

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

حجت‌اله 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 به‌کارگیری کدهای کانولوشن نظام‌مند تعریف‌شده بر روی ساختارهای حسابی نقطه ثابت، آشکارسازی و تصحیح می‌شوند. کدهای کانولوشن، امکان آشکارسازی خطا در عملیات پردازش داده عددی را، فراهم می‌کنند. بسیاری از تکنیک‌های تحمل‌پذیر خطا که مربوط به حفاظت از عملیات‌های ماتریسی عددی هستند از کدهای بلوکی با افزونگی بالا برای قابلیت تصحیح خطا، استفاده می‌کنند. در این مقاله، از کدهای کانولوشن استفاده شده و از ویژگی‌های آزمون پیوسته برای آشکارسازی خطاها استفاده می‌شود. این تکنیک آشکارسازی خطای مبتنی بر الگوریتم، بر مقایسه مقادیر توازن محاسبه‌شده 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 خطا است. این مقاله طرح محاسباتی جدیدی را به‌منظور فراهم آوردن تحمل‌پذیری خطا برای الگوریتم‌های عددی ارائه می‌دهد. درحالی‌که تصحیح داده‌های خروجی با یک کدگشای (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 خطا می‌انجامد. 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 خطا، محدوده و گستره‌ای از آستانه‌های آشکارسازی خطا را برای انتخاب به ما می‌دهد.

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

## ۱- مقدمه

روش‌های متعددی برای افزایش قابلیت اطمینان<sup>۱</sup> و تحمل‌پذیری خطای سامانه‌های محاسباتی در سال‌های اخیر ارائه شده است (اسری و همکاران، ۱۹۹۳؛ بنرجی و همکاران، ۱۹۹۰). با این همه به‌علت افزایش پیچیدگی و هزینه‌های هنگفتی که طراحی سامانه‌های مطمئن و تحمل‌پذیر خطا به همراه دارند، بسیاری از سامانه‌های موجود یا به‌طور کامل فاقد امکانات سخت‌افزاری لازم برای حفظ تحمل‌پذیری خطا هستند و یا در صورت وجود، تنها ویژگی‌های معدودی را حمایت می‌کنند (هانگ و همکاران،

۱۹۸۴؛ آشویی و همکاران، ۲۰۱۰؛ بیلینس، ۱۹۹۸؛ جویی و همکاران، ۱۹۸۶).

پردازش خطای اجراشده با پردازش‌گرهای تعبیه‌شده، در بسیاری از سامانه‌های عملی، دارای نقش محوری است. مثال‌های مهم از این پردازش‌ها، شامل پردازش سیگنال دیجیتال پرسرعت در ماهواره‌های مخابراتی<sup>۲</sup>، الگوریتم‌های محاسباتی ماتریسی بزرگ و سامانه‌های کنترل تولید<sup>۳</sup> که فعالیت‌های هم‌زمان<sup>۴</sup> بسیاری را هماهنگ می‌کنند، می‌شوند. از آنجایی‌که خطاها می‌توانند منجر به یک عملکرد غیرقابل قبول شوند، تحمل‌پذیری خطا، یک مسأله پراهمیت است؛

<sup>2</sup> Communication

<sup>3</sup> Manufacturing

<sup>4</sup> Simultaneous

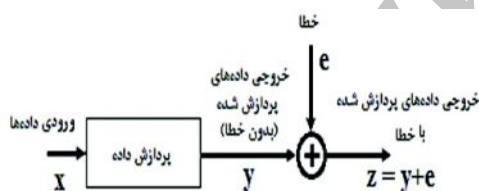
<sup>1</sup> Reliability



تحت شرایط بدون خطا، مقادیر توازن  $P_L$  به‌طور تقریبی برابر با مقادیر توازن  $P_U$  هستند. در روش تحمل‌پذیری خطای مبتنی بر الگوریتم، هنگامی خطا آشکارسازی می‌شود که تفاضل این دو مقدار توازن، از مقدار آستانه، بیشتر شود. اگر خطا در یکی از سه زیرسامانه موجود در بلوک دیاگرام شکل (۲) وجود داشته باشد، آن‌گاه خطای مدل‌سازی شده در خروجی آن زیرسامانه، به‌طور قابل‌ملاحظه‌ای افزایش خواهد یافت و موجب می‌شود که دو رشته توازن، با یکدیگر اختلاف داشته باشند.

### ۳- مدل خطا در پردازش داده

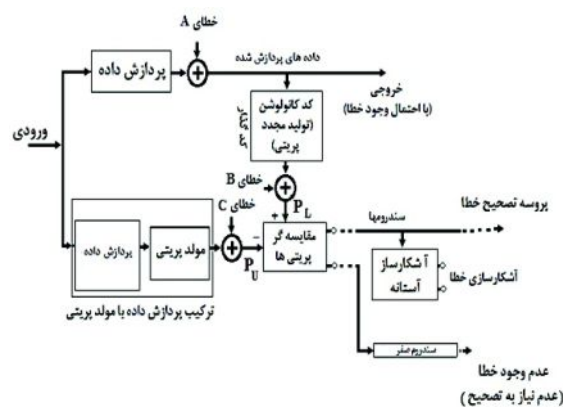
یکی از ویژگی‌های تکنیکی در تحمل‌پذیری خطا در کاربردهای پردازش سیگنال و پردازش تصویر، استفاده از مدل‌های سطح بالای خطا است. خطا به‌طور عمومی به شکل خطای جمع‌شونده و منفرد گذرا، در خروجی مدل می‌شود. در طول همه زمانی که اثر این خطا در خروجی وجود دارد خطای دیگری در سامانه رخ نمی‌دهد. تأثیر خطا بر روی مؤلفه‌های کلمه کد (خروجی کدگذار کانولوشن) با افزودن یک بردار خطای  $e$  مدل‌سازی می‌شود شکل (۳). به بیان دیگر، خطا به‌صورت افزایشی (جمع شونده) در خروجی مدل‌سازی می‌شود.



(شکل-۳): مدل خطا در پردازش داده

به‌دلیل آن‌که در این کاربرد، در ابتدا فرآیند آشکارسازی خطا صورت می‌گیرد، سطح تحمل‌پذیری خطای ایمنی<sup>۲</sup> است، و از آنجایی‌که در ادامه فرآیند آشکارسازی خطا از یک کدگشا، برای تصحیح خطاها در رشته سندروم، استفاده می‌شود، پس این ساختار، یک سطح تحمل‌پذیری بازیابی<sup>۳</sup> را نیز فراهم کرده است. در نتیجه سطح تحمل‌پذیری خطا، در این مدل هم ایمنی و هم بازیابی است.

(۲) نشان داده شده، استفاده می‌کند. سامانه پردازش داده از طریق مقادیر توازن تعیین‌شده توسط یک کد کانولوشن حقیقی نظام‌مند حفاظت می‌شود. کدهای کانولوشن به جریان‌های پیوسته داده‌ها اجازه می‌دهند که برخلاف کدهای بلوکی حقیقی، بدون قطعه‌بندی<sup>۱</sup>، مورد حفاظت قرار بگیرند. در این ساختار از مقادیر توازن محاسبه‌شده از دو مسیر مجزا و قابل‌مقایسه استفاده می‌شود تا خطاهای ناشی از محاسبه‌گر را در عملیات‌های پیوسته پردازش داده‌های عددی، آشکارسازی کنند. جریان‌های مقادیر توازن از دو مسیر محاسبه می‌شوند؛ یک گروه از طریق داده‌های پردازش شده خروجی، و جریان قابل‌مقایسه دیگر توسط ترکیب عملیات پردازش داده با تابع مولد توازن فراهم می‌شود. مقایسه توازن‌ها، امکان آشکارسازی خطا را فراهم کرده، و در ادامه از تفاضل دو مقدار توازن به‌دست‌آمده از دو مسیر بالا و پایین، رشته سندروم تولید می‌شود؛ که با استفاده از اطلاعات این رشته سندروم‌ها، خطاها قابل آشکارسازی و تصحیح می‌شوند. اگر رشته سندروم دارای مقادیر کوچکتری از حد آستانه باشد، خطایی در سامانه رخ نداده است؛ ولی اگر رشته سندروم دارای مقادیر بزرگ‌تر از حد آستانه باشد، نشان‌دهنده آن است که در سامانه خطا رخ داده است. تصحیح خطا در صورتی که مقدار خطا از سطح آستانه بیشتر باشد، توسط یک کدبردار انجام می‌شود. در این مقاله، خطاها، هم در پردازش خطی و هم در زیرسامانه‌های تولید توازن، در نظر گرفته و منابع خطا، به‌صورت منفرد گذرای جمع‌شونده در خروجی، مدل‌سازی می‌شوند.



(شکل-۲): روندنمای ساختار تحمل‌پذیری خطای مبتنی بر الگوریتم عمومی با استفاده از ترکیب پردازش داده و مولد توازن

<sup>2</sup>Safety

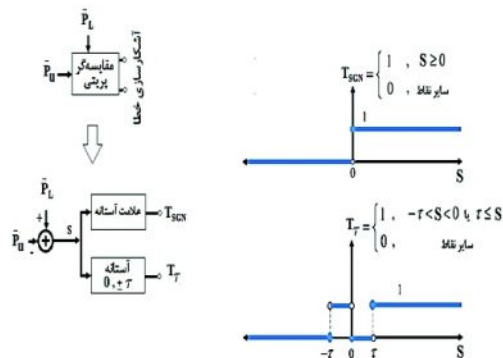
<sup>3</sup>Recovery

سال ۱۳۹۵ شماره ۲ پیاپی ۲۸

<sup>1</sup> Segmentation

### ۳-۱- آشکارسازی خطا

از تفاضل دو بردارِ توازن  $\bar{P}_L$  و  $\bar{P}_U$  در شکل ۲، که از دو مسیر مجزا محاسبه می‌شوند، بردار سنדרم به دست می‌آید. صرف نظر از خطای گرد کردن، دو مقدار توازن  $\bar{P}_L$  و  $\bar{P}_U$  می‌بایست در حالت بدون خطا، مساوی باشند. از این رو، آشکارسازی خطا، با مقایسه  $\bar{P}_L$  و  $\bar{P}_U$  در خروجی مقایسه‌گر حاصل می‌شود. این مقایسه‌گر، تفاضل بین دو مقدار توازن (سنדרم  $S$ )، را محاسبه می‌کند، اگر اندازه مقدار محاسبه شده از مقدار آستانه انتخاب شده کوچک‌تر باشد،  $(|S| < \tau)$ ، آن‌گاه خطایی رخ نداده است،  $(\tau)$  مقدار آستانه می‌باشد. هر سنדרم، بیان‌گر یک زوج قابل مقایسه از مقادیر توازن حقیقی است، که در یک زوج از مقادیر دودویی متمرکز می‌شود؛ که نشان‌دهنده یک کلمه کدی یک از دو<sup>۱</sup> است. شکل (۴)، نشان می‌دهد که چگونه دو مقدار توازن، با در نظر داشتن یک آستانه  $\tau$ ، به یک زوج کلمه کدی یک از دو، تبدیل می‌شود. زوج خروجی  $(T_{SGN}, T_r)$ ، یک کلمه کدی یک از دو معتبر خواهد بود،  $\{(1,0)$  یا  $(0,1)\}$ ، به شرط آن‌که اختلاف میان مقادیر توازن  $(S)$ ، در سطح آستانه  $\tau$  باشد. در صورتی که اختلاف مقادیر توازن، بزرگ‌تر از مقدار آستانه باشد، خروجی مقایسه‌گر، در شکل (۴)، به یک خروجی با کدی نامعتبر منجر خواهد شد، که در این حالت یک وضعیت آشکارسازی خطا، رخ می‌دهد.



(شکل - ۴): مقایسه‌گر مقادیر توازن

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

خطاهایی که در الگوریتم‌های پردازش داده رخ می‌دهند، ممکن است با به کارگیری کدهای کانولوشن نظام‌مند 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 حلقه محدود، بر مبنای کدهای تصحیح توده‌ای دودویی هستند. هدف، توصیف یک رویکرد حفاظتی است که به آسانی با تکنیک‌های پردازش داده ادغام شده و به تحمل‌پذیری مؤثرتر خطا می‌انجامد. 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 داده‌های پردازش شده، اعمال می‌کند. برخلاف کدهای بلوکی، کدهای کانولوشن، وقوع خطا را آشکارسازی می‌کنند. در این مقاله، قابلیت‌های تصحیح کدهای کانولوشن میدان محدود  $Z_2^m$  مورد بررسی قرار می‌گیرد. حساب نقطه ثابت، به‌طور 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<sup>۲</sup> اعداد صحیح  $\{0, 1, 2, \dots, (2^m - 1)\}$  است. با این همه، اعداد صحیح همچنین بسط‌های دودویی منحصر به فردی دارند، که منجر به  $m$  تایی دودویی مطابق با نشان‌گرهای اعداد صحیح می‌شوند.

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

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

### ۴-۱- ساختار کد کانولوشن و آزمون

کدهای کانولوشن، حتی آنهایی که بر روی دامن‌هایی از اعداد صحیح نقطه ثابت تعریف شده‌اند، توسط یک ماشین حالت محدود<sup>۳</sup> قابل توصیف هستند (کاستلو و همکاران، ۲۰۰۴). یک کد کانولوشن توصیف شده توسط فرآیند ماشین

### ۴- کدهای کانولوشن تعریف شده بر روی

#### اعداد صحیح

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

<sup>2</sup>Representatives

<sup>3</sup>Finite State Machine

<sup>1</sup>1-out-of-2

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 خطاها توسط 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 برده شده و با ماتریس  $H^v T$  نشان داده می‌شود. این نمایش،  $(n - k)$  معادله همگن را تعریف می‌کند:

$$0 = H^v T \bar{z}_t; \bar{z}_t = \begin{pmatrix} z_{t-v}^* \\ z_{t-v+1}^* \\ \vdots \\ z_{t-1}^* \\ z_t^* \end{pmatrix}, \quad (8)$$

$(v+1) \times n$

ماتریس آزمون توازن از  $(v + 1)$  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 نرخ  $k/n$  توسط  $\{d_{t,i}\}$  مشخص شده، و مکان‌های آزمون توسط  $\{c_{t,m}\}$  مشخص می‌شود.

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

برای کدهای نظام‌مند، زیرماتریس‌های ماتریس آزمون توازن  $\{H_i\}$  در رابطه (9) را می‌توان به شکل روابط (11) 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).

$$\{z_t^*\}_{t=0}^{+\infty} = \begin{pmatrix} z_{t-v}^* \\ z_{t-v+1}^* \\ \vdots \\ z_{t-1}^* \\ z_t^* \end{pmatrix} \quad (2)$$

که  $\{z_t^*\}_{t=0}^{+\infty}$  بردار حالت جاری به ابعاد  $(vk \times 1)$  بوده (با طول محدودیت  $v$ )، در جایی که  $\{\bar{u}_t\}_{t=0}^{+\infty}$  بردار ورودی با مرتبه  $k \times 1$  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.

$$z_{t+1}^* = \begin{pmatrix} z_{t-v}^* \\ z_{t-v+1}^* \\ \vdots \\ z_{t-1}^* \\ z_t^* \end{pmatrix} \quad (3)$$

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

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

که  $\bar{\Gamma}_t$  برابر است با:

$$\bar{\Gamma}_t = \begin{pmatrix} z_{t-v}^* \\ z_{t-v+1}^* \\ \vdots \\ z_{t-1}^* \\ z_t^* \end{pmatrix}; (v+1)k \times 1 \quad (5)$$

در جایی که ماتریس خروجی  $G$ ، با ابعاد  $n \times (v + 1)k$  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.

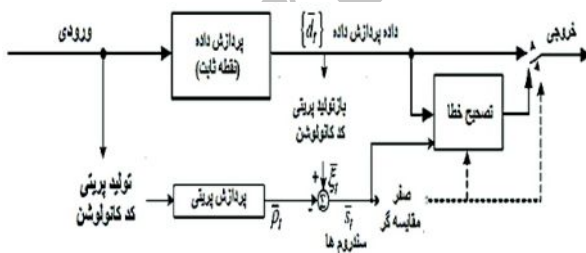
روابط (3) و (6)، یک کد کانولوشن با نرخ  $k/n$  با طول محدودیت  $v$  را توصیف می‌کنند. تأثیر خطا بر روی

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

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

$$\bar{s}_t = 0 + \sum_{i=0}^v p_i \bar{e}_{t-i}^{(d)} + \bar{e}_t^{(c)} \quad (b) \quad (16)$$

خطاها با توجه با قابلیت آشکارسازی کد، هنگامی که  $\bar{s}_t \neq 0$  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 تحمل‌پذیری خطا، یک‌سری از موقعیت‌های توازن مرتبط با موقعیت‌های داده‌های پردازش شده می‌باشند، که به‌طور 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 خطا در ساختارهای تحمل‌پذیر خطا است. این اصول آشکارسازی خطا، 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 تحمل‌پذیر خطا

داده‌های تولیدشده در زیرسامانه پردازش داده، در رشته بردار  $\{\bar{d}_{t-i}\}_{i=0}^v$  به ابعاد  $k \times 1$  نشان داده می‌شود؛ درحالی‌که 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

ماتریس واحد  $P_0^{(n-k) \times k, I_{(n-k)}}$

$$H_1^T = \begin{pmatrix} P_1 & 0 \\ \vdots & \vdots \\ P_{t-1} & 0 \end{pmatrix};$$

ماتریس صفر  $P_i^{(n-k) \times k, 0_{(n-k)}}$   $i=1,2,\dots,v$  (۱۱)

$v$  مکان در زیرماتریس صفر  $0_{(n-k)}$  و ماتریس واحد  $I_{(n-k)}$  در  $H^{VT}$  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:

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

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

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

$$\bar{E}_t = \begin{pmatrix} \bar{z}_{t-v} \\ \bar{z}_{t-v+1} \\ \vdots \\ \bar{z}_t \end{pmatrix}$$

$$\bar{z}_i = \bar{y}_i + \bar{e}_i;$$

$$i = (t-v), (t-v+1), \dots, t$$

ابعاد بردار  $\bar{z}_i$  با  $k \times 1$   $\bar{d}_i \bar{e}_i^{(d)}$

ابعاد بردار  $\bar{z}_i$  با  $(n-k) \times 1$   $\bar{c}_i \bar{e}_i^{(e)}$  (۱۴)

رابطه آزمون توازن ممکن است به‌دلیل خطا برای 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.

$$\bar{s}_t = H^{VT} \bar{E}_t;$$

با سندروم ابعاد  $(n-k) \times 1$  (۱۵)

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

## ۲-۴- کدهای کانولوشن بر روی حلقه حسابی نقطه ثابت

کدهای کانولوشن با نمادهایی از حلقه محدود  $Z_2^m$  بسط‌های دودویی به شکل برداری و ماتریسی دارند؛ برای مثال، بردار ورودی 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 خطادار،  $\bar{z}_t = \bar{y}_t + \bar{e}_t$ ،

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 توازن  $H^{vT}$  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  $Z_2^m$  به‌طورمستقیم با ماتریس دودویی آزمون توازن مرتبط است. این موضوع بدین معناست که قابلیت عملکردی یک کد کانولوشن دودویی می‌تواند تنها با استفاده از ماتریس‌های مولد و آزمون توازن دودویی، به یک کد کانولوشن حسابی، بر روی  $Z_2^m$  انتقال یابد. از این رو، همه کدهای کانولوشن دودویی می‌توانند در حلقه 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 حلقه مرتبط با  $Z_2^m$  دیده می‌شود، بنابراین تمایز  $+1$  از  $-1$  در این حلقه حائز اهمیت است. در حالت نظام‌مند، ماتریس مولد  $G$ ، با ماتریس آزمون توازن  $H^{vT}$  مرتبط است. زیرماتریس‌ها در ماتریس مولد  $G$  شامل زیرماتریس‌های توازن  $P_1$ ، می‌شوند:

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

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، به‌طورمستقیم از داده‌های پردازش شده و توسط زیرسامانه تولید مجدد توازن که در شکل ۵ نشان داده شده، تولید می‌شوند. این مقادیر توازن، با  $\bar{c}_t$  نشان داده می‌شوند.

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

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

$$\bar{\xi}_t = \sum_{r=0}^v P_r (\bar{d}_{t-r} + \bar{e}_{t-r}^{(d)}) + \bar{e}_t^{(c)} \\ \bar{\xi}_t (n-k) \times 1; \\ \bar{e}_t^{(c)} (n-k) \times 1. \quad (19)$$

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

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

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 موازی  $\bar{p}_t$  تأثیر بگذارد، آن‌گاه سندروم  $\bar{s}_t$  برابر با  $(\bar{e}_t^{(c)} - \bar{e}_t^{\prime(c)})$  می‌شود، که می‌تواند در یک مجموعه از خطاهای مدل‌سازی شده در درون سندروم  $\bar{s}_t$  نمایش داده شود.

$$\bar{s}_t = \sum_{i=0}^v P_i \bar{e}_{t-i}^{(d)} + (\bar{e}_t^{(c)} - \bar{e}_t^{\prime(c)}) \quad (21)$$

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

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

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

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.

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

در این طبقه از کدها  $(n-k)=1$  است. خواص طراحی این طبقه از کدها، تضمین می‌کند که هر خطا 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 خطا با 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 متناسط 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<sup>۱</sup> اشاره دارند که سبب تأثیرات غیرقابل پیش‌بینی خطاها در محاسبات داده‌ها می‌شوند. این کدها نیازمند باندهای محافظ<sup>۲</sup> (مناطق بدون خطا) در قبل و بعد از توده‌های خطاها هستند؛ به‌ویژه اگر به تصحیح خطا نیاز باشد. کدهای برلکمپ-پریاراتا-ماسی (BPM)<sup>۳</sup> یک طبقه بزرگ از کدهای تصحیح خطاهای توده‌ای با نرخ بالا هستند که دارای بسیاری از ویژگی‌های مطلوب در خصوص آشکارسازی خطا هستند. از مشخصات طراحی این کدها، صرف نظر از هر باند محافظی که به دنبال وقوع یک توده از خطاها به وجود می‌آید، آشکارسازی وقوع خطاهای ناشی از خطاها است. در کدهای BPM نظام‌مند،  $k=(n-1)$  بیت اطلاعات به هر زیربلوک به طول  $n$  تخصیص داده شده و کد دارای نرخ بالای  $(n-1)/n$  است. به‌ازای هر  $(n-1)$  ورودی یک بیت توازن، تولید می‌شود. طول محدودیت این کدها،  $v=(2n-1)$  است. ماتریس آزمون توازن  $B_0$ ، در واقع یک ماتریس دودویی  $2n \times n$  است که دارای یک ماتریس واحد مورب در  $n$  سطر بالای خود می‌باشد.

در رابطه (۲۵)،  $n$  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

<sup>1</sup>Possibly Contiguous

<sup>2</sup>Guard Bands

<sup>3</sup>Berlekamp-Preparata-Massey Codes



$$\bar{s}_{t+p} = \sum_{q=0}^{m-1} 2^q H^v T \bar{e}_{t-p}^{(q)}; p = 0, 1, \dots, \delta \quad (b)$$

$$\bar{e}_{t+p} = \begin{pmatrix} \bar{z}_{t-v+p} \\ \bar{z}_{t-v+p+1} \\ \vdots \\ \bar{z}_t \\ \vdots \\ \bar{z}_{t+p} \end{pmatrix};$$

بردار به ابعاد  $(v+1)n \times 1$   
 $p = 0, 1, \dots, \delta \quad (c)$

(۳۲)

در نخستین مرحله در فرآیند کدگشایی بازخورد سنדרوم،  $2^0$  عبارت از هر  $(\delta+1)$  بردار سنדרوم جدا خواهد شد و در مراحل کدگشایی مبتنی بر باینری مورد استفاده قرار خواهند گرفت. ضرایب  $2^0$  از هر بردار سنדרوم  $\bar{s}_{t+p}$  با یک کاهش به پیمانه 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 کدگشایی سنדרوم دودویی مرتبط با کد کانولوشن دودویی را می‌توان برای یافتن بردار خطای دودویی  $\bar{e}_t^{(0)}$  و بردار ضریب  $2^0$  خطاها به کار برد. مرحله بعدی در فرآیند کدگشایی بازخورد، مستلزم حذف تأثیر خطای دودویی  $\bar{e}_t^{(0)}$  است؛ با این وجود ضروری است که یک باند محافظ،  $\bar{z}_t$  را در جایی دنبال کند که خطاهای کمتری در سایر مقادیر دریافت‌شده  $\bar{z}_{t+1}, \bar{z}_{t+2}, \dots, \bar{z}_{t+\delta}$  رخ دهد.

$\bar{z}_t^{(0)}$  را با  $\bar{e}_t^{(0)} - \bar{z}_t^{(0)}$  در تمام اعضای بردار دریافت‌شده  $\bar{e}_{t+p}$  جایگزین کرده، و این بردارهای اصلاح‌شده با یک زیرنویس دوم 0 مشخص می‌شوند.

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

سنדרوم‌های اصلاح‌شده بدست آمده نیز نشانه‌گذاری اصلاح‌شده دارند.

$$\bar{s}_{t+p,0} = H^v T \bar{e}_{t+p,0}, \quad \text{به پیمانه } 2^m \quad (35)$$

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

از طرف دیگر، اگر توده خطای غیر صفری در  $\bar{e}_{t+1}, \bar{e}_{t+2}, \dots, \bar{e}_{t+v}$  وجود داشته باشد، آن گاه بردار سنדרوم  $\bar{\sigma}_t$ ، یک بردار تمام صفر نیست؛ حتی اگر توده خطا قابلیت تصحیح کد را از بین ببرد، وقوع خطا 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، امکان آزمودن این که آیا بردار سنדרوم  $\bar{\sigma}_t$  نشان‌دهنده خطاهای قابل تصحیح است یا خیر، وجود دارد.

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

#### ۴-۴- کدگشایی بازخورد تکراری با سنדרوم‌ها

کدگشایی سنדרوم از بردارهای سنדרوم دریافت‌شده استفاده می‌کند. گستره‌ای از سنדרوم‌ها برای کدگشایی نمادهای موجود در بردار دریافت‌شده  $\bar{z}_t$ ، به کار گرفته می‌شود. این گستره، نشان‌دهنده یک تأخیر کدگشایی است که با  $\delta$  نشان داده می‌شود. به طور مثال کدهای کانولوشن تصحیح خطای توده‌ای برلکمپ-پرپاراتا-ماسی، با  $v = \delta$ ، کدگشایی می‌شوند. بردارهای سنדרوم  $\bar{s}_{t+p}$  در برآورد بردار خطای  $\bar{e}_t$  استفاده می‌شوند.

$$\left( \begin{matrix} \bar{s}_t \\ \bar{s}_{t+1} \\ \vdots \\ \bar{s}_{t+\delta} \end{matrix} \right) \rightarrow \text{کدگشایی} \rightarrow \bar{e}_t. \quad (31)$$

بردار سنדרوم کلی  $\bar{s}_t$  و نیز بردار داده دریافت‌شده  $\bar{e}_t$ ، در رابطه (۱۴)، بسط داده شده و به یک بردار دودویی منجر می‌شود.

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

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

زمانی که خطاها به درستی برآورد می‌شوند، بردار سندروم جدید  $\bar{s}_{t+p,1}$ ، هیچ ضربی از  $2^0$  یا  $2^1$  را در بسط‌های دودویی خود ندارد.

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

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

$$\bar{s}_{t+p,1} \cdot 2^2 = \left( \left( \bar{s}_{t+p,1} \right)_{2^3} \right) = \left( \left( H^{VT} \bar{\Xi}_{t+p,1} \right)_{2^3} \right) \cdot 2^2 \quad (42)$$

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.

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

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

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

$$\begin{aligned} \bar{s}_{t+p,2} &= H^{VT} \bar{\Xi}_{t+p,2}, \quad \text{به پیمانه } 2^m & (a) \\ \bar{s}_{t+p,2} &= \bar{s}_{t+p,1} - H_p^T e_t^{(2)} \cdot 2^2, \quad \text{به پیمانه } 2^m & (b) \\ \bar{s}_{t+p,2} &= \sum_{q=3}^{m-1} 2^q \bar{s}_{t+p,2}^{(q)} \quad p=0,1,2,\dots,\delta & (c). \end{aligned} \quad (45)$$

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  $\bar{e}_t^{(0)}$  به درستی برآورد شده باشد، آن‌گاه محاسبه سندروم‌های اصلاح‌شده از راه دیگری نیز امکان‌پذیر است؛ یعنی توسط حذف تأثیرات  $\bar{e}_t^{(0)}$  به‌طور مستقیم با به‌کارگیری زیرماتریس‌های صحیح  $\{H_i\}$ ،  $H^{VT}$ .

$$\begin{aligned} \bar{s}_{t+p,0} &= \bar{s}_{t+p} - H_p^T \bar{e}_t^{(0)} 2^0, \\ 2^m &\text{ به پیمانه } \delta; \quad p = 0,1,\dots,\delta. \end{aligned} \quad (36)$$

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

$$\begin{aligned} \bar{s}_{t+p,0} &= \sum_{q=0}^{m-1} 2^q \bar{s}_{t+p,0}^{(q)}; \\ p &= 0,1,\dots,\delta. \end{aligned} \quad (37)$$

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

$$\begin{aligned} \bar{s}_{t+p,0}^{(1)} \cdot 2^1 &= \left( \left( \bar{s}_{t+p,0} \right)_{2^2} \right) = \\ & \left( \left( H^{VT} \bar{\Xi}_{t+p,0} \right)_{2^2} \right) \cdot 2^1; \quad p = 0,1,\dots,\delta. \end{aligned} \quad (38)$$

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 دریافت‌شده  $\bar{\Xi}_{t+p,0}$  با جایگزین کردن  $\bar{z}_{t,0}$  با  $(\bar{z}_{t,0} - 2^1 e_t^{(1)})$ ،  $\bar{z}_{t,1}$  ساده‌تر می‌شود:

$$\begin{aligned} \bar{\Xi}_{t+p,0} \bar{z}_{t,1} &= \bar{z}_{t,0} - 2^1 e_t^{(1)} \\ &\text{به پیمانه } 2^m. \end{aligned} \quad (39)$$

یک مجموعه اصلاح‌شده 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 ضرب 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 ضرایب‌هایی هستند که از توان  $2^{(r+1)}$  آغاز می‌شوند.

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

این بردارهای سندروم، معادل با بردارهای سندروم اصلاح‌شده در مرحله قبل  $\bar{s}_{t+p,(r-1)}$  هستند.

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

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

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

$$\bar{s}_{t+p,r}^{(r+1)} \cdot 2^{(r+1)} = \left( (\bar{s}_{t+p,r}) \right)_{2^{r+2}} = \left( (H^{VT} \bar{e}_{t+p,r}) \right)_2 \cdot 2^{(r+1)} \quad (b) \quad (52)$$

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 خطاها در بردارهای دریافت‌شده و اصلاح‌شده  $\bar{e}_{t+p,(m-1)}$  حذف شده و بردار  $\bar{z}_{t,(m-1)}$ ، که یک بردار  $1 \times n$  است، نشان‌دهنده داده‌های کدگذاری‌شده اصلی  $\bar{z}_t$  هستند.

## ۵- پیاده‌سازی و اجرا

به منظور بررسی درستی تکنیک کدگشایی تکراری، برخی آزمایش‌ها صورت گرفته و یک طرح کلی از این روش در ادامه ارائه شده است. مدل‌سازی خطا، مجموعه انعطاف‌پذیری از شرایط و تأثیرات خطا را 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  $3 \times 3$  در

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

$$\bar{s}_{t+p,(r-1)} = H^{VT} \bar{e}_{t+p,(r-1)}, \quad \text{به پیمانه } 2^m \quad (a)$$

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

$$\bar{s}_{t+p,(r-1)} = \sum_{q=r}^{m-1} 2^q \bar{s}_{t+p,(r-1)}^{(q)} \quad p=0,1,2,\dots,\delta \quad (c) \quad (46)$$

فرض بر این است که بردارهای مؤلفه‌های خطای  $\bar{e}_t^{(0)}$ ،  $\bar{e}_t^{(1)}$ ، ...،  $\bar{e}_t^{(r-1)}$  به درستی پرآورد 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.

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

کاهش به پیمانه  $2^{(r+1)}$  در بردارهای سندروم،  $r$  آمین مؤلفه آزمون توازن را جدا می‌کند. از این رو، مؤلفه‌های  $\bar{s}_{t+p,(r-1)}^{(r)}$  از بردار سندروم، بردارهای دودویی مجاز را برای استفاده در کدگشایی دودویی فراهم کرده و بردار خطای دودویی  $\bar{e}_t^{(r)}$  تعیین می‌شود؛ سپس تأثیر این خطای داده، در  $t$  آمین گروه از داده‌های دریافت‌شده حذف می‌شود.

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

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

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

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

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

ماتریس آزمون توازن  $H^{VT}$  که مقدار توازن را تعیین می‌کند، ابعاد  $1 \times (4 \cdot 8)$  است. یک سطر ماتریس آزمون توازن عبارت است از:

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

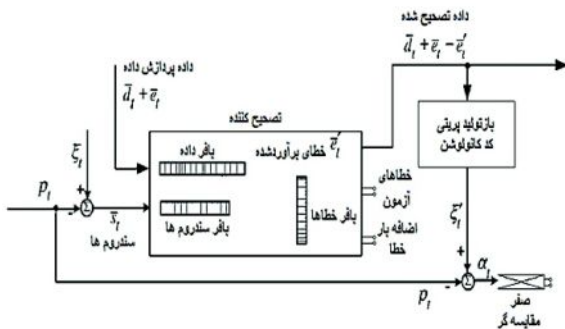
کد شبیه‌سازی، به صورت تصادفی، یک توده خطا را در هر بلوک از  $n$  نماد  $Z_2^m$  درج می‌کند. انتخاب این توده با پارامتر احتمال  $\varepsilon$  کنترل می‌شود. هنگامی که یک خطای توده‌ای رخ می‌دهد،  $n$  نماد از  $Z_2^m$  با استفاده از یک توزیع یکنواخت تولید شده و سپس به  $n$  نماد افزوده شده تا یک خطای توده‌ای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

## ۶- عملیات تصحیح داده

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

تصحیح‌کننده، خطاها را در سندروم ( $S_t$ ) تشخیص داده و به‌درستی آن‌ها را برآورد می‌کند. از این‌رو، زوج خروجی خطای آزمون، نشان می‌دهد که چه مشکلاتی در کل سامانه وجود دارد؛ هر چند این که داده‌ها صحیح باشند.

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



(شکل-۶): حفاظت تصحیح‌کننده

$$\bar{c}'_t = \sum_{r=0}^v p_r (\bar{d}_{t-r} - \bar{e}'_{t-r}) \quad (54)$$

هم زمان، اگر خطاها 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 خطا، غیر صفر خواهند شد.

$$\bar{\alpha}_t = \bar{c}'_t - \bar{p}_t = - \sum_{r=0}^v p_r \bar{e}'_{t-r} \quad (56)$$

همان‌گونه که بیان شد، از کدهای کنولوشن به‌طور مؤثری برای آشکارسازی خطاها 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.

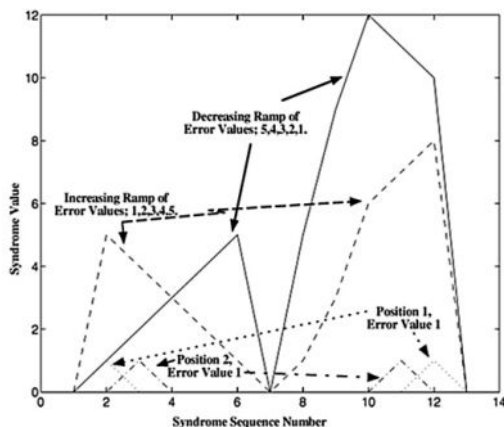
## ۸- شبیه‌سازی‌ها

### ۸-۱- برآوردها و محاسبات

از تفاضل دو بردار توازن  $\bar{P}_L$  و  $\bar{P}_H$  در شکل ۲، که از دو مسیر مجزا محاسبه می‌شوند، بردار سندرم به دست می‌آید. با صرف نظر از خطای گرد کردن، دو مقدار توازن  $\bar{P}_L$  و  $\bar{P}_H$  می‌بایست در حالت بدون خطا، مساوی باشند. از این رو، آشکارسازی خطا، با مقایسه  $\bar{P}_L$  و  $\bar{P}_H$  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  $S = \bar{P}_L - \bar{P}_H$ )، آن‌گاه خطایی رخ نداده است؛ ( $\tau$  مقدار آستانه می‌باشد). ساختار کد کانولوشن حقیقی، به گونه‌ای طراحی شده که به ازای الگوهایی از خطاهای مختلف که در خروجی بلوک پردازش، ظاهر می‌شوند، سندرم‌های قابل تفکیک و متمایزی را تولید کند (ردینبو، ۱۹۹۸). تفکیک بردارهای سندرم، با وزن همینگ<sup>۱</sup> 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 (۵،۶،۱۱) 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 کوچکتری از مقدار آستانه باشد، در آن صورت مدل‌های خطا، در شکل (۲) دارای واریانس  $\xi$  برای خطای  $A$  و واریانس  $\phi$  برای خطای  $B$  هستند، (مقدار آستانه، پنج برابر انحراف معیار مربوط به خطای گرد کردن در حالت بدون خطا در نظر گرفته می‌شود). حال اگر مقدار سندروم از مقدار آستانه تجاوز کند، در واقع خطا رخ داده است؛ که در این حالت واریانس خطاهای  $A$  و  $B$  به ترتیب با  $\theta$  و  $\xi$  نشان داده می‌شوند. که اندازه واریانس‌های  $\theta$  و  $\xi$  چندبرابر اندازه واریانس‌های  $\xi$  و  $\phi$  است، (در تمام حالت‌ها

خطا به شکل جمع‌شونده در خروجی پردازش و زیرسامانه‌های 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 متوالی (۱۱، ۵، ۶)، تولید شده‌اند، این کد، به ازای هر پنج نمونه داده ورودی، یک نمونه توازن، تولید می‌کند. چهار پاسخ سندروم متمایز، در شکل (۷)، از خطاهای واحد در موقعیت منفرد، یا شیب‌های خطا در پنج موقعیت داده، حاصل می‌شوند. یکی از این رشته سندروم‌ها، از یک شیب خطای افزایشی، با مقادیر خطای ۱، ۲، ۳، ۴، ۵ ایجاد می‌شود؛ در حالی که رشته سندروم دیگر با یک شیب خطای کاهششی، با مقادیر خطای ۵، ۴، ۳، ۲ و 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 خطا در موقعیت دو به دست می‌آید.



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

### ۸-۲- انتخاب مقدار آستانه و عملکرد

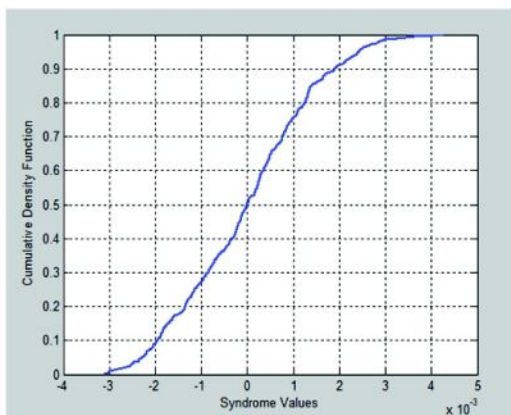
#### آشکارسازی خطا

نتایج شبیه‌سازی‌ها نشان می‌دهند که اختلاف میان خطای گرد کردن و خطای ناشی از خطای داخلی، به اندازه کافی برای متمایز شدن از یکدیگر بزرگ است. این اختلاف بزرگ میان این دو خطا، محدوده و گستره‌ای از آستانه‌های آشکارسازی خطا را، برای انتخاب، به ما می‌دهد. کوچک گرفتن این حد آستانه باعث می‌شود که خطای ناشی

<sup>2</sup> Burst

<sup>1</sup> Hamming Weight

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



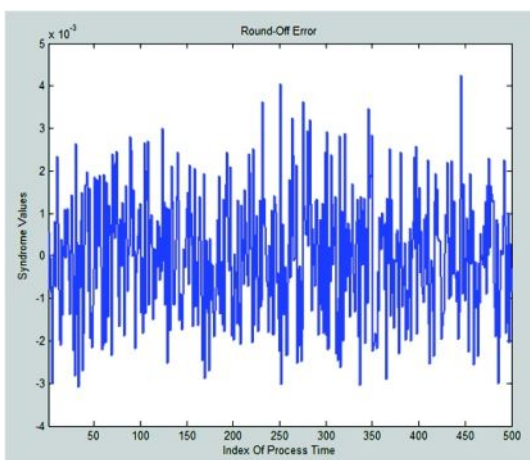
(شکل - ۹): نمودار تابع توزیع تجمعی خطا

با توجه به شکل (۹)، مشاهده می‌شود که احتمال آن که مقادیر سندروم در حالت بدون اعمال خطا، کمتر از  $3 \times 10^{-3}$  باشند، حدود 0.01 است و احتمال آنکه مقادیر سندروم بیشتر از  $3 \times 10^{-3}$  باشند، حدود 0.014 است. همچنین احتمال آنکه خطاها در محدوده  $\pm 3 \times 10^{-3}$  باشند، برابر با 0.976 است. میانگین مربعات خطا و بیشینه میانگین مربعات خطای به دست آمده از پانصد بار تکرار، در شکل‌های (۱۰) و (۱۱)، مورد ارزیابی قرار گرفته‌اند. برای شبیه‌سازی و به دست آوردن این نمودارها، ده هزار نمونه داده در هر پانصد بار تکرار، تولید شده و خطای گرد کردن هر یک از نمونه‌ها 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 خطا در برابر سطوح مختلف خطا، در شکل (۱۲) نشان داده می‌شود. در این شبیه‌سازی ابتدا یک سطح خطا مشخص شده و در هزار تکرار که در هر تکرار هزار نمونه تولید می‌شود، خطا در نمونه‌های مختلف، به طور تصادفی و مستقل از نمونه‌های قبلی تولید و سپس متوسط مربعات خطاها در تمام تکرارها محاسبه شده و این کار برای سطوح خطای بعدی نیز تکرار می‌شود (مقدار آستانه  $5 \times 10^{-3}$  در نظر گرفته شده است). عملکرد آشکارسازی خطا در برابر سطح آستانه، زمانی که خطا در سامانه خطی پردازش داده، رخ می‌دهد، 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.

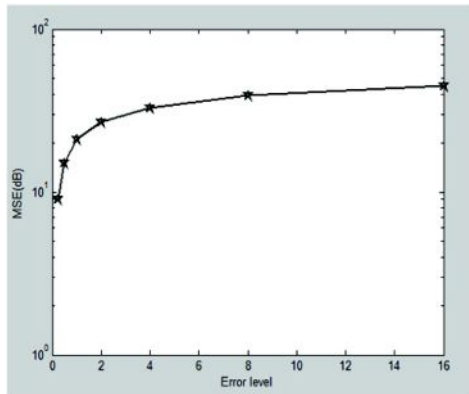
با استفاده از شبیه‌سازی، چگونگی انتخاب مقدار آستانه را برای آشکارسازی خطاها، بررسی می‌کنیم. برای ارزیابی قابلیت آشکارسازی خطا، در ساختار پیشنهادی، به عنوان مثال، از کد کانولوشن حقیقی نظام مند (۲، ۲، ۳)، با چند جمله‌ای‌های مولد  $g_1^{(3)}(D)=1+D$  و  $g_2^{(3)}(D)=1+D^2$  استفاده شده است. برای انجام این شبیه‌سازی، 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 می‌شود. همچنین، خطا توسط یک توزیع گوسی با میانگین صفر و واریانس ثابت، تولید می‌شود.

شکل (۸) نتیجه شبیه‌سازی را بعد از پانصد بار تکرار نشان می‌دهد؛ در این شکل، رشته سندروم در حالت بدون خطا نشان داده شده و مقادیر غیر صفر، مربوط به خطای گرد کردن می‌باشند. مقدار سندروم‌ها در بیشتر موارد کمتر از  $3 \times 10^{-3}$  است؛ اما اندازه تعداد کمی از خطاها بزرگ‌تر از  $3.5 \times 10^{-3}$  است. بنابراین مقدار آستانه  $17.5 \times 10^{-3} = 5 \times (3.5 \times 10^{-3})$ ، به منظور آشکارسازی خطا انتخاب می‌شود؛ یعنی پنج برابر مقدار حداکثر سندروم در حالت بدون تزریق خطا، به عنوان مقدار آستانه انتخاب شده است. برای آستانه‌های کوچک، خطاهای بزرگ، به راحتی آشکارسازی می‌شوند؛ در حالی که آستانه‌های بزرگ، عملکرد آشکارسازی خطا را کاهش می‌دهند.

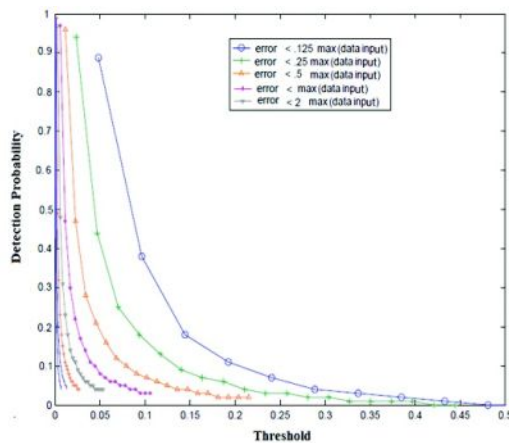


(شکل - ۸): رشته سندروم برای حالت بدون خطا

<sup>۱</sup> Cumulative Density Function



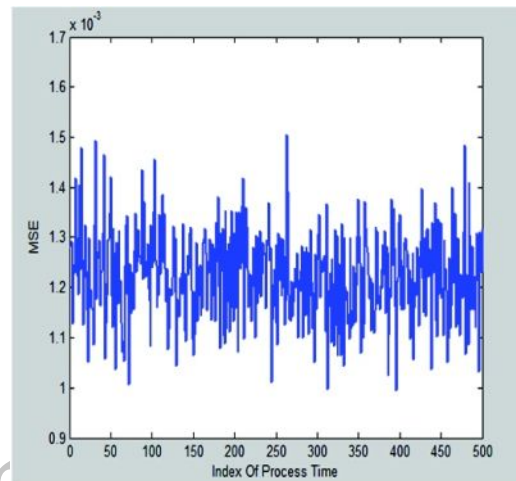
(شکل - ۱۲): میانگین مربعات خطا در مقابل سطح تزریق خطا



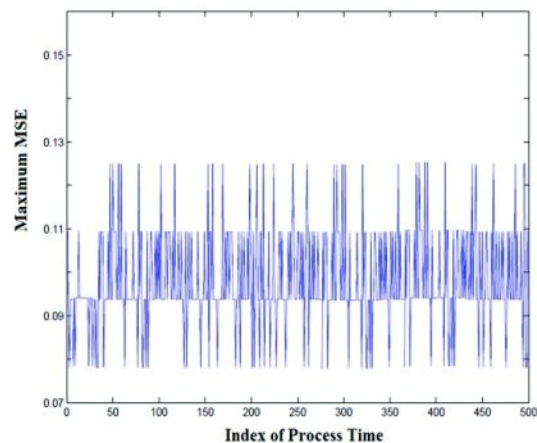
(شکل - ۱۳): عملکرد آشکار سازی خطا در مقابل آستانه

هر چه سطح تزریق خطا بیشتر باشد، درصد و احتمال آشکار سازی خطا بیشتر است؛ که این موضوع در شکل (۱۴) نیز قابل مشاهده است. درحالتی که سطح تزریق خطا از شانزده برابر حداکثر مقدار داده ورودی تجاوز نمی‌کند؛ به عبارت دیگر سطح تزریق خطا، کوچک‌تر از شانزده برابر حداکثر مقدار داده ورودی می‌باشد شکل (۱۴) - الف)، درصد آشکار سازی خطا نزدیک به صد درصد (۰/۹۹۹۹۷۰۰، جدول (۱)) است؛ ولی زمانی که سطح تزریق خطا از یک دوم حداکثر مقدار داده ورودی تجاوز نمی‌کند به کم‌ترین مقدار درصد آشکار سازی خطا می‌رسد شکل (۱۴). عملکرد آشکار سازی خطای سامانه پردازش داده، در زمانی که سطح تزریق خطا از ۱/۱۶، ۱/۸، ۱/۴، ۱/۲، ۱/۵، برابر حداکثر مقدار داده ورودی تجاوز نمی‌کند، در شکل (۱۴) نشان داده شده است. سطح حداکثر خطا برای هر تکرار، تنها یک‌بار تغییر می‌کند؛ همانطور که در شکل (۱۳) نیز مشاهده می‌شود، سطوح خطا ۱/۱۶، ۱/۸، ۱/۴، ۱/۲، ۱/۵، برابر حداکثر 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 آشکار سازی، کاهش می‌یابد. به‌ازای یک مقدار معین آستانه، هر چه دامنه خطا بیشتر باشد، احتمال عدم آشکار سازی خطا، کمتر خواهد بود. به عبارت دیگر در چنین شرایطی احتمال آشکار سازی صحیح خطا، بیشتر می‌شود.



(شکل - ۱۰): میانگین مربعات خطا



(شکل - ۱۱): ماکزیمم میانگین مربعات خطا

همچنین هر چه مقدار آستانه بیشتر باشد، در صورتی که سطح خطا ثابت بماند، احتمال آشکار سازی کمتر خواهد بود. این نمودارها به‌ازای پانصد هزار نمونه داده، استخراج شده‌اند (به شکل پانصد تکرار که در هر تکرار هزار داده با توزیع گوسی تولید شده است). سطوح خطا ۱/۱۶، ۱/۸، ۱/۴، ۱/۲، ۱/۵، ۰/۲۵، ۰/۱۲۵، برابر حداکثر مقدار داده ورودی، اعمال شده، و مقادیر آستانه از  $10^{-5}$  تا ۱، تغییر می‌کنند.

در نظر گرفته شده است. هنگامی که حداکثر سطح خطا بزرگ باشد، عملکرد آشکارسازی خطا به سرعت در مقابل افزایش آستانه آشکارسازی، کاهش می‌یابد، که در شکل (۱۴) نشان داده شده است. آخرین نمودار در شکل (۱۴)، منحنی عملکرد آشکارسازی خطا را برای سطح خطایی که کوچکتر از یک‌هشتم حداکثر مقدار داده ورودی است، نشان می‌دهد، میانگین مربعات خطا در این حالت  $0.98376$  یا  $0.71076$  - می‌باشد. مقدار آستانه می‌تواند بر اساس شرایط خطا و منحنی عملکرد آشکارسازی خطا، انتخاب شود.

جدول (۱). اگر مقدار آستانه  $T$ ، بسیار کوچک تعیین شود، آن‌گاه حتی خطای گردکردن نیز به عنوان خطای 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 خطا را تحت تأثیر قرار دهد. خطاها توسط آزمون اندازه‌سنده‌ها و مقایسه آنها در مقابل آستانه‌هایی که مقدارشان پنج برابر اندازه انحراف معیار مقدار سنده‌ها مربوط به خطاهای گردکردن هستند، آشکارسازی می‌شوند. برنامه شبیه‌سازی به صورت تصادفی، مؤلفه خطای جمع‌شونده را تولید می‌کند، که اندازه و بزرگی هر خطا از یک توزیع گوسی با میانگین صفر و واریانس ثابت، انتخاب می‌شود.

### ۸-۳- مقایسه

برای مقادیر آستانه کوچک، تعداد خطاهای زیادی آشکارسازی می‌شوند؛ در حالی که برای مقادیر آستانه بزرگ بازدهی آشکارسازی خطا کاهش می‌یابد. تغییر مقدار آستانه از یک مقدار کوچک به یک مقدار بزرگ سبب گذار از یک سطح بالای آشکارسازی خطا، به یک سطح پایین از آشکارسازی خطا خواهد بود. به هر حال، برای شبیه‌سازی، توانایی آشکارسازی خطا، تأثیر مستقیمی با واریانس خطای تریق شده دارد. همان‌گونه که بیان شد، با مقایسه تفاضل دو مقدار توازن  $\bar{P}_I$  و  $\bar{P}_H$  (شکل (۲))، می‌توان پاسخ سامانه به خطا را بررسی کرد. شکل (۱۵) چگونگی تأثیر خطاها بر روی خروجی مقایسه‌گر را نشان می‌دهد. شکل ۱۵-الف یک تفاضل بسیار کوچک میان دو مقدار توازن  $\bar{P}_I$  و  $\bar{P}_H$  را نشان می‌دهد. دلیل اختلاف غیر صفر، خطاهای گردکردن ناشی از پاسخ‌های محدود سامانه محاسباتی است. در شکل (۱۵-ب)، مقادیر  $|\bar{P}_I - \bar{P}_H|$  تأثیر خطاهای رخ داده شده را

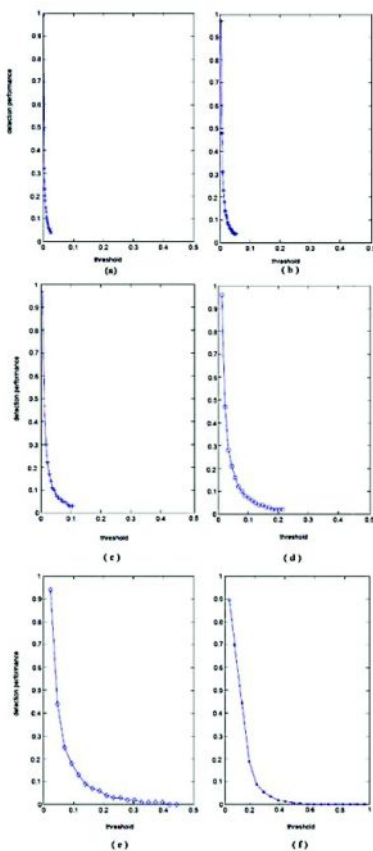
منعکس می‌کنند. اگر آستانه خطا به اندازه کافی و مناسب کوچک در نظر گرفته شود، در این صورت اکثر خطاها توسط مقایسه‌گر آشکارسازی می‌شوند؛ اما اگر، آستانه خطا بسیار کوچک انتخاب شود، مقایسه‌گر، خطاهای گردکردن را نیز به عنوان خطاهای محاسباتی در نظر می‌گیرد. بنابراین ما به پیداکردن یک مقدار آستانه مناسب نیاز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

شکل (۱۶)، عملکرد آشکارسازی خطا را در برابر مقادیر آستانه نشان می‌دهد. در مقادیر کوچک آستانه، مقایسه‌گر بیشتر خطاها 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 خطا در سطوح مختلف به طور تصادفی طبق جدول (۲) به داده‌ها اضافه می‌شود و در پایان تعداد نمونه‌هایی که خطا در آنها وجود داشته، ولی تشخیص داده نشده است، شمارش شده و احتمال عدم آشکارسازی خطا، از تقسیم این تعداد بر کل داده‌هایی که خطا در آنها بوده بدست می‌آید.

در کار انجام شده توسط Redinbo، در سال ۲۰۱۱، یک ساختار بهینه برای آشکارسازی و تصحیح خطاهای ناشی از خرابی‌های سخت‌افزاری متناسب در عملیات‌های پردازش خطی، توسعه یافته است. در این روش، یک فیلتر پیش‌گویی کالمن، اصلاح خطا را از روی رشته سنده‌ها، برعهده دارد. اگر چه روش پیشنهاد شده بر اساس تخمین‌گر کالمن می‌تواند با دقت بالا، خطا در بلوک پردازش خطی را تشخیص و اصلاح کند؛ اما در صورت بروز خطا در سایر نواحی غیر از بلوک پردازش خطی، تنها قادر است آن را 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 خطا در این نواحی چندان کمتر از بلوک پردازش اصلی نیست و نمی‌توان آنها را مصون از خطا فرض کرد. رشته سنده‌ها به علت وجود خطای گردکردن اعداد، حتی در صورت عدم وجود خطا، مخالف صفر است. این رشته در این حالت به عنوان یک نوفه سفید با میانگین و واریانس مشخص تلقی می‌شود.



تکنیک‌های پردازش داده ادغام شده و به تحمل‌پذیری مؤثرتر خطا می‌انجامد. سامانه پردازش داده از طریق مقادیر توازن تعیین شده با کدهای کانولوشن تعریف شده بر روی اعداد صحیح محافظت می‌شود. مقایسه توازن‌های محاسبه شده، از دو مسیر آشکارسازی خطا را فراهم می‌آورد، درحالی‌که تصحیح داده‌های خروجی توسط یک کدگشای (کدبردار) آستانه که تحت تأثیر خطاهای گردکردن و خطاهای ناشی از محاسبه گر است، صورت می‌گیرد. در این مقاله رویکردی برای به‌کارگیری کدهای کانولوشن بکار برده شده در فرم‌های نظام‌مند ارائه می‌دهیم، نتایج شبیه‌سازی، نشان می‌دهند که اختلاف میان خطای گردکردن و خطای ناشی از محاسبه گر به اندازه کافی برای متمایز شدن از یکدیگر بزرگ است. این اختلاف بزرگ، میان این دو خطا، محدوده و گستره‌ای از آستانه‌های آشکارسازی خطا را برای انتخاب به ما می‌دهد. در ادامه، احتمال آشکارسازی و عدم آشکارسازی خطا برای برخی از کدهای کانولوشن نشان داده شده است. عملکرد آشکارسازی در یک سامانه پردازش اطلاعات به آستانه آشکارسازی بستگی دارد که با خطای گردکردن تعیین می‌شود.



(شکل - ۱۴): عملکرد آشکارسازی خطا: (a) در زمانی که سطح تزریق خطا از ۱۶ برابر حداکثر مقدار داده ورودی تجاوز نمی‌کند، (b) در زمانی که سطح تزریق خطا از هشت برابر حداکثر مقدار

(جدول - ۱): عملکرد آشکارسازی خطا و میانگین مربعات خطا

| میانگین مربعات خطا (dB) | میانگین مربعات خطا          | احتمال آشکارسازی خطا | سطح خطا <sup>۱</sup> |
|-------------------------|-----------------------------|----------------------|----------------------|
| 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          | $1.2739657339 \times 10^2$  | 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 نویز در کار آقای Redinbo،  $10^{-1}$  در نظر گرفته شده است. هنگامی که یک خطا در سامانه رخ می‌دهد، رشته سندروم مقادیر بزرگی را نشان خواهد داد. فرض شده که مقدار واریانس در این حالت برابر هزار است؛ بنابراین بانداژه کافی از خطای گردکردن متمایز فرض می‌شود. نتایج ارائه شده در کار آقای Redinbo، نشان می‌دهد که قدر مطلق خطای تصحیح، حدود  $10^{-5}$  است؛ درحالی‌که در ساختار ارائه شده در این مقاله، این خطا در حدود  $10^{-4}$  است. علاوه بر این، از آنجایی که خود فیلتر کالمن و مسیر ایجاد توازن، مصون از خطا نیستند، برای حمایت از آنها لازم است از سخت‌افزارهای جانبی بهره گرفت؛ اما، در ساختار ارائه شده، مدار کدگشا و اصلاح خطا، با توجه به اینکه از عناصر بسیار ساده استفاده می‌کنند، مصون از خطا هستند. پس نیازی به سخت‌افزار جانبی وجود ندارد.

## ۹- نتیجه گیری و پیشنهادها

در این مقاله، ساختار ارائه شده به حفاظت از هر شکل کلی از عملیات پردازش خطی می‌پردازد. برای اهداف تصحیح، از افزونگی استفاده می‌شود که به این منظور، کدهای کانولوشن به کار گرفته می‌شوند. کدهای کانولوشن تعریف شده بر روی اعداد صحیح، یک ساختار ریاضی برای محاسبات متمایز ثابت هستند، این کدها، کدهای کانولوشن دودویی شناخته شده‌ای را به عنوان مولدهای اصلی‌شان به کار می‌برند. این طبقه از کدهای کانولوشن حسابی از نمادهای برگرفته از حلقه محدود، بر مبنای کدهای تصحیح توده‌ای دودویی هستند. هدف، توصیف یک رویکرد حفاظتی است که به آسانی با

<sup>۱</sup> سطح خطا نسبت به حداکثر 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 ( $\tau$ ) |
|---------|---|-------------------------------------|-------------------------|
| (3,2,2) | توزیع گوسی با میانگین 0 و انحراف معیار 2 (به ازای هر 5 نمونه یک خطا اضافه می‌شود) | $10^3$ (با تعداد تکرار 1000)        | 0 - 16 (با گام 0.05)    |

تعمیم ساختار ارائه شده در این مقاله با استفاده از کدهای موجک، جهت کاربردهای فشرده‌سازی داده و پردازش تصویر پیشنهاد می‌شود. در این راستا مهم‌ترین مسأله، یافتن رویکرد تحلیلی مناسب و ساده، برای اندازه‌گیری دقیق عملکرد آشکارسازی خطا در کاربردهای تحمل‌پذیری 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

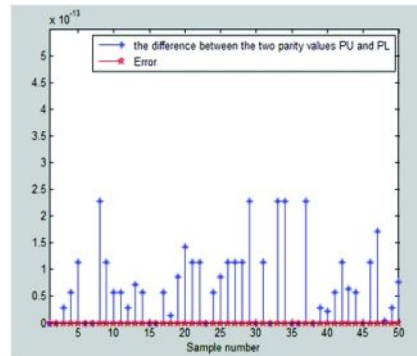
Banerjee, P., Rahmeh, P. J. T., Stunkel, C. B., Nair, V. S. S., Roy, K., & Abraham, J. A. 1990. "Algorithm-based fault tolerance on a hypercube multiprocessor." IEEE Transactions on Computers, 39(9), 1132-1145. doi:10.1109/12.57055

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

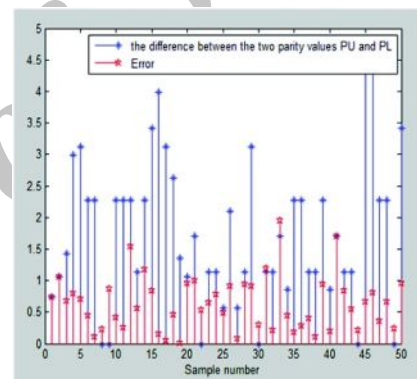
Berlekamp ER, 1962, "class of convolution codes." Inf Control. 6, 1-13.

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 حداکثر مقدار داده ورودی تجاوز نمی‌کند، (d) در زمانی که سطح تزریق خطا از دو برابر حداکثر مقدار داده ورودی تجاوز نمی‌کند، (e) در زمانی که سطح تزریق خطا، برابر حداکثر مقدار داده ورودی باشد، (f) در زمانی که سطح تزریق خطا از ۰/۵ برابر حداکثر مقدار داده ورودی، تجاوز نمی‌کند.



(a)

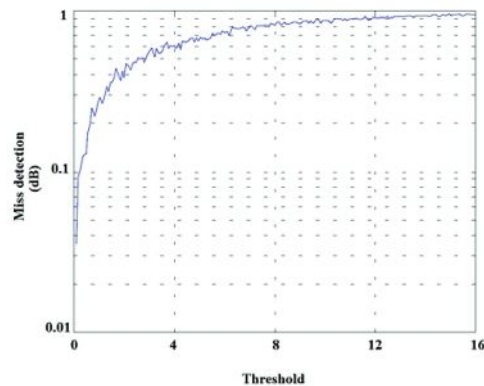


(b)

(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. Nematı (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/PRO-C.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.



**حجت‌اله 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 رایانامه ایشان عبارت است از:

**h\_hamidi@kntu.ac.ir**