

## الگوریتم فیلتر تطبیقی زیرباند توسعه یافته در شبکه توزیع شده انتشاری

محمد شمس اسفندآبادی<sup>۱</sup>، دانشیار؛ فاطمه باستی<sup>۲</sup>، دانشجوی کارشناسی ارشد؛ محمدسعید شفیعی<sup>۳</sup>، دانشجوی دکتری

۱- دانشکده مهندسی برق - دانشگاه تربیت دبیر شهید رجائی - تهران - ایران - u.ac.ir.msahms@sr

۲- دانشکده مهندسی برق - دانشگاه تربیت دبیر شهید رجائی - تهران - ایران - f.basti@sru.ac.ir

۳- دانشکده مهندسی برق - دانشگاه تربیت دبیر شهید رجائی - تهران - ایران - ms.shafiee@sru.ac.ir

**چکیده:** الگوریتم‌های تطبیقی نقش مهمی در جهت بهبود عملکرد شبکه‌های توزیع شده انتشاری دارند. الگوریتم‌های خانواده زیرباند انتشاری در مقایسه با الگوریتم حداقل میانگین مربعات نرمالیزه شده انتشاری (DNLMS) دارای سرعت همگرایی بالاتری در مواجهه با ورودی رنگی هستند. این مقاله به ارائه مسأله تخمین توزیع شده در یک شبکه انتشاری برپایه الگوریتم زیرباند توسعه یافته (IMSAF) پرداخته و الگوریتم فیلتر تطبیقی زیرباند توسعه یافته انتشاری (DMSAF) را معرفی می‌کند. در الگوریتم پیشنهادی برخلاف الگوریتم زیرباند نرمالیزه شده انتشاری (DNSAF)، به جای به کارگیری تک بردار رگرسور ورودی در هر زیرباند، از چند بردار رگرسورهای سیگنال ورودی استفاده می‌شود که این عمل منجر به بهبودی رفتار همگرایی الگوریتم پیشنهادی در مواجهه با ورودی رنگی خواهد شد. در این الگوریتم هرچه تعداد رگرسور ورودی بیشتر گردد، سرعت همگرایی نیز افزایش می‌یابد. عملکرد الگوریتم DIMSAF با الگوریتم‌های DLMS، DAPA، VSS-DAPA و DRLS مورد مقایسه قرار گرفته است. نتایج نشان می‌دهد عملکرد الگوریتم ارائه شده دارای سرعت همگرایی بالاتری در مقایسه با الگوریتم‌های ذکر شده است.

**واژه‌های کلیدی:** فیلتر تطبیقی، شبکه‌های توزیع شده انتشاری، زیرباند نرمالیزه، زیرباند نرمالیزه توسعه یافته.

## Improved Multiband-Structured Subband Adaptive Filter Algorithm in Diffusion Distributed Networks

M. S. Esfand Abadi<sup>1</sup>, Associate Professor; F. Basti<sup>2</sup>, MSc Student; M. S. Shafiee<sup>3</sup>, PhD Student

1- Faculty of Electrical Engineering, Shahid Rajaei Teacher Training University, Tehran, Iran, Email: mshams@sru.ac.ir

2- Faculty of Electrical Engineering, Shahid Rajaei Teacher Training University, Tehran, Iran, Email: f.basti@sru.ac.ir

3- Faculty of Electrical Engineering, Shahid Rajaei Teacher Training University, Tehran, Iran, Email: ms.shafiee@sru.ac.ir

**Abstract:** Adaptive algorithms play an important role in order to improve performance of diffusion distributed network. In comparison of diffusion Normalized least mean square algorithm, family of diffusion subband algorithms have faster convergence rate when the input signal is highly correlated. This paper solves the problem of distributed estimation in the diffusion networks based on improved multiband-structured subband adaptive filter (IMSAF) and diffusion improved multiband structured subband adaptive filter (DMSAF) is established. In proposed algorithm, convergence behavior improved due to using several input projections instead of single vector of input data. In addition, when the projection order is increased, the convergence rate of the proposed algorithm improves. The validity of the DIMSAF in comparison of DLMS, DAPA, VSS-DAPA and DRLS algorithms is demonstrated by several computer simulations. The results show fastest convergence rate.

**Keywords:** Adaptive filter, diffusion distributed networks, normalized subband, improved multiband-structured subband.

تاریخ ارسال مقاله: ۱۳۹۶/۵/۲۹

تاریخ اصلاح مقاله: ۱۳۹۷/۲/۲۹

تاریخ پذیرش مقاله: ۱۳۹۷/۳/۱۹

نام نویسنده مسئول: محمد شمس اسفندآبادی

نشانی نویسنده مسئول: ایران - تهران - لویزان - خیابان شهید شعبانلو - دانشگاه تربیت دبیر شهید رجائی - دانشکده مهندسی برق

## ۱- مقدمه

پرکاربردتر است. اما اگر سیگنال ورودی دارای همبستگی بالا باشد، الگوریتم LMS سرعت پابینی در همگرایی خواهد داشت و حتی ممکن است این الگوریتم در مواجهه با ورودی‌های رنگی، واگرا گردد. الگوریتم فیلتر تطبیقی زیرباند (NSAF) برای بهبود رفتار همگرایی الگوریتم LMS ارائه شده است [۷-۱۲]. در این الگوریتم سیگنال ورودی و نیز پاسخ مطلوب از طریق بانک فیلتر به دسته‌های مجزا از لحاظ فرکانسی تقسیم می‌گردد که باعث همگرایی هر زیرباند به‌طور مجزا شده و بهبود همگرایی کل را در پی خواهد داشت.

برای بهبود سرعت همگرایی الگوریتم NSAF، الگوریتم فیلتر تطبیقی زیرباند نرمالیزه توسعه‌یافته (IMSFAF) معرفی گردید که از الگوریتم استفاده مجدد از داده‌ها، مانند الگوریتم تصویر افاین (TAPA) به‌عنوان جایگزین برای افزایش سرعت همگرایی استفاده می‌کند [۱۳]. در این تحقیق به ارائه مسأله تخمین توزیع‌شده در یک شبکه انتشاری بر اساس الگوریتم زیرباند توسعه‌یافته پرداخته و الگوریتم زیرباند توسعه‌یافته انتشاری را معرفی کرده و در ادامه عملکرد مناسب الگوریتم پیشنهادی و صحت روابط تئوری حاصل‌شده، با شبیه‌سازی‌های مختلف مورد بررسی قرار می‌گیرد.

این تحقیق به‌صورت زیر سازمان‌دهی شده است: در بخش دوم، شبکه توزیع‌شده انتشاری و پارامترهای آن مورد بررسی قرار گرفته است. بررسی روابط الگوریتم IMSAF و ارائه الگوریتم پیشنهادی در شبکه انتشاری در بخش سوم بیان می‌شود. در بخش چهارم پیچیدگی محاسبات الگوریتم بررسی می‌گردد. در بخش آخر با انجام شبیه‌سازی‌های مختلف، عملکرد مناسب الگوریتم پیشنهادی و صحت روابط حاصل‌شده بررسی می‌گردد.

علائم ریاضی زیر در این تحقیق مورد استفاده قرار گرفته‌اند:

$$\begin{aligned} \|\cdot\|^2 & \text{اندازه اقلیدسی یک بردار} \\ (\cdot)^T & \text{ترانهاده یک بردار یا ماتریس} \\ (\cdot)^{-1} & \text{معکوس یک ماتریس یا یک عدد اسکالر} \end{aligned}$$

## ۲- شبکه توزیع‌شده انتشاری

پردازش‌های توزیع‌شده با استخراج داده‌ها از گره‌هایی که در یک منطقه جغرافیایی توزیع شده‌اند، عمل می‌کنند. هر گره در یک شبکه می‌تواند مشاهدات نویزی مربوط به یک پارامتر خاص یا پدیده دلخواه را جمع‌آوری کرده و مطابق با ساختار شبکه، برای رسیدن به تخمینی از پارامتر یا پدیده دلخواه، با همسایگانشان تعامل کند [۴].

در شبکه‌های قدیمی نیاز به یک پردازنده قوی با حجم ارتباطات زیاد بود. این شبکه‌ها علیرغم مصرف انرژی و پهنای باند زیاد در مقایسه با شبکه‌های توزیع‌شده، دقت پایین‌تر و کارایی کمتری داشتند. شبکه‌های توزیع‌شده تطبیقی برای پردازش اطلاعات غیرمرکزی و انجام کارهای بهینه‌سازی بسیار مناسب است. همچنین این شبکه‌ها برای مدل کردن رفتارهای پیچیده که در طبیعت با آنها مواجه می‌شویم کاربرد دارند. شبکه‌های توزیع‌شده تطبیقی شامل مجموعه‌ای از گره‌ها

در سال‌های اخیر، شبکه‌های توزیع‌شده<sup>۱</sup> به‌عنوان روشی برای ارتباطات و کنترل شبکه‌های کنونی و آینده مورد توجه قرار گرفته‌اند. این شبکه‌ها روشی کارآمد برای ارتباطات گره‌ها<sup>۲</sup> با یکدیگر هستند. شبکه‌های توزیع‌شده با توجه به اهمیت صرفه‌جویی در انرژی و پهنای باند، روشی هوشمندانه و بهینه هستند زیرا با کاهش ارتباطات، در انرژی و پهنای باند صرفه‌جویی می‌کنند [۱-۳]. در شبکه‌های متمرکز قدیمی، گره‌های موجود در شبکه، مشاهدات خود را جمع‌آوری کرده و به یک مکان مرکزی برای پردازش می‌فرستند. پردازنده مرکزی، تخمین مورد نیاز را انجام داده و نتیجه را به همه گره‌ها ارسال می‌کند. این نوع از پردازش به یک پردازنده مرکزی قدرتمند و همچنین حجم وسیعی از ارتباطات بین گره‌ها و پردازنده نیاز دارد. در راه‌حل توزیع‌شده، به دلیل متکی بودن گره‌ها به داده‌های محلی<sup>۳</sup> خود و تعامل<sup>۴</sup> با همسایگان، ارتباطات و انرژی مصرف‌شده به‌میزان قابل‌توجهی کاهش پیدا می‌کند [۴].

دو روش برای پیاده‌سازی شبکه‌های توزیع‌شده مطرح می‌شود:

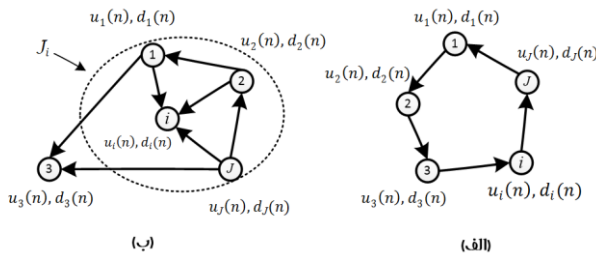
۱- روش افزایشی<sup>۵</sup> ۲- روش انتشاری<sup>۶</sup>

این دو روش از نظر ساختار و نحوه ارتباطات گره‌ها با یکدیگر متفاوت هستند. روش افزایشی، به‌صورت دوره‌ای بین گره‌های همسایه تحقق می‌پذیرد. در این روش اطلاعات در حالت زنجیره‌ای از یک گره به گره مجاور گردش می‌یابند. به این ترتیب یک گره، اطلاعاتی را که خود از محیط دریافت کرده، به‌وسیله ترکیب با اطلاعاتی که از گره قبلی گرفته است، اصلاح کرده و نتیجه را به گره همسایه منتقل می‌کند. در پایان این زنجیره، یک تخمین مناسب از پارامتر مورد نظر به‌دست می‌آید [۲]. در روش انتشاری، پیاده‌سازی به‌صورت همکاری بین گره‌ها بر مبنای همسایگی‌ها صورت می‌گیرد. هر گره در شبکه می‌تواند به‌عنوان یک فیلتر تطبیقی مجزا برای تخمین پارامترهای مطلوب عمل کند. در این روش، گره‌ها در چند دسته همسایگی قرار دارند و هر گره با همه گره‌های واقع در همسایگی خود در ارتباط بوده و اطلاعات را به اشتراک می‌گذارد. به‌عبارت دیگر اطلاعات در میان گره‌های همسایه منتشر می‌شود، سپس این تخمین‌های محلی در همسایگی‌ها با هم آمیخته می‌شوند [۴].

از کاربردهای شبکه‌های توزیع‌شده می‌توان به‌عنوان حسگرهای توزیع‌شده، دیده‌بانی محیط، فضا‌های هوشمند، تمرکز روی هدف، رادیو شناختی<sup>۷</sup> و اتوماسیون صنعتی نام برد [۱، ۳، ۵، ۶]. این شبکه‌ها نسبت به شبکه‌های متمرکز قدیمی دارای بازده بیشتری هستند. بنابراین توسعه روش‌های کارآمد در این شبکه‌ها، می‌تواند مفید باشد. به‌همین منظور ایده استفاده از الگوریتم‌های تطبیقی در این شبکه‌ها مورد توجه قرار گرفته است.

الگوریتم‌های فیلتر تطبیقی متعددی در طول ۴۰ سال گذشته پیشنهاد شده‌اند که در بین آن‌ها الگوریتم حداقل میانگین مربعات (LMS)<sup>۸</sup> به دلیل سادگی و قدرت همگرایی از بقیه مشهورتر و

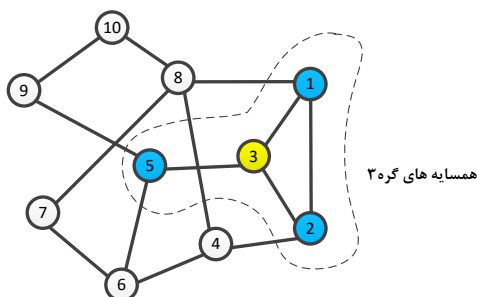
گره‌های شبکه) مرحله به‌روزرسانی باید انجام شود تا وزن جدید به‌دست آید و چرخه بعدی تکرار شود.



شکل ۲: دو مورد از همکاری میان گره‌ها در شبکه توزیع شده: (الف) افزایشی (ب) انتشاری [۱۸].

در روش انتشاری، هر گره با تمام گره‌های همسایه که با ساختار شبکه مشخص شده‌اند، ارتباط دارد. در واقع هر گره در شبکه انتشاری مانند یک فیلتر تطبیقی عمل می‌کند، به طوری که گره‌های مختلف در همسایگی‌هایی قرار می‌گیرند، که اطلاعات هر گره، ابتدا به گره‌های همسایه و سپس به همسایگان مجاور آن گره منتشر می‌شوند. حجم ارتباطات در روش انتشاری بیشتر از روش افزایشی است؛ زیرا گره‌ها با تعداد بیشتری از همسایگان خود در ارتباط بوده و دسترسی به مقادیر بالاتری از اطلاعات دارند [۲۰].

شکل ۳ یک شبکه به هم پیوسته انتشاری را با ۱۰ گره نشان می‌دهد. شبکه به هم پیوسته به شبکه‌ای اطلاق می‌گردد که بین هر دو گره دلخواه حداقل یک مسیر وجود داشته باشد. همسایگان یک گره مفروض مجموعه‌ای از گره‌ها می‌باشد که از طریق یال به این گره وصل هستند، در این بین خود گره مورد نظر نیز در این مجموعه قرار می‌گیرد. برای مثال همسایگان گره ۳ به صورت مجموعه  $J_3 = \{1, 2, 3, 5\}$  نمایش داده می‌شود که درجه این گره چهار است ( $j_p = 4$ ) [۲۱].

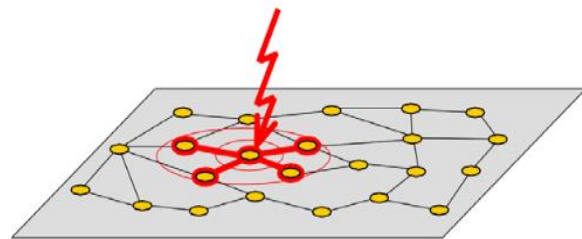


شکل ۳: یک شبکه انتشاری با ۱۰ گره که همسایه‌های گره ۳ با خط چین نمایش داده شده است [۲۱].

فرض بر آن است که همسایگی متقارن است، یعنی اگر گره  $i$  با گره  $l$  همسایه است برعکس این حالت نیز وجود دارد. البته تقارن همسایگی بدین معنی نیست که انتشار اطلاعات دارای تقارن است. بنابراین برای هر گره دلخواه ضریب  $a_{li}$  نسبت داده می‌شود. این ضریب معرف وزنی است که گره  $i$  برای مقیاس‌دهی اطلاعات دریافتی از گره  $l$  استفاده می‌کند. ضرایب وزنی تأثیر به‌سزایی در روابط به‌روزرکردن

با قابلیت پردازش و یادگیری هستند. گره‌ها از طریق ساختار شبکه به یکدیگر متصل می‌شوند و به کمک تعامل با همسایگان‌شان، برای حل مسائل بهینه‌سازی و تخمین توزیع شده به صورت بلادرنگ<sup>۳</sup>، با یکدیگر همکاری می‌کنند [۱۷-۱۴].

یک فیلتر تطبیقی، به صورت بلادرنگ به داده‌های خود و به تغییرات در خواص آماری این داده‌ها پاسخ می‌دهد. به وسیله یک شبکه تطبیقی، یک ساختار به هم پیوسته شامل گره‌های تطبیقی داریم که قادرند به داده‌ها به صورت بلادرنگ پاسخ داده و تغییرات خواص آماری را به صورت مناسبی ردیابی کنند. مطابق شکل ۱ در یک شبکه تطبیقی، هرگاه اطلاعات به یک گره خاص برسد، اطلاعات تأثیر موج‌گونه در شبکه گذاشته، کارایی و رفتار گره‌های دیگر را با توجه به ساختار شبکه تحت تأثیر قرار می‌دهد [۲].



شکل ۱: طرح کلی شبکه توزیع شده متشکل از گره‌های تطبیقی [۴].

هدف از مسأله تخمین توزیع شده، تخمین تطبیقی برخی از پارامترهای مطلوب در روش توزیع شده است. کار تخمین با محاسبات محلی در هر گره و اطلاعات به‌دست‌آمده از گره‌های همسایه صورت می‌گیرد. در مقایسه با تخمین متمرکز، تخمین توزیع شده به گره پردازنده مرکزی قدرتمند نیازی ندارد. بنابراین، این روش نسبت به خرابی گره‌ها و یال‌های ارتباطی مقاوم است. به طور مشخص، هرکدام از گره‌ها در شبکه می‌تواند با همسایگان‌شان برای افزایش دقت تخمین محلی خود همکاری کند. بدیهی است که تأثیر هر تحقق توزیع شده، به مدل‌های همکاری بین گره‌ها بستگی دارد. شکل ۲، دو مورد از مدل‌های همکاری در میان گره‌ها را نشان می‌دهد که در آن  $u_i(n)$  و  $d_i(n)$  سیگنال‌های اندازه‌گیری شده ورودی و مطلوب گره  $i$  در زمان  $n$  می‌باشد و  $J_i$  مجموعه گره‌های همسایه است که با گره  $i$  در ارتباط هستند [۱۸].

در روش افزایشی، اطلاعات در حالت رشته‌ای (زنجیره‌ای) از یک گره به گره هم‌جوار گردش می‌یابند. این مدل از عملیات نیازمند به یک الگوی چرخه‌ای از همکاری در میان گره‌ها است و به کوچک‌ترین حجم ارتباطات و توان نیاز دارد [۴، ۱۹].

الگوریتم افزایشی در برخی از کاربردها دارای محدودیت‌هایی است. مشکل اول مربوط به خرابی گره‌ها و یال‌های ارتباطی بین آنها است. اگر گره و یا یالی در حلقه الگوریتم افزایشی دچار مشکل شود، انتشار اطلاعات به وقفه می‌افتد. مشکل دوم مرتبط با ساختار شبکه در پیدا کردن حلقه‌ای است که تمامی گره‌های شبکه را دربرگیرد. محدودیت دیگر مربوط به تعداد مراحل در هر تکرار است. در هر تکرار،  $J$  (تعداد

در هر لحظه  $n$  استراتژی ATC دارای دو مرحله می‌باشد. در گام اول که مرحله تبادل اطلاعات است، هر گره  $i$  از همسایه‌های خود اطلاعات آن‌ها  $(d_i(n))$  و  $(u_i(n))$  را دریافت کرده و با کمک رابطه اصلاح وزن، وزن قبلی را به یک وزن محلی  $(\Psi_i(n))$  به‌روز می‌نماید. در مرحله دوم که مرحله ترکیب است، هر گره  $i$  مقادیر تخمین محلی  $(\Psi_i(n))$  به‌دست آمده از گره‌های همسایه خود را ترکیب نموده و به وزن جدید تبدیل می‌نماید [۲۲] و [۳۱-۲۹].

خلاصه‌ای از پیاده‌سازی الگوریتم LMS در روش انتشاری با استراتژی ATC در جدول ۲ آمده است.

**جدول ۲: شبه کد الگوریتم LMS در شبکه انتشاری با استراتژی ATC**

مقداردهی اولیه  $w_i(-1) = 0$  برای تمام گره‌ها در هر تکرار  
 ۱. وارد کردن  $(u_i(n))$  و  $(d_i(n))$  جدید برای هر گره  
 $u_i(n) = [u_i(n), \dots, u_i(n - M + 1)]^T, d_i(n) = i = 0, \dots, J$   
 ۲. محاسبه وزن محلی  $\Psi_i(n)$  در هر گره  $i$  با استفاده از داده‌های همسایگی  
 $\Psi_i(n) = w_i(n - 1) + \mu_i \sum_{l \in J_i} c_{li} u_l(n) [d_i(n) - u_l^T(n) w_i(n - 1)]$   
 ۳. ترکیب وزن محلی گره‌های همسایه برای محاسبه اصلاح وزن در هر گره  
 $w_s(n) = \sum_{l \in J_s} a_{ls} \Psi_l(n)$

**۳- الگوریتم DIMSAF**

این بخش به ارائه الگوریتم پیشنهادی و استخراج روابط اصلاح وزن آن خواهد پرداخت. در الگوریتم فیلتر تطبیقی زیرباند، سیگنال ورودی و پاسخ مطلوب با عبور از یک بانک فیلتر به زیرباندهایی تقسیم می‌شوند که از لحاظ فرکانسی از هم جدا می‌باشند. این ویژگی SAF باعث می‌شود تا بتوان خواص هر کدام از سیگنال‌های زیرباند را از هم جدا نمود [۱۲]. این عمل باعث همگرایی هر زیرباند به‌طور مجزا گردیده [۲۵] و بهبود همگرایی کل را در پی خواهد داشت. شکل ۴ ساختار الگوریتم NSAF را نشان می‌دهد.

در این شکل  $H_0, H_1, \dots, H_{N-1}$  بانک فیلتر بوده،  $(u(n))$  و  $(d(n))$  بیان‌گر دنباله‌های اصلی سیگنال ورودی و خروجی مطلوب قبل از عبور از بانک فیلتر و  $(u_s(n))$  و  $(d_s(n))$  بعد از عبور از بانک فیلتر هستند. همچنین باید توجه داشت که نشانه  $N \downarrow$  در شکل، نشان‌دهنده زیرنمونه‌برداری<sup>۱۸</sup> با فاکتور  $N$  (تعداد زیرباندها) است، و سیگنال زیرنمونه‌برداری شده با شاخص  $D$  نمایش داده می‌شود به‌گونه‌ای که  $(d_{s,D}(n))$  بیانگر سیگنال خروجی مطلوب زیرنمونه‌برداری شده است. نشانه  $\uparrow N$ ، عکس این حالت است و سیگنال‌های زیرنمونه‌برداری شده با یکدیگر الحاق<sup>۱۹</sup> می‌شوند. باید توجه داشت در روابط به‌کاررفته، اندیس  $n$  نشان‌دهنده دنباله‌های اصلی و اندیس  $k$  نشان‌دهنده سیگنال زیرنمونه‌برداری شده است؛ به‌صورتی که  $k = n/N$  است [۱۳].

وزن‌ها دارند. این ضرایب وزنی که همگی نامنفی هستند در یک ماتریس  $J \times J$  به نام ماتریس ترکیب‌گر<sup>۱۴</sup> قرار می‌گیرند [۲۲]. برای انتخاب وزن‌های ماتریس‌های ترکیب‌گر روش‌های مختلفی ارائه شده است. در این تحقیق از دو قانون میان‌گیری و Metropolis استفاده شده که در جدول ۱ آمده است. در این جدول  $J_i$  درجه گره  $i$  است.

**جدول ۱: قوانین انتخاب ضرایب ترکیب‌گر  $A = [a_{li}]$**

قانون انتخاب درایه‌های ماتریس	
$a_{li} = \begin{cases} 1/j_i & \text{if } i \neq l \text{ are neighbors or } i = l \\ 0 & \text{otherwise} \end{cases}$	قانون میانگیری انتشاری <sup>۱۵</sup> [۲۳]:
$a_{li} = \begin{cases} 1/\max\{j_i, j_l\} & \text{if } i \neq l \text{ are neighbors} \\ 1 - \sum_{t \in J_i \setminus \{l\}} a_{it} & \text{if } i = l \\ 0 & \text{otherwise} \end{cases}$	قانون Metropolis [۲۴]:

روش انتشاری با استفاده از دو استراتژی تطبیق سپس ترکیب (ATC<sup>۱۶</sup>) و ترکیب سپس تطبیق (CTA<sup>۱۷</sup>) تحقق پیدا می‌کند. در هر لحظه از زمان به روش ATC، ابتدا هر گره، در مرحله تبادل اطلاعات، از داده‌های دریافت‌شده گره‌های همسایه یک تخمین محلی به‌دست می‌آورد. سپس در مرحله همفکری، با ترکیب این تخمین‌های محلی به‌دست‌آمده از گره‌های همسایه، به تخمین جدید می‌رسد. در روش CTA ترتیب مراحل عوض می‌شود. رابطه کلی الگوریتم LMS در شبکه انتشاری توزیع‌شده به‌صورت روابط (۱-۳) است [۲۱]:

$$\Phi_i(n-1) = \sum_{l \in J_i} a_{1,li} w_l(n-1) \tag{1}$$

$$\Psi_i(n) = \Phi_i(n-1) + \mu_i \sum_{l \in J_i} c_{li} u_l(n) [d_i(n) - u_l^T(n) \Phi_i(n-1)] \tag{2}$$

$$w_i(n) = \sum_{l \in J_i} a_{2,li} \Psi_l(n) \tag{3}$$

که مقادیر  $\Phi_i(n-1)$  و  $\Psi_i(n)$  مقادیر تخمین میانی هستند که هر گره از همسایه‌های خود دریافت می‌کند. ضرایب ترکیب‌گر  $a_{1,li}$ ،  $a_{2,li}$  و  $c_{li}$  نامنفی و حقیقی هستند. این ضرایب برای وزن‌دهی تبادل اطلاعات بین هر دو گره دلخواه  $i$  و  $l$  است. پیاده‌سازی الگوریتم LMS در شبکه انتشاری توزیع‌شده در حالت ATC به‌صورت روابط (۴) و (۵) تحقق می‌یابد.

$$\Psi_i(n) = w_i(n-1) + \mu_i \sum_{l \in J_i} c_{li} u_l(n) [d_i(n) - u_l^T(n) w_i(n-1)] \tag{4}$$

$$w_i(n) = \sum_{l \in J_i} a_{li} \Psi_l(n) \tag{5}$$

که ضرایب ترکیب‌گر  $c_{li}$  و  $a_{li}$  به‌ترتیب از قانون میانگیری و Metropolis به‌دست می‌آیند. شرایط ابتدایی  $w_i(-1) = 0$  برای همه گره‌ها استفاده می‌شود.

بنابراین معادله اصلاح وزن در الگوریتم NSAF به صورت رابطه (۱۱) به دست می آید.

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu \sum_{s=0}^{N-1} \frac{\mathbf{u}_s(k)}{\|\mathbf{u}_s(k)\|^2 + \varepsilon} e_{s,D}(k) \quad (11)$$

که در آن  $\mu$  اندازه گام است و  $\varepsilon$  پارامتر تنظیم که جهت جلوگیری از صفر شدن مخرج قرار داده می شود. خطای پیشین و پسین زیرباند زیرنمونه برداری شده به ترتیب به صورت روابط (۱۲) و (۱۳) تعریف می شوند.

$$e_{s,D}(k) = d_{s,D}(k) - \mathbf{u}_s^T(k) \mathbf{w}(k) \quad (12)$$

$$\xi_{s,D}(k) = d_{s,D}(k) - \mathbf{u}_s^T(k) \mathbf{w}(k+1) \quad (13)$$

برای استفاده مجدد از داده ها، ماتریس سیگنال ورودی و بردارهای خروجی مطلوب و خطای پیشین و پسین به صورت روابط (۱۴-۱۷) تعریف می گردند.

$$\mathbf{U}(k) = [\mathbf{u}_0(k), \dots, \mathbf{u}_0(k-P+1), \mathbf{u}_1(k), \dots, \mathbf{u}_1(k-P+1), \dots, \mathbf{u}_{N-1}(k), \dots, \mathbf{u}_{N-1}(k-P+1)] \quad (14)$$

$$\mathbf{d}_D(k) = [d_{0,D}(k), \dots, d_{0,D}(k-P+1), d_{1,D}(k), \dots, d_{1,D}(k-P+1), \dots, d_{N-1,D}(k), \dots, d_{N-1,D}(k-P+1)]^T \quad (15)$$

$$\mathbf{e}_D(k) = [e_{0,D}(k), \dots, e_{0,D}(k-P+1), e_{1,D}(k), \dots, e_{1,D}(k-P+1), \dots, e_{N-1,D}(k), \dots, e_{N-1,D}(k-P+1)] \quad (16)$$

$$\xi_D(k) = [\xi_{0,D}(k), \dots, \xi_{0,D}(k-P+1), \xi_{1,D}(k), \dots, \xi_{1,D}(k-P+1), \dots, \xi_{N-1,D}(k), \dots, \xi_{N-1,D}(k-P+1)] \quad (17)$$

در این روابط  $P$  تعداد رگسورها  $M$  و  $2$  تعداد ضرایب فیلتر است. الگوریتم IMSAF تابع ارزش رابطه (۱۸) را کمینه می کند.

$$\min_{\mathbf{w}(k+1)} \|\mathbf{w}(k+1) - \mathbf{w}(k)\|^2 \quad \text{subject to} \quad \xi_D(k) = \mathbf{0} \quad (18)$$

که در آن  $\mathbf{0}$  ماتریس صفر  $NP \times I$  است. مسأله کمینه کردن قید می تواند با روش ضرایب لاگرانژ حل گردد. بنابراین تابع ارزش به صورت رابطه (۱۹) انتخاب می شود.

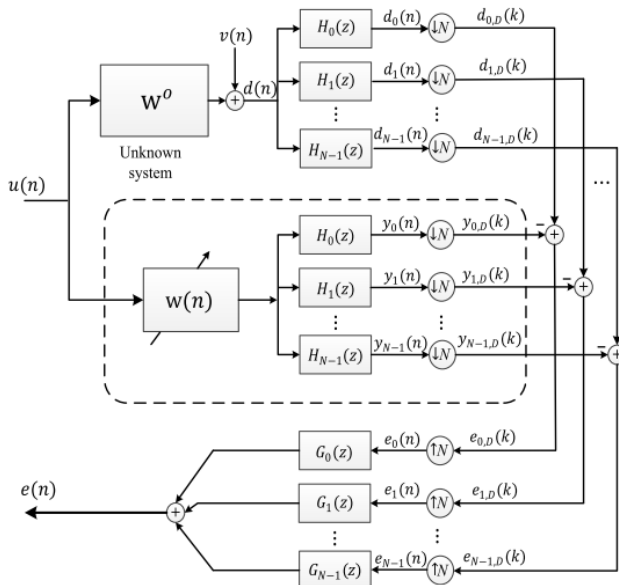
$$J[\mathbf{w}(k+1)] = \frac{1}{2} \|\mathbf{w}(k+1) - \mathbf{w}(k)\|^2 + \lambda^T \xi_D(k) \quad (19)$$

که  $\lambda = [\lambda_0, \lambda_1, \dots, \lambda_{NP-1}]^T$  بردار ضرایب لاگرانژ است.

با مشتق گرفتن از رابطه (۱۹) نسبت به  $\mathbf{w}(k+1)$  و برابر صفر قرار دادن حاصل و جای گذاری، رابطه (۲۰) حاصل می شود.

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu \mathbf{U}(k) [\mathbf{U}^T(k) \mathbf{U}(k) + \varepsilon \mathbf{I}]^{-1} \mathbf{e}_D(k) \quad (20)$$

در این رابطه  $\mathbf{I}$  ماتریس همانی به اندازه  $NP \times NP$  است.



شکل ۴: ساختار الگوریتم NSAF [۲۶]

پاسخ مطلوب  $d(n)$  که از مدل خطی با رابطه (۶) ناشی می گردد را در نظر بگیرید:

$$d(n) = \mathbf{u}^T(n) \mathbf{w}^o + v(n) \quad (6)$$

که در آن  $\mathbf{w}^o = [w_0, w_1, \dots, w_{M-1}]^T$  بردار سیستم ناشناخته به طول  $M$   $\mathbf{u}(n) = [u(n), u(n-1), \dots, u(n-M+1)]^T$  بردار سیگنال ورودی و  $v(n)$  معرف نویز سیستم با واریانس  $\sigma_v^2$  است. فرض می شود  $v(n)$  نویز سفید گوسی با میانگین صفر و مستقل از  $\mathbf{u}(n)$  است. طبق شکل ۴ سیگنال مطلوب  $d(n)$  و سیگنال ورودی  $\mathbf{u}(n)$  به  $N$  سیگنال زیرباند  $d_s(n)$  و  $\mathbf{u}_s(n)$  طبق آنالیز بانک فیلتر  $H_s(z)$  تقسیم می گردند ( $s = 0, 1, \dots, N-1$ ). سیگنال ورودی  $\mathbf{u}_s(n)$  به وسیله فیلتر تطبیقی، فیلتر شده و سیگنال خروجی  $y_s(n)$  را تولید می کند. سیگنال های زیرباند  $d_s(n)$  و  $y_s(n)$  با فاکتور  $N$  زیرنمونه برداری شده و  $d_{s,D}(k)$  و  $y_{s,D}(k)$  را ایجاد می کنند.

$$y_{s,D}(k) = \mathbf{u}_s^T(k) \mathbf{w}(k) \quad (7)$$

که  $\mathbf{u}_s(k)$  بردار داده های ورودی برای  $s$  امین زیرباند، و به صورت رابطه (۸) است.

$$\mathbf{u}_s(k) = [u_s(kN), u_s(kN-1), \dots, u_s(kN-M+1)]^T \quad (8)$$

$$s = 0, 1, \dots, N-1$$

سیگنال خطای زیرنمونه برداری شده نیز به صورت رابطه (۹) است.

$$e_{s,D}(k) = d_{s,D}(k) - \mathbf{u}_s^T(k) \mathbf{w}(k) \quad (9)$$

که در آن اسکالر  $d_{s,D}(k)$  خروجی مطلوب زیرنمونه برداری شده، مطابق رابطه (۱۰) است.

$$d_{s,D}(k) = d_s(kN) \quad (10)$$

**جدول ۳: شبه کد الگوریتم پیشنهادی در شبکه توزیع شده**

**انتشاری با استراتژی ATC**

مقداردهی اولیه  $w_i(-1) = 0$  برای تمام گره‌ها در هر تکرار

**گام اول: زیرباند:**

۱. جداسازی باندهای ورودی و مطلوب در هر گره

$$\mathbf{u}_{s,i}(n) = h_s^T \mathbf{u}_i(n) \quad i=0, \dots, J, \quad s=0, \dots, N-1$$

$$\mathbf{d}_{s,i}(n) = h_s^T \mathbf{d}_i(n) \quad i=0, \dots, J, \quad s=0, \dots, N-1$$

۲. زیرنمونه‌برداری در هر گره

$$\mathbf{u}_{s,i}(k) = [u_{s,i}(kN), u_{s,i}(kN-1), \dots, u_{s,i}(kN-M+1)]^T$$

$$d_{s,D,i}(k) = d_{s,i}(kN)$$

$$e_{s,D,i}(k) = d_{s,D,i}(k) - \mathbf{u}_{s,i}^T(k) \mathbf{w}_i(k)$$

۳. محاسبه ماتریس رگرسورهای اخیر  $(\mathbf{U}_i(k))$  و سیگنال مطلوب

زیرنمونه‌برداری شده  $\mathbf{d}_{D,i}(k)$  و خطا  $e_{D,i}(k)$  در هر گره از شبکه

$$\mathbf{U}_i(k) = [\mathbf{u}_{0,i}(k), \dots, \mathbf{u}_{0,i}(k-P+1), \dots, \mathbf{u}_{N-1,i}(k), \dots, \mathbf{u}_{N-1,i}(k-P+1)]$$

$$\mathbf{d}_{D,i}(k) = [d_{0,D,i}(k), \dots, d_{0,D,i}(k-P+1), \dots, d_{N-1,D,i}(k), \dots, d_{N-1,D,i}(k-P+1)]^T$$

$$\mathbf{e}_{D,i}(k) = [e_{0,D,i}(k), \dots, e_{0,D,i}(k-P+1), \dots, e_{N-1,D,i}(k), \dots, e_{N-1,D,i}(k-P+1)]$$

**گام دوم: شبکه**

۴. محاسبه وزن محلی  $\psi_i(n)$  در هر گره  $i$  با استفاده از داده‌های همسایگی

$$\psi_i(k) = \mathbf{w}_i(k-1) + \mu \sum_{l \in J_i} c_{li} \sum_{r=0}^{NP-1} \frac{\mathbf{x}^{(r,l)}}{\|\mathbf{x}^{(r,l)}\|^2 + \varepsilon} [d_D^{(r,l)} - (\mathbf{x}^{(r,l)})^T \mathbf{w}_i(k-1)]$$

۵. اصلاح وزن در هر گره با استفاده از تخمین‌های محلی

$$\mathbf{w}_s(n) = \sum_{i \in J_s} a_{is} \cdot \psi_i(n)$$

**۴- پیچیدگی محاسباتی**

جدول ۴ پیچیدگی محاسباتی الگوریتم ATC-DIMSAP در تعداد ضرب‌ها و جمع‌ها برای هر گره از شبکه توزیع شده در هر تکرار  $n$  را نشان می‌دهد. بدیهی است که پیچیدگی محاسباتی استراتژی CTA نیز همانند استراتژی ATC است، لذا از جدول اشاره شده برای هر دو استراتژی استفاده می‌شود. در این جدول  $M$  برابر با طول فیلتر تطبیقی  $P$  تعداد رگرسورها،  $N$  تعداد زیرباندها و  $L$  طول فیلتر بانک است.

**جدول ۴: پیچیدگی محاسباتی الگوریتم DIMSAP برای هر گره از**

**شبکه**

معادلات	تعداد ضرب‌ها	تعداد جمع‌ها
$\mu \sum_{l \in J_i} c_{ls} \sum_{r=0}^{NP-1} \frac{\mathbf{x}^{(r,l)}}{\ \mathbf{x}^{(r,l)}\ ^2 + \varepsilon} [d_D^{(r,l)} - (\mathbf{x}^{(r,l)})^T \psi_s(k-1)]$	$j_i \times [3PM + (N+2)L]$	$(j_i - 1) \times M + [PM + M + 3(L-1)]$
$\mathbf{w}_s(k-1) + \mu \sum_{l \in J_i} c_{ls} \sum_{r=0}^{NP-1} \frac{\mathbf{x}^{(r,l)}}{\ \mathbf{x}^{(r,l)}\ ^2 + \delta} [d_D^{(r,l)} - (\mathbf{x}^{(r,l)})^T \psi_s(k-1)]$	$j_i \times [3PM + (N+2)L]$	$(j_i - 1) \times M + [PM + 3(L-1) + 2M]$
$a_{is} \mathbf{w}_i(k-1)$	$M$	-
$\sum_{i \in J_i} a_{is} \mathbf{w}_i(k-1)$	$j_i \times M$	$(j_i - 1) \times M$
کل محاسبات انجام شده در یک تکرار	$j_i \times [3PM + (N+2)L + M]$	$(j_i - 1) \times 2M + [PM + 3(L-1) + 2M]$

اصلاح وزن در الگوریتم IMSAF مشابه با الگوریتم AP است؛ تنها اختلاف این است که ماتریس  $\mathbf{U}(k)$  در الگوریتم AP ترکیب بردارهای سیگنال قبل از نمونه‌برداری است، در صورتی که در الگوریتم IMSAF متشکل از بردارهای سیگنال نمونه‌برداری شده است. اگر از درایه‌های غیرقطری  $\mathbf{U}^T(k)\mathbf{U}(k)$  صرف‌نظر شود رابطه (۲۰) می‌تواند به فرم رابطه (۱۱) ساده گردد. همبستگی متقابل  $\mathbf{u}_i^T(m)\mathbf{u}_p(m-l)$  دو سیگنال زیرباند دلخواه می‌تواند با یک طراحی آنالیز فیلتر مناسب نادیده گرفته شود. در حالی که ماتریس خودهمبستگی  $\mathbf{u}_i^T(m)\mathbf{u}_i(m-l)$  نمی‌تواند نادیده گرفته شود زیرا همبستگی در زیرباندها برای سیگنال‌های بسیار رنگی، بالا است. در نتیجه از درایه‌های غیرقطری  $\mathbf{U}^T(k)\mathbf{U}(k)$  نمی‌توان صرف نظر نمود.

با در نظر گرفتن  $\mathbf{x}^{(r)}$  به‌عنوان  $r$  امین بردار ستونی ماتریس  $\mathbf{U}(k)$  و  $e^{(r)}$  به‌عنوان  $r$  امین درایه بردار سطری  $\mathbf{e}_D(k)$  در نهایت رابطه اصلاح وزن الگوریتم IMSAF به‌صورت رابطه (۲۱) به‌دست می‌آید [۱۳].

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \sum_{r=0}^{NP-1} \frac{\mu \mathbf{x}^{(r)} e^{(r)}}{\|\mathbf{x}^{(r)}\|^2 + \varepsilon} \quad (21)$$

با توجه به ویژگی برجسته الگوریتم IMSAF از نظر بالابودن سرعت همگرایی، این الگوریتم را می‌توان در شبکه‌های توزیع شده، توسعه داد. در شبکه توزیع شده انتشاری برای هر گره  $i$  یک ماتریس  $\mathbf{U}(k)$  خواهیم داشت. بنابراین معادلات این الگوریتم در حالت انتشاری با استراتژی ATC به‌صورت روابط (۲۲) و (۲۳) استخراج می‌گردد.

$$\psi_i(k) = \mathbf{w}_i(k-1) + \mu \sum_{l \in J_i} c_{li} \sum_{r=0}^{NP-1} \frac{\mathbf{x}^{(r,l)}}{\|\mathbf{x}^{(r,l)}\|^2 + \varepsilon} [d_D^{(r,l)} - (\mathbf{x}^{(r,l)})^T \mathbf{w}_i(k-1)] \quad (22)$$

$$\mathbf{w}_i(k) = \sum_{l \in J_i} a_{li} \psi_l(k) \quad (23)$$

که در آن  $\mathbf{x}^{(r,l)}$  را  $r$  امین بردار ستونی ماتریس  $\mathbf{U}(k)$  مربوط به گره  $l$  است.  $d_D^{(r,l)}$  نیز  $r$  امین درایه سیگنال خروجی مطلوب زیرنمونه‌برداری شده مربوط به گره  $l$  است. همان‌طور که در قسمت دوم بیان شد،  $c_{li}$  و  $a_{li}$  ضرایب ترکیب‌گر بوده و به‌ترتیب از قانون میانگیری و Metropolis، مطابق جدول (۱) به‌دست می‌آیند. این معادلات با استراتژی CTA مطابق روابط (۲۴) و (۲۵) تحقق می‌یابند.

$$\psi_i(k-1) = \sum_{l \in J_i} a_{li} \mathbf{w}_l(k-1) \quad (24)$$

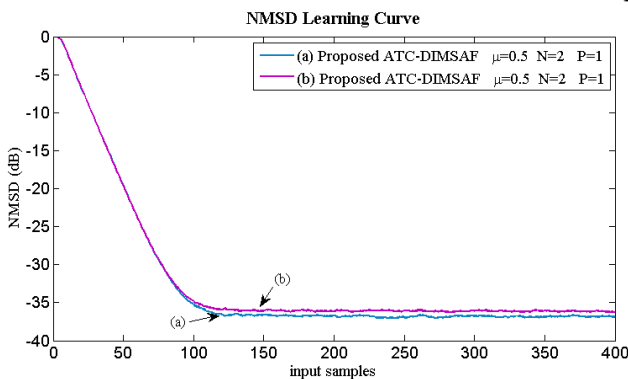
$$\mathbf{w}_i(k) = \psi_i(k-1) + \mu \sum_{l \in J_i} c_{lk} \sum_{r=0}^{NP-1} \frac{\mathbf{x}^{(r,l)}}{\|\mathbf{x}^{(r,l)}\|^2 + \varepsilon} [d_D^{(r,l)} - (\mathbf{x}^{(r,l)})^T \psi_i(k-1)] \quad (25)$$

شبه کد الگوریتم پیشنهادی با استراتژی ATC در جدول ۳ آورده شده است.

$$NMSD_{network} = \frac{1}{J} \sum_{i=1}^J E \left[ \frac{\|w^0 - w_i(k)\|^2}{\|w^0\|^2} \right] \quad (27)$$

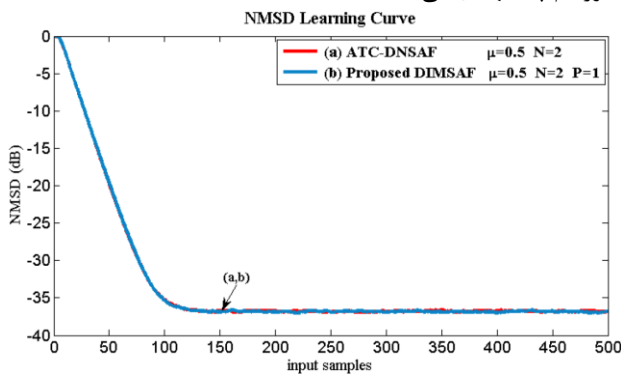
همه منحنی‌های یادگیری به دست آمده توسط متوسط‌گیری از ۲۰۰ تکرار مستقل به دست آمده‌اند. در تمام شبیه‌سازی‌ها اندازه گام برای تمام گره‌ها یکسان و برابر  $\mu_i = 0.5$  انتخاب شده است.

در شکل ۷ منحنی یادگیری NMSD الگوریتم DIMSAF در حالت ATC و شبیه‌سازی و با یکدیگر مقایسه شده است. در این آزمایش تعداد زیرباند‌ها برابر با ۲ و تعداد رگرورها برابر با ۱ می‌باشد. مشاهده می‌گردد اگرچه اختلاف در این دو حالت ناچیز است اما این الگوریتم در حالت ATC عملکرد بهتری دارد. بنابراین در شبیه‌سازی‌ها از حالت ATC استفاده شده است.



شکل ۷: مقایسه الگوریتم پیشنهادی DIMSAF با استراتژی‌های ATC و CTA

شکل ۸ شبیه‌سازی منحنی یادگیری NMSD الگوریتم DIMSAF را به‌ازای دو زیرباند ( $N=2$ ) و یک رگرور ( $P=1$ ) نشان می‌دهد. همان‌طور که انتظار می‌رود هنگامی که فقط یک رگرور در الگوریتم پیشنهادی مورد استفاده قرار گیرد خروجی‌های الگوریتم‌های DNSAF و DIMSAF یکسان بوده و این دو منحنی برهم منطبق خواهند شد. تطابق دو منحنی رسم شده بیان‌گر صحت روابط استخراجی برای الگوریتم پیشنهادی می‌باشد.



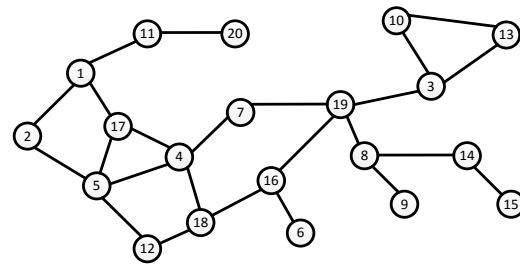
شکل ۸: مقایسه منحنی یادگیری NMSD الگوریتم‌های DNSAF و الگوریتم پیشنهادی DIMSAF

شکل ۹ شبیه‌سازی منحنی یادگیری NMSD را برای الگوریتم DIMSAF به‌ازای تعداد زیرباند یکسان ( $N=2$ ) و مقادیر مختلف تعداد

مطابق جدول ۴ پیچیدگی محاسباتی شبکه توزیع شده (کل گره‌ها) در تعداد ضرب‌ها برای الگوریتم DIMSAF در هر تکرار  $n$  به صورت  $\sum_{i=1}^J [j_i \times (3PM + (N + 2)L + M)]$  به دست می‌آید ( $j_i$  درجه همسایگی گره  $i$  است).

### ۵- نتایج شبیه‌سازی

در این بخش با استفاده از نرم‌افزار MATLAB، به بررسی عملکرد الگوریتم DIMSAF در یک شبکه انتشاری با ۲۰ گره ( $J = 20$ ) مطابق شکل ۵، برای کاربرد شناسایی سیستم پرداخته شده است.

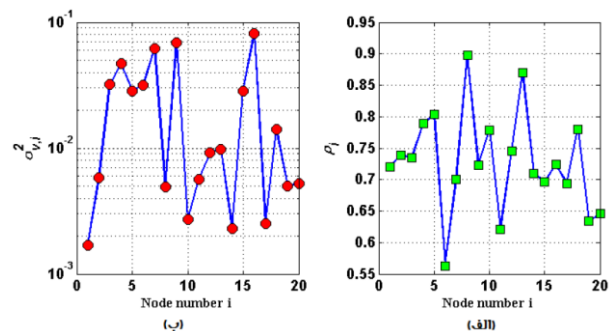


شکل ۵: توپولوژی شبکه‌ای با ۲۰ گره [۲۷]

پاسخ ضربه سیستم ناشناخته یک دنباله تصادفی گوسی با طول ۱۶ می‌باشد ( $M = 16$ ). سیگنال ورودی رنگی در هر گره از طریق گذراندن دنباله تصادفی گوسی سفید متوسط صفر با واریانس واحد ( $\sigma_{u,i}^2 = 1$ ) از یک فیلتر  $AR$  مرتبه اول با تابع تبدیل رابطه (۲۶) حاصل شده است.

$$G_i(z) = \frac{1}{(1 - \rho_i z^{-1})} \quad (26)$$

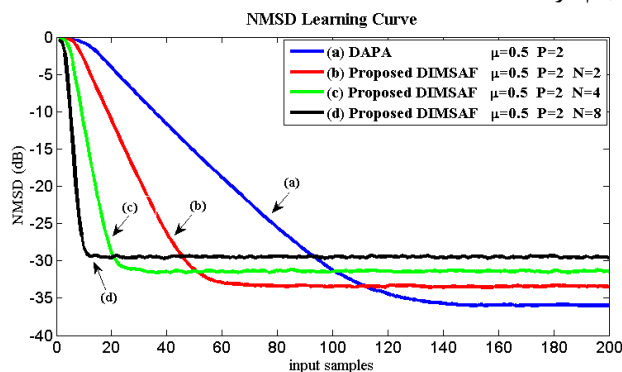
دنباله نویز افزودنی هر گره یک فرایند گوسی سفید متوسط صفر با واریانس  $\sigma_{v,i}^2$  است. شکل ۶، مقدار واریانس نویز مربوط به هر گره و مقدار  $\rho_i$  را برای هر گره نشان می‌دهد. جهت ترکیب وزن‌ها ( $a_{ii}, c_{ii}$ ) از قانون میانگیری و Metropolis استفاده شده است. فیلتربانک استفاده شده در شبیه‌سازی‌ها (ELT) [۲۸] می‌باشد.



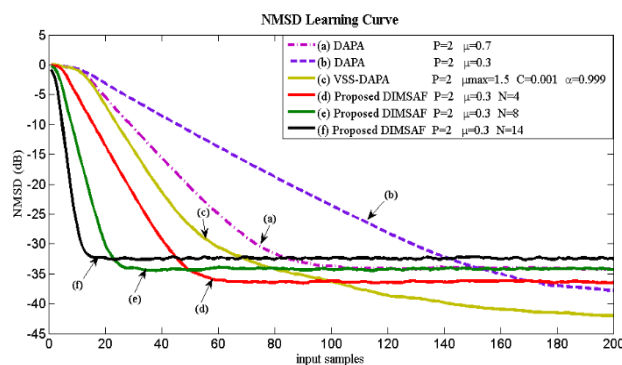
شکل ۶: (الف) مقادیر  $\rho_i$  و (ب) واریانس نویز افزودنی برای هر از گره شبکه انتشاری

عملکرد الگوریتم‌ها به وسیله منحنی یادگیری میانگین مربع انحراف نرمالیزه شده شبکه (NMSD) که به صورت رابطه (۲۷) تعریف می‌شود، سنجیده می‌شود.

الگوریتم پیشنهادی (DIMSAP) به‌ازای تعداد رگرسور یکسان ( $P=2$ ) و مقادیر مختلف زیرباند مقایسه شده است. همان‌طور که در شکل مشخص است، اگرچه الگوریتم VSS-DAPA سرعت همگرایی بالاتر و خطای حالت ماندگار کمتری را در مقایسه با DAPA دارد، اما باز هم الگوریتم پیشنهادی سرعت همگرایی بالاتری دارد که با افزایش تعداد زیرباند این برتری فزونی می‌یابد. پیاده‌سازی الگوریتم VSS-DAPA بر مبنای الگوریتم منفرد VSS-APA مرجع [۳۲] در شبکه انتشاری انجام گرفته شده است.



شکل ۱۱: مقایسه الگوریتم DAPA و الگوریتم پیشنهادی DIMSAF به‌ازای تعداد رگرسور یکسان و تعداد زیرباند متفاوت برای الگوریتم پیشنهادی

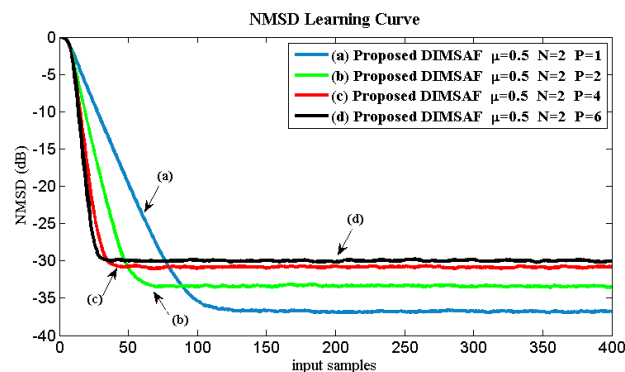


شکل ۱۲: مقایسه الگوریتم‌های DAPA و VSS-DAPA با الگوریتم پیشنهادی DIMSAF به‌ازای تعداد رگرسور یکسان برای الگوریتم‌ها و تعداد زیرباند متفاوت برای الگوریتم پیشنهادی

شکل ۱۳ مقایسه منحنی یادگیری NMSD را برای الگوریتم‌های DRLS [۳۳] برای دو مقدار  $\delta = 0.1$  و  $\delta = 0.01$  و الگوریتم DIMSAF به‌ازای تعداد رگرسور  $P=2$  و مقادیر مختلف زیرباند نشان می‌دهد. همان‌طور که مشخص است اگرچه الگوریتم DRLS خطای حالت ماندگار کمتری دارد، اما الگوریتم پیشنهادی سرعت بهتری را با افزایش تعداد زیرباند منجر می‌شود.

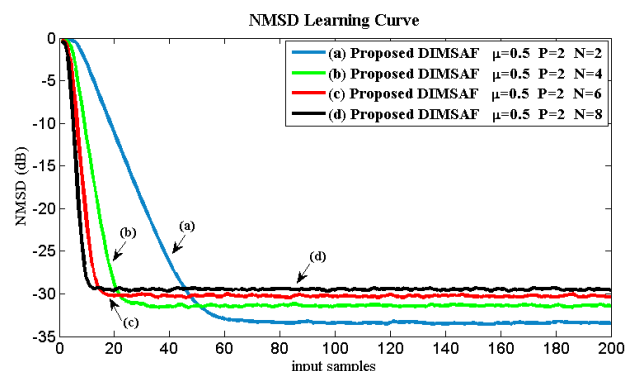
در شکل ۱۴ الگوریتم DRLS با الگوریتم پیشنهادی با تعداد زیرباند یکسان و مقادیر مختلف رگرسور مقایسه شده است. به‌ازای مقادیر  $P=1$  و  $N=2$  الگوریتم پیشنهادی خطای حالت ماندگاری تقریباً برابر با DRLS و سرعت همگرایی کمتری دارد که با افزایش تعداد

رگرسورها را نشان می‌دهد. مشاهده می‌شود که با افزایش تعداد رگرسورها، سرعت همگرایی و خطای حالت ماندگار بیشتر می‌گردد.



شکل ۹: منحنی یادگیری NMSD الگوریتم پیشنهادی DIMSAF به‌ازای تعداد زیرباند یکسان و تعداد رگرسور مختلف

شکل ۱۰ شبیه‌سازی منحنی یادگیری NMSD را برای الگوریتم DIMSAF به‌ازای تعداد رگرسور یکسان ( $P=2$ ) و مقادیر مختلف تعداد زیرباند، سرعت همگرایی الگوریتم بیشتر، پیچیدگی محاسباتی الگوریتم افزایش و خطای حالت ماندگار نیز افزایش خواهد یافت.



شکل ۱۰: منحنی یادگیری NMSD الگوریتم پیشنهادی DIMSAF به‌ازای تعداد رگرسور یکسان و تعداد زیرباند‌های مختلف

شکل ۱۱ به مقایسه منحنی یادگیری NMSD الگوریتم‌های DIMSAF و DAPA به‌ازای تعداد رگرسور یکسان ( $P=2$ ) و مقادیر مختلف زیرباند برای الگوریتم DIMSAF می‌پردازد. مشاهده می‌شود که سرعت همگرایی الگوریتم پیشنهادی به‌مراتب بیشتر از الگوریتم DAPA است. همچنین با افزایش تعداد زیرباند در الگوریتم پیشنهادی سرعت همگرایی نیز افزایش می‌یابد.

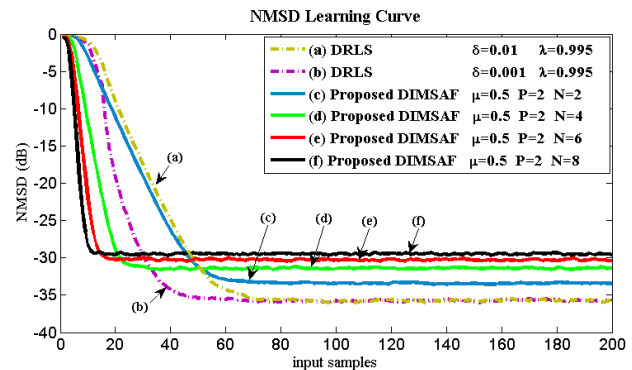
شکل ۱۲ الگوریتم تصویر افاین با اندازه‌گام متغیر در شبکه انتشاری (VSS-DAPA) به‌ازای  $\mu_{max} = 0.5$ ,  $C = 0.01$ ,  $\alpha = 0.999$  و دو اندازه‌گام متفاوت ( $\mu = 0.3$  و  $\mu = 0.7$ ) مقایسه می‌کند. الگوریتم‌های با اندازه‌گام متغیر به‌منظور دستیابی همزمان به سرعت همگرایی بالا و خطای حالت ماندگار پایین معرفی شده‌اند (همان‌گونه که در شکل نیز مشخص است). در این شکل همچنین این الگوریتم (VSS-DAPA) با



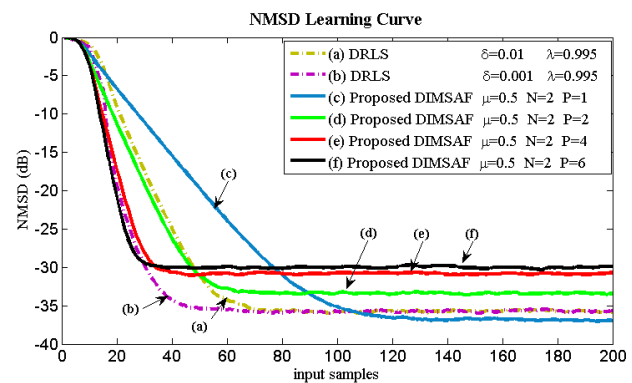
مراجع

- [1] I. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci. "A survey on sensor networks," IEEE Communications Magazine, vol. 40, pp. 102-114, 2002.
- [2] A. H. Sayed and C. G. Lopez. "Adaptive processing over distributed networks," 9th International Symposium on Signal Processing and Its Applications (ISSPA), Sharjah, United Arab Emirates, February 2007.
- [3] D. Li, K. D. Wong, Y. H. Hu and A. M. Sayeed, "Detection, classification, and tracking of targets," IEEE Signal Processing Magazine, vol. 19, pp. 17-29, 2002.
- [4] C. G. Lopes, and A. H. Sayed, "Incremental adaptive strategies over distributed networks," IEEE Trans. Signal Processing, vol. 55, pp. 4064-4077, 2007.
- [5] P. Di Lorenzo, S. Barbarossa and A. H. Sayed, "Bio-inspired swarming for dynamic radio access based on diffusion adaptation," in Proc. Eur. Signal Process. Conf. (EUSIPCO), pp. 402-406, Barcelona, Spain, August 2011.
- [6] J. Chen, X. Cao, P. Cheng, Y. Xiao and Y. Sun, "Distributed collaborative control for industrial automation with wireless sensor and actuator networks," IEEE Trans. Ind. Electron, vol. 57, no. 12, pp. 4219-4230, 2010.
- [7] M. de Courville and P. Duhamel, "Adaptive filtering in subbands using a weighted criterion," IEEE Trans. Signal Processing, vol. 46, pp. 2359-2371, September 1998.
- [8] S. S. Pradhan and V. E. Reddy, "A new approach to subband adaptive filtering," IEEE Trans. Signal Processing, vol. 47, pp. 655-664, March 1999.
- [9] J. J. Shynk, "Frequency domain and multi rate adaptive filtering," IEEE Signal Processing Mag, vol. 9, pp. 14-37, January 1992.
- [10] A. Gilloire and M. Vetterli, "Adaptive filtering in subbands with critical sampling: Analysis, experiments, and application to acoustic echo cancellation," IEEE Trans. Signal Processing, vol. 40, pp. 1862-1875, August 1992.
- [11] P. L. De Leon and D. M. Etter, "Acoustic echo cancellation using subband adaptive filtering," in Subband and Wavelet Transforms, A. N. Akansu and M. J. T. Smith, Eds. Boston, MA: Kluwer, 1996.
- [12] S. Haykin, *Adaptive Filter Theory*. Englewood Cliffs, NJ: Prentice Hall, 4<sup>th</sup> edition, 2002.
- [13] F. Yang, M. Wu, P. Ji and J. Yang, "An Improved Multiband-Structured Subband Adaptive Filter Algorithm," IEEE Signal Process. Lett, vol. 19, no. 10, pp. 647-650, October 2012.
- [14] A. H. Sayed. In: Chellapa R, Theodoridis S, eds. *Diffusion Adaptation Over Networks*, Academic Press Library in Signal Processing, vol. 3: Academic Press, Elsevier; 2014:323-454.
- [15] A. H. Sayed, *Adaptation, Learning, and Optimization Over Networks*, Foundations and Trends in Machine Learning, vol. 7, no. 4-5; NOW Publishers Inc. Boston - Delft ; 2014: 311-801.
- [16] A. H. Sayed, S.Y. Tu, J. Chen, X. Zhao and Z. J. Towfic, "Diffusion strategies for adaptation and learning over networks," IEEE Signal Process Mag, vol. 30, no. 3, pp. 155-171, May 2013.
- [17] A. H. Sayed, "Adaptive networks," Proceedings of the IEEE, vol. 102, no. 4, pp. 460-497, April 2014.
- [18] N. Takahashi, I. Yamada and A. H. Sayed, "Diffusion least-mean squares with adaptive combiners: formulation

رگرسورها سرعت همگرایی برابر با DRLS خواهد داشت. ضرایب ترکیبگر  $(a_{li}, c_{li})$  برای هر دو الگوریتم از قانون میان‌گیری و Metropolis استفاده شده است.



شکل ۱۳: مقایسه منحنی یادگیری NMSD الگوریتم DRLS و الگوریتم پیشنهادی DIMSAF به‌ازای مقادیر مختلف زیرباند برای الگوریتم پیشنهادی



شکل ۱۴: مقایسه منحنی یادگیری NMSD الگوریتم DRLS و الگوریتم پیشنهادی DIMSAF به‌ازای مقادیر مختلف رگرسور برای الگوریتم پیشنهادی

۶- نتیجه‌گیری

در این تحقیق الگوریتم فیلتر تطبیقی زیرباند نرمالیزه‌شده توسعه‌یافته در شبکه توزیع‌شده انتشاری (DIMSAF) ارائه و روابط این الگوریتم با دو استراتژی ATC و CTA استخراج گردید. مشاهده شد که اگر در الگوریتم DNSAF از ماتریس رگرسورهای ورودی به‌جای تک بردار رگرسور ورودی استفاده شود سرعت همگرایی بهبود می‌یابد. با انجام شبیه‌سازی‌های متعدد صحت روابط الگوریتم پیشنهادی بررسی شد و ملاحظه گردید که این الگوریتم در مقایسه با سایر الگوریتم‌های پیاده‌سازی شده در شبکه‌های انتشاری، همچون الگوریتم‌های DAPA، DNSAF، VSS-DAPA و DRLS دارای سرعت همگرایی بالاتری می‌باشد.

- [27] H. S. Lee, S. E. Kim, J. W. Lee and W. J. Song, "A Variable Step-Size Diffusion LMS Algorithm for Distributed Estimation," *IEEE Trans. Signal Processing*, vol. 63, no. 7, pp 1808-1820 April 2015.
- [28] H. Malvar, *Signal Processing with Lapped Transforms*. Artech House, 1992.
- [۲۹] امیر رستگاری، اعظم خلیلی و توحید یوسفی رضایی، «الگوریتم مبتنی بر شبکه تطبیقی نفوذی برای تخمین مقاوم میدان اسکالر در شبکه‌های سنسوری بی‌سیم»، *مجله مهندسی برق دانشگاه تبریز*، دوره ۴۷، شماره ۲، صفحات ۵۳۵-۵۲۷، ۱۳۹۶.
- [۳۰] مرتضی فرهید، موسی شمسی و محمدحسین صدیقی، «تأثیر توپولوژی شبکه‌های پیچیده بر روی عملکرد تخمین تطبیقی توزیع شده با مشارکت نفوذی»، *مجله مهندسی برق دانشگاه تبریز*، دوره ۴۶، شماره ۴، صفحات ۲۱۶-۲۰۷، ۱۳۹۵.
- [۳۱] قنبر آذرین، محمد علی طینتی، «افزایش همگرایی شبکه‌های تطبیقی با لینک‌های نویزی: الگوریتم حداقل میانگین مربعات توزیع شده افزایشی طول متغیر»، *مجله مهندسی برق دانشگاه تبریز*، دوره ۴۵، شماره ۳، صفحات ۱۵-۱، ۱۳۹۴.
- [32] H. C. Shin, A. H. Sayed and W. J. Song, "Variable Step-Size NLMS and Affine Projection Algorithms," *IEEE Signal Process. Lett.*, vol. 11, no. 2, pp. 132-135, February 2004.
- [33] F. S. Cattivelli, C. G. Lopes and A. H. Sayed, "Diffusion recursive least-squares for distributed estimation over adaptive networks," *IEEE Transactions on Signal Processing*, vol. 56, no. 5, pp. 1865-1877, May 2008.
- and performance analysis," *IEEE T Signal Proces*, vol. 58, no. 9, pp. 4795-4810, September 2010.
- [19] J. B. Predd, S. R. Kulkarni and H. V. Poor, "Distributed learning in wireless sensor networks," *IEEE Signal Process Mag*, vol. 23, no. 4, pp. 56-69, July 2006.
- [20] C. G. Lopes and A. H. Sayed, "Diffusion least-mean squares over adaptive networks: formulation and performance analysis," *IEEE T Signal Proces*, vol. 56, no. 7, pp. 3122-3136, July 2008.
- [21] P. Braca, S. Marano and V. Matta, "Running consensus in wireless sensor networks," In *Proc. 11th International Conference on Information Fusion*, pages 1-6, Cologne, Germany, June 2008.
- [22] A. H. Sayed, "Diffusion adaptation over networks," to appear in *E-Reference Signal Processing*, R. Chellapa and S. Theodoridis, editors, Elsevier, 2013.
- [23] V. D. Blondel, J. M. Hendrickx, A. Olshevsky and J. N. Tsitsiklis, "Convergence in multiagent coordination, consensus, and flocking," *Proc. Joint 44th IEEE Conf. on Decision and Control and European Control Conf. (CDC-ECC)*, pp. 2996-3000, Seville, Spain, December 2005.
- [24] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller and E. Teller, "Equations of state calculations by fast computing machines," *Journal of Chemical Physics*, vol. 21, no. 6, pp. 1087-1092, 1953.
- [25] M. de Courville and P. Duhamel, "Adaptive filtering in subbands using a weighted criterion," *IEEE Trans. Signal Processing*, vol. 46, no. 9, pp. 2359-2371, September 1998.
- [26] K. A. Lee and W. S. Gan, "Improving convergence of the NLMS algorithm using constrained subband updates," *IEEE Signal Process. Lett.*, vol. 11, no. 9, pp. 736-739, September 2004.

## زیر نویس‌ها

<sup>1</sup> Distributed networks<sup>2</sup> Nodes<sup>3</sup> Local data<sup>4</sup> Interaction<sup>5</sup> Incremental<sup>6</sup> Diffusion<sup>7</sup> Cognitive radios<sup>8</sup> Least Mean Square<sup>9</sup> Subband Adaptive Filter<sup>10</sup> Normalized Subband Adaptive Filter<sup>11</sup> Improved Multiband-Structured Subband Adaptive Filter<sup>12</sup> Affine Projection Algorithm<sup>13</sup> Real time<sup>14</sup> Combination matrix<sup>15</sup> Averaging<sup>16</sup> Adapt Then Combine<sup>17</sup> Combine Then Adapt<sup>18</sup> Decimated<sup>19</sup> Interpolation<sup>20</sup> Regressors