



دانشگاه صنعتی امیرکبیر

(پلی تکنیک تهران)

دانشکده مهندسی برق

پایان نامه کارشناسی

گرایش کنترل

فروود یک روبات کوادروتور با استفاده از پردازش تصویر در فضای بسته

نگارش

سیده پریسا داج خوش

استاد راهنما

دکتر فرزانه عبدالله

آبان ۱۳۹۶

به نام خدا

تاریخ: ۱۳۹۶

تعهدنامه اصالت اثر



اینجانب سیده پریسا داج خوش متعهد می‌شوم که مطالب مندرج در این پایان نامه حاصل کار پژوهشی اینجانب تحت نظارت و راهنمایی استادی دانشگاه صنعتی امیرکبیر بوده و به دستاوردهای دیگران که در این پژوهش از آنها استفاده شده است مطابق مقررات و روال متعارف ارجاع و در فهرست منابع و مأخذ ذکر گردیده است. این پایان نامه قبلاً برای احراز هیچ مدرک هم‌سطح یا بالاتر ارائه نگردیده است.

در صورت اثبات تخلف در هر زمان، مدرک تحصیلی صادر شده توسط دانشگاه از درجه اعتبار ساقط بوده و دانشگاه حق پیگیری قانونی خواهد داشت.

کلیه نتایج و حقوق حاصل از این پایان نامه متعلق به دانشگاه صنعتی امیرکبیر می‌باشد. هرگونه استفاده از نتایج علمی و عملی، و اگذاری اطلاعات به دیگران یا چاپ و تکثیر، نسخه‌برداری، ترجمه و اقتباس از این پایان نامه بدون موافقت کتبی دانشگاه صنعتی امیرکبیر ممنوع است. نقل مطالب با ذکر مأخذ بلامانع است.

سیده پریسا داج خوش

من لم يشكر المخلوق

لم يشكر الخالق

بر خود لازم می‌دانم که به پاس نعمت توفیق و ادای وظیفه، صمیمانه‌ترین قدردانی‌های خویش را نثار استاد ارجمند سرکار خانم دکتر فرزانه عبداللہی بنمایم که دلسوزانه مجموعه علم و دانش خویش را در اختیار من گذاشتند و در تهیه این پایان نامه، اگر موفقیتی حاصل گشت، قسمت اعظم آن، مدیون راهنمایی‌های بی‌شایبی آن بزرگوار گرانقدر بوده است.

از تمامی اعضای محترم آزمایشگاه کنترل روبات‌های چندوسیله‌ای که طی مدت انجام پروژه، افتخار همکاری‌شان را داشته‌ام، نهایت تشکر را دارم. از جمله امیرحسین علیخانی، سارا قائمی، نیما محمودی، سعید بیباک، حسن مجاب، حیدر سودانی، محمد خوش‌دهان، آنديا فتحي و سايرين. به طور ویژه از دوستان عزيزم دنيا شرف‌الدين‌زاده و ليلا موسى پورکه در تک تک مراحل انجام پروژه و ارائه‌ی دفاعيه بنده را ياري فرمودند، بسيار سپاس‌گزارم. همچنان از اعضای خانواده که نهایت همکاري و لطف را با اينجانب داشتند، صمیمانه تشکر می‌نمایم.

چکیده

در این گزارش به بررسی نحوه فرود این گونه پرنده‌ها و ارائه‌ی چالش‌ها و راهکارهای موجود در این زمینه پرداخته می‌شود. در واقع فرض پروژه بدین گونه خواهد بود که پهپاد در حال پرواز در مسیری با جهت‌گیری نسبتاً ثابتی به صورت کاملاً خودکار در صورت اتمام ماموریت و یا نیاز به شارژ مجدد فرود می‌آید. این فرایند را می‌توان به به مراحل زیر تقسیم کرد:

- تشخیص محل فرود با استفاده از پردازش تصویر.
- جهت گیری مناسب در راستای مسیر.
- کاهش سرعت تا رسیدن به حالت استقرار به صورت معلق در ارتفاع مناسب.
- قرارگیری در موقعیت مناسب برای فرود و کاهش ارتفاع و تکمیل پروسه‌ی فرود.

پردازش‌گر با تشخیص علامت، به کمک روش‌های پردازش تصویر و با کمک دوربین، نحوه‌ی حرکت خود را به گونه‌ای تنظیم می‌کند که بتواند در محل مربوطه فرود مناسبی داشته باشد. بنابراین، بایستی روبات مجهز به یک دوربین رو به پایین و ترجیحاً RGB باشد تا بتوان به وسیله‌ی آن علامت را تشخیص داد. همچنین یک بورد Raspberry Pi 3 بر روی روبات به عنوان پردازنده قرار گرفته است تا بتواند پردازش لازم را در داخل روبات انجام دهد.

واژه‌های کلیدی:

پهپاد چهار موتوره، فرود خودکار، پردازش تصویر، رزبری پای ۳.

صفحه

فهرست مطالب

۱	فصل اول مقدمه
۲	۱-۱- هدف پژوهش
۳	۱-۱- تاریخچه
۵	۱-۳- ساختار یک کوادکوپتر
۷	۱-۴- کاربردها
۸	۱-۵- چالش‌ها و پیشنهادهای موجود برای حل مساله
۹	۱-۵-۱- مدل‌سازی
۹	۱-۵-۲- موقعیت یابی
۱۰	۱-۵-۳- روش‌های کنترل پرواز
۱۳	فصل دوم مدل‌سازی دینامیکی
۱۴	۲-۱- مدل کوادکوپتر
۱۷	۲-۲- ماتریس‌های دوران
۱۸	۲-۳- مدل دینامیکی چرخش سیستم
۲۱	۲-۴- مدل دینامیکی جابه جایی سیستم
۲۲	۲-۵- مدل نهایی
۲۳	۲-۶- خطی‌سازی و معادلات حالت
۲۴	۲-۷-تابع تبدیل سیستم
۲۵	۲-۸- پارامترهای روبات
۲۷	فصل سوم حسگرهای
۲۸	۳-۱- سنجش زاویه
۳۰	۳-۲- سامانه موقعیت یاب جهانی (GPS) [11]
۳۱	۳-۳- فاصله یاب‌ها [13]
۳۱	۳-۳-۱- فاصله یاب نوری [14]
۳۲	۳-۳-۲- فاصله یاب صوتی [16]
Error! Bookmark not defined.	۳-۴-۴- پردازش تصویر
۳۳	۳-۴-۱- دوربین دو بعدی
۳۳	۳-۴-۲- دوربین USB
۳۴	۳-۴-۳- دوربین رزبری پای
۳۵	۳-۴-۲- دوربین کینکت [21]
۳۶	۳-۴-۳- لیزر اسکنر
۳۷	۳-۴-۴- نتیجه‌گیری : ترکیب سنسورها

۳۸	۴ فصل چهارم پردازش تصویر
۴۰	۴-۱- انواع الگو
۴۰	۴-۱-۱- علامت‌گذار نقطه‌ای [24]
۴۰	۴-۱-۲- علامت‌گذار منحنی [24]
۴۱	۴-۱-۳- QR Code
۴۲	۴-۱-۴- الگوی رنگی
۴۲	۴-۱-۵- ماتریس داده [26]
۴۳	۴-۱-۶- الگوهای واقعیت افزوده [27]
۴۴	۴-۲- الگوریتم‌های پردازش تصویر
۴۴	۴-۲-۱- فیلترسازی و [28] Thresholding
۴۵	۴-۲-۲- بخش بندی و برچسب زدن محتويات وابسته [22]
	Error! Bookmark not defined.
۴۶	۴-۳- نتیجه‌گیری
۴۶	۵ فصل پنجم موقعیت یابی و فرود
۴۸	۵-۱- کنترل موقعیت [8]
۴۹	۵-۲- کنترل زاویه [8]
۵۱	۵-۲-۱- کنترل ارتفاع
	Error! Bookmark not defined.
۵۱	۵-۲-۲- نتیجه‌گیری
۵۵	۶ فصل ششم شبیه‌سازی
۵۷	۶-۱- پارامترهای روبات
۵۷	۶-۲- شبیه‌سازی پرواز
۶۱	۶-۳- شبیه‌سازی فرود
۶۲	۶-۴- خروجی شبیه‌سازی
۶۲	۶-۴-۱- مدل دینامیکی خطی بدون نویز
۶۵	۶-۴-۲- مدل دینامیکی غیرخطی بدون نویز
۶۶	۶-۴-۳- مدل دینامیکی غیرخطی نویزی
	Error! Bookmark not defined.
۶۶	۶-۵- نتیجه‌گیری
۶۹	۷ فصل هفتم سخت افزار
۷۰	۷-۱- مکانیک
۷۰	۷-۱-۱- انتخاب جنس بدن
۷۲	۷-۲- الکترونیک
۷۲	۷-۲-۱- مغز روبات (Erle-brain 3)
۷۴	۷-۲-۲- رزبری پای
۷۶	۷-۲-۳- تغذیه

۷۷	۷-۲-۴	- موتورها و کنترل کننده‌های سرعت
۷۷	۱-۴-۲-۷	- موتورها
۸۳	۲-۴-۲-۷	ها ECS
۸۴	۷-۲-۵	- حسگرها
۸۵	۷-۲-۶	- سیستم مخابراتی برای ارتباط بی سیم روبات با کامپیوتر کنترل کننده
Error! Bookmark not defined.		۷-۳
۸۷	۸ فصل هشتم پیاده‌سازی عملی	
۸۸	۱-۸	- نرم افزار
Error! Bookmark not defined.		۸-۱-۱
۸۹	۸-۱-۲	- کتابخانه OpenCV
۹۰	۱-۲-۱-۸	- تشخیص لبه‌های تصویر
۹۰	۲-۲-۱-۸	- تشخیص دواير قرمز رنگ [39]
۹۲	۳-۲-۱-۸	- جست و جو و تشخیص یک عکس، در تصویر اصلی
۹۳	۸-۱-۳	... Aruco
۹۵	۹ فصل نهم جمع‌بندی، نتیجه‌گیری و پیشنهادات	
۹۷	۹-۱-۱	- افزودن کنترل کننده یا و
۹۷	۹-۱-۲	- استفاده از پردازش‌گرهای مجزا برای پردازش تصویر و کنترل
۹۸	۹-۱-۳	- شبیه‌سازی مدل دینامیکی زمین [40]
۹۸	۹-۱-۴	- انتقال به محیط فضای باز و استفاده از سنسور باد
۹۸	۹-۱-۵	- کنترل و شناسایی فازی
۹۹	۱۰ منابع و مراجع	
۱۰۴	۱۱ پیوست‌ها	

صفحه

فهرست اشکال

۴ شکل ۱-۱ روبات STARMAC II [2]
۵ شکل ۲-۱ تصویر روبات OS4 [3]
۶ شکل ۳-۱ نحوه اعمال نیرو به موتورها برای تغییر ارتفاع
۶ شکل ۴-۱ نحوه اعمال نیرو به موتورها برای چرخش به طرفین
۷ شکل ۵-۱ نحوه اعمال نیرو به موتورها برای حرکت به طرفین
۸ شکل ۶-۱ کوادروتور تصویربردار [4]
۱۵ شکل ۱-۲ ورودی‌های کنترلی کوادروتور [7]
۱۶ شکل ۲-۲ گشتاورهای یاو، پیچ و رول [7]
۲۹ شکل ۱-۳ مژول GY9250 [10]
۳۰ شکل ۲-۳ سنسور GPS مورد استفاده در این پروژه [12]
۳۲ شکل ۳-۳ مژول SRF02 [17]
۳۴ شکل ۴-۳ یک مدل دوربین USB ساخت شرکت Logitech [18]
۳۵ شکل ۵-۳ دوربین رزیبری پای [20]
۳۶ شکل ۶-۳ دوربین کینکت [21]
۳۶ شکل ۷-۳ یک لیزر اسکنر کوچک با برد ۵ متر [23]
۴۰ شکل ۱-۴ محل فرود به شکل H [24]
۴۱ شکل ۲-۴ محل فرود به شکل سه مثلث تو در تو [24]
۴۱ شکل ۳-۴ نمونه یک الگوی QR Code [25]
۴۲ شکل ۴-۴ نمونه یک الگوی ماتریس داده [26]
۴۳ شکل ۵-۴ نمونه یک الگوی واقعیت افزوده [27]
۴۳ شکل ۶-۴ استفاده از یک الگو برای اضافه کردن شخصیت مجازی به تصویر [27]
۴۷ شکل ۱-۵ نمای کلی کنترل کننده موقعیت
۵۲ شکل ۲-۵ محل فرود با ابعاد ۱ در ۱ متر و تصویر ۹۶۰ در ۹۶۰ پیکسل
۵۴ شکل ۳-۵ بلوک دیاگرام فرود روبات
۵۶ شکل ۱-۶ نمایی از محیط گرافیکی نمایش کنترل کوادروتور در نرم افزار SolidWorks
۵۸ شکل ۲-۶ شبیه‌سازی پرواز روبات در نرم افزار متلب
۵۹ شکل ۳-۶ پاسخ پله و پارامترهای کنترلر PID محور X
۵۹ شکل ۴-۶ پاسخ پله و پارامترهای کنترلر PID محور Y
۵۹ شکل ۵-۶ پاسخ پله و پارامترهای کنترلر PID محور Z
۶۰ شکل ۶-۶ پاسخ پله و پارامترهای کنترلر PID محور رول

..... ۶۰	شکل ۷-۶ پاسخ پله و پارامترهای کنترلر PID محور پیچ
..... ۶۰	شکل ۸-۶ پاسخ پله و پارامترهای کنترلر PID محور یاو
..... ۶۲	شکل ۹-۶ شبیه‌سازی فرود روبات در نرم افزار متلب.
..... ۶۳	شکل ۱۰-۶ حرکت مطلوب مدل خطی بدون نویز از ابتدا تا انتهای
..... ۶۳	شکل ۱۱-۶ حرکت واقعی مدل خطی بدون نویز از ابتدا تا انتهای
..... ۶۴	شکل ۱۲-۶ مسیر حرکت روبات خطی بدون نویز در صفحه xy
..... ۶۵	شکل ۱۳-۶ حرکت مطلوب مدل غیرخطی بدون نویز از ابتدا تا انتهای
..... ۶۵	شکل ۱۴-۶ حرکت واقعی مدل غیرخطی بدون نویز از ابتدا تا انتهای
..... ۶۶	شکل ۱۵-۶ مسیر حرکت روبات غیرخطی بدون نویز در صفحه xy
..... ۶۷	شکل ۱۶-۶ حرکت مطلوب مدل غیرخطی نویزی از ابتدا تا انتهای
..... ۶۷	شکل ۱۷-۶ حرکت واقعی مدل غیرخطی نویزی از ابتدا تا انتهای
..... ۶۸	شکل ۱۸-۶ مسیر حرکت نویزی غیرخطی
..... ۷۲	شکل ۱-۷ بلوک دیاگرام الکترونیک روبات [29]
..... ۷۴	شکل ۲-۷ تصویری از ۳ Erle-Brain و مشخصات آن [12]
..... ۷۵	شکل ۳-۷ Raspberry Pi 3 [32]
..... ۷۶	شکل ۴-۷ ماژول تغذیه [33] APM Power Module
..... ۷۷	شکل ۵-۷ بلوک دیاگرام اتصال موتور و ESC
..... ۸۰	شکل ۶-۷ MT2216 و کاربرد آن در کوادراتور [34]
..... ۸۱	شکل ۷-۷ نمودار نیروی بالابرده و توان خروجی پره نسبت به سرعت چرخش [34]
..... ۸۱	شکل ۸-۷ نمودار راندمان، توان خروجی و سرعت چرخش موتور نسبت به جریان مصرفی [34]
..... ۸۲	شکل ۹-۷ رابطه میان موتور، مدار راهانداز و پره برای ایجاد نیروی بالابرندم
..... ۸۳	شکل ۱۰-۷ تصویر ESC مورد انتخاب [35]
..... ۸۵	شکل ۱۱-۷ کنترل موتور به وسیله‌ی مدولاسیون عرض پالس
..... ۹۰	شکل ۱-۸ یافتن لبه‌های تصویر با استفاده از فیلتر Canny Edge Detector
..... ۹۱	شکل ۲-۸ تشخیص دوایر قرمز رنگ
..... ۹۱	شکل ۳-۸ یافتن دوایر قرمز رنگ در تصویر نویزی
..... ۹۲	شکل ۴-۸ علامت فرود
..... ۹۳	شکل ۵-۸ تشخیص نقاط کلیدی تصویر مرجع در تصویر دوربین
..... ۹۴	شکل ۶-۸ نمونه یک صفحه کالیبره کننده دوربین [7]

صفحه

فهرست جداول

جدول ۱-۲ پارامتر های اندازه‌گیری شده مدل واقعی	۲۵
جدول ۱-۵ تعداد پیکسل نمایش هر بارکد در ارتفاع های متفاوت و تعداد بارکدهای مورد مشاهده	۵۲
جدول ۱-۶ پارامترهای اندازه‌گیری شده مدل واقعی	۵۷
جدول ۱-۷ این جدول خواص مختلف ماده را با امتیاز دهی از ۱۰ مقایسه می‌نماید	۷۱
جدول ۲-۷ جدول امکانات ۳ [31] [12] [30] Erle-brain	۷۳
جدول ۳-۷ جدول ویژگی های الکترونیک مازول APM Power Module	۷۶
جدول ۴-۷ مقایسه موتورهای بدون جاروبک DC با موتور DC معمولی	۷۸
جدول ۵-۷ اتصالات Raspberry Pi	۸۴

فهرست علائم

زاویه رول	ϕ
زاویه پیچ	θ
زاویه یاو	ψ
مکان در راستای محور x	x
مکان در راستای محور y	y
مکان در راستای محور z	z
سرعت زاویه‌ای حول محور x	p
سرعت زاویه‌ای حول محور y	q
سرعت زاویه‌ای حول محور z	r
گشتاوتر ورودی حول محور x	τ_{roll}
گشتاوتر ورودی حول محور y	τ_{pitch}
گشتاوتر ورودی حول محور z	τ_{yaw}
نیروی حاصل از مجموع چهار موتور	τ_{thrust}
ممان اینرسی حول محور x	I_{xx}
ممان اینرسی حول محور y	I_{yy}
ممان اینرسی حول محور z	I_{zz}
گشتاوتر موتور i ام	M_i
نیروی موتور i ام	F_i
ضریب گشتاوتر	k_M
ضریب نیرو	k_F

سرعت زاویه‌ای موتور آم	Ω_i
ممان اینرسی ملخ‌ها	J_r
شتاب گرانش	g
انرژی پتانسیل	T
انرژی جنبشی	V
ضریب انтگرال گیر	K_i
ضریب تناسبی	K_p
ضریب مشتق گیر	K_d
ضریب پیش خورد	K_{ff}

فصل اول

مقدمه

مقدمه

کوادرورتور^۱ یا پهپاد^۲ چهار یکی از انواع وسایل پرنده بدون سرنشیین است که زیر مجموعه‌ای از مجموعه دسته‌ی چند موتوره‌ها است که به دلیل کمک گرفتن از چهار پره برای نیروی پیش رانش، چهار موتوره نامیده می‌شود. این روبات به علت داشتن قدرت مانور فوق العاده و پروازهای با تعادل بالا از کاربردهای گسترده‌ای برخوردار است.

در سال‌های اخیر توجه شرکت‌ها، دانشگاه‌ها و مراکز تحقیقاتی بیش از پیش به این نوع از پهپادها جلب شده‌است و لذا هر روز پیشرفت چشم گیری در امکانات و پرواز این نوع از پرنده‌ها مشاهده می‌شود.

۱-۱- هدف پروژه

در این پروژه قصد داریم یک روبات پرنده چهار موتوره را که در حالت پرواز قرار دارد، در زمان فرود، به سمت هدف نهایی برای فرود هدایت کنیم. این هدف را با استفاده از پردازش تصویر، شناسایی کرده، سپس در محل مربوطه فرود دقیقی انجام دهیم. از چالش‌های این پروژه می‌توان انتخاب طرح مناسب برای تشخیص و قرار گیری در محل فرود، انجام کنترل رفتار و کنترل ارتفاع روبات برای فرود در محل مناسب و پردازش تصویر با سرعت بالا با استفاده از دوربین دیجیتال رنگی را نام برد.

نحوه انجام پروژه بدین گونه است که ابتدا پارامترهای مورد نیاز برای ساخت و آماده‌سازی روبات و محل فرود شناسایی و مشخص می‌شوند. سپس مدل‌سازی دینامیکی روبات و پس از آن طراحی کنترل کننده‌ی سرعت موتورها و در نتیجه زاویه و موقعیت فیزیکی روبات انجام می‌شود. همچنین روش‌های مختلف پردازش تصویر مورد بررسی قرار می‌گیرد. در نهایت شبیه‌سازی فرود و پیاده‌سازی آن در برنامه‌ی این پروژه خواهد بود.

¹ Quadrotor

² پرنده‌ی هدایت پذیر از دور

³ Multirotor

۱-۲ - تاریخچه

برای اولین بار دو برادر فرانسوی به نام‌های جک^۴ و لویس برگه^۵، مدل آزمایشی یک پهپاد چند موتوره به نام کوادکوپتر^۶ را در سال ۱۹۰۷ ساخته و مورد آزمایش قرار دادند. با وجود اینکه آن‌ها موفق به ثابت نگه داشتن روبات نشدند، اما توانستند آن را به صورت ثابت پرواز دهند. پس از آن، در سال ۱۹۲۰ مهندسی فرانسوی به نام اتین اومنیشن^۷ اولین بالگرد چهار ملخه را اختراع کرد و مسافت ۳۶۰ متر را با آن پرواز نمود. سپس در همان سال با استفاده از روش تنظیم X، مسافت یک کیلومتر را در مدت هفت دقیقه و چهل ثانیه پرواز کرد.

در حدود دو سال بعد، جورج بودزا^۸ در آمریکا تعدادی پهپاد چهار موتوره برای ارتش ساخته و مورد آزمایش قرار داد که با وجود قابلیت کنترل و حرکت در ارتفاع کم در سه بعد، پرواز با آن بسیار دشوار بود. این پروژه به دلیل هزینه‌های بالا ادامه پیدا نکرد.

همچنین در سال‌های ۱۹۵۶ و ۱۹۵۸ مدل‌های دیگری طراحی و آزمایش شدند. به دلیل پیچیده بودن طراحی کنترل‌کننده برای پرنده‌های چهار موتوره در مقایسه با هلیکوپترها، تحقیقات بر روی پرنده‌های چهار موتوره متوقف شد و تمرکز محققان بر روی ساخت هلیکوپتر قرار گرفت. اما در سالهای اخیر با توجه به پیشرفت الکترونیک و امکان پیاده‌سازی سیستم‌های کنترلی در ابعاد کوچک، محققان مجدداً به طراحی رباتهای چهار موتوره پرداخته‌اند. [۱]

⁴ Jacques

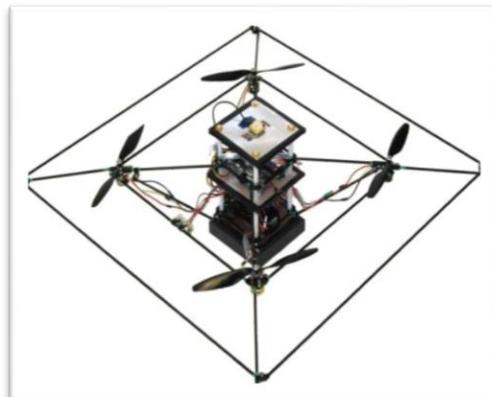
⁵ Louis Breguet

⁶ Quadcopter

⁷ Etienne Oehmichen

⁸ Botheza George

پروژه مزیکوپتر^۹، در سال ۱۹۹۹ آغاز شد و در سال ۲۰۰۱ به پایان رسید. این پروژه به مطالعه امکان ساخت کوادروتور در مقیاس سانتی متر پرداخت و هدف آن کاربرد در مناطق وسیع و جمع آوری اطلاعات جوی و یا برای هواشناسی بود. پس از آن محققان دانشگاه استنفورد در سال ۲۰۰۴ روبات پرنده چهارموتورهای به نام استارمک یک^{۱۰} طراحی نمودند. هدف از این طراحی بررسی امکان پرواز یک روبات پرنده چهارموتوره به صورت پایدار بود. با موفقیت این پروژه، نسخه‌ی جدیدی از این روبات به نام استارمک دو^{۱۱} (شکل ۱-۱) در سال ۲۰۰۶ ساخته شد تا برای تحقیق بر روی کنترل سیستمهای چند عامله مورد استفاده قرار گیرد.



شکل ۱-۱ روبات [2] STARMAC II

در سال ۲۰۰۵ در دانشگاه EPFL تحقیقات بر روی ساخت روبات کوادروتور برای انجام ماموریت به صورت خودکار، شروع شد و به همین منظور روبات OS4 (شکل ۲-۱) طراحی و ساخته شد. این روبات یک پردازنده و حسگرهای مختلف را بر روی خود حمل می‌کرد تا به کمک دوربین‌های نصب شده در آزمایشگاه و سیستم موقعیت یابی مرجع، امکان پرواز به صورت خودمختار میسر شود.

^۹ Mesicopter

^{۱۰} STARMAC I

^{۱۱} STARMAC II



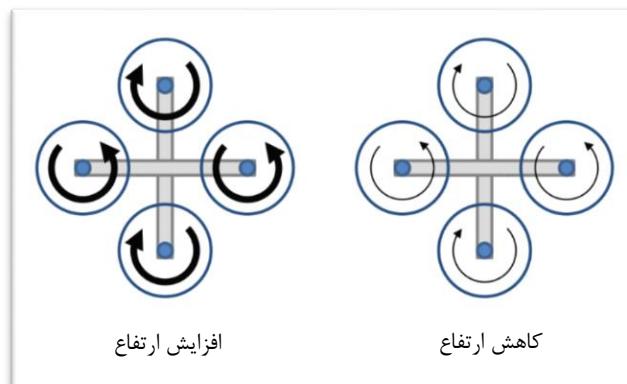
شکل ۱-۲ تصویر روبات [3]

۱-۳- ساختار یک کوادکوپتر

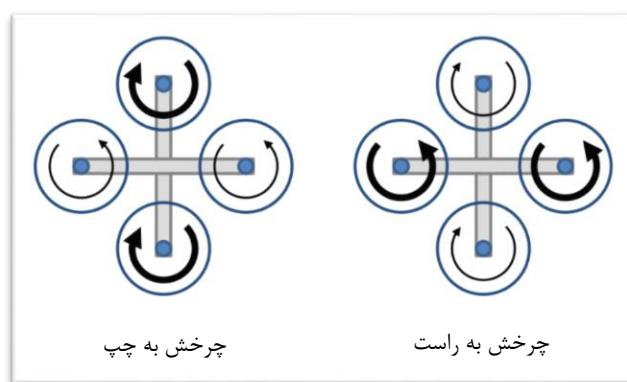
کوادکوپترها همانند انواع دیگر وسایل پرنده از ایجاد اختلاف فشار در اتمسفر پیرامون خود برای بلند شدن و حرکت در هوا استفاده می‌کنند. همان طور که هلیکوپترها به کمک پره اصلی این اختلاف فشار را ایجاد می‌کنند و نیروی برآی خود را تامین می‌نمایند. در هلیکوپتر بدلیل وجود نیروی عمل و عکس العمل، پس از اینکه پره اصلی شروع به چرخش می‌کند، با برخورد مولکول‌های هوا به این پره، نیرویی با جهت مخالف جهت چرخش پره به پره و در ادامه به شفت متصل به پره اعمال می‌شود (نیروی گشتاور) و باعث چرخش هلیکوپتر به دور خود می‌گردد. حال برای حل این مشکل از پره دم هلیکوپتر استفاده می‌شود تا نیرویی را تولید کند که مانع چرخش هلیکوپتر به دور خود شود. حال اگر هلیکوپتر به جای داشتن یک پره اصلی از دو پره اصلی که خلاف جهت یکدیگر بچرخدن استفاده می‌نمود، به دلیل خنثی شدن دو نیروی گشتاور توسط یکدیگر، دیگر هلیکوپتر به دور خود نمی‌چرخید، مانند هلیکوپترهای کبری. با توجه به توضیحات داده شده راحت‌تر می‌توان به ساختار کوادکوپترها اشاره نمود.

نحوه ایجاد فرامین کنترلی در کوادکوپترها به این صورت است که، برای تغییر ارتفاع، از تغییر سرعت چرخش همه موتورها استفاده می‌شود (شکل ۱-۳) و باعث کم یا زیاد شدن اختلاف فشار به وجود آمده می‌شود. برای چرخش کوادکوپتر به دور خود و به صورت درجا (شکل ۱-۴)، دو پره هم جهت با سرعت کمتر و دو پره هم جهت دیگر با سرعت بیشتر می‌چرخدند و نیروی گشتاور به یک سمت ایجاد می‌شود و

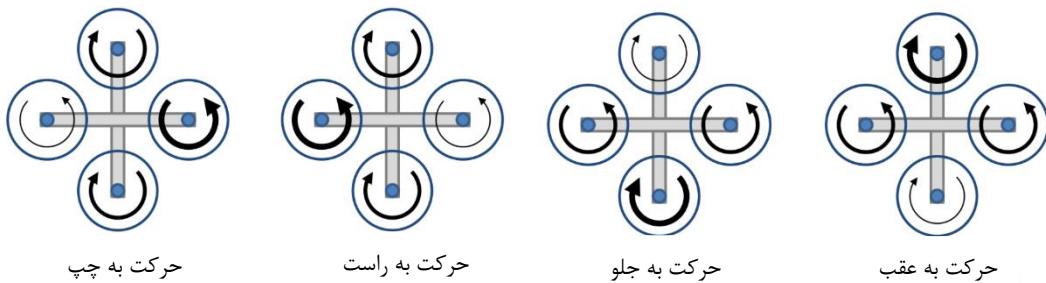
اختلاف فشار همانند قبل است (زیرا دو پره با سرعت کمتر و دو پره دیگر به همان نسبت با سرعت بیشتر می‌چرخند) لذا کوادکوپتر در ارتفاع ثابت به دور خود می‌چرخد. برای حرکت کوادکوپترها در جهت‌های مختلف (عقب، جلو، چپ و راست) توسط کم و زیاد کردن سرعت موتورها کوادکوپتر را از حالت افقی خارج کرده و باعث حرکت آن می‌گردند. (شکل ۱-۵)



شکل ۳-۱ نحوهی اعمال نیرو به موتورها برای تغییر ارتفاع



شکل ۴-۱ نحوهی اعمال نیرو به موتورها برای چرخش به طرفین



شکل ۵-۱ نحوه اعمال نیرو به موتورها برای حرکت به طرفین

تمامی این فرامین توسط بورد کنترلر داده می‌شود که در فصل (۵) به جزئیات طراحی این بورد پرداخته خواهد شد.

۱-۴- کاربردها

امروزه مالتی روتورها به ویژه مدل‌های چهارپره که به کوادکوپتر شهرت دارند یکی از پرکاربردترین وسایل پرنده بدون سرنشین هستند. به عنوان مثال می‌توان به کاربردهای گسترده تصویربرداری هوایی، نقشه برداری، جاسوسی و تفریحی و ... اشاره نمود.

با گسترده‌تر شدن روز افرون جلوه‌های بصری در تبلیغات و فیلم‌های سینمایی و تلویزیونی استفاده از وسایل پرنده و تصویربرداری هوایی بیش از پیش مورد توجه قرار گرفته است. پیش از وجود کوادکوپترها به صورت صنعتی، از هلیکوپترهای رادیو کنترلی به عنوان وسایلی برای حمل دوربین‌ها و تجهیزات تصویربرداری استفاده می‌شد. این موضوع به سال‌های ۲۰۰۰ تا ۲۰۱۰ میلادی بر می‌گردد.

امروزه یکی از تفریحات پرطرفدار کوادکوپترهای تفریحی و حرfe ای، تصویربرداری است که از جذابیت بسیار بالایی برخوردار بوده و بازار بزرگی از محصولات رادیو کنترلی را در جهان به خود اختصاص داده است. با پیشرفت وسایل پرنده و به وجود آمدن مالتی روتورها وظیفه تصویربرداری هوایی به عهده این دستگاه قرار گرفت. کوادکوپترها و هگزاکوپترها به دلیل داشتن تعادل پروازی بسیار بالا برای این کار مناسب هستند. با موجود شدن فلایت‌کنترلرها (کنترل کننده‌های پرواز) و تحقیقات صورت گرفته بر روی آن‌ها، پرواز با کوادکوپترها بسیار آسان شده و این وسایل از پرواز هوشمند بهره می‌برند. لذا تصویربرداران

می‌توانند با تمرکز بیشتر، تصاویر بهتری را تهیه کنند. همچنین با وجود GPS^{۱۲}ها پرواز این وسایل به صورت کاملاً آمن و هوشمند صورت می‌گیرد. در (شکل ۱-۶) یک کوادروتور در حال تصویربرداری از مناطق جغرافیایی مشاهده می‌گردد.



شکل ۱-۶ کوادروتور تصویربردار [4]

علاوه بر کاربردهای تصویربرداری؛ کوادکوپترها به تازگی یکی از لوازم سرگرمی و تفریحی به حساب می‌آیند. این تفریح که مخاطب‌های بزرگ‌سال را به خود جلب کرده است، روز به روز به گسترش است. و با پدید آمدن عینک‌های واقعیت مجازی و لینک شدن آنها با دوربین‌های کوادکوپترها، یک نوعی از فعالیت را به نام مسابقات کوادکوپترهای FPV^{۱۳} (نمایش اول شخص) ایجاد کرده است که به نظر در سال‌های آینده به یک رشته ورزشی پرطرفدار تبدیل خواهد شد.

۵-۱- چالش‌ها و پیشنهادهای موجود برای حل مساله

در سالهای گذشته تحقیقات فراوانی در خصوص مدل‌سازی و کنترل کوادروتور صورت گرفته است و مدل‌های مختلفی از ربات عمود پرواز چهارموتوره و چالش‌های پیش روی آن در مقاله‌های مختلف پیشنهاد

¹² Global Positioning System

¹³ First Person View

شده است. پس از معرفی مختصر هر یک از این چالش‌ها و پیشنهادات، در فصلی جداگانه به تفصیل به بررسی آن‌ها و روش انتخابی در این پروژه پرداخته خواهد شد.

۱-۵-۱- مدل‌سازی

یک روش معمول در مدل‌سازی دینامیک کوادراتور، استفاده از روش اویلر-لاگرانژ و مدل‌سازی براساس تغییرات انرژی سیستم است. روش دیگر، بررسی دینامیک جسم صلب (کوادراتور) در فضای سه بعدی با در نظر گرفتن نیروها و گشتاورهای وارد شده بر آن است. مدل به دست آمده در این روش‌ها، یک مدل غیرخطی بوده که با فرض خطی‌سازی حول نقطه‌ی کار، مدل ریاضی ساده‌تری محاسبه خواهد شد.

همچنین علاوه بر مدل‌سازی دینامیک کوادراتور، دینامیک محیط پرواز روبات نیز باید محاسبه گردد. این محیط، شامل تاثیرات گردش هوا در محیط اطراف پهپاد است. چرا که با چرخش پره‌ها و پرواز روبات، هوای اطراف آن به حرکت در آمده، به اجزای محیط همچون مولکول‌های هوا، اجسام موجود در اتاق و سطح زمین برخورد کرده، بازتاب این امواج هوا در محیط، باعث ایجاد اغتشاش در ورودی کنترل کننده می‌گردد که باستی برای بهبود کنترل و پرواز دقیق‌تر، حتماً مدل‌سازی گردد. لذا در فصل (۲) به طور کامل، روش انتخابی برای مدل‌سازی و معادلات آن شرح داده خواهد شد.

۱-۵-۲- موقعیت یابی

تخمین موقعیت افقی پرنده در محیط‌های داخل ساختمان کاری دشوارتر است زیرا حسگر مشخصی برای اینکار وجود ندارد. مرسوم‌ترین روش برای تخمین موقعیت در محیط مسقف، استفاده از دوربین و پردازش تصاویر آن است. بسته به اینکه دوربین بر روی خود روبات و یا محیط آزمایشگاه نصب شود اندازه‌گیری موقعیت به صورت نسبی یا مطلق انجام خواهد شد.

در کنار تخمین موقعیت پرنده، یکی از چالش‌های این پروژه، تشخیص و تخمین موقعیت فرود است که در محیط‌های خارج ساختمان با استفاده از سنسور GPS^{۱۴} می‌توان موقعیت حدودی روبات و محل

¹⁴ Global Positioning System

فروند را یافت اما در نهایت در هر دو محیط داخل و خارج از ساختمان، موقعیت یابی دقیق با استفاده از پردازش تصویر ممکن می‌شود. لذا پس از معرفی سنسورهای مورد استفاده در روبات به طور کامل در فصل (۳)، در فصل (۴) الگوریتم‌های پردازش تصویر و الگوریتم انتخابی توضیح داده خواهد شد.

۳-۵-۱- روش‌های کنترل پرواز

به منظور کنترل وضعیت و حفظ پایداری روبات‌های پرنده، به خصوص کوادرورتورها الگوریتم‌های مختلفی ارائه شده که به صورت شبیه‌سازی و یا پیاده‌سازی واقعی بررسی شده اند. در سال ۲۰۰۷ محققان دانشگاه زوریخ به بررسی عملکرد کنترل کننده‌های مختلف روی کوادراتور واقعی پرداختند. در این پژوهش، کنترل کننده‌های مختلفی از جمله کنترل کننده PID، خطی مربعی^{۱۵} و لغزشی^{۱۶} مورد آزمایش قرار گرفتند.

در نتایج به دست آمده بیان شد که کنترل کننده PID هم در شبیه‌سازی و هم در پیاده‌سازی به خوبی قابلیت تنظیم زوایا و موقعیت سیستم را دارد. کنترل کننده LQR با وجود اینکه در شبیه‌سازی نسبت به PID عملکرد مناسب تری دارد، در پیاده‌سازی واقعی قابل استفاده نیست. علت این امر دشوار بودن تنظیم مناسب ماتریس بهره K در این روش کنترلی است. همچنین از میان کنترل کننده‌های غیرخطی، کنترل کننده مد لغزشی به دلیل ماهیت نوسانی حول نقطه تعادل، برای پیاده‌سازی روی سیستم واقعی چندان مناسب نخواهد بود.^[۵]

در سالهای بعدی نیز تحقیقات گسترده‌ای در طراحی کنترل کننده کوادرورتور انجام شده است. به عنوان نمونه عملکرد کنترل کننده خطی‌سازی فیدبک^{۱۷} در مقایسه با مد لغزشی تطبیقی در سال ۲۰۰۹ بررسی شده است. طبق نتایج بدست آمده در این مقاله کنترل کننده خطی‌سازی فیدبک به دلیل حساسیت به نویز و پارامترهای نامعین سیستم، برای استفاده در شرایط واقعی مناسب نیست در حالی که کنترل کننده مد لغزشی در برابر نویز و پارامترهای نامعین مقاوم‌تر است.

¹⁵ Linear QuadraticRegulator (LQR)

¹⁶ Sliding mode

¹⁷ Feedback

شبکه عصبی و سایر روش‌های هوشمند نیز به عنوان کنترل‌کننده سیستم کوادراتور پیشنهاد شده‌اند. در شبیه‌سازی‌های صورت گرفته عنوان شده‌است، این کنترل‌کننده‌ها نسبت به روش PID حساسیت کمتری به نویز و اغتشاش دارند اما به دلیل محدودیت‌های پردازشی، پیاده‌سازی این روش‌ها بر روی سیستم واقعی چندان ساده نخواهد بود. [6]

در آخرین مقالات در این زمینه، یک کنترل‌کننده ترکیبی مقاوم-تطبیقی برای کوادراتور با ۹ درجه آزادی ارائه شده‌است. سیستم در نظر گرفته شده دارای عدم قطعیت پارامتری است و نویز در محیط وجود دارد. سیستم فعال به دو زیر سیستم تقسیم می‌شود. یک کنترل‌کننده مد لغزشی دینامیک داخلی را کنترل می‌کند در حالیکه کنترل‌کننده تطبیقی زیر سیستم فعال را کنترل می‌کند.

تمام پارامترهای سیستم مانند جرم، لختی، گرد و غبار و ... به طور کامل ناشناخته در نظر گرفته شده اند و با گذشت زمان تغییر می‌کنند. کنترلر تطبیقی از دو سیگنال خطا استفاده می‌کند: خطای ردیابی و خطای پیش‌بینی. از طریق اضافه کردن یک کنترل‌کننده مقاوم، کنترلر تطبیقی در حضور نویز بهبود داده شده‌است.

برای کنترل خودکار کوادراتور، اطلاع از تمام حالت‌های سیستم ضروری است. حالت‌های سیستم کوادراتور شامل زوایای اویلر، مختصات موقعیت و مشتقات آنهاست. در سیستم واقعی، امکان اندازه‌گیری هر یک از حالت‌های سیستم به طور مستقیم وجود ندارد. به همین دلیل برای دسترسی به حالت‌های سیستم، نیاز به استفاده از حسگرهای مختلف و ترکیب اطلاعات آنهاست.

فیلتر کالمان^{۱۸} یکی از پرکاربردترین روش‌های تخمین حالت‌های سیستم از روی خروجی حسگرهای است. یکی از زمینه‌های پراستفاده فیلتر کالمان برای تخمین زوایای یک جسم از روی اطلاعات حسگر IMU^{۱۹} است. به وسیله این روش می‌توان با ترکیب اطلاعات حسگرهای شتاب سنج و ژیروسکوپ ارزان قیمت MEMS^{۲۰}، زاویه و سرعت زاویه‌ای جسم را تخمین زد.

¹⁸ Kalman

¹⁹ Inertial Measurement Unit

²⁰ Micro ElectroMechanical System

در سیستم کوادراتور به دلیل دینامیک سریع، فرکانس حلقه کنترلی نیز حتی الامکان بالا انتخاب می‌شود. از طرف دیگر اغلب ریزکنترل کننده‌هایی که در این سیستم‌ها استفاده می‌شوند توان پردازشی محدودی را دارند. درنتیجه سیستم پردازشی قادر نخواهد بود محاسبات مربوط به فیلتر کالمن را با نرخ بالایی پردازش کند و این امر سبب کاهش فرکانس حلقه کنترلی می‌شود. برای رفع این مشکل، می‌توان از روش دیگری به نام فیلتر مکملی^{۲۱} استفاده کرد. مزیت اصلی روش فیلتر مکملی، محاسبات ساده‌تر آن نسبت به فیلتر کالمن است در نتیجه امکان تخمین زوایای اویلر با نرخ بالاتری وجود خواهد داشت. به همین دلیل اغلب از روش فیلتر مکملی برای تخمین زوایای کوادراتور استفاده می‌شود هرچند که روش فیلتر کالمن روشنی جامع‌تر است و برای تخمین یا ترکیب انواع داده‌ها می‌توان از آن استفاده کرد.[7]

²¹ Complementary Filter

فصل دوم

مدل‌سازی دینامیکی

مدل‌سازی دینامیکی [8] [7]

در این فصل به توسعهٔ مدل دینامیکی یک کوادرورتور و محیط اطراف آن پرداخته می‌شود. شناخت دینامیک یک سیستم برای طراحی کنترل‌کننده، ضروری است؛ علاوه بر این پس از شناخت دینامیک، می‌توان در جهت بهبود طرح اولیه سیستم نیز تلاش کرد.

۱-۲-۱ مدل کوادکوپتر

کوادرورتور با کنترل سرعت زاویه‌ای چهار موتور الکتریکی، همانطور که در (شکل ۱-۲) مشاهده می‌شود، مورد کنترل قرار می‌گیرد. هر موتور، نیروی رانش عمودی و گشتاوری ایجاد می‌کند که از ترکیب نیروهای این چهار موتور، نیروهای رانش عمودی اصلی، گشتاور یا^۱، پیچ^۲ و رول^۳ موثر بر کوادرورتور به وجود می‌آیند. نیروی پیش رانش حاصل از هر موتور (F_i) برابر است با ضریبی از محدود سرعت زاویه‌ای آن موتور که ضریب با (k_F) و سرعت زاویه‌ای موتور شماره i با (ω_i) نمایش داده شده‌است.

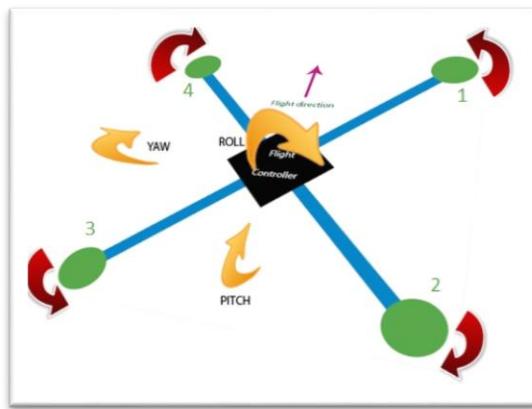
$$F_i = k_F \omega_i^2$$

در (شکل ۱-۲) جهت چرخش موتورها و جهت مثبت نیروی F_i بیان شده‌است. موتورهای ۱ و ۳ پاد ساعتگرد و موتورهای ۲ و ۴ ساعتگرد می‌چرخند. با این ترکیب و با فرض یکسان بودن فاصله موتورها از مرکز روبات، تاثیرات ژیروسکوپی و گشتاور آبرودینامیکی از بین می‌رود.

¹ Yaw (چرخش)

² Pitch (پیچش)

³ Roll (گردش)



شکل ۱-۲ ورودی‌های کنترلی کوادروتور [7]

نیروی رانش عمودی اصلی، حاصل جمع نیروهای تک تک موتورهاست. $(F_1 + F_2 + F_3 + F_4)$
 $- M_1 - M_2 + M_3 + M_4$ گشتاور پیچ، متناسب است با $M_1 - M_2 - M_3 + M_4$ گشتاور رول متناسب است با $M_1 - M_2 + M_3 - M_4$ گشتاور یاو متناسب است با $M_1 - M_2 + M_3 - M_4$ گشتاور عکس العمل موتور i ام است. این گشتاور، باتوجه به شتاب محور موتور و درگ^۱ ملخها (M_{drag}) به وجود می‌آید. لذا گشتاور موتور را می‌توانیم به صورت زیر تعریف نمائیم.

$$I_{rot} \dot{\omega} = M_i - M_{drag}$$

که در آن I_{rot} ممان اینرسی روتور حول محور خودش تعریف می‌شود و نیروی بازدارنده آرودینامیکی را به صورت زیر تعریف می‌نمائیم.

$$M_{drag} = \frac{1}{2} \rho A v_r^2$$

که در آن ρ چگالی هوا، A مساحت سطحی از هوا که ملخ در آن می‌چرخد و v_r سرعت چرخش با توجه به هواست. می‌دانیم که سرعت زوایهای ω برابر است با سرعت خطی v_r تقسیم بر شعاع چرخش b .

$$\omega = \frac{v_r}{b}$$

^۱ Drag

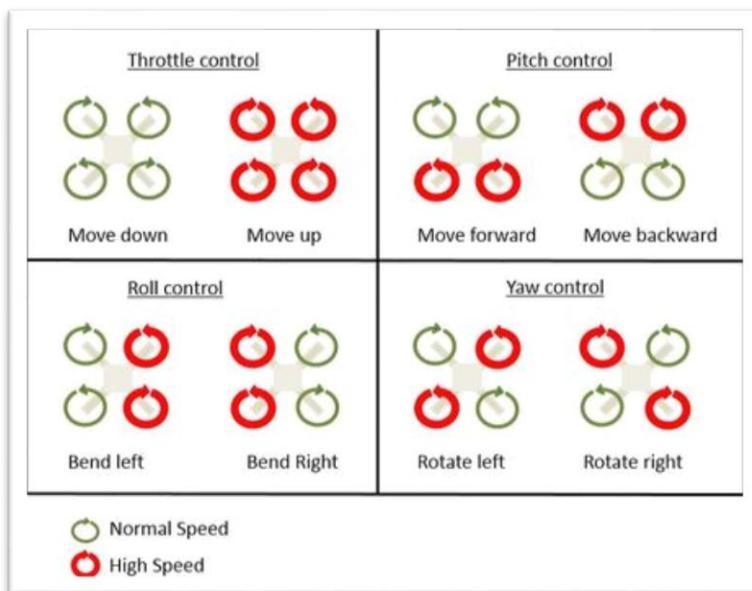
لذا می‌توانیم نیروی بازدارنده آیرودینامیکی را به صورت زیر بازنویسی کنیم.

$$M_{drag} = k_M \omega^2$$

که در آن k_M ثابتی مثبت است و مقدار آن به چگالی هوا، شعاع و شکل ملحه‌ها و دیگر عوامل بستگی دارد. برای حرکت ثابت معلق در فضا، ω ثابت است. لذا :

$$M_i = M_{drag} = k_M \omega^2$$

حرکت پیج رو به جلو با افزایش سرعت موتورهای ۲ و ۳ در حالی که سرعت موتورهای ۱ و ۴ رو به کاهش است به وجود می‌آید. به طور مشابه حرکت رول نیز با افزایش سرعت موتورهای ۴ و ۳ و کاهش سرعت موتورهای ۱ و ۲ بوجود می‌آید. حرکت یا و نیز بوسیله افزایش گشتاور موتورهای یک و سه (۱، ۳) و کاهش گشتاور موتورهای دو و چهار (۲، ۴) بوجود می‌آید در حالی که نیروی رانش عمودی کل ثابت می‌ماند. (شکل ۲-۲)



شکل ۲-۲ گشتاورهای یا و پیج و رول [7]

برای مدل کردن دینامیک یک کوادروتور از مدل مبتنی بر معادله لاغرانژ استفاده شده است [5] که در فصل (۲) به نحوه بدست آوردن آن می‌پردازیم.

۲-۲- ماتریس‌های دوران

با در نظر گرفتن زوایای دوران Tait-Bryan به عنوان پایه‌ی بیان زوایای مدل کوادراتور خواهیم

داشت:

$$x \text{ چرخش حول محور } R(x, \phi)$$

$$y \text{ چرخش حول محور } R(y, \theta)$$

$$z \text{ چرخش حول محور } R(z, \psi)$$

که در ادامه‌ی این فصل، به هر کدام این توابع به صورت زیر تعریف می‌شوند (c و s توابع سینوسی و کسینوسی هستند که به اختصار بدین شکل تعریف شده‌اند.):

$$R(x, \phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c(\phi) & -s(\phi) \\ 0 & s(\phi) & c(\phi) \end{bmatrix}$$

$$R(y, \theta) = \begin{bmatrix} c(\theta) & 0 & s(\theta) \\ 0 & 1 & 0 \\ -s(\theta) & 0 & c(\theta) \end{bmatrix}$$

$$R(z, \psi) = \begin{bmatrix} c(\psi) & -s(\psi) & 0 \\ s(\psi) & c(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

که چرخش کامل حول سه محور مختصات به صورت ضرب ماتریس‌های قبلی به ترتیب زیر در هم‌دیگر به دست می‌آید:

$$R(\phi, \theta, \psi) = R(x, \phi)R(y, \theta)R(z, \psi)$$

که نتیجه‌ی ضرب آنها در یکدیگر به صورت زیر است:

$$R(\phi, \theta, \psi) = \begin{bmatrix} c(\psi)c(\theta) & c(\psi)s(\theta)s(\phi) - s(\psi)c(\phi) & c(\psi)s(\theta)c(\phi) + s(\psi)s(\phi) \\ s(\psi) & s(\psi)s(\theta)s(\phi) + c(\psi)c(\phi) & s(\psi)s(\theta)c(\phi) - s(\theta)c(\psi) \\ -s(\theta) & c(\theta)s(\phi) & c(\theta)c(\phi) \end{bmatrix}$$

۲-۳- مدل دینامیکی چرخش سیستم

در این بخش مختصراً محاسبه‌ی مدل دینامیکی چرخش سیستم را می‌آوریم. مدل دینامیکی چرخش سیستم بر اساس معادله اویلر لاغرانژ تحت فرضیات زیر بدست می‌آید:

- سازه به صورت سخت^۱ در نظر گرفته شده.
- سازه به صورت متقارن در نظر گرفته شده.
- مرکز جرم^۲ و مرکز دستگاه بدنی بر روی هم در نظر گرفته شده‌اند.
- پروانه‌ها به صورت سخت در نظر گرفته شده‌اند.
- نیروی پیش رانش و اصطکاک متناسب با مربع سرعت پروانه‌ها می‌باشند.

مکانیک لاغرانژی فورمول بندی و نمایش دوباره ایست از مکانیک کلاسیک که بر اساس کمینه‌سازی یک استوار است (اصل کمترین کنش). بنا به تعریف، لاغرانژین (L) تفاضل انرژی جنبشی (T) و انرژی پتانسیل (V) است. یعنی داریم:

$$L = T - V$$

در اینجا، تکامل سیستم از حالتی به حالت دیگر به نحوی صورت می‌گیرد که انتگرال لاغرانژین کمینه شود. مثلاً در ساده‌ترین حالت، کُنش مکان یک ذره با جرم m در مکانیک کلاسیک (S) با توجیهی لاغرانژی به صورت زیر نوشته می‌شود:

$$S = \int_0^T \left(\frac{1}{2} m \dot{x}^2 - V(x) \right) dt$$

که اینجا $x(t)$ خود تابعی از زمان است. کمینه‌کردن کمیت S منجر به معادلاتی می‌شود که اصطلاحاً به آن‌ها معادلات اولر-لاغرانژ می‌گویند:

$$\frac{\partial L}{\partial x} - \frac{d}{dt} \frac{\partial L}{\partial \dot{x}} = 0$$

¹Rigid body

²CoG (Center of Gravity)

که در این معادلات، x_i مختصات تعمیم یافته و Γ_i نیروی تعمیم یافته را نمایش می‌دهند. که می‌شود:

$$m\ddot{x} = -\frac{\partial V}{\partial x} = F$$

که همان قانون دوم نیوتن است. [9]

در اینجا دستگاه مختصات زمین E به صورت یک دستگاه مختصات متعامد $[\vec{X} \quad \vec{Y} \quad \vec{Z}]$ و دستگاه مختصات بدنه B نیز به صورت یک دستگاه مختصات متعامد $[\vec{x} \quad \vec{y} \quad \vec{z}]$ در نظر گرفته می‌شوند. اگر برای هر نقطه در دستگاه مختصات B دوران‌های پشت سر هم را اثر بدهیم خواهیم داشت:

$$r_{(X,Y,Z)}(x, y, z) = R(\phi, \theta, \psi) \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

که $R(\phi, \theta, \psi)$ دوران حول رول، پیچ و یاو و بردار مختصات دکارتی است. لذا با انجام ضرب فوق، خواهیم داشت :

$$\begin{aligned} r_X &= (c(\psi)c(\theta))x \\ &+ (c(\psi)s(\theta)s(\phi) - s(\psi)c(\phi))y \\ &+ (c(\psi)s(\theta)c(\phi) + s(\psi)s(\phi))z \end{aligned}$$

$$\begin{aligned} r_Y &= (s(\psi)c(\theta))x \\ &+ (s(\psi)s(\theta)s(\phi) + c(\psi)c(\phi))y \\ &+ (s(\psi)s(\theta)c(\phi) - c(\psi)s(\phi))z \end{aligned}$$

$$\begin{aligned} r_Z &= (-s(\theta))x \\ &+ (c(\theta)s(\phi))y \\ &+ (c(\theta)c(\phi))z \end{aligned}$$

با مشتق گرفتن از روابط بالا نسبت به زمان سرعت (v_i) برای هر محور را می‌توان به دست آورد.

$$\nu^2 = \nu_X^2 + \nu_Y^2 + \nu_Z^2$$

با توجه به رابطه‌ی سرعت و با فرض این که ماتریس اینرسی (لختی) در هر محور (I_{ii}) به فرم قطری است، می‌توان معادله‌ی انرژی جنبشی را بر حسب ماتریس اینرسی به فرم زیر به دست آورد:

$$\begin{aligned} T &= \frac{1}{2} I_{xx} (\dot{\phi} - s(\theta) \dot{\psi})^2 \\ &+ \frac{1}{2} I_{yy} (c(\phi) \dot{\theta} + s(\phi) c(\theta) \dot{\psi})^2 \\ &+ \frac{1}{2} I_{zz} (s(\phi) \dot{\theta} - c(\phi) \dot{\psi})^2 \end{aligned}$$

همچنین با توجه به فرم کلی انرژی پتانسیل، با در نظر گرفتن این که g شتاب گرانش زمین است، در دستگاه مختصات زمین می‌توان نوشت:

$$\begin{aligned} V &= \int x dm(x) \cdot (-g s(\theta)) \\ &+ \int y dm(y) \cdot (g s(\phi) c(\theta)) \\ &+ \int z dm(z) \cdot (g c(\phi) c(\theta)) \end{aligned}$$

و در نهایت با استفاده از معادلات لاغرانژ با در نظر گرفتن تاثیر ژیروسکوپی (J_r) می‌توان معادلات چرخش را به صورت زیر به دست آورد:

$$\begin{aligned} \ddot{\phi} &= \frac{J_r \dot{\theta} \omega_t}{I_{xx}} + \frac{I_{yy} - I_{zz}}{I_{xx}} \dot{\psi} \dot{\theta} + \frac{\tau_{roll}}{I_{xx}} \\ \ddot{\theta} &= \frac{J_r \dot{\phi} \omega_t}{I_{yy}} + \frac{I_{zz} - I_{xx}}{I_{yy}} \dot{\psi} \dot{\phi} + \frac{\tau_{pitch}}{I_{yy}} \\ \ddot{\psi} &= \frac{I_{xx} - I_{yy}}{I_{zz}} \dot{\phi} \dot{\theta} + \frac{\tau_{yaw}}{I_{zz}} \end{aligned}$$

که در آن گشتاورهای رول، پیچ و یاو برابر است با :

$$\tau_{roll} = k_F \frac{l\sqrt{2}}{2} (-\omega_1^2 + \omega_2^2 + \omega_3^2 - \omega_4^2)$$

$$\tau_{pitch} = k_F \frac{l\sqrt{2}}{2} (\omega_1^2 - \omega_2^2 + \omega_3^2 + \omega_4^2)$$

$$\tau_{yaw} = k_M (-\omega_1^2 + \omega_2^2 - \omega_3^2 + \omega_4^2)$$

$$\omega_t = (\omega_1 - \omega_2 + \omega_3 - \omega_4)$$

۴-۲- مدل دینامیکی جابه جایی سیستم

در این بخش مدل دینامیکی جابجایی سیستم آورده می‌شود. نیروهای وارد بر مرکز جرم جسم عبارتند از نیروی جاذبه‌ی زمین و مجموع نیروی حاصل از چهار موتور. پس معادله‌ی حرکت سیستم را می‌توان به صورت زیر نوشت:

$$m\ddot{r} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + R(\phi, \theta, \psi) \begin{bmatrix} 0 \\ 0 \\ \tau_{thrust} \end{bmatrix}$$

که در آن:

$$\tau_{thrust} = k_F (\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2)$$

با جایگذاری، معادلات سیستم به فرم زیر به دست می‌آیند:

$$\ddot{x} = (c(\psi) s(\theta) c(\phi) + s(\psi) s(\phi)) \cdot \frac{\tau_{thrust}}{m}$$

$$\ddot{y} = (s(\psi) s(\theta) c(\phi) - s(\theta) c(\psi)) \cdot \frac{\tau_{thrust}}{m}$$

$$\ddot{z} = (c(\theta) c(\phi)) \cdot \frac{\tau_{thrust}}{m} - g$$

۵-۲- مدل نهایی

در نهایت مدل کلی یک کوادراتور به فرم زیر خواهد بود:

$$\begin{aligned}\ddot{x} &= (c(\psi) s(\theta) c(\phi) + s(\psi) s(\phi)) \cdot \frac{\tau_{thrust}}{m} \\ \ddot{y} &= (s(\psi) s(\theta) c(\phi) - s(\theta) c(\psi)) \cdot \frac{\tau_{thrust}}{m} \\ \ddot{z} &= (c(\theta) c(\phi)) \cdot \frac{\tau_{thrust}}{m} - g \\ \ddot{\phi} &= \frac{J_r \dot{\theta} \omega_t}{I_{xx}} + \frac{I_{yy} - I_{zz}}{I_{xx}} \psi \dot{\theta} + \frac{\tau_{roll}}{I_{xx}} \\ \ddot{\theta} &= \frac{J_r \dot{\phi} \omega_t}{I_{yy}} + \frac{I_{zz} - I_{xx}}{I_{yy}} \psi \dot{\phi} + \frac{\tau_{pitch}}{I_{yy}} \\ \ddot{\psi} &= \frac{I_{xx} - I_{yy}}{I_{zz}} \dot{\phi} \dot{\theta} + \frac{\tau_{yaw}}{I_{zz}}\end{aligned}$$

که در آن با در نظر گرفتن بازوی روبات (l) خواهیم داشت:

$$\begin{bmatrix} \tau_{thrust} \\ \tau_{roll} \\ \tau_{pitch} \\ \tau_{yaw} \end{bmatrix} = \begin{bmatrix} k_F & k_F & k_F & k_F \\ -k_F \frac{l\sqrt{2}}{2} & k_F \frac{l\sqrt{2}}{2} & k_F \frac{l\sqrt{2}}{2} & -k_F \frac{l\sqrt{2}}{2} \\ -k_F \frac{l\sqrt{2}}{2} & -k_F \frac{l\sqrt{2}}{2} & k_F \frac{l\sqrt{2}}{2} & k_F \frac{l\sqrt{2}}{2} \\ -k_M & k_M & -k_M & k_M \end{bmatrix} \cdot \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix}$$

با توجه به روابط فوق، کنترلر پس از محاسبه‌ی گشتاورها و نیروی مورد نیاز، با معکوس گرفتن از ماتریس بالا و حل معادله‌ی خطی فوق، سرعت زاویه‌ای مورد نیاز هر یک از موتورها را به دست می‌آورد.

۶-۲- خطی‌سازی و معادلات حالت

در مرحله‌ی بعد، لازم است تا معادلات به دست آمده در مدل نهایی، خطی‌سازی شده، ورودی‌ها و خروجی مشخص گردند و ماتریس‌های حالت محاسبه شوند.

برای این منظور، از نرم افزار متلب کمک گرفته شده تا خطی‌سازی معادلات را انجام داده و با توجه به ۴ ورودی به عنوان نیروی تراست و گشتاور‌های حول سه محور، همچنین ۴ خروجی به عنوان مقادیر رول، پیچ، یاو و ارتفاع روبات، همچنین با در نظر گرفتن ۱۲ حالت کنترلی که شامل مختصات x, y, z و مشتقات آن‌ها، زوایای چرخش رول، پیچ و یاو و مشتقات آن‌ها می‌شود.

$$\begin{aligned} states &= [x \quad y \quad z \quad \varphi \quad \theta \quad \psi \quad x' \quad y' \quad z' \quad \varphi' \quad \theta' \quad \psi']^T \\ u &= [\tau_{thrust} \quad \tau_{roll} \quad \tau_{pitch} \quad \tau_{yaw}]^T = [u_1 \quad u_2 \quad u_3 \quad u_4]^T \\ y &= [\varphi \quad \theta \quad \psi \quad z] \end{aligned}$$

برای خطی‌سازی معادلات، زوایای چرخش به جز زاویه‌ی یاو، در حالت تعادل و همچنین تمامی مشتقات برابر صفر قرار داده شده‌اند. همچنین u_i ‌ها، ورودی سیستم در نقطه‌ی تعادل هستند، بنابراین میزان نیروی تراست را برابر حاصلضرب جرم در شتاب گرانش زمین در نظر گرفته و گشتاور حول محور‌ها در نقطه‌ی تعادل نیز صفر خواهند بود.

$$states = [x_0 \quad y_0 \quad z_0 \quad 0 \quad 0 \quad \psi_0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]^T$$

با توجه به روابط فوق، مشاهده می‌شوند که در حالت تعادل، بردار مختصات مکانی و زاویه‌ی یاو می‌توانند هر مقداری را اتخاذ نمایند. در نهایت، ماتریس‌های حالت سیستم به صورت زیر محاسبه می‌گردند.

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sin(\psi)g & \cos(\psi)g & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\cos(\psi)g & \sin(\psi)g & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{1}{m} & 0 & 0 & 0 \\ 0 & \frac{1}{I_{xx}} & 0 & 0 \\ 0 & 0 & \frac{1}{I_{yy}} & 0 \\ 0 & 0 & 0 & \frac{1}{I_{zz}} \end{bmatrix}$$

$$C = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad D = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

که معادلات حالت خطی‌سازی شده به صورت زیر می‌شوند.

$$\begin{cases} f(states, u) = A \times states + B \times u \\ y(states, u) = C \times states + D \times u \end{cases}$$

۷-۲- تابع تبدیل سیستم

با استفاده از معادلات حالت خطی‌سازی شده، تابع تبدیل با استفاده از رابطه‌ی زیر محاسبه می‌شود.

$$G = C(SI - A^{-1})B + D$$

لذا خواهیم داشت:

$$G = \begin{bmatrix} 0 & \frac{1}{s^2 I_{xx}} & 0 & 0 \\ 0 & 0 & \frac{1}{s^2 I_{yy}} & 0 \\ 0 & 0 & 0 & \frac{1}{s^2 I_{zz}} \\ \frac{1}{s^2 m} & 0 & 0 & 0 \end{bmatrix}$$

۲-۸- پارامتر های روبات

در بخشهای قبلی معادلات حالت به ازای پارامتر های دینامیکی روبات مشخص شد که این پارامتر ها به صورت مجهول در رابطه قرار گرفت. در هنگام شبیه سازی، بایستی پارامتر ها تا حد امکان به روبات مورد آزمایش نزدیک باشند. (جدول ۱-۲) مقدار پارامتر های روبات مورد آزمایش را نمایش می دهد.

جدول ۱-۲ پارامتر های اندازه گیری شده مدل واقعی

بازوی روبات (متر)	Km	Kf	Jr (kg. m ²)	Izz (kg. m ²)	Iyy (kg. m ²)	Ixx (kg. m ²)	جرم (کیلو گرم)
0.44	7.5×10^{-7}	3.13×10^{-5}	3.74×10^{-5}	0.0172	0.0086	0.0086	0.47

بازوی روبات برابر است با فاصله انتهای پره های آن از مرکز جرم. Jr نیز تاثیر ژیروسکوپی ملخ ها حول محور چرخششان است. همچنین در گذشته توضیح داده شد که III برابر است با ماتریس لختی در هر محور که این مقادیر، با استفاده از نرم افزار مکانیکی SolidWorks، با شبیه سازی مکانیک روبات، قابل محاسبه است. Kr ضریبی است که مربع دور موتور را به گشتاور چرخشی وارد بر بدن می کند که می تواند به همراه Kf یا همان ضریب نیروی موتور ها، با استفاده از روش های تجربی محاسبه می گردد.

برای تشخیص کنترل پذیری و محاسبهٔ تابع تبدیل، لازم است تا این مقادیر به صورت عددی محاسبه شده و در روابط فوق جاگذاری شوند. بدین ترتیب ماتریس‌های A و B مشخص شده و می‌توان کنترل پذیری سیستم را بررسی نمود.

$$\text{rank}(A) = \text{rank}(C_B) = 12$$

بنابراین سیستم در تمام حالات کنترل پذیر است. همچنان تابع تبدیل حلقه باز به صورت زیر خواهد

بود.

$$G = \begin{bmatrix} 0 & \frac{116.2791}{s^2} & 0 & 0 \\ 0 & 0 & \frac{116.2791}{s^2} & 0 \\ 0 & 0 & 0 & \frac{58.1395}{s^2} \\ \frac{0.6803}{s^2} & 0 & 0 & 0 \end{bmatrix}$$

فصل سوم

حسگرها

حسگرها

برای کنترل خودکار یک سیستم نیاز به اندازه‌گیری و تخمین حالت‌های سیستم است. سیستم روبات پرنده کوادروتور مشابه سایر اجسام پرنده، دارای ۶ درجه آزادی است و حالت‌های این سیستم نیز متشکل از درجات آزادی و مشتقات آن هستند.

درجات آزادی هر جسم پرنده را می‌توان به دو دسته حرکت انتقالی و حرکت دورانی تقسیم نمود: مختصات قرار گیری جسم پرنده در چهارچوب زمین (x, y, z) و زاویه چهارچوب نصب شده روی جسم نسبت به چهارچوب زمین (زوایای اویلر). برای اندازه‌گیری حرکت انتقالی و حرکت دورانی یک جسم پرنده حسگرهای متفاوتی استفاده می‌شود.

در این فصل به معرفی انواع سنسورهای مورد نیاز یک روبات کوادکوپتر برای پرواز و به خصوص برای تشخیص محل فرود و عملیات فرود، پرداخته می‌شود.

۱-۳- سنجش زاویه

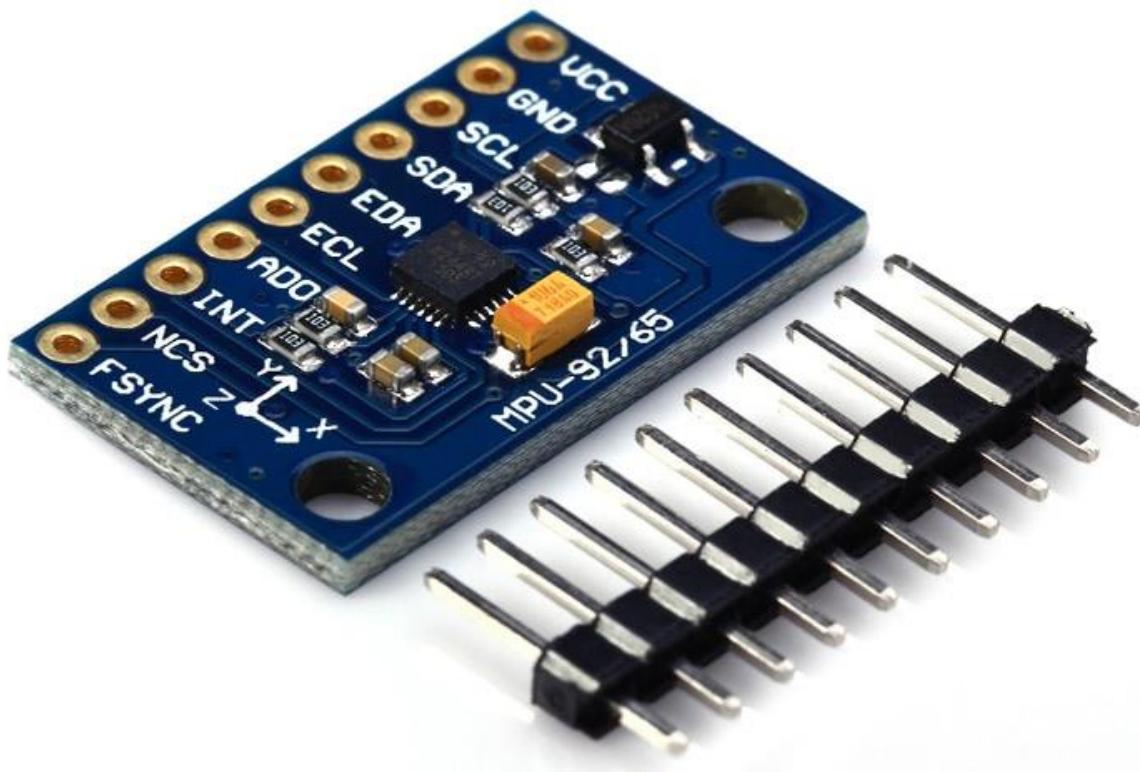
منظور از سنجش زاویه یا جهت‌گیری، بیان میزان دوران چهارچوب نصب شده بر روی جسم نسبت به چهارچوب مرجع (زمین) است. این میزان دوران توسط زوایای اویلر توصیف می‌شود (ماتریس دوران از روی زوایای اویلر ساخته می‌شود).

حسگری که برای سنجش زوایای اویلر مورد استفاده قرار می‌گیرد، واحد اندازه‌گیری اینرسی (IMU) نام دارد. حسگرهای IMU مبتنی بر تکنولوژی MEMS به دلیل ابعاد و وزن کم در وسایل پرنده کوچک، از جمله روبات پرنده کوادروتور، استفاده می‌شوند. این حسگر خود متشکل از ۳ نوع حسگر متفاوت است: حسگر ژیروسکپ، حسگر شتاب سنج و حسگر قطب نما (مغناطیس سنج).

حسگر ژیروسکپ میزان سرعت زاویه‌ای (دوران) حول محور مشخصی (محور چهارچوب جسم) را اندازه می‌گیرد، حسگر شتاب سنج میزان شتاب خطی را در یک محور مشخص می‌سنجد و نهایتاً حسگر قطب‌نما شدت میدان مغناطیسی را در یک محور مشخص تعیین می‌کند. برای اندازه‌گیری هر سه زاویه اویلر باید از سه حسگر ژیروسکپ متعامد (محورهای سنجش متعامد)، سه حسگر شتاب سنج متعامد و سه

حسگر قطب نمای متعامد استفاده نمود. در موقعی که این حسگرها به صورت جداگانه خریداری شوند، جایگذاری و متعامد سازی محور سنجش هر یک از این حسگرها کاری دشوار خواهد بود.

با رشد سریع تکنولوژی در زمینه MEMS تعداد حسگرهایی که در یک تراشه قرارداده می‌شود روز به روز افزایش پیدا می‌کند. به عنوان نمونه تراشه MPU9250 به تنها یک حاوی حسگر ژیروسکپ سه محوره، حسگر شتاب‌سنج سه محوره و حسگر قطب‌نمای سه محوره است و متعامد سازی محورهای سنجش این حسگرها در داخل تراشه و در هنگام ساخت انجام شده است (شکل ۳-۱).



شکل ۳-۱ مژول GY9250 [10]

برای کنترل یک روبات پرنده، یکی از اصلی‌ترین پارامترهای مورد نیاز، زاویه‌ی روبات نسبت به محورهای مختصات است. برای این منظور، در مدار کنترلی، مژولی تعییه شده که قادر به اندازه‌گیری سرعت و شتاب خطی در ۶ درجه آزادی (۳ درجه شتاب سنج و ۳ درجه ژیروسکوپ) است. همچنین می‌دانیم که با انجام عملیات انتگرال بر روی سرعت زاویه‌ای، عدد زاویه به دست می‌آید؛ اما این عدد با لرزش‌های کوچک روبات و موتورها به راحتی می‌تواند دچار خطا شود. لذا استفاده تنها از این واحد برای تشخیص موقعیتِ دقیق، عاقلانه نیست؛ اما برای تشخیص زاویه و جهت بسیار ضروری است.

۳-۲- [11] سامانه موقعیت یاب جهانی (GPS^۱)

سامانه‌ی موقعیت یابی جهانی یا جی‌پی‌اس مجموعه‌ای از ۲۴ ماهواره است که زمین را دور می‌زند و راکت‌های کوچکی نیز ماهواره‌ها را در مسیر صحیح نگاه می‌دارد. از این ماهواره‌ها، جهت شناسایی موقعیت جغرافیایی استفاده می‌شود. این ماهواره‌ها از محاسبات ریاضی ساده‌ای برای پخش اطلاعات استفاده می‌کنند که به عنوان طول و عرض و ارتفاع جغرافیایی، توسط گیرنده‌های زمین ترجمه شده‌اند. از راه حل‌های موجود برای یافتن موقعیت روبات، استفاده از سنسور گیرنده‌ی امواج فوق، بر روی روبات است. همچنین برای یافتن موقع حدودی محل فروود نیز، می‌توان یک سنسور گیرنده، در محل فروود نصب کرد. به علت حجم و قیمت مناسب این سنسورها و همچنین سادگی راه اندازی آن‌ها، استفاده از این سنسور در روبات و محل فروود بسیار توصیه می‌شود. (شکل ۳-۳) سنسور جی‌پی‌اس مورد استفاده در این پروژه که توسط شرکت Erle Robotics اسپانیا طراحی شده را نمایش می‌دهد.



شکل ۳-۳ سنسور GPS مورد استفاده در این پروژه [12]

^۱ GPS based

اما سوال مطرح این است که با وجود چنین سنسوری چرا همچنان نیاز به دوربین و پردازش تصویر برای یافتن محل فرود احساس می‌شود. برای پاسخ به این سوال کافی است تا میزان دقت سنسور GPS، مورد بررسی قرار گیرد. به طور متوسط، دقت این سنسورها در حدود ۳ تا ۵ متر خطأ دارد که برای این پروژه که با هدف فرود دقیق جلو می‌رود، خطای بسیار زیادی محسوب می‌شود. لذا نقطه ضعف اصلی این سنسور، عدم کارایی آن در فضای بسته است اما برای تکمیل پروژه، از آن جا که اهداف بعدی آن کنترل در فضای باز است، جایگاهی برای نصب این سنسور قرار داده خواهد شد. البته جی پی اس نیز در نهایت به علت وجود خطأ، به تنها یابی کارایی نداشته و بایستی با سایر سنسورها ترکیب شود.

۳-۳-۱- فاصله یاب‌ها [13]

از سنسورهای فاصله یاب برای تشخیص موقعیت نسبی روبات در مقایسه با اجسام اطراف آن استفاده می‌شود. بدین صورت که روبات امواجی را با فرستندهای از خود متصاعد کرده و امواج بازگشته را توسط یک سنسور گیرنده‌ی همان امواج اندازه‌گیری می‌نماید. به این ترتیب می‌توان فاصله روبات از جسم را اندازه گرفت. از این رو، موقعیت یابی دقیق توسط این سنسورها ممکن نیست و همچنین موقعیت یابی نسبی نیازمند نصب چند سنسور در چند جهت مختلف است تا فاصله در تمام محورهای سه بعدی قابل اندازه‌گیری باشد. به طور معمول در روبات‌های پرنده یک سنسور فاصله به سمت پایین برای تشخیص ارتفاع نصب می‌گردد. دو نوع سنسور به طور معمول برای تشخیص فاصله در روبات‌ها مورد استفاده قرار می‌گیرد:

- ۱- استفاده از نور
- ۲- سنسور فاصله یاب صوتی

۳-۳-۲- فاصله یاب نوری [14]

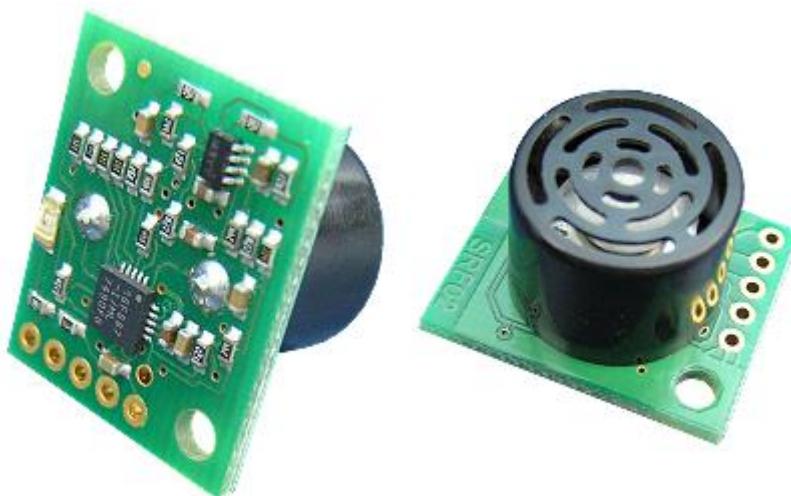
برای فاصله یابی معمولی، از نور در باند مادون قرمز استفاده می‌شود. در این روش، اغلب برای کاربردهای ساده از دیودهای منتشر کننده نور و در کاربردهای دقیق‌تر که نیاز به تمرکز نور و نقطه‌ای بودن هدف است، از لیزر به عنوان منبع نور استفاده می‌گردد.

استفاده از سنسور مادون قرمز از نظر راحتی راه اندازی مناسب‌تر است. با این وجود، مادون قرمز قابلیت تشخیص موائع شفاف را ندارد و از آن‌ها عبور می‌کند. همچنین استفاده از لیزر بسیار هزینه بر بوده و مقرنون به صرفه نیست.

۳-۲-۳- فاصله یاب صوتی [16]

در روش فاصله یابی صوتی، ابتدا یک موج صوتی با دامنه مناسب در محدوده ۱۵ کیلوهرتز به سطح مورد نظر تاییده شده و پس از بازگشت آن، با اندازه‌گیری مدت زمان رفت و برگشت موج صوتی، فاصله مورد نظر محاسبه می‌گردد.

این فاصله یاب‌ها قیمت مناسبی دارند. همچنین وزن کمی دارند و جریان کمتری هم مصرف می‌کنند. اما مشکلات زیادی دارند از جمله: رزولوشن بسیار کم، دقت زاویه‌ای بسیار کم در حدود ۳۰ درجه، سرعت بسیار پایین از این موارد هستند.(شکل ۳-۳)



[17] SRF02 مژول ۳-۳

۴-۳- پردازش تصویر

همانطور که پیش‌تر نیز توضیح داده شد، یک روش مفید برای پیدا کردن موقعیت نسبی روبات و محل فرود، استفاده از دوربین و پردازش تصویر می‌باشد. بدین ترتیب که یک دوربین بر روی روبات قرار گرفته و با پردازش تصویر اطراف روبات، به صورت نسبی محل روبات را تشخیص داده و یا با گذاشتن علامت مناسب، محل فرود را پیدا کرده به سمت آن حرکت کند. برای این منظور راه حل‌های متنوعی وجود دارد. برای مثال در انتخاب دوربین می‌توان از بین دوربین دو بعدی، دوربین سه بعدی و یا اسکنر لیزری گزینه‌ی مناسب تر را انتخاب نمود که در ادامه به انواع آن‌ها اشاره خواهد شد.

۱-۴-۳- دوربین دو بعدی

دوربین‌های دو بعدی انواع مختلف دیجیتال، آنالوگ و رنگی، یا سیاه و سفید، را دارا هستند که برای پردازش تصویر، دوربین‌های دیجیتال رنگی توصیه می‌شوند. با استفاده از این دوربین‌ها که به طور معمول با استفاده از پروتوكل یو اس بی^۱ به سیستم‌های مختلف متصل می‌شوند، می‌توان عملیات تشخیص رنگ و شکل تصویر را انجام داد اما متأسفانه قابلیت تشخیص عمق را در تصویر ندارند. البته برای تشخیص عمق توسط این دسته دوربین‌ها نیز الگوریتم‌هایی وجود دارد که در فصل (۴) معرفی خواهد شد. در بین این دوربین‌ها به علت تنوع بسیار بالا می‌توان موارد سبک، با سرعت بالا، قیمت کم و راه اندازی ساده را در انواع مختلف یافت.

۱-۴-۳-۱- USB دوربین

در ابتدا برای پردازش تصویر از یک دوربین USB ساخت شرکت Logitech و با کیفیت 720p استفاده شده است. (شکل ۴-۳) برای بررسی سرعت پردازش، ابتدا در هر دو محیط برنامه نویسی پایتون و C++ پردازش تصویر انجام شد. پس از مطالعه‌ی جزئیات مربوط به کتابخانه‌ی مذکور و نوشتن برنامه‌های متعدد و مشاهده‌ی نتیجه، دریافت شد که با افزایش فیلترها، سرعت پردازش به شدت افت کرده و از کارایی خارج می‌شود. با بررسی عوامل مختلف، می‌توان نتیجه گرفت که کند بودن پردازش به این علت است که اتصال دوربین با درگاه USB برقرار شده که موجب کند شدن سرعت انتقال داده از دوربین به مغز رزبری^۲ خواهد شد. بدین ترتیب با توجه به اینکه برای رزبری پای دوربین مخصوصی طراحی شده که مستقیماً به پردازش گر این وسیله متصل می‌گردد، بررسی این دوربین در اولویت قرار گرفت.

¹ USB

² Rasberry



شکل ۴-۳ یک مدل دوربین USB ساخت شرکت [18] Logitech

۴-۱-۳-۴- دوربین رزبری پای^۱

با توجه به سرعت پایین تصویربرداری با دوربین USB، دوربین رزبری پای مورد بررسی قرار می‌گیرد. در شروع، تصویر دوربین در نرم افزار پایتون به صورت خام مشاهده می‌گردد. به علت سرعت بالای پردازش و همچنین سادگی بیشتر محیط نرم افزار پایتون، ادامه بررسی‌ها بر روی همین نرم افزار انجام می‌شود. از آنجایی که این دوربین، منحصراً برای رزبری پای ساخته شده است، بدون استفاده از کتابخانه‌های جانبی، امکان اعمال برخی فیلترها بر روی آن وجود دارد. برای آشنایی، ابتدا تنظیماتی همچون وضوح، تضاد، نور، رنگ، زاویه و برش مورد بررسی قرار گرفتند. سپس تمام فیلترهایی که در این کتابخانه موجود است بر روی تصویر اعمال شده است.[19]

^۱ دربارهٔ رزبری پای درفصل ۷ به تفصیل توضیح داده شده است.



شکل ۳-۵ دوربین رزبری پای [20]

[21]-۲-۴-۳-دوربین کپنکت

دوربین کینکت^۱ که در (شکل ۶-۳) مشاهده می‌شود، با قیمت بسیار کم، در ابتدا برای بازی‌های کامپیوتروی استفاده می‌شد. اما به سرعت به دنیای روباتیک راه پیدا کرد زیرا می‌توانست تصویری ۳ بعدی با دقت سانتی متر، با بردی تا ۷ متر و زاویه دید ۴۷ درجه را با سرعت ۳۰ فریم در ثانیه به صورت همزمان خروجی دهد. اما استفاده از کینکت نیاز به سرعت پردازش بالا دارد، همچنین وزن و مصرف جریان بالای نیز دارد، این مشکلات آن باعث می‌شود که نتوان از آن در این سیستم به خوبی استفاده کرد.

¹ Kinect



شکل ۳-۶ دوربین کینکت [21]

۳-۴-۳- لیزر اسکنر

لیزر اسکنر در اصل یک رادار لیزری است که فاصله موانع اطراف را با دقت بسیار خوبی اندازه‌گیری می‌کند. این دقت، در حدود ۱ سانتی متر (با بردی تا ۸۰ متر) است. همچنین دقت زاویه‌ای تا ۵/۰ درجه (با میدان دید تا ۲۷۰ درجه) و سرعت بسیار بالا (تا ۲۵ اسکن در ثانیه) به سیستم کنترل، می‌تواند موقعیت یابی مناسبی را ایجاد کند. لیزر اسکنر نیازمند تامین جریان بالایی برای کار است، قیمت بسیار بالا و همچنین وزن زیادی دارد. در نتیجه استفاده از آن در این روبات به صرفه نبود.[22]



شکل ۳-۷ یک لیزر اسکنر کوچک با برد ۵ متر [23]

۴-۳-۴- نتیجه‌گیری: ترکیب سنسورها^۱

روشی که در این پژوهه توصیه می‌شود، همانطور که در تک بخش‌های پیشین توضیح داده شد، ترکیب چندین سنسور، برای تشخیص موقعیت است. بدین صورت که برای تشخیص زاویه و جهت، از یک واحد اندازه‌گیری ۹ محوره‌ی شتاب، زاویه و سرعت زاویه‌ای استفاده می‌شود. در کنار آن، برای تشخیص موقعیت در حدودی با خطای ۳ تا ۵ متر، یک سنسور مکان نمای GPS بر روی روبات و یکی بر روی محل فرود قرار داده می‌شود. همچنین یک دوربین دو بعدی رنگی رو به پایین برای تشخیص علامت فرود و یا علائم دیگر در حین پرواز استفاده می‌گردد. به علاوه با استفاده از دوربین و پردازش تصویر، می‌توان ارتفاع روبات را تشخیص داده و کنترل کرد. بدین ترتیب استفاده از سنسور فاصله برای ارتفاع ضرورتی نخواهد داشت. در در فصل بعد به معرفی روش‌ها و الگوریتم‌های تشخیص و پردازش تصویر پرداخته خواهد شد.

¹ Sensor fusion

فصل چهارم

پردازش تصویر

پردازش تصویر

برای تشخیص محل فرود، روش‌های متعددی وجود دارد که متقاضی ترین آن‌ها پردازش تصویر برای پیدا کردن محل مذکور است. یک روش، استفاده از امواج لیزر است. در این روش، یک سطح صاف و مناسب برای فرود، توسط یک دوربین، چند مازول امواج لیزر و پردازش آنها شناخته شده، سپس دستورات لازم برای فرود اعمال می‌گردد. از انجام این روش در این پروژه، به دلیل هزینه‌ی بالای مازول‌های لیزری و همچنین پیچیدگی پردازش آنها، صرف نظر می‌شود.

از دوربین می‌توان به عنوان سنسور موقعت یاب استفاده نمود. در مورد این روبات، طبق آزمایش‌ها و تست‌های انجام شده، استفاده از دوربین برای این منظور بهترین و به صرفه‌ترین روش ممکن است؛ زیرا دوربین، دارای قیمت و وزن کم است. همچنین با استفاده از آن، می‌توان با دقت خوبی موقعیت یابی کرد. به علاوه، جریان مصرفی دوربین، به نسبت مناسب است. در نتیجه از این روش برای انجام قسمت موقعیت یابی در این پروژه استفاده شد.

برای بحث موقعیت یابی از روش‌های مختلفی همچون اندازه‌گیری میزان جابه جایی تصویر^۱، تطبیق تصویر اطراف با عکس‌های از قبل ذخیره شده، الگوهای رنگی، الگوهای حاوی داده و ... استفاده کرد. از بین این روش‌ها مناسب‌ترین روش با توجه به وسعت حرکت روبات، لرزش، دید کم و محدودیت در حجم پردازش، بهترین راه استفاده از الگوها می‌باشد.

روشی که برای این منظور مورد استفاده قرار گرفته است بدین صورت است که یک الگوی ثابت در محیط قرار داده شده‌است با شناسایی این الگو در دوربین از آنجایی که ابعاد این الگواز پیش مشخص است، می‌توان موقعیت را به کمک اندازه دیده شده و تبدیل آن به فاصله نسبی و محل دیده شدن الگو در تصویر دوربین تخمین زد.

لذا مرحله‌ی بعد، انتخاب الگوی مناسب است که با توجه به شرایط نوری و نوبت محیط به راحتی قابل تشخیص توسط دوربین باشد.

¹ Optical Flow

۱-۴-۱- انواع الگو

گزینه‌های بسیاری برای یافتن الگوی تصویری، موجود است از جمله QR Code، علامت‌گذار نقطه‌ای، علامت‌گذار منحنی، الگوهای رنگی، ماتریس داده، الگوهای مورد استفاده در واقعیت افزوده که در ادامه به معرفی این گزینه‌ها می‌پردازیم.

۱-۴-۱- علامت‌گذار نقطه‌ای [24]

در این روش، نقاطی با استفاده از دیودهای نوری^۱ یا مادون سرخ^۲، در محل فرود قرار می‌گیرد تا روبات با تشخیص این نقاط با استفاده از گیرندهای نوری، محل فرود را تشخیص دهد.

۱-۴-۲- علامت‌گذار منحنی [24]

در این روش، از ترکیب منحنی‌های مختلف، اشکالی را می‌توان برای تشخیص محل فرود، طراحی نمود. برای نمونه در (شکل ۱-۴) شکل فرود به شکل H است. (تصویر معمول برای تشخیص محل فرود هلیکوپترها) همچنین استفاده از چندین دایره، مربع یا مثلث، برای ساخت تصویر، در (شکل ۲-۴) قابل مشاهده است.



شکل ۱-۴ محل فرود به شکل H [24]

¹ LED

² IR-Diodes



شکل ۲-۴ محل فرود به شکل سه مثلث تو در تو [24]

[25] QR Code - ۴-۱-۳

این گزینه، یک رمزینه ماتریسی یا بارکد (دو بعدی) است. می‌توان آن را با پوینده‌های QR، تلفن همراه دوربین دار و تلفن هوشمند خواند. این بارکد دربردارنده چیدمانی از چهارگوش‌های سیاه بر روی زمینه‌ی سفید است. داده نهفته می‌تواند نوشته، نشانی وب، شماره تلفن، یا داده دیگری باشد. (شکل ۳-۴) ایده‌ی ساخت و اجرای این رمز، در شرکت تویوتا ژاپن و در سال ۱۹۹۴ میلادی به دست آمد. استفاده از این کد در ژاپن بسیار فراگیر است و بیشترین کاربرد را میان بارکدهای دو بعدی دارد. این بارکد،^۱ (یا پاسخ سریع) نام گرفت چرا که طبق خواست سازنده‌ی آن، بایستی کد به گونه‌ای باشد که درونمایه با شتاب زیاد رمزگشایی شود.



شکل ۳-۴ نمونه یک الگوی QR Code

^۱ Quick Response

۴-۱-۴- الگوی رنگی

این الگوها از یک الگوی چند رنگی تشکیل شده اند که به وسیله دوربین در محیط، قابل تشخیص هستند. تصویر نمونه‌ای از آن قابل مشاهده است. شناسایی این الگوها بسیار سریع می‌باشد اما دو ایراد اساسی دارند:

۱- حساسیت بالا نسبت به نور دهی محیط و اشیا رنگی اطراف

۲- محدودیت در تعداد الگوهای قابل نصب در یک محیط

۴-۱-۵- ماتریس داده [26]

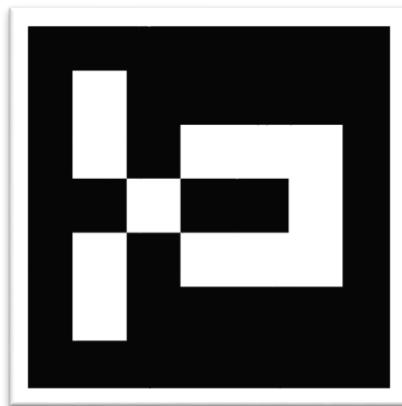
یک کد ماتریس داده، یک بارکد ماتریسی دو بعدی است که دارای سلول‌های سیاه و سفید، یا مازول هایی است که در یک الگوی مربعی یا مستطیلی مرتب شده‌اند. اطلاعات قابل کد شدن در این ماتریس می‌تواند اطلاعات خام یا متن باشد. اندازه داده‌های معمولی می‌تواند از چند بایت تا ۲ کیلوبایت تغییر کند. طول داده کد شده بستگی به اندازه نشان مورد استفاده دارد. کدهای تصحیح خطاب برای افزایش استقامت نشانه به آن اضافه می‌شوند. بدین معنی که در صورت آسیب دیدگی قسمتی از آن، باز هم قابل خواندن باشند. یک نشانه ماتریس داده می‌تواند تا ۲۳۳۵ کاراکتر الفبا را ذخیره کند. (شکل ۴-۴)



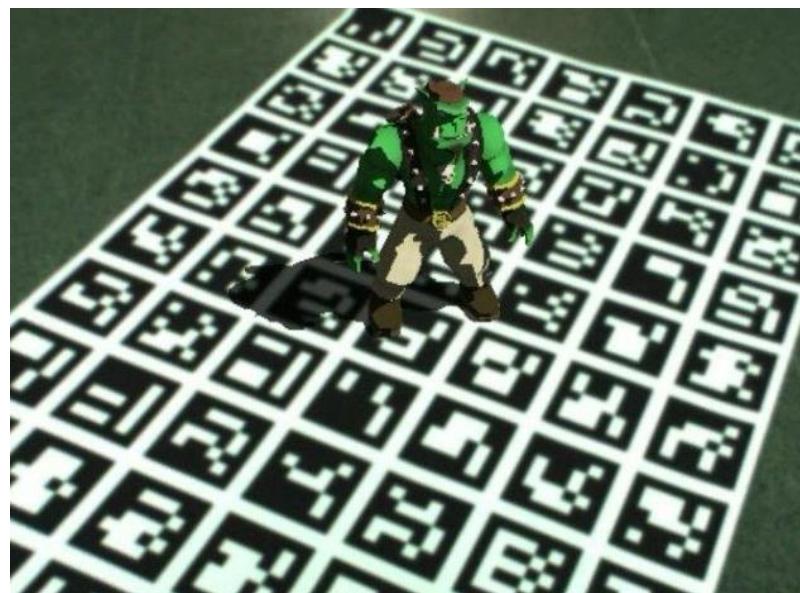
شکل ۴-۴ نمونه یک الگوی ماتریس داده [26]

۶-۱-۴- الگوهای واقعیت افزوده [27]

این الگوها که نوعی بارکد ۷ در ۷ هستند در پروژه‌های واقعیت افزوده استفاده می‌شوند. این الگوها، نمونه ساده سازی شده الگوهایی مانند ماتریس داده اند که با تغییراتی آنها را مناسب کارهای واقعیت افزوده کرده اند تا هم سریع‌تر پردازش شوند و هم از فاصله‌ی دورتر و با دوربین موبایل‌های معمولی و با حساسیت کم‌تر نسبت به نور و اشیا محیطی خوانده شوند. (شکل ۵-۴)



شکل ۴-۵ نمونه یک الگوی واقعیت افزوده [27]



شکل ۶-۴ استفاده از یک الگو برای اضافه کردن شخصیت مجازی به تصویر [27]

۴-۲- الگوریتم‌های پردازش تصویر

پس از انتخاب الگوی مناسب، نحوه تشخیص آن توسط پردازنده حائز اهمیت است. در ادامه به بررسی تعدادی از انواع روش‌ها و الگوریتم‌های مناسب برای پردازش تصویر پرداخته می‌شود.

۱- فیلترسازی و [28] Thresholding^۱

با توجه به اینکه تصویر دریافتی از دوربین، دارای نویز بوده و کیفیت پایینی دارد (برای افزایش سرعت محاسبات نمی‌توان تصویربرداری را با کیفیت بالا انجام داد)، از تصویرهای صفر و یک استفاده می‌شود تا هزینه‌ی محاسبه کاهش یافته و الگوریتم بهبود یابد. لذا ابتدا تصویر به محدوده رنگ خاکستری منتقل شده و سپس با تنظیم برخی پارامترهای تصویر (حفظ نور در حین حذف طیف رنگی و ...) با انجام معادله زیر، نتیجه‌ی دلخواه حاصل می‌گردد.

$$Y = 0.299 \times R + 0.596 \times G + 0.211 \times B$$

که در معادله‌ی فوق، R , G , B به ترتیب مشخص کننده‌ی رنگ آبی، سبز و قرمز در تصویر هستند. الگوریتم thresholding با تولید یک عکس دو دویی، نمایش دهنده‌ی هدف فرود است، با این تفاوت که اطلاعات غیر ضروری را از تصویر حذف می‌نماید. برای دریافت یک نتیجه‌ی مناسب، مقدار آستانه باید در یک محدوده‌ی مناسب انتخاب شود. (۸۰ درصد بین کمترین و بیشترین مقدار بازه خاکستری) با استفاده از یک فیلتر Median^۲ با ابعاد ۷ در ۷، می‌توان نویز باقی مانده را حذف نمود و جزئیات لبه‌های تصویر را بهتر مشاهده کرد.

^۱ جایگذاری پیکسل‌ها با پیکسل‌های سیاه و سفید بر اساس یک مقدار آستانه.

^۲ این فیلتر رفتار پایین‌گذر داشته و نویز اضافه شده‌ی سفیدرا حذف می‌کند و بسیار مناسب تشخیص اشکال هندسی مانند نشانه‌ی محل فرود است.

۲-۴-۲- بخش بندی و برچسب زدن محتويات وابسته [22]

تصویری که پس از عملیات فوق به جای می‌ماند، همچنان ممکن است شامل تصاویری اضافه بر تصویر محل فروود باشد. در این قدم، بخش‌های مختلف تصویر شناسایی شده و برچسب زده می‌شوند.

نحوه‌ی برچسب زدن به این گونه است که ابتدا تصویر، سطر به سطر اسکن شده، تا به یک پیکسل در مرز اشکال برخورد شود. تمام ۸ پیکسل در همسایگی آن نقطه، به عنوان یک شیء وابسته به آن نقطه علامت‌گذاری می‌گردد. این عملیات بدین صورت ادامه می‌یابد تا تمام پیکسل‌های وابسته به شیء مذکور، شمرده شوند. با انجام یک عملیات ضرب، می‌توان مساحت شیء را محاسبه نمود. از اشیائی که سطح آن‌ها کمتر از آستانه‌ی معمول باشد، صرف نظر می‌گردد. به طور مشابه از اشیاء با سطح بیشتر از آستانه نیز صرف نظر می‌شود. محدوده‌ی باقی مانده، محدوده‌ی مد نظر (ROI^۱) خواهد بود. آستانه‌ی مطلوب، با توجه به ارتفاع حدودی که توسط سنسور فاصله سنجش می‌شود، قابل تعیین است. برای مثال در ارتفاع بیشتر از ۲ متر، آستانه‌ی مد نظر بین ۵۰ تا ۱۰۰ پیکسل است.

۳-۴- نتیجه‌گیری

با توجه به تنوع بالای این الگوها، سرعت پردازش بالا و کتابخانه‌ی مخصوص آن‌ها، در این پژوهه مجموعه‌ای از این الگوهای مشابه (شکل ۶-۴) در کنار هم قرار می‌گیرند تا محل فروود را مشخص کنند. به این ترتیب با تشخیص تعداد الگوها در تصویر، می‌توان ارتفاع یا در واقع فاصله از زمین را تشخیص داد. سپس با استفاده از الگوریتم‌های مذکور، به همراه کتابخانه‌ی آماده‌ی برنامه نویسی که برای این الگو در دسترس قرار دارد، الگو را تشخیص داده و در مرحله‌ی بعد نیاز بایستی روبات با استفاده از تصویری که تشخیص داده، فروود بیاید. لذا در فصل بعد، به تشریح کنترل‌کننده‌های روبات برای حرکت و یافتن محل فروود پرداخته خواهد شد. همچنین در فصل پایانی پیاده‌سازی عملی پردازش تصویر توضیح داده خواهد شد.

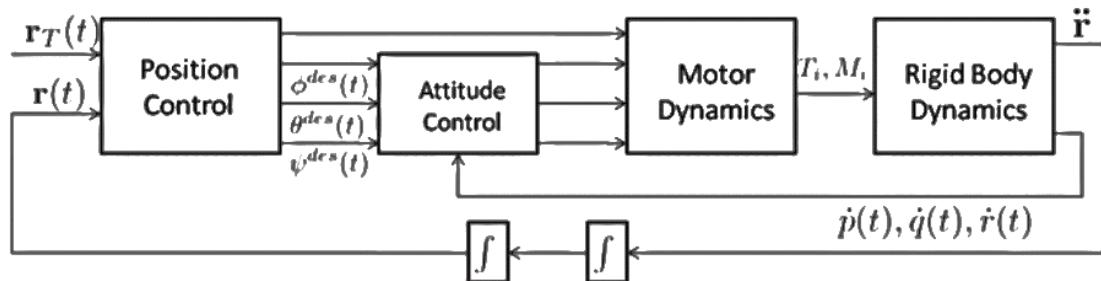
^۱ Region Of Interest

فصل پنجم

موقعیت یابی و فرود

موقعیت یابی و فرود

پس از دست یافتن به توانایی تشخیص محل فرود، چالش بعدی این است که بتوان این تصویر را در یک محیط یافته و پس از آن، در محل مربوطه فرود آمد. پیدا کردن مختصات مربوطه روش‌های بسیاری دارد؛ به خصوص که در محیط‌های باز، فرود آمدن دقیق و با هدف به علت وسعت زیاد، بدون اطلاع از مختصات محل فرود یا مختصات روبات، عملاً ممکن نیست. لذا با استفاده از سنسورهای موجود، موقعیت نسبی روبات با احتساب خطای موجود، محاسبه شده و با استفاده از داده‌های موجود، سناریوی حرکتی تعریف می‌شود تا روبات بتواند محل مورد نظر را تشخیص دهد. برای این منظور ابتدا باید حرکت روبات را کنترل کرد. از آن جا که کوادراتور یک سیستم زیر فعال است که توسط ۴ موتور، حرکت در ۶ درجه آزادی را کنترل می‌نماید، در ابتدا زوایا کنترل شده و با استفاده از داده‌های حاصل از کنترل زاویه، مختصات طول و عرض جغرافیایی نیز کنترل می‌شوند. برای کنترل ارتفاع نیز، کنترل کننده‌ی مجازی در نظر گرفته شده‌است که در ادامه توضیح داده می‌شود.



شکل ۱-۵ نمای کلی کنترل کننده‌ی موقعیت

همانطور که در (شکل ۱-۵) مشاهده می‌شود، موقعیت طول، عرض و ارتفاع مطلوب تحت عنوان بردار r_T به کنترل کننده‌ی موقعیت اعمال می‌شود. در این کنترل کننده، برای هر کدام از پارامترهای x , y و z یک کنترل PID طراحی شده‌است. سپس این مقادیر، تبدیل به زوایای مطلوب رول، پیچ و یاو می‌شوند.

در این مرحله، خطای هر یک از زوایای رول، پیچ و یا توسط کنترل کننده PID کنترل کاهش داده شده و مقادیر نهایی، به مدل موتور ارسال شده و از طریق آن به عنوان ورودی، به مدل روبات منتقل می‌گردد. در ادامه معادلات کنترل مربوط به هر دو قسمت کنترل موقعیت و کنترل زاویه محاسبه شده است.

۱-۵- کنترل موقعیت [8]

با خطی‌سازی معادلات حرکت روبات حول نقطه‌ی کاری ($\theta = 0, \phi = 0, \psi = \psi_0, u_1 = mg$) می‌توانیم بنویسیم:

$$\ddot{x} = g(\phi \sin \psi_0 + \phi \cos \psi_0)$$

$$\ddot{y} = g(\theta \sin \psi_0 - \theta \cos \psi_0)$$

$$m\ddot{z} = \Delta\tau_{thrust}$$

با حل معادلات فوق نسبت به θ ، ϕ و $\Delta\tau_{thrust}$ رابطه زوایای مورد نیاز برای رسیدن به شتاب مورد نظر در فضای سه بعدی و نیروی پیشرانش مورد نیاز به صورت زیر به دست می‌آید:

$$\phi_{des} = \frac{1}{g} (\ddot{x}_{des} \sin \psi_0 - \ddot{y}_{des} \cos \psi_0)$$

$$\theta_{des} = \frac{1}{g} (\ddot{x}_{des} \cos \psi_0 + \ddot{y}_{des} \sin \psi_0)$$

$$\Delta\tau_{thrust} = m \ddot{z}_{des}$$

برای به دست آوردن \ddot{r}_{des} برای تعقیب یک مسیر در حالت تک عامله از یک کنترل PID استفاده می‌شود:

$$\ddot{r}_{des} = \ddot{r}_T + K_{pT}(r_T - r) + K_{dT}(\dot{r}_T - \dot{r}) + K_{iT} \int (r_T - r) dt$$

که در آن $r_T = [x_T, y_T, z_T]^T$ بردار مسیر مورد نظر و K_{dT} ، K_{pT} و K_{iT} به ترتیب ماتریس قطری ضرایب P، D و I کنترلر می‌باشد. که در نتیجه معادله‌ی سیستم حلقه بسته به صورت زیر خواهد بود:

$$\ddot{\tilde{r}} + K_{dT}\dot{\tilde{r}} + K_{pT}\tilde{r} + K_{iT}\int \tilde{r}dt = 0$$

۵-۲- کنترل زاویه [8]

حلقه‌ی داخلی وظیفه‌ی کنترل زاویه و جهت گیری روبات در فضای کار را بر عهده دارد. این کنترلر با ایجاد نیروی مناسب در موتورها زاویه‌ی روبات را به زاویه‌ی مطلوب تعیین شده توسط کنترلر موقعیت می‌برد.

برای طراحی این کنترلر از فرض نزدیک بودن روبات به حالت‌هاور^۱ استفاده شده‌است. با توجه به معادلات زاویه‌ی سیستم:

$$\begin{aligned}\ddot{\phi} &= \frac{J_r\dot{\theta}\Omega_t}{I_{xx}} + \frac{I_{yy} - I_{zz}}{I_{xx}}\psi\dot{\theta} + \frac{\tau_{roll}}{I_{xx}} \\ \ddot{\theta} &= \frac{J_r\dot{\phi}\Omega_t}{I_{yy}} + \frac{I_{zz} - I_{xx}}{I_{yy}}\psi\dot{\phi} + \frac{\tau_{pitch}}{I_{yy}} \\ \ddot{\psi} &= \frac{I_{xx} - I_{yy}}{I_{zz}}\dot{\phi}\dot{\theta} + \frac{\tau_{yaw}}{I_{zz}}\end{aligned}$$

برای خطی‌سازی معادلات حول نقطه‌هاور قرار می‌دهیم:

$$\dot{\psi} \approx 0, \quad \dot{\phi} \approx 0, \quad \dot{\theta} \approx 0, \quad \Omega_t \approx 0$$

همچنین با توجه به تقارن ساخت روبات می‌توان گفت $I_{xx} \approx I_{yy}$

¹ Hover

در نتیجه مدل ساده شده‌ی زیر برای طراحی کنترلر مورد استفاده قرار می‌گیرد:

$$I_{xx}\ddot{\phi} = \tau_{roll}$$

$$I_{yy}\ddot{\theta} = \tau_{pitch}$$

$$I_{zz}\ddot{\psi} = \tau_{yaw}$$

کنترلر مورد استفاده برای کنترل زوایا یک کنترلر PD+FeedForward می‌باشد:

$$\tau_{roll} = K_{p\phi}(\phi_{des} - \phi) + K_{d\phi}(\dot{\phi}_{des} - \dot{\phi}) + K_{ff\phi}\ddot{\phi}_{des}$$

$$\tau_{pitch} = K_{p\theta}(\theta_{des} - \theta) + K_{d\theta}(\dot{\theta}_{des} - \dot{\theta}) + K_{ff\theta}\ddot{\theta}_{des}$$

$$\tau_{yaw} = K_{p\psi}(\psi_{des} - \psi) + K_{d\psi}(\dot{\psi}_{des} - \dot{\psi}) + K_{ff\psi}\ddot{\psi}_{des}$$

که در نهایت معادله‌ی زوایای روبات در حالت حلقه بسته، با ترکیب دو دسته معادله قبلی به صورت زیر خواهد بود:

$$K_{ff\phi}(\ddot{\phi}_{des} - \frac{I_{xx}}{K_{ff\phi}}\ddot{\phi}) + K_{d\phi}(\dot{\phi}_{des} - \dot{\phi}) + K_{p\phi}(\phi_{des} - \phi) = 0$$

$$K_{ff\theta}(\ddot{\theta}_{des} - \frac{I_{yy}}{K_{ff\theta}}\ddot{\theta}) + K_{d\theta}(\dot{\theta}_{des} - \dot{\theta}) + K_{p\theta}(\theta_{des} - \theta) = 0$$

$$K_{ff\psi}(\ddot{\psi}_{des} - \frac{I_{zz}}{K_{ff\psi}}\ddot{\psi}) + K_{d\psi}(\dot{\psi}_{des} - \dot{\psi}) + K_{p\psi}(\psi_{des} - \psi) = 0$$

با انتخاب ضرایب به صورت زیر:

$$K_{ff\phi} = I_{xx}, \quad K_{d\phi} = 2I_{xx}\zeta_\phi\omega_{n\phi}, \quad K_{p\phi} = I_{xx}\omega_{n\phi}^2$$

$$K_{ff\theta} = I_{yy}, \quad K_{d\theta} = 2I_{yy}\zeta_\theta\omega_{n\theta}, \quad K_{p\theta} = I_{yy}\omega_{n\theta}^2$$

$$K_{ff\psi} = I_{zz}, \quad K_{d\psi} = 2I_{zz}\zeta_\psi\omega_{n\psi}, \quad K_{p\psi} = I_{zz}\omega_{n\psi}^2$$

معادلات خطای تعقیب روبات به صورت زیر خواهد شد:

$$\ddot{\phi} + 2\zeta_\phi \omega_{n\phi} \dot{\phi} + \omega_{n\phi}^2 \tilde{\phi} = 0$$

$$\ddot{\theta} + 2\zeta_\theta \omega_{n\theta} \dot{\theta} + \omega_{n\theta}^2 \tilde{\theta} = 0$$

$$\ddot{\psi} + 2\zeta_\psi \omega_{n\psi} \dot{\psi} + \omega_{n\psi}^2 \tilde{\psi} = 0$$

که در آن $\phi = \phi_{des} - \tilde{\phi}$ خطای تعقیب رول و $\theta = \theta_{des} - \tilde{\theta}$ و $\psi = \psi_{des} - \tilde{\psi}$ به ترتیب خطای تعقیب پیج و یاو می‌باشند. در فصل (۶) به محاسبه‌ی ضرایب مذکور، پرداخته می‌شود.

۱-۲-۵- کنترل ارتفاع^۱

کنترل ارتفاع، یعنی حفظ فاصله‌ی روبات تا زمین در مقدار دلخواه. برای انجام این الگوریتم، همانطور که در قسمت حسگرها توضیح داده شد، استفاده از یک سنسور فاصله یا صوتی و یا دوربین توصیه می‌شود.
[5]

با توجه به قانون پایه‌ی پرسپکتیو^۲، با نزدیک شدن به یک جسم، آن جسم در چشم انسان و یا در تصویر دوربین، بزرگ و بزرگ‌تر مشاهده می‌شود. به طور مشابه، با کوچک شدن تصویر مشاهده شده، می‌توان تشخیص داد که فاصله از آن جسم زیاد شده‌است. با وجود درک ساده‌ی این رابطه، پیاده‌سازی عملی آن با استفاده از تصویر گرفته شده توسط دوربین، چنان ساده نیست. در واقع در تصویربرداری باید روابطی برای تشخیص بزرگ یا کوچک شدن تصویر برای پردازنده‌ی دیجیتال تعریف گردد.

برای تعریف روابط مذکور، از قوانین فیزیکی اپتیک استفاده می‌شود تا ارتفاع صحیح با توجه به ابعاد تصویر پیدا شود. با توجه به اینکه این محاسبات در واقعیت با چشم غیر مسلح قابل انجام است، می‌توان یک رابطه‌ی تجربی بین ابعاد تصویر در واحد پیکسل و ابعاد واقعی جسم در واحد سانتی متر در نظر گرفت. برای

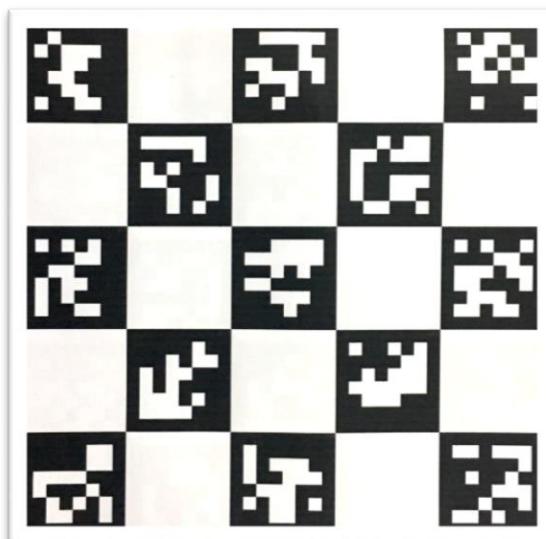
¹ Altitude control

² Perspective

به دست آوردن رابطه‌ی فوق، تصویر هدف در فواصل ۲۰ سانتی متری مشاهده شده و ابعاد واقعی هر بارکد به سانتی متر، فاصله از آن و ابعاد آن به پیکسل در جدولی تهیه می‌شوند. (جدول ۱-۵) همچنین به دلیل انتخاب بارکدهای سرعت بالا، با چیدمان فراکتالی آن‌ها، تعداد بارکدها نیز مطابق (شکل ۲-۵) برای تشخیص فاصله بسیار کمک کننده هستند. [29]

جدول ۱-۵ تعداد پیکسل نمایش هر بارکد در ارتفاع‌های متفاوت و تعداد بارکدهای مورد مشاهده

فاصله (cm)	۲۰	۴۰	۶۰	۸۰	۱۰۰	۱۲۰	۱۴۰	۱۶۰	۱۸۰	۲۰۰	۲۲۰	۲۴۰	۲۶۰
اندازه (pixel)	۹۶۰	۴۸۰	۳۲۰	۲۴۰	۱۹۲	۱۶۰	۱۳۶	۱۲۰	۱۰۶	۹۶	۸۶	۸۰	۷۴
تعداد بارکد	۱	۲-۱	۵-۳	۶-۴	-۸ ۱۰	-۱۲ ۱۵	۱۵	۱۵	۱۵	۱۵	۱۵	۱۵	۱۵



شکل ۲-۵ محل فرود با ابعاد ۱ در ۱ متر و تصویر ۹۶۰ در ۹۶۰ پیکسل

۲-۵- نتیجه‌گیری

پس از کنترل کامل رفتار روبات و تشخیص زمان مناسب برای فرود، نوبت به تشخیص محل فرود می‌رسد. با توجه به اینکه سنسور موقعیت تنها در فضای باز خروجی داشته و میزان خطای ۳ تا ۵ متر نیز دارند، برای یافتن محل فرود چه در فضای باز و چه در فضای بسته، توصیه می‌شود که روبات در ارتفاعی در حدود ۳-۲ متری قرار بگیرد. سپس محوطه‌ای در محدوده‌ای ابعاد ۵ در ۵ متر را قدم به قدم اسکن نموده، در صورت پیدا کردن علامت فرود، در ارتفاع ثابت متوقف شود و سپس به ادامه‌ی عملیات فرود بپردازد.

در ادامه‌ی عملیات فرود، بایستی دستور فرود به موتورها ارسال شود. با ارسال دستور فرود، مطابق با فلوچارت (شکل ۳-۵) ارتفاع مطلوب ارسالی به روبات اندکی کاهش داده می‌شود. پس از انجام کنترل رفتار مناسب برای قرار گرفتن دقیق بر روی علامت فرود و کنترل ارتفاع دلخواه، بررسی می‌شود که آیا فرود انجام گرفته یا نه. اگر فرود انجام گرفته باشد موتورها متوقف می‌شوند و در غیر این صورت، دوباره ارتفاع مطلوب کاهش داده می‌شود و مراحل پس از آن تکرار می‌گردند تا در نهایت فرود به صورت کامل انجام گیرد. [5]

پس از ارسال فرمان فرود، برای کنترل رفتار فرود نیز دو کنترل‌کننده‌ی PID بر روی مختصات x و y روبات در حین حرکت، مجدداً پیاده‌سازی می‌شود. البته به دلیل عدم وجود سنسور دقیق مختصات، خطای مربوط به فاصله‌ی عرضی و طولی، با توجه به (جدول ۱-۵) توسط دوربین محاسبه شده و به کنترل‌کننده ارسال می‌شود. با محاسبه‌ی مختصات مطلوب در راستای طول و عرض، همانطور که در قسمت کنترل موقعیت توضیح داده شد، مختصات مطلوب به چرخش‌های رول و پیچ تبدیل شده و حرکت مطلوب را ایجاد می‌نماید.

در فصل بعدی، مراحل سناریوی مذکور، با پیاده‌سازی مدل دینامیکی و کنترل‌کننده‌ها بر روی نرم افزار مطلب شبیه‌سازی شده و نتایج آن در شرایط مختلف بررسی می‌گردد.



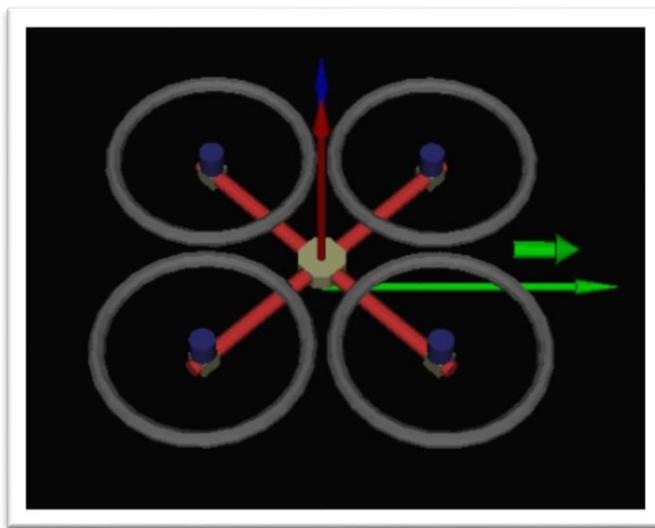
شکل ۳-۵ بلوک دیاگرام فرود روبات

فصل ششم

شبیه‌سازی

شبیه‌سازی

پس از انجام تمام محاسبات و طراحی‌های تئوری، نوبت به شبیه‌سازی مدل‌سازی دینامیکی، کنترل‌کننده، سنسورها، خطاهای و تمام اجزای مدار پیش از پیاده‌سازی می‌باشد. تا با بررسی دقیق عملکرد مدل شبیه‌سازی شده، از هرگونه عامل ناپایداری جلوگیری به عمل آید. شبیه‌سازی در محیط سیمولینک^۱ نرم افزار متلب^۲ انجام شده است. همچنین برای نمایش شهودی نتایج، از یک مدل کوادروتور گرافیکی آماده طراحی شده در نرم افزار SolidWorks (شکل ۱-۶) استفاده شده است. ورودی این مدل گرافیکی، زوایای گردش، چرخش و پیچش و موقعیت‌های x، y و z است که پس از کنترل و اعمال به مدل دینامیکی طراحی شده در نرم افزار متلب، به ورودی مدل SolidWorks نیز منتقل می‌گردد تا نحوه عملکرد یک روبات کوادروتور واقعی به صورت کاملاً شهودی مجسم گردد. مراحل شبیه‌سازی به دو بخش مجزای ۱- شبیه‌سازی پرواز ۲- شبیه‌سازی فرود تقسیم بندی شده است.



شکل ۱-۶ نمایی از محیط گرافیکی نمایش کنترل کوادروتور در نرم افزار SolidWorks

¹ Simulink

² Matlab

۱-۶- پارامترهای روبات

در فصل (۲) معادلات حالت به ازای پارامترهای دینامیکی روبات مشخص شد که این پارامترها به صورت مجهول در رابطه قرار گرفت. در هنگام شبیه‌سازی، بایستی پارامترها تا حد امکان به روبات مورد آزمایش نزدیک باشند. (جدول ۱-۶) مقدار پارامترهای روبات مورد آزمایش را نمایش می‌دهد.

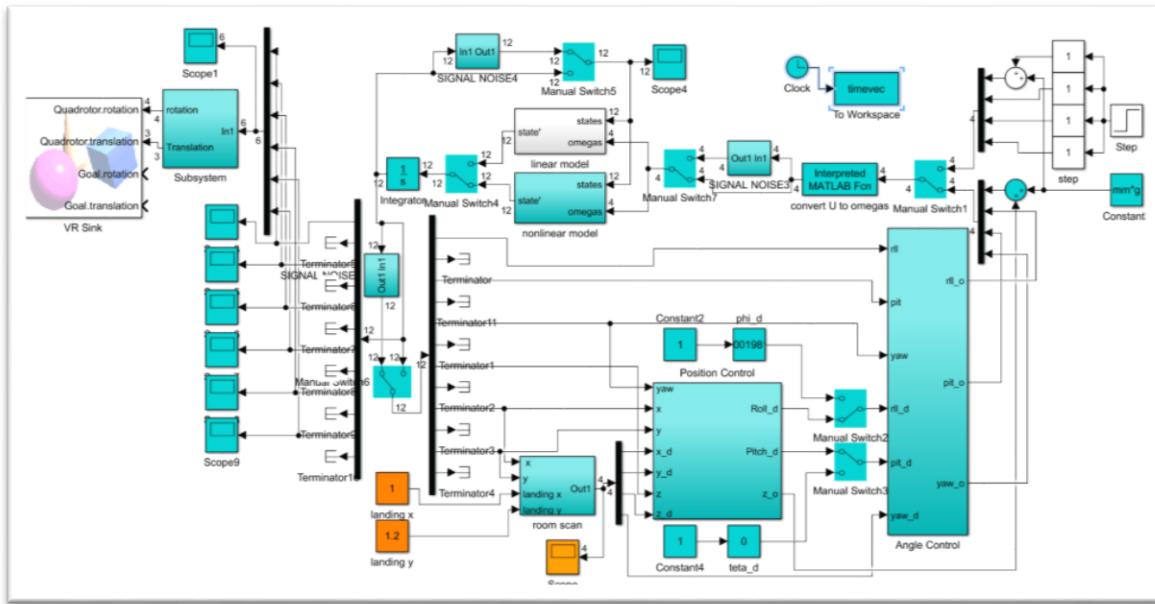
جدول ۱-۶ پارامترهای اندازه‌گیری شده مدل واقعی

بازوی روبات (متر)	$(kg.m^2)J_r$	I_{zz} ($kg.m^4$)	I_{yy} ($kg.m^4$)	I_{xx} ($kg.m^4$)	جرم (کیلوگرم)
۰/۴۴	$۳/۷۴۰۴ \times 10^{-۵}$	۰/۰۱۷۲	۰/۰۰۸۶	۰/۰۰۸۶	۱/۴۷

بازوی روبات برابر است با فاصله انتهای پره‌های آن از مرکز جرم. همچنین مقادیر اینرسی کوادکوپتر در زوایای مختلف با تخمین نرم افزاری دینامیکی محاسبه شده است که به اندازه‌ی کافی دقیق نیست و برای تست عملی لازم است تا مجدداً توسط نرم افزار قوی تری توسط یک مهندس مکانیک اندازه‌گیری شده و در محاسبات قرار گیرد.

۲-۶- شبیه‌سازی پرواز

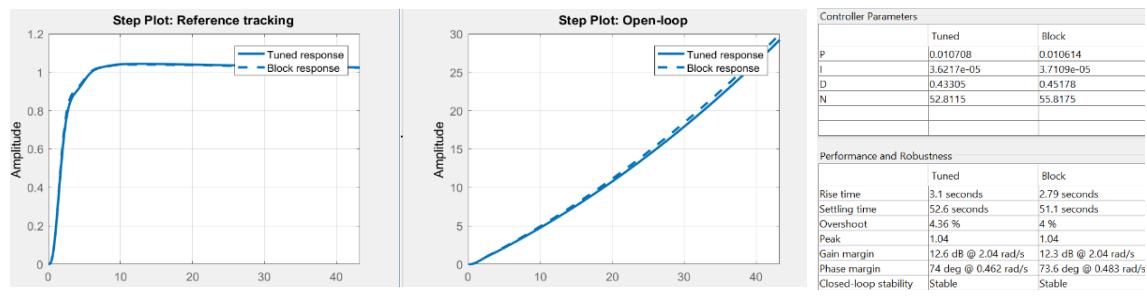
در ابتدا پرواز روبات شبیه‌سازی شده است که شامل شبیه‌سازی مدل دینامیکی روبات، کنترل کننده‌ها، سنسورها و خطاهای می‌شود. بدین گونه که مقادیر مطلوب موقعیت روبات (x, y, z) و همچنین میزان گردش مطلوب (yaw) را بتوان به سیستم اعمال نموده، در خروجی تمامی حالات مطلوب شامل موقعیت در سه جهت و زوایای چرخش در سه جهت محاسبه می‌گردد (شکل ۲-۶).



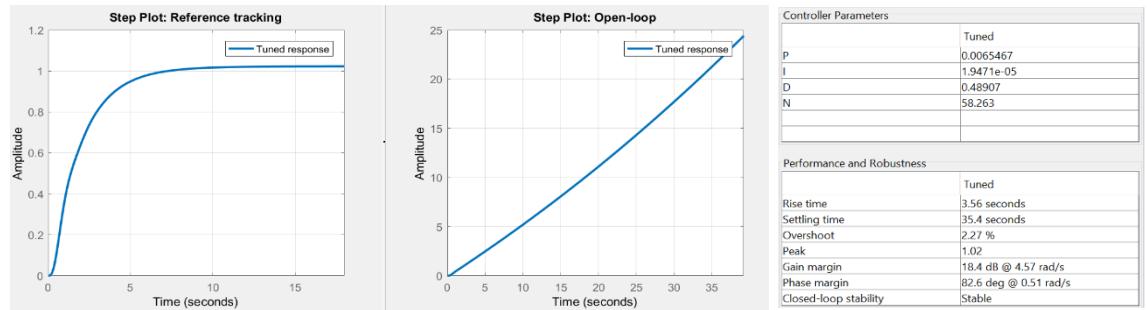
شکل ۲-۶ شبیه‌سازی پرواز روبات در نرم افزار متلب

پاسخ پله‌ی کنترل کننده‌ی هر یک از محورهای x, y, z و همینطور زوایای رول، پیچ و یاوه، به ترتیب در (شکل ۳-۶)، (شکل ۴-۶)، (شکل ۵-۶)، (شکل ۶-۶)، (شکل ۷-۶) و (شکل ۸-۶) مشاهده می‌شوند. در هر تصویر ۵ نمودار موجود است که شامل پاسخ پله‌ی حلقه بسته، پاسخ پله‌ی حلقه باز و پارامترها مشاهده می‌شوند.

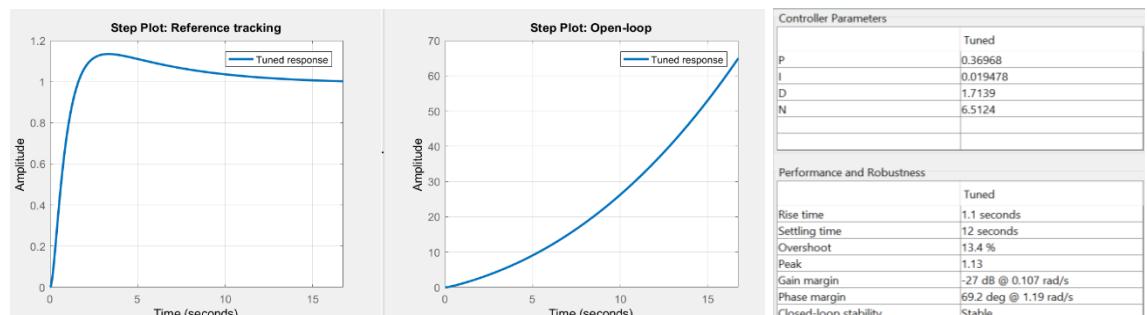
دلیل فراجهش زیاد سیستم در برخی محورها، بزرگ بودن ضریب اننتگرال گیر می‌باشد که برای دفع اغتشاشات و به خصوص برای جبران خطای اندازه‌گیری پارامترهای سیستم لازم می‌باشد. برای مثال اغتشاش ثابت گرانش زمین در راستای محور Z تنها با وجود اننتگرال گیر جبران می‌شود، و وجود این اغتشاش سبب افزایش زمان نشست و پاسخی بدون فراجهش در راستای Z می‌شود. همچنین با بالا بردن دقت اندازه‌گیری پارامترهای سیستم، می‌توان ضریب اننتگرال گیر در راستای X و Y را کاهش داد. این کار می‌تواند سبب بهبود پاسخ سیستم به طور قابل توجهی شود. اثبات می‌شود که سیستم خطی‌سازی شده به ازای گین‌های مشخص شده بدون در نظر گرفتن اثرات نویز و اغتشاش پایدار می‌شود.



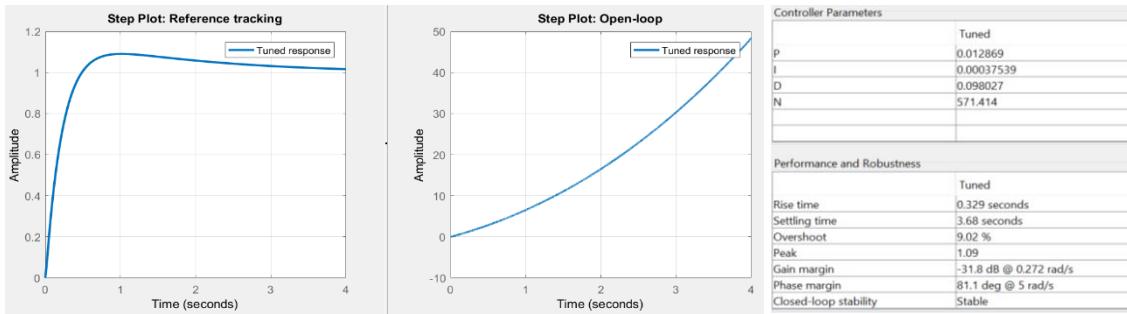
شکل ۳-۶ پاسخ پله و پارامترهای کنترلر PID محور X



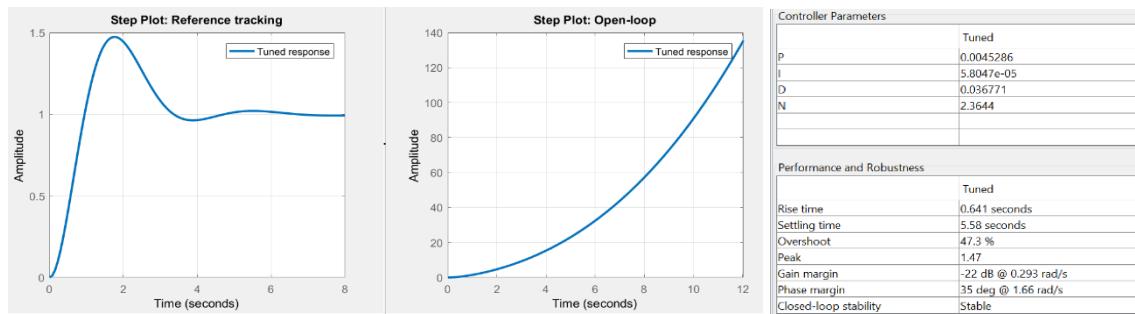
شکل ۴-۶ پاسخ پله و پارامترهای کنترلر PID محور Y



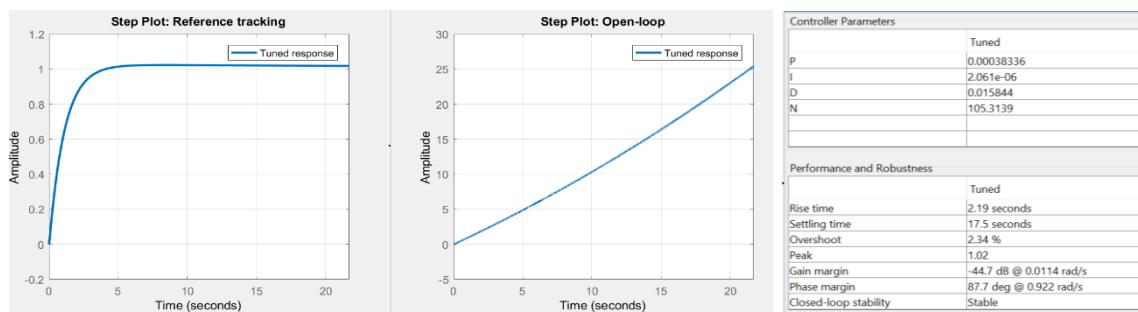
شکل ۵-۶ پاسخ پله و پارامترهای کنترلر PID محور Z



شکل ۶-۶ پاسخ پله و پارامترهای کنترلر PID محور رول



شکل ۶-۷ پاسخ پله و پارامترهای کنترلر PID محور پیچ



شکل ۶-۸ پاسخ پله و پارامترهای کنترلر PID محور یاو

۶-۳- شبیه‌سازی فرود

پس از انجام شبیه‌سازی عملیات کلی پرواز، نیاز به تعریف عملیات فرود می‌باشد. به این صورت که با توجه به اینکه تعریف پروژه، تنها فرود روبات است، به عنوان فرض اولیه، در نظر می‌گیریم که روبات در حالت پرواز است که دستور فرود صادر می‌شود. بنابراین شرایط اولیه‌ی روبات به صورت زیر خواهد بود:

$$x_0 = 35 \text{ cm}$$

$$y_0 = 35 \text{ cm}$$

$$z_0 = 2 \text{ m}$$

$$\text{yaw}_0 = 0^\circ$$

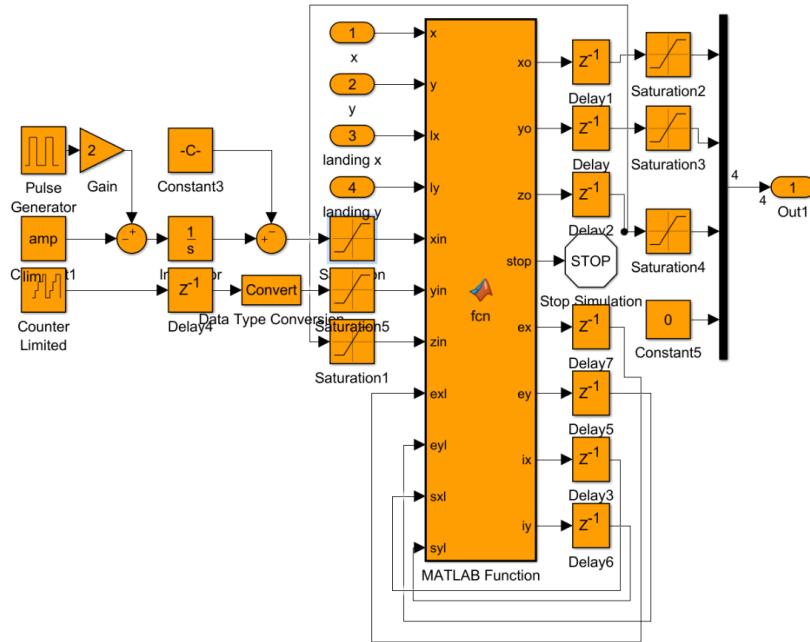
علت انتخاب این مقادیر این است که با توجه به اینکه مختصات مرکز روبات مورد کنترل قرار گرفته است، لازم است تا حرکت با یک فاصله‌ی امن از دیواره‌ها شروع شود که با توجه به اینکه طول بازوی روبات ۲۲ سانتی متر است، این محدوده‌ی امن در حدود ۱۳ سانتی متر در نظر گرفته شده است که در مجموع فاصله‌ی ۳۵ سانتی متری از مرکز روبات را موجب می‌شود.

از آن جایی که شبیه‌سازی پردازش تصویر در نرم افزار ممکن نبوده، پردازش تصویر به صورت عملی پیاده‌سازی شده و نتایج آن به محیط شبیه ساز فرود روبات منتقل می‌شود. همچنین برای شبیه‌سازی، ابعاد محیط مورد جستجو توسط کاربر قابل انتخاب است که به طور پیش فرض ابعاد اتاق ۳ در ۳ متر در نظر گرفته شده است.

در این قسمت، روبات، پس از قرار گیری در موقعیت اولیه، شروع به جستجوی اتاق برای یافتن محل فرود می‌نماید. به این صورت که در ارتفاع و عرض ثابت، شروع به افزایش طول نموده و در صورت رسیدن به انتهای اتاق، طول را ثابت نگاه داشته و عرض خود را به اندازه‌ی یک متر افزایش می‌دهد و به همین ترتیب ادامه می‌دهد تا دستور فرود ارسال گردد.

با تشخیص تصویر توسط دوربین، فرمان فرود به برنامه داده می‌شود و خطای طول و عرض، با استفاده از دوربین و تصویر دریافتی در همان زمان محاسبه شده و به مرکز کنترل فرود ارسال می‌گردد. در صورتی که مقدار این خطای ۱۰ سانتی متر کمتر باشد، روبات شروع به کاهش ارتفاع می‌نماید. به محض اینکه خطای بیش از ۱۰ سانتی متر شد، روبات در ارتفاع ثابت، شروع به کنترل موقعیت طولی و عرضی خود با استفاده از کنترل کننده‌ی PID بر روی خطاهای طولی و عرضی دریافتی از دوربین، می‌نماید. در نهایت در ارتفاع ۱۵

سانتی متری موتورها خاموش شده و روبات در محل مورد نظر، با خطای نهایتا ۱۰ سانتی متری فرود می‌آید.
شکل ۹-۶.



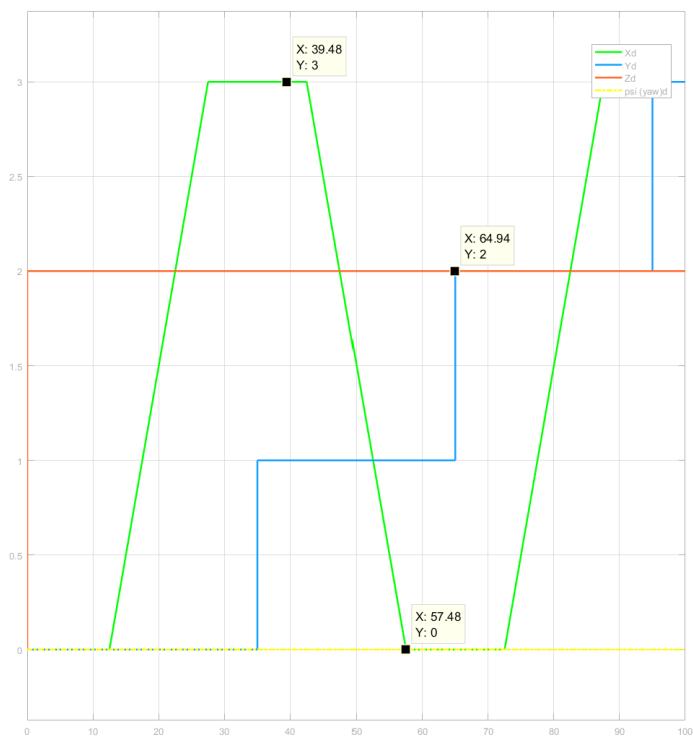
شکل ۹-۶ شبیه‌سازی فرود روبات در نرم افزار متلب

۶-۴-۶- خروجی شبیه‌سازی

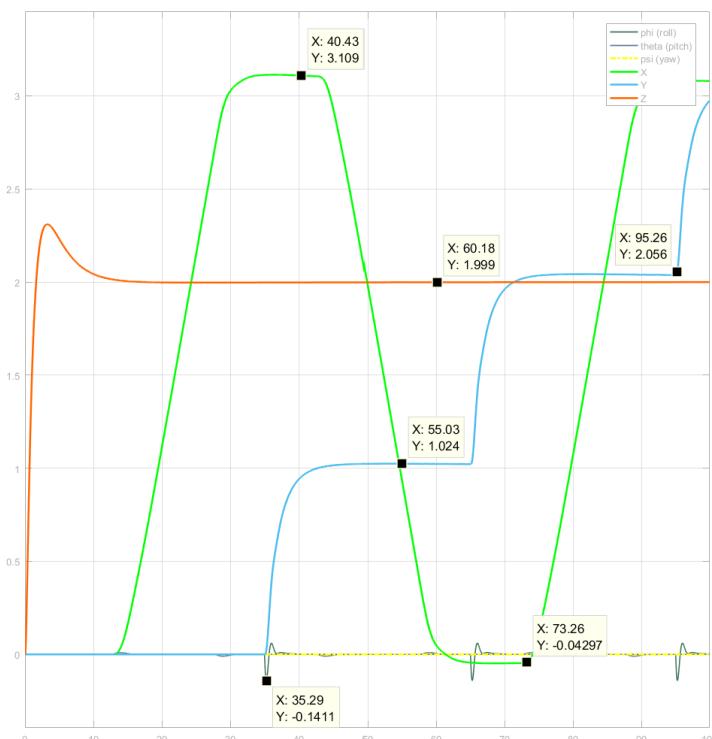
در نهایت نتایج شبیه‌سازی در سه حالت زیر مورد بررسی قرار گرفته است. در تصاویر به ترتیب رنگ‌های سبز، آبی و نارنجی مشخص کننده نمودار حرکت x ، y و z هستند.

۶-۴-۶-۱- مدل دینامیکی خطی بدون نویز

در ابتدا، برای بررسی اولیه، کنترلر به سیستم خطی‌سازی شده‌ی بدون نویز اعمال شد. برای بررسی عملکرد کلی پرواز در این حالت، تصویری قرار داده نشده و حرکت مطلوب روبات از ابتدا تا انتهای اتاق در (شکل ۱۱-۶) و حرکت واقعی آن در (شکل ۱۱-۶) قابل مشاهده است. در این مرحله مشاهده می‌شود که هنوز فاصله‌ی از دیواره‌ها رعایت نشده‌است. این مساله در آخرین مرحله اصلاح می‌گردد.



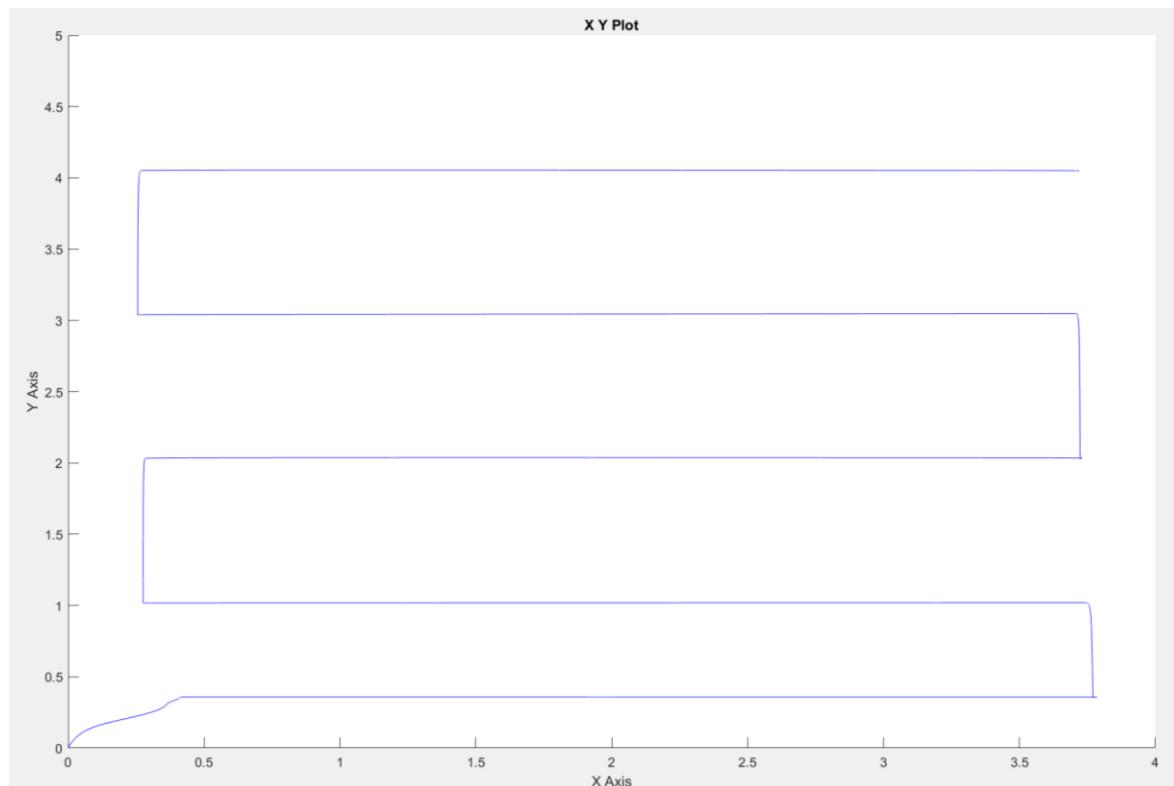
شکل ۱۰-۶ حرکت مطلوب مدل خطی بدون نویز از ابتدا تا انتهای



شکل ۱۱-۶ حرکت واقعی مدل خطی بدون نویز از ابتدا تا انتهای

همانطور که در تصاویر مشاهده شد، در حالت بدون خطأ و نویز، در حرکت در محور X خطای در حدود ۱۵ سانتی متر در پاسخ شبیه مشاهده می‌شود. همچنین خطای محور Z در بیشترین حالت ۲/۴ سانتی متر و خطای محور Z کمتر از ۱ میلیمتر است.

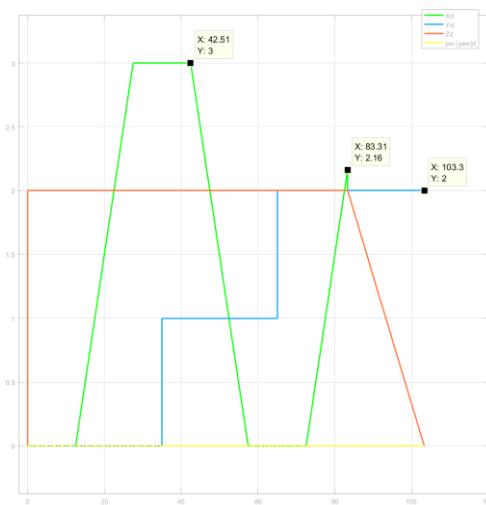
همچنین در (شکل ۱۲-۶) به طور مشابه به سیستم بدون نویز خطی و بدون محل فرود دستور فرود صادر می‌شود. تصویر، این بار مسیر حرکت روبات را در صفحه xy (دید از بالا) نمایش می‌دهد. تفاوت این حالت با حالت قبل، ابعاد اتاق است که در اینجا ۴ در ۵ متر در نظر گرفته شده است. همچنین در این قسمت، برای رفع مشکلات موجود در شبیه‌سازی‌های پیشین، فاصله‌ی روبات از دیواره‌ها مد نظر قرار گرفته شده است.



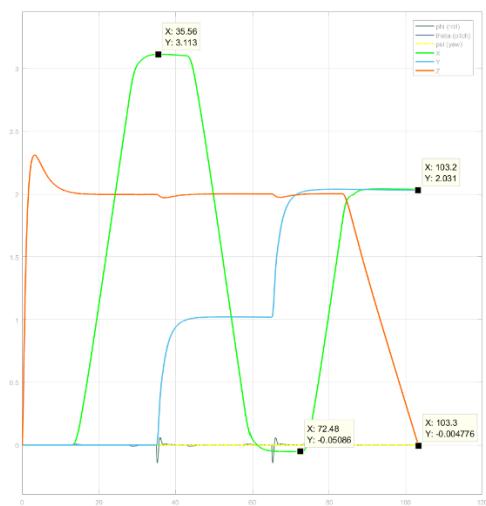
شکل ۱۲-۶ مسیر حرکت روبات خطی بدون نویز در صفحه xy

۱۴-۶-۲- مدل دینامیکی غیرخطی بدون نویز

در مرحله‌ی بعد خروجی کنترل‌کننده، به جای اعمال به سیستم خطی‌سازی شده، مستقیماً به سیستم غیرخطی اعمال می‌شود تا عملکرد روبات توسط سیستم واقعی، واضح‌تر گردد. در اینجا علامت فرود در نقطه‌ی (2,2) قرار گرفته است. (شکل ۱۳-۶) دستورات مطلوب برای ارسال به روبات و (شکل ۱۴-۶) عملکرد واقعی روبات را نمایش می‌دهند.

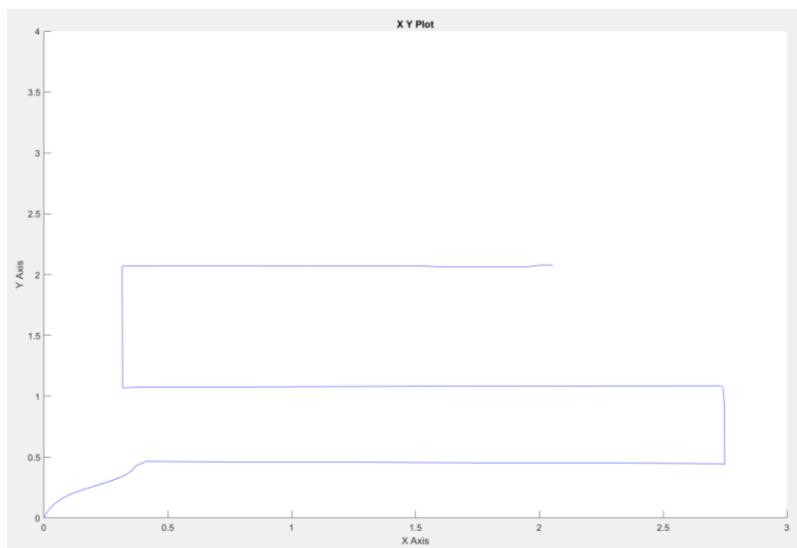


شکل ۱۳-۶ حرکت مطلوب مدل غیرخطی بدون نویز از ابتدا تا انتهای



شکل ۱۴-۶ حرکت واقعی مدل غیرخطی بدون نویز از ابتدا تا انتهای

با توجه به مقادیر نمودارها، خطای محور x در مجموع در حدود ۱۵ سانتی متر بوده و در نقطه‌ی فرود، ۳ سانتی متر خطا وجود دارد. همچنین در محور y در مقصد، ۳ سانتی متر خطا مشاهده می‌شود. در (شکل ۱۵-۶) نیز مسیر حرکت همین روبات در صفحه xy مشاهده می‌شود. با توجه به تصویر مشاهده می‌شود که روبات، در نهایت در محدوده‌ای که قصد فرود داشته، طول و عرض خود را حفظ کرده‌است.

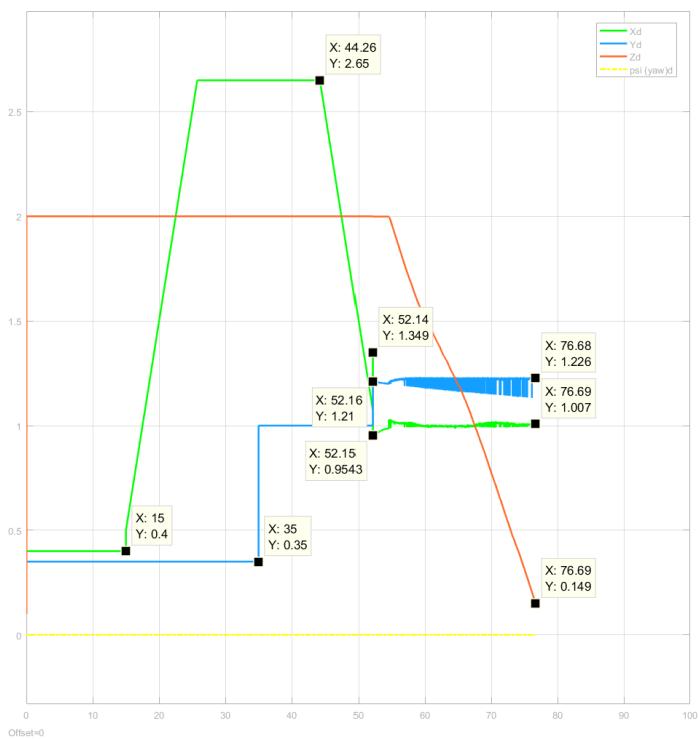


شکل ۱۵-۶ مسیر حرکت روبات غیرخطی بدون نویز در صفحه xy

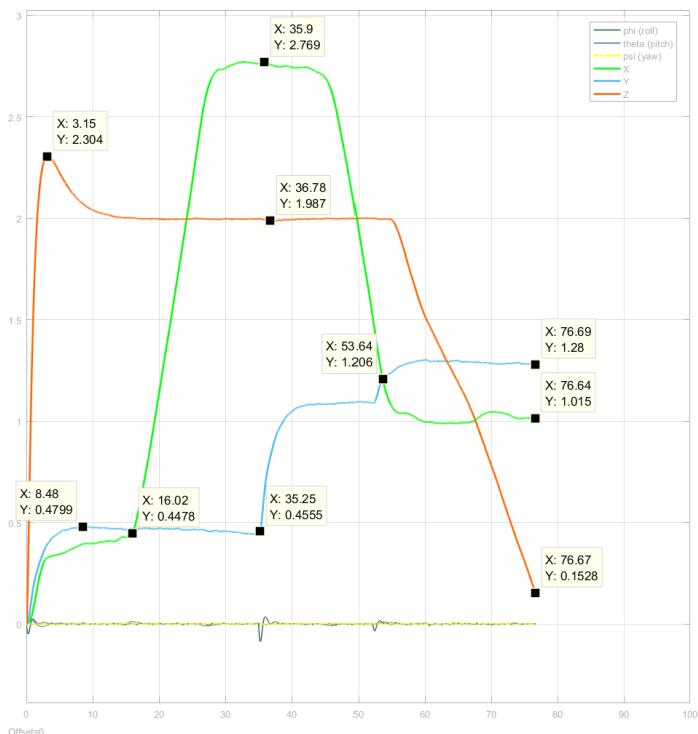
۳-۴-۶- مدل دینامیکی غیرخطی نویزی

در نهایت با اعمال نویز به سیستم غیرخطی، عملکرد شبیه‌سازی را تا حد ممکن به واقعیت نزدیک می‌نماییم. نویز اعمالی، نویزی سفید است که در همه‌ی محورها و حالات تعریف شده برای مدل و در دو نقطه‌ی ورودی کنترلر و حلقه‌ی فیدبک مدل دینامیکی با توان $^{10} - 5$ اعمال گشته است. چرا که در محیط فضای بسته، تنها نویز حاصل از مواد مغناطیسی بر روی سنسورها وجود داشته و یا خطای خود سنسورها یا اختلاف عملکرد موتورها با یکدیگر را می‌توان عامل وجود نویز دانست.

در این قسمت فاصله از دیواره‌ها رعایت شده و حتی برای بررسی تمام حالات، محل فرود در مختصات دشوار تری نسبت به حالات قبل (۱.۲، ۱.۱) قرار داده شده است. نتایج نهایی مقادیر مطلوب و مقادیر واقعی به ترتیب در (شکل ۱۶-۶) و (شکل ۱۷-۶) قابل مشاهده اند.



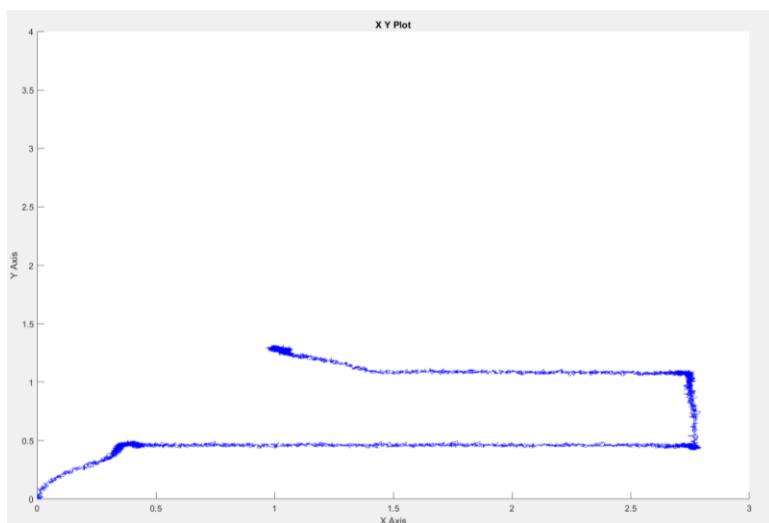
شکل ۱۶-۶ حرکت مطلوب مدل غیرخطی نویزی از ابتدا تا انتهای



شکل ۱۷-۶ حرکت واقعی مدل غیرخطی نویزی از ابتدا تا انتهای

با توجه به تصاویر فوق و با توجه به اینکه میزان خطای مجاز در نقطه‌ی فرود، ۱۵ سانتی متر در نظر گرفته شده است، با اعمال کنترل‌کننده PID به دو محور طول و عرض، خطای محورهای x، y و z در حالت کلی قابل قبول و مناسب بوده (نهایتاً ۵ سانتی متر) و در نقطه‌ی فرود در محورهای طول و عرض، به ترتیب $1/5$ و ۸ سانتی متر خطا مشاهده می‌شود که با استفاده از برنامه، می‌توان اندازه‌ی این خطا را نیز کاهش داد. همچنین با مشاهده‌ی مقدار نهایی ارتفاع، در ارتفاع ۱۵ سانتی متری، موتورها خاموش شده و برنامه متوقف می‌شود.

به طور مشابه، می‌توان حرکت روبات در این حالت را در (شکل ۱۸-۶) مشاهده کرد.



شکل ۱۸-۶ مسیر حرکت نویزی غیرخطی

۵-۶- نتیجه‌گیری

با مشاهده‌ی خروجی شبیه‌سازی، از آن جایی که پارامترهای روبات تا حد امکان به مدل واقعی نزدیک بوده و محل فرود نیز در بدترین شرایط دارای خطای بیشینه ۱۵ سانتی متر و قابل کنترل است، نوبت به ساخت روبات واقعی می‌رسد. لذا در فصل بعد به معرفی سخت افزار روبات پرداخته خواهد شد. اما پس از ساخت روبات، تلاش بر این است تا قبل از تست کامل پرواز، ابتدا نتایج پردازش تصویر نیز به شبیه‌سازی اضافه گردد و پس از نتیجه‌ی مناسب، پرواز واقعی صورت پذیرد.

فصل هفتم

سخت افزار

سخت افزار

سخت افزار روبات را می‌توان در محورهای اصلی زیر دسته‌بندی کرد:

- مکانیک
- الکترونیک

در ادامه به تفصیل هر یک از محورهای فوق، توضیح داده می‌شوند.

۱-۷-۱- مکانیک

مکانیک روبات، شامل بدن، موتورها و ملخ‌ها می‌شود. یکی از مسائل مهم هنگام ساخت کواد، انتخاب ماده‌(های) مناسب برای ساخت اجزای مختلف آن مانند قاب، بال و غیره است. از آنجایی که موتورها در هر دو محور الکترونیک و مکانیک قابل بررسی هستند، لذا با توجه به اهمیت موتورها و کنترل‌کننده‌های سرعت آن‌ها در قسمت الکترونیک، در بخش (۴-۲-۷-موتورها و کنترل‌کننده‌های سرعت) به توضیح آن‌ها و همچنین ملخ‌ها پرداخته خواهد شد.

۱-۷-۱- انتخاب جنس بدن

پارامترهایی که در انتخاب یک ماده مناسب برای این پرنده در نظر گرفته می‌شوند عبارتند از:

۱. سختی: نشان دهنده میزان الاستیسیته‌ی ماده
۲. استحکام: نشان دهنده حداکثر میزان تحمل فشار قبل از وارد آمدن آسیب جدی به ماده
۳. چگالی: نسبت وزن به حجم برای تعیین ابعاد موثر پرنده
۴. هزینه

در (پیوست ۱) به مقایسه مواد مختلف و تحلیل هر کدام پرداخته می‌شود.

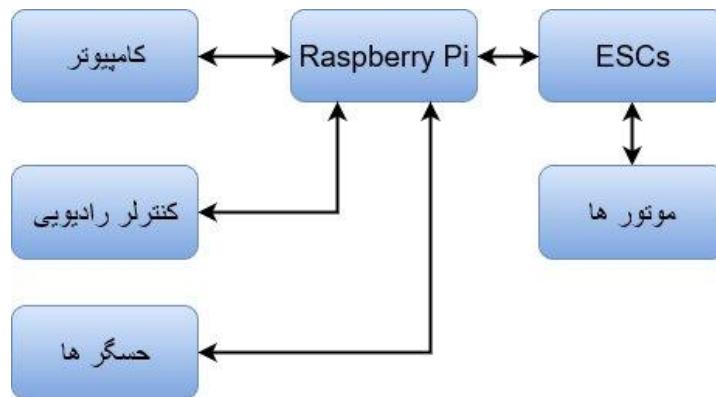
جدول ۱-۷ این جدول خواص مختلف ماده را با امتیاز دهی از ۱۰ مقایسه می‌نماید.

هزینه	در دسترس بودن	استحکام	سختی	چگالی	ماده
۱	۳	۹	۸	۸	فیبرکربن
۱	۵	۹	۹	۶	تیتانیوم
۷	۷	۶	۶	۷	آلومینیوم
۶	۷	۵	۶	۶	فایبرگلاس
۷	۳	۱۰	۱۰	۱	استیل
۹	۶	۴	۳	۴	پلی کربنات
۹	۷	۲	۲	۴	آکریلیک
۱۰	۱۰	۴	۴	۱۰	چوب بالسا

با بررسی انجام شده مشاهده شد که موادی مانند فیبرکربن، بالسا، آلومینیوم و فایبرگلاس به علت امتیازات مناسبی که نسبت به سایر مواد دارند در اولویت استفاده می‌باشند، فلز استیل با وجود سختی و استحکام بالا از چگالی زیادی نیز برخوردار است که باعث سنگینی پرنده می‌شود و در قطعاتی که نیازمند استحکام بسیار بالا می‌باشد می‌توان از آن استفاده نمود.

۷-۲- الکترونیک

برای پیاده‌سازی الکترونیک روبات، نیاز به یک پردازش‌گر مرکزی بوده که به ورودی‌ها و خروجی‌ها متصل گردد. (شکل ۱-۷) بلوک دیاگرام الکترونیک مدار را نمایش می‌دهد. در ادامه به معرفی پردازنده، تمام ورودی‌ها و خروجی‌ها پرداخته می‌شود.



شکل ۱-۷ بلوک دیاگرام الکترونیک روبات [29]

۷-۲-۱- مغز روبات (Erle-brain 3)

پردازنده‌ی هوشمند ۳ Erle-brain سومین نسل یک مغز مصنوعی شامل قابلیت‌های سامانه خلبان خودکار^۱، مبتنی بر لینوکس^۲ است که برای ساخت روبات‌ها به خصوص روبات‌های پرنده^۳ مورد استفاده قرار می‌گیرد و با پشتیبانی از سیستم عامل روبات (ROS^۴) منحصر به فرد می‌گردد. در (جدول ۲-۷) مشخصات و در (شکل ۲-۷) تصویری از آن مشاهده می‌گردد. در واقع Erle-brain یک نوع مغز هوشمند

¹ Autopilot

² Linux Based

³ Drones

⁴ Robot Operating System

روباتیک است که بر روی ریز کامپیوتر Raspberry Pi 3 پیاده‌سازی شده‌است. در ادامه به بررسی ویژگی‌های این ریز کامپیوتر و همچنین سنسورها اشاره خواهد شد.

جدول ۲-۷ جدول امکانات ۳ Erle-brain [31] [12] [30]

پردازنده	ARM Cortex-53 CPU
سنسورها	<ul style="list-style-type: none"> • سنسور جاذبه • ژیروسکوپ • قطب نمای دیجیتال • سنسور فشار • سنسور دما • برای اندازه‌گیری باتری ADC
WiFi	ماژول WiFi داخلی (802.11n, 2.4 GHz)
ورودی / خروجی‌ها ^۱	<ul style="list-style-type: none"> • USB ports • Full HDMI port • 10/100 Ethernet • جک صوتی 3.5mm
بلوتوث	کلاسیک Bluetooth 4.1
گرافیک	<p>افزایش گرافیک از ۲۵۰ مگاهرتز به ۴۰۰ مگاهرتز، Dual Core VideoCore IV GPU</p> <p>پشتیبانی از ۱۰۸۰p video at 60fps و ۲۶۴H، بالاتر از ۳۰fps</p> <p>خروجی ویدئو: HDMI rev ۱.۴ & ۱.۳ (PAL and NTSC)، Composite RCA</p>
RAM	یک گیگابایت رم (مانند نسخه قبلی)
نرم افزار	<ul style="list-style-type: none"> • Linux 3.8 kernel • Debian Wheezy file system • ROS Hydromedusa • mavros ROS • بسته

^۱ I/O s

	APM •
--	-------



شکل ۲-۷ تصویری از Erle-Brain 3 و مشخصات آن [12]

۲-۷-۲-۲- رزبری پای

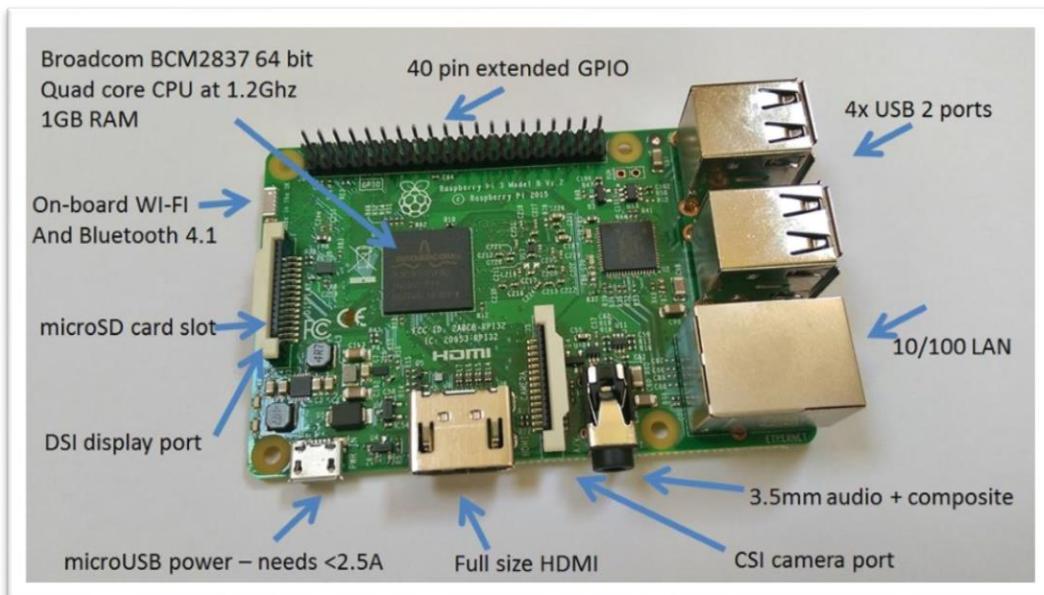
رزبری پای یک رایانه تک برد^۱ در ابعاد یک کارت اعتباری است که بنیاد رزبری پای آن را ساخته است. این رایانه دارای دیسک سخت نیست ولی به کمک کارت حافظه SD، راه اندازی می‌شود. می‌توان بر روی این رایانه، سیستم عامل‌های مختلفی نصب نمود. سیستم عامل اولیه‌ای که با آن عرضه شد توزیع لینوکس دیبان بود ولی نسخه‌هایی از اندروید نیز برای آن تهیه شده است.

پس از رونمایی از رزبری پای ۲، مایکروسافت مدعی شد که نسخه‌ای از ویندوز ۱۰ را برای رزبری پای ۲ بهینه کرده و به صورت رایگان منتشر خواهد کرد. رزبری پای قادر است به کمک یک کابل تصویر ویدئو و یک درگاه سوزنی، خروجی صدا و تصویر داشته و یا با کابل اچ دی ام آی^۲، خروجی تصویری با وضوح

^۱ Single-board computer

^۲ HDMI

بالا^۱ و صدایی استریو داشته باشد. درگاههای USB نیز برای آن در نظر گرفته شده‌است که می‌توان از آن برای ارتباطات مختلف استفاده کرد. از درگاههای USB برای صفحه کلید، ماوس، شبکه بی‌سیم و کلیه دستگاهها با پشتیبانی از درگاه مذکور می‌توان استفاده کرد. به کمک هاب USB می‌توان از پورت‌های بیشتری استفاده کرد. مدل‌های B نسخه یک و دو این رایانه یک درگاه شبکه نیز دارد که به کمک آن می‌توان به شبکه محلی و یا اینترنت متصل شد. از خروجی HDMI برای صدا، تصویر و کنترل آن باهم می‌توان بهره جست. بدین ترتیب با کمک فقط یک نمایشگر لمسی قادر است تبدیل به یک تبلت شود. این یک رایانه بسیار کوچک و ارزان با تمام قابلیت‌های یک رایانه واقعی است که می‌توان از آن در سیستمهای روباتیک، اتوماسیون، اینترنت اشیاء و سیستمهای نهفته^۲ استفاده کرد. در (شکل ۳-۷) تصویر این برد قابل مشاهده است. [30]



[32] Raspberry Pi 3 ۳-۷

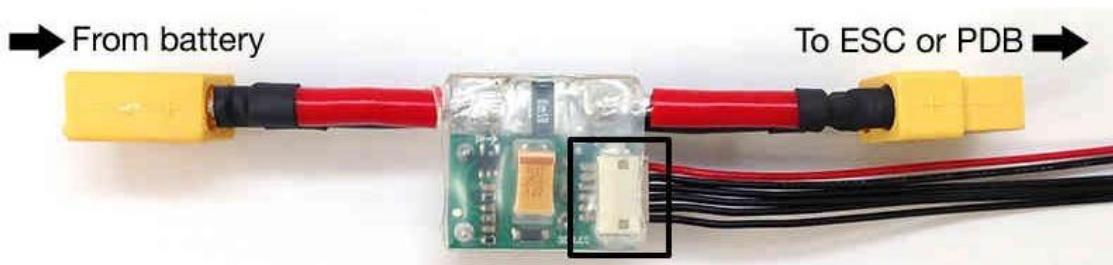
¹ HD (High Definition)

² Embedded Systems

۷-۲-۳- تغذیه

برای سیستم تغذیه‌ی این روبات، به صورت پیش فرض سه سلول باتری لیتیوم پلیمری در نظر گرفته شده که هر سلول آن، ولتاژی در محدوده‌ی $\frac{4}{7}$ تا $\frac{3}{7}$ ولت را دارا می‌باشد. لذا با توجه به قطعات و مازول‌هایی که در موارد بعدی توضیح داده می‌شوند، نیاز به ولتاژ تغذیه‌ی ۵ ولت خواهد بود. برای این منظور، از مازول APM 2.5 Power Module که به طور خاص، برای اتصال باتری به ارله (و به خصوص در روبات‌های پرنده) طراحی شده است، استفاده می‌گردد.

شکل ۴-۷) تصویری از این مازول را نمایش می‌دهد. همچنین با مشاهده (جدول ۳-۷) که نشان دهنده‌ی ویژگی‌های مازول است، به مزایای آن و علت انتخابش پی برده خواهد شد.



شکل ۴-۷ مازول تغذیه [33] APM Power Module

جدول ۳-۷ جدول ویژگی‌های الکترونیک مازول APM Power Module

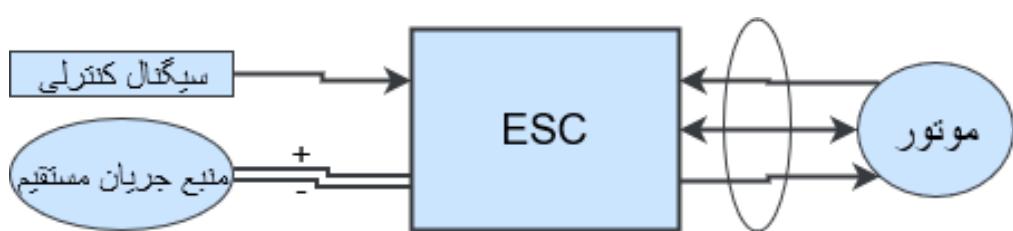
ولتاژ خروجی رگولاتور $\frac{5}{3}$ ولت	
بیشینه جریان خروجی رگولاتور $\frac{25}{2}$ آمپر	
بیشینه ولتاژ ورودی ۱۸ ولت	
بیشینه جریان قابل اندازه‌گیری ۹۰ آمپر	
اندازه‌گیری ولتاژ و جریان مبدل آنالوگ به دیجیتال ۵ ولت	

۷-۲-۴-۱- موتورها و کنترل کننده‌های سرعت

۷-۲-۴-۱-۱- موتورها

برای کنترل کوادروتورها و سایر مولتی‌روتورها از موتورهای بدون جاروبک^۱ به عنوان عملگر استفاده می‌گردد.^۲ ESC‌ها برای راه اندازی و کنترل سرعت چرخش این گونه موتورها مورد استفاده قرار می‌گیرند. (در واقع مدارهای الکترونیکی هستند که با تولید یک توان الکتریکی سه فاز، موتور را به حرکت در می‌آورند و سرعت آن را کنترل می‌نمایند.) مدار اتصال این نوع موتورها در (

شکل ۵-۷) دیده می‌شود.



جريان سه فاز ناشی از منبع ولتاژ
یابین سه فاز الکترونیکی

شکل ۵-۷ بلوک دیاگرام اتصال موتور و ESC

ها و موتورهای موجود در بازار، طیف گسترده‌ای از توان و جریان نامی را در بر می‌گیرند. در این ESC بخش، عمدتاً به بررسی موتورهای ESC و T-Motor می‌پردازیم. این موتورها جزو محدود موتورهای موجود در بازارند که اطلاعات کاملی از آن‌ها به سهولت قابل دسترسی است. همچنین از کیفیت مناسبی بهره مند هستند. برای کنترل کوادروتورها و سایر مولتی‌روتورها از موتورهای بدون جاروبک به عنوان عملگر استفاده می‌گردد.

¹ Brushless Motors

² Electronic Speed Control

موتورهای جریان مستقیم بدون جاروبک شبیه به موتورهای سنکرونی هستند که با منبع دیسی تغذیه می‌شوند که توسط اینورتر مجتمع برای به حرکت درآوردن موتور به سیگنال الکتریکی AC تبدیل می‌شود؛ سنسورها و قطعات الکترونیکی دیگری نیز خروجی اینورتر را کنترل می‌نمایند. موتورهای بدون جاروبک همچنین به صورت موتورهای پله‌ای وصف می‌شوند، هرچند عنوان موتور پله‌ای برای آن دسته از موتورها به کار می‌رود که طراحی آنها به گونه‌ای است که به حالت‌هایی عمل نمایند که روتور آن به سرعت در نقطه زاویه‌ای تعریف شده باشد.

جدول ۴-۷ مقایسه موتورهای بدون جاروبک DC با موتور DC معمولی

معیار مقایسه	موتور BLDC	موتور DC معمولی (با جاروبک)
کمotaسیون	کمotaسیون الکترونیکی(با سنسور)	کمotaسیون با جاروبک
نگهداری و تعمیر	کمتر به تعمیر نیاز دارد(به دلیل عدم وجود جاروبک)	بیشتر و به صورت منظم
- مشخصه گشتاور- سرعت	صاف و قابلیت کارکرد در رنج وسیعی از سرعت تحت بارنامی	صاف ملایم و در سرعت‌های بالا اصطکاک زیادتر است
راندمان	بالا	متوسط
نسبت توان به حجم	بالا	متوسط و کم
ایرسی روتور	کم	بالا
هزینه ساخت	زیاد	کم

یکی از برندهای معروف در زمینه موتورهای پروازی T-motor می‌باشد . از مزیت‌های این گروه موتورها می‌توان به دسترسی آسان به تمامی اطلاعات مورد نیاز آن و همچنین موجود بودن آن در مرکز فروش اشاره نمود.

در گزارش‌ها، میزان عملکرد موتور در درصدهای مختلف بار نامی و دماهای مختلف آورده شده است که بسیار حائز اهمیت می‌باشد. یکی از پارامترهای مهم در داده‌های موتور، تراست^۱ (نیروی پیشران یا جلو برنده) است که در ادامه توضیح آن را می‌بینیم.

پارامتر تراست به وزن راکت، موتور جت ، موتور پرایپلر و ... بستگی داشته، همچنین میزان عملکرد و کارامدی آن را توضیح می‌دهد که با ارائه‌ی بازده^۲ در گزارشات آورده شده است. این پارامتر برای موتورهای موجود بین ۵ تا ۱۵ می‌باشد.

موتورهایی از این قبیل، معمولاً بهترین بازدهی عملکرد خود را در ۵۰٪ ظرفیت نامی^۳ داشته و باید به گونه‌ای انتخاب گردد که محدوده‌ی نامی فعالیت آن‌ها در حدود ۵۰٪ باشد. لذا پس از ارزیابی نیروی تولیدی نامی مورد نیاز هر موتور، بایستی موتوری انتخاب شود که با ۵۰٪ ظرفیت خود، قادر به تامین نیروی نامی به انضمام نیروی مورد نیاز برای بلند کردن خود باشد. باقی مانده‌ی ظرفیت موتورها، برای انجام مانورهای حرکتی، مواجهه مناسب با اغتشاشات و ... ذخیره می‌گردد.

یکی از موتورهای مورد توجه، MT2216 با محدوده دمایی ۳۰ تا ۵۰ درجه می‌باشد. این موتور همچنین دارای ۷۵ گرم وزن و قیمتی در حدود ۵۰ دلار است. عکسی از این موتور در (شکل ۶-۷) نشان داده شده است.

¹ Thrust

² Efficiency

³ Throttle



شکل ۶-۷ MT2216 و کاربرد آن در کوادروتور [34]

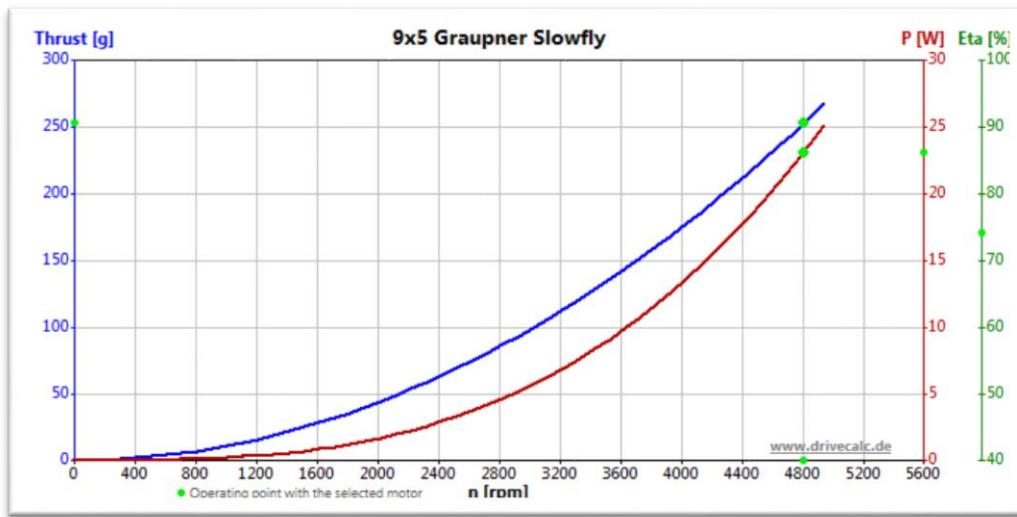
عملگرهای این وسیله پرنده از نوع موتورهای بدون جاروبک جریان مستقیم (BLDC^۱) هستند. موتور BLDC در واقع یک گونه از موتورهای سه فاز سنکرون است که نسبت به موتور جریان مستقیم با جاروبک، دارای راندمان و چگالی توان (توان به وزن) بالاتری است. از آنجایی که وزن و مدت پرواز وسیله پرنده اهمیت بسیار بالایی دارد، چهار موتور BLDC به عنوان عملگر سیستم مورد نظر انتخاب شده اند.

برای ایجاد جریان سه فاز متناوب از روی منبع تغذیه جریان ثابت (باتری)، یک مدار کلیدزنی استفاده می‌شود. این مدار که در واقع سرعت چرخش موتور BLDC را کنترل می‌نماید اصطلاحاً کنترل‌کننده الکترکی سرعت (ESC^۲) نامیده می‌شود. به دلیل کاربرد گسترده موتورهای BLDC در ساخت وسایل پرنده بدون سرنشین، مدارات کنترل و راهاندازی (ESC) این موتورها نیز به صورت تجاری و آماده در اختیار هستند.

علاوه بر موتور، مدار راهانداز موتور و باتری، پره‌ها نیز به عنوان بخشی از عملگر در نظر گرفته می‌شود. پره‌های یک کوادروتور برخلاف هلیکوپتر، دارای زاویه حمله ثابتی است و تغییر نیروی بالابرند نهایاً با تغییر سرعت چرخش پره انجام می‌شود. ابعاد و گام یک پره، پارامترهای فیزیکی‌ای هستند که میزان نیروی بالابرند نسبت به سرعت چرخش پره را مشخص می‌کنند. در (شکل ۶-۷) ارتباط بین این اجزا نشان داده شده است.

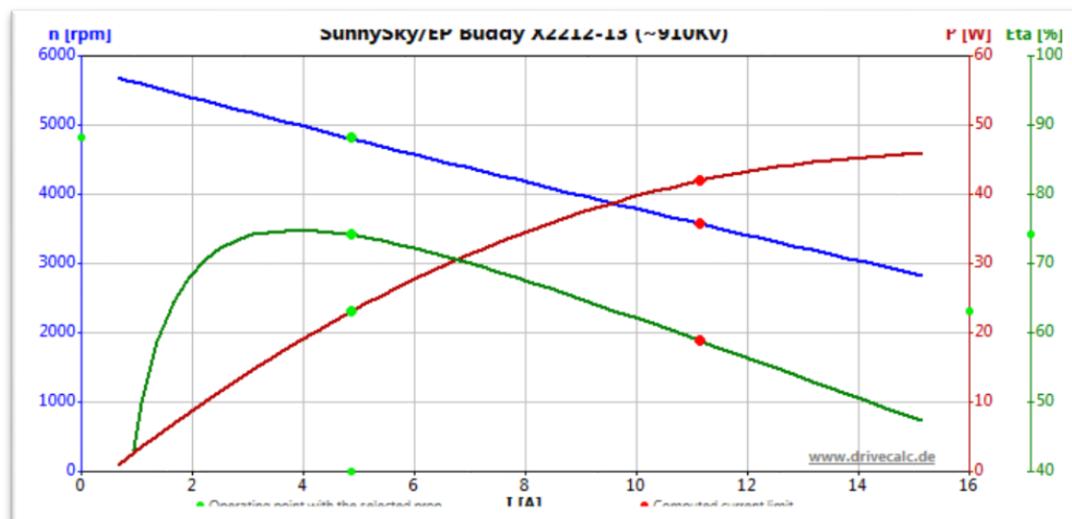
^۱BrushLess Direct Current

^۲Electrical Speed Controller



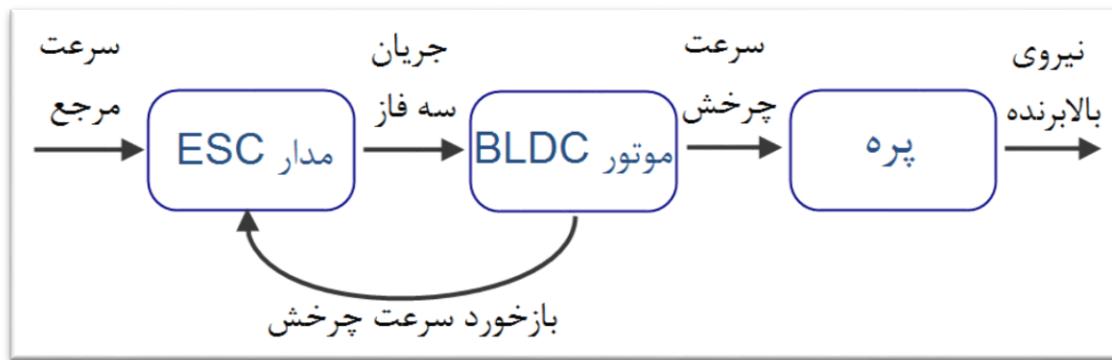
شکل ۷-۷ نمودار نیروی بالابرند و توان خروجی پره نسبت به سرعت چرخش [34]

با توجه به نمودار (شکل ۸-۷)، این پره در سرعت ۴۸۰۰ دور در دقیقه نیروی بالابرند معادل وزن یک جرم ۲۵۰ گرمی را ایجاد می‌کند. همچنین با توجه به نمودار (شکل ۸-۷)، در این سرعت چرخش، موتور مورد نظر $1\frac{1}{2}$ وات توان در خروجی خود دارد (به پره انتقال داده شده). همچنین راندمان این موتور در این نقطه کاری، حدود ۷۴ درصد است، بنابراین توان ورودی موتور (توان مصرفی) برابر $3\frac{1}{2}$ وات خواهد شد که با فرض اولیه توان زیر ۵۰ وات برای هر موتور هم‌خوانی دارد.



شکل ۸-۷ نمودار راندمان، توان خروجی و سرعت چرخش موتور نسبت به جریان مصرفی [34]

مدار راهانداز موتور، سرعت مرجع چرخش موتور را از پردازنده اصلی روبات دریافت می‌کند. ارتباط پردازنده و مدار راه انداز موتور توسط یک سیگنال با مدولاسیون عرض پالس (PWM^۱) برقرار می‌شود.



شکل ۹-۷ رابطه میان موتور، مدار راهانداز و پره برای ایجاد نیروی بالابرندۀ

انتخاب مشخصات موتور، راه انداز موتور، باتری و پره براساس وزن و ابعاد کلی پرنده انجام می‌شود. به دلیل اینکه انتخاب هر یک از این قطعات بر روی انتخاب سایر قطعات تاثیر می‌گذارد، مشخصات یکی از این قطعات را باید به عنوان پارامتر ثابت در نظر گرفت. به همین منظور در طراحی روبات ابتدا باتری از نوع لیتیوم-پلیمر با ظرفیت ۲۸۰۰ میلی آمپر ساعت و ولتاژ ۱۲ ولت انتخاب شده است (۳۳/۶ وات ساعت).

برای انتخاب مناسب موتور و پره، باید به نمودار جریان مصرفی، نیروی تولیدی، راندمان و مدت پرواز مورد نیاز توجه کرد. بافرض مدت زمان پرواز ۱۰ دقیقه و وزن کلی پرنده (حدود ۱ کیلوگرم)، متوسط توان مصرفی هر موتور باید کمتر از ۵۰ وات باشد و با این توان باید بتواند نیروی بالابرندۀ معادل وزن یک جرم ۲۵۰ گرمی را ایجاد نماید. در این نمودار نیروی بالابرندۀ بر حسب گرم (نیروی لازم برای بالابردن n گرم) و توان خروجی بر حسب وات نمایش داده شده است.

^۱ Pulse Width Modulation

۷-۴-۲ ECS‌ها

های ESC برای راه اندازی و کنترل سرعت چرخش این گونه موتورها مورد استفاده قرار می‌گیرند. Flame 25 A با ۱۶ دلار قیمت، جریان قله^۱ ۳۰ آمپر، فرکانس پردازنده ۱۰۰ مگاهرتز و تعداد دور در دقیقه^۲ ۴۰۰۰۰ rpm انتخاب بسیار مناسبی است (شکل ۱۰-۷).

نحوهی عملکرد این درایورها به این صورت می‌باشد که از سمت پایین با سه سیم به موتور وصل شده از سمت دیگر سه سیم مثبت، منفی و پالس دارد که برای راه اندازی آن کافی است به پایه پالس آن اعمال PWM گردد.



شکل ۱۰-۷ تصویر ESC مورد انتخاب [35]

^۱ Peak current

^۲ Round Per Minute

۷-۲-۵ - حسگرها

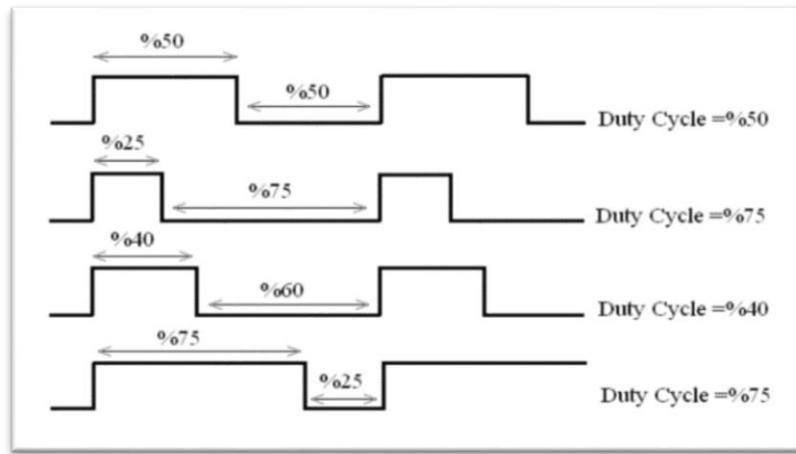
این برد که به صورت مستقیم بر روی برد رزبری پای نصب شده و دارای اتصال مستقیم به آن است، شامل بخش‌های اصلی زیر می‌شود (جدول ۵-۷).

جدول ۵-۷ اتصالات Raspberry Pi

نام واحد	چip مورد استفاده	نحوه اتصال	توضیحات
IMU ^۱ درجه آزادی	Invensense MPU-9250	SPI , I2C	شتاب سنج سه درجه آزادی MEMS قطب نمای دیجیتال سه درجه آزادی ژاپروسکوپ سه درجه آزادی قیمت مناسب
GPS	GPS uBlox Neo-7M	I2C	
دوربین دیجیتال RGB	HMC5883L	I2C	قطب نمای دیجیتال
دماسنجدیجیتال	Raspberry Pi Camera v2	Image Signal Processor (ISP)	high quality 5 megapixel Sony IMX219 image sensor custom designed add-on board for Raspberry Pi
مدار آنالوگ به دیجیتال ADC	ADS 1015	I2C	مقادیر خوانده شده از روی سنسورها، خصوصا سنسورهای آنالوگ، وابستگی به دما دارد. لذا برای تصحیح این مقادیر با توجه به دما، نیاز به دماسنجد است.
			جهت آگاهی از میزان شارژ باتری در هر لحظه

^۱ با توجه به این که ربات نیاز به کنترل در تمامی جهات در فضارا دارا می‌باشد، برای تشخیص انحراف در هر محور از فضا، از این مازول استفاده می‌گردد. چرا که با وجود قابلیت‌های بسیار زیاد، قیمت مناسبی دارد.

مدار مجتمع جهت ایجاد سیگنال PWM تا ۱۶ عدد کد برای ایجاد سیگنال کنترلی ESC مورد استفاده قرار گیرند.	I2C	PCA9685	PWM ^۱ راه انداز
--	-----	---------	----------------------------



شکل ۱۱-۷ کنترل موتور به وسیله‌ی مدولاسیون عرض پالس

۶-۲-۷- سیستم مخابراتی برای ارتباط بی سیم روبات با کامپیوتر کنترل کننده

تعامل اپراتور و پرنده توسط یک سیستم ارتباطی بی‌سیم انجام می‌شود. در حالت کنترل نیمه خودکار لازم است فرمانهای سطح بالا (کنترل موقعیت) از اپراتور به پرنده و بورد کنترلی آن ارسال شود. این ارتباط از طریق دستگاه کنترل رادیویی^۲ انجام می‌شود. این دستگاه کنترل رادیویی، متشکل از ۸ کانال ارتباطی است که در سمت گیرنده به صورت سیگنال‌های ۵۰ هرتز PWM دریافت می‌شوند. با استفاده از وقفه‌های

^۱ نحوه‌ی کنترل موتور به وسیله‌ی مدولاسیون عرض پالس (PWM) بدین گونه است که ابتدا یک فرکانس ثابت را انتخاب کرده و سپس برای افزایش سرعت موتور Duty cycle را افزایش و برای کاهش سرعت، این مقدار را کاهش می‌دهیم. مطابق (شکل ۱۱-۷)

² Radio Control

پردازنده روی لبه‌های بالا و پایین رونده و محاسبه زمان بین این وقفه‌ها توسط تایمر موجود در پردازنده، می‌توان اطلاعات این سیگنال‌ها را استخراج کرد.

آخرین سیستم ارتباطی میان ایستگاه زمینی و پرنده، مربوط به یک رابط نرم‌افزاری بین رایانه ایستگاه زمینی و پردازنده سطح بالای پرنده است. به وسیله این رابط نرم‌افزاری می‌توان نحوه اجرای برنامه در پردازنده سطح بالا را کنترل و یا در صورت نیاز، تغییراتی در برنامه آن اعمال کرد. این ارتباط از طریق شبکه بی‌سیم Wi-Fi برقرار می‌گردد.

۷-۳- نتیجه‌گیری

پس از انتخاب سخت افزار مورد نیاز و راه اندازی کامل آن، خواندن سنسورها توسط تیم دانشجویی سیستم‌های چند وسیله‌ای انجام شده و اطلاعات آن، در اختیار کنترل‌کننده قرار می‌گیرد. لذا می‌تواند کار خود را شروع کند. در فصل بعد به روند پیشرفت عملی پردازش تصویر پرداخته می‌شود.

فصل هشتم

پیاده سازی عملی

پیاده سازی عملی

در پایان گزارش‌های علمی و فنی، لازم است که جمع‌بندی یا نتیجه‌گیری نهایی ارائه شود. در این موارد می‌توان آخرین فصل پایان نامه که پیش از مراجع قرار می‌گیرد را به این امر اختصاص داد.

۱-۸-۱- نرم افزار

همانطور که اشاره شده بود، در این پروژه، از رزبری پای به عنوان پردازنده استفاده شده است. پس از نصب سیستم عامل "Raspbian GNU/Linux 8 (jessie)"، پردازنده آمادگی نصب برنامه‌های مورد نیاز را خواهد داشت. این سیستم عامل، یک سیستم عامل مبتنی بر Debian (سیستم عاملی رایگان و قدرتمند) است که با توجه به سخت افزار پردازنده‌ی رزبری پای، بهینه‌سازی شده است. سیستم عاملی که مجموعه‌ای از برنامه‌ها و نرم افزارهای پایه‌ی قابل اجرا در رزبری پای می‌باشد. که شامل ۳۵ هزار بسته^۱ و نرم افزار است که قابلیت نصب آسان را دارا هستند. علت وجود این تعداد بسته نیز، به صورت اختصاصی برای استفاده در رزبری پای بهینه‌سازی شده اند. با این حال، این سیستم عامل همچنان در دست توسعه و پیشرفت است تا بتواند به بیشترین تعداد بسته‌های موجود در Debian دسترسی پیدا کند. [36] [37]

۱-۸-۲- پردازش تصویر

برای انجام این پروژه به علت محدود بودن سرعت پردازنده مورد استفاده و نویز محیط (نور و اشیا) نیازمند استفاده یکی از موارد بالا هستیم که حجم پردازش کمتری نیاز داشته باشد و نسبت به نویز محیط مقاوم‌تر باشد. [5]

میان الگوهای مطرح شده، تحقیقات صورت گرفته، نشان می‌دهد الگوهای واقعیت افزوده بهترین گزینه هستند. در ذیل به شرح دلایل آن پرداخته می‌شود:

^۱ Package

۱- این مارکرها برای برنامه‌های موبایل‌های هوشمند ساخته شده‌اند و در نتیجه با پردازنده‌های به نسبت ضعیف موبایل قابل تشخیص و استفاده‌اند.

۲- برای کار با این مارکرها کتابخانه‌ای مبتنی بر OpenCV منتشر شده‌است که با الگوریتم‌هایی بهینه قابلیت تشخیص این مارکرها را دارد. نکته قابل توجه این است که این کتابخانه فاصله دوربین با مارکر و همچنین زاویه با صفحه آن را در سه محور مختصات دکارتی می‌دهد. در ادامه به بررسی کتابخانه مورد استفاده و نحوه‌ی کار این الگوها می‌پردازیم.

۱-۸-۱-۲- کتابخانه‌ی ^۱OpenCV

یکی از پرکاربردترین کتابخانه‌های مربوط به پردازش تصویر، کتابخانه‌ی OpenCV است که برای نصب آن، مراحل مربوطه طی شده و پس از تکمیل نصب، آمادگی برای اتصال دوربین، خواندن تصویر و پردازش آن وجود خواهد داشت.

این کتابخانه، مجموعه‌ای از کتابخانه‌های برنامه نویسی، پردازش تصویر و یادگیری ماشین^۲ است. این مجموعه، بیشتر بر پردازش تصویر بلاذرنگ^۳ تمرکز دارد و کتابخانه‌ای چند سکویی است که توسط سیستم عامل‌های ویندوز، لینکوس، مک او اس، ای او اس و اندروید پشتیبانی می‌گردد. همچنین دارای رابط برنامه نویسی به زبان‌های C، C++, پایتون، جاوا و متلب می‌باشد.

از قابلیت‌های این کتابخانه می‌توان فیلتر تصویر، سیستم تشخیص چهره، تشخیص حرکت، تعامل انسان و رایانه، رباتیک موبایل، درک حرکت، شناسایی شیء، ردیابی حرکت، واقعیت افزوده و قابلیت‌های متنوع دیگری را نام برد. [38]

با استفاده از این کتابخانه، فیلترها و آزمایش‌های اولیه‌ای انجام شده که به شرح زیر است :

¹¹ Open Computer Vision Library

² Machine Learning

³ Real Time (RL)

۱-۲-۸-۱- تشخیص لبه‌های تصویر

در این قسمت، با استفاده از فیلتر Canny Edge Detector لبه‌های تصویر تشخیص داده می‌شوند (شکل ۱-۸). بدین صورت که ابتدا یک لایه فیلتر Gaussian Blur تصویر به مقدار اندکی تار می‌شود تا نویز تصویر، اندکی گرفته شود. سپس برای فیلتر یاد شده به صورت خودکار از کتابخانه OpenCV صدا زده شده و لبه‌های تصویر را پیدا می‌کند.



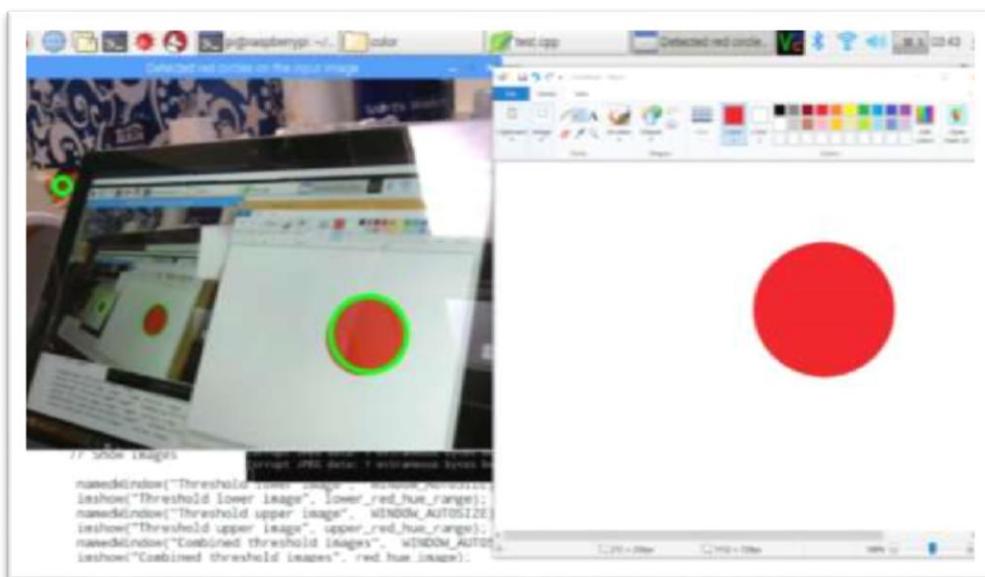
شکل ۱-۸ یافتن لبه‌های تصویر با استفاده از فیلتر Canny Edge Detector

۱-۲-۸-۲- تشخیص دواير قرمز رنگ [39]

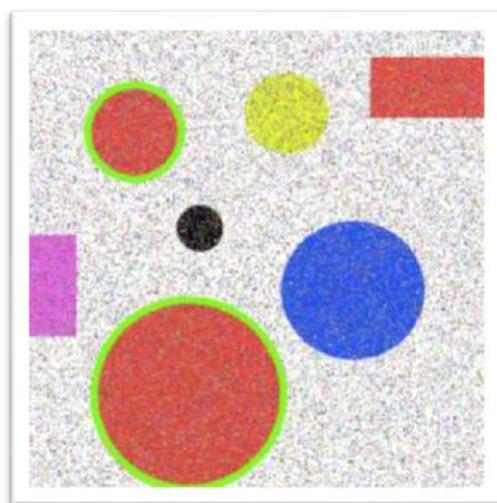
برای این کار، ابتدا تصویر گرفته شده توسط دوربین، توسط فیلتر Median Blur خالی از نویز می‌شود. سپس تصویر از فرمت^۱ BGR به فرمت HSV تبدیل می‌گردد. چرا که با استفاده از این فرمت، می‌توان یک رنگ مشخص، (در اینجا قرمز) را تنها با استفاده از یک مقدار، شناسایی نمود. در این فرمت، رنگ

^۱ در OpenCV به جای فرمت RGB از فرمت BGR استفاده می‌شود.

قرمز در محدوده‌ای بین ۰ تا ۱۶۰ و ۱۸۰ قرار دارد. بدین ترتیب، سایر رنگ‌ها از تصویر حذف شده و تنها رنگ قرمز باقی می‌ماند. حال نوبت به تشخیص دوایر می‌رسد. برای این منظور نیز در کتابخانه OpenCV دستوری برای تشخیص دایره وجود دارد که کار را برای کاربر ساده می‌نماید. نتیجه‌ی این برنامه در (شکل ۳-۸) و (شکل ۲-۸) مشاهده می‌شود.



شکل ۲-۸ تشخیص دوایر قرمز رنگ



شکل ۳-۸ یافتن دوایر قرمز رنگ در تصویر نویزی

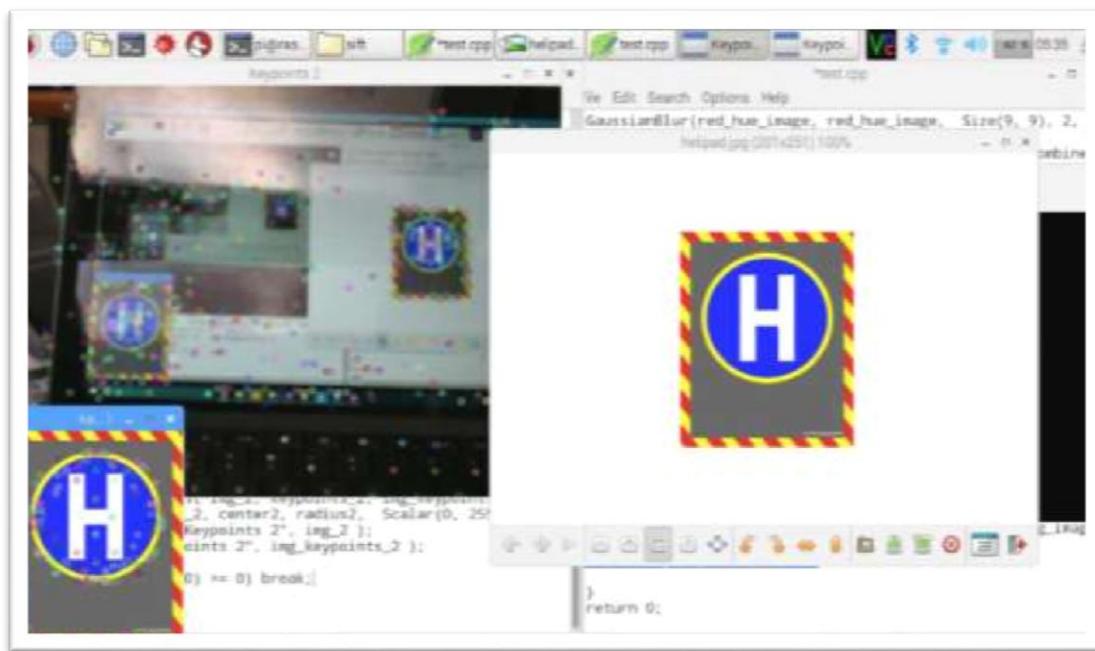
۳-۲-۱-۸- جست و جو و تشخیص یک عکس، در تصویر اصلی

در این قسمت، فرض می‌شود که برای فرود، علامتی مخصوص فرود به شکل (شکل ۴-۸) قرار گرفته است. برای تشخیص چنین علامتی در تصویر برداشته شده توسط دوربین، روش‌های متعددی وجود دارد. یکی از این روش‌ها SIFT است که با استفاده از این روش، نقاطی به عنوان نقاط کلیدی^۱ از تصویر مرجع انتخاب شده و در تصویر دوربین مورد جستجو قرار می‌گیرند. پس از پیاده‌سازی این دستور در کتابخانه، نتیجه در (شکل ۵-۸) مشاهده می‌گردد. سایر روش‌ها نیز مورد مطالعه قرار گرفته اما به دلیل سرعت پایین، از امتحان آن‌ها صرف نظر شد.



شکل ۴-۸ علامت فرود

¹ Keypoints



شکل ۵-۸ تشخیص نقاط کلیدی تصویر مرجع در تصویر دوربین

آخرین روش، به نظر روش بسیار مناسبی برای پردازش تصویر محل فرود می‌آید. اما به علت سرعت پایین پردازش این تصویر، به سراغ کتابخانه‌ی دیگری رفته و نتایج آن را نیز مورد بررسی قرار می‌دهیم.

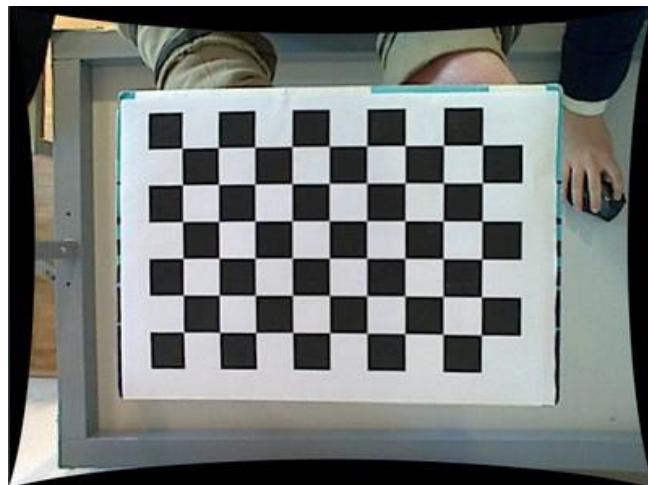
Aruco -۱-۸-۳

یک کتابخانه برای کار با بارکدهای مذکور یا الگوهاست، که بر پایه کتابخانه OpenCV نوشته شده است. از جمله ویژگی‌های این کتابخانه به موارد زیر می‌توان اشاره کرد:

- شناسایی مارکر با کد C++
- شناسایی مجموعه‌ای از چندین الگو به صورت همزمان
- قابلیت شناسایی بیش از ۲۴ نوع الگو

سريع، قابل اعتماد و قابل پیاده‌سازی بر روی چند نوع سیستم به علت متکی بودن بر OpenCV برای تکمیل سیستم موقعیت یابی فقط کافیست که دوربین کالیبره شده باشد. برای کالیبره کردن دوربین لازم است، حداقل ۵ تصویر از یک صفحه شطرنجی که از زاویه مقابل، بالا، پایین، چپ و راست گرفته شود که در این پروژه برای خروجی بهتر ۱۰ تصویر برای کالیبره کردن دوربین استفاده

شده است. در نهایت کتابخانه OpenCV با تطبیق این صفحه‌ها اعوجاج لنز دوربین را شناسایی و تأثیر آنرا در محاسبات خود جهت تشخیص فاصله‌ی نقاط، خنثی می‌کند.



شکل ۶-۸ نمونه یک صفحه کالیبره کننده دوربین [7]

فصل نهم

جمع‌بندی، نتیجه‌گیری و پیشنهادات

جمع‌بندی و نتیجه‌گیری

در جمع‌بندی، برای فرود خودکار یک روبات پهپاد چهار موتوره، ابتدا مدل‌سازی دینامیکی با استفاده از پارامترهای جرم روبات، طول بازوی آن، ممان اینرسی آن در سه محور و سایر پارامترها انجام شده و معادلات حالت نوشته شدند. با خطی‌سازی معادلات حالت با استفاده از ماتریس ژاکوبین، معادلات غیرخطی، به معادلات قابل کنترل در حوزه کنترل سیستم‌های خطی تبدیل شدند. همچنین با توجه به معادلات به دست آمده، کنترل‌کننده PID در ۲ محور طول و عرض، برای کنترل‌کننده موقعیت طراحی شد که خروجی آن‌ها با استفاده از روابط مربوطه، تبدیل به مقادیر رول و پیچ و یاو گردید که برای این سه مقدار و برای موقعیت ارتفاع روبات نیز، کنترل‌کننده PID جدا گانه طراحی شد.

در ادامه، مدل دینامیکی خطی و غیرخطی، در محیط سیمولینک نرم افزار متلب شبیه‌سازی شده و به کنترل‌کننده‌های طراحی شده متصل گشتند. برای کنترل رفتار روبات در حالت فرود نیز، یک کنترل‌کننده حلقه باز برای کنترل ارتفاع و دو کنترل‌کننده PI برای کنترل موقعیت طول و عرض مطلوب طراحی و پیاده‌سازی شدند. (ضریب اننتگرال گیر به علت وجود خطای حالت دائم افزوده شد.)

مرحله‌ی بعد ساخت روبات و طراحی اجزای مورد نیاز آن بود که توسط تیم روبات‌های چند وسیله‌ای طراحی و ساخته شد. پردازنده‌ی مربوط به انجام عملیات پردازش تصویر، خواندن سنسورها و کنترل حرکت موتورها نیز یک رزبری پای با سیستم عامل جسی استفاده شده‌است.

برای تشخیص تصویر نیز، از ۱۳ الگوی ماتریس داده در مجاورت ۱۲ خانه‌ی سفید، که توسط دوربین مخصوص رزبری پای خوانده می‌شود استفاده شد که با استفاده از الگوریتم‌های پردازش تصویر، با تشخیص محل فرود، فرمان فرود را ارسال کرده، سپس خطای موقعیت محاسبه شده و به کنترل‌کننده‌ی رفتار فرستاده می‌شود و با استفاده از کنترلر، روبات با خطای نهایتاً ۱۵ سانتی متر، در ارتفاع ۱۵ سانتی متری محل مورد نظر قرار گرفته و با توجه به ارتفاع ۱۲ سانتی متری بدنه‌ی روبات، خاموش کردن موتورها، فرود پایداری خواهد داشت. خطای مذکور نیز، توسط برنامه نویس انتخاب شده و به صورت دستی قابل کاهش است.

پیشنهادات

پس از پیاده‌سازی پروژه و با بررسی نقاط قوت و ضعف آن، پیشنهاداتی برای بهبود عملکرد پروژه مورد نظر است:

۱-۹-۱- افزودن کنترل کننده یا و

در این پروژه، فرض بر این گرفته شده که جهت فرود اهمیتی ندارد و روبات از هر زاویه‌ای به محل فرود برسد، فقط فرود می‌آید. اما در بسیاری از کاربردها، برای مثال شرایطی که محل فرود به عنوان شارژر روبات عمل می‌کند، جهت روبات، از اهمیت بالایی برخوردار است. لذا روبات پس از کاهش ارتفاع و پیش از خاموش کردن موتورها، بایستی جهت خود را بسنجد و به میزان مطلوب بچرخد و در نهایت در جهت صحیح موتورها را خاموش نماید.

۱-۹-۲- استفاده از پردازش‌گرهای مجزا برای پردازش تصویر و کنترل

پردازش تصویر، یکی از پردازش‌های پرمصرف در بین عملیات‌های پردازش تصویر در هر پروژه‌ای است. لذا در هنگام فعال سازی عملیات فرود، بخش عمداتی از عملکرد پردازش‌گر صرف انجام پردازش تصویر شده و مانع از عملکرد دقیق کنترل کننده‌ها می‌شود. لذا پیشنهاد می‌گردد تا به جای استفاده از یک پردازنده برای هر دو منظور، پردازنده‌ی قوی‌تر یا همان رزبی پای، تنها برای عملیات پردازش تصویر استفاده شود و برای انجام محاسبات کنترلی و پردازش حرکت موتورها، از یک پردازنده‌ی دیگر استفاده شود که در این پروژه با توجه به محدودیت وزن، ابعاد و هزینه، پردازنده دوم باید ضعیفتر باشد. چرا که عملیات پردازش مربوط به موتور سبک‌تر از پردازش تصویر است. برای مثال استفاده از یک بورد آردوینو و یا طراحی یک بورد آرم^۱ به صورت جداگانه، می‌تواند گزینه‌ی مناسبی باشد.

¹ ARM

۱-۹-۳- شبیه‌سازی مدل دینامیکی زمین [40]

در این پژوهه، تنها به مدل دینامیکی کوادکوپتر و اعمال نویز به آن پرداخته شد؛ در حالی که یک روبات پرنده در زمان فرود، به شدت تحت تاثیر مدل دینامیکی زمین قرار می‌گیرد. این مدل، حاصل از اثر عکس‌العمل نیروی مخل‌ها و بدنی روبات است که با نزدیک شدن به سطح زمین، از سطح زمین به سمت روبات بازتاب می‌شوند و باعث ایجاد نیروهای نا خواسته می‌شوند. بنابراین پیشنهاد می‌شود در مراحل بعدی، مدل دینامیکی زمین و حتی اشیاء موجود در محیط نیز طراحی شده و به شبیه‌سازی و پیاده‌سازی اضافه گردند تا کنترل به نحو بهینه تری انجام پذیرد.

البته در مورد کوادکوپترها، به علت سطح مقطع کم به نسبت سایر پرنده‌ها، مدل دینامیکی زمین در هنگام فرود تاثیر بسیار کمتری دارد. در واقع مدل زمین، برای پرنده‌های بالدار و در هنگام بلند شدن مورد محاسبه قرار می‌گیرد که به دلیل وجود بال، سطح مقطع زیادی با زمین دارند اما از تاثیر این نیرو در برابر یک کوادراتور در حین فرود می‌توان چشم پوشی کرد.

۱-۹-۴- انتقال به محیط فضای باز و استفاده از سنسور باد

از آنجایی که بیشتر کاربردهای روبات پرنده در فضای باز قرار دارد و تعریف پژوهه در محیط بسته، بیشتر جنبه‌ی پژوهشی و تحقیقاتی دارد، پیشنهاد می‌شود پژوهه به فضای باز منتقل شود. با توجه به اینکه در طراحی کنترل‌کننده، در همه‌ی محورها نویز اعمال شده، لذا در واقع در شبیه‌سازی، وجود باد نیز پیش‌بینی شده و می‌توان روبات را به راحتی و با انجام بهینه‌سازی‌های اندکی، به فضای باز انتقال داد.

حتی پیشنهاد می‌شود برای تشخیص جهت باد و تعریف کنترل‌کننده‌ی مناسب‌تر با استفاده از آگاهی از جهت باد، از یک سنسور تشخیص باد بر روی روبات استفاده شود.

۱-۹-۵- کنترل و شناسایی فازی

پیشنهاد دیگری که مطرح می‌شود، استفاده از یک شناساگر فازی، برای شناسایی مدل دینامیکی روبات و یا استفاده از کنترلر فازی به جای کنترلر PID است. حتی می‌توان از هر دو روش مدل‌سازی و کنترل‌کننده فازی استفاده نمود که البته با پردازشی سخت افزار روبات را بسیار سنگین می‌نمایند و در آن صورت نیازمند پردازش‌گر قوی تری خواهیم بود.

منابع و مراجع

- [1] "Quadcopter," Wikipedia, the free encyclopedia, [Online]. Available: <https://en.wikipedia.org/wiki/Quadcopter>. [Accessed 30 March 2017].
- [2] R. Chaudhary, R. M. Singari, R. C. Singh and M. Garg, "Quad-Rotor Unmanned Arial Vehicle – A Review," in *International Conference of Advance Research and Innovation (ICARI-2015)*, 2015.
- [3] T. D. SUYADHI, "OS4 QUADROTOR - EPFL," Robotics University, 10 January 2014. [Online]. Available: <http://www.robots-university.com/2014/01/os4-quadrrotor-epfl.html>. [Accessed 30 August 2017].
- [4] E. L'Eplattenier, "Real Estate Drone Photography & Video Guide 2018," Fitsmallbusiness, 26 October 2017. [Online]. Available: <https://fitsmallbusiness.com/real-estate-drone-photography/>. [Accessed 31 October 2017].
- [5] S. Bouabdallah, Design and control of quadrotors with application to autonomous flying, EPFL University, PHD Thesis, 2007.
- [6] E. Bara and A. Yesildierk, "Robust Nonlinear Composite Adaptive Control of Quadrotor.,," *International Journal of Digital Information and Wireless Communications (IJDWC) 4.2* , pp. 45 - 57, 2014.
- [7] ا. علیخانی, "کنترل سیستم چندعامله پهپادهای چهارموتوره," دانشگاه صنعتی امیرکبیر (پلی تکنیک تهران), تهران, ۱۳۹۳.
- [8] S. BOUABDALLAH, "design and control of quadrotors with application to autonomous flying," ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE, Lausanne, 2007.

- [9] "Lagrangian mechanics," Wikipedia, the free encyclopedia, [Online]. Available: https://en.wikipedia.org/wiki/Lagrangian_mechanics. [Accessed 17 October 2017].
- [10] "Arduino," Surplustronics, [Online]. Available: <https://www.surplustronics.co.nz/categories/652-arduino>. [Accessed 27 October 2017].
- [11] "Global Positioning System," Wikipedia, the free encyclopedia, [Online]. Available: https://en.wikipedia.org/wiki/Global_Positioning_System. [Accessed 24 October 2017].
- [12] "Erle-Brain 3," Erle Robotics, [Online]. Available: <http://erlerobotics.com/blog/erle-brain-3/>. [Accessed 16 Sep 2017].
- [13] "Proximity sensor," Wikipedia, the free encyclopedia, [Online]. Available: https://en.wikipedia.org/wiki/Proximity_sensor. [Accessed 24 October 2017].
- [14] "Photodetector," Wikipedia, the free encyclopedia, [Online]. Available: <https://en.wikipedia.org/wiki/Photodetector>. [Accessed 24 October 2017].
- [15] "Laser rangefinder," Wikipedia, the free encyclopedia, [Online]. Available: https://en.wikipedia.org/wiki/Laser_rangefinder. [Accessed 24 October 2017].
- [16] "Sonar," Wikipedia, the free encyclopedia, [Online]. Available: <https://en.wikipedia.org/wiki/Sonar>. [Accessed 24 October 2017].
- [17] "SRF02 Ultrasonic range finder - Robot Electronics," robot-electronics, [Online]. Available: <https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=0ahUKEwiejdKf6pvXAhWKKVAKHb40A7IQFggsMAA&url=https%3A%2F%2Fwww.robot-electronics.co.uk%2Fhtm%2Fsrf02tech.htm&usg=AOvVaw3q2tiCdj-hjQQK8yNTSdJF>. [Accessed 27 October 2017].
- [18] "Logitech HD Webcam (C270)," bestbuy, [Online]. Available: <https://www.bestbuy.ca/en-ca/product/logitech-logitech-hd-webcam-c270-960-000621/10146689.aspx>. [Accessed 27 October 2017].

- [19] "Getting started with picamera," Raspberry Pi, [Online]. Available: <https://www.raspberrypi.org/learning/getting-started-with-picamera/worksheet/>. [Accessed 28 August 2017].
- [20] "Raspberry Pi Camera Module," sparkfun, [Online]. Available: <https://www.sparkfun.com/products/retired/11868>. [Accessed 27 October 2017].
- [21] "Kinect," Wikipedia, [Online]. Available: <https://en.wikipedia.org/wiki/Kinect>. [Accessed 27 October 2017].
- [22] K. Rajaratnam, "Vision Based Autonomous Landing of an UAV with Wireless Power Transfer System," School of Engineering and Technology Thailand, 2015.
- [23] "Hokuyo URG-04LX-UG01 Scanning Laser Rangefinder," RobotShop, [Online]. Available: www.robotshop.com/en/hokuyo-urg-04lx-ug01-scanning-laser-rangefinder.html. [Accessed 27 October 2017].
- [24] "Vision-based autonomous landing for rotorcraft unmanned aerial vehicle".
- [25] "QR code," Wikipedia, the free encyclopedia, [Online]. Available: https://en.wikipedia.org/wiki/QR_code. [Accessed 27 October 2017].
- [26] "Data Matrix," Wikipedia, the free encyclopedia, [Online]. Available: https://en.wikipedia.org/wiki/Data_Matrix. [Accessed 27 October 2017].
- [27] "Detection of ArUco Markers," OpenCV, [Online]. Available: https://docs.opencv.org/3.2.0/d5/dae/tutorial_aruco_detection.html. [Accessed 27 October 2017].
- [28] J. Meyer, "A Low Cost, Vision Based Micro Helicopter System for Education and Control Experiments," UNIVERSITY OF OKLAHOMA GRADUATE COLLEGE, Norman, Oklahoma, 2014.
- [29] B. Gallenstein and Y. Rosenblum, "Autonomous Fixed-Point Landing for Quadrotor Aerial Vehicles," WORCESTER POLYTECHNIC INSTITUTE, 2015.
- [30] "Raspberry Pi," Wikipedia, the free encyclopedia, [Online]. Available: https://en.wikipedia.org/wiki/Raspberry_Pi.

- [31] inigomuguruza, "Erle Robotics WIKI," ROS Software Maintainer, 25 Apr 2016. [Online]. Available: <http://wiki.ros.org/Robots/Erle-brain>. [Accessed 16 Sept 2017].
- [32] "Raspberry Pi 3 Grand Opening," 1 March 2016. [Online]. Available: <http://tech.scargill.net/raspberry-pi-3-grand-opening/>. [Accessed 19 April 2017].
- [33] "Common Power Module — Copter documentation," ArduPilot, [Online]. Available: ardupilot.org/copter/docs/common-3dr-power-module.html. [Accessed 27 October 2017].
- [34] "MT2216," [Online]. Available: http://www.rctigermotor.com/html/2013/Professional_0912/45.html. [Accessed 20 January 2017].
- [35] "FLAME 25A," [Online]. Available: http://www.rctigermotor.com/html/2016/esc_0621/314.html. [Accessed 20 January 2017].
- [36] "Debian The universal operating system," Debian, [Online]. Available: <https://www.debian.org/>. [Accessed 28 August 2017].
- [37] "Welcome to Raspbian," Raspbian, [Online]. Available: <https://www.raspbian.org/>. [Accessed 28 August 2017].
- [38] "OpenCV," OpenCV, [Online]. Available: <http://opencv.org/>. [Accessed 28 August 2017].
- [39] Sol, "Detect red circles in an image using OpenCV," Solarian Programmer , 8 May 2015. [Online]. Available: <https://solarianprogrammer.com/2015/05/08/detect-red-circles-image-using-opencv/>. [Accessed 28 August 2017].
- [40] E. Cui and X. Zhang, "Ground Effect Aerodynamics," in *Encyclopedia of Aerospace Engineering*, Beijing, China, 2010.
- [41] "Carbon fibers," [Online]. Available: https://en.wikipedia.org/wiki/Carbon_fibers. [Accessed 19 April 2017].

- [42] "Railroad Spike Materials Overview - Carbon Steel And Structural Steel," 2016. [Online]. Available: <http://www.railway-fasteners.com/news/railroad-spikes-materials-carbon-steel-structural-steel.html>. [Accessed 19 April 2017].
- [43] "POLYCARBONATE 250 MINI QUAD FRAME – CHANEY AEROSPACE," [Online]. Available: <https://oscarliang.com/polycarbonate-mini-quad-frame-chaney-aerospace/>. [Accessed 19 April 2017].
- [44] "Balsa wood quadcopter: Part 2," [Online]. Available: <http://www.quadcopterprojects.com/balsa-wood-quadcopter-part-2/>. [Accessed 19 April 2017].

پیوست‌ها

پیوست ۱ مقایسه مواد مختلف برای ساخت بدن و اجزای روبات

فیبر کربن:

فیبر کربن یکی از مواد ایده آل برای ساختن فریم کوادراتور است. این ماده سبک، سخت و بسیار محکم می‌باشد ولی از نقاط ضعف این ماده، می‌توان به قیمت بالای آن اشاره کرد. همچنانی به دلیل سمی‌بودن، برای تنفس خطر آفرین است. فیبر کربن یک ماده مرکب، متشکل از فیبرهای بسیار بلند و نازک، کربن و اپوکسی می‌باشد. این فیبرهای کربن بسیار محکم و بسیار سخت اند، اما فقط در یک جهت قرار دارند. اپوکسی یک درجه نرم‌تر و ضعیفتر از فیبر بوده اما برای نگه داشتن فیبرها در کنار یک دیگر مورد نیاز است. این بدان معنی است که یک بخش فیبر کربن که در جهت فیبرها بوده، بسیار مستحکم و جهت‌های دیگر نسبتاً ضعیف اند.

صفحات مسطحی که با استفاده از لایه‌های فابریک به گونه‌ای ساخته شده‌اند که در هر جهت، سختی و مستحکام برابر داشته، در دسترس عموم قرار دارند. این تکنیک مزیت‌های استفاده از فیبر کربن را کاهش می‌دهد اما همچنان حتی در فرم عمومی‌خود نیز بسیار محکم و سخت هستند. فیبر کربن رسانا بوده و به همین دلیل نیازمند مراقبت در هنگام تماس با قطعات الکترونیکی یا مهم‌تر از آن پلیمرلیتیم می‌باشد.

[41]



شکل پ - ۱ دم یک هواپیمای کنترلی [41]

آلومینیوم:

از زمان جنگ جهانی دوم، آلومینیوم در ساخت هواپیما بکار میرفته است و امروزه نیز به کار می‌رود. در هواپیماهای بسیار پیشرفته امروزی نیز مانند بوینگ ۷۸۷ یا V-22 از مقدار قابل توجهی آلومینیوم استفاده می‌شود. همچنین به دلیل در دسترس و آماده بودن و هزینه پایین آن، برای ساخت کوادها بسیار عالی می‌باشد. به علت ساختار همگن آن، به راحتی توسط ابزارآلات مختلف می‌تواند شکل‌های مختلفی به خود بگیرد، ویژگی که در موادی مانند فیبرکربن یا فایبرگلاس وجود ندارد.



شکل پ - ۲ بدنه آلومینیومی یک کواد

تنوع زیادی در آلیاژهای مختلف آلومینیوم وجود دارد و با وجود این که همگی چگالی و سختی مشابهی دارند، استحکام آن‌ها می‌تواند بسیار متفاوت باشد. با اینکه 6061° درجه هوایپیما^۷ نامیده می‌شود، در هوایپیماهای مدرن استفاده نمی‌شود زیرا استحکام آن 30° پوند بر اینچ می‌باشد. در هوایپیماهای مدرن از آلیاژهایی مانند 2024° ، 7050° ، یا 7075° با استحکام 60° پوند بر اینچ استفاده می‌شود. آلومینیوم همچنین رسانای خوبی می‌باشد و به همین دلیل نیازمند حفاظت می‌باشد.

فایبرگلاس:

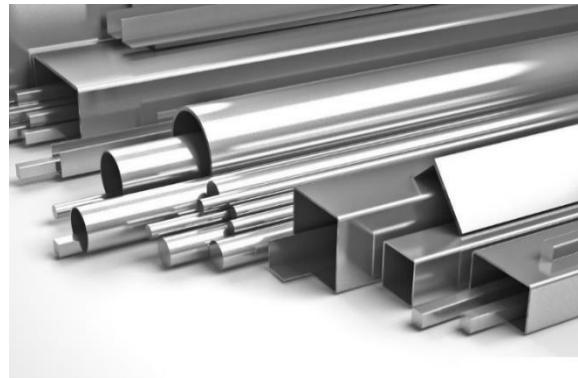
فایبرگلاس برادر کوچکتر فیبر کربن می‌باشد که تقریباً به همان گونه از فیبرهای بلند گلاس که با اپوکسی به یک دیگر بسته شده‌اند، تولید می‌شود. با این حال، فایبرگلاس سنگین‌تر، نرم‌تر، و ضعیف‌تر از فیبرکربن می‌باشد اما راحت‌تر شکل می‌پذیرد، مراقبت کمتری هنگام بریدن آن لازم است و ارزان‌تر است. هرچند در سال‌های گذشته به علت مشکلات در تعمیر کردن آن شهرت بدی پیدا نموده‌است. با این حال برای ساخت کوادها، ماده عالی می‌باشد. تنوع زیادی برای این ماده وجود دارد اما G10 Garolite/FR4 یک انتخاب عالی می‌باشد که در دسترس عام و تقریباً ارزان می‌باشد.



شکل پ - ۳ یک کواد از جنس فایبرگلاس

استیل:

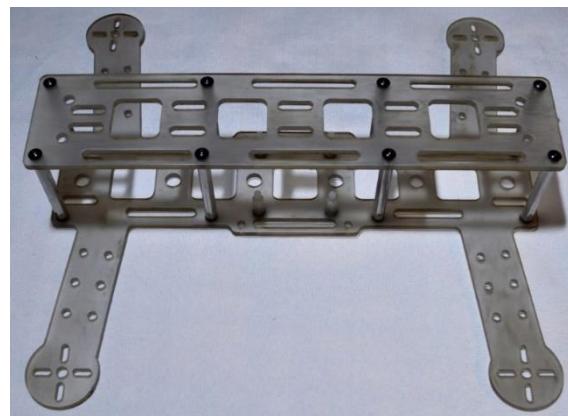
استیل یکی از محکم‌ترین فلزات بکار رفته در صنعت هوافضا می‌باشد و با وجود رشد روزافزون شهرت تیتانیوم و فیبر کردن، همچنان تعداد زیادی از قطعات هستند که می‌بایست با استیل ساخته شوند. با این حال، استیل برای ساخت بدنه کواد، به استثنای بسته‌ها که استحکام زیادی نیاز دارد یا در حالتی که نسخه پلاستیکی یا آلومینیومی آن موجود نیست، بسیار سنگین می‌باشد.^[42]



شکل پ - ۴ استیل در اشکال مختلف [42]

پلی کربنات:

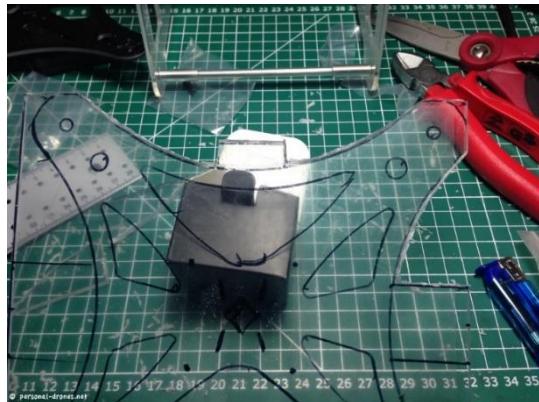
پلی کربنات یا لاسان در دسترس عام و ارزان می‌باشد اما انتخاب ضعیفی برای بدنه یک کواد می‌باشد. پلی کربنات سنگین می‌باشد و خیلی سخت نمی‌باشد هرچند استحکام مناسبی دارد. شکل دهی آن آسان و نارسانا می‌باشد به همین دلیل برای ساخت صفحاتی که قطعات الکترونیکی بر روی آن قرار بگیرد مناسب می‌باشد. اما با وجود ممکن بودن، برای ساخت یک بازوی کمکی موتور نیاز به سطح مقطع بزرگی می‌باشد که به این معنی است که بخش زیادی از نیروی موتور از بین رفته و همچنین وزن کواد نیز افزایش پیدا می‌کند. با این وجود، این ماده شفاف بوده و در مقابل شکستن مقاوم می‌باشد که به این معنی است که برای سپر قطعات الکترونیکی بودن مناسب است. [43]



شکل پ - ۵ پلی کربنات شفاف در ساخت بدنه کواد [43]

آکریلیک:

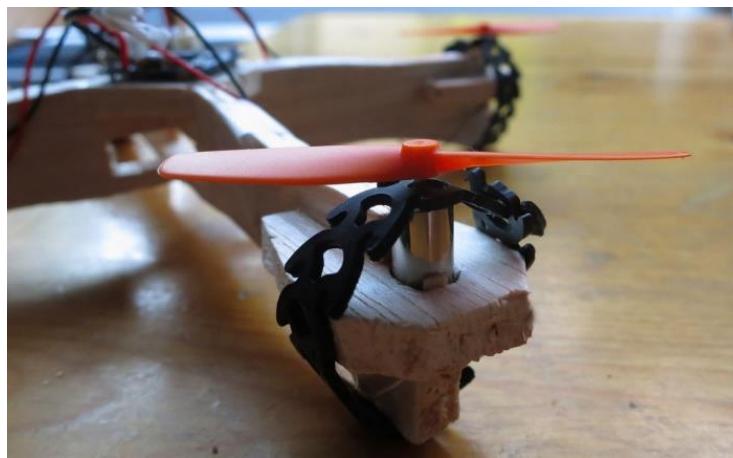
آکریلیک یا پلکسی گلاس یک انتخاب ضعیف است. این ماده شکننده و نرمتر از پلی کربنات می‌باشد و می‌باید از انتخاب آن اجتناب نمود.



شکل پ - ۶ پلکسی گلاس در کواد

چوب بالسا:

این چوب همیشه در هواپیماهای کنترلی به دلیل مناسبی استفاده می‌شده است. این ماده ارزان، بسیار سبک، سخت و محکم می‌باشد. همچنین به راحتی در دسترس بوده و به آسانی شکل می‌بذیرد. این به این معنی است که با هزینه پایین قطعات از این جنس را می‌توان تعویض نمود. با این حال به علت چگالی پایین آن، سطح مقطع ضخیمی از آن برای بازوهایی که سختی مناسبی داشته باشند لازم می‌باشد که مشابه پلی کربنات مانع بخشی از نیروی موتور می‌شود. [44]



شکل پ - ۷ چوب بالسا و سطح مقطع آن در ساخت بدنه کواد

جدول پ - ۱ مقایسه خواص مواد مختلف بر اساس چگالی (پوند بر اینچ)، مشخصه یانگ که نشان دهنده

میزان سختی یا الاستیسیته ماده می باشد.

استحکام	مشخصه یانگ	چگالی	ماده
۱۲۰	۹/۳	۰/۰۵	فیبرکربن
۱۳۰	۱۶/۸۲	۰/۱۶	تیتانیوم
۷۵-۱۵	۱۰/۳	۰/۱۰	آلومینیوم
۵۰-۱۵	۲/۷	۰/۰۷	فایبرگلاس
۲۹۰-۳۰	۲۹	۰/۲۹	استیل
۱۶-۸	۰/۷۵	۰/۰۵	پلیکربنات
۱۱-۸	۰/۳۸	۰/۰۴	آکریلیک
۴/۶-۱	۰/۱۶-۰/۹	۰/۰۰۸۱-۰/۰۰۲۷	چوب بالسا