



دانشگاه صنعتی شریف
دانشکده‌ی مهندسی هوافضا

پروژه کارشناسی
مهندسی کنترل

عنوان:

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

نگارش:

علی بنی اسد

استاد راهنما:

دکتر هادی نوبهاری

تیر ۱۴۰۱



به نام خدا
دانشگاه صنعتی شریف
دانشکده‌ی مهندسی هوافضا

پروژه کارشناسی

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

کمیته‌ی ممتحنین

استاد راهنما: دکتر هادی نوبهاری امضاء:

استاد مشاور: استاد مشاور امضاء:

استاد مدعو: استاد ممتحن امضاء:

تاریخ:

سپاس

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

چکیده

در این پژوهش، از یک روش مبتنی بر نظریه بازی^۱ به منظور کنترل وضعیت استند سه درجه آزادی چهارپره استفاده شده است. در این روش بازیکن اول سعی در ردگیری ورودی مطلوب می‌کند و بازیکن دوم با ایجاد اغتشاش سعی در ایجاد خطا در ردگیری بازیکن اول می‌کند. در این روش انتخاب حرکت با استفاده از تعادل نش^۲ که با فرض بدترین حرکت دیگر بازیکن است، انجام می‌شود. این روش نسبت به اغتشاش ورودی و همچنین نسبت به عدم قطعیت مدل‌سازی می‌تواند مقاوم باشد. برای ارزیابی عملکرد این روش ابتدا شبیه‌سازی‌هایی در محیط سیمولینک انجام شده است و سپس، با پیاده‌سازی روی استند سه درجه آزادی صحت عملکرد کنترل‌کننده تایید شده است.

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

^۱Game Theory

^۲Nash Equilibrium

فهرست مطالب

۱	یادگیری تقویتی	۱
۱	۱-۱ مفاهیم اولیه	۱
۲	۱-۱-۱ حالت و مشاهدات	۲
۲	۱-۱-۲ فضای عمل	۲
۲	۱-۱-۳ سیاست	۲
۳	۱-۱-۴ مسیر	۳
۳	۱-۱-۵ تابع پاداش و بازگشت	۳
۴	۱-۱-۶ ارزش در یادگیری تقویتی	۴
۵	۲-۱ عامل گرادیان سیاست عمیق قطعی	۵
۶	۱-۲-۱ یادگیری Q در DDPG	۶
۷	۲-۲-۱ سیاست در DDPG	۷
۸	۳-۲-۱ اکتشاف و بهره‌برداری در DDPG	۸
۸	۴-۲-۱ شبکه‌کد	۸
۱۰	۳-۱ عامل گرادیان سیاست عمیق قطعی تاخیری دوگانه	۱۰

فهرست جدول‌ها

فهرست شکل‌ها

۱-۱ حلقه تعامل عامل و محیط	۲
----------------------------	---

فصل ۱

یادگیری تقویتی

۱-۱ مفاهیم اولیه

بخش‌های اصلی یادگیری تقویتی^۱ شامل عامل^۲ و محیط^۳ است. عامل در محیط قرار دارد و با آن تعامل دارد. در هر مرحله از تعامل بین عامل و محیط، عامل یک مشاهده جزئی از وضعیت محیط انجام می‌دهد و سپس در مورد اقدامی که باید انجام دهد تصمیم می‌گیرد. وقتی عامل بر روی محیط عمل می‌کند، محیط تغییر می‌کند، اما ممکن است محیط به تنهایی نیز تغییر کند. عامل همچنین یک سیگنال پاداش^۴ از محیط دریافت می‌کند، عددی که به آن می‌گویند وضعیت فعلی محیط چقدر خوب یا بد است. هدف عامل به حداکثر رساندن پاداش انباشته خود است که بازگشت^۵ نام دارد. یادگیری تقویتی روش‌هایی هستند که عامل رفتارهای مناسب برای رسیدن به هدف خود را می‌آموزد. در شکل ۱-۱ تعامل بین محیط و عامل نشان داده شده است.

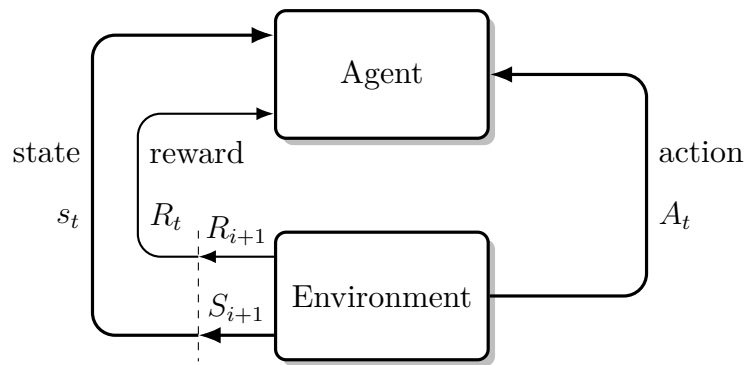
^۱ Reinforcement Learning (RL)

^۲ Agent

^۳ Environment

^۴ Reward

^۵ Return



شکل ۱-۱: حلقه تعامل عامل و محیط

۱-۱-۱ حالت و مشاهدات

حالت^۶ (s) توصیف کاملی از وضعیت محیط است. همه‌ی اطلاعات محیط در حالت وجود دارد. مشاهده^۷ (o) یک توصیف جزئی از حالت است که ممکن است تمامی اطلاعات نباشد.

۱-۱-۲ فضای عمل

فضای عمل در یادگیری تقویتی، مجموعه‌ای از تمام اقداماتی است که یک عامل می‌تواند در محیط خود انجام دهد. این فضا می‌تواند گسسته^۸ یا پیوسته^۹ باشد. در این پژوهش فضای عمل پیوسته و در یک بازه مشخص است.

۱-۱-۳ سیاست

یک سیاست^{۱۰} قاعده‌ای است که یک عامل برای تصمیم‌گیری در مورد اقدامات خود استفاده می‌کند. در این پژوهش سیاست قطعی^{۱۱} است، که به صورت زیر نشان داده می‌شود:

$$a_t = \pi(s_t) \quad (1-1)$$

^۶State

^۷Observation

^۸discrete

^۹continuous

^{۱۰}policy

^{۱۱}deterministic

در یادگیری تقویتی عمیق از سیاست‌های پارامتری شده استفاده می‌شود. خروجی این سیاست‌ها از توابعی هستند که به مجموعه‌ای از پارامترها (مثلاً وزن‌ها و بایاس‌های یک شبکه عصبی) بستگی دارند که می‌توان آنها را برای تغییر رفتار از طریق برخی الگوریتم‌های بهینه‌سازی تنظیم کرد. در این پژوهش پارامترهای سیاست را با θ نشان داده شده است و سپس نماد آن به عنوان یک زیروند روی سیاست مانند معادله (۲-۱) نشان داده شده است.

$$a_t = \pi_{\theta}(s_t) \quad (2-1)$$

۴-۱-۱ مسیر

یک مسیر^{۱۲} توالی‌ای از حالت‌ها و عمل‌ها در محیط است.

$$\tau = (s_0, a_0, s_1, a_1, \dots) \quad (3-1)$$

گذار حالت^{۱۳} به اتفاقاتی که در محیط بین حالت در زمان s و حالت در زمان $s+1$ می‌افتد، گفته می‌شود. این گذارها توسط قوانین طبیعی محیط انجام می‌شوند و تنها به آخرین اقدام انجام شده توسط عامل (a_t) بستگی دارند. گذار حالت را می‌توان به صورت زیر تعریف کرد.

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

۵-۱-۱ تابع پاداش و بازگشت

تابع پاداش^{۱۴} حالت فعلی محیط، آخرین عمل انجام شده و حالت بعدی محیط بستگی دارد. تابع پاداش را می‌توان به صورت زیر تعریف کرد.

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

در این پژوهش پاداش تنها تابعی از جفت حالت-عمل ($r_t = R(s_t, a_t)$) است. هدف عامل این است که مجموع پاداش‌های به دست آمده در طول یک مسیر را به حداکثر برساند، اما این مفهوم می‌تواند چند معنی داشته باشد. در این پژوهش این موارد را با نماد $R(\tau)$ نشان داده شده است و به آن تابع بازگشت^{۱۵}

^{۱۲}Trajectory

^{۱۳}state transition

^{۱۴}reward function

^{۱۵}Return

گفته می‌شود. یکی از انواع بازگشت، بازگشت بدون تنزیل با افق محدود^{۱۶} است که مجموع پاداش‌های به‌دست‌آمده در یک بازه زمانی ثابت از مسیر به‌صورت زیر است.

$$R(\tau) = \sum_{t=0}^T r_t \quad (۶-۱)$$

نوع دیگری از بازگشت، بازگشت تنزیل‌شده با افق نامحدود^{۱۷} است که مجموع همه پاداش‌هایی است که تا به حال توسط عامل به دست آمده است، اما با در نظر گرفتن فاصله زمانی‌ای که تا دریافت آن پاداش وجود داشته، تنزیل^{۱۸} شده است. این فرمول پاداش شامل یک فاکتور تنزیل^{۱۹} با نماد γ است.

$$R(\tau) = \sum_{t=0}^{\infty} \gamma^t r_t \quad (۷-۱)$$

۱-۱-۶ ارزش در یادگیری تقویتی

در یادگیری تقویتی، دانستن ارزش^{۲۰} یک حالت یا جفت حالت-عمل ضروری است. منظور از ارزش، بازگشت مورد انتظار^{۲۱} 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 کرد. توابع ارزش به شکلی در تقریباً تمام الگوریتم‌های یادگیری تقویتی به کار می‌روند. در اینجا به چهار تابع مهم اشاره می‌کنیم.

۱. تابع ارزش تحت سیاست^{۲۲} $(V^\pi(s))$: این تابع، بازگشت مورد انتظار را در صورتی که از حالت s شروع شود و همیشه طبق سیاست π عمل شود، خروجی می‌دهد.

$$V^\pi(s) = \mathbb{E}_{\tau \sim \pi} [R(\tau) | s_0 = s] \quad (۸-۱)$$

۲. تابع ارزش-عمل تحت سیاست^{۲۳} $(Q^\pi(s, a))$: این تابع، بازگشت مورد انتظار را در صورتی که از حالت s شروع شود، یک اقدام دلخواه a (که ممکن است از سیاست π نباشد) انجام شود و سپس برای همیشه طبق سیاست π عمل شود، خروجی می‌دهد.

$$Q^\pi(s, a) = \mathbb{E}_{\tau \sim \pi} [R(\tau) | s_0 = s, a_0 = a] \quad (۹-۱)$$

^{۱۶}Finite-Horizon Undiscounted Return

^{۱۷}Infinite-Horizon Discounted Return

^{۱۸}Discount

^{۱۹}Discount Factor

^{۲۰}Value

^{۲۱}Expected Return

^{۲۲}On-Policy Value Function

^{۲۳}On-Policy Action-Value Function

۳. تابع ارزش بهینه^{۲۴} ($V^*(s)$): این تابع، بازگشت مورد انتظار را در صورتی که از حالت s شروع شود و همیشه طبق سیاست بهینه در محیط عمل شود، خروجی می‌دهد.

$$V^*(s) = \max_{\pi}(V^{\pi}(s)) \quad (10-1)$$

۴. تابع ارزش-عمل بهینه^{۲۵} ($Q^*(s, a)$): این تابع، بازگشت مورد انتظار را 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، خروجی می‌دهد.

$$Q^*(s, a) = \max_{\pi}(Q^{\pi}(s, a)) \quad (11-1)$$

۲-۱ عامل گرادیان سیاست عمیق قطعی

گرادیان سیاست عمیق قطعی^{۲۶} الگوریتمی است که همزمان یک تابع Q و یک سیاست را یاد می‌گیرد. این الگوریتم برای یادگیری تابع Q از داده‌های غیرسیاست محور^{۲۷} 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 داده شده عمل بهینه را می‌توان با حل کردن معادله (۱۲-۱) به دست آورد.

$$a^*(s) = \arg \max_a Q^*(s, a) \quad (12-1)$$

الگوریتم DDPG ترکیبی از یادگیری تقریبی برای $Q^*(s, a)$ و یادگیری تقریبی برای $a^*(s)$ است و به نحوی طراحی شده است که برای محیط‌هایی با فضاهاى عمل پیوسته مناسب باشد. روش محاسبه $a^*(s)$ در این الگوریتم آن را برای فضای پیوسته مناسب می‌کند. از آنجا که فضای عمل پیوسته است، فرض می‌شود که تابع $Q^*(s, a)$ نسبت به آرگومان عمل مشتق‌پذیر است. مشتق‌پذیری این امکان را می‌دهد که یک روش یادگیری مبتنی بر گرادیان برای سیاست $\mu(s)$ استفاده شود. سپس، 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.

^{۲۴}Optimal Value Function

^{۲۵}Optimal Action-Value Function

^{۲۶}Deep Deterministic Policy Gradient (DDPG)

^{۲۷}Off-Policy

۱-۲-۱ یادگیری Q در DDPG

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

$$Q^*(s, a) = \mathbb{E}_{s' \sim P} \left[r(s, a) + \gamma \max_{a'} Q^*(s', a') \right] \quad (۱۳-۱)$$

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

$$L(\phi, \mathcal{D}) = \mathbb{E}_{(s, a, r, s', d) \sim \mathcal{D}} \left[\left(Q_\phi(s, a) - \left(r + \gamma(1 - d) \max_{a'} Q_\phi(s', a') \right) \right)^2 \right] \quad (۱۴-۱)$$

در الگوریتم DDPG دو ترفند برای عملکرد بهتر استفاده شده است که در ادامه به بررسی آن پرداخته شده است.

• بافرهای بازی

الگوریتم‌های یادگیری تقویتی جهت آموزش یک شبکه عصبی عمیق برای تقریب $Q^*(s, a)$ از بافرهای بازی^{۲۸} تجربه شده استفاده می‌کنند. این مجموعه \mathcal{D} شامل تجربیات قبلی است. برای داشتن رفتار پایدار در الگوریتم، بافر 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، بیش‌برازش^{۲۹} رخ می‌دهد 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.

$$r + \gamma(1 - d) \max_{a'} Q_\phi(s', a') \quad (۱۵-۱)$$

^{۲۸}Replay Buffers

^{۲۹}Overfit

در هنگام کمینه کردن تابع خطای میانگین مربعات بلمن، سعی شده است تا تابع Q شبیه تر به این هدف یعنی رابطه (۱-۱۵) شود. اما مشکل این است که هدف بستگی به پارامترهای در حال آموزش ϕ دارد. این باعث ایجاد ناپایداری در کمینه کردن تابع خطای میانگین مربعات بلمن می شود. راه حل آن استفاده از یک مجموعه پارامترهایی که با تأخیر زمانی به ϕ نزدیک می شوند. به عبارت دیگر، یک شبکه دوم ایجاد می شود که به آن شبکه هدف گفته می شود. شبکه هدف دنباله ای شبکه اول را دنبال می کند. پارامترهای شبکه هدف با نشان ϕ_{targ} نشان داده می شوند. در الگوریتم DDPG، شبکه هدف در هر به روزرسانی شبکه اصلی، با میانگین گیری پولیاک^{۳۰} به روزرسانی می شود.

$$\phi_{\text{targ}} \leftarrow \rho \phi_{\text{targ}} + (1 - \rho) \phi \quad (۱-۱۶)$$

در رابطه بالا ϕ یک فرایارامتر^{۳۱} است که بین صفر و یک انتخاب می شود. در این پژوهش این مقدار نزدیک به یک در نظر گرفته شده است.

الگوریتم DDPG نیاز به یک شبکه سیاست هدف ($\mu_{\theta_{\text{targ}}}$) برای محاسبه عمل هایی که به طور تقریبی بیشینه $Q_{\phi_{\text{targ}}}$ را حاصل کند، را دارد. برای رسیدن به این شبکه سیاست هدف از همان روشی که تابع Q به دست می آید یعنی با میانگین گیری پولیاک از پارامترهای سیاست در طول زمان آموزش استفاده می شود. با در نظر گرفتن موارد اشاره شده، یادگیری Q در DDPG با کمینه کردن تابع خطای میانگین مربعات بلمن (MSBE) یعنی معادله (۱-۱۷) با استفاده از کاهش گرادیان تصادفی^{۳۲} انجام می شود.

$$L(\phi, \mathcal{D}) = \mathbb{E}_{(s,a,r,s',d) \sim \mathcal{D}} \left[\left(Q_{\phi}(s,a) - (r + \gamma(1-d)Q_{\phi_{\text{targ}}}(s', \mu_{\theta_{\text{targ}}}(s'))) \right)^2 \right] \quad (۱-۱۷)$$

۲-۲-۱ سیاست در DDPG

در این بخش یک سیاست تعیین شده $\mu_{\theta}(s)$ یاد گرفته می شود تا عملی را انجام می دهد 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 مشتق پذیر است، معادله زیر با استفاده از صعود گرادیان^{۳۳} (تنها نسبت به پارامترهای سیاست) حل می شود.

$$\max_{\theta} \mathbb{E}_{s \sim \mathcal{D}} [Q_{\phi}(s, \mu_{\theta}(s))] \quad (۱-۱۸)$$

^{۳۰} Polyak Averaging

^{۳۱} Hyperparameter

^{۳۲} Stochastic Gradient Descent

^{۳۳} Gradient Ascent

۳-۲-۱ اکتشاف و بهره‌برداری در DDPG

برای بهبود اکتشاف^{۳۴} در سیاست‌های DDPG، در زمان آموزش نويز به عمل‌ها اضافه می‌شود. نویسندگان مقاله اصلی DDPG توصیه کرده‌اند که نويز OU^{۳۵} با زمان‌بندی هم‌ارتباطی^{۳۶} اضافه شود. در زمان سنجش بهره‌برداری^{۳۷} سیاست از آنچه یادگرفته است، نويز به عمل‌ها اضافه نمی‌شود.

۴-۲-۱ شبه‌کد

در این بخش الگوریتم DDPG پیاده‌سازی شده آورده شده است. در این پژوهش الگوریتم ۱ در محیط پایتون با استفاده از کتابخانه TensorFlow پیاده‌سازی شده است.

^{۳۴}Exploration

^{۳۵}Ornstein-Uhlenbeck

^{۳۶}Time-Correlated

^{۳۷}Exploitation

الگوریتم ۱ گرادیان سیاست عمیق قطعی

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

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

تا وقتی همگرایی رخ دهد:

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

عمل a را در محیط اجرا کنید.

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

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

اگر زمان به روزرسانی فرا رسیده است:

به ازای هر تعداد به روزرسانی:

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

اهداف 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 به روزرسانی کنید:

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

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

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

شبکه‌های هدف را با استفاده از معادلات زیر به روزرسانی کنید:

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

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

۳-۱ عامل گرادیان سیاست عمیق قطعی تاخیری دوگانه

مراجع

Abstract

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

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



Sharif University of Technology
Department of Aerospace Engineering

Bachelor Thesis

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

By:

Ali BaniAsad

Supervisor:

Dr.Hadi Nobahari

July 2022