

## مدل‌سازی و شناسایی سیستم‌های دینامیکی غیرخطی با استفاده از یک سیستم فازی عصبی خودسازماندهی برخط

حمید طباطبائی<sup>۱\*</sup>، شیرین ریخته گرمشهد<sup>۲</sup>

<sup>۱</sup>گروه مهندسی کامپیوتر، دانشگاه آزاد اسلامی واحد قوچان، ایران.

<sup>۲</sup>گروه مهندسی کامپیوتر، دانشگاه آزاد اسلامی، واحد نیشابور، ایران.

### چکیده

در این مقاله، یک سیستم فازی-عصبی خودسازمانده برای یادگیری تطبیقی برخط برای شناسایی و مدل‌سازی سیستم‌های دینامیکی غیرخطی معرفی شده است. در این سیستم، در ابتدا هیچ نودی در لایه پنهان وجود ندارد و چنانچه معیارهای تولید قوانین در طی فرآیند آموزش برآورده شود نرون RBF به لایه پنهان اضافه می‌شود. از الگوریتم آموزش حداقل مربعات بازگشتی وزن‌دار (WRLS) برای قابلیت یادگیری برخط و افزایش سرعت همگرایی، در فاز یادگیری پارامترهای قسمت تالی قوانین نوع تاکاگی سوگنو استفاده شده است. در فاز یادگیری، ساختار برای تولید تعداد قوانین مناسب، معیار جدید درجهی تطبیق و معیار متداول خطا به کار گرفته شده است. بعد از ایجاد قانون جدید، کارایی سیستم محاسبه شده و برای ایجاد شبکه‌ای با ساختار فشرده‌تر قوانینی که تاثیر کم‌تری در کارایی سیستم دارند با یک الگوریتم هرس جدید هرس می‌شوند. در پایان، برای بهینه‌سازی ساختار توابع عضویت مشابه با یکدیگر ترکیب می‌شوند. برای بررسی عملکرد سیستم، دو سیستم دینامیک غیرخطی مبنای، در دو حالت نویزی و بدون نویز در محیط Matlab مدل‌سازی شده‌اند. دقت این مدل‌سازی بر مبنای دو معیار تعداد نرون‌ها (قوانین) و ریشه‌ی میانگین مربعات خطا با سایر روش‌ها مقایسه شده است. با توجه به نتایج به دست آمده، میانگین درصد بهبود جواب‌ها در تعداد قوانین به دست آمده نسبت به روش مبنای انتخاب شده در مدل‌سازی این دو سیستم در دو حالت نویزی و بدون نویز در مثال اول ۳۵/۴۲٪ و در مثال دوم ۲۹٪ می‌باشد.

**واژه‌های کلیدی:** شناسایی سیستم‌های غیرخطی، سیستم‌های فازی-عصبی خودسازمانده، قوانین تاکاگی، سوگنو، نویز.

پذیرش: ۱۳۹۸/۱/۱۹

اصلاح: ۱۳۹۷/۱۲/۱۵

دریافت: ۱۳۹۷/۱۰/۱۱

### ۱- مقدمه

یکی از مسائل کلیدی در شناسایی سیستم‌ها، پیدا کردن یک ساختار مدل مناسب است که بتوان توسط آن مدل، رفتار فیزیکی سیستم را مشخص نمود. عموماً این مدل از روی داده‌های ورودی و خروجی سیستم به دست می‌آید. از این رو، روش‌هایی برای مدل‌سازی و شناسایی سیستم‌ها مورد نیاز است. اغلب مسائلی که در دنیای واقعی وجود دارند به‌طور ذاتی غیرخطی هستند و سیستم‌های خطی به دست آوردن مدل ریاضی برای این سیستم‌ها کار دشواری است. شناسایی سیستم‌ها به دو طریق شناسایی برون خطی و برخط امکان‌پذیر است. در شناسایی برون خطی، پارامترها و ساختار سیستم، بعد از ارائه‌ی کامل مجموعه‌ی آموزش به‌روزرسانی می‌شوند، اما در شناسایی برخط، بعد از ورود



هر جفت داده‌ی ورودی-خروجی، ساختار و پارامترهای سیستم به‌روزرسانی می‌شوند. یادگیری برخط به دلایلی مانند تغییرات دینامیک سیستم، تغییر شرایط عملکردی، افزایش حالت‌های سیستم در طی زمان و در دسترس بودن اطلاعات محدودحائز اهمیت است.

تحقیقات نشان می‌دهند که سیستم‌های فازی-عصبی در سال‌های اخیر به‌طور گسترده‌ای در شناسایی سیستم‌های غیرخطی مورد استفاده قرار گرفته‌اند. مزیت اصلی مدل‌سازی سیستم‌های فازی-عصبی، توانایی تفسیر و قابلیت یادگیری این سیستم‌هاست. به‌طور کلی مدل‌سازی فازی-عصبی شامل دو فاز شناسایی ساختار و پارامترها است. در فاز شناسایی ساختار، قوانین فازی استخراج می‌شود و در فاز شناسایی پارامترها، پارامترهای قوانین فازی (مجموعه‌های فازی و توابع عضویت فازی) به‌روزرسانی می‌شوند. چنانچه تعداد قوانین حاصل زیاد باشد، هزینه‌ی محاسباتی و زمان بیشتری در فرآیند یادگیری سیستم موردنیاز است و چنانچه تعداد قوانین کم باشد کارایی مطلوب به‌دست نخواهد آمد. از این‌رو، شناسایی ساختار، نقش مهمی در مدل‌سازی سیستم‌های فازی-عصبی ایفا می‌کند.

برای رویارویی با مشکلات فوق، طراحی سیستم‌های فازی-عصبی خودسازمانده موردتوجه قرار گرفت. در این سیستم‌ها، نه تنها پارامترها بلکه ساختار نیز به‌طور تطبیقی در طی فرآیند یادگیری تعیین می‌شوند و نیازی به دانش اولیه برای ایجاد قوانین فازی و مجموعه‌های فازی اولیه نیست. از جمله دشواری‌هایی که در طراحی این سیستم‌ها وجود دارند تعیین تعداد مناسب قوانین فازی یا تعداد نرون‌های شبکه است به‌نحوی که سیستم بتواند مجموعه‌ی آموزش را به‌طور صحیح یاد بگیرد و قابلیت تعمیم‌پذیری بر روی داده‌های تست را داشته باشد و دیگری آموزش پارامترهاست. الگوریتم پس‌انتشار خطا (بی‌پی)<sup>۱</sup> به‌طور وسیعی در آموزش پارامترها استفاده شده است. بی‌پی یکی از قدرتمندترین تکنیک‌های یادگیری در شبکه‌های عصبی چندلایه است. از آنجایی که از تکنیک بهینه‌سازی Steepest-Descent در این الگوریتم استفاده می‌شود، امکان گیرافتادن در بهینه‌های محلی در این روش وجود دارد. علاوه‌بر این مسئله، کارایی الگوریتم بی‌پی وابسته به مقادیر اولیه‌ی پارامترهای سیستم است و سرعت همگرایی در این روش بسیار پایین است.

این مقاله، یک نوع جدیدی از سیستم‌های استنتاج فازی را معرفی می‌نماید. این سیستم، یک سیستم فازی-عصبی خودسازمانده برای یادگیری تطبیقی برخط است که برای شناسایی و مدل‌سازی سیستم‌های دینامیکی غیرخطی به‌کار می‌رود. در این سیستم، در ابتدا هیچ نودی در لایه‌ی پنهان وجود ندارد و چنانچه معیارهای تولید قوانین در طی فرآیند آموزش برآورده شوند، نرون RBF به لایه‌ی پنهان اضافه می‌شود.

نوآوری‌های اصلی این مقاله عبارت است از:

- استفاده از دو معیار در فاز رشد قوانین: درجه‌ی تطبیق و معیار متداول خطا.
- ارائه‌ی یک الگوریتم هرس جدید براساس چگالی: چگالی تعداد دفعاتی است که یک قانون آتش می‌شود، هر بار که یک الگو توسط یک قانون پوشش داده می‌شود به چگالی آن قانون یک واحد اضافه می‌شود در پایان قانونی که کم‌ترین مقدار چگالی را داشته باشد از بین قوانین موجود هرس می‌شود.
- ترکیب توابع عضویت مشابه و تغییر عرض آن‌ها به‌منظور افزایش کارایی سیستم.
- ایجاد قوانین فازی بدون استفاده از الگوریتم پس‌انتشار خطا: براساس الگوریتم آموزش حداقل مربعات بازگشتی وزن‌دار (WRLS).

نتایج شبیه‌سازی با بررسی عملکرد سیستم در حضور نویز روی دو مسأله‌ی مینا، شامل شناسایی سیستم‌های دینامیک غیرخطی حاکی از دقت بالاتر و ساختار فشرده‌تر الگوریتم پیشنهادی نسبت به سایر روش‌هاست.

این مقاله، توسعه‌ای بر مقاله‌ی (ریخته‌گر مشهد و اکبرزاده توتونچی، ۱۳۹۲) است که در پنج بخش سازمانده‌ی شده است. در ادامه، در بخش ۲ بعداز مقدمه، مروری بر روش‌های به‌کار گرفته شده برای شناسایی سیستم‌های غیرخطی انجام گرفته است. ساختار OSO

<sup>۱</sup>Back Propagation (BP)



NFS که معادل با قوانین فازی نوع تاکاگی- سوگنو است و الگوریتم یادگیری شامل الگوریتم‌های رشد و هرس شبکه به تفصیل در بخش ۳ شرح داده شده است، در بخش ۴، شبیه‌سازی الگوریتم پیشنهادی ارائه شده است و در نهایت ارزیابی نتایج و مقایسه‌ی نتایج با سایر روش‌ها در بخش ۵ صورت گرفته است.

## ۲- مروری بر کارهای گذشته

اخیراً شبکه‌های فازی-عصبی خودسازمانده‌ی گوناگونی برای شناسایی سیستم‌های غیرخطی به کار گرفته شده‌اند. Er و Wu در (وو و ای آر، ۲۰۰۰) یک الگوریتم یادگیری خود سازماندهی سلسله مراتبی برای سیستم فازی-عصبی دینامیک (DFNN) ارائه کرده‌اند. DFNN، قوانین فازی به فرم تاکاگی- سوگنو را ایجاد می‌کند که در این قوانین، عرض توابع عضویت گوسی برای تمام متغیرهای ورودی یکسان است که این امر واقع‌بینانه نیست مخصوصاً زمانی که متغیرهای ورودی، بازه‌های مختلف عملکردی داشته باشند. از این رو، نسخه‌ی اصلاح شده‌ی این روش در (وو و همکاران، ۲۰۰۳) به نام سیستم GDFNN ارائه شد که در این سیستم برای هر تابع عضویت گوسی داخل هر نرون RBF، بردار عرض متناسب معرفی شده است. اگرچه در این روش نیز الگوریتم یادگیری سلسله مراتبی برای آموزش GDFNN وابسته به کل داده‌های آموزش می‌باشد، از این رو GDFNN برای یادگیری برون خطی مناسب است.

ونگ و همکاران (۲۰۰۹) یک شبکه‌ی فازی-عصبی خودسازمانده‌ی برخط سریع با دقت بالا (FAOS-PFNN) ارائه کرده است که این سیستم ابتدا بدون هیچ قانونی شروع به کار می‌کند و براساس یک معیار رشد، برحسب نیاز قوانین به لایه‌ی پنهان در طی فرآیند یادگیری اضافه می‌شود. متأسفانه در FAOS-PFNN امکان هرس نرون‌های اضافی (قوانین) لایه‌ی پنهان وجود ندارد از سوی دیگر تحلیلی برای همگرایی FAOS-PFNN ارائه نشده است. نسخه‌ی تعمیم یافته‌ی FAOS-PFNN مدل GEBF-OSFNN (ونگ، ۲۰۱۱) می‌باشد که در این مدل از توابع عضویت گوسی نامتقارن به دلیل افزایش انعطاف‌پذیری و قابلیت تفسیر بالاتر فضای ورودی استفاده شده است. به منظور ایجاد قوانین فازی توسط یک معیار رشد، از یک مجموعه‌ی داده‌ی بزرگ، یک شبکه‌ی فازی خودسازمانده (SOFNMS) توسط د جسو رایبو (۲۰۰۹) ارائه شده است که توسط الگوریتم آموزش حداقل مربعات اصلاح شده‌ی پارامترهای سیستم فازی به روزرسانی می‌شود. در SOFNLMS از الگوریتم نزدیک‌ترین همسایه برای ایجاد قوانین استفاده شده است. به این معنی که یک قانون جدید در صورتی ایجاد می‌شود که فاصله‌ی بین الگوی ورودی از مراکز نرون‌های RBF موجود از یک حد از پیش تعیین شده ای بیشتر باشد. به طور هم زمان از یک الگوریتم هرس برای هرس قوانین اضافی استفاده شده است. اگرچه قوانین فازی در SOFNLMS فقط براساس الگوی جدید تعیین می‌شود اما ایراد این روش این است که در مرحله‌ی شناسایی ساختار، تنها یک نرون می‌تواند حذف شود.

در شبکه‌ی GP-FNN که توسط هان و کیو آی آی او (۲۰۱۰) ارائه شده است، از روش آنالیز حساسیت خروجی در فاز یادگیری ساختار و از الگوریتم‌های یادگیری گرادینانی در فاز شناسایی پارامترها استفاده شده است. با وجود این که ارائه‌ی الگوریتمی قدرتمند در فاز رشد و هرس GP-FNN، زمان هم‌گرایی در فاز یادگیری پارامترها به دلیل استفاده از محاسبات گرادینانی بالاست. کاو و همکاران (۲۰۱۲) و اچ اس یو (۲۰۱۲) یک شبکه‌ی فازی عصبی خودسازمانده از نوع تاکاگی-سوگنو (TSK-SOFNN) معرفی کرده‌اند که از معیار فاصله‌ی میانگین در حافظه‌های انجمنی در فاز رشد این شبکه استفاده شده است. در این شبکه، برای هر قانون یک اندیس اهمیت در نظر گرفته شده است که چنانچه این اندیس از یک حد آستانه‌ای کم‌تر باشد قانون متناظر با آن هرس می‌شود. TSK-SOFNN برای کاربردهای برون خطی مناسب است. چن (۲۰۱۱) از روش هرس مرتبه‌ی دوم OBS که بسط تیلور مرتبه‌ی دوم تابع خطاست برای حذف قوانین اضافی استفاده کرده است که باعث افزایش بار محاسباتی می‌گردد.

در بعضی از روش‌ها از الگوریتم‌های تکاملی مانند الگوریتم ژنتیک (لنگ و همکاران، ۲۰۰۶؛ آکالا فرناندز و همکاران، ۲۰۰۹) برای به دست آوردن تعداد قوانین بهینه استفاده شده است. در روش‌های مبتنی بر GA تعداد قوانین فازی و هم چنین پارامترهای توابع عضویت در غالب یک کروموزوم پیچیده کدگذاری می‌شود. در پایان، قانونی که قسمت مقدم یا قسمت تالی صفر داشته باشد در مجموعه قوانین فازی نهایی وجود نخواهد داشت. هر چند که تعداد مجموعه‌های فازی برای هر متغیر ورودی مبتنی بر دانش از پیش تعیین شده‌ای است اما با این روش، ماکزیمم تعداد قوانین به دست می‌آید.



این تکنیک‌ها به صورت موازی و سراسری<sup>۱</sup> هستند. به این دلیل که به طور هم‌زمان نقاط زیادی را در فضای جستجو مورد ارزیابی قرار می‌دهند. در این تکنیک‌ها امکان هم‌گرا شدن به بهینه‌ی سراسری زیاد است. از این رو تکنیک‌های تکاملی در آموزش شبکه‌های فازی-عصبی یک فیلد مهم به‌شمار می‌آیند. لنگ و همکاران (۲۰۰۶) یک الگوریتم ترکیبی جدید برای شبکه‌ی فازی-عصبی خودسازمانده مبتنی بر GA ارائه کرده است. یکی از نوآوری‌های این مدل این است که در این روش، نیازی به دانش اولیه برای تقسیم‌بندی فضای ورودی و تعداد قوانین فازی وجود ندارد.

خیاط و همکاران (۲۰۰۹) الگوریتم خوشه‌بندی فازی SOFNNGAPS را برای تعیین تعداد بهینه‌ی قوانین فازی استفاده کرده‌اند و یک الگوریتم ترکیبی GA و PSO در فاز یادگیری پارامترها به‌کار گرفته شده است. چن و همکاران (۲۰۱۱) الگوریتم یادگیری RSMODE را برای یک سیستم فازی-عصبی خودسازمانده ارائه داده‌اند که در این الگوریتم از روش چند زیرجمعیتی<sup>۲</sup> در فاز یادگیری قوانین استفاده شده است که برای هر زیرجمعیت به طور جداگانه یک کروموزوم در نظر گرفته شده است که هر کروموزوم بیانگر یک قانون است و به طور جداگانه مورد ارزیابی قرار می‌گیرند. پارامترها توسط الگوریتم تکاملی تفاضلی بهبود داده شده تعیین می‌شوند.

لین و همکاران (۲۰۱۲) SOME را که الگوریتم یادگیری برای شبکه‌ی فازی-عصبی تاکاگی سوگنو معرفی نمودند. در این الگوریتم از گروه مبتنی بر تکامل هم‌زیستی<sup>۳</sup> استفاده شده است که در هر گروه فقط یک قانون فازی نمایش داده می‌شود. در فاز یادگیری ساختار از یک الگوریتم خودسازمانده‌ی دومرحله‌ای و در فاز یادگیری پارامترها از یک روش انتخاب مبتنی بر داده کاوی<sup>۴</sup> استفاده شده است. بزرگ‌ترین نقطه‌ی ضعف الگوریتم‌های مبتنی بر GA زمان آموزش طولانی است.

روش‌هایی در (جانگ و همکاران، ۲۰۰۷؛ جانگ و همکاران، ۲۰۰۸) برای ایجاد قوانین فازی مبتنی بر SVM<sup>۵</sup> ارائه شده است. در این مقالات از SVM برای ایجاد خودکار قوانین فازی استفاده شده است که تعداد قوانین برابر با تعداد SVM هاست. مهم‌ترین مسأله در یادگیری SVM، مسأله‌ی نرون‌های مرده<sup>۶</sup> می‌باشد. این‌که چگونه می‌توان توسط SVM تعداد بهینه‌ی قوانین فازی را برای NFS<sup>۷</sup> تعیین نمود هنوز یک مسأله‌ی چالش برانگیز است. داهال و همکاران (۲۰۱۵) از الگوریتم GA برای شناسایی قوانین فازی استفاده کردند. تنظیم وزن‌های قوانین به صورت جداگانه و در مراحل مختلف صورت می‌گیرد که این امر منجر به کاهش طول کروموزوم‌ها و کوچک شدن فضای جستجو می‌گردد. بزرگ‌ترین نقطه‌ی ضعف الگوریتم‌های مبتنی بر GA زمان آموزش طولانی است. نوئن و همکاران (۲۰۱۵) از یک الگوریتم خوشه‌بندی جدید برای به‌دست آوردن تعداد قوانین فازی بهینه و یک الگوریتم هرس مبتنی بر روش hebbian برای کاهش تعداد قوانین فازی استفاده نمودند. طاووسی و همکاران (۲۰۱۶) از مقادیر فازی نوع دو برای مقداردهی وزن‌ها استفاده کردند و الگوریتم KM بهبودیافته برای تنظیم پارامترها و یک الگوریتم خوشه‌بندی فازی نوع دو برای پیدا کردن نرون‌ها در لایه‌ی پنهان را به‌کار گرفتند. در مقاله‌ی (هان و همکاران، ۲۰۱۷) از روش کاهش نرخ خطا در فاز یادگیری ساختار و از یک الگوریتم تطبیقی گرادینانی جدید در فاز یادگیری پارامترها استفاده شده است. در مقاله‌ی (لیون و همکاران، ۲۰۱۷)، از یک الگوریتم ترکیبی فازی-عصبی و یکی از تکنیک‌های یادگیری ماشین به نام Adaboost برای کلاس‌بندی داده‌های سرطانی استفاده شده است. در مقاله (لین و لی، ۲۰۱۷)، نویسندگان، یک سیستم فازی-عصبی خودسازمانده‌ی فازی نوع دو بازه‌ای برای سیستم ABS<sup>۸</sup> خودروها معرفی کرده‌اند. پارامترهای سیستم با استفاده از الگوریتم گرادینانی تخمین زده می‌شوند و از الگوریتم PSO<sup>۹</sup> برای پیدا کردن نرخ یادگیری بهینه‌ی وزن‌ها استفاده شده است (هان و همکاران، ۲۰۱۳). هان و همکاران (۲۰۱۸) یک شبکه شعاعی پایه برای مدل‌سازی سیستم‌های غیرخطی ارائه کردند. در فرآیند یادگیری، دو مرحله به‌کار گرفته شده است. در مرحله اول، هدف، پیدا کردن پارامترهای

<sup>۱</sup>Global

<sup>۲</sup>Multi-subpopulation

<sup>۳</sup>Group-based symbiotic evolution

<sup>۴</sup>Selection strategy based on data mining

<sup>۵</sup>Support Vector Machine

<sup>۶</sup>Dead Neurons Problem

<sup>۷</sup>Neuro Fuzzy Systems

<sup>۸</sup>Anti-Lock Braking System

<sup>۹</sup>Partial Swarm Optimization



بهینه‌سازی استفاده از الگوریتم لوبنرگ-مارکوت<sup>۱</sup> بهبود داده‌شده می‌باشد و در مرحله‌ی دوم از الگوریتم (IOA)<sup>۲</sup> برای بهینه‌سازی ساختار استفاده شده است. نتایج به‌دست‌آمده بر روی چند مساله‌ی پایه، نشان‌دهنده‌ی دقت و سرعت بالا و ساختار فشرده الگوریتم پیشنهادی در مدل‌سازی سیستم‌های غیرخطی است. یک الگوریتم بهبود داده‌شده، روش کاهش نرخ خطا در (منگ و همکاران، ۲۰۱۷) برای مدل‌سازی فرآیندهای صنعتی ارائه شده است که در این مقاله ساختار فشرده و قدرت تعمیم رضایت‌بخش با یک مرحله یادگیری به‌دست آمده است. استفاده از مکانیزم ساختاری مرتبه‌ی دوم تضمینی برای ساختار فشرده و کارآمدی محاسبات است و پارامترهای سیستم غیرخطی در الگوریتم پیشنهادی تصادفی نیستند. همگرایی سیستم پیشنهادی نیز مورد تجزیه و تحلیل قرار گرفته است. نتایج، نشان‌دهنده‌ی کارآمدی سیستم پیشنهادی در مقایسه با روش‌های دیگری در کاربردهای صنعتی است.

نویسندگان در مقاله‌ی (کومار و همکاران، ۲۰۱۸) از شبکه‌های شعاعی پایه برای کنترل مسیر از پیش تعریف‌شده‌ی کنترل‌رهای ربات‌ها استفاده کرده‌اند. در این روش از الگوریتم گرادینتی برای به‌روزرسانی پارامترها در معادلات غیرخطی استفاده شده است. مزیت استفاده از این الگوریتم، تأثیر خوشه‌بندی در توزیع مراکز شعاعی می‌باشد. فرآیندهای کنترل و شناسایی هم‌زمان با استفاده از شبکه‌های RBF انجام شده است. کارایی روش پیشنهادی با روش‌هایی از جمله شبکه‌های عصبی چندلایه بر اساس معیارهایی مانند میانگین مربعات خطا، تحمل اختلالات و تنوع پارامترها مقایسه شده است.

### ۳- روش پیشنهادی OSO-NFS

در این بخش ساختار ۴ لایه‌ی OSO-NFS که معادل با قوانین فازی نوع تاکاگی-سوگنو است و هم‌چنین الگوریتم یادگیری شامل الگوریتم‌های رشد و هرس شبکه و ترکیب توابع عضویت مشابه به‌منظور افزایش کارایی و ساده‌سازی و کاهش پیچیدگی سیستم، ارائه شده است.

#### ۳-۱- ساختار OSO-NFS

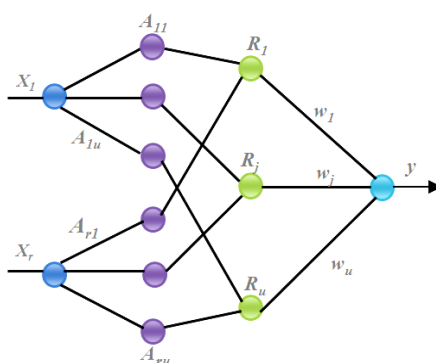
در این بخش ساختار OSO-NFS در شکل ۱ بیان شده است. یک شبکه‌ی چهار لایه‌ی مشابه با (ریخته‌گر مشهد و اکبرزاده توتونچی، ۱۳۹۲؛ وو و همکاران، ۲۰۰۰؛ وو و همکاران، ۲۰۰۱) برای پیاده‌سازی قوانین فازی نوع تاکاگی - قوانین سوگنو به‌فرم زیر ارائه شده است: فرض می‌کنیم تعداد متغیرهای ورودی  $r$  است.

*Rule j: IF  $x_1$  is  $A_{1j}$  and ... and  $x_r$  is  $A_{rj}$  THEN  $y$  is  $W_j$*

$$j = 1, 2, \dots, u \quad (1)$$

$$W_j = \alpha_{0j} + \alpha_{1j}x_1 + \dots + \alpha_{rj}x_r$$

در رابطه‌ی فوق  $A_{1j}, \dots, A_{rj}$  مجموعه‌های فازی قسمت مقدم قوانین (توابع عضویت گوسی)  $\alpha_{0j}, \alpha_{1j}, \dots, \alpha_{rj}$  پارامترهای قسمت تالی و وزن‌های شبکه هستند.



شکل ۱- ساختار OSO-NFS (ریخته‌گر مشهد و اکبرزاده تونوچی، ۱۳۹۲).

لایه‌ی اول: هر نود در این لایه نشان‌دهنده‌ی یک متغیر ورودی است  $x_i$  ( $i = 1, 2, \dots, r$ ).

لایه‌ی دوم: هر نود در این لایه معرف تابع عضویت است. هر متغیر ورودی در لایه‌ی اول  $U$  مجموعه‌ی فازی دارد  $A_{ij}$  ( $j = 1, 2, \dots, u$ ).

که  $\mu_{ij}$ ، زامین تابع عضویت متغیر  $x_i$  و  $c_{ij}$  و  $\sigma_{ij}$  به ترتیب مرکز و عرض این توابع عضویت هستند.

$$\mu_{ij}(x_i) = \exp\left(-\frac{(x_i - c_{ij})^2}{\sigma_{ij}^2}\right), i = 1, 2, \dots, r, \quad (2)$$

$$j = 1, 2, \dots, u$$

لایه‌ی سوم: هر نود در این لایه، نود قانون نامیده می‌شود که قسمت "اگر" قوانین فازی را نشان می‌دهد ( $t$ -نرم قسمت مقدم قوانین) خروجی زامین قانون  $R_j$  ( $j = 1, 2, \dots, u$ ) به فرم زیر است:

$$\varphi_j(x_1, x_2, \dots, x_r) = \exp\left(-\sum_{i=1}^r \frac{(x_i - c_{ij})^2}{\sigma_{ij}^2}\right). \quad (3)$$

$$j = 1, 2, \dots, u$$

لایه‌ی چهارم: هر نود در این لایه یک نود خروجی است و خروجی این نود از حاصل جمع سیگنال‌های وارد شده به آن به دست می‌آید.

$$y(x_1, x_2, \dots, x_r) = \sum_{j=1}^u w_j \varphi_j. \quad (4)$$

در شکل ۱ لایه‌ی خروجی دارای یک نود است برای نمایش ساده‌تر که معرف سیستم چندورودی تک خروجی است (MISO). قابل ذکر است که نتایج می‌توانند به یک سیستم چندورودی چندخروجی بسط داده شود.

### ۲-۳- الگوریتم یادگیری در OSO-NFS

دو فاز مهم در طراحی سیستم‌های فازی-عصبی عبارت‌اند از فاز یادگیری پارامترها (تنظیم پارامترهای قسمت مقدم و تالی قوانین و وزن‌های شبکه) و یادگیری ساختار که همان فاز تولید قوانین بهینه است. در این بخش، الگوریتم‌هایی به این منظور معرفی شده است.

#### ۲-۳-۱- معیارهای تولید نرون (قوانین) در لایه‌ی پنهان

در این بخش، دو معیار عبارت‌از معیار خطا و معیار درجه‌ی تطبیق برای ایجاد قوانین جدید ارائه شده است. این سیستم در ابتدا بدون هیچ نودی در لایه‌ی پنهان شروع به کار می‌کند و در طی فرآیند یادگیری چنانچه معیارهای تولید قوانین برآورده شود قوانین به لایه‌ی پنهان اضافه می‌شود.

خطای خروجی یک فاکتور مهم در تعیین این امر است که قانون جدید ایجاد شود یا خیر. معیار خطا به این صورت شرح داده می‌شود: برای هر الگوی ورودی به صورت زوج داده‌ی  $(XK, tK)$ ،  $k=1,2,\dots,n$  که  $n$  تعداد نمونه‌های آموزش است و  $XK$ ،  $k$  امین بردار ورودی و  $tK$ ،  $k$  امین خروجی مطلوب و  $y^k$  خروجی واقعی (شبکه‌ی فازی-عصبی) است که از حاصل جمع سیگنال‌های ورودی به نودهای خروجی با توجه به رابطه‌ی (۴) به دست می‌آید. خطای سیستم طبق رابطه‌ی زیر محاسبه می‌شود:

$$\|e^k\| = \|t^k - y^k\| \quad (5)$$

با توجه به رابطه‌ی فوق اگر

$$\|e^k\| > k_e \quad (6)$$

باشد یک نود قانون جدید باید به لایه‌ی پنهان اضافه شود که  $k_e$  یک ثابت از پیش تعریف شده است که در طی فرآیند یادگیری کاهش پیدا می‌کند و توسط روابط زیر محاسبه می‌شود:

$$k_e = \begin{cases} e_{max} & 1 < k < n/3 \\ \max[e_{max} \times \beta^k, e_{min}], & n/3 \leq k \leq 2n/3 \\ e_{min} & 2n/3 < k \leq n \end{cases} \quad (7)$$

که در رابطه‌ی فوق:

$e_{min}$ : دقت مطلوب خروجی OSO-NFS.

$e_{max}$ : حداکثر خطا.

$k$ : دوره‌ی آموزش.

$\beta \in (0,1)$ : ثابت همگرایی.

$$\beta = \left(\frac{e_{min}}{e_{max}}\right)^{3/n} \quad (8)$$

۳-۱-۲-۲- معیار درجه‌ی تطبیق

با ورود الگوی  $i$ ام ( $i=1,2,\dots,n$ ) درجه‌ی تطبیق این داده‌ی جدید با هر نرون قانون در لایه‌ی سوم طبق رابطه‌ی (۳) محاسبه خواهد شد. چنانچه ماکزیمم این مقدار از یک حد آستانه‌ای کم‌تر باشد به این معنا است که این داده نمی‌تواند توسط قوانین موجود پوشش داده شود و در این صورت یک قانون جدید باید ایجاد شود.

$$\phi_{max} = \max(\phi_j(x)), j = 1, \dots, u \quad (9)$$

که  $u$  تعداد نرون‌ها در لایه‌ی پنهان است. چنانچه  $\phi_{max} < \mu$  یک قانون جدید باید ایجاد شود.





بعضی اوقات ممکن است یک نرون در ابتدا فعال شود اما در پایان، سهم کوچکی در تولید خروجی سیستم داشته باشد. علاوه بر این، هرس شبکه برای سیستم‌های غیرخطی دینامیک متغیر با زمان امری ضروری است. اگر نرون‌های غیرفعال در لایه‌ی پنهان قابل شناسایی و حذف باشند در طی فرآیند یادگیری می‌توان به شبکه‌ای با توپولوژی فشرده‌تر دست پیدا کرد که این امر در نهایت منجر به کاهش زمان آموزش خواهد شد. در این بخش یک الگوریتم جدید براساس چگالی ارائه خواهد شد. چگالی، تعداد دفعاتی است که هر قانون آتش شده است. زمانی که یک قانون جدید ایجاد می‌شود چگالی آن یک قرار داده می‌شود.  $dM+1=1$  که  $M$  تعداد قوانین موجود است. با ورود هر الگوی جدید چنانچه این داده توسط یکی از قوانین موجود در سیستم پوشش داده شود چگالی آن قانون یک واحد اضافه خواهد شد. در پایان، قانون کم‌اهمیت قانونی است که کم‌ترین مقدار چگالی را داشته باشد. بعد از چند دوره  $\Delta(L)$  چنانچه مقدار چگالی یک قانون از حد آستانه‌ای کم‌تر باشد آن قانون حذف خواهد شد.

$$d_{\min} = \min(d_j), 1 \leq j \leq u \quad (10)$$

اگر  $u > 2$  و  $d_{\min} \leq du$  این قانون حذف خواهد شد که  $du$  مینیمم چگالی انتخاب شده به نام پارامتر آستانه است.

### ۳-۲-۳- تنظیم پارامترهای قسمت مقدم

بعد از رشد یک نرون، گام بعد تعیین پارامترهای قسمت تالی است. پارامترهای نود RBF طبق روابط زیر تعیین می‌شوند:

$$C_i = X_i \quad (11)$$

$$\sigma_i = \frac{\max\{|c_i - c_{i-1}|, |c_i - c_{i+1}|\}}{\sqrt{\ln\left(\frac{1}{\epsilon}\right)}} \quad (12)$$

در رابطه‌ی فوق  $C_{i+1}$  و  $C_{i-1}$  مراکز دو تابع عضویت همسایه هستند.

### ۳-۲-۴- تنظیم پارامترهای قسمت تالی

در این بخش، پارامترهای قسمت تالی (وزن‌های شبکه) به وسیله‌ی الگوریتم حداقل مربعات بازگشتی وزن دار (WRLS) (کاسابو و سانگ، ۲۰۰۲) تخمین زده می‌شوند. فرض کنید  $u$  قانون فازی برای  $n$  نمونه‌ی الگوی ورودی خروجی برای  $r$  متغییر ورودی تولید شده باشد، با نوشتن رابطه‌ی (۴) به فرم ماتریس داریم:

$$W\varphi = Y \quad (13)$$

در رابطه‌ی فوق:  $Y \in R^n$  و  $\varphi \in R^{u(r+1) \times n}$ ،  $W \in R^{u(r+1)}$

اگر خروجی مطلوب سیستم  $T = (t_1, t_2, \dots, t_n) \in R^n$  باشد، هدف، تعیین پارامترهای بهینه‌ی  $W^*$  به نحوی است که رابطه‌ی خطی  $\|W\varphi - T\|_2$  حداقل شود با استفاده از روش حداقل مربعات خطی مقدار بهینه‌ی  $W^*$  از رابطه‌ی زیر به دست می‌آید:

$$W^* = T(\varphi^T \varphi)^{-1} \varphi^T. \quad (14)$$

اگرچه در الگوریتم حداقل مربعات خطی، سرعت محاسبات بالاست و نتایج بهینه‌ای به دست می‌دهد اما حساسیت بالایی به نرخ سیگنال به نویز دارد و بسیار نویز پذیر است. برای رفع این مشکل از یک الگوریتم بهبود داده شده در این مقاله استفاده خواهد شد. در الگوریتم حداقل مربعات خطی بهبود داده شده رابطه‌ی (۱۴) به صورت رابطه‌ی زیر بیان می‌شود:





در رابطه‌ی (۱۵)،  $w_j$  فاصله‌ی بین زامین الگو و مراکز نرون‌های RBF موجود  $j = 1, 2, \dots, u$  می‌باشد. برای ساده‌سازی روابط حداقل مربعات بازگشتی وزن‌دار متغیرهای  $P_w$  و  $b_w$  را به صورت زیر در نظر می‌گیریم:

$$W^* = ((\varphi^T W \varphi)^{-1} \varphi^T W T) \quad (15)$$

$$W = \begin{bmatrix} w_1 & 0 & \dots & 0 \\ 0 & w_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & \dots & w_p \end{bmatrix} \quad (16)$$

روابط بازگشتی برای الگوریتم حداقل مربعات بازگشتی به صورت زیر است:

$$\begin{cases} P_w = (\varphi^T W \varphi)^{-1} \\ b_w = P_w \varphi^T W T \end{cases} \quad (17)$$

که در رابطه‌ی فوق،  $a_k^T = [x_{k1} x_{k2} \dots x_{kr}]$  نتایج نشان می‌دهد چون ساختار OSO-NFS بعد از فرآیند یادگیری ساختار بسیار فشرده است. ابعاد  $W$  و  $\varphi$  کوچک است و به کمک روش WRLS با انجام محاسبات بازگشتی می‌توان پارامترهای قسمت تالی را به نحو مناسبی تخمین زد.

$$\begin{cases} b_{k+1} = b_k + w_{k+1} P_{k+1} a_{k+1} (y_{k+1} a_{k+1}^T b_k) \\ P_{k+1} = \frac{1}{\lambda} \left( P_k \frac{w_{k+1} P_k a_{k+1} a_{k+1}^T P_k}{\lambda + a_{k+1}^T P_k a_{k+1}} \right) \end{cases} \quad (18)$$

### ۳-۲-۵- ترکیب توابع عضویت

زمانی که بعضی از توابع عضویت مشابه هستند، می‌توانند در یک گروه قرار بگیرند و با یکدیگر ترکیب شوند. توابع عضویتی که عرض‌های یکسانی دارند با یکدیگر ترکیب می‌شوند و یک تابع عضویت جدید ایجاد می‌شود. اگر  $n$  تعداد توابع عضویتی باشد که دارای عرض‌های یکسان  $C_s$  و مراکز مختلف  $\sigma_{s1}, \sigma_{s2}, \dots, \sigma_{sn}$  هستند، در این صورت در تابع عضویت جدید داریم:

$$C_{new} = C_s \quad (19)$$

$$\Sigma_{new} = \frac{\sigma_{s1} + \sigma_{s2} + \dots + \sigma_{sn}}{n} \quad (20)$$

به دلیل حذف توابع عضویت اضافی، تعداد توابع عضویت و در نتیجه تعداد پارامترها و هم‌چنین پیچیدگی شبکه کاهش پیدا می‌کند. مراحل اصلی الگوریتم پیشنهادی در شکل ۲ آورده شده است. ابتدا پارامترهای سیستم برای شروع فرآیند آموزش مقدردهی اولیه می‌شوند و با ورود اولین الگو، اولین قانون تولید شده و پارامترهای قسمت مقدم و تالی این قانون تنظیم می‌شوند. سپس با ورود الگوهای بعدی، قوانین فازی براساس دو معیار خطا، درجه‌ی سازگاری، تولید می‌گردد. بعد از آن، پارامترهای قسمت مقدم قانون جدید و پارامترهای قسمت تالی توسط الگوریتم WRLS تنظیم می‌شوند. در طی فرآیند آموزش، بعد از مرحله‌ی رشد، برای افزایش کارایی سیستم، کاهش بار محاسباتی و در نتیجه کاهش زمان آموزش، توابع عضویت مشابه باهم ترکیب شده و ساختار به‌روزرسانی می‌شود. سپس براساس معیار چگالی قانون، قوانینی که تاثیر کم‌تری در کارایی سیستم دارند هرس می‌شود. این فرآیند تا زمانی که الگوها کامل شوند تکرار می‌گردد.

### ۴- شبیه‌سازی OSO-NFS

در این بخش، فرآیند شناسایی دو سیستم غیرخطی با درجه‌ی غیرخطی بودن بالا به منظور نمایش کارایی OSO-NFS مورد بررسی قرار گرفته است. برای نمایش عملکرد مناسب سیستم پیشنهادی فرآیند شناسایی در دو حالت با حضور نویز و بدون نویز بررسی شده است و نتایج با الگوریتم‌های معرفی شده در مقالات متعددی مقایسه شده است. در این مقاله برای پیاده‌سازی سیستم پیشنهادی از نرم افزار MATLAB R2011a استفاده شده است. شبیه‌سازی‌ها بر روی CPU پنتیوم ۳/۲ گیگاهرتز با ۴ گیگابایت حجم اجرا شده است. بلاک دیاگرام اصلی در شکل ۳ نشان داده شده است.

مثال ۱. مدل‌سازی یک تابع غیر خطی با ۳ ورودی

یک تابع غیر خطی سه‌بعدی که به‌طور وسیعی در الگوریتم‌های گوناگون (وو و همکاران، ۲۰۰۱؛ ونگ و همکاران، ۲۰۰۹؛ چن و همکاران، ۲۰۱۱) به‌کار می‌رود به‌صورت زیر است:

$$f(x_1, x_2, x_3) = (1 + x_1^{0.5} + x_2^{-1} + x_3^{-1.5})^2. \quad (21)$$

نمونه‌های آموزش شامل ۲۱۶ جفت داده است که به‌طور تصادفی از بازه‌ی ورودی  $[1, 6] \times [1, 6] \times [1, 6]$  انتخاب شده و خروجی مطلوب از رابطه‌ی (۲۱) به‌دست می‌آید. مقادیر اولیه‌ی پارامترها برای آموزش به‌صورت زیر در نظر گرفته می‌شود:

$$\epsilon_{\min} = 0.5, \epsilon_{\max} = 0.8, e_{\min} = 0.03, e_{\max} \quad (22)$$

$$= 0.5, k_{mf} = 0.65, k_s = 0.9, k_{err} = 0.002, d_u = 6$$

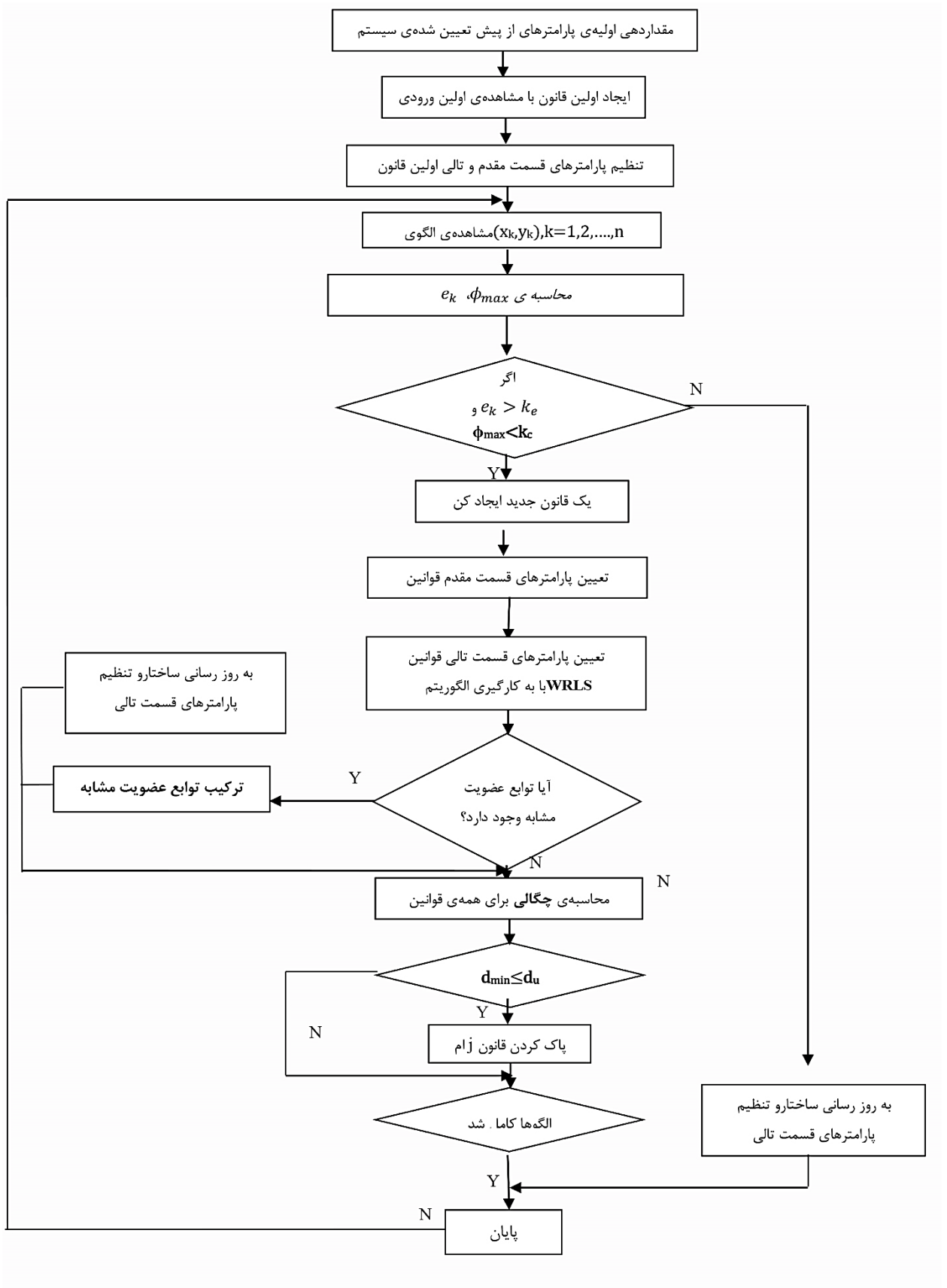
برای مقایسه‌ی کارایی با سایر روش‌ها اندیس کارایی که در (خیاط و همکاران، ۲۰۰۹) استفاده شده است به‌کار گرفته شده است:

$$APE = \frac{1}{n} \sum_{k=1}^n \frac{|t^k - y^k|}{|t^k|}. \quad (23)$$

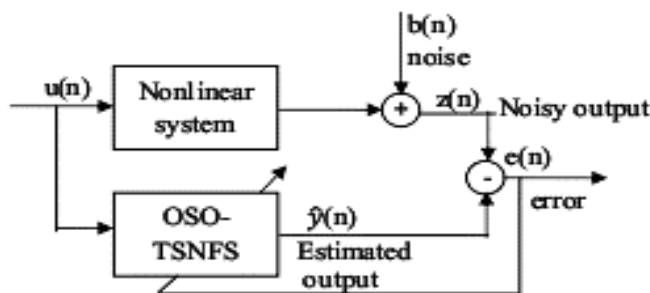
که  $n$  تعداد نمونه‌های آموزش و  $t^k$  خروجی مطلوب  $y^k$  خروجی واقعی شبکه است. ۱۲۵ نمونه به‌صورت تصادفی به‌عنوان داده‌های تست در نظر گرفته شده‌اند. هدف تخمین این تابع، سه ورودی در حضور نویز سفید گوسی و بدون نویز است. برای ارزیابی کارایی سیستم، نتایج شبیه‌سازی در شکل‌های ۳-الف تا ۳-ج آورده شده است.



Archive of SID



شکل ۲- فلوچارت الگوریتم پیشنهادی.



شکل ۳- فرآیند شناسایی سیستم غیرخطی.

باتوجه به این نکته که نتایج به‌دست‌آمده در سایر مقالات بدون در نظر گرفتن نویز می‌باشند، با اضافه کردن نویز گوسی سفید به خروجی سیستم غیرخطی فوق شبیه‌سازی توسط OSO-NFS در دو مرحله (یک بار در حضور نویز و بار دیگر بدون حضور نویز) صورت گرفته است.

مثال ۲. شناسایی یک سیستم غیرخطی ورودی محدود-خروجی محدود

سیستم زیر یکی از مسائل مناسب است که در [۲۸/۲۰/۱۹] به‌کار گرفته شده است که با رابطه‌ی زیر نشان داده می‌شود:

$$y(k) = u(k)^3 + \frac{y(k-1)}{1 + y(k-1)^2} \quad (24)$$

که  $u(k)$  ورودی فعلی و  $y(k)$  و  $y(k-1)$  به‌ترتیب خروجی لحظه‌ی فعلی و قبلی سیستم هستند.  $u(k)$  از رابطه‌ی زیر محاسبه می‌شود:

$$u(k) = \begin{cases} -0.7 + \frac{\text{mod}(k, 5)}{40}, & k \leq 80 \\ \text{rands}(1,1), & 80 < k \leq 130 \\ 0.7 - \frac{\text{mod}(k, 180)}{180}, & 130 < k \leq 250 \\ 0.6 \cos\left(\frac{\pi k}{50}\right), & 250 < k \leq 400 \end{cases} \quad (25)$$

تابع  $\text{mod}(x,y)$  باقیمانده‌ی  $x/y$  است. در این مثال، ورودی  $u(k)$  به‌ازای  $k=1 \dots 400$  به سیستم اعمال می‌شود.

در فاز تست ۴۰۰ داده در بازه‌ی  $k=1$  تا ۴۰۰ به‌صورت زوج ورودی در نظر گرفته شده است.

مقادیر اولیه‌ی پارامترها برای آموزش به‌صورت زیر در نظر گرفته می‌شود:

$$\varepsilon_{\min} = 0.2, \varepsilon_{\max} = 2, e_{\min} = 0.02$$

$$e_{\max} = 0.9, k_{mf} = 0.6, k_s = 0.6, k_{err} = 0.007, d_u = 4$$

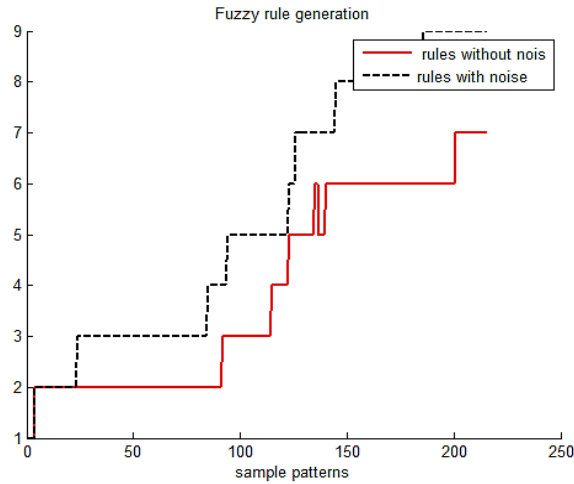
## ۵- ارزیابی نتایج

نتایج شبیه‌سازی در طی فرآیند یادگیری برخط در مورد مثال ۱ در شکل‌های ۴-الف تا ۴-چ نشان داده شده است. نتایج نشان می‌دهند که OSO-NFS می‌تواند زمانی که داده‌های آموزش به‌صورت برخط به سیستم وارد می‌شوند (الگو به الگو) سیستم غیرخطی را درحالی که خطای خروجی و RMSE به‌سمت صفر میل می‌کند، شناسایی کند. برای نمایش برتری عملکرد سیستم پیشنهادی در حضور نویز، نتایج OSO-NFS با دو روش شبیه‌سازی‌شده ANFIS و GEBF-OSFNN مقایسه شده است که در شکل‌های ۴-ج و ۴-د نتایج این شبیه‌سازی‌ها آورده شده است. هم‌چنین نتایج به‌دست‌آمده در جداول ۱ و ۲ و ۳ ارائه شده است.



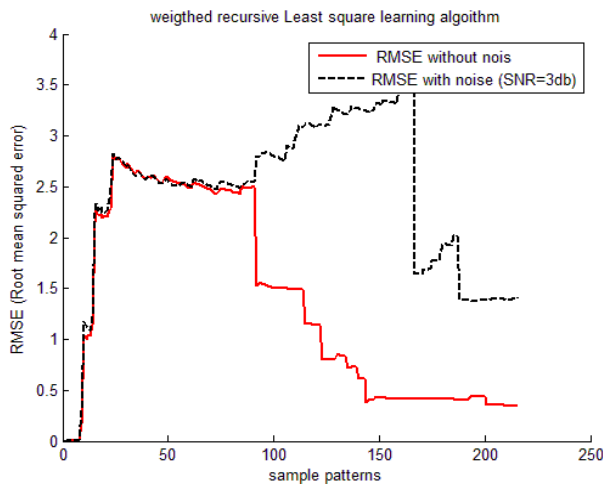
جدول ۱- مقایسه‌ی نتایج OSO-NFS با سایر روش‌ها بدون نویز.

Method	Rule#	Param#	$APE_{\text{trn}}(\%)$	$APE_{\text{chk}}(\%)$
ANFIS (لنگ و همکاران، ۲۰۰۵)	۸	۵۰	۰/۰۰۴۳	۰/۰۰۶۶
OLS (طاووسی و بادام‌چی زاده، ۲۰۱۳)	۲۲	۶۶	۲/۴۳	۲/۵۶
GDFNN (وو و همکاران، ۲۰۰۱)	۱۰	۶۴	۲/۱۱	۱/۵۴
FAOS-PFNN (ونگ و همکاران، ۲۰۰۹)	۷	۳۵	۱/۸۹	۲/۹۸
SOFNN (کاسابو و سانگ، ۲۰۰۲)	۹	۶۰	۱/۱۳۸۰	۱/۱۲۴۴
GEBF-OSFNN (ونگ، ۲۰۱۱)	۹(۸-۱۰)	۷۲	۰/۹۲(۰/۸۴-۱/۱۲)	۰/۸۵(۰/۸۲-۱/۱۰)
OSO-NFS	۷	۴۲	.۸۷	۱/۲۲



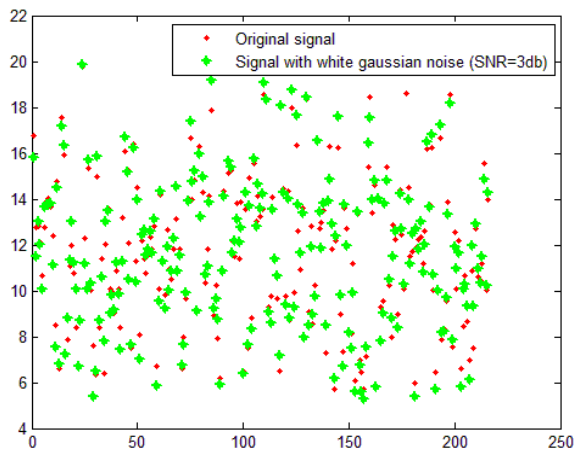
شکل ۴- الف - رشد نرون های قانون OSO-NFS در حضور نویز و بدون نویز.

باتوجه به نمودار ۴- الف واضح است که OSO-NFS تنها توسط هفت قانون، زمانی که نویز وجود ندارد و نه قانون در حضور نویز، سیستم غیر خطی را شناسایی می‌کند.



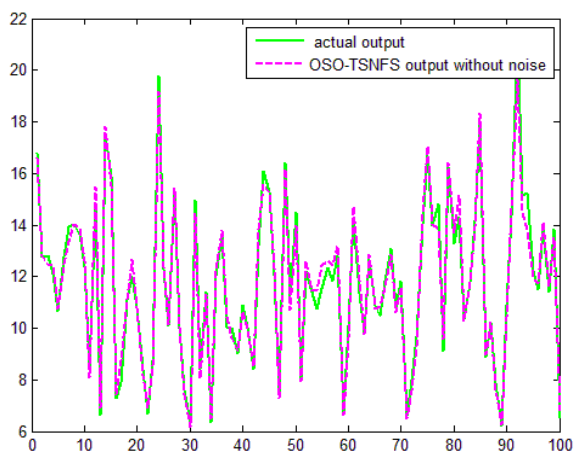
شکل ۴- ب- ریشه‌ی میانگین مربعات خطا (RMSE) در حضور نویز و بدون نویز.

در شکل ۴- ب ریشه‌ی میانگین مربعات خطا در طی فاز آموزش در حضور نویز و بدون نویز نشان داده شده است.



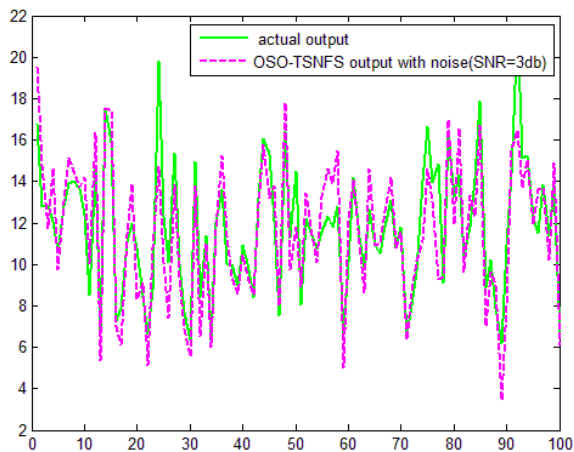
شکل ۴-پ- نمایش سیگنال اصلی و سیگنال اصلی + نویز گوسی سفید.

اعمال نویز سفید گوسی با  $SNR=3db$  و  $noise\ power=0$  مطابق با شکل ۴-پ به خروجی تابع غیر خطی در طی فرآیند شناسایی نمایش داده شده است.



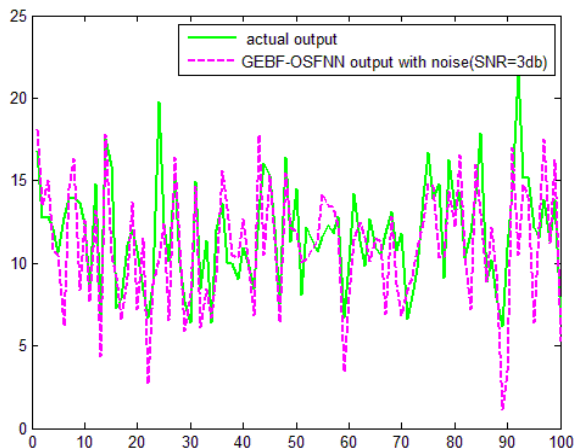
شکل ۴-ت- مقایسه‌ی خروجی واقعی و خروجی OSO-NFS بدون نویز.

خروجی مطلوب و خروجی OSO-NFS در شکل ۴-ت با یکدیگر مقایسه شده است.



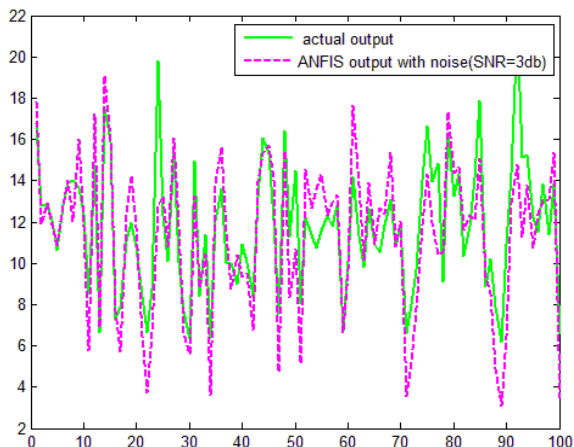
شکل ۴-ث- مقایسه‌ی خروجی واقعی و خروجی OSO-NFS در حضور نویز  $SNR=3db$  و  $noise\ power=0$ .

در شکل ۴-ث نیز مقایسه‌ای بدون حضور نویز انجام گرفته است.



شکل ۴-ج - مقایسه‌ی خروجی واقعی و خروجی GEBF-OSFNN در حضور نویز SNR=3db و noise power=0.

در شکل ۴-ج خروجی مطلوب و خروجی به‌دست آمده توسط روش GEBF-OSFNN در حضور نویز نشان داده شده است.



شکل ۴-ج - مقایسه‌ی خروجی واقعی و خروجی ANFIS در حضور نویز SNR=3db و noise power=0.

در شکل ۴-ج خروجی مطلوب و خروجی به‌دست آمده توسط روش ANFIS در حضور نویز نشان داده شده است.

با بررسی نتایج جدول ۱ واضح است که OSO-NFS سیستم غیرخطی را با بهترین تقریب در مقایسه با سایر الگوریتم‌ها، بدون حضور نویز شناسایی می‌کند. اگرچه در روش FAOS-PFNN با تعداد پارامترهای کم‌تری سیستم تقریب زده شده است، اما APE به‌دست آمده توسط OSO-NFS کم‌تر است که حاکی از کارایی خوب روش پیشنهادی است. در واقع مقادیر عددی نتایج به‌دست آمده در الگوریتم پیشنهادی بدون در نظر گرفتن نویز، از لحاظ ساختار (تعداد قوانین) و از لحاظ دقت نسبت به سایر روش‌ها بهتر است.

در جدول ۲، APE و تعداد قوانین ایجاد شده در OSO-NFS با مقادیر نرخ سیگنال به نویز و توان نویز متفاوت محاسبه شده است. با بررسی نتایج واضح است که هر چه نرخ سیگنال به نویز بیشتر و توان نویز کم‌تر باشد APE خروجی کم‌تر است.





جدول ۲- محاسبه‌ی APE مثال ۱ با مقادیر نرخ سیگنال به نویز و توان نویز مختلف.

Method	Rules#	SNR(db)	N. P (db)	$APE_{\text{trn}}(\%)$	$APE_{\text{chk}}(\%)$
OSO-NFS	۹	۳	۰	۲/۰۰۸۷	۲/۰۱۴۰
	۹	۳	۱	۲/۰۱۲۲	۲/۰۱۷۲
	۹	۳	۲	۲/۰۱۲۶	۲/۰۱۶۸
	۱۰	۲	۰	۲/۰۱۲۳	۲/۰۱۹۲
	۱۰	۲	۱	۲/۰۱۳۶	۲/۰۱۸۶
	۱۱	۲	۲	۲/۰۱۷۰	۲/۰۲۲۸
	۹	۱	۰	۲/۰۱۴۷	۲/۰۱۴۷
	۱۱	۱	۱	۲/۰۱۶۶	۲/۰۱۸۲
	۱۱	۱	۲	۲/۰۱۷۹	۲/۰۲۴۶

در جدول ۳ میزان کارایی و تعداد قوانین در سه روش GEBF-OSFNN، ANFIS و OSO-NFS در حضور نویز با مقادیر نرخ سیگنال به نویز و توان نویز مختلف به دست آمده است. بررسی نتایج نشان می‌دهد که در حضور نویز، سیستم پیشنهادی عملکرد بهتری دارد. به دلیل استفاده از توابع عضویت نامتقارن برای افزایش انعطاف پذیری و قابلیت تفسیر بیش‌تر در GEBF-OSFNN، تعداد پارامترها بیش‌تر است که این امر منجر به زمان آموزش طولانی‌تر می‌گردد. در ANFIS به دلیل استفاده از روش‌های خوشه‌بندی استاتیک، تعداد قوانین به‌ازای مقادیر مختلف سیگنال به نویز و توان نویز، ثابت است و هم‌چنین قابلیت یادگیری برخط در آن وجود ندارد. به‌طور کلی نتایج نشان می‌دهد که OSO-TSNFS در مقایسه با سایر روش‌ها در حضور نویز و بدون حضور نویز عملکرد مناسب‌تری دارد.

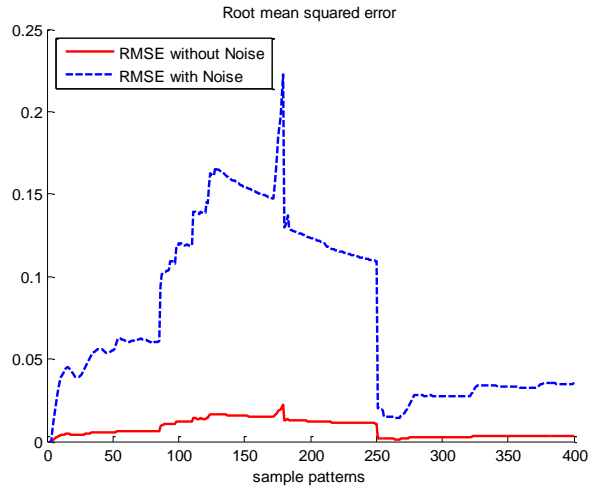
جدول ۳- مقایسه‌ی نتایج OSO-NFS مثال ۱ با روش ANFIS و GEBF-OSFNN در حضور نویز با مقادیر نرخ سیگنال به نویز و توان نویز مختلف.

Modeling a Multi-Dimensional Nonlinear Function(example 1)					
Method	Rules #	SN(db), noise power(db)	$APE_{\text{trn}}(\%)$	$APE_{\text{chk}}(\%)$	
OSO-NFS	۹	۳/۱	۲/۰۱۲۲	۲/۰۱۷۲	
	۹	۳/۲	۲/۰۱۲۶	۲/۰۱۶۸	
	۱۰	۲/۱	۲/۰۱۳۶	۲/۰۱۸۶	
	۱۱	۲/۲	۲/۰۱۷۰	۲/۰۲۲۸	
	۱۲	۳/۱	۳/۰۲۵۰	۳/۰۳۱۹	
ANFIS	۱۲	۳/۲	۳/۰۲۸۱	۳/۰۳۶۳	
	۱۲	۲/۱	۳/۰۳۹۱	۳/۰۴۹۲	
	۱۲	۲/۲	۳/۰۴۳۲	۳/۰۵۲۱	
	۹	۳/۱	۲/۵۱۸۶	۲/۵۲۴۱	
GEBF-OSFNN (د جسو رایبو، ۲۰۰۹)	۱۰	۳/۲	۲/۵۲۰۳	۲/۵۲۶۵	
	۱۱	۲/۱	۲/۵۲۶۳	۲/۵۳۳۹	
	۱۱	۲/۲	۲/۵۳۰۱	۲/۵۳۷۴	

نتایج شبیه‌سازی مثال دو در شکل های (۵-الف تا ۵-ج) نشان داده شده است. مقایسه‌ی نتایج بین OSO-NFS و سایر الگوریتم‌های معروف در جداول ۴ و ۵ آورده شده است.

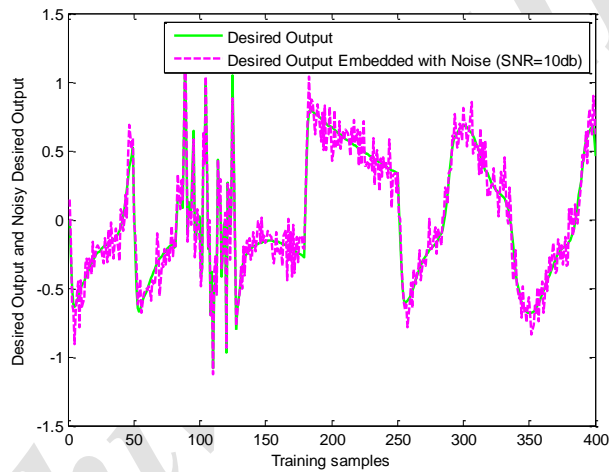
نتایج نشان می‌دهند که سیستم پیشنهادی در شناسایی سیستم‌های غیرخطی با درجه‌ی غیرخطی بودن بالا نسبت به شبکه‌های عصبی و شبکه‌های فازی-عصبی خودسازمانده نتایج قابل قبول‌تری ارائه می‌دهند. سیستم پیشنهادی با ساختار فشرده‌تر و تعداد قوانین کم‌تر توانسته است با خطای کم‌تری سیستم غیرخطی را شناسایی کند.





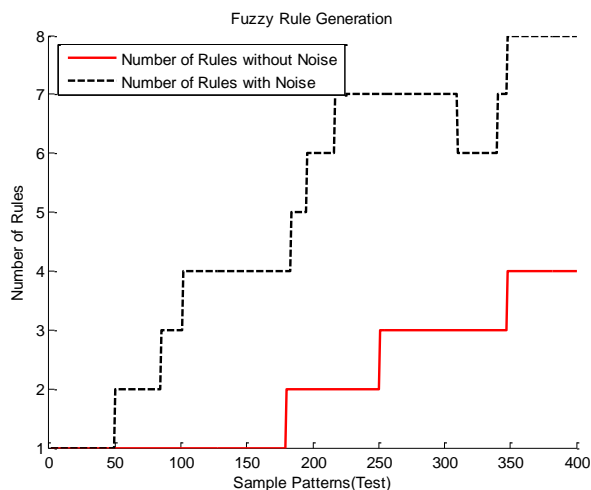
شکل ۵-الف- ریشه‌ی میانگین مربعات خطا (RMSE) در حضور نویز و بدون نویز.

در شکل ۵-الف ریشه‌ی میانگین مربعات خطا در طی فاز آموزش در حضور نویز و بدون نویز نشان داده شده است.



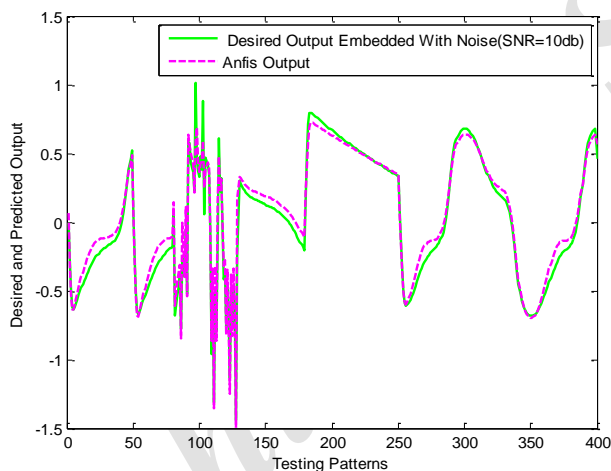
شکل ۵-ب- نمایش سیگنال اصلی و سیگنال اصلی + نویز گوسی سفید.

اعمال نویز سفید گوسی با  $\text{SNR}=10\text{db}$  و  $\text{noise power}=0$  مطابق با شکل ۵-ب به خروجی تابع غیرخطی در طی فرآیند شناسایی نمایش داده شده است.



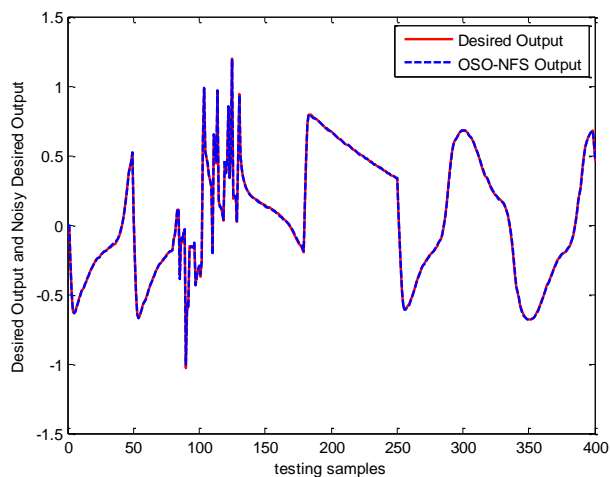
شکل ۵-پ- رشد نرون‌های قانون OSO-NFS در حضور نویز بدون نویز.

باتوجه به نمودار ۵-پ واضح است که OSO-NFS تنها توسط ۴ قانون زمانی که نویز وجود ندارد و ۸ قانون در حضور نویز سیستم غیرخطی را شناسایی می‌کند.



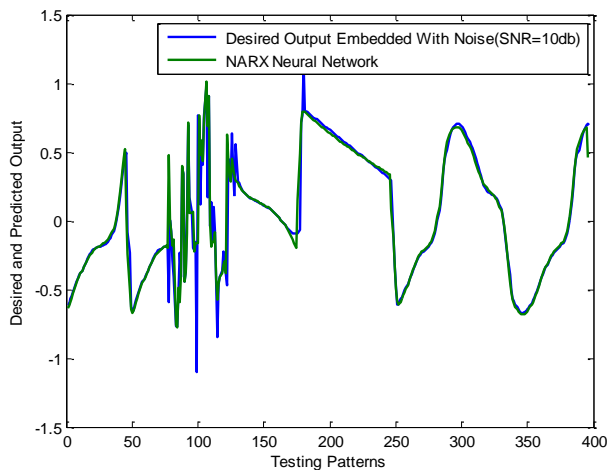
شکل ۵-ت- مقایسه‌ی خروجی واقعی و خروجی ANFIS در حضور نویز SNR=10db و noise power=0.

در شکل ۵-ت خروجی مطلوب و خروجی به‌دست‌آمده توسط روش ANFIS در حضور نویز نشان داده شده است.



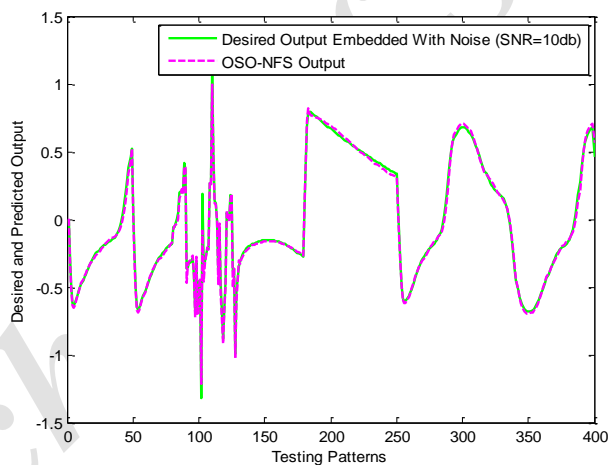
شکل ۵-ث- مقایسه‌ی خروجی واقعی و خروجی OSO-NFS بدون نویز.

خروجی مطلوب و خروجی OSO-NFS در شکل ۵-۵ با یکدیگر در حالت بدون نویز مقایسه شده است.



شکل ۵-ج- مقایسه‌ی خروجی واقعی و خروجی شبکه‌ی عصبی NARX حضور نویز  $SNR=10db$  و  $noise\ power=0$ .

در شکل ۵-ج خروجی مطلوب و خروجی به‌دست آمده توسط روش NARX در حضور نویز نشان داده شده است.



شکل ۵-چ- مقایسه‌ی خروجی واقعی و خروجی OSO-NFS در حضور نویز  $SNR=10db$  و  $noise\ power=0$ .

خروجی مطلوب و خروجی OSO-NFS در شکل ۵-چ با یکدیگر در حالت نویزی مقایسه شده است.

جدول ۴- مقایسه‌ی نتایج OSO-NFS مثال ۲ با سایر روش‌ها بدون نویز.

Method	سیستم غیرخطی ورودی محدود-خروجی محدود	
	RMSE	No. of final RBF unit
T2RBFN (طاووسی و همکاران، ۲۰۱۶)	۰/۰۰۳۲	۲
NARX Neural Network	۰/۰۳۹۶	۱۰
ANFIS	۰/۰۲۰۷	۵
OSO-NFS(without noise)	۰/۰۰۲۴	۴



نتایج جدول ۴ نشان می‌دهند شبکه‌های فازی-عصبی در مقایسه با شبکه‌های عصبی چندلایه NARX قدرت بیش‌تری در شناسایی سیستم‌های غیرخطی دارند و همان‌طور که آشکار است نسبت به ANFIS که یک شبکه‌ی فازی عصبی با ساختار ثابت است شبکه‌های فازی-عصبی خودسازمانده با ساختار فشرده‌تر و میزان خطای کم‌تر توانایی مدل‌سازی سیستم‌های غیرخطی را دارا می‌باشند.

جدول ۵- مقایسه‌ی نتایج OSO-NFS مثال ۲ روش ANFIS و NARX در حضور نویز با مقادیر مختلف نرخ سیگنال به نویز و توان نویز.

Method	سیستم غیرخطی ورودی محدود-خروجی محدود		
	Rules #	SNR(db), noise power(db)	RMSE
OSO-NFS	۹	۲۵/۱	۰/۱۸۱۲
	۱۰	۲۵/۲	۰/۱۹۹۸
	۸	۲۰/۱	۰/۱۹۲۵
	۹	۲۰/۲	۰/۲۰۰۱
NARX	۱۶	۲۵/۱	۰/۲۵۴۵
	۱۶	۲۵/۲	۰/۲۶۷۱
	۱۶	۲۰/۱	۰/۲۹۹۸
	۱۶	۲۰/۲	۰/۳۰۴۵
ANFIS	۱۲	۲۵/۱	۰/۲۱۷۸
	۱۱	۲۵/۲	۰/۲۳۳۴
	۱۳	۲۰/۱	۰/۲۴۵۷
	۱۴	۲۰/۲	۰/۲۵۳۲

بررسی نتایج جدول ۵ نشان می‌دهد که در حضور نویز، سیستم پیشنهادی عملکرد بهتری دارد.

## ۶- نتیجه‌گیری و پیشنهادات آینده

در این مقاله یک سیستم فازی-عصبی خودسازمانده با قابلیت یادگیری برخط برای شناسایی سیستم‌های دینامیک غیرخطی در حضور نویز ارائه گردید. ابتدا در این سیستم هیچ نودی در لایه‌ی پنهان وجود ندارد و در طی فرآیند آموزش چنانچه معیارهای تولید قوانین برآورده شوند نرون RBF به لایه‌ی پنهان اضافه می‌شود. به‌طورکلی نتایج نشان می‌دهند که در ANFIS به دلیل استفاده از روش‌های خوشه‌بندی استاتیک تعداد قوانین به‌ازای مقادیر مختلف سیگنال به نویز و توان نویز، ثابت است و هم‌چنین قابلیت یادگیری برخط در آن وجود ندارد. در OSO-NFS پارامترهای قانون جدید به‌فرم تاکاگی-سوگنو با استفاده از الگوریتم آموزش WRLS برای قابلیت یادگیری برخط و افزایش سرعت هم‌گرایی، تخمین زده می‌شوند. بعداز فرآیند ایجاد قانون جدید، کارایی سیستم محاسبه شده و قوانینی که تأثیر کم‌تری در کارایی سیستم دارند هرس می‌شوند. ویژگی‌های اصلی این مقاله به این شرح می‌باشد. استفاده از معیار جدید درجه‌ی تطبیق و معیار متداول خطا در فاز رشد قوانین و ارائه‌ی یک الگوریتم هرس جدید براساس چگالی، که چگالی تعداد دفعاتی است که یک قانون آتش می‌شود. هر بار که یک الگو توسط یک قانون پوشش داده می‌شود به چگالی آن قانون یک واحد اضافه می‌شود. در پایان، قانونی که کم‌ترین مقدار چگالی را داشته باشد از بین قوانین موجود هرس می‌شود. ترکیب توابع عضویت مشابه و تغییر عرض آن‌ها به منظور افزایش کارایی سیستم و ایجاد قوانین فازی بدون استفاده از الگوریتم پس‌انتشار خطا، برای بررسی عملکرد سیستم، دو سیستم دینامیک غیرخطی در دو حالت نویزی و بدون نویز مدل‌سازی شده‌اند. دقت این مدل‌سازی بر مبنای دو معیار تعداد نرون‌ها (قوانین) و ریشه‌ی میانگین مربعات خطا با سایر روش‌ها مقایسه گردیده است. باتوجه به مطالب عنوان‌شده در این مقاله می‌توان نتایج را مطابق با جداولی که برای دو حالت نویزی و بدون نویز، در جداول زیر دسته‌بندی نمود. در این جداول درصد بهبود/درصد کاهش جواب‌ها نسبت به یک روش مبنا که در هر مساله مشخص شده است به‌طور جداگانه برای هر یک از مسائل مطرح‌شده ارائه شده است.

جدول ۶- درصد بهبود جواب‌ها در مسأله‌ی شناسایی سیستم غیرخطی بدون نویز (روش مبنا GDFNN).

Method	Percentage Improvement on Nonlinear System Identification (without Noise)		
	Testing APE	Rules #	Parameters #
ANFIS (جنگ، ۱۹۹۳)	%۴۴	%۲۵	%۲۸
OLS (منگ، ۲۰۱۷)	%۳۹	-%۵۴	-%۳
GDFNN (وو و همکاران، ۲۰۰۱)	---	---	---
FAOS-PFNN (ونگ و همکاران، ۲۰۰۹)	-%۹۱	%۴۲	%۸۲
SOFNN (لنگ و همکاران، ۲۰۰۵)	%۳۷/۵	%۱۱	%۶
GEBF-OSFNN (ونگ، ۲۰۱۱)	%۸۱	%۱۱	%۱۲/۵
OSO-NFS	%۲۶	%۴۲	%۵۲

همان‌طور که از جدول ۶ مشخص می‌شود اگرچه روش GEBF-OSFNN با درصد قابل توجهی توانسته نسبت به سایر روش‌ها RMSE را بر روی داده‌های تست کاهش دهد، اما همان‌طور که نتایج جدول نشان می‌دهند در روش GEBF-OSFNN تعداد پارامترها و قوانین نسبت به روش پیشنهادی بیش‌تر است که این امر منجر به افزایش زمان آموزش می‌شود.

جدول ۷- درصد بهبود جواب‌ها در مسأله‌ی شناسایی سیستم غیرخطی در حضور نویز (روش مبنا ANFIS).

Method (ANFIS (base))	Percentage Improvement on Nonlinear System Identification (with Noise)	
	Testing RMSE	Rules #
GEBF-OSFNN	-%۵۰	%۲/۸
Proposed OSO-NFS	%۱۲/۵	%۴۲/۷

نتایج جدول ۷ نشان می‌دهند که در حضور نویز، روش پیشنهادی ۱۲/۵٪ نسبت به روش ANFIS خطای کم‌تری بر روی داده‌های تست داشته است. نکته مهم‌تر این است که ساختار سیستم پیشنهادی ۴۲/۷٪ فشرده‌تر از روش ANFIS می‌باشد که این امر منجر به انجام سادگی محاسبات و کاهش زمان آموزش سیستم می‌شود.

جدول ۸- درصد بهبود جواب‌ها در مسأله‌ی سیستم غیرخطی ورودی محدود-خروجی محدود در حضور نویز (روش مبنا ANFIS).

Method (ANFIS (base))	Percentage Improvement on Nonlinear System Identification (without Noise)	
	Testing RMSE	Rules #
T2RBFN (نوئن و همکاران، ۲۰۱۵)	%۵۴۶	%۱۵۰
NARX Neural Network	-%۴۷	%۵۰
Proposed OSO-NFS	%۷۶۲	%۲۵

باتوجه به نتایج به‌دست آمده در جدول ۸ سیستم پیشنهادی برای شناسایی سیستم‌های غیرخطی با درجه‌ی غیرقطعیت بالا کارایی بالاتری نسبت به شبکه‌های عصبی و شبکه‌های فازی-عصبی غیرخودسازمانده داراست و با ساختار فشرده‌تر و میزان خطای کم‌تری توانسته است سیستم را شناسایی کند.

جدول ۹- درصد بهبود جواب‌ها در مسأله‌ی سیستم غیرخطی ورودی محدود-خروجی محدود با حضور نویز (روش مبنا ANFIS).

Method (ANFIS (base))	Percentage Improvement on Nonlinear System Identification (without Noise)	
	Testing RMSE	Rules #
NARX Neural Network	-%۱۴	-%۲۵
Proposed OSO-NFS	%۲۰	%۳۳

نتایج جدول ۹ نیز نشان می‌دهند که در حضور نویز، سیستم پیشنهادی عملکرد بهتری در شناسایی سیستم‌های غیرخطی با درجه‌ی غیرقطعیت بالا دارد و نسبت به روش ANFIS، ۲۰٪ خطای کم‌تری بر روی داده‌های تست داشته است و با ساختار فشرده‌تری در حدود ۳۳٪ سیستم غیرخطی را شناسایی کرده است.



در پایان با مشاهده نتایج می‌توان پیشنهاداتی به شرح زیر برای انجام پژوهش‌های آینده برای بهبود عملکرد سیستم‌های فازی-عصبی خودسازمانده و به‌کارگیری چنین سیستم‌هایی در کاربردهای عملی مطرح نمود:

- نوآوری‌هایی در زمینه‌ی کاهش زمان آموزش شبکه‌های فازی-عصبی خودسازمانده که در کاربردهای بلادرنگ بسیار حائز اهمیت است.
- طراحی شبکه‌های فازی-عصبی خودسازمانده بازگشتی برای فرآیندهای دینامیکی که یک وابستگی زمانی بین ورودی‌ها و خروجی‌های زمان گذشته سیستم وجود دارد.
- استفاده از شبکه‌های فازی-عصبی خودسازمانده با توجه بانک به دقت بالا در پیش‌بینی در مسائلی مانند پیش‌بینی بازار بورس، پیش‌بینی زلزله، پیش‌بینی‌های هواشناسی، و اعتبارسنجی مشتریان.

## پیوست

جدول ۱۰- معرفی متغیرها و پارامترها.

$u$	تعداد قوانین فازی در لایه‌ی پنهان	$\mu_{ij}$	تابع عضویت ورودی $u$ ام قانون $j$ ام
$r$	تعداد متغیرهای ورودی	$\varphi_j$	درجه‌ی آتش شدن قانون $j$ ام
$c$	مراکز توابع عضویت گوسی	$\gamma$	خروجی سیستم
$\sigma$	عرض توابع عضویت گوسی	$w$	وزن‌های شبکه‌ی فازی-عصبی
$x_i$	الگوی ورودی $i$ ام	$\alpha_{ij}$	پارامترهای قسمت تالی قوانین
$A_{ij}$	مجموعه‌های فازی قسمت مقدم قوانین	$z$	اندیس زامین قانون
$i$	اندیس ورودی $i$ ام	$t$	خروجی واقعی سیستم
$md^k(j)$	فاصله مایه‌الانوبیس بین ورودی $k$ ام و مراکز واحدهای RBF	$\gamma$	ثابت کاهش

## منابع

ریخته‌گر مشهد، ش؛ اکبرزاده توتونچی، م. ا. (۱۳۹۲). طراحی یک سیستم فازی عصبی خودسازمانده برای شناسایی سیستم‌های دینامیک غیرخطی در حضور نویز. بیست و یکمین کنفرانس مهندسی برق ایران. مشهد، دانشگاه فردوسی مشهد.

- Wu, S., & Er, M. J. (2000). Dynamic fuzzy neural networks-a novel approach to function approximation. *IEEE transactions on systems, man, and cybernetics, part B (cybernetics)*, 30(2), 358-364.
- Wu, S., Er, M. J., & Gao, Y. (2001). A fast approach for automatic generation of fuzzy rules by generalized dynamic fuzzy neural networks. *IEEE transactions on fuzzy systems*, 9(4), 578-594.
- Wang, N., Er, M. J., & Meng, X. (2009). A fast and accurate online self-organizing scheme for parsimonious fuzzy neural networks. *Neurocomputing*, 72(16-18), 3818-3829.
- Wang, N. (2011). A generalized ellipsoidal basis function based online self-constructing fuzzy neural network. *Neural processing letters*, 34(1), 13-37.
- de Jesús Rubio, J. (2009). SOFMLS: online self-organizing fuzzy modified least-squares network. *IEEE transactions on fuzzy systems*, 17(6), 1296-1309.
- Han, H., & Qiao, J. (2010). A self-organizing fuzzy neural network based on a growing-and-pruning algorithm. *IEEE transactions on fuzzy systems*, 18(6), 1129-1143.
- Kao, C. H., Hsu, C. F., & Don, H. S. (2012). Design of an adaptive self-organizing fuzzy neural network controller for uncertain nonlinear chaotic systems. *Neural computing and applications*, 21(6), 1243-1253.
- Hsu, C. F. (2012). Intelligent tracking control of a DC motor driver using self-organizing TSK-type fuzzy neural networks. *Nonlinear dynamics*, 67(1), 587-600.
- Chen, C. S. (2011). Robust self-organizing neural-fuzzy control with uncertainty observer for MIMO nonlinear systems. *IEEE transactions on fuzzy systems*, 19(4), 694-706.
- Leng, G., McGinnity, T. M., & Prasad, G. (2006). Design for self-organizing fuzzy neural networks based on genetic algorithms. *IEEE transactions on fuzzy systems*, 14(6), 755-766.
- Alcalá-Fdez, J., Alcalá, R., Gacto, M. J., & Herrera, F. (2009). Learning the membership function contexts for mining fuzzy association rules by using genetic algorithms. *Fuzzy Sets and Systems*, 160(7), 905-921.
- Khayat, O., Ebadzadeh, M. M., Shahdoosti, H. R., Rajaei, R., & Khajenasiri, I. (2009). A novel hybrid algorithm for creating self-organizing fuzzy neural networks. *Neurocomputing*, 73(1-3), 517-524.
- Chen, C. H., Lin, C. J., & Liao, Y. Y. (2011). A rule-based symbiotic modified differential evolution for self-organizing neuro-fuzzy systems. *Proceedings 2011 international conference on system science and engineering* (pp. 165-170). IEEE.



- Lin, S. F., Chang, J. W., & Hsu, Y. C. (2012). A self-organization mining based hybrid evolution learning for TSK-type fuzzy model design. *Applied intelligence*, 36(2), 454-471.
- Juang, C. F., Chiu, S. H., & Chang, S. W. (2007). A self-organizing TS-type fuzzy network with support vector learning and its application to classification problems. *IEEE transactions on fuzzy systems*, 15(5), 998-1008.
- Juang, C. F., & Shiu, S. J. (2008). Using self-organizing fuzzy network with support vector learning for face detection in color images. *Neurocomputing*, 71(16-18), 3409-3420.
- Dahal, K., Almejalli, K., Hossain, M. A., & Chen, W. (2015). GA-based learning for rule identification in fuzzy neural networks. *Applied soft computing*, 35, 605-617.
- Nguyen, N. N., Zhou, W. J., & Quek, C. (2015). GSETSK: a generic self-evolving TSK fuzzy neural network with a novel Hebbian-based rule reduction approach. *Applied soft computing*, 35, 29-42.
- Tavoosi, J., Suratgar, A. A., & Menhaj, M. B. (2016). Nonlinear system identification based on a self-organizing type-2 fuzzy RBFN. *Engineering applications of artificial intelligence*, 54, 26-38.
- Han, H. G., Lin, Z. L., & Qiao, J. F. (2017). Modeling of nonlinear systems using the self-organizing fuzzy neural network with adaptive gradient algorithm. *Neurocomputing*, 266, 566-578.
- Leung, J. H., Kuo, Y. L., Weng, T. W., & Chin, C. L. (2017). Hybrid-Neuro-Fuzzy System and Adaboost-Classifier for Classifying Breast Calcification. *Journal of Computers*, 28(2), 29-42.
- Lin, C. M., & Le, T. L. (2017). PSO-self-organizing interval type-2 fuzzy neural network for antilock braking systems. *International journal of fuzzy systems*, 19(5), 1362-1374.
- Han, H., Wu, X. L., & Qiao, J. F. (2013). Nonlinear systems modeling based on self-organizing fuzzy-neural-network with adaptive computation algorithm. *IEEE transactions on cybernetics*, 44(4), 554-564.
- Han, H. G., Guo, Y. N., & Qiao, J. F. (2018). Nonlinear system modeling using a self-organizing recurrent radial basis function neural network. *Applied soft computing*, 71, 1105-1116.
- Meng, X., Rozycki, P., Qiao, J. F., & Wilamowski, B. M. (2017). Nonlinear system modeling using RBF networks for industrial application. *IEEE transactions on industrial informatics*, 14(3), 931-940.
- Kumar, R., Srivastava, S., & Gupta, J. R. P. (2018). Online modeling and adaptive control of robotic manipulators using Gaussian radial basis function networks. *Neural computing and applications*, 30(1), 223-239.
- Tavoosi, J., & Badamchizadeh, M. A. (2013). A class of type-2 fuzzy neural networks for nonlinear dynamical system identification. *Neural computing and applications*, 23(3-4), 707-717.
- Kasabov, N., & Song, Q. (2002). DENFIS: dynamic evolving neural-fuzzy inference system and its application for time series prediction. *IEEE transactions on fuzzy systems*, 10(2).
- Leng, G., McGinnity, T. M., & Prasad, G. (2005). An approach for on-line extraction of fuzzy rules using a self-organising fuzzy neural network. *Fuzzy sets and systems*, 150(2), 211-243.
- Leng, G., McGinnity, T. M., & Prasad, G. (2005). An approach for on-line extraction of fuzzy rules using a self-organising fuzzy neural network. *Fuzzy sets and systems*, 150(2), 211-243.
- Jang, J. S. (1993). ANFIS: adaptive-network-based fuzzy inference system. *IEEE transactions on systems, man, and cybernetics*, 23(3), 665-685.

Archive