شناسایی کور کدهای توربو ضربی مبتنی بر تصمیم‌گیری نرم

حمیدرضا خدادادی[۱]، استادیار؛ عباس قائدی[۲]، کارشناس ارشد؛ محمد حسین حسن[۳]، کارشناس ارشد

۱- دانشکده فناوری اطلاعات و ارتباطات - دانشگاه جامع امام حسین (ع) - تهران – ایران hkhdadi@ihu.ac.ir
۲- دانشکده فناوری اطلاعات و ارتباطات - دانشگاه جامع امام حسین (ع) - تهران – ایران aghaedi@ihu.ac.ir
۳- دانشکده فناوری اطلاعات و ارتباطات - دانشگاه جامع امام حسین (ع) - تهران - ایران mh.hassan@ihu.ac.ir

چکیده: در این مقاله یک روش جهت شناسایی کور پارامترهای کدگذاری برای کدهای توربو ضربی ارائه می‌شود. گیرنده شنود کننده فقط می‌داند که رشته بیت دریافتی با کدگذار باینری ضربی کد شده، درحالی‌که پارامترهای کدگذاری مجهول هستند. فرآیند شناسایی شامل دو قسمت اصلی است: شناسایی طول کد و بازسازی چندجمله‌ای مولد کد. در یکی از مراجع، روش شناسایی بر اساس تصمیم‌گیری سخت معرفی شده است. در این مقاله روش شناسایی در حالت تصمیم‌گیری نرم با مدولاسیون BPSK و کانال با نویز AWGN معرفی می شود. طول کد با بیشینه کردن تابع آنتروپی برای هر دو بعد، تخمین زده می شود. سپس ریشه‌های چندجمله‌ای‌ای مولد جستجو می شود تا چندجمله‌ای اولیه و چندجمله‌ای مولد بازسازی شود. در پایان، نتایج تحلیل و شبیه‌سازی بر اساس تصمیم‌گیری سخت و تصمیم‌گیری نرم ارائه می‌شود و نتایج شناسایی کور بر اساس دو روش با هم مقایسه می‌شوند. با استفاده از خروجی نرم کانال، بازدهی الگوریتم شناسایی بهبود می‌یابد و شبیه‌سازی‌ها بازدهی الگوریتم معرفی‌شده را نشان می‌دهند.

واژه‌های کلیدی: کدهای ضربی، شناسایی کور، کد برداری با تصمیم‌گیری نرم و سخت، آنتروپی پراکندگی اطلاعات ریشه‌ها.

Blind Recognition of Turbo Product Codes in Soft Decision Situations

H. R. Khodadadi[1], Assistant Professor; A. Ghaedi[2], MSc; M. H. Hassan[3], MSc

1- Faculty of Information and communication, Imam Hussein University, Tehran, Iran, Email: hkhodadadi@ihu.ac.ir
2- Faculty of Information and communication, Imam Hussein University, Tehran, Iran, Email: aghaedi@ihu.ac.ir
3- Faculty of Information and communication, Imam Hussein University, Tehran, Iran, Email: mh.hassan@ihu.ac.ir

Abstract: In this paper a method of blind recognition of the coding parameters for binary turbo product codes is proposed. The only knowledge is that the stream is encoded using binary product codes, while the coding parameters are unknown. The recognition processing includes two major procedures: code word length estimation and generator polynomial reconstruction. A hard decision method has been proposed in a previous literature. In this paper we propose the recognition approach in soft decisions situations with Binary-Phase-Shift-Key (BPSK) modulations and Additive-White-Gaussian-Noise (AWGN) channels. The code word length is estimated by maximizing the root information dispersion entropy function (RIDEF) for two dimension of the bit stream. And then we search for the code roots to reconstruct the primitive and generator polynomials. By utilizing the soft output of the channel, the recognition performance is improved and simulations show the efficiency of the proposed algorithm.

Keyword: Turbo product codes, blind recognition, soft and hard decision decoding, roots information dispersion entropy.

تاریخ ارسال مقاله: ۱۳۹۶/۱۱/۱۷
تاریخ اصلاح مقاله: ۱۳۹۷/۰۴/۰۹و ۱۳۹۷/۰۶/۰۵
تاریخ پذیرش مقاله: ۱۳۹۷/۰۷/۲۴
نام نویسنده مسئول: حمید رضا خدادادی.
نشانی نویسنده مسئول: ایران - تهران- انتهای اتوبان شهید بابائی - دانشگاه جامع امام حسین (ع) - دانشکده فناوری اطلاعات و ارتباطات

۱- مقدمه

کدگذاری کانال یکی از اجزاء جدایی ناپذیر در یک سامانه مخابراتی می‌باشد و تاکنون انواع مختلف کدگذاری کانال ارائه شده است. کدهای بلوکی به دلیل سادگی، در کاربردهای عملی بیشتر توسط متخصصان مخابرات مورد توجه قرارگرفته‌اند. یک عملیات بسیار مهم و حیاتی که در سامانه‌های شنود انجام می‌پذیرد، شناسایی کور نوع کد و پارامترهای کدگذاری سیگنال دریافتی است. برای شناسایی کدهای بلوکی روش‌های مختلفی بیان شده است که برخی از آن‌ها فقط مشخصات کلی کدگذار (طول کد و طول پیام) را تعیین می‌کنند. برخی روش‌های موجود علاوه بر مشخصات کلی کد، ساختار کد و چندجمله‌ای مولد[۱] و چندجمله‌ای اولیه[۲] را نیز شناسایی می‌کنند [۱].

کدهای توربو ضربی(TPC[۳]) یک نوع از کدهای ترکیبی هستند که از ترکیب دو کد بلوکی تولید می‌شوند و در ارتباطات مخابرات ماهواره‌ای و غیره استفاده می‌شوند. با توجه به این‌که کدهای توربو ضربی از ترکیب دو کد بلوکی ساخته می شوند، از روش‌هایی که برای شناسایی کدهای بلوکی استفاده می‌شود می‌توان برای شناسایی پارامترهای کدهای توربو ضربی استفاده کرد.

روش جبری برای باز سازی کدهای خطی و کانولوشنال با استفاده از ماتریس دوگان در [۲] پیشنهاد و نشان داده شده که حداقل نسبت سیگنال به نویز موردنیاز برای شناسایی کدی با طول n=32 و k=26 معادل 4.89dB می‌باشد. در [۳] از روش رتبه ماتریس (MR[۴]) و خواص ماتریس برر سی توازن و وزن همینگ جهت شناسایی استفاده شده است. این روش شناسایی با استفاده از ویژگی‌های خطی کدهای بلوکی و با تشکیل ماتریس و محاسبه رتبه ماتریس، اقدام به شناسایی کور پارامترهای کد می‌کند. مزیت این روش پاسخگوئی برای کدهای غیر باینری است. همچنین ذکرشده که میزان محاسبات برای کدی با طول بیش از ۲۵۵ بسیار زیاد است. در [۴] روشی مبتنی بر تصمیم‌گیری نرم جهت شناسایی کدهای چرخشی بیان شده است. این روش از خواص جبری کدهای چرخشی جهت شناسایی طول کد و چندجمله‌ای استفاده می‌کند. مزیت این روش سادگی در پیاده‌سازی و سرعت‌بالا در طول کدهای بلند می‌باشد. در [۵] روش شناسایی کور برای کدهای BCH[۵] بر اساس کمینه کردن وزن سندروم‌ها ارائه شده است. نویسندگان این مقاله روش پیشنهادی خود را با [۴] مقایسه نموده و در ازای از دست دادن زمان بیشتر، بهبود عملکرد را بدست آورده‌اند. در ضمن در این روش پیشنهادی هر چه طول کد بیشتر باشد، احتمال شناسایی کمتر می‌گردد. مرجع [۶] بر اساس روش بزرگ‌ترین شمارنده مشترک (GCD[۶]) چندجمله‌ای مولد را شناسایی می‌کند. این روش تنها در حالت تصمیم سخت بررسی شده و برای شناسائی چند جمله‌ای مولد بهتر از روش تخمین طول عمل می‌کند. مرجع [۷] از روش‌های GCD و رتبه ماتریس به‌صورت هم‌زمان برای شناسایی پارامترهای کدهای توربو ضربی استفاده کرده است. این مقاله نشان می‌دهد که در صد شناسایی برای کدهایی با طول کم در روش GCD بیشتر و از روش رتبه ماتریس سریع‌تر و مطمئن‌تر است. برای مثال در SNR=5 و کد BCH با طول کد $(15^2, 11^2)$ درصد شناسایی با روش GCD، ۷۵ و با رتبه ماتریس ۶۵ درصد می‌باشد. در [۷] برای شناسایی پارامترهای کدهای توربو ضربی از روش تصمیم‌گیری سخت استفاده شده است. مرجع [۸] برای تخمین پارامترهای کد BCH باینری در شرایط نویزی از یک روش جبری مبتنی بر بررسی بیت‌های توازن استفاده کرده که نتایج بهبود عملکرد شناسایی در شرایط نویزی به نسبت عملکرد روش GCD را نشان می‌دهد. در [۹،۱۰] از روش آنتروپی پراکندگی اطلاعات ریشه‌ها (RIDERS[۷]) و تصمیم‌گیری سخت برای شناسایی پارامترهای کدهای BCH استفاده شده و نشان داده که می‌تواند در شرایط نرخ کد بالا و پایین، پارامترهای کد را صحیح شناسایی کند. در این روش وقتی‌که طول کد بزرگ باشد حجم محاسبات زیاد می شود. نویسندگان [۱۱] روش ارائه شده در [۱۰] را توسعه داده و شناسایی کور پارامترهای کدهای BCH را بر اساس تصمیم‌گیری نرم اجرا نموده‌اند که این کار دقت شناسایی را افزایش داده است.

چنانچه گفته شد در [۷] از روش تصمیم‌گیری سخت برای شناسایی پارامترهای کدهای توربو ضربی دوبعدی استفاده شده و در [۱۱] روش آنتروپی ریشه‌ها و تصمیم‌گیری نرم برای شناسایی پارامترهای کدهای BCH که در واقع کدهای یک بعدی هستند پیشنهاد شده است. با توجه به مزیت روش آنتروپی ریشه‌ها و تصمیم‌گیری نرم، ما در این مقاله روش ارائه شده در [۱۱] را توسعه داده‌ایم تا بتواند برای شناسایی پارامترهای کدهای توربو ضربی دوبعدی مورد استفاده قرار گیرد، همچنین الگوریتم ارائه شده را برای مدولاسیون BPSK و در سیگنال به نویزهای مختلف مورد شبیه‌سازی قرار داده و نتایج را با روش تصمیم‌گیری سخت مورد استفاده در مقاله‌های [۱۰،۷،۲] مقایسه کرده‌ایم.

ادامه مقاله به‌صورت زیر سازمان‌دهی شده است:

در بخش دوم روش شناسایی کور پارامترهای کد ضربی بر اساس آنتروپی و مبتنی بر تصمیم‌گیری سخت و تصمیم‌گیری نرم، شرح داده می‌شود. در بخش سوم الگوریتم شناسایی کور پارامترهای کد ضربی بر اساس آنتروپی و مبتنی بر تصمیم‌گیری سخت و تصمیم‌گیری نرم شبیه سازی می شود و نتایج هر دو الگوریتم باهم مقایسه می شود. در بخش چهارم نیز نتیجه‌گیری ارائه می‌شود.

۲- ویژگی روش پیشنهادی

کدهای توربو ضربی از دو کد BCH تشکیل می‌شوند که کد بلوکی $C_1(n_1, k_1, d_1)$ روی سطرهای ماتریس پیام و کد بلوکی $C_2(n_2, k_2, d_2)$ روی ستون‌های ماتریس پیام اعمال می‌شود. برای شناسایی کدهای توربو ضربی نیز لازم است که پارامترهای کدگذاری اعمال‌شده روی سطرها و ستون‌های ماتریس پیام شناسایی شوند. ابتدا پارامترهای کدگذاری مربوط به ستون‌ها با روش آنتروپی تعیین می‌شود سپس پارامترهای کدگذاری مربوط به سطرها شناسایی می‌شود.

الگوریتم شناسایی برای سطرها و ستون‌ها شبیه به هم هستند به همین دلیل، فقط الگوریتم شناسایی کور پارامترهای کد BCH مربوط به ستون‌ها تشریح می‌شود. مزیت این روش شناسایی یک کد دوبعدی با استفاده از روش شناسایی کد تک‌بعدی است.

۲-۱- روش شناسایی آنتروپی بر پایه تصمیم‌گیری سخت

۲-۱-۱- تخمین طول کد و موقعیت هم‌زمانی

برای شناسایی طول کد مربوط به ستون‌ها، مرتبه میدان به‌صورت دلخواه از مقدار m_{min} تا m_{max} در نظر گرفته می‌شود. اگر مرتبه میدان برابر با m در نظر گرفته شود، طول کد از مقدار 2^{m-1} تا مقدار 2^m-1 تغییر می‌کند. اگر طول کد برابر با L در نظر گرفته شود، محل هم‌زمانی و شروع کلمه کد از مقدار ۱ تا مقدار L تغییر داده می‌شود. کوچک‌ترین چندجمله‌ای اولیه مرتبهm به‌عنوان چندجمله‌ای اولیه مبنا در نظر گرفته می‌شود و میدان دودویی گالوا $GF(2^m)$ با آن تولید می شود. به ازای طول کد L، رشته بیت دریافتی به بردارهایی با طول L تبدیل می‌شود. به‌این‌ترتیب M کلمه کد با طول L به دست خواهد آمد. یک دنباله از M کلمه کد معتبر C_1، C_2، ...، C_M مشاهده می‌شود که $C_j(x)$ کلمه کد مربوط به C_j است و مقدار $1 \le j \le M$ است. بردار صحیح $[N_1 \quad N_2 \quad \dots \quad N_i]$ با مقدار اولیه صفر تعریف می‌شود. اگر عنصر α^i به ازای $1 \le i \le 2^m-1$، ریشه کلمه کد $C_j(x)$ باشد، مقدار N_i یک واحد افزایش می یابد $N_i = N_i + 1$ و این کار برای همه کلمه کدها و همه عناصر انجام می‌شود.

مقدار N_i بیان‌گر احتمال این است که عنصر α^i ریشه کد باشد. نکته قابل توجه این است که تمام ریشه‌های کلمه کد $C(x)$، ریشه چندجمله‌ای مولد نیستند ولی تمام ریشه‌های چندجمله‌ای مولد، ریشه کلمه کد $C(x)$ هستند. عناصری که ریشه کلمه کد $C(x)$ هستند اما ریشه چندجمله‌ای مولد نیستند، در کلمه کدهای مختلف متفاوت هستند به خاطر این‌که چندجمله‌ای پیام $u(x)$ در هر عملیات کدگذاری متفاوت هستند. تعداد عناصر میدان توسعه یافته $GF(2^m)$ برابر با 2^m-1 است و مجموعه این عناصر با G_m معرفی می‌شوند. در این مقاله عناصری که ریشه $g(x)$ باشند مشخص می شوند. برای یک کد باینری BCH مبتنی بر میدان $GF(2^m)$، ریشه‌های $g(x)$ را مشخص می‌کنند که با آن ها مجموعه A تشکیل می‌شود. مجموعه A زیرمجموعه G_m است. بنابراین مقادیر N_i متناظر با مجموعه A، بزرگ‌تر از مقادیر N_i متناظر با مجموعه $\overline{A}$ است. اما این شرایط تنها وقتی درست است که داده موردنظر، کلمه کد معتبر باشد. به‌طور مثال طول کلمه کد و موقعیت هم‌زمانی درست باشند. وقتی‌که طول کلمه کد و هم‌زمانی کلمه کد درست نباشند، شرایط گفته شده فوق وجود ندارد و بیت‌های هر کلمه کد به‌صورت تصادفی خواهند بود. در این حالت در مرجع [۱۱ ، ۱۲] فرضیه زیر معرفی می‌شود:

فرضیه ۱: وقتی‌که پارامترهای کدگذاری صحیح نباشند، احتمال این‌که عناصر عضو G_m ریشه کلمه کد باشند، می‌تواند توزیع احتمال یکنواخت داشته باشد، به‌طور مثال N_i دارای توزیع یکنواخت است.

بنابراین در حالت پارامترهای صحیح، آنتروپی توزیع ریشه‌ها باید کمتر از حالت پارامترهای غلط باشد. بر اساس این حقیقت، در مرجع [۱۰و۱۱] تابع آنتروپی پراکندگی ریشه‌ها (RIDEF) معرفی می‌شود تا عدم توازن مقادیر N_i برای α^i مختلف را توصیف کند و پارامترهای صحیح کدگذاری که باعث بیشینه کردن تابع RIDEF می‌شود را معرفی کند. مقدار تابع RIDEF برابر با اختلاف میان آنتروپی در حالت توزیع یکنواخت و آنتروپی در حالت توزیع احتمال ریشه‌های کد است:

$$\Delta H = -\sum_{i=1}^{n}\frac{1}{n}\log\frac{1}{n} - \left(-\sum_{i=1}^{n} p_i \log p_i\right) = \sum_{i=1}^{n} p_i \log p_i + \log(n) \qquad (۱)$$

عبارت n برابر است با تعداد عناصر میدان یعنی $n = 2^m - 1$ است. عبارت p_i برابر احتمال این است که عنصر α^i، ریشه چندجمله‌ای مولد $g(x)$ باشد و به‌صورت زیر محاسبه می‌شود:

$$p_i = \frac{N_i}{\sum_{i=1}^{2^m-1} N_i} \qquad 1 \le i \le 2^m - 1 \qquad (۲)$$

H_{max}: بیشترین مقدار آنتروپی وقتی حاصل می‌شود که احتمال همه عناصر با هم برابر باشند و عناصر میدان توزیع احتمال یکنواخت داشته باشند و $p_i = \frac{1}{n}$ باشد. به‌عبارت‌دیگر، همه عناصر به یک اندازه شانس داشته باشند که ریشه کلمه کد باشند. در فرآیند شناسایی، تمام مقادیر ممکن طول کد و موقعیت هم‌زمانی در الگوریتم گذاشته می‌شود و هر کدام که مقدار تابع آنتروپی ΔH را بیشینه می کند به‌عنوان طول کد و موقعیت هم‌زمانی کد انتخاب می‌شود.

الگوریتم RIDEF برای شناسایی پارامترهای کد BCH بازدهی خوبی دارد، اما این الگوریتم برای حالت تصمیم‌گیری سخت است. در بخش بعد این الگوریتم برای حالت تصمیم‌گیری نرم توسعه داده می‌شود. نتایج کامل شبیه‌سازی تصمیم سخت با ابعاد مختلف از دو روشRIDEF و GCD در [۹] آمده است. که در این شبیه‌سازی‌ها از روش اول استفاده می‌شود.

۲-۲- روش شناسایی آنتروپی بر پایه تصمیم‌گیری نرم

۲-۲-۱- تخمین طول کد و موقعیت هم‌زمانی

برای توسعه الگوریتم RIDEF و تشخیص طول کد و محل هم‌زمانی با روش تصمیم‌گیری نرم، از معادله (۱) استفاده می شود با این تفاوت که برای محاسبه مقادیر p_i، از روشی جدید استفاده می شود که بر اساس داده نرم است. بنابراین معادله (۲) به شکل معادله (۵) تغییر می‌کند. عبارت α^i عنصر میدان دودویی گالوا $GF(2^m)$ است و مقدار آن از α^1 تا α^{2^m-1} تغییر می‌کند. برای محاسبه p_i در حالت تصمیم‌گیری نرم،

ماتریس MPCM[۸] متناظر با α^i در میدان $GF(2^m)$ به‌صورت $H_{min}(\alpha^i)$ طبق جدول (۱) معرفی می‌شود:

جدول ۱: سمبل‌ها در ماتریس $H_{min}(\alpha^3)$ و بردار باینری آن‌ها

Symbol	Polynomial expressions	Vector form
1	1	0 0 1
α	α	0 1 0
α^3	$\alpha+1$	0 1 1
α^6	α^2+1	1 0 1
α^9	α^3	1 0 0
α^{12}	$\alpha^2+\alpha+1$	1 1 1
α^{15}	α	0 1 0
α^{18}	$\alpha^2+\alpha$	1 1 0

$$H_{min}(\alpha^i)=\left((\alpha^i)^{L-1},(\alpha^i)^{L-2},\dots,(\alpha^i)^0\right) \quad (۳)$$

که L طول کد است. طبق تئوری کدگذاری اگر عنصر α^i ریشه کد $C_j(x)$ باشد، خواهیم داشت:

$$H_{min}(\alpha^i)\times C_j=0 \quad (۴)$$

خروجی نرم کانال می‌تواند اطلاعات بهتری در مورد قابلیت اطمینان برای تصمیم‌گیری هر سمبل فراهم کند. در الگوریتم تصمیم‌گیری نرم، عبارت $p_{j,i}$ محاسبه می‌شود که برابر احتمال این است که عنصر α^i، ریشه کلمه کد $C_j(x)$ باشد. سپس مقدار p_i طبق رابطه زیر محاسبه می‌شود:

$$p_i=\frac{\sum_{j=1}^{M}p_{j,i}}{\sum_{i=1}^{2^m-1}\sum_{j=1}^{M}p_{j,i}} \quad (۵)$$

برای محاسبه $p_{j,i}$، عناصر ماتریس MPCM به شکل باینری تبدیل شده و $Hb_{min}(\alpha^i)$ نامیده می‌شوند. برای مثال با m=3، عنصر α^3، میدان $GF(2^3)$ و با چندجمله‌ای اولیه $P(x)=x^3+x+1$، طبق رابطه (۳) ماتریس MPCM شبیه زیر به دست می‌آید:

$$L=2^3-1=7$$

$$H_{min}(\alpha^3)=(\alpha^{18}\ \alpha^{15}\ \alpha^{12}\ \alpha^9\ \alpha^6\ \alpha^3\ 1) \quad (۶)$$

چندجمله‌ای اولیه $P(x)=x^3+x+1$، به‌عنوان مبنا در نظر گرفته می‌شود. دیگر عنصرها نیز به همین روش ساخته می‌شوند. درنهایت ماتریس MPCM باینری به شکل زیر ساخته می‌شود.

$$Hb_{min}(\alpha^3)=\begin{pmatrix}1&1&1&1&1&0&0\\1&0&1&0&0&1&0\\0&1&1&0&1&1&1\end{pmatrix} \quad (۷)$$

نکته: تعداد سطرهای ماتریس $Hb_{min}(\alpha^3)$ برابر m است. ماتریس سندروم متناظر با $Hb_{min}(\alpha^3)$ و کلمه کد C_j طبق زیر محاسبه می‌شود:

$$\mathbf{S}_{j,i}=[S_{j,1}\ S_{j,2},\dots\ S_{j,m}]^T=Hb_{min}(\alpha^i)\times C_j \quad (۸)$$

عبارت $C_j(x)$ چندجمله‌ای مرتبط با کلمه کد C_j است. اگر عنصر α^i متناظر با $Hb_{min}(\alpha^i)$، ریشه کلمه کد $C_j(x)$ باشد، آنگاه رابطه $\mathbf{S}_{j,i}=0$ برقرار است. بنابراین احتمال این‌که رابطه $\mathbf{S}_{j,i}=0$ برقرار باشد را می‌توان با محاسبه میانگین احتمال $p(S_{j,k}=0)$ برای مقادیر $1\le k\le m$، طبق زیر محاسبه کرد:

$$p_{j,i}=p_r(\mathbf{S}_{j,i}=0)=\frac{1}{m}\sum_{k=1}^{m}p_r(S_{j,k}=0) \quad (۹)$$

در معادله بالا، از عبارت $p_r(\varphi)$ برای نشان دادن احتمال وقوع φ استفاده می‌شود. فرض می‌شود که فرستنده از مدولاسیون BPSK استفاده می‌کند. برای مثال فرستنده، ارقام $+1$ و -1 را به ترتیب به‌عنوان مدوله‌شده سمبل‌های 0 و 1 در نظر می‌گیرید. مدولاسیون بیت c_u را به سمبل s_u تبدیل می‌کند:

$$s_u=1-2c_u,\quad u=1,2,3,\dots \quad (۱۰)$$

فرض می‌شود که کانال انتشاری از نوع کانال باینری متقارن[۹] است که با نویز AWGN با واریانس $\sigma_n^2=\frac{N_0}{2}$ تخریب شده است. در گیرنده، سمبل r_u با تصمیم‌گیری نرم شبیه زیر بیان می‌شود که w_u دنباله نویز است.

$$r_u=s_u+w_u \qquad u=1,2,3,\dots \quad (۱۱)$$

ثابت می‌شود که احتمال $p_r(S_{j,k}=0)$ را می‌توان طبق زیر محاسبه کرد [۱۴]:

$$p_r(S_{j,k}=0)=\frac{1}{2}+\frac{1}{2}\prod_{u=1}^{n_k}\tanh\left(\frac{r_u}{\sigma_n^2}\right),\ 1\le k\le m \quad (۱۲)$$

در رابطه فوق n_k، تعداد یک موجود در سطر شماره k از ماتریس $Hb_{min}(\alpha^i)$ است و $1\le i\le 2^m-1$ است.

۲-۲-۲- پردازش تطبیقی ماتریس MPCM

به خاطر این‌که ماتریس $Hb_{min}(\alpha^i)$ تنک و خلوت نیست، یک اشتباه در تصمیم‌گیری یک بیت، تأثیر منفی در محاسبه سندروم‌ها دارد. نظر به این‌که احتمال تصمیم‌گیری نامطمئن بیت بیشتر از احتمال تصمیم‌گیری مطمئن است، مرجع [۱۳] الگوریتم پردازش تطبیقی ماتریس $Hb_{min}(\alpha^i)$ را معرفی کرده است تا اثر منفی تصمیم‌گیری نامطمئن بیت را کاهش دهد. جزئیات مراحل پردازش تطبیقی برای ماتریس $Hb_{min}(\alpha^i)$ و کلمه کد C_j با طول کد L در زیر لیست شده است:

مرحله ۱: ماتریس کنترل توازن $Hb_{min}(\alpha^i)$ و کلمه کد C_j^T باهم ترکیب می‌شوند و ماتریس جدید طبق زیر ساخته می‌شود که مقادیر r_1 و r_2 و ... r_L بیت‌های نرم کلمه کد C_j هستند.

$$H^{*}(\alpha^{i})=\left[\begin{array}{c|ccc} r_1 & r_2 & \dots\dots & r_L \\ \hline h_{11} & h_{12} & \dots\dots & h_{1L} \\ h_{21} & h_{22} & \dots\dots & h_{2L} \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ h_{m1} & h_{m2} & \dots\dots & h_{mL} \end{array}\right] \qquad (۱۳)$$

مرحله ۲: از مقادیر r_u، قدر مطلق گرفته می‌شود و آن ها از کوچک به بزرگ و از چپ به راست مرتب می‌شوند و ستون‌های ماتریس طبق آن ها جابه جا و مرتب می شوند و ماتریس جدید با نام $H_r^{*}(\alpha^{i})$ ساخته می‌شود، البته شماره ستون‌های ماتریس اصلی ذخیره می‌شود. مقدار قدر مطلق r_u، نشان‌دهنده اطمینان به بیت نرم دریافتی است.

$$H_r^{*}(\alpha^{i})=\left[\begin{array}{c|ccc} |r_{i_1}| & |r_{i_2}| & \dots\dots & |r_{i_L}| \\ \hline h_{1i_1} & h_{1i_2} & \dots\dots & h_{1i_L} \\ h_{2i_1} & h_{2i_2} & \dots\dots & h_{2i_L} \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ h_{mi_1} & h_{mi_2} & \dots\dots & h_{mi_L} \end{array}\right] \qquad (۱۴)$$

مرحله ۳: با عملیات مقدماتی، ماتریس $H_r^{*}(\alpha^{i})$ به شکل زیر تغییر می‌کند. البته سطر اول در عملیات مقدماتی شرکت نمی‌کند.

$$H_r^{*}(\alpha^{i})=\left[\begin{array}{c|ccc} |r_{i_1}| & |r_{i_2}| & \dots\dots & |r_{i_L}| \\ \hline 1 & x & \dots\dots & x \\ 0 & x & \dots\dots & x \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ 0 & x & \dots\dots & x \end{array}\right] \qquad (۱۵)$$

عملیات مقدماتی روی ماتریس $H_r^{*}(\alpha^{i})$ ادامه داده می‌شود تا تعداد عناصر با مقدار ۱ در هر ستون، به یک محدود شود.

$$\left[\begin{array}{c|ccccc} |r_{i_1}| & |r_{i_2}| & |r_{i_3}| & \dots\dots & |r_{i_L}| \\ \hline 1 & 0 & 0 & \dots\dots & x \\ 0 & 1 & 0 & \dots\dots & x \\ \vdots & \vdots & 1 & \ddots & x \\ \vdots & \vdots & \vdots & \ddots & x \\ 0 & 0 & 0 & \dots\dots & x \\ 0 & 0 & 0 & \dots\dots & x \end{array}\right] \qquad (۱۶)$$

وقتی‌که تبدیل مقدماتی غیرممکن شد، عملیات تبدیل متوقف می‌شود. با m ردیف آخر ماتریس $H_r^{*}(\alpha^{i})$، یک ماتریس جدید ساخته می‌شود. در ماتریس جدید ستون‌ها به محل اصلی خودشان برمی‌گردد و ماتریس حاصل $Hb_{min\_a}(\alpha^{i})$ نامیده می‌شود. بنابراین احتمال $p_r(S_{j,k}=0)$ در معادله (۱۲) بر اساس ماتریس $Hb_{min\_a}(\alpha^{i})$ محاسبه می‌شود.

اگر رابطه $rank\left(Hb_{min\_a}(\alpha^{i})\right)=m$ برقرار باشد، ناحیه $m\times m$ سمت چپ ماتریس $Hb_{min\_a}(\alpha^{i})$، یک ماتریس همانی است. اگر $rank\left(Hb_{min\_a}(\alpha^{i})\right)<m$ برقرار باشد، ماتریس $Hb_{min\_a}(\alpha^{i})$ شبیه زیر می‌شود:

$$Hb_{min\_a}(\alpha^{i})=\begin{bmatrix} 1 & 0 & 0 & \cdots & x & x \\ 0 & 1 & 0 & \dots & x & x \\ 0 & 0 & 1 & \dots & x & x \\ 0 & 0 & 0 & \dots & x & x \\ 0 & 0 & 0 & \dots & x & x \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 \end{bmatrix} \qquad (۱۷)$$

در این حالت فقط p سطر از ماتریس، غیر صفر است که p رتبه ماتریس $Hb_{min\_a}(\alpha^{i})$ است. $m-p$ سطر ماتریس که صفر هستند در محاسبه $p_r(S_{j,k}=0)$ دخالت ندارند. بنابراین معادله (۹) به صورت معادله (۱۸) اصلاح می‌شود:

$$p_{j,i}=\frac{1}{rank\left(Hb_{min\_a}(\alpha^{i})\right)}\sum_{k=1}^{rank\left(Hb_{min\_a}(\alpha^{i})\right)}p_r(S_{j,k}=0) \qquad (۱۸)$$

نکته: عناصر $\alpha^{2^m-1}=1$ فقط وقتی ریشه چندجمله‌ای کلمه کد است که وزن چندجمله‌ای کلمه کد زوج باشد. این واقعیت فرض توزیع احتمال یکنواخت p_i را به هم می ریزد چون که احتمال این که وزن چندجمله‌ای کلمه کد زوج باشد برابر با $\frac{1}{2}$ است و مقدار p_i مربوط به عنصر $\alpha^{2^m-1}=1$ بزرگ می‌شود. همچنین تمام درایه‌های ماتریس MPCM مربوط به عنصر $\alpha^{2^m-1}=1$، یک هستند و باعث می شود که $p_{j,i}$ مقدار بزرگی شود که این نیز باعث بزرگ شدن مقدار p_i مربوط به عنصر $\alpha^{2^m-1}=1$ می‌شود. برای جلوگیری از اشتباه بالا، از عنصر $\alpha^{2^m-1}=1$ صرف‌نظر می‌شود و معادله‌های (۱) و (۵) به شکل زیر تغییر می‌کنند:

$$p_i=\frac{\sum_{j=1}^{M}p_{j,i}}{\sum_{i=1}^{2^m-2}\sum_{j=1}^{M}p_{j,i}} \qquad (۱۹)$$

$$\Delta H=\log(2^m-2)+\sum_{i=1}^{2^m-2}p_i\log p_i \qquad (۲۰)$$

مقدار n_k برای سطرهای مختلف ماتریس $Hb_{min\_a}(\alpha^{i})$ یکسان نیست. این موضوع روی محاسبه احتمال $p_r(S_{j,k}=0)$ اثر دارد. به همین دلیل معادله شماره (۱۲) به شکل زیر اصلاح می‌شود:

$$p_r[s_{j,k}=0]=\frac{1}{2}+\frac{1}{2}\prod_{u=1}^{n_k}sign(r_u)\times\left|\prod_{u=1}^{n_k}\tanh\left(\frac{r_u}{\sigma^2}\right)\right|^{\frac{1}{n_k}} \qquad (۲۱)$$

۲-۲-۳- خلاصه مراحل شناسایی طول کلمه کد:

از معادله‌های (۱۸)، (۱۹)، (۲۰) و (۲۱) طبق مراحل زیر، جهت تعیین طول کد و موقعیت هم‌زمانی کد و مرتبه چندجمله‌ای اولیه استفاده می‌شود. فرض بر این است که گیرنده شنودکننده می‌داند که نوع کدگذاری ضربی استفاده شده است.

مرحله ۱: مرتبه میدان از m_{min} تا m_{max} در نظر گرفته می‌شود. برای هر m موردنظر، چندجمله‌ای اولیه با کم‌ترین وزن به‌عنوان

چندجمله‌ای اولیه در نظر گرفته می‌شود و میدان $GF(2^m)$ طبق آن به دست می‌آید.

مرحله ۲: به مرتبه میدان m، مقدار اولیه $m = m_{min}$ داده می‌شود.

مرحله ۳: برای m موردنظر، کمترین طول کد $L = 2^{m-1}$ تعیین می‌شود و رشته بیت دریافتی به بلوک‌های L بیتی تبدیل می‌شود تا M کلمه کد با طول L به دست بیاید.

مرحله ۴: موقعیت هم‌زمانی صفر $(T = 0)$ در نظر گرفته می‌شود که به‌منزله بیت اول رشته بیت دریافتی است.

مرحله ۵: به ازای مرتبه میدان m و طول کلمه کد L و موقعیت هم‌زمانی T، مقدار ΔH طبق روابط ذکرشده تعیین می‌شود.

مرحله ۶: اگر شرط $T< L$ صحیح باشد، موقعیت هم‌زمانی یک واحد افزایش داده شده $(T = T + 1)$ و مرحله ۵ اجرا می‌شود، در غیر این صورت اگر شرط T=L صحیح باشد، مرحله ۷ اجرا می‌شود.

مرحله ۷: اگر شرط $L < 2^m - 1$ صحیح باشد، طول کد یک واحد افزایش داده شده $(L = L + 1)$ و مرحله ۴ اجرا می‌شود، در غیر این صورت اگر شرط $L = 2^m - 1$ صحیح باشد، مرحله ۸ اجرا می‌شود.

مرحله ۸: اگر شرط $m < m_{max}$ صحیح باشد، مرتبه m یک واحد افزایش داده شده $(m = m + 1)$ و مرحله ۳ اجرا می‌شود، در غیر این صورت اگر شرط $m = m_{max}$ صحیح باشد، مرحله ۹ اجرا می‌شود.

مرحله ۹: در مراحل قبل، تمام مقادیر ΔH برای تمام حالت‌های L، m، T تعیین شد. بزرگ‌ترین مقدار ΔH مشخص می‌شود. مقادیر L، m، T متناظر با بزرگ‌ترین ΔH، به‌عنوان طول کد، مرتبه چندجمله‌ای اولیه و موقعیت هم‌زمانی کد معرفی می‌شوند.

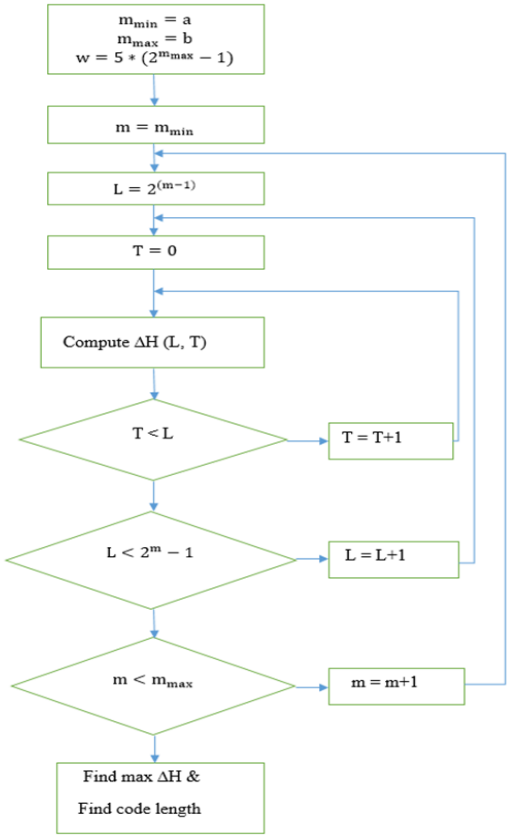
فلوچارت مراحل تعیین طول کلمه کد و موقعیت هم‌زمانی، در شکل ۱ مشاهده می‌شود.

۲-۳- شناسایی ریشه‌های چندجمله‌ای مولد

در بخش قبل طول کلمه کد، محل هم‌زمانی و مرتبه میدان تخمین زده شد. چندجمله‌ای اولیه مرتبه m با کمترین وزن به‌عنوان چندجمله‌ای اولیه مبنا در نظر گرفته می‌شود و میدان گالوا $GF(2^m)$ با آن تولید می‌شود که عناصر غیر صفر میدان گالوا به‌صورت $(\alpha^1, \alpha^2, \dots \alpha^{2^m-1})$ خواهند بود. رشته بیت دریافتی به بردارهایی با طول L تبدیل می‌شود که M کلمه کد با طول L به دست خواهد آمد.

تعریف: برای عنصر α^i عبارت $LLR^{۱۰}$، برابر است با احتمال این‌که عنصر α^i ریشه چندجمله‌ای مولد باشد تقسیم بر احتمال این‌که عنصر α^i ریشه چندجمله‌ای مولد نباشد که البته به‌صورت لگاریتمی بیان می‌شود. هرچه مقدار LLR برای یک عنصر بزرگ‌تر باشد، احتمال این‌که آن عنصر ریشه چندجمله‌ای مولد باشد، بیشتر است.

برای تعیین کردن چندجمله‌ای مولد g(x)، مقادیر LLR با روابط زیر برای تمام عناصر میدان $GF(2^m)$ محاسبه می‌شود. عناصری که دارای LLR بزرگ‌تری باشند، می‌توانند به‌عنوان ریشه‌های چندجمله‌ای مولد g(x) باشند و با این ریشه‌ها، g(x) ساخته می‌شود. در این بخش نیز از عنصر $\alpha^{2^m-1} = 1$ صرف‌نظر می‌شود و از روش پردازش تطبیقی ماتریس MPCM استفاده می‌شود.



شکل ۱: مراحل تعیین طول کلمه کد و موقعیت هم‌زمانی

$$L(\alpha^i) = L[p_r(S_i = 0)] = \sum_{j=1}^{M} L[p_r(S_i = 0)] = \sum_{j=1}^{M}\left[\frac{1}{\text{rank}(Hb_{min}(\alpha^i))}\sum_{k=1}^{\text{rank}(Hb_{min}(\alpha^i))} L(s_{j,k} = 0)\right] \quad (۲۲)$$

$$L(s_{j,k} = 0) = \log\frac{p_{r(s_{j,k}=0)}}{p_{r(s_{j,k}\neq 0)}} = 2\text{artanh}\left[\prod_{u=1}^{n_k}\text{sign}(r_u) \times \left|\prod_{u=1}^{n_k}\tanh\left(\frac{r_u}{\sigma^2}\right)\right|^{\frac{1}{n_k}}\right] \quad (۲۳)$$

مراحل تعیین ریشه‌های چندجمله‌ای مولد g(x) در زیر معرفی می‌شود:

مرحله ۱: مقدار LLR برای تمام عناصر غیر صفر میدان محاسبه می‌شود و بردار L_L شبیه زیر تشکیل می‌شود. از عنصر $\alpha^{2^m-1} = 1$ صرف‌نظر می‌شود.

$$L_L = [\,L(\alpha^1) \quad L(\alpha^2) \quad \dots\dots \quad L(\alpha^{2^m-2})\,] \quad (۲۴)$$

مرحله ۲: بردار L_L از کوچک به بزرگ مرتب می‌شود و بردار جدید با L_{LR} نشان داده می‌شود. شماره ستون‌های بردار اولیه ذخیره می‌شود.

مرحله ۳: اختلاف بین عناصر کنار هم در بردار L_{LR} محاسبه می‌شود.

$$dL(i) = L_{LR}(i+1) - L_{LR}(i), 1 \le i \le L-1 \quad (۲۵)$$

مرحله ۴: بیشترین مقدار dL مشخص می‌شود و مقدار i متناظر با آن ذخیره و در مرحله بعد از آن استفاده می‌شود.

مرحله ۵: در بردار L_{LR} از عنصر شماره i+1 تا آخرین عنصر انتخاب می شود. سپس موقعیت آن‌ها در بردار L_L به د ست می‌آید. در میدان $GF(2^m)$، عنا صر متناظر با موقعیت آن‌ها م شخص می شود. این عناصر، ریشه‌های چندجمله‌ای مولد $g(x)$ هستند.

عنصر $\alpha^{2^m-1} = 1$: برای سادگی، ماتریس MPCM باینری مربوط به ریشه‌ای که بزرگ‌ترین مقدار LLR دارد انتخاب شده و عناصر آن NOT می شود. از این ماتریس برای محاسبه $L(\alpha^{2^m-1})$ استفاده می‌شود. سپس مرحله ۲ به بعد شبیه مراحل قبل اجرا می‌شود و ریشه‌های چندجمله‌ای مولد به دست می‌آید.

می توان چندجمله‌ای مولد $g(x)$ را بر اساس ریشه‌های آن تولید نمود:

$$g(x) = (x-\alpha_1)(x-\alpha_2)\dots\dots(x-\alpha_p) \quad (۲۶)$$

در رابطه فوق، α_1، α_2 ... α_p ریشه‌های چندجمله‌ای مولد هستند که در این بخش به دست آمدند.

۲-۴- شناسایی چندجمله‌ای اولیه p(x)

بعد از شناسایی طول کد بر اساس چندجمله‌ای اولیه تصادفی $p(x)$ و مرتبه میدان m، می‌توان تعداد ریشه‌ها و قابلیت تصحیح خطا را تعیین کرد. سپس برای تمام چندجمله‌ای های اولیه متفاوت، میدان های $GF(2^m)$ متفاوت تولید می شود و عناصر هر میدان تولید می شود. هرکدام از این عناصر که در خاصیت کد BCH صدق کنند به‌عنوان ریشه چندجمله‌ای مولد در نظر گرفته می شود و چندجمله‌ای اولیه متناظر با آن به‌عنوان چندجمله‌ای اولیه معرفی می‌شود.

خلاصه مراحل تعیین چندجمله‌ای اولیه:

مرحله ۱: با استفاده از برنامه متلب، چندجمله‌ای‌های اولیه مرتبه m به دست می آید و آن ها به ترتیب $P_1(x)$ و $P_2(x)$ و ... نامیده می‌شوند.

مرحله ۲: به ازای $P_1(x)$، عنا صر میدان $GF(2^m)$ تولید می شود و ریشه‌های چندجمله‌ای مولد بر اساس بخش قبل به دست می‌آیند. این ریشه‌ها از کوچک به بزرگ مرتب می‌شوند.

مرحله ۳: برای همه چندجمله‌ای های اولیه مرتبه m موجود در مرحله ۱، عملیات مرحله ۲ انجام می‌گیرد.

مرحله ۴: هر کدام از مجموعه ریشه‌ها که دارای 2t ریشه متوالی باشند و بیشترین شباهت را با اصول کدهای BCH داشته باشند، به‌عنوان مجموعه اصلی در نظر گرفته می شود. $P(x)$ مرتبط با مجموعه اصلی به‌عنوان چندجمله‌ای اولیه مربوط به کد معرفی می‌شود.

۳- شبیه‌سازی

۳-۱- شناسایی طول کد مبتنی بر تصمیم‌گیری سخت

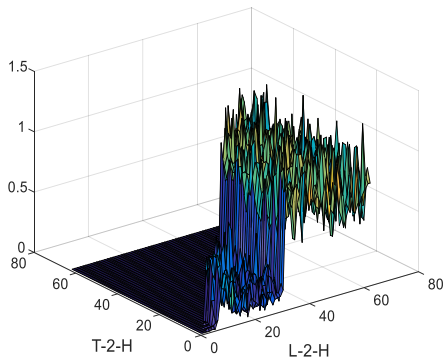
مثال ۳-۱: حالت غیر سنکرون با تصمیم‌گیری سخت

در این مثال برای بعد اول (سطرها)، از کد BCH با پارامترهای $C_1(63,51,103)$ و برای بعد دوم (ستون‌ها) از کد BCH با پارامترهای $C_2(63,45,109)$ استفاده شده است. مقادیر ۱۰۳ و ۱۰۹ مربوط به چندجمله‌ای اولیه هستند. نسبت سیگنال به نویز SNR برابر با ۵ است. مقدار کوتاه‌شدگی بعد اول، برابر با ۵ و مقدار کوتاه‌شدگی بعد دوم برابر با ۷ است. در این مثال رشته بیت دریافتی غیر سنکرون است و محل هم‌زمانی برای هر دو بعد را باید شناسایی کرد که باعث می‌شود سرعت اجرای الگوریتم پایین بیاید.

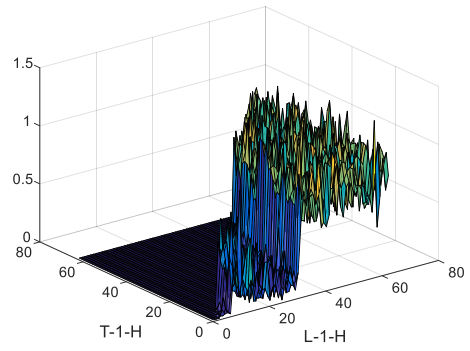
متغیر DTA رشته بیت خروجی دمدولاتور BPSK است که مقادیر دیجیتال دارد. ۱۰ بیت ابتدای DTA حذف می‌شود تا شروع کد در رشته بیت دریافتی بیت اول نباشد و محل هم‌زمانی در حالت غیر سنکرون شود. مقدار هم‌زمانی به‌منظور تشابه بیشتر رشته بیت به داده واقعی صورت گرفته است.

از الگوریتم آنتروپی مبتنی بر تصمیم‌گیری سخت برای شناسایی پارامترهای کدهای توربو ضربی استفاده می شود. ابتدا پارامترهای کدگذاری مربوط به ستون‌ها با روش آنتروپی مبتنی بر تصمیم‌گیری سخت تعیین می شود سپس پارامترهای کدگذاری مربوط به سطرها شناسایی می‌شود. در شکل۲، طول کلمه کد و محل هم‌زمانی مربوط به ستون‌ها، با روش آنتروپی مبتنی بر تصمیم‌گیری سخت، برای بعد اول تخمین زده می‌شود.

همان‌طور که در شکل ۲ و ۳ مشاهده می‌شود، الگوریتم شناسایی مبتنی بر تصمیم‌گیری سخت، طول کد و محل هم‌زمانی برای ستون‌ها و سطرها را با دقت کمی شناسایی می‌کند و احتمال شناسایی غلط، زیاد است. اشکال روش در این است که مقادیر استخراجی ازنظر مقدار عددی به هم نزدیک بوده و لذا عمل ماکزیمم گیری همراه با خطاست.



شکل۲: شناسایی طول کلمه کد بعد دوم با تصمیم‌گیری سخت



شکل ۳: شناسایی طول کلمه کد بعد اول با تصمیم‌گیری سخت

طول کلمه کد مربوط به بعد اول و بعد دوم، با روش آنتروپی مبتنی بر تصمیم گیری سخت به ترتیب ۴۴ و ۵۶ به‌دست‌آمده که غلط شناسایی شده اند و با پارامترهای کدگذاری در ورودی مثال، برابر نیستند. چون طول کلمه کد غلط شده است، دیگر پارامترهای کدگذاری نیز غلط شناسایی می‌شوند.

نتایج شناسایی با روش آنتروپی مبتنی بر تصمیم‌گیری سخت، در زیر مشاهده می‌شود:

=======Hard Detection =========

Estimated N_1, N_2: 44, 56
Estimated P_1, P_2: 67, 67
Estimated K_1, K_2: 56, 44
Estimated S_1, S_2: 19, 7

که در آن N نمایشگر طول کلمه کد، K طول پیام، S مقدار کوتاه شدگی و P نشانگر چندجمله‌ای مولد است.

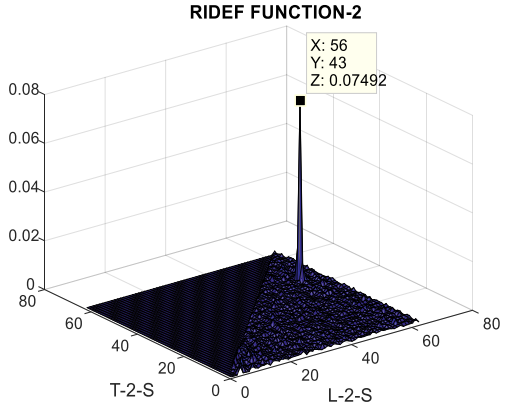
در بخش بعد مشاهده می‌شود که در شرایط آزمون یکسان، الگوریتم شناسایی مبتنی بر تصمیم‌گیری نرم دقت خیلی بهتری دارد.

۳-۲- شناسایی طول کد مبتنی بر تصمیم‌گیری نرم

مثال ۳-۲: حالت غیر سنکرون با تصمیم‌گیری نرم

از الگوریتم آنتروپی مبتنی بر تصمیم گیری نرم برای شناسایی پارامترهای کدهای توربو ضربی استفاده می‌شود. همانند مثال قبل همزمانی در حالت غیر سنکرون فرض می‌شود.

در شکل ۴، با روش آنتروپی مبتنی بر تصمیم‌گیری نرم، طول کلمه کد و محل همزمانی مربوط به ستون‌ها تخمین زده می‌شود. محور X طول کلمه کد، محور Y محل همزمانی و محور Z مقدار تابع ΔH است. طول کلمه کد بعد دوم برابر ۵۶ و محل همزمانی برابر با ۴۳ تشخیص داده شده است که صحیح هستند.



شکل ۴: شناسایی طول کلمه کد بعد دوم با تصمیم‌گیری نرم

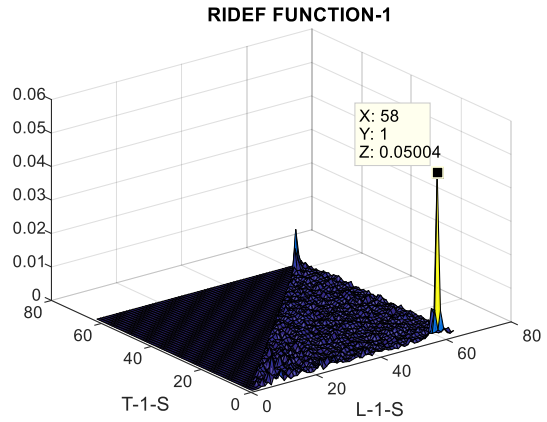
در شکل ۵، طول کلمه کد و محل همزمانی مربوط به سطرها تخمین زده می‌شود. همان‌طور که در شکل ۵ مشاهده می‌شود با استفاده از الگوریتم آنتروپی، طول کلمه کد بعد اول برابر با ۵۸ و محل همزمانی برابر با ۱ تشخیص داده شده است که صحیح هستند.

با مقایسه نمودارهای موجود در بخش ۳-۱ با نمودارهای موجود در بخش ۳-۲، ملاحظه می شود که با وجود شرایط آزمون یکسان، دقت شناسایی پارامترهای کدهای توربو ضربی مبتنی بر تصمیم‌گیری نرم، بسیار بهتر از دقت شناسایی پارامترهای کدهای توربو ضربی مبتنی بر تصمیم‌گیری سخت است. نتایج شناسایی با روش آنتروپی مبتنی بر تصمیم‌گیری نرم، در زیر مشاهده می‌شود:

=======Soft Detection =========

Estimated N_1, N_2: 58, 56
Estimated P_1, P_2: 103, 109
Estimated K_1, K_2: 46, 38
Estimated S_1, S_2: 5, 7

طول کلمه کد مربوط به بعد اول و بعد دوم، با روش آنتروپی مبتنی بر تصمیم‌گیری نرم به ترتیب ۵۸ و ۵۶، چندجمله‌ای اولیه مربوط به بعد اول و دوم، به ترتیب ۱۰۳ و ۱۰۹، طول پیام مربوط به بعد اول و دوم، به ترتیب ۳۸ و ۴۶ به‌دست‌آمده که همگی صحیح هستند. همچنین مقدار کوتاه‌شدگی مربوط به بعد اول و دوم، به ترتیب ۵ و ۷ به‌دست‌آمده که این مقادیر نیز صحیح هستند. با توجه به نتایج حاصل‌شده از شبیه‌سازی اولیه و تفاوت در شکل‌های استخراج‌شده، استفاده از تابع آنتروپی برای یافتن مقادیر طول کد، طول پیام، مقدار کوتاه شدگی و میزان هم‌زمان‌سازی الگوریتم در روش تصمیم‌نرم، پیچیدگی محاسباتی بیشتری دارد. از طرف دیگر الگوریتم سخت، دستیابی به نمودار سریع‌تر ولی تعیین و تشخیص مقدار ماکزیمم تابع آنتروپی قابل‌اعتماد و لزوماً صحیح نیست.



شکل ۵: شناسایی طول کلمه کد بعد اول با تصمیم‌گیری نرم

۳-۳- شناسایی ریشه‌های چندجمله‌ای مولد g(x)

همان طور که در بخش ۲-۳ مشاهده شد، برای تعیین کردن چندجمله‌ای مولد g(x)، مقادیر LLR برای تمام عناصر میدان $GF(2^m)$ محاسبه می شود. عناصری که دارای LLR بزرگ‌تری باشند، می‌توانند به‌عنوان ریشه چندجمله‌ای مولد g(x) باشند و با این ریشه‌ها، g(x) ساخته می‌شود. در شکل ۶، مراحل مختلف شناسایی چندجمله‌ای مولد مربوط به ستون‌های ماتریس کد، نشان داده می‌شود.

در شکل ۶ الف، مقدار LLR برای عناصر مختلف میدان به‌جز عنصر α^{2^m-1} نمایش داده می‌شود. در شکل ۶ ب، مقدار LLR برای عنصر α^{2^m-1} محاسبه می‌شود و به شکل الف اضافه می‌شود. در شکل ۶ پ، بردار LLR از کوچک به بزرگ مرتب می‌شود. در شکل ۶ ت، اختلاف بین LLR مربوط به تمام عناصر میدان، نمایش داده می‌شود.

در شکل ۷، مراحل مختلف شناسایی چندجمله‌ای مولد مربوط به سطرهای ماتریس کلمه کد، نشان داده می‌شود. در شکل ۷ الف، مقدار LLR برای تمام عناصر میدان به‌جز عنصر α^{2^m-1}، نمایش داده می‌شود. در شکل ۷ ب، مقدار LLR برای عنصر α^{2^m-1} محاسبه می‌شود و به شکل الف اضافه می‌شود. در شکل ۷ پ، بردار LLR از کوچک به بزرگ مرتب می‌شود. در شکل ۷ ت، اختلاف بین LLR مربوط به تمام عناصر میدان، نمایش داده می‌شود.

در این مثال برای بعد اول (سطرها)، از کد BCH با پارامترهای $C_1(63,51,103)$ و برای بعد دوم (ستون‌ها) از کد BCH با پارامترهای $C_2(63,45,109)$ استفاده شده است. مقادیر ۱۰۳ و ۱۰۹ مربوط به چندجمله‌ای اولیه هستند. با روش آنتروپی چندجمله‌ای مولد مربوط به کدگذاری بعد اول $g_1(x)$ و کدگذاری بعد دوم $g_2(x)$، به‌صورت زیر شناسایی شده است.

$$g_1(x)\_EST = 1 + x + x^3 + x^5 + x^7 + x^9 + x^{12} \quad (۲۷)$$

$$g_2(x)\_EST = 1 + x + x^4 + x^5 + x^9 + x^{11} + x^{13} + x^{15} + x^{18} \quad (۲۸)$$

با توجه به اصول تئوری کدگذاری و ساختار کدهای BCH، چندجمله‌ای مولد مربوط به کدگذاری بعد اول و کدگذاری بعد دوم ساخته شده است که در زیر مشاهده می‌شود:

$$g_1(x)\_theory = 1 + x + x^3 + x^5 + x^7 + x^9 + x^{12} \quad (۲۹)$$

$$g_2(x)\_theory = 1 + x + x^4 + x^5 + x^9 + x^{11} + x^{13} + x^{15} + x^{18} \quad (۳۰)$$

با مقایسه چندجمله‌ای مولد خروجی الگوریتم و خروجی تئوری، مشاهده می‌شود که رابطه (۲۷) با رابطه (۲۹) برابر و رابطه (۲۸) با رابطه (۳۰) برابر هستند که نشان‌دهنده صحت عملکرد الگوریتم آنتروپی مبتنی بر تصمیم‌گیری نرم است.

در ادامه نتایج شناسایی برای کدهای استاندارد مورد استفاده در ارتباطات ماهواره‌ای که در استانداردهای CCSDS[11] معرفی شده موردبررسی قرار می‌گیرد [۱۴]. الگوریتم تشخیص پارامترهای طول کد برای کدهای عملی مورد استفاده در استاندارد IEES , CCSDS[12] که دو بعد باهم برابر هستند، در جداول (۲) و (۳) و برای دو مقدار نسبت سیگنال به نویز مختلف نشان آورده شده است. این جدول در شرایطی که داده‌ها سنکرون هستند و فقط یک بعد از کد شناسایی شود، شبیه‌سازی شده است. در این شبیه‌سازی جست‌وجو طول کد از ۱۵ الی ۱۲۷ بوده است.

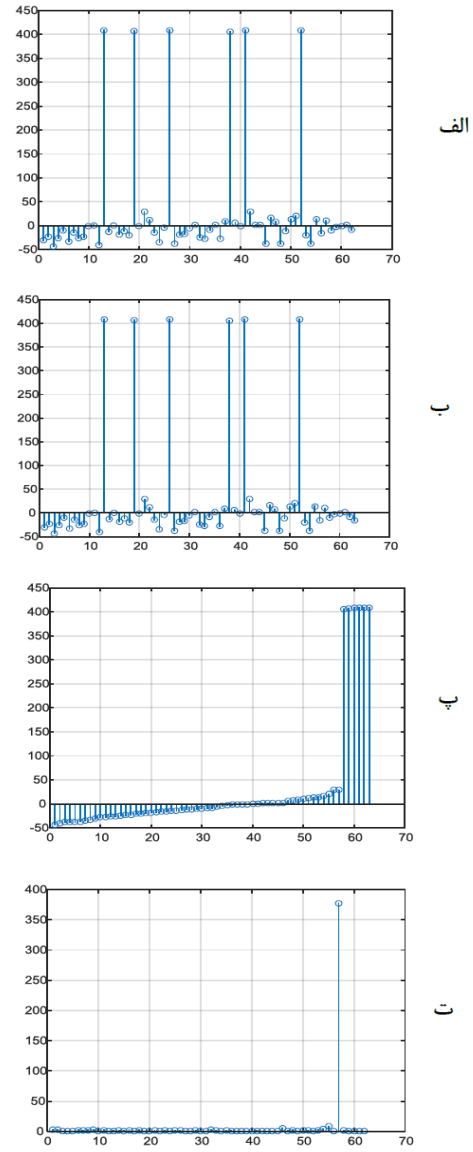
همان‌طور که از جدول(۲) و جدول(۳) ملاحظه می‌شود زمان تشخیص برای الگوریتم تصمیم نرم بیشتر است ولی دقت شناسایی آن دقیق‌تر است. ازآنجاکه شناسایی ساختار سامانه‌ها یک پردازش برخط نیست مقدار زمان نمی‌تواند چالش بزرگی برای چشم‌پوشی از این روش شناسایی باشد.

جدول ۲: شبیه‌سازی در نسبت سیگنال به نویز۱dB و کانال AWGN

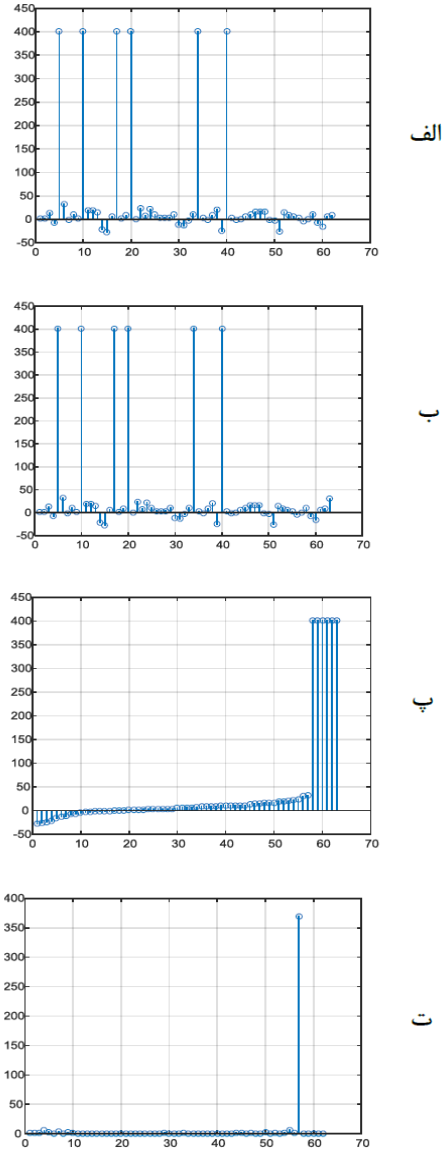
طول کد	متوسط زمان تشخیص		درصد تشخیص	
	تصمیم نرم	تصمیم سخت	تصمیم نرم	تصمیم سخت
(31, 29)	66.06	18.65	44%	16%
(64, 57)	112.23	33.72	41%	10%

جدول ۳: شبیه‌سا زی در نسبت سیگنال به نویز ۲,۲dBو کانال AWGN

طول کد	متوسط زمان تشخیص		درصد تشخیص	
	تصمیم نرم	تصمیم سخت	تمصمیم نرم	تصمیم سخت
(31, 29)	65.88	18.46	100%	68%
(64, 57)	108.42	32.41	95%	62%



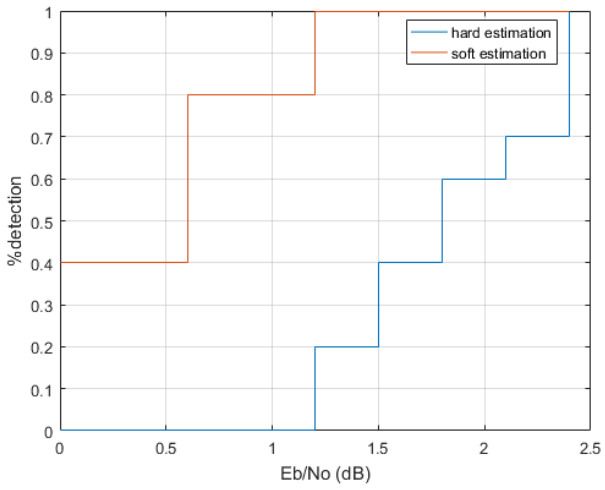
شکل۶: شناسایی ریشه‌های چندجمله‌ای مولد مربوط به بعد اول



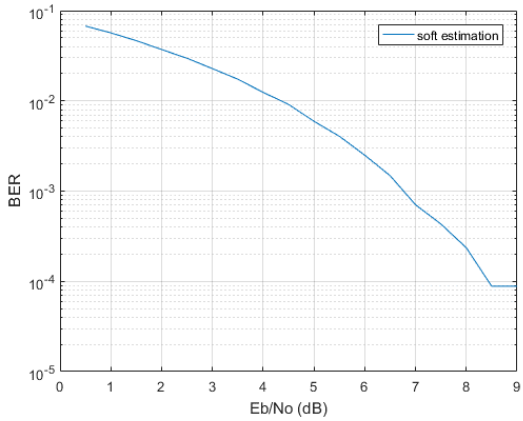
شکل ۷: شناسایی ریشه‌های چندجمله‌ای مولد مربوط به بعد دوم

برای بیان بهبود عملکرد پیشنهادی نمودار شکل ۸ نمایش دهنده یک کد ضربی (32, 26) × (32, 26) بروی کانال AWGN با مدولاسیون BPSK است. این نمودار، بهبود میزان شناسایی در الگوریتم تصمیم نرم به نسبت الگوریتم تصمیم سخت در نسبت سیگنال به نویزهای مختلف را نمایش داده است به‌طوری‌که در SNR=1dB این میزان اختلاف حدود ۸۰٪ است. از حد آستانه‌ای که برای هر الگوریتم متفاوت است به بعد تمامی متغیرهای کد به‌درستی شناسایی می‌شود که این حد آستانه برای تصمیم نرم 1dB و برای تصمیم سخت 2.5dB است.

اگر یک گیرنده غیرمجاز به‌منظور شنود داده تبادل شده بین فرستنده و گیرنده مجاز بخواهد عمل‌کننده توسط الگوریتم تصمیم نرم ارائه شده در مقاله می‌تواند در نسبت سیگنال به نویز بیش از 1dB با بهره کدگذاری نمایش داده شده در شکل ۹ همانند یک سیستم مجاز از اطلاعات تبادلی بهره ببرد.



شکل ۸: درصد تشخیص صحیح پارامترهای کد ضربی



شکل ۹: میزان خطای شناسایی در الگوریتم تصمیم نرم

۴- نتیجه‌گیری

در این مقاله روش آنتروپی برای شناسایی کدهای توربو ضربی مبتنی بر تصمیم‌گیری سخت معرفی شد سپس الگوریتم آنتروپی توسعه داده شد و برای شناسایی کدهای توربو ضربی مبتنی بر تصمیم‌گیری نرم مورداستفاده قرار گرفت.

الگوریتم آنتروپی به این صورت است که برای کدهای توربو ضربی در حالت سنکرون و غیر سنکرون اجرا می‌شود. حالت سنکرون به این معنی است که محل هم‌زمانی، بیت اول کلمه کد است و در مراحل اجرای الگوریتم نیاز به جستجو برای یافتن محل هم‌زمانی نیست.

حالت غیر سنکرون به این معنی است که محل هم‌زمانی، اولین بیت کلمه کد نیست و در مراحل اجرای الگوریتم، طول کلمه کد و محل هم‌زمانی مجهول هستند و باید شناسایی شوند. در حالت غیر سنکرون سرعت الگوریتم به‌صورت قابل‌توجه کاهش می‌یابد که این امر بدیهی است. به ازای طول کلمه کد L، مقدار محل هم‌زمانی می‌تواند از یک الی طول کلمه کد تغییر کند. محل هم‌زمانی برای بعد اول می‌تواند با محل هم‌زمانی بعد دوم متفاوت باشد. در مراحل شبیه‌سازی الگوریتم، محل هم‌زمانی بعد اول با T_1 و محل هم‌زمانی بعد دوم با T_2 نشان داده می‌شود.

در این مقاله یک قابلیت دیگر به قابلیت‌های الگوریتم آنتروپی اضافه شد. به‌عبارت‌دیگر علاوه بر قابلیت شناسایی طول کلمه کد و محل هم‌زمانی، قابلیت شناسایی مقدار کوتاه‌شدگی و قدرت تصحیح خطا نیز به الگوریتم اضافه شده است. به ازای طول کلمه کد L، مقدار کوتاه‌شدگی برای بعد اول و دوم می‌تواند از مقدار یک تا مقدار $\frac{L}{2}$ تغییر کند. مقدار کوتاه‌شدگی برای بعد اول با S_1 و برای بعد دوم با S_2 نمایش داده می‌شود.

نتایج آزمون‌های مختلف نشان می‌دهد که در شرایط آزمون یکسان و برای طول کلمه کد و SNR یکسان، الگوریتم آنتروپی مبتنی بر تصمیم‌گیری نرم دقت بهتری دارد و پارامترهای کدگذاری را با دقت بیشتری شناسایی می‌کند، البته سرعت آن به‌مراتب کمتر است.

مراجع

[1] Y. Xiaojing and W. Niancheng, “Recognition method of BCH codes based on roots information dispersion entropy and roots statistic,” Journal of Detection & Control, vol.32, no.3, pp.69–73, 2010.

[2] J. Barbier, G. Sicot, and S. Houcke, “Algebraic approach for the reconstruction of linear and convolutional error correcting codes”, Proc. 3rd International Conference on Computer Science and Engineering CISE 2006, pp.66-71, 2006.

[3] J. Barbier and J. Letessier, "Forward error correcting codes characterization based on rank properties," in proc.Int. Conf. Wireless Communications & Signal Processing, 2009, pp.1.-5.

[4] J.Wang, Y.Yue, and J.Yao, “Statistical Recognition Method of Binary BCH Codes”, Communication and Network, vol. 3, p. 17, 2011.

[5] G.Bangning, Z.XiangWen, D.Guo "Blind Recognition of BCH Code Based on Galois Field Fourier Transform" 978-1-4673-7687-7/15/$31.00 IEEE a©2015.

[6] Z.Jing, H.Zhiping, S.Shaojing, “Blind Recognition of Binary Cyclic Codes” EURASIP Journal on Wireless Communication and Network. Vol.2013, pp.1-17, 2013.

[۷] مهدی تیموری، حمید رضا کاکایی، مرتضی حدادی، « شناسایی کور کدهای ضربی BCH » مجله مهندسی برق دانشگاه تبریز، جلد۴۷، شماره ۱، بهار ۱۳۹۶.

[۸] احمد قلی زاده سوته، حسین خالقی بیزکی، « تخمین پارامترهای کد BCHباینری در شرایط نویزی با استفاده از روش مبتنی بر بررسی بیت‌های توازن» مجله مهندسی برق دانشگاه تبریز، جلد۴۷، شماره ۱، بهار ۱۳۹۶.

[9] N. Wen, X. Yang,” Recognition methods of BCH codes”. Electron. Warf. 6, 30–34, 2010.

[10] X. Yang, N. Wen, “Recognition method of BCH codes on roots information dispersion entropy and roots statistic”. J. Detect. Control. 3, pp 69–73, 2010.

[11] Jing Zhou, Zhiping Huang, Chunwu Liu, Shaojing Su and Yimeng Zhang “ information dispersion entropy Based Blind Recognition of Binary BCH Codes in Soft Decision”. Situation School of Mechatronics Engineering, National University of Defense Technology. 2013.

[12] J.Hagenauer, E.Offer, and L.Papke;” Iterative decoding of binary block and convolutional codes”. IEEE Trans. Inf. Theory 2, pp 429–445, 1996.

[13] J. Jiang, K.R. Narayanan, “Iterative soft-input-soft-output decoding of Reed-Solomon codes by adapting the parity check matrix”. IEEE Trans. Inf. Theory. 8, pp3746–3756; 2006.

[14] CCSDS 130.1-G-2 “TM Synchronization and Channel Coding--Summary of Concept and Rationale”. Green Book. Issue 2. November 2012.

زیرنویس‌ها

[1] Generator Polynomial
[2] Primitive Polynomial
[3] Turbo Product Codes
[4] Matrix Rank
[5] Bose - Chaudhuri – Hocquenghem
[6] Greatest Common Divisor
[7] Root Information Dispersion Entropy and Root Statistic
[8] Minimal Parity Check Matrix
[9] Binary Symmetric Channel
[10] Log-Likelihood Ratio
[11] Consultative Committee for Space Data Systems
[12] Intelsat Earth Station Standards