

رویکردی برای حفاظت از عملیات‌های پردازش داده در سامانه‌های محاسباتی با استفاده از کدهای کانولوشن

حجت‌الله حمیدی

گروه مهندسی فناوری اطلاعات، دانشگاه صنعتی خواجه نصیر الدین طوسی، تهران، ایران

چکیده

در این مقاله چارچوبی برای روش‌های تحمل‌پذیری خطای مبتنی بر الگوریتم در سامانه‌های محاسباتی ارائه می‌دهیم. خطاهایی که در الگوریتم‌های پردازش داده رخ می‌دهند، با به کارگیری کدهای کانولوشن نظام‌مند تعریف شده بر روی ساختارهای حسابی نقطه ثابت، آشکارسازی و تصحیح می‌شوند. کدهای کانولوشن، امکان آشکارسازی خطای در عملیات پردازش داده عددی را، فراهم می‌کنند. بسیاری از تکنیک‌های تحمل‌پذیر خطای که مربوط به حفاظت از عملیات‌های ماتریسی عددی هستند از کدهای بلوکی با افزونگی بالا برای قابلیت تصحیح خطای استفاده می‌کنند. در این مقاله، از کدهای کانولوشن استفاده شده و از ویژگی‌های آزمون پیوسته برای آشکارسازی خطاهای استفاده می‌شود. این تکنیک آشکارسازی خطای مبتنی بر الگوریتم، بر مقایسه مقادیر توازن محاسبه شده از دو مسیر تکیه دارد. خطاهای پردازش داده‌های عددی توسط مقایسه مقادیر توازن مرتبط با کدهای کانولوشن آشکار می‌شوند. این روش یک رویکرد جدید برای دست‌یابی به اصلاح خطای همزمان در سامانه‌های محاسبات تحمل‌پذیری خطای است. این مقاله طرح محاسباتی جدیدی را به منظور فراهم آوردن تحمل‌پذیری خطای برای الگوریتم‌های عددی ارائه می‌دهد. در حالی که تصحیح داده‌های خروجی با یک کدگشایی (کدبردار) آستانه که تحت تأثیر خطاهای گردکردن و خطاهای ناشی از محاسبه‌گر است، صورت می‌گیرد. هدف، توصیف یک رویکرد حفاظتی است که به آسانی با تکنیک‌های پردازش داده ادغام شده و به تحمل‌پذیری مؤثرتر خطای انجامد. نتایج شبیه‌سازی، نشان می‌دهند که اختلاف میان خطای گردکردن و خطای ناشی از محاسبه‌گر به اندازه کافی برای متمایز شدن از یکدیگر بزرگ می‌باشد. این اختلاف بزرگ میان این دو خطای محدوده و گستره‌ای از آستانه‌های آشکارسازی خطای برای انتخاب به ما می‌دهد.

واژگان کلیدی: پردازش داده؛ کدهای کانولوشن؛ مقادیر توازن؛ خطای گردکردن؛ آشکارسازی خطای.

۱۹۸۴؛ آشوبی و همکاران، ۲۰۱۰؛ بیلیس، ۱۹۹۸؛ جویی و همکاران، ۱۹۸۶).

پردازش خطی اجراسده با پردازش گرهای تعییه شده، در بسیاری از سامانه‌های عملی، دارای نقش محوری است. مثال‌های مهم از این پردازش‌ها، شامل پردازش سیگنال دیجیتال پرسرعت در ماهواره‌های مخابراتی^۱، الگوریتم‌های محاسباتی ماتریسی بزرگ و سامانه‌های کنترل تولید^۲ که فعالیت‌های همزمان^۳ بسیاری را هماهنگ می‌شوند. از آن جایی که خطاهای می‌توانند منجر به یک عملکرد غیرقابل قبول شوند، تحمل‌پذیری خطای، یک مسأله پراهمیت است؛

۱- مقدمه

روش‌های متعددی برای افزایش قابلیت اطمینان^۱ و تحمل‌پذیری خطای سامانه‌های محاسباتی در سال‌های اخیر ارائه شده است (اسری و همکاران، ۱۹۹۳؛ بنرجی و همکاران، ۱۹۹۰)، با این همه به علت افزایش پیچیدگی و هزینه هنگفتی که طراحی سامانه‌های مطمئن و تحمل‌پذیر خطای به همراه دارد، بسیاری از سامانه‌های موجود یا به طور کامل قادر امکانات سخت‌افزاری لازم برای حفظ تحمل‌پذیری خطای هستند و یا در صورت وجود، تنها ویژگی‌های محدودی را حمایت می‌کنند (هانگ و همکاران،

² Communication

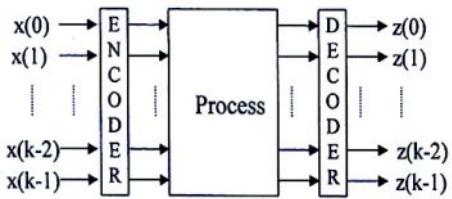
³ Manufacturing

⁴ Simultaneous

^۱ Reliability



تحمل پذیری خطا یعنی افزونگی مدولار همانند TMR، TMR باعث افزایش چشم‌گیر هزینه، پیچیدگی و مشکلات طراحی می‌شود. شکل (۱)، (حمیدی و همکاران، ۲۰۱۴)، عمومی ترین روش تحمل پذیری خطای مبتنی بر الگوریتم، اعمال یک کدگذاری بر روی اطلاعات ورودی، اجرای الگوریتم محاسباتی بر روی این اطلاعات کدشده و سپس کدگشایی از نتایج خروجی و بررسی صحت اطلاعات است. کدهای کانولوشن، آشکارسازی خطای بهشیوهای پیوسته و با استفاده از همان منابع محاسباتی فراهم می‌کنند. این نوع از تحمل پذیری خطای مبتنی بر افزونگی این کدها بر روی حلقه محدود (اعداد صحیح به پیمانه‌های تووان دو)، هستند که به طور معمول در سامانه‌های پردازش عددی مورد استفاده قرار می‌گیرند.



(شکل - ۱): روند نمای عمومی تکنیک‌های تحمل پذیری خطای مبتنی بر الگوریتم (حمیدی و همکاران، ۲۰۱۴).

بسیاری از سامانه‌های مخابراتی و پردازش سیگنال، شامل منابع محاسباتی دیجیتالی هستند که عملیات‌های حسابی نقطه ثابت را اجرا می‌کنند. کدهای کانولوشن با حافظه کدگذاری‌شان می‌توانند برای سامانه‌هایی با جریان داده‌های پیوسته، عملکرد مؤثری را ارائه دهند؛ در این خصوص می‌توان به کدهای کانولوشن برروی اعداد صحیح به پیمانه Z_2^m (۲^m، Z_2)، همراه با شیوه‌های تصمیم‌گیری خطای مبتنی بر این کدهای بازخورد، اشاره کرد.

۲- ساختار تحمل پذیری خطای مبتنی بر الگوریتم

در این بخش، ساختاری را که برای روش‌های تحمل پذیری خطای مبتنی بر الگوریتم، در سامانه‌های محاسباتی، به منظور حفاظت از عملیات‌های پردازش خطی ارائه داده‌ایم، شرح می‌دهیم. درواقع تحمل پذیری خطای مبتنی بر الگوریتم، ارائه شده یک طرح کلی آشکارسازی خطاست که از کدهای اعداد حقیقی برای پی‌بردن به خطاهای ناشی از محاسبه‌گر در داده‌های پردازش شده، همان‌گونه که در شکل

لذا تکنیک‌های تحمل پذیری خطای مبتنی بر الگوریتم، با موفقیت برای حفاظت از این نوع سامانه‌ها به کار گرفته شده‌اند (جویی و همکاران، ۱۹۸۶).

تحمل پذیری خطای مبتنی بر الگوریتم، متکی بر سازوکار کدگذاری و کدبرداری اطلاعات است. بنابراین از افزونگی اطلاعات، بهره می‌گیرد. افزونگی مورد نیاز برای روش‌های تحمل پذیری خطای مبتنی بر الگوریتم، به طور معمول توسط کدهای حقیقی بلوکی تعریف می‌شود (حمیدی و همکاران، ۲۰۱۰؛ ۲۰۱۱؛ ۲۰۱۲؛ ۲۰۱۳؛ ۲۰۱۱؛ ۲۰۱۲؛ ۲۰۱۱؛ ۲۰۰۹؛ ۱۹۹۲؛ ۲۰۱۲؛ ۲۰۱۱؛ ۲۰۰۹؛ ۱۹۹۸؛ ۲۰۰۹؛ ۲۰۱۰ و ۲۰۱۰؛ ۲۰۰۹).

در اکثر کاربردهای قبلی تحمل پذیری خطای مبتنی بر الگوریتم، تصحیح داده‌های خطاطار، به ندرت مورد توجه قرار گرفته است. با این وجود، در تحقیقاتی اخیر، که پردازش سیگنال دیجیتال را بکار می‌برند، روش‌های تصحیح خطای مورد بررسی قرار گرفته است (ردینبو و همکاران، ۲۰۰۹؛ ۱۹۹۲؛ ۲۰۱۱؛ ۲۰۰۳؛ ۲۰۱۲؛ ۲۰۱۱؛ ۲۰۰۹؛ ۱۹۹۸؛ ۲۰۰۹؛ ۲۰۱۰ و ۲۰۱۰؛ ۲۰۰۹).

بسیاری از عملیات‌های پردازش خطی، مانند کدهای کانولوشن اعداد حقیقی، حاوی یک جریان پیوسته^۱ از داده‌های ورودی هستند. (نیر و همکاران، ۱۹۹۰)، که مقادیر توازن را از طریق یک ساختار پاسخ ضربه محدود (FIR)^۲ تولید می‌کنند، خصوصیاتی دارند که به آسانی با نیازمندی‌های پردازش مرتبط با الگوریتمی که قرار است، حفاظت شود، تلفیق می‌شوند (بوسیلیکا و همکاران، ۲۰۰۹). به علاوه، کدهای کانولوشن اعداد حقیقی با کدهای کانولوشن مبتنی بر میدان محدود، مرتبط بوده و خصوصیات عملکردی تضمین شده‌ای دارند؛ بهویژه کدهای کانولوشن تصحیح خطاهای متوالی^۳، که برای آشکارسازی وقوع^۴ خطای در داده‌ها، مناسب هستند (مورالز و همکاران، ۲۰۰۶ و ویتری و همکاران، ۱۹۸۵).

در سامانه‌های پردازش داده، تقاضای زیادی برای محاسبه‌گرهای با عملکرد بالا، وجود دارد. استفاده از محاسبه‌گرهای پسرعت، نیاز به درجه بالایی از قابلیت اطمینان نتایج محاسبه شده دارد، احتمال بروز خطای در نتایج، با زیاد شدن میزان محاسبات، افزایش می‌یابد. برای اصلاح خطاهای، و افزایش قابلیت اطمینان، سامانه طراحی شده، باید تحمل پذیر خطای باشد. تشخیص و تصحیح خطاهای گذرا، در یک سامانه دیجیتال خطی، با استفاده از روش‌های عمومی

¹ Steady Flow

² Finite Impulse Response (FIR)

³ Burst-Correcting

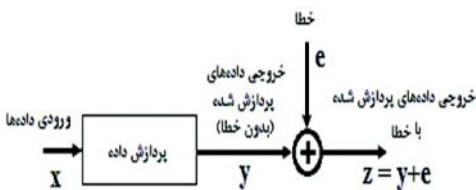
⁴ Onset



تحت شرایط بدون خطأ، مقادیر توازن P_L به طور تقریبی برابر با مقادیر توازن P_U هستند. در روش تحمل پذیری خطأ مبتنی بر الگوریتم، هنگامی خطأ آشکارسازی می شود که تفاضل این دو مقدار توازن، از مقدار آستانه، بیشتر شود. اگر خطأ در یکی از سه زیرسامانه موجود در بلوك دیاگرام شکل (۲) وجود داشته باشد، آن گاه خطأ مدل سازی شده در خروجی آن زیرسامانه، به طور قابل ملاحظه ای افزایش خواهد یافت و موجب می شود که دو رشتة توازن، با یکدیگر اختلاف داشته باشند.

۳- مدل خطأ در پردازش داده

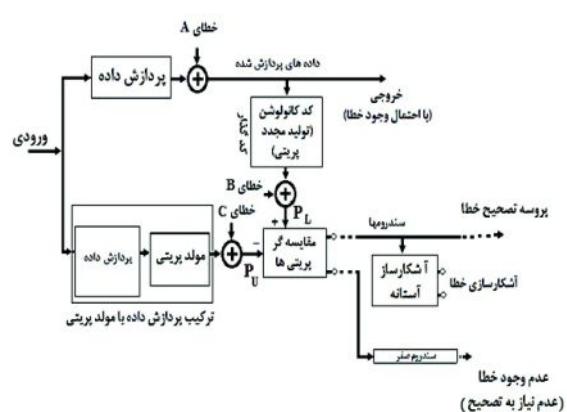
یکی از ویژگی های تکنیکی در تحمل پذیری خطأ در کاربردهای پردازش سیگنال و پردازش تصویر، استفاده از مدل های سطح بالای خطأ است. خطأ به طور عمومی به شکل خطأ جمع شونده و منفرد گذرا، در خروجی مدل می شود. در طول همه زمانی که اثر این خطأ در خروجی وجود دارد خطأ دیگری در سامانه رخ نمی دهد. تأثیر خطأ بر روی مؤلفه های کلمه کد (خروجی کدگذار کانولوشن) با افزودن یک بدرار خطأ e مدل سازی می شود شکل (۳). به بیان دیگر، خطأ به صورت افزایشی (جمع شونده) در خروجی مدل سازی می شود.



(شکل - ۳): مدل خطأ در پردازش داده

به دلیل آن که در این کاربرد، در ابتدا فرآیند آشکارسازی خطأ صورت می گیرد، سطح تحمل پذیری خطأ اینمی^۲ است، و از آن جایی که در ادامه فرآیند آشکارسازی خطأ از یک کدگشا، برای تصحیح خطاهای در رشتة سندروم، استفاده می شود، پس این ساختار، یک سطح تحمل پذیری بازیابی^۳ را نیز فراهم کرده است. درنتیجه سطح تحمل پذیری خطأ، در این مدل هم اینمی و هم بازیابی است.

(۲) نشان داده شده، استفاده می کند. سامانه پردازش داده از طریق مقادیر توازن تعیین شده توسط یک کد کانولوشن حقيقی نظام مند حفاظت می شود. کدهای کانولوشن به جريان های پیوسته داده ها اجازه می دهند که برخلاف کدهای بلوکی حقيقی، بدون قطعه بندی^۱، مورد حفاظت قرار بگيرند. در اين ساختار از مقادير توازن محاسبه شده از دو مسیر مجاوز و قابل مقایسه استفاده می شود تا خطاهای ناشی از محاسبه گر را در عملیات های پیوسته پردازش داده ای عددی، آشکارسازی کنند. جريان های مقادير توازن از دو مسیر محاسبه می شوند؛ يك گروه از طریق داده های پردازش شده خروجی، و جريان قابل مقایسه دیگر توسط ترکیب عملیات پردازش داده با تابع مولد توازن فراهم می شود. مقایسه توازن ها، امكان آشکارسازی خطأ را فراهم كرده، و در ادامه از تفاضل دو مقدار توازن به دست آمده از دو مسیر بالا و پایین، رشتة سندروم تولید می شود؛ كه با استفاده از اطلاعات اين رشتة سندروم ها، خطاهای قابل آشکارسازی و تصحیح می شوند. اگر رشتة سندروم دارای مقادیر کوچکتری از حد آستانه باشد، خطای در سامانه رخ نداده است؛ ولی اگر رشتة سندروم دارای مقادیر بزرگ تر از حد آستانه باشد، نشان دهنده آن است که در سامانه خطأ رخ داده است. تصحیح خطأ در صورتی که مقدار خطأ از سطح آستانه بیشتر باشد، توسط يك کدبردار انجام می شود. در این مقاله، خطاهای، هم در پردازش خطی و هم در زیرسامانه های تولید توازن، در نظر گرفته و منابع خطأ، به صورت منفرد گذراي جمع شونده در خروجی، مدل سازی می شوند.



(شکل - ۲): روند نمای ساختار تحمل پذیری خطای مبتنی بر الگوریتم عمومی با استفاده از ترکیب پردازش داده و مولد توازن

¹ Segmentation

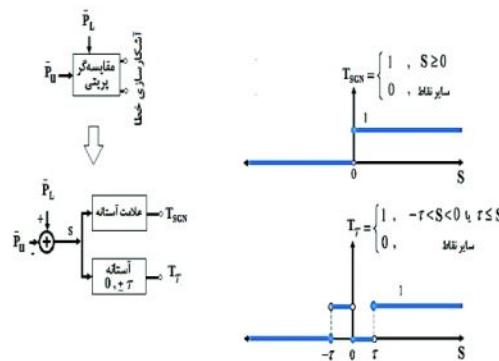
²Safety

³Recovery

سال ۱۳۹۵ شماره ۲ پیاپی ۲۸

۱-۳- آشکارسازی خط

از تفاضل دو بردار توازن \bar{P}_u و \bar{P}_l در شکل ۲، که از دو مسیر مجزا محاسبه می‌شوند، بردار سندروم به دست می‌آید. صرف نظر از خطای گرد کردن، دو مقدار توازن \bar{P}_u و \bar{P}_l می‌باشد در حالت بدون خط، مساوی باشند. از این‌رو، آشکارسازی خط، با مقایسه \bar{P}_u و \bar{P}_l در خروجی مقایسه‌گر حاصل می‌شود. این مقایسه‌گر، تفاضل بین دو مقدار توازن (سندروم S)، را محاسبه می‌کند، اگر اندازه مقدار محاسبه شده از مقدار آستانه انتخاب شده کوچک‌تر باشد، $|S| < \tau$ و $S = \bar{P}_l - \bar{P}_u$ ، آن‌گاه خطای رخ نداده است، (τ مقدار آستانه می‌باشد). هر سندروم، بیان‌گر یک زوج قابل مقایسه از مقادیر توازن حقیقی است، که در یک زوج از مقادیر دودویی متتمرکز می‌شود؛ که نشان‌دهنده یک کلمه کد یک از دو^۱ است. شکل (۴)، نشان می‌دهد که چگونه دو مقدار توازن، با درنظرداشتن یک آستانه τ ، به یک زوج کلمه کد یک از دو، تبدیل می‌شود. زوج خروجی (T_{SGN}, T_r)، یک کلمه کد یک از دو معتبر خواهد بود، $\{(0, 1), (1, 0)\}$ ، به شرط آن که اختلاف میان مقادیر توازن (S)، در سطح آستانه τ باشد. درصورتی که اختلاف مقادیر توازن، بزرگ‌تر از مقدار آستانه باشد، خروجی مقایسه‌گر، در شکل (۴)، به یک خروجی با کد نامعتبر منجر خواهد شد، که در این حالت یک وضعیت آشکارسازی خط، رخ می‌دهد.



(شکل - ۴): مقایسه‌گر مقادیر توازن

۴- کدهای کانولوشن تعریف شده بر روی اعداد صحیح

کدهای کانولوشن تعریف شده بر روی اعداد صحیح به پیمانه توان دو، یک ساختار ریاضی برای محاسبات ممیز ثابت

هستند؛ این کدها، کدهای کانولوشن دودویی شناخته شده‌ای را به عنوان مولدهای اصلی شان به کار می‌برند (برلکمپ، ۱۹۶۲؛ لی، ۱۹۹۷؛ ماسی، ۱۹۹۳).

خطاهایی که در الگوریتم‌های پردازش داده رخ می‌دهند، ممکن است با به کارگیری کدهای کانولوشن نظام مند تعریف شده بر روی ساختارهای حسابی نقطه ثابتی که از محاسبات پشتیبانی می‌کنند، آشکارسازی و حتی تصحیح شوند. این طبقه از کدهای کانولوشن حسابی از نمادهای برگرفته از حلقة محدود، بر مبنای کدهای تصحیح توده‌ای دودویی هستند. هدف، توصیف یک رویکرد حفاظتی است که به‌آسانی با تکنیک‌های پردازش داده ادغام شده و به تحمل پذیری مؤثرتر خطای انجامد. در این مقاله، کدهای کانولوشن بر روی ساختارهای حسابی نقطه ثابت، تشریح شده، و محاسبات در این ساختارها اجرا می‌شوند. ساختار بلوکی کدها، محدودیت‌هایی را بر جریان دادهای پردازش شده، اعمال می‌کند. برخلاف کدهای بلوکی، کدهای کانولوشن، وقوع خطای را آشکارسازی می‌کنند. در این مقاله، قابلیت‌های تصحیح کدهای کانولوشن میدان محدود Z_2^m مورد بررسی قرار می‌گیرد. حساب نقطه ثابت، به طور دقیق با عملیات‌های میدان محدود Z_2^m مطابقت دارد. همه عملیات‌های جمع و ضرب به شیوه معمول تعریف می‌شوند. چندین نمایش در مورد این ساختارهای حسابی وجود دارد. یکی از آن‌ها با نشان‌گرهای^۲ اعداد صحیح $-2^m, 0, 1, 2, \dots, m$ ^۳ است. با این همه، اعداد صحیح همچنین بسطهای دودویی منحصر به‌فردی دارند، که منجر به m تایی دودویی مطابق با نشان‌گرهای اعداد صحیح می‌شوند.

$$a \in Z_2^m; a \rightarrow (a_0, a_1, \dots, a_{m-1}), a = \sum_{q=0}^{m-1} a_q 2^q; a_i \in \{0, 1\} \quad (1)$$

که رابطه، $a \in Z_2^m \Rightarrow a = ((a \cdot 2^m))_2$ بیان‌گر مکمل دو است. در بخش بعدی، کدهای کانولوشن بر روی ساختار حسابی نقطه ثابت Z_2^m تشریح می‌شوند.

۱-۴- ساختار کد کانولوشن و آزمون

کدهای کانولوشن، حتی آنها که بر روی دامنه‌ای از اعداد صحیح نقطه ثابت تعریف شده‌اند، توسط یک ماشین حالت محدود^۳ قابل توصیف هستند (کاستلو و همکاران، ۲۰۰۴). یک کد کانولوشن توصیف شده توسط فرآیند ماشین

²Representatives

³Finite State Machine

فصل نهم



مؤلفه‌های کلمه کد در \bar{y}_t با افزودن یک بردار خطای \bar{e}_t به هر قسمت، مدل‌سازی شده و به کلمه کد \bar{z}_t منجر می‌شود.

$$\bar{z}_t = \bar{y}_t + \bar{e}_t \quad (7)$$

البته رابطه (7) بدین معنا نیست که این سازوکارها ماهیت افزایشی دارند؛ بلکه بدین معنا است که تأثیر تمامی Z_2^m خطاهای توسعه یک بردار، مدل‌سازی شده تا بر روی Z_2^m اضافه شوند. اصل ساختار تحمل‌پذیری خطا متکی بر کدهای نظاممندی است که موقعیت‌های توازن و داده‌های مجزای قابل تشخیص دارند. برای نمایش ماتریس آزمون توازن چنین کدهایی، حوزه محدودی به کار برد شده و با ماتریس H^T نشان داده می‌شود. این نمایش، $(n - k)$ معادله همگن را تعریف می‌کند:

$$0 = H^{vT} \bar{z}_t; \bar{z}_t = \begin{pmatrix} \#_{t-v} \\ \#_{t-v+1} \\ \vdots \\ \#_{t-1} \\ \#_t \end{pmatrix}, \quad (8)$$

ماتریس آزمون توازن از $(v + 1) \times n$ زیرماتریس H_i^T ، که هر کدام دارای ابعاد $n \times n - k$ است، شکل گرفته است.

$$H^{vT} = (H_v^T, H_{v-1}^T, \dots, H_1^T, H_0^T) \quad (9)$$

روابط آزمون توازن، شامل هم مکان‌های آزمون و هم مکان‌های داده در هر کلمه کد \bar{y}_t است. k مکان داده برای یک کد با انداخت k/n توسط $d_{t,i}$ مشخص شده، و مکان‌های آزمون توسط $c_{t,r}$ مشخص می‌شود.

$$\bar{y}_t = \begin{pmatrix} d_{t,1} \\ d_{t,2} \\ \vdots \\ d_{t,k-1} \\ d_{t,k} \\ c_{t,1} \\ c_{t,2} \\ \vdots \\ c_{t,(n-k)} \end{pmatrix} = \begin{pmatrix} d_t \\ \bar{d}_t \\ \vdots \\ \bar{d}_t \\ \bar{c}_t \\ \bar{c}_t \\ \vdots \\ \bar{c}_t \end{pmatrix}; \quad \begin{array}{l} \text{مقدار داده: } d_{t,i} \in Z_2^m \\ \text{مقدار توازن: } d_{t,r} \in Z_2^m \\ \text{بردار داده: } \bar{d}_t \\ \text{بردار توازن: } \bar{c}_t \end{array} \quad (10)$$

برای کدهای نظاممند، زیرماتریس‌های ماتریس آزمون توازن $\{H_i\}$ در رابطه (9) را می‌توان به شکل روابط نمایش داد:

$$H_0^T = \left(P_0 \mid I_{n-k} \right);$$

حالت محدود، به آسانی بر روی یک حلقه محدود، قابل اجراست. بردارها و ماتریس‌های ارائه شده در زیر، شامل عناصری از Z_2^m می‌شوند. برای بهتر معرفی کردن این نوع از کدهای کانولوشن، ابتدا عملیات کدگذاری آن‌ها بررسی می‌شود. جریان نمادهای ورودی کدگذار، مجزا شده و k نماد در واحد زمان در قالب یک بردار گنجانده می‌شود. وضعیت جاری کدگذار توسط گروه‌بندی V بردار ورودی با یکدیگر، تعریف می‌شود. (t یک عدد صحیح بوده و از مقدار اولیه $t = 0$ آغاز می‌شود).

$$\{\bar{z}_t\}_{t=0}^{+\infty} = \begin{pmatrix} \#_{t-v} \\ \#_{t-v+1} \\ \vdots \\ \#_{t-1} \\ \#_t \end{pmatrix} \quad (2)$$

که $\{\bar{z}_t\}_{t=0}^{+\infty}$ بردار حالت جاری به ابعاد $(V \times 1)$ بوده با طول محدودیت V ، در جایی که $\{\bar{z}_t\}_{t=0}^{+\infty}$ بردار ورودی با مرتبه $1 \times V$ است. بردار حالت بعدی پس از استفاده از بردار ورودی \bar{u}_t ، تابعی از حالت جاری \bar{X}_t و ورودی جاری \bar{u}_t می‌باشد.

$$\bar{z}_{t+1} = \begin{pmatrix} \#_{t-v} \\ \#_{t-v+1} \\ \vdots \\ \#_{t-1} \\ \#_t \end{pmatrix} \quad (3)$$

از طرف دیگر، خروجی جاری کدگذار (\bar{y}_t) ، یعنی n رقم خروجی از Z_2^m ، یک بردار است. این خروجی یکتابع خطی از حالت جاری \bar{X}_t و ورودی جاری \bar{u}_t است که با بردار انتقال ورودی \bar{G}_t نشان داده می‌شود.

$$\bar{y}_t = \bar{G}_t \bar{G}_t \quad (4)$$

که \bar{G}_t برابر است با:

$$\bar{G}_t = \begin{pmatrix} \#_{t-v} \\ \#_{t-v+1} \\ \vdots \\ \#_{t-1} \\ \#_t \end{pmatrix}; \quad (V+1)^{k \times 1} \quad (5)$$

در جایی که ماتریس خروجی G ، با ابعاد $(V+1) \times k$ برابر است با:

$$G = (G_V, G_{V-1}, \dots, G_1, G_0) \quad (6)$$

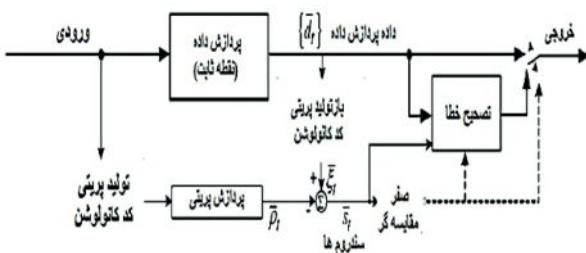
که G_i ، به ابعاد $n \times k$ است.

روابط (3) و (6)، یک کد کانولوشن با نرخ k/n با طول محدودیت V را توصیف می‌کنند. تأثیر خطا بر روی

ماتریس آزمون توازن H^{vT} , بر روی مؤلفه‌هایی که خطایی ندارند، بی‌تأثیر بوده و تنها تأثیر خطا بر روی موقعيت داده‌ها $\{\bar{e}_{t-i}^{(d)}\}$ و نیز در جدیدترین موقعیت آزمون $\bar{e}_t^{(c)}$, خواهد بود.

$$\begin{aligned}\bar{s}_t &= \sum_{i=0}^v p_i \bar{d}_{t-i} + \bar{c}_t + \sum_{r=0}^v p_i \bar{e}_t^{(d)} + \bar{e}_t^{(c)} \quad (a) \\ \bar{s}_t &= 0 + \sum_{r=0}^v p_i \bar{e}_t^{(d)} + \bar{e}_t^{(c)} \quad (b).\end{aligned}\quad (16)$$

خطاهای با توجه با قابلیت آشکارسازی کد، هنگامی که $\bar{s}_t \neq \bar{0}$ باشد، آشکارسازی می‌شوند. روابط توازن مرتبط با مقادیر توازن تولیدشده توسط کدهای کانولوشن تعریف شده بر روی اعداد صحیح و داده‌ها هستند؛ بنابراین در کاربرد، در ساختار تحمل پذیری خطای یکسری از موقعیت‌های توازن مرتبط با موقعیت‌های داده‌های پردازش شده می‌باشند، که به طور مستقل توسط یک مسیر پردازش موازی، همان‌گونه که در شکل (۵) نشان داده شده، محاسبه می‌شوند. همان‌گونه که بیان شد، یکی از کاربردهای کدهای کانولوشن تعریف شده بر روی اعداد صحیح به پیمانه 2^m آشکارسازی خطای در ساختارهای تحمل پذیر خطای است. این اصول آشکارسازی خطای، متکی بر مقایسه مقادیر توازن است که در دو مسیر مجزا محاسبه شده‌اند. پردازش موازی مقادیر توازن ورودی، به مقادیر توازن خروجی می‌انجامد که قابل مقایسه با نمونه‌های توازن دوباره تولیدشده از خروجی‌های پردازش شده اصلی است.



(شکل-۵): یک ساختار تحمل پذیر خطای

داده‌های تولیدشده در زیرسامانه پردازش داده، در رشتۀ بردار $\{\bar{d}_{t-i}\}_{i=0}^v$ به ابعاد $1 \times k$ نشان داده می‌شود؛ در حالی که در مسیر موازی آن، مقادیر توازن متناظر $\{\bar{c}_{t-i}\}_{i=0}^v$ ، تولید می‌شوند. این بردارهای توازن با بردار \bar{P}_t مشخص شده تا تأکید کنند که آنها ممکن است شامل

$$\begin{array}{ll} P_0 \begin{pmatrix} n-k \\ n-k \end{pmatrix} & \text{ماتریس واحد} \\ H_t^T = \left(\begin{array}{c|c} P_i & 0_{n-k} \end{array} \right); & \\ P_i \begin{pmatrix} n-k \\ n-k \end{pmatrix} & \text{ماتریس صفر} \\ i=1,2,\dots,v. & \end{array} \quad (11)$$

۷ مکان در زیرماتریس صفر $0_{(n-k)}$ و ماتریس واحد H^{vT} در $I_{(n-k)}$ نشان‌دهنده آن است که معادله همگن (۸) را می‌توان به‌آسانی در حوزه زمان نمایش داد. علاوه‌بر آن می‌توان مقادیر مکان آزمون و داده را در شکل دیگری از بردار خروجی \bar{E}_t نمایش داد:

$$\bar{E}_t = \begin{pmatrix} \bar{d}_{t-v} \\ \vdots \\ \bar{d}_{t-v+1} \\ \vdots \\ \bar{d}_t \\ \vdots \\ \bar{d}_1 \\ \bar{e}_t \end{pmatrix} \quad (12)$$

$$0 = \bar{c}_t + \sum_{i=0}^v P_i \bar{d}_{t-i} \quad (13)$$

در کدهای نظاممند، برای بردارهای داده، رابطه $\bar{d}_t = \bar{u}_t$ برقرار است. از طرف دیگر، با فرض این که n مکان در \bar{y}_t (بردار کلمه کد به ابعاد $1 \times n$) توسط خطاهای مدل‌سازی شده خراب شده باشند، نتیجه حاصل، در برداری به نام \bar{E}_t نمایش داده می‌شود.

$$\begin{array}{l} \bar{E}_t = \begin{pmatrix} \bar{Z}_{t-v} \\ \bar{Z}_{t-v+1} \\ \vdots \\ \bar{Z}_t \end{pmatrix} \\ \bar{Z}_i = \bar{y}_i + \bar{e}_i; \\ i = (t-v), (t-v+1), \dots, t \\ \text{ابعاد برداری } \bar{d}_i \bar{e}_i^{(d)} \text{ } k \times 1 \\ \bar{Z}_i = \begin{pmatrix} \bar{d}_i + \bar{e}_i^{(d)} \\ \bar{c}_i + \bar{e}_i^{(e)} \end{pmatrix} \text{ } \bar{c}_i \bar{e}_i^{(e)} \text{ } (n-k) \times 1 \\ \text{ابعاد برداری } \bar{c}_i \bar{e}_i^{(e)} \text{ } (n-k) \times 1 \end{array} \quad (14)$$

رابطه آزمون توازن ممکن است به دلیل خطای این مورد، صدق نکند. رابطه (۸) یک بردار \bar{s}_t غیر صفر با ابعاد $(n-k) \times 1$ را تولید می‌کند که بردار سندروم نامیده می‌شود.

$$\bar{s}_t = H^{vT} \bar{E}_t; \quad (n-k) \times 1 \quad (15)$$

از این رو تأثیر تمام خطاهای مدل‌سازی شده، در بردار سندروم \bar{s}_t ، گنجانده شده است.

۴-۲- کدهای کانولوشن بر روی حلقة حسابی نقطه ثابت

کدهای کانولوشن با نمادهایی از حلقة محدود Z_2^m بسطهای دودویی به شکل برداری و ماتریسی دارند؛ برای مثال، بردار وروید داده $\{\bar{u}_t\}$ به ابعاد $(k \times 1)$ در عملیات کدگذاری با استفاده از بردارهای دودویی بسط داده می‌شود.

$$\bar{u}_t = \sum_{q=0}^{m-1} 2^q \bar{u}_t^{(q)}; \quad (22)$$

همچنین بردار کلمه کد خطدار، $\bar{e}_t + \bar{y}_t = \bar{z}_t$ ، همانگونه که در بردار خروجی \bar{z}_t نشان داده می‌شود، نیز دارای یک بسط بردار دودویی است.

$$\bar{z}_t = \sum_{q=0}^{m-1} 2^q \bar{z}_t^{(q)} \quad (23)$$

به شیوه‌ای مشابه، ماتریس مولد G و ماتریس آزمون H^T را نیز می‌توان بسط داد، که این ماتریس‌ها نیز ضرایب دودویی دارند. یک نتیجه مهم برای کدهای بلوکی چرخشی بر روی Z_2^m وجود دارد که می‌تواند به کدهای کانولوشن تعمیم داده شود؛ به بیان دیگر، حداقل فاصله برای تصحیح خطای کد بر روی Z_2^m به طور مستقیم با ماتریس دودویی آزمون توازن مرتبط است. این موضوع بدین معناست که قابلیت عملکردی یک کانولوشن دودویی می‌تواند تنها با استفاده از ماتریس‌های مولد و آزمون توازن دودویی، به یک کانولوشن حسابی، بر روی Z_2^m انتقال یابد. از این‌رو، همه کدهای کانولوشن دودویی می‌توانند در حلقة محدود Z_2^m قرار بگیرند. از ماتریس آزمون توازن دودویی کدهای نظاممند در آشکارسازی و تصحیح کدهای کانولوشن حسابی استفاده می‌شود. مقادیر باین‌ری صفر و یک در حلقة مرتبط با Z_2^m دیده می‌شود، بنابراین تمایز $+1$ از -1 در این حلقة حائز اهمیت است. در حالت نظاممند، ماتریس مولد G ، با ماتریس آزمون توازن H^T مرتبط است. زیرماتریس‌ها در ماتریس مولد G شامل زیرماتریس‌های توازن P_i می‌شوند:

$$G = \begin{pmatrix} 0 & 0 & \dots & 0 & I_k \\ -P_v & -P_{v-1} & \dots & -P_1 & -P_0 \\ \vdots & & & & \\ G_v & G_{v-1} & \dots & G_1 & G_0 \end{pmatrix} \quad (24)$$

در کدهای کانولوشن دودویی استاندارد، نیازی به تمایز میان $+1$ از -1 نیست، با این وجود، این موضوع برای حالت Z_2^m ضروری است. عملیات‌های حسابی نقطه ثابت، در

خطاهای پردازش باشند و از طریق تفاضل مقادیر توازن تولیدشده در دو مسیر محاسبه می‌شوند و نه با مقادیر آزمون \bar{c}_t .

$$\bar{p}_t \leftrightarrow \sum_{i=0}^v P_i \bar{d}_{t-i} \quad (17)$$

گروه دوم مقادیر توازن، به طور مستقیم از داده‌های پردازش شده و توسط زیرسامانه تولید مجدد توازن که در شکل ۵ نشان داده شده، تولید می‌شوند. این مقادیر توازن، با \bar{z}_t نشان داده می‌شوند.

$$\bar{z}_t \leftrightarrow \sum_{r=0}^v P_r \bar{d}_{t-r} \quad (18)$$

با فرض این که خطاهای مدل‌سازی شده به همراه بردارهای داده وجود داشته باشند ($\bar{d}_{t-i} + \bar{e}_{t-i}^{(d)}$)، و همچنین با درنظر گرفتن خطاهای مدل‌سازی شده فرآیند تولید مجدد $\bar{e}_t^{(c)}$ می‌توان مقادیر توازن خطدار را که توسط فرآیند تولید مجدد \bar{z}_t تولید شده‌اند، به صورت رابطه (۱۹) نشان داد:

$$\begin{aligned} \bar{z}_t &= \sum_{r=0}^v P_r \left(\bar{d}_{t-r} + \bar{e}_{t-r}^{(d)} \right) + \bar{e}_t^{(c)} \\ \bar{z}_t^{(n-k)} &\times 1; \\ \bar{e}_t^{(c)} &\times 1. \end{aligned} \quad (19)$$

از تفاضل مقادیر توازن محاسبه شده به صورت موارد در (۱۷)، با مقادیر توازن محاسبه شده توسط فرآیند تولید مجدد \bar{z}_t ، مجموعه‌ای از سندروم‌ها به دست می‌آید:

$$\bar{z}_t - \bar{p}_t = \sum_{i=0}^v p_i \bar{e}_{t-i}^{(d)} + \bar{e}_t^{(c)} + \sum_{r=0}^v P_r \bar{d}_{t-r} - \sum_{i=0}^v P_i \bar{d}_{t-i}$$

$$\bar{z}_t - \bar{p}_t = \sum_{i=0}^v p_i \bar{e}_{t-i}^{(d)} + \bar{e}_t^{(c)} = \bar{s}_t. \quad (20)$$

همان‌گونه که نشان داده می‌شود، تفاضل مقادیر محاسبه شده توازن، بردار سندروم \bar{s}_t را ایجاد می‌کند. اگر خطای $\bar{e}_t^{(c)}$ بر روی محاسبات توازن موارد \bar{p}_t تأثیر بگذارد، آن‌گاه سندروم \bar{s}_t برابر با $(\bar{e}_t^{(c)} - \bar{e}_t'^{(c)})$ می‌شود، که می‌تواند در یک مجموعه از خطاهای مدل‌سازی شده در درون سندروم \bar{s}_t نمایش داده شود.

$$\bar{s}_t = \sum_{i=0}^v p_i \bar{e}_{t-i}^{(d)} + (\bar{e}_t^{(c)} - \bar{e}_t'^{(c)}) \quad (21)$$

(n-1) سطر و (n) ستون، مقادیر دودویی بوده تا خواص تصحیح و آشکارسازی مطلوب را ارائه دهد. درواقع، سطرهای B_0 شامل زیرماتریس‌های توازن P_i مورد نیاز در H^{VT} هستند.

$$B_0 = \begin{bmatrix} 0 & \dots & \dots & 0 & 0 & 1 \\ 0 & \dots & \dots & 0 & 1 & 0 \\ 0 & \dots & \dots & 1 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & \dots & \dots & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 \\ x & 0 & 0 & \dots & 0 & 0 \\ x & x & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots & 0 \\ x & \dots & x & x & 0 & 0 \\ x & x & x & \dots & x & 0 \end{bmatrix} \quad (25)$$

$$B_0 = \begin{pmatrix} P_0 & | & I_1 \\ P_1 & | & 0 \\ P_2 & | & 0 \\ \vdots & | & \vdots \\ P_{V-1} & | & 0 \\ P_V & | & 0 \end{pmatrix} \quad (26)$$

در آخرین ستون این ماتریس، ذرایه نخست، یک بوده و تمام ذرایه‌های زیر آن، صفر هستند. به علاوه، n ستون ماتریس B_0 به عنوان یک زیرفضای n بعدی هستند. بردار سندروم با $(v+1)$ مقدار، به منظور آشکارسازی و تصحیح، با $\bar{\sigma}_t$ نشان داده می‌شود.

$$\bar{\delta}_t = \begin{pmatrix} S_{t,1} \\ S_{t,2} \\ \vdots \\ S_{t+v} \end{pmatrix}, \quad (v+1)=2n. \quad (27)$$

در این طبقه از کدها $=1(n-k)$ است. خواص طراحی این طبقه از کدها، تضمین می‌کند که هر خطای در بردار خطای \bar{e}_t ، که در بردار سندروم $\bar{\sigma}_t$ ظاهر می‌شود، به‌طور خطی مستقل از سندروم‌های ایجادشده توسط بردارهای خطای بعدی $\bar{e}_{t+1}, \bar{e}_{t+2}, \dots, \bar{e}_{t+v}$ است. زمانی که یک خطای توده‌ای به بردار خطای \bar{e}_t محدود شود، تصحیح خطای مجزا کردن تأثیرات خطای توده‌ای، ممکن است.

کاربردهای فشرده‌سازی داده در قالب‌های ۸، ۱۲ یا ۱۶ بیتی ظاهر می‌شوند، کدهای کانولوشن، یک روش حفاظت طبیعی را برای سامانه‌های فشرده‌سازی، ارائه می‌دهند. یک کد کانولوشن با نرخ بالا بر روی حلقه اعداد صحیح متضایر با قالب حسابی، می‌تواند برای تعیین و کنترل مقادیر توازن مورد استفاده قرار گیرد.

۳-۴- کدهای کانولوشن حسابی تصحیح خطای توده‌ای

خطاهای توده‌ای بر روی چندین کلمه کد پخش می‌شوند و توانایی آشکارسازی خطای آن‌ها را از بین می‌برند. کدهای کانولوشن تصحیح خطای توده‌ای دودویی، در آشکارسازی و تصحیح خطاهای محاسباتی داخلی، مؤثر و بسیار مناسب هستند. خطاهای توده‌ای به گروهی از خطاهای هم‌جوار^۱ اشاره دارند که سبب تأثیرات غیرقابل پیش‌بینی خطاهای محاسبات داده‌ها می‌شوند. این کدها نیازمند باندهای محافظه^۲ (مناطق بدون خطای) در قبل و بعد از توده‌های خطاهای هستند؛ بهویژه اگر به تصحیح خطای نیاز باشد. کدهای برلکمپ-پریاراتا-ماسی (BPM)^۳ یک طبقه بزرگ از کدهای تصحیح خطاهای توده‌ای با نرخ بالا هستند که دارای بسیاری از ویژگی‌های مطلوب درخصوص آشکارسازی خطای هستند. از مشخصات طراحی این کدها، صرف نظر از هر باند محافظی که به دنبال وقوع یک توده از خطاهای به وجود می‌آید، آشکارسازی وقوع خطاهای ناشی از خطاهای است. در کدهای BPM نظام‌مند، $k=(n-1)$ بیت اطلاعات به هر زیربلوک n به طول تخصیص داده شده و کد دارای نرخ بالای $n/(n-1)$ است. بهارای هر $(n-1)$ ورودی یک بیت توازن، تولید می‌شود. طول محدودیت این کدها، $(2n-1)=v$ است. ماتریس آزمون توازن B_0 ، درواقع یک ماتریس دودویی $n \times n$ است که دارای یک ماتریس واحد مورب در n سطر بالای خود می‌باشد.

در رابطه (25)، n سطر بالا، ماتریس واحد موربی است که $(n-1)$ مین سطر آن همه صفر است و $(n-1)$ سطر پایین با $(n-1)$ ستون سمت چپ، ماتریس مثلثی پایینی محسوب می‌شود. تمام اطلاعات لازم برای شکل‌گیری ماتریس آزمون توازن نظام‌مند H^{VT} در ماتریس اصلی B_0 گنجانده شده است. بخش مثلثی پایینی این ماتریس با

فصل نهم

¹Possibly Contiguous

²Guard Bands

³Berlekamp-Preparata-Massey Codes



$$\begin{aligned} \bar{s}_{t+p} &= \sum_{q=0}^{m-1} 2^q H^{\nu T} \bar{z}_{t-p}^{(q)}; p \\ &= 0, 1, \dots, \delta \quad (b), \\ \bar{z}_{t+p} &= \begin{pmatrix} \bar{z}_{t-v+p} \\ \bar{z}_{t-v+p+1} \\ \vdots \\ \bar{z}_t \\ \vdots \\ \bar{z}_{t+p} \end{pmatrix}; \\ &\text{بردار به ابعاد } (v+1)n \times 1 \\ &p = 0, 1, \dots, \delta \quad (c). \end{aligned} \quad (32)$$

در نخستین مرحله در فرآيند کدگشایي بازخورد سندروم، 2^0 عبارت از هر $(\delta+1)$ بردار سندروم جدا خواهد شد و در مراحل کدگشایي مبتنی بر باینري موردن استفاده قرار خواهند گرفت. ضرائب 2^0 از هر بردار سندروم \bar{s}_{t+p} با يك کاهش به پيمانه دو استخراج می‌شود.

$$\bar{s}_{t+p}^{(0)} = \left((\bar{s}_{t+p}) \right)_2 \quad p = 0, 1, \dots, \delta. \quad (33)$$

هر روش کدگشایي سندروم دودویي مرتبط با کد کانولوشن دودویي را می‌توان برای یافتن بردار خطای دودویي (0) و بردار ضريب 2^0 خطاهای به کار برد. مرحله بعدی در فرآيند کدگشایي بازخورد، مستلزم حذف تأثير خطای دودویي (0) است؛ با اين وجود ضروري است که يك باند محافظه، \bar{z}_t را در جايی دنبال کند که خطاهای کمتری در سایر مقادير دریافت شده در $\bar{z}_{t+1}, \bar{z}_{t+2}, \dots, \bar{z}_{t+\delta}$ را با $\bar{z}_t - \bar{e}_t^{(0)}$ در تمام اعضای بردار دریافت شده \bar{E}_{t+p} جایگزين کرده، و اين بردارهاي اصلاح شده با يك زيرنويس دوم ۰ مشخص می‌شوند.

$$\bar{E}_{t+p,0}: \bar{z}_{t,0}^{(0)} \leftarrow \left(\bar{z}_t^{(0)} - \bar{e}_t^{(0)} \right). \quad (34)$$

سندروم‌های اصلاح شده بدست آمده نيز نشانه‌گذاري اصلاح شده دارند.

$$\bar{s}_{t+p,0} = H^{\nu T} \bar{z}_{t+p,0}, \quad (35)$$

به پيمانه 2^m

$$\bar{e}_t = \begin{pmatrix} e_{t,1} \\ e_{t,2} \\ \vdots \\ e_{t+n} \end{pmatrix} \Rightarrow \begin{cases} e_{t,1} = s_t \\ e_{t,n-1} = s_{t+1} \\ \vdots \\ e_{t,i} = s_{t+n-i} \\ \vdots \\ e_{t,1} = s_{t+n-i} \end{cases} \quad (28)$$

از طرف ديگر، اگر توده خطای غير صفری در سندروم \bar{S}_t ، يك بردار تمام صفر نیست، حتی اگر توده خطای قابلیت تصحیح کد را از بین ببرد، وقوع خطای قابل آشکارسازی خواهد بود. این قابلیت آشکارسازی، با استفاده از ماتریس C_0^T به ابعاد $n \times 2n$ به دست می‌آید که دارای ویژگی زیر است:

$$C_0^T B_0 = 0_n \quad (29)$$

بنابراین، امكان آزمودن اين که آيا بردار سندروم \bar{S}_t نشان‌دهنده خطاهای قابل تصحیح است یا خير، وجود دارد.

$$C_0^T \bar{e}_t = 0 \quad (30)$$

۴-۴- کدگشایي بازخورد تکراری با سندروم‌ها

کدگشایي سندروم از بردارهای سندروم دریافت شده استفاده می‌کند. گستره‌ای از سندروم‌ها برای کدگشایي نمادهای موجود در بردار دریافت شده \bar{z}_t ، به کار گرفته می‌شود. اين گستره، نشان‌دهنده يك تأخیر کدگشایي است که با δ نشان داده می‌شود. به طور مثال کدهای کانولوشن تصحیح خطای سودهای برلکمپ-پرپاراتا-ماسی، با $\delta = v$ ، کدگشایي می‌شوند. بردارهای سندروم \bar{s}_{t+p} در برآورد بردار خطای \bar{e}_t استفاده می‌شوند.

$$\begin{pmatrix} \bar{s}_t \\ \bar{s}_{t+1} \\ \vdots \\ \bar{s}_{t+\delta} \end{pmatrix} \rightarrow \text{کدگشایي} \rightarrow \text{سندروم‌ها} \quad (31)$$

بردار سندروم کلي \bar{s}_t و نيز بردار داده دریافت شده \bar{z}_t ، در رابطه (۱۴)، بسط داده شده و به يك بردار دودویي منجر می‌شود.

$$\bar{s}_{t+p} = \sum_{q=0}^{m-1} 2^q \bar{s}_{t+p}^{(q)} \quad (a)$$

$$\begin{aligned} \bar{s}_{t+p,1} &= H^{vT} \bar{\Xi}_{t+p,1} & (a) \\ \bar{s}_{t+p,1} &= \bar{s}_{t+p,0} - H_P^T \bar{e}_t^{(1)} & (b) \end{aligned} \quad (40)$$

زمانی که خطاهای بدرستی برآورد شوند، بردار سندروم جدید $\bar{s}_{t+p,1}$ ، هیچ ضریبی از 2^0 یا 2^1 را در بسطهای دودویی خود ندارد.

$$\begin{aligned} \bar{s}_{t+p,1} &= \sum_{q=2}^{m-1} 2^q \bar{s}_{t+p,1}^{(q)}; \\ p &= 0, 1, \dots, \delta. \end{aligned} \quad (41)$$

ضرایب مرتبط با توان 2^2 ، توسط اعمال یک کاهش به پیمانه 2^3 از این سندرومها جدا می‌شوند.

$$\bar{s}_{t+p,1}^{(2)} \cdot 2^2 = ((\bar{s}_{t+p,1}))_2 \cdot 2^2 \quad (42)$$

این کاهش همچنین یک موقعیت به پیمانه دو را برای تولید سندروم مؤلفه $\bar{s}_{t+p,1}^{(2)}$ ایجاد می‌کند. این مؤلفه‌های سندروم دودویی، برای کدگشایی استفاده می‌شوند. مؤلفه‌های بردار خطای $\bar{e}_t^{(2)}$ برای اصلاح بردار داده دریافت شده جاری $\bar{\Xi}_{t+p,1}$ مورد استفاده قرار می‌گیرند.

$$\bar{z}_{t,2} = \bar{z}_{t,1} - \bar{e}_t^{(2)} \cdot 2^2 \quad (43)$$

$$\bar{z}_{t+p,2} \leftarrow \bar{z}_{t+p,1}, \quad \bar{z}_{t,2} \text{ با } \bar{z}_{t,1} \text{ با جایگزین کردن} \quad (44)$$

بردارهای سندروم اصلاح شده $\bar{s}_{t+p,2}$ مرتبط با بردارهای داده دریافت شده و اصلاح شده $\bar{\Xi}_{t+p,2}$ ، هیچ بردار ضریب $2^0, 2^1, 2^2$ یا 2^3 را شامل نمی‌شوند.

$$\begin{aligned} \bar{s}_{t+p,2} &= H^{vT} \bar{\Xi}_{t+p,2}, & (a) \\ \bar{s}_{t+p,2} &= \bar{s}_{t+p,1} - H_P^T \bar{e}_t^{(2)} \cdot 2^2, & (b) \\ \bar{s}_{t+p,2} &= \sum_{q=3}^{m-1} 2^q \bar{s}_{t+p,2}^{(q)}, \quad p = 0, 1, 2, \dots, \delta & (c) \end{aligned} \quad (45)$$

مرحله بعد برای کدگشایی شامل کاهش به پیمانه 2^4 (مرتبط با $\bar{s}_{t+p,2}$) است که به بردار خطای دودویی $\bar{e}_t^{(3)}$ منجر می‌شود. در مراحل کلی بازگشتی در چرخه

اگر خطای $\bar{e}_t^{(0)}$ بدرستی برآورده باشد، آن‌گاه محاسبه سندروم‌های اصلاح شده از راه دیگری نیز امکان‌پذیر است؛ یعنی توسط حذف تأثیرات $\bar{e}_t^{(0)}$ به طور مستقیم با $\{H_i\}, H^{vT}$ به کارگیری زیرماتریس‌های صحیح

$$\begin{aligned} \bar{s}_{t+p,0} &= \bar{s}_{t+p} - H_P^T \bar{e}_t^{(0)} 2^0, \\ 2^m & \text{ به پیمانه } p = 0, 1, \dots, \delta. \end{aligned} \quad (46)$$

در هر دو صورت، واضح است که بردارهای $\bar{s}_{t+p,0}$ ، $\bar{s}_{t+p,1}$ ، هیچ بردار ضریب 2^0 را در بسطهای دودویی خود ندارند.

$$\begin{aligned} \bar{s}_{t+p,0} &= \sum_{q=0}^{m-1} 2^q \bar{s}_{t+p,0}^{(q)}; \\ p &= 0, 1, \dots, \delta. \end{aligned} \quad (47)$$

در سطح بیت بعدی فرآیند کدگشایی، عبارت $\bar{s}_{t+p,0}^{(1)}$ در بسطهای دودویی $\bar{s}_{t+p,0}$ ، با به کاربردن یک کاهش به پیمانه 2^2 مجزا می‌شود.

$$\begin{aligned} \bar{s}_{t+p,0}^{(1)} \cdot 2^1 &= ((\bar{s}_{t+p,0}))_2 = \\ &= ((H^{vT} \bar{\Xi}_{t+p,0}^{(1)}))_2 \cdot 2^1; \quad p = 0, 1, \dots, \delta. \end{aligned} \quad (48)$$

هر روش کدگشایی دودویی را که از بردارهای سندروم دودویی $\bar{s}_{t+p,0}^{(1)}$ استفاده می‌کند، برای تولید بردار خطای دودویی $\bar{e}_t^{(1)}$ می‌توان به کار برد. با این بردار خطای دودویی، اصلاح بردار دریافت شده $\bar{\Xi}_{t+p,0}$ با $\bar{\Xi}_{t+p,1}$ جایگزین کردن $\bar{z}_{t,0} - 2^1 e_t^{(1)}$ با $\bar{z}_{t,1}$ ساده‌تر می‌شود:

$$\bar{\Xi}_{t+p,0} \bar{z}_{t,1} = \bar{z}_{t,0} - 2^1 e_t^{(1)} \quad , \quad \text{به پیمانه } 2^m \quad (49)$$

یک مجموعه اصلاح شده از $(\delta + 1)$ بردار سندروم (یعنی $\bar{s}_{t+p,1}$)، ممکن است، پس از اصلاح از دو راه مشابه توسعه یابد، جایی که دو ضریب نخست بردار خطای \bar{e}_t تعیین و حذف شده‌اند.



به راحتی می‌توان مشاهده کرد که سندروم‌های وابسته $\bar{s}_{t+p,r}$ شامل بسط دودویی با ضریب‌هایی هستند که از توان $2^{(r+1)}$ آغاز می‌شوند.

$$\bar{s}_{t+p,r} = H^{vT} \bar{\Xi}_{t+p,r} \quad p=0,1,2,\dots,\delta \quad (50)$$

این بردارهای سندروم، معادل با بردارهای سندروم

اصلاح شده در مرحله قبل $\bar{s}_{t+p,(r-1)}$ هستند.

$$\bar{s}_{t+p,r} = \bar{s}_{t+p,(r-1)} - H_p^T e_t^{-(r)}. 2^{(r)}, \quad p=0,1,2,\dots,\delta \quad (51)$$

در هر دو مورد، ممکن است در بسطهای دودویی این بردارها، کاهش به پیمانه $2^{(r+2)}$ ، به صورت ضرایب وزنی تفکیک شده $2^{(r+1)}$ نوشته شوند:

$$\begin{aligned} \bar{s}_{t+p,r} &= \sum_{q=r+1}^{m-1} 2^q \bar{s}_{t+p,r}^{(q)} \quad (a) \\ \bar{s}_{t+p,r}^{(r+1)} \cdot 2^{(r+1)} &= \left((\bar{s}_{t+p,r}) \right)_{2^{r+1}} = \\ &\left((H^{vT} \bar{\Xi}_{t+p,r}) \right)_2 \cdot 2^{(r+1)} \quad (b) \end{aligned} \quad (52)$$

در کدگشایی تکراری، هنگامی که $r=(m-1)$ شود، و در نقطه‌ای که تمام مؤلفه‌های بردار خطای \bar{e}_t تعیین شده باشند، کدگشایی خاتمه می‌یابد.

$$\bar{e}_t = \sum_{q=0}^{m-1} 2^q \bar{e}_t^{(q)} \quad (53)$$

تأثیر تمام خطاهای در بردارهای دریافت شده و اصلاح شده $\bar{\Xi}_{t+p,(m-1)}$ ، حذف شده و بردار $\bar{z}_{t,(m-1)}$ ، که یک بردار $1 \times n$ است، نشان‌دهنده داده‌های کدگذاری شده اصلی \bar{y}_t هستند.

کدگشایی بازخورد، فرض بر آن است که بردارهای سندروم اصلاح شده جاری در مرحله $(r-1)$ تعیین شده و همچ ضریبی از $2^0, 2^1, \dots, 2^{(r-1)}$ در بسطهای دودویی سندروم‌های $\bar{s}_{t+p,(r-1)}$ و نیز بردارهای داده دریافت شده و اصلاح شده جاری $\bar{\Xi}_{t+p,(r-1)}$ وجود ندارد.

$$\bar{s}_{t+p,(r-1)} = H^{vT} \bar{\Xi}_{t+p,(r-1)}, \quad 2^m \quad (a)$$

$$\bar{s}_{t+p,(r-1)} = \bar{s}_{t+p,(r-1)} - H_p^T e_t^{-(r-1)} \cdot 2^{(r-1)}, \quad 2^m \quad (b)$$

$$\bar{s}_{t+p,(r-1)} = \sum_{q=r}^{m-1} 2^q \bar{s}_{t+p,(r-1)}^{(q)} \quad p=0,1,2,\dots,\delta \quad (c)$$

(46)

فرض بر این است که بردارهای مؤلفه‌های خطای $\bar{e}_t^{(0)}, \bar{e}_t^{(1)}, \dots, \bar{e}_t^{(r-1)}$ به درستی برآورده شده‌اند. استخراج ضرایب 2^r مرتبط با بسط دودویی $\bar{s}_{t+p,(r-1)}$ امکان‌پذیر بوده و از نخستین ضرایب غیر صفر در هر بسط هستند.

$$\bar{s}_{t+p,(r-1)}^{(r)} \cdot 2^r = \left((\bar{s}_{t+p,(r-1)}) \right)_{2^{r+1}} = \left((H^{vT} \bar{\Xi}_{t+p,(r-1)}) \right)_2 \cdot 2^r \quad p=0,1,2,\dots,\delta \quad (47)$$

کاهش به پیمانه $2^{(r+1)}$ در بردارهای سندروم، $\bar{s}_{t+p,(r-1)}^{(r)}$ از بردار سندروم، بردارهای دودویی فراهم کرده و بردار خطای دودویی $\bar{e}_t^{(r)}$ تعیین می‌شود؛ سپس تأثیر این خطای داده، در t امین گروه از داده‌های دریافت شده حذف می‌شود.

$$\bar{z}_{t,r} = \bar{z}_{t,(r-1)} - e_t^{(r)} \cdot 2^r, \quad 2^m \quad (48)$$

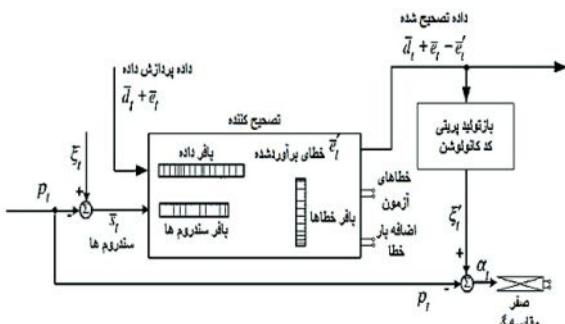
این مقادیر اصلاح شده با جایگزین کردن بردار $\bar{z}_{t,r}$ با بردارهای پیشین $\bar{z}_{t,(r-1)}$ ، در مقادیر دریافت شده و به تازگی اصلاح شده به دست می‌آیند.

$$\bar{\Xi}_{t+p,r} \leftarrow \bar{\Xi}_{t+p,(r-1)} \quad \text{با} \quad \bar{z}_{t,(r-1)} \leftarrow \bar{z}_{t,r} \quad (49)$$

۵- پیاده‌سازی و اجرا

به منظور بررسی دُرستی تکنیک کدگشایی تکراری، برخی آزمایش‌ها صورت گرفته و یک طرح کلی از این روش در ادامه ارائه شده است. مدل‌سازی خطای مجموعه اعطاف‌پذیری از شرایط و تأثیرات خطای فراهم می‌کند. برای مثال ماتریس‌های G و H^{vT} برای $n=4$ در نظر گرفته شده است. ماتریس کدگذاری G ، در سه سطر بالای خود شامل صفر بوده و دارای یک ماتریس واحد 3×3 در

سندروم‌های خطدار همچنین می‌توانند موجب اضافه‌بار تصحیح‌کننده شوند و برآوردهای بسیار بیشتری نسبت به آنچه که قابلیت‌های کد اجازه می‌دهد، تولید کنند. این حالت توسط نشان‌گر اضافه‌بار خطای نشان داده می‌شود شکل (۶). علاوه‌بر این خطای برآورده شده $\{\bar{e}_t'\}$ را می‌توان از داده‌های صحیح، کم کرد، که مقدار $\{\bar{e}_t\} - \{\bar{e}_t'\}$ در خروجی تصحیح کننده حاصل می‌شود. در این مورد، تولید مجدد مقادیر توازن، منجر به $\{\bar{e}_t\} - \{\bar{e}_t'\}$ می‌شود.



(شکل - ۶): حفاظت تصحیح کننده

$$\bar{\xi}_t = \sum_{r=0}^v p_r (\bar{d}_{t-r} - \bar{e}_{t-r}) \quad (54)$$

هم زمان، اگر خطاهای بر مقادیر تولید شده توازن $\{\bar{P}_t\}$ تأثیر نگذارند، آن‌گاه مقادیر توازن $\{\bar{P}_t\}$ صحیح هستند.

$$\bar{p}_t = \sum_{i=0}^v p_i \bar{d}_{t-i} \quad (55)$$

سندروم‌های آزمون خروجی $\{\bar{\alpha}_t\}$ ، به دلیل ماهیت آشکارسازی توده‌ای کد، با وقوع خطای غیر صفر خواهد شد.

$$\bar{\alpha}_t = \bar{\xi}_t - \bar{p}_t = - \sum_{r=0}^v p_r \bar{e}_{t-r} \quad (56)$$

همان‌گونه که بیان شد، از کدهای کانولوشن به‌طور مؤثری برای آشکارسازی خطاهای در سامانه‌های پردازش عددی استفاده شد. الگوریتم‌های کدگشایی دودویی، با یک شیوه بازخورد تکراری مورد استفاده قرار گرفته تا نمادها را با استفاده از روش‌های پردازش سندروم، تصحیح کنند.

راستترین سه ستون است. آخرین سطر G ، عضو خنثی در جمع است:

$$(-1,0,0,-1,-1,0,0,0,-1,0,0,-1,0,-1,-1,0,0,0)$$

ماتریس آزمون توازن H^{VT} که مقدار توازن را تعیین می‌کند، ابعاد $(8 \times v+1) = 2n+8$ است. یک سطر ماتریس آزمون توازن عبارت است از:

$$(1,1,1,0,1,1,0,0,1,0,0,0,0,0,0,1,0,0,0,0,1,0,0,0,0,1)$$

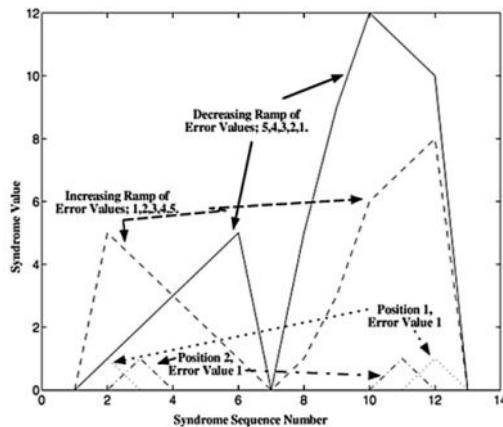
کد شبیه‌سازی، به صورت تصادفی، یک توده خطای در هر بلوک از n نماد Z_2^m درج می‌کند. انتخاب این توده با پارامتر احتمال ϵ کنترل می‌شود. هنگامی که یک خطای توده‌ای رخ می‌دهد، n نماد از Z_2^m ، با استفاده از یک توزیع یکنواخت تولید شده و سپس به n نماد افزوده شده تا یک خطای توده‌ای را مدل‌سازی کند. مراحل شبیه‌سازی، با نرخ‌های متفاوت خطای ϵ اجرا می‌شود. برای یک مقدار به‌طور نسبی بالای ϵ ، توده‌ای خطای آن فقر تکرار می‌شوند که ممکن است گاهی اوقات آن‌ها تا نقص شرایط باند محافظه برای تصحیح، پیش بروند و به تصحیح نادرست (عملکرد نامناسب) منجر شوند. در یک آزمون ساده، به‌هزای $n=5$ با $\epsilon=10^{-3}$ و 10^7 بلوک کلمه کد، نسبت توده‌های ایجاد شده در یک اجرا از 10^7 بلوک کلمه کد، 0.001001 است. احتمال شرطی تصحیح نیز، 0.9791209 است، در حالی که احتمال شرطی تصحیح نادرست، 0.020879 خواهد بود. زمانی که احتمال یک توده خطای $\epsilon=10^{-6}$ کاهش یابد، هیچ تصحیح نادرستی برای اجراهای طولانی وجود نخواهد داشت.

۶- عملیات تصحیح داده

عملیات تصحیح داده، سندروم‌ها و خطاهای مرتبط با تصحیح کننده، در شکل (۶) نشان داده شده‌اند. خروجی‌های آشکارساز خطای، در این شکل شامل: خطاهای آزمون، اضافه بار خطای، و تصحیح نادرست، می‌باشند. داده‌های پردازش شده \bar{d}_t ممکن است شامل خطای \bar{e}_t باشند، که تصحیح کننده، خطای برآورده شده \bar{e}_t' را از آنها کم می‌کند، با این وجود، اگر یکی از بردارهای توازن محاسبه شده $\bar{\alpha}_t$ یا \bar{P}_t ، از یک زیرسamanه خراب تولید شده باشد، آن‌گاه ورودی‌های تصحیح کننده ممکن است، نادرست باشند.

تصحیح کننده، خطاهای در سندروم (S_t) تشخیص داده و به درستی آن‌ها را برآورد می‌کند. از این‌رو، زوج خروجی خطای آزمون، نشان می‌دهد که چه مشکلاتی در کل سامانه وجود دارد؛ هرچند این که داده‌ها صحیح باشند.

خطابه‌شکل جمع‌شونده در خروجی پردازش و زیرسamanه‌های محاسبات توازن، مدل می‌شود). شکل ۷، مقادیر سندروم غیر صفر را نشان می‌دهد، این سندروم‌ها از خطاهای غیر صفر ظاهر شده در خروجی بلوک پردازش در شکل (۲)، نتیجه می‌شوند. مقادیر توازن توسط یک کد کانولوشن حقیقی تصحیح خطای متواتی^۳ (۱۱,۵,۶)، تولید شده‌اند، این کد، بهارای هر پنج نمونه داده ورودی، یک نمونه توازن، تولید می‌کند. چهار پاسخ سندروم متمایز، در شکل (۷)، از خطاهای واحد در موقعیت منفرد، یا شبیه‌های خطا در پنج موقعیت داده، حاصل می‌شوند. یکی از این رشته سندروم‌ها، از یک شبیه خطای افزایشی، با مقادیر خطای ۱، ۲، ۳، ۴، ۵ ایجاد می‌شود؛ در حالی که رشته سندروم دیگر با یک شبیه خطای کاهشی، با مقادیر خطای ۴، ۳، ۵، ۲، ۱، ایجاد می‌شود. دو رشته سندروم دیگر، از خطاهایی با مقادیر واحد، در یک موقعیت منفرد، حاصل می‌شوند، که یکی از سندروم‌ها از خطای موجود در موقعیت یک و سندروم دیگر از خطای در موقعیت دو به دست می‌آید.



شکل-۷: نمایش چهار رشته سندرم متمایز ناشی از الگوهای خطای متفاوت در پردازش، برای کد کانولوشن (۱۱,۵,۶,۱۰). (ردینبو، ۱۹۹۸)

۲-۸- انتخاب مقدار آستانه و عملکرد آشکارسازی خطا

نتایج شبیه‌سازی‌ها نشان می‌دهند که اختلاف میان خطای گردکرن و خطای ناشی از خطای داخلی، به اندازه کافی برای متمایزشدن از یکدیگر بزرگ است. این اختلاف بزرگ میان این دو خطای محدود و گستره‌ای از آستانه‌های آشکارسازی خطا، برای انتخاب، به ما می‌دهد. کوچک‌گرفتن این حد آستانه باعث می‌شود که خطای ناشی

۸- شبیه‌سازی‌ها

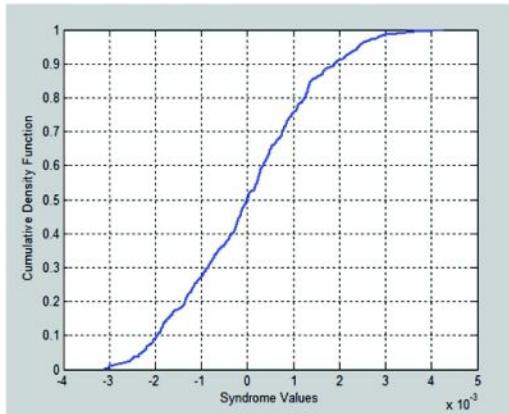
۸-۱- برآوردها و محاسبات

از تفاضل دو بردار توازن \bar{P}_u و \bar{P}_L در شکل ۲، که از دو مسیر مجزا محاسبه می‌شوند، بردار سندرم به دست می‌آید. با صرف نظر از خطای گردکردن، دو مقدار توازن \bar{P}_u و \bar{P}_L می‌باشد در حالت بدون خطای مساوی باشند. از این‌رو آشکارسازی خطای مساوی با مقایسه \bar{P}_u و \bar{P}_L در خروجی مقایسه‌گر حاصل می‌شود. این مقایسه‌گر، تفاضل بین دو مقدار توازن (سندروم S)، را محاسبه می‌کند؛ اگر اندازه مقدار محاسبه شده از مقدار آستانه انتخاب شده، کوچک‌تر باشد، ($\tau < |S|$ و $S = \bar{P}_L - \bar{P}_u$)، آن‌گاه خطای رخداده است؛ (τ مقدار آستانه می‌باشد). ساختار کد کانولوشن حقیقی، به‌گونه‌ای طراحی شده که بهارای الگوهای از خطاهای مختلف که در خروجی بلوک پردازش، ظاهر می‌شوند، سندروم‌های قابل تفکیک و متمایزی را تولید کند (ردینبو، ۱۹۹۸). تفکیک بردارهای سندرم، با وزن همینگ^۱ کد حقیقی، در ارتباط است، (وزن همینگ کد حقیقی کانولوشن مشابه با وزن همینگ یک کد کانولوشن میدان محدود است). در حقیقت در (حمیدی و همکاران، ۲۰۱۲). نشان داده شده است که اگر یک کد میدان محدود، به یک کد حقیقی تبدیل شود، آن‌گاه کد حاصله، ساختار وزن همینگی دارد که اگر بهتر از کد میدان محدود اصلی نباشد، ولی به خوبی آن خواهد بود.

این تمایز میان مقادیر سندرم، در شکل (۷) نمایش داده می‌شود که نشان‌دهنده چهار رشته سندرم ناشی از الگوهای خطای متفاوت در پردازش، برای کد کانولوشن (۱۱,۵,۶,۱۰) است (حمیدی و همکاران، ۲۰۱۴). این کد، برای هر $k=5$ نمونه ورودی، یک مقدار سندرم تولید می‌کند و یک طول محدودیت $m+1=12$ نمونه‌ای را به کار می‌برد. زمانی که سندروم، دارای مقدار کوچک‌تر از مقدار آستانه باشد، در آن صورت مدل‌های خطای در شکل (۲) دارای واریانس θ برای خطای A و واریانس ϕ برای خطای B هستند، (مقدار آستانه، پنج برابر انحراف معیار مربوط به خطای گردکردن در حالت بدون خطای در نظر گرفته می‌شود). حال اگر مقدار سندروم از مقدار آستانه تجاوز کند، درواقع خطای رخداده است؛ که در این حالت واریانس خطاهای A و B به ترتیب با θ و ϕ نشان داده می‌شوند. که اندازه واریانس‌های θ و ϕ چندبرابر اندازه واریانس‌های θ و ϕ است، (در تمام حالتها

^۱ Hamming Weight

برای درک بهتر محدوده خطاهای ناشی از گرد کردن مقادیر سندروم در حالت بدون اعمال خطا، نمودار تابع توزیع تجمعی خطا (CDF)^۱ در این حالت، در شکل (۹)، ترسیم شده است.



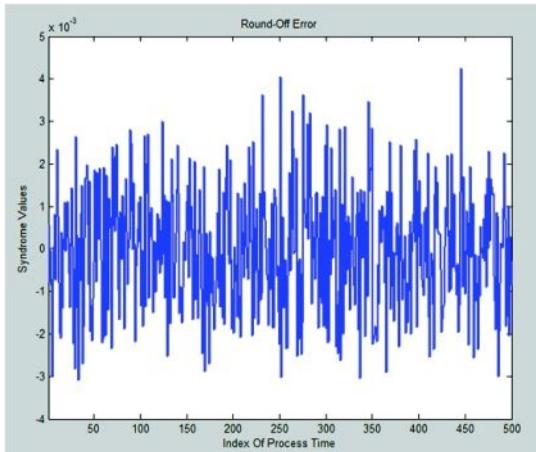
(شکل-۹): نمودار تابع توزیع تجمعی خطا

با توجه به شکل (۹)، مشاهده می شود که احتمال آن که مقادیر سندروم در حالت بدون اعمال خطا، کمتر از 3×10^{-3} باشند، حدود ۰.۰۱ است و احتمال آنکه مقادیر سندروم بیشتر از $+3 \times 10^{-3}$ باشند، حدود $0.014 \pm 3 \times 10^{-3}$ است. همچنین احتمال آنکه خطاهای در محدوده $\pm 3 \times 10^{-3}$ باشند، برابر با 0.976 ± 0.001 است. میانگین مربعات خطا و بیشینه میانگین مربعات خطاهای به دست آمده از پانصد بار تکرار، در شکل های (۱۰) و (۱۱)، مورد ارزیابی قرار گرفته اند. برای شبیه سازی و به دست آوردن این نمودارها، ده هزار نمونه داده در هر پانصد بار تکرار، تولید شده و خطای گرد کردن هر یک از نمونه ها محاسبه شده است، مقدار میانگین مربعات خطا در شکل (۱۰) و بیشینه میانگین مربعات خطا در شکل (۱۱) نشان داده شده است. نمودار میانگین مربعات خطاهای در برابر سطوح مختلف خطا، در شکل (۱۲) نشان داده می شود. در این شبیه سازی ابتدا یک سطح خطا مشخص شده و در هزار تکرار که در هر تکرار هزار نمونه تولید می شود، خطا در نمونه های مختلف، به طور تصادفی و مستقل از نمونه های قبلی تولید و سپس متوسط مربعات خطاهای در تمام تکرارها محاسبه شده و این کار برای سطوح خطاهای بعدی نیز تکرار می شود (مقدار آستانه 5×10^{-3} در نظر گرفته شده است). عملکرد آشکار سازی خطا در برابر سطح آستانه، زمانی که خطا در سامانه خطی پردازش داده، رخ می دهد، در شکل

از گرد کردن اعداد نیز به عنوان خطای ناشی از نقص (خطای داخلی) تلقی شود و از سوی دیگر بزرگ انتخاب کردن آن، باعث نادیده گرفتن برخی از خطاهای کوچک ناشی از خطاهای داخلی خواهد بود.

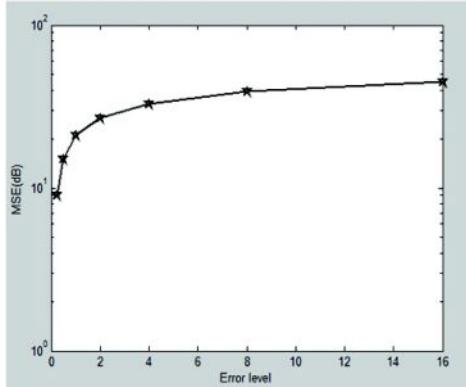
با استفاده از شبیه سازی، چگونگی انتخاب مقدار آستانه را برای آشکار سازی خطاهای، بررسی می کنیم. برای ارزیابی قابلیت آشکار سازی خطا، در ساختار پیشنهادی، به عنوان مثال، از کد کانولوشن حقیقی نظام مند (۳،۲،۲)، با چند جمله ای های مولد $g_1^{(3)}(D)=1+D$ و $g_2^{(3)}(D)=1+D^2$ استفاده شده است. برای انجام این شبیه سازی، ده هزار نمونه تصادفی با توزیع گوسی با میانگین صفر و انحراف معیار ۱۰ تولید شده و هر نمونه به طور متوسط به طور تصادفی پانصد بار انتخاب می شود. همچنین، خطا توسط یک توزیع گوسی با میانگین صفر و واریانس ثابت، تولید می شود.

شکل (۸) نتیجه شبیه سازی را بعد از پانصد بار تکرار نشان می دهد؛ در این شکل، رشته سندروم در حالت بدون خطای نشان داده شده و مقادیر غیر صفر، مربوط به خطاهای گرد کردن می باشند. مقدار سندروم ها در بیشتر موارد کمتر از $\pm 3 \times 10^{-3}$ است؛ اما اندازه تعداد کمی از خطاهای بزرگ تر از $\pm 3.5 \times 10^{-3}$ است. بنابراین مقدار آستانه $3.5 \times 10^{-3} = 17.5 \times 10^{-3}$ (۳.۵ هزار نمونه) به منظور آشکار سازی خطا انتخاب می شود؛ یعنی پنج برابر مقدار حداقل سندروم، در حالت بدون تزییق خطا، به عنوان مقدار آستانه انتخاب شده است. برای آستانه های کوچک، خطاهای بزرگ، به راحتی آشکار سازی می شوند؛ در حالی که آستانه های بزرگ، عملکرد آشکار سازی خطا را کاهش می دهند.

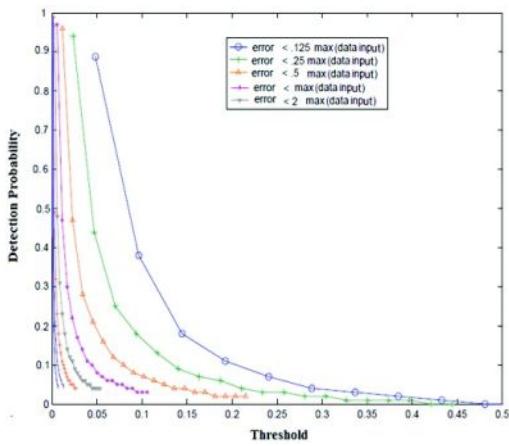


(شکل-۸): رشته سندروم برای حالت بدون خطا

^۱ Cumulative Density Function



(شکل - ۱۲): میانگین مربعات خطا در مقابل سطح تزریق خطا

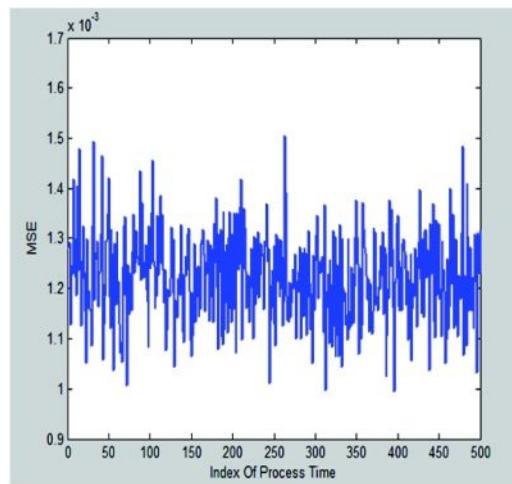


(شکل - ۱۳): عملکرد آشکارسازی خطا در مقابل آستانه

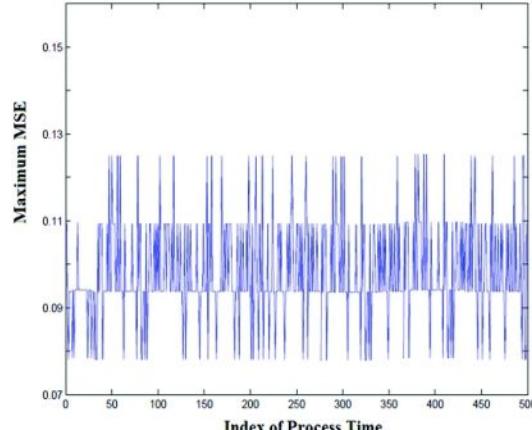
هر چه سطح تزریق خطا بیشتر باشد، درصد و احتمال آشکارسازی خطا بیشتر است؛ که این موضوع در شکل (۱۴) نیز قابل مشاهده است. درحالی که سطح تزریق خطا از شانزده برابر حداقل مقدار داده ورودی تجاوز نمی‌کند؛ به عبارت دیگر سطح تزریق خطا، گوچک‌تر از شانزده برابر حداقل مقدار داده ورودی می‌باشد شکل (۱۴-الف)، درصد آشکارسازی خطا نزدیک به صد درصد (۰/۹۹۹۷۰۰)، جدول (۱) است؛ ولی زمانی که سطح تزریق خطا از یک دوم حداقل مقدار داده ورودی تجاوز نمی‌کند به کمترین مقدار درصد آشکارسازی خطا می‌رسد شکل (۱۴).

عملکرد آشکارسازی خطا سامانه پردازش داده، در زمانی که سطح تزریق خطا از ۸، ۴، ۱، ۰/۵، ۰/۱۶ برابر حداقل مقدار داده ورودی تجاوز نمی‌کند، در شکل (۱۴) نشان داده شده است. سطح حداقل خطا برای هر تکرار، تنها یکبار تغییر می‌کند؛ همانطور که در شکل (۱۳) نیز مشاهده می‌شود، سطوح خطا ۱۶، ۸، ۴، ۱، ۰/۵، ۰/۱۶ برابر حداقل مقدار داده ورودی است، که برای هر تکرار، سطح بیشینه خطا متفاوت

(۱۳) نشان داده می‌شود. همان‌گونه که در این شکل مشاهده می‌شود، با افزایش مقدار آستانه، عملکرد آشکارسازی، کاهش می‌یابد. به ازای یک مقدار معین آستانه، هر چه دامنه خطا بیشتر باشد، احتمال عدم آشکارسازی خطا، کمتر خواهد بود. به عبارت دیگر در چنین شرایطی احتمال آشکارسازی صحیح خطا، بیشتر می‌شود.



(شکل - ۱۰): میانگین مربعات خطا



(شکل - ۱۱): ماکریم میانگین مربعات خطا

همچنین هر چه مقدار آستانه بیشتر باشد، درصورتی که سطح خطا ثابت بماند، احتمال آشکارسازی کمتر خواهد بود. این نمودارها به ازای پانصد هزار نمونه داده، استخراج شده‌اند (به شکل پانصد تکرار که در هر تکرار هزار داده با توزیع گوسی تولید شده است). سطوح خطا ۸، ۴، ۱، ۰/۵، ۰/۲۵، ۰/۱۲۵ برابر حداقل مقدار داده ورودی، اعمال شده، و مقادیر آستانه از 10^{-5} تا 10^{-6} تغییر می‌کنند.

منعکس می‌کنند. اگر آستانه خطابه اندازه کافی و مناسب کوچک در نظر گرفته شود، در این صورت اکثر خطاهای توسعه مقایسه‌گر آشکارسازی می‌شوند؛ اما اگر، آستانه خطابه بسیار کوچک انتخاب شود، مقایسه‌گر، خطاهای گردکردن را نیز به عنوان خطاهای محاسباتی در نظر می‌گیرد. بنابراین ما به پیداکردن یک مقدار آستانه مناسب نیاز داریم که خطاهای ناشی از محدودیت دقت محاسبه‌گر (خطاهای گردکردن) را از خطاهای داخلی، تفکیک کند.

شکل (۱۶)، عملکرد آشکارسازی خطابه در برابر مقادیر آستانه نشان می‌دهد. در مقادیر کوچک آستانه، مقایسه‌گر بیشتر خطاهای را تشخیص می‌دهد. عملکرد آشکارسازی خطای سامانه زمانی که مقدار آستانه بزرگ می‌شود، بسیار بد است. برای انجام این شبیه‌سازی ابتدا یک سطح آستانه انتخاب و سپس خطابه در سطوح مختلف به طور تصادفی طبق جدول (۲) به داده‌ها اضافه می‌شود و در پایان تعداد نمونه‌هایی که خطابه در آنها وجود داشته، ولی تشخیص داده نشده است، شمارش شده و احتمال عدم آشکارسازی خطابه، از تقسیم این تعداد بر کل داده‌هایی که خطابه در آنها بوده به دست می‌آید.

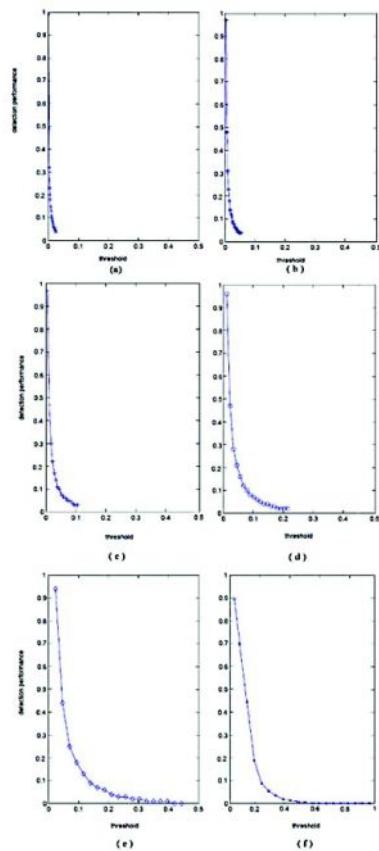
۳-۸ مقایسه

در کار انجام‌شده توسط Redinbo، در سال ۲۰۱۱، یک ساختار بهینه برای آشکارسازی و تصویح خطاهای ناشی از خرابی‌های سخت‌افزاری متناوب در عملیات‌های پردازش خطی، توسعه یافته است. در این روش، یک فیلتر پیش‌گویی کالمون، اصلاح خطابه را از روی رشته سندروم، بر عهده دارد. اگر چه روش پیشنهادشده بر اساس تخمین گر کالمون می‌تواند با دقت بالا، خطابه در بلوک پردازش خطی را تشخیص و اصلاح کند؛ اما در صورت بروز خطابه در سایر نواحی غیر از بلوک پردازش خطی، تنها قادر است آنرا تشخیص دهد، ولی خروجی سامانه در این شرایط غیر قابل اعتماد است. از آنجایی که بار محاسباتی سایر نواحی نیز در مقابل بلوک پردازش خطی کم نبوده و از نظر پیچیدگی سخت‌افزار نیز قابل مقایسه است (زیرا نیازمند انجام محاسبات جمع و ضرب ممیز شناور هستند)، بنابراین احتمال بروز خطابه در این نواحی چندان کمتر از بلوک پردازش اصلی نیست و نمی‌توان آنها را مصون از خطابه فرض کرد. رشته سندروم به علت وجود خطاهای گردکردن اعداد، حتی در صورت عدم وجود خطابه، مخالف صفر است. این رشته در این حالت به عنوان یک نوافه سفید با میانگین و واریانس مشخص تلقی می‌شود.

در نظر گرفته شده است. هنگامی که حداقل سطح خطابه بزرگ باشد، عملکرد آشکارسازی خطابه سرعت در مقابل افزایش آستانه آشکارسازی، کاهش می‌یابد، که در شکل (۱۴) نشان داده شده است. آخرین نمودار در شکل (۱۴)، منحنی عملکرد آشکارسازی خطابه را برای سطح خطایی که کوچکتر از یک هشتاد حداقل مقدار داده ورودی است، نشان می‌دهد، میانگین مربعات خطابه در این حالت 98376 dB یا -0.71076 dB می‌باشد. مقدار آستانه می‌تواند بر اساس شرایط خطابه و منحنی عملکرد آشکارسازی خطابه، انتخاب شود جدول (۱). اگر مقدار آستانه T ، بسیار کوچک تعیین شود، آن گاه حتی خطاهای گردکردن نیز به عنوان خطاهای داخلی در نظر گرفته می‌شود؛ که در این صورت، به محاسبات مجدد غیرضروری خواهد انجامید. در حالت کلی، صرف نظر کردن از تعداد کمی از خطاهای بسیار کوچک که در محدوده سطح خطای گردکردن می‌باشند، به طور عمومی مجاز است. به هر حال، شبیه‌سازی نشان می‌دهد که انتخاب مقدار آستانه می‌تواند، تشخیص و یا عدم تشخیص خطابه را تحت تأثیر قرار دهد. خطاهای توسط آزمودن اندازه سندروم‌ها و مقایسه آنها در مقابل آستانه‌هایی که مقدارشان پنج برابر اندازه انحراف معیار مقدار سندروم مربوط به خطاهای گردکردن هستند، آشکارسازی می‌شوند. برنامه شبیه‌سازی به صورت تصادفی، مؤلفه خطای جمع‌شونده را تولید می‌کند، که اندازه و بزرگی هر خطابه از یک توزیع گوسی با میانگین صفر و واریانس ثابت، انتخاب می‌شود.

برای مقادیر آستانه کوچک، تعداد خطاهای زیادی آشکارسازی می‌شوند؛ در حالی که برای مقادیر آستانه بزرگ بازدهی آشکارسازی خطابه کاهش می‌یابد. تغییر مقدار آستانه از یک مقدار کوچک به یک مقدار بزرگ سبب گذار از یک سطح بالای آشکارسازی خطابه، به یک سطح پایین از آشکارسازی خطابه خواهد بود. به هر حال، برای شبیه‌سازی، توانایی آشکارسازی خطابه، تأثیر مستقیمی با واریانس خطاهای تزریق شده دارد. همان‌گونه که بیان شد، با مقایسه تفاضل دو مقدار توازن \bar{P}_1 و \bar{P}_2 شکل (۱۵)، می‌توان پاسخ سامانه به خطابه را بررسی کرد. شکل (۱۵) چگونگی تأثیر خطاهای بر روی خروجی مقایسه‌گر را نشان می‌دهد. شکل ۱۵-الف یک تفاضل بسیار کوچک میان دو مقدار توازن \bar{P}_1 و \bar{P}_2 را نشان می‌دهد. دلیل اختلاف غیر صفر، خطاهای گردکردن ناشی از پاسخ‌های محدود سامانه محاسباتی است. در شکل (۱۵-ب)، مقادیر $|\bar{P}_1 - \bar{P}_2|$ تأثیر خطاهای رخداده شده را

تکنیک های پردازش داده ادغام شده و به تحمل پذیری مؤثرتر خطا می انجامد. سامانه پردازش داده از طریق مقادیر توازن تعیین شده با کدهای کانولوشن تعریف شده بر روی اعداد صحیح محافظت می شود. مقایسه تووانه های محاسبه شده، از دو مسیر آشکارسازی خط را فراهم می آورد، در حالی که تصحیح داده های خروجی توسط یک کدگشای (کدبردار) آستانه که تحت تأثیر خطاهای گرد کردن و خطاهای ناشی از محاسبه گر است، صورت می گیرد. در این مقاله رویکردی برای به کار گیری کدهای کانولوشن بکار برده شده در فرم های نظام مند ارائه می دهیم، نتایج شبیه سازی، نشان می دهند که اختلاف میان خطای گرد کردن و خطای ناشی از محاسبه گر به اندازه کافی برای تمایز شدن از یکدیگر بزرگ است. این اختلاف بزرگ، میان این دو خطا، محدود و گستره ای از آستانه های آشکارسازی خط را برای انتخاب به ما می دهد. در ادامه، احتمال آشکارسازی و عدم آشکارسازی خط برای برخی از کدهای کانولوشن نشان داده است. عملکرد آشکارسازی در یک سامانه پردازش اطلاعات به آستانه آشکارسازی بستگی دارد که با خطای گرد کردن تعیین می شود.



(شکل ۱۴): عملکرد آشکارسازی خط: (a) در زمانیکه سطح تزریق خطا از ۱۶ برابر حداقل مقدار داده ورودی تجاوز نمی کند، (b) در زمانی که سطح تزریق خطا از هشت برابر حداقل مقدار

(جدول ۱): عملکرد آشکارسازی خط و میانگین مربعات خط

میانگین مربعات خط (dB)	میانگین مربعات خط	احتمال آشکارسازی خط	سطح خط
45.14869076514	$3.27242028818 \times 10^4$	0.99997	16
39.09163378535	$8.11266193102 \times 10^3$	0.99985	8
33.147958056082	$2.06440929402 \times 10^3$	0.99981	4
27.06071061413	$5.08242596927 \times 10^2$	0.99966	2
21.05157746866	1.2739657339×10^2	0.99918	1
15.04809777764	31.974942912151	0.99848	0.5
8.99879804144	7.9410842591893	0.9970	0.25
0.071076467690	0.9837672332412	0.9937	0.125

واریانس این نویه در کار آقای Redinbo 10^{-1} در نظر گرفته شده است. هنگامی که یک خط را در سامانه رخداده، رشته سندروم مقادیر بزرگی را نشان خواهد داد. فرض شده که مقدار واریانس در این حالت برابر هزار است؛ بنابراین باندara کافی از خطای گرد کردن متایز فرض می شود. نتایج ارائه شده در کار آقای Redinbo ، نشان می دهد که قدر مطلق خطای تصحیح، حدود 10^{-5} است؛ در حالی که در ساختار ارائه شده در این مقاله ، این خط در حدود 10^{-4} است. علاوه براین، از آنجایی که خود فیلتر کالمن و مسیر ایجاد توازن، مصون از خط نیستند، برای حمایت از آنها لازم است از سخت افزارهای جانبی بهره گرفت؛ اما، در ساختار ارائه شده، مدار کدگشا و اصلاح خط، با توجه به اینکه از عناصر بسیار ساده استفاده می کنند، مصون از خط هستند. پس نیازی به سخت افزار جانبی وجود ندارد.

۹- نتیجه گیری و پژوهادها

در این مقاله ، ساختار ارائه شده به حفاظت از هر شکل کلی از عملیات پردازش خطی می پردازد. برای اهداف تصحیح، از افزونگی استفاده می شود که به این منظور، کدهای کانولوشن به کار گرفته می شوند. کدهای کانولوشن تعریف شده بر روی اعداد صحیح، یک ساختار ریاضی برای محاسبات ممیز ثابت هستند، این کدها، کدهای کانولوشن دودویی شناخته شده ای را به عنوان مولدهای اصلی شان به کار می برنند. این طبقه از کدهای کانولوشن حسابی از نمادهای برگرفته از حلقة محدود، بر مبنای کدهای تصحیح توده ای دودویی هستند. هدف، توصیف یک رویکرد حفاظتی است که به آسانی با

^۱ سطح خط نسبت به حداقل مقدار داده ورودی

(جدول - ۲): پارامترهای آزمایش عملکرد عدم آشکارسازی خطا در مقابل آستانه برای شکل (۱۶)

کد	خطا	تعداد نمونه‌های ورودی (در هر تکرار)	مقدار آستانه (τ)
(3.2.2)	توزیع گوسی با میانگین ۰ و انحراف معیار ۲ (به ازای هر ۵ نمونه یک خطای اضافه می‌شود)	10^3 تعداد تکرار (۱۰۰۰)	۰ - ۱۶ (۰.۰۵ کام)

تعیین ساختار ارائه شده در این مقاله با استفاده از کدهای موجک، جهت کاربردهای فشرده‌سازی داده و پردازش تصویر پیشنهاد می‌شود. در این راستا مهم‌ترین مسئله، یافتن رویکرد تحلیلی مناسب و ساده، برای اندازه‌گیری دقیق عملکرد آشکارسازی خطا در کاربردهای تحمل پذیری خطای مبتنی بر الگوریتم، با استفاده از کدهای موجک، است؛ زیرا هیچ رویکرد تحلیلی آسانی برای اندازه‌گیری دقیق عملکرد آشکارسازی خطا با استفاده از کدهای موجک وجود ندارد.

۱۰- مراجع

Acree, R. K., Ullah, N., Karia, A., Rahmeh, J. T., & Abraham, J. A. 1993. "An object-oriented approach for implementing algorithm-based fault tolerance." In Proceedings of the 12th Annual International Phoenix Computers and Communications Conference (pp. 210-216).

Ashouei, M., & Chatterjee, A. 2009. "Checksum-based probabilistic transient-error compensation for linear digital systems." IEEE Transactions on Very Large Scale Integration Systems, 17(10), 1447-1460. doi:10.1109/TVLSI.2008.2004587

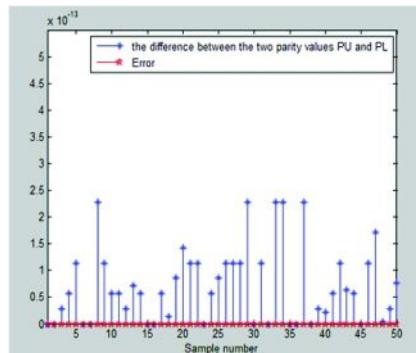
Banerjee, P., Rahmeh, P. J. T., Stunkel, C. B., Nalr, V. S. S., Roy, K., & Abraham, J. A. 1990. "Algorithm-based fault tolerance on a hypercube multiprocessor." IEEE Transactions on Computers, 39(9), 1132-1145. doi:10.1109/12.57055

Baylis J. 1998. "Error-correcting codes: A mathematical introduction." New York, NY: Chapman and Hall.

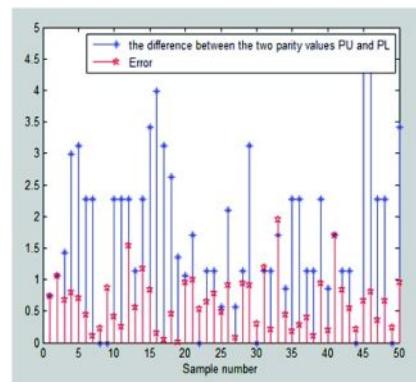
Berlekamp ER, 1962, "class of convolution codes." Inf Control. 6, 1-13.

Bosilca G, Delmas R, Dongarra J, Langou J, 2009, "Algorithm-based fault tolerance applied to high performance computing." J Parallel Distrib Comput. 69(4), 410-416 . doi:10.1016/j.jpdc.2008.12.002

داده ورودی تجاوز نمی‌کند، (c) در زمانی که سطح تزریق خطا از چهار برابر حداقل مقدار داده ورودی تجاوز نمی‌کند، (d) در زمانی که سطح تزریق خطا از دو برابر حداقل مقدار داده ورودی تجاوز نمی‌کند، (e) در زمانی که سطح تزریق خطا، برابر حداقل مقدار داده ورودی باشد، (f) در زمانی که سطح تزریق خطا از ۰.۵ برابر حداقل مقدار داده ورودی، تجاوز نمی‌کند.



(a)

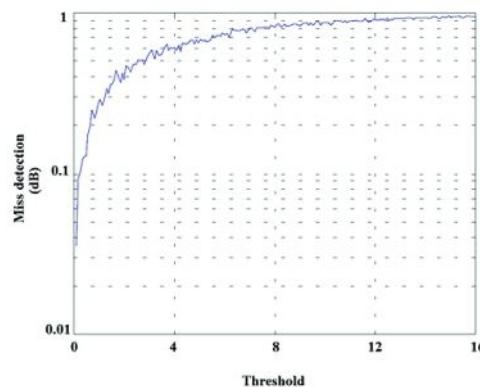


(b)

(شکل - ۱۵): پاسخ به خطاهای تزریق شده در عملیات پردازش

داده: (a) بدون تزریق خطا، (b) مقادیر تزریق خطا به همراه

مقادیر تفاضل میان دو مقدار توازن \bar{P}_L و \bar{P}_U



(شکل - ۱۶): عملکرد عدم آشکارسازی خطا در مقابل آستانه

- Redinbo GR, 2003, "Failure-detecting arithmetic convolution codes and an iterative correcting strategy." *IEEE Trans Comput.* 52(11), 1434–1442. doi:10.1109/TC.2003.1244941
- Redinbo GR, 2010, "Wavelet codes for algorithm-based fault tolerance applications." *IEEE Trans Depend Secure Comput.* 7(3), 315–328.
- Redinbo GR, 2011, "Systematic wavelet Sub codes for data protection." *IEEE Trans Comput.* 60(6), 904–909.
- Rexford, J., & Jha, N. K. 1992. "Algorithm-based fault tolerance for floating-point operations in massively parallel systems." In Proceedings of the International Symposium on Circuits and Systems (pp. 649-652).
- Roche T, Cunche M, Roch JL, 2009, "Algorithm-based fault tolerance applied to P2P computing networks." *ap2ps*, 2009 First International Conference on Advances in P2P Systems 144–149.
- Salfner, F., Lenk, M., & Malek, M. 2010, "A survey of online failure prediction methods." *ACM Computing Surveys*, 42(3). doi:10.1145/1670679.1670680
- Veeravalli V. S. 2009. "Fault tolerance for arithmetic and logic unit." In Proceedings of the IEEE Southeast Conference (pp. 329-334).
- Viterbi AJ, Omura JK, 1985, "Principles of Digital Communication and Coding." (Mc-Grawhill).
- Costello D., & Lin S. 2004. "Error control coding fundamentals and applications" (2nd Ed.). Upper Saddle River, NJ: Pearson Education.
- Hamidi, H., Vafaei, A., & Monadjemi, A. H., 2010. "A fault-tolerant approach for matrix functions in image processing." Paper presented at the 6th Iranian Machine Vision and Image Processing Conference.
- Hamidi, H., Vafaei, A., & Monadjemi, A. H. 2011, "A framework for ABFT techniques in the design of fault-tolerant computing systems." *EURASIP Journal on Advances in Signal Processing*, 1, 90. doi:10.1186/1687-6180-2011-90
- Hamidi, H., Vafaei, A., & Monadjemi, A. H. 2012., "Analysis and evaluation of a new algorithm based fault tolerance for computing systems.", *International Journal of Grid and High Performance Computing*, 4(1), 37–51. doi:10.4018/jghpc.2012010103
- Hamidi, H., Vafaei, A., & Monadjemi, A. H., 2009. "Algorithm based fault tolerant and checkpointing for high performance computing systems." *Journal of Applied Science*, 9, 3947–3956. doi:10.3923/jas.2009.3947.3956
- Hamidi, H. 2014., "A General Framework of Algorithm-Based Fault Tolerance Technique for Computing Systems.", In H. Nemat (Ed.), *Analyzing Security, Trust, and Crime in the Digital World* (pp. 1-21). Hershey, PA.: doi:10.4018/978-1-4666-4856-2.ch001
- Huang, K. H., & Abraham, J. A. 1984. "Algorithm-based fault tolerance for matrix operations." *IEEE Transactions on Computers*, 33, 518–528. doi:10.1109/TC.1984.1676475
- Jou, J. Y., & Abraham, J. A. 1986. "Fault-tolerant matrix arithmetic and signal processing on highly concurrent computing structures." *Proceedings of the IEEE*, 74(5), 732–741. doi:10.1109/PROC.1986.13535
- Jou, J. Y., & Abraham, J. A. 1988. "Fault-tolerant FFT networks." *IEEE Transactions on Computers*, 37, 548–561. doi:10.1109/12.4606
- Lee LHC, 1997, "Coding: Fundamentals and Applications", (Artech House)
- Massey JL, 1965, "Implementation of burst-correcting convolution codes." *IEEE Trans Inf Theory*. 11, 416–422. doi:10.1109/TIT.1965.1053798
- Morelos-Zaragoza, R. H. 2006. "The art of error correcting coding." (2nd Ed.). New York, NY: John Wiley & Sons. doi:10.1002/0470035706
- Nair, V. S. S., & Abraham, J. A. 1990. "Real number codes for fault-tolerant matrix operations on process arrays." *IEEE Transactions on Computers*, 426–435. doi:10.1109/12.54836
- Redinbo GR, 1998, "Generalized algorithm-based fault tolerance: error correction via Kalman estimation." *IEEE Trans Comput.* 47(6), 1864–1876.

حجت‌الله حمیدی مدرک کارشناسی و



کارشناسی ارشد را در مهندسی کامپیوتر در سال ۱۳۷۹ از دانشگاه علم و صنعت، و دکترای خود را در رشته مهندسی کامپیوتر در سال ۱۳۹۱ از دانشگاه اصفهان دریافت کرده است. وی در حال حاضر استادیار دانشگاه صنعتی خواجه نصیرالدین طوسی است. زمینه‌های پژوهشی مورد علاقه ایشان کدگذاری اطلاعات، امنیت اطلاعات، کسب و کار هوشمند و محاسبات نرم است.

نشانی رایانمۀ ایشان عبارت است از:

h_hamidi @kntu.ac.ir