

هدایت و کنترل یکپارچه کانال فراز موشک آشیانه‌یاب زمین به هوا با استفاده از کنترل

عصبی بهینه

تاریخ دریافت: ۱۴۰۱/۰۹/۰۱

تاریخ پذیرش: ۱۴۰۱/۱۲/۰۷

محمد مهدی سوری^۱، سیدحسین ساداتی^۲

۱- دانشجوی دکتری، دانشکده مهندسی مکانیک، دانشگاه صنعتی خواجه نصیرالدین طوسی، تهران.

۲- دانشیار، دانشکده مهندسی مکانیک، دانشگاه صنعتی خواجه نصیرالدین طوسی، تهران. sadati@kntu.ac.ir

چکیده

شکل مرسوم سیستم هدایت و کنترل موشک از سه زیرسیستم ناوبری، هدایت و کنترل تشکیل می‌شود. این مقاله به تشریح روند طراحی و شبیه‌سازی عملکرد کنترل‌کننده عصبی بهینه می‌پردازد که به منظور هدایت موشک در یک مسئله دوبعدی کمینه‌سازی زمان برخورد و فاصله تا هدف ایجاد شده است. در طراحی کنترل‌کننده عصبی بهینه، ابتدا نوع کلاسیک و معمول یک کنترل‌کننده عصبی بهینه شامل دو شبکه عصبی کم‌عمق شناساگر و کنترل‌کننده، طراحی شده و با انجام شبیه‌سازی نشان داده شده است که عملکرد این کنترل‌کننده در مسئله مورد بررسی مطلوب نیست؛ بنابراین با جایگزینی شبکه‌های کم‌عمق شناساگر و کنترل‌کننده، با نوع عمیق و بهره‌گیری از مفاهیم یادگیری تقویتی، نتایج شبیه‌سازی‌های انجام شده بهبود عملکرد این نوع کنترل‌کنترل را نشان می‌دهد. در این تحقیق مدل یکپارچه موشک با تلفیقی از شبکه عصبی یادگیری عمیق با الگوریتم‌های بهینه‌سازی انجام شده و استفاده از کنترل شبکه عصبی و الگوریتم‌های بهینه‌سازی باعث افزایش دقت برخورد و کاهش زمان پرواز شد.

واژه‌های کلیدی: موشک، هدایت و کنترل یکپارچه، کنترل بهینه، کنترل عصبی، شبکه عصبی عمیق

۲۵

سال ۱۳ - شماره ۱

بهار و تابستان ۱۴۰۲
شهره علی

دانش و فناوری هوا فضا



Integrated guidance and control of the surface-to-air homing missile pitch channel using optimal neural network

Mohammad Mahdi Soori¹, Seyed Hossein Sadati²

1- PhD student, Mechanical Engineering Department, K. N. Toosi University of Technology, Tehran

2- Associate Professor, Mechanical Engineering Department, K. N. Toosi University of Technology, Tehran. sadati@kntu.ac.ir

Abstract

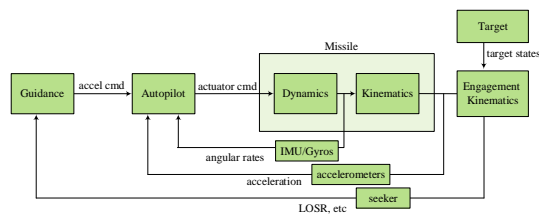
Conventionally, the missile guidance and control system consists of three subsystems: navigation, guidance, and control. The task of these sub-systems is to calculate the deviation of the guided vehicle from the desired path so as to determine the appropriate movement or acceleration to compensate for the deviation. This paper describes the process of designing and simulating the performance of the optimal neural controller, which was created in order to guide the missile in a two-dimensional problem of minimizing the collision time and the distance from the target. In the design of the optimal neural controller, first the classical optimal neural controller including two Multilayer Perceptron (MLP) neural networks, the identifier, and the controller was designed and through simulation it was shown that the performance of this controller is not satisfactory. Therefore, by replacing the estimator MLP networks and controller with the deep type network, along with the use of the concepts of reinforcement learning, a quite improved performance was demonstrated through simulation. In this research, the integrated rocket model was made by integrating deep learning neural network with optimization algorithms, and the use of neural network control and optimization algorithms increased collision accuracy and reduced flight time.

Keywords: missile, integrated guidance and control, optimal control, neural network control, deep neural network



۱. مقدمه

عملکردهای هدایت، ناوبری و کنترل^۱ برای همه انواع وسایل نقلیه هوایی و فضایی از جمله موشک‌ها، حیاتی است. در عمل، این توابع به‌صورت سری برای مانور یک وسیله نقلیه باهم کار می‌کنند. ابتدا هر یک از توابع GNC تعریف می‌شود. ناوبری با استفاده از اندازه‌گیری که حس‌گرها انجام می‌دهند، وضعیت فعلی موشک (یعنی بردار حالت) را تعیین می‌کند. هدایت، شتاب و جهت موشک را برای مسیر مطلوب به سمت هدف یا مسیر وضعیت، برای زمان فعلی و آینده مشخص می‌کند. کنترل، شتاب و جهت موشک به سمت هدف را ردیابی می‌کند. در این مقاله تمرکز بر مسئله هدایت و کنترل موشک است. فرض بر این است که ناوبری به‌درستی عمل می‌کند و در نتیجه باعث می‌شود مسیر پرواز را هدایت و کنترل تعیین کنند. هدایت دستوراتی را با استفاده از بردار حالت ناوبری به کنترل ارائه می‌دهد، سپس کنترل‌کننده به عملگر موشک فرمان می‌دهد تا وضعیت ناوبری دستوری که هدایت داده است را ردیابی کند. در حال حاضر مرسوم است که هدایت کاملاً جدا از کنترل (اتوپیلوت) و بالعکس توسعه یابد. تقریباً همه کتاب‌های درسی و مقاله‌های فنی این موضوع را بررسی کرده‌اند [۱]. شکل شماره ۱، نمودار بلوکی اتوپیلوت سه حلقه هدایت و کنترل موشک را نشان می‌دهد.



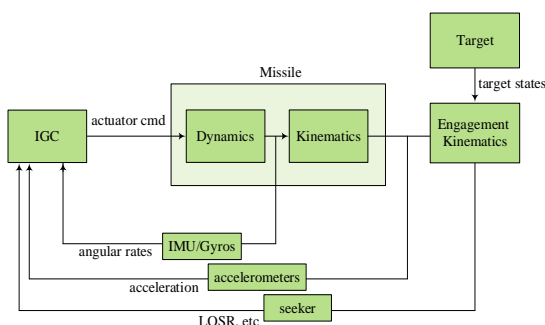
شکل ۱. نمودار بلوکی اتوپیلوت سه حلقه

برخی از الگوریتم‌های هدایت پیشرفته‌تر نه تنها به رهگیری دست می‌یابند، بلکه زاویه برخورد^۲ رهگیری موشک در هنگام برخورد را نیز کنترل می‌کنند. باین‌حال، همه این الگوریتم‌ها ریشه در مفهوم مثلث برخورد^۳ دارند، که تغییر خط دید^۴ بین رهگیر و هدف را به حداقل می‌رساند. در ساختار چندحلقه، هدایت

با استفاده از سینماتیک درگیری، ایجاد می‌شود درحالی‌که اتوپیلوت دینامیک بدنه را تثبیت می‌کند و شتاب ارائه‌شده توسط هدایت را دنبال می‌کند.

۱-۲ هدایت و کنترل یکپارچه

بر خلاف ساختار اتوپیلوت سه حلقه مرسوم، هدایت و کنترل یکپارچه (IGC^۵) چارچوبی یکپارچه است که در آن هدایت و کنترل به جای آنکه مستقل از هم باشند، به‌صورت یکپارچه درون هم در نظر گرفته می‌شود. بلوک دیاگرام IGC در شکل ۲ نشان داده شده است.



شکل ۲. بلوک دیاگرام هدایت و کنترل یکپارچه (IGC)

مزیت IGCها توانایی آن‌ها در استفاده از تعاملات بین زیرسیستم‌های هدایت و کنترل است. IGC قصد دارد با بهره‌گیری از هم‌افزایی بین فرایندهای هدایت و کنترل، عملکرد موشک را افزایش دهد. قرار دادن G&C در یک سیستم واحد IGC پتانسیل بهینه‌سازی آن را بهبود می‌بخشد. زیرا بهینه‌سازی پارامترها می‌تواند مستقیم انجام شود. توابع هزینه شامل پارامترهای کلیدی عملکرد مانند سرعت نسبی نزدیک شدن موشک و هدف، زاویه خط‌دید، زاویه برخورد و پارامترهای بسیاری که به‌آسانی در دسترس اتوپیلوت نیستند، اکنون به‌طور مستقیم در دسترس هستند.

در رویکرد متداول، قانون هدایت هیچ اطلاعی از میزان چرخش یا شتاب وارد شده به موشک را ندارد، در عوض هدایت فقط موقعیت نسبی و سرعت درگیری را می‌داند. با کاهش محدوده هدف، تغییرات کوچک در هندسه به دستورات شتاب بزرگ منجر می‌شود که می‌تواند از محدوده عملکرد اتوپیلوت فراتر رود. علاوه‌براین، اتوپیلوت نمی‌تواند خود را بر

اساس سینماتیک درگیری نسبی تنظیم کنند، زیرا این اطلاعات را دریافت نمی‌کند.

در نتیجه سیستم‌های G&C معمولی برای بهبود پایداری به کوچک کردن ثابت زمانی اتوپیلوت تا حد ممکن متکی هستند. ثابت زمانی اتوپیلوت، فاصله از دست‌دهی^۶ تا هدف را در سیستم‌های G&C معمول طراحی می‌کند [۲].

تحقیقات در زمینه هدایت و کنترل یکپارچه با لین^۷ و یوئه^۸ در سال ۱۹۸۸ آغاز شد. آن‌ها شرکت GNC آمریکایی را تأسیس کردند. در مقاله ۱۹۹۲ آن‌ها، IGC با استفاده از رویکرد نظری بازی^۹ تدوین شد [۳]. آن‌ها از رویکرد تئوری بازی استفاده کردند که در آن زمان به‌طور گسترده مورد مطالعه قرار گرفت. کلوتیر^{۱۰}، یکی از نویسندگان [۴]، یک تنظیم‌کننده غیرخطی را که از معادله ریکاتی وابسته به حالت (SDRE^{۱۱}) در [۵] تجزیه و تحلیل کرد. این روش سپس برای توسعه قانون هدایت در [۶] و اتوپیلوت در [۷] به کار گرفته شد. با موفقیت کلوتیر در استفاده از SDRE، منون^{۱۲} و همکاران در [۸] و پالمبو^{۱۳} در [۹]، ترمیم تکنیک SDRE را به سمت یک سیستم IGC انتخاب کردند. در [۹]، پلومبو رویکرد مشابهی دارد، اما IGC خود را بر اساس یک اتوپیلوت معکوس پویا همراه با یک سیستم هدایت مطلوب پایه‌گذاری می‌کند. این مطالعه شامل یک آزمایش رفت و برگشت مونت کارلو^{۱۴} با پویایی خطای اندازه‌گیری واقع‌بینانه بود. او نتیجه می‌گیرد که معادله دیفرانسیل ریکاتی وابسته به حالت

IGC به‌طور میانگین فاصله بین ۲۳ تا ۳۱ درصد را از دست می‌دهد. در سال ۲۰۰۴، منون، سوئیدوک^{۱۵} و وادی^{۱۶}، تکنیک کنترل غیرخطی و زمان گسسته را در [۱۰] مقایسه کردند. این تکنیک‌ها روش خطی‌سازی بازخورد زمان‌گسسته، روش معادله ریکاتی وابسته به حالت گسسته و یک روش بازگشت به عقب زمان‌گسسته بود. در سال ۲۰۰۶، یک رویکرد نظری بازی توسط منون، وادی و اولمایر^{۱۷} در [۱۱] ارائه شد. در آن رویکرد، یک سیستم فعال‌شده با جرم متحرک مدل شد. سیستم هدایت و کنترل یکپارچه افق محدود به‌عنوان یک بازی دیفرانسیل محدودشده توسط بدترین اختلالات ارائه شد. این مطالعه

امکان‌سنجی این مفهوم را ثابت کرد، اما با یک سیستم پایه دیگر مقایسه نشد. همچنین در سال ۲۰۰۶، رویکرد برگشت مجدد^{۱۸} توسط هوانگ^{۱۹} و تهک^{۲۰} در [۱۲] ارائه شد. به‌طور خلاصه، IGC با روش‌های مختلفی از جمله رویکرد نظریه بازی، معادله ریکاتی وابسته به حالت و بازگشت به عقب بررسی شده است. رویکرد بعدی نسبت به IGC با استفاده از SMC در [۱۳] را هارل^{۲۱} و بالاکریشنان^{۲۲} با استفاده از کنترل حالت لغزشی مرتبه دوم ترمینال^{۲۳} (TSM) ارائه کردند.

در سال ۲۰۱۹ وانگ^{۲۴} و همکاران برای اینکه موشک به توانایی حمله همه‌جانبه دست یابد [۱۴]، یک روش هدایت و کنترل یکپارچه با زاویه ضربه محدود ارائه کرد. برای بهبود توانایی آسیب رساندن به هدف، او و همکاران [۱۵]، یک قانون هدایت و کنترل یکپارچه با محدودیت زاویه ضربه برای مقابله با مشکل رهگیری اهداف مانور ناشناخته طراحی کرد. برای مقابله با محدودیت‌های اشباع محرک در سیستم‌های واقعی، ما^{۲۵} و همکاران [۱۶] یک قانون هدایت و کنترل یکپارچه را با استفاده از کنترل سطح پویا، کنترل پس‌گام^{۲۶} و شبکه عصبی تطبیقی بررسی کرد. و میشل و استشل کنترل مود لغزشی را برای مدل صفحه‌ای یکپارچه به‌طور کامل بررسی کردند [۱۷].

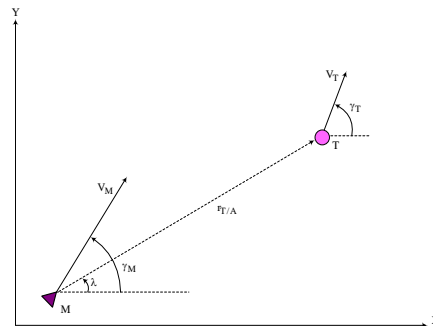
در روش‌های سنتی هدایت و کنترل موشک، سیستم‌های هدایت و کنترل به‌عنوان دو فرایند متفاوت در نظر گرفته می‌شوند. سیستم هدایت به‌عنوان یک حلقه بیرونی طراحی شده است که شتابی را ایجاد می‌کند که توسط اتوپیلوت حلقه داخلی ردیابی می‌شود که معمولاً بدون در نظر گرفتن اطلاعات موقعیت و سرعت بین موشک و هدف طراحی می‌شود [۱۸، ۱۹]. با این حال، کوچک‌تر شدن فاصله نسبی بین موشک و هدف و تغییر سریع هندسه نسبی ممکن است به کاهش عملکرد سیستم و حتی شکست روش طراحی جداسازی منجر شود. برای اجتناب از این کاستی‌ها، طراحی هدایت و کنترل یکپارچه توسط محققان داخلی و بین‌المللی بررسی شده است [۲۰].



۲- مدل سازی ریاضی

سناریوی درگیری موشک-هدف شامل تلاش موشک برای رهگیری یک هدف از طریق تغییر جهت مسیر است. در حین هدایت آشیانه یابی، از حسگرهای داخل موشک برای هدایت تا زمانی که برخورد رخ دهد استفاده می شود. شرایط اولیه این سناریو شامل سه فرض اصلی است. یک: هدایت میانه مسیر موفقیت آمیز است. دو: سرعت موشک و هدف در مسیر برخورد نزدیک به هم است. سه: در لحظه برخورد سرعت نسبی موشک-هدف صفر خواهند شد. هندسه این سناریوی درگیری در شکل ۳ نشان داده شده است.

هدف از این مقاله طراحی کنترلی مناسب به منظور رهگیری دقیق هدف خواهد بود. از این رو ابتدا به بیان مسئله درگیری موشک-هدف که شامل تمام موضوعات مورد نیاز برای مدل سازی دقیق از جمله سینماتیک درگیری، دینامیک موشک، مدل هدایت و کنترل یکپارچه، استراتژی رهگیری و هدایت است پرداخته می شود.



شکل ۳. سینماتیک درگیری [۲۱]

هندسه درگیری در شکل ۳ در نظر گرفته می شود که در آن بردار خط دید، موقعیت هدف را نسبت به موشک مشخص می کند. اگر از سیستم مختصات ثابت اندازه گیری استفاده شود، رابطه نسبی بین بردارهای موقعیت موشک r_M و هدف r_T به صورت زیر نوشته می شود [۲۱].

$$r_T = r_M - r_{T/M} \quad (1)$$

با مشتق گیری از رابطه (۱) نسبت به زمان، رابطه نسبی سرعت ها و شتاب ها نیز به دست می آیند. همچنین علاوه بر مختصات اصلی که بیان شد، برای ردیابی بهتر یک سیستم مختصات چرخشی ω, n, r برای این سینماتیک مورد نیاز است. این سیستم مختصات روی موشک ثابت است و فرض بر این است که محور اصلی آن همیشه در امتداد خط دید قرار دارد. بنابراین این سیستم نسبت به مرجع اینرسی ثابت Z, Y, X می چرخد. در نتیجه معادله های سرعت و شتاب نسبی به صورت معادله های (۲ و ۳) می شوند [۲۱].

$$v_T - v_M = \dot{r}\hat{r} + r\Omega_\omega\hat{n} \quad (2)$$

$$a_T - a_M = (\ddot{r} - r\Omega_\omega^2)\hat{r} + (r\dot{\Omega}_\omega - 2\dot{r}\Omega_\omega)\hat{n} + r\Omega_r\Omega_\omega\hat{\omega} \quad (3)$$

در معادله های فوق r فاصله نسبی موشک و هدف و $\Omega_M = \dot{r}$ نرخ تغییرات زاویه خط دید موشک هدف است. همچنین Ω_ω به صورت معادله (۴) نوشته می شود [۲۱].

$$\Omega_\omega = \frac{\dot{v}_n}{r} - \frac{v_n\dot{r}}{r^2} \quad (4)$$

با ترکیب معادله های فوق و در نظر گرفتن متغیرهای حالت مناسب، معادله های فضای حالت سینماتیک درگیری به صورت معادله (۵) ارائه می شوند [۲۱].

$$\begin{bmatrix} \dot{r} \\ \dot{v}_r \\ \dot{\lambda} \\ \dot{v}_n \end{bmatrix} = \begin{bmatrix} \frac{v_r}{r} \\ \frac{v_n^2}{r} \\ \frac{v_n}{r} \\ -\frac{v_r v_n}{r} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ -1 & 0 \\ 0 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} a_{M_r} \\ a_{M_n} \end{bmatrix} + \begin{bmatrix} 0 \\ a_{T_r} \\ 0 \\ a_{T_n} \end{bmatrix} \quad (5)$$



$$\begin{pmatrix} \dot{\alpha} \\ \dot{q} \\ \dot{\theta} \end{pmatrix} = \begin{bmatrix} \frac{\bar{q}S}{mU_0} C_{za} & 1 & 0 \\ \frac{\bar{q}Sl}{I_{yy}} C_{ma} & \frac{\bar{q}Sl^2}{I_{yy}2U_0} C_{mq} & 0 \\ 0 & \frac{1}{m} \left[\frac{\bar{q}S}{m} C_{z\delta} \right] & 0 \end{bmatrix} \begin{pmatrix} \alpha \\ q \\ \theta \end{pmatrix} + \begin{bmatrix} \frac{\bar{q}S}{m} C_{z\delta} \\ \frac{\bar{q}Sl}{I_{yy}} C_{m\delta} \\ 0 \end{bmatrix} \delta \quad (6)$$

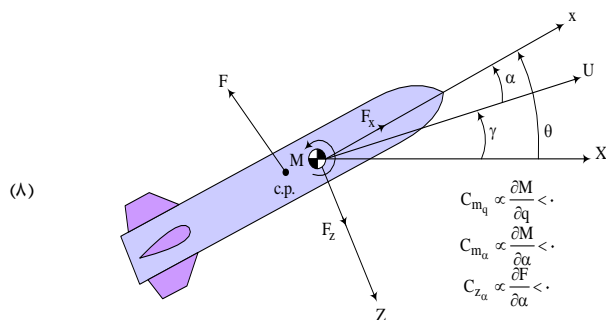
در معادله‌های فوق C_{za} و C_{ma} و C_{mq} مشتقات پایداری و $C_{z\delta}$ و $C_{m\delta}$ مشتقات کنترل هستند. مدل درگیری یکپارچه با ترکیب سینماتیک درگیری و دینامیک موشک استخراج می‌شود که واسطه بین سینماتیک و دینامیک، شتاب جانبی موشک است [۲۲].

$$a_{M,z} = \frac{\bar{q}A}{m} (C_{za}\alpha + C_{z\delta}\delta) \quad (7)$$

که در نتیجه معادلات این سیستم به فرم معادله (۸) خواهد شد [۲۲].

$$\begin{pmatrix} \dot{r} \\ \dot{v}_r \\ \dot{\lambda} \\ \dot{v}_n \\ \dot{\alpha} \\ \dot{q} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \frac{v_n^2}{r} + \frac{\bar{q}S}{m} C_{za}\alpha \sin(\lambda - \theta) \\ \frac{v_n}{r} \\ -\frac{v_n v_r}{r} + \frac{\bar{q}S}{m} C_{za}\alpha \cos(\lambda - \theta) \\ \frac{\bar{q}S}{mU_0} C_{za}\alpha + q \\ \frac{\bar{q}Sl}{I_{yy}} C_{ma}\alpha + \frac{\bar{q}Sl^2}{I_{yy}2U_0} C_{mq}q \\ q \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \frac{\bar{q}Sl}{m} C_{z\delta} \\ \frac{\bar{q}Sl}{I_{yy}} C_{m\delta} \\ 0 \end{pmatrix} \delta + \begin{pmatrix} 0 \\ a_{Tr} \\ 0 \\ a_{Tn} \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

این نمایش به شکل $\dot{x} = A(x) + Bu + \rho$ است که بردار حالت، x بردار کنترل و ρ اغتشاش است. اگرچه این معادله به طور دقیق سینماتیک درگیری را توصیف می‌کند، اما دینامیک موشک را به طور مستقیم لحاظ نمی‌کند و یک بردار کنترل غیرعملی ایجاد می‌کند. همان‌طور که در سینماتیک درگیری نشان داده شد، یک موشک برای رهگیری یک هدف در حال مانور باید شتاب a_{Mn} داشته باشد. برای موشکی که درون جو قرار دارد، چرخش بدنه مهم‌ترین عمل برای ایجاد نیروهای لازم برای شتاب دادن به موشک خنثی کردن مانور هدف است. بنابراین، دینامیک موشک چرخشی باید به طور دقیق مدل شود. در شکل ۴ دینامیک سیستم نمایش داده شده است که در آن X, Z سیستم مختصات متصل به بدنه، U محور اولیه متصل به بدنه، X محور اولیه متصل به همان چارچوب مرجع اینرسی است که در سینماتیک درگیری استفاده می‌شود. زاویه حمله α ، زاویه مسیر پرواز γ ، و زاویه اوج θ ، جهت‌گیری این سیستم‌های مختصات را نسبت به یکدیگر دنبال می‌کنند. در این سناریو، زاویه حمله α منجر به نیروی آیرودینامیکی F روی بدنه در مرکز فشار موشک می‌شود. سپس این نیرو به صورت مجموعه‌ای از نیروها F_x, F_z و یک گشتاور M در مرکز جرم نمایش داده می‌شود.



شکل ۴. دینامیک موشک [۲۲]

معادلات دینامیک موشک در فضای حالت به صورت معادله (۶) در نظر گرفته می‌شود [۲۲].



معادله (۸) را می‌توان به فرم معادله (۹) بیان کرد که d اغتشاش وارد به سیستم است [۲۲].

$$\dot{x} = a(x, t) + B\delta(t) + d \quad (9)$$

$$y = v_n$$

۳- طراحی کنترل کننده

برخورد موشک- هدف زمانی رخ می‌دهد که r فاصله موشک تا هدف به کمتر از مقدار r_{hit} کاهش یابد، هدف در نظر گرفته می‌شود. این شرط در عمل کاملاً منطقی به نظر می‌رسد. با توجه به اینکه r در معادلات حالت در مخرج ظاهر شده است پس شرط صفر شدن آن مقدار مطلوبی برای شبیه‌سازی نیست چراکه در این حالت سیستم ناپایدار خواهد شد. رابطه موجود برای تغییر نرخ r به صورت معادله (۱۰) می‌باشد [۲۲، ۲۱].

$$\dot{r}(t) = v_r(t) \quad (10)$$

با تعریف سرعت $v_c(t) = -v_r(t)$ به عنوان سرعت نزدیک شونده، اگر این سرعت در لحظه اول بزرگ‌تر از صفر باشد و مقدار مثبت خود را حفظ کند این بدان معناست که برخورد رخ خواهد داد. رابطه نرخ تغییرات به صورت معادله (۱۱) قابل بیان است.

$$v_c = a_M \cos(\lambda - \theta) M - a_T \cdot \hat{r} - \frac{v_n^2}{r} \quad (11)$$

با ساده‌سازی معادله (۱۱)، معادله (۱۲) نوشته می‌شود.

$$v_c = -\frac{v_n^2}{r} \quad (12)$$

با توجه به اینکه سرعت نزدیک‌شونده باید مثبت باشد، بنابراین عبارت v_n باید صفر شود تا از منفی شدن v_c جلوگیری شود. بنابراین راه دستیابی به رهگیری در مدل یکپارچه فضای حالت این است که

δ به گونه‌ای محاسبه شود که شرط (۱۳) برقرار باشد [۲۱] [۲۲].

$$\delta: v_n \rightarrow 0 \quad (13)$$

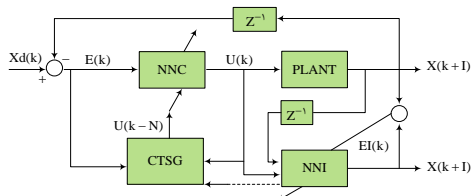
بنابراین هدف کنترلی را در این مسئله می‌توان صفر کردن مؤلفه عمودی بردار سرعت نسبی تعریف نمود.

۳-۱ طراحی کنترل کننده بهینه عصبی

در این قسمت دو نوع کنترل کننده عصبی با ساختار نسبتاً مشابه برای حل مسئله توسعه داده شده‌اند. هر چند هر دوی این کنترل کننده‌ها از دو شبکه عصبی شناساگر و کنترل کننده که با یکدیگر در تعامل هستند تشکیل شده‌اند، تعداد لایه‌های پنهان و روش آموزش شبکه عصبی کنترل کننده در آن‌ها متفاوت است. در بخش شبیه‌سازی نشان داده خواهد شد که همین دو تفاوت، تأثیر عمده‌ای در عملکرد این کنترل کننده‌ها خواهند داشت.

۳-۱-۱ طراحی کنترل کننده عصبی بهینه کلاسیک

این کنترل کننده که بر مبنای توضیحات مرجع [۲۳] با عنوان کنترل کننده عصبی بهینه توسعه یافته است، از دو شبکه عصبی کم‌عمق شناساگر و کنترل کننده تشکیل شده است. شکل زیر این ساختار کنترلی را نمایش می‌دهد که در آن هر دو شبکه شناساگر و کنترل کننده فقط دو لایه پنهان دارند.



شکل ۵. ساختار کنترل کننده عصبی بهینه [۲۳]

بردار خطایی که برای آموزش وزن‌های شبکه عصبی در این روش ایجاد می‌شود توسط بلوک تولیدکننده سیگنال آموزشی کنترل^{۲۷} (CTSG) و با کمینه‌کردن تابع هزینه (۱۴) ایجاد می‌شود.

با فاکتورگیری از عوامل مشترک، تابع تبدیل سیستم به صورت رابطه (۱۷) می‌شود.

$$H(z) = \frac{Y(z)}{X(z)} = \frac{b_0 + b_1 z^{-1} + \dots + b_m z^{-m}}{1 - a_1 z^{-1} - \dots - a_n z^{-n}} \quad (۱۷)$$

$$= \frac{b_0 z^n + b_1 z^{n-1} + \dots + b_m z^{n-m}}{z^n - a_1 z^{n-1} - \dots - a_n}$$

فرم برداری رابطه (۱۵) به صورت رابطه (۱۸) است.

$$y(k) = \theta^T \phi(k) \quad (۱۸)$$

$$\theta = [a_1, \dots, a_n, b_0, \dots, b_m]^T$$

$$\phi(k) = [y(k-1), \dots, y(k-n), x(k), \dots, x(k-m)]^T$$

$\theta, \phi(k)$ به ترتیب بردار پارامترهای سیستم و بردار اندازه‌گیری هستند. در شناسایی پارامتریک سیستم، هدف شناساگر دستیابی به تخمینی از بردار پارامترهای سیستم به صورت $\hat{\theta} = [\hat{a}_1, \dots, \hat{a}_n, \hat{b}_0, \dots, \hat{b}_m]^T$ است به نحوی که با استفاده از رابطه ۱۸ بتوان خروجی سیستم را از طریق رابطه (۱۹) در هر لحظه تخمین زد.

$$\hat{y}(k) = \hat{\theta}^T \phi(k) \quad (۱۹)$$

طبیعتاً هدف شناساگر کمینه کردن خطای شناسایی است.

$$\lim_{k \rightarrow \infty} e(k) = \lim_{k \rightarrow \infty} (y(k) - \hat{y}(k)) = 0 \quad (۲۰)$$

معادله‌های پیش روی شبکه به صورت رابطه (۲۱) قابل بیان است.

$$net_h(k) = \sum_{j=1}^n v_{hj}(k) y(k-j) + \sum_{i=0}^m w_{hi}(k) x(k-i) \quad (۲۱)$$

که در آن وزن بین سیگنال $y(k-j)$ و نورون h م لایه پنهان و w_{hi} وزن بین سیگنال $x(k-i)$ و

$$J = \frac{1}{2} \sum_{p=t^0}^{t^0+N-1} \left[\sum_{i=1}^{num_x} (x_i^d(p+1) - x_i(p+1))^2 + \lambda \sum_{j=1}^{num_u} u_j^2(p) \right] \quad (۱۴)$$

که در آن N تعداد گام‌های زمانی قبلی است که برای آموزش شبکه کنترل‌کننده در نظر گرفته می‌شود.

شناسایی سیستم فرایند استفاده از مدل‌های ریاضی مناسب و الگوریتم‌های یادگیری برای نگاشت داده‌های تجربی از طریق کمینه کردن یک معیار خطا بین خروجی سیستم و خروجی مدل است [۲۴]. مدل‌های میانگین متحرک رگرسیون خودکار^{۲۸} مدل‌های رگرسیون خطی هستند که از معادلات تفاضلی برای مرتبط کردن خروجی مدل در زمان حال به ورودی‌های فعلی، ورودی‌های گذشته و خروجی‌های گذشته استفاده می‌کنند. یک مدل $ARMA$ در حالت کلی در معادله (۱۵) قابل بیان است.

$$y(k) = a_1 y(k-1) + \dots + a_n y(k-n) + b_0 x(k) + \dots + b_m x(k-m) \quad (۱۵)$$

$$= \sum_{j=1}^n a_j y(k-j) + \sum_{i=0}^m b_i x(k-i)$$

با اعمال تبدیل

$$\Xi(z) = Z\{\xi[k]\} = \sum_{n=-\infty}^{\infty} \xi[k] z^{-n}$$

معادلات تفاضلی سیستم فضای گسسته k به فضای پیوسته Z منتقل می‌شود.

$$Y(z) = a_1 z^{-1} Y(z) + \dots + a_n z^{-n} Y(z) + b_0 X(z) + \dots + b_m X(z) \quad (۱۶)$$



$$\begin{aligned} v_{hj}(k+1) & \quad (28) \\ &= v_{hj}(k) - \eta^1 \frac{\partial E(k)}{\partial v_{hj}(k)} \\ &= v_{hj}(k) \\ &+ \eta^1 w_{out_h}(k) e(k) f'(net_h(k)) y_l \end{aligned}$$

در این رابطه η^1 نرخ آموزش وزن‌های لایه پنهان است. در آخر برای آموزش وزن‌های بین ورودی فعلی و تأخیرهای آن و نورون‌های لایه پنهان معادله (۲۹) نوشته می‌شود.

$$\begin{aligned} w_{hi}(k+1) & \quad (29) \\ &= w_{hi}(k) - \eta^1 \frac{\partial E(k)}{\partial w_{hi}(k)} \\ &= w_{hi}(k) \\ &+ \eta^1 w_{out_h}(k) e(k) f'(net_h(k)) x_i \end{aligned}$$

پیشتر اشاره شد که ضروری است تابع فعال‌ساز به‌کاررفته در این مسئله تحلیلی باشد تا بتوان سری تیلور آن را محاسبه کرد. با توجه به اینکه دامنه توابع فعال‌ساز متداول معمولاً یک همسایگی به مرکز نقطه صفر است، در صورتی‌که این تابع در یک همسایگی صفر تحلیلی باشد، با استفاده از سری تیلور تابع فعال‌ساز نورون h ام لایه پنهان مطابق (۳۰) نوشته می‌شود.

$$\begin{aligned} \forall net_h(k) \approx 0: f(net_h(k)) & \quad (30) \\ &= \sum_{n=0}^{\infty} \frac{1}{n!} f^{(n)}(0) (net_h(k))^n \\ &= f(0) + f'(0)(net_h(k)) + \dots \end{aligned}$$

با حذف ترم‌های غیرخطی سری تیلور تابع فعال‌ساز می‌توان خروجی نورون لایه پنهان را در اطراف نقطه صفر به شکل (۳۱) تقریب زد.

$$\begin{aligned} \forall net_h(k) \approx 0: f(net_h(k)) & \quad (31) \\ &\approx f(0) \\ &+ f'(0)(net_h(k)) \end{aligned}$$

حال با جایگزینی مقدار $net_h(k)$ از رابطه (۳۱) رابطه (۳۲) نوشته می‌شود.

نورون h ام لایه پنهان است. خروجی این نورون را می‌توان به‌صورت معادله (۲۲) محاسبه کرد.

$$o_h(k) = f(net_h(k)) \quad (22)$$

در این رابطه f تابع فعال‌ساز لایه پنهان است که در مسئله موردبحث می‌توان از هر تابع تحلیلی در این قسمت استفاده کرد. با توجه به شکل، لایه خروجی تنها از یک نورون تشکیل شده است.

$$net_{out}(k) = \sum_{h=1}^H w_{out_h}(k) o_h(k) \quad (23)$$

در رابطه بالا w_{out_h} وزن بین سیگنال $o_h(k)$ یعنی خروجی نورون h ام لایه پنهان و نورون لایه خروجی است. در ساختار پیشنهادی، از تابع فعال‌ساز خطی در لایه خروجی استفاده شده است. بنابراین خروجی شبکه به صورت معادله (۲۴) قابل بیان است.

$$\hat{y}_{net}(k) = net_{out}(k) \quad (24)$$

خطای شبکه در هر گام زمانی اختلاف بین مقدار خروجی پلنت و خروجی شبکه است.

$$e(k) = y(k) - \hat{y}_{net}(k) \quad (25)$$

برای محاسبه گرادیان خطا می‌توان از معیار مجموع مربعات خطا^{۲۹} استفاده کرد.

$$E(k) = \frac{1}{2} e^2(k) \quad (26)$$

آموزش شبکه به یکی از روش‌های مرسوم مبتنی بر گرادیان^{۳۰} مانند گرادیان نزولی قابل‌دستیابی است. برای آموزش وزن‌های لایه خروجی معادله (۲۷) نوشته می‌شود.

$$\begin{aligned} w_{out_h}(k+1) &= w_{out_h}(k) \\ &- \eta^2 \frac{\partial E(k)}{\partial w_{out_h}(k)} \\ &= w_{out_h}(k) \\ &+ \eta^2 e(k) o_h(k) \end{aligned} \quad (27)$$

در رابطه بالا η^2 نرخ آموزش وزن‌های لایه خروجی است. برای آموزش وزن‌های بین خروجی‌های گام‌های قبلی و نورون‌های لایه پنهان می‌توان از معادله (۲۸) استفاده کرد.



$$\hat{y}_{net}(k) \approx \sum_{h=1}^H \underbrace{f(0) w_{out_h}(k)}_{dc\ offset} + \sum_{j=1}^n \underbrace{\sum_{h=1}^H f'(0) w_{out_h}(k) v_{hj}(k) y(k-j)}_{\hat{a}_j} + \sum_{i=0}^m \underbrace{\sum_{h=1}^H f'(0) w_{out_h}(k) w_{hi}(k) x(k-i)}_{\hat{b}_i} \quad (35)$$

از مقایسه تقریب خطی خروجی شبکه با معادله تفاضلی شناساگر پارامتریک در رابطه (۳۵) می‌توان تخمینی از پارامترهای مدل تفاضلی به دست آورد.

$$\hat{a}_j = f'(0) \sum_{h=1}^H w_{out_h}(k) v_{hj}(k)$$

$$\hat{b}_i = f'(0) \sum_{h=1}^H w_{out_h}(k) w_{hi}(k) \quad (36)$$

$$dc\ offset = f(0) \sum_{h=1}^H w_{out_h}(k)$$

چنانچه تابع تبدیل سیستم شناسایی شده ناپایدار باشد، می‌توان از الگوریتم پیشنهادی مرجع [۲۴] برای انتقال قطب‌های ناپایدار به داخل دایره واحد در فضای Z استفاده کرد. این الگوریتم بدون تغییر فاز قطب‌های ناپایدار، تنها اندازه آن‌ها را تا حدی کاهش می‌دهد که سیستم شناسایی شده پایدار شود.

الگوریتم ۱. پایدارسازی تابع تبدیل شناسایی شده	
۱	محاسبه قطب‌های تابع تبدیل شناسایی شده با به دست آوردن ریشه‌های معادله مشخصه شناساگر
۲	به ازای قطب‌هایی که اندازه آن‌ها بزرگ‌تر از ۱ است: ضرب قطب در ۰.۹ و جایگزینی آن با مقدار قبلی تا زمانی که اندازه قطب کوچک‌تر مساوی ۱ شود.
۳	به دست آوردن معادله مشخصه سیستم پایدار شده با محاسبه معادله یک چندجمله‌ای که ریشه‌های آن قطب‌های اصلاح شده هستند.

$$o_h(k) \approx f(0) + f'(0) \left(\sum_{j=1}^n v_{hj}(k) y(k-j) + \sum_{i=0}^m w_{hi}(k) x(k-i) \right) \quad (32)$$

و با استفاده از رابطه‌های (۳۱) و (۳۲) خروجی شبکه را می‌توان به شکل معادله (۳۳) تقریب زد.

$$\hat{y}_{net}(k) = \sum_{h=1}^H w_{out_h}(k) o_h(k) \approx \sum_{h=1}^H w_{out_h}(k) \left[f(0) + f'(0) \left(\sum_{j=1}^n v_{hj}(k) y(k-j) + \sum_{i=0}^m w_{hi}(k) x(k-i) \right) \right] \quad (33)$$

با جداسازی جملات سری در رابطه (۳۳)، معادله (۳۴) به دست خواهد آمد.

$$\hat{y}_{net}(k) \approx \sum_{h=1}^H w_{out_h}(k) f(0) + \sum_{h=1}^H w_{out_h}(k) f'(0) \sum_{j=1}^n v_{hj}(k) y(k-j) + \sum_{h=1}^H w_{out_h}(k) f'(0) \sum_{i=0}^m w_{hi}(k) x(k-i) \quad (34)$$

و در نهایت با جابه‌جایی تقدم سری‌های دوگانه و بازنویسی معادله (۳۴)، معادله (۳۵) به دست می‌آید.



۲-۳ طراحی کنترل کننده عصبی مبتنی بر یادگیری تقویتی

گرچه واژگان فنی^{۳۱} به کاررفته در زمینه یادگیری تقویتی با واژگان مصطلح کنترل کننده های عصبی تفاوت دارند، اما کنترل کننده های ایجاد شده بر اساس یادگیری تقویتی مشابهت ساختاری زیادی به آنچه که در کنترل کننده عصبی بهینه کلاسیک بیان شد دارند. در این نوع کنترل کننده ها نیز دو شبکه عصبی مجزا با مشارکت یکدیگر به یادگیری رفتار پلنت و کنترل آن می پردازند. در اینجا شبکه عصبی عامل^{۳۲} (کنترل کننده) وظیفه ایجاد عمل^{۳۳} (تولید دستور کنترلی) و تعامل با محیط^{۳۴} (پلنت) را داشته و شبکه عصبی نقاد^{۳۵} (شناساگر) با توجه تابع پاداش^{۳۶} (شاخص عملکرد بهینه سازی) سیاست^{۳۷} رفتار عامل را به گونه ای تغییر می دهد (آموزش شبکه ها) که تابع پاداش در پایان بیشینه شود. با توجه به اینکه شبکه های عصبی به کاررفته در این نوع کنترل کننده ها اغلب از نوع عمیق هستند، عملکرد صحیح آنها به شدت وابسته به در اختیار داشتن حجم داده زیاد است. چرا که اولاً پارامترهای قابل آموزش شبکه (عناصر ماتریس های وزن شبکه) در این کنترل کننده ها بسیار زیاد بوده و ثانیاً توابع فعال ساز آنها از نوع Relu^{۳۸} هستند که رفتاری شبه خطی دارند و ایجاد نداشت غیرخطی ورودی-خروجی توسط آنها صرفاً با داشتن داده های عظیم امکان پذیر است. در آموزش کنترل کننده های مبتنی بر یادگیری تقویتی این حجم عظیم داده از تکرار متعدد سناریوی کنترلی در محیط شبیه سازی ایجاد می شود. برای طراحی کنترل کننده عصبی مبتنی بر یادگیری عمیق معادله سیستم به صورت $\dot{x} = Ax + bu$ در نظر گرفته می شود. ورودی کنترلی u مطابق (۳۷) بیان می شود [۲۵].

$$u = -\frac{1}{2}R^{-1}G^T(\xi)\nabla\phi(\xi)^T\hat{W} \quad (37)$$

در (۳۷)، R ماتریس وزنی مثبت است که مقدار آن توسط بهینه سازی تعیین خواهد شد. \hat{W} وزن های مرتبط با شبکه هستند. ϕ بردار رگرسیون و $\nabla\phi$

گرادیان آن است. $\xi = [e^T, x_d^T]^T$ متغیر حالت تعمیم یافته نام دارد که در آن $e = x - x_d$ خطای تعقیب می باشد. همچنین G از (۳۸) به دست می آید.

$$G = \begin{bmatrix} b \\ 0 \end{bmatrix} \quad (38)$$

بردار رگرسیون به صورت (۳۹) در نظر گرفته می شود.

$$\phi = [e^2, e \cdot x_d, x_d^2]^T \quad (39)$$

در نتیجه گرادیان آن به صورت (۴۰) خواهد بود.

$$\nabla\phi = \begin{bmatrix} 2e & 0 \\ x_d & e \\ 0 & 2x_d \end{bmatrix} \quad (40)$$

همچنین وزن ها با رابطه (۴۱) به روزرسانی می شوند.

$$\begin{aligned} \dot{\hat{W}} &= -\frac{\alpha_1\kappa}{(\kappa^T\kappa + 1)^2} \left\{ \int_{t-\Delta t}^t [Q_1 \right. \\ &- \gamma\hat{W}^T\phi + \frac{1}{4}\hat{W}^T\nabla\phi\nabla\phi^T\hat{W}] d\tau \\ &\left. + \Delta\phi\hat{W} \right\} - \frac{1}{2}\beta(\xi, u)\alpha_2\nabla\phi\nabla\phi^T \end{aligned} \quad (41)$$

در رابطه (۴۱)، $\alpha_1, \alpha_2, \gamma$ و Γ ثابت هایی هستند که از بهینه سازی به دست می آیند. Q_1 از (۴۲) به دست خواهد آمد.

$$Q_1 = \xi^T Q_{11} \xi \quad (42)$$

$$Q_{11} = \begin{bmatrix} Q & 0 \\ 0 & 0 \end{bmatrix}$$

در (۴۲)، Q یک ماتریس وزنی است که توسط بهینه سازی تعیین می شود. همچنین تابع $\beta(\xi, u)$ از معادله ۴۳ به دست خواهد آمد.



جدول ۱. پارامترهای موشک [۲۲]

پارامتر	اندازه	واحد
I_{yy}	1000	$\frac{kg\ m^2}{s}$
m	100	kg

جدول ۲. ضرایب آیرودینامیکی [۲۲]

پارامتر	اندازه	واحد
S	0.7854	m^2
$C_{N\alpha}$	0.11	-
$C_{M\alpha}$	-0.01	-
$C_{M\delta}$	-0.015	-
C_{Mq}	-0.001	-

جدول ۳. پارامترهای دینامیکی فرمان کنترلی [۲۲]

پارامتر	اندازه	واحد
δ_{max}	± 20	deg
$\dot{\delta}_{max}$	± 600	deg/s

جدول ۴. شرایط اولیه موشک [۲۲]

پارامتر	اندازه	واحد
$\alpha(0)$	0	deg
$\lambda(0)$	20	deg
$(0)r$	4	km
$(0)v$	500	$\frac{m}{s}$
$\gamma_M(0)$	10	deg

جدول ۵. پارامترهای هدف [۲۲]

پارامتر	اندازه	واحد
V_T	100	$\frac{m}{s}$
a_T	6	g

۴-۱ شبیه‌سازی عملکرد کنترل‌کننده عصبی بهینه

در این قسمت نحوه اجرا و عملکرد دو نوع کنترل‌کننده عصبی توسعه داده شده برای حل مسئله موجود بررسی شده و نتایج آن تحلیل می‌شود. در بخش قبل بیان شد که این دو کنترل‌کننده از نظر ساختاری شباهت زیادی به یکدیگر دارند و تفاوت اصلی آن‌ها در تعداد لایه‌های پنهان شبکه‌های عصبی

$$\beta(\xi, u) = \begin{cases} 0, & \text{if } \begin{pmatrix} \xi(t)^T \xi(t) - \\ \xi(t - \Delta t)^T \\ \xi(t - \Delta t) \end{pmatrix} \leq 0 \\ 1, & \text{else} \end{cases} \quad (43)$$

$$\kappa = \Delta\phi(\xi(t)) \quad (44)$$

۳-۳ مسئله بهینه‌سازی

برای این مسئله، یک تابع هزینه به صورت معادله (۴۵) در نظر گرفته شده است.

$$J(x) = v_n^2 + (t_{final} - 10)^2 \quad (45)$$

تابع هزینه معادله (۴۵) از دو قسمت تشکیل شده

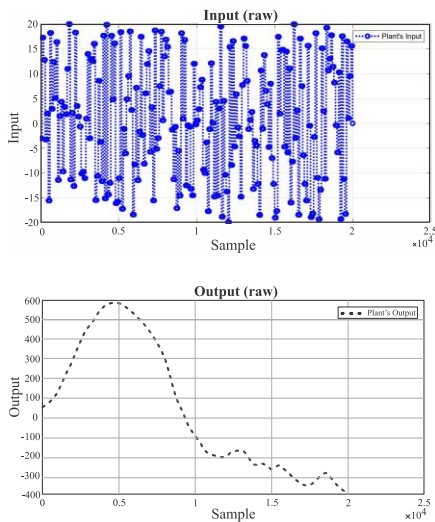
است. قسمت اول، توان دوم مؤلفه عمودی سرعت نسبی است تا دقت برخورد بیشتر شود و موشک به اهداف اصابت کند. این قسمت باعث می‌شود تا موشک سریع‌تر در مسیر اصلی خود قرار گیرد. همان‌طور که در قسمت طراحی کنترلر ذکر شد، هدف اصلی طراحی کنترلر صفر شدن مؤلفه عمودی سرعت نسبی است. قسمت دوم تابع هزینه نیز کمک می‌کند تا در زمان حدود ۱۰ ثانیه برخورد موشک و جسم هدف صورت گیرد. این زمان یک زمان منطقی در برخورد دو جسم است و شبیه‌سازی را واقعی‌تر می‌کند.

۴- شبیه‌سازی‌ها و نتایج

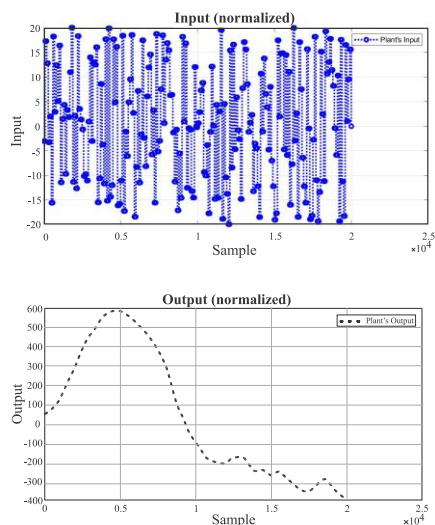
پس از تکمیل طراحی کنترل‌کننده‌های استفاده‌شده، در این بخش عملکرد کنترل‌کننده‌های طراحی‌شده بررسی می‌شود. پارامترهای پلنت و شرایط اولیه در تمام این شبیه‌سازی‌ها یکسان و مطابق جدول ۱ تا ۵ است. در ضمن برای مطمئن شدن از اینکه ورودی کنترلی از حد مجاز تعیین‌شده در صورت مسئله تجاوز نمی‌کند، یک تابع اشباع در ورودی پلنت قرار داده شده است تا در صورتی که کنترل‌کننده مقدار ورودی کنترلی بهینه را خارج از بازه مجاز عملگر تعیین نمود، کران بالا یا پایین تابع اشباع به‌عنوان ورودی کنترلی به پلنت اعمال شود.



پلنت را پیش و پس از مرحله نرمال‌سازی نشان می‌دهد. این داده‌ها در مرحله بعد به شکل مناسب مرتب می‌شوند تا در قالب ورودی‌ها و خروجی‌های شبکه شناساگر استفاده شوند.



شکل ۷. داده‌های ورودی و خروجی پلنت غیرخطی پیش از فرایند نرمال‌سازی



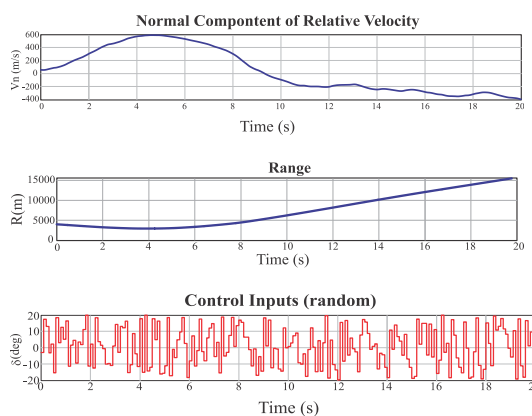
شکل ۸. داده‌های ورودی و خروجی پلنت غیرخطی پس از فرایند نرمال‌سازی

با توجه به اینکه معادلات حلقه باز سیستم ۷ متغیر حالت دارد، مرتبه سیستم خطی‌سازی شده ۷ در نظر گرفته می‌شود. بنابراین تعداد ۷ تأخیر خروجی پلنت به همراه ۷ تأخیر ورودی آن به همراه مقدار ورودی در لحظه فعلی به عنوان ورودی‌های شبکه عصبی شناساگر استفاده شده و خروجی شبکه

به کار رفته و نحوه آموزش شبکه‌هاست. با توجه به اینکه تعداد لایه‌های پنهان شبکه‌های عصبی استفاده شده در کنترل‌کننده مبتنی بر یادگیری تقویتی و دفعات آموزش این شبکه‌ها به مراتب بیشتر از شبکه‌های عصبی کنترل‌کننده بهینه کلاسیک است، انتظار می‌رود کنترل‌کننده مبتنی بر یادگیری عمیق عملکرد بهتری از خود به نمایش بگذارد.

۴-۲ شبیه‌سازی عملکرد کنترل‌کننده عصبی بهینه کلاسیک

همان‌طور که در بخش طراحی بیان شد، کنترل‌کننده شبکه عصبی توسعه داده شده در این مسئله برای شروع کنترل پلنت نیاز به شناسایی مدل و ایجاد مدلی خطی از دینامیک آن دارد. بدین منظور الگوریتم کنترلی ابتدا با اعمال یک رشته از ورودی‌های تصادفی و ثبت پاسخ سیستم به این ورودی‌ها، داده لازم برای آموزش شبکه عصبی شناساگر را فراهم می‌آورد. این ورودی تصادفی و پاسخ سیستم به آن در شکل ۶ نشان داده شده است.

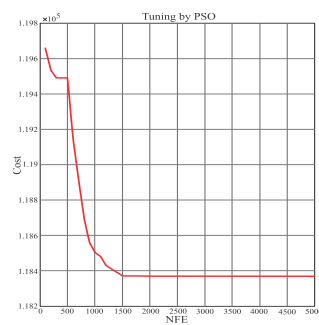
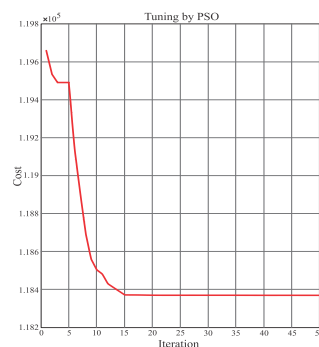


شکل ۶. داده استفاده شده برای شناسایی پلنت توسط شبکه عصبی شناساگر

هر چند نرمال‌سازی داده‌ها به دلیل کاهش احتمال اشباع توابع فعال‌ساز لایه اول به آموزش بهتر شبکه منتهی می‌شود، به دلیل غیرخطی بودن پلنت مورد بررسی و مطابق با توضیحات ارائه شده در بخش طراحی از نرمال کردن داده‌ها در این قسمت اجتناب می‌شود. شکل‌های ۷ و ۸ داده‌های ورودی و خروجی



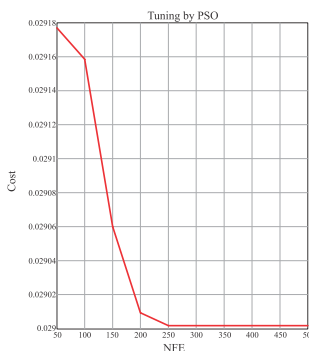
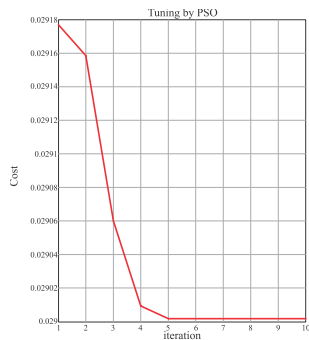
نیز تخمین خروجی پلنت در لحظه فعلی تعیین می‌شود. از ۱۹۹۹۴ داده ورودی و خروجی پلنت، ۷۰ درصد به‌عنوان داده آموزش و ۳۰ درصد به‌عنوان داده آزمایش استفاده شده‌اند. تعداد نورون‌های لایه پنهان در این آزمایش ۵ عدد بوده و از تابع فعال‌ساز دوقطبی در این لایه استفاده شده است. علاوه بر این، وزن‌های اولیه توسط الگوریتم بهینه‌سازی اجتماع ذرات با جمعیت ۱۰۰ ذره و تعداد ۵۰ تکرار انتخاب شده است. این بدان معنی است که معادلات پیش رو به تعداد ۵۰۰۰ مرتبه برای کل داده‌های آموزش (۱۳۹۹۶ داده) محاسبه شده است. به عبارتی معادلات پیش‌رو به تعداد ۶۹۹۸۰۰۰۰ بار تکرار شده‌اند تا حدس مناسبی برای وزن‌های اولیه ایجاد شود. روند تغییرات شاخص خطا در فرایند بهینه‌سازی وزن‌های اولیه شبکه عصبی در شکل ۹ نشان داده شده است.



شکل ۹. تغییرات تابع هزینه برحسب تکرار بهینه‌سازی به کمک الگوریتم اجتماع ذرات وزن‌های اولیه شبکه عصبی شناساگر

در این آزمایش مقادیر مناسب نرخ آموزش لایه پنهان و لایه خروجی نیز توسط الگوریتم بهینه‌سازی اجتماع ذرات تعیین شده‌اند. جمعیت ذرات در این قسمت ۵۰ و تعداد تکرارها ۱۰ انتخاب شده‌اند و این بدان معنی است که شبکه به تعداد ۵۰۰ بار با مقادیر

متفاوت نرخ‌های آموزش لایه اول و دوم آموزش دیده (در این قسمت روند آموزش تا ۷۰ درصد ایپاک بیشینه ادامه می‌یابد) تا مقادیر بهینه انتخاب شوند. شکل ۱۰ روند تغییرات شاخص خطا برای بهینه‌سازی نرخ‌های آموزش را نشان می‌دهد.

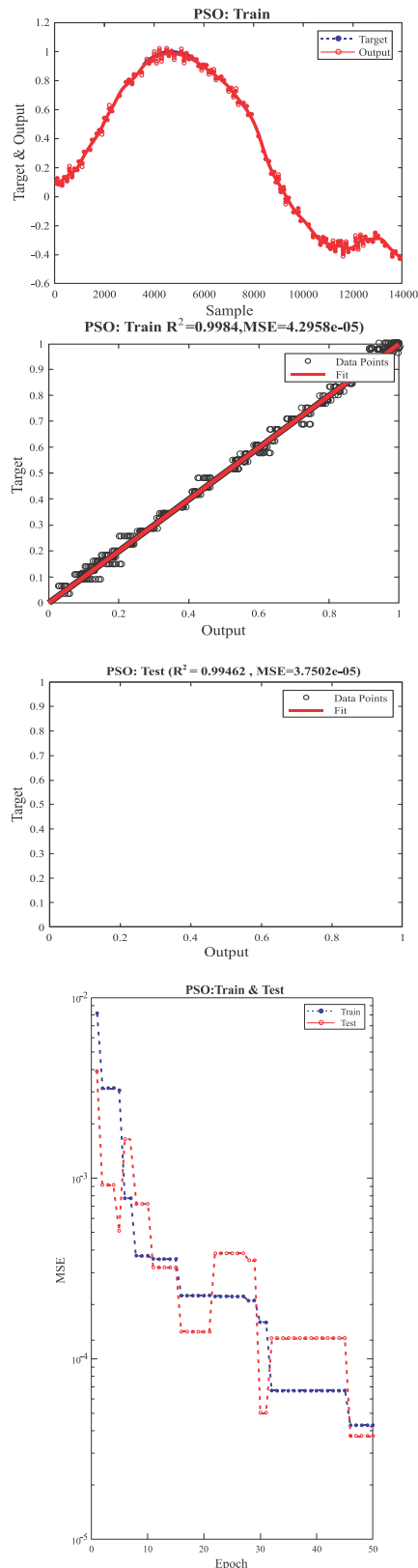


شکل ۱۰. تغییرات تابع هزینه برحسب تکرار بهینه‌سازی به کمک الگوریتم اجتماع ذرات ضرایب یادگیری شبکه عصبی شناساگر

مشاهده می‌شود که تغییرات شاخص خطا در فرایند بهینه‌سازی نرخ‌های آموزش بسیار کوچک می‌باشد و دلیل این امر را می‌توان عملکرد مناسب الگوریتم بهینه‌سازی در تعیین وزن‌های اولیه دانست. چرا که نقطه کمینه کلی برای شاخص خطا توسط این الگوریتم قبلاً پیدا شده و بهینه‌سازی نرخ آموزش تأثیر چندانی در رسیدن به این نقطه توسط الگوریتم آموزش مبتنی بر گرادیان ندارد. با محاسبه‌های مقدار مناسب برای وزن‌های اولیه لایه‌های پنهان و خروجی و نیز تعیین مقدار بهینه نرخ‌های آموزش، نوبت به اجرای الگوریتم آموزش مبتنی بر گرادیان می‌رسد. در این آزمایش از الگوریتم لونیبرگ-مارکوارت برای آموزش شبکه استفاده شده است (این الگوریتم پیش‌تر در بهینه‌سازی نرخ‌های آموزش استفاده شده

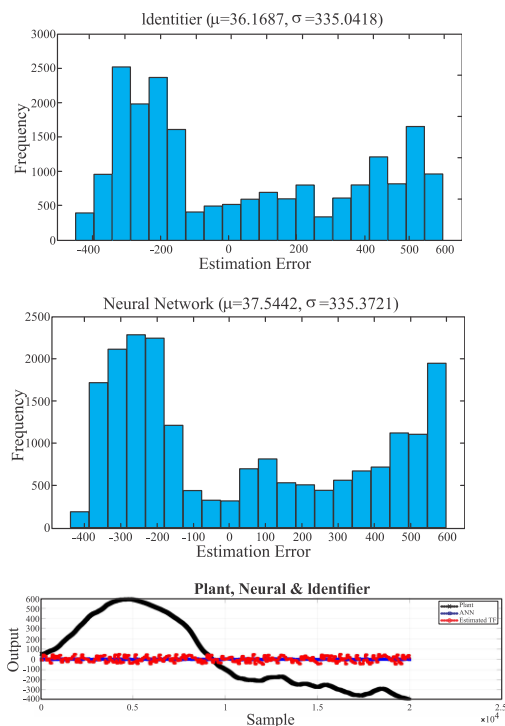


است). شکل ۱۱ نتایج مربوط به آموزش موفق شبکه شناساگر را به تصویر می کشد.



شکل ۱۱. نتیجه آموزش شبکه عصبی شناساگر با پارامترهای بهینه‌سازی شده

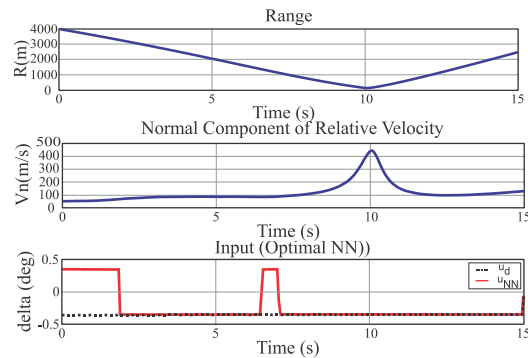
همان‌طور که در شکل ۱۱ مشخص است، عملکرد شبکه در آموزش وزن‌ها و رسیدن به نقطه بهینه بسیار خوب بوده و خروجی شبکه و پلنت برای داده‌های آموزش و آزمایش به‌خوبی متناسب شده‌اند. در واقع دلیل توقف الگوریتم آموزش پیش از رسیدن به بیشینه ایپاک هم، رسیدن به حدنصاب تعیین شده برای تغییرات معیار خطاست. پس از آموزش شبکه و همگرا شدن وزن‌های دو لایه پنهان و خروجی به مقدار مشخص (این مقادیر در فایل اکسل خروجی برنامه به شکل خودکار درج می‌شوند)، الگوریتم تبدیل وزن‌های شبکه به پارامترهای مدل $ARMA$ فعال شده و تابع تبدیل تخمین‌گر پلنت محاسبه می‌شود. به منظور صحت‌گذاری عملکرد این تخمین‌گر و نیز تخمین‌گر شبکه عصبی ایجاد شده، در این قسمت برنامه داده‌های صحت‌گذاری از فایل اکسل دیتاست خوانده شده و در اختیار تخمین‌گرها قرار می‌گیرد. داده‌ها به شکل نرمال نشده در بخش صحت‌گذاری استفاده می‌شوند و تخمین‌گرها فقط به ورودی پلنت در هر لحظه دسترسی دارند. نتایج عملکرد این تخمین‌گرها در مواجهه با ورودی‌های صحت‌گذاری در شکل ۱۲ نشان داده شده است.



شکل ۱۲. نتیجه خطی‌سازی پلنت با استفاده از شناساگر عصبی و الگوریتم مرجع [۲۴]



برخلاف انتظار مشاهده می‌شود به‌رغم آموزش موفق شبکه عصبی شناساگر، این شبکه و تابع تبدیل ایجاد شده به کمک وزن‌های آن تخمین درستی از رفتار پلنت ندارند. دلیل این امر را می‌توان در دینامیک بالای داده‌ها دانست. بدین معنی که داده‌های آموزش به‌طور کامل معرف دینامیک سیستم نبوده‌اند و نگاهت ایجاد شده به این روش به‌درستی قادر به تخمین رفتار سیستم نیست. با توجه به اینکه آموزش شبکه عصبی کنترل‌کننده به‌طور کامل وابسته به صحت عملکرد شبکه شناساگر در کنترل‌کننده عصبی بهینه است، انتظار این است که این کنترل‌کننده به‌درستی قادر به تأمین اهداف کنترلی نباشد. شکل ۱۳ عملکرد کنترل‌کننده بهینه را در شبیه‌سازی مسئله نشان می‌دهد.



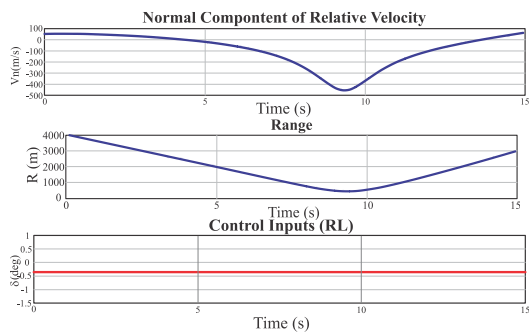
شکل ۱۳. عملکرد کنترل‌کننده عصبی بهینه کلاسیک

مشاهده می‌شود که مطابق استدلال بیان‌شده، کنترل‌کننده عصبی بهینه کلاسیک به دلیل ضعف شبکه شناساگر در شناسایی دینامیک سیستم عملکرد قابل قبولی ندارد.

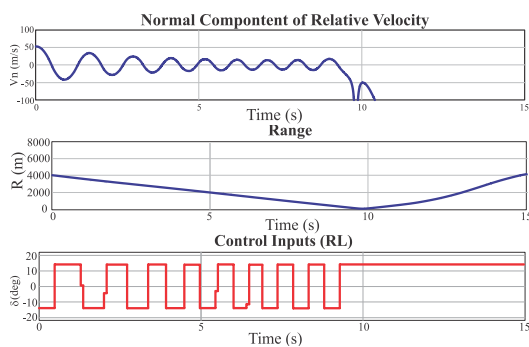
۳-۴ شبیه‌سازی کنترل‌کننده عصبی بهینه مبتنی بر یادگیری عمیق

همان‌طور که پیشتر اشاره شد، ساختار کلی کنترل‌کننده عصبی بهینه مبتنی بر یادگیری عمیق مشابه کنترل‌کننده عصبی بهینه کلاسیک می‌باشد با این تفاوت که در این ساختار از شبکه عصبی عمیق به‌جای شبکه کم‌عمق استفاده شده است و یادگیری هر دو شبکه نقاد (شناساگر) و عامل (کنترل‌کننده) به شکل هم‌زمان و بر اساس تابع پاداش (تابع هزینه)

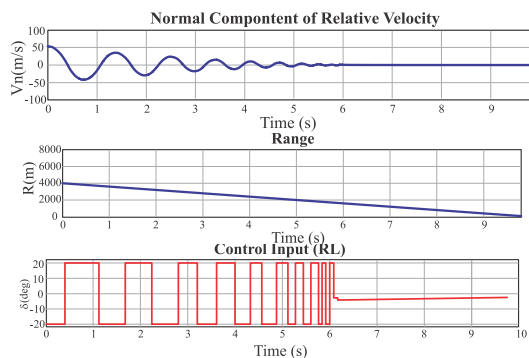
انجام می‌شود. برای هر بار آموزش شبکه در این رویکرد کنترلی، لازم است کل شبیه‌سازی یک بار از ابتدا تا انتها اجرا شود. بنابراین هزینه محاسباتی فرایند آموزش شبکه‌های عصبی در این رویکرد کنترلی فوق‌العاده زیاد است. نمودار عملکرد کنترل‌کننده مذکور پس از انجام دفعات متفاوت آموزش در شکل‌های ۱۴ تا ۱۷ نشان داده شده است.



شکل ۱۴. عملکرد کنترل‌کننده عصبی بهینه مبتنی بر یادگیری عمیق با ۱۰۰ تکرار سناریوی آموزشی

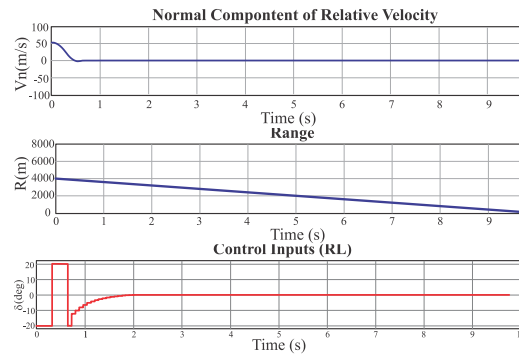


شکل ۱۵. عملکرد کنترل‌کننده عصبی بهینه مبتنی بر یادگیری عمیق با ۱۰۰۰ تکرار سناریوی آموزشی



شکل ۱۶ - عملکرد کنترل‌کننده عصبی بهینه مبتنی بر یادگیری عمیق با ۵۰۰۰ تکرار سناریوی آموزشی





شکل ۱۷. عملکرد کنترل کننده عصبی بهینه مبتنی بر یادگیری عمیق با ۱۰۰۰۰ تکرار سناریوی آموزشی

مشاهده می‌شود که در این مسئله، عامل پس از ۵۰۰۰ تکرار شبیه‌سازی شروع به کنترل قابل قبول سیستم نموده و پس از ۱۰۰۰۰ تکرار رفتار مطلوبی از خود به نمایش گذاشته است. در این نوع کنترل کننده مناسب‌تر آن است که عملکرد کنترل کننده با انجام آزمایش‌های متعدد و به شکل آمار دفعات موفق نسبت به دفعات آزمایش سنجیده شود چرا که موفقیت یک عامل یادگیری تقویتی در یک آزمایش لزوماً به معنی موفقیت آن در آزمایش بعدی نیست.

۵- نتیجه‌گیری

در این مقاله مسئله طراحی کنترلر عصبی بهینه برای مدل یکپارچه موشک هدف مدل دوبعدی بررسی شده است. ابتدا معادلات یکپارچه موشک به‌طور کامل شرح داده شد و سپس طراحی کنترلر عصبی بهینه بررسی شد. شبکه عصبی ابتدا به‌صورت کلاسیک طراحی شد. با توجه به تغییر زیاد دینامیک سیستم در مسئله تحت بررسی، مناسب‌تر بود که فرایند شناسایی سیستم پس از گذشت چند گام زمانی تکرار شود، اما فرایند شناسایی سیستم توسط شبکه‌های عصبی معمولاً بسیار زمان‌گیر است و صرفه محاسباتی ندارد. با ایجاد سازوکار مناسب برای ایجاد سیگنال خطا برای آموزش شبکه کنترل کننده، کنترل کننده عصبی بهینه در هر گام زمانی به اعمال ورودی به پلنت پرداخته و سپس بر اساس سیگنال‌های خطای ایجاد شده، شبکه‌های عصبی شناسگر و کنترل کننده آموزش می‌یابند، بهینه بودن کنترل کننده عصبی کاملاً وابسته به بهینه بودن سیگنال مرجع ایجاد شده

توسط تنظیم کننده خطی مرتبه دو است. به دلیل عدم موفقیت شبکه عصبی شناسگر در شناسایی سیستم و خطی‌سازی مدل، سیگنال مرجع آموزش شبکه عصبی کنترل کننده بهینه نبوده و در نتیجه کنترل کننده عصبی بهینه ایجاد شده به این روش در تحقق هدف کنترلی تعیین شده ناموفق است. در ادامه طراحی کنترل کننده عصبی مبتنی بر یادگیری تقویتی انجام شد. در شبیه‌سازی‌ها مشاهده شد که در این مسئله، پس از ۵۰۰۰ تکرار شبیه‌سازی شروع به کنترل قابل قبول سیستم نموده و پس از ۱۰۰۰۰ تکرار رفتار مطلوبی از خود به نمایش گذاشته است.

در پژوهش‌های آتی، دو مورد را در نظر خواهیم گرفت که در حال انجام است که در مقاله‌های آینده ارائه خواهد شد. مورد اول، استفاده از کنترل کننده مدل پیش‌بین است که در این کنترل کننده به لحاظ بهینه بودن کنترل کننده ضمن رعایت قیود کنترلی به شکل ساختاری، امکان تنظیم رفتار کنترل کننده برحسب نیازهای مسئله و نیز سرعت قابل قبول دارای مزیت است. مورد دوم استفاده از کنترل کننده مود لغزشی بهینه نیز به دلیل تحمل بهتر اغتشاشات و نامعینی‌های سیستم و نیز سرعت بیشتر نسبت به سایر رویکردهای کنترلی دارای مزیت است.

۶. مآخذ

- [1] P. Zarchan, Tactical and strategic missile guidance. American Institute of Aeronautics and Astronautics, 2012
- [2] Neil F Palumbo, Ross A Blauwkamp, and Justin M Lloyd. Basic principles of homing guidance. Johns Hopkins APL Technical Digest (Applied Physics Laboratory), 2010.
- [3] Ching-Fang Lin, John Bibel, Ernest J Ohlmeyer, and Steve Malyevac. Optimal design of integrated missile guidance and control. In AI A A and SAE, 1998 World Con/erence, page 5519, 1998.
- [4] James R Cloutier, Christopher N D'Souza, and Curtis P Mracek. Nonlinear regulation and nonlinear H infinity control via the state-dependent. Riccati equation technique: Part 1, theory. In Proceedings of the First International Conference on Nonlinear Problems in Aviation and Aerospace, pages 117-130. Embry-Riddle



and Dynamics, vol. 40, no. 10, pp. 2653-2661, 2017.

- [16] J. Ma, H. Guo and P. Li, Adaptive integrated guidance and control design for a missile with input constraints, IFAC Proceedings Volumes, vol. 46, no. 20, pp. 206-211, 2013.
- [17] Michael A Cross. Missile interceptor integrated guidance and control: single loop higher order sliding mode approach-, PhD thesis, The University of Alabama in Huntsville, 2020
- [18] K. W. Lee and S. N. Singh, Longitudinal nonlinear adaptive autopilot design for missiles with control constraint, Proc. of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering, vol. 232, no. 9, pp. 1655-1670, 2018.
- [19] M. Ma, K. Zhao and S. Song, Adaptive sliding mode guidance law with prescribed performance for intercepting maneuvering target, International Journal of Innovative Computing, Information and Control, vol. 16, no. 2, pp. 631-648, 2020
- [20] M. Hou and G. Duan, Integrated guidance and control of homing missiles against ground _xed targets, Chinese Journal of Aeronautics, vol. 21, no. 2, pp. 162-168, 2008.
- [21] Michael A Cross and Yuri B Shtessel. Single-Loop Integrated Guidance and Control Using High-Order Sliding Mode Control. In Variable Structure Systems and Sliding-Mode Control, chapter 14. Springer, 2020.
- [22] Michael A Cross and Yuri B Shtessel. Integrated Guidance Navigation and Control Using High-Order Sliding Mode Control for a Missile Interceptor. In 2018 AIAA Guidance, Navigation, and Control Conference, page 1121, 2018
- [23] Y. Li and F. Yang, "Optimal neural network control," in *Control Problems*, Toronto, Springer, 2009, pp. 37-42.
- [24] T. A. Tutunji, "Parametric system identification using neural networks," *Applied Soft Computing*, vol. 47, p. 251-261, 2016.
- [25] T. A. Tutunji, "neural network -based Optimal Tracking Control of continous-Time uncertain Nonlinear System via Reinforcement Learning," *neural process left* 51, 2513-2530, 2020.
- [5] James R Cloutier. Adaptive matched augmented proportional navigation, 1994.
- [6] Curtis P Mracek and James R Cloutier. Missile longitudinal autopilot design using the state-dependent Riccati equation method. In Proceedings of the IntematzomaZ Con/eremce cm TVcmZmear Problems m Araatmn and Aerospace, pages 387-396, 1996.
- [7] P K Menon and Ernest J Ohlmeyer. Integrated design of agile missile guidance and autopilot systems. *Control Engineering Practice*, 9(10): 1095-1106, 1999.
- [8] Neil F Palumbo and Todd D Jackson. Integrated missile guidance and control: A state dependent Riccati differential equation approach. In *Control Applications*, 1999. Proceedings of the 1999 IEEE International Conference on, volume 1, pages 243-248. IEEE, 1999.
- [9] P K Menon, Gregory Sweriduk, Ernest J Ohlmeyer, and Steve Malyevac. Integrated guidance and control of moving-mass actuated kinetic warheads. *Journal of Guidance, control, and Dynamics*, 27(1): 118 126, 2004.
- [10] P K Menon, Sessa Vaddi, and Ernest Ohlmeyer. Finite-horizon robust integrated guidance-control of a moving-mass actuated kinetic warhead. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, page 6787, 2006.
- [11] Tae-Won Hwang and Min-Jea Tahk. Integrated backstopping design of missile guidance and control with robust disturbance observer. In *SICE-ICASE*, 2006. International Joint Conference, pages 4911-4915. IEEE, 2006.
- [12] Nathan Harl, S N Balakrishnan, and Craig Phillips. Sliding mode integrated missile guidance and control. In *AIAA Guidance, Navigation, and Control Conference*, number August, page 7741, 2010.
- [13] Nathan Harl and S N Balakrishnan. Reentry terminal guidance through sliding mode control. *Journal of guidance, control, and dynamics*, 33(1): 186-199, 2010.
- [14] X. H. Wang, C. P. Tan and L. P. Cheng, Impact time and angle constrained integrated guidance and control with application to salvo attack, *Asian Journal of Control*, 2019.
- [15] S. He, T. Song and D. Lin, Impact angle constrained integrated guidance and control for maneuvering target interception, *Journal of Guidance, Control,*

۷- فهرست علائم

علائم انگلیسی

r بردار موقعیت (m)

a بردار شتاب (m/s^2)



- 17) Ohlmeyer
- 18) backstepping
- 19) Hwang
- 20) Tahk
- 21) Harl
- 22) Balakrishnan
- 23) Terminal second-order sliding mode
- 24) Wang
- 25) Ma
- 26) backstepping
- 27) Control Training Signal Generator
- 28) Auto-Regressive Moving-Average (ARMA)
- 29) Sum Square Error (SSE)
- 30) Gradient-base
- 31) Terminology
- 32) Agent
- 33) Action
- 34) Environment
- 35) Critic
- 36) Refward Funcion
- 37) Policy
- 38) Rectifier Linear Unit

A	منطقه مرجع آیرودینامیک (m^2)
Ω_ω	بردار نرخ خط دید (m/s)
XY	مختصات ایترسی
$\hat{r}\hat{n}\hat{\omega}$	مختصات خط دید
xyz	مختصات بدنه
x	بردار حالت
δ	بردار ورودی کنترلی (deg)
d	بردار اغتشاش
F	نیروی آیرودینامیکی (N)
M	گشتاور وارد بر بدنه ($N.m$)
m	جرم (kg)
\bar{q}	فشار دینامیکی (pa)
l	طول مرجع آیرودینامیک (m)
α	زاویه حمله (deg)
q	نرخ زاویه فراز (deg)
θ	زاویه فراز (deg)
λ	زاویه خط دید (deg)
U	محور اولیه متصل به بدنه
N	تعداد گام‌های زمانی قبلی
γ_M	زاویه مسیر پرواز (deg)

زیرنویس

M	موشک
T	هدف
r	خط دید
n	عمود بر خط دید
f	تابع فعال‌ساز لایه پنهان

 ۴۲

 سال ۱۴ - شماره ۱

 بهار و تابستان ۱۴۰۲

 نشریه علمی
 دانش و فناوری هوا فضا



۸. پی‌نوشت

- 1) GNC
- 2) impact angle
- 3) Collision triangle
- 4) line-of-sight
- 5) Integrated Guidance and Control
- 6) Miss distance
- 7) Lin
- 8) Yueh
- 9) game-theoretic
- 10) Cloutier
- 11) State-Dependent Riccati Equation
- 12) Menon
- 13) Palumbo
- 14) Monte-Carlo
- 15) Sweriduk
- 16) Vaddi

هدایت و کنترل یکپارچه کانال فرار موشک آشیانه‌یاب زمین به هوا با استفاده از کنترل عصبی بهینه