ساختار و شبیه­سازی ربات پرنده کوادکوپتر نگهبان

محمد مهدی اسماعیلی آبدر1، محمد علائی2

|  |  |
| --- | --- |
| 1\*کارشناسی مهندسی کامپیوتر، دانشکده فنی و مهندسی، گروه مهندسی کامپیوتر، دانشگاه ولی عصر (عج) رفسنجان | Mahdis377@gmail.com |
| 2دانشيار، دانشكده فني و مهندسي، گروه مهندسي کامپيوتر، دانشگاه ولي­عصر (عج) رفسنجان | alaeim@vru.ac.ir |
|  |  |

# چكيده

با توجه به گسترش استفاده از وسایل هوایی بدون سرنشین در کاربردهای مختلف و صنایع رباتیک، نیاز به قابلیت پرواز خودکار این گونه وسایل، بیش از پیش به وجود آمده است. در این پژوهش، شبیه­سازی و ساختار کنترل‌کننده‌های پیچیده و غیرخطی در راستای بهبود کیفیت مکانیکی و کامپیوتری ربات‌های پرنده با هدف ایجاد امنیت برای اشخاص،­ محافظت و تعقیب، بررسی شده است. این فرآیند برای پیاده‌سازی در سامانه‌های کنترل پرواز رایج همچون آردوپایلوت بهتر است. در این سامانه‌ها، بهره‌های سامانه کنترل یک بار توسط کاربر تنظیم می‌شوند و پس از آن در مأموریت‌های مختلف ثابت نگه داشته می‌شوند و بهینه‌سازی سامانه هدایت در ایستگاه زمینی انجام می‌شود و می‌تواند خود را با سامانه کنترل سازگار نماید و در این حالت برای اینکه ربات به درستی و با کیفیت بتواند شخص را رهگیری و دنبال کند و خطرات اطراف شخص را تشخیص دهد، نیاز به این است که برای هر کدام از واحدهای ادراکی و عملگرکدهایی نوشته شود و ارتباط مؤثری بین این کدها برقرار گردد تا بتوان با استفاده از پردازش محیط اطراف، خطرات را تشخیص و پیام نوع خطر را اعلام کند. سیستم‌عامل ربات‌ و دیگر ابزار­های متصل به­آن بستری قدرتمند و سریع را برای یکپارچه‌سازی کدهای مربوط به هر یک از واحدهای ربات می­باشد .

کليدواژه­ها: سیستم‌عامل ربات، کوادروتور، شبیه‌ساز گزبو، ربات پرنده، تکنولوژی، ربات، کنترل از راه دور.

**Structure and simulation of the guard quadcopter flying robot**

**Mohammad Mahdi Esmaeili\*, Mohammad Alaei2**

|  |  |
| --- | --- |
| 1\*Bachelor of Computer Engineering, Faculty of Engineering, Department of Computer Engineering, Vali-e-Asr University, Rafsanjan | Mahdis377@gmail.com |
| 2Associate Professor, Faculty of Engineering, Department of Computer Engineering, Vali-e-Asr University, Rafsanjan | alaeim@vru.ac.ir |
|  |  |

**Abstract**

Due to the expansion of the use of unmanned aerial vehicles in various applications and robotics industries, the need for automatic flight capabilities of such devices has become more and more. In this research, the simulation and structure of complex and nonlinear controllers in order to improve the mechanical and computer quality of flying robots with the aim of providing security for individuals, و protection and pursuit, has been investigated. This process is better for implementation in conventional flight control systems such as Ardopilot. In these systems, the benefits of the control system are adjusted once by the user and then kept constant in various missions, and the guidance system is optimized at the ground station and can adapt to the control system. To be able to track and follow the person correctly and with quality and identify the dangers around the person, it is necessary to write codes for each of the perceptual units and functions and to establish an effective connection between these codes in order to be able to use the processing of the environment. , Identify hazards and announce the type of hazard message. The robot operating system and other tools connected to it provide a powerful and fast platform for integrating the code associated with each of the robot units.

**Keywords**: Robot operating system, quadrotor, gazbo simulator, flying robot, technology, robot, remote control.

مقدمه

سیستم‌عامل ربات[[1]](#footnote-1) برای برقراری ارتباط بین وسایل جانبی مختلف و استفاده بهینه از حافظه و پردازنده کامپیوتر، به فایل‎‌های سیستمی نیاز دارد و سیستم‌عامل ربات را نمی­توان به طور صریح در یکی از دسته‌بندی‌های نرم‌افزاری قرار داد. لذا به خاطر نزدیکی به فایل‌های سیستمی و هسته سیستم‌عامل [[2]](#footnote-2)، آن را سیستم‌عامل[[3]](#footnote-3) می‌نامند، اما از یک نگاه دیگر به دلیل مستقل نبودن آن نمی‌توان آن را سیستم‌عامل نامید در نتیجه محیط سیستم‌عامل ربات برای این­که قابلیت برقراری ارتباط بین قسمت‌های مختلف سخت‌افزاری را دارد، در دسته میان‌افزارها[[4]](#footnote-4) قرار می­گیرد. کتابخانه‌ها و بسته‌های نرم‌افزاری متعدد که هر روز به تعداد آن‌ها افزوده می‌شود، موجب شده که سیستم‌عامل ربات به ابزاری کارآمد برای کارهای اتوماتیک تبدیل شود. در ابتدای شکل‌گیری سیستم‌عامل ربات به دلیل اینکه در کارهای رباتیکی نیاز به دسترسی به فایل‌های سیستمی بود و سیستم‌عامل ویندوز این دسترسی را پشتیبانی نمی­کرد، سیستم‌عامل­ ربات تحت لینوکس اجرا و این سیستم برخلاف ویندوز، سیستم‌عاملی متن‌باز[[5]](#footnote-5) بوده که کد منبع‌های آن در دسترس می­باشد و به سادگی می‌توان در فایل‌های سیستمی آن تغییر ایجاد کرد.

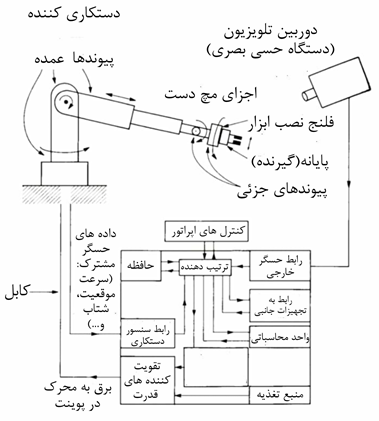
در جهان، در دهه‌ی گذشته، علاقه به ربات‌های پرنده کوچک، افزایش یافته است این ربات‌ها، از جمله کوادروتور ها[[6]](#footnote-6) می‌توانند با پتانسیل بالا در زمینه­ی عملیات جستجو و نجات در فضاهای محدود و از روی سطح­های مختلف پرواز سریعی به سمت هدف مورد نظر داشته و فورا فرود آید. این باعث شده است که کوادروتورهای کوچک به یک پلتفرم هوایی با پتانسیل بسیار بالا و جذاب برای برنامه‌های تحقیقاتی و تجاری تبدیل گردند. ربات‌های پرنده تجاری زیادی در محدوده‌ی پانصدگرم تا پنج­کیلوگرم، در بازار جهان موجود می‌باشند. اما در زمینه های امنیتی به خوبی روی آنها کار نشده است که قصد بررسی آنرا داریم. در حال حاضر پیشرفت‌هایی در زمینه‌های باتری‌های با توان بالا، موتورهای سبک، کم حجم و سنسورهای سیستم میکرو الکترومکانیکی[[7]](#footnote-7) صورت گرفته است. معمولا در محیط‌های محدود، ربات‌های پرنده بزرگ‌تر قابل استفاده نمی‌باشند. از کوادروتورهای کوچک می‌توان در مکان‌های خطرناک یا غیرقابل‌دسترس استفاده نمود و کاهش حجم ربات‌های پرنده چهار ملخه باعث افزایش قابل توجه‌ای در چابکی آن‌ها می‌شود که بتوان از آنها با کمک سیستم عامل ربات و پردازش های هوش مصنوعی خطرات را تشخیص داد و در سریعترین زمان به شخص هشدار لازم را بدهد. مقاله در شش بخش به جز مقدمه تنظیم گردیده، در بخش ابتدایی درباره­ی پیش زمینه و مباحث رباتیک مطالب آورده شده­اند. در بخش­ بعد به ساختار ربات پرنده و شیوه­های پیاده­سازی و در بخش های دیگر به معرفی طرح پرداخته­ایم. در ادامه شامل محاسبات، نتایج و عملیات های حاصل از شبیه­سازی پژوهش بیان شده است.

پیش زمینه تحقیق

ربات وسیله‌ای مکانیکی جهت انجام وظایف مختلف است. یک ماشین که می‌تواند برای عمل به دستورات مختلف برنامه‌ریزی و یک سری اعمال ویژه انجام دهد. مخصوصا آن دسته از کارها که فراتر از حد توانایی‌های طبیعی بشر باشند. این ماشین‌های مکانیکی برای بهتر به انجام رساندن اعمالی از قبیل احساس کردن، درک نمودن، جابجایی اشیا و اعمال تکراری شبیه جوشکاری تولید می‌شوند [۱۴]. الجزری (دانشمند مسلمان کردتبار ایرانی) در قرن ششم نخستین ربات قابل برنامه‌ریزی انسان­نما را در اواخر عمرش ساخت. به این علت او به عنوان پدر علم مهندسی رباتیک جهان شناخته می‌شود. سه شاخه اصلی الکترونیک (مغز ربات)، مکانیک (بدنه فیزیکی ربات)، نرم‌افزار (قوه تفکر و تصمیم گیری ربات) علم رباتیک را تشکیل می­دهند.

کوادرتور یا کوادکوپتر یک ربات پرنده چهارملخه[[8]](#footnote-8) است. پیکربندی آن به صورتی است که پرنده بتواند به‌راحتی و به‌طور مساوی در تمامی جهات حرکت کند. در سال‌های اخیر وسایل نقلیه هوایی بدون سرنشین در کاربردهای نظامی و غیرنظامی به دلیل دارا بودن قدرت مانور بالا، کاهش هزینه تولید و نگهداری، کاهش احتمال شناسایی توسط رادار، طولانی‌تر بودن مدت پرواز و کاهش خطر برای خدمه پرواز به‌خصوص در انجام مأموریت‌های نظامی در مقابل پرنده‌های دیگر از مزایای بیشتری برخوردارند. البته انگیزه‌های غیرنظامی هم برای توسعه این وسایل وجود دارد که برای آن می‌توان به عملیات امداد و نجات، نظارت از طریق هوا بر کنترل ترافیک شهری، مدیریت خطرات محیطی نظیر آتش‌سوزی در جنگل‌ها، اندازه‌گیری آلودگی به تشعشعات اتمی، نظارت بر عملیات احداث تأسیسات مانند سدها و خطوط انتقال و نیز در زمینه‌های کشاورزی، نقشه‌برداری هوایی و فیلم‌برداری اشاره نمود.

در هلی­کوپتر‌ها یک روتور[[9]](#footnote-9) اصلی تمام قدرت بالابرنده را تأمین و یک روتور دم نیز برای خنثی کردن گشتاور جانبی وجود دارد تا برای چرخاندن بدنه بالگرد در جهت مخالف روتور اصلی فعال باشد. بر خلاف هلی کوپتر، در یک کوادکوپتر از چهار ملخ برای تولید نیروی بالابرنده استفاده می‌شود و هر ملخ یک چهارم وزن را تحمل می‌کند و کنترل حرکت پرنده به وسیله تغییر دادن نیروی نسبی ناشی از هر ملخ به دست می‌آید. حرکت کوادکوپتر با تغییر سرعت یا به عبارت بهتر رانش‌های نسبی هر روتور کنترل می‌شود. این روتورها به طور معمول در یک مربع تراز می‌شوند. دو گوشه مستقر بر یک قطر مربع در جهت عقربه‌های ساعت و دو مورد دیگر در جهت خلاف آن می‌چرخند. برای اینکه کوادکوپتر بتواند ارتفاع خود را حفظ کند لازم است تا مجموع نیروی بالابرندگی چهار موتور پرنده ثابت باقی بماند. ارتفاع نیز به علت ثابت ماندن مجموع نیروی بالابرندگی موتورها تغییر نخواهد کرد. با افزایش سرعت چرخش تمام موتورها، کوادکوپتر در هوا بالا می‌رود. برای کاهش ارتفاع کافیست سرعت چرخش تمامی موتورها کم شود. حرکت کوادکوپتر به جهت‌های جلو، عقب، چپ و راست نیز با برهم زدن تعادل کوادکوپتر صورت می‌گیرد. با تغییر سرعت موتورهای هم جهت کوادکوپتر در هوا می‌چرخد، به این صورت که دو موتور هم جهت سرعت خود را کم می‌کنند و دو موتور هم جهت دیگر به همان میزان سرعت حرکت خود را افزایش می‌دهند تا بدون تغییر در ارتفاع، کوادکوپتر به دور خود بچرخد. کوادکوپترهای معمولی حداکثر هفت دقیقه زمان پرواز را ارائه می‌دهند، که ممکن است برای کسانی که چگونگی پرواز را یاد می‌گیرند خسته­کننده باشد. هواپیماهای بدون سرنشین درجه یک اغلب در مورد سرعت و دور پرواز و مدت زمان ماندن در هوا محدود هستند به طور معمول کواد کوپترهای سطح بالا تنها می‌توانند حدود بیست ­وپنج دقیقه پرواز کنند و حداکثر سرعت آن‌ها پنجاه مایل در ساعت باشد. معمولاً ساختار ربات پرنده چهارملخه کوچک، به صورت یک ساختار چهار گوش می­باشد. که رایج‌ترین شکل آن‌ها به صورت "+" یا "×" می‌باشد که در هر گوشه آن یک موتور و ملخ قرار داده می­شود. هر ربات داراي پنج مؤلفه­ مي‌باشد (شکل 1).



شکل 1 : مؤلفه‌هاي يك ربات

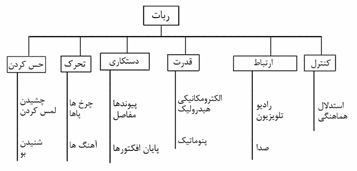
بازوي مكانيكي شامل چندين واصل است كه با مفصل‌ها به هم وصل مي‌شوند. اين واصل‌ها در جهات مختلف در فضاي كاري قادر به حركت مي‌باشند. حركت يك مفصل بخصوص باعث حركت يك يا چند واصل مي‌شود. عامل تحريك مفصل مي‌تواند مستقيماً يا از طريق بعضي انتقالات مكانيكي بر واصل بعدي متصل شود به واصل نهايي بازوي مكانيكي وسيله كاري ربات متصل شده است كه آن عامل نهايي است. هر يك از مفصل‌هاي ربات يك محور مفصل دارند كه واصل حول آن مي‌چرخد. هر محور مفصل يك درجه آزادي تعريف می‌كند. بيشتر ربات‌ها داراي شش درجه آزادي مي‌باشند به عبارت ديگر داراي شش مفصل، به منظور حركت در شش جهت. اولين سه مفصل ربات به عنوان محورهاي اصلي شناخته مي‌شوند. به‌طوركلي صرف‌نظر از جزئيات، محورهايي كه براي محاسبه موقعيت و استقرار مچ استفاده مي‌شونــد، محورهاي اصلي ربات هستند. محورهاي مفصل‌هاي باقي‌مانده، جهت قرار گرفتن دست ربات را مشخص کرده و محورهاي فرعي مي‌باشند. دو نوع مفصل اصلي به صورت گسترده در صنعت ربات‌ها بكار گرفته مي‌شود. مفصل دوراني كه نمايش دهنده حركت چرخشي حول يك محور است و مفصل انتقالي يا لغزشي كه نمايش دهنده حركت خطي در طول يك محور است.

سنسور المان حس‌کننده‌اي است که کميت‌هاي فيزيکي مانند فشار، حرارت،رطوبت، دما، و غیره را به کميت‌هاي الکتريکي پيوسته[[10]](#footnote-10) يا غيرپيوسته[[11]](#footnote-11) تبديل مي‌کند. اين سنسورها در انواع دستگاه‌هاي اندازه‌گيري، سيستم‌هاي کنترل آنالوگ و ديجيتال مانند پی ال سی مورد استفاده قرار مي‌گيرند. عملکرد سنسورها و قابليت اتصال آن‌ها به دستگاه‌هاي مختلف از جمله پی­ال­سی باعث شده است که سنسور بخشي از اجزاي جدانشدني دستگاه کنترل اتوماتيک باشد. سنسورها اطلاعات مختلف از وضعيت اجزاي متحرک سيستم را به واحد کنترل ارسال نموده و باعث تغيير وضعيت عملکرد دستگاه‌ها مي‌شوند.

کنترلر بخشي است كه به بازوي مكانيكي، هوش انجام كار را می‌دهد. كنترلر رباتها كلاً به پنج دسته تقسيم­بندي می‌شوند: كنترل با قدم ساده[[12]](#footnote-12)، سيستم منطقي پنوماتيكي[[13]](#footnote-13)، كنترلر با قدم‌هاي الكترونيكي[[14]](#footnote-14)، ميكروكامپيوتر[[15]](#footnote-15) و ميني­كامپيوتر[[16]](#footnote-16) . سه كنترلر اول در ربات‌هاي كم‌ هزينه به كار برده مي‌شوند. بيشتر كنترلرهاي امروزي براساس ميكروكامپيوترهاي معمولي مي‌باشند و سيستم كنترل براساس ميني كامپيوتر زياد رايج نمي‌باشد، چرا كه نسبت به ميكروكامپيوترها هزينه بالاتري دارند. اين واحد سيگنال‌هاي كنترلر را گرفته و به يك سيگنال در سطح توان محرك‌ها و موتورها، برای حركت، تبديل مي‌كند. اين واحــد شامل تقويت كننده‌هاي توان الكترونيكي براي ربات‌هاي الكتريكي، شيرهاي كنترلي و راه اندازهاي هيدروليكي براي ربات‌هاي هيدروليكي مي‌باشد.

اين وسائـل تحت يك ســري شرايط كنتـــرل شده و دقيــق توان لازم را برای مفصل‌ها فراهم مي‌آورند. اين توان مي‌تواند الكتريكي، هيدروليكي يا پنوماتيكي[[17]](#footnote-17) باشد. يك ربات داراي مؤلفه‌هاي؛ اسكلت ساختاري، سيستم تحريك كننده مفصل‌ها، سيستم سنسوري، كنترلر، سيستم تغيير سيگنال (تغيير سيگنال آنالوگ به ديجيتال و بالعكس، تقويت سيگنال، فيلتر كردن سيگنال و غیره) مي‌باشد. هم اينك رباتها با مفصل­هاي متحرك به خاطر سادگي ساخت و سرعت عمل و نزديكي آن به قابليت انعطاف بازوي انسان، در صنعت به طور وسيع استفاده مي‌شوند. شکل 2 سلسله مراتب يك ربات متحرك می­باشد و هر ماژول قابل تجزيه به زير سيستم‌هايي است [۱۴].

سیستم‌عامل ربات یک میان‌افزار تحت لینوکس رباتیک یعنی مجموعه‌ای از چارچوب‌های نرم‌افزاری برای توسعه نرم‌افزار ربات است. اگر چه سیستم‌عامل ربات یک سیستم‌عامل مجزا نیست اما سرویس‌هایی را فراهم می‌کند که برای خوشه‌ای از کامپیوترهای همگون طراحی شده‌اند. سرویس هایی مثل فراهم کردن یک لایه انتزاعی برای سخت‌افزار، کنترل دستگاه در سطح پایین، پیاده‌سازی عملیات‌های معمول رباتیک، انتقال پیام بین فرایندها، و مدیریت بسته‌ها است.کدها و منابع سیستم عامل ربات همیشه در دسترس و قابل اصلاح هستند که هدف آن ساده‌سازی و برنامه­ریزی برای انواع پلتفرم‌های رباتیکی است و وظیفه آن برقراری ارتباط بین کاربر، سیستم‌عامل و تجهیزات ربات‌ها است.



شکل 2 : سلسله مراتب زير سيستم‌هاي يك ربات متحرك نمونه

نرم‌افزارها در اکوسیستم سیستم‌عامل ربات به سه گروه مجزا تقسیم‌بندی می­شوند. 1) ابزارهای مستقل از زبان و مستقل از پلتفرم برای ساخت و توزیع نرم‌افزارهای برپایه سیستم‌عامل ربات، 2) کتابخانه کلاینت سیستم‌عامل ربات، 3) بسته حاوی کدهای مرتبط با نرم‌افزار.

ابزار مستقل از زبان و کتابخانه‌های کلاینت اصلی سی­پلاس­پلاس، پایتون وLisp تحت مجوز BSD منتشر شده‌اند. اغلب بسته‌ها رایگان و تحت مجوزهای متن‌ باز هستند که عملیات‌های معمول و کاربردهایی مثل درایورهای سخت‌افزار، مدل‌های ربات، انواع‌داده‌ای، طرح‌ریزی، ادراک، محلی‌سازی، نقشه‌برداری همزمان، ابزارهای شبیه‌سازی و غیره را پیاده‌سازی می‌کنند. کتابخانه‌های کلاینت اصلی C++ , Python و Lisp بیشتر به دلیل وابستگی‌شان به مجموعه بزرگی از نرم‌افزاهای متن‌ باز، منطبق با سیستم‌های شبه­یونیکس هستند. برای این کتابخانه‌های کلاینت، اوبونتو به عنوان «پشتیبانی‌شده» لیست شده‌است، در حالی که سایر سیستم‌عامل‌ها مثل فدورا، مک و ماکروسافت ویندوز «آزمایش» هستند. کتابخانه مشتری سیستم‌عامل-ربات، جاوا نیتیو، rosjava، این محدودیت‌ها را ندارد و امکان نوشتن نرم‌افزارهای بر پایه رآس برای اندروید را فراهم کرده‌است. بسته‌های سیستم‌عامل ربات در حوزه‌های ادراک، شی‌شناسی، تقسیم‌بندی، تشخیص چهره، حرکت، ردیابی حرکت، فهم حرکت، ساختار حرکت، ادراک عمق از طریق دو دوربین، قابل کنترل و برنامه‌ریزی هستند.

ایجاد یک نرم‌افزار کاملاً جامع برای ربات، کار بسیار سختی است، زیرا رفتار یک ربات، در مواجهه با یک وظیفه محوله یا قرار گرفتن در یک محیط، بسیار با انسان متفاوت است. مسائلی که انجام آن‌ها برای انسان‌ها ممکن است کم­‌اهمیت باشد، برای یک ربات شرایط پیچیده‌ای را به وجود می‌آورد. لذا برخورد با این تفاوت‌ها بسیار دشوار است. درنتیجه، سیستم‌عامل ربات زمینه‌ای را فراهم کرده تا توسعه ‌دهندگان نرم‌افزارهای رباتیک با یکدیگر همکاری کنند. به‌عنوان‌مثال، یک آزمایشگاه ممکن است درزمینه نقشه‌برداری محیط‌های داخلی تخصص داشته و بتواند سیستم طبقه‌بندی جهانی را برای تولید نقشه‌ها به کار گیرد. گروه دیگری ممکن است در استفاده از نقشه‌ها برای حرکت تخصص داشته باشد، و درعین‌حال گروهی ممکن است دارای یک رویکرد بینایی کامپیوتری باشد که به‌ خوبی برای شناخت اشیاء کوچک پیچیده کار می‌کند. سیستم‌عامل ربات به‌طور خاص برای همه گروه­ها طراحی‌شده تا با یکدیگر همکاری کنند.

گزبو[[18]](#footnote-18) نرم‌افزار رایگان برای شبیه‌سازی در سیستم‌عامل ربات است. شبیه‌سازی ربات؛ ابزاری اساسی در هر نرم‌افزار رباتیک است. یک شبیه‌ساز خوب، امکان آزمایش سریع الگوریتم‌ها، طراحی ربات‌ها، انجام تست رگرسیون و آموزش هوش‌مصنوعی را با استفاده از سناریوهای واقع ‌گرایانه فراهم می‌آورد. گزبو با دراختیار داشتن این قابلیت‌ها، امکان شبیه‌سازی دقیق و کارآمد انواع ربات‌ها در محیط‌های پیچیده داخلی و خارجی را فراهم می‌کند. گزبو از دو بخش تشکیل شده است: یک سرور که با عنوان gzserver اجرا و یک کلاینت که با عنوان gzclient اجرا می­شود. gzserver در واقع هسته مرکزی گزبو بوده و محاسبات فیزیکی و تولید داده‌های سنسورها را برعهده دارد. در مواردی که نیاز به رابط کاربری نیست می‌تواند به صورت مستقل اجرا شود. عملکرد بخش کلاینت نیز به­‌صورت ارائه یک نمایش گرافیکی رابط کاربری است که می‌تواند تجسم خوبی از شبیه‌سازی و کنترل راحت بر روی خواص مختلف شبیه‌سازی را فراهم کند [۱۱].

یک قابلیت مؤثر و با ارزش شبیه‌‌ساز گزبو به این صورت است، که سنسورهایی نظیر سنسورهای نیرو، شتاب­سنج، سونار، LIDARs، دوربین‌های رنگی و سنسورهای ابر نقاط را شبیه‌سازی می­کند، در نتیجه شما قادر خواهید بود تا در شبیه‌سازی خودتان به راحتی سنسورهای مورد نیاز را به مدلتان اضافه کنید. با این کار خروجی‌هایی نزدیک به خروجی سنسورهای واقعی را از آن انتظار داشته باشید.

ساختار کوادکوپتر

کوادکوپتر­های چهارموتوره معروف‌ترین و پرکاربردترین نوع مولتی‌روتورها هستند. محبوب‌ترین پلتفرم روتور­کرافت، کوادکوپتر است. این ماشین ها یک پهپاد پایدار و ساده برای کار را فراهم می‌کنند که می‌تواند محموله‌های متنوعی را حمل نموده، در یک فضای محدود فرود آمده و به دلیل کوچک بودن روتورهای بی صدا و آرام، تشخیص آن از بسیاری از دیگر پهپاد دشوارتر می­باشد. کوادکوپترها دارای چهار بازو سوار بر یک قاب (هر بازو دارای یک موتور به همراه پروانه ای در انتهای آن) است، پس کوادکوپتر از چهار موتور تشکیل شده است که معمولا نحوه قرار گرفتن موتورها در آن به دلیل دید بهتر دوربین، جلوی ربات، به شکل X می‌باشد ولی به حالت‌های مختلف می‌توان موتورها را کنار هم قرار داد (شکل 3). ثبات این هواگردهای شگفت‌انگیز با چرخش جفت‌ پروانه­ها در جهت خلاف یک دیگر حاصل می‌شود. کوادکوپترها در اندازه‌های مختلف یک کف دست، تا به بزرگی توان حمل مسافر ساخته می­شوند .



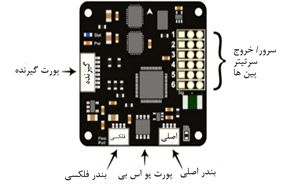
شکل 3 : نحوه قرارگرفتن موتورها در کوادکوپترها

برای ساخت یک کوادکوپتر به قاب یا فریم[[19]](#footnote-19)، چهار عدد موتور (موتور براشلس )، چهار عدد برد کنترل سرعت الکترونیکی یا اسپید کنترل[[20]](#footnote-20) و برد کنترل پرواز یا فلایت‌کنترلر[[21]](#footnote-21) لازم داریم. فریم به معنای قاب یا پوسته؛ در واقع چهارچوب و شاسی اصلی یک مولتی‌روتور است که همه تجهیزات بر روی آن نصب می‌گردند.

موتور براشلس، موتور الکتریکی هدایت و تغییر و جز موتورهای همزمان می­باشد و از منبع برق مستقیم تغذیه می‌شود و از طریق یک اینورتر برای به حرکت درآوردن موتور به سیگنال الکتریکی متناوب تبدیل می‌شود. کنترل کوادکوپتر اساساً يک مسئله سخت و از طرفي جالب توجه می‌‌باشد. داشتن دينامیک شش درجه آزادی ( سه تا انتقالي و سه تا چرخشي) و داشتن تنها چهار ورودی کنترلي مستقل (دور چهار موتور) باعث شده است تا کوادکوپترها جزو سیستم‌های دينامیكي ناقص عملگر شمرده شوند؛ بنابراين برای رسیدن به شش درجه آزادی، مجبوراً حرکت انتقالي و چرخشي آن باهم کوپل مي‌شوند. درنتیجه دينامیک آن شديداً غیرخطي مي‌باشد، مخصوصاً زماني که نیروهای ژيروسكوپي ناشي از دوران موتورها و اثرات آئرودينامیكي پیچیده آن نیز در نظر گرفته شوند.

کنترل­کننده‌ی الکترونیکی سرعت، یک مدار الکترونیکی است که سرعت یک موتور الکتریکی را کنترل و تنظیم می‌کند. در یک کوادکوپتر معمولا چهار کنترل‌کننده الکتریکی سرعت وجود دارد که هر کدام به یک موتور، منبع توزیع برق و کنترل‌کننده پرواز متصل می‌شوند. زمانی که سیگنال مربوطه از کنترل‌کننده پرواز به اسپیدکنترل داده می‌شود این قطعه از طریق کنترل برق ورودی به موتورها می‌تواند سرعت چرخش آن را کنترل کند. اسپیدکنترل یکی از مهم‌ترین قسمت یک کوادکوپتر محسوب می‌شود زیرا بسیاری از عملکردهای کوادکوپترها از جمله سرعت، ارتفاع و غیره به وسیله موتورها و سرعت چرخش آن‌ها تعیین می‌شود.

سیستم کنترل پرواز یکی از مهم‌ترین اجزای مولتی‌روتورها می‌باشد که در حقیقت مغز متفکر یک مولتی‌روتور است و عملیات هماهنگی چهار موتور پرنده را انجام می‌دهد. فلایت‌کنترلر در واقع یک کامپیوتر کوچک است که وظیفه آن کنترل و تعیین میزان دور در دقیقه هر موتور است. زمانی که خلبان دستور تغییر سرعت یا جهت می‌دهد، برد کنترل پرواز آن‌ها را از دستگاه گیرنده دریافت می‌کند و پس از پردازش، به کمک قطعات اسپید­کنترل، سرعت هر یک از موتورها را مطابق با دستور تغییر می‌دهد. فلایت‌کنترل مهمترین قطعه یک هلی‌شات می­باشد. در هنگام پرواز هلی‌شات به دلایل مختلفی ممکن است از مسیر حرکت خود خارج شود و قطعه‌ای که وظیفه کنترل پرواز هلی‌شات را بر عهده دارد، همان فلایت‌کنترل است. با پیشرفت علم الکترونیک، قطعه فلایت‌کنترل به کمک انواع سنسورها می‌تواند تعادل پرواز هلی‌شات را به خوبی حفظ کند، شکل 4 نمونه ای از فلایت­کنترل به­همراه نام پورت­هایش است.

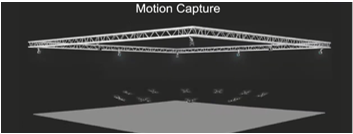


شکل 4 : پورت‌های کنترل پرواز

تعدادی از اجزاء تشکیل دهنده کوادکوپتر:

1. رادیو کنترل[[22]](#footnote-22) یا فرستنده گیرنده رادیویی: وسیله‌ای است که دستورات کنترل کوادکوپتر را به صورت بی‌سیم و از طریق امواج رادیویی برای کوادکوپتر ارسال می­کند. رادیو کنترل دستور خلبان برای هدایت کوادکوپتر را به سیگنال قابل درک برای فلایت‌کنترلر تبدیل می‌نماید. فرکانس ارتباطی که اکثر رادیو کنترل‌ها از آن استفاده می‌کنند، چهار و دو دهم گیگاهرتز است و تمامی مولتی‌روتورها برای هدایت دارای رادیو کنترل هستند.
2. ملخ چهار عدد ( دو عدد ساعتگرد و دو عدد پادساعتگرد): ملخ‌ها در مولتی‌روتورها وظیفه ایجاد نیروی بالابرنده یا جلو رونده برای به حرکت در آمدن پرنده و شناور شدن آن در هوا و تغییر مسیر را دارند که این نیرو توسط حرکت همزمان ملخ‌ها انجام می‌شود. کوادکوپترها با توجه به تعداد موتورها از چهار ملخ استفاده می‌کنند. ملخ‌ها در مولتی‌روتور نیروی لازم را از موتورها گرفته و همیشه دو تا از ملخ‌ها و موتور درون آن‌ها در جهت مخالف هم کار می‌کنند و با توجه به نوع پرنده این حرکت متفاوت است.
3. باتری و شارژر: در کوادکوپتر از باتری لیتیم پلیمر[[23]](#footnote-23) که دارای میزان ولتاژ هفت و سه دهم ولت تا دو و بیست و دو صدم ولت و میزان آمپر صد میلی آمپر تا ده هزار میلی‌آمپر باشد، استفاده می‌شود و برای شارژ کردن باتری‌های کوادکوپتر به یک شارژر استاندارد باتری‌های لیتیومی نیاز است.
4. توزیع‌کننده انرژی: وظیفه توزیع‌کننده انرژی دریافت برق مورد نیاز از باتری و توزیع بین قسمت‌های مختلف کوادکوپتر است.
5. ماژول جی‌پی‌اس: در کوادکوپتر به منظور تعیین موقعیت مکانی و جهت آن مورد استفاده قرار می‌گیرد و اطلاعات لازم را در اختیار فلایت‌کنترلر قرار می‌دهد.
6. سیستم FPV : اصطلاح FPV مخفف عبارت First Person View است و تصاویری از نمای اول شخص یا همان “زاویه دید پرنده” معروف در میان خلبانان سیستم های پروازی کنترلی است. در سیستم FPV اغلب یک عدد دوربین نصب شده روی کوادکوپتر، فرستنده تصویری، گیرنده تصویری و صفحه نمایش یا عینک گاگلز مورد استفاده قرار می گیرد.

7 . فلایت‌کنترل Pixhawk2 بر مبنای آردینو، مشهور به Pixhawk Cube آخرین نسخه از فلایت‌کنترل مشهور جهان PX4 است. دارای پردازنده ۳۲ بیتی STM32، جایروسکوپ L3GD20H و همچنین سنسور شتاب سنج LSM303D است. فلایت‌کنترل Pixhawk2.1 یک اتوپایلوت مناسب برای سیستم‌های بدون سرنشین از جمله هواپیماهای بدون سرنشین چند منظوره، مولتی‌روتورها، هلیکوپترها، هواپیماهای برخاست و فرود عمود[[24]](#footnote-24)، قایق‌ها، اتومبیل‌ها و غیره است. با استفاده از Pixhawk2.1 و سامانه هوشمند Mission Planner میتوان به پرنده خود ماموریت داد و ربات پرنده به صورت خودکار ماموریت خود را اجرا خواهد نمود. کافی است نقاط جی‌پی‌اس را روی نقشه تعیین و بقیه موارد را به Pixhawk و سیستم هوشمند سپرد و پرواز یا حرکت دستگاه را تا رسیدن به مقصد و سپس برگشت به خانه دنبال کرد. همچنین این فلایت‌کنترل قادر خواهد بود یکبار یک مسیر حرکتی را رفته و نقاط حرکت را ذخیره نماید. با ذخیره این نقاط در دفعات بعد پرنده همان مسیر را بدون دخالت فرد به هر تعداد دفعات خواهد پیمود. در فلایت‌کنترل Pixhawk2.1 ایستگاه زمینی می‌تواند لپتاپ، گوشی هوشمند با سیستم‌عامل اندروید و یا آیپد باشد و با استفاده از ماژول تله­متری[[25]](#footnote-25)و ماژول OSD می‌توان رادیو کنترل را کنار گذاشت و با لپتاپ پرنده را هدایت نمود (شکل 5).



شکل 5 : کنترل زمینی

روش‌های کنترل

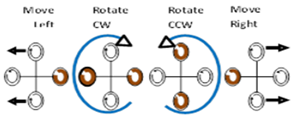
روش‌های کنترل پیشنهاد شده برای چهارپره را می‌توان به سه دسته خطی، غیرخطی و مبتنی بر یادگیری تقسیم‌بندی کرد. مزایای روش‌های خطی، امکان پیاده‌سازی آسان و امکان تنظيم ورودی‌های کنترلی برحسب عملکرد سامانه می‌باشد. معایب این روش‌ها در مشکل بودن اثبات پایداری و تنزل عملکرد در خارج از دامنه طراحی شده می‌باشد. روش‌های تناسبی، انتگرالی، مشتقى، رگولاتور درجه دوم خطی و برنامه‌ریزی بهره در دسته روش‌های خطی قرار می‌گیرند که برای چهارپره پیاده‌سازی شده‌اند. روش‌های غیرخطی توانایی بالایی در اجرای مانورهای پرشتاب داشته و نسبت به اغتشاشات و نامعینی‌ها قوام بالایی دارند. این روش‌ها معمولاً برای غلبه بر محدودیت‌های روش‌های خطی ابداع شده‌اند، با این حال روش‌های غیرخطی برتری چندانی در آزمایش‌های عملی به دست نیاورده‌اند. دلیل این امر در حساسیت بالای آن‌ها به پارامترهای مدل دینامیکی می‌باشد که پیاده‌سازی این روش‌ها را مشکل نموده است.

از جمله روش‌های غیرخطی که برای کنترل چهارپره به کار رفته اند می‌توان به وارون‌سازی دینامیکی، گام رو به عقب و مود لغزشی اشاره کرد. مهم‌ترین مشخصه روش‌های مبتنی بر یادگیری این است که مدل دینامیکی چهارپره در آن‌ها به کار نمی‌رود، بلکه از داده‌های حاصل از پرواز واقعی برای آموزش به سامانه کنترل استفاده می‌شود. از جمله این روش‌ها که برای کنترل چهارپره به کار رفته‌اند می‌توان به منطق فازی و شبکه‌های عصبی اشاره کرد. این روش‌ها بر مبنای مدل دینامیکی شکل نمی‌گیرند و درنتیجه می‌توان آن‌ها را روی پلتفرم‌های گوناگون به کار گرفت. با این حال پایداری و قوام این روش‌ها به راحتی قابل تحلیل نیست و ارزیابی عملی در مقیاس بزرگ بر روی این روش‌های کنترلی تاکنون صورت نگرفته است میتوانیم با بررسی در مقیاس های کوچک و پردازش ویدیو های گرفته شده از محیط اطراف، شخص معرفی شده به سامانه را پیگیری و دنبال کنیم و خشونت و خطرات محیط اطراف را شناسایی کند.

سامانه هدایت نقش تعیین‌کننده‌ای در پرواز چهارپر‌‌ها بر عهده دارد. مهم‌ترین رویکردها در هدایت پرواز ربات شامل رهبر- پیرو، ساختار مجازی و رفتار محور می‌باشد. همچنین چهار رفتار تعقیب مسیر، حفظ چیدمان، اجتناب از برخورد و اجتناب از مانع در پروازها مورد توجه قرار می‌گیرند که هر یک از رویکردهای مذکور، روی یک یا چند رفتار تاکید بیشتری دارند و مرز دقیقی بین آن‌ها وجود ندارد. در این پژوهش تمرکز اصلی بر تعقیب مسیر ،هدف و اجتناب از مانع خواهد بود که با هر دو روش کنترلی خطی و غیرخطی بر مبنای وارون‌سازی دینامیکی انجام شده و نتایج مربوطه مقایسه می‌گردند.

مدل‌سازی دینامیکی ربات پرنده

فرضیات استفاده شده برای مدل‌سازی چهارپره را می‌توان به این صورت خلاصه نمود: 1) سازه چهارپره صلب و متقارن است. 2) مرکز جرم و مرکز دستگاه بدنی چهارپره روی یک نقطه قرار دارند. 3) نیروهای آیرودینامیکی ناشی از چرخش روتورها با مربع سرعت زاویه‌ای روتورها متناسب هستند. چهارپره، یک وسیله نقلیه کم عملگر می‌باشد، و با وجود داشتن شش درجه آزادی، تنها در چهار درجه امکان اعمال ورودی کنترلی مستقل دارد. چهارپره از هیچ گونه سطوح کنترلی برای ایجاد نیرو استفاده نمی‌کند و تنها به تغییر سرعت زاویه‌ای روتورهای خود وابسته است. در شکل 6 محورهای بدنه و شماره روتورها به طور قراردادی نشان داده شده است و وضعیت ترآست موتورها در مانور به صورت شناوری، غلتش یا چرخش و گردش است. [۱۳]



شکل 6 : نحوه حرکت ربات چهارپره

نیروها و گشتاورهای وارد بر چهارپره از جنس آیرودینامیکی و جاذبه می‌باشند. معمولا نیروها و گشتاورهای آیرودینامیکی وارد­ بر چهارپره به­صورت روابط 1 مدل‌سازی می‌شوند. تأثیرات آیرودینامیکی[[26]](#footnote-26) دیگری همچون تأثیر زاویه حمله انحراف تیغه و اغتشاش جریان هوا در مراجع بررسی شده‌اند. با این حال به دلیل این که مدل‌سازی این تأثیرات بسیار پیچیده است و تنها در سرعت‌های بالا مقدار قابل توجهی پیدا می‌کنند، در این پژوهش از وارد کردن آن‌ها به مدل چهارپره صرف‌نظر شد.



(1)

نکته مهم درباره سرعت زاویه‌ای روتورها، تاخیر موجود در روتورها می‌باشد که تأثیر منفی در پایداری چهارپره دارد. در واقع چهارپره بر خلاف هواپیماهای بال ثابت به فرکانس بالایی برای اعمال ورودی‌های کنترلی نیاز دارد و در صورتی که روتورها در پیاده‌سازی ورودی‌های با فرکانس بالا موفق نباشند چهارپره دچار ناپایداری خواهد شد. تاخیر روتورها ناشی از مسائل مکانیکی و الکتریکی می‌باشد و می‌توان آن را به طور نسبتا دقیق با تابع تبديل مرتبه دوم مدل‌سازی کرد. با این حال ثابت زمانی ناشی از تاخیر مکانیکی بسیار بیشتر از تاخیر الکتریکی است و در نتیجه دینامیک روتور با یک معادله دیفرانسیل مرتبه اول به صورت رابطه 2 قابل تقریب زدن است.



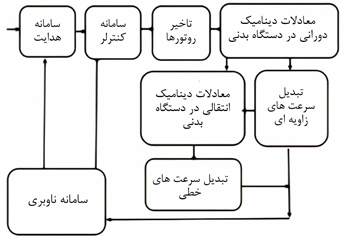
(2)

که در آن، T ثابت زمانی موتور می‌باشد یعنی مدت زمانی که برای روتور طول می‌کشد تا به شصت و سه درصد مقدار ورودی پلهΩd برسد همچنین مقدار سرعت زاویه ای Ω مطلوب و مقدار سرعت زاویه‌ای واقعی می‌باشند. معمولا هرچه اندازه روتور و ملخ بزرگ‌تر باشد این ثابت زمانی بیشتر خواهد بود. در نتیجه تابع تبدیل روتور در فضای حالت به صورت رابطه 3 به دست می‌آید.



(3)

شایان ذکر است، تابع تبدیل تاخیر روتورها در مقدار ورودی‌ها ضرب شده و سپس به سامانه چهارپره وارد می‌شود. در نهایت بلوک دیاگرام مدل دینامیکی چهارپره در شکل 7 خلاصه شده است، معادلات دینامیک انتقالی و دورانی در دستگاه بدنی نوشته و سپس به دستگاه اینرسی تبدیل شده‌اند.



شکل 7 : دیاگرام مدل دینامیکی چهارپره در سامانه حلقه بسته

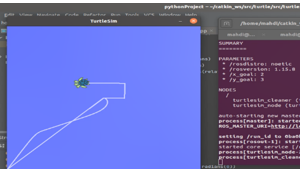
معرفی طرح

به منظور دسترسی به حالت پایدار پروازی در شرایط معلق در هوا، در این پروژه بهبود کیفیت مکانیکی و کامپیوتری ربات‌های پرنده، از نظر توان و همچنین از نظر سرعت موقع پرواز مورد نظر می‌باشد که در دستیابی به اهداف و همراهی هدف تعیین شده مفید باشد. ما برای شبیه‌سازی از شبیه‌ساز گزبو استفاده کردیم برای اینکه هم امکان ایجاد شرایط محیطی مثل جریان‌های هوایی و نیروی جاذبه و دما وجود دارد و هم ایجاد محیط‌های متنوعی مثل کوه و ساختمان و غیره را داراست. از طریق ارتباط سیستم‌عامل ربات و زبان‌های برنامه‌نویسی‌سی‌پلاس‌پلاس و پایتون ارتباطات لازم با الگوریتم های هوش مصنوعی و پردازش تصویر را برقرار و به کواد دستورات مربوطه را می‌دهیم، امکان شبیه‌سازی در دیگر نرم‌افزارها مشابه نیز همچون متلب و ویبات وجود دارد اما قدرتمندی لینوکس و سیستم‌عامل ربات ما را علاقه­مند به یادگیری و پیاده سازی ربات در این محیط کرد.

روش شبیه‌سازی و پیاده‌سازی

یکی از قدرتمندترین ماشینهای مجازی وی­ام­ویر[[27]](#footnote-27)است که قابلیت ایجاد چند سیستم‌عامل مجازی هم زمان، نداشتن مشکل با سخت‌افزار رایانه، قابلیت تعیین میزان حافظه مورد استفاده، گرافیک خوب و ظاهر مناسب را دارا است. محیط شبیه­ساز رآس و لاک­پشت[[28]](#footnote-28) در این ماشین مجازی راه­اندازی شده­اند. کوچک‌ترین واحد پردازش در رآس همانند یک فایل اجرایی داخل پکیج، برای هر هدف یک گره[[29]](#footnote-29) جهت مدیریت بهتر سیستم تعریف می­شود. به عنوان مثال یک ربات شامل بخش‌هایی نظیر حسگر و موتور می­باشد که کد مربوط به هر کدام از بخش‌ها را می‌توان در یک گره نوشت، گره می‌تواند به زبان پایتون یا سی پلاس پلاس نوشته شود و به کمک تاپیک با یکدیگر ارتباط برقرار می‌کنند. علی‌رغم اهمیت واکنشی‌بودن و زمان تأخیر کم در کنترل ربات، رآس یک سیستم‌عاملی بی‌درنگ نیست، گرچه امکان ادغام کدهای بی‌درنگ با رآس وجود دارد. مشکل عدم پشتیبانی سیستم‌عامل بی‌درنگ در نسخه رآس دو حل شده‌است. دو دستور برای اندازه‌گیری سرعت انتشار پیام و پهنای باند مورد استفاده‌ وجود دارد: rostopic hz topic-name و rostopic bw topic-name

این دستورات به پیام‌ها گوش داده و در خروجی به ترتیب تعداد پیام در ثانیه و تعداد بایت در ثانیه را اندازه می‌گیرند. این دستورات می‌توانند به راحتی نشان دهند که پیام‌ها به صورت متناوب روی تاپیک‌های مشخصی منتشر شده‌اند. یک گره برای شبیه‌سازی لاک‌پشت، دریافت‌کننده یک فرمان سرعت و ارسال موقعیت لاک‌پشت می­باشد که حرکت و شبیه­سازی در آن به­صورت شکل 8 است. یک گره دریافت‌کننده پیام از دکمه‌های کیبورد و ارسال یک فرمان سرعت می­باشد. دو راه برای اتصال گره‌ها به همدیگر وجود دارد، 1) استفاده از ارتباط دائمی و یک طرفه توسط تاپیک‌ها، 2) یک ارتباط دوطرفه و هم زمان توسط سرویس‌ها. سرویس‌ها در رآس به این صورت هستند که مشتری[[30]](#footnote-30) ارسال درخواست انجام یک از کار‌ها را می‌فرستد پاسخ را از سرور دریافت می‌کند. سرور آن کار را انجام می‌دهد و منتظر می‌ماندن برای درخواست بعدی که از سمت مشتری ارسال می­شود .



شکل 8 : نود ترتلسیم

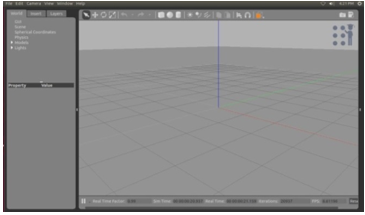
ساختار گزبو متشکل از سرور، کلاینت، سنسورها ،پردازش ها و غیره که در شکل 9 قابل مشاهده است. برای اضافه‌شدن یک یا چند ربات به گزبو به یک محیط نیاز داریم، این محیط ممکن است شامل جزئیات و عوارض زمین، ساختمان‌ها، دیوار، میز و غیره و همچنین شامل یک دسته از ربات‌های گروهی باشد.



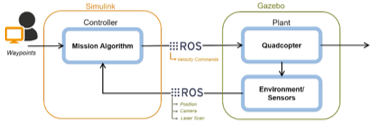
شکل 9 : ساختار گزبو

یک مدل مرسوم پیش‌فرض در گزبو شامل فقط یک سطح صاف از زمین با یک ‌جهت عمودی برای بردار جاذبه است (شکل 10) [۱۱].

در نوار پایینی محیط شبیه‌سازی گزبو، چند رابط کنترلی و نمایشگر وجود دارد. ما می‌توانیم به روش‌های گوناگونی مدل‌های دیگر را وارد شبیه‌ساز بکنیم. با استفاده از تب Insert می‌توان هر مدل از قبل تعریف‌ شده‌ای در لیست مدل‌ها را اضافه کرد. لیست مدل‌های قابل ‌دسترس، شامل مدل‌های آنلاین موجود در پایگاه داده‌های گزبو و همچنین مدل‌های محلی تعریف‌شده که در مسیر (مخفی) gazebo./~ می‌باشد. علاوه براین، می‌توان یک مدل را به‌صورت دستی با فراخوانی یک گره گزبو به همراه مسیری که مدل مربوطه در آن قرار دارد، فراخوانی کرد. برای اجرای مدلهای محلی، باید فایل‌‌های را مطابق استاندارد ایجاد و یا از دیتابیس گزبو دانلود کرد.دیاگرام طرح ما به صورت شکل 11 از شبیه‌سازی و سیستم‌عامل ربات و گزبو تشکیل شده است که نمایش می­دهد رآس ارتباط این دو را به خوبی برقرار می‌کند.



شکل 10 : محیط گزبو



شکل 11 : معماری سیستم

سالیدورکس

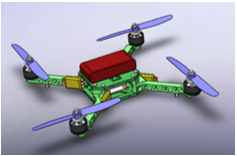
سالیدورکس[[31]](#footnote-31) یک نرم‌افزار مهندسی طراحی به کمک رایانه تحت ویندوز است. این نرم‌افزار دارای سه محیط به نام‌های پارت، اسمبلی و دراوینگ می‌باشد. محیط اول برای رسم قطعه بوده، در محیط دوم قطعات یک مکانیسم بر روی هم سوار شده و در محیط آخر از آن‌ها نقشه مهندسی (معمولاً برای نسخه چاپی ) تهیه می‌شود.

محیط قطعه (پارت) : هر سیستم مکانیکی از قطعات و اجزا مختلفی تشکیل یافته ‌است. در این بخش از نوار ابزار بیشترین استفاده صورت می‌گیرد. محیط مونتاژ (اسمبلی) : پس از اینکه قطعات به صورت مجزا در محیط پارت ایجاد شدند آن‌ها را وارد محیط اسمبلی می‌نمایند. در اینجا با اعمال قیود خاص توسط دستور Mate از نوار ابزار اسمبلی قطعات مجزا را به یکدیگر مرتبط می‌کنند. از جمله این قیود می‌توان به موازی، هم مرکز، عمود یا قیود پیشرفته نظیر قیود ایجاد چرخ­دنده یا بادامک اشاره نمود. یکی از امکانات جالب در این محیط امکان ایجاد تسمه برای پولی‌ها است که آن‌ها را به یکدیگر مرتبط می‌کند. می‌توان با اعمال سرعت خطی و سرعت زاویه‌ای و همچنین شتاب ثقل، عملکرد مکانیزم را برآورد نموده تا تقریبی از کار مکانیزم به دست آید. محیط نقشه‌کشی (دراوینگ[[32]](#footnote-32)) : در این محیط می‌توان اقدام به طراحی نقشه‌های دو بعدی نموده یا از قطعات و مکانیزم‌هایی که در دو محیط قبلی ساخته شده‌اند استفاده نمود. قابلیت‌های متعدد این محیط یک تخته رسم الکترونیکی را برای کاربر به ارمغان می‌آورد که به­راحتی می‌تواند هر گونه ترسیمی را به شکل دلخواه و بدون مشکلات استفاده از راپیدوگراف و غیره در کوتاه­ترین زمان انجام دهد. از جمله قابلیت‌ها می‌توان به اندازه‌گذاری خودکار نقشه‌های ایجاد شده از روی قطعات، تعیین نوع‌هاشور برای سطح مقطع با توجه به جنس قطعه، تعیین نوع خطوط و غیره اشاره کرد.

با کمک سالیدورکس به راحتی می‌توان نقشه‌های دوبعدی از اجسام ترسیم کرده و آن‌ها را مدل کرد. در واقع می‌توان بدون درگیری با محاسبات مورد نیاز نقشه‌کشی صنعتی، شکل خود را ساخت و با استفاده از محیط طراحی نما و نقشه، سه جهت شکل خود را استخراج کرد [۱۵]. سالیدورکس ابزاری ساده، با سرعت بالاتر و قابلیت ارتباط با تمامی نرم‌افزارهای ماشین کاری و نرم‌افزارهای تحلیل و امکان استفاده از نرم افزارهایی مثل متلب، گزبو و شبیه‌ساز ویبات و غیره می­باشد. و در این ابزار امکان معادله‌نویسی بین پارامترها و انداز‌ه‌های مختلف در مدل و امکان استفاده از جداول طراحی به منظور برقراری ارتباط بین اندازه‌ها و معادلات در محیط اکسل را دارد. استفاده از نرم‌افزار سالیدورکس در کارخانجات تولید قطعات خودرو، رباتیک، ماشین آلات و همچنین در خیلی از موارد توسط افرادی که کار نقشه­کشی­صنعتی و همچنین طراحی مدل‌های چند بعدی انجام می‌دهند، رواج دارد. برای طراحی یک پرنده در محیط سالیدورکس ما ابتدا پارت‌های مجزا طراحی کرده همانطور که در شکل 12 تمامی پارت‌ها قابل مشاهده‌اند که شامل سیزده پارت و دو فایل اسمبل شده می‌شود. درنتیجه فایل نهایی اسمبل شده توسط سالیدورکس به صورت شکل 13 و دارای قید حرکت پره‌های شبیه‌ساز است.



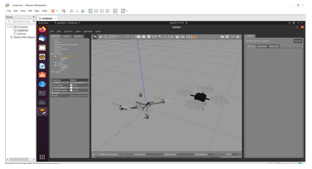
شکل 12 : پارت‌های طراحی پروژه



شکل 13 : کوادکوپتر در سالیدورکس

شبیه‌سازی و نتایج

در محیط گزبو، داشتن سنسورها و ماژول و قالب‌های آماده زیاد این امکان را به ما می‌دهد به راحتی بتوانیم روی هر رباتی تست و آزمایش انجام دهیم و یا به اون ربات سنسور اضافه یا کم کنیم تست و آزمایشی که بر روی ربات SJTU\_drone انجام شده به صورت شکل 15 می‌باشد همچنین سنسورهای به کار گرفته شده که سنسور بینایی یا دوربین و صفحه کنترل و سنسور imu می‌باشد که قابل مشاهده ویرایش ‌اند. پرنده طراحی شده توسط سالیدورک به شبیه‌ساز اضافه گردید (شکل 14 ) در شبیه‌ساز گزبو می‌توانیم عوامل محیطی و فیزیکی مفصل‌ها و گشتاورها را به ربات داده و برای برنامه نویسی آماده‌اش کنیم که بتوانیم ارتباط بین گره‌های مفاصل و کنترل‌های خطی برقرار کنیم و ویژگی‌هایی مثل گرانش زمین و جریان‌های هوایی را تنظیم نمود.



شکل 14 : ربات شبیه‌ساز و طراحی شده



شکل 15 : پرنده sjtu

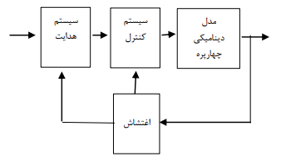
**همچنین** ویبات یک شبیه‌ساز ربات سه‌بعدی رایگان و منبع باز برای گنولینوکس، مک و ویندوز. این نرم‌افزار برای شبیه‌سازی ربات‌های موبایل اغلب برای اهداف آموزشی استفاده می‌شود. یکی از مزایای اصلی آن این است که به کاربر امکان تعامل با مدل را هنگام شبیه‌سازی می‌دهد. این برنامه بر اساس موتور فیزیک و موتور رندر[[33]](#footnote-33) ساخته شده است. با استفاده از این نرم‌افزار، کاربران می‌توانند سلاح‌های صنعتی، وسایل نقلیه هوافضا، ربات‌های چند پا، ربات‌های مدولار، اتومبیل، هواپیماهای بدون سرنشین پرنده، وسایل نقلیه خودکار زیر آب و انواع ربات‌ها را مدلسازی، برنامه‌ریزی و شبیه‌سازی کنند.

ویبات از ODE استفاده می‌کند برای تشخیص برخورد و شبیه‌سازی بدنه صلب پویا است. کتابخانه ODE به شما امکان شبیه‌سازی فیزیک اشیا و ربات‌ها را از طریق تعریف هندسی و دینامیکی قطعات تشکیل دهنده آن فراهم می‌کند. همچنین به شما امکان می‌دهد برای تجسم بهتر، رنگ‌ها و بافت‌ها را مشخص کنید. این نرم‌افزار شامل تعدادی از سنسورها و محرک‌هایی است که به طور مکرر در رباتیک استفاده می‌شود، با مدل‌های دینامیکی دیگر می‌توان کنترل ربات را با سی، سی پلاس پلاس، جاوا، پایتون، متلب و رآس نوشت. ربات با زبان سی ‌برنامه‌ریزی شده است و با دریافت دستورات از صفحه کلید آن‌ها را به شبیه‌ساز اعمال می‌کند. این نرم‌افزار بسیار ساده‌تر و با رابط کاربری بهتری نسبت به سیستم‌عامل ربات و گزبو است به گونه‌ای که با کم‌ترین زمان آموزش امکان پیاده‌سازی این پروژه فراهم شد، ویبات اکثر ویژگی‌های گزبو را دارد همین‌طور که در شکل 16 قابل مشاهده است با استفاده از برنامه و الگوریتم های سی پلاس پلاس و پایتون ربات در حال پرواز و بالا سمت چپ نمای دوربین ربات را مشاهده می‌کنیم .

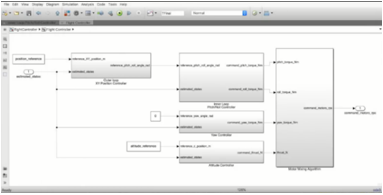


شکل 16 : محیط ویبات وصفحه کدنویسی و دوربین ربات

برای شبیه‌سازی خطی پرواز چهارپره از محیط شبیه‌سازی سیمولینک بهره می‌بریم. در این محیط امکان اعمال تأخیر روتور و نویز سنسورها به سامانه حلقه بسته به راحتی امکان‌پذیر است و با استفاده از توابع ترسیمی متلب قادر خواهیم بود نحوه حرکت چهارپره‌ها در فضای سه‌بعدی را رسم نماییم. بلوک‌های ساخته شده برای مدل­دینامیکی، سامانه کنترل، هدایت و اغتشاش اعمال شده در شکل 17 نشان داده شده‌اند. همچنین مدل سازی ربات پرنده در متلب به صورت شکل 18 انجام گرفت. [۱۲]



شکل 17 : بلوک‌های طراحی شده در محیط سیمولینک



شکل 18 : بلوک دیاگرام سیمیولینک متلب

مقدار پارامترهای آیرودینامیکی و سازهای اعمال شده در شبیه‌سازی در جدول 2 مشاهده می‌شود. چهارپره یک وسیله پرنده ذاتأ ناپایدار است که از هیچ سطح کنترلی استفاده نمی‌کند و برای کنترل وضعیت و موقعیت خود در فضای سه‌بعدی تنها به تغییر دادن دور موتورهای خود متکی است در نتیجه تأخیر موجود در روتورها اصلی‌ترین عامل کاهنده پایداری چهارپره می‌باشد. طبق آنچه که در بخش مدلسازی گفته شده معادله دینامیکی روتور به صورت یک تابع تبديل مرتبه اول وارد سامانه می‌شود. برای اعمال تأخیر موتور و نویز سامانه ناوبری از منابع مشابه کمک می‌گیریم. مقدار تأخیر موتور به میزان پنج صدم ثانیه مناسب به نظر می‌رسد.

جدول 1 : پارامترهای آیرودینامیکی و سازهای چهارپره

|  |  |  |
| --- | --- | --- |
| پارامتر | مقدار | واحد |
| Ixx | 075/0 | Kgm2 |
| Iyy | 075/0 | Kgm2 |
| Izz | 13/0 | Kgm2 |
| m | 65/1 | Kg |
| J2 | 00006/0 | Kgm2 |
| l | 23/0 | M |
| b | 0000313/0 | Ns2 |
| d | 00000075/0 | Nms2 |

همچنین نویزی با میانگین صفر به سیگنال‌های موقعیت و سرعت و واریانس یک ده هزارم به سیگنال‌های زاویه اضافه می‌شود. بنابراین قادر خواهیم بود عوامل اصلی تضعیف‌کننده عملکرد چهارپره را در شبیه‌سازی‌ها لحاظ کرده و به نتایج دقیق‌تری برسیم در این سناریو سامانه کنترل را به صورت دستی تنظیم می‌نماییم. این فرآیند برای پیاده‌سازی در سامانه‌های کنترل پرواز رایج همچون آردوپایلوت بهتر است، چرا که در این سامانه‌ها، بهره‌های سامانه کنترل یک بار توسط کاربر تنظیم می‌شوند و پس از آن در مأموریت‌های مختلف ثابت نگه داشته می‌شوند. همچنین بهینه‌سازی سامانه هدایت در ایستگاه زمینی انجام می‌شود و می‌تواند خود را با سامانه کنترل سازگار نماید. سامانه هدایت بر مبنای رویکرد رفتارمحور طراحی شده‌اند. با محدود کردن الگوریتم رفتار محور به رفتار تعقیب هدف و مسیر زمینه برای بهینه‌سازی آن فراهم شده است. شبیه‌سازی صورت گرفته در شرایط وجود نامعینی و عدم ایده‌آل‌نگری در سامانه، نشان­دهنده عملکرد خوب سامانه هدایت و کنترل پرواز در اجرای ماموریت می‌باشد. در این پژوهش عملکرد کنترل‌کننده‌های خطی و غیرخطی برمبنای وارون­سازی دینامیکی و توانایی آن‌ها در تعقیب هدف و مسیر یک سیستم غیرخطی با پردازش های تصویری به خوبی قابل ملاحظه است. با انجام این پروژه، زمینه برای اعمال الگوریتم اجتناب از موانع و پرواز گروهی در آینده فراهم شده است که توانایی دفاع دربرابر خطراتی مانند ترور شخصیت ها را داشته باشند. در این پژوهش تأثیر سنسورها به صورت نویز سفید مدل‌سازی شد ولی برای پیاده‌سازی لازم است تا مدل دقیقی از سنسورها در محاسبات آورده شود. روش بهتر برای افزایش دقت شبیه‌سازی از طریق تحلیل داده‌های پروازی برای شناسایی مدل دینامیکی چهارپره می‌باشد.

نتیجه­گیری

در ربات‌های پرنده چهارپره جهت افزایش قدرت و سرعت نیاز به تنظیم موتورها و انتخاب نوع موتورها با تناسب یک سوم است. به این صورت که ما با تنظیم و کم کردن وزن هرکدام می‌توانیم وزن دیگری را افزایش دهیم. هر مدل پرنده برای حالت و کاری خاص طراحی و پیاده‌سازی می‌شود برای مثال کواد یا چهار موتور و ملخ‌های کوچک بهترین گزینه برای سرعت است که این قابلیت‌ها قدرت مانور بالایی به ربات می‌دهند و رباتهای پرنده­ با هشت موتور که به صورت بالا پایین‌اند یا شانزده موتور برای قدرت بالا و بلند کردن وزنه‌های سنگین مناسب اند. برای **ربات در زمینه ها محافظت از افراد، سرعت و دقت کار برای ما اهمیت دارد که میتواند ربات های چهارموتور با فلایت کنترلر و قابلیت های زیاد اتوپایلوت و پردازنده های قدرتمند که سیستم عامل ربات مدیریت منابع را در دست میگیرد در دنیا واقعی بهره برد و پردازش­های هوش مصنوعی را با­استفاده از انتقال تصاویر بر­روی سرور پیاده سازی کرد که از سنگین شدن سخت افزار ربات و پردازش های سنگین تصویر جلوگیری شود و در­ربات اختلال ناگهانی پیش نیاید و ربات نگهبان به­خوبی بتواند ماموریت­اش را کامل کند .**

مراجع و منابع

[1] ام اُکین، جیسن، ورود آسان به رآس، زهرا بروجنی، 1395.

[2] قرباني، محمدرضا، رهیافت کنترل بهینه برای پرواز آرايش يافته کوادکوپترها، پايان­نامه کارشناسي ارشد دينامیک پرواز و کنترل، دانشگاه صنعتی شریف، 1396.

[3] شفیعی، آرمین، مدلسازی وشناسایی سیستم کوادروتور، پایان نامه کارشناسی ارشد رشته مهندسی برق کنترل، موسسه شهاب دانش، 1395.

[4] احسان زیبایی، محمدعلی امیری آتشگاه و نفیسه محمدیان افتح، سامانه هدایت و کنترل رفتار- محور ربات پرنده جهت تعقیب راهبر، فصلنامه علمی- پژوهشی مكانيك هوافضا، جلد 15، شماره1، 1398.

[5] قربانی، مجتبی، حسینی، سید کمال،کنترل پیش­بینی غیرخطی ربات شش درجه ازادی سکوی استوارت، پایان نامه کارشناسی ارشد ،دانشگاه فردوسی مشهد، 1395.

[6] میرزاعلی، حامد، کریمی، جلال، شاهی اشتیانی،محمد علی، بهینه‌سازی چند معیاری مقید پروفیل ماموریتی ربات پرنده، فصلنامه علمی- پژوهشی مكانيك هوافضا، جلد 14، شماره4، 1397.

[7] سید مهدی،سید محمد، محمدی مقدم،مجید، طراحی، ساخت و کنترل ربات پرنده چهار ملخه کوچک برای طی مسیر خاص با استفاده از روش پرواز کور، جهاد دانشگاهی، سال سوم، 1396.

[8] J. Gandhinathan and Joseph, Ramkumar, Ros Robotics Projects, 2019, Second edition.

[9] Martinez and Fernández , Aaron and Enrique , Learning ROS for Robotics Programming, 2013 , First published.

[10] YoonSeok Pyo , HanCheol Cho, RyuWoon Jung , TaeHoon Lim , ROS Robot programing, 2017, First Edition

[11] N. Koenig and A. Howard, "Design and use paradigms for Gazebo, an open-source multi-robot simulator," IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2004.

[12] Z. Bo, X. Bin, Z. Yao and Z. Wei, "Hardware-in-loop simulation testbed for quadrotor aerial vehicles," Proceedings of the 31st Chinese Control Conference, 2012.

[13] R. Lalantha and R. Munasinghe, "Feasibility Study of a Novel Cross Assembled Multi-quadrotor Unmanned Aerial Vehicle," IEEE International Conference on Information and Automation for Sustainability (ICIAfS), 2018.

[14] D. Shatat and T. A. Tutunji, "UAV quadrotor implementation: A case study," IEEE 11th International Multi-Conference on Systems, Signals and Devices (SSD14), 2014.

[15] G. T. Poyi, M. H. Wu, A. Bousbaine and B. Wiggins, "Validation of a quad-rotor helicopter Matlab/Simulink and solidworks models," IET Conference on Control and Automation: Uniting Problems and Solutions, 2013

1. Robot Operating System(ROS) [↑](#footnote-ref-1)
2. Kernel [↑](#footnote-ref-2)
3. Operating System [↑](#footnote-ref-3)
4. Middleware [↑](#footnote-ref-4)
5. Open Source [↑](#footnote-ref-5)
6. Quadrotor [↑](#footnote-ref-6)
7. Micro electro mechanical systems (MEMS) [↑](#footnote-ref-7)
8. Quadcopter [↑](#footnote-ref-8)
9. Rotor [↑](#footnote-ref-9)
10. Analog [↑](#footnote-ref-10)
11. Digital [↑](#footnote-ref-11)
12. Simple Step Sequencer [↑](#footnote-ref-12)
13. Pneumatic Logic System [↑](#footnote-ref-13)
14. Electronic Sequencer [↑](#footnote-ref-14)
15. Micro Computer [↑](#footnote-ref-15)
16. Mini Computer [↑](#footnote-ref-16)
17. Pneumatique [↑](#footnote-ref-17)
18. GazeboSim [↑](#footnote-ref-18)
19. Frame [↑](#footnote-ref-19)
20. Electronic Speed Controller [↑](#footnote-ref-20)
21. flight controller [↑](#footnote-ref-21)
22. Radio control [↑](#footnote-ref-22)
23. Lithium-Polymer [↑](#footnote-ref-23)
24. Vertical Take off and Landing (VTOL) [↑](#footnote-ref-24)
25. **سیستم کنترل اطلاعات بی سیم از راه دور** [↑](#footnote-ref-25)
26. Aerodynamics [↑](#footnote-ref-26)
27. VMware [↑](#footnote-ref-27)
28. Turtlesim [↑](#footnote-ref-28)
29. Node [↑](#footnote-ref-29)
30. Client [↑](#footnote-ref-30)
31. Solidworks [↑](#footnote-ref-31)
32. Drowing [↑](#footnote-ref-32)
33. OpenGL [↑](#footnote-ref-33)