دانشگاه آزاد اسلامی ـ واحد علوم تحقیقات
دانشکده‌ی مهندسی برق کنترل

گزارش پروژه ی نهایی
درس کنترل عصبی

عنوان:

# رویتگر پایدار شبکه عصبی با کاربرد در بازوهای روباتیک انعطاف پذیر

نگارش:

**کیان خانقاهی**

استاد راهنما:

**دکتر علی معرفیان پور**

مرداد ۱۴۰۰

بسم الله الرحمن الرحیم

## چکیده

در این مقاله یک رویتگر از نوع شبکه ی عصبی پایدار برای سیستم های غیرخطی چندمتغیره پیشنهاد شده است. برعکس بیشتر رویتگر های شبکه عصبی قبلی، این رویتگر از یک شبکه عصبی غیرخطی در پارامتر ها (NLPNN) استفاده می کند. پس، می تواند به سیستم هایی با درجه غیرخطی بالاتر اعمال شود بدون هیچ دانش قبلی درباره ی دینامیک سیستم. قانون یادگیری برای شبکه عصبی یک روش نو براساس اصلاح الگوریتم روش پس انتشار خطا(backprobagation) می باشد. یک عبارت اصلاحیه هم اضافه شده تا مقاوم بودن رویتگر تضمین گردد. در ضمن هیچ فرض بزرگی مثل SPR روی این روش پیشنهادی وجود ندارد. پایداری رویتگر شبکه عصبی بازگشتی توسط روش مستقیم لیاپانوف نشان داده می شود. در آخر هم نتایج شبیه سازی برای یک بازوی روباتیک انعطاف پذیر ارائه داده می شود تا کارایی افزوده ی این روش نشان داده شود.



**کلیدواژه‌ها**: بازوهای انعطاف پذیر، شبکه های عصبی، رویتگر غیرخطی

# فهرست مطالب

**۱ مقدمه** **۹**

**۲ تعریف مسئله** **۱۱**

۲ـ۱ تعاریف ریاضی . . . . . . . . . . . . . . . . . . . . . . . . . . . . . ۱۱

۲ـ۲ رویتگر عصبی ـ تطبیقی پیشنهاد شده . . . . . . . . . . . . . . . . . . . . . . ۱۲

**۳ قانون آپدیت وزن ها و تحلیل پایداری** **۱۶**

۳ـ۱ قانون آپدیت وزن ها . . . . . . . . . . . . . . . . . . . . . . . . . . . . ۱۶

۳ـ۲ تحلیل پایداری . . . . . . . . . . . . . . . . . . . . . . . . . . . . . ۲۰

**۴ مدل بازوی ربات انعطاف پذیر** **۲۵**

۴ـ۱ مدل خطی بازوی ربات انعطاف پذیر تک پیونده . . . . . . . . . . . . . . . . . . ۲۶

۴ـ۲ مدل غیر خطی بازوی ربات انعطاف پذیر تک پیونده . . . . . . . . . . . . . . . . . ۲۷

**۵ شبیه سازی** **۲۸**

۵ـ۱ آماده کردن پارامتر های شبیه سازی در فایل MATLAB برای استفاده در SIMULINK ۲۸

۵ـ۲ نحوه ی شبیه سازی طراحی سیستم مورد نظر در SIMULINK . . . . . . . . . . ۳۰

۵ـ۳ نتایج شبیه سازی . . . . . . . . . . . . . . . . . . . . . . . . . . . . ۳۶

**۶ نتیجه‌گیری** ۶۰

# فهرست شکل‌ها


۲ــ۱ شمای ساختار کلی رویتگر پیشنهاد شده همراه با سیستم . . . . . . . . . . . . . ۱۳

۴ــ۱ شماتیک یک بازوی ربات انعطاف پذیر تک پیونده مدل شده با فنر گردشی . . . . . ۲۵

۵ــ۱ شماتیک همه ی سیستم کنترل حلقه بسته در SIMULINK . . . . . . . . . . . ۳۱
۵ــ۲ ردیابی حالت ها برای سیستم اصلی خطی و سیگنال مرجع ۱ . . . . . . . . . . . ۳۶
۵ــ۳ ردیابی حالت های رویتگر بر سیستم اصلی برای سیستم اصلی خطی و سیگنال مرجع ۱ ۳۷
۵ــ۴ خطای ردیابی حالت های رویتگر بر سیستم اصلی برای سیستم اصلی خطی و سیگنال مرجع ۱ . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . ۳۸
۵ــ۵ ردیابی وزن های رویتگر برای سیستم اصلی خطی و سیگنال مرجع ۱ . . . . . . . ۳۹
۵ــ۶ ردیابی حالت ها برای سیستم اصلی خطی و سیگنال مرجع ۲ . . . . . . . . . . . ۴۰
۵ــ۷ ردیابی حالت های رویتگر بر سیستم اصلی برای سیستم اصلی خطی و سیگنال مرجع ۲ ۴۱
۵ــ۸ خطای ردیابی حالت های رویتگر بر سیستم اصلی برای سیستم اصلی خطی و سیگنال مرجع ۲ . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . ۴۲
۵ــ۹ ردیابی وزن های رویتگر برای سیستم اصلی خطی و سیگنال مرجع ۲ . . . . . . . ۴۳
۵ــ۱۰ ردیابی حالت ها برای سیستم اصلی خطی و سیگنال مرجع۳ . . . . . . . . . . . ۴۴
۵ــ۱۱ ردیابی حالت های رویتگر بر سیستم اصلی برای سیستم اصلی خطی و سیگنال مرجع ۳ ۴۵

۵ـ۱۲خطای ردیابی حالت های رویتگر بر سیستم اصلی برای سیستم اصلی خطی و سیگنال مرجع ۳ . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . ۴۶
۵ـ۱۳ردیابی وزن های رویتگر برای سیستم اصلی خطی و سیگنال مرجع ۳ . . . . . . . . ۴۷
۵ـ۱۴ردیابی حالت ها برای سیستم اصلی غیرخطی و سیگنال مرجع ۱ . . . . . . . . . . ۴۸
۵ـ۱۵ردیابی حالت های رویتگر بر سیستم اصلی برای سیستم اصلی غیرخطی و سیگنال مرجع ۱ . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . ۴۹
۵ـ۱۶خطای ردیابی حالت های رویتگر بر سیستم اصلی برای سیستم اصلی غیرخطی و سیگنال مرجع ۱ . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . ۵۰
۵ـ۱۷ردیابی وزن های رویتگر برای سیستم اصلی غیرخطی و سیگنال مرجع ۱ . . . . . . . ۵۱
۵ـ۱۸ردیابی حالت ها برای سیستم اصلی غیرخطی و سیگنال مرجع ۲ . . . . . . . . . . ۵۲
۵ـ۱۹ردیابی حالت های رویتگر بر سیستم اصلی برای سیستم اصلی غیرخطی و سیگنال مرجع ۲ . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . ۵۳
۵ـ۲۰خطای ردیابی حالت های رویتگر بر سیستم اصلی برای سیستم اصلی غیرخطی و سیگنال مرجع ۲ . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . ۵۴
۵ـ۲۱ردیابی وزن های رویتگر برای سیستم اصلی غیرخطی و سیگنال مرجع ۲ . . . . . . . ۵۵
۵ـ۲۲ردیابی حالت ها برای سیستم اصلی غیرخطی و سیگنال مرجع۳ . . . . . . . . . . . ۵۶
۵ـ۲۳ردیابی حالت های رویتگر بر سیستم اصلی برای سیستم اصلی غیرخطی و سیگنال مرجع ۳ . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . ۵۷
۵ـ۲۴خطای ردیابی حالت های رویتگر بر سیستم اصلی برای سیستم اصلی غیرخطی و سیگنال مرجع ۳ . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . ۵۸
۵ـ۲۵ردیابی وزن های رویتگر برای سیستم اصلی غیرخطی و سیگنال مرجع ۳ . . . . . . . ۵۹

# فهرست جدول‌ها

# فصل ۱

## مقدمه

در این فصل مقدمه ای بر این موضوع گفته می شود.

در قرن اخیر، مدل های رویتگر غیرخطی زیادی برای کنترل سیستم های غیرخطی پیشنهاد شده اند. از این گونه رویتگر ها به رویتگر های کلاسیک مد لغزشی و رویتگر های گین بالا اشاره کرد. ولی این گونه رویتگر ها فقط موقعی قابل طراحی هستند که مدل سیستم اصلی مشخص و یا فرم کلی آن مشخص باشد. مثلاً در بعضی سیستم ها یا مدل سازی آن بسیار سخت است و یا اصلاً به مدل دقیق دسترسی نداریم که این نامعینی در مدل پیشنهادی ایجاد می کند. بازوی ربات انعطاف پذیر از این گونه سیستم ها است که مدل پیچیده ای دارد و مدل دقیق آن هم در دسترس نیست.

در این میان، شبکه های عصبی به عنوان رویتگر بسیار خوب عمل می کنند چرا که نیازی به دانستن مدل سیستم اصلی ندارند و کافی است خروجی و ورودی سیستم را دریافت کنند و براساس آن آموزش ببینند. این به این معنا است که شبکه های عصبی در سیستم های دارای نامعینی هم بسیار خوب عمل می کنند و سیستم را بهتر تقریب می زنند. برای تقریب زدن یک مدل سیستم غیر خطی هم می توان از شبکه های عصبی خطی نسبت به پارامتر مثل SLP یا MLP استفاده کرد و هم از شبکه های غیرخطی نسبت به پارامتر مثل RBF استفاده کرد. گزینه ی اولی زیاد مورد پسند نمی باشد چرا که هر چه درجه ی غیرخطی سیستم بیشتر شود باید تعداد نورون های لایه ی مخفی و تعداد لایه ها را هم بیشتر کرد که میزان پردازش را بیشتر می کند. ولی شبکه هایی مانند شبکه RBF به علت داشتن توابع غیرخطی قابل تنظیم در لایه ی مخفی، می توانند هر مدل غیرخطی با هر درجه غیرخطی را تقریب بزنند.

برای قانون آموزش رویتگر شبکه عصبی این نوع سیستم ها معمولاً از روش پس انتشار خطا استفاده

می شود. در این زمینه کارهای زیادی انجام شده ولی کمتر دیده می شود که تحلیل اثبات پایداری آموزش هم انجام شود. برای همین تعداد کمی کار وجود دارد که در آن، از قانون های آپدیت وزن های اصلاحی روش پس انتشار خطا استفاده شده است و سپس پایداری آن هم توسط تئوری لیاپانوف تضمین شده است. همین طور می توان به این قانون آموزش، جمله هایی را هم اضافه که پایداری آن را مقاوم تر کند.

در این گزارش، یک رویتگر تطبیقی شبکه عصبی برای سیستم عمومی غیرخطی چندمتغیره پیشنهاد شده است. این شبکه عصبی از نوع غیرخطی نسبت به پارامتر است که نوع دقیق تر آن هم شبکه ی عصبی RBF است. قانون آپدیت وزن ها برای این شبکه عصبی تشکیل شده است از یک جمله ی مربوط به روش پس انتشار خطا و همین طور یک جمله ی اصلاحی تجربی که پایداری مقاوم تر را تضمین کند. شرط SPR هم از روی معادله ی خطای خروجی برداشته شده است. در انتها، در شبیه سازی، این رویتگر را به یک سیستم بازوی ربات انعطاف پذیر تک پیونده وصل می کنیم و نتایج شبیه سازی را مشاهده می کنیم.

# فصل ۲

# تعریف مسئله

در این فصل تعدادی تعریف های ریاضی مورد نیاز در پروژه ارائه می شوند و سپس یک مدل رویتگر عصبی_تطبیقی برای رسیدن به اهداف مورد نظر پیشنهاد می شود.

## ۲_۱ تعاریف ریاضی

تعاریف ریاضی مورد نیاز برای سهولت درک مطالب، در ادامه آمده اند.

نرم یک بردار $x \in R^n$ و نرم طیفی یک ماتریس $A \in R^{m\times n}$ به صورت زیر تعریف می شوند.

$$\|x\| = \sqrt{x^T x} \quad \|A\|_s = \sqrt{\lambda_{\max}\left[A^T A\right]} \tag{۲_۱}$$

که در آن، $\lambda_{\max}[\cdot]$ نشان دهنده ی بزرگترین مقدار ویژه ی ماتریس مثبت معین یا مثبت نیمه معین $[\cdot]$ می باشد. همین طور کمترین مقدار ویژه ی این ماتریس را نیز با $\lambda_{\min}[\cdot]$ نشان می دهیم. اگر $A = [a_{ij}]$ و $B \in R^{m\times n}$، نرم فروبنیوس[۱] به صورت زیر تعریف می شود.

$$\|A\|_F^۲ = \mathrm{tr}\left(A^T A\right) = \Sigma a_{ij}^۲ \tag{۲_۲}$$

که در آن، $\mathrm{tr}(\cdot)$ به معنای اثر[۲] ماتریس است. ضرب داخلی مرتبط با رابطه ی بالا هم به صورت زیر

[۱] Frobenius
[۲] trace

تعریف می شود.

$$\langle A,B\rangle_F = \mathrm{tr}\left(A^T B\right) \tag{۲–۳}$$

فضای $L_\infty$ یک سیگنال کراندار به صورت زیر تعریف می شود.

$$x(t) \in L_\infty \text{ ess if } \sup_t |x(t)| < \infty \tag{۲–۴}$$

حال اگر $x \in L_\infty$، نرم $L_\infty$ سیگنال $x(t)$ به صورت زیر تعریف می شود.

$$\|x\|_{L_\infty} = \operatorname{ess\,sup}_t |x(t)| \tag{۲–۵}$$

## ۲–۲ رویتگر عصبی–تطبیقی پیشنهاد شده

در این قسمت مدل کلی سیستم تعریف می شود و بر اساس آن اول مدلی کلی برای رویتگر پیشنهاد می شود و سپس در این مدل یک رابطه ی شبکه ی عصبی گنجانده می شود و در آخر هم معادلات خطا به صورت کلی بدست می آیند.

اگر یک سیستم چندمتغیره[۳] غیرخطی داشته باشیم، مدل جامع آن به صورت زیر است.

$$\begin{aligned} \dot{x}(t) &= f(x,u) \\ y(t) &= Cx(t) \end{aligned} \tag{۲–۶}$$

که در آن، $u \in R^{m_u}$ بردار ورودی سیستم، $y \in R^{m_y}$ بردار خروجی سیستم، $x \in R^n$ بردار حالت های سیستم و $f$ یک تابع برداری غیرخطی نامعلوم می باشد. برای ادامه ی کار لازم است که این سیستم غیرخطی در رابطه ی ۲–۶ رویت پذیر باشد. یک فرض مهم دیگری هم می تونیم داشته باشیم و آن این است که سیستم حلقه بسته پایدار است. به عبارت دیگر حالت های سیستم $x(t)$ در فضای $L_\infty$ کراندار هستند. حال اگر $Ax$ را از معادله ی ۲–۶ کم و زیاد کنیم، رابطه ی زیر حاصل می شود.

$$\begin{aligned} \dot{x}(t) &= Ax + g(x,u) \\ y(t) &= Cx(t) \end{aligned} \tag{۲–۷}$$

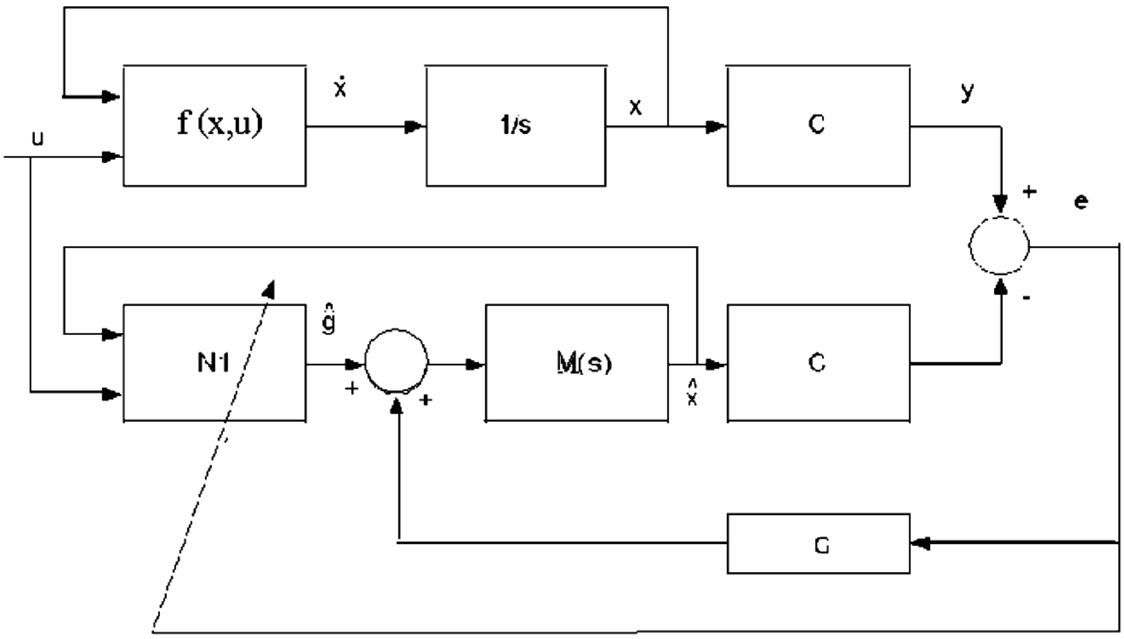
---

[۳] MIMO

که در آن $A$ یک ماتریس هرویتز است، جفت ماتریس های $(C, A)$ رویتگر هستند و $g(x,u) = f(x,u) - Ax$ است. حال بر اساس مدل سیستم گفته شده، مدل رویتگر آن به صورت زیر انتخاب می شود.

$$\begin{aligned} \dot{\hat{x}}(t) &= A\hat{x} + \hat{g}(\hat{x}, u) + G(y - C\hat{x}) \\ \hat{y}(t) &= C\hat{x}(t) \end{aligned} \qquad (۲\text{ـ}۸)$$

که در آن، $\hat{x}$ نشان دهنده ی بردار حالت رویتگر می باشد و گین رویتگر $G \in R^{n\times m_y}$ طوری انتخاب می شود که ماتریس $A - GC$ هرویتز باشد. این گین حتماً وجود دارد چرا که $A$ طوری انتخاب شده است که $(C, A)$ رویت پذیر باشد. شمای کلی این رویتگر در شکل ۲ـ۱ آمده است.




شکل ۲ـ۱: شمای ساختار کلی رویتگر پیشنهاد شده همراه با سیستم

در این شکل، $\hat{x}$ بردار حالت های مدل رویتگر بازگشتی در رابطه ی ۲ـ۸ را نشان می دهد. همین طور، $M(s) = (sI - A)^{-1}$ یک ماتریس $n \times n$ است که المان های آن تابع تبدیل های پایدار هستند چرا که $A$ خود ماتریس هرویتز است. برای اینکه بتوانیم یک رویتگر شبکه ی عصبی طراحی کنیم، باید یک شبکه ی عصبی به کار گیریم تا قسمت غیرخطی سیستم را شناسایی کند و در کنار آن یک رویتگر کلاسیک هم در نظر بگیریم تا حالت ها را تقریب بزنند. به علت ماهیت غیر خطی سیستم، در اینجا از شبکه ی عصبی RBF برای قسمت مستقیم شبکه عصبی استفاده شده است که تابع فعال سازی لایه ی مخفی آن غیرخطی و تابع فعال سازی لایه ی خروجی آن خطی می باشد. بر اساس تجربه ها و اطلاعات

قبلی، یک شبکه ی عصبی ی سه لایه می تواند هر سیستم غیرخطی را با هر درجه ی غیرخطی، تقریب بزند. این موضوع توسط بسیاری از محققان اثبات شده است به صورتی که برای هر $x$ محدود شده در مجموعه ی پیوسته ی $S$ به صورتی که $x \in R^n$ باشد، با تعدادی زیاد و کافی از نورون های لایه ی مخفی، وزن ها و سطح آستانه هایی وجود دارد که این شبکه هر تابع پیوسته در مجموعه ی پیوسته ی $S$ را به صورت رابطه ی زیر تقریب بزند.

$$g(x,u) = W\sigma(V\bar{x}) + \epsilon(x) \qquad (۲-۹)$$

که در آن $W$ و $V$ ماتریس های وزن های به ترتیب لایه ی خروجی و لایه ی مخفی می باشند. $\bar{x} = \begin{bmatrix} x & , & u \end{bmatrix}^T$ ورودی شبکه عصبی می باشد که ترکیب بردار حالت ها و ورودی سیستم است. $\epsilon(x)$ هم خطای تقریب شبکه عصبی است که خود این خطا کراندار است. $\sigma(.)$ تابع فعال سازی نورون های لایه ی مخفی می باشد که معمولاً به صورت تابع سیگموید در نظر گرفته می شود و تابع فعال سازی سیگمویدی که در اینجا در نظر گرفته شده، برای هر نورون لایه ی مخفی به صورت زیر است.

$$\sigma_i\left(V_i\bar{x}\right) = \frac{۲}{۱ + \exp^{-۲V_i\bar{x}}} - ۱ \qquad (۲-۱۰)$$

که رد آن، $V_i$ ردیف i م ماتریس $V$ و $\sigma_i\left(V_i\bar{x}\right)$ المان یا ردیف i م بردار $\sigma(V\bar{x})$ می باشد. فرض می کنیم برای ماتریس های وزن ها و تابع فعال سازی نورون های لایه ی مخفی، کران های بالا به صورت روابط زیر وجود دارد.

$$\|W\|_F \leqslant W_M \qquad (۲-۱۱)$$

$$\|V\|_F \leqslant V_M \qquad (۲-۱۲)$$

$$\|\sigma(V\bar{x})\| \leqslant \sigma_m \qquad (۲-۱۳)$$

اگر تابع $g$ را با این شبکه عصبی تقریب بزنیم مدل خروجی رویتگر بدون در نظر گرفتن خطای تقریب به صورت رابطه ی زیر می باشد.

$$\hat{g}(\hat{x},u) = \hat{W}\sigma(\hat{V}\hat{\bar{x}}) \qquad (۲-۱۴)$$

اکنون، با در نظر گرفتن روابط قبلی، مدل کلی رویتگر پیشنهاد شده به صورت رابطه ی زیر بدست می آید.

$$\begin{aligned} \dot{\hat{x}}(t) &= A\hat{x} + \hat{W}\sigma(\hat{V}\hat{\bar{x}}) + G(y - C\hat{x}) \\ \hat{y}(t) &= C\hat{x}(t) \end{aligned} \qquad (۲-۱۵)$$

اگر خطای تقریب حالت را به صورت $\tilde{x} = x - \hat{x}$ تعریف کنیم، با استفاده از روابط ۲ـ۱۵، ۲ـ۱۴، ۲ـ۹ و ۲ـ۸ معادلات خطای رویتگر به صورت زیر بدست می آیند.

$$\begin{aligned} \dot{\tilde{x}}(t) =& Ax + W\sigma(V\bar{x}) - A\hat{x} - \hat{W}\sigma(\hat{V}\hat{x}) \\ & - G(Cx - C\hat{x}) + \epsilon(x) \\ \tilde{y}(t) =& C\tilde{x}(t) \end{aligned} \tag{۲ـ۱۶}$$

حال اگر از این معادله، عبارت $W\sigma(\hat{V}\hat{\bar{x}})$ را کم و اضافه کنیم، رابطه ی زیر برای معادلات خطای رویتگر بدست می آید.

$$\begin{aligned} \dot{\tilde{x}}(t) &= A_c\tilde{x} + \tilde{W}\sigma(\hat{V}\hat{\bar{x}}) + w(t) \\ \tilde{y}(t) &= C\tilde{x}(t) \end{aligned} \tag{۲ـ۱۷}$$

که در آن،

$$\tilde{W} = W - \hat{W} \tag{۲ـ۱۸}$$

$$A_c = A - GC \tag{۲ـ۱۹}$$

$$w(t) = W[\sigma(V\bar{x}) - \sigma(\hat{V}\hat{\bar{x}})] + \epsilon(x) \tag{۲ـ۲۰}$$

به علاوه، $A_c$ همان طور که قبلاً گفته شد، باید هرویتز انتخاب شود و $\|w(t)\| \leqslant \bar{w}$ به عنوان یک جمله ی اغتشاش کراندار در نظر گرفته می شود با در نظر گرفتن عدد ثابت مثبت $\bar{w}$. دلیل این کراندار بودن جمله ی اغتشاش هم به ماهیت تابع سیگموید و کراندار بودن ماتریس های وزن ایده آل شبکه عصبی یا $(V, W)$ می باشد.

# فصل ۳

# قانون آپدیت وزن ها و تحلیل پایداری

در این فصل قانون آپدیت وزن ها را بر اساس روش اصلاح شده ی پس انتشار خطا بدست می آوریم. سپس پایداری رویتگر با این قانون و خطای تخمین را با استفاده از تئوری لیاپانوف اثبات می کنیم.

## ۳ـ۱ قانون آپدیت وزن ها

تا الان، ساختار شبکه ی عصبی معین شده است و فقط کافی است روشی برای آموزش شبکه انتخاب کنیم و به عبارت دیگر باید قانون آپدیت وزن ها را مشخص کنیم. در این قسمت قانون آپدیت وزن ها طوری ارائه می گردد که پایداری رویتگر را تضمین کند. همین طور، این قانون تطبیقی نباید تحت هیچ گونه قید محکمی محدود گردد و یا پیچیده شود. یکی از معروف ترین الگوریتم هایی که برای آموزش تطبیقی رویتگر شبکه عصبی استفاده می شود الگوریتم پس انتشار خطا یا BP[۱] می باشد. در کل، این الگوریتم به علت ساده بودن در ساختار، در زمینه های کاربردی زیادی به کار می رود که می توان به رده بندی، تشخیص الگو، تعیین هویت، رویت کردن و مسائل کنترلی اشاره کرد.

هر چند، در بسیاری از کار های تحقیقی اثبات ریاضی پایداری شبکه عصبی لحاظ نشده است و در این قسمت یک قانون آپدیت وزن های جدید ارائه می شود و در قسمت بعد، پایداری آن توسط روش مستقیم لیاپانوف اثبات می شود. این قانون آپدیت وزن ها بر اساس اصلاح شده ی الگوریتم BP می

[۱] Back Propagation Error

باشد به علاوه ی یک جمله ی اصلاحیه ی تجربی[۲] که مقاوم بودن این پایداری این قانون را تضمین کند. قضیه ی زیر، جواب اصلی این قسمت و قسمت بعد را نشان می دهد که در ادامه ما آن را اثبات می کنیم.

**قضیه‌ی ۳ـ۱** *فرض کنید مدل سیستم به صورت رابطه ی ۲ـ۶ و مدل رویتگر پیشنهاد شده ی آن به صورت رابطه ی ۲ـ۱۵ باشد. شبکه ی عصبی هم به صورت NLPNN [۳] یا شبکه ی عصبی غیرخطی نسبت به پارامتر ها است. حال اگر وزن های NLPNN به صورت قانون زیر آپدیت شوند،*

$$\dot{\hat{W}} = -\eta_1 \left(\tilde{y}^T C A_c^{-1}\right)^T (\sigma(\hat{V}\hat{\bar{x}}))^T - \rho_1 \|\tilde{y}\| \hat{W} \quad (۳ـ۱)$$

$$\begin{aligned}\dot{\hat{V}} = &- \eta_2 \left(\tilde{y}^T C A_c^{-1} \hat{W}(I - \Lambda(\hat{V}\hat{\bar{x}}))\right)^T \operatorname{sgn}(\hat{\bar{x}})^T \\ &- \rho_2 \|\tilde{y}\| \hat{V}\end{aligned} \quad (۳ـ۲)$$

*که در آن،*

$$\Lambda(\hat{V}\hat{\bar{x}}) = \operatorname{diag}\left\{\sigma_i^2\left(\hat{V}_i\hat{\bar{x}}\right)\right\}, i = 1, 2, \ldots, m \quad (۳ـ۳)$$

$$\operatorname{sgn}(\hat{\bar{x}}) = \begin{cases} 1, & \textit{for } \hat{\bar{x}} > 0 \\ 0, & \textit{for } \hat{\bar{x}} = 0 \\ -1, & \textit{for } \hat{\bar{x}} < 0 \end{cases} \quad (۳ـ۴)$$

*آنگاه، $\tilde{x}, \tilde{W}, \tilde{V}, \tilde{y} \in L_\infty$ یا خطاهای تخمین حالت، خطای های وزن ها و خطای خروجی، همه کراندار هستند. در این معادلات $\eta_1, \eta_2 > 0$ نرخ های یادگیری هستند و $\rho_1, \rho_2$ هم اعداد مثبت کوچک می باشند. تابع هزینه ی در نظر گرفته شده هم به صورت زیر می باشد.*

$$J = (1/2)\left(\tilde{y}^T \tilde{y}\right) \quad (۳ـ۵)$$

اثبات:

ابتدا قانون کلی اصلاح شده ی BP را به صورت زیر می نویسیم.

$$\dot{\hat{W}} = -\eta_1 \left(\frac{\partial J}{\partial \hat{W}}\right) - \rho_1 \|\tilde{y}\| \hat{W} \quad (۳ـ۶)$$

---

[۲] e-modification term

[۳] Non-linear in Parameter Neural Network

$$\dot{\hat{V}} = -\eta_{۲}\left(\frac{\partial J}{\partial \hat{V}}\right) - \rho_{۲}\|\tilde{y}\|\hat{V} \qquad (۳\text{ـ}۷)$$

که در هر معادله، جمله ی اول جمله ی BP است و جمله ی دوم جمله ی اصلاحیه ی تجربی است تا مقاوم بودن قانون را تضمین کند و در معادلات تعدیل ایجاد کند. حال، خروجی های لایه های مخفی و خروجی هم به ترتیب به صورت زیر می باشند.

$$\mathrm{net}_{\hat{v}} = \hat{V}\hat{\bar{x}} \qquad (۳\text{ـ}۸)$$

$$\mathrm{net}_{\hat{w}} = \hat{W}\sigma(\hat{V}\hat{\bar{x}}) \qquad (۳\text{ـ}۹)$$

سپس، مشتق های تابع هزینه نسبت به وزن ها یعنی $(\partial J/\partial \hat{W})$ و $(\partial J/\partial \hat{V})$ به صورت قاعده ی زنجیره ای زیر محاسبه می شوند.

$$\begin{aligned} \frac{\partial J}{\partial \hat{W}} &= \frac{\partial J}{\partial\,\mathrm{net}_{\hat{w}}}\cdot\frac{\partial\,\mathrm{net}_{\hat{w}}}{\partial \hat{W}} \\ \frac{\partial J}{\partial \hat{V}} &= \frac{\partial J}{\partial\,\mathrm{net}_{\hat{v}}}\cdot\frac{\partial\,\mathrm{net}_{\hat{v}}}{\partial \hat{V}} \end{aligned} \qquad (۳\text{ـ}۱۰)$$

که در آن،

$$\begin{aligned} \frac{\partial J}{\partial\,\mathrm{net}_{\hat{w}}} &= \frac{\partial J}{\partial \tilde{y}}\cdot\frac{\partial \tilde{y}}{\partial \hat{x}}\cdot\frac{\partial \hat{x}}{\partial\,\mathrm{net}_{\hat{w}}} = -\tilde{y}^T C\cdot\frac{\partial \hat{x}}{\partial\,\mathrm{net}_{\hat{w}}} \\ &= -\tilde{x}^T C^T C\cdot\frac{\partial \hat{x}}{\partial\,\mathrm{net}_{\hat{w}}} \\ \frac{\partial J}{\partial\,\mathrm{net}_{\hat{v}}} &= \frac{\partial J}{\partial \tilde{y}}\cdot\frac{\partial \tilde{y}}{\partial \hat{x}}\cdot\frac{\partial \hat{x}}{\partial\,\mathrm{net}_{\hat{v}}} = -\tilde{y}^T C\cdot\frac{\partial \hat{x}}{\partial\,\mathrm{net}_{\hat{v}}} \\ &= -\tilde{x}^T C^T C\cdot\frac{\partial \hat{x}}{\partial\,\mathrm{net}_{\hat{v}}} \end{aligned} \qquad (۳\text{ـ}۱۱)$$

$$\begin{aligned} \frac{\partial\,\mathrm{net}_{\hat{w}}}{\partial \hat{W}} &= \sigma(\hat{V}\hat{\bar{x}}) \\ \frac{\partial\,\mathrm{net}_{\hat{v}}}{\partial \hat{V}} &= \hat{\bar{x}} \end{aligned} \qquad (۳\text{ـ}۱۲)$$

اکنون، با استفاده از رابطه های ۲ـ۱۵، ۳ـ۸ و ۳ـ۹ و مشتق گیری از آنها می توانیم به رابطه ی زیر دست پیدا کنیم.

$$\begin{aligned} \frac{\partial \dot{\hat{x}}(t)}{\partial\,\mathrm{net}_{\hat{w}}} &= A_c\frac{\partial \hat{x}}{\partial\,\mathrm{net}_{\hat{w}}} + I \\ \frac{\partial \dot{\hat{x}}(t)}{\partial\,\mathrm{net}_{\hat{v}}} &= A_c\frac{\partial \hat{x}}{\partial\,\mathrm{net}_{\hat{v}}} + \hat{W}(I - \Lambda(\hat{V}\hat{\bar{x}})) \end{aligned} \qquad (۳\text{ـ}۱۳)$$

معادلات بالا از نوع سیستم های دینامیکی غیرخطی می باشند و برای اینکه گرادیان های $(\partial\hat{x}/\partial\,\mathrm{net}_{\hat{w}})$ و $(\partial\hat{x}/\partial\,\mathrm{net}_{\hat{v}})$ را بدست آوریم باید از پس انتشار خطا در زمان[۴] که پس انتشار خطای دینامیکی نیز می باشد استفاده کنیم. ولی این کار به پیچیدگی رویتگر می افزاید و پیاده سازی بلادرنگ این راهکار را بسیار سخت می کند. در نتیجه پیشنهاد می شود که از تقریب استاتیکی این گرادیان ها استفاده شود به طوری که $\dot{\hat{x}} = 0$ باشد و این گونه، گرادیان ها به صورت زیر تقریب زده می شوند.

$$\begin{aligned}\frac{\partial\hat{x}}{\partial\,\mathrm{net}_{\hat{w}}} &\approx -A_c^{-1}\\ \frac{\partial\hat{x}}{\partial\,\mathrm{net}_{\hat{v}}} &\approx -A_c^{-1}\hat{W}(I-\Lambda(\hat{V}\bar{\hat{x}}))\end{aligned} \tag{۳ـ۱۴}$$

اکنون، با استفاده از روابط ۳ـ۱۴، ۳ـ۱۰، ۳ـ۱۱ و ۳ـ۱۲ قانون های یادگیری ۳ـ۶ و ۳ـ۷ به صورت زیر نوشته می شوند.

$$\begin{aligned}\dot{\hat{W}} =& -\eta_1\left(\tilde{x}^T C^T C A_c^{-1}\right)^T(\sigma(\hat{V}\bar{\hat{x}}))^T\\ &-\rho_1\|C\tilde{x}\|\hat{W}\end{aligned} \tag{۳ـ۱۵}$$

$$\begin{aligned}\dot{\hat{V}} =& -\eta_2\left(\tilde{x}^T C^T C A_c^{-1}\hat{W}(I-\Lambda(\hat{V}\bar{\hat{x}}))\right)^T\bar{\hat{x}}^T\\ &-\rho_2\|C\tilde{x}\|\hat{V}\end{aligned} \tag{۳ـ۱۶}$$

سپس، اگر بخواهیم معادلات خطای این قوانین یادگیری را بنویسیم و $\tilde{W} = W - \hat{W}$ و $\tilde{V} = V - \hat{V}$ باشد، معادلات زیر بدست می آیند. دقت کنید که مشتق خطای وزن ها مساوی منفی مشتق خود وزن ها است.

$$\dot{\tilde{W}} = \eta_1\left(\tilde{x}^T C^T C A_c^{-1}\right)^T(\sigma(\hat{V}\bar{\hat{x}}))^T + \rho_1\|C\tilde{x}\|\hat{W} \tag{۳ـ۱۷}$$

$$\begin{aligned}\dot{\tilde{V}} =& \eta_2\left(\tilde{x}^T C^T C A_c^{-1}\hat{W}(I-\Lambda(\hat{V}\bar{\hat{x}}))\right)^T\bar{\hat{x}}^T\\ &+\rho_2\|C\tilde{x}\|\hat{V}\end{aligned} \tag{۳ـ۱۸}$$

برای اینکه تحلیل پایداری را آسان تر کنیم به جای $\bar{\hat{x}}$، $\mathrm{sgn}(\bar{\hat{x}})$ را قرار می دهیم و رابطه ی زیر بدست می آید که در آن بر عکس خود بردار ورودی شبکه که کراندار نیست علامت آن حتماً کراندار است که

[۴] backpropagation in time

در ادامه این خصوصیت مفید واقع می شود. در ضمن، اگر از علامت $\hat{\hat{x}}$ استفاده کنیم، تضمین می شود که قانون آپدیت وزن ها در جهت درست حرکت کند.

$$\dot{\hat{V}} = \eta_2 \left(\tilde{x}^T C^T C A_c^{-1} \hat{W}(I - \Lambda(\hat{V}\hat{\hat{x}}))\right)^T \operatorname{sgn}(\hat{\hat{x}})^T + \rho_2 \|C\tilde{x}\| \hat{V} \tag{۳-۱۹}$$

اگر بخواهیم قوانین نهایی آپدیت وزن ها را با ملاحظات قبلی، دوباره بنویسیم به صورت روابط زیر بدست می آیند.

$$\dot{\hat{W}} = -\eta_1 \left(\tilde{y}^T C_1 A_c^{-1}\right)^T (\sigma(\hat{V}\hat{\hat{x}}))^T - \rho_1 \|\tilde{y}\| \hat{W} \tag{۳-۲۰}$$

$$\dot{\hat{V}} = -\eta_2 \left(\tilde{y}^T C_1 A_c^{-1} \hat{W}(I - \Lambda(\hat{V}\hat{\hat{x}}))\right)^T \operatorname{sgn}(\hat{\hat{x}})^T - \rho_2 \|\tilde{y}\| \hat{V} \tag{۳-۲۱}$$

در این روابط به جای خطای حالت ها از خطای خروجی استفاده شده است که در دسترس ما هست. همین طور به جای $C$ از ماتریس $C_1$ استفاده شده است چرا که بیشتر وقت ها همه ی حالت ها در خروجی ظاهر نمی شوند و مقادیر آنها ماتریس $C$ باید صفر شود و برای اینکه الگوریتم پس انتشار خطا را کند نکند در مراحل آموزش، این گونه آن را از سیستم اصلی مجزا می کنیم.

## ۳ـ۲ تحلیل پایداری

برای تحلیل پایداری رویتگر شبکه عصبی، تابع لیاپانوف زیر را در نظر می گیریم.

$$L = \frac{1}{2}\tilde{x}^T P \tilde{x} + \frac{1}{2}\operatorname{tr}\left(\tilde{W}^T \tilde{W}\right) + \frac{1}{2}\operatorname{tr}\left(\tilde{V}^T \tilde{V}\right) \tag{۳-۲۲}$$

که در آن $P = P^T$ یک ماتریس مثبت معین متقارن است که در رابطه ی زیر با شرط ماتریس هرویتز $A_c$ و یک ماتریس مثبت معین $Q$ صدق می کند.

$$A_c^T P + P A_c = -Q \tag{۳-۲۳}$$

مشتق زمانی این تابع لیاپانوف به صورت رابطه ی زیر می شود.

$$\dot{L} = \frac{1}{2}\dot{\tilde{x}}^T P \tilde{x} + \frac{1}{2}\tilde{x}^T P \dot{\tilde{x}} + \operatorname{tr}\left(\tilde{W}^T \dot{\tilde{W}}\right) + \operatorname{tr}\left(\tilde{V}^T \dot{\tilde{V}}\right) \tag{۳-۲۴}$$

اکنون با اعمال روابط ۲ـ۱۷، ۳ـ۱۷، ۳ـ۱۹ و ۳ـ۲۳ بر رابطه ی ۳ـ۲۴، رابطه ی زیر بدست می آید.

$$\begin{aligned}\dot{L} = &-\frac{1}{2}\tilde{x}^T Q\tilde{x} + \tilde{x}^T P(\tilde{W}\sigma(\hat{V}\hat{\tilde{x}}) + w)\\ &+ \operatorname{tr}\left(\tilde{W}^T l_1 \tilde{x}\sigma(\hat{V}\hat{\tilde{x}})^T + \tilde{W}^T\rho_1\|C\tilde{x}\|(W-\tilde{W})\right)\\ &+ \operatorname{tr}\left(\tilde{V}^T(I-\Lambda(\hat{V}\hat{\tilde{x}}))^T\hat{W}^T l_2\tilde{x}\,\operatorname{sgn}(\hat{\tilde{x}})^T\right.\\ &\quad \left.+\tilde{V}^T\rho_2\|C\tilde{x}\|(V-\tilde{V})\right)\end{aligned} \tag{۳ـ۲۵}$$

که در آن،

$$l_1 = \eta_1 A_c^{-T}C^TC \quad , \quad l_2 = \eta_2 A_c^{-T}C^TC \tag{۳ـ۲۶}$$

بر اساس کراندار بودن وزن ها و تابع فعال سازی، نامعادلات زیر برای بعضی عبارات در رابطه ی قبل بدست می آیند.

$$\begin{aligned}\operatorname{tr}\left(\tilde{W}^T(W-\tilde{W})\right) &\leqslant W_M\|\tilde{W}\| - \|\tilde{W}\|^2\\ \operatorname{tr}\left(\tilde{V}^T(V-\tilde{V})\right) &\leqslant V_M\|\tilde{V}\| - \|\tilde{V}\|^2\\ \operatorname{tr}\left(\tilde{W}^T l_1\tilde{x}\sigma(\hat{V}\hat{\tilde{x}})^T\right) &\leqslant \sigma_m\left\|\tilde{W}^T\right\|\|l_1\|\,\|\tilde{x}\|\end{aligned} \tag{۳ـ۲۷}$$

ضمناً، آخرین نامعادله توسط قانون زیر برای دو بردار ستونی $A$ و $B$ بدست می آید.

$$\operatorname{tr}\left(AB^T\right) = B^TA \tag{۳ـ۲۸}$$

حال، با استفاده از قانون اثر ماتریس بالا و نامعادلات $\|\hat{W}\| = \|W-\tilde{W}\| \leqslant W_M + \|\tilde{W}\|,\ 1-\sigma_m^2 \leqslant 1$ ، نامعادله ی زیر بدست می آید.

$$\begin{aligned}&\operatorname{tr}\left(\tilde{V}^T(I-\Lambda(\hat{V}\hat{\tilde{x}}))^T\hat{W}^T l_2\tilde{x}\,\operatorname{sgn}(\hat{\tilde{x}})^T\right)\\ &\quad\leqslant \|\tilde{V}\|\left(W_M + \|\tilde{W}\|\right)\|l_2\|\,\|\tilde{x}\|\end{aligned} \tag{۳ـ۲۹}$$

سپس با استفاده از روابط ۳ــ۲۷ و ۳ــ۲۹ رابطه ی زیر حاصل می شود.

$$\begin{aligned}\dot{L} \leqslant & -\frac{1}{2}\lambda_{\min}(Q)\|\tilde{x}\|^2 + \|\tilde{x}\|\|P\|\left(\|\tilde{W}\|\sigma_m + \bar{w}\right) \\ & + \sigma_m\|\tilde{W}\|\,\|l_1\|\,\|\tilde{x}\| + \left(W_M\|\tilde{W}\| - \|\tilde{W}\|^2\right)\rho_1\|C\|\|\tilde{x}\| \\ & + \|\tilde{V}\|\,\|l_2\|\left(W_M + \|\tilde{W}\|\right)\|\tilde{x}\| \\ & + \rho_2\|C\|\|\tilde{x}\|\left(V_M\|\tilde{V}\| - \|\tilde{V}\|^2\right) \\ = & F\end{aligned} \tag{۳ــ۳۰}$$

اکنون، باید برای برای $\|\tilde{x}\|$ کران بالا پیدا کنیم که باعث بشود مستقل از خطای وزن های شبکه عصبی، مشتق تابع لیاپانوف منفی شود. برای این کار، $K_1 = (\|l_2\|/2)$ را تعریف می کنیم و از سمت راست نامعادله $K_1^2\|\tilde{W}\|^2\|\tilde{x}\|$ و $\|\tilde{V}\|^2\|\tilde{x}\|$ را کم و اضافه می کنیم که در نتیجه به رابطه ی زیر می رسیم.

$$\begin{aligned}F = & -\frac{1}{2}\lambda_{\min}(Q)\|\tilde{x}\|^2 \\ & + \Big[\|P\|\bar{w} - \left(\rho_1\|C\| - K_1^2\right)\|\tilde{W}\|^2 \\ & + \left(\|P\|\sigma_m + \sigma_m\|l_1\| + \rho_1\|C\|W_M\right)\|\tilde{W}\| \\ & + \left(\rho_2\|C\|V_M + \|l_2\|W_M\right)\|\tilde{V}\| \\ & - \left(\rho_2\|C\| - 1\right)\|\tilde{V}\|^2 - \left(K_1\|\tilde{W}\| - \|\tilde{V}\|\right)^2\Big]\|\tilde{x}\|.\end{aligned} \tag{۳ــ۳۱}$$

حال، عبارات زیر را نیز تعریف می کنیم.

$$\begin{aligned}K_2 &= \frac{\rho_1 W_M\|C\| + \sigma_m\|l_1\| + \|P\|\sigma_m}{2\left(\rho_1\|C\| - K_1^2\right)} \\ K_3 &= \frac{\rho_2\|C\|V_M + \|l_2\|W_M}{2\left(\rho_2\|C\| - 1\right)}\end{aligned} \tag{۳ــ۳۲}$$

سپس، به رابطه ی ۳ــ۳۱ عبارات $K_2^2\|\tilde{x}\|$ و $K_3^2\|\tilde{x}\|$ را اضافه و کم می کنیم که رابطه ی زیر را نتیجه می دهد.

$$\begin{aligned}F = & -\frac{1}{2}\lambda_{\min}(Q)\|\tilde{x}\|^2 \\ & + [\|P\|\bar{w} + \left(\rho_1\|C\| - K_1^2\right)K_2^2 + \left(\rho_2\|C\| - 1\right)K_3^2 \\ & - \left(\rho_1\|C\| - K_1^2\right)\left(K_2 - \|\tilde{W}\|\right)^2 - \left(\rho_2\|C\| - 1\right) \\ & \times\left(K_3 - \|\tilde{V}\|\right)^2 - \left(K_1\|\tilde{W}\| - \|\tilde{V}\|\right)^2\Big]\|\tilde{x}\|\end{aligned} \tag{۳ــ۳۳}$$

حال، اگر فرض کنیم $\rho_1 \geqslant (K_1^2/\|C\|)$ و $\rho_2 \geqslant (1/\|C\|)$ سه جمله ی آخر رابطه ی بالا حتماً منفی است چرا که توان ۲ هم دارد. در نتیجه، رابطه ی بالا به صورت نامعادله ی زیر تبدیل می شود.

$$\begin{aligned} F &\leqslant -\frac{1}{2}\lambda_{\min}(Q)\|\tilde{x}\|^2 + \|\tilde{x}\| \\ &\times \left(\|P\|\bar{w} + \left(\rho_1\|C\| - K_1^2\right)K_2^2 + \left(\rho_2\|C\| - 1\right)K_3^2\right) \end{aligned} \tag{۳ـ۳۴}$$

بنابراین، با یک فاکتور گیری و جابجایی ساده در نامعادله ی بالا، شرط زیر برای $\|\tilde{x}\|$ بدست می آید که منفی معین بودن $\dot{L}$ را تضمین می کند.

$$\|\tilde{x}\| > \frac{2\left(\|P\|\bar{w} + \left(\rho_1\|C\| - K_1^2\right)K_2^2 + \left(\rho_2\|C\| - 1\right)K_3^2\right)}{\lambda_{\min}(Q)} = b \tag{۳ـ۳۵}$$

و این در واقع یک توپ است با شعاع $b$ که $\dot{L}$ در بیرون توپ یعنی $\chi = \{\tilde{x} \mid \|\tilde{x}\| > b\}$ منفی معین است و بالاخره $\tilde{x}$ هم به صورت یکنواخت کراندار شد. در مورد ناحیه ی داخلی توپ هم می توان گفت که به سمت سطح توپ که یک سیکل حدی است، همگرا است. یعنی در این ناحیه، بیشتر شدن $\dot{L}$ برای مقادیر کم $\|\tilde{x}\|$ باعث می شود که $\tilde{x}$ و $L$ افزایش یابند و $\tilde{x}$ را به خارج از توپ هدایت می کند که در آنجا $\dot{L}$ منفی است و بعد به سمت توپ هدایت می شود. این اثبات می کند که $\tilde{x}$ به صورت جهانی پایدار و کراندار است. برای اینکه کراندار بودن $\tilde{W}$ و $\tilde{V}$ را هم نشان دهیم، رابطه های ۳ـ۱۷ و ۳ـ۱۹ را به صورت زیر بازنویسی می کنیم.

$$\begin{aligned} \dot{\hat{W}} &= f_1(\tilde{x}, \hat{V}) + \rho_1\|C\tilde{x}\|\hat{W} \\ &= f_1(\tilde{x}, \hat{V}) + \alpha_1 W - \alpha_1\tilde{W} \end{aligned} \tag{۳ـ۳۶}$$

$$\begin{aligned} \dot{\hat{V}} &= f_2(\tilde{x}, \hat{W}, \hat{V}) + \rho_2\|C\tilde{x}\|\hat{V} \\ &= f_2(\tilde{x}, \hat{W}, \hat{V}) + \alpha_2 V - \alpha_2\tilde{V} \end{aligned} \tag{۳ـ۳۷}$$

که در آن،

$$f_1(\tilde{x}, \hat{V}) = \eta_1\left(\tilde{x}^T C^T C A_c^{-1}\right)^T (\sigma(\hat{V}\hat{\bar{x}}))^T \tag{۳ـ۳۸}$$

$$\begin{aligned} f_1(\tilde{x}, \hat{V}) &= \eta_1\left(\tilde{x}^T C^T C A_c^{-1}\right)^T (\sigma(\hat{V}\hat{\bar{x}}))^T \\ f_2(\tilde{x}, \hat{W}, \hat{V}) &= \eta_2\left(\tilde{x}^T C^T C A_c^{-1}\hat{W}(I - \Lambda(\hat{V}\hat{\bar{x}}))\right)^T \operatorname{sgn}(\hat{\bar{x}})^T \\ \alpha_1 &= \rho_1\|C\tilde{x}\| \\ \alpha_2 &= \rho_2\|C\tilde{x}\| \end{aligned} \tag{۳ـ۳۹}$$

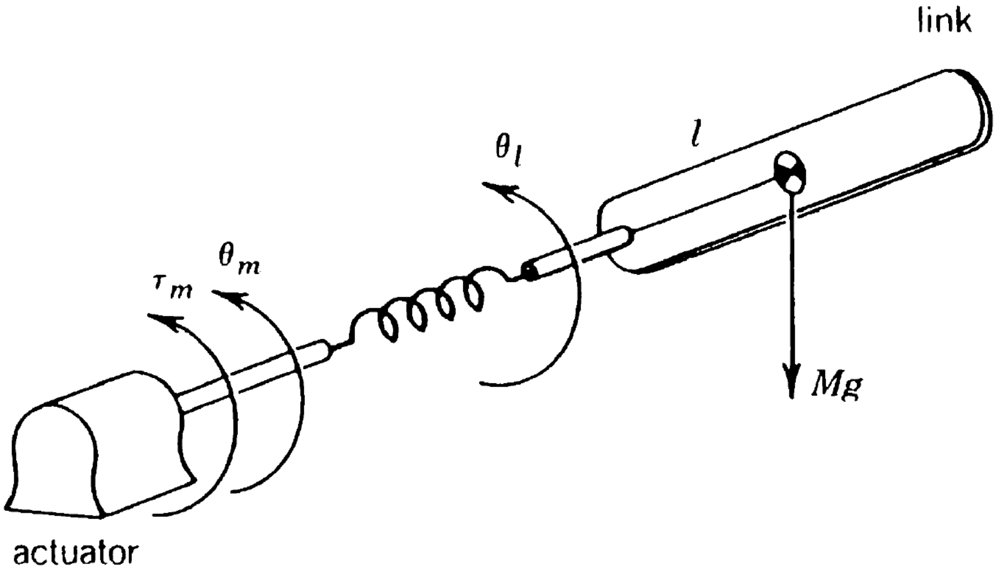
همان گونه که دیده می شود، $f_1(\cdot)$ کراندار است چرا که هر دوی $\tilde{x}$ و $\sigma(\hat{V}\hat{\tilde{x}})$ کراندار هستند، $C$ هم کراندار است و $A_c$ هم یک ماتریس هرویتز است. اگر فرض کنیم که وزن های ایده آل ثابت هستند، معادله ی دینامیک $W$ مثل یک سیستم خطی با ورودی کراندار $\left(f_1(\tilde{x},\hat{V})+\alpha_1 W\right)$ می شود. معلوم است که این سیستم پایدار است چرا که $\alpha_1$ مثبت است و ورودی سیستم هم کراندار است. پس کراندار بودن $\tilde{W}$ اثبات شد. همین طور، می توان دید که $f_2(\cdot)$ هم کراندار است چرا که همه ی آرگومان های آن کراندار هستند از جمله $\Lambda(\cdot)$. بر اساس این، رابطه ی دینامیک $V$ همانند قبلی مثل یک سیستم خطی پایدار با ورودی کراندار عمل می کند و لذا $\tilde{V}\in L_\infty$. دقت کنید که $\tilde{V}$ در توابع $f_1(\cdot)$ و $f_2(\cdot)$ فقط در توابع کراندار $\sigma(\cdot)$ و $\Lambda(\cdot)$ وجود دارد که این در تحلیل بالا اثر داشته است.

در انتها، می توان گفت که پارامتر های زیادی در آموزش و همگرایی آن تاثیر دارند. مثلاً نرخ های یادگیری را باید بزرگ تر بگیریم تا همگرایی زودتر انجام شود ولی آنقدر هم بزرگ نباشد که باعث فراجهش و نوسان بزرگ شود که این مشکل ایجاد می کند. ضرایب تعدیل اگر افزایش یابند در پایداری سیستم بهبود ایجاد می شود ولی اگر بیش از اندازه زیاد شوند می توانند به همگرایی زودرس وزن ها ختم شوند که این وزن ها با وزن های ایده آل فاصله دارند. در انتخاب ماتریس هرویتز $A$ هم چند نکته مطرح است. اولاً باید طوری انتخاب شود که جفت $(C,A)$ رویت پذیر باشد. دوماً، اگر ماتریس $A$ و در نتیجه $A_c$ خیلی پایدار باشد باعث می شود که همگرایی وزن ها کند شود چرا که $A_c^{-1}$ در قانون آپدیت وزن ها استفاده می شود. یک راه حل این است که از یک $A$ پایدارتر برای دقت بیشتر و از ضرایب نرخ یادگیری بزرگتر برای همگرایی بیشتر استفاده کنیم.

# فصل ۴

# مدل بازوی ربات انعطاف پذیر

در این فصل مدلی برای بازوی ربات انعطاف پذیر تک پیونده ارائه می شود تا در شبیه سازی فصل بعد از آن استفاده کنیم. لازم به ذکر است که به علت نیافتن مدل درست و دقیق ربات انعطاف پذیر دو پیونده، در اینجا هم نیامده است و در فصل بعد شبیه سازی های آن هم انجام نشده است. شکل کلی یکی بازوی ربات انعطاف پذیر تک پیونده در شکل ۴ـ۱ دینامیک کلی یک ربات انعطاف پذیر به صورت رابطه ی زیر می باشد.




شکل ۴ـ۱: شماتیک یک بازوی ربات انعطاف پذیر تک پیونده مدل شده با فنر گردشی

$$\begin{aligned} D_l\left(q_۱\right)\ddot{q}_۱+C_۱\left(q_۱,\dot{q}_۱\right)+g\left(q_۱\right)+B_۱\dot{q}_۱&=\tau_s \\ J\ddot{q}_۲+\tau_s+B_۲\dot{q}_۲&=\tau \end{aligned} \tag{۱\_۴}$$

که در آن، $q_۱\in\Re^n$ بردار موقعیت زاویه ای پیوندهاست و $q_۲\in\Re^n$ بردار موقعیت زاویه ای موتور ها است. $g\left(q_۱\right)\in\Re^n$ جمله ی نیروی گرانش بار می باشد، $C_۱\left(q_۱,\dot{q}_۱\right)\in\Re^n$ جمله ی دارای نیروهای سنتریفوگال و کوریولیس است، $B_۱\in\Re^{n\times n}$ و $B_۲\in\Re^{n\times n}$ ماتریس های تعدیل ویسکوزیته در خروجی و ورودی میله ی موتور هستند، $D_l\left(q_۱\right)\in\Re^{n\times n}$ و $J\in\Re^{n\times n}$ ماتریس های اینرسی ربات و علمگر هستند و $\mathcal{T}$ هم گشتاور ورودی است. گشتاور انعکاسی $\mathcal{T}_s$ هم به صورت زیر مدل می شود.

$$\tau_s=K\left(q_۲-q_۱\right)+\beta\left(q_۱,\dot{q}_۱,q_۲,\dot{q}_۲\right) \tag{۲\_۴}$$

که در آن، $K\in\Re^{n\times n}$ ماتریس مثبت معین سفت بودن فنر گردشی است که انعطاف پذیری را بین ورودی و خروجی نشان می دهد. عبارت $\beta\left(q_۱,\dot{q}_۱,q_۲,\dot{q}_۲\right)$ هم یک عبارت غیرخطی پیچیده است که معمولاً نامعلوم گرفته می شود و یا ساده شده ی آن مدل می شود.

## ۴_۱ مدل خطی بازوی ربات انعطاف پذیر تک پیونده

مدل خطی یک بازوی ربات انعطاف پذیر تک پیونده که در این جا استفاده شده است به صورت روابط زیر می باشد.

$$\begin{aligned} \dot{x}_۱&=x_۲ \\ \dot{x}_۲&=-\frac{۱}{J_l}.(B_l.x_۲+k.(x_۱-x_۳)) \\ \dot{x}_۳&=x_۴ \\ \dot{x}_۴&=-\frac{۱}{J_m}.(B_m.x_۴-k.(x_۱-x_۳)-u) \end{aligned} \tag{۳\_۴}$$

که در آن،

$$k=۰/۸ \quad , \quad J_m=۰/۰۰۰۴ \quad , \quad J_l=۰/۰۰۰۴ \quad , \quad B_m=۰/۰۱۵ \quad , \quad B_l=۰/۰۰۱ \tag{۴\_۴}$$

## ۴ـ۲ مدل غیر خطی بازوی ربات انعطاف پذیر تک پیونده

مدل غیر خطی یک بازوی ربات انعطاف پذیر تک پیونده که در این جا استفاده شده است به صورت روابط زیر می باشد.

$$\begin{aligned}
\dot{x}_۱ &= x_۲ \\
\dot{x}_۲ &= -\frac{k_s}{J_h}x_۱ + \frac{k_m^۲ k_g^۲}{R_m J_h}x_۴ - \frac{k_m k_g}{R_m J_h}u - \frac{k_s}{J_l}x_۱ + \frac{mgh}{J_l}\sin\left(x_۱ + x_۳\right) \\
\dot{x}_۳ &= x_۴ \\
\dot{x}_۴ &= \frac{k_s}{J_h}x_۱ - \frac{k_m^۲ k_g^۲}{R_m J_h}x_۴ + \frac{k_m k_g}{R_m J_h}u
\end{aligned} \tag{۴ـ۵}$$

که در آن،

(۴ـ۶)

$$K_s = ۱/۶۱ \quad , \quad J_h = ۰/۰۰۲۱ \quad , \quad m = ۰/۴۰۳ \quad , \quad g = -۹/۸۱ \quad , \quad h = ۰/۰۶ \quad ,$$
$$K_m = ۰/۰۰۷۶۷ \quad , \quad K_g = ۷۰ \quad , \quad J_l = ۰/۰۰۵۹ \quad , \quad R_m = ۲/۶$$

# فصل ۵

# شبیه سازی

در این فصل شبیه سازی طراحی های پیشنهادی در فصل های قبل، در فضای MATLAB و SIMULINK انجام شده است و نحوه ی انجام آنها آمده است. در انتها نیز نتایج این شبیه سازی آمده است.

## ۵ـ۱ آماده کردن پارامتر های شبیه سازی در فایل MATLAB برای استفاده در SIMULINK

در این قسمت پارامتر های مورد نیاز سیستم های طراحی شده در فصل های قبل که در فایل MATLAB راحتتر محاسبه می شوند، بدست می آیند. این قسمت فقط شامل پارامتر های اولیه ی سیستم است و در قسمت های بعدی تابع های MATLAB بیشتری خواهد بود که بقیه پارامتر ها در آنها تعریف شده و یا مستقیماً در سیمولینک به عنوان مقدار ثابت تعریف شده اند.

کد فایل MATLAB به نام parameters.m برای مقدار دهی پارامتر های اصلی مسئله است که به صورت زیر می باشد..

```
clear all;
clc;

eta1 = 1000;
```

```
    eta2 = 1000;
    rho1 = 1.5;
    rho2 = 1.5;

    A = [[-20 1;0 -20] zeros(2);zeros(2) [-20 1;0 -20]];
    C = eye(4);
    poles_desired = -6*[1 1 1 1];
    G = place(A',C',poles_desired)';
    Ac = A - G*C;

    size_hiddenlayerneurons = 5;
    size_systeminput = 1;
    size_states = size(A,1);

    W0 = 0.1*ones(size_states,size_hiddenlayerneurons);
    V0 = 0.1*ones(size_hiddenlayerneurons,size_states +
        size_systeminput);
```

در این کد ابتدا ضرایب نرخ یادگیری و ضرایب تعدیل یادگیری بر اساس ملاحظات قبلی و تجربی انتخاب شده اند. سپس ماتریس $A$ بر اساس مقاله انتخاب شده است و با یک جایابی قطب تقریباً مناسب، ماتریس های $G$ و در نتیجه $A_c$ بدست می آیند. ماتریس $C$ هم طوری انتخاب شده که به طور فرض تمام حالت ها در خروجی باشند. سپس اندازه ی ورودی سیستم که یک است را تعریف کردیم و اندازه ی حالت ها را بدست آوردیم. اندازه ی نورون های لایه ی مخفی ۵ گرفته شده که مساوی جمع تعداد ورودی سیستم و تعداد حالت ها می باشد یا به عبارت دیگر با تعداد ورودی شبکه یکی است. دلیل اینکه این مقدار بیشتر لحاظ نشده این است که هرچند افزایش تعداد نورون های لایه ی مخفی قدرت تقریب زدن غیر خطی شبکه را بیشتر می کند ولی میزان نوسانات هم بیشتر می شود که نقطه ضعف آن است پس این عدد ۵ مقدار مناسبی برای لایه ی مخفی این شبکه می باشد.

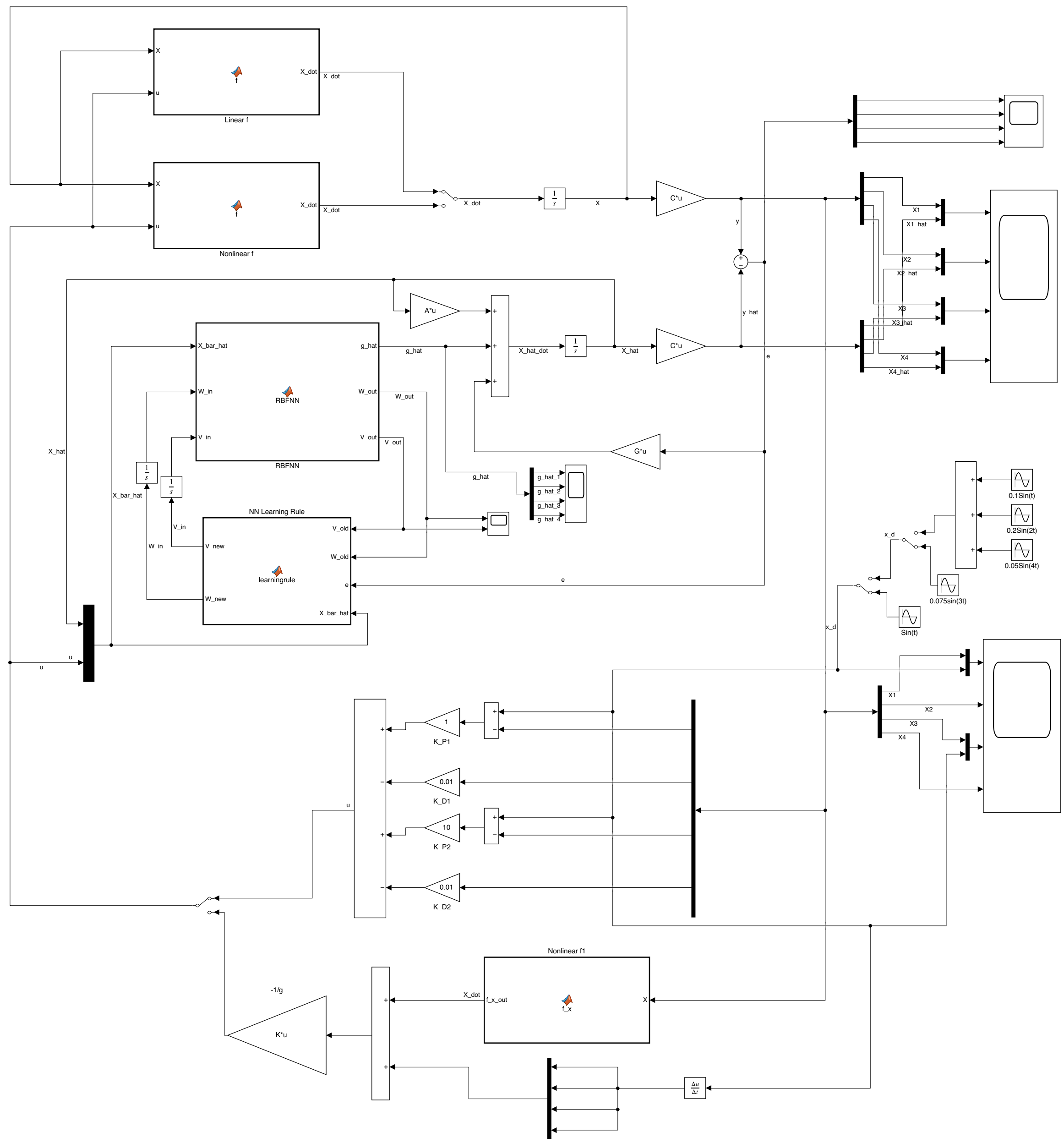
در آخر هم به وزن ها یک مقدار اولیه ی غیر صفر ثابت داده شده است. باید دقت شود که این وزن ها به علت وجود در لایه های متفاوت دارای ابعاد متفاوت هستند و ابعاد آنها باید درست بدست آورده شود تا در ادامه شاهد خطا در شبیه سازی نباشیم. همین طور این مقادیر اولیه نباید شامل صفر باشد چرا که اگر مقدار ها صفر باشد، روی آن باقی می ماند و شبکه اصلاً آموزش نمی بیند. اینکه این مقدار

ثابت تقریباً در چه حدودی باشد را باید بر اساس شبیه سازی بدست آورد که همان کاری است که در اینجا انجام شده و هدف های کاهش نوسان آموزش و همگرایی و ... در نظر گرفته شده است.

## ۵ـ۲ نحوه ی شبیه سازی طراحی سیستم مورد نظر در SIMULINK

در این قسمت، شماتیک های شبیه سازی طراحی فصل های قبل در SIMULINK نشان داده خواهد شد. شماتیک همه ی سیستم کنترل حلقه بسته که شامل رویتگر و سیستم دینامیک واقعی بازوی ربات انعطاف پذیر و کنترل کننده ی سیستم حلقه بسته می باشد، در شکل ۵ـ۱ آمده است. در ضمن همان طور که قبلاً هم گفته شده، حالت ها به ترتیب شامل موقعیت پیوند، سرعت پیوند، موقعیت موتور و سرعت موتور می باشند. همان گونه که در شکل دیده می شود، این سیستم دارای ۵ تابع MATLAB است که در ادامه محتوای ۴ تای آنها نمایش داده می شود. همین طور، در این فایل که به نام FlexibleJoint-NNO.slx است، از سوئیچ های دستی برای تغییر مسیر ها در سیستم کلی استفاده شده است. مثلاً، اولین سوئیچ در بالای شکل در انتخاب بین مدل خطی و غیر خطی بازوی ربات انعطاف پذیر تک پیونده می باشد. مدل دو پیونده نیز به خاطر توضیحات در فصل قبل اصلاْ شبیه سازی نشده است. انتخاب دوم بین روش کنترل حلقه بسته در پایین شکل است که یا کنترل کننده ی PD (بالایی) است و یا کنترل کننده ی خطی سازی فیدبک برای سیستم غیرخطی (پایینی) است. هر چند کنترل کننده ی خطی سازی فیدبک به علت اینکه پیشنهاد مقاله نبوده و اصلاً در اینجا درست کنترل نمی کند و نتایج اشتباه داده است، در شبیه سازی ها استفاده نشده است. در این جا فقط کنترل کننده ی PD که پیشنهاد خود مقاله هم بود استفاده شده است که پارامتر های آن دقیقاً پارامتر های درون شکل و در طول همه ی شبیه سازی ها ثابت است. این پارامتر ها با پیدا کردن حالت بهینه ی تقریبی در شبیه سازی ها تنظیم شده اند تا هم خطا کم شود و همگرایی بیشتر شود و در عین حال نواسات هم زیاد نشود. انتخاب آخر هم بین سیگنال ۳ سیگنال مرجع در وسط سمت راست شکل است که به ازای آنها شبیه سازی ها انجام شده است.

اگر دقت کرده باشید شبکه ی عصبی در شکل از دو تابع MATLAB تشکیل شده است که تابع RBFNN خروجی شبکه به ازای وزن های آن مرحله را می دهد و تابع learningrule دارای قانون آپدیت وزن ها می باشد و آموزش شبکه را انجام می دهد. برای سهولت در کار و گرفتن نتایج دقیق تر و دوری از خطاهای حین کار، به جای تبدیل کل سیستم به حالت گسسته، قانون آپدیت وزن ها را به حالت پیوسته تبدیل کردیم چرا که در معادلات هم خروجی قانون آپدیت وزن ها، مشتق وزن ها می باشد و در




شکل ۵ـ۱: شماتیک همه ی سیستم کنترل حلقه بسته در SIMULINK

نتیجه با یک انتگرال گیر ساده و در برگیرنده مقادیر اولیه ی وزن ها آن را به خود وزن ها تبدیل می کنیم و به شبکه RBFNN وارد می کنیم.

در شبیه سازی ها از اسکوپ هایی که موارد خطای تخمین، وزن های شبکه، ردیابی حالت ها و ردیابی سیستم واقعی را نشان می دهند استفاده شده است.

کد MATLAB تابع خطی f به صورت زیر می باشد.

```
function X_dot = f(X, u)

X1 = X(1,1);
X2 = X(2,1);
X3 = X(3,1);
X4 = X(4,1);

k = 0.8;
J_m = 0.0004;
J_l = 0.0004;
B_m = 0.015;
B_l = 0.001;

X1_dot = X2;
X2_dot = (-1/J_l)*(B_l*X2 + k*(X1 - X3));
X3_dot = X4;
X4_dot = (-1/J_m)*(B_m*X4 - k*(X1 - X3) -u);

X_dot = [X1_dot;X2_dot;X3_dot;X4_dot];
end
```

کد MATLAB تابع غیرخطی f به صورت زیر می باشد.

```
function X_dot = f(X, u)

X1 = X(1,1);
X2 = X(2,1);
X3 = X(3,1);
X4 = X(4,1);

K_s = 1.61;
J_h = 0.0021;
```

```
    m = 0.403;
    g = -9.81;
    h = 0.06;
    K_m = 0.00767;
    K_g = 70;
    J_l = 0.0059;
    R_m = 2.6;

    X1_dot = X2;
    X2_dot = - (K_s/J_h)*X1 + (((K_m^2)*(K_g^2))/(R_m*J_h))*X4 -
        (((K_m)*(K_g))/(R_m*J_h))*u - (K_s/J_l)*X1 + ((m*g*h)/J_l)*
        sin(X1 + X3);
    X3_dot = X4;
    X4_dot = (K_s/J_h)*X1 - (((K_m^2)*(K_g^2))/(R_m*J_h))*X4 + (((
        K_m)*(K_g))/(R_m*J_h))*u;

    X_dot = [X1_dot;X2_dot;X3_dot;X4_dot];

end
```

کد MATLAB تابع RBFNN به صورت زیر می باشد.

```
function [g_hat,W_out,V_out] = RBFNN(X_bar_hat, W_in,V_in,
    size_hiddenlayerneurons, size_systeminput)

m = size_hiddenlayerneurons;                % number of hidden-
    layer neurons
m_xu = size(X_bar_hat,1);         % input size
m_u = size_systeminput;           % system input size
n = m_xu - m_u;                 % output size

% creating the output of the RBF network :

output = zeros(n,1);
```

```
for outputindex = 1 : n

for neuronindex = 1 : m

output(outputindex,1) = output(outputindex,1) + W_in(outputindex,neuronindex) * RBF(X_bar_hat,V_in(neuronindex,:));

end

end

g_hat = output;
W_out = W_in;
V_out = V_in;
end

function sigma_i = RBF(X_bar_hat,V_i)

sigma_i = 2/(1 + exp(-2*V_i*X_bar_hat)) -1;

end
```

کد MATLAB تابع learningrule به صورت زیر می باشد.

```
function [V_new,W_new] = learningrule(V_old, W_old, e, X_bar_hat, Ac, C, size_hiddenlayerneurons, eta1, eta2, rho1, rho2)

W_hat = W_old;
V_hat = V_old;

Lambda_arg = zeros(1,size_hiddenlayerneurons);
```

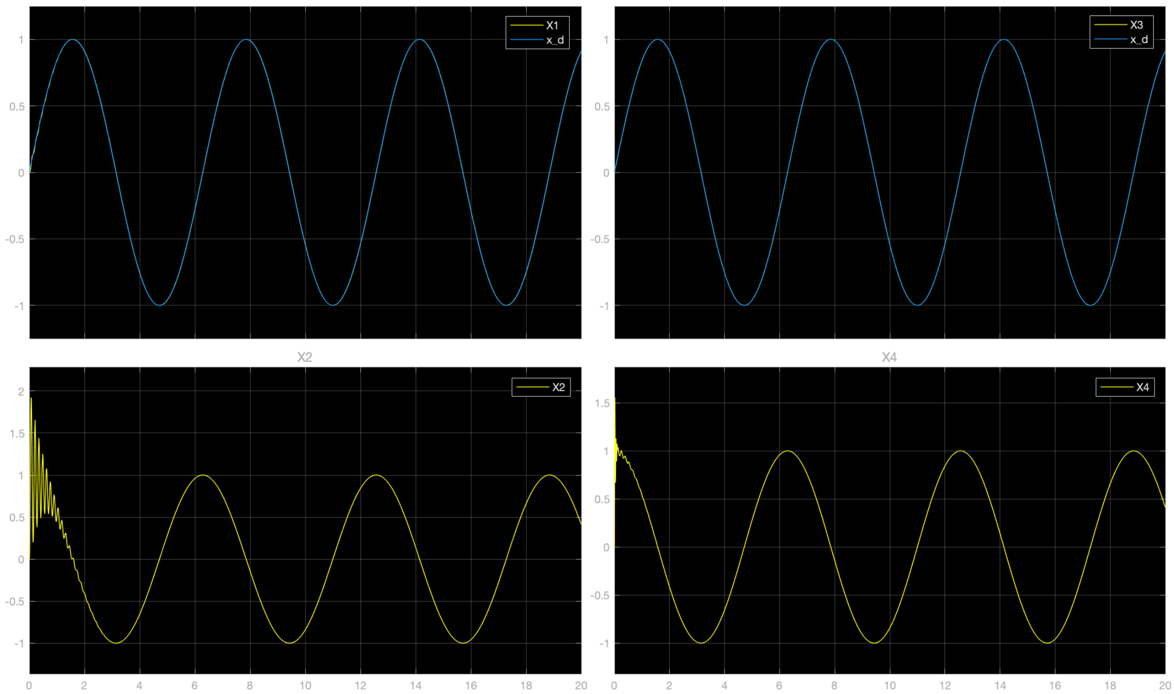
```
Sigma_hl = zeros(size_hiddenlayerneurons,1);
for m = 1 : size_hiddenlayerneurons

Lambda_arg(1,m) = (RBF(X_bar_hat,V_hat(m,:)))^2;
Sigma_hl(m,1) = RBF(X_bar_hat,V_hat(m,:));

end
Lambda = diag(Lambda_arg);

W_hat_dot = - eta1 * ((e' * C * inv(Ac))') * (Sigma_hl') - rho1*norm(e,2)*W_hat;
V_hat_dot = - eta2 * ((e' * C * inv(Ac) * W_hat * (eye(size_hiddenlayerneurons)-Lambda))') * (sign(X_bar_hat)') - rho2*norm(e,2)*V_hat;

W_hat = W_hat_dot;
V_hat = V_hat_dot;

W_new = W_hat;
V_new = V_hat;

end

function sigma_i = RBF(X_bar_hat,V_i)

sigma_i = 2/(1 + exp(-2*V_i*X_bar_hat)) -1;

end
```

کد MATLAB تابع f-x چون از استفاده نشده است در اینجا هم نیاورده شده است.

## ۵ـ۳ نتایج شبیه سازی

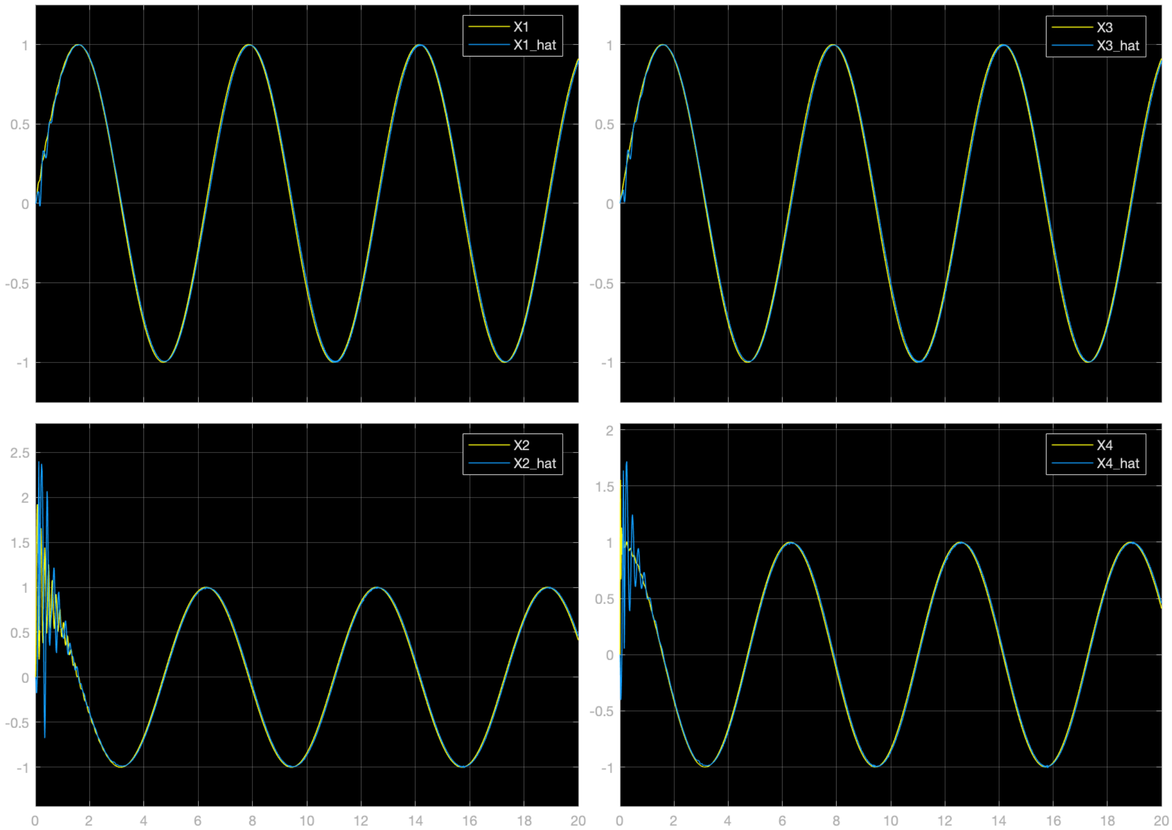
در این قسمت نتایج شبیه سازی مدل سیستم قسمت های قبل با پارامترهای آن آورده شده است. ابتدا، شبیه سازی برای مدل خطی سیستم و سپس برای مدل غیرخطی سیستم انجام می شود. حالت های مرجع ردیابی هم به ترتیب از پایین به بالای در شکل ۵ـ۱ در شبیه سازی ها اعمال می شوند. برای سهولت کار به هر کدام از این سیگنال های مرجع یک شماره داده ایم که به ترتیب سیگنال مرجع ۱ شامل $\sin(t)$ است، سیگنال مرجع ۲ شامل $۰/۰۷۵\sin(۳t)$ و سیگنال مرجع ۳ شامل $۰/۱\sin(t) + ۰/۲\sin(۲t) + ۰/۰۵sin(۴t)$ می باشد. روش کنترلی هم فقط با کنترل کننده ی PD است.

در شکل ۵ـ۲ خروجی ردیابی برای حالت های سیستم اصلی خطی و سیگنال مرجع ۱ آمده است. همان گونه که در این شکل دیده می شود ردیابی سیستم اصلی خوب کار کرده است و در نتیجه کنترل کننده ی PD درست کار کرده است.




شکل ۵ـ۲: ردیابی حالت ها برای سیستم اصلی خطی و سیگنال مرجع ۱

در شکل ۵ـ۳ ردیابی حالت های رویتگر بر سیستم اصلی برای سیستم اصلی خطی و سیگنال مرجع ۱ آمده است. در این شکل دیده می شود که رویتگر درست عمل کرده است هرچند نوساناتی هم وجود دارد که مربوط به حالت گذرای سیستم اصلی و قانون آموزش شبکه می باشند.



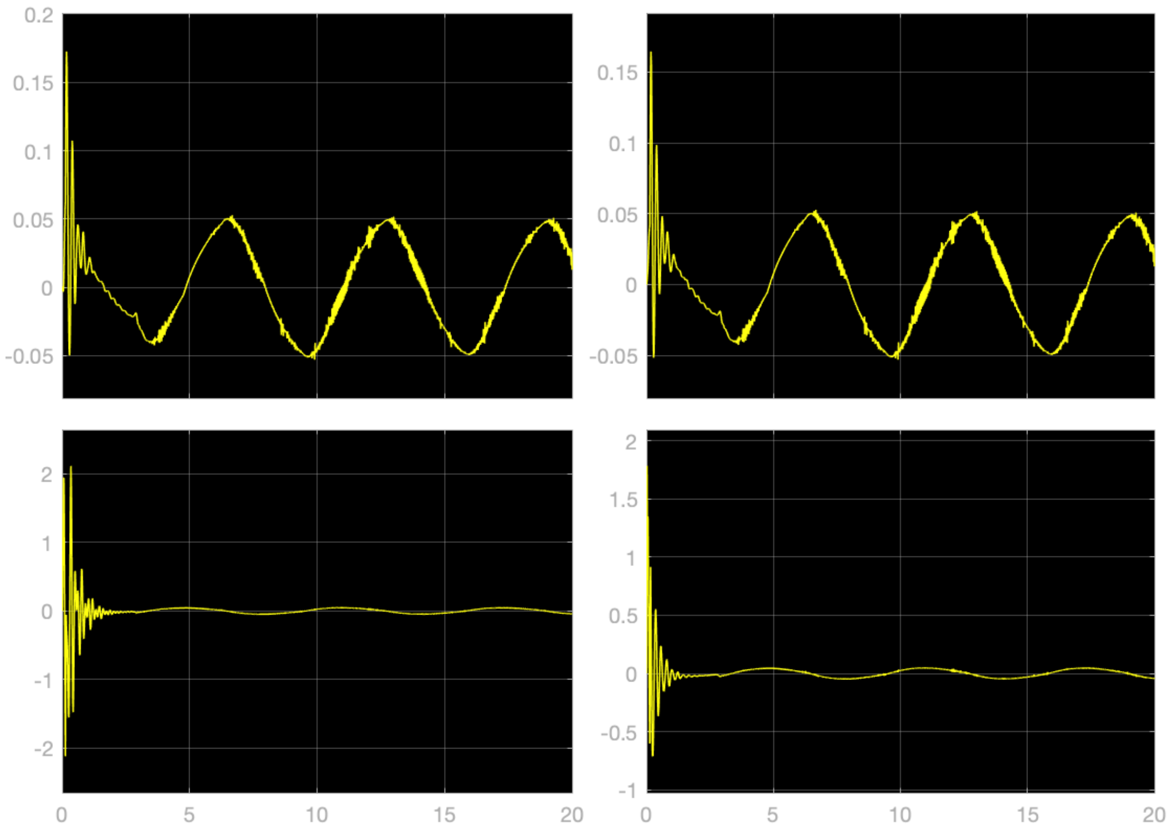

شکل ۵ـ۳: ردیابی حالت های رویتگر بر سیستم اصلی برای سیستم اصلی خطی و سیگنال مرجع ۱

در شکل ۵ـ۴ خطای ردیابی حالت های رویتگر بر سیستم اصلی برای سیستم اصلی خطی و سیگنال مرجع ۱ آمده است. همان گونه که دیده می شود خطا کراندار است.

در شکل ۵ـ۵ خروجی ردیابی برای وزن های رویتگر سیستم اصلی خطی و سیگنال مرجع ۱ آمده است.

در شکل ۵ـ۶ خروجی ردیابی برای حالت های سیستم اصلی خطی و سیگنال مرجع ۲ آمده است. همان گونه که در این شکل دیده می شود ردیابی سیستم اصلی خوب کار کرده است و در نتیجه کنترل کننده ی PD درست کار کرده است.

در شکل ۵ـ۷ ردیابی حالت های رویتگر بر سیستم اصلی برای سیستم اصلی خطی و سیگنال مرجع ۲ آمده است. در این شکل دیده می شود که رویتگر تقریباً درست عمل کرده است هرچند نوساناتی هم وجود دارد که مربوط به حالت گذرای سیستم اصلی و قانون آموزش شبکه می باشند. همان گونه که



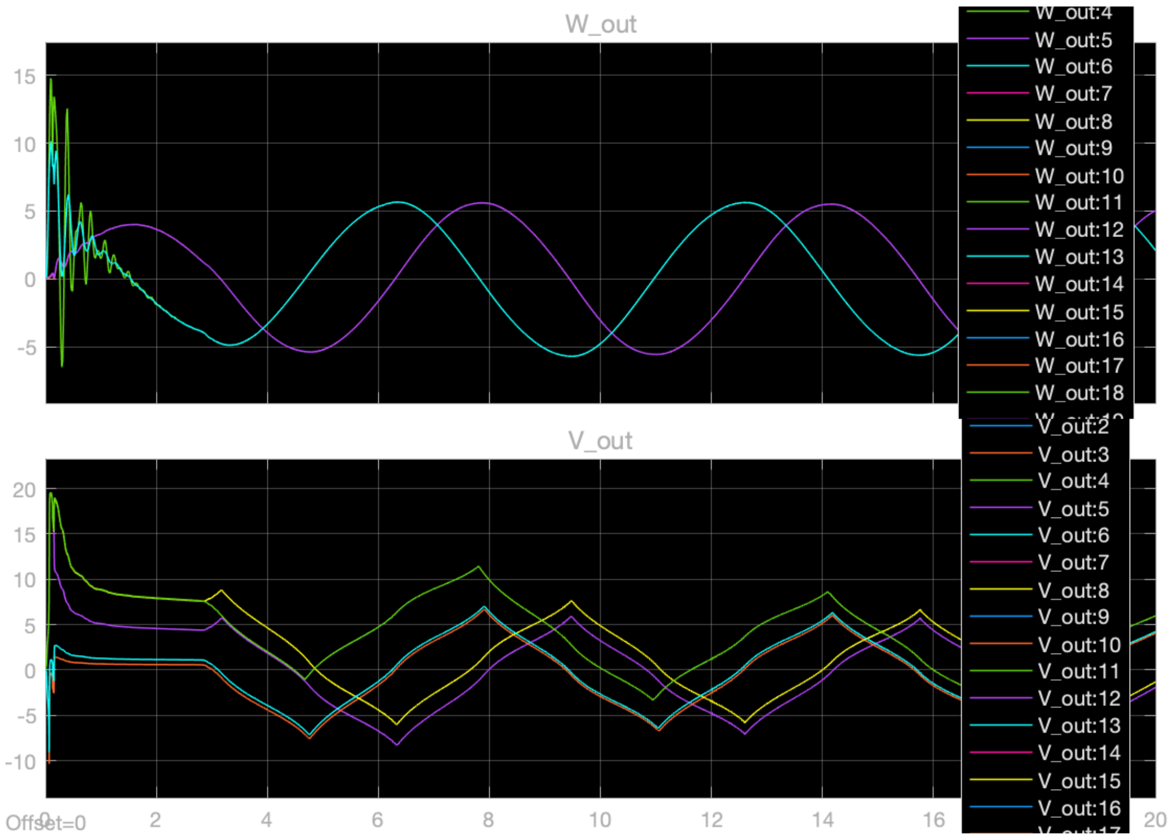

شکل ۵ـ۴: خطای ردیابی حالت های رویتگر بر سیستم اصلی برای سیستم اصلی خطی و سیگنال مرجع ۱

دیده می شود به علت کم شدن دامنه ی سیگنال نوسانات آموزش خود را بیشتر نشان می دهند. هر چند این نوسانات با تغییر پارامتر های آموزش قابل بهبود هستند ولی در این شبیه سازی برای شهود این گونه پدیده ها، پارامتر ها تغییر داده نشده اند.

در شکل ۵ـ۸ خطای ردیابی حالت های رویتگر بر سیستم اصلی برای سیستم اصلی خطی و سیگنال مرجع ۲ آمده است. همان گونه که دیده می شود خطا کراندار است.

در شکل ۵ـ۹ خروجی ردیابی برای وزن های رویتگر سیستم اصلی خطی و سیگنال مرجع ۲ آمده است.

در شکل ۵ـ۱۰ خروجی ردیابی برای حالت های سیستم اصلی خطی و سیگنال مرجع ۳ آمده است. همان گونه که در این شکل دیده می شود ردیابی سیستم اصلی خوب کار کرده است و در نتیجه کنترل



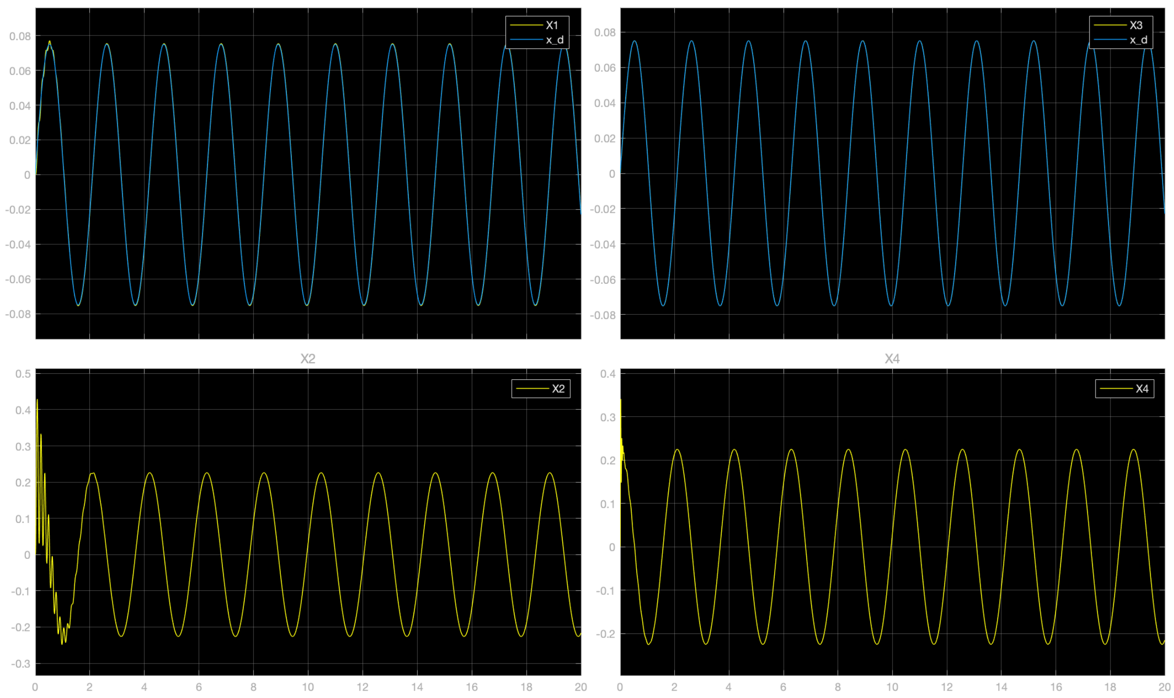

شکل ۵ــ۵: ردیابی وزن های رویتگر برای سیستم اصلی خطی و سیگنال مرجع ۱

کننده ی PD درست کار کرده است.

در شکل ۵ــ۱۱ ردیابی حالت های رویتگر بر سیستم اصلی برای سیستم اصلی خطی و سیگنال مرجع ۳ آمده است. در این شکل دیده می شود که رویتگر تقریباً درست عمل کرده است هرچند نوساناتی هم وجود دارد که مربوط به حالت گذرای سیستم اصلی و قانون آموزش شبکه می باشند. همان گونه که دیده می شود به علت کم شدن دامنه ی سیگنال نوسانات آموزش خود را بیشتر نشان می دهند. هر چند این نوسانات با تغییر پارامتر های آموزش قابل بهبود هستند ولی در این شبیه سازی برای شهود این گونه پدیده ها، پارامتر ها تغییر داده نشده اند.

در شکل ۵ــ۱۲ خطای ردیابی حالت های رویتگر بر سیستم اصلی برای سیستم اصلی خطی و سیگنال مرجع ۳ آمده است. همان گونه که دیده می شود خطا کراندار است.

در شکل ۵ــ۱۳ خروجی ردیابی برای وزن های رویتگر سیستم اصلی خطی و سیگنال مرجع ۳ آمده




شکل ۵ـ۶: ردیابی حالت ها برای سیستم اصلی خطی و سیگنال مرجع ۲

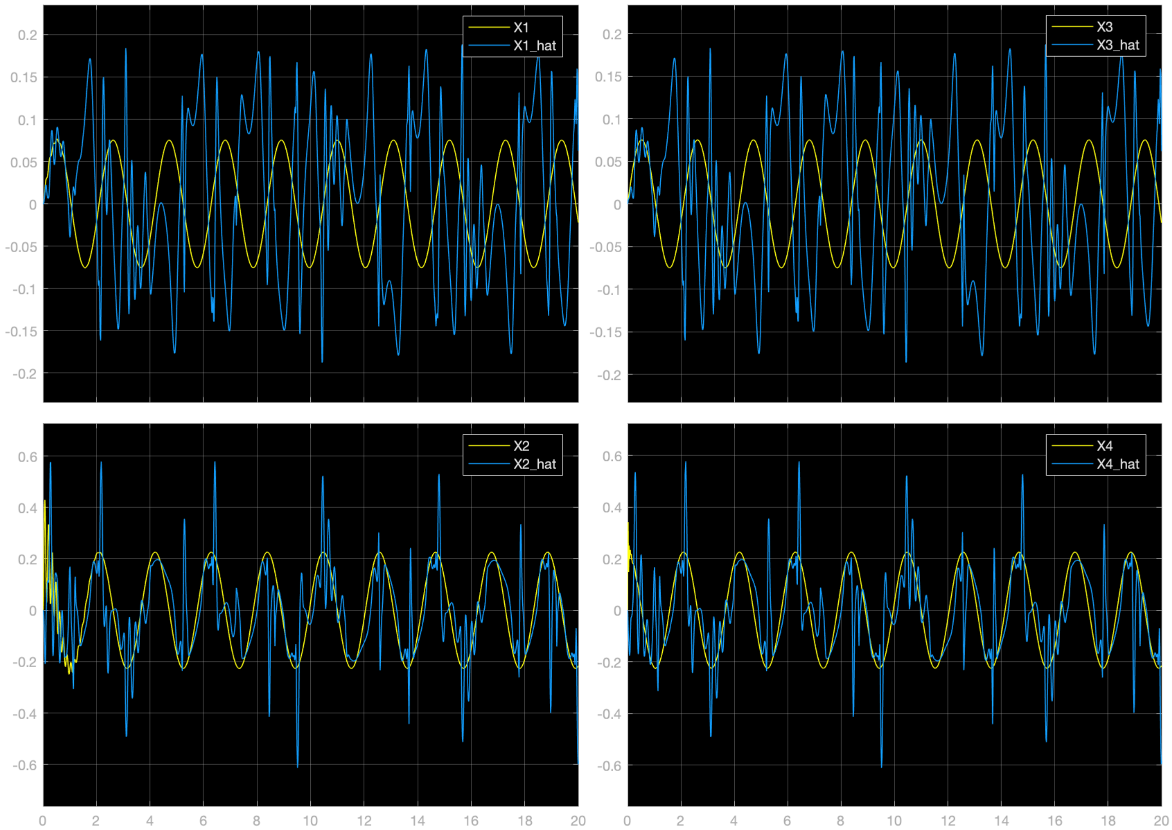
است.

حال همین شبیه سازی ها را با استفاده از سیستم اصلی غیر خطی تکرار می کنیم.

در شکل ۵ـ۱۴ خروجی ردیابی برای حالت های سیستم اصلی غیرخطی و سیگنال مرجع ۱ آمده است. همان گونه که در این شکل دیده می شود، ردیابی فقط برای موقعیت موتور درست کار کرده و برای موقعیت پیوند دارای کرانی می باشد که از آن بالاتر نمی رود و کنترل کننده ی PD درست کار نکرده است.

در شکل ۵ـ۱۵ ردیابی حالت های رویتگر بر سیستم اصلی برای سیستم اصلی غیرخطی و سیگنال مرجع ۱ آمده است. در این شکل دیده می شود که رویتگر درست عمل کرده است و تابع غیر خطی سیستم را خوب تقریب زده است. هرچند نوساناتی هم وجود دارد که مربوط به حالت گذرای سیستم اصلی و قانون آموزش شبکه می باشند.

در شکل ۵ـ۱۶ خطای ردیابی حالت های رویتگر بر سیستم اصلی برای سیستم اصلی غیرخطی و سیگنال مرجع ۱ آمده است. همان گونه که دیده می شود خطا کراندار است.

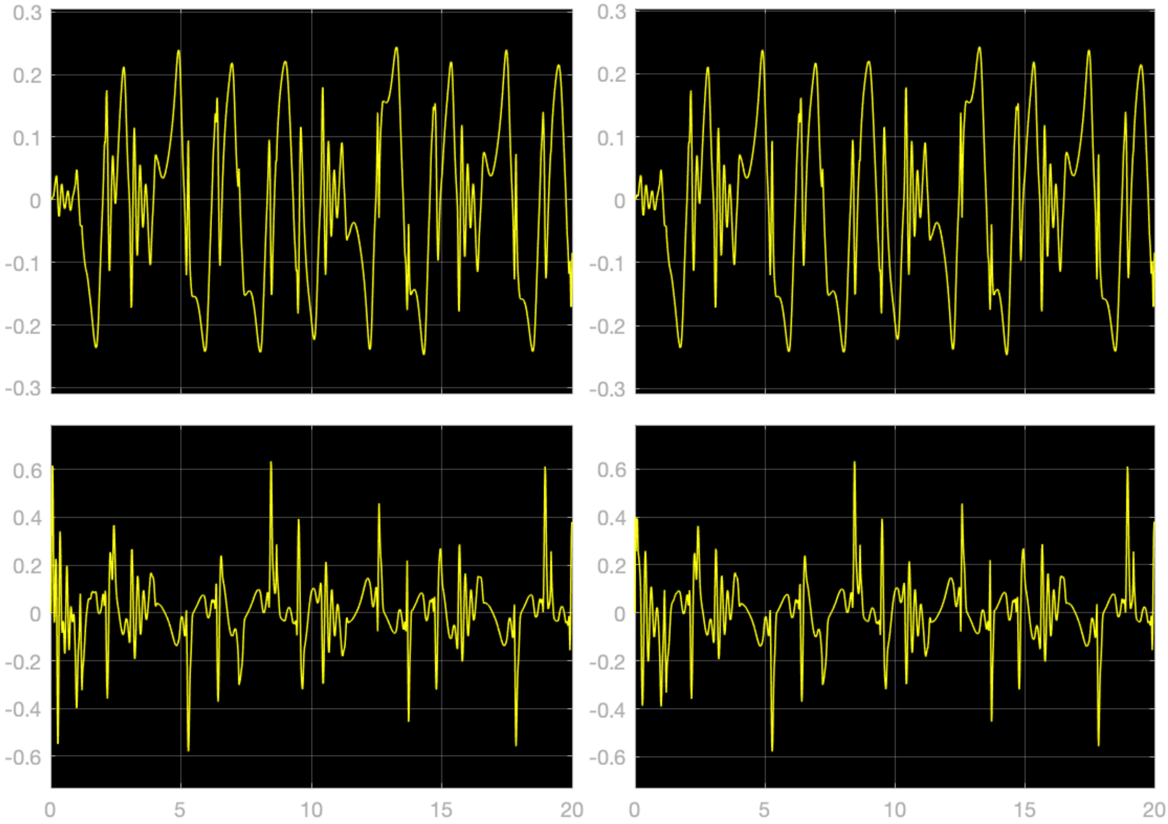



شکل ۵ـ۷: ردیابی حالت های رویتگر بر سیستم اصلی برای سیستم اصلی خطی و سیگنال مرجع ۲

در شکل ۵ـ۱۷ خروجی ردیابی برای وزن های رویتگر سیستم اصلی غیرخطی و سیگنال مرجع ۱ آمده است.

در شکل ۵ـ۱۸ خروجی ردیابی برای حالت های سیستم اصلی غیرخطی و سیگنال مرجع ۲ آمده است. همان گونه که در این شکل دیده می شود، ردیابی فقط برای موقعیت موتور درست کار کرده و برای موقعیت پیوند دارای کرانی می باشد که از آن بالاتر نمی رود و کنترل کننده ی PD درست کار نکرده است.

در شکل ۵ـ۱۹ ردیابی حالت های رویتگر بر سیستم اصلی برای سیستم اصلی غیرخطی و سیگنال مرجع ۲ آمده است. در این شکل دیده می شود که رویتگر تقریباً درست عمل کرده است هرچند نوساناتی هم وجود دارد که مربوط به حالت گذرای سیستم اصلی و قانون آموزش شبکه می باشند. همان گونه که دیده می شود به علت کم شدن دامنه ی سیگنال نوسانات آموزش خود را بیشتر نشان می دهند. هر چند



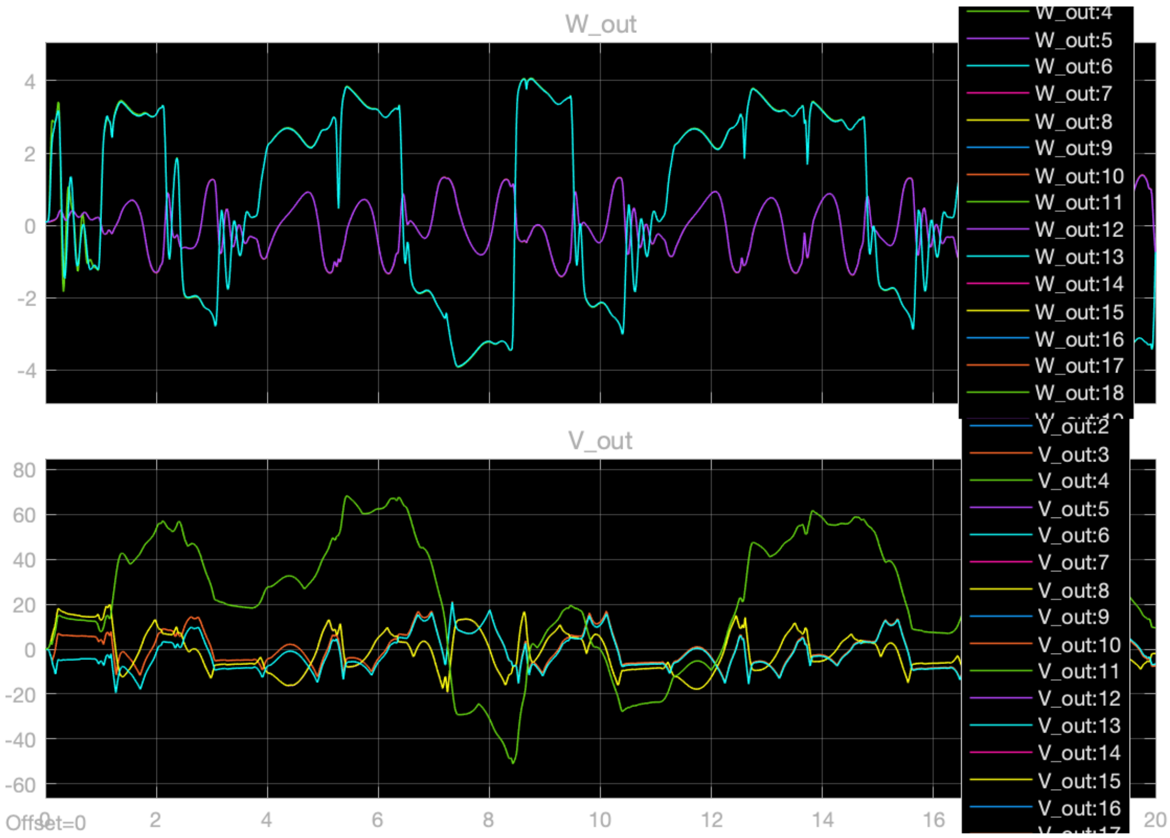

شکل ۵ـ۸: خطای ردیابی حالت های رویتگر بر سیستم اصلی برای سیستم اصلی خطی و سیگنال مرجع ۲

این نوسانات با تغییر پارامتر های آموزش قابل بهبود هستند ولی در این شبیه سازی برای شهود این گونه پدیده ها، پارامتر ها تغییر داده نشده اند.

در شکل ۵ـ۲۰ خطای ردیابی حالت های رویتگر بر سیستم اصلی برای سیستم اصلی غیرخطی و سیگنال مرجع ۲ آمده است. همان گونه که دیده می شود خطا کراندار است.

در شکل ۵ـ۲۱ خروجی ردیابی برای وزن های رویتگر سیستم اصلی غیرخطی و سیگنال مرجع ۲ آمده است.

در شکل ۵ـ۲۲ خروجی ردیابی برای حالت های سیستم اصلی غیرخطی و سیگنال مرجع ۳ آمده است. همان گونه که در این شکل دیده می شود، ردیابی فقط برای موقعیت موتور درست کار کرده و برای موقعیت پیوند دارای کرانی می باشد که از آن بالاتر نمی رود و کنترل کننده ی PD درست کار



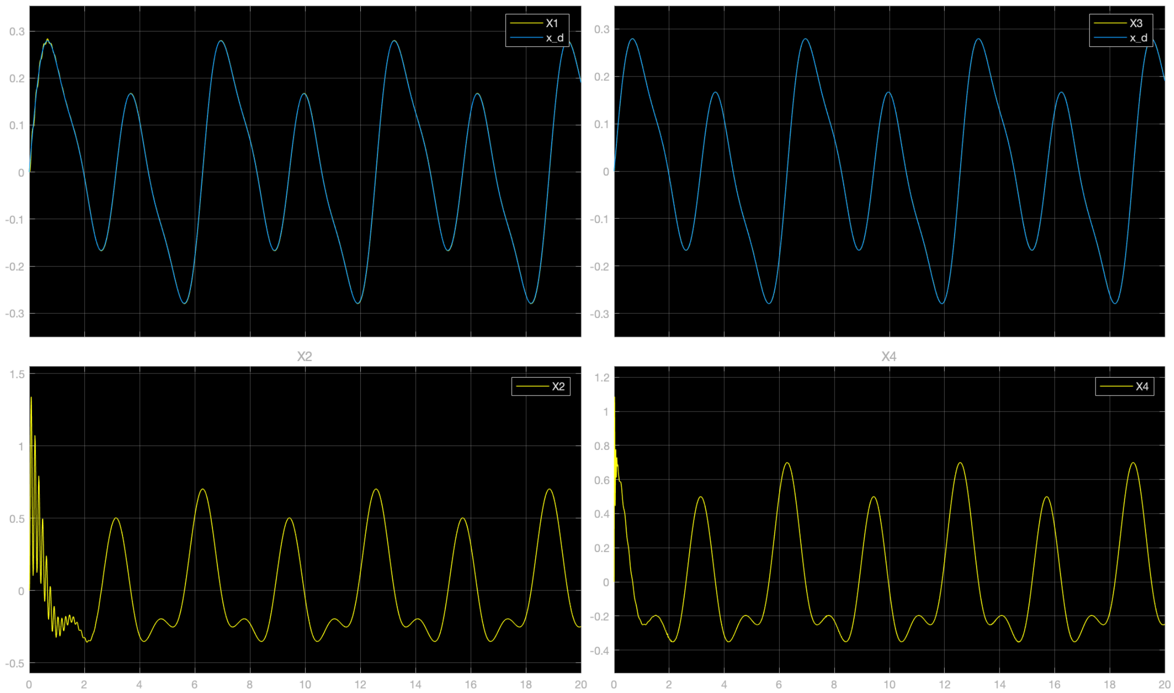

شکل ۵ـ۹: ردیابی وزن های رویتگر برای سیستم اصلی خطی و سیگنال مرجع ۲

نکرده است.

در شکل ۵ـ۲۳ ردیابی حالت های رویتگر بر سیستم اصلی برای سیستم اصلی غیرخطی و سیگنال مرجع ۳ آمده است. در این شکل دیده می شود که رویتگر تقریباً درست عمل کرده است هرچند نوساناتی هم وجود دارد که مربوط به حالت گذرای سیستم اصلی و قانون آموزش شبکه می باشند. همان گونه که دیده می شود به علت کم شدن دامنه ی سیگنال نوسانات آموزش خود را بیشتر نشان می دهند. هر چند این نوسانات با تغییر پارامتر های آموزش قابل بهبود هستند ولی در این شبیه سازی برای شهود این گونه پدیده ها، پارامتر ها تغییر داده نشده اند.

در شکل ۵ـ۲۴ خطای ردیابی حالت های رویتگر بر سیستم اصلی برای سیستم اصلی غیرخطی و سیگنال مرجع ۳ آمده است. همان گونه که دیده می شود خطا کراندار است.

در شکل ۵ـ۲۵ خروجی ردیابی برای وزن های رویتگر سیستم اصلی غیرخطی و سیگنال مرجع ۳

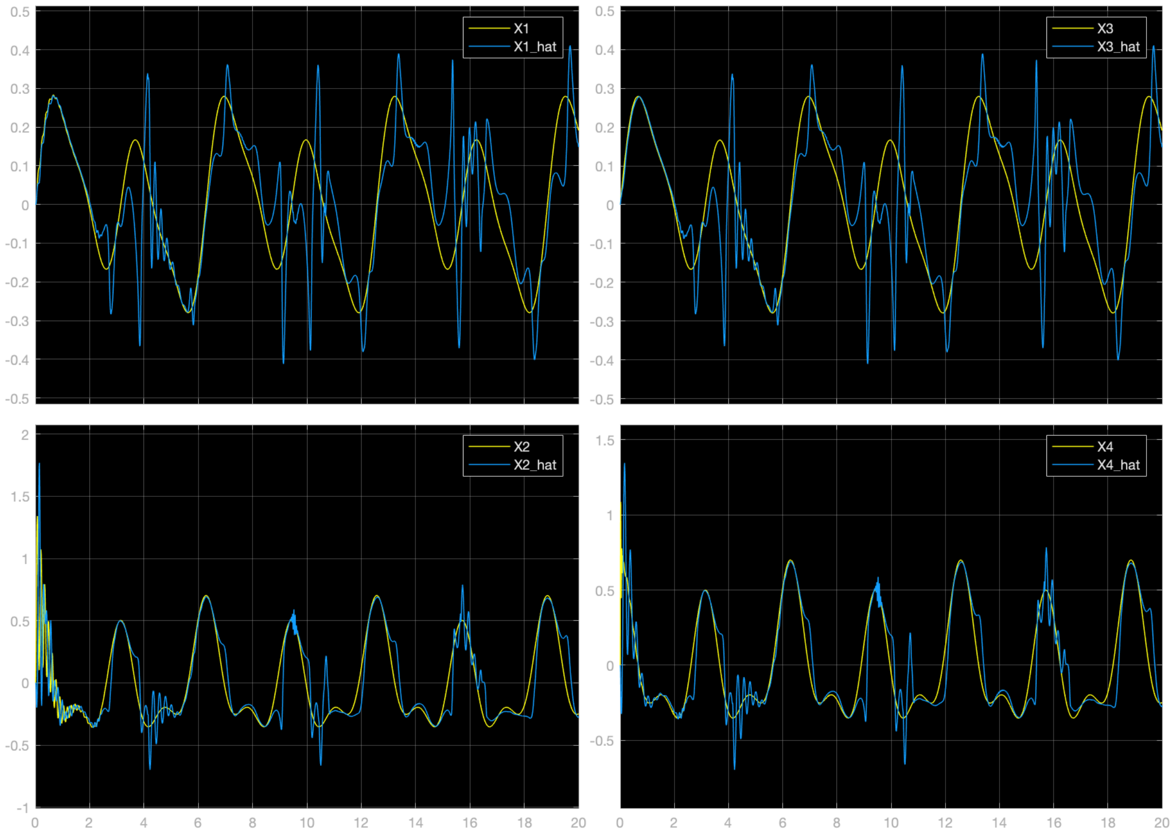



شکل ۵ـ۱۰: ردیابی حالت ها برای سیستم اصلی خطی و سیگنال مرجع۳

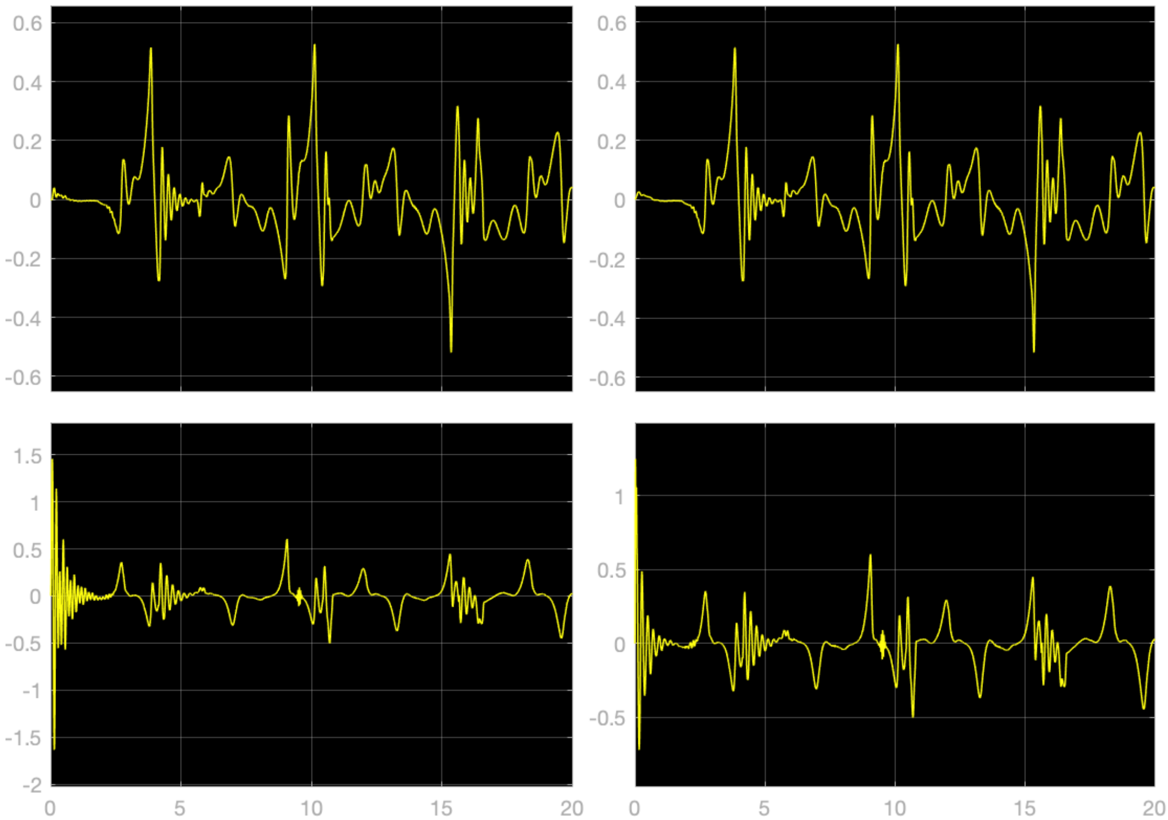
آمده است.

نتایج کلی رویتگر برای سیستم خطی و غیرخطی و برای همه ی سیگنال های مرجع موقعیت، به شرط تنظیم درست پارامتر های آموزش برای هر سناریو قابل قبول می باشند. در اینجا پارامتر های آموزش در تمام شبیه سازی ثابت بودند که یک حالت تقریباً بهینه ی کلی را شامل می شدند و اینکه نتیجه ی این پروژه از نظر بحث روی تاثیر سیگنال مرجع بر روی عملکرد رویتگر بهتر است.

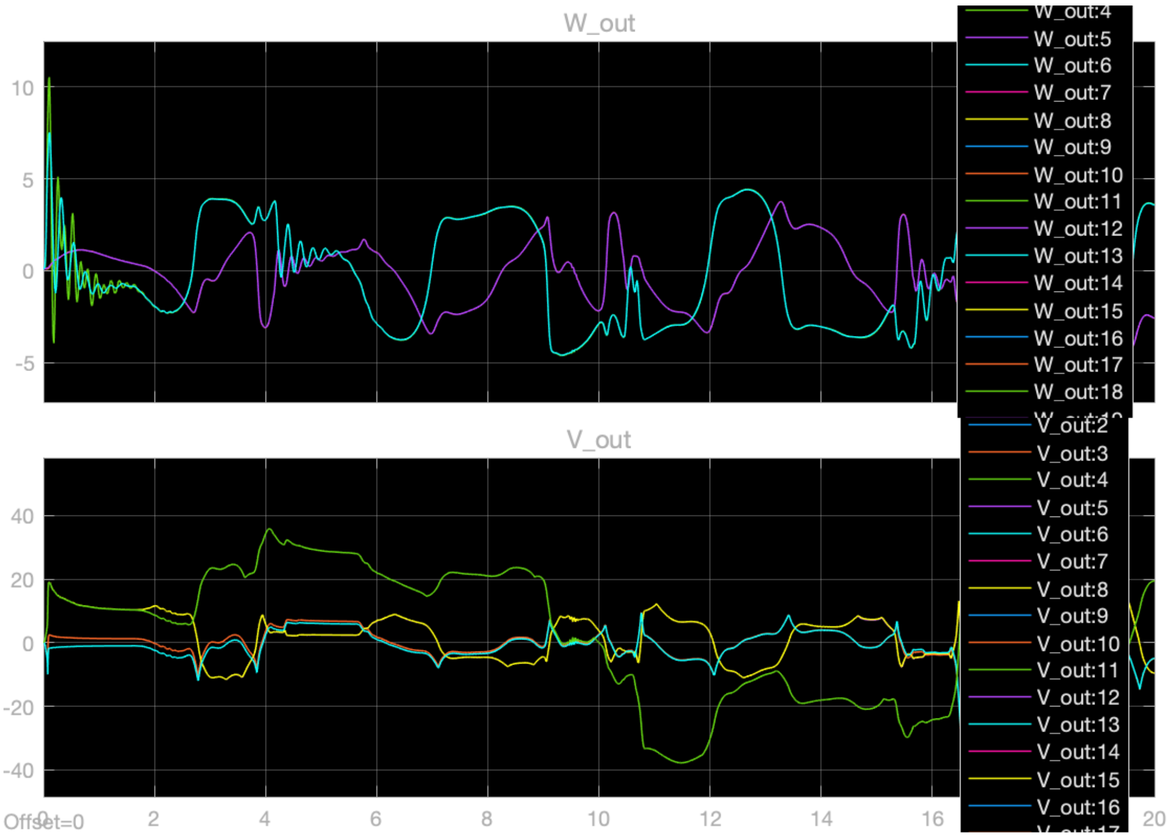
همین طور یا مدل سیستم اصلی غیرخطی دارای کران برای حالت موقعیت پیوند است که باعث می شود درست کنترل نشود و یا کنترل کننده ی PD پیشنهادی استفاده شده در این پروژه برای این کار مناسب نیست.



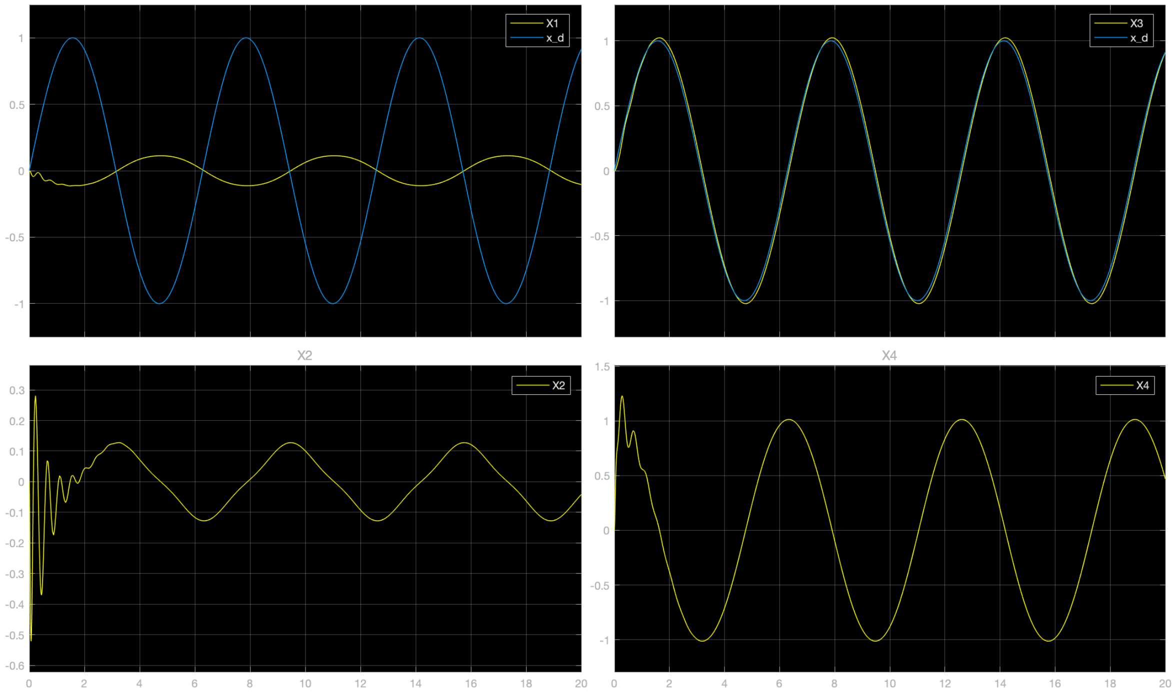

شکل ۵ـ۱۱: ردیابی حالت های رویتگر بر سیستم اصلی برای سیستم اصلی خطی و سیگنال مرجع ۳



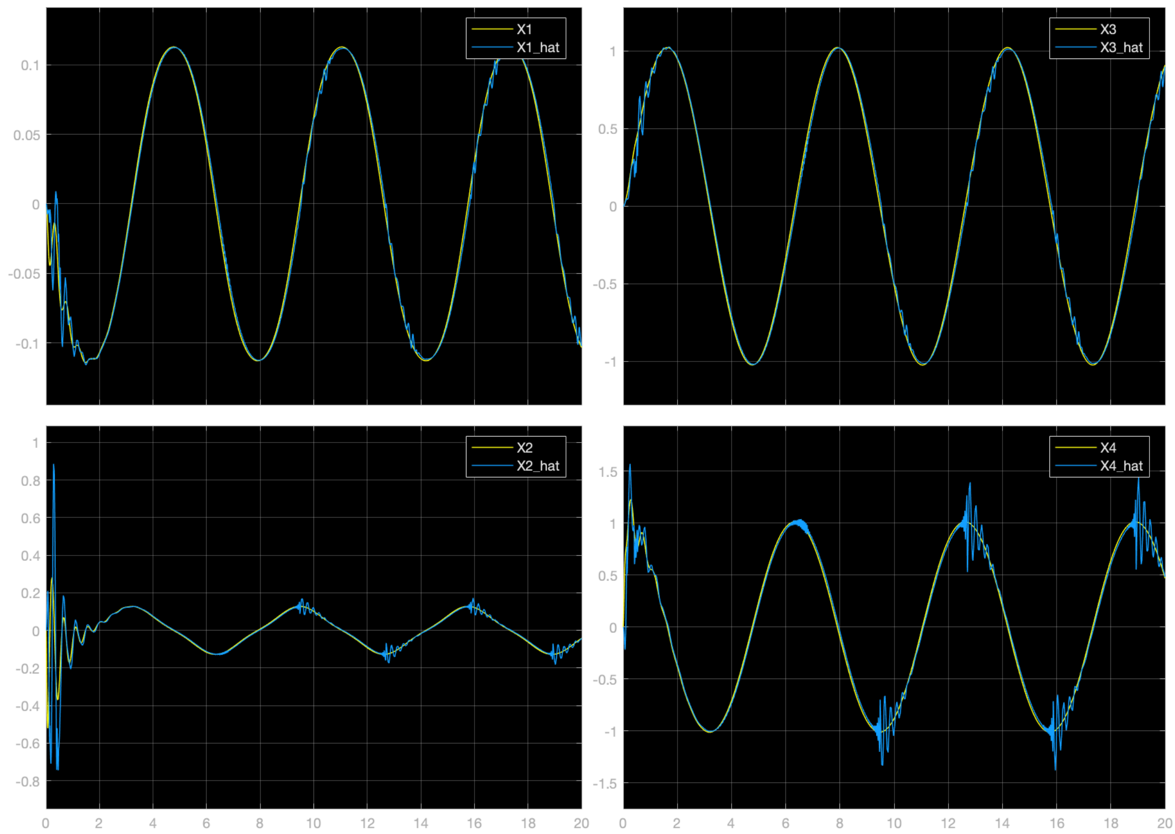

شکل ۵ـ۱۲: خطای ردیابی حالت های رویتگر بر سیستم اصلی برای سیستم اصلی خطی و سیگنال مرجع ۳



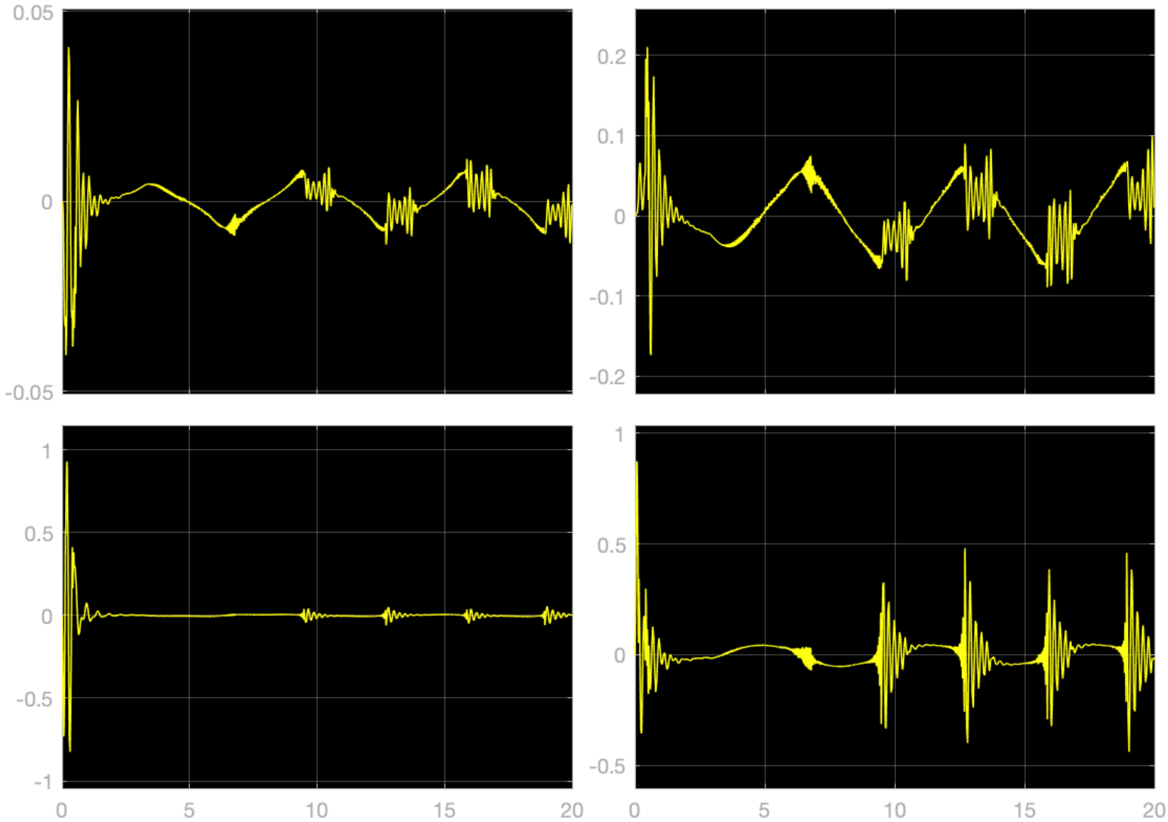

شکل ۵ـ۱۳: ردیابی وزن های رویتگر برای سیستم اصلی خطی و سیگنال مرجع ۳



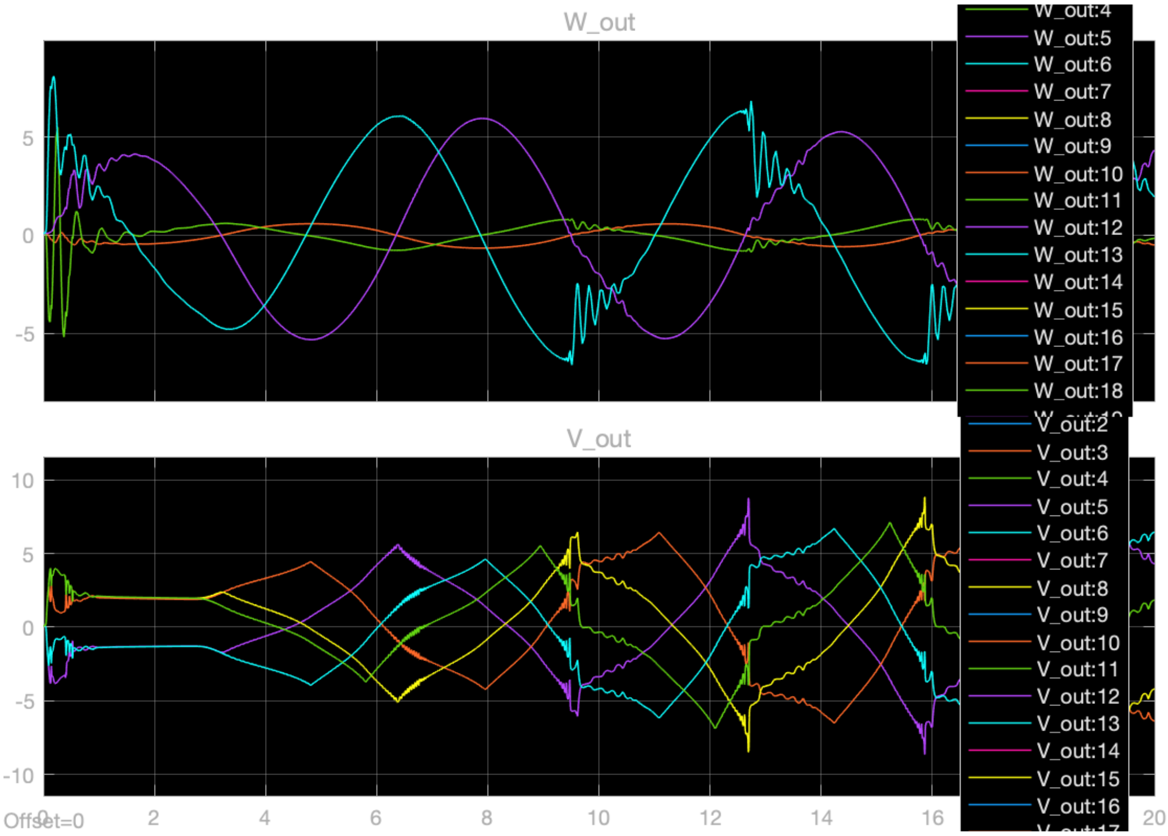

شکل ۵-۱۴: ردیابی حالت ها برای سیستم اصلی غیرخطی و سیگنال مرجع ۱



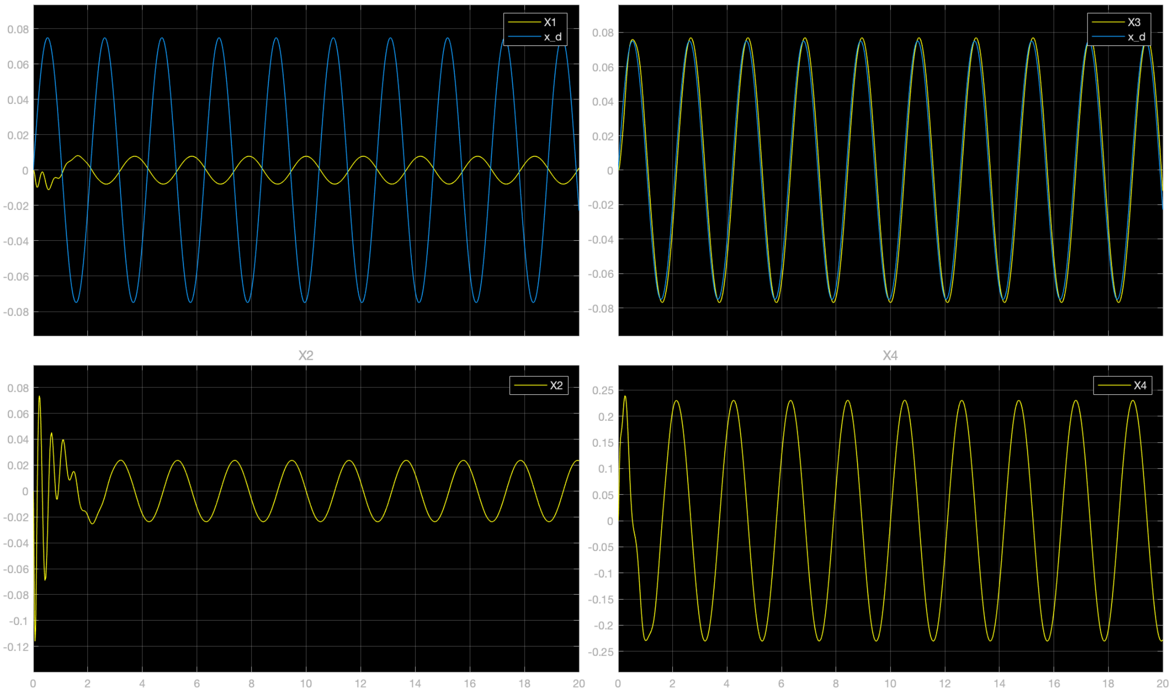

شکل ۵ـ۱۵: ردیابی حالت های رویتگر بر سیستم اصلی برای سیستم اصلی غیرخطی و سیگنال مرجع ۱



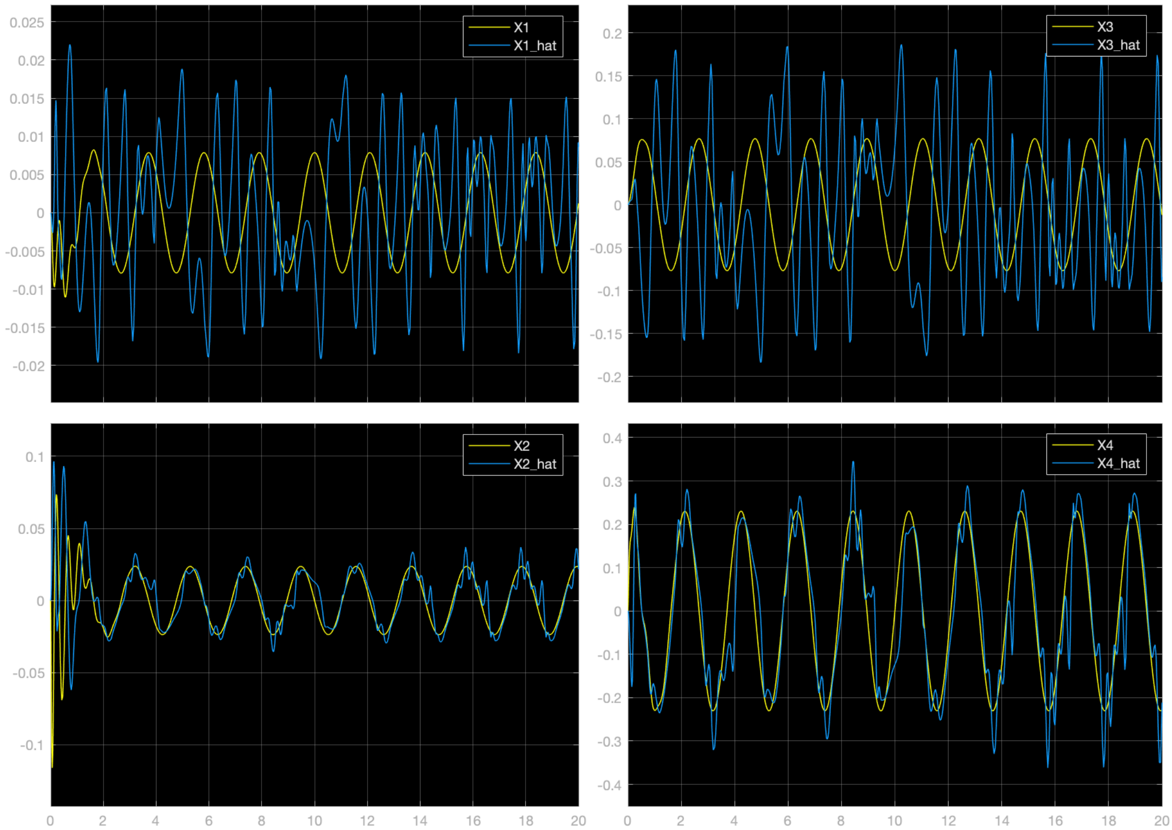

شکل ۵ـ۱۶: خطای ردیابی حالت های رویتگر بر سیستم اصلی برای سیستم اصلی غیرخطی و سیگنال مرجع ۱



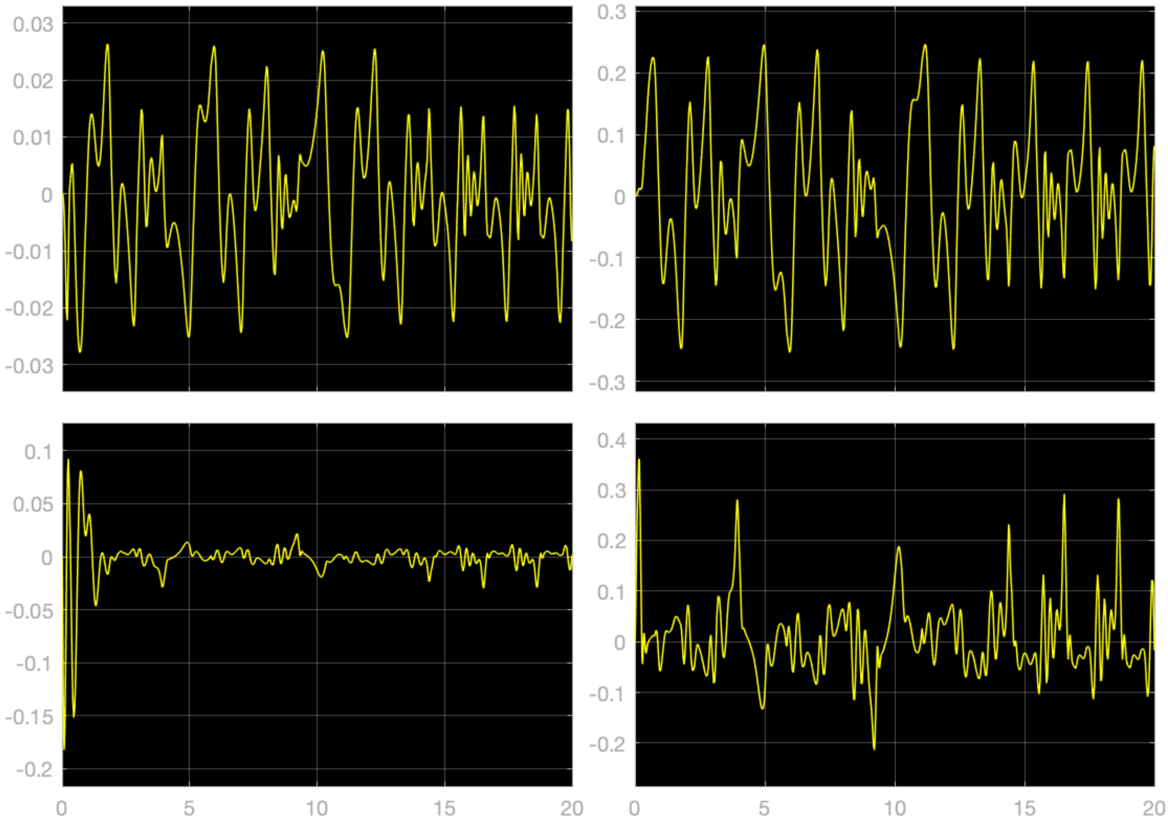

شکل ۵ـ۱۷: ردیابی وزن های رویتگر برای سیستم اصلی غیرخطی و سیگنال مرجع ۱



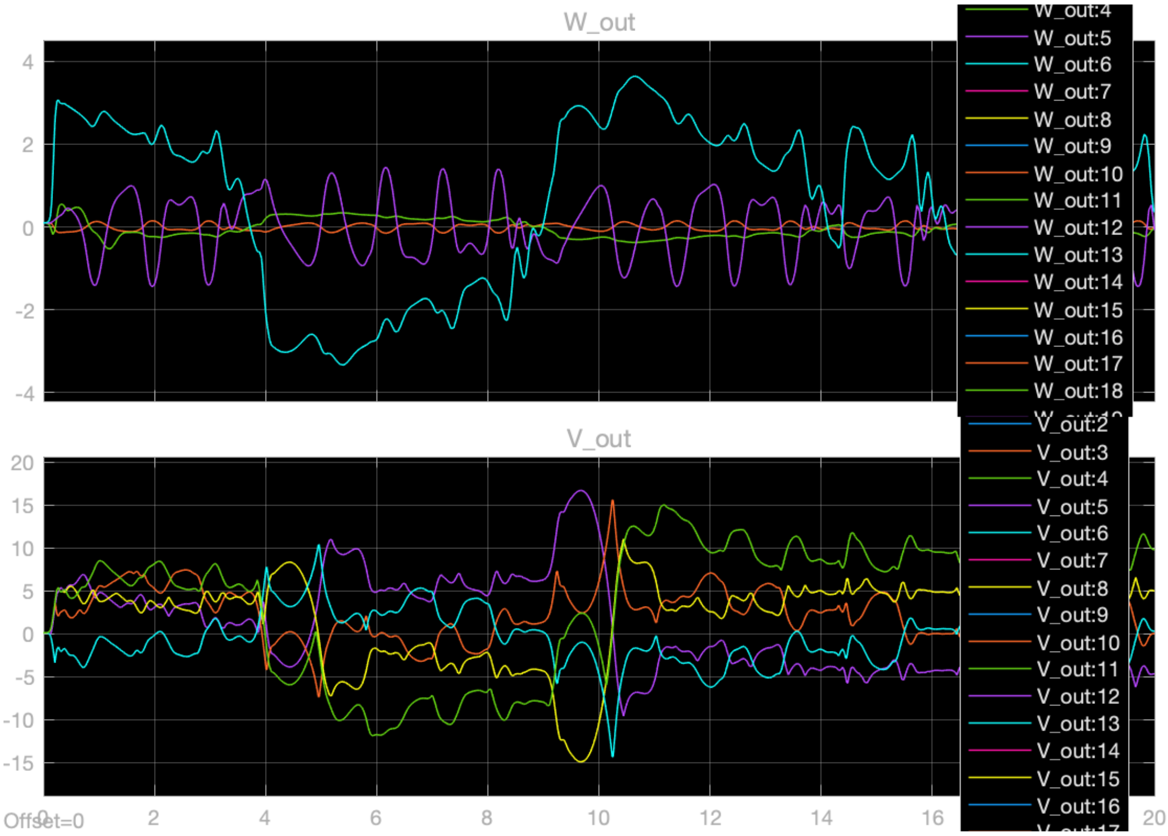

شکل ۵ـ۱۸: ردیابی حالت ها برای سیستم اصلی غیرخطی و سیگنال مرجع ۲



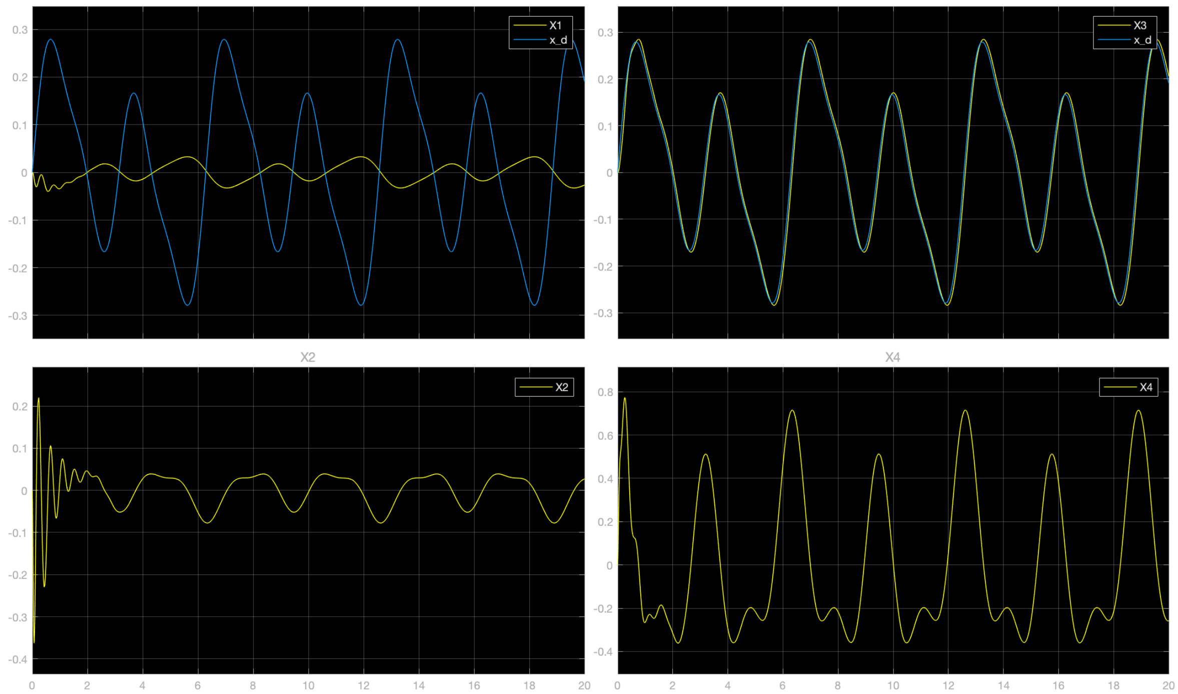

شکل ۵ـ۱۹: ردیابی حالت های رویتگر بر سیستم اصلی برای سیستم اصلی غیرخطی و سیگنال مرجع ۲



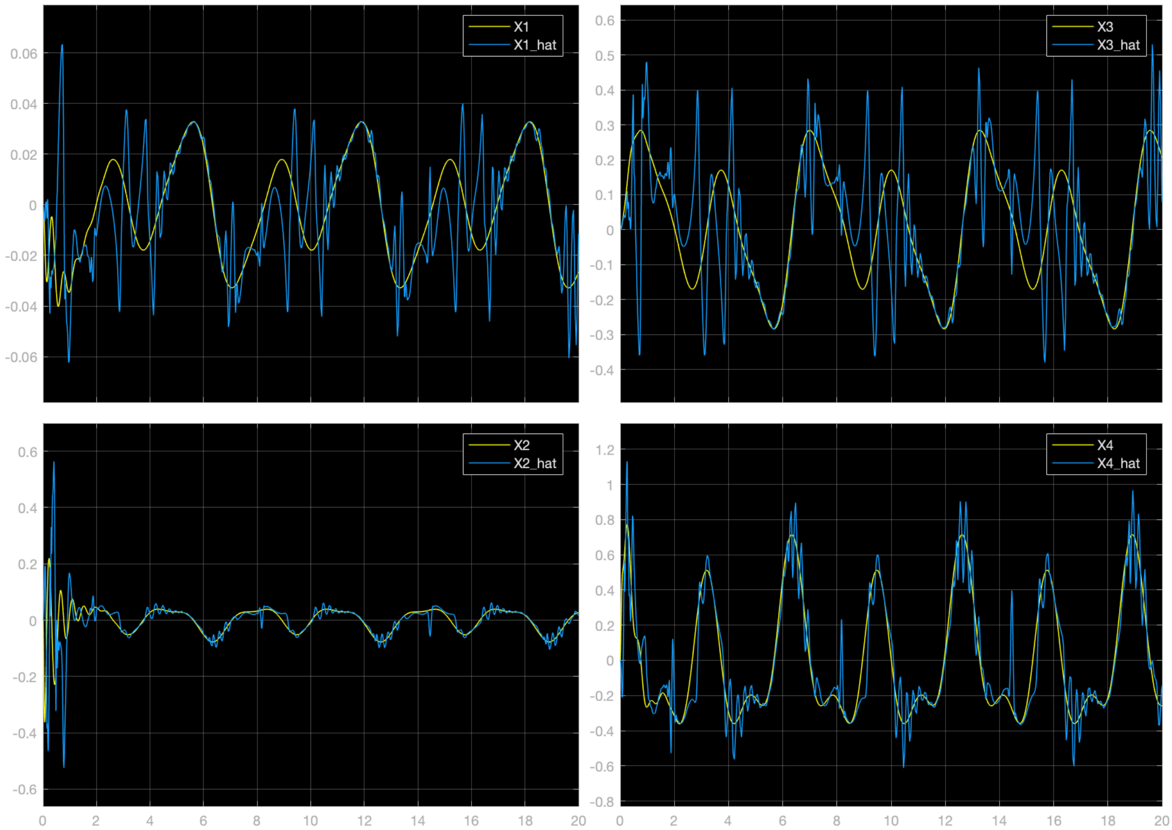

شکل ۵ـ ۲۰: خطای ردیابی حالت های رویتگر بر سیستم اصلی برای سیستم اصلی غیرخطی و سیگنال مرجع ۲



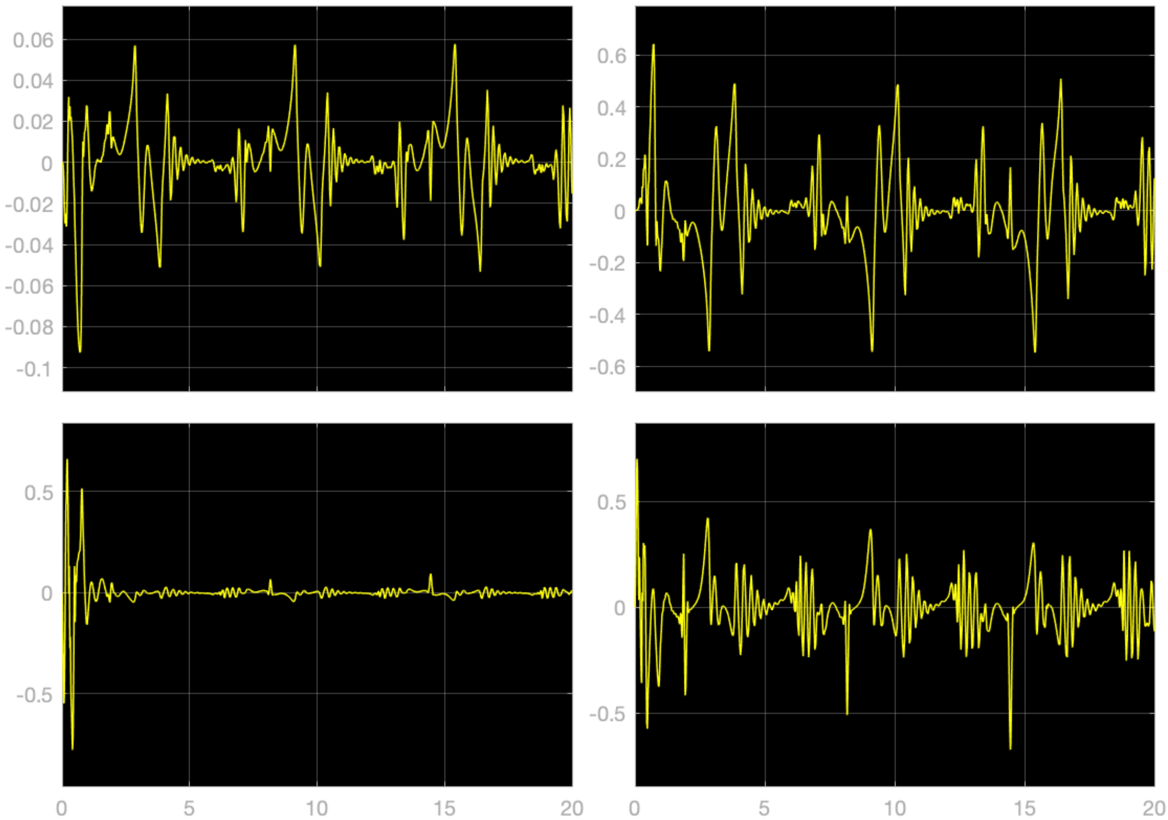

شکل ۵ـ۲۱: ردیابی وزن های رویتگر برای سیستم اصلی غیرخطی و سیگنال مرجع ۲



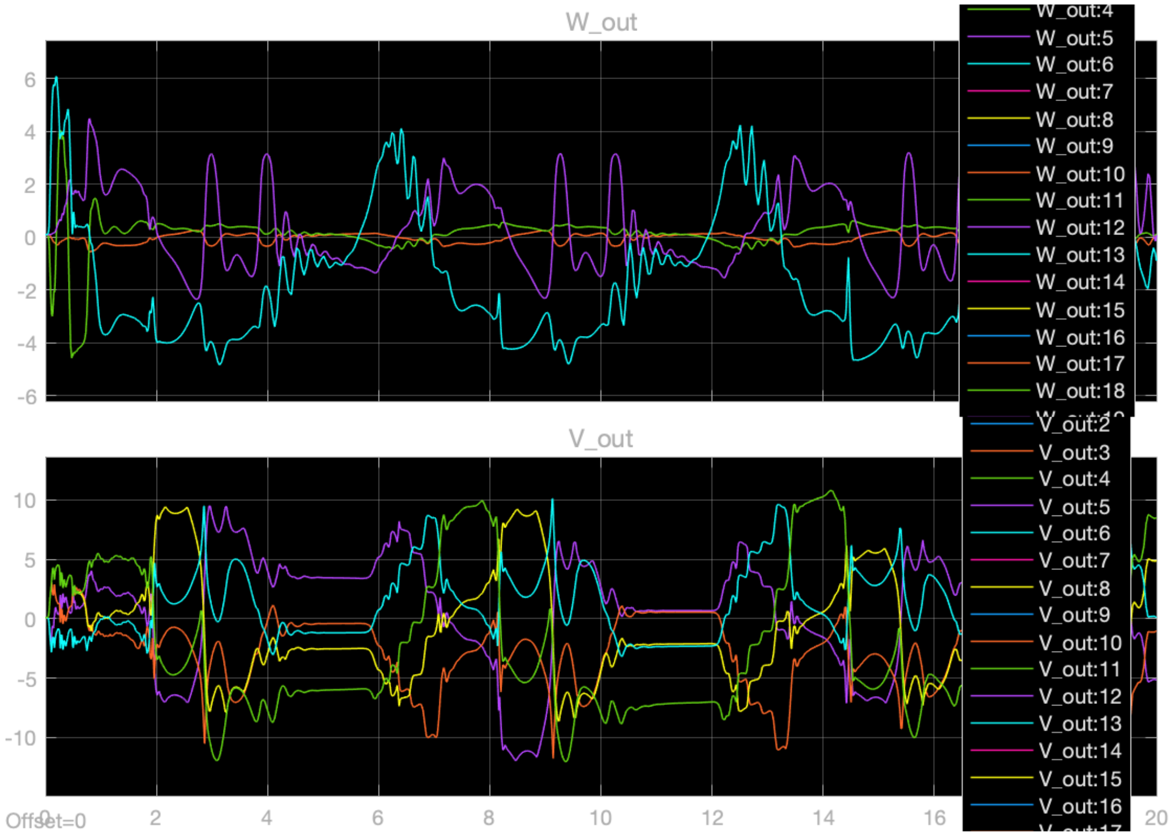

شکل ۵ـ۲۲: ردیابی حالت ها برای سیستم اصلی غیرخطی و سیگنال مرجع۳




شکل ۵ـ۲۳: ردیابی حالت های رویتگر بر سیستم اصلی برای سیستم اصلی غیرخطی و سیگنال مرجع ۳




شکل ۵ــ۲۴: خطای ردیابی حالت های رویتگر بر سیستم اصلی برای سیستم اصلی غیرخطی و سیگنال مرجع ۳




شکل ۵ــ۲۵: ردیابی وزن های رویتگر برای سیستم اصلی غیرخطی و سیگنال مرجع ۳

# فصل ۶

# نتیجه‌گیری

در این گزارش، ابتدا تعاریف ریاضی مورد نیاز را ارائه دادیم. سپس، برای مدل کلی سیستم غیرخطی چندمتغیره، رویتگری تطبیقی پیشنهاد دادیم که در نهایت هم مدل شبکه ی عصبی RBF بازگشتی را در آن گنجاندیم. در ادامه، قانون آموزش این شبکه عصبی را به صورت اصلاحی بر روش پس انتشار خطا بدست آوردیم و بعد هم پایداری آن را توسط روش مستقیم لیاپانوف اثبات کردیم. در قسمت بعد، مدل های خطی و غیر خطی برای سیستم بازوی ربات انعطاف پذیر تک پیونده که سیستم شبیه سازی خاص این پروژه است، ارائه دادیم. در آخر هم، مدل سیستم مورد نظر در MATLAB و SIMULINK را نشان دادیم و نتایج شبیه سازی را به اشتراک گذاشتیم. نتایج شبیه سازی نشان داد که رویتگر پیشنهاد شده با تنظیم درست پارامتر های آموزش به خوبی کار می کند. هر چند نوساناتی هم دارد که برای برطرف کردن آن با پارامتر ها باید ببنیم کدام هدف های کنترلی را حاظریم در مقابل آن از دست بدهیم. هم چنین موقعیت پیوند در سیستم غیرخطی نسبت به کنترل کننده ی PD جواب خوبی نداد و موقعیت را درست دنبال نمی کرد و می توان گفت دارای کرانی بود که از آن بالاتر نمی رفت که این می توان اشکال در مدل سیستم اصلی غیرخطی و یا کنترل کننده ی PD پیشنهاد شده باشد. همین طور می توان گفت که با کم کردن دامنه ی سیگنال مرجع، نواسانات بیشتر تاثیر می گذارند چرا که دامنه ی آن ها ثابت است.

**Abstract**

A stable neural network (NN)-based observer for general multivariable nonlinear systems is presented in this paper. Unlike most previous neural network observers, the proposed observer uses a nonlinear-in-parameters neural network (NLPNN). Therefore, it can be applied to systems with higher degrees of nonlinearity without any a priori knowledge about system dynamics. The learning rule for the neural network is a novel approach based on the modified backpropagation (BP) algorithm. An e-modification term is added to guarantee robustness of the observer. No strictly positive real (SPR) or any other strong assumption is imposed on the proposed approach. The stability of the recurrent neural network observer is shown by Lyapunov's direct method. Simulation results for a flexible-joint manipulator are presented to demonstrate the enhanced performance achieved by utilizing the proposed neural network observer.

**Keywords**: Flexible joint manipulators, neural networks (NN), nonlinear observer.

دانشگاه آزاد اسلامی واحد علوم و تحقیقات

Islamic Azad Univiersity - Science And Research Branch

Department of Control Engineering

Neural Control Course Final Project Report

# A Stable Neural Network-Based Observer With Application to Flexible-Joint Manipulators

By:

**Kian Khaneghahi**

Supervisor:

**Dr. Ali Moarefianpour**

Augest 2021