جهت تهیه فایل WORD این مقاله به سایت DaneshResan.com مراجعه نمایید و عنوان مقاله را جستجو کنید
بیش از ۲ میلیون مقاله فارسی در این سایت موجود میباشد

Archvie
دانشگاه صنعتی شریف
دانشکده مهندسی کامپیوتر

جزیره کیش، خلیج فارس، ایران ۱۹ الی ۲۱ اسفند ۱۳۸۶

انجمن کامپیوتر ایران
Computer Society of Iran

مقاله کامل

اثر هموارکردن نویز چندی‌کردن برروی کدگذاری ADPCM غیرخطی سیگنال صحبت

قاسم علی‌پور
دانشکده مهندسی برق و کامپیوتر
دانشگاه شهید بهشتی، تهران، ایران
mh_savoji@yahoo.com

محمد حسن ساوجی
دانشکده مهندسی برق و کامپیوتر
دانشگاه شهید بهشتی، تهران، ایران
mh_savoji@yahoo.com

چکیده: با وجود توانایی بالای فیلترهای ولترا[1] در توصیف بسیاری از ویژگی‌های غیرخطی سیگنال صحبت و کارایی خوب آنها در بسیاری از کاربردهای پردازش صحبت توجه اندکی به استفاده از این فیلترها در کدگذاری صحبت شده است. این موضوع هدف اصلی ما از پژوهشی است که پیش از این برخی از نتایج آن مربوط به استفاده از فیلترهای وفقی درجه‌دو[2] در چارچوب تکنیک ADPCM[3] با پیش‌بینی وفقی پیشرو[4] (APF) و پسرو[5] (APB) گزارش شده است. در اینجا در ادامه بررسی‌های انجام شده برروی کدگذار ADPCM با پیش‌بینی پسرو روشی برای کاهش اثر نویز چندی کردن برروی کیفیت نهایی سیگنال بازسازی‌شده با هموار کردن این نویز پیشنهاد می‌شود. نتایج شبیه‌سازی‌ها نشان می‌دهند که هموار کردن نویز براساس دو معیار SNR و SNR تکه‌ای[6] (SEGSNR) باعث بهبود خوبی در کیفیت نهایی سیگنال بازسازی‌شده در هر دو حالت خطی و غیرخطی با طول متغیر می‌شود. این بهبود در عوض افزایش بسیار کمی در پیچیدگی الگوریتم ناشی از استفاده از فیلتر هموارساز به دست می‌آید.

کلمات کلیدی: کدگذاری صحبت، فیلترهای ولترا، نویز چندی کردن، هموار کردن، الگوریتم کمترین میانگین مربع‌ها[7]، الگوریتم کمترین مربع‌های بازگشتی[8]

۱. مقدمه

امروزه به دلیل محدودیت پهنای باند بیشتر سیستم‌های ارسال سیگنال و ظرفیت سیستم‌های ذخیره اطلاعات استفاده از کدکننده‌های صحبت به یک نیاز پایه‌ای تبدیل شده است. در بیشتر روش‌های کدگذاری افزونگی[1]های موجود در سیگنال صحبت تا حد ممکن حذف شده و بخش‌های غیرزائد صحبت در یک نرخ بیت پایین‌تر به روشی کد می‌شود که از نظر شنیداری قابل قبول باشد. یک دسته مهم از این روش‌ها کدکننده‌های شکل موج[2] است که بر مبنای حفظ شکل موج کلی سیگنال صحبت عمل می‌کنند. این کدکننده‌ها می‌توانند هر شکل موجی را در باند صوتی قبول کنند و صرفاً مختص صحبت نیستند. از آنجا که در این کدکننده‌ها کدگذاری به صورت نمونه به نمونه انجام می‌شود عملکرد آن‌ها همانند چندی‌کننده‌ها[3] توسط نسبت سیگنال به نویز (SNR) اندازه‌گیری می‌شود. در تکنیک کدگذاری ADPCM حذف این افزونگی‌ها با انجام یک پیش‌بینی از مقدار هر نمونه سیگنال و کم‌کردن این پیش‌بینی از مقدار واقعی و چندی کردن سیگنال باقیمانده و یا خطای پیش‌بینی در یک نرخ بیت پایین‌تر انجام می‌گیرد [3,5].

فیلترهای خطی به دلیل داشتن ویژگی‌های شناخته‌شده، سادگی ذاتی و کارایی قابل قبولی که در بسیاری از کاربردها دارند به وفور در پردازش سیگنال صحبت به کار رفته‌اند. ولی به دلیل وجود ویژگی‌های غیرخطی سیگنال صحبت مدل‌های غیرخطی در عوض داشتن پیچیدگی‌های بیشتر می‌توانند همبستگی‌های غیرخطی موجود در سیگنال صحبت را به شکل مناسبی استخراج کنند. یکی از قدیمی‌ترین و شناخته‌شده‌ترین دسته‌های سیستم‌های غیرخطی فیلترهای ولترا برپایه بسط سری‌های ولترا[4] است [4,7]. علاوه بر توانایی بالای اینگونه فیلترها در مدل‌سازی بسیاری از فرایندهای واقعی مهم‌ترین مزیت آنها خطی بودن خروجی فیلتر نسبت به ضرایب توصیف‌کننده آن است. این واقعیت علاوه بر امکان محاسبه سریع‌تر ضرایب و پیاده‌سازی آسان‌تر فیلتر گسترش بسیاری از الگوریتم‌ها از حالت شناخته شده فیلتر خطی به حالت فیلتر ولترا را فراهم می‌سازد. از طرفی علاوه بر ناپایداری ذاتی عمده‌ترین مشکل این فیلترها افزایش نمایی تعداد ضرایب توصیف‌کننده فیلتر با درجه و نیز طول حافظه فیلتر غیرخطی است؛ هرچند که در سال‌های اخیر الگوریتم‌های بسیاری برای پیاده‌سازی مؤثر و پایدار اینگونه فیلترها پیشنهاد شده است [8,9].

---

[1] Volterra
[2] quadratic
[3] Adaptive Pulse Code Modulation
[4] Adaptive Prediction – Forward
[5] Adaptive Prediction – Backward
[6] Segmental SNR
[7] Least Mean Square
[8] Recursive Least Squares

جهت تهیه فایل WORD این مقاله به سایت DaneshResan.com مراجعه نمایید و عنوان مقاله را جستجو کنید
بیش از ۲ میلیون مقاله فارسی در این سایت موجود میباشد
Archvie of SID

فیلرهای ولترا به شکل وسیعی در بسیاری از کاربردهای پردازش سیگنال از جمله نویززدایی، جبران‌سازی کانال‌های مخابراتی، مدل‌سازی فرایندهای طبیعی و پردازش سیگنال تصویر مورد استفاده قرار گرفته‌اند [4,7]. با این وجود توجه بسیار ناچیزی به استفاده از این فیلترها در کدگذاری سیگنال صحبت شده است [10,11]. ما در [۱] و [۲] به ترتیب به این مقوله در قالب ساختارهای پیش‌بینی‌کننده پیشرو (APF) و پیشرو (APB) پرداخته‌ایم. در اینجا هدف هموار کردن[۵] نویز واردشده در جریان چندی‌کردن به منظور بهبود نتایج به دست آمده با استفاده از فیلتر ولترا در قالب تکنیک کدگذاری ADPCM با پیش‌بینی وفقی پسرو است. برای پرهیز از پیچیدگی محاسباتی بالا باز بررسی‌های خود را به حالت فیلتر ولترای درجه‌دو محدود می‌کنیم. بر این اساس در آغاز و در بخش بعدی تکنیک کدگذاری ADPCM معرفی می‌شود. سپس کدگذاری ADPCM سیگنال صحبت با استفاده از فیلترهای درجه‌دوی ولترا در شمای APB در بخش ۳ توسعه داده می‌شود. در این بخش روشی برای کاهش اثر نویز چندی‌کردن با هموار کردن این نویز بر اساس ساختار ارائه شده در [۲] برای استفاده بهینه از پیش‌بینی‌کننده غیرخطی پیشنهاد می‌شود. بخش ۴ هم شامل برخی نتایج به دست‌آمده از شبیه‌سازی الگوریتم‌ها است. در پایان هم یک جمع‌بندی و نتیجه‌گیری ارئه می‌شود.

برای آزمایش الگوریتم‌ها و ساختارهای معرفی‌شده در این مقاله همواره چهار سیگنال صحبت برگرفته از پایگاه داده[۶] TIMIT [13] شامل دو نمونه اداشده توسط زن و دو نمونه اداشده توسط مرد هرکدام شامل یک جمله کامل در زبان انگلیسی به‌کار رفته‌اند. سیگنال‌ها در فرکانس ۸kHz نمونه‌برداری شده‌اند و دامنه هر کدام به صورت غیرفشرده با ۸ بیت چندی شده است. همچنین پیش از انجام آزمایش‌ها دامنه این سیگنال‌ها به مقدار ۱ نرمالیزه شده اند.

۲. تکنیک کدگذاری ADPCM

دلیل اصلی بهبود کارایی کدکننده‌های تفاضلی کاهش محدوده پویا[۷]ی سیگنال ورودی به چندی‌کننده است. این کاهش با حذف افزونگی‌های کوتاه‌مدت[۸] موجود در سیگنال اصلی صحبت با انجام یک برآورد از سیگنال اصلی و چندی کردن سیگنال تفاضلی یا خطای پیش‌بینی زیر انجام می‌گیرد [5,3]:

$$d(n)=s(n)-\tilde{s}(n) \qquad (۱)$$

$\tilde{s}(n)$ یک پیش‌بینی از سیگنال گفتار در لحظه n است و بهره پیش‌بینی متناسب با این پیش‌بینی به صورت

$$g=\sigma_s^2/\sigma_d^2 \qquad (۲)$$

تعریف می‌شود که σ_s^2 و σ_d^2 به ترتیب واریانس سیگنال‌های گفتار و باقی‌مانده می‌باشند. از آنجا که نویز چندی‌کردن متناسب با اندازه پله چندی کردن است در یک نرخ بیت ثابت یک سیگنال با محدوده پویای کوچک‌تر می‌تواند با دقت بیشتر و خطای چندی‌کردن کمتری کد شود.

در بیشتر موارد برآورد با یک پیش‌بینی خطی به صورت یک ترکیب خطی از چند نمونه قبلی سیگنال انجام می‌گیرد. ولی اگر فیلتر پیش‌بینی را به شکل یک فیلتر غیرخطی شامل یک بخش درجه دو با طول حافظه N_2 و با ضرایب $\{h_2(i,j)\}$ به همراه یک بخش خطی با طول حافظه N_1 و با ضرایب $\{h_1(i)\}$ در نظر بگیریم مقدار پیش‌بینی‌شده $\tilde{s}(n)$ (خروجی پیش‌بینی‌کننده) در هر زمان به صورت زیر خواهد بود [4,7]:

$$\tilde{s}(n)=\sum_{i=1}^{N_1}h_1(i)s(n-i)+\sum_{k=1}^{N_2}\sum_{j=k}^{N_2}h_2(k,j)s(n-k)s(n-j) \qquad (۳)$$

برای سادگی ماتریس ضرایب بخش درجه‌دو به صورت بالامثلثی درنظر گرفته شده است.

به دلیل ناایستان بودن سیگنال گفتار در یک کدکننده ADPCM پیش‌بینی کننده و چندی‌کننده به صورت وفقی ویژگی‌های آماری سیگنال صحبت را دنبال می‌کنند. پیش‌بینی وفقی می‌تواند پیشرو یا پسرو باشد. در پیش‌بینی وفقی پیشرو (APF) برآورد پارامترها با توجه به نمونه‌های سیگنال صحبت انجام می‌گیرد که در دکدکننده موجود نیست. پس به منظور بازسازی سیگنال صحبت در دکدکننده لازم است که ضرایب فیلتر پیش‌بینی به همراه سیگنال خطای چندی‌شده به سمت دکدکننده فرستاده شود. ولی در پیش‌بینی وفقی پسرو (APB) پارامترها به صورت متوالی و با توجه به نمونه‌های قبلی سیگنال چندی‌شده‌ای به روز می‌شوند که در دکدکننده نیز موجود هستند. پس می‌توان پارامترهای پیش‌بینی را به صورت محلی در دکدکننده برآورد کرد تا نیازی به ارسال اطلاعات اضافی نباشد [5,3].

در اینجا یک چندی‌کننده بی‌حافظه[۹] و لحظه‌ای[۱۰] را در نظر می‌گیریم که در آن تبدیل در هر زمان وابسته به نمونه‌های قبلی و بعدی نیست. در این حالت عبارت چندی‌کننده وفقی می‌تواند بیان‌کننده یک چندی‌کننده با ساختار ثابت با اندازه پله متغیر باشد. مسئله اصلی تطبیق اندازه پله $\Delta(n)$ به واریانس متغیر ورودی است. در حالت ایده‌آل اندازه پله متغیر با زمان $\Delta(n)$ پیوسته مقدار بهینه‌ای از Δ (Δ_{opt}) را دنبال می‌کند که یک چندی‌کننده ثابت با ورودی ایستان یا ایستان محلی با واریانس معلوم s_x^2 به کار می‌برد. می‌توان دید که Δ_{opt} از طریق یک پارامتر f (که تنها وابسته به تابع چگالی احتمال (pdf) ورودی و تعداد بیت بر نمونه است) متناسب با s_x (انحراف معیار[۱۱] سیگنال ورودی) است. مقدار این پارامتر را می‌توان برای یک pdf و تعداد بیت بر نمونه خاص به صورت عددی به دست آورد[3] .

پس کارکرد یک چندی‌کننده وفقی را می‌توان به شکل زیر بیان کرد:

$$\Delta(n)=\phi\hat{\sigma}_x(n) \qquad f \text{ ثابت} \qquad (۴\text{-الف})$$

$\hat{\sigma}_x(n)$ برآوردی از واریانس سیگنال x در زمان n است که برای ورودی ناایستان متغیر با زمان است. مطابق رابطه بالا مسئله اصلی در

جهت تهیه فایل WORD این مقاله به سایت DaneshResan.com مراجعه نمایید و عنوان مقاله را جستجو کنید
بیش از ۲ میلیون مقاله فارسی در این سایت موجود میباشد
Archvie of SID

چندی کردن وفقی برآورد وفقی و پیوسته $\hat{\sigma}_x$ است. با وزن‌دار کردن نمایی نمونه‌ها می‌توان یک برآوردکننده واریانس به صورت زیر به دست آورد:

$$s_x^2(n) = a\, s_x^2(n-1) + (1-a)x^2(n-1) \qquad \text{(۴-ب)}$$

که در آن $0 < \alpha < 1$ طول حافظه را کنترل می‌کند. هراندازه سیگنال ورودی به چندی‌کننده ناایستان‌تر باشد مقدار α باید به مقدار صفر نزدیک‌تر شود. دو نوع چندی کردن وفقی را می‌توان در نظر گرفت: چندی کردن وفقی با برآورد پیشرو[۱۲] (AQF) و چندی کردن وفقی با برآورد پسرو[۱۳] (AQB) که به ترتیب همراه با APF و APB مورد استفاده قرار می‌گیرند. درAQF برآورد σ_x برپایه نمونه‌های چندی‌نشده و در AQB برآورد σ_x بر پایه نمونه‌های خروجی چندی‌کننده انجام می‌گیرد.

نتایج به دست آمده بااستفاده از شمای APF و APB پیش از این به ترتیب در مراجع [۱] و [۲] گزارش شده است. آزمایش‌ها نشان داده بودند که با وجود توانایی بالا در پیش‌بینی سیگنال صحبت مهم‌ترین مشکل فیلترهای غیرخطی ولترا ناپایداری فیلتر بازسازی بود. ساختارهایی برای رفع این مشکل به همراه الگوریتم‌هایی برای بهره‌گیری بهینه از فیلتر پیش‌بینی غیرخطی پیشنهاد شد. در اینجا و در ادامه نتایج گزارش‌شده در [۲] برای شمای APB روشی برای کاهش اثر نویز چندی کردن با هموار کردن این نویز پیشنهاد می‌شود.

۳. پیش‌بینی وفقی - پسرو

معایب اصلی APF تأخیر در کدگذاری و نیاز به بافرکردن داده و نیز ظرفیت کانال اضافی مورد نیاز برای ارسال اطلاعات جانبی می‌باشد. اگر ضرایب پیش‌بینی‌کننده بهینه را بتوان برپایه داده چندی‌شده و ارسال‌شده برآورد کرد می‌توان آنها را در دکدکننده بازسازی کرده تا دیگر نیازی به ارسال این ضرایب نباشد. بر این اساس کدکردن با پیش‌بینی وفقی پسرو برپایه انجام عملیات یکسان پیش‌بینی در کدکننده و دکدکننده معرفی شده است. از آنجا که نویز چندی کردن یک فرایند تصادفی باندعریض[۱۴] است ولی خود سیگنال صحبت یک سیگنال باند باریک[۱۵] است برای کاهش سطح نویز در سیگنال بازسازی‌شده یک فیلتر پایین گذر برای حذف نویز خارج از باند[۱۶] پیشنهاد می‌شود. این فیلتر برای نمونه می‌تواند مطابق با شکل ۱ به همراه الگوریتم کدگذاری پیاده شود.

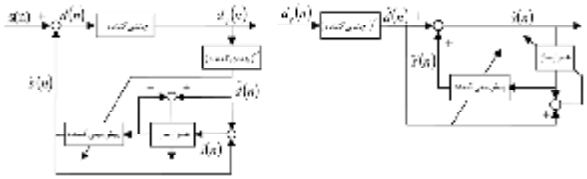
شناخته‌شده‌ترین الگوریتم‌های وفقی متوالی عبارتند از الگوریتم‌های کمترین میانگین مربع (LMS) و کمترین مربع‌های بازگشتی (RLS) [6]. الگوریتم LMS در مقایسه با الگوریتم RLS از نظر محاسباتی مؤثرتر و از نظر عددی پایدارتر است. ولی علاوه بر وابستگی رفتار همگرایی و ویژگی‌های حالت دایم آن به ویژگی‌های سیگنال ورودی نرخ همگرایی آن نیز پایین‌تر است. هرچند در سال‌های اخیر توجه زیادی به توسعه پیاده‌سازی‌های مؤثر و پایدار فیلترهای ولترای وفقی برپایه الگوریتم RLS شده است ولی با این وجود سادگی و پایداری بیشتر الگوریتم LMS منجر به کاربرد بیشتر آن در مبحث فیلترهای وفقی ولترا شده است. در هر حال برتری نسبی هر کدام از این دو الگوریتم وابسته به مشخصات سیگنال ورودی همچون پهنای باند و SNR و نیز زمان همگرایی و پیچیدگی محاسباتی مورد قبول در کاربرد مورد نظر دارد [12].

یک نتیجه مهم خطی بودن خروجی فیلتر ولترا نسبت به ضرایب این است که می‌توان نشان داد که همه الگوریتم‌های وفقی معتبر در حالت خطی را در اصل می‌توان به حالت فیلتر ولترا هم گسترش داد [4,7]. پیش از معرفی فیلترهای وفقی ولترا بر پایه الگوریتم‌های متوالی LMS و RLS بردارهای ورودی و ضرایب در لحظه n را به ترتیب به صورت زیر تعریف می‌کنیم:

$$\begin{aligned}\mathbf{H}_n = [&h_1(1:n),\ h_1(2:n), \mathbf{L}, h_1(N_1:n),\\ &h_2(1,1:n), h_2(1,2:n), \mathbf{L}, h_2(1,N_2:n),\\ &h_2(2,2:n), h_2(2,3:n), \mathbf{L}, h_2(N_2,N_2:n)]^T\end{aligned} \qquad \text{(۵-الف)}$$

$$\begin{aligned}\mathbf{X}_n = [&\hat{s}(n-1), \hat{s}(n-2), \mathbf{L}, \hat{s}(n-N_1), \hat{s}(n-1)\hat{s}(n-1)\\ &, \hat{s}(n-1)\hat{s}(n-2), \mathbf{L}, \hat{s}(n-1)\hat{s}(n-N_2),\\ &\hat{s}(n-2)\hat{s}(n-2), \mathbf{L}, \hat{s}(n-N_2)\hat{s}(n-N_2)]^T\end{aligned} \qquad \text{(۵-ب)}$$

$\hat{s}(n)$ نمونه بازسازی‌شده سیگنال در لحظه n است که هم در کدکننده و هم در دکدکننده قابل دسترسی است.



الف) دکدکننده ب) کدکننده

شکل (۱): ساختار کلی مورداستفاده در کدگذار ADPCM-APB همراه با فیلتر هموارساز برای الف) کدکننده و ب) دکدکننده

۱.۳. الگوریتم LMS

الگوریتم LMS یک عضو مهم از خانواده الگوریتم‌های وفقی بر پایه گرادیان[۱۷] است. می‌توان این الگوریتم را به صورت یک حل تقریبی از یک مسئله بهینه‌سازی آماری در نظر گرفت که هدف آن رسیدن به کمینه میانگین مربع خطا[۱۸] (mmse) است [6]. در تقریب LMS نرمالیزه (NLMS) در هر لحظه ضرایب فیلتر پیش‌بینی با استفاده از الگوریتم تندترین شیب[۱۹] و با کمینه کردن مقدار $d^2(n)$ به صورت زیر به روز می‌شود [4,7]:

$$\mathbf{H}_n = \mathbf{H}_{n-1} + \frac{2\mu \mathbf{X}_n \hat{d}(n)}{\tilde{\sigma}_n^2} \qquad \text{(۶-الف)}$$

Archvie of SID

$0 \le m << 1$ پارامتر همگرایی برای کنترل طول حافظه فیلتر پیش‌بینی کننده و سرعت همگرایی است. همچنین

$$\tilde{\sigma}_n^2 = \alpha \hat{s}^2(n) + (1-\alpha)\tilde{\sigma}_{n-1}^2 \qquad (۶-ب)$$

برآوردی از توان سیگنال ورودی به فیلتر در لحظه n است که در آن a عامل تعیین‌کننده حافظه در برآورد مذکور است.

۲.۳. الگوریتم RLS

الگوریتم RLS را می‌توان یک حالت خاص از فیلتر کالمن[۲۰] در نظر گرفت. این الگوریتم حل دقیق یک مسئله بهینه‌سازی جبری است که هدف آن کمینه کردن یک تابع هزینه به صورت زیر است [6]:

$$J(n) = \sum_{k=1}^{n} l^{n-k} (s(k) - \mathbf{H}_n^T \mathbf{X}_k)^2 \qquad (۷)$$

$0 << l \le 1$ را عامل وزن‌گذاری یا فراموشی گویند. با استفاده از قضیه عکس کردن ماتریس[۲۱] می‌توان حل بازگشتی زیر را برای الگوریتم RLS متداول[۲۲] (CRLS) در پیش‌بینی پسرو سیگنال صحبت به کار برد [4,7]:

$$d(n) = s(n) - \mathbf{X}_n^T(n)\mathbf{H}_{n-1} \qquad (۸-الف)$$

$$\mathbf{H}_n = \mathbf{H}_{n-1} + \mathbf{K}_n \hat{d}(n) \qquad (۸-ب)$$

$$\mathbf{K}_n = \frac{\mathbf{P}_{n-1}\mathbf{X}_n}{l + \mathbf{X}_n^T \mathbf{P}_{n-1}\mathbf{X}_n} \qquad (۸-پ)$$

$$\mathbf{P}_n = \frac{1}{l}\mathbf{P}_{n-1} - \frac{1}{l}\mathbf{K}_n \mathbf{X}_n^T \mathbf{P}_{n-1} \qquad (۸-ت)$$

معمولاً الگوریتم بافرض $\mathbf{P}_0 = \frac{1}{d}\mathbf{I}$ و با یک مقدار کوچک مثبت از d آغاز می‌شود که I یک ماتریس یکه[۲۳] است.

۴. نتایج

نتایج به دست آمده از شمای APB با استفاده از الگوریتم‌های LMS و RLS در حالتی که فیلتر هموارساز استفاده نشده است پیش از این و در [۲] آورده شده است. در آنجا بیان شد که به دلیل بزرگ‌تر بودن پنجره مشاهده الگوریتم RLS در عوض پیچیدگی بیشتر منجر به بهره پیش‌بینی بالاتری می‌شود. همچنین بیان شد که با استفاده از فیلتر خطی تنها پس از بازسازی سیگنال صحبت برای چندین مقدار N_1 افزایش N_1 از مقدار ۱۰ به بعد تأثیر بسیار اندکی بر روی کیفیت سیگنال بازسازی‌شده دارد. بنابراین با ثابت نگه داشتن N_1 در مقدار ۱۰ به بررسی اثر افزودن بخش درجه‌دو با N_2های مختلف بر روی کارایی کلی کدگذار پرداختیم.

دیده شد که استفاده از فیلتر غیرخطی در تکنیک ADPCM به خودی خود باعث بهبود در کیفیت سیگنال دکدشده نمی‌شود؛ هر چند که بر روی برخی از بخش‌های سیگنال ورودی منجر به کاهش سطح نویز در سیگنال بازسازی‌شده می‌شود. بنابراین ساختاری را امتحان کردیم که بر اساس آن برای هر قالب از سیگنال ورودی به طول T پس از آزمودن L فیلتر غیرخطی با N_1=۱۰ و L مقدار N_2 (L-۱ تا ۰=N_2) و بازسازی سیگنال فیلتری را که منجر به سیگنال بازسازی‌شده با کمترین سطح نویز شود را برای پیش‌بینی انتخاب کنیم. در این ساختار برای هر قالب نیاز به $\log_2 L$ بیت اضافی جهت مشخص کردن طول حافظه فیلتر درجه‌دو داریم که این خود باعث افزایش $\log_2 L/T$ بیت در نرخ بیت نهایی می‌شود. به علاوه قالب‌بندی کردن سیگنال و آزمودن چند فیلتر غیرخطی منجر به یک تأخیر T ثانیه‌ای و افزایش پیچیدگی الگوریتم می‌شود. همچنین باید توجه داشت که برای بازسازی سیگنال در دکدکننده برای هر قالب باید همه فیلترهای درجه‌دو در هر تکرار هم در کدکننده و هم در دکدکننده به روز شوند؛ هرچند که برای هر قالب در نهایت تنها یک فیلتر غیرخطی مورداستفاده قرار می‌گیرد.

این تکنیک (استفاده از فیلتر غیرخطی با طول حافظه متغیر) در اینجا و با استفاده از فیلتر هموارساز بر پایه ساختار شکل ۱ برای هر دو الگوریتم LMS و RLS مورد استفاده قرار گرفت. فیلتر هموارساز یک فیلتر خطی تنها با طول حافظه ۲ انتخاب شد. در الگوریتم LMS فیلتر هموارساز براساس معیار LMS به روز می‌شود و تطبیق این فیلتر در الگوریتم RLS مطابق با معیار RLS انجام می‌گیرد.

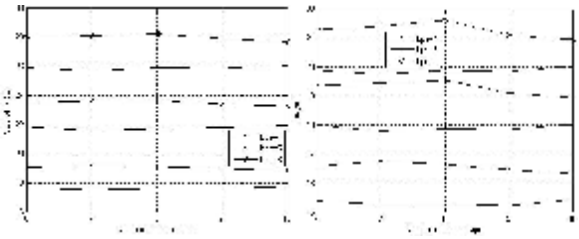
در آغاز با استفاده از یک فیلتر پیش‌بینی خطی تنها به بررسی اثر افزودن فیلتر هموارساز با طول ۲ برروی کیفیت سیگنال بازسازی‌شده از نقطه نظر دو معیار SNR و SEGSNR پرداختیم. براین اساس با استفاده از الگوریتم‌های LMS و RLS به ترتیب نتایج شکل‌های ۲ و ۳ برای چند مقدار طول حافظه فیلتر خطی پیش‌بینی و سه مقدار بیت بر نمونه در مقایسه با حالتی که از فیلتر هموارساز استفاده نشده است به دست می‌آید. دیده می‌شود که استفاده از فیلتر هموارساز در همه حالت‌ها باعث بهبود در کیفیت سیگنال بازسازی‌شده می‌شود. در اینجا مقدار SEGSNR به صورت میانگین مقدار SNR بر روی قالب‌هایی از سیگنال به طول ۱۰ms (۸۰ نمونه) تعریف شده است.

مقدار بهبود در میانگین SNR و SEGSNR سیگنال بازسازی‌شده با استفاده از ساختار غیرخطی در مقایسه با ساختار خطی با N_1=۱۰ در جدول‌های ۱ و ۲ نشان داده شده است. این نتایج با T=۱۰ms و دو مقدار ۸ و۴=L و نیز سه مقدار ۶ و ۵ و۴=bps به دست آمده‌اند. باز مقدار SEGSNR به صورت میانگین مقدار SNR بر روی قالب‌هایی از سیگنال به طول ۱۰ms (۸۰ نمونه) تعریف شده است.

دیده می‌شود که استفاده از این ساختار در مقایسه با فیلتر خطی تنها باعث افزایش نسبتاً خوبی در کیفیت سیگنال بازسازی‌شده می‌شود. این افزایش برای معیار SEGSNR که یک معیار ملموس‌تر و نزدیک‌تر به کیفیت سیگنال صحبت است برجسته‌تر است. همچنین با مقایسه نتایج جدول‌های ۱ و ۲ با نتایج [علی۸۶] مشاهده می‌شود که استفاده از فیلتر هموارساز باعث بهبود خوبی در کیفیت نهایی سیگنال بازسازی‌شده می‌شود. این بهبود در عوض افزودن یک فیلتر خطی تنها

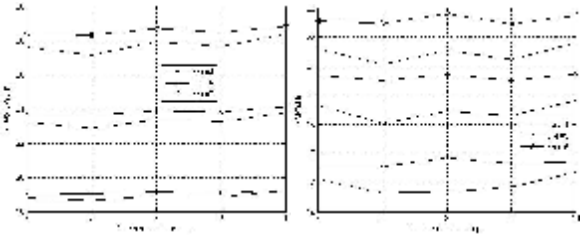
Archvie of SID

با طول ۲ به دست می‌آید این در حالی است که افزایش بیشتر طول فیلتر پیش‌بینی خطی منجر به بهبود چندانی در کیفیت نهایی سیگنال بازسازی‌شده نمی‌شود.



(ب) (الف)

شکل (۲): اثر افزودن فیلتر هموارساز برروی کیفیت سیگنال بازسازی‌شده در کدگذار با پیش‌بینی پسرو خطی با الگوریتم LMS از نقطه نظر معیار الف) SNR و ب) SEGSNR



(ب) (الف)

شکل (۳): اثر افزودن فیلتر هموارساز برروی کیفیت سیگنال بازسازی‌شده در کدگذار با پیش‌بینی پسرو خطی با الگوریتم RLS از نقطه نظر معیار الف) SNR و ب) SEGSNR

جدول (۱): افزایش در میانگین الف) SNR و ب) SEGSNR سیگنال بازسازی‌شده با افزودن بخش درجه‌دو با الگوریتم LMS

الف) SNR

SNR(dB)	bps=۴	bps=۵	bps=۶
L=۴	۱/۱۱۳۸۴	۱/۰۴۳۸	۰/۹۶۱۸
L=۸	۱/۲۸۷۸	۱/۱۰۲۰	۱/۰۱۷۸

ب) SEGSNR

SEGSNR (dB)	bps=۴	bps=۵	bps=۶
L=۴	۱/۸۰۱۳۴	۱/۹۹۲۴	۱/۱۴۳۷
L=۸	۱/۹۹۹۷	۲/۲۰۹۴	۲/۳۴۱۳

جدول (۲): افزایش در میانگین الف) SNR و ب) SEGSNR سیگنال بازسازی‌شده با افزودن بخش درجه‌دو با الگوریتم RLS

الف) SNR

SNR (dB)	bps=۴	bps=۵	bps=۶
L=۴	۱/۲۳۶۲	۱/۱۶۱۳۰	۱/۳۲۱۸
L=۸	۲/۱۰۷۲	۱/۷۲۹۸	۱/۹۱۸۱

ب) SEGSNR

SEGSNR (dB)	bps=۴	bps=۵	bps=۶
L=۴	۱/۰۱۲۲	۱/۱۱۳۷۸	۰/۹۶۸۰
L=۸	۱/۸۳۰۲	۲/۳۰۳۱	۲/۲۳۱۲

۵. نتیجه‌گیری

پیش از این در چهارچوب تکنیک کدگذاری ADPCM با پیش‌بینی پسرو الگوریتمی غیرخطی با استفاده از فیلترهای ولترا با طول متغیر برای استفاده بهینه از پیش‌بینی غیرخطی پیشنهاد شده بود. در اینجا براساس این الگوریتم روشی برای هموار کردن نویز چندی کردن و از آنجا افزایش کیفیت سیگنال بازسازی‌شده پیشنهاد شد. ساختار هموارکننده شامل یک فیلتر خطی وفقی با طول حافظه ۲ بر روی سیگنال بازسازی‌شده است. نتایج شبیه‌سازی‌ها نشان دادند که افزودن این فیلتر هموارساز باعث بهبود قابل توجهی در کیفیت سیگنال بازسازی‌شده نسبت به حالت خطی تنها و نیز نسبت به نتایج گزارش‌شده برای فیلتر غیرخطی با طول حافظه متغیر می‌شود. این خود به دلیل این واقعیت است که نویز چندی کردن یک سیگنال باند عریض است در حالی که سیگنال صحبت یک سیگنال باند باریک است و فیلتر هموارساز در واقع یک فیلتر پایین‌گذر است که باعث حذف نویز خارج از باند و از آنجا افزایش کیفیت سیگنال بازسازی شده می‌شود. این بهبود درعوض افزایش بسیار کمی در پیچیدگی الگوریتم ناشی از استفاده از فیلتر هموارساز به دست آمد.

منابع

[۱] ساوجی، محمد حسن و علی‌پور، قاسم، "**کدگذاری گفتار بااستفاده از پیش‌بینی غیرخطی برپایه بسط سری‌های ولترا**" نشریه مهندسی برق و مهندسی کامپیوتر ایران، سال ۱۳، شماره ۱، تهران، بهار ۱۳۸۶.

[۲] علی‌پور، قاسم و ساوجی، محمد حسن "**بهره‌گیری بهینه از پیش‌بینی‌کننده غیرخطی پسرو در کدگذاری سیگنال صحبت**" پانزدهمین کنفرانس مهندسی برق ایران، تهران، اردیبهشت ۱۳۸۶.

[3] Jayant, N. S. and Noll, P., Digital Coding of Waveforms: Principles and Applications to Speech and Video, Prentice-Hall, Englewood Cliffs, New Jersey, 1984.

[4] Mathews, V. J. and Sicuranza, G. L., Polynomial Signal Processing, John Wiley, New York, 2000.

[5] Deller, J. R. et al., Discrete-Time Processing of Speech Signals, Macmillan Publication Company, New York, 1993.

[6] Haykin, S., Adaptive Filter Theory, Prentice-Hall, New Jersey, 1991.

[7] Sicuranza, G. L., "Quadratic Filters for Signal Processing", IEEE Proc., Vol. 80, pp. 1263–12813, 1992.

[8] Bernardini, R., "A Fast Algorithm for General Volterra Filtering", IEEE Trans. on Communication, Vol. 48, No. 11, November 2000.

[9] Reed, M. J., and Hawksford, M. O. J., "Efficient Implementation of the Volterra Filter", IEE Proc. Image Signal Process., Vol. 147, No. 2, April 2000.

[10] Mumolo, E., and Francescato, D., "Adaptive Predictive Coding of Speech by Means of Volterra Predictors", IEEE Proc.

Archvie of SID

Winter Workshop on Non-linear Digital Signal Processing, pp. 2.1.4.1-2.1.4.4, 1993.
[11] Thyssen, J., and Hansen, S., "Non-linear Short Term Prediction", Proc. ICASSP, pp. 1813-188, 1994.
[12] Wei, P. C., et al, "Comparative Tracking Performance of the LMS and RLS Algorithms for Chirped Narrowband Signal Recovery", IEEE Trans. on Signal Process., Vol. 130, No. 7, July 2002.
[13] "DARPA TIMIT — Acoustic-Phonetic ontinuous Speech Corpus," National Institute of Standards and Technology document NISTIR 4930, 1993.

---

[1] redundancy
[2] waveform coder
[3] quantizer
[4] Volterra
[5] smoothing
[6] Data Base
[7] dynamic range
[8] short-term redundancy
[9] memoryless
[10] instantaneous
[11] standard deviation
[12] adaptive quantization with forward estimation
[13] adaptive quantization with backward estimation
[14] Broad-Band
[15] Narrow-Band
[16] Out-of-Band
[17] gradient based adaptive algorithm
[18] minimum mean squared error
[19] steepest descent
[20] Kalman
[21] matrix inversion lemma
[22] Conventional RLS
[23] singular matrix