

بررسی عملکرد الگوریتم‌های تطبیقی زیرباند در شبکه‌های توزیع شده بر پایه روش افزایشی

محمد شمس اسفندآبادی، علی‌رضا دانایی و محمدسعید شفیعی

الگوریتم‌های تطبیقی در این شبکه‌ها مورد توجه قرار گرفته است. در سال‌های اخیر الگوریتم‌های تطبیقی توزیع شده بر مبنای روش افزایشی مانند $dLMS^1$ ، $dNLMS^2$ و $dAPA^3$ ارائه شده‌اند [۳] و [۴]. روش افزایشی به صورت دوره‌ای بین گره‌های همسایه تحقق می‌پذیرد که در این روش اطلاعات در حالت زنجیره‌ای از یک گره به گره هم‌جوار گردش می‌یابند. به این ترتیب یک گره، اطلاعاتی را که خود از محیط دریافت کرده است به وسیله ترکیب با اطلاعاتی که از گره قبلی گرفته است اصلاح می‌نماید و نتیجه را به گره همسایه منتقل می‌کند. در پایان این زنجیره، یک تخمین مناسب از پارامتر مورد نظر به دست می‌آید [۲]. الگوریتم‌های مختلفی برای کاهش پیچیدگی محاسباتی در شبکه‌های توزیع شده افزایشی ارائه گردیده‌اند [۵].

الگوریتم فیلتر تطبیقی زیرباند برای بهبود رفتار همگرایی الگوریتم LMS ارائه شده است. در این الگوریتم سیگنال ورودی و همچنین پاسخ مطلوب از طریق بانک فیلتر به دسته‌های مجزا تقسیم می‌شوند که این عمل باعث همگرایی هر زیرباند به طور مجزا گردیده و بهبود همگرایی کل را در پی خواهد داشت [۶] و [۷].

الگوریتم SAF با یک قید مسئله بهینه‌سازی به الگوریتم NSAF توسعه یافته [۸] و در مقایسه با الگوریتم NLMS، الگوریتم NSAF از سرعت همگرایی خوبی برخوردار است. برای کاهش پیچیدگی محاسباتی در الگوریتم NSAF، الگوریتم‌های متفاوتی معرفی شدند [۹] و [۱۰]. در الگوریتم [SPU-NSAF] ضرایب فیلتر تطبیقی به صورت جزئی در هر زیرباند اصلاح می‌شوند. یکی دیگر از راه‌های کاهش پیچیدگی محاسباتی، انتخاب پویای زیرباندها است که بر این اساس، الگوریتم DS-NSAF ارائه شده است [۱۰]. در این الگوریتم برای اصلاح وزن‌ها، زیرباندها طبق یک معیار به صورت پویا انتخاب می‌گردند. با وجود انتخاب پویای زیرباندها به جای انتخاب کل آنها، این الگوریتم عملکرد بسیار نزدیکی به NSAF دارد. در این تحقیق به ارائه مسئله تخمین توزیع شده در یک شبکه افزایشی بر اساس الگوریتم‌های خانواده زیرباند نرمالیزه شده $(NSAAs)^4$ پرداخته می‌شود. الگوریتم زیرباند نرمالیزه شده توزیع شده $(dNSAA)^5$ ، الگوریتم زیرباند نرمالیزه شده با انتخاب جزئی ضرایب $(dSPU-NSAA)^6$ و الگوریتم زیرباند نرمالیزه شده با انتخاب پویای زیرباندها $(dDS-NSAA)^7$ در قالب واحد معرفی می‌گردند. همچنین با

چکیده: این تحقیق به ارائه مسئله تخمین توزیع شده در یک شبکه افزایشی بر اساس الگوریتم‌های خانواده زیرباند نرمالیزه شده (NSAAs) پرداخته است و الگوریتم‌های زیرباند نرمالیزه شده توزیع شده (dNSAA)، زیرباند نرمالیزه شده توزیع شده با انتخاب جزئی ضرایب (dSPU-NSAA)، زیرباند نرمالیزه شده توزیع شده با انتخاب پویای زیرباندها (dDS-NSAA) و زیرباند نرمالیزه شده توزیع شده با انتخاب جزئی ضرایب و انتخاب پویای زیرباندها (dSPU-DS-NSAA) را در یک قالب واحد معرفی می‌کند. الگوریتم زیرباند نرمالیزه شده توزیع شده تطبیقی نسبت به الگوریتم حداقل میانگین مربعات نرمالیزه شده توزیع شده تطبیقی (dNLMS)، دارای عملکرد بهتری از لحاظ سرعت همگرایی در مواجهه با ورودی‌های رنگی در گره‌ها است. در مقایسه با (dNSAA)، الگوریتم‌های (dSPU-NSAA) و (dDS-NSAA)، علی‌رغم عملکردی مشابه دارای پیچیدگی محاسباتی کمتری هستند. همچنین با ترکیب این دو الگوریتم، الگوریتم (dSPU-DS-NSAA) برقرار می‌شود که از لحاظ پیچیدگی محاسباتی کاهش بیشتری دارد. به علاوه، یک روش واحد برای تحلیل تئوری عملکرد میانگین مربعات برای هر گره ارائه شده و روابط جامعی برای بررسی عملکرد حالت گذرا، ماندگار و باند پایداری برای الگوریتم‌های نوین توزیع شده به دست می‌آید. با انجام شبیه‌سازی‌های مختلف در شبکه‌های توزیع شده، عملکرد مناسب این الگوریتم‌ها و صحت روابط تئوری حاصل شده مورد بررسی قرار گرفته است.

کلیدواژه: الگوریتم تطبیقی، خطای میانگین مربع، روش افزایشی، شبکه توزیع شده.

۱- مقدمه

در سال‌های اخیر، شبکه‌های توزیع شده به عنوان روشی برای ارتباطات و کنترل شبکه‌های کنونی و آینده مورد توجه قرار گرفته‌اند. شبکه‌های توزیع شده روشی کارآمد برای ارتباطات گره‌ها با یکدیگر هستند. این شبکه‌ها 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. از کاربردهای این شبکه‌ها می‌توان به عنوان حسگرهای توزیع شده، دیده‌بانی محیط، فضاهای هوشمند و تمرکز روی هدف نام برد [۱] و [۲]. شبکه‌های توزیع شده نسبت به شبکه‌های متمرکز قدیمی دارای بازده بیشتری هستند و بنابراین توسعه روش‌های کارآمد در این شبکه‌ها می‌تواند مفید باشد. به همین منظور ایده استفاده از

این مقاله در تاریخ ۲۲ آذر ماه ۱۳۹۵ دریافت و در تاریخ ۹ بهمن ماه ۱۳۹۶ بازنگری شد.

محمد شمس اسفندآبادی (نویسنده مسئول)، دانشکده مهندسی برق، دانشگاه تربیت دبیر شهید رجایی، لویزان، تهران ایران، (email: mshams@sru.ac.ir).

علی‌رضا دانایی، دانشکده مهندسی برق، دانشگاه تربیت دبیر شهید رجایی، لویزان، تهران، ایران، (email: ar.danaee@sru.ac.ir).

محمدسعید شفیعی، دانشکده مهندسی برق، دانشگاه تربیت دبیر شهید رجایی، لویزان، تهران، ایران، (email: ms.shafiee@sru.ac.ir).

1. Distributed Least Mean Square
2. Distributed Normalized Least Mean Square
3. Distributed Affine Projection Algorithm
4. Normalized Subband Adaptive Algorithms
5. Distributed Normalized Subband Adaptive Algorithm
6. Distributed Selective Partial Update Normalized Subband Adaptive Algorithm
7. Distributed Dynamic Selection Normalized Subband Adaptive Algorithm

۲- مقدمه‌ای بر الگوریتم NSAF

در الگوریتم فیلتر تطبیقی زیرباند، سیگنال ورودی و پاسخ خروجی مطلوب با عبور از یک بانک فیلتر به زیرباندهایی تقسیم می‌شوند که از لحاظ فرکانسی از هم جدا می‌باشند. این ویژگی SAA باعث می‌شود تا بتوان خواص هر کدام از سیگنال‌های زیرباند را از هم جدا کرد [۱۱]. این عمل باعث همگرایی هر زیرباند به طور مجزا گردیده [۶] و بهبود همگرایی کل را در پی خواهد داشت. شکل ۱ ساختار الگوریتم NSAF را نشان می‌دهد [۸]. در این شکل f_0, f_1, \dots, f_{N-1} بانک فیلتر و $x(n)$ و $d(n)$ بیانگر دنباله‌های اصلی سیگنال ورودی و خروجی مطلوب قبل از عبور از بانک فیلتر است. $x_i(n)$ و $d_{i,D}(n)$ دنباله‌های اصلی عبوری از بانک فیلتر هستند. همچنین باید به این نکته توجه داشت که شاخص D نشان‌دهنده سیگنال زیر نمونه‌برداری شده است به طوری که $d_{i,D}(n)$ بیانگر سیگنال خروجی مطلوب زیر نمونه‌برداری شده است.

رابطه اصلاح وزن الگوریتم NSAF به صورت (۱) ارائه می‌گردد [۸]

$$w(n) = w(n-1) + \mu \sum_{i=0}^{N-1} \frac{x_i(n)}{\|x_i(n)\|^2 + \varepsilon} e_{i,D}(n) \quad (1)$$

که در آن $x_i(n) = [x_i(nN), \dots, x_i(nN - M + 1)]^T$ یک بردار $M \times 1$ از سیگنال زیر نمونه‌برداری شده، $e_{i,D}(n) = d_{i,D}(n) - x_i^T(n)w(n-1)$ سیگنال خطای زیر نمونه‌برداری شده، μ اندازه گام و همچنین ε پارامتر تنظیم برای جلوگیری از تقسیم بر صفر می‌باشد. حال با تعریف ماتریس‌های $X(n) = [x(nN), x(nN-1), \dots, x(nN-(P-1))]^T$ و بردار $d(n) = [d(nN), d(nN-1), \dots, d(nN-(P-1))]^T$ این عبارت $x(nN) = [x(nN), x(nN-1), \dots, x(nN-(M+1))]^T$ است،

رابطه اصلاح وزن برای NSAF به صورت (۲) بیان می‌شود

$$w(n) = w(n-1) + \mu X(n)F^T [eI + \text{diag}\{\text{diag}\{F^T X^T(n)X(n)F\}\}]^{-1} F^T e(n) \quad (2)$$

به طوری که

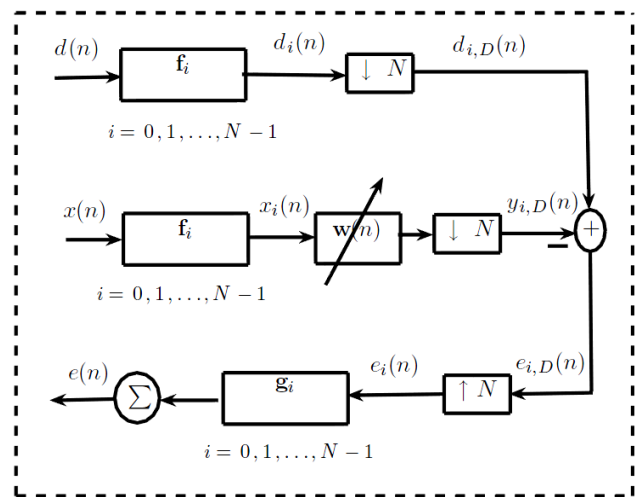
$$e(n) = d(n) - X^T(n)w(n-1) \quad (3)$$

در (۲)، F ماتریس با اندازه $P \times N$ است که ستون‌های آن، پاسخ ضربه کانال فیلترهای مربوط به فیلتر بانک هستند. همچنین N تعداد زیرباندها و P طول فیلتر بانک هستند.

۳- تحقق خانواده توزیع شده DNSAAs

با توجه به ویژگی‌های برجسته خانواده الگوریتم‌های فیلتر تطبیقی زیرباند (NSAAs) مانند سرعت همگرایی بالا، پیچیدگی محاسباتی کم و میانگین مربعات خطای حالت ماندگار کم، این الگوریتم‌ها را در یک شبکه توزیع شده افزایشی به کار می‌بریم. در این بخش خانواده الگوریتم‌های زیرباند توزیع شده را بر اساس یک رابطه جامع پایه‌گذاری می‌کنیم. بر این اساس با انتخاب مناسب ماتریس‌ها و پارامترها می‌توان خانواده الگوریتم‌های زیرباند توزیع شده شامل dNSAA، dSPU-NSAA، dDS-NSAA و dSPU-DS-NSAA را تحقق داد. این روش جدید می‌تواند برای نشان دادن تفاوت‌ها و شباهت‌های الگوریتم‌های نوین توزیع شده تطبیقی و تحلیل عملکرد آنها بر اساس یک رابطه جامع مفید باشد.

رابطه اصلاح وزن برای خانواده الگوریتم‌های نرمالیزه شده زیرباند در



شکل ۱: ساختار الگوریتم NSAF.

ترکیب این دو الگوریتم، الگوریتم زیرباند نرمالیزه شده با انتخاب جزئی ضرایب و انتخاب پویای زیرباندها (dSPU-DS-NSAA) برقرار می‌شود که از لحاظ پیچیدگی محاسباتی، کاهش بیشتری دارد. در ادامه تحقیق به ارائه یک روش واحد برای تحلیل تئوری عملکرد میانگین مربعات برای هر گره پرداخته و با ارائه یک رابطه جامع به بررسی عملکرد حالت گذرا، ماندگار و باند پایداری الگوریتم‌های نوین توزیع شده می‌پردازیم. در نهایت عملکرد مناسب الگوریتم‌های نوین تطبیقی و صحت روابط تئوری حاصل شده به وسیله شبیه‌سازی‌های مختلف در شبکه‌های توزیع شده مورد بررسی قرار گرفته است.

این تحقیق به صورت زیر سازمان‌دهی شده است: در بخش ۲ الگوریتم NSAF را به صورت خلاصه مورد بررسی قرار داده و سپس در بخش بعد به ارائه الگوریتم‌های خانواده زیرباند نرمالیزه شده توزیع شده در یک قالب واحد می‌پردازیم. در بخش ۴ تحلیل میانگین مربع برای هر گره ارائه می‌گردد. باند پایداری و پیچیدگی محاسباتی در بخش ۵ و ۶ مورد بررسی قرار می‌گیرند. در انتهای تحقیق، عملکرد مناسب الگوریتم‌های نوین تطبیقی و صحت روابط تئوری حاصل شده با انجام شبیه‌سازی‌های مختلف در شبکه‌های توزیع شده، بررسی می‌گردد.

علایم ریاضی زیر در این تحقیق مورد استفاده قرار گرفته‌اند:

$\|\cdot\|$: اندازه یک اسکالر

$\|\cdot\|_F$: اندازه اقلیدسی یک بردار

$\text{Tr}(\cdot)$: مجموع عناصر روی قطر اصلی یک ماتریس

$(\cdot)^T$: ترانپوز یک بردار یا ماتریس

$E\{\cdot\}$: امید ریاضی

$A \otimes B$: ضرب کرونیگر ماتریس‌های A و B

$\text{diag}\{\dots\}$: ماتریس قطری با عناصر $\{\dots\}$ روی قطر اصلی و

صفر در بقیه موقعیت‌ها

$\text{vec}(T)$: تبدیل ماتریس T با اندازه $M \times M$ به یک بردار

ستونی t با اندازه $M^T \times 1$ ، با قراردادن پشت سر هم ستون‌های

ماتریس T .

$\text{vec}(t)$: تبدیل یک بردار ستونی t با اندازه $M^T \times 1$ به یک

ماتریس T با اندازه $M \times M$.

جدول ۱: الگوریتم‌های توزیع شده تطبیقی DNSAAS.

الگوریتم	$C_k(n)$	$W_k(n)$
dNSAA	I	$F[\varepsilon I + \text{diag}\{\text{diag}\{F^T X_k^T(n) X_k(n) F\}\}]^{-1} F^T$
dSPU-NSAA	$A_k(n)$	$F[\varepsilon I + \text{diag}\{\text{diag}\{F^T X_k^T(n) A_k(n) X_k(n) F\}\}]^{-1} F^T$
dDS-NSAA	I	$FB_k(n)[\varepsilon I + \text{diag}\{\text{diag}\{B_k^T(n) F^T X_k^T(n) X_k(n) FB_k(n)\}\}]^{-1} B_k^T(n) F^T$
dSPU-DS-NSAA	$A_k(n)$	$FB_k(n)[\varepsilon I + \text{diag}\{\text{diag}\{B_k^T(n) F^T X_k^T(n) A_k(n) X_k(n) FB_k(n)\}\}]^{-1} B_k^T(n) F^T$

L بر روی قطر اصلی است. موقعیت بلوک‌های L بر روی قطر اصلی مشخص می‌کند که کدام یک از ضرایب در هر تکرار در هر گره اصلاح شوند. موقعیت بلوک‌های 1 (N بلوک و $N \leq K$) در قطر اصلی ماتریس $A_k(n)$ برای هر تکرار با انجام مراحل زیر مشخص می‌شوند:

(۱) بردار ورودی $X_k(n)F$ به K بلوک به طول $L \times N$ به صورت $X_k(n)F = [X_{1,k}^T(n), X_{2,k}^T(n), \dots, X_{K,k}^T(n)]^T$ تقسیم‌بندی می‌شود.

(۲) مقدار (۶) برای $1 \leq j \leq K$ محاسبه و مرتب می‌شود

$$\text{Tr}(X_{j,k}^T(n) X_{j,k}(n)) \quad (۶)$$

(۳) با توجه به S مقدار بزرگ‌تر (۶)، موقعیت بلوک‌های 1 مشخص می‌شوند.

ماتریس $B_k(n)$ یک ماتریس قطری $N \times N$ با اعضای 1 و 0 روی قطر اصلی است. موقعیت 1 ها روی قطر اصلی مشخص می‌کند که کدام یک از زیرباند‌ها در رابطه اصلاح وزن انتخاب می‌شوند. با تعریف بردار خطای زیرباند کاهش‌یافته $F^T e_k(n) = [e_{1,k}(n), e_{2,k}(n), \dots, e_{N,k}(n)]^T$ ، موقعیت 1 ها، شاخص‌هایی از i هستند که شرط (۷) را ارضا کنند

$$|e_{i,k}(n)| > \sqrt{\frac{2}{2-\mu}} \sigma_{v,k} \quad (۷)$$

با استفاده از (۴) می‌توان به الگوریتم‌های کلاسیک تطبیقی توزیع شده رسید. با حذف ماتریس F در رابطه‌ها و قراردادن $N=1$ ، الگوریتم توزیع شده تصویر افین [۴] حاصل می‌گردد. برای الگوریتم dAPA باید ماتریس $C_k(n)$ را برابر I و ماتریس $W_k(n)$ را برابر با $[\varepsilon I + X_k^T(n) X_k(n)]^{-1}$ قرار دهیم. در این حالت اگر $P=1$ باشد الگوریتم dNLMS حاصل می‌گردد. همچنین با حذف ماتریس‌های $C_k(n)$ و $W_k(n)$ و قراردادن $N=P=1$ الگوریتم dLMS را خواهیم داشت [۳]. در جدول ۱، الگوریتم‌های کلاسیک تطبیقی توزیع شده بر مبنای (۴) را می‌توان مشاهده نمود.

۴- تحلیل عملکرد خانواده DNSAAS

برای تحلیل عملکرد الگوریتم‌های خانواده زیرباند نرمالیزه شده در شبکه‌های توزیع شده افزایشی، یک روش جامع بر اساس رابطه بقای انرژی که در [۴] برای الگوریتم تطبیقی به کار برده شده است ارائه خواهد شد. جهت تحلیل عملکرد میانگین مربعات در این الگوریتم‌ها، نیاز به بررسی و مطالعه تکامل مکانی-زمانی مقدار $E\{\tilde{h}_k(n)_{\Sigma_k}^2\}$ در هر گره است به طوری که Σ_k یک ماتریس هرمیتین و مثبت معین و $\tilde{h}_k(n)$ بردار خطای وزن است که به صورت $\tilde{h}_k(n) = h_i - h_k(n)$ تعریف شده است. حال می‌توان رابطه جامع اصلاح بردار خطای وزن را به صورت (۸) نوشت

$$\tilde{h}_k(n) = \tilde{h}_{k-1}(n) - \mu_k C_k(n) X_k(n) W_k(n) e_k(n) \quad (۸)$$

با توجه به مدل داده خطی، بردار خطا را می‌توان به صورت (۹) نمایش داد

یک شبکه افزایشی توزیع شده روی J گره به صورت (۴) تعریف می‌شود

$$w(n) = w(n-1) + \sum_{k=1}^J \mu_k C_k(n) X_k(n) W_k(n) e_k(n) \quad (۴)$$

به طوری که

$$e_k(n) = d_k(n) - X_k^T(n) h_{k-1}(n) \quad (۵)$$

بردار خطا و μ_k اندازه گام هر گره است. در (۵)، $X_k(n) = [x_k(nN), \dots, x_k(nN - (P-1))]$ ماتریس سیگنال داده با اندازه $M \times P$ و $d_k(n) = [d_k(nN), d_k(nN-1), \dots, d_k(nN - (P-1))]^T$ بردار داده با اندازه $P \times 1$ هستند به طوری که $x_k(nN) = [x_k(nN), x_k(nN-1), \dots, x_k(nN - (M+1))]^T$ در اینجا مدل داده خطی بین بردار سیستم ناشناخته h_i ، $d_k(n)$ و $X_k(n)$ را به صورت $d_k(n) = X_k^T(n) h_i + v_k(n)$ می‌گیریم به گونه‌ای که $v_k(n)$ یک رشته نویز تصادفی و مستقل مکانی با واریانس $\sigma_{v,k}^2$ در هر گره است. با انتخاب ماتریس $C_k(n)$ و $W_k(n)$ از جدول ۱ می‌توان DNSAAS را به خوبی الگوریتم‌های تطبیقی کلاسیک توزیع شده تحقق داد.

شکل ۲ الگوریتم یادگیری DNSAAS را بر اساس رابطه جامع در یک شبکه توزیع شده افزایشی نشان می‌دهد. برای تخمین محلی در هر لحظه زمانی n ، بر اساس داده محلی $(X_k(n)$ و $d_k(n)$) و وزن اصلاح شده $(h_{k-1}(n))$ از گره قبلی $k-1$ در چرخه، اصلاح صورت می‌گیرد. در نهایت تخمین محلی $h_i(n)$ برای تخمین سراسری $w(n)$ استفاده می‌شود. این مقدار برای تخمین محلی اولیه $h_i(n+1)$ در لحظه زمانی بعد $n+1$ به کار می‌رود.

در DNSAA داده محلی در هر گره به باندهایی تقسیم می‌شود و سپس روش تخمین تطبیقی زیرباند نرمالیزه شده طی هر تکرار در یک شبکه افزایشی انجام می‌شود. در dSPU-NSAA ضرایب به صورت انتخاب جزئی بلوک‌ها برای هر گره در هر زیرباند اصلاح می‌شوند. این الگوریتم دارای عملکرد نزدیکی به DNSAA است در حالی که پیچیدگی محاسباتی را نیز پایین می‌آورد. در dDS-NSAA در هنگام اصلاح وزن تعدادی از زیرباند‌ها به صورت پویا انتخاب می‌شود. کاهش تعداد زیرباند‌ها در هنگام اصلاح وزن به کاهش پیچیدگی محاسباتی در dDS-NSAA می‌انجامد. با ترکیب روش SPU و DS الگوریتم dSPU-DS-NSAA معرفی می‌شود که در هر گره در حین بهینه‌سازی، ضرایب به صورت جزئی و زیرباند‌ها به صورت بهینه انتخاب می‌شوند. برای برقراری روش جامع رابطه اصلاح وزن خانواده زیرباند، ماتریس‌های $A_k(n)$ و $B_k(n)$ معرفی می‌شوند. برای انتخاب جزئی ضرایب در dSPU-NSAA از ماتریس $A_k(n)$ و برای انتخاب پویای زیرباند‌ها در dDS-NSAA از ماتریس $B_k(n)$ استفاده می‌گردد. در الگوریتم dSPU-DS-NSAA بدیهی است که به علت انتخاب جزئی ضرایب و همچنین انتخاب پویای زیرباند‌ها باید از هر دو ماتریس $A_k(n)$ و $B_k(n)$ استفاده نمود. ماتریس $A_k(n)$ یک ماتریس قطری $M \times M$ شامل بلوک‌های 0 و 1 به طول

$$\begin{aligned} \sigma_k' &= \sigma_k - \mu_k (E\{Z_k^T(n)\} \otimes I) \cdot \sigma_k - \\ &\mu_k (I \otimes E\{Z_k^T(n)\}) \cdot \sigma_k + \\ &\mu_k^\gamma (E\{Z_k^T(n)\} \otimes E\{Z_k^T(n)\}) \cdot \sigma_k \end{aligned} \quad (15)$$

به طوری که $\sigma_k = \text{vec}(\Sigma_k)$ و $\sigma_k' = \text{vec}(\Sigma_k')$ با تعریف ماتریس G_k با اندازه $M^r \times M^r$ به صورت

$$G_k = I - \mu_k (E\{Z_k^T(n)\} \otimes I) - \mu_k (I \otimes E\{Z_k^T(n)\}) + \mu_k^\gamma (E\{Z_k^T(n)\} \otimes E\{Z_k^T(n)\}) \quad (16)$$

رابطه (۱۵) را به صورت زیر می‌توان بازنویسی کرد

$$\sigma_k' = G_k \cdot \sigma_k \quad (17)$$

همچنین با تعریف γ_k به صورت رابطه زیر

$$\gamma_k = \text{vec}(E\{C_k(n)X_k(n)W_k(n)W_k^T(n)X_k^T(n)C_k^T(n)\}) \quad (18)$$

جمله دوم سمت راست (۱۴) به صورت (۱۹) بیان می‌شود

$$\text{Tr}(E\{Y_k(n)\}) = \gamma_k^T \cdot \sigma_k \quad (19)$$

با توجه به روابط اخیر، (۱۴) را می‌توان به شکل رابطه زیر به دست آورد

$$E\{\|\tilde{h}_k(n)\|_{\sigma_k}^2\} = E\{\|\tilde{h}_{k-1}(n)\|_{G_k \cdot \sigma_k}^2\} + \mu_k^\gamma \sigma_{v,k}^\gamma \gamma_k^T \sigma_k \quad (20)$$

این رابطه تنها شامل اطلاعات مکانی محلی بین دو گره است اما برای تحلیل عملکرد کل شبکه به گردش انرژی در کل گره‌ها نیاز است. استفاده از روش ارائه‌شده در [۴] و در نظر گرفتن رابطه خطی به صورت $\sigma_{k-1} = G_k \cdot \sigma_k$ ، رابطه زیر معرفی می‌شود

$$E\{\|\tilde{h}_k(n)\|_{\sigma_k}^2\} = E\{\|\tilde{h}_{k-1}(n)\|_{\sigma_{k-1}}^2\} + g_k \sigma_k \quad (21)$$

به طوری که $g_k = \mu_k^\gamma \sigma_{v,k}^\gamma \gamma_k^T$ است. حال با به کار بردن روش تکراری در (۲۱) می‌توان رابطه $E\{\|\tilde{h}_{k-1}(n)\|_{\sigma_{k-1}}^2\}$ در گره k را به صورت (۲۲) به دست آورد

$$\begin{aligned} E\{\|\tilde{h}_{k-1}(n)\|_{\sigma_{k-1}}^2\} &= E\{\|\tilde{h}_{k-1}(-1)\|_{(\Pi_{k-1,1})^n \sigma_{k-1}}^2\} + \\ &\alpha_{k-1} (I + \dots + (\Pi_{k-1,1})^n) \sigma_{k-1} \end{aligned} \quad (22)$$

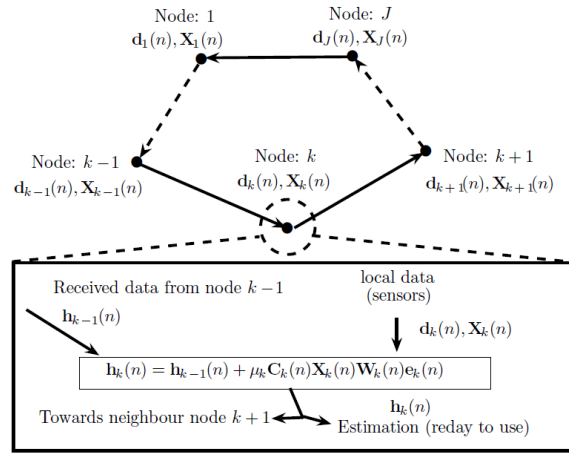
و مقادیر به شکل $\Pi_{k-1,l} = G_{k+l} \cdot G_{k+l-1} \dots G_{k+1} \cdot G_{k-1}$ ، $l = 1, 2, \dots, J$ و $\alpha_{k-1} = g_k \Pi_{k-1,2} + g_{k+1} \Pi_{k-1,3} + \dots + g_{k-2} \Pi_{k-1,J} + g_{k-1}$ می‌باشد. با استفاده از این روابط بازگشتی می‌توانیم عملکرد حالت گذرای الگوریتم‌های dNSAAs را ارزیابی کنیم. وقتی که $\sigma_{k-1} = \text{vec}(I)$ ، عملکرد حالت گذرا به صورت تئوری برای MSD در گره k به دست می‌آید. برای $\sigma_{k-1} = \text{vec}(R_k)$ ، مقادیر EMSE به صورت تئوری تحقق می‌یابد. همچنین وقتی $n \rightarrow \infty$ ، (۲۳) را به دست می‌آوریم

$$E\{\|\tilde{h}_k(\infty)\|_{\sigma_k}^2\} = E\{\|\tilde{h}_{k-1}(\infty)\|_{G_k \cdot \sigma_k}^2\} + \mu_k^\gamma \sigma_{v,k}^\gamma \gamma_k^T \sigma_k \quad (23)$$

حال با تکرار (۲۳) برای شبکه افزایشی و با استفاده از تعریف $\Pi_{k-1,1}$ و α_{k-1} ، رابطه زیر برای توصیف رفتار حالت ماندگار این الگوریتم‌ها حاصل می‌گردد

$$E\{\|\tilde{h}_{k-1}(\infty)\|_{(I - \Pi_{k-1,1}) \sigma_{k-1}}^2\} = \alpha_{k-1} \sigma_{k-1} \quad (24)$$

اگر $(I - \Pi_{k-1,1}) \sigma_{k-1} = \text{vec}(I)$ و $(I - \Pi_{k-1,1}) \sigma_{k-1} = \text{vec}(R_k)$ باشند روابط حالت ماندگار MSD و EMSE در گره k نیز به همین ترتیب بیان



شکل ۲: پردازش داده‌ها در یک شبکه توزیع‌شده با استفاده از الگوریتم جامع.

$$e_k(n) = X_k^T(n) \tilde{h}_{k-1}(n) + v_k(n) \quad (9)$$

با جایگذاری (۹) در (۸) و گرفتن نرم وزن دار از دو طرف (۸)، (۱۰) به دست می‌آید

$$\begin{aligned} \|\tilde{h}_k(n)\|_{\Sigma_k}^2 &= \|\tilde{h}_{k-1}(n)\|_{\Sigma_k}^2 + \mu_k^\gamma v_k^T(n) Y_k(n) v_k(n) + \\ &\{\text{Cross terms involving one instance of } v_k(n)\} \end{aligned} \quad (10)$$

به طوری که

$$\begin{aligned} \Sigma_k' &= \Sigma_k - \mu_k \Sigma_k Z_k(n) - \mu_k Z_k^T(n) \Sigma_k + \\ &\mu_k^\gamma Z_k^T(n) \Sigma_k Z_k(n) \end{aligned} \quad (11)$$

که در رابطه بالا $Z_k(n) = C_k(n) X_k(n) W_k(n) X_k^T(n)$ و $Y_k(n) = W_k^T(n) X_k^T(n) C_k^T(n) \Sigma_k C_k(n) X_k(n) W_k(n)$ گرفتن امید ریاضی از دو طرف (۱۰) خواهیم داشت

$$\begin{aligned} E\{\|\tilde{h}_k(n)\|_{\Sigma_k}^2\} &= E\{\|\tilde{h}_{k-1}(n)\|_{\Sigma_k}^2\} + \\ &\mu_k^\gamma E\{v_k^T(n) Y_k(n) v_k(n)\} \end{aligned} \quad (12)$$

رابطه (۱۲) به رابطه بازگشتی زمانی- مکانی خطای وزن معروف است. برای ساده‌کردن (۱۲) نیاز به استفاده از فرضیات استقلال زیر داریم [۴] و [۱۲]:

(۱) ماتریس $X_k(n)$ از $X_{k-1}(n)$ مستقل است. این فرض تضمین می‌کند که $\tilde{h}_{k-1}(n)$ از هر دوی Σ_k' و $X_k(n)$ مستقل است.
(۲) $\tilde{h}_{k-1}(n)$ از $Z_k(n)$ مستقل است.
با استفاده از این فرضیات می‌توان ماتریس Σ_k' را به وسیله (۱۳) جایگزین کرد

$$\begin{aligned} \Sigma_k' &= \Sigma_k - \mu_k \Sigma_k E\{Z_k(n)\} - \mu_k E\{Z_k^T(n)\} \Sigma_k + \\ &\mu_k^\gamma E\{Z_k^T(n) \Sigma_k Z_k(n)\} \end{aligned} \quad (13)$$

از آنجا که $E\{v_k(n) v_k^T(n)\} = \sigma_{v,k}^\gamma I$ ، می‌توان (۱۲) را به صورت رابطه زیر بازنویسی کرد

$$\begin{aligned} E\{\|\tilde{h}_k(n)\|_{\Sigma_k}^2\} &= E\{\|\tilde{h}_{k-1}(n)\|_{\Sigma_k}^2\} + \\ &\mu_k^\gamma \sigma_{v,k}^\gamma \text{Tr}(E\{Y_k(n)\}) \end{aligned} \quad (14)$$

اکنون برای ساده‌سازی با به کارگیری عملگر $\text{vec}(\cdot)$ در دو طرف (۱۳) و استفاده از $\text{vec}(P\Sigma Q) = (Q^T \otimes P) \text{vec}(\Sigma)$ ، (۱۵) را به دست می‌آوریم

جدول ۲: مقایسه پیچیدگی محاسباتی الگوریتم‌های مختلف توزیع شده افزایشی تطبیقی در یک تکرار در یک گره برای داده حقیقی.

الگوریتم	تعداد ضرب‌کننده‌ها	تعداد مقایسه‌کننده‌ها
dNLMS	$3M + 1$	-
dSPU-NLMS	$3NL + 1$	$K \log_r N + O(K)$
dRLS	$M^2 + 5M + 1$	-
dAPA	$(P^2 + 2P)M + P^2 + P$	-
dSPU-APA	$(P^2 + 2P)NL + P^2 + P + 1$	$K \log_r N + O(K)$
dSR-APA	$(Q^2 + 2Q)M + Q^2 + Q^2 + (P - Q)M + P + 1$	$P \log_r N + O(P)$
dDS-APA	$(Q^2(N) + 2Q(n))M + Q^2(n) + Q^2(n)$	P
dSPU-SR-APA	$(Q^2 + 2Q)NL + Q^2 + Q^2 + (P - Q)NL + P + 1$	$K \log_r N + O(K) + P \log_r N + O(P)$
dSPU-DS-APA	$(Q^2(N) + 2Q(n))NL + Q^2(n) + Q^2(n)$	$K \log_r N + O(K) + P$
dNSAA	$3M + 3NP + 1$	-
dSPU-NSAA	$3M + SL + 3NP + 1$	$K \log_r S + O(K)$
dDS-NSAA	$(1 + 2 \frac{N(k)}{N})M + 3NP + N$	N
dSPU-DS-NSAA	$M + (M + SL) \frac{N(k)}{N} + 3NP + N$	$K \log_r S + O(K) + N$

$$H_k = \begin{bmatrix} \frac{1}{2} M_k & -\frac{1}{2} N_k \\ I & . \end{bmatrix} \quad (31)$$

۶- پیچیدگی محاسباتی

جدول ۲ تعداد ضرب‌کننده‌ها و مقایسه‌کننده‌ها را در یک تکرار در یک گره به ازای الگوریتم‌های متفاوت توزیع شده افزایشی نشان می‌دهد. برای پیچیدگی محاسباتی dNLMS، dRLS و dAPA از [۴] استفاده شده و برای dAPA پیچیدگی به تعداد رگرورها (P) و تعداد ضرایب فیلتر (M) بستگی دارد. پیچیدگی محاسباتی برای dRLS، $M^2 + 5M + 1$ ضرب‌کننده است که در مقادیر بالای M بسیار زیاد می‌گردد. در dNSAA پیچیدگی محاسباتی به طول ضرایب فیلتر (M)، تعداد زیرباندها (N) و طول فیلتر بانک (P) بستگی دارد. پیچیدگی محاسباتی برای dNSAA تعداد $3M + 3NP + 1$ ضرب‌کننده است [۹]. با توجه به جدول ۲ مشاهده می‌شود که تعداد ضرب‌کننده‌ها در dNSAA کمتر از dRLS و dAPA است. از [۹] پیچیدگی محاسباتی برای dSPU-NSAA تعداد $3M + SL + 3NP + 2$ ضرب‌کننده و $K \log_r S + O(K)$ مقایسه‌کننده به دست آمده است. در مقایسه با dNSAA کاهش تعداد ضرب‌کننده‌ها $M - SL$ است که برای مقادیر بزرگ M بسیار بالا است. در DS-NSAF به $(1 + 2N(k)/N)M + 3NP + N$ ضرب‌کننده و N مقایسه‌کننده نیاز است. با توجه به انتخاب تعدادی از زیرباندها در dDS-NSAA، تعداد ضرب‌کننده‌ها در حین اصلاح وزن کاهش می‌یابد. با ترکیب SPU و DS می‌توان تعداد ضرب‌کننده‌ها را به میزانی بیشتر کاهش داد. این مقدار برای dSPU-DS-NSAA برابر $M + (M + SL)N(k)/N + 3NP + N$ است و این الگوریتم همچنین به $K \log_r S + O(K) + N$ مقایسه‌کننده نیز نیاز دارد.

۷- نتایج شبیه‌سازی

در این بخش، عملکرد dNSAAs را برای یک شبکه افزایشی با $J = 20$ گره برای کاربرد شناسایی سیستم بررسی می‌کنیم. پاسخ ضربه سیستم ناشناخته به صورت $h_i = 1/\sqrt{M} [1, 1, \dots, 1]^T$ که $M = 16$ می‌باشد انتخاب شده است. سیگنال ورودی رنگی با عبور دادن فرایند

می‌شوند. با انجام این کار، نتایج نهایی برای حالت ماندگار به صورت (۲۵) و (۲۶) درمی‌آیند

$$MSD_k = \alpha_{k-1} (I - \Pi_{k-1,1})^{-1} \text{vec}(I) \quad (25)$$

$$EMSE_k = \alpha_{k-1} (I - \Pi_{k-1,1})^{-1} \text{vec}(R_k) \quad (26)$$

همچنین با توجه به (۹) برای مقادیر MSE در گره k خواهیم داشت

$$MSE_k = EMSE_k + \sigma_{v,k}^2 \quad (27)$$

در اینجا یادآور می‌شویم که با توجه به بخش ۳، روابط تئوری به دست آمده را می‌توان برای سایر الگوریتم‌های کلاسیک مانند dLMS، dNLMS و dAPA در شبکه‌های افزایشی به کار برد.

۵- باند پایداری میانگین و میانگین مربع خانواده DNSAAs

با جایگذاری (۹) در (۸) و گرفتن امید ریاضی از دو طرف (۸)، (۲۸) به دست می‌آید

$$E\{\tilde{h}_k(n)\} = [I - \mu_k E\{Z_k(n)\}] E\{\tilde{h}_{k-1}(n)\} \quad (28)$$

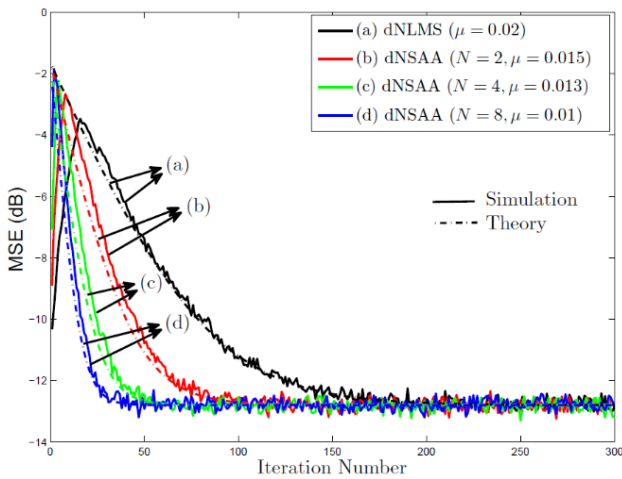
از (۲۸) می‌توان نتیجه گرفت که شرط همگرایی در متوسط الگوریتم‌های dNSAAs از (۲۰) به دست می‌آید

$$\mu_k < \frac{2}{\lambda_{\max} E\{Z_k(n)\}} \quad (29)$$

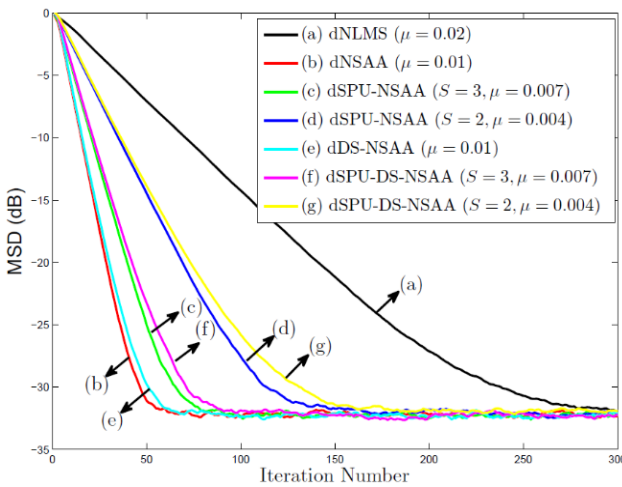
معادله عمومی اصلاح وزن (۲۰) در صورتی پایدار است که ماتریس G_k پایدار باشد. با توجه به (۱۶) می‌دانیم $G_k = I - \mu_k M_k + \mu_k^2 N_k$ که در این رابطه M_k برابر است با $(I \otimes E\{Z_k^T(n)\}) - (I \otimes E\{Z_k(n)\}) \otimes I$ و برای N_k داریم $N_k = E\{Z_k^T(n)\} \otimes E\{Z_k(n)\}$. با راه حلی مشابه در [۱۲]، شرطی که اندازه گام μ_k را برای همگرایی میانگین مربعات dNSAAs تضمین می‌کند از (۳۰) به دست می‌آید

$$0 < \mu_k < \min\left\{\frac{1}{\lambda_{\max} M_k^{-1} N_k}, \frac{1}{\lambda_{\max}(H_k)} \in \mathfrak{R}^+\right\} \quad (30)$$

به طوری که



شکل ۵: نمودار سراسری شبیه‌سازی و تئوری MSE برای dNLMS و dNSAA.



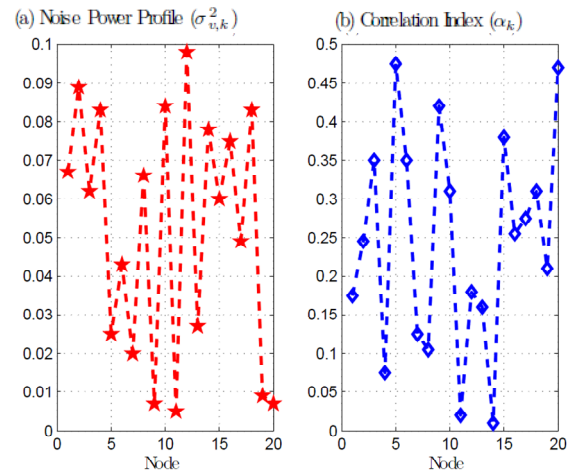
شکل ۶: نمودار شبیه‌سازی MSD سراسری برای dNSAAs.

dNLMS را برابر 0.02 قرار داده و اندازه گام بقیه الگوریتم‌ها را طوری تنظیم می‌کنیم که خطای ماندگار یکسانی داشته باشند. نتایج نشان می‌دهد که dNSAA با تعداد زیرباندهای مختلف نسبت به dNLMS سرعت همگرایی بیشتری دارد. با افزایش تعداد زیرباندها (N)، سرعت همگرایی نیز افزایش می‌یابد. منحنی تئوری بر اساس (۲۲) به دست آمده و همان طور که نتایج نشان می‌دهد منحنی تئوری تطابق خوبی با نتایج شبیه‌سازی دارد که این بیانگر صحت روابط تئوری ارائه شده است.

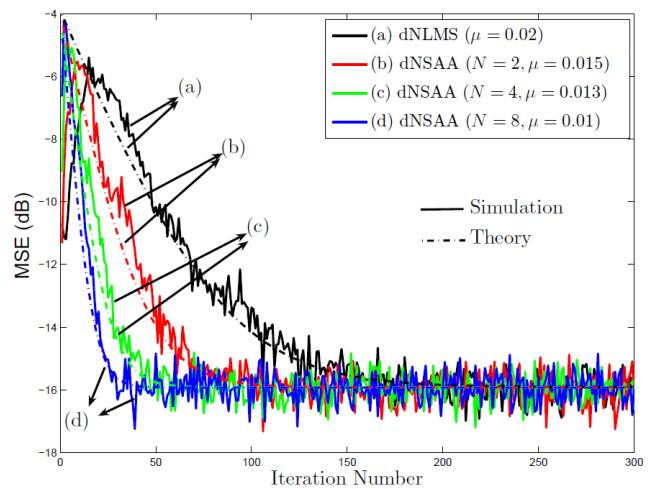
شکل ۵ نمودارهای سراسری شبیه‌سازی شده و تئوری MSE^2 را برای dNLMS و dNSAA به ازای N های مختلف نشان می‌دهد. نمودار یادگیری سراسری MSE با استفاده از میانگین‌گیری از نمودار MSE همه گره‌ها به دست آمده است.

نتایج نشان می‌دهد که dNSAA با تعداد زیرباندهای مختلف سرعت همگرایی بالاتری نسبت به dNLMS دارد و این بیانگر عملکرد بهتر dNSAA در مقایسه با dNLMS در برابر ورودی‌های رنگی است. با افزایش تعداد زیرباندها (N)، سرعت همگرایی نیز افزایش می‌یابد. بار دیگر تطابق مناسبی میان منحنی‌های شبیه‌سازی شده و تئوری مشاهده می‌گردد.

برای مقایسه سایر الگوریتم‌های dNSAAs از منحنی سراسری MSD^۳ در شکل ۶ استفاده شده است. پارامترهای N و K به ترتیب ۸



شکل ۷: مقادیر (راست) میزان همبستگی و (چپ) واریانس نویز افزودنی در هر گره در شبکه افزایشی.



شکل ۸: نمودار شبیه‌سازی و تئوری MSE برای dNLMS و dNSAA در گره ۵.

گوسی سفید با واریانس واحد از یک فیلتر AR^1 مرتبه اول با تابع تبدیل $\alpha_k \in [0, 0.5]$ و $\sqrt{1-\alpha_k^2}/1-\alpha_k z^{-1}$ تولید شده است. رشته نویز افزودنی برای هر گره، یک فرایند گوسی سفید با واریانس $\sigma_{v,k}^2 \in (0, 1]$ است.

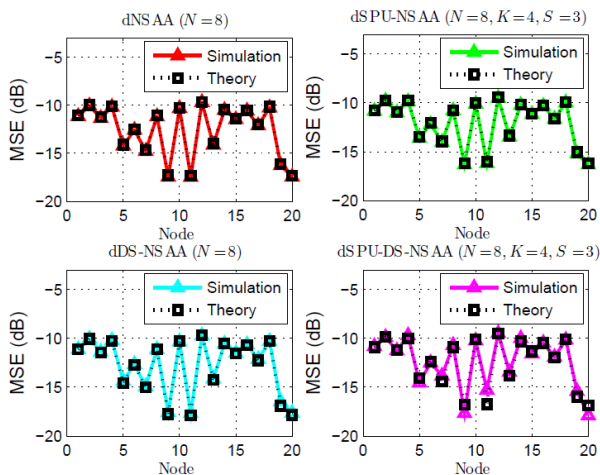
شکل ۳ مشخصات هر گره شامل α_k و $\sigma_{v,k}^2$ را نشان می‌دهد. همان طور که می‌بینیم گره ۵ دارای ورودی با بیشینه رنگی بودن است و بنابراین ما نتایج حالت گذرا را برای این گره نشان می‌دهیم. در تمام شبیه‌سازی‌ها نمودار یادگیری خطاهای MSD و MSE با میانگین‌گیری از ۲۰۰ آزمایش مستقل به دست آمده است. همچنین مقادیر حالت ماندگار MSE و MSD با میانگین‌گیری از هزار نمونه پایدار و ۱۰۰ آزمایش مستقل به دست آمده‌اند. اندازه گام در هر نمودار یادگیری طوری انتخاب شده است تا خطای حالت ماندگار برای هر نمودار یکسان باشد. برای بانک فیلتر در dNSAAs از فیلتر بانک ELT [۱۳] و [۱۴] استفاده شده و همچنین پارامتر تنظیم ε مقدار 0.001 را داراست.

۷-۱ نتایج شبیه‌سازی برای عملکرد حالت گذرا

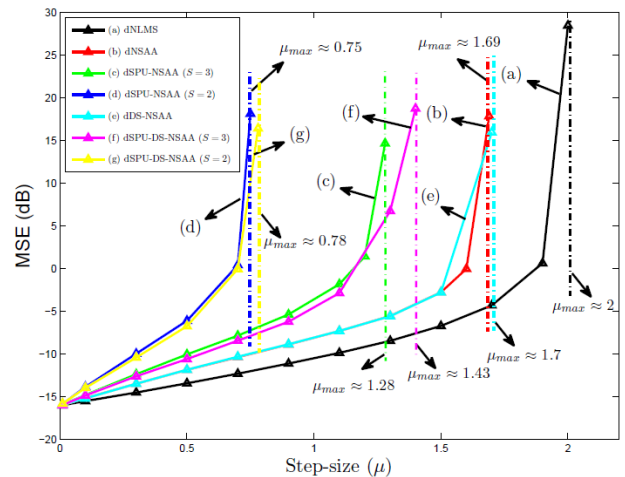
شکل ۴ نمودار شبیه‌سازی شده و تئوری MSE را برای dNLMS و dNSAA به ازای N های مختلف در گره ۵ نشان می‌دهد. شایان ذکر است که برای داشتن یک مقایسه منصفانه، اندازه گام برای الگوریتم

2. Mean Square Error
3. Mean Square Deviation

1. Autoregressive



شکل ۸: مقادیر خطای حالت ماندگار MSE همه گروه‌ها با اندازه گام ۰٫۲ برای dNSAAs



شکل ۹: مقادیر شبیه‌سازی شده حالت ماندگار MSE برای dNSAAs بر حسب اندازه گام در گروه ۵

جدول ۳: باند پایداری اندازه گام برای dNSAAs در گروه ۵

الگوریتم	$\frac{2}{\lambda_{\max} E\{Z_{\delta}(n)\}}$	$\frac{1}{\lambda_{\max} M_{\delta}^{-1} N_{\delta}}$	$\frac{1}{\lambda_{\max}(H_{\delta})} \in \mathcal{R}^+$	μ_{\max}
dNLMS	۱۲٫۳۱۷۹	۲٫۰۰۰۲	۶٫۵۶۳۹	۲٫۰۰۰۲
dNSAA ($N=2$)	۱۲٫۱۰۰۳	۱٫۹۸۶۷	۶٫۳۱۸۶	۱٫۹۸۶۷
dNSAA ($N=4$)	۶٫۹۴۰۷	۱٫۹۲۴۵	۴٫۱۱۴۵	۱٫۹۲۴۵
dNSAA ($N=8$)	۳٫۴۶۴۱	۱٫۶۹۶۴	۲٫۷۰۴۰	۱٫۶۹۶۴
dSPU-NSAA ($N=8, K=4, S=3$)	۳٫۵۹۴۳	۱٫۲۸۳۳	۲٫۵۹۱۹	۱٫۲۸۳۳
dSPU-NSAA ($N=8, K=4, S=2$)	۳٫۴۹۷۱	۰٫۷۵۰۱	۲٫۰۹۰۱	۰٫۷۵۰۱
dSPU-NSAA ($N=8, K=4, S=1$)	۳٫۰۳۰۸	۰٫۲۳۵۶	۱٫۰۱۶۹	۰٫۲۳۵۶
dDS-NSAA ($N=2$)	۱۵٫۰۱۴۵	۱٫۹۸۹۱	۷٫۶۵۳۶	۱٫۹۸۹۱
dDS-NSAA ($N=4$)	۱۱٫۹۳۸۳	۱٫۹۲۳۹	۷٫۰۷۴۶	۱٫۹۲۳۹
dDS-NSAA ($N=8$)	۴٫۸۳۴۰	۱٫۷۰۵۴	۴٫۹۹۶۵	۱٫۷۰۵۴
dSPU-DS-NSAA ($N=8, K=4, S=3$)	۸٫۷۶۴۷	۱٫۴۳۹۰	۶٫۱۶۳۶	۱٫۴۳۹۰
dSPU-DS-NSAA ($N=8, K=4, S=2$)	۴٫۶۶۹۲	۰٫۷۸۹۹	۲٫۷۸۰۵	۰٫۷۸۹۹
dSPU-DS-NSAA ($N=8, K=4, S=1$)	۳٫۳۴۱۵	۰٫۲۳۹۸	۱٫۱۳۱۸	۰٫۲۳۹۸

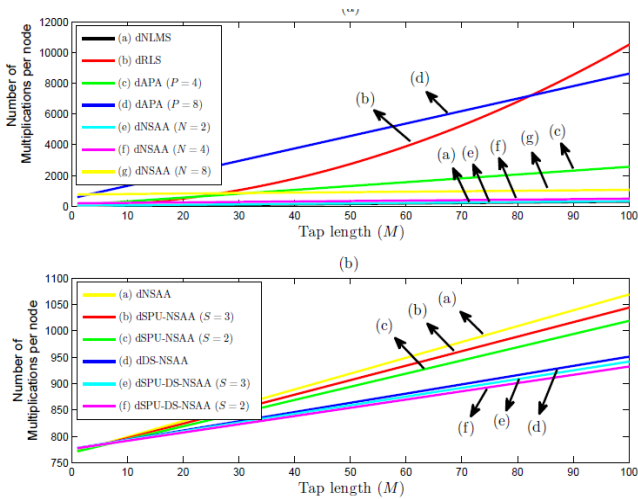
و ۴ انتخاب شده‌اند. این شکل نشان می‌دهد که dDS-NSAA عملکرد نزدیکی از لحاظ سرعت همگرایی نسبت به dNSAA دارد در حالی که پیچیدگی محاسباتی dDS-NSAA از dNSAA بسیار کمتر است. این نتیجه را در مورد dSPU-NSAA به خصوص با $S=3$ نیز می‌توان مشاهده کرد. همچنین با توجه به شکل، مشخص است که منحنی MSD برای dSPU-DS-NSAA با $S=2,3$ عملکرد بسیار نزدیکی به dSPU-NSAA دارد. ترکیب SPU و DS باعث می‌شود که پیچیدگی محاسباتی به میزان بسیاری کاهش یابد. این شکل نشان می‌دهد که dSPU-DS-NSAA با $S=3$ دارای عملکرد نزدیکی به dNSAA است. همان‌طور که در شکل مشاهده می‌شود منحنی‌های تئوری تطابق خوبی با نتایج شبیه‌سازی شده دارند.

۳-۷ نتایج شبیه‌سازی حالت ماندگار

شکل ۸ مقادیر شبیه‌سازی شده و تئوری حالت ماندگار MSE را برای dNSAAs با اندازه گام ۰٫۲ در هر گروه نشان می‌دهد. با توجه به شکل مشاهده می‌شود که dDS-NSAA در حالت ماندگار عملکرد نزدیکی به dNSAA دارد در حالی که پیچیدگی محاسباتی dDS-NSAA از dNSAA بسیار کمتر است. این نتیجه را در مورد dSPU-NSAA به خصوص با $S=3$ نیز می‌توان مشاهده کرد. در dSPU-DS-NSAA

۲-۷ نتایج شبیه‌سازی برای باند پایداری

جدول ۳ مقدار اندازه گام بیشینه را برای dNSAAs به ازای پارامترهای مختلف در گروه ۵ نشان می‌دهد. مقادیر تئوری از (۲۹) و (۳۰) به دست آمده‌اند. برای مقایسه باند پایداری الگوریتم‌های مختلف dNSAAs مقادیر متفاوتی برای N انتخاب کردیم. همان‌طور که مشاهده می‌گردد با افزایش تعداد زیرباندها، باند پایداری کمتر می‌شود. در dSPU-NSAA با انتخاب مقادیر کمتری از S ، باند پایداری نیز کمتر می‌شود. الگوریتم



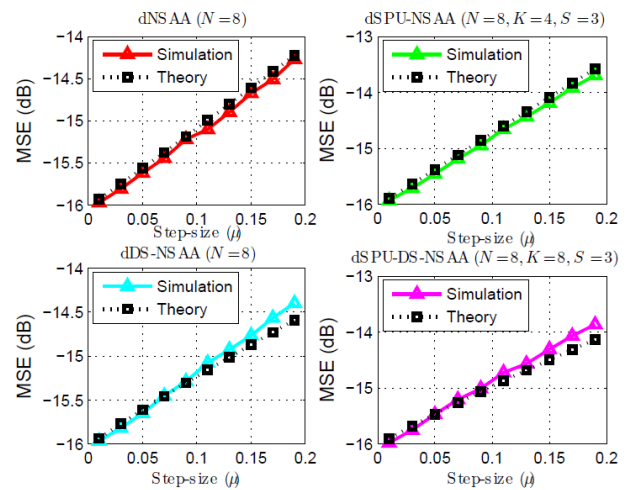
شکل ۱۱: تعداد ضرب‌کننده‌ها در یک گره بر حسب طول فیلتر (بالا) مقایسه dNSAA با الگوریتم‌های کلاسیک تطبیقی (پایین) مقایسه خانواده dNSAA با یکدیگر.

افزایش تکرار، تعداد زیرباند‌های انتخابی به جای تمام زیرباند‌ها ($N=8$) از ۵ زیرباند به ۲ زیرباند می‌رسد که این بیانگر کاهش پیچیدگی محاسباتی در dDS-NSAA است. همچنین در الگوریتم dSPU-DS-NSAA نیز این روند کاهش تعداد زیرباند‌ها که به کاهش پیچیدگی محاسباتی می‌انجامد مشاهده می‌گردد. در این شبیه‌سازی اندازه گام برابر با ۰.۰۱ قرار داده شده است.

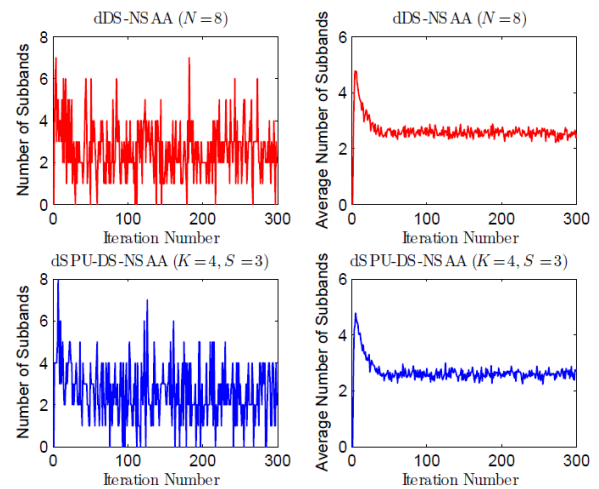
در انتها تعداد ضرب‌کننده‌ها را در یک تکرار برای یک گره به ازای طول فیلتر (M) برای الگوریتم‌های dNLMS، dRLS، dAPA و dNSAA در شکل آورده‌ایم. در شکل (بالا) مشاهده می‌شود که تعداد ضرب‌کننده‌ها در dNLMS نسبت به سایر الگوریتم‌های توزیع‌شده کمتر است. تعداد ضرب‌کننده‌ها با افزایش M در dNSAA با تعداد زیرباند ۲ با dNLMS تقریباً یکسان است. با افزایش تعداد زیرباند‌ها که باعث سرعت همگرایی بیشتر نسبت به dNLMS می‌شود، تعداد ضرب‌کننده‌ها همچنان به dNLMS تقریباً نزدیک است. با افزایش M در dAPA با $P=4$ تعداد ضرب‌کننده‌ها از dNSAA بیشتر است. با افزایش P تعداد ضرب‌کننده‌ها در dAPA بیشتر می‌شود که به دلیل وابستگی تعداد ضرب‌کننده‌ها در این الگوریتم نسبت به P است. همچنین تعداد ضرب‌کننده‌ها در dRLS که با نسبت درجه دو از M افزایش می‌یابد، باعث پیچیدگی محاسباتی بالای این الگوریتم با افزایش M نسبت به دیگر الگوریتم‌های توزیع‌شده تطبیقی می‌گردد. در شکل (پایین) تعداد ضرب‌کننده‌ها در dNSAA نشان داده شده است. با توجه به شکل مشاهده می‌شود که با اعمال روش‌های SPU و DS تعداد ضرب‌کننده‌ها در dNSAA کمتر می‌شود. در dSPU-NSAA با کاهش S تعداد ضرب‌کننده‌ها نسبت به dNSAA کمتر شده است. برای نمودارهای dDS-NSAA و dSPU-DS-NSAA تعداد زیرباند‌های انتخابی با توجه به شکل برابر $N(k)=2$ در نظر گرفته شده است. با توجه به شکل مشاهده می‌شود که با افزایش M تعداد ضرب‌کننده‌ها در نمودارهای dDS-NSAA و dSPU-DS-NSAA نسبت به dNSAA کاهش می‌یابد در حالی که عملکرد سرعت همگرایی و خطای حالت ماندگار این الگوریتم‌ها به dNSAA نزدیک است.

۸- نتیجه‌گیری

در این تحقیق، الگوریتم‌های تطبیقی خانواده زیرباند نرمالیزه‌شده در شبکه توزیع‌شده افزایشی (dNSAAs) ارائه و رابطه جامعی برای تحقق



شکل ۹: نمودار خطای حالت ماندگار MSE بر حسب اندازه گام در گره ۵ برای dNSAAs.



شکل ۱۰: تعداد زیرباند‌های انتخاب‌شده در یک تکرار و میانگین تعداد زیرباند‌های انتخابی برای گره ۵.

با $S=3$ نیز خطای حالت ماندگار برای هر گره به dSPU-NSAA بسیار نزدیک است. همچنین با توجه به شکل می‌توان نتیجه گرفت که dSPU-DS-NSAA با $S=3$ دارای عملکرد نزدیکی به dNSAA در حالت ماندگار است. منحنی‌های تئوری رسم‌شده از (۲۷) به دست آمده‌اند. با توجه به نتایج مشاهده می‌شود که مقادیر تئوری با نتایج شبیه‌سازی تطابق خوبی دارند و این صحت روابط تئوری به دست آمده در بخش ۴ را نشان می‌دهد. شکل ۹ مقدار حالت ماندگار MSE را بر حسب اندازه گام برای ۵ گره نشان می‌دهد. اندازه گام از ۰.۰۱ تا ۰.۲ تغییر می‌کند که در بازه باند پایداری الگوریتم‌های مورد نظر است. با افزایش اندازه گام خطای حالت ماندگار dDS-NSAA نسبت به dNSAA نزدیک می‌ماند. همچنین dSPU-NSAA دارای عملکرد نزدیکی نسبت به dNSAA است. در dSPU-NSAA خطای حالت ماندگار با افزایش اندازه گام نسبت به dSPU-NSAA یکسان است. تطابق مناسبی میان مقادیر تئوری و نمودارهای شبیه‌سازی شده می‌توان مشاهده کرد که با افزایش اندازه گام، این تطابق کمتر می‌شود.

۷-۴ نتایج شبیه‌سازی برای پیچیدگی محاسباتی

شکل ۱۰ تعداد زیرباند‌های انتخاب‌شده در یک تکرار و میانگین تعداد زیرباند‌های انتخابی را در هنگام اصلاح وزن برای dDS-NSAA و dSPU-DS-NSAA در گره ۵ نشان می‌دهد. در dDS-NSAA

- [8] K. A. Lee and W. S. Gan, "Improving convergence of the NLMS algorithm using constrained subband updates," *IEEE Signal Processing Letters*, vol. 11, no. 9, pp. 736-739, Sept. 2004.
- [9] M. S. E. Abadi and J. H. Husøy, "Selective partial update and set-membership subband adaptive filters," *Signal Processing*, vol. 88, no. 10, pp. 2463-2471, Oct. 2008.
- [10] S. E. Kim, Y. S. Choi, M. K. Song, and W. J. Song, "A subband adaptive filtering algorithm employing dynamic selection of subband filters," *IEEE Signal Processing Letters*, vol. 17, no. 3, pp. 245-248, Mar. 2010.
- [11] S. Haykin, *Adaptive Filter Theory*, NJ: Prentice-Hall, 4th Edition, 2002.
- [12] H. C. Shin and A. H. Sayed, "Mean-square performance of a family of affine projection algorithms," *IEEE Trans. Signal Processing*, vol. 52, no. 1, pp. 90-102, Jan. 2004.
- [13] D. E. Knuth, *Sorting and Searching* vol. 3 of the Art of Computer Programming, 2nd Ed. Reading, MA: Addison-Wesley, 1973.
- [14] H. Malvar, *Signal Processing with Lapped Transforms*, Artech House, 1992.

این الگوریتم‌ها استخراج گردید. همچنین به توسعه روش‌های انتخاب جزئی ضرایب، انتخاب پویای زیرباندها و ترکیب این دو روش بر روی الگوریتم زیرباند نرمالیزه شده در شبکه‌های توزیع شده به منظور کاهش پیچیدگی محاسباتی، پرداخته شد. سپس با ارائه روابط تئوری عملکرد الگوریتم‌های پیشنهادی بررسی گردید. در انتها با انجام شبیه‌سازی‌های متعدد صحت روابط الگوریتم‌های پیشنهادی ارزیابی شد و مشاهده گردید که الگوریتم‌های خانواده زیرباند نرمالیزه شده در شبکه افزایشی (dNSAAs) دارای عملکرد مناسبی از لحاظ سرعت همگرایی، خطای حالت ماندگار و باند پایداری نسبت به الگوریتم حداقل میانگین مربعات نرمالیزه شده (dNLMS) هستند. همچنین تطابق نتایج شبیه‌سازی با تئوری، درستی روابط تئوری ارائه شده را نشان داد.

مراجع

محمد شمس اسفندآبادی در سال ۱۳۷۹ مدرک کارشناسی خود را در رشته مهندسی برق از دانشگاه صنعتی نوشیروانی بابل دریافت کرد. وی تحصیلات خود را در مقاطع کارشناسی ارشد و دکتری در رشته مهندسی برق- بیوالکترونیک در دانشگاه تربیت مدرس در سال‌های ۱۳۸۲ و ۱۳۸۶ به پایان رساند. او هم‌اکنون دانشیار دانشکده مهندسی برق دانشگاه تربیت دبیر شهید رجایی می‌باشد. زمینه‌های تحقیقاتی مورد علاقه ایشان عبارتند از پردازش تصویر، فیلترهای تطبیقی و شبکه‌های توزیع شده تطبیقی.

علی‌رضا دانایی تحصیلات خود را در مقاطع کارشناسی و کارشناسی ارشد مهندسی الکترونیک به ترتیب در سال‌های ۱۳۸۶ و ۱۳۹۱ در دانشگاه کردستان و تربیت دبیر شهید رجایی به پایان رسانده است و هم‌اکنون دانشجوی مقطع دکتری مهندسی برق در دانشگاه ریودزانیروی برزیل می‌باشد. زمینه تحقیقاتی مورد علاقه ایشان الگوریتم‌های تطبیقی و شبکه‌های توزیع شده تطبیقی است.

محمدسعید شفیعی تحصیلات خود را در مقاطع کارشناسی و کارشناسی ارشد مهندسی الکترونیک به ترتیب در سال‌های ۱۳۸۸ و ۱۳۹۱ در دانشگاه تربیت دبیر شهید رجایی به پایان رسانده است و هم‌اکنون دانشجوی مقطع دکتری مهندسی برق در آن دانشگاه می‌باشد. زمینه تحقیقاتی مورد علاقه ایشان الگوریتم‌های تطبیقی و شبکه‌های توزیع شده تطبیقی است.

- [1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102-114, Aug. 2002.
- [2] D. Li, K. D. Wong, Y. H. Hu, and A. M. Sayeed, "Detection, classification, and tracking of targets," *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 17-29, Mar. 2002.
- [3] C. G. Lopes and A. H. Sayed, "Incremental adaptive strategies over distributed networks," *IEEE Trans. Signal Processing*, vol. 55, no. 8, pp. 4064-4077, Aug. 2007.
- [4] L. Li, J. A. Chambers, C. G. Lopes, and A. H. Sayed, "Distributed estimation over an adaptive incremental network based on the affine projection algorithm," *IEEE Trans. Signal Processing*, vol. 58, no. 1, pp. 151-164, Jan. 2010.
- [5] M. S. E. Abadi and A. R. Danaee, "Low computational complexity family of affine projection algorithms over adaptive distributed incremental networks," *AEU-International J. of Electronics and Communications*, vol. 68, no. 2, pp. 97-110, Feb. 2014.
- [6] 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, Sept. 1998.
- [7] J. J. Shynk, "Frequency domain and multirate adaptive filtering," *IEEE Signal Processing Magazine*, vol. 9, no. 1, pp. 14-37, Jan. 1992.