پایدارسازی دستگاه‌های کنترل غیرخطی با استفاده از قضیه زوبوف و شبکه‌های عصبی مصنوعی

اژدر سلیمان پور باکفایت، نادر دسترنج؛ دانشگاه پیام نور، گروه ریاضی، ایران

دریافت ۹۲/۹/۱۰ پذیرش ۹۳/۷/۲۹

چکیده

قضیهٔ زوبوف یکی از قضایایی است که برای پایداری یک دستگاه غیرخطی با دامنه ربایش معلوم شرایطی را بیان می‌کند. از شبکه‌های عصبی استفاده کرده و با آن‌ها، تعدادی از توابع موجود در قضیهٔ زوبوف را تقریب می‌زنیم، بدین‌ترتیب کنترل‌کنندهٔ یک دستگاه کنترل غیرخطی، که به لحاظ ریاضی یافتن ضابطهٔ کنترل آن آسان نیست، به‌دست می‌آید. در این تحقیق دو استراتژی مختلف را به‌کار می‌گیریم و نهایتاً تأثیر و قابلیت روش‌های مفروض، با مثال‌های عددی توضیح داده شده است.

واژه‌های کلیدی: دستگاه‌های کنترل غیرخطی، پایدارسازی، شبکه‌های عصبی مصنوعی، قضیهٔ زوبوف، بهینه‌سازی نلدر-مید.

مقدمه

پایداری از مفاهیم مهم و ضروری در بررسی و تحلیل دستگاه‌های دینامیکی است. روش‌های پایدارسازی زیادی برای دستگاه‌های کنترل وجود دارد [۶]، در بسیاری از این روش‌ها وجود توابع لیاپانوف ابزار پایدارسازی است. اما این روش در عمل خیلی کاربردی و آسان نیست و از توابع لیاپانوف به‌روش‌های متفاوتی در پایدارسازی دستگاه‌ها استفاده می‌شود و اگر در کاربرد خاصی به‌کار بردن روش تابع لیاپانوف به‌علت دشوار بودن یافتن آن با مشکل روبرو باشد، آنگاه روش‌های عددی ممکن است مفید واقع گردند. به‌خصوص این مقاله گزینهٔ جدیدی در برابر چنین مشکلاتی قرار می‌دهد.

برای پایدارسازی سیستم‌های کنترل ناپایدار با استفاده از شبکه‌های عصبی روش معکوس فرآیند[1] روشی قدیمی اما مهم است. در این روش ورودی‌های مختلف از ناحیه‌ای مشخص اطراف نقطهٔ تعادل به سیستم اعمال شده و خروجی‌های متناظر نگه داشته می‌شوند، سپس شبکه‌ای عصبی عکس فرآیند را آموزش می‌بیند یعنی خروجی‌ها برای شبکه، ورودی محسوب شده و ورودی‌های سیستم خروجی قرار می‌گیرند. پس از آموزش شبکه، شبکهٔ آموزش دیده شده را در مسیر اصلی مدار قرار داده و سیستم پایدار عمل می‌کند (حالت‌ها به نقطه تعادل همگرا می‌شوند). در این حالت خروجی شبکه در خلال یادگیری به‌ازای هر ورودی مشخص است (حالت باناظر)، اما حالتی را در نظر بگیرید که از شبکه‌ای به‌عنوان تقریبی از جواب معادله یا کنترل‌کننده u استفاده می‌شود. در این حالت برای یافتن پارامترهای شبکه (یادگیری) چون خروجی متناظر با هر ورودی در دست

*نویسنده مسئول soleymbakef@gmail.com

1. Plant Invers

[DOR: 20.1001.1.25882546.1394.1.1.4.8]
[DOI: 10.29252/mmr.1.1.51]

نیست از این رو باید معادلات شبکه را در رابطهٔ مربوط به سیستم قرار داده و از روش بهینه‌سازی برای یافتن پارامترهای شبکه استفاده شود (بدون ناظر).

روش دیگر برای پایدارسازی، به‌کارگیری شبکه‌های عصبی مصنوعی است. شبکه‌ها با توانایی یادگیری خود توابع وجودی را آموزش دیده و در موارد مختلف به‌کار می‌روند. شرایط آموزش گاهی موجب می‌شود که یادگیری با ناظر نباشد، یعنی برای یافتن پارامترهای بهین شبکه، خروجی‌های متناظر با ورودی‌ها، معلوم نیستند و شبکه عصبی به‌عنوان یک تابع نامعلوم در دستگاه قرار داده می‌شود. در نتیجه برای آموزش شبکه (تعیین پارامترهای بهین) باید روشی بهینه‌سازی[۲] به‌کار رود.

سو[۳] و همکارانش [۱]، کنترل‌کننده‌ای عصبی طراحی کردند که به‌صورت off-line آموزش دیده به‌نحوی که مشتق زمانی تابعی معین مثبت بر حسب متغیرهای حالت، در تمام ناحیهٔ کاری منفی می‌شود و دستگاه حلقه بسته با کنترل‌کننده عصبی، پایدار می‌گردد. در این مرجع کنترلر u در دستگاه دل‌خواه $\dot{x} = f(x, u)$ با شبکه‌ای RBF به‌صورت $u = N(x, W)$ تقریب زده می‌شود که در آن W بردار پارامترهای شبکه است.

در [۲] عمل پایدارسازی دستگاه کنترل غیرخطی با شبکه‌ای چندلایه و با روش بهینه‌سازی نلدر- مید انجام شده است به‌طوری‌که نتیجه به‌صورت پایداری مجانبی حاصل شده است.

فوراتی و همکارانش [۳]، مسئلهٔ پایدارسازی دستگاه‌های کنترل غیرخطی گسسته با تابع نامعلوم را با شبکه‌ای چندلایه انجام داده‌اند که در آن عمل‌کرد شبکه این بوده است که معکوس فرآیند را به شیوه‌ای متفاوت از معکوس فرآیند مرسوم انجام می‌داد، در این کار یادگیری شبکه به‌صورت off-line و نیز on-line انجام شده است.

در این مقاله، با در نظر گرفتن قضیه زوبوف[۴]، توابع موجود در این قضیه را طی دو استراتژی مجزا و با استفاده از شبکه‌های عصبی تقریب زده و در نتیجه یک روش پایدارسازی برای دستگاه‌های کنترل غیرخطی پیوسته حاصل می‌شود.

شبکه‌های عصبی

شبکه‌های عصبی مصنوعی، از نظر توپولوژی و قانون یادگیری انواع مختلف دارند. یکی از ویژگی‌های مهم شبکه‌های عصبی که کاربرد آن را در علوم مختلف میسر کرده است آموزش‌پذیری آن‌هاست. با این ویژگی یک شبکه عصبی فقط با لایه‌ای پنهان می‌تواند هر تابع پیوسته را تقریب بزند [۷]. در نتیجه شبکه‌های عصبی پرکاربرد مانند شبکه‌های چندلایه و شبکه‌های عصبی RBF[۵] تقریب زننده‌های جهانی[۶] هستند. در این مقاله شبکهٔ چندلایه را به‌کار می‌بریم.

یک شبکهٔ سه‌لایه که لایه‌ای پنهان و دو ورودی و یک خروجی دارد در شکل ۱ دیده می‌شود چنین شبکه‌ای را با علامت $(2 - n - 1)$ نیز نشان می‌دهند که در آن n تعداد نرون‌های لایه پنهان است. توجه داریم که این شبکه نیز تقریب‌زنندهٔ جهانی است و ما همین شبکه را برای برآورد توابع به‌کار می‌گیریم. برای یافتن وزن‌ها و

2. Optimization
3. SSU
4. Zubov's Theorem
5. Radial Basis Function
6. Universal Approximater

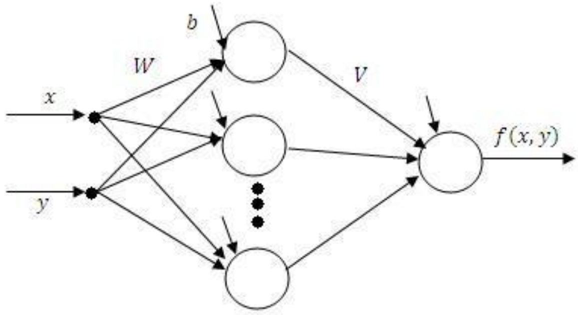
[DOR: 20.1001.1.25882546.1394.1.1.4.8]
[DOI: 10.29252/mmr.1.1.51]

بایاس‌های این شبکه که همان ماتریس‌های W ، V و بردار b هستند روش مشهور یادگیری پس‌انتشار خطا[۷] را به‌کار می‌بریم. برای نشان دادن توانایی تقریب شبکهٔ مذکور شبکه‌ای طراحی می‌کنیم که تابعی دو متغیره را تقریب بزند.

مثال۱: شبکه‌ای چندلایه را برای تقریب تابع $\mathrm{f(x,y) = x^2 + sin(\pi xy)}$ در ناحیهٔ $\mathrm{D = [-1,1] \times [-1,1]}$ در نظر می‌گیریم. برای این‌کار شبکهٔ $(1 - 80 - 2)$ را در نظر می‌گیریم و در هر سیکل[۸]، ۲۰ عدد تصادفی از بازهٔ $[-1,1]$ و در نتیجه ۴۰۰ نقطه از ناحیهٔ D اختیار کرده و شروع به آموزش شبکه می‌کنیم. قبل از شروع یادگیری لازم است پارامترهایی از شبکه تنظیم شوند. این تنظیمات شامل معماری شبکه (شکل ۱) و توابع فعالیت و وزن‌ها و بایاس‌های اولیه هستند. توابع فعالیت زیگموئید است و وزن‌ها و بایاس‌های اولیه به‌طور تصادفی و از بازهٔ $(0,1)$ اختیار شده‌اند. لازم به ذکر است در اکثر موارد بهتر است وزن‌های اولیه کوچک اختیار شوند [۸].

تقریب‌های حاصل، پس از ۸۰۰۰۰ و ۱۰۰۰۰۰ سیکل در شکل ۲ نشان داده شده است. همچنین نمودار دقیق به‌همراه مقادیر خطا در هر مرحلهٔ یادگیری در شکل ۳ آورده شده است. توجه داریم که اگر تابع f دارای ورودی‌ها یا خروجی‌های زیادتری باشد عمل تقریب به‌راحتی به‌همان روش مذکور انجام می‌شود و در مقایسه با درون‌یابی ریاضی، عمل درون‌یابی به‌وسیلهٔ شبکه‌ها در ابعاد بزرگتر خیلی راحت‌تر از درون‌یابی به مفهوم محاسبات عددی است، برای توضیح بیش‌تر تابع $f(x,y) = x^2 + \sin(\pi xy)$ را در نظر می‌گیریم. برای یافتن تابعی درونیاب برای این تابع در ناحیه‌ای مانند D باید از قوانین درون‌یابی دوبعدی استفاده شود، و اگر به‌جای تابع f تابعی مانند $f: \mathbb{R}^5 \to \mathbb{R}^4$ مطرح شود کار به‌مراتب مشکل‌تر می‌شود. اما اگر همان درون‌یابی با شبکه‌ای عصبی انجام شود، با افزایش ابعاد مسئله فقط تعداد نرون‌ها و تعداد ورودی‌ها و خروجی‌های شبکه تغییر می‌یابد و قانون یادگیری ثابت می‌ماند.

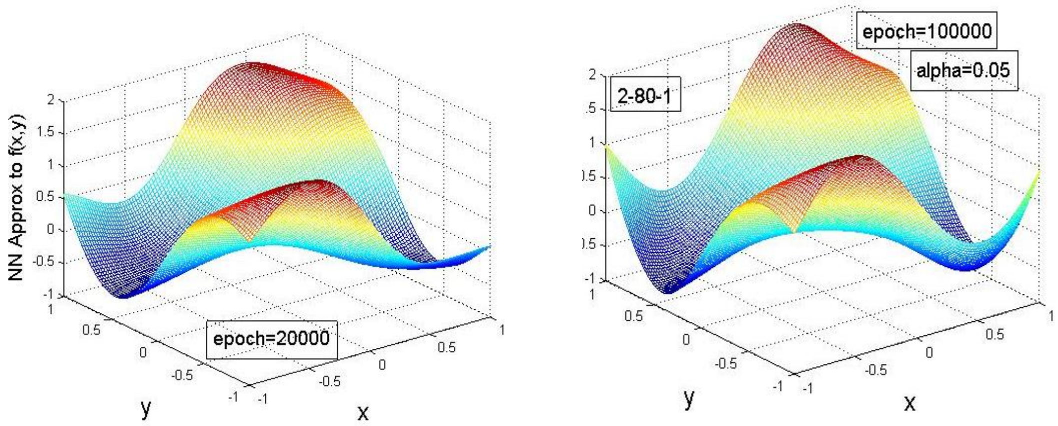
زمانی‌که خطای محاسبه به‌مقدار مورد نظر برسد آموزش را تمام کرده شبکه را استفاده می‌کنیم به این نوع آموزش که پس از آموزش کلی شبکه، از آن استفاده می‌کنیم آموزش off-line می‌نامیم.



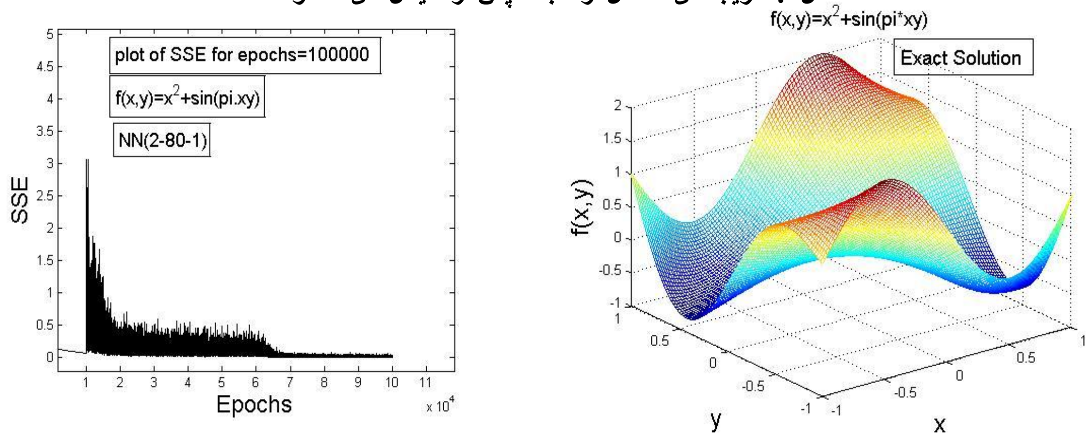
شکل۱. ساختار یک شبکه چندلایه با یک لایه پنهان

7. Back Propagation
8. Epoch

[DOR: 20.1001.1.25882546.1394.1.1.4.8]
[DOI: 10.29252/mmr.1.1.51]



شکل۲. تقریب‌های حاصل از شبکه پس از سیکل‌های متفاوت



شکل۳. نمودار دقیق تابع f در مثال۱ و مقادیر خطا در روند یادگیری

قضیهٔ زوبوف و پایدارسازی توسط آن

در این بخش قضیهٔ زوبوف [۴] را بیان کرده و تأثیر آن در پایدارسازی دستگاه‌های کنترل غیرخطی را بررسی می‌کنیم.

قضیهٔ (زوبوف) دستگاه دینامیکی غیرخطی زیر را با شرط $f(0)=0$ در نظر می‌گیریم:

$$\dot{x}(t)=f\big(x(t)\big) \qquad (۱)$$

فرض می‌کنیم $D\subset\mathbb{R}^n$ مجموعه‌ای کران‌دار است و تابع به‌طور پیوسته مشتق‌پذیر $V\colon D\to\mathbb{R}$ و یک تابع پیوسته مانند $h\colon\mathbb{R}^n\to\mathbb{R}$ موجود باشند که $V(0)=0$ و $h(0)=0$ و

$$0<V(x)<1, x\in D, x\neq 0 \qquad (۲)$$

$$\lim_{x\to\partial D} V(x)=1 \qquad (۳)$$

$$h(x)>0, x\in\mathbb{R}^n, x\neq 0 \qquad (۴)$$

$$\dot{V}=\nabla V(x).f(x)=-h(x)[1-V(x)] \qquad (۵)$$

آن‌گاه جواب صفر $x=0$ برای دستگاه (۱) پایدار مجانبی با دامنهٔ ربایش D است. یکی از ویژگی‌های مهم قضیهٔ مذکور این است که پس از پایدارسازی، دامنهٔ ربایش دستگاه معلوم می‌شود و در واقع از ابتدا دامنهٔ ربایش

[DOR: 20.1001.1.25882546.1394.1.1.4.8]
[DOI: 10.29252/mmr.1.1.51]

(D) را مشخص می‌کنیم. طی دو استراتژی با استفاده از قضیهٔ زوبوف و شبکه‌ها به پایدارسازی دستگاه‌های کنترلی ناپایدار می‌پردازیم.

استراتژی اول

استفاده از قضیهٔ زوبوف مستلزم مشخص کردن توابع V ، h و u است. به‌لحاظ ریاضی یافتن ضابطهٔ توابع مذکور آسان نیست زیرا به حل معادلهٔ دیفرانسیل با مشتقات جزئی با برخی توابع نامعلوم می‌انجامد. از این رو، در این استراتژی از شبکهٔ عصبی چندلایه استفاده می‌کنیم. دستگاه کنترل $\dot{x} = f(x, u)$ را در نظر می‌گیریم که به‌ازای $u = 0$ ناپایدار است، هدف یافتن کنترل‌کننده u است به‌طوری‌که دستگاه حلقه بسته پایدار باشد. ابتدا دامنهٔ ربایش را به‌نام $D \subset \mathbb{R}^n$ مشخص می‌کنیم یک انتخاب برایD بدین‌صورت است:

$$D = \left\{ x \in \mathbb{R}^n | \sum_{i=1}^{n} x_i^2 \leq 1 \right\} \tag{۶}$$

توجه داریم که می‌توان ناحیهٔ D را بزرگ‌تر از (۶) نیز اختیار کرد. اکنون تابع V را به‌صورت $V(x) = \sum_{i=1}^{n} x_i^2$ در نظر می‌گیریم. با این تعریف،V در شرایط قضیهٔ زوبوف صدق می‌کند، در نتیجه باید تابع مثبت $h(x)$ و تابع کنترل u طوری مشخص شوند که این رابطه برقرار باشد:

$$\Delta V(x) f(x, u) = -h(x)[1 - V(x)] \tag{۷}$$

برای یافتن توابع h و u از شبکه‌ای چند لایه استفاده می‌کنیم در این شبکه حدود ۱۰ نرون در لایهٔ میانی قرار دارد و دارای n ورودی و دو خروجی است. ورودی‌ها مؤلفه‌های بردار $x \in \mathbb{R}^n$ است و اولین خروجی برای تابع h و دومین خروجی برای u در نظر گرفته می‌شود (ممکن است u نیز دارای مؤلفه‌های زیاد باشد). تابعی که باید بر اساس (۷) کمینه شود بدین‌صورت است:

$$E(x) = \frac{1}{2} \sum_{x \in D} \varepsilon^2(x) \tag{۸}$$

که در آن

$$\varepsilon(x) = m - n \ , \qquad m = \left[\frac{\partial V}{\partial x_1}, \cdots, \frac{\partial V}{\partial x_n} \right] . \begin{bmatrix} f_1(x, u) \\ \vdots \\ f_n(x, u) \end{bmatrix} \ , n = -h(x) . \left(1 - \sum_{i=1}^{n} x_i^2 \right)$$

نتیجه این‌که تابع E بر حسب u و h بیان خواهد شد. چون هر دوی این توابع از شبکه حاصل می‌شوند و برحسب پارامترهای شبکه بیان شده‌اند، می‌توان گفت تابع E بر حسب وزن‌ها و بایاس‌های شبکه بیان شده است. انتخاب‌های زیادی برای تابع V وجود دارد که در استراتژی دوم به آن‌ها اشاره خواهیم کرد. در آموزش شبکه، داده‌های آموزشی باید از تمام ناحیه D انتخاب شوند تا تمامD بتواند به عنوان دامنه ربایش قرار گیرد.

استراتژی دوّم

در این استراتژي با مشخص کردن توابع $h(x)$ و $V(x)$، تنها تابع مجهول در قضیهٔ زوبوف کنترلر u است که مقدار آن را با شبکهٔ چندلایه تقریب می‌زنیم.

[DOR: 20.1001.1.25882546.1394.1.1.4.8]
[DOI: 10.29252/mmr.1.1.51]

می‌دانیم تابع نرم به‌صورت $\|.\|: X \to \mathbb{R}^{\geq 0}$ تعریف می‌شود که در آن X فضایی برداری است. این تابع، مثبت و پیوسته و $\|0\| = 0$ است. در نتیجه می‌توان در قضیهٔ زوبوف به‌جای تابع h از تابع نرم استفاده کرد. چون فضاهای کاری در مسائل کنترل متناهی‌البعد هستند و در این فضاها همهٔ نرم‌ها معادل هستند از این رو، نرم $\|.\|$ را همان نرم اقلیدسی در نظر می‌گیریم. اکنون تابع V را بدین‌صورت در نظر می‌گیریم:

$$V(x) = \frac{1 - e^{-c\|x\|}}{1 + e^{-c\|x\|}} = \frac{e^{c\|x\|} - 1}{e^{c\|x\|} + 1} \tag{۹}$$

اگر مقدار $V(x)$ را در (۹) برابر k فرض کرده و مقدار c را بیابیم خواهیم داشت:

$$c = \frac{1}{\|x\|} ln\left(\frac{1 + k}{1 - k}\right) \tag{۱۰}$$

به‌ازای cی به‌دست آمده در (۱۰) ، در نقطهٔ x ، مقدار تابع V برابر k می‌شود. به‌عبارت دیگر اگر $\|x\| = m$ فرض شود آن‌گاه به‌ازای همین مقدار در (۱۰) و cی به‌دست آمده، مقدار تابع V در دامنهٔ $[-m, m]$ متعلق به بازهٔ $[0, k]$ است به‌طوری‌که در نقاط انتهایی بازهٔ مذکور مقادیر V به k می‌رسد برای تکمیل شرایط روی تابع V ، آن را داخل قدرمطلق می‌گذاریم تا در قسمت منفی نیز شرایط منطبق بر شرایط V در قضیه زوبوف باشد. در نهایت به‌ازای

$$c = \frac{1}{m} ln\left(\frac{1 + k}{1 - k}\right) \tag{۱۱}$$

تابع

$$V(x) = \left|\frac{e^{c\|x\|} - 1}{e^{c\|x\|} + 1}\right| \tag{۱۲}$$

تابع مذکور در قضیهٔ زوبوف است. نمودار این تابع در $\mathbb{R}^1$ به‌ازای $k = \sqrt{0.99}$ و mهای مختلف در شکل ۴ (a) و این نمودار در $\mathbb{R}^2$ به‌ازای $k = \sqrt{0.99}$ و $m = 10$ ، در شکل ۴ (b) نشان داده شده است. تقریبی از دامنهٔ ربایش D، در اعداد حقیقی، به‌صورت $[-m, m]$ است. با مفروضات مذکور، با توجه به رابطهٔ (۷) تنها مجهول مسئلهٔ تابع کنترل u است که در این استراتژی مقدار این تابع را با یک شبکه عصبی دارای لایه‌ای پنهان تقریب می‌زنیم، در واقع پارامترهای شبکه شکل ۱ را طوری می‌یابیم که شرایط قضیه زوبوف برقرار شود. به‌عبارت دیگر پس از جای‌گذاری معادلات شبکه در دستگاه، که دارای پارامترهای تصادفی است از روش بهینه‌سازی استفاده کرده و پارامترهای بهین را برای شبکه پیدا می‌کنیم. پس در این قسمت شبکهٔ عصبی ورودی متناظر با مؤلفه‌های x و خروجی متناظر با u دارد.

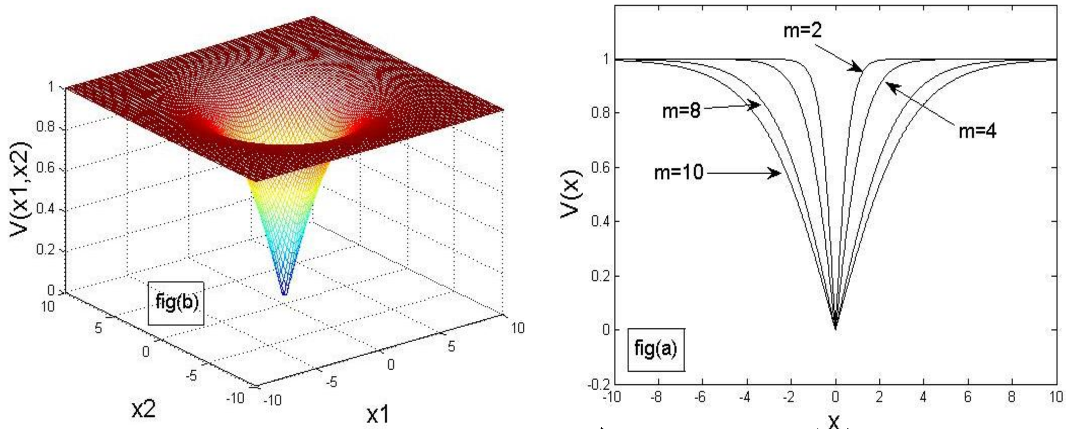
تابع انرژی که در این قسمت باید کمینه شود عبارتست از

$$E = \frac{1}{2}\sum_{x \in D} \varepsilon_x^2 \tag{۱۳}$$

که در آن

$$\varepsilon_x = \nabla V(x) f(x, u) + \|x\|\big(1 - V(x)\big)$$

[DOR: 20.1001.1.25882546.1394.1.1.4.8]
[DOI: 10.29252/mmr.1.1.51]



شکل ۴. نمودارهای مربوط به تابع $V(x)$ در دو و سه بعدی

$$= \left[\frac{\partial V}{\partial x_1}, \cdots, \frac{\partial V}{\partial x_n}\right] \cdot \begin{bmatrix} f_1(x,u) \\ \vdots \\ f_n(x,u) \end{bmatrix} + \|x\|\big(1 - V(x)\big) \qquad (۱۴)$$

همچنین $\|x\| = \sqrt{x_1^2 + \cdots + x_n^2}$. با قرار دادن شبکه در سیستم دینامیکی به‌جای کنترل‌کننده u شرایط قضیهٔ زوبوف را برقرار می‌کنیم. با این توضیحات، شرایط قضیهٔ زوبوف برقرار شده است مگر شرط (۵). با تعریف ε_x مطابق با رابطهٔ (۱۴) زمانی‌که E مینیمم شود شرط (۵) از قضیهٔ زوبوف نیز برقرار می‌شود.

برای کمینه کردن مقدار E در هردو استراتژی باید یکی از روش‌های بهینه‌سازی نامقید را به‌کار ببریم. از جمله این روش‌ها می‌توان به‌روش نیوتن، روش اصلاح شده نیوتن بنام لیونبرگ[۹] و روش نلدر-مید اشاره کرد. برای انجام کمینه‌سازی مذکور از روش بهینه‌سازی بدون مشتق نلدر-مید استفاده کردیم. این روش از روش‌های بهینه‌سازی نامقید است که بر اساس جستجو و محاسبه مقدار تابع در رئوس یک سادک اجرا می‌شود و در یافتن کمینه تابعی چند متغیره، کارایی خوبی دارد. برای توضیح بیش‌تر در رابطهٔ با الگوریتم مربوط به [۵] مراجعه شود.

در رابطه با اثبات همگرایی الگوریتم به‌کار رفته شده می‌توان گفت: همگرا بودن روش نلدر-مید به نقطه بهینه یک تابع که دارای نقطه بهینه است در مرجع [۵] ثابت شده است. اما این‌که چرا فرآیند یادگیری برای رسیدن به پارامترهای بهینه شبکه، همگرا است؟ پاسخ این است که تابع هزینه ایجاد شده در E مجموع مجذورات جملاتی است که حاصل جمع آن‌ها یک تابع محدب ایجاد می‌کنند و همواره دارای نقطهٔ بهین است و در چنین حالتی روش نلدر-مید همگرا است.

مثال۱ (استراتژی اول) دستگاه کنترل غیرخطی و ناپایدار زیر را در نظر می‌گیریم:

$$\begin{cases} \dot{x}_1(t) = 2x_1(t) + x_2(t) + x_1^2(t) + x_1(t)x_2(t)u \\ \dot{x}_2(t) = x_1(t) - x_2(t) + x_1(t)x_2(t) - 0.5x_1(t)u \end{cases} \qquad (۱۵)$$

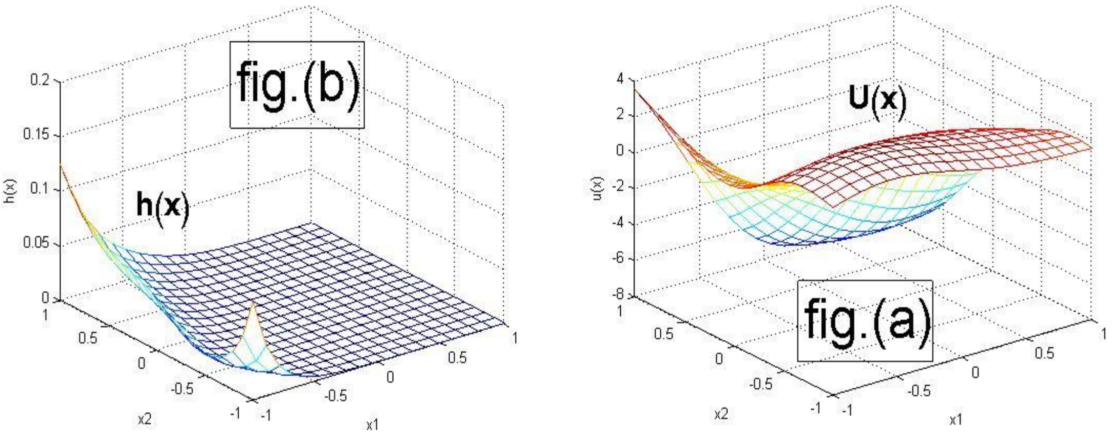
این دستگاه را با استفاده از استراتژی اول پایدار می‌کنیم. فرض می‌کنیم $V(x) = x_1^2 + x_2^2$ بوده و پس از ساده کردن داریم:

$$m = 4x_1^2 - 2x_2^2 + 2x_1^3 + 2x_1x_2(2 + x_2) + x_1x_2(2x_1 - 1)u$$
$$n = h(x)(x_1^2 + x_2^2 - 1)$$

9. Levenberg Marquardt (LM).

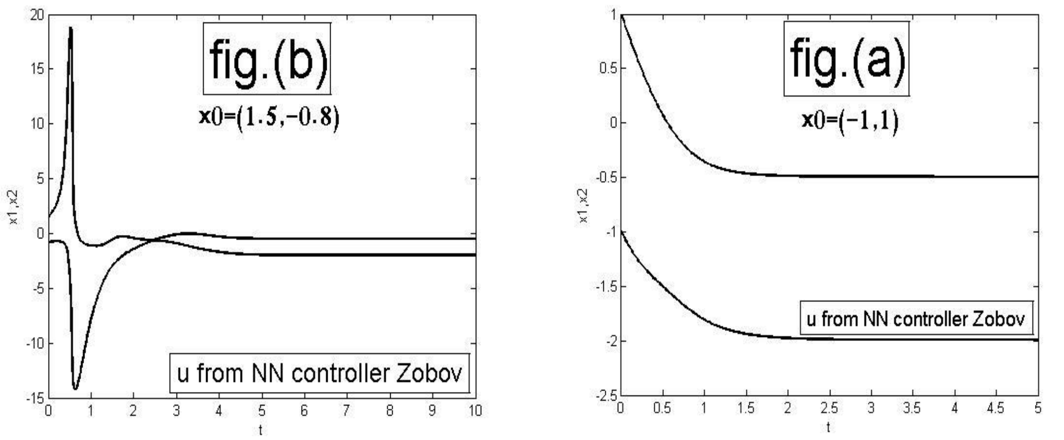
[DOR: 20.1001.1.25882546.1394.1.1.4.8]
[DOI: 10.29252/mmr.1.1.51]

و می‌دانیم: $D=\{(x_1,x_2)|x_1^2+x_2^2\le ۱\}$. پس از آموزش شبکه روی ناحیهٔ D به‌طوری‌که E کم‌ترین مقدار را دارد، نمودارهای مربوط به $h(x)$ و $u(x)$ که از شبکه حاصل شده‌اند در شکل ۵ دیده می‌شوند.



شکل۵. نمودارهای مربوط به توابع $h(x)$ و $u(x)$ در مثال ۱

اکنون اگر کنترل‌کننده عصبی آموزش دیده شده (شبکه‌ای که پارامترهای بهین آن با روش بهینه‌سازی با مینیمم کردن E پیدا شده است) را به‌دستگاه اعمال کنیم آن‌گاه پاسخ دستگاه به‌ازای حالت‌های اولیّه $x_0=(1.5,-0.8)$ و یک‌بار دیگر به‌ازای $x_0=(-1,1)$ در شکل۶ دیده می‌شود.



شکل۶. نمودارهای مربوط به پاسخ دستگاه (۱۵) به‌ازای حالات اولیه متفاوت

مثال۲ (استراتژی دوم) دستگاه کنترل غیرخطی و ناپایدار زیر را در نظر می‌گیریم:

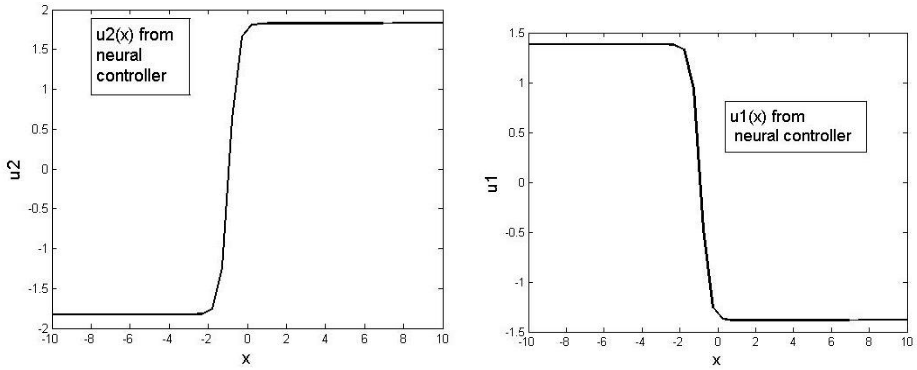
$$\dot{x}=3x+x^2+u_1x+u_2^2 \qquad (۱۶)$$

برای یافتن تابع کنترل $u=(u_1,u_2)$ از شبکه‌ای عصبی چندلایه با لایه‌ای پنهان استفاده می‌کنیم توجه داریم در این استراتژی در روابط موجود در قضیهٔ زوبوف فقط تابع کنترل u مجهول است و بقیه معلوم هستند. در شبکهٔ عصبی مفروض فرض می‌کنیم بردار وزن‌های لایه ورودی به لایهٔ پنهان W ، بردار بایاس در لایهٔ پنهان b ، و بردار وزن‌های لایه خروجیV باشند. دستگاه داده شده در (۱۶) به‌ازای ورودی صفر ناپایدار است و هدف ما یافتن تابع $u=(u_1,u_2)$ است به‌طوری‌که با شروع بردارهای حالت اولیه در ناحیه‌ای دستگاه به‌صورت پایدار عمل کند.

[DOR: 20.1001.1.25882546.1394.1.1.4.8]
[DOI: 10.29252/mmr.1.1.51]

تعداد نرون‌ها در لایهٔ پنهان را برابر ۱۲ فرض کرده و از روش نلدر-مید شروع به آموزش شبکه می‌کنیم. عدد ۱۲ تعداد نرون‌ها در شبکه عصبی است. روش‌های زیادی برای این که تعداد نرون‌ها چند تا باشد در مباحث شبکه‌های عصبی وجود دارند اما هیچ روش جامع وجود ندارد که در همه موارد به‌کار رود. یکی از آن‌ها این است که با تعداد کم شروع کرده و تا زمانی که حاصل تابع هزینه E کم می‌شود آن‌ها را افزایش دهیم.

پس از ۸۰۰۰ تکرار از روش نلدر- مید و فرض این‌که $D = [-10,10]$ ، توابع u_1 و u_2 مشخص می‌شوند نمودار این دو تابع در شکل۷ دیده می‌شوند.



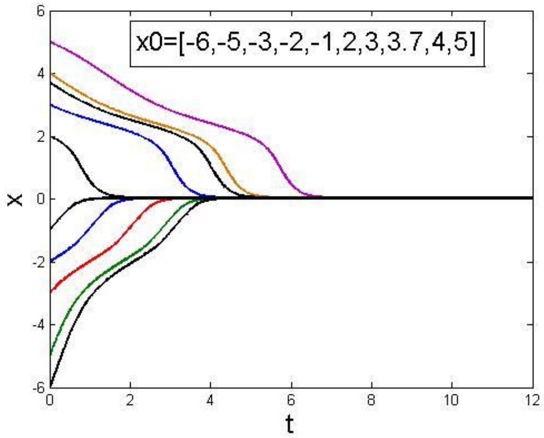
شکل۷. نمودارهای مربوط به توابع کنترل u_1 و u_2 در مثال۲

همچنین پاسخ دستگاه به‌ازای نقاط حالت‌های اولیه $x_0 = [-6, -5, -3, -2, -1,2,3,3.7,4,5]$ در دستگاه مختصات پس از اعمال کنترلر عصبی حاصل در شکل ۸ دیده می‌شود. در این شکل و چنان‌که قبلاً توضیح داده شد انتظار داریم دامنه ربایش دستگاه برابر $D = [-10,10]$ باشد و حالت‌های اولیه نیز از این ناحیه اختیار شده‌اند. اگر آموزش شبکه کامل صورت گیرد بنا به قضیه زوبوف پایداری مجانبی حاصل می‌شود.

پارامترهای شبکه به‌کار رفته شده پس از اتمام آموزش بدین‌صورت هستند:

$$W = [2.2937 , -2.5761 , 0.2837, -0.2234 , 0.0398 , -1.3215$$
$$-0.1947 , -2.2379 , -1.5364 , 2.9923 , 1.2426 , -0.2989]^t$$
$$b = [2.1186 , -1.8594 , -2.4551, -1.1010 , 0.1047 , 1.9417$$
$$-2.7199 , -0.5538 , -1.3765 , 0.1346 , -1.6347 , -0.8470]^t$$
$$V = \begin{bmatrix} -1.3838 , 1.8257 , -2.5217 , 2.2784 , -4.4860 , 0.7919 , 3.3278 & -0.4618 , 0.9270 , -0.0332 , 0.2307 , 0.4756 \\ 0.1068 , -0.9855 , -0.6845 , 0.6527 , -0.6338 , 0.2966 , 1.3178 & -0.2090 , -0.1386 , -0.6036 , 1.5941 , -2.2884 \end{bmatrix}$$

[DOR: 20.1001.1.25882546.1394.1.1.4.8]
[DOI: 10.29252/mmr.1.1.51]



شکل۸. پاسخ دستگاه (۱۶) **با کنترل کننده عصبی برای حالت‌های اولیه متفاوت**

توجه: اگر تعداد حالت‌ها بیش‌تر از یک باشد آن‌گاه در استفاده از قضیهٔ زوبوف لازم می‌شود که مشتق V نسبت به هر x_i ها محاسبه شود. در حالت کلی از رابطهٔ (۹) و این‌که $x \in \mathbb{R}^n$ داریم:

$$\frac{\partial V}{\partial x_i} = \frac{2cx_i e^{c\|x\|}}{\|x\|(1+e^{c\|x\|})^2}$$

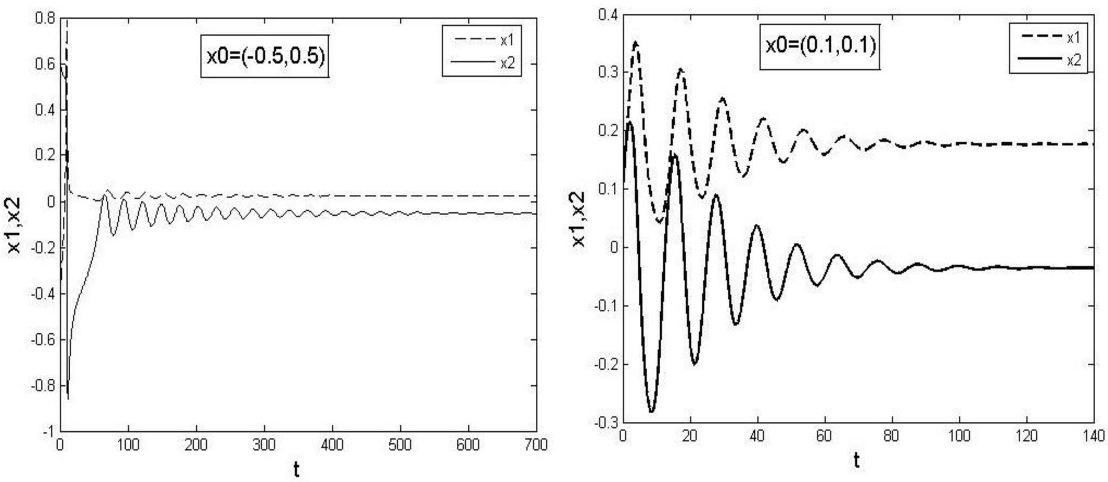
مثال۳ (استراتژی دوم) دستگاه کنترل غیرخطی و ناپایدار زیر را در نظر می‌گیریم:

$$\begin{cases} \dot{x}_1(t) = x_1^3(t) + x_1(t)x_2(t) + u_1 x_1^2(t) + u_2^2 \\ \dot{x}_2(t) = -x_1(t) + x_2^2(t) + x_1(t)x_2(t) - x_1^3(t) + u_1 u_2 \end{cases} \tag{۱۷}$$

به‌ازای ورودی صفر این دستگاه ناپایدار است. شبکهٔ به‌کار رفته در این مثال برای یافتن توابع کنترل دارای دو ورودی و دو خروجی است. تعداد نرون‌های لایه پنهان را برابر ۱۰ تا فرض کرده و

$$D = [-1,1] \times [-1,1]$$

پس از ۶۰۰۰ تکرار روش نلدر- مید و اتمام آموزش شبکه، پاسخ دستگاه به‌ازای حالت‌های اولیه متفاوت در شکل ۹ آورده شده است.



شکل۹. نمودارهای پاسخ دستگاه (۱۷) **در مثال۳**

پارامترهای شبکه به‌کار رفته شده عبارتند از:

$$W = \begin{bmatrix} 0.2004 , -0.7459 , 0.3592 , 0.9912 , 0.0607 , 1.0414 , -0.63274 \\ 0.7336 , 0.6182 , 0.7918 , 0.6348 , -0.5378 , 0.9503 , 0.9882 \\ 0.2198 , 0.6748 , 0.6549 \\ 0.3127 , 0.0682 , 0.1489 \end{bmatrix}^t$$

$$b = [0.1549 , -0.1560 , -0.5767, 1.0795 , 0.6788 , -0.7433 , -0.4201 \quad 0.2340 , 0.5846 , 0.8318]^t$$

$$V = \begin{bmatrix} 0.5743 , 1.2107 , 1.3436 , 0.2215 , 0.1073 , -0.5759 , 0.7495 \\ 0.0704 , -0.0035 , 0.2753 , 0.0056 , 0.5225 , -0.0883 , -0.3264 \\ 0.3630 , -0.3302 , 0.2381 \\ -0.3430 , 0.3667 , -0.7979 \end{bmatrix}$$

نتیجه‌گیری و پیشنهاد

با توجه به این تحقیق می‌توان گفت یکی از روش‌های استفاده از شبکه‌ها در پایدارسازی این است که روشی تحلیلی مانند قضیهٔ زوبوف معلوم باشد و تعدادی از توابع موجود در آن را با شبکه‌ها تقریب بزنیم. در این راستا نوع شبکه‌ها، نوع قانون یادگیری و قانون بهینه‌سازی در صورت لزوم توانایی روش را در پایدارسازی مشخص می‌کنند. بدون استفاده از روش‌های تحلیل نیز می‌توان از شبکه‌ها در پایدارسازی استفاده کرد مانند روش معکوس فرآیند. یک پیشنهاد برای کار بعدی این است که در تمام قضایایی مانند قضیه زوبوف که متضمن پایداری دستگاه هستند می‌توان برای تقریب بهتر توابع، قسمتی را از شبکه‌های عصبی و قسمتی را با چندجمله‌ای‌های متعامد مانند لژاندر[10] یا چبیشف[11] تقریب زد.

در مورد محدودیت‌های استفاده از شبکه‌های عصبی می‌توان گفت: شبکه‌های عصبی روشی نیست که با آن تمام مسائل حل شوند بلکه توانایی یادگیری آن باعث شده است در موارد محاسباتی خاصی مانند این مقاله، و هزاران مورد دیگر عمل‌کردی داشته باشد که تحلیل ریاضی مستقیم از انجام آن ناتوان است.

تشکر و قدردانی

از همهٔ اساتید و دانشجویانی که در این تحقیق به نوعی همکاری داشته‌اند، به خصوص از زحمات اساتید بزرگوار آقای دکتر منهاج و دکتر خالوزاده، همچنین از عوامل مجلهٔ علوم دانشگاه خوارزمی، خصوصاً داورهای معزز نهایت تشکر را دارم. از خداوند متعال توفیق روزافزون آن‌ها را خواستارم.

منابع

1. Yu H., SSU, Annaswamy A.M., "Stable Neural Controllers for Nonlinear Dynamics Systems, Automatica", 34 (1998) 641-650.
2. Soleymanpour Bakefayat A., Heydari A., "Stabilization of Nonlinear Control Systems using a Hybrid Neural Network-Optimization Method", Wulfenia, 19 (2012) 8-21.

10. Legendre Polynomials
11. Chebyshev Polynomials

3. Fourati F., Chtouron M., Kamoun M., "Stabilization of Unknown Nonlinear Systems using Neural Networks", Applied Soft Computing 8 (2008) 1121-1130.
4. Haddad W.M., Chellboina V., "Nonlinear Dynamical Systems and Control", Princeton University Press, Princeton and Oxford (2008).
5. Rao S.S., "Engineering Optimization", Theory and Practice, Purdue University, West Lafayette, Hudiana (1996).
6. Slotine J-J.E., Li W., "Applied Nonlinear Control. Prentice-Hall, Englewood Cliffs", New Jersey (1991).
7. Fausett L., "Fundamental of Neural Networks: Architecture", Algorithms, and Applications, Prentice Hall (1994).
۸. سلیمانپور باکفایت اژدر، *شبکه‌های عصبی مصنوعی در علوم پایه*، انتشار حقیقی، چاپ اول، بهار (۱۳۹۳).

[DOR: 20.1001.1.25882546.1394.1.1.4.8]
[DOI: 10.29252/mmr.1.1.51]