دانشگاه صنعتی شریف
دانشکده‌ی مهندسی هوافضا

پروژه کارشناسی ارشد
مهندسی فضا

عنوان:

هدایت یادگیری تقویتی مقاوم مبتنی بر بازی دیفرانسیلی در محیط‌های پویای چندجسمی با پیشران کم

نگارش:

علی بنی اسد

استاد راهنما:

دکتر هادی نوبهاری

دی ۱۴۰۳

بسم الله الرحمن الرحیم

به نام خدا

دانشگاه صنعتی شریف

دانشکده‌ی مهندسی هوافضا

پروژه کارشناسی ارشد

عنوان: هدایت یادگیری تقویتی مقاوم مبتنی بر بازی دیفرانسیلی در محیط‌های پویای چندجسمی با پیشران کم

نگارش: علی بنی اسد

کمیته‌ی ممتحنین

استاد راهنما: دکتر هادی نوبهاری امضاء:

استاد مشاور: استاد مشاور امضاء:

استاد مدعو: استاد ممتحن امضاء:

تاریخ:

سپاس

از استاد بزرگوارم جناب آقای دکتر نوبهاری که با کمک‌ها و راهنمایی‌های بی‌دریغشان، بنده را در انجام این پروژه یاری داده‌اند، تشکر و قدردانی می‌کنم. از پدر دلسوزم ممنونم که در انجام این پروژه مرا یاری نمود. در نهایت در کمال تواضع، با تمام وجود بر دستان مادرم بوسه می‌زنم که اگر حمایت بی‌دریغش، نگاه مهربانش و دستان گرمش نبود برگ برگ این دست نوشته و پروژه وجود نداشت.

چکیده

در این پژوهش، از یک روش مبتنی بر نظریه بازی[1] به‌منظور کنترل وضعیت استند سه درجه آزادی چهارپره استفاده شده‌است. در این روش بازیکن اول سعی در ردگیری ورودی مطلوب می‌کند و بازیکن دوم با ایجاد اغتشاش سعی در ایجاد خطا در ردگیری بازیکن اول می‌کند. در این روش انتخاب حرکت با استفاده از تعادل نش[2] که با فرض بدترین حرکت دیگر بازیکن است، انجام می‌شود. این روش نسبت به اغتشاش ورودی و همچنین نسبت به عدم قطعیت مدل‌سازی می‌تواند مقاوم باشد. برای ارزیابی عملکرد این روش ابتدا شبیه‌سازی‌هایی در محیط سیمولینک انجام شده‌است و سپس، با پیاده‌سازی روی استند سه درجه آزادی صحت عملکرد کنترل‌کننده تایید شده‌است.

کلیدواژه‌ها: چهارپره، بازی دیفرانسیلی، نظریه بازی، تعادل نش، استند سه درجه آزادی، مدل‌مبنا، تنظیم‌کننده مربعی خطی

[1] Game Theory
[2] Nash Equilibrium

فهرست مطالب

فهرست جداول

فهرست تصاویر

فهرست الگوریتم‌ها

فصل ۱

مقدمه

۱-۱ انگیزه پژوهش

۱-۲ تعریف مسئله

در سال‌های اخیر، پیشرفت‌های فناوری در زمینه‌های مختلف، از جمله کنترل پرواز، پردازش سیگنال و هوش مصنوعی، به افزایش کاربردهای ماهواره با پیشران کم در منظومه زمین–ماه کمک کرده است. ماهواره با پیشران کم می‌تواند برای تعقیب ماهواره‌ها، انتقال مداری و استقرار ماهواره‌ها استفاده شود. روش‌های هدایت بهینه قدیمی جهت کنترل ماهواره‌ها اغلب نیازمند فرضیات ساده‌کننده، منابع محاسباتی فراوان و شرایط اولیه مناسب هستند. الگوریتم‌های مبتنی بر یادگیری تقویتی این توانایی را دارند که بدون مشکلات اشاره‌شده هدایت ماهواره را انجام دهند. به همین دلیل، این الگوریتم‌ها می‌توانند امکان محاسبات درونی (On-board Computing) را فراهم می‌کنند.

۱-۳ اهداف و نوآوری

۱-۴ یادگیری تقویتی

۱-۵ یادگیری تقویتی چند عاملی

۱-۶ محتوای گزارش

فصل ۲

پیشینه پژوهش

۲-۱ ماموریت‌های بین مداری

هدایت فضاپیماها معمولاً با استفاده از ایستگاه‌های زمینی انجام می‌شود. با این حال، این تکنیک‌ها دارای محدودیت‌هایی از جمله حساسیت به قطع ارتباطات، تاخیرهای زمانی، و محدودیت‌های منابع محاسباتی هستند. الگوریتم‌های یادگیری تقویتی و بازی‌های دیفرانسیلی می‌توانند برای بهبود قابلیت‌های هدایت فضاپیماها، از جمله مقاومت در برابر تغییرات محیطی، کاهش تاخیرهای ناشی از ارتباطات زمینی، و افزایش کارایی محاسباتی، مورد استفاده قرار گیرند.

هدایت فضاپیماها معمولاً پیش از پرواز انجام می‌شود. این روش‌ها می‌توانند از تکنیک‌های بهینه‌سازی فراگیر [۱] یا برنامه‌نویسی غیرخطی برای تولید مسیرها و فرمان‌های کنترلی بهینه استفاده کنند. با این حال، این روش‌ها معمولا حجم محاسباتی زیادی دارند و برای استفاده درون‌سفینه نامناسب هستند [۲]. یادگیری ماشین می‌تواند برای بهبود قابلیت‌های هدایت فضاپیماها استفاده شود. کنترل‌کننده شبکه عصبی حلقه‌بسته می‌تواند برای محاسبه سریع و خودکار تاریخچه کنترل استفاده شود. یادگیری تقویتی نیز می‌تواند برای یادگیری رفتارهای هدایت بهینه استفاده شود.

روش‌های هدایت و بهینه‌سازی مسیر فضاپیماها به‌طور کلی به راه‌حل‌های اولیه مناسب نیاز دارند. در مسائل چند جسمی، طراحان مسیر اغلب حدس‌های اولیه کم‌هزینه‌ای برای انتقال‌ها با استفاده از نظریه سیستم‌های دینامیکی و منیفولدهای ثابت [۳،۴] ایجاد می‌کنند.

شبکه‌های عصبی ویژگی‌های جذابی برای فعال‌سازی هدایت در فضاپیما دارند. به‌عنوان مثال، شبکه‌های عصبی می‌توانند به‌طور مستقیم از تخمین‌های وضعیت به دستورهای پیشران کنترلی که با محدودیت‌های مأموریت

سازگار است، برسند. عملکرد هدایت شبکه‌های عصبی در مطالعاتی مانند فرود بر سیارات [۵]، عملیات نزدیکی به سیارات [۶] و کنترل فضاپیما با پیشران ازدست‌رفته [۷] نشان داده شده است. تازه‌ترین پیشرفت‌های تکنیک‌های یادگیری ماشین در مسائل خودکارسازی درونی به‌طور گسترده‌ای مورد مطالعه قرار گرفته‌اند؛ از پژوهش‌های اولیه تا توانایی‌های پیاده‌سازی. به‌عنوان مثال، الگوریتم‌های یادگیری ماشین ابتدایی در فضاپیماهای مریخی نبرد برای کمک به شناسایی ویژگی‌های زمین‌شناسی تعبیه شده‌اند. الگوریتم AEGIS توانایی انتخاب خودکار هدف توسط یک دوربین در داخل فضاپیماهای Spirit، Opportunity و Curiosity را فعال دارد [۸]. درکامپیوتر پرواز اصلی، فرآیند دقت افزایی (Refinement Process) نیاز به ۹۴ تا ۹۶ ثانیه دارد [۹]، که به طور قابل توجهی کمتر از زمان مورد نیاز برای ارسال تصاویر به زمین و انتظار برای انتخاب دستی توسط دانشمندان است. برنامه‌های آینده برای کاربردهای یادگیری ماشین درون‌سفینه شامل توانایی‌های رباتیکی درون‌سفینه برای فضاپیمای Perseverance [۱۰، ۱۱] و شناسایی عیب برای Europa Clipper [۱۲] می‌شود. الگوریتم‌های یادگیری ماشین پتانسیلی برای سهم مهمی در مأموریت‌های اتوماسیون آینده دارند.

علاوه بر رباتیک سیاره‌ای، پژوهش‌های مختلفی به استفاده از تکنیک‌های مختلف یادگیری ماشین در مسائل نجومی پرداخته‌اند. در طراحی مسیر عملکرد رگرسیون معمولاً مؤثرتر هست. به عنوان مثال، از یک شبکه عصبی (NN) در بهینه‌سازی مسیرهای رانشگر کم‌پیشران استفاده شده است [۱۳]. پژوهش‌های جدید شامل شناسایی انتقال‌های هتروکلینیک [۱۴]، اصلاح مسیر رانشگر کم‌پیشران [۱۵] و تجزیه و تحلیل مشکلات ازدست‌رفتن رانشگر [۷] می‌شود.

تکنیک‌های یادگیری نظارتی می‌توانند نتایج مطلوبی تولید کنند؛ اما، دارای محدودیت‌های قابل توجهی هستند. یکی از این محدودیت‌ها این است که این رویکردها بر وجود دانش پیش از فرآیند تصمیم‌گیری متکی هستند. این امر مستلزم دقیق‌بودن داده‌های تولیدشده توسط کاربر برای نتایج مطلوب و همچنین وجود تکنیک‌های موجود برای حل مشکل کنونی و تولید داده است.

در سال‌های اخیر، قابلیت یادگیری تقویتی (RL) در دستیابی به عملکرد بهینه در دامنه‌هایی با ابهام محیطی قابل توجه، به اثبات رسیده است [۱۶، ۱۷]. هدایت انجام‌شده توسط RL را می‌توان به‌صورت گسترده بر اساس فاز پرواز دسته‌بندی کرد. مسائل فرود [۱۸، ۱۹] و عملیات در نزدیکی اجسام کوچک [۵، ۶]، از حوزه‌های پژوهشی هستند که از RL استفاده می‌کنند. تحقیقات دیگر شامل مواجهه تداخل خارجی جوی [۲۰]، نگهداری ایستگاهی [۲۱] و هدایت به‌صورت جلوگیری از شناسایی [۲۲] است. مطالعاتی که فضاپیماهای رانشگر کم‌پیشران را در یک چارچوب دینامیکی چند بدنی با استفاده از RL انجام‌شده‌است، شامل طراحی انتقال با استفاده از Q-learning [۲۳]، Proximal Policy Optimization [۲۴] و هدایت نزدیکی مدار [۲۵] است.

۲-۲ یادگیری تقویتی

۳-۲ یادگیری تقویتی چندعاملی

فصل ۳

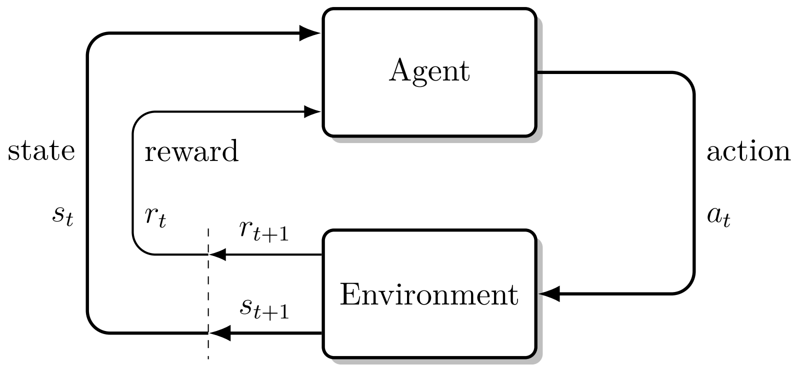
یادگیری تقویتی

در این فصل به بررسی یادگیری تقویتی پرداخته شده‌است. ابتدا در فصل ۳-۱ مفاهیم اولیه یادگیری تقویتی ارائه شده‌است. در ادامه عامل‌های گرادیان سیاست عمیق قطعی ۳-۲، گرادیان سیاست عمیق قطعی تاخیری دوگانه ۳-۳، عملگر نقاد نرم ۳-۴ و بهینه‌سازی سیاست مجاور ۳-۵ توضیح داده شده‌است.

۳-۱ مفاهیم اولیه

دو بخش اصلی یادگیری تقویتی[1] شامل عامل[2] و محیط[3] است. عامل در محیط قرار دارد و با آن در تعامل است. 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 روی محیط عمل می کند، محیط تغییر می‌کند؛ اما، ممکن است محیط به تنهایی نیز تغییر کند. عامل همچنین یک سیگنال پاداش[4] از محیط دریافت می‌کند؛ سیگنالی که به عامل می‌گوید وضعیت تعامل فعلی آن با محیط چقدر خوب یا بد است. هدف عامل بیشینه‌کردن پاداش انباشته خود است که برگشت[5] نام دارد. یادگیری تقویتی به روش‌هایی گفته می‌شود که در آن‌ها عامل رفتارهای مناسب برای رسیدن به هدف خود را می‌آموزد. در شکل ۳-۱ تعامل بین محیط و عامل نشان داده شده‌است.

[1] Reinforcement Learning (RL)
[2] Agent
[3] Environment
[4] Reward
[5] Return



شکل ۳-۱: حلقه تعامل عامل و محیط

۳-۱-۱ حالت و مشاهدات

حالت[۶] (s) توصیف کاملی از وضعیت محیط است. همه‌ی اطلاعات محیط در حالت وجود دارد. مشاهده[۷] (o) یک توصیف جزئی از حالت است که ممکن است شامل تمامی اطلاعات نباشد. در این پژوهش مشاهده توصیف کاملی از محیط هست؛ در نتیجه، حالت و مشاهده برابر هستند.

۳-۱-۲ فضای عمل

فضای عمل (a) در یادگیری تقویتی، مجموعه‌ای از تمام اقداماتی است که یک عامل می‌تواند در محیط انجام دهد. این فضا می‌تواند گسسته[۸] یا پیوسته[۹] باشد. در این پژوهش فضای عمل پیوسته و محدود به یک بازه مشخص است.

۳-۱-۳ سیاست

سیاست[۱۰] قاعده‌ای است که یک عامل برای تصمیم‌گیری در مورد اقدامات خود استفاده می‌کند. در این پژوهش به تناسب الگوریتم پیاده‌سازی‌شده از سیاست قطعی[۱۱] یا تصادفی[۱۲] استفاده شده‌است که به دو صورت زیر نشان

[6] State
[7] Observation
[8] Discrete
[9] Continuous
[10] Policy
[11] Deterministic
[12] Stochastic

داده می‌شود:

$$a_t = \mu(s_t) \tag{۳-۱}$$

$$a_t \sim \pi(\cdot|s_t) \tag{۳-۲}$$

که زیروند t بیانگر زمان است. در یادگیری تقویتی عمیق از سیاست‌های پارامتری‌شده استفاده می‌شود. خروجی این سیاست‌ها تابعی پارامترهای سیاست (وزن‌ها و بایاس‌های یک شبکه عصبی) هستند که می‌توان از الگوریتم‌های بهینه‌سازی جهت تعیین مقدار بهینه این پارامترها استفاده‌کرد. در این پژوهش پارامترهای سیاست با θ نشان داده شده‌است و سپس نماد آن به‌عنوان زیروند سیاست مانند معادله (۳-۳) نشان داده شده‌است.

$$\begin{aligned} a_t &= \mu_\theta(s_t) \\ a_t &\sim \pi_\theta(\cdot|s_t) \end{aligned} \tag{۳-۳}$$

۳-۱-۴ مسیر

یک مسیر[۱۳] یک توالی از حالت‌ها و عمل‌ها در محیط است.

$$\tau = (s_0, a_0, s_1, a_1, \cdots) \tag{۳-۴}$$

گذار حالت[۱۴] به اتفاقاتی که در محیط بین زمان t در حالت s_t و زمان $t+1$ در حالت s_{t+1} رخ می‌دهد، گفته می‌شود. این گذارها توسط قوانین طبیعی محیط انجام می‌شوند و تنها به آخرین اقدام انجام‌شده توسط عامل (a_t) بستگی دارند. گذار حالت را می‌توان به‌صورت زیر تعریف کرد.

$$s_{t+1} = f(s_t, a_t) \tag{۳-۵}$$

۳-۱-۵ تابع پاداش و برگشت

تابع پاداش[۱۵] در حالت کلی به حالت فعلی محیط، آخرین عمل انجام‌شده و حالت بعدی محیط بستگی دارد. تابع پاداش را می‌توان به‌صورت زیر تعریف کرد.

$$r_t = R(s_t, a_t, s_{t+1}) \tag{۳-۶}$$

[13] Trajectory
[14] State Transition
[15] Reward Function

در این پژوهش، پاداش تنها تابعی از جفتِ حالت–عمل $(r_t = R(s_t, a_t))$ فرض شده‌است. هدف عامل این است که مجموع پاداش‌های به‌دست‌آمده در طول یک مسیر را به حداکثر برساند. در این پژوهش مجموع پاداش‌ها در طول یک مسیر را با نماد $R(\tau)$ نشان داده شده‌است و به آن تابع برگشت[۱۶] گفته می‌شود. یکی از انواع برگشت، برگشت بدون تنزیل[۱۷] با افق محدود[۱۸] است که مجموع پاداش‌های به‌دست‌آمده در یک بازه زمانی ثابت و از مسیر τ است که در معادله (۳–۷) نشان داده شده‌است.

$$R(\tau) = \sum_{t=0}^{T} r_t \tag{۳–۷}$$

نوع دیگری از برگشت، برگشت تنزیل‌شده با افق نامحدود[۱۹] است که مجموع همه پاداش‌هایی است که تا به حال توسط عامل به‌دست آمده‌است. اما، فاصله زمانی تا دریافت پاداش باعث تنزیل ارزش آن می‌شود. این معادله برگشت (۳–۸) شامل یک فاکتور تنزیل[۲۰] با نماد γ است که عددی بین صفر و یک است.

$$R(\tau) = \sum_{t=0}^{\infty} \gamma^t r_t \tag{۳–۸}$$

۳–۱–۶ ارزش در یادگیری تقویتی

در یادگیری تقویتی، دانستن ارزش[۲۱] یک حالت یا جفتِ حالت–عمل ضروری است. منظور از ارزش، برگشت مورد انتظار[۲۲] است. یعنی اگر از آن حالت یا جفت حالت–عمل شروع شود و سپس برای همیشه طبق یک سیاست خاص عمل شود، به‌طور میانگین چه مقدار پاداش دریافت خواهد شد. توابع ارزش تقریباً در تمام الگوریتم‌های یادگیری تقویتی به کار می‌روند. در اینجا به چهار تابع مهم اشاره شده‌است.

۱. تابع ارزش تحت سیاست[۲۳] $(V^{\pi}(s))$: خروجی این تابع برگشت مورد انتظار است در صورتی که از حالت s شروع شود و همیشه طبق سیاست π عمل شود و به‌صورت زیر بیان می‌شود:

$$V^{\pi}(s) = \underset{\tau \sim \pi}{\mathbb{E}} [R(\tau) | s_0 = s] \tag{۳–۹}$$

۲. تابع ارزش–عمل تحت سیاست[۲۴] $(Q^{\pi}(s, a))$: خروجی این تابع برگشت مورد انتظار است در صورتی که از حالت s شروع شود، یک اقدام دلخواه a (که ممکن است از سیاست π نباشد) انجام شود و سپس

[16] Return
[17] Discount
[18] Finite-Horizon Undiscounted Return
[19] Infinite-Horizon Discounted Return
[20] Discount Factor
[21] Value
[22] Expected Return
[23] On-Policy Value Function
[24] On-Policy Action-Value Function

برای همیشه طبق سیاست π عمل شود و به‌صورت زیر بیان می‌شود:

$$Q^{\pi}(s,a) = \underset{\tau\sim\pi}{\mathbb{E}}[R(\tau)|s_0 = s, a_0 = a] \quad (۱۰-۳)$$

۳. تابع ارزش بهینه[۲۵] ($V^*(s)$): خروجی این تابع برگشت مورد انتظار است در صورتی که از حالت s شروع شود و همیشه طبق سیاست بهینه در محیط عمل شود و به‌صورت زیر بیان می‌شود:

$$V^*(s) = \max_{\pi}(V^{\pi}(s)) \quad (۱۱-۳)$$

۴. تابع ارزش-عمل بهینه[۲۶] ($Q^*(s,a)$): خروجی این تابع برگشت مورد انتظار است در صورتی که از حالت s شروع شود، یک اقدام دلخواه a انجام شود و سپس برای همیشه طبق سیاست بهینه در محیط عمل شود و به‌صورت زیر بیان می‌شود:

$$Q^*(s,a) = \max_{\pi}(Q^{\pi}(s,a)) \quad (۱۲-۳)$$

۳-۱-۷ معادلات بلمن

توابع ارزش اشاره‌شده از معادلات خاصی که به آن‌ها معادلات بلمن گفته می‌شود، پیروی می‌کنند. ایده اصلی پشت معادلات بلمن این است که ارزش نقطه شروع برابر است با پاداشی است که انتظار دارید از آنجا دریافت کنید، به علاوه ارزش مکانی که بعداً به آنجا می‌رسید. معادلات بلمن برای توابع ارزش سیاست محور به شرح زیر هستند:

$$V^{\pi}(s) = \underset{\substack{a\sim\pi \\ s'\sim P}}{\mathrm{E}}[r(s,a) + \gamma V^{\pi}(s')] \quad (۱۳-۳)$$

$$Q^{\pi}(s,a) = r(s,a) + \underset{\substack{a\sim\pi \\ s'\sim P}}{\mathrm{E}}\left[\gamma \underset{a'\sim\pi}{\mathrm{E}}[Q^{\pi}(s',a')]\right] \quad (۱۴-۳)$$

که در آن $V^{\pi}(s)$ تابع ارزش حالت s تحت سیاست π است؛ $Q^{\pi}(s,a)$ تابع ارزش عمل a در حالت s تحت سیاست π است؛ $R(s,a)$ پاداش دریافتی پس از انجام عمل a در حالت s است؛ γ ضریب تنزیل است که ارزش پاداش‌های آینده را کاهش می‌دهد؛ $s' \sim P(\cdot|s,a)$ نشان می‌دهد که حالت بعدی s' از توزیع انتقال محیط P با شرط‌های s و a نمونه‌برداری می‌شود؛ و $a' \sim \pi(\cdot|s')$ نشان می‌دهد که عمل بعدی a' از سیاست

[25] Optimal Value Function
[26] Optimal Action-Value Function

π با شرط حالت جدید s' نمونه‌برداری می‌شود. این معادلات بیانگر این هستند که ارزش یک حالت یا عمل، مجموع پاداش مورد انتظار آن و ارزش حالت بعدی است که بر اساس سیاست فعلی تعیین می‌شود. معادلات بلمن برای توابع ارزش بهینه به شرح زیر هستند:

$$V^*(s) = \max_a \mathop{\mathrm{E}}_{s'\sim P}[r(s,a) + \gamma V^*(s')] \tag{۳-۱۵}$$

$$Q^*(s,a) = r(s,a) + \mathop{\mathrm{E}}_{s'\sim P}\left[\gamma \max_{a'} Q^*(s',a')\right] \tag{۳-۱۶}$$

تفاوت حیاتی بین معادلات بلمن برای توابع ارزش سیاست محور و توابع ارزش بهینه، عدم حضور یا حضور عملگر $\max$ بر روی اعمال است. حضور آن منعکس‌کننده این است که هرگاه عامل بتواند عمل خود را انتخاب کند، برای عمل بهینه، باید هر عملی را که منجر به بالاترین ارزش می‌شود انتخاب کند.

۳-۱-۸ تابع مزیت

گاهی در یادگیری تقویتی، نیازی به توصیف میزان خوبی یک عمل به صورت مطلق نیست، بلکه تنها می‌خواهیم بدانیم که چه مقدار بهتر از سایر اعمال به‌طور متوسط است. به عبارت دیگر، مزیت نسبی آن عمل مورد بررسی قرار می‌گیرد. این مفهوم با تابع مزیت[۲۷] توضیح داده می‌شود.

تابع مزیت $A^\pi(s,a)$ که مربوط به سیاست π است، توصیف می‌کند که انجام یک عمل خاص a در حالت s چقدر بهتر از انتخاب تصادفی یک عمل بر اساس $\pi(\cdot|s)$ است، با فرض اینکه شما برای همیشه پس از آن مطابق با π عمل می‌کنید. به صورت ریاضی، تابع مزیت به صورت زیر تعریف می‌شود:

$$A^\pi(s,a) = Q^\pi(s,a) - V^\pi(s)$$

که در آن $A^\pi(s,a)$ تابع مزیت برای عمل a در حالت s است. $Q^\pi(s,a)$ تابع ارزش عمل a در حالت s تحت سیاست π است و $V^\pi(s)$ تابع ارزش حالت s تحت سیاست π است. این تابع مزیت نشان می‌دهد که انجام عمل a در حالت s نسبت به میانگین اعمال تحت سیاست π چقدر مزیت دارد. اگر $A^\pi(s,a)$ مثبت باشد، نشان‌دهنده این است که عمل a بهتر از میانگین اعمال است و اگر منفی باشد، نشان‌دهنده کمتربودن عملکرد آن نسبت به میانگین است.

[27] Advantage Function

۳-۲ عامل گرادیان سیاست عمیق قطعی

گرادیان سیاست عمیق قطعی[۲۸] الگوریتمی است که همزمان یک تابع Q و یک سیاست را یاد می‌گیرد. این الگوریتم برای یادگیری تابع Q از داده‌های غیرسیاست محور[۲۹] و معادله بلمن استفاده می‌کند. این الگوریتم برای یادگیری سیاست نیز از تابع Q استفاده می‌کند.

این رویکرد وابستگی نزدیکی به یادگیری Q دارد. اگر تابع ارزش–عمل بهینه مشخص باشد، در هر حالت داده‌شده عمل بهینه را می‌توان با حل معادله (۳-۱۷) به دست آورد.

$$a^*(s) = \arg\max_a Q^*(s,a) \qquad (۳\text{-}۱۷)$$

الگوریتم DDPG ترکیبی از یادگیری تقریبی برای $Q^*(s,a)$ و یادگیری تقریبی برای $a^*(s)$ است و به‌صورتی طراحی شده‌است که برای محیط‌هایی با فضاهای عمل پیوسته مناسب باشد. آنچه این الگوریتم را برای فضای عمل پیوسته مناسب می‌کند، روش محاسبه $a^*(s)$ است. فرض می‌شود که تابع $Q^*(s,a)$ نسبت به آرگومان عمل مشتق‌پذیر است. مشتق‌پذیری این امکان را می‌دهد که یک روش یادگیری مبتنی بر گرادیان برای سیاست $\mu(s)$ استفاده شود. سپس، به‌جای اجرای یک بهینه‌سازی زمان‌بر در هر بار محاسبه $\max_a Q(s,a)$، می‌توان آن را با رابطه $\max_a Q(s,a) \approx Q(s,\mu(s))$ تقریب زد.

۳-۲-۱ یادگیری Q در DDPG

معادله بلمن که تابع ارزش عمل بهینه $(Q^*(s,a))$ را توصیف می‌کند، در پایین آورده شده‌است.

$$Q^*(s,a) = r(s,a) + \underset{s'\sim P}{\mathrm{E}}\left[\gamma \max_{a'} Q^*(s',a')\right] \qquad (۳\text{-}۱۸)$$

عبارت $s' \sim P$ به این معنی است که وضعیت بعدی یعنی s' از توزیع احتمال $P(\cdot|s,a)$ نمونه گرفته می‌شود. در معادله بلمن نقطه شروع برای یادگیری $Q^*(s,a)$ یک مقداردهی تقریبی است. پارامترهای شبکه عصبی $Q_\phi(s,a)$ با علامت ϕ نشان داده شده‌است. مجموعه $\mathcal{D}$ شامل اطلاعات جمع‌آوری شده تغییر از یک حالت به حالت دیگر (s,a,r,s',d) (که d نشان می‌دهد که آیا وضعیت s' پایانی است یا خیر) است. در بهینه‌سازی از تابع خطای میانگین مربعات بلمن[۳۰] (MSBE) استفاده شده‌است که معیاری برای نزدیکی Q_ϕ به حالت بهینه برای برآورده کردن معادله بلمن است.

[28] Deep Deterministic Policy Gradient (DDPG)
[29] Off-Policy
[30] Mean Squared Bellman Error

$$L(\phi,\mathcal{D}) = \underset{(s,a,r,s',d)\sim\mathcal{D}}{\mathrm{E}}\left[\left(Q_\phi(s,a) - \left(r + \gamma(1-d)\max_{a'} Q_\phi(s',a')\right)\right)^2\right] \qquad (۳-۱۹)$$

در الگوریتم DDPG دو ترفند برای عمکرد بهتر استفاده شده‌است که در ادامه به بررسی آن پرداخته شده‌است.

- بافرهای تکرار بازی

الگوریتم‌های یادگیری تقویتی جهت آموزش یک شبکه عصبی عمیق برای تقریب $Q^*(s,a)$ از بافرهای تکرار بازی[۳۱] تجربه‌شده استفاده می‌کنند. این مجموعه $\mathcal{D}$ شامل تجربیات قبلی عامل است. برای داشتن رفتار پایدار در الگوریتم، بافر تکرار بازی باید به اندازه کافی بزرگ باشد تا شامل یک دامنه گسترده از تجربیات شود. انتخاب داده‌های بافر به دقت انجام شده‌است چرا که اگر فقط از داده‌های بسیار جدید استفاده شود، بیش‌برازش[۳۲] رخ می‌دهید و اگر از تجربه بیش از حد استفاده شود، ممکن است فرآیند یادگیری کند شود.

- شبکه‌های هدف

الگوریتم‌های یادگیری Q از شبکه‌های هدف استفاده می‌کنند. اصطلاح زیر به‌عنوان هدف شناخته می‌شود.

$$r + \gamma(1-d)\max_{a'} Q_\phi(s',a') \qquad (۳-۲۰)$$

در هنگام کمینه کردن تابع خطای میانگین مربعات بلمن، سعی شده‌است تا تابع Q شبیه‌تر به هدف یعنی رابطه (۳-۲۰) شود. اما مشکل این است که هدف بستگی به پارامترهای در حال آموزش ϕ دارد. این باعث ایجاد ناپایداری در کمینه کردن تابع خطای میانگین مربعات بلمن می‌شود. راه حل آن استفاده از یک مجموعه پارامترهایی است که با تأخیر زمانی به ϕ نزدیک می‌شوند. به عبارت دیگر، یک شبکه دوم ایجاد می‌شود که به آن شبکه هدف گفته می‌شود. شبکه هدف پارامترهای شبکه اول را با تاخیر دنبال می‌کند. پارامترهای شبکه هدف با نشان ϕ_{targ} نشان داده می‌شوند. در الگوریتم DDPG، شبکه هدف در هر به‌روزرسانی شبکه اصلی، با میانگین‌گیری پولیاک[۳۳] به‌صورت زیر به‌روزرسانی می‌شود.

$$\phi_{\text{targ}} \leftarrow \rho\phi_{\text{targ}} + (1-\rho)\phi \qquad (۳-۲۱)$$

در رابطه بالا ρ یک ابرپارامتر[۳۴] است که بین صفر و یک انتخاب می‌شود. در این پژوهش این مقدار نزدیک به یک درنظر گرفته شده‌است.

[31] Replay Buffers
[32] Overfit
[33] Polyak Averaging
[34] Hyperparameter

الگوریتم DDPG نیاز به یک شبکه سیاست هدف ($\mu_{\theta_{\text{targ}}}$) برای محاسبه عمل‌هایی که به‌طور تقریبی بیشینه $Q_{\phi_{\text{targ}}}$ را حاصل کند، را دارد. برای رسیدن به این شبکه سیاست هدف از همان روشی که تابع Q به دست می‌آید یعنی با میانگین‌گیری پولیاک از پارامترهای سیاست در طول زمان آموزش استفاده می‌شود.

با درنظرگرفتن موارد اشاره‌شده، یادگیری Q در DDPG با کمینه‌کردن تابع خطای میانگین مربعات بلمن (MSBE) یعنی معادله (۳-۲۲) با استفاده از کاهش گرادیان تصادفی[۳۵] انجام می‌شود.

$$L(\phi, \mathcal{D}) = \underset{(s,a,r,s',d)\sim\mathcal{D}}{\mathrm{E}}\left[\left(Q_\phi(s,a) - \left(r + \gamma(1-d)Q_{\phi_{\text{targ}}}(s', \mu_{\theta_{\text{targ}}}(s'))\right)\right)^2\right] \qquad (۳-۲۲)$$

۳-۲-۲ سیاست در DDPG

در این بخش یک سیاست تعیین‌شده $\mu_\theta(s)$ یاد گرفته می‌شود تا عملی را انجام می‌دهد که بیشینه $Q_\phi(s,a)$ رخ دهد. از آنجا که فضای عمل پیوسته است و فرض شده‌است که تابع Q نسبت به عمل مشتق‌پذیر است، رابطه زیر با استفاده از صعود گرادیان[۳۶] (تنها نسبت به پارامترهای سیاست) بیشینه می‌شود.

$$\max_\theta \underset{s\sim\mathcal{D}}{\mathrm{E}}\left[Q_\phi(s, \mu_\theta(s))\right] \qquad (۳-۲۳)$$

۳-۲-۳ اکتشاف و بهره‌برداری در DDPG

برای بهبود اکتشاف[۳۷] در سیاست‌های DDPG، در زمان آموزش نویز به عمل‌ها اضافه می‌شود. نویسندگان مقاله DDPG [۲۶] توصیه کرده‌اند که نویز OU[۳۸] با همبندی زمانی[۳۹] اضافه شود. در زمان بهره‌برداری[۴۰] سیاست، از آنچه یاد گرفته است، نویز به عمل‌ها اضافه نمی‌شود.

۳-۲-۴ شبه‌کد DDPG

در این بخش، شبه‌کد الگوریتم DDPG پیاده‌سازی‌شده آورده شده‌است. در این پژوهش الگوریتم ۱ در محیط پایتون با استفاده از کتابخانه TensorFlow [۲۷] پیاده‌سازی شده‌است.

[35] Stochastic Gradient Descent
[36] Gradient Ascent
[37] Exploration
[38] Ornstein–Uhlenbeck
[39] Time-Correlated
[40] Exploitation

الگوریتم ۱ گرادیان سیاست عمیق قطعی

ورودی: پارامترهای اولیه سیاست (θ)، پارامترهای تابع Q (ϕ)، بافر تکرار بازی خالی ($\mathcal{D}$)

پارامترهای هدف را برابر با پارامترهای اصلی قرار دهید $\theta_{\text{targ}} \leftarrow \theta$، $\phi_{\text{targ}} \leftarrow \phi$

تا وقتی همگرایی رخ دهد:

وضعیت s را مشاهده کرده و عمل $a = \text{clip}(\mu_\theta(s) + \epsilon, a_{\text{Low}}, a_{\text{High}})$ را انتخاب کنید به‌طوری که $\epsilon \sim \mathcal{N}$ است.

عمل a را در محیط اجرا کنید.

وضعیت بعدی s'، پاداش r و سیگنال پایان d را مشاهده کنید تا نشان دهد آیا s' پایانی است یا خیر.

اگر s' پایانی است، وضعیت محیط را بازنشانی کنید.

اگر زمان به‌روزرسانی فرا رسیده است:

به ازای هر تعداد به‌روزرسانی:

یک دسته تصادفی گذر از یک حالت به حالت دیگر، $B = \{(s, a, r, s', d)\}$، از $\mathcal{D}$ نمونه‌گیری شود.

هدف را محاسبه کنید:

$$y(r, s', d) = r + \gamma(1 - d) Q_{\phi_{\text{targ}}}(s', \mu_{\theta_{\text{targ}}}(s'))$$

تابع Q را با یک مرحله از نزول گرادیان با استفاده از رابطه زیر به‌روزرسانی کنید:

$$\nabla_\phi \frac{1}{|B|} \sum_{(s,a,r,s',d) \in B} \left(Q_\phi(s, a) - y(r, s', d)\right)^2$$

سیاست را با یک مرحله از صعود گرادیان با استفاده از رابطه زیر به‌روزرسانی کنید:

$$\nabla_\theta \frac{1}{|B|} \sum_{s \in B} Q_\phi(s, \mu_\theta(s))$$

شبکه‌های هدف را با استفاده از معادلات زیر به‌روزرسانی کنید:

$$\phi_{\text{targ}} \leftarrow \rho \phi_{\text{targ}} + (1 - \rho)\phi$$

$$\theta_{\text{targ}} \leftarrow \rho \theta_{\text{targ}} + (1 - \rho)\theta$$

۳-۳ عامل گرادیان سیاست عمیق قطعی تاخیری دوگانه

عامل گرادیان سیاست عمیق قطعی تاخیری دوگانه[41] یکی از الگوریتم‌های یادگیری تقویتی است که برای حل مسائل کنترل در محیط‌های پیوسته طراحی شده‌است. این الگوریتم بر اساس الگوریتم DDPG توسعه یافته و با استفاده از تکنیک‌های مختلف، پایداری و کارایی یادگیری را بهبود می‌بخشد. در حالی که DDPG گاهی اوقات می‌تواند عملکرد بسیار خوبی داشته باشد، اما اغلب نسبت به ابرپارامترها و سایر انواع تنظیمات یادگیری حساس است. یک حالت رایج شکستِ عامل DDPG در یادگیری این است که تابع Q یادگرفته‌شده شروع به بیش‌برآورد مقادیر Q می‌کند که منجر به واگرایی سیاست می‌شود. واگرایی به این دلیل رخ می‌دهد که در فرآیند یادگیری سیاست از تخمین تابع Q استفاده می‌شود که افزایش خطای تابع Q منجر به ناپایداری در یادگیری سیاست می‌شود.

الگوریتم (Twin Delayed DDPG) TD3 از دو ترفند زیر جهت بهبود مشکلات اشاره‌شده استفاده می‌کند.

- یادگیری دوگانه‌ی محدودشده[42]: الگوریتم TD3 به جای یک تابع Q، دو تابع Q_{ϕ_1} و Q_{ϕ_2} را یاد می‌گیرد (از این رو دوگانه[43] نامیده می‌شود) و از کوچک‌ترین مقدار این دو Q_{ϕ_1} و Q_{ϕ_2} در تابع بلمن استفاده می‌شود. نحوه محاسبه هدف بر اساس دو تابع Q اشاره‌شده در رابطه (۳-۲۴) آورده شده‌است.

$$y(r, s', d) = r + \gamma(1-d) \min_{i=1,2} Q_{\phi_{i,\text{targ}}}(s', a'(s')) \tag{۳-۲۴}$$

سپس، در هر دو تابع Q_{ϕ_1} و Q_{ϕ_2} یادگیری انجام می‌شود.

$$L(\phi_1, \mathcal{D}) = \underset{(s,a,r,s',d)\sim\mathcal{D}}{\mathrm{E}}\left(Q_{\phi_1}(s,a) - y(r,s',d)\right)^2 \tag{۳-۲۵}$$

$$L(\phi_2, \mathcal{D}) = \underset{(s,a,r,s',d)\sim\mathcal{D}}{\mathrm{E}}\left(Q_{\phi_2}(s,a) - y(r,s',d)\right)^2 \tag{۳-۲۶}$$

- به‌روزرسانی‌های تاخیری سیاست[44]: الگوریتم TD3 سیاست را با تاخیر بیشتری نسبت به تابع Q به‌روزرسانی می‌کند. در مرجع [۲۸] توصیه شده‌است که برای هر دو به‌روزرسانی تابع Q، یک به‌روزرسانی سیاست انجام شود.

[41] Twin Delayed Deep Deterministic Policy Gradient (TD3)
[42] Clipped Double-Q Learning
[43] twin
[44] Delayed Policy Updates

این دو ترفند منجر به بهبود قابل توجه عملکرد TD3 نسبت به DDPG پایه می‌شوند. در نهایت سیاست با بیشینه‌کردن Q_{ϕ_1} آموخته می‌شود:

$$\max_{\theta} \mathop{\mathrm{E}}_{s\sim\mathcal{D}} \left[Q_{\phi_1}(s, \mu_\theta(s))\right] \tag{۲۷-۳}$$

۳-۳-۱ اکتشاف و بهره‌برداری در TD3

الگوریتم TD3 یک سیاست قطعی را به‌صورت غیرسیاست‌محور آموزش می‌دهد. از آنجایی که سیاست قطعی است، در ابتدا عامل تنوع کافی از اعمال را برای یافتن روش‌های مفید امتحان نمی‌کند. برای بهبود اکتشاف سیاست‌های TD3، در زمان آموزش نویز به عمل‌ها اضافه می‌شود. در این پژوهش، نویز گاوسی با میانگین صفر بدون هم‌بندی زمانی اعمال شده‌است. شدت نویز جهت بهره‌برداری بهتر در طول زمان کاهش می‌یابد.

۳-۳-۲ شبه‌کد TD3

در این بخش الگوریتم TD3 پیاده‌سازی شده آورده شده‌است. در این پژوهش الگوریتم ۴ در محیط پایتون با استفاده از کتابخانه PyTorch [۲۹] پیاده‌سازی شده‌است.

الگوریتم ۲ عامل گرادیان سیاست عمیق قطعی تاخیری دوگانه

ورودی: پارامترهای اولیه سیاست (θ)، پارامترهای تابع Q (ϕ_1, ϕ_2)، بافر بازی خالی ($\mathcal{D}$)

پارامترهای هدف را برابر با پارامترهای اصلی قرار دهید $\theta_{\text{targ}} \leftarrow \theta$، $\phi_{\text{targ},1} \leftarrow \phi_1$، $\phi_{\text{targ},2} \leftarrow \phi_2$

تا وقتی همگرایی رخ دهد:

وضعیت (s) را مشاهده کرده و عمل $a = \text{clip}(\mu_\theta(s) + \epsilon, a_{\text{Low}}, a_{\text{High}})$ را انتخاب کنید، به‌طوری که $\epsilon \sim \mathcal{N}$ است.

عمل a را در محیط اجرا کنید.

وضعیت بعدی s'، پاداش r و سیگنال پایان d را مشاهده کنید تا نشان دهد آیا s' پایانی است یا خیر.

اگر s' پایانی است، وضعیت محیط را بازنشانی کنید.

اگر زمان به‌روزرسانی فرا رسیده است:

به ازای j در هر تعداد به‌روزرسانی:

یک دسته تصادفی گذر از یک حالت به حالت دیگر، $B = \{(s, a, r, s', d)\}$، از $\mathcal{D}$ نمونه‌گیری شود.

هدف را محاسبه کنید:

$$y(r, s', d) = r + \gamma(1 - d) \min_{i=1,2} Q_{\phi_{\text{targ},i}}(s', a'(s'))$$

تابع Q را با یک مرحله از نزول گرادیان با استفاده از رابطه زیر به‌روزرسانی کنید:

$$\nabla_{\phi_i} \frac{1}{|B|} \sum_{(s,a,r,s',d) \in B} \left(Q_{\phi_i}(s, a) - y(r, s', d)\right)^2 \quad \text{for } i = 1, 2$$

اگر باقیمانده j بر تاخیر سیاست برابر 0 باشد :

سیاست را با یک مرحله از صعود گرادیان با استفاده از رابطه زیر به‌روزرسانی کنید:

$$\nabla_\theta \frac{1}{|B|} \sum_{s \in B} Q_{\phi_1}(s, \mu_\theta(s))$$

شبکه‌های هدف را با استفاده از معادلات زیر به‌روزرسانی کنید:

$$\phi_{\text{targ},i} \leftarrow \rho\phi_{\text{targ},i} + (1 - \rho)\phi_i \quad \text{for } i = 1, 2$$

$$\theta_{\text{targ}} \leftarrow \rho\theta_{\text{targ}} + (1 - \rho)\theta$$

۳-۴ عامل عملگر نقاد نرم

عملگرد نقاد نرم[۴۵] الگوریتمی است که یک سیاست تصادفی را به‌صورت غیرسیاست‌محور بهینه می‌کند و پلی بین بهینه‌سازی سیاست تصادفی و رویکردهای غیرسیاست‌محور مانند DDPG ایجاد می‌کند. این الگوریتم جانشین مستقیم TD3 نیست (زیرا تقریباً همزمان منتشر شده‌است)؛ اما، ترفند یادگیری دوگانه محدودشده را در خود جای داده است و به‌دلیل سیاست تصادفی SAC، از روشی به نام صاف‌کردن سیاست هدف[۴۶] استفاده شده‌است. یکی از ویژگی های اصلی SAC، تنظیم آنتروپی است. آنتروپی معیاری از تصادفی‌بودن انتخاب عمل در سیاست است. آموزش سیاست در جهت تعادل بهینه بین آنتروپی و بیشنه‌سازی بازده مورد انتظار است. این شرایط ارتباط نزدیکی با تعادل اکتشاف–بهره‌برداری دارد. افزایش آنتروپی منجر به اکتشاف بیشتر می‌شود که می‌تواند یادگیری را در مراحل بعدی تسریع کند. همچنین، می‌تواند از همگرایی زودهنگام سیاست به یک بهینه محلی بد جلوگیری کند. برای توضیح SAC، ابتدا باید به بررسی یادگیری تقویتی تنظیم‌شده با آنتروپی[۴۷] پرداخته شود. در RL تنظیم‌شده با آنتروپی، روابط تابع ارزش کمی متفاوت است.

۳-۴-۱ یادگیری تقویتی تنظیم‌شده با آنتروپی

آنتروپی معیاری برای سنجش میزان عدم قطعیت یا تصادفی بودن یک متغیر تصادفی یا توزیع احتمال آن است. به عبارت دقیق‌تر، آنتروپی برای یک توزیع احتمال، میانگین اطلاعات حاصل از نمونه‌برداری از آن توزیع را اندازه‌گیری می‌کند. در زمینه یادگیری تقویتی، تنظیم با آنتروپی تکنیکی است که با افزودن یک ترم متناسب با آنتروپی سیاست به تابع هدف، عامل را تشویق به اکتشاف بیشتر و اتخاذ سیاست‌های تصادفی‌تر می‌کند. این امر می‌تواند به بهبود پایداری فرآیند یادگیری و جلوگیری از همگرایی زودهنگام به بهینه‌های محلی کمک کند.

فرض کنید X یک متغیر تصادفی پیوسته با تابع چگالی احتمال $p(x)$ باشد. آنتروپی $H(X)$ این متغیر تصادفی به صورت امید ریاضی لگاریتم منفی چگالی احتمال آن تعریف می‌شود:

$$H(X) = \mathrm{E}_{x\sim p}\left[-\log p(x)\right] \tag{۳-۲۸}$$

۳-۴-۲ سیاست در SAC

در یادگیری تقویتی تنظیم‌شده با آنتروپی، عامل در هر مرحله زمانی متناسب با آنتروپی سیاست در آن مرحله زمانی پاداش دریافت می‌کند. بر اساس توضیحات اشاره شده روابط یادگیری تقویتی به‌صورت زیر می‌شود.

[45] Soft Actor Critic (SAC)
[46] Target Policy Smoothing
[47] Entropy-Regularized Reinforcement Learning

$$\pi^* = \arg\max_{\pi} \mathop{\mathrm{E}}_{\tau\sim\pi} \sum_{t=0}^{\infty} \gamma^t \Big(R(s_t, a_t, s_{t+1}) + \alpha H\left(\pi(\cdot|s_t)\right) \Big) \tag{۲۹-۳}$$

که در آن ($\alpha > 0$) ضریب مبادله[۴۸] است.

۳-۴-۳ تابع ارزش در SAC

اکنون می‌توان تابع ارزش کمی متفاوت را بر اساس این مفهوم تعریف کرد. V^π به گونه‌ای تغییر می‌کند که پاداش‌های آنتروپی را از هر مرحله زمانی شامل می‌شود.

$$V^\pi(s) = \mathop{\mathrm{E}}_{\tau\sim\pi} \left[\sum_{t=0}^{\infty} \gamma^t \Big(R(s_t, a_t, s_{t+1}) + \alpha H\left(\pi(\cdot|s_t)\right) \Big) \Bigg| s_0 = s \right] \tag{۳۰-۳}$$

۴-۴-۳ تابع Q در SAC

تابع Q^π به گونه ای تغییر می‌کند که پاداش های آنتروپی را از هر مرحله زمانی به جز مرحله اول شامل می‌شود.

$$Q^\pi(s,a) = \mathop{\mathrm{E}}_{\tau\sim\pi} \left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t, s_{t+1}) + \alpha \sum_{t=1}^{\infty} \gamma^t H\left(\pi(\cdot|s_t)\right) \Bigg| s_0 = s, a_0 = a \right] \tag{۳۱-۳}$$

با این تعاریف رابطه V^{π} و Q^π به‌صورت زیر است.

$$V^\pi(s) = \mathop{\mathrm{E}}_{a\sim\pi} \left[Q^\pi(s,a)\right] + \alpha H\left(\pi(\cdot|s)\right) \tag{۳۲-۳}$$

۵-۴-۳ معادله بلمن در SAC

معادله بلمن در حالت تنظیم‌شده با آنتروپی به‌صورت زیر ارائه می‌شود.

$$Q^\pi(s,a) = \mathop{\mathrm{E}}_{\substack{s'\sim P \\ a'\sim\pi}} \left[R(s,a,s') + \gamma\left(Q^\pi(s',a') + \alpha H\left(\pi(\cdot|s')\right)\right)\right] \tag{۳۳-۳}$$

$$= \mathop{\mathrm{E}}_{s'\sim P} \left[R(s,a,s') + \gamma V^\pi(s')\right] \tag{۳۴-۳}$$

[48] Trade-Off

۳-۴-۶ یادگیری Q

با درنظرگرفتن موارد اشاره‌شده، یادگیری Q در SAC با کمینه‌کردن تابع خطای میانگین مربعات بلمن (MSBE) یعنی معادله (۳-۳۵) با استفاده از کاهش گرادیان انجام می‌شود.

$$L(\phi_i, \mathcal{D}) = \underset{(s,a,r,s',d)\sim\mathcal{D}}{\mathrm{E}}\left[\left(Q_{\phi_i}(s,a) - y(r,s',d)\right)^2\right] \tag{۳-۳۵}$$

در معادله (۳-۳۵) تابع هدف برای روش یادگیری تقویتی SAC به صورت زیر تعریف می‌شود.

$$y(r,s',d) = r + \gamma(1-d)\left(\min_{j=1,2} Q_{\phi_{\mathrm{targ},j}}(s',\tilde{a}') - \alpha \log \pi_\theta(\tilde{a}'|s')\right), \qquad \tilde{a}' \sim \pi_\theta(\cdot|s') \tag{۳-۳۶}$$

نماد عمل بعدی را به جای a' به $\tilde{a}'$ تغییر داده شده تا مشخص شود که عمل‌های بعدی باید آخرین سیاست نمونه‌برداری شوند در حالی که r و s باید از بافر تکرار بازی آمده باشند.

۳-۴-۷ سیاست در SAC

سیاست باید در هر وضعیت برای به حداکثر رساندن بازگشت مورد انتظار آینده به همراه آنتروپی مورد انتظار آینده عمل کند. یعنی باید $V^\pi(s)$ را به حداکثر برساند، بسط تابع ارزش در ادامه آمده است.

$$V^\pi(s) = \underset{a\sim\pi}{\mathrm{E}}\left[Q^\pi(s,a)\right] + \alpha H\left(\pi(\cdot|s)\right) \tag{۳-۳۷}$$

$$= \underset{a\sim\pi}{\mathrm{E}}\left[Q^\pi(s,a) - \alpha \log \pi(a|s)\right] \tag{۳-۳۸}$$

روش بهینه‌سازی سیاست از ترفند پارامترسازی مجدد[۴۹] استفاده می‌کند، که در آن نمونه ای از $\pi_\theta(\cdot|s)$ با محاسبه یک تابع قطعی از وضعیت، پارامترهای سیاست و نویز مستقل استخراج می‌شود. در این پژوهش مانند نویسندگان مقاله SAC [۳۰]، از یک سیاست گاوسی فشرده[۵۰] استفاده شده‌است. بر اساس این روش نمونه‌ها مطابق با رابطه زیر بدست می‌آیند:

$$\tilde{a}_\theta(s,\xi) = \tanh\left(\mu_\theta(s) + \sigma_\theta(s) \odot \xi\right), \qquad \xi \sim \mathcal{N}(0, I) \tag{۳-۳۹}$$

تابع tanh در سیاست SAC تضمین می‌کند که اعمال در یک محدوده متناهی محدود شوند. این مورد در سیاست‌های VPG، TRPO و PPO وجود ندارد. همچنین اعمال این تابع توزیع را از حالت گاوسی تغییر می‌دهد.

[49] Reparameterization
[50] Squashed Gaussian Policy

در الگوریتم SAC با استفاده از ترفند پارامتری‌سازی مجدد، عمل‌ها از یک توزیع نرمال به‌وسیله نویز تصادفی تولید شده و به این ترتیب امکان محاسبه مشتق‌ها به‌طور مستقیم از طریق تابع توزیع فراهم می‌شود، که باعث ثبات و کارایی بیشتر در آموزش می‌شود. اما در حالت بدون پارامتری‌سازی مجدد، عمل‌ها مستقیماً از توزیع سیاست نمونه‌برداری می‌شوند و محاسبه گرادیان نیازمند استفاده از ترفند نسبت احتمال[51] است که معمولاً باعث افزایش واریانس و ناپایداری در آموزش می‌شود.

$$\underset{a\sim\pi_\theta}{\mathrm{E}}\left[Q^{\pi_\theta}(s,a)-\alpha\log\pi_\theta(a|s)\right]=\underset{\xi\sim\mathcal{N}}{\mathrm{E}}\left[Q^{\pi_\theta}(s,\tilde{a}_\theta(s,\xi))-\alpha\log\pi_\theta(\tilde{a}_\theta(s,\xi)|s)\right] \quad (۴۰-۳)$$

برای به‌دست آوردن تابع هزینه سیاست، گام نهایی این است که باید Q^{π_θ} را با یکی از تخمین‌زننده‌های تابع خود جایگزین کنیم. برخلاف TD3 که از Q_{ϕ_1} (فقط اولین تخمین‌زننده Q) استفاده می‌کند، SAC از $\min_{j=1,2} Q_{\phi_j}$ (کمینه‌ی دو تخمین‌زننده Q) استفاده می‌کند. بنابراین، سیاست طبق رابطه زیر بهینه‌سازی می‌شود:

$$\max_{\theta}\underset{\substack{s\sim\mathcal{D}\\ \xi\sim\mathcal{N}}}{\mathrm{E}}\left[\min_{j=1,2}Q_{\phi_j}(s,\tilde{a}_\theta(s,\xi))-\alpha\log\pi_\theta(\tilde{a}_\theta(s,\xi)|s)\right] \quad (۴۱-۳)$$

که تقریباً مشابه بهینه‌سازی سیاست در DDPG و TD3 است، به جز ترفند min-double-Q، تصادفی بودن و عبارت آنتروپی.

۳-۴-۸ اکتشاف و بهره‌برداری در SAC

الگوریتم SAC یک سیاست تصادفی با تنظیم‌سازی آنتروپی آموزش می‌دهد و به صورت سیاست محور به اکتشاف می‌پردازد. ضریب تنظیم آنتروپی α به طور صریح تعادل بین اکتشاف و بهره‌برداری را کنترل می‌کند، به‌طوری‌که مقادیر بالاتر α به اکتشاف بیشتر و مقادیر پایین‌تر α به بهره‌برداری بیشتر منجر می‌شود. مقدار بهینه α (که به یادگیری پایدارتر و پاداش بالاتر منجر می‌شود) ممکن است در محیط‌های مختلف متفاوت باشد و نیاز به تنظیم دقیق داشته باشد. در زمان آزمایش، برای ارزیابی میزان بهره‌برداری سیاست از آنچه یاد گرفته است، تصادفی بودن را حذف کرده و از عمل میانگین به جای نمونه‌برداری از توزیع استفاده می‌کنیم. این روش معمولاً عملکرد را نسبت به سیاست تصادفی بهبود می‌بخشد.

[51] Likelihood Ratio Trick

۳-۴-۹ شبه‌کد SAC

در این بخش الگوریتم SAC پیاده‌سازی شده آورده شده‌است. در این پژوهش الگوریتم ۳ در محیط پایتون با استفاده از کتابخانه PyTorch [۲۹] پیاده‌سازی شده است.

الگوریتم ۳ عامل عملگرد نقاد نرم

ورودی: پارامترهای اولیه سیاست (θ)، پارامترهای تابع Q (ϕ_1, ϕ_2)، بافر بازی خالی $(\mathcal{D})$

پارامترهای هدف را برابر با پارامترهای اصلی قرار دهید $\theta_{\text{targ}} \leftarrow \theta$، $\phi_{\text{targ},1} \leftarrow \phi_1$، $\phi_{\text{targ},2} \leftarrow \phi_2$

تا وقتی همگرایی رخ دهد:

وضعیت (s) را مشاهده کرده و عمل $a \sim \pi_\theta(\cdot|s)$ را انتخاب کنید.

عمل a را در محیط اجرا کنید.

وضعیت بعدی s'، پاداش r و سیگنال پایان d را مشاهده کنید تا نشان دهد آیا s' پایانی است یا خیر.

اگر s' پایانی است، وضعیت محیط را بازنشانی کنید.

اگر زمان به‌روزرسانی فرا رسیده است:

به ازای j در هر تعداد به‌روزرسانی:

یک دسته تصادفی گذر از یک حالت به حالت دیگر، $B = \{(s, a, r, s', d)\}$، از $\mathcal{D}$ نمونه‌گیری شود.

هدف را محاسبه کنید:

$$y(r, s', d) = r + \gamma(1 - d) \left(\min_{i=1,2} Q_{\phi_{\text{targ},i}}(s', \tilde{a}') - \alpha \log \pi_\theta(\tilde{a}'|s') \right), \qquad \tilde{a}' \sim \pi_\theta(\cdot|s')$$

تابع Q را با یک مرحله از نزول گرادیان با استفاده از رابطه زیر به‌روزرسانی کنید:

$$\nabla_{\phi_i} \frac{1}{|B|} \sum_{(s,a,r,s',d) \in B} \left(Q_{\phi_i}(s, a) - y(r, s', d)\right)^2 \quad \text{for } i = 1, 2$$

سیاست را با یک مرحله از صعود گرادیان با استفاده از رابطه زیر به‌روزرسانی کنید:

$$\nabla_\theta \frac{1}{|B|} \sum_{s \in B} \left(\min_{i=1,2} Q_{\phi_i}(s, \tilde{a}_\theta(s)) - \alpha \log \pi_\theta \left(\tilde{a}_\theta(s) | \, s \right) \right)$$

شبکه‌های هدف را با استفاده از معادلات زیر به‌روزرسانی کنید:

$$\phi_{\text{targ},i} \leftarrow \rho \phi_{\text{targ},i} + (1 - \rho) \phi_i \quad \text{for } i = 1, 2$$

۳-۵ عامل بهینه‌سازی سیاست مجاور

الگوریتم بهینه‌سازی سیاست مجاور[52] یک الگوریتم بهینه‌سازی سیاست مبتنی بر گرادیان است که برای حل مسائل کنترل مسئله‌های یادگیری تقویتی استفاده می‌شود. این الگوریتم از الگوریتم TRPO[53] الهام گرفته شده‌است و با اعمال تغییراتی بر روی آن، سرعت و کارایی آن را افزایش داده است. در این بخش به بررسی این الگوریتم و نحوه عملکرد آن می‌پردازیم. الگوریتم PPO همانند سایر الگوریتم‌های یادگیری تقویتی، به‌دنبال یافتن بهترین گام ممکن برای بهبود عملکرد سیاست با استفاده از داده‌های موجود است. این الگوریتم تلاش می‌کند تا از گام‌های بزرگ که می‌توانند منجر به افت ناگهانی عملکرد شوند، اجتناب کند. برخلاف روش‌های پیچیده‌تر مرتبه دوم مانند TRPO، PPO از مجموعه‌ای از روش‌های مرتبه اول ساده‌تر برای حفظ نزدیکی سیاست‌های جدید به سیاست‌های قبلی استفاده می‌کند. این سادگی در پیاده‌سازی، PPO را به روشی کارآمدتر تبدیل می‌کند، در حالی که از نظر تجربی نشان داده شده‌است که عملکردی حداقل به اندازه TRPO دارد. از جمله ویژگی‌های مهم این الگوریتم می‌توان به سیاست محور بودن آن اشاره کرد. این الگوریتم برای عامل‌های یادگیری تقویتی که سیاست‌های پیوسته و گسسته دارند، مناسب است.

الگوریتم PPO داری دو گونه اصلی PPO-Clip و PPO-Penalty است. در ادامه به بررسی هر یک از این دو گونه پرداخته شده‌است.

- روش **PPO-Penalty:** روش PPO-Penalty به‌دنبال حل تقریبی و به‌روزرسانی با محدودیت واگرایی کولباک–لیبلر[54] است، مشابه روشی که در الگوریتم TRPO استفاده شده‌است. با این حال، به جای اعمال یک محدودیت سخت[55]، PPO-Penalty واگرایی KL را در تابع هدف جریمه می‌کند. این جریمه به طور خودکار در طول آموزش تنظیم می‌شود تا از افت ناگهانی عملکرد جلوگیری کند.

- روش **PPO-Clip:** در این روش، هیچ عبارت واگرایی KL در تابع هدف وجود ندارد و هیچ محدودیتی اعمال نمی‌شود. در عوض، PPO-Clip از یک عملیات بریدن[56] خاص در تابع هدف استفاده می‌کند تا انگیزه سیاست جدید برای دور شدن از سیاست قبلی را از بین ببرد.

در این پژوهش از روش PPO-Clip برای آموزش عامل‌های یادگیری تقویتی استفاده شده‌است.

[52] Proximal Policy Optimization (PPO)
[53] Trust Region Policy Optimization
[54] Kullback-Leibler (KL) Divergence
[55] Hard Constraint
[56] Clipping

۳-۵-۱ سیاست در الگوریتم PPO

تابع سیاست در الگوریتم PPO به صورت یک شبکه عصبی پیاده‌سازی شده‌است. این شبکه عصبی ورودی‌های محیط را دریافت کرده و اقدامی را که باید عامل انجام دهد را تولید می‌کند. این شبکه عصبی می‌تواند شامل چندین لایه پنهان با توابع فعال‌سازی مختلف باشد. در این پژوهش از یک شبکه عصبی با سه لایه پنهان و تابع فعال‌سازی ReLu استفاده شده‌است. تابع سیاست در الگوریتم PPO به صورت زیر به‌روزرسانی می‌شود:

$$\theta_{k+1} = \arg\max_{\theta} \underset{s,a\sim\pi_{\theta_k}}{\mathrm{E}} \left[L(s,a,\theta_k,\theta)\right] \qquad (۳-۴۲)$$

در این پژوهش برای به حداکثر رساندن تابع هدف، چندین گام بهینه‌سازی گرادیان کاهشی تصادفی[۵۷] اجرا شده‌است. در معادله بالا L به‌صورت زیر تعریف شده‌است:

$$L(s,a,\theta_k,\theta) = \min\left(\frac{\pi_\theta(a|s)}{\pi_{\theta_k}(a|s)}A^{\pi_{\theta_k}}(s,a),\ \ \textbf{clip}\left(\frac{\pi_\theta(a|s)}{\pi_{\theta_k}(a|s)}, 1-\epsilon, 1+\epsilon\right)A^{\pi_{\theta_k}}(s,a)\right) \qquad (۳-۴۳)$$

که در آن ϵ یک فراپامتر است که مقدار آن معمولا کوچک است. این فراپامتر مشخص می‌کند که چقدر اندازه گام بهینه‌سازی باید محدود شود. در این پژوهش مقدار $\epsilon = 0.2$ انتخاب شده‌است. جهت سادگی در پیاده‌سازی معادله (۳-۴۳) به معادله تغیر داده شده‌است.

$$L(s,a,\theta_k,\theta) = \min\left(\frac{\pi_\theta(a|s)}{\pi_{\theta_k}(a|s)}A^{\pi_{\theta_k}}(s,a),\ \ g(\epsilon, A^{\pi_{\theta_k}}(s,a))\right) \qquad (۳-۴۴)$$

که تابع g به صورت زیر تعریف شده‌است.

$$g(\epsilon, A) = \begin{cases} (1+\epsilon)A & A \geqslant 0 \\ (1-\epsilon)A & A < 0 \end{cases} \qquad (۳-۴۵)$$

در حالی که این نوع محدود کردن (PPO-Clip) تا حد زیادی به اطمینان از به‌روزرسانی‌های معقول سیاست کمک می‌کند، همچنان ممکن است با سیاست به‌دست آید که بیش از حد از سیاست قدیمی دور باشد. برای جلوگیری از این امر، پیاده‌سازی‌های مختلف PPO از مجموعه‌ای از ترفندها استفاده می‌کنند. در پیاده‌سازی این پژوهش، از روشی ساده به نام **توقف زودهنگام**[۵۸] استفاده شده‌است. اگر میانگین واگرایی کولباک–لیبلر (KL) خط‌مشی جدید از خط‌مشی قدیمی از یک آستانه فراتر رود، گام‌های گرادیان (بهینه‌سازی) را متوقف می‌شوند.

[57] Stochastic Gradient Descent (SGD)
[58] Early Stopping

۳-۵-۲ اکتشاف و بهره‌برداری در PPO

الگوریتم PPO از یک سیاست تصادفی به‌صورت سیاست‌محور برای آموزش استفاده می‌کند. این به این معنی است که اکتشاف محیط با نمونه‌گیری عمل‌ها بر اساس آخرین نسخه از این سیاست تصادفی انجام می‌شود. میزان تصادفی بودن انتخاب عمل به شرایط اولیه و فرآیند آموزش بستگی دارد.

در طول آموزش، سیاست به طور کلی به تدریج کمتر تصادفی می‌شود، زیرا قانون به‌روزرسانی آن را تشویق می‌کند تا از پاداش‌هایی که قبلاً پیدا کرده است، بهره‌برداری کند. البته این موضوع می‌تواند منجر به رسیدن سیاست به بهینه‌های محلی[59] شود.

۳-۵-۳ شبه‌کد PPO

در این بخش الگوریتم PPO پیاده‌سازی شده آورده شده‌است. در این پژوهش الگوریتم ۴ در محیط پایتون با استفاده از کتابخانه PyTorch [۲۹] پیاده‌سازی شده است.

[59]Local Optima

الگوریتم ۴ بهینه‌سازی سیاست مجاور (PPO-Clip)

ورودی: پارامترهای اولیه سیاست (θ_0)، پارامترهای تابع ارزش (ϕ_0)

۱: به ازای $k = 0, 1, 2, \ldots$:

۲: مجموعه‌ای از مسیرها به نام $\mathcal{D}_k = \{\tau_i\}$ با اجرای سیاست $\pi_k = \pi(\theta_k)$ در محیط جمع‌آوری شود.

۳: پاداش‌های باقی‌مانده ($\hat{R}_t$) محاسبه شود.

۴: برآوردهای مزیت را محاسبه کنید، $\hat{A}_t$ (با استفاده از هر روش تخمین مزیت) بر اساس تابع ارزش فعلی V_{ϕ_k}.

۵: سیاست را با به حداکثر رساندن تابع هدف PPO-Clip به‌روزرسانی کنید:

$$\theta_{k+1} = \arg\max_{\theta} \frac{1}{|\mathcal{D}_k| T} \sum_{\tau \in \mathcal{D}_k} \sum_{t=0}^{T} \min\left(\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_k}(a_t|s_t)} A^{\pi_{\theta_k}}(s_t, a_t), \quad g(\epsilon, A^{\pi_{\theta_k}}(s_t, a_t)) \right)$$

معمولاً از طریق گرادیان افزایشی تصادفی Adam.

۶: برازش تابع ارزش با رگرسیون بر روی میانگین مربعات خطا:

$$\phi_{k+1} = \arg\min_{\phi} \frac{1}{|\mathcal{D}_k| T} \sum_{\tau \in \mathcal{D}_k} \sum_{t=0}^{T} \left(V_\phi(s_t) - \hat{R}_t \right)^2$$

معمولاً از طریق برخی از الگوریتم‌های کاهشی گرادیان.

فصل ۴

یادگیری تقویتی چند عاملی

کاربردهای پیچیده در یادگیری تقویتی نیازمند اضافه کردن چندین عامل[1] برای انجام همزمان وظایف مختلف هستند. با این حال، افزایش تعداد عامل‌ها چالش‌هایی در مدیریت تعاملات میان آن‌ها به همراه دارد. در این فصل، بر اساس مسئله بهینه‌سازی برای هر عامل، مفهوم تعادل[2] معرفی شده تا رفتارهای توزیعی چندعاملی را تنظیم کند. رابطه رقابت میان عامل‌ها در سناریوهای مختلف تحلیل شده و آن‌ها با الگوریتم‌های معمول یادگیری تقویتی چندعاملی ترکیب شده‌اند. بر اساس انواع تعاملات، یک چارچوب نظریه بازی برای مدل‌سازی عمومی در سناریوهای چندعاملی استفاده شده است. با تحلیل بهینه‌سازی و وضعیت تعادل برای هر بخش از چارچوب، سیاست بهینه یادگیری تقویتی چندعاملی برای هر عامل بررسی شده است.

۴-۱ تعاریف و مفاهیم اساسی

یادگیری تقویتی چندعاملی[3] به بررسی چگونگی یادگیری و تصمیم‌گیری چندین عامل مستقل در یک محیط مشترک پرداخته می‌شود. برای تحلیل دقیق و درک بهتر این حوزه، اجزای اصلی آن شامل عامل، سیاست و مطلوبیت[4] در نظر گرفته می‌شوند که در ادامه به صورت مختصر و منسجم تشریح می‌گردند.

- عامل: یک موجودیت مستقل به عنوان عامل تعریف می‌شود که به صورت خودمختار با محیط تعامل کرده و بر اساس مشاهدات رفتار سایر عامل‌ها، سیاست‌هایش انتخاب می‌گردند تا سود حداکثر یا ضرر حداقل حاصل شود. در سناریوهای مورد بررسی، چندین عامل به صورت مستقل عمل می‌کنند؛ اما اگر

[1] Multi-Agent
[2] Equilibrium
[3] Multi-Agent Reinforcement Learning (MARL)
[4] Utility

تعداد عامل‌ها به یک کاهش یابد، MARL به یادگیری تقویتی معمولی تبدیل می‌شود.

- سیاست: برای هر عامل در MARL، سیاستی خاص در نظر گرفته می‌شود که به عنوان روشی برای انتخاب اقدامات بر اساس وضعیت محیط و رفتار سایر عامل‌ها تعریف می‌گردد. این سیاست‌ها با هدف به حداکثر رساندن سود و به حداقل رساندن هزینه طراحی شده و تحت تأثیر محیط و سیاست‌های دیگر عامل‌ها قرار می‌گیرند.

- مطلوبیت: مطلوبیت هر عامل بر اساس نیازها و وابستگی‌هایش به محیط و سایر عامل‌ها تعریف شده و به صورت سود منهای هزینه، با توجه به اهداف مختلف محاسبه می‌شود. در سناریوهای چندعاملی، از طریق یادگیری از محیط و تعامل با دیگران، مطلوبیت هر عامل بهینه می‌گردد.

در این چارچوب، برای هر عامل در MARL تابع مطلوبیت خاصی در نظر گرفته شده و بر اساس مشاهدات و تجربیات حاصل از تعاملات، یادگیری سیاست به صورت مستقل انجام می‌شود تا ارزش مطلوبیت به حداکثر برسد، بدون اینکه مستقیماً به مطلوبیت سایر عامل‌ها توجه شود. این فرآیند ممکن است به رقابت یا همکاری میان عامل‌ها منجر گردد. با توجه به پیچیدگی تعاملات میان چندین عامل، تحلیل نظریه بازی‌ها به عنوان ابزاری مؤثر برای تصمیم‌گیری در این حوزه به کار گرفته می‌شود. بسته به سناریوهای مختلف، این بازی‌ها در دسته‌بندی‌های متفاوتی قرار داده شده که در بخش‌های بعدی بررسی خواهند شد.

۴-۲ نظریه بازی‌ها

نظریه بازی‌ها شاخه‌ای از ریاضیات است که به مطالعه تصمیم‌گیری در موقعیت‌هایی می‌پردازد که نتیجه انتخاب‌های هر فرد به تصمیمات دیگران وابسته است. این نظریه چارچوبی برای تحلیل تعاملات میان بازیکنان ارائه می‌دهد و در حوزه‌های مختلفی مانند اقتصاد، علوم سیاسی، زیست‌شناسی و علوم کامپیوتر کاربرد دارد. در این فصل، دو مفهوم کلیدی نظریه بازی‌ها یعنی تعادل نش و بازی‌های مجموع صفر بررسی شده است.

۴-۲-۱ تعادل نش

تعادل نش[5] یکی از بنیادی‌ترین مفاهیم در نظریه بازی‌ها است که توسط جان نش در سال ۱۹۵۰ معرفی شد. این مفهوم به مجموعه‌ای از بازی‌ها اشاره دارد که در آن هیچ بازیکنی نمی‌تواند با تغییر یک‌جانبه سیاست خود، سود بیشتری به دست آورد، به شرطی که سیاست‌های سایر بازیکنان ثابت بماند.

[5] Nash Equilibrium

- تعریف تعادل نش: فرض کنید یک بازی با n بازیکن داریم. هر بازیکن i دارای مجموعه سیاست‌های Π_i و تابع مطلوبیت $u_i : \Pi_1 \times \Pi_2 \times \cdots \times \Pi_n \rightarrow \mathbb{R}$ است. یک مجموعه سیاست $\pi^* = (\pi_1^*, \pi_2^*, \ldots, \pi_n^*)$ تعادل نش نامیده می‌شود اگر برای هر بازیکن i و هر سیاست $\pi_i \in \Pi_i$ در وضعیت s داشته باشیم:

$$u_i(\pi_i^*, \pi_{-i}^*, s) \geqslant u_i(\pi_i, \pi_{-i}^*, s) \tag{۴-۱}$$

در اینجا، π_{-i}^* نشان‌دهنده سیاست‌های همه بازیکنان به جز بازیکن i است. در ادامه پژوهش جهت استفاده از چارچوب نظریه بازی در یادگیری تقویتی تابع مطلوبیت به‌گونه‌ای تعریف شده است که برابر با تابع ارزش $u_i(\pi_i, \pi_{-i}, s) = V_i^{\pi_i, \pi_{-i}}(s)$ باشد.

- اهمیت تعادل نش: تعادل نش نقطه‌ای را در بازی مشخص می‌کند که هر بازیکن بهترین پاسخ را نسبت به انتخاب‌های دیگران ارائه داده است. این مفهوم به‌ویژه در بازی‌های غیرهمکارانه، به‌عنوان پیش‌بینی رفتار منطقی بازیکنان استفاده می‌شود و در زمینه‌هایی مانند یادگیری تقویتی چند عامله کاربرد گسترده‌ای دارد.

۴-۲-۲ بازی مجموع صفر

بازی‌های مجموع صفر[۶] دسته‌ای از بازی‌ها هستند که در آن‌ها تابع ارزش یک بازیکن دقیقاً برابر با ضرر بازیکن دیگر است. به عبارت دیگر، مجموع ارزشهای همه بازیکنان در هر مرحله صفر است.

- تعریف بازی مجموع صفر: در یک بازی دو نفره، اگر تابع ارزش بازیکن اول ($V_1^{(\pi_1,\pi_2)}(s)$) و بازیکن دوم ($V_2^{(\pi_1,\pi_2)}(s)$) به‌گونه‌ای باشد که برای هر مجموعه سیاست (π_1, π_2) به صورت زیر باشد را یک بازی مجموع صفر نامیده می‌شود.

$$V_1^{(\pi_1,\pi_2)}(s) + V_2^{(\pi_1,\pi_2)}(s) = 0 \rightarrow V_1^{(\pi_1,\pi_2)}(s) = -V_2^{(\pi_1,\pi_2)}(s) \tag{۴-۲}$$

- سیاست بهینه در بازی مجموع صفر: در بازی‌های مجموع صفر، سیاست بهینه هر بازیکن، انتخابی است که تابع ارزش خود را در برابر بهترین پاسخ حریف به حداکثر برساند. این سیاست اغلب به تعادل نش منجر می‌شود. سیاست بهینه دو بازیکن در بازی مجموع صفر با تابع ارزش معادله (۴-۲) به صورت زیر است.

[6] Zero-Sum Games

$$V_1^*(s) = \max_{\pi_1} \min_{\pi_2} V_1^{(\pi_1,\pi_2)}(s) \tag{۴-۳}$$

$$V_2^*(s) = \max_{\pi_2} \min_{\pi_1} V_2^{(\pi_1,\pi_2)}(s) \tag{۴-۴}$$

فصل ۵

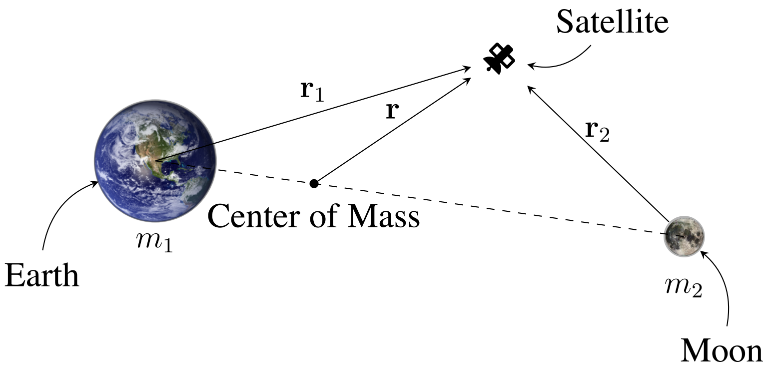
مدل‌سازی محیط یادگیری سه جسمی

مسیرهای فضایی سنتی تحت تأثیر گرانش یک جسم مرکزی (خورشید، زمین یا سیاره‌ای دیگر) شکل می‌گیرند و توسط اجسام سوم تحت تأثیر قرار می‌گیرند. تحلیل‌های مخروطی پیوسته نشان می‌دهند که چگونه می‌توان طراحی را از یک جسم مرکزی به جسم دیگر منتقل کرد، هنگامی که فضاپیما از حوزه تأثیر یک جسم عبور می‌کند. در برخی موارد، مأموریت فضاپیما آن را در ناحیه‌ای از فضا قرار می‌دهد که به‌طور همزمان تحت تأثیر دو جسم بزرگ است. این مسیرها نمی‌توانند از تحلیل دو جسم با اختلالات جسم سوم استفاده کنند، بلکه باید تأثیرات هر دو جسم به‌طور همزمان در نظر گرفته شوند. برای درک این مسئله، ابتدا به مطالعه مسئله عمومی سه‌جسمی خواهیم پرداخت و چگونگی اعمال آن به سیستم‌های واقعی را بررسی خواهیم کرد. سپس، برخی از مسیرها و تحلیل‌های جالبی که توسط فضاپیماهای مدرن استفاده می‌شوند، ارائه خواهیم داد.

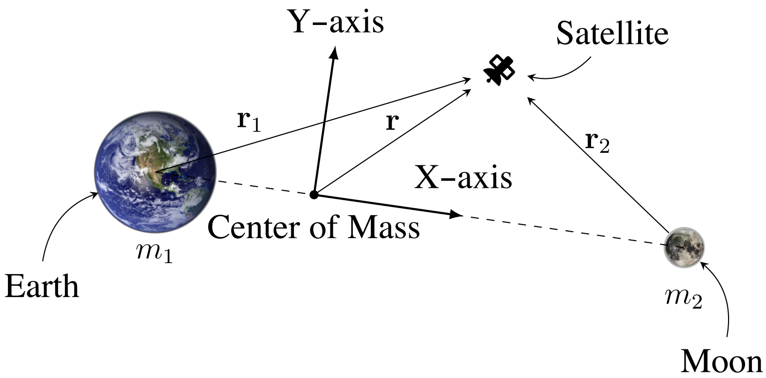
در فصل اول معادلات عمومی حرکت اجسام متعدد را معرفی کردیم. علی‌رغم وجود ده ثابت حرکت، هیچ‌کس نتوانسته است مسئله عمومی سه‌جسمی را به‌صورت تحلیلی بسته حل کند—ممکن است که این کار غیرممکن باشد. بنابراین، تحقیقات بر روی ساده‌سازی مسئله عمومی متمرکز شده است. ساندمن در سال ۱۹۱۲ یک راه‌حل سری توانی یافت. زمانی که این راه‌حل با شرایط اولیه ترکیب شود، ارزیابی‌های عددی از مسیرها را در یک بازه زمانی محدود ارائه می‌دهد. برای مطالعه کامل مسئله، به کتاب Szebehely (۱۹۶۷) مراجعه کنید. یکی از راه‌حل‌های تحلیلی خاص—مسئله سه‌جسمی محدود—از زمان اویلر و لاگرانژ شناخته شده است. اخیراً، از تکنیک‌های عددی برای تولید راه‌حل‌ها استفاده شده است.

دانشمندان برای صدها سال مسئله سه‌جسمی را مطالعه کرده‌اند، اگرچه بیشتر تحقیقات از دهه ۱۹۶۰، با استفاده از فناوری محاسباتی مدرن، انجام شده است. تحلیل‌های آنها انواع مختلفی از مدارهای سه‌جسمی را کشف کرده است که بسیاری از آنها برای فضاپیماها بسیار مفید هستند. در این بخش، چندین گزینه مدار سه‌جسمی را بررسی خواهیم کرد.

۵-۱ مسئله سه‌جسمی محدود دایره‌ای



شکل ۵-۱: هندسه مسئله سه بدنه محدود



شکل ۵-۲: هندسه مسئله سه بدنه محدود

جدول ۵-۱: مقادیر عددی برای مسئله سه‌جسمی محدود (سیستم زمین-ماه)

پارامتر	توصیف	مقدار عددی
m_1	جرم زمین	5.972×10^{24} kg
m_2	جرم ماه	7.348×10^{22} kg
μ	نسبت جرمی	0.0121505856
ω	سرعت زاویه‌ای سیستم	2.6617×10^{-6} rad/s

در مسئله‌ی سه‌جسمی محدود دایره‌ای (CRTBP) دو جرم بزرگ زمین و ماه در حال گردش دایره‌ای حول مرکز جرم مشترک هستند و جرم سوم فضاپیما که بسیار کوچک و تقریباً بی‌اثر است، تحت تاثیر گرانش آن دو حرکت می‌کند. فرض شده است که جرم سوم آن‌قدر کوچک است که تاثیری بر حرکت دو جرم اصلی ندارد. برای تحلیل این مسئله، یک دستگاه مختصات چرخان هم‌دوران با دو جرم اصلی انتخاب شده است به‌طوری‌که مرکز

مختصات در مرکز جرم سیستم باشد و محور x خط واصل دو جرم و محور y عمود بر آن در صفحه‌ی حرکت باشد (صفحه‌ی مداری دو جرم اصلی). سرعت زاویه‌ای چرخش این دستگاه برابر سرعت مداری دو جرم اصلی ($\omega = 1$ در دستگاه بی‌بعد) در نظر گرفته شده است. واحد طول برابر با فاصله‌ی بین دو جرم اصلی و واحد زمان چنان انتخاب شده است که دوره‌ی مداری دو جرم اصلی 2π (و در نتیجه $\omega = 1$) شود. همچنین مجموع جرم‌های دو جرم اصلی برابر یک در نظر گرفته شده است. بر این اساس اگر μ نسبت جرمی جرم دوم (کوچک‌تر) باشد ($\mu = \dfrac{m_2}{m_1 + m_2}$)، آنگاه جرم اول $m_1 = 1 - \mu$ و جرم دوم $m_2 = \mu$ خواهد بود. در مبدأ مختصات چرخان که مرکز جرم است، موقعیت جرم اول روی محور x برابر $(-\mu, 0)$ و جرم دوم برابر $(1 - \mu, 0)$ است.

در این چارچوب مرجع چرخان، جرم سوم یک **لاگرانژی** (تابع لاگرانژ) دارد که شامل انرژی جنبشی آن و پتانسیل گرانشی دو جرم اصلی به‌همراه پتانسیل موثر مرکزگرا (ناشی از چارچوب غیرلخت) است. با نرمال‌سازی ثابت گرانش به $G = 1$، می‌توان **لاگرانژی** سیستم را به صورت زیر نوشت [۳۱]:

$$L = T - V = \frac{1}{2}\Big(\dot{x}^2 + \dot{y}^2 + \dot{z}^2\Big) - \left[-\frac{1-\mu}{r_1} - \frac{\mu}{r_2} - \frac{1}{2}(x^2 + y^2) \right] ,$$

که در آن T انرژی جنبشی ذره‌ی کوچک و V پتانسیل آن است. عبارت داخل کروشه در واقع پتانسیل گرانشی دو جرم اصلی به‌همراه پتانسیل گریز از مرکز (با علامت منفی) است: به‌ترتیب $-\dfrac{1-\mu}{r_1}$ و $-\dfrac{\mu}{r_2}$ پتانسیل گرانشی جرم‌های اول و دوم در نقطه‌ی (x, y, z) ذره‌ی کوچک (با فواصل r_1 و r_2 تا آن دو جرم)، و $-\dfrac{1}{2}(x^2 + y^2)$ پتانسیل سانتریفیوژ معادل در دستگاه چرخان است. بنابراین تابع لاگرانژ را می‌توان به صورت جمع انرژی جنبشی و «پتانسیل موثر» زیر نوشت:

$$L = \frac{1}{2}\Big(\dot{x}^2 + \dot{y}^2 + \dot{z}^2\Big) + (1-\mu)\frac{1}{r_1} + \mu\frac{1}{r_2} + \frac{1}{2}(x^2 + y^2) ,$$

که در آن $r_1 = \sqrt{(x+\mu)^2 + y^2 + z^2}$ فاصله‌ی جرم سوم از جرم اول و $r_2 = \sqrt{(x-1+\mu)^2 + y^2 + z^2}$ فاصله از جرم دوم است. اکنون با به‌کارگیری اصل کمینه‌کردن کنش ($\delta S = 0$) و نوشتن معادلات اویلر–لاگرانژ برای مختصات x, y, z، معادلات حرکت در این دستگاه به‌دست می‌آیند. شرط اویلر–لاگرانژ برای هر مختصه (مثلاً x) به شکل $\dfrac{d}{dt}\dfrac{\partial L}{\partial \dot{x}} - \dfrac{\partial L}{\partial x} = 0$ نوشته می‌شود. مشتقات لازم را محاسبه می‌کنیم:

– $\dfrac{\partial L}{\partial \dot{x}} = \dot{x}$، پس $\dfrac{d}{dt}\dfrac{\partial L}{\partial \dot{x}} = \ddot{x}$، – $\dfrac{\partial L}{\partial x} = (1-\mu)\,\dfrac{\partial}{\partial x}\dfrac{1}{r_1} + \mu\,\dfrac{\partial}{\partial x}\dfrac{1}{r_2} + \dfrac{\partial}{\partial x}\Big(\dfrac{1}{2}x^2\Big)$

با توجه به $r_1 = \sqrt{(x+\mu)^2 + y^2 + z^2}$ و $r_2 = \sqrt{(x-1+\mu)^2 + y^2 + z^2}$، مشتق‌های جزئی پتانسیل‌ها عبارتند از:

$$\frac{\partial}{\partial x}\frac{1}{r_1} = -\frac{x+\mu}{r_1^3} \; , \qquad \frac{\partial}{\partial x}\frac{1}{r_2} = -\frac{x-(1-\mu)}{r_2^3} \; , \qquad \frac{\partial}{\partial x}\Big(\frac{1}{2}x^2\Big) = x \; .$$

بنابراین:

$$\frac{\partial L}{\partial x} = -(1-\mu)\frac{x+\mu}{r_1^3} - \mu\,\frac{x-(1-\mu)}{r_2^3} + x \; .$$

با قرار دادن این در معادله‌ی اویلر–لاگرانژ $\ddot{x} - \frac{\partial L}{\partial x} = 0$، معادله‌ی x به‌دست می‌آید. به‌طور مشابه برای y و z نیز عمل می‌کنیم. بدین‌ترتیب، **معادلات لاگرانژ (معادلات حرکت)** در دستگاه چرخان برای ذره‌ی سوم به صورت زیر حاصل می‌شوند:

$$\begin{cases} \ddot{x} - 2\,\dot{y} = x - \dfrac{1-\mu}{r_1^3}(x+\mu) - \dfrac{\mu}{r_2^3}\Big(x-(1-\mu)\Big) \; , \\[2ex] \ddot{y} + 2\,\dot{x} = y - \dfrac{1-\mu}{r_1^3}\,y - \dfrac{\mu}{r_2^3}\,y \; , \\[2ex] \ddot{z} = -\,\dfrac{1-\mu}{r_1^3}\,z - \dfrac{\mu}{r_2^3}\,z \; , \end{cases}$$

که در آن $r_1 = \sqrt{(x+\mu)^2+y^2+z^2}$ و $r_2 = \sqrt{\big(x-(1-\mu)\big)^2+y^2+z^2}$ همان‌گونه که تعریف شد. این معادلات لاگرانژ حاصل اصل کمینه‌سازی کنش هستند و معادلات کامل حرکت جرم سوم (با جرم ناچیز) را تحت گرانش دو جرم اولیه در حالت دایره‌ای بیان می‌کنند. توجه شود که وجود ترم‌های کورولیس ($2\,\dot{y}$ در معادله x و $2\,\dot{x}$ در معادله y) نتیجه‌ی استفاده از دستگاه مرجع چرخان است. همچنین می‌توان این معادلات را به شکل برداری $\ddot{\mathbf{r}} + 2\,\omega \times \dot{\mathbf{r}} = \nabla\Omega(\mathbf{r})$ نیز بیان کرد که در آن $\Omega(x,y,z) = \frac{1}{2}(x^2+y^2) + \frac{1-\mu}{r_1} + \frac{\mu}{r_2}$ پتانسیل موثر در دستگاه چرخان است و گرادیان آن نیروهای موثر (گرانشی و گریز از مرکز) را نتیجه می‌دهد.

حتماً. معادلات کامل لاگرانژ برای سیستم سه‌جسمی محدود دایره‌ای را به‌صورت دقیق در قالب لاتک آماده می‌کنم و توضیح می‌دهم که این معادلات از کجا به‌دست می‌آیند، چگونه با استفاده از آن‌ها نقاط تعادل استخراج می‌شوند، و در نهایت یک جدول شامل مختصات عددی نقاط لاگرانژی (L۱ تا L۵) برای سامانه زمین–ماه ارائه می‌دهم.

۲-۵ معادلات لاگرانژ در مسألهی سهجسمی محدود دایرهای

۳-۵

تعریف مسئله و چارچوب مرجع چرخان

۴-۵

نقاط تعادل (نقاط لاگرانژ L_1 تا L_5) منظور از نقطهی تعادل، حالتی است که جرم سوم (ذرهی آزمون) در دستگاه مرجع چرخان نسبت به دو جرم اصلی **ساکن** بماند. این شرایط زمانی رخ میدهد که سرعت و شتاب جرم سوم در دستگاه چرخان صفر شود. بهعبارت دیگر در معادلات بالا باید $\dot{x} = \dot{y} = \dot{z} = 0$ و $\ddot{x} = \ddot{y} = \ddot{z} = 0$ قرار دهیم. با اعمال این شرط به معادلات لاگرانژ، مجموعهای از معادلات جبری بهدست میآید که مختصات نقاط تعادل (موسوم به **نقاط لاگرانژ**) را تعیین میکند. با قراردادن $\dot{x} = \dot{y} = 0$ و $\ddot{x} = \ddot{y} = 0$ در معادلات داریم:

$$\begin{cases} 0 \ = \ x \ - \ \dfrac{1-\mu}{r_1^3}(x+\mu) \ - \ \dfrac{\mu}{r_2^3}\Big(x-(1-\mu)\Big) \ , \\ 0 \ = \ y \ - \ \dfrac{1-\mu}{r_1^3}\, y \ - \ \dfrac{\mu}{r_2^3}\, y \ , \\ 0 \ = \ -\dfrac{1-\mu}{r_1^3}\, z \ - \ \dfrac{\mu}{r_2^3}\, z \ . \end{cases}$$

معادلهی سوم در بالا نشان میدهد یا باید $z = 0$ باشد (نقاط تعادل همگی در صفحهی حرکت هستند) یا عبارت داخل آن صفر شود که برای $z \neq 0$ تنها در حالت خاصی مانند $\mu = 0$ امکانپذیر است. بنابراین برای یافتن نقاط تعادل، حرکت جرم سوم را در همان صفحهی مداری در نظر میگیریم ($z = 0$). در این صورت r_1 و r_2 در صفحه به ترتیب $\sqrt{(x+\mu)^2+y^2}$ و $\sqrt{\big(x-(1-\mu)\big)^2+y^2}$ خواهند بود. دو معادلهی اول را میتوان به صورت سادهتری نوشت:

$$\begin{cases} x - \dfrac{1-\mu}{r_1^3}(x+\mu) - \dfrac{\mu}{r_2^3}\Big(x-(1-\mu)\Big) = 0 \ , \\ y\left[1 - \dfrac{1-\mu}{r_1^3} - \dfrac{\mu}{r_2^3}\right] = 0 \ . \end{cases}$$

از معادله‌ی دوم نتیجه می‌شود که یا $y = 0$ (نقطه‌ی تعادل روی محور x واقع است) و یا پرانتز دوم صفر باشد (که منجر به رابطه‌ای بین فواصل می‌شود). بر این اساس، نقاط تعادل به دو دسته‌ی کلی تقسیم می‌شوند:

- **نقاط هم‌خط (Collinear):** این سه نقطه روی خط واصل دو جرم اصلی (محور x) واقع شده و بنابراین $y = 0$ دارند. این نقاط را به ترتیب L_1، L_2 و L_3 می‌نامند. - **نقاط سه‌گوش (Triangular):** این دو نقطه در صفحه به‌صورت رئوس مثلث متساوی‌الاضلاع با دو جرم اصلی قرار می‌گیرند و دارای $y \neq 0$ هستند. این دو نقطه L_4 و L_5 نام دارند.

در ادامه، هر دسته را جداگانه بررسی می‌کنیم.

۵-۵

نقاط لاگرانژ هم‌خط (L_1، L_2، L_3) برای نقاط واقع بر محور x، شرط $y = 0$ را در معادلات تعادل اعمال می‌کنیم. آنگاه معادله‌ی دوم به‌طور خودکار ارضا می‌شود (زیرا $y = 0$ آن را صفر می‌کند) و تنها معادله‌ی اول باقی می‌ماند:

$$x - \frac{1-\mu}{|x+\mu|^3}(x+\mu) - \frac{\mu}{|x-(1-\mu)|^3}\Big(x-(1-\mu)\Big) = 0\ ,$$

که در آن به علت قدرمطلق، بسته به ناحیه‌ی قرارگیری x، علامت عبارت‌ها مشخص می‌شود. خوشبختانه می‌توان سه ناحیه‌ی متمایز را در نظر گرفت که متناظر با سه ریشه‌ی این معادله هستند:

- **L_1:** بین دو جرم اصلی واقع است. در این حالت x بین $-\mu$ و $1-\mu$ قرار دارد (بین موقعیت‌های جرم اول و جرم دوم). برای L_1 فاصله‌ی آن از جرم دوم را با d_1 نشان می‌دهیم (پس $x = (1-\mu) - d_1$). این فاصله را می‌توان با حل معادله‌ی بالا به‌دست آورد. معادله‌ی تعادل در این ناحیه را می‌توان پس از ساده‌سازی به صورت یک معادله‌ی درجه پنج (در x یا d_1) نوشت که جواب تحلیلی ساده‌ای ندارد و معمولاً به روش عددی (مثلاً روش نیوتن) حل می‌شود. با این وجود، برای μهای کوچک (مثلاً سیستمی مانند خورشید-زمین یا زمین-ماه) می‌توان تخمین خوبی به‌دست آورد. اگر $\mu \ll 1$ باشد (جرم دوم بسیار کوچک‌تر از جرم اول)، آنگاه L_1 بسیار نزدیک به جرم دوم خواهد بود و فاصله‌ی آن از جرم دوم (در واحد فاصله‌ی دو جرم اصلی) تقریباً برابر **شعاع کره‌ی هیل** جرم دوم است. شعاع هیل تقریباً $r_h \approx (\mu/3)^{1/3}$ است. بنابراین می‌توان نوشت:

$$d_1 \approx \left(\frac{\mu}{3}\right)^{1/3},$$

و در نتیجه مختصات L_1 تقریباً برابر است با:

$$x_{L_1} \approx (1-\mu) - (\mu/3)^{1/3}, \qquad y_{L_1} = 0.$$

(توجه شود که برای دقت بیشتر، در صورت نیاز باید اثر μ را در قسمت $(1-\mu)$ نیز لحاظ کرد.) این فرمول یک تخمین از موقعیت L_1 است. در عمل حل عددی دقیق معادله، مقدار دقیق‌تری برای x_{L_1} به‌دست می‌دهد. این نقطه جایی است که نیروی گرانش دو جرم (یکی کشنده به چپ و دیگری به راست) و نیروی گریز از مرکز دقیقا همدیگر را خنثی می‌کنند.

- **L_2:** بیرون جرم دوم (کوچک) و در امتداد همان محور قرار دارد. این نقطه در سمت راست جرم دوم واقع است ($x > 1-\mu$) و جرم دوم بین آن و جرم اول قرار می‌گیرد. اگر فاصله‌ی L_2 از جرم دوم را d_2 بگیریم ($x = (1-\mu) + d_2$)، معادله‌ی تعادل در این ناحیه نیز پس از ساده‌سازی یک معادله‌ی درجه پنج برای d_2 خواهد بود که باید عددی حل شود. برای μ کوچک (جرم دوم خیلی کوچک‌تر)، این فاصله نیز تقریباً برابر $(\mu/3)^{1/3}$ به‌دست می‌آید. یعنی:

$$d_2 \approx \left(\frac{\mu}{3}\right)^{1/3},$$

و بنابراین:

$$x_{L_2} \approx (1-\mu) + (\mu/3)^{1/3}, \qquad y_{L_2} = 0.$$

این نقطه در بیرون مدار جرم دوم قرار دارد و تعادل نیروها در آن به این صورت است که نیروی گریز از مرکز به سمت بیرون توسط مجموع نیروی گرانشی دو جرم (هر دو به سمت داخل) بالانس می‌شود.

- **L_3:** بیرون جرم اول (بزرگ) و در امتداد خط محور x در سمت مقابل جرم دوم واقع است ($x < -\mu$). این نقطه در سوی دیگر جرم بزرگ‌تر قرار دارد به‌طوری که جرم اول بین L_3 و جرم دوم است. برای L_3 نیز معادله‌ی تعادل به روش عددی حل می‌شود؛ و در حد $\mu \ll 1$ (جرم دوم بسیار کوچک)، موقعیت آن اندکی فراتر از مدار جرم اول (جرم بزرگ) است. می‌توان نشان داد برای μهای بسیار کوچک:

$$x_{L_3} \approx -1 - \frac{5}{12}\mu, \qquad y_{L_3} = 0,$$

که نشان می‌دهد L_3 تقریباً به اندازه‌ی فاصله‌ی دو جرم اصلی دورتر از جرم اول است (حد $\mu \to 0$ منجر به $x = -1$ می‌شود که درست در سمت مقابل جرم دوم و هم‌فاصله با آن است) و با در نظر گرفتن جرم کوچک دوم، اندکی بیش از آن (چند صدم درصد بسته به مقدار μ) فاصله می‌گیرد. به بیان دیگر، گرانش جرم دوم

(هرچند کوچک) باعث می‌شود برای حفظ تعادل، جرم سوم واقع در L_3 کمی به جرم بزرگ‌تر نزدیک‌تر شود تا نیروی گریز از مرکز کمتر شده و با کاهش کمی در فاصله، گرانش جرم بزرگ افزایش یابد و تعادل برقرار گردد.

معمولاً معادله‌ی دقیق مربوط به L_1، L_2 و L_3 را به صورت یک چندجمله‌ای درجه ۵ نسبت به متغیری کمکی بیان می‌کنند (که به **معادله‌ی کوئینتیک لاگرانژ** مشهور است) و سپس آن را با تقریب یا روش‌های عددی حل می‌نمایند. به عنوان مثال، در مراجع کلاسیک نشان داده می‌شود که اگر $\rho = x + \mu$ فاصله‌ی مرکز جرم تا نقطه‌ی تعادل در یک جهت در نظر گرفته شود، معادله‌ی درجه پنج در ρ قابل بیان است. اما در کاربردهای عملی، استفاده از روش‌های عددی سریع‌تر به جواب منجر می‌شود.

۵-۶

نقاط لاگرانژ سه‌گوش (L_4، L_5) دو نقطه‌ی تعادل دیگر در صفحه‌ی مدار به‌صورتی قرار می‌گیرند که همراه با دو جرم اصلی تشکیل یک مثلث متساوی‌الاضلاع بدهند. در چنین آرایشی، جرم سوم می‌تواند در حالت تعادل نسبی (هم‌دوران با دو جرم دیگر) باقی بماند. برای این نقاط، واضح است که $y \neq 0$ خواهد بود. بنابراین در معادلات تعادل باید قسمت داخل کروشه (که در معادله‌ی تعادل y ظاهر شد) صفر گردد:

$$1 - \frac{1-\mu}{r_1^3} - \frac{\mu}{r_2^3} = 0.$$

این رابطه همراه با معادله‌ی تعادل در x باید همزمان ارضا شوند. از آنجا که مثلث متساوی‌الاضلاع است، فاصله‌ی جرم سوم از هر دو جرم اصلی برابر است ($r_1 = r_2$). با توجه به واحد طول نرمال‌شده (فاصله‌ی بین دو جرم اصلی برابر ۱)، این فاصله باید برابر با ۱ (طول ضلع مثلث) باشد. لذا $r_1 = r_2 = 1$. در این صورت، رابطه‌ی فوق به سادگی $1 - (1-\mu) - \mu = 0$ خواهد بود که برقرار است. با استفاده از هندسه‌ی مثلث متساوی‌الاضلاع در دستگاه مختصات تعریف‌شده می‌توان مختصات دقیق L_4 و L_5 را به‌دست آورد. اگر ضلع پایینی مثلث روی محور x باشد (جرم‌ها روی محور x قرار دارند) و جرم سوم رأس مثلث باشد، آنگاه مختصات آن عبارت‌اند از:

$$x_{L_4} \;=\; x_{L_5} \;=\; \frac{1}{2} - \mu\ , \qquad y_{L_4} \;=\; +\frac{\sqrt{3}}{2}\ , \qquad y_{L_5} \;=\; -\frac{\sqrt{3}}{2}\ .$$

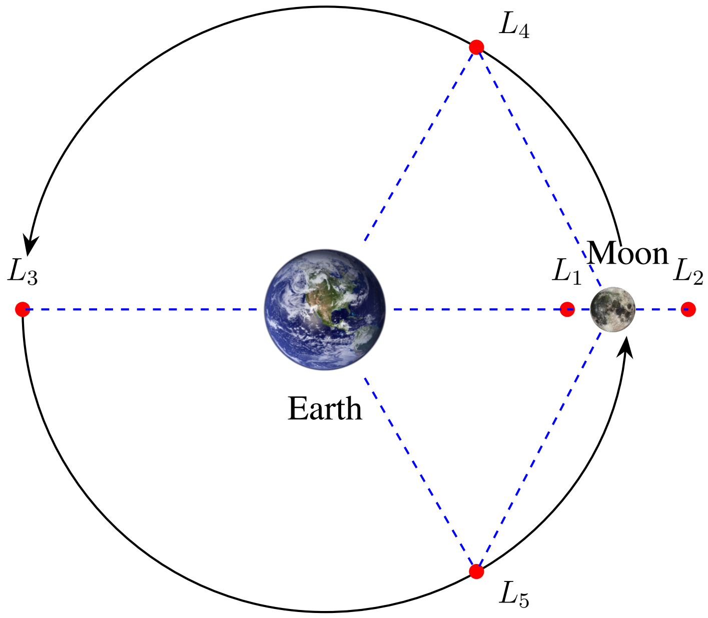
به بیان دیگر، در دستگاه باریسنتر چرخان، هر دوی L_4 و L_5 به اندازه‌ی نیمی از فاصله‌ی دو جرم روی محور x از مرکز جرم فاصله دارند (کمی جابه‌جاشده به سمت جرم بزرگ‌تر به اندازه‌ی μ) و مؤلفه‌ی عمودی (محور y) آن‌ها برابر $\pm\frac{\sqrt{3}}{2}$ است که نشان‌دهنده‌ی زاویه‌ی 60° نسبت به محور x می‌باشد. این دو نقطه، در

صورت کافی‌بودن نسبت جرم‌ها (بزرگ‌تر بودن قابل توجه جرم اول نسبت به جرم دوم؛ معمولاً $\frac{m_1}{m_2} > 24.96$ شرط پایداری است)، **نقاط تعادل پایدار** هستند. به عنوان مثال سیاره‌ی مشتری تعداد زیادی سیارک‌های تروجان در نقاط L_4 و L_5 خود نسبت به خورشید دارد. در مقابل، سه نقطه‌ی هم‌خط (L_1, L_2, L_3) پایداری ذاتی ندارند و اجسام در آن‌ها بدون تصحیح مداری معمولا از نقطه‌ی تعادل دور می‌شوند، با این حال برای تحقیقات فضایی (مانند رصدخانه‌ها یا رله‌های مخابراتی) محبوب‌اند زیرا مدارهای هاله‌ای یا لیساژور حول این نقاط با صرف انرژی اندک قابل حفظ هستند.

۷-۵

نمونه: نقاط لاگرانژ در سیستم زمین–ماه برای سیستم زمین–ماه مقدار $\mu \approx 0.01215$ است (نسبت جرم ماه به مجموع جرم زمین و ماه). در این سیستم با اعمال روابط به‌دست‌آمده می‌توان موقعیت عددی هر پنج نقطه‌ی لاگرانژ را محاسبه کرد. جدول زیر مختصات بی‌بعد (بر حسب فاصله‌ی زمین–ماه به عنوان واحد طول) این نقاط را در دستگاه مختصات تعریف‌شده‌ی باریسنتر (مرکز جرم در مبدأ، زمین در مختصات $(-\mu, 0)$ و ماه در $(1-\mu, 0)$) نشان می‌دهد. لازم به ذکر است که در این مقیاس، مختصات زمین تقریباً $(-0.01215,\ 0)$ و ماه $(0.98785,\ 0)$ می‌باشد. همان‌طور که مشاهده می‌شود، L_1 حدوداً در ۸۴% فاصله‌ی زمین–ماه (از سمت زمین) واقع شده و L_2 کمی بیرون مدار ماه قرار دارد. نقطه‌ی L_3 اندکی آن‌سوی مدار زمین (در جهت مخالف ماه) است. نقاط L_4 و L_5 نیز به‌ترتیب در بالا و پایین صفحه (در زاویه‌ی $60°$) و تقریباً در فاصله‌ی مساوی از زمین و ماه قرار گرفته‌اند.

نقطه‌ی لاگرانژ	x (بی‌بعد)	y (بی‌بعد)
L_1	$+0.83692$	0
L_2	$+1.15568$	0
L_3	-1.00506	0
L_4	$+0.48785$	$+0.86603$
L_5	$+0.48785$	-0.86603



شکل ۵-۳: نقاط لاگرانژ

جدول ۵-۲: مقادیر عددی برای مسئله سه‌جسمی محدود (سیستم زمین–ماه)

نقطه‌ی لاگرانژ	x (بی‌بعد)	y (بی‌بعد)
L_1	$+0.83692$	0
L_2	$+1.15568$	0
L_3	-1.00506	0
L_4	$+0.48785$	$+0.86603$
L_5	$+0.48785$	-0.86603

در جدول بالا علامت مثبت/منفی x نشان‌دهنده‌ی موقعیت نسبت به مرکز جرم (مبدأ مختصات) روی محور زمین–ماه است (مثبت به سمت ماه و منفی به سمت زمین) و محور y عمود بر آن صفحه‌ی مداری در جهت چرخش سیستم تعریف شده است. مقادیر عددی ارائه‌شده نشان می‌دهند که در منظومه‌ی زمین–ماه، L_1 تقریباً در فاصلهٔ 0.84 برابر فاصلهٔ زمین–ماه از زمین قرار دارد (یعنی فاصلهٔ آن از ماه حدود 0.16 برابر فاصلهٔ زمین–ماه است). به طور مشابه L_2 حدود 1.156 برابر فاصلهٔ زمین–ماه از زمین فاصله دارد (حدود 0.156 برابر فاصلهٔ زمین–ماه آن‌سوی مدار ماه). نقطه L_3 تقریباً در فاصله‌ای برابر با فاصلهٔ زمین–ماه در سمت مقابل (پشت زمین نسبت به ماه) قرارگرفته است. نقاط L_4 و L_5 نیز تقریباً در مختصات $(0.488, \ \pm0.866)$ واقع شده‌اند

که نشان‌دهندهٔ تشکیل مثلث متساوی‌الاضلاع با زمین و ماه می‌باشد. این مقادیر و تحلیلات تأیید می‌کنند که روابط تحلیلی به‌دست‌آمده برای معادلات لاگرانژ و نقاط تعادل، به‌خوبی می‌توانند برای توصیف وضعیت‌های پایدار و ناپایدار یک جرم کوچک در میدان گرانشی دو جرم آسمانی به‌کار روند.

مراجع: ۱. Richard H. Battin، *An Introduction to the Mathematics and Methods of Astrodynamics*، Revised Edition، AIAA Education Series، Eq.(۱) for Lagrange's quintic equation. (معادله درجه پنج لاگرانژ برای محاسبه نقاط L_1, L_2, L_3)

فصل ۶

شبیه‌سازی عامل درمحیط سه جسمی

در این فصل، فرآیند شبیه‌سازی عامل هوشمند کنترل‌کننده فضاپیما در محیط دینامیکی سه جسمی بررسی شده است. در بخش ۶-۱ به طراحی و در بخش ۶-۲ به شبیه‌سازی عامل هدایت‌کننده مبتنی بر یادگیری تقویتی است پرداخته شده است. این عامل طراحی و شبیه‌سازی شده باید توانایی این را داشته باشد که فضاپیما را به‌طور مؤثر به سمت اهداف تعیین‌شده هدایت کند، در حالی که محدودیت‌هایی نظیر مصرف سوخت و وجود اغتشاش دارد.

۶-۱ طراحی عامل

در این زیربخش، معماری عامل هوشمند کنترل‌کننده فضاپیما در محیط سه جسمی شرح داده شده است. این معماری شامل تعریف عامل، فضای حالت و عمل، و تابع پاداش است.

۶-۱-۱ فضای حالت

فضای حالت[1] در این پژوهش به‌گونه‌ای طراحی شده است که وضعیت دینامیکی فضاپیما را نسبت به یک مسیر و سرعت مرجع مشخص می‌کند. این فضا شامل اختلاف‌های موقعیت و سرعت از مسیر و سرعت مرجع است و به‌صورت زیر تعریف می‌شود:

$$S = \{\delta x, \delta y, \delta \dot{x}, \delta \dot{y}\}$$

که در آن:

[1]State Space

- $\delta x, \delta y$: اختلاف موقعیت فضاپیما نسبت به مسیر مرجع در محورهای x, y .
- $\delta \dot{x}, \delta \dot{y}$: اختلاف سرعت فضاپیما نسبت به سرعت مرجع در محورهای x, y .

هر یک از این متغیرها به‌طور مستقل وضعیت فضاپیما را در یک جهت خاص توصیف می‌کنند و امکان تحلیل دقیق انحرافات را فراهم می‌سازند. استفاده از اختلاف‌های موقعیت و سرعت به جای مقادیر مطلق، به دلایل زیر انجام شده است:

- **تمرکز بر انحرافات:** هدف اصلی سیستم کنترلی، کاهش انحرافات از مسیر و سرعت مطلوب است. با استفاده از اختلاف‌ها، کنترلر می‌تواند به طور مستقیم بر این انحرافات اثر بگذارد و نیازی به محاسبه مقادیر مطلق موقعیت و سرعت ندارد.
- **سازگاری با یادگیری تقویتی:** در الگوریتم‌های یادگیری تقویتی، فضاهای حالت مبتنی بر اختلاف معمولاً دامنه محدودتری دارند که فرآیند یادگیری را سریع‌تر و پایدارتر می‌کند.

۶-۱-۲ فضای عمل

فضای عمل[۲] فضاپیما با پیشران‌کم مجموعه‌ای از عمل‌های پیوسته است که فضاپیما می‌تواند در محیط شبیه‌سازی انجام دهد. این فضا به‌گونه‌ای طراحی شده که امکان اعمال نیرو در جهت‌های مشخص و با مقادیر متناسب با توان واقعی فضاپیماها فراهم شود. به‌طور خاص، فضای اقدام شامل موارد زیر است:

- **نیروی اعمال‌شده در جهت x:** این متغیر پیوسته، مقدار نیرویی را که در جهت محور x به فضاپیما وارد می‌شود، تعیین می‌کند. دامنه این نیرو بر اساس توان پیشرانه‌های موجود در فضاپیماهای واقعی انتخاب شده است. به عبارت دیگر، اگر حداکثر نیروی قابل اعمال در جهت x برابر با $f_{x,\max}$ باشد، این متغیر می‌تواند مقادیری در بازه $[-f_{x,\max}, f_{x,\max}]$ داشته باشد.
- **نیروی اعمال‌شده در جهت y:** این متغیر پیوسته، مقدار نیرویی را که در جهت محور y به فضاپیما وارد می‌شود، مشخص می‌کند. مشابه جهت x، دامنه این نیرو نیز بر اساس توان پیشرانه‌های موجود تعیین شده و می‌تواند در بازه $[-f_{y,\max}, f_{y,\max}]$ قرار گیرد.

انتخاب این نیروها بر اساس ویژگی‌های واقعی فضاپیماها، به‌ویژه توان و محدودیت‌های پیشرانه‌های آن‌ها، صورت گرفته است. این امر اطمینان می‌دهد که شبیه‌سازی تا حد ممکن به شرایط واقعی نزدیک باشد و نتایج

[2] Action Space

به‌دست‌آمده قابلیت تعمیم به کاربردهای عملی را داشته باشند. همچنین، تعریف فضای اقدام به‌صورت پیوسته، امکان کنترل دقیق و انعطاف‌پذیر بر حرکت فضاپیما را فراهم می‌کند، که برای دستیابی به اهداف کنترلی در محیط‌های دینامیکی پیچیده ضروری است. به‌طور خلاصه، فضای اقدام به‌صورت زیر تعریف می‌شود:

$$a = \{f_x, f_y \mid f_x \in [-f_{x,\text{max}}, f_{x,\text{max}}],\ f_y \in [-f_{y,\text{max}}, f_{y,\text{max}}]\}$$

۶-۱-۳ تابع پاداش

تابع پاداش[۳] به‌منظور هدایت رفتار عامل طراحی شده و شامل دو مؤلفه اصلی است:

- **پاداش برای دستیابی به هدف:** تشویق عامل برای نزدیک شدن به مدار هدف.
- **جریمه برای مصرف سوخت:** تنبیه برای استفاده بیش از حد از پیشرانه.
- **جریمه برای انحراف از مسیر مرجع:** تنبیه برای خروج از مسیر مرجع.

تابع پاداش به‌صورت زیر تعریف می‌شود:

$$r(s,a) = r_{\text{target}}(s) + r_{\text{thrust}}(a) + r_{\text{divergence}}(s)$$

که در آن مؤلفه‌های تابع پاداش به‌صورت زیر تعریف شده‌اند:

$$r_{\text{target}}(s) = -k_1 \cdot d(s, s_{\text{target}}) \tag{۶-۱}$$

$$r_{\text{thrust}}(a) = -k_2 \cdot |0a|0 \tag{۶-۲}$$

$$r_{\text{divergence}}(s) = \begin{cases} -k_3 & \text{if } d(s, s_{\text{reference}}) > \epsilon \\ 0 & \text{otherwise} \end{cases} \tag{۶-۳}$$

تابع $d(s,s')$ فاصله بین دو وضعیت s و s' را نشان می‌دهد که معمولاً به‌صورت فاصله اقلیدسی محاسبه می‌شود. ضرایب k_1, k_2, k_3 از طریق آزمایش و خطا تنظیم شده‌اند تا تعادل مناسبی بین دستیابی به هدف، بهینه‌سازی مصرف سوخت، و حفظ مسیر مرجع برقرار شود. علاوه بر این، این ضرایب تأثیر مستقیمی بر پایداری و فرآیند یادگیری عامل دارند. به‌عنوان مثال، انتخاب مقادیر بیش از حد بزرگ برای k_1 ممکن است باعث شود عامل به‌سرعت به سمت هدف حرکت کند اما پایداری مسیر را از دست بدهد، در حالی که مقادیر بزرگ k_3 می‌تواند عامل را بیش از حد محافظه‌کار کرده و فرآیند یادگیری را کند نماید. تنظیم دقیق این ضرایب، نه‌تنها عملکرد عامل را بهینه می‌کند، بلکه پایداری عددی و سرعت همگرایی الگوریتم یادگیری تقویتی را نیز تضمین می‌نماید.

[3] Reward Function

۶-۲ شبیه‌سازی عامل

در این زیربخش، فرآیند شبیه‌سازی و آموزش عامل با استفاده از الگوریتم‌های یادگیری تقویتی پیشرفته شرح داده می‌شود. الگوریتم‌های مورد استفاده، مراحل آموزش، و نتایج حاصل از شبیه‌سازی ارائه می‌گردند.

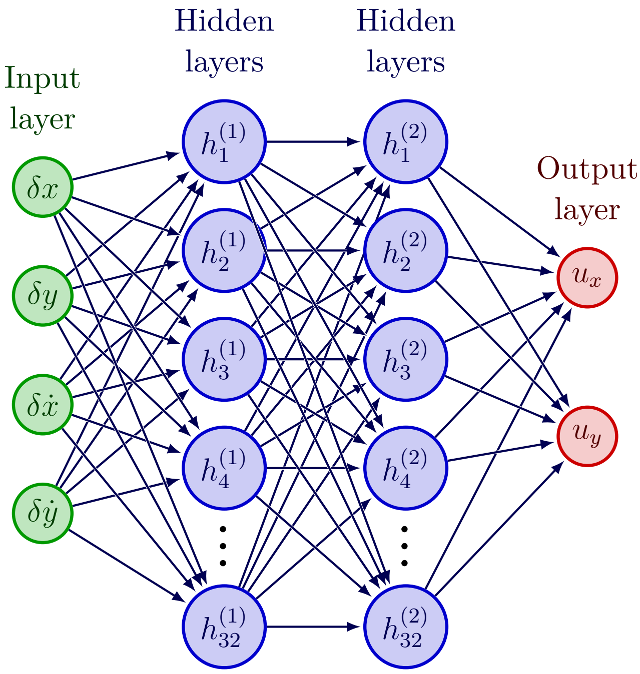
۶-۲-۱ الگوریتم‌های مورد استفاده

برای آموزش عامل، الگوریتم‌های زیر به‌کار گرفته شده‌اند:

جدول ۶-۱: ویژگی‌های الگوریتم‌های مورد استفاده در شبیه‌سازی

الگوریتم	شبکه Actor		شبکه Critic		تعداد پارامترها
	لایه‌ها	نودها	لایه‌ها	نودها	
DDPG	3	$(2^8, 2^7, 2^6)$	3	$(2^8, 2^7, 2^6)$	150×10^3
PPO	2	$(2^7, 2^6)$	2	$(2^7, 2^6)$	50×10^3
SAC	3	$(2^8, 2^7, 2^6)$	3	$(2^8, 2^7, 2^6)$	160×10^3
TD3	3	$(2^8, 2^7, 2^6)$	4	$(2^8, 2^7, 2^7, 2^6)$	200×10^3

این الگوریتم‌ها به دلیل توانایی در مدیریت فضاهای پیوسته و عملکرد مؤثر در محیط‌های پیچیده انتخاب شده‌اند. در شکل ۶-۱ شبکه عصبی شبیه‌سازی شده آورده شده است.



شکل ۶-۱: ساختار شبکه عصبی عامل

۶-۲-۲ فرآیند آموزش

آموزش عامل به‌صورت کلی در چند مرحله انجام شده است. ابتدا، کاوش اولیه در محیط با استفاده از یک سیاست تصادفی صورت گرفته و تجربه‌های اولیه جمع‌آوری شده‌اند. سپس، شبکه‌های عصبی الگوریتم‌ها با بهره‌گیری از این تجربه‌ها به‌روزرسانی شده‌اند. در نهایت، پارامترهای کلیدی مانند نرخ یادگیری و اندازه بافر تجربه تنظیم شده‌اند تا پایداری فرآیند تضمین شود.

برای پیاده‌سازی این فرآیند، از چارچوب PyTorch استفاده شده است. همچنین، به‌منظور جلوگیری از بیش‌برازش، تکنیک Noise Exploration به‌کار گرفته شده است. آموزش تا زمانی ادامه یافته که موفقیت عامل در بیش از ۹۰ درصد موارد به‌دست آمده باشد. در این راستا، برای بهینه‌سازی پارامترهای شبکه‌های عصبی، از روش Backpropagation استفاده شده است. این روش بر اساس گرادیان تابع خطا نسبت به پارامترها عمل می‌کند که به‌صورت زیر بیان می‌شود:

$$\frac{\partial L}{\partial w} = \frac{\partial L}{\partial y} \cdot \frac{\partial y}{\partial w} \qquad (۶-۴)$$

که در آن L تابع خطا، w وزن‌های شبکه، و y خروجی شبکه عصبی است. به‌روزرسانی وزن‌ها با استفاده از

روش گرادیان نزولی انجام شده است:

$$w_{t+1} = w_t - \eta \cdot \frac{\partial L}{\partial w} \qquad (۵-۶)$$

که η نرخ یادگیری است و به‌عنوان یک پارامتر کلیدی تنظیم شده است.

فصل ۷

سخت افزار در حلقه عملکرد عامل در محیط

فصل ۸

ارزیابی و نتایج یادگیری

در این فصل، نتایج حاصل از فرآیند یادگیری تقویتی در محیط سه‌جسمی ارائه و تحلیل شده است. هدف، بررسی عملکرد الگوریتم‌های استفاده‌شده و ارزیابی توانایی آن‌ها در دستیابی به اهداف تعیین‌شده می‌باشد.

۸-۱ تنظیمات آزمایشی

تنظیمات شبیه‌سازی، شامل پارامترهای محیط، نرخ یادگیری، و اندازه بافر تجربه، در این بخش تشریح شده است.

۸-۲ نتایج عملکرد الگوریتم‌ها

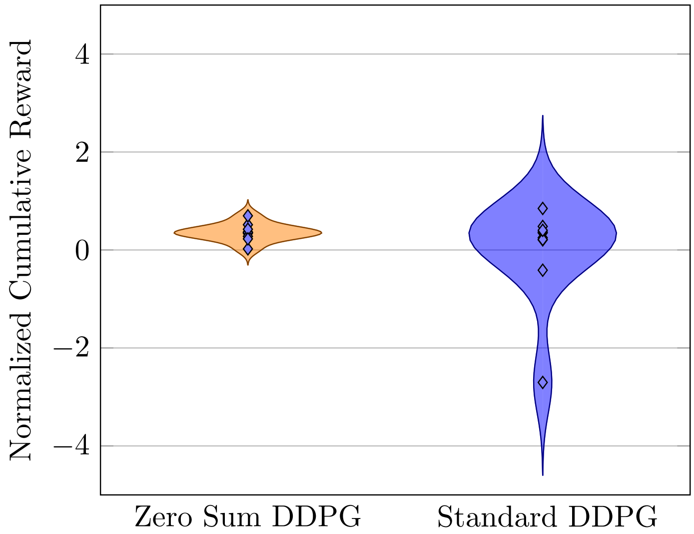
نتایج عملکرد الگوریتم‌های DDPG، PPO، SAC، و TD3 با معیارهایی نظیر زمان رسیدن به هدف و مصرف سوخت گزارش شده است.

۸-۳ تحلیل پایداری و همگرایی

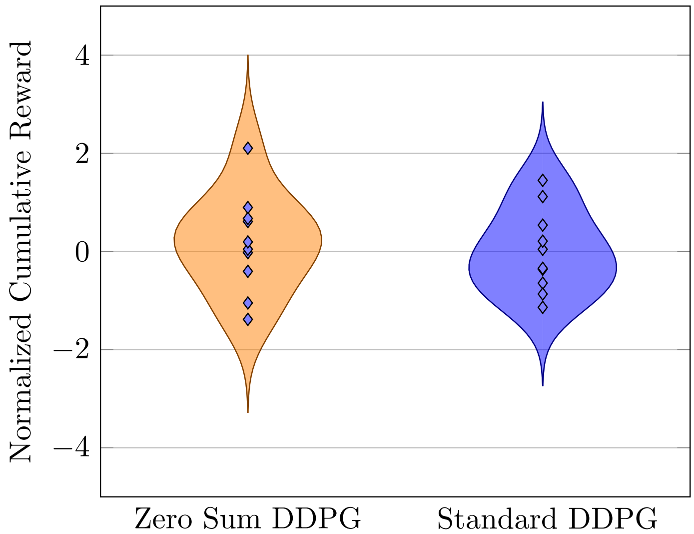
پایداری و سرعت همگرایی فرآیند یادگیری با استفاده از نمودارهای پاداش و معیارهای عددی مورد بررسی قرار گرفته است.

۸-۴ مقایسه با معیارهای مرجع

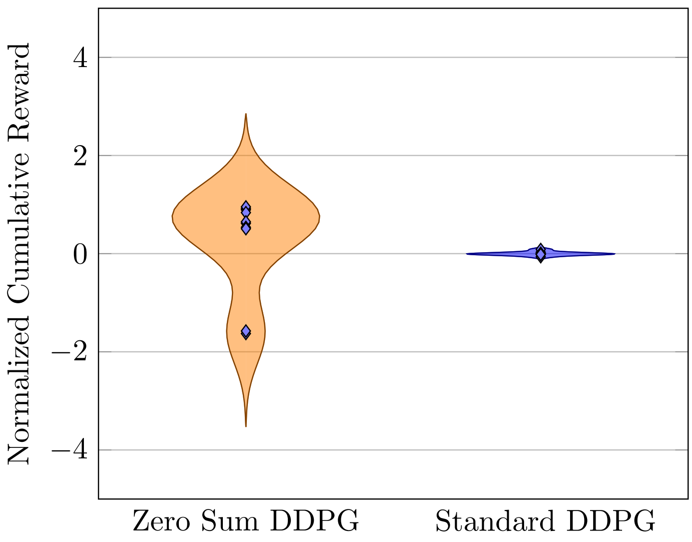
عملکرد الگوریتم‌ها با روش‌های مرجع مقایسه شده تا برتری‌ها و محدودیت‌های آن‌ها مشخص گردد.



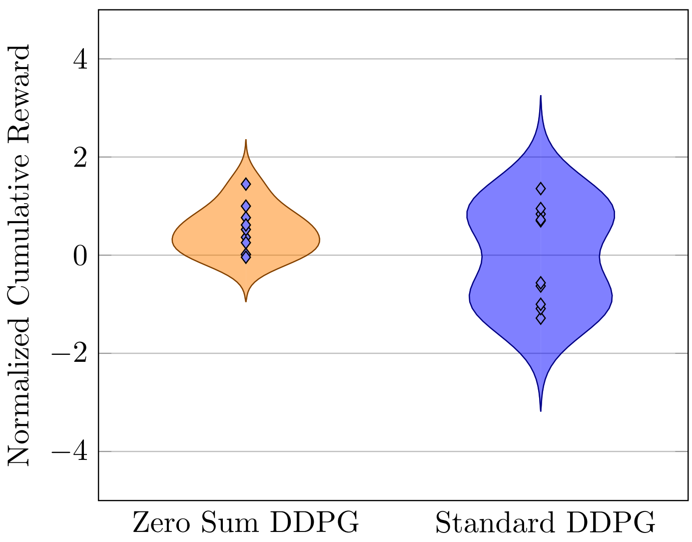
شکل ۸-۱: مقایسه مجموع پاداش دو الگوریتم تک‌عاملی و چندعاملی DDPG در شرایط اولیه تصادفی



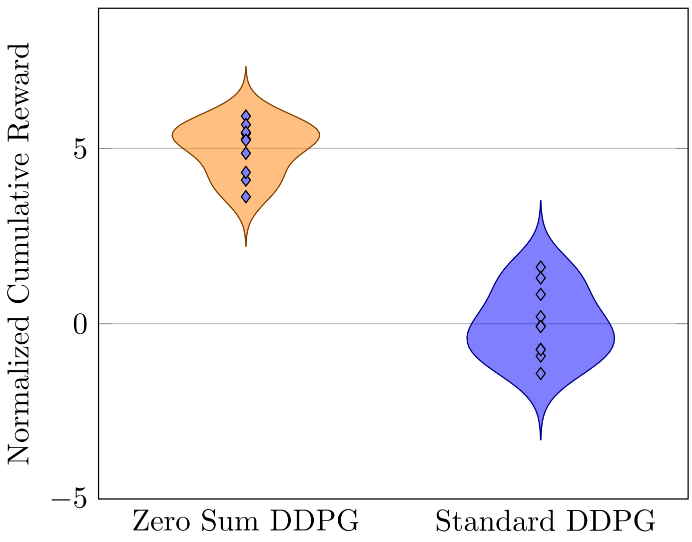
شکل ۸-۲: مقایسه مجموع پاداش دو الگوریتم تک‌عاملی و چندعاملی DDPG در حضور اغتشاش در عملگرها



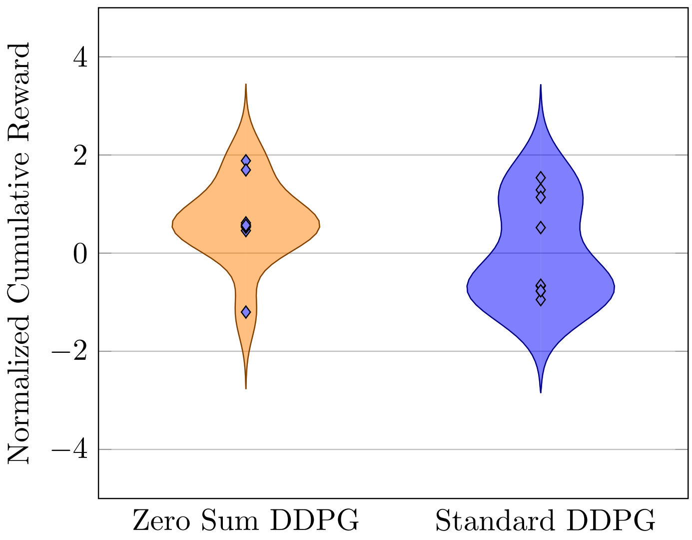
شکل ۸-۳: مقایسه مجموع پاداش دو الگوریتم تک‌عاملی و چندعاملی **DDPG** در مواجهه با عدم تطابق مدل



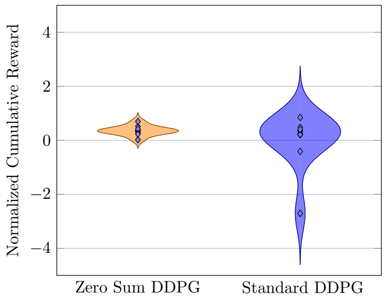
شکل ۸-۴: مقایسه مجموع پاداش دو الگوریتم تک‌عاملی و چندعاملی **DDPG** در شرایط مشاهده ناقص



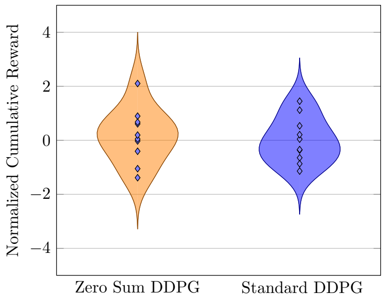
شکل ۸-۵: مقایسه مجموع پاداش دو الگوریتم تک‌عاملی و چندعاملی DDPG در حضور نویز حسگر



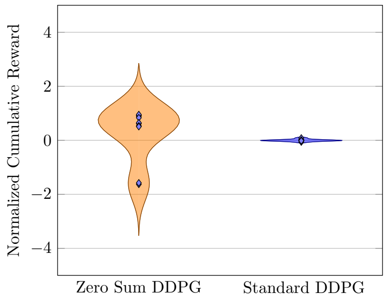
شکل ۸-۶: مقایسه مجموع پاداش دو الگوریتم تک‌عاملی و چندعاملی DDPG در شرایط تأخیر زمانی



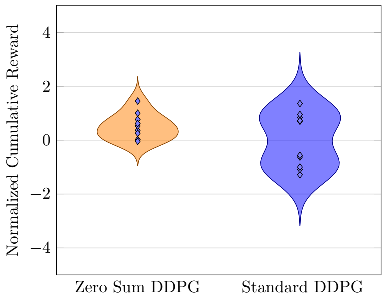
(آ) شرایط اولیه تصادفی



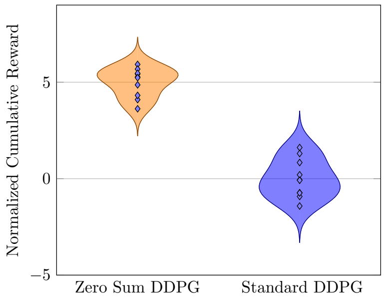
(ب) اغتشاش در عملگرها



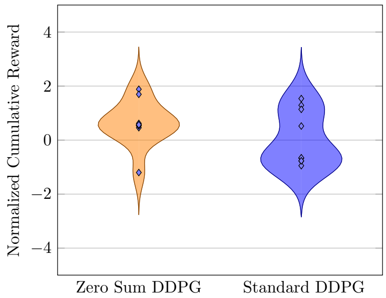
(ج) عدم تطابق مدل



(د) مشاهده ناقص

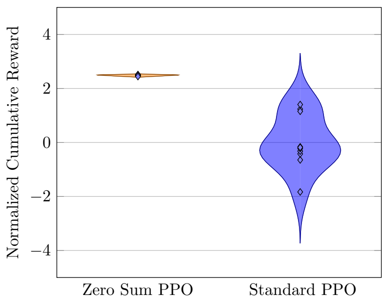


(ه) نویز حسگر

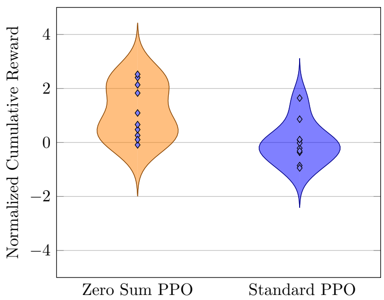


(و) تأخیر زمانی

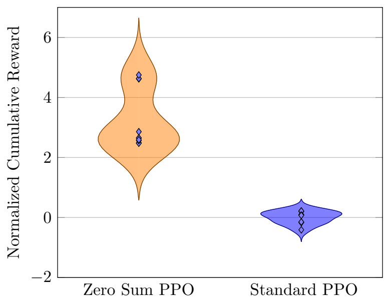
شکل ۸-۷: مقایسه مجموع پاداش دو الگوریتم تک‌عاملی و چندعاملی DDPG در سناریوهای مختلف



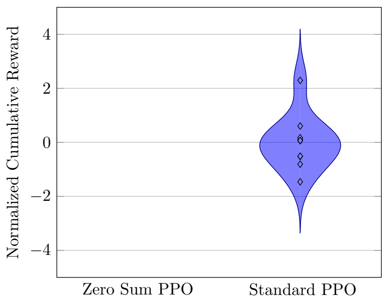
(آ) شرایط اولیه تصادفی



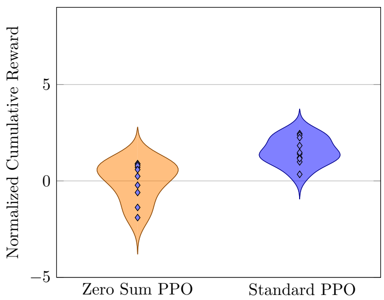
(ب) اغتشاش در عملگرها



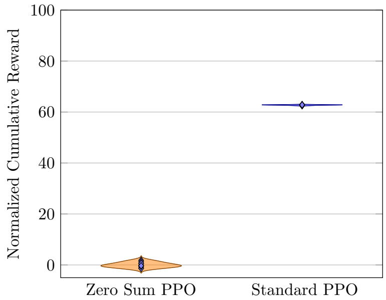
(ج) عدم تطابق مدل



(د) مشاهده ناقص



(ه) نویز حسگر



(و) تأخیر زمانی

شکل ۸-۸: مقایسه مجموع پاداش دو الگوریتم تک‌عاملی و چندعاملی PPO در سناریوهای مختلف

Bibliography

[1] M. A. Vavrina, J. A. Englander, S. M. Phillips, and K. M. Hughes. Global, multi-objective trajectory optimization with parametric spreading. In *AAS AIAA Astrodynamics Specialist Conference 2017*, 2017. Tech. No. GSFC-E-DAA-TN45282.

[2] C. Ocampo. Finite burn maneuver modeling for a generalized spacecraft trajectory design and optimization system. *Annals of the New York Academy of Sciences*, 1017:210–233, 2004.

[3] B. G. Marchand, S. K. Scarritt, T. A. Pavlak, and K. C. Howell. A dynamical approach to precision entry in multi-body regimes: Dispersion manifolds. *Acta Astronautica*, 89:107–120, 2013.

[4] A. F. Haapala and K. C. Howell. A framework for constructing transfers linking periodic libration point orbits in the spatial circular restricted three-body problem. *International Journal of Bifurcation and Chaos*, 26(05):1630013, 2016.

[5] B. Gaudet, R. Linares, and R. Furfaro. Six degree-of-freedom hovering over an asteroid with unknown environmental dynamics via reinforcement learning. In *20th AIAA Scitech Forum*, Orlando, Florida, 2020.

[6] B. Gaudet, R. Linares, and R. Furfaro. Terminal adaptive guidance via reinforcement meta-learning: Applications to autonomous asteroid close-proximity operations. *Acta Astronautica*, 171:1–13, 2020.

[7] A. Rubinsztejn, R. Sood, and F. E. Laipert. Neural network optimal control in astrodynamics: Application to the missed thrust problem. *Acta Astronautica*, 176:192–203, 2020.

[8] T. A. Estlin, B. J. Bornstein, D. M. Gaines, R. C. Anderson, D. R. Thompson, M. Burl, R. Castaño, and M. Judd. Aegis automated science targeting for the mer opportunity rover. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 3:1–19, 2012.

[9] R. Francis, T. Estlin, G. Doran, S. Johnstone, D. Gaines, V. Verma, M. Burl, J. Frydenvang, S. Montano, R. Wiens, S. Schaffer, O. Gasnault, L. Deflores, D. Blaney, and B. Bornstein. Aegis autonomous targeting for chemcam on mars science laboratory: Deployment and results of initial science team use. *Science Robotics*, 2, 2017.

[10] S. Higa, Y. Iwashita, K. Otsu, M. Ono, O. Lamarre, A. Didier, and M. Hoffmann. Vision-based estimation of driving energy for planetary rovers using deep learning and terramechanics. *IEEE Robotics and Automation Letters*, 4:3876–3883, 2019.

[11] B. Rothrock, J. Papon, R. Kennedy, M. Ono, M. Heverly, and C. Cunningham. Spoc: Deep learning-based terrain classification for mars rover missions. In *AIAA Space and Astronautics Forum and Exposition, SPACE 2016*. American Institute of Aeronautics and Astronautics Inc, AIAA, 2016.

[12] K. L. Wagstaff, G. Doran, A. Davies, S. Anwar, S. Chakraborty, M. Cameron, I. Daubar, and C. Phillips. Enabling onboard detection of events of scientific interest for the europa clipper spacecraft. In *25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2191–2201, Anchorage, Alaska, 2019.

[13] B. Dachwald. Evolutionary neurocontrol: A smart method for global optimization of low-thrust trajectories. In *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, pages 1–16, Providence, Rhode Island, 2004.

[14] S. D. Smet and D. J. Scheeres. Identifying heteroclinic connections using artificial neural networks. *Acta Astronautica*, 161:192–199, 2019.

[15] N. L. O. Parrish. *Low Thrust Trajectory Optimization in Cislunar and Translunar Space.* PhD thesis, University of Colorado Boulder, 2018.

[16] N. Heess, D. TB, S. Sriram, J. Lemmon, J. Merel, G. Wayne, Y. Tassa, T. Erez, Z. Wang, S. M. A. Eslami, M. A. Riedmiller, and D. Silver. Emergence of locomotion behaviours in rich environments. *CoRR*, abs/1707.02286, 2017.

[17] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. van den Driessche, T. Graepel, and D. Hassabis. Mastering the game of go without human knowledge. *Nature*, 550, 2017.

[18] R. Furfaro, A. Scorsoglio, R. Linares, and M. Massari. Adaptive generalized zem-zev feedback guidance for planetary landing via a deep reinforcement learning approach. *Acta Astronautica*, 171:156–171, 2020.

[19] B. Gaudet, R. Linares, and R. Furfaro. Deep reinforcement learning for six degrees of freedom planetary landing. *Advances in Space Research*, 65:1723–1741, 2020.

[20] B. Gaudet, R. Furfaro, and R. Linares. Reinforcement learning for angle-only intercept guidance of maneuvering targets. *Aerospace Science and Technology*, 99, 2020.

[21] D. Guzzetti. Reinforcement learning and topology of orbit manifolds for station-keeping of unstable symmetric periodic orbits. In *AAS/AIAA Astrodynamics Specialist Conference*, Portland, Maine, 2019.

[22] J. A. Reiter and D. B. Spencer. Augmenting spacecraft maneuver strategy optimization for detection avoidance with competitive coevolution. In *20th AIAA Scitech Forum*, Orlando, Florida, 2020.

[23] A. Das-Stuart, K. C. Howell, and D. C. Folta. Rapid trajectory design in complex environments enabled by reinforcement learning and graph search strategies. *Acta Astronautica*, 171:172–195, 2020.

[24] D. Miller and R. Linares. Low-thrust optimal control via reinforcement learning. In *29th AAS/AIAA Space Flight Mechanics Meeting*, Ka'anapali, Hawaii, 2019.

[25] C. J. Sullivan and N. Bosanac. Using reinforcement learning to design a low-thrust approach into a periodic orbit in a multi-body system. In *20th AIAA Scitech Forum*, Orlando, Florida, 2020.

[26] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller. Deterministic policy gradient algorithms. In *International conference on machine learning*, pages 387–395. Pmlr, 2014.

[27] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[28] S. Fujimoto, H. van Hoof, and D. Meger. Addressing function approximation error in actor-critic methods, 2018.

[29] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. 2017.

[30] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *CoRR*, abs/1801.01290, 2018.

[31] D. Vallado and W. McClain. *Fundamentals of Astrodynamics and Applications*. Fundamentals of Astrodynamics and Applications. Microcosm Press, 2001.

Abstract

In this study, a quadcopter stand with three degrees of freedom was controlled using game theory-based control. The first player tracks a desired input, and the second player creates a disturbance in the tracking of the first player to cause an error in the tracking. The move is chosen using the Nash equilibrium, which presupposes that the other player made the worst move.. In addition to being resistant to input interruptions, this method may also be resilient to modeling system uncertainty. This method evaluated the performance through simulation in the Simulink environment and implementation on a three-degree-of-freedom stand.

Keywords: Quadcopter, Differential Game, Game Theory, Nash Equilibrium, Three Degree of Freedom Stand, Model Base Design, Linear Quadratic Regulator

Sharif University of Technology

Department of Aerospace Engineering

Master Thesis

Robust Reinforcement Learning Differential Game Guidance in Low-Thrust, Multi-Body Dynamical Environments

By:

Ali BaniAsad

Supervisor:

Dr.Hadi Nobahari

December 2024