به نام خدا

دانشگاه تهران
پردیس دانشکده‌های فنی
دانشکده مکانیک

کنترل آونگ معکوس دورانی به کمک الگوریتم‌های یادگیری تقویتی

پروژه تخصصی کارشناسی

رشته مهندسی مکانیک

نویسنده: آرش هاتفی

استاد راهنما: دکتر مسعود شریعت پناهی

بهمن ۱۳۹۹

تشکر و قدردانی

بدینوسیله، از آقایان جناب دکتر شریعت‌پناهی و جناب دکتر کاشانی برای حمایت‌های دلسوزانه‌شان در تمامی مراحل انجام این پروژه تشکر و سپاسگزاری می‌نمایم.

چکیده

در سال‌های اخیر، در پی پیشرفت‌های چشمگیر در هوش مصنوعی و یادگیری ماشینی، روش‌های هوشمند در حوزه‌ی کنترل محبوبیت فراوانی یافته‌اند. تاکنون عملکرد این الگوریتم‌ها همواره توسط بسیاری از مسائل مختلف به چالش کشیده شده و بسیاری از پیشرفت‌های صورت گرفته در روش‌های مذکور ضمن تلاش برای بهبود عملکردشان روی یک مسئله‌ی خاص اتفاق افتاده است. در این پروژه قصد داریم به بررسی عملکرد الگوریتم‌های یادگیری تقویتی، دسته‌ای از الگوریتم‌های کنترلی هوشمند، در کنترل آونگ معکوس دورانی (Rotary Inverted Pendulum) بپردازیم. دینامیک پیچیده و بسیار ناپایدار آونگ معکوس دورانی، همواره این سیستم را برای پژوهشگران حوزه‌ی کنترل جذاب نموده تا جایی که این مسئله را به‌عنوان یکی از مسائل معیار در حوزه‌ی مهندسی کنترل جهت مقایسه‌ی عملکرد روش‌های مختلف بدل نموده است. در گام نخست پروژه، یک آونگ معکوس دورانی جهت اجزای فرایند آموزش الگویتم‌های هوشمند ساخته و آماده‌سازی شد. به‌موازات فرایند ساخت، به پیاده‌سازی یک مدل کامپیوتری دقیق از آونگ جهت تسریع فرایند آموزش در محیط شبیه‌سازی مجازی نیز پرداخته شد. درنهایت، سعی شد تا با بهره‌برداری مناسب از امکانات محاسباتی موجود، عملکرد دو الگوریتم یادگیری تقویتی از نوع گسسته (Deep Q-Learning) و پیوسته (Deep Deterministic Policy Gradient) بر روی مسئله‌ی کنترل آونگ معکوس دورانی سنجیده شود. هدف نهایی پروژه، دستیابی به درکی دقیق از نقاط ضعف و قوت روش‌های یادگیری تقویتی در مواجهه با مسائل پیچیده‌ی کنترل بی‌درنگ می‌باشد چراکه این کار گام اولیه و اساسی در راستای برطرف شدن نقاط ضعف و ایرادات وارد بر این روش‌ها است. عملگرد الگوریتم یادگیری تقویتی پیوسته، در محیط شبیه‌سازی از عملکرد الگوریتم گسسته پیشی‌گرفت. در نقطه‌ی مقابل، به علت کمتربودن حجم محاسبات، الگوریتم گسسته توانست در عمل سازه‌ی واقعی را بهتر کنترل نماید.

کلمات کلیدی: آونگ معکوس دورانی، کنترل هوشمند، یادگیری تقویتی، یادگیری تقویتی گسسته، یادگیری تقویتی پیوسته

فهرست مطالب

فهرست اشکال

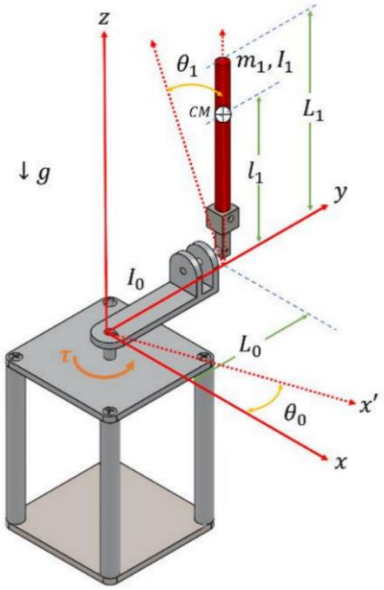
فهرست جداول

فهرست شبه‌کدها

۱- مقدمه

۱-۱- تعریف مسئله

آونگ معکوس دورانی[۱] یک سیستم مکانیکی با دو درجه آزادی و یک محرک است. این سیستم، شامل یک بازوی دوار[۲] است که به‌وسیله یک موتور حرکت دورانی می‌کند. بر لبه بازو، یک آونگ[۳] متصل شده است که در صفحه‌ی عمود بر بازو حرکت دورانی می‌کند. شکل ۱، طرح شماتیک آونگ معکوس دورانی را نشان می‌دهد:



شکل ۱: شماتیک آونگ معکوس دورانی *[1]*

هدف از کنترل آونگ معکوس دورانی، نگه‌داشتن آونگ در وضعیت قائم (به‌صورت معکوس) به کمک حرکت دورانی موتور است. دینامیک غیرخطی و ناپایدار آونگ معکوس دورانی موجب اهمیت آن به‌عنوان یکی از مسائل معیار[۴] در حوزهٔ کنترل‌شده است. در سال‌های اخیر، کنترل پایداری مکانیزم‌های مبتنی بر آونگ معکوس کاربرد گسترده‌ای در صنایع مختلف یافته است. کنترل پایداری موتورسیکلت‌های بدون سرنشین، تخته‌روها (مانند Segway) و ربات‌های دوچرخ و کروی از نمونه‌های کاربرد این فناوری به شمار می‌آیند.

---

[1] Rotary Inverted Pendulum
[2] Arm
[3] Pendulum
[4] Benchmark

۱-۲- ضرورت اجرای پروژه

در سال‌های اخیر، در پی پیشرفت‌های چشمگیر در هوش مصنوعی و یادگیری ماشینی، روش‌های هوشمند در حوزه‌ی کنترل محبوبیت فراوانی یافته‌اند. توانایی این روش‌ها در فراگیری روش کنترل محیط‌هایی پیچیده، بدون نیاز به آگاهی از معادلات و دینامیک آن‌ها، همواره موردتوجه علاقه‌مندان این حوزه بوده است. بدون شک یکی از مهم‌ترین ویژگی الگوریتم‌های هوشمند، توانایی هماهنگ شدنشان با تغییرات ناخواسته در سیستم و مقاومت بالایشان در مقابل عدم قطعیت‌های موجود است. در مقابل، بزرگ‌ترین ایراد وارده به این روش‌ها، پیچیدگی پیاده‌سازی آن‌ها و فرایند آموزش طولانی می‌باشد. به همین علت، قبل از جایگزینی روش‌های کنترل کلاسیک، الگوریتم‌های هوشمند راه طولانی در پیش دارند. تحقیقات بسیاری در زمینه‌ی کاهش پیچیدگی و مدت‌زمان یادگیری الگوریتم‌های هوشمند انجام‌شده است و بسیاری از موفقیت‌های این زمینه، ضمن به چالش کشیدن الگوریتم‌های هوشمند روی مسائل خاص صورت گرفته است.

مسئله‌ی کنترل آونگ معکوس دورانی (Rotary Inverted Pendulum)، به علت دینامیک پیچیده و به‌شدت ناپایدارش، همواره موردتوجه پژوهشگران حوزه‌ی کنترل قرارگرفته تا آنجا که از آن به‌عنوان یک مسئله‌ی معیار جهت مقایسه‌ی روش‌های مختلف کنترل یاد می‌شود. از این تئوری موجود در این سیستم، برای مدل‌سازی وسایل گوناگونی ازجمله سگوی[1] و ربات خود تعادل[2] استفاده می‌شود. در این پروژه، قصد داریم عملکرد کنترلرهای مبتنی بر یادگیری تقویتی[3] پیوسته و گسسته را بر روی سیستم آونگ معکوس دورانی بسنجیم. به‌عنوان یک مسئله‌ی پیچیده و معیار، انتظار می‌رود که نتایج حاصل از این پژوهش دیدی دقیق و کامل از نقاط ضعف و قوت الگوریتم‌های یادگیری تقویتی به ما بدهد.

گام بعدی این پژوهش، می‌تواند استفاده از نتایج به‌دست‌آمده جهت بهبود عملکرد این الگوریتم‌ها در مسئله‌ی کنترل آونگ معکوس دورانی باشد. این کار نه‌تنها منجر به طراحی یک سیستم کنترلی بهتر مبتنی بر یادگیری تقویتی برای آونگ معکوس دورانی شود، می‌تواند به‌طور غیرمستقیم به توسعه‌ی روش‌های طراحی سیستم‌های یادگیری تقویتی نیز کمک شایانی نماید.

[1] Segway
[2] Self-Balancing Robot
[3] Reinforcement Learning

۱-۳- پیشینه‌ی پژوهش

۱-۳-۱- پیشینه‌ی پژوهش در دنیا

مقاله‌های گوناگونی پیرامون کنترل سیستم آونگ معکوس با روش‌های مختلف کنترلی در مجله‌های علمی به چاپ رسیده است. از میان این روش‌ها می‌توان به کنترلرهای خانواده PID [2] [3]، کنترل فازی [4]، روش LQR [5] و یا روش‌های هوشمند مانند الگوریتم ژنتیک، الگوریتم ازدحام ذرات[1] [6] و ... اشاره نمود. باوجود بررسی عملکرد تعداد زیادی از روش‌های کنترلی هوشمند جهت کنترل آونگ معکوس دورانی، اکثر پژوهش‌های انجام شده بر روی مدل‌سازی کامپیوتری تمرکز دارند و در کمتر مقاله‌ای به بررسی عملکرد کنترلر بر روی سیستم فیزیکی پرداخته‌شده است.

درزمینه‌ی ساخت و کنترل مکانیزم آونگ معکوس، شرکت کوآنسر[2]، از شرکت‌های مطرح درزمینه‌ی ساخت لوازم آزمایشگاهی، فعال می‌باشد. این شرکت محصولی به نام Quanser Qube تولید کرده که قابلیت تبدیل‌شدن به تعدادی از مسائل کلاسیک و معیار حوزه‌ی کنترل، ازجمله مسئله‌ی آونگ معکوس دورانی، را دارد.

شکل ۲: *Quanser Qube* برای کنترل آونگ معکوس دورانی[3]

[1] Particle Swarm Optimization (PSO)
[2] Quanser

[3] برگرفته از Quanser.com

۱-۳-۲- پیشینه پژوهش در دانشکده مکانیک دانشگاه تهران

پروژه ساخت آونگ معکوس دورانی از اواخر سال ۱۳۹۸ در آزمایشگاه کاربردهای هوش مصنوعی در مکانیک (AIME[1] Lab.) به هدف بررسی عملکرد انواع روش‌های کنترلی بر روی یک مسئله معیار[2] در حوزه کنترل آغاز شد. در گام نخست، سازه‌ی آونگ معکوس دورانی ساخته شد و جهت بررسی عملکرد کنترلر PID مورد مطالعه قرار گرفت [7]. تصویر زیر، نمایی از آونگ دورانی معکوسی ساخته‌شده می‌باشد:

شکل ۳: نمونه اولیه آونگ معکوس دورانی ساخته‌شده

در پژوهش بعدی به بررسی عملکرد الگوریتم Q-Learning گسسته (از الگوریتم‌های حوزه یادگیری تقویتی[3]) جهت پایدار کردن آونگ معکوس دورانی پرداخته‌شد و نتایج به‌دست‌آمده از الگوریتم هوشمند با نتایج پیشین به‌دست‌آمده از کنترل PID مقایسه گردید [8]. متاسفانه، برخی از مشکلات موجود در سازه باعث ایجاد اختلال در روند آموزش آونگ شد و عملکرد نهایی الگوریتم Q-Learning از مقدار مورد انتظار ضعیف‌تر بود.

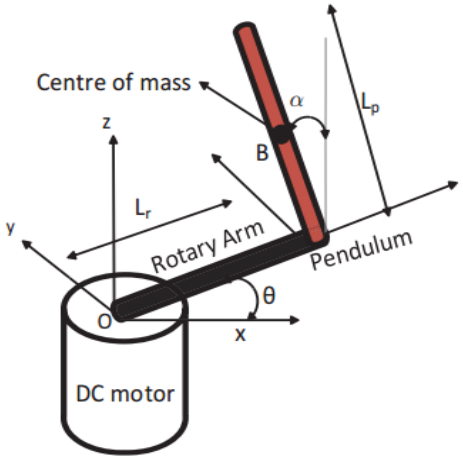
[1] Artificial Intelligence in Mechanical Engineering
[2] Benchmark
[3] Reinforcement Learning

۲- معادلات ریاضی حاکم بر سیستم

در این بخش به بررسی قوانین و معادلات فیزیکی حاکم بر سازه‌ی آونگ معکوس دورانی خواهیم پرداخت. اگرچه استفاده از این روابط در کنترل سازه به کمک الگوریتم‌های یادگیری تقویتی ضروری نیست، دانستن این معادلات برای فرایند شبیه‌سازی[1] بسیار مهم و حائز اهمیت می‌باشد.

۲-۱- دینامیک آونگ معکوس

شکل زیر، شماتیک آونگ معکوس دورانی را نشان می‌دهد:



شکل ۴: آونگ معکوس دورانی[2]

همچنین، پارامترهای دخیل در مدل‌سازی دینامیک آونگ به شرح موجود در جدول زیر می‌باشند:

جدول ۱: پارامترهای دخیل در دینامیک آونگ معکوس دورانی

نماد	معنی	واحد
θ	مکان زاویه‌ای بازو	rad
α	مکان زاویه‌ای آونگ	rad
$[x_r, y_r, z_r]$	مؤلفه‌های مکان انتهای بازو	m
$[x_p, y_p, z_p]$	مؤلفه‌های مکان مرکز جرم آونگ	m
m_p	جرم آونگ	kg
J_p	اینرسی آونگ	$kg.m^2$
L_p	طول آونگ	m
c_p	ضریب میرایی ویسکوز اتصال آونگ و بازو	$N.m.s/rad$
L_r	طول بازوی دوار	m
J_a	اینرسی بازوی دوار	$kg.m^2$
c_a	ضریب میرایی ویسکوز اتصال بازو و موتور	$N.m.s/rad$

[1] Simulation

[2] برگرفته از degruyter.com

در ادامه، معادلات دینامیکی حاکم بر آونگ را به دست می‌آوریم.

برای نقطه‌ی محل اتصال آونگ به بازو داریم:

$$\begin{bmatrix} x_r \\ y_r \end{bmatrix} = \begin{bmatrix} L_r \cos\theta \\ L_r \sin\theta \end{bmatrix} \quad (Equation\ 1)$$

با مشتق‌گیری از معادله‌ی ۱ سرعت انتهای بازو به‌صورت زیر به دست می‌آید.

$$\begin{bmatrix} \dot{x}_r \\ \dot{y}_r \end{bmatrix} = \begin{bmatrix} -L_r\dot{\theta}\sin\theta \\ L_r\dot{\theta}\cos\theta \end{bmatrix} \quad (Equation\ 2)$$

با فرض توزیع یکنواخت جرم آونگ در طول آن، مرکز جرم آونگ در وسط آن قرار می‌گیرد. برای این نقطه می‌توان نوشت:

$$\begin{bmatrix} x_p \\ y_p \\ z_p \end{bmatrix} = \begin{bmatrix} L_r\cos\theta + \dfrac{L_p \sin\alpha \sin\theta}{2} \\ L_r\sin\theta - \dfrac{L_p \sin\alpha \cos\theta}{2} \\ \dfrac{L_p}{2}\cos\alpha \end{bmatrix} \quad (Equation\ 3)$$

با مشتق‌گیری از معادله‌ی ۳ سرعت مرکز جرم آونگ به‌صورت زیر به دست می‌آید.

$$\begin{bmatrix} \dot{x}_p \\ \dot{y}_p \\ \dot{z}_p \end{bmatrix} = \begin{bmatrix} -L_r\dot{\theta}\sin\theta + \dfrac{L_p\dot{\alpha}\cos\alpha\sin\theta}{2} + \dfrac{L_p\dot{\theta}\sin\alpha\cos\theta}{2} \\ L_r\dot{\theta}\sin\theta - \dfrac{L_p\dot{\alpha}\cos\alpha\cos\theta}{2} + \dfrac{L_p\dot{\theta}\sin\alpha\sin\theta}{2} \\ \dfrac{L_p}{2}\dot{\alpha}\sin\alpha \end{bmatrix} \quad (Equation\ 4)$$

نیروی جنبشی کل سیستم از جمع انرژی جنبشی آونگ و بازوی دوار به دست می‌آید که به‌صورت زیر است:

$$KE = KE_{arm} + KE_{pend} \quad (Equation\ 5)$$

با استفاده از معادله‌ی ۵ و مقادیر به‌دست‌آمده برای سرعت‌ها داریم:

$$KE = \frac{1}{2}\left(m_p {L_r}^2 + J_r + m_p \frac{{L_p}^2}{4} sin^2\,\alpha\right)\dot{\theta}^2 + \frac{1}{2}\left(m_p \frac{{L_p}^2}{4} + J_p\right)\dot{\alpha}^2 - \frac{1}{2}\left(m_p L_r L_p \dot{\theta}\dot{\alpha}\, cos\,\alpha\right) \qquad (Equation\ 6)$$

به‌طور مشابه، برای انرژی پتانسیل کل سیستم داریم:

$$PE = PE_{arm} + PE_{pend} \qquad (Equation\ 7)$$

ازآنجاکه بازوی دوار حرکت عمودی ندارد و تغییرات انرژی پتانسیل صفر است، مجموع انرژی جنبشی از رابطه‌ی زیر به دست می‌آید:

$$PE = 0 + \left(-m_p g\left(\frac{L_p}{2} - \frac{L_p}{2} cos\,\alpha\right)\right) \qquad (Equation\ 8)$$

پارامتر لاگرانژین[1] برای سیستم، عبارت است از تفاضل انرژی جنبشی و انرژی پتانسیل سیستم که به‌صورت زیر تعریف می‌شود:

$$\mathcal{L} = KE - PE \qquad (Equation\ 9)$$

با قرار دادن مقادیر به‌دست‌آمده برای انرژی جنبشی و پتانسیل سیستم در معادله‌ی ۹ داریم:

$$\mathcal{L} = \frac{1}{2}\left(m_p {L_r}^2 + J_r + m_p \frac{{L_p}^2}{4} sin^2\,\alpha\right)\dot{\theta}^2 + \frac{1}{2}\left(m_p \frac{{L_p}^2}{4} + J_p\right)\dot{\alpha}^2 - \frac{1}{2}\left(m_p L_r L_p \dot{\theta}\dot{\alpha}\, cos\,\alpha\right) + m_p g\left(\frac{L_p}{2} - \frac{L_p}{2} cos\,\alpha\right) \qquad (Equation\ 10)$$

تابع رایلی[2] برای این سیستم عبارت است از:

$$R = \frac{1}{2} c_r \dot{\theta}^2 + \frac{1}{2} c_p \dot{\alpha}^2 \qquad (Equation\ 11)$$

[1] Lagrangian
[2] Riley Function

به دلیل این‌که سیستم را می‌توان به ۲ درجه آزادی تقسیم کرد، درنتیجه ۲ معادله لاگرانژ برای این سیستم وجود دارد که به‌صورت زیر نوشته می‌شوند:

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\theta}}\right)-\frac{\partial \mathcal{L}}{\partial \theta}+\frac{\partial R}{\partial \dot{\theta}}=\tau \qquad , \qquad \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\alpha}}\right)-\frac{\partial \mathcal{L}}{\partial \dot{\alpha}}+\frac{\partial R}{\partial \dot{\alpha}}=0 \qquad (Equations\ 12\&13)$$

حال با قرار دادن پارامترهای به‌دست‌آمده از معادلات قبل و خطی سازی سیستم پیرامون نقطه‌ی کاری سیستم ($\alpha = 0$) به کمک سری تیلور[1] حول صفر، معادلات حرکت سیستم حول نقطه‌ی $\alpha = 0$ به‌صورت زیر به دست می‌آید:

$$(m_p L_r^2 + J_r)\ddot{\theta} - \frac{1}{2}m_p L_p L_r \ddot{\alpha} + c_r \dot{\theta} = \tau \quad (Equation\ 14)$$

$$-\frac{1}{2}m_p L_p L_r \ddot{\theta} + (J_p + \frac{1}{4}m_p L_p^2)\ddot{\alpha} - \frac{m_p L_p g \alpha}{2} + c_p \dot{\alpha} = 0 \quad (Equation\ 15)$$

با ساده‌سازی معادلات بالا، روابط زیر برای شتاب زاویه‌ای بازوی دوار و آونگ به دست می‌آید:

$$\ddot{\theta} = \frac{1}{J_T}\Big(-\Big(J_p + \frac{1}{4}m_p L_p^2\Big)c_r \dot{\theta} - \frac{1}{2}m_p L_p L_r c_p \dot{\alpha} + \frac{1}{4}m_p^2 L_p^2 L_r g \alpha + \Big(J_p + \frac{1}{4}m_p L_p^2\Big)\tau\Big) \quad (Equation\ 16)$$

$$\ddot{\alpha} = \frac{1}{J_T}(\frac{1}{2}m_p L_p L_r c_r \dot{\theta} - (J_r + m_p L_r^2)c_p \dot{\alpha} + \frac{1}{2}m_p L_p g \alpha - \frac{1}{2}m_p L_p L_r \tau) \quad (Equation\ 17)$$

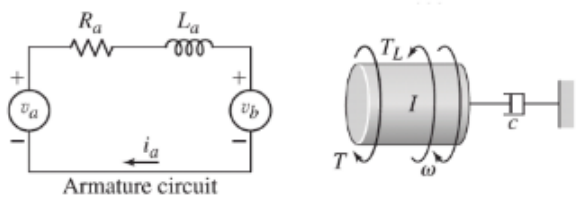
در معادلات ۱۶ و ۱۷، J_T یک پارامتر کمکی است که به‌صورت زیر تعریف می‌شود:

$$J_T = J_P m_p L_r^2 + J_r J_p + \frac{1}{4}J_r m_p L_p^2 \quad (Equation\ 18)$$

[1] Tylor Series

۲-۲- موتور DC

در آونگ معکوس دورانی ساخته شده، از یک موتور DC، به‌عنوان محرک سیستم استفاده می‌شود. در این بخش، فرمولاسیون ریاضی برای مدل‌سازی موتور DC آورده شده است. موتور DC را می‌توان به‌صورت یک سیستم خطی با شماتیک زیر در نظر گرفت:



شکل ۵: مدل موتور *DC* آرمیچر[1]

همچنین، پارامترهای دخیل در مدل‌سازی موتور DC به شرح موجود در جدول زیر می‌باشند:

جدول ۲: پارامترهای دخیل در مدل‌سازی موتور *DC*

نماد	معنی	واحد
V_a	ولتاژ آرمیچر	$Volt$
i_a	جریان آرمیچر	$Amps$
R_a	مقاومت آرمیچر	$Ohms$
L_a	اندوکدانس آرمیچر	$Henry$
V_b	ولتاژ Back emf	$Volts$
T	گشتاور موتور	$N.m$
I	اینرسی روتور	$Kg.m^2$
T_L	گشتاور بار	$N.m$
θ	زاویه شفت موتور	Rad
C	ضریب میرایی	$N.m.s/rad$
K_t	ضریب گشتاور	$N.m/A$
K_b	ضریب Back emf	$V.s/m$

در ادامه، معادلات حاکم بر موتور را می‌نویسیم:

$$R_a i_a + L_a \frac{di_a}{dt} + V_b = V_a \qquad (Equation\ 19) \qquad V_b = K_b \frac{d\theta}{dt} \quad (Equation\ 20)$$

$$I \frac{d^2\theta}{dt^2} + C \frac{d\theta}{dt} = T - T_L \quad V_b = K_b \frac{d\theta}{dt} (Equation\ 21) \qquad T = K_t i_a \qquad (Equation\ 22)$$

[1] برگرفته از ctms.engin.umich.edu

۳- اصلاحات سازه‌ی آونگ معکوس دورانی

فرایند یادگیری در روش‌های کنترل هوشمند، ازجمله یادگیری تقویتی[1]، اکثراً زمان‌بر هستند. به همین جهت، لازم است تا سازه‌ی فیزیکی برای کارکرد چندین ساعته مستحکم و آماده شود. سازه آونگ معکوس پیشین، دارای چندین نقص بود که پیش از هر چیز لازم بود تا اصلاح شوند. برخی از این نقایص در زیر آورده شده‌اند:

۱- بازوی سازه‌ی آونگ معکوس به‌وسیله‌ی یک پیچ به‌موازات شفت موتور، به این شفت متصل می‌شد. لذا، جدای از میزان سفت شدن، به علت اینرسی نسبتاً زیاد بازو و آونگ، بعد از کارکرد چنددقیقه‌ای پیچ شل می‌شد.
۲- سیستم در هنگام چرخش موتور در بالاترین دور، ارتعاشات شدیدی داشت و ثابت نمی‌ایستاد.
۳- انکودر ثابت که به‌وسیله تسمه تایم و پولی به شفت دوار موتور متصل بود، با چسب به بدنه چسبانده شده بود. این موضوع باعث می‌شد امکان رگلاژ تسمه تایم و تنظیم کشش آن وجود نداشته باشد.
۴- ازآنجاکه انکودرهای استفاده‌شده، از نوع افزایشی[2] هستند، بعد از مدتی کار کردن از حالت کالیبره خارج می‌شدند و خطای اندازه‌گیری قابل‌توجهی به وجود می‌آمد.
۵- عمر باتری لیتیم-یون متصل به مدار متحرک بازو، با توجه به مصرف مدار، کمتر از ۲ ساعت بود و سیستم از پس فرایندهای آموزش طولانی‌مدت برنمی‌آمد.
۶- جهت پیاده‌سازی مدارهای الکتریکی از برِد بورد استفاده‌شده بود و شل بودن اتصالات، امکان بروز انواع خرابی و قطعی در مدار در هنگام کار را تشدید می‌کرد. (علی‌الخصوص در مدار متحرک متصل روی بازو)
۷- شفت موتور، به علت کارکرد زیاد، به میزان جزئی از حالت عمودی انحراف داشت.

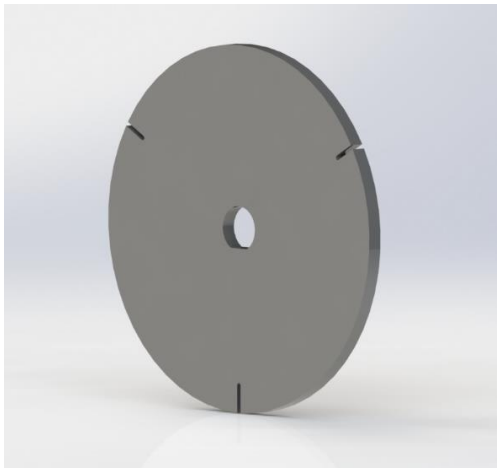
جهت آماده کردن سازه‌ی موجود برای اجرای الگوریتم‌های کنترل هوشمند، لازم بود تا نواقص فوق تا حد امکان اصلاح شوند. در ادامه، به توضیح مختصر اصلاحات صورت گرفته روی سازه‌ی آونگ می‌پردازیم.

۳-۱- طراحی سیستم کالیبراسیون برای انکودرها

در سازه‌ی پیشین از انکودرهای افزایشی[3] جهت سنجش موقعیت زاویه‌ای موتور و آونگ در هر لحظه استفاده می‌شد. علت استفاده از این نوع انکودر به جای انکودر مطلق[4]، قیمت بسیار پایین‌تر آن بود. در طرف مقابل، انکودرهای افزایشی، به دلیل نداشتن مبداء مشخص، بعد از مدت زمان طولانی کار کردن از حالت کالیبره خارج می‌شوند و اعداد اشتباهی را نشان می‌دهند. این موضوع علی‌الخصوص در هنگام خواندن زاویه‌ی آونگ بسیار مشکل‌ساز می‌شود زیرا حتی اختلاف اندک زاویه‌ی قرائت شده برای آونگ از حالت مقدار واقعی آن می‌تواند موجب ناپایداری سیستم و سقوط آونگ شود.

[1] Reinforcement Learning
[2] Incremental Encoder
[3] Incremental Encoder
[4] Absolute Encoder

جهت رفع این مشکل یک سیستم کالیبراسیون برای انکودر نصب شده روی بازو (برای اندازه‌گیری زاویه‌ی آونگ) طراحی شد. در این سیستم، از یک اپتوکانتر[1] مدل GK152 (حاوی یک فرستنده و گیرنده‌ی مادون‌قرمز) و یک دیسک استفاده می‌شود. دیسک دوار به همراه شفت انکودر و آونگ دوران می‌کند و بخشی از آن همواره بین فرستنده و گیرنده‌ی مادون‌قرمز قرار دارد. در سه‌نقطه از مسیر دیسک مقابل حس‌گر، سوراخ‌هایی ایجادشده است و در هنگام قرار گرفتن این نقاط در بین فرستنده و گیرنده‌ی مادون‌قرمز، نور فرستنده توسط گیرنده دریافت و انکودر کالیبره می‌شود.

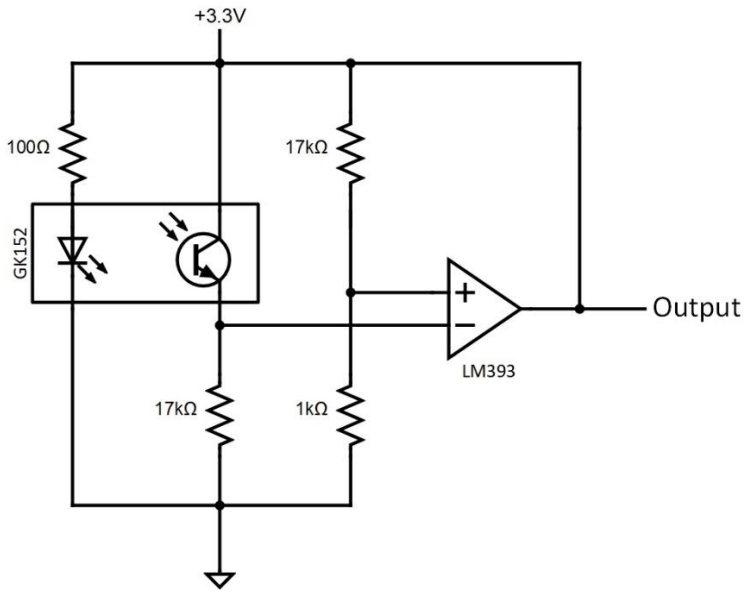


شکل ۶: دیسک دوار متصل به انکودر

شکل ۷: گیرنده و فرستنده مادون‌قرمز مدل *GK152*

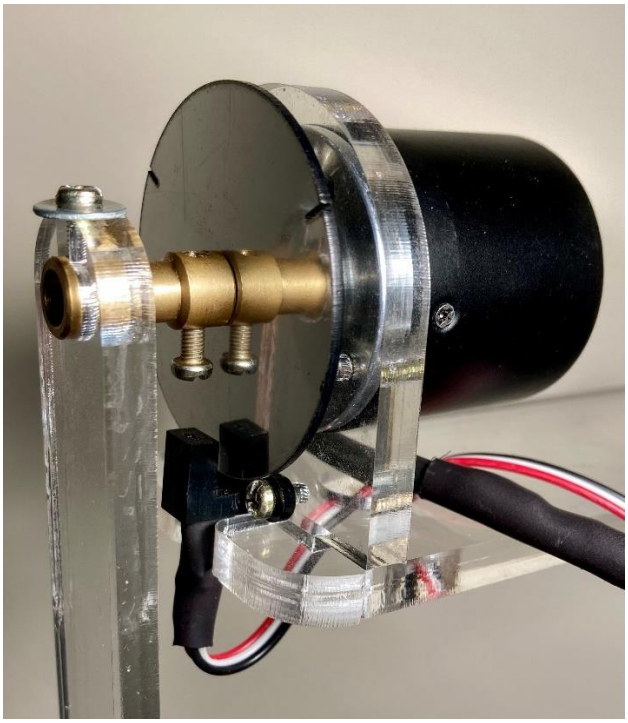
همچنین، از یک تقویت‌کننده‌ی عملیاتی[2] مقایسه‌گر جهت تبدیل خروجی‌های گیرنده‌ی مادون‌قرمز به مقادیر باینری[3] استفاده شد. در این مدار، ترانزیستور گیرنده‌ی مادون‌قرمز به‌صورتی در مدار قرار می‌گیرد که بسته به دریافت یا عدم دریافت نور فرستنده، در ناحیه اشباع یا خاموش باشد. درنهایت، ولتاژ پایه امیتر[4] به کمک یک تقویت‌کننده عملیاتی مقایسه‌گر (LM393) با یک مقدار مرجع مقایسه می‌شود و مقادیر باینری در خروجی تقویت‌کننده عملیاتی تولید می‌شوند. شکل زیر، شماتیک مدار را نشان می‌دهد.

[1] Opto Counter
[2] Operational Amplifier
[3] Binary
[4] Emitter



شکل ۸: مدار تبدیل خروجی اپتوکانتر به مقادیر باینری

شکل زیر، نمایی از مکانیزم کالیبراسیون انکودر متصل به آونگ را نشان می‌دهد.



شکل ۹: مکانیزم کالیبراسیون انکودر متصل به آونگ

۳-۲- تغییر میکروکنترلرها

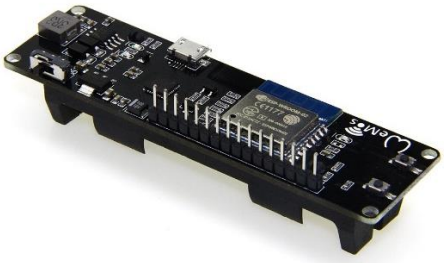
در سازه‌ی پیشین از دو بورد آردوینو[1] بر روی بازو و پایه‌ی سازه جهت پیاده‌سازی الگوریتم‌های کنترلی استفاده می‌شد و این دو برد به کمک NRF باهم تبادل اطلاعات می‌کردند. مدار موجود روی بازوی سازه به کمک یک باتری لیتیم و بورد نصب شده روی پایه از USB لپ‌تاپ تغذیه می‌شد. ساختار مذکور دو ایراد برجسته داشت:

۱- این ساختار برای پیاده‌سازی کنترلرهایی با حجم پردازش پایین مانند کنترلر PID بسیار مناسب است اما توان پردازشی میکروکنترلرهای آردوینو بسیار کمتر از مقدار موردنیاز برای پیاده‌سازی کنترلرهای هوشمند مبتنی بر الگوریتم‌های هوش مصنوعی می‌باشد.

۲- NRF و مدار شارژ باتری لیتیم به‌صورت ماژول‌های جداگانه به آردوینو متصل می‌شدند که باعث افزایش تعداد سیم‌کشی در مدار بود.

لذا، آردوینوها با دو ماژول Womos D1 با هسته ESP8266 جایگزین شدند. ماژول نصب شده روی بازوی سازه، به یک مدار شارژ باتری لیتیم مجهز است و نیاز به مدار شارژ مجزا را برطرف می‌سازد.

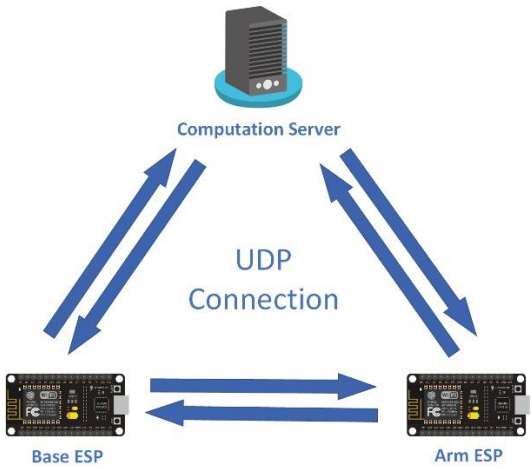
شکل ۱۰: ماژول *Womos D1* دارای هسته *ESP8266* جهت استفاده روی پایه سازه (*Base ESP*)



شکل ۱۱: ماژول *Womos D1* دارای هسته *ESP8266* و جا باتری و شارژر باتری لیتیم-یون ۱۸۶۵۰ جهت استفاده روی بازوی سازه (*Arm ESP*)

این دو بورد، علاوه بر برخورداری از قیمتی پایین به نسبت سایر مدل‌های مشابهشان (ازجمله مدل‌های مجهز به بلوتوث[2])، به کمک WiFi به یکدیگر و به یک سرور محاسباتی روی کامپیوتر متصل می‌شوند. اتصال به سرور محاسباتی امکان اجرای الگوریتم‌هایی با نیاز به توان محاسباتی بالاتر را مهیا می‌کنند. همچنین، در الگوریتم‌های با توان محاسباتی پایین به‌طور مستقیم روی خود ماژول‌ها قابل‌اجرا هستند.

[1] Arduino
[2] Bluetooth



شکل ۱۲: ارتباط بین بوردهای *ESP* و سرور محاسباتی

۳-۳- طراحی بورد مدار چاپی [۱]

مدارات الکتریکی سازه‌ی پیشین، بر روی برِد بورد[۲] پیاده‌سازی شده بودند و برطرف کردن مشکلات احتمالی در فرایند آموزش (مانند قطعی‌های احتمالی در مدار) به علت پیچیدگی سیم‌کشی، بسیار سخت و زمان‌بر بود. جهت بهبود عملکرد مدارات الکتریکی در هنگام فرایندهای آموزش طولانی‌مدت، دو بورد مدار چاپی مجزا برای سازه طراحی و ساخته شد. در ادامه، به بررسی این مدارها می‌پردازیم:

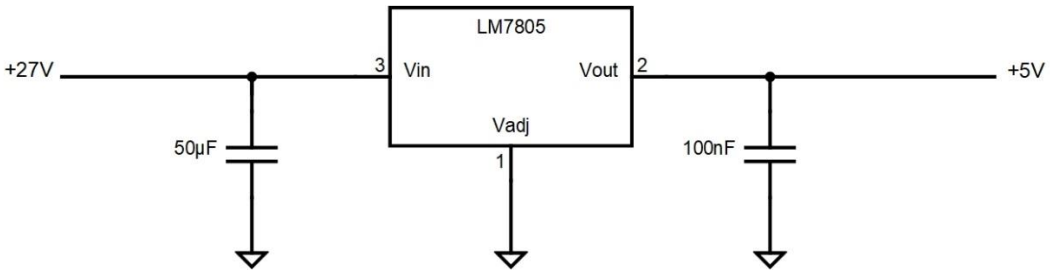
۳-۳-۱- بورد نصب‌شده روی پایه

مدار قرارگرفته روی پایه‌ی سازه، دارای بخش‌های زیر است:

۱- پورت اتصال به درایور موتور[۳] BTS7960
۲- پورت اتصال Womos D1
۳- پورت اتصال انکودر
۴- پورت اتصال اپتوکانتر[۴] به‌عنوان مرجع انکودر
۵- پورت تغذیه مدار
۶- مدار تغذیه Womos D1
۷- مدار تبدیل خروجی اپتوکانتر به مقادیر باینری
۸- Debugger LED

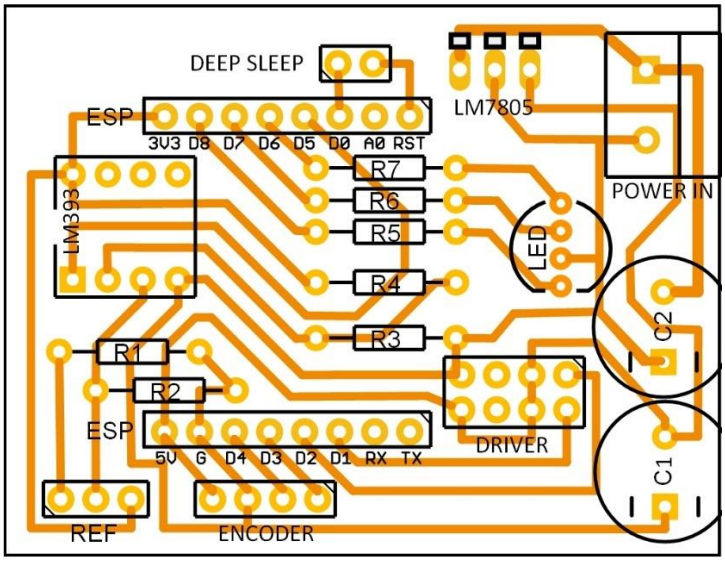
[1] Printed Circuit Board (PCB)
[2] Bread Board
[3] Motor Driver
[4] Opto Counter

این مدار، از برق ۲۷ ولت دی‌سی خروجی پاور سوییچ[1] تغذیه می‌شود. سطح ولتاژ ۲۷ ولت به کمک یک رگلاتور ولتاژ[2] مدل 7805 به ۵ ولت کاهش می‌یابد که برای اجزای مدار قابل‌استفاده است.



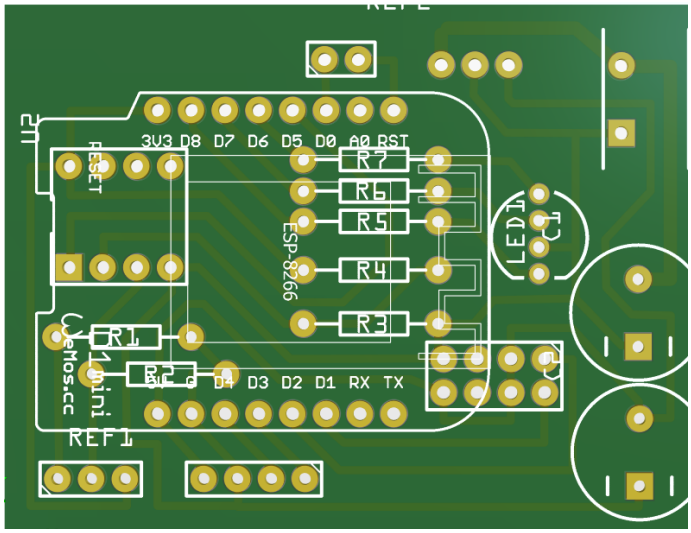
شکل ۱۳: مدار کاهنده ولتاژ برای *PCB* نصب شده روی پایه سازه

شماتیک مدار چاپی طراحی‌شده و تصویر سه‌بعدی آن در نرم‌افزار Altium Designer در دو شکل بعدی آمده است:



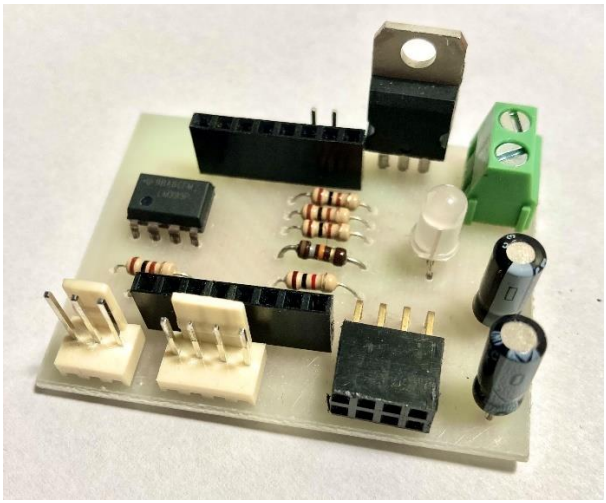
شکل ۱۴: شماتیک مدار چاپی نصب شده بر روی پایه سازه

[1] Power Switch
[2] Voltage Regulator

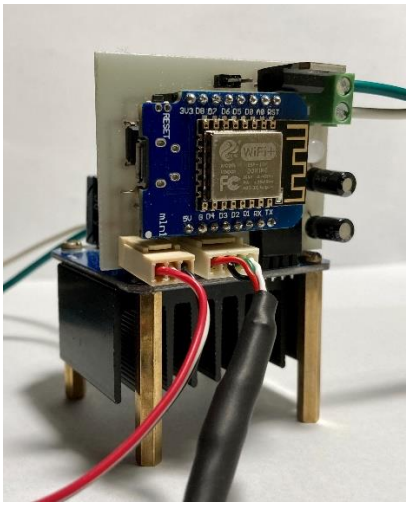


شکل ۱۵: تصویر سه‌بعدی مدار چاپی نصب شده روی پایه در نرم‌افزار *Altium Designer*

همچنین، مدار چاپی نصب شده روی پایه در حالت جدا از سیستم و متصل به آن در ادامه آورده شده است.



شکل ۱۶: مدار چاپی متصل به پایه جدا از سیستم



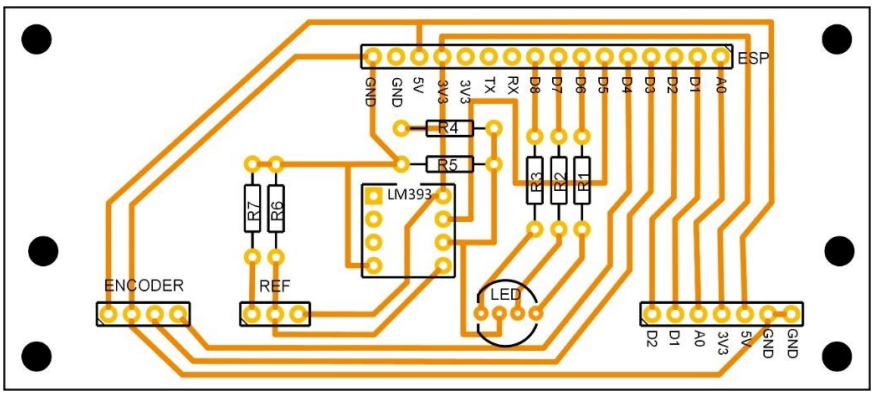
شکل ۱۷: : مدار چاپی متصل به پایه متصل

۳-۳-۲- بورد نصب‌شده روی بازو

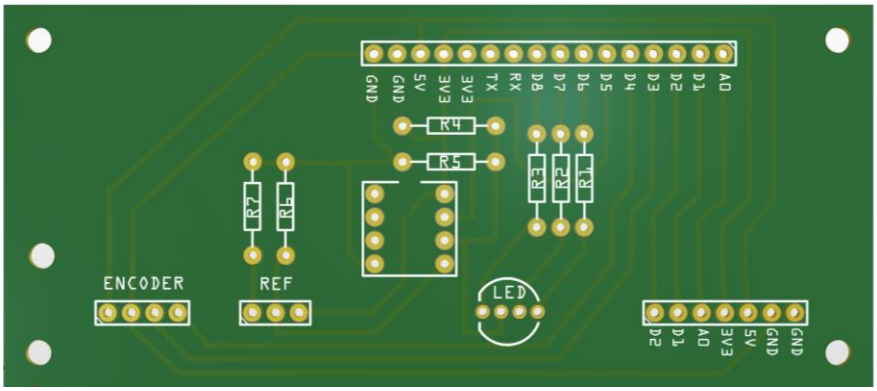
مدار قرارگرفته روی بازوی سازه، دارای بخش‌های زیر است:

۱- پورت اتصال Womos D1
۲- پورت اتصال انکودر
۳- پورت اتصال اپتوکانتر به‌عنوان مرجع انکودر
۴- مدار تبدیل خروجی اپتوکانتر به مقادیر باینری
۵- Debugger LED

شماتیک مدار چاپی طراحی‌شده و تصویر سه‌بعدی آن در نرم‌افزار Altium Designer در دو شکل بعدی آمده است:

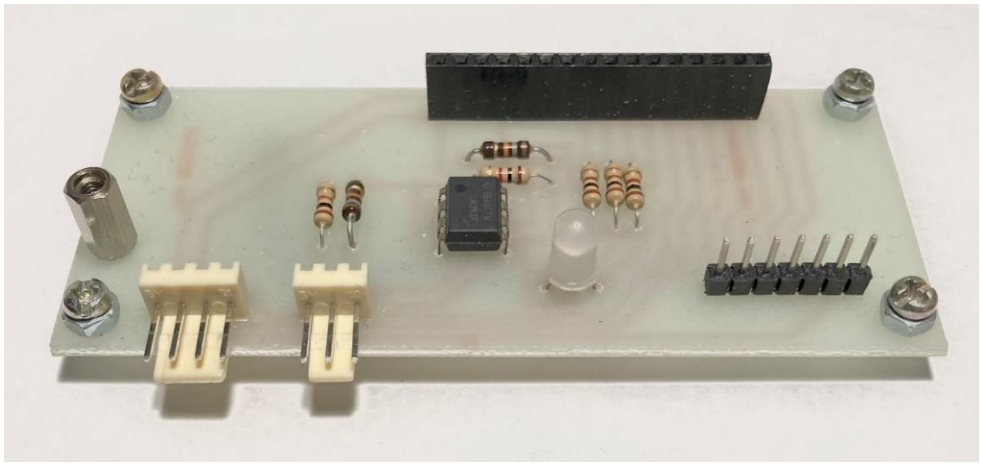


شکل ۱۸: شماتیک مدار چاپی نصب شده روی بازوی سازه

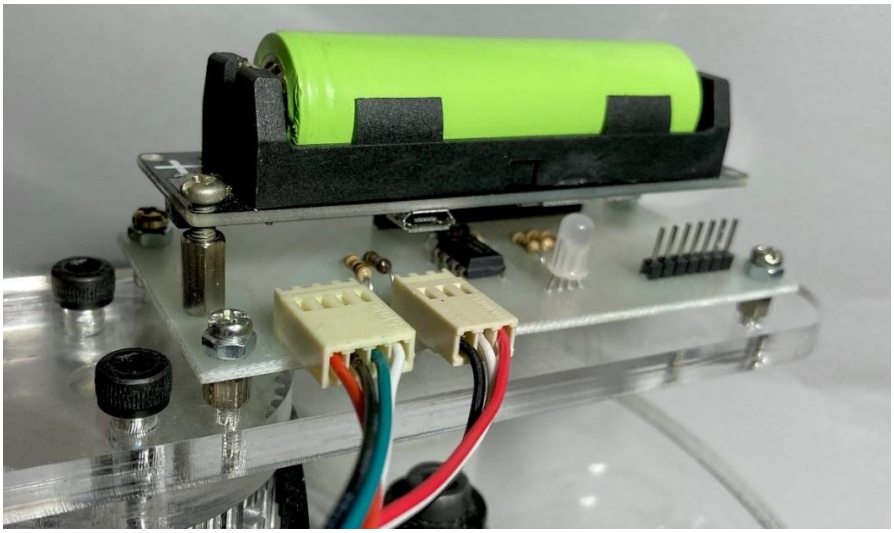


شکل ۱۹: تصویر سه‌بعدی مدار چاپی نصب شده روی بازو در نرم‌افزار *Altium Designer*

همچنین، مدار چاپی نصب شده روی بازو در حالت جدا از سیستم و متصل به آن در ادامه آورده شده است.



شکل ۱۸: مدار چاپی متصل به بازو در حالت جدای از سیستم

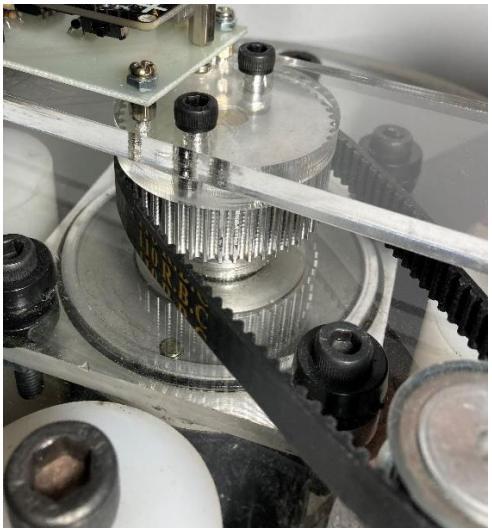


شکل ۱۷: مدار چاپی متصل به بازو در حالت نصب شده روی سیستم

۳-۴- تغییر شیوه‌ی اتصال بازو به پایه

در سازه‌ی آونگ معکوس دورانی پیشین، شفت موتور تنها با استفاده از یک پیچ به‌موازات آن، به بازوی سازه متصل شده بود. بدون توجه به میزان سفتی پیچ، اتصال بین شفت موتور و بازو بعد از مدتی کارکرد شل می‌شد و لازم بود تا پیچ دوباره سفت شود. برای حال این مشکل، اتصال بین آونگ و موتور با یک اتصال فلنجی جایگزین شد. بدین منظور، یک پولی سری تراش‌کاری و دو رزوه در طرفین محل قرارگیری شفت موتور در آن ایجاد شد. بر روی بازوی سازه نیز دو سوراخ تعبیه شد و به کمک دو پیچ، بازوی آونگ دورانی معکوس به پولی متصل می‌شود. اتصال پولی به موتور نیز از طریق دو پیچ مغزی صورت می‌گیرد. اشکال زیر، نمایی از پولی تراش‌کاری شده و شیوه‌ی قرارگیری آن روی سازه است:

شکل ۲۰: پولی متصل به شفت موتور و بازوی سازه‌ی آونگ معکوس دورانی



شکل ۲۱: اتصال بازو به موتور در سازه‌ی آونگ دورانی معکوس

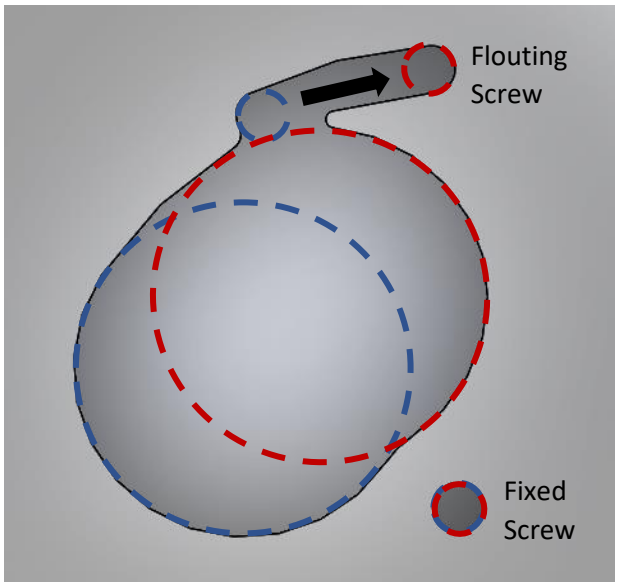
نقشه‌ی کارگاهی پولی در ضمیمه ۱ آورده شده است.

۳-۵- طراحی مکانیزم رگلاژ برای تسمه تایم متصل به موتور

جهت اندازه‌گیری مکان زاویه‌ای موتور در هرلحظه، از یک انکودر متصل به پایه سازه استفاده می‌شود. این انکودر به‌وسیله‌ی تسمه تایم و پولی به موتور متصل می‌شود. در سازه‌ی آونگ معکوس دورانی پیشین، انکودر به‌وسیله‌ی چسب به پایه‌ی سازه چسبانده شده بود. همچنین، به علت ایجاد امکان تعویض و جایگذاری مجدد تسمه، فاصله‌ی مرکز تا مرکز شفت انکودر تا شفت موتور کمتر از حد استاندارد بود و این موضوع باعث لقی بیش‌ازاندازه‌ی تسمه و ایجاد خطا در اندازه‌گیری می‌شد. به جهت رفع این مشکل، یک مکانیزم رگلاژ ساده برای تغییر فاصله‌ی مرکز شفت‌های انکودر و موتور طراحی شد. این مکانیزم، به انکودر و شفت انکودر به میزان حدوداً ۱ سانتی‌متر اجازه‌ی جابه‌جایی می‌دهد.

تصویر زیر، نمایی از مکانیزم رگلاژ طراحی‌شده در نرم‌افزار سالیدورکس[1] می‌باشد:

شکل ۲۲: نمایی از مکانیزم رگلاژ تسمه تایم

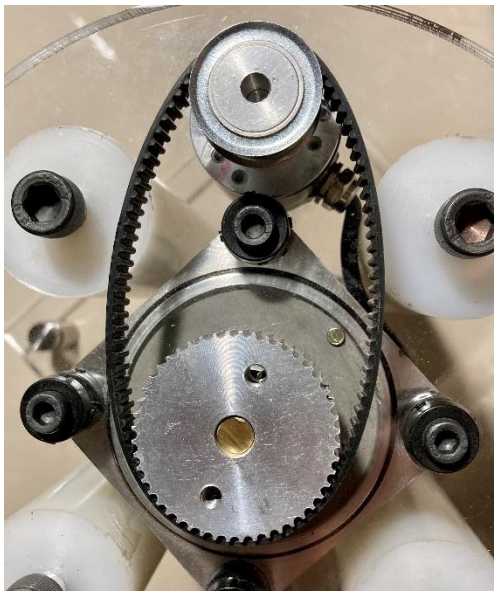


شکل ۲۳: مکانیزم تغییر مکان انکودر جهت رگلاژ تسمه تایم

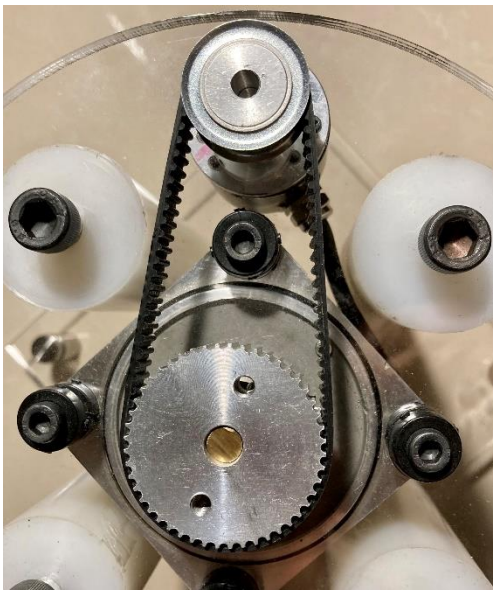
نقشه‌ی این مکانیزم رگلاژ تسمه تایم در ضمیمه ۱ آورده شده است.

[1] Solid Works

دو تصویر زیر از تسمه تایم در حالت آزاد و سفت هستند:



شکل ۲۴: تسمه تایم متصل‌کننده شفت‌های موتور و انکودر حالت آزاد



شکل ۲۵: تسمه تایم متصل‌کننده شفت‌های موتور و انکودر حالت سفت

۳-۶- لرزش سازه در هنگام کار موتور در سرعت‌های بالا

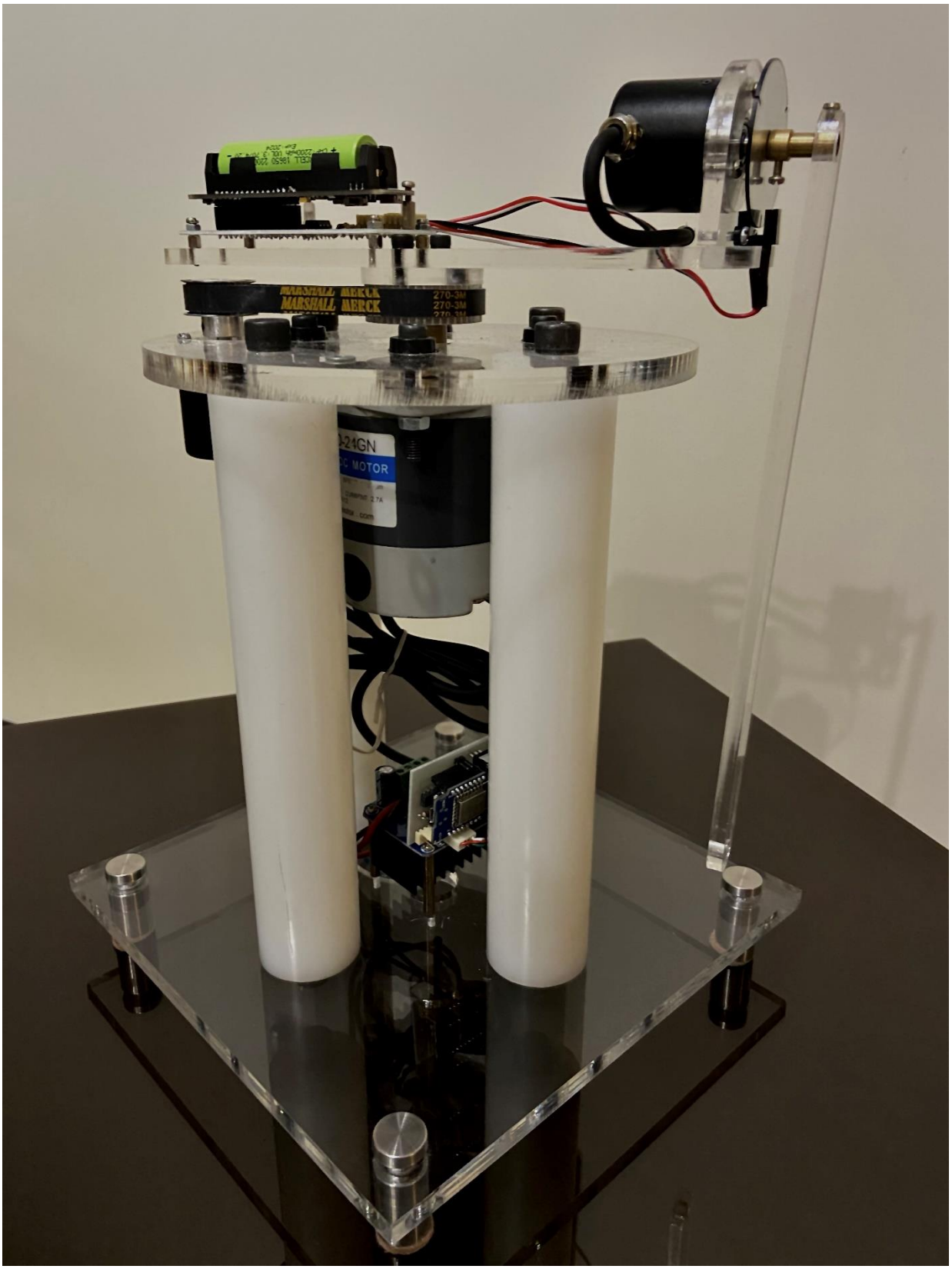
این مشکل عمدتاً ناشی از سه علت بود:

۱- وجود نابالانسی در سازه‌ی بازو و آونگ متصل به موتور
۲- عدم وجود پایه مناسب برای دستگاه
۳- شیوه‌ی اتصال موتور به سازه
۴- به وجود نابالانسی جزئی در موتور

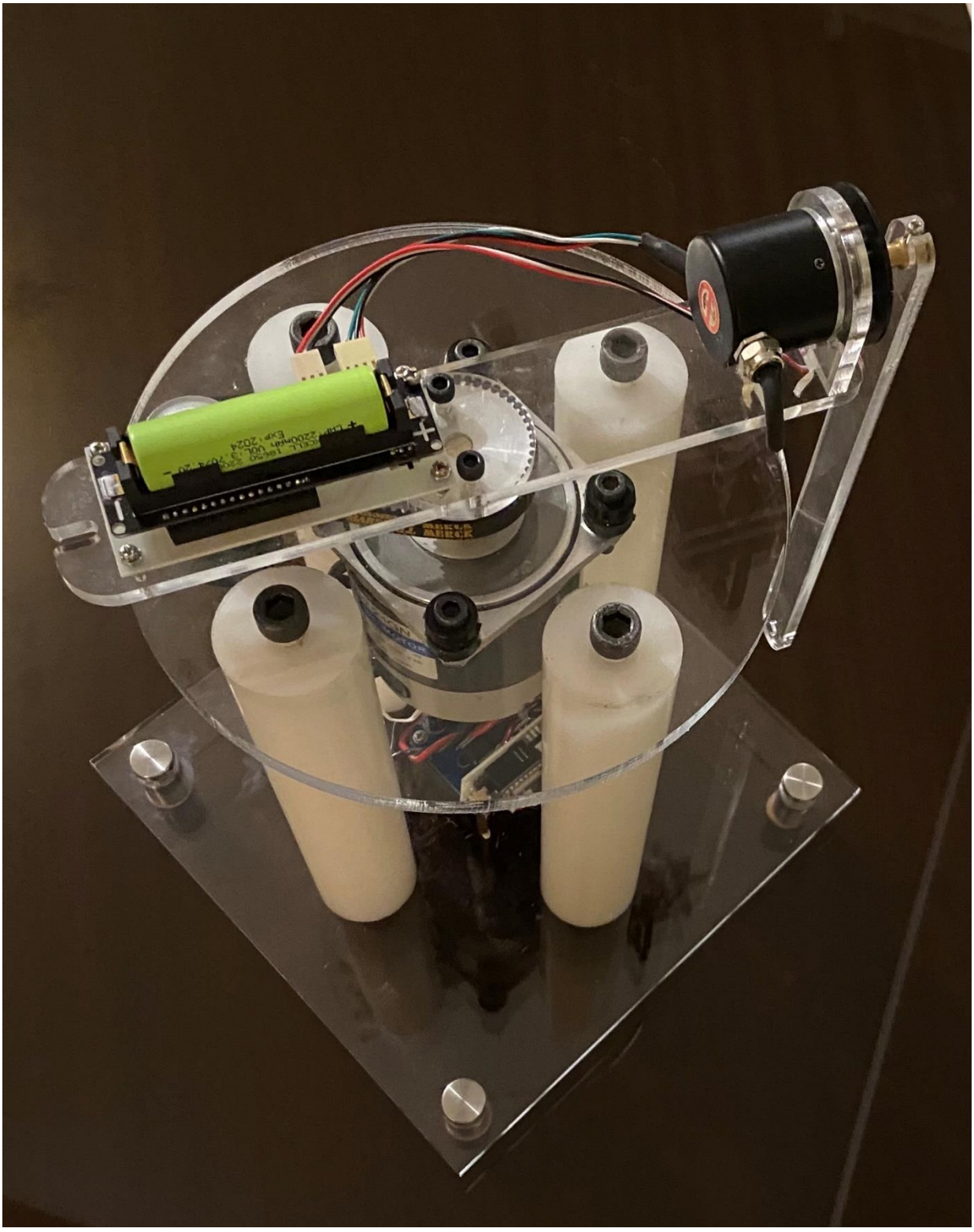
جهت حل مشکل اول، یک محل در انتهای آونگ جهت اضافه کردن وزنه برای بالانس کردن آونگ تعبیه شد. اگرچه به علت چرخش دائمی آونگ رسیدن به بالانس کامل برای سازه‌ی آونگ و بازوی در حال دوران ممکن نیست، ازآنجاکه جرم آونگ نسبت به سایر بخش‌ها به‌مراتب کمتر است، امکان رسیدن به بالانس حدودی حول محور دوان وجود دارد. مشکلات دوم و سوم به ترتیب با اضافه کردن پایه‌ی مناسب به سازه و واشرهای پلاستیکی ضخیم به‌عنوان دمپر در محل اتصال موتور به بدنه تا حدودی رفع شدند. اما متأسفانه ایراد چهارم همچنان در سازه پابرجا است. همچنین، به علت کهنه‌گی موتور و بریده شدن بخشی از شفت آن، شفت اندکی از حالت قائم خارج‌شده است. بنابراین، برای رسیدن به بهترین بازده دستگاه به نظر می‌رسد بهترین گزینه تعویض موتور با یک موتور جدید است.

با توجه به اعمال تغییرات مذکور در سازه، لازم بود تا آونگ، بازو و نگه‌دارنده‌ی انکودر متصل به بازو دوباره طراحی و ساخته شوند. نقشه‌ی کارگاهی این بخش‌ها در ضمیمه‌ی اول آورده شده است.

دو تصویر از سازه‌ی کامل آونگ معکوس دورانی اصلاح شده در ادامه آورده شده است.



شکل ۲۶: سازه‌ی آونگ معکوس دورانی - نمای روبرو



شکل ۲۷: سازه‌ی آونگ معکوس دورانی – نمای روبرو

۴- محیط شبیه‌سازی[1]

در حین اجرای اصلاحات ذکرشده در بخش ۳ بر روی سازه، سعی شد تا هم‌زمان عملکرد الگوریتم‌های یادگیری تقویتی پیوسته در محیط شبیه‌سازی‌شده در کامپیوتر سنجیده شود. هدف از این کار را به‌طور خلاصه می‌توان در سه مورد زیر بیان کرد:

۱- آشنا شدن با نحوه‌ی پیاده‌سازی الگوریتم‌ها
۲- یافتن بازه‌های معقول برای پارامترهای اساسی[2] در الگوریتم‌ها (مانند معماری شبکه‌ی مورداستفاده و پارامترهای مورداستفاده در آموزش آن ازجمله ضریب یادگیری[3])
۳- استفاده از مدل‌های آموزش داده‌شده به کمک شبیه‌ساز[4] برای کنترل سازه‌ی اصلی

به‌منظور شبیه‌سازی سیستم، در ابتدا یک شبیه‌ساز در زبان برنامه‌نویسی پایتون[5] از پایه نوشته شد. برای این کار، معادلات دیفرانسیل حاکم بر آونگ معکوس دورانی در حالت گسسته بازنویسی شدند و از آن‌ها برای به‌روزرسانی سینماتیک آونگ معکوس در شبیه‌ساز در زمان‌های گسسته استفاده شد. مهم‌ترین مزیت مدل نوشته‌شده در پایتون، سرعت بالای اجرای آن می‌باشد. در مقابل، با توجه به دینامیک نسبتاً پیچیده‌ی آونگ معکوس دورانی، از بسیاری از جزئیات در شبیه‌سازی اولیه صرف‌نظر شد. در گام دوم شبیه‌سازی، از بخش سیمولینک[6] نرم‌افزار متلب[7] برای ساخت یک مدل کامپیوتری دقیق‌تر از آونگ معکوس دورانی استفاده شد. این کار، امکان واردکردن ظریف‌ترین جزئیات در فرایند مدل‌سازی را نیز فراهم نمود اما در عوض، سرعت اجرای مدل پیاده‌سازی شده به نسبت مدل نوشته‌شده در پایتون به‌مراتب پایین‌تر است.

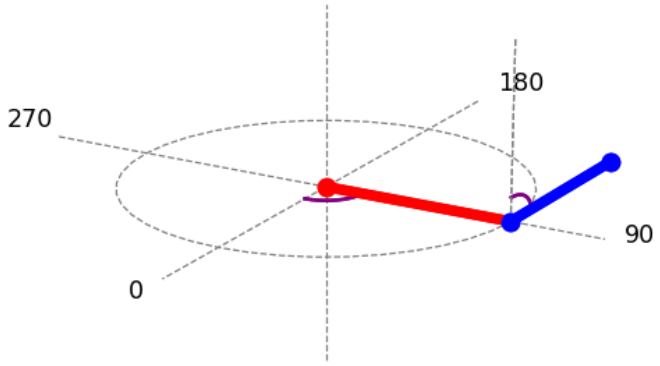
در ادامه، به توضیح مدل دقیق‌تر دو مدل پیاده‌سازی شده می‌پردازیم.

۴-۱- مدل‌سازی در زبان پایتون

این مدل‌سازی پیش از ساخته‌شدن سازه‌ی اولیه‌ی آونگ معکوس دورانی انجام شد. برای مدل‌سازی، آونگ معکوس دورانی به‌صورت یک مدل اجرای محدود[8] متشکل از یک موتور DC، یک بازوی[9] متصل به موتور و یک آونگ[10] در انتهای آن در نظر گرفته شد.

[1] Simulation Environment
[2] Hyperparameters
[3] Learning Rate
[4] Simulator
[5] Python
[6] Simulink
[7] Matlab
[8] Lumped
[9] arm
[10] Pendulum

شکل زیر، نمایی از پنجره گرافیکی نوشته‌شده به زبان پایتون جهت نمایش وضعیت لحظه‌ای آونگ معکوس دورانی است:



شکل ۲۸: پنجره گرافیکی نوشته‌شده به زبان پایتون جهت نمایش وضعیت لحظه‌ای آونگ معکوس دورانی شبیه‌سازی‌شده

در ادامه، به بررسی جنبه‌های مختلف مدل می‌پردازیم.

۴-۱-۱- مفروضات مدل‌سازی

فرض‌های اساسی ساده کننده اعمال‌شده در این مدل، به شرح زیر می‌باشند:

۱- از وجود تمام اجزای سازه‌ی آونگ معکوس دورانی به‌جز آونگ و بازو صرف‌نظر می‌شود.
۲- هر دوی بازو و آونگ به‌صورت دو میله با مقطع دایره‌ای در نظر گرفته می‌شوند.
۳- توزیع جرم بازو و آونگ در راستای طولشان یکنواخت است.
۴- اصطکاک در محل اتصال آونگ به بازو از نوع اصطکاک ویسکوز[1] متناسب با سرعت چرخش مفصل در نظر گرفته می‌شود و از سایر انواع اصطکاک صرف‌نظر می‌شود.
۵- از تغییرات مشخصه‌های موتور با افزایش دما صرف‌نظر می‌شود.

۴-۱-۲- ورودی و خروجی مدل

ورودی و خروجی‌های مدل به شرح زیراند:

جدول ۳: ورودی‌ها و خروجی‌های مدل اولیه

ورودی مدل	
ولتاژ اعمالی به پایه‌های موتور	
خروجی‌های مدل	
مکان زاویه‌ای بازو	مکان زاویه‌ای آونگ
سرعت زاویه‌ای بازو	سرعت زاویه‌ای آونگ
شتاب زاویه‌ای بازو	شتاب زاویه‌ای آونگ

[1] Viscous Friction

۴-۱-۳- پارامترهای مدل‌سازی

به‌این‌ترتیب، پارامترهای دخیل در مدل‌سازی به شرح موجود در جدول زیر می‌باشند:

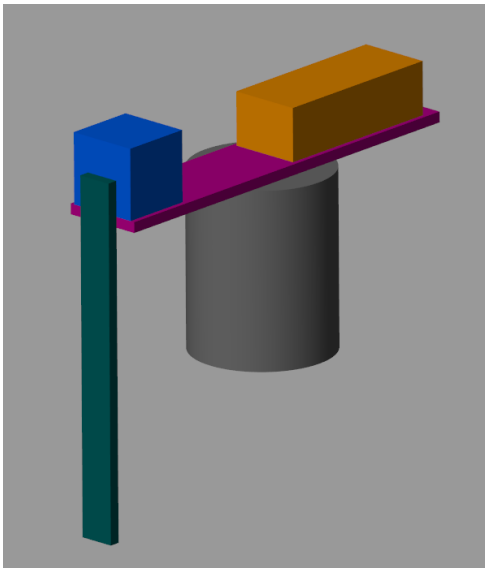
جدول ۴: پارامترهای دخیل در مدل‌سازی اولیه

پارامترهای دخیل در مدل‌سازی اولیه	
پارامترهای بازو	چگالی / طول / شعاع مقطع
پارامترهای آونگ	چگالی / طول / شعاع مقطع
مفصل بین آونگ و بازو	ضریب اصطکاک ویسکوز
پارامترهای موتور	مقاومت موتور / اندوکدانس موتور / ضریب Back emf موتور / اینرسی روتور / ضریب میرایی روتور / ضریب گشتاور موتور

در هنگام مدل‌سازی، به علت در دست نبودن سازه فیزیکی، پارامترهای فوق با مقادیر منطقی جایگزین شدند.

۴-۲- مدل‌سازی در سیمولینک[1]

در هنگام نوشتن کد مدل اولیه در پایتون، ساخت سازه آغاز نشده بود و جزئیات طراحی نیز به‌طور کامل مشخص نبود. پس از مشخص شدن جزئیات طراحی سازه، لزوم اعمال این جزئیات در مدل‌سازی برای رسیدن به اهداف مدل‌سازی احساس می‌شد. اما پیاده‌سازی تمامی جزئیات از پایه به کمک پایتون امری بسیار دشوار و طاقت‌فرسا بود و به همین دلیل، یک مدل دیگر به کمک بخش سیم‌اسکیپ[2] در سیمولینک پیاده‌سازی شد. اگرچه که سرعت اجرای مدل ثانویه به‌مراتب از مدل اولیه پایین‌تر است، در آن جزئیات سازه نیز مدل شده‌اند.



شکل ۲۹: پنجره گرافیکی سمولینک جهت نمایش وضعیت لحظه‌ای آونگ معکوس دورانی شبیه‌سازی‌شده

[1] Simulink
[2] Simscape

در ادامه، به بررسی جنبه‌های مختلف این مدل می‌پردازیم.

۴-۲-۱- مفروضات مدل‌سازی

چند فرض اعمال‌شده در مدل ثانویه، به شرح زیر می‌باشند:

۱- از جرم پیچ‌های موجود در سازه به علت کوچک بودن صرف‌نظر می‌شود.
از جرم مکانیزم کالیبراسیون انکودر متصل به بازو به علت کوچک بودن صرف‌نظر می‌شود
۲- انکودر و برد چاپی متحرک نصب شده روی بازوی سازه به‌عنوان اجرام متمرکز در نظر گرفته می‌شود.
۳- اصطکاک در محل اتصال آونگ به بازو از نوع اصطکاک ویسکوز[1] متناسب با سرعت چرخش مفصل در نظر گرفته می‌شود و از سایر انواع اصطکاک صرف‌نظر می‌شود.
۴- از تغییرات مشخصه‌های موتور با افزایش دما صرف‌نظر می‌شود.

۴-۲-۲- ورودی و خروجی‌های مدل

ورودی مدل نوشته‌شده در سیمولینک دوره‌ی کاری[2] سیگنال PWM[3] ورودی به درایور موتور و خروجی آن، یک سری زمانی از خروجی زمان حال سیستم به همراه تعداد دلخواهی از خروجی‌های آن در زمان‌های گذشته است. به‌وسیله‌ی این مقادیر و با در دست داشتن زمان نمونه‌برداری[4] از مدل می‌توان سرعت و شتاب زاویه‌ای آونگ و بازو را محاسبه نمود.

۴-۲-۳- پارامترهای مدل‌سازی

به‌این‌ترتیب، پارامترهای دخیل در مدل‌سازی به شرح موجود در جدول زیر می‌باشند:

جدول ۵: پارامترهای دخیل در مدل‌سازی ثانویه

پارامترهای دخیل در مدل‌سازی دقیق	
پارامترهای بازو	چگالی / طول / عرض و ضخامت مقطع مستطیلی
پارامترهای آونگ	چگالی / طول / عرض و ضخامت مقطع مستطیلی
پارامترهای انکودر	جرم / فاصله مرکز جرم از مرکز دوران / ضریب اصطکاک ویسکوز شفت
پارامترهای مدار چاپی	جرم / فاصله مرکز جرم از مرکز دوران
پارامترهای موتور	مقاومت موتور / اندوکدانس موتور / ضریب Back emf موتور / اینرسی روتور / ضریب میرایی روتور / ضریب گشتاور موتور

[1] Viscous Friction
[2] Duty Cycle
[3] Pulse-Width Modulation
[4] Sample Time

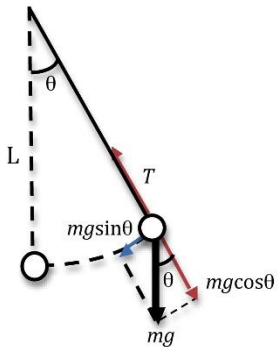
از بین پارامترهای ذکرشده:

- طول، عرض و ضخامت بازو و آونگ جزو پارامترهای طراحی سازه و معلوم هستند.
- جنس بازو و آونگ از پلکسی‌گلس است و چگالی بازو و آونگ مشخص می‌باشد.
- جرم انکودر و برد چاپی به کمک ترازو دقیق اندازه‌گیری شد.
- مراکز جرم انکودر و مدار چاپی برابر مراکز حجم آن‌ها تخمین زده شد و به‌این‌ترتیب، فاصله مراکز جرم انکودر و مدار چاپی از مرکز دوران، به دست آمد.

حال، به بررسی روش‌های به دست آوردن سایر پارامترها می‌پردازیم.

۴-۲-۲-۱- به‌دست آوردن ضریب اصطکاک ویسکوز شفت

برای به‌دست آوردن این ضریب، در ابتدا معادله حاکم بر نوسان آونگ را می‌نویسم. با نوشتن معادله گشتاور حول نقطه‌ی دوران برای آونگ نشان داده‌شده در شکل روبرو خواهیم داشت:



شکل ۳۰: آونگ ساده

$$I\ddot{\theta} + c\dot{\theta} = mglsin\theta \quad (Equation\ 23)$$

در معادله‌ی فوق، c ضریب اصطکاک آونگ با مفصل و I اینرسی آونگ حول مرکز دوران است.

همان‌گونه که مشاهده می‌شود، معادله‌ی به‌دست‌آمده برای نوسان آونگ یک معادله دیفرانسیلی غیرخطی است و حل آن از طریق روابط مرسوم ممکن نیست. اما، در نزدیکی نقطه تعادل آونگ ($\theta \approx 0$) طبق سری مک‌لورن[1] برای تابع سینوس، هم‌عرضی $sin\theta = \theta$ برقرار است و رابطه‌ی به‌دست‌آمده را می‌توان به‌صورت زیر خطی سازی کرد:

$$I\ddot{\theta} + c\dot{\theta} = mgl\theta \qquad (Equation\ 24)$$

معادله‌ی جدید، یک معادله دیفرانسیل مرتبه دو است و پاسخ آن به شرح زیر است:

$$\theta = c_1 e^{-\frac{\sqrt{c^2+4Imgl}+c}{2I}x} + c_2 e^{\frac{\sqrt{c^2+4Imgl}-c}{2I}x} \qquad (Equation\ 25)$$

[1] Maclaurin Series

در حل فوق، مقادیر c_1 و c_2 از شرایط مرزی برای مکان و سرعت زاویه‌ای آونگ به‌دست می‌آیند. اگر آونگ از زاویه اولیه‌ی θ_0 با مقدار نزدیک به صفر و بدون سرعت اولیه رها شود، می‌توان ضرایب c_1 و c_2 را به‌صورت زیر محاسبه نمود:

$$c_1 = \frac{\sqrt{c^2 + 4Imgl} - c}{2\sqrt{c^2 + 4Imgl}} \quad , \quad c_2 = \frac{\sqrt{c^2 + 4Imgl} + c}{2\sqrt{c^2 + 4Imgl}}$$

با جاگذاری در معادله اصلی خواهیم داشت:

$$\theta = \frac{\sqrt{c^2 + 4Imgl} - c}{2\sqrt{c^2 + 4Imgl}}\theta_0 e^{-\frac{\sqrt{c^2+4Imgl}+c}{2I}x} + \frac{\sqrt{c^2 + 4Imgl} + c}{2\sqrt{c^2 + 4Imgl}}\theta_0 e^{\frac{\sqrt{c^2+4Imgl}-c}{2I}x} \qquad (Equation\ 25)$$

برای اندازه‌گیری ضریب اصطکاک ویسکوز در شفت انکودر (اتصال بین بازو و آونگ) از معادله‌ی فوق استفاده شد. برای آونگ معکوس دورانی ساخته‌شده، تمامی پارامترهای دخیل در معادله فوق، به‌جز مقدار c مشخص است. پس با رها کردن آونگ از یک زاویه‌ی θ_0 نزدیک به صفر، و مشاهده‌ی رفتار آونگ، می‌توان مقدار ضریب اصطکاک ویسکوز را محاسبه کرد.

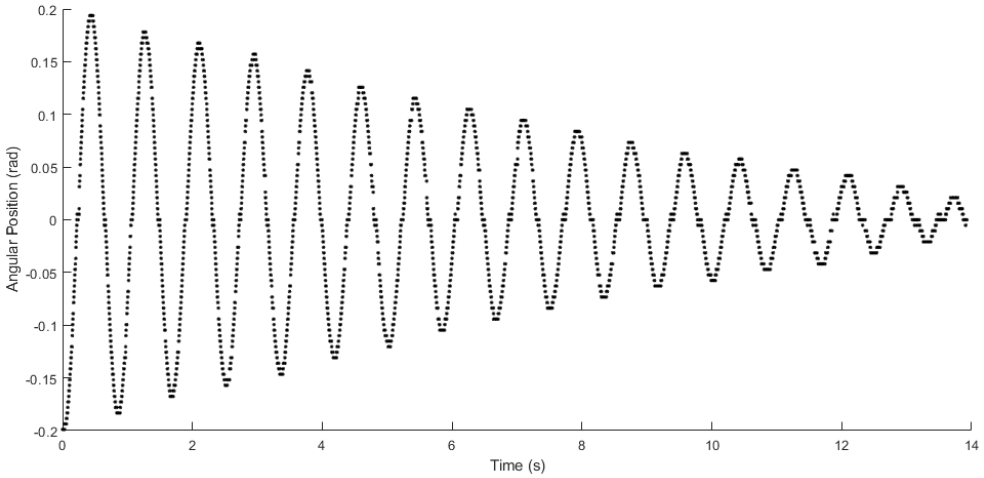
به‌این‌ترتیب، مراحل محاسبه‌ی ضریب اصطکاک ویسکوز به‌صورت زیر خواهد بود:

گام ۱: آونگ را از نقطه تعادل به‌اندازه‌ی کم منحرف می‌کنیم به‌گونه‌ای که فرض خطی ساز معادله دیفرانسیل آونگ ($sin\theta = \theta$) تقریباً برقرار شود.

گام ۲: نوسانات آونگ تا زمان سکون را به کمک انکودر اندازه‌گیری می‌کنیم.

گام ۳: به کمک برازش منحنی به فرم جواب به‌دست‌آمده، مقدار پارامتر c را تعیین می‌نماییم.

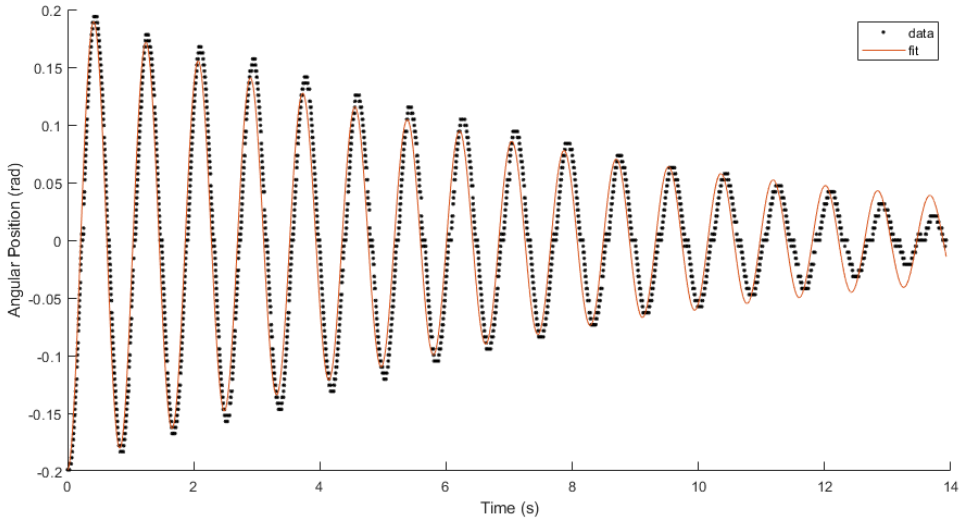
منحنی مکان زاویه‌ای آونگ از زمان رها شدن تا هنگام سکون در زیر آورده شده است:



شکل ۳۱: نمودار مکان زاویه‌ای برحسب زمان برای آونگ در هنگام رها شدن از زاویه θ_0

لازم به ذکر است که نقاط شکستگی موجود در نمودار در هنگام عبور از مبدأ، به علت کالیبره شدن انکودر توسط مکانیزم شرح داده‌شده است.

حال، با در دست داشتن پارامترهای آونگ و به کمک Tool Box برازش منحنی نرم‌افزار متلب یک منحنی به فرم جواب به‌دست‌آمده و با پارامتر c مجهول به داده‌های به‌دست‌آمده از آزمایش برازش می‌کنیم. منحنی برازش شده در ادامه آمده است.



شکل ۳۲: برازش منحنی نمودار مکان زاویه‌ای برحسب زمان برای آونگ در هنگام رها شدن از زاویه θ_0

همچنین، مقدار به‌دست‌آمده برای c از طریق برازش منحنی در زیر آورده شده است:

$$c = 0.000211\ N.m.s/rad\ (0.0002093, 0.0002127)\ 95\%\ confidence$$

۴-۲-۲-۲- به‌دست آوردن پارامترهای موتور (مکانیزم اندازه‌گیری مکان آن)

لازم است تا تابع تبدیل موتور DC را به دست آوردیم. برای این کار، ابتدا معادلات موتور از بخش ۲-۲ را بار دیگر بازنویسی می‌کنیم:

$$R_a i_a + L_a \frac{di_a}{dt} + V_b = V_a \qquad (Equation\ 19) \qquad V_b = K_b \frac{d\theta}{dt} \qquad (Equation\ 20)$$

$$I \frac{d^2\theta}{dt^2} + C \frac{d\theta}{dt} = T - T_L \quad V_b = K_b \frac{d\theta}{dt} \quad (Equation\ 21) \quad T = K_t i_a \qquad (Equation\ 22)$$

متأسفانه، باوجود جست‌وجوی فراوان، کاتالوگ موتور *DC* مورداستفاده یافت نشد. لذا لازم بود تا این پارامترهای دخیل در موتور از روی رفتار فیزیکی آن اندازه‌گیری شوند.

در طی این فرایند، انکودر به‌وسیله‌ی تسمه و پولی همواره به موتور *DC* متصل بود و این دو بخش سیستم، به‌عنوان یک بخش واحد در نظر گرفته شدند. این موضوع، موجب می‌شود که اینرسی پولی‌ها و تسمه و اصطکاک ویسکوز در انکودر نیز در محاسبات به‌طور خودکار در محاسبات وارد شوند.

در ابتدا، به کمک یک *RLC* متر، مقادیر مقاومت الکتریکی[1] بین پایه‌های موتور و اندوکدانس[2] اندازه‌گیری شد. مقادیر اندازه‌گیری شده به شرح زیر می‌باشند:

$$R_a = 2.5\ \Omega \quad , \quad L_a = 2.1\ mH$$

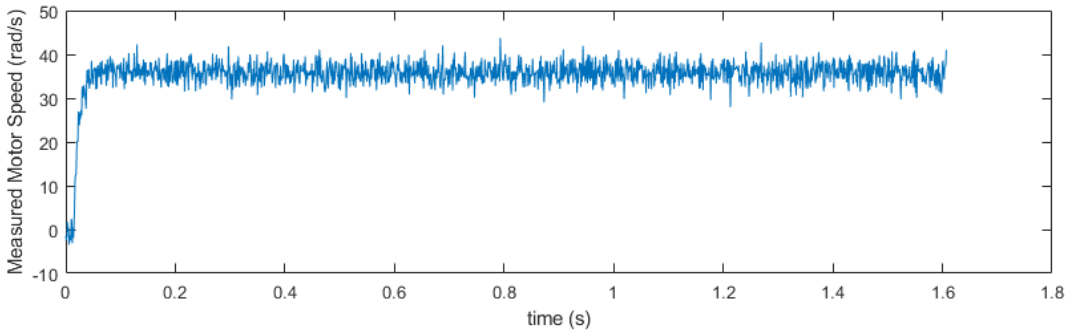
لازم به ذکر است که در فرایند مدل‌سازی از هرگونه تغییرات این مقادیر برحسب دما صرف‌نظر شد.

در ادامه‌ی کار، جهت به‌دست آوردن ضریب *Back emf* (K_b) مراحل زیر را دنبال می‌کنیم:

- **گام ۱:** شفت موتور را به سر دریل متصل می‌کنیم و آن را می‌چرخانیم.
- **گام ۲:** ولتاژ پایه‌های موتور (همان ولتاژ *Back emf*) را به کمک پین آنالوگ آردوینو[3] و سرعت دوران شفت موتور را به کمک انکودر برای مدتی اندازه‌گیری می‌کنیم.
- **گام ۳:** از معادله‌ی ۲، می‌توان K_b را به‌صورت زیر حساب کرد:

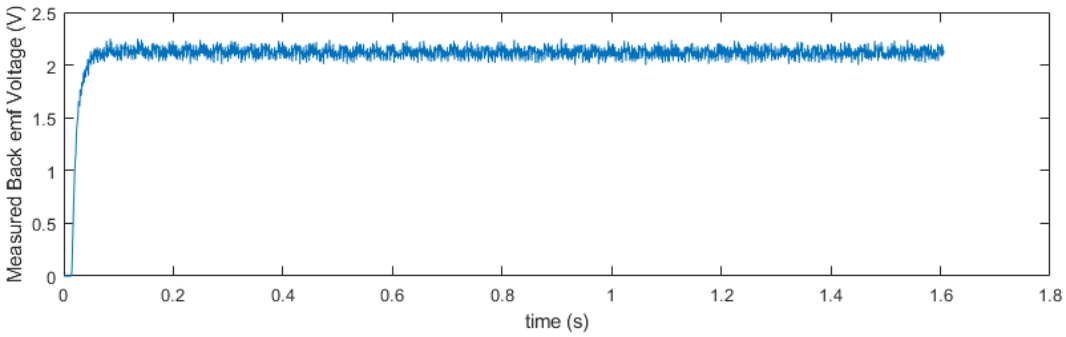
$$V_b = K_b \frac{d\theta}{dt} \rightarrow K_b = \frac{V_b}{\dot{\theta}}$$

نمودار اندازه‌ی سرعت دورانی موتور، بر حسب ولتاژ اندازه‌گیری شده در پایه‌های آن در ادامه آورده شده است:



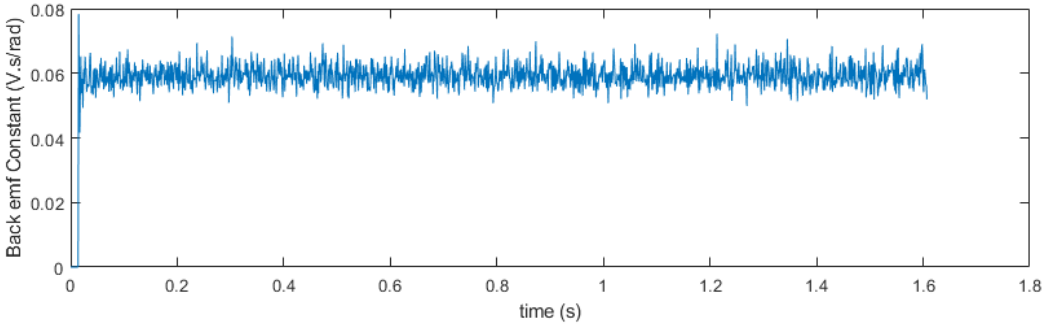
شکل ۳۳: نمودار سرعت دوران شفت موتور در هنگام چرخاندن آن به‌وسیله دریل

[1] Resistance
[2] Inductance
[3] Arduino



شکل ۳۴: نمودار ولتاژ اندازه‌گیری شده در پایه‌های موتور در هنگام چرخاندن شفت آن به‌وسیله دریل

همچنین، مقادیر متناظر K_b حاصل از تقسیم دو نمودار قبلی بر هم در نمودار زیر آورده شده است:



شکل ۳۵ مقادیر به‌دست‌آمده برای K_b در طول زمان

به‌این‌ترتیب، مقدار متوسط K_b به‌صورت زیر به دست می‌آید:

$$K_b = 0.0587 \frac{Vs}{rad}$$

تا به اینجا، مقادیر R_a، L_a و k_b محاسبه شدند. برای به‌دست آوردن سایر پارامترهای موتور، از پاسخ پله‌ی[1] آن استفاده شد. برای این کار، لازم است تا تابع تبدیل[2] موتور محاسبه شود. برای محاسبه‌ی تابع تبدیل، در ابتدا از طرفین معادلات موتور تبدیل لاپلاس[3] می‌گیریم:

$$R_a\, i_a(s) + L_a\, s\, i_a(s) + V_b(s) = V_a(s) \quad (Equation\ 26) \qquad V_b(s) = K_b s\theta(s) \quad (Equation\ 27)$$

$$Is^2\theta(s) + Cs\theta(s) = T(s) - T_L \qquad (Equation\ 28) \qquad T(s) = K_t i_a(s) \qquad (Equation\ 29)$$

[1] Step Response
[2] Transfer Function
[3] Laplace Transform

با جاگذاری مقدار $V_b(s)$ از معادله ۲۷ در معادله ۲۶ و $T(s)$ از معادله ۲۹ در معادله ۲۸ به‌دست می‌آید:

$$R_a\, i_a(s) + L_a\, s\, i_a(s) + K_b s\theta(s) = V_a(s) \qquad (Equation\ 30)$$

$$Is^2\theta(s) + Cs\theta(s) = K_t i_a(s) - T_L \qquad (Equation\ 31)$$

همچنین، با جاگذاری مقدار $i_a(s)$ از معادله ۳۰ در معادله ۳۱ و اعمال $T_L = 0$، خواهیم داشت:

$$\frac{\theta(s)}{V_a(s)} = \frac{k_t}{(R_a + L_a s)(Is^2 + cs) + k_b k_t s}$$

نهایتاً داریم:

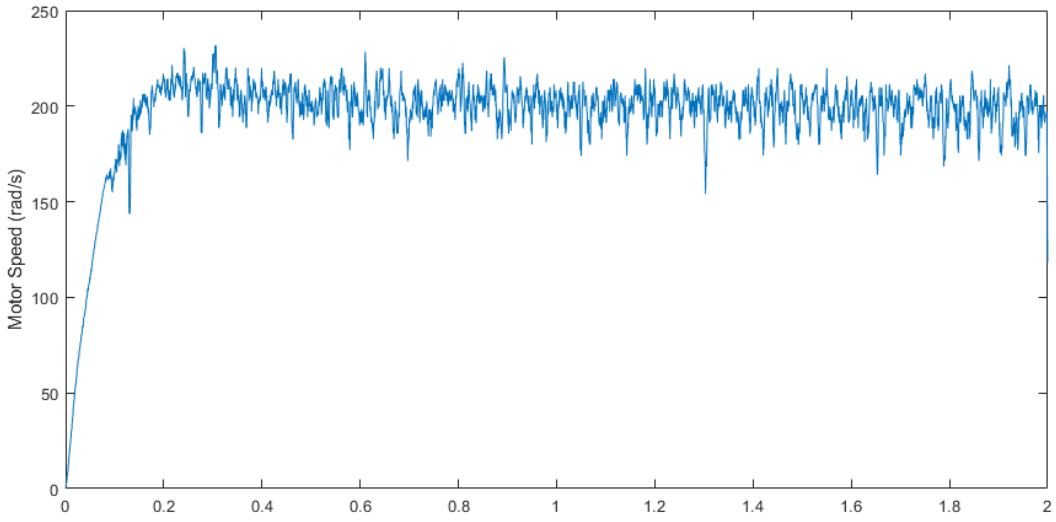
$$G_1 = \frac{\theta(s)}{V_a(s)} = \frac{k_t}{IL_a s^3 + (IR_a + cL_a)s^2 + (R_a c + k_b k_t)s} \qquad (Equation\ 32)$$

همچنین، اگر خروجی تابع تبدیل را سرعت زاویه‌ای در نظر بگیریم:

$$G_1 = \frac{\dot{\theta}(s)}{V_a(s)} = \frac{k_t}{IL_a s^2 + (IR_a + cL_a)s + (R_a c + k_b k_t)} \qquad (Equation\ 32)$$

در تابع تبدیل به‌دست‌آمده مقادیر I، c و k_t مجهول هستند. با به دست آوردن ضرایب تابع تبدیلی به فرم به‌دست‌آمده از پاسخ پله[1] موتور می‌توان پارامترهای مجهول را پیدا کرد.

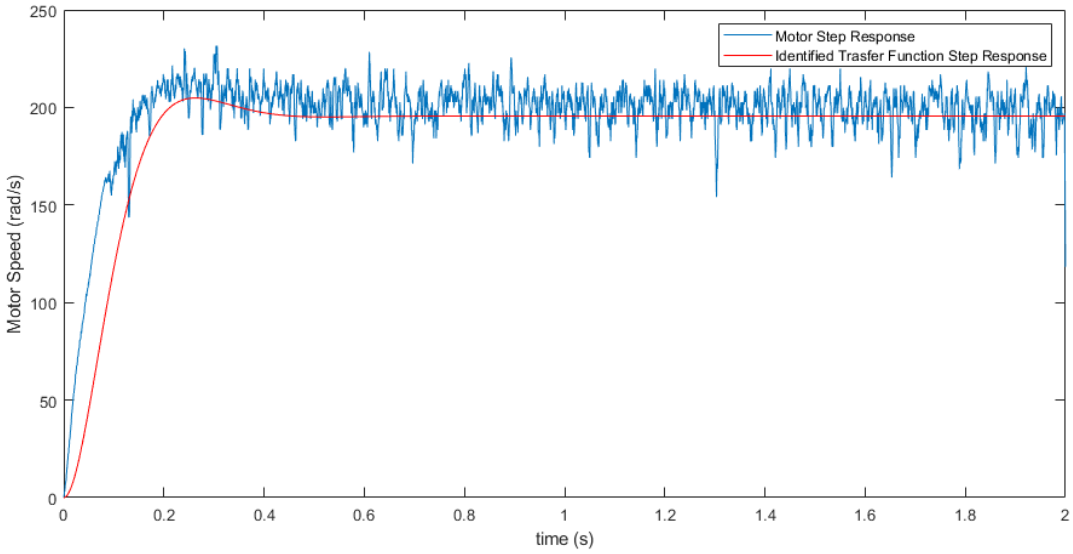
پاسخ پله‌ی موتور به ازای ولتاژ *24V* و با فرکانس نمونه‌برداری[2] ۲ کیلوهرتز به‌مانند زیر است:



شکل ۳۶: پاسخ پله‌ی موتور به ازای ولتاژ *24V*

[1] Step Response
[2] Sampling Frequency

همچنین، پاسخ پله‌ی تابع تبدیل به‌دست‌آمده برای پاسخ فوق، در toolbox شناسایی سیستم[1] متلب به شرح زیر است:



شکل ۳۷: پاسخ پله‌ی موتور در کنار پاسخ پله‌ی تابع تبدیل شناسایی‌شده برای آن

فرم تابع تبدیل شناسایی‌شده به‌صورت زیر است:

$$\hat{G}_1 = \frac{\dot{\theta}(s)}{V_a(s)} = \frac{1}{۰.۰۰۰۴۴۳۲s^2 + 0.01027s + 0.1227}$$

با مقایسه‌ی این تابع با G_1، مجهولات موجود در معادلات به‌دست می‌آید:

$$\frac{IL_a}{k_t} = ۰.۰۰۰۴۴۳۲ \quad , \quad \frac{IR_a + cL_a}{k_t} = 0.01027 \quad , \quad \frac{R_a c + k_b k_t}{k_t} = 0.1227$$

با معلوم بودن R_a، L_a و k_b، دستگاه سه معادله و سه مجهول غیرخطی فوق به‌راحتی حل می‌شود و مقادیر I، K_t و c به‌صورت زیر به دست می‌آیند:

$$k_t = 0.0584\ N.m/A \quad , \quad I = 0.0002\ kg.m^2 \quad , \quad c = 0.0015\ N.m.s/rad$$

به‌این‌ترتیب، فرایند شناسایی پارامترهای موتور به پایان می‌رسد.

[1] System Identification

۴-۲-۲-۳- به‌دست آوردن مشخصه انتقالی درایور موتور[1]

برای به حرکت درآوردن موتور، از یک درایور موتور مدل BTS7960 استفاده می‌شود. ورودی درایور، یک سیگنال PWM[2] است و متناسب با دوره کاری[3] این سیگنال، ولتاژ اعمالی در پایانه‌های موتور تعیین می‌شود. ولتاژ خروجی درایور موتور را می‌توان با تقریب مناسب به‌صورت خطی نسبت به دوره کاری سیگنال PWM ورودی به آن در نظر گرفت. در تعیین مشخصه انتقالی درایور، موارد زیر حائز اهمیت است:

۱- افت ولتاژ درونی درایور مطابق مقدار عنوان‌شده در کاتالوگ و مقادیر مشاهده‌شده در عمل برای سازه در حدود $0.5V$ می‌باشد. این مقدار تقریباً مستقل از شدت‌جریان گرفته‌شده از درایور است.

۲- با توجه به این‌که بیشینه ولتاژ خروجی درایور برابر ولتاژ منبع تغذیه ($25V$) و در حالت دوره کاری ۱۰۰٪ است، شیب تغییرات ولتاژ خروجی درایور به دوره کاری با تقریب مناسب برابر 0.25 است.

۳- با تست کردن روی سازه‌ی آونگ معکوس، آستانه‌ی حرکت پیوسته‌ی دستگاه حدوداً در ولتاژ $4.5V$ بین دوپایه موتور است.

با توجه به موارد فوق، می‌توان مشخصه انتقالی درایور موتور را به‌صورت زیر به‌دست آورد:

$$V_a = MAX(0.25 \times Duty\ Cycle - 4.5, 0)$$

در عبارت فوق، V_a، ولتاژ خروجی درایور و پایه‌های موتور و $Duty\ Cycle$ برحسب درصد بیان می‌شود.

به‌این‌ترتیب، مقدار تمامی پارامترهای دخیل در مدل به دست می‌آیند. بار دیگر، مقدار تمامی این پارامترها در جداول زیر آورده شده است:

جدول ۶: مقادیر پارامترهای دخیل در مدل‌سازی بازو

پارامترهای دخیل در مدل‌سازی بازو	
پارامتر	**مقدار**
چگالی	$1180\ kg/m^3$
طول	$270\ mm$
عرض	$47\ mm$
ضخامت	$6\ mm$

[1] Motor Driver
[2] Pulse-Width Modulation
[3] Duty Cycle

جدول ۷: مقادیر پارامترهای دخیل در مدل‌سازی آونگ

پارامترهای دخیل در مدل‌سازی آونگ	
پارامتر	**مقدار**
چگالی	$1180\ kg/m^3$
طول	$250\ mm$
عرض	$20\ mm$
ضخامت	$6\ mm$

جدول ۸: مقادیر پارامترهای دخیل در مدل‌سازی موتور *DC* (و انکودر متصل به آن)

پارامترهای دخیل در مدل‌سازی موتور ***DC*** **(و انکودر متصل به آن)**	
پارامتر	**مقدار**
مقاومت موتور	$2.5\ \Omega$
اندوکدانس موتور	$2.1\ mH$
ضریب Back emf موتور	$0.0587\ V.s/rad$
اینرسی روتور	$0.0002\ kg.m^2$
ضریب میرایی روتور	$0.0015\ N.m.s/rad$
ضریب گشتاور موتور	$0.0584\ N.m/A$

جدول ۹: مقادیر پارامترهای دخیل در مدل‌سازی انکودر متصل به آونگ

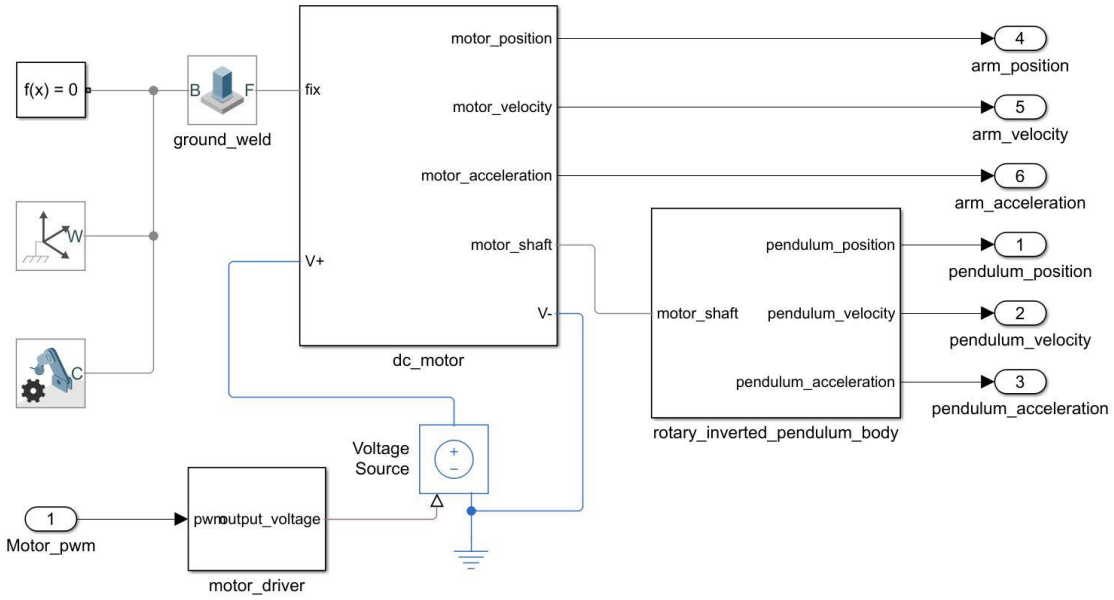
پارامترهای دخیل در مدل‌سازی انکودر متصل به آونگ	
پارامتر	**مقدار**
جرم	$95\ g$
فاصله مرکز جرم از مرکز دوران	$97\ mm$
ضریب اصطکاک ویسکوز شفت	$0.0002\ \ N.m.s/rad$
ضخامت	$6\ mm$

جدول ۱۰: مقادیر پارامترهای دخیل در مدل‌سازی بورد چاپی متصل به بازو

پارامترهای دخیل در مدل‌سازی بورد چاپی متصل به بازو	
پارامتر	**مقدار**
جرم	$100\ g$
فاصله مرکز جرم از مرکز دوران	$50\ mm$

۴-۲-۳- پیاده‌سازی سیستم در سیمولینک[1]

برای پیاده‌سازی سیستم آونگ معکوس دورانی در سیمولینک، از بخش Simscape استفاده شد. این بخش امکان شبیه‌سازی انواع سیستم‌های فیزیکی اعم از مغناطیسی، حرارتی، دینامیکی، ارتعاشاتی، الکتریکی و .... و از همه مهم‌تر امکان ایجاد ارتباطات بین این سیستم‌ها را برای کاربر مهیا می‌کند. در زیر، شمای کلی پیاده‌سازی مدل پیاده‌سازی شده آورده شده است:



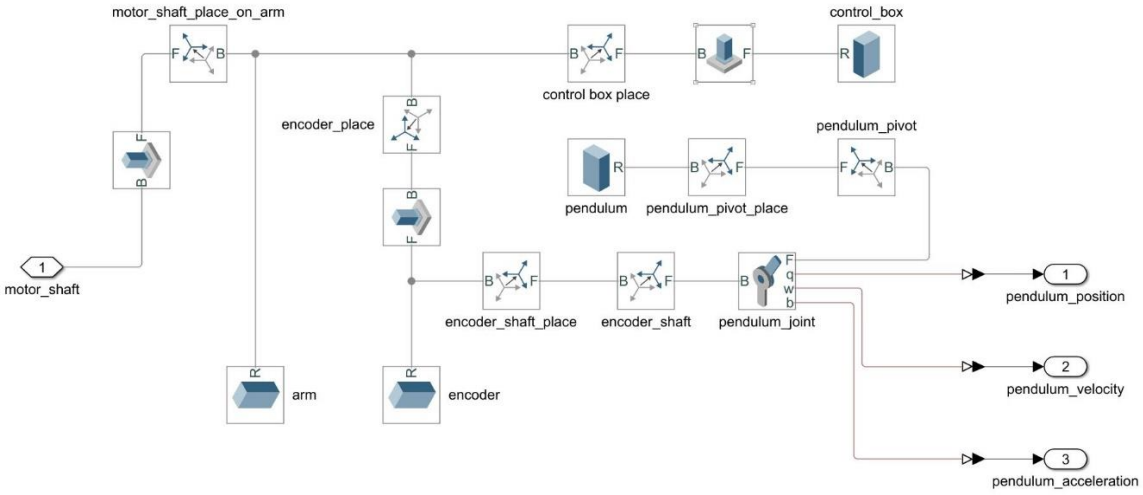
شکل ۳۸: شمای کلی سیستم آونگ معکوس دورانی در سیمولینک

در تصویر فوق، سه زیرسیستم[2] دیده می‌شود:

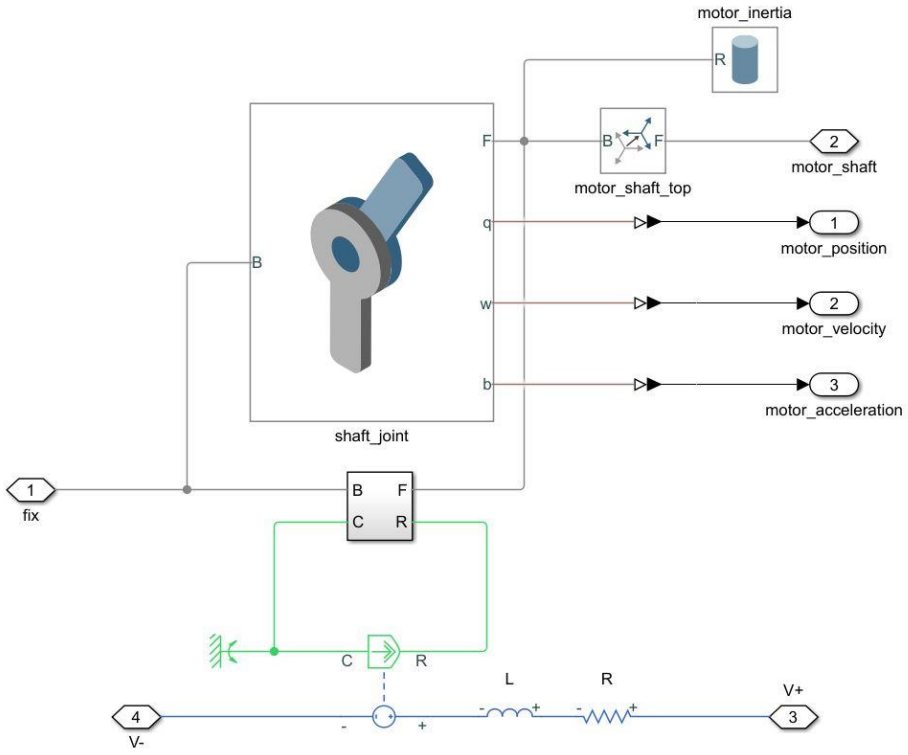
۱- زیرسیستم rotary_inverted_pendulum_body که در آن بخش بازو و آونگ سیستم آونگ معکوس دورانی به کمک بلوک‌های بخش Multibudy از بخش Simscape پیاده‌سازی شده‌اند
۲- زیرسیستم dc_motor که در آن مدلی از موتور dc پیاده‌سازی شده است.
۳- زیرسیستم motor_driver که یک تابع در زبان برنامه‌نویسی Matlab است که ورودی آن مقدار سیگنال pwm ورودی به درایور و خروجی آن ولتاژ اعمالی بر پایه‌های موتور است.

[1] Simulink
[2] Subsystem

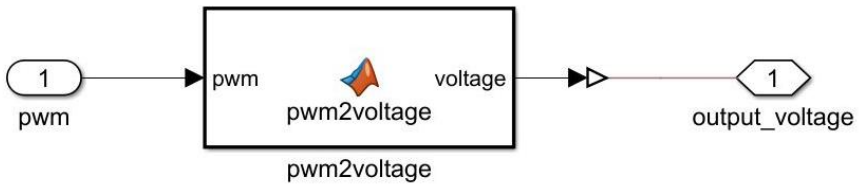
در سه شکل بعدی، نماهایی از بخش‌های این سه زیرسیستم آورده شده است:



شکل39 : نمایی از زیرسیستم *rotary_inverted_pendulum_body*



شکل ۴۰: نمایی از زیرسیستم *dc_motor*



شکل ۴۱: نمایی از زیرسیستم *motor_driver*

لازم به ذکر است که پارامترهای اساسی مدل فوق تماماً به‌صورت متغیر[1] در محیط متلب پیاده‌سازی شده‌اند تا فرایند تغییر و به‌روزرسانی آن‌ها سهولت یابد.

۴-۲-۴- کار با محیط شبیه‌سازی

کار با محیط شبیه‌سازی گرافیکی در سیمولینک از دو طریق میسر است:

۱- به‌طور مستقل از محیط متلب
۲- به‌طور غیرمستقل از محیط پایتون

نرم‌افزار متلب بستری مناسب برای پیاده‌سازی انواع الگوریتم‌های کنترلی کلاسیک است و برای پیاده‌سازی این روش‌ها می‌توان از امکانات نرم‌افزار متلب به همراه مدل سیمولینک آونگ معکوس دورانی استفاده کرد.

از طرف دیگر، پایتون، قوی‌ترین زبان برنامه‌نویسی جهت پیاده‌سازی الگوریتم‌های یادگیری ماشینی[2] می‌باشد. به کمک افزونه[3] Matlab engine در نرم‌افزار متلب و کتابخانه‌ی engine.matlab در زبان پایتون، می‌توان یک درگاه ارتباطی بین متلب و پایتون ایجاد کرد و به کمک این درگاه می‌توان مدل آونگ معکوس دورانی را کنترل کرد. جهت برقراری این ارتباط و کنترل مدل، یک کلاس به نام SimulinkModel در زبان پایتون نوشته شد که با فراخوانی آن می‌توان به مدل سیمولینک متصل شد.

[1] Variable
[2] Machine Learning
[3] Plug-in

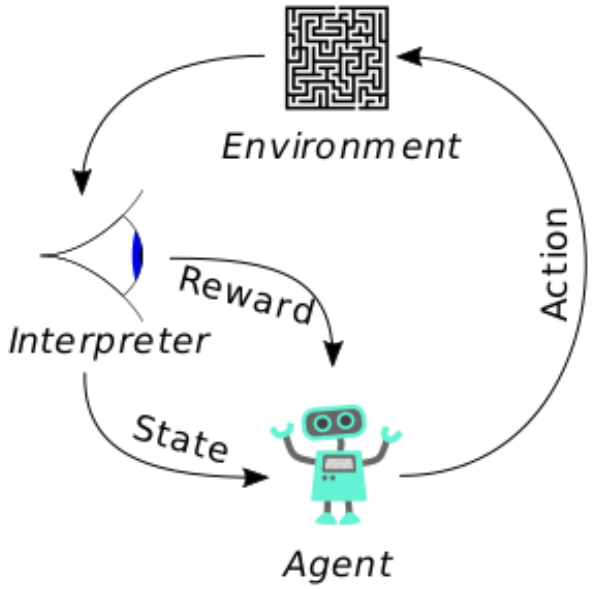
۵- یادگیری تقویتی[۱]

یادگیری تقویتی یکی از شاخه‌های یادگیری ماشین[۲] است که با الهام گرفتن از روند طبیعی یادگیری موجودات زنده، به کشف شیوه‌ی عملکرد صحیح در یک محیط[۳] می‌پردازد. در طی فرایند فراگیری رفتار صحیح که اصطلاحاً یادگیری[۴] نامیده می‌شود، یک یا چند عملگر[۵] از طریق تعامل با یک محیط سعی در بیشینه کردن پاداش[۶] دریافتی خود از محیط دارند. یادگیری تقویتی، در کنار یادگیری با نظارت و یادگیری بدون نظرات، از انواع الگوهای یادگیری در حوزه یادگیری ماشین می‌باشد.

در این بخش، به بررسی مفاهیم مرتبط با حوزه یادگیری تقویتی خواهیم پرداخت.

۵-۱- تعاریف پایه

شماتیک زیر، کلیت فرایند طی شده در الگوریتم‌های یادگیری تقویتی را نشان می‌دهد:



شکل ۴۲: شماتیک فرایندهای یادگیری تقویتی[۷]

[1] Reinforcement Learning
[2] Machine Learning
[3] Environment
[4] Learning
[5] Agent
[6] Reward

[۷] برگرفته از Wikipedia.org

حال، به بررسی مفاهیم کلیدی در سازوکار نشان داده‌شده در شکل می‌پردازیم:

- **عملگر (Agent)**: تصمیم‌گیرنده در محیط است که سعی دارد از طریق بیشینه کردن پاداش دریافتی‌اش، رفتار صحیح در محیط را یاد بگیرد.
- **محیط (Environment)**: سازوکاری است که Agent با آن در تعامل است و سعی در یادگیری عملکرد بهینه در آن دارد.
- **وضعیت (State)**: حالات ممکن از محیط است که Agent می‌تواند در آن‌ها قرار گیرد.
- **عمل (Action)**: کاری است که Agent می‌تواند متناسب با یک State خاص انجام دهد.
- **پاداش (Reward)**: جایزه‌ای است که Agent متناسب با عملکردش از محیط دریافت می‌نماید. عملکردهای بهتر، پاداش‌های بیشتری دریافت می‌کنند.

سایر مفاهیم اساسی یادگیری تقویتی به شرح زیر هستند:

- **سیاست (Policy)**: استراتژی است که Agent آن را در محیط دنبال می‌کند و با توجه به آن در State های مختلف، Actionهای متناسب را انتخاب می‌نماید. به‌عبارت‌دیگر، Policy تابعی است که State یک Agent را به یک Action نگاشت می‌کند.
- **سیاست بهینه (Optimal Policy)**: سیاستی است که پاداش ناشی از دنبال کردن آن، از پاداش ناشی از دنبال کردن همه‌ی سیاست‌های ممکن دیگر بیشتر باشد. ثابت می‌شود که در یک مسئله‌ی یادگیری تقویتی، همواره سیاست بهینه به‌طور یکتا وجود دارد.
- **اپیزود (Episode)**: مجموعه اعمال یک Agent از زمان شروع فرایند تصمیم‌گیری در محیط تا پایان را یک اپیزود می‌نامند. فرایندهای یادگیری تقویتی که در قالب چند اپیزود انجام می‌شوند را اپیزودیک[1] و سایر فرایندها را ادامه‌دار[2] می‌نامند. در بسیاری از مواقع، با تعریف یک نقطه‌ی شروع و پایان دلخواه برای فرایندهای ادامه‌دار، آن‌ها را به فرایندهای اپیزودیک تبدیل می‌کنیم.
- **بهره‌برداری از دانش قبلی (Exploitation)**: زمانی است که Agent از دانش به‌دست‌آمده‌ی خود در محیط برای تصمیم‌گیری استفاده می‌کند.
- **اکتشاف (Exploration)**: زمانی است که Agent بدون توجه به دانش به‌دست‌آمده‌ی خود در محیط، برای تصمیم‌گیری دست به اعمال رندم می‌زند.

[1] Episodic Tasks
[2] Continuing Tasks

- **استفاده از دانش قبلی در مقابل اکتشاف (Exploration Vs. Exploitation):** هردوی تصمیم‌گیری بر مبانی دانش پیشین و یا اکتشاف، مزایا و معایب منحصر به خود را دارند. در هنگامی‌که Agent دانش محیطی کافی جمع‌آوری کرده باشد، تکیه‌بر دانشش عملی عاقلانه به نظر می‌رسد. اما اگر تمام تصمیم‌گیری‌های Agent بر مبنای دانش پیشینش باشد، Agent ممکن است بسیاری از Action ها را تجربه نکرده باشد و نتواند دانش خود را در مورد رفتار صحیح در محیط و بیشینه کردن پاداش تکمیل نماید. از طرفی، استفاده از دانش قبلی در هنگام فرایند یادگیری عملاً بی‌معنا است. این موضوع را اصطلاحاً Exploration Vs. Exploitation Trade-off می‌نامند. روش‌های زیادی جهت رفع این مشکل معرفی‌شده است که از میان آن‌ها می‌توان به سیاست epsilon-greedy اشاره نمود. در این سیاست، Agent در هر مرحله تصمیم‌گیری به‌احتمال ε عمل تصادفی انجام می‌دهد و اکتشاف می‌کند و به‌احتمال $1-\varepsilon$ بر دانش خود تکیه می‌کند. لازم به ذکر است که ε یک عدد مثبت بین صفر و یک می‌باشد. مرسوم است که باگذشت زمان و افزایش دانش Agent مقدار ε را کاهش می‌دهند تا جایی که نهایتاً Agent برای تصمیم‌گیری تنها به دانش پیشین خود متوسل می‌شود.

نمادگذاری متداول برای مفاهیم فوق در ادبیات حوزه‌ی یادگیری تقویتی در جدول زیر نمایش داده‌شده‌اند.

جدول ۱۱: نمادگذاری متداول برای مفاهیم پایه‌ای در یادگیری تقویتی

مفهوم	نماد
$State$	S
$Action$	A
$Reward$	R
$Policy$	π
$Optimal\ Policy$	π_*
$Episode$	E

اندیس t در زیر هر یک از نمادهای S، A و R به معنی مربوط بودن نماد به زمان tام اپیزود است. به‌عنوان‌مثال، S_t به معنی State بازدید شده در زمان t است.

۲-۵- هدف نهایی مسائل یادگیری تقویتی

در مسائل حوزه‌ی یادگیری تقویتی، هدف نهایی، فراگیری یک سیاست (Policy) بهینه برای تصمیم‌گیری در محیط است که Agent به کمک آن بتواند Reward دریافتی خود از محیط را بیشینه سازد. به‌عبارت‌دیگر، در طی فرایند آموزش، Agent با استفاده از مشاهدات محیطی سعی می‌کند سیاست خود را به سیاست بهینه (π_*) نزدیک نماید.

۵-۳- مسائل یادگیری تقویتی گسسته

در مسائل یادگیری تقویتی گسسته، Stateها و Actionها دارای مقادیر گسسته هستند. در مقابل مسائل یادگیری تقویتی گسسته، آن دسته از مسائل که Stateها و Actionهایشان مقادیر پیوسته دارد.

در ادامه، به بررسی فرمولاسیون مسائل یادگیری تقویتی پیوسته و شیوه‌ی حل آن‌ها خواهیم پرداخت.

۵-۳-۱- فرمولاسیون ریاضی مسائل یادگیری گسسته

همان‌طور که پیش‌تر گفته شد، در مسائل یادگیری تقویتی گسسته، مقادیر State و Action همواره گسسته است. برای مدل‌سازی این دسته از مسائل، از فرایند تصمیم‌گیری مارکوف[1] استفاده می‌شود.

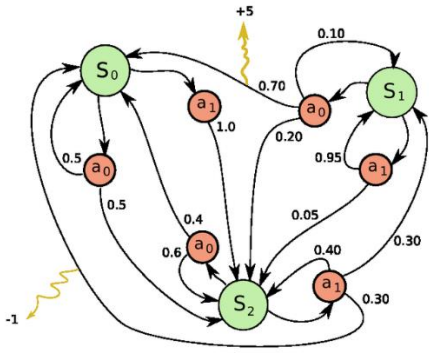
۵-۳-۱-۱- فرایند تصمیم‌گیری مارکوف

فرایند تصمیم‌گیری مارکوف چارچوبی ریاضی است برای مدل‌سازی تصمیم‌گیری در شرایطی که نتایج تا حدودی تصادفی و تا حدودی تحت کنترل یک تصمیم گیر است. فرایندهای تصمیم‌گیری مارکوف، فرایندهای کنترلی تصادفی زمان گسسته هستند. در این فرایندها، تصمیم‌گیرنده در زمان t با اتخاذ تصمیم A_t به‌احتمال P از وضعیت S_t به وضعیت S_{t+1} منتقل می‌شود و پاداش R_{t+1} را از محیط دریافت می‌کند.

$$S_t \rightarrow A_t \rightarrow R_{t+1} \rightarrow S_{t+1}$$

این روند، تا زمان رسیدن به State پایانی[2] ادامه می‌یابد.

شکل زیر، یک فرایند تصمیم‌گیری مارکوف با سه حالت (State) و دو عمل (Action) را نشان می‌دهد. احتمال‌های انتقال از هر وضعیت به وضعیت دیگر در شکل نشان داده‌شده‌اند.



شکل ۴۳: یک فرایند تصمیم‌گیری مارکوف ساده با سه حالت (*State*) و دو عمل (*Action*)[3]

[1] Markov Decision Process
[2] Terminal State

[3] برگرفته از Wikipedia.org

فرایندهای تصمیم‌گیری مارکوف، به علت نزدیکی به مفاهیم پایه‌ای مسائل یادگیری تقویتی گسسته، بستری مناسب برای مدل‌سازی این مسائل می‌باشند.

۵-۳-۱-۲- بهینه‌سازی در تصمیم‌گیری مارکوف

در فرایندهای تصمیم‌گیری، قصد داریم سلسله تصمیماتی[1] را اتخاذ نماییم که منجر به بیشترین پاداش[2] دریافتی از محیط شود. به‌طور مشخص، درزمانی مشخص t، قصد داریم به‌گونه‌ای عمل کنیم که عبارت زیر بیشینه شود:

$$G_t = E\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1}\right]$$

در عبارت فوق، E امید ریاضی است و به علت وجود عدم قطعیت در فرایند تصمیم‌گیری مارکوف وارد معادلات می‌شود. γ نیز نرخ کاهش[3] نام دارد و یک عدد بین صفر و یک است. مقدار γ، میزان توجه به پاداش آینده را مشخص می‌کند. به ازای $\gamma = 0$، Agent تمام تصمیم‌گیری‌ها را با توجه به پاداش سریع و به‌صورت حریصانه[4] اتخاذ می‌کند. در مقابل، به ازای $\gamma = 1$، Agent در هنگام تصمیم‌گیری نه‌تنها مقادیر پاداش فوری، بلکه تمامی پاداش‌های آینده را نیز در نظر می‌گیرد. این ضریب یکی از پارامترهای اساسی[5] در الگوریتم‌های یادگیری تقویتی است و بسته به شرایط مقدار آن انتخاب می‌شود.

۵-۳-۱-۳- مفهوم State Value و Action Value

در راستای بیشینه کردن پاداش، دو State value و Action Value به‌عنوان معیاری از مناسب بودن ***State*** ها و ***Action*** ها به‌صورت زیر تعریف می‌شوند:

- ***State Value***: مقدار پاداشی است که انتظار می‌رود با شروع از وضعیت S و دنبال کردن سیاست π به آن دست‌یابیم.

$$V_\pi(s) \doteq E_\pi[G_t|S_t = s] = E_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1}\,|S_t = s\right]$$

[1] Action
[2] Reward
[3] Discount Factor
[4] Greedy
[5] Hyper Parameter

- *Action Value*: مقدار پاداشی است که انتظار می‌رود با شروع از وضعیت S، انجام عمل A و دنبال کردن سیاست π به آن دست‌یابیم.

$$\mathrm{Q}_\pi(s,a) \doteq E_\pi[G_t|S_t=s,A_t=a] = E_\pi\left[\sum_{k=0}^{\infty}\gamma^k R_{t+k+1}\,|S_t=s,A_t=a\right]$$

۵-۳-۱-۴- معادلات بِلمَن[1]

در فرایند تصمیم‌گیری مارکوف، معادلات بلمن، *State Value* و *Action Value* ها را در زمان‌های مختلف به‌طور بازگشتی به هم مرتبط می‌سازد. برای هر یک از State Value ها می‌توان نوشت:

$$\begin{aligned}V_\pi(s) \doteq E_\pi[G_t|S_t=s] &= E_\pi\left[\sum_{k=0}^{\infty}\gamma^k R_{t+k+1}\,|S_t=s\right]\\ &= \sum_a \pi(a|s)\sum_{s'}\sum_r p(s',r|s,a)\big[r+\gamma E_\pi[G_{t+1}|S_{t+1}=s']\big]\\ &= \sum_a \pi(a|s)\sum_{s'}\sum_r p(s',r|s,a)[r+\gamma V_\pi(s')]\end{aligned}$$

همچنین، برای هر یک از Action Value ها داریم:

$$\begin{aligned}\mathrm{Q}_\pi(s,a) \doteq E_\pi[G_t|s_t=s,a_t=a] &= E_\pi\left[\sum_{k=0}^{\infty}\gamma^k R_{t+k+1}\,|S_t=s,A_t=a\right]\\ &= \sum_{s'}\sum_r p(s',r|s,a)\left[r+\gamma\sum_{a'}\pi(a'|s')E_\pi[G_{t+1}|S_{t+1}=s',A_{t+1}=a']\right]\\ &= \sum_{s'}\sum_r p(s',r|s,a)\left[r+\gamma\sum_{a'}\pi(a'|s')\mathrm{Q}_\pi(s',a')\right]\end{aligned}$$

به‌این‌ترتیب، به دو رابطه‌ی بازگشتی می‌رسیم که *State Value* و *Action Value* ها را در زمان‌های مختلف به هم مرتبط می‌سازد. به این معادلات، معادلات بلمن می‌گویند. بار دیگر، این دو معادله را بازنویسی می‌کنیم:

$$V_\pi(s) = \sum_a \pi(a|s)\sum_{s'}\sum_r p(s',r|s,a)[r+\gamma V_\pi(s')]$$

$$\mathrm{Q}_\pi(s,a) = \sum_{s'}\sum_r p(s',r|s,a)\left[r+\gamma\sum_{a'}\pi(a'|s')\mathrm{Q}_\pi(s',a')\right]$$

[1] Bellman Equations

۵-۳-۲- شیوه‌ی پیاده‌سازی تابع سیاست[1] در محیط‌های گسسته

پیش‌تر بیان شد که سیاست، استراتژی است که Agent آن را در محیط دنبال می‌کند و با توجه به آن در State های مختلف، Actionهای مناسب را انتخاب می‌نماید. برای محیط‌های گسسته‌ی ساده با تعداد State و Action محدود، تابع هدف به‌صورت یک جدول قابل پیاده‌سازی است که سرهای آن State ها و ستون‌هایش Actionهای متناظر هر State می‌باشد. به این روش پیاده‌سازی تابع سیاست روش جدولی[2] می‌گویند. Agent برای تصمیم‌گیری در هر State می‌تواند به ردیف متناظر State در جدول مراجعه نماید و با توجه به Action Value های موجود عمل کند.

باوجود سادگی پیاده‌سازی، روش جدولی برای محیط‌های پیچیده با تعداد زیاد State و Action عملی نیست زیرا جدول متناظر بسیار بزرگ خواهد بود و ذخیره‌سازی آن دشوار است. در این موارد به جای استفاده از جدول، از تخمین زن‌های تابعی[3] استفاده می‌شود که در ورودی خود، State را دریافت می‌کنند و در خروجی Action Value های متناظر جهت تصمیم‌گیری را می‌دهند. استفاده از شبکه‌های عصبی مصنوعی[4] به‌عنوان قوی‌ترین تخمین زن‌های تابعی، در یادگیری تقویتی جهت ذخیره‌ی سیاست بسیار متداول و مرسوم است. این شبکه‌ها، به علت توانایی بی‌نظیرشان در ایجاد نگاشت‌های غیرخطی پیچیده و سهولت فرایند به‌روزرسانی پارامترهایشان در سال‌های اخیر به یکی از اجزاء جدایی‌ناپذیر یادگیری تقویتی بدل شده‌اند.

۵-۳-۳- یادگیری در محیط‌های گسسته

در یادگیری تقویتی، فرایند یادگیری[5] به صورت فراگرفتن رفتار صحیح در محیط درنتیجه‌ی تعامل با آن تعریف می‌شود. در فرایندهای یادگیری تقویتی گسسته، فرایند یادگیری به دو بخش کلی تقسیم می‌شود:

بخش ۱: فرایند ارزیابی سیاست[6]

با مشخص بودن سیاست π و به کمک معادلات فوق، از روش‌های گوناگونی ازجمله برنامه‌نویسی پویا یا روش مونت‌کارلو می‌توان مقادیر *State Value* و *Action Value* ها را تعیین کرد. به این فرایند به‌اصطلاح ارزیابی سیاست می‌گویند. با به دست آوردن مقادیر *State Value* و *Action Value* ها، Agent می‌تواند در هر مرحله، بهترین عمل متناسب با یک سیاست π را انجام دهد.

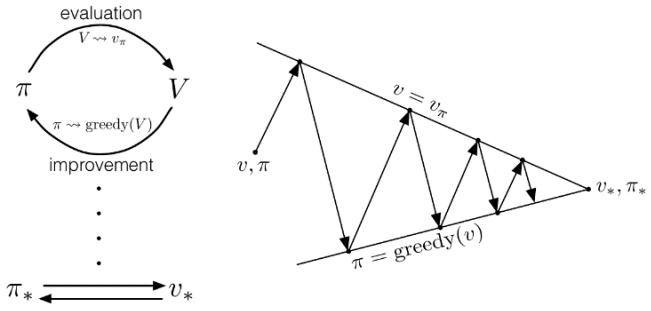
بخش ۲: بهبود سیاست[7]

به فرایند ارتقاء سیاست برای رسیدن به سیاستی بهتر، اطلاق می‌شود.

---

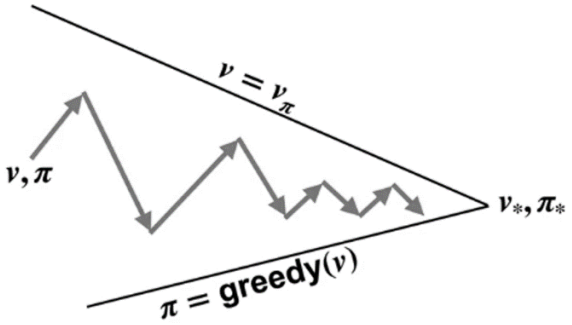
[1] Policy Function
[2] Tabular Method
[3] Function Approximator
[4] Artificial Neural Networks
[5] Learning
[6] Policy Evaluation
[7] Policy Improvement

در الگوریتم‌های پایه‌ای یادگیری تقویتی، این دو فرایند مستقلاً از یکدیگر دنبال می‌شوند و بعد از هر مرحله از ارزیابی سیاست، متناسب با مقادیر به‌دست‌آمده برای Action Value و State Value ها سیاست بهبود پیدا می‌کند. این روند به‌صورت تکراری[1] انجام می‌شود تا به یک سیاست قابل‌قبول برسیم. شکل زیر نمایی از این فرایند را نشان می‌دهد:



شکل ۴۴: فرایند تکراری ارزیابی و بهبود سیاست[2]

در مواجهه با مسائل بزرگ با محیط‌های پیچیده این روند عملی نیست زیرا تعداد Stateها و Actionها معمولاً بسیار زیاد است و به‌دست آوردن مقادیر Action Value و State Value فرایندی بسیار طولانی است. در عوض، در این محیط‌ها این فرایندها به‌طور هم‌زمان و در کنار هم انجام می‌شوند تا به‌تدریج به سیاست بهینه در محیط نزدیک شویم. شکل زیر نمایانگر این فرایند است:



شکل ۴۵: اجرای هم‌زمان ارزیابی و بهبود سیاست[3]

الگوریتم‌هایی که به شکل ثانویه عمل می‌کنند، عملاً امکان یادگیری در محیط‌های پیچیده‌ی گسسته را فراهم می‌نمایند. در اینجا، ما به یکی از مهم‌ترین این الگوریتم‌ها به نام Q-Learning اشاره می‌کنیم.

[1] Iterative

[2] برگرفته از coursera.org

[3] برگرفته از coursera.org

۵-۳-۳-۱- الگوریتم Q-Learning

همان‌گونه که از نامش مشخص است، در Q-Learning به یادگیری Action Value ها پرداخته می‌شود.

فرمولاسیون به‌روزرسانی مقادیر Action Value در Q-Learning به شرح زیر است:

$$Q^{new}(s_t, a_t) = (1 - \alpha)Q^{old}(s_t, a_t) + \alpha(r_t + \gamma\, max_a Q(s_{t+1}, a))$$

در عبارت فوق:

- γ همان نرخ کاهش[1] است.
- α یک عدد مثبت بین صفر و یک موسوم به ضریب یادگیری[2] است.

به ازای ضرایب یادگیری بزرگ ($\alpha \to 1$)، *Agent* بدون توجه به دانش کسب‌شده‌ی پیشین خود، مقدار *Action Value* را تنها بر اساس مشاهدات حال حاضرش به‌روزرسانی می‌نماید. در نقطه‌ی مقابل، به ازای ضرایب یادگیری کوچک ($\alpha \to 0$)، مقادیر *Action Value* تغییری نمی‌کنند و یادگیری انجام نمی‌شود.

به‌این‌ترتیب، شبه کد[3] الگوریتم Q-Learning با استفاده از سیاست *Epsilon-greedy* به‌صورت زیر است:

شبه کد ۱: الگوریتم *Q-Learning*

Require:
Sates $\mathcal{X} = \{1, \ldots, n_x\}$
Actions $\mathcal{A} = \{1, \ldots, n_a\}, \qquad A : \mathcal{X} \Rightarrow \mathcal{A}$
Reward function $R : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$
Black-box (probabilistic) transition function $T : \mathcal{X} \times \mathcal{A} \to \mathcal{X}$
Learning rate $\alpha \in [0, 1]$, typically $\alpha = 0.1$
Discounting factor $\gamma \in [0, 1]$
procedure QLEARNING($\mathcal{X}$, A, R, T, α, γ)
  Initialize $Q : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$ arbitrarily
  while Q is not converged **do**
    Start in state $s \in \mathcal{X}$
    while s is not terminal **do**
      Calculate π according to Q and exploration strategy (e.g. $\pi(x) \leftarrow \arg\max_a Q(x, a)$)
      $a \leftarrow \pi(s)$
      $r \leftarrow R(s, a)$ ▷ Receive the reward
      $s' \leftarrow T(s, a)$ ▷ Receive the new state
      $Q(s', a) \leftarrow (1 - \alpha) \cdot Q(s, a) + \alpha \cdot (r + \gamma \cdot \max_{a'} Q(s', a'))$
      $s \leftarrow s'$
  return Q

در *Q-Learning* معمولی از روش‌های جدولی[4] جهت پیاده‌سازی سیاست استفاده می‌شود.

[1] Discount Factor
[2] Learning Rate
[3] Pseudocode
[4] Tabular Methods

۵-۳-۳-۲- الگوریتم Deep Q-Learning

Deep Q-Learning استفاده از یک شبکه‌ی عصبی عمیق[1] جهت پیاده‌سازی سیاست در کنار قانون به‌روزرسانی مطرح‌شده در Q-Learning است. به این شبکه اصطلاحاً Policy Network گویند. ورودی این شبکه، یک State به‌خصوص و خروجی آن Action Value های متناظر Actionهای آن State می‌باشد. در این الگوریتم، به‌روزرسانی مقادیر Action Value عملاً معادل به‌روزرسانی وزن‌های شبکه عصبی است.

تابع هزینه‌ی تعریف‌شده برای آموزش شبکه عصبی در این الگوریتم به فرم زیر است:

$$Loss = \left[R_{t+1} + \gamma \max_{a'} q(s', a')\right] - q(s, a)$$

این تابع، معادل قانون به‌روزرسانی برای Q-Learning معمولی است. برای محاسبه‌ی تابع هزینه‌ی فوق، می‌توان مراحل زیر را دنبال می‌کنیم:

- **گام ۱:** مقدار Action Value های یک State را به کمک شبکه‌ی عصبی محاسبه می‌نمایی.
- **گام ۲:** بسته به نوع سیاست (حریصانه، epsilon-greedy یا ...) Action موردنظر را انتخاب می‌نماییم.
- **گام ۳:** Action را انجام می‌دهیم و به State بعدی می‌رویم (s'). همچنین، پاداش R_{t+1} را از محیط دریافت می‌کنیم.
- **گام ۴:** مقادیر Action Value برای State جدید را به کمک شبکه محاسبه می‌نماییم. و بیشینه‌ی آن ($\gamma \max_{a'} q(s', a')$) را به‌دست می‌آوریم.
- **گام ۵:** Loss را حساب می‌کنیم.

روش فوق، یک ایراد اساسی دارد: وزن‌های شبکه هم با توجه به $q(s, a)$ و هم با توجه به $q(s', a')$ به‌روزرسانی می‌شوند. این موضوع در کار شبکه اختلال ایجاد می‌کند زیرا مثلاً برای حالتی که $R_{t+1} + \gamma \max_{a'} q(s', a')$ بزرگتر از $q(s, a)$ باشد، وزن‌ها طوری به‌روزرسانی می‌شوند که $\max_{a'} q(s', a')$ کوچک و $q(s, a)$ بزرگ شود. این در حالی است که ما فقط می‌خواهیم به‌روزرسانی شبکه در راستای بزرگ شدن $q(s, a)$ باشد. جهت رفع این مشکل، یک کپی از شبکه‌ی اصلی تعیین می‌کنیم، وزن‌های آن را ثابت نگه می‌داریم و مقدار $\max_{a'} q(s', a')$ را به کمک آن حساب می‌کنیم. به این شبکه اصطلاحاً Target Network می‌گویند. بعد از گذشت تعداد مشخصی اپیزود، مقدار Target Network را دوباره برابر Policy Network قرار می‌دهیم.

در هنگام آموزش شبکه‌های عصبی عمیق، معمولاً چند داده به‌طور هم‌زمان وارد شبکه می‌شوند، گرادیان روی تمامی داده‌ها حساب می‌شود و پارامترهای شبکه به کمک میانگینی از گرادیان به‌دست‌آمده به‌روزرسانی می‌شود. این کار دو مزیت بزرگ دارد: اول آنکه استفاده از میانگینی از چند گرادیان به جای یک گرادیان باعث کاهش نویز گرادیان شده و همگرایی را تسریع می‌نماید. همچنین، این کار باعث جلوگیری از تقلید رفتار یک داده‌ی خاص توسط شبکه می‌شود. آموزش شبکه به کمک چند داده و به‌صورت هم‌زمان موجب استفاده از پردازش

[1] Deep Neural Network

موازی، بهره‌برداری مناسب از توان محاسباتی و کوتاه شدن فرایند آموزش می‌شود. برای استفاده از این خاصیت در آموزش شبکه‌های عصبی مورداستفاده در Q-Learning، مفهومی تحت عنوان Replay Memory برای الگوریتم تعریف می‌شود. Replay Memory عبارت است از یک لیست از وضعیت‌های مشاهده‌شده‌ی قبلی و پاداش دریافتی از محیط که در هر زمان از به‌روزرسانی شبکه تعدادی از آن‌ها ذخیره و استفاده می‌شود. به‌عبارت‌دیگر، در Replay Memory تعدادی لیست به فرم زیر ذخیره می‌شود:

$$(s_t, a_t, r_{t+1}, s_{t+1})$$

در هر بار تعامل با محیط، یک لیست به فرم زیر به Replay Memory اضافه می‌شود. همچنین، در هر مرحله از به‌روزرسانی وزن‌ها، تعدادی از لیست‌های فوق انتخاب‌شده و یک دسته داده وارد شبکه می‌شوند. فرایند به‌روزرسانی وزن‌های شبکه روی این دست داده انجام می‌شود.

به‌این‌ترتیب، شبه کد[1] الگوریتم Deep Q-Learning با استفاده از سیاست *Epsilon-greedy* به‌صورت زیر است:

شبه کد ۲ :الگوریتم *Deep Q-Learning*

Initialize replay memory $\mathcal{D}$ to capacity N
Initialize action-value function Q with random weights
for episode $= 1, M$ **do**
  Initialise sequence $s_1 = \{x_1\}$ and preprocessed sequenced $\phi_1 = \phi(s_1)$
  for $t = 1, T$ **do**
    With probability ϵ select a random action a_t
    otherwise select $a_t = \max_a Q^*(\phi(s_t), a; \theta)$
    Execute action a_t in emulator and observe reward r_t and image x_{t+1}
    Set $s_{t+1} = s_t, a_t, x_{t+1}$ and preprocess $\phi_{t+1} = \phi(s_{t+1})$
    Store transition $(\phi_t, a_t, r_t, \phi_{t+1})$ in $\mathcal{D}$
    Sample random minibatch of transitions $(\phi_j, a_j, r_j, \phi_{j+1})$ from $\mathcal{D}$
    Set $y_j = \begin{cases} r_j & \text{for terminal } \phi_{j+1} \\ r_j + \gamma \max_{a'} Q(\phi_{j+1}, a'; \theta) & \text{for non-terminal } \phi_{j+1} \end{cases}$
    Perform a gradient descent step on $(y_j - Q(\phi_j, a_j; \theta))^2$ according to equation 3
  end for
end for

۵-۴- مسائل یادگیری تقویتی پیوسته

در بسیاری از مسائل یادگیری تقویتی، Agent باید Action خود را از یک فضای پیوسته انتخاب کند. در چنین مسائلی، تغییر مقادیر Action حتی به مقدار اندک ممکن است موجب تغییرات بزرگ در محیط و شکست Agent در مأموریتش شود پس

[1] Pseudocode

ضروری است تا بتوان فضای Action را به‌صورت پیوسته مدل‌سازی نمود. روش‌های گسسته‌ای که تا به اینجای کار موردبررسی قرار گرفت، ازجمله Deep Q-Learning، دارای Actionهای گسسته هستند و بنابراین جهت حل مسائل یادگیری تقویتی پیوسته مناسب نیستند.

زمانی که فضای Action گسسته است، می‌توان با محاسبه‌ی Action Value برای هر Action و یافتن Action با بیشترین Action Value تصمیمات بهینه در راستای سیاست گرفت. برای پیاده‌سازی رویکردی مشابه در فضای پیوسته، لازم است تا یک تابع پیوسته از Actionها و Stateها مانند $Q(s,a)$ داشته باشیم و در هنگام قرارگیری در یک State مانند s_0، مقداری از Action را انتخاب نماییم که تابع $Q(s_0,a)$ را بیشینه نماید. این به معنی حل یک مسئله‌ی بهینه‌سازی در هر بار فرایند تصمیم‌گیری است. واضح است که این روش برای محیط‌های پیوسته به علت طولانی و پیچیده بودن محاسبات به‌هیچ‌عنوان عملی نیست.

۵-۴-۱- فرایند یادگیری در محیط پیوسته

از مهم‌ترین الگوریتم‌های یادگیری تقویتی پیوسته می‌توان به الگوریتم Deep Deterministic Policy Gradient (DDPG) اشاره نمود. در ادامه به توضیح این الگوریتم می‌پردازیم.

۵-۴-۱-۱- الگوریتم Deep Deterministic Policy Gradient (DDPG)

الگوریتم Deep Deterministic Policy Gradient (DDPG) را می‌توان به‌عنوان همتای پیوسته‌ی Deep Q-Learning در نظر گرفت. به‌مانند Q-Learning، این الگوریتم بر پایه‌ی این نکته استوار است که اگر مقدار $Q^{\pi_*}(s,a)$ را داشته باشیم، برای هر State بهترین Action به‌صورت زیر قابل‌تعیین است:

$$a^{\pi_*}(s) = argmax_a Q^{\pi_*}(s,a)$$

بنابراین الگوریتم DDPG سعی به یافتن Action بهینه برای هر وضعیت از طریق یافتن $Q^{\pi_*}(s,a)$ دارد.

این الگوریتم به‌طور هم‌زمان دو تابع را فرامی‌گیرد:

۱- یک تابع به نام Q-function به فرم $Q(s,a)$ که ورودی آن یک State مانند s و یک Action مانند a و خروجی آن Action Value متناظر است.

۲- یک تابع سیاست به فرم $\mu(s)$ که ورودی آن یک State و خروجی آن Action منتخب است.

زمانی که فضای Action گسسته است، می‌توان با محاسبه‌ی Action Value برای هر Action و یافتن Action با بیشترین Action Value تصمیمات بهینه در راستای سیاست گرفت. برای پیاده‌سازی رویکردی مشابه در فضای پیوسته، لازم است تا یک تابع پیوسته از Actionها و Stateها مانند $Q(s,a)$ داشته باشیم و در هنگام قرارگیری در یک State مانند s_0، مقداری از Action را انتخاب نماییم که تابع $Q(s_0,a)$ را بیشینه نماید. این به معنی حل یک

مسئله‌ی بهینه‌سازی در هر بار فرایند تصمیم‌گیری است. واضح است که این روش برای محیط‌های پیوسته به علت طولانی و پیچیده بودن محاسبات به‌هیچ‌عنوان عملی نیست.

ازآنجاکه فضای Action پیوسته است، تابع $Q^*(s,a)$ نسبت به a مشتق‌پذیر است. این زمینه‌ی اجرای یک الگوریتم مبتنی بر گرادیان مؤثر جهت محاسبه‌ی یک سیاست به فرم $\mu(s)$ را می‌دهد. این تابع، به‌عنوان ورودی، یک State می‌گیرد و در خروجی، Action انتخابی متناظر آن State و مطابق با سیاست را می‌دهد. در ادامه، به جای اجرای یک فرایند بهینه‌سازی جهت یافتن مقدار a که تابع $Q(s_0,a)$ را بیشینه می‌کند، می‌توان بیشینه‌ی این تابع را با مقدار $Q(s_0,\mu(s))$ تخمین زد.

جهت بررسی دقیق‌تر الگوریتم DDPG، ابتدا معادله‌ی بهینه‌ی بلمن برای Action Valueها را در نظر می‌گیریم:

$$Q^{\pi_*}(s,a) = E_{s\prime\sim P}[r(s,a) + \gamma \max_a Q^{\pi_*}(s',a')]$$

در عبارت فوق، منظور از $s'\sim P$ آن است که State بعدی یا s' از یک توزیع احتمالاتی به فرم $P(.|s,a)$ می‌آید. معادله‌ی فوق نقطه‌ی شروعی برای یافتن یک تخمین‌زن برای $Q^{\pi_*}(s,a)$ است. فرض کنید تخمین‌زن به فرم $Q_\phi(s,a)$ در دست داریم که ϕ پارامترهای تخمین‌زن است. تابع هزینه‌ی[1] مورداستفاده جهت سنجش عملکرد تخمین‌زن به فرم زیر می‌باشد:

$$L = \left(Q_\phi(s,a) - \left(r + \gamma \max_{a'} Q_\phi(s',a')\right)\right)^2$$

چالش اصلی در محاسبه‌ی عبارت فوق، محاسبه‌ی مقدار $\max_{a'} Q_\phi(s',a')$ است. در الگوریتم DDPG فرض می‌شود:

$$\max_{a'} Q_\phi(s',a') = \mu(s')$$

و بنابراین خواهیم داشت:

$$L = \left(Q_\phi(s,a) - (r + \gamma\mu(s'))\right)^2$$

ذکر دو نکته‌ی پایانی در مورد این الگوریتم خالی از لطف نیست:

۱- بنا به دلایلی کاملاً مشابه با آنچه برای Deep Q-Learning گفته شد، استفاده از Replay Memory در الگوریتم DDPG نیز مرسوم و رایج است.

۲- به‌مانند Deep Q-Learning، در DDPG نیز از یک Target Network جهت به‌روزرسانی $Q_\phi(s,a)$ استفاده می‌شود.

[1] Loss Function

در الگوریتم DDPG، گاهی اوقات به شبکه‌ای که وضیفه‌ی تعیین Action از State ورودی را دارد، Actor و به شبکه‌ای که وضیفه‌ی تعیین Action Value متناظر را دارد Critic می‌گویند.

شبه کد الگوریتم DDPG در ادامه آورده شده است:

شبه کد ۳: الگوریتم *Deep Deterministic Policy Gradient (DDPG)*

Input: initial policy parameters θ, Q-function parameters ϕ, empty replay buffer $\mathcal{D}$
Set target parameters equal to main parameters $\theta_{\text{targ}} \leftarrow \theta$, $\phi_{\text{targ}} \leftarrow \phi$
repeat
  Observe state s and select action $a = \text{clip}(\mu_\theta(s) + \epsilon, a_{Low}, a_{High})$, where $\epsilon \sim \mathcal{N}$
  Execute a in the environment
  Observe next state s', reward r, and done signal d to indicate whether s' is terminal
  Store (s, a, r, s', d) in replay buffer $\mathcal{D}$
  If s' is terminal, reset environment state.
  if it's time to update **then**
    for however many updates **do**
      Randomly sample a batch of transitions, $B = \{(s, a, r, s', d)\}$ from $\mathcal{D}$
      Compute targets

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

      Update Q-function by one step of gradient descent using

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

      Update policy by one step of gradient ascent using

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

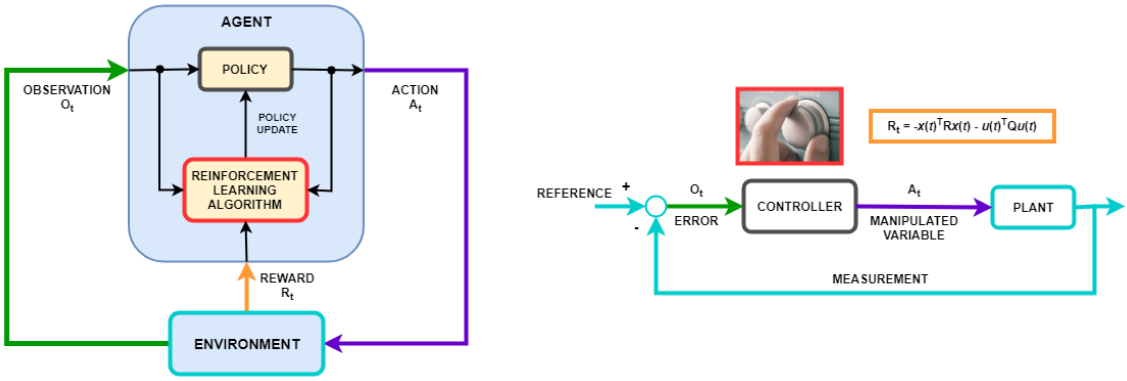
      Update target networks with

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

    end for
  end if
until convergence

۵-۵- کاربرد یادگیری تقویتی در مسائل حوزه مهندسی کنترل

رفتار یک تابع سیاست در یادگیری تقویتی (مشاهده‌ی وضعیت محیط و تصمیم‌گیری برای انجام اعمال مختلف بر اساس آن) شباهت به نقش کنترلر در مسائل حوزه‌ی کنترل دارد. یادگیری تقویتی می‌تواند به صورت یک مسئله‌ی کنترل در نظر گرفته شود. شکل زیر نمایانگر نزدیکی یادگیری تقویتی به فرایند کنترل با بازخورد[1] است.



شکل ۴۶: تشابه یادگیری تقویتی و فرایند کنترل با بازخورد[2]

ارکان اساسی مسائل این دو حوزه، مطابق جدول زیر به هم ارتباط دارند:

جدول ۱۲: تشابه مفاهیم حوزه‌ی کنترل و یادگیری تقویتی

یادگیری تقویتی	کنترل سیستم‌ها
سیاست (Policy)	کنترلر (Controller)
محیط (Environment)	هر آنچه کنترلر نیست (شامل Plant، فیلتر، نویز اندازه‌گیری، مبدل آنالوگ به دیجیتال و برعکس و ...)
مشاهده (Observation)	اندازه‌گیری (Measurement)
عمل (Action)	متغییر کنترلی (Control Variable)
پاداش (Reward)	سیگنال خطای ورودی به کنترلر (Error Signal)

به این ترتیب، یادگیری تقویتی را می‌توان به عنوان یک کنترلر در محیطش درنظر گرفت. برخلاف کنترلرهای کلاسیک (مانند PID) برای طراحی کنترلر مبتنی بر یادگیری تقویتی هیچ نیازی به دانستن دینامیک محیط نیست و مستقل از پیچیدگی معادلات حاکم بر سیستم، می‌توان از یادگیری تقویتی جهت کنترل آن استفاده نمود.

[1] Feedback

[2] برگرفته از Mathworks.com

۶- کنترل آونگ معکوس دورانی به کمک یادگیری تقویتی

چندین مسئله‌ی کنترلی را می‌توان برای سازه‌ی آونگ دورانی معکوس متصور شد. در ساده‌ترین حالت، صرفاً موقعیت زاویه‌ای آونگ حایز اهمیت و به موقعیت و نوع حرکت بازو توجهی نمی‌شود. این مسئله، یک مسئله‌ی کنترلی تک ورودی – تک خروجی[۱] است. در مسائل کنترلی پیچیده‌تر، می‌توان موقعیت بازو را نیز وارد سیستم کرد تا مسئله‌ی کنترلی به یک مسئله‌ی چند ورودی – تک خروجی[۲] بدل شود. در این حالات، می‌توان موقعیت زاویه‌ای دلخواهی برای بازو تعیین نمود تا دقیقاً در آن قرار گیرد. همچنین، می‌توان حرکت بازو را طوری محدود کرد تا با حداقل حرکت، آونگ را کنترل نماید.

در این پروژه، به علت ضیق وقت، تنها به بررسی عملکرد یادگیری تقویتی روی مسئله‌ی اول (کنترل وضعیت آونگ بدون توجه به وضعیت بازو) می‌پردازیم و بررسی حالات دیگر مسئله‌ی کنترل آونگ دورانی معکوس را به پژوهش‌های بعدی موکول می‌نماییم. برای کنترل آونگ معکوس دورانی، به کمک یادگیری تقویتی ابتدا لازم است تا ورودی ها و خروجی‌های سیستم را مشخص نماییم:

- **ورودی سیستم کنترلی** دوره‌ی کاری[۳] سیگنال PWM ورودی به سیستم و جهت چرخش موتور می‌باشد.
- **خروجی سیستم کنترلی** موقعیت آونگ در صفحه‌ی چرخش می‌باشد.

جهت تعریف یک مسئله‌ی یادگیری تقویتی برای کنترل آونگ معکوس دورانی با دو مسئله‌ی زیر مواجه هستیم:

۱- تعیین ورودی‌الگوریتم کنترلی
۲- تعیین خروجی الگوریتم کنترلی

در ادامه، در دو بخش مجزا سعی می‌شود به مسائل فوق پاسخ مناسب داده شود.

۶-۱- ورودی الگوریتم کنترلی

یک رویکرد ممکن برای ورودی سیستم کنترلی این است که در هر لحظه تنها موقعیت آونگ به الگوریتم داده شود. اما با این رویکرد هرگز نمی‌توان کنترلر پایدار مدنظرمان را بسازیم. علت این است که آونگ معکوس دورانی، یک سیستم با مرتبه‌ی مخالف صفر است و وضعیت آونگ در هر زمان به وضعیت گذشته‌اش وابسطه است. بنابراین برای نمایش کامل وضعیت آونگ، لازم است تا علاوه بر وضعیت زمان حال، اطلاعاتی در مورد گذشته‌ی آن نیز در اختیار الگوریتم قرار داده شود. این کار می‌تواند به دو صورت انجام شود:

۱- موقعیت زاویه‌ای آونگ و تعداد دلخواهی از مشتقات عددی آن (سرعت زاویه‌ای، شتاب زاویه‌ای و ...) به صورت یک بردار به شبکه داده شود.
۲- یک سری زمانی به طول دلخواه از موقعیت‌های اخیر آونگ به شبکه داده شود.

---

[1] Single Input, Single Output (SISO)
[2] Multiple Inputs, Single Output (MISO)
[3] Duty Cycle

استفاده از روش اول به نظر منطقی‌تر است زیرا کاری که در این روش انجام می‌پذیرد باعث مستقل شدن ورودی‌ها از زمان نمونه‌برداری[1] می‌شود. این درحالی است که در الگوریتمی با ورودی‌های از نوع دوم، زمان نمونه‌برداری نمی‌تواند تغییر کند و به عنوان مثال، الگوریتمی که با داده‌های نمونه‌برداری شده با فرکانس 1000 Hz آموزش داده شده است نمی‌تواند خروجی صحیحی به ازای ورودی‌هایی از فرکانس‌های نمونه برداری دیگر تولید کند. علت این موضوع آن است که فاصله‌ی زمانی بین داده‌های موجود در سری زمانی حائز اهمیت است. بنا به دلیل ذکر شده، از روش اول برای تولید ورودی‌های الگوریتم استفاده می‌شود.

۶-۲- خروجی الگوریتم کنترلی

بدیهی است که ورودی الگوریتم یادگیری تقویتی مورد استفاده جهت کنترل سیستم (دوره‌ی کاری سیگنال PWM ورودی به سیستم)، مقداری پیوسته است. به این ترتیب، جهت حل طراحی یک کنترلر مناسب برای آونگ معکوس دورانی به کمک یادگیری تقویتی، دو راهکار کلی وجود دارد:

- **راهکار اول:** فضای تصمیم‌گیری (مقادیر دوره‌ی کاری سیگنال PWM ورودی به سیستم) را گسسته‌سازی[2] کنیم. جهت دریافت جواب مناسب از سیستم در این حالت، لازم است تا تعداد مقادیر گسسته نهایی به اندازه‌ی کافی زیاد باشد تا بتواند به درستی تخمینی از حالت پیوسته را ارائه دهد. در این راهکار، می‌توان مسئله را با الگوریتم Deep Q-Learning حل نماییم.
- **راهکار ثانویه:** بدون گسسته‌سازی فضای تصمیم‌گیری، مستقیماً از روش‌های یادگیری تقویتی پیوسته جهت کنترل آونگ معکوس دورانی استفاده نماییم. در این‌حالت می‌توانیم مسئله را به کمک الگوریتم Deep Deterministic Policy Gradient (DDPG) حل نماییم.

هر یک از دو راهکار فوق مزایا و معایب مختص به خود را دارند و تا قبل از مشاهده‌ی نتایج عملی عملکردشان نمی‌توان به طور قطعی نتیجه‌گیری در مورد آنها انجام داد.

- پیاده‌سازی الگوریتم Deep Q-Learning و آموزش آن به مراتب آسان‌تر از پیاده‌سازی الگوریتم DDPG و فرایند آموزش آن است. علت این موضوع، ساختار پیچیده‌تر الگوریتم DDPG می‌باشد و همانطور که پیش‌تر توضیح داده شده بود، در DDPG لازم است تا دو شبکه جهت دریافت Action انتخابی متناسب با State و Action Value متناظر با State و Action به صورت مجزا پیاده‌سازی و آموزش داده شوند. این درحالی است که در Q-Learning تنها یک شبکه Policy Network آموزش داده می‌شود.

- با وجود آسانی پیاده‌سازی Deep Q-Learning، لازم است تا جهت رسیدن به تقریب مناسبی از محیط Action پیوسته، تعداد Actionهای گسسته بیشتر شود. این موضوع، موجب می‌شد که ساختار شبکه‌ی عصبی در Q-Learning اندکی پیچیده‌تر از ساختار شبکه‌ی عصبی در DDPG شود. باتوجه به این که با یک مسئله‌ی کنترل بی‌درنگ مواجه هستیم، پیچیده‌تر شدن ساختار می‌تواند موجب ایجاد تاخیر زیاد در محاسبات و نهایتاً ایجاد

[1] Sampling Time
[2] Discretize

اشکال در فرایند کنترل آونگ شود. البته، مطلب ذکر شده بسته به میزان حساسیت سیستم به تغییرات کوچک در سیگنال کنترلی ورودی دارد. اگر در تجربه ثابت شود که سیستم به تغییرات جزئی در ولتاژ موتور حساسیت کمی دارد، می‌توان مقادیر گسسته‌ی ورودی را با فاصله‌ی بیشتری از هم انتخاب نمود تا فضای Action های گسسته کوچکتر شود.

در طی فرایند آموزش، عملکرد هر دو شبکه‌ی فوق را برکارکرد آونگ معکوس دورانی بررسی می‌نماییم.

۷- فرایند آموزش

جهت پیاده‌سازی هر دو الگوریتم Deep Q-Learning و Deep Deterministic Policy Gradient از فریم‌ورک[1] Keras از کتابخانه‌ی Tensor-Flow در زبان برنامه‌نویسی پایتون استفاده می‌شود. همچنین، در اجرای فرایند آموزش از یک جی‌پی‌یو[2] Nvidia GTX-1050 استفاده می‌کنیم. فرایند آموزش به کمک هر دو الگوریتم، از دو بخش کلی تشکیل می‌شود:

- **بخش ۱:** آموزش در محیط شبیه‌سازی Simulink که به موازات فرایند ساخت دستگاه انجام شد
- **بخش ۲:** آموزش روی سازه‌ی فیزیکی آونگ معکوس دورانی. در طی این فرایند، از دانش کسب شده در محیط شبیه‌سازی به عنوان یک دانش اولیه[3] برای کنترل سازه‌ی فیزیکی استفاده شد. به عبارتی، شبکه‌های عصبی مورد استفاده در آموزش سازه‌ی فیزیکی، معماری مشابه شبکه‌های آموزش داده شده در محیط شبیه‌سازی داشتند و با وزن‌های مشابه، وزن‌دهی اولیه شده بودند.

در فرایند آموزش به کمک هردو الگوریتم ذکر شده در بالا، مقادیر مکان، سرعت و شتاب زاویه‌ای در هر لحظه به عنوان State انتخاب می‌شوند. همچنین به عنوان تابع پاداش، از تابعی به فرم زیر استفاده خواهیم کرد. این تابع هدف، از محیط آونگ معکوس خطی کتابخانه‌ی Gym[4] در پایتون اخذ شده است.

$$R = -\left(\theta_*^2 + 0.1\dot{\theta}_*^{\,2} + 0.0001 * (motor\ voltage)^2\right)$$

در این معادله:

- θ_* به صورت $\frac{\theta}{\pi}$ تعریف می‌شود که θ موقعیت زاویه‌ای آونگ نسبت به محور قائم است. (مطابق شکل ۱)
- i اندیس Action انتخابی می‌باشد.

[1] Framework
[2] Graphical Processing Unit (GPU)
[3] Prior Knowledge
[4] کتابخانه‌ی Gym یک کتابخانه‌ی گرافیکی برای Python است که در آن تعدادی از سیستم‌های فیزیکی و غیر فیزیکی جهت اجرای آموزش یادگیری تقویتی پیاده‌سازی شده‌اند.

رابطه‌ی فوق از سه بخش مجزا تشکیل شده است که به توضیح علت وچود هر سه بخش می‌پردازیم:

- ترم θ_*^2 در تابع پاداش سعی در نزدیک نگه داشتن آونگ به نقطه‌ی تعادل بالایی‌اش دارد.
- ترم $0.1\dot{\theta}_*{}^2$ سعی در کوچک نگه داشتن سرعت آونگ و نزدیک کردن آن به وضع پایدار دارد.
- ترم $0.0001 * (motor\ voltage)^2$ سعی در نزدیک نگه داشتن ولتاژ پایه‌های موتور به صفر دارد.

۷-۱- الگوریتم Deep Q-Learning

همانطور که در بخش ۶ توضیح داده شده بود، جهت استفاده از الگوریتم Deep-Q Learning برای کنترل آونگ معکوس دورانی، لازم است تا فضای Actionها (دوره‌ی کاری سیگنال PWM ورودی به موتور) را گسسته‌سازی نماییم. بازه‌ی تغییر ولتاژ ورودی مجاز برای موتور از 24V تا 24V- می‌باشد. از طرفی، مطابق آنچه در بخش ۴ گفته شد، آستانه‌ی حرکت موتور در هنگام اتصال به آونگ و بازو حدود 4.5V است. فرایند گسسته‌سازی را به صورتی انجام می‌دهیم که ۹ ورودی مجاز در این بازه داشته باشیم. به عبارتی، مقادیر گسسته برای سیگنال‌های PWM را طوری تعیین می‌کنیم که مقادیر ولتاژ متناظر موتور مطابق سری زیر شود:

$$V_1 = -24V\ , V_2 = -18V\ , V_3 = -12V\ , V_4 = -6V\ ,$$
$$V_5 = 0V\ , V_6 = 6V\ , V_7 = 12V\ , V_8 = 18V\ , V_9 = 24V$$

Action های متناظر جهت تولید ولتاژهای بالا در پایه‌های موتور، به ترتیب a_1، a_2، a_3، ... و a_9 است. به این ترتیب، خروجی لایه‌ی آخر شبکه‌ی عصبی استفاده شده در الگوریتم Deep-Q Learning ۹ نورون خواهد داشت.

همچنین، جهت فراگیری سیاست مناسب، از یک شبکه‌ی عصبی ۴ لایه با معماری زیر استفاده می‌شود:

Input Size: State Size (3)
Output Size: Number of Actions (9)
Architecture:
Layer 1: Dense (n_neurons=64, input_dim= Input Size, activation=Leaky ReLu)
Layer 2: Dense (n_neurons=128, input_dim= 64, activation= Leaky ReLu)
Layer 3: Dense (n_neurons=64, input_dim= 128, activation=Sigmoid)
Layer 3: Dense (n_neurons=Output Size, input_dim=32, activation=Linear)

همچنین، پارامترهای آموزش شبکه به شرح زیر می‌باشند:

- Optimizer: ADAM
- Learning Rate: $\alpha = 0.001$
- Decay Rate: $\gamma = 0.95$
- Replay Memory Size: 2000
- Initial Exploration Rate: $\epsilon = 1$
- Initial Exploration Rate: $\varepsilon = 0.995$
- Minimum Exploration Rate: $\epsilon_{min} = 0.01$

پارامترهای آموزشی تعیین شده همگی از روی مدل‌های موفق برای آموزش آونگ معکوس خطی در محیط Gym الهام گرفته شده‌اند.

۷-۲- الگوریتم Deep Deterministic Policy Gradient

بر خلاف الگوریتم Deep-Q Learning، در هنگام استفاده از الگوریتم Deep Deterministic Policy Gradient دیگر نیازی به گسسته‌سازی مقادیر Action نیست. خروجی این الگوریتم، یک مقدار بین منفی یک و یک است که به کمک یک نگاشت خطی، به مقادیر متناظر برای دوره‌ی کاری سیگنال PWM تبدیل می‌شود. جهت تولید این مقادیر، در لایه‌ی آخر شبکه از یک تابع فعالساز تانژانت هایپربولیک[1] استفاده می‌شود.

معماری شبکه‌ی Actor استفاده شده در این بخش به شرح زیر می‌باشد:

Input Size: State Size (3)
Output Size: Action Size (1)
Architecture:
Layer 1: Dense (n_neurons=256, input_dim= Input Size, activation=Leaky ReLu)
Layer 2: Dense (n_neurons=512, input_dim= 256, activation= Leaky ReLu)
Layer 3: Dense (n_neurons=Output Size, input_dim=256, activation=Tanh)

معماری شبکه‌ی Critic استفاده شده اندکی پیچیده تر است. در این شبکه، ورودی State و Action به صورت مجزا وارد دو لایه از نوع Dense با نام‌های Action Dense و State Dense می‌شوند. در ادامه، خروجی این دو لایه به هم چسبانده شده و به عنوان ورودی یک شبکه‌ی عصبی دولایه‌ی دیگر داده می‌شود تا نهایتاً مقدار Action Value متناظر تولید گردد. معماری این شبکه در زیر آورده شده است:

State Dense:
Input Size: State Size (3)
Output Size: 256
Architecture:
Layer 1: Dense (n_neurons=256, input_dim= Input Size, activation=Leaky ReLu)
Layer 2: Dense (n_neurons=Output Size, input_dim= 256, activation=Leaky ReLu)

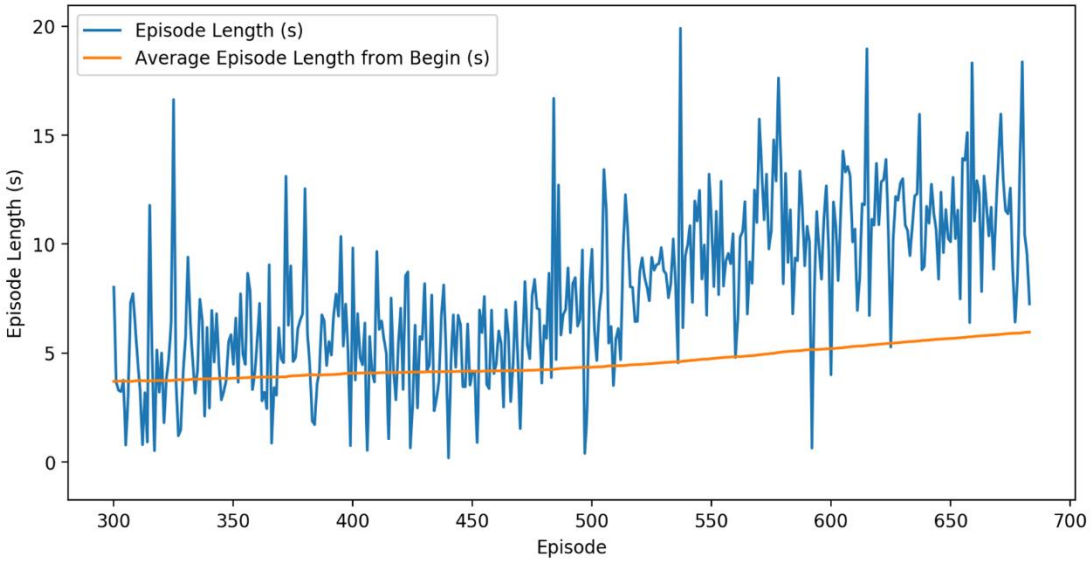
Action Dense:
Input Size: Action Size (1)
Output Size: 256
Architecture:
Layer 1: Dense (n_neurons=Output Size, input_dim= Input Size, activation=Leaky ReLu)

Trunk:
Input Size: State Dense Output Size **+** Action Dense Output Size
Output Size: 1
Architecture:
Layer 1: Dense (n_neurons=256, input_dim= Input Size, activation=Leaky ReLu)
Layer 3: Dense (n_neurons=Output Size, input_dim=256, activation=Tanh)

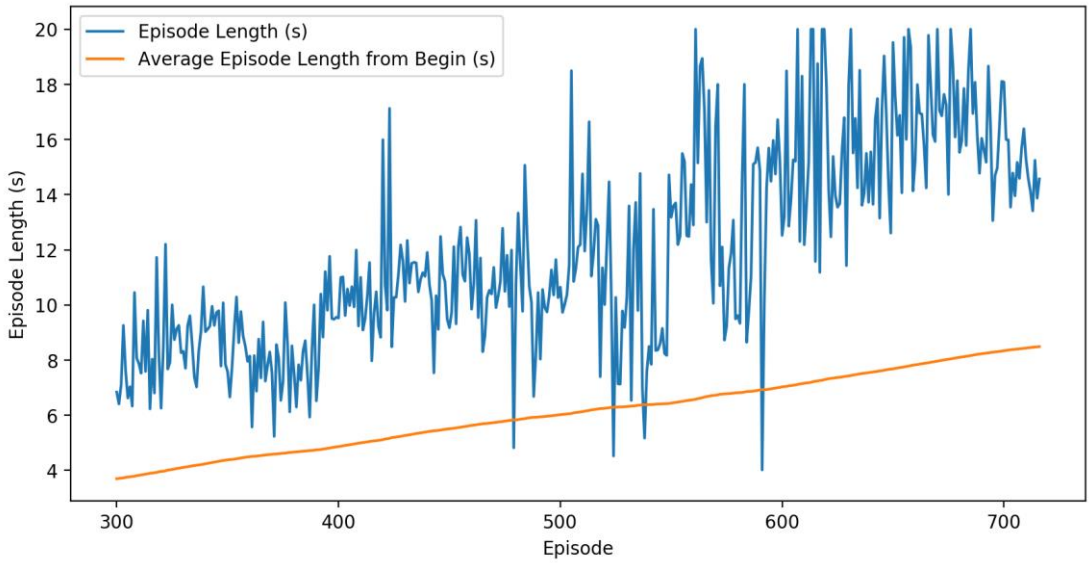
[1] Tangent Hyperbolic (Tanh)

۸- نتایج آموزش

فرایند آموزش هردو الگوریتم Deep Q-Learning و DDPG ابتدا در محیط شبیه‌سازی و برای 700 اپیاک انجام شد. نمودار مدت زمان اپیزودها برای ۴۰۰ اپیزود آخر در زیر آورده شده است:

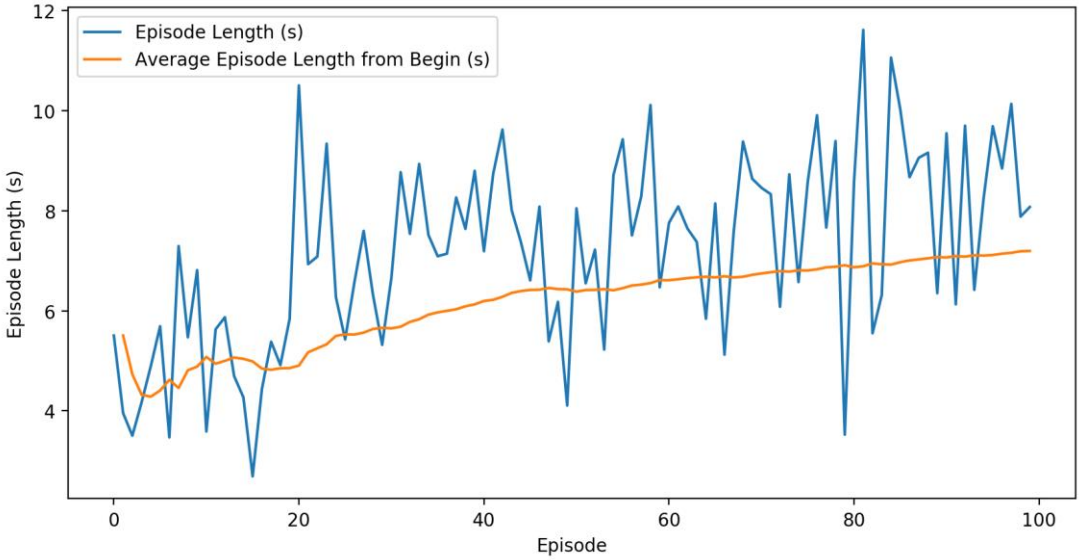


شکل ۴۷: نتایج آموزش الگوریتم *Deep Q-Learning* در ۴۰۰ اپیزود پایانی

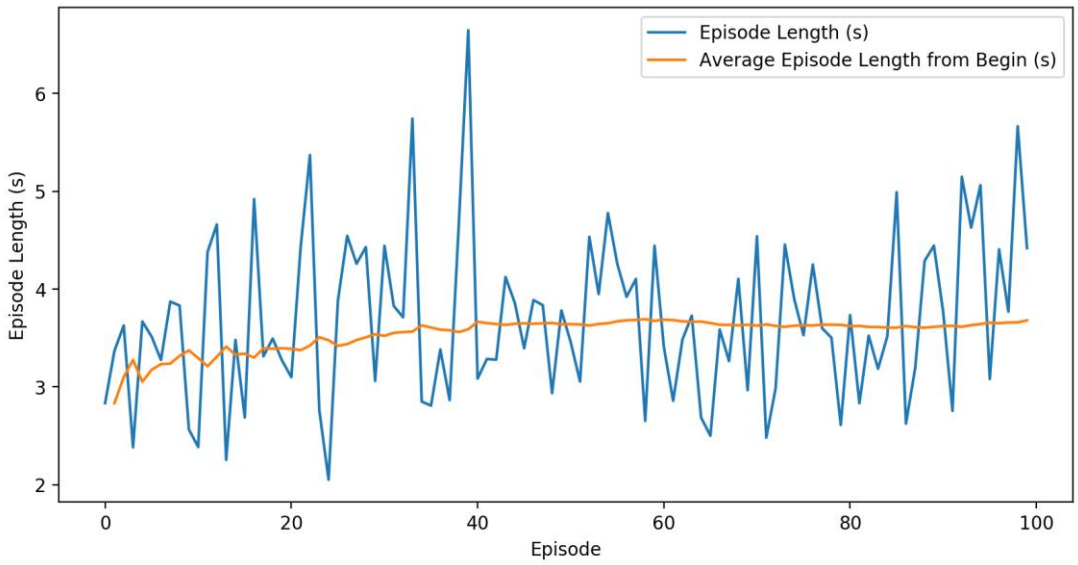


شکل ۴۸: نتایج آموزش الگوریتم *Deep Deterministic Policy Gradient* در ۴۰۰ اپیزود پایانی

در ادامه، شبکه‌های آموزش داده شده در محیط شبیه‌سازی روی سازه‌ی واقعی تست شدند. متاسفانه، به علت طولانی بودن فرایند آموزش، این فرایند در ثالب پند بخش اجرا شد و نمودار کاملی از آموزش سازه‌ی آونگ معکوس دورانی برای همه‌ی اپیزودها در دست نیست. نمودارهای زیر، عمکلرد سازه‌ی آونگ برای ۱۰۰ اپیزود آموزشی برای در الگوریتم را نمایش می‌دهند.



شکل ۴۹: نتایج آموزش الگوریتم *Deep Q-Learning* در ۱۰۰ اپیزود آموزش روس سازی آونگ معکوس دورانی



شکل ۵۰: نتایج آموزش الگوریتم *Deep Deterministic Policy Gradient* در ۱۰۰ اپیزود آموزش روس سازی آونگ معکوس دورانی

۹- تفسیر و بررسی نتایج

همانطور که در بخش نتایج مشاهده شد، روش Deep Deterministic Policy Gradient در کنترل آونگ معکوس دورانی عملکرد بهتری نسبت به Deep Q-Learning از خود نشان داد. علت این موضوع را می‌توان در این نکته جستوجو کرد که آونگ معکوس دورانی محیطی ذاتاً پیوسته و بسیار ناپایدار است و کوچکترین تغییر در ورودی آن می‌تواند موجب برهم خوردن تعادل و سقوطش شود.

در هنگام تست روی سازه‌ی فیزیکی، Deep Q-Learning به بازدهی بالاتری رسید. به نظر می‌رسد این موضوع نه به خاطر برتری Deep Q-Learning نسبت به Deep Deterministic Policy Gradient بلکه به علت حجم محاسبات کمتر آن باشد. قابل ذکر است که در Deep Deterministic Policy Gradient ، شبکه‌ی Actor و Critic همزمان آموزش می‌بینند درحالی که در روش Deep Q-Learning تنها یک شبکه برای آموزش دادن وجود دارد. انتظار می‌رود که در صورت کوتاه‌تر کردن فرایند آموزش شبکه‌های الگوریتم Deep Deterministic Policy Gradient از طریق اصلاح معماری شبکه‌های عصبی آن و یا استفاده از پردازنده‌های قوی‌تر، این عیب الگوریتم برطرف شده و بازدهی آن در محیط واقعی نیز، مانند محیط شبیه‌سازی، از الگوریتم Deep Q-Learning پیشی‌گیرد.

۱۰- پیشنهادهایی جهت بهبود نتایج

در این بخش، با توجه به فعالیت‌های انجام شده در این پروژه و دانش به‌دست آمده، برخی پیشنهادها که احتمالا موجب بهبود عملکرد شبکه می‌شود را عنوان می‌نماییم:

۱- یکی از ارکان ضروری جهت اجرای کامل و بهینه‌ی آموزش در فرایندهای مبتنی بر یادگیری، داشتن توان محاسباتی مناسب است. کلیه‌ی فرایندهای آموزش اجرا شده در این پروژه به کمک یک جی‌پی‌یو[1] Nvidia GTX-1050 انجام شد. همچنین، به علت لزوم برقراری سریع ارتباط با سیستم واقعی یا محیط شبیه‌ساز، امکان استفاده از سیستم‌های پردازش ابری[2] مانند Google Colab وجود نداشت. این موضوع باعث شد تا نتوان بیشتر فرایندهای آموزش را تا نقطه‌ی بهینه ادامه داد. انتظار می‌رود با اجرای روش‌ها به کمک پردازنده‌های قوی‌تر و در طی فرایندهای طولانی‌تر آموزش، عمکلرد کنترلر مبتنی بر یادگیری تقویتی بهبود زیادی داشته باشد.

۲- در این پروژه، مکان، سرعت و شتاب زاویه‌ای آونگ به عنوان State ورودی در نظر گرفته شدند. انتظار می‌رود که با وارد کردن مشتقات مرتبه بالاتر مکان زاویه‌ای آونگ در کنار مشتقات مرتبه اول و دوم به سیستم، شاهد بهبود عملکرد آن باشیم. علت این موضوع این است که با وارد کردن تعداد بیشتری از مشتقات، اطلاعات الگوریتم در مورد وضعیت پیشین سیستم، پیش‌بینی رفتار آتی آن و احتمال گرفتن تصمیم‌های صحیح بیشتر می‌شود.

[1] Graphical Processing Unit (GPU)
[2] Cloud Computing

۳- متاسفانه، به علت کمبود وقت، امکان بررسی تاثیر تمامی پارامترهای اساسی[1] الگوریتم‌های Deep Q-Learning و DDPG، به مانند اندازه‌ی Replay Memory و یا مدت زمان بین بروزرسانی‌های متوالی Target Network توسط Policy Network بر عملکرد کنترلر یادگیری تقویتی میسر نشد. احتمال می‌رود که با تغییر این مقادیر بتوان به نتایج بهتر دست‌یافت.

۴- بهینه‌سازی معماری شبکه‌های عصبی یکی از مواردی است که انتظار می‌رود تاثیر قابل توجهی بر عملکرد هر دو الگوریتم Deep Q-Learning و DDPG داشته باشد. متاسفانه تاکنون الگوریتم مشخصی برای اصلاح معماری شبکه‌های عصبی معرفی نشده است اما استفاده از روش‌هایی مانند به‌تصویر کشیدن گرادیان بازگشتی و توزیع خروجی هر لایه‌ی شبکه‌ی عصبی به کمک Tensor Board می‌توانند اطلاعات مفیدی در ارتباط با تاثیر بخش‌های مختلف شبکه‌ی عصبی در اختیارمان قرار دهند. با استفاده از این داده‌ها می‌توان معماری شبکه را تا حدودی ارتقاء داد به طوری که ضمن ساده‌تر شدن، بازده عملکرد الگوریتم یادگیری تقویتی افت نکند.

۵- در مقاله‌ی ، به‌جای شبکه‌های عصبی Feed Forward، از لایه‌های کانولوشن[2] تک بعدی جهت پردازش ورودی‌های شبکه استفاده شده است. عملکرد الگوریتم یادگیری تقویتی با استفاده از این معماری در محیط شبیه‌سازی، نسبت به قبل پیشرفت چشمگیری داشته است. این موضوع تا حدی قابل پیش‌بینی است زیرا شبکه‌های عصبی کانولوشنی[3] را می‌توان به صورت شبکه‌های Feed Forward درنظر گرفت که هر نورون تنها با برخی از نورون‌های لایه‌ی قبلی ارتباط دارد. این موضوع، ضمن بدتر نکردن عمکلرد شبکه نسبت به شبکه‌ی Feed Forward متناظرش، تعداد پارامترهای آن را کاهش می‌دهد که باعث می‌شود شبکه قدرت تعمیم[4] بهتر و سرعت آموزش بیشتری داشته باشد. لذا، جایگزین کردن معماری عنوان شده برای شبکه‌ی عصبی با یک شبکه‌ی کانولوشنی به احتمال زیاد موجب بهبود فرایند کنترل آونگ می‌شود.

۶- یکی دیگر از الگوریتم‌های حوزه‌ی یادگیری تقویتی گسسته، الگوریتم Double Deep Q-Learning (DDQN) نام دارد. این الگوریتم معرفی شد تا چندی از مشکلات روش Q-Learning از جمله مشکل بیش از حد برآورد کردن Action Valueها در Q-Learning و Deep Q-Learning را برطرف سازد. به علت کمبود وقت، امکان تست عملکرد این الگوریتم میسر نشد اما پیشنهاد می‌شود جهت بهبود عملکرد روش گسسته، این روش نیز در کنار روش Deep Q-Learning امتحان گردد.

۷- متاسفانه به علت کمبود وقت، نتوانستیم عملکرد توابع پاداش[5] مختلف را بر کارکرد الگوریتم‌ها بسنجیم. به قطع امتحان کردن توابع پاداش مختلف می‌تواند موجب بهبود عملکرد سیستم شود.

---

[1] Hyperparameter
[2] Convolution
[3] Convolutional Neural Networks
[4] Generalization
[5] Reward

۱۱- جمع‌بندی و نتیجه‌گیری

در این پروژه به ساخت و شبیه‌سازی سیستم آونگ معکوس دورانی پرداخته شد. همچنین، عملکرد دو الگوریتم مهم حوزه‌ی یادگیری تقویتی پیوسته و گسسته (Deep Q-Learning و Deep Deterministic Policy Gradient) در شبیه‌سازی و واقعیت سنجیده شد. با وجود پیچیده‌تر و طولانی‌تر بودن فرایند آموزش Deep Deterministic Policy Gradient، این الگوریتم توانست تا حدودی از Deep Q-Learning در محیط شبیه‌سازی پیشی گیرد. علت این موضوع را می‌توان در پیوستگی ذاتی ورودی‌های آونگ معکوس دورانی و ناپایداری آن جستوجو کرد. در مقابل، الگوریتم Deep Q-Learning به علت حجم کمتر محاسبات، عملکرد بهتری در کنترل سازه‌ی آونگ معکوس دورانی داشت. نتایج به‌دست آمده از الگوریتم‌های بررسی‌شده در کنترل آونگ معکوس دورانی، نشان‌دهنده‌ی توانایی این الگوریتم‌های حوزه‌ی یادگیری تقویتی در کنترل سیستم‌های غیرخطی و پیچیده است.

فعالیت‌های صورت گرفته در این پروژه، تنها گوشه‌ی کوچکی از پتانسیل‌های فراوان تحقیقاتی بر روی سازه‌ی آونگ معکوس دورانی می‌باشد. امید است فعالیت‌های صورت گرفته تاکنون، بتواند بستری مناسب برای پژوهش‌های بیشتر درمورد عملکرد الگوریتم‌های هوشمند روی آونگ معکوس دورانی، به عنوان یک مسئله‌ی معیار در حوزه‌ی مهندسی کنترل، شود.

مراجع

[1] J L Duart, B Montero, P A Ospina and E González, "*Dynamic Modeling and Simulation of a Rotational Inverted Pendulum*", Journal of Physics: Conference Series, Volume 792, VIII International Congress of Engineering Physics 7–11 November 2016, Mérida, Yucatán, Mexico.

[2] Wudai Liao, Zhengbo Liu, Shengjun Wen, Shuhui Bi, Dongyun Wang, "*Fractional PID based stability control for a single link rotary inverted pendulum*", 2015 International Conference on Advanced Mechatronic Systems (ICAMechS).

[3] Jia-JunWang, "*Simulation studies of inverted pendulum based on PID controllers*" Simulation Modelling Practice and Theory, Volume 19, Issue 1, January 2011, Pages 440-449.

[4] Krishen, J., Becerra, V.M., "*Efficient fuzzy control of a rotary inverted pendulum based on LQR mapping*", 2006 IEEE International Symposium on Intelligent Control.

[5] Minho Park, Yeoun-Jae Kim, Yeoun-Jae KimJu-Jang Lee, "*Swing-up and LQR stabilization of a rotary inverted pendulum*", Artificial Life and Robotics, 2013.

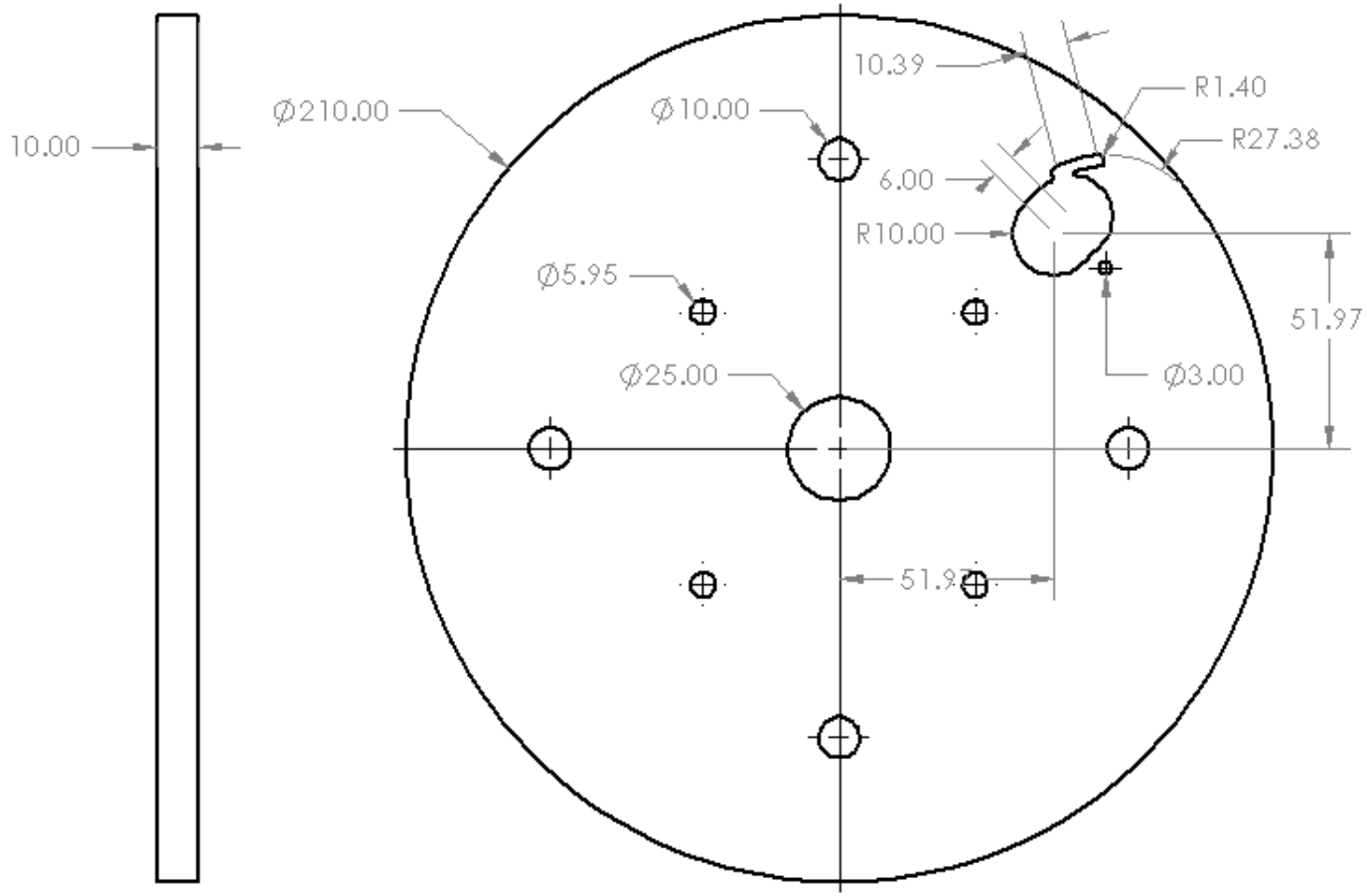
[6] Iraj Hassanzadeh, Saleh Mobayen, "*PSO-Based Controller Design for Rotary Inverted Pendulum System*", Journal of Applied Sciences, 2008.

[۷] آ. گشتاسبی, "*ساخت و کنترل PID پاندول معکوس دورانی*" , ۱۳۹۹.

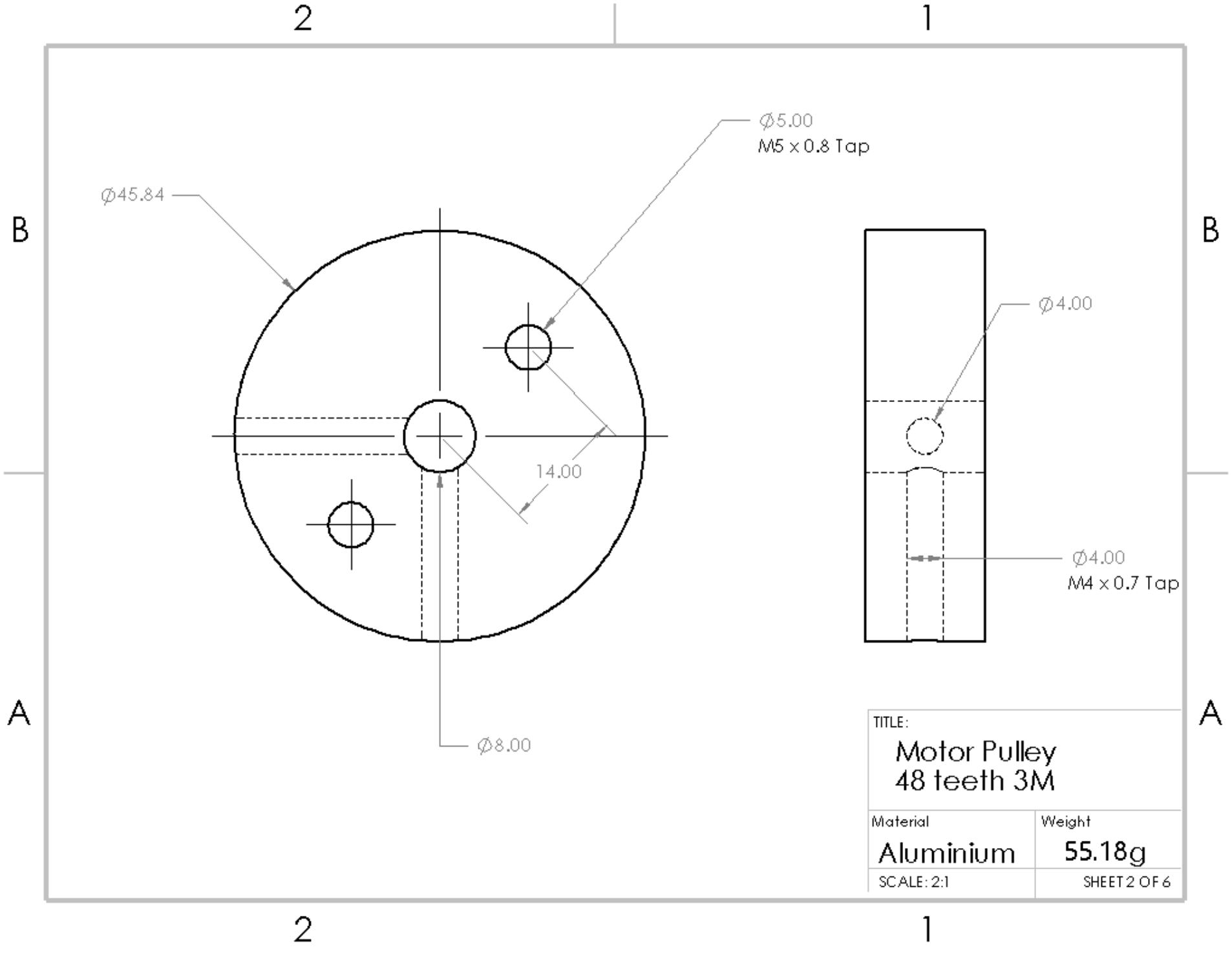
[۸] م. گرجی, "*کنترل پاندول معکوس دورانی به کمک روش‌های هوشمند*" , ۱۳۹۹.

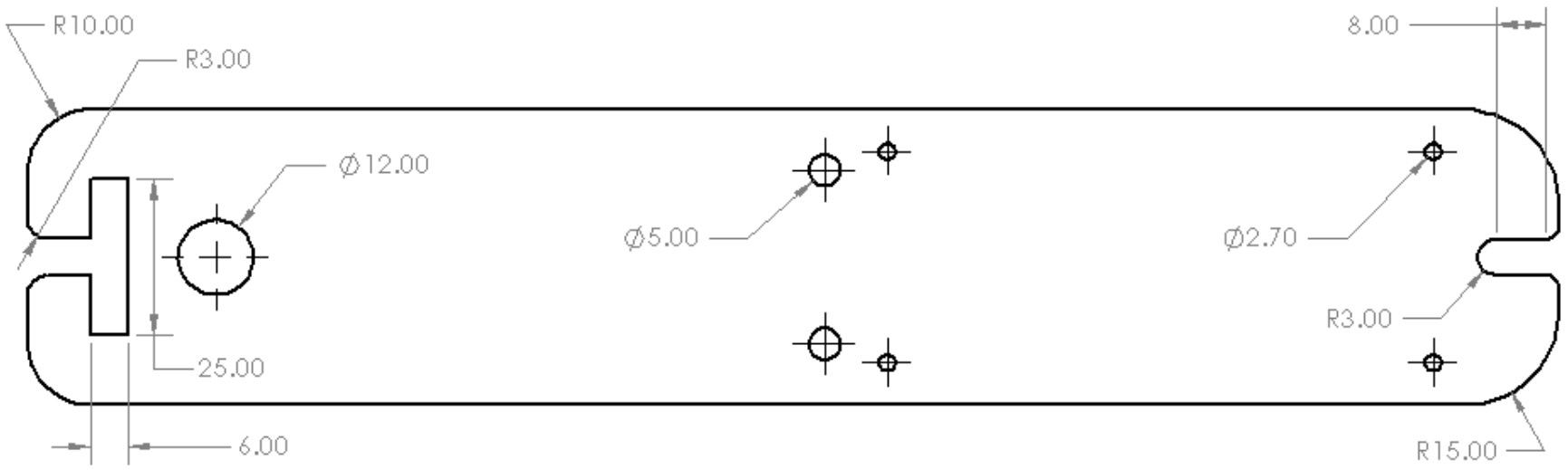
پیوست‌ها

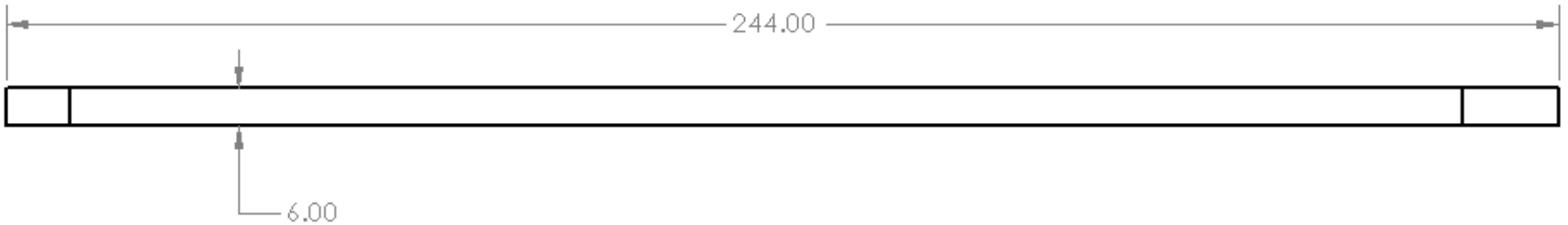
پیوست ۱: نقشه کارگاهی قطعات تغییر یافته در آونگ معکوس دورانی



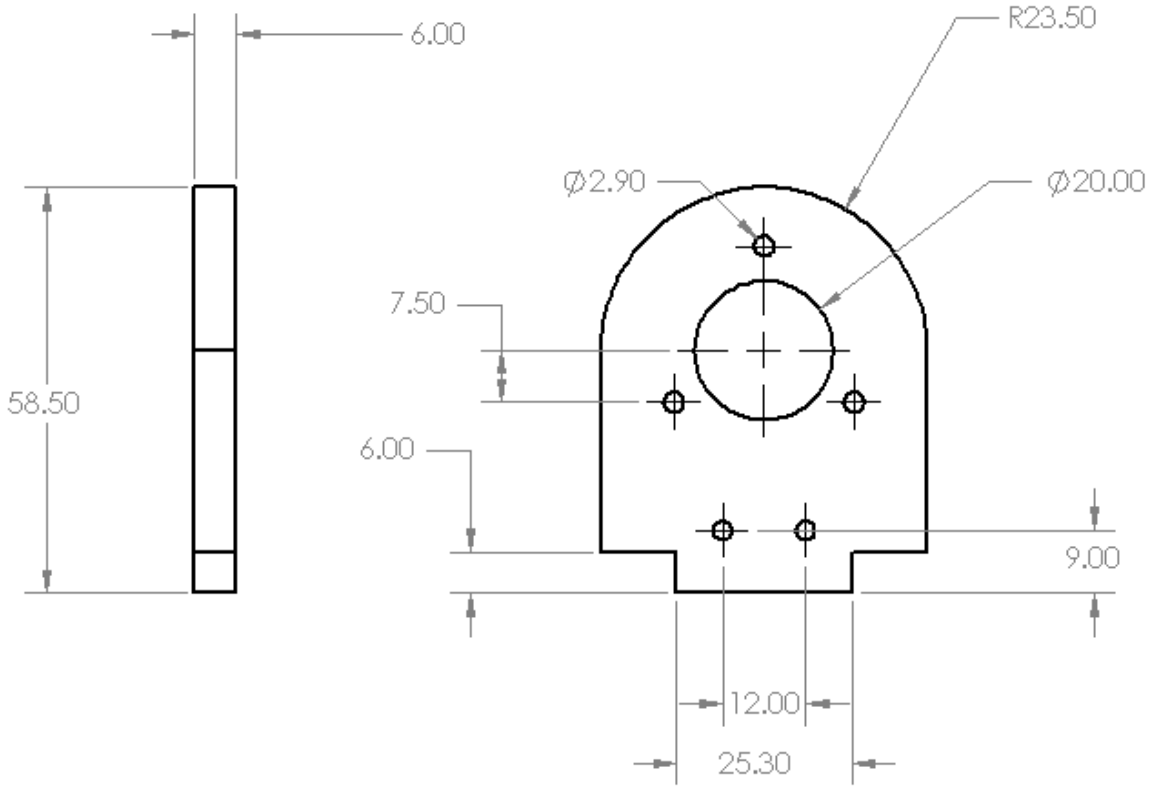
TITLE: Top Plate	
Material: Plexiglass	Weight: 332.41g
SCALE: 3:5	SHEET 1 OF 6


2
1
B
A
Ø5.00
M5 x 0.8 Tap
Ø45.84
14.00
Ø8.00
Ø4.00
Ø4.00
M4 x 0.7 Tap
TITLE:
Motor Pulley
48 teeth 3M
Material
Aluminium
Weight
55.18g
SCALE: 2:1
SHEET 2 OF 6



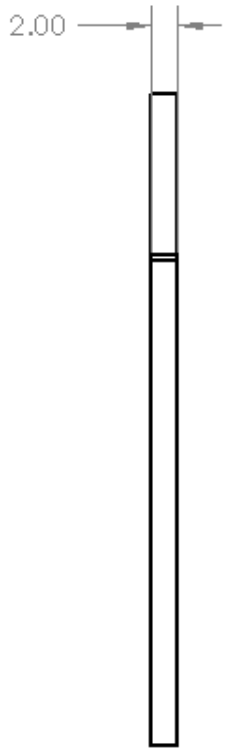


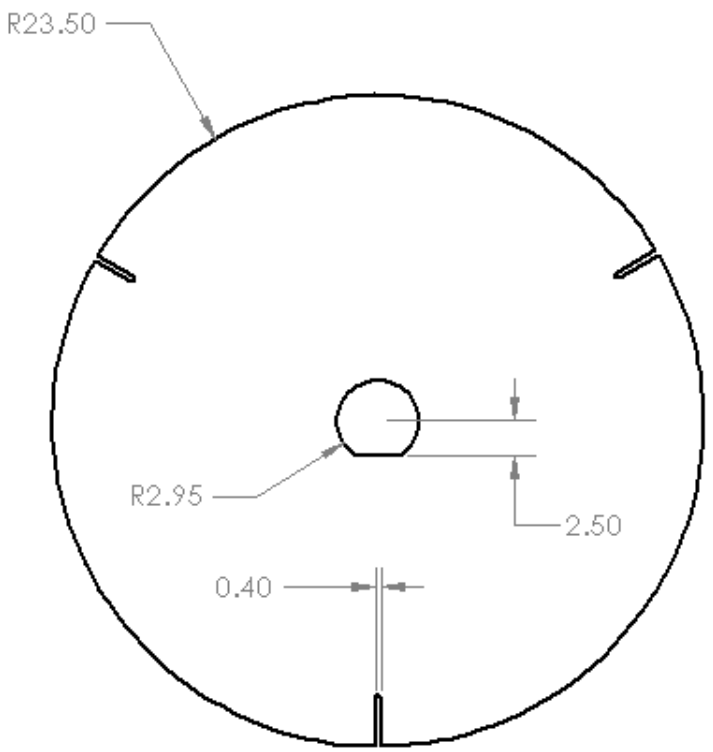
TITLE: Arm	
Material: Plexiglass	Weight: 76.94g
SCALE: 1:1	SHEET 3 OF 6



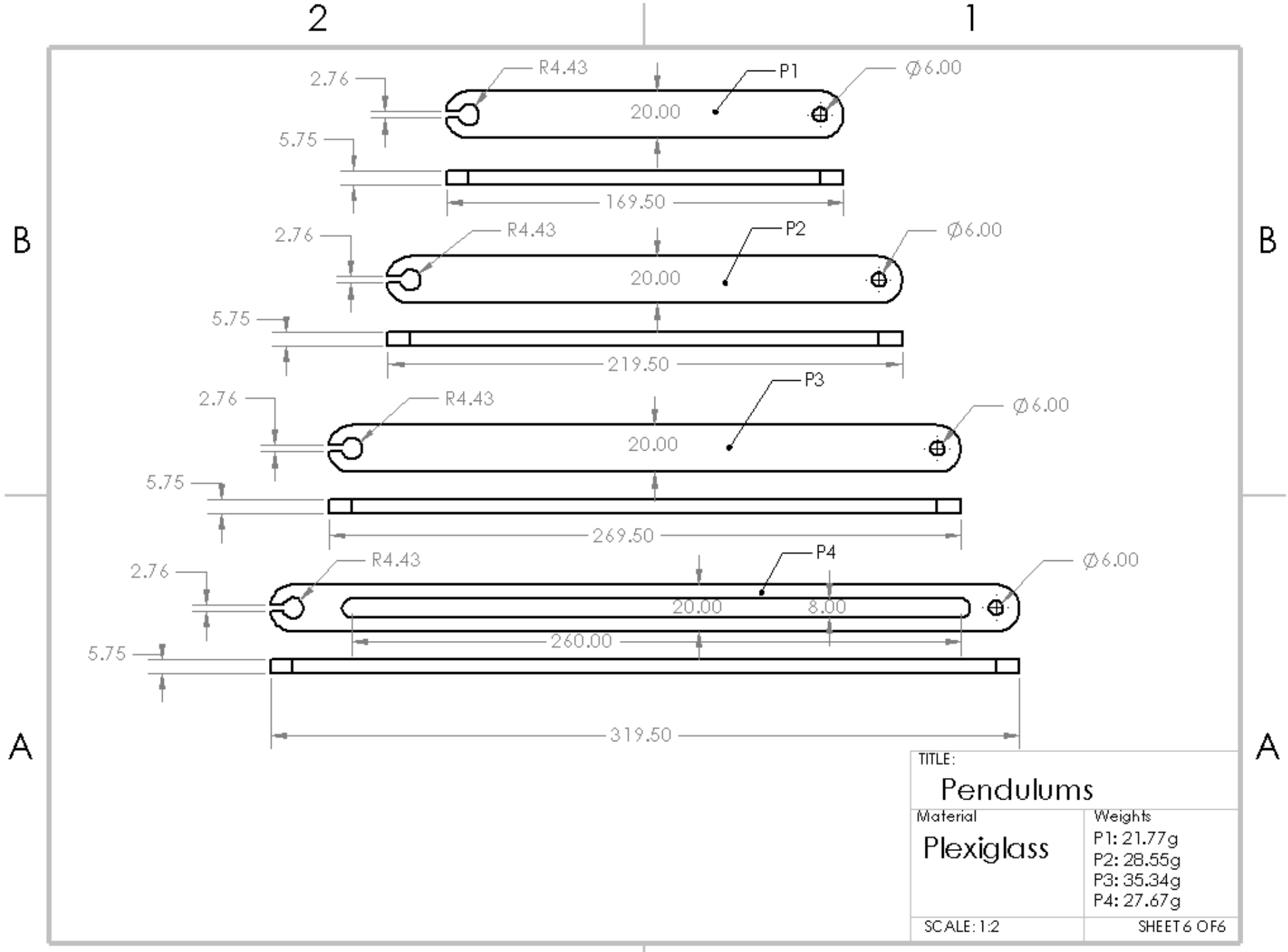
TITLE:
Pendulum Encoder Mounting

Material	Weight
Plexiglass	14.42g
SCALE: 1:1	SHEET 4 OF 6

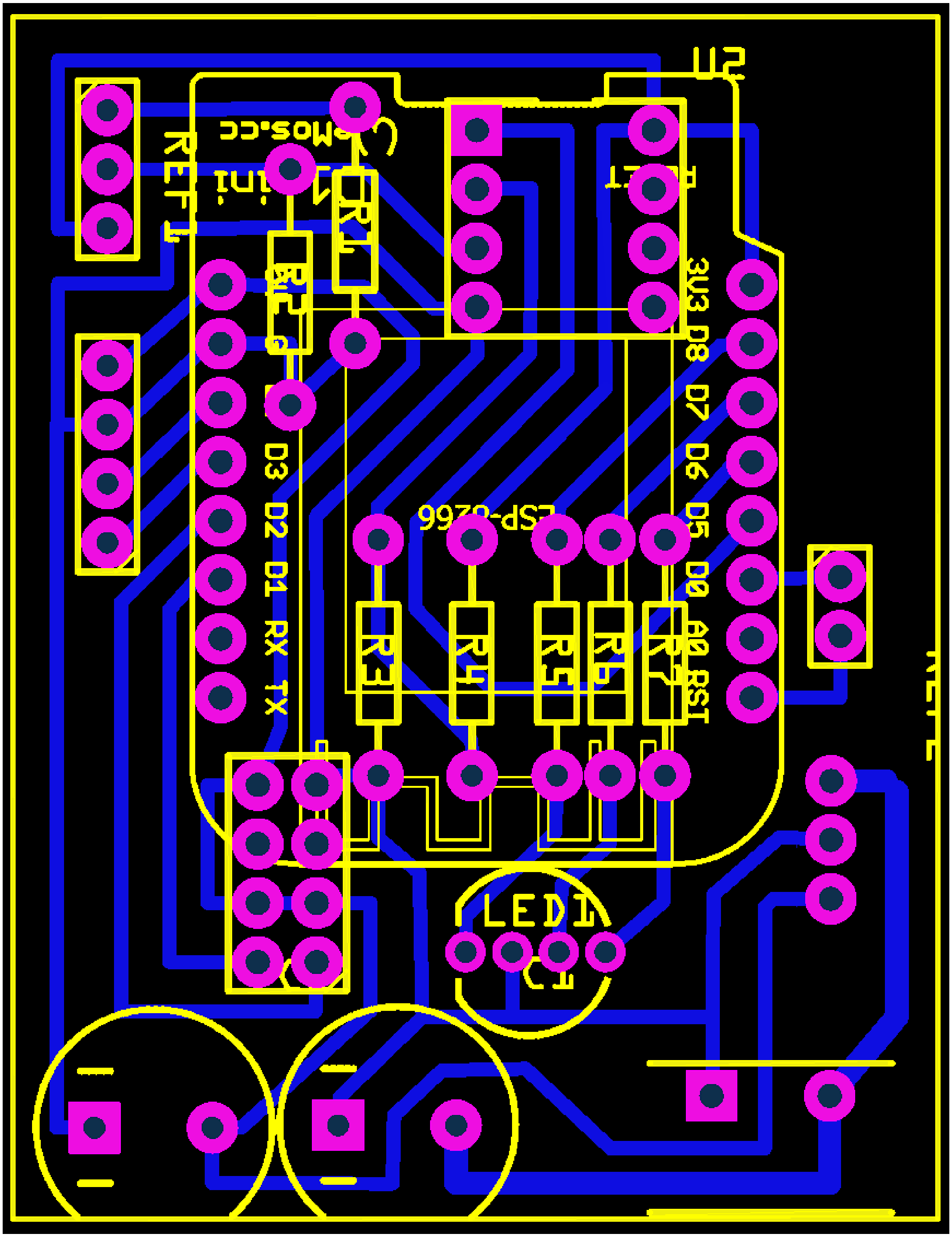


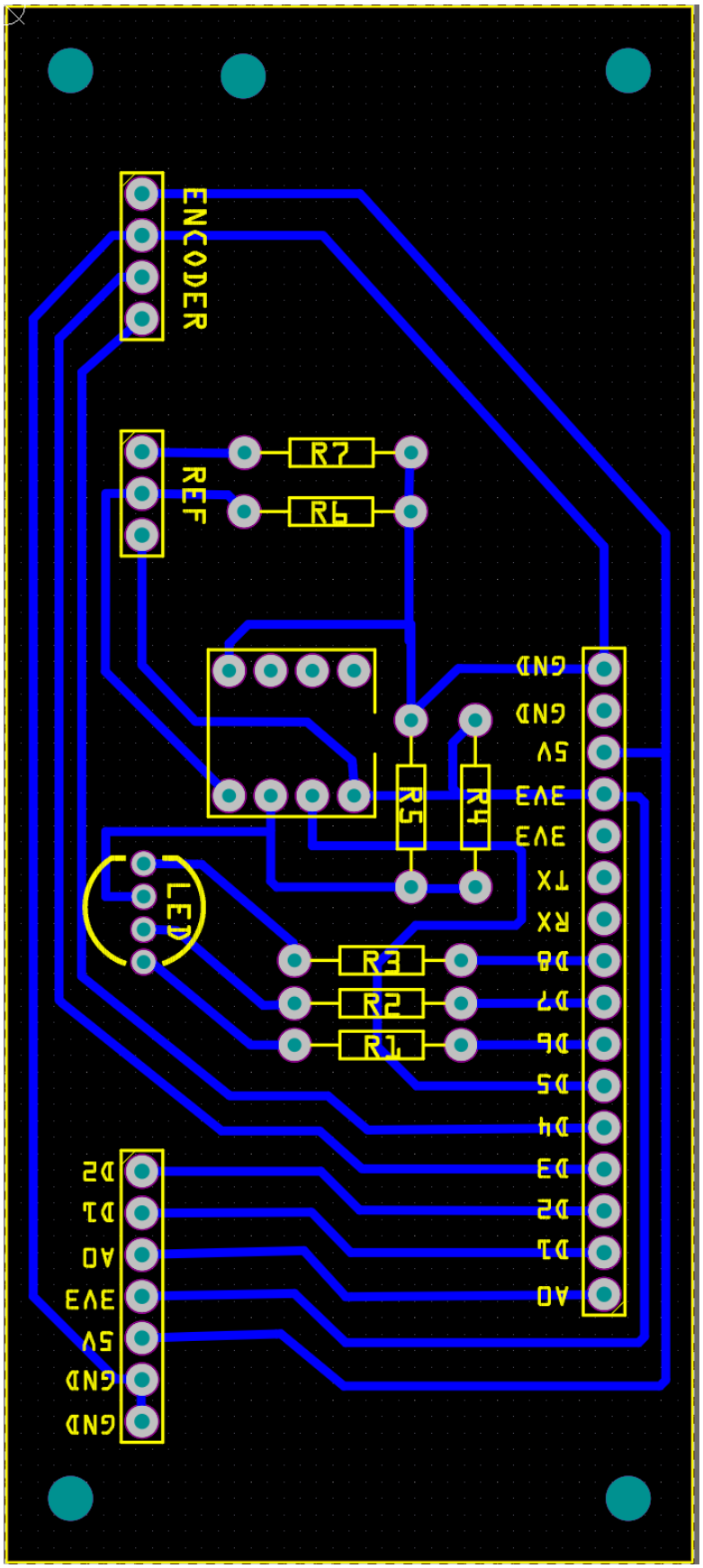


TITLE: Encoder Calibration Disk	
Material: Plexiglass	Weight: 4.02g
SCALE: 2:1	SHEET 5 OF 6


2
1
B
A
R4.43
P1
Ø6.00
2.76
20.00
5.75
169.50
P2
219.50
P3
269.50
P4
8.00
260.00
319.50
TITLE:
Pendulums
Material
Plexiglass
Weights
P1: 21.77g
P2: 28.55g
P3: 35.34g
P4: 27.67g
SCALE: 1:2
SHEET 6 OF6

پیوست ۲: تصاویر PCBهای طراحی شده در محیط Altium Designer


REF1
R1
D3 D2 D1 RX TX
3V3 D8 D7 D6 D5 D0 A0 RST
R3
R4
R5
R6
R7
LED1


ENCODER
REF
R7
R6
R5
R4
R3
R2
R1
LED
GND
GND
5V
3V3
3V3
TX
RX
D8
D7
D6
D5
D4
D3
D2
D1
A0
D2
D1
A0
3V3
5V
GND
GND

University of Tehran
College of Engineering
School of Mechanical Engineering

Control of the Rotary Inverted Pendulum Using Reinforcement Learning Algorithms

Senior Design Project

Mechanical Engineering B.Sc.

Autor: Arash Hatefi

Supervisor: Dr. Masoud Shariatpanahi

February 2021