

دانشگاه صنعتی شریف

دانشکده مکانیک

پایان‌نامه کارشناسی

عنوان

کنترل حرکت ربات ماهی مغناطیسی متحرک در سیال

نگارش

نوید شیرمحمدی

استاد راهنما

دکتر حسین نجات پیش کناری

استاد همکار

دکتر علی مقداری

اردیبهشت 1398

به‌ نام خدا

دانشگاه صنعتی شریف

دانشکده منهدسی مکانیک

پایان‌نامه کارشناسی

عنوان: کنترل حرکت ربات ماهی مغناطیسی متحرک در سیال

نگارش: نوید شیرمحمدی

کمیته ممتحنین:

استاد راهنما : ...................................... امضا:

استاد راهنمای همکار: ............................ امضا:

استاد مشاور: ....................................... امضا:

استاد مدعو: ........................................ امضا:

استاد مدعو: ........................................ امضا:

تاریخ دفاع: ......................

از استاد گران قدر، دکتر حسین نجات برای فراهم آوردن فرصت و امکانات لازم جهت انجام این پروژه و همراهی ایشان در طی این مسیر صمیمانه قدردانی میکنم.

همچنین با تشکر فراوان از مهندس مبین صالحی جهت کمک‌های بی‌دریغی که در این زمینه مبذول فرمودند.

# چکیده:

ربات ماهی مانند مغناطیسی یک ربات کوچک در حد یک سانتی متر است که دارای یک سر و یک دم است که به هم لولا شده اند. در قسمت مفصل این ربات یک آهنربا قرار گرفته است که توسط میدان مغناطیسی بیرونی جهت آن کنترل می گردد. میدان مغناطیسی خارجی توسط یک ستاپ سیم پیچ هلمهولتز تولید می شود که توانایی ایجاد میدان مغناطیسی بدون تولید گرادیان را دارد. هدف ما در این پروژه این است که با استفاده از یک دوربین مکان ربات را اندازه گیری نموده و در یک حلقه بسته کنترلی مکان ربات را کنترل کنیم. این کار با استفاده از یک میکروپروسسور رزبری پای که دارای قابلیت پردازش تصویر است انجام خواهد شد.

# واژه های کلیدی:

ربات ماهی‌ تصویر، رزبری پای، کنترل، میدان مغناطیسی، سیم پیچ هلمهولتز، OpenCV.

Contents

[چکیده: 5](#_Toc7614105)

[واژه های کلیدی: 5](#_Toc7614106)

[1 مقدمه 1](#_Toc7614107)

[1-1 اهمیت و کاربرد موضوع 1](#_Toc7614108)

[1-2 اهداف پژوهش 1](#_Toc7614109)

[1-3 محتوای پروژه 1](#_Toc7614110)

[2 پیشینه پژوهش 2](#_Toc7614111)

[3 مفاهیم پایه الکترومغناطیس حاکم برمساله 6](#_Toc7614112)

[3-1 میدان مغناطیسی ناشی از سیم حامل جریان 6](#_Toc7614113)

[3-2 سیم پیچ هلمهولتز 6](#_Toc7614114)

[3-3 ثابت زمانی سیستم 8](#_Toc7614115)

[4 مشخصات سامانه آزمایش 9](#_Toc7614116)

[4-1 سامانه هولمتز 9](#_Toc7614117)

[4-1-1 سیم‌ پیچ ها 9](#_Toc7614118)

[4-1-2 کنترلر آردوینو 9](#_Toc7614119)

[4-1-3 درایو BTS7960 10](#_Toc7614120)

[4-2 دوربین 12](#_Toc7614121)

[4-3 رزبری پای 13](#_Toc7614122)

[5 ربات ماهی 14](#_Toc7614123)

[6 منحنی مدولاسیون پهنای باند (PWM) و جریان سیم پیچ 15](#_Toc7614124)

[7 موقعیت یابی ربات 17](#_Toc7614125)

[7-1 استفاده از برچسب رنگی 17](#_Toc7614126)

[7-1-1 فضای رنگ BGR 17](#_Toc7614127)

[7-1-2 فضای رنگ HSV 18](#_Toc7614128)

[7-1-3 سایر فضا‌های رنگ مورد بررسی 18](#_Toc7614129)

[7-2 استفاده از برچسب آروکو ArUco 20](#_Toc7614130)

[7-3 استفاده از فیلتر های تشخیص حرکت 20](#_Toc7614131)

[8 تنظیمات دوربین 22](#_Toc7614132)

[9 تنظیمات لینوکس 23](#_Toc7614133)

[10 موازی سازی برنامه 24](#_Toc7614134)

[10-1 اهمیت موازی سازی برنامه 24](#_Toc7614135)

[10-2 موازی سازی در پایتون 24](#_Toc7614136)

[References 29](#_Toc7614137)

# 1 مقدمه

## 1-1 اهمیت و کاربرد موضوع

ربات های متحرک در ابعاد میلیمتر وکوچکتر به علت قابلیت استفاده در کاربردهای پزشکی همانند منیپولیشن سلولها، جراحی و انتقال هدفمند دارو در سال های اخیر بسیار مورد توجه قرار گرفته‌اند. برای تامین نیروی محرکهی رباتها، با توجه به محدودیت ابعادی آنها از راههای متفاوتی مانند روشهای نوری، شیمیایی، حرارتی، الکترواستاتیکی و الکترومغناطیسی استفاده شده است که در میان این روش ها، میدان مغناطیسی مزایای بسیاری دارد که برای کاربردهای پزشکی مطلوب است؛ از جمله آن که قابلیت نفوذ بالایی در بافتهای انسان دارد و به آنها آسیب نمی‌رساند. مزیت دیگر این روش این است که نیازی به نصب عملگرهای محرک روی ربات که حجم زیادی را اشغال میکنند، ندارد و نیازی به ارتباط سیمی نیز وجود ندارد در نتیجه امکان ساخت ربات در اندازه کمتر را ممکن می‌کند.

## 1-2 اهداف پژوهش

هدف این پژوهش پیاده سازی یک سیستم کنترلی مدار بسته جهت کنترل یک ربات ماهی تصویر کوچک توسط میدان مغناطیسی خارجی در مسیر دلخواه می‌باشد. میدان مغناطیسی خارجی توسط یک ستاپ سیم پیچ هلمهولتز[[1]](#footnote-1) تولید می‌شود که توانایی ایجاد میدان مغناطیسی بدون تولید گرادیان را دارد. در این پروژه با استفاده از یک دوربین مکان ربات را مشخص نموده و در یک حلقه بسته[[2]](#footnote-2) کنترلی مکان ربات را کنترل کنیم.

## 1-3 محتوای پروژه

پایان نامه حاضر شامل # فصل است که جزییات مختلف پروژه را در بر میگیرند. در فصل اول به #

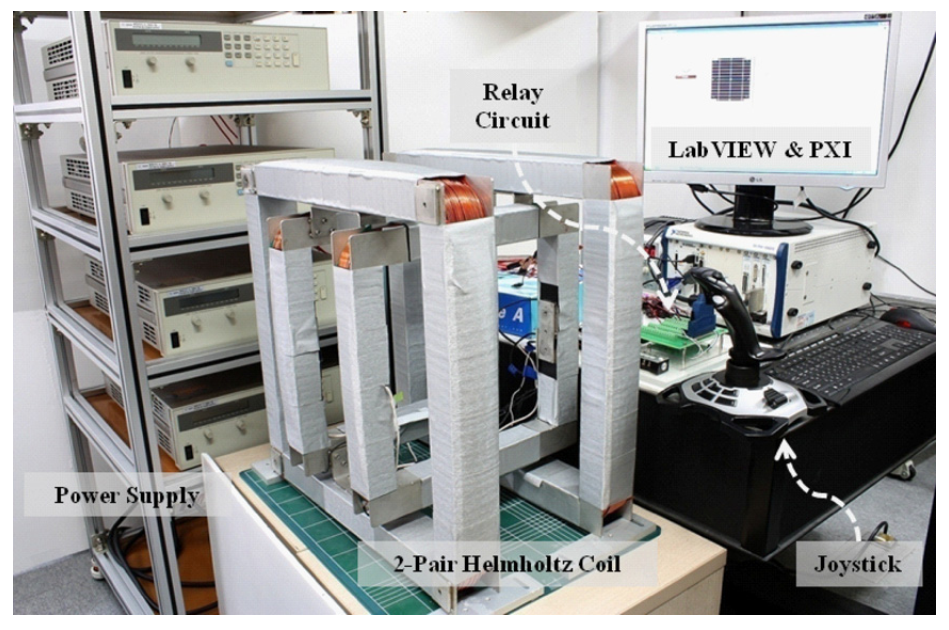
آدرس گیت

# 2 پیشینه پژوهش

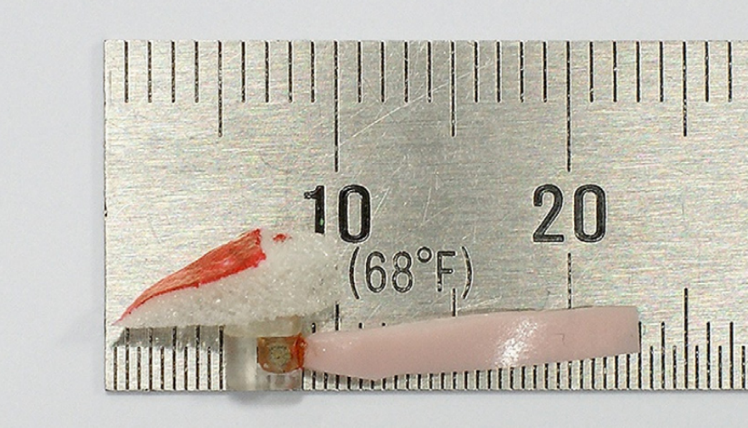
در این فصل به مرور تعدادی از مقالات مطرح شده در این زمینه پرداخته می‌شود.

در مقاله ]1[، Donghak Byun و Jongho Choiبا استفاد ازسیم پیچ هلمهولتز مربعی سعی در کنترل یک ربات ماهی مطابق تصویر # را دارند. علت استفاده از سیم پیچ مربعی در این پژوهش ناحیه عملکر[[3]](#footnote-3) بزرگتر نسبت به حالت دایره ‌ای ذکر شده. در این پژوهش محققان در نظر دارند تا در کاربری پزشکی با استفاده از دستگاه [[4]](#footnote-4)MRI موقعیت میکرو ربات در بدن بیمار تشخیص دهند شود و به کمک سیم پیچ خارجی میدان مغناطیسی لازم برای کنترل ربات را ایجاد کنند.

تصویر #و # به ترتیب سیم پیچ هلمهولتز و ربات ماهی مورد استفاده در این پژوهش را نشان می دهند.

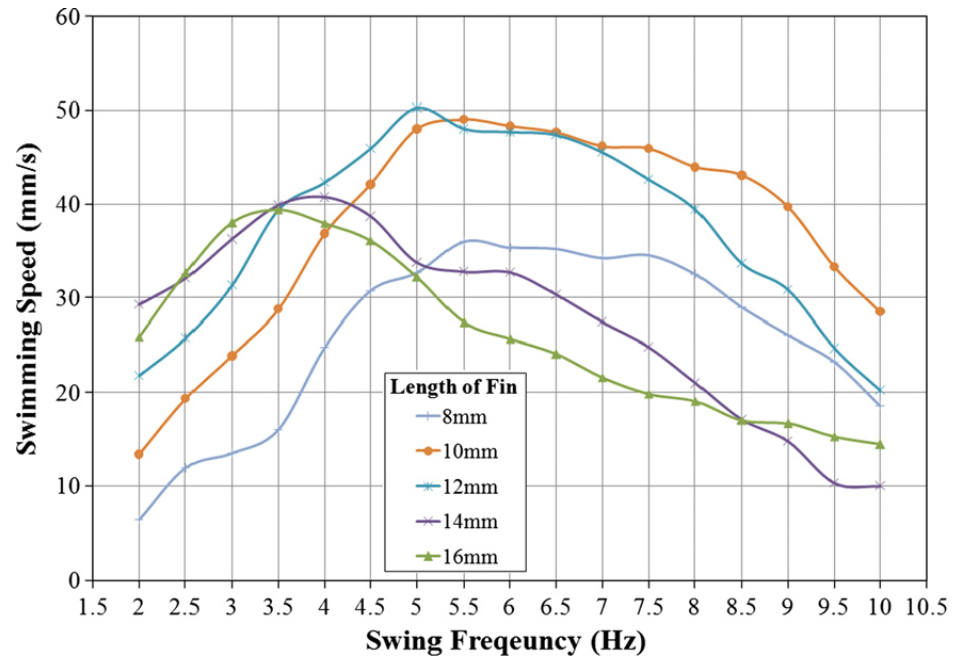


تصویر 1 سیم پیچ هلمهولتز مربعی

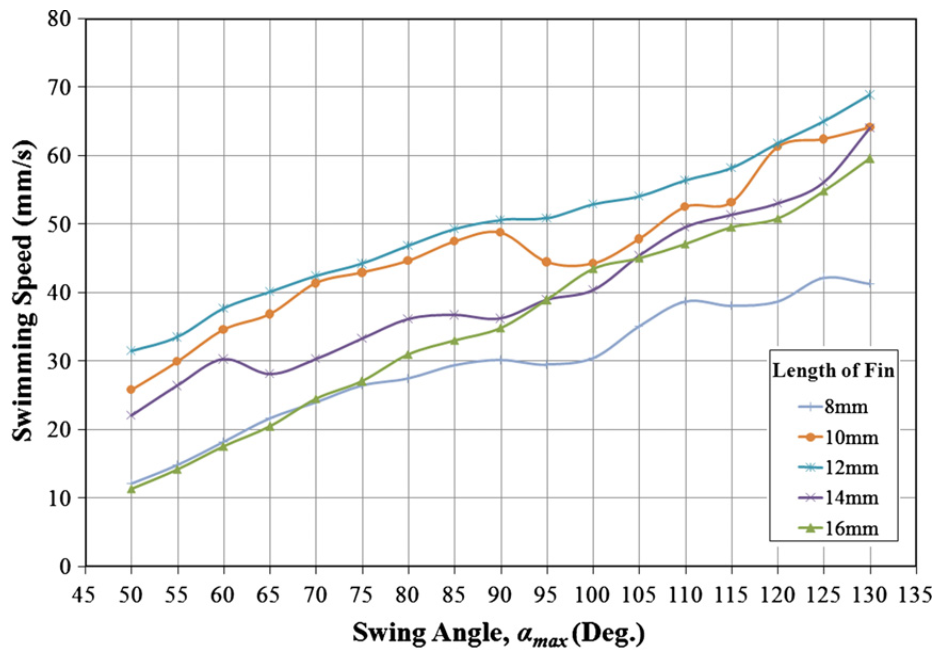


تصویر 2 ربات ماهی

متغیر‌های مورد بررسی در بهبود حرکت ربات که در این پژوهش مورد بررسی قرار گرفته اند عبارتند از طول دم ماهی، فرکانس میدان خارجی و زاویه انحراف دم ماهی. تصویر #و # تاثیر این متغیرها را در ربات های مورد بررسی این پژوهش نشان می دهند.

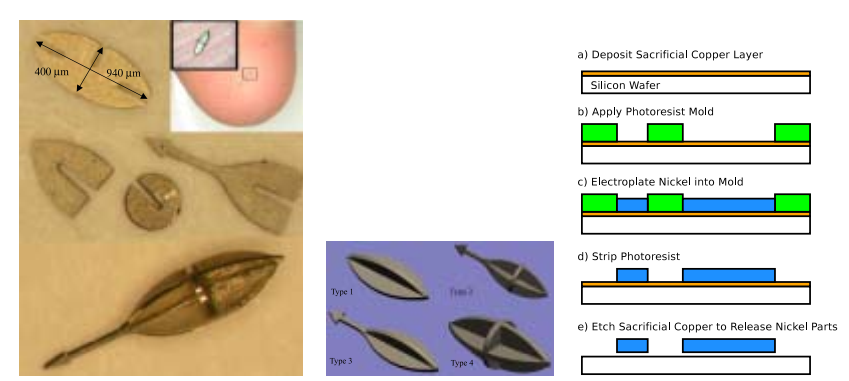


تصویر 3 منحنی فرکانس-سرعت برای طول دم‌های مختلف ربات ماهی

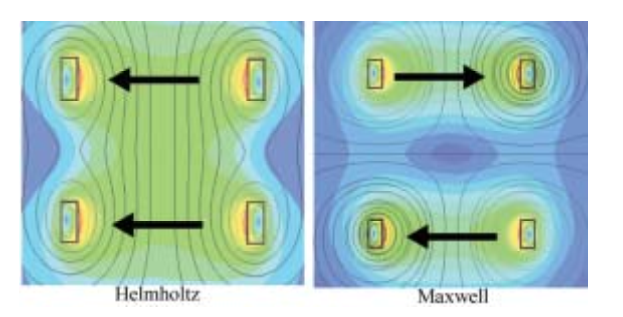


تصویر 4 منحنی زاویه انحراف دم-سرعت برای طول دم‌های مختلف ربات ماهی

در مقاله ]2[، K. Berk Yaser, Karl Vollmers و Bradlley J. Nelson به مسایل بنیادین در ساخت، طراحی و کنترل یک میکروربات ماهی تصویر میپردازند. ربات استفاده شده در این مقاله با استفاده از فناوری MEMS تولید شده و تصویر # آن را نشان می‌دهد. بررسی اثرات درگ، هیسترزیس، تغییرات دما و ارتعاشات ماهی از جمله موارد مهم بررسی شده در این مقاله‌اند. سیم پیچ ماکسول[[5]](#footnote-5) نیز مورد دیگری است که در کنار سیم پیچ هلمهولتز برای کنترل میکروربات در این مقاله بررسی شده.



تصویر 5 میکروربات مقاله ]2[

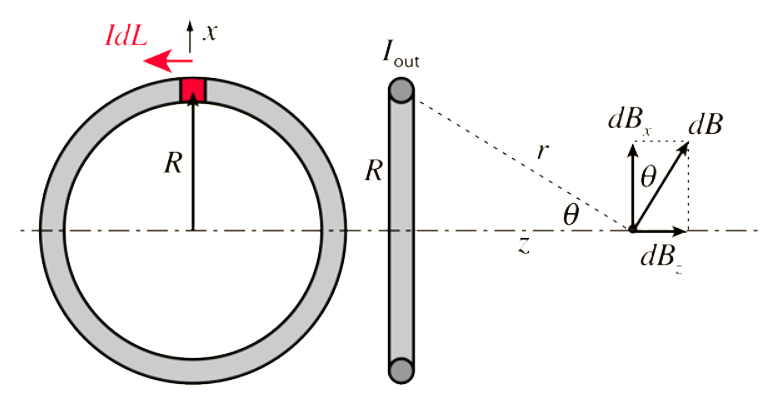


تصویر 6 سیم پیچ های ماکسول و هلمهولتز

# 3 مفاهیم پایه الکترومغناطیس حاکم برمساله

## 3-1 میدان مغناطیسی ناشی از سیم حامل جریان

طبق قانون بیوساوار، میدان مغناطیسی روی محور یک حلقه سیم پیچ دایروی به شعاع R و حامل جریان I در فاصله z از محور آن از رابطه (1) بدست می‌آید. میدان مغناطیسی ناشی از المان سیم پیچ در فاصله Z از محور سیم پیچ در تصویر 1 نشان داده شده است.



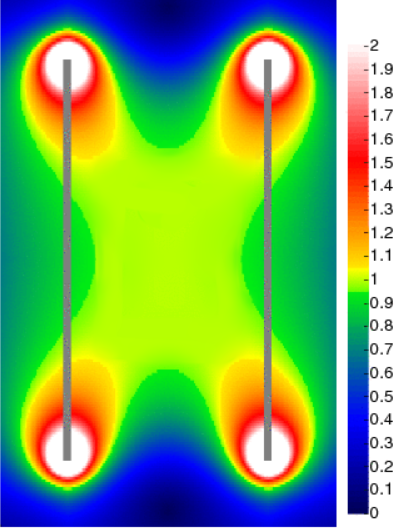
تصویر 7 بردار های میدان مقناطیسی المان حلقه باردار

در رابطه (1)، ضریب تراوایی مغناطیسی خلا نام دارد و مقدار آن می‌باشد.

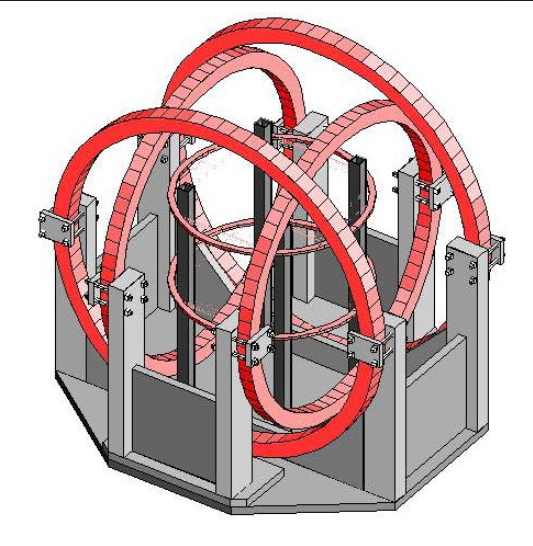
## 3-2 سیم پیچ هلمهولتز

سیم پیچ هلمهولتز از قرار دادن دو دو سیم پیچ یکسان و هم محور با جریان عبوری هم انداری و هم جهت از آنها و به فاصله قطر سیم پیچ آنها از هم، ساخته می‌شود. در این سیم پیچ در فضای بین دو سیم پیچ میدان مغناطیسی یکنواخت ایجاد میشود و گرادیان میدان وجود نخواهد داشت. تصویر 3 توزیع میدان مغناطیسی را برای این سیم پیچ نسبت به اندازه میدان در مرکز سیم پیچ نشان میدهد.

به کمک سه سیم پیچ هلمهولتز مطابق تصویر 3 می‌توان میدان مغناطیسی یکنواخت در فضای مشترک بین سیم پیچ ها با اندازه و جهت دلخواه در سه بعد ایجاد کرد.



تصویر 8 توزیع میدان مغناطیسی درسیم پیچ هلمهولتز

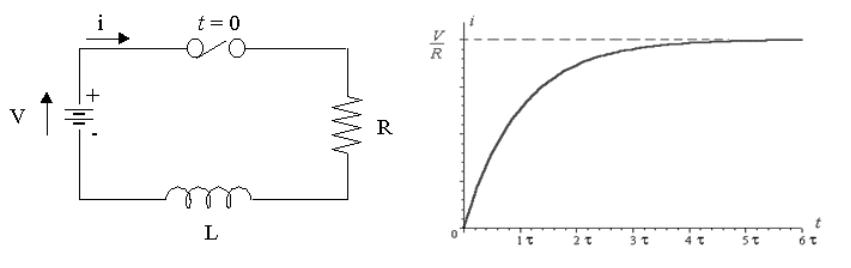


تصویر 9 ایجاد میدان مغناطیسی یکنواخت دلخواه با سه سیم پیچ هلمهولتز

اندازی میدان مغناطیسی برای سیم پیچ هلمهولتز، با استفاده از رابطه ی (1) و به صورت زیر محاسبه میشود.

## 3-3 ثابت زمانی سیستم

در مدار RL مطابق تصویر 4 با تغییر ناگهانی ولتاژ دو سر مدار، القاگر با اعمال نیروی محرکه القایی در مدار با تغییر جریان مدار جلوگیری میکند. در نتیجه فرکانس کنترل ربات باید در محدوده ای قرار گیرد که مدار فرصت لازم برای رسیدن به جریان مورد انتظار و ایجاد میدان مورد نظر را داشته باشد.



تصویر 8 مدار RL و اثر ولتاژ خودالقایی القاگر

ثابت زمانی مدار ، مدت زمانی است که ولتاژ مدار به 99.5 درصد ولتاژ اعمال شده برسد. محاسبه ثابت زمانی N حلقه جریان با مقاومت کل R به صورت زیر است.

در رابطه (2)، N تعداد دور حلقه، D قطر حلقه و d قطر سیم است. ضریب تراوایی نسبی هسته سیم پیچ است که در این پروژه با توجه به نبود هسته، این ضریب برابر 1 در نظر گرفته میشود.

# 4 مشخصات سامانه آزمایش

## 4-1 سامانه هولمتز

سامانه هولمتز مورد استفاده در این پژوهش را در سه بخش سیم پیج، کنترلر و درایور میتوان بررسی کرد. در ادامه به بررسی مشخصات هر بخش میپردازیم.

### 4-1-1 سیم‌ پیچ ها

در این بخش مشخصات سیم پیچ های سامانه هولمتز آورده شده است. پارامتر‌های سیم پیچ ها عبارتند از قطر مفتول، قطر حلقه سیم پیچ، تعداد حلقه و مقاومت کل برای هر سیم پیچ. با توجه به کنترل ربات ماهی در صفحه، مشخصات سیم پیچ محور Z در این پروژه آورده نمیشود. مشخصات سیم پیچ های محور X و Y در جدول 1 آورده شده اند. در ستون آخر ثابت زمانی با استفاده از رابطه (2) که در3-3 ارایه شد، محاسبه شده است.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| محور سیم پیچ | D [mm]\* | d [mm] | R [ohm] | N |  |
| X | 305 | 1.3 | 0.17 | 270 | 1.68 |
| Y | 215 | 1.3 | 0.12 | 382 | 2.23 |

جدول 1پارامترهای سیم پیچ. \*: میانگین قطر داخلی و خارجی حلقه

بالا ترین فرکانسی که کنترلر ایجاد میکند تا مدار بتواند به آن پاسخ مورد نظر را دهد، در حالت ایده‌آل برابر عکس بالاترین ثابت زمانی سیم پیچ خواهد بود. با توجه به داده های جدول، بالاترین فرکانسی کنترلی مدار در حالت ایده‌آل برابر 448 Hz است.

### 4-1-2 کنترلر آردوینو

با توجه به حساسیت بالا میکروپروسسور رزبری پای به جریان عبوری، برای کنترل درایو های کنترل کننده PWM نیاز به استفاده از میکروکنترلر Arduino در کنار رزبری برای کنترل درایور ها استفاده شود. در این پروژه از میکرو کنترلر Arduino UNO برای این کار استفاده شد هر چند استفاده از مدل های دیگر این میکرو کنترلر به شرط توان عبور جریان برابر یا بیشتر و تعداد پین کافی مشکلی ایجاد نخواهد کرد. جدول 2 مشخصات کاری مهم Arduino Uno و Raspberry Pi را نشان میدهد.

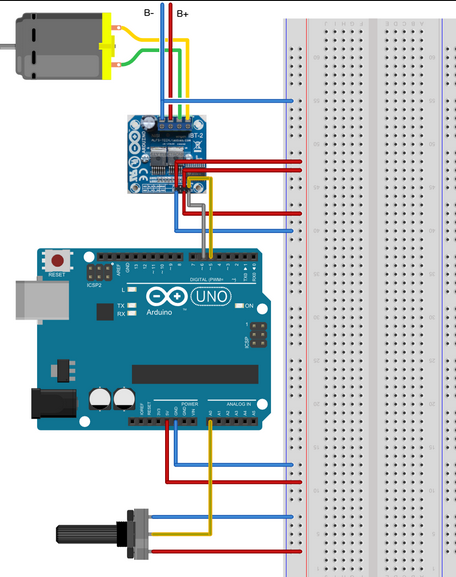
|  |  |  |
| --- | --- | --- |
|  | Arduino Uno | Raspberry Pi |
| کلاک | 16 MHz | 700 MHz |
| حداکثر جریان ورودی/خروجی | 40 mA | 5-10 mA |
| رم | 2k | 512 MB |

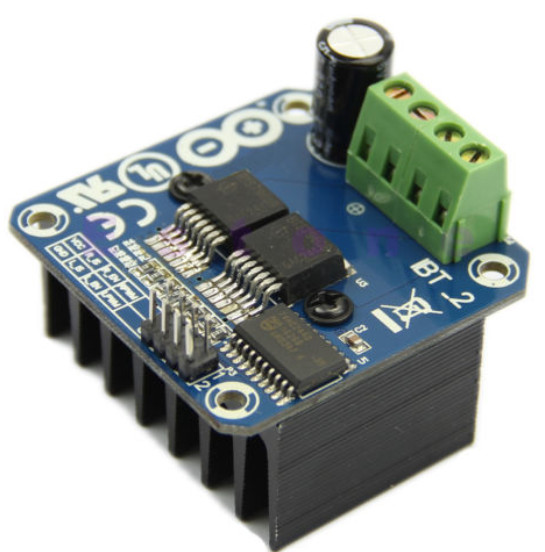
جدول 2 مقایسه برخی مشخصات کاری Raspberry Pi و Arduino Uno

در نتیجه میکروکنترلر آردوینو جریان لازم برای درایو های متصل به سیم پیچ ها را تامین خواهد کرد.

### 4-1-3 درایو BTS7960

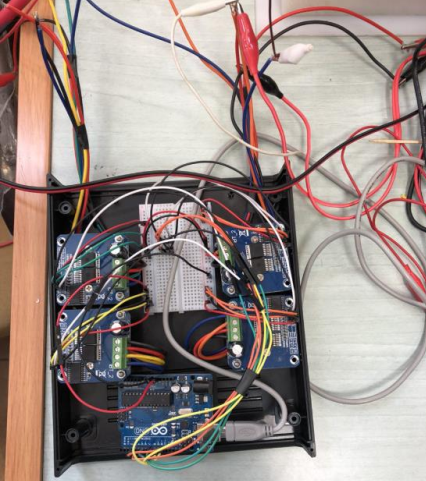
نقش درایوها کنترل جریان بالای درون سیم پیچ ها بر اساس دستورات میکروکنترلر آردوینو است. درایو BTS7960 میتواند تا ولتاژ 27.5 ولت و جریان 43 آمپر را برای سیم پیچ ها فراهم کند. تصویر 5 تصویری از این درایو و مدار اتصال آن به آردوینو برای تامین توان لازم برای یک موتور DC را نشان میدهد.





تصویر 9 درایو BTS 7690 و اتصال آن

تصویر از بالای مدار آردوینو و درایورها برای کنترل جریان سیم پیچ ها در تصویر 6 آمده است.



تصویر 10 تصویر از بالای اتصال درایورها به آردوینو

## 4-2 دوربین

برای دادن فیدبک به لوپ کنترلی، تصویر از بالای ربات توسط وب کم A4Tech PK-750MJ به کنترلر داده می‌شود و کنترلر پس از یافتن موقعیت ربات، جهت و اندازه میران مغناطیسی را مشخص میکند. تعدادی از پارامتر های موثر دوربین برای آزمایش در جدول دو آورده شده است ]منبع#[.

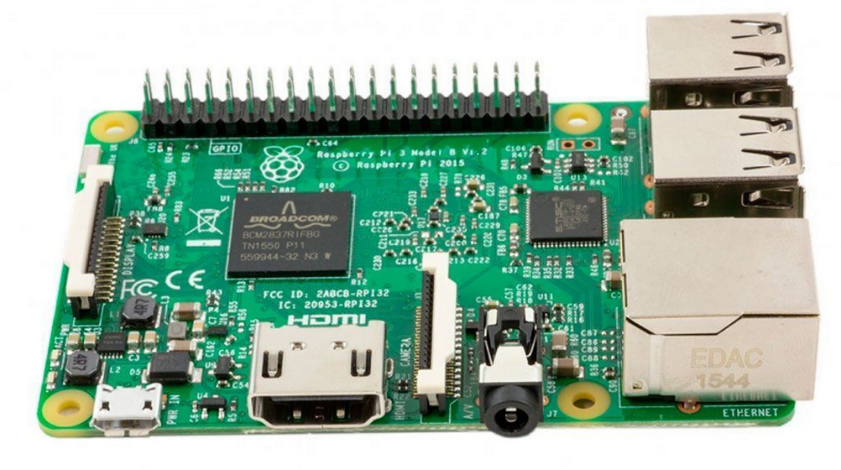
|  |  |
| --- | --- |
| مشخصات وب کم A4Tech PK-750MJ | |
| زاویه دید | 60 درجه |
| تعداد پیکسل | 16 مگاپیکسل |
| فرکانس کاری | حداکثر 30 فریم بر ثانیه |

جدول 2 مشخصات وب کم

با توجه به جدول 2 عامل مهم دیگری که فرکانس کاری دستگاه را محدود میکند، فرکانس دوربین است. مشاهدات نشان داد در اتصال به رزبری پای و بدون اعمال پردازش روی تصویر فرکانس کاری دستگاه به 13 تا 17 fps محدود می‌شود. این عدد برای کامپیوتر در بازه 19 تا 29 fps وابسته به تصویر تغییر می‌کرد.

## 4-3 رزبری پای

برای کنترل ربات در نهایت کد تست شده روی کامپیوتر باید روی میکروپروسسور پیاده سازی و تست شود. به دلیل نیاز به پردازش تصویر در پروژه استفاده از میکروکنترلر Arduino برای این کار امکان پذیر نخواهد بود. میکرو پروسسور رزبری پای ورژن B3 با سیستم عامل Raspbian آپدیت 2019 April برای این پروژه استفاده شد. تصویر # این میکروپروسسور را نشان می‌دهد. طبق کاتالوگ این میکروپروسسور تنها یک پورت استاندارد برای تولید PWM دارد اما در عمل سایر پورت‌های این میکروپروسسور با نصب کتابخانه ی GPIO پایتون می‌توانند برای ایجاد PWM به کار روند.



تصویر 11 رزبری پای

تصویر نهایی سامانه آزمایش در زمان تست کد روی رایانه در تصویر 8 آمده است.



تصویر 12

## 5 ربات ماهی

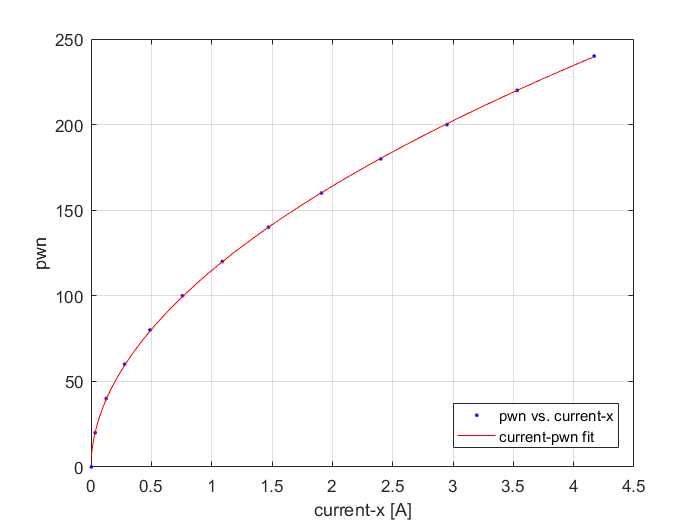
تصویر #، ربات مورد بررسی در این پروژه را نشان می‌دهد . در دم ربات آهنربای NdFeBr به قطر 2.5 میلی متر و طول 3.8 میلی متر، مطابق تصویر #، قرار داده شده است. بدنه ربات توسط چاپگر سه بعدی و از PLA ساخته شده. مشخصات ربات عبارتند از ....

# 6 منحنی مدولاسیون پهنای باند (PWM) و جریان سیم پیچ

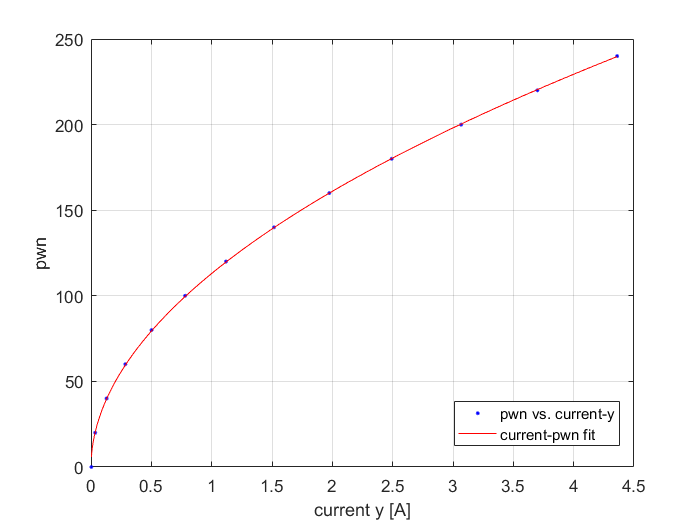
در بخش 1-3 ، رابطه ی بین میدان مغناطیسی سیم پیچ هلمهولتز و جریان عبوری از آن را محاسبه کردیم ولی ارتباط ما با سیم پیچ با تنظیم مدولاسیون پهنای باند[[6]](#footnote-6) است نه جریان، در نتیجه باید رابطه ی میان مدولاسیون پهنای باند و جریان را بدست آوریم. جدول # نتایج داده برداری از سیم پیچ ها در PWM های مختلف و جریان عبور از آنها را نشان میدهد. فرمول منحنی عبوری از داده ها به کمک cftool متلب در انتهای جدول آمده است.

|  |  |  |
| --- | --- | --- |
| PWM | جریان سیم پیچ محور x [A] | جریان سیم پیچ محورy [A] |
| 0 | 0.003 | 0.003 |
| 20 | 0.035 | 0.035 |
| 40 | 0.127 | 0.129 |
| 60 | 0.279 | 0.285 |
| 80 | 0.489 | 0.502 |
| 100 | 0.758 | 0.780 |
| 120 | 1.088 | 1.119 |
| 140 | 1.470 | 1.517 |
| 160 | 1.910 | 1.975 |
| 180 | 2.402 | 2.492 |
| 200 | 2.951 | 3.068 |
| 220 | 3.533 | 3.700 |
| 240 | 4.170 | 4.360 |
| بهترین منحنی عبوری | PWM = 114.8\*current\_x^0.5157 | PWM = 113.0\*current\_y^0.5106 |

تصویر های # و #، داده های جدول # و منحنی عبوری از آنها را نشان میدهد.



تصویر 13 داده‌های جدول# و منحنی جریان سیم پیچ x و PWM



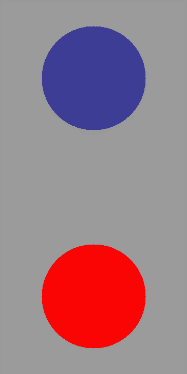
تصویر 14 داده‌های جدول# و منحنی جریان سیم پیچ y و PWM

# 7 موقعیت یابی ربات

برای یافتن موقعیت ماهی از روی فیلم وب کم، از کتابخانه‌ OpenCV و زبان برنامه نویسی پایتون 3.7 استفاده شد. برای یافتن موقعیت ماهی میتوان از روش های مختلفی استفاده کرد که در ادامه به بررسی آنها و مشکلات پیاده سازی هر کدام پرداخته می‌شود. در دو فصل بعد به تنظیمات دوربین و سیستم عامل رزبری برای بهبود عملکرد روش برچسب ArUco پرداخته می‌شود.

## 7-1 استفاده از برچسب رنگی

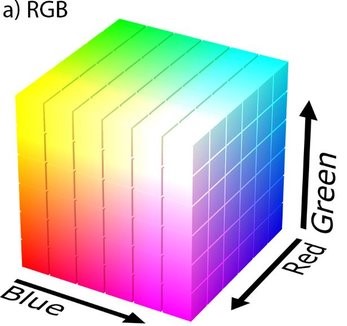
‌در این روش روی بدنه ربات یک برچسب رنگی نصب میشود و با فیلتر کردن بازه‌ای که رنگ برچسب در آن قرار دارد موقعیت و جهت گیری ربات مشخص می‌شود. تصویر # یک نمونه برچسب تولید شده توسط کد # را برای استفاده روی ماهی نشان می‌دهد. به علت ابعاد کوچک ربات و عملکرد ضعیف دوربین در زمان سرعت گرفتن ربات استفاده از برچسب دو رنگ (تصویر #) روی ربات برای ذخیره‌ی جهت و موقعیت ربات امکان پذیر نشد. مساله مهم در این روش انتخاب فضای رنگ مناسب است. در ادامه به فضا های رنگی موجود در OpenCV و نحوه ی عملکرد آنها پرداخته میشود. در طی آزمایش مشخص شد به علت تغییرات رنگ تصویر وب کم در زمان حرکت ماهی و مات شدن مسیر حرکت، استفاده از برچسب های رنگی نمیتواند نتیجه ی مطلوب را بدهد. مشکل دیگر این روش ابعاد کوچک ماهی است که امکان نصب برجسب بزرگ را میگیرد و در نتیجه خطا در تشخیص موقعیت درست برچسب ها رخ می‌دهد.



تصویر 15 برچسب رنگی استفاده شده در تست. پس زمینه در این تصویر کمی تیره تر شده

### 7-1-1 فضای رنگ BGR

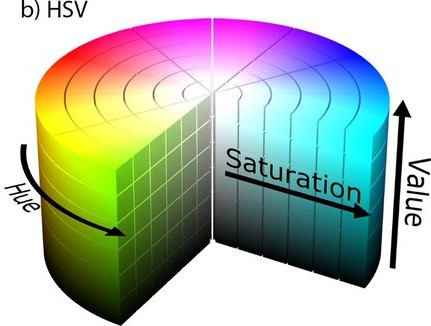
OpenCV میتواند از فضای رنگ آبی سبز قرمز یا BGR استفاده می کند که مشابه RGB است و تنها ترتیب رنگ‌ها متفاوت لیست می‌شود. تصویر # توزیع رنگ این فضا را نشان میدهد. به علت مشکلات تعیین بازه ی مورد نظر در این فضا استفاده از آن توصیه نمی‌شود.



تصویر 16 فضای RGB

### 7-1-2 فضای رنگ HSV

فضای رنگ HSV رنگ ها را مشابه تصویر # دسته بندی می‌کند. در این فضا انتخاب بازه‌ی مورد نظر به سادگی انجام می‌شود.

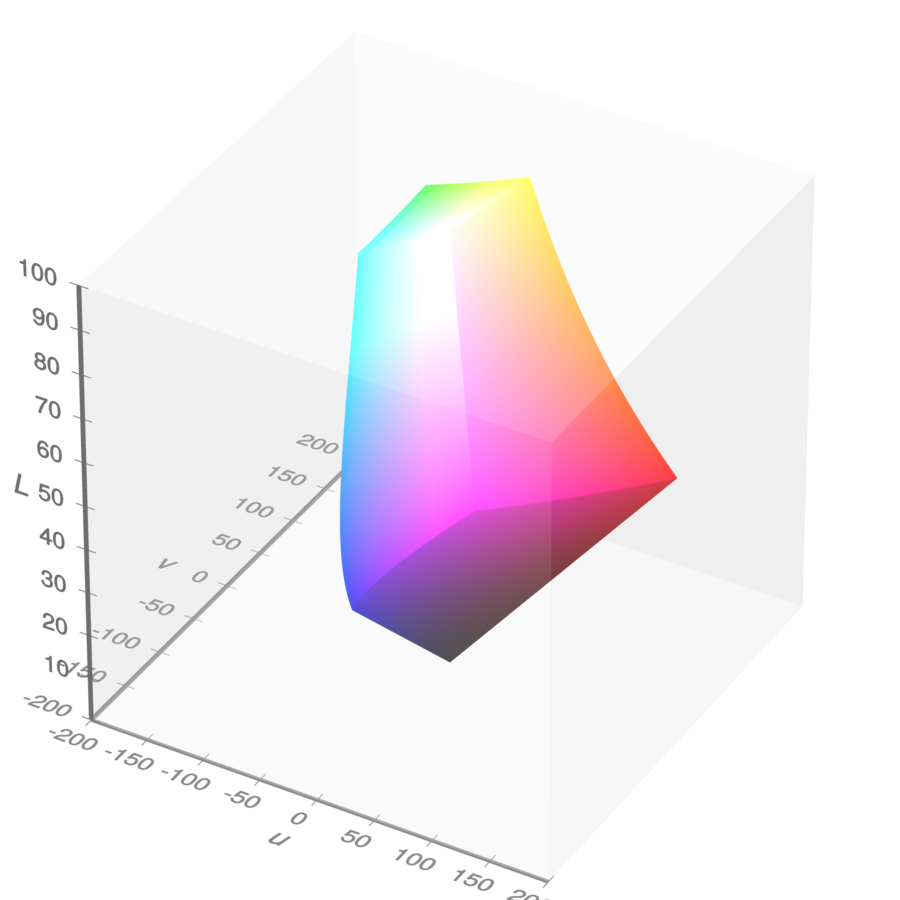
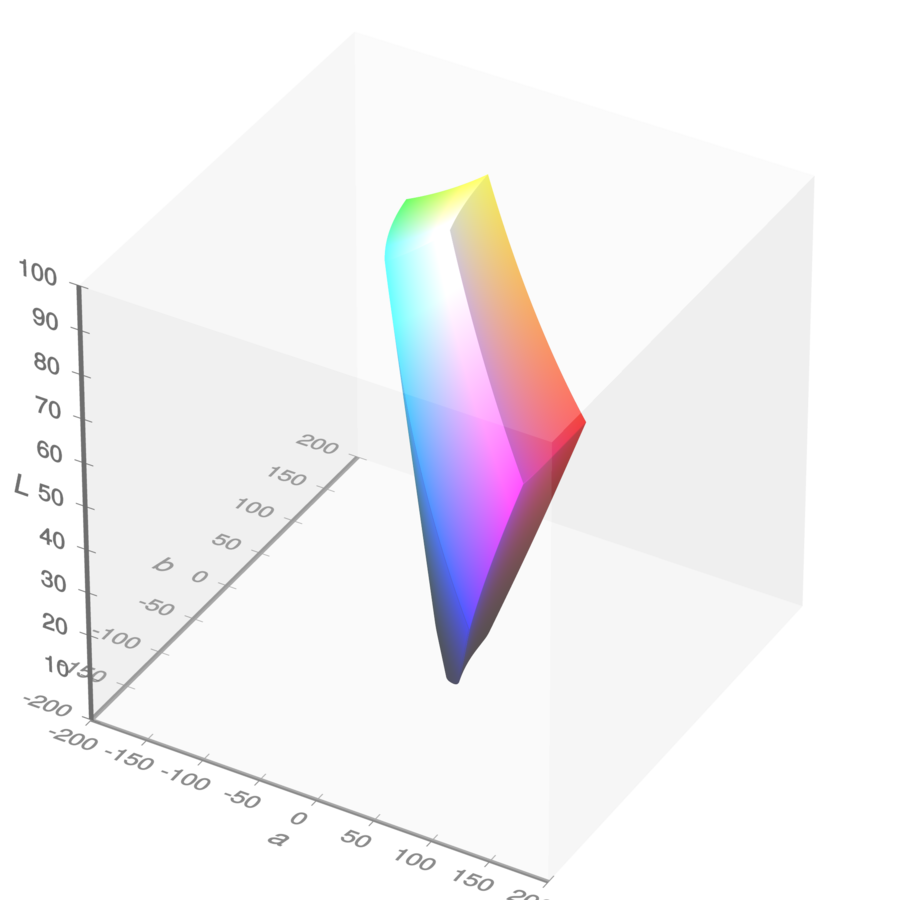
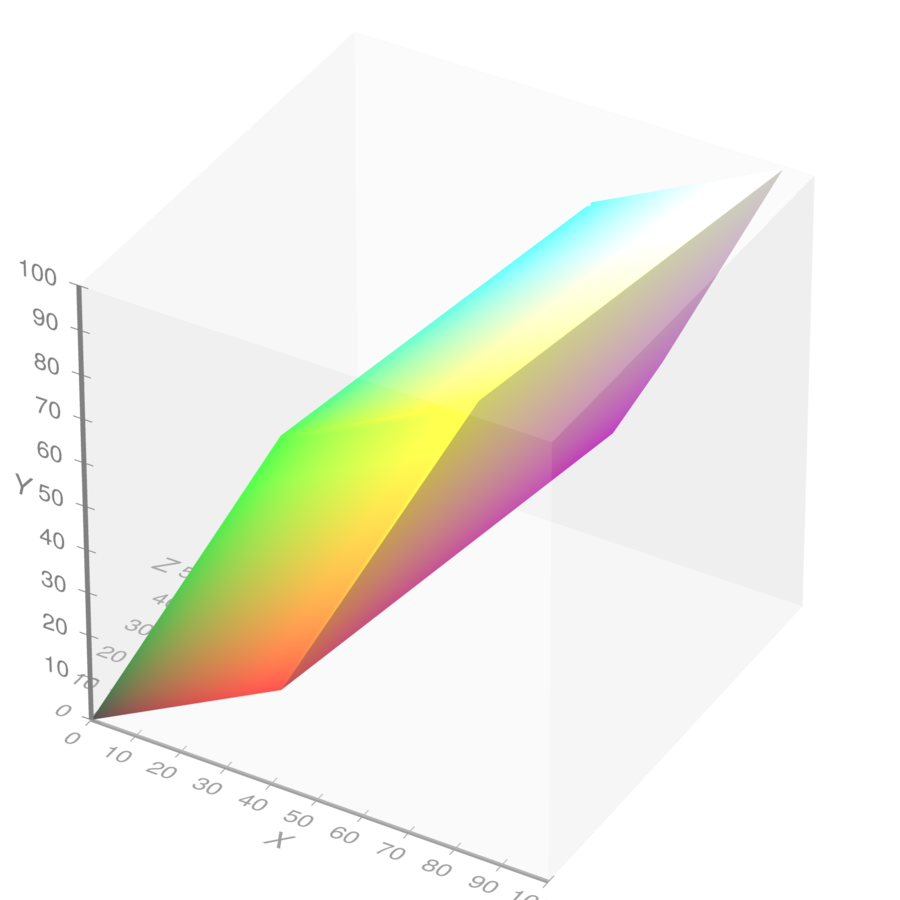


تصویر 17فضای HSV

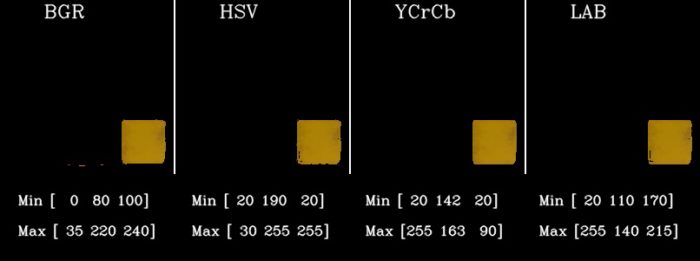
### 7-1-3 سایر فضا‌های رنگ مورد بررسی

برای فیلتر کردن رنگ برچسب بر اساس فضای رنگ دلخواه برنامه ی # که در آدرس گیت پروژه قرار دارد را نوشتم. عملکر هر یک از این فضا ها برای فیلتر رنگ برچسب در نور های مختلف محیط بررسی شد. تصویر #### تعدادی از این فضا های رنگی را نشان می‌دهد. تصویر # نمونه ای از انتخاب بازه صحیح برای فیلتر رنگ نارنجی در چند فضای رنگی را نشان می‌دهد.

در طی آزمایش در زمان سکون ماهی استفاده از فضای رنگ Lab بهترین نتیجه را در مقابل تغییرات نور محیط از خود نشان داد و بعد از آن به ترتیب YCrCb، YUV و hsv بهترین فضا های رنگ تشخیص داده شدند.



تصویر 18 تعدادی از فضا‌های رنگ OpenCV. به ترتیب از بالا سمت راست: XYZ، Lab و Luv



تصویر 19 فیاتر رنگ نارنجی در چند فضای رنگی در OpenCV

## 7-2 استفاده از برچسب آروکو ArUco

بعد از نتایج ضعیف فیلتر رنگ در مقابل نور تصویر، از برچسب ArUco استفاده شد. تصویر # یک نمونه از این برچسب را نشان می‌دهد. کد # نحوه‌ی ساخت این برچسب را در OpenCV نشان می دهد. تصویر # نمونه ای از این برچسب روی بدنه ی ماهی را نشان می‌دهد. تصویر # بردار های رسم شده توسط کد # را روی ماهی نشان می‌دهد. این برچسب توسط OpenCV و با دریافت کد عددی برچسب، ابعاد برچسب و دسته بندی این برچسب ساخته می‌شود. از علل استفاده از این برچسب قابلیت دریافت موقعیت و جهت ماهی به طور همزمان با وجود ابعاد کم ربات، عدم حساسیت به نور محیط و دقت بالای موقعیت یابی ربات را میتوان ذکر کرد. مشکل اصلی در استفاده از این برچسب مات شدن تصویر دوربین در زمان حرکت ماهی است.



تصویر 20 یک نمونه برچسب ArUco\_6x6\_250 با کد 12

## 7-3 استفاده از فیلتر های تشخیص حرکت

راه حل نهایی برای تشخیص دقیق موقعیت ماهی، تشخیص محل وقوع حرکت در تصویر است. در OpenCV به کمک توابع cv2.TrackerCSRT\_create برای استفاده از تابع دنبال تشخیص حرکت و cv2.selectRIO برای تعیین موقعیت اولیه‌ی جسم میتوان این روش را پیاده ساه کرد. کد #، پیاده سازی این روش را نشان می‌دهد. در این روش تشخیص جهت گیری ماهی ممکن نخواهد بود ولی موقعیت دقیق ماهی در زمان حرکت تضمین می‌شود.

مساله مهم در این روش استفاده از الگوریتم درست برای تشخیص دقیق موقعیت در حداقل زمان است. در جدول #، لیست توابع موجود برای تشخیص موقعیت و مشکلات آنها به طور مختصر آمده اند.

|  |  |  |  |
| --- | --- | --- | --- |
| نام الگوریتم | تابع | کیفیت تشخیص موقعیت در حرکت | فرکانس مفید CPU |
| BOOSTING | cv2.TrackerBoosting\_create() | ضعیف | 10 |
| MIL | cv2.TrackerMIL\_create() | ضعیف | 15 |
| KCF | cv2.TrackerKCF\_create() | متوسط | 200 |
| TLD | cv2.TrackerTLD\_create() | خوب | 15 |
| MEDIANFLOW | cv2.TrackerMedianFlow\_create() | خوب | 250 |
| GOTURN[[7]](#footnote-7) | cv2.TrackerMosse\_create() | - | - |
| MOSSE | cv2.TrackerBoosting\_create() | متوسط | 1400 |
| CSRT | cv2.TrackerCSRT\_create() | عالی | 30 |

جدول 3 الگوریتم‌های تعقیب

فرکانس مفید CPU از کد زیر محاسبه شده است

timer = cv2.getTickCount()

# some code  
fps = cv2.getTickFrequency() / (cv2.getTickCount() - timer);

تصویر # نمونه ای از خروجی کد تشخیص موقعیت ماهی را نشان می‌دهد.

# 8 تنظیمات دوربین

کتابخانه OpenCV امکان اعمال تنظیمات دلخواه روی دوربین را فراهم میکند. لیست توابع موجود برای اعمال تغییرات روی دوربین با دستور cap.set() قابل مشاهده اند.

جدول # برخی از این توابع کاربردی و مورد کاربرد آنها را نشان می‌دهد.

|  |  |
| --- | --- |
| عملکرد | تابع |
| CV\_CAP\_PROP\_FPS | تعداد فریم بر ثانیه |
| CV\_CAP\_PROP\_BRIGHTNESS | روشنایی تصویر |
| CV\_CAP\_PROP\_CONTRAST | کانتراست تصویر |
| CV\_CAP\_PROP\_SATURATION | وضوح رنگ تصویر |
| CV\_CAP\_PROP\_EXPOSURE | زمان نور دهی تصویر |
| CV\_CAP\_PROP\_GAIN | تنظیم gain دوربین |

جدول 4 برخی از توابع تنظیم دوربین در OpenCV

میتاسفانه بسیاری از دوربین ها از این دستورات پشتیبانی نمی کنند و در موارد زیادی نیاز به نوشتن patch برای دوربین یا نصب آن است، در نتیجه استفاده از آن ممکن است نتیجه ی مورد انتظار را ندهد.

## 8-1 ابزار v2l4-ctl

ابزار v2l4-ctl برنامه ای برای کنترل دوربین های متصل به سیستم عامل لینوکس است. به کمک این ابزار میتوان به لیست کاملی از متغیر های قابل تغییر برای هر نوع دوربین متصل به دستگاه دست یافت و آنها را به دلخواه تغییر داد [1] [2].

نمونه خروجی این ابزار برای وب کم مورد استفاده در این پژوهش به صورت زیر است.

از خروجی برنامه مشخص میشود که پارامتر های قابل تغییر برای دوربین عبارتند از #

## 8-2 تست عملکرد دوربین

برای بررسی عملکرد دوربین در فیلم برداری از هدف متحرک و ارایه تصویر با کیفیت نیاز به طراحی تست وجود دارد. #

# 9 تنظیمات لینوکس

سیستم عامل لینوکس برنامه ها را بر اساس اولویت، در 40 سطح دسته بندی می‌کند و منابع موجود را بین آنها تقسیم میکند[[8]](#footnote-8). برنامه ها بر اساس اهمیت ضریب -19 تا 20 میگیرند و اولویت آنها با افزایش ضریب آنها کاهش می‌یابد.

برای دیدن اولویت برنامه می‌توان از دستور زیر استفاده کرد

$ ps -fl -C "python test.py"

F S UID PID PPID C PRI NI ADDR SZ WCHAN STIME TTY TIME CMD  
0 R navid 6884 6424 99 80 0 - 1556 - 13:45 pts/3 00:05:54 python test.py

ستون NI اولویت برنامه را نشان می‌دهد که در این مورد برنامه در سطح صفر است.

برای اجرای برنامه با اولویت دلخواه از دستور زیر می‌توان استفاده کرد. در مثال زیر اولویت از صفر به 5- افزایش یافته است.

# nice -n --5 python test.py

# 10 موازی سازی برنامه

موازی سازی برنامه به معنی شکستن اطلاعات برنامه به اجزای کوچکتر و محاسبه همزمان آنها با هدف اجرای سریعتر آن اطلاق میشود. چند رشته ای کردن[[9]](#footnote-9) برنامه حالت خاصی از موازی سازی در نظر گرفته میشود که با تقسیم داده ها همراه نیست. از جمله ماژول های مورد استفاده در موازی سازی در پایتون میتوان به pyCuda، multithread و thread اشاره کرد.

## 10-1 اهمیت موازی سازی برنامه

در موازی سازی برنامه، قسمت های مختلف برنامه با توجه به تنظیمات برنامه نویس بر روی چند رشته[[10]](#footnote-10) به طور موازی[[11]](#footnote-11) اجرا می‌شوند، در نتیجه در صورت وجود چند بخش زمانبر، سایر قسمت های برنامه بدون مشکل به کار خود ادامه می‌دهند.

در پروژه حاضر، دریافت فریم از دوربین روی رشته مجزا از پردازش آن انجام می شود که در نتیجه آن پس از پردازش موقعیت ربات زمانی برای درخواست دریافت ورودی جدید از دوربین تلف نمی‌شود.

## 10-2 موازی سازی در پایتون

در پایتون با استفاده از ماژول thread به ساده ترین شکل ممکن میتوان برنامه را به صورت چند رشته ای اجرا کرد.

تنظیم زمان و منحنی دامنه و حداکثر آن و مساله زاویه

منحنی برای سه ربات – اتمالا رابطه با وزن و طول دم ربات – انتظار افرایش با وزن و کاهش با دم – تست ساچمه در ربات

نتایج برای ربات های مختلف

8 کد سمت اردوینو

تفاوت کد رایانه و رزبری

نصب پیش نیاز ها

محاسبات میدان در کد پایتون با منحنی عبوری از نقاط جدول بالا pwn مناسب را محاسبه و به آردوینو ارسال میکند.

|  |
| --- |
| Still Image: Up to 16 Megapixel, 4608x3456 (Software Interpolated) |
| Image Sensor: 1/6"CMOS, 640×480 pixels |
| Frame Rate: 30 fps@160x120, @320x240, @640x480 |
| Lens: F=2.4, f=3.5 mm |
| View Angle: 60° |
| Focus Range: Automatic Focus, 10 cm to infinity |
| Exposure Control: Automatic |
| White Balance: Automatic |
| Microphone: Built-in |
| Computer interface: USB 2.0 |
| System Requirements: Windows XP / Vista / 7 / 8 / 8.1 / 10 |

sudo apt purge libreoffice\*

sudo apt clean

sudo apt autoremove

کد #: ساخت برچسب رنگی

import cv2  
import numpy as np  
  
img = np.ones((720, 360, 3), np.uint8)\*255  
img = cv2.cvtColor(img, cv2.COLOR\_BGR2HSV)  
  
blue = (120, 150, 150)  
red = (0, 250, 250)  
  
img = cv2.circle(img, (180, 150), 100, blue, -1)  
img = cv2.circle(img, (180, 720-150), 100, red, -1)  
  
img = cv2.cvtColor(img, cv2.COLOR\_HSV2BGR)  
  
cv2.imwrite('test.png', img)

کد#: ساخت برچسب ArUco

import cv2  
from cv2 import aruco  
  
aruco\_dict = aruco.Dictionary\_get(aruco.DICT\_6X6\_250)  
parameters = aruco.DetectorParameters\_create()  
  
code = 12  
img = cv2.aruco.drawMarker(aruco\_dict, code, 250)  
cv2.imwrite(str(code)+'.png', img)

کد #: tracker

import cv2  
  
if \_\_name\_\_ == '\_\_main\_\_':  
   
 tracker = cv2.TrackerCSRT\_create()  
 video = cv2.VideoCapture(0)  
   
 \_, frame = video.read()  
   
 bbox = cv2.selectROI(frame, False)  
   
 while True:  
 ok, frame = video.read()  
 if not ok:  
 break  
   
 timer = cv2.getTickCount()  
 ok, bbox = tracker.update(frame)  
   
 if ok:  
 # Tracking success  
 p1 = (int(bbox[0]), int(bbox[1]))  
 p2 = (int(bbox[0] + bbox[2]), int(bbox[1] + bbox[3]))  
 cv2.rectangle(frame, p1, p2, (255, 0, 0), 2, 1)  
 else:  
 # Tracking failure  
 cv2.putText(frame, "Tracking failure detected", (100, 80), cv2.FONT\_HERSHEY\_SIMPLEX, 0.75, (0, 0, 255), 2)  
   
 cv2.imshow("Tracking", frame)  
   
 k = cv2.waitKey(1) & 0xff  
 if k == 27:

break

[+] <https://www.electrical4u.com/rl-series-circuit/> : محاسبه ضریب خود القایی حلقه

[+] <http://hyperphysics.phy-astr.gsu.edu/hbase/magnetic/curloo.html/> : منبع عکس 1

[+] <https://phys.cst.temple.edu/~meziani/Temple_University_3He_Lab_files/helmholtz.htm> : منبع عکس 2

[+] <https://www.electrical4u.com/rl-series-circuit/> : منبع عکس 3

[+] <http://a4tech.com/product.aspx?id=146> : مشخصات دوربین

[+] <https://stackoverflow.com/questions/45713327/is-there-a-way-to-adjust-shutter-speed-or-exposure-time-of-a-webcam-using-python> : تنظیمات دوربین رزبری

[+] <https://www.thegeekstuff.com/2013/08/nice-renice-command-examples/?utm_source=tuicool> : تغییر اهمیت برنامه در لینوکس

[+] <http://t-sato.in.coocan.jp/compliance/magnetic-field/helmholtz-coil.html> عکس میدان

[+] <https://www.learnopencv.com/object-tracking-using-opencv-cpp-python> : تشخیص حرکت

[+] <https://www.kurokesu.com/main/2016/01/16/manual-usb-camera-settings-in-linux/> v4l2

[+] [https://stackoverflow.com/questions/45713327/is-there-a-way-to-adjust-shutter-speed-or-exposure-time-of-a-webcam-using-python v4l2](https://stackoverflow.com/questions/45713327/is-there-a-way-to-adjust-shutter-speed-or-exposure-time-of-a-webcam-using-python%20v4l2)

<https://www.learnopencv.com/color-spaces-in-opencv-cpp-python/>

<https://commons.wikimedia.org/wiki/File:YCbCr-CbCr_Scaled_Y0.png>

<https://commons.wikimedia.org/wiki/File:SRGB_gamut_within_CIELUV_color_space_isosurface.png>

<https://www.makeuseof.com/tag/raspberry-pi-performance-tips/>

<https://www.pyimagesearch.com/2015/12/28/increasing-raspberry-pi-fps-with-python-and-opencv/>

# References

|  |  |
| --- | --- |
| [1] | "kurokesu," [Online]. Available: https://www.kurokesu.com/main/2016/01/16/manual-usb-camera-settings-in-linux/. |
| [2] | [Online]. Available: https://www.mankier.com/1/v4l2-ctl. |

1. Helmholtz coil [↑](#footnote-ref-1)
2. Close loop [↑](#footnote-ref-2)
3. ROI or Region of Interest [↑](#footnote-ref-3)
4. Magnetic resonance imaging [↑](#footnote-ref-4)
5. Maxwell coil [↑](#footnote-ref-5)
6. PWM [↑](#footnote-ref-6)
7. الگوریتم تعقیب مخصوص یادگیری ماشین در OpenCV . نیازمند تربیت مدل است. [↑](#footnote-ref-7)
8. Process priority [↑](#footnote-ref-8)
9. Multi threading [↑](#footnote-ref-9)
10. thread [↑](#footnote-ref-10)
11. parallel [↑](#footnote-ref-11)