# کاهش سطح گلبرگ‌های فرعی در فشرده‌سازی پالس رادار برای آشکارسازی اهداف همسان با استفاده از شبکه عصبی

**علیرضا حکم‌آبادی[۱]  آرزو ذاکری[۲]  احمدرضا شرافت[۳]**

۱- استادیار- گروه مهندسی برق - واحد علی آباد کتول - دانشگاه آزاد اسلامی - علی آباد کتول - ایران
hokmabadi@aliabadiau.ac.ir

۲- استادیار- گروه مهندسی برق - واحد علی آباد کتول - دانشگاه آزاد اسلامی - علی آباد کتول - ایران
zakeri@aliabadiau.ac.ir

۳- استاد- دانشکده مهندسی برق و کامپیوتر- دانشگاه تربیت مدرس- تهران- ایران
sharafat@isc.iranet.net

**چکیده: در این مقاله از یک شبکه عصبی چند لایه برای فشرده‌سازی پالس رادار که ورودی آن برخلاف روش‌های پیشین، کد پلی فاز است، استفاده می‌کنیم. مزیت کدهای پلی فاز نسبت به کدهای باینری، داشتن سطح گلبرگ‌های فرعی کمتر و تولرانس داپلر بسیار بهتر است. بعلاوه، برخلاف کدهای باینری، برای هر طول کد، کدهای پلی فاز متنوعی وجود دارد که این موضوع باعث سودمندی بیشتر شبکهٔ پیشنهادی می‌شود. برای آموزش شبکه نیز از روش بهینه‌سازی Levenberg-Marquardt استفاده کرده‌ایم که سرعت همگرایی شبکه را تا حد قابل توجهی افزایش می‌دهد. پس از ۱۰۰ بار آموزش شبکه مورد اشاره، سطح گلبرگ‌های فرعی تا حد خیره‌کننده‌ای کاهش می‌یابد. به عنوان نمونه، در مورد کد P3 (*N*=۳۰) این مقدار برابر ۲۳۸/۶۶ dB- و در مورد کد P4 (*N*=۴۵) این مقدار برابر۲۴۰/۵۱ dB- است. این طرح برای SNR بسیار پایین نیز عملکرد بسیار خوبی دارد و همچنین در مقایسه با سایر روش‌های موجود تولرانس داپلر بسیار بالاتری را در خروجی شبکه ارائه می‌دهد.**

**کلمات کلیدی:** بهینه‌سازی Levenberg-Marquardt، تولرانس داپلر، شبکهٔ عصبی چند لایه، فشرده‌سازی پالس، گلبرگ‌های فرعی.

**تاریخ ارسال مقاله:** ۱۳۹۱/۱۰/۱۹

**تاریخ پذیرش مشروط مقاله:** ۱۳۹۲/۱۱/۱۲

**تاریخ پذیرش مقاله:** ۱۳۹۵/۸/۱۶

**نام نویسنده‌ی مسئول:** دکتر علیرضا حکم‌آبادی

**نشانی نویسنده‌ی مسئول:** گروه مهندسی برق - واحد علی آباد کتول - دانشگاه آزاد اسلامی - علی آباد کتول - ایران

[ Downloaded from jiaeee.com on 2026-11-15 ]

[ DOR: 20.1001.1.26765810.1396.14.2.7.0 ]

Journal of Iranian Association of Electrical and Electronics Engineers Vol14 No.2 Summer 2017

## ۱- مقدمه

امروزه در سیستم‌های مدرن رادار برای دستیابی به دقت و تفکیک مناسب در برد مورد نظر و همچنین افزایش نسبت سیگنال به نویز (SNR)، بدون افزایش حداکثر توان سیگنال، به‌طور وسیعی از روش‌های فشرده‌سازی پالس استفاده می‌شود [۱]، [۲]. اما متأسفانه این کار سبب به‌وجود آمدن گلبرگ‌های فرعی ناخواسته‌ای در توان خروجی برای ناحیه‌هایی که مورد نظر نیستند، خواهد شد. حضور این گلبرگ‌های فرعی نامطلوب است، زیرا در مواردی باعث ایجاد هشدارهای غلط شده و یا ممکن است گلبرگ‌های فرعی هدفی با SNR بزرگ‌تر سبب پوشانده شدن هدف‌های کوچک‌تر مجاور شوند.

برای کاهش سطح گلبرگ‌های فرعی روش‌های مختلفی پیشنهاد شده است که از مهمترین آنها می‌توان به استفاده از یک شبکهٔ وزن‌دهی پس از فیلتر منطبق [۳]، [۴]، طراحی فیلترهای غیرمنطبق [۵]، [۶]، فشرده‌سازی تطبیقی پالس [۹]-[۷] و استفاده از شبکهٔ عصبی [۱۳]-[۱۰] اشاره کرد. استفاده از شبکهٔ عصبی در فشرده‌سازی پالس رادار در مواردی سودمند است که دامنهٔ سیگنال دریافتی از اهداف مختلف تفاوت زیادی با یکدیگر نداشته باشد، در غیر اینصورت ممکن است بعضی از اهداف آشکارسازی نشوند. برای مواجه با چنین حالاتی در [۹] روش فشرده‌سازی تطبیقی پالس بهبود یافته را ارائه کرده‌ایم که سطح گلبرگ‌های فرعی را تا حد قابل توجهی کاهش داده، تولرانس داپلر بالایی داشته و در کنار آن کوچک‌ترین هدف‌ها را نیز آشکارسازی می‌کند. برای مواردی که دامنه سیگنال دریافتی از اهداف مختلف تفاوت زیادی با یکدیگر ندارند، استفاده از شبکه عصبی می‌تواند نتایج بهتری حاصل کند. شبکهٔ عصبی استفاده شده در [۱۳]-[۱۰] برای فشرده‌سازی پالس، یک شبکهٔ چند لایه پس انتشار خطا است. این روش در مقایسه با روش‌های سنتی عملکرد بسیار خوبی داشته و سطح گلبرگ‌های فرعی را تا حد قابل توجهی کاهش می‌دهد. اما متأسفانه سرعت همگرایی آن بسیار پایین بوده و برای آموزش شبکه به تعداد دوره‌های تکرار زیادی نیاز است. بعلاوه، به علت بکارگیری کدهای باینری در ورودی آن، این روش نسبت به اعوجاج حاصل از شیفت فاز داپلر بسیار حساس بوده و تولرانس داپلر خوبی ندارد.

در این مقاله طرح شبکهٔ عصبی [۱۳]-[۱۰] را به گونه‌ای تغییر داده‌ایم که بتوان از کدهای پلی فاز در ورودی شبکه استفاده کرد. بدین ترتیب می‌توان تولرانس داپلر شبکه را تا حد قابل توجهی افزایش داد. بعلاوه، برای آموزش شبکه از روش بهینه‌سازی Levenberg-Marquardt [۱۴] استفاده کرده‌ایم که سرعت همگرایی شبکه را بسیار بالا می‌برد. در این روش برخلاف روش متداول پس انتشار خطا، بسته به شیب منحنی خطا طول گام‌ها تغییر کرده و در نتیجه سرعت همگرایی به شدت افزایش خواهد یافت. نتایج حاصل از شبیه‌سازی نیز عملکرد بسیار موفق روش پیشنهادی را تأیید می‌کند.

ادامه مقاله به شرح زیر است؛ در بخش دوم، مدل سیستم و ساختار شبکه عصبی پیشنهادی معرفی شده است. نحوه آموزش شبکه عصبی در بخش سوم بطور کامل بیان شده است. پس از آن در بخش چهارم به تحلیل نتایج حاصل از شبیه‌سازی‌ها و بررسی عملکرد شبکه در مقابله با نویز، داپلر و قابلیت آشکارسازی اهداف مجاور هم پرداخته شده است. جمع‌بندی مقاله در بخش پنجم ارائه شده است.

## ۲- مدل سیستم و ساختار شبکهٔ عصبی

در حالت ایده‌آل، سیگنال خروجی فیلتر فشرده‌ساز باید به شکل تابع ضربه باشد، اما در عمل، گلبرگ‌های فرعی ناخواسته‌ای در خروجی فیلتر منطبق تولید می‌شود. هدف اصلی از به‌کارگیری شبکهٔ عصبی به عنوان فیلتر فشرده‌ساز، شکل‌دهی سیگنال خروجی به‌صورت مطلوب است. فرض کنید دنبالهٔ سیگنال ارسالی به‌صورت $\mathbf{s}=[s_1, s_2, \ldots, s_N]^T$ و دنبالهٔ سیگنال دریافتی به‌صورت $\mathbf{r}=[r_1, r_2, \ldots, r_N]^T$ باشد که $N$ طول دنباله و علامت $(\cdot)^T$ معرف عمل‌گر ترانهاده است. در این صورت سیگنال خروجی فیلتر منطبق را می‌توان با رابطهٔ ماتریسی زیر نوشت

$$\mathbf{y}=\mathbf{R}\,\mathbf{s}^* \qquad (۱)$$

که $\mathbf{y}=[y_{-N+1}, \ldots, y_{-1}, y_0, y_1, \ldots, y_{N-1}]^T$ بردار سیگنال خروجی فیلتر منطبق و

$$\mathbf{R}=\begin{bmatrix} r_1 & 0 & 0 & \cdots & 0 & 0 \\ r_2 & r_1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ r_{N-1} & r_{N-2} & r_{N-3} & \cdots & r_1 & 0 \\ r_N & r_{N-1} & r_{N-2} & \cdots & r_2 & r_1 \\ 0 & r_N & r_{N-1} & \cdots & r_3 & r_2 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & r_N & r_{N-1} \\ 0 & 0 & 0 & \cdots & 0 & r_N \end{bmatrix} \qquad (۲)$$

ماتریسی متشکل از نسخه‌های شیفت یافتهٔ دنبالهٔ دریافتی، و $(\cdot)^*$ معرف عمل‌گر مزدوج مختلط است.

با توجه به روابط (۱) و (۲)، مسئلهٔ کاهش سطح گلبرگ‌های فرعی در فشرده‌سازی پالس را می‌توان به‌صورت یک نگاشت غیرخطی از فضای ورودی‌ها (سطرهای ماتریس $\mathbf{R}$) به فضای خروجی (تابع ضربه) در نظر گرفت. بدین منظور یک شبکهٔ عصبی چند لایه در نظر گرفته و از روش بهینه‌سازی Levenberg-Marquardt برای آموزش شبکه استفاده کرده‌ایم. برخلاف [۱۳]-[۱۰]، کدهای مورد استفاده در این شبکه کدهای پلی فاز بوده و محدودیتی در نوع و طول آنها نداریم. در واقع مهمترین ویژگی کدهای باینری پیچیدگی بسیار کم آنها است، زیرا برای پیاده‌سازی آنها نیازی به ضرب کننده نیست. در مقابل، سطح گلبرگ‌های فرعی این کدها نسبتاً بزرگ و تولرانس داپلر آنها بسیار کوچک است. برای رفع این نقیصه می‌توان از کدهای پلی‌فاز استفاده

[ Downloaded from jiaeee.com on 2026-11-15 ]
[ DOR: 20.1001.1.26765810.1396.14.2.7.0 ]
Journal of Iranian Association of Electrical and Electronics Engineers - Vol.14- No.2 Summer 2017

نمود. در این کدها به خاطر استفاده از مقادیر مختلف فاز (غیر باینری)، سطح گلبرگ‌های فرعی کاهش و تولرانس داپلر افزایش می‌یابد. در این میان، دسته‌ای از کدهای پلی‌فاز نظیر کدهای Frank، P1، P2، P3 و P4 وجود دارد که به کدهای پلی‌فاز Chirplike معروف‌اند. کدهای Frank، P1 و P2 (نسخه‌های اصلاح شده کد Frank) تنها برای طول‌های مربع یعنی $N = L^2$ قابل کاربرد بوده و از بسیاری جهات شبیه هم هستند. اما برخلاف آنها کدهای P3 و P4 برای تمامی طول کدها قابل کاربردند. بعلاوه، این کدها از فاز شکل موج FM[1] خطی مشتق شده‌اند و نسبت به کدهای Frank، P1 و P2 که از فاز شکل موج FM پله‌ای مشتق شده‌اند، تولرانس داپلر بیشتری دارند [۱۵]، [۱۶]. فاز این کدها با استفاده از روابط زیر بدست می‌آید

$$\text{P3}: \ \varphi_n = \frac{\pi}{N}(n-1)^2 \qquad \text{for} \ \ 1 \le n \le N \qquad (۳)$$

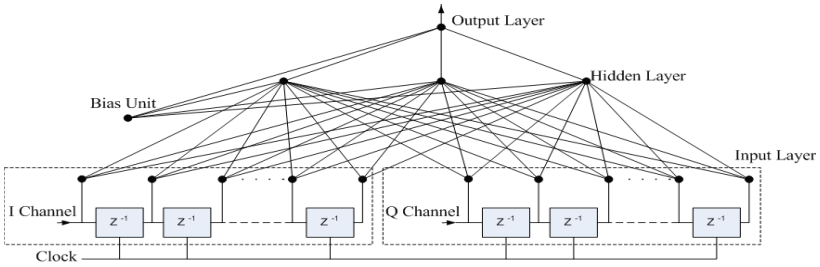
$$\text{P4}: \ \varphi_n = \frac{\pi}{N}(n-1)^2 - \pi(n-1) \quad \text{for} \ \ 1 \le n \le N \qquad (۴)$$

به‌طور مفهومی، کد P3 با آوردن سیگنال LFM[2] به باند پایه توسط یک نوسان‌ساز محلی و نمونه‌برداری از آن با نرخ نایکوئیست تولید می‌شود [۱۶]. بیشترین تغییر فاز بین المان‌های این کد در بخش میانی آن اتفاق می‌افتد. به همین جهت این کد در مقابل محدودیت پهنای باند آسیب‌پذیر است. کد P4 نیز همانند کد P3 تولید می‌شود، اما با این تفاوت که در این کد برای غلبه بر محدودیت پهنای باند، فرکانس نوسان‌ساز محلی تغییر می‌کند. با این راهکار بیشترین تغییر فاز بین المان‌های این کد در کناره‌های آن صورت می‌گیرد و در نتیجه این کد نسبت به کد P3 در مقابل محدودیت پهنای باند مصون‌تر است.

شبکهٔ عصبی ارائه شده از سه لایه تشکیل شده که تعداد نورون‌های لایه ورودی، دو برابر تعداد المان‌های کد ارسالی ($2N$)، تعداد نورون‌های لایه میانی، برابر سه و تعداد نورون‌های لایه خروجی، برابر یک است. بعلاوه، برای بهتر شدن عملکرد شبکه، یک نورون بایاس به هر یک از لایه‌های میانی و خروجی اضافه کرده‌ایم. تابع آستانه نیز سیگموئید است.

همانطور که در شکل (۱) نشان داده شده است، برای به‌کارگیری کدهای پلی فاز در ورودی شبکه، نیمی از نورون‌های لایه ورودی برای عبور مؤلفه‌های هم فاز سیگنال (کانال I) و نیمی دیگر برای عبور مؤلفه‌های فاز متقابل سیگنال (کانال Q) در نظر گرفته شده است. از طرف دیگر، در سیستم‌های مدرن رادار، گیرنده به‌صورت دیجیتالی پیاده‌سازی شده و کانال‌های I و Q به‌طور مستقل از هم محاسبه می‌شوند. بنابراین نیازی به محاسبهٔ مجدد آنها نیست. تعداد نورون‌های لایه میانی نیز با توجه به نتیجهٔ شبیه‌سازی‌ها، برابر سه انتخاب شده است که بهترین گزینه برای داشتن حداکثر کارآیی و حداقل پیچیدگی است.

الگوهای آموزشی شبکه، سطرهای ماتریس **R** بعلاوهٔ یک بردار تمام صفر که به معنای نداشتن ورودی است، خواهد بود. خروجی مطلوب تابع ضربه بوده که برای الگوی کد صحیح برابر یک و برای سایر الگوها برابر صفر است. برای آموزش شبکه نیز تعداد ۱۰۰ دورهٔ تکرار در نظر گرفته‌ایم.




**شکل (۱): ساختار شبکه عصبی پیشنهادی**

## ۳- آموزش شبکه عصبی

مهمترین مشکل شبکه متداول پس انتشار خطا، سرعت همگرایی بسیار پایین آن است. در این شبکه، با یک گام ثابت به سمت کمینهٔ محلی خطا حرکت می‌کنیم، درحالی‌که استفاده از طول گام‌های متغیر بسته به شیب منحنی خطا، باعث افزایش سرعت همگرایی شبکه خواهد شد. برای حل این مسئله در این مقاله از روش بهینه‌سازی Levenberg-Marquardt برای آموزش شبکه بهره گرفته‌ایم.

مدل کلی سیستم را به‌صورت $f(\mathbf{x};\mathbf{w})$ در نظر می‌گیریم. هنگام آموزش شبکه، بردار ورودی $\mathbf{x}$ ثابت و بردار وزن $\mathbf{w}$ متغیر است. تابع خطای $e$ به‌صورت زیر تعریف می‌شود

$$e(\mathbf{w}) = E\left[\left(f(\mathbf{x};\mathbf{w}) - f_{opt}\right)^2\right] \qquad (۵)$$

که $f_{opt}$ مقدار مطلوب و $E[\cdot]$ بیان‌گر مقدار متوسط است. با فرض خطی بودن سیستم، تابع خطا به‌صورت رابطهٔ مرتبهٔ دوم زیر است

$$e(\mathbf{w}) = a + 2\,\mathbf{b}^T\mathbf{w} + \mathbf{w}^T\mathbf{C}\,\mathbf{w} \qquad (۶)$$

که ضریب $a$، بردار $\mathbf{b}$ و ماتریس $\mathbf{C}$ به مقدار متوسط الگوهای ورودی و خروجی وابسته‌اند. با حداقل کردن خطا، مقدار وزن‌ها به‌صورت زیر بدست می‌آید

$$\nabla e(\mathbf{w}) = 0 \quad \Rightarrow \quad \mathbf{w}_{opt} = -\mathbf{C}^{-1}\mathbf{b} \qquad (۷)$$

که $\nabla$ معرف عملگر گرادیان است. مشاهده می‌شود که در یک گام می‌توان به سمت نقطهٔ کمینهٔ منحنی خطا پرش کرد. اما در مورد توابع

[ Downloaded from jiaeee.com on 2026-11-15 ]
[ DOR: 20.1001.1.26765810.1396.14.2.7.0 ]
Journal of Iranian Association of Electrical and Electronics Engineers Vol14 No.2 Summer 2017

غیرخطی، تابع خطا بسیار پیچیده‌تر بوده و نمی‌توان مانند بالا عمل کرد. با این وجود، در مجاورت نقطه کمینهٔ منحنی خطا می‌توان مدل سیستم را به‌صورت خطی تقریب زد و از روش بالا استفاده کرد. در این روش اگر تقریب خوبی زده باشیم، حتی اگر بار اول هم نتوانیم دقیقاً به نقطهٔ کمینهٔ منحنی خطا برسیم، نسبت به روش پس انتشار خطا بسیار سریع‌تر همگرا خواهیم شد.

برای تقریب زدن تابع خطا به شکل مرتبهٔ دوم، تابع جدید $\hat{f}(\mathbf{x};\mathbf{w})$ را که تقریب خطی $f(\mathbf{x};\mathbf{w})$ در همسایگی مقدار وزن معین $\mathbf{w}_0$ است، به‌صورت زیر تعریف می‌کنیم

$$\hat{f}(\mathbf{x};\mathbf{w}) = f(\mathbf{x};\mathbf{w}_0) + (\mathbf{w}-\mathbf{w}_0)^T \nabla f(\mathbf{x};\mathbf{w}_0) \qquad (۸)$$

با جایگزینی $\hat{f}$ به‌جای $f$ در رابطهٔ (۵) و به‌دست آوردن خطای $\hat{e}$، به روابط زیر می‌رسیم

$$\nabla \hat{e}(\mathbf{w}) = 2\,\mathbf{H}\,(\mathbf{w}-\mathbf{w}_0) + 2\,\mathbf{d} \qquad (۹)$$

$$\nabla \hat{e}(\mathbf{w}) = 0 \quad \Rightarrow \quad \mathbf{w}_{opt} = -\,\mathbf{H}^{-1}\,\mathbf{d} + \mathbf{w}_0 \qquad (۱۰)$$

که $\mathbf{d} = E[\,(f(\mathbf{x};\mathbf{w}_0) - f_{opt})\,\nabla f(\mathbf{x};\mathbf{w}_0)]$ گرادیان متوسط خطا و $\mathbf{H} = E[\,\nabla f(\mathbf{x};\mathbf{w}_0)\,\nabla f(\mathbf{x};\mathbf{w}_0)^T\,]$ تقریبی از ماتریس Hessian است. این تقریب، تنها زمانی صحیح است که $f$ خطی باشد. بنابراین از رابطهٔ (۱۰) تنها در ناحیه‌هایی که برای $f$ یک تقریب خطی قابل قبول وجود دارد، استفاده می‌کنیم.

با توجه به تعریف **d**، رابطهٔ اصلاح وزن در روش پس انتشار خطا را می‌توان به‌صورت زیر نوشت

$$\mathbf{w}_{i+1} = \mathbf{w}_i - \mu\,\mathbf{d} \qquad (۱۱)$$

که $\mu$ عددی ثابت بین صفر و یک است. از طرف دیگر، رابطهٔ اصلاح وزن بدست آمده از تقریب مرتبهٔ دوم به‌صورت زیر است

$$\mathbf{w}_{i+1} = \mathbf{w}_i - \mathbf{H}^{-1}\,\mathbf{d} \qquad (۱۲)$$

مسلماً تقریب مرتبهٔ دوم در همهٔ نقاط بهتر عمل نمی‌کند، چون در بدست آوردن آن برای $f$ تقریب خطی در نظر گرفتیم که این تقریب تنها در همسایگی نقطه کمینهٔ خطا قابل قبول است.

برای حل این مسئله در [۱۴] ترکیبی از این دو حالت کرانه‌ای (روابط (۱۱) و (۱۲)) در نظر گرفته شده است. در این الگوریتم، ابتدا از روش پس انتشار خطا استفاده کرده تا به نقطهٔ کمینهٔ خطا نزدیک شویم. سپس به‌تدریج رابطهٔ اصلاح وزن را به سوی تقریب مرتبهٔ دوم تغییر وضعیت داده تا سریع‌تر به نقطهٔ کمینهٔ خطا برسیم. بدین ترتیب با تغییر میزان خطا در هر مرحله می‌توان میزان نزدیکی به کمینهٔ خطا را حدس زد. در این روش رابطهٔ اصلاح وزن به‌صورت زیر است

$$\mathbf{w}_{i+1} = \mathbf{w}_i - (\mathbf{H} - \lambda\,\mathbf{I})^{-1}\,\mathbf{d} \qquad (۱۳)$$

که $\mathbf{I}$ ماتریس یکه و $\lambda$ ضریبی متغیر است و میزان تأثیر دو حالت پس انتشار خطا و تقریب مرتبهٔ دوم را تعیین می‌کند. با توجه به این رابطه، اگر $\lambda$ کوچک باشد، رابطهٔ اصلاح وزن به سمت تقریب مرتبهٔ دوم و بالعکس، اگر $\lambda$ بزرگ باشد، رابطهٔ اصلاح وزن به سمت روش پس انتشار خطا تغییر وضعیت می‌دهد. مقدار $\lambda$ نیز بر اساس تغییرات خطای $e$ به‌صورت زیر تنظیم می‌شود:

- وزن ها را مطابق رابطهٔ (۱۳) اصلاح کرده و خطای حاصل از بردار وزن جدید را محاسبه می‌کنیم.
- اگر خطا افزایش یافت، اصلاح وزن‌ها را نپذیرفته و با تغییر $\lambda$ به یک عدد بزرگ‌تر (مثلاً ۱۰ برابر) مجدداً اصلاح وزن‌ها را انجام داده و خطا را محاسبه می‌کنیم. این فرآیند تا زمانی‌که خطا بزرگ‌تر از مقدار قبلی شود، تکرار می‌شود.
- اگر خطا کاهش یافت، اصلاح وزن‌ها مورد تایید بوده و حال با تغییر $\lambda$ به یک عدد کوچک‌تر (مثلاً ۱۰ برابر) سعی می‌کنیم سرعت همگرایی شبکه را افزایش دهیم. با محاسبهٔ مجدد خطا مراحل فوق تکرار شده تا نهایتاً به یک جواب پایدار برسیم.

به‌طور شهودی، فرآیند اصلاح مقدار $\lambda$ را می‌توان به‌صورت زیر بیان کرد: اگر پس از اصلاح وزن، خطا افزایش یافت، واضح است که تقریب مرتبهٔ دوم مناسب نبوده و در مجاورت نقطهٔ کمینهٔ خطا قرار نداریم. بنابراین باید $\lambda$ را افزایش داد تا تأثیر روش پس انتشار خطا بیشتر شود. برعکس، اگر خطا کاهش یافت، تقریب مرتبهٔ دوم مناسب بوده و برای آنکه بیشتر به نقطهٔ کمینهٔ خطا نزدیک شویم، لازم است $\lambda$ را کاهش داده و تأثیر ماتریس Hessian را بیشتر کنیم.

برای افزایش هر چه بیشتر سرعت همگرایی در [۱۴] نشان داده شده که هنگامی‌که $\lambda$ بزرگ است و از روش پس انتشار خطا استفاده می‌کنیم، به نحو مؤثرتری می‌توان از ماتریس Hessian تخمین زده شده بهره گرفت. بدین منظور ماتریس یکه در رابطهٔ اصلاح وزن Levenberg با ماتریس قطری Hessian جایگزین گردیده و رابطهٔ (۱۳) به‌صورت زیر اصلاح می‌شود

$$\mathbf{w}_{i+1} = \mathbf{w}_i - (\mathbf{H} - \lambda\,diag[\mathbf{H}])^{-1}\,\mathbf{d} \qquad (۱۴)$$

در عمل، این رابطهٔ اصلاح وزن که به روش بهینه‌سازی Levenberg-Marquardt معروف است، برای ساختارهایی از شبکهٔ عصبی با تعداد پارامترهای کم (تعداد وزن‌ها)، بسیار خوب عمل کرده و سرعت همگرایی را به‌طور قابل ملاحظه‌ای بالا می‌برد.

## ۴- نتایج شبیه‌سازی

بر اساس مطالبی که در بخش ۲ به آنها اشاره شد، از کدهای پلی فاز به عنوان ورودی شبکه استفاده می‌کنیم. کدهایی که در این مقاله برای شبیه‌سازی در نظر گرفته‌ایم، کد P3 [۱۶] با طول $N$=۳۰ و کد P4 [۱۶] با طول $N$=۴۵ است. برای آموزش شبکه نیز تنها ۱۰۰ دورهٔ تکرار در نظر گرفته‌ایم. شکل‌های (۲) و (۳)، خروجی شبکه را پس از آموزش نشان می‌دهند. می‌بینیم که با وجود تعداد بسیار کم دوره‌های آموزشی، سطح گلبرگ‌های فرعی تا حد خیره‌کننده‌ای کاهش یافته است. برای نمونه، در مورد کد P3 ($N$=۳۰)، سطح بزرگ‌ترین گلبرگ



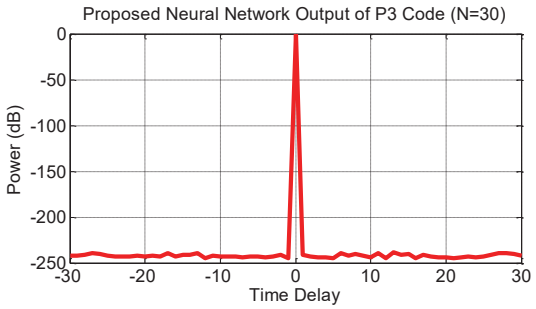
[ Downloaded from jiaeee.com on 2026-11-15 ]

[ DOR: 20.1001.1.26765810.1396.14.2.7.0 ]

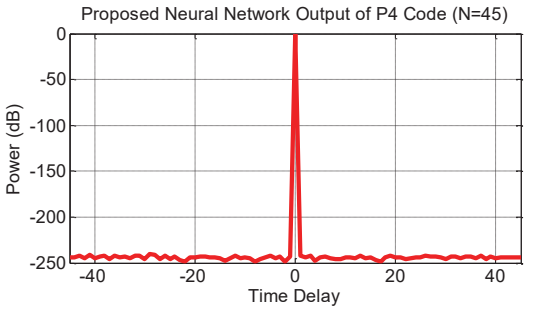
Journal of Iranian Association of Electrical and Electronics Engineers - Vol.14- No.2 Summer 2017

فرعی ($PSL^{۳}$) برابر ۲۳۸/۶۶ dB- و در مورد کد P4 ($N$=۴۵)، برابر ۲۴۰/۵۱ dB- است.

در شبکه پیشنهادی، تنها مجاز به استفاده از کدهای پلی‌فاز هستیم، در حالی‌که در ساختار پیشنهاد شده [۱۳]-[۱۰] تنها کدهای باینری قابل کاربرد است. بنابراین نمی‌توان عملکرد این روش‌ها را در مورد کدهای مشابه با یکدیگر مقایسه نمود. با این وجود، برای آنکه بتوانیم عملکرد بسیار موفق الگوریتم آموزشی Levenberg-Marquardt را نشان دهیم، همان ساختار شبکه عصبی [۱۳]-[۱۰] را در نظر گرفته و از الگوریتم Levenberg-Marquardt برای اصلاح وزن‌ها استفاده می‌کنیم. در این حالت تنها پس از گذشت ۱۰۰ دوره تکرار، مقدار PSL خروجی شبکه عصبی در مورد کد Barker ($N$=۱۳) به dB ۲۳۶/۱۶- می‌رسد. این در حالی است که در روش [۱۰] پس از ۵۰۰ دوره تکرار، مقدار PSL برابر با dB ۴۲/۷۳- خواهد بود. در مورد کد *m-Sequence* ($N$=۳۱) نیز با استفاده از الگوریتم Levenberg-Marquardt، تنها پس از گذشت ۱۰۰ دوره تکرار مقدار PSL برابر dB ۲۲۴/۸۷- خواهد بود، در حالی‌که در [۱۳] پس از گذشت $۱۰^{۴}$ دوره تکرار، مقدار PSL از dB ۶۲- تجاوز نخواهد کرد. بنابراین مشاهده می‌شود که استفاده از روش بهینه‌سازی Levenberg-Marquardt مشکل کند شدن سرعت همگرایی شبکه عصبی را به نحو شایسته‌ای حل کرده است.



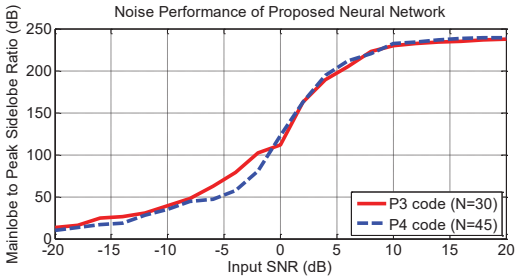

**شکل (۲): شکل موج سیگنال خروجی شبکه با استفاده از کد P3 ($N$=۳۰)**




**شکل (۳): شکل موج سیگنال خروجی شبکه با استفاده از کد P4 ($N$=۴۵)**

## ۴-۱- عملکرد شبکه در مقابل نویز

برای بررسی عملکرد شبکه عصبی در مقابل نویز، SNR سیگنال ورودی را از dB ۲۰- تا dB ۲۰ تغییر داده و نسبت گلبرگ اصلی به بزرگ‌ترین گلبرگ فرعی ($MPSR^{۴}$) سیگنال خروجی را محاسبه می‌کنیم. در شکل (۴)، منحنی‌های حاصل از تغییرات MPSR در مقابل SNR ورودی در مورد این دو کد رسم شده است. می‌بینیم که طرح پیشنهادی برخلاف سایر روش‌ها در SNR بسیار پایین نیز کارآیی فوق‌العاده‌ای دارد، به‌طوری‌که در SNR=-۱۳ dB مقدار MPSR برای هر دو کد تقریبا برابر ۳۰ dB است که این مقدار برای غالب کاربردهای رادار کفایت می‌کند. بعلاوه می‌بینیم که با افزایش SNR، عملکرد شبکه به نحو چشمگیری افزایش می‌یابد، به عنوان مثال در SNRهای dB ۸-، dB ۰ و dB ۱۵ مقدار MPSR برای هر دو کد به ترتیب بزرگ‌تر از dB ۵۰، dB ۱۰۰ و dB ۲۱۰ است. بنابراین می‌توان چنین نتیجه گرفت که شبکه ارائه شده علاوه بر SNRهای بزرگ، در SNRهای بسیار پایین نیز عملکرد بسیار موثری داشته و نیازهای یک سیستم رادار را در حد مطلوبی فراهم می‌کند.



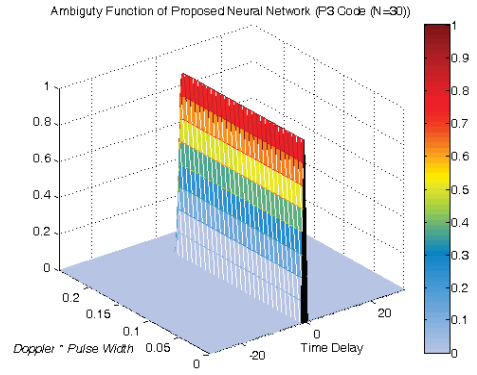

**شکل (۴): عملکرد شبکه در مقابل نویز**

## ۴-۲- عملکرد شبکه در مقابل اثر داپلر

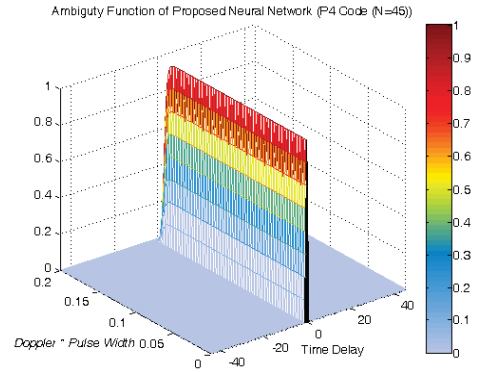
یکی از چالش‌های مهمی که در طراحی فیلتر فشرده‌ساز مطرح است، رفتار سیستم در مواجه با اهداف متحرک است. در چنین سناریوهایی وجود سرعت نسبی معمولاً بزرگ بین هدف و رادار، سبب به‌وجود آمدن اثر داپلر و تغییر فرکانس سیگنال برگشتی خواهد شد [۱۷، ۱۸]. در این بخش اثر داپلر را در شبیه‌سازی لحاظ کرده و رفتار شبکه عصبی پیشنهادی را در مواجه با آن بررسی کرده‌ایم. باید توجه داشت که در آموزش شبکه تنها از سطرهای ماتریس **R** استفاده شده و هیچ‌گونه تغییری ناشی از نویز و داپلر در آموزش شبکه استفاده نشده است. نتایج حاصله به‌صورت تابع ابهام نرمالیزه که یک تابع دوبعدی وابسته به زمان و فرکانس داپلر ($\chi(\tau, f)$) است، بوده و بیان‌گر تغییرات خروجی فیلتر فشرده‌ساز به ازاء مقادیر مختلف تاخیر زمانی و داپلر است. شکل‌های (۵) و (۶) به‌ترتیب مربوط به تابع ابهام نرمالیزه کدهای P3 ($N$=۳۰) و P4 ($N$=۴۵) می‌باشند. می‌بینیم که با وجود طولانی بودن کدها، شبکه پیشنهادی در برخورد با پدیده داپلر بسیار موفق

[ Downloaded from jiaeee.com on 2026-11-15 ]
[ DOR: 20.1001.1.26765810.1396.14.2.7.0 ]
Journal of Iranian Association of Electrical and Electronics Engineers Vol14 No.2 Summer 2017

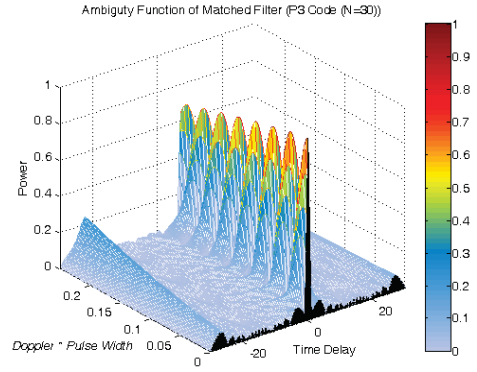
عمل کرده و تولرانس داپلر بسیار خوبی دارد. همچنین بـرای مقایسـه، در شکل‌های (۷) و (۸) تابع ابهام نرمالیزه شده فیلتر منطبق نیز رسـم شده است. می‌بینیم که تابع ابهام حاصل از شبکه پیشنهادی، تـولرانس داپلر و تزویج برد-داپلر مشابه بـا تـابع ابهـام حاصـل از فیلتـر منطبـق داشته، درحالی‌که سطح گلبرگ‌های فرعی تقریباً در تمـام سـطح تـابع ابهام نزدیک صفر است.



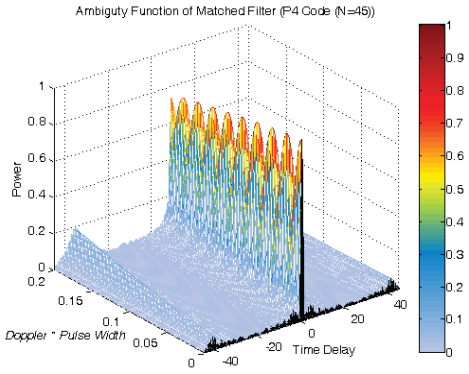

**شکل (۵): تابع ابهام نرمالیزه شبکه عصبی در مورد کد P3 (*N*=۳۰)**




**شکل (۶): تابع ابهام نرمالیزه شبکه عصبی در مورد کد P4 (*N*=۴۵)**




**شکل (۷): تابع ابهام نرمالیزه فیلتر منطبق در مورد کد P3 (*N*=۳۰)**

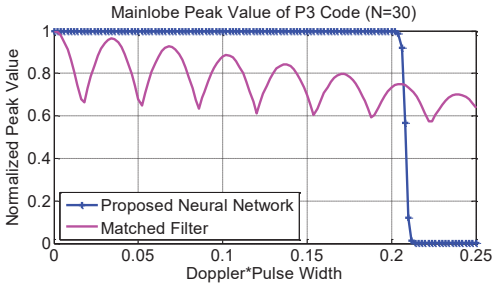



**شکل (۸): تابع ابهام نرمالیزه فیلتر منطبق در مورد کد P4 (*N*=۴۵)**

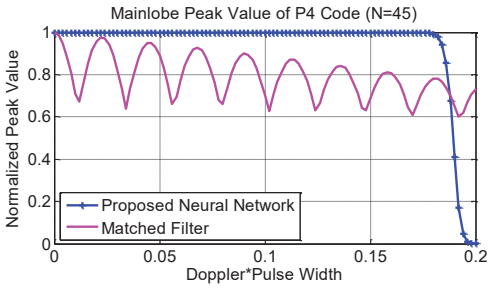
برای مقایسه دقیق‌تر، در شـکل‌هـای (۹) و (۱۰) منحنـی افـت انـدازه گلبرگ اصلی این کدها در مقابل تغییرات داپلـر (بـه صـورت نرمـالیزه شده) رسم شده است. می‌بینیم که افت اندازه گلبـرگ اصـلی ناشـی از شبکه پیشنهادی بسیار ناچیز بوده و حتی برای داپلرهای بزرگ نیز این مقدار تغییر چندانی نخواهد کرد، در حالی‌که افت اندازه گلبرگ اصـلی سیگنال خروجی فیلتر منطبق با افزودن داپلر با شیب ملایمـی کـاهش می‌یابد. البته همان‌طور که در شکل‌ها نیز مشهود است، چنانچه مقـدار فرکانس داپلر بیش از حد بزرگ شود شبکه پیشنهادی دیگـر قـادر بـه شناسایی هدف نخواهد بود. اما این مسئله نگران کننده نیست؛ چراکـه در سناریوهای واقعی با چنین سرعت‌هایی روبرو نخواهیم شد. به عنوان نمونه با توجه به شکل‌های (۹) و (۱۰) عملکرد شبکه پیشنهادی به ازاء مقادیر حاصل‌ضرب داپلر در عرض پالس (Doppler * Pulse Width) بزرگ‌تر از ۰/۲ مختل می‌شود؛ این مقدار داپلر در یک سیسـتم راداری معمول با مشخصـات $f_0 = 9\,GHz$ (فرکـانس مرکـزی) و $\tau = 2\,\mu s$ (عرض پالس) معادل با سـرعت ۱۶۰۰ متـر بـر ثانیـه خواهـد بـود کـه سرعت بسیار بزرگی است.

در شکل‌های (۱۱) و (۱۲) نیز منحنی مقدار PSL این کدها در مقابـل تغییرات داپلر (به صورت نرمالیزه شده) رسم شده است. می‌بینـیم کـه برخلاف فیلتر منطبق، حتی برای داپلرهـای بسـیار بـزرگ نیـز مقـدار PSL کماکان نزدیک صفر (dB ۲۰۰- >) بـاقی مـی‌مانـد. بـه عبـارت دیگر ساختار پیشنهادی با استفاده از کدهای پلی فاز در ورودی شـبکه عصـبی بسـیار کارآمـد بـوده و تـولرانس داپلـر بـالایی دارد. در شبیه‌سازی‌های انجام شده با اسـتفاده از دیگـر کـدهای پلـی فـاز و بـا طول‌های مختلف نیز نتـایج مشـابهی بدسـت آمـده اسـت کـه صـحت عملکرد ساختار پیشنهادی را تأیید می‌کنند.

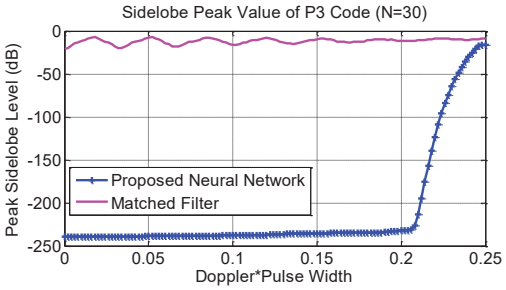
[ Downloaded from jiaeee.com on 2026-11-15 ]
[ DOR: 20.1001.1.26765810.1396.14.2.7.0 ]



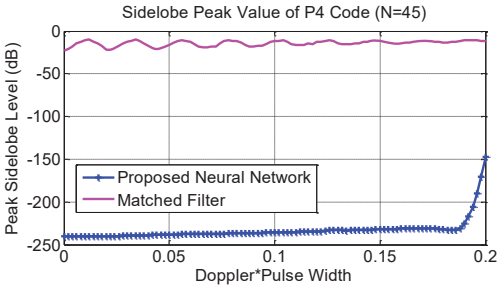

**شکل (۹): منحنی افت اندازه گلبرگ اصلی کد P3 ($N$=۳۰) در مقابل تغییرات داپلر (به صورت نرمالیزه شده)**




**شکل (۱۰): منحنی افت اندازه گلبرگ اصلی کد P4 ($N$=۴۵) در مقابل تغییرات داپلر (به صورت نرمالیزه شده)**




**شکل (۱۱): منحنی مقدار PSL کد P3 ($N$=۳۰) در مقابل تغییرات داپلر (به صورت نرمالیزه شده)**




**شکل (۱۲): منحنی مقدار PSL کد P4 ($N$=۴۵) در مقابل تغییرات داپلر (به صورت نرمالیزه شده)**

## ۴-۳- قابلیت آشکارسازی اهداف مجاور هم و تولرانس قابل قبول دامنه برای سیگنال‌های دریافتی

در این قسمت برای شبیه‌سازی سیگنال حاوی دو موج منعکس شده از اهداف مستقل که با یکدیگر همپوشانی دارند، دو کد مشابه با طول $N$ را نسبت به یکدیگر $L$ بیت تأخیر داده $(L \in \mathrm{N} \ \& \ L < N)$ و با نسبت دامنه‌های متفاوت با یکدیگر جمع می‌کنیم. سپس از این سیگنال به عنوان ورودی شبکه عصبی استفاده خواهیم کرد. بدین ترتیب می‌توان با تغییر مقدار تأخیر و نسبت دامنه‌ها، قابلیت تفکیک و تولرانس قابل قبول دامنه برای سیگنال‌های دریافتی را بررسی کرد.

نتایج بدست آمده برای حالت‌های مختلف تأخیر و نسبت دامنه در مورد کد P3 ($N$=۳۰) و P4 ($N$=۴۵) در جدول (۱) آورده شده است. در این جدول نسبت دامنه ورودی‌ها به‌صورت نسبت دامنه کد اول به دامنه کد تأخیر یافته تعریف می‌شود. نسبت دامنه‌های خروجی نیز به‌صورت نسبت اندازه گلبرگ‌های اصلی در سیگنال خروجی شبکه تعریف می‌شود.

**جدول (۱): نتایج بدست آمده از شبیه‌سازی برای حالت‌های مختلف تأخیر و نسبت دامنه.**

| مقدار تاخیر | نسبت دامنه‌های ورودی | نسبت دامنه‌های خروجی | | PSMR [dB] | |
|---|---|---|---|---|---|
| | | P4 ($N$=۴۵) | P3 ($N$=۳۰) | P4 ($N$=۴۵) | P3 ($N$=۳۰) |
| ۲ بیت تاخیر | ۱ | $-۲/۳\times۱۰^{-۱۰}$ | $-۲/۷۷\times۱۰^{-۱۱}$ | -۲۳۳/۲۶ | -۲۳۴/۰۳ |
| | ۲ | -۳۶/۶۰ | -۳۴/۶۲ | -۲۰۲/۶۰ | -۲۰۲/۵۰ |
| | ۳ | -۱۳۵/۸۴ | -۱۸۸/۵۶ | -۱۰۳/۸۳ | -۴۸/۹۲ |
| | ۴ | -۱۷۷/۷۴ | -۲۲۱/۰۱ | -۶۲/۱۵ | -۱۶/۶۱ |
| | ۵ | -۲۰۱/۰۶ | -۲۳۱/۰۰ | -۳۸/۹۶ | -۶/۷۰ |
| ۳ بیت تاخیر | ۱ | $-۱/۹\times۱۰^{-۱۴}$ | $-۱/۲\times۱۰^{-۱۰}$ | -۲۳۲/۳۹ | -۲۳۱/۹۴ |
| | ۲ | $-۲/۸\times۱۰^{-۸}$ | -۱۴/۷۹ | -۲۳۹/۱۰ | -۲۲۲/۰۷ |
| | ۳ | -۱۵/۸۹ | -۱۲۹/۵۵ | -۲۲۳/۷۷ | -۱۰۸/۴۶ |
| | ۴ | -۱۲۸/۱۵ | -۱۸۲/۲۶ | -۱۱۱/۷۷ | -۵۶/۰۷ |
| | ۵ | -۱۷۶/۳۸ | -۲۰۶/۰۴ | -۶۳/۶۷ | -۳۲/۲۱ |
| ۴ بیت تاخیر | ۱ | $-۴/۴\times۱۰^{-۱۱}$ | $-۵/۶\times۱۰^{-۱۲}$ | -۲۳۳/۶۸ | -۲۳۲/۵۹ |
| | ۲ | -۴۱/۲۴ | -۶/۰۴ | -۱۹۸/۹۶ | -۲۳۱/۹۳ |
| | ۳ | -۱۷۶/۸۱ | -۱۷۸/۱۸ | -۶۳/۸۸ | -۶۰/۲۵ |
| | ۴ | -۲۱۷/۷۹ | -۲۱۷/۹۰ | -۲۳/۱۰ | -۲۰/۵۶ |
| | ۵ | -۲۳۱/۳۱ | -۲۹۹/۷۴ | -۹/۶۷ | -۸/۷۱ |

[ Downloaded from jiaeee.com on 2026-11-15 ]

[ DOR: 20.1001.1.26765810.1396.14.2.7.0 ]

مقدار PSMR۵ را نیز به‌صورت نسبت اندازه بزرگ‌ترین گلبرگ فرعی به اندازه گلبرگ اصلی هدف کوچک‌تر تعریف کرده‌ایم. با توجه به مقادیر جدول (۱) می‌بینیم که برای نسبت دامنه ورودی ۵ و تأخیرهای مختلف، به سختی می‌توان سیگنال کوچک‌تر را تشخیص داد. با شبیه‌سازی حالت‌های دیگر نیز می‌توان به نتایج مشابهی رسید. بنابراین بطور کلی می‌توان چنین نتیجه گرفت که این شبکه عصبی همانند [۱۳]-[۱۰]، تنها قادر به تشخیص سیگنال‌های نزدیک هم با نسبت دامنه کمتر از ۵ است.

## ۵- نتیجه‌گیری:

در این مقاله از یک شبکهٔ عصبی چند لایه برای فشرده‌سازی پالس رادار استفاده کردیم که ورودی آن برخلاف روش پیشین، کد پلی فاز است. علت استفاده از کدهای پلی فاز به‌جای کدهای باینری در ورودی شبکه، بالا بودن تولرانس داپلر آنها است. بعلاوه، برخلاف کدهای باینری، برای هر طول کد، کدهای پلی فاز متنوعی وجود دارد که این مسئله باعث سودمندی بیشتر شبکهٔ پیشنهادی می‌شود. به منظور افزایش سرعت همگرایی شبکه نیز از روش بهینه‌سازی Levenberg-Marquardt برای آموزش شبکه استفاده کرده‌ایم. در این روش برخلاف الگوریتم متداول پس انتشار خطا که با یک گام ثابت به سمت کمینهٔ محلی حرکت می‌کنیم، بسته به شیب منحنی خطا طول گام‌ها را تغییر داده و در نتیجه سرعت همگرایی به شدت افزایش می‌یابد. استفاده از روش پیشنهادی نتایج بسیار خوبی را حاصل می‌کند. تنها پس از ۱۰۰ بار آموزش، سطح گلبرگ‌های فرعی تقریباً به صفر رسیده است. این روش در SNR بسیار پایین نیز کارآیی قابل توجهی از خود نشان می‌دهد. بعلاوه، پس از بررسی اثر داپلر بر روی عملکرد شبکه و استخراج تابع ابهام آن، می‌بینیم که این طرح همانند فیلتر منطبق، تولرانس داپلر بالایی دارد.

به منظور بررسی عملکرد شبکه عصبی در آشکارسازی اهداف مجاور هم با نسبت دامنه‌های مختلف، شبیه‌سازی‌هایی انجام شد که مشخص گردید که این شبکه عصبی همانند [۱۳]-[۱۰]، تنها قادر به تشخیص سیگنال‌های نزدیک هم با نسبت دامنه کمتر از ۵ است. بنابراین شاید بتوان تنها نقطه ضعف این روش را زیاد بالا نبودن تولرانس دامنه برای سیگنال‌های دریافتی دانست. البته در غالب موارد عملی چنین مشکلی پیش نخواهد آمد و شبکه عصبی ارائه شده بسیار موفق عمل خواهد کرد.

## مراجع


[1] M. Phillips, Roger, Bell, G., Storz, G., and Lilburn, L. "Frequency Correction for Pulse Compression Radar," U.S. Patent Application 15/004,752, filed January 22, 2016.

[2] Richards, M.A., Fundamentals of Radar Signal Processing, 2ed.,McGraw-Hill Education, 2013.

[3] Ranganath, Praveen, and Rao, S., "Effect of Pulse Shaping on Autocorrelation Function of Barker and Frank Phase Codes," Journal of Advanced Electrical and Computer Engineering, no. 1 pp. 22-31, 2014.

[4] Nunn, Carroll J., and Gregory E., "Polyphase pulse compression codes with optimal peak and integrated sidelobes," IEEE Transactions on Aerospace and Electronic Systems, vol. 45, no. 2, 2009.

[5] Kulpa, Janusz, S. "Mismatched filter for range sidelobes suppression of pseudo-noise signals," In IEEE Signal Processing Symposium (SPSympo), pp. 1-4, 2015.

[6] Rabaste, Olivier, and Savy, L., "Mismatched filter optimization for radar applications using quadratically constrained quadratic programs," IEEE Transactions on Aerospace and Electronic Systems, vol. 51, no. 4, pp. 3107-3122, 2015.

[7] S. D. Blunt, and K. Gerlach, "Adaptive pulse compression via MMSE estimation," IEEE Trans. Aero. and Electron. Sys., vol. 42, no. 2, Apr. 2006.

[8] Henke, Dakota, McCormick, P., Shannon D., and Higgins, D., "Practical aspects of optimal mismatch filtering and adaptive pulse compression for FM waveforms," In IEEE Radar Conference (RadarCon), pp. 1149-1155, 2015.

[۹] علیرضا حکم‌آبادی، احمدرضا شرافت، و آرزو ذاکری، "بهبود فشرده‌سازی تطبیقی پالس با در نظر گرفتن اثر داپلر در سامانه رادار،" هفدهمین کنفرانس مهندسی برق ایران، دانشگاه علم و صنعت ایران، اردیبهشت ۱۳۸۸.

[10] Padaki, Aditya, V., and George, K., "Improving performance in neural network based pulse compression for binary and polyphase codes," 12th International IEEE Conference on Computer Modelling and Simulation (UKSim), pp. 278-283, 2010.

[11] Baghel, V., and Panda, G., "Development of an efficient hybrid model for range sidelobe suppression in pulse compression radar." Aerospace Science and Technology, vol. 27, no. 1, pp. 156-162, 2013.

[12] Duh, F. B., Juang, C. F., Lin, C. T., "A neural fuzzy network approach to radar pulse compression," IEEE Geosci. Remote Sens. Lett., vol. 1 no. 1, pp. 15–20, 2004.

[13] Khairnar, D. G., Merchant, S. N., and Desai, U. B., "Radial basis function neural network for pulse radar detection," IET Radar Sonar Navig. vol. 1, no. 1, pp. 8–17, 2007.

[14] L. S. H. Ngia, and J. Sjoberg, "Efficient training of neural nets for nonlinear adaptive filtering using a recursive Levenberg-Marquardt algorithm," IEEE Trans. Sig. Proc., vol. 48, pp. 1915-1927, Jul. 2000.

[15] Jamil, Momin, Zepernick, H., and Pettersson, M. I., "Performance assessment of polyphase pulse compression codes," IEEE 10th International Symposium on Spread Spectrum Techniques and Applications, pp. 166-172, 2008.

[16] Blunt, Shannon D., Cook, M., Jakabosky, J., Graaf, J. D., and Perrins, E., "Polyphase-coded FM (PCFM) radar waveforms, part I: implementation," IEEE Transactions on Aerospace and Electronic Systems vol. 50, no. 3, pp. 2218-2229, 2014.

[۱۷] محمد علائی، و حمیدرضا امین‌داور، " روشی جدید در بازشناسایی خودکار اهداف متحرک زمینی با استفاده از رادارهای مراقبت زمینی پالس داپلر"، مجله انجمن مهندسین برق و الکترونیک ایران، سال ششم، شماره دوم، پائیز و زمستان ۱۳۸۸.



[ Downloaded from jiaeee.com on 2026-11-15 ]



[ DOR: 20.1001.1.26765810.1396.14.2.7.0 ]

[۱۸] معین احمدی، و کمال محامدپور، " تشخیص مدولاسیون فاصله تکرار پالس های راداری"، مجله انجمن مهندسین برق و الکترونیک ایران، سال ششم، شماره دوم، پائیز و زمستان ۱۳۸۸.

## زیرنویس‌ها

[1] Frequency Modulation
[2] Linear Frequency Modulation
[3] Peak Sidelobe Level
[4] Mainlobe to Peak Sidelobe Ratio
[5] Peak Sidelobe to Mainlobe Ratio

[ Downloaded from jiaeee.com on 2026-11-15 ]

[ DOR: 20.1001.1.26765810.1396.14.2.7.0 ]