

پایان نامه جهت دریافت درجه کارشناسی ارشد " M.SC "

مهندسی برق-کنترل

عنوان:

کاربرد فیلتر کالمن عصبی در بهبود سیگنالهای صوتی

عنوان:

کاربرد فیلتر کالمن عصبی در بهبود سیگنالهای صوتی

تقدیم به

همسر مهربان

و

پسر عزیزم

تقدیر و تشکر

انجام این پایان نامه مرهون زحمات بی دریغ استادی گرانقدر است که در تمام مراحل این پروژه، دوستانه و صبورانه راهنما و مشوق من بود. لذا بدینوسیله از زحمات بی شائبه استاد عزیزم

فهرست مطالب

<u>صفحه</u>	<u>عنوان</u>
۱۲	چکیده
۱۳	مقدمه

فصل ۱

مروری بر پژوهشهای پیشین

۱۴	۱-۱- فیلتر وینر
۱۵	۲-۱- فیلتر کالمن
۱۶	۳-۱- فیلتر کالمن توسعه یافته
۱۷	۴-۱- روشهای کاهش طیفی
۱۸	۵-۱- استفاده مستقیم از شبکه عصبی
۱۹	۶-۱- فیلتر کالمن توسعه یافته دوگانه
۲۰	۷-۱- فیلتر کالمن درک نشده
۲۱	۸-۱- فیلتر کالمن درک نشده دوگانه

فصل ۲

معرفی فیلترهای کالمن

۲۵	۱-۲- مقدمه
۲۶	۲-۲- فیلتر کالمن
۲۶	۱-۲-۲- مدل سیگنال و مشاهده
۲۹	۲-۲-۲- یافتن پارامترهای فیلتر کالمن
۳۱	۳-۲-۲- خلاصه ای از روابط فیلتر کالمن
۳۲	۴-۲-۲- خطای سیستم

- ۳۳ ۵-۲-۲- خصوصیات فیلتر کالمن
- ۳۴ ۳-۲- فیلتر کالمن توسعه یافته (EKF)
- ۳۹ ۴-۲- فیلتر کالمن توسعه یافته دوگانه (DEKF)
- ۴۱ ۱-۴-۲- فیلتر تخمین حالت کالمن توسعه یافته
- ۴۴ ۲-۴-۲- فیلتر تخمین وزن کالمن توسعه یافته

فصل ۳

شناسایی سیستم‌های خطی و غیر خطی

- ۴۵ ۱-۳- مقدمه
- ۴۷ ۲-۳- ساختارهای مدل کردن سیستم‌های خطی
- ۵۲ ۳-۳- مدل فضای حالت سیگنال‌های صوتی
- ۵۳ ۴-۳- اعتبار تخمین
- ۵۴ ۱-۴-۳- تست تابع خود همبستگی
- ۵۶ ۵-۳- شناسایی و مدل کردن سیستم‌های غیر خطی
- ۵۷ ۱-۵-۳- روش NARX
- ۶۱ ۲-۵-۳- روش NAR
- ۶۳ ۳-۵-۳- نمایش شبیه سازیهای مدلسازی صوت

فصل ۴

نمایش شبیه سازیهای انجام شده و نتایج آن

- ۶۷ ۱-۴- مقدمه
- ۶۸ ۲-۴- شرح خلاصه مطالب
- ۶۹ ۳-۴- ساختار مدل
- ۷۰ ۱-۳-۴- حلقه شناسایی سیستم

۷۲	۲-۳-۴- ساختار شبکه عصبی
۷۳	۳-۳-۴- نحوه آموزش شبکه عصبی
۸۷	۴-۳-۴- نحوه همگرا شدن شبکه عصبی
۸۹	۴-۴- موضوع تخمین
۹۲	۴-۴-۱- نحوه تخمین سیگنال صوتی
۱۰۰	۴-۵- نتیجه گیری
۱۰۱	۴-۶- پیشنهاد برای کارهای آینده

صفحه	فهرست نمودارها
۶۴	۳-۵-۷- نمودار مقایسه خروجی شبکه (نمودار آبی) با Target (نمودار قرمز)
۶۴	۳-۵-۸- نمودار مقایسه خروجی شبکه (نمودار آبی) با Target (نمودار قرمز) در ۵۰۰۰ تکرار اول
۶۵	۳-۵-۹- نمودار مقایسه خروجی شبکه (نمودار آبی) با Target (نمودار قرمز) در ۵۰۰۰ تکرار چهارم
۶۵	۳-۵-۱۰- نمودار مقایسه خروجی شبکه (نمودار آبی) با Target (نمودار قرمز) در ۵۰۰۰ تکرار ششم
۶۵	۳-۵-۱۱- نمودار مقایسه خروجی شبکه (نمودار آبی) با Target (نمودار قرمز) در ۵۰۰۰ تکرار آخر
۶۶	۳-۵-۱۲- نمودار نمایش خطای آموزش شبکه به همراه خروجی شبکه
۷۷	۴-۶- نمودار مقایسه خروجی شبکه (نمودار آبی) با سیگنال هدف (نمودار قرمز)
۷۸	۴-۷- نمودار نمایش خطای یادگیری به همراه خروجی شبکه
۷۹	۴-۸- نمودار مقایسه خروجی شبکه (نمودار آبی) با سیگنال هدف (نمودار قرمز)
۷۹	۴-۹- نمودار نمایش خطای آموزش به همراه خروجی شبکه
۸۰	۴-۱۰- نمودار مقایسه خروجی شبکه (نمودار آبی) با سیگنال هدف (نمودار قرمز)
۸۱	۴-۱۱- نمودار نمایش خطای Training به همراه خروجی شبکه
۸۲	۴-۱۲- نمودار مقایسه خروجی شبکه (نمودار آبی) با سیگنال هدف (نمودار قرمز)
۸۲	۴-۱۳- نمودار نمایش خطای یادگیری به همراه خروجی شبکه
۸۳	۴-۱۴- نمودار مقایسه خروجی شبکه (نمودار آبی) با سیگنال هدف (نمودار قرمز)
۸۳	۴-۱۵- نمودار نمایش خطای یادگیری به همراه خروجی شبکه
۸۴	۴-۱۶- نمودار مقایسه خروجی شبکه (نمودار آبی) با Target (نمودار قرمز)
۸۵	۴-۱۷- نمودار نمایش خطای Training به همراه خروجی شبکه
۸۶	۴-۱۸- نمودار مقایسه خروجی شبکه (نمودار آبی) با سیگنال هدف (نمودار قرمز)
۸۶	۴-۱۹- نمودار نمایش خطای آموزش به همراه خروجی شبکه
۸۷	۴-۲۰- نمودار مقایسه خروجی شبکه (نمودار آبی) با سیگنال هدف (نمودار قرمز)
۸۸	۴-۲۱- نمودار مقایسه خروجی شبکه (نمودار آبی) با سیگنال هدف (نمودار قرمز) در ۳۰۰ تکرار ابتدایی
۸۸	۴-۲۲- نمودار مقایسه خروجی شبکه (نمودار آبی) با سیگنال هدف (نمودار قرمز) در ۲۰۰ تکرار میانی

۸۸	۲۳-۴- نمودار مقایسه خروجی شبکه (نمودار آبی) با سیگنال هدف (نمودار قرمز) در ۳۰۰ تکرار انتهایی
۸۹	۲۴-۴- نمودار نمایش خطای آموزش به همراه خروجی شبکه
۹۴	۲۵-۴- نمودار مقایسه سیگنال تمیز (شکل اول) سیگنال نویزی (شکل دوم) و سیگنال تخمینی (شکل سوم)
۹۴	۲۶-۴- نمودار مقایسه سیگنال تمیز (نمودار قرمز) و سیگنال تخمینی (نمودار آبی)
۹۵	۲۷-۴- نمودار مقایسه دقیق سیگنال تمیز (نمودار قرمز) و سیگنال تخمینی (نمودار آبی) در ۵۰۰ تکرار
۹۵	۲۸-۴- نمودار مقایسه دقیق سیگنال تمیز (نمودار قرمز) و سیگنال تخمینی (نمودار آبی) در ۵۰۰ تکرار
۹۵	۲۹-۴- نمودار مقایسه دقیق سیگنال تمیز (نمودار قرمز) و سیگنال تخمینی (نمودار آبی) در ۵۰۰ تکرار
۹۶	۳۰-۴- نمودار مقایسه دقیق سیگنال تمیز (نمودار قرمز) و سیگنال تخمینی (نمودار آبی) در ۵۰۰ تکرار
۹۶	۳۱-۴- نمودار مقایسه دقیق سیگنال تمیز (نمودار قرمز) و سیگنال تخمینی (نمودار آبی) در ۵۰۰ تکرار
۹۶	۳۲-۴- نمودار مقایسه دقیق سیگنال تمیز (نمودار قرمز) و سیگنال تخمینی (نمودار آبی) در ۵۰۰ تکرار
۹۷	۳۳-۴- نمودار مقایسه سیگنال تمیز (شکل اول) سیگنال نویزی (شکل دوم) و سیگنال تخمینی (شکل سوم)
۹۷	۳۴-۴- نمودار مقایسه سیگنال تمیز (نمودار قرمز) و سیگنال تخمینی (نمودار آبی)
۹۸	۳۵-۴- نمودار مقایسه دقیق سیگنال تمیز (نمودار قرمز) و سیگنال تخمینی (نمودار آبی) در ۲۵۰ تکرار
۹۸	۳۶-۴- نمودار مقایسه دقیق سیگنال تمیز (نمودار قرمز) و سیگنال تخمینی (نمودار آبی) در ۲۵۰ تکرار
۹۸	۳۷-۴- نمودار مقایسه دقیق سیگنال تمیز (نمودار قرمز) و سیگنال تخمینی (نمودار آبی) در ۲۵۰ تکرار
۹۹	۳۸-۴- نمودار مقایسه دقیق سیگنال تمیز (نمودار قرمز) و سیگنال تخمینی (نمودار آبی) در ۲۵۰ تکرار
۹۹	۳۹-۴- نمودار مقایسه دقیق سیگنال تمیز (نمودار قرمز) و سیگنال تخمینی (نمودار آبی) در ۲۵۰ تکرار
۹۹	۴۰-۴- نمودار مقایسه دقیق سیگنال تمیز (نمودار قرمز) و سیگنال تخمینی (نمودار آبی) در ۲۵۰ تکرار

صفحه	فهرست اشکال
۲۷	۱-۲-۲- نمایش گراف یک سیستم دینامیکی خطی زمان گسسته
۳۹	۱-۴-۲- نمایش فیلتر کالمن توسعه یافته دوگانه [۳۳]
۴۰	۲-۴-۲- نمایش تخمین ترتیبی جهت مسئله تخمین دوگانه
۵۷	۱-۵-۳- نمایش مدلسازی سیستم غیر خطی با روش ARX غیر خطی
۵۸	۲-۵-۳- نمایش شبکه عصبی بکار رفته در روش ARX غیر خطی
۶۰	۳-۵-۳- نمایش ساختار موازی شبکه عصبی پیشرو
۶۰	۴-۵-۳- نمایش ساختار موازی شبکه عصبی پیشرو
۶۲	۵-۵-۳- نمایش مدلسازی سیستم غیر خطی با روش AR غیر خطی
۶۳	۶-۵-۳- نمایش شبکه عصبی بکار رفته در روش AR غیر خطی
۶۸	۱-۴- تخمین سیگنال و مدل دو امر وابسته به یکدیگر است و پیش بینی به هر دوی آنها نیازمند است
۶۹	شکل ۲-۴- در این نمایش فرض شده که اطلاعات با یک مدل AR غیر خطی نامعلوم تهیه شده و با نویز مشاهده افزودنی تخریب شده است.
۷۰	۳-۴- حلقه شناسایی سیستم: پس از N بار تکرار و همگرا شدن خطا، $N_k = X_k$ خواهد شد
۷۱	۴-۴- حلقه شناسایی سیستم: پس از N بار تکرار و همگرا شدن خطا، $N_k = X_k$ خواهد شد
۷۳	۵-۴- شبکه عصبی MLP با ۵ ورودی- ۵ لایه مخفی و ۱ خروجی

چکیده پایان نامه:

در فصل اول، پایان نامه مروری بر پژوهشهای انجام شده توسط محققان و دانشمندان طی نیم قرن اخیر دارد و سعی می کند تا روند تکامل خانواده فیلتر کالمن را مورد بررسی قرار دهد. در فصل دوم به معرفی فیلتر کالمن، فیلتر کالمن توسعه یافته و فیلتر کالمن دوگانه می پردازد و روابط آنها و چگونگی استخراج این روابط تشریح می گردد. در فصل سوم موضوع شناسایی سیستمهای خطی و غیر خطی مطرح می گردد و انواع روشهای شناسایی سیستمهای خطی و غیر خطی مورد مطالعه قرار می گیرد. همچنین کاربرد شبکه عصبی در شناسایی سیستمهای غیر خطی و نحوه آموزش و انتخاب ساختار شبکه عصبی تشریح می گردد و در ادامه نحوه آموزش شبکه عصبی انتخاب شده ارائه می گردد. در فصل چهارم نیز به ارائه شبیه سازیهای انجام شده با نرم افزار Matlab 6.5 و مقایسه آنها با یکدیگر پرداخته شده است.

مقدمه:

موضوع بهبود گفتار با نیاز به افزایش کیفیت عملکرد سیستمهای ارتباطی صوتی در محیط های نوپزی، مطرح گردید. رنج عملکردی وسیعی برای سیستمهای تشخیص گفتار جهت بهبود ارتباط از راه دور در هوانوردی، صنایع نظامی، گفتگوهای راه دور و محیط های سلولی وجود دارد. هدف ما نیز در این پایان نامه بهبود کیفیت قابل ملاحظه گفتار یا افزایش قابلیت فهم آن می باشد.

کاربردهای فراوانی از بهبود گفتارهای صوتی تا پیش بینی های اقتصادی و کنترل تطبیقی نیازمند تخمین و مدلسازی دنباله های زمانی نوپزی می باشند. از این جمله می توان به بهبود گفتارهای صوتی، پیش بینی اقتصادی، مدلسازی ژئوفیزیکی و بسیاری کاربردهای دیگر اشاره کرد. یک دنباله زمانی نوپزی می تواند با یک مدل احتمالی که هر دوی اجزای تقریبی و دقیق دینامیک ها را تخمین می زند، توصیف شود. چنین مدلی می تواند به همراه فیلتر کالمن (یا فیلتر کالمن توسعه یافته) جهت تخمین و پیش بینی سری زمانی از مشاهدات نوپزی بکار گیری شود.

فیلتر کالمن یک فیلتر بهینه خطی است که بر روی فضای حالت سیستمهای خطی استاتیکی و دینامیکی اثر گذاشته و یک تخمین بهینه از حالت های سیستم با استفاده از معادلات برگشت پذیر و دینامیکی خود در شرایطی که دسترسی به آنها میسر نباشد ارائه می دهد. همچنین این فیلتر می تواند تاثیر کلیه اطلاعات گذشته و ابتدایی سیستم را نیز در تخمین هر لحظه خود لحاظ نماید.

بنابراین با توضیحات ارائه شده در بالا می توان فهمید که جهت بازبایی یک سیگنال صوتی به یک تخمین خوب نیاز است اما موضوع مهم این است که فیلتر کالمن خطی قادر به ارائه تخمین از مدل های سیستمهای غیر خطی نمی باشد حال آنکه اکثر سیستمهای واقعی که سیستمهای صوتی نیز از این جمله می باشند ماهیت غیر خطی دارند. لذا در این پایان نامه بر آن شدیم تا با مطالعه بر روی فیلتر کالمن توسعه یافته و به کمک شبکه های عصبی بتوانیم موضوع مدلسازی و تخمین سیگنال صوتی را با فرض غیر خطی بودن آن بررسی نماییم.

۱ مروری بر پژوهشهای

پیشین

۱-۱- فیلتر وینر:

روبرت وینر^۱ به همراه نویلسون از دانشگاه MIT آمریکا در خلال سالهای ۱۹۴۰ تا ۱۹۴۹ تحقیق گسترده ای را جهت یافتن یک فیلتر بهینه به منظور تخمین حالت‌های سیستم‌های خطی انجام دادند [۱] که نهایتاً در سال ۱۹۴۹ منجر به معرفی فیلتر وینر گردید. این فیلتر یک تخمین خطی با حداقل کردن میانگین مربعات خطا (LMMSE)^۲ برای سیگنال مشاهده شده ارائه می‌دهد. اما این فیلتر دارای چند مشکل بود. اول آنکه ما نیازمند تخمین‌های علی هستیم و در نتیجه برای تخمین علی بایستی از فیلتر وینر علی استفاده شود در حالیکه فیلتر وینر فقط N مشاهده جدید را جهت تخمین بکارگیری می‌کرد و در نتیجه ممکن بود اطلاعاتی از سیگنال در مشاهدات گذشته وجود داشته باشد که در این صورت از بین می‌رفت. دوم آنکه این فیلتر فقط برای نویزهای ثابت با میانگین صفر طراحی شده بود در حالیکه در دنیای واقعیت انواع نویزهای غیر ثابت وجود دارد. سوم آنکه فیلتر وینر جهت ارائه تخمین سیگنال نیاز به چگالی طیفی و تابع خود همبستگی سیگنال دارد که در همه مواقع در دسترس نمی‌باشد. و چهارم آنکه تخمین‌های LMMSE و

^۱ Robert Wiener

^۲ Linear Minimum Mean Square error

MAP^۴ و ML^۳ که جهت محاسبات این فیلتر بکار می روند در هر زمان نیاز به مشاهدات جدید بودند که مستلزم حافظه بالا بود. همه این عوامل محققان را در طی سالهای ۱۹۵۰ تا ۱۹۶۰ به فکر انداخت تا در پی معرفی فیلتری جدید با قابلیت ارائه تخمین خطی بهینه از طریق مینیمم کردن میانگین خطا و عاری از مشکلات فیلتر وینر باشند. [۲-۴]

۱-۲- فیلتر کالمن:

لذا رادولف المان کالمن^۵ در سال ۱۹۶۰ [۲-۳] روش معروف خود را به همراه نتایج آن مبنی بر ارائه یک روش بازگشتی فیلتر سازی اطلاعات زمان گسسته مطرح نمود که بر مبنای تخمین MSE^۶ طراحی شده بود و دارای هیچ یک از مشکلات فیلتر وینر نبود. به تدریج این نظریه به نام فیلتر کالمن معرفی شد و در محاسبات دیجیتال و کنترل دیجیتال بکارگیری شد اما توانایی فوق العاده و بالای این فیلتر در تخمین حالت‌های سیستم‌های خطی باعث فراگیر شدن آن در حوزه‌های مختلف کاربردی گردید. این فیلتر با مینیمم کردن میانگین مربعات خطای تخمین می توانست تخمین‌های بسیار خوبی را از حالت‌های گذشته - حال و آینده سیستم ارائه دهد. در حقیقت این فیلتر نیاز به پارامترهای سیستم مورد تخمین داشت که در صورت ناشناس بودن سیستم بایستی توسط یکی از روش‌های خطی شناسایی سیستمها (...^۹ARMAX^۹-ARX^۸-AR^۷) این پارامترها در اختیار فیلتر کالمن جهت تخمین حالت‌های مورد نظر قرار می گرفت. قابلیت های این فیلتر در مقایسه با فیلتر وینر به شرح ذیل است:

۱- نیاز به فضای حالت سیستم بجای طیف فرکانسی جهت محاسبه تخمین.

۲- دارای قابلیت ارائه تخمین در حضور انواع نویز و فضای حالت متغیر با زمان.

^۳ Maximum Likelihood

^۴ Maximum a posteriori

^۵ Rudolph Elman Kalman

^۶ Minimum Square Error

^۷ Auto Regressive

^۸ Auto Regressive External Input

^۹ Auto Regressive Moving Average With External Input

۳- عدم نیاز به دانستن اطلاعات در مورد نحوه تولید سیگنال مورد تخمین. یعنی اطلاعات مربوط به سیستم از طریق روشهای شناسایی سیستم به فیلتر کالمن داده می شود و اطلاعات سیستم بطور مستقیم در تخمین کالمن وارد نمی شود.

لذا در خلال سالهای ۱۹۶۰ تا ۱۹۷۰ کاربردهای بیشماری از فیلتر کالمن خطی در حوزه های مختلف کنترل و مخابرات مطرح گردید که البته محدود به سیستمهای خطی می شد نمونه هایی از آن در [۳] و [۴] و [۵] آمده است..

۱-۳- فیلتر کالمن توسعه یافته:

اما یک مشکل عمده در اواسط سال ۱۹۷۵ دانشمندان را به فکر انداخت تا به فکر توسعه فیلتر کالمن باشند. مشکل عمده فیلتر کالمن این بود که محدود به سیستمهای خطی می شد و قابلیت تخمین حالتیهای سیستمهای غیر خطی را نداشت حال آنکه در دنیای واقعیت اکثر سیستمها دارای ماهیت غیر خطی می باشند. لذا با ایجاد تغییراتی در فیلتر کالمن، فیلتری بنام فیلتر کالمن توسعه یافته^{۱۰} معرفی شد که قابلیت تخمین حالتیهای سیستمهای غیر خطی را نیز دارا بود. در واقع این فیلتر مدل غیر خطی را با عنوان مدل خطی تغییر پذیر با زمان تخمین می زد. از آنجائیکه ماهیت صوت نیز غیر خطی است در نتیجه با معرفی این فیلتر محققان توانستند زمینه جدیدی را در تخمین و بهبود واقعی تر سیگنالهای صوتی مطرح کنند. [۶]

در سال ۱۹۷۹ دکتر لونگ، پیشنهاد ترکیب بردار سیگنال تاخیر دار X_{k-1} و پارامترهای ناشناس W را در یک بردار حالت ترکیبی جهت تشکیل یک نمایش فضای حالت غیر خطی مطرح کرد. که در آن فیلتر کالمن توسعه یافته جهت تخمین غیر خطی سیگنال بکارگیری می شد. این

¹⁰ Extended kalman filter

روش بنام فیلتر کالمن توسعه یافته مشترک^{۱۱} (Joint EKF) [۷] بود. که البته این روش بدلیل پیچیدگی محاسباتی آن، مورد توجه محققان قرار نگرفت و خیلی کاربردی نشد.

در سال ۱۹۸۷ پروفیسور کی پالیوال بحث کاربرد فیلتر کالمن در بهبود گفتار را در مقاله خود در کنفرانس بین المللی پردازش سیگنال امریکا مطرح کرد [۸]. او در این مقاله با استفاده از شناسایی AR سیستم خطی که همان گفتار تمیز بود و تلفیق فیلتر کالمن با آن توانست سیگنال صوتی را بازیافت کند. که البته او فرض کرده بود که واریانسهای نویز قابل شناسایی است و سپس با مقایسه نسبت سیگنال به نویز (SNR^{12}) سیگنال اصلی و نویزی، بهبود سیگنال را اثبات کرده بود. طی سالهای ۱۹۸۷ تا ۱۹۹۷ مقالات متعددی در خصوص حذف نویز از سیگنالهای صوتی توسط محققان مطرح گردید که اکثر آنها بر استفاده از فیلتر کالمن تاکید داشتند البته در خلال این سالها از فیلتر کالمن توسعه یافته در بسیاری از تخمینهای سیستمهای غیر خطی بهره گیری شد اما بدلیل وجود مشکل اصلی دیگری که فیلتر کالمن توسعه یافته نتوانسته بود به آن پاسخ دهد و آن عدم توانایی تخمین واریانس نویزهای پروسه و مشاهده بود، محققان در پی معرفی یک فیلتر جدید بودند که توانایی تخمین نویز را داشته باشد زیرا در سیستمهای واقعی پردازش صوت، ما به سیگنال تمیز دسترسی نداریم و در نتیجه در بهره گیری از فیلتر کالمن با مشکل مواجه می شدند.

۱-۴- کاهش طیفی:

تلاشهای زیادی در سالهای ۱۹۷۹ تا ۱۹۸۳ توسط دانشمندان مختلف از جمله اپنهایم، لیم و مالپاس [۹-۱۱] در مورد روش کاهش طیفی جهت بهبود سیگنالهای صوتی انجام شد تا سرانجام روشی بنام روش کاهش طیفی نیز توسط دکتر هیرچ در سال ۱۹۹۳ مطرح گردید [۱۲]. مبنای این روش بر این اساس بود که بایستی سیگنال نویزی و نویز توسط تبدیل فوریه به حوزه فرکانس تصویر شوند و سپس توان طیفی سیگنال نویزی (\hat{P}_y) و نویز (\hat{P}_n) محاسبه گردد و پس از تفریق

¹¹ Joint Extended kalman filter

¹² Signal to Noise Ratio

این ۲ پارامتر از یکدیگر مجدداً سیگنال بدست آمده $(\hat{P}_x = [\hat{P}_y^\gamma - \alpha \hat{P}_n^\gamma]^\frac{1}{\gamma})$ با عکس تبدیل فوریه به حوزه زمان تصویر گردند. که این روش دارای ۴ مشکل عمده بود. اول آنکه این روش نیازمند بود تا سیگنال در دسترس به ۲ بخش گفتار و غیر گفتار تقسیم شود که این امر به سادگی امکان پذیر نبود و در نتیجه در تفریق ۲ سیگنال از هم معمولاً خواص سیگنال تمیز نیز از بین می رفت و سیگنال بازیافتی از مرغوبیت خوبی برخوردار نبود و دوم آنکه نیاز به داشتن اطلاعات کافی از نویزی بودیم که سیگنال تمیز به آن آغشته شده است که این هم در بسیاری از موارد میسر نبود. سوم آنکه اطلاعات بازیافت شده فقط شامل اطلاعات بهره سیگنال بود و اطلاعات مربوط به فاز آن از بین می رفت. چهارم آنکه بایستی به کل سیگنال در ابتدای پردازش بصورت بسته ای دسترسی می داشتند که این باعث محدودیت این روش به کاربردهای نابهنگام می گشت. لذا این روش نیز خیلی مورد توجه محققان واقع نگردید.

۱-۵- استفاده مستقیم از شبکه عصبی:

همچنین در خلال سالهای ۱۹۸۶ تا ۱۹۹۶ دانشمندان متخصص شبکه های عصبی روشهای مستقیم پردازش سیگنالهای صوتی توسط شبکه های عصبی را مطرح و آزمایش نمودند در واقع برای اولین بار دکتر تامورا در سال ۱۹۸۷ [۱۳] این موضوع را مطرح کرد. مبنای کار او و محققان دیگر که تا سال ۱۹۹۶ این حوزه را دنبال کردند [۱۴-۱۶] بر این اساس بود که یک شبکه عصبی چند لایه جهت تصویر کردن بخش های پنجره شده گفتار نویزی به تخمین سیگنال تمیز بکار گیری می شد. تعداد ورودی ها به نرخ نمونه برداری سیگنال گفتار بستگی داشت و معمولاً برای پوشش ۵ تا ۱۰ میلی ثانیه اطلاعات تنظیم می شد. همچنین تعداد خروجی ها معمولاً با تعداد ورودی ها برابر بود. برای آموزش شبکه، گفتار تمیز به طور مصنوعی تخریب می شد تا اطلاعات ورودی نویزی را بسازد. سپس سیگنال گفتار تمیز به عنوان هدفی که با ورودی ها به طور زمانی

ردیف شده است بکار می رفت. روش پس انتشار خطای استاندارد^{۱۳} (BP) یا هر روش دیگر آموزش شبکه های عصبی نیز جهت آموزش شبکه با سیگنال هدف ذکر شده بکارگیری می شد. مزیت این روشها اجرای آسان و راندمان بالای آنها بود اما مشکل اصلی آنها این بود که محدود به زمانی می شدند که ما به سیگنال صوتی تمیز دسترسی داشته باشیم زیرا سیگنال تمیز بایستی به عنوان هدف به شبکه عصبی تحت آموزش اعمال می شد و در غیر اینصورت این روشها هیچ کاربردی نمی داشتند و همین محدودیت باعث شد این روشها محدود به آزمایشگاههای پردازش سیگنال بمانند و کاربرد عملی پیدا نکنند.

۱-۶- فیلتر کالمن توسعه یافته دوگانه:

اما بالاخره در سال ۱۹۹۷ پروفیسور اریک ون و پروفیسور الکس نلسون از دانشگاه MIT امریکا با ارائه ۲ مقاله راه گشا فیلتری بنام فیلتر کالمن توسعه یافته دوگانه^{۱۴} (DEKF) را معرفی نمودند [۱۷-۱۸]. این فیلتر دوگانه، متشکل از یک فیلتر وزن و یک فیلتر حالت بود. بدین ترتیب که فیلتر حالت تخمینهای سیگنال را با فرض معلوم بودن مدل تهیه می کرد و فیلتر وزن تخمینهای پارامترها را با فرض معلوم بودن سیگنال بدست می داد. همچنین این فیلتر با مینیمم کردن یک سری توابع معیار قادر به تخمین واریانس نویزهای مشاهده و پروسه نیز بود. که البته این نظریه تا سال ۱۹۹۹ بصورت اجرا در نیامد تا بالاخره در سال ۲۰۰۰ با ارائه نتایج کامل شبیه سازیهای آزمایشگاهی و واقعی که در لابراتوار موسسه پردازش سیگنال امریکا^{۱۵} (CSLU) [۱۹] صورت گرفته بود، اثبات نمودند که این روش بسیار کارآمد و کاربردی جهت بهبود سیگنالهای صوتی در محیط های نویزی واقعی می باشد. این دو محقق با ارائه مقالات در خلال سالهای ۱۹۹۸ تا ۲۰۰۰، [۲۰-۲۲]، فیلتر کالمن دوگانه را بطور کامل معرفی کردند. آنها در تمام این مقالات اذعان داشتند که برتری این فیلتر نسبت به فیلترهای قبلی بدلیل ۱ توانایی منحصر بفرد این فیلتر

¹³ Back Propagation

¹⁴ Dual Extended Kalman filter

¹⁵ Central of Speech Language Understanding

است و آن این بود که این فیلتر قادر به ارائه تخمین سیگنال تمیز در زمانی است که ما فقط به سیگنال نویزی دسترسی داشته باشیم و هیچ گونه اطلاعاتی از سیگنال تمیز و حالت‌های آن در دسترس ما نباشد.

در واقع معرفی این فیلتر بسیاری از ابهامات و ایرادات موجود در راه پردازش سیگنال‌های صوتی را که محققان قبلی را با مشکل مواجه کرده بود برطرف ساخت و توانست به عنوان اولین فیلتر کالمن کاربردی در دنیای صنعت معرفی شود که البته هنوز در مقالات اشاره ای به کاربردهای عملی آن نشده و فعلا در حد کاربردهای آزمایشگاهی با درصد اطمینان بالا، قابل ارائه می باشد.

۱-۷- فیلتر کالمن درک نشده:

اما یک مشکل دیگر باعث شد تا دانشمندان به فکر بهبود این فیلتر نیز باشند. آن مشکل این بود که در فیلتر کالمن توسعه یافته بدلیل خطی سازی معادلات سیستم غیر خطی که از درجه ۱ می باشد، آنها در محاسبه خطای میانگین و کوواریانس درای انحراف می شدند و همین انحراف در معادلات بازگشتی به تدریج موجبات واگرایی تخمین را فراهم می نمود. لذا فیلتری جدید در سالهای ۱۹۹۵ تا ۱۹۹۷ توسط دکتر جولیر و دکتر آلمن در کنفرانس های کنترل امریکا معرفی شد [۲۳-۲۵]. که البته دارای نواقص تئوری و عملی بود که در سال ۲۰۰۰ توسط پروفیسور اریک ون و پروفیسور ون در مورو از دانشگاه MIT امریکا ، اصلاح گردید و بصورت مقاله ای در کنفرانس پردازش سیگنال‌های مخابراتی و کنترل کانادا در سال ۲۰۰۰ ارائه گردید [۲۶-۲۷]. نام این فیلتر فیلتر کالمن درک نشده^{۱۶} (UKF) بود که علت این نام بدلیل جدید و ناشناس بودن این فیلتر به آن اطلاق شد که مبنای تئوری این فیلتر بر خطی سازی درجه ۲ تیلور استوار بود. که در واقع تخمین‌هایی به مراتب دقیق تر از فیلتر کالمن توسعه یافته که دارای خطی سازی درجه ۱ بود از خود نشان می داد. همچنین اگرچه به ظاهر معادلات این فیلتر مشکل تر از فیلتر کالمن توسعه

¹⁶ Unscented Kalman Filter

یافته می باشد اما در واقع اجرای آن به مراتب راحت تر از معادلات EKF می باشد. علت این امر عدم نیاز به محاسبه مشتق های بازگشتی (دیورژانس) در معادلات UKF است. مزیت دیگر این فیلتر قابلیت ارائه تخمینهای دقیق تر در مواقعی است که سیستم بصورت ناشناس^{۱۷} است. یعنی بدلیل عدم نیاز به محاسبه مشتقات جزئی، در نتیجه نیاز به اطلاعات دقیق از سیستم هم تا حد زیادی منتفی است. اما این روش نیز دارای یک مشکل بود. محدودیت این روش وابسته بودن شدید تخمینهای آن به انتخاب ۳ پارامتر تبدیل Unscented است (α, β, κ) که بسته به نوع تخمین انتخاب آنها متفاوت است و اگر دقیق انتخاب نشوند (همانند انتخاب ضرایب وزن اولیه، ضریب یادگیری و فاکتور فراموشی در شبکه های عصبی)، نه تنها تخمینهای خوبی ارائه نمی دهند بلکه باعث واگرایی نیز می گردند. لذا محققان از سال ۲۰۰۴ تا کنون در حال مطالعه روشهایی هستند که بتوانند این پارامترها را بصورت دقیق و بهینه به آنها ارائه دهند.

۱-۸- فیلتر کالمن درک نشده دوگانه:

همچنین فیلتر کالمن درک نشده دوگانه^{۱۸} (DUKF) که مبنای عملکرد آن مانند فیلتر کالمن توسعه یافته دوگانه بود و متشکل از ۲ فیلتر وزن و حالت جهت تخمینهای بازگشتی وزن و حالت بود در سال ۲۰۰۰ توسط پروفیسور اریک ون و پروفیسور ون در مورو مطرح گردید [۲۸]. که البته تنها تفاوت آن با فیلتر درک نشده، قابلیت تخمین نویزهای سیگنال صوتی می باشد و در بقیه موارد از منطق عملکردی فیلتر کالمن درک نشده پیروی می کند.

لذا با توجه به مروری که بر کارهای انجام شده در حوزه بهبود سیگنالهای صوتی توسط خانواده فیلترهای کالمن انجام شد، مشاهده می شود که بسیاری از معضلات و مشکلاتی که فیلتر کالمن اولیه که توسط کالمن در سال ۱۹۶۰ مطرح شد دارا بود، اکنون پس از گذشت ۴۶ سال و چند

¹⁷ Black Box

¹⁸ Dual Unscented Kalman Filter

دوره تغییر و بهبود بطور کامل رفع شده و این امر باعث شده تا فیلتر کالمن بیش از پیش در عرصه عمل موثر و کاربردی نمایان گردد.

بنابراین بطور کلی می توان دوره های جهش و تغییر فیلتر کالمن را بصورت زیر دسته بندی کرد که البته تمرکز و شاخص این تقسیم بندی با توجه به حوزه پردازش و بهبود سیگنالهای صوتی انجام گرفته است:

۱- سالهای ۱۹۶۰ تا ۱۹۷۵ : معرفی فیلتر کالمن و بسط و توسعه آن در کاربردهای کنترلی و مخابراتی جهت ارائه تخمینهای خطی..

۲- سالهای ۱۹۷۵ تا ۱۹۹۰ : معرفی فیلتر کالمن توسعه یافته و بکارگیری آن در مخابرات و کنترل جهت ارائه تخمینهای غیر خطی.

۳- سالهای ۱۹۹۰ تا ۲۰۰۰ : ارائه تئوریهای مختلف جهت تخمین نویز پروسه و مشاهده سیستم به همراه بکارگیری فیلتر کالمن توسعه یافته در کاربردهای عملی- بکارگیری شبکه های عصبی مختلف جهت شناسایی سیستم به منظور استفاده در فیلتر کالمن توسعه یافته.

۴- سالهای ۲۰۰۰ تا ۲۰۰۶ : معرفی فیلتر کالمن توسعه یافته دوگانه (DEKF)- معرفی فیلتر کالمن درک نشده (UKF)- معرفی فیلتر کالمن درک نشده دوگانه (DUKF) و بکارگیری آنها در کاربردهای واقعی بخصوص پردازش سیگنالهای صوتی و دست یابی به توانایی بهبود گفتار نویزی بدون دسترسی به کوچکترین اطلاعاتی از گفتار اصلی با درصد بهبود خوب.

۵- سال ۲۰۰۶ به بعد : محققان در حال تلاش جهت رسیدن به یک سیستم محاسبه بهینه فاکتورهای فیلتر کالمن درک نشده و بسط و گسترش این فیلتر و کاربردهای آن در صنایع دیگری باشند.

همانطور که مشاهده می شود ، روند پیشرفت در این عرصه همانند عرصه های علمی دیگر به مرور زمان شتابی چند برابر از گذشته به خود گرفته است و در خلال سالهای ۲۰۰۰ تا ۲۰۰۶ مشکلاتی که حدود ۴۰ سال در مورد فیلتر کالمن و کاربرد آن در بهبود سیگنالهای صوتی لا ینحل باقی مانده بود به سرعت با تلاش محققان برطرف گشته است.

لذا لازم است تا محققان ایرانی نیز همگام با محققان کشورهای مختلف در راستای بکارگیری این فیلتر در کاربردهای صنعتی جدید و بسط و گسترش آن گامهای موثری بردارند. و اما مواردی که باعث شد اینجانب فیلتر کالمن توسعه یافته را جهت کار انتخاب نمایم در حالیکه تا سال ۲۰۰۰ کار بر روی این فیلتر در امریکا کامل گشته است چند عامل است که در ذیل به آنها اشاره می شود :

۱- علیرغم اینکه در امریکا و کشورهای اروپایی بر روی این فیلتر موفقیت هایی حاصل گشته است اما موضوع کاربرد فیلتر کالمن توسعه یافته در بهبود سیگنالهای صوتی در ایران موضوعی کاملا جدید است و تنها کار انجام شده در این حوزه مربوط به پایان نامه خانم نیره سلیم پور از دانشگاه خواجه نصیر است [۲۹] که آن هم به کاربرد فیلتر کالمن خطی در حوزه بهبود گفتار پرداخته که با سیستمهای واقعی و روز دنیا که ماهیت غیرخطی دارند تفاوت دارد .

۲- بدلیل کمبود امکانات دانشگاهی در کشور و عدم دسترسی مناسب به سایتهای اینترنتی از قبیل IEEE^{۱۹} و بدنبال آن عدم دسترسی آسان و بروز به مقالات و پیشرفت های روز دنیا (بخصوص که اغلب این پیشرفت ها در کشور امریکا صورت گرفته است)، باعث شد تا آشنایی کامل با پیشرفت های روز دنیا جهت پیاده سازی یا ادامه آنها میسر نباشد. در حالیکه تمام پیشرفت های سال ۲۰۰۰ تا ۲۰۰۶ در این حوزه منحصر به دانشگاه MIT

¹⁹ Institute of Electrical and Electronics Engineering

امریکاست که با حمایت یک آزمایشگاه بزرگ پردازش سیگنالهای مخابراتی [۲۲] به انجام رسیده است که از حیث امکانات قابل مقایسه نمی باشد.

۳- همچنین بدلیل پیش نیاز بودن آشنایی محققان کشور با کاربرد فیلتر کالمن توسعه یافته در پردازش گفتار ، جهت ادامه راه که همان آشنایی با فیلترهای جدید کالمن توسعه یافته دوگانه (DEKF) و کالمن درک نشده دوگانه (DUKF) و کاربردهای آنها می باشد بر آن شدیم تا ابتدا از فیلتر کالمن توسعه یافته آغاز کنیم.

معرفی انواع فیلترهای کالمن

۲-۱- مقدمه :

همانطور که ذکر شد جهت بازیابی یک سیکنال صوتی به یک تخمین خوب نیاز است، لذا این فصل مطالعه ای بر ۳ نوع از انواع فیلتر کالمن خواهد داشت. در ابتدا به معرفی فیلتر کالمن و چگونگی استخراج روابط آن پرداخته می شود. سپس خصوصیات و ویژگیهای آن برشمرده می شود و در ادامه مشکلات آن که باعث معرفی فیلتر بعدی یعنی فیلتر کالمن توسعه یافته می گردد بیان می شود.

همچنین فیلتر کالمن توسعه یافته به همراه روابط ریاضی آن، چگونگی استخراج این روابط و ویژگیهای آن معرفی می گردد. که البته شبیه سازیهای مربوط به این فیلتر با کاربرد بهبود سیگنالهای صوتی بطور کامل در فصل ۴ آمده است لذا از پرداختن به جزئیات شبیه سازی آن در این بخش خودداری می کنیم.

و در پایان نیز گریزی بر معرفی یک فیلتر جدید کالمن با نام فیلتر کالمن توسعه یافته دوگانه خواهیم شد که دارای روابط ریاضی جدید و پیچیده خاص خود است و لزوم معرفی و کاربردهای آن مطرح می شود. همچنین عواملی که باعث معرفی این فیلتر جدید به عرصه علم گردید مطرح می گردد. که البته به دلایل ذکر شده در فصل اول، این پایان نامه فقط به معرفی این فیلتر اکتفا کرده و از ارائه شبیه سازیهای جدید و پیچیده در مورد آن پرهیز نموده است.

۲-۲-۲- فیلتر کالمن (KF) :

فیلتر کالمن یک فیلتر بهینه خطی است که بر روی فضای حالت سیستمهای خطی استاتیکی و دینامیکی اثر گذاشته و یک تخمین بهینه از حالت‌های سیستم با استفاده از معادلات بر گشت پذیر و دینامیکی خود در شرایطی که دسترسی به آنها میسر نباشد ارائه می دهد. همچنین این فیلتر می تواند تاثیر کلیه اطلاعات گذشته و ابتدایی سیستم را نیز در تخمین هر لحظه خود لحاظ نماید. همچنین در مورد تخمینهای MAP، MSE و ML در مرجع [۳۴] بطور کامل بحث شده است لذا این پایان نامه از ارائه توضیحات اضافی در مورد آنها در این فصل اجتناب کرده و فقط به ذکر این نکته بسنده کرده که معادلات تخمین LMMSE به عنوان ایده ای جهت حداقل کردن مقدار خطای مربعات (MSE) بکار رفته است. [۳۴]

۲-۲-۱- مدل سیگنال و مشاهده:

در اینجا فرضیهایی در مورد $s(n)$ و $z(n)$ مطرح شده است و فرض بر این است که سیگنال را بتوان به فضای حالت برد.

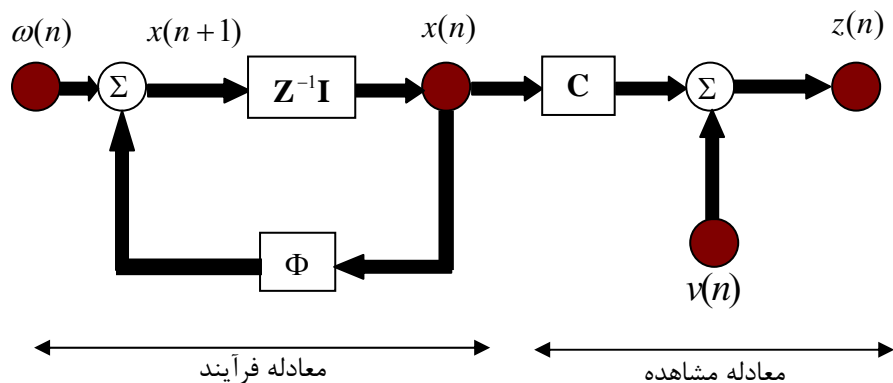
$$s(n) = \begin{bmatrix} s_1(n) \\ s_2(n) \\ \vdots \\ s_p(n) \end{bmatrix} \quad z(n) = \begin{bmatrix} z_1(n) \\ z_2(n) \\ \vdots \\ z_p(n) \end{bmatrix} \quad (1-2-2)$$

سپس مدل فضای حالت $s(n)$ بصورت زیر آمده است. فرض شده که سیگنال $s(n)$ بتواند بصورت خطی و غیر متغیر با زمان مدل شود. حال اگر نویزی هم به مشاهده اضافه شود خواهیم داشت:

$$\begin{aligned} x(n+1) &= \Phi x(n) + \Gamma \omega(n) \\ s(n) &= c x(n) \\ z(n) &= s(n) + v(n) \end{aligned} \quad \text{SMM Model} \quad (2-2-2)$$

Φ, Γ, c ماتریسهای شناخته شده اند.

همچنین بلوک دیاگرام حلقه کنترلی این سیستم به صورت زیر است:



شکل ۲-۲-۱- نمایش گراف یک سیستم دینامیکی خطی زمان گسسته

و دارای مشخصات زیر است:

- ماتریس Φ ($N \times N$) ، ماتریس Γ ($N \times m$) و ماتریس C ($p \times N$) است.
- $x(n)$ ماتریس ($N \times 1$) که $x(n)$ متغیر حالت است.
- $x(0)$ ماتریس شرایط اولیه است.
- $\omega(n)$ که نویز فرآیند نامیده می شود نویز سفید گوسی با میانگین صفر است و ماتریس $(m \times m)$ با واریانس Q می باشد.

$$E[\omega(i)\omega^T(j)] = Q\delta(i-j) \quad (3-2-2)$$

- $v(n)$ که نویز مشاهده نامیده می شود نویز سفید گوسی با میانگین صفر است و ماتریس $(p \times p)$ با واریانس R می باشد.

$$E[v(i)v^T(j)] = R\delta(i-j) \quad (4-2-2)$$

و در تمام n ها، $x(0)$ ، $v(n)$ ، $\omega(n)$ مستقل از هم می باشند.

قابل تخمین $s(n)$ است زیرا با توجه به معلوم بودن مقدار $x(n)$ مسأله، تخمین متغیر حالت

$$\hat{s}(n) = c \hat{x}(n) \text{ محاسبه می باشد.}$$

یافتن تخمین $\hat{x}(n)$ مسأله فیلتر کالمن است و روابط فضای حالت و ماتریسهای واریانس

فرضهای اصلی فیلتر کالمن هستند.

این ویژگیها باعث می شود فیلتر کالمن بگونه ای عمل کند که فیلتر وینر قادر به آن نبود. فیلتر وینر به نویزی ثابت با میانگین صفر نیاز داشت تا فرآیندی پایدار داشته باشد اما فیلتر کالمن می تواند با روابط فضای حالت متغیر با زمان و نویز غیر ثابت بکار رود. همچنین در فیلتر وینر به چگالی طیفی و تابع خود همبستگی نیاز است، در حالیکه برای فیلتر کالمن مدل فضای حالت سیگنال نیاز می باشد [۳۰].

در اینجا پارامترهایی مورد بررسی قرار خواهد گرفت که مورد استفاده در فیلتر کالمن است.

همانطور که می دانیم تخمینگر ما از ۳ قسمت تشکیل شده بود: [۳۴]

۱. Filter gain update

۲. Estimate update

۳. MSE update

فیلتر کالمنی با همین قسمتها در این پایان نامه معرفی شده است. چیزی که در اینجا لازم است تخمینهای قبل و بعد است. تخمین قبل $x(n)$ تخمینی است که قبل از در دسترس بودن $z(n)$ ساخته می شود و تخمین بعد $x(n)$ تخمینی است که بعد از در دسترس بودن $z(n)$ ساخته می شود.

$Z^- = \{z(1), \dots, z(n-1)\}$ به اینصورت تعریف شده است:

که از آن می توان $x(n)$ را تخمین زد.

تخمین $x(n)$ بطوریکه z^- را داریم $\hat{x}^-(n)$ (priori estimate)

و وقتی $z(n)$ در دسترس باشد یعنی: $Z = \{z(1), \dots, z(n)\}$ در اینصورت خواهیم داشت:

تخمین $x(n)$ بطوریکه z را داریم $\hat{x}(n)$ (posteriori estimate)

خطای تخمین نیز به این صورت تعریف شده است:

$$\tilde{x}^-(n) = x(n) - \hat{x}^-(n) \quad (5-2-2)$$

$$\tilde{x}(n) = x(n) - \hat{x}(n)$$

و از آن $P(n)$ را که واریانس خطای تخمین است می توان یافت:

$$P^-(n) = \text{cov}[\tilde{x}^-(n)] = E[\tilde{x}^-(n) \cdot (\tilde{x}^-(n))^T] \quad (6-2-2)$$

$$P(n) = \text{cov}[\tilde{x}(n)] = E[\tilde{x}(n) \cdot \tilde{x}^T(n)]$$

و برای MSE خواهیم داشت:

$$\text{MSE}(\hat{x}^-(n)) = E[(\tilde{x}^-(n))^T \cdot \tilde{x}^-(n)] = \text{tr}(P^-(n)) \quad (7-2-2)$$

$$\text{MSE}(\hat{x}(n)) = E[\tilde{x}^T(n) \cdot \tilde{x}(n)] = \text{tr}(P(n))$$

۲-۲-۲- یافتن پارامترهای فیلتر کالمن:

:Time Update-

حال فرض شده که تخمین $\hat{x}(n-1)$ موجود است و لازم است با داشتن z^- ، مقدار $\hat{x}(n)$ یافته شود. برای آن ابتدا باید $P^-(n)$ و $\hat{x}^-(n)$ محاسبه شوند.

طبق قضیه ای که برای تخمین LMMSE گفته شد [۳۴] خطای تخمین با Z بصورت زیر رابطه دارد:

$$E[(s - \hat{s})^T z^T] = 0$$

از این قضیه استفاده می شود:

$$\begin{aligned} E[(x(n) - \hat{x}^-(n)) Z^T(i)] &= 0 \\ E[(x(n) - \hat{x}^-(n)) Z_{n-1}^T] &= 0 \end{aligned} \quad (8-2-2)$$

$$Z_{n-1} = \begin{bmatrix} z(1) \\ \vdots \\ z(n-1) \end{bmatrix}$$

معلوم است که:

$$x(n-1) = \hat{x}(n-1) + \tilde{x}(n-1)$$

$$x(n) = \Phi x(n-1) + \Gamma \omega(n-1) = \Phi \hat{x}(n-1) + \Phi \tilde{x}(n-1) + \Gamma \omega(n-1)$$

با بکارگیری رابطه (۸-۲-۲) رابطه جدید خواهد شد:

$$\begin{aligned} E[(\Phi \hat{x}(n-1) - \hat{x}^-(n)) Z_{n-1}^T] + \Phi E[\tilde{x}(n-1) Z_{n-1}^T] + \Gamma E[\omega(n-1) Z_{n-1}^T] &= 0 \Rightarrow \\ E[(\Phi \hat{x}(n-1) - \hat{x}^-(n)) Z_{n-1}^T] &= 0 \Rightarrow \end{aligned}$$

$$\hat{x}^-(n) = \Phi \hat{x}(n-1) \quad (9-2-2)$$

و ماتریس واریانس خطا می شود:

$$P^-(n) = \text{cov}[x(n) - \hat{x}^-(n)] = \text{cov}[\Phi x(n-1) + \Gamma \omega(n-1) - \Phi \hat{x}(n-1)] = \text{cov}[\Phi \tilde{x}(n-1) + \Gamma \omega(n-1)] \quad (10-2-2)$$

و چون $\omega(n-1)$ و $\tilde{x}(n-1)$ مستقل از هم هستند:

$$E[\tilde{x}(n-1) \omega^T(n-1)] = 0$$

در نتیجه رابطه جدید بصورت زیر می شود:

$$P^-(n) = \Phi P(n-1) \Phi^T + \Gamma Q \Gamma^T \quad (11-2-2)$$

این روابط نشان می دهد تخمین $x(n)$ و واریانس P چگونه از زمان $(n-1)$ برای زمان n قابل استخراج هستند قبل از اینکه $Z(n)$ در دسترس باشد.

- Measurement Update :

حال وقتی مشاهده $Z(n)$ در دسترس باشد تخمین $x(n)$ با مشاهدات $Z = \{z(1), \dots, z(n)\}$ قابل محاسبه است. این نشان می دهد که چگونه مشاهده جدید $Z(n)$ برای تخمین $\hat{x}(n)$ و استخراج $P(n)$ استفاده می شود.

همانطور که گفته شد:

$$E[(x(n-1) - \hat{x}(n-1)) Z_{n-1}^T] = 0 \quad (12-2-2)$$

$$\hat{x}(n-1) = \sum_{j=1}^{n-1} H_{n-1}(j) Z(j) = J(n-1) Z_{n-1}$$

که ماتریس $J(n-1)$ به اینصورت تعریف شده است:

$$J(n-1) = [H_{n-1}(1) \quad H_{n-1}(2) \quad \dots \quad H_{n-1}(n-1)]_{N \times P}$$

$$\hat{x}(n) = \sum_{j=0}^n H_n(j) Z(j) \quad (13-2-2)$$

$$E[(x(n) - \hat{x}(n)) Z_n^T] = 0 \quad , \quad Z_n = \begin{bmatrix} Z_{n-1} \\ z(n) \end{bmatrix}$$

با جایگزینی در رابطه (13-2-2) این رابطه بصورت زیر خواهد شد:

$$\hat{x}(n) = k(n)z(n) + G(n)Z_{n-1} \quad (۱۴-۲-۲)$$

$$k(n) = H_n(n)$$

$$G(n) = [H_n(1) \quad H_n(2) \quad \dots \quad H_n(n-1)]$$

حال باید $G(n)$ و $K(n)$ را بدست آید. با خلاصه سازی روابط :

$$G(n) = [1 - k(n)c]\Phi J(n-1) \quad (۱۵-۲-۲)$$

و طبق رابطه روبرو و رابطه (۱۴-۲-۲) رابطه زیر بدست خواهد آمد:

$$\hat{x}(n-1) = J(n-1)Z_{n-1} \quad (۱۶-۲-۲)$$

$$\hat{x}(n) = \hat{x}^-(n) + k(n)[z(n) - c\hat{x}^-(n)]$$

روابط فوق نشان می دهد که $\hat{x}(n)$ قابل محاسبه بصورت برگشتی است از $\hat{x}^-(n)$ و $z(n)$.

: Filter Gain Update -

مرحله سوم پیدا کردن گین فیلتر کالمن است. از روابط قبل:

$$\tilde{x}(n) = \Phi x(n-1) + \Gamma \omega(n-1) - \Phi \hat{x}(n-1) - k(n)z(n) + k(n)c\Phi \hat{x}(n-1)$$

$$E[\tilde{x}(n).Z^T(n)] = 0$$

$$k(n) = P^-(n)c^T [cP^-(n)c^T + R]^{-1} \quad (۱۷-۲-۲)$$

$k(n)$ را گین کالمن گویند. توجه شود $k(n)$ باید قبل از محاسبه $\hat{x}(n)$ محاسبه شود.

در آخر نیز رابطه بین $P(n)$ و $P^-(n)$ ذکر می شود.

$$P(n) = P^-(n) - k(n)cP^-(n) \quad (۱۸-۲-۲)$$

۳-۲-۲- خلاصه ای از روابط فیلتر کالمن:

در این بخش تساویهای فیلتر کالمن دنبال شده است. ابتدا از $n=0$ شروع شده است. باید مقادیر اولیه یافته شود.

$$\hat{x}^-(0) = E[x(0)] \quad (۱۹-۲-۲)$$

$P^-(0)$ نیز می بایست ماتریس مثبت معین باشد:

$$P^-(0) = \lambda I, \quad \lambda > 0$$

- روابط فیلتر:

۱- Measurement Update :

$z(n)$ در دسترس است:

$$A. k(n) = P^-(n)c^T [cP^-(n)c^T + R]^{-1}$$

$$B. \hat{x}(n) = \hat{x}^-(n) + k(n)[z(n) - c\hat{x}^-(n)] \quad (20-2-2)$$

$$C. P(n) = P^-(n) - k(n)cP^-(n)$$

۲- Time Update :

$$A. \hat{x}^-(n+1) = \Phi\hat{x}(n)$$

$$B. P^-(n+1) = \Phi P(n)\Phi^T + \Gamma Q \Gamma^T \quad (21-2-2)$$

سپس n را افزایش داده و دوباره روابط شروع شده و به مرحله ۱ باز گشته است. واضح است:

$$\hat{s}(n) = c\hat{x}(n)$$

$$\hat{s}^-(n+1) = c\hat{x}^-(n+1)$$

پس ابتدا $P^-(0), \hat{x}^-(0)$ یک مقدار از پیش تعیین شده فرض شده است. سپس $n=1$ می شود و

روابط بازگشتی کالمن آغاز می شود. از روابط Time Update $P^-(n), \hat{x}^-(n)$ بدست می آید و با

داشتن $z(n)$ روابط $k(n)$ و $\hat{x}(n)$ و $P(n)$ محاسبه شده و n افزایش یافته است.

۲-۲-۴- خطای سیستم:

در اینجا پایان نامه در مورد خطای تخمین یعنی $\tilde{x}(n)$ بحث کرده است.

$$E[\tilde{x}(n)] = E[x(n) - \hat{x}(n)] = [1 - k(n)c]\Phi.E[\tilde{x}(n-1)]$$

$$E[\tilde{x}(n)] = [1 - k(n)c]E[\tilde{x}^-(n)] \quad (22-2-2)$$

این روابط کمک کرده تا در مورد رفتار خطای سیستم مطالعه بهتری صورت گیرد. در اینجا لازم

است یک رابطه مهم در مورد ماتریس $P^-(n)$ و $P^-(n-1)$ بیان شود.

$$P^-(n+1) = \Phi P(n)\Phi^T + \Gamma Q \Gamma^T = \Phi P^-(n)\Phi^T - \Phi k(n)cP^-(n)\Phi^T + \Gamma Q \Gamma^T$$

با جایگزینی رابطه $k(n)$ در این رابطه، رابطه جدید بصورت زیر خواهد شد:

$$P^-(n+1) = \Phi \left\{ P^-(n) - P^-(n)c^T (cP^-(n)c^T + R)^{-1} cP^-(n) \right\} \Phi^T + \Gamma Q \Gamma^T \quad (2-2-23)$$

این تساوی را معادله ریکاتی می نامند.

۲-۲-۵- خصوصیات فیلتر کالمن:

این بخش به چند نکته مهم در مورد فیلتر کالمن اشاره کرده است. خصوصیات کلی این فیلتر عبارت است از:

۱- فیلتر کالمن سیستمی متغیربا زمان است. اگر ماتریسهای روابط فضای حالت (R, c, Φ, Γ) ثابت باشند $P(n)$ و $P^-(n)$ فقط به n بستگی دارد.

۲- با توجه به روابط فیلتر کالمن ما دو مورد خاصیت برگشتی مستقل داریم که گین $k(n)$ در هر دو مورد وجود دارد:

- روابط مربوط به $\hat{x}^-(n+1), \hat{x}(n), k(n)$ که آنها را تخمین برگشتی می نامند.
- روابط مربوط به $P^-(n+1), P(n), k(n)$ واریانس خطا را محاسبه می کند که واریانس برگشتی می نامند و با استفاده از آن می توان میانگین مربعات خطا را یافت. توجه شود که $P(n)$ مستقل از مشاهده $z(n)$ است و به همین علت بدون داشتن $z(n)$ می توان MSE را طبق روابط (۲-۲-۸) بدست آورد.

این توضیحات مختصری در مورد چگونگی استخراج روابط فیلتر کالمن و خود روابط بازگشتی فیلتر کالمن بود. شبیه سازیهای مربوط به این فیلتر با موضوع تخمین حالت‌های یک سیستم دینامیکی (Tracker) در [۳۴] به تفصیل آمده است لذا از ارائه شبیه سازیهای مربوط به این فیلتر اجتناب می کنیم. اما در بخش بعد مروری بر فیلتر کالمن توسعه یافته خواهد شد.

۲-۳- فیلتر کالمن توسعه یافته (EKF):

در بخش ۲-۲ در مورد سیستمهای خطی و تخمین حالت‌های آنها توسط فیلتر کالمن بحث شد. در حالیکه بسیاری از سیستمهای فیزیکی دارای دینامیک غیرخطی هستند و در نتیجه نیاز به فیلتری بر مبنای محاسبات غیر خطی جهت تخمین حالت‌های آنها می باشد. این مسئله در بسیاری از سیگنالهای صوتی نیز به چشم می خورد. یعنی بسیاری از سیگنالهای صوتی دارای پدیده های غیرخطی آیرودینامیکی هستند که توسط مدل‌های خطی قابل مدل کردن و از طریق معادلات خطی کالمن قابل تخمین نمی باشند. در این مواقع اغلب از شبکه عصبی برای مدل کردن سیستم غیرخطی استفاده می شود و از فیلتر کالمن توسعه یافته نیز جهت تخمین حالت‌های سیستم استفاده می شود.

در این بخش فضای حالت یک سیستم غیرخطی ثابت با زمان در نظر گرفته شده است:

$$\begin{aligned} x(n+1) &= \phi(x(n)) + \Gamma\omega(n) \\ z(n) &= \gamma(x(n)) + v(n) \end{aligned} \quad (1-3-2)$$

چون Γ یک ماتریس است بنابراین دینامیکهای ورودی خطی و مستقل از $x(n)$ است و اپراتورهای $\phi(x)$ و $\gamma(x)$ توابع غیرخطی از x و n می باشند.

همچنین فرض شده که x دارای m حالت و z دارای p حالت باشد در نتیجه:

$$\phi(x) = \begin{bmatrix} \phi_1(x) \\ \phi_2(x) \\ \vdots \\ \phi_m(x) \end{bmatrix} \quad \gamma(x) = \begin{bmatrix} \gamma_1(x) \\ \gamma_2(x) \\ \vdots \\ \gamma_p(x) \end{bmatrix}$$

چون فیلتر کالمن معمولی یک SMM خطی را در نظر می گیرد بنابراین اولین قدم، خطی سازی SMM مطرح شده در بالا می باشد. همچنین فرض شده ϕ و γ به اندازه کافی در x هموار هستند. بنابراین هر کدام دارای بسط تیلور معتبر می باشند.

اگر $\hat{x}(n) = \hat{x}(n)$ باشد آنگاه بسط تیلور ϕ حول $\hat{x}(n)$ بصورت زیر است:

$$\phi(x(n)) = \phi(\hat{x}(n)) + J_\phi(\hat{x}(n))[x(n) - \hat{x}(n)] + \dots \quad (2-3-2)$$

$$J_{\phi}(x) = \frac{\partial \phi}{\partial x} = \begin{bmatrix} \frac{\partial \phi_1}{\partial x_1} & \frac{\partial \phi_1}{\partial x_2} & \dots & \frac{\partial \phi_1}{\partial x_N} \\ \frac{\partial \phi_2}{\partial x_1} & \frac{\partial \phi_2}{\partial x_2} & \dots & \frac{\partial \phi_2}{\partial x_N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \phi_m}{\partial x_1} & \frac{\partial \phi_m}{\partial x_2} & \dots & \frac{\partial \phi_m}{\partial x_N} \end{bmatrix}$$

به همین ترتیب بسط تیلور γ حول $\hat{x}^-(n)$ بصورت زیر است:

$$\gamma(x(n)) = \gamma(\hat{x}^-(n)) + J_{\phi}(\hat{x}^-(n)) [x(n) - \hat{x}^-(n)] + \dots \quad (3-3-2)$$

در بسط های آمده در بالا فقط دو ترم اول عبارت را در نظر می گیریم. نتیجه آنها تقریب مرتبه اول ϕ و Γ خواهد بود که یک تقریب خطی از x می باشد.

در نتیجه SMM جدید خطی شده بصورت زیر است:

$$\begin{aligned} x(n+1) &= \phi(\hat{x}(n)) + J_{\phi}(\hat{x}(n)) [x(n) - \hat{x}(n)] + \Gamma \omega(n) \\ z(n) &= \gamma(\hat{x}^-(n)) + J_{\gamma}(\hat{x}^-(n)) [x(n) - \hat{x}(n)] + v(n) \end{aligned} \quad (4-3-2)$$

بستگی دارد که در مراحل بعد تعیین می شوند. $\hat{x}^-(n)$ و $\hat{x}(n)$ بالا به تخمینهای SMM که البته

: Time update -

در ابتدا فرض می کنیم $\hat{x}(n-1)$ غیر بایاس است و می خواهیم $\hat{x}^-(n)$ نیز غیر بایاس باشد. در نتیجه خواهیم داشت:

$$\begin{aligned} E[x(n) - \hat{x}^-(n) | Z = Z^-] &= 0 \\ \hat{x}^-(n) &= E[x(n) | Z = Z^-] = E[\phi(\hat{x}(n-1)) + J_{\phi}(\hat{x}(n-1)) [x(n-1) - \hat{x}(n-1)] + \Gamma \omega(n-1) | Z = Z^-] \\ \Gamma \omega(n-1) | Z = Z^- &= 0 \end{aligned} \quad (5-3-2)$$

حال اگر فرض کنیم $\hat{x}(n-1)$ برداری ثابت باشد داریم:

$$\hat{x}^-(n) = \phi(\hat{x}(n-1)) + J_{\phi}(\hat{x}(n-1)) E[x(n-1) - \hat{x}(n-1) | Z = Z^-] + \Gamma E[\omega(n-1) | Z = Z^-]$$

چون $\hat{x}(n-1)$ غیر بایاس است پس:

$$E[x(n-1) - \hat{x}(n-1) | Z = Z^-] = 0 \quad (6-3-2)$$

و چون $\omega(n-1)$ از $z(1), \dots, z(n-1)$ مستقل است بنابراین:

$$E[\omega(n-1)|Z = Z^-] = 0 \quad (7-3-2)$$

سپس تخمین خود را به شکل زیر خواهیم داشت:

$$\hat{x}^-(n) = \phi(\hat{x}(n-1)) \quad (8-3-2)$$

که در واقع ماتریس غیر خطی ϕ همان شبکه عصبی MLP است که $\hat{x}(n-1)$ ورودی آن است نقش یک تابع غیرخطی از متغیر حالت تخمینی را برعهده دارد. در مورد روش آموزش شبکه و کاربرد آن در شناسایی سیستم غیر خطی مدل رابطه (2-3-1) در فصل سوم بحث خواهد شد.

ذکر این نکته لازم است که در اکثر سیستمهای واقعی γ همچنین در مورد ماتریس غیر خطی و مقالات بررسی شده که در فصل اول در مورد آنها بحث شد، این فاکتور را بصورت یک ماتریس در نظر می گیرند و فرض بر این است که مشاهده ما یک مشاهده خطی از حالتهاى C خطی سیستم است. لذا در این پایان نامه نیز این فاکتور بصورت خطی در نظر گرفته شده است.

همچنین برای محاسبه ماتریس واریانس خطا خواهیم داشت:

$$P^-(n) = Cov[x(n) - \hat{x}^-(n)|Z = Z^-] = E[(x(n) - \hat{x}^-(n))(x(n) - \hat{x}^-(n))^T | Z = Z^-] \quad (9-3-2)$$

با استفاده از شرط ثابت بودن $\hat{x}(n-1)$ و بکارگیری رابطه (1-4-8) داریم:

$$P^-(n) = J_\phi(\hat{x}(n-1))Cov[\tilde{x}(n-1)|Z = Z^-]J_\phi^T(\hat{x}(n-1)) + \Gamma Q(n-1)\Gamma^T \quad (10-3-2)$$

حال از تعریف زیر استفاده شده:

$$P(n) = Cov[\tilde{x}(n)|Z = Z] = Cov[x(n) - \hat{x}(n)|Z = Z] \quad (11-3-2)$$

و در نهایت با استفاده از روابط (2-3-10) و (2-3-11) ماتریس واریانس خطا محاسبه گردیده است:

$$P^-(n) = J_\phi(\hat{x}(n-1))P(n-1)J_\phi^T(\hat{x}(n-1)) + \Gamma Q(n-1)\Gamma^T \quad (12-3-2)$$

که ماتریس ژاکوبین J_ϕ ، همان ماتریسی است که متشکل از آرایه های کسری است که صورت آنها خروجی شبکه عصبی و مخرج آنها ورودی شبکه عصبی است که البته بسته به تعداد ورودیها

صورت کسر و مخرج آن می توانند به ترتیب تفاضل دو خروجی و ورودی متوالی شبکه عصبی باشند. که در مورد این ماتریس و نحوه تشکیل آن در فصل چهارم به تفصیل بحث خواهد شد.

: Measurement update-

در ابتدا فرض شده $\hat{x}(n)$ دارای فرم زیر و غیر بایاس باشد:

$$\hat{x}(n) = b(n) + K(n)z(n) \quad (13-3-2)$$

$$E[x(n) - \hat{x}(n)|Z = Z^-] = 0$$

با استفاده از روابط (13-3-2) و (5-3-2) داریم:

$$E[x(n) - b(n) - K(n)[\gamma(\hat{x}^-(n)) + J_\gamma(\hat{x}^-(n))\tilde{x}^-(n) + v(n)]|Z = Z^-] = 0$$

ثابت است داریم: $\hat{x}^-(n)$ حال با فرض اینکه

$$b(n) = E[b(n)|Z = Z^-] = -K(n)\gamma(\hat{x}^-(n)) - K(n)J_\gamma(\hat{x}^-(n))E[\tilde{x}^-(n)|Z = Z^-]$$

$$+ E[x(n)|Z = Z^-] - K(n)E[v(n)|Z = Z^-]$$

و چون $\hat{x}^-(n)$ غیر بایاس است و $v(n)$ مستقل از Z است در نتیجه خواهیم داشت:

$$b(n) = -K(n)\gamma(\hat{x}^-(n)) + E[x(n)|Z = Z^-] \quad (14-3-2)$$

با استفاده از (5-3-2) خواهیم داشت:

$$b(n) = \hat{x}^-(n) - K(n)\gamma(\hat{x}^-(n)) \quad (15-3-2)$$

در نتیجه رابطه (13-3-2) بصورت زیر خواهد شد:

$$\boxed{\hat{x}(n) = \hat{x}^-(n) + K(n)[z(n) - \gamma(\hat{x}^-(n))]} \quad (16-3-2)$$

با استفاده از اصل تعامد و فرض عمود بودن $\tilde{x}^-(n)$ و $z(1), \dots, z(n-1)$ خواهیم داشت:

$$E[(x(n) - \hat{x}(n))z^T(i)|Z = Z] = 0 \quad (17-3-2)$$

با ترکیب (16-3-2) و (17-3-2) خواهیم داشت:

$$E\left\{[x(n) - \hat{x}^-(n) - K(n)[z(n) - \gamma(\hat{x}^-(n))]]z^T(i)|Z = Z\right\} = 0$$

با استفاده از رابطه (4-4-1) و خاصیت ثابت بودن بردار $\hat{x}^-(n)$ و غیربایاس بودن و مستقل از $z(n)$

بودن آن خواهیم داشت:

$$E[\tilde{x}^-(n)|Z = Z] = E[\tilde{x}^-(n)|Z^- = Z^-] = 0 \quad (18-3-2)$$

$$P^-(n)J_\gamma^T(\hat{x}^-(n)) - K(n)J_\gamma^T(\hat{x}^-(n))P^-(n)J_\gamma^T(\hat{x}^-(n)) - K(n)R(n) = 0$$

از حل معادله بالا برای $K(n)$ خواهیم داشت:

$$K(n) = P^-(n)J_\gamma^T(\hat{x}^-(n)) \left[J_\gamma(\hat{x}^-(n))P^-(n)J_\gamma^T(\hat{x}^-(n)) + R(n) \right]^{-1} \quad (19-3-2)$$

و از روابط (11-3-2) و (19-3-2) و (4-3-2) داریم:

$$P(n) = P^-(n) - K(n)J_\gamma(\hat{x}^-(n))P^-(n) \quad (20-3-2)$$

در نتیجه با خلاصه کردن روابط بدست آمده در بالا خواهیم داشت:

: Measurement update-1

$$A. K(n) = P^-(n)J_\gamma^T(\hat{x}^-(n)) \left[J_\gamma(\hat{x}^-(n))P^-(n)J_\gamma^T(\hat{x}^-(n)) + R(n) \right]^{-1}$$

$$B. \hat{x}(n) = \hat{x}^-(n) + K(n) \left[z(n) - \gamma(\hat{x}^-(n)) \right] \quad (21-3-2)$$

$$C. P(n) = P^-(n) - K(n)J_\gamma(\hat{x}^-(n))P^-(n)$$

: Time update -2

$$A. P^-(n+1) = J_\phi(\hat{x}(n))P(n)J_\phi^T(\hat{x}(n)) + \Gamma Q(n)\Gamma^T \quad (22-3-2)$$

$$B. \hat{x}^-(n+1) = \phi(\hat{x}(n))$$

: Time increment -3

n را افزایش داده و به مرحله 1 باز می گردد.

توجه به این نکته قابل اهمیت است که $J_\phi(\hat{x}(n))$ و $J_\gamma(\hat{x}^-(n))$ و ماتریسهای کوواریانس

$P(n)$ و $P^-(n)$ نیازمند تخمینهای $\hat{x}(n)$ و $\hat{x}^-(n)$ می باشند. در نتیجه فیلتر کالمن توسعه یافته

نمی تواند بصورت off-line عمل کند و نیازمند اطلاعات بهنگام می باشد.

لازم بذکر است کاربرد این فیلتر در بهبود سیگنالهای صوتی و چگونگی پیاده سازی و شبیه

سازیهای آن بطور کامل در فصل چهارم آمده است.

۴-۲- فیلتر کالمن توسعه یافته دوگانه (DEKF):

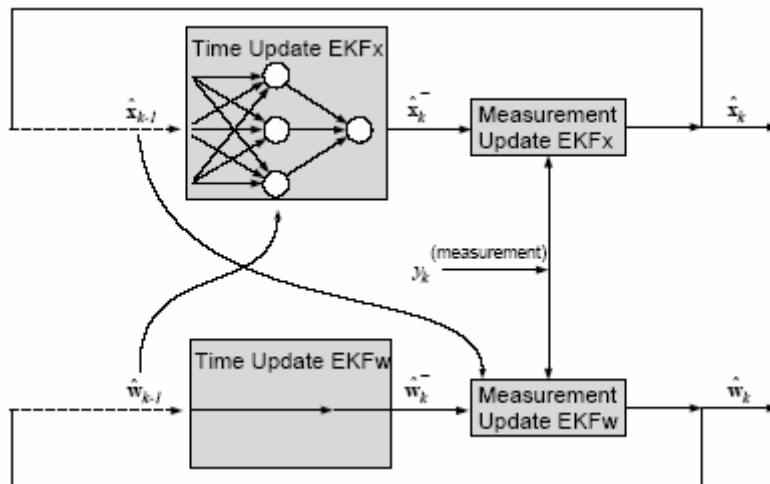
این بخش سیستم دینامیکی غیرخطی زمان گسسته را بصورت زیر در نظر گرفته است:

$$\begin{aligned} x_{k+1} &= F(x_k, u_k, w) + v_k \\ y_k &= H(x_k, w) + n_k \end{aligned} \quad (1-4-2)$$

که x_k حالت سیستم دینامیکی غیر خطی است که با نویز فرآیند v_k تخریب شده است و سیگنال y_k مشاهده سیستم ماست که با نویز مشاهده n_k تخریب گشته است.

همان شبکه های عصبی هستند که پس از آموزش دیدن و مدلسازی H و F و اما توابع غیر خطی سیستم جایگزین این توابع غیر خطی گشته و به عنوان مدل سیستم دینامیکی غیر خطی نقش ایفا نیز به عنوان یک پارامتر داخل آنها گنجانده شده است. w می کنند به همین دلیل پارامتر مسئله اصلی ما در این بخش یافتن یک تخمین بهینه از حالتها و وزنها سیستم دینامیکی است. لذا فیلتر کالمن دوگانه جهت این امر معرفی می گردد. بنابراین در ابتدا تعریفی از فیلتر کالمن دوگانه ارائه می گردد.

فیلتر کالمن دوگانه متشکل از دو فیلتر حالت و وزن است. فیلتر حالت تخمینهای سیگنال را با توجه به معلوم بودن مدل تهیه می کند و فیلتر وزن تخمینهای پارامترها را با فرض معلوم بودن سیگنال بدست می دهد نمایش این فیلتر بصورت زیر است: [۳۳]



شکل ۴-۲-۱- نمایش فیلتر کالمن توسعه یافته دوگانه [۳۳]

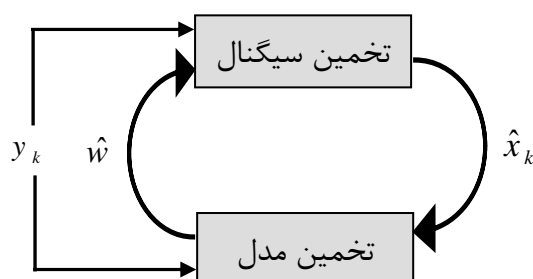
که البته دستیابی به نوع ارتباط ۲ فیلتر وزن و حالت پس از بیان روابط ریاضی آنها مشخص تر خواهد گردید. اما آنچه که از شکل بالا مشخص است، هر کدام از فیلترها دارای معادلات بروز آوری زمان. بروز آوری اندازه گیری مختص خود هستند. همچنین این دو فیلتر بصورت موازی با یکدیگر کار می کنند که مفهوم آن بصورت زیر است:

حالت \hat{x}_{k-1} که بعنوان ورودی تابع غیرخطی F (شبکه عصبی شکل بالا) در معادلات بروز آوری زمان فیلتر حالت بکار می رود بطور همزمان در معادلات بروز آوری اندازه گیری فیلتر وزن نیز بکار می رود.

وزن \hat{w}_{k-1} که به عنوان ورودی معادلات بروز آوری زمان فیلتر وزن بکار می رود بطور همزمان در معادلات بروز آوری زمان فیلتر حالت جهت تنظیم وزنهای شبکه عصبی (تابع غیر خطی F) بکار می رود.

مشاهده y_k نیز به عنوان ورودی به هر دو فیلتر اعمال می گردد و در نهایت فیلتر وزن تخمین \hat{w}_k و فیلتر حالت، تخمین \hat{x}_k را ارائه می دهد و این تخمینها با یک فیدبک به معادلات بروز آوری زمان دو فیلتر اعمال می گردند و این سیکل با تکرار مورد نظر طراح تا همگرا شدن الگوریتم ادامه می یابد.

لذا این فیلتر یک تخمین ترتیبی ارائه می دهد که بلوک دیاگرام آن بصورت زیر است:



شکل ۲-۴-۲- نمایش تخمین ترتیبی جهت مسئله تخمین دوگانه

مزیت دیگر فیلتر کالمن دوگانه قابلیت آن در ارائه تخمینهای دقیق و بهینه از سیگنال تمیز بدون دسترسی به اطلاعات سیگنال تمیز است. یعنی در واقع فیلتر کالمن دوگانه می تواند با فقط دسترسی به سیگنال نویزی، تخمینهای دقیق از سیگنال تمیز را به ما ارائه دهد. که علت این امر متشکل بودن فیلتر دوگانه از ۲ فیلتر وزن و حالت است که در هر زمان با دریافت اطلاعات از یک فیلتر، فیلتر دیگر اقدام به ارائه تخمین مناسب می نماید.

۲-۴-۱- فیلتر تخمین حالت کالمن توسعه یافته :

فیلتر تخمین حالت دارای معادلاتی کاملا مشابه با فیلتر کالمن توسعه یافته بخش قبل است. در نتیجه فقط به ارائه روابط ریاضی آن در این بخش بسنده شده است: [۳۳]

- Initialize :

$$\begin{aligned} \hat{x}_0 &= E[x_0] \\ P_{x_0} &= E[(x_0 - \hat{x}_0)(x_0 - \hat{x}_0)^T] \end{aligned} \quad (2-4-2)$$

که در واقع \hat{x}_0 مقدار اولیه جهت تخمین حالت سیستم و P_{x_0} مقدار اولیه کوواریانس حالت تخمینی می باشد. [۳۳]

- Time Update :

$$\begin{aligned} \hat{x}_k^- &= F(\hat{x}_{k-1}, u_k, w) \\ P_{x_k}^- &= A_{k-1} P_{x_{k-1}} A_{k-1}^T + R^v \end{aligned} \quad (3-4-2)$$

که F نمایانگر شبکه عصبی MLP است که پس از آموزش جایگزین تابع غیر خطی گردیده است و

A_k ماتریس خطی شده تابع غیر خطی F است که بصورت زیر محاسبه می گردد: [۳۳]

$$A_k = \left. \frac{\partial F(x, u_k, w)}{\partial x} \right|_{\hat{x}_k} \quad (4-4-2)$$

همچنین ماتریس R^v ، ماتریس کوواریانس نویز فرآیند می باشد.

Measurement Update-

$$\begin{aligned} K_k^x &= P_{x_k}^- C_k^T (C_k P_{x_k}^- C_k^T + R^n)^{-1} \\ \hat{x}_k &= \hat{x}_k^- + K_k^x [y_k - H(\hat{x}_k^-, w)] \\ P_{x_k} &= (I - K_k^x C_k) P_{x_k}^- \end{aligned} \quad (5-4-2)$$

در اینجا H نیز نمایانگر شبکه عصبی MLP است و ماتریس R^n ، ماتریس واریانس نویز مشاهده می باشد. [۳۳]

می باشد و از فرمول زیر محاسبه می گردد: H ماتریس خطی شده تابع غیر خطی C_k همچنین

$$C_k = \left. \frac{\partial H(x, w)}{\partial x} \right|_{\hat{x}_k} \quad (6-4-2)$$

۲-۴-۲- فیلتر تخمین وزن کالمن توسعه یافته :

همانطور که در فصل اول نیز ذکر شد فیلتر کالمن توسعه یافته می تواند جهت تخمین پارامترهای مدل‌های غیر خطی (آموزش شبکه های عصبی) نیز بکار رود.

فرض کنید مسئله، یک تصویر سازی عمومی با بکار گیری تابع غیر خطی پارامتری شده $G(x_k, w)$ باشد. همچنین مجموعه آموزشی ما شامل ورودیهای شناخته شده و خروجیهای خواسته شده $\{x_k, d_k\}$ باشد. در نتیجه خطا در مدل مذکور با رابطه زیر بیان می شود:

$$e_k = d_k - G(x_k, w) \quad (7-4-2)$$

و هدف یادگیری ارائه راه حلی جهت پارامتر w است تا خطای مربعات مورد نظر را حداقل کند. در نتیجه EKF بکارگیری می شود تا پارامترهای w را با نوشتن معادلات فضای تخمین بزند. صورت معادلات به شرح زیر است:

$$\begin{aligned} w_{k+1} &= w_k + r_k \\ d_k &= G(x_k, w) + e_k \end{aligned} \quad (8-4-2)$$

که w_k به یک فرآیند ایستا با ماتریس انتقال حالت مشخص مربوط است که با نویز فرآیند r_k جمع شده است.

همچنین خروجی d_k مربوط به مشاهده غیر خطی روی w_k است.

سپس فیلتر کالمن توسعه یافته می تواند بطور مستقیم جهت تخمین پارامترهای سیستم بکارگیری شود. در نتیجه روابط ریاضی فیلتر وزن کالمن توسعه یافته نیز بصورت زیر است:

- Initialize :

$$\begin{aligned}\hat{w}_0 &= E[w] \\ P_{w_0} &= E[(w_0 - \hat{w}_0)(w_0 - \hat{w}_0)^T]\end{aligned}\quad (9-4-2)$$

که در واقع \hat{w}_0 مقدار اولیه جهت تخمین پارامترهای سیستم و P_{w_0} مقدار اولیه کوواریانس وزن تخمینی می باشد. [۳۳]

- Time Update :

$$\begin{aligned}\hat{w}_k^- &= \hat{w}_{k-1} \\ P_{w_k}^- &= P_{w_{k-1}} + R_{k-1}^r\end{aligned}\quad (10-4-2)$$

که ماتریس R_k^r ، ماتریس کوواریانس نویز فرآیند می باشد. [۳۳]

- Measurement Update :

$$\begin{aligned}K_k^w &= P_{w_k}^- (C_k^w)^T (C_k^w P_{w_k}^- (C_k^w)^T + R^e)^{-1} \\ \hat{w}_k &= \hat{w}_k^- + K_k^w [d_k - G(\hat{w}_k^-, x_{k-1})] \\ P_{w_k} &= (I - K_k^w C_k^w) P_{w_k}^-\end{aligned}\quad (11-4-2)$$

در اینجا G نیز نمایانگر شبکه عصبی MLP است و ماتریس R^e ، ماتریس واریانس نویز مشاهده می باشد. [۳۳]

همچنین C_k^w ماتریس خطی شده تابع غیر خطی G می باشد و از فرمول زیر محاسبه می گردد:

$$C_k^w = \left. \frac{\partial G(x_{k-1}, w)}{\partial w} \right|_{\hat{w}_k^-} \quad (12-4-2)$$

بطور معمول مقدار انتخابی برای R^e بصورت زیر است:

$$R^e = 0.5 I$$

همچنین چند انتخاب برای R_k^r وجود دارد که در ذیل به آنها اشاره می گردد: [۳۳]

- R_k^f را با یک ماتریس قطری قراردادی تنظیم می شود تا دلالت بر غیر وابستگی پارامترها کند و سپس به سمت صفر میل داده می شود تا یادگیری ادامه یابد.
- $R_k^f = (\lambda^{-1} - 1)P_{w_k}$ قرار داده می شود که $\lambda \in (0,1]$ اغلب به عنوان فاکتور فراموشی بکار می رود.
- $R_k^f = (1 - \alpha)R_k^f + \alpha K_k^w [d_k - G(x_k, \hat{w})][d_k - G(x_k, \hat{w})]^T (K_k^w)^T$ قرار داده می شود که به نام فرمول روبینز-مونورو مشهور است [۳۳].

که البته اغلب مقدار آیتم دوم برای واریانس نویز فرآیند R_k^f انتخاب می گردد.

در پایان این فصل ذکر این نکته لازم است که توضیحات تکمیلی در مورد چگونگی استخراج روابط فیلتر وزن و حالت کالمن توسعه یافته در مرجع [۳۳] آمده است.

همچنین در فصل بعد بطور کامل در مورد بحث شناسایی سیستمها و مدلسازی سیستمهایی که بایستی حالت‌های خطی و غیر خطی آنها توسط فیلتر کالمن خطی و فیلتر کالمن توسعه یافته تخمین زده شود صحبت خواهد شد.

۳ شناسایی سیستم‌های خطی و غیر خطی

۳-۱- مقدمه:

شناسایی سیستم، مسأله ساختن مدل ریاضی سیستم‌های دینامیکی است که بر پایه مشاهدات از داده های سیستم می باشد. یعنی تخمین یک مدل از سیستم بر پایه اطلاعات ورودی-خروجی می باشد.

پس ابتدا برای تعیین هویت یک سیستم، دینامیک آنرا باید به خوبی بررسی کرد. یک سیستم شامل ورودی یا ورودیها و یک سری سیگنالهای قابل مشاهده به عنوان خروجی می باشد. در نتیجه برای یک سیستم باید بتوان به درستی ورودیها و خروجیهای سیستم را تشخیص داد.

در مواردی که دسترسی به محرک بیرونی سیستم به عنوان ورودی امکان پذیر است، ورودی سیستم شناخته می شود و همچنین مشاهدات سیستم، معمولاً به عنوان خروجی سیستم در نظر گرفته می شود. بعد از این هرچه بماند به عنوان نویز است که ممکن است در مشاهدات سیستم باشد یا با ورودی وارد شود. به هر حال برای اینکه بتوان مدلی از سیستم ارائه داد اولین مرحله این است که دینامیک آن سیستم به خوبی شناخته شود و اطلاعات کافی از سیستم جمع شود.

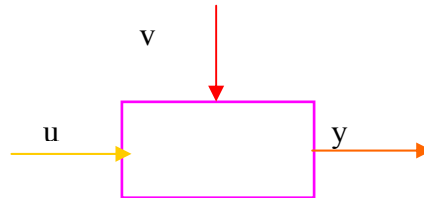
مرحله بعد لازم است که روابط بین پارامترهای سیستم را تشخیص داد و در واقع باید رابطه بین ورودی و خروجی یافته شده و سیستم مدل شود. برای یک سیستم ممکن است بتوان مدلهای مختلفی ارائه داد که باید با توجه به اطلاعاتی که از سیستم وجود دارد بهترین مدل را انتخاب کرد. بعد از انتخاب مدل، سؤال اصلی این است که آیا مدل به اندازه کافی مشابه واقعیت سیستم می باشد و انتظار ما از سیستم برآورده می شود که با انجام تستهایی روی مدل این مسأله قابل تشخیص است.

پس در واقع شناسایی سیستمها شامل سه مرحله عمده است:

۱. ابتدا اطلاعاتی را که از سیستم بدست آمده جمع آوری می شود.
 ۲. سپس باید مدلهای مناسب برای تخمین سیستم یافته شده و از بین آنها بهترین مدل انتخاب شود.
 ۳. و در آخر باید بررسی شود که آیا مدل به اندازه کافی برای هدف خوب بوده که این مسأله در بحث اعتبار مدل انجام می شود.
- این فصل ابتدا به روشهای شناسایی سیستمهای خطی و سپس روشهای شناسایی سیستمهای غیر خطی می پردازد. در بخش شناسایی سیستمهای خطی به روشهای مختلف اعتبار تخمین پرداخته می شود و بعد از آن به روش مناسب مدل کردن سیگنالهای صوتی پرداخته خواهد شد. در بخش شناسایی سیستمهای غیر خطی نیز کاربرد شبکه های عصبی در شناسایی سیستمها و نمونه هایی از شبیه سازیهای انجام شده در مورد شناسایی سیگنالهای صوتی معرفی می گردد.
- همچنین بدلیل توضیح کامل در مورد ۳ روش عمده تخمین یعنی روش حداقل مربعات، حداقل مربعات بازگشتی و روش حداقل مربعات تعمیم یافته در [۳۴]، لذا اشاره ای به این روشها در این فصل نشده است. بعد از آن به روش مناسب مدل کردن سیگنالهای صوتی پرداخته شده و در آخر به روشهای مختلف اعتبار تخمین و شبیه سازی آنها می پردازد.

۳-۲- ساختارهای مدل کردن سیستمهای خطی:

تخمین یک مدل از یک سیستم بر پایه اطلاعات ورودی-خروجی است. ساختار ورودی و خروجی سیستمها را به شکل زیر در نظر می گیریم:



برای این سیستم ، u ورودی سیستم و v تابعی از نویز است.

$$y(t) = G(q, \theta).u(t) + H(q, \theta).e(t) \quad (۱-۲-۳)$$

$$v(t) = H(q, \theta).e(t)$$

در اینجا پیش بینی تخمین یک پله از $y(t)$ محاسبه می شود:

$$\hat{y}(t|t-1) = G(q, \theta).u(t) + \hat{v}(t|t-1) = G(q, \theta).u(t) + [1 - H^{-1}(q, \theta)]v(t) \quad (۲-۲-۳)$$

$$\hat{y}(t|t-1) = H^{-1}(q, \theta).G(q, \theta).u(t) + [1 - H^{-1}(q, \theta)]y(t)$$

که در اینصورت خطای پیشگویی برابر است با:

$$y(t) - \hat{y}(t|t-1) = e(t) \quad (۳-۲-۳)$$

راههای گوناگونی برای پارامتریک کردن G و H وجود دارد. عمده راهها این است که پارامترها بصورت توابع گویا نوشته شود و آنها را در صورت و مخرج بگذاریم. این مدلها را معادلات خطا می نامند. که به چند نمونه از آنها اشاره شده است:

۱- مدل ARX ^{۲۰}:

ساده ترین حالتی که بین ورودی و خروجی می تواند برقرار باشد این است که آنها بصورت یک رابطه خطی ساده نوشته شود:

$$y(t) + a_1 y(t-1) + \dots + a_{n_a} y(t-n_a) = b_1 u(t-1) + \dots + b_{n_b} u(t-n_b) + e(t) \quad (۴-۲-۳)$$

²⁰ Auto Regressive External Input

در این مدل $e(t)$ نویز سفید است. پارامترهایی که در این معادله باید تعیین شود:

$$\theta = [a_1 \ a_2 \ \dots \ a_{n_a} \quad b_1 \ b_2 \ \dots \ b_{n_b}]^T$$

اگر $A(q)$ و $B(q)$ بصورت زیر تعریف شوند:

$$A(q) = 1 + a_1 q^{-1} + \dots + a_{n_a} q^{-n_a}$$

$$B(q) = b_1 q^{-1} + \dots + b_{n_b} q^{-n_b}$$

آنگاه پارامترهای G و H بصورت پارامتریک زیر می شوند:

$$G(q, \theta) = \frac{B(q)}{A(q)}, \quad H(q, \theta) = \frac{1}{A(q)}$$

در نتیجه معادله (۲-۵-۴) بصورت زیر بازنویسی می شود:

$$A(q)y(t) = B(q)u(t) + e(t) \quad (۲-۵-۳)$$

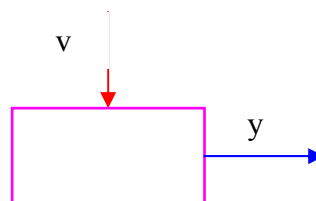
مدل معادله بالا مدل ARX نامیده می شود. [۳۱]

این مدل بسیار مهم است و در بسیاری از کاربردها اولین انتخاب ما می باشد.

فرم ساده تر مدل بالا بدین صورت است که پارامتر $B(q)=0$ باشد، در نتیجه مدل بصورت

$A(q).y(t) = e(t)$ در می آید. این مدل را مدل AR می گویند که در واقع در این مدل ورودی

نداریم. [۳۱]



و مدل ARX با توجه به پارامترهای $G(q)$ و $H(q)$ در معادله (۲-۵-۲) خواهیم شد:

$$\hat{y}(t) = B(q).u(t) + [1 - A(q)].y(t) \quad (۲-۵-۶)$$

یا می توان بجای این معادله روی مسئله زیر که همان تخمین بازگشتی است کار شود:

$$\theta = \begin{bmatrix} a_1 \\ \vdots \\ a_{n_a} \\ b_1 \\ \vdots \\ b_{n_b} \end{bmatrix}, \quad \varphi(t) = [-y(t-1) \quad \dots \quad -y(t-n_a) \quad u(t-1) \quad \dots \quad u(t-n_b)]^T$$

در اینصورت تخمین $y(t)$ بصورت زیر خواهد شد:

$$\hat{y}(t) = \varphi^T(t) \cdot \theta \quad (7-2-3)$$

که $\varphi(t)$ شناخته شده است و θ پارامترهای ناشناخته است. نوشتن معادله بصورت (7-5-2) را مدل خطی بازگشتی گویند.

۲- مدل ARMAX^{۲۱}:

در یک راه دیگر مدل کردن، می توان درجه آزادی کار را بیشتر کرد. در این راه می توان معادله خطا را روی نویز سفید گسترش داد:

$$y(t) + a_1 y(t-1) + \dots + a_{n_a} y(t-n_a) = b_1 u(t-1) + \dots + b_{n_b} u(t-n_b) + e(t) + c_1 e(t-1) + \dots + c_{n_c} e(t-n_c) \quad (8-2-3)$$

پارامتر $C(q)$ بصورت زیر تعریف می شود:

$$C(q) = 1 + c_1 q^{-1} + \dots + c_{n_c} q^{-n_c}$$

و پارامترهای G و H بصورت پارامتریک زیر تعریف می شوند:

$$G(q, \theta) = \frac{B(q)}{A(q)}, \quad H(q, \theta) = \frac{C(q)}{A(q)}$$

با جایگزینی پارامترهای بالا در معادله (1-5-2) خواهیم داشت:

$$\boxed{A(q)y(t) = B(q)u(t) + C(q)e(t)} \quad (9-2-3)$$

مدل معادله بالا را مدل ARMAX می نامند. برای این مدل معادله (2-5-2) بصورت زیر می شود:

²¹ Auto Regressive Moving Average With External Input

$$\hat{y}(t) = \frac{B(q)}{C(q)}u(t) + \left[1 - \frac{A(q)}{C(q)}\right]y(t) \quad (10-2-3)$$

برای شروع این تخمین لازم است پارامترهای زیر مشخص باشند:

$$y(0) \cdots y(-n^* + 1) \quad n^* = \max(n_c, n_a)$$

$$u(0) \cdots u(-n_b + 1)$$

و اگر این مقادیر مشخص نباشند، می توان همه را برابر صفر قرار داد. با قرار دادن خطای $e(t)$ در

معادله (10-5-2) می توان آنرا بصورت زیر بازنویسی کرد:

$$\hat{y}(t) = \varphi^T(t)\theta \quad (11-2-3)$$

که در آن $\varphi(t)$ شناخته شده و θ ناشناخته است و بصورت زیر است:

$$\varphi(t) = [-y(t-1) \quad \cdots \quad -y(t-n_a) \quad u(t-1) \quad \cdots \quad u(t-n_b) \quad e(t-1, \theta) \quad \cdots \quad e(t-n_c, \theta)]^T$$

$$\theta = [a_1 \quad \cdots \quad a_{n_a} \quad b_1 \quad \cdots \quad b_{n_b} \quad c_1 \quad \cdots \quad c_{n_c}]$$

۳- مدل ARARX :

در این مدل معادله خطا بصورت زیر بسط پیدا می کند: [۳۱،۳۴]

$$A(q)y(t) = B(q)u(t) + \frac{1}{D(q)}e(t) \quad (12-2-3)$$

$$D(q) = 1 + d_1q^{-1} + \cdots + d_{n_d}q^{-n_d} \quad \text{که در آن:}$$

۴- مدل ARMAX :

معادله خطای این مدل نیز بصورت زیر است: [۳۱،۳۴]

$$A(q)y(t) = B(q)u(t) + \frac{C(q)}{D(q)}e(t) \quad (13-2-3)$$

۵- مدل ARMA :

معادله خطای این مدل نیز بصورت زیر است: [۳۱،۳۴]

$$A(q).y(t) = C(q).e(t) \quad (14-2-3)$$

۶- ساختار خروجی خطا (OE)^{۲۲}:

معادله این ساختار بصورت زیر می باشد: [۳۱،۳۴]

$$y(t) = \frac{B(q)}{F(q)} u(t) + e(t) \quad (۱۵-۲-۳)$$

که در آن:

$$F(q) = 1 + f_1 q^{-1} + \dots + f_{n_f} q^{-n_f}$$

پارامترهایی که باید مشخص شوند:

$$\theta = [b_1 \quad \dots \quad b_{n_b} \quad f_1 \quad \dots \quad f_{n_f}]^T$$

اگر $\omega(t)$ را بصورت $\omega(t) = y(t) - e(t)$ تعریف کنیم آنگاه معادله (۱۵-۲-۳) بصورت باز شده چنین است:

$$\omega(t) + f_1 \omega(t-1) + \dots + f_{n_f} \omega(t-n_f) = b_1 u(t-1) + \dots + b_{n_b} u(t-n_b)$$

آنگاه معادله تخمین (۱۵-۲-۳) چنین می شود:

$$\hat{y}(t) = \varphi^T(t) \theta$$

$$\varphi(t) = [u(t-1) \quad \dots \quad u(t-n_b) \quad -\omega(t-1, \theta) \quad \dots \quad -\omega(t-n_f, \theta)]$$

۶- ساختار باکس - جنکینس (BJ²³):

این ساختار نیز راه دیگری برای گسترش مدل خروجی خطا می باشد.

$$y(t) = \frac{B(q)}{F(q)} u(t) + \frac{C(q)}{D(q)} e(t) \quad (۱۶-۲-۳)$$

حالتهای ذکر شده در بالا حدود ۳۲ حالت مختلف به ما می دهد. که البته همه یک حالت خاص از حالت عمومی زیر است:

$$A(q)y(t) = \frac{B(q)}{F(q)} u(t) + \frac{C(q)}{D(q)} e(t) \quad (۱۷-۲-۳)$$

²³ Box-Jenkins

۳-۳- مدل فضای حالت سیگنالهای صوتی:

در مورد سیگنالهای صوتی، ورودی که برای ما قابل مشاهده باشد نداریم و فقط صدایی که ممکن است همراه با نویز باشد می شنویم که به عنوان خروجی است. بنابراین مدل این سیستمها بصورت AR خواهد بود. [۳۴،۲۹]

اگر $s(n)$ سیگنال خروجی باشد که ما می شنویم، فرم AR آن چنین است:

$$A(q)s(n) = e(n) \quad (۱-۳-۳)$$

$$s(n) = a_1s(n-1) + a_2s(n-2) + \dots + a_p s(n-p) + e(n)$$

که هدف یافتن ماتریس θ با فرم زیر است:

$$\theta = [a_1 \quad a_2 \quad \dots \quad a_p] \quad (۲-۳-۳)$$

حال برای یافتن مدل فضای حالت این سیگنالها از معادله (۴-۵-۱) تبدیل Z گرفته شده است: [۴]

$$s(z) = a_1 z^{-1} s(z) + a_2 z^{-2} s(z) + \dots + a_p z^{-p} s(z) + e(z)$$

همچنین متغیرهای حالت بصورت زیر تعریف شده است:

$$\begin{array}{ll} x_1 = z^{1-p} s(z) & x_1(n+1) = x_2 \\ x_2 = z^{2-p} s(z) & x_2(n+1) = x_3 \\ \vdots & \vdots \\ x_p = s(z) & x_p(n+1) = e(n+1) + a_1 x_p + \dots + a_p x_1 \end{array} \Rightarrow$$

در اینصورت فضای حالت خواهد بود:

$$\begin{bmatrix} x_1(n+1) \\ x_2(n+1) \\ \vdots \\ x_{p-1}(n+1) \\ x_p(n+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ a_p & a_{p-1} & \dots & a_2 & a_1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{p-1} \\ x_p \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} e(n+1) \quad (۳-۳-۳)$$

$$s(n) = [0 \quad \dots \quad 1] \begin{bmatrix} x_1 \\ \vdots \\ x_p \end{bmatrix}$$

و با توجه به فرم کلی زیر خواهیم داشت:

$$x(n+1) = Ax(n) + Be(n) \quad (4-3-3)$$

$$s(n) = Cx(n)$$

- A یک ماتریس مربعی $P \times P$ می باشد که P درجه مدل AR سیستم مذکور می باشد
- ماتریسهای B و C ترانهاده یکدیگرند.

همانطور که از مدل فضای حالت مشخص است پارامترهای مجهول فقط a_i ها در ماتریس A هستند که از مدل AR قابل محاسبه می باشند. پس هدف تخمین ماتریس θ از رابطه (4-5-1) می باشد. برای یافتن پارامترهای مدل AR یا همان درایه های ماتریس θ می توان از روشهای مختلف حداقل مربعات تخمین استفاده نمود.

$$\theta = [a_1 \ a_2 \ \dots \ a_p]^T \quad (5-3-3)$$

$$\varphi = [-s(n-1) \ -s(n-2) \ \dots \ -s(n-p)]^T \quad (6-3-3)$$

$$\hat{s}(n) = \varphi^T \cdot \theta \quad (7-3-3)$$

4-3- اعتبار تخمین:

یکی از مسائل مهم در تخمینها، بحث میزان اعتبار تخمین است. در شناسایی سیستمها نیز علاوه بر روش شناسایی، معتبر بودن تخمین نیز بسیار مهم است.

در مسأله تخمین این سؤال وجود دارد که کدام مدل بهتر است و آیا این بهترین مدل از نظر ما به اندازه کافی خوب است و تا کجا اعتبار دارد و آیا هدف ما را از تخمین برآورده می سازد. بنابراین

اعتبار تخمین دارای جنبه های مختلف زیر است: [31]

- آیا مدل به میزان لازم برای اطلاعات مشاهده شده مناسب است؟
- آیا مدل به اندازه کافی برای هدف ما خوب است؟
- آیا مدل سیستم حقیقی را توضیح می دهد؟

بهترین راه برای رسیدن به این نتیجه که آیا مدل سیستم ما را بخوبی توصیف می کند بررسی راههای مختلف تست خطا می باشد. اگر خروجی مدل تخمین زده شده را $y_m(t)$ بنامیم که با

اطلاعات مشاهده شده ساخته ایم و $y(t)$ خروجی واقعی سیستم باشد، خطای تخمین بصورت زیر تعریف می شود:

$$\varepsilon(t, \theta_N) = y(t) - y_m(t) \quad (1-4-3)$$

برای یک مدلسازی خوب، باید مقدار رابطه (۲-۷-۱) مینیمم باشد. بنابراین راههای مختلفی برای تست کردن روی خطا وجود دارد که یکی از کاربردی ترین آنها روش آنالیز باقیمانده ها^{۲۴} است که مبنای تستهای مختلفی از جمله تست تابع خودهمبستگی^{۲۵}، تست تابع همبستگی متقابل^{۲۶}، تست تغییر علامت و غیره می باشد.

۳-۴-۱- تست تابع خود همبستگی:

این تست بر مبنای اینکه $\varepsilon(t)$ نویز سفید با میانگین صفر است استوار می باشد. بنابراین برای تابع همبستگی نویز سفید که بصورت زیر تعریف می شود خواهیم داشت :

$$\hat{r}_\varepsilon(\tau) = \frac{1}{N} \sum_{t=1}^{N-\tau} \varepsilon(t+\tau)\varepsilon(t) \quad (2-4-3)$$

$$\hat{r}_\varepsilon(\tau) = 0, \tau \neq 0$$

پس حالت کلی روابط بالا بصورت زیر است:

$$\hat{r}_\varepsilon(\tau) \rightarrow 0, \tau \neq 0 \quad (3-4-3)$$

$$\hat{r}_\varepsilon(0) = E[\varepsilon^2(t)] = \lambda^2$$

که می توان روابط (۲-۷-۳) را بصورت زیر نرمالیزه کرد: [۳۱]

$$x_\varepsilon = \frac{\hat{r}_\varepsilon(\tau)}{\hat{r}_\varepsilon(0)} \quad (4-4-3)$$

با توجه به روابط (۲-۷-۳) در می یابیم که X باید مقدار کوچکی باشد. برای مشخص شدن میزان کوچک بودن لازم است جزئیات بررسی شود. ماتریس I بصورت زیر تعریف می شود: [۳۱]

²⁴ Residuals

²⁵ Auto-Correlation

²⁶ Cross-Correlation

$$r = \frac{1}{N} \sum_{t=1}^N \begin{pmatrix} \varepsilon(t-1) \\ \vdots \\ \varepsilon(t-m) \end{pmatrix} \cdot \varepsilon(t) = \begin{pmatrix} \hat{r}_\varepsilon(1) \\ \vdots \\ \hat{r}_\varepsilon(m) \end{pmatrix} \quad (5-4-3)$$

که m درجه آزادی می باشد.

و اگر P ماتریس کوواریانس باشد:

$$P = \lim_{N \rightarrow \infty} E[Nrr^T] \quad (6-4-3)$$

$$P_{i,j} = \lambda^4 \delta_{i,j} \Rightarrow P = \lambda^4 I$$

در نتیجه داریم:

$$Nr^T P^{-1} r = Nr^T r / \hat{r}_\varepsilon^2(0) \quad (7-4-3)$$

برای استفاده از این تست باید ابتدا مقدار سمت راست معادله بالا محاسبه شود. X متغیر تصادفی است که از آن می توان مقدار a را یافت.

$$a = P(x > \chi_a^2(m))$$

برای a هایی که بین $0,1$ تا $0,01$ انتخاب می شوند داریم:

$$Nr^T r / \hat{r}_\varepsilon^2(0) \leq \chi_a^2(m) \quad \text{مدل پذیرفته شده است} \quad (8-4-3)$$

$$Nr^T r / \hat{r}_\varepsilon^2(0) > \chi_a^2(m) \quad \text{مدل پذیرفته شده نیست} \quad (9-4-3)$$

a در واقع درصد ریسک و خطاست و $\chi_a^2(m)$ تابع تصادفی با درجه آزادی m است که مقدار آن

برای a های مختلف در جدول مربوط به آن وجود دارد. [۳۱]

مقدار a معمولاً $0,05$ فرض می شود و جوابها با 95% اطمینان بدست می آیند.

۳-۴-۲- تست تغییر علامت:

این تست بر مبنای دو فرض استوار است:

▪ $\varepsilon(t)$ نویز سفید با میانگین صفر است.

▪ $\varepsilon(t)$ دارای توزیع متقارن است.

اگر \tilde{x}_N را تعداد تغییر علامت مؤلفه های خطا $(\varepsilon(1), \varepsilon(2), \dots, \varepsilon(N))$ بدانیم، با توجه به دو فرض بالا مقدار $\varepsilon(t)$ باید در هر بخش تغییر علامت دهد. یعنی $\tilde{x}_N \cong \frac{N}{2}$ باشد.

در اینصورت با احتمال ۹۵٪ اطمینان، \tilde{x}_N بایستی در رابطه زیر صدق کند: [۳۱، ۳۲]

$$\frac{N}{2} - \frac{1.96}{2} \sqrt{N} \leq \tilde{x}_N \leq \frac{N}{2} + \frac{1.96}{2} \sqrt{N} \quad (۱۰-۴-۲)$$

بنابراین می توان نتیجه گرفت هر سیگنالی که در رابطه بالا صدق کند، نویز سفید است. لازم بذکر است روشهای متفاوت زیادی جهت اعتبار تخمین وجود دارد که به علت عدم استفاده گسترده و طولانی شدن مطلب از ارائه آنها صرفنظر کرده و خواننده را به مرجع [۳۲] جهت اطلاعات بیشتر راهنمایی می کنیم.

۳-۵- شناسایی و مدل کردن سیستمهای غیر خطی:

در بحث شناسایی سیستمهای غیر خطی که صوت نیز از آن جمله بشمار می رود بدلیل متغیر بودن دینامیک های سیستم غیر خطی در هر لحظه از زمان و غیر قابل پیش بینی بودن آن بطور دقیق، لذا راههای قبل قابل ارائه و بحث نمی باشد. در نتیجه تنها راه مناسب جهت اینگونه مدلسازی ها کمک گرفتن از شبکه های عصبی است.

شبکه های عصبی و عصبی-فازی بدلیل قابلیت یادگیری تنها ابزارهای قابل استفاده در شناسایی و پیش بینی سیستمهای غیر خطی می باشند. این شبکه ها می توانند با موازی قرار گرفتن با سیستم مورد مدلسازی و یادگیری هوشمند رفتار سیستم پس از مدتی دقیقاً رفتاری مشابه سیستم از خود نشان دهند تا جایی که می توان آنها را جایگزین سیستم مدل شده نمود. این توانایی فوق العاده به ما امکان می دهد تا با بهره گیری از شبکه های عصبی بسیاری از دینامیک های غیر خطی و پیچیده سیستمهای واقعی را با دقت بالا مدل کنیم.

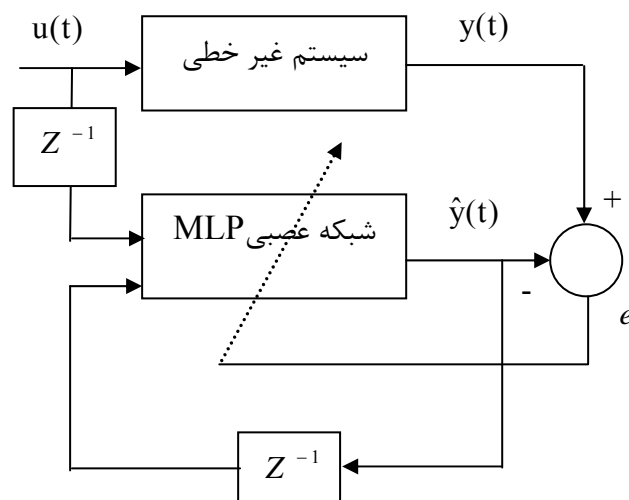
در ادامه ۲ روش بسیار مهم و پایه ای شناسایی سیستمهای غیر خطی که مشابه آنها در مورد سیستمهای خطی نیز توضیح داده شده ارائه می گردد.

۳-۵-۱- روش NARX :

همانطور که از بخش قبل می دانیم روش ARX روشی مبتنی بر شناسایی سیستم با مشخص بودن ورودی و خروجی آن بود. این امر در مورد سیستمهای غیر خطی نیز صادق است. بدین ترتیب که اگر معادله زیر معرف یک مدل ARX غیر خطی باشد:

$$y(t) = f(y(t-1), y(t-2), \dots, y(t-n_y), u(t-1), u(t-2), \dots, u(t-n_u)) \quad (۱-۵-۳)$$

که در آن خروجی سیستم تابع تمامی خروجیها و ورودیهای قبلی است، لذا پیاده سازی آن با شبکه عصبی بصورت بلوک دیاگرام زیر می باشد: [۳۳]

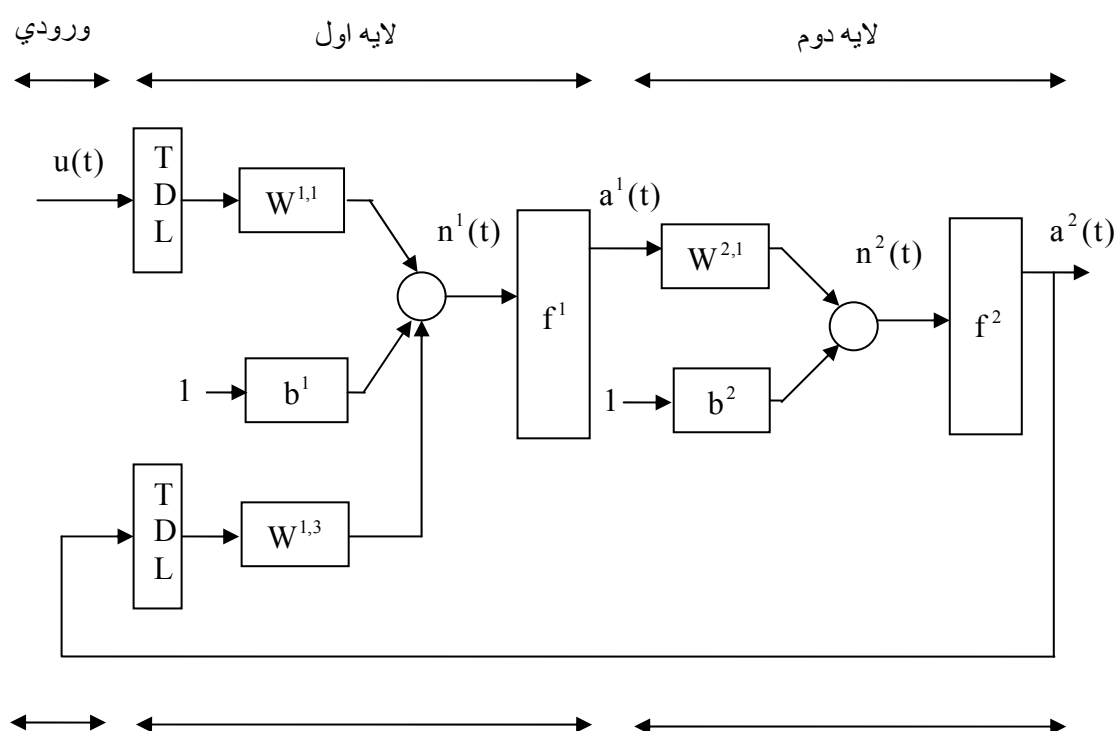


شکل ۳-۵-۱- نمایش مدلسازی سیستم غیر خطی با روش ARX غیر خطی

روش شناسایی سیستم نیز بدین صورت است که سیستم غیر خطی مورد نظر با شبکه عصبی که معمولاً از نوع MLP انتخاب می شود موازی می گردد. سپس ورودی سیستم به شبکه عصبی اعمال و خروجی شبکه عصبی پس از محاسبه به ورودی بازخورد داده می شود. همچنین خروجی محاسبه شده توسط شبکه با خروجی واقعی سیستم مقایسه و خطای بدست آمده جهت تنظیم

پارامترهای شبکه به شبکه برگردانده می شود. لذا پس از همگرا شدن خطای یادگیری شبکه به مقدار مورد نظر می توان مطمئن بود که شبکه بدرستی آموزش دیده است و می تواند جایگزین سیستم واقعی غیر خطی گردد همچنین جهت تست شبکه عصبی می توان پس از آموزش کامل شبکه، ورودیهای یکسان به سیستم مدل شده و شبکه اعمال نمود. در صورتیکه هر دوی سیستم و شبکه در خروجی دارای پاسخ یکسانی بودند می توان فهمید که از آن پس شبکه می تواند بعنوان یک مدل خوب و دقیق جایگزین سیستم غیر خطی مربوطه گردد.

ذکر این نکته نیز قابل اهمیت است که خطای یادگیری شبکه و پاسخ شبکه به ورودیهای تست پس از آموزش می توانند معیارهای خوبی جهت اعتبار مدلسازی ما باشند. حال به نمایش بلوک دیاگرام شبکه عصبی مورد استفاده جهت شناسایی سیستم غیر خطی می پردازیم. نمایش آن بصورت زیر است:



شکل ۳-۵-۲- نمایش شبکه عصبی بکار رفته در روش ARX غیر خطی

همانطور که از نمایش بلوک دیاگرام شبکه عصبی بالا مشخص است شبکه عصبی دارای یک ورودی $u(t)$ است که پس از عبور از یک بلوک تاخیر به لایه اول شبکه اعمال شده است. خروجی لایه اول که همان $a^1(t)$ می باشد از رابطه زیر محاسبه می گردد:

$$\begin{aligned} n^1(t) &= W^{1,1}.u(t-1) + W^{1,3}.a^2(t-1) + b^1 \\ a^1(t) &= f^1(n^1(t)) \end{aligned} \quad (2-5-3)$$

که تابع آتش لایه اول شبکه که همان f^1 می باشد می تواند یکی از توابع غیرخطی تانژانت هایپربولیک، سیگموئید و توابعی از این دست و یا توابع خطی باشد. لذا با فرض اینکه تابع f^1 بصورت تانژانت هایپربولیک انتخاب شود خواهیم داشت:

$$a^1(t) = \tanh(W^{1,1}.u(t-1) + W^{1,3}.a^2(t-1) + b^1) \quad (3-5-3)$$

سپس خروجی لایه اول که همان $a^1(t)$ می باشد به عنوان ورودی لایه دوم به این لایه اعمال می شود. حال خروجی نرون لایه دوم که $n^2(t)$ می باشد پس از عبور از تابع f^2 بعنوان خروجی لایه دوم که همان $a^2(t)$ می باشد ظاهر می گردد. اگر تابع f^2 بصورت سیگموئید انتخاب شود لذا رابطه خروجی شبکه بصورت زیر خواهد بود:

$$a^2(t) = \text{sigmoid}(W^{2,1}.a^1(t) + b^2) \quad (4-5-3)$$

که با ترکیب رابطه (3-5-3) و (4-5-3) به رابطه کلی محاسبه خروجی شبکه عصبی بصورت زیر خواهد بود:

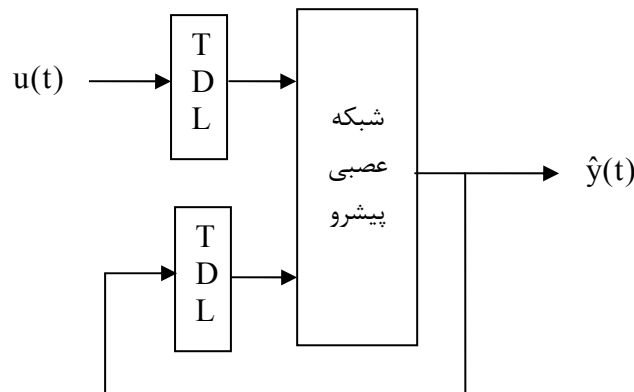
$$a^2(t) = \text{sigmoid}(W^{2,1}.\tanh(W^{1,1}.u(t-1) + W^{1,3}.a^2(t-1) + b^1) + b^2) \quad (5-5-3)$$

که می توان با استفاده از روشهای مختلف آموزش شبکه های عصبی مثل پس انتشار خطا شبکه را آموزش داد و در نهایت بجای مدلی که فقط از ورودی و خروجیهای آن مطلع هستیم از این شبکه استفاده نمود.

همچنین در این بخش به ۲ ساختار پیکربندی شبکه اشاره و مزایا و معایب آنها نسبت به یکدیگر بررسی شده است. این ساختارها بصورت زیر می باشند:

- ساختار موازی:

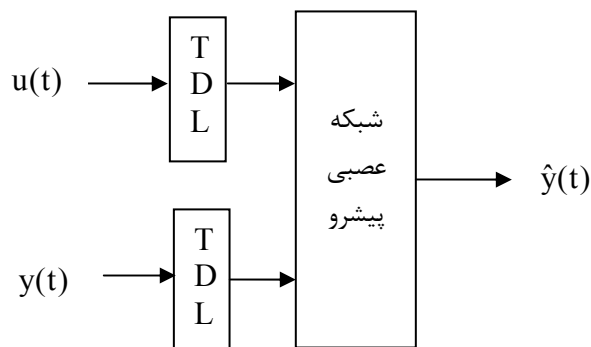
در این ساختار که همان ساختار شبکه عصبی قبل است که معرفی گردید، ۲ ورودی به نرون لایه اول شبکه اعمال می گردد. یکی ورودی تاخیر دار سیستم غیر خطی می باشد و دیگری خروجی تاخیر دار محاسبه شده توسط شبکه عصبی است که با یک فیدبک به ورودی شبکه اعمال شده است. نمایش آن نیز بصورت زیر است:



شکل ۳-۵-۳- نمایش ساختار موازی شبکه عصبی پیشرو

- ساختار سریال - موازی:

در این ساختار که با ساختار قبل تفاوت دارد نیز، ۲ ورودی به نرون لایه اول شبکه اعمال می گردد. اما این بار یکی ورودی تاخیر دار سیستم غیر خطی می باشد و دیگری خروجی تاخیر دار خود سیستم غیر خطی است نه خروجی محاسبه شده توسط شبکه عصبی که نمایش آن نیز بصورت زیر است:



شکل ۳-۵-۴- نمایش ساختار موازی شبکه عصبی پیشرو

آموزش شبکه عصبی با این ساختار دارای ۲ مزیت نسبت به ساختار قبل است:

۱- ورودی اعمال شده به شبکه عصبی دارای مقدار دقیق تری است لذا با توجه به ماهیت روش پس انتشار خطا در آموزش شبکه های $MLP^{۲۷}$ که خطای یادگیری را بصورت بازگشتی کاهش می دهند، شبکه با سرعت بیشتر و خطای کمتری (در تعداد تکرار مشخص) همگرا خواهد شد.

۲- شبکه دارای ساختار پیشروی دقیق تری است لذا روش پس انتشار خطای استاتیک می تواند جهت شبکه بکار گیری شود.

در نتیجه پیشنهاد مقالات و محققان در استفاده از ساختار دوم در آموزش شبکه های عصبی جهت شناسایی سیستمهای غیر خطی با روش ARX می باشد.

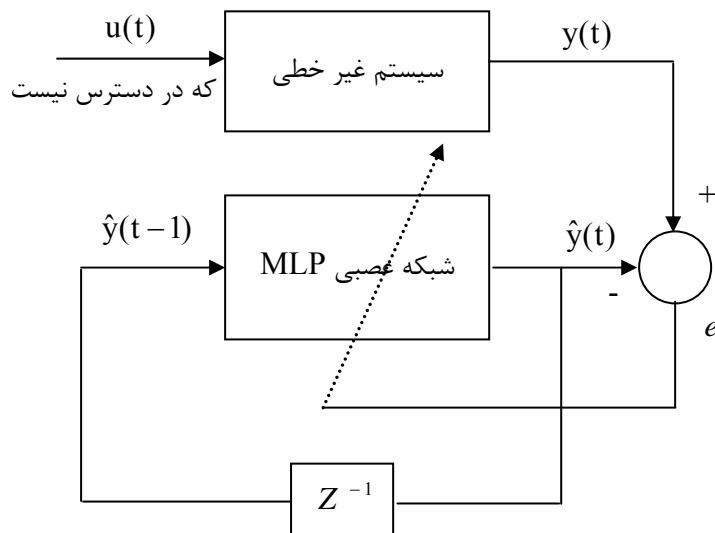
۳-۵-۲- روش NAR :

این روش نیز مبتنی بر اصول و قواعد روش AR است که جهت سیستمهای غیر خطی معرفی شده است. تنها تفاوت این روش با روش $NARX$ در این است که این روش در مواقعی بکار گیری می شود که اطلاعاتی از ورودی سیستم غیر خطی در دسترس نباشد. به عبارت دیگر تنها اطلاعات مشخص از سیستم غیر خطی که می خواهیم آنرا شناسایی کنیم خروجی سیستم می باشد. در نتیجه رابطه معرف این روش بصورت زیر است:

$$y(t) = f(y(t-1), y(t-2), \dots, y(t-n_y)) \quad (۳-۵-۶)$$

و بلوک دیاگرام شناسایی سیستم با استفاده از این روش و تلفیق آن با شبکه عصبی بصورت زیر است:

^{۲۷} Multi Layer Perceptron

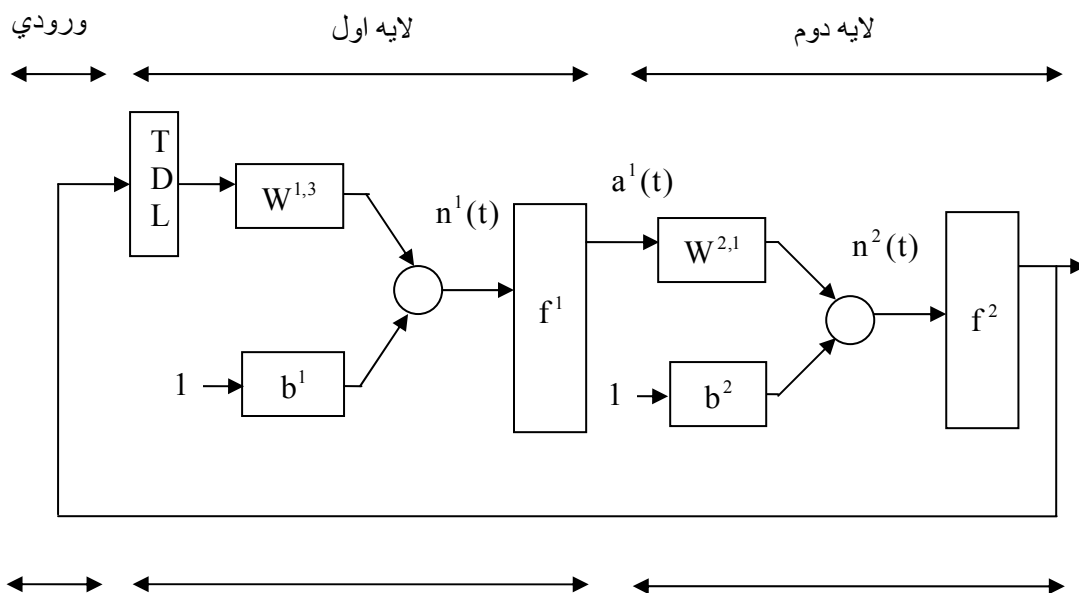


شکل ۳-۵-۵- نمایش مدل‌سازی سیستم غیر خطی با روش AR غیر خطی

که روش بکارگیری شبکه عصبی مانند بخش قبل است با این تفاوت که ورودی شبکه عصبی فقط خروجیهای تاخیر دار محاسبه شده توسط شبکه عصبی و بازخورد داده شده به ورودی شبکه است. در بقیه موارد روش همانند قبل است و خروجی محاسبه شده توسط شبکه عصبی با خروجی سیستم که همان سیگنال هدف شبکه عصبی است مقایسه شده و خطای آن جهت تنظیم پارامترهای شبکه به شبکه اعمال خواهد شد.

ذکر این نکته حائز اهمیت است که در اکثر مواقع ورودی سیستمهای خطی یا غیر خطی قابل دسترس نیست لذا در بیشتر کاربردها از روش AR یا NAR استفاده می شود. لذا در این پایان نامه نیز فرض بر نامعلوم بودن ورودی سیستم ساخت سیگنال صوتی شده است لذا از روش NAR جهت شناسایی دینامیکهای سیستم صوتی استفاده شده است که جزئیات آن در فصل چهارم و بخشی از آن در انتهای این فصل خواهد آمد.

همچنین نمایش بلوک دیاگرام شبکه عصبی بکار رفته با استفاده از این روش نیز بصورت زیر است:



شکل ۳-۵-۶- نمایش شبکه عصبی بکار رفته در روش AR غیر خطی

همچنین نحوه محاسبه خروجی لایه اول با تغییر ناچیزی نسبت به روش قبل بصورت زیر است:

$$n^1(t) = W^{1,3} \cdot a^2(t-1) + b^1$$

$$a^1(t) = \tanh(W^{1,3} \cdot a^2(t-1) + b^1) \quad (7-5-3)$$

و در مورد خروجی لایه دوم خواهیم داشت:

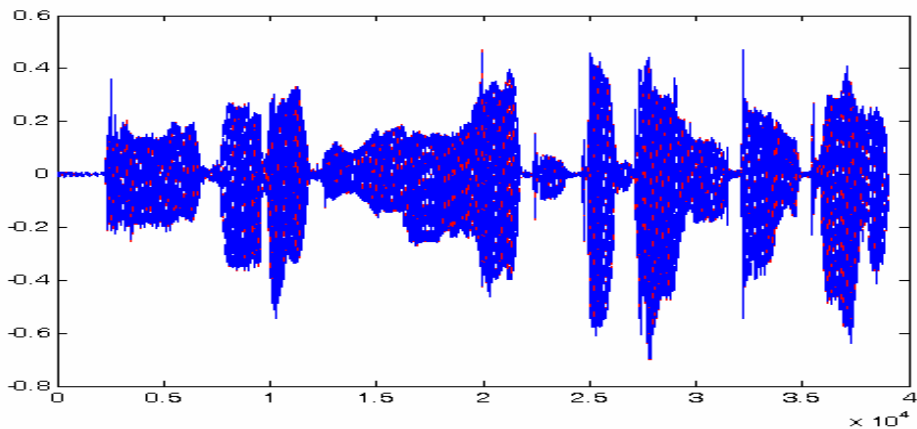
$$a^2(t) = \text{sigmoid}(W^{2,1} \cdot a^1(t) + b^2)$$

$$a^2(t) = \text{sigmoid}(W^{2,1} \cdot \tanh(W^{1,3} \cdot a^2(t-1) + b^1) + b^2) \quad (8-5-3)$$

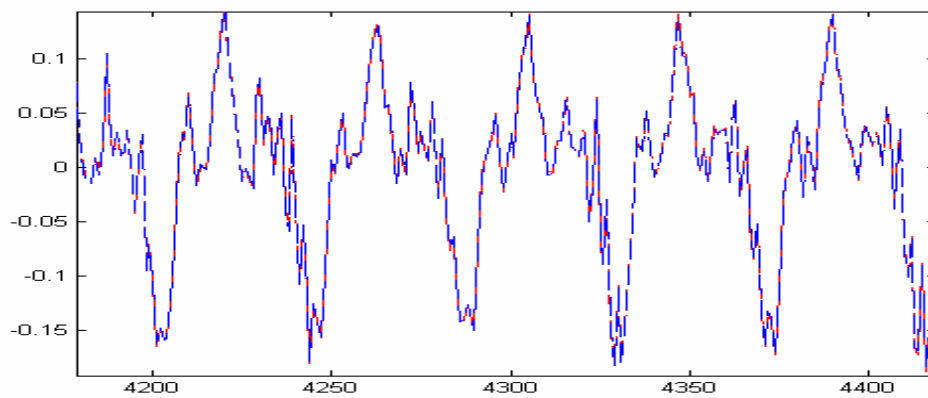
۳-۵-۳- نمایش شبیه سازیهای مدل سازی صوت :

در پایان این فصل به نمایش شبیه سازیهای شناسایی غیر خطی صوت که با روش NAR انجام شده پرداخته شده است. ذکر این نکته لازم است که این فصل به ارائه نمایش شبیه سازیها اکتفا کرده و در فصل چهار بطور مفصل در مورد نحوه پیاده سازی آنها بحث خواهد شد. با مقادیر اولیه زیر شبکه عصبی MLP ۱-۵-۵ با روش EBP با تکرار ۳۹۰۰۰ بار که طول بردار صوت است تحت آموزش قرار گرفت که نتایج آن بصورت زیر است:

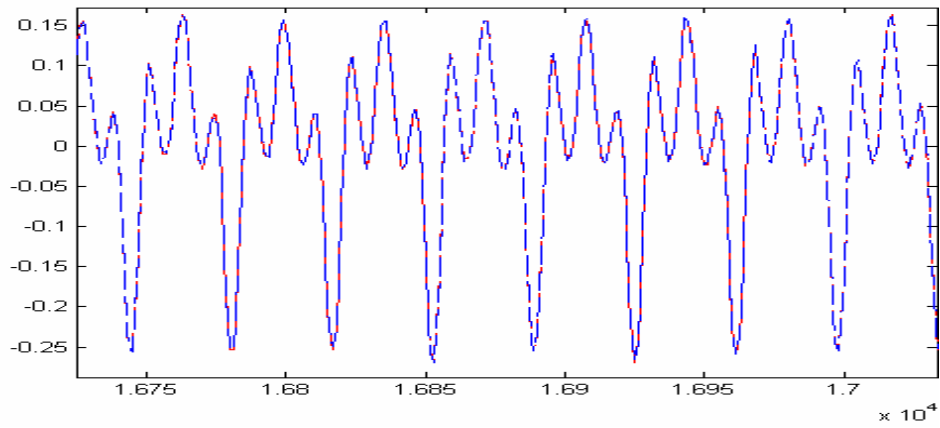
$W1, W2 = \text{randn}(x)$	$A0 = \text{randn}(x)$
$B1, B2 = \text{randn}(x)$	$\alpha = 0.05$
$dW1, dW2 = \text{zeros}(x)$	$\eta = 0.01$
$dB1, dB2 = \text{zeros}(x)$	$\text{iteration} = 1 * 39000$



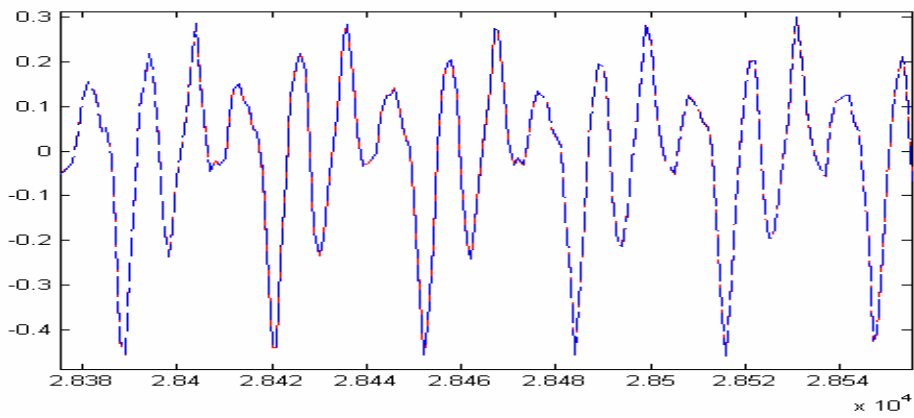
شکل ۳-۵-۷- مقایسه خروجی شبکه (نمودار آبی) با Target (نمودار قرمز)



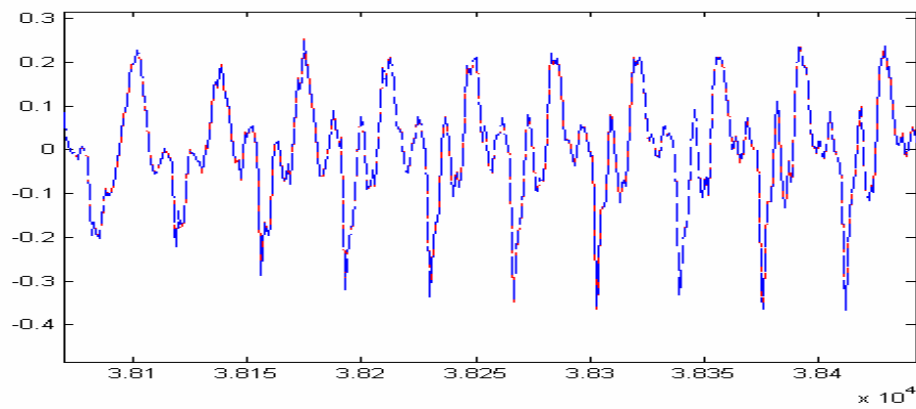
شکل ۳-۵-۸- مقایسه خروجی شبکه (نمودار آبی) با Target (نمودار قرمز) در ۵۰۰۰ تکرار اول



شکل ۳-۵-۹- مقایسه خروجی شبکه (نمودار آبی) با Target (نمودار قرمز) در ۵۰۰۰ تکرار چهارم



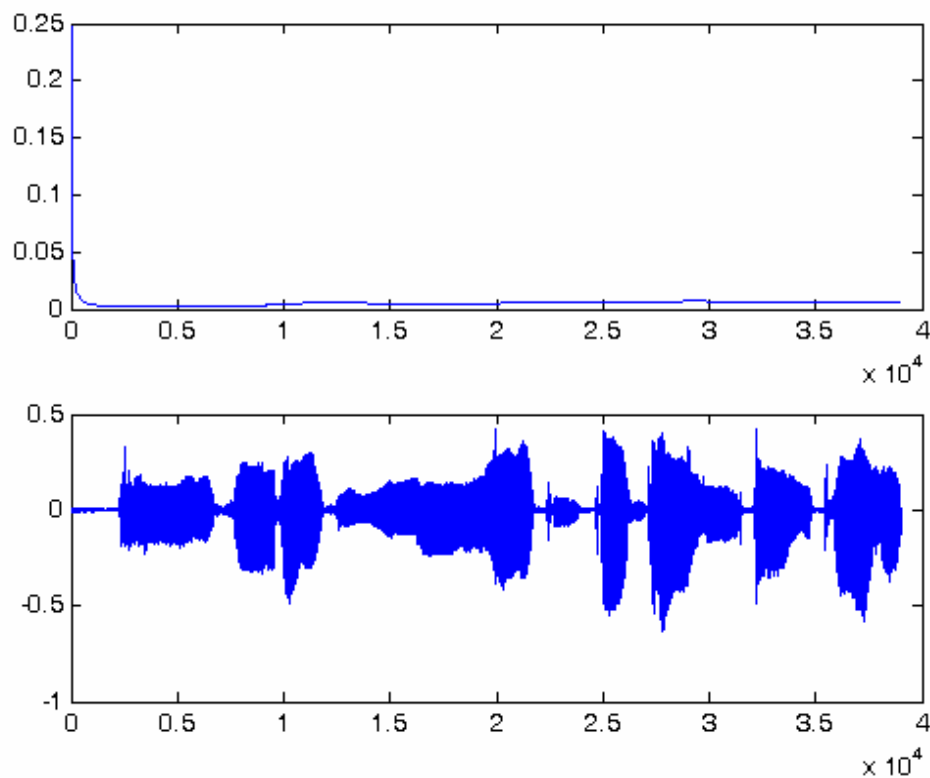
شکل ۳-۵-۱۰- مقایسه خروجی شبکه (نمودار آبی) با Target (نمودار قرمز) در ۵۰۰۰ تکرار ششم



شکل ۳-۵-۱۱- مقایسه خروجی شبکه (نمودار آبی) با Target (نمودار قرمز) در ۵۰۰۰ تکرار آخر

همانطور که در شکل ۶ بالا مشخص است خروجی شبکه عصبی با خروجی در دسترس سیستم غیر خطی که به عنوان سیگنال هدف شبکه عصبی در نظر گرفته شده است بطور تقریبی برابر است. لذا می توان فهمید که سیستم غیر خطی با دقت بسیار خوبی توسط شبکه عصبی MLP مدل شده است.

همچنین در ذیل خطای یادگیری شبکه عصبی و نحوه همگرایی آن طی ۳۹۰۰۰ تکرار نمایش داده شده است. همانگونه که مشخص است این خطا با دقت بسیار خوبی به سمت صفر همگرا شده است که خود دلیل دیگری بر دقت مناسب مدل سازی سیستم غیر خطی می باشد.



شکل ۳-۵-۱۲- نمایش خطای آموزش شبکه به همراه خروجی شبکه

در پایان نیز ذکر این نکته لازم است که سیستم غیر خطی مورد شناسایی در این فصل یک صوت بوده است که جزئیات و مشخصات آن در فصل چهارم توضیح داده خواهد شد. همچنین برنامه نوشته شده با نرم افزار MATLAB جهت این مدل سازی نیز در پیوست پایان نامه آمده است.

۴ بررسی شبیه سازیهای انجام شده و نتایج آن

۴-۱- مقدمه:

با توجه به معرفی روابط فیلتر کالمن توسعه یافته در فصل دوم و معرفی روشهای شناسایی سیستمهای غیر خطی در فصل سوم، در این فصل به ارائه شبیه سازیهای بدست آمده از پیاده سازی الگوریتم شناسایی و تخمین صوت پرداخته خواهد شد. همچنین نتایج شبیه سازیهای انجام شده با نرم افزار Matlab ارائه و در مورد هر یک و مقایسه آنها بحث خواهد شد و در نهایت به نحوه رسیدن به نتیجه بهینه جهت بازیابی صوت نویزی شده می پردازد .

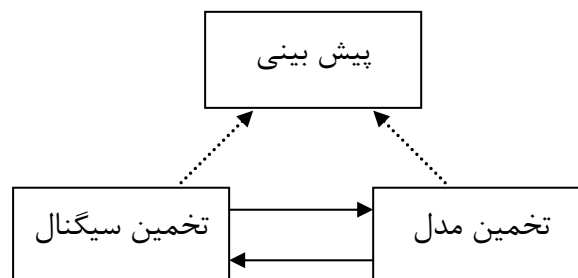
همچنین یک نتیجه گیری از شبیه سازیهای انجام شده با شبکه عصبی جهت شناسایی صوت غیر خطی و همچنین برنامه فیلتر کالمن توسعه یافته ارائه خواهد شد و در پایان پیشنهاد جهت ادامه این پایان نامه در آینده ارائه خواهد شد.

۴-۲- شرح خلاصه مطالب :

موضوع این پایان نامه مدل کردن و تخمین سیگنالهای نویزی زمان گسسته یا همان دنباله های زمانی است که در بهبود سیگنالهای صوتی کاربرد دارد. کاربردهای فراوانی از بهبود گفتارهای صوتی نیازمند تخمین و مدلسازی دنباله های زمانی نویزی می باشند

مدل سازی که در فرآیند تخمین سیگنالهای نویزی اولین بخش لازم جهت تخمین می باشد (از آن بعنوان شناسایی سیستم نیز یاد می شود) در واقع فرآیند تخمین دینامیکهای اساسی سیستم است . همچنین تخمین که بخش بعد از مدلسازی محسوب می شود از دینامیکهای سیستم که در بخش مدلسازی بدست آمده اند جهت بدست آوردن سریهای زمانی تمیز استفاده می کند. همچنین در بخش تخمین از همه اطلاعات سیستم تا زمان جاری جهت تخمین مقدار فعلی سری زمانی تمیز استفاده می شود.

لازم بذکر است ۲ بخش تخمین و مدلسازی شدیداً به یکدیگر وابسته هستند. به عنوان مثال یک مدل دقیق از سیستم که از سریهای زمانی تهیه شده است می تواند برای تخمین سیگنال تمیز بکار رود و بر عکس اگر سیگنال تمیز در دسترس باشد می تواند جهت ساخت یک مدل دقیق از دینامیکهای سیستم بکارگیری شود. علاوه بر این اگر یک مدل دقیق و تخمین های مناسب از سیگنال در دسترس باشند ، یک پیش بینی خوب می تواند با بکار گیری تخمین ها بعنوان ورودیهای مدل بدست آید.



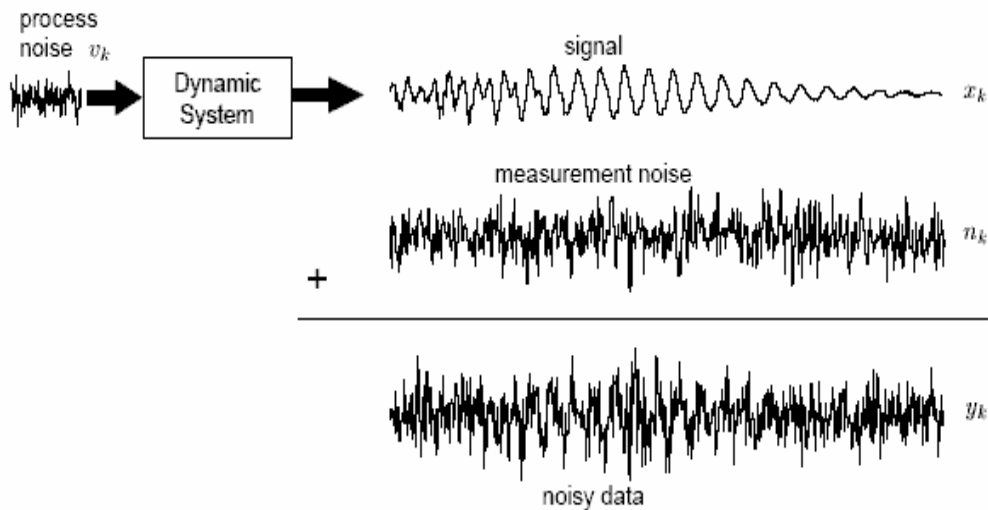
شکل ۴-۱- تخمین سیگنال و مدل دو امر وابسته به یکدیگر است و پیش بینی به هر دوی آنها نیازمند است

۴-۳- ساختار مدل :

یک سری نویزی را فرض کنید که با تابع AR غیر خطی و نویز مشاهده اضافه شده تهیه شده است:

$$\begin{aligned} x_k &= f(x_{k-1}, \dots, x_{k-M}, w) + v_k \\ y_k &= x_k + n_k \quad \forall k \in \{1 \dots N\} \end{aligned} \quad (1-1)$$

که x_k مربوط به سری زمانی صحیح است که با نویز فرآیند v_k جمع شده است و $f(\cdot)$ یک تابع غیر خطی است (مثل شبکه عصبی) از M مقادیر گذشته x_k که با w پارامتری شده است و تنها مشاهده در دسترس y_k است که شامل نویز اضافه شده n_k می باشد. توجه به این نکته نیز حائز اهمیت است که دنباله های زمانی یک بعدی هستند یعنی مشاهده نویزی y_k یک اسکالر است. وضعیت مذکور در شکل ۱-۲ نمایش داده شده است. نشان $\{y_k\}_t^1$ بکار رفته تا دنباله اطلاعات $\{y_1, y_2, \dots, y_t\}$ را نمایش دهد.



شکل ۴-۲- در این نمایش فرض شده که اطلاعات با یک مدل AR غیر خطی نامعلوم تهیه شده و با نویز مشاهده افزودنی تخریب شده است.

در یک فرمول بندی عمومی دیگر ممکن است تاثیر غیر خطی بودن کانال نیز بصورت $y_k = g(x_k, \dots, x_{k-M+1}, n_k)$ نمایش داده شود، لذا ساختاری که در این پایان نامه ارائه شده است

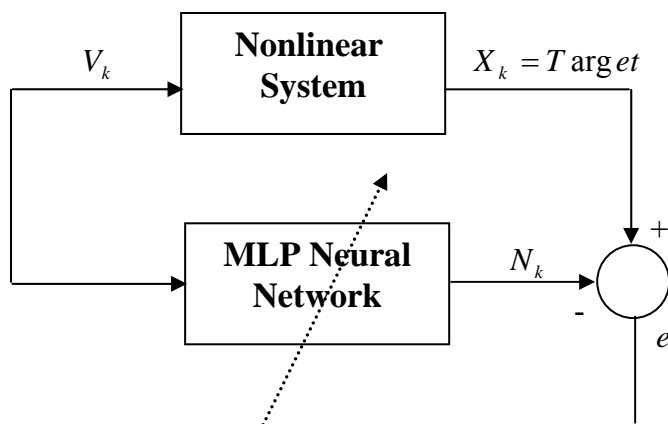
تا زمانیکه تابع کانال $g(.)$ معلوم و تشخیص پذیر باشد، می تواند بسادگی جهت در بر گرفتن چنین معادلات اندازه گیری غیر خطی تنظیم شود. در معادله (۱-۱) این تابع به فرم خاص $x_k + n_k$ نشان داده شده است.

همچنین الگوریتمهایی که در این پایان نامه استفاده شده است با فرض گوسی بودن نویز مشاهده و پروسه و همچنین در دسترس بودن آنها تنظیم و ارائه شده است که در صورت عدم اطلاع از واریانس نویزهای پروسه و مشاهده نیازمند تخمین دوگانه هستیم که در فصل دوم اشاره ای به اطلاعات تئوری آن شد و از نوشتن برنامه و شبیه سازی آن بدلیل طولانی بودن صرفنظر شده است که در کارهای بعدی می تواند در دستور کار دانشجویان محترم قرار گیرد.

۴-۳-۱- حلقه شناسایی سیستم :

روش ARX غیر خطی :

در این روش شناسایی سیستم که جزئیات آن در فصل قبل بیان شد فرض بر این است که ورودی و خروجی سیستم معلوم و شناخته شده است. در نتیجه ورودی سیستم به شبکه عصبی اعمال و خروجی سیستم به عنوان سیگنال هدف شبکه در نظر گرفته می شود و سپس شبکه شروع به دیدن آموزش می نماید که نمایش بلوکی این روش بصورت زیر است:

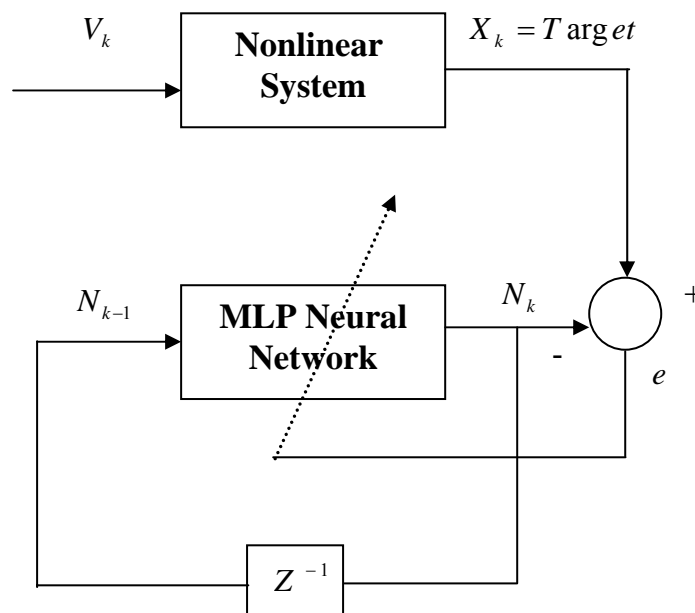


شکل ۴-۳- حلقه شناسایی سیستم : پس از N بار تکرار و همگرا شدن خطا، $N_k = X_k$ خواهد شد

روش AR غیر خطی :

در این روش شناسایی سیستم نیز که در فصل سوم به تشریح آن و روابط خاصش پرداخته شد فرض بر این است که فقط خروجی سیستم معلوم و شناخته شده است . در نتیجه ورودی سیستم برای ما ناشناس است و ما فقط به خروجی سیستم دسترسی داریم بنابراین خروجی سیستم به عنوان Target شبکه در نظر گرفته می شود و فیدبکهایی از خروجی بعنوان ورودی سیستم در نظر گرفته می شوند و سپس شبکه شروع به دیدن آموزش می نماید که

نمایش بلوکی این روش بصورت زیر است :



شکل ۴-۴- حلقه شناسایی سیستم : پس از N بار تکرار و همگرا شدن خطا ، $N_k = X_k$ خواهد شد

در واقع منطق بکار رفته در مورد شناسایی سیستم غیر خطی بدین صورت است که با توجه به توضیحات قبل، فرض بر این است که ورودی سیستم غیر خطی که دینامیکهای آن برای ما ناشناس است در حالت ARX نویز پروسه V_k و در حالت AR دینامیکهای خروجی شبکه می باشد و خروجی سیستم غیر خطی سیگنال تمیز X_k می باشد

در نتیجه ما برای شناسایی سیستم یک شبکه عصبی MLP را با سیستم موازی کرده و ورودی سیستم را بعنوان ورودی به شبکه اعمال کرده و خروجی سیستم به عنوان سیگنال

هدف^{۲۸} برای شبکه تعریف شده است. سپس با استفاده از روش پس انتشار خطای بازگشتی دسته ای^{۲۹} اقدام به آموزش شبکه عصبی شده است که در هر بار تکرار خروجی شبکه با مقدار Target مقایسه شده و خطا بعنوان آیتم اصلاحی به شبکه عصبی اعمال شده است. بدلیل اینکه طول بردار X_k که یک سیگنال صوتی می باشد ۳۹۰۰۰ بوده است لذا نیاز به حداقل تکرار ۳۹۰۰۰ برای شبکه عصبی مطرح شده است.

۴-۳-۲- ساختار شبکه عصبی:

در این بخش به معرفی ساختار شبکه و چگونگی انتخاب این ساختار پرداخته می شود. ساختاری که جهت شبکه عصبی در این پایان نامه در نظر گرفته شده ۱-۵-۵ است که دلایل این انتخاب به شرح ذیل می باشد:

۱- بدلیل اینکه در تشکیل ماتریس A که در برنامه فیلتر کالمن توسعه یافته استفاده می شود

و بعد به آن اشاره خواهد شد نیاز به بدست آوردن $\frac{\partial f(x_k, w)}{\partial x_k}$ بود که در واقع $f(x_k, w)$

همان خروجی شبکه عصبی ما بود. ما نیاز به بدست آوردن یک ماتریس A حداقل 2×2

داشتیم که بعد ماتریس A را تعداد ورودیهای شبکه تشکیل می داد و در نتیجه ما ۵

ورودی برای شبکه انتخاب نمودیم و بعد ماتریس A نیز 5×5 قرار گرفت.

۲- نحوه ارسال ورودیها به شبکه نیز بدین صورت می باشد که ابتدا یک بردار ۵ سطری به

شبکه اعمال می شود و سپس خروجی شبکه پس از محاسبه بعنوان متغیر ۶ به بردار

ورودی اولیه اضافه می شود و این بار برداری جدید که شامل متغیر دوم تا ششم می باشد

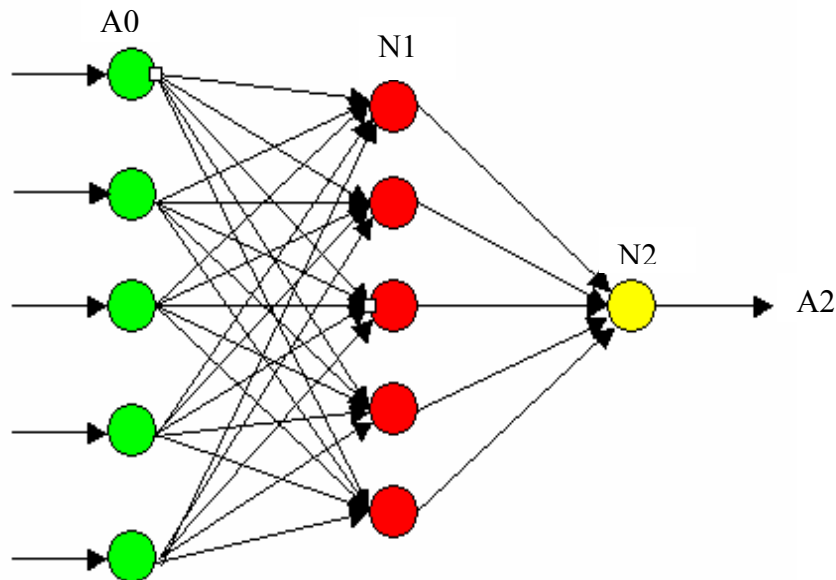
به شبکه اعمال می شود و این حلقه تا انتهای همگرا شدن خطای شبکه و آموزش دیدن

کامل آن ادامه می یابد.

²⁸ Target

²⁹ Batch Error Back Propagation

بنابراین شماییک شبکه عصبی $f(.)$ بعنوان شبکه feedforward با ۵ ورودی ($M = 5$) ، ۵ لایه مخفی و یک خروجی که بعنوان مدل غیرخطی درجه ۵ ($M = 5$) تعریف شود بصورت زیر است :



شکل ۴-۵- شبکه عصبی MLP با ۵ ورودی-۵ لایه مخفی و ۱ خروجی

که در ادامه نحوه آموزش شبکه عصبی با ساختار فوق معرفی می گردد.

۴-۳-۳- نحوه آموزش شبکه عصبی :

در این بخش، پایان نامه به نحوه آموزش شبکه عصبی جهت شناسایی سیستم غیر خطی می پردازد.

نحوه آموزش بدین صورت است که توسط نرم افزار Matlab دو برنامه جداگانه که یکی از آنها تابع خاص آموزش شبکه MLP به روش EBP^{۳۰} می باشد نوشته شده و بکارگیری شده است. ابتدا در برنامه اول با نام Neural network system Identification بردار صوت تمیز توسط دستور نرم افزاری Wavread به بردار تبدیل می شود و به عنوان هدف جهت آموزش شبکه عصبی در نظر گرفته می شود. سپس بردارهای وزن شبکه و بایاس های مربوطه با

³⁰ Error Back Propagation

ابعاد متناسب با فرمول بندی شبکه MLP بکار گیری شده که ۱-۵-۵ می باشد در برنامه مقدار دهی اولیه می شوند و در نهایت تابع آموزش شبکه که با نام Batchbp2 در برنامه دوم که Train Network with EBP Method نام دارد تعریف شده است، فرخوانده می شود. در این تابع نیز که از روش پس انتشار خطا جهت آموزش شبکه عصبی استفاده شده است ابتدا مقادیر اولیه بردارهای دلتای وزنها و بایاس های شبکه مقدار دهی اولیه می شوند تا در صورت استفاده از فاکتور فراموشی در برنامه مشکلی در محاسبه خروجی لایه ها بوجود نیاید.

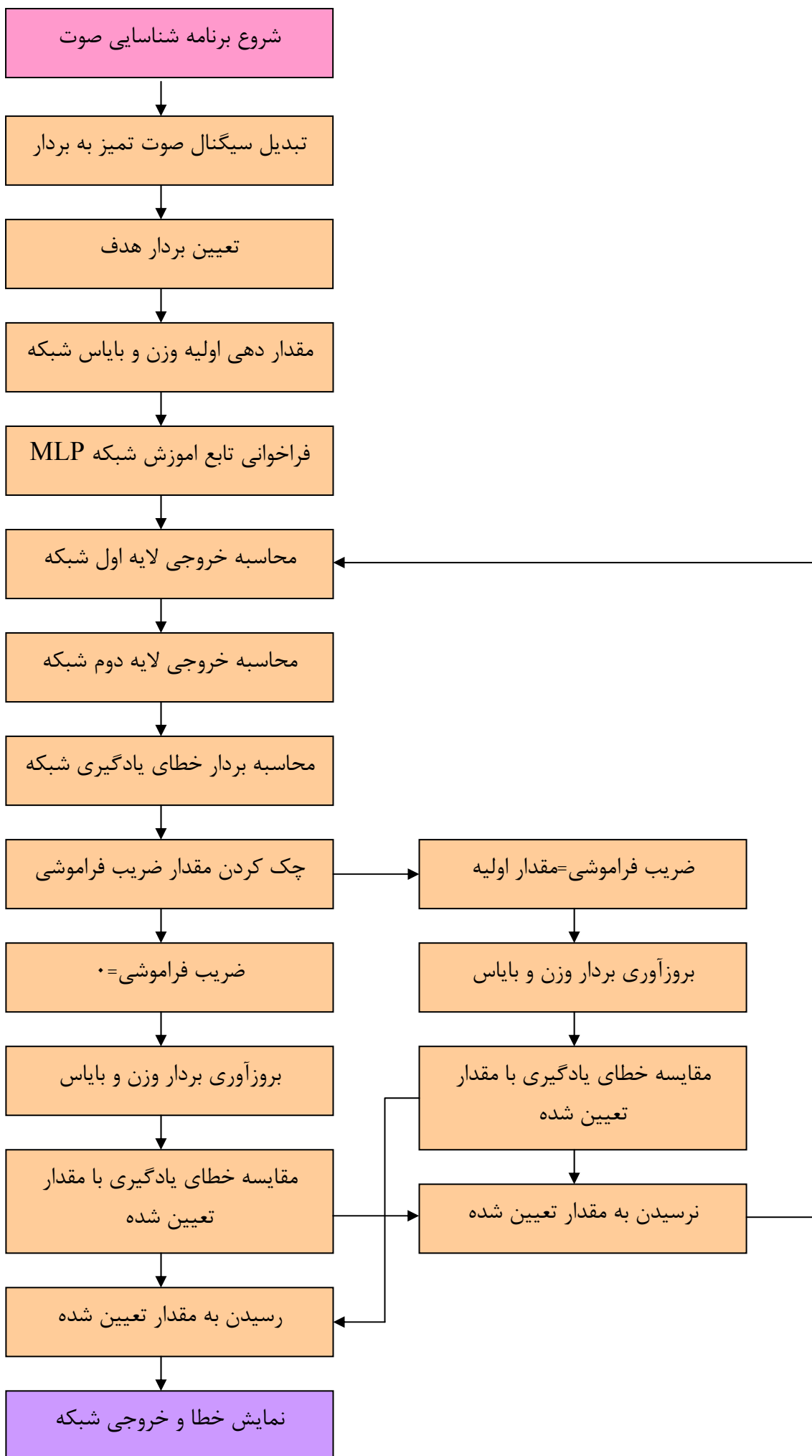
در ادامه خروجی لایه اول و سپس خروجی لایه دوم که همان خروجی شبکه می باشد محاسبه می شود. سپس بردار خطای شبکه که حاصل تفاضل بردار خروجی محاسبه شده توسط شبکه و بردار هدف می باشد محاسبه می گردد.

خطای محاسبه شده، در معادلات بروز آوری بردارهای وزن و بایاس شبکه بکارگیری می شود و پس از بروز شدن بردارهای مذکور خطای محاسبه شده با مقدار مورد انتظار مقایسه می شود که در صورت عدم رسیدن خطا به حد مورد نظر چرخه مذکور دوباره از محاسبه مقدار خروجی لایه اول شبکه تکرار می گردد. اما این بار ورودی شبکه، خروجی محاسبه شده شبکه در مرحله قبل می باشد که بجای بردار اولیه ورودی به شبکه اعمال می گردد.

همچنین فاکتور فراموشی برای شبکه بصورت انتخابی تعریف شده است که در صورت بکارگیری آن توسط کاربر معادلات بروز آوری بردارهای وزن و بایاس متفاوت می گردد که این موضوع فرآیند آموزش را تغییر نمی دهد و فقط معادلات بروز آوری مذکور اندکی تفاوت می کنند. لذا این مورد نیز بصورت یک مورد انتخابی در برنامه گنجانده شده است که در ادامه مشاهده خواهد شد که همین فاکتور فراموشی در نهایت باعث همگرا شدن شبکه و آموزش مناسب آن خواهد شد.

بدین ترتیب با تعداد تکرار مورد نظر که حداقل بایستی برابر طول بردار صوتی (39000×1) باشد می توان شبکه را آموزش داد. همچنین در برنامه آموزش شبکه عصبی امکان تکرار این چرخه تا N بار که هر کدام شامل یک تکرار 39000 تایی می باشد وجود دارد.

جهت درک بهتر نحوه آموزش شبکه عصبی، فلوجارت مطالب ذکر شده در بالا در صفحه بعد بطور کامل آورده شده است که توجه خواننده محترم به آن جلب می شود:



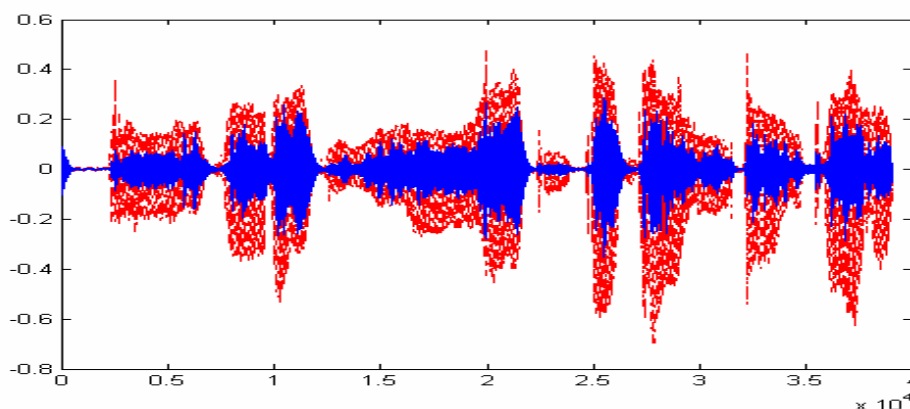
پس از تست های مختلف شبکه مذکور که نتایج آن در ذیل آمده است این نتیجه حاصل شده که شبکه با توجه به مقدار پارامترهای وزن، بایاس و بردار ورودی اولیه در تکرار ۳۰۰۰ تا ۱۰۰۰۰ در مینیمم های محلی گرفتار می شود و برای مدت زیادی پارامترهای آن تغییر نمی کند و در نتیجه آموزش به کندی و نامناسب صورت می گیرد. همچنین در هر تست و با مقایسه سیگنال نخمینی و سیگنال هدف مشاهده می شود که علیرغم همگرا شدن حدودی مجموع مربعات خطا، شبکه بدرستی آموزش ندیده است.

دلایل عدم همگرایی شبکه و نحوه رفع ایراد آن در بخش بعد تشریح شده است. لذا در ادامه نتایج تست های مختلف با پارامترهای اولیه وزن و بایاس مختلف که به عدم همگرایی شبکه منجر شده اند آمده است.

در اولین تست جهت آموزش شبکه عصبی MLP با پارامترهای زیر نتایج ذیل حاصل شده

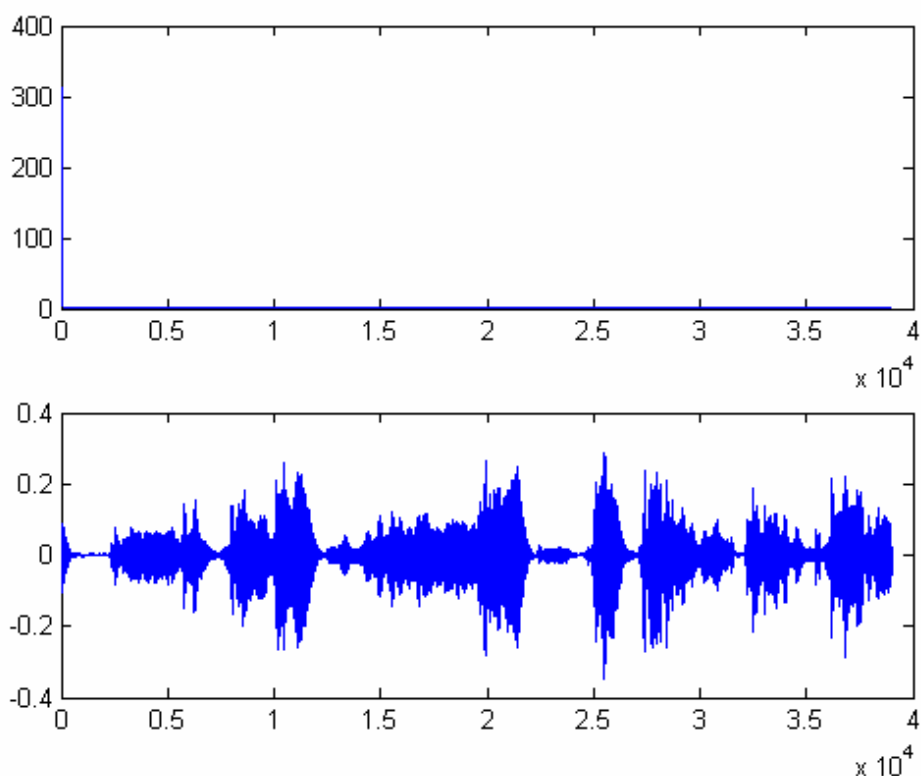
است:

$W1, W2 = \text{randn}(x)$	$A0 = \text{randn}(x)$
$B1, B2 = \text{randn}(x)$	$\alpha = 0.01$
$dW1, dW2 = \text{zeros}(x)$	$\eta = 0$
$dB1, dB2 = \text{zeros}(x)$	$\text{iteration} = 1 * 39000$



شکل ۴-۶- مقایسه خروجی شبکه (نمودار آبی) با سیگنال هدف (نمودار قرمز)

همانطور که در شکل بالا مشاهده می شود خروجی شبکه نتوانسته همپوشانی مناسبی با سیگنال هدف که همان سیگنال صوت تمیز می باشد داشته باشد. همچنین خطای یادگیری شبکه که بصورت میانگین مجموع مربعات خطا³¹ نمایش داده شده است بصورت زیر است:

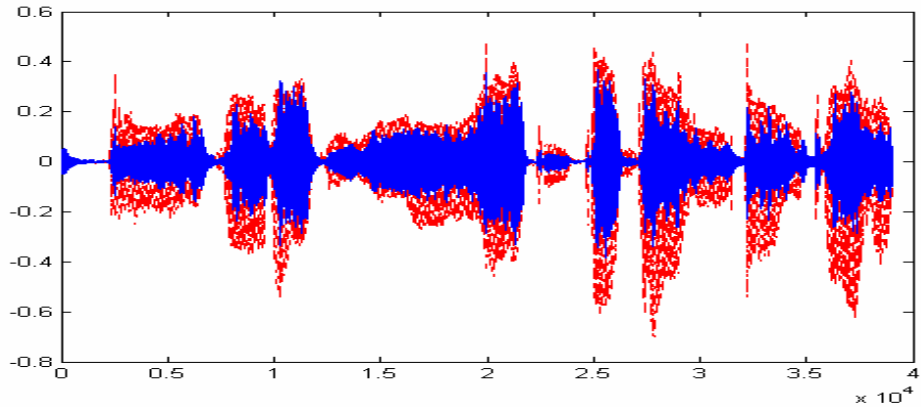


شکل ۴-۷- نمایش خطای یادگیری به همراه خروجی شبکه

همانطور که مشاهده می شود بنظر میرسد که خطای آموزش به صفر همگرا شده است در حالیکه پس از مشاهده شکل صفحه قبل و شنیدن صدای صوت خروجی شبکه عصبی که با دستورات خاصی بازیافت شده است می بینیم که آموزش بطور مناسب صورت نگرفته است لذا پارامتر α را تغییر داده و دوباره تلاش جهت آموزش شبکه آغاز شد که نتایج مشابه آن به همراه پارامترهای آموزش در زیر آمده است:

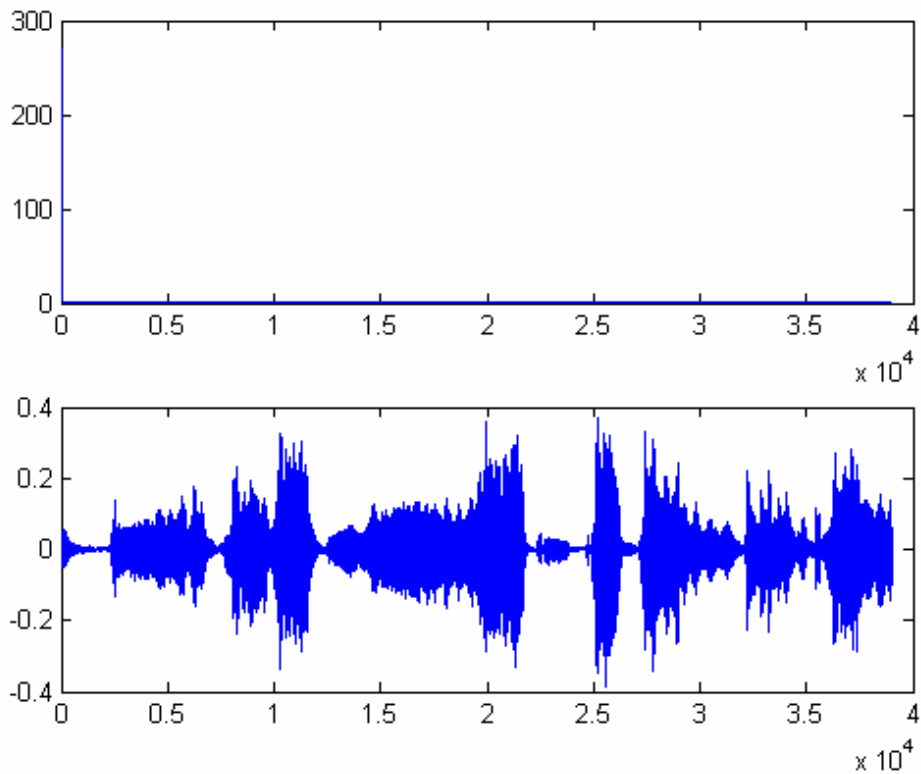
³¹ Minimum Mean Squared Error

$W1, W2 = \text{randn}(x)$	$A0 = \text{randn}(x)$
$B1, B2 = \text{randn}(x)$	$\alpha = 0.05$
$dW1, dW2 = \text{zeros}(x)$	$\eta = 0$
$dB1, dB2 = \text{zeros}(x)$	$\text{iteration} = 2 * 39000$



شکل ۴-۸- مقایسه خروجی شبکه (نمودار آبی) با سیگنال هدف (نمودار قرمز)

در این شکل مشاهده شد که تا حدی آموزش شبکه بهبود یافته و خروجی شبکه به سیگنال هدف نزدیک شده است و نمایش خطای آن نیز بصورت زیر است:

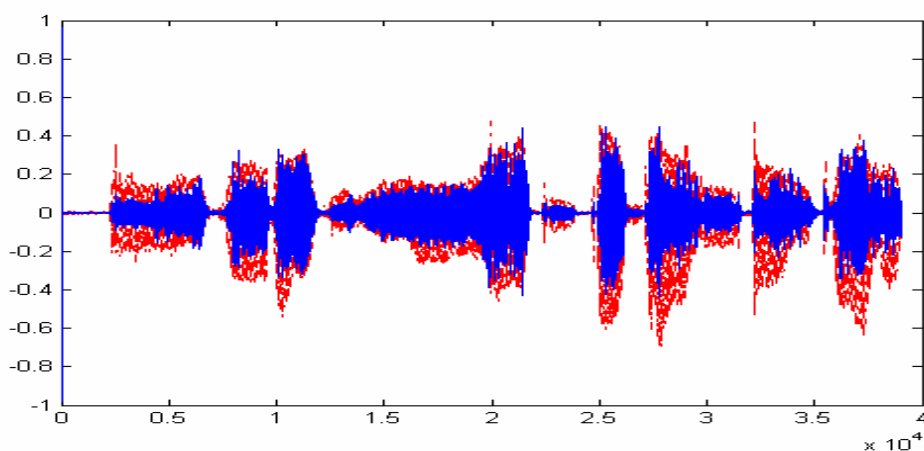


شکل ۴-۹- نمایش خطای آموزش به همراه خروجی شبکه

در این نمایش نیز خطا به سمت صفر همگرا شده است در حالیکه در شنیدن صدای صوت و مشاهده میزان همپوشانی آن با صوت تمیز که در صفحه قبل آمده است متوجه شدیم که هنوز شبکه بطور مناسب آموزش ندیده است.

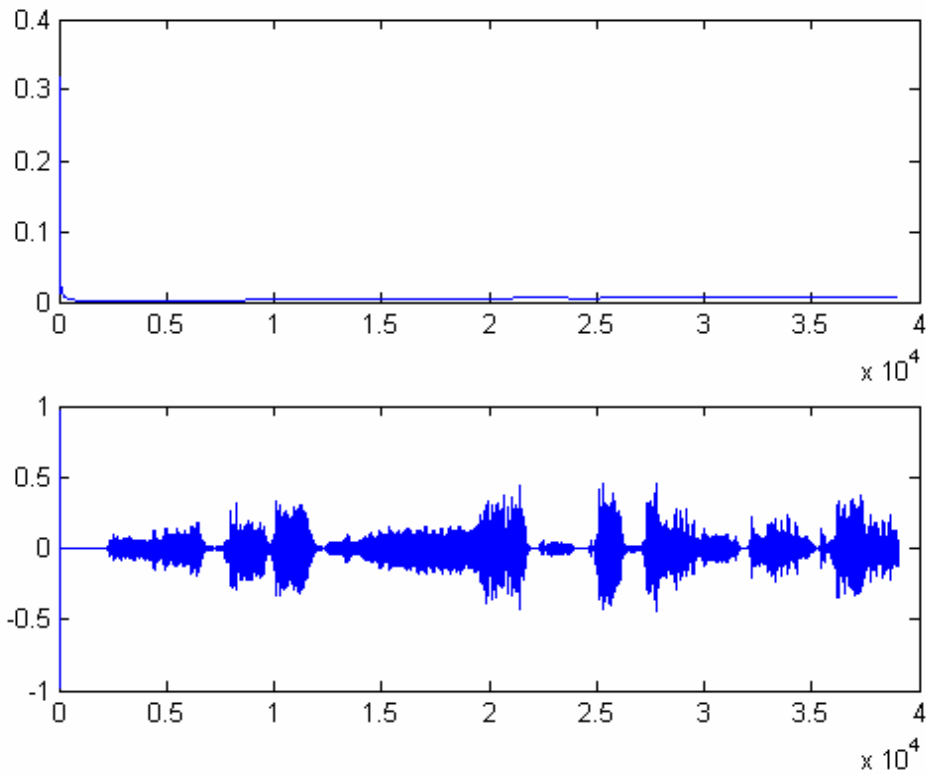
که بدلیل عدم مناسب بودن پاسخ شبکه و گیر افتادن در نقاط مینیمم محلی به فکر افتادیم تا ضریب فراموشی η را نیز در برنامه اضافه کنیم. که با افزایش این ضریب در برنامه پاسخ شبکه بهتر شد اما هنوز مشکلی در برنامه وجود داشت زیرا صوت خروجی از کیفیت بالایی برخوردار نبود. نمایش پاسخهای شبکه با ضریب فراموشی به شرح زیر است:

$W1, W2 = \text{randn}(x)$	$A0 = \text{randn}(x)$
$B1, B2 = \text{randn}(x)$	$\alpha = 0.01$
$dW1, dW2 = \text{zeros}(x)$	$\eta = 0.01$
$dB1, dB2 = \text{zeros}(x)$	$\text{iteration} = 1 * 39000$



شکل ۴-۱۰- مقایسه خروجی شبکه (نمودار آبی) با سیگنال هدف (نمودار قرمز)

همانطور که از شکل بالا مشخص است خروجی شبکه تا حد زیادی توانسته خود را با سیگنال هدف که همان سیگنال تمیز می باشد تطبیق نماید لذا اضافه کردن ضریب فراموشی نتایج را تا حد زیادی بهبود بخشید و خطای آموزش شبکه بصورت زیر تغییر کرد:

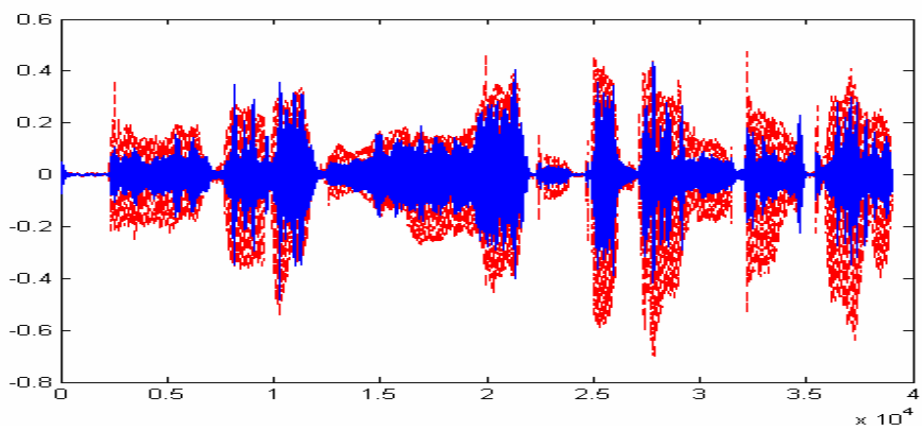


شکل ۴-۱۱- نمایش خطای Training به همراه خروجی شبکه

همانطور که مشاهده می شود مجموع مربعات خطا نه تنها بهبود نیافته بلکه ظاهراً بدتر از شکل‌های قبل شده است در حالیکه مقایسه نتایج خروجی شبکه و سیگنال هدف موید این مطلب است که یادگیری شبکه بهبود یافته است لذا مجموع مربعات خطا به تنهایی نمی تواند شاخص مناسبی جهت ارزیابی شناسایی یک سیستم غیر خطی باشد.

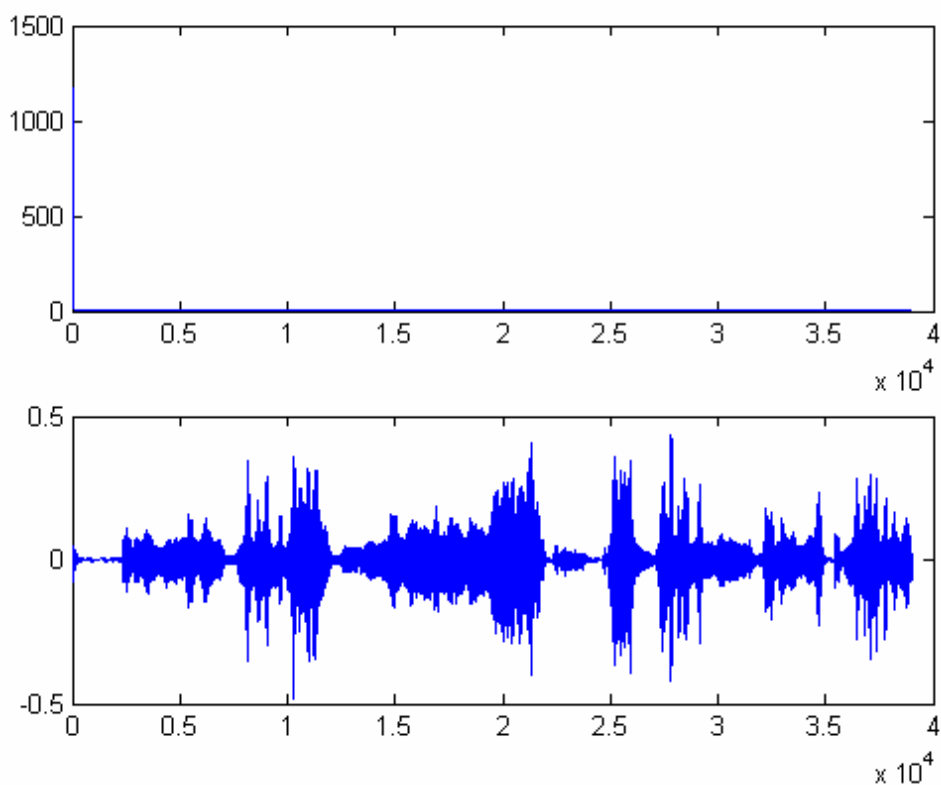
پارامترهای تست بعدی به همراه نتایج آن به شرح ذیل است:

$W1, W2 = 0.1 * \text{randn}(x)$	$A0 = \text{randn}(x)$
$B1, B2 = 0.1 * \text{randn}(x)$	$\alpha = 0.01$
$dW1, dW2 = \text{zeros}(x)$	$\eta = 0.005$
$dB1, dB2 = \text{zeros}(x)$	$\text{iteration} = 1 * 39000$



شکل ۴-۱۲- مقایسه خروجی شبکه (نمودار آبی) با سیگنال هدف (نمودار قرمز)

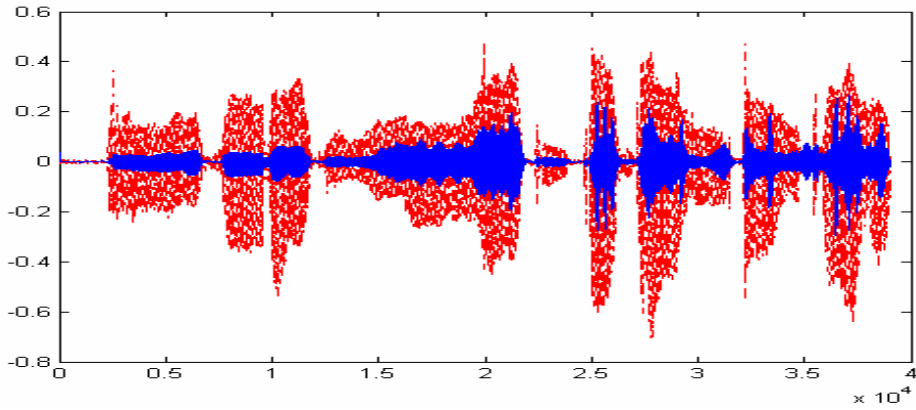
که همانگونه که مشاهده می شود آموزش با این پارامترها نتایج خوبی را در بر نداشته است. و نمایش خطای آموزش شبکه بصورت زیر است:



شکل ۴-۱۳- نمایش خطای یادگیری به همراه خروجی شبکه

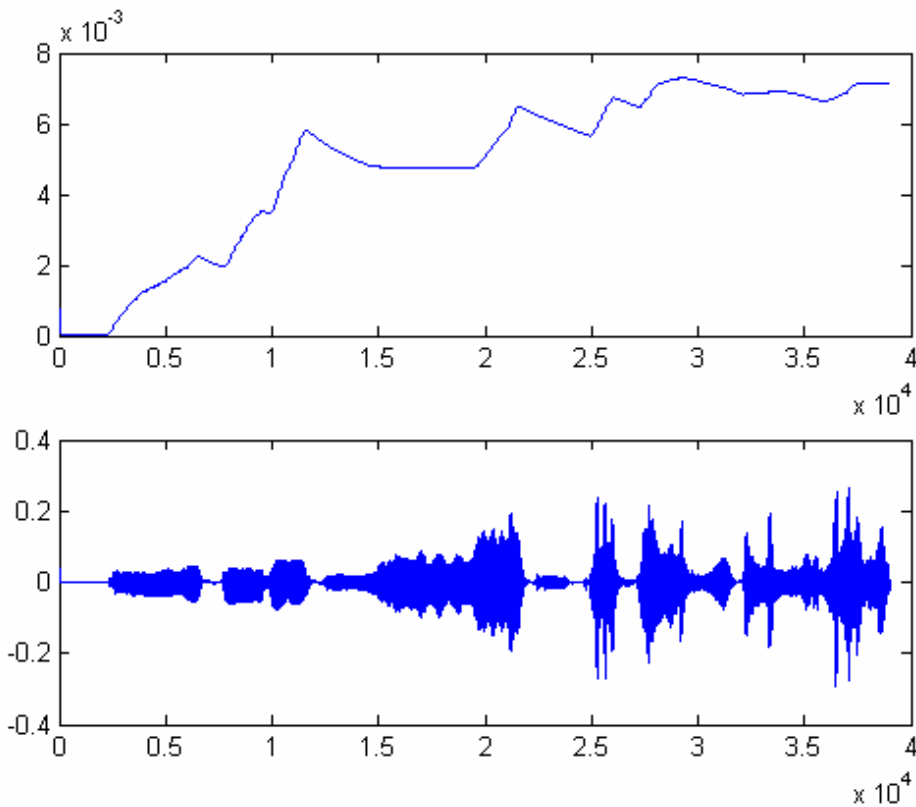
و همانطور که مشخص است خطای آموزش به صفر همگرا شده است ولی این نتایج مورد رضایت نبود لذا تست این بار با تعداد تکرار بیشتر ادامه یافت:

$W1, W2 = 0.1 * \text{randn}(x)$	$A0 = \text{randn}(x)$
$B1, B2 = 0.1 * \text{randn}(x)$	$\alpha = 0.01$
$dW1, dW2 = 0.1 * \text{randn}(x)$	$\eta = 0.005$
$dB1, dB2 = 0.1 * \text{randn}(x)$	$\text{iteration} = 2 * 39000$



شکل ۴-۱۴- مقایسه خروجی شبکه (نمودار آبی) با سیگنال هدف (نمودار قرمز)

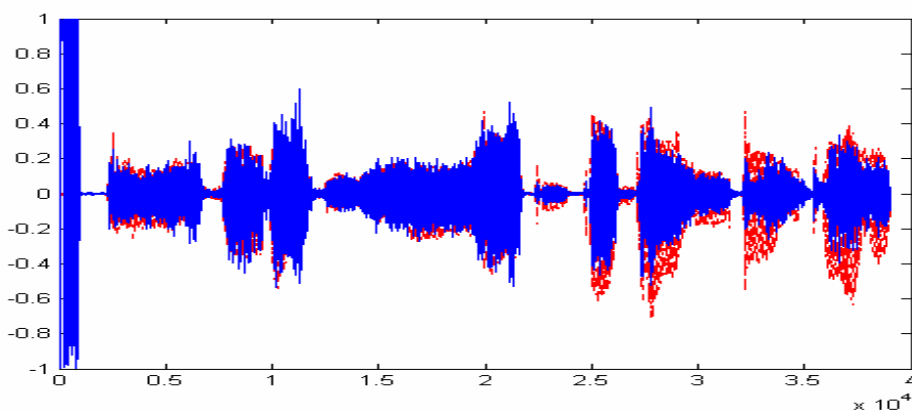
همانطور که مشاهده می شود با افزایش تکرار الگوریتم آموزشی اوضاع شبکه وخیم تر شد و حتی خطای آن نیز بشدت واگرا شد:



شکل ۴-۱۵- نمایش خطای یادگیری به همراه خروجی شبکه

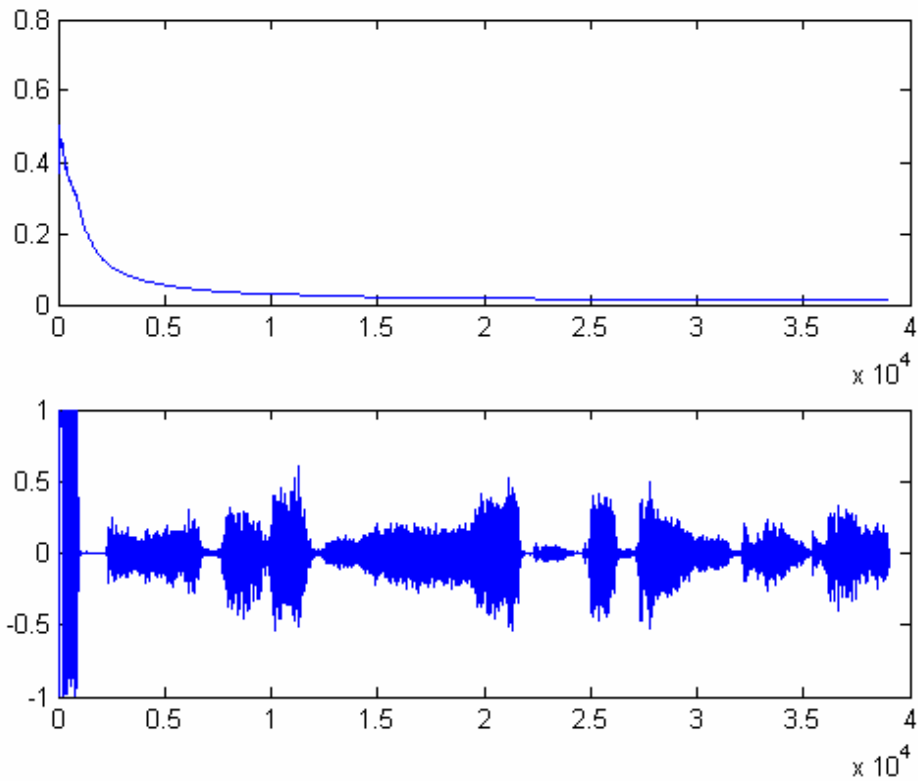
همانطور که در شکل بالا مشخص است خطای یادگیری بشدت واگرا شده است لذا در مرحله بعد با تغییر پارامترها و کاهش تکرار الگوریتم به نتایج زیر رسیدیم:

$W1, W2 = \text{randn}(x)$	$A0 = 10 * \text{randn}(x)$
$B1, B2 = \text{randn}(x)$	$\alpha = 0.01$
$dW1, dW2 = \text{zeros}(x)$	$\eta = 0.01$
$dB1, dB2 = \text{zeros}(x)$	$\text{iteration} = 1 * 39000$



شکل ۴-۱۶- مقایسه خروجی شبکه (نمودار آبی) با Target (نمودار قرمز)

که البته در این مرحله و با این پارامترهای جدید نتایج شبکه خیلی بهبود یافت ولی هنوز از نتایج رضایت کامل حاصل نشده بود. همچنین نتایج خطای آن در ذیل آمده است:

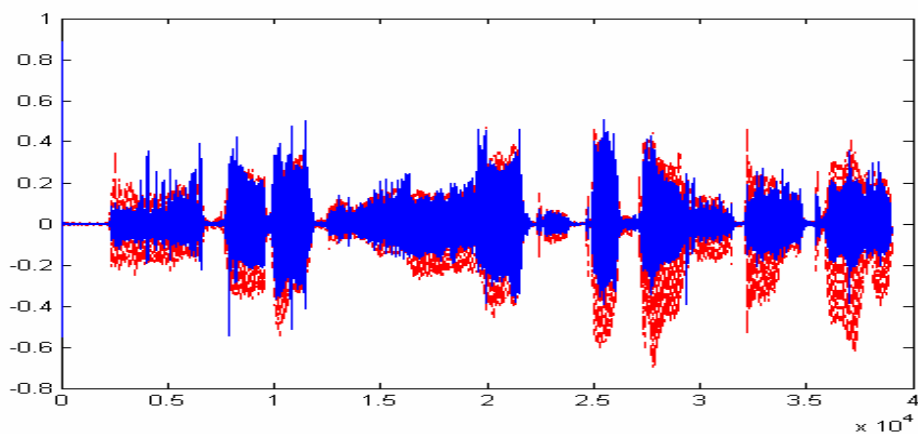


شکل ۴-۱۷- نمایش خطای Training به همراه خروجی شبکه

و همانطور که مشاهده می شود خطای آموزش که همان مجموع مربعات خطا می باشد دارای همگرایی مناسبی به صفر می باشد که البته از سرعت کمتری نسبت به حالت‌های قبل برخوردار است.

لذا تست را با پارامترهای ذیل ادامه دادیم:

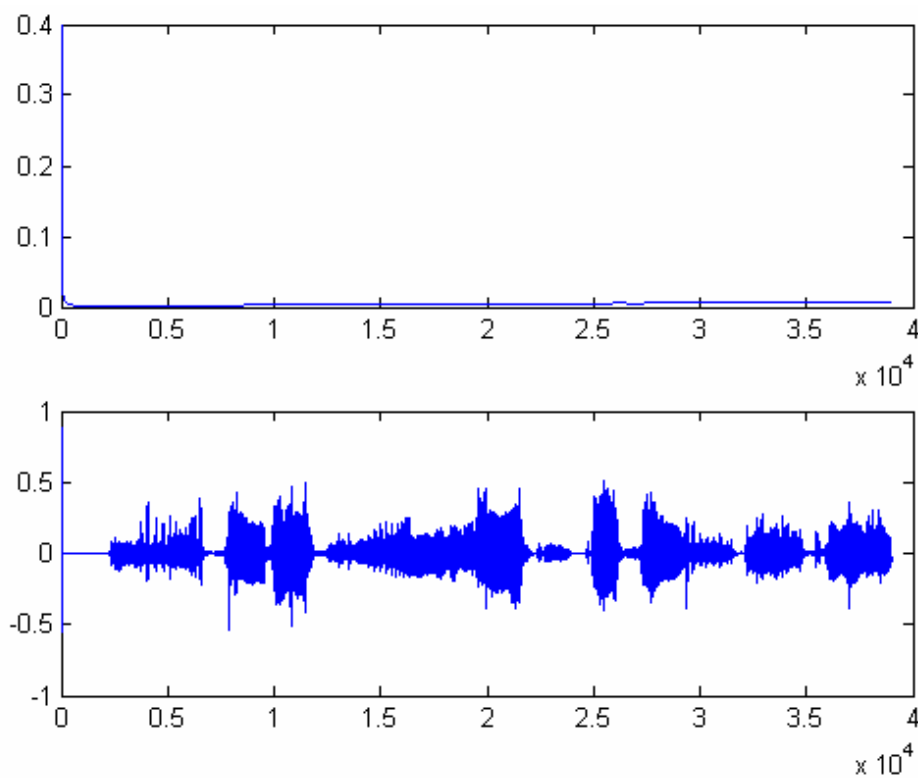
$W1, W2 = \text{randn}(x)$	$A0 = \text{randn}(x)$
$B1, B2 = \text{randn}(x)$	$\alpha = 0.05$
$dW1, dW2 = \text{zeros}(x)$	$\eta = 0.01$
$dB1, dB2 = \text{zeros}(x)$	$\text{iteration} = 1 * 39000$



شکل ۴-۱۸- مقایسه خروجی شبکه (نمودار آبی) با سیگنال هدف (نمودار قرمز)

در این مرحله نیز نتایج تا حدی بهبود یافت ولی به نتایج مورد نظر نرسیدیم. نتایج خطای آموزش

شبکه نیز بصورت زیر است:



شکل ۴-۱۹- نمایش خطای آموزش به همراه خروجی شبکه

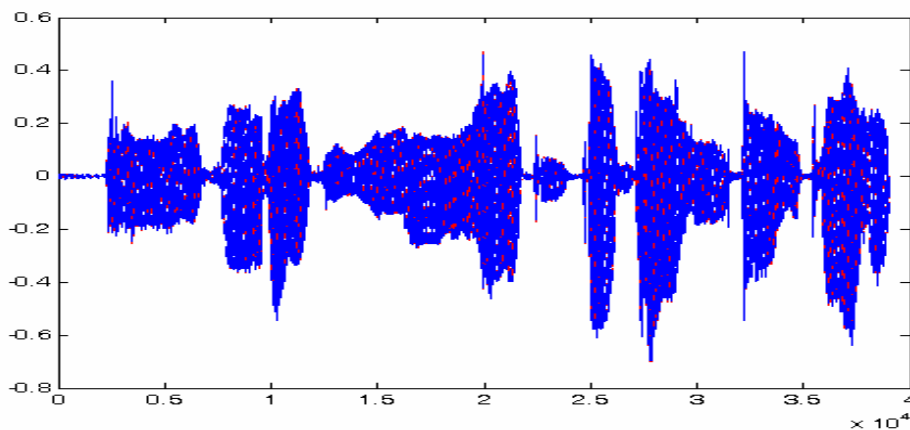
۴-۳-۴- نحوه همگرا شدن شبکه عصبی :

همانطور که مشخص است همه خطاهای یادگیری و خروجیها دارای بهبودهای نسبی می شدند اما بطور کامل همگرا نمی شدند و همانطور که مشاهده می شود هنوز شبکه آنطور که باید و شاید آموزش ندیده بود. لذا پس از بازنگری کامل و جستجوی فراوان متوجه شدیم که:

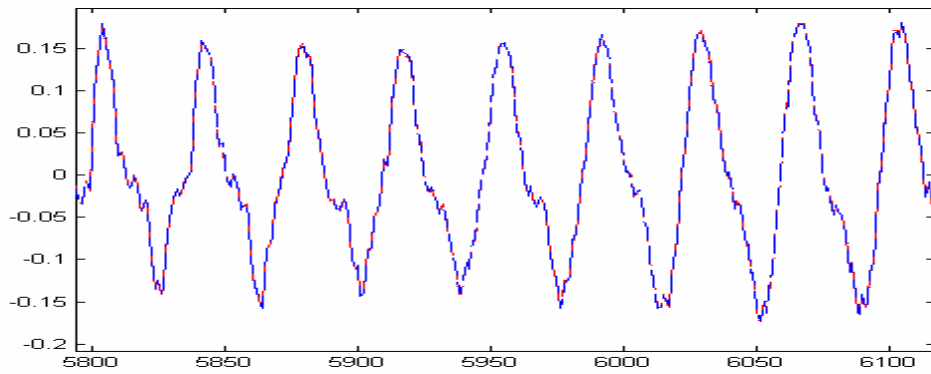
- ۱- مقدار $A0$ پس از هر ۳۹۰۰۰ تکرار که در برخی موارد لازم بود این ۳۹۰۰۰ بار K بار دیگر تکرار گردد، دوباره مقادیر جدیدی می گرفت و این امر باعث می شد کلیه بازخوردهای قبلی که از خروجی به ورودی شبکه داده شده بود بی اثر گردد لذا افزایش تکرار اثری در همگرایی شبکه نداشت. که این امر با تغییر در تابع نوشته شده اصلاح گردید.
- ۲- این بخش با تلفیق مورد بالا با مقادیر اولیه مناسبی که از تست های قبلی نتیجه گیری شده بود نهایتا به آموزش مناسب با تکرار ۳۹۰۰۰ بار می رسد.

نمایش خروجی شبکه در حالت نهایی به شرح زیر می باشد:

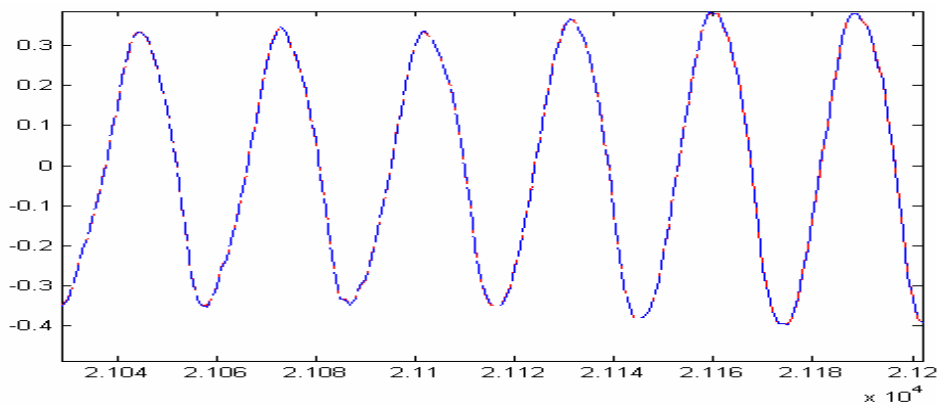
$W1, W2 = \text{randn}(x)$	$A0 = \text{randn}(x)$
$B1, B2 = \text{randn}(x)$	$\alpha = 0.05$
$dW1, dW2 = \text{zeros}(x)$	$\eta = 0.01$
$dB1, dB2 = \text{zeros}(x)$	$\text{iteration} = 1 * 39000$



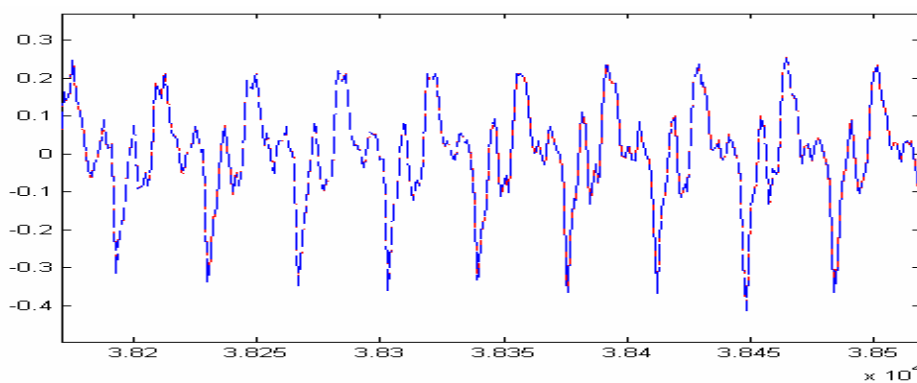
شکل ۴-۲۰- مقایسه خروجی شبکه (نمودار آبی) با سیگنال هدف (نمودار قرمز)



شکل ۴-۲۱- مقایسه خروجی شبکه (نمودار آبی) با سیگنال هدف (نمودار قرمز) در ۳۰۰ تکرار ابتدایی



شکل ۴-۲۲- مقایسه خروجی شبکه (نمودار آبی) با سیگنال هدف (نمودار قرمز) در ۲۰۰ تکرار میانی

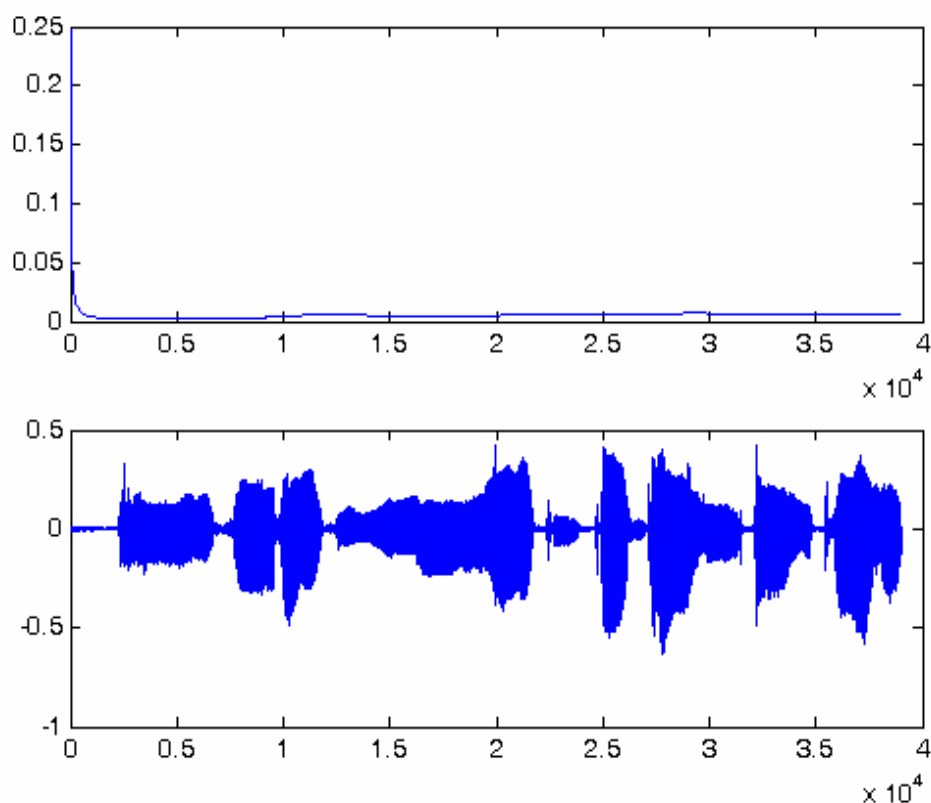


شکل ۴-۲۳- مقایسه خروجی شبکه (نمودار آبی) با سیگنال هدف (نمودار قرمز) در ۳۰۰ تکرار انتهایی

همانگونه که از شکل ۴ بالا مشخص است می بینیم که بالاخره با بکارگیری ضرایب اولیه و ضریب فراموشی مناسب به همراه رفع ایراد برنامه که نحوه آن در صفحه قبل تشریح شد توانستیم

خروجی شبکه عصبی را کاملاً منطبق بر سیگنال هدف ببینیم که این نشان دهنده آموزش کامل و در نهایت شناسایی سیستم غیرخطی صوت می باشد.

همچنین خطای آموزش شبکه عصبی نیز بصورت زیر به نمایش درآمد:



شکل ۴-۲۴- نمایش خطای آموزش به همراه خروجی شبکه

همچنین خطای نمایش داده شده در بالا خیال ما را از همگرا شدن شبکه مطمئن می کند و نشان می دهد که صحت نمایشهای ۴-۲۰ تا ۴-۲۳ کاملاً دقیق و صحیح است.

۴-۴- موضوع تخمین :

پس از طی مراحل بالا که منجر به مدلسازی و هموارسازی راه جهت مشخص شدن دینامیکهای سیستم شد نوبت به تخمین مدل غیرخطی می رسد. اما این بار بخاطر اینکه با یک مدل غیر خطی روبرو هستیم ناگزیر بایستی بجای فیلتر کالمن از فیلتر کالمن توسعه یافته (EKF) استفاده کنیم.

فیلتر کالمن توسعه یافته (EKF) یک روش تقریبی برای مدل‌های غیر خطی است که مدل غیر خطی را با عنوان مدل خطی متغیر با زمان تخمین می‌زند. ذکر این نکته حائز اهمیت است که یک ویژگی تغییر ناپذیر فیلتر کالمن نیلز به معلوم بودن دینامیک‌های سیستم است. بعبارت دیگر جهت تخمین سیگنال با فیلتر کالمن از هر نوع (معمولی، توسعه یافته، توسعه یافته دوگانه) تخمین مدل یک امر ضروری است.

بنابراین در این بخش ابتدا به نمایش فضای حالت سیستم غیر خطی پرداخته شده و در ادامه معادلات فیلتر کالمن توسعه یافته معرفی گردیده است:

$$X_k = F(X_{k-1}, W) + B.V_k \quad (1-4-4)$$

$$\begin{bmatrix} x_k \\ x_{k-1} \\ \vdots \\ x_{k-M+1} \end{bmatrix} = \begin{bmatrix} f(x_{k-1}, \dots, x_{k-M}, W) \\ 1 & 0 & 0 & 0 \\ 0 & \ddots & 0 & \ddots \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_{k-1} \\ \vdots \\ x_{k-M} \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} .v_k$$

$$Y_k = C . X_k + n_k \quad (2-4-4)$$

$$y_k = [1 \quad 0 \quad \dots \quad 0] \cdot \begin{bmatrix} x_k \\ x_{k-1} \\ \vdots \\ x_{k-M+1} \end{bmatrix} + n_k$$

نمایش معادلات فیلتر کالمن توسعه یافته :

۱- مقدار دهی اولیه :

$$\hat{x}(0) = E(x_0) \quad (3-4-4)$$

$$P_0 = E[(x_0 - \hat{x}_0)(x_0 - \hat{x}_0)^T]$$

۲- معادلات بروز کننده زمان :

$$A_k = \left. \frac{\partial F(X, W)}{\partial X} \right|_{x=\hat{x}_k}$$

$$A_k = \begin{bmatrix} \frac{\partial f(\hat{x}_k, w)^T}{\partial x} \\ \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \ddots & 0 & \ddots \\ 0 & 1 & 0 & 0 \end{bmatrix} \end{bmatrix} \quad (4-4-4)$$

$$\hat{x}_k^- = F(\hat{x}_{k-1}, w) \quad (5-4-4)$$

$$P_k^- = AP_{k-1}A^T + B\sigma_v^2B^T$$

۳- معادلات بروز کننده مشاهده :

$$K = P_k^- C^T (CP_k^- C^T + \sigma_n^2)^{-1}$$

$$\hat{x}_k = \hat{x}_k^- + K(y_k - C\hat{x}_k^-) \quad (6-4-4)$$

$$P_k = (I - KC).P_k^-$$

که همانطور که در بخش قبل توضیح داده شد بخاطر ۵ ورودی شبکه عصبی بعد ماتریس A 5×5 تعریف شد که در واقع مقدار k از عدد ۶ شروع شد و به ترتیب به ۷ تا ۳۹۰۰۶ رسید که در نهایت به ما یک بردار تخمین 1×39000 را ارائه داد.

نحوه بدست آوردن ماتریس A_6 که اولین ماتریس A می باشد بصورت زیر است:

$$A_6 = \begin{bmatrix} \frac{\hat{x}_7^- - \hat{x}_6^-}{\hat{x}_2^- - \hat{x}_1^-} & \frac{\hat{x}_7^- - \hat{x}_6^-}{\hat{x}_3^- - \hat{x}_2^-} & \frac{\hat{x}_7^- - \hat{x}_6^-}{\hat{x}_4^- - \hat{x}_3^-} & \frac{\hat{x}_7^- - \hat{x}_6^-}{\hat{x}_5^- - \hat{x}_4^-} & \frac{\hat{x}_7^- - \hat{x}_6^-}{\hat{x}_6^- - \hat{x}_5^-} \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

سپس با قرار گرفتن در برنامه بالا مقدار \hat{x}_6 محاسبه می شود و این محاسبات به تعداد ۳۹۰۰۰ بار یعنی تا \hat{x}_{39006} ادامه یافته و این مقادیر در حافظه برنامه ثبت می شود تا در انتهای برنامه فرخوانی شده و با صوت تمیز مقایسه گردد.

همچنین برنامه های Network Initialize و EKF و توابع Batchbp1 Function و Batchbp2 Function که منجر به شناسایی و تخمین حالت صوت گردیده در پیوست ۱ موجود می باشند..

۴-۴-۱- نحوه تخمین سیگنال صوتی:

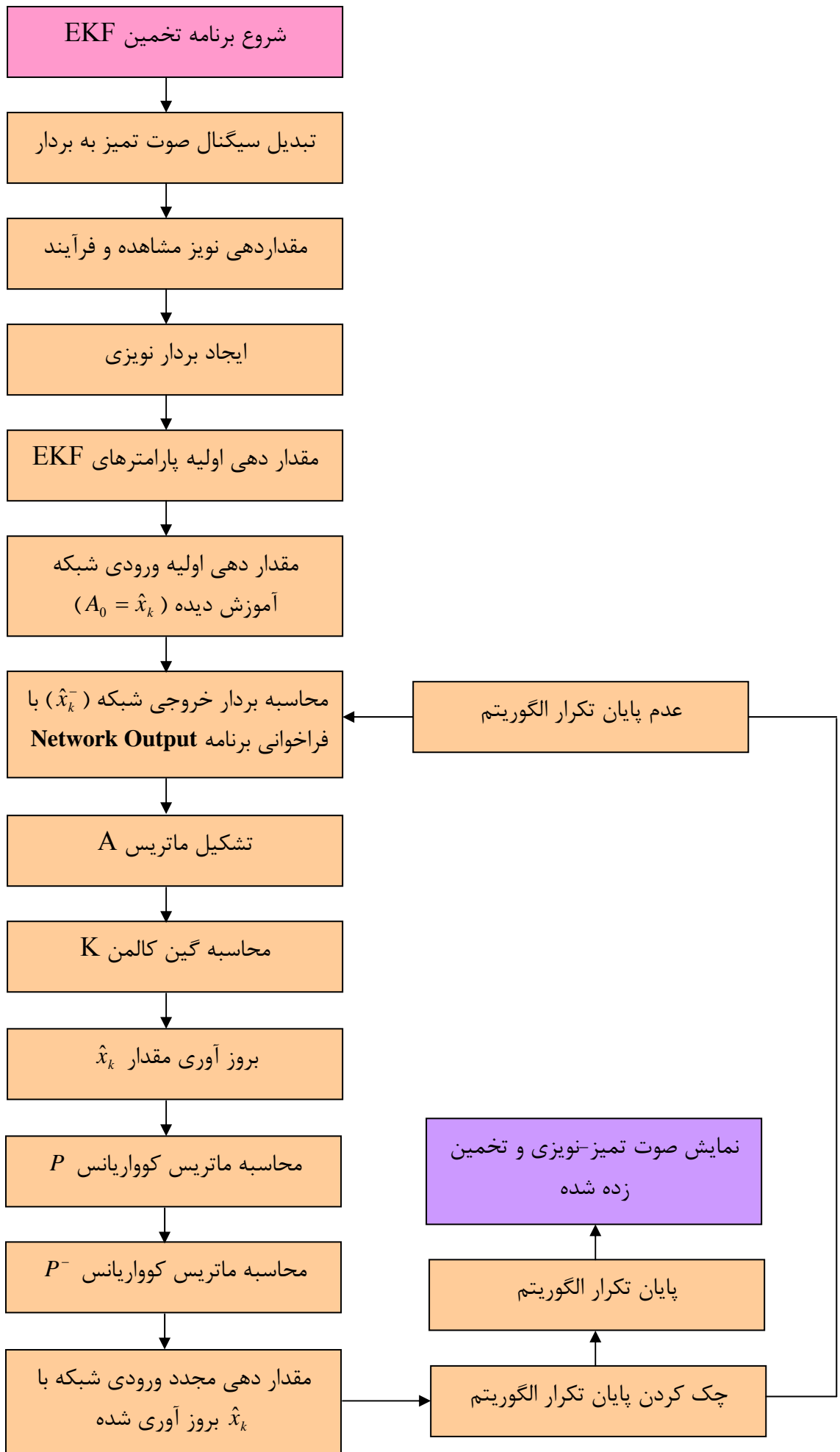
با توجه به توضیحات ارائه شده در بالا در این بخش به تشریح چگونگی اجرای برنامه مربوط به تخمین سیگنال صوتی پرداخته شده است.

پس از شناسایی صوت توسط شبکه عصبی و همگرا شدن شبکه که نتایج آن در بخش قبل ارائه شد ۲ برنامه دیگر که یکی جهت محاسبه خروجی شبکه عصبی و دیگری جهت محاسبه تخمین EKF می باشد نوشته شده است.

تابعی که جهت محاسبه خروجی شبکه عصبی بکار رفته Network Output و برنامه ای که جهت محاسبه تخمین فیلتر کالمن توسعه یافته بکار رفته Extended Kalman Filter نام دارد. نحوه اجرای برنامه EKF بدین صورت است که ابتدا سیگنال صوتی تمیز با دستور Wavread به بردار تبدیل می شود. سپس واریانس نویز مشاهده و نویز پروسه در برنامه تعریف می شود و در ادامه با جمع بردارهای نویز مشاهده و سیگنال تمیز، بردار سیگنال نویزی ساخته می شود.

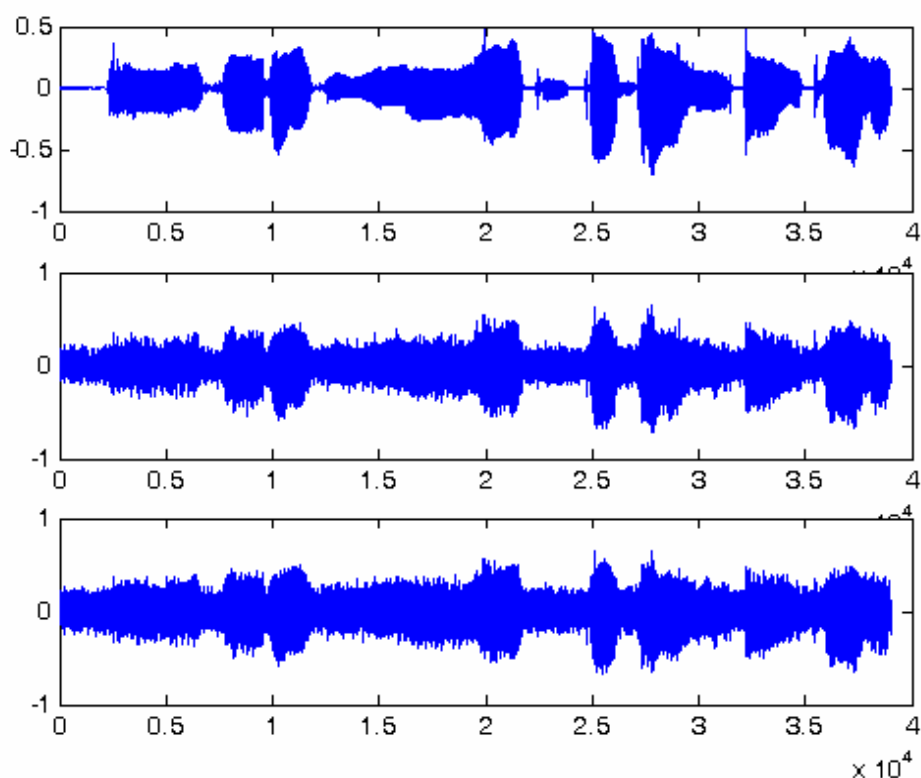
سپس پارامترهای فیلتر کالمن توسعه یافته جهت شروع برنامه مقدار دهی اولیه می شوند در ادامه برنامه Network Output فراخوانی می شود که در آن یک ورودی اولیه به شبکه عصبی آموزش دیده اعمال شده و خروجی شبکه محاسبه می گردد. سپس به برنامه اصلی EKF برگشته و برنامه با محاسبه ماتریس A که چگونگی تشکیل آن در بخش قبل تشریح شد ادامه می یابد. سپس ماتریس گین فیلتر کالمن توسعه یافته (K) محاسبه می گردد. همچنین بدنبال آن حالت \hat{x}_k^- بروز آوری می شود. سپس ماتریسهای کوواریانس P محاسبه می گردد و در ادامه \hat{x}_k بعنوان ورودی جدید به شبکه اعمال می گردد و چرخه برنامه تا برآورده شدن شرط آن که برابر شدن سیگنال تخمینی حاصل از محاسبه EKF و سیگنال تمیز است ادامه می یابد.

فلوچارت برنامه نوشته شده جهت تخمین صوت بصورت زیر است:

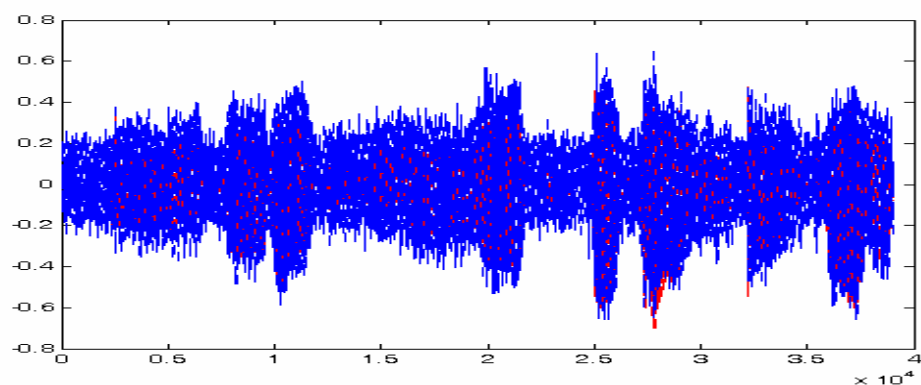


در ادامه نتایج برنامه EKF جهت تخمین سیگنال صوتی تمیز در زمانی که شناسایی بدرستی انجام نگرفته بود و شبکه درست آموزش ندیده بود نمایش داده شده است و همانطور که از نمایشهای زیر مشخص است بدلیل عدم شناسایی درست سیستم توسط شبکه عصبی که دلایل آن در بخش قبل ذکر گردید فیلتر کالمن توسعه یافته نتوانسته یک تخمین مناسب از سیگنال صوتی تمیز ارائه دهد .

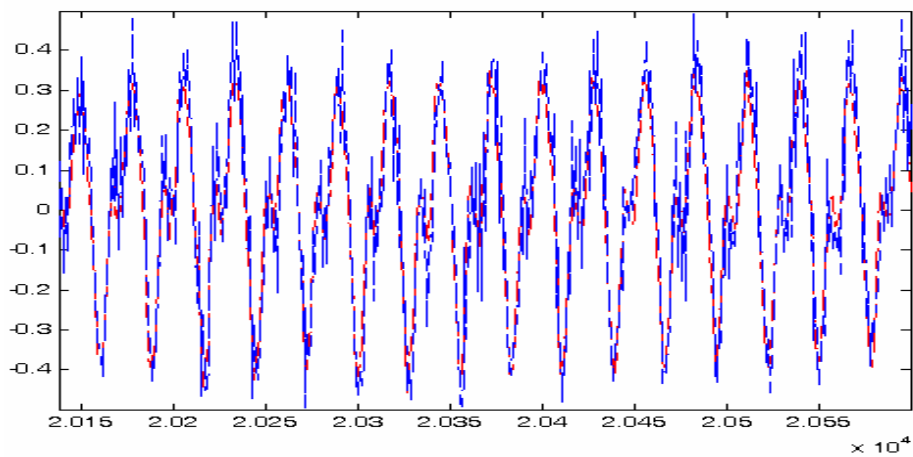
نمایش سیگنال تخمینی به همراه سیگنال تمیز در ذیل آمده است :



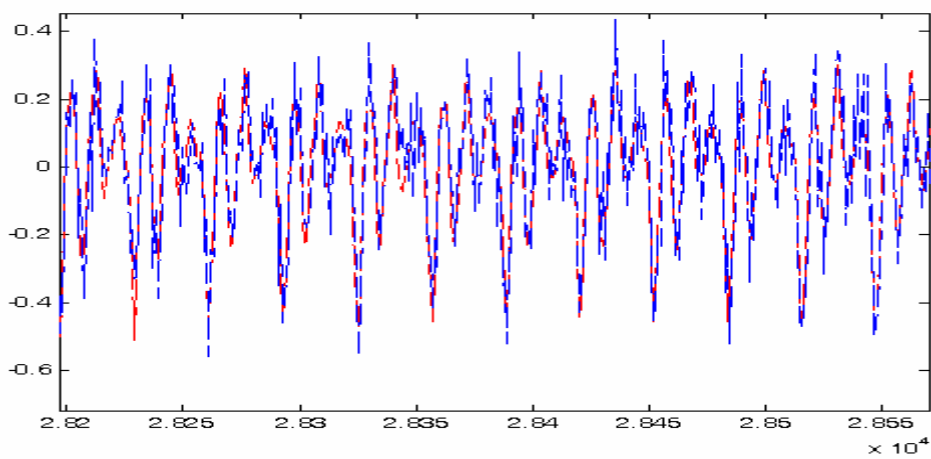
شکل ۴-۲۵- مقایسه سیگنال تمیز (شکل اول) سیگنال نویزی (شکل دوم) و سیگنال تخمینی (شکل سوم)



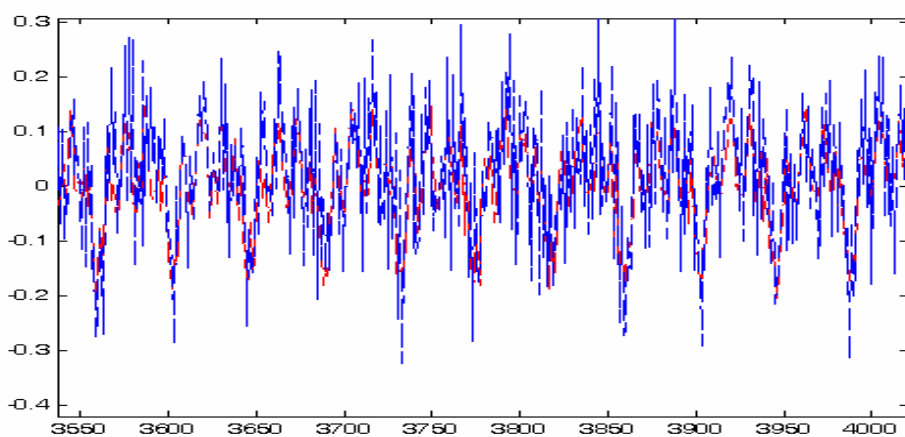
شکل ۴-۲۶- مقایسه سیگنال تمیز (نمودار قرمز) و سیگنال تخمینی (نمودار آبی)



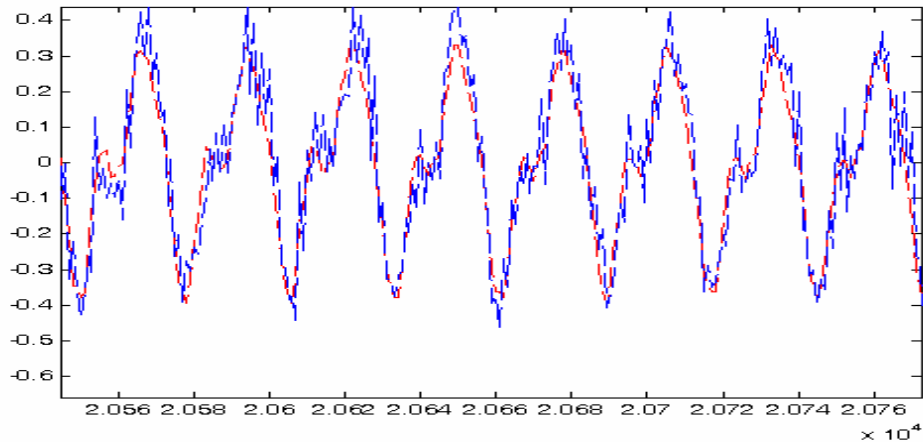
شکل ۴-۲۷- مقایسه دقیق سیگنال تمیز (نمودار قرمز) و سیگنال تخمینی (نمودار آبی) در ۵۰۰ تکرار



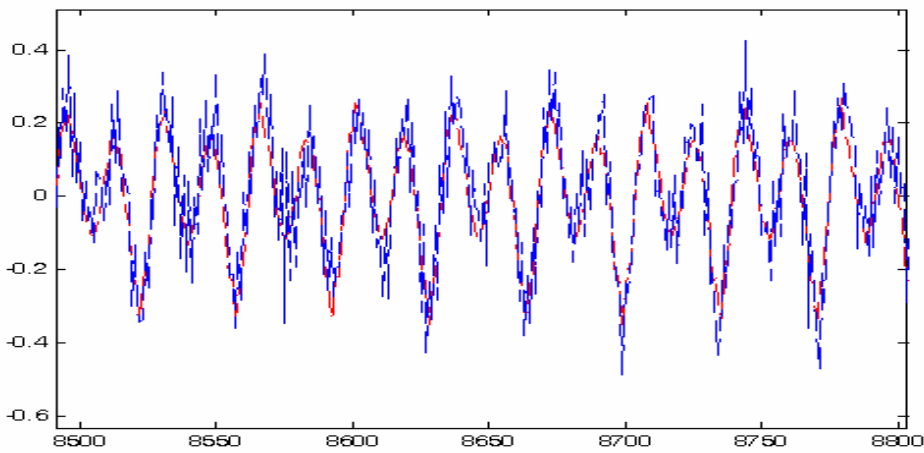
شکل ۴-۲۸- مقایسه دقیق سیگنال تمیز (نمودار قرمز) و سیگنال تخمینی (نمودار آبی) در ۵۰۰ تکرار



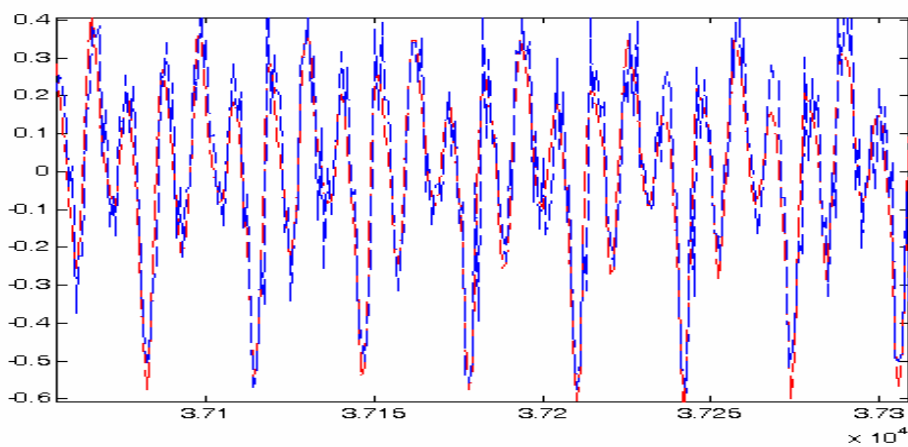
شکل ۴-۲۹- مقایسه دقیق سیگنال تمیز (نمودار قرمز) و سیگنال تخمینی (نمودار آبی) در ۵۰۰ تکرار



شکل ۴-۳۰- مقایسه دقیق سیگنال تمیز (نمودار قرمز) و سیگنال تخمینی (نمودار آبی) در ۵۰۰ تکرار

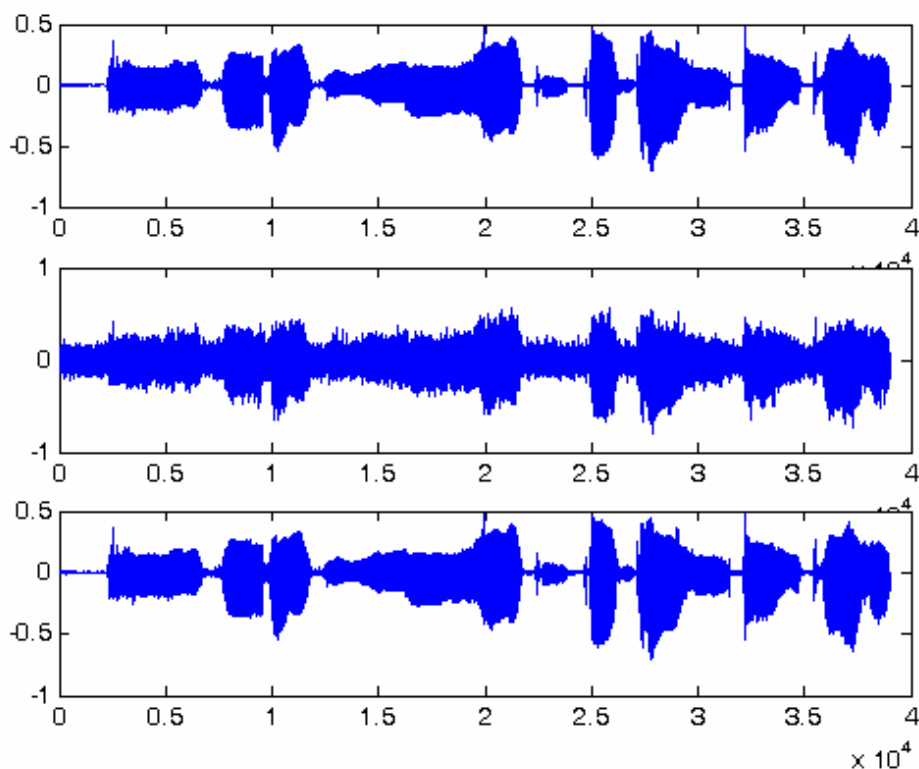


شکل ۴-۳۱- مقایسه دقیق سیگنال تمیز (نمودار قرمز) و سیگنال تخمینی (نمودار آبی) در ۵۰۰ تکرار

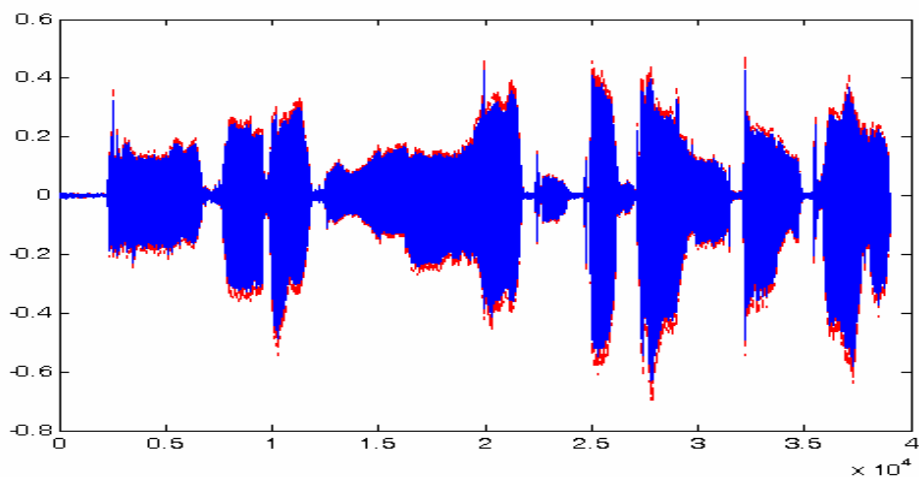


شکل ۴-۳۲- مقایسه دقیق سیگنال تمیز (نمودار قرمز) و سیگنال تخمینی (نمودار آبی) در ۵۰۰ تکرار

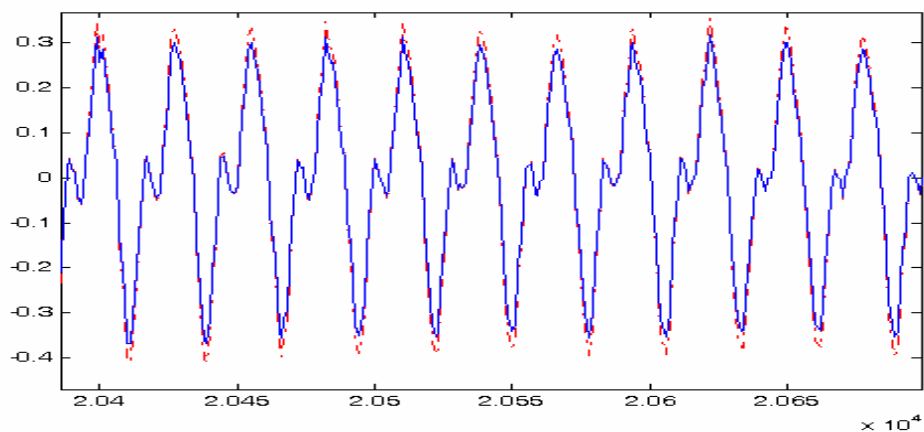
اما پس از شناسایی درست و صحیح سیستم غیر خطی توسط شبکه عصبی و آموزش دیدن صحیح شبکه عصبی، دوباره کالمن تسعه یافته جهت تخمین سیگنال صوتی تمیز بکارگیری شد که این بار مشاهده شد بخوبی کالمن فیلتر توانست سیگنال را تخمین زده و یک تخمین دقیق به ما ارائه دهد. نمایش این تخمینها بصورت زیر است:



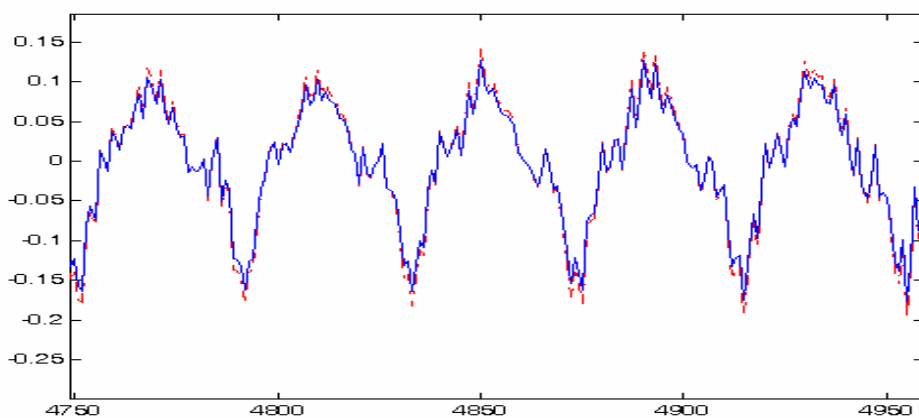
شکل ۴-۳۳- مقایسه سیگنال تمیز (شکل اول) سیگنال نویزی (شکل دوم) و سیگنال تخمینی (شکل سوم)



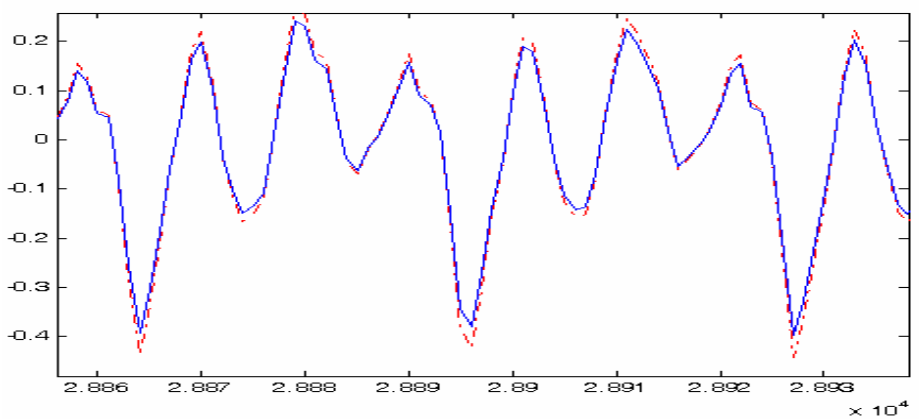
شکل ۴-۳۴- مقایسه سیگنال تمیز (نمودار قرمز) و سیگنال تخمینی (نمودار آبی)



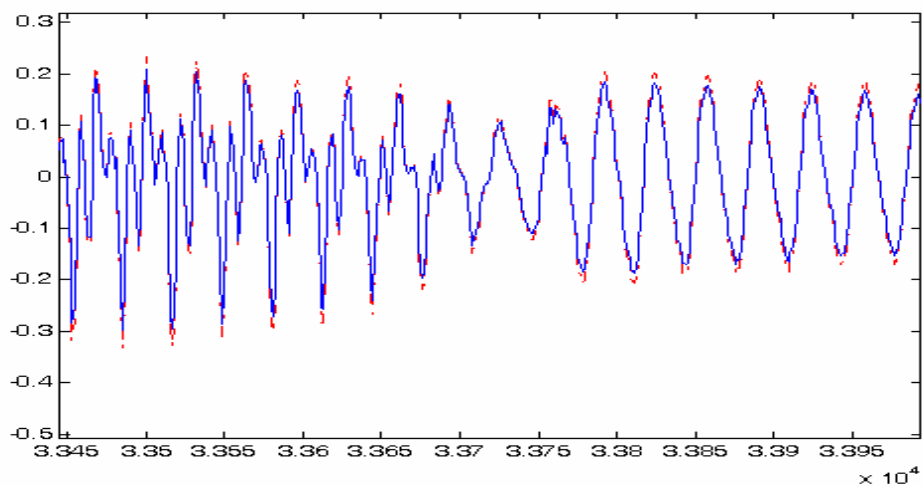
شکل ۴-۳۵- مقایسه دقیق سیگنال تمیز (نمودار قرمز) و سیگنال تخمینی (نمودار آبی) در ۲۵۰ تکرار



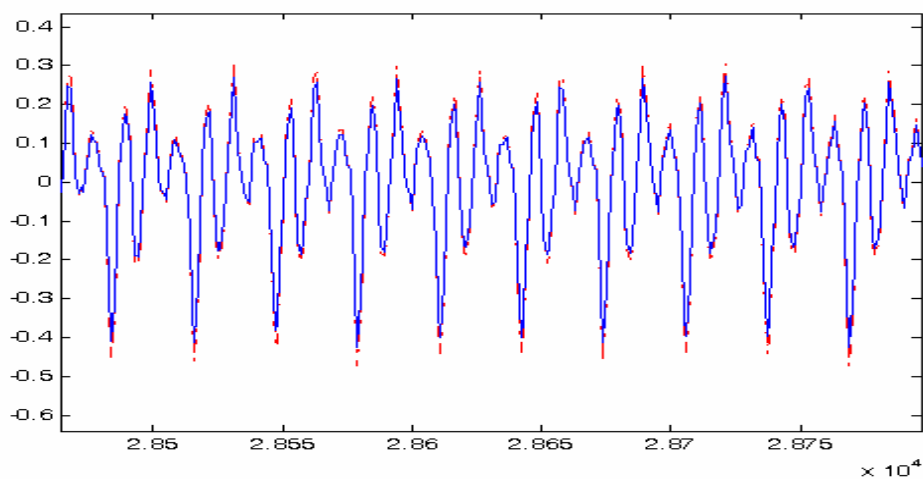
شکل ۴-۳۶- مقایسه دقیق سیگنال تمیز (نمودار قرمز) و سیگنال تخمینی (نمودار آبی) در ۲۵۰ تکرار



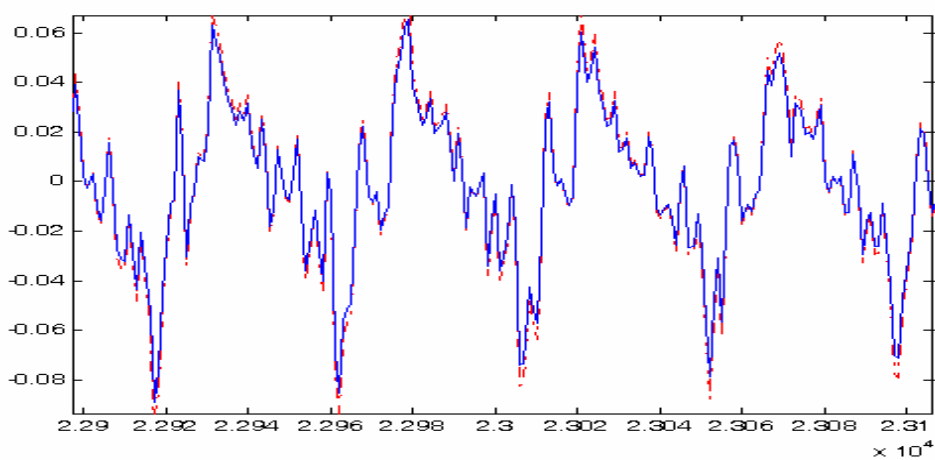
شکل ۴-۳۷- مقایسه دقیق سیگنال تمیز (نمودار قرمز) و سیگنال تخمینی (نمودار آبی) در ۲۵۰ تکرار



شکل ۴-۳۸- مقایسه دقیق سیگنال تمیز (نمودار قرمز) و سیگنال تخمینی (نمودار آبی) در ۲۵۰ تکرار



شکل ۴-۳۹- مقایسه دقیق سیگنال تمیز (نمودار قرمز) و سیگنال تخمینی (نمودار آبی) در ۲۵۰ تکرار



شکل ۴-۴۰- مقایسه دقیق سیگنال تمیز (نمودار قرمز) و سیگنال تخمینی (نمودار آبی) در ۲۵۰ تکرار

۴-۵- نتیجه گیری :

همانطور که در فصل اول مشاهده شد مطالعات گسترده ای جهت بسط و توسعه فیلتر کالمن در زمینه های مختلف توسط محققان صورت پذیرفته است که از این جمله می توان به کاربرد فیلتر کالمن توسعه یافته در تخمین دینامیکهای سیستمهای غیر خطی اشاره کرد. همچنین کاربرد این فیلتر در آموزش شبکه های عصبی نیز از دیگر قابلیت های این فیلتر بحساب می آید که در فیلتر تخمین وزن فیلتر کالمن دوگانه به آن اشاره شد.

کاربرد شبکه های عصبی در شناسایی سیستمهای غیر خطی که صوت نیز از آن جمله بشمار می آید از موضوعات مهم دیگری بود که در این پایان نامه به آن پرداخته شد. برنامه های مختلفی نیز جهت اجرای شناسایی صوت توسط شبکه عصبی MLP نوشته شد که در پیوست پایان نامه موجود می باشد لذا در واقع در پایان این پروژه به این مطلب می توان دست یافت که شبکه های عصبی در بسیاری از مواردی که کنترل کلاسیک قادر به پاسخگویی به آنها نیست مثل شناسایی سیستمهای غیر خطی، بسیار خوب عمل می کنند و می تواند راهگشای کاربردهای فیلتر کالمن توسعه یافته و یا فیلتر کالمن دوگانه (می تواند در کارهای بعدی مورد توجه دانشجویان قرار گیرد و در این پایان نامه فقط به توضیح مختصر جهت آشنایی با آن اکتفا شده است) در ارائه تخمین از دینامیکهای سیستمهای غیر خطی مثل صنایع موشکی، صنایع هوانوردی و هواپیمایی، صنایع پتروشیمی و پالایشگاهها و بسیاری صنایع دیگر باشد.

همچنین فیلتر کالمن توسعه یافته به همراه روابط ریاضی آن و چگونگی استخراج این روابط مورد مطالعه قرار گرفت و توابع خاص نرم افزاری با استفاده از نرم افزار Matlab 6.5 جهت اجرای روابط بازگشتی این فیلتر نوشته شد و نتایج تخمین سیگنال صوتی تمیز از روی سیگنال نویزی پس از اجرای برنامه بدست آمد لذا حتی در مقایسه عملکردی این فیلتر با فیلتر کالمن معمولی که بر روی سیستمهای خطی عمل می کند می بینیم که این فیلتر در بسیاری موارد مشابه نه تنها با دقت فیلتر کالمن بلکه به مراتب تخمینهای با دقت بهتری نیز ارائه می دهد.

۴-۶- پیشنهاد برای کارهای آینده :

در فصل اول پایان نامه که به مروری بر کارهای انجام شده توسط محققان مختلف بر روی خانواده فیلتر کالمن پرداخت، نحوه معرفی فیلتر کالمن دوگانه توسط محققان بطور کامل تشریح شد و مزایای آن نسبت به فیلترهای قبلی بیان گردید. همچنین مقالات و کتب مرجع که توضیحاتی در مورد این فیلتر و این روش داشته اند در مراجع پایان نامه مشخص گردید. در فصل دوم نیز نحوه بازیابی و روابط این فیلتر بطور کامل تشریح گردید. اما بدلیل آشناییهای اولیه ای که جهت پیاده سازی روش فیلتر کالمن دوگانه لازم بود در این پایان نامه به نوشتن برنامه جهت شبیه سازی و تست این روش پرداخته نشد. لذا این روش می تواند در کارهای دانشجویی آینده مورد توجه دانشجویان قرار گیرد و اطمینان می رود که با توجه به مطالعه سمینار و پایان نامه اینجانب و با توجه به ارائه مراجع کامل اعم از مقالات و کتب مختلف بتوانند به سهولت به نتایج مناسب و قابل توجهی دست یابند.

فهرست منابع فارسی :

[۲۹] سلیم پور،ن، بررسی فیلتر وینر و استفاده از فیلتر کالمن در بهبود سیگنالهای صوتی- دانشگاه

صنعتی خواجه نصیرالدین طوسی، ۱۳۸۳

[۳۴] مقتدایی راد،ا، کاربرد فیلتر کالمن در بهبود سیگنالهای صوتی - دانشگاه آزاد اسلامی واحد

تهران جنوب، ۱۳۸۴

- [1] N. Levinson, "The Wiener RMS error criterion in filter design and prediction," *Journal of mathematics and physics*, vol. 25, pp 261-278, 1946.
- [2] R. E. Kalman. "A new approach to linear filtering and prediction problems," *Transaction of the ASME. Ser. D. Journal of Basic Engineering*. 82.34-45 (1960).
- [3] R. E. Kalman. and R. S. Bucy. *New results in linear filtering and prediction theory*. Journal of Basic Engineering, 83D, pp. 95-108, 1960.
- [4] R. E. Kalman. "Contributions to the theory of optimal control," *Boletin de la Sociedad Matematica Mexicana*, vol. 5, pp. 102-119, Apr 1960.
- [5] T. Nishimura, "On the a priori information in sequential estimation problems," *IEEE Trans. Automatic Control*, vol. AC-11, pp. 197-204, Apr. 1966.
- [6] C. F. Price, "An analysis of the divergence problem in the Kalman filter," *IEEE Trans. Automatic control*, vol. AC-13, pp. 699-702, Dec. 1968.
- [7] Lennart Ljung. "Asymptotic behavior of the extended Kalman filter as a parameter estimator for linear systems". *IEEE Trans. Automatic Control*, AC-24(1), pp. 36-50, 1979.
- [8] K. K. Paliwal and A. Basu. "A speech enhancement method based on Kalman filtering". *International Conference on Acoustics, and Signal Processing*, ICASSP, pp 177-180, IEEE. 1987.
- [9] J. Lim. *Speech Enhancement*. Prentice-Hall, 1983. Collection of reprints.
- [10] J. Lim and A. Oppenheim. "Enhancement and bandwidth compression of noisy speech". *Proceedings of the IEEE*, 67(12): 1586-604, 1979.

- [11] R. McAulay and M. Malpass. "Speech enhancement using a soft-decision noise suppression filter". *IEEE Trans. Acoustics Speech and Signal Processing*, 28:137-145, Apr 1980.
- [12] H. G. Hirsch. "Estimation of noise spectrum and its application to SNR-estimation and speech enhancement". *Technical Report TR-93-012, International Computer Science Institute*, 1993.
- [13] S. Tamura. "An analysis of a noise reduction neural network". *ICASSP 87*, pp 2001-2004, 1987.
- [14] P. Moakes and S. Beet, "Radial basis function networks for noise reduction of speech", *4th International Conf. on Artificial neural networks*, pp 7-12 IEE, 1995.
- [15] T. Le and J. Mason. "Artificial neural networks for nonlinear time-domain filtering of speech", *IEE Proceeding – Vis. Image signal Process*, Vol 143, pp 149-154, June 1996.
- [16] M. dahl and I. Claesson. "A neural network trained microphone array system for noise reduction", *IEEE Neural networks for signal processing VI*, pp 311-319, 1996.
- [17] E.A. Wan and A.T. Nelson, "Dual kalman filtering methods for nonlinear prediction, estimation, and smoothing," in *Advances in Neural Information Processing Systems 9*, Cambridge, MA: MIT Press, 1997.
- [18] E.A. Wan and A.T. Nelson, "Neural Dual Extended kalman filtering: Applications in speech enhancement and monaural blind signal separation," in *Proceeding of IEEE Workshop on Neural Networks for Signal Processing VII*, pp 466-475, 1997.
- [19] www.cslu.cse.ogi.edu/nsl.
- [20] Eric A. Wan and Alex T. Nelson, "A two-observation kalman framework for maximum-likelihood modeling of noisy time series," *Proceedings of international Joint Conference on Neural Networks*, IEEE/INNS, May 1998.

- [21] Alex T. Nelson, "Nonlinear Estimation and Modeling of Noisy Time-Series by Dual kalman filter Methods," *PhD thesis, Oregon Graduate institute of science and technology*, 2000.
- [22] Eric A. Wan and Alex T. Nelson, "Removal of noise from speech using the dual EKF algorithm," *International Conference on Acoustics, Speech, and Signal Processing, ICASSP*, volume 1, pp 381-384. IEEE, 1998.
- [23] S.J. Julier , J.K. Uhlmann and H. Durrant-Whyte, "A new approach for filtering nonlinear systems," *Proceedings of American Control Conference*, pp 1628-1632, 1997.
- [24] S.J. Julier , J.K. Uhlmann , "A general methode for approximation nonlinear transformations of probabality distributions," *Technical report, RRG, Department of engineering Science. University of oxford*, Nov 1996.
- [25] S.J. Julier and J.K. Uhlmann, "A new extension of the kalman filter to nonlinear systems," *Proceedings of AeroSense: The 11th International Symposium on Aerospace/Defence Sensing, simulation and Controls*, 1997.
- [26] E.A. Wan, R. van der Merwe, "The Unscented kalman filter for nonlinear estimation," *Proceeding of Symposium 2000 on adaptive Systems for Signal Processing, Communication and Control (AS-SPCC)*, IEEE, canada, October 2000
- [27] R. van der Merwe, J.F.G. de Freitas,D Doucet, and E.A. Wan, "The Unscented particle filter," *Technical report CUED/F-INFENG/TR 380 Cambridge engineering university department*, Augut 2000
- [28] E.A. Wan, R. van der Merwe, and A.T. Nelson, "Dual estimation and the unscented transformation," *Neural Information Processing systems 12*, Cambridge, MA:MIT Press, pp 667-672, 2000

- [30] E.W. Kaman and J.K. Su, "Introduction to Optimal Estimation", *Springer, Prentice hall* 1999.
- [31] L.Ljung, "System Identification", Springer 1999.
- [32] Peter Stoica, Torsten Soderstrom, "System Identification", PRENTICE.
- [33] E.A. Wan, A.T. Nelson and Van der Merwe, "kalman filtering and neural networks," *Published by John Wiley and sons*, pp 123-166, 221-263, 2001
- [35] L. Ljung and T.Soderstorm, "Theory and Practice of Recursive Identification," Cambridge, MA: MIT Press, 1983.
- [36] Rafic A.Goubran, "A Perceptual Kalman Filtering-Based Approach For Speech Enhancement", 1998

فهرست نام ها

	الف
۱۴	اپنهايم، اى.ب.
۲۰	آلمن، ج.ك.
	پ
۱۷	پاليوال، ا.ك
	ت
۱۸	تامورا، س.اى.
	ج
۲۰	جوليور، س.ج.
	س
۲۳	سليم پور، نيره
	ك
۱۵	كالمن، ر.ا.
	ل
۱۶	لونگ، ل.ا.
۱۴	لوينسون، ان.
۱۷	ليم، ج.ى.
	م
۲۱-۲۰	مورو، و.د.
۱۷	مالپاس، ا.م.
	ن
۱۹	نلسون، ا.ل.
	و
۱۴	ويتز، ر.ب.
۱۹	ون، ا.اى.
	ه
۱۷	هيرچ، ج.جى.

پیوست شماره ۱

برنامه های نوشته شده با نرم افزار

MATLAB 6.5

```

<<<<<<%Neural network system Identification>>>>>>

***** %Network Initialize*****
clc;
x=wavread('clean Input',39000);
v=sqrt(5e-6)*randn(39000,1);
n=sqrt(0.005)*randn(39000,1);
y=x+n;
T=x-v; % x_k=F(x_k-1,W)+v_k So (x_k-v_k)=Network Target==x_k
A0=randn(5,1);
wavwrite(y,8000,'noisy Output');
W1=randn(5,5);
W2=randn(1,5);
B1=randn(5,1);
B2=0.1;
trparms=[0.05 39000 1e-12 0.01];

***** %Network Training*****
batchbp2(W1,W2,B1,B2,A0,T,trparms);

```



```

-----%Hidden layer weights and Bias update (with momentum)-----
dW1=eta*dW1_old+(1-eta)*alpha*Delta1*A0(NI:4+NI:);
W1 =W1 +dW1;
dW1_old=dW1;
dB1=eta*dB1_old+(1-eta)*alpha*Delta1;
B1=B1+dB1;
dB1_old=dB1;
A0(5+NI,:)=A2;
end
fprintf('iteration # %i Error = %f\r',NI,P) % Print on-line
inform

-----%update for next iteration-----
if SSE<stop_crit
break,end
end
-----End of training-----
end
P_vector=P_vector(1:N);
c = fix(clock);
fprintf('\n\nNetwork training ended at %2i.%2i.%2i\n',c(4),c(5),c(6))

-----%Plot Network Error and Network Output-----
figure(1)
subplot(2,1,1);plot(P_vector)
subplot(2,1,2); plot(A3)

-----%Compare Network Target and Network Output-----
figure(2)
plot(T,'r.-');
hold on
plot(A3)

figure(3)
subplot(2,1,1)
specgram(T,8000,window)
subplot(2,1,2)
specgram(A3,8000,window)

-----%Listening Network Target and Network Output-----
wavwrite(A3,8000,'learning clean');

```

```

***** %Extended kalman filter*****
x=wavread('clean Input',39000);
v=sqrt(5e-6)*randn(39000,1);
n=sqrt(0.005)*randn(39000,1);
y=x+n;
xhat=randn(5,1);
A0=xhat;
pn=(10^15)*randn(5);
Sigma2_v=5e-6;
Sigma2_n=0.005;
B=[1;0;0;0;0];
C=B;

-----%Extended Kalman Loop-----
clc;
for k=1:39000
    xhatn=batchbp3(W1,W2,B1,B2,A0,trparms);
    A0(6,:)=xhatn(1,:);
    for L=1:5
        D=(xhatn(2,:)-xhatn(1,:))/(A0(7-L,:)-A0(6-L,:));
        D1(L,:)=D;
    end
    A1=fliplr(D1');
    A=[A1;eye(4,4) zeros(4,1)];
    K=pn*C'*(1/(C*pn*C'+Sigma2_n));
    xhat=xhatn(1:5,:)+K*(y(k,:)-C*xhatn(1:5,:));
    p=(eye(5)-K*C)*pn;
    pn=A*p*A'+B*Sigma2_v*B;
    A0=xhat;
    xhat1(k,:)=xhat(1,:);
end

-----%Compare Estimated Voice With Clean and Noisy Voice-----
figure(4)
subplot(3,1,1);plot(x)
subplot(3,1,2); plot(y)
subplot(3,1,3); plot(xhat1)

-----%Compare Estimated Voice With Clean Voice-----
figure(5)
plot(x,'r--')
hold on
plot(xhat1,'b-')

figure(6)
subplot(2,1,1)
specgram(x,8000,window)
subplot(2,1,2)
specgram(xhat1,8000,window)

-----%Listening Estimated Voice and Clean Voice-----
wavwrite(xhat1,8000,'estim clean');

```


بهبود سیگنالهای صوتی با استفاده از فیلتر کالمن عصبی

پروفسور علی خاکی صدیق
استاد راهنما

امیر مقتدایی راد
کارشناسی ارشد مهندسی برق - (کنترل)

دانشگاه آزاد اسلامی - واحد تهران جنوب - دانشکده تحصیلات تکمیلی

چکیده:

موضوع بهبود گفتار با نیاز به افزایش کیفیت عملکرد سیستمهای ارتباطی صوتی در محیط های نویزی، مطرح گردید. رنج عملکردی وسیعی برای سیستمهای تشخیص گفتار جهت بهبود ارتباط از راه دور در هوانوردی، صنایع نظامی، گفتگوهای راه دور و محیط های سلولی وجود دارد. همچنین تثبیت کیفیت گفتار صوتی می تواند در کاهش خستگی شنونده در محیط های نویزی یا پر فشار مهم باشد. هدف ما نیز در این مقاله بهبود کیفیت قابل ملاحظه گفتار یا افزایش قابلیت فهم آن توسط فیلتر کالمن توسعه یافته می باشد. بنابراین با توضیحات ارائه شده در بالا می توان فهمید که جهت بازیابی یک سیگنال صوتی به یک تخمین خوب نیاز است اما موضوع مهم این است که فیلتر کالمن خطی قادر به ارائه تخمین از مدل های سیستم های غیر خطی نمی باشد حال آنکه اکثر سیستم های واقعی که سیستم های صوتی نیز از این جمله می باشند ماهیت غیر خطی دارند. لذا در این مقاله بر آن شدیم تا با مطالعه بر روی فیلتر کالمن توسعه یافته و به کمک شبکه های عصبی بتوانیم موضوع مدلسازی و تخمین سیگنال صوتی را با فرض غیر خطی بودن آن بررسی نماییم.

Abstract:

The subject of speech enhancement is introduced with requirement to increase of speech communication systems operation quality in noisy environment. A broad operation range exist for recognizing systems to enhance communication in Aviation, military industrial, telecommunications and cellular environment. Also confirmation of voice speech can be important in decrease of listener tired in noisy or high stress environment. Our object in this paper is increasing speech quality or speech enhancement with extended kalman filter.

So with above explanations can find that speech enhancement require to all right estimation but important subject is that the linear kalman filter is disable in giving estimation of nonlinear systems model whereas almost of real systems as if voice systems that is suchlike have nonlinear nature. So we decide to study on extended kalman filter and with help of neural networks can check modeling and estimation of voice speech with nonlinearity nature assumption.

کلید واژه: سیگنالهای صوتی - فیلتر کالمن توسعه یافته (EKF) - شبکه عصبی MLP

مقدمه:

از سال ۱۹۶۰ که فیلتر کالمن معرفی گردید همواره محققان درصدد بودند تا بتوانند از این فیلتر در جهت تخمین سیگنالهای صوتی نویزی شده استفاده کنند اما بدلیل ماهیت غیرخطی بودن صوت و همچنین عدم قابلیت فیلتر کالمن خطی در تخمین دینامیکهای سیستمهای غیرخطی، قادر به این امر نبودند تا اینکه در اواسط سال ۱۹۷۵ فیلتری با نام فیلتر کالمن توسعه یافته معرفی گردید که قابلیت ارائه تخمینهای بهینه از حالتیهای سیستمهای غیر خطی [۱-۴] را با استفاده از مدلسازی سیستم غیرخطی توسط شبکه عصبی [۷ و ۸ و ۱۰] را دارا بود. در نتیجه به تدریج راه برای کار بر روی بازیابی سیگنالهای صوتی با استفاده از فیلتر کالمن توسعه یافته باز شد.

محققان مختلف ایده های متفاوتی را جهت روشهای تخمین غیر خطی سیستمهای مختلف مطرح نمودند که هر یک دارای معایب و مزایای مربوط به خود است [۵-۶] اما در مورد بازیابی سیگنالهای صوتی که دارای ماهیت غیر خطی نیز می باشند بسیار کم سخن به میان آمده است.

لذا در این مقاله سعی شده تا با ارائه یک الگوریتم ساده شناسایی سیستم غیر خطی صوت، ابتدا سیگنال صوتی توسط شبکه عصبی شناسایی شود و سپس توسط الگوریتم تخمین فیلتر کالمن توسعه یافته (EKF) سیگنال صوتی نویزی شده بازیابی شده است. همچنین در انتها سه سیگنال تمیز، نویزی شده و بازیابی شده توسط نرم افزار MATLAB 6.5 رسم و با هم مقایسه گردیده است.

۱- شناسایی سیگنال صوت:

در بحث شناسایی سیستمهای غیر خطی که صوت نیز از آن جمله بشمار می رود بدلیل متغیر بودن دینامیک های سیستم غیر خطی در هر لحظه از زمان و غیر قابل پیش بینی بودن آن بطور دقیق، لذا راههای قبل قابل ارائه و بحث نمی باشد. در نتیجه تنها راه مناسب جهت اینگونه مدلسازی ها کمک گرفتن از شبکه های عصبی است.

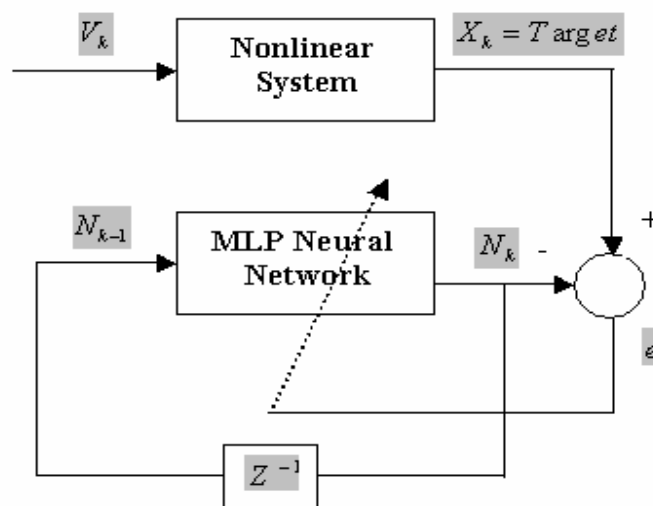
شبکه های عصبی و عصبی-فازی بدلیل قابلیت یادگیری تنها ابزارهای قابل استفاده در شناسایی و پیش بینی سیستمهای غیر خطی می باشند. این شبکه ها می توانند با موازی قرار گرفتن با سیستم مورد مدلسازی و یادگیری هوشمند رفتار سیستم پس از مدتی دقیقاً رفتاری مشابه سیستم از خود نشان دهند تا جایی که میتوان آنها را جایگزین سیستم مدل شده نمود. این توانایی فوق العاده به ما امکان می دهد تا با بهره گیری از شبکه های عصبی بسیاری از دینامیک های غیر خطی و پیچیده سیستمهای واقعی را با دقت بالا مدل کنیم.

۲-۱- شناسایی NAR:

در این روش شناسایی سیستم فرض بر این است که فقط خروجی سیستم معلوم و شناخته شده است در نتیجه ورودی سیستم برای ما ناشناس است و ما فقط به خروجی سیستم دسترسی داریم بنابراین خروجی سیستم به عنوان Target شبکه در نظر گرفته می شود و فیدبکهایی از خروجی به عنوان ورودی سیستم در نظر گرفته می شوند و سپس شبکه شروع به دیدن آموزش می نماید که نمایش بلوکی این روش بصورت زیر است:

$$y(t) = f(y(t-1), y(t-2), \dots, y(t-n_y)) \quad (1-2-1)$$

بلوک دیاگرام شناسایی سیستم با استفاده از این روش و تلفیق آن با شبکه عصبی بصورت زیر است:



شکل ۱-۲-۱- حلقه شناسایی سیستم: پس از N بار تکرار و همگرا شدن خطا، $N_k = X_k$ خواهد شد

در نتیجه ما برای شناسایی سیستم یک شبکه عصبی^۱ MLP را با سیستم موازی کرده و ورودی سیستم را بعنوان ورودی به شبکه اعمال کرده و خروجی سیستم به عنوان سیگنال هدف^۲ برای شبکه تعریف شده است. سپس با استفاده از روش پس انتشار خطای بازگشتی دسته ای^۳ (EBP) اقدام به آموزش شبکه عصبی شده است که در هر بار تکرار خروجی شبکه با مقدار Target مقایسه شده و خطا بعنوان آیتم اصلاحی به شبکه عصبی اعمال شده است. بدلیل اینکه طول بردار X_k که یک سیگنال صوتی می باشد ۳۹۰۰۰ بوده است لذا نیاز به حداقل تکرار ۳۹۰۰۰ برای شبکه عصبی مطرح شده است.

^۱ Multi Layer Perceptron

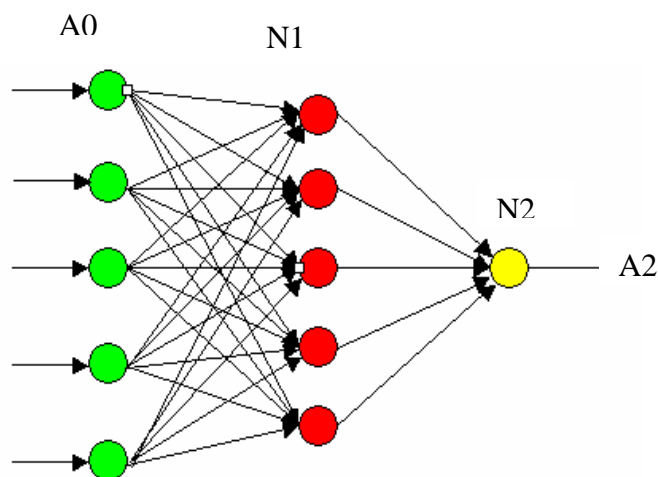
^۲ Target

^۳ Error Back Propagation

۱-۳-۳- ساختار شبکه عصبی بکارگیری شده:

بدلیل اینکه در تشکیل ماتریس A که در برنامه فیلتر کالمن توسعه یافته استفاده می شود و بعد به آن اشاره خواهد شد نیاز به بدست آوردن $\frac{\partial f(x_k, w)}{\partial x_k}$ بود که در واقع همان خروجی شبکه عصبی ما بود. ما نیاز به بدست آوردن یک ماتریس A حداقل 2×2 داشتیم که بعد ماتریس A را تعداد ورودیهای شبکه تشکیل می داد و در نتیجه ما ۵ ورودی برای شبکه انتخاب نمودیم و بعد ماتریس A نیز 5×5 قرار گرفت.

بنابراین شماییک شبکه عصبی $f(.)$ بعنوان شبکه feedforward با ۵ ورودی ($M = 5$)، ۵ لایه مخفی و یک خروجی که بعنوان مدل غیرخطی درجه ۵ ($M = 5$) تعریف شود بصورت زیر است:

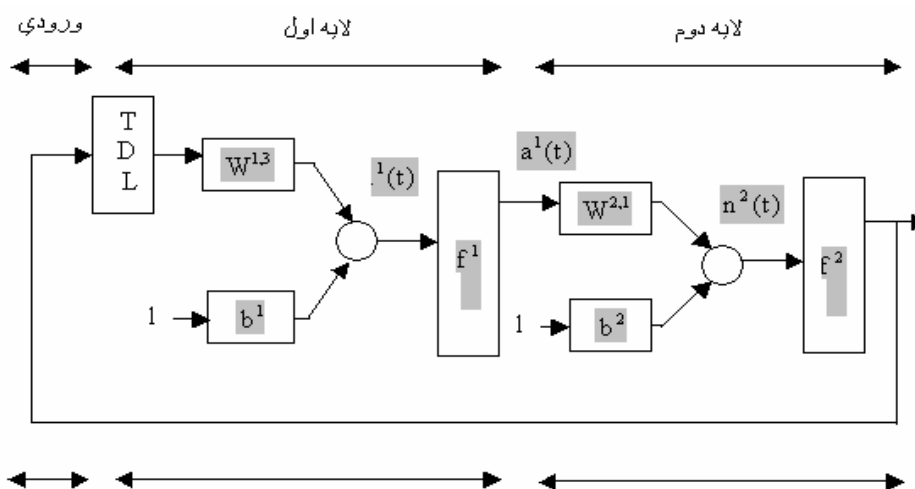


شکل ۱-۳-۱- شبکه عصبی MLP با ۵ ورودی-۵ لایه مخفی و ۱ خروجی

در مورد نحوه آموزش شبکه عصبی نیز ذکر این نکته لازم است که شبکه عصبی MLP با روش پس انتشار خطا آموزش دیده است. نحوه آموزش نیز بدین صورت می باشد که پس از محاسبه خروجی لایه اول و به دنبال آن لایه دوم شبکه، خروجی شبکه با سیگنال Target مقایسه شده و خطای محاسبه شده به شبکه بازخورد داده می شود. بازخورد خطا نیز در بروز آوری بردارهای وزن و بایاس شبکه عصبی نمودار می گردد و سپس شبکه چرخه جدید را با در نظر گرفتن خروجی زمان قبل به عنوان ورودی جدید آغاز می کند.

همچنین فاکتور فراموشی برای شبکه بصورت انتخابی تعریف شده است که در صورت بکارگیری آن توسط کاربر معادلات بروز آوری بردارهای وزن و بایاس متفاوت می گردد که این موضوع فرآیند آموزش را تغییر نمی دهد و فقط معادلات بروز آوری مذکور اندکی تفاوت می کنند.

همچنین نمایش بلوک دیاگرام شبکه عصبی بکار رفته با استفاده از این روش نیز بصورت زیر است:



شکل ۱-۳-۲- نمایش بلوک دیاگرام شبکه عصبی آموزش دیده با روش EBP

۲- تخمین EKF سیگنال صوتی تمیز:

پس از طی مراحل بالا که منجر به مدلسازی و هموارسازی راه جهت مشخص شدن دینامیکهای سیستم شد نوبت به تخمین مدل غیرخطی می رسد. اما این بار بخاطر اینکه با یک مدل غیر خطی روبرو هستیم ناگزیر بایستی بجای فیلتر کالمن از فیلتر کالمن توسعه یافته (EKF) استفاده کنیم. فیلتر کالمن توسعه یافته (EKF) یک روش تقریبی برای مدل‌های غیر خطی است که مدل غیر خطی را با عنوان مدل خطی متغیر با زمان تخمین می زند. ذکر این نکته حائز اهمیت است که یک ویژگی تغییر ناپذیر فیلتر کالمن نیلز به معلوم بودن دینامیک های سیستم است. بعبارت دیگر جهت تخمین سیگنال با فیلتر کالمن از هر نوع (معمولی، توسعه یافته، توسعه یافته دوگانه) تخمین مدل یک امر ضروری است. بنابراین در این بخش به نمایش معادلات فیلتر کالمن توسعه یافته می پردازیم:

۱- مقداردهی اولیه:

$$\hat{x}(0) = E(x_0) \quad (1-2)$$

$$P_0 = E[(x_0 - \hat{x}_0)(x_0 - \hat{x}_0)^T]$$

همانطور که از معادلات بالا مشخص است ابتدا بردار ورودی اولیه و واریانس اولیه نویز مقداردهی میشوند

۲- معادلات بروز کننده زمان:

در ادامه معادلات بروز کننده زمان معرفی شده اند که بردار A_k مهمترین جز تشکیل دهنده آنهاست و با مشتق گیری از تابع F که همان شبکه عصبی MLP می باشد بدست می آید. نحوه محاسبه آن نیز با تشکیل

ماتریس ژاکوبین با درایه هایی که صورت آنها تفاضل دو خروجی فعلی و قبلی و مخرج آنها تفاضل دو ورودی فعلی و قبلی می باشد و تعداد این درایه ها نیز وابسته به تعداد نورونها در لایه ورودی شبکه می باشد که در این مقاله با توجه به ۵ نورون در لایه ورودی ابعاد ماتریس A 5×5 می باشد. ذکر این نکته نیز الزامی است که ماتریس A_k با افزایش K از ۱ تا ۳۹۰۰۰ تغییر می کند و همانند ماتریس A در فیلتر کالمن ثابت نیست.

$$A_k = \left. \frac{\partial F(X, W)}{\partial X} \right|_{x=\hat{x}_k} \quad (2-2)$$

$$A_k = \begin{bmatrix} \frac{\partial f(\hat{x}_k, w)}{\partial x} \\ \left[\begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & \ddots & 0 & \ddots \\ 0 & 1 & 0 & 0 \end{array} \right] \end{bmatrix}$$

$$\begin{aligned} \hat{x}_k^- &= F(\hat{x}_{k-1}, w) \\ P_k^- &= AP_{k-1}A^T + B\sigma_v^2B^T \end{aligned} \quad (3-2)$$

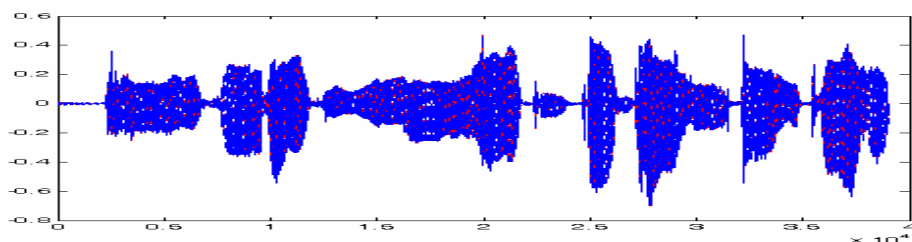
۳- معادلات بروز کننده مشاهده :

$$\begin{aligned} K &= P_k^- C^T (CP_k^- C^T + \sigma_n^2)^{-1} \\ \hat{x}_k &= \hat{x}_k^- + K(y_k - C\hat{x}_k^-) \\ P_k &= (I - KC).P_k^- \end{aligned} \quad (4-2)$$

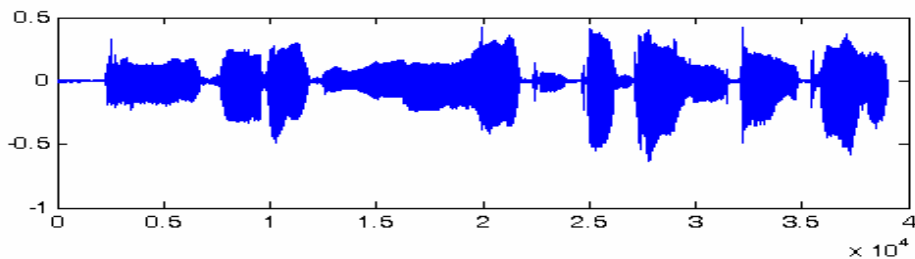
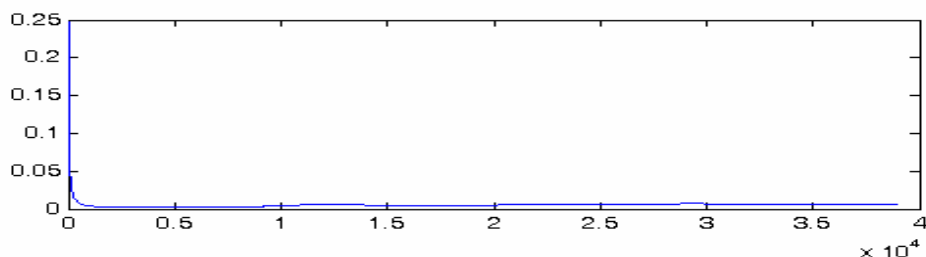
در معادلات بروز کننده مشاهده نیز همانطور که مشاهده می شود گین K و تخمین سیگنال و واریانس نويز بروز آوری شده اند.

۳- شبیه سازیهای انجام شده :

در ابتدا به نمایش خروجی شبکه عصبی و سیگنال تمیز که شبکه روی آن آموزش دیده و شناسایی را انجام داده است می پردازیم:



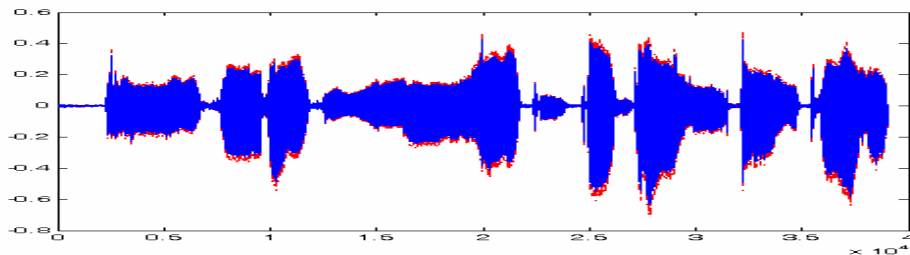
شکل ۳-۱- مقایسه خروجی شبکه (نمودار آبی) با سیگنال هدف (نمودار قرمز)



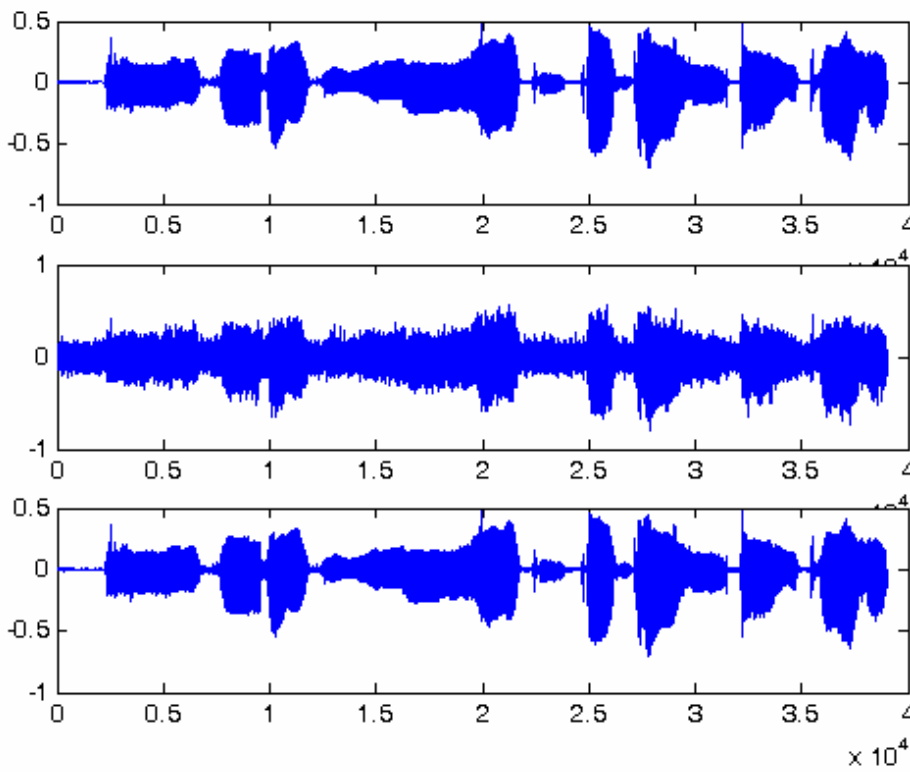
شکل ۳-۲- نمایش خطای آموزش به همراه خروجی شبکه

همانطور که از شکل‌های بالا مشخص است مدل‌سازی صوت غیر خطی بخوبی انجام گرفته و سیگنال خروجی شبکه عصبی و سیگنال هدف که همان سیگنال صوتی تمیز می باشند همپوشانی بسیار مناسبی را دارا می باشند.

همچنین در ادامه نمایش سیگنال صوت تخمینی که با استفاده از مدل‌سازی صوت و سپس تخمین صوت تمیز از روی صوت نویزی شده با استفاده از فیلتر کالمن توسعه یافته انجام گرفته است می پردازیم:



شکل ۳-۳- مقایسه سیگنال صوت تمیز (نمودار قرمز) و سیگنال صوت بازیافت شده (نمودار آبی)



شکل ۳-۴- مقایسه سیگنال تمیز (شکل اول) سیگنال نویزی (شکل دوم) و سیگنال تخمینی (شکل سوم)

نتیجه گیری:

در پایان می توان به این مطلب دست یافت که شبکه های عصبی در بسیاری از مواردی که کنترل کلاسیک قادر به پاسخگویی به آنها نیست مثل شناسایی سیستمهای غیر خطی، بسیار خوب عمل می کنند و می تواند راهگشای کاربردهای فیلتر کالمن توسعه یافته در ارائه تخمین از دینامیکهای سیستمهای غیر خطی مانند صوت باشد.

مراجع:

- [1] E.A. Wan and A.T. Nelson, "Dual kalman filtering methods for nonlinear prediction, estimation, and smoothing," in *Advances in Neural Information Processing Systems 9*, Cambridge, MA: MIT Press, 1997.
- [2] E.A. Wan and A.T. Nelson, "Neural Dual Extended kalman filtering: Applications in speech enhancement and monaural blind signal separation," in *Proceeding of IEEE Workshop on Neural Networks for Signal Processing VII*, pp 466-475, 1997.
- [3] www.cslu.cse.ogi.edu/nse1.
- [4] Eric A. Wan and Alex T. Nelson, "A two-observation kalman framework for maximum-likelihood modeling of noisy time series," *Proceedings of international Joint Conference on Neural Networks, IEEE/INNS*, May 1998.
- [5] Alex T. Nelson, "Nonlinear Estimation and Modeling of Noisy Time-Series by Dual kalman filter Methods," *PhD thesis, Oregon Graduate institute of science and technology*, 2000.
- [6] Eric A. Wan and Alex T. Nelson, "Removal of noise from speech using the dual EKF algorithm," *International Conference on Acoustics, Speech, and Signal Processing, ICASSP*, volume 1, pp 381-384. IEEE, 1998.
- [7] L.Ljung, "System Identification", Springer 1999.
- [8] Peter Stoica, Torsten Soderstrom, "System Identification", PRENTICE.
- [9] E.A. Wan, A.T. Nelson and Van der Merwe, "kalman filtering and neural networks," *Published by John Wiley and sons*, pp 123-166, 221-263, 2001
- [10] L. Ljung and T.Soderstorm, "Theory and Practice of Recursive Identification," Cambridge, MA: MIT Press, 1983.
- [11] Rafic A.Goubran, "A Perceptual Kalman Filtering-Based Approach For Speech Enhancement", 1998

Abstract:

In the first chapter thesis review on the recent 50 years research that have be done with scientist and researchers and try to study on complement of kalman filter family. In the 2th chapter it introduce kalman filter, extended kalman filter and dual kalman filter and explain theirs equations and relation between them and also the method of obtaining equations.

In the 3th chapter the subject of linear and nonlinear system identification is introduced and all kind of linear and nonlinear system identification methods are studied. Also neural network application in nonlinear system identification and how training and network architecture are explained. In the 4th chapter the simulink results that done with Matlab 6.5 software are shown and compare them with each other. Also this chapter include the result of thesis and future work.

The list of programs are shown in appendix A.



ISLAMIC AZAD UNIVERSITY

TEHRAN SOUTH BRANCH

FACULTY OF GRADUATE STUDIES

DEPARTMENT OF ELECTRICAL ENGINEERING

" M.SC " THESIS

ON ELECTRICAL

SUBJECT:

**APPLICATION OF NEURAL KALMAN FILTER IN SPEECH
ENHANCEMENT**

THESIS ADVISOR:

A. KHAKI SEDIGH PH.D

BY:

A. MOGHTADAEI RAD

SEPTEMBER,2006