سپهر ماه‌فر، «کنترل ربات چهارپا با استفاده از یادگیری تقویتی و ایجاد حلقه هدایت» ، پایان‌نامه کارشناسی، استاد راهنما: دکتر علیرضا شریفی، دانشگاه صنعتی شریف، دانشکده مهندسی هوافضا، شهریور ۱۴۰۳.

دانشگاه صنعتی شریف
دانشکده مهندسی هوافضا

پایان‌نامه کارشناسی

عنوان

# کنترل ربات چهارپا با استفاده از یادگیری تقویتی و ایجاد حلقه هدایت

نگارش
سپهر ماه‌فر

استاد راهنما
دکتر علیرضا شریفی

شهریور ۱۴۰۳

---

سپهر ماه‌فر، «کنترل ربات چهارپا با استفاده از یادگیری تقویتی و ایجاد حلقه هدایت» ، پایان‌نامه کارشناسی، استاد راهنما: دکتر علیرضا شریفی، دانشگاه صنعتی شریف، دانشکده مهندسی هوافضا، شهریور ۱۴۰۳.

# کنترل ربات چهارپا با استفاده از یادگیری تقویتی و ایجاد حلقه هدایت

## چکیده

هدف از انجام این پژوهش، کنترل ربات چهارپا با استفاده از یادگیری تقویتی و ایجاد حلقه هدایت به‌منظور دنبال کردن موقعیت مطوب است. به این منظور، ربات Unitree Go2 برای شبیه‌سازی و پیاده‌سازی انتخاب شده‌است. یک کنترل‌کننده مبتنی بر یادگیری تقویتی با استفاده از عامل PPO در نقش کنترل‌کننده اولیه طراحی شده که فرامین کنترلی سرعت را به موقعیت زاویه‌ای مفاصل تبدیل می‌کند. سپس، موقعیت زاویه‌ای مفاصل وارد یک کنترل‌کنندهٔ تناسبی-مشتقی شده و میزان گشتاور مورد نیاز برای رساندن هر مفصل به موقعیت زاویه‌ای محاسبه شده توسط کنترل‌کننده اول، به دست می‌آید. در گام بعد، یک حلقه هدایت با هدف تبدیل فرامین کنترلی موقعیت به فرامین کنترلی سرعت بر اساس دو رویکرد متفاوت ایجاد می‌شود. در رویکرد اول، ربات با حرکت در راستای محور x و y بدنی خود را به موقعیت مطلوب می‌رسلند و در رویکرد دوم، ربات با تنظیم زاویه سمت خود و حرکت در راستای محور x بدنی به موقعیت مطلوب می‌رسد.

## واژه‌های کلیدی:

هدایت ربات چهارپا، کنترل ربات چهارپا، یادگیری تقویتی، Unitree Go2.

## فهرست مطالب

صفحه

## فهرست اشکال

صفحه

صفحه

## فهرست جداول

# ۱ مقدمه

در سال‌های اخیر، ربات‌های چهارپا به دلیل توانایی بالا در پیمایش مسیرهای ناهموار و دامنه حرکتی گسترده، مورد توجه بسیاری قرار گرفته‌اند. شباهت این ربات‌ها به حیوانات، به‌ویژه حیوانات خانگی، باعث شده که در برخی موارد به‌عنوان جایگزینی مناسب برای این موجودات مطرح شوند. افزون بر این، توانایی حرکت در مسیرهای دشوار، استفاده از ربات‌های چهارپا را در مأموریت‌های جست‌وجو و نجات، به‌ویژه در مناطق صعب‌العبور یا ناپیدا از چشم پهپادها و ریزپرنده‌ها، افزایش داده است.

با وجود کارآیی بالا، استفاده از این ربات‌ها تا به امروز به دلیل پیچیدگی دینامیک و دشواری کنترل آن‌ها محدود بوده است. در گذشته، تلاش زیادی برای کنترل و هدایت ربات‌های چهارپا با استفاده از کنترل‌کننده‌های کلاسیک و یا حتی کنترل‌کننده‌های مدرن صورت گرفته‌است. طراحی کنترل‌کننده‌ای که بتنواند ضمن حفظ تعادل ربات، فرامین کنترلی سرعت را به موقعیت زاویه‌ای مفاصل تبدیل کند و هم‌چنین طراحی یک حلقه هدایت که بتواند فرامین کنترلی موقعیت را به فرامین کنترلی سرعت تبدیل کند و این ربات‌ها را به‌صورت موزون و به‌مانند حیوانات از مبدأ به مقصد برساند، چالشی جذاب و ارزشمند به شمار می‌رود.

ربات چهارپا دوازده درجهٔ آزادی (مفصل) دارد. نحوه حرکت این ربات‌ها مبتنی بر اعمال گشتاور بر هر یک از مفاصل و حرکت پاها به‌نتیجهٔ آن است. اولین گام به‌منظور اعمال فرامین موقعیت به ربات، طراحی کنترل‌کننده‌ای است که بتواند فرامین کنترلی سرعت را به موقعیت زاویه‌ای مفاصل تبدیل کند. معرفی الگوریتم گرادیان سیاست قطعی عمیق (DDPG[1]) در سال ۲۰۱۵، استفاده از یادگیری تقویتی (RL[2]) را به منظور کنترل سیستم‌هایی با فضای کنش[3] پیوسته ممکن ساخت. این رویکرد دریچه‌های جدیدی را برای کنترل سیستم‌های پیچیده گشود. از زمان معرفی الگوریتم DDPG تا کنون، الگوریتم‌های جدیدتری با هدف بهبود عملکرد و کاهش حجم محاسبات برای پیاده‌سازی یادگیری تقویتی معرفی

[1] Deep Deterministic Policy Gradient

[2] Reinforcement Learning

[3] Action

شده‌اند. بهینه‌سازی سیاست مجاور (PPO[1]) یکی از جایگزین‌ها برای الگوریتم DDPG می‌باشد که به دلیل تولید سیاست‌های مقاوم، سادگی تنظیم هایپرپارامترها و سادگی پردازش موازی[2] بسیار مورد توجه واقع شده‌است. با در نظر گرفتن موارد ذکر شده، یک کنترل‌کننده مبتنی بر یادگیری تقویتی با استفاده از عامل PPO در نقش کنترل‌کننده اولیه طراحی شده که فرامین کنترلی سرعت را به موقعیت زاویه‌ای مفاصل تبدیل می‌کند. سپس، موقعیت زاویه‌ای مفاصل وارد یک کنترل‌کنندهٔ تناسبی-مشتقی (PD[3]) شده و میزان گشتاور مورد نیاز برای رساندن هر مفصل به موقعیت زاویه‌ای محاسبه شده توسط کنترل‌کننده اول، به دست می‌آید. در گام بعد، یک حلقه هدایت با هدف تبدیل فرامین کنترلی موقعیت به فرامین کنترلی سرعت بر اساس دو رویکرد متفاوت ایجاد می‌شود. در رویکرد اول، ربات با حرکت در راستای محور x و y بدنی خود را به موقعیت مطلوب می‌رساند و در رویکرد دوم، ربات با تنظیم زاویه سمت خود و حرکت در راستای محور x بدنی به موقعیت مطلوب می‌رسد.

این گزارش در شش بخش تنظیم شده‌است. بخش دوم این گزارش به مدل ربات Unitree Go2 می‌پردازد. در این قسمت نحوه مدل‌سازی ربات و دستگاه‌های مختصات مورد استفاده مشخص شده‌اند. در بخش سوم، دو کنترل‌کننده طراحی شده‌است. کنترل‌کننده اول فرامین کنترلی سرعت را به موقعیت زاویه‌ای مطلوب برای مفاصل ربات تبدیل کرده و کنترل‌کننده دوم میزان گشتاور مورد نیاز برای رساندن مفاصل به آن موقعیت زاویه‌ای را محاسبه می‌کند. در بخش چهارم، حلقه هدایت با دو رویکرد متفاوت طراحی شده است و در بخش پنجم نیز عملکرد کنترل‌کننده و حلقه هدایت طراحی‌شده در محیط شبیه‌ساز Isaac Gym بررسی شده‌است. بخش ششم نیز نتایج این پژوهش را مورد تحلیل قرار می‌دهد.

[1] Proximal Policy Optimization

[2] Parallel

[3] Proportional Derivative

# ۲ مدل ربات چهارپای Unitree Go2

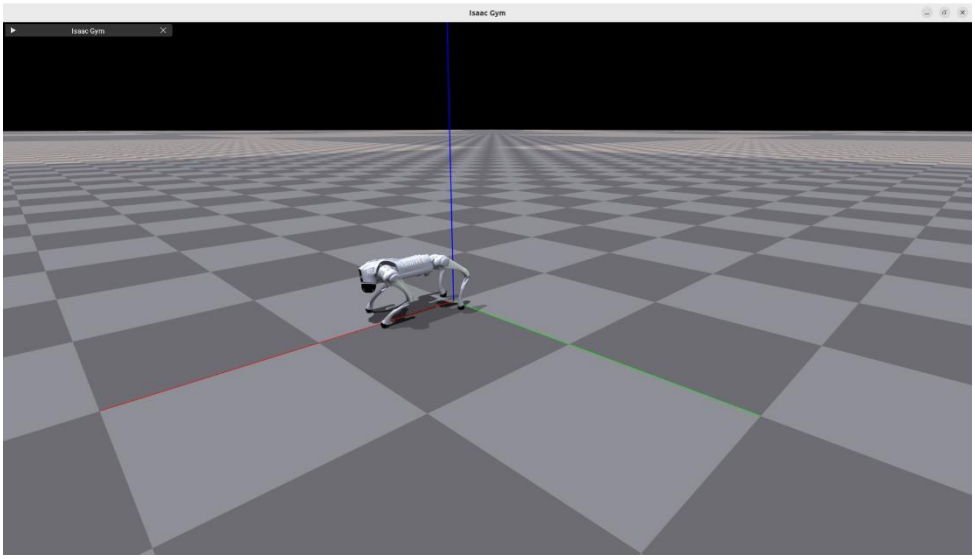
برای شبیه‌سازی مدل ربات چهارپای Unitree Go2 از فایل URDF ارائه شده توسط شرکت سازنده استفاده می‌شود. فایل URDF شامل تمامی مشخصه‌های فیزیکی ربات از جمله تمامی قطعات، اتصالات، مفاصل و محل اتصال آن‌ها، محدودیت‌های فیزیکی (برای مثال بیشینه زاویه چرخش مفاصل)، جرم و ممان اینرسی قطعات و ... است.

## ۱.۲ دستگاه‌های مختصات

برای هر یک از اجزای ربات در محیط شبیه‌ساز Isaac Gym یک دستگاه مختصات بدنی در نظر گرفته می‌شود.

### ۱.۱.۲ دستگاه مختصات اینرسی

دستگاه اینرسی مورداستفاده در شبیه‌سازی در مرکز و دقیقا بر روی سطح زمین قرار گرفته به‌گونه‌ای که محور z آن به سمت بالا باشد.

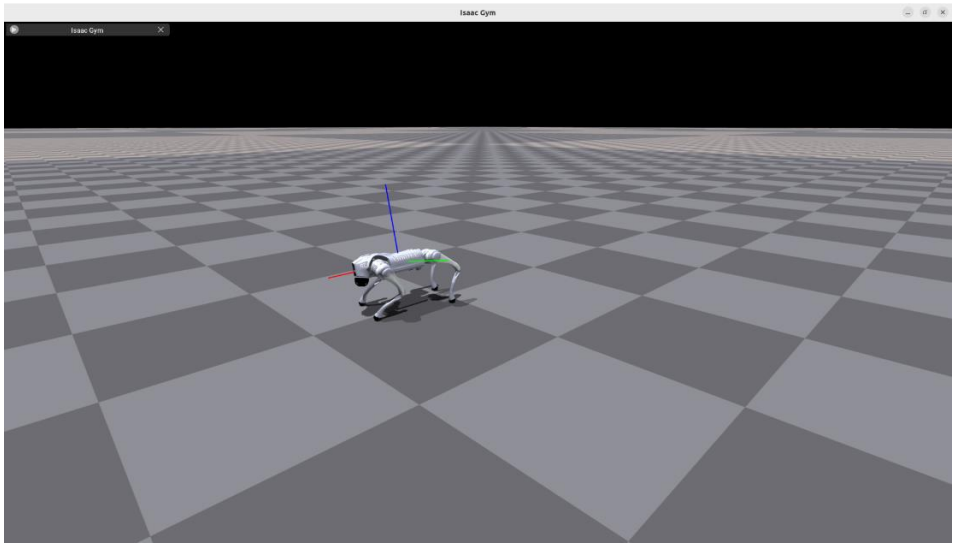



**شکل ۱.۲ دستگاه مختصات اینرسی**

در شکل ۱.۲ خط قرمز محور x، خط سبز محور y و خط آبی محور z دستگاه مختصات اینرسی را نشان می‌دهد.

### ۲.۱.۲ دستگاه مختصات تنه[1] ربات

سرعت و وضعیت ربات باتوجه به سرعت و وضعیت تنه آن نسبت به اینرسی سنجیده می‌شود.




**شکل ۲.۲ دستگاه مختصات تنه ربات**

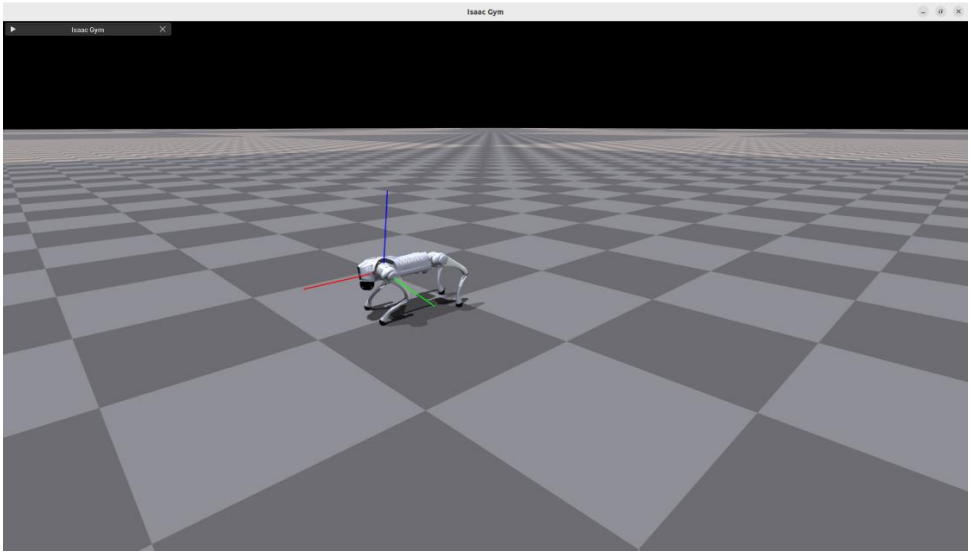
دستگاه مختصات بدنی تنه ربات بر روی مرکز جرم آن قرار دارد. مطابق شکل ۲.۲ محور x به سمت جلوی تنه (خط قرمز)، محور y به سمت چپ تنه (خط سبز) و محور z این دستگاه نیز به سمت بالای تنه (خط آبی) قرار گرفته‌است.

### ۳.۱.۲ دستگاه مختصات لگن[2]

قرار گیری موتور در ناحیه مفصل میان لگن و هر یک از پاها باعث شده پاهای ربات بتوانند به صورت عرضی حرکت کنند.

[1] Torso

[2] Hip

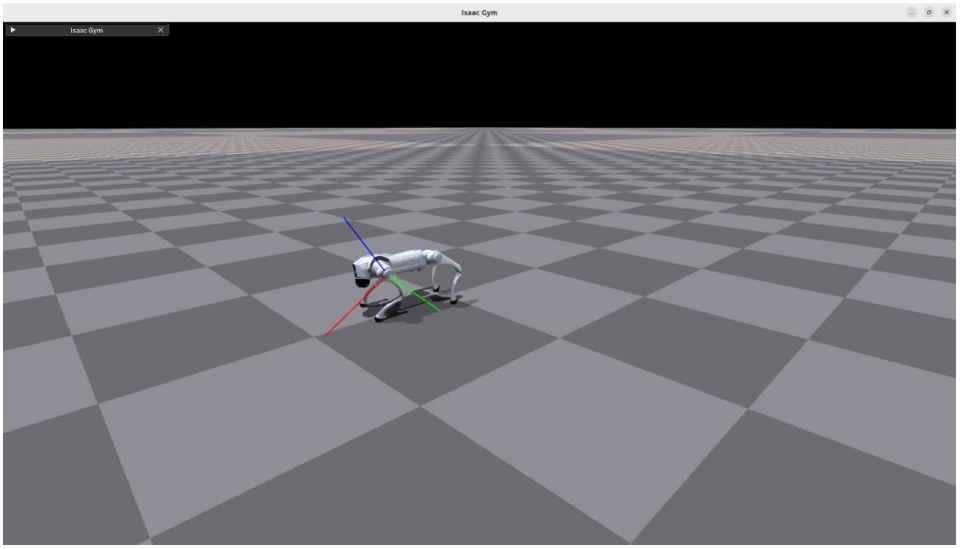



**شکل ۳.۲ دستگاه مختصات مفصل ناحیه لگن**

شکل ۳.۲ دستگاه مختصات بدنی قرارگرفته بر مفصل میان لگن و پای جلو-چپ را نشان می‌دهد. خط قرمز محور x، خط سبز محور y و خط آبی محور z این دستگاه مختصات را نشان می‌دهد.

### ۴.۱.۲ دستگاه مختصات ران[1]

در ران هر یک از پاها نیز یک موتور قرار گرفته که منجر به حرکت طولی پاها از قسمت فوقانی می‌شود.



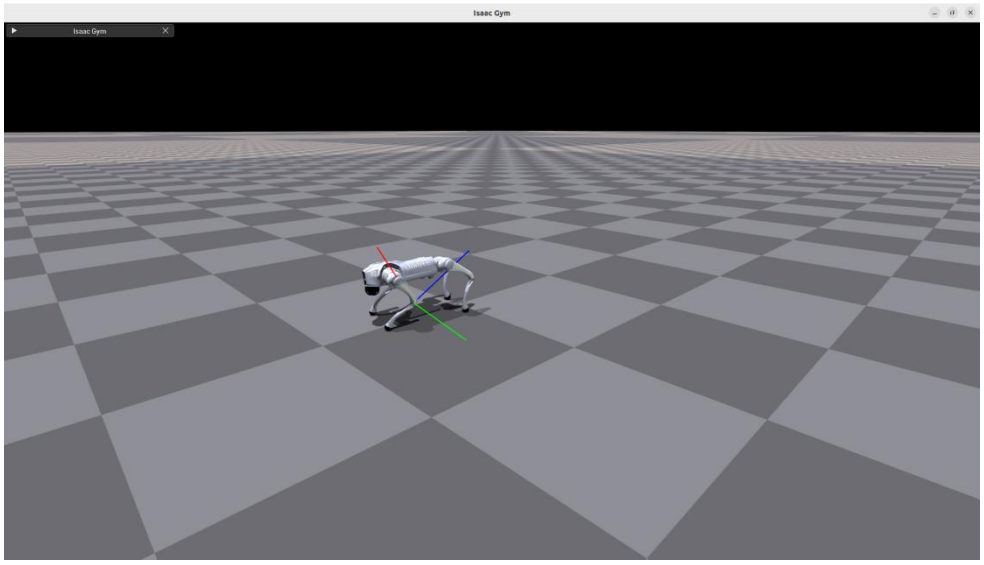

**شکل ۴.۲ دستگاه مختصات ناحیه ران**

[1] Tigh

شکل ۴.۲ دستگاه مختصات بدنی قرارگرفته در ناحیه ران پای جلو-چپ را نشان می‌دهد. خط قرمز محور x، خط سبز محور y و خط آبی محور z این دستگاه مختصات را نشان می‌دهد.

### ۵.۱.۲ دستگاه مختصات ساق پا[1]

قسمت زیرین هر یک از پاهای ربات در ناحیه ساق پا نیز دارای یک موتور است که به واسطه آن، هر یک از ساق‌ها نیز می‌توانند در راستای طولی حرکت کنند.




**شکل ۵.۲ دستگاه مختصات مفصل ناحیه ساق پا**

شکل ۵.۲ دستگاه مختصات بدنی قرارگرفته در ناحیه ساق پای جلو-چپ را نشان می‌دهد. خط قرمز محور x، خط سبز محور y و خط آبی محور z این دستگاه مختصات را نشان می‌دهد.

## ۲.۲ موقعیت زاویه‌ای قراردادی[2]

منظور از موقعیت زاویه‌ای قراردادی، موقعیت زاویه‌ای هر یک از مفاصل در زمانی که هیچ گشتاوری به آن‌ها وارد نمی‌شود است.

[1] Calf

[2] Default Joint Angle

اندازه موقعیت زاویه‌ای قراردادی هر یک از مفاصـــل ربات چهارپای Unitree GO2 در جدول زیر آمده‌است:

**جدول ۱.۲ اندازه موقعیت زاویه‌ای قراردادی مفاصل ربات Unitree Go2**

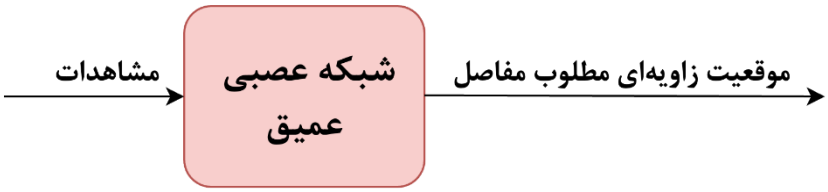
| مفصل | اندازه موقعیت زاویه‌ای قراردادی (رادیان) |
|---|---|
| لگن جلو-چپ | ۰/۱ |
| لگن پشت-چپ | ۰/۱ |
| لگن جلو-راست | -۰/۱ |
| لگن پشت-راست | -۰/۱ |
| ران جلو-چپ | ۰/۸ |
| ران پشت-چپ | ۱ |
| ران جلو-راست | ۰/۸ |
| ران پشت-راست | ۱ |
| ساق جلو-چپ | -۱/۵ |
| ساق پشت-چپ | -۱/۵ |
| ساق جلو-راست | -۱/۵ |
| ساق پشت-راست | -۱/۵ |

# ۳ طراحی کنترل‌کننده

در سیستم کنترلی طراحی شده، از دو کنترل‌کننده استفاده شده‌است. کنترل‌کننده اول موقعیت زاویه‌ای هر یک از مفاصل را در راستای دنبال کردن فرامین کنترلی سرعت محاسبه کرده و کنترل‌کننده دوم نیز، وظیفه دارد میزان گشتاور مورد نیاز برای رسیدن هر یک از مفاصل به موقعیت زاویه‌ای مطلوب محاسبه کند. این ساختار منجر به تشکیل یک حلقه کنترلی بسته می‌شود و به دلیل گرفتن بازخورد از موقعیت زاویه‌ای هر یک از مفاصل، از رسیدن آن‌ها به موقعیت زاویه‌ای مطلوب، مطمئن می‌شود.

## ۱.۳ کنترل‌کننده اول

کنترل‌کننده اول، یک شبکه عصبی عمیق است. این شبکه طی فرآیند یادگیری تقویتی عمیق که یکی از رویکردهای یادگیری عمیق می‌باشد، آموزش می‌بیند.




**شکل ۱.۳ کنترل‌کننده اول**

مطابق شکل ۱.۳ مشاهدات[1] به عنوان ورودی وارد شبکه عصبی عمیق شده و موقعیت زاویه‌ای مطلوب هر یک از مفاصل به منظور دنبال کردن فرامین کنترلی سرعت توسط شبکه عصبی محاسبه می‌شود.

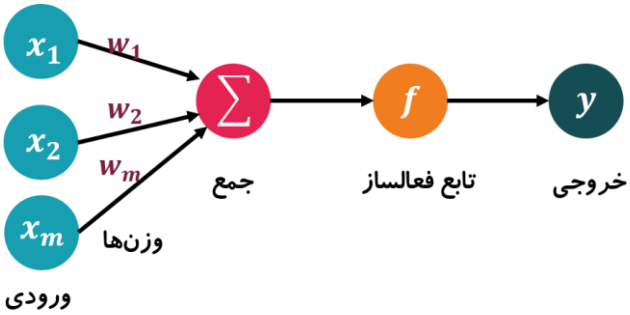
### ۱.۱.۳ مفاهیم اولیه یادگیری عمیق

برخی از اصطلاحات مورد استفاده در یادگیری عمیق عبارت‌اند از:

[1] Observations

**شبکه عصبی:** یک مدل محاسباتی است که برای پردازش داده‌ها و انجام وظایف یادگیری ماشینی به کار می‌رود. این شبکه‌ها از تعداد زیادی واحد محاسباتی به نام نورون تشکیل شده‌اند که به هم متصل هستند و به صورت لایه‌ای سازماندهی می‌شوند. هر اتصال بین نورون‌ها دارای وزنی است که مشخص می‌کند هر سیگنال چه تأثیری بر نورون بعدی دارد. این وزن‌ها طی فرآیند آموزش تنظیم می‌شوند.

**تابع فعال‌ساز**[1]: هر نورون بعد از دریافت سیگنال، توسط یک تابع فعال‌سازی به تولید و یا عدم تولید خروجی می‌گیرد. تابع‌های فعال‌سازی مانند سیگموید[2]، خانواده ReLU[3]، تانژانت هایپربولیک و ... به نورون‌ها کمک می‌کنند که تصمیم‌گیری‌های غیرخطی داشته باشند.




**شکل ۲.۳ ساختار نورون‌ها در شبکه عصبی**

در شکل ۲.۳ ساختار نورون‌ها در یک شبکه عصبی ساده نشان داده شده‌است. در این شکل منظور از $x_i$ نورون‌ها می‌باشد. خروجی این شبکه عصبی به‌صورت زیر محاسبه می‌شود:

$$y = f\left(\sum_{i=1}^{m} x_i w_i\right) \qquad (۱.۳)$$

در عبارت فوق، $f$ همان تابع فعال‌ساز می‌باشد.

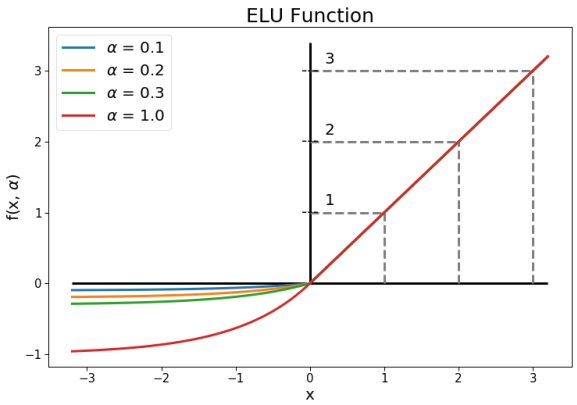
تابع فعال‌ساز استفاده‌شده برای همه لایه‌ها در این پژوهش ELU[4] است. ELU از خانواده توابع فعال‌ساز ReLU است. مزیت توابع فعال‌ساز خانواده ReLu، بار محاسباتی کم‌تر نسبت به سایر فعال‌سازها است.

[1] Activation Function

[2] Sigmoid

[3] Rectified Linear Unit

[4] Exponential Linear Unit




**شکل ۳.۳ رفتار تابع فعال‌ساز ELU**

شکل ۳.۳ رفتار تابع فعال‌ساز ELU در مواجهه با ورودی x را نشان می‌دهد. خروجی تابع فعال‌ساز ELU به مثبت و یا منفی بودن ورودی آن بستگی دارد و به صورت زیر محاسبه می‌شود:

$$f(x)=\begin{cases} x & \text{if } x>0 \\ \alpha\left(e^{x}-1\right) & \text{if } x\le 0 \end{cases} \qquad (۲.۳)$$

در رابطه فوق، $\alpha$ بزرگی مقدار خروجی ورودی‌های منفی را مشخص می‌کند. بر خلاف ReLU که به ازای تمامی ورودی‌های منفی، خروجی صفر تولید می‌کند ELU به ازای ورودی‌های منفی خروجی کوچکی تولید می‌کند. این رویکرد منجر به جلوگیری از وقوع مشکلاتی مانند «نورون مرده[1]»، زمانی که تمام نورون‌ها خروجی صفر می‌دهند و فرآیند یادگیری متوقف می‌شود، «ناپدید شدن گرادیان[2]»، زمانی که گرادیان تابع زیان نسبت به وزن‌ها بسیار کوچک و فرآیند یادگیری به‌شدت کند می‌شود، و هم‌چنین به دلیل نزدیک شدن میانگین خروجی نورون‌ها به صفر، همگرایی در زمان بهینه‌سازی سریع‌تر رخ می‌دهد.

**شبکه عصبی عمیق:** نوعی شبکه عصبی است که از چندین لایه‌ی پنهان بین لایه ورودی و لایه خروجی تشکیل شده‌است. این لایه‌های پنهان مدل‌سازی روابط پیچیده‌تر بین داده‌ها را برای شبکه ممکن می‌کند.

**زیان[3]:** زیان در یک شبکه عصبی، هزینه متحمل شده ناشی از نادرستی پیش‌بینی را اندازه می‌گیرد.

[1] Dead Neuron

[2] Vanishing Gradient

[3] Loss

$$\mathcal{L}\Big( f(x^{(i)}; \mathbf{W}), y^{(i)} \Big) \qquad (۳.۳)$$

**تابع هدف**[1]: تابع هدف تمام هزینه متحمل شده ناشی از نادرستی پیش‌بینی را بر روی کل داده‌ها اندازه می‌گیرد.

$$J(\mathbf{W}) = \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}\Big( f(x^{(i)}; \mathbf{W}), y^{(i)} \Big) \qquad (۴.۳)$$

**بهینه‌سازی زیان**[2]: زیان تابعی از وزن‌های شبکه است. هدف از یک فرآیند یادگیری عمیق، محاسبه وزن‌هایی است که منجر به کم‌ترین زیان شود.

$$\mathbf{W}^* = \underset{\mathbf{w}}{\operatorname{argmin}}\, J(\mathbf{W}) \qquad (۵.۳)$$

در عبارت بالا، منظور از $\mathbf{W}^*$، وزن‌های بهینه برای رسیدن به کم‌ترین زیان است.

**گرادیان نزولی**[3]: برای پیدا کردن $\mathbf{W}^*$ از روش گرادیان نزولی استفاده می‌شود. در این روش ابتدا وزن‌های شبکه عصبی به صورت تصادفی (نرمال) مقداردهی می‌شوند. در گام بعد، گرادیان تابع هدف نسبت به هر یک از وزن‌ها محاسبه شده و با حرکت در خلاف جهت گرادیان به مقداری مشخص که نرخ یادگیری[4] نام دارد، وزن‌های جدید محاسبه می‌شوند. این روند تا زمانی که وزن‌ها به مقدار مشخصی همگرا شوند به صورت حلقه‌وار ادامه می‌یابد. نرخ یادگیری کوچک منجر به کاهش سرعت یادگیری و گیرکردن الگوریتم در کمینه‌های محلی[5] می‌شود. هم‌چنین در صورت بزرگ بودن نرخ یادگیری، فراجهش و ناپایداری در فرآیند آموزش رخ می‌دهد. امروزه نرخ یادگیری به صورت تطبیقی و با توجه به بزرگی گرادیان، سرعت فرآیند یادگیری و ... تنظیم می‌شود و دیگر به صورت ثابت استفاده نمی‌شود.

$$\mathbf{W} \leftarrow \mathbf{W} - \eta \frac{\partial J(\mathbf{W})}{\partial \mathbf{W}} \qquad (۶.۳)$$

[1] Objective Function

[2] Loss Optimization

[3] Gradient Descent

[4] Learning Rate

[5] Local Minima

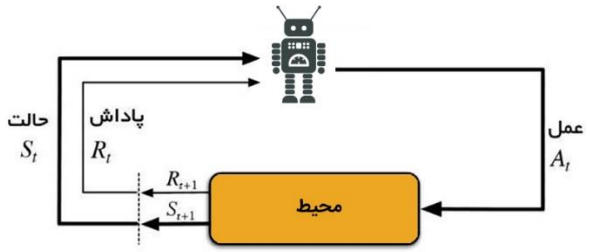
**گرادیان نزولی تصادفی**[1]: محاسبه گرادیان تابع هدف نسبت به وزن‌ها از لحاظ محاسباتی به‌شدت سنگین است، برای همین به جای استفاده از تمام داده‌ها، یک نمونه تصادفی از داده‌ها انتخاب شده و گرادیان تابع هدف نسبت به وزن‌ها با استفاده از آن تخمین زده می‌شود.

**Adam**: یک بهینه‌ساز مبتنی بر ویژگی‌های گرادیان نزولی است که به منظور جلوگیری از گیرکردن الگوریتم در کمینه‌های محلی، به صورت خودکار نرخ یادگیری را بر اساس میانگین واریانس گرادیان‌ها تنظیم می‌کند. در این پژوهش از بهینه‌ساز Adam استفاده می‌شود.

**گرادیان صعودی**[2]: دقیقا مانند گرادیان نزولی است منتها با حرکت در جهت گرادیان سعی در بیشینه کردن تابع هدف دارد.

### ۲.۱.۳ مفاهیم یادگیری تقویتی

بخش‌های اصلی یادگیری تقویتی محیط و عامل است. عامل در محیط قرار دارد و با آن تعامل دارد. در هر مرحله از تعامل بین عامل و محیط، عامل یک مشاهده جزئی از وضعیت محیط انجام میدهد و سپس در مورد اقدامی که باید انجام دهد تصمیم می‌گیرد. وقتی عامل بر روی محیط عمل می‌کند، محیط تغییر می‌کند، اما ممکن است محیط به تنهایی نیز تغییر کند. عامل یک سیگنال پاداش نیز از محیط دریافت می‌کند، عددی که به آن می‌گوید وضعیت فعلی محیط چقدر خوب یا بد است. هدف عامل به حداکثر رساندن پاداش انباشته خود است که بازگشت نام دارد.




**شکل ۴.۳ تعامل بین محیط و عامل در یادگیری تقویتی**

درشکل ۴.۳ تعامل بین محیط و عامل در یادگیری تقویتی نشان داده شده‌است. یادگیری تقویتی شامل روش‌هایی است که عامل رفتارهای مناسب برای رسیدن به هدف خود را می‌آموزد.

[1] Stochastic Gradient Descent

[2] Gradient Ascent

برخی از اصطلاحات مورد استفاده در یادگیری تقویتی عبارتند از:

**حالت[1] و مشاهدات:** حالت (s) توصیف کاملی از وضعیت محیط است. همه اطلاعات محیط در حالت وجود دارد. مشاهدات یک توصیف جزئی از حالت است. در این پژوهش، مشاهدات شامل سرعت‌های خطی و زاویه‌ای ربات، موقعیت و سرعت زاویه‌ای مفاصل ربات، زاویه رول و پیچ ربات، فرامین کنترلی و اقدامات صورت گرفته توسط عامل در گام زمانی قبلی می‌باشد.

**فضای عمل[2]:** فضای عمل در یادگیری تقویتی، مجموعه‌ای از تمام اقداماتی است که یک عامل می‌تواند در محیط خود انجام دهد. این فضا می‌تواند پیوسته و یا گسسته باشد. فضای عمل در این پژوهش پیوسته و به صورت یک توزیع احتمال گاوسی برای موقعیت زاویه‌ای هر یک از مفاصل است.

**سیاست[3]:** سیاست قاعده‌ای است که عامل برای تصمیم‌گیری در مورد اقدامات خود از آن استفاده می‌کند. در این پژوهش، سیاست در قالب یک شبکه عصبی آموزش می‌بینند.

**تابع پاداش[4] و بازگشت[5]:** تابع پاداش حالت فعلی محیط به آخرین عمل انجام شده و حالت بعدی محیط بستگی دارد. تابع پاداش را می توان به صورت زیر تعریف کرد:

$$r_t = R\left(s_t, a_t, s_{t+1}\right) \tag{۷.۳}$$

هدف عامل رسیدن به بیش‌ترین پاداش در طول زمان است. به مجموع پاداش‌ها در طول زمان تابع بازگشت گفته می‌شود و با $R(\tau)$ نشان داده می‌شود. تابع بازگشت به صورت زیر محاسبه می‌شود:

$$R(\tau) = \sum_{t=0}^{T} \gamma^t r_t \tag{۸.۳}$$

به $\gamma$ در معادله بالا، فاکتور تنزیل[6] می‌گویند. فاکتور تنزیل عددی بین صفر تا یک است و باعث کاهش ارزش پاداش‌ها در زمان‌های دورتر می‌شود.

[1] State

[2] Action

[3] Policy

[4] Reward Function

[5] Return Function

[6] Discount Factor

**ارزش**[1]**:** منظور از ارزش در یادگیری تقویتی، بازگشت مورد انتظار است. یعنی اگر مسیر از یک حالت و یا جفت حالت-عمل شروع شود و سپس برای همیشه طبق یک سیاست خاص عمل شود، به‌طور میانگین چه مقدار پاداش دریافت خواهدشد. در اینجا به چهار نوع تابع ارزش اشاره می‌شود:

۱. تابع ارزش تحت سیاست[2] $(V^{\pi}(s))$: این تابع، بازگشت مورد انتظار را در صورتی که مسیر از حالت $s$ شروع شده و همیشه طبق سیاست $\pi$ عمل شود، خروجی می‌دهد.

$$V^{\pi}(s) = \underset{\tau\sim\pi}{\mathbb{E}}\left[R(\tau)\,|\,s_0 = s\right] \qquad (۹.۳)$$

۲. تابع ارزش-عمل تحت سیاست[3] $(Q^{\pi}(s,\,a))$: این تابع، بازگشت مورد انتظار را در صورتی که مسیر از حالت $s$ شروع شده، یک اقدام دلخواه $a$ (که ممکن است از سیاست $\pi$ نباشد) انجام شود و سپس برای همیشه طبق سیاست $\pi$ عمل شود، خروجی می‌دهد.

$$Q^{\pi}(s,\,a) = \underset{\tau\sim\pi}{\mathbb{E}}\left[R(\tau)\,|\,s_0 = s, a_0 = a\right] \qquad (۱۰.۳)$$

۳. تابع ارزش بهینه[4] $(V^{*}(s))$: این تابع، بازگشت مورد انتظار را در صورتی که مسیر از حالت $s$ شروع شده و همیشه طبق سیاست بهینه عمل شود، خروجی می‌دهد.

$$V^{*}(s) = \max_{\pi}\left(V^{\pi}(s)\right) \qquad (۱۱.۳)$$

۴. تابع ارزش-عمل بهینه[5] $(Q^{\pi}(s,\,a))$: این تابع، بازگشت مورد انتظار را در صورتی که مسیر از حالت $s$ شروع شده، یک اقدام دلخواه $a$ انجام شود و سپس برای همیشه طبق سیاست بهینه عمل شود، خروجی می‌دهد.

$$Q^{*}(s,a) = \max_{\pi}\left(Q^{\pi}(s,a)\right) \qquad (۱۲.۳)$$

[1] Value

[2] On-Policy Value Function

[3] On-Policy Action-Value Function

[4] Optimal Value Function

[5] Optimal Action-Value Function

**تابع مزیت**[1] $A(s,a)$: این تابع، مزیت انجام اقدام $a$ در وضعیت $s$ را در قیاس با پاداش مورد انتظار میانگین برای آن وضعیت تحت سیاست فعلی را محاسبه می‌کند. این تابع به ارزیابی میزان بهتر یا بدتر بودن یک عمل نسبت به عملکرد متوسط آن در یک حالت خاص کمک می‌کند.

$$A(s,a) = Q(s,a) - V(s) \qquad (۱۳.۳)$$

**دوره**[2]: به‌منظور استفاده کامل از تجربه‌های به دست‌آمده در یک مسیر، قبل از تولید تجربه‌های جدید، چند دوره از یک بسته تجربه مشخص استفاده می‌شود.

**ساختار عملگر-منتقد**[3]: این ساختار ترکیبی از روش‌های مبتنی بر سیاست و ارزش است و هدف آن یادگیری سیاستی بهینه برای انجام عمل‌ها در یک محیط با استفاده از پاداش‌های دریافت شده از محیط است. در الگوریتم‌های مبتنی بر این ساختار، دو شبکه عصبی آموزش می‌بینند. شبکه عصبی اول منتقد نام دارد. این شبکه عصبی وظیفه ارزیابی عملکرد عملگر را دارد. منتقد با محاسبه تابع ارزش یا تابع مزیت، به عملگر بازخورد می‌دهد تا سیاست را بهبود دهد. شبکه عصبی دوم عملگر است. این شبکه عصبی مسئول تصمیم‌گیری است، یعنی سیاست را یاد می‌گیرد و اعمال را در هر حالت انتخاب می‌کند.

### ۳.۱.۳ الگوریتم PPO

در این پژوهش برای پیاده‌سازی فرآیند یادگیری تقویتی، از الگوریتم PPO مبتنی بر ساختار منتقد-عملگر استفاده می‌شود. عملگر به سیاست $\pi_\theta(a|s)$ که $\theta$ در آن نشان‌دهنده پارامترهای سیاست است اشاره دارد. منظور از منتقد نیز تابع ارزش $V_\phi(s)$ است که $\phi$ در آن نشان‌دهنده پارامترهای تابع ارزش است. سیاست $\pi_\theta(a|s)$ یک توزیع احتمال بر روی عمل‌ها در یک حالت مشخص $s$ است. هدف یادگیری پارامترهای $\theta$ که منجر به بیشینه شدن بازگشت مورد انتظار می‌شوند، است. الگوریتم PPO شامل دو گونه PPO-Clip و PPO-Penalty است. در این پژوهش از الگوریتم PPO-Clip استفاده می‌شود. هدف اصلیPPO-Clip جلوگیری از تغییرات بیش از حد در سیاست عامل است.

**تابع هدف جانشین**[4]: تابع هدف الگوریتم PPO-Clip به صورت زیر است:

[1] Advantage Function

[2] Epoch

[3] Actor-Critic

[4] Surrogate Objective Function

$$L^{PPO-Clip}(\theta)=\mathbb{E}_t\left[\min\left(r_t(\theta)A_t,\text{clip}\left(r_t(\theta),1-\varepsilon,1+\varepsilon\right)A_t\right)\right] \quad (۱۴.۳)$$

$r_t(\theta)$ در عبارت فوق نسبت احتمال بین سیاست جدید و سیاست قدیم است و به صورت زیر محاسبه می‌شود:

$$r_t(\theta)=\frac{\pi_\theta(a\mid s)}{\pi_{\theta_{\text{old}}}(a\mid s)} \quad (۱۵.۳)$$

$\varepsilon$ نیز یک فراپارامتر است که مقدار آن معمولا کوچک است. این فراپارامتر مشخص می‌کند که سیاست جدید تا چه میزان اجازه دارد از سیاست قبلی دور شود.

**تخمین مزیت:** PPO عموما از تخمینگر مزیت تعمیم‌یافته(GAE[1]) برای محاسبه تابع مزیت $A(s,a)$ استفاده می‌کند. در این رویکرد مزیت به‌عنوان تفاوت بازگشت اعمال و تخمین تابع ارزش تعریف می‌شود.

$$A_t=\delta_t+(\gamma\lambda)\delta_{(t+1)}+(\gamma\lambda)^2\delta_{(t+2)}+\cdots \quad (۱۶.۳)$$

در عبارت فوق، $\gamma$ و $\lambda$ به ترتیب فاکتور تنزیل و پارامتر هموارسازی[2] هستند. پارامتر هموارسازی تخمین مزیت را در طول زمان را با کمک تعدیل بازگشت‌های زمان کوتاه و زمان بلند هموار می‌کند.با تنظیم پارامتر هموارسازی می‌توان میزان اثرگذاری پاداش‌های زودهنگام را در برابر پاداش‌های زمان‌های دورتر در زمان محاسبه تابع مزیت کنترل کرد. $\delta_t$ نیز خطای اختلاف موقت[3] است که به صورت زیر محاسبه می‌شود:

$$\delta_t=r_t(\theta)+\gamma V_\phi(s_{t+1})-V_\phi(s_t) \quad (۱۷.۳)$$

**فرآیند بهینه‌سازی:** فرآیند بهینه‌سازی شامل تکرار بر روی دسته‌ای از داده‌ها، اعمال گرادیان صعودی و بیشینه کردن تابع هدف است. جمع‌آوری داده‌ها توسط تعاملات ربات با محیط و ذخیره‌سازی تحولات در یک حافظه موقت[4] صورت انجام می‌شود. هر گام بهینه‌سازی بین نمونه‌برداری بسته‌های کوچک از حافظه موقت و بروزرسانی سیاست و تابع ارزش به‌صورت متناوب انجام می‌شود. به منظور استفاده کامل از اطلاعات

[1] Generalized Advantage Estimator

[2] Smoothing Parameter

[3] Temporal Difference Error

[4] Buffer

هر بسته، چند دوره[1]بروزرسانی بر روی یک دسته از داده مشخص اعمال می‌شود. شیوه بروزرسانی پارامترهای سیاست در هر گام بهینه‌سازی به‌صورت زیر است:

$$\theta_{k+1} = \arg\max_{\theta} \mathop{\mathbb{E}}_{s,a\sim\pi_{\theta_k}} \left[ L\left(s,a,\theta_k,\theta\right) \right] \qquad (۱۸.۳)$$

**زیان تابع ارزش**[2]: PPO علاوه‌بر بروزرسانی سیاست، تابع ارزش را نیز را به منظور افزایش دقت در تخمین بازگشت مورد انتظار بروزرسانی می‌کند. زیان تابع ارزش به‌عنوان میانگین خطای مربعی[3]بین بازگشت‌های واقعی و بازگشت‌های پیش‌بینی شده محاسبه می‌شود:

$$L^{VF}\left(\phi\right) = \mathbb{E}_t \left[ \left( R_t - V_\phi(s_t) \right)^2 \right] \qquad (۱۹.۳)$$

**تنظیم آشفتگی**[4]: برای تشویق الگوریتم به اکتشاف و جلوگیری از همگرایی ناپخته به سیاست‌های زیربهینه[5] از امتیاز آشفتگی استفاده می‌شود:

$$H\left(\pi_\theta\left(a_t \mid s_t\right)\right) = -\sum_a \pi_\theta\left(a \mid s\right) \log \pi_\theta\left(a \mid s\right) \qquad (۲۰.۳)$$

علامت منفی مطمئن می‌شود که عدم قطعیت بیش‌تر منجر به ارزش مثبت برای آشفتگی شود. در نظر گرفتن احتمالات مساوی برای اقدام‌های مختلف توسط سیاست $\pi_\theta\left(a \mid s\right)$ به معنای اکتشافی بودن سیاست است که منجر به بالا بودن آشفتگی می‌شود. بالا بودن آشفتگی عملگر را به اکتشاف اقدام‌های متفاوت می‌کند. این موضوع در مراحل اولیه یادگیری که عملگر نیاز به جمع‌آوری تجربه‌های متنوع از محیط دارد، بسیار مهم است. درنهایت، تابع هدف کل به صورت زیر محاسبه می‌شود:

$$L^{total}\left(\theta\right) = L^{CLIP}\left(\theta\right) - \mathrm{c}_1 L^{VF}\left(\phi\right) + \mathrm{c}_2 \mathbb{E}_t \left[ H\left(\pi_\theta\left(a_t \mid s_t\right)\right) \right] \qquad (۲۱.۳)$$

در عبارت فوق $\mathrm{c}_1$ و $\mathrm{c}_2$ به‌ترتیب فاکتور وزنی برای ارزش زیان و امتیاز آشفتگی هستند. این رویکرد منجر به عملکرد تصادفی سیاست در آموزش‌های اولیه و همگرایی تدریجی به یک رفتار قطعی می‌شود.

[1] Epoch

[2] Value Function Loss

[3] Mean Squared Error

[4] Entropy Regularization

[5] Suboptimal Policies

### ۴.۱.۳ محیط شبیه‌سازی

برای پیاده‌سازی فرآیند یادگیری تقویتی از محیط شبیه‌ساز یادگیری تقویتی Isaac Gym استفاده می‌شود. مزیت این محیط شبیه‌ساز تولید تعداد بسیار زیاد نمونه[1] از ربات می‌باشد. این موضوع پردازش موازی محاسبات با کارت گرافیک را ممکن می‌کند و سرعت فرآیند جمع‌آوری داده را به طرز چشم‌گیری افزایش می‌دهد. هم‌چنین به منظور ساده‌سازی فرآیند آماده‌سازی محیط برای پیاده‌سازی یادگیری، از محیط legged_gym که برای پیاده‌سازی فرآیند یادگیری ربات‌های چهار پا در Isaac Gym ساخته‌شده استفاده می‌شود. این محیط با استفاده از کتابخانه rsl_rl، آماده‌سازی محیط را بساده‌تر می‌کند.

### ۵.۱.۳ تابع پاداش

برای تولید سیاستی که بتواند فرامین کنترلی سرعت را به موقعیت زاویه‌ای مطلوب هر یک از مفاصل تبدیل کند، تعداد زیادی پاداش و مجازات مقیاس‌شده متناسب با ارزش، با یکدیگر جمع شده و تابع پاداش کامل را تشکیل می‌دهند. این پاداش‌ها و مجازات‌ها عبارت‌اند از:

**دنبال کردن سرعت خطی:** عامل در ازای وجود خطا در دنبال کردن فرامین کنترلی سرعت خطی در راستای محور x و y بدنی پاداش مجازات می‌شود:

$$\text{Punishment } v_{xy} = e^{\frac{-Error\, v_{xy}}{0.25}} \qquad (۲۲.۳)$$

**دنبال کردن سرعت زاویه‌ای:** عامل در ازای وجود خطا در دنبال کردن سرعت زاویه‌ای در راستای محور z بدنی مجازات می‌شود:

$$\text{Punishment } \omega_z = e^{\frac{-Error\, \omega_z}{0.25}} \qquad (۲۳.۳)$$

**سرعت خطی در راستای محور z بدنی:** در صورت ایجاد هرگونه سرعت در راستای محور z بدنی عامل مجازات می‌شود. این مجازات باعث تشویق عامل به حفظ وضعیت ربات می‌شود.

**سرعت زاویه‌ای در راستای محور x و y:** در صورت ایجاد هرگونه سرعت زاویه ای در راستای محور x و یا y بدنی عامل مجازات می‌شود. این مجازات باعث تشویق عامل به حفظ وضعیت ربات می‌شود.

[1] Instance

**گشتاور مفاصل:** در صورت تولید گشتاور توسط موتور قرار گرفته بر روی هر یک از مفاصل، عامل مجازات می‌شود. این مجازات منجر به تشویق عامل به پیدا کردن سیاست بهینه از منظر مصرف انرژی می‌شود.

**شتاب زاویه‌ای مفاصل:** در صورت شتاب گرفتن هر یک از مفاصل، عامل مجازات می‌شود. این مجازات عامل را به پیدا کردن سیاستی که منجر به حرکت هموارتر و بهینه‌تر ربات شود تشویق می‌کند. هم‌چنین به‌دلیل کاهش تنش مکانیکی بر روی مفاصل، باعث افزایش عمر سخت‌افزار مکانیکی ربات می‌شود.

**قدم‌های بلند:** عامل درصورت استفاده از قدم‌های بلند تشویق می‌شود. به این‌منظور از فاصله زمانی برداشتن هر یک از پاها از روی سطح زمین و برخورد مجدد پا با سطح زمین به عنوان معیاری برای اندازه‌گیری طول قدم برداشته‌شده استفاده می‌شود.

**برخورد:** عامل درصورت برخورد ساق و یا ران هر یک از پاهای ربات با زمین مجازات می‌شود.

**نرخ عمل‌ها:** در صورت تغییر هر یک عمل‌ها، عامل مجازات می‌شود. این مجازات عامل را به پیدا کردن سیاستی که منجر به حرکت هموارتر و بهینه‌تر ربات شود تشویق می‌کند.

**نزدیک شدن به آستانه موقعیت زاویه‌ای مفاصل:** درصورت عبور هر یک از مفاصل از ۹۰ درصد آستانه موقعیت زاویه‌ای آن مفصل، عامل مجازات می‌شود.

هر یک از این پاداش‌ها و یا مجازات‌ها متناسب با ارزش و بازه بزرگی آن‌ها باید مقیاس شوند. به این منظور هر یک از آن‌ها قبل از جمع شدن با یکدیگر در یک ضریب ضرب می‌شوند.

**جدول ۱.۳ مقیاس پاداش‌ها و مجازات‌ها**

| پارامتر | مقدار |
|---|---|
| دنبال کردن سرعت خطی | ۱ |
| دنبال کردن سرعت زاویه‌ای | ۰/۵ |
| سرعت خطی در راستای محور z بدنی | -۲ |
| سرعت زاویه‌ای در راستای محور x و y | -۰/۰۵ |
| گشتاور مفاصل | -۰/۰۰۰۲ |
| شتاب زاویه‌ای مفاصل | -۰/۰۰۰۰۰۰۲۵ |
| قدم‌های بلند | ۱ |
| برخورد | -۱ |
| نرخ عمل‌ها | -۰/۰۱ |

| | |
|---|---|
| نزدیک شدن به آستانه موقعیت زاویه‌ای مفاصل | -۱۰ |

### ۶.۱.۳ پارامترهای یادگیری تقویتی

**شبکه‌های عصبی:** در این پژوهش هر دو شبکه عصبی منتقد و عملگر سه لایه پنهان دارند. در هر دوی آن‌ها لایه اول ۵۱۲، لایه دوم ۲۵۶ و لایه سوم نیز ۱۲۸ گره دارد.

**تعداد گام‌های زمانی:** به‌دلیل تعداد بسیار زیاد نمونه برای جمع‌آوری داده در محیط Isaac Gym ، حداکثر تعداد گام زمانی کوچک انتخاب می‌شود. برای این پژوهش در هر قسمت به‌ازای هر نمونه ربات حداکثر ۲۴ گام زمانی طی می‌شود.

**نرخ یادگیری:** بهینه‌ساز در این پژوهش از نرخ یادگیری تطبیقی مبتنی بر واگرایی KL[1] استفاده می‌کند. واگرایی KL بین سیاست فعلی و سیاست قدیمی در زمان آموزش محاسبه می‌شود. سپس با در نظر گرفتن یک آستانه مشخص، نرخ یادگیری تنظیم می‌شود:

$$D_{KL}\left(N\left(\mu_1,\sigma_1^2\right)\middle\|N\left(\mu_2,\sigma_2^2\right)\right)=\log\left(\frac{\sigma_2}{\sigma_1}\right)+\frac{\sigma_1^2+\left(\mu_1-\mu_2\right)^2}{2\sigma_2^2}-0.5 \qquad (۲۴.۳)$$

ضریب واگرایی KL مطابق با معادله (۲۴.۳) محاسبه می‌شود. منظور از $\sigma$ و $\mu$ در معادله فوق، به‌ترتیب میانگین و انحراف معیار توزیع اقدام‌های سیاست است.

پس از محاسبه ضریب KL، نرخ یادگیری با در نظر گرفتن یک آستانه مشخص تنظیم می‌شود:

$$\text{Learning Rate}=\begin{cases}\dfrac{\text{Default Learning Rate}}{1.5} & \text{if KL} > \text{Desired KL}*2\\ \text{Default Learning Rate}*1.5 & \text{if KL} < \text{Desired } \dfrac{\text{KL}}{2}\end{cases} \qquad (۲۵.۳)$$

سایر پارمترهای تنظیم‌شده نیز در جدول زیر آمده‌اند:

**جدول ۲.۳ پارامترهای الگوریتم PPO**

| پارامتر | مقدار |
|---|---|
| تعداد نمونه‌های ربات | ۴۰۹۶ |
| پارامتر برش | ۰/۲ |

[1] Kullback-Leibler

| | |
|---|---|
| ضریب زیان تابع ارزش | ۱ |
| ضریب آشفتگی | ۰/۰۱ |
| تعداد دوره | ۵ |
| تعداد بسته‌های کوچک | ۴ |
| نرخ یادگیری اولیه | ۰/۰۰۱ |
| $\gamma$ | ۰/۹۹ |
| $\lambda$ | ۰/۹۵ |

### ۷.۱.۳ تصادفی‌سازی

یکی از چالش‌های الگوریتم‌های یادگیری تقویتی پیاده‌سازی الگوریتم به‌دست آمده در شبیه‌سازی بر روی نمونه واقعی[1] است. راهکار موجود تصادفی‌سازی برخی پارامترها در شبیه‌سازی است. به این منظور موارد زیر در شبیه‌سازی یادگیری به صورت تصادفی تغییر می‌کنند:

- **ضریب اصطکاک زمین:** ضریب اصطکاک به‌صورت تصادفی و در بازه [۰/۵ ۱/۲۵] انتخاب می‌شود.
- **اعمال نویز بر روی مشاهدات:** بر خلاف شبیه‌سازی، در واقعیت اندازه‌گیری‌ها دقیق مشاهدات ممکن نیست. به این منظور برای پیشگیری از مشکلات پیاده‌سازی و نزدیک کردن شرایط شبیه‌سازی به واقعیت، بر روی تمامی مشاهدات به صورت مستقل و متناسب با مقدار نویز و بازه تغییرات هر کدام در واقعیت نویز اعمال می‌شود. هر یک از مشاهدات مقیاس‌شده در یک مقیاس نویز مشخص ضرب می‌شوند. سپس مقدار به‌دست آمده نیز در یک عدد به صورت تصادفی در بازه [۱ ۱-] ضرب می‌شود تا مقدار نویز اعمال شده محاسبه شود. مقیاس اعمال شده برای نویز هر یک از مشاهدات مقیاس‌شده در جدول ۳.۳ آمده‌است:

**جدول ۳.۳ مقیاس نویز مشاهدات**

| پارامتر | مقیاس |
|---|---|
| سرعت‌های خطی | ۰/۱ |
| سرعت‌های زاویه‌ای | ۰/۲ |
| جاذبه زمین | ۰/۰۵ |
| اندازه‌گیری‌های ارتفاع | ۰/۱ |

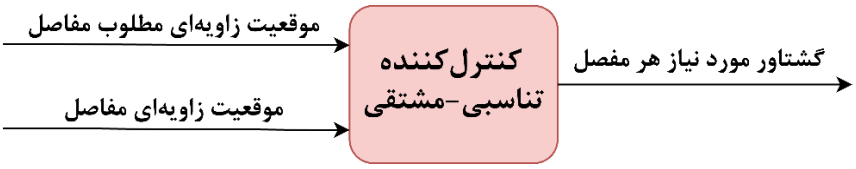
[1] Plant

| | |
|---|---|
| موقعیت زاویه‌ای مفاصل | ۰/۰۱ |
| سرعت زاویه‌ای مفاصل | ۱/۵ |

- **هل دادن تصادفی:** به منظور تولید یک سیاست مقاوم، هر ۱۵ ثانیه یک سرعت خطی به‌اندازه حداکثر $1\ \frac{m}{s}$ در جهات مختلف به مرکز جرم ربات اعمال می‌شود تا سیاست تولید شده بتواند با اغتشاش‌های خارجی مقابله کند.

## ۲.۳ کنترل‌کننده دوم

کنترل‌کننده دوم یک کنترل‌کننده تناسبی-مشتقی است.




**شکل ۵.۳ کنترل‌کننده دوم**

مطابق شکل ۵.۳ موقعیت زاویه‌ای مطلوب مفاصل که توسط شبکه عصبی محاسبه شده‌است به همراه بازخورد موقعیت زاویه‌ای هر یک از مفاصل ربات وارد کنترل کننده تناسبی-مشتقی می‌شود. سپس این کنترل‌کننده میزان گشتاور مورد نیاز برای هر مفصل به منظور رساندن آن مفصل از موقعیت زاویه‌ای فعلی‌اش به موقعیت زاویه‌ای مطلوب را محاسبه می‌کند.

### ۱.۲.۳ سازوکار کنترل‌کننده تناسبی-مشتقی استفاده شده

سازوکار کنترل‌کننده تناسبی-مشتقی استفاده شده بر اساس استفاده از خطای موقعیت زاویه‌ای هر یک از مفاصل و رساندن آن به صفر است. این رویکرد برای تمام ۱۲ مفصل ربات چهارپا به صورت مستقل دنبال می‌شود.

$$\text{torques} = k_p\left(\text{Desired DOF Positions} - \text{DOF Positions}\right) - k_d\left(\text{DOF Velocities}\right) \quad (۲۶.۳)$$

منظور از $k_p$ و $k_d$ در رابطه فوق به ترتیب بهره تناسبی و بهره مشتقی و منظور از DOF نیز هر یک از مفاصل می‌باشد. Positions و Velocities نیز به موقعیت زاویه‌ای و سرعت زاویه‌ای هر یک از مفاصل اشاره دارد. Desired DOF Positions نیز به صورت زیر محاسبه می‌شود:

$$\text{Desired DOF Positions} = (\text{Action Scale}) \cdot (\text{Action}) + \text{Default Angle} \qquad (۲۷.۳)$$

در معادله فوق Action Scale یک ثابت با مقدار ۰/۲۵ است.

### ۲.۲.۳ پارامترهای کنترل‌کننده تناسبی-مشتقی

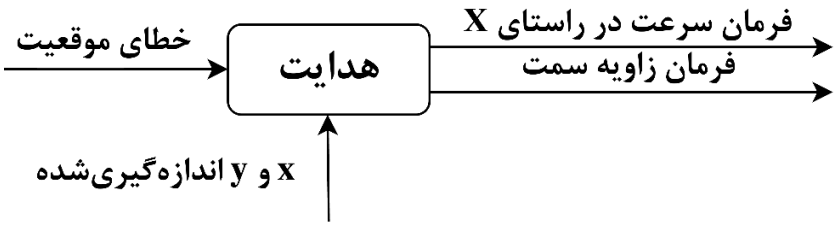
در این پژوهش، مقدار بهره تناسبی و بهره مشتقی برای تمام مفاصل یکسان در نظر گرفته می‌شود. این مقدار برای بهره تناسبی برابر با $۲۰ \frac{\text{N} \cdot \text{m}}{\text{rad}}$ و برای بهره مشتقی برابر با $۰/۵ \frac{\text{N} \cdot \text{m} \cdot \text{s}}{\text{rad}}$ است.

# ۴ طراحی حلقه هدایت

ربات چهارپا با استفاده از کنترل‌کننده‌های طراحی شده می‌تواند فرامین کنترلی سرعت در راستای محور x و y بدنی و فرمان کنترلی سرعت زاویه‌ای در راستای محور z بدنی (و یا زاویه سمت[1]) را دنبال کند. در راستای طراحی حلقه هدایت (به منظور تبدیل فرامین کنترلی سرعت به فرامین کنترلی موقعیت) دو رویکرد درنظر گرفته می‌شود. در رویکرد دوم، ربات با حرکت در راستای محور x و y بدنی، خود را به هدف می‌رساند.

## ۱.۴ رویکرد اول

در رویکرد اول ربات با حرکت در راستای محور x بدنی و تنظیم زاویه سمت خود به‌سوی هدف حرکت می‌کند.




**شکل ۱.۴ بلوک هدایت در رویکرد اول**

شکل ۱.۴ ورودی و خروجی‌های بلوک هدایت در رویکرد اول را نشان می‌دهد.

### ۱.۱.۴ تنظیم سرعت خطی در راستای x بدنی

به منظور تنظیم سرعت خطی در راستای x بدنی، از یک کنترل‌کننده تناسبی-انتگرالی-مشتقی (PID[2]) استفاده می‌شود:

[1] Heading Angle

[2] Proportional Integral Derivative

$$\left|\mathbf{v}^{B}_{\text{Desired}}\right| = \left|\mathbf{v}^{B}_{x_{\text{Desired}}}\right| = \mathrm{k_p}\left(\sqrt{\left(x_d - x_m\right)^2 + \left(y_d - y_m\right)^2}\right) + \mathrm{k_d}\left(-\sqrt{v_x^2 + v_y^2}\right) + \mathrm{k_i}\left(\int \sqrt{\left(x_d - x_m\right)^2 + \left(y_d - y_m\right)^2}\,\mathrm{d}t\right) \quad (۱.۴)$$

در معادله فوق مقادیر $\mathrm{k_p}$، $\mathrm{k_d}$ و $\mathrm{k_i}$ به‌ترتیب برابر با $\frac{3.5}{\mathrm{s}}$، ۶ و $\frac{0.5}{\mathrm{s}}$ است.

درصورت تولید فرمان سرعت با اندازه بزرگتر از $۲\frac{\mathrm{m}}{\mathrm{s}}$، اندازه آن به مقدار $۲\frac{\mathrm{m}}{\mathrm{s}}$ محدود می‌شود.

### ۲.۱.۴ تنظیم زاویه سمت

به منظور تنظیم زاویه سمت، در گام اول باید زاویه سمت مطلوب رو مشخص کرد:

$$\psi_{\text{Desired}} = \arctan\left(\frac{y_d - y_m}{x_d - x_m}\right) \quad (۲.۴)$$

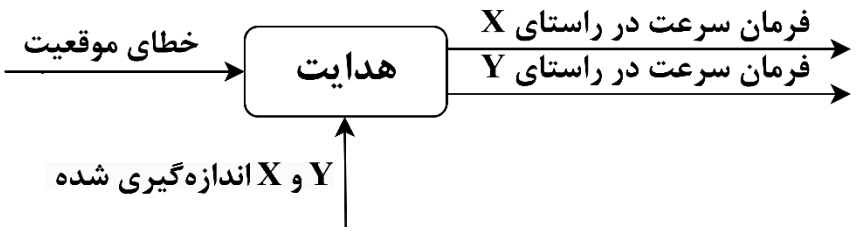
منظور از $x_d$ و $y_d$ در عبارت فوق به‌ترتیب موقعیت x مطلوب و y مطلوب و هم‌چنین منظور از $x_m$ و $y_m$ نیز به‌ترتیب موقعیت x اندازه‌گیری شده و y اندازه‌گیری شده در محیط شبیه‌ساز Isaac Gym است.

پس از مشخص شدن زاویه سمت مطلوب، یک کنترل‌کننده P با استفاده از خطای زاویه سمت مطلوب محاسبه‌شده و زاویه سمت اندازه‌گیری شده ربات در محیط شبیه‌ساز Isaac Gym ، سرعت زاویه‌ای مطلوب در راستای رسیدن به زاویه سمت مطلوب را محاسبه می‌کند:

$$\left|\omega\right| = \mathrm{k_p}\left(\psi_m - \psi_{\text{Desired}}\right) \quad (۳.۴)$$

## ۲.۴ رویکرد دوم

در این رویکرد، به صـــورت همزمان با حرکت مفاصـــل، در هر دو جهت محور x بدنی و y بدنی فرمان کنترلی سرعت تولید می‌شود. در این حالت حرکت عرضی ربات با شدت بیشتری انجام می‌شود.

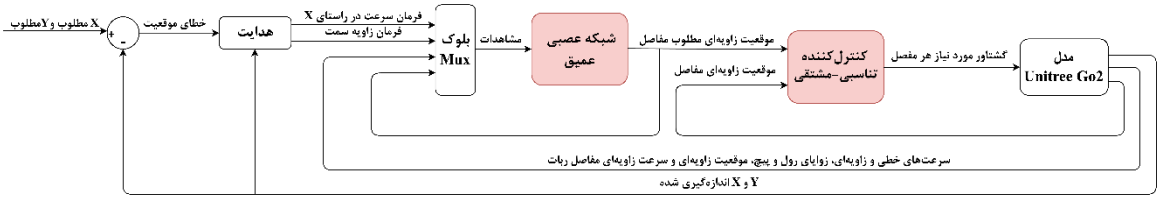



**شکل ۲.۴ بلوک هدایت در رویکرد دوم**

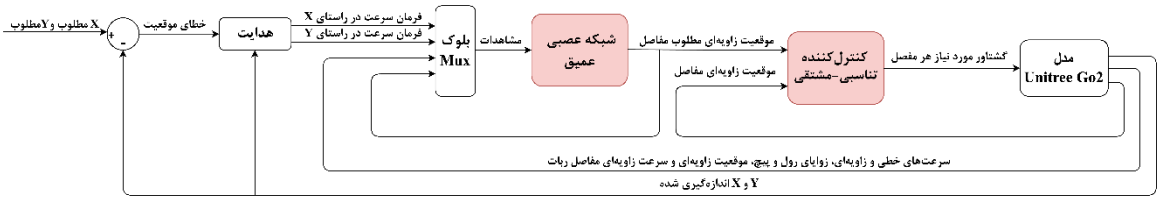
شکل ۲.۴ ورودی و خروجی‌های بلوک هدایت در رویکرد دوم را نشان می‌دهد. فرامین کنترلی موقعیت با استفاده از دو عدد کنترل‌کننده PID به فرامین کنترلی سرعت تبدیل می‌شوند.

# ۵ شبیه‌سازی

به‌منظور ارزیابی عملکرد کنترل‌کننده و حلقه هدایت طراحی‌شـــده، فرامین کنترلی در حالت‌های مختلف در قالب ورودی به ربات داده شده و عملکرد ربات در محیط شبیه‌ساز Isaac Gym شبیه‌سازی شده و ارزیابی می‌شـــود. پس از تکمیل حلقه هدایت، حلقه کنترلی ربات با توجه به نوع الگوریتم هدایت استفاده‌شده به صورت زیر خواهد بود:




**شکل ۱.۵ حلقه کنترلی کامل در رویکرد اول هدایت**



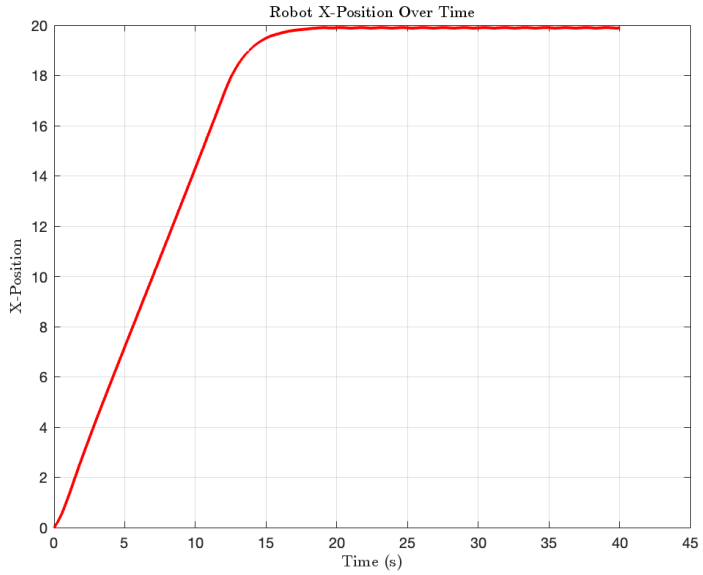

**شکل ۲.۵ حلقه کنترلی کامل در رویکرد دوم هدایت**

## ۱.۵ رسیدن به موقعیت مطلوب

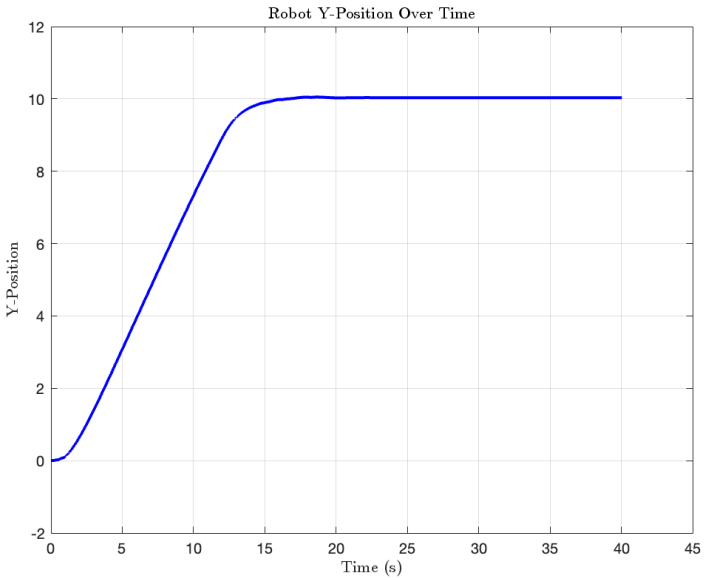
در این بخش، موقعیت x مطلوب ۲۰ متر و موقعیت y مطلوب نیز ۱۰ متر درنظر گرفته می‌شـود. ربات از یک نقطه تصادفی شروع به حرکت می‌کند و خود را موقعیت مطلوب می‌رساند. سپس از عملکرد ربات با استفاده از رویکرد اول هدایت در دو حالت ایده‌آل و حضور نویز و اغتشاش بررسی می‌شود.

### ۱.۱.۵ عدم وجود نویز و اغتشاش

در این حالت حرکت ربات در محیط ایده‌آل و در غیاب نویز و اغتشاشات خارجی شبیه‌سازی می‌شود.




**شکل ۳.۵ نمودار موقعیت x ربات نسبت به زمان در محیط ایده‌آل با رویکرد اول هدایت**

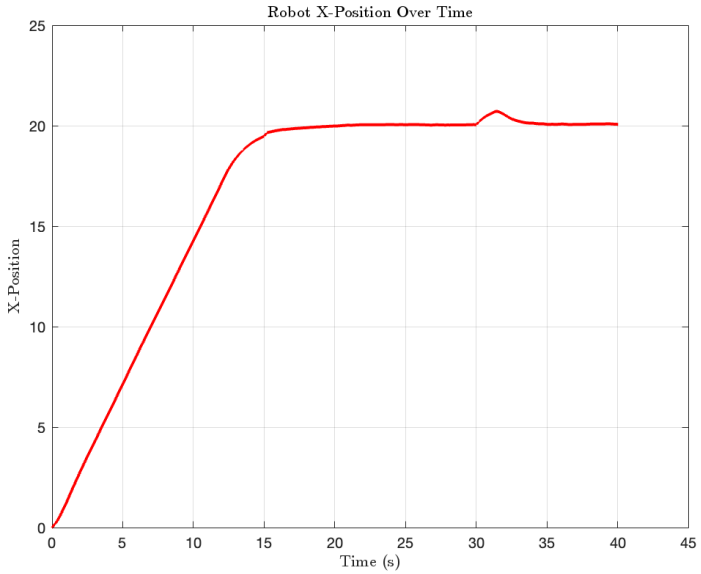



**شکل ۴.۵ نمودار موقعیت y ربات نسبت به زمان در محیط ایده‌آل با رویکرد اول هدایت**

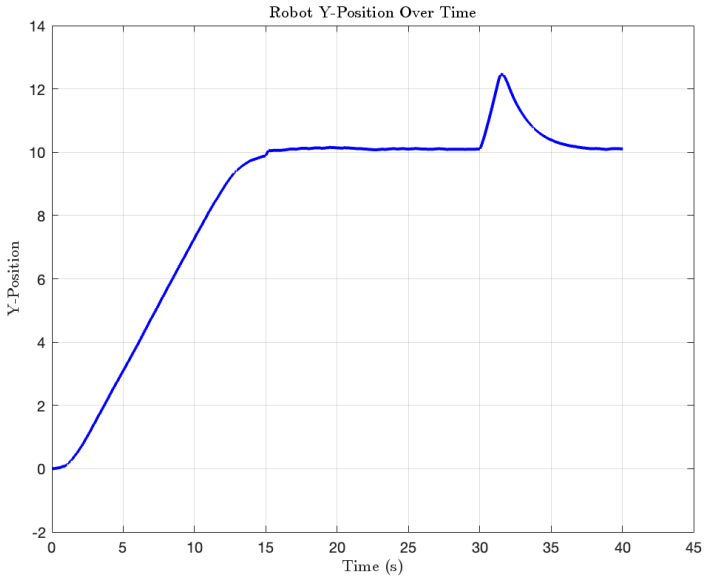
مطابق شکل ۳.۵ و شکل ۴.۵ ربات در مدت زمان قابل‌قبول و با دقت خوبی به مقصد می‌رسد.

### ۲.۱.۵ وجود نویز و اغتشاش

در این حالت بر تمامی مشاهدات ربات به صورت متناسب نویز اعمال می‌شود. ضریب اصطکاک سطح زمین نیز به صورت تصادفی تغییر می‌کند. هم‌چنین در طی بازه‌های ۱۵ ثانیه‌ای، به ربات در راستاهای مختلف سرعت خطی با مقدار $۱ \frac{m}{s}$ اعمال می‌شود.




**شکل ۵.۵ نمودار موقعیت x ربات نسبت به زمان در حضور اغتشاش با رویکرد اول هدایت**




**شکل ۶.۵ نمودار موقعیت y ربات نسبت به زمان در حضور اغتشاش با رویکرد اول هدایت**

مطابق شکل ۵.۵ و شکل ۶.۵ ربات با دقت خوبی به موقعیت مطلوب می‌رسد. در ثانیه ۱۵ اغتشاش به‌صورت تقریبی در راستای حرکت ربات بوده و باعث انحراف جزئی ربات می‌شود ولی ربات تعادل خود را حفظ کرده و مسیر را ادامه می‌دهد. در ثانیه ۳۰ اغتشاش مؤلفه عرضی قوی‌تری دارد و باعث انحراف شدیدتر ربات می‌شود ولی ربات درنهایت تعادل خود را حفظ کرده و به دوباره به سمت موقعیت مطلوب حرکت می‌کند.

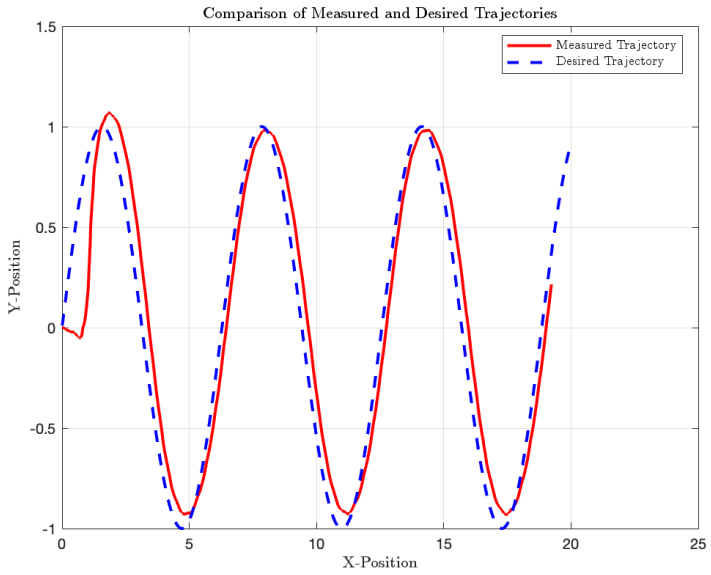
## ۲.۵ دنبال کردن موج سینوسی

هدف در این حالت ایجاد یک مسیر به‌شکل موج سینوسی با دامنه یک متر و دوره تناوب $4\pi$ است. به این منظور موقعیت x مطلوب به صورت خطی و متناسب با زمان به ربات داده می‌شود. ورودی y مطلوب نیز در قالب سینوس موقعیت x مطلوب به ربات داده می‌شود. سپس عملکرد ربات در با رویکرد اول هدایت در محیط شبیه‌ساز Isaac Gym ارزیابی می‌شود.

$$\begin{cases} x_{\text{Desired}} = 0.5t \\ y_{\text{Desired}} = \sin(0.5t) \end{cases} \quad (۱.۵)$$

### ۱.۲.۵ عدم وجود نویز و اغتشاش

در این حالت حرکت ربات در محیط ایده‌آل و در غیاب نویز و اغتشاشات خارجی شبیه‌سازی می‌شود.

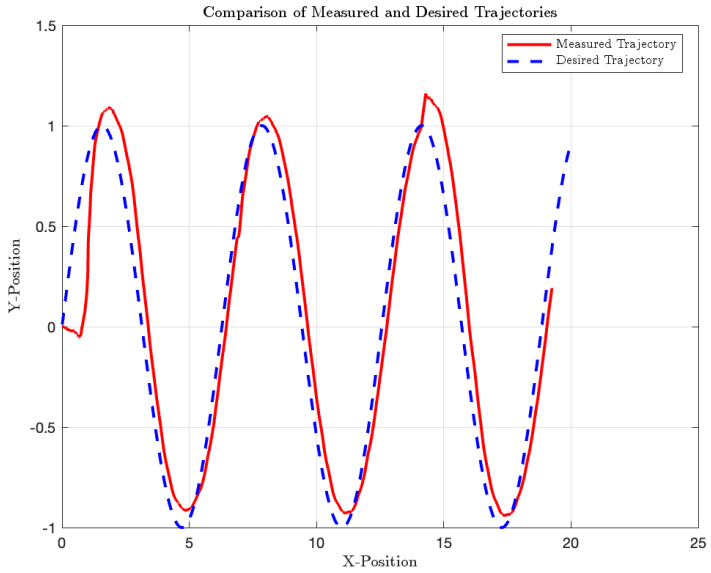



**شکل ۷.۵ دنبال کردن مسیر سینوسی در محیط ایده‌آل با رویکرد اول هدایت**

مطابق شکل ۷.۵ ربات با وجود کوچک بودن نسبی دامنه، به‌خوبی مسیر را دنبال می‌کند.

### ۲.۲.۵ وجود نویز و اغتشاش

در این حالت بر تمامی مشاهدات ربات به صورت متناسب نویز اعمال می‌شود. ضریب اصطکاک سطح زمین نیز به صورت تصادفی تغییر می‌کند. هم‌چنین در طی بازه‌های ۱۵ ثانیه‌ای، به ربات در راستاهای مختلف سرعت خطی با مقدار $۱ \frac{m}{s}$ اعمال می‌شود.




**شکل ۸.۵ دنبال کردن موج سینوسی در حضور اغتشاش با رویکرد اول هدایت**

مطابق شکل ۸.۵ ربات باز هم موفق به تعقیب موج سینوسی با دقت بسیار بالایی می‌شود. در ثانیه ۱۵ اغتشاش به‌صورت تقریبی در راستای حرکت ربات بوده و باعث انحراف جزئی ربات می‌شود ولی ربات تعادل خود را حفظ کرده و مسیر را ادامه می‌دهد. در ثانیه ۳۰ اغتشاش مؤلفه عرضی قوی‌تری دارد و باعث انحراف شدیدتر ربات می‌شود ولی ربات درنهایت تعادل خود را حفظ کرده و به دوباره به سمت موقعیت مطلوب حرکت می‌کند.

# ۶ نتیجه‌گیری

در گزارش حاضـــر نحوه اســـتفاده از یادگیری تقویتی برای کنترل ربات چهارپا و هم‌چنین طراحی حلقه هدایت برای آن شرح داده‌شد و عملکرد حلقه کنترلی طراحی شده در محیط Isaac Gym مورد ارزیابی قرار گرفت. تعقیب موفق موج سینوسی، استفاده از الگوریتم‌های طراحی مسیر[1] که مبتنی بر تولید نقطه راه[2] هستند را در آینده ممکن می‌کند.

## ۱.۶ نوآوری‌های پایان‌نامه

نوآوری‌های این پایان‌نامه شامل موارد زیر است:

- طراحی الگوریتم هدایت به‌منظور حرکت موزون ربات چهارپا.

## ۲.۶ پیشنهادها برای ادامه کار

پیشنهادهایی که برای ادامه این کار وجود دارد، شامل موارد زیر است:

- پیاده‌سازی فرآیند یادگیری در زمین ناهوار به‌منظور افزایش دامنه حرکتی ربات چهارپا.
- آموزش سیاست‌های جدید به‌منظور انجام حرکات نمایشی توسط ربات چهارپا.
- پیاده‌سازی بر روی ربات واقعی Unitree Go2.

[1] Path Planning

[2] Waypoint

# منابع و مراجع


[۱] نام و نام خانوادگی مولفان یا مترجمان؛ *عنوان کتاب*؛ عنوان فرعی کتاب (جزئیات عنوان کتاب در صورت وجود داخل پرانتز)، نام سایر افراد دخیل در تالیف یا ترجمه، ناشر، محل انتشار، شماره جلد، شماره ویرایش، سال انتشار به عدد.

[۲] محمدباقر منهاج؛ *هوش محاسباتی* (جلد اول: مبانی شبکه‌های عصبی)، انتشارات دانشگاه صنعتی امیرکبیر، تهران، ویرایش اول، ۱۳۷۹.

[۳] نام و نام خانوادگی مولفان؛ «عنوان مقاله به‌صورت عادی و داخل گیومه»، *نام کامل مجله به صورت ایتالیک*، شماره دوره یا جلد، شماره مجله، شماره صفحات، سال انتشار.

[۴] نام و نام خانوادگی مجری یا مجریان؛ *عنوان طرح پژوهشی به‌صورت ایتالیک*، شماره ثبت، نام کامل محل انجام و سفارش دهنده، سال انجام طرح.

[۵] مریم اسدی و خیرالنسا سیفی؛ *دستورالعمل نحوه نگارش پایان‌نامه کارشناسی‌ارشد و رساله دکتری*، ویرایش دوم، دانشگاه صنعتی شریف، ۱۳۹۲.

[۶] Book authors' names; *Book Title in Italic*, Edition number, Publisher, publication Date.

[۷] Van de Vegte, J.; *Feedback Control Systems*, 2nd Edition, Prentice Hall, 1990.

[۸] Authors' names separated by commas; "Paper title in Regular Times New Roman 12pt", *Paper Address in Italic*, Publishing Place, paper page, Year of Publish.

[۹] Safonov, M.; "Stability margins of diagonally perturbed multivariable feedback systems", *IEEE Proceedings*, Part D, p. p. 251-256, Nov. 1982.

[۱۰] Company Name/ Person Name; Page Title; *Internet Address*.

[۱۱] Hadi Nobahari and Alireza Sharifi, "Continuous ant colony filter applied to online estimation and compensation of ground effect in automatic landing of quadrotor", *Engineering Applications of Artificial Intelligence*, Vol. 32, June 2014, pp. 100-111, 2014.

[۱۲] H. Nobahari, S. A. Hosseini Kordkheili and S. SarayGordAfshari, 'Hardware in the Loop Optimization of an Active Vibration Controller in a Flexible Beam Structure Using Evolutionary Algorithms', *Journal of Intelligent Material*

*Systems and Structures*, Vol. 25, Issue 10, July 2014, pp. 1211-1223, DOI: 10.1177/1045389X13502874.

# پیوست‌ها

# Thesis Title


## Abstract

Write English abstract of your thesis here.



## Keywords

Write four to seven keywords, separated by comma.

**Sharif University of Technology**
**.Department of ……**

**MSc Thesis (PhD Thesis)**
**Area: ……**

# Thesis Title

**By:**
**Author Name**

**Advisor:**
**Advisor Name**

**Month and Year**