****

**دانشکدگان علوم و فناوری های نوین**

**گروه بین رشته‌ای فناوری، مکاترونیک**

**کنترل همکارانه سیستم چند رباته چرخدار مبتنی بر یادگیری با در نظر گرفتن قیود متغیرهای حالت و ورودی کنترلی**

**نام دانشجو:**

**علی صفرزاده**

**استادان راهنما:   
 دکتر بهرام تارویردی‌زاده**

**دکتر خلیل عالی‌پور**

**استاد مشاور:  
 دکتر آرش قمری**

**پایان نامه برای دریافت درجه کارشناسی ارشد  
 در رشته مهندسی مکاترونیک**

**دی ماه 1402**

****

****

**دانشکده علوم و فنون نوین**

**گروه بین رشته‌ای فناوری، مکاترونیک**

**کنترل همکارانه سیستم چند رباته چرخدار مبتنی بر یادگیری با در نظر گرفتن قیود متغیرهای حالت و ورودی کنترلی**

**نگارنده:  
علی صفرزاده**

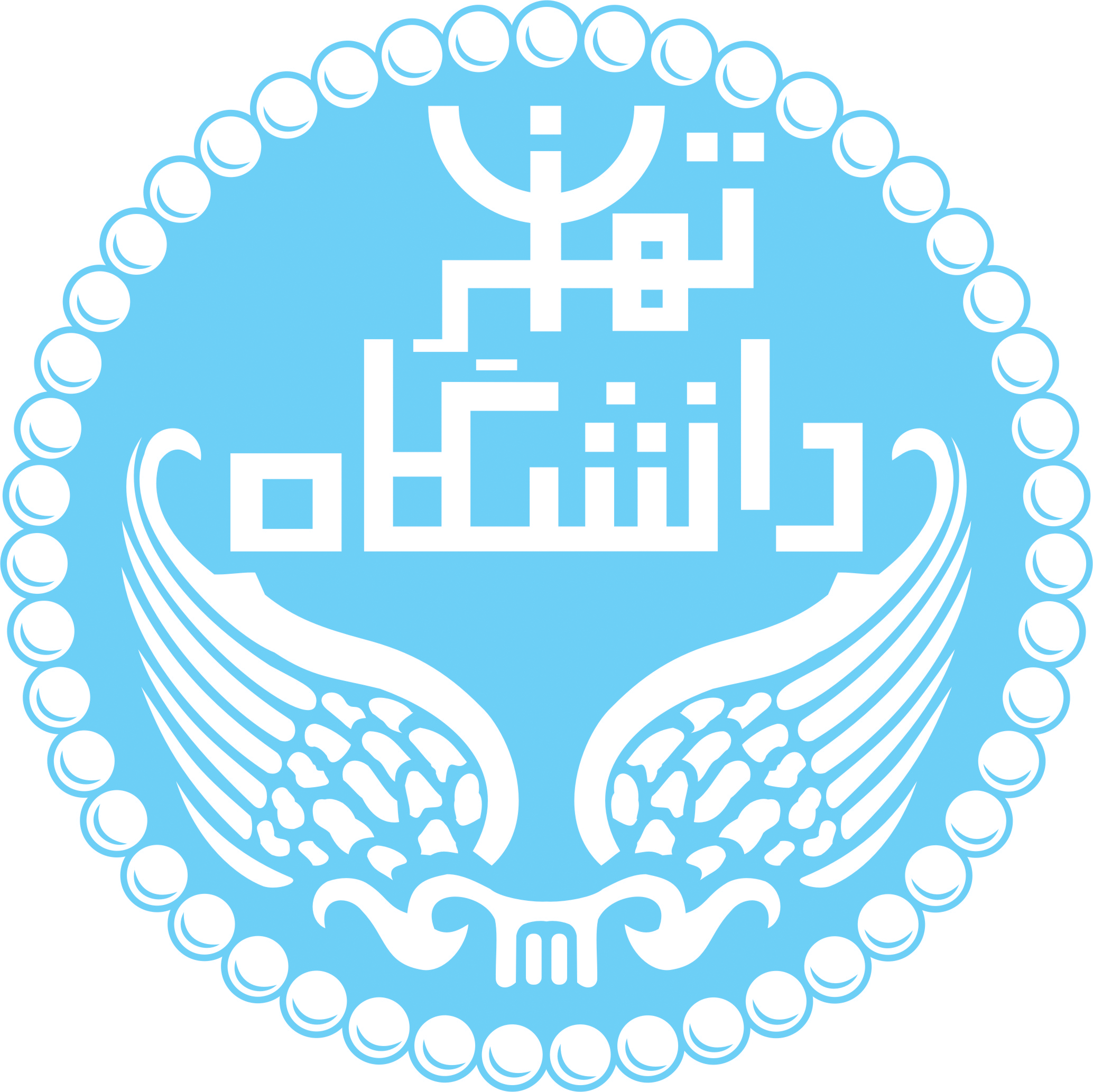
**استادان راهنما:   
دکتر بهرام تارویردی‌زاده**

**دکتر خلیل عالی‌پور**

**استاد مشاور:  
دکتر آرش قمری**

**پایان نامه برای دریافت درجه کارشناسی ارشد  
در رشته مهندسی مکاترونیک**

**بهمن ماه 1402**



**دانشگاه تهران**

**دانشکده علوم و فنون نوين**

**گروه** **بین رشته‌ای فناوری، مکاترونیک**

گواهي دفاع از پايان‌نامه كارشناسي ارشد

هيأت داوران پايان‌نامه كارشناسي ارشد ناپيوسته یا رساله دکتری

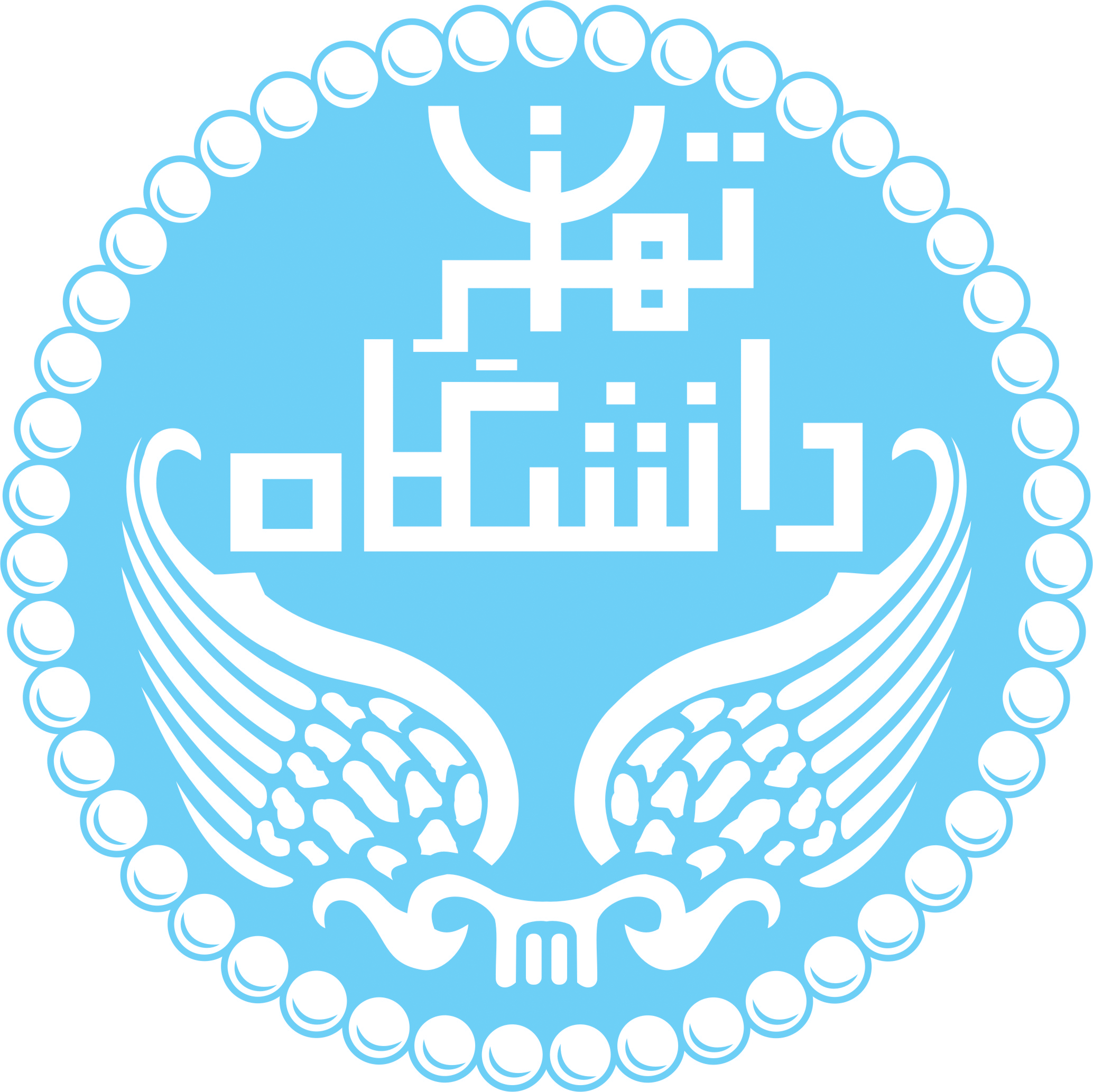
در رشته: **مهندسی مکاترونیک** گرايش:

با عنوان " **کنترل همکارانه سیستم چند رباته چرخدار مبتنی بر یادگیری با در نظر گرفتن قیود متغیرهای حالت و ورودی کنترلی**"

را در تاريخ با نمره نهايي به عدد به حروف

و درجه: ارزيابي نمود.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ردیف** | **عنوان هيأت داوران** | **نام و نام خانوادگي** | **مرتبه دانشگاهي** | **دانشگاه يا مؤسسه** | **امضاء** |
| 1 | استاد راهنما |  |  |  |  |
| 2 | استاد مشاور |  |  |  |  |
| 3 | استاد داور خارجی |  |  |  |  |
| 4 | استاد داور داخلی و نماينده كميته تحصيلات تكميلي |  |  |  |  |



**تعهد نامه اصالت اثر**

اینجانب علی صفرزاده دانش آموخته مقطع کارشناسی ارشد در رشتۀ مهندسی مکاترونیک که در تاریخ ........................... از پایان‌نامه یا رساله خود تحت عنوان: « کنترل همکارانه سیستم چند رباته چرخدار مبتنی بر یادگیری با در نظر گرفتن قیود متغیرهای حالت و ورودی کنترلی» با کسب درجۀ کارشناسي ارشد دفاع نموده‌ام، شرعاً و قانوناً متعهد می‌شوم:

1. مطالب مندرج در این پایان‌نامه یا رساله حاصل تحقیق و پژوهش اینجانب بوده و در مواردی که از دستاوردهای علمی‌و پژوهشی دیگران اعم از پایان‌نامه، کتاب، مقاله و غیره استفاده نموده‌ام، رعایت کامل امانت را نموده، مطابق مقررات، ارجاع و در فهرست منابع و مآخذ اقدام به ذكر آنها نموده‌ام.
2. تمامی ‌یا بخشی از این پایان‌نامه یا رساله قبلاً برای دریافت هیچ مدرک تحصیلی (هم سطح، پایین تر یا بالاتر) در سایر دانشگاه‌ها و مؤسسات آموزش عالی ارائه نشده است.
3. مقالات مستخرج از اين پایان‌نامه یا رساله کاملا حاصل کار اینجانب بوده و از هر گونه جعل داده و یا تغییر اطلاعات پرهیز نموده‌ام.
4. از ارسال همزمان و یا تکراری مقالات مستخرج از اين پایان‌نامه یا رساله (با بیش از 30 درصد همپوشانی) به نشریات و یا کنگره‌های گوناگون خودداری نموده و مي­نمایم.
5. کلیه حقوق مادّی و معنوی حاصل از این پایان‌نامه یا رساله متعلق به دانشگاه تهران بوده و متعهد می‌شوم هر گونه بهره مندی و یا نشر دستاوردهای حاصل از این تحقیق اعم از چاپ کتاب، مقاله، ثبت اختراع و غیره (چه در زمان دانشجویی و یا بعد از فراغت از تحصیل) با کسب اجازه از تیم استادان راهنما و مشاور و حوزه پژوهشی دانشکده باشد.

در صورت اثبات تخلف (در هر زمان) مدرك تحصيلي صادر شده توسط دانشگاه تهران از درجه اعتبار ساقط و اينجانب هیچگونه ادعایی نخواهم داشت.

امضا و نام و نام خانوادگی دانشجو: علی صفرزاده



تقدیم به پدر و مادر عزیزم که در فراز و نشیب زندگی، همواره پشتیبان و همراه من بوده‌اند.

**با تشکر و تقدیر از:**

**اساتید گرانقدرم، جناب آقای دکتر تارویردی‌زاده و جناب آقای دکتر خلیل عالی‌پور، که در مسیر اعتلای این تحقیق، بنده را بی دریغ یاری رساندند. امید که در تمامی مراحل زندگی در سایه سار لطف پروردگار باشند.**

**چکیده**

نقش ربات‌ها در زندگی انسان،‌ در بیش از دو دهه‌ی اخیر بسیار مورد توجه قرار گرفته و در حال افزایش است. از همین روی، دسته بندی‌های متنوعی برای علم رباتیک وجود دارد. یکی از مهمترین آن‌ها،‌ ربات‌های چرخدار است. همچنین با افزایش نقش ربات‌ها در زندگی بشر، کنترل آن‌ها و چالش‌های آن نیز طبعاً مورد بحث و بررسی قرار می‌گیرد. در این مبحث، به مفهوم کنترل خودکار ربات‌ها برمی‌خوریم که خود شامل الگوریتم‌های متعدد و پیچیدگی خاص خود است. بنابراین کنترل ربات‌های چرخدار که موضوع بحث این تحقیق است بسیار حائز اهمیت است. از سویی دیگر،‌ وظایفی وجود دارند که نیازمند همکاری بیش از یک ربات، جهت نیل به هدف می ‌باشد. در اینجا به مساله‌ی کنترل همکارانه‌ی ربات‌ها می‌رسیم که به طور کلی، چگونگی همکاری ربات‌ها در انجام یک وظیفه و چالش‌های کنترلی آن‌ را شامل می‌شود. همانطور که گفته شد تا بحال الگوریتم‌های بسیاری جهت کنترل ربات چرخدار ارائه شده‌اند که هر یک دارای نواقصی هستند. در این تحقیق سعی شده است با در نظر گرفتن این نواقص،‌ راه حل جدیدی برای یک سیستم جدید چند رباته ارائه گردد و بر روی پلتفرم‌های واقعی پیاده سازی شود. ابتدا برای یک سیستم تک پلتفرم مدل سینماتیکی و دینامیکی استخراج می‌شود. سپس مدل بدست آمده مورد شبیه سازی و تست قرار میگیرد. می‌دانیم که در این مدل عدم قطعیت وجود دارد. برای رسیدن به مدل دقیق تر و حل مشکل عدم قطعیت، با استفاده از شبکه عصبی RBF-NN به عنوان یک شبکه عصبی تخمینگر، مدل دینامیکی ربات در حال حرکت تخمین زده می‌شود و وزن‌های شبکه عصبی بهبود می‌یابند. حال این الگوریتم را برای سیستم چند رباته با ربات‌های همگن و یکسان تعمیم میدهیم و وزن‌های شبکه عصبی ربات‌ها بین آن‌ها به اشتراک گذاشته و رد و بدل می‌شوند. برای تک تک ربات‌های سیستم مسیر زمانی مرجع خاصی در نظر گرفته می‌شود، سپس با ارائه‌ی الگوریتم مناسب کنترلی همکارانه، و اعمال الگوریتم به سیستم، عمل دنبال کردن مسیر را بررسی می‌کنیم. برای اعمال الگوریتم کنترل همکارانه به سیستم چند رباته،‌ نیاز به تهیه و تولید پلتفرم‌های ربات چرخدار به تعداد ۳ عدد هست. بنابراین پس از ساخت پلتفرم‌ها، تمام الگوریتم را بر روی ۳ پلتفرم واقعی پیاده سازی و تست کرده، و نتایج بررسی می‌شوند.

**کلمات کلیدی:** سیستم چند رباته، شبکه عصبی تخمینگر، الگوریتم کنترلی همکارانه، پلتفرم ربات چرخدار، مدل دینامیکی غیر قطعی

**فهرست مطالب**

[1- فصل اول 5](#_Toc156402658)

[1˗1˗ مقدمه 6](#_Toc156402659)

[1˗2˗ انواع ربات‌های پایه متحرک چرخدار 7](#_Toc156402660)

[1˗2˗1˗ ربات‌های یک چرخ 7](#_Toc156402661)

[1˗2˗2˗ ربات‌های دو چرخ 8](#_Toc156402662)

[1˗2˗3˗ ربات‌های سه چرخ 9](#_Toc156402663)

[1˗2˗4˗ ربات‌های چهار چرخ 10](#_Toc156402664)

[1˗3˗ مرور ادبیات 13](#_Toc156402665)

[1˗3˗1˗ کنترل همکارانه‌ی سیستم چند رباته 13](#_Toc156402666)

[1˗3˗2˗ قیود حالت 14](#_Toc156402667)

[1˗4˗ نوآوری تحقیق 15](#_Toc156402668)

[1˗5˗ خلاصه‌ی فصول 15](#_Toc156402669)

[2- فصل دوم 17](#_Toc156402670)

[2˗1˗ مقدمه 18](#_Toc156402671)

[2˗2˗ تابع لیاپانوف 18](#_Toc156402672)

[2˗3˗ تابع لیاپانوف مانع 19](#_Toc156402673)

[2˗4˗ مدل سازی سینماتیکی 22](#_Toc156402674)

[2˗5˗ کنترلر سینماتیکی 23](#_Toc156402675)

[2˗6˗ شبیه سازی سینماتیکی 27](#_Toc156402676)

[2˗6˗1˗ دایره‌ی تک پلتفرم 27](#_Toc156402677)

[2˗6˗1˗1˗ مقادیر عددی 27](#_Toc156402678)

[2˗6˗1˗2˗ نتایج شبیه سازی 28](#_Toc156402679)

[2˗6˗2˗ حرکت ∞ 29](#_Toc156402680)

[2˗6˗2˗1˗ مقادیر عددی 29](#_Toc156402681)

[2˗6˗2˗2˗ نتایج شبیه سازی 30](#_Toc156402682)

[2˗6˗3˗ بیضی برای سه ربات 32](#_Toc156402683)

[2˗6˗3˗1˗ مقادیر عددی 32](#_Toc156402684)

[2˗6˗3˗2˗ نتایج شبیه سازی 34](#_Toc156402685)

[2˗7˗ نتیجه گیری 37](#_Toc156402686)

[3- فصل سوم 38](#_Toc156402687)

[3˗1˗ مقدمه 39](#_Toc156402688)

[3˗2˗ کنترل همکارانه 39](#_Toc156402689)

[3˗2˗1˗ قوانین حرکتی رینولدز 41](#_Toc156402690)

[3˗2˗2˗ گراف‌ها 41](#_Toc156402691)

[3˗2˗3˗ پیاده سازی قوانین رینولدز و نظریه‌ي وفاق 43](#_Toc156402692)

[3˗3˗ شبکه عصبی RBF 45](#_Toc156402693)

[3˗4˗ مدلسازی دینامیکی 47](#_Toc156402694)

[3˗5˗ کنترلر دینامیکی 50](#_Toc156402695)

[3˗6˗ کنترلر دینامیکی بر پایه‌ی یادگیری 55](#_Toc156402696)

[3˗7˗ 56](#_Toc156402697)

[4- فصل چهارم 58](#_Toc156402698)

[4˗1˗ مقدمه 59](#_Toc156402699)

[5- فصل پنجم 60](#_Toc156402700)

[5˗1˗ مقدمه 61](#_Toc156402701)

[6- مراجع 62](#_Toc156402702)

[7- واژه نامه 72](#_Toc156402703)

**فهرست شکل‌ها**

[شکل ‏1‑1 یک ربات تک چرخ Unicycle 10](#_Toc156131618)

[شکل ‏1‑2 ربات دو چرخ ساده 10](#_Toc156131619)

[شکل ‏1‑3 ربات سه چرخ 3-wheel drive 11](#_Toc156131620)

[شکل ‏1‑4 ربات سه چرخ 2-wheel drive 12](#_Toc156131621)

[شکل ‏1‑5 ربات چهار چرخ با ۴ موتور مجزا 13](file:///C:\Users\User\Documents\GitHub\TrajectoryTracking\نگارش%20پایان%20نامه\1%20پایان%20نامه%20علی%20صفرزاده.docx#_Toc156131622)

[شکل ‏1‑6 ربات چهار چرخ با سیستم حرکتی تانکی 14](file:///C:\Users\User\Documents\GitHub\TrajectoryTracking\نگارش%20پایان%20نامه\1%20پایان%20نامه%20علی%20صفرزاده.docx#_Toc156131623)

[شکل ‏1‑7 ربات چهار چرخ با سیستم حرکتی دو موتور و فرمان 14](file:///C:\Users\User\Documents\GitHub\TrajectoryTracking\نگارش%20پایان%20نامه\1%20پایان%20نامه%20علی%20صفرزاده.docx#_Toc156131624)

[شکل ‏2‑1 تابع لیاپانوف مانع. سمت راست نامتقارن. سمت چپ متقارن 21](#_Toc156131625)

[شکل ‏2‑2 تابع مربع تانژانت که به ازای به بینهایت میل می‌کند 22](#_Toc156131626)

[شکل ‏2‑3 تابع مربع لگاریتم که به ازای x = 0 به بینهایت میل می‌کند 22](#_Toc156131627)

[شکل ‏2‑4 نمونه‌ی یک پلتفرم ربات چرخ دار غیر هولونومیک با سه چرخ 23](#_Toc156131628)

[شکل ‏2‑5 مسیر حرکت ربات سمت راست مختصات واقعی سمت چپ مختصات تحت تبدیل همومورفیسم 29](#_Toc156131629)

[شکل ‏2‑6خطای حالت‌های تبدیل شده 29](#_Toc156131630)

[شکل ‏2‑7 خطای حالت‌ها در مختصات واقعی 30](#_Toc156131631)

[شکل ‏2‑8 مسیر حرکت ربات سمت راست مختصات واقعی سمت چپ مختصات تحت تبدیل همومورفیسم 31](#_Toc156131632)

[شکل ‏2‑9 تغییرات زاویه‌ی دلخواه و واقعی 31](#_Toc156131633)

[شکل ‏2‑10 تغییرات زاویه دلخواه و واقعی تا زمان 10 ثانیه 32](#_Toc156131634)

[شکل ‏2‑11 خطای حالت‌های تبدیل شده 32](#_Toc156131635)

[شکل ‏2‑12 خطای حالت‌ها در مختصات واقعی 33](#_Toc156131636)

[شکل ‏2‑13 مسیر حرکت ربات 1 سمت راست مختصات واقعی سمت چپ مختصات تحت تبدیل همومورفیسم 35](#_Toc156131637)

[شکل ‏2‑14 مسیر حرکت ربات 2 سمت راست مختصات واقعی سمت چپ مختصات تحت تبدیل همومورفیسم 35](#_Toc156131638)

[شکل ‏2‑15 مسیر حرکت ربات 3 سمت راست مختصات واقعی سمت چپ مختصات تحت تبدیل همومورفیسم 36](#_Toc156131639)

[شکل ‏2‑16 تغییرات زاویه دلخواه و واقعی ربات 1 تا زمان 10 ثانیه 36](#_Toc156131640)

[شکل ‏2‑17 تغییرات زاویه دلخواه و واقعی ربات 2 تا زمان 10 ثانیه 37](#_Toc156131641)

[شکل ‏2‑18 تغییرات زاویه دلخواه و واقعی ربات 3 تا زمان 10 ثانیه 37](#_Toc156131642)

[شکل ‏2‑19 خطای حالت‌های هر سه ربات در مختصات واقعی 38](#_Toc156131643)

[شکل ‏3‑1حرکت جمعی یک دسته پرنده‌ی سار 41](#_Toc156131644)

[شکل ‏3‑2 حرکت جمعی یک دسته ماهی 41](#_Toc156131645)

[شکل ‏3‑3گراف بدون جهت Undirected Graph 44](#_Toc156131646)

[شکل ‏3‑4 گراف جهت دار Directed Graph 44](#_Toc156131647)

[شکل ‏3‑5 ساختار شبکه عصبی RBF 46](#_Toc156131648)

[شکل ‏3‑6 تابع RBF 47](#_Toc156131649)

**فهرست جداول**

[جدول ‏3‑1 نمادها و پارامترهای فیزیکی مدل دینامیکی 49](#_Toc156131760)

# فصل اول

کلیات تحقیق

## مقدمه

امروزه مبحث رباتیک و استفاده از ربات، برای بهبود بخشی به زندگی انسان و سهولت بیشتر انجام کارها، خصوصا مواردی که موجب آسیب فیزیکی به انسان میشود، یا انجام آن برای انسان دشوار است، بسیار مورد توجه و در حال پیشرفت و گسترده‌تر شدن می‌باشد. یکی از زیر مجموعه‌های مهم رباتیک، بخش ربات‌های چرخدار وکنترل آنهاست. در این تحقیق، مسأله، چگونگی کنترل یک سیستم چند رباته[[1]](#footnote-1) به صورت همکارانه[[2]](#footnote-2) می‌باشد به طوری که ربات‌ها بتوانند مسیر مورد نظر را با دقت مناسب دنبال کنند[[3]](#footnote-3). این همکاری توسط یک شبکه[[4]](#footnote-4) با جابجایی اطلاعات بین ربات‌ها تعریف می‌شود. می‌دانیم که در مدلسازی و اندازه‌گیری پارامترهای فیزیکی ربات عدم قطعیت[[5]](#footnote-5) وجود دارد. چرا که دقت اندازه گیری تابع مستقیمی از ابزار اندازه گیری و دقت شخص اندازه گیرنده است. بنابراین به طور کلی عدم قطعیت در مدل دینامیکی به معنای عدم دقت در اندازه گیری پارامترهای فیزیکی (مانند جرم ربات، ممان اینرسی،‌ شعاع چرخ‌ها و...) می‌باشد. فلذا برای کنترل بهتر ربات، باید کنترلر به گونه‌ای طراحی شود که بر این مشکل چیره شود. جهت تخمین دینامیک ربات و چیره‌گی بر عدم قطعیت در مدل سیستم (عدم دقت در اندازه گیری پارامترهای فیزیکی در مدلسازی دینامیکی) از شبکه عصبی RBF-NN که نوعاً یک شبکه عصبی تخمینگر است استفاده میشود. به این صورت که در حالت معمول، یک مدلسازی دینامیکی از ربات صورت می‌پذیرد که شامل پارامترهای فیزیکی است. سپس با اندازه گیری این پارامترها،‌ مدل دینامیکی به صورت عددی (و با عدم قطعیت) بیان می‌شود و مسلما کنترلر طراحی شده بر پایه‌ی این مدل دقیق نیست. اما در این روش پس از اندازه گیری پارامترها و بیان به صورت عددی، بخش اعظم مدل دینامیکی (شامل ماتریس جرم و ممان، ماتریس گشتاور گرانشی، ماتریس تبدیل سرعت و...) توسط RBF-NN تخمین زده میشود و مدل دینامیکی ربات را تکمیل می‌کند. ورودی این شبکه عصبی مکان، سرعت و شتاب لحظه‌ای ربات است که با پس‌خور بدست می‌آیند و خروجی آن تکمیل مدل دینامیکی است[1].

هدف از انجام این تحقیق حصول و تولید یک کنترل کننده است که ربات‌ها در سیستم چند رباته بتوانند با دقت بالایی مسیر مورد نظر را دنبال کنند و این الگوریتم بر روی یک سیستم ۳ رباته واقعی پیاده سازی شود.

در دسته بندی ربات‌ها، می‌توان به سه دسته‌ی عمده اشاره نمود. ربات‌های پرنده، ربات‌های متحرک در آب، و ربات های پایه متحرک. که ربات‌های پایه متحرک را نیز به دو دسته‌ی کلی ربات‌های پایه متحرک دارای پا[[6]](#footnote-6) و ربات‌های پایه متحرک چرخ‌دار[[7]](#footnote-7)، می‌توان تقسیم نمود. از این میان، ربات‌های پایه متحرک چرخدار پرکاربردترین و از نگاهی ساده ‌ترین ربات‌ها هستند. این ربات‌ها مدل سینماتیکی و دینامیکی ساده‌تری دارند و همچنین برای کنترل کردن آن‌ها، با توجه به سلب بودن زمین، چالش‌های حرکتی در آب و در هوا برای آن‌ها مطرح نمی‌شود. و نیز مصرف انرژی برای جابجا شدن آنها کمینه‌است و معمولاً سرعت حرکت آنها از نوع‌های دیگر ربات‌های پایه متحرک بیشتر است.

## انواع ربات‌های پایه متحرک چرخدار

ربات‌های پایه متحرک چرخدار انواع مختلفی دارند و به تبع، مدلسازی آنها نیز متفاوت می‌شود. در این قسمت به تعریف و تشریح انواع زیر می‌پردازیم:

* ربات‌های یک چرخ
* ربات‌های دو چرخ
* ربات‌های سه چرخ
* ربات‌های چهار چرخ

### ربات‌های یک چرخ

به دلیل آنکه پایدار سازی و کنترل ربات‌های تک چرخ ساده نیست، این نوع ربات‌‌‌ها کم کاربردتر هستند. اما به لحاظ ساخت، مقرون به صرفه‌تر هستند و هزینه‌ی کمتری دارند. به دلیل آنکه نقطه‌ی تماس ربات تک چرخ با زمین، تنها یک نقطه است، حفظ تعادل، پایدار سازی و کنترل آن دشوار است. با این اوصاف استفاده از یک چرخ کروی برای حفظ تعادل و امکان حرکت در تمامی جهات، می‌تواند گزینه‌ی مناسبی باشد. با این حال رواج استفاده از چرخ استوانه‌ای و تایر بیشتر است. به این علت که حرکت دادن تایر توسط یک موتور بسیار ساده تر است. در شکل زیر یک نمونه ربات تک چرخ دیده می‌شود که تعادل خود را حفظ نموده است.

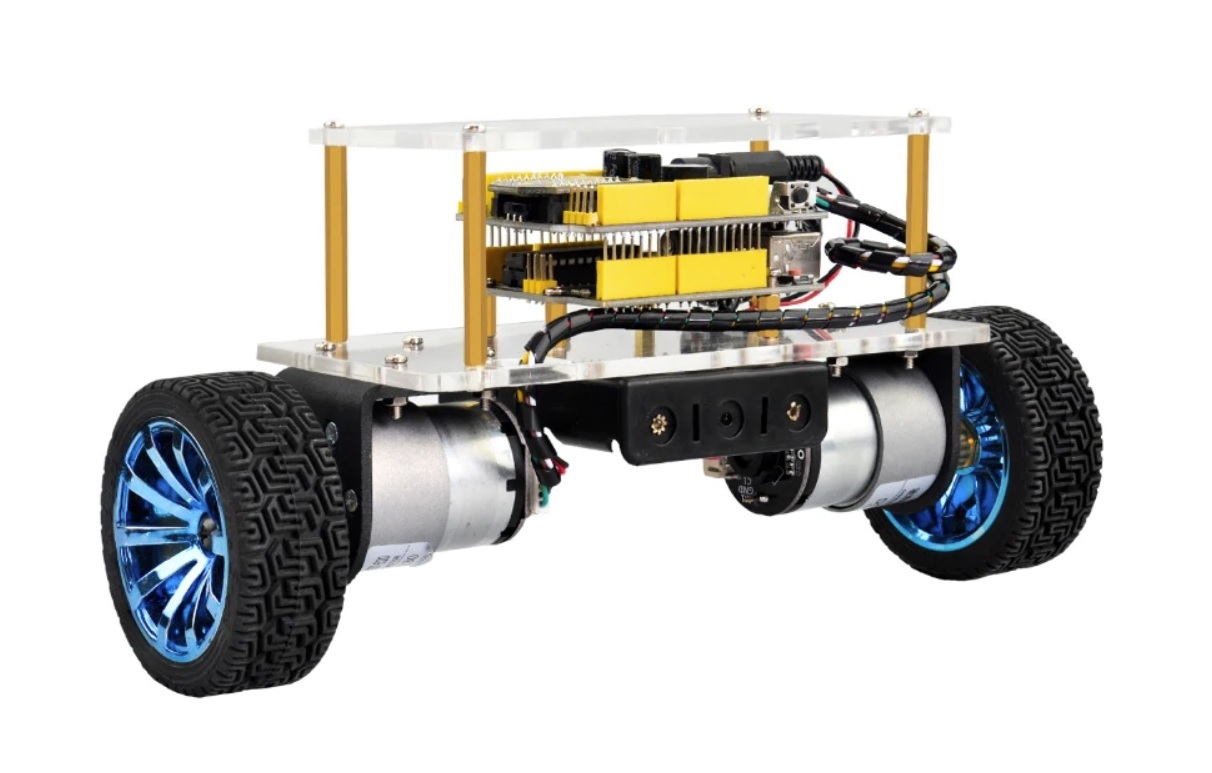


شکل ‏1‑1 یک ربات تک چرخ Unicycle

### ربات‌های دو چرخ

در نگاه اول شاید کنترل ربات دو چرخ ساده به نظر بیاید. اما در عمل مبحث دشواری ست. ربات دو چرخ یک نوع پلتفرم با دو چرخ موازی زیر ربات است که مدار و متعلقات مکانیکی بر روی آن دو سوار هستند. برای حفظ تعادل و پایدار سازی ربات دو چرخ عموماً مرکز ثقل آنرا پایین و در یک نقطه نگه می‌دارند. برای این منظور معمولاُ باتری یا دیگر متعلقات با وزن بیشتر را پایین‌تر قرار می‌دهند. حرکت ربات دو چرخ، ساده است و برای حرکت مستقیم کافی ست هر دو چرخ با دور بر دقیقه[[8]](#footnote-8) یکسان حرکت کنند. و اگر یک چرخ با سرعت بیشتر از چرخ دیگر بچرخد، ربات حرکت دورانی در جهت مقابل اجام می‌دهد.

شکل زیر یک نمونه ربات دو چرخ را نمایش می‌دهد.



شکل ‏1‑2 ربات دو چرخ ساده

### ربات‌های سه چرخ

ربات‌های سه چرخ شاید از رایج ترین انواع ربات‌های چرخدار باشند. چرا که سرعت حرکت این نوع در تمامی جهات از دو نوع قبلی بیشتر بوده، و همچنین حفظ تعادل آنها مساله‌ی چالش برانگیزی نیست. در این نوع ربات‌ها، دو دسته بندی وجود دارد. دسته‌ی اول ربات‌هایی هستند که هر به سه چرخ موتور متصل بوده و کار انقال قدرت صورت می‌پذیرد. مانند شکل زیر



شکل ‏1‑3 ربات سه چرخ 3-wheel drive

نوع دیگر ربات‌های سه چرخ، که پلتفرم استفاده شده در این تحقیق نیز از همین مدل است، نوعی ست که دو چرخ متصل به موتور در عقب ربات قرار دارند. و یک چرخ هرزگرد که وظیفه‌ی حفظ تعادل ربات را بر عهده دارد، در جلوی ربات قرار دارد. مانند شکل زیر



شکل ‏1‑4 ربات سه چرخ 2-wheel drive

### ربات‌های چهار چرخ

رواج ربات‌های چهار چرخ، از ربات‌های سه چرخ نیز بیشتر است. به لحاظ پایداری و کنترل پذیری، این ربات‌ها دارای ارجحیت هستند. همچنین برای تعیین مرکز ثقل ربات چهار چرخ چالش کمتری وجود دارد چرا که ربات در چهار نقطه روی زمین قرار دارد. از معایب ربات چهار چرخ اما می‌توان به بحث هزینه اشاره کرد. با توجه به اینکه ربات چهار چرخ دارای چهار چرخ و بعضاً چهار موتور می‌باشد، به لحاظ هزینه‌ی ساخت از ربات‌های ساده‌تر در رده‌ی پایین تری قرار می‌گیرد.

برای دسته بندی ربات‌های چهار چرخ از لحاظ کنترل چرخ‌ها و انتقال قدرت، می‌توان آنها را به سه دسته‌ی کلی تقسیم نمود.

دسته اول به این صورت است که برای انتقال توان، از هر ۴ چرخ استفاده می‌کنیم و همه‌ی چرخ‌ها را به موتور متصل می‌کنیم. در این حالت کنترل و جهت دهی به ربات به این صورت است که برای چرخش به سمت چپ، دور موتورهای راست جلو و راست عقب را افزایش داده، و برای چرخش به سمت چپ دور موتورهای چپ جلو و چپ عقب را زیاد می‌کنیم. البته لازم به ذکر است که برای کنترل مناسب ربات، می‌بایست کنترل روی هر چرخ دقیق صورت پذیرد که از لغزش اضافی و ناخواسته جلوگیری شود.

موتور

شکل ‏1‑5 ربات چهار چرخ با ۴ موتور مجزا

دسته‌ی دوم به صورتی‌ست که چرخ‌های کناری به یکدیگر وصل هستند هر دو چرخ به یک موتور متصل اند. به این سیستم انتقال قدرت و پیکربندی موتورها، سیستم کنترل تانکی می‌گویند. بنابراین همانطور که در شکل 6 مشخص است، در این سیستم حرکتی تنها دو موتور وجود دارد. موتور سمت راست و موتور سمت چپ. برای حرکت دو چرخ سمت راست موتور سمت راست، و برای حرکت دو چرخ سمت چپ موتور سمت راست را درایو میکنیم. همچنین برای گردش به راست دور موتور سمت چپ را زیاد کرده، و برای گردش به چپ دور موتور سمت راست را زیاد می‌کنیم.

سیستم حرکتی تانکی برای چرخش درجا با سرعت بالا بسیار مناسب است. همچنین برای کنترل چرخ‌ها، نسبت به حالت پیشین،‌ لغزش ناخواسته‌ی کمتری را در عمل خواهد داشت.

دو چرخ پشت هم را به وسیله‌ی تسمه و یا گیربکس و... به یکدیگر می‌توان متصل کرد. از دلایل استفاده از این سیستم و متصل کردن چرخ‌های کناری به هم نیز به ‌این می‌توان اشاره نمود که در هنگام بالاروى از صخره­ها و موانع، بالارفتن يك چرخ موجب بالا رفتن چرخ ديگر مى­شود. به همين دليل در طراحى بدنه بايد موارد خاصی را در نظر گرفت.

موتور

شکل ‏1‑6 ربات چهار چرخ با سیستم حرکتی تانکی

دسته سوم به اين صورت است كه دو چرخ عقب كار انتقال قدرت و درایو را انجام مى­دهند و دو چرخ جلو به صورت هرزگرد به اين طرف و آن طرف كشيده مى شوند. در حقیقت، مانند سیستم حرکتی خودرو، دو چرخ جلو نقش جهت دهی را دارند. در این حالت برای جهت دهی می‌توان از اتصال موتورهایی مانند سروو موتور به دو چرخ جلویی استفاده کرد.

موتور

شکل ‏1‑7 ربات چهار چرخ با سیستم حرکتی دو موتور و فرمان

## مرور ادبیات

### کنترل همکارانه‌ی سیستم چند رباته

سیستم چند رباته[[9]](#footnote-9)،‌ سیستمی است که در آن تمام ربات‌ها با تبادل اطلاعات با یکدیگر همکاری می‌کنند تا وظایف کنترلی مختلفی را به انجام برسانند، چشم اندازهای کاربردی گسترده‌ای دارد. به دلیل کارآیی بالاتر، انعطاف بیشتر و تطابق با محیط‌های ناشناخته، و قابلیت همکاری، سیستم چند رباته می‌تواند به طور مناسب از منابع استفاده کند، قابلیت اتکا به سیستم را افزایش دهد، و کیفیت اجرای وظایف را نسبت به یک سیستم تک ربات بهبود بخشد. در یک سیستم همکارانه چند رباته‌ي متحرک، وظیفه و پیچیدگی تک تک ربات‌ها متمرکز، و تقویت می‌شود. بنابراین، همکاری بین ربات‌هاست که توانایی سیستم را بیان می‌کند. این مزایای مرتبط با سیستم‌های چند رباته مبتنی بر همکاری بین ربات‌ها، توجه محققین را بسیار افزایش داده و در سال‌های اخیر تلاش‌های زیادی برای تولید روش‌های کنترل همکارانه چند رباتی شده است. [1-6].

در بحث تعقیب مسیر زمانی[[10]](#footnote-10)، در کنترل سیستم چند رباته‌ی متحرک، پیشرفت‌های بسیاری حاصل شده است [7-16]. به ویژه، نویسندگان مقاله [7] چارچوب[[11]](#footnote-11) جدیدی برای مدل‌سازی فرم‌دهی[[12]](#footnote-12) ربات‌های متحرک بر اساس نظریه گراف ایجاد کردند و تغییر فرم‌دهی را به تغییر ساختار گراف مرتبط کردند. در [9-11]، کنترل فرم‌دهی چند ربات متحرک غیرهولونومیک، به یک مسئله‌ی وفاق[[13]](#footnote-13) در حالت‌های سیستم تبدیل شده و سپس کنترلر سینماتیک توزیع‌شده و کنترلر دینامیکی تطابقی طراحی شده است، تا تمام ربات‌ها به سمت به مسیرهای زمانی مرجع حرکت کنند و به طور مجانبی[[14]](#footnote-14) به الگوی هندسی مطلوب همگرا شوند. با وجود تحقیقات متعدد، هنوز چالش‌های زیادی باقی مانده است، یکی از آن‌ها نامعلومی[[15]](#footnote-15) یا عدم قطعیت مدل غیرخطی است. عدم قطعیت‌های مدل‌سازی نقش بسیار مخربی در سیستم‌های کنترل توزیع‌شده غیرخطی[[16]](#footnote-16) دارند و باعث می‌شوند کنترل حرکتی دشوارتر شود. در [17] و [18]، فرض شده است که مدل سیستم معلوم است و عدم قطعیت غیرخطی مد نظر گرفته نشده است. در [19]، روشی برای طراحی کنترلر دینامیکی ارائه شده است که زمانی که دینامیک ربات معلوم است، و ورودی کنترلی گشتاور دنبال‌کننده، شامل دینامیک خود و دینامیک رهبر گروه[[17]](#footnote-17) است. در [20-22]، کنترل فرم‌دهی توزیع‌شده ربات متحرک چرخدار غیر هولونومیک با استفاده از شبکه عصبی انجام شده‌است. به طوریکه برای حل دینامیک بدون ساختار و مدل‌نشدهی سیستم ربات و تخمین آن از شبکه‌ی عصبی استفاده شده است.

سیستم‌های ربات‌های متحرک معمولاً وظایف نسبتاً ساده اما تکراری انجام می‌دهند. از دیدگاه توسعه فناوری، بهبود کارایی این سیستم‌ها و کاهش مصرف انرژی آن‌ها مبحثی مهم و غیرقابل چشم پوشی است. استفاده‌ی منطقی و مؤثر از دانش به دست آمده از فرآیند کنترل، می‌تواند از بسیاری از رفتارهای ناموثر جلوگیری کرده و وظایف کنترلی را به بهترین شکل اجرا کند، به عبارت دیگر، توسعه‌ی روش‌های کنترل مبتنی بر یادگیری می‌تواند در کاربردهای عملی مهندسی مزایای اجتماعی و اقتصادی بسیاری داشته باشد. در روش‌های کنترل مبتنی بر یادگیری ذکر شده، با وجود اینکه مساله‌ی وفاق بین همه‌ی ربات‌ها قابل پیاده سازی‌ست، اطلاعات وزن‌های شبکه عصبی در فرایند کنترل به طور کامل مورد بررسی قرار نمی‌گیرد. نظریه یادگیری تعیینی[[18]](#footnote-18) با استفاده از شبکه عصبی RBF به طور کامل مورد تحقیق واقع شده است. مانند [23-26]، جایی که دینامیک حلقه بسته‌ی ناشناخته‌ی سیستم غیرخطی، می‌تواند به دقت توسط شبکه عصبی RBF تقریب زده شود و شبکه‌ی آموزش دیده می‌تواند در کنترل یادگیری مجدد استفاده شود. با این حال، وقتی که نظریه‌ی یادگیری تعیینی در سیستم‌های چند رباته استفاده می‌شود، یادگیری شبکه‌های عصبی به طور کاملاً توزیع‌شده اتفاق می‌افتد، یعنی هر ربات وزن‌های شبکه‌ی عصبی خود را با دیگران به اشتراک نمی‌گذارد. در واقع هر ربات به صورت مستقل یادگیری تخمین سیستم غیرخطی را انجام می‌دهد. شبکه‌ی عصبی هر ربات تنها برای مسیر زمانی مرجع خاصی که به هر ربات اختصاص داده شده است، قابل استفاده است، که منجر به این می‌شود که تعمیم مدل شبکه عصبی آموزش دیده محدود شود. از [27] الهام گرفته شده است که مزیت توان یادگیری شبکه عصبی می‌تواند استفاده شود. زمانی که به ربات‌ها اجازه‌ی به اشتراک گذاشتن وزن‌های عص داده شود می‌توان این مزیت را دید. فضای تخمین توسط شبکه عصبی، در حالی می‌تواند گسترش یابد که وزن‌های تمام ربات‌ها به یک مقدار بهینه‌ی مشترک همگرا شوند.

### قیود حالت

به علاوه، با توجه به ایمنی در عمل، محدودیت‌های فیزیکی موتورها، امکان ساخت مکانیکی و سایر عوامل، معمولاً در سیستم کنترلی واقعی محدودیت‌های[[19]](#footnote-19) مختلفی ایجاد می‌شود. با توجه ملاحظات عملکرد و ایمنی سیستم ربات یاد شده، به محدود کردن حوزه عملیاتی برخی متغیرهای کلیدی (مانند متغیرهای خروجی سیستم)، قیود حالت[[20]](#footnote-20) گفته می‌شود. نادیده گرفتن قیود حالت ممکن است به تخریب سیستم به طور جدی، ناپایداری سیستم و حتی خرابی تجهیزات منجر شود. برای ارايه‌ی کنترلر با در نظر گرفتن قیود حالت در سیستم‌های غیرخطی، برخی از روش‌های مناسب پیشنهاد شده‌اند، مانند کنترل پیش‌بینی مدل[[21]](#footnote-21) [28، 29]، تابع تبدیل خطا[[22]](#footnote-22) [30] و تابع لیاپانوف مانع[[23]](#footnote-23) (BLF)[31]. برای سیستم‌های غیرخطی پس‌خور[[24]](#footnote-24) دار با قیود خروجی، یک روش کنترل با استفاده از تابع تانژانت محور tan-BLF در [32] پیشنهاد شده است تا دنبال کردن بهینه برای مسیر زمانی مرجع بدون نقض قیود را تضمین کند. در [33]، یک کنترل کننده شبکه عصبی تطابقی با استفاده از روش تابع BLF برای حل مساله‌ی کنترلی دنبال کردن مسیر ربات n-link، با در نظر گرفتن قیود تمامی حالت‌های سیستم عدم قطعیت در مدلسازی ارائه شده است. در کنترل همکارانه‌ی چند ربات، اگر دنبال کردن مسیر به طور پایدار و با رعایت قیود حالت هر تک ربات ممکن شود، عملکرد کنترلی سیستم ربات‌ها می‌تواند به طور چشمگیری بهبود یابد. بدین ترتیب از برخورد ربات‌ها با یکدیگر و این قبیل حوادث می‌توان جلوگیری نمود.

## نوآوری تحقیق

نوآوری این تحقیق شامل موارد پیش رو است. ابتدا از روش‌های ارائه‌ی کنترلر در تحقیقات پیشین استفاده شده و برای مسیرهای زمانی مرجع پیچیده مثل حرکت ∞ مانند، شبیه سازی آورده شده و جامعیت کنترلر ارائه شده را افزایش داده است. سپس پیاده سازی[[25]](#footnote-25) فیزیکی یک کنترلر ارائه شده در تحقیقات پیشین در زمینه‌ی کنترل همکارانه‌ی سیستم چند رباته‌ی چرخدار انجام پذیرفته است. این پیاده سازی بر روی 3 پلتفرم چرخدار دارای 3 چرخ انجام شده است. و نیز صحت عملکرد فیزیکی این سیستم سه رباته بررسی و خطای حرکتی آن‌ها آورده شده است. نهایتاً با توجه به اینکه برای دریافت مکان حرکت ربات نیازمند پس‌خور از مختصات ربات بر روی زمین هست، و شرایط نوری در محیط آزمایشگاهی می‌تواند تغییر کند، از روشی مناسب برای تفکیک رنگ و مکان یابی ربات‌ها با استفاده از پردازش تصویر بهره برداری شده است.

## خلاصه‌ی فصول

با توجه به بحث‌های فوق، در این تحقیق به ارایه‌ی کنترلری همکارانه مبتنی بر یادگیری دینامیک غیرخطی، برای تعقیب مسیر زمانی چند ربات چرخدار یکسان با در نظر گرفتن قیود حالت پرداخته شده است. در این تحقیق، تمام ربات‌ها در سیستم چند رباته به عنوان سیستم‌های غیر قطعی[[26]](#footnote-26) یکسان در نظر گرفته می‌شوند و هر ربات مسیر زمانی مرجع مختص به خود را دارد. با استفاده از شبکه‌ی عصبی با تابع فعالیتRBF[[27]](#footnote-27) برای تخمین دینامیک غیرخطی ناشناخته، یک کنترلر دینامیکی همکارانه ارائه خواهد شد.

در فصل دوم به مدلسازی سینماتیکی ربات، ارائه‌ی کنترلر سینماتیکی به همراه توضیح مفهوم تابع لیاپانوف مانع و شبیه سازی مرتبط با اعمال کنترلر به شرایط سینماتیکی ربات می‌پردازیم.

سپس در فصل سوم مبتنی بر تابع لیاپانوف مانع، با استفاده از نظریه‌ی وفاق، و توضیح توپولوژی شبکه‌ی ارتباطی بین ربات‌ها، کنترلر دینامیکی را ارائه نموده و تخمینگر شبکه عصبی RBF برای تخمین دینامیک ربات را معرفی می‌کنیم. سپس نتایج شبیه سازی حاصل از اعمال کنترلر به دینامیک ربات را ارائه میکنیم.

در فصل چهارم به طراحی مکانیکی و الکترونیکی ۳ پلتفرم ربات چرخدار، و چگونگی ایجاد پس‌خور مکان بر پایه‌ی پردازش تصویر و پیکربندی رنگی هر پلتفرم توضیح داده خواهد شد.

و در فصل پنجم نتایج حاصل شده از پیاده سازی الگوریتم بر روی ۳ پلتفرم ربات چرخدار در حرکت واقعی آن‌ها ارائه می‌شود.

# فصل دوم

**طراحی کنترلر سینماتیکی**

## مقدمه

در این بخش از پایان نامه، ابتدا مفهوم تابع لیاپانوف و چگونگی اثبات پایداری و سپس تابع لیاپانوف مانع توضیح داده می‌شود. این توابع برای رعایت قیود حالت بسیار پرکاربرد و ملزوم هستند. در ادامه برای ارائه‌ی کنترلر سینماتیکی و همچنین دینامیکی، از این توابع استفاده خواهیم نمود. سپس برای یک پلتفرم ربات چرخدار دارای سه چرخ، مدلسازی سینماتیکی را ارائه می‌کنیم. این مدلسازی به طور کامل سرعت خطی و دورانی ربات را با مکان مورد نظر در صفحه‌ی مختصات دکارتی، پیوند می‌دهد. سپس برای این مدل یک کنترلر سینماتیکی ارائه نموده و با استفاده از شبیه سازی و تابع لیاپانوف مانع،‌ پایداری سیستم را اثبات نموده و در نهایت نتایج شبیه سازی ارائه می‌شوند.

## تابع لیاپانوف

در نظریه‌ی معادلات دیفرانسیل عادی یا ODE، توابع لیاپانوف، توابعی اسکالر هستند که می‌توان از آن‌ها برای اثبات پایداری نقطه‌ی تعادل[[28]](#footnote-28) ODE استفاده نمود. این معادلات ODE می‌توانند همان معادلات فضای حالت سیستم و بیان کننده‌ی رفتار سیستم باشند. این توابع به نام که به نام الکساندر لیاپانوف[[29]](#footnote-29) شناخته می‌شوند. استفاده از توابع لیاپانوف شامل دو روش اصلی است، که روش دوم برای اثبات پایداری مرسوم و جامع تر است. توابع لیاپانوف در تئوری پایداری سیستم‌های دینامیکی و تئوری کنترل اهمیت بسیاری دارند. چرا که اثبات پایداری مبحث مهمی در کنترل به شمار می‌رود. همچنین یک مفهوم مشابه در نظریه‌ی زنجیره‌های مارکوف[[30]](#footnote-30) برای فضای حالت[[31]](#footnote-31) عمومی نیز وجود دارد، که معمولاً با نام توابع فاستر-لیاپانوف شناخته می‌شود.

برای انواع خاصی از ODEها، یا همان معادلات فضای حالت سیستم‌ها، وجود توابع لیاپانوف شرط لازم و کافی برای پایداری است. در حالی که هیچ روش عمومی برای ساخت توابع لیاپانوف برای معادلات سیستم وجود ندارد، در بسیاری از موارد خاص، ساخت توابع لیاپانوف دشوار نیست. به عنوان مثال، برای سیستم‌های با یک حالت، انتخاب توابع چند جمله‌ای کافی ست. همچنین حل یک نامساوی از یک ماتریس خطی خاص، تابع لیاپانوف را برای سیستم‌های خطی می‌سازد، و در آخر استفاده از قوانین حفظ[[32]](#footnote-32) معمولاً می‌تواند برای ساخت توابع لیاپانوف برای سیستم‌های فیزیکی استفاده شود.

اگر سیستم ما به صورت زیر باشد

|  |  |
| --- | --- |
| رابطه 2-1 |  |

آنگاه اگر نقطه‌ی x = 0، نقطه‌ی تعادل سیستم باشد تابع لیاپانوف V(x) به صورتی تعریف می‌شود که پیوسته،‌ دارای مشتق مرتبه اول، و برای x ≠ 0 همیشه مثبت است. همچنین شرط زیر به عنوان تابع نامحدود شعاعی مطرح است.

|  |  |
| --- | --- |
| رابطه 2-2 |  |

حال چگونگی استفاده از تابع کاندید لیاپانوف برای اثبات پایداری به این صورت است که اگر تابع کاندید لیاپانوف با شرایط بالا انتخاب شد، حال می‌بایست تابع مشتق بر حسب زمان آن بدست آید. اگر این تابع مشتق زمانی، به ازای هر x در محدوده‌ی تعریف x،‌ منفی معین باشد، سیستم ما در نقطه‌ی تعادل پایدار مجانبی عام سراسری[[33]](#footnote-33) یا پایدار لیاپانوف تعریف می‌شود.

|  |  |
| --- | --- |
| *رابطه 2-3* |  |

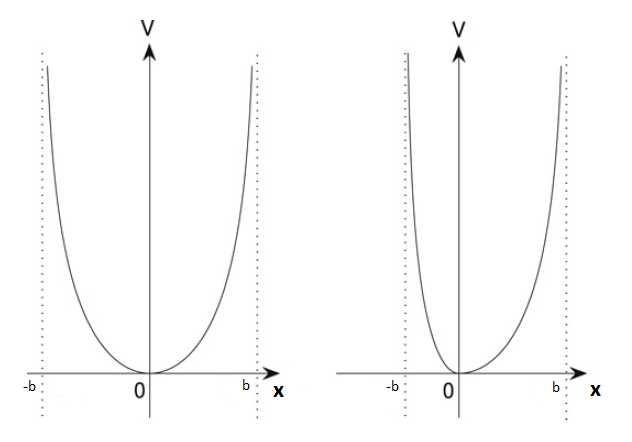
## تابع لیاپانوف مانع

همانطور که اشاره شد، مبحث رعایت و در نظر گیری قیود برای حالت‌ها، ورودی‌ها، و خروجی کنترلی، بسیار حائز اهمیت است. چرا که در مواردی سلامت عملکردی سیستم ربات، در عمل، مشروط و منوط به تخطی نکردن از این قیود و رعایت آن‌هاست. برای نیل به این هدف و رعایت این قیود، در ارائه‌ی کنترلرها از توابع لیاپانوف مانع یا به اختصار BLFها استفاده می‌شود. این توابع اولین بار در [32] مطرح شد و پس از آن در مقالات بسیاری به آن ارجاع داده شد. بنابراین این توابع نقش مهمی را در تولید کنترلرهای مبتنی بر رعایت قیود ایفا می‌کنند.

برای تعریف این توابع باید گفت یک تابع لیاپانوف مانع یک تابع اسکالر V(x) است که با توجه به سیستم Pₓ = ƒ(x) در یک منطقه باز D که مبتنی بر x است، تعریف می‌شود. این تابع پیوسته[[34]](#footnote-34)، مثبت معین، دارای مشتقات جزئی مرتبه اول در هر نقطه از D است. خاصیت بسیار مهم این تابع این است که هر گاه x به عنوان ورودی تابع،‌ به مرز D نزدیک می‌شود،V(x) به ∞ میل می‌کند. همچنین به ازای یک مقدار مثبت ثابت b شرط-b ≤ V(x(t)) ≤ b برای همه t ≥ 0در xهای حاصل از حل برقرار است. همچنین می‌بایست شرط x(0) ∈ D برقرار باشد و مقدار ورودی در لحظه‌ی 0 عضوی از بازه‌ی D باشد.

|  |  |
| --- | --- |
| رابطه 2-4 |  |
| رابطه 2-5 |  |

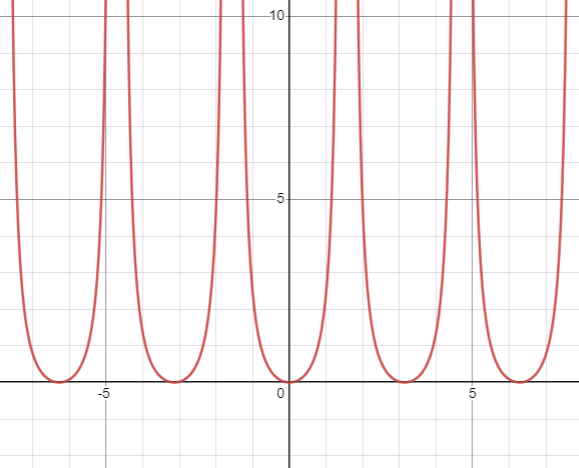
همانطور که در شکل 2-1 نشان داده شده است، تابع لیاپانوف مانع ممکن است به صورت متقارن یا یا نامتقارن باشد.



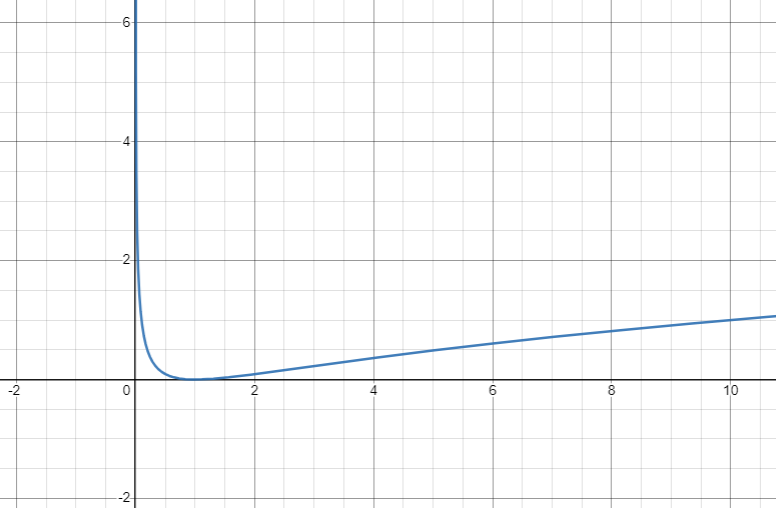
شکل ‏2‑1 تابع لیاپانوف مانع. سمت راست نامتقارن. سمت چپ متقارن

با توجه به آنچه گفته شد، از خصوصیات مهم توابع لیاپانوف مانع آن است که این توابع دارای دامنه‌ی به خصوصی و برد آن‌ها شامل مثبت بی‌نهایت و همیشه مثبت باشد. چرا که می‌بایست با نزدیک شدن متغیر ورودی به مرزهای مشخصی، مقدار خروجی تابع به بی نهایت میل کند. همچنین می‌دانیم انتخاب تابع کاندید لیاپانوف بسته به نوع معادلات سیستم می‌تواند متفاوت باشد. از همین روی انواع مختلفی از توابع لیاپانوف مانع قابل انتخاب هستند.

از جمله‌ی این توابع می‌توان به مربع تانژانت، مربع لگاریتم و لگاریتم بر پایه‌ی نپر و... اشاره کرد. این توابع به ازای ورودی‌های خاصی به بینهایت میل می‌کنند.



شکل ‏2‑2 تابع مربع تانژانت که به ازای به بینهایت میل می‌کند



شکل ‏2‑3 تابع مربع لگاریتم که به ازای x = 0 به بینهایت میل می‌کند

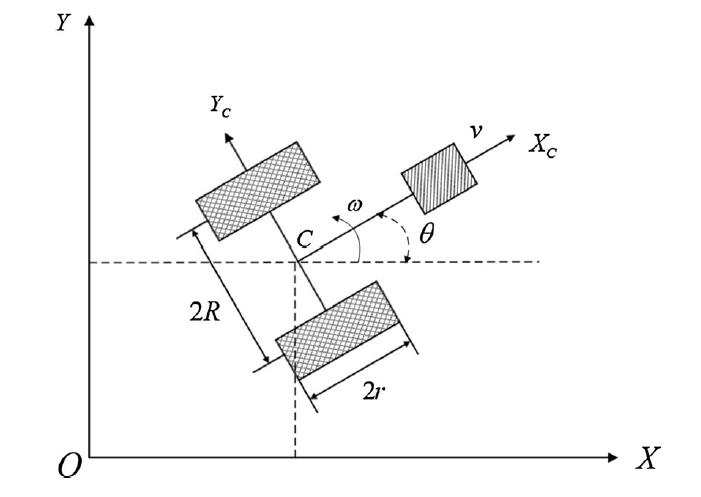
بعنوان مثال تابع زیر می‌تواند یک تابع کاندید لیاپانوف مانع باشد.

|  |  |
| --- | --- |
| رابطه 2-6 |  |

مشخصاً در تابع فوق اگر مقدار x به 5 میل کند، مقدار V(x) به بینهایت میل می‌کند. همچنین این تابع مثبت معین است.

## مدل سازی سینماتیکی

برای ارائه‌ی کنترلر سینماتیکی ابتدا می‌بایست پلتفرم را در فضای سینماتیک مدل سازی کنیم. برای این مقصود یک پلتفرم را به شکل زیر در صفحه‌ی مختصات دکارتی در نظر می‌گیریم.



شکل ‏2‑4 نمونه‌ی یک پلتفرم ربات چرخ دار غیر هولونومیک با سه چرخ

در این شکل نمونه‌ی یک پلتفرم ربات چرخ دار غیر هولونومیک با سه چرخ مشاهده می‌شود. نقطه‌ی C مرکز ثقل ربات که در نقطه‌ی ( واقع است در نظر گرفته می‌شود. دو چرخ عقب متصل به موتور و برای انتقال قدرت به سیستم و چرخ جلو هرزگرد می‌باشد. مقدار r شعاع چرخ و 2R برابر با عرض ربات می‌باشد. مقدار برابر با زاویه‌ی چرخش ربات و در جهت پادساعتگرد مثبت است. بنابراین در این جهت سرعت زاویه‌ای مثبت خواهد بود. همچنین سرعت خطی v در جهت بردار است.

بنابراین برای مدلسازی سینماتیکی، معادله‌ی ماتریسی زیر مطرح می‌شود

|  |  |
| --- | --- |
| رابطه 2-7 |  |
| رابطه 2-8 |  |
| رابطه 2-9 |  |

که در آن . به این صورت معادلات دیفرانسیل حاکم بر سیستم مشخص شده و ارتباط دهنده‌ی مکان ربات و سرعت خطی و دورانی آن تبیین می‌گردد.

در اینجا برای تسهیل ادامه‌ی روند و ارائه‌ي کنترلر سینماتیکی، از یک تبدیل همومورفیسم[[35]](#footnote-35) برای تبدیل المان‌های مکانی سیستم به صورت ، و یک ماتریس تبدیل حالت برای تبدیل سرعت‌های خطی و دورانی سیستم بهره می‌گیریم.

|  |  |
| --- | --- |
| رابطه 2-10 تبدیل همومورفیسم |  |
| رابطه 2-11 ماتریس تبدیل حالت |  |

در این صورت سیستم می‌تواند به حالت معادلات زنجیری زیر در بیاید

|  |  |
| --- | --- |
| رابطه 2-12 |  |

## کنترلر سینماتیکی

حال برای ارائه‌ي کنترلر سینماتیکی،‌ ابتدا باید خطاهای حرکتی را تعریف کنیم. اگر مقدار دلخواه را با اندیس d نشان دهیم،‌ خطاهای سرعت تبدیل شده به صورت زیر تعریف می‌شوند

|  |  |
| --- | --- |
| رابطه 2-13 |  |
| رابطه 2-14 |  |

که در رابطه 2-14 یک متغیر واسط و شامل می‌باشد که تعریف آن ارائه خواهد شد.

حال به سراغ تعریف مقادیر قیود متغیرهای حالت بر روی و و می‌رویم. همانطور که مشخص است از آنجا که و قیدی بر روی معادلات نخواهیم داشت، پس قیود روی دو حالت دیگر تعریف می‌شوند. حال معادلات زیر را در فرض می‌گیریم.

|  |  |
| --- | --- |
| رابطه 2-15 |  |
| رابطه 2-16 |  |

که در آنها و همان قیود متغیرهای حالت ما هستند. در حقیقت می‌خواهیم که متغیرهای حالت ما از این دو مقدار تجاوز نکنند. از طرفی برای مقادیر دلخواه متغیرهای حالت نیز قیودی در نظر می‌گیریم.

|  |  |
| --- | --- |
| رابطه 2-17 |  |
| رابطه 2-18 |  |

بنابراین، می‌توان از معادلات بالا با توجه به روابط 2-13 و 2-14 به قیودی روی و رسید.

|  |  |
| --- | --- |
| رابطه 2-19 |  |
| رابطه 2-20 |  |

بنابراین، مشخص نمودیم که میخواهیم خطاهای و از مقادیر و تجاوز نکنند.

همچنین با توجه به معادلات زیر

|  |  |
| --- | --- |
| رابطه 2-21 |  |
| رابطه 2-22 |  |

و مشتق گیری از طرفین، معادلات زیر را داریم

|  |  |
| --- | --- |
| رابطه 2-23 |  |

حال یک تابع کاندید لیاپانوف مانع به صورت زیر تعریف می‌کنیم.

|  |  |
| --- | --- |
| رابطه 2-24 |  |

در معادلات بالا همانطور که مشخص است، تابع مثبت معین بوده و اگر خطای یا به مقادیر و نزدیک شوند، تابع به بینهایت میل می‌کند.

|  |  |
| --- | --- |
| رابطه 2-25 |  |

همچنین این تابع پیوسته و دارای مشتق مرتبه اول است. بنابراین یک کاندید مناسب به عنوان تابع لیاپانوف مانع محسوب می‌شود. علاوه براین،‌ با توجه به یکسان بودن L پلتفرمی که کنترل همکارانه را بر روی آن‌ها اعمال خواهیم کرد، میتوان این معادلات را به شکل زیر برای L پلتفرم تعمیم داد.

|  |  |
| --- | --- |
| رابطه 2-26 |  |

با مشتق مرتبه‌ی اول نسبت به زمان از تابع لیاپانوف مانع بالا داریم:

|  |  |
| --- | --- |
| رابطه 2-27 |  |

بنابراین کافی‌ست مقادیر و را به گونه‌ای طراحی کنیم که منفی معین شده و پایداری سیستم با در نظر گیری قیود حالت اثبات شود. برای این منظور داریم

|  |  |  |  |
| --- | --- | --- | --- |
| رابطه 2-28 | |  | |
| رابطه 2-29 |  | |

بنابراین و را طراحی نمودیم. حال با جایگذاری در داریم

|  |  |
| --- | --- |
| رابطه 2-30 |  |

در معادله‌ی بالا ترم سمت چپ یعنی یک عبارت همیشه منفی به ازای و است.

ترم سمت راست یعنی نیز به ازای طراحی مناسب، به سمت 0 خواهد رفت. بنابراین می‌بایست را به درستی طراحی کنیم. از معادله‌ي و با توجه به اینکه مقدار مشمول قید نمی‌شود، می‌توان به سادگی به مقدار زیر رسید.

|  |  |
| --- | --- |
| رابطه 2-31 |  |

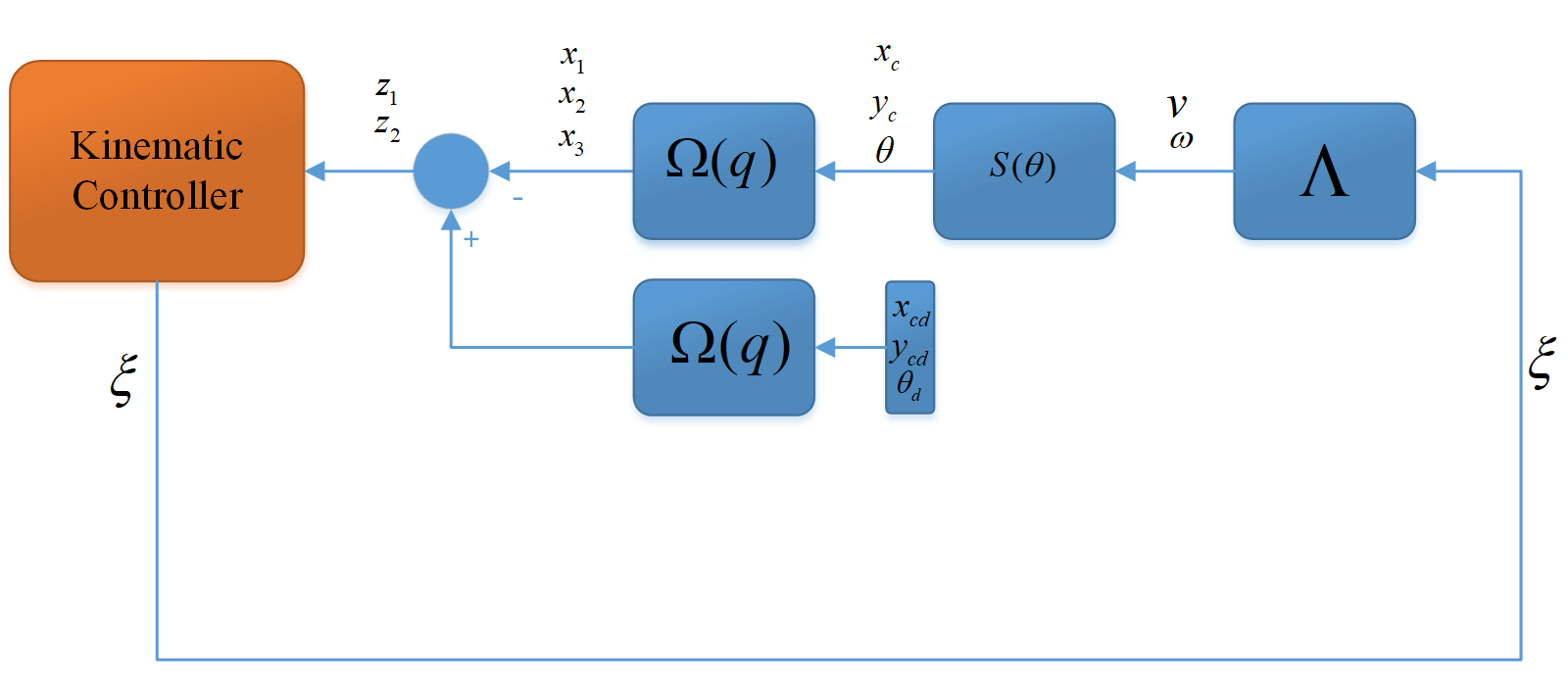
در این صورت نیز، یک تابع لیاپانوف ساده به صورت زیر مطرح می‌شود.

|  |  |  |
| --- | --- | --- |
| رابطه 2-32 | |  |
| رابطه 2-33 |  | |

برای منفی شدن باید باشد.

بنابراین تا یه اینجا مقدار تابع لیاپانوف برابر با را دیدیم و مقدار همیشه منفی شد. پس کنترلر سینماتیکی معرفی شده، سیستم سینماتیکی ما را پایدار خواهد کرد. همچنین با توجه به تعریف تابع لیاپانوف مانع ، میدانیم که خطاهای و و به تبع آنها و و به تبع آن‌ها و از قیود خود تخطی نمی‌کنند.

همچنین برای درک بهتر، در شکل نحوه‌ی عملکرد کنترلر سینماتیکی به صورت تصویری آورده شده است.



شکل ‏2‑5 نحوه عملکرد کنترلر سینماتیکی

## شبیه سازی سینماتیکی

در این بخش ابتدا مقادیر عددی مسیر زمانی مرجع مربوط به هر ربات، قیود حالت و ضرایب کنترلی و... را ارائه کرده سپس شبیه سازی و نتایج آن در فضای متلب را نشان خواهیم داد. برای این منظور و برای نشان دادن عمومیت کنترلر ارائه شده، ابتدا برای یک پلتفرم، دو مسیر زمانی متفاوت دایره و ∞ را شبیه سازی کرده سپس برای هر سه پلتفرم مسیر زمانی بیضی ارائه می‌شود. این نتایج نشان دهنده‌ی پایداری سیستم و کاهش خطای حالت‌ها به سمت 0 می‌باشند که تضمین کننده‌ی صحت کنترلر سینماتیکی ارائه شده هستند.

مقادیر عددی شبیه سازی، فارغ از شکل مسیر زمانی مرجع که شامل قیود حالت، و ضرایب کنترلی می‌باشند به صورت زیر هستند.

جدول ‏2‑1 مقادیر عددی شبیه سازی سینماتیکی

|  |  |
| --- | --- |
| مقدار | متغیر |
| 0.35 m |  |
| 0.35 m |  |
| 1.3 |  |
| 1.5 |  |
| 1.5 |  |

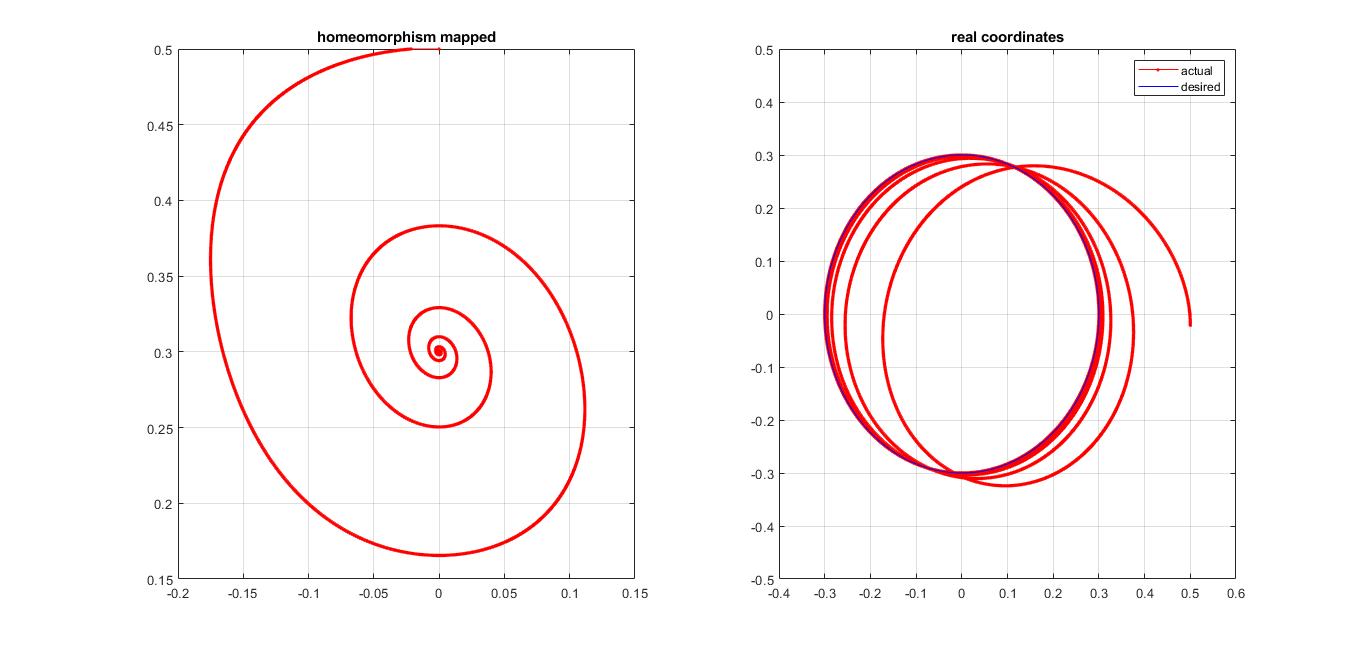
### دایره‌ی تک پلتفرم

#### مقادیر عددی

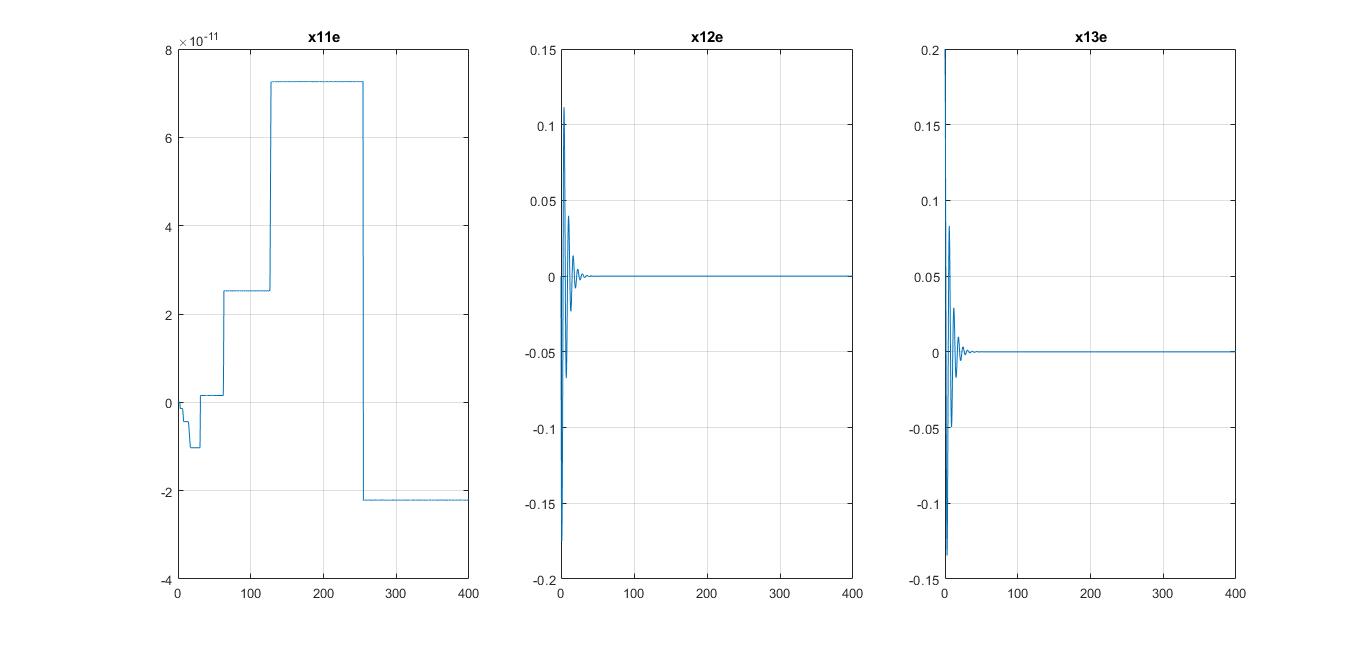
ابتدا برای تک پلتفرم یک مسیر ساده‌ی دایره در نظر می‌گیریم.

|  |  |  |
| --- | --- | --- |
| رابطه 2-34 مسیر زمانی دایره |  | |
|  | |
| رابطه 2-35 شرایط اولیه: | |  |

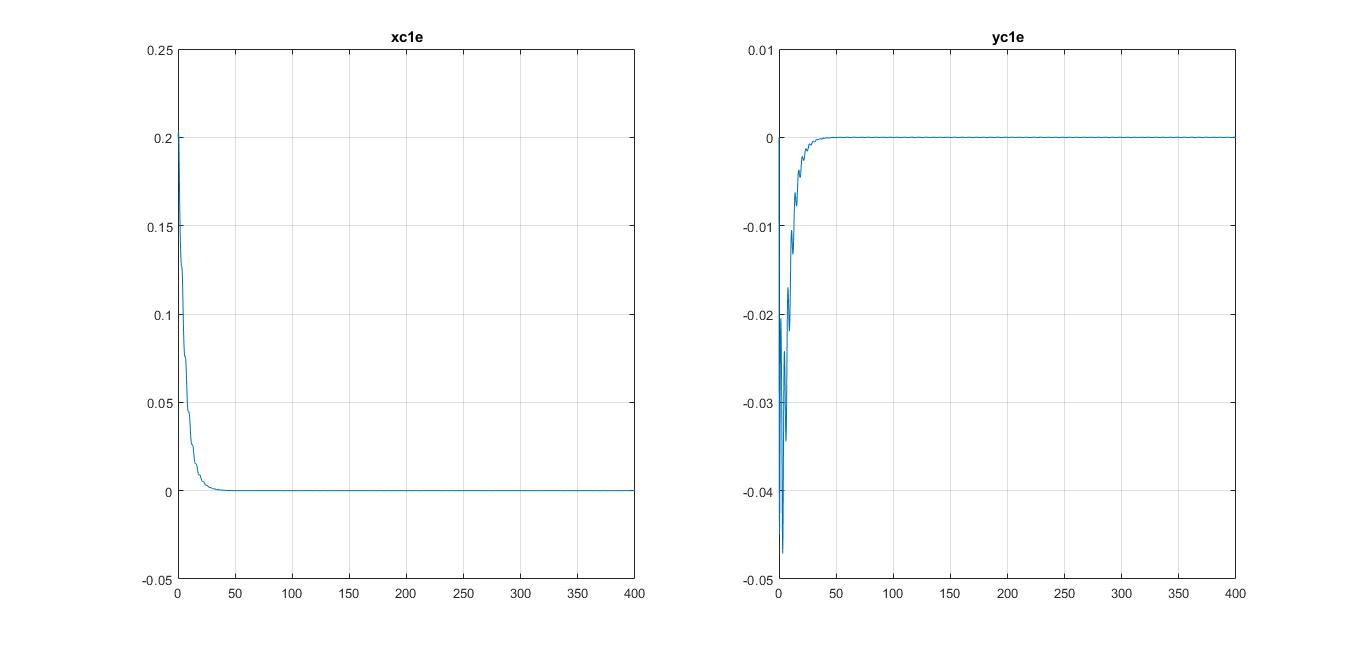
#### شبیه سازی



شکل ‏2‑6 مسیر حرکت ربات سمت راست مختصات واقعی سمت چپ مختصات تحت تبدیل همومورفیسم



شکل ‏2‑7خطای حالت‌های تبدیل شده



شکل ‏2‑8 خطای حالت‌ها در مختصات واقعی

### حرکت ∞

#### مقادیر عددی

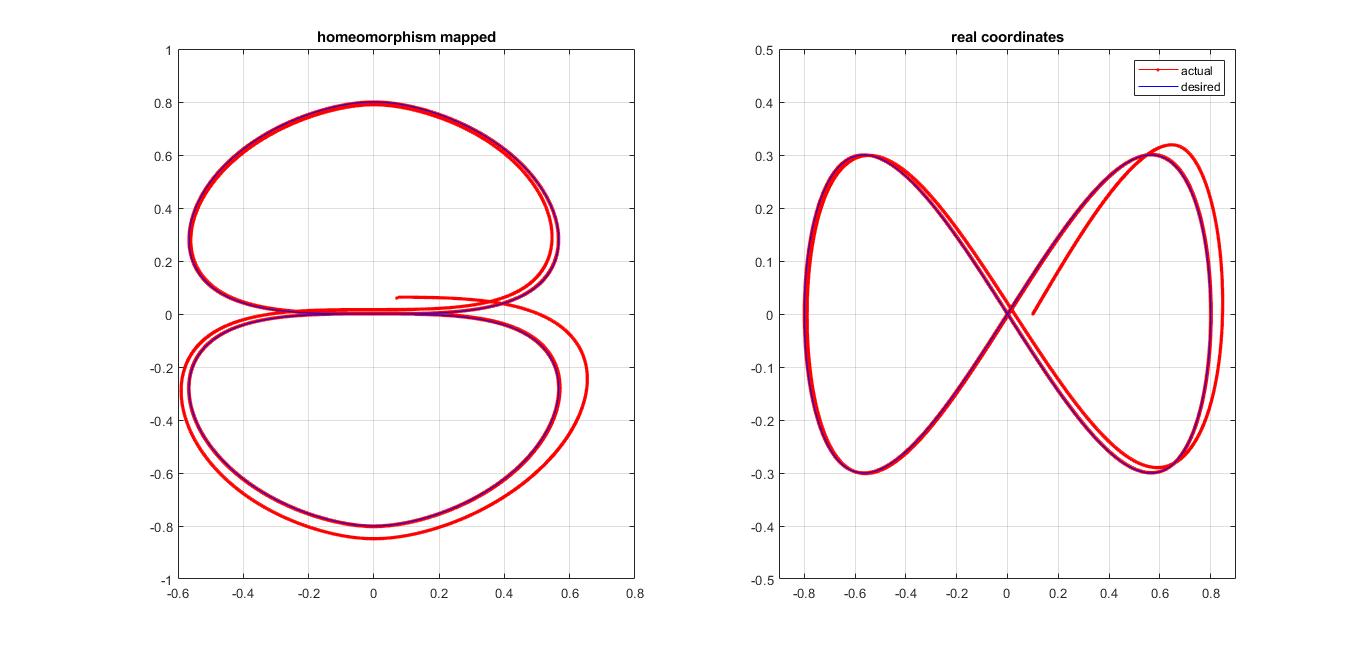
ابتدا برای تک پلتفرم یک مسیر ساده‌ی دایره در نظر می‌گیریم.

|  |  |
| --- | --- |
| رابطه 2-36 مسیر زمانی ∞ |  |
|  |

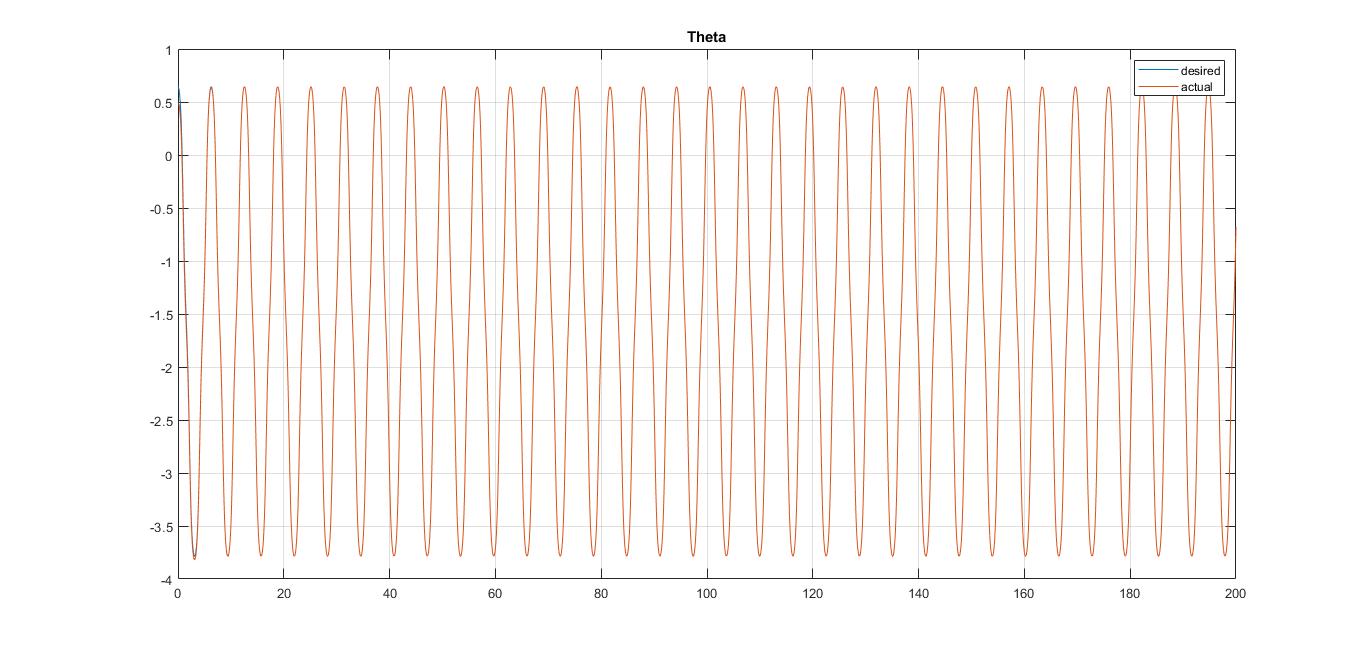
|  |  |
| --- | --- |
| رابطه 2-37 شرایط اولیه |  |

لازم به ذکر است که این مسیر زمانی ∞ از مبدا شروع شده و به سمت راست و بالا می‌رود.

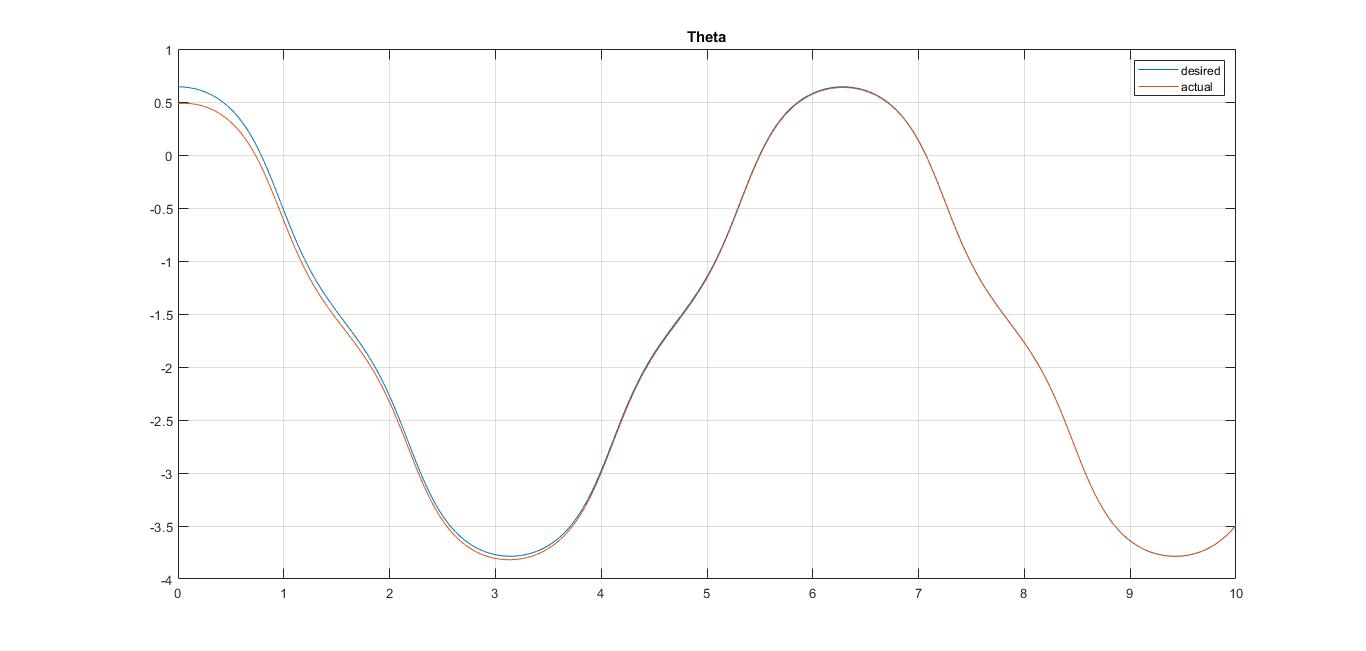
#### نتایج شبیه سازی



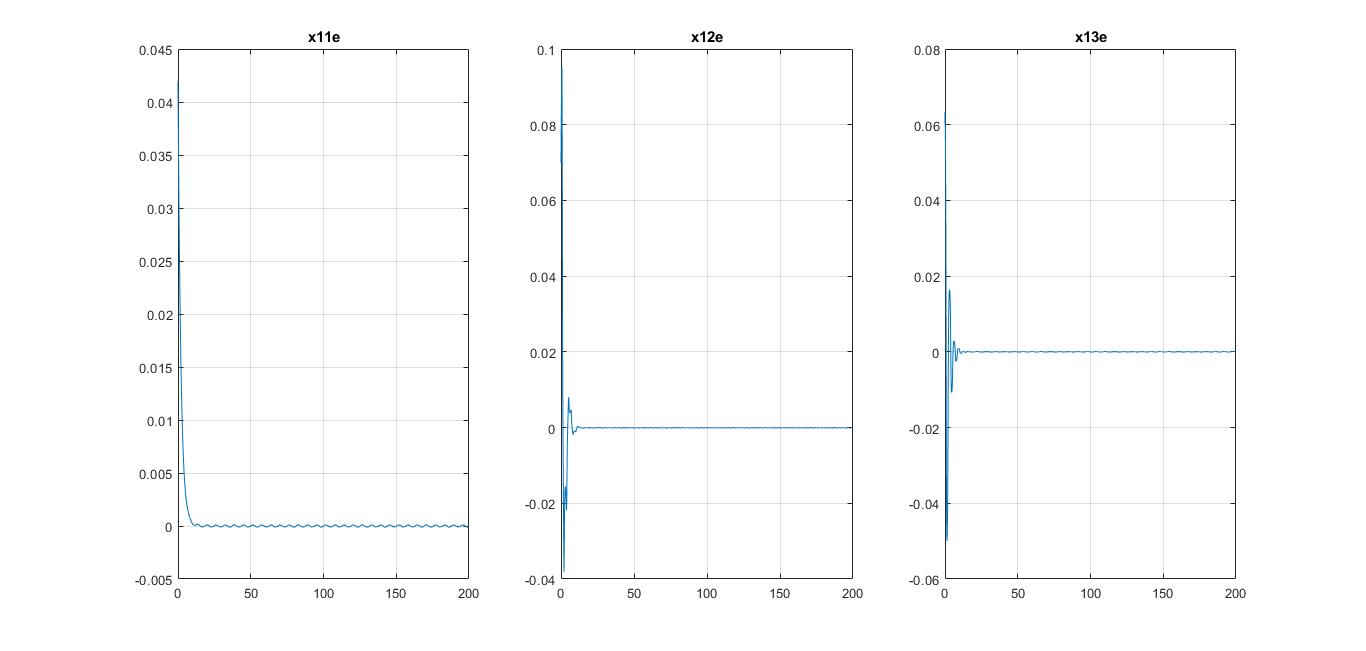
شکل ‏2‑9 مسیر حرکت ربات سمت راست مختصات واقعی سمت چپ مختصات تحت تبدیل همومورفیسم



شکل ‏2‑10 تغییرات زاویه‌ی دلخواه و واقعی



شکل ‏2‑11 تغییرات زاویه دلخواه و واقعی تا زمان 10 ثانیه



شکل ‏2‑12 خطای حالت‌های تبدیل شده



شکل ‏2‑13 خطای حالت‌ها در مختصات واقعی

### بیضی برای سه ربات

#### مقادیر عددی

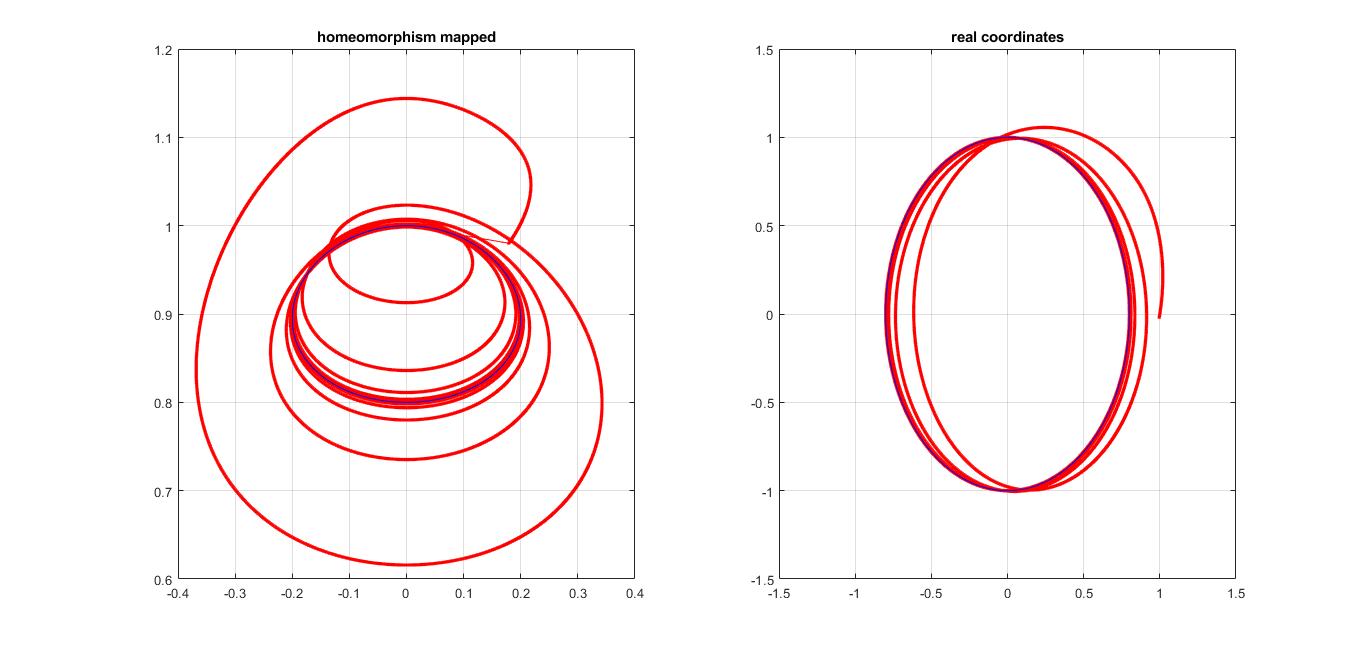
شبیه سازی‌های انجام شده برای تعداد L = 3 پلتفرم ارائه می‌شوند. همانطور که اشاره شد، به هر یک از ربات‌ها یک مسیر زمانی مجزا و متفاوت تخصیص داده می‌شود. همچنین می‌بایست و برای هر ربات با توجه به مسیر زمانی مرجع، محاسبه شود. به این صورت و همچنین مقدار است که برای هر ربات این مقادیر متفاوت خواهند بود. همچنین شرایط اولیه‌ی تخصیص داده شده به هر ربات متفاوت است. بنابراین مقادیر عددی به شرح زیر ارائه می‌شوند.

جدول ‏2‑2 مسیرهای زمانی بیضی برای ربات‌های 1، 2 و 3

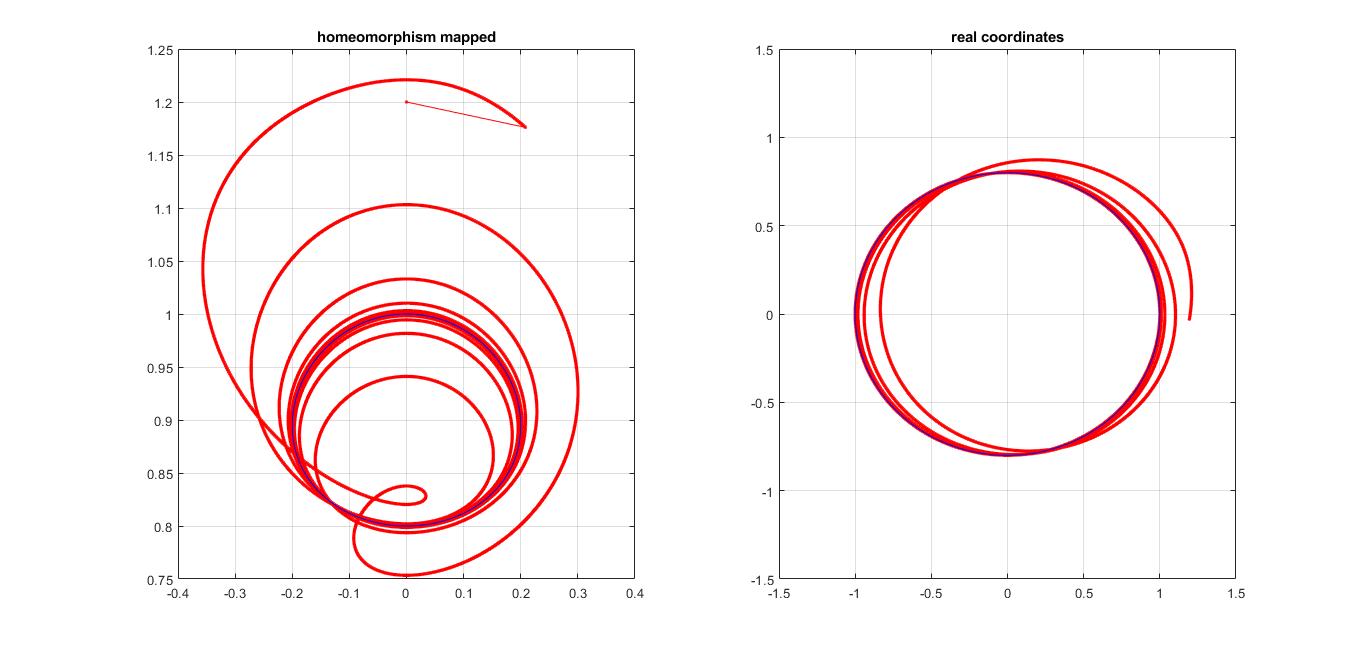
|  |  |
| --- | --- |
| رابطه 2-38 مسیر زمانی بیضی، ربات 1 |  |
|  |
| رابطه 2-39 مسیر زمانی بیضی، ربات 2 |  |
|  |
| رابطه 2-39 مسیر زمانی بیضی، ربات 3 |  |
|  |

|  |  |
| --- | --- |
| رابطه 2-40 شرایط اولیه برای هر سه ربات |  |

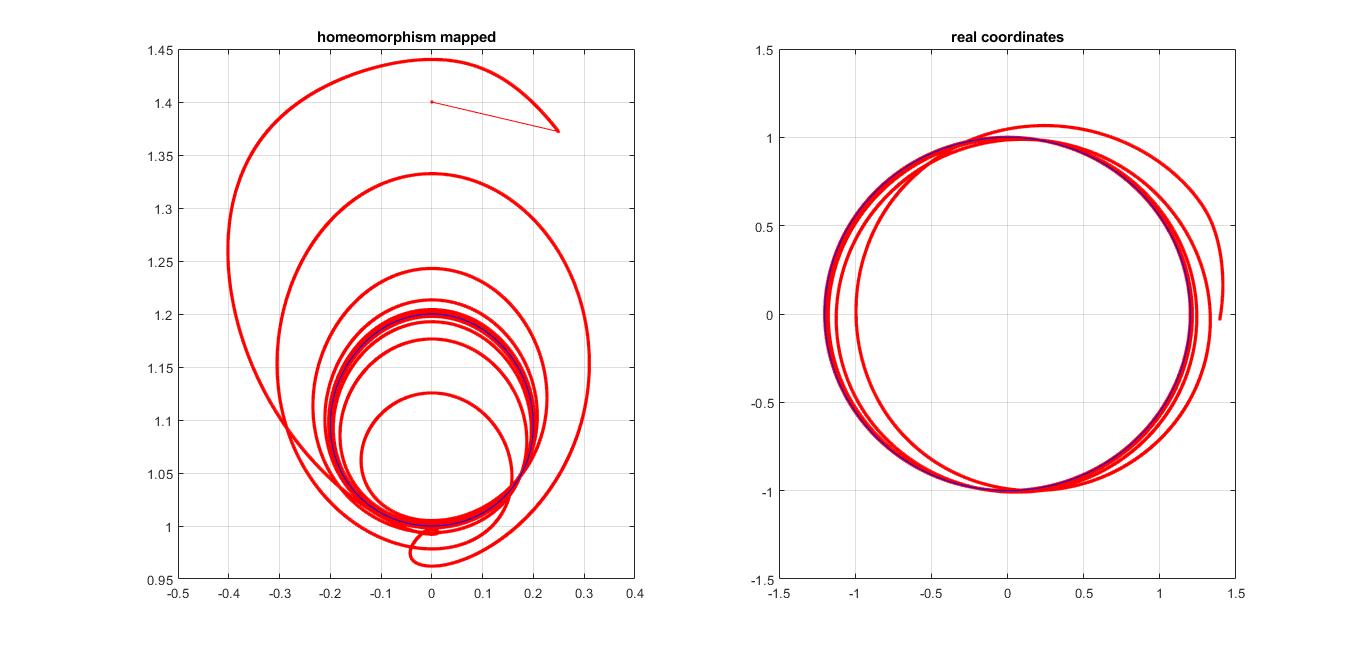
#### نتایج شبیه سازی



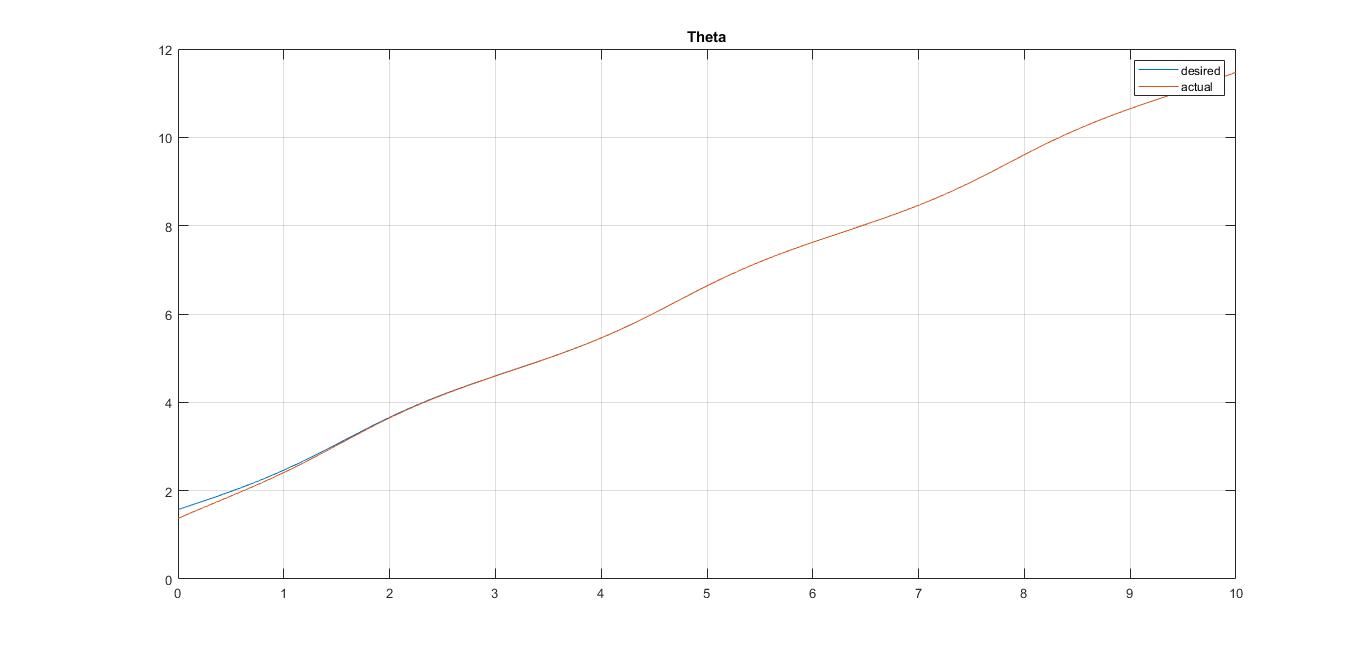
شکل ‏2‑14 مسیر حرکت ربات 1 سمت راست مختصات واقعی سمت چپ مختصات تحت تبدیل همومورفیسم



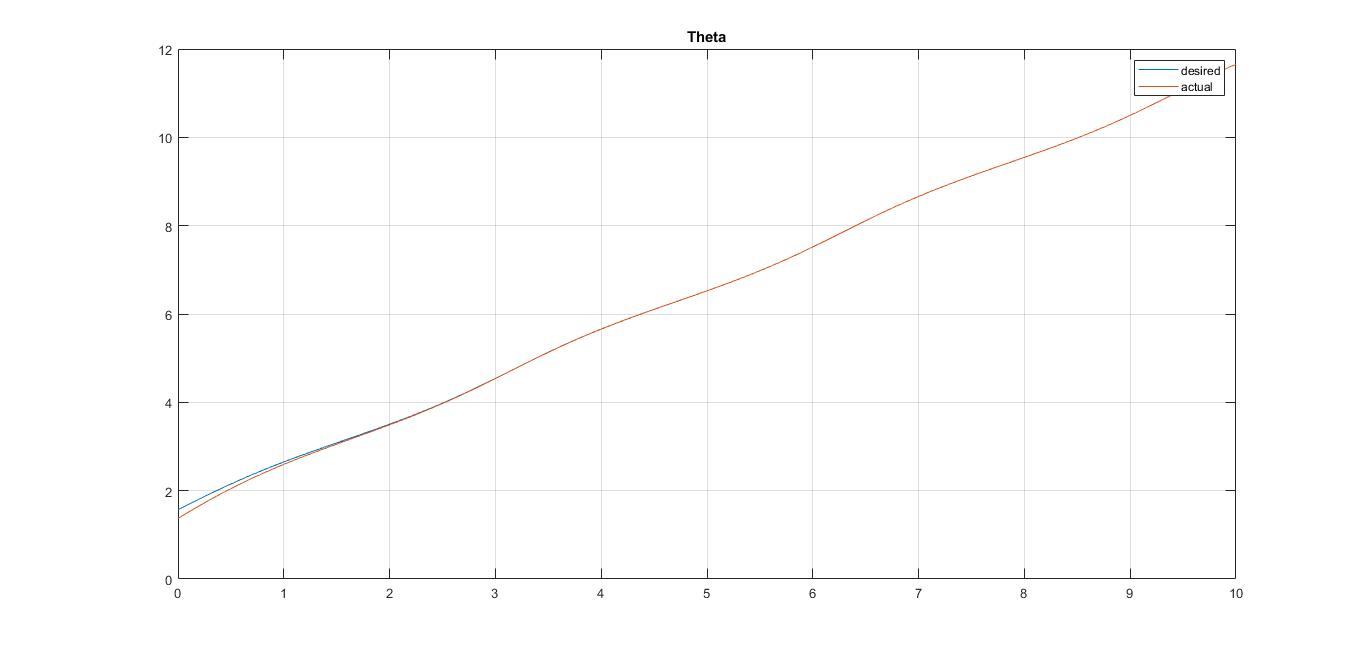
شکل ‏2‑15 مسیر حرکت ربات 2 سمت راست مختصات واقعی سمت چپ مختصات تحت تبدیل همومورفیسم



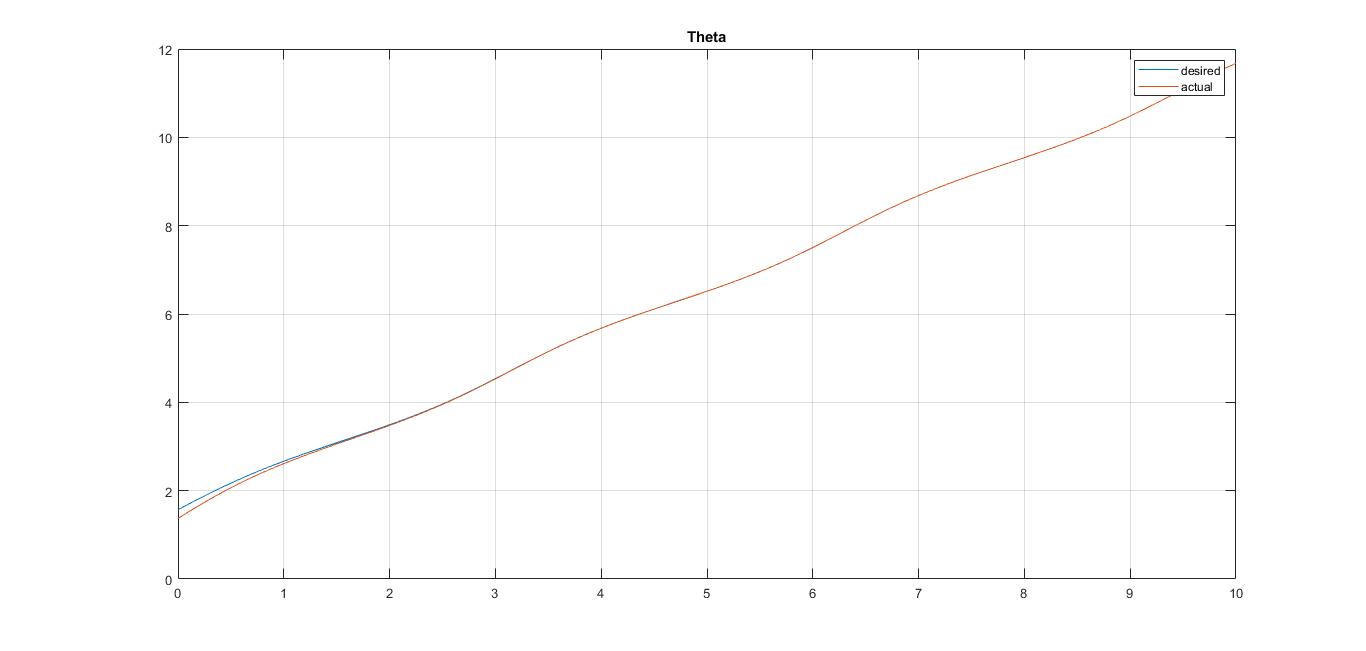
شکل ‏2‑16 مسیر حرکت ربات 3 سمت راست مختصات واقعی سمت چپ مختصات تحت تبدیل همومورفیسم



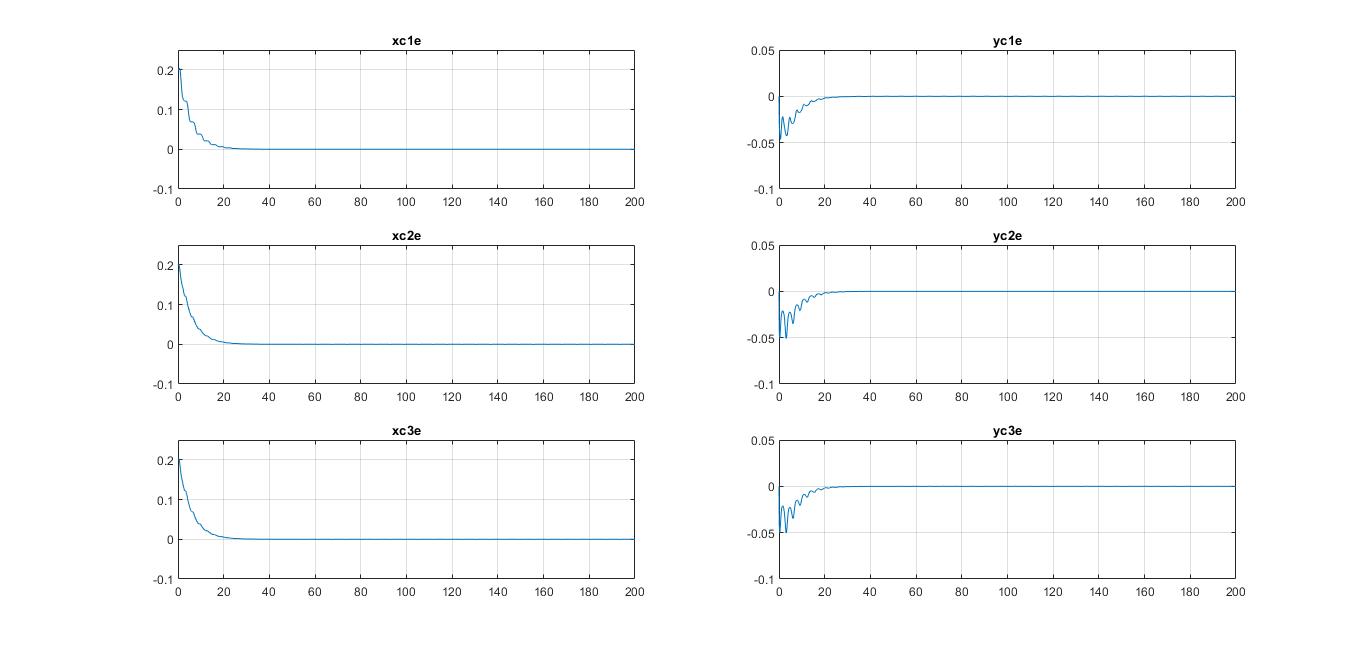
شکل ‏2‑17 تغییرات زاویه دلخواه و واقعی ربات 1 تا زمان 10 ثانیه



شکل ‏2‑18 تغییرات زاویه دلخواه و واقعی ربات 2 تا زمان 10 ثانیه



شکل ‏2‑19 تغییرات زاویه دلخواه و واقعی ربات 3 تا زمان 10 ثانیه



شکل ‏2‑20 خطای حالت‌های هر سه ربات در مختصات واقعی

## نتیجه گیری

با توجه به نتایج ارائه شده، می‌توان این گونه نتیجه گیری کرد که اعمال کنترلر سینماتیکی ارائه شده، با دریافت پس‌خورهای لحظه‌ای مناسب از مکان پلتفرم به ربات، موجب پایداری سیستم شده و خطای ناشی از شرایط اولیه‌های مختلف به صفر میل می‌کند. همچنین عمومیت این کنترلر با توجه به تغییر مسیر زمانی مرجع به اشکال مختلف، مانند حرکت دایروی و حرکت ∞ مناسب است. بنابراین سیستم در مسیر زمانی‌های پیچیده تر از دایره و بیضی نیز عملکرد مناسبی از خود نشان می‌دهد و برای مسیر زمانی‌های متفاوت می‌توان از این کنترلر سینماتیکی استفاده نمود.

# فصل سوم

طراحی کنترلر دینامیکی

## مقدمه

در فصل پیش،‌ طراحی کنترلر سینماتیکی و اثبات پایداری آن را به طور جزیی بررسی نمودیم. در این فصل از سرعت یا های تولید شده در فصل دوم استفاده نموده و کنترلر دینامیکی را طراحی می‌نماییم. در کنترلر سینماتیکی پس از فرض و ایجاد مسیر زمانی مرجع، قانونی تولید می‌کنیم که خروجی آن سرعت لازم برای رسیدن ربات‌ها به مسیرهای زمانی مرجع می‌باشد. حال برای آنکه هر ربات به این سرعت‌ها برسد، نیاز است که با دینامیک ربات در تعامل باشیم. چرا که هر سیستم دینامیک مختص خود را دارد و میزان گشتاور لازم برای رسیدن به سرعت مورد نظر باید با توجه به دینامیک سیستم تولید و به آن اعمال شود. همچنین پس از تولید گشتاور، با توجه به شرایط فیزیکی موتورها از قبیل مقاومت آرمیچر، نسبت گیربکس و... گشتاور مورد نظر می‌تواند تبدیل به ولتاژ مورد نیاز برای رسیدن به گشتاور گفته شده شود. موضوعی که در روش اشاره شده به آن خواهیم رسید و خروجی کنترلر دینامیکی ما ولتاژ اعمالی به موتورهای چپ و راست خواهد بود.

برای حفظ توالی موضوعی، ابتدا تعریف کنترل همکارانه[[36]](#footnote-36)،‌ سیستم‌‌های چند رباته، نظریه‌ی وفاق و... ارائه شده سپس به سراغ ارائه‌ی کنترلر دینامیکی خواهیم رفت.

## کنترل همکارانه

حرکات جمعی گروه‌های حیوانات از زیباترین مناظر طبیعت است. هر فرد از گروه، حرکات خاص خود را دارد، با این حال این حرکت جمعی باعث می شود که این گروه یک موجودیت واحد با قوانین حرکتی و تعاملات خاص خود به وقایع بیرونی به نظر برسد. دسته‌های پرندگان، دسته های ماهی‌ها و گله‌های حیوانات مجموعه‌هایی از این قبیل هستند.

گروه برای رسیدن به اهداف خود، مانند جستجوی غذا یا مهاجرت، و یا فرار از شکارچیان و عبور از موانع حرکت می‌کند و اشکال مختلفی می‌گیرد. چنین حرکات هماهنگ و موزونی، باعث می شود که شخص ناظر آن‌را به حرکتی از پیش تعیین شده و بر اساس یک سناریو تشبیه کند. و حال آنکه این حرکات محصول برنامه ریزی پیشین نیستند، بلکه حاصل تصمیم گیری‌ها و پاسخ‌های لحظه‌ای هر یک از اعضا به وقایع بیرونی هستند. بنابراین حرکات گروهی، به گروه اجازه می دهد تا به هدفی که رسیدن به آن به صورت فردی دشوار است ساده‌تر دست یابد.



شکل ‏3‑1حرکت جمعی یک دسته پرنده‌ی سار



شکل ‏3‑2 حرکت جمعی یک دسته ماهی

### قوانین حرکتی رینولدز

در کنترل مدرن نیز، مفهوم حرکت جمعی و همکاری با الهام گیری از حرکات گروهی حیوانات در طبیعت، وجود داشته و استفاده‌ی بسیاری از آن می‌شود. برای توضیح کنترل همکارانه از پایه‌ای ترین قوانین حاکم بر حرکت اعضای گروه در یک سیستم با چند عامل[[37]](#footnote-37) آغاز می‌کنیم. این قوانین به قوانین رینولدز موسوم اند. حرکات فردی اعضای یک گروه، نتیجه‌ی ایجاد تعادل بین دو رفتار متضاد است: ۱- میل به نزدیک بودن به گروه. ۲- میل به اجتناب از برخورد با اعضای دیگر. قوانین رینولدز، تمایلات حاکم بر حرکات اعضای گروه را از طریق سه قانون خود به بیان می‌کند. این قوانین به حرکات همسایه‌های هر عضو در گروه بستگی دارد و به صورت زیر است:

* جلوگیری از برخورد: از برخورد با همسایگان اجتناب شود
* تطبیق سرعت: سرعت و جهت حرکت با همسایگان مطابق باشد
* فاصله از مرکز دسته: فاصله از مرکز دسته حفظ شود

قوانین رینولدز به خوبی حرکت دسته جمعی گروه‌های حیوانات را به بیان می‌کند و همچنین می تواند به عنوان قوانین پایه‌ای برای کنترل همکارانه‌ی سیستم های چند عاملی انسانی مانند وسایل نقلیه استفاده شود. این قوانین به دانش هر عضو از خصوصیات مختلف همسایگان بستگی دارد. می‌دانیم که تبادل اطلاعات بین اعضای یک گروه اجتماعی در تعیین حرکت کل گروه بسیار مؤثر است. بنابراین بین اعضای گروه می‌بایست یک شبکه‌ی ارتباطی جهت تبادل اطلاعات برقرار باشد. این شبکه ارتباطی را می‌توان به صورت یک گراف مدل سازی نمود. پیش از آنکه روابط کنترل همکارانه مطرح شود، می‌بایست دانش حداقلی‌ای از نظریه‌ی گراف‌ها و توپولوژی‌های آن‌ها داشت.

### گراف‌ها

گراف‌ها به طور کلی می‌توانند جهت دار[[38]](#footnote-38) یا بدون جهت[[39]](#footnote-39) باشند. جهت دار بودن یال‌های گراف به معنی آن است که اطلاعات به صورت یک طرفه مجاز به ارسال در جهت یال هستند. و همچنین بدون جهت بودن گراف به این معنی است که اطلاعات به صورت دو طرفه بین دو عضو قابل تبادل است. اعضا نیز به عنوان راس‌های گراف مدل می شوند و همچنین عامل نامیده می شوند. نمایش گراف G به صورت یک زوج می‌باشد به طوری که یک مجموعه از N گره یا راس بوده و E، مجموعه ای از یال ها است. عناصر E نیز به صورت مشخص می شوند. این یال نشان دهنده‌ی اتصال بین راس و است. مجموعه‌ی همسایگان راس به صورت زیر نمایش داده می‌شود.

|  |  |
| --- | --- |
| رابطه 3-1 |  |

مجموعه‌ی همسایه‌های در رابطه 3-1 نشان دهنده گره‌های است به طوریکه عامل iام با آن‌ها تبادل اطلاعات داشته و در پاسخ به آنها حرکت خود را تعیین می کند.

همچنین در بحث گراف‌ها، یک ماتریس مهم مطرح است، به نام گراف مجاورت[[40]](#footnote-40). این ماتریس یک ماتریس مربعی شامل درایه‌های 0 و غیر 0 است که نشان دهنده‌ی همسایگی یا عدم همسایگی دو عامل در یک گراف است.

|  |  |
| --- | --- |
| رابطه 3-2 |  |

در ماتریس رابطه 3-2، اگر عامل iام با عامل jام همسایه نباشد،‌ درایه‌ی متناظر و اگر بین این دو عامل ارتباط برقرار باشد، است. همچین درایه‌های قطر اصلی است. همچنین در این ماتریس می‌توان جهت دار بودن یا نبودن گراف را نمایش داد. شرط گراف بدون جهت در رابطه 3-3 دیده می‌شود.

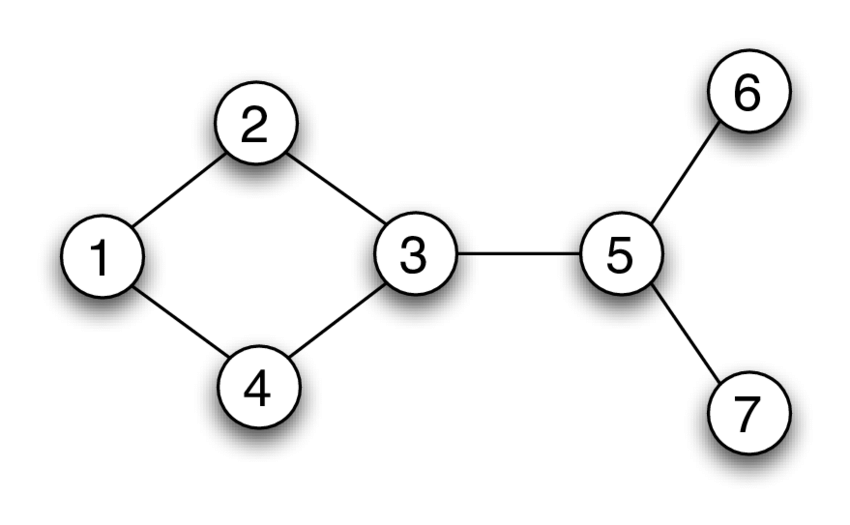
|  |  |
| --- | --- |
| رابطه 3-3 |  |

ماتریس مجاورت متناظر با گراف بدون جهت متقارن خواهد بود.

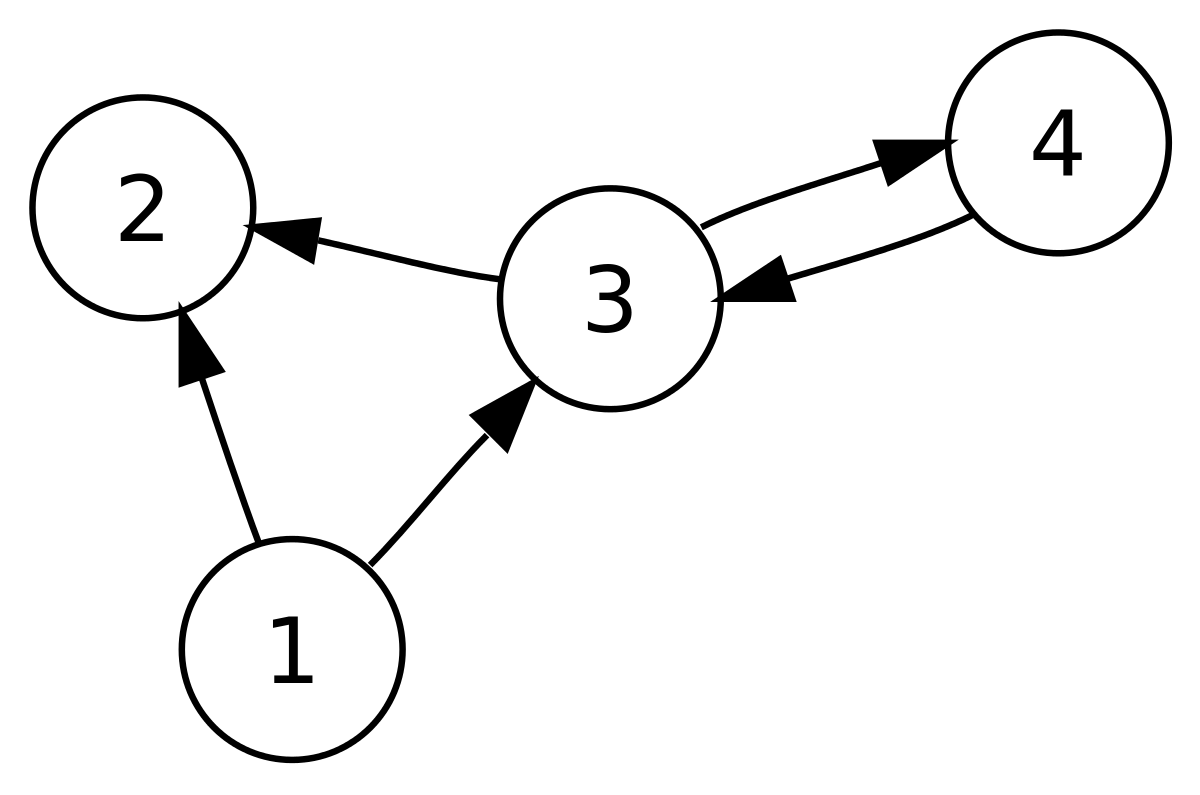
در شکل 29، یک گراف بدون جهت با ۷ راس دیده می‌شود. اگر این گراف، نشان دهنده‌ی توپولوژی ارتباطی یک سیستم چند رباته باشد، هر یک از رئوس نشان دهنده‌ی یک ربات یا عامل، و هر یال نشان دهنده‌ي همسایگی عامل‌های دو سمت خود است می‌باشد. همچنین با توجه به جهت دار نبودن این گراف، عامل‌های همسایه به یکدیگر اطلاعات را ارسال و از یکدیگر دریافت می‌کنند و تبادل اطلاعات به صورت دو طرفه است. عامل شماره 1 اطلاعات را به عامل شماره‌ی 2 ارسال کرده و از آن دریافت می‌کند. در حالی که عامل شماره‌ی 1 با عامل شماره‌ی 3 به طور مستقیم با عامل شماره‌ی 3 همسایه نبوده و تبادل اطلاعات با آن را ندارد.

در شکل 30، یک گراف جهت دار با ۴ عامل دیده می‌شود. مجددا اگر این گراف، نشان دهنده‌ی توپولوژی ارتباطی یک سیستم چند رباته باشد، ارسال اطلاعات بین عامل‌ها، در جهت نشان داده شده توسط هر یال انجام می‌شود و این تبادل اطلاعات به صورت یک طرفه است. بنابراین عامل شماره‌ي 1 اطلاعات را به عامل شماره‌ی 2 ارسال می‌کند، ولی از عامل شماره‌ی 2 اطلاعاتی دریافت نمی‌کند. همچنین تبادل اطلاعات بین عامل شماره‌ی 3 و 4 به صورت دو طرفه می‌باشد چرا که دو یال جهت دار از سمت هریک به دیگری متصل شده است.

با این تفاسیر، تعاریف مربوط به گراف‌های جهت دار و بدون جهت تبیین شد و می‌توان از آن‌ها در توصیف شبکه ارتباطی سیستم چند عامله استفاده نمود.



شکل ‏3‑3گراف بدون جهت Undirected Graph



شکل ‏3‑4 گراف جهت دار Directed Graph

### پیاده سازی قوانین رینولدز و نظریه‌ي وفاق[[41]](#footnote-41)

حال با علم به کلیات نظریه‌ی گراف‌ها‌، می‌توانیم با بهره گیری از وفاق بین عامل‌های یک سیستم چند عاملی، قوانین حرکتی رینولدز را پیاده سازی نماییم. برای مثال اگر حرکت یک عامل را در صفحه‌ی مختصات با ورودی کنترلی سرعت به صورت زیر فرض کنیم:

|  |  |
| --- | --- |
| رابطه 3-4 |  |

برای پیاده سازی قانون جلوگیری از برخورد، می‌توان از قانون رابطه 3-5 استفاده نمود

|  |  |
| --- | --- |
| رابطه 3-5 |  |

که در اینجا برابر است با مجموعه‌ی رابطه 3-6

|  |  |
| --- | --- |
| رابطه 3-6 |  |

مقدار برابر است با مقدار شعاعی که عامل‌ها باید فاصله‌ی‌ خود را از عامل‌های دیگر با آن بسنجند. و همچنین فاصله‌ی شعاعی برابر است با

|  |  |
| --- | --- |
| رابطه 3-7 |  |

با این اوصاف قانون کنترلی رابطه 3-5 موجب می‌شود عامل iام در محدوده‌ی از باقی عامل‌ها دور بماند. چرا که در صورتی که تفاوت حاصله از در محدوده‌ی مقدار مثبتی باشد،‌ یعنی ، برای کمتر نشدن این فاصله می‌بایست مقدار سرعت عامل iام یا مقداری منفی شود تا فاصله حفظ شود.

برای پیاده سازی قانون حفظ فاصله از مرکز دسته نیز، می‌توان از قانون رابطه 3-8 استفاده نمود

|  |  |
| --- | --- |
| رابطه 3-8 |  |

که در اینجا برابر با درایه‌های ماتریس مجاورت گراف نشان دهنده‌ی ارتباط عامل‌ها در سیستم ما است. همچنین برابر است با مجموعه‌ی رابطه 3-9

|  |  |
| --- | --- |
| رابطه 3-9 |  |

و در آن است. مجموعه رابطه 3-9 مشخص کننده‌ی عامل‌هایی ست که فاصله از آنها باید حفظ شود.

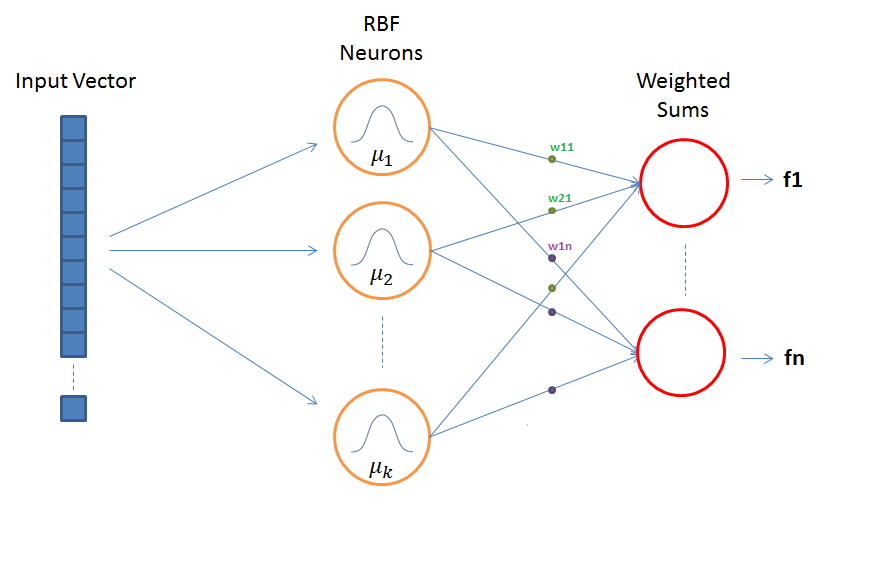
با این اوصاف قانون کنترلی بالا موجب می‌شود عامل iام در محدوده‌ی به سمت عامل‌های دیگر حرکت کند. چرا که در صورتی که تفاوت حاصله از در محدوده‌ی مقدار مثبتی باشد،‌ یعنی ، برای بیشتر نشدن این فاصله می‌بایست مقدار سرعت عامل iام یا مقداری مثبت شود تا فاصله حفظ شود.

در قوانین بالا، چه در حفظ فاصله از مرکز دسته رابطه 3-5، و چه در جلوگیری از برخورد رابطه 3-8، منطق تنظیم سرعت بر اساس برابر شدن و است. به این معنا که سرعت عامل iام یا تنها زمانی برابر با 0 می‌شود که .

به این حالت نظریه‌ی وفاق[[42]](#footnote-42) گفته می‌شود. چرا که این نظریه بیانگر شرایطی ست که در آن حالت‌های عامل‌های مختلف به یک مقدار واحد رسیده و پایداری سیستم در آن شرایط اتفاق می‌افتد. بنابراین وفاق در این سیستم بین عامل‌های مختلف برقرار است.

## شبکه عصبی RBF

شبکه عصبی RBF (Radial Basis Function) یک نوع شبکه عصبی مصنوعی است که بر اساس توابع پایه شعاعی عمل می‌کند. این شبکه عصبی به طور معمول به مقصود دسته‌بندی، و یا به عنوان تخمینگر توابع مورد استفاده قرار می‌گیرد. نام این شبکه از توابع پایه شعاعی (RBF) الهام گرفته شده است که و شامل پارامترهای مرکز هر تابع و یک شعاع می‌شود. شبکه RBF یک شبکه‌ی دو لایه است: لایه پنهان[[43]](#footnote-43) و لایه خروجی. این شبکه در بخش ورودی اطلاعات ورودی را دریافت می‌کند و آنها را به لایه‌ی پنهان، بدون وزن و مستقیماً منتقل می‌کند. لایه‌ی پنهان شامل تعدادی نورون[[44]](#footnote-44) است که هر نورون دارای تابع فعالت بر پایه‌ی RBF بوده و عملیات محاسباتی را روی ورودی انجام می‌دهد. سپس خروجی‌های این نورون‌ها پس از ضرب شدن در وزن‌های متناظر به لایه خروجی منتقل می‌شوند. در نهایت لایه‌ی خروجی نتیجه نهایی را تولید می‌کند.

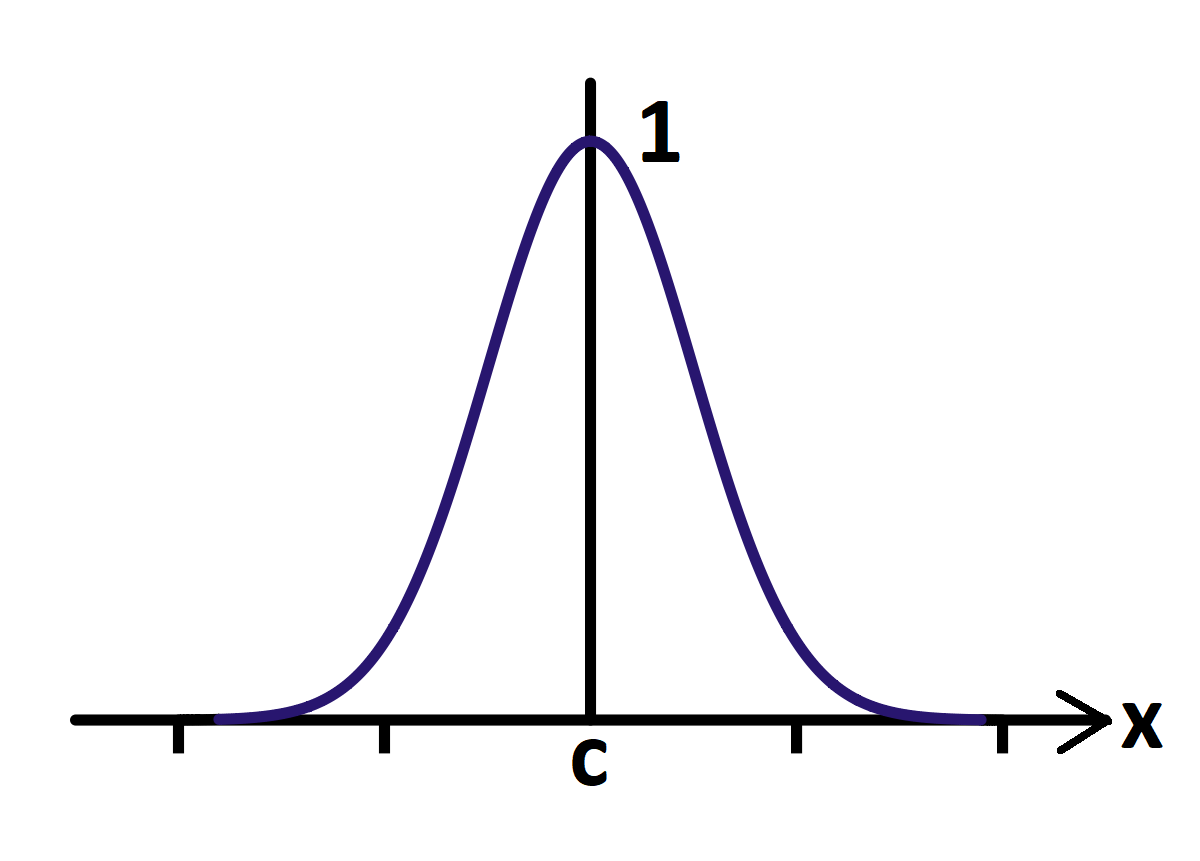


شکل ‏3‑5 ساختار شبکه عصبی RBF

در شکل ۳۱ اگر فرض کنیم ورودی x است و در لایه‌ی پنهان m نورون و در لایه خروجی n خروجی وجود داشته باشد، روابط زیر را داریم.

|  |  |
| --- | --- |
| رابطه 3-10 |  |
| رابطه 3-11 |  |

در رابطه 3-10 هر نورون دارای شعاع r و مرکز مخصوص به خود است. از این پارامترها برای طراحی شبکه عصبی به صورتی که مقادیر دلخواه را در خروجی تولید نماید استفاده می‌شود. همانطور که از رابطه 3-10 برای تابع فعالیت پیداست، این شبکه مقادیر فاصله‌ی را محاسبه نموده، و در ، هایی که فاصله‌ی بیشتری از دارند را مقداری نزدیک به 0 و آن‌هایی که به نزدیکترند مقداری نزدیک به 1 خروجی می‌دهد.



شکل ‏3‑6 تابع فعالیت شبکه عصبی RBF

برای آموزش شبکه عصبی RBF، ابتدا مراکز توابع و رابطه‌ی به‌روزرسانی[[45]](#footnote-45) وزن‌های مربوط به هر تابع را باید مشخص نمود. سپس از روش‌های یادگیری مانند الگوریتم هپس[[46]](#footnote-46) و یا الگوریتم نزول گرادیانی[[47]](#footnote-47) برای بهینه‌سازی وزن‌ها استفاده می‌شود.

شبکه عصبی RBF به خوبی برای مسائلی که با الگوهای غیرخطی سر و کار دارند، مناسب است. به عنوان مثال، برای تشخیص الگوهای صوتی، تشخیص چهره، تخمین تابع پیچیده‌ی غیر خطی و مسائل پیش‌بینی، این شبکه می‌تواند عملکرد مناسبی داشته باشد.

## مدلسازی دینامیکی

در این بخش برای مدلسازی دینامیکی سیستم از روش لاگرانژ استفاده می‌کنیم. این روش بر پایه انرژی نوشته می‌شود که با فرض معادلات قیدهای غیرهولونومیک برای سیستم چند رباته، معادلات لاگرانژ به شکل رابطه‌ی زیر خواهد بود.

|  |  |
| --- | --- |
| رابطه 3-12 |  |

که در رابطه‌ 3-12 فوق، ، تابع لاگرانژین نامیده می‌شود. مقدار برابر با مقدار کل انرژی جنبشی سیستم و برابر با انرژی پتانسیل سیستم می‌باشد. ورودی متناظر با مختصات تعمیم یافته‌ی است که در سیستم ما برابر با ولتاژ ورودی است. بردار ضرایب لاگرانژ[[48]](#footnote-48) متناظر با نیروهای قیدی می‌باشد، و سطر j ام از ماتریس است.

با توجه به این که سیستم ربات چرخدار ما بر روی زمین قرار دارد، مقدار انرژی پتانسیل برابر با 0 خواهد بود. همچنین انرژی جنبشی سیستم به صورت رابطه 3-13 خواهد بود.

|  |  |
| --- | --- |
| رابطه 3-13 |  |

همانطور که در فصل دوم گفته شد . بنابراین معادلات لاگرانژ را برای سیستم ربات چرخدار خود می‌نویسیم و به رابطه 3-14 می‌رسیم.

|  |  |
| --- | --- |
| رابطه 3-14 |  |

رابطه‌ی 3-14 به عنوان مدل دینامیکی ربات شناخته می‌شود. که در آن مقدار ورودی به پارامترهای فیزیکی و دینامیکی ربات مرتبط می‌گردد. در رابطه 3-14 مقادیر ماتریس‌ها به صورت جدول 3-1 است.

جدول ‏3‑1 عناوین ماتریس مدل دینامیکی ربات

|  |  |
| --- | --- |
| نام ماتریس | نماد و مقدار ماتریس |
| ماتریس جرم و ممان |  |
| ماتریس کوریولیس |  |
| ماتریس تبدیل سرعت |  |
| ماتریس گشتاور جاذبه |  |
| ماتریس تبدیل ورودی |  |
| ماتریس اختلال بیرونی[[49]](#footnote-49) |  |
| ماتریس اصطکاک |  |

مقادیر فرض شده برای و در بخش شبیه سازی به صورت عددی ارائه خواهند شد.

همچنین مقادیر پارامترهای فیزیکی در مدل دینامیکی رابطه 3-14 به صورت جدول 3-2 است.

جدول ‏3‑2 نمادها و پارامترهای فیزیکی مدل دینامیکی

|  |  |
| --- | --- |
| پارامتر فیزیکی | نماد |
| جرم ربات | m |
| ممان اینرسی ربات حول محور Z | J |
| شعاع چرخ‌ها | r |
| نصف عرض ربات | R |
| مقاومت درونی آرمیچر |  |
| ثابت گشتاور موتور |  |
| ثابت Back EMF موتور |  |
| نسبت گیربکس |  |
| ولتاژ ورودی به چرخ چپ و راست |  |

در تمامی معادلات بالا اندیس i نشان دهنده‌ي شماره‌ی پلتفرم در سیستم چند رباته می‌باشد و فرض می‌کنیم که L ربات در سیستم ما موجود اند. معادلات برای یک ربات چرخدار استخراج شده و با توجه به فرض یکسان بودن ربات‌ها، قابل تعمیم برای تمامی ربات‌ها در سیستم چند رباته می‌باشند.

حال با توجه به ماتریس تبدیل حالت رابطه 2-11 که در فصل دوم بیان شد، برای تبدیل کردن متغیر معادلات مدل دینامیکی رابطه 3-14 از به ، باید تبدیل‌های زیر را انجام دهیم.

|  |  |
| --- | --- |
| رابطه 3-15 |  |

بنابراین مقادیر جدید ماتریس‌ها به شکل جدول 3-3 خواهند بود.

جدول ‏3‑3 عناوین ماتریس مدل دینامیکی ربات پس از تبدیل

|  |  |
| --- | --- |
| نام ماتریس | نماد و مقدار ماتریس |
| ماتریس جرم و ممان |  |
| ماتریس کوریولیس |  |
| ماتریس تبدیل سرعت |  |
| ماتریس گشتاور جاذبه |  |
| ماتریس تبدیل ورودی |  |
| ماتریس اختلال بیرونی |  |
| ماتریس اصطکاک |  |

با این تبدیلات، مدلسازی سیستم به صورت رابطه 3-16 خواهد بود.

|  |  |
| --- | --- |
| رابطه 3-16 |  |

و می‌دانیم که .

بنابراین، با رابطه 3-16،‌ دینامیک سیستم شبیه سازی و تبدیل شده و می‌توانیم کنترلر متناظر با آن، کنترلر دینامیکی را ارائه نماییم.

## کنترلر دینامیکی

در این قسمت با توجه به مدل دینامیکی ارائه شده در بخش قبل، کنترلر دینامیکی را ارائه می‌کنیم. ابتدا مقادیر تولید شده در فصل دوم در کنترلر سینماتیکی را با نشان می‌دهیم. هدف از کنترلر دینامیکی بدست آوردن u مناسبی است که بتواند سیستم را طوری هدایت کند که آن به برسد.

خطای را به صورت زیر تعریف می‌کنیم. اندیس i نشان دهنده‌ی شماره‌ی ربات در سیستم چند رباته است.

|  |  |
| --- | --- |
| رابطه 3-17 |  |

در رابطه 3-17 مقدار برابر با مقدار ای است که از اعمال ورودی u به مدل دینامیکی 3-16 ارائه شده در بخش قبل به دست می‌آید. سپس با تنها کردن از معادله‌ی 3-16، داریم

|  |  |
| --- | --- |
| رابطه 3-18 |  |

که در آن . حال با توجه به تعریف در رابطه 3-17 و

|  |  |
| --- | --- |
| رابطه 3-19 |  |

بنابراین را در رابطه‌ی 3-18 جایگزین میکنیم. و داریم

|  |  |
| --- | --- |
| رابطه 3-20 |  |

که در آن

|  |  |
| --- | --- |
| رابطه 3-21 |  |

در رابطه 3-21، همان دینامیک غیرخطی غیر قطعی ما است که توسط یک شبکه عصبی RBF آنرا تخمین میزنیم. اگر شبکه‌ی عصبی ما دارای N نورون در لایه‌ی پنهان باشد، آرایه‌ی تابع فعالیت آن یک آرایه‌ی ستونی به شکل زیر رابطه 3-22 است.

|  |  |
| --- | --- |
| رابطه 3-22 |  |

که در آن ورودی شبکه عصبی است. ورودی شبکه‌ی عصبی با توجه به تابع شامل پس‌خور‌های لحظه‌ای از شتاب، سرعت و مکان ربات است.

|  |  |
| --- | --- |
| رابطه 3-23 |  |

مشاهده می‌شود که همان تبدیل یافته‌ی شتاب تولید شده از کنترلر سینماتیک، تبدیل یافته‌ی سرعت تولید شده از کنترلر سینماتیک، تبدیل یافته‌ی سرعت، و تبدیل یافته‌ی مکان ربات است. بنابراین شبکه‌ی عصبی به صورت آنی پس‌خور‌هایی از ربات گرفته و خروجی دینامیک را تولید می‌کند.

همچنین یک ماتریس است.

|  |  |
| --- | --- |
| رابطه 3-24 |  |

بنابراین در لایه‌ی خروجی شبکه‌ی RBF دو خروجی خواهیم داشت. پس وزن‌های شبکه عصبی یک ماتریس خواهد بود به شکل رابطه 3-25

|  |  |
| --- | --- |
| رابطه 3-25 |  |

بنابراین رابطه 3-26 را برای تخمین دینامیک غیرخطی غیر قطعی ربات داریم.

|  |  |
| --- | --- |
| رابطه 3-26 |  |

که در آن برابر با وزن‌های شبکه‌ی عصبی هر یک از ربات‌ها است به طوریکه پس از گذشت زمان به یک مقدار واحد دست یابند. با توجه به آنکه فرض می‌نماییم هر L ربات یکسان و همگن هستند، می‌توان در نظر گرفت که شبکه‌ی عصبی تخمینگر دینامیک آنها، به وزن‌های بسیار مشابهی خواهد رسید. بنابراین در اینجا می‌توانیم از نظریه‌ی وفاق استفاده نماییم. به این صورت که برای به‌روزرسانی وزن‌های شبکه عصبی از رابطه‌ 3-27 زیر استفاده می‌کنیم.

|  |  |
| --- | --- |
| رابطه 3-27 |  |

در رابطه 3-27، برابر با مقداری از وزن شبکه عصبی است که نزدیک به مقدار واحد نهایی است. یعنی

|  |  |
| --- | --- |
| رابطه 3-28 |  |

و مقداری کوچک است. همچنین یک ماتریس قطری برای طراحی ست. و نیز درایه‌های ماتریس مجاورت بوده و بسته و به توپولوژی‌ گراف نمایشگر ارتباط ربات‌های سیستم، تغییر میکند. همانطور که دیده می‌شود، برای به‌روزرسانی وزن‌های شبکه عصبی ربات iام، از خطای ضربدر مقدار خروجی استفاده می‌شود به این صورت که مقدار تغییرات یا همان ، در حالی بزرگتر می‌شود که بزرگتر باشد. یعنی هم که خطای است، و هم . در توضیح خروجی لایه‌ی پنهان باید گفت که همانطور که اشاره شد برابر یک RBF است و در این کاربرد به شکل رابطه 3-29 است.

|  |  |
| --- | --- |
| رابطه 3-29 |  |

بنابراین برای هر می‌بایست یک مرکز با 6 درایه انتخاب نمود. پس مرکزهای لایه‌ی پنهان، نقاطی در یک فضای 6 بعدی خواهند بود. حال می‌بینیم که اگر فاصله‌ی از مراکز انتخاب شده‌ی مورد نظر بیشتر باشد، مقدار نیز بسیار کوچک خواهد شد.

همچنین در رابطه‌ی به‌روزرسانی وزن 3-27، در صورتی که ربات i با ربات j در توپولوژی گراف ارتباطی، ارتباط داشته باشد، برابر با 1 و در غیر این صورت برابر 0 خواهد بود. در صورتی که این دو ربات در ارتباط باشند، ترم دوم رابطه یعنی بیانگر همان قضیه‌ی وفاق است. و اگر مقدار بزرگتری از داشته باشد، حاصل ترم دوم مثبت است و را به سمت سوق می‌دهد. بنابراین، ترم دوم این رابطه کمک می‌کند که وزن‌های شبکه عصبی تمام ربات‌ها به سمت میل کنند.

پیش از آنکه مقدار ورودی کنترلی u را ارائه کنیم، به مفهوم معکوس ماتریس مور-پنروز[[50]](#footnote-50) اشاره می‌کنیم. این معکوس برای ماتریس‌های غیر مربعی تعریف شده و به شبه معکوس[[51]](#footnote-51) نیز معروف است. به طوریکه اگر ماتریس ای مثل را داشته باشیم، مقدار شبه معکوس آنرا با نشان می‌دهیم.

همچنین تعریف زیر را برای ماتریس مور-پنروز داریم.

|  |  |
| --- | --- |
| رابطه 3-30 |  |

حال می‌توانیم ورودی کنترلی u را که یک ماتریس برابر با ولتاژهای موتور چپ و راست است بیان کنیم.

|  |  |
| --- | --- |
| رابطه 3-31 |  |

در رابطه 3-31، از چهار ضریب کنترلی طراحی استفاده شده است. ضرایب ، ، و . مقادیر این ضرایب به این صورت است که یک ماتریس مربعی قطری است. مابقی ضرایب برابر با عدد هستند. این ضرایب برای بهتر عمل نمودن کنترلر در شرایط اولیه‌های متفاوت و مسیر زمانی‌های متفاوت تاثیر مهمی دارند. در رابطه 3-31 اندیس آخر در هر مورد، نشان دهنده‌ی شماره سطر می‌باشد. برای مثال به معنی سطر اول از آرایه‌ی است.

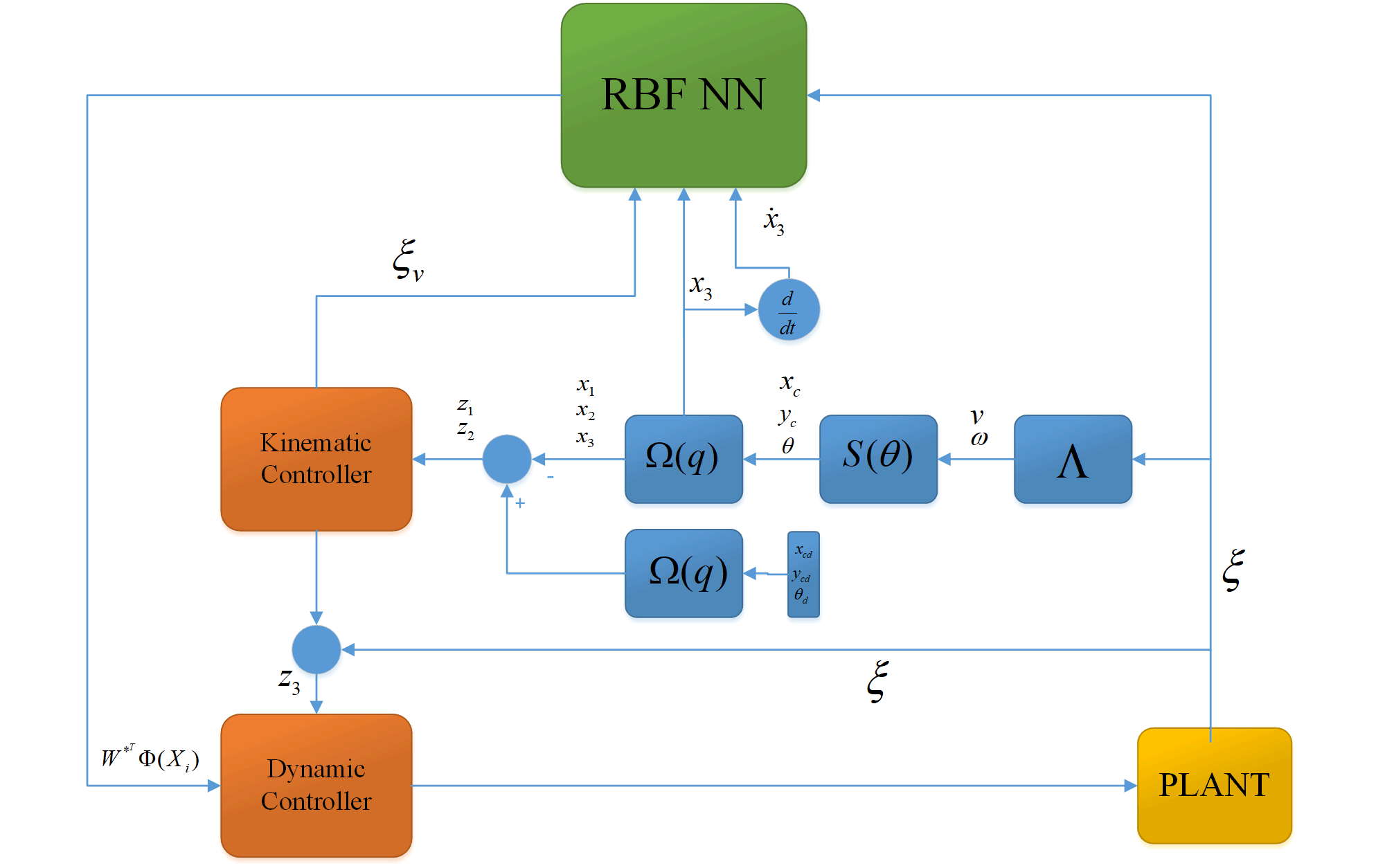
در رابطه‌ 3-31، ترم اول مربوط به شبکه عصبی تخمینگر دینامیک ربات است. همچنین ترم سوم یعنی برابر با ضریب در تابع علامت[[52]](#footnote-52) با ورودی خطای است. که تابع علامت به شکل رابطه‌ 3-32 است.

|  |  |
| --- | --- |
| رابطه 3-32 |  |

همچنین مقادیر و ، مانند آنچه در روابطه 2-19 و 2-20 اشاره شد، برابر با قیودی است که به ترتیب بر روی و اعمال می‌شود. و نیز مقادیر و در رابطه 3-18 تعریف شد.

بنابراین، ربات‌‌ها در حین حرکت، پس‌خورهای متناسب با شبکه عصبی RBF خود، یا همان در رابطه 3-23 را به آن‌ها وارد نموده و خروجی‌های شبکه عصبی پس از ضرب در وزن‌های به‌روزرسانی شده، ترم را می‌سازند. مابقی ترم‌های رابطه 3-32 نیز با پس‌خور و تشکیل ماتریس بدست آمده و نهایتاً u که ولتاژ اعمالی به موتور چپ و راست است، حاصل شده و وارد رابطه 3-16 می‌شود. از این رابطه پس‌خور گرفته شده و را تشکیل داده و چرخه‌ی کنترلی ربات‌ها بسته می‌شود.

همچنین در رابطه‌ی به‌روزرسانی 3-27، با استفاده از ترم دوم مربوط به نظریه‌ی وفاق، ارتباط بین ربات‌ها مفهوم می‌یابد و با توجه به توپولوژی ارتباطی بین ربات‌ها، وزن‌های شبکه عصبی هریک از آن‌ها، به‌روزرسانی می‌شود. برای درک بهتر در شکل 3-7 عملکرد کنترلر دینامیکی آورده شده است.



شکل ‏3‑7 عملکرد کنترلر دینامیکی

به این صورت کنترلر دینامیکی، برپایه‌ی آموزش شبکه‌ی عصبی هر یک از ربات‌ها برای تخمین دینامیک غیرخطی و غیر قطعی خود، و با استفاده از نظریه‌ي وفاق مبتنی بر کنترل همکارانه، تکمیل می‌شود و هر L ربات درون سیستم چند رباته، با خطای کمینه‌ای، به مسیر زمانی مرجع خود میل می‌کنند. همچنین با استفاده از انتخاب تابع لیاپانوف مانع مناسب، می‌توان پایداری لیاپانوف این کنترلر را نیز اثبات نمود.

## کنترلر دینامیکی بر پایه‌ی یادگیری

پس از ارائه‌ی کنترلر دینامیکی، و فرض بر آن که تمامی L ربات‌ یکسان هستند، می‌توان نشان داد که به کمک نظریه‌ی وفاق استفاده شده در رابطه به‌روزرسانی وزن‌های شبکه عصبی در هر ربات، وزن‌های تمامی ربات‌ها به یک مقدار مشابه میل می‌کنند و با گذر زمان و یادگیری شبکه عصبی، خطای تخمین دینامیک ربات توسط شبکه عصبی RBF کمینه شده و در واقع شبکه یاد می‌گیرد در صورت دریافت ورودی چه مقدار ای را به عنوان دینامیک غیرخطی و غیر قطعی هر ربات تولید کند.

بنابراین با حرکت دادن همزمان L ربات، و به‌روزرسانی وزن‌های شبکه عصبی به صورت همکارانه با رابطه 3-27، پس از گذشت زمان مناسب، می‌توان وزن‌های شبکه‌های تمامی ربات‌ها را ذخیره نمود و برای حرکت مجد استفاده کرد. به این صورت عمل میکنیم که اگر ربات‌ها مدت زمان ثانیه را در حال حرکت بوده باشند، و وزن‌های آن‌ها در ذخیره شده باشد، زمان را به‌گونه‌ای انتخاب می‌کنیم که به قدر کافی به نزدیک بوده و . در این محدوده، از رابطه 3-33 استفاده می‌کنیم.

|  |  |
| --- | --- |
| رابطه 3-32 |  |

به این صورت، در مدت زمان از میانگین گیری می‌نماییم. با توجه به این که یک ماتریس است، هر یک از درایه‌های آن به صورت جداگانه در رابطه‌ی 3-32 ظاهر شده، و در واقع از تمامی وزن‌های ربات iام در مدت میانگین گرفته می‌شود.

*حال اگر رابطه 3-31 را با وزن‌های بدست آمده از رابطه 3-32 یا بازنویسی کنیم داریم*

|  |  |
| --- | --- |
| رابطه 3-33 |  |

در رابطه 3-33، ترم اول بعنوان تخمینگر دینامیک ربات، با بازنویسی شده است و در واقع با شروع حرکت ربات‌ iام، وزن‌های شبکه عصبی برابر با است که به علت یادگیری شبکه عصبی، به مقدار صحیح بسیار نزدیک است. این امر موجب آن می‌شود که در حرکت سیستم L ربات، با وجود آنکه مسیر زمانی‌های مرجع متفاوت اند، خطای حرکتی آن‌ها خیلی زودتر به کمینه میل می‌کند.

## شبیه سازی

در این بخش، ابتدا مقادیر عددی لازم برای شبیه سازی دینامیکی آورده می‌شود. نتایج شبیه سازی مانند فصل دوم برای L = 3 ربات در مدت زمان 400 ثانیه ارائه می‌شوند. این مقادیر شامل ضرایب طراحی، مسیر زمانی‌های مرجع برای هر سه ربات و شرایط اولیه می‌باشد. سپس با بررسی خطای بدست آمده از حرکت ربات‌ها نسبت به مسیر زمانی مرجع خودشان، نتیجه حاصل می‌شود که کنترلر ارائه شده با پایدار نمودن ربات‌ها، خطای حرکتی به سمت مسیر زمانی مرجع را کمینه می‌نماید.

### حرکت بیضوی برای سیستم سه رباته

#### مقادیر عددی

مقادیر عددی شبیه سازی، فارغ از شکل مسیر زمانی مرجع که شامل قیود حالت، و ضرایب کنترلی می‌باشند به صورت زیر هستند.

جدول ‏3‑4 مقادیر عددی شبیه سازی دینامیکی

|  |  |  |
| --- | --- | --- |
| مقدار | متغیر | |
| 1.5 m |  |  |
| 1.2 m |  |
| 2 |  |
| 2 |  |
| 0.1 |  |
|  |  |

همچنین برای شبکه عصبی RBF، با توجه به 6 بعدی بودن ورودی شبکه که در رابطه 3-23 آورده شد، ‌می‌بایست مراکز تابع‌های فعالیت را در یک فضای 6 بعدی، به طوری که تغییرات هر یک از درایه‌های را در بر بگیرد، به طور یکنواخت پخش کنیم. برای این منظور مراکز را به صورت زیر انتخاب می‌کنیم.

جدول ‏3‑5 مرکزهای انتخابی برای شبکه عصبی RBF

|  |  |  |
| --- | --- | --- |
| تعداد نورون | محدوده‌ی انتخابی | درایه‌ی |
| 4 |  |  |
| 5 |  |  |
| 3 |  |  |
| 4 |  |  |
| 3 |  |  |
| 3 |  |  |
| مجموع نورون‌های لایه پنهان: | | |

همانطور که پیش تر در رابطه 3-10 اشاره شد، برای تابع فعالیت شبکه عصبی RBF علاوه بر مرکز، به شعاع نیز نیاز است. بنابراین مقادیر باقی‌مانده به شکل جدول زیر است.

جدول ‏3‑6 مقادیر عددی شبیه سازی دینامیکی، مربوط به شبکه عصبی

|  |  |  |
| --- | --- | --- |
| مقدار | متغیر | |
|  |  |  |
| 1.4 |  |
| 2 |  |

حال مسیر‌های زمانی مرجع برای هر ربات را به صورت ارائه شده در جدول 3-7 معین می‌کنیم.

جدول ‏3‑7 مسیرهای زمانی بیضی برای ربات‌های 1، 2 و 3

|  |  |
| --- | --- |
| رابطه 2-38 مسیر زمانی بیضی، ربات 1 |  |
|  |
| رابطه 2-39 مسیر زمانی بیضی، ربات 2 |  |
|  |
| رابطه 2-39 مسیر زمانی بیضی، ربات 3 |  |
|  |

و همچنین شرایط اولیه برای این حرکت به صورت زیر است.

جدول ‏3‑8 شرایط اولیه حرکت بیضوی برای هر سه ربات

|  |  |  |
| --- | --- | --- |
| مقدار اولیه | متغیر | |
|  |  | |
|  |  | |
|  |  | |
|  |  |  |
|  |  |

# فصل چهارم

آموزش شبکه و نتایج

## مقدمه

# فصل پنجم

نتیجه گیری و پیشنهادات

## مقدمه

# مراجع

[1] M. Loey, A. ElSawy, and M. Afify, “Deep learning in plant diseases detection for agricultural crops: A survey,” *Int. J. Serv. Sci. Manag. Eng. Technol.*, vol. 11, no. 2, pp. 41–58, Apr. 2020, doi: 10.4018/IJSSMET.2020040103.

[2] A. Darwish, D. Ezzat, and A. E. Hassanien, “An optimized model based on convolutional neural networks and orthogonal learning particle swarm optimization algorithm for plant diseases diagnosis,” *Swarm Evol. Comput.*, vol. 52, p. 100616, Feb. 2020, doi: 10.1016/j.swevo.2019.100616.

[3] U. N. D. of E. and S. A. P. Division, “World population prospects 2019,” *Dep. Econ. Soc. Aff. World Popul. Prospect. 2019.*, no. 141, 2019.

[4] D. Radovanovic and S. Dukanovic, “Image-Based Plant Disease Detection: A Comparison of Deep Learning and Classical Machine Learning Algorithms,” in *2020 24th International Conference on Information Technology, IT 2020*, IEEE, Feb. 2020, pp. 1–4. doi: 10.1109/IT48810.2020.9070664.

[5] I. Z. Mukti and D. Biswas, “Transfer Learning Based Plant Diseases Detection Using ResNet50,” in *2019 4th International Conference on Electrical Information and Communication Technology, EICT 2019*, IEEE, Dec. 2019, pp. 1–6. doi: 10.1109/EICT48899.2019.9068805.

[6] J. G. A. Barbedo, “A review on the main challenges in automatic plant disease identification based on visible range images,” *Biosyst. Eng.*, vol. 144, pp. 52–60, Apr. 2016, doi: 10.1016/j.biosystemseng.2016.01.017.

[7] S. P. Mohanty, D. P. Hughes, and M. Salathé, “Using deep learning for image-based plant disease detection,” *Front. Plant Sci.*, vol. 7, no. September, Sep. 2016, doi: 10.3389/fpls.2016.01419.

[8] J. G. A. Barbedo, “Factors influencing the use of deep learning for plant disease recognition,” *Biosyst. Eng.*, vol. 172, pp. 84–91, Aug. 2018, doi: 10.1016/j.biosystemseng.2018.05.013.

[9] C. H. Bock, G. H. Poole, P. E. Parker, and T. R. Gottwald, “Plant disease severity estimated visually, by digital photography and image analysis, and by hyperspectral imaging,” *CRC. Crit. Rev. Plant Sci.*, vol. 29, no. 2, pp. 59–107, Mar. 2010, doi: 10.1080/07352681003617285.

[10] J. G. Arnal Barbedo, “Digital image processing techniques for detecting, quantifying and classifying plant diseases,” *Springerplus*, vol. 2, no. 1, pp. 1–12, Dec. 2013, doi: 10.1186/2193-1801-2-660.

[11] G. L. Grinblat, L. C. Uzal, M. G. Larese, and P. M. Granitto, “Deep learning for plant identification using vein morphological patterns,” *Comput. Electron. Agric.*, vol. 127, pp. 418–424, Sep. 2016, doi: 10.1016/j.compag.2016.07.003.

[12] A. Ramcharan, K. Baranowski, P. McCloskey, B. Ahmed, J. Legg, and D. P. Hughes, “Deep learning for image-based cassava disease detection,” *Front. Plant Sci.*, vol. 8, Oct. 2017, doi: 10.3389/fpls.2017.01852.

[13] L. Li, S. Zhang, and B. Wang, “Plant Disease Detection and Classification by Deep Learning - A Review,” *IEEE Access*, vol. 9, pp. 56683–56698, 2021, doi: 10.1109/ACCESS.2021.3069646.

[14] A. Johannes *et al.*, “Automatic plant disease diagnosis using mobile capture devices, applied on a wheat use case,” *Comput. Electron. Agric.*, vol. 138, pp. 200–209, Jun. 2017, doi: 10.1016/j.compag.2017.04.013.

[15] M. C. Enebe and O. O. Babalola, “The impact of microbes in the orchestration of plants’ resistance to biotic stress: a disease management approach,” *Appl. Microbiol. Biotechnol.*, vol. 103, no. 1, pp. 9–25, Jan. 2019, doi: 10.1007/s00253-018-9433-3.

[16] S. Savary, L. Willocquet, S. J. Pethybridge, P. Esker, N. McRoberts, and A. Nelson, “The global burden of pathogens and pests on major food crops,” *Nat. Ecol. Evol.*, vol. 3, no. 3, pp. 430–439, Feb. 2019, doi: 10.1038/s41559-018-0793-y.

[17] A. J. Tatem, D. J. Rogers, and S. I. Hay, “Global Transport Networks and Infectious Disease Spread,” in *Advances in Parasitology*, 2006, pp. 293–343. doi: 10.1016/S0065-308X(05)62009-X.

[18] J. R. Rohr, T. R. Raffel, J. M. Romansic, H. McCallum, and P. J. Hudson, “Evaluating the links between climate, disease spread, and amphibian declines,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 105, no. 45, pp. 17436–17441, Nov. 2008, doi: 10.1073/pnas.0806368105.

[19] T. van der Zwet, “Present Worldwide Distribution of Fire Blight,” *Acta Hortic.*, no. 590, pp. 33–34, Nov. 2002, doi: 10.17660/actahortic.2002.590.1.

[20] S. A. Miller, F. D. Beed, and C. L. Harmon, “Plant disease diagnostic capabilities and networks,” *Annu. Rev. Phytopathol.*, vol. 47, no. 1, pp. 15–38, Sep. 2009, doi: 10.1146/annurev-phyto-080508-081743.

[21] H. Al Hiary, S. Bani Ahmad, M. Reyalat, M. Braik, and Z. ALRahamneh, “Fast and Accurate Detection and Classification of Plant Diseases,” *Int. J. Comput. Appl.*, vol. 17, no. 1, pp. 31–38, Mar. 2011, doi: 10.5120/2183-2754.

[22] X. Bai *et al.*, “Rice heading stage automatic observation by multi-classifier cascade based rice spike detection method,” *Agric. For. Meteorol.*, vol. 259, pp. 260–270, Sep. 2018, doi: 10.1016/j.agrformet.2018.05.001.

[23] K. P. Ferentinos, “Deep learning models for plant disease detection and diagnosis,” *Comput. Electron. Agric.*, vol. 145, pp. 311–318, Feb. 2018, doi: 10.1016/j.compag.2018.01.009.

[24] R. Thapa, K. Zhang, N. Snavely, S. Belongie, and A. Khan, “The Plant Pathology Challenge 2020 data set to classify foliar disease of apples,” *Appl. Plant Sci.*, vol. 8, no. 9, Sep. 2020, doi: 10.1002/aps3.11390.

[25] J. G. Arnal Barbedo, “Plant disease identification from individual lesions and spots using deep learning,” *Biosyst. Eng.*, vol. 180, pp. 96–107, Apr. 2019, doi: 10.1016/j.biosystemseng.2019.02.002.

[26] X. Liu, W. Min, S. Mei, L. Wang, and S. Jiang, “Plant Disease Recognition: A Large-Scale Benchmark Dataset and a Visual Region and Loss Reweighting Approach,” *IEEE Trans. Image Process.*, vol. 30, pp. 2003–2015, 2021, doi: 10.1109/TIP.2021.3049334.

[27] R. Thapa, K. Zhang, N. Snavely, S. Belongie, and A. Khan, “The Plant Pathology Challenge 2020 data set to classify foliar disease of apples,” *Appl. Plant Sci.*, vol. 8, no. 9, Apr. 2020, doi: 10.1002/aps3.11390.

[28] W. Albattah, M. Nawaz, A. Javed, M. Masood, and S. Albahli, “A novel deep learning method for detection and classification of plant diseases,” *Complex Intell. Syst.*, vol. 8, no. 1, pp. 507–524, Feb. 2022, doi: 10.1007/s40747-021-00536-1.

[29] L. Liu *et al.*, “Deep Learning for Generic Object Detection: A Survey,” *Int. J. Comput. Vis.*, vol. 128, no. 2, pp. 261–318, Feb. 2020, doi: 10.1007/s11263-019-01247-4.

[30] M. Goyal, N. D. Reeves, A. K. Davison, S. Rajbhandari, J. Spragg, and M. H. Yap, “DFUNet: Convolutional Neural Networks for Diabetic Foot Ulcer Classification,” *IEEE Trans. Emerg. Top. Comput. Intell.*, vol. 4, no. 5, pp. 728–739, Oct. 2018, doi: 10.1109/tetci.2018.2866254.

[31] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, “A survey of the recent architectures of deep convolutional neural networks,” *Artif. Intell. Rev.*, vol. 53, no. 8, pp. 5455–5516, Dec. 2020, doi: 10.1007/s10462-020-09825-6.

[32] Tee KP, Ge SS, Tay EH (2009) Barrier Lyapunov Functions for

the control of output-constrained nonlinear systems. Automatica

45(4):918–927. https://doi.org/10.1016/j.automatica.2008.11.017

J. Carranza-Rojas, H. Goeau, P. Bonnet, E. Mata-Montero, and A. Joly, “Going deeper in the automated identification of Herbarium specimens,” *BMC Evol. Biol.*, vol. 17, no. 1, p. 181, Dec. 2017, doi: 10.1186/s12862-017-1014-z.

[33] X. Yang and T. Guo, “Machine learning in plant disease research,” *Eur. J. Biomed. Res.*, vol. 3, no. 1, p. 6, Mar. 2017, doi: 10.18088/ejbmr.3.1.2017.pp6-9.

[34] J. G. A. Barbedo, “Impact of dataset size and variety on the effectiveness of deep learning and transfer learning for plant disease classification,” *Comput. Electron. Agric.*, vol. 153, pp. 46–53, Oct. 2018, doi: 10.1016/j.compag.2018.08.013.

[35] M. C. Press, “The functional significance of leaf structure: A search for generalizations,” *New Phytol.*, vol. 143, no. 1, pp. 213–219, Jul. 1999, doi: 10.1046/j.1469-8137.1999.00432.x.

[36] S. Baas, S. Conforti, S. Ahmed, and G. Markova, “The impact of disasters and crises on agriculture and food security: 2021,” *impact disasters Cris. Agric. food Secur. 2021*, 2021, doi: 10.4060/cb3673en.

[37] Y. Toda and F. Okura, “How convolutional neural networks diagnose plant disease,” *Plant Phenomics*, vol. 2019, Jan. 2019, doi: 10.34133/2019/9237136.

[38] J. Boulent, S. Foucher, J. Théau, and P. L. St-Charles, “Convolutional Neural Networks for the Automatic Identification of Plant Diseases,” *Front. Plant Sci.*, vol. 10, Jul. 2019, doi: 10.3389/fpls.2019.00941.

[39] W. E. MacHardy, “Apple Scab: Biology, Epidemiology, and Management,” *St. Paul, MN Am. Phytopathol. Soc. Press.*, 1996.

[40] D. W. L. Manktelow, R. M. Beresford, T. A. Batchelor, and J. T. S. Walker, “Use patterns and economics of fungicides for disease control in New Zealand apples,” *Acta Hortic.*, vol. 422, no. 422, pp. 187–192, Jul. 1996, doi: 10.17660/ActaHortic.1996.422.31.

[41] O. Carisse and J. Bernier, “Effect of environmental factors on growth, pycnidial production and spore germination of Microsphaeropsis isolates with biocontrol potential against apple scab,” *Mycol. Res.*, vol. 106, no. 12, pp. 1455–1462, Dec. 2002, doi: 10.1017/S0953756202006858.

[42] R. M. Beresford and D. W. L. Manktelow, “Economics of reducing fungicide use by weather-based disease forecasts for control of venturis inaequalis in apples,” *New Zeal. J. Crop Hortic. Sci.*, vol. 22, no. 2, pp. 113–120, Jun. 1994, doi: 10.1080/01140671.1994.9513814.

[43] A. L. Jones and T. B. Sutton, “Diseases of Tree Fruits in the East,” *Coop. Ext. Serv. Michigan State Univ. East Lansing.*, p. 98, 1996.

[44] C. E. Main and Gurtz, S., “1987 Crop lossses in North Carolina due to plant diseases and nematodes,” *N. C. State Univ. Dep. Plant Pathol. Spec. Publ.*, vol. 7, p. 183, 1988.

[45] P. M. Miller, “Piles of Apple Prunings as Sources of Conidia of Physalospora obtusa,” *Phytopathology*, vol. 63, no. 8, p. 1080, 1973, doi: 10.1094/phyto-63-1080.

[46] D. Strickland, J. Carrol, and K. Cox, “Cedar Apple Rust,” *New York State Integr. Pest Manag. Program, Subj. Agric. IPM; Fruits; Tree Fruit; Apples*, 2020.

[47] M. A. Ellis, “ Photosynthesis, Transpiration, and Carbohydrate Content of Apple Leaves Infected by Podosphaera leucotricha ,” *Phytopathology*, vol. 71, no. 4, p. 392, 1981, doi: 10.1094/phyto-71-392.

[48] X. M. Xu, “Modelling and forecasting epidemics of apple powdery mildew (Podosphaera leucotricha),” *Plant Pathol.*, vol. 48, no. 4, pp. 462–471, Aug. 1999, doi: 10.1046/j.1365-3059.1999.00371.x.

[49] T. B. Sutton, H. S. Aldwinckle, A. M. Angello, and J. F. Walgenbach, *Compendium of Apple and Pear Diseases and Pests, Second Edition*. The American Phytopathological Society, 2014. doi: 10.1094/9780890544334.

[50] A. Urbanietz and F. Dunemann, “Isolation, identification and molecular characterization of physiological races of apple powdery mildew (Podosphaera leucotricha),” *Plant Pathol.*, vol. 54, no. 2, pp. 125–133, Apr. 2005, doi: 10.1111/j.1365-3059.2005.01156.x.

[51] S. R. Maniyath *et al.*, “Plant disease detection using machine learning,” in *Proceedings - 2018 International Conference on Design Innovations for 3Cs Compute Communicate Control, ICDI3C 2018*, IEEE, Apr. 2018, pp. 41–45. doi: 10.1109/ICDI3C.2018.00017.

[52] W. K. Pratt, “Digital Image Processing,” *Eur. J. Eng. Educ.*, vol. 19, no. 3, p. 377, 1994, doi: 10.1080/03043799408928319.

[53] R. Gonzalez and R. Woods, “A text book on ‘Digital Image Processing,’” *Publ. Pearson*, vol. 2nd Editio, 2002.

[54] S. Phadikar and J. Sil, “Rice disease identification using pattern recognition techniques,” in *Proceedings of 11th International Conference on Computer and Information Technology, ICCIT 2008*, IEEE, Dec. 2008, pp. 420–423. doi: 10.1109/ICCITECHN.2008.4803079.

[55] D. Al Bashish, M. Braik, and S. Bani-Ahmad, “A framework for detection and classification of plant leaf and stem diseases,” in *Proceedings of the 2010 International Conference on Signal and Image Processing, ICSIP 2010*, IEEE, Dec. 2010, pp. 113–118. doi: 10.1109/ICSIP.2010.5697452.

[56] S. H. Lee, H. Goëau, P. Bonnet, and A. Joly, “New perspectives on plant disease characterization based on deep learning,” *Comput. Electron. Agric.*, vol. 170, p. 105220, Mar. 2020, doi: 10.1016/j.compag.2020.105220.

[57] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.

[58] M. Loey, F. Smarandache, and N. E. M. Khalifa, “Within the lack of chest COVID-19 X-ray dataset: A novel detection model based on GAN and deep transfer learning,” *Symmetry (Basel).*, vol. 12, no. 4, p. 651, Apr. 2020, doi: 10.3390/SYM12040651.

[59] D. Rong, L. Xie, and Y. Ying, “Computer vision detection of foreign objects in walnuts using deep learning,” *Comput. Electron. Agric.*, vol. 162, pp. 1001–1010, Jul. 2019, doi: 10.1016/j.compag.2019.05.019.

[60] G. Eraslan, Ž. Avsec, J. Gagneur, and F. J. Theis, “Deep learning: new computational modelling techniques for genomics,” *Nat. Rev. Genet.*, vol. 20, no. 7, pp. 389–403, Jul. 2019, doi: 10.1038/s41576-019-0122-6.

[61] Y. Lecun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015, doi: 10.1038/nature14539.

[62] Y. Peng *et al.*, “DeepSeeNet: A Deep Learning Model for Automated Classification of Patient-based Age-related Macular Degeneration Severity from Color Fundus Photographs,” *Ophthalmology*, vol. 126, no. 4, pp. 565–575, Apr. 2019, doi: 10.1016/j.ophtha.2018.11.015.

[63] I. G. and Y. B. and A. Courville, “Deep learning 简介 一 、 什么是 Deep Learning ？,” *Nature*, vol. 29, no. 7553, pp. 1–73, 2016, [Online]. Available: http://deeplearning.net/

[64] J. P. Shah, H. B. Prajapati, and V. K. Dabhi, “A survey on detection and classification of rice plant diseases,” in *2016 IEEE International Conference on Current Trends in Advanced Computing, ICCTAC 2016*, IEEE, Mar. 2016, pp. 1–8. doi: 10.1109/ICCTAC.2016.7567333.

[65] H. B. Prajapati, J. P. Shah, and V. K. Dabhi, “Detection and classification of rice plant diseases,” *Intell. Decis. Technol.*, vol. 11, no. 3, pp. 357–373, Aug. 2017, doi: 10.3233/IDT-170301.

[66] J. G. A. Barbedo *et al.*, “Annotated plant pathology databases for image-based detection and recognition of diseases,” *IEEE Lat. Am. Trans.*, vol. 16, no. 6, pp. 1749–1757, Jun. 2018, doi: 10.1109/TLA.2018.8444395.

[67] M. Brahimi, M. Arsenovic, S. Laraba, S. Sladojevic, K. Boukhalfa, and A. Moussaoui, “Deep Learning for Plant Diseases: Detection and Saliency Map Visualisation,” 2018, pp. 93–117. doi: 10.1007/978-3-319-90403-0\_6.

[68] T. Wiesner-Hanks *et al.*, “Image set for deep learning: Field images of maize annotated with disease symptoms,” *BMC Res. Notes*, vol. 11, no. 1, p. 440, Dec. 2018, doi: 10.1186/s13104-018-3548-6.

[69] yuanyuan@iim.ac.cn and chenlei@iim.ac.cn, “Image Caption Machine Learning Dataset,” *http://www.icgroupcas.cn/website\_bchtk/index.html*, 2018.

[70] D. Singh, N. Jain, P. Jain, P. Kayal, S. Kumawat, and N. Batra, “PlantDoc: A dataset for visual plant disease detection,” in *ACM International Conference Proceeding Series*, New York, NY, USA: ACM, Jan. 2020, pp. 249–253. doi: 10.1145/3371158.3371196.

[71] Makerere University AI Lab, “Cassava Leaf Disease Classification,” *Kaggle* , 2021, [Online]. Available: https://www.kaggle.com/c/cassava-leaf-disease-classification

[72] J. Laaksonen, M. Koskela, and E. Oja, “PicSOM: Self-Organizing Maps for content-based image retrieval,” in *Proceedings of the International Joint Conference on Neural Networks*, IEEE, 1999, pp. 2470–2473. doi: 10.1109/ijcnn.1999.833459.

[73] G. Wang, Y. Sun, and J. Wang, “Automatic Image-Based Plant Disease Severity Estimation Using Deep Learning,” *Comput. Intell. Neurosci.*, vol. 2017, pp. 1–8, 2017, doi: 10.1155/2017/2917536.

[74] J. Graham, K. Clare, C. Bill, B. Rosie, and T. Lindy, “the Encyclopaedia of Cereal Diseases,” *Encycl. Cereal Dis.*, pp. 42–72, 2008, [Online]. Available: http://www.agricentre.basf.co.uk/agroportal/uk/media/marketing\_pages/cereal\_fungicides/BASF\_Disease\_Encyclopedia.pdf

[75] E. C. Too, L. Yujian, S. Njuki, and L. Yingchun, “A comparative study of fine-tuning deep learning models for plant disease identification,” *Comput. Electron. Agric.*, vol. 161, pp. 272–279, Jun. 2019, doi: 10.1016/j.compag.2018.03.032.

[76] S. K. Noon, M. Amjad, M. A. Qureshi, and A. Mannan, “Overfitting Mitigation Analysis in Deep Learning Models for Plant Leaf Disease Recognition,” in *Proceedings - 2020 23rd IEEE International Multi-Topic Conference, INMIC 2020*, IEEE, Nov. 2020, pp. 1–5. doi: 10.1109/INMIC50486.2020.9318044.

[77] R. Hummel, “Image Enhancement By Histogram Transformation.,” *Comput Graph. Image Process*, vol. 6, no. 2, pp. 184–185, Apr. 1977, doi: 10.1016/s0146-664x(77)80011-7.

[78] G. Yadav, S. Maheshwari, and A. Agarwal, “Contrast limited adaptive histogram equalization based enhancement for real time video system,” in *Proceedings of the 2014 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2014*, IEEE, Sep. 2014, pp. 2392–2397. doi: 10.1109/ICACCI.2014.6968381.

[79] G. Sambasivam and G. D. Opiyo, “A predictive machine learning application in agriculture: Cassava disease detection and classification with imbalanced dataset using convolutional neural networks,” *Egypt. Informatics J.*, vol. 22, no. 1, pp. 27–34, Mar. 2021, doi: 10.1016/j.eij.2020.02.007.

[80] K. Vora and D. Padalia, “An Ensemble of Convolutional Neural Networks to Detect Foliar Diseases in Apple Plants,” Oct. 2021.

[81] A. Yadav, U. Thakur, R. Saxena, V. Pal, V. Bhateja, and J. C. W. Lin, “AFD-Net: Apple Foliar Disease multi classification using deep learning on plant pathology dataset,” *Plant Soil*, vol. 477, no. 1–2, pp. 595–611, Aug. 2022, doi: 10.1007/s11104-022-05407-3.

[82] S. S. Chouhan, U. P. Singh, A. Kaul, and S. Jain, “A Data Repository of Leaf Images: Practice towards Plant Conservation with Plant Pathology,” in *2019 4th International Conference on Information Systems and Computer Networks, ISCON 2019*, IEEE, Nov. 2019, pp. 700–707. doi: 10.1109/ISCON47742.2019.9036158.

[83] S. Parez, N. Dilshad, T. M. Alanazi, and J. Weon Lee, “Towards Sustainable Agricultural Systems: A Lightweight Deep Learning Model for Plant Disease Detection,” *Comput. Syst. Sci. Eng.*, vol. 47, no. 1, pp. 515–536, 2023, doi: 10.32604/csse.2023.037992.

[84] T. A. Prasetyo, V. L. Desrony, H. F. Panjaitan, R. Sianipar, and Y. Pratama, “Corn plant disease classification based on leaf using residual networks-9 architecture,” *Int. J. Electr. Comput. Eng.*, vol. 13, no. 3, pp. 2908–2920, Jun. 2023, doi: 10.11591/ijece.v13i3.pp2908-2920.

[85] S. U. P. U. Stanford Vision Lab, “http://www.image-net.org/,” 2014.

[86] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the Inception Architecture for Computer Vision,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, Jun. 2016, pp. 2818–2826. doi: 10.1109/CVPR.2016.308.

[87] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, Sep. 2015.

[88] M. Tan and Q. V. Le, “EfficientNet: Rethinking model scaling for convolutional neural networks,” *36th Int. Conf. Mach. Learn. ICML 2019*, vol. 2019-June, pp. 10691–10700, May 2019.

[89] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, Jun. 2016, pp. 770–778. doi: 10.1109/CVPR.2016.90.

[90] Z. Liu *et al.*, “Swin Transformer: Hierarchical Vision Transformer using Shifted Windows,” in *Proceedings of the IEEE International Conference on Computer Vision*, IEEE, Oct. 2021, pp. 9992–10002. doi: 10.1109/ICCV48922.2021.00986.

[91] S. Cuenat and R. Couturier, “Convolutional Neural Network (CNN) vs Vision Transformer (ViT) for Digital Holography,” *2022 2nd Int. Conf. Comput. Control Robot. ICCCR 2022*, pp. 235–240, Aug. 2022, doi: 10.1109/ICCCR54399.2022.9790134.

[92] I. Radosavovic, R. P. Kosaraju, R. Girshick, K. He, and P. Dollár, “Designing network design spaces,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, Jun. 2020, pp. 10425–10433. doi: 10.1109/CVPR42600.2020.01044.

[93] M. I. Jordan and T. M. Mitchell, “Machine learning: Trends, perspectives, and prospects,” *Science (80-. ).*, vol. 349, no. 6245, pp. 255–260, Jul. 2015, doi: 10.1126/science.aaa8415.

[94] R. M. Neal, *Pattern Recognition and Machine Learning*, vol. 49, no. 3. Springer, 2007. doi: 10.1198/tech.2007.s518.

[95] J. Hu, H. Niu, J. Carrasco, B. Lennox, and F. Arvin, “Voronoi-Based Multi-Robot Autonomous Exploration in Unknown Environments via Deep Reinforcement Learning,” *IEEE Trans. Veh. Technol.*, vol. 69, no. 12, pp. 14413–14423, Dec. 2020, doi: 10.1109/TVT.2020.3034800.

[96] M. Yoosefzadeh-Najafabadi, H. J. Earl, D. Tulpan, J. Sulik, and M. Eskandari, “Application of Machine Learning Algorithms in Plant Breeding: Predicting Yield From Hyperspectral Reflectance in Soybean,” *Front. Plant Sci.*, vol. 11, Jan. 2021, doi: 10.3389/fpls.2020.624273.

[97] R. Féraud and F. Clérot, “A methodology to explain neural network classification,” *Neural Networks*, vol. 15, no. 2, pp. 237–246, Mar. 2002, doi: 10.1016/S0893-6080(01)00127-7.

[98] Y. Freund and R. E. Schapire, “Large margin classification using the perceptron algorithm,” *Mach. Learn.*, vol. 37, no. 3, pp. 277–296, 1999, doi: 10.1023/A:1007662407062.

[99] J. Schmidhuber, “Annotated History of Modern AI and Deep Learning,” Dec. 2022, [Online]. Available: http://arxiv.org/abs/2212.11279

[100] M. Matsugu, K. Mori, Y. Mitari, and Y. Kaneda, “Subject independent facial expression recognition with robust face detection using a convolutional neural network,” *Neural Networks*, vol. 16, no. 5–6, pp. 555–559, Jun. 2003, doi: 10.1016/S0893-6080(03)00115-1.

[101] M. Bahri, R. Ashino, and R. Vaillancourt, “Convolution theorems for quaternion fourier transform: Properties and applications,” *Abstr. Appl. Anal.*, vol. 2013, pp. 1–10, 2013, doi: 10.1155/2013/162769.

[102] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, “Convolutional neural networks: an overview and application in radiology,” *Insights Imaging*, vol. 9, no. 4, pp. 611–629, Aug. 2018, doi: 10.1007/s13244-018-0639-9.

[103] K. Fukushima, “Cognitron: A self-organizing multilayered neural network,” *Biol. Cybern.*, vol. 20, no. 3–4, pp. 121–136, 1975, doi: 10.1007/BF00342633.

[104] P. Ramachandran, B. Zoph, and Q. V. Le, “Searching for activation functions,” *6th Int. Conf. Learn. Represent. ICLR 2018 - Work. Track Proc.*, Oct. 2018.

[105] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, “Activation Functions: Comparison of trends in Practice and Research for Deep Learning,” Nov. 2018, [Online]. Available: http://arxiv.org/abs/1811.03378

[106] J. Han and C. Moraga, “The influence of the sigmoid function parameters on the speed of backpropagation learning,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 1995, pp. 195–201. doi: 10.1007/3-540-59497-3\_175.

[107] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” *32nd Int. Conf. Mach. Learn. ICML 2015*, vol. 1, pp. 448–456, Feb. 2015.

[108] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[109] G. Ghiasi, T. Y. Lin, and Q. V. Le, “Dropblock: A regularization method for convolutional networks,” *Adv. Neural Inf. Process. Syst.*, vol. 2018-Decem, pp. 10727–10737, Oct. 2018.

[110] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, “Aggregated residual transformations for deep neural networks,” in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, IEEE, Jul. 2017, pp. 5987–5995. doi: 10.1109/CVPR.2017.634.

[111] M. Tan and Q. V. Le, “EfficientNetV2: Smaller Models and Faster Training,” *Proc. Mach. Learn. Res.*, vol. 139, pp. 10096–10106, Apr. 2021.

[112] A. Dosovitskiy *et al.*, “an Image Is Worth 16X16 Words: Transformers for Image Recognition At Scale,” *ICLR 2021 - 9th Int. Conf. Learn. Represent.*, Oct. 2021.

[113] Z. Liu, H. Mao, C. Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, “A ConvNet for the 2020s,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, Jun. 2022, pp. 11966–11976. doi: 10.1109/CVPR52688.2022.01167.

[114] S. Zagoruyko and N. Komodakis, “Wide Residual Networks,” *Br. Mach. Vis. Conf. 2016, BMVC 2016*, vol. 2016-Septe, pp. 87.1-87.12, May 2016, doi: 10.5244/C.30.87.

[115] M. Lukasik, S. Bhojanapalli, A. K. Menon, and S. Kumar, “Does label smoothing mitigate label noise?,” *37th Int. Conf. Mach. Learn. ICML 2020*, vol. PartF16814, pp. 6404–6414, Mar. 2020.

[116] W.-T. Pan, “A new Fruit Fly Optimization Algorithm: Taking the financial distress model as an example,” *Knowledge-Based Syst.*, vol. 26, pp. 69–74, Feb. 2012, doi: 10.1016/j.knosys.2011.07.001.

[117] S. Mirjalili, S. M. Mirjalili, and A. Lewis, “Grey Wolf Optimizer,” *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014, doi: 10.1016/j.advengsoft.2013.12.007.

[118] S. Mirjalili, “The Ant Lion Optimizer,” *Adv. Eng. Softw.*, vol. 83, pp. 80–98, May 2015, doi: 10.1016/j.advengsoft.2015.01.010.

[119] S. Mirjalili, “Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm,” *Knowledge-Based Syst.*, vol. 89, pp. 228–249, Nov. 2015, doi: 10.1016/j.knosys.2015.07.006.

[120] R. Venkata Rao, “Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems,” *Int. J. Ind. Eng. Comput.*, pp. 19–34, 2016, doi: 10.5267/j.ijiec.2015.8.004.

[121] S. Mirjalili and A. Lewis, “The Whale Optimization Algorithm,” *Adv. Eng. Softw.*, vol. 95, pp. 51–67, May 2016, doi: 10.1016/j.advengsoft.2016.01.008.

[122] S. Mirjalili, “Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems,” *Neural Comput. Appl.*, vol. 27, no. 4, pp. 1053–1073, May 2016, doi: 10.1007/s00521-015-1920-1.

[123] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, and S. M. Mirjalili, “Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems,” *Adv. Eng. Softw.*, vol. 114, pp. 163–191, Dec. 2017, doi: 10.1016/j.advengsoft.2017.07.002.

[124] M. Bakhshipour, M. Jabbari Ghadi, and F. Namdari, “Swarm robotics search & rescue: A novel artificial intelligence-inspired optimization approach,” *Appl. Soft Comput.*, vol. 57, pp. 708–726, Aug. 2017, doi: 10.1016/j.asoc.2017.02.028.

[125] R. Masadeh, B. A., and A. Sharieh, “Sea Lion Optimization Algorithm,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 5, 2019, doi: 10.14569/IJACSA.2019.0100548.

[126] H. Yapici and N. Cetinkaya, “A new meta-heuristic optimizer: Pathfinder algorithm,” *Appl. Soft Comput.*, vol. 78, pp. 545–568, May 2019, doi: 10.1016/j.asoc.2019.03.012.

[127] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, “Harris hawks optimization: Algorithm and applications,” *Futur. Gener. Comput. Syst.*, vol. 97, pp. 849–872, Aug. 2019, doi: 10.1016/j.future.2019.02.028.

[128] L. Abualigah, D. Yousri, M. Abd Elaziz, A. A. Ewees, M. A. A. Al-qaness, and A. H. Gandomi, “Aquila Optimizer: A novel meta-heuristic optimization algorithm,” *Comput. Ind. Eng.*, vol. 157, p. 107250, Jul. 2021, doi: 10.1016/j.cie.2021.107250.

[129] O. O. Obadina, M. A. Thaha, K. Althoefer, and M. H. Shaheed, “Dynamic characterization of a master–slave robotic manipulator using a hybrid grey wolf–whale optimization algorithm,” *J. Vib. Control*, vol. 28, no. 15–16, pp. 1992–2003, Aug. 2022, doi: 10.1177/10775463211003402.

[130] A. Faramarzi, M. Heidarinejad, S. Mirjalili, and A. H. Gandomi, “Marine Predators Algorithm: A nature-inspired metaheuristic,” *Expert Syst. Appl.*, vol. 152, p. 113377, Aug. 2020, doi: 10.1016/j.eswa.2020.113377.

[131] F. A. Hashim, E. H. Houssein, K. Hussain, M. S. Mabrouk, and W. Al-Atabany, “Honey Badger Algorithm: New metaheuristic algorithm for solving optimization problems,” *Math. Comput. Simul.*, vol. 192, pp. 84–110, Feb. 2022, doi: 10.1016/j.matcom.2021.08.013.

[132] A. Seyyedabbasi and F. Kiani, “Sand Cat swarm optimization: a nature-inspired algorithm to solve global optimization problems,” *Eng. Comput.*, vol. 39, no. 4, pp. 2627–2651, Aug. 2023, doi: 10.1007/s00366-022-01604-x.

[133] L. Xie, T. Han, H. Zhou, Z.-R. Zhang, B. Han, and A. Tang, “Tuna Swarm Optimization: A Novel Swarm-Based Metaheuristic Algorithm for Global Optimization,” *Comput. Intell. Neurosci.*, vol. 2021, pp. 1–22, Oct. 2021, doi: 10.1155/2021/9210050.

[134] L. Wang, Q. Cao, Z. Zhang, S. Mirjalili, and W. Zhao, “Artificial rabbits optimization: A new bio-inspired meta-heuristic algorithm for solving engineering optimization problems,” *Eng. Appl. Artif. Intell.*, vol. 114, p. 105082, Sep. 2022, doi: 10.1016/j.engappai.2022.105082.

# واژه نامه

|  |  |
| --- | --- |
|  |  |
| 5-Fold Cross Validation | ارزیابی 5-قسمتی |
| Accuracy | دقت |
| Agent | نماینده |
| Apple Foliar Disease Network | بیماری‌های قارچی |
| Apple Scab | زخم سیب |
| Artificial Intelligence | هوش مصنوعی |
| Artificial Neural Network | شبکه عصبی مصنوعی |
| Automatic | خودکار |
| Average Pooling | ادغام میانگین |
| Backpropagation Algorithm | الگوریتم پس‌انتشار |
| Batch normalization | نرمالسازی دسته |
| Batch Size | سایز دسته |
| Bayes Classifier | طبقه‌بند بایزی |
| Bias | بایاس |
| Binary Classification | طبقه‌بندی دوتایی |
| Black Rot | لکه‌ی سیاه |
| Bottleneck | گلوگاه |
| Cardinality | کاردینالیته |
| Cedar Apple Rust | زنگار سیب-سرو |
| Chloroplast | کلروپلاست |
| Class | طبقه |
| Classification | طبقه‌بندی |
| Class-weight | طبقه-وزن |
| Cluster-based Region Reweighting | وزن‌بندی ناحیه ای خوشه ای |
| Complex | پیچیده |
| Compound Coefficient | ضریب ترکیب |
| Computational Intelligence | هوش محاسباتی |
| Computer Science | علوم کامپیوتر |
| Convolutional Neural Network | شبکه‌ی عصبی پیچشی |
| Cuticle | کوتیکول |
| Cutmix | برش-ترکیب |
| Cutout | برش تصویر |
| Data-Augmentation | داده‌افزایی‌ |
| Dataset | مجموعه‌داده |
| Deep Learning | یادگیری عمیق |
| Deep Neural Network | شبکه عصبی عمیق |
| Defoliation | برگ‌زدایی |
| Pollinators | گرده‌افشانی |
| Distribution | پراکندگی |
| Drone | پهپاد |
| Drop Block | بلوک برون‌انداز |
| Dropout | برون‌انداز |
| Dying | مرگ |
| Ensemble | ترکیب مدل |
| Epidermis | اپی‌درم |
| Epoch | دوره |
| False Negative | منفی غلط |
| False Positive | مثبت غلط |
| Feature Map | نقشه ویژگی |
| Features | ویژگی‌ها |
| Fine-Grained Visual Categorization | طبقه‌بندی بصری دقیق |
| Fine-Tune | تنظیم دقیق |
| Fitness Function | تابع تناسب |
| Flip | آیینه کردن |
| Float | اعشاری |
| Food security | امنیت غذایی |
| Frame | قاب |
| Frog Eye Leaf Spot | لکه برگ چشم غورباقه‌ای |
| Fully Connected Layer | تمام متصل |
| Gas Chromatography | رنگ‌شناسی گازی |
| Gaussian Blur | تارسازی گوسی |
| Gaussian error Linear Unit | واحد خطی خطای گوسی |
| Generative Adaptive Neural Network | شبکه‌ی مولد تخاصمی |
| Genetic Algorithm | الگوریتم ژنتیک |
| Global Average Pooling | ادغام میانگین کلی |
| Hierarchial | سلسله مراتبی |
| Hyperparameter | ابرپارامتر |
| Image Analysis | بررسی تصویر |
| Image Enhancement | بهبود تصویر |
| Image Preprocessing | پیش‌پردازش تصویر |
| Image Processing | پردازش تصویر |
| Image Sensors | حسگر‌های تصویر |
| Image Transformation | تبدیل تصویر |
| Inverted Bottleneck | گللوگاه معکوس |
| Kernel | کرنل |
| K-means clustering | خوشه‌بندی ک-میانگین |
| Label | برچسب |
| Lambda | لامبدا |
| Learning From Scratch | یادگیری از ابتدا |
| Learning Rate | نرخ یادگیری |
| Learning Rate Scheduler | تغییردهنده نرخ یادگیری |
| Long Short-Term Memory | حافظه بلند کوتاه‌مدت |
| Loss | هزینه |
| Loss Function | تابع هزینه |
| Machine Learning | یادگیری ماشین |
| Max Pooling | ادغام بیشینه |
| Median Blur | تارسازی میانه‌ای |
| Mega Flop | مگافلاپ |
| Mesophyll | مزوفیل |
| Microorganisms | ریزجانداران |
| Mixup | مخلوط‌کردن |
| Movement Strategies | استراتژی‌های حرکتی |
| Multi-Class Classification | طبقه‌بندی چند طبقه |
| Multi-Label Classification | طبقه‌بندی چند برچسبه |
| Multiple Disease | چندین بیماری |
| Natural Laguage Proccessing | پردازش زبان طبیعی |
| Neighborhood Structure | ساختار همسایگی |
| Node | گره |
| Northern Leaf Blight | لهیدگی شمالی برگ |
| Open-Source | متن - باز |
| Optimizer | بهینه‌ساز |
| Overfitting | بیش‌برازش |
| Padding | پَدینگ |
| Patch | بخش |
| Perceptron | پرسپترون |
| Pesticide | آفت‌کش |
| Photosynthesis | فتوسنتز |
| Polymerase Chain Reaction | واکنش زنجیره‌ای پلیمراز |
| Pooling | ادغام |
| Powdery Mildew | کپک پودری |
| Progressive Training | آموزش پیشرفتی |
| RAM | حافظه موقت |
| Raw Data | داده‌ی خام |
| Reccurrent Neural Networks | شبکه‌های عصبی بازگشتی |
| Rectified Linear Unit | واحد یکسوساز خطی |
| Regularization | تنوع‌سازی |
| Reinforcement Learning | یادگیری تقویتی |
| RGB code | کد قرمز – سبز – آبی |
| Rust | زنگار |
| Semi-Supervised Learning | یادگیری نیمه نظارت |
| Shifted Windows Transformer | مبدل پنجره‌ای جابه‌جا شده |
| Shift-Scale | انتقال مرکز |
| Sigmoid | سیگموید |
| Softmax | بیشینه هموار |
| Spores | اسپور |
| SSD | حافظه جامد |
| Step Activation Function | تابع فعال‌سازی پله‌ای |
| Supervised Learning | یادگیری با نظارت |
| Swarm Intelligence | هوش ازحامی |
| Swarm-based Optimization Algorithm | الگوریتم بهینه‌سازی ازدحامی |
| Swish | سوویش |
| Tensor | تنسور |
| Test | آزمون |
| TF Record | کد رکوردی |
| Threshold | آستانه |
| Thresholidng | آستانه‌گذاری |
| Training with Loss Reweighting | آموزش با وزن بندی اختلاف خطا |
| Transformers | مبدل‌ها |
| Triangular Learning Rate Scheduler | تنظیم‌کننده نرخ یادگیری مثلثی |
| True Positive | مثبت صحیح |
| TrueNegative | مثبت غلط |
| Unsupervised Learning | یادگیری بدون نظارت |
| Update | به‌روزرسانی |
| User Interface | رابط کاربری |
| Validation | اعتبارسنجی |
| Vanishing Gradient | محوشدگی گرادیان |
| VRAM | حافظه گرافیکی |
| Weakly-Supervised Training | آموزش نظارتی ضعیف |
| Weight Decay | کاهش وزن |
| Weighted Feature Integration | اعمال ویژگی‌های وزن بندی شده |

**Abstract:**

Plant diseases severrly affect agricultural products and cause their destruction, it is necessary to identify and deal with as soon as possible. One of the important organs of a plant is its leaf; which is crucial for it’s survival and growth. The traditional methods of identifying plant leaf diseases include specialists visiting the farm or orchard and observing a large number of plant leaves by eye or sending samples of the leaf to the laboratory. These methods are time-consuming and often cost a lot. With the remarkable advancements in Image Processing and Deep Learning, it is possible to design systems capable of automatically classifying leaf diseases and diagnosing them with high accuracy. In this research, ten Convolutional Neural Networks and Image Transformers are investigated to find the best performance. The dataset used to train, validate and test the models is Plant Pathology 2021, which contains 18632 images of apple leaves in 12 classes. The classification method used in this research is Multi-Label Claassification. The structures of models have been modified and Dropout, DropBlock and a 6 layer fully-connected layer have been added to the networks. Best performance was achieved by Resnext5032x4D, SwinV2T and SwinV2S with an accuracy of 92.62%, 92.22% and 92.51% and average F-1 scores of 95.31%, 94.77% and 94.75%, respectively. Another test has been carried out in which the thresholds of the models are changed by Swarm Optimization Algorithms. 19 of these algorithms have been chosen which had two parameters. After finding the optimized thresholds, an increase in accuracy in all the networks and an increase in F-1 score in most the models were observed. The best results after the optimization are for the models Resnext5032x4D, SwinV2T and SwinV2S with accuracies of 93.32%,92.18% and 93% and F-1 scores of 95.4%, 94.96% and 94.88%, respectively. Also the optimization algorithms were ranked according to their performance on the networks and the best ones were chosen. The training methods and newly introduced models have better performance on the dataset than previous reasearches.

**Keywords:** Deep Learning, Convolutional Neural Networks, Image Transformers, Swarm Optimization Algorithms, DropBlock, Precision Agriculture, Image Processing, Plant Pathology Systems

|  |  |  |
| --- | --- | --- |
|  | P1969#yIS1 |  |
| University of Tehran  Faculty of New Sciences and Technologies  Department of Interdisciplinary Technology | | |
| Title:  Classification of Plant Leaves Diseases Based on Convolutional Neural Network Learning | | | |
| By:  Soroush Toutounchian | | | |
| Supervisor:  Dr. Alireza Rezaee  Advisor: Dr. Farshid Hajati | | | |
|  | | | |
| A Thesis Submitted to the Graduate Office in Fulfillment  of Requirements for the Degree of Master of Science  in Mechatronics Engineering | | | |
| September 2023 | | | |

1. Multi-robot [↑](#footnote-ref-1)
2. Cooperative [↑](#footnote-ref-2)
3. Trajectory tracking [↑](#footnote-ref-3)
4. Network [↑](#footnote-ref-4)
5. Uncertainty [↑](#footnote-ref-5)
6. Legged Mobile Robots [↑](#footnote-ref-6)
7. Wheeled Mobile Robots [↑](#footnote-ref-7)
8. RPM [↑](#footnote-ref-8)
9. Multi-Robot system [↑](#footnote-ref-9)
10. Trajectory tracking [↑](#footnote-ref-10)
11. Framework [↑](#footnote-ref-11)
12. Formation [↑](#footnote-ref-12)
13. Consensus [↑](#footnote-ref-13)
14. Asymptotically [↑](#footnote-ref-14)
15. Uncertainty [↑](#footnote-ref-15)
16. Distributed control systems [↑](#footnote-ref-16)
17. Leader [↑](#footnote-ref-17)
18. Deterministic learning theory [↑](#footnote-ref-18)
19. Constraints [↑](#footnote-ref-19)
20. State constraints [↑](#footnote-ref-20)
21. Model predictive control [↑](#footnote-ref-21)
22. Error transformation function [↑](#footnote-ref-22)
23. Barrier Lyapunov function [↑](#footnote-ref-23)
24. Feedback [↑](#footnote-ref-24)
25. Implementing [↑](#footnote-ref-25)
26. Uncertain [↑](#footnote-ref-26)
27. Radial basis function [↑](#footnote-ref-27)
28. Equilibrium point [↑](#footnote-ref-28)
29. Aleksandr Lyapunov [↑](#footnote-ref-29)
30. Markov chains [↑](#footnote-ref-30)
31. State space [↑](#footnote-ref-31)
32. Conservation laws [↑](#footnote-ref-32)
33. Global [↑](#footnote-ref-33)
34. Continuous [↑](#footnote-ref-34)
35. Homeomorphism [↑](#footnote-ref-35)
36. Cooperative Control [↑](#footnote-ref-36)
37. Multi agent [↑](#footnote-ref-37)
38. Directed graph [↑](#footnote-ref-38)
39. Undirected graph [↑](#footnote-ref-39)
40. Adjacency matrix [↑](#footnote-ref-40)
41. Consensus theory [↑](#footnote-ref-41)
42. Consensus theory [↑](#footnote-ref-42)
43. Hidden layer [↑](#footnote-ref-43)
44. Neuron [↑](#footnote-ref-44)
45. Update [↑](#footnote-ref-45)
46. Hebbian algorithm [↑](#footnote-ref-46)
47. Gradient Descent [↑](#footnote-ref-47)
48. Lagrange's multiplyer [↑](#footnote-ref-48)
49. External disturbance [↑](#footnote-ref-49)
50. Moore–Penrose [↑](#footnote-ref-50)
51. Pseudo inverse [↑](#footnote-ref-51)
52. Sign [↑](#footnote-ref-52)