国内市场上的大葱收获机械功能不全面，所完成的工作量较小，后期仍然需要大量人力劳动，且伤葱率高，收获的大葱质量不佳。

针对这些问题，本团队设计一种小型集成化大葱联合收获机，该收获机集松土、拔葱、清土、去叶、鉴别、装筐等功能于一体。该项目不再局限于单一功能，在拔起大葱后，有抖土机构可以去除大葱根部的土壤，有剪切机构可以对大葱进行去叶操作，还有视觉识别系统对大葱的质量好坏进行判断，若大葱个体有腐烂、枯萎、患病等，可主动舍弃该个体。功能的集成化代替了收获大葱后的人工去土、去半叶、筛选等耗时耗力的操作。夹持机构的柔性化设计可自适应大葱葱白直径，从而减少对大葱的损伤。另外该机器还设计辅助转向系统，通过手动调节离合可实现左右转向。

## 1.3 设计思路

根据市场上已有大葱收获机构的优缺点，通过比较与论证，选用科学合理的机构，在功能上突出集成化、人性化的特点。

为了达到功能集成化，大葱收获机包含以下 6 个模块：松土模块、夹持运输模块、清土模块、剪切模块、视觉识别筛选模块、动力及辅助转向模块。

为了降低产品的成本，本项目在做到多功能多模块的同时尽量减少电机数量，使一个电机能够为 2~3 个模块提供动力。松土模块、清土模块、动力及转向模块三者都在整机下方，采用同一个电机提供动力；剪切模块与夹持运输模块空间位置上距离较小，采用另外一个电机驱动；由于视觉识别筛选模块需要根据大葱质量做出筛选动作，单独采用一个舵机驱动。

由于松土模块、清土模块、动力及转向模块三者使用同一电机，剪切模块与夹持运输模块两者使用同一电机，而不同模块对转速要求不同，则需要根据不同的转速要求对电机输出转速做变速处理。

考虑葱农对大葱的加工需求不同，去叶模块应设计为高度可调的机构，甚至可以选择不对葱叶进行剪切处理。考虑到整机在田间转向的问题，同时考虑到用舵机实现差速转向成本太大，设计一种手动转向结构，降低成本。考虑到田间来回不方便的问题，在整机旁边设计一个篮筐储存架，当大葱装满篮筐后，可立即更换空篮筐，保证作业连续性。

## 二.设计内容

该项目包含 3 个电机，其中有一个舵机，两个电机。其中一个电机为松土机构、清土机构、动力机构提供动力，另外一个电机带动剪切机构与夹持运输机构运行，舵机则与摄像头联结，视觉识别的结果传输到舵机，使筛选机构运作。

## 2.1 松土模块

松土模块由底部电机通过两对锥齿轮传动，如图 2.1 1，松土转子上连接 4 个松刀片，其结构设计灵感来源于旋耕机的刀片，如图 2.1 2，本项目每个松土刀片均可拆卸，若某个磨损严重，更换刀片即可。其松土深度最深可达 25cm。

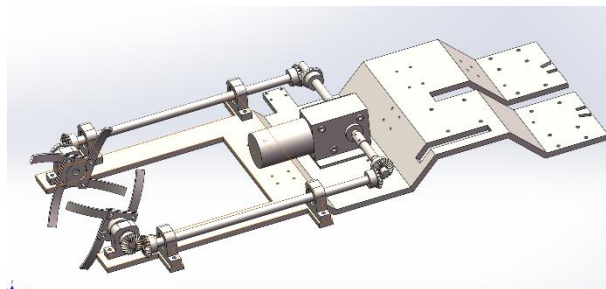


图 2.1 1 松土模块



图 2.1 2 旋耕机

## 2.2 夹持运输模块

夹持运输模块如图 2.2 1。该模块由左电机通过带传动带动左皮带顺时针转动，接着通过两对锥齿轮啮合改变旋转方向，使右皮带逆时针转动，两皮带沿相反方向旋转，即可夹持大葱向上运输，以便完成后续的清理土壤与剪切葱叶等操作。该模块中左右两边分别有 1 个主动带轮、1 个从动带轮、10 个惰轮来支撑皮带。左侧 10 个惰轮分别安装在左机架的刚性轴上，右侧 10 个惰轮则分别安装在 10 个滑块轴上。

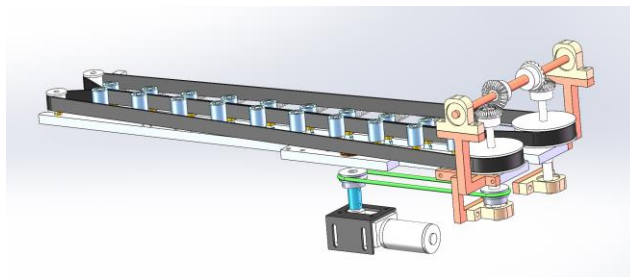


图 2.2 1 夹持运输模块

该模块创新点在于其通过拉簧、杠杆与滑块的结合使两皮带之间的间隙可随着一颗颗大葱直径的变化而改变，如图 2.2 2，从而使大葱在被夹持的过程

中葱白不会被挤压致损伤，可大大减少收获大葱时的伤葱率，提高大葱的美观程度与质量品质。拉簧、杠杆、滑块三者细节图如图 2.2 3。

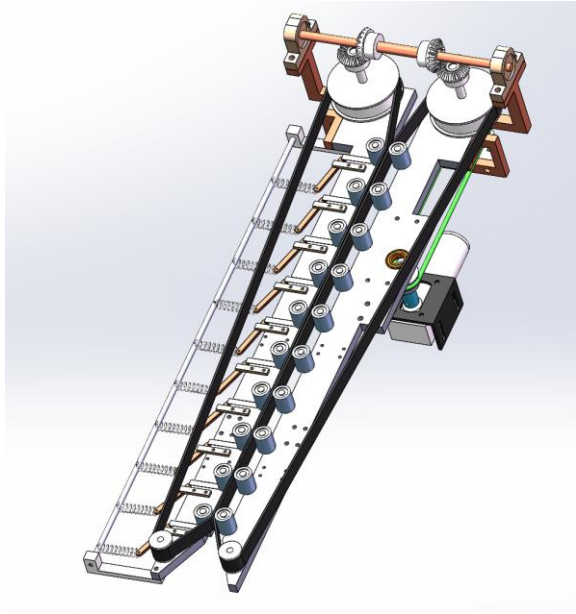


图 2.2 2 柔性夹持结构

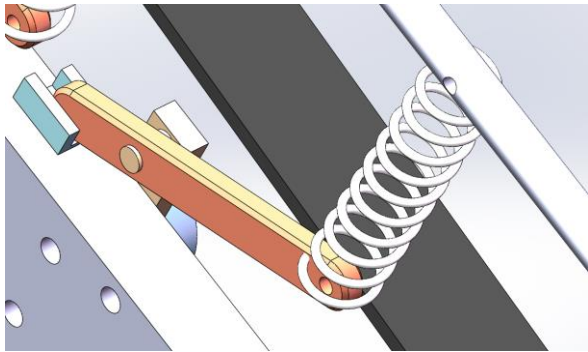


图 2.2 3 拉簧滑块细节图

下图为夹持带间隙变化前后的对比图，如图 2.2 4、图 2.2 5。

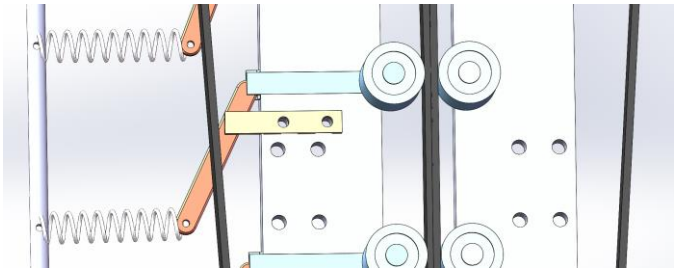


图 2.2 4 缝隙无大葱时

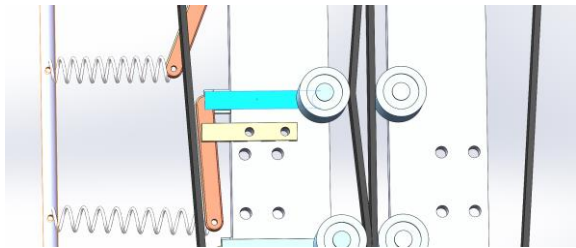


图 2.2 5 缝隙有大葱时



## 2.3 清土模块

清土模块如图 2.3 1。该模块动力来源于底部电机。电机带动曲柄轴（如图）转动，曲柄通过槽口与一滑块连接，使滑块（如图）上下移动，而滑块上下移动则会带动抖土杆（图中黄色件）绕支架左右摆动，做出抖土动作，实现清理刚刚拔起运输至两杆之间的大葱根部土壤的工作。

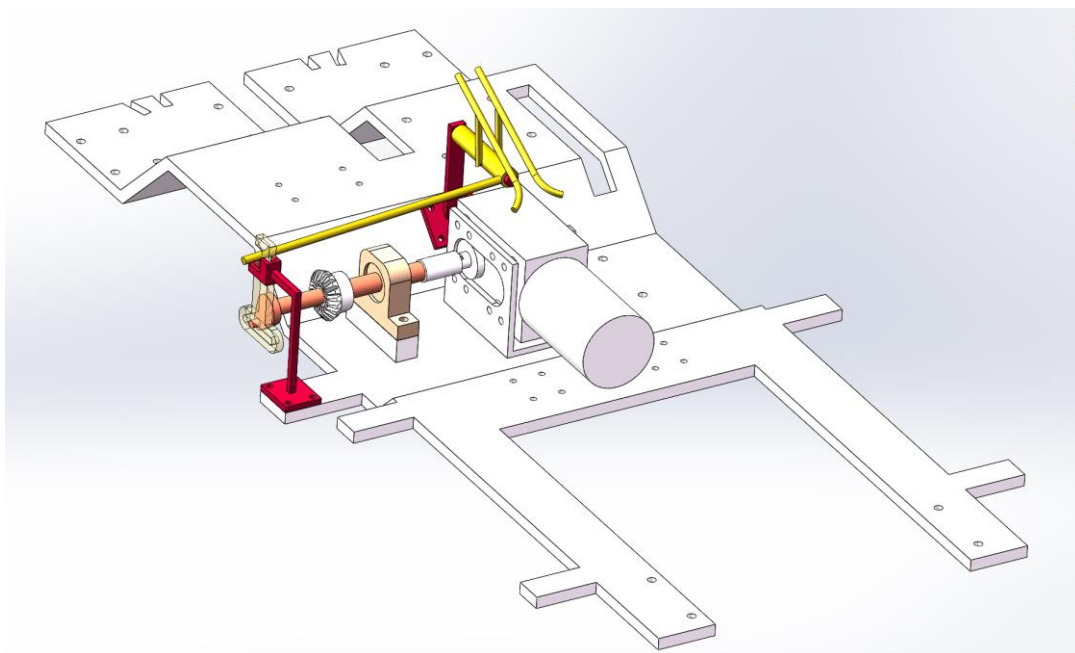


图 2.3 1 清土模块

曲柄轴、滑块、抖土杆如图 2.3 2、图 2.3 3、图 2.3 4。

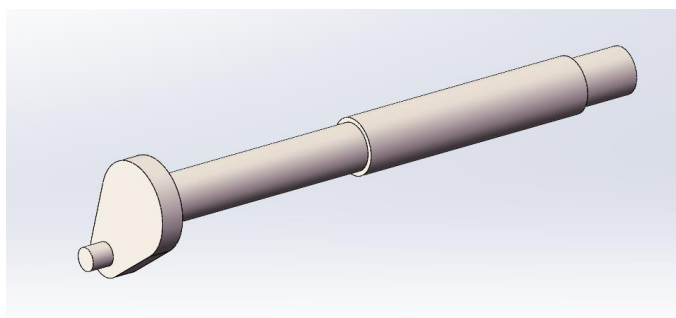


图 2.3 2 曲柄轴

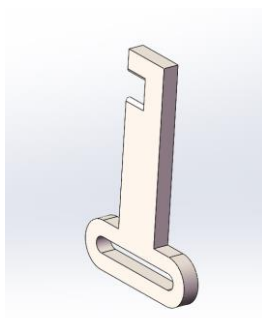


图 2.3 3 槽口滑块



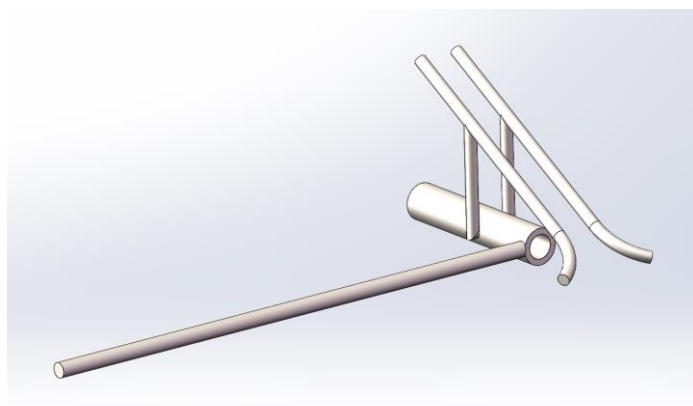


图 2.3 4 抖土杆

滑块形状如同一把钥匙，其上面的缺口是用来卡住抖土部件的长杆，滑块上下滑动，从而使抖土部件往复摆动。其摆动角度范围  $20^{\circ}$ 。

抖土杆前端外张，便于大葱根部进入两杆之间。两杆长度分别为 30cm，及在此 30cm 范围内，均会持续清土。

以上三者配合关系如图 2.3 5。

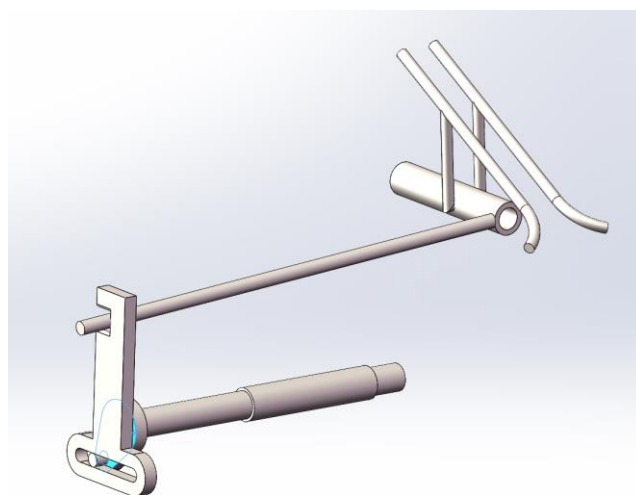


图 2.3 5 曲柄轴、滑块、抖土杆配合关系

## 2.4 剪切模块

剪切模块如图 2.4 1。剪切机构由左电机带动一对直齿轮传动，通过曲柄带动刀具左右往复移动。刀具的刀齿间距 15mm，更利于剪切大片葱叶。为了让剪切工作顺利进行，设计两块挡板对葱叶约束集中，每块挡板由两个支架固定。

为了适应不同收获需求，该剪切模块在剪切高度上设计成可调机构，高度调节方式仿照自行车座椅的高度调节锁紧装置（如图 2.4 2），通过螺栓锁紧的方式调节空心管的夹紧程度，手动放置刀架高度（调节前后对比如图 2.4 4、图 2.4 5）。若要完全保留葱叶，还可将该刀具取下，如图 2.4 6。两块挡板入口处成  $40^{\circ}$  外张，便于葱叶进入两块挡板之间。

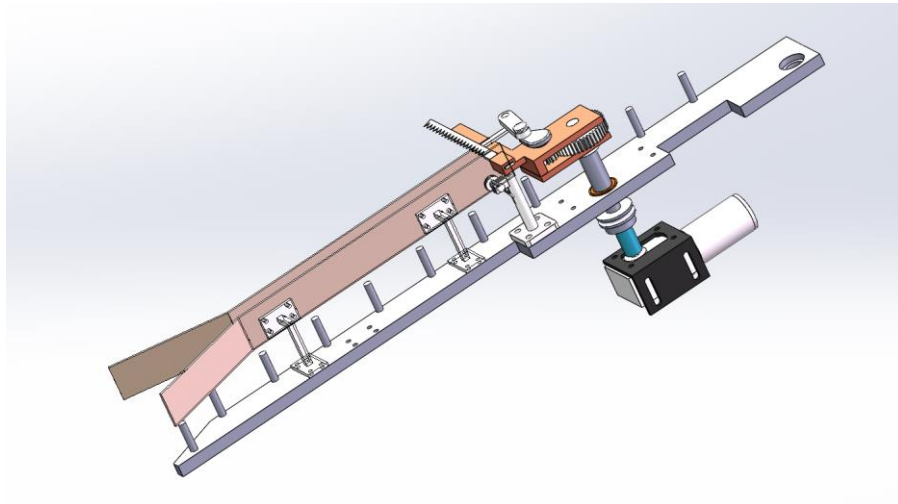


图 2.4 1 剪切模块



图 2.4 2 自行车座椅高度调节

由于剪切高度可调，同时确保轴的动力传输，该处设计方轴与方孔，使该轴能够进行轴向移动，同时又能随主动轴旋转。方轴与方孔配合如图 2.4 3。

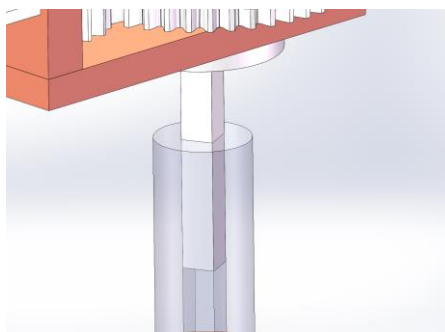


图 2.4 3 方孔方轴配合

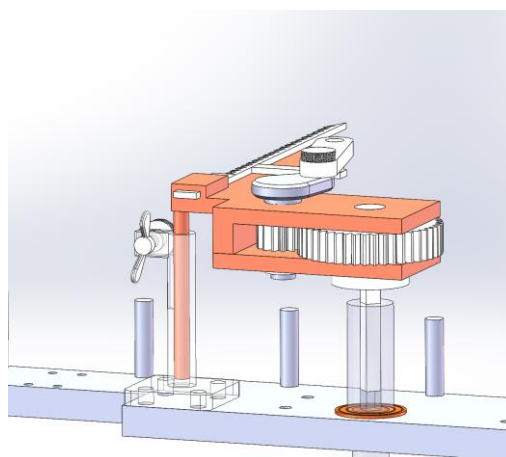


图 2.4 4 剪切高度调节前

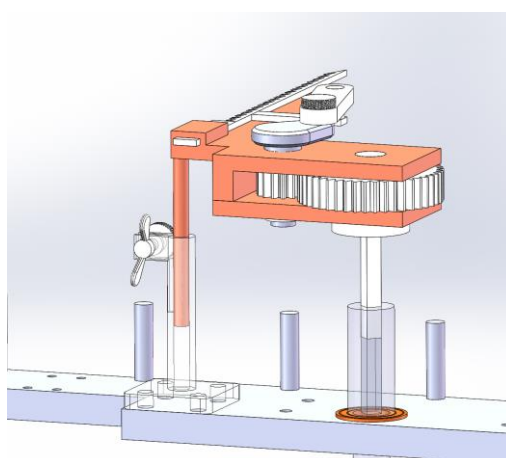


图 2.4 5 剪切高度调节后

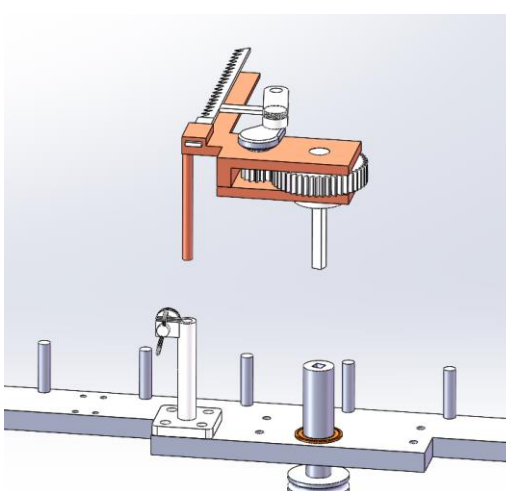


图 2.4 6 取下剪切机构

齿轮、曲柄、连杆、滑块组成的刀具装配如图 2.4 7。由于该模块电机控制的夹持运输带转速不宜过大，但剪切刀具的往复速度也不宜过小，因此通过大小齿轮啮合既能满足皮带的慢转速，又能满足刀具的快速往复

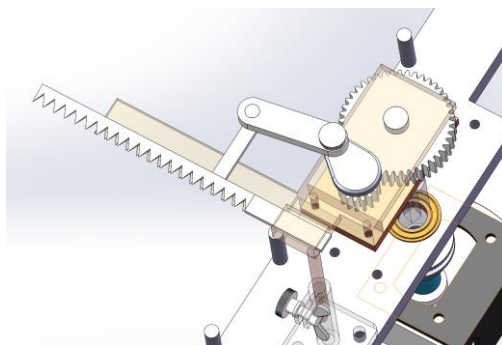


图 2.4 7 曲柄、滑块、刀片装配关系

## 2.5 视觉识别筛选模块

### 2.5.1 机械结构层次

视觉识别筛选模块如图 2.5.1 1。该模块包含摄像头、舵机、篮筐、移动平台四个部分组成。摄像头对大葱进行视觉识别，当识别到大葱葱叶上有病斑、枯萎、腐烂特征时，将信号传输至后方舵机，舵机通过连杆将篮筐从接葱位移开，使质量较低的大葱不进入篮筐，达到筛选良品的目的。

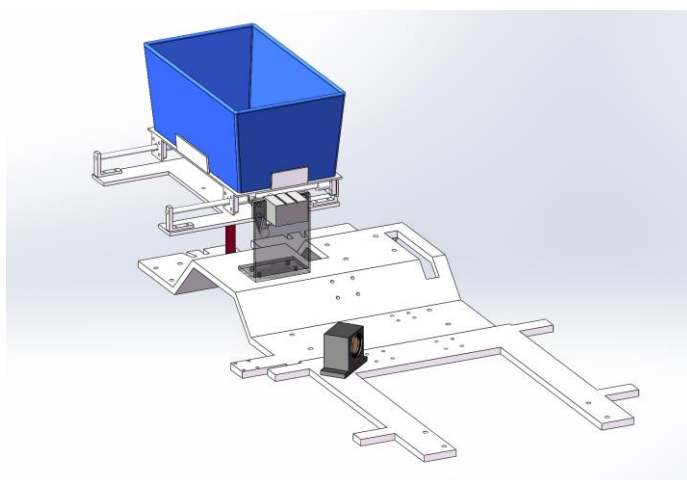


图 2.5.1 1 视觉识别筛选模块

篮筐搭载在移动平台上（如图 2.5.1 2），该移动平台包含一个篮筐架（用于放置篮筐，如图 2.5.1 3）、四个直线轴承、两根光轴、一个平台本体。篮筐架的 4 个角分别与 4 个直线轴承螺栓连接，直线轴承两两装配在光轴上，光轴通过支架固定在平台上。

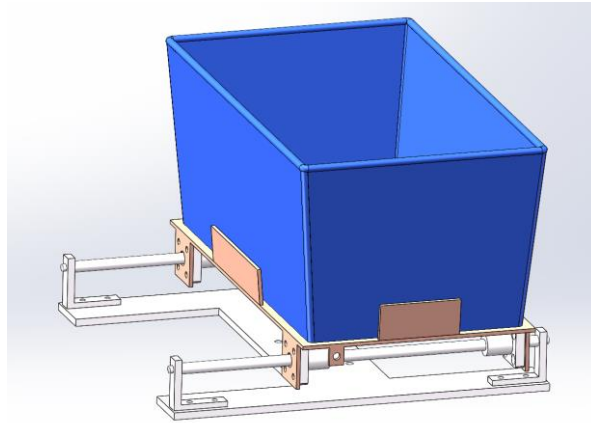


图 2.5.1 2 移动平台上的篮筐

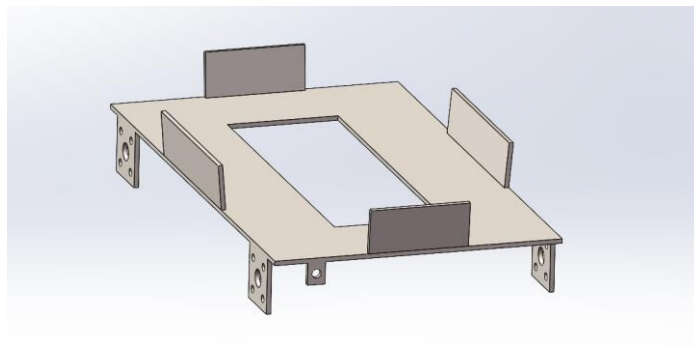


图 2.5.1 3 篮筐架

舵机驱动平台移动采用曲柄滑块结构，如图 2.5.1 4。选用直线轴承目的是减小滑块移动时的摩擦力，从而使驱动扭矩无需太大。

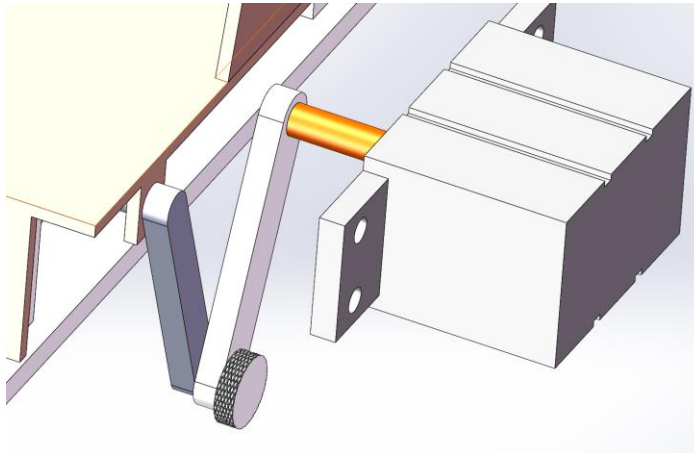


图 2.5.1 4 舵机与移动平台的连接

当篮筐大葱装满后，可卸掉该装满大葱的篮筐，从旁边的篮筐架上取空篮筐，如图 2.5.1 5，收获机左侧悬挂多个空篮筐，便于连续收获作业。

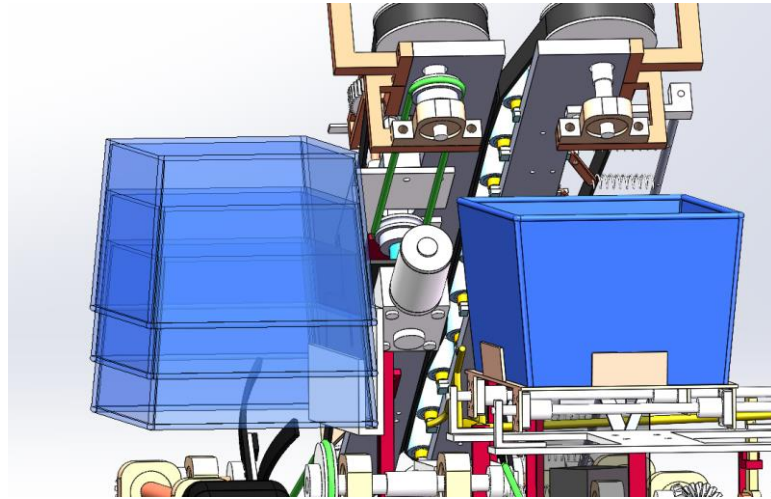


图 2.5.1 5 左侧挂有空篮筐

## 2.5.2 视觉物联网层次

智能视觉识别系统由两大部分组成：分别为理论与实物部分和功能与演示部分。

### （1）理论与实物部分

鉴于促进生活的便捷以及对大葱分类体现更大的性价比，本大葱收割农机基于深度学习的目标检测理论，采用拓展 STM32 控制板与 Orange Pi Zero3 协同控制。其中，以 Orange Pi Zero3 作为上位机来进行决策，STM32 作为下位机来控制农机执行指令。

Orange Pi Zero3 的大葱目标检测是该智能控制系统的关键部分之一，它首先利用 USB 可连接多广度摄像头作为农机识别大葱的眼睛，然后根据不同需求、不同类型大葱进行视觉分类，接着在根据需求的前提下进行决策，通过一些特定的约束条件与通信协议，向下位机 STM32 发送不同类型的 PWM 波信号，STM32 接收到来自 Orange Pi Zero3 的 PWM 波之后，会根据任务需求来控制农机进行相应的操作，当关于病斑大葱的 PWM 波传来之后，STM32 对波进行分解、解释操作，随后控制农机的机械臂将病斑类大葱推送丢弃掉，当关于高品质大葱的 PWM 波传来之后，STM32 会重复以上操作，不同的是会将高品质大葱放入正常大葱存储盒。智能视觉识别系统主要分为视觉决策系统和指令执行系统，视觉决策系统是对观察到的大葱进行分类，以分辨出高品质大葱与病斑大葱，指令执行系统则是根据视觉决策，对已分类完的大葱进行实践操作，将高品质大葱、病斑大葱放入相应的存储盒中（流程图 2.5.2 1）。



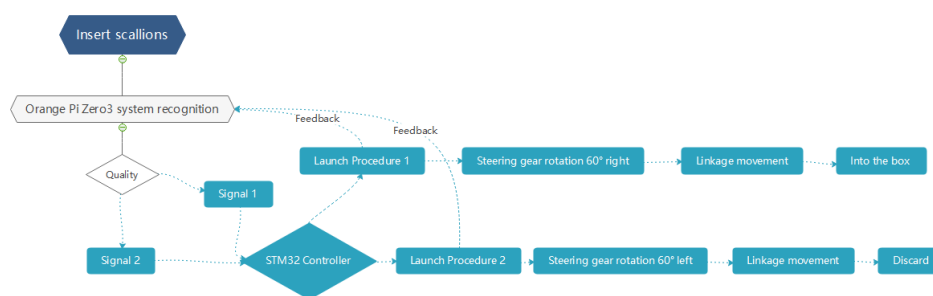


图 2.5.2 1 智能视觉识别系统流程图

大葱农机所使用到的 USB 信号传输装置和 USB RGB 相机以及 42 步进电机控制模块。其中 USB 信号传输装置根据是 Orange Pi Zero3 和 STM32 之间的传输通信协议选定，Orange Pi Zero3 作为传入设备，将信号送到 USB 传输线中，而 STM32 则作为接收设备，将 USB 传输线中的信号转化为具体的操作。

RGB USB 摄像头是一个多功能的传感系统，除了基本的回传视频画面外还可以用来收集不同的信号，如视觉色块识别，线条探测，小球识别，分类物体等。在该大葱智能视觉识别系统中，USB 摄像头可以通过视觉色块识别，通过事先的反复实验，来对不同类型的大葱进行一个准确的物体分类，在摄像头分类物体模块达到相应条件后，就会触发信号通信模块的跳转，从而对下位机 STM32 发送 PWM 波，控制机械臂进行具体的大葱分类实操。（Orange Pi Zero3 安装图和 STM32 结构图如下图所示）



图 2.5.2 2 Orange Pi Zero 安装图



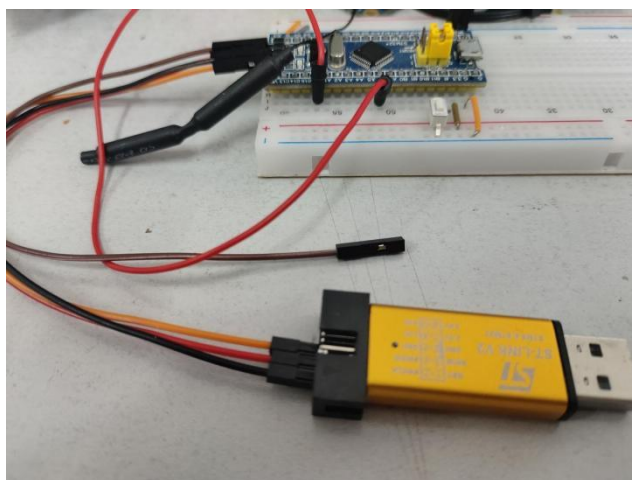


图 2.5.2 3 STM32 结构图

42 步进电机控制模块是由下位机 STM32 模块进行控制机械臂转动的一个装置，当 STM32 接收到来自上位机 Orange Pi Zero3 传来的判断信号之后，STM32 模块会做出反应，控制 42 步进电机进行旋转，之后控制机械臂完成大葱分类装拣操作。42 步进电机模块是以控制电流方向、大小来控制旋转方向以及转速。本农机 42 步进电机型号为 42BYGH39-401AS，其转矩有  $0.4\text{N} \cdot \text{m}$ ，步距角为  $1.8^\circ$ ，可供电流为  $1.2\text{A}$ 。（42 步进电机图如下）



图 2.5.2 4 42 步进电机图

## （2）功能与演示部分

在智能视觉识别系统中，功能与演示部分是相应理论的反馈，也是确认理论部分是否正确的基准，因此该功能与演示部分作为一个结果来进一步确认理论的正确性，同时，也以直观可视化的角度来展现智能大葱收割农机的智能性和其对大葱分类的准确性。（下图为对大葱识别的展示）

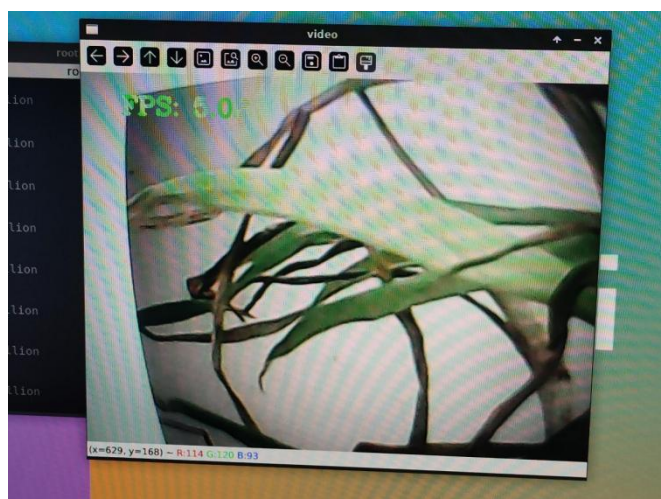


图 2.5.2 5 未识别的正常大葱图

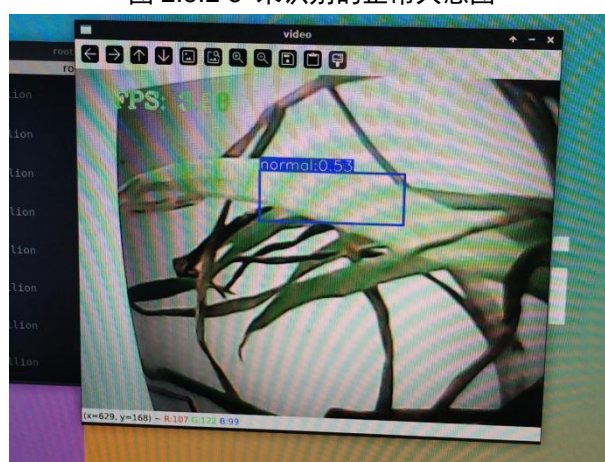


图 2.5.2 5 经识别正常的大葱图

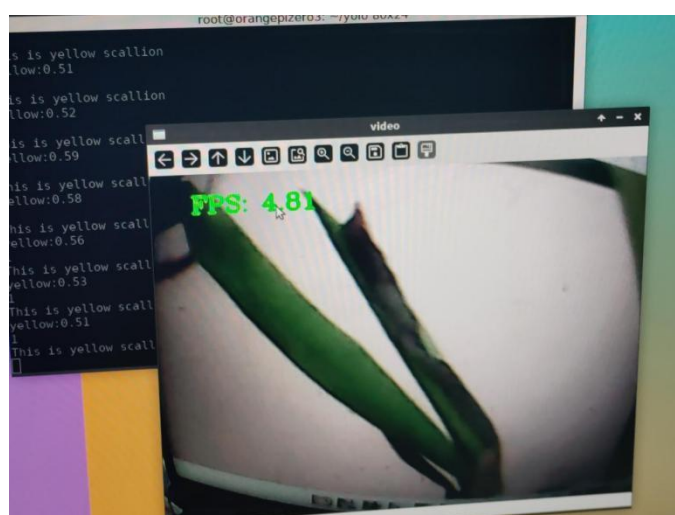


图 2.5.2 6 未识别的病斑大葱图

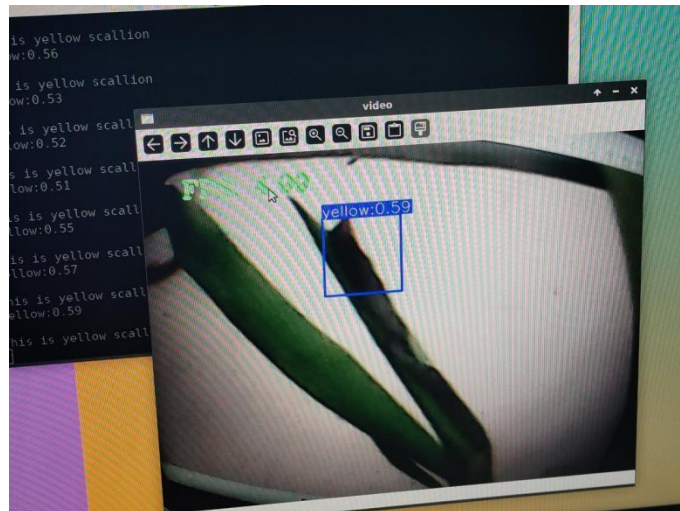


图 2.5.2 7 经识别带有病斑的大葱病斑图

## 2.6 动力及辅助转向模块

动力及辅助转向模块如图 2.6 1。该模块动力由底部电机提供，动力经两级带轮减速后传至后方动力轴，如图 2.6 2。

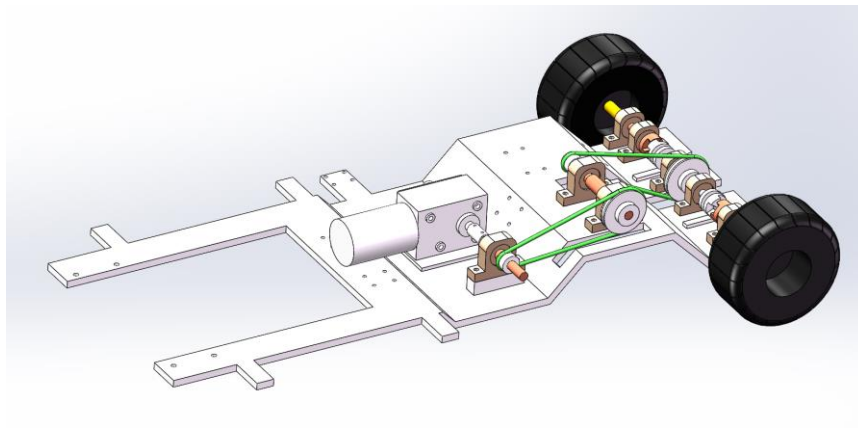


图 2.6 1 动力及辅助转向模块

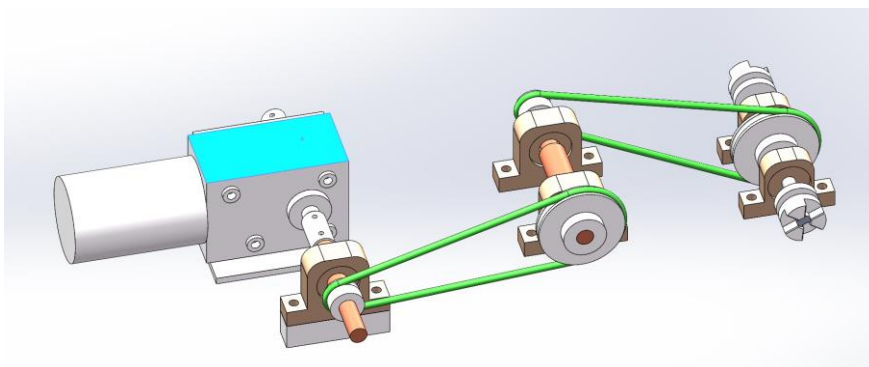


图 2.6 2 动力的传输方式

动力轴两端均装配长键，爪式离合器在动力轴两端且可沿轴向移动，如图 2.6 3。

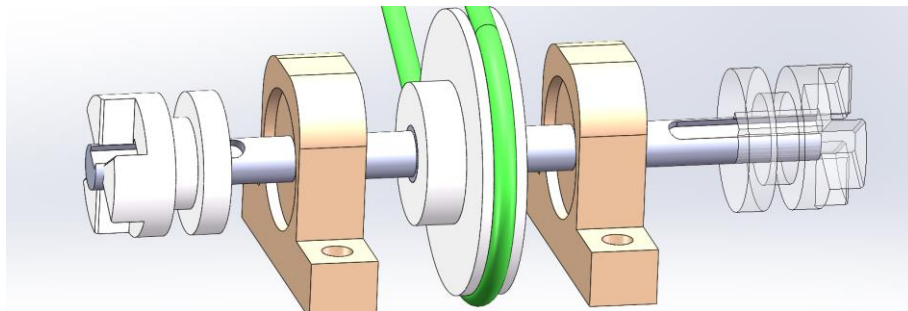


图 2.6 3 爪式离合

该模块创新点在于动力传至后方后可选择性将动力分配到两轮。每个轮轴上均设计爪式离合（如图 2.6 4），动力轴上有可滑动的另一半联轴器，动力轴与轮轴可通过离合器进行连接，当离合器啮合后，动力轴便与轮轴同步转动，为整机提供前进动力。当需要转向时，可单独将某一边的离合啮合，则只有一边轮子转动，从而实现差速转向。动力轴与轮轴之间离合器的啮合是通过轴向移动动力轴上的半联轴器实现的。移动半联轴器则是通过拨杆与半联轴器环槽配合实现，如图 2.6 5。

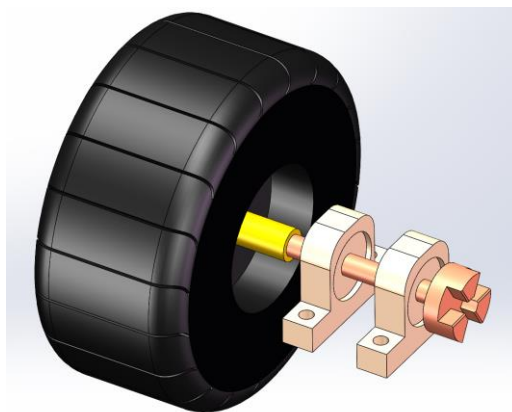


图 2.6 4 轮轴上的离合器

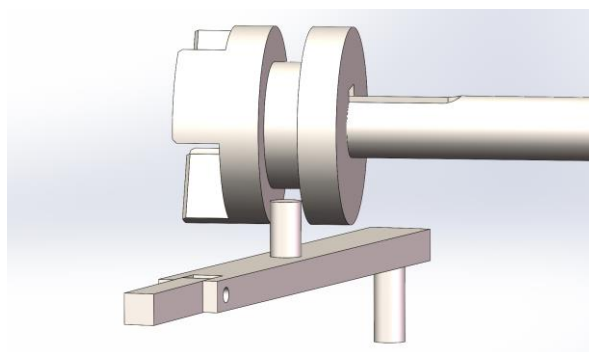


图 2.6 5 拨杆与离合的装配关系

拨杆结构如图 2.6 6。拨杆前端为支点，插入底座的孔中，末端有限位块，



限位块可与底座“离”与“合”两个插口配合，对拨杆进行限位，从而使半离合器的轴向位置固定。左侧拨杆的“合”与“离”状态如图 2.6 7、图 2.6 8。

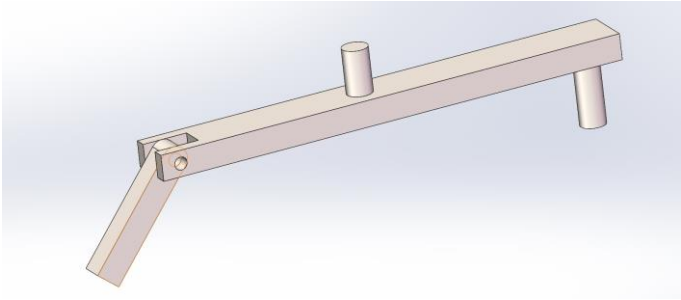


图 2.6 6 离合拨杆

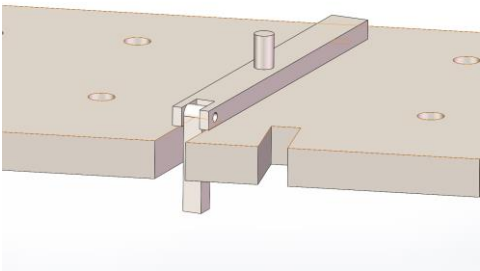


图 2.6 7 左侧拨杆“合”

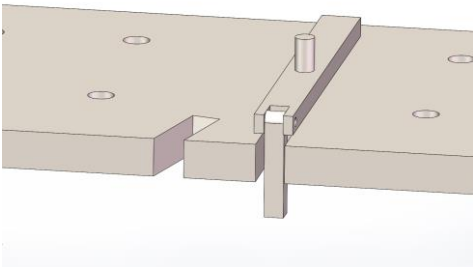


图 2.6 8 左侧拨杆“离”

当需要进行右转向时，将左侧拨杆置为“合”，右侧置为“离”，及此时只有左侧车轮转动，从而实现右转。如图 2.6 9，右转向时，左侧离合器啮合，右侧离合器分开。同理，左转向原理相同。

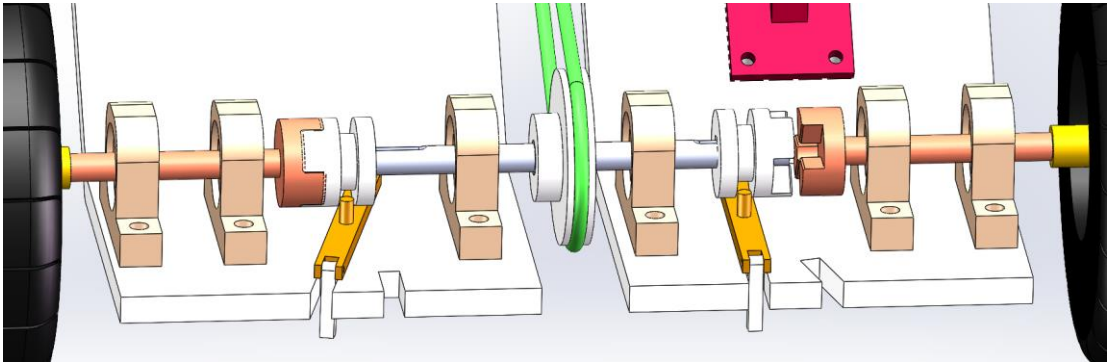
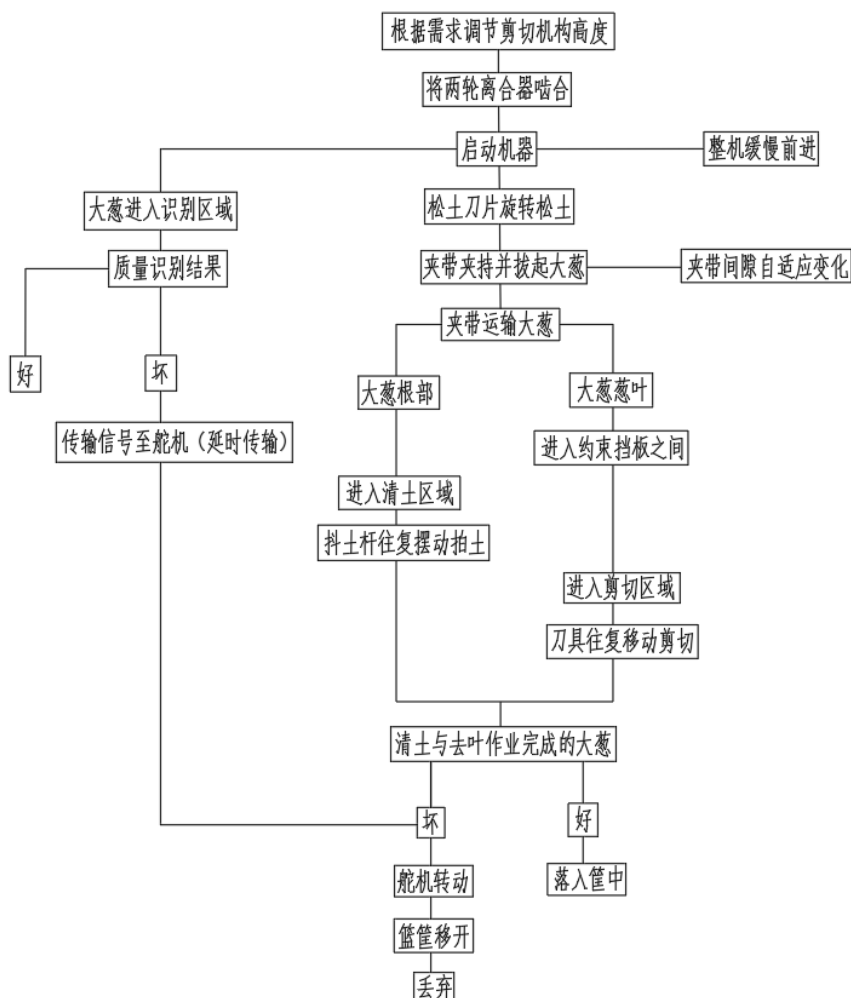


图 2.6 9 右转向时拨杆的位置

### 三.工作流程图

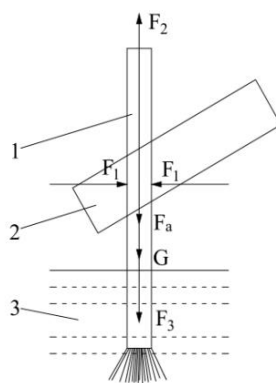


### 四.理论计算与仿真分析

#### 4.1 理论计算

##### 4.1.1 皮带夹持力计算

图 4.1.1 1 为夹持带拔起大葱时的受力， $F_1$  为夹持力； $F_2$  为拔起力； $F_3$  为土壤阻力； $F_a$  为拔起过程中大葱的惯性力； $G$  为大葱重力； $f_1$  为大葱茎秆与输送带之间的摩擦系数。



1.大葱茎秆 2.夹持输送装置 3.土壤

图 4.1.1.1 夹持带拔起大葱时受力分析

根据所查资料，土壤阻力公式如下：

$$F_3 = \int_0^z 2\pi r \mu \gamma K h dh$$

其中  $z$  为土壤中葱秆长度， $r$  为大葱葱秆直径， $\mu$  为土壤与茎秆之间的最大静摩擦系数， $\gamma$  为土的重度， $K$  为侧向土压力系数， $h$  为埋藏深度。

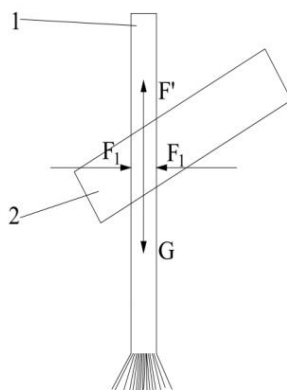
大葱拔起时，由于拔起过程缓慢且大葱质量较小，故也可忽略拔起过程中大葱的惯性力，则联立得到拔起力与夹持力：

$$F_2 = \int_0^z 2\pi r \mu \gamma K h dh + G$$

$$F_1 = \frac{\int_0^z 2\pi r \mu \gamma K h dh + G}{f_1}$$

查资料得大葱茎秆与输送带之间的摩擦系数  $f_1=0.8$ ，土壤与茎秆之间的最大静摩擦系数  $\mu=0.75$ ；土的重度指的是土的天然重度，由土总质量与土总体积的比值决定，通过试验测得土的重度  $\gamma=14528 \text{ N/m}^3$ ；侧向土压力系数近似按  $K=1-\sin\psi$  计算，土体内摩擦角  $\psi$  取  $30^\circ$ ，大葱平均质量在  $300\text{g}$  左右。将各参数代入公式计算，得到夹持输送装置拔起大葱时所需要的拔起力  $F_2=63\text{N}$ ，夹持力  $F_1=79\text{N}$ 。

下图为夹持带运输大葱时的受力图。



1.大葱茎秆 2.夹持输送装置

图 4.1.1.2 夹持带运输大葱时受力分析



通过输送时力学模型进行受力分析可得：

$$F' = G$$

$$F' = f_1 F_1$$

式中， $F_1$  为夹持力，N； $F'$  为输送带与大葱茎秆之间的摩擦力，N； $G$  为大葱重力，N； $f_1$  为大葱茎秆与输送带间的摩擦系数。

联立可得：

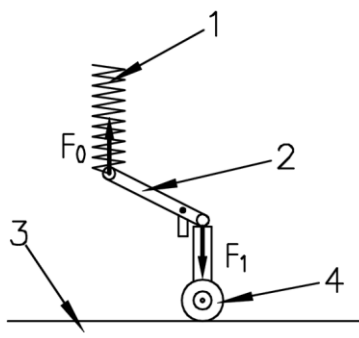
$$F_1 = \frac{G}{f_1}$$

通过计算得到夹持输送装置输送大葱时所需要的夹持力  $F_1=37.5\text{N}$ 。

通过分别分析两个阶段的夹持力发现，拔起时的夹持力大于输送时的夹持力，若夹持输送装置可以顺利拔起大葱，则肯定可以稳定的完成输送过程。大葱拔起时所需要的拔起力小于大葱茎秆的拉断力，不会出现大葱被拉断的现象，所需要的夹持力小于大葱茎秆的压断力，也不会出现大葱被压断的现象。因此，选择拔起时的夹持力作为弹簧的设计依据。

夹持输送装置输送大葱的动力来源于左侧电机，而夹持大葱的力则来源于弹簧的弹力，

如图，拉簧拉着杠杆，使其压迫带轮，从而为夹带提供夹持力  $F_1$ ：



1.拉簧 2.杠杆 3.运输夹带 4.夹持带轮

图 4.1.1 3 夹持带受力分析

由杠杆原理可知：

$$F_0 L_0 = F_1 L_1$$

其中  $L_0$ 、 $L_1$  分别为 25mm、5mm，及：

$$F_0 = F_1 / 5$$

通过计算得到夹带拔起大葱时所需要的夹持力  $F_0=15.8\text{N}$ ，及拉簧的初始压力大于  $F_0$  即可。

## 4.1.2 夹持运输机构参数分析

整机前进速度与夹带运输速度需要有良好配合，如果夹带速度过慢，则会

使大葱在进入夹带的过程中堵塞，若是速度太快，则会发生打滑或大葱葱白扯断的现象。根据以往经验，整机前进速度在  $0.2\sim 0.6\text{m/s}$  最合适。参考其他类根茎收获机械的设计，夹持运输装置倾角一般在  $25^\circ$  到  $35^\circ$  之间，本设计采取倾角  $30^\circ$ 。

如图 4.1.2， $V_a$  为整机前进速度， $V_b$  为夹带运输速度， $V_c$  为大葱拔起时的速度，角  $\alpha$  为夹带倾角  $30^\circ$ 。由于大葱最佳的拔取状态是沿大葱生长的轴心方向，所以大葱拔起速度的方向应与轴心平行（即垂直于地面方向），则角  $\beta$  为  $60^\circ$ 。

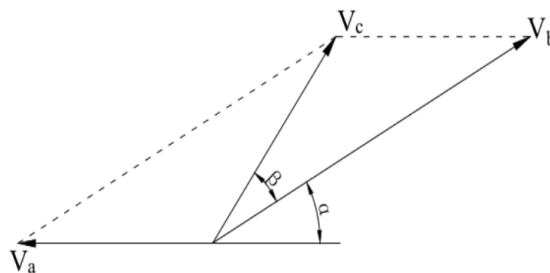


图 4.1.2 大葱、夹持带、整机速度矢量关系图

本设计整机前进速度取  $0.3\text{m/s}$ ，即  $V_a = 0.3\text{m/s}$ ，则  $V_b = 0.6\text{m/s}$ ， $V_c = 0.52\text{m/s}$ 。

### 4.1.3 各电机运行速度分析

由于整机前进速度已确定（ $V_a = 0.3\text{m/s}$ ），即选定车轮直径后即可确定底部电机转速。车轮选定直径  $D=40\text{cm}$ 。

由公式  $v=wr$  得： $w=v/r$ ，则  $w=1.5\text{ rad/s}$ 。如图 4.1.3 1，轮轴与电机轴经过两级带轮传动，且传动比  $i_1 = 2.5$ ， $i_2 = 2.5$ ，则电机轴转速  $w_1 = wi_1i_2 = 9.38\text{ rad/s}$ 。

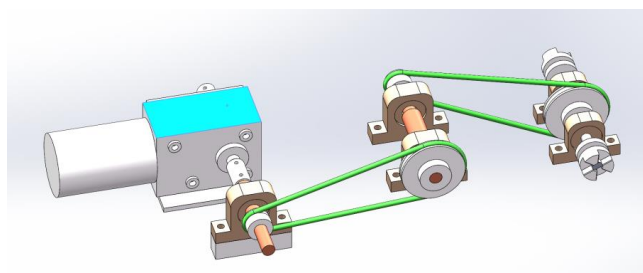


图 4.1.3 1 车轮轴传动示意图

由于夹带的运输速度已确定（ $V_b = 0.6\text{m/s}$ ），即选定主动带轮直径后即可确定左侧电机转速。带轮选定直径  $d=10.6\text{cm}$ 。

同上，如图 4.1.3 2，电机轴与转动带轮用一级皮带传动，传动比  $i_3 = 1$ ，求得电机轴转速  $w_2 = 11.32\text{ rad/s}$ 。

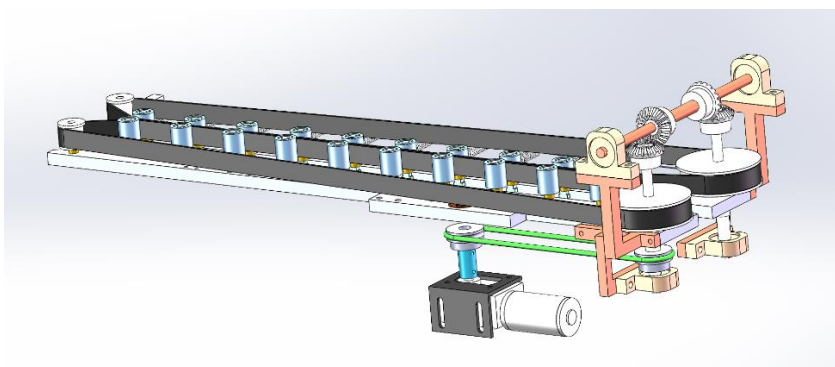


图 4.1.3 1 夹持运输带传动示意图

#### 4.1.4 抖土杆相关参数设计

如图 4.1.4，抖土杆相关参数包括抖土两杆间的间距和抖土杆的长度，其中抖土两杆的间距由大葱直径决定，而抖土杆的长度则与大葱抖土的时长（即大葱根部经过抖土区域的时间）有关。

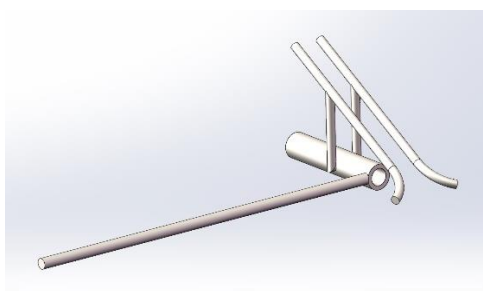


图 4.1.4 抖土杆

查阅相关论文发现大葱葱白平均直径在 30mm 左右，为了使大葱根部能够顺利进入并通过两杆，同时又保证两杆能够对根部进行拍打，两杆间距设计为 3 倍大葱直径，即两杆轴线距离 90mm。

该去土机构的抖土杆由底部电机带动滑槽滑块上下往复移动从而使抖土杆往复摆动，即底部电机轴转动一圈，抖土杆往复摆动一次，拍打两次。底部电机转速  $w_1 = 9.38 \text{ rad/s}$ ，即一秒钟拍打 9.38 次。查阅资料发现大葱根部需要拍打 9~18 次，次数过多会拍伤大葱葱白，次数太少则去土效果不好。本设计拟定拍打次数  $n=18$  次，即去土过程中电机轴需要转动  $k=4.5$  圈。

去土时间  $t=k/w_1=0.48\text{s}$ ，则输送带运输距离  $S=V_b t = 0.29\text{m}$ ，即抖土杆长度  $L$  等于  $S$  时，可以做到拍土次数 18 次，取抖土杆长度  $L=0.29\text{m}$ 。

#### 4.1.5 剪切机构相关参数分析

如图 4.1.5，剪切机构由曲柄滑块机构执行剪切动作，曲柄转动一周，刀片往复移动一次。其参数包括刀片往复移动的频率与刀架高度调节范围。

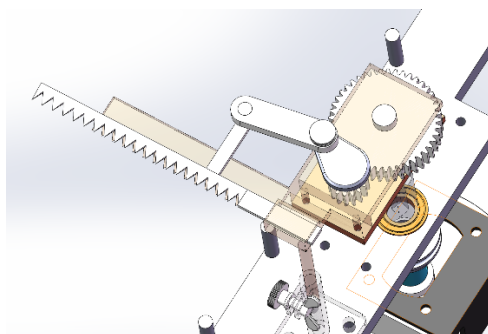


图 4.1.5 剪切机构示意图

为了提高剪切频率，电机轴经过一级齿轮传动曲柄，齿轮传动比  $i=2$ ，则曲柄转速  $w_0 = iw_2 = 22.64 \text{ rad/s}$ ，即刀片往复频率  $f=22.64$ ，满足剪切频率要求。

加工后的大葱葱叶长度在 10~30cm，即去叶机构的高度调节范围设计在 20cm 左右。

## 4.2 关键零件仿真分析

如图 4.1 1，该构件为机架与底板的支撑件。选定材料 HT200，底部固定后沿重力方向对其上部添加等效于 100kg 机架的重力。Ansys 分析如图 4.1 2，其最大应力发生在支架上部，达到 124.93MPa，而 HT200 材料屈服强度为 200MPa，该支架强度足够。

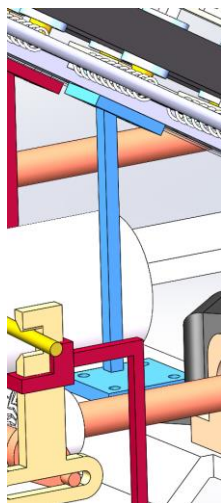


图 4.1 1 连接底座与机架的支架

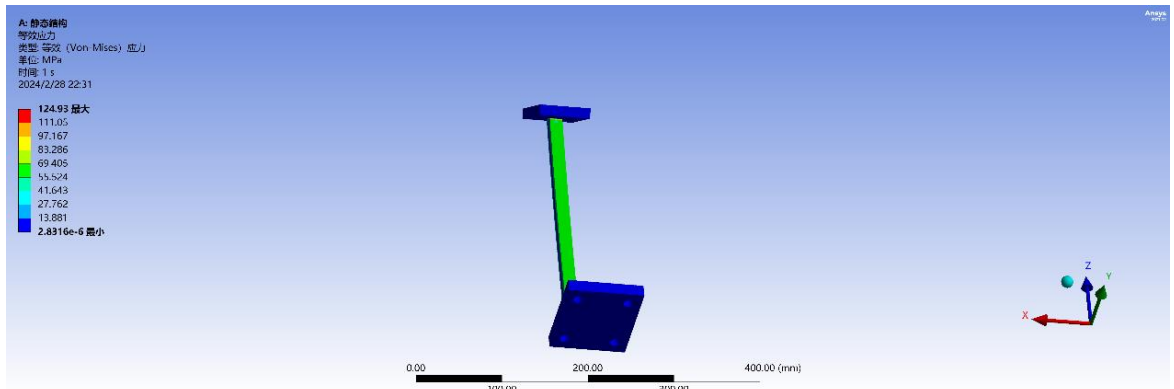


图 4.1 2 支架应力分析云图

如图 4.1 3，该构件为松土刀片。选定材料 HT200，模拟其在土壤中松土时刀片受力，对其中一块刀片进行分析，将土壤对刀片的阻力视为垂直于刀片 500N 的力，Ansys 分析得到最大应力在刀片根部（如图 4.1 4），达到 128.69MPa，该松土盘强度足够。

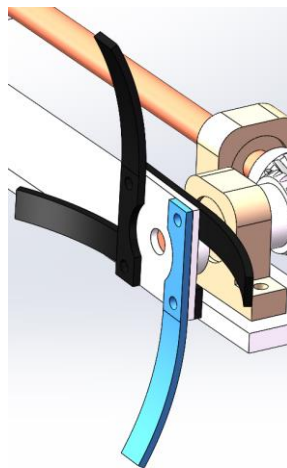


图 4.1 3 松土耙

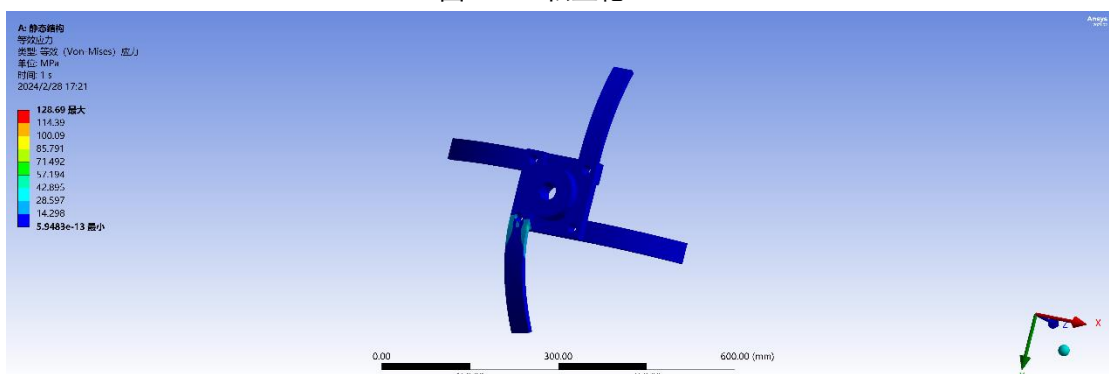


图 4.1 4 松土刀片应力分析云图

如图 4.1 5，该构件为底部电机动力传至松土机构的传动轴。选定材料 HT200，该轴两端分别安装一锥齿轮，且用两个轴承座支撑。为了简化计算，

对其进行简化分析（即不考虑径向力），则视为两端施加等大反向的力，根据松土刀片受力与转速，推算出该轴两端的力在 1000N 左右，Ansys 分析得到该应力最大值在轴承座支点处如图 4.1 6，达到 59.68MPa，该轴强度足够。

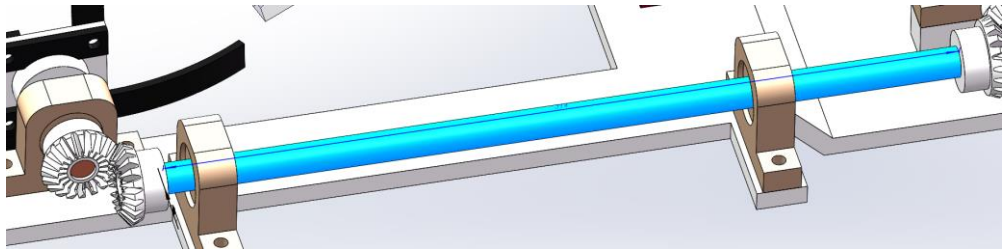


图 4.1 5 松土传动轴

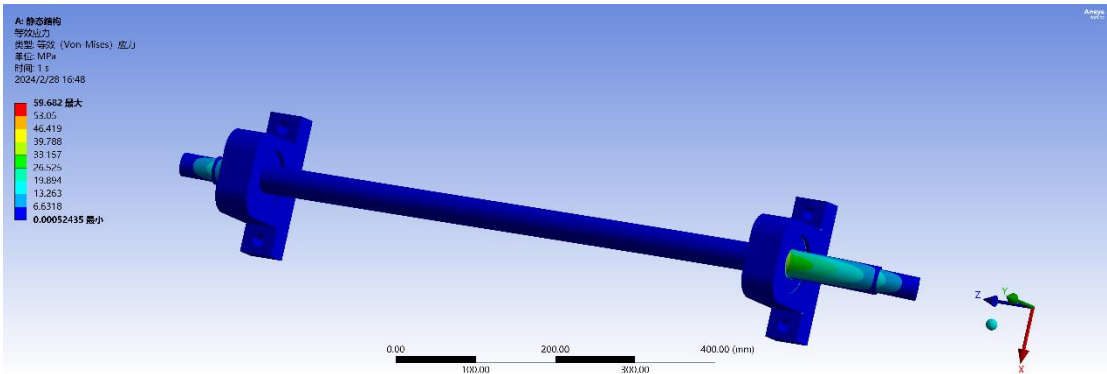


图 4.1 6 传动轴应力分析云图

## 五.总结

### 5.1 创新点总结

#### 柔性夹持运输

夹持机构一侧为柔性，十组拉簧、杠杆、滑块组成。柔性夹持即夹持过程中夹带间距可随大葱直径变化而变化，既可以满足不同品种大葱的夹持，又可以保护大葱葱白部分不受损伤，大大减小伤葱率。

#### 剪切机构高度可调

剪切机构的高度调节仿照自行车桌椅的高度调节，利用螺栓锁紧的方式实现剪切高度的固定，甚至取下剪切机构。当葱农对大葱葱叶有不同处理要求时，可根据需求调节剪切高度，或选择取下剪切机构，保留葱叶。减少后续人工去叶的处理时间与劳动成本。

#### 离合器控制动力分配

机械后方有分别控制两轮动力的拨杆，通过调节拨杆，可实现机械前进、左转向、右转向等不同状态。该结构不需要舵机实现差速转向，无需数控，减少成本与整机重量。

#### 智能物联网识别筛选

利用物联网识别系统实现对病斑、枯烂、虫洞等状况的大葱进行筛选，若

品质太差，将会对其进行舍弃，实现智能化收获。该创新减小后期人工筛选成本，大大提高收获质量

## 5.2 展望

本项目大葱收获机构相较于市场其他产品，可实现的功能更加完善，损伤率较小，且工作流程简单，结构易实现。将大葱收获的各个作业集成化，实现松土、拔葱、抖土、去叶、筛选、入筐一体化。剪切机构高度可调满足葱农对葱叶处理的不同要求；夹带间隙自适应可大大减小机械对葱白的损伤率；离合器分配动力可不依赖舵机实现转向，减小产品成本与重量；智能视觉识别可对大葱质量进行筛选，提高收获质量，降低后期筛选成本；收获机左侧有空筐储存架，当篮筐大葱收集满后，可立即更换篮筐，提高收获连续性与便捷性。

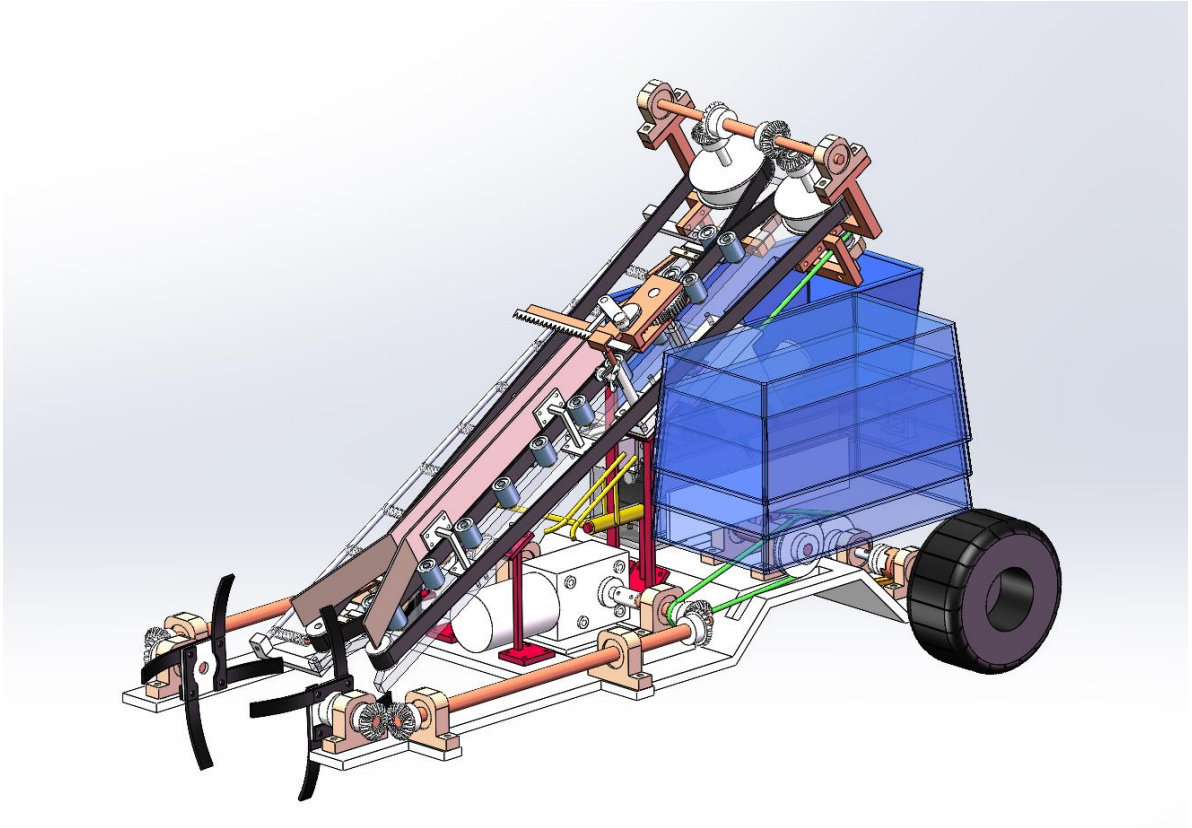
随着科技的不断进步，大葱联合收获机的技术也将继续提升和创新，未来的发展前景有望更加广阔。大葱联合收获机在提高生产效率、节约劳动力成本和适应农业科技进步等方面具备广阔的市场前景。



## 六.参考文献

- [1] 吴昊,肖冰,王琦,等.果苗苗木起苗机的初步分析.林业机械与木工设备,2017,45(10):27-29. DOI:10.3969/j.issn.2095- 2953.2017.10.008
- [2] 基于 SolidWorks 的组合式大葱起苗机的设计与试验. 张吉强;牛子孺;李天华;吴彦强;李广华;侯加林.农机化研究,2020
- [3] 4CZ-1 型履带自走式大葱联合收获机的研制. 张德学;张军强;刘学峰;秦喜田;李青江;闵令强;任冬梅;张翠英.农机化研究,2022
- [4] 大葱收获机挖掘装置的关键技术研究. 王超.青岛理工大学,2019
- [5] 贾智博,侯加林,刘敬伟,等.大葱茎秆起拔力测试系统的设计与试验 [J].农机化研究,2019,41(02):109-113.

## 附录 1 产品三维图



大葱收获机三维图

## 附录 2 视觉识别程序

```
import cv2
import numpy as np
import onnxruntime as ort
import time

def plot_one_box(x, img, color=None, label=None, line_thickness=None):
    """
    description: Plots one bounding box on image img,
                  this function comes from YOLOv5 project.
    param:
        x:       a box likes [x1,y1,x2,y2]
        img:      a opencv image object
        color:    color to draw rectangle, such as (0,255,0)
        label:    str
        line_thickness: int
    return:
        no return
    """
    tl = (
        line_thickness or round(0.002 * (img.shape[0] + img.shape[1]) / 2) + 1
    ) # line/font thickness
    color = color or [random.randint(0, 255) for _ in range(3)]
    c1, c2 = (int(x[0]), int(x[1])), (int(x[2]), int(x[3]))
    cv2.rectangle(img, c1, c2, color, thickness=tl, lineType=cv2.LINE_AA)
    if label:
        tf = max(tl - 1, 1) # font thickness
        t_size = cv2.getTextSize(label, 0, fontScale=tl / 3, thickness=tf)[0]
        c2 = c1[0] + t_size[0], c1[1] - t_size[1] - 3
        cv2.rectangle(img, c1, c2, color, -1, cv2.LINE_AA) # filled
        cv2.putText(
            img,
            label,
            (c1[0], c1[1] - 2),
            0,
            tl / 3,
            [255, 255, 255],
            thickness=tf,
            lineType=cv2.LINE_AA,
        )

def _make_grid(nx, ny):
    xv, yv = np.meshgrid(np.arange(ny), np.arange(nx))
    return np.stack((xv, yv), 2).reshape((-1, 2)).astype(np.float32)

def cal_outputs(outs, nl, na, model_w, model_h, anchor_grid, stride):
    row_ind = 0
    grid = [np.zeros(1)] * nl
    for i in range(nl):
        h, w = int(model_w / stride[i]), int(model_h / stride[i])
        length = int(na * h * w)
        if grid[i].shape[2:4] != (h, w):
            grid[i] = _make_grid(w, h)

        outs[row_ind:row_ind + length, 0:2] = (outs[row_ind:row_ind + length, 0:2] * 2. - 0.5 + np.tile(
            grid[i], (na, 1))) * int(stride[i])
        outs[row_ind:row_ind + length, 2:4] = (outs[row_ind:row_ind + length, 2:4] * 2) ** 2 * np.repeat(
            anchor_grid[i], h * w, axis=0)
        row_ind += length
    return outs

def post_process_opencv(outputs, model_h, model_w, img_h, img_w, thred_nms, thred_cond):
    conf = outputs[:, 4].tolist()
    c_x = outputs[:, 0] / model_w * img_w
    c_y = outputs[:, 1] / model_h * img_h
    w = outputs[:, 2] / model_w * img_w
    h = outputs[:, 3] / model_h * img_h
    p_cls = outputs[:, 5:]
    if len(p_cls.shape) == 1:
        p_cls = np.expand_dims(p_cls, 1)
    cls_id = np.argmax(p_cls, axis=1)

    p_x1 = np.expand_dims(c_x - w / 2, -1)
    p_y1 = np.expand_dims(c_y - h / 2, -1)
    p_x2 = np.expand_dims(c_x + w / 2, -1)
    p_y2 = np.expand_dims(c_y + h / 2, -1)
    areas = np.concatenate((p_x1, p_y1, p_x2, p_y2), axis=-1)

    areas = areas.tolist()
    ids = cv2.dnn.NMSBoxes(areas, conf, thred_cond, thred_nms)
    if len(ids) > 0:
        return np.array(areas)[ids], np.array(conf)[ids], cls_id[ids]
    else:
        return [], [], []
```

```

def infer_img(img0,net,model_h,model_w,nl,na,stride,anchor_grid,thred_nms=0.4,thred_cond=0.5):
    # 图像预处理
    img = cv2.resize(img0, [model_w,model_h], interpolation=cv2.INTER_AREA)
    img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    img = img.astype(np.float32) / 255.0
    blob = np.expand_dims(np.transpose(img, (2, 0, 1)), axis=0)

    # 模型推理
    outs = net.run(None, {net.get_inputs()[0].name: blob})[0].squeeze(axis=0)

    # 输出坐标矫正
    outs = cal_outputs(outs,nl,na,model_w,model_h,anchor_grid,stride)

    # 检测框计算
    img_h,img_w,_ = np.shape(img0)
    boxes,confs,ids = post_process_opencv(outs,model_h,model_w,img_h,img_w,thred_nms,thred_cond)

    return boxes,confs,ids

if __name__ == "__main__":

    # 模型加载
    model_pb_path = "best.onnx"
    so = ort.SessionOptions()
    net = ort.InferenceSession(model_pb_path, so)

    # 标签字典
    dic_labels= {0:'normal',
                  1:'yellow'}

    # 模型参数
    model_h = 320
    model_w = 320
    nl = 3
    na = 3
    stride=[8.,16.,32.]
    anchors = [[10, 13, 16, 30, 33, 23], [30, 61, 62, 45, 59, 119], [116, 90, 156, 198, 373, 326]]
    anchor_grid = np.asarray(anchors, dtype=np.float32).reshape(nl, -1, 2)

    video = 1
    cap = cv2.VideoCapture(video)
    flag_det = False

    while True:
        success, img0 = cap.read()
        if success:

            if flag_det:
                t1 = time.time()
                det_boxes,scores,ids = infer_img(img0,net,model_h,model_w,nl,na,stride,anchor_grid,thred_nms=0.4,thred_cond=0.5)
                t2 = time.time()

                for box,score,id in zip(det_boxes,scores,ids):
                    label = '%s: %.2f'%(dic_labels[id],score)
                    print(label)
                    plot_one_box(box.astype(np.int16), img0, color=(255,0,0), label=label, line_thickness=None)
                    print(id)
                    if(id==1):
                        print("This is yellow scallion")
                    else:
                        print("This is normal scallion")
                str_FPS = "FPS: %.2f"%(1./(t2-t1))

                cv2.putText(img0,str_FPS,(50,50),cv2.FONT_HERSHEY_COMPLEX,1,(0,255,0),3)

            cv2.imshow("video",img0)
            key=cv2.waitKey(1) & 0xFF
            if key == ord('q'):

                break
            elif key & 0xFF == ord('s'):
                flag_det = not flag_det
                print(flag_det)

    cap.release()

```

## 上位机 Orange Pi Zero3 的视觉识别程序