



دانشکده مهندسی مکانیک

درس مکاترونیک

پروژه پایانی درس

ساخت و کنترل پنجه مکانیکی

استاد درس: دکتر شهبازی

دانشجویان:

علی محمد دوست 994000001

محمد طالبی 99542231

نرجس خیری 400541405

بِسْمِ اللّٰهِ الرَّحْمٰنِ الرَّحِيْمِ

## چکیده

این پروژه به منظور طراحی و ساخت یک پنجه رباتیکی (گریپر) به عنوان ابزاری مکاترونیکی اجرا شد. هدف اصلی پروژه، ساخت پنجه‌ای است که قادر باشد اجسام با حداکثر جرم 70 گرم و قطر 5 سانتی‌متر را بگیرد و بدون آسیب رسانی به آن‌ها، نگهداری کند.

در مرحله اول، با بررسی مکانیزم‌های متداول، مکانیزم رک و پینیون انتخاب شد که به دلیل ساده‌تر بودن، حجم کم‌تر و تناسب مکانیزم با موتور سرво مشخص شده در شرح پروژه، مدلی مناسب برای پروژه به شمار می‌رفت. با استفاده از این مکانیزم، با حرکت 180 درجه ای سروموتور بازه حرکتی پنجه به 7 تا 56 میلی‌متر یعنی به طور کلی 49 میلی‌متر قابلیت جا به جایی داشت.

برای استفاده از سنسور نیرو (FSR)، این سنسور در شرایط واقعی با اوزان مختلف و با مقاومت‌های مختلف تست شد. از این طریق رابطه بین نیروی واقعی و خروجی سنسسور مشخص شد و با یک رگرسیون خطی، ضریب کالیبراسیون سنسور به دست آمد. همچنین به دلیل نوسانات بالای سنسور و عدم پایداری آن، برای کاهش خطای فیلتر میانگین Moving average استفاده شد.

برای شناسایی اجسام نرم و سفت و اعمال نیروی مناسب برای هر کدام، مفهوم «سفتی» تعریف شد. در تستی جداگانه، پنجه با لمس جسم و اندازه‌گیری جایه‌جایی، مقدار سفتی محاسبه شد. همچنین مقدار نیروی مناسب نیرو برای گرفتن جسم در حالتی که نیرو نه زیاد باشد و نه کم، به دست آمد و در اینجا هم با یک رگرسیون خطی رابطه بین سفتی و نیروی لازم برای گرفتن اجسام به دست آمد. و بدین ترتیب در تست‌ها پس از محاسبه سفتی هر جسم، میزان نیروی هدف آن تخمین زده می‌شود.

در نهایت، روش‌های کنترلی مورد بررسی قرار گرفتند. ابتدا از کنترلر PID استفاده شد، اما با انجام چندین تست مشخص شد که کنترلر P به تنها‌ی می‌تواند نتایج مطلوبی ارائه دهد. این روش نه تنها ساده‌تر است، بلکه به راحتی توانست با دریافت فیدبک از سنسور و مقدار نیروی هدف، گریپر را به مقدار لازم باز یا بسته کند. همچنین از کلید برای شروع پروسه گریپینگ، چراغ LED برای هشدار باز یا بسته شدن گریپر و LCD برای نمایش سفتی و نیرو‌ها استفاده شد.

در نهایت عملکرد گریپر بر روی چندین جسم مختلف با سفتی‌های خیلی متفاوت، مانند لیوان کاغذی و تخم مرغ و توب فوتbal دستی تست شد که هر کدام از اجسام را با نیروی مناسب و به طور پایدار نگه داشت.

## فهرست

2 .....	
3 .....	چکیده
5 .....	مقدمه
5 .....	معرفی هدف پژوهش
6 .....	شرح مسئله پژوهش
7 .....	طراحی مدل و مکانیزم گریپر
7 .....	مقایسه مکانیزم ها
9 .....	مدلسازی گریپر
19 .....	اجزای سیستم
19 .....	معرفی قطعات و روش های کالیبراسیون
25 .....	لیست هزینه قطعات
26 .....	چالش ها
27 .....	پیاده سازی کنترل و تست
27 .....	چالش اجسام با جنس مختلف
32 .....	روش کنترل
33 .....	شبیه سازی
36 .....	تست کد بر روی گریپر
40 .....	تست نهایی
42 .....	جمع بندی و نتیجه گیری
43 .....	منابع

## مقدمه

### معرفی هدف پژوهه

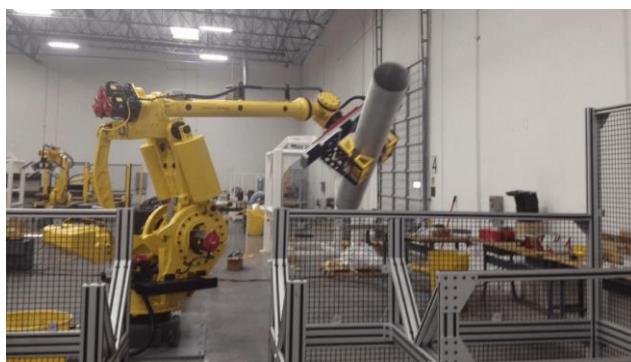
گریپرهای صنعتی برای افزایش سرعت، دقت و بهره‌وری در فرآیندهای تولیدی استفاده می‌شوند. در حوزه‌های غیرصنعتی، این ابزارها در تحقیقات علمی، پزشکی، آموزش، هنر و کمک به افراد معلول و سالمندان کاربرد دارند.

کاربردهای صنعتی:

1. خودروسازی : جابه‌جایی قطعات خودرو.
2. الکترونیک : برداشتن قطعات کوچک روی بردگاهی مدار.
3. صنایع سنگین : جابه‌جایی قطعات فلزی.

کاربردهای غیرصنعتی:

1. تحقیقات علمی : ربات‌های فضایی و زیردریایی.
2. پزشکی : ربات‌های جراحی و پروتزهای مصنوعی.
3. آموزش : آموزش رباتیک و پژوهش‌های دانشگاهی.
4. هنر : ربات‌های هنرمند و مسابقات رباتیک.
5. توان بخشی : دستهای رباتیکی و ربات‌های کمک‌رسان.
6. محیط‌زیست : جمع‌آوری زباله و نجات حیوانات.



تصویر 2- گریپر متصل به بازوی رباتیک صنعتی برای جابه‌جایی قطعات



تصویر 1- گریپر متصل به بازو در ربات‌های جراح

## شرح مسئله پروژه

پروژه پایانی درس مقدمه‌ای بر مکاترونیک، طراحی و ساخت یک پنجه رباتیکی (Gripper) است. این پنجه به عنوان ابزاری مکاترونیکی، به انتهای بازوهای رباتیکی متصل می‌شود و قادر است اجسام مختلف را با ابعاد و جنس‌های متفاوت نگهدارد. هدف اصلی این پروژه، ساخت پنجه‌ای است که بتواند اجسام با حداکثر جرم ۷۰ گرم و قطر ۵ سانتی‌متر را با نیروی مناسب بگیرد و از آسیب‌رسانی به اجسام جلوگیری کند.

پنجه رباتیکی طراحی شده باید قابلیت نگهداری اجسام با شکل‌های مختلف (مانند استوانه‌ای، تخم مرغ و حتی اجسام انعطاف‌پذیر مانند توپ تنیس) را داشته باشد. برای اطمینان از اتصال کامل پنجه به جسم و جلوگیری از آسیب‌رسانی به اجسام، از سنسورهای نیرو (FSR) در نقطه تماس جسم و پنجه استفاده می‌شود. همچنین، میزان نیروی اعمال شده توسط پنجه باید به کمک یک نمایشگر LCD به کاربر نشان داده شود.

### محدودیت‌ها و الزامات پروژه

در طراحی و ساخت این پنجه رباتیکی، محدودیت‌ها و الزامات زیر باید رعایت شوند:

#### ۱. استفاده از دو فک (انگشت) و یک سروو موتور:

- طراحی پنجه باید شامل دو فک و یک سروو موتور باشد.
- گروه‌ها مجاز به استفاده از سروو موتورهای سری MG995 و MG996 هستند.

#### ۲. توانایی نگهداری اجسام تا ۷۰ گرم:

- پنجه باید قادر به نگهداری اجسام با حداکثر جرم ۷۰ گرم و قطر ۵ سانتی‌متر باشد.
- اجسام می‌توانند شامل اشیاء شکننده (مانند تخم مرغ) و انعطاف‌پذیر (مانند توپ تنیس) باشند.

#### ۳. طراحی پایه برای پنجه:

- پنجه باید روی یک پایه طراحی شده قرار گیرد و در صفحه عمودی عمل کند.
- پایه باید به گونه‌ای طراحی شود که پنجه از بالا به آن متصل شود.

#### ۴. عملکرد خودکار:

- پنجه باید دارای یک کلید باشد که با فشردن آن، پنجه به طور کامل باز شود و آماده گرفتن جسم باشد.
- پس از ۵ ثانیه از باز شدن کامل، پنجه باید به صورت خودکار بسته شود و جسم را بگیرد.
- پنجه باید جسم را به مدت ۱۰ ثانیه نگهدارد و سپس به آرامی آن را رها کند.

#### ۵. ساخت بدنه و اتصالات:

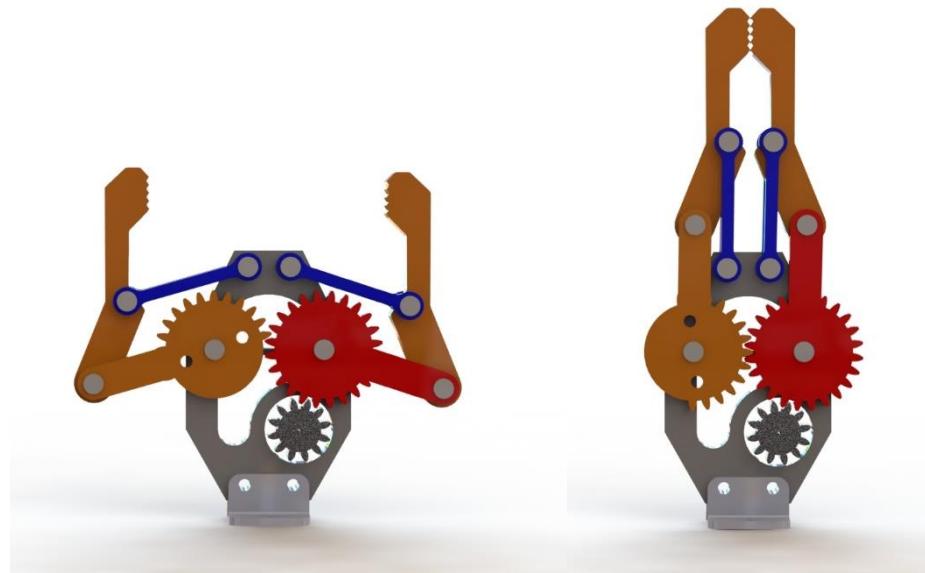
- برای ساخت بدن پنجه و اتصالات آن، می‌توان از روش‌هایی مانند پرینت سه‌بعدی، برش لیزری یا ساخت دستی استفاده کرد.
- استفاده از پنجه‌های آماده غیرمجاز است.

## طراحی مدل و مکانیزم گریپر

ما برای طراحی مکانیزم گریپر به دنبال مکانیزمی بودیم تا با توجه به نوع سروموتور، حجم بهینه و هزینه کمینه برای ما در بهترین حالت باشد. با جست و جو در پروژه‌های مشابه چندین نمونه مکانیزم رایج برای گریپرها یافت شد که معیارهای بالا برای هر کدام بررسی و در نهایت بهترین گزینه انتخاب شد و طراحی با الهام از آن و با اصلاحاتی مناسب با نیاز مسئله ما انجام شد.

### مقایسه مکانیزم‌ها

#### مکانیزم اول



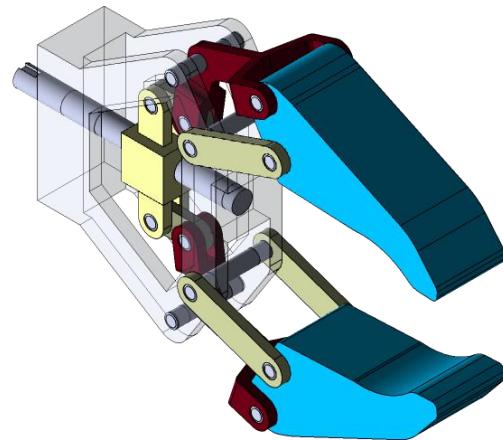
تصویر ۳- مکانیزم اول

معایب:

- حجم بزرگ

- باز و بسته شدن غیر خطی
- نا مناسب برای موتور سرورو با محدودیت زاویه

#### مکانیزم دوم

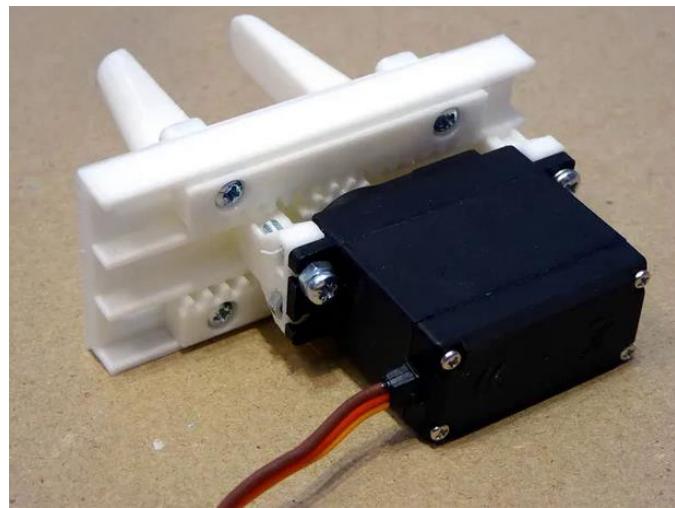


تصویر 4- مکانیزم دوم

معایب:

- نا مناسب برای موتور سرورو با محدودیت زاویه
- باز و بسته شدن غیر خطی

#### مکانیزم سوم



تصویر5- مکانیزم سوم

محاسن:

- مناسب برای موتور سرورو با محدودیت زاویه
- باز و بسته شدن خطی
- حجم کم

### مدلسازی گریپر

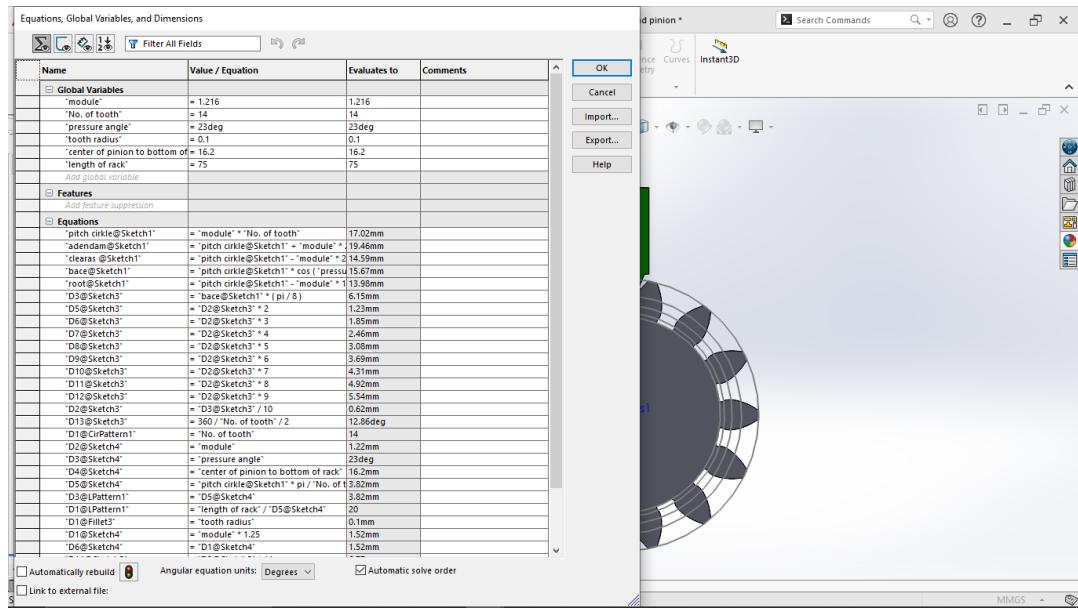
- پس از اینکه مکانیزم مطلوب [1] مشخص شد نیاز بود مکانیزم گریپری طراحی کنیم که با چرخش سرورو موتور در بازه 0 درجه ، انگشت های ان بین بازه 1 تا 5 سانتی متر باز و بسته بشوند.

مکانیزم های مختلفی برای تبدیل حرکت دورانی سرورو به حرکت خطی وجود داشت اما با توجه به اینکه نیرو ها در این پروژه اهمیت دارد و هدف بر این بود تا نیروی گشتاور سرورو بسیار ساده و با حداقل لینک ها به انگشت های گریپر برسد .

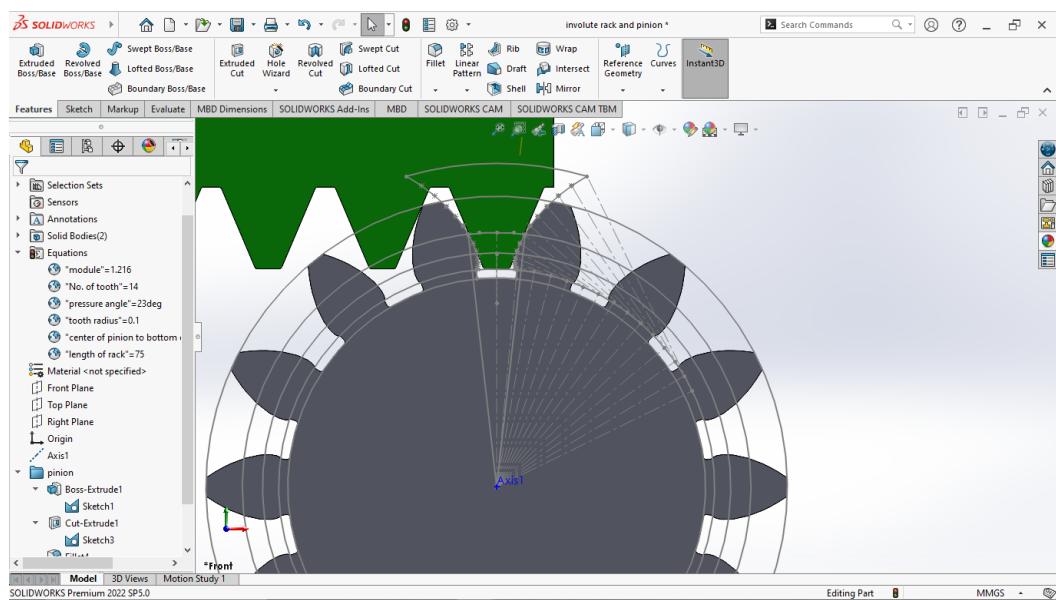
### مرحله اول

اولین قدم طراحی رک و پینیونی بود که به ازای 175 درجه چرخش پینیون (5 میلی متر از بازه سرورو برای در نظر گرفتن خطا ) ، رک به میزان 26 میلی متر ( 6 میلی متر بیشتر از 20 باز هم برای در نظر گرفتن خطا و افزایش بازه کل ) جابجا شود . برای افزایش دقت از پکیج اماده طراحی رک و پینیون سالید ورک استفاده نکردیم و رک و پینیون اینگلوت را از صفر طراحی کردیم .

در ادامه مراحل این طراحی اورده شده است:



تصویر ۶-مراحل طراحی پینیون



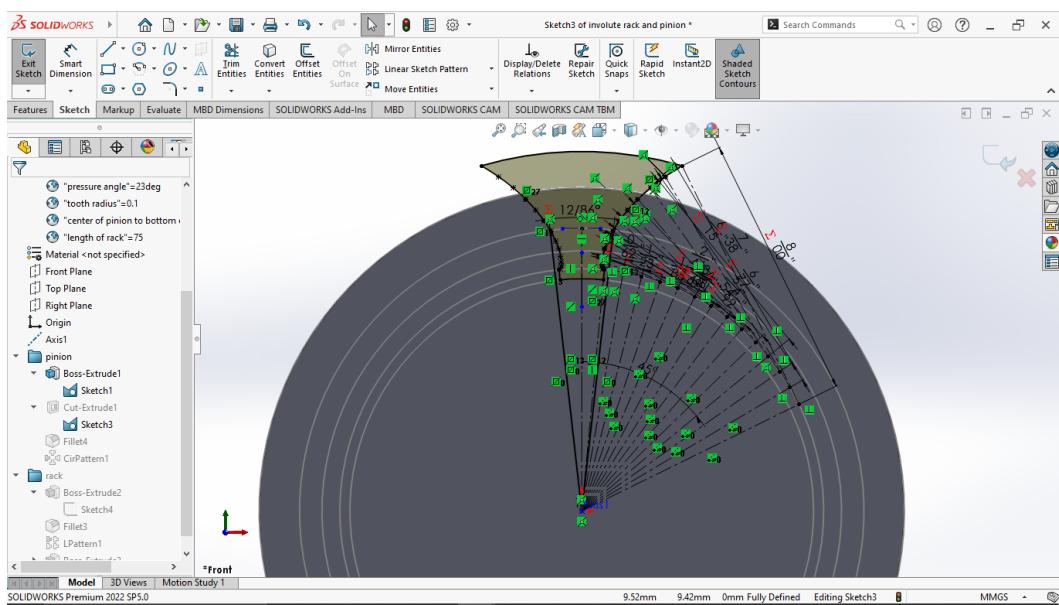
تصویر ۷-مراحل طراحی پینیون

از ابتدای طراحی تمام پارامتر ها در قسمت معادلات سالید ورک وارد شدند تا ویرایش ویرگی های مکانیزم ممکن باشد .

مراحل :

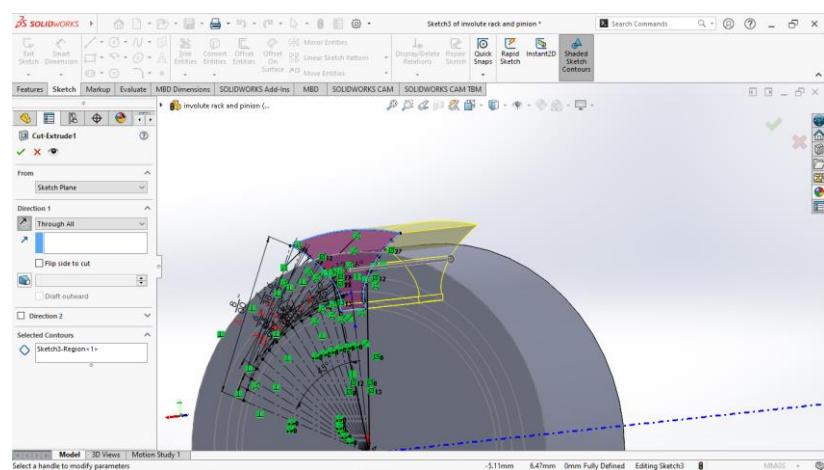
1. رسم دایره های مورد نیاز برای طراحی چرخ دنده اینگلوت .
2. حجم دادن به بزرگ ترین دایره .

.3 رسم منحنی اینگلوت با استفاده از خطوط مماس و روابط مربوطه .



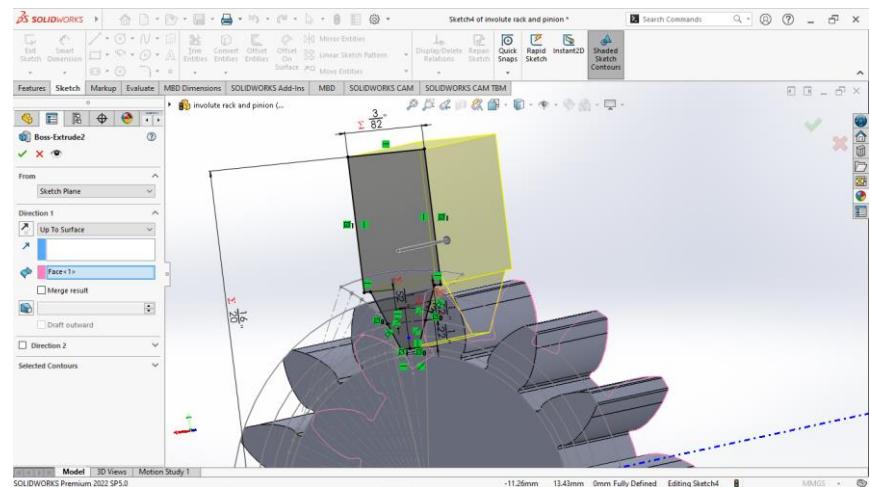
تصویر8-مراحل طراحی پینیون

.4 کم کردن اینگلوت ها از حجم کل برای تشکیل دندانه ها.



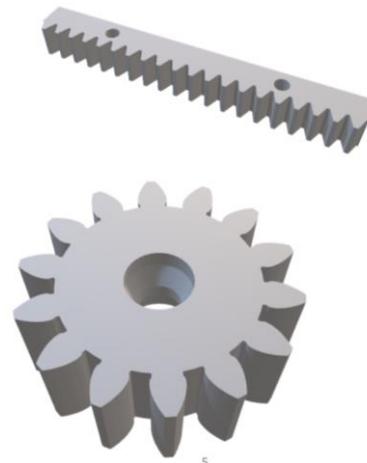
تصویر9-مراحل طراحی پینیون

## 5. طراحی دندانه رک با توجه به ابعاد چرخدنده.



تصویر 10- طراحی دندانه رک

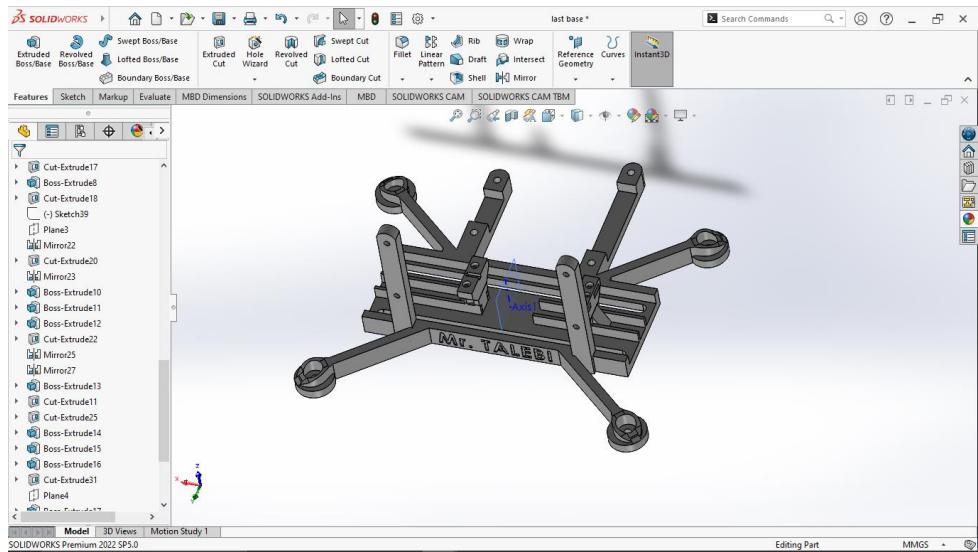
در این طراحی سعی شده بازه چرخش سروو با بازه گریپر منطبق باشد تا از حداکثر دقیق سروو استفاده شود.



تصویر 11- رک و پینیون طراحی شده

## مرحله دوم

پس از طراحی رک و پینیون و اطمینان از عملکرد صحیح آن در محیط اسمنلی، طراحی سازه اصلی گریپر را اغاز کردیم.



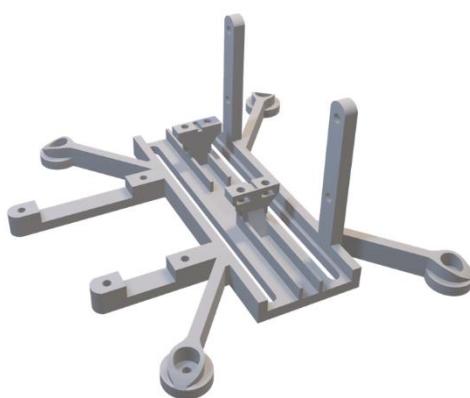
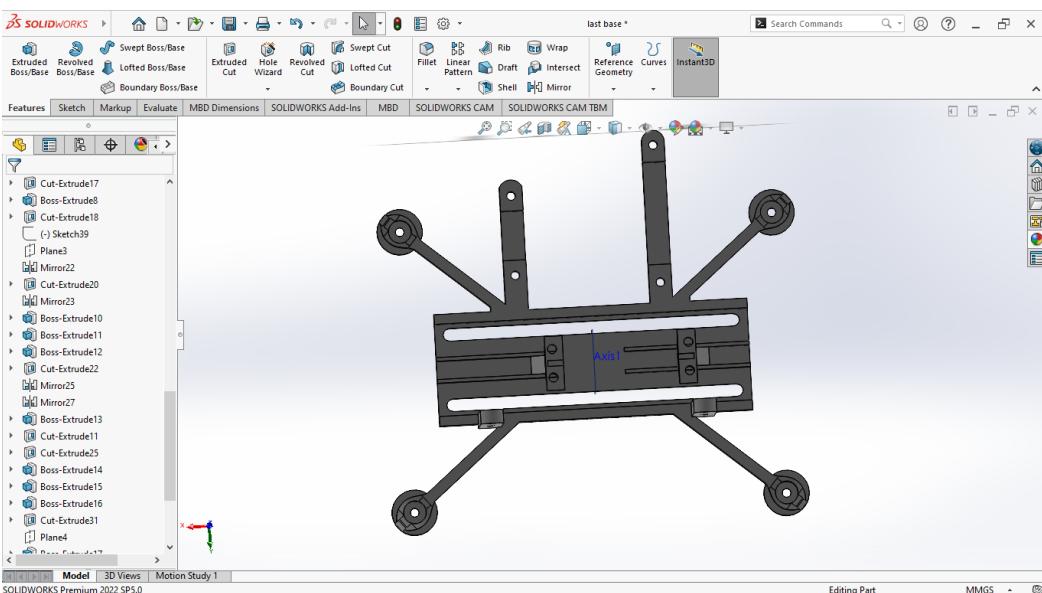
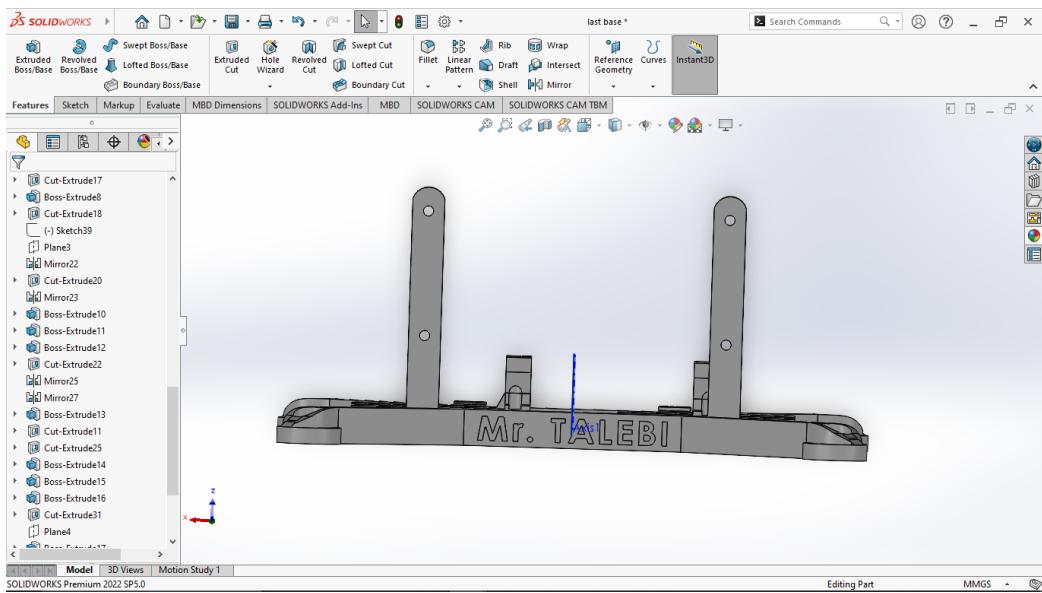
تصویر 12- بنده گریپر

در این قسمت نیاز بود با نکات زیادی رعایت شود .

به تشریح جز به جز مراحل طراحی و ابعاد ان نمیپردازیم اما فایل ها پیوست است و میتوان مراحل را مشاهده کرد .

اما برای این قسمت به این نکات توجه شده :

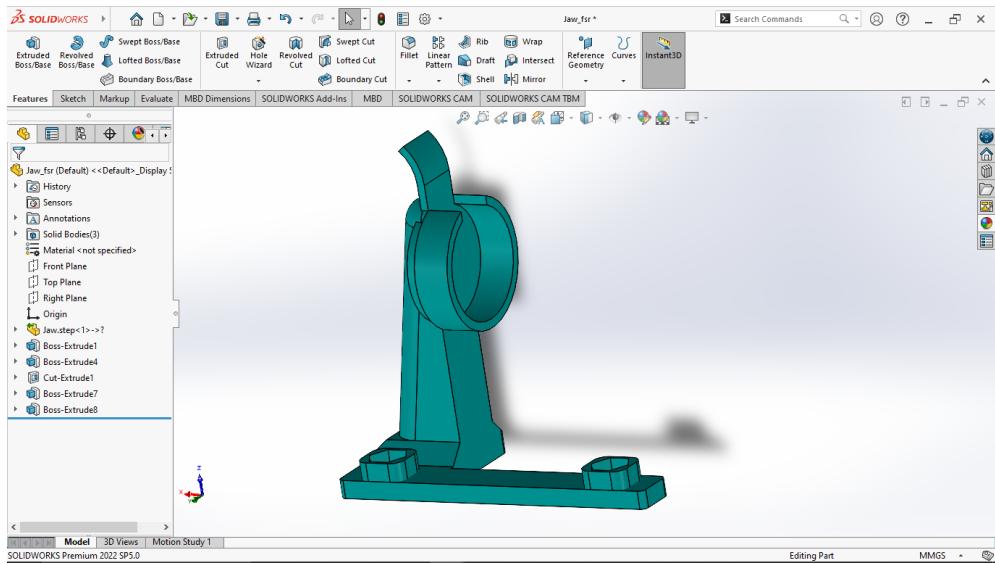
1. ابعاد کلی سازه و منطبق بودن با ابعاد مورد نیاز در دستور کار .
2. ایجاد پایه ها برای نصب lcd 16\*2 .
3. ایجاد پایه ها برای نصب ادیونو UNO R3 .
4. ایجاد 4 بازو برای نصب پایه های عمودی زیر ان تا بتوان کل مکانیزم را روی میز قرار داد .
5. ایجاد پایه های سررو متناسب با ابعاد ان .
6. رعایت کردن ضخامت های مورد نیاز در نواحی که احتمال بالا رفتن تنش وجود دارد .
7. ایجاد ریل ها با ابعاد مناسب برای قرار گیری رک ها .
8. در کل طراحی سعی شده از کمترین حجم استفاده شود تا هنگام پرینت کمترین مقدار فیلامنت مصرف شود .
9. این سازه هنگام پرینت به هیچ عنوان نیاز به ساپورت ندارد .



تصویر ۱۳- طراحی بنده گریپر

## مرحله سوم

ایجاد گیره اولیه:



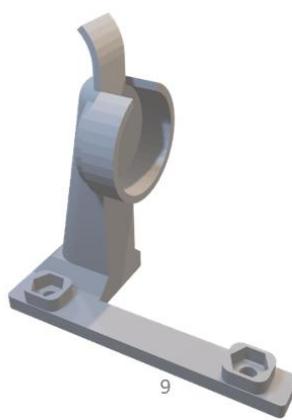
تصویر14- طراحی اولیه گیره

این گیره اول میباشد که با جانمانیی دقیق محل پیچ ها و مهره ها به رک متصل میشود.

مهم ترین نکته در این قسمت ایجاد یک محفظه متناسب با ابعاد دقیق سنسور fsr-402 است.

با قرار گیری سنسور در این قسمت و قرار گرفتن گیره نهایی روی ان میتوان تمام نیروی وارد شده از سمت جسم را به سنسور اعمال کرد تا خطایی نداشته باشیم.

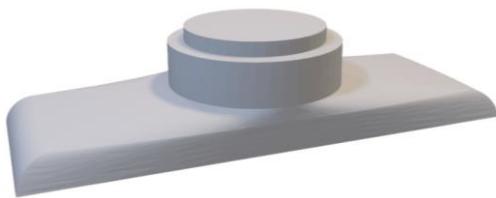
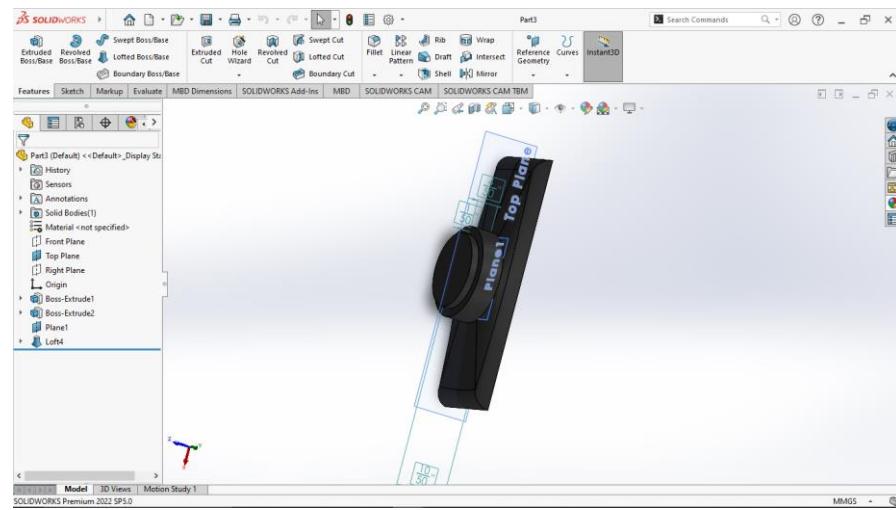
همچنین در این قسمت ملاحظات لازم برای کاهش تنش، جاگیری درست سنسور و سیم ان برای جلوگیری از پاره شدن سیم سنسور، ایجاد محفظه قرار گیری مهره، در نظر گرفته شده است.



تصویر15- طراحی نهایی گیره

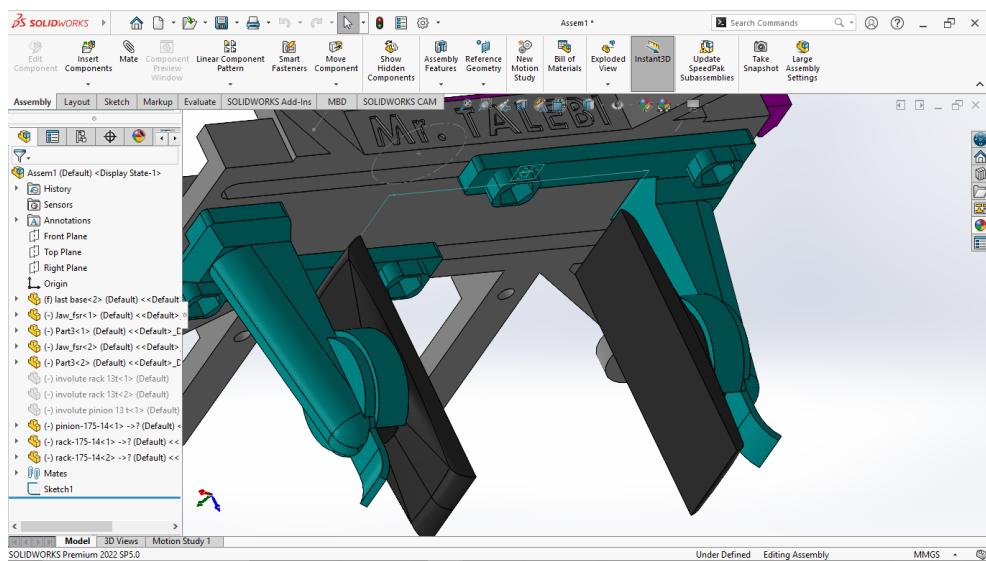
## مرحله چهارم

این بخش مربوط به طراحی گیره نهایی میباشد که از یک سمت با جسم و از سمت دیگر به همراه یک چسب دو طرفه تنها با سطح سنسور نیروسنج تماس دارد.



تصویر ۱۶- گیره نهایی

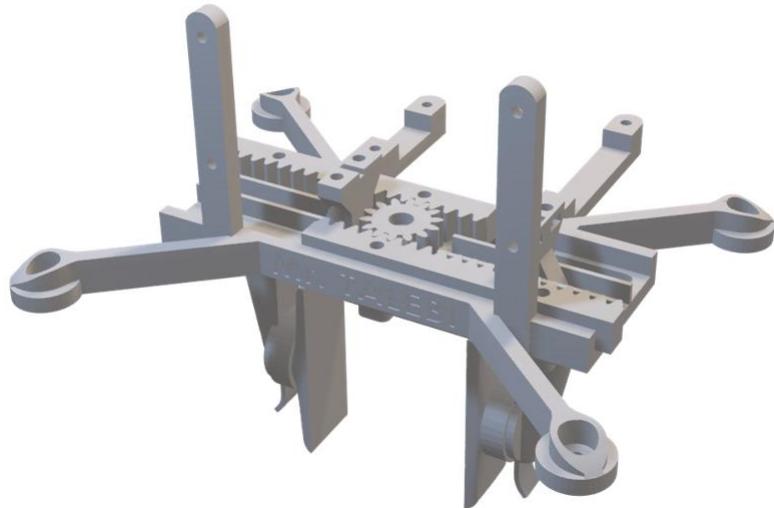
این قطعه مانند یک پیستون در محفظه سنسور قرار میگیرد و تمام نیروی حاصل از نگه داشتن اجسام به ارتفاع تماش ۵ سانتی متر را به سنسور منتقل میکند .



تصویر ۱۷- نمایش عملکرد گیره

بین سیلندر و پیستون این قسمت فاصله ۰.۳ میلی متری وجود دارد تا اصطکاک جلوگیری کند و همینطور گیره به راحتی خارج نشود .

در نهایت با اسمبل کردن این قطعات میتوان ساختار نهایی این گریپر را مشاهده کرد .



تصویر18- خروجی نهایی

#### مرحله پنجم(ساخت)

برای ساخت این قطعات مسلما بهترین گرینه پرینتر 3 بعدی میباشد .

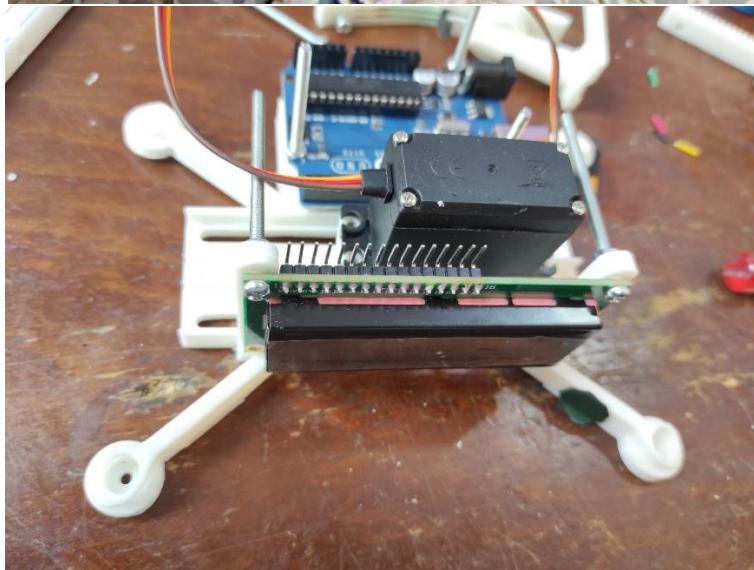
هگام پرینت این قطعات نیاز بود نکاتی را رعایت کرد:

1. جنس فیلامنت انتخاب شده PLA میباشد . ( بهتر اسب به دلیل استحکام کم از ABS استفاده نشود . )
2. تنها پارتی که برای پرینت ان نیاز به ساپورت هست ، پارت jaw2 میباشد .
3. برای باقی پارت ها به هیچ عنوان نیازی به استفاده از ساپورت نیست و درصورت فعل بودن گرینه ساپورت گذاری میتواند باعث ایجاد ضایعات ناخواسته ای شود که قرار گیری مهره ها را سخت میکند .
4. برای پرینت رک و پینیون پیشنهاد میشود از نازل با قطر 0.3 میلی متر یا در صورت امکان کوچک تر ، استفاده شود تا دقیق داشته باشد و حتما با چگالی 100 % پرینت شود .
5. برای پرینت باقی پارت ها نازل با قطر 0.3 تا 0.5 میلی متر مناسب است اما بیشتر از این بهتر است استفاده نشود . و چگالی 80 % مناسب میباشد .
6. ظحامت های کمتر از 3 میلی متر باید دیواره فرض شوند و با 100% چگالی پرینت شوند .

برای پرینت ما تمام این نکات را رعایت کردیم .

مدت پرینت حدود 5 ساعت بود . مقدار 94 گرم فیلامنت مصرف شد .

در نهایت میتوان گریپر ساخته شده را در این تصاویر مشاهده کرد :



تصویر 19- گریپر پس از ساخت

## اجزای سیستم

طبق فایل قوانین پروژه، تنها مجاز به استفاده از وسایل ذکر شده بودیم، که در ادامه هر وسیله و عملکرد آن را شرح خواهیم داد.

### معرفی قطعات و روش های کالیبراسیون

- آردوینو UNO: 1

ما برای کنترل کردن اجزای سیستم از بورد آردوینو UNO R3 استفاده کردیم مشخصات و ویژگی های آن در تصویر زیر آمده است.

Microcontroller	ATmega328v
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)v	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

تصویر 20-مشخصات بورد آردوینو

### 2- سرووموتور

این سرو دیجیتال MG996R با گشتاور توقف بسیار بالای 10 کیلوگرم و دندنهای فلزی، در یک بسته کوچک ارائه می شود . نسخه بهبودیافته سروی سروروی معروف MG995 است.

مشخصات سرووموتور طبق دیتاشیت:

گشتاور توقف(Stall torque) : 9.4 کیلوگرم سانتی متر (4.8 ولت) و 11 کیلوگرم سانتی متر (6 ولت)

سرعت عملکرد:(Operating speed): 0.17 ثانیه برای 4.8 ولت و 0.14 ثانیه برای 60 درجه (6 ولت)

ولتاژ عملیاتی:(Operating voltage): 4.8 ولت تا 7.2 ولت

✓ ما برای تست ها از یک آداتور 6 ولت و 2.5 آمپر استفاده کردیم.

جریان هنگام کار: (Running Current): 500 میلی آمپر تا 900 میلی آمپر (6 ولت)

جریان توقف: (Stall Current): 2.5 آمپر (6 ولت)

پهنهای باند مرده: (Dead band width) 5 میکروثانیه

محدوده دمایی (Temperature range): از 0 درجه سانتی گراد تا 55 درجه سانتی گراد

در ابتدای استفاده از سروو جهت تست صحت عملکرد آن، با افزایش درجه به درجه سروو از آن تست گرفتیم.

```
1 #include <Servo.h>
2
3 Servo myServo; //
4
5 void setup() {
6     myServo.attach(9);
7     Serial.begin(9600);
8 }
9
10 void loop() {
11     // حرکت آرام از 0 تا 180
12     for (int angle = 0; angle <= 180; angle++) {
13         myServo.write(angle);
14         Serial.print("زاویه: ");
15         Serial.println(angle);
16         delay(15);
17     }
18     delay(500);
19 }
20
```

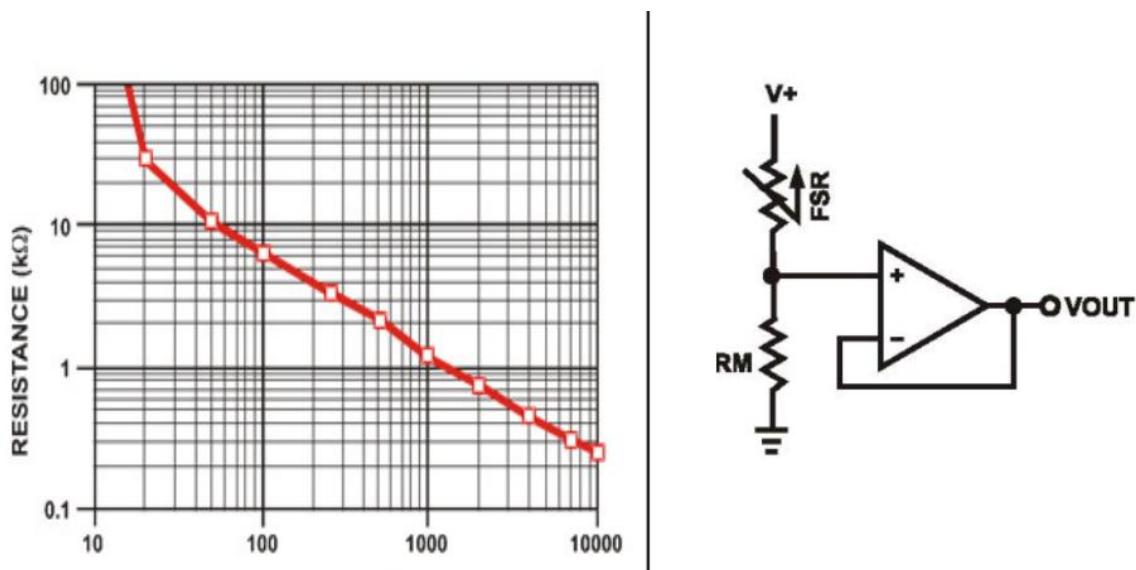
تصویر 21- تست سروو

طبق کد بالا زاویه سروو را یک درجه-یک درجه تغییردادیم، و متوجه شدیم از زاویه 176 به بعد سروو حرکت نمیکند. بنابراین بازه حرکتی آن را 0 تا 176 در نظر گرفتیم. و برای طراحی پینینون بازه 0 تا 175 را مدنظر قرار دادیم.

- 3- سنسور FSR، یا همان مقاومت‌های حساس به نیرو، دستگاه‌های مقاوم مبتنی بر فیلم پلیمری ضخیم (PTF) هستند که با افزایش نیروی واردشده به سطح سنسور، مقاومت آن‌ها کاهش می‌یابد. مشخصات این سنسور طبق دیتاشیت به شکل زیر است.

Feature	Condition	Value*	Notes
<b>Actuation Force</b>		0.1 Newtons	
<b>Force Sensitivity Range</b>		0.1 - $10.0^2$ Newtons	
<b>Force Repeatability<sup>3</sup></b>	(Single part)	$\pm 2\%$	
<b>Force Resolution<sup>3</sup></b>		continuous	
<b>Force Repeatability<sup>3</sup></b>	(Part to Part)	$\pm 6\%$	
<b>Non-Actuated Resistance</b>		10M W	
<b>Size</b>		18.28mm diameter	
<b>Thickness Range</b>		0.2 - 1.25 mm	
<b>Stand-Off Resistance</b>		>10M ohms	Unloaded, unbent
<b>Switch Travel</b>	(Typical)	0.05 mm	Depends on design
<b>Hysteresis<sup>3</sup></b>		+10%	$(R_{F+} - R_{F-})/R_{F+}$ .
<b>Device Rise Time</b>		<3 microseconds	measured w/steel ball
<b>Long Term Drift</b>		<5% per $\log_{10}$ (time)	35 days test, 1kg load
<b>Temp Operating Range</b>	(Recommended)	-30 - +70 °C	
<b>Number of Actuations</b>	(Life time)	10 Million tested	Without failure

تصویر 22- مشخصات FSR



تصویر 23- شماتیک سنسور FSR و نمودار تغییرات

بررسی سنسور FSR از لحاظ عملکرد و دقت:

مقاومت یک حسگر نیروی مقاومتی (**FSR**) با افزایش نیروی اعمال شده کاهش می‌یابد. سنسورهای **FSR** معمولاً در یک مدار تقسیم ولتاژ همراه با یک مقاومت کشیده شده به زمین ( $R_f$ ) استفاده می‌شوند تا تغییرات مقاومت به ولتاژ قابل اندازه‌گیری تبدیل شود. که رابطه آن به شکل زیر است. در اینجا  $R$  مقاومت سنسور است و  $R_f$  مقاومت  $up$  است و  $V_{out}$  ای که ما متصل می‌کنیم.

$$V_{out} = V_{in} \times \frac{R_{FSR}}{R_{FSR} + R_f}$$

دقت سنسورهای **FSR**، مانند سری 400، به طور کلی متوسط است و برای کاربردهایی که نیاز به اندازه‌گیری تقریبی نیرو یا فشار دارند، مناسب است.

#### 1. غیرخطی بودن پاسخ:

- رابطه‌ی مقاومت با نیروی وارد شده غیرخطی است. به این معنا که تغییرات مقاومت به صورت یکنواخت با افزایش نیرو تغییر نمی‌کند.
- اغلب باید از یک جدول کالیبراسیون یا معادله‌ی تجربی برای تبدیل مقاومت به نیرو استفاده کنید.

#### 2. حساسیت به محدوده نیرو:

- در نیروی پایین (کمتر از 1 نیوتن) و نیروی بسیار زیاد (بالاتر از محدوده مشخص)، دقت سنسور کاهش می‌یابد.
- این سنسورها بیشتر برای نیروی متوسط (1 تا 10 نیوتن) مناسب هستند.

#### 3. تکرارپذیری:

- تکرارپذیری اندازه‌گیری نیرو با این سنسورها حدود  $\pm 10\%$  است.
- این یعنی اگر چند بار نیروی یکسانی اعمال شود، ممکن است خروجی کمی متفاوت باشد.

#### 4. پایداری در طول زمان:

- مقاومت سنسور ممکن است در طول زمان و پس از استفاده مکرر تغییر کند، که می‌تواند دقت اندازه‌گیری را کاهش دهد.

موارد بالا باعث شد تا در صدد آن باشیم تا قبل از استفاده از سنسور **FSR** آن را کالیبره کنیم. برای اینکار از یک مقاله الهام گرفتیم.<sup>[4]</sup> در این مقاله کالیبراسیون و مشاهده عملکرد سنسور بر اساس به دست آوردن رابطه بین خروجی دیجیتال سنسور و مقدار واقعی نیرو مورد ارزیابی قرار گرفت. و ما نیز از این روش برای استفاده کردیم.

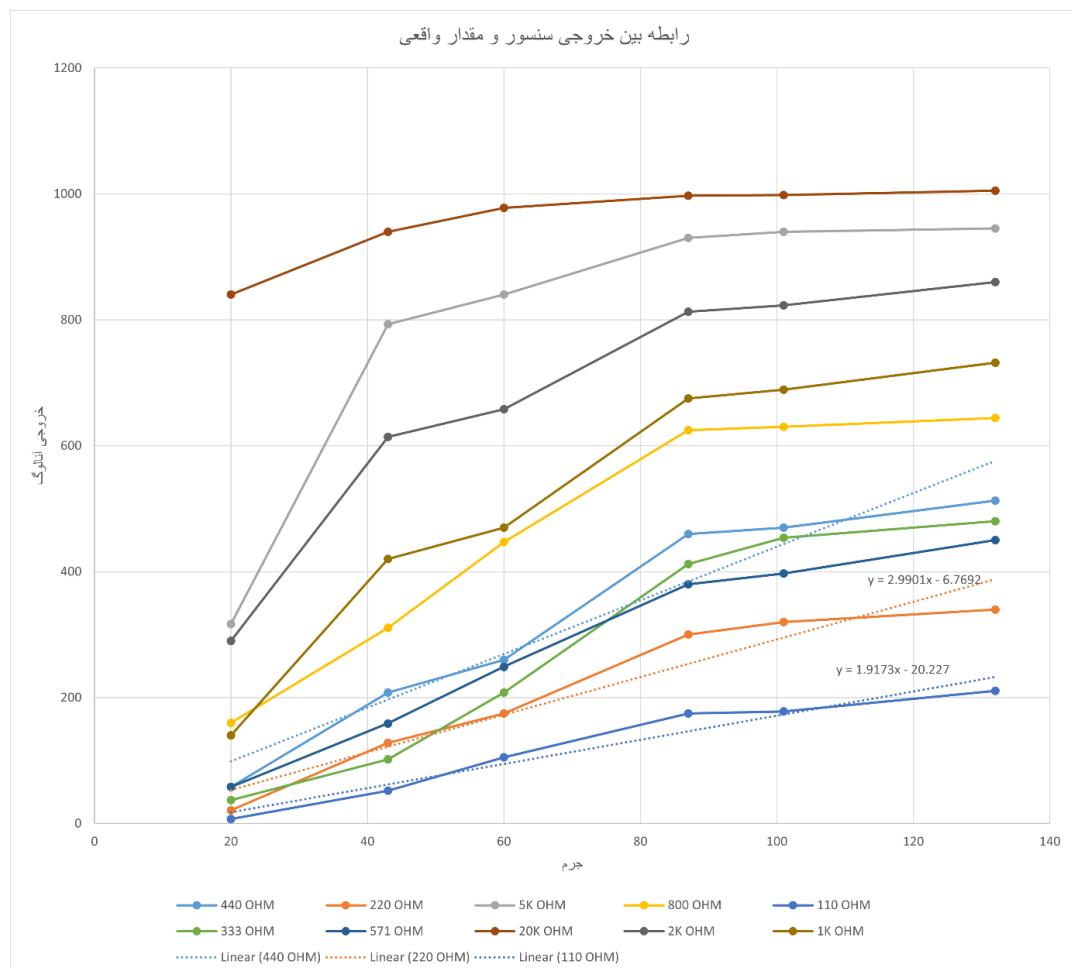
ما در این تست، با فراهم کردن شرایط واقعی برای سنسور و اتصال آن به گریپر با قرار دادن اجرامی با وزن‌های متفاوت رابطه بین خروجی با مقدار واقعی را به دست آوردیم. همچنین این تست را برای مقاومت‌های سری متفاوت تست کردیم تا بتوانیم از بهترین و دقیق‌ترین مقاومت سری استفاده کنیم.

در ابتدا جرم چندین جسمه نمونه به وسیله یک ترازو سنجیده شد سپس برای تست **FSR** از آنان استفاده شد.



تصویر ۲۴- اجسام نمونه

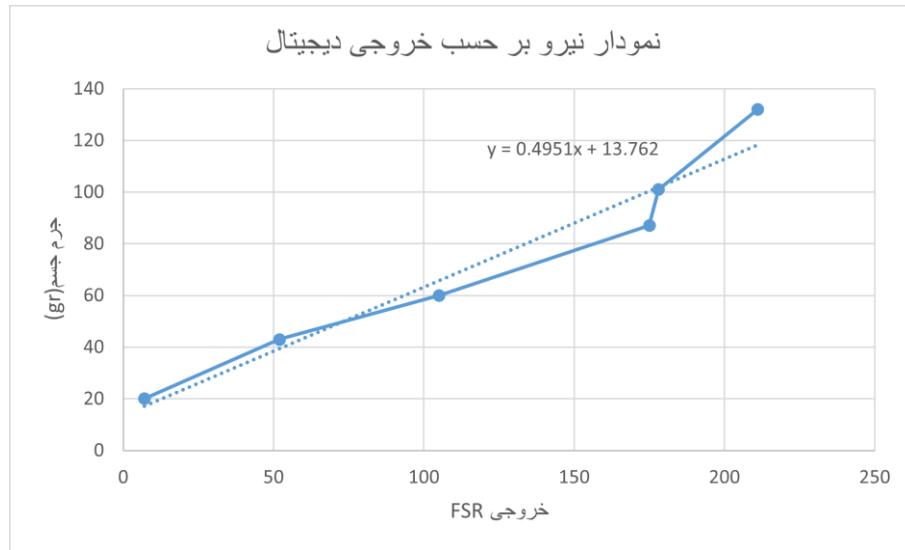
سپس این تست برای ۱۰ مدل مقاومت مختلف امتحان شد تا به بهترین و خطی ترین نمودار ممکن برسیم. که نتایج به شکل زیر است.



نمودار ۱- مقایسه مقاومت های مختلف برای کالیبراسیون FSR

به طور قطعی، آنچه برای ما مطلوب است خطی ترین رابطه ممکن برای سنسور است، که دو مقاومت 220 و 110 بهترین ها هستند، در ابتدا مقاومت 220 به دلیل شیب خط بیشتر چه تمایز بیشتر و گستره بیشتری به ما می دهد انتخاب شد، اما پس از چندین بار تست، متوجه شدیم در نیرو های بالا، و درست در نقاطی که رابطه خطی نمی باشد (در انتهای خط) نیروی تخمین زده خطی زیادی دارد ف در حالی که نیروهای کمتر دقت مطلوبی دارند. بنابراین تصمیم گرفتیم تا مقاومت را تعویض کنیم و از مقاومت 110 اهمی استفاده کنیم که خطی غیرخطی کمتری دارد. سپس از طریق اکسل trend line نمودار را مشخص میکنیم، معکوس شیب خط(تبديل سیگنال به نیرو) مقداری برابر با 0.4951 است که در صورت تبدیل جرم به نیرو این ضریب برابر با 0.0051 میشود و بیانگر رابطه زیر است.

$$\text{مقدار نیرو} = \text{خروجی سنسور} \times 0.0051$$



نمودار ۲- رابطه مقاومت 110 اهم با نیرو

که پس از به دست آوردن نسبت بین این دو، مجدد چند جسم را تست کردیم که نتایج در جدول زیر قرار دارد. همانطور که مشاهده می شود دقت سنسور تا حد بسیار خوبی بالا رفته و خطای بسیار کمی دارد.

جدول ۱- تست کالیبراسیون سنسور

	FSR	تخمین جرم	جرم حقیقی جسم
20		19.4	
43		43.1	
60		59.0	
101		103.4	
133		119.2	

حالا از کد زیر هم برای کالیبره کردن سنسور استفاده شد. که در ابتدا با میانگین گیری از چند داده اول مقدار base line را مشخص میکند و خروجی را 0 میکند و با استفاده از ضریب مقادیر نیرو را اعلام میکند.

```

1  const int fsrPin = A0; // پایه آنالوگ
2  float calibrationFactor = 0.00511; // خریب کالیبراسیون - برآسانس تست و تنظیم شده
3
4  int fsrBaseline = 0; // محاسبه می‌شود که در (tare) مقدار مبنای
5
6  void setup() {
7      Serial.begin(9600);
8      Serial.println("Calibrating the sensor...");
9
10     // در نظر بگیریم baseline چند بار سنسور را می‌خوانیم تا یک میانگین بگیریم و به عنوان
11     const int numReads = 200;
12     long sum = 0;
13     for (int i = 0; i < numReads; i++) {
14         sum += analogRead(fsrPin);
15         delay(30);
16     }
17     fsrBaseline = sum / numReads; // میانگین خوانده شده
18     Serial.print("Baseline value is: ");
19     Serial.println(fsrBaseline);
20     Serial.println("Calibration done. Starting main loop...");
21     delay(1000);
22 }
23
24 void loop() {
25     // خواندن مقدار سنسور
26     int fsrReading = analogRead(fsrPin);
27
28     // برای صفر شدن در حالت بدون تیرو تقریق مقدار
29     int fsrOffset = fsrReading - fsrBaseline;
30
31
32     // محاسبه وزن (گرم) از مقدار جبرانشده با baseline
33     float weightGram = fsrOffset * calibrationFactor;
34
35     Serial.print("ADC Raw: ");
36     Serial.print(fsrReading);
37     Serial.print(" | offset: ");
38     Serial.print(fsrOffset);
39     Serial.print(" | Weight: ");
40     Serial.print(weightGram);
41     Serial.println(" g");
42
43     delay(200);
44 }
45

```

تصویر 25- کد کالیبراسیون سنسور

## فیلتر کردن داده های سنسور

فیلتر کردن سیگنال یکی از گامهای اساسی در پردازش داده های سنسوری است. به دلیل وجود نویز در داده های خام سنسور (مانند FSR)، خروجی می تواند ناپایدار یا غیرقابل اعتماد باشد. استفاده از فیلترها به ما کمک می کند تا داده هایی صاف تر و پایدار تر برای تحلیل یا کنترل سیستم داشته باشیم.

ما در کدها، از فیلتر **Moving Average** استفاده کردیم تا داده های خروجی سنسور FSR صاف تر و قابل اعتماد تر شوند. این فیلتر به این صورت عمل می کند که 50 داده اخیر سنسور را جمع آوری کرده و میانگین آنها را به عنوان خروجی بازمی گرداند. این روش باعث کاهش اثر نویز می شود و داده های نهایی به شرایط واقعی نزدیک تر است.

## لیست هزینه قطعات

جدول 2- هزینه

ردیف	عنوان	قیمت (تومان)
1	سرموتور MG996R	200,000
2	SENSOR FSR	156,000
3	آردوینو UNO R3	340,000

40,000	سیم جامپر	<b>4</b>
60,000	LCD 2*16	<b>5</b>
30,000	مقاومت و سیم جامپر	<b>6</b>
300,000	پرینت گریپر	<b>7</b>
80,000	باتری (8 عدد قلمی)	<b>8</b>
1,205,000		<b>مجموع</b>

## چالش ها

ما در مراحل این فصل، چالش های بسیار زیادی داشتیم که در ادامه راه حل مرتفع کردن آن ها شرح داده خواهد شد.

- 1 ما در ابتدا تست های سنسور نیرو را با قرار دادن بر روی میز، و گذاشتن اجسام بر روی فک گریپر انجام دادیم و ضریب کالیبراسیون را به دست آوردیم. اما پس از راه اندازی کامل گریپر متوجه شدیم، شرایط میز با شرایط گریپر متصل به بدن و بدون تکیه گاه متفاوت است و تفاوت زیادی در نیرو ها دارد.
- ✓ راه حل: ب ایکسان سازی شرایط گریپر و محیط تست، تمامی مراحل کالیبراسیون را مجدد انجام دادیم.
- 2 وجود دیواره قسمت مشخص شده در شکل زیر باعث عدم انتقال تمام نیرو به سنسور و ایجاد خطأ میشد.



- ✓ برش و جدا کردن کامل دیواره، هرچند چالش های دیگری در مهار لقی فک برای ما ایجاد کرد اما نهایتاً باعث شد دقیق به طرز قابل توجهی بهتر شود.
- 3 نحوه و جنس اتصال چسبی سنسور به بدن در انتقال نیرو به FSR و شرایط تست و خروجی ها بسیار تاثیرگذار بود. دقیق سنسور در اتصال با چسب قطره ای و یا چسب برق و یا چسب ژله ای بسیار متفاوت و پر خطأ بود.
- ✓ با چندین بار تست در حالت های مختلف اتصالات، نهایتاً بهترین خروجی با اتصال سنسور به وسیله چسب خودش به بدن و چسب دوطرفه به فک گریپر حاصل شد.

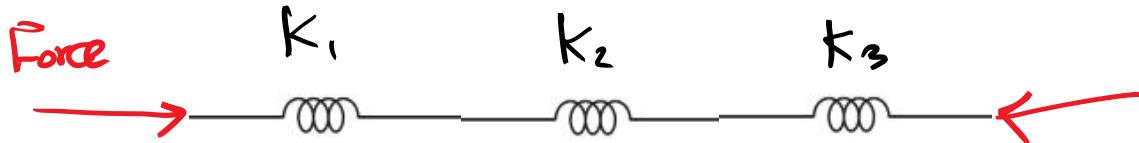
## پیاده سازی کنترل و تست

### چالش اجسام با جنس مختلف

یکی از چالش هایی که ما در پروژه با آن روبه رو بودیم، تشخیص نیروی مناسب برای نگه داشتن اجسام مختلف با جرم و سفتی مختلف بود. با الهام از یک مقاله که بازاره بندی سفتی ها به هر کدام نیروی خاصی اختصاص داده بود، روشی برای به دست آوردن سفتی اجسام طراحی کردیم تا با به دست آوردن اختلاف نیرو و اختلاف جا به جایی، گریپر ابتدا ضربه سفتی اجسام مختلف را به دست بیاورد و سپس نیروی مناسب برای نگه داشتن جسم را تخمین بزند [5]. رابطه سفتی جسم هم به صورت زیر است:

$$K = F/X \quad \text{و} \quad F = K^* X$$

\* نکته: باید توجه داشت سفتی که ما به دست می آوریم، سفتی سه جسم به صورت سری است. اگر هر کدام از دو فک گریپر و جسم را فر در نظر بگیریم، سفتی که ما محاسبه میکنیم در واقع سفتی  $K_{tot} = K_1 + K_2 + K_3$  است. که چون این ضرایب فنریت در هر بار گرسپینگ حضور دارند، ما  $K_{tot}$  را به عنوان سفتی جسم که مورد بررسی قرار می دهیم در نظر میگیریم. نکته قابل توجه این است که در این شرایط حتی اگر جسم وسط ما کاملاً صلب بود در اثر جابه جایی خیلی کوچک (یک میلی متر) برای تعیین سفتی، فنریت گریپر مانع از آسیب رسیدن به سروو یا جسم میشود.



بنابراین کدی نوشته شد تا مراحل زیر را طی کند:

- 4- ابتدا گریپر از بازترین حالت کم کم بسته شود.
- 5- زمانی که نیروی سنسور از صفر به یک حد معینی که بیانگر تماس با جسم است (ما  $0.1\text{ N}$  در نظر گرفتیم) تشخیص بدهد که در حال تماس با جسم برای نگه داشتن است.
- 6- سپس به آرامی و مرحله به مرحله 4 درجه افزایش زاویه دهد، این تغییر زاویه منتج به تغییر جابه جایی به میزان  $1.188\text{ میلی متر}$  می شود. (طبق رابطه زاویه سروو با رک و پینیون) و با داشتن جا به جایی برحسب زاویه سروو و بازه گریپر، و با داشتن اختلاف نیروی گزارش شده توسط FSR مقدار سفتی به دست می آید.
- 7- در این قسمت نیروی مناسب برای نگه داشتن را پیدا میکنیم. اگر گریپر نیروی کمی داشت با ارسال  $+$  در سریال مانیتور گریپر بسته میشود و بر عکس.
- 8- این تا زمانی ادامه پیدا میکند که نیرو به حد مطلوب برسد. (حد مطلوب نیرو به صورت تجربی احساس میشود)
- 9- زمانی که نیرو به حد مطلوب رسید، نیرو و سفتی را یادداشت کرده و فرآیند را تمام میکنیم.

```

1 #include <Servo.h>
2
3 const int fsrPin = A0;           // بین سنسور
4 int fsrBaseline = 0;            // مقدار میتا
5 const float fsrFactor = 0.00511; // ADC فریب تبدیل اختلاف //
6
7 // تابعی برای ذخیره و مرتبسازی داده های ADC
8 // و گرفتن میانگین پس از حذف چند نمونه افزایی //
9 void getStableForce() {
10    const int numSamples = 50;   // تعداد نمونه هایی که می خواهیم بگیریم
11    int readings[numSamples];
12
13    // نمونه بردازی
14    for(int i = 0; i < numSamples; i++){
15        readings[i] = analogRead(fsrPin);
16        delay(5); // فاصله بین هر نمونه بردازی 5ms
17    }
18
19    // بروش دلخواه (bubble sort) به روشن ساخته شده
20    for(int i = 0; i < numSamples - 1; i++){
21        for(int j = 0; j < numSamples - i - 1; j++){
22            if(readings[j] > readings[j+1]) {
23                int temp = readings[j];
24                readings[j] = readings[j+1];
25                readings[j+1] = temp;
26            }
27        }
28    }
29
30    // حذف 2 نمونه کمینه و 2 نمونه بیشینه //
31    // برای کاهش تاثیر نویز و داده های پرت)
32    long sum = 0;
33    int count = 0;
34    for(int i = 4; i < numSamples - 2; i++){
35        sum += readings[i];
36        count++;
37    }
38
39    // میانگین نمونه های میانی
40    float avgReading = sum / (float)count;
41
42    // کم کردن خط پایه (Baseline)
43    float rawValue = avgReading - fsrBaseline;
44    if(rawValue < 0) {
45        rawValue = 0;
46    }
47
48    // محاسبه نیرو (نیوتون)
49    float force = rawValue * fsrFactor;
50    return force;
51 }
52
53 Servo myServo;           // شیء سرو
54 int servoAngle = 0;       // زاویه سرو (0 تا 180 درجه)
55
56 // متغیرهایی برای تست سفتی //
57 bool stiffnessTestDone = false;
58 float initialForce = 0.0; // 0.1 نیروی لحظه اولی که به می رسم
59 float finalForce = 0.0;   // نیروی پس از 4 درجه حرکت اضافه
60 float stiffnessValue = 0.0; // مقدار سفتی
61 float deltaForce = 0.0;   // (finalForce - initialForce)
62
63 // =====
64 // تابع کالیبراسیون سنسور
65 // =====
66 void calibrateFSR(int pin) {
67     Serial.println("Calibrating the sensor...");
68
69     // در نظر بگیریم baseline چند بار سنسور را می خوانیم تا یک میانگین بگیریم و به عنوان
70     const int numReads = 200;

```



```

138 stiffnessValue = deltaForce / displacement;
139
140     گزارش سفتی و اختلاف نیرو // 
141 Serial.println("=====");
142 Serial.print("N) نیروی اولیه: ");
143 Serial.println(initialForce, 3);
144 Serial.print("N) نیروی ثانویه: ");
145 Serial.println(finalForce, 3);
146 Serial.print(" اختلاف نیرو (DeltaF) = ");
147 Serial.print(deltaForce, 3);
148 Serial.println(" N");
149 Serial.print(" مقدار سفتی جسم (Stiffness) = ");
150 Serial.print(stiffnessValue, 3);
151 Serial.println(" N/m");
152 Serial.println("=====");
153
154 // اعلام پایان تست
155 Serial.println("(-). تست سفتی تمام شد. منتظر فرمان های سریال هستیم (+ یا
156
157 حال دیگر تست تمام شده است و وارد مرحله کنترل دستی خواهیم شد
158 stiffnessTestDone = true;
159 }
160 else {
161     در این مرحله، تست انجام شده و فقط منتظر دستورات از سریال هستیم
162     // اگر '+' دریافت شود -> یک درجه بینند
163     // اگر '-' دریافت شود -> یک درجه باز کن
164     // سپس نیرو را بخوان و گزارش کن
165     if (Serial.available() > 0) {
166         char command = Serial.read();
167
168         if (command == '+') {
169             servoAngle++;
170
171             if (servoAngle > 180) {
172                 servoAngle = 180; // جلوتر از 180 نرود
173             }
174             myServo.write(servoAngle);
175
176             نیرو را بخوانیم و نمایش دهیم
177             float force = getStableForce();
178             Serial.print(" زاویه: ");
179             Serial.print(servoAngle);
180             Serial.print(" درجه، نیرو: ");
181             Serial.print(force, 3);
182             Serial.println(" نیوتون");
183
184         else if (command == '-') {
185             servoAngle--;
186             if (servoAngle < 0) {
187                 servoAngle = 0; // نرود
188             }
189             myServo.write(servoAngle);
190
191             نیرو را بخوانیم و نمایش دهیم
192             float force = getStableForce();
193             Serial.print(" زاویه: ");
194             Serial.print(servoAngle);
195             Serial.print(" درجه، نیرو: ");
196             Serial.print(force, 3);
197             Serial.println(" نیوتون");
198
199         }
200     }
}

```

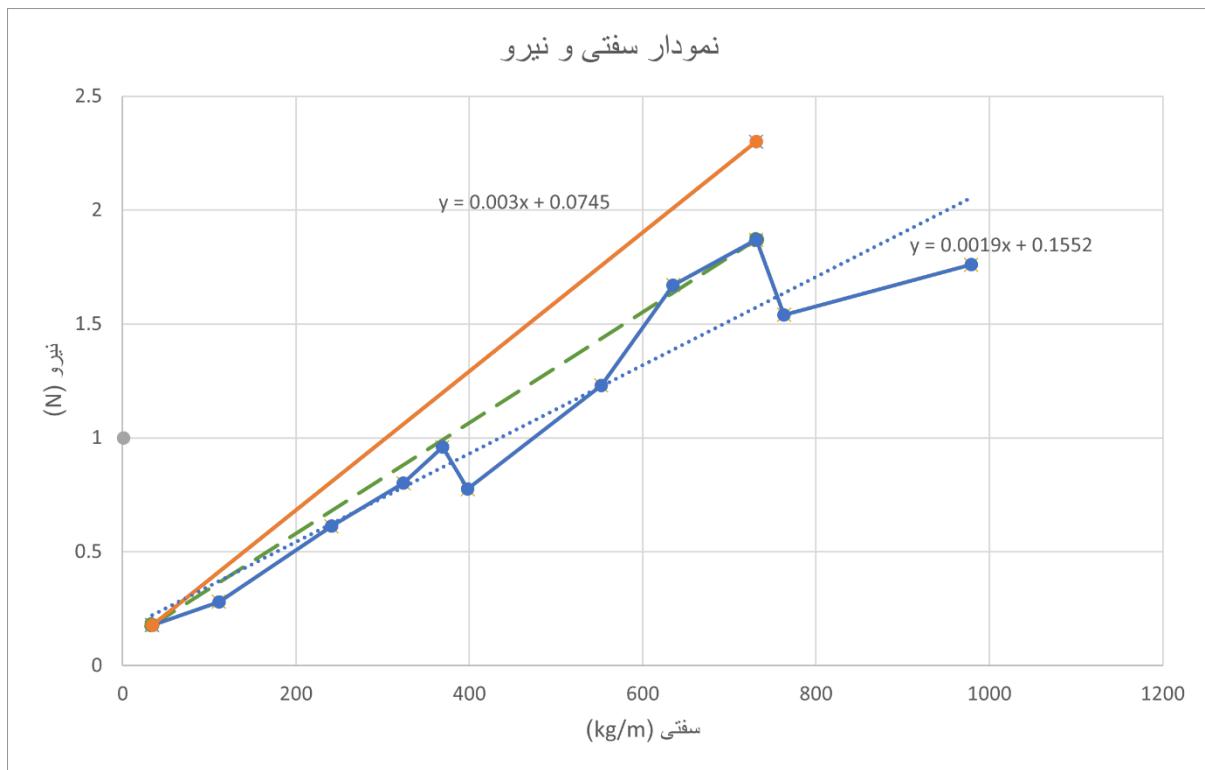
با استفاده از این تست، سفتی و نیروی مطلوب برای 10 جسم را محاسبه میکنیم و با خطی سازی ارتباط بین نیرو و سفتی رابطه بین این دو را به دست می آوریم تا در کنترل برای تخمین نیروی هدف از آن استفاده کنیم.



تصویر 27 - اجسام نمونه

جدول 2- سفتی و نیروی اجسام

جسم	softi (kg/m)	نیروی مناسب(N)
لیوان کاغذی	34	0.178
واکس	111	0.28
لواشک	241	0.612
قوطی ابمیوه	324	0.8
در اسپری	369	0.959
ته لیوان	398	0.775
تخته پاک کن	552	1.23
توپ	635	1.67
تخم مرغ	731	1.87
باتری	763	1.54
ایرپاد	979	1.76



نمودار ۳- سفتی و نیرو

طبق نمودار شیب خطی که از رگرسیون نمودار به دست آمده است، نمیتواند برای همه اجسام مناسب باشد. به طور مثال، نمودار زیر نقطه قله می‌افتد، بنابراین نمی‌تواند به خوبی آن جسم را نگه دارد. از آنجا که گرفتن اجسام با نیروی کمی بیشتر بهتر از رها شدن جسم است، تصمیم گرفتیم تا خط جدید را بر قله مماس کنیم. (در نمودار با رنگ سبز نمایش داده می‌شود). اما در طی تست هایی که میگرفتیم متوجه شدیم همچنان بازه‌ی نوسان نیرو برای جسم‌های سنگین مناسب نیست. بنابراین شیب خط و ضریب تبدیل سفتی به نیرو را بالاتر بردیم. (تصویر نارنجی). و در نهایت رابطه بین سفتی و نیرو ۰.۰۰۳ شد.

حال برای آنکه کنترلر دچار نوسان نشود، به جای یک نیروی هدف، یه بازه هدف درنظر گرفتیم. حد بالا ۱۱۰٪ نیروی هدف و حد پایین ۹۰٪. در صورتی که نیروی فیدبک به این بازه رسید، به هدف رسیده و متوقف می‌شود. اما اگر چنین نباشه و نیروی هدف تنها یک نقطه باشد، با توجه به دقت پایین سنسور، نوسانات زیاد حول نیروی هدف سیسیتم را به شدت ناپایدار می‌کند.

### روش کنترل

پس از بررسی روش‌های معمول برای کنترل گریپر تصمیم گرفتیم از کنترل PID استفاده کنیم. چرا که:

- پیچیدگی نسبتاً پایین دارد
- کنترل نیرو را با دقت انجام میدهد
- قابلیت انعطاف (با تغییر ضرایب) را دارد.

کنترل **PID** یکی از رایج‌ترین روش‌های کنترل در سیستم‌های مکانیکی و الکترونیکی است. این کنترلر با دریافت خطای سیستم (اختلاف بین مقدار هدف و مقدار اندازه‌گیری شده) و پردازش آن، سیگنال کنترلی مناسبی تولید می‌کند تا سیستم به مقدار مطلوب برسد.

مفهوم کنترل PID :

کنترل PID از سه بخش تشکیل شده است:

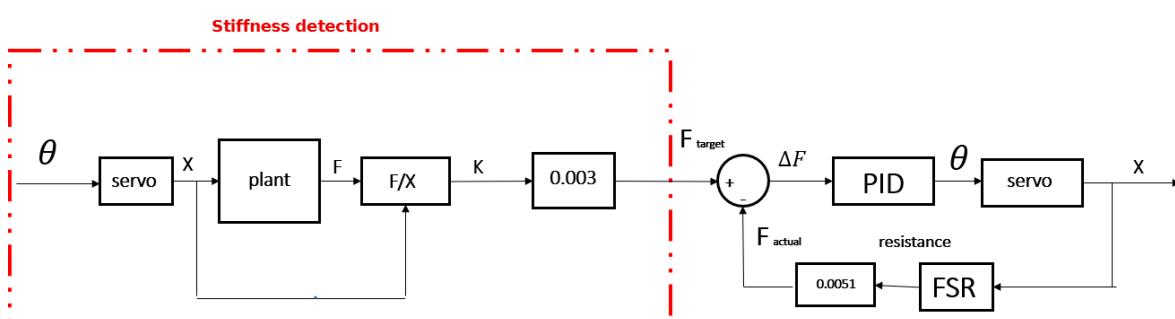
- کنترل تناوبی (Proportional - P): متناسب با مقدار خطای فعلی، یک سیگنال کنترلی تولید می‌کند مقدار و خطای در نظر می‌گیرد.
- کنترل انتگرالی (Integral - I): مجموع خطاهای قبلی را در نظر می‌گیرد تا از خطای ماندگار جلوگیری کند و تغییرات تجمعی خطای در نظر می‌گیرد.
- کنترل مشتقی (Derivative - D): ترخ تغییر خطای را بررسی کرده و نوسانات را کاهش می‌دهد. و سرعت تغییرات خطای در نظر می‌گیرد.

رابطه ریاضی:

$$\frac{de(t)}{dt}dK + K_i \int e(t)dt + K_p e(t) = u(t)$$

- خطای لحظه‌ای است.
- ضریب تناوبی  $K_p$  است.
- ضریب انتگرالی  $K_i$  است.
- ضریب مشتقی  $K_d$  است.
- سیگنال خروجی کنترل است.

رونده گرفتن جسم به این صورت خواهدند طبق مراحلی در قسمت تشخیص سفتی گفته شد، ابتدا سفتی جسم را تشخیص دهد و سپس با کنترل PID نیرو را به حد مطلوب برساند. دیاگرام جعبه‌ای این روش به شکل زیر می‌باشد.



تصویر 28- دیاگرام جعبه‌ای کنترل

### شبیه سازی

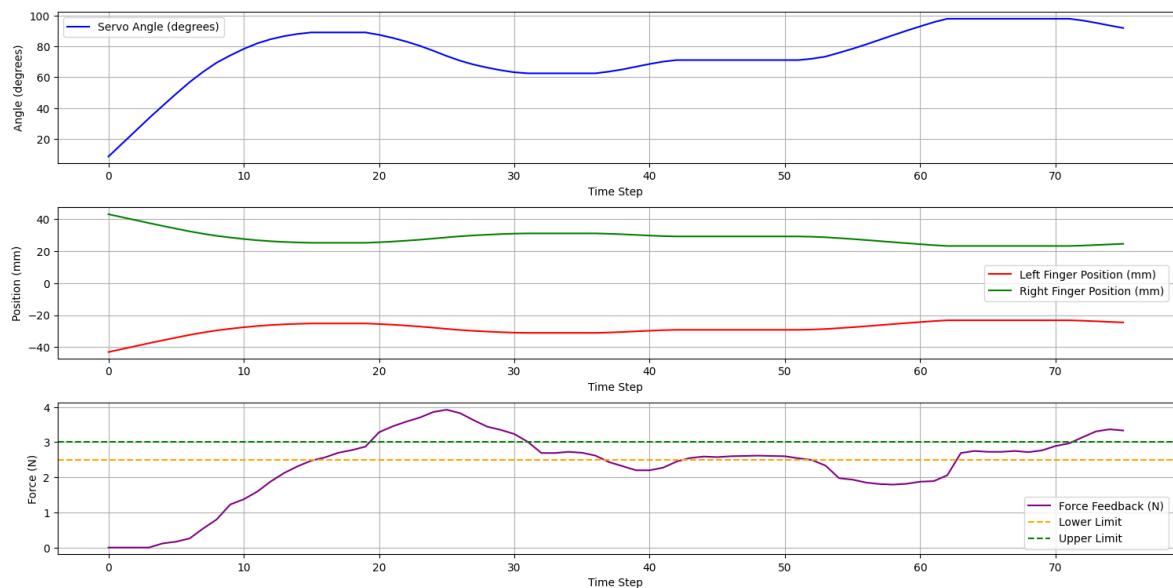
در مرحله بعد، برای آشنایی با نحوه عملکرد کنترل PID، شبیه سازی از عملکرد سروروموتور و کنترلر در محیط پایتون انجام داده شد. در این شبیه سازی، از سنسور FSR برای دریافت فیدبک نیرو استفاده کردیم. به این دلیل که شبیه سازی فیدبک

نیرو امکان پذیر نبود. برای دریافت نیروی واقعی و اعمال فرایند کنترل، نیرو مستقیماً از سریال مانیتور آردوبینو به پایتون متصل میشود و با اعمال فشار بر سنسور و تغییر در نیرو کنترلر زاویه سرووی شبهیه سازی شده را تغییر میدهد و موقعیت فک ها، نیروها و زاویه را نمایش میدهد. این تست برای چند حالت مختلف انجام شد.

\*توجه: مقدار و حدود نیروی هدف و همچنین موقعیت فک های گریپر در این شبیه ساز با شرایط واقعی متفاوت است و این شبیه سازی تنها به منظور درک الگوریتم کنترلی انجام شده است.

حالت اول

$$kp, ki, kd = 0.5, 0.1, 0.05$$

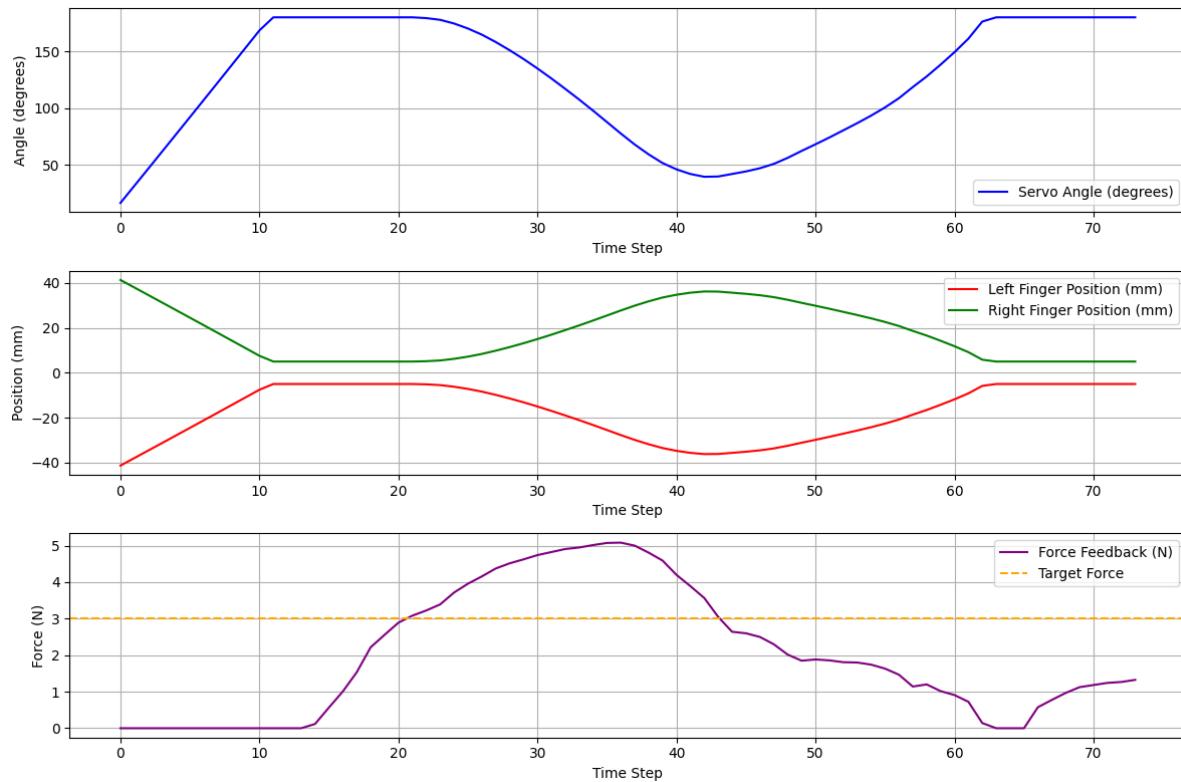


نمودار 4- شبیه سازی حالت 1

در این نمودار پاسخ کنترلر به افزایش بیش از حد نیرو بسیار کم است که میتواند ناشی از ضریب  $kp$  پایین باشد.

حالت دوم

$$kp, ki, kd = 3, 0.1, 0.05$$



نمودار 5 - نتایج شبیه سازی برای حالت دوم

در اینجا هم سرعت پاسخ گویی به نوسان پایین است که می تواند به خاطر مقدار  $k_i$  باشد.

در نهایت کد آردوینوی نوشته میشود که تمام فرامین را در قالب 7 مرحله اجرا کند. مراحل به ترتیب:

مرحله 1- انتظار برای فشردن شستی

مرحله 2 - کالیبراسیون طولانی

مرحله 3- تست سفتی(گرفتن جسم تا 4 درجه پس از نیروی 0.1 نیوتون) در حین جمع شدن فقط زاویه و نیرو را نشان می دهیم

مرحله 4- اجرای PID تارسیدن به نیروی هدف (یا اتمام زمان) > = LCD "PID in progress..." روی

مرحله 5- نگهداری 10 ثانیه (چشمک در 3 ث آخر) > = در تمام این مدت سفتی، نیروی هدف و نیروی نهایی را نشان می دهیم

مرحله 6- باز کردن گریپر

مرحله 7- انتظار 5 ث + کالیبراسیون سریع در 2 ث اول + چشمک در 3 ث آخر + بازگشت به مرحله 3

کد آردوینو به علت خطوط زیاد کد در این گزارش آورده نشده، فایل IDE همراه با گزارشکار با نام **final** ضمیمه شده است.

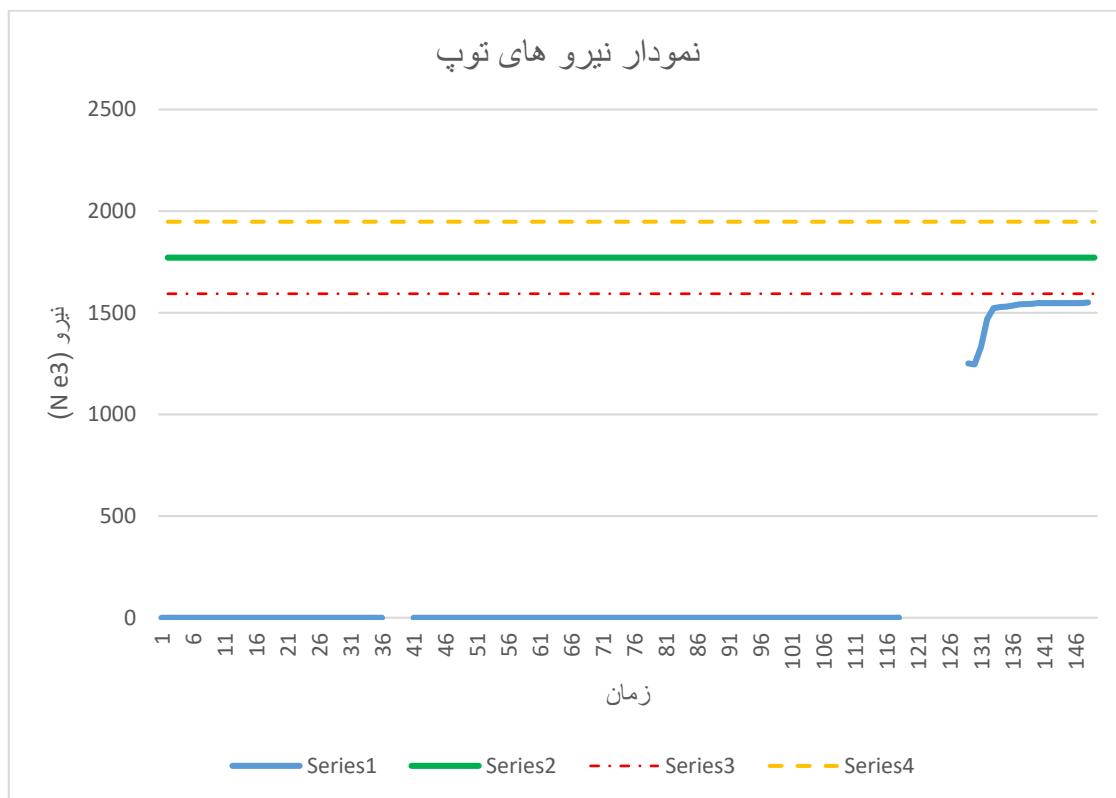
## تست کد بر روی گریپر

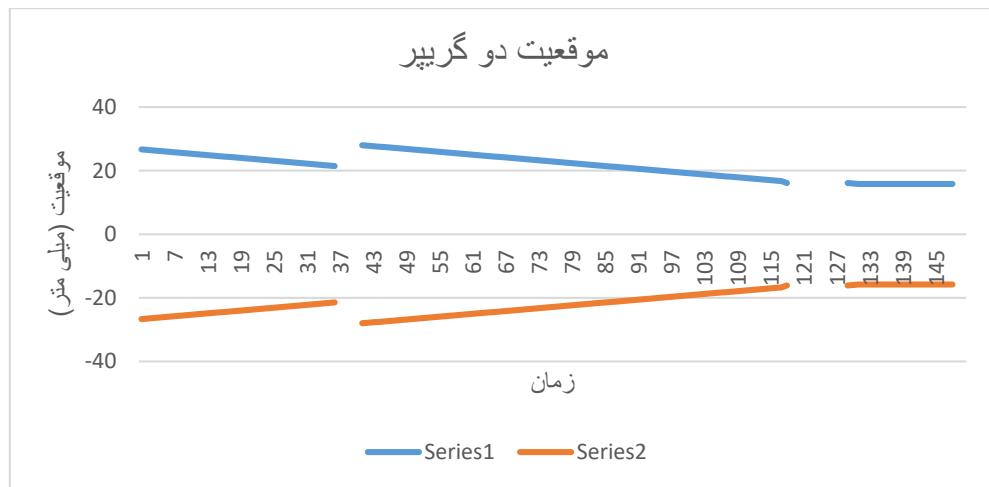
کنترل PID با سه سری ضرایب بر روی گریپر تست شد و نتایج مورد بررسی قرار داده شد تا از بهترین ضرایب استفاده شود. برای مشاهده داده های نیرو و موقعیت، سریال مانیتور آردوبینو را به پایتون متصل کردیم تا خروجی اکسل از داده ها بگیریم و با رسم نمودار عملکرد کنترل را بهتر تحلیل کنیم.

(Kp=3 Kd=0, Ki=0) -1 حالت اول

برای توپ •

Stiffness = 590.309 N/m
Target Force = 1.771
LowerBound = 1.594
UpperBound = 1.948

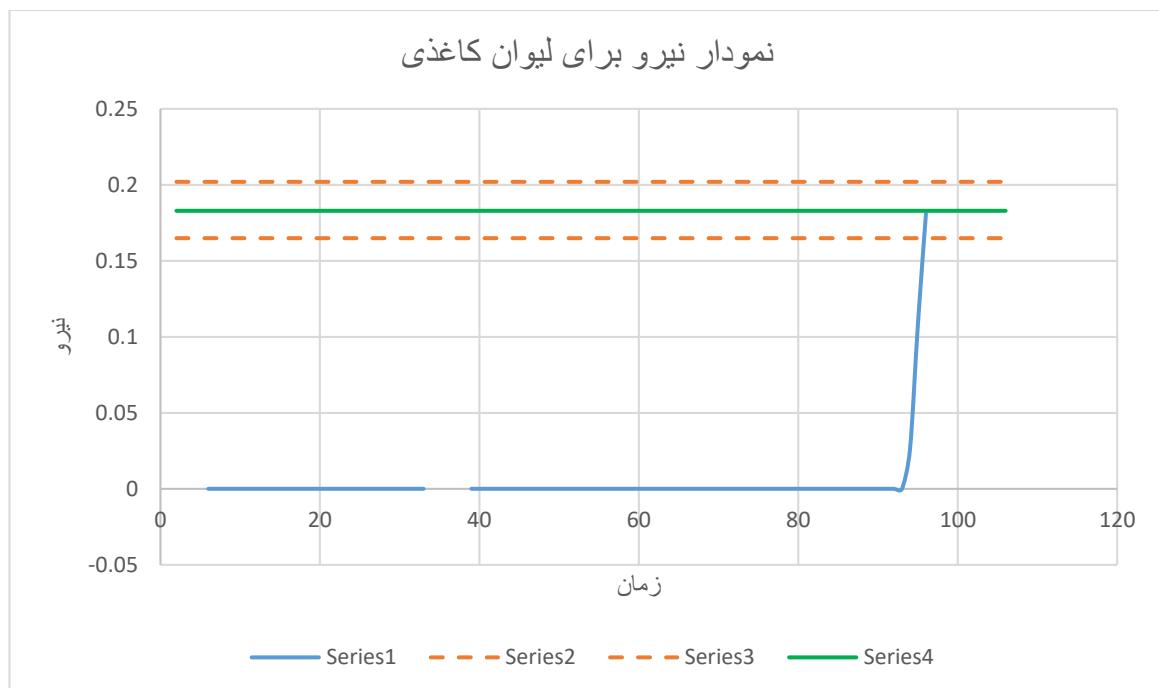




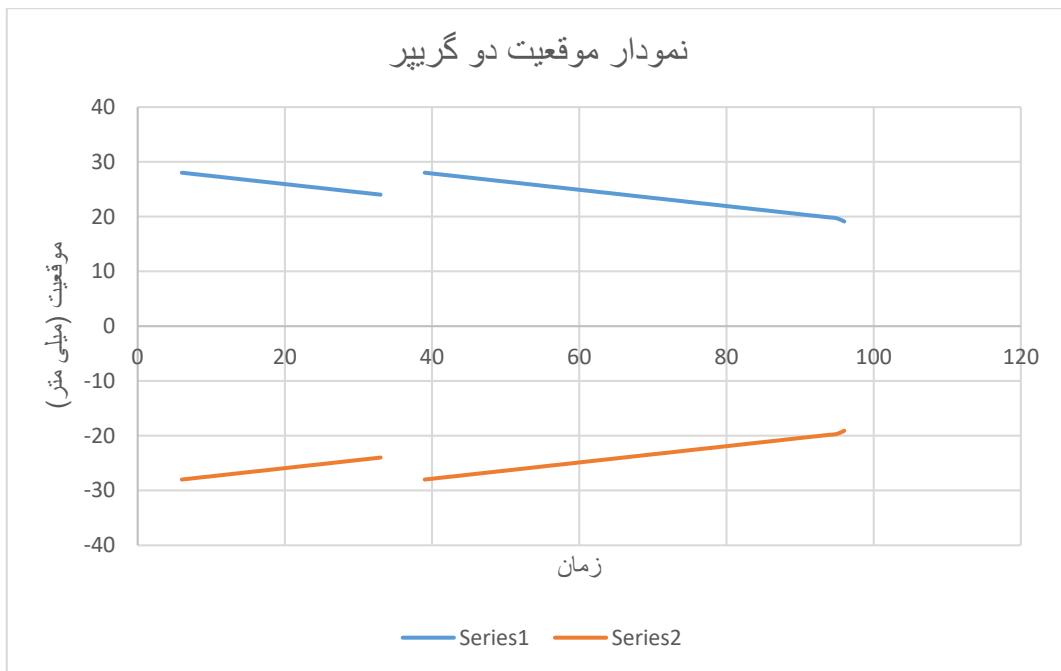
نمودار-7- موقعیت گریپر در حالت اول برای توب

برای لیوان کاغذی •

Stiffness = 61.141 N/m
Target Force = 0.183
LowerBound = 0.165
UpperBound = 0.202



نمودار-8- نیروی وارد به لیوان کاغذی برای حالت اول



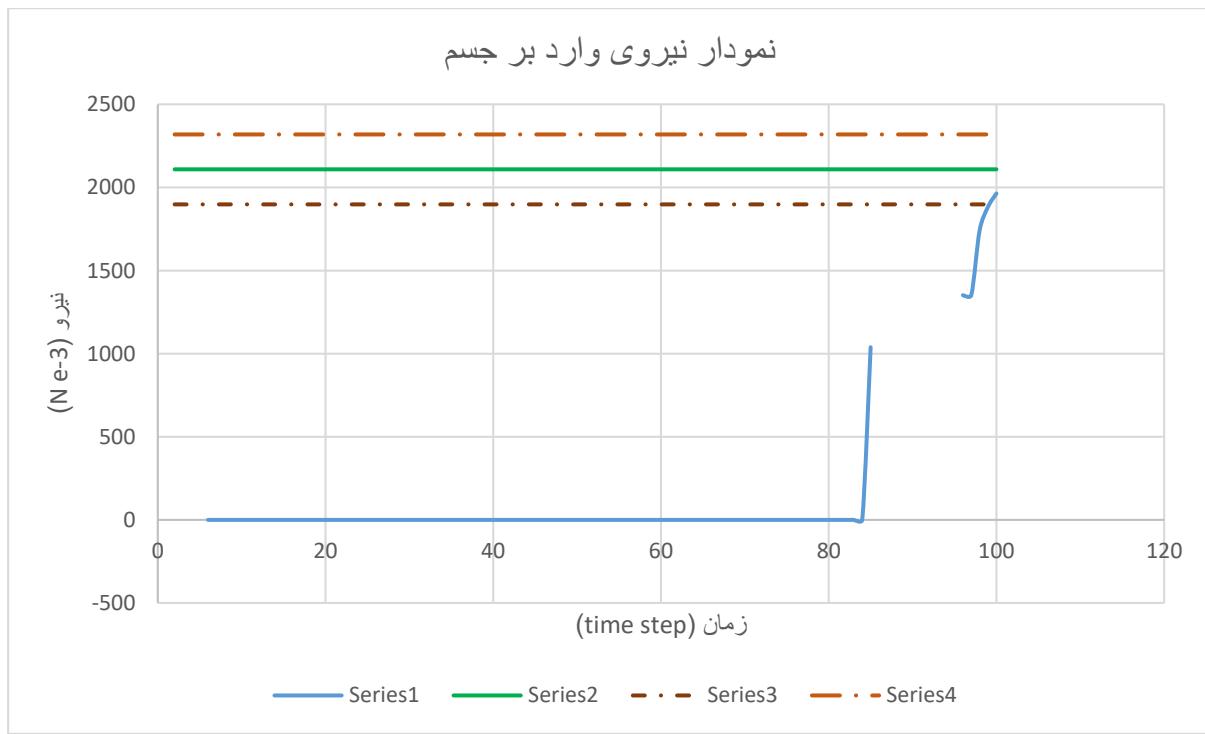
نمودار ۹- موقعیت گریپر در حالت اول برای لیوان کاغذی

تحلیل: در این مرحله و با ضرایبی که داریم طبق نمودار ۶ میبینیم که نیرو به مرز بازه رسیده ولی نمیتواند به بازه هدف برسد. این مشکل می تواند ناشی از مقدار کم  $k_p$  برای آپدیت زاویه و یا مقدار کم  $K_i$  در محاسبه خطای ماندگار باشد. اما برخلاف توپ، نیروی لازم برای نگه داشتن لیوان کاغذی به خوبی و به دقت به جسم وارد شده است. در مجموع این حالت برای ما مطلوب نیست چرا که نیروی کم برای نگه داشتن جسم سنگین مشکلات بیشتری از نیروی زیاد برای جسم سبک به همراه دارد. حال میتوان ضریب  $k_p$  را افزایش داد و یا ضرایب  $K_i$  و  $K_d$  را مقدار داد. که در وهله اول مطلوب این است که از ساده ترین روش ممکن استفاده کرد. بنابراین صرفاً مقدار  $K_p$  را تغییر میدهیم و مجدداً نیرو را برای جسم سنگین بررسی میکنیم.

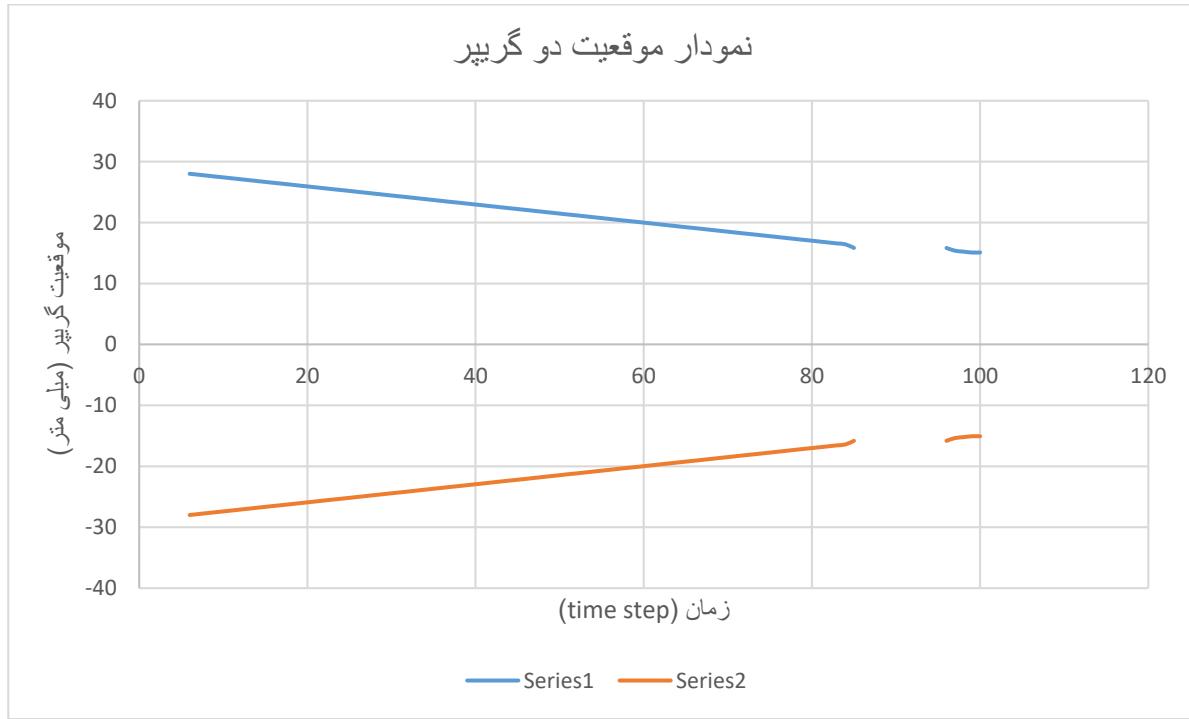
-2 حالت دوم ( $K_p=5$ ,  $K_i=0$ ,  $K_d=0$ )

برای توپ •

Stiffness = 702.861 N/m
Target Force = 2.109
LowerBound = 1.898
UpperBound = 2.319



نمودار 10- نمودار نیروی وارد به توب در حالت دوم



نمودار 11 - موقعیت گریپر در حالت دوم برای توب

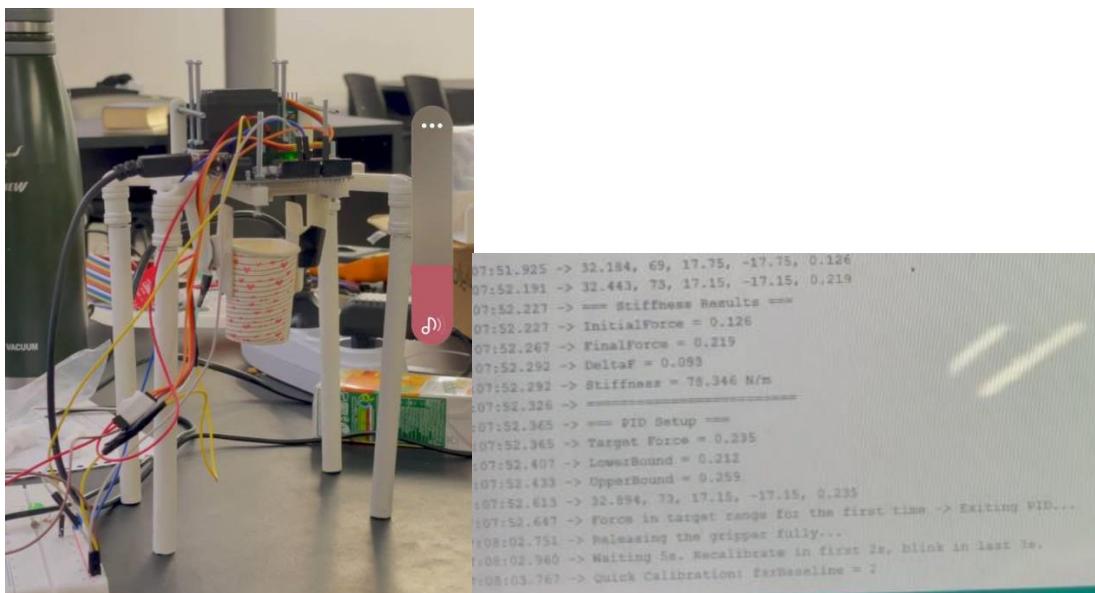
تحلیل: در این تست میبینیم که نیروی هدف ما برای جسم سفت و سنگین در بازه مطلوب قرار گرفته است. همچنین کنترل PID با این ضریب 10 ها بار بر روی قطعات دیگر مورد ارزیابی واقع شد که در تمامی حالات(منهای اجسام با سطوح خاص) عملکرد خیلی خوبی داشت، که دو نمونه از آنها در بخش تست نهایی بررسی خواهد شد.

نهایتاً تصمیم بر آن شد تا از کنترلر ساده P استفاده شود چرا که روش غیر پیچیده و بسیار ساده ای هست و پاسخ مطلوبی هم دارد. همچنین سرعت بالایی دارد که زمانی که گرفتن جسم در زمان کوتاه مطرح میشود میتواند گزینه خوبی باشد. البته این روش ممکن ناپایار باشد و نوسان کند. اما با توجه به روش کنترلی ما که یک بازه را در نظر میگیرد، با تست های متعدد متوجه شدیم مشکلی ایجاد نمیکند.

## تست نهایی

در نهایت این کنترلر را بر روی انواع و اقسام اجسام تست کردیم و نتایج بسیار خوبی گرفتیم. دو نمونه از این تست ها که بر روی اجسام خاص هستند، مانند تخم مرغ که سنگین و شکننده است و لیوان کاغذی که بسیار سبک و منعطف است انجام شده و نتایج به دست آمده است.

### تست با لیوان کاغذی



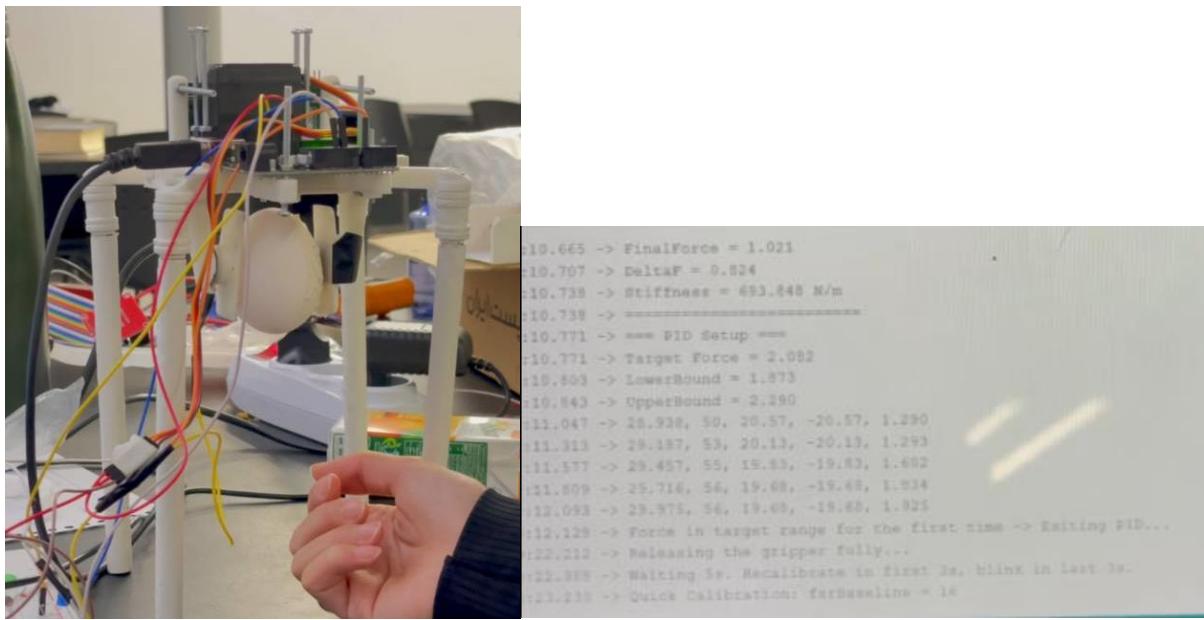
تصویر 28 - تست با لیوان کاغذی به همراه نتایج در سریال مانیتور

نتایج پرینت شده در سریال مانیتور

- سفتی: 78.346
- نیروی هدف 0.235
- حد پایین نیرو: 0.212
- حد بالای نیرو: 0.253
- نیروی گریپر: 0.219

گریپر به خوبی و با نیروی صحیح جسم را نگه داشت است.

### تست با تخم مرغ



تصویر 30- تست با تخم مرغ

نتایج پرینت شده در سریال مانیتور

- سفتی: 693.848
- نیروی هدف: 2.082
- حد پایین نیرو: 1.873
- حد بالای نیرو: 2.290
- نیروی گریپر: 1.925
- گریپر به خوبی و با نیروی صحیح جسم را نگه داشت است.

توجه: ویدئو این تست ها در فایل ضمیمه قابل مشاهده است.

## جمع بندی و نتیجه گیری

در این پروژه، یک پنجه رباتیکی (**Gripper**) طراحی و ساخته شد که قادر است اجسام مختلف را با وزن حداقل ۷۰ گرم و قطر ۵ سانتی‌متر نگه‌داری کند. برای طراحی مکانیزم حرکتی، با بررسی گزینه‌های مختلف، مکانیزم رک و پینیون انتخاب شد، زیرا حجم کمتری داشت و بازه حرکتی مناسبی برابر با ۰,۷ تا ۰,۶ سانتی‌متر فراهم می‌کرد.

برای اعمال نیروی مناسب به اجسام، از سنسور نیرو (**FSR**) استفاده شد که با آزمایش وزنهای و مقاومت‌های مختلف کالیبره شد تا مقادیر دقیق تری نمایش دهد و برای جلوگیری از نوسان و خطای عدم پایداری از فیلتر moving average اسفاده شد. همچنین برای تشخیص سختی (سفتی) اجسام، آزمایشی انجام شد که در آن با لمس جسم و اندازه‌گیری جایه‌جایی، مقدار سفتی جسم استخراج شد. سپس رابطه‌ای بین سفتی جسم و نیروی مناسب به دست آمد تا پنجه بتواند اجسام نرم و سخت را بدون آسیب‌دیدگی بگیرد.

در بخش کنترل، سه سری ضرایب متفاوت امتحان شد، اما در نهایت از بین حالت‌هایی که تست گرفته شده بود، حالت سوم از بقیه مناسب تر بود چرا که اجسام سنگین را با نیروی بهتر و دقیق تری نگه می‌داشت. هرچند که در نگه داشتن جسم سبک خطای بیشتری داشت، اما اعمال نیروی زیاد به جسم سبک بهتر از اعمال نیروی کم به جسم سنگین است. پس از انجام تست‌ها مشخص شد که کنترل تناسبی (**P-Control** به تنهایی کافی است، زیرا نیروی اعمالی نیاز به اصلاح انتگرالی و مشتقی نداشت. بنابراین، با استفاده از کنترل **P** ساده‌تر و کارآمدتر، نیروی مناسب برای گرفتن اجسام تنظیم شد.

در نهایت، پنجه طراحی شده با موفقیت عملکردی ساده و مطلوب در گرفتن اجسام مختلف داشت و نیازهای پروژه را برآورده کرد.

## منابع

- [1] Ashutosh Kumar1 , Abburi Lakshman Kumar,(2020), Robotic Gripper With Force Feedback System
  - [2] servoMG996R datasheet
  - [3] FSR 402 datasheet
  - [4] Gottipati S Lakshmi,H emavathi R(2020), Calibration of Force Sensitive Resistor used in Force Controlled Grippers
- [5] ساناز جباری، مهدی مدبری فرد، مجتبی قدسی(1396) یک گریپر قابل برنامه ریزی حساس به سفتی اشیا برای جا به جایی در سیستم های هپتیکی رباتیکی