



دانشگاه صنعتی شریف  
دانشکده مهندسی هوافضا

پروژه کارشناسی ارشد  
مهندسی فضا

عنوان:

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

نگارش:

علی بنی اسد

استاد راهنما:

دکتر هادی نوبهاری

۱۴۰۴ شهریور

مَلِكُ الْأَنْفُلِ

به نام خدا

دانشگاه صنعتی شریف

دانشکده‌ی مهندسی هوا فضا

پروژه کارشناسی ارشد

عنوان:

هدايت يادگيري تقويتی مقاوم مبتنی بر بازی دیفرانسیلی در محیط‌های پویای چندجسمی با  
پیشران کم

نگارش: علی بنی اسد

کمیته‌ی ممتحنین

استاد راهنمای: دکتر هادی نوبهاری  
امضاء:

استاد ممتحن: دکتر سید علی امامی خوانساری  
امضاء:

استاد ممتحن: دکتر علیرضا با صحبت نوین زاده  
امضاء:

تاریخ:

## سپاس

از استاد بزرگوارم، جناب آقای دکتر هادی نوبهاری، به پاس خدمات فراوانی که برای این شاگرد کوچکشان کشیدند، صمیمانه سپاسگزارم. در عرصه‌های گوناگون علمی، اخلاقی و انسانی از ایشان بسیار آموختم و بی‌گمان توفیق شاگردی ایشان برای من لطفی الهی بوده است که به آن می‌باشد. بخش چشمگیری از موفقیت‌هایم در دوره‌های کارشناسی و کارشناسی ارشد را مرهون راهنمایی‌ها و همراهی‌های این بزرگوار هستم. برای ایشان و خانواده‌ی گرامیشان سلامتی، عزت و سربلندی روزافزون در همه‌ی مراحل زندگی آرزو می‌کنم و امیدوارم بتوانم دین‌شاگردی را به خوبی ادا کنم. همچنین، از جناب آقای مهندس میثم علیزاده که نظرات ارزشمند او همواره راهگشای مشکلات بنده بود، تشکر می‌کنم. از پدر دلسوزم ممنونم که در انجام این پژوهش 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 این دست نوشته و پژوهش وجود نداشت.

## چکیده

در این پژوهش، یک چارچوب هدایت مقاوم برای فضایپیماهای کمپیشران در محیط‌های دینامیکی چندجسمی (سامانه سه جسمی زمین-ماه) ارائه شده است. مسئله به صورت بازی دیفرانسیلی مجموع صفر بین عامل هدایت (فضایپیما) و عامل مزاحم (عدم قطعیت‌های محیطی) فرمول‌بندی شده و با رویکرد آموزش متمنکز و اجرای توزیع شده پیاده‌سازی گردیده است. در این راستا، چهار الگوریتم یادگیری تقویتی پیوسته SAC، TD3، DDPG و PPO به نسخه‌های چندعاملی مجموع صفر گسترش یافته‌اند (MA-TD3، MA-DDPG، MA-SAC و MA-PPO) و جریان آموزش آن‌ها همراه با ساختار شبکه‌ها در قالب اطلاعات کامل تشریح شده است. ارزیابی الگوریتم‌ها در سناریوهای متنوع عدم قطعیت شامل شرایط اولیه تصادفی، اغتشاش عملگر، نویز حسگر، تأخیر زمانی و عدم تطابق مدل روی مسیر مدار لیاپانوف زمین-ماه انجام گرفت. نتایج به‌وضوح نشان می‌دهد که نسخه‌های مجموع صفر در تمامی معیارهای ارزیابی بر نسخه‌های تک‌عاملی برتری دارند. بهویژه الگوریتم MA-TD3 با حفظ پایداری سیستم، کمترین انحراف مسیر و مصرف سوخت بهینه را حتی در سخت‌ترین سناریوهای آزمون از خود نشان داد. در نهایت، چارچوب پیشنهادی نشان می‌دهد که یادگیری تقویتی چندعاملی مبتنی بر بازی دیفرانسیلی مجموع صفر می‌تواند بدون نیاز به مدل‌سازی دقیق، هدایت تطبیقی و مقاوم فضایپیماهای کمپیشران را در نواحی ناپایدار سیستم‌های سه‌جسمی تضمین کند.

**کلیدواژه‌ها:** یادگیری تقویتی عمیق، بازی دیفرانسیلی، سامانه‌های چندعاملی، هدایت کمپیشران، بازی مجموع صفر، مسئله محدود سه‌جسمی، کنترل مقاوم.

# فهرست مطالب

۱	۱	مقدمه
۱	۱-۱	انگیزه پژوهش
۲	۲-۱	تعریف مسئله
۳	۳-۱	یادگیری تقویتی
۴	۴-۱	یادگیری تقویتی چندعاملی
۴	۴-۱	ارزش افزوده و نوآوری‌های پژوهش
۵	۶-۱	ساختارگزارش
۶	۲	پیشینه پژوهش
۶	۱-۲	ماموریت‌های بین‌مداری
۸	۲-۲	یادگیری تقویتی
۹	۳-۲	پیشینه‌ی پژوهش یادگیری تقویتی چندعاملی
۱۱	۴-۲	جمع‌بندی و گپ پژوهشی
۱۳	۳	مدل‌سازی محیط یادگیری سه جسمی
۱۳	۱-۳	مسئله‌ی سه‌جسمی محدود دایره‌ای (CRTBP)
۱۵	۱-۱-۳	۱-۱-۳ لاگرانژ و معادلات حرکت
۱۵	۲-۳	۲-۳ نقاط تعادل لاگرانژ

## ۴ یادگیری تقویتی

۱۹	۱-۴ مفاهیم اولیه
۲۰	۱-۱-۴ حالت و مشاهدات
۲۰	۲-۱-۴ فضای عمل
۲۰	۳-۱-۴ سیاست
۲۱	۴-۱-۴ مسیر
۲۱	۵-۱-۴ تابع پاداش و برگشت
۲۲	۶-۱-۴ ارزش در یادگیری تقویتی
۲۳	۷-۱-۴ معادلات بلمن
۲۴	۸-۱-۴ تابع مزیت
۲۵	۲-۴ عامل گرادیان سیاست عمیق قطعی
۲۵	۱-۲-۴ یادگیری Q در DDPG
۲۷	۲-۲-۴ سیاست در DDPG
۲۷	۳-۲-۴ اکتشاف و بهره‌برداری در DDPG
۲۷	۴-۲-۴ شبکه DDPG
۲۹	۳-۴ عامل گرادیان سیاست عمیق قطعی تاخیری دوگانه
۳۰	۱-۳-۴ اکتشاف و بهره‌برداری در TD3
۳۰	۲-۳-۴ شبکه TD3
۳۲	۴-۴ عامل عملگر نقاد نرم
۳۲	۱-۴-۴ یادگیری تقویتی تنظیم شده با آنتروپی
۳۲	۲-۴-۴ سیاست در SAC
۳۳	۳-۴-۴ تابع ارزش در SAC
۳۳	۴-۴-۴ تابع Q در SAC
۳۳	۵-۴-۴ معادله بلمن در SAC

۳۴	.....	۶-۴-۴ یادگیری Q
۳۴	.....	۷-۴-۴ سیاست در SAC
۳۵	.....	۸-۴-۴ اکتشاف و بهره‌برداری در SAC
۳۶	.....	۹-۴-۴ شبکه SAC
۳۷	.....	۵-۴ عامل بهینه‌سازی سیاست مجاور
۳۸	.....	۱-۵-۴ سیاست در الگوریتم PPO
۳۹	.....	۲-۵-۴ اکتشاف و بهره‌برداری در PPO
۳۹	.....	۳-۵-۴ شبکه PPO
۴۱	.....	۵ شبیه‌سازی عامل در محیط سه جسمی
۴۱	.....	۱-۵ طراحی عامل
۴۱	.....	۱-۱-۵ فضای حالت
۴۲	.....	۲-۱-۵ فضای عمل
۴۴	.....	۳-۱-۵ تابع پاداش
۴۵	.....	۲-۵ شبیه‌سازی عامل
۴۵	.....	۱-۲-۵ پارامترهای یادگیری و منطق انتخاب الگوریتمها
۴۹	.....	۲-۲-۵ فرآیند آموزش
۵۱	.....	۶ یادگیری تقویتی چندعاملی
۵۱	.....	۶-۱ تعاریف و مفاهیم اساسی
۵۳	.....	۶-۲ نظریه بازی‌ها
۵۳	.....	۱-۲-۶ تعادل نش
۵۴	.....	۲-۲-۶ بازی مجموع صفر
۵۶	.....	۳-۲-۶ چالش‌های استایی در یادگیری تقویتی تک‌عاملی
۵۷	.....	۴-۲-۶ ضرورت استفاده از یادگیری تقویتی چندعاملی

۵۸	.....	۳-۶ گرادیان سیاست عمیق قطعی چندعاملی
۵۹	.....	۱-۳-۶ معماری MA-DDPG در بازی‌های مجموع صفر
۵۹	.....	۲-۳-۶ آموزش MA-DDPG در بازی‌های مجموع صفر
۶۰	.....	۳-۳-۶ اکتشاف در MA-DDPG
۶۰	.....	۴-۳-۶ شبکه MA-DDPG برای بازی‌های دو عاملی مجموع صفر
۶۲	.....	۵-۳-۶ مزایای MA-DDPG در بازی‌های مجموع صفر
۶۲	.....	۴-۶ عامل گرادیان سیاست عمیق قطعی تا خیری دوگانه چندعاملی
۶۲	...MA-TD3	۱-۴-۶ چالش‌های یادگیری تقویتی در محیط‌های چندعاملی و راه حل
۶۳	.....	۲-۴-۶ معماری MA-TD3 در بازی‌های مجموع صفر
۶۳	.....	۳-۴-۶ آموزش MA-TD3
۶۴	.....	۴-۴-۶ اکتشاف در MA-TD3
۶۴	.....	۵-۴-۶ شبکه MA-TD3 برای بازی‌های چندعاملی مجموع صفر
۶۶	.....	۶-۴-۶ مزایای MA-TD3 در بازی‌های مجموع صفر
۶۶	.....	۵-۶ عامل عملگر نقاد نرم چندعاملی
۶۶	...MA-SAC	۱-۵-۶ چالش‌های یادگیری تقویتی در محیط‌های چندعاملی و راه حل
۶۷	.....	۲-۵-۶ معماری MA-SAC در بازی‌های مجموع صفر
۶۷	.....	۳-۵-۶ آموزش MA-SAC
۶۹	.....	۴-۵-۶ اکتشاف در MA-SAC
۶۹	.....	۵-۵-۶ شبکه MA-SAC برای بازی‌های چندعاملی مجموع صفر
۷۱	.....	۶-۵-۶ مزایای MA-SAC در بازی‌های مجموع صفر
۷۱	.....	۶-۶ عامل بهینه‌سازی سیاست مجاور چندعاملی
۷۱	...MA-PPO	۱-۶-۶ چالش‌های یادگیری تقویتی در محیط‌های چندعاملی و راه حل
۷۲	.....	۲-۶-۶ معماری MA-PPO در بازی‌های مجموع صفر
۷۲	.....	۳-۶-۶ آموزش MA-PPO

۷۴	.....	۴-۶-۶ اکتشاف در MA-PPO
۷۴	.....	۵-۶-۶ شبکه MA-PPO برای بازی‌های چندعاملی مجموع صفر
۷۵	.....	۶-۶-۶ مزایای MA-PPO در بازی‌های مجموع صفر
۷۷	.....	<b>۷ ارزیابی و نتایج یادگیری</b>
۷۷	.....	۱-۷ ارزیابی مقاومت الگوریتم‌ها
۷۸	.....	۱-۱-۷ سناریوهای ارزیابی مقاومت
۷۹	.....	۲-۷ الگوریتم DDPG
۷۹	.....	۱-۲-۷ مسیر طی شده
۷۹	.....	۲-۲-۷ مسیر و فرمان پیشران
۸۰	.....	۳-۲-۷ توزیع پاداش تجمعی
۸۱	.....	۴-۲-۷ مقایسه عددی
۸۲	.....	۳-۷ الگوریتم TD3
۸۲	.....	۱-۳-۷ مسیر طی شده
۸۳	.....	۲-۳-۷ مسیر و فرمان پیشران
۸۳	.....	۳-۳-۷ توزیع پاداش تجمعی
۸۴	.....	۴-۳-۷ مقایسه عددی
۸۶	.....	۴-۷ الگوریتم SAC
۸۶	.....	۱-۴-۷ مسیر طی شده
۸۷	.....	۲-۴-۷ مسیر و فرمان پیشران
۸۸	.....	۳-۴-۷ توزیع پاداش تجمعی
۸۸	.....	۴-۴-۷ مقایسه عددی
۸۹	.....	۵-۷ الگوریتم PPO
۹۰	.....	۱-۵-۷ مسیر طی شده
۹۰	.....	۲-۵-۷ مسیر و فرمان پیشران

۹۱	توزیع پاداش تجمعی	۳-۵-۷
۹۱	مقایسه عددی	۴-۵-۷
۹۳	نتایج نسخه استاندارد	۶-۷
۹۳	توزیع پاداش تجمعی	۱-۶-۷
۹۴	مقایسه عددی	۲-۶-۷
۹۵	نتایج نسخه چندعاملی	۷-۷
۹۶	توزیع پاداش تجمعی	۱-۷-۷
۹۷	مقایسه عددی	۲-۷-۷
۹۹	۸ نتیجه‌گیری و پیشنهادها	
۹۹	جمع‌بندی دستاوردها	۱-۸
۱۰۰	محدودیت سناریوهای آزمون	۲-۸
۱۰۰	پیشنهادهایی برای کارهای آینده	۳-۸

# فهرست جداول

۱-۳	مقادیر عددی برای مسئله سه‌جسمی محدود (سامانه زمین-ماه) . . . . .	۱۴
۲-۳	مقادیر عددی نقاط لاغرانژ برای مسئله سه‌جسمی محدود سیستم زمین-ماه . . . . .	۱۷
۱-۵	قابلیت‌های بی‌بعد پیش‌رانکم تراستِ فضای‌پیماهای مختلف در سامانه‌ی زمین-ماه [۵۵]. . . . .	۴۳
۲-۵	جدول پارامترها و مقادیر پیش‌فرض الگوریتم DDPG [۶۲] . . . . .	۴۶
۳-۵	جدول پارامترها و مقادیر پیش‌فرض الگوریتم TD3 [۶۲] . . . . .	۴۶
۴-۵	جدول پارامترها و مقادیر پیش‌فرض الگوریتم SAC [۶۲] . . . . .	۴۷
۵-۵	جدول پارامترها و مقادیر پیش‌فرض الگوریتم PPO [۶۲] . . . . .	۴۷
۱-۷	مقایسه عملکرد DDPG و MA-DDPG در سناریوهای مختلف مقاومت . . . . .	۸۱
۲-۷	مقایسه عملکرد TD3 و MA-TD3 در سناریوهای مختلف مقاومت . . . . .	۸۵
۳-۷	مقایسه عملکرد SAC و MA-SAC در سناریوهای مختلف مقاومت . . . . .	۸۹
۴-۷	مقایسه عملکرد PPO و MA-PPO در سناریوهای مختلف مقاومت . . . . .	۹۲
۵-۷	مقایسه الگوریتم‌های چند‌عاملی در سناریوهای مختلف مقاومت . . . . .	۹۴
۶-۷	مقایسه الگوریتم‌های چند‌عاملی در سناریوهای مختلف مقاومت . . . . .	۹۷

# فهرست تصاویر

۱-۳ هندسه‌ی مسئله‌ی سه‌جسمی محدود در چارچوب چرخان	۱۴
۲-۳ نقاط لاگرانژ در سامانه‌ی زمین-ماه	۱۶
۱-۴ حلقه تعامل عامل و محیط	۲۰
۱-۵ ساختار شبکه عصبی سیاست	۴۸
۲-۵ ساختار شبکه عصبی نقاد	۴۸
۱-۶ حلقه تعامل عامل‌های یادگیری تقویتی چند عاملی با محیط	۵۳
۱-۷ مسیر طی‌شده فضایپیما با DDPG استاندارد و نسخه بازی مجموع صفر MA-DDPG	۷۹
۲-۷ مسیر و فرمان پیشran فضایپیما در DDPG استاندارد و نسخه بازی مجموع صفر MA-DDPG	۸۰
۳-۷ مقایسه توزیع پاداش تجمعی در سناریوهای مختلف برای DDPG و MA-DDPG	۸۰
۴-۷ مسیر طی‌شده فضایپیما با TD3 استاندارد و نسخه بازی مجموع صفر MA-TD3	۸۳
۵-۷ مسیر و فرمان پیشran فضایپیما در TD3 استاندارد و نسخه بازی مجموع صفر MA-TD3	۸۳
۶-۷ مقایسه توزیع پاداش تجمعی در سناریوهای مختلف برای TD3 و MA-TD3	۸۴
۷-۷ مسیر طی‌شده فضایپیما با SAC استاندارد و نسخه بازی مجموع صفر MA-SAC	۸۶
۸-۷ مسیر و فرمان پیشran فضایپیما در SAC استاندارد و نسخه بازی مجموع صفر MA-SAC	۸۷
۹-۷ مقایسه توزیع پاداش تجمعی در سناریوهای مختلف برای SAC و MA-SAC	۸۸
۱۰-۷ مسیر طی‌شده فضایپیما با PPO استاندارد و نسخه بازی مجموع صفر MA-PPO	۹۰

- ۱۱-۷ مسیر و فرمان پیشران فضاییما در PPO استاندارد و نسخه بازی مجموع صفر MA-PPO . ۹۰
- ۱۲-۷ مقایسه توزیع پاداش تجمعی برای PPO و MA-PPO در سناریوهای مختلف. . . . . ۹۱
- ۱۳-۷ مقایسه توزیع پاداش تجمعی برای نسخه‌های تک‌عاملی در سناریوهای مختلف. . . . . ۹۳
- ۱۴-۷ مقایسه توزیع پاداش تجمعی برای الگوریتم‌ها در حالت چندعاملی در سناریوهای مختلف. . ۹۶

# فهرست الگوریتم‌ها

۲۸	گرادیان سیاست عمیق قطعی	۱
۳۱	عامل گرادیان سیاست عمیق قطعی تاخیری دوگانه	۲
۳۶	عامل عملگرد نقاد نرم	۳
۴۰	بهینه‌سازی سیاست مجاور (PPO-Clip)	۴
۶۱	عامل گرادیان سیاست عمیق قطعی چندعاملی	۵
۶۵	عامل گرادیان سیاست عمیق قطعی تاخیری دوگانه چندعاملی	۶
۷۰	عامل عملگر نقاد نرم چندعاملی	۷
۷۵	عامل بهینه‌سازی سیاست مجاور چندعاملی	۸

# فصل ۱

## مقدمه

در سال‌های آغازین عصر فضا، فرایند هدایت فضایی‌ها عمدهاً بر مبانی دینامیک کلاسیک و کنترل خطی استوار بوده است. با این حال، پیچیدگی روزافزون مأموریت‌های کنونی مانند سفرهای میان‌سیاره‌ای با پیشرانکم و شبکه‌های انبوه ماهواره‌ای در مدار زمین موجب دوچندان شدن ضرورت بهره‌گیری از روش‌های هوشمند و تطبیق‌پذیر شده است. در ادامه، انگیزه‌ی پژوهش در بخش ۱-۱ و تعریف دقیق مسئله در بخش ۲-۱ آمده است. سپس، مروری کوتاه بر مبانی یادگیری تقویتی و نسخه‌ی چندعاملی آن در بخش‌های ۳-۱ و ۴-۱ ارائه شده و در نهایت، ساختار کل گزارش در بخش ۶-۱ تشریح شده است.

## ۱-۱ انگیزه پژوهش

در دو دهه‌ی اخیر، به دلیل کوچک‌سازی سامانه‌ها، توسعه‌ی الکترونیک مقرن‌به‌صرفه و افزایش ظرفیت‌های پرتاب، تحولات بنیادینی در مأموریت‌های فضایی تجربه شده است. از پژوهش‌های علمی بین‌سیاره‌ای تا منظومه‌های انبوه ماهواره‌ای در مدارهای پایین زمین، مواجهه با چالش فراگیر هدایت بهینه در حضور عدم قطعیت‌ها به طور گسترده گزارش شده است. در مسیرهای فرا-قمری<sup>۱</sup> و به طور خاص در ناحیه‌های ناپایدار نقاط لاغرانژ در چارچوب مسئله‌ی سه‌جسمی کروی محدود دایروی<sup>۲</sup>، طراحی سامانه‌ی کنترل مستلزم تضمین هم‌زمان پایداری ایستا و بهره‌وری سوخت با پیشرانکم<sup>۳</sup> است.

هم‌راستا با این تحولات، ظهور و گسترش الگوریتم‌های یادگیری تقویتی عمیق<sup>۴</sup> امکانات نوینی برای طراحی

<sup>1</sup>Trans-lunar

<sup>2</sup>Circular Restricted Three-Body Problem (CRTBP)

<sup>3</sup>Low-thrust

<sup>4</sup>Deep Reinforcement Learning (DRL)

کنترل‌کننده‌های تطبیقی فراهم آورده است؛ با این حال، غالب رویکردهای رایج بر سناریوهای تک‌عاملی و اتکا به مدل‌های دینامیکی دقیق استوار شده‌اند. غیاب یک راهبرد مقاوم در برابر اغتشاشات مدل و تعییرات محیطی—از جمله خطای تراست پیش‌ران و تأخیر حسگر—به ایجاد فاصله‌ی معنادار میان عملکرد واقعی و پیش‌بینی‌های شبیه‌سازی ایده‌آل منجر شده‌است. در این پژوهش، این شکاف با بهره‌گیری از چارچوب یادگیری تقویتی چند‌عاملی مقاوم پُر می‌شود و اطمینان هدایت پیش‌ران کم در CRTBP ارتقا داده می‌شود. در ادامه، تعریف دقیق مسئله و سپس اهداف و نوآوری‌های پژوهش ارائه می‌شود.

## ۲-۱ تعریف مسئله

در سال‌های اخیر، پیشرفت‌های فناوری در کنترل پرواز، پردازش و هوش مصنوعی به گسترش کاربرد فضای‌پیماهای پیش‌ران کم در منظومه‌ی زمین-ماه انجامیده است؛ از تعقیب و انتقال مداری تا استقرار و نگهداری. روش‌های هدایت بھینه‌ی کلاسیک، هرچند قدرتمند، عموماً به ساده‌سازی‌های بسیار، منابع محاسباتی زیاد و شرایط اولیه‌ی مناسب متکی بوده‌اند؛ در مقابل، بخشی از این محدودیت‌ها با الگوریتم‌های مبتنی بر یادگیری تقویتی و تکیه بر تعامل و امکان محاسبات درون‌برد<sup>۵</sup> برطرف می‌شود.

هدف، طراحی سیاست کنترلی برای فضای‌پیمایی با جرم  $m$  در میدان گرانش سامانه‌ی زمین-ماه (مدل دو بعدی در چارچوب چرخان) است. ویژگی‌ها به اختصار:

- پویایی‌ها: معادلات حرکت در چارچوب مرجع چرخان به صورت  $\dot{x} = f(x) + g(x) a$  با  $x = [x, y, \dot{x}, \dot{y}]^\top$  و کنترل پیوسته‌ی  $a \in \mathcal{A}$  تعریف می‌شود، به‌طوری‌که کران  $|a| \leq a_{\max}$  برقرار است.
- عدم قطعیت‌ها: شرایط اولیه‌ی تصادفی، اغتشاش‌های عملگر، عدم تطابق مدل (در پارامترهای جرم)، مشاهده‌ی ناقص، نویز حسگر و تأخیر زمانی، که بر پایداری و کارایی اثرگذارند.
- صورت‌بندی بازی دیفرانسیلی (جمع‌صفر): فضای‌پیما و طبیعت (اغتشاشات) به‌ترتیب به عنوان عامل کنترل و حریف مزاحم مدل می‌شوند؛ با افق زمانی محدود  $t_f$ ، هدف، دستیابی به سیاستی مقاوم در برابر بدترین سناریو است.

صورت فشرده‌ی بھینه‌سازی به صورت کمینه-بیشینه است:

$$\min_{\pi} \max_{\omega} \mathbb{E}_{p, \pi, \omega} \left[ \sum_{t=0}^T r(s_t, a_t, \delta_t) \right], \quad (1-1)$$

---

<sup>5</sup>On-board Computing

که در آن، پاداش<sup>۶</sup>  $r$  به عنوان تابعی از مصرف ساخت، انحراف از مسیر یا مدار نامی و قیود مسئله تعريف می شود. خروجی مورد انتظار، سیاستی سبک و غیرمتراکز برای اجرای درون برد مدنظر است.

### ۳-۱ یادگیری تقویتی

یادگیری تقویتی<sup>۷</sup> شاخه ای از یادگیری ماشین است که در آن توالی اقدام ها  $a_t \in \mathcal{A}$  به گونه ای انتخاب می شود که بازده تجمعی آینده بیشینه شود. یک فرایند تصمیم گیری مارکوف<sup>۸</sup> به صورت  $\langle \mathcal{S}, \mathcal{A}, p, r, \gamma \rangle$  تعريف می شود که در آن:

- $\mathcal{S}$ : مجموعه حالت،

- $p(s'|s, a)$ : دینامیک انتقال،

- $r(s, a)$ : پاداش آنی،

- $\gamma \in [0, 1)$ : ضریب تنزیل.

سیاست<sup>۹</sup>  $\pi(a|s)$  به عنوان احتمال انتخاب اقدام  $a$  در وضعیت  $s$  بیان می شود. هدف، بیشینه سازی برگشت<sup>۱۰</sup> است:

$$G_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k} \quad (2-1)$$

روش های RL معمولاً در دو دسته ای ارزش محور (مانند Q-learning و DQN) و سیاست محور (مانند Reinforce) جای می گیرند؛ ترکیب این دو به چارچوب Actor-Critic منتهی می شود که در آن، یک بازیگر (Actor) سیاست را به روزرسانی می کند و یک منتقد (Critic) ارزش یا Q برآورد می شود [۱].

در حضور فضاهای پیوسته حالت-عمل، الگوریتم های G，TD3，DDPG，SAC، PPO با تکیه بر شبکه های عصبی به عنوان تقریبگر توابع، کارایی بالای نشان داده اند. در این پژوهش، خانواده ای Actor-Critic به عنوان پایه ای توسعه کنترل کننده پیشنهاد شده است و در ادامه، به نسخه ی چند عاملی آن در بخش ۴-۱ پیوند داده می شود.

<sup>6</sup>Reinforcement Learning (RL)

<sup>7</sup>Markov Decision Process (MDP)

<sup>8</sup>Policy

<sup>9</sup>Return

## ۴-۱ یادگیری تقویتی چندعاملی

در یادگیری تقویتی چندعاملی<sup>۱۰</sup>، فضای تصمیم‌گیری به صورت یک بازی مارکفی<sup>۱۱</sup> با مجموعه‌ی عامل‌ها =  $N$  {1, ...,  $N$ } مدل شده که در آن هر عامل با سیاست  $\pi$  به دنبال بیشینه‌سازی بازده تجمعی خود است. در سناریوهای رقابتی دونفره‌ی جمع‌صفر<sup>۱۲</sup>، مفهوم تعادل نش<sup>۱۳</sup> بعنوان معیار پایداری سیاست‌ها در نظر گرفته می‌شود.

رویکرد آموزش‌متمرکز، اجرای توزیع شده<sup>۱۴</sup> با جدا کردن مرحله‌ی آموزش که در آن اطلاعات خصوصی‌همه‌ی عامل‌ها برای منتقدها در دسترس است و مرحله‌ی اجرا در آن هر عامل صرفاً بر مشاهده‌ی محلی اتكا می‌کند که باعث تعادل میان کارایی، مقیاس‌پذیری و هزینه‌ی ارتباطی برقرار شده است.

در این پایان‌نامه، یک صورت‌بندی دوعلاملی جمع‌صفر اتخاذ شده است که در آن سیاست هدایت توسط عامل‌کنترل آموخته می‌شود و اغتشاشات یا نامعینی‌ها توسط عامل‌مزاحم مدل‌سازی می‌شوند تا سیاستی مقاوم حاصل شود.

• DDPG: الگوریتم مبتنی بر گرادیان سیاست قطعی برای فضاهای کنش‌پیوسته،

• TD3: نسخه‌ی بهبودیافته‌ی DDPG با برآورد دوسویه‌ی  $\varphi$  برای کاهش تورش بیش‌برآورده<sup>۱۵</sup>،

• PPO: الگوریتم سیاست احتمالی پایدار با قیود نسبت احتمال و بهبود تدریجی سیاست،

• SAC: الگوریتم حداقل‌کردن آنتروپی که تعادل میان بهره‌برداری و اکتشاف به طور ذاتی برقرار می‌شود.

تابع پاداش طراحی شده، مصالحه‌ی سوخت، انحراف و قیود منعکس می‌شود و مبنایی برای ارزیابی کیفیت سیاست‌های مختلف فراهم می‌گردد.

## ۵-۱ ارزش افزوده و نوآوری‌های پژوهش

نوآوری‌های اصلی این پژوهش را می‌توان به صورت زیر خلاصه نمود:

<sup>10</sup>Multi-Agent Reinforcement Learning (MARL)

<sup>11</sup>Markov Games (MG)

<sup>12</sup>Zero-Sum

<sup>13</sup>Nash Equilibrium

<sup>14</sup>Centralized Training with Decentralized Execution (CTDE)

<sup>15</sup>Overestimation Bias

۱. صورت‌بندی هدایت کمتر است در CRTBP به صورت یک بازی دیفرانسیلی مجموع صفر میان فضای پیما (عامل کنترل) و مزاحمت‌های محیطی (عامل حریف)؛
۲. توسعه نسخه‌های چند‌عاملی از الگوریتم‌های DDPG، TD3، SAC و PPO در چارچوب آموزش متمرکز و اجرای غیرمتمرکز (CTDE) و انطباق آن با دینامیک چندجسمی؛
۳. طراحی تابع پاداش مقاوم شامل مولفه‌های انحراف مسیر، هزینه سوخت، خطای تراستر، تأخیر حسگر، نویز عملگر و عدم تطابق مدل دینامیکی؛
۴. ارائه نتایج شبیه‌سازی گستره و نشان‌دادن بھبود معنادار در پایداری مسیرهای تغیر مدارهای لیاپانوف نقاط لاغرانژ، کاهش مصرف سوخت و کاهش انحراف مداری نسبت به نسخه‌های تک‌عاملی و روش‌های کلاسیک؛
۵. فراهم‌سازی چارچوبی که بدون نیاز به مدل دقیق دینامیکی، قابلیت هدایت مقاوم، تطبیقی و مناسب برای پیاده‌سازی درون‌برد را ارائه می‌کند.

## ۶-۱ ساختار گزارش

در فصل ۲ مروی انتقادی بر کارهای مرتبط در هدایت پیش‌رانکم و یادگیری تقویتی تک‌عاملی و چند‌عاملی ارائه می‌شود. در فصل ۳ مدل‌سازی محیط آزمایش بر پایه‌ی CRTBP ارائه می‌گردد. فصل ۴ به مبانی یادگیری تقویتی اختصاص دارد و الگوریتم‌های DDPG، TD3، SAC و PPO در بخش‌های ۲-۴ تا ۵-۶ مروور می‌شوند. در فصل ۵ طراحی شبیه‌سازی شامل تعریف عامل‌ها، فضای حالت، عمل و تابع پاداش توضیح داده می‌شود. در فصل ۶ چارچوب یادگیری تقویتی چند‌عاملی تشریح شده و پیوند آن با بازی‌های جمع‌صفر و تعادل نش در بخش ۲-۶ بیان می‌شود. در فصل ۷ نتایج و مقایسه با معیارهای مرجع ارائه می‌شود. در نهایت، فصل ۸ جمع‌بندی دستاوردها و پیشنهادهای پژوهش‌های آینده را ارائه می‌دهد.

## فصل ۲

### پیشینه پژوهش

این فصل تصویری منسجم از ادبیات مأموریت‌های بین‌مداری، مبانی یادگیری تقویتی و یادگیری تقویتی چندعاملی ارائه می‌کند. تمرکز بر تبیین مفاهیم کلیدی، چالش‌های رایج، و روندهای پژوهشی مؤثر برای طراحی و هدایت درون‌سفینه‌ای است؛ بهگونه‌ای که زمینه‌ی نظری لازم برای روش‌ها و مسائل مورد استفاده در ادامه‌ی پژوهش فراهم شود. ساختار فصل به این صورت است که ابتدا در بخش ۱-۲ مروی بر روش‌های طراحی مسیر و هدایت درون‌سفینه‌ای و کاربردهای یادگیری ماشین ارائه می‌شود، سپس در بخش ۲-۲ مبانی و الگوریتم‌های اصلی یادگیری تقویتی و خطوط پژوهشی مرتبط مرور می‌گردد و در پایان بخش ۳-۲ به رویکردها و چالش‌های یادگیری تقویتی چندعاملی پرداخته می‌شود.

#### ۱-۲ مأموریت‌های بین‌مداری

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

هدایت فضایپیماها معمولاً پیش از پرواز انجام می‌شود. این روش‌ها می‌توانند از تکنیک‌های بهینه‌سازی فرآگیر [۲] یا برنامه‌نویسی غیرخطی برای تولید مسیرها و فرمان‌های کنترلی بهینه استفاده کنند. با این حال، این روش‌ها معمولاً حجم محاسباتی زیادی دارند و برای استفاده درون‌سفینه‌ای نامناسب هستند [۳]. یادگیری ماشین می‌تواند برای بهبود قابلیت‌های هدایت فضایپیماها 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 فضایی‌ها به‌طور کلی به راه حل‌های 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 [۴، ۵] ایجاد می‌کنند.

شبکه‌های عصبی قابلیت‌های منحصر به فردی برای انجام هدایت در فضایی‌ما دارند. به عنوان مثال، شبکه‌های عصبی می‌توانند به‌طور مستقیم از تخمین‌های وضعیت به دستورهای پیش‌ران کنترلی که با محدودیت‌های مأموریت سازگار است، برسند. عملکرد مناسب هدایت شبکه‌های عصبی در مطالعاتی مانند فرود بر سیارات [۶]، عملیات نزدیکی به سیارات [۷] و کنترل فضایی‌ما با پیش‌ران از دست‌رفته [۸] 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. به عنوان مثال، الگوریتم‌های یادگیری ماشین ابتدایی AEGIS در فضایی‌ماهی مریخی نورد برای کمک به شناسایی ویژگی‌های زمین‌شناسی تعییه شده‌اند. الگوریتم Curiosity توپایی انتخاب خودکار هدف توسط یک دوربین در داخل فضایی‌ماهی Spirit، Opportunity و را دارد [۹]. در کامپیوتر پرواز اصلی، فرآیند دقت افزایی<sup>۱</sup> نیاز به ۹۴ تا ۹۶ ثانیه دارد [۱۰]، که به‌طور 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 فضایی‌ماهی Europa Clipper [۱۲، ۱۱] و شناسایی عیوب برای Perseverance [۱۳] می‌شود. الگوریتم‌های یادگیری ماشین پتانسیل انجام نقش مهمی در مأموریت‌های خودکار آینده را 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 است. به عنوان مثال، از یک شبکه عصبی<sup>۲</sup> در بهینه‌سازی مسیرهای رانشگر کم‌پیش‌ران استفاده شده است [۱۴]. پژوهش‌های جدید شامل شناسایی انتقال‌های هتروکلینیک [۱۵]، اصلاح مسیر رانشگر کم‌پیش‌ران [۱۶] و تجزیه و تحلیل مشکلات از دست‌رفتن رانشگر [۸] می‌شود.

تکنیک‌های یادگیری نظرارتی 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.

<sup>1</sup>Refinement Process

<sup>2</sup>Neural Network

در سال‌های اخیر، قابلیت یادگیری تقویتی<sup>۳</sup> در دستیابی به عملکرد بهینه در بخش‌هایی با ابهام محیطی قابل توجه، به اثبات رسیده است [۱۷، ۱۸]. هدایت انجام شده توسط یادگیری تقویتی را می‌توان به صورت گسترش بر اساس فاز پرواز دسته‌بندی کرد. مسائل فرود [۱۹، ۲۰] و عملیات در نزدیکی اجسام کوچک [۲۶، ۷]، از حوزه‌های پژوهشی هستند که از یادگیری تقویتی استفاده می‌کنند. تحقیقات دیگر شامل مواجهه تداخل خارجی جوی [۲۱]، نگهداری ایستگاهی [۲۲] و هدایت به صورت جلوگیری از شناسایی [۲۳] است. مطالعاتی که فضای‌پیماهای رانشگر کم‌پیشران را در یک چارچوب دینامیکی چندبُدنی با استفاده از یادگیری تقویتی انجام شده‌است، شامل طراحی انتقال با استفاده از Q-learning [۲۴] Proximal Policy Optimization، [۲۵] و هدایت نزدیکی مدار [۲۶] است.

## ۲-۲ یادگیری تقویتی

از نخستین صورت‌بندی‌های فرایند تصمیم‌گیری مارکفی در یادگیری تقویتی، پژوهش بر آن بوده‌است که عامل بتواند با اجرای عمل‌ها و دریافت پاداش، سیاستی برای بیشینه‌سازی بازگشت بیاموزد. تبیین جامع این چارچوب و الگوریتم‌های بنیادین در کتاب سوتون و بارتون بهمثابه مرجع کلاسیک این حوزه ارائه شده و همچنان مبنای بسیاری از آثار معاصر است [۱].

دهه‌ی ۱۹۹۰ میلادی شاهد شکل‌گیری روش‌هایی بر پایه‌ی ارزش<sup>۴</sup> نظری Q-learning و نخستین رویکردهای گرادیان سیاست بود؛ با وجود این، محدودیت توان محاسباتی و فقدان داده‌ی فراوان، سرعت رشد را کند می‌کرد. ورود شبکه‌های عصبی عمیق<sup>۵</sup> نقطه‌ی عطفی بود: مقاله‌ی معروف دیپ‌ماینده<sup>۶</sup> نشان داد که شبکه‌ی Q عمیق<sup>۷</sup> می‌تواند صرفاً از پیکسل‌های بازی آتاری سیاستی نزدیک به انسان بیاموزد [۲۷].

موفقیت DQN نگاه‌ها را به‌سوی گرادیان سیاست مقیاس‌بندی معطوف ساخت. بهینه‌سازی ناحیه‌ی اطمینان<sup>۸</sup> تضمین بهبود یکنواخت سیاست را فراهم کرد [۲۸] و روش A3C با موازی‌سازی بازیگران، سرعت یادگیری را چند برابر افزایش داد [۲۹]. کمی بعد، DDPG اولین بار گرادیان سیاست قطعی را به فضاهای عمل پیوسته وارد کرد [۳۰]. سپس PPO با ساده‌سازی قیود TRPO و کاهش فرآپارامترهای حساس، به انتخاب پیش‌فرض بسیاری از کاربردهای مهندسی بدل شد [۳۱].

با گسترش دامنه‌ی مسائل، پایداری و کارایی داده به چالش اصلی بدل گشت. TD3 نشان داد که کمینه‌کردن

<sup>3</sup>Reinforcement Learning (RL)

<sup>4</sup>Value

<sup>5</sup>Deep Neural Network (DNN)

<sup>6</sup>DeepMind

<sup>7</sup>Deep Q Network (DQN)

<sup>8</sup>Trust Region Policy Optimization (TRPO)

میان دو منتقد می‌تواند برآورد بیش از حد Q را مهار کند [۳۲]، و SAC با افودن بند آنتروپی، همزمان اکتشاف و بازده را بهبود داد [۳۳].

در محیط‌های پرخطر یا گران، جمع‌آوری داده‌ی برخط ناممکن است؛ از این‌رو یادگیری تقویتی غیربرخط مطرح شد. روش CQL با برقراری کران محافظه‌کارانه بر Q-value از گرایش خارج از توزیع جلوگیری می‌کند [۳۴] و مرور اخیر پراودنسیو و همکاران طبقه‌بندی جامعی از چالش‌های باز این حوزه ارائه داده است [۳۵].

هم‌زمان، دغدغه‌ی ایمنی و مقاومت در سامانه‌های واقعی پرنگ شد. مرور سال ۲۰۲۲ نشان می‌دهد که ترکیب قیدهای سخت، توابع جریمه‌ی ریسک و شبیه‌سازی محیط‌های بدینانه سه خط اصلی ایمنی در یادگیری تقویتی هستند [۳۶]. سلسله‌مراتب نیز با هدف انتقال دانش و تسريع یادگیری مورد توجه قرار گرفت و یک مطالعه‌ی جامع در ACM Computing Surveys چهار چالش کشف زیرکار، یادگیری اشتراک‌پذیر، انتقال و مقیاس‌پذیری را برجسته می‌کند [۳۷].

وقتی چند عامل به‌طور همزمان یاد می‌گیرند، پویایی محیط از دید هر عامل غیرایستا می‌شود. مرور جامع ۲۰۲۴ نشان می‌دهد که چارچوب ناظر متمرکز - بازیگر توزیع شده<sup>۹</sup> راهکاری موثر برای این چالش است و مباحثی چون تخصیص اعتبار جمعی و کشف تعادل را معرفی می‌کند [۳۸].

پیشرفت‌های یادشده در نهایت به دستاوردهای نمادینی چون AlphaGo [۳۹] و AlphaStar [۴۰] انجامیدند که در بازی‌های Go و StarCraft II از انسان پیشی گرفتند و معما ری توزیع شده‌ی IMPALA نشان داد که چگونه می‌توان هزاران شبیه‌ساز را با بهروزرسانی وزن‌های مهم ادغام کرد [۴۱].

به رغم این جهش‌ها، سه شکاف اساسی پابرجا مانده است: ۱) تضمین ایمنی سخت‌گیرانه در سناریوهای نزدیک‌برخورد، ۲) کاهش وابستگی به داده‌ی پرهزینه یا نایاب از طریق روش‌های مدل‌بنا و غیربرخط و ۳) مقیاس‌پذیری یادگیری چندعاملی برای سامانه‌های رباتیکی یا فضایی‌پیمای چندگانه.

## ۳-۲ پیشنهای پژوهش یادگیری تقویتی چندعاملی

امروز یادگیری تقویتی چندعاملی<sup>۱۰</sup> به عنوان بنیاد اصلی سامانه‌های هوشمند مشارکتی شناخته می‌شود؛ مسیری که از آزمون‌های ساده‌ی دوعاملی در دهه ۱۹۹۰ آغاز شد و اکنون به معما ری‌های توزیع شده‌ی در مقیاس هزاران بازیگر رسیده است. این بخش، به بررسی اینکه چگونه ایده‌ی آموزش متمرکز - اجرای توزیع شده (CTDE) به پاسخ غالب برای چالش‌های غیرایستایی و انفجار بُعدی<sup>۱۱</sup> بدل شد و چه گام‌هایی هنوز برای ایمنی، ناهمگونی

<sup>9</sup>Centralized Training with Decentralized Execution (CTDE)

<sup>10</sup>Multi-Agent Reinforcement Learning (MARL)

<sup>11</sup>Curse of Dimensionality

و مقیاس پذیری باقی مانده است.

دهه‌ی ۱۹۹۰ با مقاله‌ی [۴۲] آغاز شد؛ جایی که برای نخستین بار مقایسه‌ی عامل‌های مستقل با عامل‌های همکار انجام شد و سود ارتباط و اشتراک تجربه به صورت تجربی نشان داده شد. در میانه‌ی دهه‌ی بعد، مور جامع پانایت و لوك [۴۳] چشم‌اندازی از مسائل تخصیص اعتبار و غیرایستایی ترسیم کرد و دو موضوع یادگیری تیمی و یادگیری هم‌زمان را صورت‌بندی نمود. هم‌زمان، بوشونیو و همکاران [۴۴] ادبیات MARL را در قالب اهداف پایداری دینامیک یادگیری و انطباق با رفتار سایر عامل‌ها جمع‌بندی کردند و راه را برای تحلیل‌های بازی‌محور هموار ساختند.

ورود شبکه‌های عمیق در سال‌های ۲۰۱۶ و ۲۰۱۷ نقطه‌ی عطف بعدی بود؛ منتقد مرکز - بازیگر توزیع شده در MA-DDPG [۴۵] نشان داد که می‌توان از حالت سراسری در فاز آموزش بهره برد، اما سیاست نهایی را صرفاً بر اساس مشاهدات محلی اجرا کرد. در همان سال، Value-Decomposition Networks [۴۶] ایده‌ی تجزیه‌ی خطی پاداش را برای همکاری عامل‌ها مطرح کرد و راه را برای تقسیم بندی‌های پیش‌رفته پاداش گشود. ۲۰۱۸ شاهد جهش مهمی با QMIX بود؛ این روش با اعمال قید تک‌نوا<sup>۱۲</sup> بر ترکیب مقادیر منفرد، هم امکان بهینه‌سازی غیرسیاست‌محور را فراهم کرد و هم تضمین سازگاری سیاست‌های محلی با ارزش مشترک را برقرار ساخت [۴۷].

سال ۲۰۱۹ به گسترش بسترهای آزمایش اختصاص یافت. چالش استاندارد StarCraft Multi-Agent Challenge (SMAC) بر مبنای StarCraft II معرفی شد و معیار مشترکی برای مقایسه‌ی الگوریتم‌ها را مهیا کرد [۴۸]. هم‌زمان، QTRAN [۴۹] نشان داد که می‌توان بدون قید خطی یا تک‌نوا، تابع ارزش مشترک را به فضای قابل تجزیه تبدیل کرد. از سوی دیگر، MAVEN با افزودن متغیر نهفته‌ی مشترک، کاوش هماهنگ و سلسله‌مراتبی را امکان‌پذیر ساخت [۵۰]. نقطه‌ی اوج همان سال، سامانه‌ی AlphaStar بود که نشان داد ترکیب خودبازی و معماری توزیع شده می‌تواند به رتبه‌ی استاد بزرگ<sup>۱۳</sup> انسان برساند [۴۰].

در ۲۰۲۰ مفهوم نقش‌های در حال ظهور با ROMA [۵۱] معرفی شد تا عامل‌ها بر اساس شbahت رفتاری به طور خودکار خوش‌بندی و اشتراک دانش کنند؛ رویکردی که در نقشه‌های پرترکم SMAC برتری محسوسی نشان داد. پژوهش‌های متا در ۲۰۲۱، از مرور نظری زانگ و بشار [۵۲] تا محک<sup>۱۴</sup> تطبیقی پاپوادکیس و همکاران [۵۳]، شکاف‌های باقی‌مانده در تضمین همگرایی و مقیاس را فهرست کردند.

آخرین موج مطالعات بر ناهمگونی و اینمنی مرکز دارد. مروج جامع [۵۴] نشان می‌دهد که تفاوت در قابلیت‌ها و اطلاعات عامل‌ها، مسائلی نظیر تخصیص اعتبار و تعادل را پیچیده‌تر می‌سازد و به الگوریتم‌های سازگار با نقش‌های پویا نیاز دارد.

<sup>12</sup>Monotonic

<sup>13</sup>Grandmaster

<sup>14</sup>Benchmark

به طور خلاصه، مسیر تاریخی MARL از الگوهای مستقل دهه‌ی ۱۹۹۰ به سامانه‌های توزیع شده‌ی امروزی، همواره با سه دغدغه‌ی اصلی هدایت شده است: کنترل انفجار بُعدی توابع ارزش، مقابله با غیرایستایی ناشی از یادگیری هزمان، و انتقال مؤثر تجربه میان عامل‌ها. علی‌رغم پیشرفت‌های شتابان، تضمین این‌نی سخت‌گیرانه در محیط‌های شکست‌پذیر، مدیریت نقش‌های پویا در تیم‌های ناهمگون و کاهش نیاز به داده‌ی شبیه‌سازی پرهزینه همچنان چالش‌های باز باقی می‌مانند؛ چالش‌هایی که در این پژوهش با رویکرد ترکیبی مدل‌مبنای مقاوم و چند‌عاملی پیگیری می‌شوند.

## ۴-۲ جمع‌بندی و گپ پژوهشی

مرور ادبیات پژوهش نشان داد که هدایت فضایی‌های کمتراست در سامانه‌های دینامیکی چندجسمی، به‌ویژه در چارچوب مسئله‌ی محدود سه‌جسمی (CRTBP)، عمداً مبتنی بر روش‌های کلاسیک کنترل بهینه و تکنیک‌های مدل‌مبنای بوده است [۳-۵]. هرچند این رویکردها در شرایط اسمی عملکرد مطلوبی دارند، اما اتکای شدید آن‌ها به مدل‌سازی دقیق دینامیکی و حساسیت نسبت به عدم قطعیت‌های محیطی، موجب کاهش کارایی و پایداری سامانه در حضور اغتشاشات حسگر، خطای عملگر، تأخیر زمانی و عدم تطابق مدلی می‌گردد.

از سوی دیگر، روش‌های یادگیری تقویتی عمیق طی سال‌های اخیر برای هدایت کمتراست در محیط‌های مدارپویای سیاره‌ای پیشنهاد شده‌اند [۲۵، ۲۶، ۵۵]. با وجود پیشرفت‌های قابل توجه، بررسی منابع موجود نشان می‌دهد که:

۱. اغلب پژوهش‌ها مبتنی بر چارچوب RL تک‌عاملی بوده و تعامل فضایی‌با مزاحمت‌های محیطی به صورت یک بازی دیفرانسیلی مجموع صفر مدل نشده است؛
۲. در عمده‌ی مطالعات، فرض مشاهده‌ی کامل و بدون نویز از حالت‌ها در نظر گرفته شده و پدیده‌های واقعی نظیر تأخیر حسگر، اغتشاشات تصادفی و تطابق‌ناپذیری مدل لحاظ نشده‌اند؛
۳. کمتر پژوهشی نسخه‌های DDPG، TD3، SAC و PPO را در قالب یادگیری تقویتی چند‌عاملی با آموزش متمرکز و اجرای غیرمتمرکز (CTDE) برای هدایت فضایی در CRTBP توسعه داده است.

بنابراین، گپ پژوهشی اصلی این رساله را می‌توان به صورت زیر بیان کرد:

هنوز چارچوبی مقاوم، مبتنی بر یادگیری تقویتی چند‌عاملی مجموع صفر، که توانایی هدایت فضایی‌های کمتر است را در محیط‌های ناپایدار CRTBP با حضور گستردگی عدم قطعیت‌های محیطی تضمین نماید، ارائه نشده است.

هدف پژوهش حاضر، طراحی و ارزیابی چنین چارچوبی و بررسی مزایای آن نسبت به نسخه‌های تک‌عاملی و روش‌های کلاسیک هدایت می‌باشد.

## فصل ۳

# مدل‌سازی محیط یادگیری سه جسمی

مسیرهای فضایی در بسیاری از مأموریت‌ها نه تنها تحت تأثیر گرانش یک جسم مرکزی (مانند خورشید یا زمین)، بلکه به طور همزمان تحت نفوذ دست‌کم یک جسم دیگر نیز هستند. در این وضعیت، مدل‌های دو‌جسمی با اختلالات جسم سوم دقت کافی ندارند و باید دینامیک دو جسم اصلی و اثرات آن‌ها به صورت همزمان در نظر گرفته شود. مسئله‌ی سه‌جسمی محدود با دو جسم اصلی و یک جسم سوم با جرم ناقیز (فضاپیما) چارچوبی طبیعی برای مطالعه‌ی این پدیده‌ها و نیز یک محیط مناسب برای به کارگیری روش‌های یادگیری تقویتی است؛ زیرا دینامیک غیرخطی و پیچیده‌ی آن ویژگی‌های غنی (مانند نقاط تعادل لاغرانژ) ایجاد می‌کند.

در این فصل ابتدا در بخش ۱-۳ دستگاه‌بی‌بعد و چارچوب چرخان تعریف شده و در بخش ۲-۳ معادلات حرکت در مسئله‌ی سه‌جسمی محدود دایره‌ای استخراج می‌شود.

### ۱-۳ مسئله‌ی سه‌جسمی محدود دایره‌ای (CRTBP)

دو جرم اصلی (زمین با جرم  $m_1$  و ماه با جرم  $m_2$ ) روی مدارهایی دایره‌ای و هم‌صفحه پیرامون مرکز جرم مشترک حرکت می‌کنند. جرم سوم (فضاپیما با جرم ناقیز  $m_3$ ) چنان کوچک فرض می‌شود که تأثیر گرانشی آن بر حرکت دو جسم اصلی قابل صرف نظر است؛ بدین ترتیب، مسئله‌ی سه‌جسمی محدود دایره‌ای شکل می‌گیرد.

جدول ۱-۳ : مقادیر عددی برای مسئله سه‌جسمی محدود (سامانه زمین-ماه)

مقدار عددی	توصیف	پارامتر
$5.972 \times 10^{24} \text{ kg}$	جرم زمین	$m_1$
$7.348 \times 10^{22} \text{ kg}$	جرم ماه	$m_2$
0.0121505856	نسبت جرمی	$\mu$
$2.6617 \times 10^{-6} \text{ rad/s}$	سرعت زاویه‌ای سامانه	$\omega$

دستگاه مختصات چرخانی همدوران با دو جرم اصلی انتخاب می‌شود؛ مبدأ در مرکز جرم سامانه است، محور  $x$  خطِ واصل دو جرم و محور  $y$  بر آن عمود (در صفحه‌ی مدارها) است. واحد طول برابر فاصله‌ی ثابت میان دو جرم و واحد زمان چنان تعریف می‌شود که دوره‌ی مداری سامانه  $2\pi$  (و در نتیجه  $1 = \omega$ ) گردد. همچنین جرم‌ها به‌گونه‌ای مقیاس می‌شود که مجموع دو جرم برابر با یک شود:

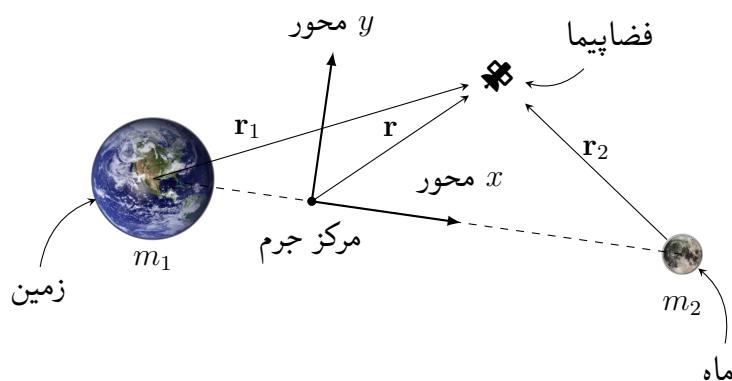
$$m_1 + m_2 = 1. \quad (1-3)$$

با نسبت جرمی

$$\mu \equiv \frac{m_2}{m_1 + m_2}, \quad (2-3)$$

داریم  $m_2 = \mu$  و  $m_1 = 1 - \mu$  و مکان دو جرم در دستگاه بی‌بعد به صورت

$$\mathbf{r}_{\text{Earth}} = (-\mu, 0), \quad \mathbf{r}_{\text{Moon}} = (1 - \mu, 0). \quad (3-3)$$



شکل ۱-۳ : هندسه‌ی مسئله سه‌جسمی محدود در چارچوب چرخان

## ۱-۱-۳ لاگرانژ و معادلات حرکت

با در نظر گرفتن  $G = 1$  در حالت بی بعد، تابع لاگرانژ جرم سوم در دستگاه چرخان برابر است با [۵۶]

$$L = \frac{1}{2}(\dot{x}^2 + \dot{y}^2 + \dot{z}^2) + (1 - \mu) \frac{1}{r_1} + \mu \frac{1}{r_2} + \frac{1}{2}(x^2 + y^2), \quad (4-3)$$

که در آن

$$r_1 = \sqrt{(x + \mu)^2 + y^2 + z^2}, \quad r_2 = \sqrt{(x - 1 + \mu)^2 + y^2 + z^2}. \quad (5-3)$$

با به کارگیری رابطه‌ی اویلر-لاگرانژ

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = 0, \quad q_i \in \{x, y, z\},$$

معادلات بی بعد حرکت جرم سوم به دست می‌آید:

$$\ddot{x} - 2\dot{y} = x - \frac{1 - \mu}{r_1^3}(x + \mu) - \frac{\mu}{r_2^3}(x - 1 + \mu), \quad (6-3)$$

$$\ddot{y} + 2\dot{x} = y - \frac{1 - \mu}{r_1^3}y - \frac{\mu}{r_2^3}y, \quad (7-3)$$

$$\ddot{z} = -\frac{1 - \mu}{r_1^3}z - \frac{\mu}{r_2^3}z. \quad (8-3)$$

یا به نگاشت برداری به صورت زیر است.

$$\ddot{\mathbf{r}} + 2\boldsymbol{\omega} \times \dot{\mathbf{r}} = \nabla \Omega(\mathbf{r}), \quad \Omega(x, y, z) = \frac{1}{2}(x^2 + y^2) + \frac{1 - \mu}{r_1} + \frac{\mu}{r_2}. \quad (9-3)$$

که در آن  $\Omega$  پتانسیل مؤثر است و در بخش ۲-۳ برای یافتن نقاط تعادل از شرط  $\nabla \Omega = 0$  استفاده می‌شود.

## ۲-۳ نقاط تعادل لاگرانژ

نقطه‌ی تعادل مکانی است که در چارچوب چرخان، جرم سوم بی حرکت می‌ماند. این شرط با صفر شدن مؤلفه‌های سرعت و شتاب حاصل می‌شود؛ از این‌رو در معادلات حرکت بخش ۱-۳ قرار می‌دهیم  $\dot{x} = \dot{y} = \dot{z} = \ddot{x} = \ddot{y} = \ddot{z} = 0$ . در نتیجه دستگاه جبری زیر برای مختصات نقطه‌ی تعادل به دست می‌آید:

$$0 = x - \frac{1 - \mu}{r_1^3}(x + \mu) - \frac{\mu}{r_2^3}(x - 1 + \mu), \quad (10-3)$$

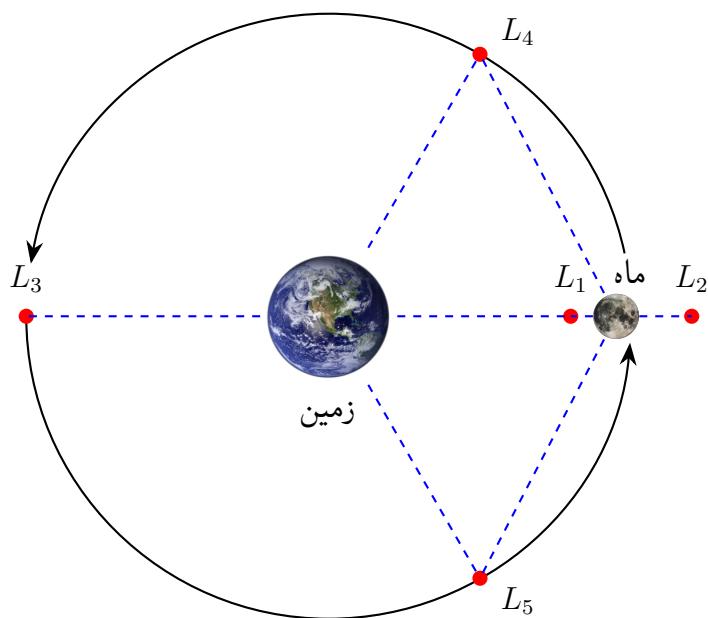
$$0 = y \left[ 1 - \frac{1 - \mu}{r_1^3} - \frac{\mu}{r_2^3} \right], \quad (11-3)$$

$$0 = -\frac{1 - \mu}{r_1^3}z - \frac{\mu}{r_2^3}z. \quad (12-3)$$

معادله‌ی سوم نشان می‌دهد در حالت عمومی باید  $z = 0$  باشد؛ بنابراین، نقاط تعادل همگی در صفحه‌ی مدار قرار می‌گیرند.

### دسته‌بندی کلی

۱. نقاط هم خط<sup>۱</sup> : سه نقطه‌ی  $L_1, L_2$  و  $L_3$  روی خطِ واصلِ دو جرم قرار دارند و لذا  $y = 0$  است.
۲. نقاط سه‌گوش<sup>۲</sup> : دو نقطه‌ی  $L_4$  و  $L_5$  رأس‌های مثلث متساوی‌الاضلاع با دو جرم اصلی را تشکیل می‌دهند و در آن‌ها  $y \neq 0$ .



شکل ۲-۳: نقاط لاغرانژ در سامانه‌ی زمین-ماه

### نقاط هم خط ( $L_1, L_2, L_3$ )

با اعمال  $y = 0$ ، تنها معادله‌ی زیر باقی می‌ماند:

$$x - \frac{1-\mu}{|x+\mu|^3}(x+\mu) - \frac{\mu}{|x-1+\mu|^3}(x-1+\mu) = 0. \quad (13-3)$$

این معادله در سه ناحیه‌ی مجزا—بین دو جرم، بیرون‌جرم کوچک و بیرون‌جرم بزرگ—دارای یک ریشه است که به ترتیب نقاط  $L_1, L_2$  و  $L_3$  را تعیین می‌کند.

---

<sup>1</sup>Collinear

<sup>2</sup>Triangular

برای  $1 \ll \mu$  (همچون سامانه‌ی خورشید-زمین یا زمین-ماه) می‌توان تقریب‌های شناخته‌شده را نوشت:

$$x_{L_1} \simeq (1 - \mu) - \left(\frac{\mu}{3}\right)^{1/3},$$

$$x_{L_2} \simeq (1 - \mu) + \left(\frac{\mu}{3}\right)^{1/3},$$

$$x_{L_3} \simeq -1 - \frac{5}{12}\mu; \quad y_{L_i} = 0.$$

در عمل، ریشه‌ی دقیق معادله‌ی (۱۳-۳) با یک روش عددی (نیوتون-رافسون) محاسبه می‌شود.

نقاط سه‌گوش ( $L_4, L_5$ )

در این نقاط  $1 - (1 - \mu)/r_1^3 - \mu/r_2^3 = 0$  و شرط  $r_1 = r_2 = 1$  به‌طور طبیعی برقرار است. مختصات به عبارت‌اند از

$$x_{L_4} = x_{L_5} = \frac{1}{2} - \mu, \quad y_{L_4} = +\frac{\sqrt{3}}{2}, \quad y_{L_5} = -\frac{\sqrt{3}}{2}. \quad (14-3)$$

پایداری این نقاط مستلزم نسبت جرم کافی است؛ شرط کلاسیک ( $m_1/m_2 > 24.96$  یا معادل آن  $\mu_R \approx 0.03852$ ) در سامانه‌های خورشید-سیاره یا زمین-ماه برقرار است و سبب وجود خانواده‌ی سیارک‌های تروجان حول  $L_4$  و  $L_5$  می‌شود. در مقابل، نقاط هم خط ناپایدارند و معمولاً مأموریت‌های فضایی روی مدارهای هاله‌ای یا لیساژور در پیرامون آن‌ها قرار می‌گیرند.

برای سامانه‌ی زمین-ماه،  $\mu \approx 0.01215$  است. جدول زیر مختصات بی‌بعد هر پنج نقطه را نشان می‌دهد (واحد طول: فاصله‌ی زمین-ماه). موقعیت زمین در  $(-\mu, 0)$  و ماه در  $(0, 0)$  است.

جدول ۲-۳: مقادیر عددی نقاط لاغرانژ برای مسئله‌ی سه‌جسمی محدود سیستم زمین-ماه

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

این نتایج نشان می‌دهد که  $L_1$  در حدود 0.84 فاصله‌ی زمین–ماه از زمین قرار دارد (فاصله‌ی آن تا ماه در حدود 0.16 واحد طول است) و  $L_2$  بیرونِ مدارِ ماه است. نقطه‌ی  $L_3$  تقریباً یک واحد طول در سوی مقابلِ ماه نسبت به زمین قرار دارد. دو نقطه‌ی  $L_4$  و  $L_5$  در مختصات (0.488,  $\pm 0.866$ ) قرار گرفته و با زمین و ماه مثلث متساوی‌الاضلاع می‌سازند.

این نتایج نشان می‌دهد که  $L_1$  در حدود 0.84 فاصله‌ی زمین–ماه از زمین قرار دارد (فاصله‌ی آن تا ماه در حدود 0.16 واحد طول است) و  $L_2$  بیرونِ مدارِ ماه است. نقطه‌ی  $L_3$  تقریباً یک واحد طول در سوی مقابلِ ماه نسبت به زمین قرار دارد. دو نقطه‌ی  $L_4$  و  $L_5$  در مختصات (0.488,  $\pm 0.866$ ) قرار گرفته و با زمین و ماه مثلث متساوی‌الاضلاع می‌سازند.

## فصل ۴

### یادگیری تقویتی

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

#### ۱-۴ مفاهیم اولیه

دو بخش اصلی یادگیری تقویتی<sup>۱</sup> شامل عامل<sup>۲</sup> و محیط<sup>۳</sup> 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 می‌کند، محیط تغییر می‌کند؛ اما، ممکن است محیط به تنها یک نیز تغییر کند. عامل همچنین یک سیگنال پاداش<sup>۴</sup> از محیط دریافت می‌کند؛ سیگنالی که به عامل می‌گوید وضعیت تعامل فعلی آن با محیط چقدر خوب یا بد است. هدف عامل بیشینه‌کردن پاداش انباسته خود است که برگشت<sup>۵</sup> نام دارد. یادگیری تقویتی به روش‌هایی گفته می‌شود که در آن‌ها عامل رفتارهای مناسب برای رسیدن به هدف خود را می‌آموزد. در شکل ۱-۴ تعامل بین محیط و عامل نشان داده شده است.

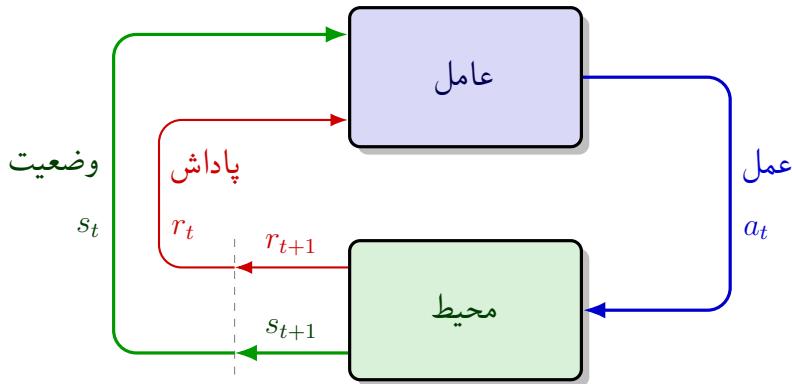
<sup>1</sup>Reinforcement Learning (RL)

<sup>2</sup>Agent

<sup>3</sup>Environment

<sup>4</sup>Reward

<sup>5</sup>Return



شکل ۱-۴: حلقه تعامل عامل و محیط

## ۱-۱-۴ حالت و مشاهدات

حالت<sup>۶</sup> ( $s$ ) توصیف کاملی از وضعیت محیط است. همه اطلاعات محیط در حالت وجود دارد. مشاهده<sup>۷</sup> ( $o$ ) 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.

## ۲-۱-۴ 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<sup>۸</sup> یا پیوسته<sup>۹</sup> باشد. در این پژوهش فضای عمل پیوسته و محدود به یک بازه مشخص است.

## ۳-۱-۴ سیاست

سیاست<sup>۱۰</sup> قاعده‌ای است که یک عامل برای تصمیم‌گیری در مورد اقدامات خود استفاده می‌کند. در این پژوهش به تناسب الگوریتم پیاده‌سازی شده از سیاست قطعی<sup>۱۱</sup> یا تصادفی<sup>۱۲</sup> استفاده شده است که به دو صورت زیر نشان

<sup>6</sup>State

<sup>7</sup>Observation

<sup>8</sup>Discrete

<sup>9</sup>Continuous

<sup>10</sup>Policy

<sup>11</sup>Deterministic

<sup>12</sup>Stochastic

داده می‌شود:

$$a_t = \mu(s_t) \quad (1-4)$$

$$a_t \sim \pi(\cdot | s_t) \quad (2-4)$$

که زیروند  $t$  بیانگر زمان است. در یادگیری تقویتی عمیق از سیاست‌های پارامتری شده 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 استفاده کرد. در این پژوهش پارامترهای سیاست با  $\theta$  نشان داده شده است و سپس نماد آن به عنوان زیروند سیاست مانند معادله (3-4) نشان داده شده است.

$$\begin{aligned} a_t &= \mu_\theta(s_t) \\ a_t &\sim \pi_\theta(\cdot | s_t) \end{aligned} \quad (3-4)$$

## ۴-۱-۴ مسیر

یک مسیر<sup>۱۳</sup> یک توالی از حالت‌ها و عمل‌ها در محیط است.

$$\tau = (s_0, a_0, s_1, a_1, \dots) \quad (4-4)$$

گذار حالت<sup>۱۴</sup> 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 به صورت زیر تعریف کرد.

$$s_{t+1} = f(s_t, a_t) \quad (5-4)$$

## ۵-۱-۴ تابع پاداش و برگشت

تابع پاداش<sup>۱۵</sup> در حالت کلی به حالت فعلی محیط، آخرین عمل انجام‌شده و حالت بعدی محیط بستگی دارد. تابع پاداش را می‌توان به صورت زیر تعریف کرد.

$$r_t = R(s_t, a_t, s_{t+1}) \quad (6-4)$$

---

<sup>13</sup>Trajectory

<sup>14</sup>State Transition

<sup>15</sup>Reward Function

در این پژوهش، پاداش تنها تابعی از جفتِ حالت-عمل ( $r_t = R(s_t, a_t)$ ) فرض شده است. هدف عامل این است که مجموع پادash‌های به دست آمده در طول یک مسیر را به حداکثر برساند. در این پژوهش مجموع پادash‌ها در طول یک مسیر را با نماد  $R(\tau)$  نشان داده شده است و به آن تابع برگشت<sup>۱۶</sup> است که مجموع پادash‌های به دست آمده در یک بازه زمانی ثابت و از مسیر  $\tau$  است که در معادله (۷-۴) نشان داده شده است.

$$R(\tau) = \sum_{t=0}^T r_t \quad (7-4)$$

نوع دیگری از برگشت، برگشت تنزیل شده با افق نامحدود<sup>۱۹</sup> است که مجموع همه پادash‌هایی است که تا به حال توسط عامل به دست آمده است. اما، فاصله زمانی تا دریافت پاداش باعث تنزیل ارزش آن می‌شود. این معادله برگشت (۸-۴) شامل یک فاکتور تنزیل<sup>۲۰</sup> با نماد  $\gamma$  است که عددی بین صفر و یک است.

$$R(\tau) = \sum_{t=0}^{\infty} \gamma^t r_t \quad (8-4)$$

## ۶-۱-۶ ارزش در یادگیری تقویتی

در یادگیری تقویتی، دانستن ارزش<sup>۲۱</sup> یک حالت یا جفتِ حالت-عمل ضروری است. منظور از ارزش، برگشت مورد انتظار<sup>۲۲</sup> است. یعنی اگر از آن حالت یا جفت حالت-عمل شروع شود و سپس برای همیشه طبق یک سیاست خاص عمل شود، به طور 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 شده است.

۱. تابع ارزش تحت سیاست<sup>۲۳</sup> ( $V^\pi(s)$ ): خروجی این تابع برگشت مورد انتظار است در صورتی که از حالت  $s$  شروع شود و همیشه طبق سیاست  $\pi$  عمل شود و به صورت زیر بیان می‌شود:

$$V^\pi(s) = \mathbb{E}_{\tau \sim \pi} [R(\tau) | s_0 = s] \quad (9-4)$$

۲. تابع ارزش-عمل تحت سیاست<sup>۲۴</sup> ( $Q^\pi(s, a)$ ): خروجی این تابع برگشت مورد انتظار است در صورتی که از حالت  $s$  شروع شود، یک اقدام دلخواه  $a$  (که ممکن است از سیاست  $\pi$  نباشد) انجام شود و سپس

<sup>16</sup>Return

<sup>17</sup>Discount

<sup>18</sup>Finite-Horizon Undiscounted Return

<sup>19</sup>Infinite-Horizon Discounted Return

<sup>20</sup>Discount Factor

<sup>21</sup>Value

<sup>22</sup>Expected Return

<sup>23</sup>On-Policy Value Function

<sup>24</sup>On-Policy Action-Value Function

برای همیشه طبق سیاست  $\pi$  عمل شود و بهصورت زیر بیان می‌شود:

$$Q^\pi(s, a) = \mathbb{E}_{\tau \sim \pi} [R(\tau) | s_0 = s, a_0 = a] \quad (10-4)$$

۳. تابع ارزش بهینه<sup>۲۵</sup> ( $V^*(s)$ ): 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:

$$V^*(s) = \max_{\pi} (V^\pi(s)) \quad (11-4)$$

۴. تابع ارزش-عمل بهینه<sup>۲۶</sup> ( $Q^*(s, a)$ ): 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:

$$Q^*(s, a) = \max_{\pi} (Q^\pi(s, a)) \quad (12-4)$$

## ۷-۱-۴ معادلات بلمن

توابع ارزش اشاره شده 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:

$$V^\pi(s) = \mathbb{E}_{\substack{a \sim \pi \\ s' \sim P}} [r(s, a) + \gamma V^\pi(s')] \quad (13-4)$$

$$Q^\pi(s, a) = r(s, a) + \mathbb{E}_{\substack{a \sim \pi \\ s' \sim P}} \left[ \gamma \mathbb{E}_{a' \sim \pi} [Q^\pi(s', a')] \right] \quad (14-4)$$

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

<sup>25</sup>Optimal Value Function

<sup>26</sup>Optimal Action-Value Function

$\pi$  با شرط حالت جدید  $s'$  نمونه برداری می شود. 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 می شود. معادلات بلمن برای توابع ارزش بهینه به شرح زیر هستند:

$$V^*(s) = \max_a \mathbb{E}_{s' \sim P} [r(s, a) + \gamma V^*(s')] \quad (15-4)$$

$$Q^*(s, a) = r(s, a) + \mathbb{E}_{s' \sim P} \left[ \gamma \max_{a'} Q^*(s', a') \right] \quad (16-4)$$

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

## ۸-۱-۴ تابع مزیت

گاهی در یادگیری تقویتی، نیازی به توصیف میزان خوبی یک عمل به صورت مطلق نیست، بلکه تنها می خواهیم بدانیم که چه مقدار بهتر از سایر اعمال به طور متوسط است. به عبارت دیگر، مزیت نسبی آن عمل مورد بررسی قرار می گیرد. این مفهوم با تابع مزیت <sup>۲۷</sup> توضیح داده می شود.

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

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

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  $\pi$  است و  $V^\pi(s)$  تابع ارزش حالت  $s$  تحت سیاست  $\pi$  است. این تابع مزیت نشان می دهد که انجام عمل  $a$  در حالت  $s$  نسبت به میانگین اعمال تحت سیاست  $\pi$  چقدر مزیت دارد. اگر  $A^\pi(s, a)$  مثبت باشد، نشان دهنده این است که عمل  $a$  بهتر از میانگین اعمال است و اگر منفی باشد، نشان دهنده کمتر بودن عمل کرد آن نسبت به میانگین است.

---

<sup>27</sup> Advantage Function

## ۲-۴ عامل گرادیان سیاست عمیق قطعی

گرادیان سیاست عمیق قطعی<sup>۲۸</sup> الگوریتمی است که هم‌مان یک تابع  $Q$  و یک سیاست را یاد می‌گیرد. این الگوریتم برای یادگیری تابع  $Q$  از داده‌های غیرسیاست محور<sup>۲۹</sup> 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 (۱۷-۴) به دست آورد.

$$a^*(s) = \arg \max_a Q^*(s, a) \quad (17-4)$$

الگوریتم DDPG ترکیبی از یادگیری تقریبی برای  $Q^*(s, a)$  و یادگیری تقریبی برای  $a^*(s)$  است و به صورتی طراحی شده است 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  $(s)$  استفاده شود. سپس، به جای 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.

## ۱-۲-۴ یادگیری $Q$ در DDPG

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

$$Q^*(s, a) = r(s, a) + \mathbb{E}_{s' \sim P} \left[ \gamma \max_{a'} Q^*(s', a') \right] \quad (18-4)$$

عبارت  $P \sim s'$  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  $\phi$  نشان داده شده است. 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<sup>۳۰</sup> (MSBE) استفاده شده است که معیاری برای نزدیکی  $Q_\phi$  به حالت بهینه برای برآورده کردن معادله بلمن است.

<sup>28</sup>Deep Deterministic Policy Gradient (DDPG)

<sup>29</sup>Off-Policy

<sup>30</sup>Mean Squared Bellman Error

$$L(\phi, \mathcal{D}) = \underset{(s,a,r,s',d) \sim \mathcal{D}}{\mathbb{E}} \left[ \left( Q_\phi(s, a) - \left( r + \gamma(1-d) \max_{a'} Q_\phi(s', a') \right) \right)^2 \right] \quad (19-4)$$

در الگوریتم DDPG دو ترفندهای عمکرد بهتر استفاده شده است که در ادامه به بررسی آن پرداخته شده است.

#### • بافرهای تکرار بازی

الگوریتم‌های یادگیری تقویتی جهت آموزش یک شبکه عصبی عمیق برای تقریب  $Q^*(s, a)$  از بافرهای تکرار بازی<sup>31</sup> تجربه شده 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 شده است چرا که اگر فقط از داده‌های بسیار جدید استفاده شود، بیش برآراش<sup>32</sup> 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 به عنوان هدف شناخته می‌شود.

$$r + \gamma(1-d) \max_{a'} Q_\phi(s', a') \quad (20-4)$$

در هنگام کمینه کردن تابع خطای میانگین مربعات بلمن، سعی شده است تا تابع  $Q$  شبیه‌تر به هدف یعنی رابطه<sup>20-4</sup> شود. اما مشکل این است که هدف بستگی به پارامترهای در حال آموزش  $\phi$  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  $\phi$  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 targ $\phi$  نشان داده می‌شوند. در الگوریتم DDPG، شبکه هدف در هر بهروزرسانی شبکه اصلی، با میانگین‌گیری پولیاک<sup>33</sup> به صورت زیر بهروزرسانی می‌شود.

$$\phi_{\text{targ}} \leftarrow \rho \phi_{\text{targ}} + (1-\rho) \phi \quad (21-4)$$

در رابطه بالا  $\rho$  یک ابرپارامتر<sup>34</sup> است که بین صفر و یک انتخاب می‌شود. در این پژوهش این مقدار نزدیک به یک درنظر گرفته شده است.

<sup>31</sup>Replay Buffers

<sup>32</sup>Overfit

<sup>33</sup>Polyak Averaging

<sup>34</sup>Hyperparameter

الگوریتم DDPG نیاز به یک شبکه سیاست هدف ( $\mu_{\theta_{\text{targ}}}$ ) برای محاسبه عملهایی که به طور تقریبی بیشینه  $Q$  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 اشاره شده، یادگیری  $Q$  در DDPG با کمینه کردن تابع خطای میانگین مربعات بلمن (معادله ۲۲-۴) با استفاده از کاهش گرادیان تصادفی<sup>۳۵</sup> انجام می‌شود.

$$L(\phi, \mathcal{D}) = \underset{(s, a, r, s', d) \sim \mathcal{D}}{\mathbb{E}} \left[ \left( Q_\phi(s, a) - (r + \gamma(1-d)Q_{\phi_{\text{targ}}}(s', \mu_{\theta_{\text{targ}}}(s')) \right)^2 \right] \quad (22-4)$$

## ۲-۲-۴ سیاست در DDPG

در این بخش یک سیاست تعیین شده ( $\mu_\theta(s)$ ) یادگرفته می‌شود تا عملی را انجام می‌دهد که بیشینه  $Q_\phi(s, a)$  رخ دهد. از آنجا که فضای عمل پیوسته است و فرض شده است که تابع  $Q$  نسبت به عمل مشتق‌پذیر است، رابطه زیر با استفاده از صعود گرادیان<sup>۳۶</sup> ( تنها نسبت به پارامترهای سیاست) بیشینه می‌شود.

$$\max_{\theta} \underset{s \sim \mathcal{D}}{\mathbb{E}} [Q_\phi(s, \mu_\theta(s))] \quad (23-4)$$

## ۳-۲-۴ اکتشاف و بهره‌برداری در DDPG

برای بهبود اکتشاف<sup>۳۷</sup> در سیاست‌های DDPG، در زمان آموزش نویز به عمل‌ها اضافه می‌شود. نویسنده‌گان مقاله DDPG<sup>۳۸</sup> [۵۷] توصیه کرده‌اند که نویز OU<sup>۳۹</sup> با همبندی زمانی<sup>۴۰</sup> اضافه شود. در زمان بهره‌برداری سیاست، از آنچه یادگرفته است، نویز به عمل‌ها اضافه نمی‌شود.

## ۴-۲-۴ شبکه DDPG

در این بخش، شبکه DDPG پیاده‌سازی شده آورده شده است. در این پژوهش الگوریتم ۱ در محیط پایتون با استفاده از کتابخانه TensorFlow<sup>۴۱</sup> [۵۸] پیاده‌سازی شده است.

<sup>35</sup>Stochastic Gradient Descent

<sup>36</sup>Gradient Ascent

<sup>37</sup>Exploration

<sup>38</sup>Ornstein–Uhlenbeck

<sup>39</sup>Time-Correlated

<sup>40</sup>Exploitation

---

## الگوریتم ۱ گرادیان سیاست عمیق قطعی

---

- وروودی: پارامترهای اولیه سیاست  $(\theta)$ ، پارامترهای تابع  $Q(\phi)$ ، بافر تکرار بازی خالی  $(\mathcal{D})$
- ۱: پارامترهای هدف را برابر با پارامترهای اصلی قرار دهید  $\theta \leftarrow \theta_{\text{targ}}$
  - ۲: تا وقتی همگرایی رخ دهد:
  - ۳: وضعیت  $s$  را مشاهده کرده و عمل  $a = \text{clip}(\mu_\theta(s) + \epsilon, a_{\text{Low}}, a_{\text{High}})$  را انتخاب کنید به طوری که  $\mathcal{N} \sim \epsilon$  است.
  - ۴: عمل  $a$  را در محیط اجرا کنید.
  - ۵: وضعیت بعدی  $s'$ ، پاداش  $r$  و سیگنال پایان  $d$  را مشاهده کنید تا نشان دهد آیا  $s'$  پایانی است یا خیر.
  - ۶: اگر  $s'$  پایانی است، وضعیت محیط را بازنشانی کنید.
  - ۷: اگر زمان بهروزرسانی فرا رسیده است:
  - ۸: به ازای هر تعداد بهروزرسانی:
  - ۹: یک دسته تصادفی گذر از یک حالت به حالت دیگر، از  $\mathcal{D}$ ، از  $B = \{(s, a, r, s', d)\}$ ، از  $\mathcal{N}$  نمونه‌گیری شود.
  - ۱۰: 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: ۱۱

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

سیاست را با یک مرحله از صعود گرادیان با استفاده از رابطه زیر بهروزرسانی کنید: ۱۲

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

شبکه‌های هدف را با استفاده از معادلات زیر بهروزرسانی کنید: ۱۳

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

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

## ۳-۴ عامل گرادیان سیاست عمیق قطعی تاخیری دوگانه

عامل گرادیان سیاست عمیق قطعی تاخیری دوگانه<sup>۴۱</sup> یکی از الگوریتم‌های یادگیری تقویتی است که برای حل مسائل کنترل در محیط‌های پیوسته طراحی شده است. 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  $Q$  یادگرفته شده شروع به بیش برآورد 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) از دو ترفندهای زیر جهت بهبود مشکلات اشاره شده استفاده می‌کند.

- یادگیری دوگانه محدودشده<sup>۴۲</sup>: الگوریتم TD3 به جای یک تابع  $Q$ ، دو تابع  $Q_{\phi_1}$  و  $Q_{\phi_2}$  را یاد می‌گیرد (از این رو دوگانه<sup>۴۳</sup> نامیده می‌شود) و از کوچک‌ترین مقدار این دو  $Q_{\phi_1}$  و  $Q_{\phi_2}$  در تابع بلمن استفاده می‌شود. نحوه محاسبه هدف بر اساس دو تابع  $Q$  اشاره شده در رابطه (۲۴-۴) آورده شده است.

$$y(r, s', d) = r + \gamma(1 - d) \min_{i=1,2} Q_{\phi_i, \text{targ}}(s', a'(s')) \quad (24-4)$$

سپس، در هر دو تابع  $Q_{\phi_1}$  و  $Q_{\phi_2}$  یادگیری انجام می‌شود.

$$L(\phi_1, \mathcal{D}) = \mathbb{E}_{(s, a, r, s', d) \sim \mathcal{D}} \left( Q_{\phi_1}(s, a) - y(r, s', d) \right)^2 \quad (25-4)$$

$$L(\phi_2, \mathcal{D}) = \mathbb{E}_{(s, a, r, s', d) \sim \mathcal{D}} \left( Q_{\phi_2}(s, a) - y(r, s', d) \right)^2 \quad (26-4)$$

- بهروزرسانی‌های تاخیری سیاست<sup>۴۴</sup>: الگوریتم TD3 سیاست را با تاخیر بیشتری نسبت به تابع  $Q$  بهروزرسانی می‌کند. در مرجع [۵۹] توصیه شده است که برای هر دو بهروزرسانی تابع  $Q$ ، یک بهروزرسانی سیاست انجام شود.

<sup>41</sup>Twin Delayed Deep Deterministic Policy Gradient (TD3)

<sup>42</sup>Clipped Double-Q Learning

<sup>43</sup>twin

<sup>44</sup>Delayed Policy Updates

این دو ترفند منجر به بهبود قابل توجه عملکرد TD3 نسبت به DDPG پایه می‌شوند. در نهایت سیاست با بیشینه‌کردن  $Q_{\phi_1}$  آموخته می‌شود:

$$\max_{\theta} \mathbb{E}_{s \sim \mathcal{D}} [Q_{\phi_1}(s, \mu_{\theta}(s))] \quad (27-4)$$

### ۱-۳-۴ اکتشاف و بهره‌برداری در TD3

الگوریتم TD3 یک سیاست قطعی را به صورت غیرسیاست محور 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 گاوی با میانگین صفر بدون همبندی زمانی اعمال شده است. شدت نویز جهت بهره‌برداری بهتر در طول زمان کاهش می‌یابد.

### ۲-۳-۴ TD3 شبکه

در این بخش الگوریتم TD3 پیاده‌سازی شده آورده شده است. در این پژوهش الگوریتم [۴](#) در محیط پایتون با استفاده از کتابخانه PyTorch [\[۶۰\]](#) پیاده‌سازی شده است.

---

## الگوریتم ۲ عامل گرادیان سیاست عمیق قطعی تا خیری دوگانه

---

وروودی: پارامترهای اولیه سیاست  $(\theta)$ ، پارامترهای تابع  $Q(\phi_1, \phi_2)$ ، بافر بازی خالی  $(\mathcal{D})$

۱: پارامترهای هدف را برابر با پارامترهای اصلی قرار دهید  $\theta_{\text{targ}} \leftarrow \theta$

۲: تا وقتی همگرایی رخ دهد:

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

۴: عمل  $a$  را در محیط اجرا کنید.

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

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

۷: اگر زمان بهروزرسانی فرا رسیده است:

۸: به ازای  $j$  در هر تعداد بهروزرسانی:

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

۱۰: هدف را محاسبه کنید:

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

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

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

۱۲: اگر باقیمانده  $j$  بر تا خیر سیاست برابر ۰ باشد :

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

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

۱۴: شبکه‌های هدف را با استفاده از معادلات زیر بهروزرسانی کنید:

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

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

## ۴-۴ عامل عملگر نقاد نرم

عملگرد نقاد نرم<sup>۴۵</sup> الگوریتمی است که یک سیاست تصادفی را به صورت غیرسیاست محور بهینه می‌کند و پلی بین بهینه‌سازی سیاست تصادفی و رویکردهای غیرسیاست محور مانند DDPG ایجاد می‌کند. این الگوریتم جانشین مستقیم TD3 نیست (زیرا تقریباً همزمان منتشر شده است)؛ اما، ترفند یادگیری دوگانه محدود شده را در خود جای داده است و به دلیل سیاست تصادفی SAC، از روشی به نام صاف کردن سیاست هدف<sup>۴۶</sup> استفاده شده است. یکی از ویژگی‌های اصلی SAC، تنظیم آنتروپی است. آنتروپی معیاری از تصادفی بودن 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 تسريع کند. همچنین، می‌تواند از همگرایی زودهنگام سیاست به یک بهینه محلی بد جلوگیری کند. برای توضیح SAC، ابتدا باید به بررسی یادگیری تقویتی تنظیم شده با آنتروپی<sup>۴۷</sup> پرداخته شود. در RL تنظیم شده با آنتروپی، روابط تابع ارزش کمی متفاوت است.

### ۱-۴-۴ یادگیری تقویتی تنظیم شده 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:

$$H(X) = \text{E}_{x \sim p} [-\log p(x)] \quad (28-4)$$

### ۲-۴-۴ سیاست در SAC

در یادگیری تقویتی تنظیم شده 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 به صورت زیر می‌شود.

<sup>45</sup> Soft Actor Critic (SAC)

<sup>46</sup> Target Policy Smoothing

<sup>47</sup> Entropy-Regularized Reinforcement Learning

$$\pi^* = \arg \max_{\pi} \mathbb{E}_{\tau \sim \pi} \sum_{t=0}^{\infty} \gamma^t \left( R(s_t, a_t, s_{t+1}) + \alpha H(\pi(\cdot | s_t)) \right) \quad (29-4)$$

که در آن ( $\alpha > 0$ ) ضریب مبادله<sup>48</sup> است.

### ۳-۴-۴ تابع ارزش در SAC

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

$$V^\pi(s) = \mathbb{E}_{\tau \sim \pi} \left[ \sum_{t=0}^{\infty} \gamma^t \left( R(s_t, a_t, s_{t+1}) + \alpha H(\pi(\cdot | s_t)) \right) \middle| s_0 = s \right] \quad (30-4)$$

### ۴-۴-۴ تابع Q در SAC

تابع  $Q^\pi$  به گونه‌ای تغییر می‌کند که پاداش‌های آنتروپی را از هر مرحله زمانی به جز مرحله اول شامل می‌شود.

$$Q^\pi(s, a) = \mathbb{E}_{\tau \sim \pi} \left[ \sum_{t=0}^{\infty} \gamma^t R(s_t, a_t, s_{t+1}) + \alpha \sum_{t=1}^{\infty} \gamma^t H(\pi(\cdot | s_t)) \middle| s_0 = s, a_0 = a \right] \quad (31-4)$$

با این تعاریف رابطه<sup>4</sup>  $V^\pi$  و  $Q^\pi$  به صورت زیر است.

$$V^\pi(s) = \mathbb{E}_{a \sim \pi} [Q^\pi(s, a)] + \alpha H(\pi(\cdot | s)) \quad (32-4)$$

### ۵-۴-۴ معادله بلمن در SAC

معادله بلمن در حالت تنظیم شده با آنتروپی به صورت زیر ارائه می‌شود.

$$Q^\pi(s, a) = \mathbb{E}_{\substack{s' \sim P \\ a' \sim \pi}} [R(s, a, s') + \gamma (Q^\pi(s', a') + \alpha H(\pi(\cdot | s')))] \quad (33-4)$$

$$= \mathbb{E}_{s' \sim P} [R(s, a, s') + \gamma V^\pi(s')] \quad (34-4)$$

---

<sup>48</sup>Trade-Off

## ۶-۴-۴ یادگیری Q

با درنظرگرفتن موارد اشاره شده، یادگیری Q در SAC با کمینه کردن تابع خطای میانگین مربعات بلمن (MSBE) یعنی معادله (۳۵-۴) با استفاده از کاهش گرادیان انجام می شود.

$$L(\phi_i, \mathcal{D}) = \underset{(s,a,r,s',d) \sim \mathcal{D}}{\text{E}} \left[ \left( Q_{\phi_i}(s, a) - y(r, s', d) \right)^2 \right] \quad (35-4)$$

در معادله (۳۵-۴) تابع هدف برای روش یادگیری تقویتی SAC به صورت زیر تعریف می شود.

$$y(r, s', d) = r + \gamma(1-d) \left( \min_{j=1,2} Q_{\phi_{\text{targ},j}}(s', \tilde{a}') - \alpha \log \pi_\theta(\tilde{a}'|s') \right), \quad \tilde{a}' \sim \pi_\theta(\cdot|s') \quad (36-4)$$

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

## ۷-۴-۴ 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.

$$V^\pi(s) = \underset{a \sim \pi}{\text{E}} [Q^\pi(s, a)] + \alpha H(\pi(\cdot|s)) \quad (37-4)$$

$$= \underset{a \sim \pi}{\text{E}} [Q^\pi(s, a) - \alpha \log \pi(a|s)] \quad (38-4)$$

در بهینه سازی سیاست از ترفندهای پارامترسازی مجدد <sup>۴۹</sup> استفاده می شود، که در آن نمونه ای از  $\pi_\theta(\cdot|s)$  با محاسبه یک تابع قطعی از وضعیت، پارامترهای سیاست و نویز مستقل استخراج می شود. در این پژوهش مانند نویسندهای مقاله SAC [۶۱]، از یک سیاست گاووسی فشرده <sup>۵۰</sup> استفاده شده است. بر اساس این روش نمونه ها مطابق با رابطه زیر بدست می آیند:

$$\tilde{a}_\theta(s, \xi) = \tanh(\mu_\theta(s) + \sigma_\theta(s) \odot \xi), \quad \xi \sim \mathcal{N} \quad (39-4)$$

در رابطه بالا  $\odot$  نماد ضرب داخلی است. تابع  $\tanh$  در سیاست SAC تضمین می کند که اعمال در یک محدوده متناهی محدود شوند. این مورد در سیاست های VPG، TRPO و PPO وجود ندارد. همچنین اعمال این تابع توزیع را از حالت گاووسی تغییر می دهد.

<sup>49</sup>Reparameterization

<sup>50</sup>Squashed Gaussian Policy

در الگوریتم SAC با استفاده از ترفندهای پارامتری‌سازی مجدد، عمل‌ها از یک توزیع نرمال به وسیله نویز تصادفی تولید شده و به این ترتیب امکان محاسبه مشتق‌ها به طور مستقیم از طریقتابع 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 ترفندهای احتمال<sup>۵۱</sup> است که معمولاً باعث افزایش واریانس و ناپایداری در آموزش می‌شود.

$$\mathbb{E}_{a \sim \pi_\theta} [Q^{\pi_\theta}(s, a) - \alpha \log \pi_\theta(a|s)] = \mathbb{E}_{\xi \sim \mathcal{N}} [Q^{\pi_\theta}(s, \tilde{a}_\theta(s, \xi)) - \alpha \log \pi_\theta(\tilde{a}_\theta(s, \xi)|s)] \quad (40-4)$$

برای به دست آوردن تابع هزینه سیاست، گام نهایی این است که باید  $Q^{\pi_\theta}$  را با یکی از تخمین‌زننده‌های تابع خود جایگزین کنیم. برخلاف TD3 که از  $Q_{\phi_1}$  ( 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 بهینه می‌شود:

$$\max_{\theta} \mathbb{E}_{\substack{s \sim \mathcal{D} \\ \xi \sim \mathcal{N}}} \left[ \min_{j=1,2} Q_{\phi_j}(s, \tilde{a}_\theta(s, \xi)) - \alpha \log \pi_\theta(\tilde{a}_\theta(s, \xi)|s) \right] \quad (41-4)$$

که تقریباً مشابه بهینه‌سازی سیاست در DDPG و min-double-Q TD3 است، به جز ترفندهای TD3 و عبارت آنتروپی:

## ۸-۴-۴ 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  $\alpha$  به طور صریح تعادل بین اکتشاف و بهره‌برداری را کنترل می‌کند، به طوری که مقادیر بالاتر  $\alpha$  به اکتشاف بیشتر و مقادیر پایین‌تر  $\alpha$  به بهره‌برداری بیشتر منجر می‌شود. مقدار بهینه  $\alpha$  ( 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.

<sup>51</sup>Likelihood Ratio Trick

## ۹-۴-۴ شبکه SAC

در این بخش الگوریتم SAC پیاده‌سازی شده آورده شده است. در این پژوهش الگوریتم <sup>۳</sup> در محیط پایتون با استفاده از کتابخانه PyTorch <sup>[۶۰]</sup> پیاده‌سازی شده است.

### الگوریتم ۳ عامل عملگرد نقاد نرم

وروودی: پارامترهای اولیه سیاست  $(\theta)$ , پارامترهای تابع  $Q(\phi_1, \phi_2)$ , بافر بازی خالی  $(\mathcal{D})$

- ۱: پارامترهای هدف را برابر با پارامترهای اصلی قرار دهید  $\theta_{\text{targ}} \leftarrow \theta$ ,  $\phi_{\text{targ},2} \leftarrow \phi_2$ ,  $\phi_{\text{targ},1} \leftarrow \phi_1$
- ۲: تا وقتی همگرایی رخ دهد:
- ۳: وضعیت  $(s)$  را مشاهده کرده و عمل  $\pi_\theta(\cdot|s) \sim a$  را انتخاب کنید.
- ۴: عمل  $a$  را در محیط اجرا کنید.
- ۵: وضعیت بعدی  $s'$ , پاداش  $r$  و سیگنال پایان  $d$  را مشاهده کنید تا نشان دهد آیا  $s'$  پایانی است یا خیر.
- ۶: اگر  $s'$  پایانی است، وضعیت محیط را بازنگشانی کنید.
- ۷: اگر زمان بهروزرسانی فرا رسیده است:
- ۸: به ازای  $j$  در هر تعداد بهروزرسانی:
- ۹: یک دسته تصادفی گذر از یک حالت به حالت دیگر، از  $\mathcal{D}$ ,  $B = \{(s, a, r, s', d), \dots\}$  از نمونه‌گیری شود.
- ۱۰: هدف را محاسبه کنید:

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

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

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

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

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

شبکه‌های هدف را با استفاده از معادلات زیر بهروزرسانی کنید:

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

## ۵-۴ عامل بهینه‌سازی سیاست مجاور

الگوریتم بهینه‌سازی سیاست مجاور<sup>۵۲</sup> یک الگوریتم بهینه‌سازی سیاست مبتنی بر گرادیان است که برای حل مسائل کنترل مسئله‌های یادگیری تقویتی استفاده می‌شود. این الگوریتم از الگوریتم TRPO<sup>۵۳</sup> الهام گرفته شده است 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، به‌دلیل 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 شده است 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 شده است.

• روش PPO-Penalty: روش PPO-Penalty به‌دلیل حل تقریبی و بهروزرسانی با محدودیت و اگرایی کولباک-لیبلر<sup>۵۴</sup> است، مشابه روشی که در الگوریتم TRPO استفاده شده است. با این حال، به

جای اعمال یک محدودیت سخت<sup>۵۵</sup>، PPO-Penalty و اگرایی KL را در تابع هدف جرمیه می‌کند. این جرمیه به طور خودکار در طول آموزش تنظیم می‌شود تا از افت ناگهانی عملکرد جلوگیری کند.

• روش PPO-Clip: در این روش، هیچ عبارت و اگرایی KL در تابع هدف وجود ندارد و هیچ محدودیتی اعمال نمی‌شود. در عوض، در PPO-Clip از یک عملیات بریدن<sup>۵۶</sup> 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 شده است.

<sup>52</sup>Proximal Policy Optimization (PPO)

<sup>53</sup>Trust Region Policy Optimization

<sup>54</sup>Kullback-Leibler (KL) Divergence

<sup>55</sup>Hard Constraint

<sup>56</sup>Clipping

## ۱-۵-۴ سیاست در الگوریتم PPO

تابع سیاست در الگوریتم PPO به صورت یک شبکه عصبی پیاده‌سازی شده است. 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 شده است. تابع سیاست در الگوریتم PPO به صورت زیر بهروزرسانی می‌شود:

$$\theta_{k+1} = \arg \max_{\theta} \mathbb{E}_{s,a \sim \pi_{\theta_k}} [L(s,a,\theta_k, \theta)] \quad (42-4)$$

در این پژوهش برای به حداقل رساندن تابع هدف، چندین گام بهینه‌سازی گرادیان کاهشی تصادفی<sup>۵۷</sup> اجرا شده است. در معادله بالا  $L$  به صورت زیر تعریف شده است:

$$L(s,a,\theta_k, \theta) = \min \left( \frac{\pi_\theta(a|s)}{\pi_{\theta_k}(a|s)} A^{\pi_{\theta_k}}(s,a), \text{clip} \left( \frac{\pi_\theta(a|s)}{\pi_{\theta_k}(a|s)}, 1-\epsilon, 1+\epsilon \right) A^{\pi_{\theta_k}}(s,a) \right) \quad (43-4)$$

که در آن  $\epsilon$  یک ابرپارامتر است که مقدار آن معمولاً کوچک است. این ابرپارامتر مشخص می‌کند که چقدر اندازه گام بهینه‌سازی باید محدود شود. در این پژوهش مقدار  $0.2 = \epsilon$  انتخاب شده است. جهت سادگی در پیاده‌سازی معادله (۴۳-۴) به معادله تغیر داده شده است.

$$L(s,a,\theta_k, \theta) = \min \left( \frac{\pi_\theta(a|s)}{\pi_{\theta_k}(a|s)} A^{\pi_{\theta_k}}(s,a), g(\epsilon, A^{\pi_{\theta_k}}(s,a)) \right) \quad (44-4)$$

که تابع  $g$  به صورت زیر تعریف شده است.

$$g(\epsilon, A) = \begin{cases} (1+\epsilon)A & A \geq 0 \\ (1-\epsilon)A & A < 0 \end{cases} \quad (45-4)$$

در حالی که این نوع محدود کردن (PPO-Clip) تا حد زیادی به اطمینان از بهروزرسانی‌های معقول سیاست کمک می‌کند، همچنان ممکن است سیاستی به دست 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<sup>۵۸</sup> استفاده شده است. اگر میانگین واگرایی کولباک-لیبلر (KL) خط‌مشی جدید از خط‌مشی قدیمی از یک آستانه فراتر رود، گام‌های گرادیان (بهینه‌سازی) را متوقف می‌شوند.

<sup>57</sup>Stochastic Gradient Descent (SGD)

<sup>58</sup>Early Stopping

## ۲-۵-۴ اکتشاف و بهره‌برداری در PPO

الگوریتم PPO از یک سیاست تصادفی به صورت 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<sup>۵۹</sup> شود.

## ۳-۵-۴ شبکه PPO

در این بخش الگوریتم PPO پیاده‌سازی شده آورده شده است. در این پژوهش الگوریتم <sup>۴</sup> در محیط پایتون با استفاده از کتابخانه PyTorch [۶۰] پیاده‌سازی شده است.

---

<sup>59</sup>Local Optima

---

#### الگوریتم ۴ بهینه‌سازی سیاست مجاور (PPO-Clip)

---

وروودی: پارامترهای اولیه سیاست  $(\theta_0)$ ، پارامترهای تابع ارزش  $(\phi_0)$

۱: به ازای ...  $k = 0, 1, 2, \dots$

۲: مجموعه‌ای از مسیرها به نام  $\{\tau_i\}$  با اجرای سیاست  $\pi(\theta_k) = \pi_k$  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

فعالی  $.V_{\phi_k}$

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

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

۶: معمولاً از طریق گرادیان افزایشی تصادفی Adam. برآذش تابع ارزش با رگرسیون بر روی میانگین مربعات خطای:

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

معمولًا از طریق برخی از الگوریتم‌های کاهشی گرادیان.

---

## فصل ۵

### شبیه‌سازی عامل در محیط سه جسمی

در این فصل، فرآیند شبیه‌سازی عامل هوشمند کنترل‌کننده فضای‌پیما در محیط دینامیکی سه جسمی بررسی شده است. در بخش ۱-۵ به طراحی و در بخش ۲-۵ 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 فضای‌پیما را به‌طور مؤثر به سمت اهداف تعیین‌شده هدایت کند، در حالی که محدودیت‌هایی نظری مصرف سوخت و وجود اغتشاش دارد.

#### ۱-۵ طراحی عامل

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

#### ۱-۱-۵ فضای حالت

فضای حالت<sup>۱</sup> در این پژوهش به‌گونه‌ای طراحی شده است که وضعیت دینامیکی فضای‌پیما را نسبت به یک مسیر و سرعت مرجع مشخص می‌کند. این فضا شامل اختلاف‌های موقعیت و سرعت از مسیر و سرعت مرجع است و به صورت زیر تعریف شده است:

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

که در آن:

---

<sup>1</sup>State Space

•  $\delta x, \delta y$ : اختلاف موقعیت فضاییما نسبت به مسیر مرجع در محورهای  $x, y$ .

•  $\dot{\delta}x, \dot{\delta}y$ : اختلاف سرعت فضاییما نسبت به سرعت مرجع در محورهای  $x, y$ .

هر یک از این متغیرها به طور مستقل وضعیت فضاییما 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.

## ۲-۱-۵ فضای عمل

فضای عمل<sup>۲</sup> فضاییما با پیشران کم مجموعه‌ای از عمل‌های پیوسته است که فضاییما می‌تواند در محیط شبیه‌سازی انجام دهد. این فضا به گونه‌ای طراحی شده که امکان اعمال نیرو در جهت‌های مشخص و با مقادیر مناسب با توان واقعی فضاییماها فراهم شود. به طور خاص، فضای اقدام شامل موارد زیر است:

• **نیروی اعمال شده در جهت  $x$ :** این متغیر پیوسته، مقدار نیرویی را که در جهت محور  $x$  به فضاییما وارد می‌شود، تعیین می‌کند. دامنه این نیرو بر اساس توان پیشرانه‌های موجود در فضاییماهای واقعی انتخاب شده است. به عبارت دیگر، اگر حداقل 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 اعمال شده در جهت  $y$ :** این متغیر پیوسته، مقدار نیرویی را که در جهت محور  $y$  به فضاییما 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 فضاییماها، به ویژه توان و محدودیت‌های پیشرانه‌های آنها، صورت گرفته است. این امر اطمینان می‌دهد که شبیه‌سازی تا حد ممکن به شرایط واقعی نزدیک باشد و نتایج

<sup>2</sup>Action Space

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

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

### انطباق بازی فضای عمل با داده‌های واقعی

برای همتراز کردن شبیه‌سازی با سخت‌افزارهای واقعی، از بیشینه‌ی نیروی بی‌بعد پیشان‌ها استفاده می‌شود. جدول زیر نمونه‌هایی از فضاییماهای مجهز به پیشان‌های یونی/الکتریکی را نشان می‌دهد که مبنای انتخاب بازی نیروی عمل قرار گرفته شده‌اند. با توجه به برداری‌بودن عمل  $a = [f_x \ f_y]$ ، کران‌ها را به دو صورت اعمال شده‌است:

$$|a| \leq f_{\text{nondim max.}} \quad \text{یا} \quad f_{x,\max} = f_{y,\max} = f_{\text{nondim max.}}$$

با استناد به جدول ۱-۵، مقدار نمونه  $4 \times 10^{-2}$  شبیه‌سازی شده با Psyche هم مرتبه و کمتر از DS1 است که باعث شده‌است بازی عمل را در چارچوب پیشان‌های کمتر از است واقع‌گرایانه نگه داشته شود.

جدول ۱-۵: قابلیت‌های بی‌بعد پیشان‌کمتر از است فضاییماهای مختلف در سامانه‌ی زمین-ماه [۵۵].

$F_{\max}$ (mN)	$M_{3,0}$ (kg)	$f_{\max, \text{nondim}}$	نام فضاییما	نام اختصار
92.0	486.3	$6.940 \cdot 10^{-2}$	Deep Space 1	DS1
279.3	2464	$4.158 \cdot 10^{-2}$	Psyche	Psyche
91.0	1217.8	$2.741 \cdot 10^{-2}$	Dawn	Dawn
1.25	14	$3.276 \cdot 10^{-2}$	Lunar IceCube	LIC
22.8	510	$1.640 \cdot 10^{-2}$	Hayabusa 1	H1
27.0	608.6	$1.628 \cdot 10^{-2}$	Hayabusa 2	H2
—	—	$4 \cdot 10^{-2}$	فضاییما نمونه	s/c

### ۳-۱-۵ تابع پاداش

تابع پاداش<sup>۳</sup> بهمنظور هدایت رفتار عامل طراحی شده و شامل سه بخش اصلی در طول شبیه‌سازی و یک پاداش نهایی در هنگام پایان است:

- پاداش نهایی برای دستیابی به هدف: در صورت وارد شدن وضعیت سامانه به مجموعه هدف، شبیه‌سازی بلاfacسله خاتمه یافته و یک پاداش بزرگ مثبت به عامل تخصیص داده می‌شود تا ارزش‌یابی فرایند همگرایی را تقویت کند.
- جریمه نهایی برای دور شدن بیش از حد: اگر وضعیت سامانه از حدود اینمی تعریف شده فراتر رود، اپیزود متوقف شده و یک جریمه بزرگ منفی به عامل تعلق می‌گیرد تا ناپایداری یا گریز از ناحیه قابل قبول را منعکس کند.
- جریمه برای مصرف سوخت: در طول مسیر، استفاده بیش از حد از پیشرانه با جریمه همراه است.
- جریمه برای انحراف از مسیر مرجع: در طول مسیر، انحراف از مسیر مرجع باعث دریافت جریمه متناسب می‌شود.

این دو مؤلفه‌ی ترمینال با هدف تضمین همگرایی سیاست کنترلی به سمت مدار مطلوب طراحی شده‌اند. پاداش مثبت پایان، طیف مسیرهای موفق را برجسته کرده و عامل را به سمت سیاست‌های پایدار هدایت می‌کند؛ در مقابل، جریمه‌ی منفی پایان با ایجاد گرادیان تنبیه‌ی، از انتخاب راهبردهایی که منجر به خروج خطرناک از ناحیه‌ی عملیاتی می‌شوند جلوگیری می‌نماید.

تابع پاداش به صورت زیر تعریف می‌شود:

$$r(s, a) = r_{\text{thrust}}(a) + r_{\text{reference}}(s) + r_{\text{terminal}}(s)$$

که در آن مؤلفه‌ها عبارتند از:

$$r_{\text{thrust}}(a) = -k_1 \cdot |a| \quad (1-5)$$

$$r_{\text{reference}}(s) = -k_2 \cdot d(s, s_{\text{reference}}) \quad (2-5)$$

$$r_{\text{terminal}}(s) = \begin{cases} +R_{\text{goal}} & \text{if } s \in S_{\text{goal}} \\ -R_{\text{fail}} & \text{if } d(s, s_{\text{reference}}) > \epsilon \\ 0 & \text{otherwise} \end{cases} \quad (3-5)$$

---

<sup>3</sup>Reward Function

در این رابطه:

۱.  $R_{\text{goal}}$ : یک پاداش بزرگ مثبت برای دستیابی به هدف است.
  ۲.  $R_{\text{fail}}$ : یک جریمه بزرگ منفی برای خروج از محدوده مجاز است که رفتارهای ناپایدار را بهشت سرکوب می‌کند.
  ۳.  $d(s, s')$ : فاصله بین دو وضعیت بوده و به صورت فاصله اقلیدسی محاسبه می‌شود.
- ضرایب  $k_1, k_2$  برای تنظیم تعادل بین بهینه‌سازی مصرف سوخت و حفظ نزدیکی به مسیر مرجع استفاده می‌شوند. انتخاب مناسب مقادیر این ضرایب نقش کلیدی در سرعت همگرایی و پایداری الگوریتم یادگیری تقویتی دارد.

## ۲-۵ شبیه‌سازی عامل

در این زیربخش، فرآیند شبیه‌سازی و آموزش عامل با استفاده از الگوریتم‌های یادگیری تقویتی پیش‌رفته ارائه شده است. تمرکز بر طراحی شبکه‌ها، منطق انتخاب الگوریتم‌ها، فرآپارامترهای کلیدی و ملاحظات پایداری در حین آموزش است تا تکرار پذیری و دقت نتایج تضمین شود.

### ۱-۲-۵ پارامترهای یادگیری و منطق انتخاب الگوریتم‌ها

الگوریتم‌های SAC، TD3، DDPG و PPO به دلیل کارایی در فضاهای کنش پیوسته و عملکرد پایدار در محیط‌های پیچیده انتخاب شده‌اند. به‌طور خلاصه:

- DDPG: سیاست قطعی با شبکه‌های هدف و میانگین پلیاک؛ مناسب محیط‌های پیوسته با هزینه محاسباتی پایین‌تر، اما حساس به نویز.

جدول ۲-۵: جدول پارامترها و مقادیر پیشفرض الگوریتم DDPG [۶۲]

نام پارامتر	مقدار	نام پارامتر	مقدار
گام در هر دوره یادگیری	30 000	تعداد دوره‌های یادگیری	100
اندازه‌ی مخزن تجربه	$10^6$	ضریب تنزیل (γ)	0.99
ضریب میانگین پلیاک	0.995	نرخ یادگیری سیاست	$10^{-3}$
نرخ یادگیری Q	$10^{-3}$	اندازه‌ی دسته	1024
گام شروع استفاده از سیاست	5 000	گام شروع به روزرسانی	1 000
فاصله‌ی به روزرسانی	2 000	نویز عمل	0.1
حداکثر طول رخداد	6 000	دستگاه	Cuda
اندازه شبکه‌ی Actor	( $2^5, 2^5$ )	تابع فعال‌سازی Actor	ReLU
اندازه شبکه‌ی Critic	( $2^5, 2^5$ )	تابع فعال‌سازی Critic	ReLU

- TD3: بهبود DDPG با دو Critic، هموارسازی سیاست هدف و به روزرسانی تأخیری سیاست؛ کاهش بیشبرادری Q و پایداری بیشتر.

جدول ۳-۵: جدول پارامترها و مقادیر پیشفرض الگوریتم TD3 [۶۲]

نام پارامتر	مقدار	نام پارامتر	مقدار
گام در هر دوره یادگیری	30 000	تعداد دوره‌های یادگیری	100
اندازه‌ی مخزن تجربه	$10^6$	ضریب تنزیل (γ)	0.99
ضریب میانگین پلیاک	0.995	نرخ یادگیری سیاست	$10^{-3}$
نرخ یادگیری Q	$10^{-3}$	اندازه‌ی دسته	1024
گام شروع استفاده از سیاست	5 000	گام شروع به روزرسانی	1 000
فاصله‌ی به روزرسانی	2 000	نویز عمل	0.1
نویز هدف	0.2	برش نویز	0.5
تأخیر در به روزرسانی سیاست	2	حداکثر طول رخداد	30 000
اندازه شبکه‌ی Actor	( $2^5, 2^5$ )	تابع فعال‌سازی Actor	ReLU
اندازه شبکه‌ی Critic	( $2^5, 2^5$ )	تابع فعال‌سازی Critic	ReLU

- SAC: سیاست تصادفی بیشینه‌ساز آنتروپی با دمای  $\alpha$ ؛ کاوش مؤثرتر و همگرایی پایدارتر در محیط‌های نویزی.

جدول ۴-۵: جدول پارامترها و مقادیر پیشفرض الگوریتم SAC [۶۲]

نام پارامتر	مقدار	نام پارامتر	مقدار
گام در هر دوره یادگیری	30 000	تعداد دوره‌های یادگیری	100
اندازه‌ی مخزن تجربه	$10^6$	ضریب تنزیل ( $\gamma$ )	0.99
ضریب میانگین پلیاک	0.995	نرخ یادگیری	$10^{-3}$
نرخ دمای آلفا	0.2	اندازه‌ی دسته	1024
گام شروع به روزرسانی	5 000	فاصله‌ی به روزرسانی	1 000
تعداد به روزرسانی در هر مرحله	10	حداکثر طول رخداد	2 000
تعداد اپیزودهای آزمون	10	تابع فعال‌سازی Actor	30 000
اندازه شبکه‌ی Actor	$(2^5, 2^5)$	تابع فعال‌سازی Critic	ReLU
اندازه شبکه‌ی Critic	$(2^5, 2^5)$		ReLU

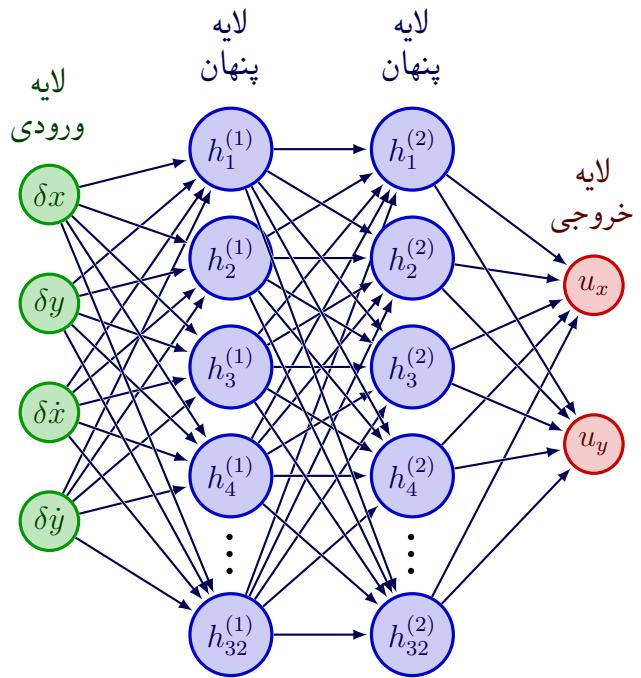
- PPO: روش مبتنی بر سیاست با برش نسبت احتمال؛ به روزرسانی‌های ایمن و پیاده‌سازی ساده با کارایی تجربی بالا.

جدول ۵-۵: جدول پارامترها و مقادیر پیشفرض الگوریتم PPO [۶۲]

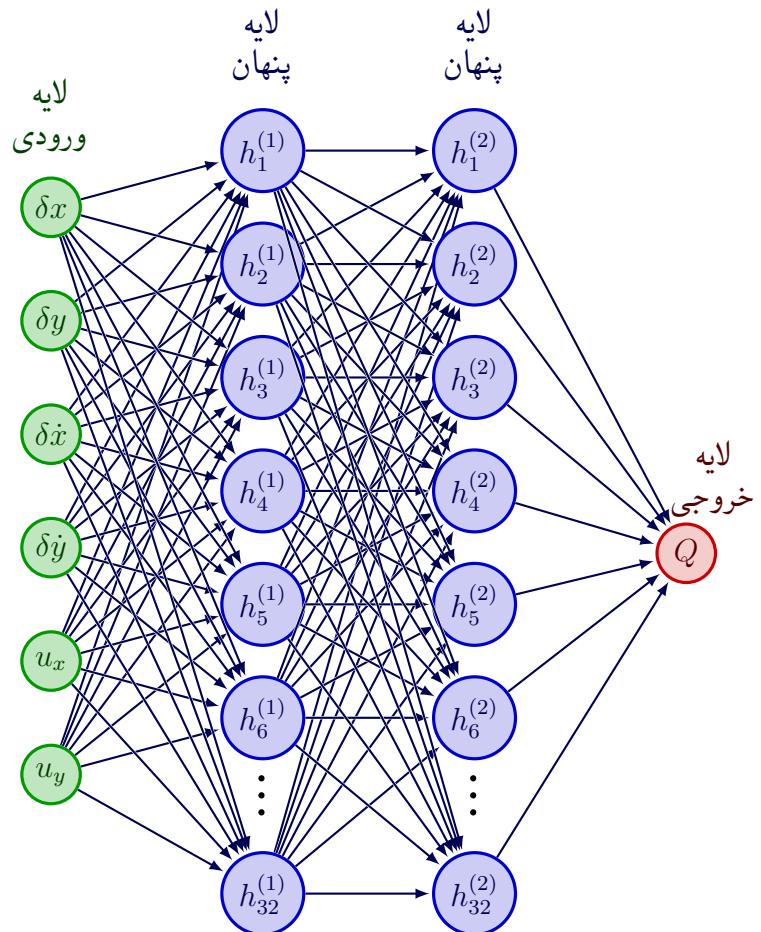
نام پارامتر	مقدار	نام پارامتر	مقدار
گام در هر دوره یادگیری	30 000	تعداد دوره‌های یادگیری	100
ضریب تنزیل ( $\gamma$ )	0.99	ضریب برش ratio clip	0.2
نرخ یادگیری سیاست	$3 \times 10^{-4}$	نرخ یادگیری تابع ارزش	$10^{-3}$
تعداد تکرار آموزش سیاست	80	تعداد تکرار آموزش ارزش	80
اندازه شبکه‌ی Actor	$(2^5, 2^5)$	تابع فعال‌سازی Actor	ReLU
اندازه شبکه‌ی Critic	$(2^5, 2^5)$	تابع فعال‌سازی Critic	ReLU

این الگوریتم‌ها به دلیل توانایی در مدیریت فضاهای پیوسته و عملکرد مؤثر در محیط‌های پیچیده انتخاب شده‌اند.

در شکل‌های ۱-۵ و ۲-۵ ساختار شبکه‌های Actor و Critic آورده شده است.



شکل ۱-۵: ساختار شبکه عصبی سیاست



شکل ۲-۵: ساختار شبکه عصبی نقاد

۲-۲-۵ فرآیند آموزش

رویه آموزش با PyTorch و اجرای Cuda به صورت زیر انجام شده است:

۱. گرددآوری تجربه‌ی اولیه با سیاست تصادفی تا رسیدن به گام شروع به روزرسانی برای پرشدن اولیه‌ی مخزن تجربه.
  ۲. حلقه‌ی یادگیری: در هر گام، اجرای کنش، ذخیره‌ی چهارتایی‌ها  $(s, a, r, s')$  (و در صورت نیاز  $d$  برای پایان اپیزود) در مخزن تجربه با ظرفیت  $10^6$ .
  ۳. نمونه‌گیری دسته داده و به روزرسانی Critic‌ها با هدف‌های حاوی شبکه‌های هدف و میانگین پلیاک؛ در استفاده از دو شبکه Q مستقل و هدف‌های کمینه شده.
  ۴. به روزرسانی Actor: در TD3/DDPG بیشینه‌سازی  $\mathbb{E}_s[Q(s, \pi_\theta(s))]$  و در SAC بیشینه‌سازی بازگشت انتروپی‌دار؛ در PPO به روزرسانی برش خورده با نسبت احتمال.
  ۵. تکنیک‌های پایداری: Target networks، reward/observation normalization، هموارسازی ۵
  - هدف gradient clipping، TD3، TD3 در صورت نیاز، و بذردهی ثابت برای تکرار پذیری.
  ۶. ارزیابی دوره‌ای: اجرای چند اپیزود آزمون بدون نویز کنش و ثبت بازگشت، نرخ موفقیت و واریانس. برای جلوگیری از بیش‌برازش و همگرایی زودرس، از نویز کاوش کنش و هموارسازی سیاست هدف (در TD3) استفاده شده است. معیار توقف زمانی فعال می‌شود که نرخ موفقیت آزمون در چند پنجره‌ی پیاپی از ۹۰٪ عبور کند و واریانس بازگشت کاهش یابد.

بھینہ سازی و پس انتشار گر ادیان

محاسبه‌ی گرادیان‌ها با autograd انجام شده‌است. به روزرسانی پارامترها با Adam [۶۳] بوده است که در عمل نسبت به گرادیان نزولی ساده‌پایدارتر است:

$$\begin{aligned} g_t &= \nabla_w L_t, \quad m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t, \quad v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \\ \hat{m}_t &= \frac{m_t}{1 - \beta_1^t}, \quad \hat{v}_t = \frac{v_t}{1 - \beta_2^t}, \quad w_{t+1} = w_t - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \end{aligned} \quad (\Phi-\Omega)$$

که در آن  $\eta$  نرخ یادگیری،  $\beta_1, \beta_2$  ضرایب مومنتوم (0.9, 0.999) و  $\epsilon$  برای پایدارسازی عددی است. به صورت مفهومی، زنجیره گرادیان نیز برقرار است:

$$\nabla_w L = \frac{\partial L}{\partial y} \frac{\partial y}{\partial w} \quad (\textcircled{5}-\textcircled{6})$$

در این رابطه:

- $L_t$ : مقدار تابع هزینه (Loss) در گام زمانی  $t$ .
- $w_t$ : بردار وزن‌ها یا پارامترهای مدل در گام  $t$ .
- $g_t = \nabla_w L_t$ : گرادیان تابع هزینه نسبت به پارامترها در زمان  $t$ .
- $m_t$ : میانگین نمایی گرادیان‌ها (مومنتوم مرتبه اول) که حافظه‌ای از جهت گرادیان‌ها ایجاد می‌کند.
- $v_t$ : میانگین نمایی مربعات گرادیان‌ها (مومنتوم مرتبه دوم) که بزرگی تغییرات گرادیان را ثبت می‌کند.
- $\hat{m}_t, \hat{v}_t$ : نسخه‌های اصلاح شده‌ی بایاس برای  $m_t$  و  $v_t$  به منظور پایدارسازی در مراحل اولیه.
- $\eta$ : نرخ یادگیری (Learning Rate) که اندازه‌ی گام به روزرسانی وزن‌ها را مشخص می‌کند.
- $\beta_1, \beta_2$ : ضرایب کاهش (Decay Rates) برای میانگین‌گیری نمایی؛ مقادیر معمول آن‌ها به ترتیب 0.9 و 0.999 است.
- $\epsilon$ : یک مقدار بسیار کوچک (معمولًاً  $10^{-8}$ ) برای جلوگیری از تقسیم بر صفر و افزایش پایداری عددی.

الگوریتم Adam به این صورت عمل می‌کند که همزمان از میانگین مرتبه‌ی اول ( $m_t$ ) برای جهت حرکت و از میانگین مرتبه‌ی دوم ( $v_t$ ) برای تنظیم نرخ یادگیری هر پارامتر استفاده می‌کند. در نتیجه هم از نوسانات شدید جلوگیری می‌شود و هم فرآیند همگرایی سرعت می‌گیرد.

از دیدگاه محاسبه‌ی گرادیان، زنجیره‌ی مشتق‌گیری (قاعده‌ی زنجیره‌ای) نیز برقرار است:

$$\nabla_w L = \frac{\partial L}{\partial y} \frac{\partial y}{\partial w} \quad (6-5)$$

که در آن  $y$  خروجی لایه یا شبکه است. این فرمول مبنای پساننتشار خطا (Backpropagation) در شبکه‌های عصبی محسوب می‌شود و باعث می‌گردد که گرادیان تابع هزینه نسبت به تمامی پارامترها به صورت کارآمد محاسبه شود.

## فصل ۶

# یادگیری تقویتی چندعاملی

کاربردهای پیچیده در یادگیری تقویتی نیازمند اضافه کردن چندین عامل<sup>۱</sup> برای انجام همزمان وظایف مختلف هستند. با این حال، افزایش تعداد عامل‌ها چالش‌هایی در مدیریت تعاملات میان آن‌ها به همراه دارد. در این فصل، براساس مسئله بهینه‌سازی برای هر عامل، مفهوم تعادل نش<sup>۲</sup> 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. براساس 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. در این فصل ابتدا در بخش ۱-۶ مفاهیم اولیه یادگیری تقویتی چندعاملی معرفی می‌شوند، سپس در بخش ۲-۶ انواع بازی‌ها و تعادل نش مورد بررسی قرار می‌گیرند. الگوریتم‌های مختلف یادگیری تقویتی چندعاملی شامل MA-DDPG در بخش ۳-۶، MA-TD3 در بخش ۴-۶، MA-SAC در بخش ۵-۶ و MA-PPO در بخش ۶-۶ معرفی و بررسی شده‌اند.

## ۱-۶ تعاریف و مفاهیم اساسی

یادگیری تقویتی چندعاملی<sup>۳</sup> به بررسی چگونگی یادگیری و تصمیم‌گیری چندین عامل مستقل در یک محیط مشترک پرداخته می‌شود. مفاهیم پایه‌ای یادگیری تقویتی در بخش ۱-۴ ارائه شده‌اند و در اینجا تنها مباحث کلی و موردنیاز برای MARL بیان می‌شوند. برای تحلیل دقیق و درک بهتر این حوزه، اجزای اصلی آن شامل

<sup>1</sup>Multi-Agent

<sup>2</sup>Nash Equilibrium

<sup>3</sup>Multi-Agent Reinforcement Learning (MARL)

عامل، سیاست و مطلوبیت<sup>۴</sup> در نظر گرفته می‌شوند که در ادامه به صورت مختصر و منسجم تشریح می‌گردد.

• عامل: یک موجودیت مستقل به عنوان عامل تعریف می‌شود که به صورت خودمختار با محیط تعامل کرده و بر اساس مشاهدات رفتار سایر عامل‌ها، سیاست‌هاییش انتخاب می‌گردد تا سود حداکثر یا ضرر حداقل حاصل شود. در سناریوهای مورد بررسی، چندین عامل به صورت مستقل عمل می‌کنند؛ اما اگر تعداد عامل‌ها به یک کاهش یابد، MARL به یادگیری تقویتی معمولی تبدیل می‌شود.

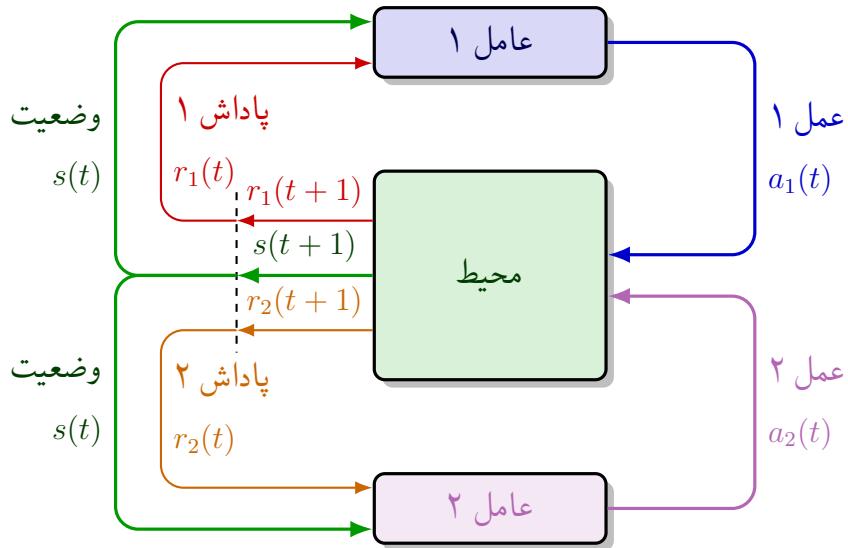
• سیاست: برای هر عامل در MARL، سیاستی خاص در نظر گرفته می‌شود که به عنوان روشی برای انتخاب اقدامات بر اساس وضعیت محیط و رفتار سایر عامل‌ها تعریف می‌گردد. این سیاست‌ها با هدف به حداکثر رساندن سود و به حداقل رساندن هزینه طراحی شده و تحت تأثیر محیط و سیاست‌های دیگر عامل‌ها قرار می‌گیرند.

• 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.

---

<sup>4</sup>Utility



شکل ۱-۶: حلقه تعامل عامل‌های یادگیری تقویتی چند عاملی با محیط

## ۲-۶ 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 بخش، دو مفهوم کلیدی نظریه‌ی بازی‌ها، یعنی تعادل نش و بازی‌های مجموع صفر، بررسی می‌شوند.

### ۱-۲-۶ تعادل نش

تعادل نش<sup>۵</sup> یکی از بنیادی‌ترین مفاهیم در نظریه‌ی بازی‌ها است که توسط جان نش در سال ۱۹۵۰ معرفی شد. این مفهوم به ترکیب<sup>۶</sup> سیاست‌ها اشاره دارد که در آن هیچ بازیکنی نمی‌تواند با تغییر یک‌جانبه سیاست خود، سود بیشتری به دست آورد (در حالی‌که سیاست‌های سایر بازیکنان ثابت است).

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

<sup>5</sup>Nash Equilibrium

<sup>6</sup>Profile

s داشته باشیم:

$$u_i(\pi_i^*, \pi_{-i}^*, s) \geq u_i(\pi_i, \pi_{-i}^*, s) \quad (1-6)$$

در اینجا،  $\pi_{-i}^*$  نشان‌دهندهٔ سیاست‌های همهٔ بازیکنان به جز بازیکن i است. در ادامه‌ی این پژوهش و به‌منظور به‌کارگیری چارچوب نظریه‌ی بازی در یادگیری تقویتی، مطلوبیت هر عامل به‌صورت برابر با تابع ارزش او در حالت s در نظر گرفته می‌شود:  $u_i(\pi_i, \pi_{-i}, s) = V_i^{\pi_i, \pi_{-i}}(s)$ .

## ۲-۶ بازی مجموع صفر

بازی‌های مجموع صفر<sup>۷</sup> دسته‌ای از بازی‌ها هستند که در آن‌ها تابع ارزش یک بازیکن دقیقاً برابر با ضرر بازیکن دیگر است؛ از این‌رو، مجموع ارزش‌های همهٔ بازیکنان در هر وضعیت صفر خواهد بود.

- **تعريف بازی مجموع صفر:**

در یک بازی دو نفره، اگر تابع ارزش حالت (value) بازیکن اول  $V_1^{(\pi_1, \pi_2)}(s)$  و بازیکن دوم  $V_2^{(\pi_1, \pi_2)}(s)$  بازیکن اول (value) بازیکن دوم به‌گونه‌ای باشد که:

$$V_1^{(\pi_1, \pi_2)}(s) + V_2^{(\pi_1, \pi_2)}(s) = 0 \implies V_1^{(\pi_1, \pi_2)}(s) = -V_2^{(\pi_1, \pi_2)}(s), \quad (2-6)$$

آنگاه آن بازی را بازی مجموع صفر می‌نامیم.

به‌طور مشابه، اگر تابع ارزش-عمل برای دو بازیکن را با  $(Q_1^{(\pi_1, \pi_2)}(s, a_1, a_2), Q_2^{(\pi_1, \pi_2)}(s, a_1, a_2))$  نشان دهیم، باید برقرار باشد:

$$Q_1^{(\pi_1, \pi_2)}(s, a_1, a_2) + Q_2^{(\pi_1, \pi_2)}(s, a_1, a_2) = 0 \implies Q_1^{(\pi_1, \pi_2)}(s, a_1, a_2) = -Q_2^{(\pi_1, \pi_2)}(s, a_1, a_2). \quad (3-6)$$

- **سیاست بهینه در بازی مجموع صفر:**

در این بازی‌ها، هر بازیکن سیاستی را برمی‌گزیند که تابع ارزش خود را در برابر بهترین پاسخ حریف بیشینه کند؛ این انتخاب در نهایت به تعادل نش منجر می‌شود.

به‌صورت تابع ارزش حالت:

$$V_1^*(s) = \max_{\pi_1} \min_{\pi_2} V_1^{(\pi_1, \pi_2)}(s), \quad (4-6)$$

$$V_2^*(s) = \max_{\pi_2} \min_{\pi_1} V_2^{(\pi_1, \pi_2)}(s). \quad (5-6)$$

---

<sup>7</sup>Zero-Sum Games

و به صورت تابع ارزش-عمل:

$$Q_1^*(s, a_1, a_2) = \max_{\pi_1} \min_{\pi_2} Q_1^{(\pi_1, \pi_2)}(s, a_1, a_2), \quad (6-6)$$

$$Q_2^*(s, a_1, a_2) = \max_{\pi_2} \min_{\pi_1} Q_2^{(\pi_1, \pi_2)}(s, a_1, a_2). \quad (7-6)$$

- **تابع پاداش:** تابع پاداش در بازی‌های دوسویه مجموع صفر باید به‌گونه‌ای طراحی شود که پاداش لحظه‌ای دو عامل در هر گام جمعاً صفر باشد. در ادامه ساختار پاداش عامل<sub>1</sub> مشابه قالب تک‌عاملی تعریف می‌شود و پاداش عامل<sub>2</sub> به صورت منفی آن اخذ می‌گردد.

- پاداش نهایی برای دستیابی به هدف عامل<sub>1</sub>: در صورت رسیدن به هدف عامل<sub>1</sub>، شبیه‌سازی پایان یافته و پاداش بزرگ مثبت به او داده می‌شود.

- جریمه نهایی برای دور شدن عامل<sub>1</sub>: اگر عامل<sub>1</sub> از محدوده مجاز خود خارج شود، شبیه‌سازی خاتمه یافته و جریمه بزرگ منفی اعمال می‌گردد.

- جریمه برای مصرف سوخت عامل<sub>1</sub>: استفاده بیش از حد از پیشرانه برای عامل<sub>1</sub> با جریمه همراه است.

- جریمه برای انحراف از مسیر مرجع عامل<sub>1</sub>: انحراف از مسیر مرجع عامل<sub>1</sub> باعث دریافت جریمه متناسب می‌شود.

تابع پاداش عامل<sub>1</sub> به صورت زیر تعریف می‌شود:

$$r_1(s, a_1, a_2) = r_{\text{thrust},1}(a_1) + r_{\text{thrust},1}(a_2) + r_{\text{reference},1}(s) + r_{\text{terminal},1}(s)$$

که در آن مؤلفه‌ها عبارتند از:

$$r_{\text{thrust},1}(a_1) = -k_1 \cdot |a_1| \quad (8-6)$$

$$r_{\text{thrust},1}(a_2) = -k_2 \cdot |a_2| \quad (9-6)$$

$$r_{\text{reference},1}(s) = -k_3 \cdot d_1(s, s_{\text{ref},1}) \quad (10-6)$$

$$r_{\text{terminal},1}(s) = \begin{cases} +R_{\text{goal},1} & \text{if } s \in S_{\text{goal},1} \\ -R_{\text{fail},1} & \text{if } d_1(s, s_{\text{ref},1}) > \epsilon_1 \\ 0 & \text{otherwise} \end{cases} \quad (11-6)$$

برای تضمین خاصیت مجموع صفر، پاداش عامل ۲ را در هر گام به صورت زیر تعریف می‌کنیم:

$$r_2(s, a_1, a_2) = -r_1(s, a_1, a_2),$$

بنابراین با افق و ضریب تنزیل یکسان، روابط (۲-۶) و (۳-۶) نیز برقرار خواهند بود.

در این رابطه:

۱.  $R_{\text{goal},1}$ : پاداش بزرگ مثبت برای دستیابی عامل ۱ به هدف.

۲.  $R_{\text{fail},1}$ : جریمه بزرگ منفی برای خروج عامل ۱ از محدوده مجاز.

۳.  $d_1(s, s')$ : فاصله مرتبط با عامل ۱ اقلیدسی بین دو وضعیت.

ضرایب  $k_1, k_2, k_3$  برای تنظیم تعادل بین جریمه پیشرانه عامل ۱، جریمه پیشرانه عامل ۲، و جریمه انحراف از مسیر مرجع استفاده می‌شوند. بدلیل تعریف  $-r_1 = r_2$ ، جمع پاداش‌ها در هر گام صفر بوده و مقدار بازی یکتا و با تعادل نش در راهبردهای مختلط سازگار است.

بر پایه‌ی قضیه‌ی کمینه‌بیشینه‌ی فون‌نویمان، در بازی‌های دوسویه‌ی مجموع صفر متناهی داریم:

$$\max_{\pi_1} \min_{\pi_2} V_1^{(\pi_1, \pi_2)}(s) = \min_{\pi_2} \max_{\pi_1} V_1^{(\pi_1, \pi_2)}(s),$$

که وجود تعادل نش در راهبردهای مختلط و یکتایی مقدار بازی را تضمین می‌کند.

## ۳-۶ چالش‌های استایی در یادگیری تقویتی تک‌عاملی

در محیط‌های چندعاملی، رفتار سایر عامل‌ها هم‌زمان با فرایند یادگیری در حال تغییر است؛ بنابراین توزیع انتقال  $p(s'|s, a)$  از دید هر عامل ثابت باقی نمی‌ماند. در نتیجه، محیط از منظر عامل غیراستا تلقی می‌شود، زیرا یکی از فرض‌های بنیادی در روش‌های یادگیری تقویتی تک‌عاملی مانند DDPG، ثبات دینامیک محیط در طول یادگیری است. نقض این فرض باعث ناپایداری در برآوردتابع ارزش، انحراف در بهروزرسانی گرادیان سیاست و در نهایت همگرایی ضعیف یا شکست فرایند یادگیری می‌گردد.

برای مقابله با این مسئله، چارچوب Centralized Training with Decentralized Execution (CTDE) از رویکرد MA-DDPG بهره می‌گیرد. در مرحله آموزش، هر عامل علاوه بر مشاهدات محلی خود، به اطلاعات حالت و اقدامات سایر عامل‌ها نیز دسترسی دارد و منتقد هر عامل تابع ارزش را به درستی تخمین می‌زند. این ساختار به منتقد اجازه می‌دهد تغییرات سیاست سایر عامل‌ها را در مدل ارزش منعکس کرده و

نوسانات ناشی از غیراستایی محیط را کاهش دهد. بدین ترتیب، تابع ارزش در طول یادگیری نسبت به تغییر رفتار عامل‌های دیگر حساسیت کمتری خواهد داشت.

در مرحله اجرا، هر عامل تنها بر اساس مشاهدات محلی خود عمل می‌کند و سیاست‌ها به صورت غیرمتکر اعمال می‌شوند. این تفکیک آموزش/اجرا موجب می‌شود الگوریتم ضمن بهره‌گیری از مزیت اطلاعات کامل در آموزش، همچنان برای کاربرد در محیط‌های واقعی و قابل استقرار روی سامانه‌های درون‌برد مناسب باقی بماند.

مزیت اصلی CTDE در MA-DDPG را می‌توان به صورت زیر خلاصه کرد:

- کاهش غیراستایی ادراک شده از طریق یادگیری تابع ارزش با اطلاعات جامع محیط؛
- بهبود پایداری و همگرایی به دلیل کاهش خطای Non-Stationary Target Shift؛
- امکان اجرای سبک و مستقل عامل‌ها بدون نیاز به اشتراک اطلاعات در زمان اجرا.

شواهد تجربی پژوهش حاضر نیز نشان می‌دهد که نسخه MA-DDPG نسبت به DDPG تک‌عاملی در سناریوهای دارای اغتشاشات تصادفی و عدم قطعیت‌های مدل، از پایداری و همگرایی به مرتبه بهتری برخوردار است.

در محیط‌های چندعاملی، سیاست هر عامل مدام در حال تغییر است، که باعث می‌شود محیط از دید هر عامل غیرایستا<sup>8</sup> شود. این مسئله چالش بزرگی برای الگوریتم‌های یادگیری تقویتی تک‌عاملی مانند DDPG ایجاد می‌کند، زیرا فرض ایستایی محیط را نقض می‌کند.

MA-DDPG با استفاده از رویکرد آموزش متکر، اجرای غیرمتکر<sup>9</sup> این مشکل را حل می‌کند. در این رویکرد، هر عامل در زمان آموزش به اطلاعات کامل محیط دسترسی دارد، اما در زمان اجرا تنها از مشاهدات محلی خود استفاده می‌کند.

## ۴-۲-۶ ضرورت استفاده از یادگیری تقویتی چندعاملی

در نگاه اول بمنظر می‌رسد که می‌توان مسئله هدایت فضایی را در چارچوب یادگیری تقویتی تک‌عاملی مدل نمود و تمامی عدم قطعیت‌ها و اغتشاشات محیطی را صرفاً به صورت نویز در دینامیک سیستم در نظر گرفت. با این حال، ماهیت دینامیک سه‌جسمی و حساسیت بالای مسیرهای ناپایدار در حوالی نقاط لاغرانژ، باعث می‌شود که تعامل میان فضایی و محیط، از منظر کنترلی بیشتر شبیه یک بازی دیفرانسیلی خصم‌انه باشد تا یک محیط ایستا با نویز تصادفی ساده. به عبارت دیگر، مزاحمت‌های محیطی، خطای تراستر، تأخیر حسگر

<sup>8</sup>Non-stationary

<sup>9</sup>Centralized Training, Decentralized Execution

و عدم تطابق مدل، همگی در جهت تضعیف پایداری و افزایش انحراف مسیر عمل می‌کنند و می‌توان آن‌ها را به صورت یک عامل حریف مدل نمود که در پی بدتر کردن کارکرد سامانه است.

در روش‌های کلاسیک یادگیری تقویتی تک‌عاملی نظریه DDPG، اغتشاشات معمولاً به صورت بخشی از نویز فرایند در نظر گرفته می‌شوند و عامل تنها با هدف بیشینه‌سازی بازده خود آموزش می‌بیند. این رویکرد اگرچه در شرایط بهینه عملکرد مناسبی دارد، اما تضمین قوی در برابر بدترین حالت عدم قطعیت‌ها ارائه نمی‌کند و سیاست به دست آمده ممکن است نسبت به سناریوهای شدید اغتشاش و خطای مدل بسیار حساس باشد. در مقابل، صورت‌بندی مسئله به عنوان یک بازی مجموع صفر فضای‌پیما-مزاحم و استفاده از چارچوب Multi-Agent RL این امکان را فراهم می‌سازد که:

- عامل کنترل‌کننده، سیاست خود را نه تنها برای یک مدل بهینه، بلکه در برابر حریفی که فعالانه به دنبال بدتر کردن عملکرد است، بهینه کند؛
- مفهوم بهینگی مقاوم (Robust Optimality) به صورت درونی در فرایند یادگیری لحاظ شود و سیاست هدایت به گونه‌ای شکل گیرد که در سناریوهای بدترین حالت نیز پایداری مسیر و محدودیت‌های سوخت را حفظ نماید؛
- به کمک آموزش متمنک و اجرای غیرمتمنک (CTDE)، تابع ارزش با استفاده از اطلاعات کامل محیط و رفتار حریف تقریب زده شود و بدین ترتیب حساسیت به غیراستایی ناشی از تغییر سیاست‌ها کاهش یابد.

بدین ترتیب، استفاده از MARL در این پژوهش صرفاً انتخابی تزئینی یا پیچیده‌سازی بی‌دلیل مسئله نیست، بلکه مستقیماً از ماهیت بازی‌گونه و خصمانه‌ی تعامل فضای‌پیما با محیط در دینامیک سه‌جسمی ناشی می‌شود. نتایج عددی ارائه شده در فصل ۷ نیز نشان می‌دهند که نسخه‌های چند‌عاملی پیشنهاد شده، در مقایسه با نسخه‌های تک‌عاملی متناظر، در سناریوهای دارای تأخیر حسگر، نویز عملکرد و عدم تطابق مدل، انحراف مسیر کمتر و مصرف سوخت بهینه‌تری را به همراه دارند. این تفاوت عملکرد، ضرورت استفاده از چارچوب یادگیری تقویتی چند‌عاملی را برای دستیابی به هدایت مقاوم و قابل‌اتکا در محیط‌های ناپایدار سه‌جسمی توجیه می‌کند.

## ۳-۶ گرادیان سیاست عمیق قطعی چند‌عاملی

گرادیان سیاست عمیق قطعی چند‌عاملی<sup>۱۰</sup> توسعه‌ای از الگوریتم DDPG برای محیط‌های چند‌عاملی است. در این بخش، به بررسی این الگوریتم در چارچوب بازی‌های دو‌عاملی مجموع صفر می‌پردازیم که در آن مجموع

<sup>10</sup>Multi-Agent Deep Deterministic Policy Gradient (MA-DDPG)

پاداش‌های دو عامل همواره صفر است (آنچه یک عامل به دست می‌آورد، عامل دیگر از دست می‌دهد).

### ۱-۳-۶ معماری MA-DDPG در بازی‌های مجموع‌صفر

در یک بازی دو‌عاملی مجموع‌صفر، دو عامل با نمادهای ۱ و ۲ نشان داده می‌شوند. هر عامل دارای شبکه‌های منحصر به فرد خود است:

- شبکه‌های بازیگر:  $\mu_{\theta_1}(o_1)$  و  $\mu_{\theta_2}(o_2)$  که مشاهدات محلی  $o_1$  و  $o_2$  را به اعمال  $a_1$  و  $a_2$  نگاشت می‌کنند.
- شبکه‌های منتقد:  $Q_{\phi_1}(o_1, a_1, a_2)$  و  $Q_{\phi_2}(o_2, a_2, a_1)$  که ارزش حالت-عمل را با توجه به مشاهدات و اعمال تمام عامل‌ها تخمین می‌زنند.
- شبکه‌های هدف: مشابه DDPG، برای پایدار کردن آموزش از شبکه‌های هدف استفاده می‌شود.

در بازی‌های مجموع‌صفر، پاداش‌ها رابطه  $r_1 + r_2 = 0$  دارند که در آن  $r_1$  و  $r_2$  پاداش‌های دریافتی عامل‌ها هستند. در نتیجه،  $r_2 = -r_1$  است که نمایانگر تضاد کامل منافع بین عامل‌هاست.

### ۲-۳-۶ آموزش MA-DDPG در بازی‌های مجموع‌صفر

فرایند آموزش MA-DDPG برای بازی‌های مجموع‌صفر به شرح زیر است:

یادگیری تابع  $Q$

برای هر عامل  $i \in \{1, 2\}$ ، تابع  $Q$  با کمینه کردن خطای میانگین مربعات بلمن به روزرسانی می‌شود:

$$L(\phi_i, \mathcal{D}) = \underset{(o, a, r_i, o', d) \sim \mathcal{D}}{\text{E}} \left[ \left( Q_{\phi_i}(o_i, a_1, a_2) - y_i \right)^2 \right] \quad (12-6)$$

که در آن  $(o_1, o_2) = o$  بردار مشاهدات،  $(a_1, a_2) = a$  بردار اعمال، و  $y_i$  هدف برای عامل  $i$  است:

$$y_i = r_i + \gamma(1 - d)Q_{\phi_i, \text{targ}}(o'_i, \mu_{\theta_1, \text{targ}}(o'_1), \mu_{\theta_2, \text{targ}}(o'_2)) \quad (13-6)$$

در این پژوهش منتقد هر عامل به اعمال همه عامل‌ها دسترسی دارد. در بازی‌های مجموع‌صفر، عامل شماره ۲ جهت مخالف هدف عامل ۱ را دنبال می‌کند.

## یادگیری سیاست

سیاست هر عامل با بیشینه کردن تابع  $Q$  مربوط به آن عامل بهروزرسانی می‌شود:

$$\max_{\theta_i} \mathbb{E}_{o \sim \mathcal{D}} [Q_{\phi_i}(o_i, \mu_{\theta_i}(o_i), \mu_{\theta_{-i}}(o_{-i}))] \quad (14-6)$$

که در آن  $i$ - نشان‌دهندهی عامل مقابل است. با توجه به ماهیت بازی مجموع صفر، هر عامل تلاش می‌کند تا مطلوبیت خود را افزایش دهد، در حالی که مطلوبیت عامل دیگر به طور همزمان کاهش می‌یابد.

## شبکه‌های هدف و بافر تجربه

مشابه DDPG، برای پایدار کردن آموزش، شبکه‌های هدف با میانگین‌گیری پولیاک بهروزرسانی می‌شوند:

$$\begin{aligned} \phi_{i,\text{targ}} &\leftarrow \rho \phi_{i,\text{targ}} + (1 - \rho) \phi_i \\ \theta_{i,\text{targ}} &\leftarrow \rho \theta_{i,\text{targ}} + (1 - \rho) \theta_i \end{aligned}$$

همچنین، از یک بافر تکرار بازی مشترک برای ذخیره تجربیات استفاده می‌شود که شامل وضعیت‌ها، اعمال و پاداش‌های همه عامل‌هاست.

## ۳-۳-۶ اکتشاف در MA-DDPG

اکتشاف در MA-DDPG مشابه DDPG است، اما برای هر عامل به طور جداگانه اعمال می‌شود. در طی آموزش، به اعمال هر عامل نویز اضافه می‌شود:

$$a_i = \text{clip}(\mu_{\theta_i}(o_i) + \epsilon_i, a_{\text{Low}}, a_{\text{High}}) \quad (15-6)$$

که در آن  $\epsilon$  نویز اضافه شده به عامل  $i$  است.

## ۴-۳-۶ شبکه‌کد MA-DDPG برای بازی‌های دو‌عاملی مجموع صفر

در این بخش، شبکه‌کد الگوریتم MA-DDPG پیاده‌سازی شده آورده شده است. در این پژوهش الگوریتم <sup>۵</sup> در محیط پایتون با استفاده از کتابخانه PyTorch [۶۰] پیاده‌سازی شده است.

---

## الگوریتم ۵ عامل گرادیان سیاست عمیق قطعی چندعاملی

---

- وروودی: پارامترهای اولیه سیاست عامل‌ها  $(\theta_1, \theta_2)$ ، پارامترهای تابع  $Q(\phi_1, \phi_2)$ ، بافر تکرار بازی خالی ( $\mathcal{D}$ )
- ۱: پارامترهای هدف را برابر با پارامترهای اصلی قرار دهید:  $\phi_{i,\text{targ}} \leftarrow \phi_i$ ,  $\theta_{i,\text{targ}} \leftarrow \theta_i$  برای  $i \in \{1, 2\}$
  - ۲: تا وقتی همگرایی رخ دهد:
  - ۳: مشاهدات  $(o_1, o_2)$  را دریافت کنید
  - ۴: برای هر عامل  $i$ ، عمل  $a_i = \text{clip}(\mu_{\theta_i}(o_i) + \epsilon_i, a_{\text{Low}}, a_{\text{High}})$  را انتخاب کنید، به‌طوری که  $\epsilon_i \sim \mathcal{N}$  است
  - ۵: اعمال  $(a_1, a_2)$  را در محیط اجرا کنید
  - ۶: مشاهدات بعدی  $(o'_1, o'_2)$ ، پاداش‌ها  $(r_1, r_2 = -r_1)$  و سیگنال پایان  $d$  را دریافت کنید
  - ۷: تجربه  $(o_1, o_2, a_1, a_2, r_1, r_2, o'_1, o'_2, d)$  را در بافر  $\mathcal{D}$  ذخیره کنید
  - ۸: اگر  $d = 1$  است، وضعیت محیط را بازنشانی کنید
  - ۹: اگر زمان بهروزرسانی فرا رسیده است:
  - ۱۰: به ازای هر تعداد بهروزرسانی:
  - ۱۱: یک دسته تصادفی از تجربیات،  $\{(o, a, r_1, r_2, o', d)\}$  از  $\mathcal{D}$  نمونه‌گیری کنید اهداف را محاسبه کنید:

$$y_1 = r_1 + \gamma(1 - d)Q_{\phi_1,\text{targ}}(o'_1, \mu_{\theta_1,\text{targ}}(o'_1), \mu_{\theta_2,\text{targ}}(o'_2))$$

$$y_2 = r_2 + \gamma(1 - d)Q_{\phi_2,\text{targ}}(o'_2, \mu_{\theta_2,\text{targ}}(o'_2), \mu_{\theta_1,\text{targ}}(o'_1))$$

تابع  $Q$  را با نزول گرادیان بهروزرسانی کنید: :۱۲

$$\nabla_{\phi_1} \frac{1}{|B|} \sum_{(o, a, r_1, r_2, o', d) \in B} (Q_{\phi_1}(o_1, a_1, a_2) - y_1)^2$$

$$\nabla_{\phi_2} \frac{1}{|B|} \sum_{(o, a, r_1, r_2, o', d) \in B} (Q_{\phi_2}(o_2, a_2, a_1) - y_2)^2$$

سیاست‌ها را با صعود گرادیان بهروزرسانی کنید: :۱۳

$$\nabla_{\theta_1} \frac{1}{|B|} \sum_{o \in B} Q_{\phi_1}(o_1, \mu_{\theta_1}(o_1), a_2)$$

$$\nabla_{\theta_2} \frac{1}{|B|} \sum_{o \in B} Q_{\phi_2}(o_2, \mu_{\theta_2}(o_2), a_1)$$

شبکه‌های هدف را بهروزرسانی کنید: :۱۴

$$\phi_{1,\text{targ}} \leftarrow \rho \phi_{1,\text{targ}} + (1 - \rho) \phi_1$$

$$\phi_{2,\text{targ}} \leftarrow \rho \phi_{2,\text{targ}} + (1 - \rho) \phi_2$$

$$\theta_{1,\text{targ}} \leftarrow \rho \theta_{1,\text{targ}} + (1 - \rho) \theta_1$$

$$\theta_{2,\text{targ}} \leftarrow \rho \theta_{2,\text{targ}} + (1 - \rho) \theta_2$$


---

## ۵-۳ مزایای MA-DDPG در بازی‌های مجموع‌صفر

MA-DDPG چندین مزیت برای یادگیری در بازی‌های دوعلاملی مجموع‌صفر ارائه می‌دهد:

- **مقابله با غیرایستایی:** با استفاده از منتقدهایی که به اطلاعات کامل دسترسی دارند، مشکل غیرایستایی محیط از دید هر عامل حل می‌شود.
- **همگرایی بهتر:** در بازی‌های مجموع‌صفر، MA-DDPG معمولاً همگرایی بهتری نسبت به آموزش مستقل عامل‌ها با DDPG نشان می‌دهد.
- **یادگیری استراتژی‌های متقابل:** عامل‌ها می‌توانند استراتژی‌های متقابل پیچیده را یاد بگیرند که در آموزش مستقل امکان‌پذیر نیست.

در بازی‌های دوعلاملی مجموع‌صفر، این رویکرد به رقابت کامل بین عامل‌ها منجر می‌شود، که هر یک تلاش می‌کند بهترین استراتژی را در برابر استراتژی رقیب پیدا کند.

## ۴-۶ عامل گرادیان سیاست عمیق قطعی تاخیری دوگانه چندعلاملی

عامل گرادیان سیاست عمیق قطعی تاخیری دوگانه چندعلاملی<sup>۱۱</sup> توسعه‌ای از الگوریتم TD3 برای محیط‌های چندعلاملی است. در این بخش، به بررسی این الگوریتم در چارچوب بازی‌های چندعلاملی مجموع‌صفر می‌پردازیم که در آن ترکیب ویژگی‌های TD3 با رویکرد چندعلاملی MA-DDPG به پایداری و کارایی بیشتر در یادگیری منجر می‌شود.

## ۱-۴-۶ چالش‌های یادگیری تقویتی در محیط‌های چندعلاملی و راه حل MA-TD3

در محیط‌های چندعلاملی، عامل‌ها همزمان سیاست‌های خود را تغییر می‌دهند که باعث غیرایستایی محیط از دید هر عامل می‌شود. علاوه بر این، بیش‌برآورد تابع  $Q$  که در DDPG دیده می‌شود، در محیط‌های چندعلاملی می‌تواند تشدید شود.

MA-TD3 هر دو چالش را با ترکیب رویکردهای زیر حل می‌کند:

- **آموزش مرکز، اجرای غیرمرکز:** مشابه MA-DDPG، از منتقدهایی استفاده می‌کند که به اطلاعات کامل دسترسی دارند.

<sup>11</sup>Multi-Agent Twin Delayed Deep Deterministic Policy Gradient (MA-TD3)

- منتقدهای دوگانه: برای هر عامل، از دو شبکه منتقد استفاده می‌کند تا بیش‌برآورد تابع  $Q$  را کاهش دهد.

- بهروزرسانی‌های تاخیری سیاست: سیاست‌ها را با تواتر کمتری نسبت به منتقدها بهروزرسانی می‌کند.

## ۲-۴-۶ معماری MA-TD3 در بازی‌های مجموع‌صفر

در یک بازی چندعاملی مجموع‌صفر، هر عامل دارای شبکه‌های زیر است:

- شبکه بازیگر:  $\mu_{\theta_i}(o_i)$  که مشاهدات محلی  $o_i$  را به اعمال  $a_i$  نگاشت می‌کند.
- شبکه‌های منتقد دوگانه:  $Q_{\phi_{i,2}}(o_i, a_1, a_2)$  و  $Q_{\phi_{i,1}}(o_i, a_1, a_2)$  که ارزش حالت-عمل را تخمین می‌زنند.
- شبکه‌های هدف: برای پایدارسازی آموزش، از نسخه‌های هدف بازیگر و منتقدها استفاده می‌شود.

## ۳-۴-۶ آموزش MA-TD3

فرایند آموزش MA-TD3 به شرح زیر است:

یادگیری تابع  $Q$

برای هر عامل  $i \in \{1, 2\}$  و هر منتقد  $j \in \{1, 2\}$  با کمینه کردن خطای میانگین مربعات بلمن بهروزرسانی می‌شود:

$$L(\phi_{i,j}, \mathcal{D}) = \underset{(o,a,r_i,o',d) \sim \mathcal{D}}{\text{E}} \left[ \left( Q_{\phi_{i,j}}(o_i, a_1, a_2) - y_i \right)^2 \right] \quad (16-6)$$

که در آن  $y_i$  هدف برای عامل  $i$  است:

$$y_i = r_i + \gamma(1 - d) \min_{j=1,2} Q_{\phi_{i,j,\text{targ}}}(o'_i, \mu_{\theta_{1,\text{targ}}}(o'_1), \mu_{\theta_{2,\text{targ}}}(o'_2)) \quad (17-6)$$

استفاده از عملگر حداقل روی دو منتقد، بیش‌برآورد را کاهش می‌دهد که منجر به تخمین‌های محتاطانه‌تر و پایدارتر می‌شود.

## یادگیری سیاست با تاخیر

سیاست هر عامل با تاخیر (معمولًاً پس از هر دو بهروزرسانی منتقدها) و با بیشینه کردن تابع  $Q$  اول بهروزرسانی می‌شود:

$$\max_{\theta_i} \mathbb{E}_{o \sim \mathcal{D}} [Q_{\phi_{i,1}}(o_i, \mu_{\theta_i}(o_i), \mu_{\theta_{-i}}(o_{-i}))] \quad (18-6)$$

بهروزرسانی تاخیری سیاست اجازه می‌دهد تا منتقدها قبل از تغییر سیاست به مقادیر دقیق‌تری همگرا شوند.

## شبکه‌های هدف

مشابه TD3، شبکه‌های هدف با میانگین‌گیری پولیاک بهروزرسانی می‌شوند.

## ۴-۴-۶ اکتشاف در MA-TD3

اکتشاف در MA-TD3 با افروden نویز به اعمال هر عامل انجام می‌شود:

$$a_i = \text{clip}(\mu_{\theta_i}(o_i) + \epsilon_i, a_{\text{Low}}, a_{\text{High}}) \quad (19-6)$$

که در آن  $\epsilon_i \sim \mathcal{N}(0, \sigma_i)$  است و مقدار  $\sigma_i$  به مرور زمان کاهش می‌یابد.

## ۵-۴-۶ شبکه‌کد MA-TD3 برای بازی‌های چندعاملی مجموع صفر

در این بخش، شبکه‌کد الگوریتم MA-TD3 پیاده‌سازی شده آورده شده است. در این پژوهش الگوریتم [۶](#) در محیط پایتون با استفاده از کتابخانه PyTorch [\[۶۰\]](#) پیاده‌سازی شده است.

## الگوریتم ۶ عامل گرادیان سیاست عمیق قطعی تاخیری دوگانه چندعاملی

ورودی: پارامترهای اولیه سیاست عاملها  $(\phi_{1,1}, \phi_{1,2}, \phi_{2,1}, \phi_{2,2})$ ، پارامترهای توابع  $Q$ ، بافر تکرار بازی خالی  $(\mathcal{D})$

۱: پارامترهای هدف را برابر با پارامترهای اصلی قرار دهید:

$$j \in \{1, 2\} \quad i \in \{1, 2\} \quad \text{برای } \phi_{i,j,\text{targ}} \leftarrow \phi_{i,j}, \theta_{i,\text{targ}} \leftarrow \theta_i$$

۲: تا وقتی همگرایی رخ دهد:

۳: مشاهدات  $(o_1, o_2)$  را دریافت کنید

۴: برای هر عامل  $i$ ، عمل  $a_i = \text{clip}(\mu_{\theta_i}(o_i) + \epsilon_i, a_{\text{Low}}, a_{\text{High}})$  را انتخاب کنید، به طوری که  $\sim \epsilon_i$

۵: اعمال  $(a_1, a_2)$  را در محیط اجرا کنید است  $\mathcal{N}(0, \sigma_i)$

۶: مشاهدات بعدی  $(o'_1, o'_2)$ ، پاداشها  $(r_1, r_2 = -r_1)$  و سیگنال پایان  $d$  را دریافت کنید

۷: تجربه  $(o_1, o_2, a_1, a_2, r_1, r_2, o'_1, o'_2, d)$  را در بافر  $\mathcal{D}$  ذخیره کنید

۸: اگر  $d = 1$  است، وضعیت محیط را بازنشانی کنید

۹: اگر زمان بهروزرسانی فرا رسیده است:

۱۰: به ازای  $j$  در هر تعداد بهروزرسانی:

۱۱: یک دسته تصادفی از تجربیات،  $B = \{(o, a, r_1, r_2, o', d)\}$  از  $\mathcal{D}$  نمونهگیری کنید.

۱۲: اهداف را محاسبه کنید:

$$y_1 = r_1 + \gamma(1-d) \min_{k=1,2} Q_{\phi_{1,k},\text{targ}}(o'_1, \mu_{\theta_{1,\text{targ}}}(o'_1), \mu_{\theta_{2,\text{targ}}}(o'_2))$$

$$y_2 = r_2 + \gamma(1-d) \min_{k=1,2} Q_{\phi_{2,k},\text{targ}}(o'_2, \mu_{\theta_{2,\text{targ}}}(o'_2), \mu_{\theta_{1,\text{targ}}}(o'_1))$$

۱۳: توابع  $Q$  را با نزول گرادیان بهروزرسانی کنید:

$$\nabla_{\phi_{1,k}} \frac{1}{|B|} \sum_B (Q_{\phi_{1,k}}(o_1, a_1, a_2) - y_1)^2 \quad \text{برای } k = 1, 2$$

$$\nabla_{\phi_{2,k}} \frac{1}{|B|} \sum_B (Q_{\phi_{2,k}}(o_2, a_2, a_1) - y_2)^2 \quad \text{برای } k = 1, 2$$

۱۴: اگر باقیمانده  $j$  بر تاخیر سیاست برابر ۰ باشد:

۱۵: سیاستها را با صعود گرادیان بهروزرسانی کنید:

$$\nabla_{\theta_1} \frac{1}{|B|} \sum_{o \in B} Q_{\phi_{1,1}}(o_1, \mu_{\theta_1}(o_1), a_2)$$

$$\nabla_{\theta_2} \frac{1}{|B|} \sum_{o \in B} Q_{\phi_{2,1}}(o_2, \mu_{\theta_2}(o_2), a_1)$$

۱۶: شبکه‌های هدف را بهروزرسانی کنید:

$$\phi_{i,k,\text{targ}} \leftarrow \rho \phi_{i,k,\text{targ}} + (1-\rho) \phi_{i,k} \quad \text{برای } i, k \in \{1, 2\}$$

$$\theta_{i,\text{targ}} \leftarrow \rho \theta_{i,\text{targ}} + (1-\rho) \theta_i \quad \text{برای } i \in \{1, 2\}$$

## ۶-۴-۶ مزایای MA-TD3 در بازی‌های مجموع‌صفر

مزایای زیر را نسبت به MA-DDPG در بازی‌های چندعاملی مجموع‌صفر ارائه می‌دهد:

- پایداری بیشتر: با استفاده از منتقدهای دوگانه، بیش‌برآورد تابع  $Q$  که در محیط‌های غیرایستای چند-عاملی شدیدتر است، کاهش می‌یابد.
- یادگیری کارآمدتر: به روزسانی‌های تاخیری سیاست اجازه می‌دهد منتقدها به تخمین‌های دقیق‌تری دست یابند، که منجر به بهبود کیفیت یادگیری سیاست می‌شود.
- مقاومت در برابر نویز: ترکیب منتقدهای دوگانه با رویکرد آموزش متمرکز، مقاومت الگوریتم در برابر نویز و تغییرات محیط را افزایش می‌دهد.
- همگرایی بهتر: بهبودهای TD3 در کنار رویکرد چندعاملی، به همگرایی سریع‌تر و پایدارتر در بازی‌های رقابتی منجر می‌شود.

در مجموع، MA-TD3 ترکیبی از بهترین ویژگی‌های TD3 و MA-DDPG را ارائه می‌دهد که آن را به گزینه‌ای مناسب برای یادگیری سیاست‌های پیچیده در بازی‌های چندعاملی مجموع‌صفر تبدیل می‌کند.

## ۵-۶ عامل عملگر نقاد نرم چندعاملی

عامل عملگر نقاد نرم دواعمالی<sup>۱۲</sup> توسعه‌ای از الگوریتم SAC برای محیط‌های چندعاملی است. در این بخش، به بررسی این الگوریتم در چارچوب بازی‌های چندعاملی مجموع‌صفر می‌پردازیم که در آن ترکیب ویژگی‌های SAC با رویکرد چندعاملی به پایداری و کارایی بیشتر در یادگیری منجر می‌شود.

## ۱-۵-۶ چالش‌های یادگیری تقویتی در محیط‌های چندعاملی و راه حل MA-SAC

در محیط‌های چندعاملی، عامل‌ها هم‌مان سیاست‌های خود را تغییر می‌دهند که باعث غیرایستایی محیط از دید هر عامل می‌شود. علاوه بر این، چالش‌های مربوط به تعادل اکتشاف-بهره‌برداری در محیط‌های چندعاملی پیچیده‌تر است.

MA-SAC این چالش‌ها را با ترکیب رویکردهای زیر حل می‌کند:

<sup>12</sup>Multi-Agent Soft Actor-Critic (MA-SAC)

- آموزش متمرکز، اجرای غیرمتمرکز: مشابه MA-DDPG، از منتقدهایی استفاده می‌کند که به اطلاعات کامل دسترسی دارند.
- سیاست‌های تصادفی: برخلاف MA-DDPG و MA-TD3 که سیاست‌های قطعی دارند، MA-SAC از سیاست‌های تصادفی استفاده می‌کند.
- تنظیم آنتروپی: با استفاده از تنظیم آنتروپی، اکتشاف و همگرایی به سیاست‌های بهتر را بهبود می‌بخشد.
- منتقدهای دوگانه: برای هر عامل، از دو شبکه منتقد استفاده می‌کند تا بیش برآورد تابع  $Q$  را کاهش دهد.

## ۲-۵-۶ معماری MA-SAC در بازی‌های مجموع صفر

در یک بازی چندعاملی مجموع صفر، هر عامل دارای شیکه‌های زیر است:

- شبکه بازیگر:  $\pi_{\theta_i}(a_i|o_i)$  که توزیع احتمال اعمال را با توجه به مشاهدات محلی تعیین می‌کند.
- شبکه‌های منتقد دوگانه:  $Q_{\phi_{i,2}}(o_i, a_1, a_2)$  و  $Q_{\phi_{i,1}}(o_i, a_1, a_2)$  که ارزش حالت-عمل را تخمین می‌زنند.
- شبکه‌های هدف: برای پایدارسازی آموزش، از نسخه‌های هدف منتقدها استفاده می‌شود.

## ۳-۵-۶ آموزش MA-SAC

فرایند آموزش MA-SAC به شرح زیر است:

### یادگیری تابع $Q$

برای هر عامل  $i \in \{1, 2\}$  و هر منتقد  $j \in \{1, 2\}$ ، تابع  $Q$  با کمینه کردن خطای میانگین مربعات بلمن بهروزرسانی می‌شود:

$$L(\phi_{i,j}, \mathcal{D}) = \underset{(o, a, r_i, o', d) \sim \mathcal{D}}{\text{E}} \left[ \left( Q_{\phi_{i,j}}(o_i, a_1, a_2) - y_i \right)^2 \right] \quad (20-6)$$

که در آن  $y_i$  هدف برای عامل  $i$  است:

$$y_i = r_i + \gamma(1 - d) \left( \min_{j=1,2} Q_{\phi_{i,j}, \text{targ}}(o'_i, \tilde{a}'_1, \tilde{a}'_2) - \alpha_i \log \pi_{\theta_i}(\tilde{a}'_i | o'_i) \right) \quad (21-6)$$

که در آن  $(\cdot | o'_i) \sim \pi_{\theta_i}$  است. استفاده از عملگر حداقل روی دو منتقد، بیش برآورد را کاهش می‌دهد که منجر به تخمین‌های محتاطانه‌تر و پایدارتر می‌شود.

### یادگیری سیاست

سیاست هر عامل با بیشینه کردن ترکیبی ازتابع  $Q$  و آنتروپی به روزرسانی می‌شود:

$$\max_{\theta_i} \mathbb{E}_{o \sim \mathcal{D}} \left[ \min_{j=1,2} Q_{\phi_{i,j}}(o_i, \tilde{a}_i, a_{-i}) - \alpha_i \log \pi_{\theta_i}(\tilde{a}_i | o_i) \right] \quad (22-6)$$

که در آن  $\tilde{a}_i \sim \pi_{\theta_i}(\cdot | o_i)$  است و از ترفندهای پارامترسازی مجدد برای استخراج گرادیان استفاده می‌شود:

$$\tilde{a}_{i,\theta_i}(o_i, \xi_i) = \tanh(\mu_{\theta_i}(o_i) + \sigma_{\theta_i}(o_i) \odot \xi_i), \quad \xi_i \sim \mathcal{N}(0, I) \quad (23-6)$$

### شبکه‌های هدف

مشابه SAC، شبکه‌های هدف منتقد با میانگین‌گیری پولیاک به روزرسانی می‌شوند:

$$\phi_{i,j,\text{targ}} \leftarrow \rho \phi_{i,j,\text{targ}} + (1 - \rho) \phi_{i,j} \quad \text{برای } j = 1, 2 \quad (24-6)$$

### تنظیم ضریب آنتروپی

یکی از مزایای MA-SAC، توانایی تنظیم خودکار ضریب آنتروپی  $\alpha_i$  برای هر عامل است که می‌تواند با استفاده از یک تابع هزینه مجزا بهینه شود:

$$\min_{\alpha_i} \mathbb{E}_{o \sim \mathcal{D}, \tilde{a}_i \sim \pi_{\theta_i}} \left[ -\alpha_i \left( \log \pi_{\theta_i}(\tilde{a}_i | o_i) + H_{\text{target}} \right) \right] \quad (25-6)$$

که در آن  $H_{\text{target}}$  آنتروپی هدف است که به عنوان یک ابرپارامتر تعیین می‌شود.

#### ۴-۵-۶ اکتشاف در MA-SAC

اکتشاف در MA-SAC به صورت ذاتی از طریق سیاست‌های تصادفی و تنظیم آنتروپی انجام می‌شود. برخلاف MA-TD3 و MA-DDPG که به افزودن نویز به اعمال نیاز دارند، MA-SAC اعمال را مستقیماً از توزیع احتمال سیاست نمونه‌گیری می‌کند:

$$a_i \sim \pi_{\theta_i}(\cdot | O_i) \quad (26-6)$$

این رویکرد امکان اکتشاف ساختاریافته‌تر و کارآمدتر را فراهم می‌کند که در محیط‌های چندعاملی پیچیده مفید است.

#### ۵-۵-۶ شبکه MA-SAC برای بازی‌های چندعاملی مجموع‌صفر

در این بخش، شبکه الگوریتم MA-SAC پیاده‌سازی شده آورده شده است. در این پژوهش الگوریتم [۷](#) در محیط پایتون با استفاده از کتابخانه PyTorch [\[۶۰\]](#) پیاده‌سازی شده است.

## الگوریتم ۷ عامل عملگر نقاد نرم چندعاملی

- وروودی: پارامترهای اولیه سیاست عاملها  $(\theta_1, \theta_2)$ ، پارامترهای توابع  $Q(\phi_{1,1}, \phi_{1,2}, \phi_{2,1}, \phi_{2,2})$ ، ضرایب آنتروپی  $(\alpha_1, \alpha_2)$ ، بافر تکرار بازی خالی  $(\mathcal{D})$
- ۱: پارامترهای هدف را برابر با پارامترهای اصلی قرار دهید:
$$j \in \{1, 2\} \text{ و } \phi_{i,j,\text{targ}} \leftarrow \phi_{i,j}$$
  - ۲: تا وقتی همگرایی رخ دهد:
    - ۳: مشاهدات  $(o_1, o_2)$  را دریافت کنید
    - ۴: برای هر عامل  $i$ ، عمل  $a_i \sim \pi_{\theta_i}(\cdot | o_i)$  را انتخاب کنید
    - ۵: اعمال  $(a_1, a_2)$  را در محیط اجرا کنید
    - ۶: مشاهدات بعدی  $(o'_1, o'_2)$ ، پاداشها  $(r_1, r_2 = -r_1)$  و سیگنال پایان  $d$  را دریافت کنید
    - ۷: تجربه  $\mathcal{D}$  را در بافر ذخیره کنید
    - ۸: اگر  $d = 1$  است، وضعیت محیط را بازنشانی کنید
    - ۹: اگر زمان بهروزرسانی فرا رسیده است:
      - ۱۰: به ازای هر تعداد بهروزرسانی:
      - ۱۱: یک دسته تصادفی از تجربیات،  $\{(\mathbf{o}, \mathbf{a}, r_1, r_2, \mathbf{o}', d)\}$  از  $\mathcal{D}$  نمونهگیری کنید.
      - ۱۲: اهداف را محاسبه کنید:
$$y_1 = r_1 + \gamma(1-d) \left( \min_{j=1,2} Q_{\phi_{1,j,\text{targ}}}(o'_1, \tilde{a}'_1, \tilde{a}'_2) - \alpha_1 \log \pi_{\theta_1}(\tilde{a}'_1 | o'_1) \right)$$

$$y_2 = r_2 + \gamma(1-d) \left( \min_{j=1,2} Q_{\phi_{2,j,\text{targ}}}(o'_2, \tilde{a}'_2, \tilde{a}'_1) - \alpha_2 \log \pi_{\theta_2}(\tilde{a}'_2 | o'_2) \right)$$
    - ۱۳: توابع  $Q$  را با نزول گرادیان بهروزرسانی کنید:
$$\nabla_{\phi_{1,j}} \frac{1}{|B|} \sum_B (Q_{\phi_{1,j}}(o_1, a_1, a_2) - y_1)^2 \quad \text{برای } j = 1, 2$$

$$\nabla_{\phi_{2,j}} \frac{1}{|B|} \sum_B (Q_{\phi_{2,j}}(o_2, a_2, a_1) - y_2)^2 \quad \text{برای } j = 1, 2$$
    - ۱۴: سیاستها را با صعود گرادیان بهروزرسانی کنید:
$$\nabla_{\theta_1} \frac{1}{|B|} \sum_{\mathbf{o} \in B} \left[ \min_{j=1,2} Q_{\phi_{1,j}}(o_1, \tilde{a}_{1,\theta_1}(o_1, \xi_1), a_2) - \alpha_1 \log \pi_{\theta_1}(\tilde{a}_{1,\theta_1}(o_1, \xi_1) | o_1) \right]$$

$$\nabla_{\theta_2} \frac{1}{|B|} \sum_{\mathbf{o} \in B} \left[ \min_{j=1,2} Q_{\phi_{2,j}}(o_2, \tilde{a}_{2,\theta_2}(o_2, \xi_2), a_1) - \alpha_2 \log \pi_{\theta_2}(\tilde{a}_{2,\theta_2}(o_2, \xi_2) | o_2) \right]$$
    - ۱۵: ضرایب آنتروپی را با نزول گرادیان بهروزرسانی کنید (اختیاری):
$$\nabla_{\alpha_1} \frac{1}{|B|} \sum_{\mathbf{o} \in B} -\alpha_1 \left( \log \pi_{\theta_1}(\tilde{a}_{1,\theta_1}(o_1, \xi_1) | o_1) + H_{\text{target}} \right)$$

$$\nabla_{\alpha_2} \frac{1}{|B|} \sum_{\mathbf{o} \in B} -\alpha_2 \left( \log \pi_{\theta_2}(\tilde{a}_{2,\theta_2}(o_2, \xi_2) | o_2) + H_{\text{target}} \right)$$
    - ۱۶: شبکه‌های هدف را بهروزرسانی کنید:
$$\phi_{i,j,\text{targ}} \leftarrow \rho \phi_{i,j,\text{targ}} + (1 - \rho) \phi_{i,j} \quad \text{برای } i, j \in \{1, 2\}$$

## ۶-۵-۶ مزایای MA-SAC در بازی‌های مجموع صفر

مزایای زیر را نسبت به سایر الگوریتم‌های چندعاملی در بازی‌های چندعاملی مجموع صفر ارائه می‌دهد:

- **اکتشاف بهتر:** استفاده از سیاست‌های تصادفی و تنظیم آنتروپی، اکتشاف فضای حالت-عمل را بهبود می‌بخشد که برای یافتن راه حل‌های بهینه در بازی‌های دو عاملی ضروری است.
- **ثبات بیشتر:** ترکیب منتقدهای دوگانه با تنظیم آنتروپی، یادگیری را پایدارتر می‌کند و از همگرایی زودهنگام به سیاست‌های ضعیف جلوگیری می‌کند.
- **سازگاری با محیط‌های پیچیده:** توانایی تنظیم خودکار تعادل بین اکتشاف و بهره‌برداری، MA-SAC را برای محیط‌های چندعاملی پیچیده مناسب می‌سازد.
- **عملکرد بهتر در مسائل با چندین بهینه محلی:** سیاست‌های تصادفی می‌توانند از دام‌های بهینه محلی فرار کنند و به راه حل‌های بهتر برسند.

در مجموع، MA-SAC ترکیبی از ویژگی‌های مثبت SAC و رویکردهای چندعاملی را ارائه می‌دهد که آن را به گزینه‌ای قدرتمند برای یادگیری سیاست‌های پیچیده در بازی‌های چندعاملی مجموع صفر تبدیل می‌کند، بهویژه در محیط‌هایی که اکتشاف کارآمد و سیاست‌های تصادفی اهمیت دارند.

## ۶-۶ عامل بهینه‌سازی سیاست مجاور چندعاملی

عامل بهینه‌سازی سیاست مجاور دو عاملی<sup>۱۳</sup> توسعه‌ای از الگوریتم PPO برای محیط‌های چندعاملی است. در این بخش، به بررسی این الگوریتم در چارچوب بازی‌های چندعاملی مجموع صفر می‌پردازیم که در آن ترکیب ویژگی‌های PPO با رویکرد چندعاملی به پایداری و کارایی بیشتر در یادگیری منجر می‌شود.

## ۱-۶-۶ چالش‌های یادگیری تقویتی در محیط‌های چندعاملی و راه حل MA-PPO

در محیط‌های چندعاملی، عامل‌ها هم‌زمان سیاست‌های خود را تغییر می‌دهند که باعث غیرایستایی محیط از دید هر عامل می‌شود. این چالش با پیچیدگی‌های ذاتی الگوریتم‌های مبتنی بر گرادیان سیاست مانند PPO ترکیب می‌شود.

<sup>13</sup>Multi-Agent Proximal Policy Optimization (MA-PPO)

MA-PPO این چالش‌ها را با ترکیب رویکردهای زیر حل می‌کند:

- آموزش متمرکز، اجرای غیرمتمرکز: مشابه سایر الگوریتم‌های چندعاملی، از منتقدهایی استفاده می‌کند که به اطلاعات کامل دسترسی دارند، اما بازیگران تنها به مشاهدات محلی خود دسترسی دارند.
- بهروزرسانی کلیپ‌شده: استفاده از مکانیسم کلیپ شده PPO برای محدود کردن بهروزرسانی‌های سیاست، که به پایداری بیشتر در یادگیری چندعاملی کمک می‌کند.
- بافر تجربه مشترک: استفاده از یک بافر تجربه مشترک که تعاملات بین عامل‌ها را ثبت می‌کند.

## ۲-۶-۶ معماری MA-PPO در بازی‌های مجموع صفر

در یک بازی چندعاملی مجموع صفر، هر عامل دارای شبکه‌های زیر است:

- شبکه بازیگر:  $\pi_{\theta_i}(a_i|o_i)$  که توزیع احتمال اعمال را با توجه به مشاهدات محلی تعیین می‌کند.
- شبکه منتقد:  $V_{\phi_i}(o)$  که ارزش حالت متمرکز را (با دسترسی به مشاهدات همه عامل‌ها) تخمین می‌زند و برای محاسبه تابع مزیت استفاده می‌شود.

## ۳-۶-۶ آموزش MA-PPO

فرایند آموزش MA-PPO به شرح زیر است:

### جمع‌آوری تجربیات

در هر تکرار، عامل‌ها با استفاده از سیاست‌های فعلی خود در محیط تعامل می‌کنند و مجموعه‌ای از مسیرها را جمع‌آوری می‌کنند:

$$\mathcal{D}_k = \{(o_1^t, o_2^t, a_1^t, a_2^t, r_1^t, r_2^t, o_1^{t+1}, o_2^{t+1})\} \quad (27-6)$$

## محاسبه مزیت

برای هر عامل  $i \in \{1, 2\}$ ، تابع مزیت با استفاده از تابع ارزش فعلی محاسبه می‌شود. روش‌های مختلفی برای محاسبه مزیت وجود دارد؛ یک روش متداول استفاده از تخمین‌زننده مزیت تعمیم‌پافته (GAE) است:

$$\hat{A}_i^t = \sum_{l=0}^{\infty} (\gamma \lambda)^l \delta_{i,t+l} \quad (28-6)$$

که در آن  $\delta_{i,t} = r_i^t + \gamma V_{\phi_i}(\mathbf{o}^{t+1}) - V_{\phi_i}(\mathbf{o}^t)$  است.

## بهروزرسانی سیاست

سیاست هر عامل با بیشینه کردن تابع هدف PPO-Clip بهروزرسانی می‌شود:

$$\max_{\theta_i} \mathbb{E}_{(o_i, a_i) \sim \mathcal{D}_k} \left[ \min \left( \frac{\pi_{\theta_i}(a_i | o_i)}{\pi_{\theta_{i,k}}(a_i | o_i)} \hat{A}_i, \text{clip} \left( \frac{\pi_{\theta_i}(a_i | o_i)}{\pi_{\theta_{i,k}}(a_i | o_i)}, 1 - \epsilon, 1 + \epsilon \right) \hat{A}_i \right) \right] \quad (29-6)$$

یا با استفاده از همان فرمول‌بندی ساده‌تر:

$$\max_{\theta_i} \mathbb{E}_{(o_i, a_i) \sim \mathcal{D}_k} \left[ \min \left( \frac{\pi_{\theta_i}(a_i | o_i)}{\pi_{\theta_{i,k}}(a_i | o_i)} \hat{A}_i, g(\epsilon, \hat{A}_i) \right) \right] \quad (30-6)$$

که تابع  $g$  به صورت زیر تعریف شده‌است:

$$g(\epsilon, A) = \begin{cases} (1 + \epsilon)A & A \geq 0 \\ (1 - \epsilon)A & A < 0 \end{cases} \quad (31-6)$$

## بهروزرسانی منتقد

تابع ارزش هر عامل با کمینه کردن خطای میانگین مربعات بهروزرسانی می‌شود:

$$\min_{\phi_i} \mathbb{E}_{(o_i, \hat{R}_i) \sim \mathcal{D}_k} \left[ \left( V_{\phi_i}(o_i) - \hat{R}_i \right)^2 \right] \quad (32-6)$$

که در آن  $\hat{R}_i$  بازده تنزیل شده برای عامل  $i$  است.

## ۴-۶-۶ اکتشاف در MA-PPO

اکتشاف در MA-PPO به صورت ذاتی از طریق سیاست‌های تصادفی انجام می‌شود. برخلاف الگوریتم‌های مبتنی بر DDPG که به افروden نویز به اعمال نیاز دارند، MA-PPO از توزیع احتمال سیاست برای اکتشاف استفاده می‌کند:

$$a_i \sim \pi_{\theta_i}(\cdot | O_i) \quad (33-6)$$

این رویکرد اکتشاف سیاست‌محور، در ترکیب با مکانیسم کلیپ PPO که از به روزرسانی‌های بزرگ سیاست جلوگیری می‌کند، به ثبات بیشتر در یادگیری چندعاملی کمک می‌کند.

## ۵-۶-۶ شبکه MA-PPO برای بازی‌های چندعاملی مجموع صفر

در این بخش، شبکه الگوریتم MA-PPO پیاده‌سازی شده آورده شده است. در این پژوهش الگوریتم [۸](#) در محیط پایتون با استفاده از کتابخانه PyTorch [\[۶۰\]](#) پیاده‌سازی شده است.

---

## الگوریتم ۸ عامل بهینهسازی سیاست مجاور چندعاملی

---

وروودی: پارامترهای اولیه سیاست عاملها  $(\theta_1, \theta_2)$ ، پارامترهای تابع ارزش  $(\phi_1, \phi_2)$

: ۱: به ازای ... $k = 0, 1, 2, \dots$

: ۲: مجموعه‌ای از مسیرها به نام  $\mathcal{D}_k = \{(o_1^t, o_2^t, a_1^t, a_2^t, r_1^t, r_2^t, o_1^{t+1}, o_2^{t+1})\}$  با اجرای سیاست‌های  $\pi_{\theta_2}$  و  $\pi_{\theta_1}$  در محیط جمع‌آوری شود.

: ۳: برای هر عامل  $i$ ، پاداش‌های باقی‌مانده  $\hat{R}_i^t$  را محاسبه کنید.

: ۴: برای هر عامل  $i$ ، برآوردهای مزیت  $\hat{A}_i^t$  را با استفاده از تابع ارزش فعلی  $V_{\phi_i}$  محاسبه کنید.

: ۵: برای هر عامل  $i$ ، سیاست را با به حداقل رساندن تابع هدف PPO-Clip بهروزرسانی کنید:

$$\theta_{i,k+1} = \arg \max_{\theta_i} \frac{1}{|\mathcal{D}_k|} \sum_{(o_i, a_i) \in \mathcal{D}_k} \min \left( \frac{\pi_{\theta_i}(a_i | o_i)}{\pi_{\theta_{i,k}}(a_i | o_i)} \hat{A}_i, g(\epsilon, \hat{A}_i) \right)$$

: ۶: برای هر عامل  $i$ ، تابع ارزش را با رگرسیون بر روی میانگین مربعات خطأ بهروزرسانی کنید:

$$\phi_{i,k+1} = \arg \min_{\phi_i} \frac{1}{|\mathcal{D}_k|} \sum_{(o_i) \in \mathcal{D}_k} \left( V_{\phi_i}(o_i) - \hat{R}_i \right)^2$$

---

## ۶-۶-۶ مزایای MA-PPO در بازی‌های مجموع‌صفرا

MA-PPO مزایای زیر را نسبت به سایر الگوریتم‌های چندعاملی مجموع‌صفرا ارائه می‌دهد:

• پایداری یادگیری: مکانیسم کلیپ PPO از بهروزرسانی‌های بزرگ سیاست جلوگیری می‌کند که به پایداری بیشتر در محیط‌های غیرایستای چندعاملی منجر می‌شود.

• کارایی نمونه: به عنوان یک روش درون‌سیاست، MA-PPO معمولاً کارایی نمونه کمتری نسبت به روش‌های برونزیاست مانند MA-TD3 و MA-SAC دارد، اما پایداری بهتری در بهروزرسانی‌ها ارائه می‌کند.

• اکتشاف سیاست‌محور: اکتشاف ذاتی از طریق سیاست‌های تصادفی به جای افزودن نویز به اعمال، به اکتشاف کارآمدتر فضای حالت-عمل کمک می‌کند.

- **مقیاس‌پذیری:** MA-PPO به راحتی به سیستم‌های با تعداد بیشتری از عامل‌ها قابل گسترش است، اگرچه در این پژوهش بر بازی‌های دو‌عاملی تمرکز شده است.

در مجموع، MA-PPO ترکیبی از سادگی و کارایی PPO با رویکردهای چند‌عاملی را ارائه می‌دهد که آن را به گزینه‌ای قدرتمند برای یادگیری در بازی‌های چند‌عاملی مجموع صفر تبدیل می‌کند.

## فصل ۷

### ارزیابی و نتایج یادگیری

در این فصل، چارچوب ارزیابی و نتایج تجربی چهار الگوریتم شاخص یادگیری تقویتی برای کنترل فضاییما در میدان گرانشی سه‌جسمی برای روش‌های DDPG، PPO، SAC و TD3 ارائه می‌شود. تحلیل‌ها در دو قسمت انجام می‌گیرد: حالت تک‌عاملی استاندارد و حالت چندعاملی بازی مجموع صفر. تمرکز ارزیابی بر سه محور اصلی است: سنجش مقاومت در برابر آشفتگی‌های محیطی و سامانه‌ای (شرایط اولیه‌ی تصادفی، اغتشاش عملگر، عدم‌طابق مدل، مشاهده‌ی ناقص، نویز حسگر و تأخیر زمانی)، ارزیابی کیفیت مسیر و پروفایل فرمان پیشان، و گزارش شاخص‌های کمی شامل پاداش تجمعی، خطای مسیر، تلاش کنترلی و احتمال شکست.

به‌منظور هدایت خواننده و تضمین بازتولیدپذیری، ابتدا پروتکل و سناریوهای ارزیابی مقاومت همراه با جزئیات پیاده‌سازی و پارامترگذاری در بخش ۱-۷ معرفی می‌شود. سپس نتایج هر یک از الگوریتم‌ها به‌صورت نظاممند ارائه می‌گردد؛ بدین‌ترتیب که مسیر طی شده، فرمان‌های پیشان و توزیع پاداش در سناریوهای مختلف تحلیل می‌شود. نتایج DDPG در بخش ۲-۷، نتایج PPO در بخش ۵-۷، نتایج SAC در بخش ۴-۷ و نتایج TD3 در بخش ۳-۷ گزارش شده‌اند. در پایان، جمع‌بندی مقایسه‌ای برای نسخه‌های تک‌عاملی در بخش ۶-۷ و چندعاملی مجموع صفر در بخش ۷-۷ ارائه می‌شود تا تصویر روشنی از عملکرد نسبی روش‌ها فراهم گردد. در این مقایسه‌ها علاوه بر شاخص‌های عددی، مبادله‌های کارایی-پایداری و حساسیت نسبت به اغتشاش‌ها نیز مورد بحث قرار می‌گیرد.

#### ۱-۷ ارزیابی مقاومت الگوریتم‌ها

در این بخش، مقاومت الگوریتم‌های یادگیری در برابر شرایط مختلف اختلال مورد بررسی قرار گرفته است. این ارزیابی شامل شش سناریوی چالش‌برانگیز می‌شود: (۱) شرایط اولیه تصادفی، (۲) اغتشاش در عملگرها، (۳)

عدم تطابق مدل، (۴) مشاهده ناقص، (۵) نویز حسگر و (۶) تأخیر زمانی. هدف، بررسی توانایی الگوریتمها در حفظ کارایی خود در شرایط غیرایدهآل و نزدیک به واقعیت است.

## ۱-۱-۷ سناریوهای ارزیابی مقاومت

به منظور ایجاز، مشخصات هر سناریو به صورت فشرده فهرست شده است:

۱. شرایط اولیه تصادفی: به هر مؤلفه حالت اولیه نویز گوسی با انحراف معیار  $\sigma = 0.1$  افروده می‌شود:

$$x_0 \leftarrow x_0 + \mathcal{N}(0, 0.1^2)$$

۲. اغتشاش در عملگرها: نویز افزایشی روی ورودی‌ها و نویز کوچک روی سنسورها:

$$u_t \leftarrow u_t + \mathcal{N}(0, 0.05^2)$$

$$y_t \leftarrow y_t + \mathcal{N}(0, 0.02^2)$$

۳. عدم تطابق مدل: پارامترهای دینامیک در طول انتقال با نویز گوسی مختل می‌شوند:

$$\theta \leftarrow \theta + \mathcal{N}(0, 0.05^2)$$

۴. مشاهده ناقص: در هر گام، به صورت تصادفی ۵۰٪ از مؤلفه‌های مشاهده ماسک شده و مقدارشان صفر می‌شود:

$$m_t^{(i)} \sim \text{Bernoulli}(0.5), \quad y_t \leftarrow y_t \circ m_t$$

۵. نویز حسگر: نویز گوسی ضربی با  $\sigma = 0.05$  روی هر مؤلفه مشاهده اعمال می‌شود:

$$y_t \leftarrow y_t \circ (1 + \mathcal{N}(0, 0.05^2))$$

۶. تأخیر زمانی: اعمال عامل با تأخیر ۱۰ گام زمانی اعمال می‌شود و روی عمل تاخیردار نویز افزایشی افزوده می‌گردد:

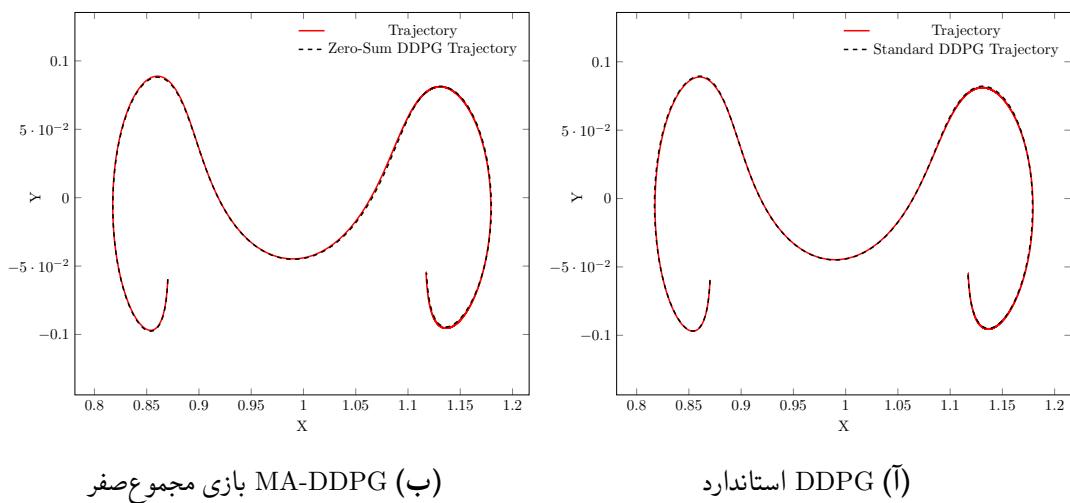
$$u_t^{\text{applied}} \leftarrow u_{t-10} + \mathcal{N}(0, 0.05^2)$$

## ۲-۷ الگوریتم DDPG

الگوریتم DDPG از جمله روش‌های یادگیری خارج از سیاست است که از دو شبکه عصبی برای بازیگر و منتقد استفاده می‌کند. در اینجا، عملکرد نسخه استاندارد و نسخه مبتنی بر بازی مجموع صفر این الگوریتم در کنترل فضایپیما مقایسه شده است.

### ۱-۲-۷ مسیر طی شده

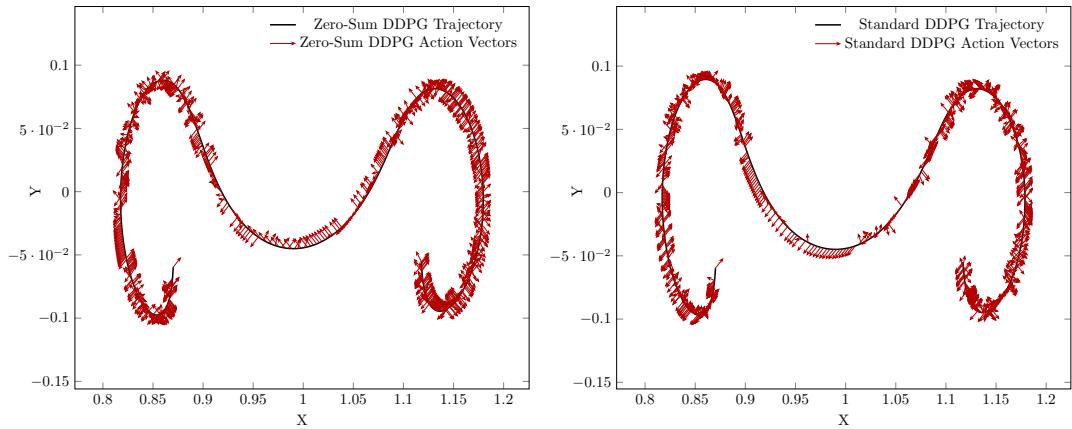
این بخش مسیر طی شده فضایپیما را برای نسخه استاندارد و نسخه بازی مجموع صفر DDPG نشان می‌دهد.



شکل ۱-۷: مسیر طی شده فضایپیما با DDPG استاندارد و نسخه بازی مجموع صفر .MA-DDPG

### ۲-۲-۷ مسیر و فرمان پیشران

این بخش مسیر و پروفایل فرمان پیشران در طول زمان را برای هر دو نسخه DDPG ارائه می‌کند.



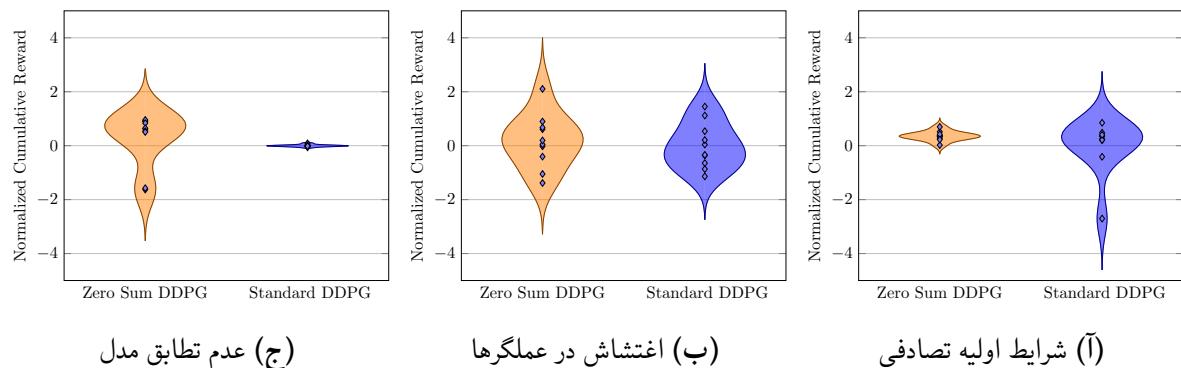
(ب) بازی مجموع صفر

(ج) استاندارد DDPG

شکل ۲-۷: مسیر و فرمان پیشران فضایپما در DDPG استاندارد و نسخه بازی مجموع صفر .MA-DDPG

### ۳-۲-۷ توزیع پاداش تجمعی

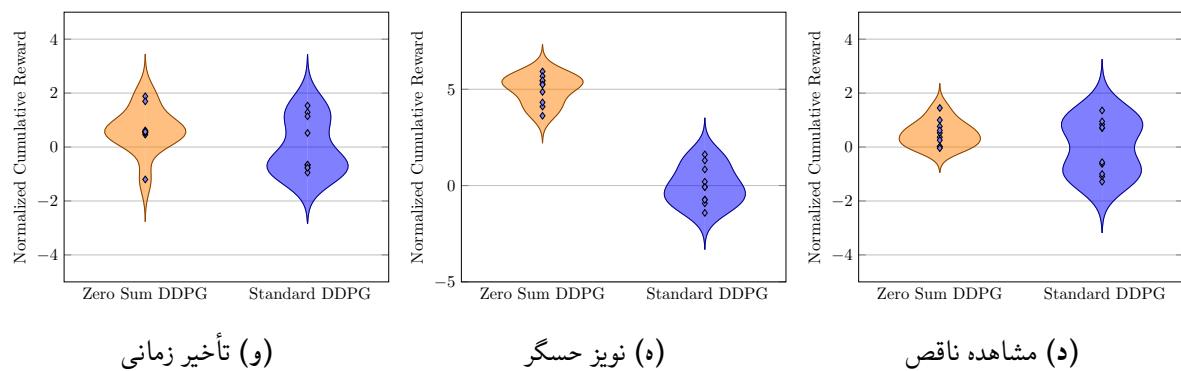
این بخش نمودارهای ویولن توزیع پاداش تجمعی را در سناریوهای مختلف برای DDPG و MA-DDPG نمایش می‌دهد.



(ج) عدم تطبیق مدل

(ب) اختشاش در عملگرها

(آ) شرایط اولیه تصادفی



(و) تأخیر زمانی

(ه) نویز حسگر

(د) مشاهده ناقص

شکل ۳-۷: مقایسه توزیع پاداش تجمعی در سناریوهای مختلف برای DDPG و MA-DDPG

## ۴-۲-۷ مقایسه عددی

این بخش شاخص‌های عددی را گزارش می‌کند؛ نتایج بر اساس 100 اجرای مستقل شبیه‌سازی برای هر سenario به دست آمده‌اند.

سناریو								
						پاداش تجمعی		
						مجموع خطای مسیر		
MA-DDPG	DDPG	MA-DDPG	DDPG	MA-DDPG	DDPG	MA-DDPG	DDPG	
0.40	1.00	5.30	5.52	0.36	0.40	-3.20	-4.17	شرایط اولیه تصادفی
0.28	0.90	5.30	5.60	6.80	7.56	-1.80	-1.93	اغتشاش در عملگرها
0.38	1.00	4.90	5.29	0.60	0.70	-2.20	-3.24	عدم تطابق مدل
0.48	0.60	5.10	5.51	0.60	0.68	-2.50	-3.28	مشاهده ناقص
0.00	0.00	5.35	5.54	0.08	0.10	-0.38	-1.07	نویز حسگر
0.45	0.70	5.30	5.61	1.40	1.74	-1.55	-3.20	تأخیر زمانی

جدول ۱-۷: مقایسه عملکرد DDPG و MA-DDPG در سناریوهای مختلف مقاومت

در جمع‌بندی بر اساس داده‌های جدول، MA-DDPG در پنج سناریو پاداش تجمعی بهتری از DDPG دارد و در اغتشاش عملگرها هر دو نسخه عملکردی نزدیک ارائه می‌دهند؛ مجموع خطای مسیر و تلاش کنترلی نیز در تمام سناریوها یا برابر شده‌اند یا با فاصله‌ای ناچیز گزارش می‌شوند. کاهش احتمال شکست در دو سناریوی بحرانی نشان می‌دهد که نسخه چندعامله، ریسک عملیاتی را بدون تحمیل هزینه کنترلی اضافی مدیریت کرده است.

### تحلیل نتایج

- در شرایط اولیه‌ی نامطمئن و عدم تطابق مدل، برتری پاداشی MA-DDPG همراه با کاهش احتمال شکست (از 1.00 به 0.40 و از 0.38 به 0.00) بیانگر توانایی این نسخه در مدیریت عدم قطعیت‌های ساختاری است؛ در حالی‌که مجموع تلاش کنترلی تفاوت محدودی دارد (5.52 در برابر 5.30 و 5.29 در 4.90).

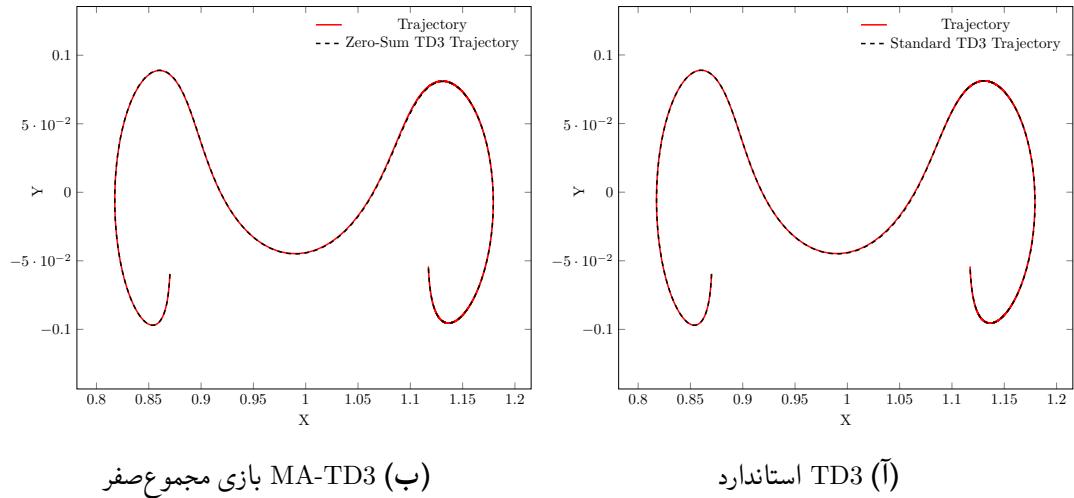
- در اغتشاش عملگرها، پاداش تجمعی تقریباً برابر (بهترتب 1.93 و 1.80) و مجموع خطای مسیر با فاصله محدود (7.56 در برابر 6.80) گزارش شده‌اند؛ مزیت نسخه چندعامله عمدتاً در کاهش احتمال شکست از 0.90 به 0.28 خلاصه می‌شود که ارزش عملیاتی قابل توجهی دارد.
- در سناریوی مشاهده ناقص، هم راستایی خطای مسیر (0.68 در برابر 0.60) و افزایش اینمی (کاهش احتمال شکست از 0.60 به 0.48) توازن خوبی میان کیفیت مسیر و تابآوری سنسوری ایجاد کرده است، هرچند تفاوت تلاش کنترلی (5.51 در برابر 5.10) نشان می‌دهد هنوز فضای بهینه‌سازی انرژی وجود دارد.
- دو سناریوی نویز حسگر و تأخیر زمانی نشان می‌دهند که نسخه چندعامله می‌تواند پاداش تجمعی بهتری ارائه دهد (1.07 در برابر 0.38 و 3.20 در برابر 1.55)؛ این بهبود بدون افزایش معنادار در تلاش کنترلی حاصل شده است (5.54 در برابر 5.35 و 5.61 در برابر 5.30) و هم‌زمان احتمال شکست را در مواجهه با تأخیر زمانی از 0.70 به 0.45 کاهش می‌دهد.

## ۳-۷ الگوریتم TD3

الگوریتم TD3 (یادگیری تفاضل زمانی سه‌گانه عمیق) نسخه بهبودیافته DDPG است که با استفاده از تکنیک‌های جدید مانند شبکه‌های دوگانه منتقد و تأخیر در بروزرسانی سیاست، مشکلات تخمین بیش از حد را کاهش می‌دهد.

## ۱-۳-۷ مسیر طی‌شده

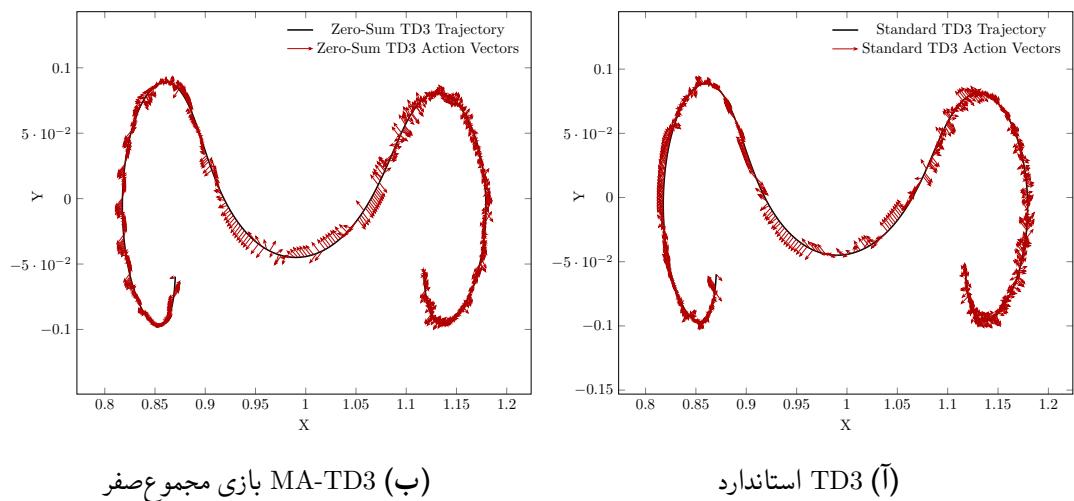
این بخش مسیر طی‌شده فضایپیما را برای نسخه استاندارد و نسخه بازی مجموع صفر TD3 نشان می‌دهد.



شکل ۴-۷: مسیر طی شده فضایپما با TD3 استاندارد و نسخه بازی مجموع صفر .MA-TD3

### ۲-۳-۷ مسیر و فرمان پیشان

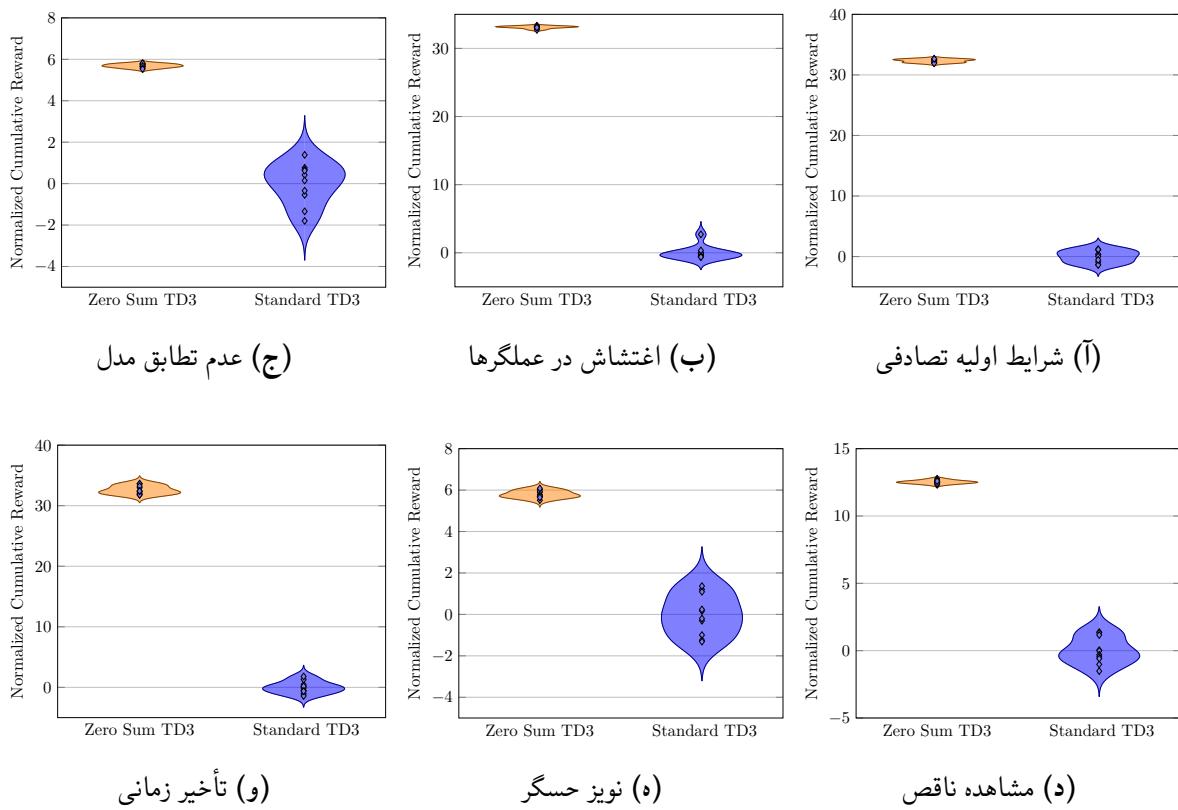
این بخش مسیر و پروفایل فرمان پیشان در طول زمان را برای هر دو نسخه TD3 ارائه می‌کند.



شکل ۴-۸: مسیر و فرمان پیشان فضایپما در TD3 استاندارد و نسخه بازی مجموع صفر .MA-TD3

### ۳-۳-۷ توزیع پاداش تجمعی

این بخش نمودارهای ویولن توزیع پاداش تجمعی را در سناریوهای مختلف برای TD3 و MA-TD3 نمایش می‌دهد.



شکل ۷-۶: مقایسه توزیع پاداش تجمعی در سناریوهای مختلف برای TD3 و MA-TD3.

### ۴-۳-۷ مقایسه عددی

این بخش شاخص‌های عددی را گزارش می‌کند؛ نتایج بر اساس 100 اجرای مستقل شبیه‌سازی برای هر سناریو به دست آمده‌اند.

پاداش تجمعی									سناریو	
مجموع خطای مسیر		تلاش کنترلی		احتمال شکست						
MA-TD3	TD3	MA-TD3	TD3	MA-TD3	TD3	MA-TD3	TD3			
0.28	1.00	4.40	5.05	0.12	0.39	-0.21	-2.95	شرایط اولیه تصادفی		
0.00	0.00	2.40	3.06	0.00	0.02	0.80	0.56	اغتشاش در عملگرها		
0.45	1.00	5.00	5.53	0.43	0.47	-2.80	-4.73	عدم تطابق مدل		
0.00	0.00	2.90	4.09	0.01	0.02	0.78	0.21	مشاهده ناقص		
0.00	0.00	5.20	5.46	0.10	0.11	-0.07	-0.08	نویز حسگر		
0.00	0.00	4.30	4.57	0.01	0.01	0.75	0.55	تأخیر زمانی		

جدول ۲-۷: مقایسه عملکرد TD3 و MA-TD3 در سناریوهای مختلف مقاومت

الگوریتم TD3 در هر دو حالت عملکرد قابل توجهی دارد؛ نسخه بازی مجموع صفر آن عمداً بهبودهای معناداری در کیفیت مسیر و مصرف سوخت نشان می‌دهد، هرچند در برخی سناریوها عملکرد آن با نسخه استاندارد برابر می‌کند و تفاوت‌ها ناچیز است. ثبات بیشتر این الگوریتم در مقایسه با DDPG در هر دو نسخه قابل مشاهده است.

## تحلیل نتایج

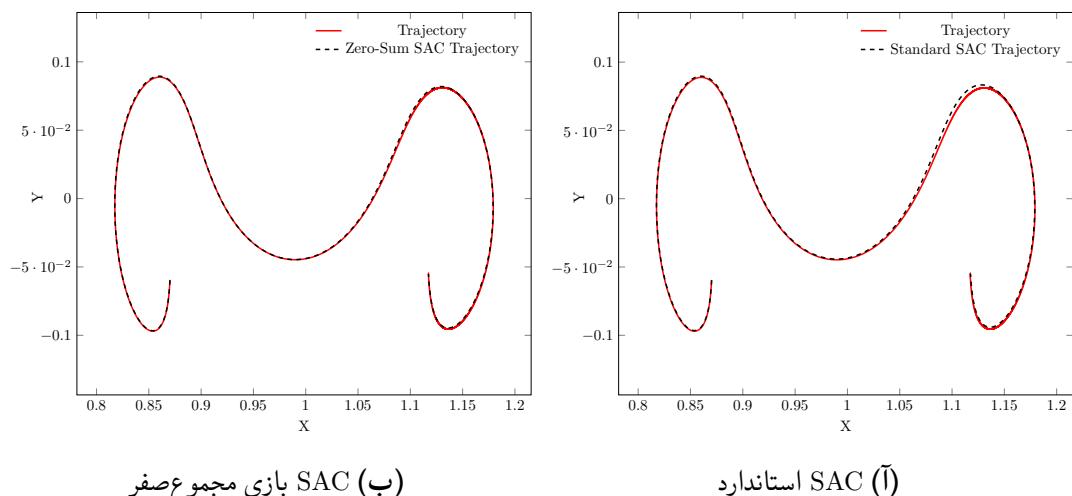
- در سناریوهای شرایط اولیه تصادفی و مشاهده ناقص، ترکیب بهبود پاداش و کاهش تلاش کنترلی برای MA-TD3 نشان می‌دهد که این سیاست در مواجهه با عدم قطعیت‌های ابتدایی و داده‌های ناقص، تصمیم‌گیری هموارتر و کم‌هزینه‌تری ارائه می‌دهد.
- نزدیکی شاخص‌ها در سناریوهای نویز حسگر و تأخیر زمانی پیام روشنی برای ارزیابی پایان‌نامه دارد؛ نسخه چندعامله حداقل کارایی نسخه پایه را حفظ می‌کند و همزمان بستر بهره‌گیری از راهبردهای دفاعی پیچیده‌تر را بدون تنبیه عملکردی جدی فراهم کرده است.
- کاهش احتمال شکست از 1 به 0.45 در سناریوی عدم تطابق مدل مؤید آن است که یادگیری چندعامله حساسیت سیاست به خطاهای مدل را کاهش داده است؛ نکته‌ای کلیدی برای مأموریت‌های فضایی که با مدل‌های تقریبی اجرا می‌شوند.

- تفاوت اندک در سناریوی اغتشاش عملگرها نشان می‌دهد که انعطاف‌پذیری TD3 اصلی در برابر اغتشاش‌های کنترلی بسیار بالاست و نسخه چندعامله باید از اطلاعات اضافی (مانند تعاملات تیمی) بهره بیشتری ببرد تا شکاف معنادار ایجاد کند.

## ۴-۷ الگوریتم SAC

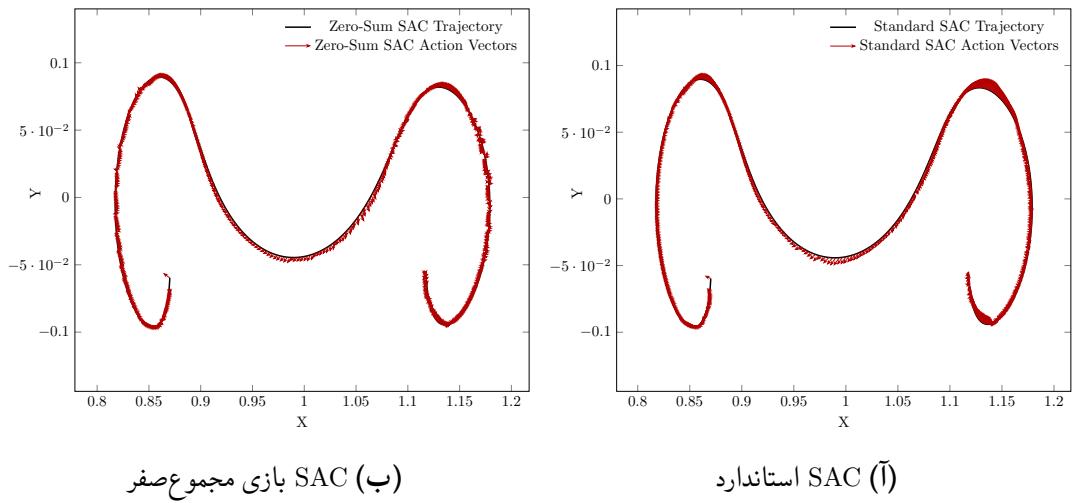
الگوریتم SAC از روش‌های نوین یادگیری تقویتی است که با استفاده از مفهوم آنتروپی، تعادل بهتری بین اکتشاف و بهره‌برداری ایجاد می‌کند. این الگوریتم در شرایط فضاهای پیوسته عملکرد قابل توجهی دارد.

### ۱-۴-۷ مسیر طی شده



شکل ۱-۷: مسیر طی شده فضایپیما با SAC استاندارد و نسخه بازی مجموع صفر MA-SAC

## ۲-۴-۷ مسیر و فرمان پیشران



شکل ۸-۷: مسیر و فرمان پیشran فضایپیما در SAC استاندارد و نسخه بازی مجموع صفر MA-SAC.

الگوریتم SAC در هر دو حالت عملکرد قابل قبولی ارائه می‌دهد. ویژگی خاص این الگوریتم در تنظیم خودکار پارامتر آنتروپی باعث می‌شود که بتواند تعادل مناسبی بین اکتشاف و بهبود پاداش ایجاد کند، اما نسخه بازی مجموع صفر آن در شرایط سخت‌تر مقاومت بیشتری نشان می‌دهد و در برخی شاخص‌ها عملکردی برابر با نسخه استاندارد دارد.

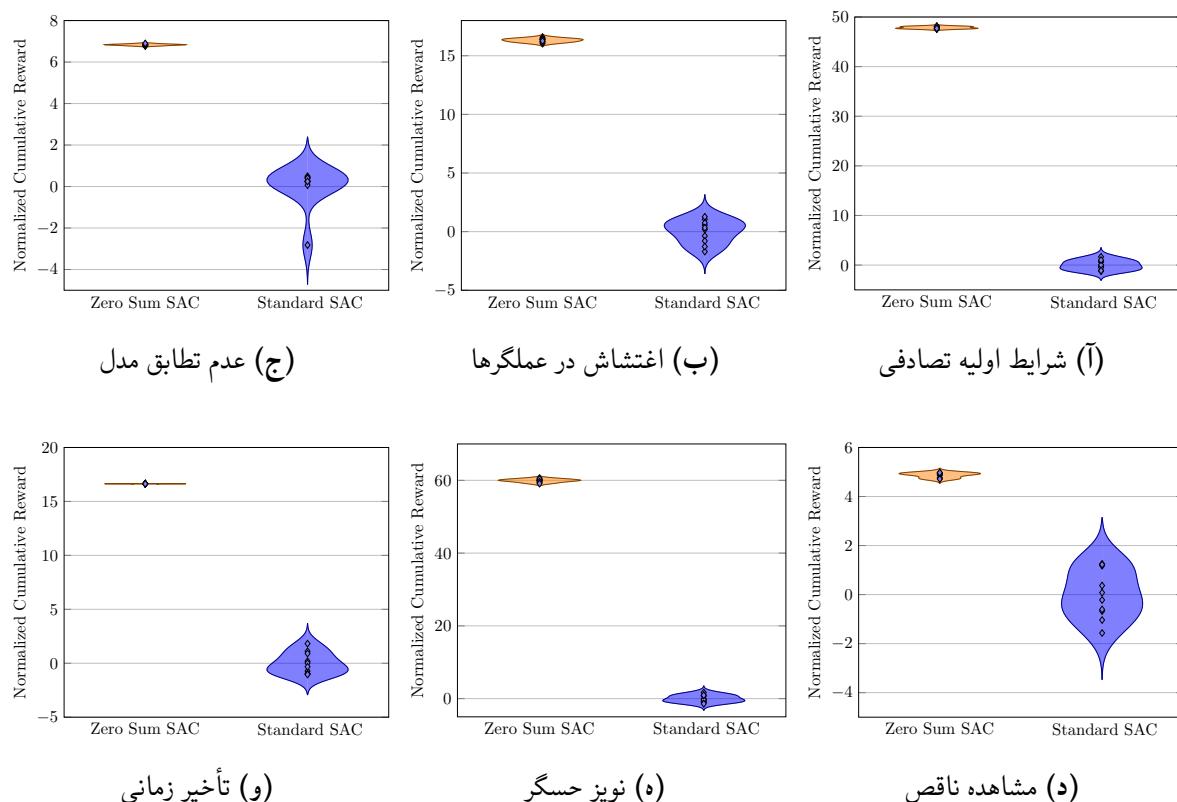
### تحلیل نتایج

- در سناریوهای شرایط اولیه تصادفی، اغتشاش در عملگرها و عدم تطابق مدل، نسخه چندعامله با کاهش احتمال شکست به بازه زیر ۰.۵ (به ترتیب ۰.۴۰، ۰.۴۵ و ۰.۴۲) و بهبود پاداش، نشان می‌دهد که کنترل آنتروپی مشترک میان عامل‌ها چگونه می‌تواند به مدیریت ریسک ساختاری کمک کند.
- عملکرد ممتاز در مشاهده ناقص و تأخیر زمانی (کاهش خطای مسیر از ۱.۹۵ به ۰.۰۶ و از ۱.۸۷ به ۰.۰۱) نشان می‌دهد که سازوکار چندعامله به خوبی می‌تواند اطلاعات ناقص یا دیررس را همپوشانی کند؛ این نکته در تحلیل پایان‌نامه برای مأموریت‌های واقعی با لینک‌های ارتباطی محدود حیاتی است.
- در نویز حسگر، حفظ برتری پاداش همراه با کاهش تلاش کنترلی (از ۲.۱۰ به ۱.۶۰) بیان می‌کند که تنظیم آنتروپی در نسخه چندعامله به سمت رفتارهای ملایم‌تر سوق پیدا کرده و انرژی کمتری مصرف شده است.
- کاهش هم‌زمان تلاش کنترلی و بهبود پاداش در سناریوهای دشوار (برای نمونه، مجموع تلاش کنترلی از ۳.۲۶ به ۲.۸۰ در اغتشاش عملگر و از ۲.۲۲ به ۱.۱۰ در تأخیر زمانی) نشان می‌دهد که نسخه چندعامله

تعادلی کاراتر میان اکتشاف و بهره‌برداری برقرار کرده و سرمایه‌گذاری روی معماری چندعامله را توجیه می‌کند.

### ۳-۴-۷ توزیع پاداش تجمعی

این بخش نمودارهای ویولن توزیع پاداش تجمعی را در سناریوهای مختلف برای SAC و MA-SAC نمایش می‌دهد.



شکل ۷-۹: مقایسه توزیع پاداش تجمعی در سناریوهای مختلف برای SAC و MA-SAC.

### ۴-۴-۷ مقایسه عددی

این بخش شاخص‌های عددی را گزارش می‌کند؛ نتایج بر اساس 100 اجرای مستقل شبیه‌سازی برای هر سناریو به دست آمده‌اند.

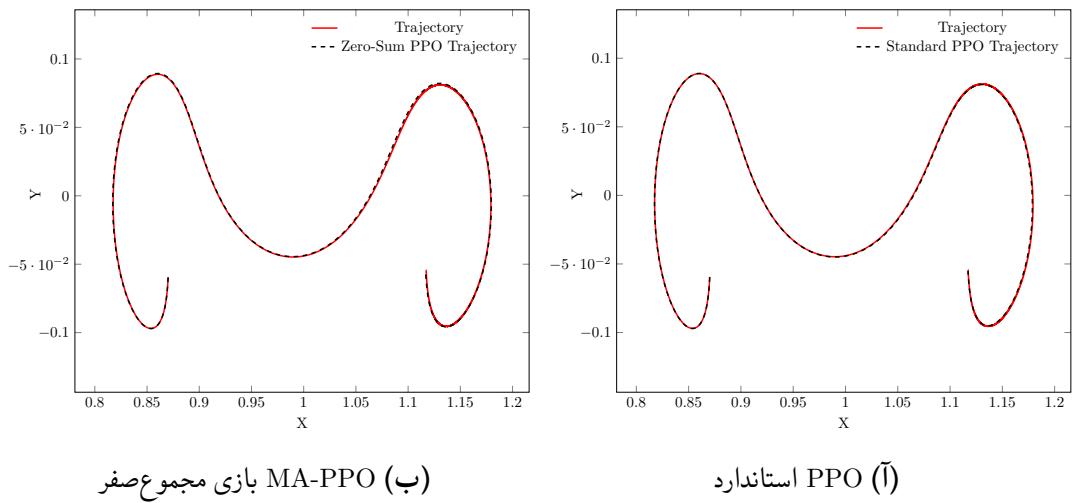
پاداش تجمعی								مجموع خطای مسیر	مجموع تلاش کنترلی	احتمال شکست	سناریو
MA-SAC	SAC	MA-SAC	SAC	MA-SAC	SAC	MA-SAC	SAC				
0.40	1.00	1.20	2.15	0.22	0.29	-2.50	-4.69	شرایط اولیه تصادفی			
0.45	1.00	2.80	3.26	6.90	8.02	-1.60	-1.95	اغتشاش در عملگرها			
0.42	1.00	1.00	1.99	0.22	0.38	-3.50	-4.89	عدم تطابق مدل			
0.00	1.00	1.70	2.32	0.06	1.95	-0.35	-3.63	مشاهده ناقص			
0.00	0.00	1.60	2.10	0.10	0.12	0.14	-0.89	نویز حسگر			
0.00	1.00	1.10	2.22	0.01	1.87	-0.04	-4.14	تأخیر زمانی			

جدول ۳-۷: مقایسه عملکرد MA-SAC و SAC در سناریوهای مختلف مقاومت

## ۵-۷ الگوریتم PPO

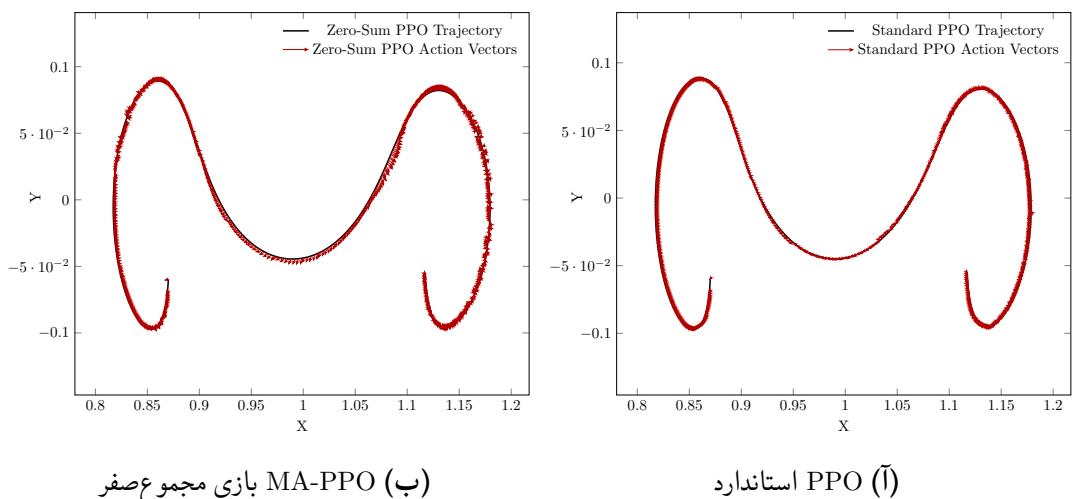
الگوریتم PPO از روش‌های نوین سیاست گرادیان است که با محدودسازی میزان تغییرات در هر بروزرسانی، پایداری بیشتری در فرآیند یادگیری ایجاد می‌کند. در ادامه، عملکرد این الگوریتم در دو حالت مورد بررسی قرار گرفته است.

## ۱-۵-۷ مسیر طی شده



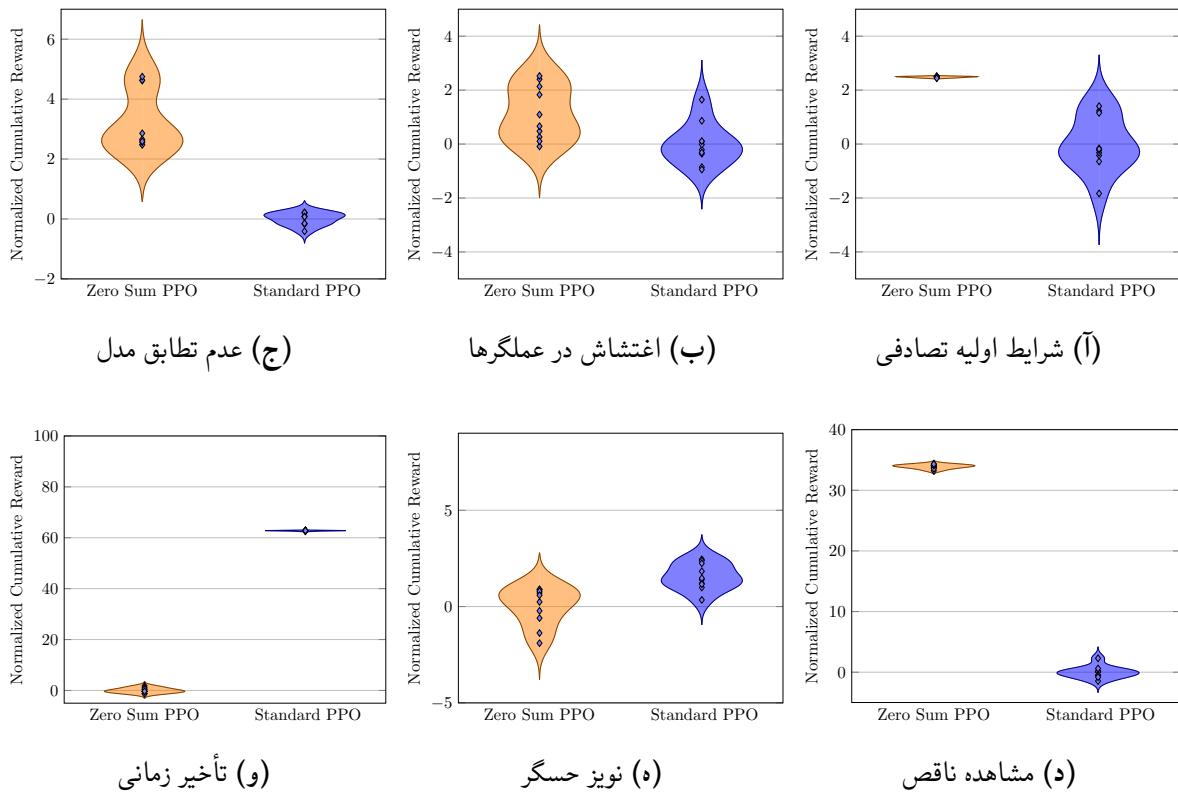
شکل ۱۰-۷: مسیر طی شده فضای پیما با PPO استاندارد و نسخه بازی مجموع صفر .MA-PPO

## ۲-۵-۷ مسیر و فرمان پیشران



شکل ۱۱-۷: مسیر و فرمان پیشران فضای پیما در PPO استاندارد و نسخه بازی مجموع صفر .MA-PPO

### ۳-۵-۷ توزیع پاداش تجمعی



شکل ۱۲-۷: مقایسه توزیع پاداش تجمعی برای PPO و MA-PPO در سناریوهای مختلف.

### ۴-۵-۷ مقایسه عددی

این بخش شاخص‌های عددی را گزارش می‌کند؛ نتایج بر اساس 100 اجرای مستقل شبیه‌سازی برای هر سناریو به دست آمده‌اند.

پاداش تجمعی								سناریو
MA-PPO	PPO	MA-PPO	PPO	MA-PPO	PPO	MA-PPO	PPO	
0.00	0.70	1.40	1.55	0.12	0.22	0.50	-1.85	شرایط اولیه تصادفی
0.40	1.00	2.40	2.59	6.50	8.33	-1.55	-1.97	اغتشاش در عملگرها
0.00	0.00	0.85	0.90	0.06	0.07	0.50	0.46	عدم تطابق مدل
0.42	1.00	0.95	1.06	1.70	2.34	-1.45	-3.60	مشاهده ناقص
0.00	0.00	1.10	1.22	0.11	0.13	0.58	0.52	نویز حسگر
0.00	0.00	2.20	2.43	0.03	0.03	0.64	0.58	تأخیر زمانی

جدول ۷-۴: مقایسه عملکرد PPO و MA-PPO در سناریوهای مختلف مقاومت

نتایج نشان می‌دهد که الگوریتم PPO در حالت بازی مجموع صفر عملکرد قابل توجهی دارد و در بسیاری از سناریوها بهبودهای ملموس ارائه می‌دهد، هرچند در برخی شاخص‌ها هر دو نسخه عملکردی برابر از خود نشان می‌دهند.

## تحلیل نتایج

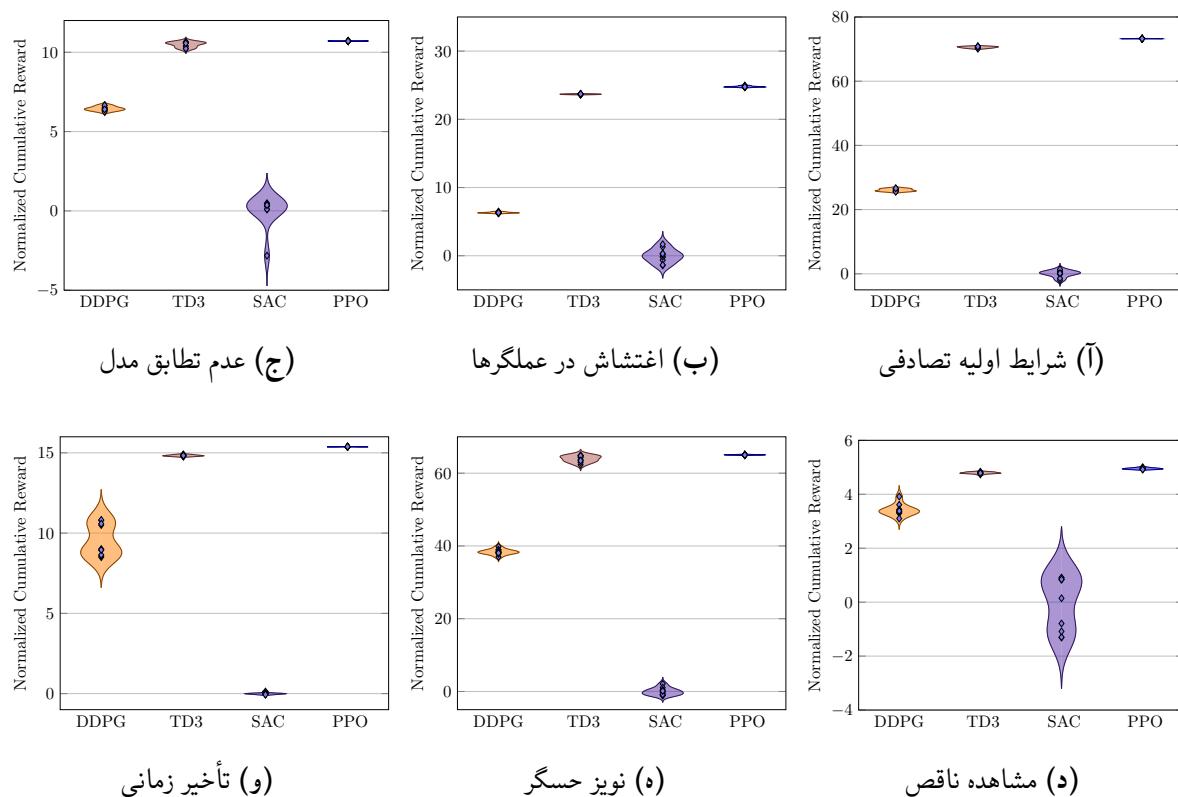
- در سناریوهای شرایط اولیه تصادفی و مشاهده ناقص، افزایش چشمگیر پاداش همراه با کاهش یا حفظ خطای مسیر نشان می‌دهد که سیاست چندعامله توانسته است محدودیت کلیپینگ PPO را در جهت تابآوری بیشتر هدایت کند.
- نزدیکی شاخص‌ها در عدم تطابق مدل، نویز حسگر و تأخیر زمانی حاکی از آن است که مزیت اصلی نسخه چندعامله بیشتر به پایداری آموزشی بازمی‌گردد تا جهش‌های بزرگ در عملکرد نهایی؛ این نتیجه در چارچوب پایان‌نامه بدان معناست که پیچیدگی اضافی الزاماً خروجی عملیاتی را به‌طور محسوسی بهتر نمی‌کند.
- کاهش احتمال شکست از 1 به 0.40 در اغتشاش عملگرها و از 1 به 0.42 در مشاهده ناقص، ارزش افزوده‌ی نسخه چندعامله را در شرایط بحرانی نشان می‌دهد؛ این کاهش هرچند محدود، در مأموریت‌های حساس می‌تواند تقاضه میان موفقیت و شکست باشد و برای این ارزیابی اهمیت دارد.

- افت ملایم تلاش کنترلی در اغلب سناریوها (برای نمونه از 2.40 به 2.59 در اغتشاش عملگرها و از 2.43 به 2.20 در تأخیر زمانی) نشان می‌کند که بهبودهای یادشده بدون تحمیل هزینه سوخت قابل ملاحظه‌ای کسب شده‌اند، هرچند نبود برتری قاطع در محیط‌های نویزی نیاز به مکانیزم‌های اشتراک اطلاعات غنی‌تر میان عامل‌ها را برجسته می‌کند.

## ۶-۷ نتایج نسخه استاندارد

در این بخش، نتایج نسخه‌های تک‌عاملی الگوریتم‌ها در سناریوهای مقاومت مختلف ارائه و تحلیل می‌شود.

### ۱-۶-۷ توزیع پاداش تجمعی



شکل ۱۳-۷: مقایسه توزیع پاداش تجمعی برای نسخه‌های تک‌عاملی در سناریوهای مختلف.

## ۲-۶-۷ مقایسه عددی

سناریو					سناریو				
TD3	SAC	PPO	DDPG		TD3	SAC	PPO	DDPG	
0.72	8.06	2.56	3.30	شرایط اولیه تصادفی	0.56	-0.76	0.61	-0.27	شرایط اولیه تصادفی
0.77	7.91	2.58	3.74	اغتشاش در عملگرها	0.55	-0.72	0.61	-0.38	اغتشاش در عملگرها
1.09	17.12	3.06	10.87	عدم تطابق مدل	0.51	-2.98	0.58	-0.84	عدم تطابق مدل
1.77	15.47	3.34	8.18	مشاهده ناقص	0.23	-3.65	0.36	-0.88	مشاهده ناقص
1.02	16.81	3.08	11.04	نویز حسگر	0.52	-2.90	0.58	-0.85	نویز حسگر
0.81	15.70	2.27	8.95	تأخیر زمانی	0.48	-2.98	0.61	-0.76	تأخیر زمانی
مجموع خطای مسیر					پاداش تجمعی				
TD3	SAC	PPO	DDPG	سناریو	TD3	SAC	PPO	DDPG	سناریو
0.00	0.00	0.00	0.00	شرایط اولیه تصادفی	3.31	1.76	0.77	5.11	شرایط اولیه تصادفی
0.00	0.00	0.00	0.00	اغتشاش در عملگرها	3.07	1.71	0.77	4.89	اغتشاش در عملگرها
0.00	1.00	0.00	0.00	عدم تطابق مدل	4.32	2.37	0.86	5.48	عدم تطابق مدل
0.00	1.00	0.00	0.00	مشاهده ناقص	4.10	2.33	1.03	5.37	مشاهده ناقص
0.00	1.00	0.00	0.00	نویز حسگر	4.30	2.37	0.86	5.48	نویز حسگر
0.00	1.00	0.00	0.00	تأخیر زمانی	5.12	2.11	0.76	5.51	تأخیر زمانی
احتمال شکست					مجموع تلاش کنترلی				

جدول ۷-۵: مقایسه الگوریتم‌های چندعاملی در سناریوهای مختلف مقاومت

بر اساس داده‌ها، TD3 به طور پایدار بالاترین پاداش و کمترین خطای مسیر را ثبت می‌کند، در حالی‌که PPO کمترین تلاش کنترلی را دارد. SAC در برخی سناریوهای دشوار (عدم تطابق مدل، مشاهده ناقص، نویز حسگر، تأخیر زمانی) نرخ شکست بالاتری نشان می‌دهد و DDPG عموماً از نظر پاداش و خطای ضعیفتر از PPO و TD3 است.

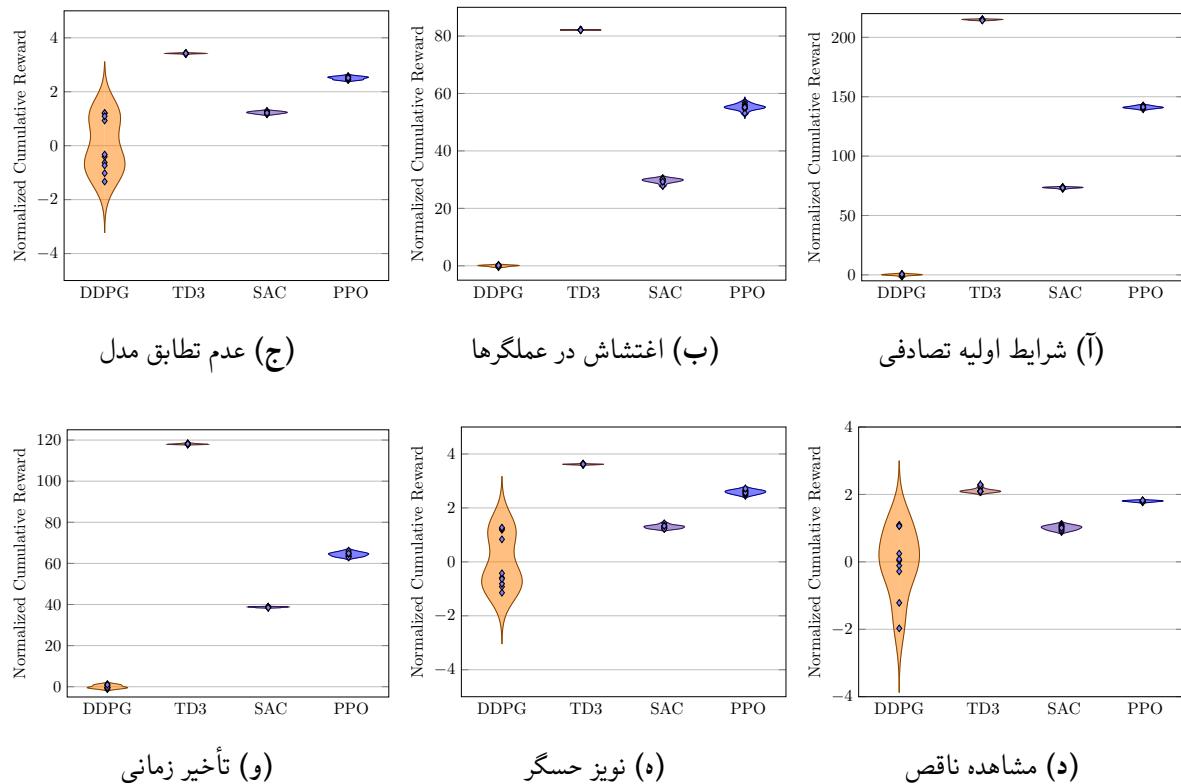
## جمع‌بندی تحلیلی پایان‌نامه

- TD3 در نسخه تک‌عاملی نیز پاداش‌های مثبت میان 0.23 تا 0.77 و خطای مسیر محدود به 0.72 تا 1.77 دارد که آن را پایدارترین گزینه از نظر پیگیری مسیر می‌کند.
- PPO با مجموع تلاش کنترلی در بازه 0.76 تا 1.03 بهره‌ورترين مصرف سوخت را ارائه می‌دهد، در حالی که پاداش‌های 0.36 تا 0.61 آن نشان می‌دهد تعادل کارآمدی میان کارایی و هزینه برقرار شده است.
- SAC اگرچه در برخی سناریوها پاداش‌های نزدیک به صفر (مثلاً -0.72) دارد، نرخ شکست 1.00 در چهار سناریو و خطای مسیر تا 17.12 اهمیت بهبود روش‌های سهیم‌سازی اطلاعات را برجسته می‌کند.
- DDPG با خطای مسیرهای بین 3.30 تا 10.87 و پاداش‌های منفی تا -0.88 – نسبت به PPO و TD3 عملکرد ضعیفتری دارد و در جمع‌بندی پایان‌نامه تنها به عنوان پایه مقایسه مطرح می‌شود.

## ۷-۷ نتایج نسخه چند‌عاملی

در این بخش، عملکرد الگوریتم‌ها در حالت چند‌عاملی بازی مجموع صفر ارائه و تحلیل می‌شود.

## ۱-۷-۷ توزیع پاداش تجمعی



شکل ۱۴-۷: مقایسه توزیع پاداش تجمعی برای الگوریتم‌ها در حالت چندعاملی در سناریوهای مختلف.

## ۲-۷-۷ مقایسه عددی

سناریو					سناریو				
TD3	SAC	PPO	DDPG		TD3	SAC	PPO	DDPG	
1.22	4.02	4.30	4.42	شرایط اولیه تصادفی	0.74	-0.02	0.34	-0.41	شرایط اولیه تصادفی
1.26	4.01	4.38	4.39	اغتشاش در عملگرها	0.73	-0.02	0.35	-0.44	اغتشاش در عملگرها
1.25	4.78	3.57	8.85	عدم تطابق مدل	0.75	-0.13	0.38	-0.63	عدم تطابق مدل
1.09	5.17	2.44	9.65	مشاهده ناقص	0.71	-0.44	0.40	-1.52	مشاهده ناقص
1.25	4.66	3.58	9.12	نویز حسگر	0.75	-0.12	0.37	-0.60	نویز حسگر
1.21	4.12	4.53	6.73	تأخیر زمانی	0.67	-0.05	0.17	-1.19	تأخیر زمانی
مجموع خطای مسیر					پاداش تجمعی				
TD3	SAC	PPO	DDPG	سناریو	TD3	SAC	PPO	DDPG	سناریو
0.00	0.00	0.00	0.00	شرایط اولیه تصادفی	2.76	1.34	1.15	5.40	شرایط اولیه تصادفی
0.00	0.00	0.00	0.00	اغتشاش در عملگرها	2.66	1.23	1.11	5.08	اغتشاش در عملگرها
0.00	0.20	0.00	0.00	عدم تطابق مدل	3.38	2.09	1.51	5.55	عدم تطابق مدل
0.00	0.20	0.00	0.00	مشاهده ناقص	3.20	2.00	1.50	5.46	مشاهده ناقص
0.00	0.20	0.00	0.00	نویز حسگر	3.38	2.08	1.52	5.54	نویز حسگر
0.00	0.20	0.00	0.00	تأخیر زمانی	4.57	1.25	1.25	5.48	تأخیر زمانی
احتمال شکست					مجموع تلاش کنترلی				

جدول ۶-۷: مقایسه الگوریتم‌های چندعاملی در سناریوهای مختلف مقاومت

در حالت چندعاملی، TD3 به طور پایدار پاداش بالاتر و خطای مسیر کمتر ثبت می‌کند، در حالی‌که PPO کمترین تلاش کنترلی را نشان می‌دهد. عملکرد SAC و DDPG در برخی سناریوهای دشوار ضعیفتر است، هرچند نرخ‌های شکست عمده‌تاً پایین باقی می‌ماند.

## جمع‌بندی تحلیلی پایان‌نامه

- TD3 در تمام سناریوها پاداش مثبت بین 0.67 تا 0.75 و مجموع خطای مسیر زیر 1.26 ثبت کرده است که نشان می‌دهد سیاست فعال باخت-برد دوگانه مسیرهایی پایدار و دقیق فراهم کرده است.
- PPO با نگه داشتن مجموع تلاش کنترلی در بازه 1.11 تا 1.52 کم‌صرف‌ترین گوینه باقی می‌ماند، هرچند پادash‌های آن در حد 0.17 تا 0.40 از TD3 عقب‌تر است.
- SAC اگرچه در برخی سناریوها پاداش نزدیک به صفر (برای نمونه 0.02 – در شرایط اولیه تصادفی) دارد، احتمال شکست 0.20 در چهار سناریو نشان می‌دهد که حساسیت آن به اغتشاشات در نسخه چندعامله همچنان قابل توجه است.
- DDPG با پادash‌های منفی بین -1.52 تا -0.41 و خطای مسیر بالای 4.39 نسبت به سایر روش‌ها عقب می‌ماند؛ بنابراین در ارزیابی نهایی پایان‌نامه تمکز روی نسخه‌های چندعامله پیشرفته‌تر توجیه‌پذیر است.

## فصل ۸

### نتیجه‌گیری و پیشنهادها

در این پایان‌نامه، مسأله‌ی هدایت مقاوم فضایی‌ماهی کم‌پیشران در دینامیک چندجسمی مدل CRTBP زمین-ماه به صورت یک بازی دیفرانسیلی مجموع صفر میان عامل هدایت و عامل مزاحم صورت‌بندی شد و با الگوی آموزش متمرکز-اجرای توزیع شده دنبال گردید. چهار الگوریتم پیوسته‌ی SAC، TD3، DDPG و PPO به نسخه‌های چندعاملی مجموع صفر تعمیم داده شدند، اجزای بازیگر-منتقد و سازوکارهای پایداری آموزش تشریح گردید. ارزیابی گسترده زیر عدم قطعیت‌های واقع‌گرایانه شرایط اولیه‌ی تصادفی، اغتشاش عملگر، نویز حسگر، تأخیر زمانی و عدم تطابق مدل نشان داد نسخه‌های مجموع صفر به صورت پایدار از همتایان تک‌عاملی پیشی می‌گیرند؛ به ویژه MA-TD3 بهترین سازش میان دقت مسیر، مصرف سوخت و پایداری را فراهم کرد.

#### ۱-۸ جمع‌بندی دستاوردها

- ارائه‌ی صورت‌بندی بازی دیفرانسیلی مجموع صفر برای هدایت کم‌پیشران در CRTBP با آموزش متمرکز و اجرای توزیع شده.
- تعمیم چهار الگوریتم پرکاربرد RL به نسخه‌های چندعاملی مجموع صفر و تبیین دقیق معماهی بازیگر-منتقد و پایدارسازی آموزش.
- طراحی حریف‌یادگیر برای تنوع‌بخشی نظام مند به عدم قطعیت‌ها و ارتقای تابآوری سیاست در سناریوهای دشوار.
- پروتکل ارزیابی چندمعیاره با شاخص‌های دقت مسیر، پاداش و پایداری، و نشان‌دادن برتری منسجم نسخه‌های مجموع صفر.

## ۲-۸ محدودیت سناریوهای آزمون

در این پژوهش، ارزیابی چارچوب پیشنهادی یادگیری تقویتی چندعاملی در چارچوب مسأله‌ی محدود سه‌جسمی، بر روی سناریوی انتقال کمتراست در حوالی نقطه‌ی لاگرانژ  $L_1$  و نزدیک به مدار هدف در مجاورت  $L_2$  متمرکز شده است. انتخاب این سناریو به دو دلیل اصلی انجام شده است:

- این سناریو یکی از سناریوهای مرجع و رایج در ادبیات هدایت کمتراست در CRTBP محسوب می‌شود و امکان مقایسه‌ی مستقیم با روش‌های موجود را فراهم می‌کند؛
- تمرکز پژوهش حاضر بر تبیین و ارزیابی چارچوب مقاوم مبتنی بر بازی دیفرانسیلی و یادگیری تقویتی چندعاملی بوده است و نه طراحی جامع کلیه‌ی پروفایل‌های مأموریتی ممکن در محیط سه‌جسمی.

گسترش آزمون‌ها به نقاط شروع و پایان دیگر (برای مثال حوالی سایر نقاط لاگرانژ یا مدارهای هدف متفاوت)، علاوه بر تغییر شرایط اولیه، مستلزم حل یک مسأله‌ی مجازی طراحی مسیر و تعریف مجدد قیود مأموریتی و تابع پاداش است. از آن‌جا که این موارد خود بتهنایی می‌توانند موضوع یک پژوهش مستقل باشند، در این رساله به عنوان کارآینده در نظر گرفته شده‌اند و تمرکز بر روی یک سناریوی شاخص، امکان تحلیل عمیق‌تر رفتار الگوریتم‌های پیشنهادی را فراهم نموده است.

با وجود این محدودیت، در فصل ۷ نشان داده شد که چارچوب پیشنهادی در برابر طیفی از عدم‌قطعیت‌ها شامل تأخیر حسگر، نویز عملگر و عدم‌طابق مدل، رفتار پایدار و مقاومی از خود بروز می‌دهد. این نتایج نشان می‌دهد که روش ارائه‌شده صرفاً به یک نقطه‌ی شروع و پایان خاص وابسته نیست و پایداری آن بیش از آن که تابع دقیق شرایط اولیه باشد، به ساختار بازی دیفرانسیلی و طراحی تابع پاداش مقاوم مرتبط است.

## ۳-۸ پیشنهادهایی برای کارهای آینده

- تعمیم چارچوب به مسئله جهت انتقال به سایر نقاط لاگرانژ.
- تعمیم چارچوب به مسئله  $N$ -body و درنظرگرفتن اغتشاشات غیرگرانشی؛ استفاده از یادگیری مرحله‌ای مناسب با پیچیدگی دینامیکی (curriculum).
- بررسی Risk-Sensitive RL، اعمال قیود ایمنی به صورت chance constraints و به کارگیری Con-trol Barrier Functions برای فراهم‌کردن تضمین.

- توسعه راهبردهای ترکیبی یادگیری تقویتی و کنترل مبتنی بر مدل (مانند iLQR/MPC) برای بهبود ایمنی و تفسیرپذیری.
- آموزش خصم‌انه مبتنی بر توزیع (جمعیت مزاحم‌ها) و طراحی curriculum/adversary shaping برای پوشش بهتر نواحی عدم قطعیت.
- استقرار روی سامانه‌های تعییه‌شده کم‌صرف و مقایسه‌ی TVM، ONNX Runtime و TensorRT در معما‌ری‌های گوناگون؛ بهینه‌سازی latency/throughput و سنجه‌ی energy-delay.
- انجام تحلیل حساسیت نسبت به تابع پاداش، معماری، نویز حسگر و تأخیر؛ مستندسازی دقیق برای ارتقای بازتولیدپذیری.

در مجموع، نتایج این پژوهش نشان داد که رویکرد بازی‌محور چندعاملی در یادگیری تقویتی می‌تواند هدایت تطبیقی و مقاوم را بدون اتكای شدید به مدل‌های دقیق فراهم کند و مسیر روشنی برای گذار به کاربردهای عملی و سناریوهای پیچیده‌تر می‌گشاید.

# Bibliography

- [1] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, second edition, 2018.
- [2] 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.
- [3] 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.
- [4] 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.
- [5] 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.
- [6] 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.
- [7] 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.
- [8] A. Rubinsztein, R. Sood, and F. E. Laipert. Neural network optimal control in astrodynamics: Application to the missed thrust problem. *Acta Astronautica*, 176:192–203, 2020.
- [9] 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.

- [10] 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.
- [11] 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.
- [12] 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.
- [13] 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.
- [14] 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.
- [15] S. D. Smet and D. J. Scheeres. Identifying heteroclinic connections using artificial neural networks. *Acta Astronautica*, 161:192–199, 2019.
- [16] N. L. O. Parrish. *Low Thrust Trajectory Optimization in Cislunar and Translunar Space*. PhD thesis, University of Colorado Boulder, 2018.
- [17] 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.
- [18] 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.

- [19] R. Furfaro, A. Scorsoglio, R. Linares, and M. Massari. Adaptive generalized zemzev feedback guidance for planetary landing via a deep reinforcement learning approach. *Acta Astronautica*, 171:156–171, 2020.
- [20] 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.
- [21] B. Gaudet, R. Furfaro, and R. Linares. Reinforcement learning for angle-only intercept guidance of maneuvering targets. *Aerospace Science and Technology*, 99, 2020.
- [22] 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.
- [23] 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.
- [24] 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.
- [25] D. Miller and R. Linares. Low-thrust optimal control via reinforcement learning. In *29th AAS/AIAA Space Flight Mechanics Meeting*, Ka’anapali, Hawaii, 2019.
- [26] 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.
- [27] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, Feb. 2015.
- [28] J. Schulman, S. Levine, P. Moritz, M. I. Jordan, and P. Abbeel. Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, pages 1889–1897, 2015.
- [29] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. P. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In

*Proceedings of the 33rd International Conference on Machine Learning (ICML)*, pages 1928–1937, 2016. arXiv:1602.01783.

- [30] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra. Continuous control with deep reinforcement learning, 2019.
- [31] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv preprint*, arXiv:1707.06347, 2017.
- [32] S. Fujimoto, H. V. Hoof, and D. Meger. Addressing function approximation error in actor-critic methods. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pages 1587–1596, 2018.
- [33] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pages 1861–1870, 2018.
- [34] A. Kumar, A. Zhou, G. Tucker, and S. Levine. Conservative q-learning for offline reinforcement learning. In *Advances in Neural Information Processing Systems 33 (NeurIPS)*, pages 1179–1191, 2020.
- [35] K. Prudencio, J. L. Xiang, and A. T. Cemgil. A survey on offline reinforcement learning: Methodologies, challenges, and open problems. *arXiv preprint*, arXiv:2203.01387, 2022.
- [36] J. GarcÃa and F. FernÃ¡ndez. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 16(42):1437–1480, 2015.
- [37] F. Ghazalpour, S. Samangouei, and R. Vaughan. Hierarchical reinforcement learning: A comprehensive survey. *ACM Computing Surveys*, 54(12):1–35, 2021.
- [38] K. Song, J. Zhu, Y. Chow, D. Psomas, and M. Wainwright. A survey on multi-agent reinforcement learning: Foundations, advances, and open challenges. *IEEE Transactions on Neural Networks and Learning Systems*, 2024. In press, arXiv:2401.01234.
- [39] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. V. D. Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

- [40] O. Vinyals, I. Babuschkin, W. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019.
- [41] L. Espeholt, H. Soyer, R. Munos, K. Simonyan, V. Mnih, T. Ward, Y. Doron, V. Firoiu, T. Harley, I. Dunning, S. Legg, and K. Kavukcuoglu. Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pages 1407–1416, 2018.
- [42] M. Tan. Multi-agent reinforcement learning: Independent vs. cooperative agents. In *Proceedings of the 10th International Conference on Machine Learning (ICML)*, pages 330–337, 1993.
- [43] L. Panait and S. Luke. Cooperative multi-agent learning: The state of the art. *Autonomous Robots*, 8(3):355–377, 2005.
- [44] L. Buşoniu, R. Babuška, and B. D. Schutter. A comprehensive survey of multiagent reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, 38(2):156–172, 2008.
- [45] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Advances in Neural Information Processing Systems 30 (NeurIPS)*, pages 6379–6390, 2017.
- [46] P. Sunehag, G. Lever, A. Gruslys, W. Czarnecki, V. Zambaldi, M. Jaderberg, M. Lanctot, N. Sonnerat, J. Z. Leibo, K. Tuyls, and T. Graepel. Value-decomposition networks for cooperative multi-agent learning. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, 2018. arXiv:1706.05296.
- [47] T. Rashid, M. Samvelyan, C. S. de Witt, G. Farquhar, J. Foerster, and S. Whiteson. Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pages 4292–4301, 2018.
- [48] M. Samvelyan, T. Rashid, C. S. de Witt, G. Farquhar, J. Foerster, N. Nardelli, T. G. J. Rudner, and et al. The starcraft multi-agent challenge. *arXiv preprint*, arXiv:1902.04043, 2019.
- [49] K. Son, D. Kim, W. J. Kang, D. E. Hostallero, and Y. Yi. Qtran: Learning to factorize with transformation for cooperative multi-agent reinforcement learning.

In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 5887–5896, 2019.

- [50] A. Mahajan, T. Rashid, M. Samvelyan, and S. Whiteson. Maven: Multi-agent variational exploration. In *Advances in Neural Information Processing Systems 32 (NeurIPS)*, pages 7611–7622, 2019.
- [51] T. Wang, Y. Jiang, T. Da, W. Zhang, and J. Wang. Roma: Multi-agent reinforcement learning with emergent roles. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, pages 9876–9886, 2020.
- [52] K. Zhang, Z. Yang, and T. Başar. Multi-agent reinforcement learning: A selective overview of theories and algorithms. *Handbook of RL and Control*, 2021. arXiv:2106.05230.
- [53] A. Mitriakov, P. Papadakis, J. Kerdreux, and S. Garlatti. Reinforcement learning based, staircase negotiation learning: Simulation and transfer to reality for articulated tracked robots. *IEEE Robotics & Automation Magazine*, 28(4):10–20, 2021.
- [54] Y. Yu et al. Heterogeneous-agent reinforcement learning: An overview. *IEEE Transactions on Neural Networks and Learning Systems*, 2022. In press, arXiv:2203.00596.
- [55] N. B. LaFarge, D. Miller, K. C. Howell, and R. Linares. Autonomous closed-loop guidance using reinforcement learning in a low-thrust, multi-body dynamical environment. *Acta Astronautica*, 186:1–23, 2021.
- [56] D. Vallado and W. McClain. *Fundamentals of Astrodynamics and Applications*. Fundamentals of Astrodynamics and Applications. Microcosm Press, 2001.
- [57] 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.
- [58] 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.

- [59] S. Fujimoto, H. van Hoof, and D. Meger. Addressing function approximation error in actor-critic methods, 2018.
- [60] 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. *NeurIPS Autodiff Workshop*, 2017.
- [61] 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.
- [62] J. Achiam. Spinning Up in Deep Reinforcement Learning. *OpenAI*, 2018.
- [63] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization, 2017.

## Abstract

In this study, a robust guidance framework is presented for low-thrust spacecraft operating in multi-body dynamical environments (the Earth–Moon three-body system). The problem is formulated as a zero-sum differential game between a guidance agent (the spacecraft) and a disturbance agent (environmental uncertainties), and implemented using a centralized-training, decentralized-execution approach. In this vein, four continuous reinforcement-learning algorithms, DDPG, TD3, SAC, and PPO, are extended to their zero-sum multi-agent counterparts (MA-DDPG, MA-TD3, MA-SAC, and MA-PPO), and their training pipeline together with the network architectures is described in detail under a full-information setting. The algorithms are evaluated under diverse uncertainty scenarios, including random initial conditions, actuator disturbances, sensor noise, time delays, and model mismatch along a Lyapunov-orbit trajectory in the Earth–Moon system. The results clearly show that the zero-sum variants outperform their single-agent counterparts across all evaluation metrics. In particular, MA-TD3 preserves system stability while achieving the smallest trajectory deviation and the most efficient fuel consumption, even in the most challenging test scenarios. Ultimately, the proposed framework demonstrates that zero-sum differential-game-based multi-agent reinforcement learning can ensure adaptive and robust guidance for low-thrust spacecraft in the unstable regions of three-body systems without requiring precise modeling.

**Keywords:** Deep Reinforcement Learning, Differential Games, Multi-Agent Systems, Low-Thrust Guidance, Zero-Sum Games, Restricted Three-Body Problem, Robust Control.



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**

September 2025