

الگوریتمهای جدید بر اساس اتوماتای یادگیر برای یافتن پارامترهای ویولت

محمد رفیع خوارزمی

دانشکده صنعت الکترونیک - دانشگاه شیراز

محمدرضا میبدی

استاد دانشکده مهندسی کامپیوتر - دانشگاه صنعتی امیرکبیر

(تاریخ دریافت ۸۱/۳/۱۹، تاریخ دریافت روایت اصلاح شده ۸۲/۱۱/۳، تاریخ تصویب ۸۳/۱/۲۹)

چکیده

ویولتها قادر هستند که یک تابع را در دقتهای مختلف تقریب بزنند. این خاصیت آنها را برای مدل سازی و تقریب سیستمهای غیر خطی مناسب ساخته است. برای بدست آوردن یک تقریب مناسب محاسبه پارامترهای توابع پایه ویولت از اهمیت زیادی برخوردار است. برای محاسبه پارامترهای توابع پایه ویولت روشهای متعددی از جمله روشهای بر مبنای شبکه های عصبی (ویولت) بکار برده شده است. در این مقاله روش جدیدی بر اساس اتوماتای یادگیر برای محاسبه پارامترهای ویولت ارائه گردیده است. روش پیشنهادی از قدرت تصمیم گیری و جستجوی اتوماتای یادگیر استفاده کرده و در هر مرحله با اضافه کردن ویولت جدید به تقریب قبلی و حذف ویولتهایی که به کاهش خطا کمکی نمی کنند سعی در کاهش خطا می نماید. روشهای پیشنهادی بر خلاف روش یادگیری L_2 که نیاز به محاسبات بسیار بالائی (از جمله معکوس کردن یک ماتریس با ابعاد بالا برای تعداد زیاد نمونه ها) از محاسبات کمتر و بسیار ساده تری برخوردار است. همچنین بر خلاف برخی روشهای بر پایه شبکه های عصبی که دارای ساختار ثابتی هستند دارای ساختاری پویا می باشد یعنی تعداد توابع پایه از قبل مشخص نمیگردد بلکه در حین فرایند تقریب تعیین میشود. روش پیشنهادی بر روی مسائل متنوعی آزمایش شده است و نتایج بسیار خوبی بدست آمده است.

واژه های کلیدی: ویولت، اتوماتای یادگیر، الگوریتمهای یادگیری، شبکه های عصبی

مقدمه

می توان با آنها تقریب زد [۴][۳][۲]. در سالهای اخیر کارهای متعددی در ارتباط با تلفیق شبکه های عصبی و بسط ویولت انجام شده است. زانگ و بنونسیت [۴] یک شبکه $(1+1/2)$ لایه ای با الگوریتم آموزش انتشار خطا به عقب برای تعیین پارامترهای ویولت پیشنهاد کرده اند. بخشی و استفانوپولس [۵] شبکه ای به نام ویولت پیشنهاد نمودند که در آن از الگوریتم انتشار خطا به عقب برای آموزش آن استفاده میشود. برای تخمین ضرائب ویولت توسط شبکه های ویولت، آموزش دقیق شبکه معمولاً با مشکل مواجه می شود و یا اینکه خطا از حد معینی کمتر نمی شود. در روش یادگیری L_2 خطای بدست آمده حداقل خواهد بود [۶] اما محاسبات زیاد از جمله محاسبه معکوس یک ماتریس با ابعاد بزرگ از مشکلات آن می باشد.

در این مقاله روشی مبتنی بر اتوماتای یادگیر برای تعیین پارامترهای ویولت بمنظور تقریب توابع پیشنهاد میگردد. یکی از مزایای روشهای پیشنهادی تعیین تعداد توابع پایه

نمایش یک تابع به صورت بسط یک سری توابع پایه نقش مهمی در پردازش سیگنال، مدل سازی و تقریب توابع دارد. به عنوان مثال از بسط فوریه در دستگاه مثلثاتی برای پردازش سیگنال در حوزه فرکانس ویا از شبکه های عصبی به عنوان ابزاری برای مدل سازی و تقریب سیستم های غیر خطی بر اساس داده های ورودی خروجی استفاده شده است. اما ضرائب فوریه بیشتر خواص فرکانسی سیگنال را بیان می کنند. بنابراین بسط یک تابع بر اساس یک سری توابع پایه که به خوبی در بعد زمان و فرکانس متمرکز شده باشند بسیار مفید خواهد بود. بسط توابع با توجه به تئوری ویولت، که یک تابع را به صورت مجموعه ای از توابع پایه در وضوحهای مختلف نشان می دهد این نیاز را برآورده می سازد [۱]. کارهای گزارش شده نشان میدهد که توابع پایه ای وجود دارد که هر تابع قابل اندازه گیری روی محور حقیقی R را

$$F_0(x) = \sum_k a_{0,k} \phi_{0,k}(x)$$

$\phi_{m,k}(x)$ را تابع مقیاس کننده می نامند و به صورت زیر تعریف می شود.

$$m, k \in \mathbb{Z} \quad \phi_{m,k}(x) = 2^{-m/2} \phi(2^{-m}x - k) \quad (۳)$$

بنابر این خواهیم داشت

$$F_0(x) = \sum_k a_{0,k} \phi_{0,k}(x)$$

$$F_{-1}(x) = F_0(x) + \sum_k d_{0,k} \psi_{0,k}(x)$$

$$F_{m-1}(x) = F_m(x) + \sum_k d_{m,k} \psi_{m,k}(x)$$

و این عمل تا زمانی که یک تقریب مناسب بدست آید ادامه می یابد. تابع هار^۲ یکی از این توابع پایه می باشد که به صورت زیر بیان می شود

$$\psi(x) = \begin{cases} 1 & 0 \leq x < 0.5 \\ -1 & 0.5 \leq x < 1 \\ 0 & \text{otherwise} \end{cases} \quad (۴)$$

$$\phi(x) = \begin{cases} 1 & 0 \leq x < 1 \\ 0 & \text{otherwise} \end{cases}$$

معادله (۱) را میتوان در قالب شبکه های عصبی نشان داد. این شبکه ها که ویونت نامیده میشوند توسط بخشی و استفانوپولس در سال ۹۳ معرفی گردید [۹][۵]. آنها ویونتها را به صورت یک شبکه عصبی چند دقتی سلسله مراتبی با یک لایه مخفی که توابع فعالیت نرونهای آنها توابع پایه متعامد میباشند در نظر گرفتند. البته می توان به جای ویولتهای متعامد یکه از نمونه های دیگر ویولتها نیز استفاده کرد. برای طراحی کامل شبکه های ویونت باید نوع توابع بکاررفته، تعداد گره ها یعنی تعداد ویولتها و یک

به صورت پویا می باشد. بدین معنی که تعداد توابع پایه از قبل مشخص نمی باشد بلکه همزمان با فرایند تقریب مشخص می شود. در این روشها در هر مرحله با اضافه کردن و یا کم کردن یک ویولت سعی در کاهش خطا میشود. نتایج شبیه سازیها با روشهای پیشنهادی نشان می دهد که با تعداد توابع پایه کمتری میتوان به نتایج مشابه روشهای موجود دست یافت.

قسمتهای بعدی مقاله بصورت زیر سازماندهی شده اند. خلاصه ای از تئوری ویولت در بخش ۲ و مروری بر اتوماتاهای یادگیر در بخش ۳ آمده است. در بخش ۴ الگوریتم های یادگیری ویولت را مرور خواهیم کرد. الگوریتمهای پیشنهادی در قسمت ۵ و نتایج شبیه سازی بر روی چند مساله نمونه در قسمت ۶ آمده است. در انتهای مقاله نتایج را مورد بررسی قرار خواهیم داد.

اصول ویولت^۱

تئوری ویولت ابتدا توسط مورلت و گروسمن در زمینه پردازش سیگنال مطرح شد [۷]. سپس تقریب چند دقتی توسط ملات ارائه شد و دوبیشیز خانواده وسیعی از ویولتهای مختلف را ارائه نمود [۳][۲][۸]. بر اساس این تئوری تمام توابع حقیقی $F(X) \in L^2(\mathbb{R})$ می توانند به صورت یک ترکیب خطی از یک سری توابع پایه متعامد ویولت به فرم $\psi_{m,k}(x)$ بیان شوند.

$$F(x) = \sum_{m=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} d_{m,k} \psi_{m,k}(x) \quad (۱)$$

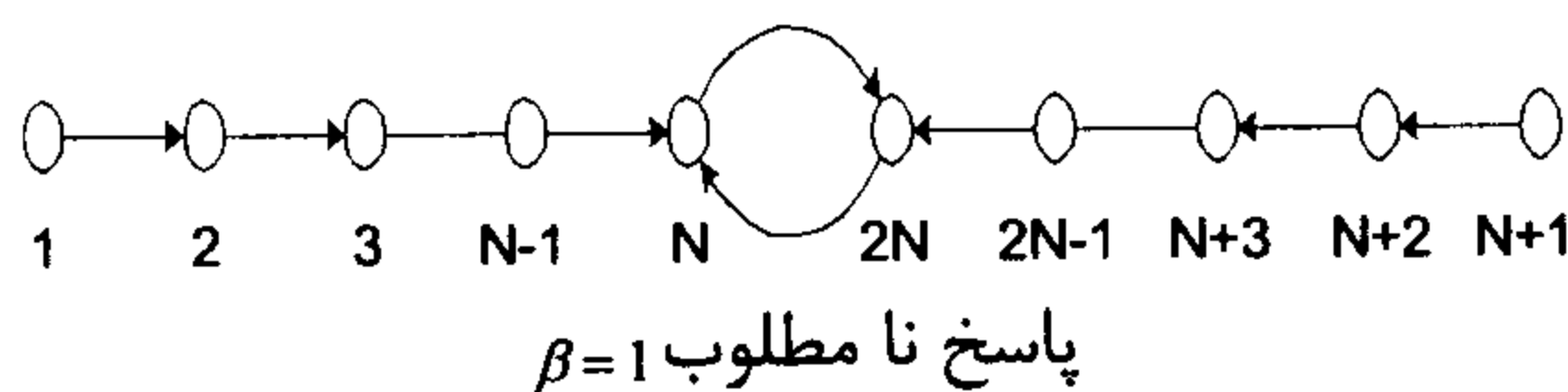
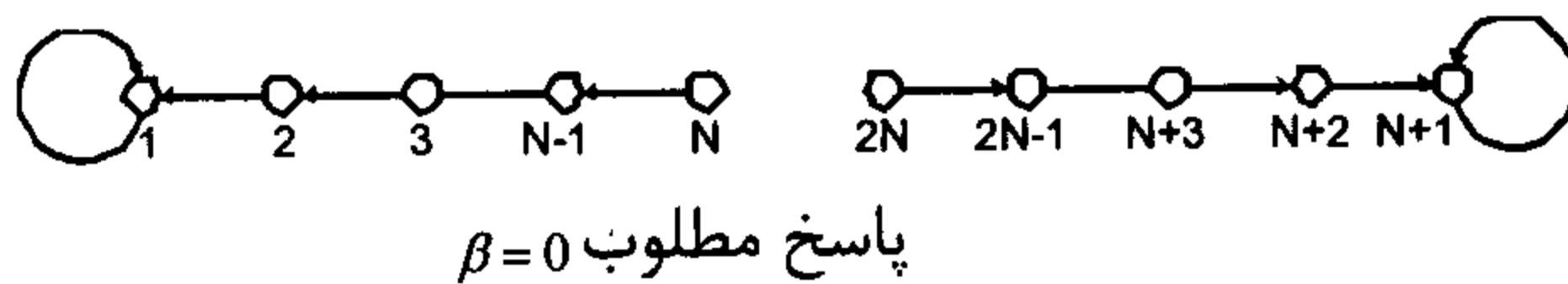
$$m, k \in \mathbb{Z} \quad \psi_{m,k}(x) = 2^{-m/2} \psi(2^{-m}x - k)$$

که در آن 2^{-m} و k به ترتیب ضرائب انتقال و تأخیر و $2^{-m/2}$ ضریب نرمالیز کردن انرژی می باشند. معادله (۱) نشان می دهد که مجموعه ای از ویولتها با تأخیر و انتقال یک تابع ویولت مادر $\psi_{m,k}(x)$ بدست می آید. اکنون اگر از تقریب $m=0$ شروع کنیم معادله (۱) را می توان به صورت زیر نشان داد.

$$F(x) = F_0(x) + \sum_{m=-\infty}^0 \sum_{k=-\infty}^{+\infty} d_{m,k} \psi_{m,k}(x) \quad (۲)$$

$\{\alpha, \beta, F, G, \phi\}$ نشان داده می شود که در آن
 $\alpha \equiv \{\alpha_1, \alpha_2, \dots, \alpha_r\}$ مجموعه عمل های اتوماتا
 $\beta \equiv \{\beta_1, \beta_2, \dots, \beta_m\}$ مجموعه ورودیهای اتوماتا
 $\phi \equiv \{\phi_1, \phi_2, \dots, \phi_s\}$ مجموعه وضعیتهای داخلی اتوماتا ،
 $F: \phi \times \beta \rightarrow \phi$ تابع تولید وضعیت جدید اتوماتا و $G: \phi \rightarrow \alpha$
 تابع خروجی که وضعیت کنونی را به خروجی بعدی
 نگاشت میکند. اکنون به بررسی چند نمونه از اتوماتاهای با
 ساختار ثابت می پردازیم.

اتوماتای $L_{2N,2}$: این اتوماتا تعداد پاداش ها و
 جریمه های در یافت شده برای هر عمل را نگهداری کرده
 و تنها زمانی که تعداد جریمه ها بیشتر از تعداد پاداش ها
 می گردد ، عمل دیگر را انتخاب می کند. نمودار تغییر
 وضعیت این اتوماتا مطابق شکل (۲) می باشد.



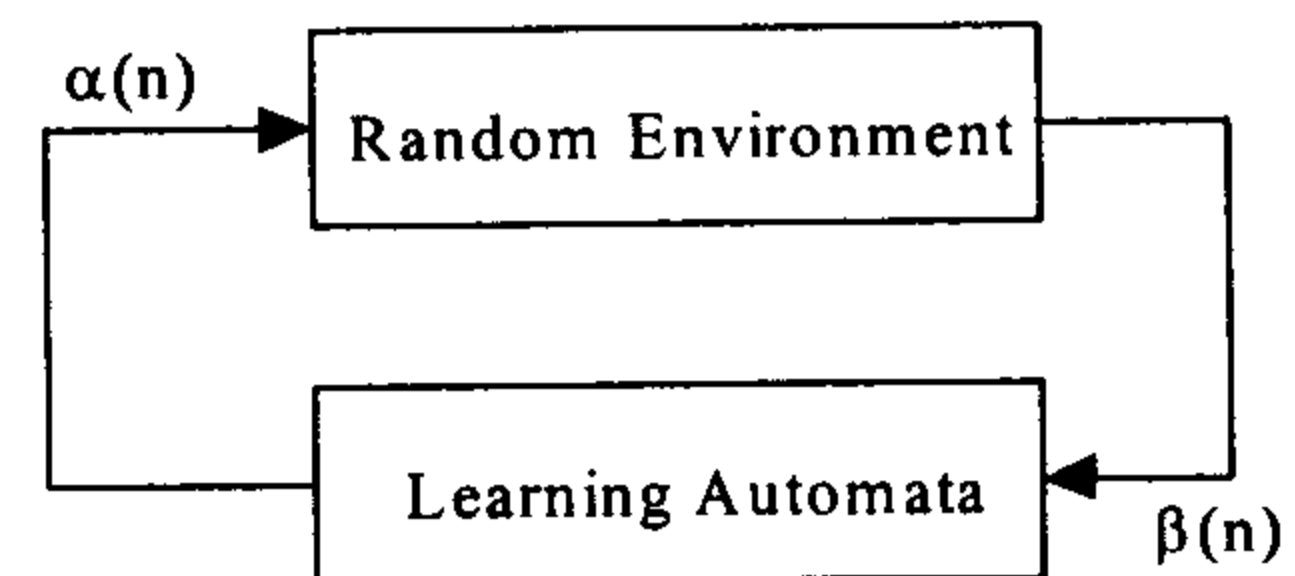
شکل ۲: نمودار تغییر وضعیت اتوماتان $L_{2N,2}$.

اتوماتای یاد گیر با ساختار متغیر^۱: اتوماتای یادگیر با
 ساختار متغیر توسط ۴ تایی $\{\alpha, \beta, p, T\}$ نشان داده
 می شود که در آن $\alpha \equiv \{\alpha_1, \alpha_2, \dots, \alpha_r\}$ مجموعه
 عمل های اتوماتا، $\beta \equiv \{\beta_1, \beta_2, \dots, \beta_m\}$ مجموعه
 ورودیهای اتوماتا، $p \equiv \{p_1, p_2, \dots, p_r\}$ بردار احتمال انتخاب
 هر یک از عملها و $p(n+1) = T[\alpha(n), \beta(n), p(n)]$ الگوریتم
 یادگیری می باشد. در این نوع از اتوماتاها، اگر عمل α_i در
 مرحله n ام انتخاب شود و پاسخ مطلوب از محیط دریافت
 نماید، احتمال $p_i(n)$ افزایش یافته و سایر احتمالات کاهش
 می یابند. و برای پاسخ نامطلوب احتمال $p_i(n)$ کاهش یافته
 و سایر احتمالات افزایش می یابند. در هر حال، تغییرات
 به گونه ای صورت می گیرد تا حاصل جمع $p_i(n)$ ها
 همواره ثابت و مساوی یک باقی بماند. الگوریتم
 زیر یک نمونه از الگوریتمهای یادگیری خطی در اتوماتای

الگوریتم یادگیری برای بدست آوردن ثابتهای a و d را
 مشخص کرد [۹].

اتوماتای یادگیر

اتوماتای یادگیر: اتوماتای یادگیر یک مدل انتزاعی است
 که تعداد معدودی عمل را می تواند انجام دهد. هر عمل
 انتخاب شده توسط محیطی احتمالی ارزیابی شده و پاسخی
 به اتوماتای یادگیر داده می شود. اتوماتای یادگیر از این
 پاسخ استفاده نموده و عمل خود را برای مرحله بعد انتخاب
 می کند [۱۲] [۱۱] [۱۰]. شکل (۱) ارتباط بین اتوماتای
 یادگیر و محیط را نشان می دهد.



شکل ۱: ارتباط بین اتوماتای یادگیر و محیط.

محیط^۳: محیط را می توان توسط سه تایی $E \equiv \{\alpha, \beta, c\}$
 نشان داد که در آن $\alpha \equiv \{\alpha_1, \alpha_2, \dots, \alpha_r\}$ مجموعه ورودیها،
 $\beta \equiv \{\beta_1, \beta_2, \dots, \beta_m\}$ مجموعه خروجیها و $c \equiv \{c_1, c_2, \dots, c_r\}$
 مجموعه احتمالهای جریمه می باشد. هر گاه β مجموعه
 دو عضوی باشد ، محیط از نوع P می باشد. در چنین
 محیطی $\beta_1 = 1$ به عنوان جریمه و $\beta_2 = 0$ به عنوان
 پاداش در نظر گرفته می شود. در محیط از نوع Q ، $\beta(n)$
 می تواند به طور گسسته یک مقدار از مقادیر محدود در
 فاصله $[0,1]$ و در محیط از نوع S ، $\beta(n)$ متغیر تصادفی
 در فاصله $[0,1]$ است.

C_i احتمال اینکه عمل α_i نتیجه نامطلوب^۴ داشته باشد
 می باشد. در محیط ایستا^۵ مقادیر C_i بدون تغییر می مانند
 ، حال آنکه در محیط غیر ایستا^۶ این مقادیر در طی زمان
 تغییر می کنند. اتوماتای یادگیر به دو گروه با ساختار ثابت
 و با ساختار متغیر تقسیم می گردند.

اتوماتای یادگیر با ساختار ثابت^۷:

اتوماتای یادگیر با ساختار ثابت توسط ۵ تایی

$$f(x, n+1) = \begin{cases} a[f(x, n) + \beta H(x, r)] & \text{if } x \in X \\ 0 & \text{otherwise} \end{cases}$$

$$H(x, r) = \lambda \exp\left(-\frac{(x-r)^2}{2\delta^2}\right)$$

که در آن α به گونه ای انتخاب می شود تا رابطه

$$\int_{x \min}^{x \max} f(x, n) dx = 1 \quad \text{تابع } H(x, r) \text{ برقرار شود.}$$

همسایگی متتقارن گوسی شکل و λ و δ پارامترهای تعیین کننده ارتفاع و عرض تابع همسایگی می باشند. برای اطلاعات بیشتر درباره اتوماتاهای یادگیر می توان به [۱۳] [۱۲] [۱۱] مراجعه نمود.

الگوریتمهای یادگیری ویونت

معادله (۲) ساختار کلی شبکه ویونت را بیان کرده و نشان می دهد که چگونه هر ویولت در فرایند تقریب مشارکت دارد. همچنان نشان می دهد که تابع مقیاس کننده فقط در مرحله اول تقریب برای ایجاد F_0 بکار می رود. بعد از ایجاد F_0 تقریب بهتر با اضافه کردن ویولت با همان دقت یعنی $m=0$ انجام می گیرد و F_{-1} را تولید می کند. اضافه کردن ویولتهای با دقت بالاتر مثل $m=-1$ باعث تقریب بهتر یعنی F_{-2} می شود. این فرایند ادامه می یابد تا تابع اصلی ایجاد شود. اکنون به بررسی روشهای محاسبه ضرائب شبکه یعنی $a_{m,k}$ و $d_{m,k}$ می پردازیم.

نزول در امتداد گرادیان^{۱۳}: یادگیری در روش نزول در امتداد گرادیان شبیه یادگیری انتشار خطا به عقب در شبکه های عصبی می باشد. الگوریتم بر اساس

$$\text{معادله } F(x) = \sum_{i=1}^n w_i \psi\left(\frac{x-t_i}{s_i}\right) + \bar{f} \quad \text{که نمایش غیر}$$

متعامد و پیوسته معادله (۳) می باشد عمل می کند که در آن \bar{f} مقدار میانگین تابع $F(x)$ با توجه به نمونه های ورودی خروجی و w, t, s پارامترهای ویولت می باشند. این الگوریتم از روش جستجو در امتداد گرادیان برای حداقل نمودن یک تابع هزینه J استفاده می کند. اگر \ominus بردار متغیرهایی باشد که میخواهیم محاسبه کنیم در مرحله یادگیری گرادیان J نسبت به پارامترهای مختلف محاسبه شده و مقادیر بردار \ominus در جهت عکس گرادیان تابع هزینه تغییر می کند. توضیحات بیشتر در مورد چگونگی

با ساختار ثابت است [۱۴].

الف- پاسخ مطلوب

$$p_i(n+1) = p_i(n) + a[1 - p_i(n)]$$

$$p_j(n+1) = (1-a)p_j(n) \quad j \neq i \quad \forall j$$

ب- پاسخ نامطلوب

$$p_i(n+1) = (1-b)p_i(n)$$

$$p_j(n+1) = \frac{b}{r-1} + (1-b)p_j(n) \quad j \neq i \quad \forall j$$

در روابط فوق، a پارامتر پاداش و b پارامتر جریمه می باشد. با توجه به مقادیر a و b سه حالت زیر را می توان در نظر گرفت. زمانیکه a و b با هم برابر باشند، الگوریتم را L_{RP} می نامیم. زمانیکه b از a خیلی کوچکتر باشد، الگوریتم را L_{REP} می نامیم و زمانیکه b مساوی صفر باشد، الگوریتم را L_{RI} می نامیم.

اتوماتای یادگیر با اقدام پیوسته: این اتوماتا دارای یک اقدام است اما این اقدام در هر مرحله از تکرار می تواند مقادیر مختلف را اختیار کند [۱۵]. (به بیان دیگر این اتوماتا میتواند دارای بینهایت اقدام باشد.) این اقدام به صورت یک متغیر تصادفی پیوسته x ، که مقادیر خود را در فاصله $[x_{\min}, x_{\max}]$ اختیار میکند در نظر گرفته می شود. بنابراین تابع توزیع احتمال گسسته در هر مرحله از تکرار n با تابع چگالی احتمال $f(x, n)$ جایگزین میشود که رابطه زیر در مورد آن برقرار است.

$$\int_{-\infty}^{\infty} f(x, n) dx = \int_x f(x, n) dx = 1 \quad f(x, n) \geq 0 \quad \forall n, \forall x$$

تابع توزیع اولیه (مرحله صفر) را تابع یکنواخت به صورت $f(x, 0) = \frac{1}{x_{\min} - x_{\max}}$ در نظر می گیریم. سپس در هر

مرحله تکرار یک عدد تصادفی $z(n)$ در فاصله $[0, 1]$ ایجاد کرده و اقدام $x(n)$ به صورتی انتخاب می شود تا

$$\int_{x \min}^{x(n)} f(x, n) dx = z(n) \quad \text{بر } x(n)$$

روی محیط اعمال می شود. نتیجه اقدام ممکن است به صورت موفقیت یا شکست یا یک تابع ارزیابی به صورت $\beta(n)$ بیان شود. اکنون با توجه به مقدار $\beta(n)$ چگالی احتمال $f(x, n)$ به صورت زیر تصحیح می شود.

برابر $\bar{F}(X) = F(X) - F_m(X)$ می باشد که می توانیم آن را به فرم ماتریسی به صورت $\bar{F}(X) = AC$ نشان دهیم و در آن A ماتریس مقادیر توابع ویولت و C ماتریس ضرائب می باشد. بوسیله روش حداقل مربعات ضرائب C از رابطه زیر محاسبه می شوند

$$C = ((A^T A)^{-1} A^T) \bar{F} = A^+ \bar{F}$$

اگر ویولتها را متعامد یکه انتخاب کنیم $A^T A$ یک ماتریس واحد خواهد بود و رابطه فوق به شکل نهایی $C = A^T \bar{F}$ تبدیل خواهد شد. برای اطلاعات بیشتر در باره روشهای فوق به [۶] مراجعه شود.

تخمین ضرائب ویولت اتوماتای یادگیر

در این قسمت ۳ الگوریتم مبتنی بر اتوماتاهای یادگیر برای تخمین ضرائب ویولت معرفی می شود. این الگوریتم ها به جای استفاده از یک ساختار ثابت، از یک ساختار پویا بر خوردار است و بهمین دلیل نیازی به دانستن تعداد ویولت ها از قبل نمیباشد. همچنین بر خلاف روش یادگیری L_2 که محاسبات زیاد و بدست آوردن معکوس یک ماتریس با ابعاد بالا برای تعداد زیاد نمونه ها لازمه آن است از محاسبات کمتر و بسیار ساده تری برخوردار است. الگوریتمهای پیشنهادی همانطور که در قسمت شبیه سازی نشان داده شده است نتایج نزدیک به حداقل خطا را تولید میکند. این الگوریتمها برای تقریب از یک مقدار اولیه برای ضریب شروع می کنند و سپس مقدار بهینه برای این ضریب را در اطراف مقدار اولیه جستجو می کنند. مقدار اولیه ضرائب را می توان با استفاده از ضرب داخلی مقادیر تابع و مقادیر ویولتهای پایه در نقاط خاص، و یا با استفاده از روش پیشنهاد شده در مرجع [۴] محاسبه کرد. اکنون به شرح الگوریتمهای پیشنهادی می پردازیم.

الگوریتم ۱: اولین الگوریتم، مبتنی بر اتوماتای یادگیر با ساختار متغیر از نوع L_{RP} میباشد. در این الگوریتم به هر ضریب یک اتوماتای یادگیر تخصیص داده میشود که وظیفه پیدا کردن مقدار بهینه برای آن ضریب را بعهده دارد. هر اتوماتا دارای N اقدام متفاوت می باشد و احتمال انتخاب هر کدام از آنها در ابتدا مساوی و برابر $p_i = 1/N$

عملکرد الگوریتم در مرجع [۴] آمده است.

ضرب داخلی مستقیم^{۱۴}: در این روش در هر مرحله تابع $F(x)$ با استفاده از یک سری توابع متعامد پایه جدید تقریب زده می شود. در اولین مرحله که دقت $m = 0$ نامیده می شود تابع $F(x)$ به صورت $F_0(x) = \sum_k a_{0,k} \phi_{0,k}(x)$ تقریب زده می شود. در مراحل بعدی هر تقریب جدید تقریب قبلی را بهبود می بخشد. به عنوان مثال در یک تقریب دلخواه m تابع تقریب به صورت زیر خواهد بود.

$$F_{m-1}(x) = F_m(x) + \sum_k d_{m,k} \psi_{m,k}(x)$$

در این حالت ضرائب هر کدام از این بسط، تصویر عمود تابع اصلی $F(x)$ بر روی توابع پایه خواهد بود [۶]. بنابراین می توان نوشت:

$$a_{m,k} = \langle \phi_{m,k}(x), F(x) \rangle,$$

$$d_{m,k} = \langle \psi_{m,k}(x), F(x) \rangle$$

در این روش توابع پایه ویولت باید متعامد یکه باشند. همچنین باید تعداد نمونه های کافی از ورودی در دسترس باشد.

روش یادگیری L_2 : در این روش با استفاده از خواص توابع ویولت ضرائب به شکلی محاسبه می شوند تا مجموع مربعات خطای تقریب حداقل شود. فرض می کنیم که بردارهای ورودی و خروجی نمونه های گسسته از یک تابع پیوسته به صورت $X = [x_1, x_2, \dots, x_n]^T$ و $F = [f(x_1), f(x_2), \dots, f(x_n)]^T$ باشند. با توجه به اینکه بردارهای پایه مقیاس کننده و ویولت شبیه هم می باشند برای سادگی آنها را با $\theta(x)$ نشان می دهیم و ضرائب a و d را با c جایگزین می کنیم. آنگاه در هر سطح از تقریب مسئله تقریب را می توان با رابطه
$$\bar{F}(x) = \sum_{k=1}^K c_k \theta_k(x)$$
 بیان کرد. که در این رابطه K

تعداد توابع پایه می باشد. اولین تقریب یعنی F_0 با استفاده از تابع مقیاس کننده بدست می آید و تقریب جدید به کمک تقریب قبلی به شکل زیر محاسبه میشود.

$$F_{m-1}(x) = F_m + \sum_k c_{m,k} \psi_{m,k} \quad m, k \in z$$

که F_m تقریب قبلی و F_{m-1} تقریب جدید است. پس می توانیم بگوییم میزان تغییرات در هر تقریب

۳- تابع توزیع احتمال یکنواخت $f_i(k_i, n)$ برای k_i در تکرار n را تعریف کن

۴- تابع توزیع احتمال $f_i(k_i, n)$ برای مقادیر $i=1..m$ در محدوده تعیین شده مقدار اولیه دهی کن

۵- مراحل زیر را تکرار کن تا اینکه خطا به حد معینی برسد.

یک عدد تصادفی بین صفر و یک $Z_i(n)$ را با استفاده

از یک تولید کننده اعداد تصادفی انتخاب کن

: $k_i \in A(n)$ را به گونه ای انتخاب کن تا

$$\int_{x_i \min}^{k_i(n)} f(x, n) dx = Z_i(n) \text{ شود}$$

: اقدام را بر روی سیستم اعمال کن

: $\beta(n)$ را محاسبه کن

: تابع توزیع احتمال $f_i(k_i, n)$ را به هنگام کن

نتایج شبیه سازی

در این قسمت عملکرد الگوریتمهای پیشنهادی برای چند مثال نمونه مورد بررسی و با الگوریتمهای موجود مقایسه میگردد.

مثال اول: در این مثال یک سیستم دیگ واکنشی پیوسته (CSTR)^{۱۵} که در آن یک واکنش شیمیایی گرمزای برگشت پذیر در دمای ثابت انجام میگیرد را مدل سازی می کنیم. واکنش انجام گرفته نسبت به غلظت مواد ورودی از درجه یک بوده که توسط معادله زیر بیان میشود.

$$\text{Rate} = -k_0 C V e^{-\frac{E}{RT}}$$

که در آن V حجم دیگ، k_0 ضریب واکنش، C غلظت مواد ورودی، E/R مقدار ثابت، T دمای واکنش و Rate میزان واکنش انجام شده می باشد. در این مثال غلظت مواد ورودی را ثابت در نظر گرفته و دما را که میزان جریان آب است به عنوان ورودی فرض می کنیم. ورودی از یکصد نقطه که به صورت یکنواخت در فاصله $[0, 150]$ ایجاد شده تشکیل شده است. نیمی از ورودی ها که به صورت یک در میان انتخاب شده اند برای آموزش و نیمی دیگر برای آزمایش شبکه بکار برده شده اند. ورودی و خروجی اولیه در شکل (۳) نمایش داده شده است.

می باشد. هر اقدام مقادیر خود را در فاصله $\{k_{\min}, k_{\max}\}$ اختیار میکند. بر اساس اقدام انتخاب شده یک مقدار پابت در ضریب ویولت ضرب می شود. اگر مقدار اقدام انتخاب شده توسط اتوماتا در یک مرحله k_c باشد ضریب ویولت استفاده شده توسط الگوریتم برای مرحله بعد k_c برابر ضریب در مرحله فعلی خواهد بود. در هر مرحله از الگوریتم هر یک از اتوماتاها یکی از اقدامهای خود را انتخاب میکنند و با توجه به مقدار انتخاب شده توسط اتوماتا مقدار ضریب ویولت مورد نظر برای مرحله بعد محاسبه میشود. با استفاده از ضرایب تعیین شده توسط اتوماتاها مقدار جدید تابع $F_1(x) = \sum_j c_{1,j} \theta_{1,j}(x)$ و خطای تفریب به صورت $e = F(x) - F_1(x)$ محاسبه می گردد. اگر خطای بدست آمده از خطای مرحله قبل کمتر باشد اقدام انتخاب شده پاداش می گیرد و در غیر اینصورت جریمه می شود. بر اساس نتیجه اقدام هر اتوماتا، بردار احتمالات انتخابهای آن اتوماتا تصحیح می شود.

الگوریتم ۲: این الگوریتم، مبتنی بر اتوماتای یادگیر با ساختار ثابت از نوع $L_{2N,2}$ میباشد. در این الگوریتم نیز همانند الگوریتم ۱ برای هر ضریب ویولت یک اتوماتا تخصیص داده میشود ولی مقادیر اقدامهای هر اتوماتا مقادیر متفاوت افزایش یا کاهش به ضریب ویولت در نظر گرفته میشود. هر اتوماتا دارای k اقدام هر یک با عمق حافظه N می باشد. بر اساس اقدام انتخاب شده یک مقدار معین به ضریب ویولت افزوده و یا کاسته می شود. در صورتی که این افزایش یا کاهش باعث بهبود خطا شود این عمل تکرار می شود و در غیر اینصورت بر اساس تعداد دفعات پاسخ ناموفق اقدام تغییر کرده و میزان افزودن و یا کاستن ضریب تغییر میکند.

الگوریتم ۳: این الگوریتم، مبتنی بر اتوماتای یادگیر با اقدام پیوسته میباشد. در این الگوریتم به ازای هر ضریب یک اتوماتا با ساختار پیوسته در نظر گرفته میشود. الگوریتم یادگیری به صورت زیر بیان می شود.

۱- دفعات تکرار (n) را برابر صفر قرار بده

۲- مجموعه اقدامها را برای m پارامتر ویولت به صورت

$$A(n) = \{k_1(n), k_2(n), \dots, k_m(n)\}$$

جدول ۲: مقادیر متوسط خطا برای الگوریتم ۲.

الگوریتم	دقت تقریب	-۲	-۱	۰
الگوریتم ۲ با عمق ۱	۰.۰۳۵۰	۰.۰۵۴۵۸	۰.۰۹۷۳	
الگوریتم ۲ با عمق ۲	۰.۰۱۲۱	۰.۰۲۴۵۱	۰.۰۸۵۲	
الگوریتم ۲ با عمق ۳	۰.۰۲۴۲	۰.۰۳۶۵	۰.۰۹۴۸۵	
الگوریتم ۲ با عمق ۴	۰.۰۲۶۹	۰.۰۳۷۸	۰.۰۹۵۸۶	
روش یادگیری L2	۰.۰۰۰۰۸	۰.۰۰۰۸۵۹	۰.۰۰۷۳۶	

شبیه سازی برای الگوریتم ۲: در این قسمت اقدامها را برابر مقادیر ثابت افزایش یا کاهش به ضریب در نظر گرفتیم. هر اتوماتا دارای هشت اقدام متفاوت است که این اقدامها عبارتند از: $[-0.5, 0.3, 0.12, -0.05, 0.012]$. میانگین مربع خطا و نتایج شبیه سازی شده به ازای مقادیر مختلف عمق در جدول (۲) نشان داده شده است. نتایج جدول (۲) نشان می دهد که الگوریتم ۲ که در آن از اتوماتای با ساختار ثابت استفاده شده است نتایج بهتری را تولید می نماید. این موضوع بدلیل این است که این اتوماتای از تجربیات گذشته خود برای تصمیم گیری های آینده استفاده می کند. کمترین خطا برای عمق ۲ بدست آمده است. افزایش عمق به بیش از ۲ تاثیر چندانی روی دقت نداشته است. نتایج شبیه سازی با روش یادگیری L2 نیز در جدول (۲) آمده است.

مثال دوم: در این مثال یک تابع تفاضلی با ورودی سینوسی که در زیر آمده است تقریب زده میشود.

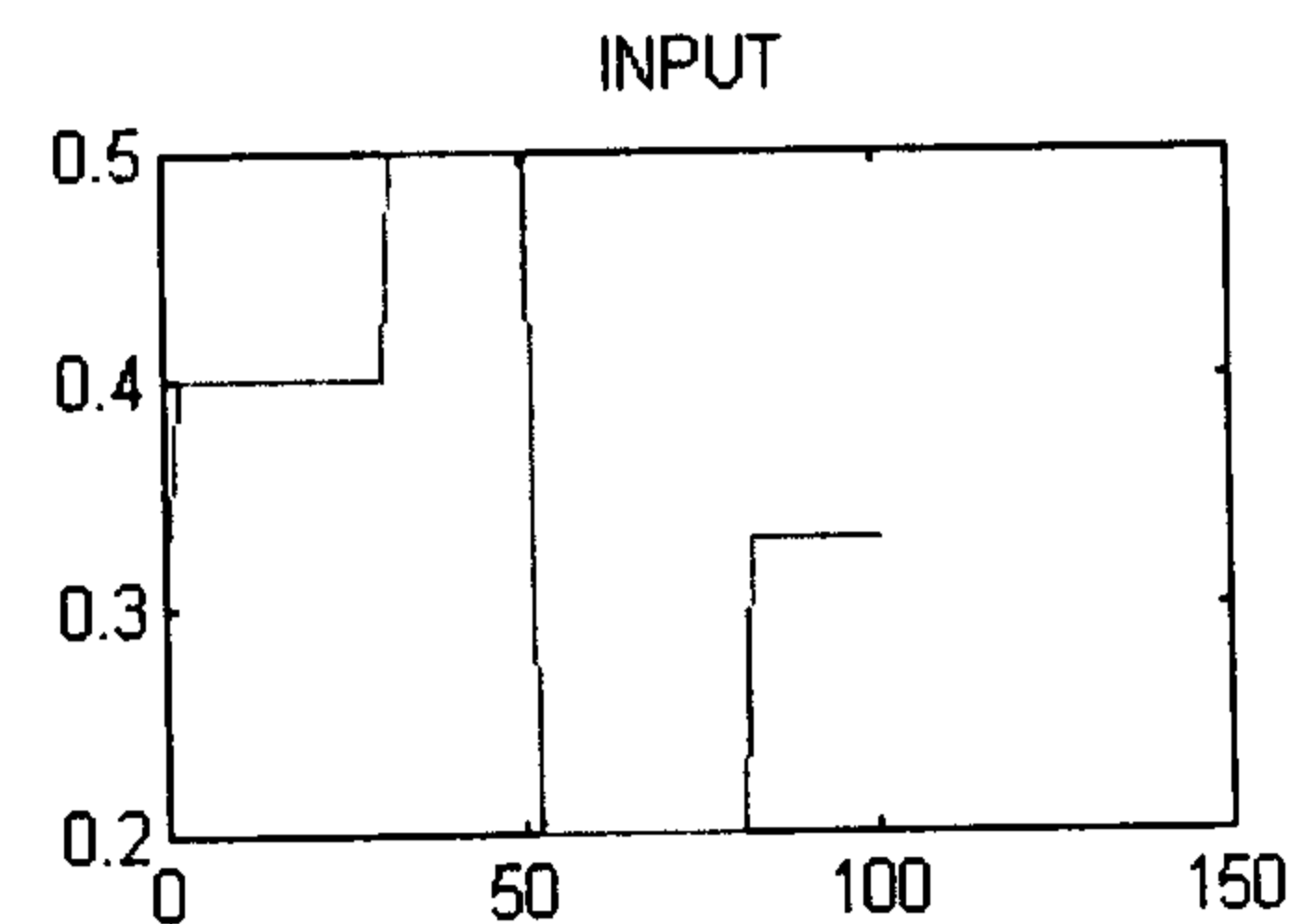
$$y_p(k+1) = f[y_p(k), y_p(k-1)] + u(k)$$

$$f[y_p(k), y_p(k-1)] = (y_p(k)y_p(k-1)[y_p(k)+25]) / (1 + y_p^2(k) + y_p^2(k-1))$$

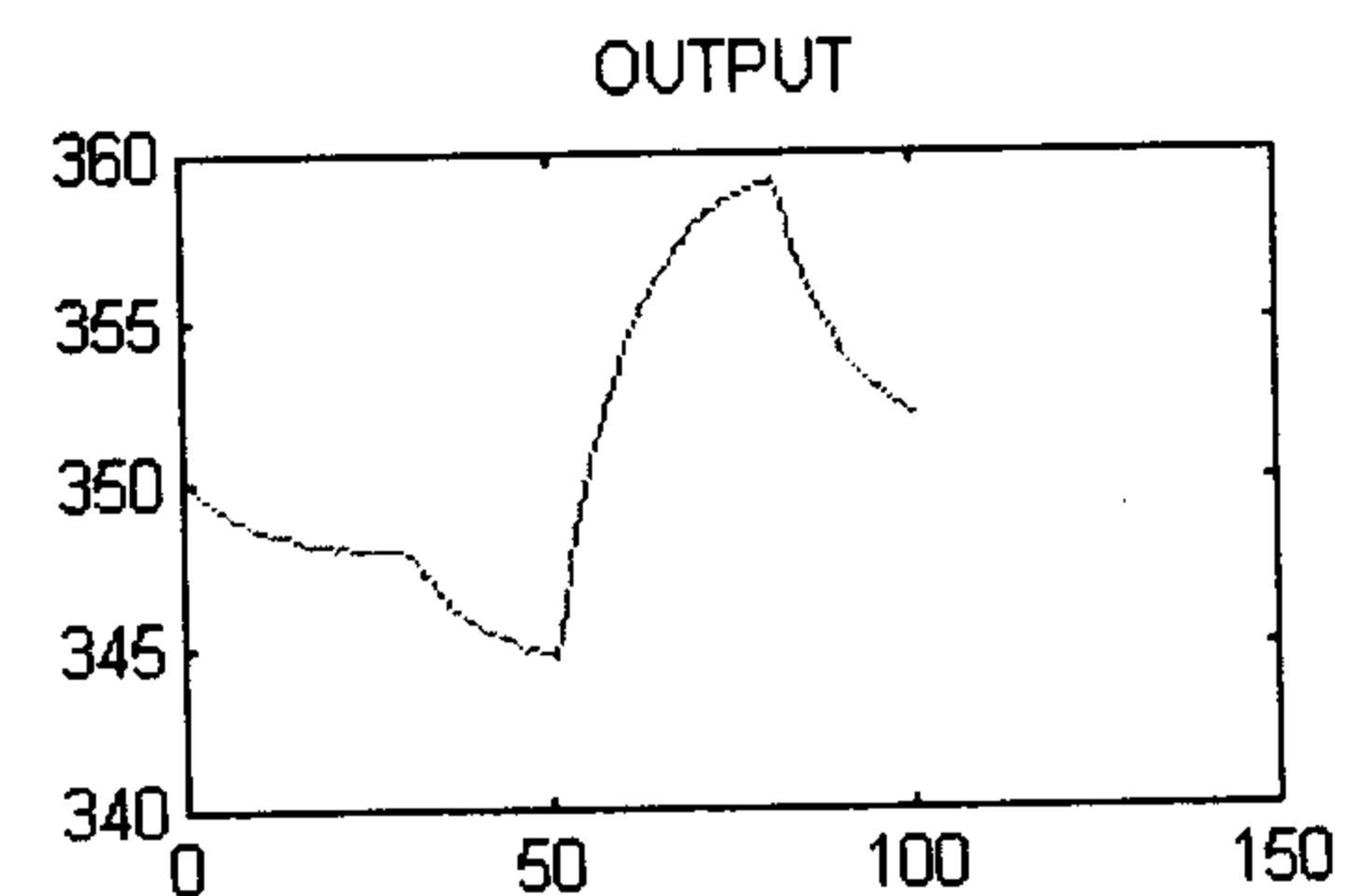
$$u(k) = \sin(2\pi k / 25)$$

ورودی از یکصد نقطه که به صورت یکنواخت در فاصله $[0, 100]$ ایجاد شده تشکیل شده است. ورودی و خروجی اولیه در شکل (۵) نمایش داده شده است. شبیه سازی هم بوسیله اتوماتای یادگیر با ساختار متغیر و هم بوسیله اتوماتای یادگیر با ساختار ثابت با شرایط مثال قبل انجام شده است. میانگین مربع خطای متوسط پس از ۱۵ بار اجرا و به ازای مقادیر مختلف a و b در جدول (۳) و میانگین مربع خطا به ازای مقادیر مختلف عمق در جدول

شبیه سازی برای الگوریتم ۱: اتوماتا دارای یکصد اقدام متفاوت در فاصله $[0, 20]$ با طول گام 0.2 می باشد احتمال انتخاب هر کدام از این اقدامها در ابتدا مساوی و برابر 0.1 می باشد. مقدار خطای متوسط پس از ۱۵ بار اجرا و به ازای مقادیر مختلف a و b در جدول (۱) و یک نمونه از تقریب در شکل (۴) نشان داده شده است. با توجه به مقادیر بدست آمده در جدول (۱) کمترین میانگین مربع خطا برای اتوماتای متغیر با مقادیر $a = 0.5, b = 0.5$ حاصل شده است.



(a)



(b)

شکل ۳:

(a) ورودی CSTR (b) خروجی CSTR.

جدول ۱: مقادیر متوسط خطای الگوریتم ۱ به ازای مقادیر مختلف a و b .

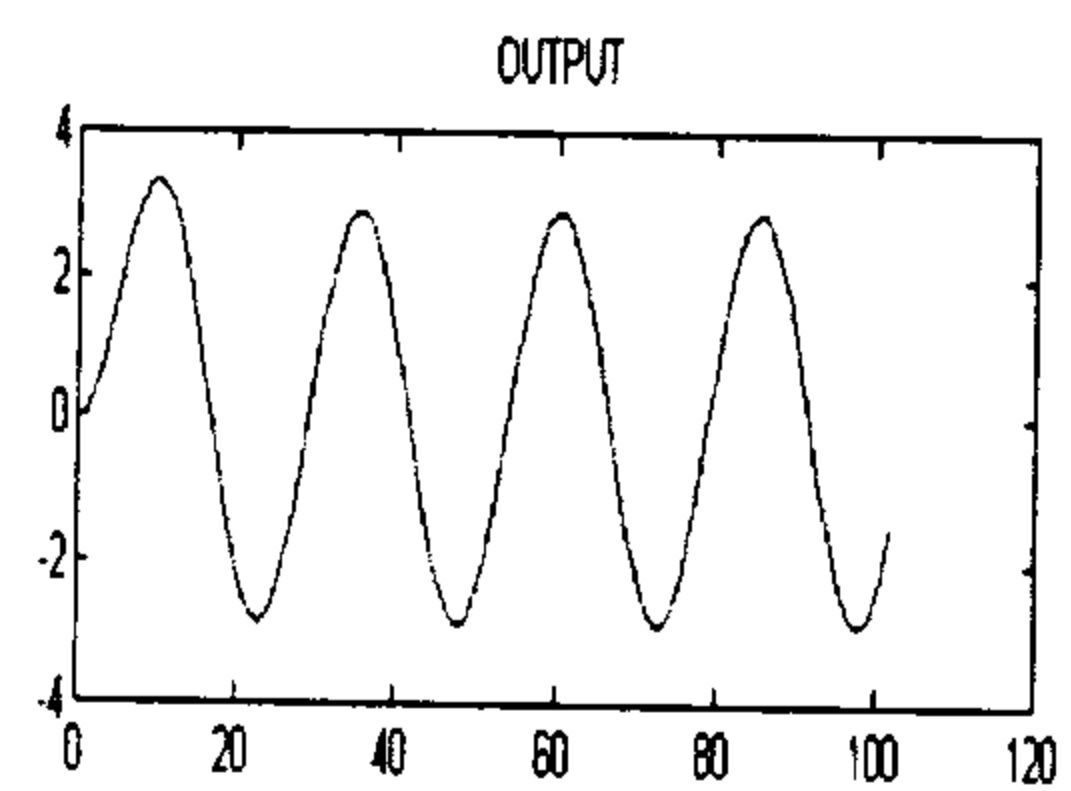
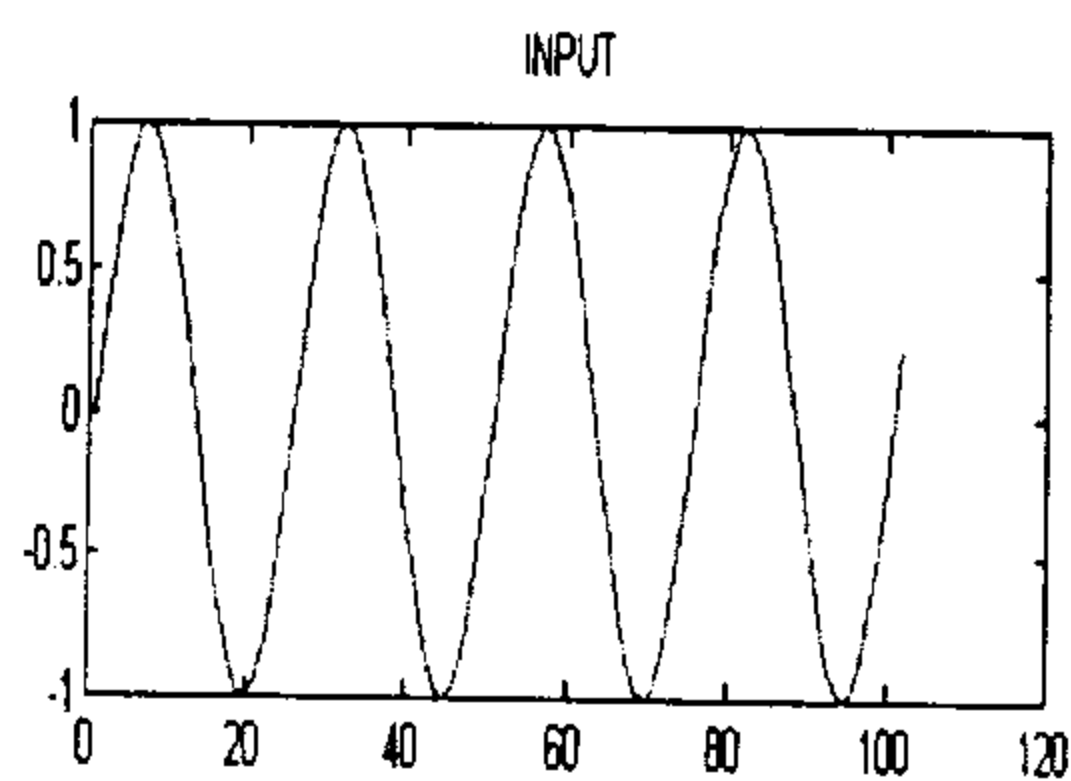
دقت تقریب پارامتر	-۲	-۱	۰
$a = 0.9, b = 0.1$	۰.۰۷۷۱	۰.۰۸۴۵۱	۰.۰۹۸۱۱
$a = 0.8, b = 0$	۰.۰۳۷۴۴	۰.۰۶۲۵۵	۰.۱۴۷۱
$a = 0.7, b = 0.9$	۰.۰۳۱۰۱	۰.۰۷۴۱	۰.۱۳۳۵
$a = 0.7, b = 0.1$	۰.۰۴۱۳	۰.۰۷۱۲	۰.۱۸۵
$a = 0.5, b = 0.5$	۰.۰۱۹۵	۰.۰۵۲۵	۰.۱۰۶۵

جدول ۳: مقادیر متوسط خطای اتوماتای متغیر به ازای مقدار ثابت a و مقادیر مختلف b .

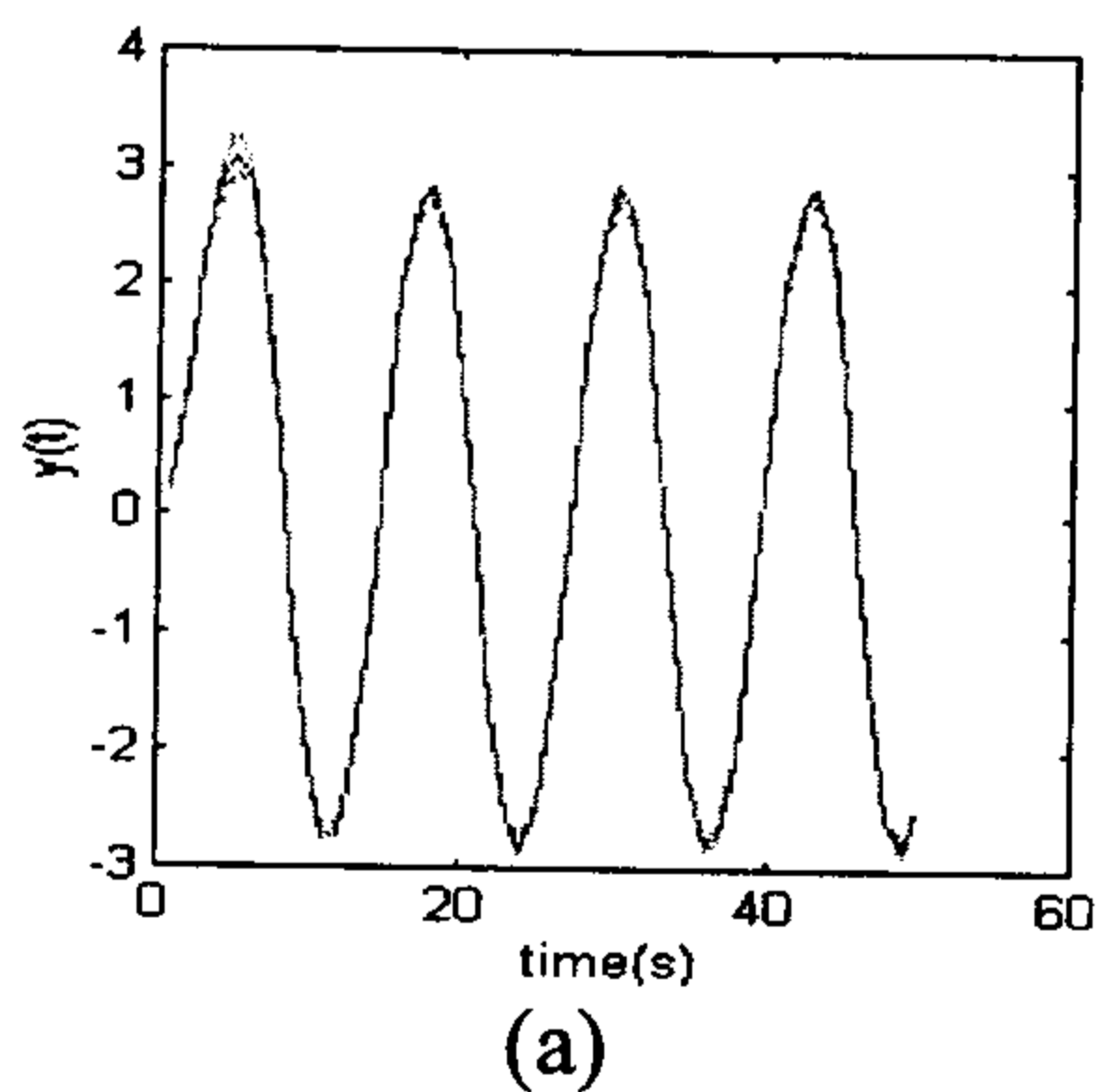
پارامتر	دقت تقریب	-۲	-۱	۰
$a = 0.8, b = 0.1$	۰.۰۶۵۷	۰.۰۸۳۲	۰.۰۹۲۱	
$a = 0.8, b = 0.2$	۰.۰۸۷۴	۰.۰۹۸۴۵	۰.۲۱۴۳	
$a = 0.8, b = 0.3$	۰.۰۵۳۴	۰.۰۶۷۳۲	۰.۱۸۵۴	
$a = 0.8, b = 0.4$	۰.۰۴۹۸	۰.۰۵۴۷۱	۰.۱۷۴۵	
$a = 0.8, b = 0.6$	۰.۰۴۷۲	۰.۰۵۶۳	۰.۶۴۸۵	
$a = 0.8, b = 0.8$	۰.۰۲۱۶	۰.۰۳۲۴	۰.۱۴۳۵	

جدول ۴: مقادیر متوسط خطا برای الگوریتمهای ۲ و ۳.

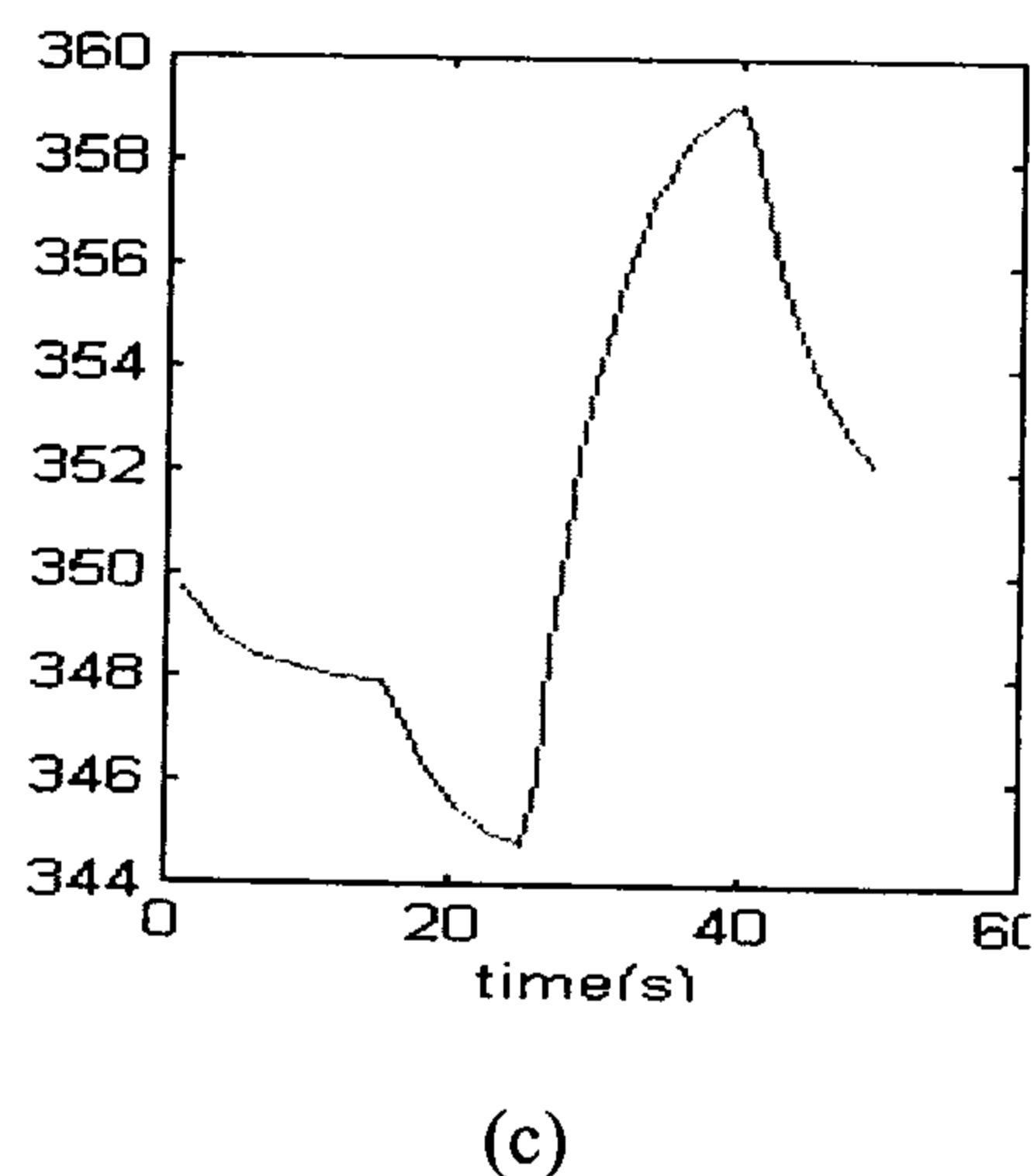
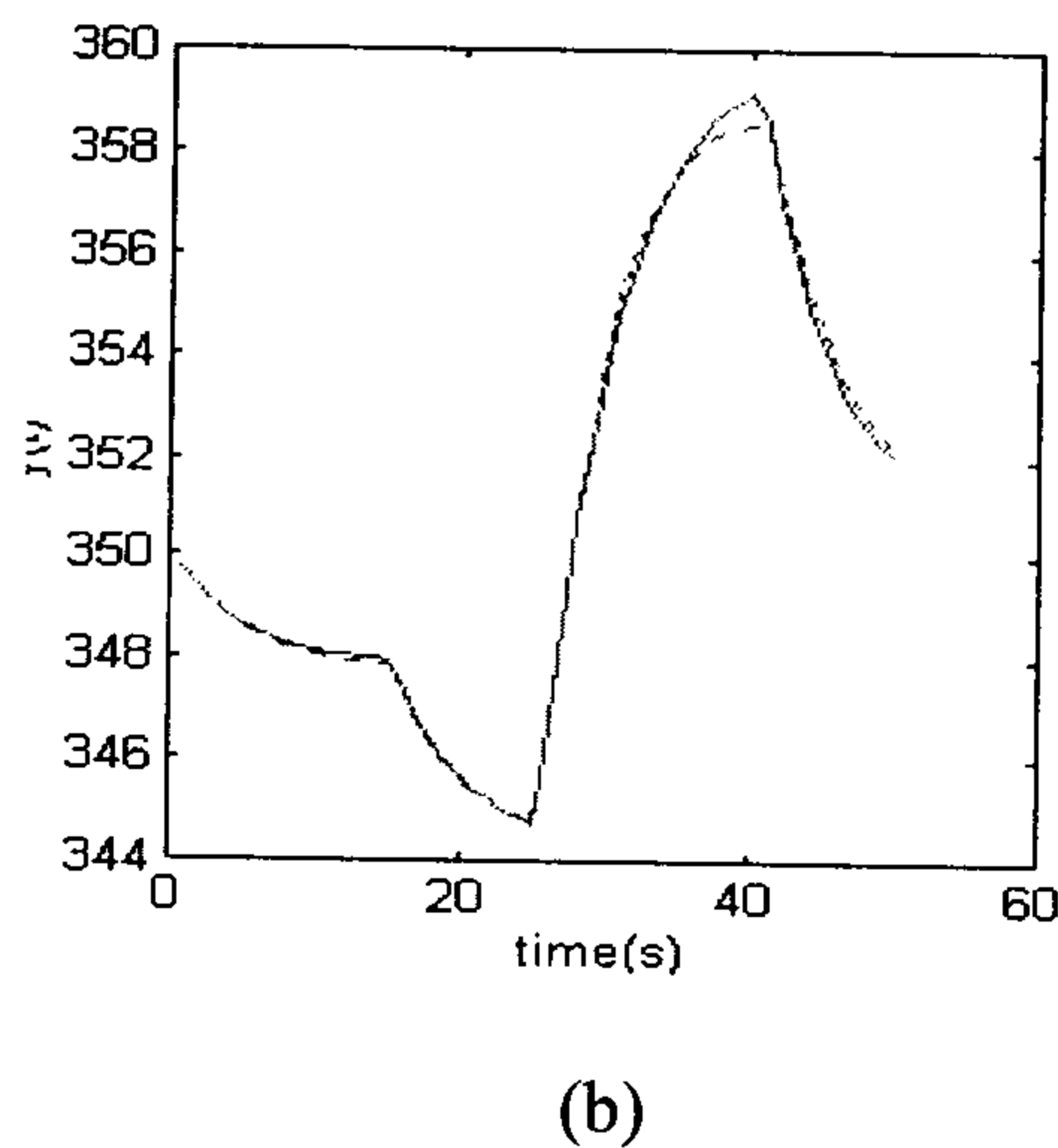
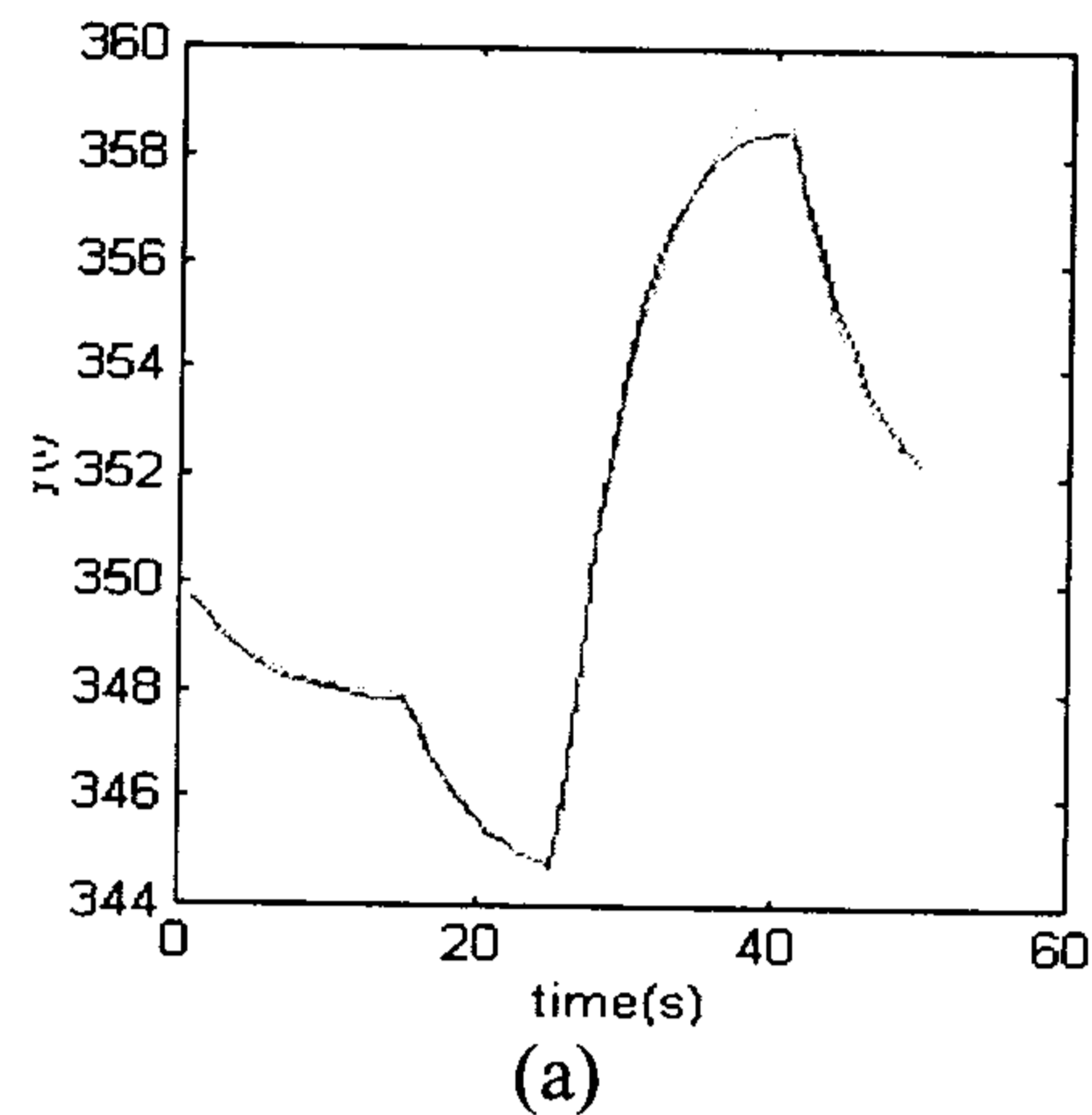
الگوریتم	دقت تقریب	-۲	-۱	۰
الگوریتم ۳	۰.۰۹۸۴۱	۰.۱۶۴۴	۰.۱۹۲۳	
الگوریتم ۲ با عمق ۱	۰.۰۲۷۸۳	۰.۰۶۷۳۸	۰.۱۰۲۳	
الگوریتم ۲ با عمق ۲	۰.۰۱۴۷	۰.۰۲۶۶۹	۰.۰۶۷۸	
الگوریتم ۲ با عمق ۳	۰.۰۲۹۸۸	۰.۰۴۷۳	۰.۰۸۷۳۲	
الگوریتم ۲ با عمق ۴	۰.۰۳۲۹	۰.۰۵۶۴۳	۰.۱۰۰۳۴	
L2 روش یادگیری	۰.۰۰۰۱۱	۰.۰۰۲۳۴	۰.۰۰۹۸۴	



شکل ۵: ورودی خروجی تابع تفاضلی.



(۴) و یک نمونه از تقریب در شکل (۶) نشان داده شده است.



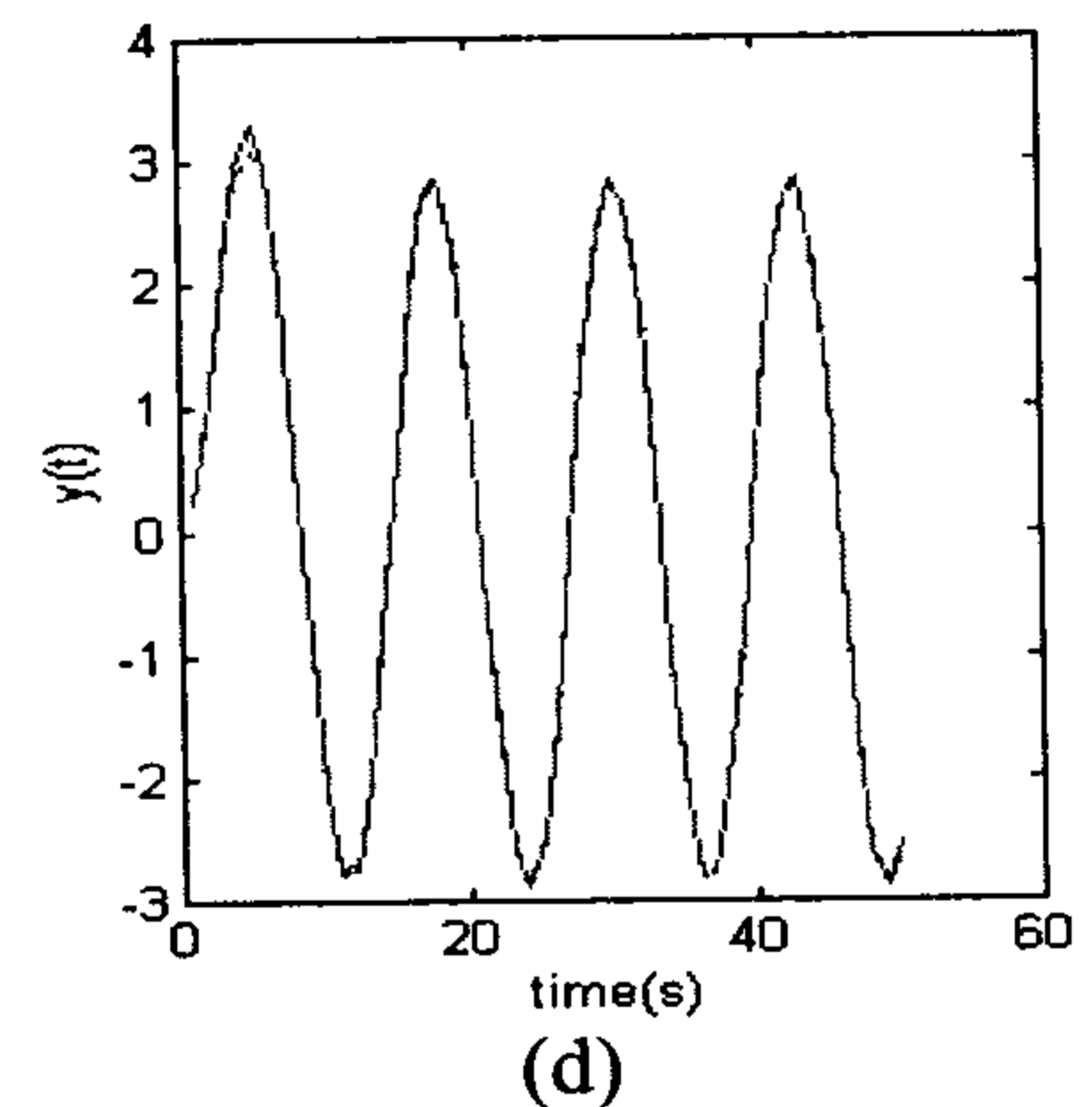
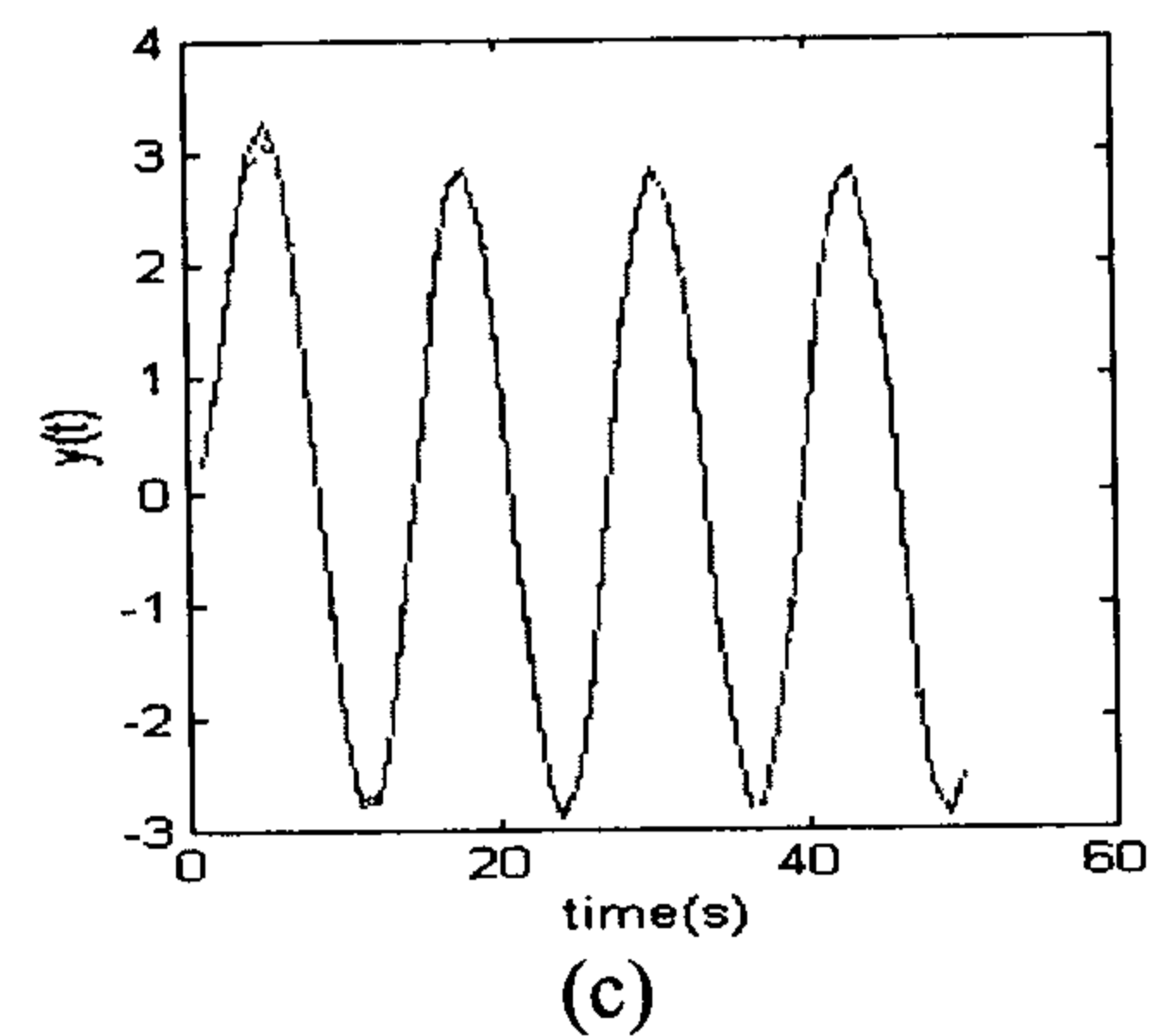
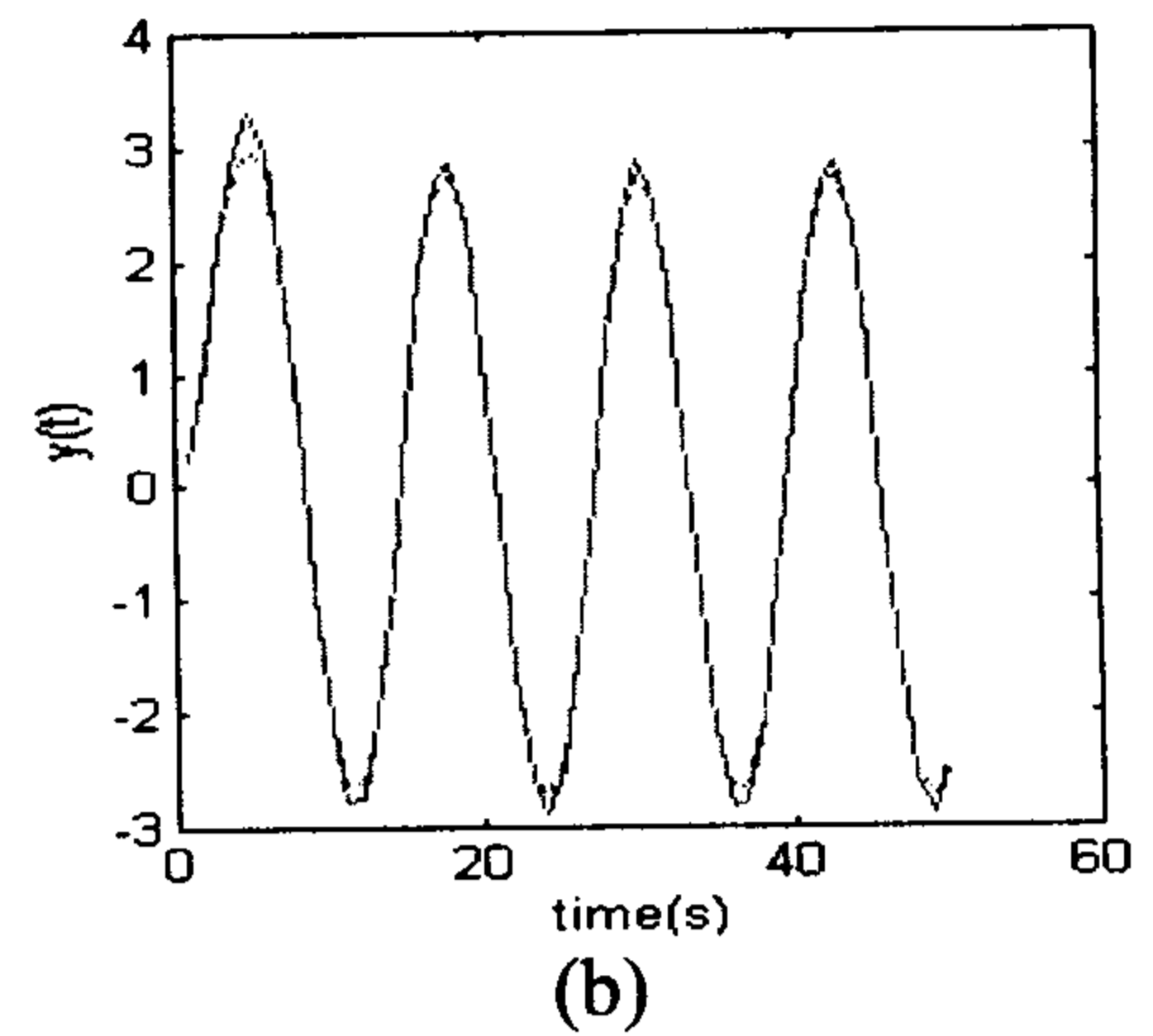
شکل ۴:

(a) : تقریب با دقت ۰ (b) : تقریب با دقت -۱

(c) : تقریب با دقت -۲

خط پیوسته تابع اصلی خط (- -) تابع تقریب

فاصله $[-10, 10]$ نمونه برداری شده استفاده شده است. ۲۰۰ نمونه برای محاسبه تقریب و ۲۰۰ نمونه برای آزمون تقریب تابع به کار برده شده است. شده برای هر سه الگوریتم تابع ویولت به کار برده شده تابع گوسی $\psi(x) = \text{Exp}(-\frac{1}{2}x^2)$ می باشد. در اتوماتای پیوسته برای هر ضریب یک اتوماتا با تابع توزیع احتمال پیوسته به کار برده شده است. چگونگی عملکرد این اتوماتا به این ترتیب است که تابع توزیع احتمال در همسایگی بهترین اقدام به سمت ماکزیمم خود میل می کند. همانگونه که در شکل (۷) نشان داده شده است بهترین مقادیر بدست آمده برای دو ضریب اول به سمت -۰.۹۵ و ۰.۰۱ میل می کند که مقدار واقعی آنها -۰.۷۰۶۳ و -۰.۰۱۹۲ می باشد. به منظور مقایسه چهار پارامتر اول بدست آمده با الگوریتمهای مختلف در جدول (۵) و یک نمونه از تقریب در شکل (۸) و مقادیر متوسط خطای تقریب در جدول (۶) نشان داده شده است.



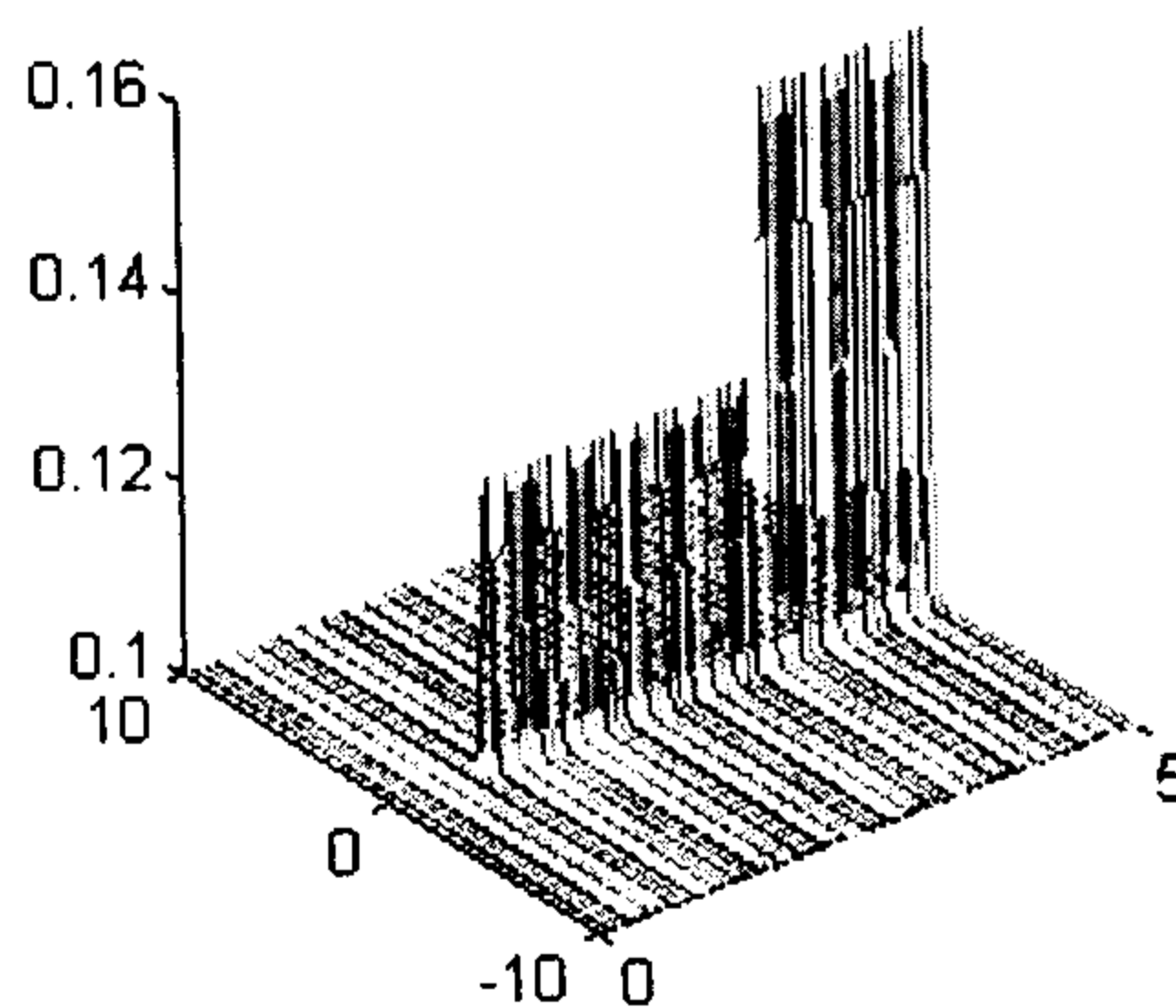
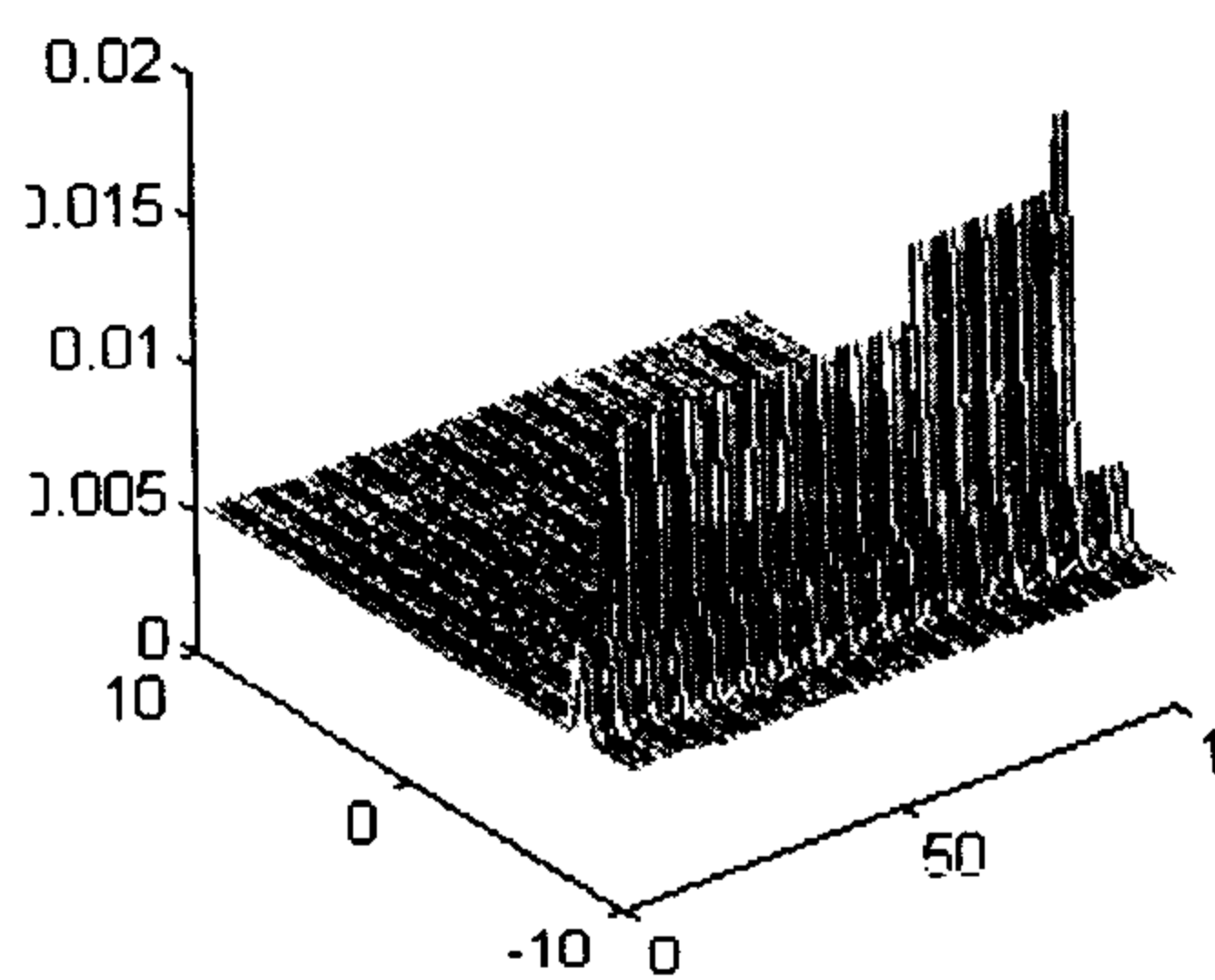
شکل ۶:

(a): تقریب با دقت ۰ (b): تقریب با دقت ۱-
(c): تقریب با دقت ۲- (d): تقریب با دقت ۳-
خط پیوسته تابع اصلی خط (- -) تابع تقریب.

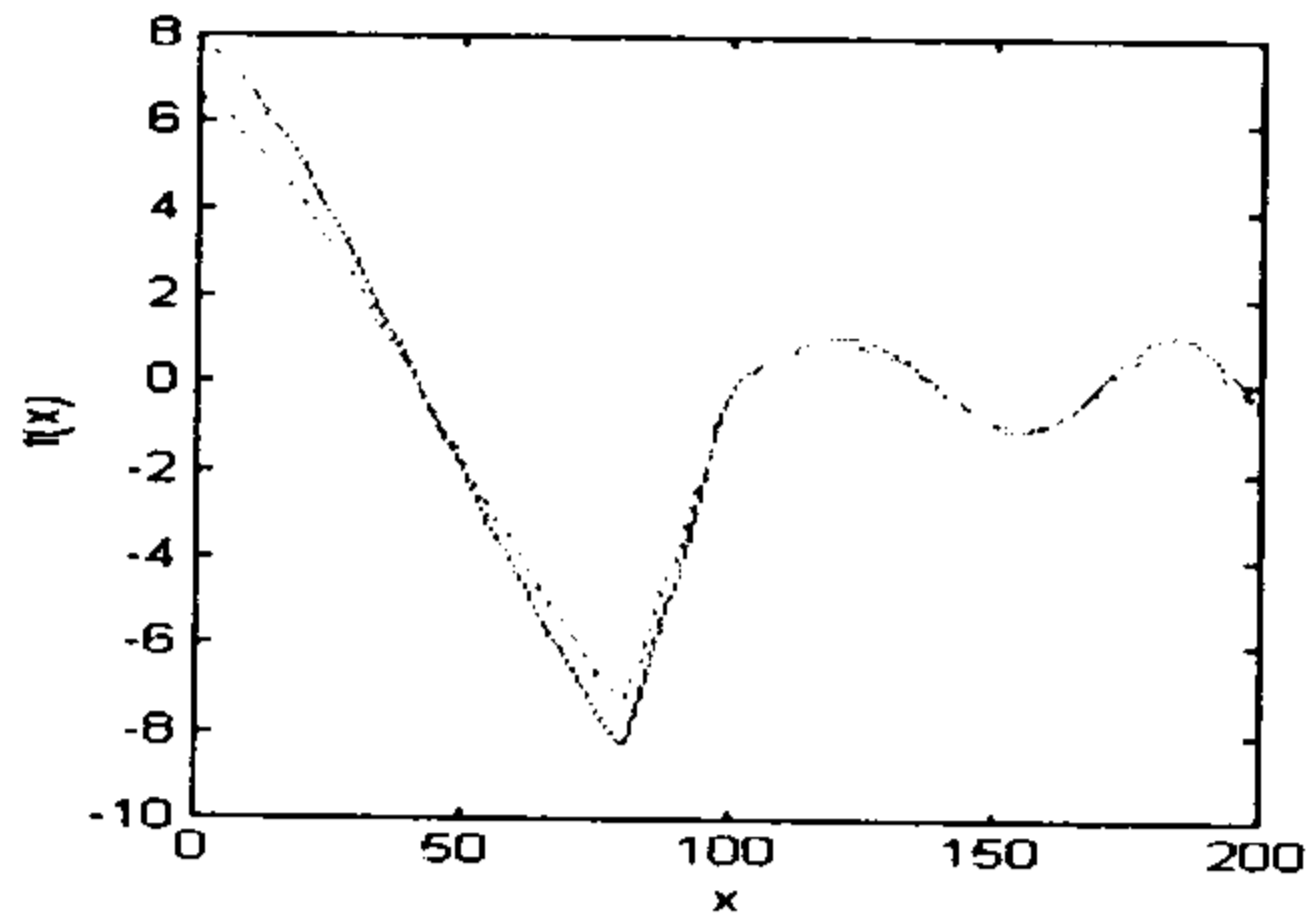
مثال سوم: در این مثال تابع زیر با استفاده از سه الگوریتم ۱ و ۳ و الگوریتم آموزش نزول در امتداد گرادیان تقریب زده شده است.

$$F(x) = \begin{cases} -2.186x - 12.864 & -10 \leq x < -2 \\ 4.246x & -2 \leq x < 0 \\ \sin((0.03x + 0.7)x) & 0 \leq x < 10 \end{cases}$$

برای تقریب تابع از ۴۰۰ نمونه که به صورت یکنواخت در

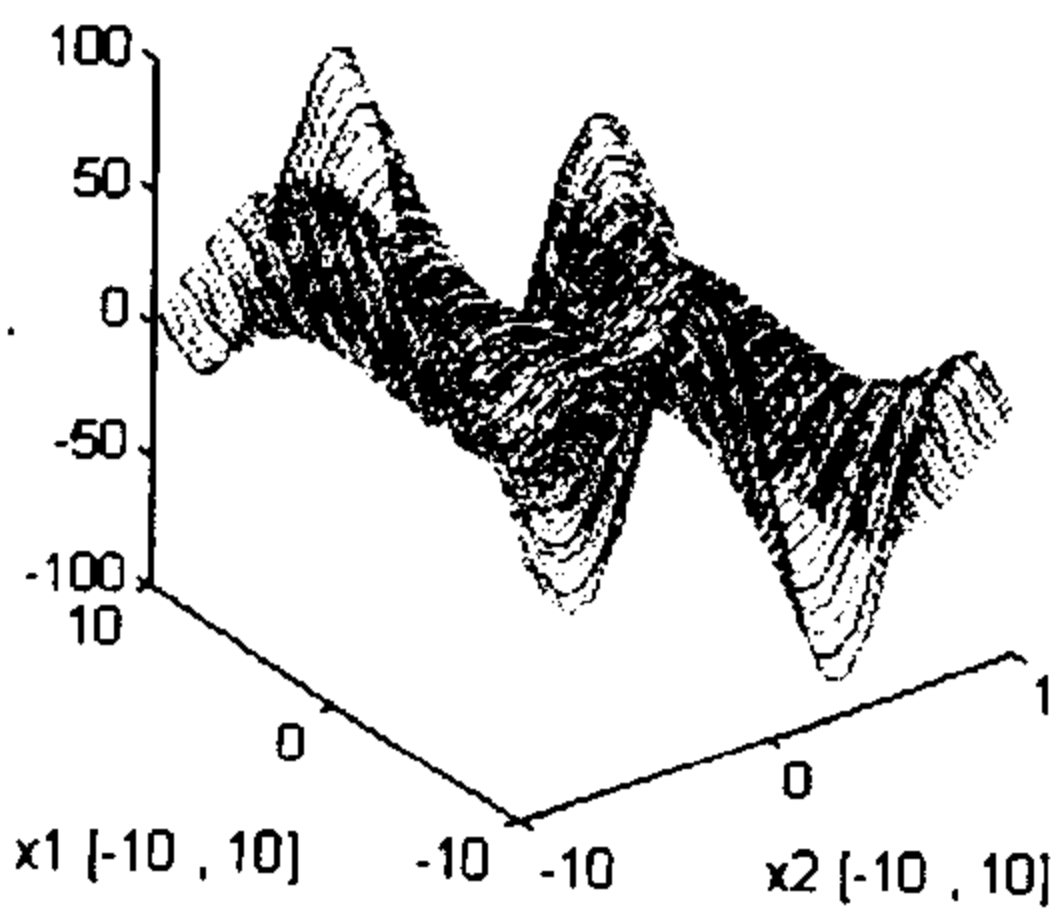


شکل ۷: تابع توزیع احتمال بر اساس تعداد دفعات تکرار.

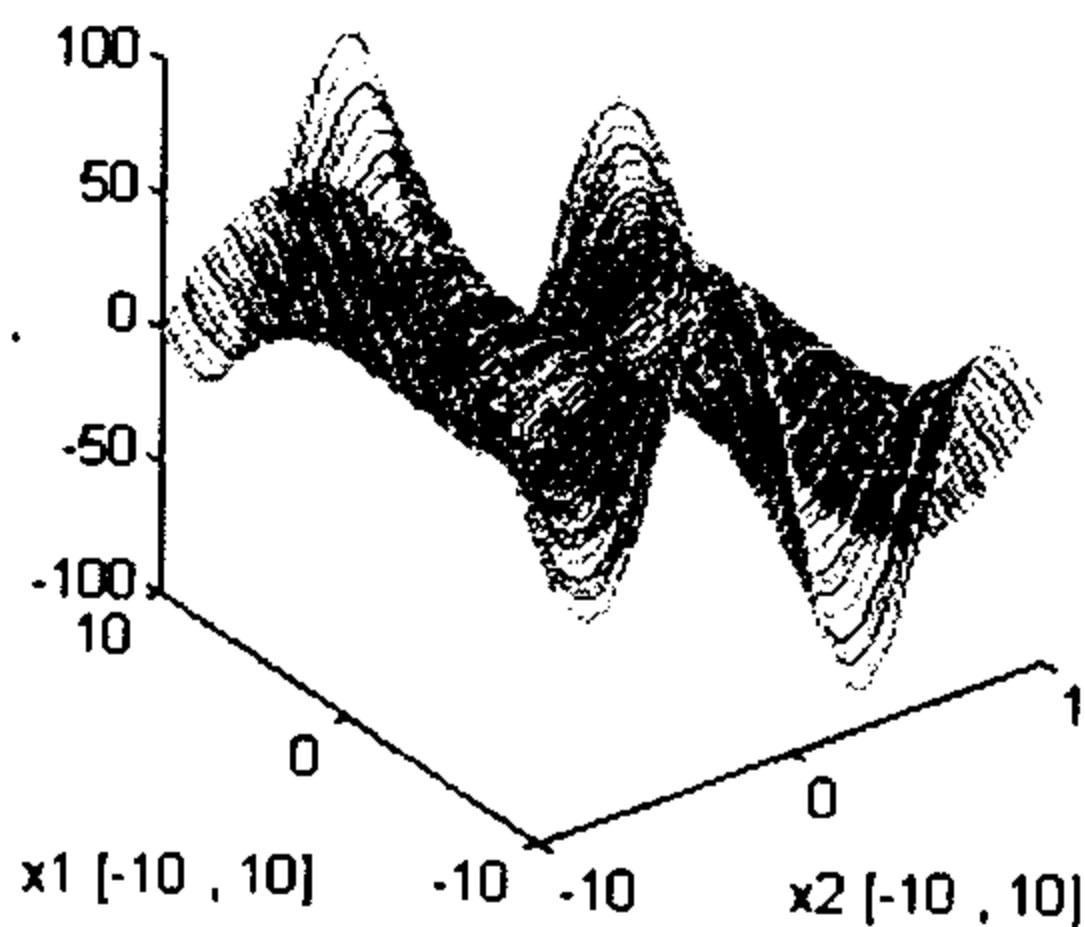


(c)

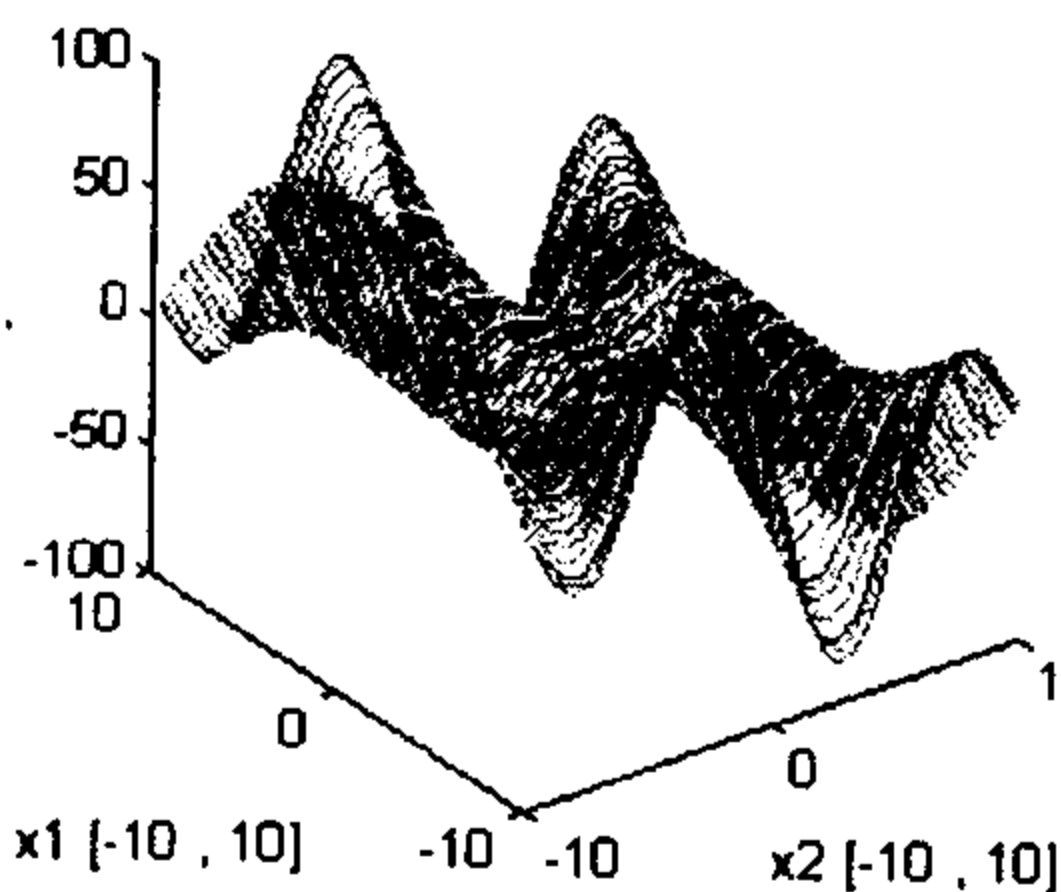
شکل ۸: (a): تقریب تابع با استفاده الگوریتم ۳ (b): تقریب تابع توسط شبکه عصبی (c): تقریب تابع با استفاده الگوریتم ۱ خط پیوسته تابع اصلی خط (- -) تابع تقریب.



(a)



(b)



(c)

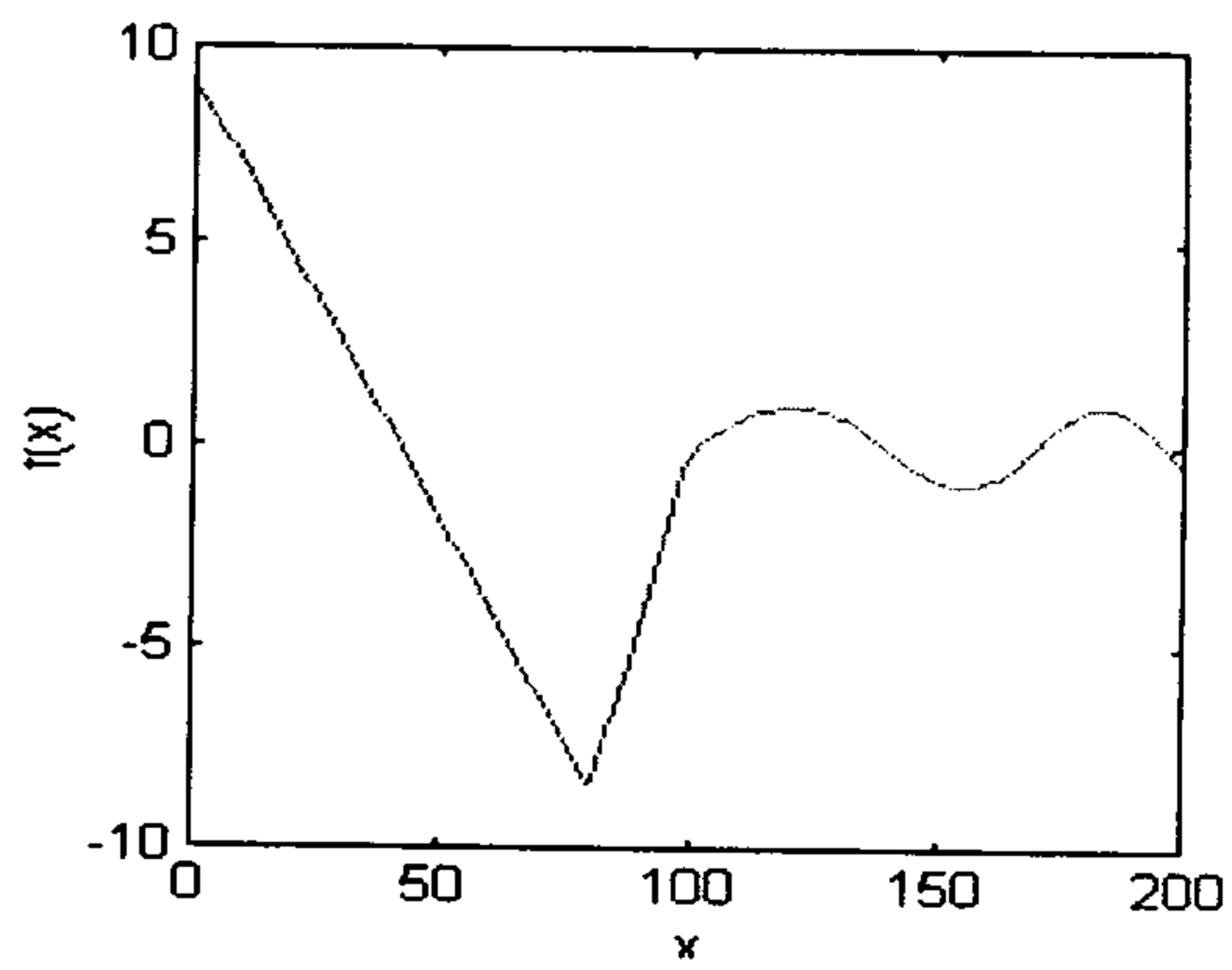
شکل ۹: (a): تابع اصلی (b): تقریب تابع توسط الگوریتم ۳ (c): تقریب تابع توسط الگوریتم ۱.

جدول ۵: چهار پارامتر اول بدست آمده با الگوریتمهای مختلف.

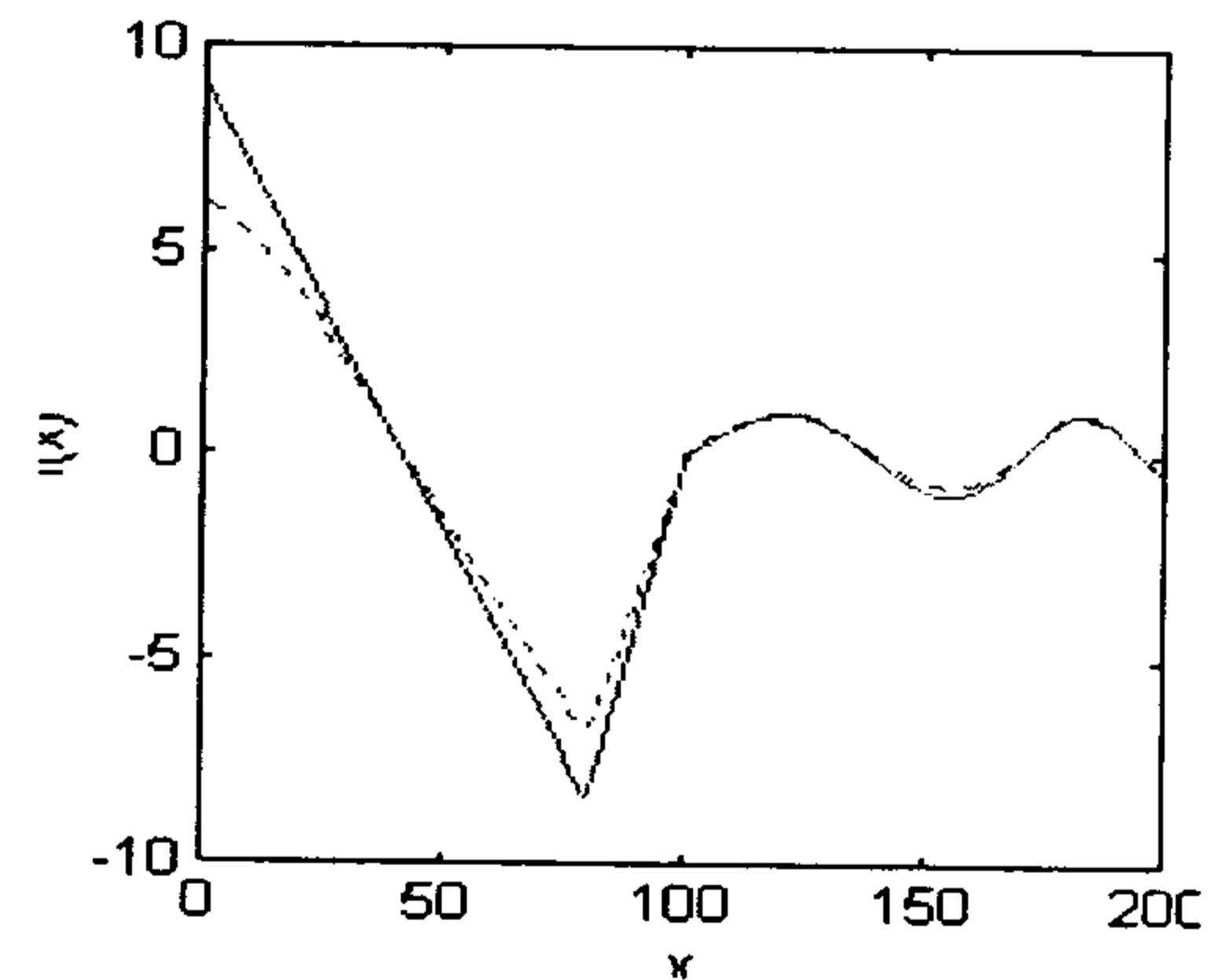
شماره پارامترها الگوریتم	۱	۲	۳	۴
الگوریتم ۳	-۰.۹۰۶	۰.۰۱۹	۰.۰۳۱۳	۰.۰۳۹۷
الگوریتم ۱	-۱.۴۸۳	-۰.۰۰۷۶	۰.۰۴۶۴	۰.۰۴۶
ویونت	-۴.۲۲۷	-۰.۰۱۵	۰.۸۳۸۶	۰

جدول ۶: مقادیر متوسط خطای تقریب برای الگوریتمهای مختلف.

تعداد پارامترها الگوریتم	۲	۶	۱۴
الگوریتم ۳	۰.۱۴۵۲	۰.۰۹۷۸	۰.۰۰۹۳۴
الگوریتم ۱	۰.۱۹۱۱	۰.۱۲۶۴	۰.۰۶۷۵۱
ویونت	۰.۲۷۳۶	۰.۲۱۵۸	۰.۱۳۲۸



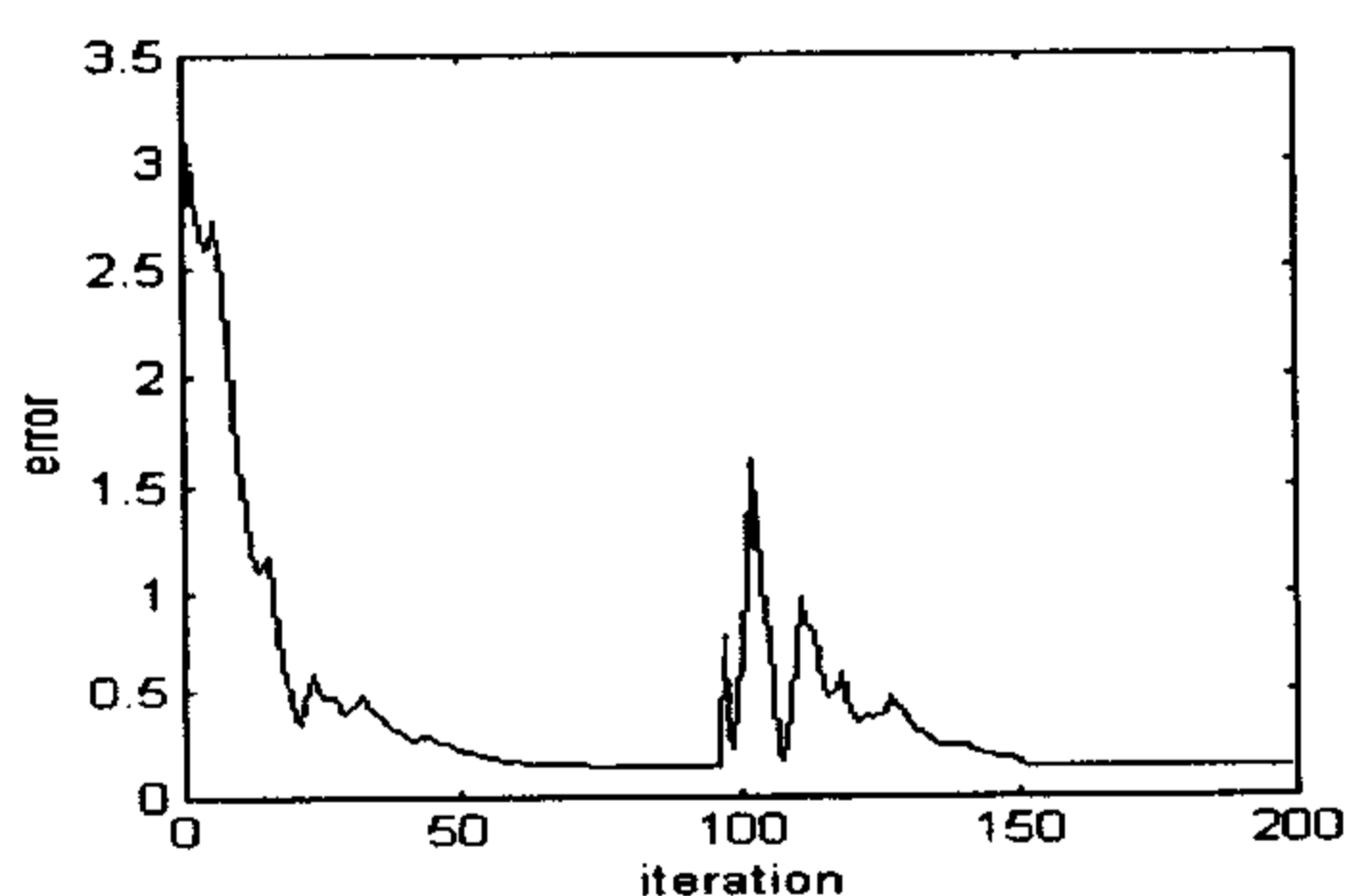
(a)



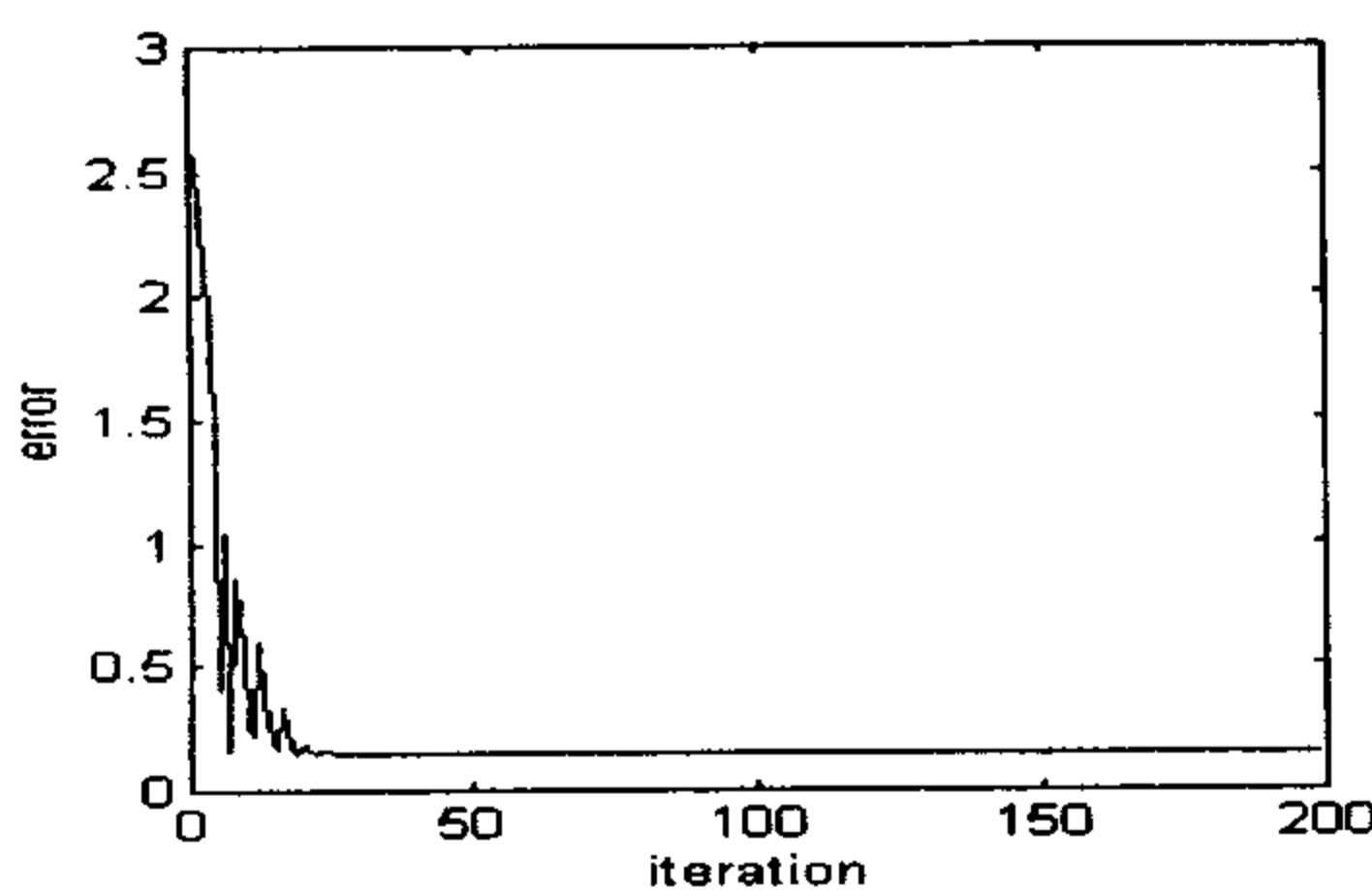
(b)

الگوریتمها قابل توجه می باشد این است که در هر مرحله میتوان فقط یک ویولت اضافه نمود و آنهایی را که کمکی به بهبود تقریب نمی کنند حذف کرد. بنابر این با حفظ دقت تقریب میتوان تعداد پارامترها یعنی ویولت ها را کاهش داد. به عبارت دیگر الگوریتمهای پیشنهادی بطور اتوماتیک تعداد ویولتهای مورد نیاز را تعیین میکند. نتایج به دست آمده را می توان به صورت زیر خلاصه نمود.

۱- روشهای پیشنهادی هم در تقریب تابع و هم در پیش بینی تابع در نقاط بین نقاط نمونه برداری شده بهتر از شبکه های عصبی با آموزش در امتداد نزول گرادیان عمل می کنند. ۲- خطای بدست آمده از اتوماتای پیوسته کمتر از دو اتوماتای دیگر است. ۳- اتوماتای اتوماتای پیوسته سریعتر همگرا می شود. چگونگی کاهش خطای تقریب با توجه به دفعات تکرار برای الگوریتم ۳ در شکل (۱۰) نشان داده شده است. همانگونه که مشاهده می شود در الگوریتم ۳ خطا پس از نوسانات اولیه سریعاً کاهش یافته و به سمت مقدار پایدار خود میل میکند اما در آموزش در امتداد نزول گرادیان خطا پس کاهش بدلیل ناهمواری سطح جستجو و تغییر گرادیان تابع ارزیابی ممکن است دوباره حالت نوسانی پیدا کند.



(a)



(b)

شکل ۱۰: کاهش خطای تقریب نسبت به تعداد دفعات تکرار (a): ویولت (b): اتوماتای یادگیر.

مثال چهارم: به عنوان آخرین مثال تقریب تابع دو متغیره $F(x) = (x_1^2 - x_2^2) \sin(0.5 x_1)$ در فاصله $x = [-10, 10]$ را با استفاده از الگوریتم ۳ نشان می دهیم. تابع ویولت استفاده شده گوسی دو متغیره $\psi(x) = x_1 x_2 \text{Exp}(-\frac{1}{2}(x_1^2 - x_2^2))$ انتخاب شده است. جهت شبیه سازی به ازای هر پارامتر تقریب یک اتوماتای پیوسته که که اقدام ان در فاصله فاصله [۵۰،۰] تغییر می کند در نظر گرفته شده است. نتایج شبیه سازی برای سه روش آموزش در امتداد نزول گرادیان و الگوریتم و اتوماتای پیوسته در شکل (۹) و خطای بدست آمده در جدول (۷) نشان داده شده است. میزان متوسط خطای بدست آمده پس از بیست بار تکرار برای آموزش در جهت نزول گرادیان برابر ۰/۰۷۵ و برای الگوریتمهای ۱ و ۳ بترتیب ۰/۰۱۶۷ و ۰/۰۱۲۷ می باشد.

جدول ۷: مقادیر متوسط خطای تقریب برای الگوریتم های ۱ و ۳.

تعداد پارامتر	۶	۲
الگوریتم ۱	۰/۰۱۶۷	۰/۳۷۴۲
الگوریتم ۳	۰/۰۱۲۷	۰/۴۵۶۴
ویولت	۰/۰۷۵۲۳	۰/۵۲۳۴

نتیجه گیری

در این مقاله ۳ الگوریتم مبتنی بر اتوماتای یادگیر برای محاسبه ضرائب ویولت ارائه گردیده است. این الگوریتمها خاصیت سلسله مراتبی ویولتها را حفظ کرده و به صورت پویا عمل می کنند. نکته ای که باید مورد توجه قرار گیرد این است که اتوماتای یادگیر مقادیر بهینه پارامترها را بین دو مقدار حداقل و حد اکثر جستجو می کند بنابر این باید از قبل اطلاعاتی در مورد مقادیر حداقل و حداکثر پارامترها در دست باشد. در مورد مسائل کاربردی با توجه به اینکه محدوده عملیاتی سیستم از قبل قابل پیش بینی می باشد بدست آوردن مقادیر حداقل و حد اکثر پارامترها امکان پذیر است اما در مورد سایر مسائل می توان از روشهای دیگر مانند ضرب مسقیم یک مقدار اولیه برای هر پارامتر بدست آورد سپس مقدار بهینه را اطراف این مقدار اولیه جستجو کرد. نکته دیگری که در این

مراجع

- 1 - Poggio, T. T. and Girosi, F. (1990). "Networks for approximation and learning." *Proc. IEEE*, Vol. 78, No. 9.
- 2 - Daubechies, I. (1990). "The wavelet transform, time-frequency localization and signal analysis." *IEEE Trans. Information Theory*, Vol. 36.
- 3 - Mallat, S. G. (1989). "A theory or multiresolution signal decomposition: the wavelet representation." *IEEE Trans. PAMI*, Vol. 11.
- 4 - Zhang, Q. and Benveniste, A. (1992). "Wavelet networks." *IEEE Trans. on Neural Networks*, No. 6, PP. 889-898.
- 5 - Bakhshi, B. R. and Stephanopoulos, G. (1993). "Wave-net: a mmultiresolution, hierarchical neural nnetworks with localized learning." *AICHE Journal*, No. 1.
- 6 - Safavi, A. A. (1997). "On the learning algorithms of wave-net." *Proceedings of Fifth Iranian Conference on Electrical Engineering*.
- 7 - Grossman, A. and Morlet, J. (1984). "Decomposition of hardly functions into square integrable wavlets." *SIAM. Math. Anal.*
- 8 - Daubechies, I. (1988). "Orthonormal basis of compactly suported waveles." *Communications on Pure and Applied Math.*, Vol. 41, PP. 909-999.
- 9 - Rao, S. and Pappu, R. S. (1993). "Hierarchical wavelet neural networks. " *IEEE Trans. on Neural Networks*.
- 10 - Narendra, K. S. and Thathachar, M. A. L. (1989). *Learning automata : an introduction*. Printice -Hall.
- 11 - Narendra, K. S. and Thathachar, M. A. L. (1974). "Learning automata :a survey." *IEEE Trans. MC 4*.
- 12 - Mars, P., Narendra, K. S. and Chrystall, M. (1983). "Learning automata control of computer communication networks." *Proc. of Third Yale Workshop on Applications of Adaptive Systems Theory*, Yale University.
- 13 - Meybodi, M. R. and Lakshmiarhan, S. (1982). "Optimality of a general class of learning algorithm." *Information Science*, Vol. 28, PP. 1-20.
- 14 - Meybodi, M. R. and Lakshmiarhan, S. (1984). *On a class of learning algorithms which have a symmetric behavior under success and failure*. Springer Verlag Lecture Notes in Statistics, PP.145-155.
- 15 - Howell, M. N. and Gordon, T. J. (1998). "Continous learning automata and adaptive digital designs." *Proceedings of UKACC International Conference on Control*, PP. 100-105.

واژه های کلیدی به ترتیب استفاده در متن

- | | | |
|-------------------------------------|------------------------------------|---------------------------|
| 1 – Wavelet | 2 – Harr | 3 – Environment |
| 4 – Unfavorable | 5 – Stationary | 6 – Non-Stationary |
| 7 – Fixed Structure | 8 – Actions | 9 – Variable Structure |
| 10 – Linear Reward Penalty | 11 – Linear Reward Epsilon Penalty | |
| 12 - Linear Reward Inaction | 13 – Gradient Decent | 14 – Direct Inner Product |
| 15 – Continuous Srrred Tank Reactor | | |