دانشگاه صنعتی امیرکبیر
(پلی‌تکنیک تهران)
دانشکده مهندسی برق

پروژه کارشناسی

گرایش مخابرات

ماشین های خودران-
با استفاده از یادگیری تقویتی

نگارش

محمد رضیئی فیجانی

استادان راهنما

دکتر وحید پوراحمدی و دکتر حمیدرضا امین‌داور

شهریور ۱۳۹۸


بسم الله الرحمن الرحيم

صفحه فرم ارزیابی و تصویب پایان نامه- فرم تأیید اعضاء کمیته دفاع

در این صفحه فرم دفاع یا تایید و تصویب پایان نامه موسوم به فرم کمیته دفاع- موجود در پرونده آموزشی- را قرار دهید.

نکات مهم:

- نگارش پایان نامه/رساله باید به زبان فارسی و بر اساس آخرین نسخه دستورالعمل و راهنمای تدوین پایان نامه های دانشگاه صنعتی امیرکبیر باشد.(دستورالعمل و راهنمای حاضر)
- رنگ جلد پایان نامه/رساله چاپی کارشناسی، کارشناسی ارشد و دکترا باید به ترتیب مشکی، طوسی و سفید رنگ باشد.
- چاپ و صحافی پایان نامه/رساله بصورت پشت و رو(دورو) بلامانع است و انجام آن توصیه می شود.

دانشگاه صنعتی امیرکبیر
(پلی‌تکنیک تهران)

به نام خدا

تعهدنامه اصالت اثر

تاریخ: شهریور ۱۳۹۸

اینجانب **محمد رضیئی فیجانی** متعهد می‌شوم که مطالب مندرج در این پایان‌نامه حاصل کار پژوهشی اینجانب تحت نظارت و راهنمایی اساتید دانشگاه صنعتی امیرکبیر بوده و به دستاوردهای دیگران که در این پژوهش از آنها استفاده شده است مطابق مقررات و روال متعارف ارجاع و در فهرست منابع و مآخذ ذکر گردیده است. این پایان‌نامه قبلاً برای احراز هیچ مدرک هم‌سطح یا بالاتر ارائه نگردیده است.

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

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

محمد رضیئی فیجانی

امضا

نویسنده پایان نامه، در صورت تمایل میتواند برای سپاسگزاری پایان نامه خود را به شخص یا اشخاص و یا ارگان خاصی تقدیم نماید.

سپاس‌گزاری

نویسنده پایان‌نامه می‌تواند مراتب امتنان خود را نسبت به استاد راهنما و استاد مشاور و یا دیگر افرادی که طی انجام پایان‌نامه به نحوی او را یاری و یا با او همکاری نموده‌اند ابراز دارد.

محمدرضیئ فیجانی

شهریور ۱۳۹۸

چکیده

در این قسمت چکیده پایان نامه نوشته می‌شود. چکیده باید جامع و بیان‌کننده خلاصه‌ای از اقدامات انجام‌شده باشد. در چکیده باید از ارجاع به مرجع و ذکر روابط ریاضی، بیان تاریخچه و تعریف مسئله خودداری شود.

واژه‌های کلیدی:

کلیدواژه اول، ...، کلیدواژه پنجم (نوشتن سه تا پنج واژه کلیدی ضروری است)

فهرست مطالب

عنوان صفحه

فهرست اشکال

شکل صفحه

فهرست جداول

جدول صفحه

فهرست نمادها

نماد	مفهوم
$\mathbb{R}^n$	فضای اقلیدسی با بعد n
$\mathbb{S}^n$	کره یکه n بعدی
M^m	خمینه m-بعدی M
$\mathfrak{X}(M)$	جبر میدان‌های برداری هموار روی M
$\mathfrak{X}^1(M)$	مجموعه میدان‌های برداری هموار یکه روی (M,g)
$\Omega^p(M)$	مجموعه p-فرمی‌های روی خمینه M
Q	اپراتور ریچی
$\mathcal{R}$	تانسور انحنای ریمان
ric	تانسور ریچی
L	مشتق لی
Φ	۲-فرم اساسی خمینه تماسی
∇	التصاق لوی-چویتای
Δ	لاپلاسین ناهموار
∇^*	عملگر خودالحاق صوری القا شده از التصاق لوی-چویتای
g_s	متر ساساکی
∇	التصاق لوی-چویتای وابسته به متر ساساکی
Δ	عملگر لاپلاس-بلترامی روی p-فرم‌ها

فصل اول

یادگیری تقویتی با استفاده از gym

۱-۱ معرفی مفاهیم یادگیری تقویتی

۱-۲ معرفی OpenAI gym

۱-۲-۱ مقدمه

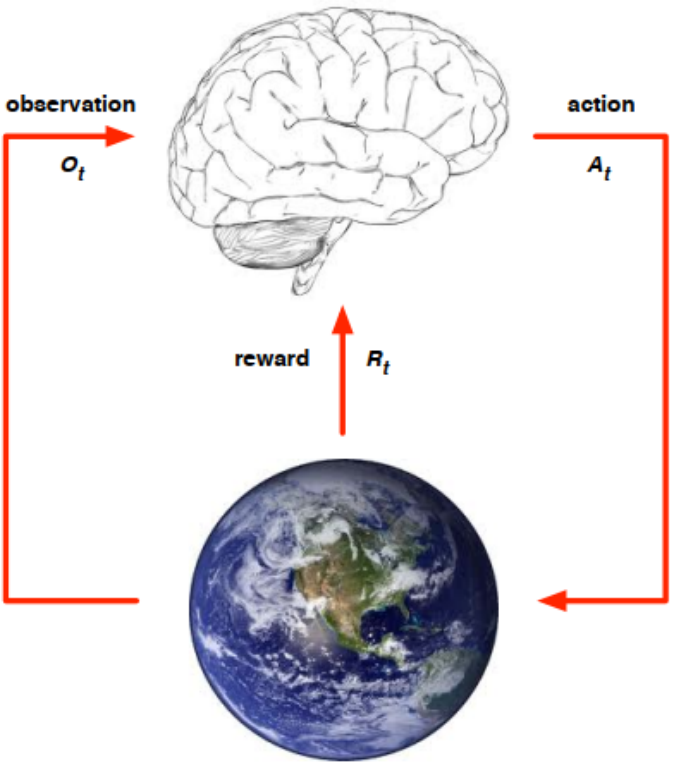
پروژه gym از قوی ترین پروژه های Open AI [1] می‌باشد.

۱-۲-۲ نصب

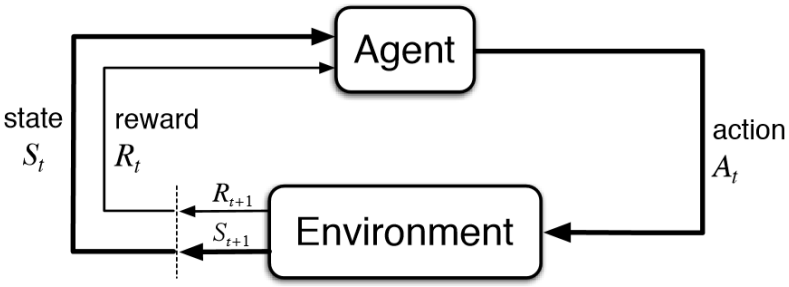
[1] https://github.com/openai

فصل دوم

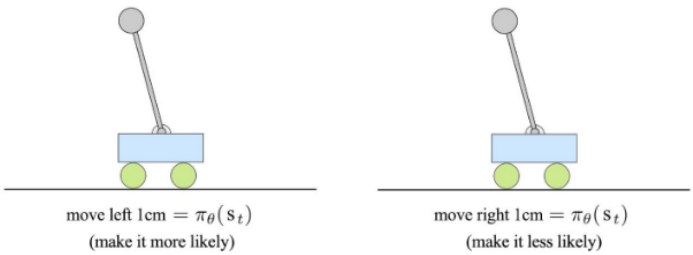
معرفی یادگیری ماشین



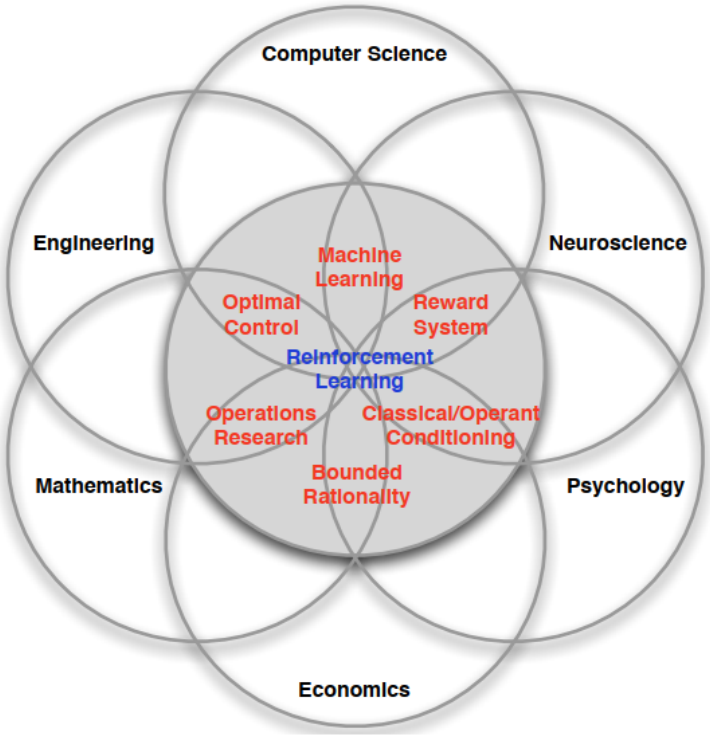
شکل ۲-۱:



شکل ۲-۲:



شکل ۲-۳:


Computer Science
Engineering
Neuroscience
Machine Learning
Optimal Control
Reward System
Reinforcement Learning
Operations Research
Classical/Operant Conditioning
Bounded Rationality
Mathematics
Psychology
Economics

شکل ۲-۴:

فصل سوم

جزئیات فنی پروژه

۳-۱ مقدمه

در این پروژه از جهت آنکه نسخه قبلی و پیشینی برای آن نبوده است، به ناچار می‌بایست که کد آن از صفر تا صد آن به صورت دستی نوشته شود. از این‌رو، پیچیدگی های بسیار فراوان را به طور خاص در پی داشت. ابزار های زیادی نیز بنابه شرایط در آن استفاده شد که ارتباط بین آن ابزار ها و اجزا، بر این پیچیدگی پیاده سازی طرح افزوده بود.

ابزار های اصلی و کلی که در این پروژه استفاده شده بود، عبارتند از:

- **نرم افزار پری اسکن** [۱] ، نسخه 8.5.0
- **نرم افزار قدرتمند متلب** [۲] ، نسخه R2017b
- **زبان برنامه نویسی پایتون** ، نسخه 3.6.9

بنابراین برای راه اندازی مجدد کد این پروژه لازم است که موارد بالا روی کامپیوتر شخص به صورت کامل نصب باشد.

همچنین لازم به ذکر است که برخی ابزارات دیگر نیز در این پروژه استفاده شده است که احتمالا با نصب موارد بالا دیگر نیازی به نصب آن ها به صورت جداگانه نیست. هدف این ابزار ها ایجاد اتصال بین اجزای اصلی گفته شده است. این گروه شامل موارد زیر هستند:

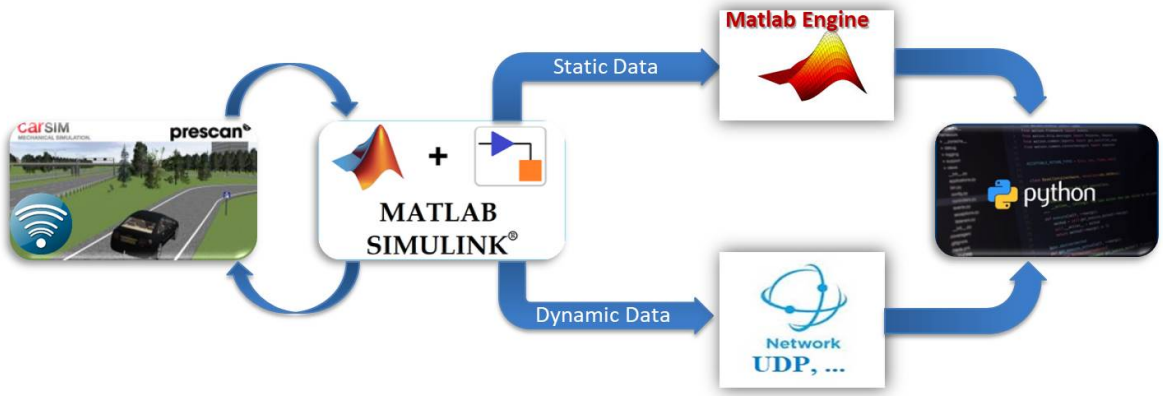
- **سیمولینک** [۳] ، جهت اتصال بین متلب و پری اسکن
- **شبکه UDP** [۴] ، جهت اتصال داده های پویا [۵] بین پایتون و سیمولینک
- **موتور متلب** [۶] ، جهت اتصال داده های ساکن [۷] بین پایتون و سیمولینک

در این فصل جزئیات بیشتری در مورد لزوم و دلیل استفاده از این ابزار ها بررسی می‌شود.

[1] PreScan
[2] Matlab
[3] Simulink
[۴] برای این منظور از ماژول socket در پایتون استفاده شده است.
[5] Dynamic Data
[6] Matlab Engine
[7] Static Data

۳-۲ دورنمای کلی طرح

همان‌طور که گفته شد، در این پروژه از ابزار های مختلفی استفاده شده است. برخی ابزارات دیگر نیز جهت ایجاد اتصال بین آن ابزار ها استفاده شده اند. در این بخش، این اجزا به تفصیل بررسی خواهد شد. هر کدام از این اجزا کار مشخصی را بر عهده دارند. شکل ۳-۱ این ارتباط را نشان می‌دهد.



شکل ۳-۱: بلوک دیالگرام لایه های کلی

در شکل ۳-۱ از سمت چپ به راست اجزا یاد شده و نحوه ارتباط آن‌ها بایک‌دیگر را به‌خوبی نشان می‌دهد. این بلاک ها و ارتباط ها عبارتند از:

- اولین بلاک آن، نرم افزار **پری‌اسکن** می‌باشد. وظیفه اصلی این نرم افزار، شبیه سازی دینامیک یک اتومبیل و یا موتور و ... می‌باشد. همچنین ایجاد یک محیط گرافیکی زیبا و یک پنل کاربری گرافیکی برای ساخت ماشین ها از دیگر حسن های این نرم افزار است.

  فایل های مهم ایجاد شده توسط این بخش، `.pex` و `.pb` می‌باشد.

- بلاک بعدی ترکیبی از **متلب و سیمولینک** است. چرا که نرم افزار پری‌اسکن این امکان را دارد که برای کنترل و دسترسی بیشتر به قسمت های کنترلی مختلف، چیزی به نام API ارائه می‌دهد. این API یک فایل سیمولینک را در اختیار کابران قرار میدهد که در آن بلوک های مشخصی به یکدیگر متصل هستند و با مطالعه و تغییر آن بلوک ها می‌توان کنترل سیستم را به دست گرفت.

  فایل های مهم این بخش نیز در فرمت `.slx` و `.m` در دسترس هستند.

  همچنین API یاد شده، دستورات دیگری را جهت دریافت داده های استاتیک محیط ساخته شده در این نرم افزار را به کاربران خویش در محیط متلب می‌دهد.

- دو بلوک بعدی، مربوط به اتصال بین متلب و یا سیمولینک با پایتون هستند.

بلوک بالایی این اتصال را بین داده های استاتیک شامل طول جاده و عرض هر لاین، موقعیت اولیه اتومبیل و جاده، و بسیاری اطلاعات دیگر که بسیاری از آن اطلاعات استفاده نشده اند زیرا در این پروژه مفید نبوده اند. این بلوک، فایل سیمولینک را تغییر نمی‌دهد.

بلوک پایینی نیز با استفاده از روش های شبکه کردن، می‌تواند داده های پویا را از محیط سیمولینک به پایتون منتقل کند. این داده های پویا عبارتند از موقعیت و سرعت و اطلاعات دیگری از اتومبیل در حال حرکت، اطلاعات سنسورها و ... باشد.

- بلوک بعدی پایتون است که خود شامل لایه های دیگری است که در شکل ۳-۱۳ به تفضیل بیان شده است. نکته جالب در آن این است که در آن لایه ها اثری نیز از دو بلوک پیشین آمده است. همچنین بخش اصلی کار، یا به عبارتی مغز و هوش این کار در این قسمت توسعه یافته است.

۳-۲-۱ تقسیم بندی وظایف هر بخش

بخش اصلی کار که وظیفه آن تصمیم گیری و انتخاب مسیر درست توسط یک عامل [8] (که در این پروژه، عامل همان اتومبیل می‌باشد) در پایتون انجام می‌شود. وظیفه اصلی بخش متلب و سیمولینک و پری‌اسکن، ایجاد یک محیط شبیه سازی است.

یادداشت ۳-۲-۱. این محیط شبیه سازی اهمیت زیادی در الگوریتم های یادگیری تقویتی دارد. زیرا در این الگوریتم ها یک «عامل» با «محیط [9] » در تعامل است. تعامل در این الگوریتم ها به معنای این است که «عامل» در یک «حالت [10] » قرار دارد. سپس متناسب با آن یک «حرکت [11] » انجام می‌دهد. با این «حرکت»، «محیط» به آن یک مقدار «امتیاز» و یک «حالت» جدید برمی‌گرداند. بنابراین داشتن یک محیط شبیه سازی کامل و دقیق از اجزای ضروری کار است.

بخش های دیگر مربوط به ارتباط این قسمت ها به یک‌دیگر بودند که پیچیدگی های زیادی را رقم زده است.

به طور ساده‌تر و کلی‌تر می‌توان گفت که پایتون نقش «عامل» و پری‌اسکن نقش «محیط» را دارد.

[8] Agent
[9] Environment
[10] State
[11] Action

شکل ۳-۲: بخشی از توانایی های نرم افزار پری‌اسکن در شبیه سازی

۳-۳ معرفی نرم افزار پری‌اسکن

می‌توان در ابتدا گفت که این نرم‌افزار یک افزونه متلب و سیمولینک است اما توانایی زیادی که آن دارد باعث می‌شود که بگوییم این محصول از متلب و سیمولینک جعت رسیدن به هدف خود کمک می‌گیرد.

نرم افزار پری‌اسکن یکی از نرم افزار های بسیار قدرتمند در زمینه شبیه سازی مسایل مربوط به وسایل نقلیه است که می‌تواند حرکت یک ویله نقلیه را به طور خیلی دقیق و مناسب شبیه سازی کند. همچنین در کنار این وظیفه مهم، یک محیط گرافیکی مناسب را در اختیار کاربران خود قرار می‌دهد که از دیگر حسن های آن است. شکل ۳-۲ این توانایی ها را به تصویر کشیده است.

همچنین این نرم افزار یک سری فایل خروجی به کاربر می‌دهد که یکی از فایل های آن فایل سیمولینک است که اجازه تغییر و دسترسی به داخل برخی بلاک ها به ما کمک می‌کند که اطلاعات خود را از دل آن نرم افزار بیرون بکشیم. [۱۲]

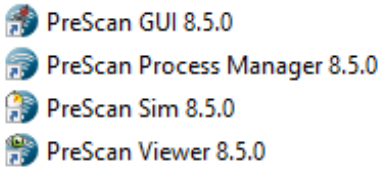
از این رو در مقابسه با محیط های دیگر، محاسن زیادی را داشت که هدف این پروژه را در به کارگیری این ابزار تحت تاثیر قرار داد.

۳-۳-۱ بخش های مختلف نرم افزار پری‌اسکن

پس از دانلود و نصب نسخه 8.5.0 این نرم‌افزار چهار آیکون مانند شکل ۳-۳ به محیط دسکتاپ اضافه می‌کند. اصلی ترین آن ها PreScan Proccess Manager 8.5.0 نام دارد.

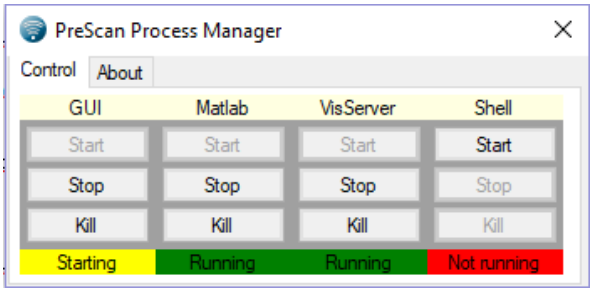
[۱۲] جهت کسب اطلاعات بیشتر و تهیه این نرم افزار به لینک زیر مراجعه کنید:

https://tass.plm.automation.siemens.com/prescan



شکل ۳-۳: آیکون های اضافه شده بر روی محیط دسکتاپ پس از نصب پری‌اسکن

با انتخاب آن صفحه ای مانند زیر باز می‌شود.



شکل ۳-۴: پنل مدیریت نرم‌افزار پری‌اسکن

این پنجره شامل گزینه های زیر است:

- GUI
- Matlab
- VisServer
- Shell

برای ایجاد یک محیط جدید باید GUI را استارت کرد. پس از مدتی صفحه ای مانند شکل ۳-۵ باز می‌شود.

پس از ایجاد مدل ها و ذخیره آن، فایل های `**.pex` و `**.pb` و `**_cs.slx` ساخته می‌شود.[۱۳] جهت استفاده از فایل سیمولینک باید در شکل ۳-۴ متلب را استارت کنید.

نکته ۳-۳-۱. برای اجرای فایل های سیمولینک خروجی، لازم است که متلب را فقط و فقط با استفاده از نرم افزار پری‌اسکن و با استفاده از پنل مدیریت نرم افزار معرفی شده در شکل ۳-۴ باز شود. در صورتی که به صورت مستقیم این کار انجام شود، به مشکل منتهی می‌شود.

دو قسمت دیگر نیز در شکل ۳-۴ وجود دارد که نیازی به استارت کردن آن ها نیست و خودشان در صورت لزوم به صورت خودکار فراخوانی می‌شوند.

[۱۳] علامت ** به معنای یک اسم مشترک در این سه فایل استفاده شده است.

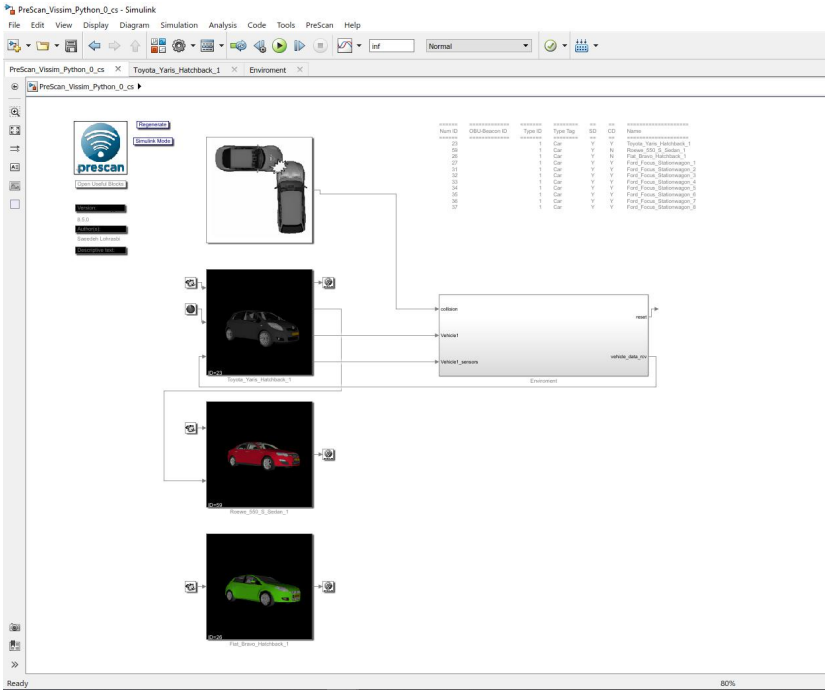
شکل ۳-۵: صفحه گرافیکی محیط پری‌اسکن

۳-۳-۲ فرمت های فایل های خروجی

نرم‌افزار پری‌اسکن پس از ایجاد یک محیط جدید، فایل ها و پوشه های بسیار زیادی را ایجاد می‌کند. اما در خارج آن پوشه ها ۳ فایل وجود دارد که پسوند آن ها `**.pex` و `**.pb` و `**_cs.slx` می‌باشد. علامت ** همان اسم پروژه‌ای است که ایجاد کرده ایم. هر یک از این فایل ها به یک بلوک از شکل ۳-۱ مربوط می‌شود.

فرمت فایل	توضیحات
`**.pex`	این فایل مربوط به اولین بلوک شکل ۳-۱ است و ارتباط مستقیم با GUI دارد. برای تغییر محیط گرافیکی باید این فایل را باز کرد.
`**.pb`	این فایل برخی از اطلاعات فایل `**.pex` را در اختیار دارد و با تغییر آن فایل این فایل نیز عوض می‌شود. این فایل حاوی اطلاعات استاتیک محیط ایجاد شده است و مهم ترین کاربرد آن در بلوک موتور متلب که در شکل ۳-۱ نشان داده شده است می‌باشد. پایتون از طریق این فایل این اطلاعات را دریافت می‌کند.
`**_cs.slx`	این فایل سیمولینک است که برای کار کردن با آن باید از پنل مدیریت شکل ۳-۴ استفاده کرد. این فایل پس از ایجاد از فایل `**.pex` مستقل می‌شود. این فایل خود قابلیت تغییر دارد و می‌توان بلوک‌های آن‌را در محیط سیمولینک تغییر داد و بلوک های دیگری به آن افزود. در صورتی که فایل `**.pex` تغییر کند، این امکان را نیز دارد که از داخل خود سیمولینک با فشردن دکمه ای این تغییرات جدید اعمال شود بدون آن که به تغییرات خود کاربر لطمه ای وارد شود. در این پروژه این فایل، تغییرات بسیاری را تجربه کرد.

جدول ۳-۱: توضیحات فرمت فایل خروجی



شکل ۳-۶: فایل سیمولینک ایجاد شده توسط نرم افزار پری‌اسکن همراه با تغییرات

جدول ۳-۱ توضیحات لازم را جهت آشنایی با این خروجی ها آورده است.

همچنین در بخش ۳-۴ در مورد فایل `**_cs.slx` توضیحات دقیق‌تری در مورد جزییات آن گفته خواهد شد.

۳-۴ بررسی دقیق‌تر فایل سیمولینک

فایل سیمولینک ایجاد شده توسط نرم افزار پری‌اسکن، قابلیت تغییر به دست کاربر را دارد. شکل ۳-۶ فایل تغییریافته مربوط به این پروژه را نشان می‌دهد.

بلوک های سمت راست نشان داده شده در سمت راست شکل ۳-۶ توسط نرم افزار پری اسکن ایجاد شده است که البته دست‌خوش تغییراتی نیز بوده اند.

در صورتی که با استفاده از محیط گرافیکی GUI فایل `**.pex` تغییر کند، فایل سیمولینک تغییر نمی‌کند. در برخی موارد این تغییرات ممکن است منجر به پیغام خطا شود.

نکته ۳-۴-۱. در صورتی که فایل `**.pex` تغییر کند، برای اعمال این تغییرات، باید روی کلمه Regenerate که در شکل ۳-۷ آمده است، کلیک کرد. با این کار، تغییرات جدید اعمال می‌شود بی‌آن‌که تغییرات کاربر تحت تاثیر قرار بگیرد.

شکل ۳-۷: روش صحیح اعمال تغییرات روی فایل سیمولینک

در شکل ۳-۷ همانطور که در نکته ۳-۴-۱ به آن اشاره شد، دکمه ای تحت عنوان Regenerate وجود دارد که استفاده از آن در همان نکته مشخص شده است. همچنین در این تصویر در زیر لوگوی برنامه پری‌اسکن، اطلاعاتی مانند شماره نسخه نرم افزار (که در این‌جا 8.5.0 می‌باشد.)، نام نویسنده مشاهده می‌شود.

در شکل ۳-۶ اولین بلوک سمت راست همان ماشینی است که ما آن‌را تحت کنترل گرفته‌ایم. اگه به آن وارد شویم، شکل ۳-۸ را مشاهده می‌کنیم. در این تصویر ورودی و خروجی ها نقش خیلی مهمی دارند. این اطلاعات در جدول ۳-۲ آمده‌اند.

۳-۴-۱ معرفی بلوک Environment و بررسی جزییات آن

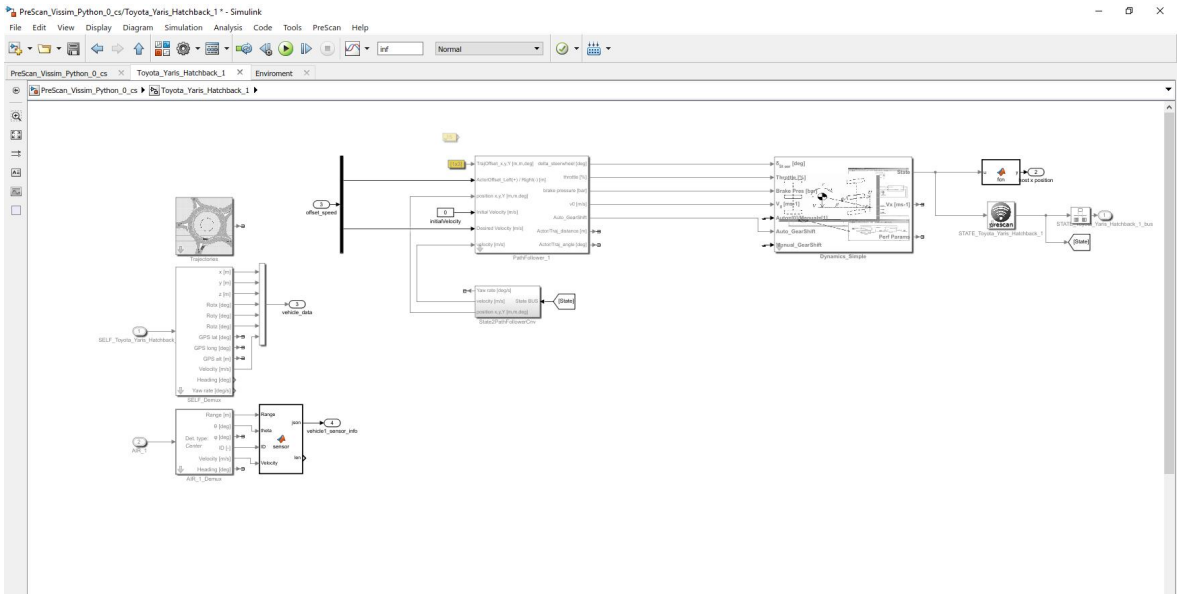
در جدول ۳-۲ صرفا ورودی خروجی های مهم مورد بررسی قرار گرفته است. این ورودی ها و خروجی های مهم به یک بلوک دیگر منتقل می‌شود. بلوک Environment همان بلوک است که در شکل ۳-۶ نیز مشخص است و در شکل ۳-۹ نیز از زاویه نزدیک تر با جزییات بیشتر می‌توان آن‌را مشاهده نمود.

وظیفه اصلی بلوک Environment جمع آوری اطلاعات محیط سیمولینک و همچنین ارسال دستورات کنترلی به آن هاست. اطلاعات جمع آوری شده از محیط سیمولینک شامل موارد زیر است.

- اطلاعات ماشینی که نقش «عامل» را در الگوریتم دارد.

[14] json

[۱۵] بعدا خواهید دید که مشخص بودن طول بسیار بسیار اهمیت دارد!

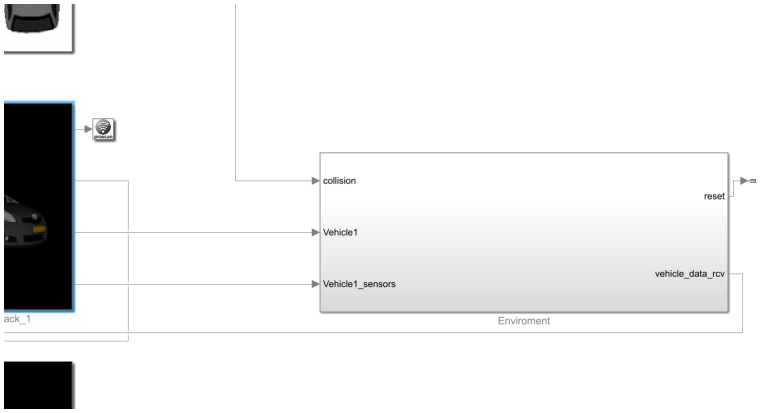


شکل ۳-۸: فایل سیمولینک - شبیه سازی اتومبیل

نوع	عنوان	بلوک مربوط	توضیحات
خروجی	اطلاعات ماشین	SELF_Demux	اطلاعات ماشین، شامل اطلاعات موقعیت(y،x و z) و اطلاعات چرخش (حول y،x و z) به همراه سرعت ماشین را خروجی می‌دهد. این ۷ داده قبل از خروجی توسط یک mux با یکدیگر ادغام می‌شوند.
خروجی	اطلاعات سنسور ماشین	AIR_1_Demux	اطلاعات سنسور V2C را خروجی می‌دهد. از آنجا که این سنسور فاصله و زاویه و ... تا ده ماشین نزدیک خود را می‌دهد. بنابراین هریک از این اطلاعات یک بردار ده تایی است. برای فرستادن آن اطلاعات به خروجی، ابتدا آن‌ها را به طریقی به فرمت جیسون [۱۴] تبدیل می‌کند و یک رشته کاراکتر با طول مشخص [۱۵] را خروجی می‌دهد.
ورودی	کنترل لاین و سرعت ماشین	PathFollower_1	دستورات کنترلی اتومبیل، شامل کدام خط بودن و مقدار سرعت نهایی، به صورت ورودی وارد یک demux می‌شود و پس از جدا سازی، به بلوک مربوط متصل می‌شود.

جدول ۳-۲: بررسی ورودی ها و خروجی های مهم در شکل ۳-۸

- اطلاعات سنسور های ماشین «عامل»
- اطلاعات مربوط به تصادف کردن و کدوم ماشین با کدوم ماشین تصادف کرده است.



شکل ۳-۹: نگاهی از نزدیک به بلوک Environmnet

این اطلاعات به صورت ورودی به این بلوک وارد می‌شود و دستورات کنترلی شامل کنترل لاین ماشین به همراه مقدار سرعت نهایی ماشینی که نقش «عامل» در الگوریتم دارد، از این بلوک خارج می‌شود.

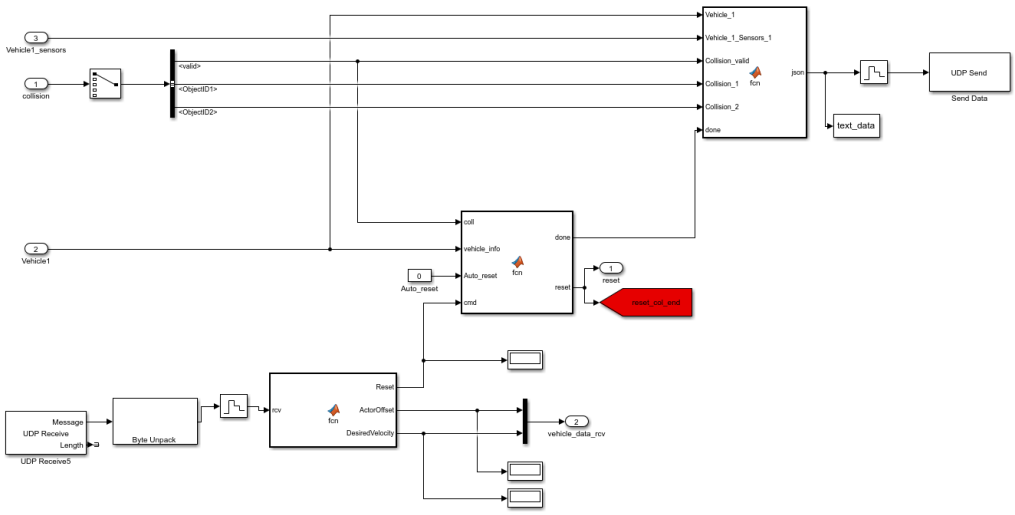
پیشتر صحبت شد که وظیفه تصمیم گیری بر عهده کد پایتون است. با این حساب سیمولینک قدرت تشخیص و تصمیم گیری ندارد. از این رو وظیفه بلوک Environment نیز تصمیم گیری نمی‌باشد بلکه با تکنیک هایی سعی بر برقراری ارتباط با کد پایتون دارد. در حقیقت این بلوک نقش واسط بین سیمولینک و پایتون را دارد. این بلوک علاوه بر دستورات کنترلی لاین و سرعت، دستور «شروع کردن دوباره» و یا ریست را نیز دریافت می‌کند. با این دستور تمامی اطلاعات به حالت اول بر خواهد گشت و ماشین «عامل» به جای اول بر خواهد گشت و منتظر دستورات جدید می‌ماند.

شکل ۳-۱ توضیح بسیار کلی در مورد این نحوه ارتباط نشان داده است و در شکل ۳-۱۰ جزییات بیشتری را در مورد داخل این بلوک را نشان می‌دهد.

اگر به شکل ۳-۱۰ دقت شود دو بلوک UDP دیده می‌شود. (بلوک UDP Send در بالا سمت راست و بلوک UDP Receive در پایین سمت چپ شکل ۳-۱۰) این دو بلوک برای فرستادن داده های مورد نیاز به پایتون و گرفتن دستور های کنترلی از پایتون در سیمولینک تعبیه شده اند.

یادداشت ۳-۴-۲. اگر نمودار شکل ۳-۱ در نظر داشته باشیم، خواهیم یافت که این بلوک در شاخه پایینی ارتباط بین سیمولینک و پایتون قرارا دارد که از روش های شبکه کردن بین این دو انجام می‌شود. این بلوک صرفا داده‌های پویا را از متلب به سیمولینک انتقال می‌دهد و با داده‌های استاتیک کاری ندارد.

در جدول ۳-۳ اطلاعات دقیق شبکه برای این دو بلوک دیده می‌شود. گفتنی است که این داده ها به نحوه مناسب کد شده‌اند تا تنها با استفاده از این دو بلوک بتوان داده ها را منتقل کرد.



شکل ۳-۱۰: فایل سیمولینک - داخل بلوک Environment

نام بلوک	نوع بلوک	IP	Port	توضیحات
UDP Receiver	گیرنده	localhost	۸۰۷۰	
Send Data	فرستنده	localhost	۸۰۳۱	

جدول ۳-۳: اطلاعات بلوک های فرستندگی گیرندگی در سیمولینک

روش کد شدن [16] قبل از ارسال و یا دریافت در این دو بلوک کاملا با یکدیگر متفاوت هستند. این کار توسط بلوک‌های قبل و بعد دو بلوک مذکور انجام می‌شود.

در شکل ۳-۱۰ سه بلوک تابع متلب [17] وجود دارد که از مهم‌ترین نقش را دارند. این نقش‌ها در ادامه توضیح توضیح مفصل داده شده‌اند.

خلاصه این وظایف در زیر آمده است:

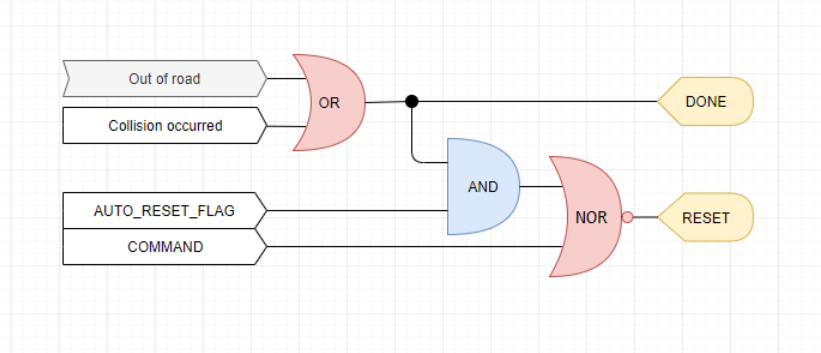
- **بلوک فرستنده** : این بلوک در سمت بالا سمت راست تمام اطلاعاتی که به بلوک کلی Environment وارد می‌شود را به نحوی مناسب به همراه متغیر `done` که از یک تابع متلب دیگر خارج می‌شود را برای پایتون طی ساختار مشخصی (جیسون) ارسال می‌کند. موضوع به صورت بسیط در بخش ۳-۴-۲ مورد بررسی قرار گرفته است.

- **بلوک گیرنده** : این بلوک تنها وظیفه آن این است که داده هایی که پس از بلوک Byte Unpacking آمده است ، را انتخاب می‌کند که هر یک از این دیتا ها چه مفهومی هستند. از آن‌جا که ۳ داده کنترلی (لاین و سرعت و ریست) از طرف پایتون ارسال می‌شود، بلوک Byte Unpacking وظیفه این ۳ داده را به فرمت `double` در می‌آورد (این فرمت در سمت پایتون نیز به همین شکل قرار داد شده‌است). بنابراین این بلوک Byte Unpacking است که کار اصلی را انجام می‌دهد و تنها وظیفه این بلوک تفکیک و اسم گذاری برروی خروجی بلوک Byte Unpacking است.

- **بلوک محاسبه تمام شدن و ریست کردن** : این بلوک با دو خروجی مهم می‌دهد. یکی از آن خروجی ها `done` است که توسط بلوک فرستنده نیز برای پایتون ارسال می‌شود. و مقدار دیگری را که خروجی می‌دهد دستور ریست کردن است. این دستور در لبه مثبت (از صفر به یک برسد) ریست می‌کند. منطق این بلوک یک منطق باینری است که در شکل ۳-۱۱ روش محاسبه آن نیز آمده است. همچنین این بلوک یک مقدار به اسم `Auto_reset_flag` نیز به عنوان ورودی دریافت می‌کند. این مقدار که می‌تواند صفر و یا یک باشد، تصمیم می‌گیرد که در صورتی که `done = 1` شد آیا ریست کند و یا ریست نکند. اگر یک باشد ریست می‌کند وگرنه منتظر رسیدن دستور از پایتون می‌ماند.

[16] Encoding

[17] Matlab-function block



شکل ۳-۱۱: منطق محاسبه تمام شدن و دستور ریست کردن

۳-۴-۲ بررسی ساختار داده های ارسالی و کد آن در سیمولینک

بلوک فرستنده در شکل ۳-۱۰ اطلاعات دریافت شده را طی ساختار مشخصی به ساختار جیسون در می‌آورد. نمونه ای از این ساختار در شکل ۳-۱ آمده است.

(آ) اطلاعات خام جیسون کلی (ب) جیسون سنسور (ج) شکل ادغام شده

شکل ۳-۱۲: تصویر (آ) نحوه ساختاردهی کلی در بلوک تابع متلب واقع در زیرسیستم Environment را نشان می‌دهد. در این قسمت اطلاعات ماشین شامل دو قسمت data و Sensors است. اطلاعات سنسور های این ماشین از تصویر (ب) تامین می‌شود. این ساختار در همان زیرسیستم ماشین تبدیل به جیسون شده و برای زیرسیستم Environmnet ارسال می‌شود. قسمت سنسور یک آرایه است تا بتوان چندین سنسور مختلف را برای آن ارسال کرد. بنابراین ساختاری مانند تصویر (ج) باید را در نهایت برای پایتون ارسال می‌کند.[۱۹]

[۱۹] این عکس با استفاده از سایت http://jsonviewer.stack.hu/ تهیه شده است.

شکل ۳-۱۲(ب) همان ساختاری است که پیشتر در مورد آن صحبت شد. با توجه به این که این ساختار (در ادامه خواهید دید که) در اطلاعات سنسور ماشین به کار گرفته می‌شود و هر سنسور ممکن است اطلاعات خاص خود را داشته باشد. برای یکسان‌سازی این اطلاعات، در اولین لایه دو گزینه data و name را به طور قرار دادی بین همه سنسور ها یکسان است تا در کد پایتون بتوان با توجه به اسم آن ها ادامه ساختار که در data می‌باشد، قابل تشخیص باشد. اما از آن‌جا که صرفا از اطلاعات یک سنسور استفاده شده است این شکل ساختاردهی تنها یک گزینه برای توسعه های آتی آن است.

در این ساختار ۴ اطلاعات نشان داده شده که هر یک از آن اطلاعات خود یک بردار ۱۰ تایی است، برای پایتون ارسال می‌شود.

شکل ۳-۱۲(آ) ساختار کلی است که محیط Environmnet از آن استفاده می‌کند. در این بلوک علاوه بر داده هایی مانند اطلاعات ماشین و سنسورش، اطلاعات تصادف و اطلاعات تمام شدن و یا نشدن شبیه سازی که پیش‌تر در مورد آن اطلاعات صحبت شد، دو اطلاعات اضافه دیگر نیز می‌فرستد.

- `Time` : زمان شبیه سازی را همراه با دیتا دیگر ارسال می‌کند و به اصلاح یک ساختار زمانی ایجاد می‌شود. [۲۰]
- `Object` : این عبارت کمک به کد پایتون می‌کند که ماشین هدف و یا عامل را از بین ماشین های موجود در لیست `Vehicles` بیابد. [۲۱]

در نهایت با ادغام شدن اطلاعات سنسور نیز، ساختار کلی به شکل ۳-۱۲(ج) در خواهد آمد . در بخش ۳-۶ از برخی از چالش های ساختاردهی کردن به این شکل، صحبت خواهد شد. همچنین خروجی نهایی آن نیز را می‌توانید در همان بخش بیابید.

۳-۵ بررسی جزییات بخش پایتون

بارها اشاره شد که پایتون بخش اصل تصمیم‌گیری را برعهده دارد. در بخش ۳-۴ سعی شد تا به نحو مناسبی دیتای محیط شبیه‌سازی را به پایتون منتقل کند و دستورات کنترلی را نیز از پایتون به محیط شبیه‌سازی ارسال کند. در این بخش بر روی مفاهیم پایتونی آن مانور می‌دهیم.

[20] Time-struct

[۲۱] از آنجایی که در این پروژه لیست Vehicles یک آبجکت بیشتر ندارد، پس این بخش نیز صرفا ظرفیت توسعه‌پذیری این کد را بالا می‌برد.

شکل ۳-۱۳: بلوک دیالگرام لایه های پایتون

از آن‌جا که کد پایتون باید به سیمولینک متصل شود و دیتا را به الگوریتم کنترلی خود (که از الگوریتم های یادگیری تقویتی استفاده شده است) ، برساند و از آن جا دستورات را دریافت کند و به سیمولینک برساند، نیازمند لایه هایی است که هر لایه بخشی از این کار ها انجام دهد.

۳-۵-۱ معرفی لایه های کد پایتون

کد پایتون از لایه های مختلف تشکیل شده است. از هر لایه به لایه بعدی سطح زبان بالاتر می‌رود. این لایه ها در شکل ۳-۵-۱ مشخص شده اند. در لایه های ابتدایی، سطح استفاده از دستورات بسیار ابتدایی است و در هر لایه با تعریف توابع و کلاس هایی این امکان را ایجاد کرده اند که بدون در نظر گرفتن این که در سطوح پایین‌تر چه اتفاقاتی می‌افتد، در سطوح بالاتر از آن امکانات استفاده کرد. در ادامه این بحث مفصل توضیح داده خواهد شد.

در شکل ۳-۱۳ ۴ لایه برنامه نویسی شده با ۵ عدد نشان داده شده است. توضیحات زیر متناسب با هریک از این شماره ها در نظر گرفته شده اند:

۱) **Model** : در این لایه، با استفاده از موتور متلب با متلب و سیمولینک ارتباط برقرار می‌کند و با این ارتباط دو وظیفه بسیار مهم را انجام می‌دهد.

آ) ساخت مدل هایی مانند اتومبیل و جاده و دریافت اطلاعات ضروری آن ها در متلب

ب) تعریف کلاس sim و ارسال دستوراتی مانند شروع کردن، توقف کردن، مکث کردن، ادامه دادن و ... به سیمولینک.

این لایه صرفا از اطلاعات استاتیک سیمولینک استفاده می‌کند.

۲ **Environment** : این لایه برای ارتباط با زیرسیستم Environment که پیشتر آن‌را در شکل ۳-۹ مشاهده کردید، ساخته شده است. در این لایه با استفاده از ابزار هایی که لایه Model در اختیار آن قرار می‌دهد، به سادگی آبجکت هایی ماند ماشین و جاده را می‌سازد و با استفاده از اطلاعات شبکه که در جدول ۳-۳ آمده است، با سیمولینک ارتباط برقرار می‌کند و آن اطلاعات پویا را دریافت می‌کند و دستورات کنترلی را برای آن ارسال می‌کند.

این لایه از موتور متلب استفاده نمی‌کند بلکه از روش های شبکه ای استفاده می‌کند.

۳ **Env** : این لایه یک آبجکت از کلاس Environment را دریافت می‌کند و تمامی نیاز های خود را در مورد هر مسیله‌ای که به ارتباط با محیط متلب و یا سیمولینک به آن واگذار می‌کند و تمرکز آن برروی مفاهیم یادگیری تقویتی مانند (۱) فضای مشاهده [۲۲] (۲) فضای حرکت [۲۳] (۳) تعریف حرکت (۴) تعریف حالت (۵) تعریف امتیاز و ... می‌باشد.

۴ **gym** : در نهایت یک مجموعه داریم که هر یک وظایف مربوط به خود را دارند. در این قسمت کد ما که در لایه Env به gym نزدیک شده است، با فراخوانی لایه های پیشین به ترتیب از بالا به پایین (از نظر سطح) فراخوانی می‌شود و در هر لایه آبجکت‌هایی از کلاس‌های موجود در لایه پایین تر فراخوانی می‌شود. از این رو جهت پیکان بر عکس است.

در مورد gym در فصل ؟؟ به صورت مفصل بحث شده‌است.

۵ **Algorithm** : در لایه های پیشین سینتکس کد gym ایجاد شده است و این لایه بر توسعه الگوریتم های یادگیری تقویتی که در فصل ؟؟ مورد بررسی مفصل قرار گرفته است، تمرکز دارد.

می‌توان گفت این لایه نقش رییس و یا مغز کار را دارد و باقی لایه ها کارگرانی هستند که وظیفه دارند که اطلاعات را به شکلی مناسب این لایه، برای آن منتقل کنند.

۳-۶ بررسی دقیق تر برخی چالش های فنی پروژه

[22] Observation space

[23] Action space

```
'{"Time":0,"Object":0,"Vehicles":[{"name":"Toyota_Yaris_Hatchback_1","data":{"Position":{"x":0.0000
0e+00,"y":0.00000e+00,"z":5.70000e-01},"Rotation":{"x":0.00000e+00,"y":0.00000e+00,"z":0.00000e+00}
,"Velocity":0.00000e+00},"Sensors":[{"name":"AIR_1","data":{"Range":[0.00000e+00,0.00000e+00,0.0000
0e+00,0.00000e+00,0.00000e+00,0.00000e+00,0.00000e+00,0.00000e+00,0.00000e+00,0.00000e+00],"theta":
[0.00000e+00,0.00000e+00,0.00000e+00,0.00000e+00,0.00000e+00,0.00000e+00,0.00000e+00,0.00000e+00,0.
00000e+00,0.00000e+00],"ID":[0.00000e+00,0.00000e+00,0.00000e+00,0.00000e+00,0.00000e+00,0.00000e+0
0,0.00000e+00,0.00000e+00,0.00000e+00,0.00000e+00],"Velocity":[0.00000e+00,0.00000e+00,0.00000e+00,
0.00000e+00,0.00000e+00,0.00000e+00,0.00000e+00,0.00000e+00,0.00000e+00,0.00000e+00]}}]}],"
Collision":{"Occurred":0,"col_1": 0,"cal_2": 0},"done":0}'
```

منابع و مراجع

پیوست

موضوعات مرتبط با متن گزارش پایان نامه که در یکی از گروه‌های زیر قرار می‌گیرد، در بخش پیوست‌ها آورده شوند:

۱. اثبات های ریاضی یا عملیات ریاضی طولانی.

۲. داده و اطلاعات نمونه (های) مورد مطالعه (Case Study) چنانچه طولانی باشد.

۳. نتایج کارهای دیگران چنانچه نیاز به تفصیل باشد.

۴. مجموعه تعاریف متغیرها و پارامترها، چنانچه طولانی بوده و در متن به انجام نرسیده باشد.

کد میپل

```
with(DifferentialGeometry):
with(Tensor):
DGsetup([x, y, z], M)
frame name: M
a := evalDG(D_x)
D_x
b := evalDG(-2 y z D_x+2 x D_y/z^3-D_z/z^2)
```

واژه‌نامه‌ی فارسی به انگلیسی

آ

اسکالر . . . . . Scalar

ب

بالابر . . . . . Lift

پ

پایا . . . . . Invariant

ت

تناظر . . . . . Correspondence

ث

ثابت‌ساز . . . . . Stabilizer

ج

جایگشت . . . . . Permutation

چ

چند جمله‌ای . . . . . Polynomial

ح

حاصل‌ضرب دکارتی . . . . Cartesian product

خ

خودریختی . . . . . Automorphism

د

درجه . . . . . Degree

ر

ریزپردازنده . . . . . microprocessor

ز

زیرمدول . . . . . Submodule

س

سرشت . . . . . Character

ص

صادقانه . . . . . Faithful

ض

ضرب داخلی . . . . . Inner product

ط

طوقه . . . . . Loop

ظ

ظرفیت . . . . . Valency

ع

عدم مجاورت . . . . . Nonadjacency

ف

فضای برداری . . . . . Vector space

ک

کاملاً تحویل‌پذیر . . . Complete reducibility

گ

گراف . . . . . Graph

م

ماتریس جایگشتی . . . Permutation matrix

ن

ناهمبند . . . . . Disconnected

و

وارون‌پذیر . . . . . Invertible

ه

همبند . . . . . Connected

ی

یال . . . . . Edge

واژه‌نامه‌ی انگلیسی به فارسی

A

Automorphism . . . . . خودریختی

B

Bijection . . . . . دوسویی

C

Cycle group . . . . . گروه دوری

D

Degree . . . . . درجه

E

Edge . . . . . یال

F

Function . . . . . تابع

G

Group . . . . . گروه

H

Homomorphism . . . . . همریختی

I

Invariant . . . . . پایا

L

Lift . . . . . بالابر

M

Module . . . . . مدول

N

Natural map . . . . . نگاشت طبیعی

O

One to One . . . . . یک به یک

P

Permutation group . . . . . گروه جایگشتی

Q

Quotient graph . . . . . گراف خارج‌قسمتی

R

Reducible . . . . . تحویل پذیر

S

Sequence . . . . . دنباله

T

Trivial character . . . . . سرشت بدیهی

U

Unique . . . . . منحصربفرد

V

Vector space . . . . . فضای برداری

Abstract

This page is accurate translation from Persian abstract into English.

Key Words:

Write a 3 to 5 KeyWords is essential. Example: AUT, M.Sc., Ph. D,..